

Il mondo della psicologia è oggi in evoluzione continua, grazie alla sempre maggior presenza di nuove tecnologie dell'informatica e della comunicazione. Lo psicologo contemporaneo deve fare i conti con sfide sempre maggiori e opportunità di studio e di ricerca fino a pochi anni fa impensabili. Elementi di Psicometria Computazionale, rappresenta un prezioso strumento per formare gli psicologi di domani, attenti al passato, orientati al futuro e con una corposa conoscenza del presente. In modo pratico e semplice il volume accompagna il lettore all'uso degli strumenti della misura in psicologia, alla luce delle più recenti tecnologie. Un percorso che parte dall'acquisizione dei dati con questionari elettronici, biosensori, social networks, realtà virtuale e altro, per proseguire con la gestione avanzata dei dati e chiudere con una prima introduzione ai modelli computazionali. Caratteristica importante del volume è il totale orientamento al mondo open source e la costante disponibilità di software gratuito per tutti gli strumenti utilizzati.

Il volume include materiali supplementari gratuiti su Metricamente.it tra cui:

- i software gratuiti a licenza aperta (la maggior parte di quelli usati)
- tutto il codice realizzato e utilizzato nel volume
- un corso pratico online completo e gratuito
- piattaforma per realizzare questionari elettronici online gratuita
- altri corsi, link e informazioni sulla psicometria computazionale
- esercitazioni pratiche



Pietro Cipresso è autore di oltre 100 pubblicazioni scientifiche nell'ambito della Psicometria Computazionale. Statistico, Dottore di ricerca in Comunicazione e nuove tecnologie, a indirizzo Psicologia generale, Ricercatore all'Istituto Psicologico Italiano e Assegnista di ricerca in Psicometria presso l'Università Cattolica di Milano. È stato Visiting Researcher in centri internazionali, come il Massachusetts Institute of Technology (MIT). Tra i libri internazionali dell'autore "Modeling Emotions at the Edge of Chaos", "Computing Paradigms for Mental Health", "The Psychology of Social Networking" e "Virtual Reality: Technologies, Medical Applications and Challenges".



€ 36.96



Pietro Cipresso



ELEMENTI DI PSICOMETRIA COMPUTAZIONALE

Pietro Cipresso



ELEMENTI DI
PSICOMETRIA
COMPUTAZIONALE



**ELEMENTI DI
PSICOMETRIA
COMPUTAZIONALE**

Pietro Cipresso



Eccetto dove diversamente indicato il volume "Elementi di Psicometria Computazionale" di Pietro Cipresso è distribuito con Licenza Creative Commons Attribuzione - Non opere derivate 2.0. Quello di seguito rappresenta un riassunto in linguaggio accessibile a tutti (e non un sostituto) della licenza Creative Common. Per informazioni:

<http://creativecommons.org/licenses/by-nd/2.0/legalcode>

Tu sei libero di:

- **Condividere** — riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare questo materiale con qualsiasi mezzo e formato
- per qualsiasi fine, anche commerciale.
- Il licenziante non può revocare questi diritti fintanto che tu rispetti i termini della licenza.

Ai seguenti termini:

- **Attribuzione** — Devi **attribuire adeguatamente la paternità sul materiale**, fornire un link alla licenza e **indicare se sono state effettuate modifiche**. Puoi realizzare questi termini in qualsiasi maniera ragionevolmente possibile, ma non in modo tale da suggerire che il licenziante avalli te o il modo in cui usi il materiale.
- **NoDerivatives** — Se **remixi, trasformi il materiale o ti basi su di esso**, non puoi distribuire il materiale così modificato.
- **Divieto di restrizioni aggiuntive** — Non puoi applicare termini legali o **misure tecnologiche** che impongano ad altri soggetti dei vincoli giuridici su quanto la licenza consente loro di fare.

Note:

- Non sei tenuto a rispettare i termini della licenza per quelle componenti del materiale che siano in pubblico dominio o nei casi in cui il tuo uso sia consentito da una **eccezione o limitazione** prevista dalla legge.
- Non sono fornite garanzie. La licenza può non conferirti tutte le autorizzazioni necessarie per l'uso che ti prefiggi. Ad esempio, diritti di terzi come **i diritti all'immagine, alla tutela dei dati personali e i diritti morali** potrebbero restringere gli usi che ti prefiggi sul materiale.

ISBN: 978-1-291-72944-3

Copyright: © 2014 di Pietro Cipresso. Tutti i diritti riservati.

Distribuito da: Lulu Press, Inc. Tutti i diritti riservati

A Silvia, Federica e Fabrizia

Indice

PREFAZIONE	13
PARTE I · METODOLOGIA COMPUTAZIONALE	19
Capitolo 1 Dal carta e matita alle nuove tecnologie	19
Capitolo 2 Dati psicofisiologici	55
Capitolo 3 Movimenti oculari	77
Capitolo 4 Dati comportamentali.....	91
Capitolo 5 Dati relazionali.....	137
Capitolo 6 Dal Reale al Virtuale	155
PARTE II · GESTIONE AVANZATA DEI DATI	183
Capitolo 7 Software per la gestione avanzata dei dati	185
Capitolo 8 Scalari, Vettori e Matrici.....	209
Capitolo 9 Operazioni multiple generando codice.....	251
Capitolo 10 Plotting dei dati	279
Capitolo 11 Gestione di dati complessi	303
Capitolo 12 Organizzazione strutturale dei dati complessi.....	339
Capitolo 13 Gestire e organizzare i dati relazionali.....	365
Capitolo 14 Computazione dati in realtime	391
Capitolo 15 Cenni di signal processing	413

PARTE III · MODELLI COMPUTAZIONALI.....	487
Capitolo 16 Introduzione ai Sistemi Complessi	489
Capitolo 17 Modellizzazione di Sistemi Complessi.....	525
Capitolo 18 Modelli basati su Agenti.....	539
Capitolo 19 Microanalisi di simulazione tra agenti	571
Capitolo 20 Un modello computazionale in reti complesse	587
References	621
Indice analitico.....	671
Note di copyrights	695
Informazioni sul materiale supplementare al volume	695

Prefazione

Laudamus veteres, sed nostri utemur annis
(Lodiamo i tempi antichi, ma sappiamoci muovere nei nostri)

Ulteriora mirari, praesentia sequi
(Guardare al futuro, stare nel proprio tempo)

L'idea di questo volume nasce dalla richiesta di un considerevole numero di Colleghi, Studenti e curiosi che ormai da tempo hanno capito che dietro le mie elaborazioni relative a diversi miliardi di dati non esistono trucchi magici o computer ultrapotenti, ma solo una raccolta di conoscenze che, se opportunamente combinate, danno vita alla psicometria computazionale.

Il volume vuole dunque essere uno strumento didattico e di ricerca per chi ha l'esigenza di lavorare con dati psicometrici di una certa complessità. Non è dunque un manuale di psicometria classica e come tale non contiene informazioni su come analizzare statisticamente un dataset. Chi legge questo volume ha come obiettivo arrivare a costruire quel dataset che resterà comunque oggetto di analisi tramite la psicometria classica.

Qualche anno fa un volume come questo non avrebbe potuto essere neanche concepibile. Oggigiorno, invece non esiste psicologo che a un certo punto di una raccolta dati di qualche tipo non senta l'esigenza di trovare risposte concrete nell'ambito della psicomетria computazionale.

I primi quattordici capitoli del volume possono essere letti da chiunque abbia un minimo di competenze informatiche e matematiche di base, anche se una conoscenza dell'algebra lineare e l'esperienza con un qualunque linguaggio di programmazione è sicuramente un valido aiuto nella comprensione delle parti più avanzate. In particolare nella prima parte si presentano i principali strumenti di raccolta dei dati complessi, per poi trattarne la gestione avanzata nella seconda parte. Dal capitolo 15 al 20 gli argomenti sono più complessi e richiedono competenze matematiche più avanzate e sono per loro natura e specie più adatti a chi voglia iniziare a muovere i primissimi passi per diventare psicometrista computazionale.

Nel cominciare a scrivere questo volume ho anche tenuto presente un'altra importante esigenza sempre sottolineata da parte degli psicologi, ovvero quella di avere accesso a strumenti gratuiti e utilizzabili su un normale computer moderno.

Ho cercato di rispondere a questa esigenza quanto più possibile ed è stato un percorso spesso assai complesso. Posso oggi affermare che è possibile coprire tutte le aree trattate nel volume con dei software gratuiti, dei quali trovate sempre il link al sito web di riferimento.

Ma mi sono voluto spingere molto oltre e ho reso chiari, gratuiti e totalmente accessibili tutti i programmi e gli script che ho usato nel volume e che dunque il lettore potrà usare per la gestione dei propri dati. Il codice che ho prodotto, soprattutto quello relativo alla gestione dei dati complessi, è dettagliatamente commentato per spiegare passo dopo passo ogni procedimento. Inoltre non ho semplicemente scritto il codice, ma l'ho provato su file, matrici e dati che ho reso disponibili come esempi. Ogni riga di codice è dunque rigorosamente testata e fornita gratuitamente. In psicologia, lo sappiamo bene, trattiamo con le persone e non possiamo permetterci di essere superficiali neanche negli esempi. Tutto questo materiale supplementare al volume (codici, programmi e piattaforme operative) è reso disponibile a tutti sul sito MetricaMente.it insieme a un corso gratuito con circa 300 slide che coprono soprattutto la gestione dati trattata nella seconda parte del volume in modo più colloquiale e dinamico. Sul sito MetricaMente.it sono anche disponibili gratuitamente altri corsi di base e avanzati, che spero renderanno questo volume dinamico e aperto a tutti. Da MetricaMente.it è anche possibile accedere a una piattaforma gratuita per i questionari elettronici, alla piattaforma di eLearning e all'area download per i materiali supplementari al volume. Potrete seguire i miei corsi online ed entrare in contatto con me, sperabilmente anche per critiche e suggerimenti che possano rendere questo volume veramente dinamico, nello spazio e nel tempo.

Veniamo adesso ai ringraziamenti. Credetemi, non basterebbero mille pagine per ringraziare tutti. Davvero. So già che mi scorderò moltissime persone, scusatemi, così son fatto, mi conoscete e sapete bene che non è cattiveria, solo sbadataggine.

Sin da quando ho iniziato a fare ricerca, il mio obiettivo è sempre stato quello di trasmettere le mie conoscenze. Silvia Serino, Federica Pallavicini e Fabrizia Mantovani hanno assorbito più di altri le cose che racconto in questo volume, ascoltandomi pazientemente e segnalandomi continuamente cosa non poteva essere chiaro a chi non aveva una formazione matematico-informatica, anche quando loro avevano capito pienamente. Di tanto vi sono grato e a voi dunque va il merito di aver avviato questo mio lavoro e soprattutto di averlo perfezionato con suggerimenti preziosi. Silvia, Federica e Fabrizia non si sono limitate a raccogliere i frutti della mia conoscenza, ma hanno cercato di capire come poter fare anche loro almeno una parte di quello che facevo io e questo fa la differenza. Non sono le mie figlie (come la dedica poteva lasciar pensare!) ma saranno per loro il miglior modello da seguire e dunque a loro è dedicato questo volume, in rappresentanza ufficiale di tutti quelli che mi hanno permesso quel processo di aperta trasmissione delle mie conoscenze che tanto mi è caro. *"A Silvia, Federica e Fabrizia"* è stata la prima cosa che ho scritto, prima ancora di dare un titolo al volume e lì è giusto che rimanga per sempre. Grazie.

Grazie agli studenti tutti, imparo tantissimo anche da voi. Grazie.

Nella stessa direzione vanno i miei maestri di vita. Partendo da quando sono nato, ovviamente, la mia famiglia, sempre unita nonostante le avversità della vita e sempre pronta, incluso mio fratello, autore della copertina e vittima delle mie richieste tecniche su database, php e tante altre cose che gli ho insegnato io e adesso sa meglio di me. Papà, Mamma, Sergio, Adele e me. Una famiglia. Sempre. Grazie. Giorno dopo giorno, la famiglia, mi ha educato ai valori più importanti e mi ha soprattutto insegnato che i soldi sono sempre l'ultima cosa ed ecco perché il libro è con licenza Creative Common (<http://creativecommons.org/licenses/by-nd/2.0/legalcode>), è infatti gratuitamente scaricabile, stampabile e venduto in copia cartacea a prezzo di costo. Un libro per tutti.

Continuo i ringraziamenti in ordine cronologico.

Maestro di vita, durante la mia prima formazione accademica, è Donato Michele Cifarelli, il più grande esperto di statistica matematica che io abbia mai conosciuto e non sono l'unico a dirlo. Le cose che lui mi ha insegnato continuano a stupire i miei colleghi esperti in materia. E' tutto merito suo, non mio. Grazie.

Maestro di vita, durante il mio dottorato di ricerca, è Marco Alessandro Villamira, medico, psicoterapeuta e sicuramente tra i più importanti esperti in psicologia della complessità e molte altre cose, che peraltro non finirà mai di stupirmi per le sue conoscenze che vanno dal latino, alla fisica quantistica, passando per il biofeedback e la psicologia della comunicazione. A lui devo dire grazie innanzitutto per avermi dato fiducia, accogliendomi nel suo laboratorio. Nell'ambito del dottorato di ricerca, Marco, mi ha

fatto studiare una quantità innumerevole di libri e articoli di psicologia e mi ha insegnato a leggerli con i miei occhi e trarne nuove idee alla luce delle mie competenze. Marco mi ha anche insegnato che la comunicazione è complessa. Grazie.

Maestri di vita come Pawan Sihna ed Emery Brown, ne ho trovati pure nella mia esperienza al Massachusetts Institute of Technology (MIT), al Massachusetts Medical Hospital (MGH) e all'Harvard Medical School (HMS). Mi hanno accolto con amicizia pur non conoscendomi e mi hanno più volte invitato a uscire, a passare weekend insieme e fare molte attività extra - accademiche. A casa di professori Emeriti di Harvard ed MIT, ho ricevuto un'accoglienza splendida, tanto da lasciarmi veramente sorpreso. In ambito accademico mi hanno lasciato sperimentare e partecipare a molte ricerche e hanno sempre cercato di imparare i miei trucchi del mestiere, insegnandomi i loro. In questo contesto un particolare ringraziamento a Luca Citi e a Riccardo Barbieri, italiani all'estero, come tanti altri, persone eccezionali che seppur lontani da decenni, non hanno perso il meglio dell'essere italiani, prima d'ogni cosa i valori.

Maestro di vita, dopo il dottorato di ricerca, è Giuseppe Riva che mi ha accolto come uno di famiglia e mi ha supportato come nessun altro prima, fa così con tutti, è un grande e va detto. Non è certo morbido, ma i giovani li porta avanti con tutte le sue forze. Forse perché è giovane anche lui o forse perché ha capito che nella ricerca bisogna soprattutto divertirsi, con professionalità, rigore scientifico e pragmaticità. A Beppe devo dire grazie, non solo per

questo libro, ma proprio per tutto, non sarei qui se non avesse creduto in me. Grazie.

Maestro di vita e compagno di grandi avventure è Andrea Gaggioli con il quale mi lancio sempre in discorsi meravigliosi. Andrea ha capito quanto fosse importante la psicomatria computazionale ben prima di me e non si tira mai indietro di fronte a nuove entusiasmanti sfide tecnologiche o matematiche, anzi ne è sempre affascinato. Andrea mi ha insegnato miliardi di cose, sempre con simpatia ed eleganza.

Maestra di vita, nella mia avventura di ricerca all'Università Cattolica di Milano, è Margherita Lanz, che mi ha accolto in uno splendido gruppo, insieme al quale stiamo facendo interessantissimi lavori sulla social network analysis applicata alla psicologia sociale. Lavorare con Margherita mi sta aprendo la mente verso nuove prospettive, che non avevo considerato e devo ringraziarla perché riesce a dare una prospettiva teorica maggiormente legata all'individuo come entità complessa piuttosto che all'emergenza sistemica in quanto tale. Grazie.

Maestra di vita e parte del gruppo di Margherita, è Semira Tagliabue, che ha scritto anche un bellissimo libro su come trattare i dati (Psicologia ed Excel) che a tutti gli effetti è da considerare un precursore di questo volume. La sua esperienza sull'insegnamento della metodologia della ricerca e i discorsi sulle necessità degli studenti che ha condiviso con me, sono state fonte di ispirazione. Grazie.

Nello stesso gruppo, anche Sara Alfieri e Antonella Morgano, due giovani e brave ricercatrici che offrono al team sempre spunti critici e innovativi, animando e migliorando la discussione. Grazie. Maestra di vita, sempre in Università Cattolica, è Daniela Villani, amica carissima prima che collega. A lei va un ringraziamento speciale, per esser parte di quel gruppo di persone che non si limita a lavorare con me, ma prima mi chiede come sto e poi ascolta perfino la risposta. E' raro, credetemi. Grazie. E grazie anche a Gaia, la meravigliosa figlia appena nata, in rappresentanza di tutte le future generazioni.

Ci sono molte altre persone all'Università Cattolica che devo ringraziare per avermi dato affetto, supporto, spunti di riflessione, consigli e molto altro. Alcuni professori mi si rivolgono sempre con affetto genitoriale e mi fanno sentire parte di una famiglia. In rappresentanza di questa affettuosa categoria, Alessandro Antonietti ed Elena Marta.

Grazie davvero per tutto anche a Guendalina Graffigna, Chiara Libreri, Julia Menichetti, Claudio Bosio, Carlo Galimberti, Daniela Barni, Claudia Repetto, Daniela Marzana e molti altri colleghi del Dipartimento di Psicologia dell'Università Cattolica.

Grazie anche a Iafrate Raffaella, Gennari Maria Luisa e Silvio Ripamonti, che mi hanno supervisionato nel meraviglioso lavoro di tutoring delle prove finali degli studenti di laurea triennale. Grazie anche ai colleghi tutor. Seguire le prove finali rientra pienamente nel mio desiderio di trasmissione libera e aperta della conoscenza, che parte proprio da chi è in formazione: gli studenti.

Grazie a quei ricercatori dell'Istituto Auxologico Italiano che hanno accolto le mie idee sulla psicomетria computazionale con favore, entusiasmo e curiosità, ne cito solo alcuni che mi hanno fornito spunti creativi per questo volume: Marco Stramba Badiale, Giovanni Albani, Enrico Molinari, Gianluca Castelnuovo, Alessandro Mauro, Gianfranco Parati, Luigi Sellitti, Vincenzo Silani e molti altri. Grazie davvero per il supporto, la comprensione e la stima, a tutto il personale dell'Istituto Auxologico Italiano e in particolare Luca Grappiolo, il direttore della segreteria scientifica e il Prof. Zanchetti, direttore scientifico, riconosciuto tra i più grandi scienziati italiani nel mondo, ma soprattutto uomo di stile come ne sono rimasti pochi.

Un ringraziamento molto particolare va anche alla mia squadra di rugby "Amatori Rugby Milano", dei veri e propri guerrieri, tutti più giovani e bravi di me, che mi hanno accolto tra loro come un fratello e con pazienza mi aiutano a crescere, nello spirito del rugby. Forza Tori !!

Ringraziamenti simili ai "ragazzi" del Judo del "Dojo Jigoro Kano di Milano", tutti dalla grandissima esperienza, sto imparando tantissimo anche da voi.

Per chiudere i ringraziamenti, con un classico "ultimi, ma decisamente non ultimi, anzi!!", gli amici del lab, quelli di tutti i giorni, con i quali ci si rimbecca le maniche e lavoro, lavoro, lavoro, anche quando la situazione è dura, difficile e soprattutto precaria. Ho già ringraziato Silvia Serino e Federica Pallavicini, ma valeva la pena ritornarci. Grazie. Poi, Chiara Scaratti, una

terapeuta con i fiocchi, dalle meravigliose doti, sia in termini di empatia (e simpatia), sia come capacità intellettive; Luca Morganti, con il quale ci lanciamo in avventure di psicologia dello sport, anche lui bravissimo terapeuta; Elisa Pedroli, bravissima neuropsicologa bionda dalle grandi conoscenze, a dimostrazione che le bionde possono essere anche molto intelligenti; Cinzia Vigna, con la quale mi faccio sempre affettuose litigate per contrasti latenti tra "fuffa" e scienza; infine una serie di giovani che sono transitati per brevi o lunghi periodi e che comunque hanno lasciato il segno, Guglielmo Puglisi, Alessandra Grassi, Sofia Mardegan, Federica "Cameron" Solca, Tomaso Vido de Zaccaria, Federico Giaimo, Stefano Triberti, Fabio Ferrara, Irina Bogani, Maddalena Volterrani, Valentina Virginia Zoffoli, che fu anche mia studentessa sette anni prima e veramente tanti altri. Grazie a tutti.

Pietro Cipresso

PARTE I

**METODOLOGIA
COMPUTAZIONALE**

Dal carta e matita alle nuove tecnologie

In questo primo capitolo affrontiamo uno dei più farraginosi scogli che ci richiederà un profondo sforzo: il cambiamento.

Sebbene i cambiamenti richiedono un impegno consistente, possono anche offrire grandi opportunità. Tra le grandi sfide di questo nuovo millennio, la tecnologia è sicuramente tra quelle più complesse soprattutto perché i cambiamenti sono profondi e continui e lo sviluppo in tal senso sembra essere senza fine.

Se le tecnologie cambiano in continuazione, qual è dunque l'utilità di apprendere nuovi concetti? Con questa domanda in mente ho voluto soffermarmi di seguito sulla metodologia legata alle tecnologie che gli psicologi di oggi e di domani possono fattivamente usare verso un approccio che, senza dimenticare gli insegnamenti della classica raccolta sperimentale, tenga conto delle opportunità che le nuove tecnologie possono offrire.

1.1 Dal questionario al questionario

I più classici mi vogliono scusare per la chiara provocazione contenuta nel titolo di questo paragrafo. Tuttavia mi sembrava doveroso sottolineare che, nonostante l'uso di nuove tecnologie, il classico questionario non cambia, anche se i puristi avranno da ridire in merito, ma di questo ce ne faremo una ragione dando ampi rimandi alla letteratura in merito.

In psicometria computazionale si tratta sia la presentazione del questionario, sia la raccolta dati e la loro gestione mediante l'ausilio di metodi e tecniche computazionali. Non sarà possibile nel breve spazio di tale volume dare una completa spiegazione di tutti gli strumenti oggi disponibili. In particolare cercheremo di trattare le argomentazioni più semplici e di immediato utilizzo, demandando a letture più specifiche i vari argomenti che sempre più spesso mi sento chiedere (quali i Big Data), ma che richiedono una più complessa preparazione informatica e matematica rispetto a quella richiesta in questo contesto. Cercherò comunque di affrontare temi più avanzati, quali l'uso di *database relazionali* e relativi linguaggi di programmazione in modo sempre accessibile ai non esperti nel campo. Di seguito faremo riferimento ai questionari in modo molto ampio, includendo sondaggi, test, indagini e in generale tutto ciò che richiede un qualche tipo di risposta.

1.2 Preparazione e presentazione del questionario

In generale, nel mondo computazionale (così come in tutti i mondi) esistono diverse prospettive, tra cui quella dell'utente finale e quella dello sperimentatore. La prospettiva dell'utente finale, parlando di questionari, è necessariamente legata al modo in cui questo viene somministrato. E' bene ricordare, innanzitutto, che è pur sempre possibile passare dal questionario elettronico a quello cartaceo in modo molto semplice: per mezzo di una stampa. Per quanto scontata possa sembrare questa affermazione, vi sono molte situazioni in cui la cara vecchia carta torna assai utile e non va perciò rinnegato il suo utilizzo, a volte assolutamente insostituibile (ad esempio ove i rispondenti non dispongano di supporti elettronici). Carta a parte, i dispositivi elettronici oggi disponibili sono accomunati, dal nostro punto di vista, dall'aver la possibilità di collegamento a Internet, ma sono molto diversi tra loro e meritano una piccolissima riflessione:

- PC, la forma più usata per rispondere a un questionario elettronico ricevuto tramite email, social networks o altro;
- Tablet, hanno il vantaggio di avere un monitor grande a sufficienza per poter compilare un questionario e sono sempre più usati;
- Smartphone, hanno lo svantaggio di avere un piccolo monitor, ma il vantaggio di essere a portata di mano quasi sempre e ovunque.

Ricordiamo che esistono anche altre possibili forme di "rete" oltre Internet, che possono essere usate per i questionari. Vale la pena citare a tal proposito la rete telefonica (viene ancora oggi molto usata per i questionari) e gli SMS (usati generalmente per brevi questionari).

Fatta questa breve premessa sulla presentazione del questionario, di seguito ci concentreremo sulla sua preparazione e in particolare sugli strumenti gratuiti oggi a disposizione per crearne di ogni tipo.

1.3 Per cominciare: questionari con Google Drive

Molti credono che il modo più semplice per fare un questionario elettronico è quello di prepararlo con un elaboratore di testi. Tuttavia questo è quanto di più sbagliato si possa pensare, per due ordini di motivi: il primo è che esistono servizi gratuiti più semplici e immediati, ma soprattutto già pensati e strutturati per creare i questionari. Il secondo motivo è che preparare un questionario elettronico non significa soltanto prepararne la visualizzazione, ma significa soprattutto strutturarne le risposte in apposite tabelle immediatamente disponibili per l'analisi dei dati.

Un servizio gratuito che risponde ad entrambi questi criteri è quello offerto da Google. Per capire appieno come creare i questionari con i documenti Google è sufficiente creare un account gratuito (su www.google.com), attivando così una "google drive"

ovvero una piattaforma cloud, in pratica una cartella online a cui potete accedere solo voi e chi voi vogliate (anche chiunque). Una volta attivata la Google Drive, sarà possibile accedere alle molte funzioni, app e servizi messi gratuitamente a disposizione da Google. Avrete a disposizione circa 15 Giga Byte (generalmente Google in modo gratuito aumenta periodicamente la quota a disposizione, che originariamente era infatti 1 Giga Byte).

Una volta aperta la Google Drive da <https://drive.google.com> cliccando sul pulsante "crea" e scegliendo "modulo" potrete creare un questionario secondo le vostre esigenze.

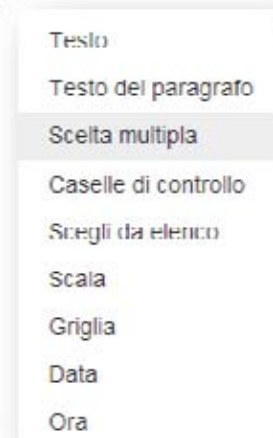
A screenshot of the Google Forms 'Esempio' (Example) form creation interface. The form is titled 'Esempio' and has a description field. The 'Titolo domanda' (Question title) field contains 'Eta'. The 'Tipo di domanda' (Question type) dropdown is set to 'Testo'. Below the question type, there is a section for 'Risposta utenti' (User responses) with a 'Convalida dati' (Validate data) section. In this section, there are checkboxes for 'Numero' (checked), 'Compreso fra' (checked), '10', 'e', '30', and a text box containing 'Questo questionario è rivolto a giovani tra i 10 e i 30 anni'. At the bottom, there is a 'Fare' button and a checkbox for 'Domanda obbligatoria' (checked). There is also an 'Aggiungi elemento' (Add element) button at the bottom left.

Figura 1.1. Creazione di un questionario con i moduli di Google Drive. Basterà indicare la domanda e scegliere il tipo di risposta.

Una volta creato il questionario sarà possibile diffonderlo immediatamente condividendo il link o indicando le email a cui spedirlo.

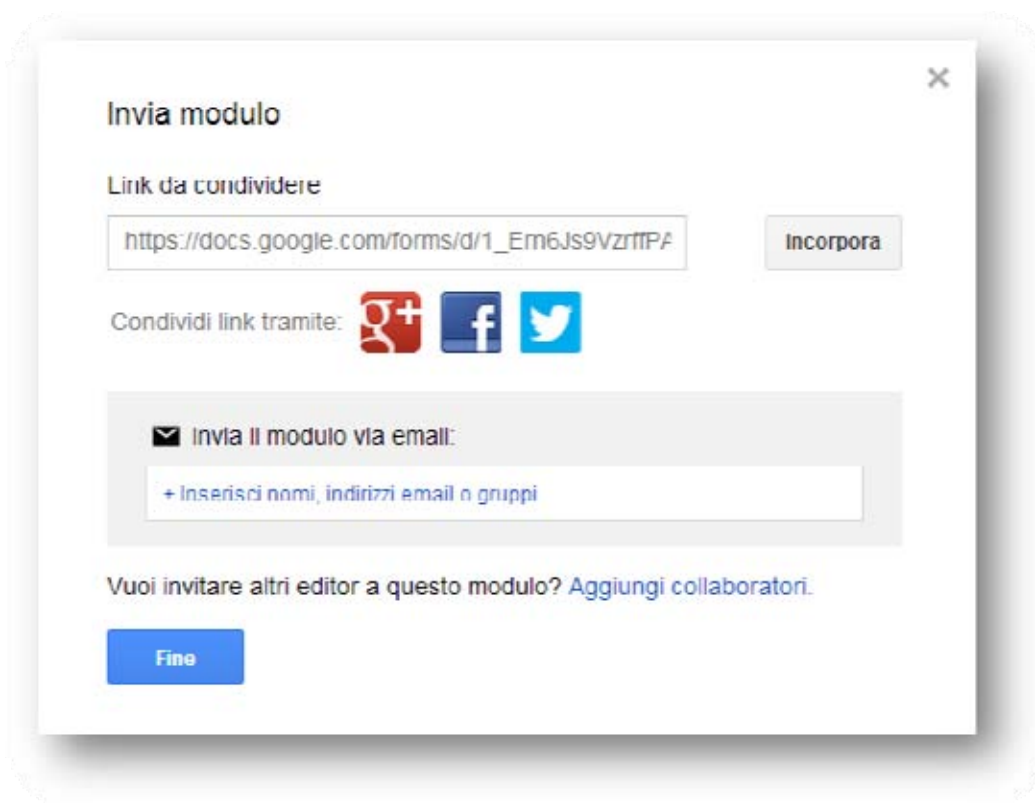


Figura 1.2. Condivisione del questionario mediante link o email.

Un'alternativa alla spedizione diretta del modulo è quella di inglobarlo in un blog appositamente creato (o in qualunque altra pagina web. Basterà cliccare su "incorpora" (vedi Figura 1.2) per generare in modo automatico il codice (vedi Figura 1.3).

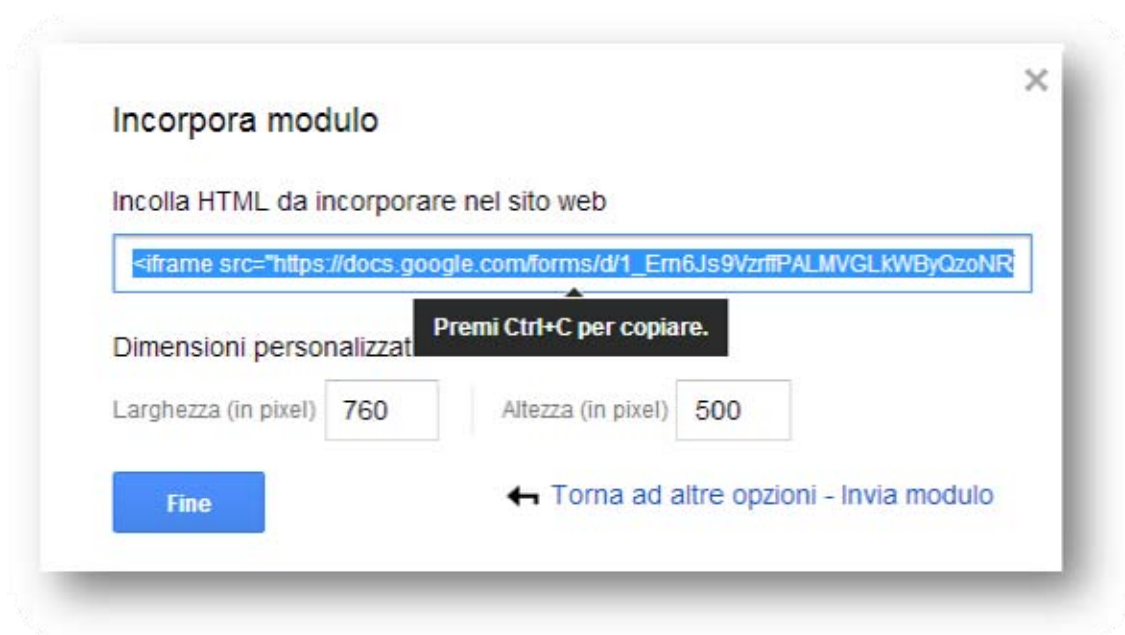


Figura 1.3. Generazione del codice da incorporare nel blog o in qualunque sito web di cui si dispongano i diritti.

Creare un blog per l'inserimento di un questionario è un'operazione assai semplice. Ad esempio, usando lo stesso account Google prima creato basterà andare su <https://www.blogger.com/>, cliccare sul pulsante "Nuovo blog", scegliere un titolo, uno stile e un nome per poi cliccare su "Nuovo post" (nella home page o su un'altra pagina) e infine incollare il codice precedentemente copiato (avendo cura di selezionare HTML come da Figura 1.4 in alto). L'operazione per incorporare il codice del questionario, qui effettuata su [blogger.com](https://www.blogger.com/), è possibile su quasi tutti i siti web oggi disponibili. In questo modo il nostro questionario potrà essere integrato in un sito web in modo molto semplice (Figura 1.4 in basso).



Figura 1.4. Incorporare il codice in una pagina del blog e risultato di anteprima della pagina del blog con il questionario.

Come già detto, una delle motivazioni a distribuire un questionario in formato elettronico è di poter disporre dei risultati derivanti in modo efficiente, immediato e in tabelle pronte alla successiva elaborazione statistica. Questa operazione con Google Drive è molto semplice: infatti il questionario archiverà le risposte su un apposito file nella stessa cartella (Figura 1.5 e 1.6).



Figura 1.5. Elenco dei due file (il modulo con il questionario e il foglio di calcolo con le risposte ottenute) in Google Drive.

The screenshot shows a Google Sheet titled 'Esempio (Risposte)'. The sheet contains a table with the following data:

Informazioni cronologiche	Età	Genere	Indica il tuo livello di Stress	Indica il tuo livello di Relax	Ti è piaciuto questo test?
01/01/2014 16:32:24	22	Maschio	3		
01/01/2014 16:33:15	24	Maschio	2		
01/01/2014 16:41:46	26	Maschio	4	5	Si, molto
01/01/2014 16:42:12	23	Maschio	4	5	No, per niente
01/01/2014 16:42:48	34	Maschio	3	5	No, per niente
01/01/2014 16:44:07	27	Femmina	5	5	Moderatamente
01/01/2014 16:44:50	26	Maschio	5	5	No, per niente
01/01/2014 16:44:56	27	Femmina	2	5	Si, molto
01/01/2014 16:52:17	27	Maschio	4	5	Si, molto
01/01/2014 17:03:29	26	Maschio	4	5	Moderatamente

Figura 1.6. Il foglio di calcolo con le risposte ottenute.

Il foglio di calcolo con tutte le risposte ottenute può essere facilmente esportato in molti formati (Menu File > Scarica come) quali Microsoft Excel o txt, per poi essere importato in un programma statistico per la successiva analisi.

La spiegazione che abbiamo appena dato di Google Drive è ovviamente una semplice introduzione e non è certo qui il contesto per renderla esaustiva. Si invita dunque il lettore a voler andare su <https://support.google.com/drive/> per ulteriori approfondimenti e chiarimenti in merito. Quanto qui mostrato è al solo scopo illustrativo, per far comprendere come strumenti molto semplici possono rilevarsi molto utili per creare dei questionari elettronici molto efficienti.

1.4 Questionari avanzati con LimeSurvey

Come abbiamo visto, tramite Google Drive è possibile creare dei questionari in modo semplice e veloce. Esistono molte opportunità nel fare una versione elettronica di un questionario, ma che tuttavia non sono implementabili con la piattaforma di Google. Servono dunque piattaforme, che seppur più complesse, offrono maggiore flessibilità e maggiori possibilità di personalizzazione. Ad esempio funzioni spesso richieste ed assai utili in un questionario elettronico sono la possibilità di randomizzare un set di domande o quella di condizionare la risposta a delle funzioni complesse o ancora la ramificazione degli item sulla base delle

risposte ricevute. E' proprio sulle funzioni più sofisticate che si comprende l'utilità del questionario elettronico. Il poter limitare la risposta a valori obbligati o richiedere l'obbligatorietà di una risposta si riflette in una minore presenza di dati mancanti e - soprattutto - in una maggiore spiegazione fornita al lettore, che in questo modo viene corretto su errore, avviando di fatto un processo di apprendimento per prove ed errori.

Esistono sul mercato molte piattaforme elettroniche per creare questionari avanzati e molte di queste sono di eccellente qualità.: ad esempio SurveyMonkey (<https://it.surveymonkey.com>) e Qualtrics (<http://qualtrics.com>) sono ottime piattaforme per realizzare questionari elettronici.

Come abbiamo già introdotto, lo scopo di questo volume è quello di offrire strumenti della psicomatria computazionale gratuiti e - ove possibile - open source (a codice aperto). Procedendo con questa logica, di seguito, verrà presentata una delle più potenti piattaforme gratuite per la realizzazione di questionari elettronici oggi disponibile: LimeSurvey (<http://www.limesurvey.org>).

Vorrei innanzitutto sottolineare che piattaforma gratuita non significa priva di costi. Qualunque software richiede dei costi di avvio, gestione e manutenzione anche se è gratuito. Non considerando i costi legati al tempo che il ricercatore deve dedicare a tale piattaforma, vanno calcolati i costi di acquisto o noleggio di un server web o quanto meno di uno spazio web. Generalmente si tratta di costi assai limitati, nell'ordine delle 50 o

100 Euro l'anno, a patto che non si parli di migliaia o anche milioni di utenti.

LimeSurvey necessita di almeno 80 MB di spazio su disco (spazio web) e di un database MySQL 4.1.0 o successivo oppure Microsoft SQL Server 2005 o successivo oppure Postgres 8.1 o successivo. Il server deve avere PHP 5.1.6 o successivo con i alcuni moduli e librerie installate e abilitate: libreria mbstring (Multibyte String Functions), PDO database driver per MySQL o Postgres (pdo_pgsql) o MSSQL. Inoltre le principali librerie standard di PHP devono essere abilitate. Per maggiori dettagli meglio guardare l'apposito sito web. Molti provider di web hosting forniscono la compatibilità a LimeSurvey, in ogni caso basta chiedere prima di noleggiare lo spazio.

In generale la maggior parte delle piattaforme web che possono essere usate nell'ambito della psicomетria computazionale richiedono queste caratteristiche di spazio web. In particolare potrebbe essere un buon investimento prendere uno spazio web illimitato a basso costo con un set di Database MySQL e cominciare a fare qualche esercizio, ma per iniziare suggerisco di usare una distribuzione Apache, da installare sul proprio computer, contenente MySQL, PHP e Perl. Una volta diventati più pratici sarà possibile installare piattaforme di ricerca sempre più avanzate e non solo per i questionari, ma anche per eLearning (come moodle) o per siti web avanzati (come Joomla) e molto altro. Nel capitolo 12 daremo tutti i dettagli in merito all'installazione e alla gestione di diversi database e relativi software.

Di seguito, invece, vedremo come configurare e usare LimeSurvey. Al primo avvio, dopo aver definito la lingua (tra le oltre 50 disponibili!), il pannello di amministrazione di LimeSurvey si presenta all'utente con semplici e intuitive icone, come possiamo vedere in Figura 1.7. Inoltre, soffermandosi con il puntatore del mouse sopra un'icona, comparirà una breve descrizione dell'azione associata, aiutando in modo semplice e intuitivo la creazione e la gestione del questionario.



Figura 1.7. Pannello di amministrazione di LimeSurvey.

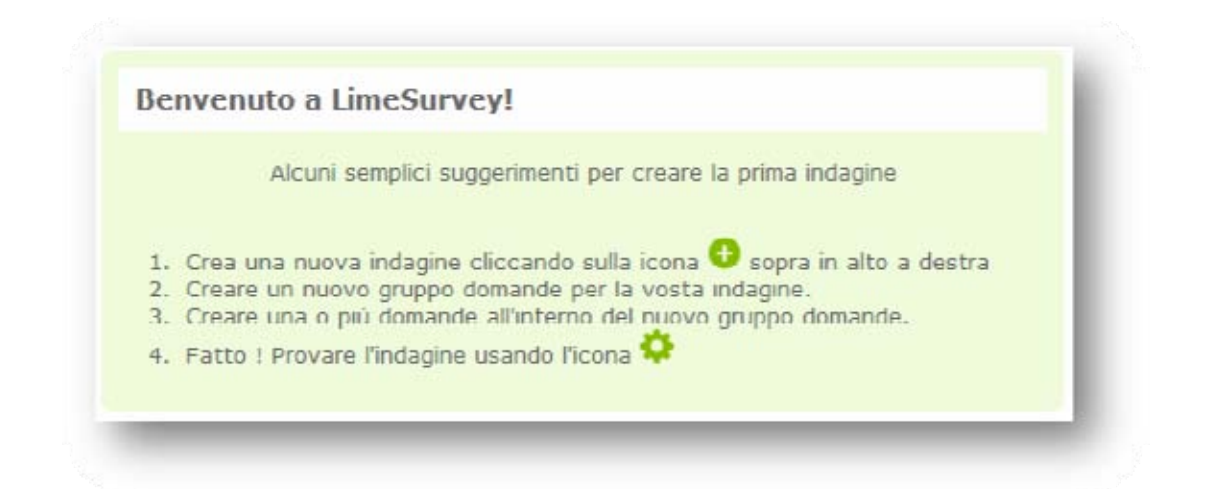


Figura 1.8. Finestra di benvenuto a LimeSurvey con utili suggerimenti e indicazioni.

I quattro punti esecutivi spiegati in Figura 1.8 dimostrano quanto sia semplice creare un questionario in LimeSurvey. La logica è infatti molto semplice: creare un'indagine (il nostro questionario), dentro ogni indagine creare gruppi di domande (ogni gruppo corrisponderà a una differente pagina web) e dentro ogni gruppo creare tutte le domande che vogliamo.

LimeSurvey ci darà ad ogni passo tutte le necessarie indicazioni e gli elementi mancanti, configurandosi in tal modo come uno strumento versatile ma anche molto semplice da usare e configurare (Figura 1.9).



Figura 1.9. La schermata con il riassunto dell'indagine creata, in cui viene indicato (in rosso) di aggiungere gruppi e domande.

Dunque LimeSurvey usa una struttura logica che permette di strutturare tutti i questionari che vogliamo, ciascuno diviso in gruppi (pagine web) di domande. Non ci sono limiti al numero di questionari (indagini), gruppi o domande che possiamo inserire, se non quelli imposti dallo spazio web che utilizziamo (molto spesso virtualmente infinito). Inoltre non ci sono limiti al numero di utenti che possono compilare il questionario elettronico, anche contemporaneamente e da diversi dispositivi dotati di connessione internet (come PC, Tablet e Smartphone).



Figura 1.10. La struttura logica "Indagini > Gruppi > Domande" in LimeSurvey.

Fino adesso abbiamo visto come strutturare il questionario, dividendolo in gruppi composti da più domande. Adesso vedremo

cosa si intende per domande e in particolare quali tipi di domande possono essere strutturate.

Innanzitutto quando definisco una nuova domanda in un gruppo (cliccando sull'apposito tasto "+" come indicato in figura 9), dovrò compilare alcuni campi come dare un codice e inserire il testo della domanda da visualizzare. Questa operazione è piuttosto semplice, tuttavia merita un piccolo approfondimento, in quanto a volte può essere necessario avere discrezionalità di forma nel porre il quesito. Ad esempio si può voler usare un carattere particolare o inserire un'immagine, un'emoticon, un'equazione o un link a una pagina. LimeSurvey consente di poter inserire tutti questi elementi in una domanda in modo molto semplice, ovvero usando una interfaccia WYSIWYG (What You See Is What You Get, ovvero "ottieni quanto vedi"), come in figura 1.11.

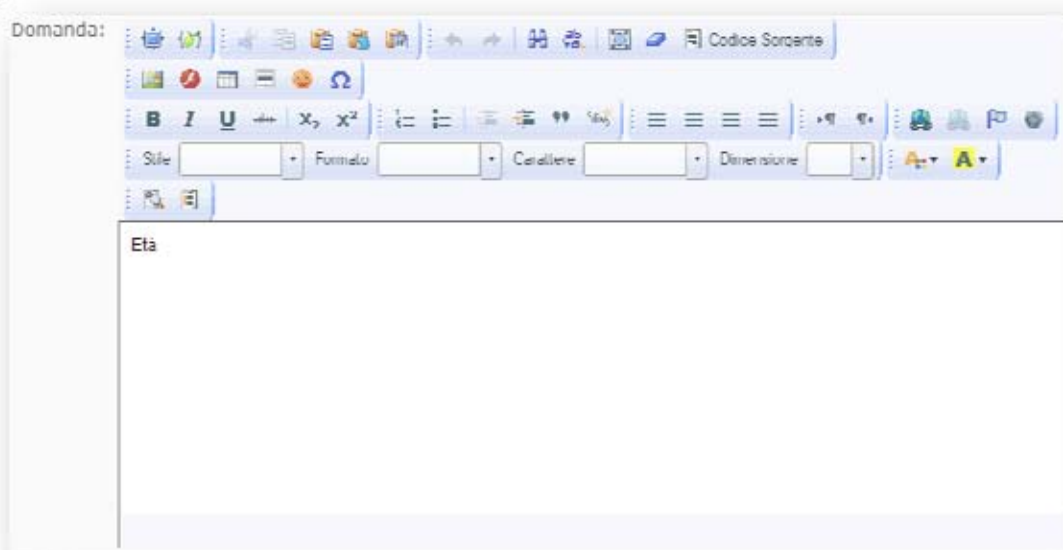


Figura 1.11. L'interfaccia WYSIWYG di LimeSurvey.

Con questa semplice interfaccia è possibile definire con dei semplici pulsanti tutte le caratteristiche della domanda, senza limiti di spazio, ma tenendo a vista l'armonia estetica e funzionale di quanto debba essere visualizzato. E' anche possibile inserire un oggetto Flash (ovvero un oggetto realizzato con Adobe Flash). Dopo aver inserito la domanda possiamo specificare la tipologia scelta, tra una vasta serie, come da Figura 1.12.

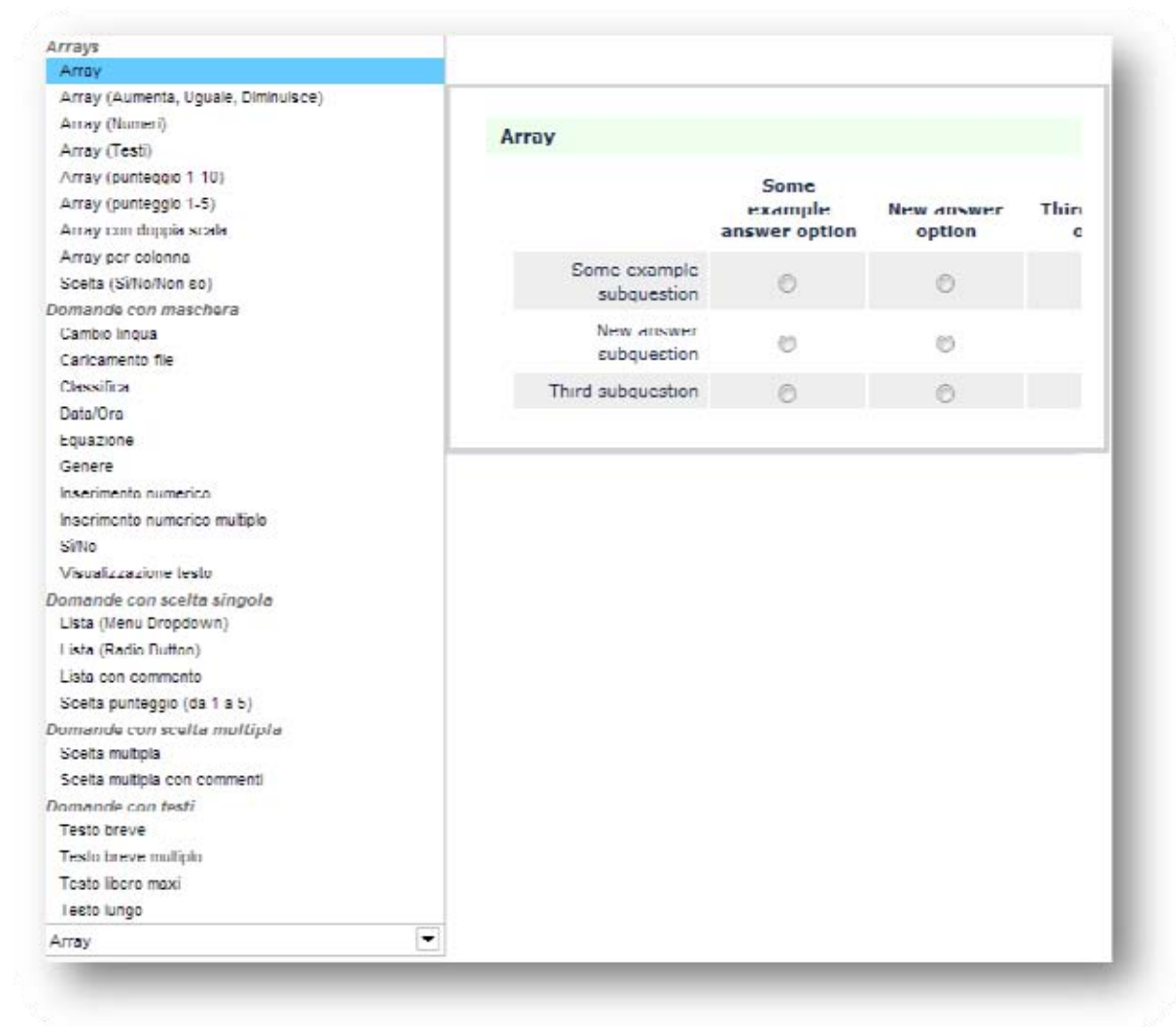


Figura 1.12. La scelta della tipologia di domanda tra le 28 disponibili attualmente.

LimeSurvey contiene per il momento 28 differenti tipologie di domanda, ma sono in continua crescita. Molte tipologie sono totalmente modulabili potendo virtualmente generare infinite strutture. Di seguito ci soffermeremo sull'Array come tipologia di esempio, per meglio comprendere come strutturare il questionario. Per ogni approfondimento andate sul manuale italiano gratuito sul sito web ufficiale (http://manual.limesurvey.org/Question_types/it) contenente tutti i dettagli e in continuo aggiornamento.

Una volta scelta e salvata la domanda di tipo "Array" ci spunterà il messaggio in Figura 1.13.

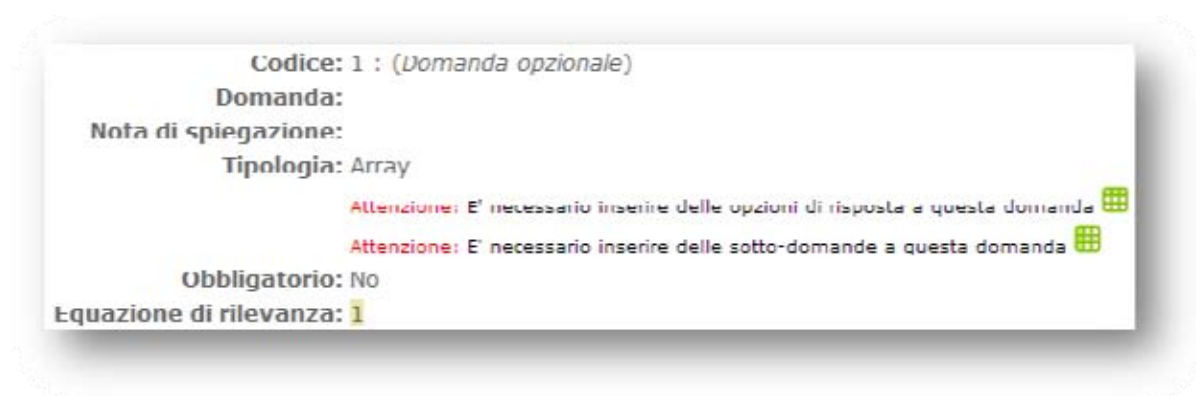



Figura 1.13. LimeSurvey ci avvisa delle opzioni di domanda da configurare. Nel caso della domanda di tipo Array ci chiede di indicare chiaramente il testo delle righe e delle colonne.

Premendo dunque sul tasto relativo alle sotto-domande andremo a configurare le varie righe della matrice di risposta, come si può vedere in Figura 1.14.



Figura 1.14. Gestione delle sotto-domande.

Premendo invece il tasto relativo alle opzioni di risposta, andremo a configurare le varie colonne della matrice di risposta, come si può vedere in Figura 1.15. Tramite il simbolo  sarà possibile aprire una finestra con un'interfaccia WYSIWYG sia per le sotto-domande, sia per le opzioni di risposta.

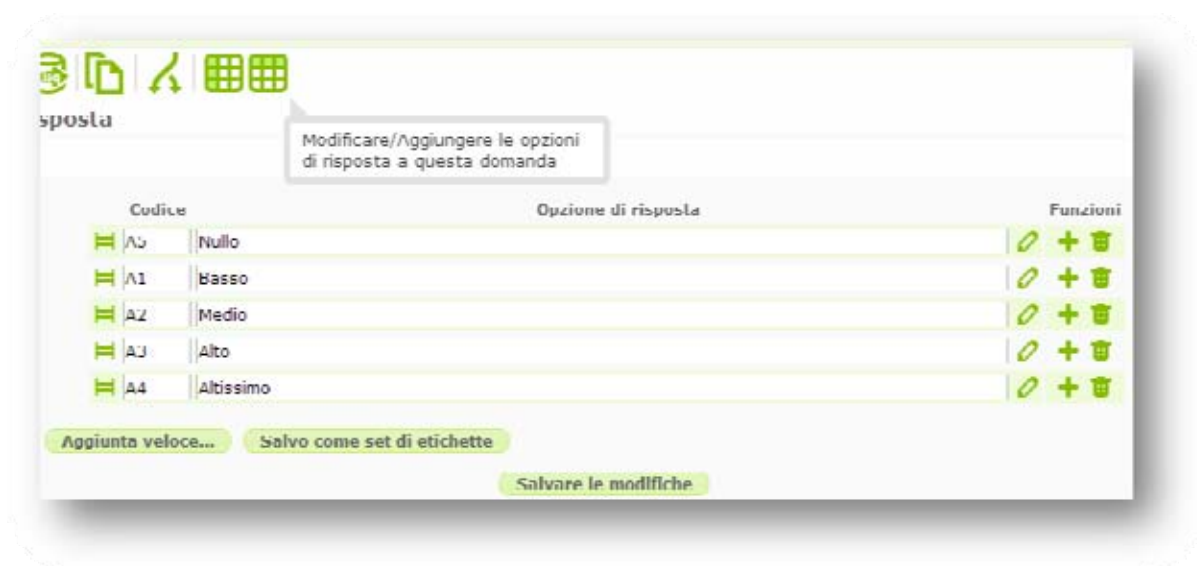


Figura 1.15. Gestione delle opzioni di risposta.


In ogni momento possiamo visualizzare un'anteprima del nostro questionario cliccando sul pulsante . In Figura 1.16, è possibile vedere l'anteprima della prima pagina del questionario creato, con il messaggio di benvenuto.



Figura 1.16. Pagina di benvenuto del questionario finale.

Il messaggio di benvenuto può essere creato facilmente con l'interfaccia WYSIWYG, ma è anche possibile scrivere un semplice codice html come di seguito specificato, per completezza.

```
<h1 style="text-align: center;">
  &nbsp;  </h1>
<h1 style="text-align: center;">
  <span style="font-size:36px;">Elementi di </span></h1>
<h1 style="text-align: center;">
  <span style="font-size:36px;">Psicometria Computazionale</span></h1>
<p>
  &nbsp;  </p>
```

```
<p style="text-align: center;">
  <span style="font-size: 20px;">Pietro Cipresso</span></p>
```

In Figura 1.17 è possibile vedere sopra la domanda sull'Età (con campo numerico) e la domanda sul genere (con scelta a "radio buttons") e sotto altre domande con l'uso di Array.

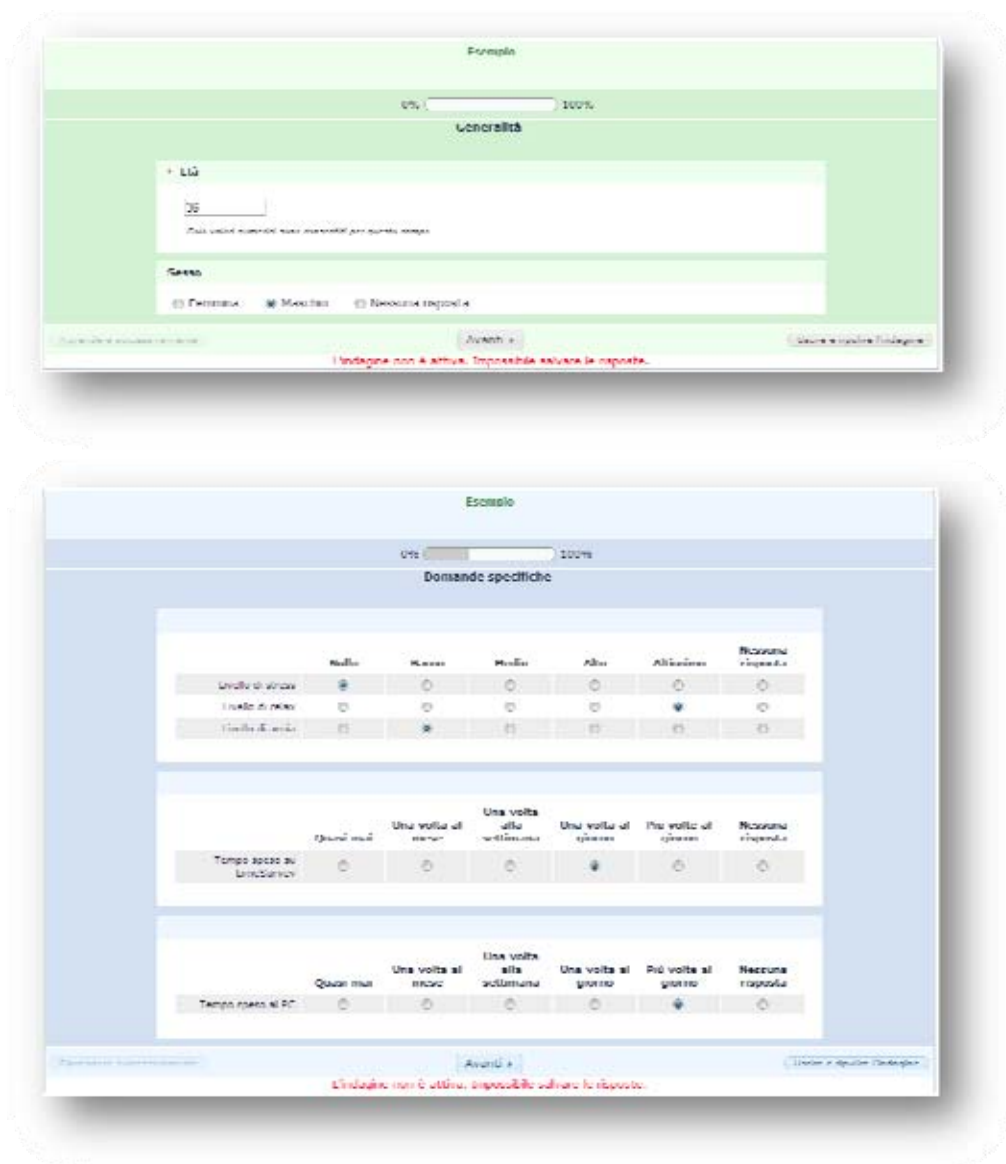



Figura 1.17. Visualizzazione del questionario con semplici domande (sopra) e con domande di tipo Array (sotto).



Una volta concluso il questionario, è consigliabile effettuare un controllo. Cliccando sul pulsante  e scegliendo l'apposita voce "File della logica di indagine" si aprirà una pagina con la seguente tabella (Tabella 1.1).

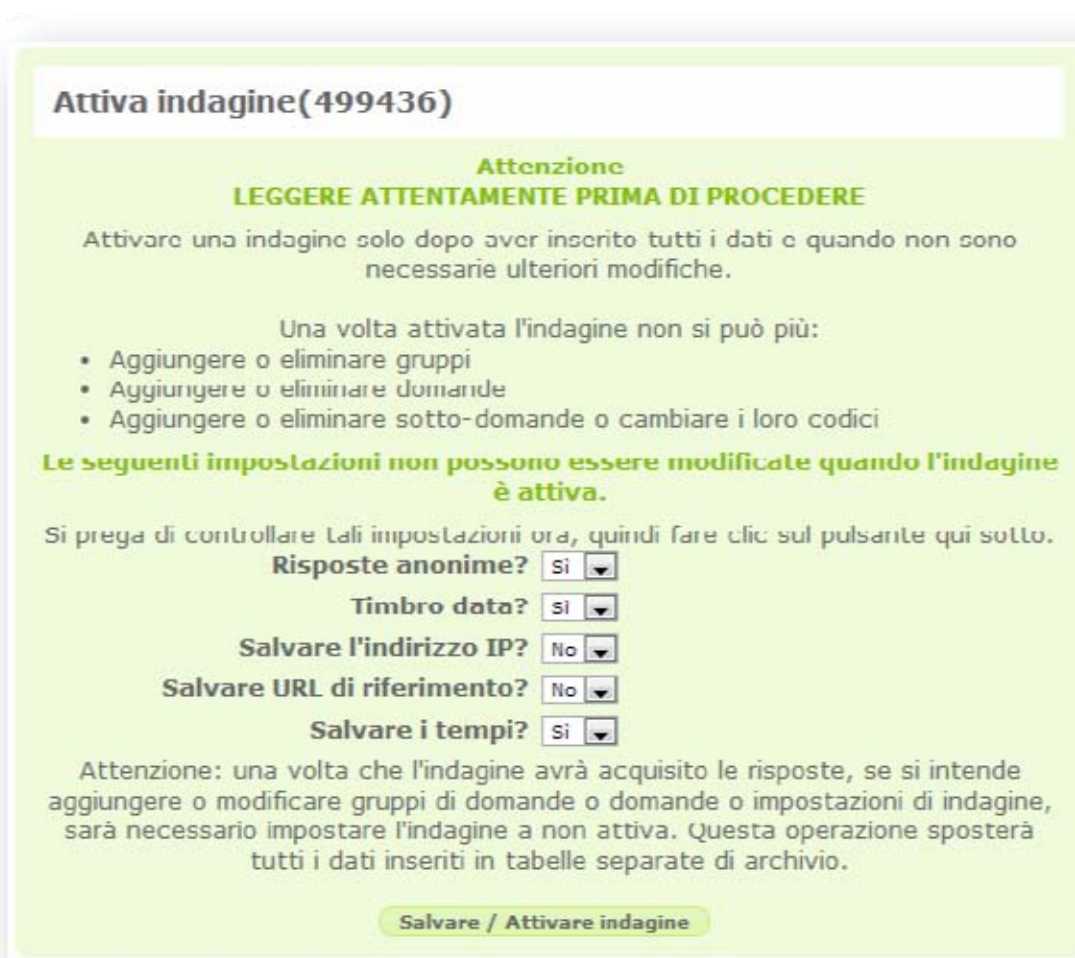
Non sono stati trovati errori di sintassi in questa indagine

File della logica di indagine #[499436]: Esempio

Messaggio di benvenuto:		Elementi di Psicometria Computazionale					
Messaggio finale:		Grazie per aver compilato questo questionario !!					
#	Nome [ID]	Relevance [Validation] (Default value)	Testo [Aiuto] (Suggerimento)				
G-0	Generalità [GID 2]	1					
Q-0	*1 [QID 2] Inserimento numerico [N]	1	Età <table border="1"> <tr> <th>Attributo della domanda</th> <th>Valore</th> </tr> <tr> <td>num_value_int_only</td> <td>1</td> </tr> </table>	Attributo della domanda	Valore	num_value_int_only	1
Attributo della domanda	Valore						
num_value_int_only	1						
Q-1	2 [QID 9] Genere [G]	1	Sesso				
G-1	Domande specifiche [GID 3]	1					
Q-2	2 [QID 3] Array [F]	1					
SQ-1	2_SQ001		Livello di stress				
SQ-2	2_SQ002		Livello di relax				
SQ-3	2_SQ003		Livello di ansia				
A[0]-1	A5	[VALUE: 0]	Nulla				
A[0]-2	A1	[VALUE: 0]	Basso				
A[0]-3	A2	[VALUE: 0]	Medio				
A[0]-4	A3	[VALUE: 0]	Alto				
A[0]-5	A4	[VALUE: 0]	Elevatissimo				
Q-3	3 [QID 4] Array [F]	1					
SQ-1	3_SQ001		Tempo speso su LimeSurvey				
A[0]-1	A1	[VALUE: 0]	Quasi mai				
A[0]-2	A5	[VALUE: 0]	Una volta al mese				
A[0]-3	A2	[VALUE: 0]	Una volta alla settimana				
A[0]-4	A3	[VALUE: 0]	Una volta al giorno				
A[0]-5	A4	[VALUE: 0]	Più volte al giorno				
Q-4	5 [QID 10] Array [F]	1					
SQ-1	5_SQ001		Tempo speso al PC				
A[0]-1	A1	[VALUE: 0]	Quasi mai				
A[0]-2	A5	[VALUE: 0]	Una volta al mese				
A[0]-3	A2	[VALUE: 0]	Una volta alla settimana				
A[0]-4	A3	[VALUE: 0]	Una volta al giorno				
A[0]-5	A4	[VALUE: 0]	Più volte al giorno				

Tabella 1.1. Logica di indagine.

Una volta effettuato il controllo cliccare sulla freccia (verde) del pulsante  per attivare l'indagine e iniziare a raccogliere dati. Nonostante il messaggio intimidatorio, in Figura 1.18, è sempre possibile disattivare l'indagine ed effettuare tutte le modifiche necessarie cliccando sul quadrato (rosso) del pulsante . Si può attivare e disattivare un questionario tutte le volte che si vuole.



Attiva indagine(499436)

Attenzione
LEGGERE ATTENTAMENTE PRIMA DI PROCEDERE

Attivare una indagine solo dopo aver inserito tutti i dati e quando non sono necessarie ulteriori modifiche.

Una volta attivata l'indagine non si può più:

- Aggiungere o eliminare gruppi
- Aggiungere o eliminare domande
- Aggiungere o eliminare sotto-domande o cambiare i loro codici

Le seguenti impostazioni non possono essere modificate quando l'indagine è attiva.

Si prega di controllare tali impostazioni ora, quindi fare clic sul pulsante qui sotto.

Risposte anonime? ▼

Timbro data? ▼

Salvare l'indirizzo IP? ▼


Salvare URL di riferimento? ▼

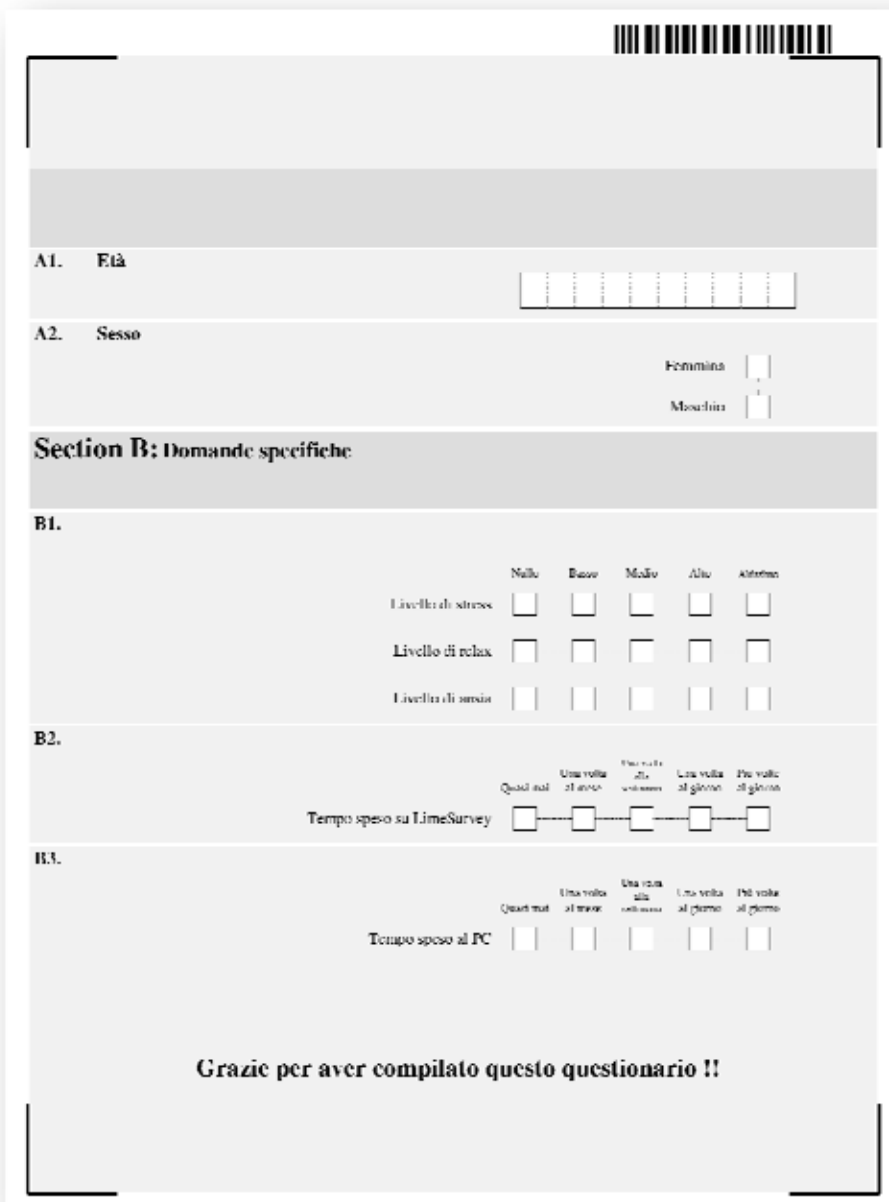
Salvare i tempi? ▼

Attenzione: una volta che l'indagine avrà acquisito le risposte, se si intende aggiungere o modificare gruppi di domande o domande o impostazioni di indagine, sarà necessario impostare l'indagine a non attiva. Questa operazione sposterà tutti i dati inseriti in tabelle separate di archivio.

Salvare / Attivare indagine

Figura 1.18. Attivazione indagine e relative impostazioni.

Cliccando sul pulsante  sarà possibile esportare il questionario creato in una vasta varietà di formati, per ogni necessità. E' anche possibile un'esportazione per la stampa (a schermo o in formato pdf), come da Figura 1.19.



The image shows a questionnaire form designed for printing. At the top right, there is a barcode. The form is divided into several sections:


- A1. Età**: A horizontal row of 10 small input boxes for entering age.
- A2. Sesso**: Two rows of checkboxes. The first row is for "Femmina" and the second for "Maschista".
- Section B: Domande specifiche**: A header for the specific questions section.
- B1.**: A section with three rows of questions. Each row has five checkboxes corresponding to the labels: "Nullo", "Basso", "Medio", "Alto", and "Altissimo". The questions are "Livello di stress", "Livello di relax", and "Livello di ansia".
- B2.**: A section with one row of questions. It has five checkboxes corresponding to the labels: "Quasi mai", "Una volta al mese", "Una volta alla settimana", "Una volta al giorno", and "Più volte al giorno". The question is "Tempo speso su LimeSurvey".
- B3.**: A section with one row of questions. It has five checkboxes corresponding to the labels: "Quasi mai", "Una volta al mese", "Una volta alla settimana", "Una volta al giorno", and "Più volte al giorno". The question is "Tempo speso al PC".

At the bottom of the form, there is a message: **Grazie per aver compilato questo questionario !!**

Figura 1.19. Versione pdf per la stampa del questionario.

Come si è potuto vedere, LimeSurvey è uno strumento completo, che permette di fare questionari professionali. Non è sicuramente questo il contesto per dare una completa trattazione di uno strumento tanto ampio e si rimanda, dunque, per ulteriori approfondimenti al manuale in italiano disponibile alla pagina http://manual.limesurvey.org/LimeSurvey_Manual/it. L'obiettivo qui era far capire che questa piattaforma gratuita può rappresentare un prezioso strumento per la misura in psicologia mediante l'uso di nuove tecnologie.

Di seguito vedremo ancora una funzionalità di LimeSurvey che ci fa anche capire che questa piattaforma può essere usata anche per ricerche ben più complesse di quanto possibile con un semplice questionario. In particolare presentiamo la possibilità di creare una domanda che si basa sulle precedenti risposte, non tanto nel calcolo (quello è semplice e può esser fatto anche a posteriori), ma nella formulazione della stessa.

Nell'interfaccia WYSIWYG della creazione della domanda (che abbiamo precedentemente visto) premendo il pulsante  (che è un lime tra parentesi!) si otterrà in modo semplice una stringa, da inserire nella propria domanda, che prenderà il dato dalla risposta a una precedente domanda fatta allo stesso soggetto (vedi Figura 1.20). Si veda ad esempio il testo che ho creato in Figura 1.21, dove {INSERTANS:499436X2X2} rappresenta la risposta alla prima domanda fatta (età) e {INSERTANS:499436X3X3SQ003} rappresenta la risposta alla domanda "livello di ansia" (vedi anche precedente Figura 1.17).

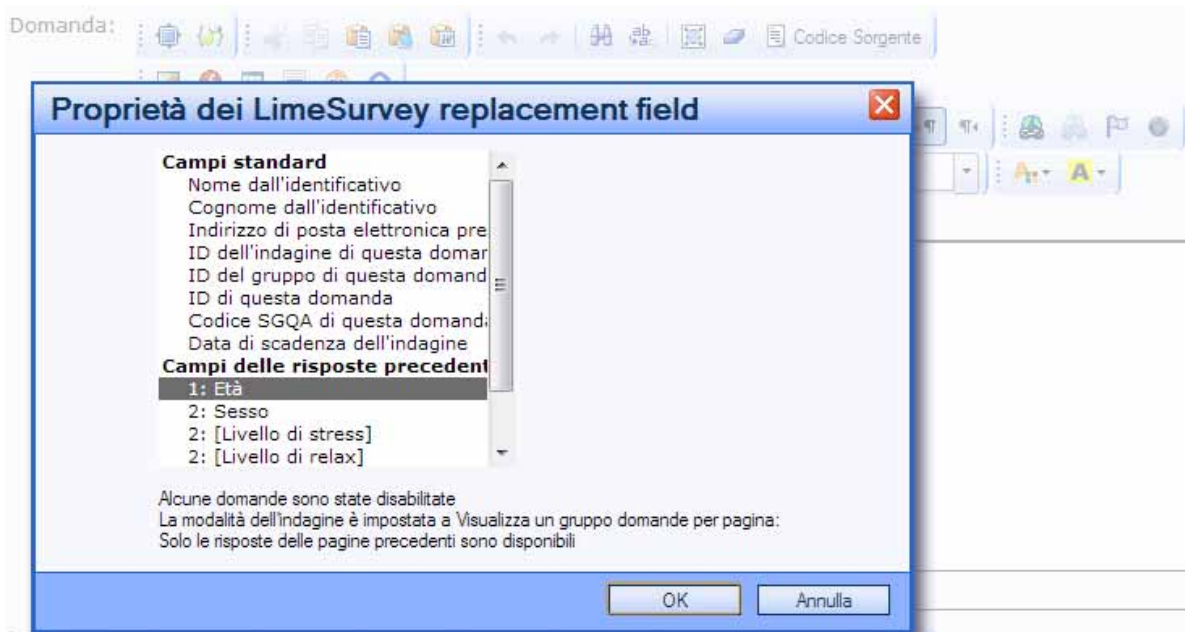


Figura 1.20. Interfaccia per la creazione della stringa basata sulla risposta a una domanda precedente.

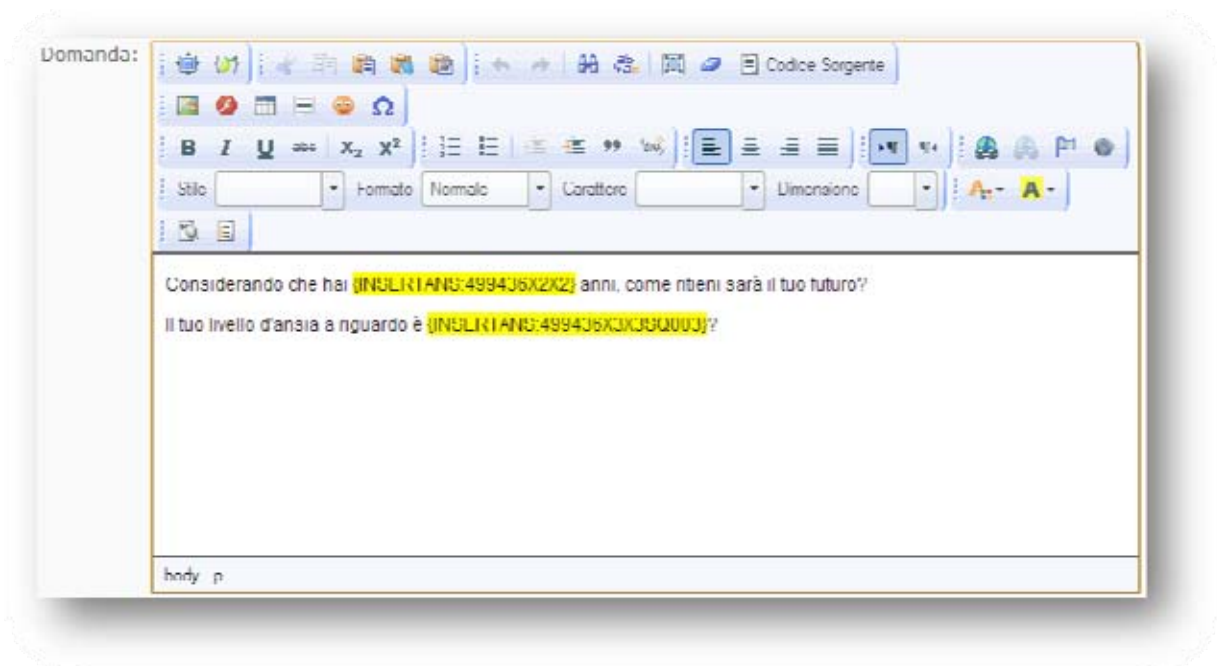




Figura 1.21. Domanda che contiene le stringhe create con il semplice tool di cui in Figura 1.20.



Figura 1.22. Domanda con inglobate le risposte a domande precedenti (età 36 e livello di ansia basso).

Una volta pubblicato il questionario online (o inviato il link via email e altri media), la piattaforma raccoglierà i dati in modo ben strutturato e pronto per successive analisi statistiche. Tuttavia, una volta conclusa la fase di raccolta dati è necessario effettuare una esportazione dal database di LimeSurvey e una importazione in un software statistico, come R (<http://www.r-project.org>), che è anch'esso gratuito e a codice aperto. Come abbiamo visto, LimeSurvey è strutturato con un vero database (MySQL) e questo garantisce solidità e capacità di gestione dei dati professionali. D'altra parte accedere a un database MySQL ed estrarre dei dati richiede saper scrivere una "query" in SQL (cosa a dire il vero assai semplice) e - soprattutto - sapere dove "mettere le mani" e come. Come vedremo in seguito, non sono operazioni impossibili e avremo modo di giocare ampiamente nel Capitolo 12, anche tramite l'uso di appositi client e software specifici, atti a semplificarci la vita.

Per adesso il modo più semplice ed efficace per esportare i risultati raccolti con LimeSurvey è cliccare sul pulsante , selezionare "risposte e statistiche" e cliccare su  per attivare il semplice modulo di esportazione di cui in Figura 1.23.



Esportare i risultati

Generale

Intervallo: Da a

Stato di completamento

Titoli

Codice domanda
 Testo abbreviato della domanda
 Testo completo delle domanda
 Converti gli spazi sui testi delle domande in trattino basso (underscore)

Risposte

Codice risposta
 Converti Y in
 Converti N in
 Risposte complete

Formato

Files CSV (Tutti i set di carattere)
 Microsoft Excel (Tutti i tipi di carattere)
 Microsoft Word (set di caratteri Latino)
 PDF

Controllo colonne


Scegliere le colonne:

- 1 : id - ID risposta
- 2 : submitdate - Data invio
- 3 : lastpage - Ultima pagina
- 4 : startlanguage - Lingua iniziale
- 5 : startdate - Data di inizio
- 6 : datestamp - Data dell'ultima azione
- 7 : 1 - ELa
- 8 : 2 - Sesso
- 9 : 2[:(:)] [Livello di stress]
- 10 : 2[SQ002] - [Livello di relax]
- 11 : 2[SQ003] - [Livello di ansia]
- 12 : 1[:(:)] [Tempo speso su LimeSurvey]
- 13 : 5[SQ001] - [Tempo speso al PC]
- 14 : 10 - Considerando che hai (INSERTANS:49...
- 15 : interviewtime - Tempo totale
- 16 : groupTime2 - Tempo per il gruppo di doma...
- 17 : 1Time - Tempo per la domanda: 1
- 18 : 2Time - Tempo per la domanda: 2
- 19 : groupTime3 - Tempo per il gruppo di doma...
- 20 : 2Time - Tempo per la domanda: 2


Figura 1.23. Modulo per l'esportazione dei risultati per le successive analisi statistiche.

Esportando il file in Excel o in csv è certamente già possibile preparare i dati per il software statistico. Ma cercheremo di fare molto più di questo e venire incontro alle esigenze di chi come me non ha pazienza di etichettare tutte le variabili e definire tutte le loro proprietà, almeno quando questo può esser fatto in modo del tutto automatico.

Istruzioni, con SPSS, per chi non ha pazienza:

1. Scaricare i dati e il file di sintassi cliccando su 
2. Aprire il file della sintassi in SPSS in modalità Unicode
3. Modificare la riga contenente `"/FILE=` e completare il nome del file con un percorso completo al file dei dati scaricato
4. Scegliere 'Esegui > All' dal menu per eseguire l'importazione

Istruzioni, con R, per chi non ha pazienza:

1. Scaricare i dati e il file della sintassi cliccando su 
2. Salvare entrambe nella directory di lavoro R (utilizza `getwd()` e `setwd()` nella finestra di comando R per prendere ed impostarli)
3. scrivere: `source("survey_499436_R_syntax_file.R", encoding = "UTF-8")` nella finestra di comando R

Il `data.frame` è nominato "data", le `variable.labels` sono attributi dei dati (`attributes(data)$variable.labels`).

Per completezza di seguito riportiamo, qui di seguito, il file di sintassi di SPSS.

```
SET UNICODE=ON.
SHOW LOCALE.
PRESERVE LOCALE.
SET LOCALE='en_UK'.
GET DATA
/TYPE=TXT
/FILE='C:\WiP\Psicometria Computazionale\Capitoli - materiale\Capitolo
1\LimeSurvey\DATA\SPSS\survey_499436_SPSS_data_file.dat'
/DELCASE=LINE
/DELIMITERS=","
/QUALIFIER="""
/ARRANGEMENT=DELIMITED
/FIRSTCASE=1
/IMPORTCASE=ALL
/VARIABLES=
V1 F7
V2 DATETIME23.2
V4 A19
V5 DATETIME23.2
V6 DATETIME23.2
V7 F13.12
V8 F1
V9 A2
V10 A2
V11 A2
V12 A2
V13 A2
V14 F1.
CACHE.
EXECUTE.
*Define Variable Properties.
VARIABLE LABELS V1 "id".
VARIABLE LABELS V2 "submitdate".
VARIABLE LABELS V4 "startlanguage".
VARIABLE LABELS V5 "startdate".
VARIABLE LABELS V6 "datestamp".
VARIABLE LABELS V7 "Età".
VARIABLE LABELS V8 "Sesso".
VARIABLE LABELS V9 "[Livello di stress]".
VARIABLE LABELS V10 "[Livello di relax]".
VARIABLE LABELS V11 "[Livello di ansia]".
VARIABLE LABELS V12 "[Tempo speso su LimeSurvey]".
```

```
VARIABLE LABELS V13 "[Tempo speso al PC]".
VARIABLE LABELS V14 "Considerando che hai {INSERTANS:499436X2X2}
anni, come ritieni sarà il tuo futuro? Il tuo livello d'ansia a riguardo
è {INSERTANS:499436X3X3SQ003}?" .
*Define Value labels.
VALUE LABELS V8
 1 "Femmina"
 2 "Maschio".
VALUE LABELS V9
"A5" "Nullo"
"A1" "Basso"
"A2" "Medio"
"A3" "Alto"
"A4" "Altissimo".
VALUE LABELS V10
"A5" "Nullo"
"A1" "Basso"
"A2" "Medio"
"A3" "Alto"
"A4" "Altissimo".
VALUE LABELS V11
"A5" "Nullo"
"A1" "Basso"
"A2" "Medio"
"A3" "Alto"
"A4" "Altissimo".
VALUE LABELS V12
"A1" "Quasi mai"
"A5" "Una volta al mese"
"A2" "Una volta alla settimana"
"A3" "Una volta al giorno"
"A4" "Più volte al giorno".
VALUE LABELS V13
"A1" "Quasi mai"
"A5" "Una volta al mese"
"A2" "Una volta alla settimana"
"A3" "Una volta al giorno"
"A4" "Più volte al giorno".
VALUE LABELS V14
 1 "Sì"
 2 "No".
VARIABLE LEVEL V7(SCALE).
```

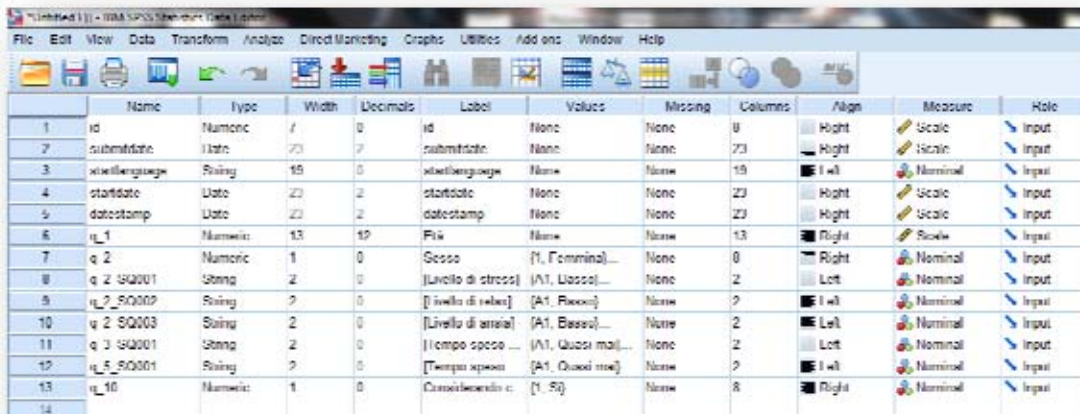


```

RENAME VARIABLE ( V1 = id ).
RENAME VARIABLE ( V2 = submitdate ).
RENAME VARIABLE ( V4 = startlanguage ).
RENAME VARIABLE ( V5 = startdate ).
RENAME VARIABLE ( V6 = datestamp ).
* Variable name was incorrect and was changed from 1 to q_1 .
RENAME VARIABLE ( V7 = q_1 ).
* Variable name was incorrect and was changed from 2 to q_2 .
RENAME VARIABLE ( V8 = q_2 ).
* Variable name was incorrect and was changed from 2_SQ001 to q_2_SQ001 .
RENAME VARIABLE ( V9 = q_2_SQ001 ).
* Variable name was incorrect and was changed from 2_SQ002 to q_2_SQ002 .
RENAME VARIABLE ( V10 = q_2_SQ002 ).
* Variable name was incorrect and was changed from 2_SQ003 to q_2_SQ003 .
RENAME VARIABLE ( V11 = q_2_SQ003 ).
* Variable name was incorrect and was changed from 3_SQ001 to q_3_SQ001 .
RENAME VARIABLE ( V12 = q_3_SQ001 ).
* Variable name was incorrect and was changed from 5_SQ001 to q_5_SQ001 .
RENAME VARIABLE ( V13 = q_5_SQ001 ).
* Variable name was incorrect and was changed from 10 to q_10 .
RENAME VARIABLE ( V14 = q_10 ).
RESTORE LOCALE.

```

Come risultato del lancio di questa sintassi di SPSS otterremo un'importazione dei risultati e delle variabili come da figura 1.24.



	Name	Type	Width	Decimals	Label	Values	Missing	Column	Align	Measure	Role
1	id	Numeric	7	0	id	None	None	7	Right	Scale	Input
2	submitdate	Date	73	2	submitdate	None	None	73	Right	Scale	Input
3	startlanguage	String	19	0	startlanguage	None	None	19	Left	Nominal	Input
4	startdate	Date	23	2	startdate	None	None	23	Right	Scale	Input
5	datestamp	Date	23	2	datestamp	None	None	23	Right	Scale	Input
6	q_1	Numeric	13	10	Sex	None	None	13	Right	Scale	Input
7	q_2	Numeric	1	0	Sexo (1, Femmina)...	None	None	0	Right	Nominal	Input
8	q_2_SQ001	String	2	0	[Livello di stress] (1, Basso)...	None	None	2	Left	Nominal	Input
9	q_2_SQ002	String	2	0	[Livello di relax] (1, Rilaxo)...	None	None	2	Left	Nominal	Input
10	q_2_SQ003	String	2	0	[Livello di ansia] (1, Basso)...	None	None	2	Left	Nominal	Input
11	q_3_SQ001	String	2	0	[Tempo spesso ...] (1, Quasi mai)...	None	None	2	Left	Nominal	Input
12	q_5_SQ001	String	2	0	[Tempo spesso ...] (1, Quasi mai)...	None	None	2	Left	Nominal	Input
13	q_10	Numeric	1	0	Caratterizzazio c:	[1, 50]	None	8	Right	Nominal	Input
14											

Figura 1.24. Vista delle variabili di SPSS già configurate.

Infine, qui di seguito riportiamo il file di sintassi di R, dalla quale si evince chiaramente tutta la procedura di importazione.

```
#$Rev: 10193 $ all.
data <- read.csv("survey_499436_R_data_file.csv", quote = "\"", na.strings=c("",
"\\"), stringsAsFactors=FALSE)

# LimeSurvey Field type: F
data[, 1] <- as.numeric(data[, 1])
attributes(data)$variable.labels[1] <- "id"
names(data)[1] <- "id"

# LimeSurvey Field type: DATETIME23.2
data[, 2] <- as.character(data[, 2])
attributes(data)$variable.labels[2] <- "submitdate"
names(data)[2] <- "submitdate"

#Field hidden

# LimeSurvey Field type: A
data[, 3] <- as.character(data[, 3])
attributes(data)$variable.labels[3] <- "startlanguage"
names(data)[3] <- "startlanguage"

# LimeSurvey Field type: DATETIME23.2
data[, 4] <- as.character(data[, 4])
attributes(data)$variable.labels[4] <- "startdate"
names(data)[4] <- "startdate"

# LimeSurvey Field type: DATETIME23.2
data[, 5] <- as.character(data[, 5])
attributes(data)$variable.labels[5] <- "datestamp"
names(data)[5] <- "datestamp"

# LimeSurvey Field type: F
data[, 6] <- as.numeric(data[, 6])
attributes(data)$variable.labels[6] <- "EtÃ "
names(data)[6] <- "q_1"

# LimeSurvey Field type: F
```

```
data[, 7] <- as.numeric(data[, 7])
attributes(data)$variable.labels[7] <- "Sesso"
data[, 7] <- factor(data[, 7], levels=c(1,2),labels=c("Femmina", "Maschio"))
names(data)[7] <- "q_2"

# LimeSurvey Field type: A
data[, 8] <- as.character(data[, 8])
attributes(data)$variable.labels[8] <- "[Livello di stress]"
data[, 8] <- factor(data[, 8], levels=c("A5","A1","A2","A3","A4"),labels=c("Nullo",
"Basso", "Medio", "Alto", "Altissimo"))
names(data)[8] <- "q_2_SQ001"

# LimeSurvey Field type: A
data[, 9] <- as.character(data[, 9])
attributes(data)$variable.labels[9] <- "[Livello di relax]"
data[, 9] <- factor(data[, 9], levels=c("A5","A1","A2","A3","A4"),labels=c("Nullo",
"Basso", "Medio", "Alto", "Altissimo"))
names(data)[9] <- "q_2_SQ002"

# LimeSurvey Field type: A
data[, 10] <- as.character(data[, 10])
attributes(data)$variable.labels[10] <- "[Livello di ansia]"
data[, 10] <- factor(data[, 10],
levels=c("A5","A1","A2","A3","A4"),labels=c("Nullo", "Basso", "Medio", "Alto",
"Altissimo"))
names(data)[10] <- "q_2_SQ003"

# LimeSurvey Field type: A
data[, 11] <- as.character(data[, 11])
attributes(data)$variable.labels[11] <- "[Tempo speso su LimeSurvey]"
data[, 11] <- factor(data[, 11],
levels=c("A1","A5","A2","A3","A4"),labels=c("Quasi mai", "Una volta al mese",
"Una volta alla settimana", "Una volta al giorno", "Più volte al giorno"))
names(data)[11] <- "q_3_SQ001"

# LimeSurvey Field type: A
data[, 12] <- as.character(data[, 12])
attributes(data)$variable.labels[12] <- "[Tempo speso al PC]"
data[, 12] <- factor(data[, 12],
levels=c("A1","A5","A2","A3","A4"),labels=c("Quasi mai", "Una volta al mese",
"Una volta alla settimana", "Una volta al giorno", "Più volte al giorno"))
names(data)[12] <- "q_5_SQ001"
```

```
# LimeSurvey Field type: F
data[, 13] <- as.numeric(data[, 13])
attributes(data)$variable.labels[13] <- "Considerando che
hai {INSERTANS:499436X2X2} anni, come ritieni sarÃ il tuo futuro? Il tuo
livello d'ansia a riguardo Ã {INSERTANS:499436X3X3SQ003}?"
data[, 13] <- factor(data[, 13], levels=c(1,2),labels=c("SÃ-", "No"))
names(data)[13] <- "q_10"

# Variable name was incorrect and was changed from 1 to q_1 .
# Variable name was incorrect and was changed from 2 to q_2 .
# Variable name was incorrect and was changed from 2_SQ001 to q_2_SQ001 .
# Variable name was incorrect and was changed from 2_SQ002 to q_2_SQ002 .
# Variable name was incorrect and was changed from 2_SQ003 to q_2_SQ003 .
# Variable name was incorrect and was changed from 3_SQ001 to q_3_SQ001 .
# Variable name was incorrect and was changed from 5_SQ001 to q_5_SQ001 .
# Variable name was incorrect and was changed from 10 to q_10 .
```


Dati psicofisiologici

Nel precedente capitolo abbiamo visto come i questionari possano rappresentare un prezioso strumento di misura in psicologia e come con l'avvento delle nuove tecnologie questi strumenti si siano ammodernati dando sempre maggiori possibilità di utilizzo e permettendo sempre maggiori manipolazioni sperimentali. Dunque abbiamo verificato che la psicometria computazionale offre nuove occasioni di approfondimento per i classici strumenti, cioè quelli esistenti da ben prima delle nuove tecnologie.

D'altra parte, la sempre maggiore capacità elaborativa dei calcolatori e l'abbattimento dei costi di produzione e vendita, ha consentito sempre di più lo sviluppo di tecnologie alla portata di tutti. Rientrano in questa categoria apparecchiature mediche usate da oltre un secolo (come l'elettroencefalografo o l'elettrocardiografo), divenute oggi strumenti della psicofisiologia. Il mondo della psicologia si nutre ormai da diversi decenni dei dati tratti da tali strumenti e si è creato un vasto

interesse in tutti i settori della psicologia. Questo interesse è poi cresciuto notevolmente nell'ultima decade proprio grazie alla maggiore disponibilità della tecnologia a basso costo.

Tuttavia lo psicologo moderno che voglia trarre beneficio dall'uso di queste nuove tecnologie psicofisiologiche deve affrontare da un lato il problema della conoscenza specifica delle misure che possono essere rilevate tramite il loro utilizzo, dall'altro deve interfacciarsi con nuove strutture di dati e modi completamente differenti di acquisizione di tali dati, rispetto ai dati sui questionari che prima abbiamo visto.

In questo capitolo daremo una breve introduzione agli strumenti psicofisiologici e ci concentreremo su come raccogliere e preparare all'analisi i dati che ne derivano. Nella seconda parte del volume impareremo poi la gestione avanzata e l'analisi di questi dati, per loro natura e specie assai complessi.

2.1 Modalità di raccolta dei dati psicofisiologici

Uno psicologo che oggi voglia avvalersi di misure psicofisiologiche si trova di fronte a uno scenario impressionante in termini di quantità e qualità di strumentazioni disponibili. Tuttavia si dovrà anche interfacciare con delle scelte difficili che comprometteranno necessariamente la qualità e la quantità delle misurazioni che potrà effettuare. Infatti, ipotizzando che non si abbia a disposizione un capitale illimitato, nella psicofisiologia

non si può pensare di avere tutto, ma bisogna concentrarsi sulle misure che si vogliono raccogliere e scegliere di conseguenza lo strumento più adatto. In questo volume non c'è spazio o volontà di voler approfondire questo argomento, per il quale si rimanda a volumi più specifici da selezionare anch'essi sulla base delle proprie necessità. Tuttavia tratteremo i principali strumenti (ovviamente non tutti) che si usano in psicofisiologia e focalizzeremo l'argomentazione sui dati che se ne possono estrarre, di modo che, a prescindere dallo strumento che si usa, si impari leggendo questo volume a trattare e gestire tali dati, come già detto, assai complessi.

Una prima confusione in merito agli strumenti da utilizzare è relativa alla presenza di biosensori che a parità di "nome" hanno costi che variano dai cento agli oltre centomila euro. Va dunque fatto un breve approfondimento in merito. Se è vero che attrezzature molto costose e spesso anche certificate per uso clinico, costituiscono un gold standard, è anche vero che strumenti a bassissimo costo possono, in certi casi, costituire un prezioso strumento. Un caso tipico è la misurazione (come resistenza elettrica cutanea o conduttanza) dell'attività elettrodermica (EDA, ElectroDermal Activity), per mezzo di un biosensore il cui costo parte da poche centinaia di euro. Nonostante il basso costo si ottiene un sensore adatto a effettuare importanti misurazioni: infatti, le ghiandole sudoripare sono controllate dal sistema nervoso simpatico, dunque la conduttanza cutanea è usata come indicazione di un attivazione (arousal) psicofisiologica diretta (nel

senso che non c'è un intervento del sistema nervoso parasimpatico). La conduttanza cutanea rilevata 32 volte al secondo è sufficiente per una sua analisi e dunque anche con un sensore a basso costo è possibile fare delle rilevazioni adatte a molti studi (tra cui quelli relativi al marcatore somatico che usa proprio questo strumento!).

Per chi leggesse per la prima volta "32 volte al secondo" si comincerebbe ad aprire un universo fatto di profondi dubbi, meglio dunque dissiparli sin dall'inizio. Una caratteristica comune di molti dati psicofisiologici è la rilevazione di dati più volte al secondo. Normalmente si lavora con almeno 256 rilevazioni al secondo, generando così una considerevole quantità di dati a fronte di un'unica rilevazione psicofisiologica. Proprio per questo motivo la psicometria computazionale è di rilevanza primaria e di alta applicazione nella psicofisiologia. Per fare un esempio pratico e cominciare a capire la complessità dei dati di cui stiamo parlando, si pensi che una sperimentazione che richieda la rilevazione di anche un solo parametro psicofisiologico per dieci minuti a 256 dati al secondo implica una mole di dati pari a 153.600 rilevazioni (= 256 dati X 60 secondi X 10 minuti). Considerando che spesso le sperimentazioni psicofisiologiche si effettuano su più soggetti, con più sensori e il calcolo di una moltitudine di indici, si capisce ben presto che una singola sperimentazione richiede l'elaborazione di diversi miliardi di dati, da cui deriva un'ovvia complessità anche solo per quel che concerne la loro gestione.

I dati al secondo che vengono rilevati, si misurano in hertz (Hz) e sono per la psicomatria computazionale tra le prime unità di misura di cui tener conto: da questa quantità derivano successive frammentazioni. Gli Hz si calcolano secondo la seguente equazione:

$$Hz = S^{-1} = \frac{1}{S}$$

Dove S rappresenta i secondi.

Più in generale, la frequenza di campionamento è la misura espressa in hertz del numero di volte al secondo in cui un segnale analogico viene misurato e memorizzato in forma digitale (da "digit" ovvero numero).

Gli Hz sono l'unità di misura ufficiale del Sistema Internazionale della frequenza e sono molto più semplici di quanto non sembrino. In pratica, dire che la rilevazione psicofisiologica viene effettuata a 256 Hz significa semplicemente che il biosensore rileva 256 dati al secondo: è solo più immediato e formalmente più elegante.

Detto ciò, sul campionamento c'è moltissimo da dire ed è pertanto il caso di rimandare a letture più specifiche. Tuttavia, alcuni argomenti trattati nel volume si incrociano con tali temi, come ad esempio i discorsi sulla complessità che affronteremo al capitolo 16 in relazione al Teorema del campionamento di Nyquist-Shannon. Di frequenza invece torneremo a parlare nel capitolo 15, per dare qualche strumento analitico per l'analisi dei segnali.

Un altro aspetto molto importante da capire è che la rilevazione dei dati psicofisiologici è effettuata tramite una rilevazione continua nel tempo (anche se campionata e dunque registrata in modo discreto) e proprio per questo nella psicofisiologia si parla di segnale psicofisiologico, distinguendolo dagli indici che da questo segnale possono essere calcolati. A partire da un singolo segnale, possono essere calcolati anche centinaia di indici.

La prima estrazione necessaria a partire dal biosensore è dunque il segnale psicofisiologico: in Figura 2.1 un esempio.

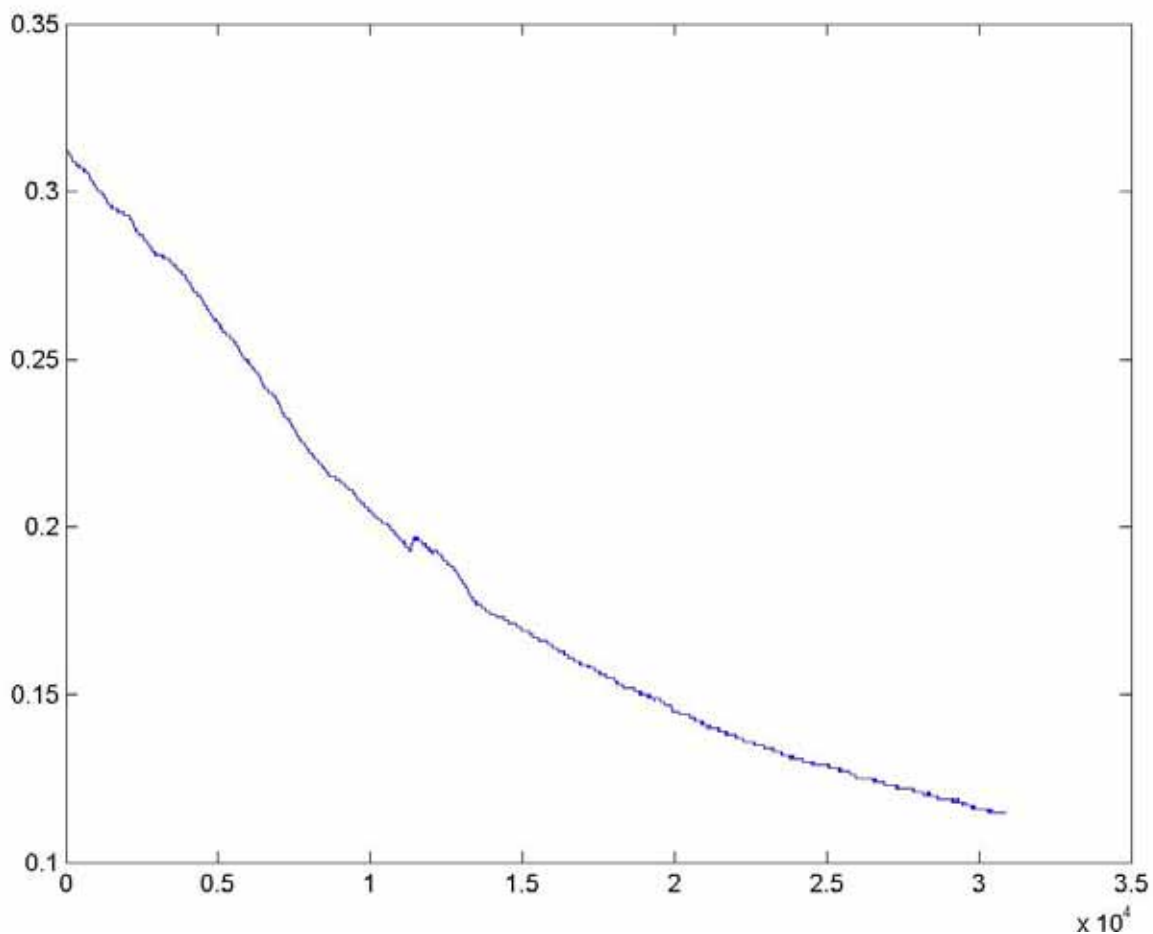


Figura 2.1. Un esempio di segnale psicofisiologico.

Il segnale viene graficato come linea continua, ma bisogna sempre tener conto che si tratta di un insieme di punti discreti (staccati l'uno dall'altro) che siccome vengono rilevati a brevissima distanza di tempo (dicevamo prima ogni duecentocinquantesimo di secondo, se abbiamo campionato a 256 Hz), non hanno modo di variare considerevolmente e appaiono dunque vicini (Figura 2.1). Ovviamente questa affermazione vale in psicofisiologia e comunque non sempre, ma è utile per dare un'idea del processo di misurazione.

E' bene sapere che il segnale psicofisiologico non è sempre perfetto e può presentare problemi anche considerevoli dovuti ad esempio a interferenze, disturbi elettrici o molto altro. Ad esempio i cellulari accessi nell'arco di un metro circa producono delle interferenze elettriche alterando il segnale.

Un segnale alterato è spesso visibile dalle peculiarità della forma grafica (Figura 2.2) e proprio per questo è sempre indispensabile l'ispezione visiva di tutti i segnali rilevati.

Anche quando dal segnale intendiamo estrarre solo ed esclusivamente la media, i segnali alterati vanno esclusi in quanto la situazione di interferenza potrebbe aver influito sulle soglie del segnale e non soltanto sulla sua forma generale.

Da notare in Figura 2.1 e 2.2 che l'asse delle ascisse (orizzontale), che rappresenta il tempo, è su scala da 0 a 3.5×10^4 (= 35,000) ovvero, dato un campionamento a 256Hz, poco più di due minuti e 15 secondi ($35,000 / 256 \sim 137$ secondi). Con questa densità di

dati, se il grafico presenta linee orizzontali c'è qualcosa di sbagliato.

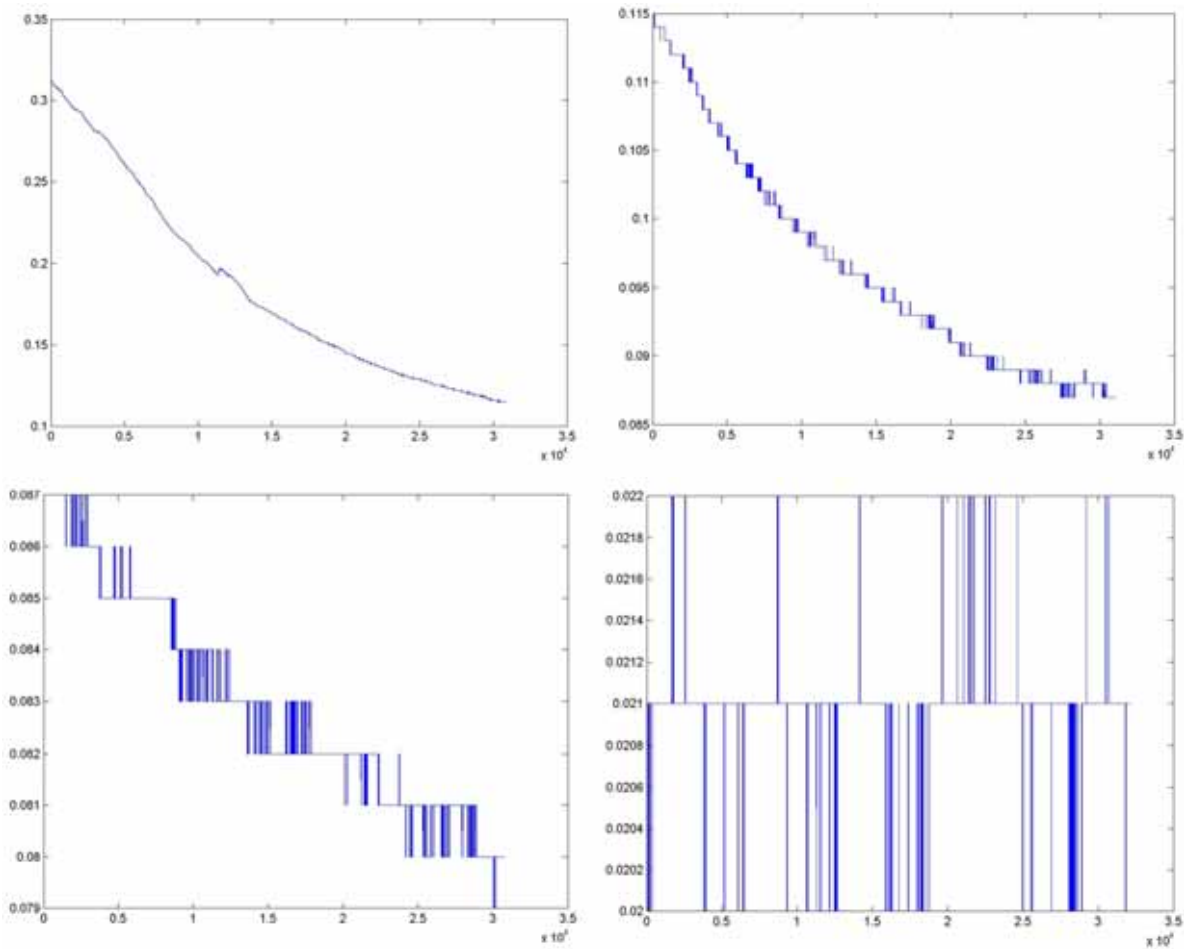


Figura 2.2. Differenza tra un segnale normale (il primo in alto a sinistra) e segnali alterati (gli altri).

I dati psicofisiologici vengono rilevati tramite biosensori specifici che normalmente rilevano un segnale (elettrico o di altro tipo) tramite un contatto in prestabilite aree specifiche del corpo. Pur non essendo il focus specifico di questo capitolo, vedremo di

seguito la rilevazione dei segnali psicofisiologici a partire dalla specifica di posizionamento di alcuni biosensori. Questa argomentazione tornerà utile anche più avanti, nel capitolo 14 del volume, quando parleremo della computazione dei dati in realtime.

Conduttanza cutanea: misurazione della conduttanza elettrica della pelle (EDA, come visto in precedenza), detta anche risposta galvanica cutanea o risposta elettrodermale. Il sensore è composto da due elettrodi posizionabili su indice e anulare della mano, o sul palmo della stessa, tramite *patch* adesive sulle quali vengono posizionati gli elettrodi. Essi creano un circuito elettrico contrapposto al soggetto che si pone come resistore. Il segnale GSR rappresenta variazioni del Sistema Nervoso Simpatico ed è espresso in micro-Siemens.

Temperatura periferica: un termistore misura i cambiamenti dello stato termico di sedi periferiche. La temperatura rilevata dipende dalla quantità di sangue presente in corrispondenza del sensore e dallo stato di attivazione simpatica. Lo stato termico è misurato in gradi Celsius o Fahrenheit.

Respirazione: i sensori sono sensibili ai movimenti di torace e addome; attraverso l'applicazione di due fasce dotate di un elastico, il sensore ne misura la tensione durante l'atto respiratorio.

Elettrocardiografia: registrazione dell'attività elettrica del cuore generata dai muscoli cardiaci durante le contrazioni.

Mentre in clinica si usano 12 derivazioni, che permettono un'analisi completa dell'attività cardiaca, in psicofisiologia si usano più spesso tre elettrodi posizionati secondo l'antico principio fisiologico del Triangolo di Einthoven (Mann, 1920).

Il segnale (ECG, elettrocardiogramma) è misurato in microvolt (μV). In ambito clinico si utilizzano molte derivazioni in quanto, come è possibile vedere in Figura 2.3, il segnale prodotto è analizzabile sotto parecchi aspetti. In psicofisiologia, ciò che più spesso interessa è la distanza tra picchi R (distanza R-R o RR).

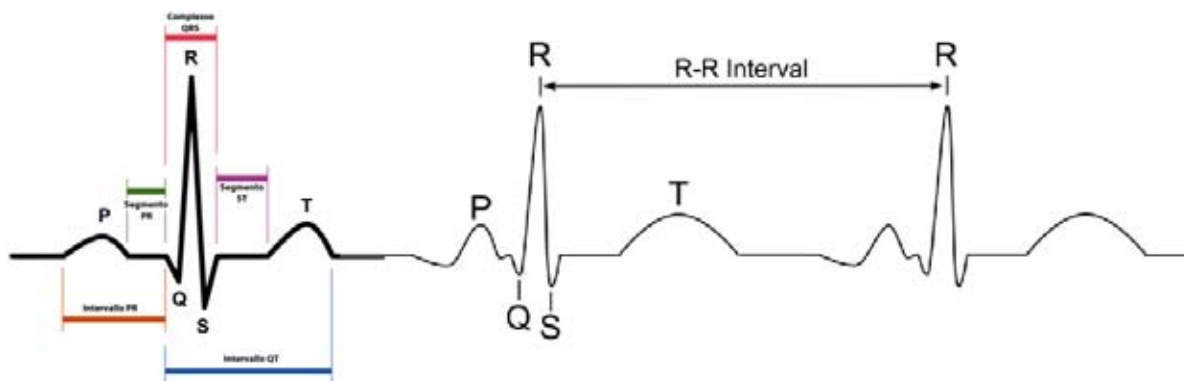


Figura 2.3. Tracciato ECG e sue componenti, con indicazione dell'intervallo RR.

Blood Volume Pulse (BVP): un sensore, posto sul polpastrello di una mano, cattura il riflesso dell'infrarosso; il sangue riflette la luce rossa assorbendo la luce degli altri colori; la quantità di luce riflessa è proporzionale al volume di sangue presente, il quale

aumenta a ogni battito cardiaco. In psicofisiologia è spesso usato l'Inter-Beat Interval (IBI), ovvero la successione degli intervalli fra una pulsazione e l'altra, come alternativa all'intervallo RR dell'EKG. Seppur i due segnali (IBI dal BVP e RR dall'ECG) non si equivalgono sono certamente assai assimilabili e in alcune sperimentazione ove è necessario ridurre l'invasività dell'ECG a favore di un più ergonomico BVP, questa soluzione può avere senso.

Elettrooculografia: L'occhio è sede di un potenziale elettrico, che può anche essere rilevato con l'occhio chiuso o in totale oscurità. L'elettrooculografia si usa infatti per gli studi sul sonno, ma è anche assai utile per rimuovere gli artefatti sull'EEG, derivanti da movimenti oculari o da ammiccamento (blinking).

Elettromiografia di superficie: analisi dell'attività delle fibre muscolari. Il sensore permette di monitorare i segnali elettrici generati dai muscoli durante le contrazioni. L'intensità del segnale elettrico è proporzionale alla forza delle contrazioni e la sua unità di misura è il microvolt (μV). In psicofisiologia è spesso usata in corrispondenza di specifici muscoli facciali per specifiche analisi sulle emozioni.

A livello clinico, l'elettromiografia viene spesso eseguita tramite l'utilizzo di aghi inseriti nei muscoli che si desidera indagare. In psicofisiologia si preferisce invece un'analisi cutanea di superficie del segnale EMG non invasiva, quindi senza l'inserimento di aghi.

Elettroencefalografia: analisi dell'attività dell'encefalo. L'elettroencefalografia viene misurata attraverso il posizionamento di elettrodi; l'unità di misura del segnale è il microvolt (μV). Il sensore rileva un segnale grezzo dal quale è possibile anche estrarre le frequenze delle onde Beta, Alpha, Theta, Delta. In base al tipo di sperimentazione il numero di elettrodi può arrivare anche a 256 o oltre, seguendo schemi predefiniti come quello in Figura 2.4.

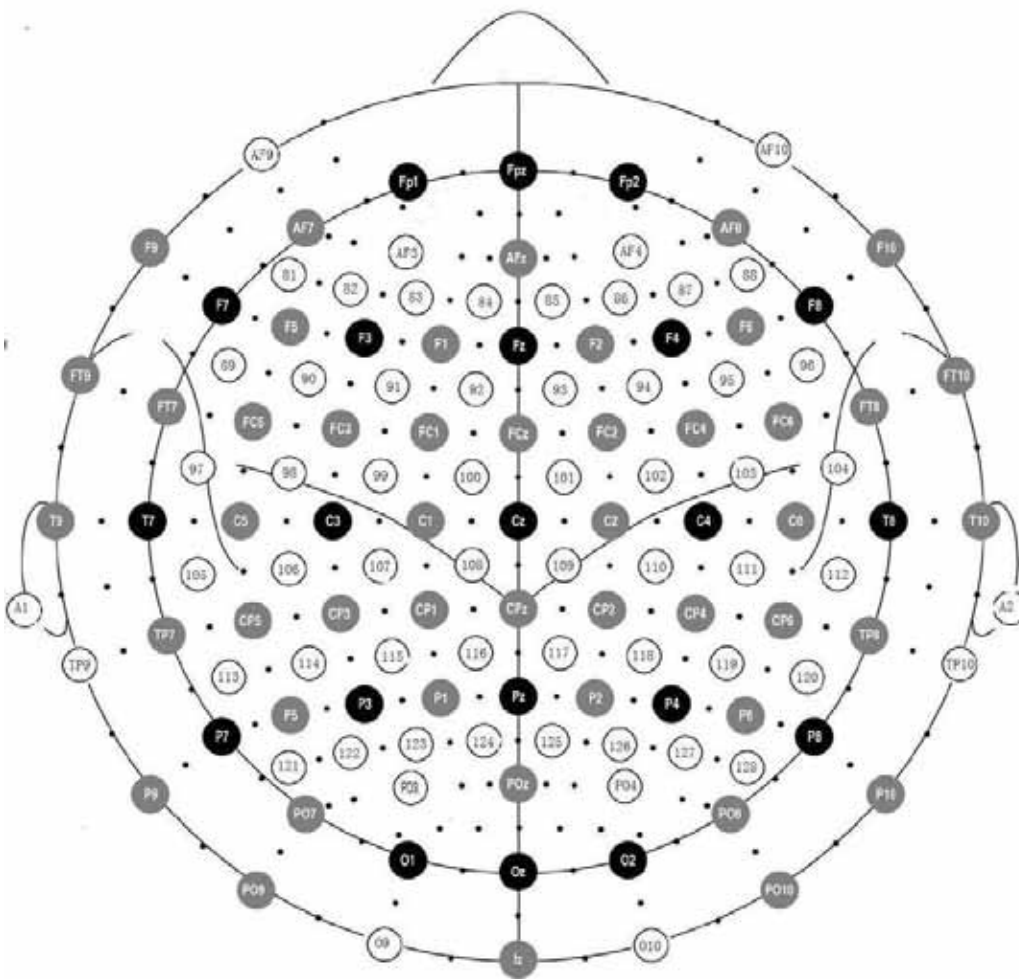


Figura 2.4. Schema di posizionamento degli elettrodi per l'elettroencefalografia.

2.2 Registrazione e gestione del segnale psicofisiologico

La maggior parte dei sensori per la rilevazione di segnali psicofisiologici viene venduta con un software specifico per la registrazione del segnale grezzo e il calcolo di alcuni indici. Ovviamente non è qui possibile spiegare anche solo alcuni di tali software. Inoltre, l'uso di questi software spesso semplifica l'analisi a rischio di commettere gravi errori. Si pensi ad esempio a un software che consenta la registrazione, tramite l'apposito biosensore, di segnali di conduttanza cutanea e l'estrazione delle medie da questi segnali. Ove il software consentisse questa estrazione sarebbe certamente una semplificazione, in quanto ci evita di dover esportare tutti i singoli segnali per poi procedere al calcolo della media. Tuttavia pensate se tra i segnali raccolti alcuni sono come quelli prima presentati in Figura 2.2. In tal caso andremmo a calcolare un indice su un segnale che contiene delle chiare incongruenze e che sarebbe da escludere.

Per risolvere tale problema la soluzione è affidarsi alla visualizzazione dei segnali. Molti software, fortunatamente, non si limitano ad estrarre gli indici ma consentono anche una visualizzazione, dei segnali acquisiti. Un'altro caso da esaminare è la presenza dei cosiddetti artefatti. Ve ne sono di molti tipi e non tutti i software, purtroppo, prevedono la rimozione di questi artefatti, con conseguenze disastrose in merito alle conclusioni che si traggono da una sperimentazione. Per avere un'idea di come si presenta un artefatto si osservi la Figura 2.5. Se il programma

permette la rimozione di segmenti specifici (contenenti artefatti), bisognerà estrarre tutto il segnale e procedere con una gestione avanzata dei dati, come vedremo nella seconda parte del volume.

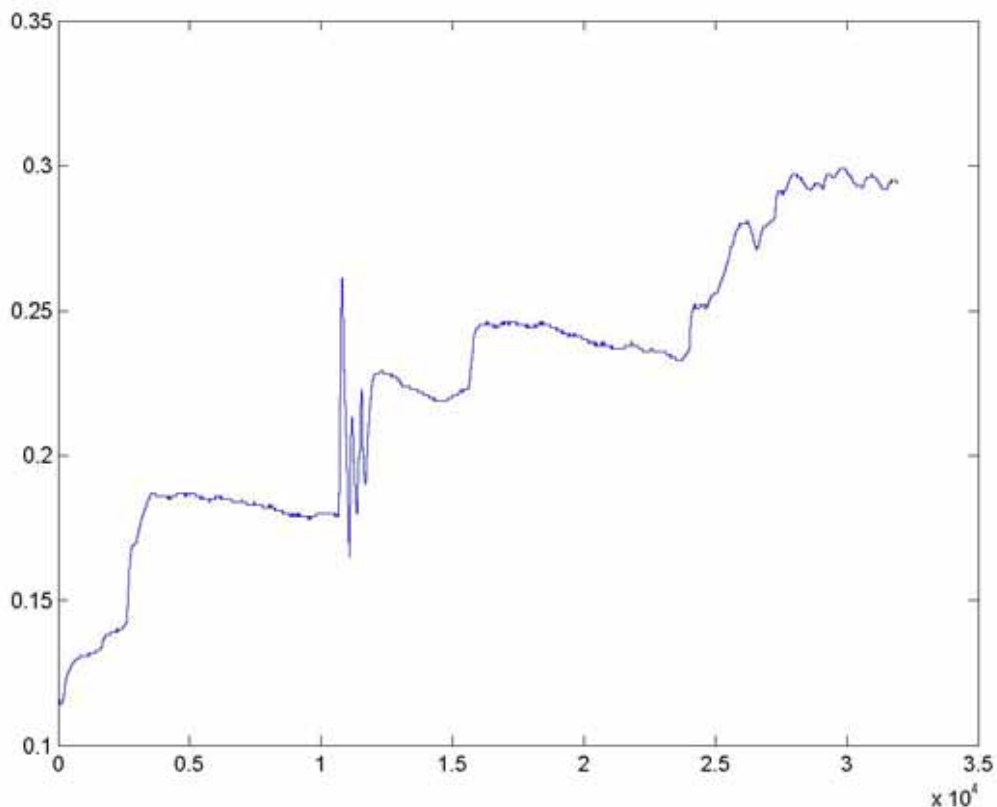


Figura 2.5. Un segnale contenente un artefatto generato toccando l'elettrodo durante la rilevazione.

I software in dotazione con i biosensori possono dunque essere limitati e di questo bisognerà sempre tenerne conto.

Tutti i software contengono comunque la possibilità di estrarre i dati grezzi (i segnali campionati) e su questi vale la pena lavorare.

Ci sono anche dei biosensori che non registrano i dati mediante l'uso di un computer, ma campionano i segnali e li memorizzano in una memoria locale (SD card, Flash card, o similari) o trasmettono questi dati in bluetooth a un dispositivo (fisso o mobile). Questa modalità di registrazione va comunque benissimo, il punto è raccogliere i dati grezzi e averne disponibilità.

Il vantaggio dei sistemi che registrano in locale è, molto spesso, la portabilità elevata; il vantaggio dei sistemi che registrano su computer è il poter visualizzare i dati in tempo reale per una prima verifica. Ad esempio quando si raccolgono i dati elettroencefalografici si guarda l'elettrocardiogramma facendo ammiccare il partecipante più volte, digrignare i denti e chiudere gli occhi, proprio per vedere le variazioni conseguenti sull'elettroencefalogramma e verificare che il segnale è ottimale.

Sul calcolo automatico del software degli indici a partire dai dati grezzi, come già detto prima con l'esempio della media, ci sono molte criticità e in generale più complesso è il segnale e l'indice da estrarre e più improbabile sarà la qualità e la veridicità di un calcolo automatico.

Su questo torneremo più volte nel corso del volume, in particolare sia nella seconda parte, dove impareremo a impostare i dati nel modo corretto, sia al capitolo 14 dove l'analisi del segnale in tempo reale impone un calcolo automatico da parte di un software specifico e vedremo quando e quanto questo processo può funzionare efficacemente.

2.3 Estrazione del segnale psicofisiologico

Il formato con cui si estraggono i dati è importante, ma generalmente non è critico in quanto esistono degli standard specifici che rendono questi dati facilmente leggibili una volta esportati. Ove possibile, soprattutto all'inizio e fino a quando non si diventa maggiormente esperti, meglio affidarsi a un semplice formato ASCII o esportare in formato CSV (comma separated values, ovvero valori separati da virgola). Questi sono semplici sia nella struttura, sia nella successiva loro importazione in altri programmi da usare per la gestione avanzata dei dati.

Molto più critico nel processo di estrazione del segnale è invece il cosa e come esportare i segnali raccolti. Può sembrare banale, ma questo processo è molto delicato e soggetto a errori importanti.

Ipotizziamo ad esempio di voler fare una semplice sperimentazione con due sessioni: una di relax (presentazione di diapositive di paesaggi rilassanti) e una di stress cognitivo (compito aritmetico complesso). Entrambe le sessioni durano 300 secondi e vengono bilanciate tra i partecipanti alla sperimentazione, ovvero metà farà prima relax e poi stress cognitivo e l'altra metà in ordine inverso. Decidiamo anche di fare una sessione baseline di 200 secondi a inizio esperimento. Le sessioni di baseline, relax e stress si chiameranno epoche sperimentali o semplicemente epoche e sono i periodi (spazi di tempo) oggetto di analisi e confronto statistico per uno sperimentatore. Lo schema del disegno sperimentale è rappresentato in Figura 2.6. Fare uno schema della

sperimentazione è consigliato, proprio perché aiuta a visualizzare le epoche e il loro bilanciamento (o randomizzazione).

Al complessificarsi di condizioni e disegno sperimentale, uno schema aiuta a non perdere di vista le unità di analisi statistica che poi saranno oggetto dell'indagine: è dunque altamente consigliato.



Figura 2.6. Esempio di disegno sperimentale con definizione delle epoche di Baseline, Relax e Stress.

A questo punto, il cosa e come estrarre dipende innanzitutto da come si sono effettuate le registrazioni dei segnali. Ci sono diverse possibilità in merito.

Una prima possibilità è far partire e stoppare la registrazione rispettivamente all'inizio e alla fine di ciascuna epoca, nominando la sessione di volta in volta.

Una seconda possibilità è quella di registrare un'unica sessione sperimentale per ciascun soggetto e avere dei markers che contraddistinguono la fine e l'inizio delle sessioni, così come permettono alcuni software di registrazione dei segnali (ad

esempio premendo un numero o una lettera o la barra spaziatrice), con il rischio che lo sperimentatore sbagli o si dimentichi e conseguente perdita di tempo e rischio di inficiare la sessione sperimentale, introducendo anche eterogeneità nelle condizioni sperimentali.

La terza possibilità è automatizzare la presentazione degli stimoli con un programma specifico (possono essere anche delle slide se non si hanno esigenze di precisione al millisecondo) per poi tagliare i segnali di conseguenza (sequenza: 200, 300, 300) stando attenti a tener traccia di quali partecipanti hanno fatto prima relax e poi stress cognitivo e viceversa. Quest'ultima soluzione ha il vantaggio di rendere più fluida la sperimentazione e ridurre al minimo gli errori e l'interazione tra partecipante e sperimentatore, che potrebbe influire alterando i segnali psicofisiologici.

2.4 Come assegnare i nomi ai segnali estratti

Parlare di assegnamento dei nomi può sembrare assai semplice e scontato. Tuttavia, come vedremo anche in seguito, la corretta assegnazione dei nomi sta alla base di tutta la psicomatria computazionale. La logica dell'automatizzare e trattare processi di dati complessi funziona se e solo se le unità che trattiamo sono state nominate correttamente con una logica sequenziale e in modo da ridurre al minimo l'errore umano, sempre comunque presente nei processi di esportazione di dati.

Il problema non è solamente dare il corretto nome a partecipanti, epoche e segnali, ma codificare il tutto correttamente, cosicché qualunque successivo intervento su tutti i dati può essere fatto anche da un algoritmo, in modo del tutto logico e automatico.

Il modo per fare questo nel modo corretto è codificare il processo di assegnazione dei nomi dei file.

Tornando al precedente esempio delle epoche sperimentali, ipotizziamo di dover estrarre i segnali grezzi di EEG, EMG ed ECG di 30 partecipanti per le tre epoche. Dal momento che ogni segnale richiederà una diversa analisi posso iniziare a organizzare tre differenti cartelle, una per ciascuno di essi. Se invece il programma mette tutti i segnali insieme questo passaggio può ovviamente essere saltato, ci complicheremo e semplificheremo la vita, in merito a questo processo, al Capitolo 8.

L'esportazione del segnale potrebbe dunque essere codificata inserendo il numero del soggetto (da 1 a 30) seguito dall'epoca: 1_bas, 1_rel, 1_str, 2_bas, 2_rel, 2_str, ..., 30_bas, 30_rel, 30_str.

In generale potremmo dire: n_epoca. Se i tre segnali non sono contenuti in ciascun file possiamo decidere se mettere in cartella con gli stessi nomi dando eventuali future istruzioni al computer del tipo: \EEG\7_rel, \EMG\7_rel e \ECG\7_rel per indicare i tre segnali del partecipante 7 durante la sessione di relax. In alternativa si può includere questa informazione nel nome invece che nella cartella: i nomi sarebbero dunque 7_rel_EEG, 7_rel_EMG e 7_rel_ECG. C'è dunque vasta possibilità di scelta ma

è comunque essenziale codificare i nomi, altrimenti l'approccio computazionale ai dati è compromesso sin dall'inizio.

In conclusione dunque è importante sottolineare come la raccolta dei dati psicofisiologici non sia la sola acquisizione di un segnale, ma piuttosto un processo di organizzazione e strutturazione sperimentale che, generando miliardi di dati, si deve preoccupare di avere un'organizzazione accessibile computazionalmente, ovvero per mezzo di algoritmi specifici che possano far leggere a un elaboratore le informazioni di cui lo sperimentatore necessita.

Infine si segnala un ottimo tool gratuito per effettuare cambi di nomi a intere cartelle in modo automatico (Figura 2.7).

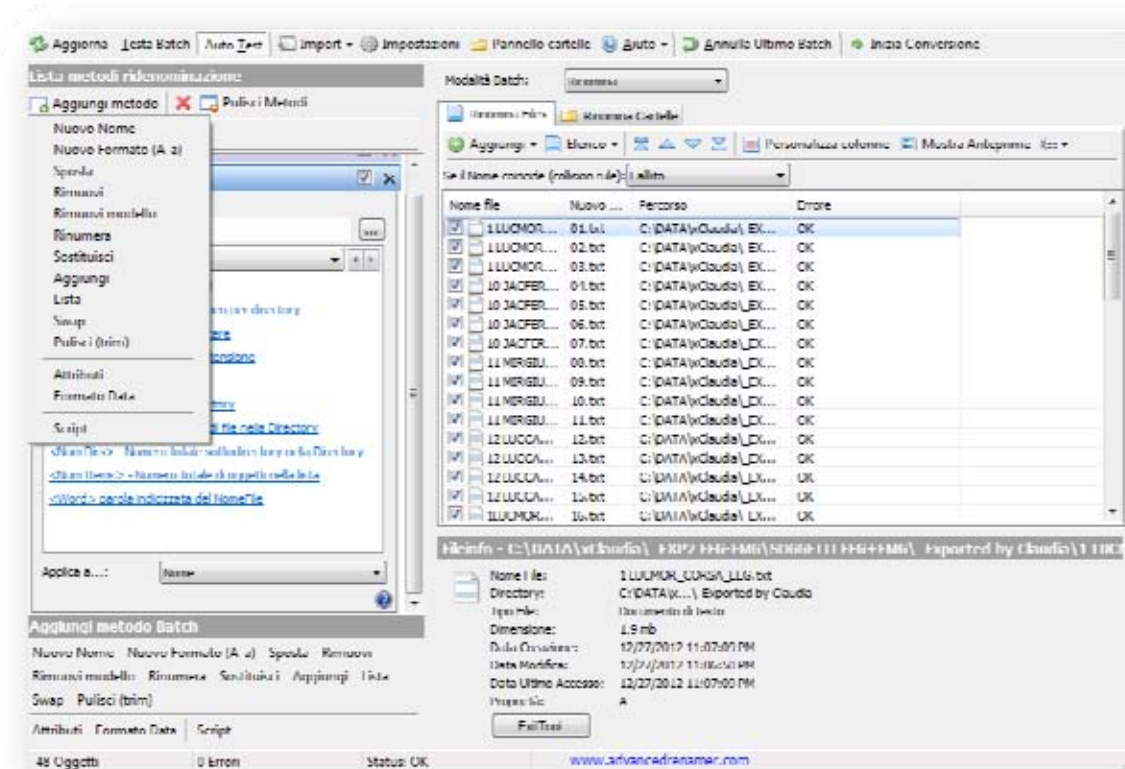


Figura 2.7. Interfaccia di Advanced Renamer 3.60.

Questo utilissimo tool per MS Windows è gratuitamente scaricabile da <http://www.advancedrenamer.com>.

In pratica trascinando al suo interno tutti i file che dobbiamo rinominare, avremo la possibilità di configurare nei più diversi modi i nuovi nomi sulla base di caratteristiche intrinseche (come ad esempio le prime 7 lettere) o estrinseche (come una sequenza numerica per tutti i file selezionati, vedi Figura 2.7, colonna "Nuovo"). Questa operazione ci permetterà di effettuare una codifica strutturata dei file, stando attenti a tenere comunque traccia delle associazioni tra vecchi nomi e nuovi nomi assegnati, magari lavorando su delle cartelle di backup, onde evitare disastrose cancellazioni dei file.

Movimenti oculari

L'analisi dei movimenti oculari è diventata sempre più usata negli ultimi anni grazie allo sviluppo di sistemi avanzati di monitoraggio a distanza, che non richiedono impiego di patches (come invece avviene per l'EOG visto nel capitolo precedente) agendo anche a circa un metro di distanza dal volto del soggetto.

Questo strumento, detto eye-tracker, consente un rapido e preciso monitoraggio dei movimenti oculari. Ne esistono di diversi tipi, per ogni esigenza, con costi che variano da circa i mille euro agli oltre centomila euro. Trovano impiego in diverse discipline dalla neurologia al marketing e vi è un vasto interesse in merito. Nel mondo clinico tra i paradigmi più comuni ci sono i classici saccadico, antisaccadico e smooth pursuit. Questi e simili paradigmi hanno scopi e misurazioni ampiamente validate e utilizzate dal mondo scientifico. Tuttavia l'Eye-tracker viene anche usato in modo più applicativo ed esplorativo verso stimoli visivi, onde poter valutare processi attentivi e cognitivi di vario tipo.

3.1 Definizione stimoli e acquisizione dati

Il processo di raccolta dati tramite eye-tracker, è molto simile a quello già descritto nel precedente capitolo in merito ai dati psicofisiologici. Valgono dunque le stesse regole per il disegno sperimentale e la conseguente estrazione dei dati.

In merito alla presentazione di eventuali stimoli, così come avviene per altre sperimentazioni in psicologia, si possono usare degli specifici software quali Presentation o E-Prime (a pagamento) o una moltitudine di tools gratuiti.

A volte gli eye-tracker dispongono di un loro software per la gestione della presentazione degli stimoli e questo può essere molto utile, in quanto evita di dover programmare un'adeguata sincronizzazione tra stimolo e movimento oculare.

Da un punto di vista sperimentale e applicativo la semplice analisi dei movimenti oculari non è sempre il principale oggetto di indagine. Se non si adotta uno specifico paradigma sperimentale uno studio si può concentrare maggiormente su specifiche regioni e, spesso, sul confronto tra diverse regioni.

I software in dotazione agli eye-tracker sono generalmente dotati di semplici interfacce grafiche che permettono, sia la definizione di una timeline sperimentale, sia la creazione di specifiche regioni sulle quali si concentrerà la raccolta dati.

Un esempio di timeline è quella in Figura 3.1, dove è possibile definire temporalmente la successione di stimoli da presentare ai

partecipanti. E' generalmente anche possibile la randomizzazione degli stimoli.

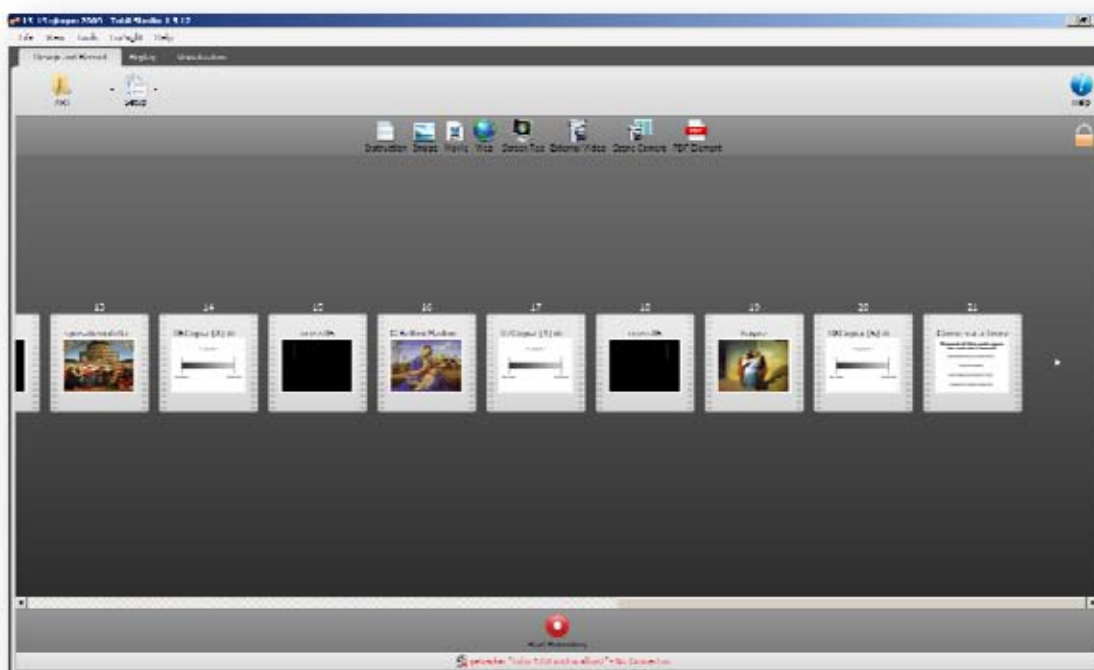


Figura 3.1. Definizione della timeline tramite il software Tobii Studio.

Nel seconda parte del volume impareremo la gestione avanzata dei dati e questo ci consentirà di poter definire specifiche aree o sessioni temporali anche a posteriori, senza doverci eccessivamente preoccupare di quale software abbiamo utilizzato per la presentazione degli stimoli.

Bisogna in generale tenere presente che a volte la semplificazione dei processi di elaborazione dei dati si paga con una minore flessibilità rispetto alle esigenze che si possono avere.

Molti software, tuttavia, permettono una personalizzazione delle caratteristiche di base, a patto di saper programmare e gestire le relative sdk (software development kit) appositamente rilasciate.

Alcuni produttori stanno anche optando per dei software ibridi, che consentono una gestione dell'eye-tracker con una logica di programmazione più "grafica", come si può vedere in Figura 3.2.

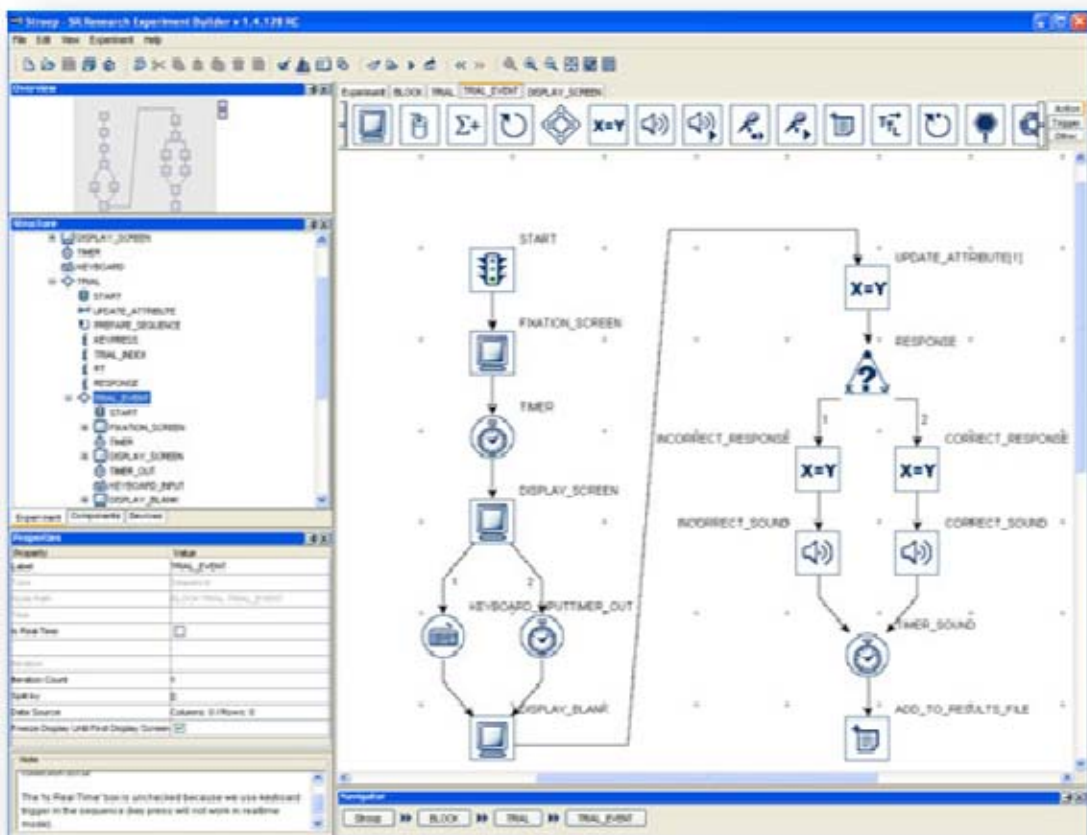


Figura 3.2. Programmazione ibrida tramite il software Experimental builder della SR Research.

3.2 Definizione delle aree di interesse specifico

Come già spiegato, spesso, si è maggiormente interessati a esaminare le osservazioni di specifiche aree di interesse e si definisce pertanto una porzione di stimolo codificata per analisi di comparazione tra queste diverse aree, ammesso ovviamente che questo abbia un senso, o in modo esplorativo, per una più profonda analisi e conseguente interpretazione.

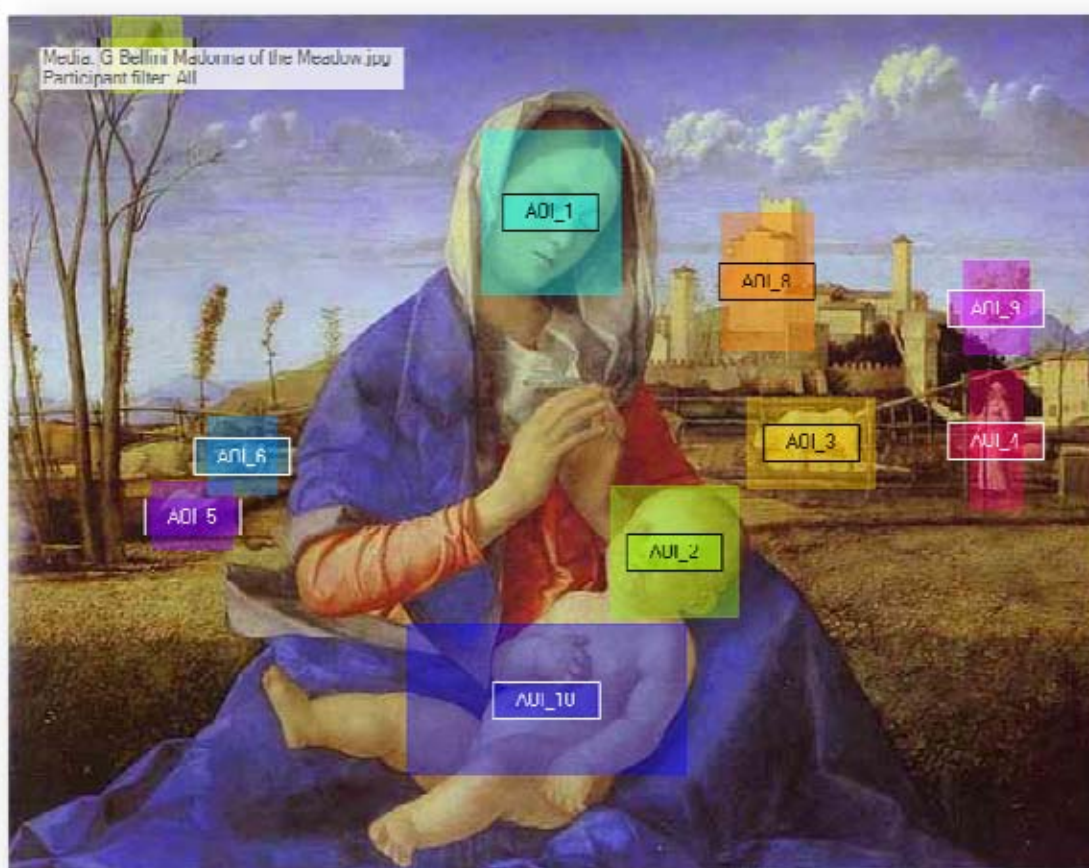


Figura 3.3. Definizione di aree di interesse (AOI, area of interest) rettangolari, in corrispondenza di elementi specifici.

Quello che è più interessante nella definizione di queste specifiche AOI, è che il software rende poi possibile un'estrazione dei dati con la specifica dell'area a cui questi appartengono.

Ci sono due tipi di estrazione dati possibili. La prima è l'estrazione dei valori di sintesi rispetto alle aree specifiche: si avranno ad esempio la media del numero di fissazioni in quell'area o il tempo totale trascorso nell'area e così via. Questi dati, essendo valori di sintesi sono assai semplici da trattare e spesso, infatti, vengono inseriti in un normale dataset come variabili (misure) relative ai vari casi (partecipanti) per la successiva analisi statistica. Il secondo tipo di estrazione è il dato grezzo delle coordinate relative ad ogni singolo punto in cui il soggetto ha guardato, con un marker per identificare l'AOI. Da un punto di vista spaziale generalmente il numero di punti è corrispondente al numero di pixel di presentazione dello stimolo. Dunque un'immagine presentata in formato 1024x768 (un classico formato di un monitor 4:3), presenterà lo stesso schema di coordinate (1024 in orizzontale e 768 in verticale) e i dati indicheranno per ogni rilevazione il punto esatto dello stimolo in cui il soggetto stava guardando, ad esempio 137, 325. Il punto (0,0) rappresenterà l'angolo in basso a sinistra dello stimolo e il punto (1024,768) rappresenterà l'angolo in alto a destra dello stimolo.

Da un punto di vista temporale, bisogna considerare che il movimento oculare è rapidissimo, in particolare i movimenti saccadici, come passaggi da un punto di fissazione a un altro. Inoltre, anche quando stiamo fissando un punto il nostro occhio

non è in realtà fermo, anche se noi lo percepiamo come tale. L'occhio è in continuo micromovimento, questo perché la corteccia visiva per elaborare l'informazione ha bisogno di variabilità (in condizioni di entropia non si può elaborare informazione, ma questo lo capiremo meglio dopo un'attenta lettura del capitolo 16). Questi micromovimenti attorno al punto di fissazione si chiamano microsaccadi e vanno considerate nell'elaborazione dei dati rilevati con eye-tracker. L'elevatissima velocità dei movimenti oculari richiede uno strumento preciso anche in termini temporali. Gli eye-tracker usati per scopi clinici e di ricerca, partono da oltre i 1000 Hz (ricordiamo: mille rilevazioni al secondo) fino arrivare anche a 2000 Hz. Quando invece lo strumento è usato in modo più applicativo o per ricerche dove non sia necessaria una precisione temporale tanto elevata, si usano eye-tracker da 50 o 60 Hz circa, meno di tale frequenza è generalmente sconsigliabile. Come si può facilmente intuire, anche le rilevazioni dei movimenti oculari generano con facilità diversi miliardi di rilevazioni, da cui l'importanza delle tecniche di psicomatria computazionale per la gestione di tale complessità.

3.3. Ispezione visiva dei movimenti oculari

Non appena effettuata la raccolta, si può procedere con una prima ispezione visiva dei movimenti oculari e delle relative fissazioni. Questo processo può essere assai utile anche per farsi una prima

idea su come sta procedendo la sperimentazione. Vi sono due tipologie di grafico che normalmente si usano per un primo controllo: il gazeplot e la heatmap.

Il gazeplot traccia una sequenza temporale delle fissazioni individuate per mezzo di circoletti la cui grandezza è tanto più grande quanto più ampia è la fissazione (Figura 3.4).



Figura 3.4. Gazeplot relativo a un'immagine (stimolo).

La heatmap, invece, ci dà una rappresentazione spaziale dei dati, prendendo di fatto la distribuzione dei movimenti oculari (o delle

fissazioni) nella finestra temporale di presentazione dello stimolo, come in Figura 3.5, dove si possono vedere le zone calde (heatmap significa appunto mappa di calore) per numero di fissazioni nell'arco dei 60 secondi di presentazione dell'immagine.



Figura 3.5. Heatmap relativa a un'immagine (stimolo).

3.4. Peculiarità dei dati relativi ai movimenti oculari

Dei dati relativi ai movimenti oculari abbiamo già detto (al paragrafo 3.2) che se si procede ad un'estrazione del dato grezzo si ottengono facilmente miliardi di dati. Come già visto nel capitolo precedente questi dati possono essere analizzati come segnale campionato e valgono qui tutte le considerazioni già fatte per il segnale psicofisiologico. Tuttavia i dati dei movimenti oculari hanno alcune peculiarità che necessitano un minimo approfondimento.

Innanzitutto nel calcolare le fissazioni bisogna necessariamente affidarsi alla letteratura scientifica oggetto della specifica indagine che si vuole effettuare. Alcuni software già considerano un certo valore, come raggio di azione del movimento, per definire la fissazione e in tal caso bisogna tenere presente che potrebbe essere necessario modificare questo valore. Se invece nei dati grezzi non troviamo l'indice relativo alla fissazione, possiamo procedere direttamente con il semplice calcolo. Stesso discorso vale per altri indici volessimo calcolare.

Altro aspetto che merita un'attenzione particolare nel trattare questi dati è la costante presenza di dati mancanti a intervalli regolari. Questo non è un problema dovuto allo strumento, bensì è dovuto a un normale processo fisiologico detto in termini medici ammiccamento (blinking), ovvero il naturale battere degli occhi che negli adulti è di circa 10 volte al minuto e nei bambini di circa 1 o 2 volte al minuto. Il blink dura normalmente da un decimo di

secondo ai due quinti di secondo. Quando gli occhi si chiudono il segnale con l'eye-tracker si interrompe e si generano sequenze di dati mancanti. In particolare, dunque, in una misurazione su un adulto, usando un eye-tracker che campiona a 1000 Hz, ci aspettiamo ogni circa 10.000 righe (10 secondi) tra le 100 e le 400 righe di dati mancanti. Molti eye-tracker registrano tra i dati il valore "-1" proprio per indicare l'assenza di segnale (dunque il blink). Questa considerazione non è banale e bisogna tenerne strettamente conto quando si analizzano questi dati, altrimenti si rischia di calcolare indici (come ad esempio una media) che incorporano una serie di valori "-1" alterando notevolmente la misurazione effettiva.

3.5 Pupillometria

La maggior parte degli eye-tracker, rileva, insieme al movimento oculare, anche il diametro della pupilla. L'analisi della misurazione delle variazioni della pupilla viene detta pupillometria e purtroppo è raramente presente negli articoli scientifici di psicologia, anche quando si fa uso dell'eye-tracker. Ciò è dovuto proprio alla complessità di estrazione dei dati relativi alla pupilla che richiedono una conoscenza, seppur di base, della psicometria computazionale. Auspico che la lettura di questo volume possa contribuire a una maggiore attenzione, da parte di chi si occupa di effettuare studi con eye-tracker, verso questa importante

misurazione, che viene tra l'altro indicata come ottima misura di intensità emotiva e carico cognitivo.

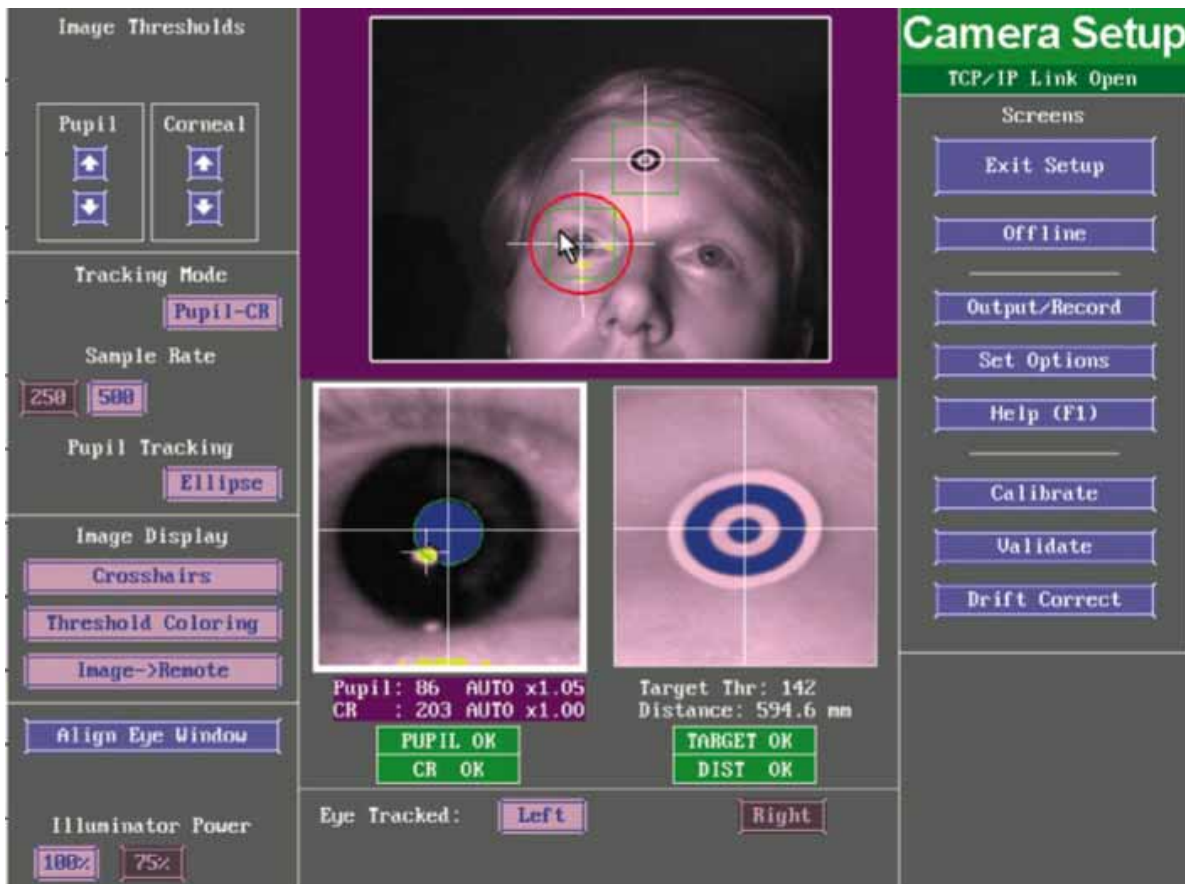


Figura 3.6. Interfaccia operatore di un eye-tracker EyeLink 1000 con misurazione della pupilla destra. Nel riquadro bianco con zoom sull'occhio si vede la pupilla individuata con un cerchio blu.

Anche nella pupillometria avremo ovviamente presenza di valori mancanti in corrispondenza dell'ammiccamento e dobbiamo tenerne conto nell'analizzare il segnale derivante dalla misurazione della dilatazione della pupilla.

Un'altra importante considerazione sulla pupillometria va fatta in termini di disegno sperimentale. Siccome la pupilla si restringe all'aumentare della luce e si dilata al suo diminuire, bisogna stare sempre attenti alla luminosità dei diversi stimoli (oltre, ovviamente, alla costanza di luminosità nel laboratorio), altrimenti si rischia di trarre conclusioni su differenze cognitivo-emotive che in realtà sono solo differenze di luminosità tra gli stimoli.

Dati comportamentali

L'analisi del comportamento è da lungo tempo studiata in psicologia, in neuroscienze e nelle scienze cognitive, con metodi di analisi sia qualitativi, sia quantitativi.

Le nuove tecnologie hanno radicalmente cambiato molti metodi di osservazione, permettendo misurazioni impensabili fino a qualche anno fa. Alcuni strumenti, esistono da decenni e restano ancora dei validi supporti alla misura del comportamento, quali le pulsantiere per l'interazione con quesiti specifici con la misurazione dei tempi di reazione. Altri strumenti, invece, sono diventati tool di ricerca in quanto entrati a far parte del vissuto quotidiano, come ad esempio gli smartphone che inglobando accelerometri, GPS e una connessione a Internet, possono diventare dei veri e propri "lettori del comportamento".

In questo capitolo scopriremo quali dati per la misura del comportamento possiamo estrarre dalle tecnologie che usiamo ogni giorno, mediante l'utilizzo di software gratuiti.

4.1 Periferiche come strumenti di laboratorio

Che in un laboratorio si usi il computer per misurare il comportamento è oggi scontato, ma che il comportamento che deriva dall'uso del computer sia oggetto di indagine, facendolo diventare uno strumento di misurazione per mezzo delle sue periferiche, potrebbe essere un ragionamento meno immediato. In particolare è certamente scontato l'uso del computer come elaboratore dei dati e come strumento per acquisirne di nuovi a partire da device esterni (periferiche), ma quello che si può anche fare è acquisire dati quantitativi del comportamento dalle periferiche che normalmente fanno parte integrante del computer stesso. Principalmente facciamo riferimento a tastiera, mouse, microfono e videocamera, caratteristiche di base oggi presenti in qualunque portatile o desktop, anche in quelli a basso costo.

Qualunque ricercatore dunque oggi ha a disposizione questi strumenti di laboratorio e renderli preziosi per misurare il comportamento è a questo punto questione di conoscenze e del software giusto, non certo questione di costo.

Cerchiamo, però, innanzitutto di capire come possono essere utili questi strumenti per raccogliere dati comportamentali.

Nel caso di tastiera e mouse l'utilizzo più immediato è lo stesso che si fa con le classiche pulsantiere per la risposta. Tuttavia, mentre le pulsantiere si devono interfacciare con specifici programmi, tastiera e mouse possono registrare dati da qualunque programma. E' ad esempio possibile esaminare il comportamento di pressione dei tasti o di uso del mouse durante la navigazione su

Facebook o, più in generale, mentre sto eseguendo un qualunque compito specifico all'uopo assegnatomi dallo sperimentatore.

L'idea è dunque quella di avviare la registrazione e tenere traccia di tutto ciò che si è fatto con tastiera e mouse fino a quando non si stoppa la registrazione stessa. Nell'eseguire questo tipo di sperimentazione bisogna però tenere sempre in considerazione alcuni aspetti metodologici di base. Innanzitutto bisogna individuare un punto di partenza o meglio un momento di partenza. Si può anche stabilire che il partecipante prema il tasto CTRL tre volte di seguito e parta a eseguire il compito subito dopo. Questa semplice operazione ci può aiutare a scartare i dati iniziali (tempi morti della sperimentazione). Ovviamente se si può capire l'esecuzione del compito dai tasti che ha premuto o dal nome del programma in cui è entrato, questa operazione non servirà in quanto questi eventi vengono tutti automaticamente archiviati. Un altro caso da considerare è invece quando si usi un altro software specifico, nella stessa sessione, che rileva autonomamente degli eventi (come l'avvio dei dati psicofisiologici). Infatti se questo avvio avviene sullo stesso computer, non ci sono problemi in quanto i due elenchi di eventi possono essere sincronizzati tramite l'orologio di sistema, se invece sono su diversi computer sarà necessario avere un marker di sincronizzazione o comunque un modo per sincronizzare gli orologi di sistema dei due diversi computer e questo è un'operazione più complessa.

In generale quindi usando lo stesso computer (anche se con un monitor esterno per la presentazione ai partecipanti), non ci sono problemi nell'archiviare tutti gli eventi.

Esistono molti prodotti, anche commerciali, per poter effettuare questa operazione, tecnicamente detta di logging, a mio parere comunque uno dei migliori è basicKeyLogger, open source (a codice aperto) e liberamente scaricabile dall'apposita pagina SourceForge: <http://sourceforge.net/projects/basickeylogger/>.

Una volta installato il programma in un'apposita cartella, saranno disponibili una serie di file ".exe" per far partire il programma. Il principale e più semplice modo per partire con la registrazione avviene eseguendo il file "startKeyLogger.exe" in conseguenza del quale apparirà la finestra di cui in Figura 4.1.

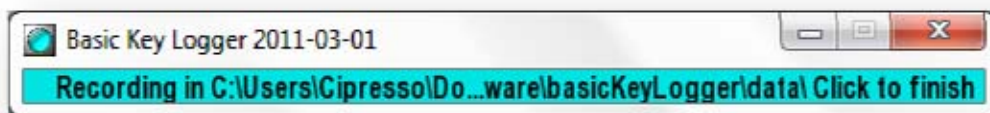


Figura 4.1. Avvio del logging di tastiera e mouse.

A questo punto si può eseguire (o fare eseguire al partecipante alla ricerca) il compito specifico e si terrà traccia di tutti gli eventi di tastiera e mouse con una precisione stimata di 10-15 millisecondi (ms), un impiego di capienza disco di 2.5 k/s per il log degli eventi e di 0.2 k/s per il log delle operazioni. Il software funziona

mediante una copia locale standalone di Python e dunque non avrà bisogno di ulteriori installazioni di software di terze parti.

Nella cartella del programma è presente una sottocartella data dove saranno memorizzati i due file log, riempiti sequenzialmente ad ogni riavvio dell'applicazione. Contenendo i timestamp, ovvero l'ora esatta per ciascuna rilevazione, non si rischiano sovrapposizioni. In Figura 4.2 sono state selezionate le righe di registrazione della pressione dei tasti durante la scrittura della parola "PROVA". Come si può notare sono presenti sempre due eventi, il primo indica la pressione e il secondo indica il rilascio del pulsante, sebbene la differenza di tempo tra questi due eventi è normalmente di pochi decimi di secondo (una scrittura a velocità sostenuta avviene a circa 120 battute al minuto, ovvero una media di due battute al secondo, quindi la pressione del tasto dovrebbe durare meno di 200 millisecondi). Guardando in figura le differenze tra timestamps, proprio a livello di millisecondi, si vede che la pressione del tasto e di poco più di cento millisecondi.

The image shows a screenshot of a log file with columns for time, event type, key code, and coordinates. The log entries are grouped by the word 'PROVA', with each letter having a corresponding 'keyDown' and 'keyUp' event. The time difference between these two events for each letter is consistently around 100-150 milliseconds.

Time	Event	Key Code	Time Diff (ms)
1718	mouseMoveWindow	keyLogger:release	-
1719	mouseMoveWindow	keyLogger:release	-
1720	mouseMoveWindow	keyLogger:release	-
1721	mouseMoveWindow	keyLogger:release	-
1722	mouseMoveWindow	keyLogger:release	-
1723	mouseMoveWindow	keyLogger:release	-
1724	mouseMoveWindow	keyLogger:release	-
1725	keyDown	keyLogger:release	124.0233
1726	keyUp	keyLogger:release	124.0233
1727	keyDown	keyLogger:release	124.0233
1728	keyUp	keyLogger:release	124.0233
1729	keyDown	keyLogger:release	124.0233
1730	keyUp	keyLogger:release	124.0233
1731	keyDown	keyLogger:release	124.0233
1732	keyUp	keyLogger:release	124.0233
1733	keyDown	keyLogger:release	124.0233
1734	keyUp	keyLogger:release	124.0233
1735	mouseMoveWindow	keyLogger:release	-
1736	mouseMoveWindow	keyLogger:release	-
1737	mouseMoveWindow	keyLogger:release	-
1738	mouseMoveWindow	keyLogger:release	-
1739	mouseMoveWindow	keyLogger:release	-
1740	mouseMoveWindow	keyLogger:release	-

Figura 4.2. Log dei dati raccolti scrivendo la parola "PROVA".

Una volta estratti i dati, è possibile fare alcune osservazioni sui dati, tramite dei semplici grafici, come in Figura 4.3 e calcolare tutti gli indici necessari, per ciascun evento, come in Figura 4.4.

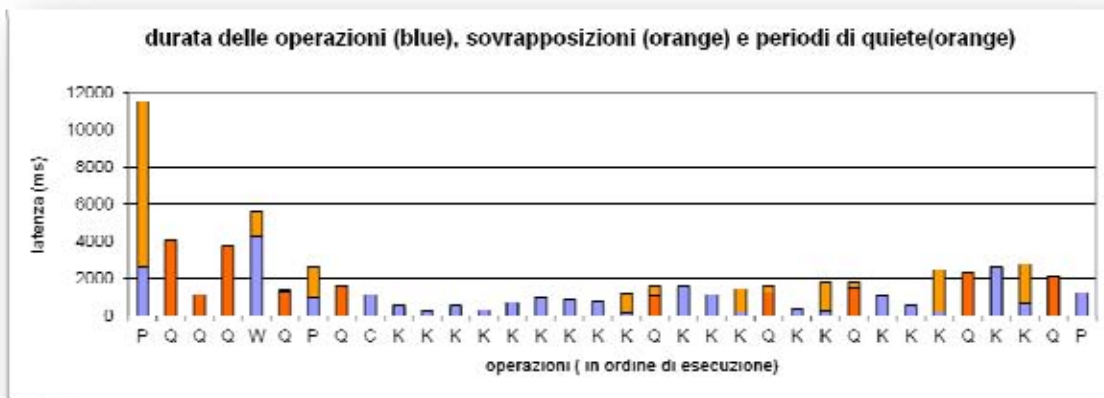


Figura 4.3. Grafico delle operazioni eseguite e relativa latenza.

Ks	19	overlap	strict	As		overlap	strict
total duration(s)	21.97	8.22	13.75	total duration(s)			
average (s)	1.16	0.43	0.72	average (s)			
stdev(s)	0.78	0.79	0.60	stdev(s)			
Ps	3	overlap	strict	Cs	1	overlap	strict
total duration(s)	15.39	10.56	4.84	total duration(s)	1.10	0.00	1.10
average (s)	5.13	3.52	1.61	average (s)	1.10	0.00	1.10
stdev(s)	5.58	4.76	0.86	stdev(s)			
distance (pix)	1547	(m)	0.44				
length (pix)	2831	(m)	0.80				
Ws	1	overlap	strict	Qs	10	overlap	strict
total duration(s)	5.578	1.312	4.266	total duration(s)	21.31	1.22	20.09
average (s)	5.578	1.312	4.266	average (s)	2.13	0.12	2.01
stdev(s)				stdev(s)	1.00	0.20	1.08
moved (step)	9						
number of steps	51						

Figura 4.4. Tabella con il calcolo delle operazioni eseguite.

E' anche possibile tracciare, con un semplice grafico, il percorso effettuato con il mouse e i relativi click, come si può vedere in Figura 4.5.

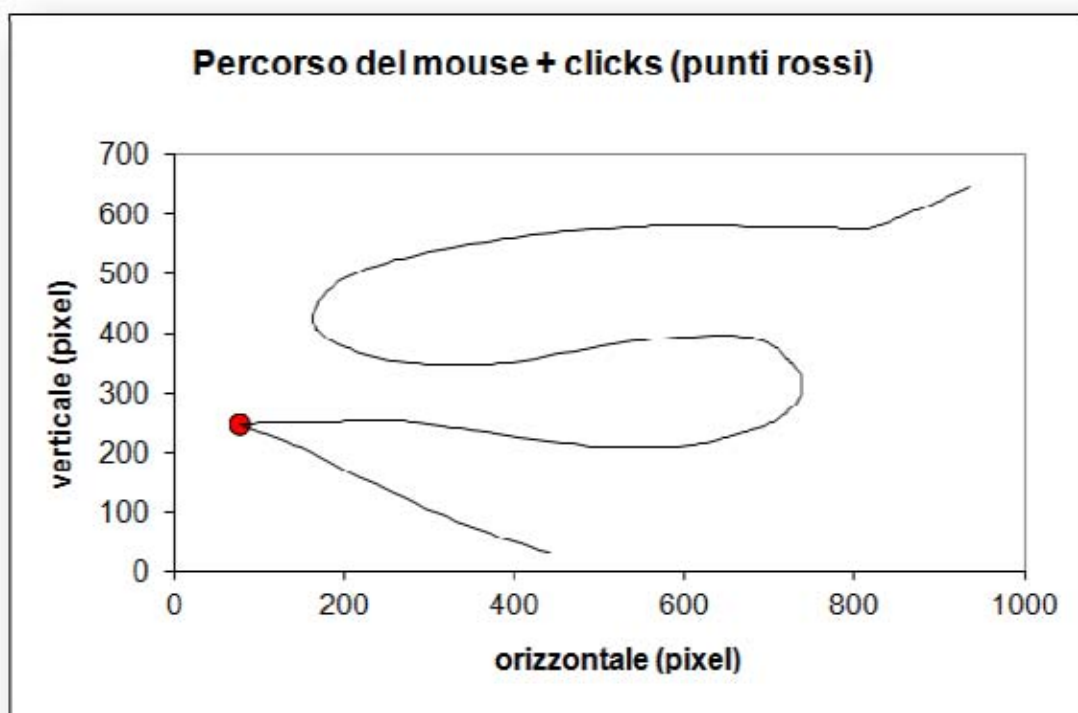


Figura 4.5 Percorso del mouse e relativi clicks effettuati.

In questa sede ci siamo limitati a effettuare qualche semplice analisi di insieme. Tuttavia questi dati possono risultare assai complessi, in virtù del contesto entro cui la sperimentazione opera. Nella seconda parte del volume si daranno gli strumenti atti a gestire anche dati molto complessi ed effettuare le più avanzate sistematizzazioni possibili anche a partire da questi strumenti.

Così come è possibile registrare dati da tastiera e mouse, allo stesso modo saremo capaci di rilevare importanti dati da altre periferiche e registrarli di conseguenza, onde poter effettuare attente sperimentazioni ed una più profonda analisi di nuovi possibili dati comportamentali. In particolare facciamo riferimento a microfono e video. Anche per questi strumenti il costo è molto basso e spesso si trovano già integrati su portatili e desktop anche di fascia bassa. Esistono sul mercato molti prodotti professionali per la gestione audio e video che possono essere utili per misure comportamentali di vario tipo. Di seguito daremo indicazione, come sempre nel volume, di software gratuiti e possibilmente a codice aperto, che possano costituire la base sia per la misurazione del comportamento, sia per usi sperimentali in psicologia.

Per la semplice registrazione audio, si può usare Audio Recording Interface (<http://sourceforge.net/projects/audiorec/>), un software che peraltro si interfaccia molto bene con basicKeyLogger, prima visto per il logging di tastiera e mouse. La comodità di Audio Recording Interface risiede nel poter partire e stoppare la registrazione audio prestando una combinazione di tasti da premere contemporaneamente come possiamo vedere in Figura 4.5. Se ad esempio vogliamo definire come combinazione di tasti da premere contemporaneamente il control di sinistra, lo Shift di sinistra (è il tasto per fare una singola lettera in maiuscolo) e le lettere "k" e "j", basterà effettuare questa pressione contemporanea nell'apposita finestra (Figura 4.6) e il programma genererà la sequenza: LCtrl+LShift+K+J.



Figura 4.6. Menu di configurazione di Audio Recording Interface.

Inoltre, sempre nel menu "Configure", andando sulla scheda "General" sarà possibile inserire l'opzione "Minimize to tray icon" che terrà il software in stand-by senza infastidire sul monitor (Figura 4.7).



Figura 4.7. L'icona del microfono sulla system tray indica che il software è in stand-by.

Con il software in stand-by (Figura 4.7), potrò in ogni momento iniziare la registrazione audio e stopparla usando la combinazione predefinita (opzione "toggle" in Figura 4.6) o farla partire e stopparla tramite le due diverse combinazioni (opzione "start" e opzione "stop" in Figura 4.6).

Audio Recording Interface si integra automaticamente con basicKeyLogger in quanto la combinazione di tasti prestabilita sarà normalmente loggata e quindi facilmente rilevabile tra i dati. Il timestamp relativo al rilascio dell'ultimo tasto della sequenza sarà anche il timestamp di avvio della registrazione. Allo stesso modo ricaviamo facilmente il timestamp di fine registrazione.

In pratica, dunque, quando facciamo una sperimentazione e raccogliamo i dati di tastiera e mouse, possiamo anche raccogliere i dati audio, tramite la pressione dei tasti che abbiamo predefinito e l'avvio della registrazione troverà un posto sulla nostra timeline sperimentale per via del logging della combinazione dei tasti.

Tutte le registrazioni sperimentali verranno salvate in una cartella pronte per l'analisi. L'elaborazione dei dati audio verrà trattata nella seconda parte del volume, con dei tools gratuiti che ci permetteranno di effettuare analisi avanzate. Qui invece vediamo come fare un primo rapido trattamento sui dati audio, in particolare tagliarli, operazione molto importante anche per la preparazione di stimoli audio sperimentali, infatti come abbiamo visto al Capitolo 2, a volte è utile definire un'unica sessione sperimentale dentro cui dividere in epoche predefinite e temporalmente precise. Dunque, ad esempio, per un disegno bilanciato che prevede due fasi controbilanciate e un primo periodo di baseline, creeremo due file audio: uno contenente il tempo di baseline (con silenzio), il primo audio e il secondo audio e l'altro file con baseline (con silenzio), il primo audio e il secondo audio. I due file avranno la stessa durata e, per

costruzione, epoche ben definite (ad esempio 2 minuti ciascuna). Lo sperimentatore si dovrà solo preoccupare a questo punto di assegnare un file a metà gruppo e l'altro ai restanti partecipanti. Per questa prima modifica dei dati audio, di cui abbiamo appena parlato, possiamo usare il software open source gratuito Audacity (<http://audacity.sourceforge.net/?lang=it>).



Figura 4.8. Due tracce divise in tre epoche per l'uso in sperimentazioni con stimoli audio.

Audacity è un programma molto semplice e intuitivo da usare. Si fa quasi tutto quel che serve selezionando, tagliando, copiando e incollando parti delle tracce, lavorando anche a più livelli per più tracce, contemporanee o staccate. Si può anche aggiungere il silenzio nel punto che desideriamo (menu "Generate" > "Silence...") e fare semplici collage di diversi pezzi audio. Audacity è dunque il programma ideale per manipolare l'audio

nelle sperimentazione: come prima descritto si possono creare tre parti di audio bilanciate, in due tracce separate da esportare singolarmente, una per file (come da Figura 4.8). Audacity è anche molto utile per una primissima analisi e correzione del segnale audio registrato a scopi sperimentali.

Veniamo adesso, invece, alla registrazione dei dati video. La registrazione video genera una maggiore quantità di dati ed è in generale più complessa.

Trattiamo qui della registrazione tramite webcam commerciali a basso costo o già incluse nel nostro portatile, per la registrazione con camere professionali si dispone generalmente di hardware e software specifico in dotazione. Della registrazione con stereocamere ci occuperemo invece nel prossimo paragrafo.

Per modificare e tagliare i video in modo non professionale si può usare Windows Movie Maker (<http://windows.microsoft.com/it-it/windows-live/movie-maker>) che ci permette di tagliare, copiare e incollare parti di video, così da poter creare una sequenza sperimentale, come abbiamo visto per l'audio.

Per registrare video da webcam, invece, possiamo usare il software open source gratuito ISPY (<http://www.ispyconnect.com>). ISPY nasce come software di videosorveglianza (Figura 4.9), ma risolve molti problemi anche in ambito sperimentale, consentendo molte funzioni, quali la registrazione continua, l'attivazione programmabile, anche su eventi, il controllo remoto e l'intervento tramite voce dello sperimentatore a distanza, anche via

smartphone. Può inoltre gestire più webcam alla volta e registrare anche su un server remoto i video.



Figura 4.9. Registrazione video con ISPY.

ISPI è anche in grado di riconoscere il movimento, la presenza di volti, o i codici a barre. Si può anche definire una lista di testi di codici a barre che una volta scansionati attivano un programma o un suono o qualche altro evento.

Esiste anche un'altra importante tipologia di video recording, oltre a registrare dalla webcam si può infatti voler registrare quanto accade a monitor. Questo è piuttosto frequente nelle sperimentazioni, soprattutto se il programma specifico che si sta usando prevede la visualizzazione ma non una registrazione.

Un programma di questo tipo è "on screen capture recorder to video free" (<http://sourceforge.net/projects/screencapturer/>), open source gratuito che consente di registrare tutto ciò che avviene a monitor in un file video mp4 a risoluzione nativa (Figura 4.10).

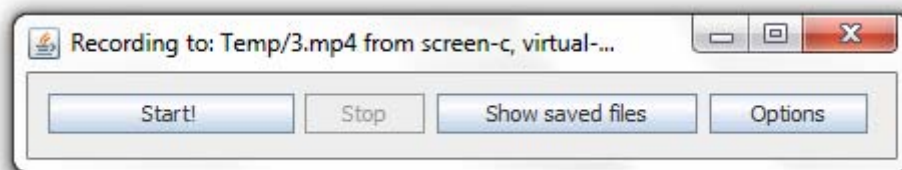


Figure 4.10. La semplice interfaccia di on screen capture recorder to video free.

Un'ultima parentesi rispetto al video la si deve fare per il download di video da YouTube, dove sono presenti molti "stimoli" gratuitamente messi a disposizione dagli utenti di tutto il mondo. Esistono molti software gratuiti per effettuare il download, ne segnaliamo uno open source: YouTube Video Downloader (<http://sourceforge.net/projects/ytdopensource/>), davvero molto semplice da usare. In pratica, si apre il programma, si inserisce il link, si aspetta, una volta comparse le varie opzioni sul formato di download se ne sceglie una e poi si preme download (Figura 4.11).

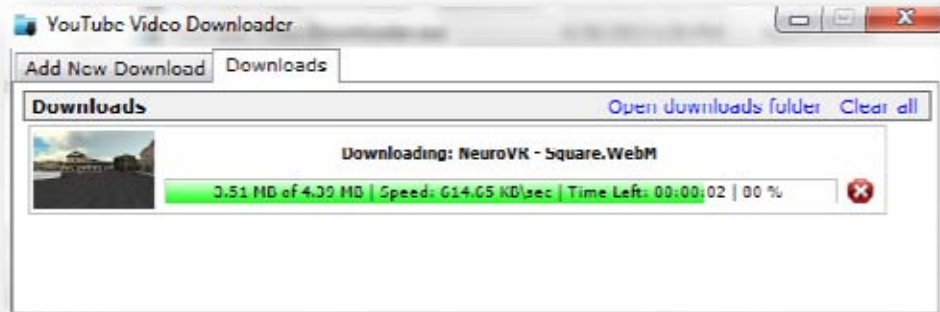


Figura 4.11. YouTube Video Downloader, il software open source gratuito per scaricare video da YouTube.

4.2 Raccolta dati comportamentali in laboratorio

Come abbiamo visto anche nel paragrafo precedente, esistono una moltitudine di dati che possono essere raccolti in un laboratorio. Non è lo scopo di questo volume fornire una trattazione sullo scopo della raccolta dati, per la quale rimandiamo a letture più focalizzate, piuttosto si vogliono fornire strumenti operativi e ove possibile gratuiti e open source per poter effettuare sperimentazioni e ricerche, qualitative e quantitative su più argomenti di ricerca: la psicomatria computazionale si occupa, infatti, di fornire strumenti di misura in psicologia mediante l'ausilio di computer e nuove tecnologie.

Abbiamo già visto come usare le periferiche di un normale computer per la raccolta di dati comportamentali, tuttavia adesso espanderemo questa trattazione presentando sia software per analisi avanzate di questi dati, sia possibili combinazioni di hardware a basso costo per un'efficace raccolta di tali dati.

Ci sono, innanzitutto, due tipologie di software che approfondiremo: i software per la ricerca qualitativa (anche detti Computer Assisted/Aided Qualitative Data Analysis, CAQDAS) e i software per la ricerca comportamentale sperimentale (behavioral experiment software). Non è possibile, in così poco spazio, fare un'analisi approfondita di tutti i software possibili, daremo dunque un piccolissimo approfondimento privilegiando, come sempre in questo volume, la disponibilità in open source, la gratuità e semplicità d'uso. Cerchiamo cioè di dare strumenti e metodi che possano essere usati da tutti.

Il panorama dei software gratuiti di analisi qualitativa è assai vasto e ve ne sono disponibili molti. Tuttavia molti di questi non sono molto immediati, da un punto di vista dell'utente non esperto. Nell'ottica di voler dare strumenti pragmatici per tutti, ho cercato di trovare un software che potesse essere abbastanza potente da fare delle serie analisi ma anche abbastanza semplice da poter essere usato anche da non esperti.

Un software che riscontra entrambe queste caratteristiche è ELAN (<http://tla.mpi.nl/tools/tla-tools/elan/>). ELAN è molto intuitivo e ha un'interfaccia grafica per visualizzare audio e video e per effettuare annotazioni di vario tipo. Il codice sorgente è disponibile per usi non commerciali e il software dispone di versioni per Windows, Mac OS X e Linux. E' un software molto versatile anche in termini di formati importabili, caratteristica da non sottovalutare in questa tipologia di ricerche.

ELAN offre un'interfaccia con schede, pulsanti e timeline molto intuitive, come si può vedere in Figura 4.12.

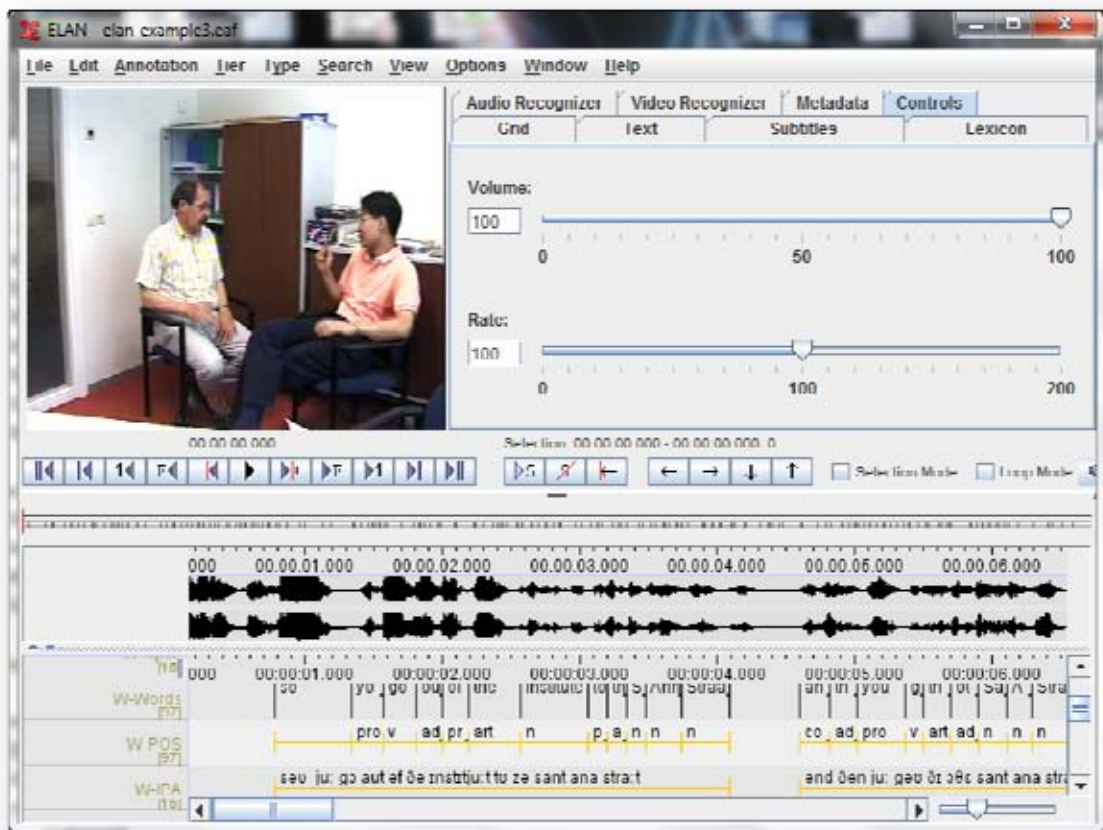


Figura 4.12. Interfaccia di ELAN.

Alla prima apertura di ELAN si può selezionare "File > New..." o premere CTRL+N per aprire la prima, importante interfaccia di importazione dei nostri file multimediali, necessari per l'analisi qualitativa.

In Figura 4.13 riporto una descrizione completa delle varie parti da settare per un corretto utilizzo. Queste impostazioni ci permettono

di importare automaticamente audio e video che tramite il software verranno annotate puntualmente, seguendo una precisa timeline.

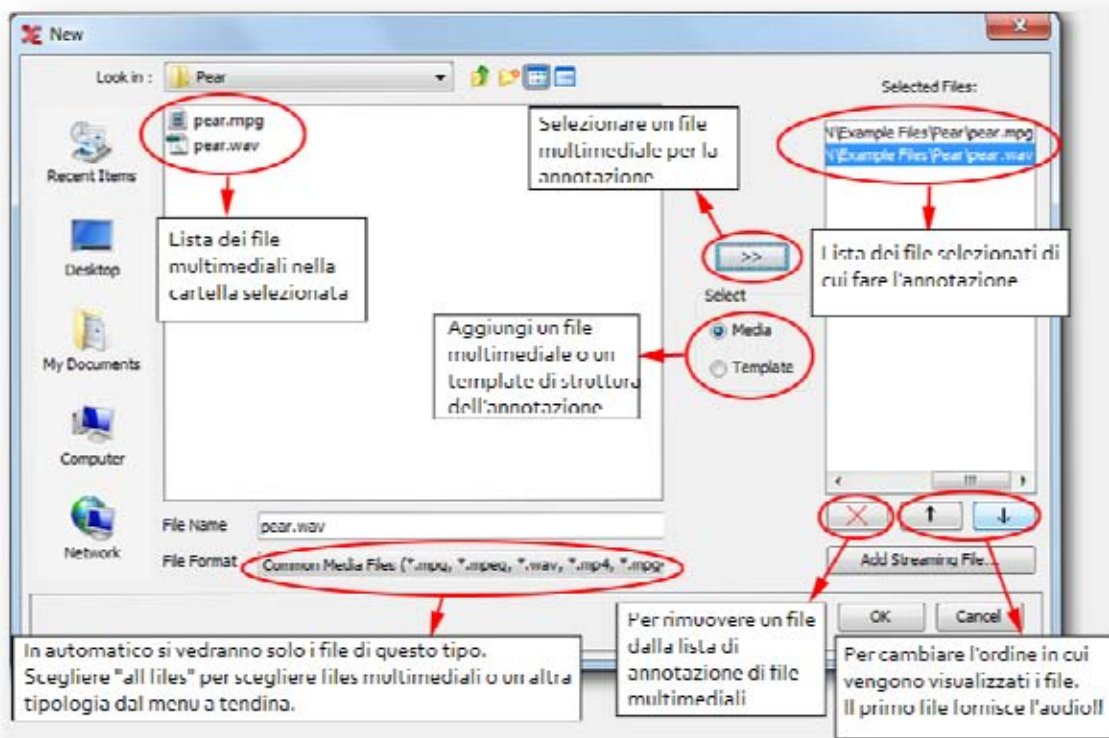


Figura 4.13. Creazione di un progetto in ELAN.

Per effettuare una semplice annotazione basterà andare sulla timeline in basso e selezionare la parte che vogliamo annotare anche direttamente tenendo premuto il tasto SHIFT e selezionando la parte con il mouse, come in Figura 4.14. Basterà poi cliccare con il tasto destro e scegliere "New Annotation Here" per aprire una casella di testo che permetterà l'annotazione.

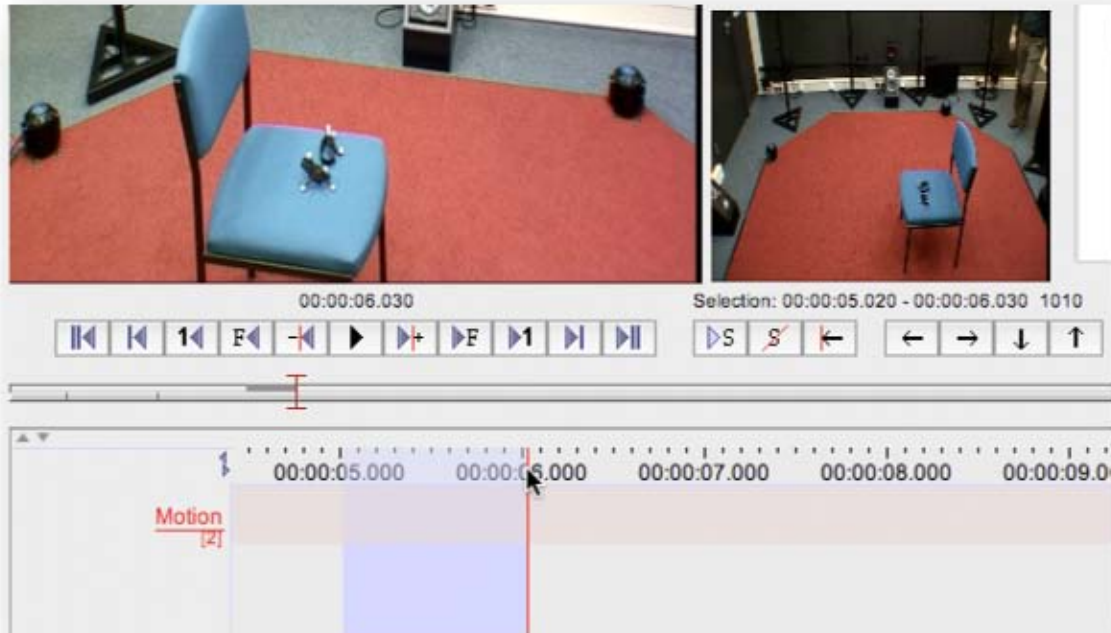


Figura 4.14. Annotazione in ELAN.

Con ELAN è anche possibile lavorare su più livelli di annotazione (tier) che possono essere creati sia per individuare diversi annotatori, sia per separare diverse caratteristiche del file multimediale (gesti, lingua usata, ...).

Per aggiungere, modificare e configurare i vari livelli basta andare su "Tier > Add New Tier" e impostare tutte le descrizioni di ciascun livello come da Figura 4.15.

Il risultato sarà, come mostrato in Figura 4.16, una serie di timeline differenziate, una per ogni livello definito.

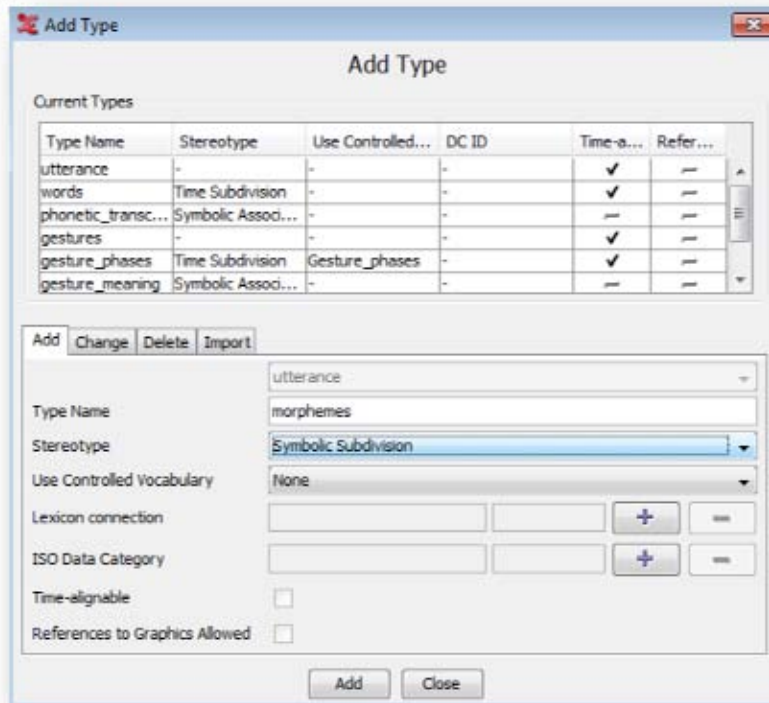


Figura 4.15. Aggiunta di un tier sul quale fare le annotazioni.

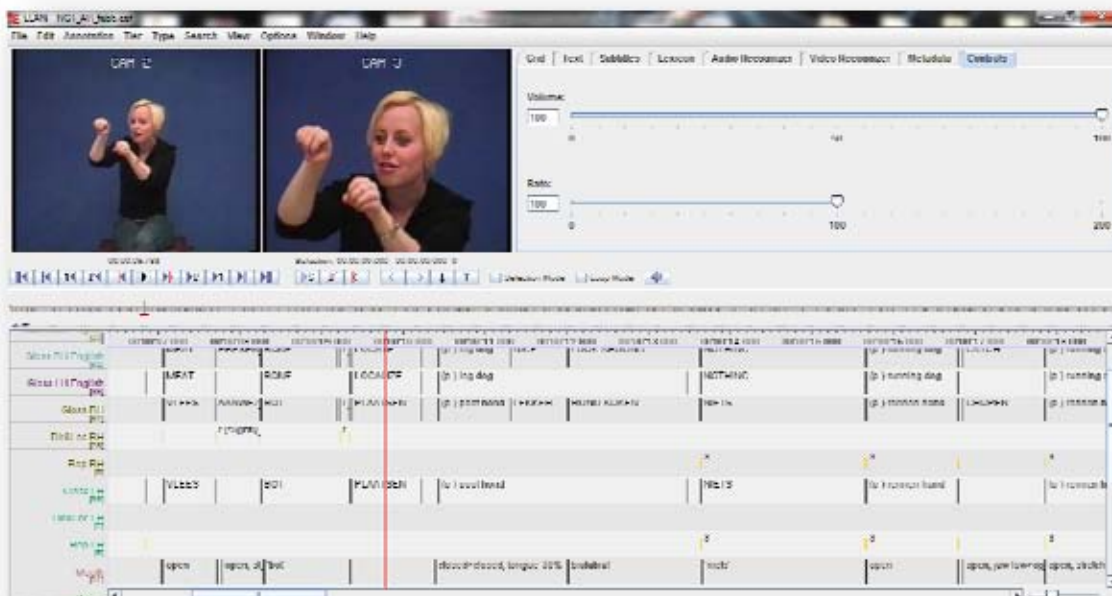


Figura 4.16. Timeline con vari livelli di annotazione.

Abbiamo visto come annotare i file multimediali tramite ELAN per raccogliere dati qualitativi. Esistono molti altri software per l'annotazione multimediali e per la ricerca qualitativa anche per i file testuali. Tra questi uno particolarmente flessibile, che per completezza segnalo, è RQDA (R Quality Data Analysis, <http://rqda.r-forge.r-project.org>), un pacchetto R (<http://www.r-project.org>) per l'analisi dei dati qualitativi: in particolare è ottimo per le analisi dei dati testuali, importabili anche come pdf. Si presenta inoltre con una semplice interfaccia, come si può vedere in Figura 4.17. Nella seconda parte del Volume vedremo come installare e configurare R ed RStudio e in particolare come installare i vari pacchetti come RQDA. Esiste anche un video corso completo in 24 lezioni (la momento in cui scrivo) su RQDA: <http://www.youtube.com/user/RQDAuto/videos?view=1/>

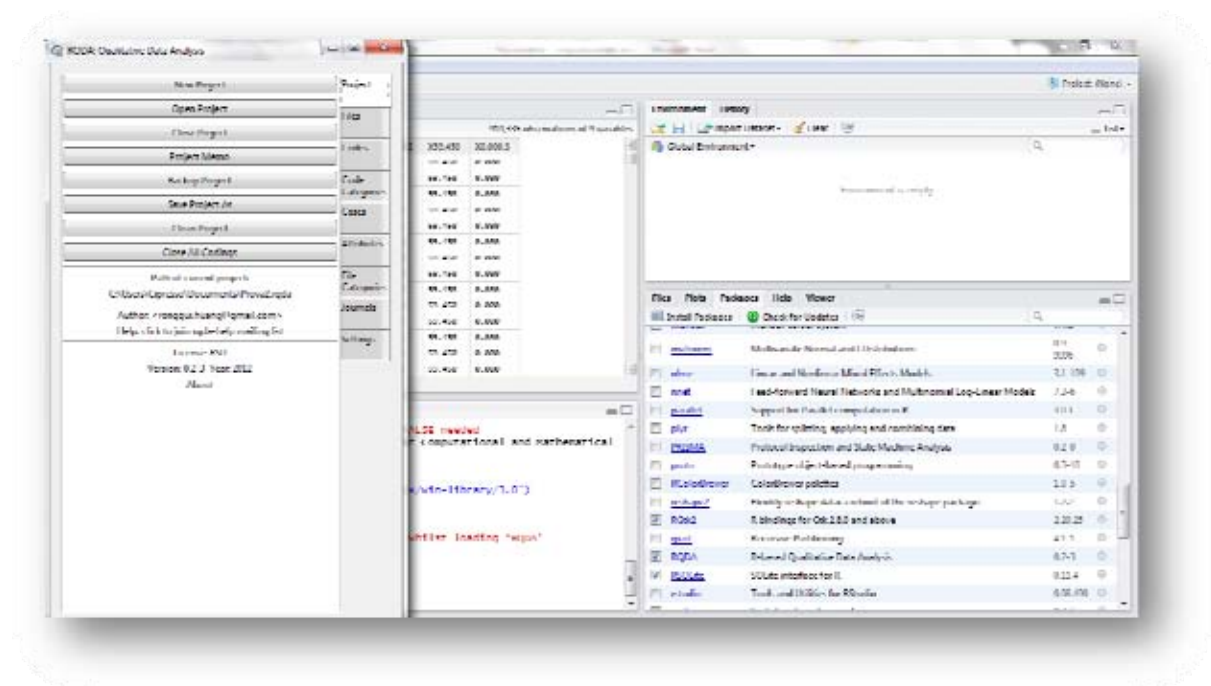


Figura 4.17. Interfaccia di RQDA in RStudio.

Come già prima detto, visti i software per l'analisi qualitativa, presentiamo, sempre nell'ambito della raccolta di dati comportamentali, i software per la ricerca comportamentale sperimentale (behavioral experiment software).

Abbiamo già presentato alcuni strumenti utili per la raccolta delle informazioni tramite tastiera e mouse, che rientrano a pieno titolo in questa categoria. Adesso vediamo come poter fare ricerche e paradigmi sperimentali più avanzati in un setting di laboratorio.

Esistono molti software a pagamento per questo tipo di analisi, come E-Prime (<http://www.pstnet.com/eprime.cfm>), Presentation (<http://www.neurobs.com>), Inquisit (<http://www.millisecond.com>) e Paradigm (<http://www.paradigmexperiments.com>). Spesso quando si compra un hardware per la ricerca comportamentale sperimentale, come una pulsantiera per i tempi di reazione o un eye-tracker (come visto nel capitolo 3), il produttore mette a disposizione anche una licenza di un software a pagamento, ottimizzato per quello specifico hardware.

Tuttavia molto spesso è necessario fare delle sperimentazioni per la raccolta di dati comportamentali ma non si ha a disposizione un software di questo tipo. Un caso tipico è in neuropsicologia, quando è necessario analizzare alcuni comportamenti con dei paradigmi specifici per indagare alcune funzioni cognitive.

Mi è personalmente successo più volte vedere Colleghi fare un Go/no go task senza nessuno strumento, battendo il dito sul tavolo.

Fattibile? Sì. Corretto? Sì. Ci si può chiedere dunque a cosa possa servire automatizzare un processo che si faceva ben prima dell'avvento delle nuove tecnologie.

Non esiste un modo semplice per rispondere a questa domanda, tuttavia ci sono elementi che tramite un computer e il software giusto è possibile raccogliere in modo assai semplice: ad esempio la misurazione dei tempi di risposta, che è rappresenta una variabile importante in psicologia.

Se pertanto è vero che non tutti dispongono di hardware specifico per la raccolta di dati comportamentali, è anche vero che quasi tutti dispongono di un computer e che questo può essere usato fattivamente per rispondere in modo più preciso e completo alle esigenze degli psicologi.

Come dicevamo prima esistono molti software su questi temi e anche qui ho cercato di conciliare la disponibilità in forma gratuita del software con la semplicità di utilizzo, ma con un elemento in più che in questa tipologia di software può fare la differenza: la possibilità di intervenire sulle modalità standard creazione sperimentale tramite un apposito codice, possibilmente semplice e accessibile anche a non esperti.

Il software che risponde a tutte queste complesse esigenze è OpenSesame (<http://osdoc.cogsci.nl>), gratuito, open source, con una semplicissima interfaccia grafica e programmabile in Python, come si può vedere in Figura 4.18.

OpenSesame consente di costruire esperimenti anche molto complessi con il minimo sforzo.

Il plug-in framework e il Python scripting consentono di incorporare device e tecnologie esterne quali pulsantiere, eye-trackers e device su porta parallela.

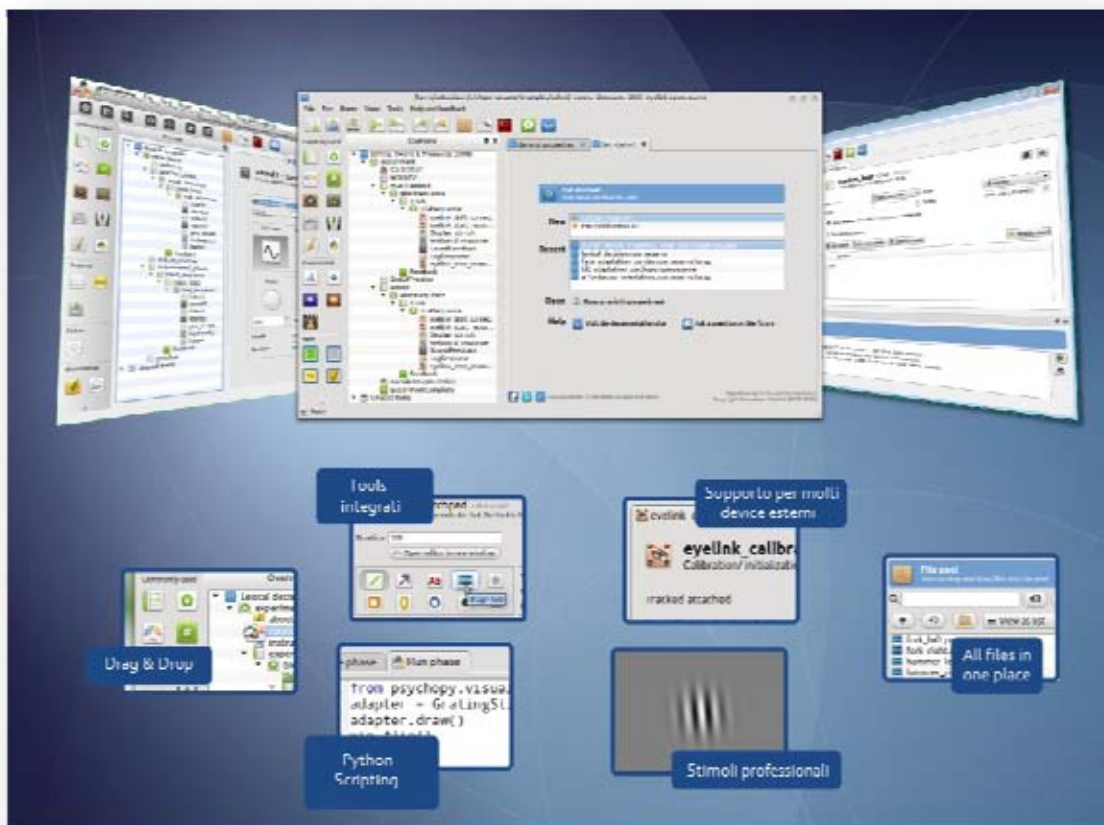


Figura 4.18. Interfaccia e caratteristiche principali del software OpenSesame.

Con OpenSesame si può ad esempio progettare una sperimentazione di presentazione di stimoli rendendola attiva tramite una continua interazione del partecipante, tramite risposte

e una conseguente raccolta dati di varia natura (tempi di risposta, movimenti oculari, ...), il tutto con semplici opzioni grafiche (Figura 4.19)

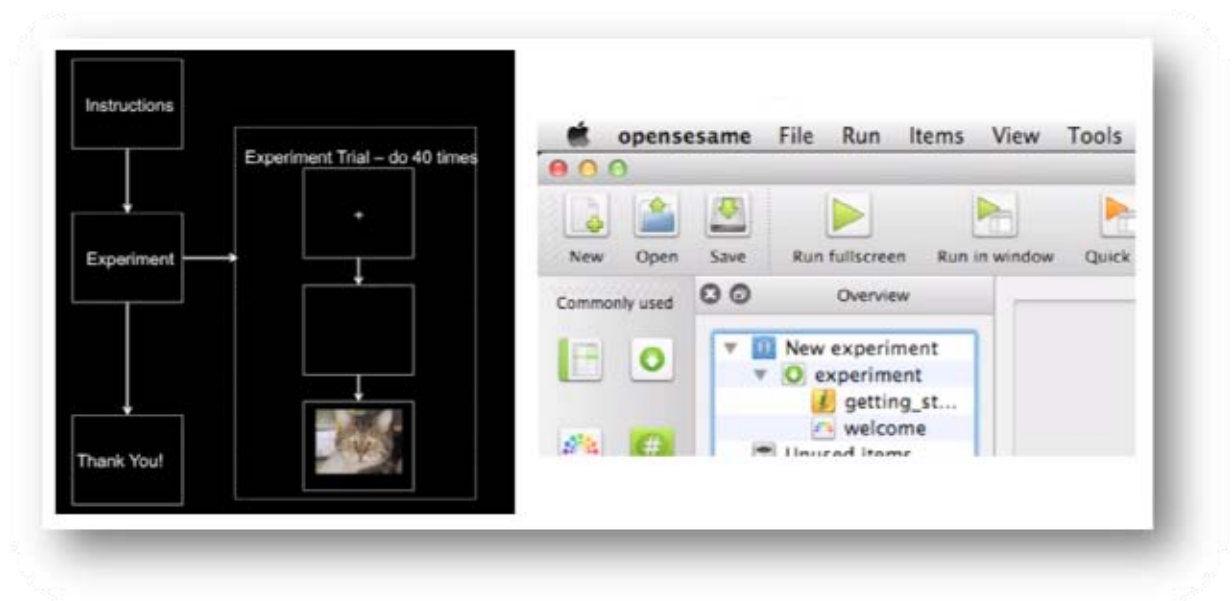


Figura 4.19. Il processo sperimentale con OpenSesame.

Non mi dilungherò a spiegare come costruire una sperimentazione con OpenSesame, perché ha un'interfaccia talmente semplice e intuitiva che una volta installato basterà usarlo uno paio di volte per capire subito come funziona. In pratica, comunque, basta fare un "drag and drop" (trascinamento) degli oggetti presenti nella barra degli articoli nell'area overview e definire i singoli contenuti nell'area TAB, come si può vedere in Figura 4.20.

Realizzare ad esempio sperimentazioni di tipo RSVP (Rapid Serial Visual Presentation) sarà dunque molto semplice con OpenSesame e contemporaneamente sarà possibile interfacciarsi con molti

device esterni, quali eye-trackers o EEG, triggerando i dati (ovvero avendo a disposizione dei markers, come visto nel capitolo 2).



Figura 4.20. Processo sperimentale di utilizzo di OpenSesame.

Nell'utilizzo di OpenSesame una caratteristica molto importante è il poter creare un loop (ciclo a ripetizione) con degli stimoli che cambiano uno dopo l'altro, possibilmente anche randomizzati da soggetto a soggetto.

Basterà caricare le immagini (stimoli) nel programma e poi definire quando e come visualizzarle con le funzioni di loop e pool dei file (Figura 4.21).

All'inizio, per orientarsi in OpenSesame, è consigliabile esplorare gli esempi messi a disposizione dal programma stesso nella cartella examples (File > Open).

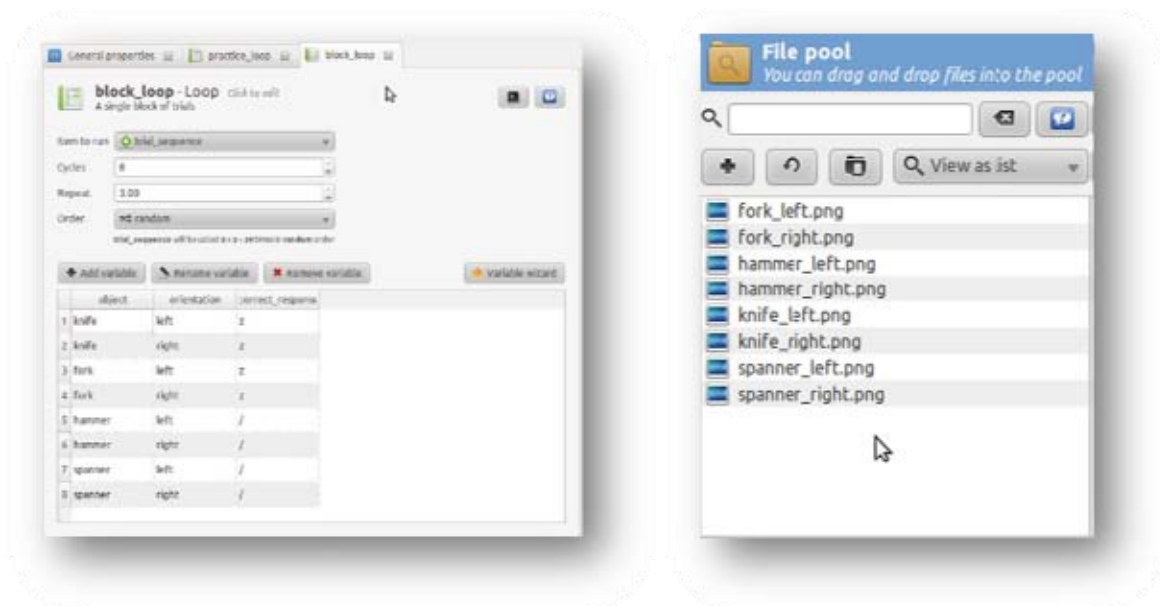


Figura 4.21. Creazione di un loop (ciclo) di immagini (stimoli).

Come prima detto, in OpenSesame è anche possibile usare Python per eseguire dei comandi ed effettuare operazioni più avanzate. Il caso più semplice è quando vogliamo tracciare delle figure geometriche o simili come stimoli.

```
#
# Questo è un commento
# Comandi per tracciare varie tipologie di stimoli (uno per riga)
#
draw ellipse [left] [top] [width] [height] [keywords]
draw circle [x] [y] [radius] [keywords]
draw line [left] [right] [top] [bottom] [keywords]
draw arrow [left] [right] [top] [bottom] [keywords]
draw textline [x] [y] [text]
draw image [x] [y] [path]
draw gabor [x] [y]
draw noise [x] [y]
draw fixdot [x] [y]
```

Nel codice sono riportati tra parentesi quadre i parametri spiegati nella Tabella 4.1.

left	La coordinata x più a sinistra
right	La coordinata x più a destra
top	La coordinata y più in alto
bottom	La coordinata y più in basso
x	La coordinata x
y	La coordinata y
text	Stringa di testo
path	Percorso al file di un immagine

Tabella 4.1. Parametri relativi al codice di tracciamento delle figure geometriche e degli altri elementi.

Altre istruzioni in codice possono, per esempio essere date per certe operazioni effettuate con il mouse. Il seguente semplice script consente di muoversi attorno a un punto di fissazione con il mouse fino a quando non viene premuto un bottone.

```
# uso della classe mouse usata per raccogliere input dal mouse
from openexp.mouse import mouse
from openexp.canvas import canvas
my_mouse = mouse(exp)
my_canvas = canvas(exp)
```

```

while True:
    button, position, timestamp = my_mouse.get_click(timeout=20)
    if button != None:
        break
    pos, time = my_mouse.get_pos()
    my_canvas.clear()
    my_canvas.fixdot(pos[0], pos[1])
    my_canvas.show()

```

Invece, il seguente script aspetta che vengano premuti 'z' o 'x' con un timeout di 3000 ms (millisecondi).

```

# uso della classe keyboard usata per raccogliere input dalla tastiera
from openexp.keyboard import keyboard
my_keyboard = keyboard(exp, keylist=['z', 'x'], timeout=3000)
start_time = self.time()
key, end_time = my_keyboard.get_key()
self.experiment.set('response', key)
self.experiment.set('response_time', end_time - start_time)

```



Infine ricordiamo che su OpenSesame è possibile configurare device esterni come l'eye-tracker (vedi anche Capitolo 3).

```

# uso dello script per l'eye-tracker EyeLink.
exp.eyelink.log("Message for the EDF")
time, start_pos = exp.eyelink.wait_for_saccade_start()
time, start_pos, end_pos = exp.eyelink.wait_for_saccade_end()
x, y = exp.eyelink.sample()

```

Si possono usare direttamente le PyLink API, ma si raccomanda l'uso delle funzioni libeyelink che sono più facili da usare.

Abbiamo visto come fare complesse sperimentazioni di laboratorio raccogliendo una moltitudine di dati sperimentali, qualitativi e quantitativi. Il panorama dell'analisi comportamentale è comunque assai vasto ed esistono moltissimi altri software e hardware che potrebbero essere esaminati, ma non certamente nel breve spazio di questo volume. Tuttavia ci sono tecnologie che non abbiamo ancora visto e che offrono a costi assai ridotti molte possibilità, grazie all'enorme sviluppo avuto negli ultimi anni in tal senso. Una di queste tecnologie è la realtà virtuale, della quale parleremo in modo più approfondito nel Capitolo 6, che consente molte sperimentazioni anche nell'ambito comportamentale, superando i limiti, imposti dal mondo fisico, quali gravità e vincoli spazio-temporali. Un'altra tecnologia che invece accenneremo subito è la Microsoft Kinect (Figura 4.22) o similari (come la ASUS xtion). La Kinect nasce come prodotto commerciale per il gaming, ma trova oggi vasto uso in psicologia, riabilitazione e neuroscienze.



Figura 4.22. Microsoft Kinect per Windows.

La Kinect ha una telecamera RGB (Red, Green, Blue) e un sensore di profondità doppio a raggi infrarossi con in più una telecamera sensibile alla stessa banda.

Nel mondo del gaming la Kinect è usata per riconoscere il giocatore e farlo interagire in tempo reale con il gioco stesso.

Nel mondo della psicomatria computazionale ci sono diversi usi possibili. Sintetizzando al massimo viene usata sia come periferica di input in tempo reale, sia come unità di misura a distanza del comportamento. Nella prima prospettiva ha un ruolo importantissimo nella riabilitazione: negli ultimi anni si sono moltiplicati gli studi scientifici sull'uso della Kinect, in particolare con pazienti colpiti da ictus, per i quali la riabilitazione motoria mediata da un feedback diretto e immediato, è importantissima. Trova comunque, sempre in quest'ambito, vasto utilizzo nella riabilitazione cognitiva e, in generale, in psicologia. Nella seconda prospettiva, dell'utilizzo come strumento di misurazione, viene usata principalmente o tramite la rilevazione dei gesti (gesture analysis) o tramite processamento video (video processing analysis). L'uso a fini scientifici della Kinect, è dunque molto importante e sempre più utilizzato, tuttavia richiede una certa complessità di gestione e programmazione e competenze multidisciplinari tra elementi di diversi gruppi.

E' dunque un ambito di ricerca molto complesso, ma che si sta sempre di più aprendo al grande pubblico (non solo del gaming) e che offre oggi anche degli strumenti semplici e gratuiti per la ricerca e la misura in psicologia. Vediamo dunque, di seguito,

come poter usare Kinect per delle applicazioni nell'ambito della psicologia.

Innanzitutto, è importante sottolineare che esiste già una comunità, OpenKinect (<http://openkinect.org/>) (Figura 4.23), di oltre duemila ricercatori, incluso chi scrive, che attivamente contribuiscono gratuitamente allo sviluppo di librerie open source per la creazione delle migliori applicazioni possibili per la Kinect.



Figura 4.23. OpenKinect, la più grande comunità di sviluppo per la Kinect, sviluppa librerie per ogni tipo di applicazione.

Vi sono già disponibili moltissime librerie e applicazioni pronte per l'uso con Kinect ed è impossibile darne spiegazioni nel breve spazio di questo volume. L'invito che faccio a chi fosse interessato è quello di partecipare alle varie attività della community, se non si è un programmatore, come utilizzatore. La Comunità è molto attiva e la maggior parte delle applicazioni possono essere scaricate e provate gratuitamente. Per cominciare a lavorare anche con periferiche esterne si può indagare ulteriormente OpenNI (<http://openni.org>), Open Natural Interaction, che promuove e certifica la compatibilità e l'interoperabilità di device, applicazioni e middleware di vario tipo. In OpenNI, nella sezione download,

sono disponibili moltissimi software, alcuni dei quali a pagamento, ma una demo è sempre disponibile per chi si iscrive al sito (gratuitamente). Provare a installare questi software è un utilissimo esercizio per cominciare a lavorare fattivamente con la Kinect, iniziando anche a raccogliere dati di ricerca o trovando soluzioni a problemi complessi: è ad esempio possibile scaricare software che tramite kinect ci consentono di realizzare in casa uno scanner 3D, cosa fino a qualche anno fa impensabile a basso costo. Altre applicazioni, invece, si concentrano sul più tradizionale riconoscimento delle pose, che ovviamente ha importantissime applicazioni nell'ambito della psicologia.

Va qui aperta una parentesi metodologica, ci sono infatti due possibili modalità di riconoscimento che cercherò di spiegare in modo intuitivo e non formale. Un riconoscimento è quello che si basa su un modello già disponibile ed è quindi il riconoscimento di un gesto comune, come ad esempio un saluto (un uomo che alza il braccio e agita la mano). Un altro tipo di riconoscimento invece è quello di costrutti di comportamento: ad esempio riconoscere lo stress in un individuo sulla base di come si comporta, ad esempio toccandosi la testa, giocando con le sue stesse mani o facendo tremare vistosamente una gamba. Questo secondo tipo di riconoscimento richiede tecniche matematiche e informatiche abbastanza avanzate, oltre a una forte competenza in psicologia e statistica, fa parte insomma della psicomatria computazionale avanzata, che non tratteremo ovviamente qui. Per i curiosi, il riconoscimento dello stress dai comportamenti usando una Kinect

esiste veramente ed è stato ideato e pubblicato da un gruppo di ricercatori Greci, Inglesi e Italiani (tra cui il presente) su PLoS ONE, una nota rivista che consente l'accesso gratuito all'articolo (<http://dx.plos.org/10.1371/journal.pone.0043571>).

Un'altra importante distinzione da fare è tra pose e gesti: la posa è una posizione del corpo che può essere rilevata tramite Kinect (ad esempio un uomo con un braccio alzato), invece la gesture è più complessa in quanto ingloba la dinamica di passaggio da una posa a un'altra (ad esempio è un uomo che alza un braccio). Lavorare sulle pose è abbastanza semplice, invece le gesture richiedono una complessità maggiore.

Se la necessità, soprattutto per iniziare, è solo quella di definire e riconoscere semplici gesture si può usare KinectDTW (<http://kinectdtw.codeplex.com>) - Kinect SDK Dynamic Time Warping (DTW) Gesture Recognition. Il programma è ottimo e oltre essere gratuito è anche a codice aperto. A dispetto del nome, dopo averlo provato molte volte devo dire che difficilmente riesce a catturare le gesture addestrate, ma si comporta molto bene con le quelle molto semplici, a patto che l'addestramento sia fatto con estrema pazienza.

Il programma gode di una semplice interfaccia, come si può vedere in Figura 4.24 ed è utilizzabile anche senza avere competenze matematiche e informatiche particolari. La cosa importante è addestrare il software in modo procedurale, seguendo le semplici indicazioni e il contatore di fps (frame per second). Una volta riconosciuta la gesture non c'è comunque molto che si possa fare

se non si sa programmare a un certo livello. Detto ciò è un ottimo esercizio per iniziare a capire le dinamiche della Kinect e magari iniziare qualche collaborazione con un collega più esperto sulla parte informatico-matematica.

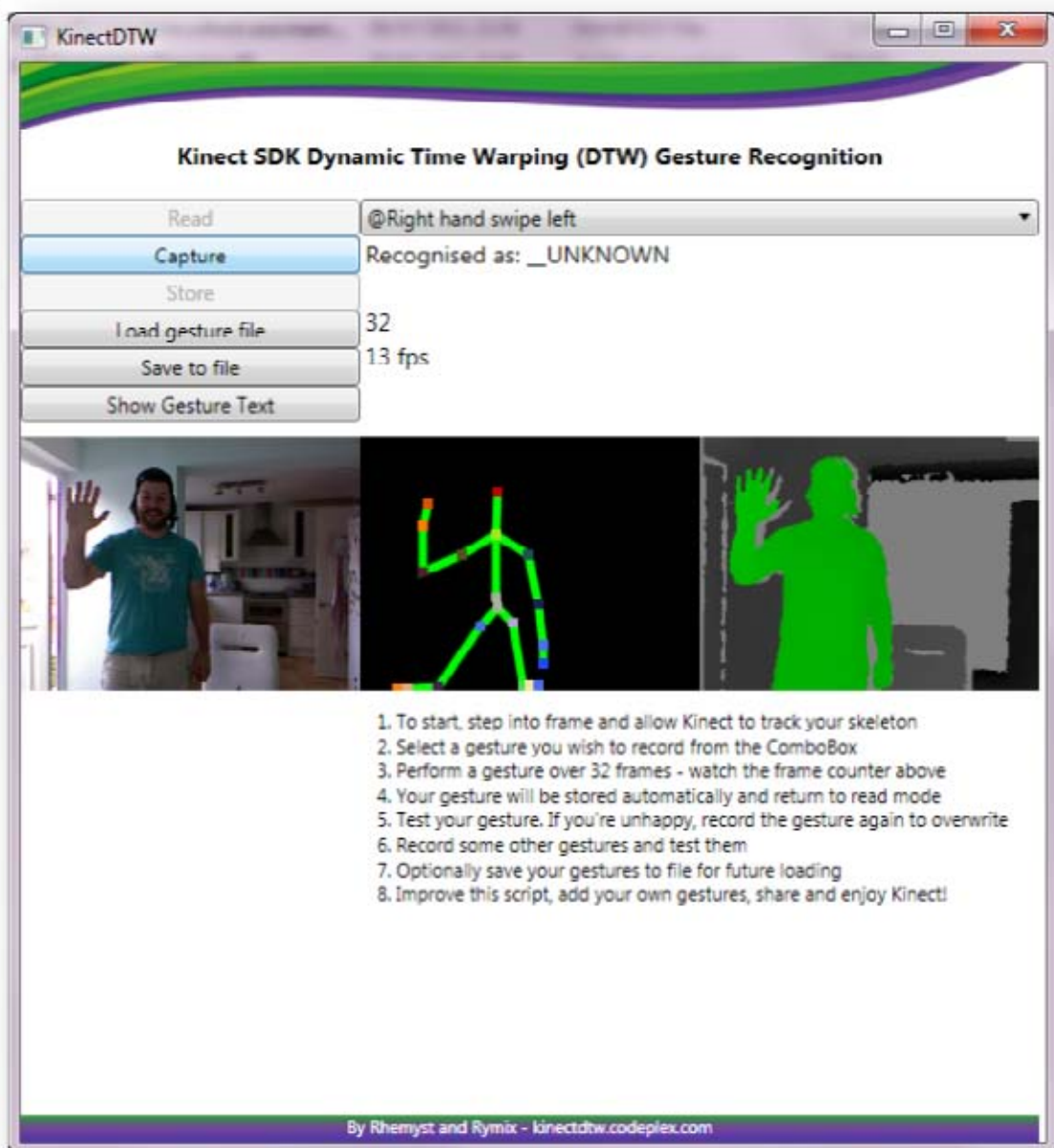


Figura 4.24. Interfaccia di KinectDTW

Vedremo nel Capitolo 6, come usare la Kinect per effettuare sperimentazioni e protocolli di riabilitazione in Realtà Virtuale, usando un software (gratuito per scopi didattici e di ricerca) capace di integrare il riconoscimento di pose e la dinamica di un avatar. Questo permetterà un corposo uso della Kinect anche per misurazioni comportamentali, pure da parte di chi non ha competenze matematiche e informatiche avanzate in merito.

4.3 Raccolta dati comportamentali in contesti ecologici

Nei paragrafi precedenti abbiamo visto come raccogliere dati comportamentali in un contesto di laboratorio, usando sia periferiche in dotazione a qualsiasi normale computer, sia strumenti più avanzati, seppur disponibili a basso costo.

In quest'ultimo paragrafo concludiamo la trattazione dei dati comportamentali esaminando la loro possibile raccolta in contesti ecologici, esplorando dunque la possibilità di effettuare ricerche sul campo in modo longitudinale.

Fino ad alcuni anni fa, effettuare una raccolta sul campo era assai complesso e ci si doveva accontentare di usare dei diari, su cui il partecipante annotava quanto richiesto dalla sperimentazione, in pratica risposte a questionari con domande aperte e/o chiuse, che poi lo sperimentatore doveva codificare per la successiva analisi qualitativa e/o quantitativa. La raccolta di dati era in passato (ovvero fino a meno di dieci anni fa) dunque assai complessa e gli studi che raccoglievano sul campo altri dati erano davvero limitati,

con pioneristici lavori in cui veniva effettuata la raccolta di campioni di saliva (con salivette o similari) e uso di holter clinici che memorizzano dati cardiaci in unità di memoria locale.

Il panorama oggi è tuttavia molto diverso, grazie infatti allo sviluppo e all'enorme diffusione di smartphone la ricerca sul campo sta diventando da un lato sempre più semplice ed efficace, dall'altro consente la raccolta di ulteriori interessantissimi dati comportamentali e d'altro tipo (psicofisiologici, relazionali, sociali, ...). La raccolta di questionari sul campo, anche allo scopo di studiare risposte relative a specifiche situazioni (ad esempio questionari durante lo shopping), può essere effettuata usando lo strumento LimeSurvey (visto nel Capitolo 1), che può essere visualizzato da un qualunque smartphone connesso a Internet. Comunque le opportunità di misurazione del comportamento sul campo, mediante l'uso di nuove tecnologie è molto più vasta che le semplici risposte a un questionario, che comunque restano di un'importanza enorme e vanno considerate complementari rispetto ad altre misure e quasi mai sostituite.

Le misure più immediate che un cellulare può fornire sono quelle relative alla posizione geografica del partecipante: infatti la maggior parte dei moderni smartphone ha accesso alle coordinate GPS (Global Positioning System, ovvero Sistema di Posizionamento Globale) e relativo orario, ricevute dall'interfacciamento con il sistema di posizionamento e navigazione satellitare civile, che include una vasta rete di satelliti in orbita intorno alla terra. L'accuratezza del sistema GPS è

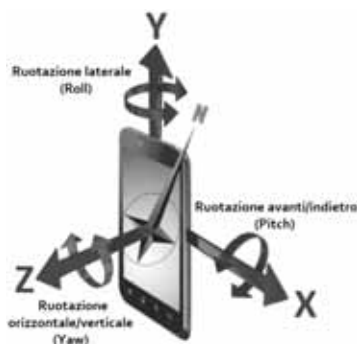
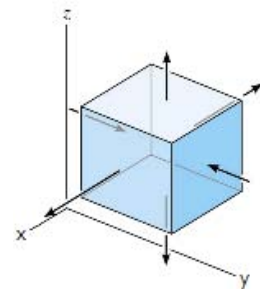
nell'ordine dei metri. L'accesso al sistema è garantito a chiunque sia dotato di un apposito ricevitore, come vedevamo prima, oggigiorno presente in quasi tutti gli smartphone.

Le misure relative al posizionamento globale possono essere assai importanti in psicologia, in quanto forniscono sia un dato quantitativo abbastanza preciso rispetto a posizionamento e percorso effettuato dal partecipante, sia in quanto possono dare un'idea del contesto entro cui il partecipante opera (casa, lavoro, vacanza, ...) e dei relativi passaggi deducibili dal percorso (ad esempio da casa al lavoro e viceversa).

Espandendo il discorso relativo alle tecnologie che possiamo trovare all'interno di uno smartphone possiamo categorizzare come segue:

- *Sensori di movimento*

Questi sensori misurano le forze di accelerazione e le forze di rotazione lungo i tre assi cartesiani (x,y e z, ovvero gli assi che identificano lo spazio entro cui può essere mosso il device e quindi, indirettamente, il partecipante che "indossa" lo smartphone. Questa categoria di



sensori include accelerometri, sensori di gravità, giroscopi e sensori dei vettori di rotazione. Quasi tutte queste tipologie di sensori sono incluse nei normali smartphone oggi a disposizione.

- *Sensori ambientali*

Questi sensori misurano vari parametri ambientali, come temperatura e pressione atmosferica, illuminazione e umidità. La categoria include dunque barometri, fotometri e termometri. Anche questi sensori si trovano negli smartphone.

- *Sensori di posizione*

Questi sensori misurano la posizione fisica del device. Questa categoria include sensori di orientamento e magnetometri.

Gli smartphone contemporanei contengono dunque una vasta serie di sensori che possono raccogliere un ampio numero di dati che possono essere elaborati per la misura di comportamenti generici o specifici.

L'unico problema a questo punto resta l'accesso a tali dati e la loro elaborazione. Di come gestire e processare questi dati ci occuperemo nella seconda parte di questo volume. Sul come raccogliere questi dati, invece, ce ne occupiamo subito.

In generale, come per un computer, l'accesso ai dati di uno smartphone è possibile tramite apposito software. Una delle caratteristiche che ha reso gli smartphone tanto popolari è proprio la disponibilità di software per ogni esigenza. Questa tipologia di software, negli smartphone, è comunemente chiamata App (da Application, ovvero applicazione).

Le App, infatti, non sono altro che dei software creati e ottimizzati per essere usati nei moderni smartphone, usandone le peculiari caratteristiche di mobilità, pervasività e connettività con la rete internet e con altri device e sensori.

Esistono molte App per le più svariate operazioni e molte di queste possono essere utilissime per psicologi intraprendenti e ricercatori di ogni tipo. Possono servirci per raccogliere dati, ma anche per effettuare semplici operazioni, come prendere appunti vocali o fare dei rapidi schemi.

Una caratteristica molto importante delle App è comunque l'interfacciamento con il device su cui sono installate. Le App accedono, dietro autorizzazione dell'utente, ai sensori dello smartphone e ne fanno sia usi più tradizionali, sia usi molto fantasiosi. Invito il lettore possessore di uno smartphone a fare molto di più che limitarsi a cercare App, è necessario infatti pensare in termini di uso di un'App quando si fa qualcosa nel proprio quotidiano e non quando si è sullo store designato al loro acquisto. Per cui se sono un neuropsicologo e penso che usare la tecnologia sia usare il cronometro dello smartphone, forse sarà il caso che rifletti sul mondo delle App e potrei un giorno scoprire che esiste un'App per prendere più tempi contemporaneamente e gestire in modo molto semplice i tempi intermedi. Bisogna considerare anche che la maggior parte delle App sono gratuite o disponibili a prezzi assai contenuti (sotto i quattro Euro). Se siete dei ricercatori, molti sviluppatori ve le daranno gratuitamente, basta chiedere. Non credereste a quante me ne hanno regalate.

Visto l'interesse, di cui prima, per l'accesso ai dati raccolti dallo smartphone per l'analisi del comportamento, vedremo adesso come usare le App per ottimizzare questo processo. In generale esistono molte App, gratuite o sotto i 5 Euro, per avere accesso ai dati interni dello smartphone e dei relativi sensori. I sistemi operativi più in uso dagli smartphone sono attualmente iOS (per i device Apple) e Android (per molte altre marche di device) e per entrambi questi sistemi operativi esistono molte applicazioni di accesso ai dati e non è certamente possibile nel breve spazio di questo volume passarle in rassegna, dunque do solo qualche indicazione e un profondo invito a provarne alcune, magari tra quelle gratuite. In particolare per iOS mi sembra interessante, come App gratuita, Data Collection Free (<https://itunes.apple.com/it/app/data-collection-free/id485523535?l=en&mt=8>) (Figura 4.25).

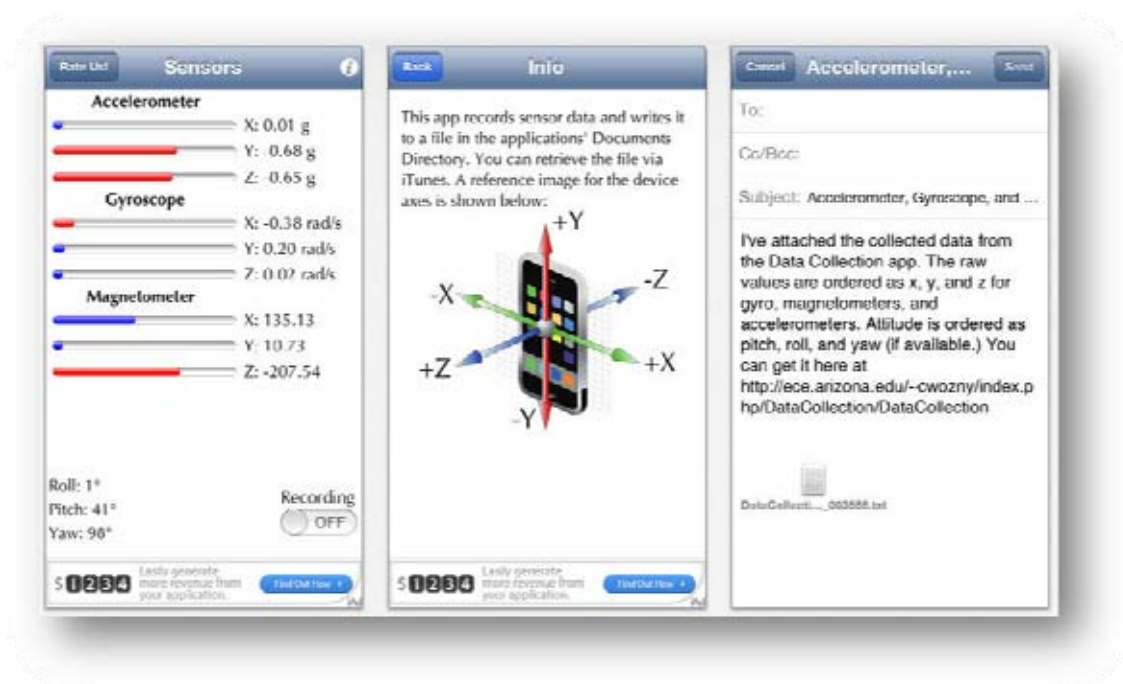


Figura 4.25. Schermate della App Data Collection Free.

Per ricercatori più esigenti segnale Sensor Data, un'App dal basso costo (Euro 7.99) <https://itunes.apple.com/it/app/sensor-data/id397619802?l=en&mt=8>, che oltre a raccogliere e archiviare in appositi file i dati raccolti (pronti per il download su PC), consente due interessanti opzioni che potrebbero essere assai utili: la scelta della frequenza di campionamento da 1Hz a 100 Hz (vedi Capitolo 2) e la possibilità di mandare dati in streaming (Figura 4.26).

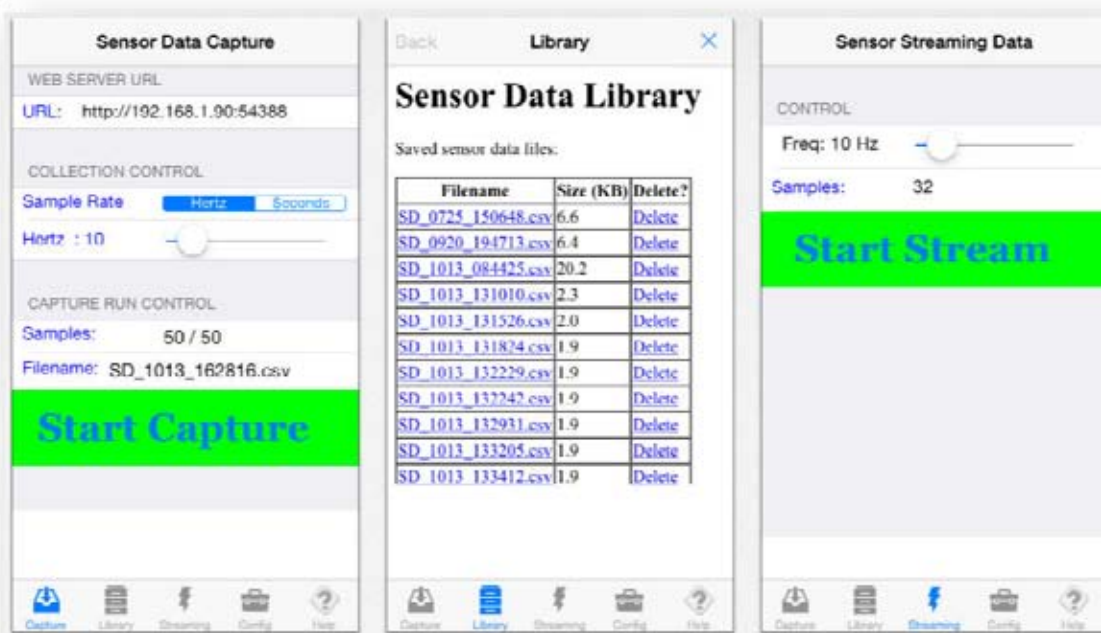


Figura 4.26. Schermate e opzioni della App Sensor Data.

Tutte le App gratuite che ho verificato per la raccolta dati mi hanno sempre dato qualche limitazione (10 secondi di registrazione Data Collection Free) o mi hanno proposto un pagamento o erano

un po' complesse nel trasferire i dati (con protocolli di trasmissione in streaming tramite una porta appositamente creata, che potrebbe essere un po' difficile per chi non ha dimestichezza). Per gli utenti del mondo Android suggerisco di usare Purple Robot (https://play.google.com/store/apps/details?id=edu.northwestern.cb.its.purple_robot_manager) (Figura 4.27).

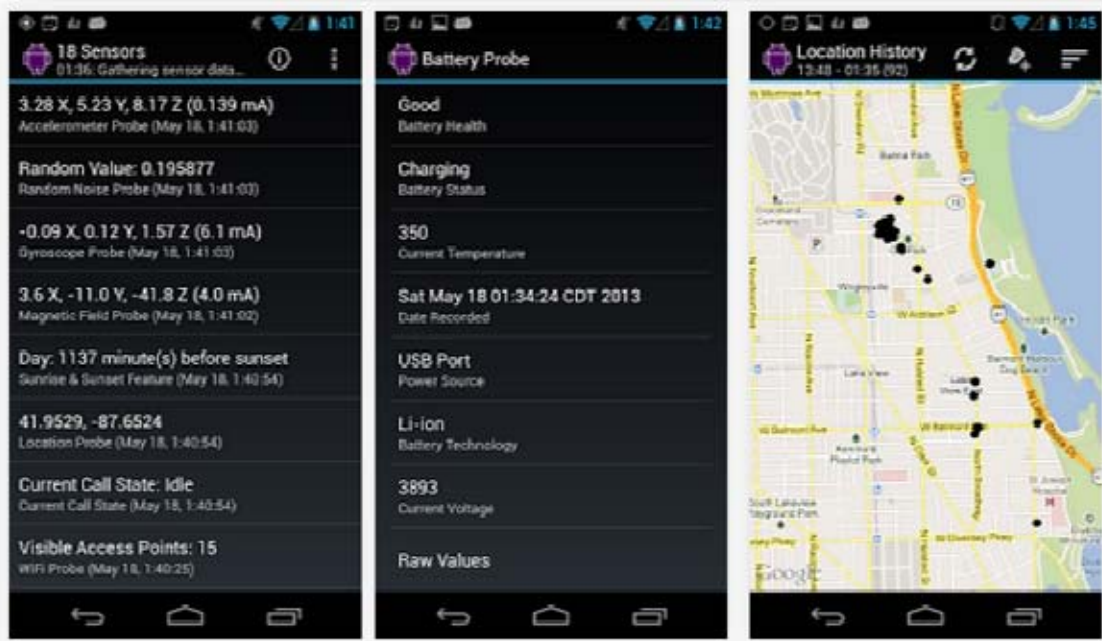


Figura 4.27. Schermate e opzioni della App Purple Robot.

Questa interessante App oltre alla classica raccolta dati propone una vera e propria creazione di interventi ed esperienze bastate sul comportamento sulla base del contesto (Purple Robot is a sensing and scripting application that enables the creation of context-aware

behavioral interventions and experiences). La App permette di essere la base per più complessi e strutturati progetti di ricerca esponendo delle HTTP API apposite per interagire con altri software e altre App, attivando oltretutto anche una serie di trigger appositamente configurabili.

Questo progetto della Northwestern University (<http://tech.cbits.northwestern.edu/purple-robot/>) permette dunque di lavorare sullo sviluppo per lo sviluppo. Tuttavia, la App può essere usata anche solo per registrare i dati (Figura 4.28).

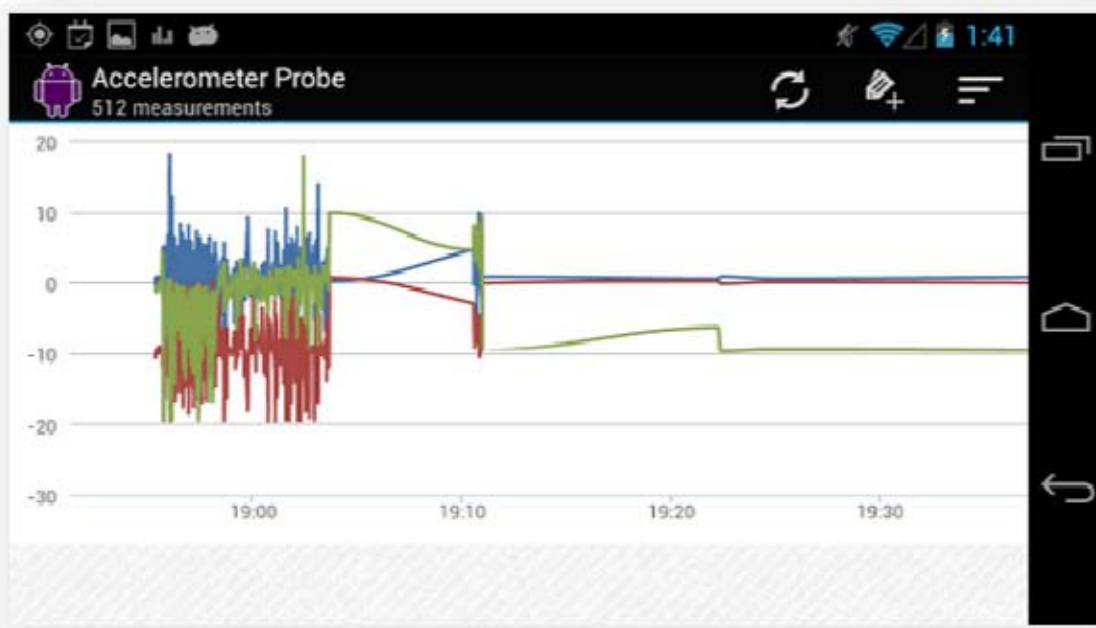


Figura 4.28. Registrazione e visualizzazione dati con la App Purple Robot.

In conclusione, possiamo affermare che l'uso delle App per raccogliere dati comportamentali in contesti ecologici, non solo è possibile, ma è anche sia alla portata di chiunque possieda uno smartphone, sia fattibile con semplici applicazioni dal costo assai contenuto o gratuitamente disponibili.

Infine, è d'obbligo sottolineare che così come esistono molte App per la raccolta di dati dai sensori, esistono anche molte App che possono essere usate per indagini qualitative. Quanto prima visto per la realizzazione di audio e video può esser fatto anche con delle App per l'editing. Inoltre esistono molte App che consentono un'efficace raccolta e organizzazione di audio, video e appunti.

Dati relazionali

Nei capitoli precedenti abbiamo visto come raccogliere dati di ogni tipologia, dal classico questionario al dato psicofisiologico o comportamentale. Non ci siamo tuttavia ancora occupati, se non marginalmente, di come poter raccogliere dei dati relativi a due o più individui, ovvero la raccolta di dati relazionali.

Innanzitutto va sottolineato che, essendo questo un volume di psicomatria computazionale, ci si concentrerà sui dati seppur in un quadro retorico metodologico, ma tuttavia senza entrare nelle complesse dinamiche della psicologia sociale e di relazione, che ovviamente esulano dalla trattazione qui affrontata. Detto ciò va comunque sottolineato che tratteremo, seppur in modo distaccato, dati relativi alle relazioni: familiari, coniugali, romantiche, amicali e più in generale sociali. La struttura che useremo per la definizione e la misura di tali relazioni è la rete. Avremo dunque reti sociali, reti amicali, reti familiari e così via. In particolare in questo capitolo approfondiremo gli strumenti a disposizione dello

psicologo per "costruire" la rete sociale a partire da dati empirici (raccolti tra individui). Sarà invece nella terza parte del volume che tratteremo sia la prospettiva teorica della complessità per la modellazione di relazioni e reti sociali, sia i modelli di simulazione per studiare le proprietà emergenti a partire dai rapporti tra singoli individui. In Tabella 5.1 diamo una prima panoramica della prospettiva dei sistemi complessi.

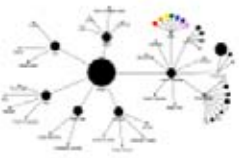

Numero di individui	Approccio	Strumenti e teorie
2 individui ■ ... ■	Teorico - matematico (Sistemi complessi)	Entropia Informazione Comunicazione } Analisi e Interazione
Molti individui reali (reti sociali) 	Teorico - matematico (Sistemi complessi) e Empirico (<i>in vivo</i>)	Modelli di Reti casuali e Modelli strategici ----- Strutture di rete ----- Raccolta dati empirici
Molti individui artificiali (ALife) 	Teorico - matematico (Sistemi complessi) e Basato su simulazioni (<i>in silico</i>)	Modelli Agent-based + Intelligenza artificiale } Processi di interazione, diffusione e trasmissione

Tabella 5.1. Modelli e strumenti delle relazioni nella prospettiva dei sistemi complessi.

Questa distinzione sarà più chiara nella terza parte del volume, ma è qui utile per farci capire il contesto entro cui operiamo.

Le domande a cui pertanto rispondiamo in questo capitolo sono relative alle modalità di raccolta dati nel caso in cui abbiamo molti individui reali (reti sociali, in Tabella 5.1) legati tra di loro in qualche modo.

Proprio dal legame parte la nostra raccolta di dati in reti sociali, infatti il legame è ciò che caratterizza la rete rispetto ad altre strutture. Lo studio di un gruppo di individui è cosa ben diversa a seconda che ci siano dei dati relativi ai loro legami (di qualunque tipo essi siano) oppure no. Nel secondo caso lo studio non è meno nobile, ma semplicemente diverso, l'oggetto di indagine potrebbe ad esempio essere il comportamento di un singolo individuo in un contesto di gruppo. Per questa tipologia di studi non siamo interessati al legame e useremo una o più tecniche di misura tra quelle viste nei capitoli precedenti.

Definito il contesto entro cui operiamo indagheremo i dati relativi ai legami e in particolare le modalità di raccolta dei dati relazionali. L'analisi relativa a questi dati si chiama Social Network Analysis (SNA) e trova vastissimo impiego nella psicologia e in molte altre discipline.

Come già detto, il legame costituisce l'elemento fondante della rete e spesso è anche l'elemento che caratterizza lo studio oggetto d'indagine. In una rete gli elementi possono essere connessi arbitrariamente, non vincolati dalla loro posizione spaziale. Perciò possiamo definire relazioni logiche e sociali tra elementi. Presentiamo di seguito, brevemente, un esempio illuminante circa l'utilità della network analysis, fornito da Padgett and Ansell.

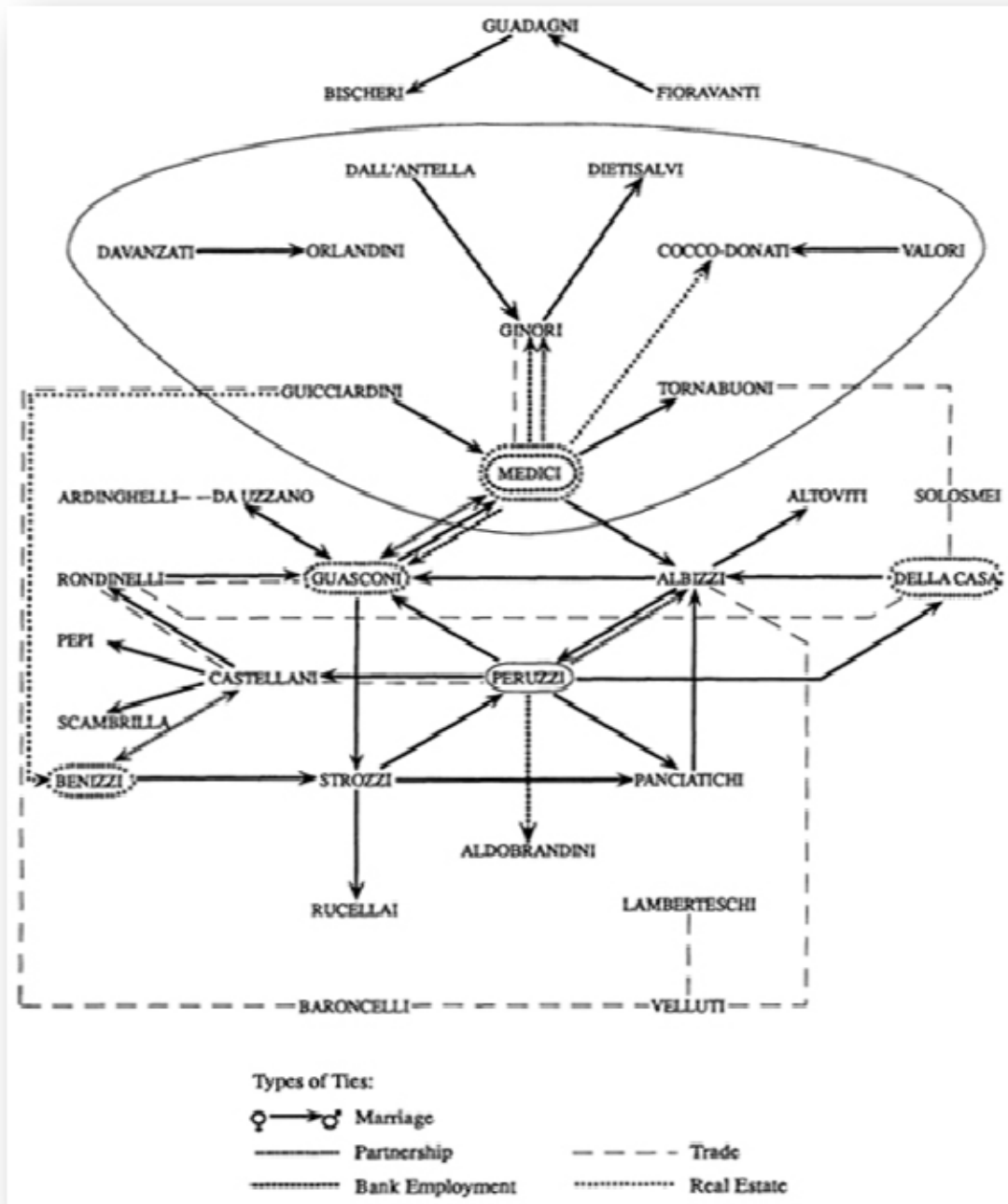


Figura 5.1. Struttura delle relazioni di matrimonio ed economiche. Già solo osservando la figura possiamo provare a capire il ruolo delle reti sociali nell'ascesa dei Medici (spesso chiamati "i padrini del Rinascimento") nella Firenze tra il 1400 e il 1434.

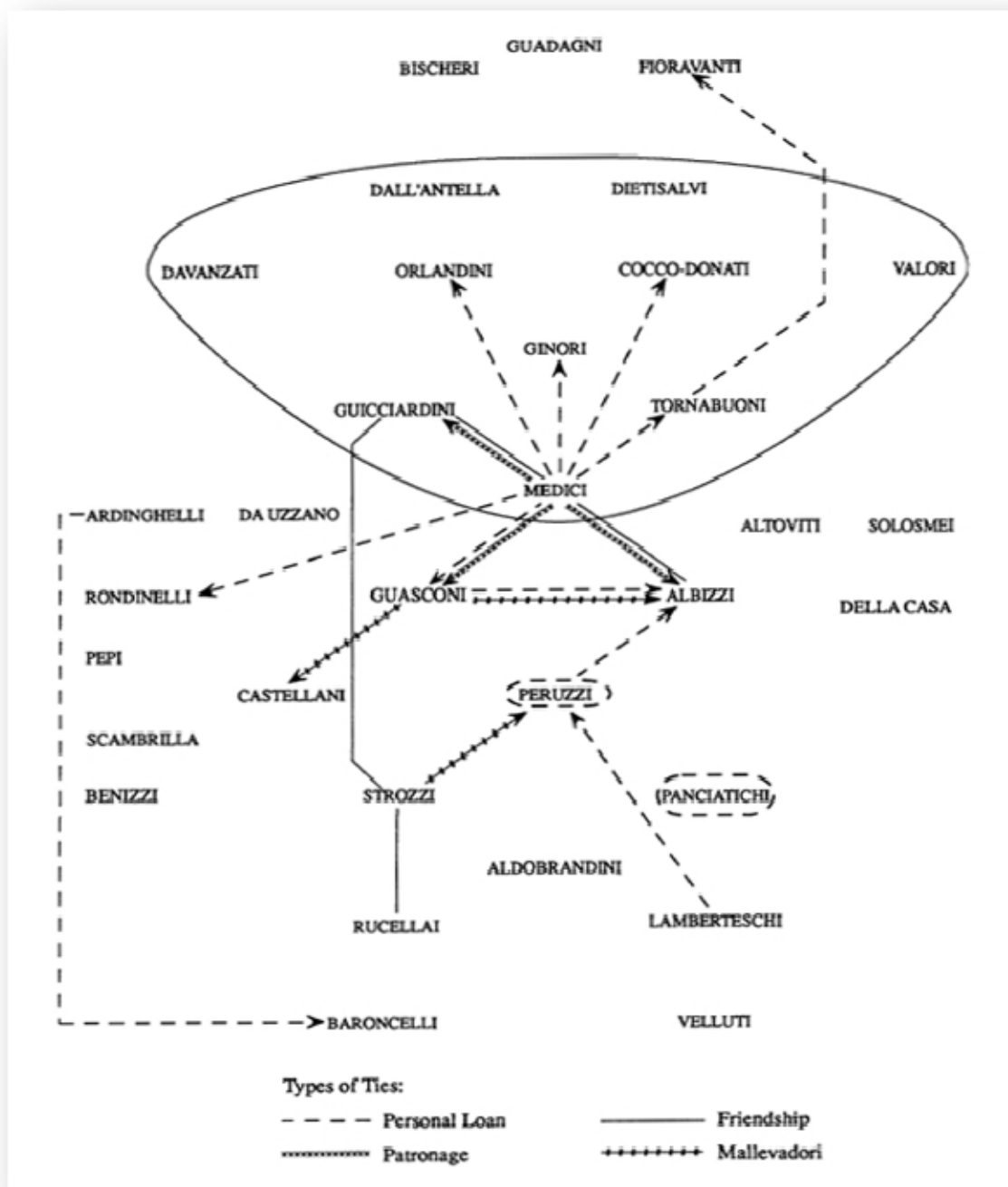


Figura 5.2. Struttura delle relazioni politiche e di amicizia. Grazie a Cosimo de' Medici la famiglia ha consolidato il potere economico e politico.

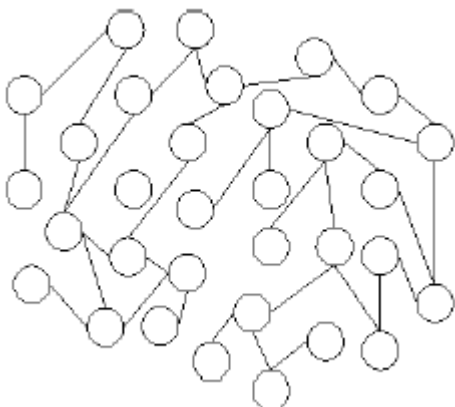
Limitiamo le nostre considerazioni, per il momento, presentando solo graficamente la struttura delle relazioni di matrimonio ed economiche e la struttura delle relazioni politiche e di amicizia nella Firenze rinascimentale.

Nei prossimi capitoli svilupperemo un'ampia serie di strumenti e metodi analitici e computazionali per un'analisi più approfondita.

La letteratura scientifica sullo studio delle connessioni e dei comportamenti in reti complesse è molto vasta e come detto attraverso discipline quali la psicologia, la matematica (teoria dei grafi), la statistica fisica, l'economia e molte altre.

Da un punto di vista teorico è molto utile formalizzare tali reti per ottenere informazioni e anche per la modellazione e la teorizzazione di nuovi paradigmi.

Da un punto di vista empirico le strutture di rete consentono l'analisi di molte componenti sulla base dell'osservazione del fenomeno oggetto di studio.



In una rete possono essere considerati molti soggetti interagenti ed estendere in tal modo l'analisi rispetto a modelli che considerano solo una relazione, come succedeva nelle prime teorie della comunicazione dove si aveva solo un emittente e un ricevente.

La rappresentazione formale di una rete complessa avviene generalmente tramite matrici che vengono poi analizzate mediante l'uso di software specifico per l'elaborazione di indici sintetici.

La teoria delle reti si sviluppa largamente mediante il ricorso a due differenti prospettive (come già indicato in Tabella 1):

- 1) "random-graph models" (modelli di reti casuali) secondo la quale esiste un processo stocastico che governa lo sviluppo dei collegamenti in una rete;
- 2) "strategic models of network formation" (modelli strategici), nella quale lo sviluppo dei collegamenti è basato su costi e benefici tramite l'uso di tecniche di teoria dei giochi.

L'interesse nello studio delle reti è dovuto al fatto che la loro struttura è un'importante determinante di come il rapporto rappresentato funziona (Figura 5.3).

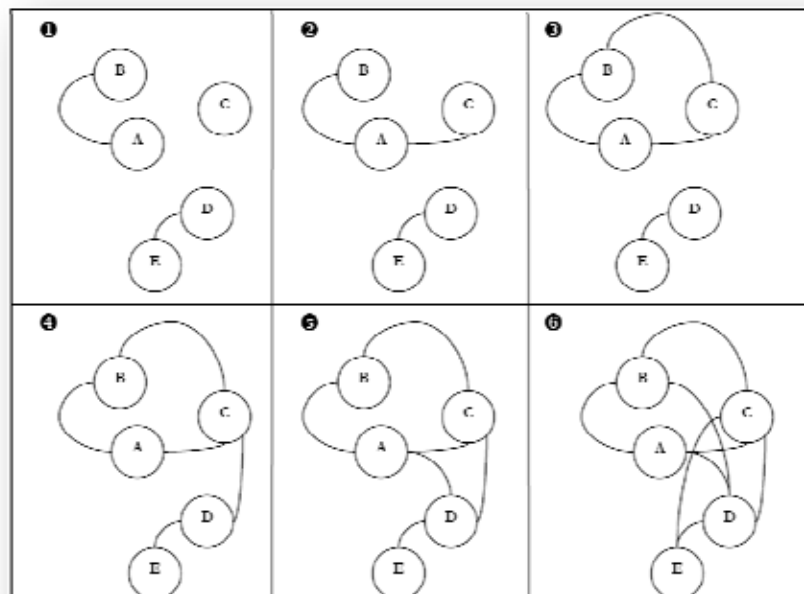


Figura 5.3. Tipologie di struttura diverse rappresentano diverse modalità di interazione tramite i legami.

Nello studio del comportamento tramite analisi delle reti complesse emergono due principali questioni:

- 1) in che modo le strutture di reti complesse influenzano il comportamento;
- 2) quali strutture di rete hanno maggiori probabilità di emergere, dato un certo tipo di comportamento.

È possibile effettuare molte analisi sulle reti complesse ed evidenziare dei ruoli chiave, come ad esempio l'analisi dei soggetti che occupano una posizione di collegamento, come si vede in Figura 5.4.

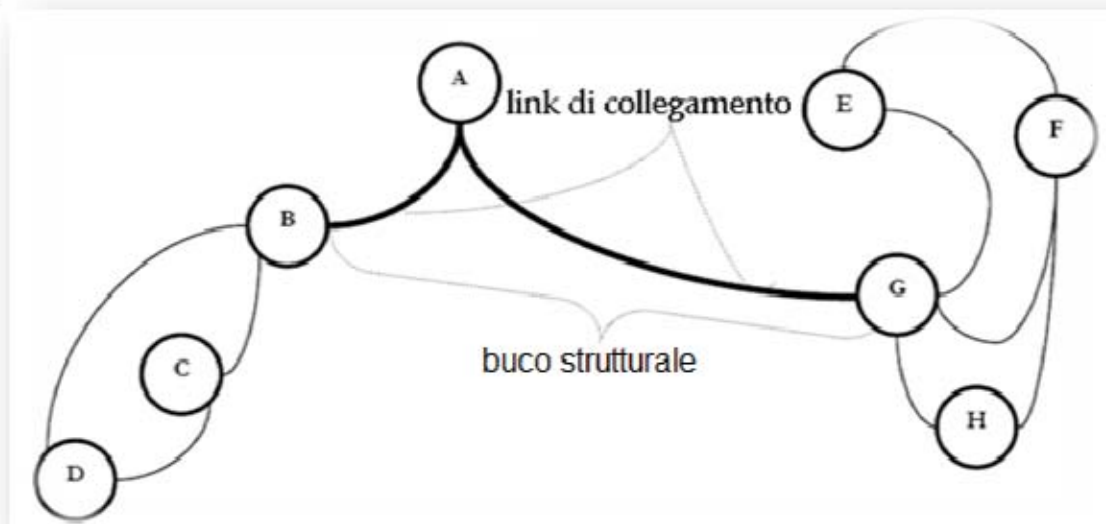


Figura 5.4. Una rete dove l'individuo A occupa una posizione di collegamento, coprendo un potenziale buco strutturale. Nel linguaggio della Social Network Analysis l'individuo A è detto broker.

Oltre l'analisi teorica ed empirica della struttura di una rete complessa, è anche possibile strutturare delle reti rappresentanti determinati comportamenti e studiarne la diffusione, elemento molto importante in psicologia.

La rete potrebbe in tal modo rappresentare e teorizzare processi di diffusione dei comportamenti, potendo – peraltro – rappresentare agevolmente la diffusione tramite comunità internet, social networks, telefono, e-mail, media, giornali e così via. E' quanto fatto da Nicholas A. Christakis che ha studiato importanti fenomeni legati al comportamento su reti sociali. Ad esempio Christakis ha analizzato empiricamente in uno studio longitudinale (35 anni dal 1971 al 2003) una rete di oltre dodici mila Obesi, dimostrando che la probabilità di diventare obeso è di circa il 57% maggiore se si ha un amico che è diventato obeso in un dato intervallo.

Dunque le reti possono essere utili per capire se si è in un contesto che favorisce un dato comportamento o, in generale, per approfondire la dinamica di certi comportamenti.

5.1 Struttura dei dati relazionali

Come abbiamo già visto in alcuni esempi, moltissimi fenomeni possono essere rappresentati in rete. Tuttavia, il problema spesso è relativo a come strutturare i dati o come estrarli da informazioni già in qualche modo strutturate computazionalmente.

Da un punto di vista strutturale il dato di relazione è facilmente implementabile in forma di matrice quadrata, ovvero una matrice con un numero di righe uguale al numero di colonne.

Se, ad esempio, voglio strutturare la rete di tre individui che si mandano email avrò una matrice con tre righe e tre colonne (3x3).

	Pietro	Silvia	Federica
Pietro	0	7	3
Silvia	3	0	1
Federica	1	6	0

Figura 5.5. La matrice quadrata di ordine tre che rappresenta la rete di comunicazione via email composta da tre individui.

La matrice in Figura 5.5 descrive una rete molto complessa. Innanzitutto è una rete diretta (anche detta orientata), ovvero la direzione dei legami conta: infatti possiamo ad esempio notare che Silvia ha inviato 3 mail a Pietro, ma ne ha ricevute da Pietro 7 (o se preferite Pietro ne ha inviate 7 a Silvia). In una rete non diretta il legame Pietro-Silvia o Silvia-Pietro è identico e questo, necessariamente, si riflette in una matrice simmetrica rispetto alla diagonale principale (quella con tutti zeri in Figura 5.5). Un altro aspetto che rende questa rete tanto complessa è il fatto che è una rete pesata, ovvero ogni legame ha un peso, dato dal numero di email inviate. In una rete non pesata un legame o esiste o non esiste, invece in una rete pesata il legame può esistere a pesi differenti.

Questa struttura dei dati a matrice può sembrare meno informativa rispetto a un grafico, ma non è così, soprattutto per reti dirette e pesate dove il grafico ha molti elementi da visualizzare (direzione dei legami tramite frecce e spessore della linea per i pesi).

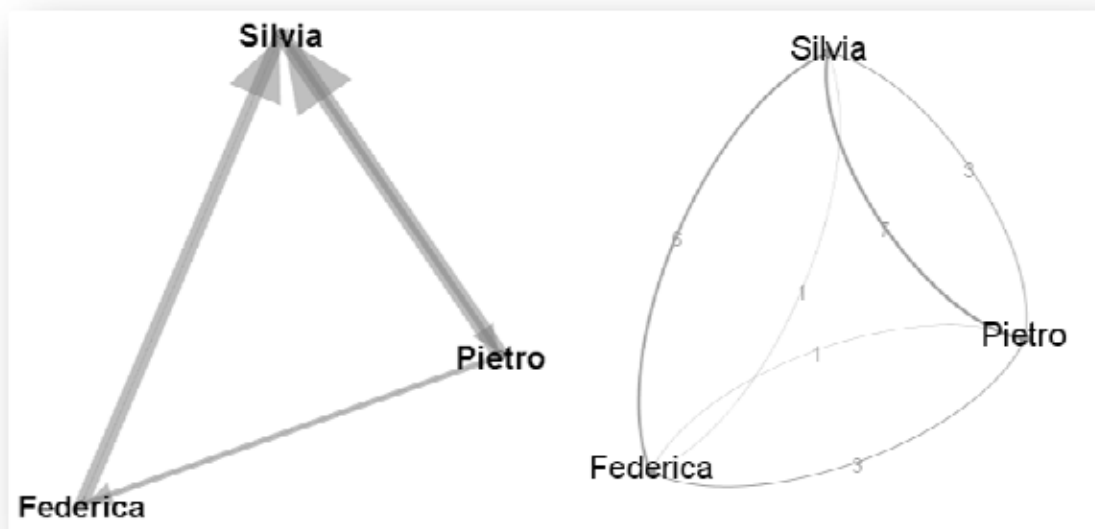


Figura 5.6. Due diversi grafici della stessa rete di cui sopra.

In realtà sia la matrice sia il grafico hanno vantaggi di visualizzazione. Se è vero che il grafico dà una più immediata visione di insieme, è anche vero che la matrice fornisce informazioni immediate, così come la matrice in Figura 5.5 ci dice che Silvia invia poche email agli altri ma ne riceve molte.

Ovviamente questo non ci dice chi è Silvia: potrebbe essere il capo dei due che invia poche email solo per dare delle direttive, ricevendo frequenti feedback. Oppure potrebbe essere una Collega

al pari degli altri due e preferisce dare risposte in forma orale alle varie email ricevute.

Questo aspetto appena discusso è assai importante, in quanto ci fa comprendere come la Social Network Analysis è uno strumento che seppur potente può generare sbagliate interpretazioni. Senza entrare troppo nel merito di queste problematiche, basti tenere presente il tipo di legame che si sta analizzando e il significato di questo legame. Se i legami riportati nella matrice di Figura 5.5, invece di essere il numero di email inviate rappresentavano la difficoltà del lavoro affidato al collega, valutato in una scala da 1 a 7, allora l'interpretazione del dato cambiava ovviamente nella sostanza, ma cambiava anche nell'affidabilità delle affermazioni fatte: se ad esempio dicevo che i colleghi percepiscono di dare un maggior carico di lavoro a Silvia, questa interpretazione è corretta. Quando gli individui in una rete, invece di essere tre sono in numero maggiore, le interpretazioni diventano più complesse e ci si affida a degli indici statistico-matematici, piuttosto che a un'ispezione visiva, sebbene questa resti sempre utile.

Per quanto attiene la raccolta di questi dati, invece, si seguono le regole di raccolta già viste nei capitoli precedenti, che possono andare dal classico questionario in cui c'è la richiesta dell'identificazione degli altri nodi e la specifica del legame (peso e direzione del legame), alla trascrizione dei nodi e dei legami da parte di uno sperimentatore che li deduce da un video, da un comportamento o da quant'altro già visto sia con metodi qualitativi, sia con metodi quantitativi.

5.2 Estrarre i dati relazionali da Facebook

I dati relazionali possono essere assai complessi anche per dimensione e accessibilità alle informazioni. Si pensi ai Social Network Sites (SNSs) come Facebook e LinkedIn o anche alla propria casella email. Se da un lato è chiaro che questi strumenti contengono molte informazioni e dati relazionali, d'altra parte è altrettanto chiaro che la creazione di una matrice, come quella vista sopra, andando a reperire manualmente tali informazioni, se è possibile non è certo facile e veloce.

Normalmente, infatti, nell'ambito della Social Network Analysis si raccolgono i dati relazionali in modo automatizzato, tramite algoritmi e/o procedure effettuate direttamente sul database che contiene tali dati. Ad esempio nel caso della email si accede al server di posta elettronica aziendale e si scaricano le informazioni di relazione contenute nelle migliaia, se non milioni, di messaggi scambiati in un'azienda. L'operazione non è certamente semplice e veloce, ma almeno è fattibile. Se devo scaricare invece i dati di una singola email o di un singolo individuo in Facebook, l'operazione richiede solo minuti o ore di elaborazione e strumenti non sofisticati (normalmente basta un normale PC). In merito all'intervento diretto su un database, ne torneremo a parlare, vedendo come fare, nel Capitolo 12. Su come estrarre le proprie informazioni di rete dal proprio profilo Facebook ne parliamo subito, in quanto è operazione abbastanza semplice per cui basta una semplice applicazione.

Esistono in effetti diverse applicazioni su Facebook per accedere ai propri dati di Facebook. Qui di seguito do le istruzioni su come usare NameGenWeb (<https://apps.facebook.com/namegenweb/>) che a mio parere è ottima sia per la estrema semplicità di utilizzo, sia perché consente di anonimizzare completamente i dati.

Dopo aver cliccato sul link e aver effettuato il login a Facebook, comparirà la finestra di dialogo di cui in Figura 5.7, che avvia il semplice processo automatizzato.



Figura 5.7. L'interfaccia iniziale di NameGenWeb che ci guida verso l'estrazione dei dati relazionali dal nostro profilo Facebook.

L'applicazione è stata sviluppata da un gruppo di ricercatori, che non chiede alcun pagamento ma, salvo consenso ("Step 1: Help us with our research"), richiede di poter scrivere via email a esclusivi fini di ricerca.

Allo step successivo ("Step 2: Selecting Attributes") ci viene chiesto a quali specifiche informazioni siamo interessati (Figura 5.8).

Step 2: Selecting Attributes

First, select which information about your network you are interested in.

An attribute is a piece of information about someone in your Facebook network. A person's gender, likes, or relationship status can all be thought of as attributes. The form below allows you to select which attributes you may download from the 252 friends available to NameGenWeb.

Default attributes are usually always available about any member of Facebook, whereas **extended attributes** may or may not be available depending on the privacy settings of your friends.

Please be aware that if you have a large number of friends, selecting more attributes may significantly increase the time it takes to download your network. If you encounter difficulties, please return to this page and try again with fewer attributes selected.

Default Attributes:	Extended Attributes:
<input type="checkbox"/> Facebook User ID	<input checked="" type="checkbox"/> Biography
<input checked="" type="checkbox"/> Full Name	<input checked="" type="checkbox"/> Hometown
<input checked="" type="checkbox"/> First Name	<input checked="" type="checkbox"/> Birthday
<input checked="" type="checkbox"/> Middle Name	<input checked="" type="checkbox"/> Political Beliefs
<input checked="" type="checkbox"/> Last Name	<input checked="" type="checkbox"/> Relationship Status
<input checked="" type="checkbox"/> Gender	<input checked="" type="checkbox"/> Religious Beliefs
<input checked="" type="checkbox"/> Profile Picture URL	<input checked="" type="checkbox"/> Likes Count
<input checked="" type="checkbox"/> Locale	<input checked="" type="checkbox"/> Friend Count
<input checked="" type="checkbox"/> Mutual Friend Count	

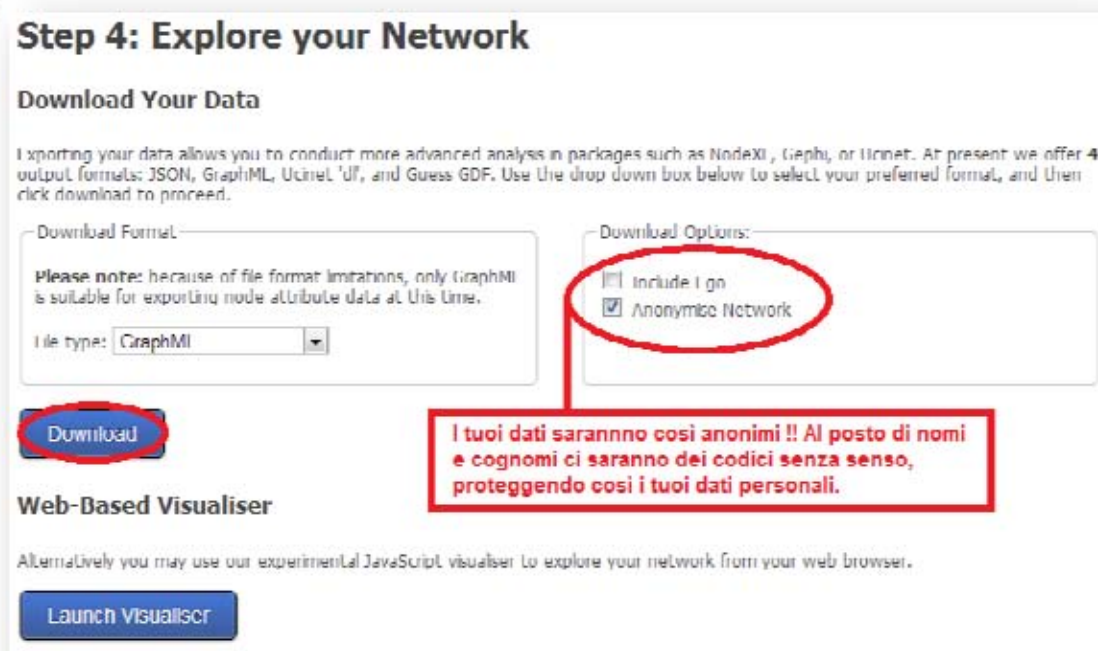
[Download](#)

Figura 5.8. Step 2: selezione degli attributi che ci interessano.

Allo Step successivo ("Step 3: Downloading ties") avviene la creazione automatica del file di dati relazionali in merito ai propri contatti Facebook.

Infine, all'ultimo Step ("Step 4: Explore your Network") abbiamo le ultime opzioni e il download. Tra le opzioni di quest'ultima fase,

come si può vedere in Figura 5.9, c'è la tipologia del file da scaricare (possiamo usare GraphML, vedremo al Capitolo 12 come farne uso), le opzioni di download: includere o meno "Ego", ovvero il nodo che rappresenta se stessi nel network e "Anonymise Network" per rendere totalmente anonima la propria rete, in cui i nomi degli amici di Facebook saranno sostituiti da una serie di codici opportunamente generati.



Step 4: Explore your Network

Download Your Data

Exporting your data allows you to conduct more advanced analysis in packages such as NodeXL, Gephi, or Ucinet. At present we offer 4 output formats: JSON, GraphML, Ucinet 'df', and Guess GDF. Use the drop down box below to select your preferred format, and then click download to proceed.

Download Format:

Please note: because of file format limitations, only GraphML is suitable for exporting node attribute data at this time.

File type: GraphML

Download Options:

- Include Ego
- Anonymise Network

Download

Web-Based Visualiser

Alternatively you may use our experimental JavaScript visualiser to explore your network from your web browser.

Launch Visualiser

I tuoi dati saranno così anonimi !! Al posto di nomi e cognomi ci saranno dei codici senza senso, proteggendo così i tuoi dati personali.

Figura 5.9. L'ultimo Step dell'Applicazione prima del download.

Una volta premuto il tasto "Download" (Figura 5.9) si potrà decidere dove archiviare il file della propria rete Facebook, per la successiva elaborazione.

Nella seconda parte del volume vedremo, come gestire e organizzare i dati relazionali e come usare un semplice software gratuito per il calcolo dei più classici indici di Social Network Analysis (closeness, degree, ecc.).

Dal reale al virtuale

Chiudiamo questa prima parte del volume parlando di Realtà Virtuale (RV), uno strumento che è computazionale per definizione, sia perché normalmente fa uso di computer o altri dispositivi di elaborazione, sia perché l'elaborazione grafica richiesta da tale strumento è strettamente legata alla computazione continua di grafica tridimensionale, rendendo le operazioni di gestione di tale processamento esse stesse fortemente computazionali.

Approfondiremo qui gli aspetti legati all'uso della RV per tutti, presentando come sempre le opportunità a costo zero e in particolare il software gratuito a disposizione per la raccolta di dati sperimentali, clinici e di ricerca.

A differenza di altri strumenti, come il questionario, la RV maggiormente si presta a essere un tool di intervento e non solo di raccolta dati. In particolare, più di recente, è stato prodotto un numero crescente di ricerche che dimostrano l'efficacia dell'uso

della RV in ambito clinico, con particolare riferimento ai settori della psicoterapia, della neuropsicologia e della neuroriabilitazione.

6.1 Le basi della Realtà Virtuale

La Realtà Virtuale si basa sull'uso di una tecnologia che consente di simulare esperienze della vita quotidiana mediante ambienti 3D interattivi generati dal computer.



Figura 6.1. Un ambiente virtuale 3D navigabile.

Combinando la simulazione di ambienti tridimensionali con dispositivi di tipo uditivo, tattile e display immersivi, come caschi

virtuali, è possibile indurre nell'utente la sensazione di interagire in un ambiente reale e quindi ricreare delle situazioni e degli esercizi che possono essere di beneficio all'azione terapeutica, condotti tuttavia nel contesto "protetto" del laboratorio del clinico. L'uso di sensori di movimento come l'head-tracker (un sensore posto sul casco per la RV per tracciare il movimento della testa), consente di registrare molti movimenti dell'utente e di utilizzare queste informazioni per aumentare la precisione della diagnosi o del monitoraggio del trattamento. Inoltre, l'head-tracker, ha anche una funzionalità pratica molto importante, ovvero rende l'ambiente che si naviga veramente immersivo. In pratica l'immagine dell'ambiente virtuale 3D proiettata nel casco di RV, cambia seguendo in modo perfettamente sincronizzato il movimento della testa. Questo significa che se sto guardando i palazzi di Figura 6.1 e alzo la testa, allora in automatico e in modo continuo verrà proiettata l'immagine congruente con il movimento, ovvero man mano che alzo la testa vedrò porzioni di cielo via via maggiori, fino a visualizzare solo il cielo quando la mia testa sarà completamente orientata verso il soffitto della stanza.

Mentre diamo per scontato questo processo nella realtà che ci circonda, altrettanto non può esser fatto nella realtà virtuale, se non per mezzo dell'head tracker. In fondo è come se il casco di realtà virtuale fosse un monitor di computer piazzato davanti ai nostri occhi: è chiaro che muovere il monitor non cambia quanto in esso contenuto ed è dunque necessario un tracciamento del movimento e un complesso software per la sua codifica.

Già da questa breve descrizione possiamo capire la complessità computazionale richiesta dalla RV per il processamento di dati in tempo reale. Cercheremo comunque in questo capitolo di concentrare l'attenzione sull'uso della RV e non sulla sua complessità computazionale, che per il momento affronteremo usando degli appositi software per consentirci, a meno di avere un buon computer, una discreta esperienza immersiva, atta a raccogliere interessanti dati per il mondo della psicologia.

Vedremo invece nel capitolo 14 del volume gli aspetti legati alla computazione dati in realtime, che consentiranno la creazione e la personalizzazione di complessi ambienti tridimensionali, sempre con software gratuito.

Inizieremo parlando di Second Life, una piattaforma lanciata nel 2003 conosciuta anche per la diffusione mediatica che ha avuto al suo avvio e negli sviluppi successivi fino all'attuale versione. Il nome evoca immediatamente l'immersione in una vita parallela nel mondo virtuale. Nell'intento dei creatori lo scopo di questa piattaforma era quello di dotare di una vita virtuale con regole e ambienti che simulassero la vita reale e anche per questo motivo hanno inserito un sistema di scambio basato sulla moneta: il Linden Dollar, convertibile in Dollari Americani (USD) reali.

Pur non essendo un vero sistema monetario, questo espediente ha permesso alla Linden (l'azienda proprietaria di Second Life) la creazione di un mercato di scambio (detto Marketplace) dove l'offerta è costituita da prodotti della Linden e da qualunque grafico o progettista 3D che voglia offrire prodotti virtuali,

monetizzando il lavoro fatto, mentre la domanda è costituita da qualunque iscritto a Second Life che voglia acquistare tali prodotti, non avendo il tempo, il modo o le competenze per realizzarsi in autonomia.



Figura 6.2. Il Marketplace di Second Life, dove è possibile comprare e vendere prodotti virtuali.

Second Life, se correttamente usato, può essere uno strumento molto utile per psicologi, terapeuti e ricercatori. Nel 2007 una lettera a Science (Science 7 December 2007: Vol. 318 no. 5856 p. 1549), dal titolo "Virtual World, Real Healing" spiegava come i

mondi virtuali possano essere utili per la gestione, in ambiente controllato, di molti disturbi, come ad esempio la fobia sociale e in particolare di come la dimensione sociale (così come esiste in Second Life) può giocare un importante ruolo per l'intervento del terapeuta come osservatore di un'esposizione controllata per lo sviluppo di contatti sociali.



Figura 6.3. Terapia virtuale: un esempio di terapia con supporto di gruppo in Second Life.

Questo è ovviamente uno dei molteplici esempi che possono essere fatti, ma rende già l'idea di come, con un corretto uso, Second Life può essere un utile strumento.

Il maggior vantaggio di Second Life rispetto ad altre piattaforme virtuali è comunque la creazione di contenuti da parte degli stessi utenti. Questo processo ha permesso una vasta creazione di oggetti, molti dei quali sono disponibili gratuitamente, così come l'accesso alla piattaforma stessa. Dunque è possibile entrare gratuitamente in Second Life e avere a disposizione una vasta gamma di oggetti tridimensionali pronti da essere usati. Con un budget sotto le cinquanta euro nel 2011 ho progettato con Silvia Serino, Cinzia Vigna, Andrea Gaggioli e Giuseppe Riva un aeroporto virtuale in Second Life, per la gestione della paura di volare.



Figura 6.4. Un aeroporto virtuale in Second Life.

In pratica l'unico problema è stato quello di trovare uno spazio dove ospitare aeroporto e aereo, i quali sono stati l'unico costo effettivo sostenuto (come dicevo prima sotto i cinquanta euro).

La personalizzazione degli spazi per mezzo di immagini incollate sopra le superfici 3D, ha richiesto ovviamente qualche giorno di lavoro e un team abituato a manipolare oggetti 3D.



Figura 6.5. Interno dell'aereo virtuale in Second Life.

Il contenuto del nostro progetto (Paura di Volare 3D) ha vinto il premio eContent Award del 2011 (<http://www.econtentaward.it>), Premio per i migliori contenuti e servizi in formato digitale, come migliore progetto dell'area eHealth. Ovviamente il merito di un progetto di successo è nell'esperienza e nelle capacità del team che se ne occupa, però questa esperienza dimostra come sia possibile

fare dei progetti di valore pur avendo a disposizione budget assai limitati.

Second Life può dunque essere un utile strumento e se usato con un po' di fantasia e flessibilità può produrre ottimi risultati.

Tuttavia nel mondo della ricerca e dell'intervento mediante l'uso di strumenti della realtà virtuale, Second Life presenta alcuni considerevoli limiti. Innanzitutto è una piattaforma pensata per il gaming e non per la terapia. Dunque può andare bene per produrre contenuti da esporre a dei pazienti, ma non è un ambiente adatto per incontrare pazienti. Per ovviare a questo problema si può comprare uno spazio virtuale e darne accesso limitato ai soli pazienti secondo le esigenze del terapeuta e possibilmente in sua presenza. Ovviamente però questo ha un costo aggiuntivo che dipende dagli spazi necessari. Un'altra problematica di Second Life è la compatibilità con device che anche quando disponibile richiede una certa competenza per la configurazione. Complessa è anche la definizione di azioni specifiche relative a certi comportamenti (triggering), che spesso richiede capacità di programmazione, seppur non avanzate. Anche importare semplici video e usare head mounted display richiede una certa complessità. Non è peraltro possibile, per il momento, importare video con trasparenza, che vedremo essere molto importante.

Infine, vista anche la limitata disponibilità di dati da elaborare (ad esempio l'assenza del logging sui percorsi effettuati con il tempo delle attività di triggering) rende Second Life uno strumento limitato per la misura in psicologia.

Più in generale, Second Life, è una piattaforma pensata per il gaming e l'entertainment, non certamente per il mondo clinico. Per quanto a volte, con le dovute precauzioni, può essere adattata a usi clinici, questa piattaforma non è certamente nata e configurata per questo scopo e va pertanto usata con estrema cautela da personale altamente qualificato e ben cosciente dei limiti e degli svantaggi sottolineati.

Per ovviare a queste limitazioni di Second Life, esistono in commercio alcuni programmi di grafica professionale che possono essere usati con grandissima flessibilità di azione, ma con altrettanta grande difficoltà di utilizzo. Parliamo infatti di programmi per grafici 3D professionisti, programmatori, ingegneri e altri tecnici, spesso in team.

Se questo processo ha senso per produrre un videogame con budget da vari milioni di euro, ha certamente molti limiti per l'uso in ambito clinico e/o di ricerca.

Viste tutte queste limitazioni, o di strumento o di difficoltà di utilizzo, presento di seguito una piattaforma molto innovativa che, oltre a essere gratuita, ha il vantaggio di essere pensata per essere usata da psicologi e più in generale da clinici e da ricercatori che non possiedano competenze di programmazione. Questa piattaforma, NeuroVR (<http://www.neurovr.org/>), già a partire dal nome, è pensata per il mondo clinico e della ricerca. E' una piattaforma totalmente gratuita ed è espandibile con i propri progetti e contenuti, utilizzando semplici comandi interattivi e assolutamente senza esigenza di programmazione.

NeuroVR è costituito da un Editor che consente di creare, importare e manipolare ambienti virtuali e un Player, per avviare gli ambienti virtuali secondo le specifiche necessarie e le periferiche a disposizione, come da Figura 6.6.

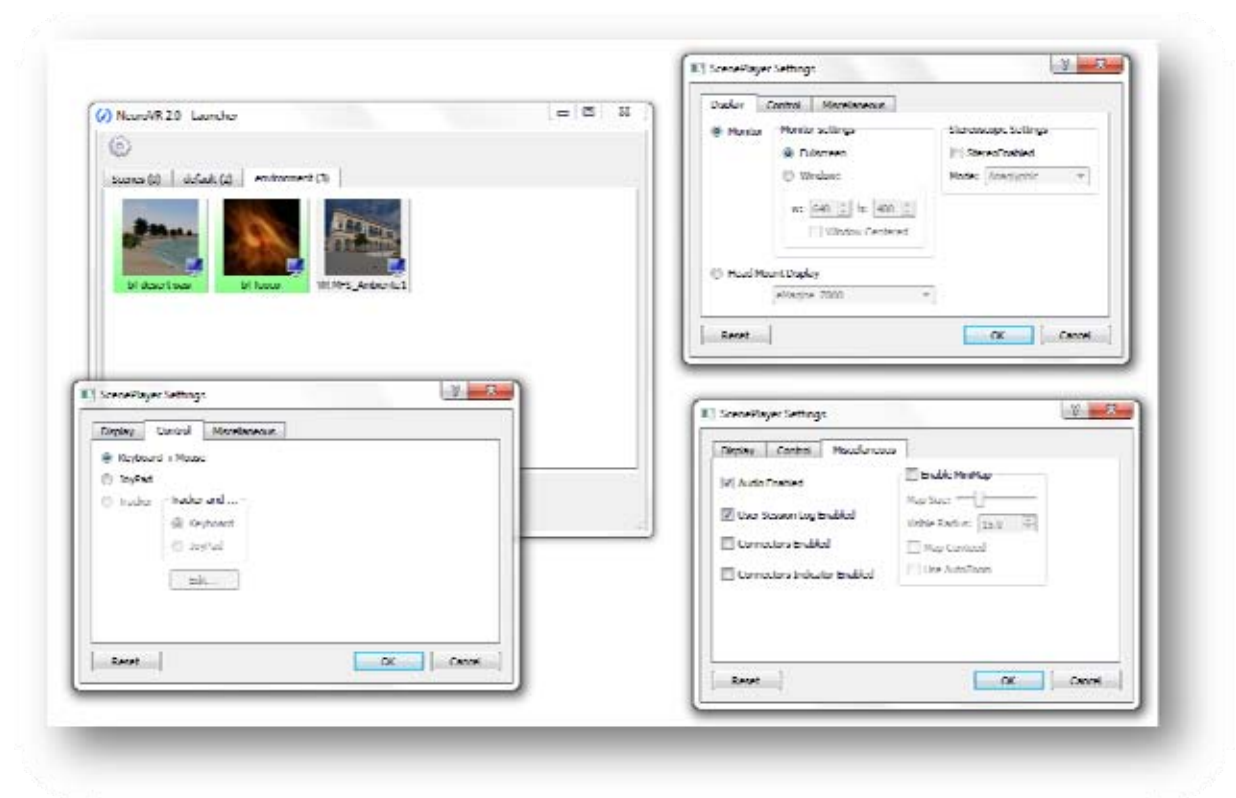



Figura 6.6. Interfaccia e settaggi del player di NeuroVR.

Tramite il Player si avviano gli ambienti e si entra in RV, secondo le indicazioni fornite dei settaggi accessibili semplicemente premendo il pulsante .

La configurabilità del Player permette la navigazione di ambienti virtuali anche senza essere in possesso di Head Mounted Display (HDM, il caschetto) o Joypad (l'unità di controllo), usando solo

tastiera e mouse. Questo implica che chiunque abbia a disposizione un computer non troppo lento e con scheda video adeguata, può usare la realtà virtuale. Un computer a configurazione minima costa da circa 500 euro in su. Per quanto attiene i contenuti, ne sono disponibili molti, totalmente gratuiti, come segue:



- *base contents pack* (ufficio, classe, appartamento, bivrs)
- *body perception pack* (scale, piscina, ristorante)
- *green nature pack* (lago, falò, montagna, parco, valle, cascata)
- *warm and sandy pack* (spiagia, deserto, gazebo, isola, onde)
- *shopping pack* (supermarket e minimarket)
- *public areas pack* (auditorium, cinema, piazza)
- *hospital and station* (ospedale e stazione, anche in italiano)

Inoltre, all'interno del programma è già disponibile una vasta libreria di oggetti virtuali di ogni tipo (vedi Figura 6.7).

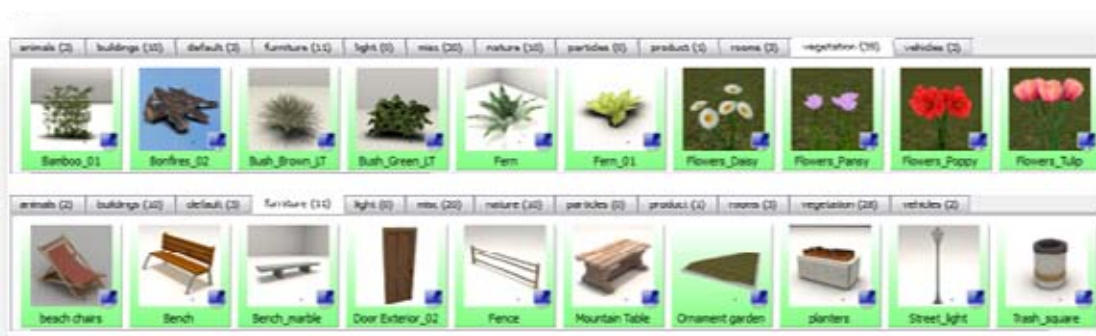


Figura 6.7. Alcuni oggetti virtuali disponibili in NeuroVR.

Sono anche disponibili, sempre gratuitamente, una serie di edifici e altri elementi naturali 3D (cielo, deserto, valle, ...), come si può vedere in Figura 6.8.

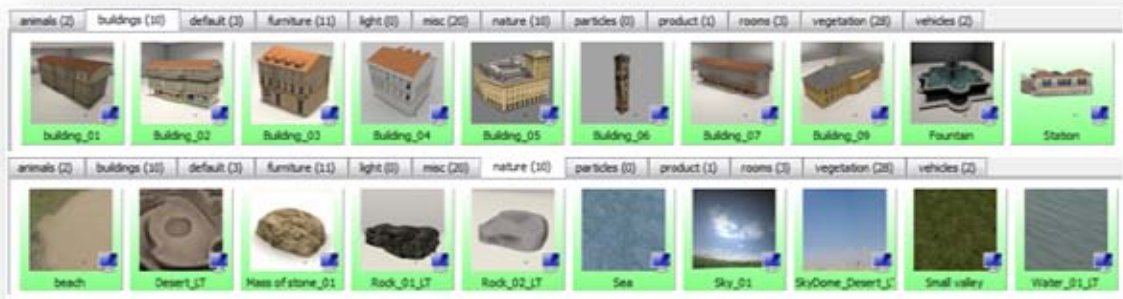


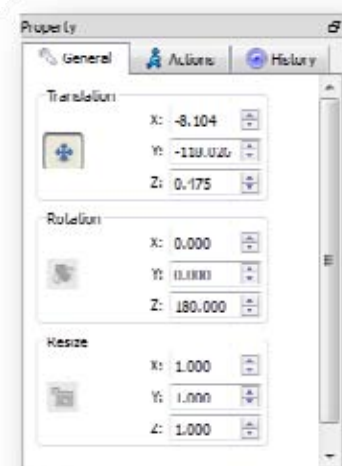
Figura 6.8. Elementi ambientali 3D.

In Neuro VR è comunque sempre possibile importare altri oggetti 3D, appositamente realizzati con altri programmi. Inoltre, è anche possibile importare immagini, suoni (anche musica) e video.

L'importazione di oggetti 3D, immagini, audio o video in NeuroVR è estremamente semplice, basta infatti prendere l'oggetto dalla libreria e trascinarlo all'interno della scena.

La posizione che occuperà l'oggetto sarà quella tracciata dal marker giallo, come si può vedere dalla Figura 6.9.

Una volta importato l'oggetto è poi possibile spostarlo (*Translation*), ruotarlo (*Ruotation*) e cambiarne le dimensioni (*Resize*) per adattarlo ulteriormente.



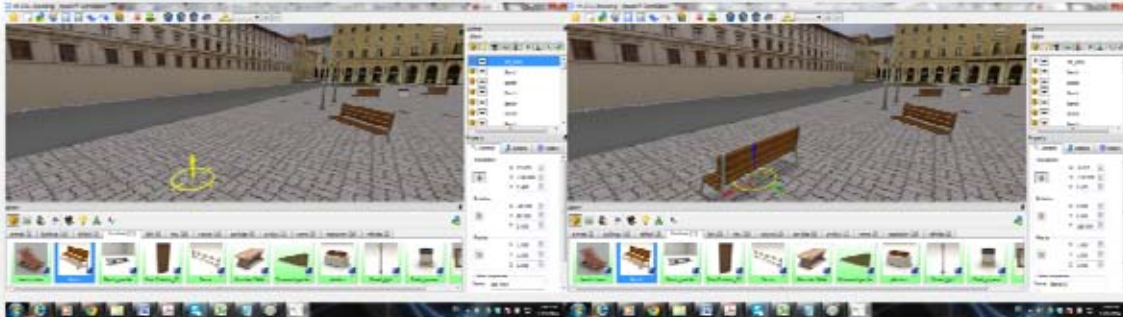


Figura 6.9. Importazione di oggetti 3D, immagini, audio o video in NeuroVR.

In particolare sui video è anche possibile realizzarne con la tecnica del chroma key per poi importarli in NeuroVR con il canale di trasparenza. Questo vuol dire che possiamo realizzare un video di una o più persone per parlano e agiscono in qualche modo con dietro uno sfondo a tinta uniforme il cui colore viene sottratto e sostituito con trasparenza (chroma key), come da Figura 6.10.



Figura 6.10. Dal video reale al mondo virtuale.

I video così realizzati ed importati in NeuroVR appariranno come sagome di personaggi reali in un ambiente virtuale. Appositamente configurati nell'ambiente appariranno come personaggi veri che interagiscono nell'ambiente virtuale.

Questa modalità è molto utile in situazioni cliniche, sia per esporre il paziente a situazioni reali, ma in un ambiente controllato, sia per inserire elementi di aiuto o realismo a supporto di un trattamento.

Infine, un altro elemento che rende NeuroVR un software adatto ad ogni scopo clinico e di ricerca, è l'immediata disponibilità di dati.

Per ogni sessione di NeuroVR vengono creati due file:

1. un file di log contenente i dati temporali (timestamp degli eventi), nome degli eventi, tipo di utente (user o system), dati di navigazione (coordinate di posizione) e dati di visualizzazione (vettore di direzione anteriore), come si può vedere in Figura 6.11.
2. un'immagine dell'ambiente virtuale preso dall'alto (mappa) con una linea rossa sovrainposta, indicante il percorso effettuato dall'utente durante la navigazione dall'inizio alla fine della sessione di realtà virtuale, come da Figura 6.12.

Time-Jump	CHANGE SCENARIO	SYSTEM	Forward Vector	Position
1390413719531	PLAYER_MOVED	USER	"Forward Vector(-1 -0.000796414 0)"	Position(83.35 -126.855 5.76)"
1390413721741	PLAYER_MOVED	USER	"Forward Vector(-0.985593 0.019981 -0.167952)"	Position(83.35 -126.855 2.20167)"
1390413722750	PLAYER_MOVED	USER	"Forward Vector(0.908102 0.0479593 0.415993)"	Position(83.35 126.855 2.20167)"
1390413722753	PLAYER_MOVED	USER	"Forward Vector(-0.772338 0.065482 -0.631828)"	Position(83.35 -126.855 2.20167)"
1390413723257	PLAYER_MOVED	USER	"Forward Vector(-0.69197 0.077178 -0.718597)"	Position(83.35 -126.855 2.20167)"
1390413723761	PLAYER_MOVED	USER	"Forward Vector(0.780624 0.105215 0.616081)"	Position(83.35 126.855 2.20167)"
1390413724264	PLAYER_MOVED	USER	"Forward Vector(-0.986391 0.163866 0.0134695)"	Position(83.35 -126.855 2.20167)"

Figura 6.11. Logging di NeuroVR.

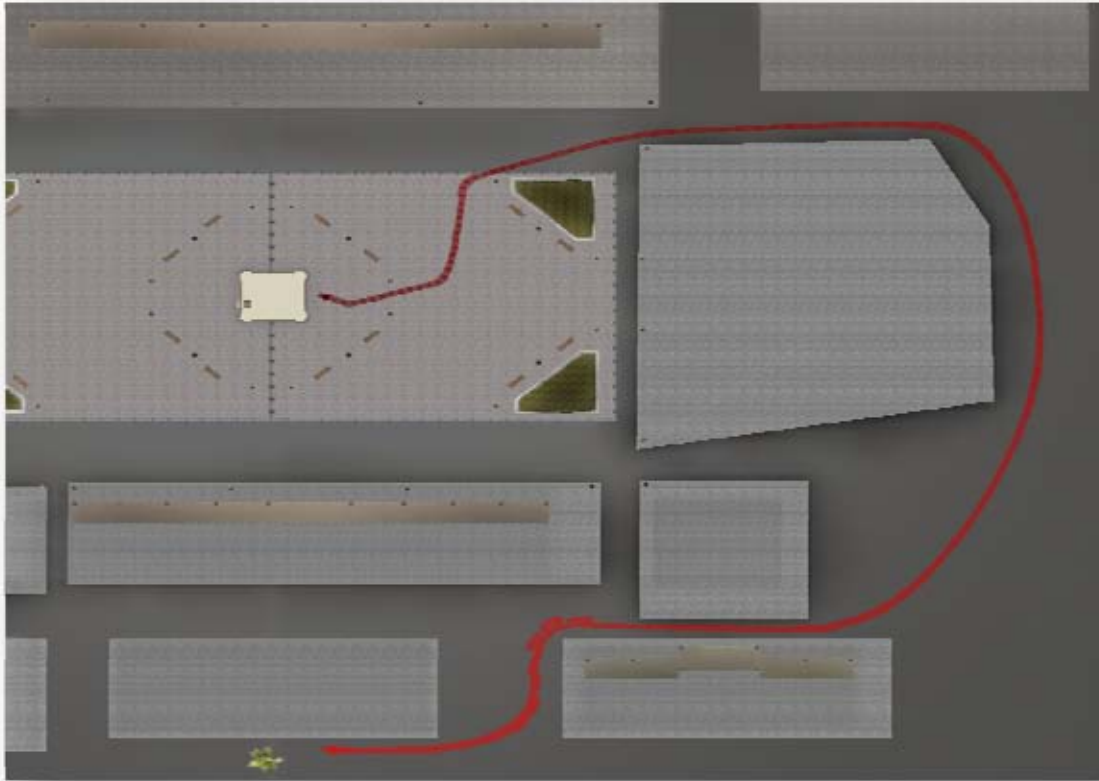


Figura 6.12. Percorso, visto dall'alto, effettuato dall'utente durante la navigazione nell'ambiente virtuale.

La figura naturalmente si presta a una prima esplorazione delle dinamiche spaziali legate al comportamento del partecipante. Con il file di logging è invece possibile effettuare analisi quantitative sia in merito al percorso, sia approfondendo la dimensione temporale guardando, ad esempio, il tempo trascorso da un evento a un altro.

Nella seconda parte di questo volume impareremo a gestire questa tipologia di dati e in particolare, al capitolo 14, vedremo come

gestire i dati in realtime che derivano da opportuni trigger di eventi opportunamente inseriti in NeuroVR.

6.2 Realtà Virtuale in ambito clinico

In anni recenti, l'uso della RV in ambito clinico, ad esempio in neuroriabilitazione, è aumentato in modo significativo e un numero crescente di evidenze sperimentali suggerisce che questa tecnologia può avere un impatto positivo sul recupero funzionale sia a livello cognitivo, sia a livello motorio (si vedano le rassegne: Holden 2005; Adamovich, Fluet et al. 2009; Lucca 2009, Gaggioli et al. 2009, Saposnik et al. 2011). Il rationale dell'uso dei sistemi RV nell'ambito della riabilitazione è basato su una serie di vantaggi oggi ampiamente documentati dalla letteratura scientifica specialistica, di seguito esposti.

- *Neuroplasticità*: la RV consente di utilizzare scenari progettati sui principi che regolano e facilitano la neuroplasticità (ad esempio: intensità dell'esercizio, frequenza dell'esercizio, "stimolazione arricchita" ecc), ovvero il processo neurobiologico alla base del recupero delle funzioni cognitive e motorie.
- *Training personalizzato*: la RV si basa su meccanismi di funzionamento altamente automatizzati che richiedono un minimo intervento da parte del professionista della riabilitazione, che può avvalersi della possibilità di

personalizzare l'intensità e la difficoltà del training sulla base delle specifiche esigenze del paziente.

- *compiti coinvolgenti*: in RV, il contenuto degli esercizi riabilitativi può essere progettato in modo da definire dei compiti orientati a riaddestrare specifiche abilità (ad esempio, afferrare un oggetto), al tempo stesso inserendo negli scenari dei contenuti ludici per mantenere elevato il livello di coinvolgimento e compliance del paziente nell'esecuzione degli esercizi stessi. In particolare, diverse ricerche hanno evidenziato il ruolo giocato dal RV nell'aumentare il senso di Presenza (Riva e Gaggioli, 2009) e di esperienza ottimale (Gaggioli, 2005) nel processo riabilitativo.
- *tracking e misurazione obiettiva/quantitativa*: grazie ai sensori integrati nei sistemi di RV (es. sensori di tracciamento dei movimenti del cranio, sensori per gli arti superiori, dataglove, ...) è possibile registrare una quantità elevata di dati relativi alle azioni eseguite dal paziente all'interno dello scenario virtuale e utilizzare questi dati per costruire degli indici di performance con cui misurare in modo quantitativo e obiettivo i miglioramenti della performance osservabili durante il percorso riabilitativo.
- *trasferimento del training alle attività di vita quotidiana*: molti studi che hanno esaminato l'uso della RV in campo riabilitativo hanno evidenziato il potenziale offerto da questa metodologia di trasferire i risultati del riapprendimento di

abilità cognitive o motorie deteriorate a causa di danni cerebrali nelle attività di vita quotidiana (Activity of Daily Living, ADL). L'impatto positivo della RV sulle ADL è stato documentato da diversi studi ed è spiegato dal fatto che la RV offre la possibilità di inserire gli esercizi riabilitativi in simulazioni di contesti di vita quotidiana (es. acquistare un oggetto in un supermercato virtuale per riabilitare le funzioni esecutive in pazienti con lesioni frontali).

Questi vantaggi, documentati da un'ampia letteratura e casistica clinica in diversi ambiti patologici (dai disturbi mentali a quelli neuropsicologici, dalle lesioni cerebrali acquisite alle malattie neurodegenerative e all'ictus) hanno aumentato negli ultimi anni l'interesse delle organizzazioni sanitarie e dei professionisti della riabilitazione nei confronti di questa innovativa metodologia. Tale crescente interesse è documentato dall'aumento del numero di studi pubblicato, dalla proliferazione di conferenze e riviste specialistiche e dall'aumento di finanziamenti pubblici e privati alla ricerca sulle applicazioni cliniche della RV. Dal punto di vista dello sviluppo di software specifici, invece, esiste ancora un'offerta piuttosto limitata ed eterogenea di sistemi di RV progettati per la riabilitazione (neurologica o psicologica).

Se da un lato l'uso riabilitativo della RV rappresenta oggi un trend scientifico consolidato e in costante crescita, l'uso di questo strumento nella pratica clinica è ancora molto limitato, soprattutto a livello nazionale. Secondo resoconti recenti, in Italia e a livello

Europeo la RV è ancora utilizzata soprattutto nell'ambito di progetti di ricerca clinica, mentre l'uso professionale è ancora estremamente limitato. Questo dato non può essere spiegato dal livello di immaturità delle componenti tecnologiche, che si sono molto evolute grazie all'enorme sviluppo del mercato dei videogiochi, né dalla mancanza di evidenze scientifiche rispetto all'efficacia di questo approccio. Una spiegazione più probabile della mancata diffusione della RV in campo riabilitativo è sostanzialmente riconducibile a due problemi specifici: a) la mancanza di strumenti di RV di facile uso, basso costo e di elevata affidabilità; b) la scarsa disponibilità di contenuti di tipo riabilitativo, dove per contenuti ci si riferisce segnatamente alle simulazioni interattive finalizzate all'esercizio e alle stimolazioni terapeutiche.

Un ulteriore problema è rappresentato dalla mancanza di soluzioni integrate tra ricerca e clinica: spesso i terapeuti sono interessati non solo alla cura del paziente attraverso l'impiego della RV, ma anche alla raccolta di dati preziosi per il miglioramento dell'efficacia delle soluzioni terapeutiche. Infine, un aspetto da non trascurare riguarda la sempre più sentita esigenza di misurare il cosiddetto "transfer of training", ovvero di poter stabilire in quale misura i risultati ottenuti attraverso l'esposizione alla realtà virtuale possono essere trasferiti nelle attività di vita quotidiana. In questo senso, un bisogno emergente è quello di poter utilizzare efficacemente le possibilità offerte dalle nuove tecnologie mobili (smartphone, sensori indossabili) per consentire al paziente di

proseguire gli esercizi anche a casa propria e fornire al clinico importanti indicazioni rispetto al livello di compliance con le istruzioni terapeutiche.

6.3 Sfide future

La realtà clinica, come abbiamo visto, si sta aprendo sempre di più alle nuove tecnologie e alla realtà virtuale, tuttavia si nota sempre di più una tendenza anche del mondo delle nuove tecnologie ad aprirsi verso nuove soluzioni cliniche e nuovi mercati potenziali. A volte però accade che tale processo è coadiuvato da progetti di ricerca e innovazione: è questo il caso di NeuroVirtual 3D, finanziato dalla Regione Piemonte, per la progettazione, lo sviluppo e la valutazione di un Sistema di Simulazione 3D multiplatforma per applicazioni in campo psicologico e neuroriabilitativo.



Figura 6.13. Logo del progetto NeuroVirtual 3D

Abbiamo prima evidenziato come le esigenze del mondo clinico devono essere supportate da tecnologie progettate e realizzate sulle specifiche peculiarità di tale ambito.

Il progetto NeuroVirtual 3D intende affrontare queste esigenze attraverso la progettazione, lo sviluppo, la valutazione clinico/ergonomica e la divulgazione internazionale di un sistema all'avanguardia.

La piattaforma verrà sviluppata estendendo le caratteristiche del software NeuroVR, che abbiamo prima descritto, attraverso queste aree funzionali:

- sviluppo di interfacce per periferiche hardware di input/output per applicazioni neuroriabilitative (es. dataglove, dispositivi aptici, Kinect);
- integrazione con dispositivi di *eye-tracking* e *brain-computer interface*;
- sviluppo modalità di interazione e comunicazione multi-utente mediante avatar;
- sviluppo di contenuti 3D su dispositivi mobili (android, iPhone/iPad)
- sviluppo di un *web repository* delle scene 3D per la condivisione degli ambienti tra gli utenti del software.

In particolare, la piattaforma NeuroVirtual 3D verrà sviluppata estendendo le caratteristiche del software NeuroVR, attraverso cinque dimensioni funzionali:

1. Sviluppo di interfacce per periferiche hardware di input/output per applicazioni neuroriabilitative (es. dataglove, dispositivi aptici, Kinect).

Recentemente, abbiamo assistito a una progressiva diffusione nel mercato *home gaming* di avanzate tecnologie di gioco (come Microsoft Kinect, Nintendo Wii) che ha reso disponibili una serie di dispositivi interattivi di basso costo (Bobick and J. Davis, 2011; D. Glowinski et al., 2011; Rigas, G. et al., 2009). La maggior parte di questi dispositivi, oltre ad avere un basso costo (perché destinati ad un mercato consumer) è supportata da Software Development Kit che consentono l'integrazione con programmi di terze parti. Il progetto NeuroVirtual 3D intende avvalersi di questi dispositivi di gaming commerciale per integrare nuove funzionalità interattive nella piattaforma NeuroVR che siano utilizzabili e testabili in campo neuroriabilitativo. Questa strategia mira ad ottenere un duplice risultato: da un lato, aggiungere nuove potenzialità interattive alla piattaforma di RV per supportare un più vasto *range* di contenuti ed esercizi riabilitativi; dall'altro, di trarre vantaggio dall'elevata diffusione di queste tecnologie per offrire una soluzione che integra dispositivi di basso costo e facile reperibilità.

2. Integrazione con dispositivi di eye-tracking e brain-computer interface

Da diversi decenni la registrazione e l'elaborazione dell'attività cerebrale e oculomotoria costituiscono un prezioso strumento per

la valutazione in ambito clinico e neuropsicologico (Diefendorf, A.R. and R. Dodge, 1908; Hallett, P.E., 1978). In particolare l'eye-tracker, come abbiamo già visto nel terzo capitolo, ha visto un uso crescente nella valutazione in ambito clinico. Questa avanzata tecnologia/metodologia è capace di individuare e registrare con elevata precisione spaziale e temporale i movimenti oculari, sincronizzandoli con gli stimoli presentati. Tradizionalmente la metodologia impiegata è tramite paradigmi saccadici, antisaccadici e di *smooth pursuit* (come già visto). Tuttavia di recente l'ambito di sviluppo di tali paradigmi si è notevolmente esteso, passando dall'assessment clinico a un utilizzo attivo dello strumento come piattaforma di input. In tal senso, l'uso più esteso è l'Alternative and Augmented Communication (AAC), tramite cui è possibile fare comunicare "tramite lo sguardo" anche i pazienti con deficit motori e/o comunicativi (Cipresso, P. et al., 2011). Attraverso l'integrazione del tracciamento oculare nella piattaforma NeuroVR sarà possibile ottenere un'efficace metodologia per monitorare i movimenti oculari durante l'esperienza immersiva, ottenendo preziose informazioni sull'attività cognitiva del partecipante, come l'attenzione visiva, la percezione, il ragionamento, la ricerca di informazioni e la valutazione di stimoli ambientali di complessità variabile. In particolare, l'integrazione con eye-tracking e Brain Computer Interface (BCI) determinerà i seguenti vantaggi:

- Incremento dell'accuratezza diagnostica nella valutazione dei progressi riabilitativi;

- Possibilità di correlare specifici stati mentali a specifiche attività eseguite all'interno degli ambienti virtuali, attraverso l'uso di marker ambientali che consentiranno di sincronizzare il tracciato oculare con le azioni compiute dall'utente;
- Possibilità di studiare le variabili attentive, percettive e cognitive nell'ambito di simulazioni che riproducono realisticamente situazioni e contesti quotidiani, aumentando la validità ecologica dei dati raccolti.

3. Sviluppo modalità di interazione e comunicazione multi-utente mediante avatar

Un altro elemento di innovazione tecnologica introdotto dal progetto NeuroVirtual 3D riguarda lo sviluppo della funzionalità multi-utente, attualmente non presente all'interno della piattaforma NeuroVR, ma della cui importanza abbiamo detto parlando di Second Life. La fruizione degli ambienti virtuali in modalità multiutente consente di ampliare il potenziale applicativo di questa piattaforma in ambito riabilitativo, consentendo un uso esteso di paradigmi altrimenti complessi, come ad esempio quelli che richiedono la simulazione di situazioni sociali (ad esempio il paziente che deve imparare a gestire una fobia sociale, come la paura di parlare in pubblico, in un ambiente di RV) o situazioni di tipo riabilitativo nelle quali la co-presenza del terapeuta può rappresentare un valore aggiunto nella motivazione del paziente

allo svolgimento degli esercizi (ad esempio in contesti di telemedicina).

4. Sviluppo di contenuti 3D per dispositivi mobili (android, iPhone/iPad)

La crescente diffusione delle piattaforme mobili, come smartphone e tablet, rappresenta una significativa opportunità per chi si occupa di riabilitazione, purtroppo ancora non sufficientemente esplorata. Grazie al progressivo incremento della loro capacità computazionale e di memorizzazione questi dispositivi consentono di supportare simulazioni 3D interattive di elevato realismo e complessità, che possono essere efficacemente utilizzate a fini riabilitativi per consentire al paziente di continuare ad esercitarsi in contesti domiciliari o in mobilità. L'obiettivo è riuscire a realizzare contenuti ed esercizi di riabilitazione virtuale che possano essere utilizzati in ambito ospedaliero in presenza del terapeuta e quindi liberamente ripetuti presso il domicilio del paziente mediante App installate su dispositivi mobili come iPhone, iPad e Android.

5. Sviluppo di un repository online delle scene 3D per la condivisione degli ambienti tra gli utenti del software

Un'ulteriore innovazione proposta dal progetto NeuroVirtual 3D riguarda la creazione di un repository online di contenuti 3D clinicamente testati, corredati da protocolli e procedure da condividere con la comunità dei professionisti della riabilitazione

che operano con la realtà virtuale o sono interessati a sperimentare questo approccio. La possibilità di accedere con facilità a tali contenuti favorirà una maggiore diffusione di protocolli e consentirà ai ricercatori di eseguire sperimentazioni su larga scala per riuscire a ottenere un maggior numero di evidenze cliniche e raggiungere una massa critica di studi a supporto dell'uso della RV nella ricerca e nella clinica.

PARTE II

GESTIONE AVANZATA DEI DATI

Software per la gestione avanzata dei dati

In questo capitolo cominciamo a entrare nel vivo della psicometria computazionale cominciando a vedere le operazioni e i software che ci permettono di effettuare operazioni avanzate sui dati.

Quello a cui siamo maggiormente interessati nell'ambito di questo volume è mettere ordine tra i dati per consentire un'analisi coerente a partire da informazioni di sintesi. Nella prima parte del volume abbiamo visto come la metodologia computazionale permette la raccolta, a uso di svariati strumenti, di una grande massa di dati. Se è dunque vero che strumenti come i questionari elettronici, i biosensori e l'eye-tracker consentono una efficace raccolta di una moltitudine di dati, è anche vero che questa grande numerosità e complessità va gestita con opportuni strumenti sia per quanto attiene la loro gestione, sia per tutte le procedure di analisi e riorganizzazione che ne conseguono.

Il problema che si riscontra, infatti, nella psicometria computazionale è che a fronte di una grande capacità di raccolta dati deve esserci anche una loro educata riorganizzazione tramite strutture coerenti e capaci di gestire tale complessità.

7.1 Organizzazione dei dati in un foglio di calcolo

Nei sei capitoli precedenti abbiamo visto le varie metodologie che ci consentono una raccolta dati, generalmente archiviati in file di testo, come quello di esempio in Figura 7.1. Questi dati presentano diversi livelli di complessità a volte data dalla dimensione stessa, altre volte derivante dalla struttura che lega gruppi di informazioni in essi contenute.

<pre>Export Channel Data with rate of 256 samples per second. Client: AUXO-IULM EXP Channel Set: PI 1 TT av sync AUXO.chs Session Date: 21/10/2010 Session Time: 16.38.44 Start Time: 00:00:04.000 End Time: 00:15:22.063</pre>	Descrizione generale dei dati
<pre>AV Sync - 1A,EKG-Pro/Flex - 1B,MyoScan-Pro 400 - 1C,EEG-2 - 1D,E: Skin Cond,MyoScan-Pro 400 - 1F,Resp-Pro/Flex - 1G,HR/BVP-Pro/Flex - 1H,IBI from BVP,Respiration Period,Respiration Amplitude,HR Max - HR Min,LF Total power,HF Total power</pre>	Descrizione dei segnali registrati
<pre>2048,2048,256,256,256,256,256,256,256,256,256,256,16,16 -1.660,-2.114,- 55.348,3.188,29.524,42.952,22.736,33.550,0.000,0.000,0.000,0.000, 0.000,0.000 -1.660,-2.760,- 78.829,3.095,29.524,40.569,22.733,33.676,0.000,0.000,0.000,0.000, 0.000,0.000 -1.660,-2.563,- 103.116,3.145,29.523,38.833,22.730,33.834,0.000,0.000,0.000,0.000 ,0.000,0.000</pre>	Dati

Figura 7.1. File di testo contenente vari dati psicofisiologici.

In questo capitolo affrontiamo la primissima gestione di questi dati complessi e cominciamo a capire se o quando conviene usare i classici fogli di calcolo (come Microsoft Excel) o se siano necessari dei software più evoluti.

Ad una prima ispezione visiva, questi file di testo, appariranno piuttosto confusi e disordinati e, sebbene siano identificabili sia i dati sia le loro descrizioni, risulterà complesso effettuare un'accurata descrizione, anche solo descrittiva. Inoltre, se i file di dati sono di grosse dimensioni, potremmo avere grosse difficoltà ad aprirli con un normale visualizzatore, rischiando anche di bloccare il computer. Dunque, per una prima visualizzazione e trattazione di dati, uso di seguito Microsoft Excel, mostrandone vantaggi e limiti strutturali di questo software. In Figura 7.2 è possibile vedere, in un nuovo file appositamente aperto, la scheda dati e il pulsante (cerchiato) per l'importazione di dati di testo.

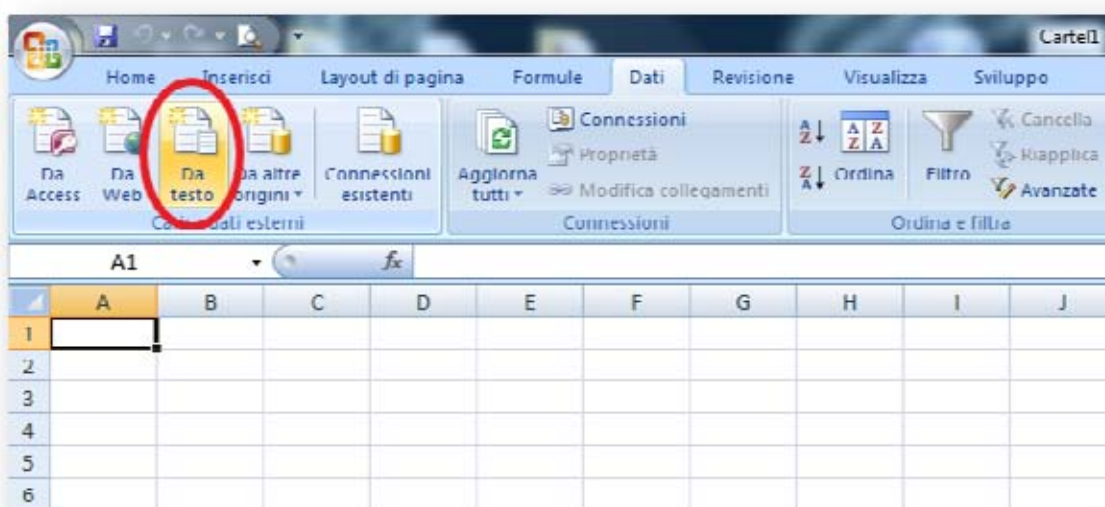


Figura 7.2. Importazione di dati da testo, in Microsoft Excel.

L'importazione in Excel viene eseguita tramite una procedura guidata piuttosto semplice, che di seguito esporremo dettagliatamente, usando come riferimento un file di testo (1.txt, come da Figura 7.3) ottenuto da una registrazione di dati psicofisiologici, disponibile nel materiale supplementare al volume.

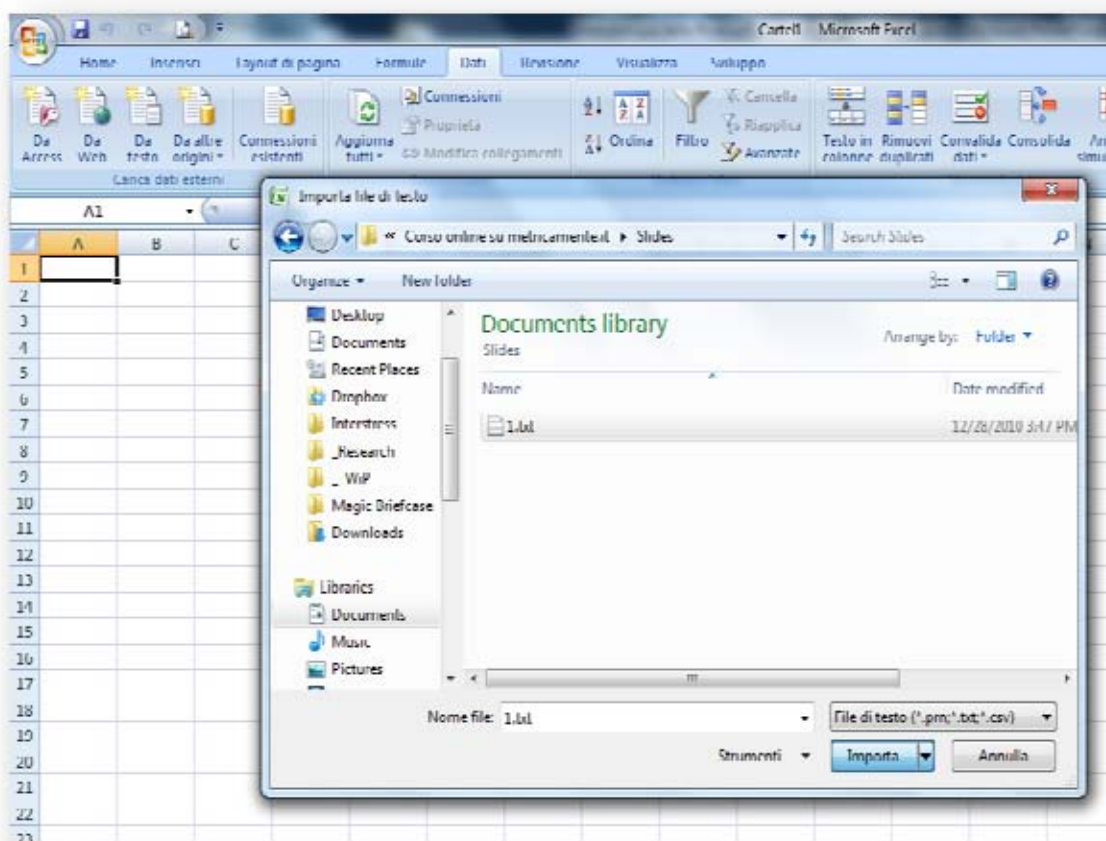


Figura 7.3. Importazione del file di dati psicofisiologici "1.txt".

La selezione dei file è tanto più accurata quanto più si sono seguite le indicazioni già date nel precedente paragrafo 2.4. In questo

capitolo la procedura di corretta codifica dei nomi dei file, ha un solo scopo organizzativo per chi deve effettuare l'importazione. Invece, nei capitoli seguenti, vedremo come questa codifica è strettamente necessaria, in psicomatria computazionale, per una corretta importazione ed elaborazione di dati complessi.

La procedura di importazione dei dati in Microsoft Excel si avvia automaticamente subito dopo aver selezionato il file (1.txt nel nostro esempio). E' un processo a tre passi, il primo dei quali in Figura 7.4. All'occorrenza si può interrogare la guida utente premendo il tasto F1 (posizionato accanto al tasto ESC).

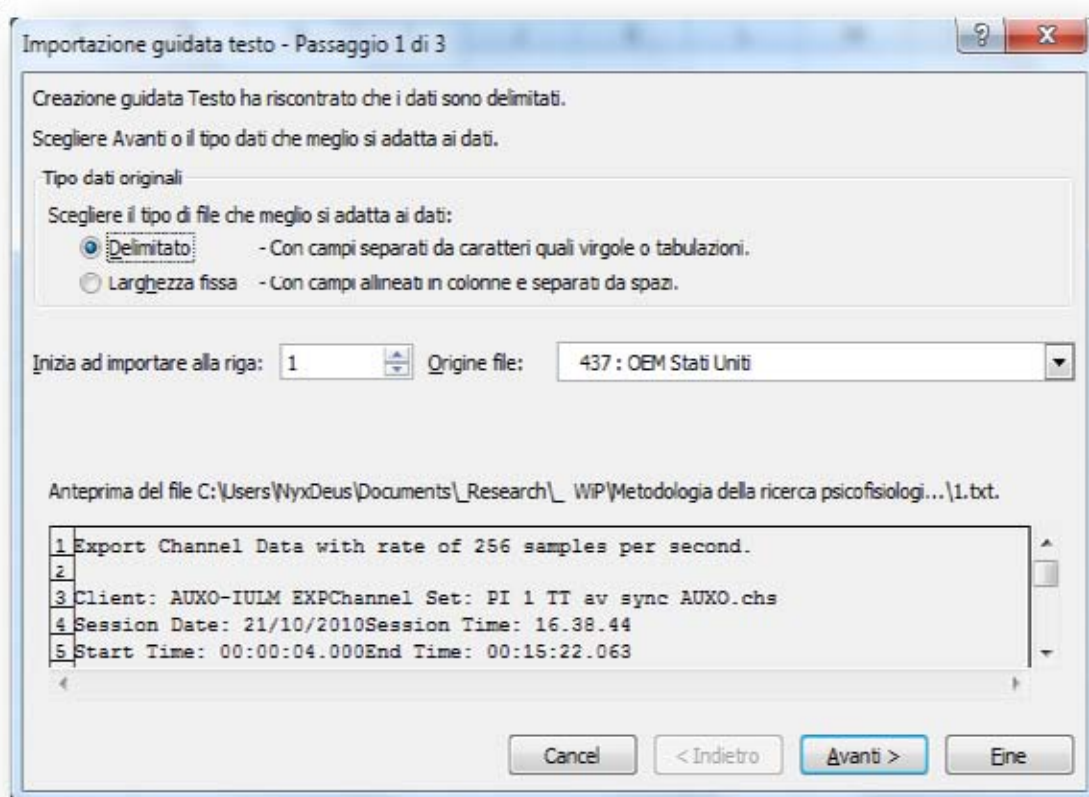


Figura 7.4. Primo passo dell'importazione guidata.

Il processo di importazione è assai semplice e l'utente viene guidato passo dopo passo in modo intuitivo e con spiegazioni subito evidenti (come la riga da cui iniziare per l'importazione), ma anche disponibili in modalità più approfondita nella guida. I software visti nei primi sei capitoli prevedono sempre una registrazione di dati in file di testo, generalmente organizzando al loro interno informazioni divise per mezzo di un qualche delimitatore (virgola, spazio, tabulazioni, ...). Settando al primo passo dell'importazione "Delimitato", avremo la possibilità di scegliere al secondo passo il simbolo che divide le informazioni in modo coerente e sarà anche possibile verificare se il simbolo, o i simboli, scelti forniscono una coerente divisione (Figura 7.5).

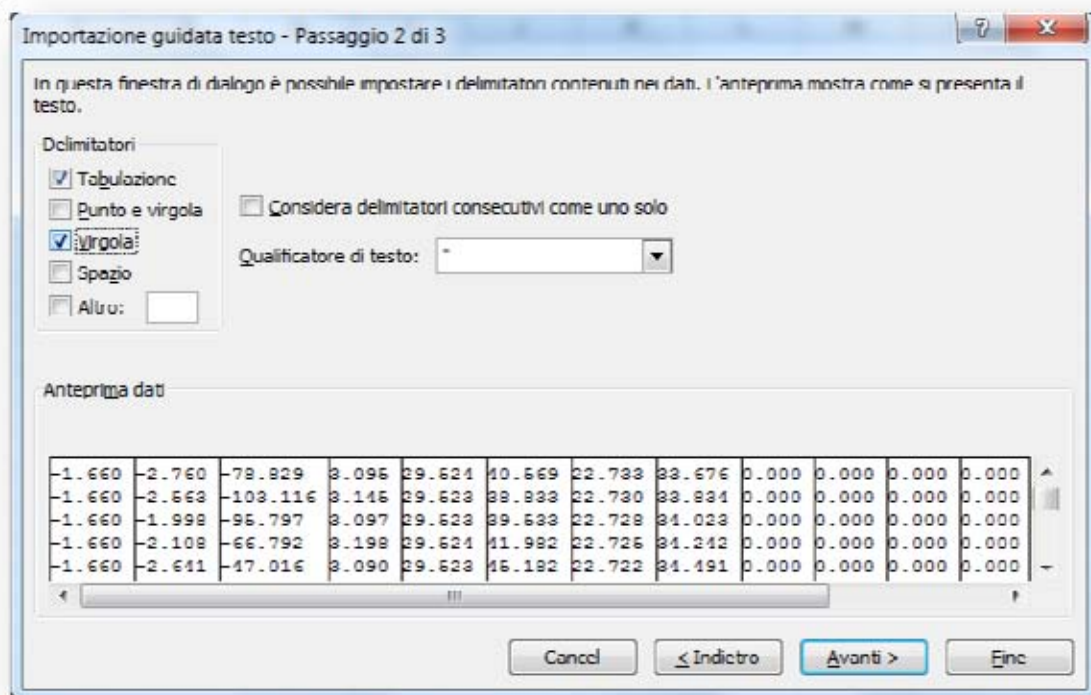


Figura 7.5. Importazione guidata: scelta del delimitatore di testo.

Al terzo passo dell'Importazione guidata, definiamo i parametri di separazione dei decimali (premendo il tasto "Avanzate...") e la posizione all'interno del foglio di lavoro dei nuovi dati da importare: generalmente si parte dalla cella \$A\$1, la prima in alto a sinistra, come da Figura 7.6.

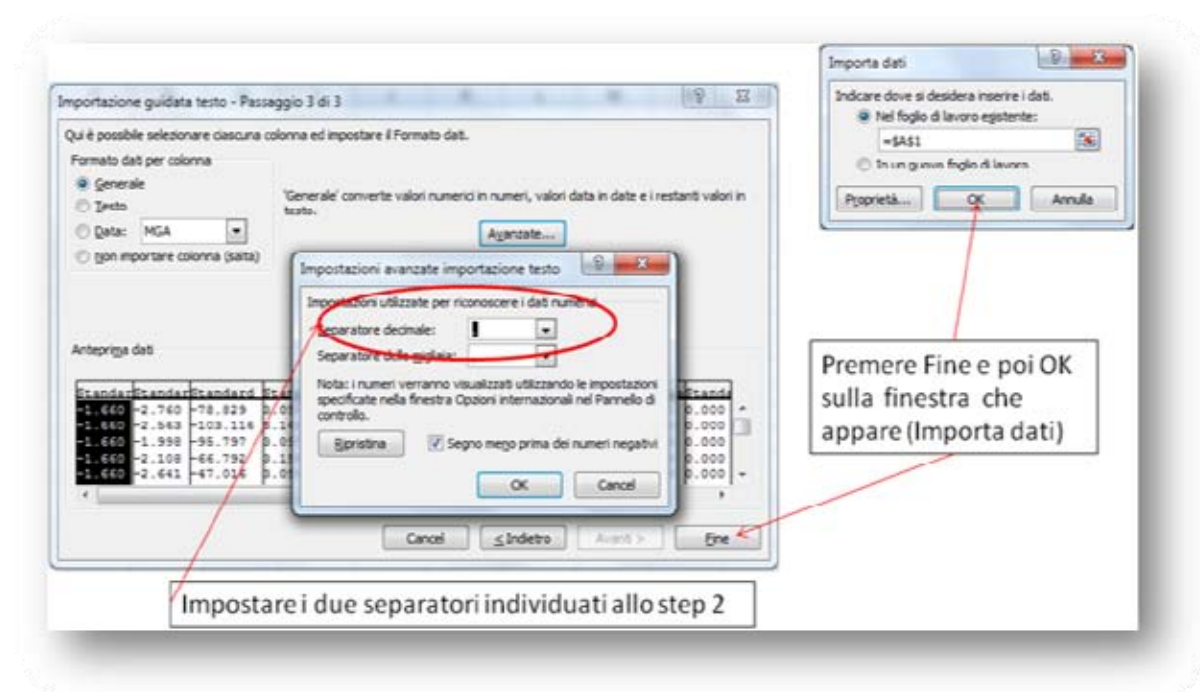


Figura 7.6. Finalizzazione della procedura di importazione.

Come risultato otterremo una matrice ordianata di dati suddivisi per colonna a partire da una riga di intestazione, generalmente già prevista dai software di acquisizione dei dati o comunque facilmente inseribile (Figura 7.7). Questa suddivisione permette già una prima visualizzazione dei dati e la possibilità di effettuare

semplici operazioni statistiche (medie, somme, conteggi, ...) o grafici dei dati.

The screenshot shows a Microsoft Excel spreadsheet with the following data structure:

AV Type - TA	1000-Phz) (Sec - 10)	Mythos-Pro 100 - 10	11 (G) - 10	1:5km Exand	Mythos-Pro 100 - 11	1000-Phz) (Sec - 10)	1000-Phz) (Sec - 11)
7	-1.88	-2.118	-55.188	3.108	21.521	82.787	22.728
10	-1.88	-2.18	-81.079	3.078	21.521	81.983	22.721
11	-1.88	-2.583	-103.118	3.108	21.521	98.833	22.721
17	-1.88	-1.938	-98.297	3.097	21.521	99.833	22.728
13	-1.88	-2.108	-88.297	3.108	21.521	81.983	22.728
14	-1.00	-2.081	-87.010	3.00	21.521	85.183	22.722
15	-1.00	-1.920	-83.183	3.095	21.521	87.183	22.721
16	-1.00	-1.717	-51	3.095	21.521	85.183	22.717
17	1.00	3.007	88.188	3.118	24.528	83.281	25.715
18	1.00	3.081	90.848	3.111	24.528	84.188	25.718
19	1.00	2.120	24.205	2.807	24.528	26.127	22.71
20	1.00	2.247	28.18	2.807	24.528	28.127	22.712
21	1.00	2.828	82.081	2.807	24.528	82.188	22.715
22	1.00	2.71	72.287	2.824	24.528	78.284	22.718
23	1.00	3.141	34.727	3.401	29.522	40.927	22.701
24	1.00	3.289	34.289	3.285	29.522	41.701	22.689
25	1.00	4.278	88.424	4.529	29.522	41.728	22.692

Figura 7.7. I dati importati in Microsoft Excel e visibili con una separazione coerente rispetto alle informazioni contenute.

Se è vero che la procedura di Microsoft Excel è molto rapida e consente un'importazione in tre semplici passi, assai intuitivi, è anche vero però che il programma presenta dei limiti strutturali e di gestione. Vediamone alcuni:

- gestisce solo un milione di righe, quindi se campiono a 2048 Hz (2048 righe per ogni secondo) posso importare poco più di 8 minuti di dati;

- può graficare non più di 32.000 righe, dunque ci si trova facilmente a non poter visualizzare i dati o a doverli visualizzare un po' per volta;
- è tanto più lento quanti più dati ci sono, dunque anche solo aprire un file può richiedere molto tempo;
- Può effettuare un numero limitato di operazioni.

Tuttavia a volte Microsoft Excel può essere utile:

- Nel caso in cui si devono fare semplici operazioni di visualizzazione e calcolo di segnale a basso campionamento. Se per esempio si deve analizzare la conduttanza cutanea (SC, skin conductance), si può campionare anche a 32 Hz (32 righe al secondo), allora può avere senso usare Excel per fare i grafici (max 16 minuti di segnale) e calcolare le medie e altri indici statistici.
- Quando si è già finita l'analisi del segnale e si deve lavorare sugli indici estratti, dunque su matrici di piccole dimensioni.

L'importazione dei file di testo in Microsoft Excel, presenta dunque dei vantaggi in termini di semplicità di visualizzazione e di prima gestione dei dati. E' molto utile, anche quando si devono riportare gli indici estratti dalle complesse informazioni contenute nei dati e consente anche una prima esplorazione di dati statistici descrittivi. Infine Microsoft Excel ha l'ulteriore vantaggio, da non sottovalutare, di essere disponibile a un vasto pubblico, pur essendo un prodotto commerciale.

Nel materiale supplementare al volume potete trovare il file di esempio "1.txt", contenente alcuni dati psicofisiologici, da usare per eseguire l'esercitazione seguente (Figura 7.8).



Figura 7.8. Esercitazione sull'import dei dati e prima visualizzazione di un segnale psicofisiologico (Skin Conductance).

Nel prossimo paragrafo descriveremo dei software atti a gestire la complessità generata dai dati visti nella prima parte del volume, ma attenzione: non abbandoneremo comunque i fogli di calcolo, anzi impareremo a usarli al meglio delle loro possibilità e

soprattutto quando aiutano a semplificare processi che con altri software sarebbero complessi.

7.2 Analisi accurata dei dati

Come abbiamo più volte detto, i dati generati con i metodi visti nella prima parte del volume sono tanti o comunque assai complessi. Se ad esempio importiamo il file "1.txt", fornito per l'esercitazione, vedremo che ci sono oltre 235 mila righe di Excel in 14 colonne. Anche se il programma è in grado di visualizzare tutte le celle, avremo molti problemi legati alla loro gestione. Se proviamo a graficare il segnale di una colonna ci verrà dato un messaggio che ci avvisa che non è possibile graficare più di 32 mila righe in Excel (vedi Figura 7.9).

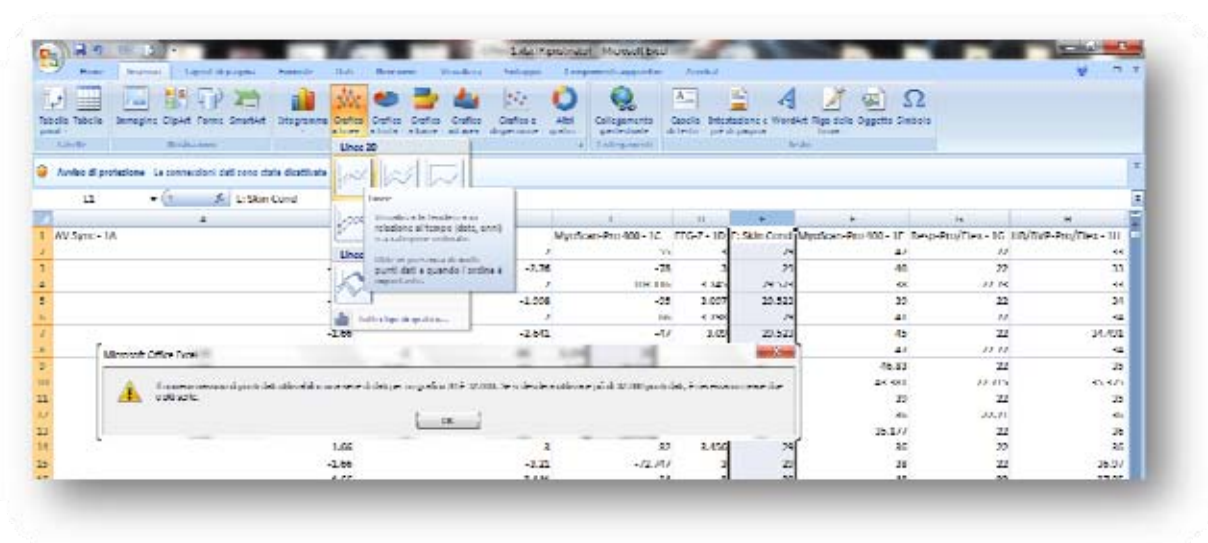


Figura 7.9. I grafici in Excel non possono essere per più di 32 mila righe.

Inoltre il grafico prodotto non è visivamente accettabile, come si può veder in Figura 7.10, in quanto Excel non è pensato per graficare un tale ammontare di dati, anche se è ottimo per graficare dati di sintesi, quali medie, somme, conteggi. Non si discute quindi sulla qualità dei fogli di calcolo, ma piuttosto sulla loro inadeguatezza per questa fase di analisi della psicometria computazionale.

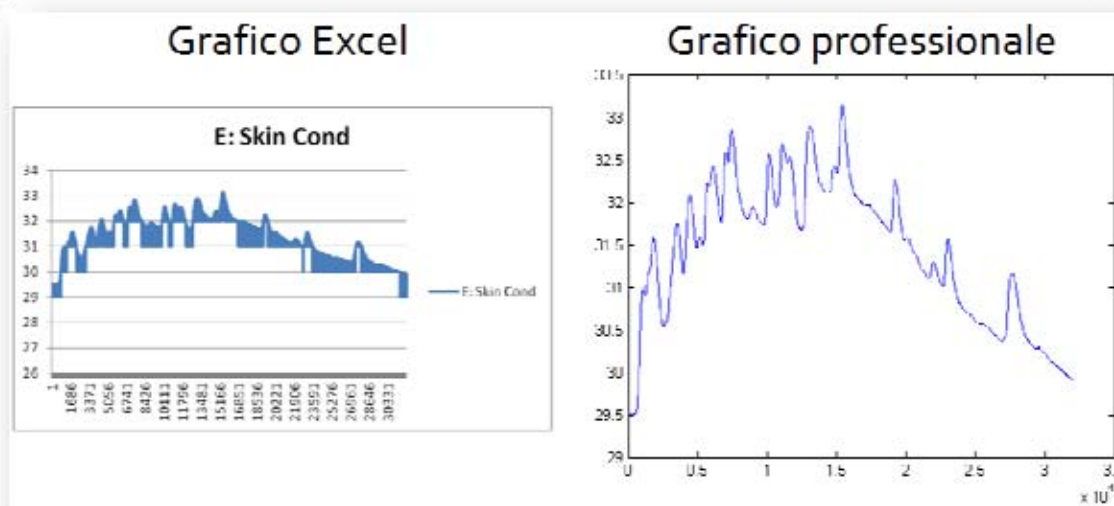


Figura 7.10. Grafici di 32 mila righe con differenti software.

Evidenziati dunque alcuni limiti dei fogli di calcolo, vedremo di seguito alcuni software per la gestione di dati complessi.

Il corretto uso di questi software ci semplificherà notevolmente il lavoro, permettendo:

- Risparmio di tempo;
- Possibilità di trattare dati a complessità variabile;

- Semplificazione di operazioni complesse;
- Gestione di tutte le fasi della ricerca.

Nella visione di questo volume cercheremo sempre di usare software gratuiti, ma vorrei sottolineare che quanto appreso in questa seconda parte del volume è utile in termini di logica di processo e non di singola procedura di un software. Che si usi un software o un altro, per effettuare un'operazione (ad esempio per graficare dei dati), è irrilevante, ma è il processo che seguo quello che conta, il resto è solo sintassi.

Fatta questa importantissima premessa, passiamo adesso a esaminare i tre software più adatti a gestire dati complessi:

- Matlab (<http://www.mathworks.it/products/matlab/>);
- GNU Octave (<https://www.gnu.org/software/octave/>);
- ed R (<http://www.r-project.org/>).

Questi tre software sono anche contenitori per una vasta gamma di altri software usati per funzioni più specifiche di intervento sui dati. R è anche molto usato nella psicometria classica, tuttavia i dati che trattiamo in questo volume sono per loro natura e specie a maggiore complessità dei classici dataset usati per effettuare analisi statistiche. Ciò implica che anche chi ha esperienza con l'uso di R per la statistica, potrebbe non essere in grado di effettuare certi semplici calcoli più tipici della psicometria computazionale e anche se ovviamente risulterà più preparato all'apprendimento articolato nei prossimi capitoli, dovrà imparare

alcune logiche di funzionamento di base per il trattamento dei dati complessi, come ad esempio i cicli e le condizioni.

Mentre R è un open source gratuito, Matlab è a pagamento. Esistono tuttavia una vasta gamma di software gratuiti e open source che usano una sintassi molto simile a quella di Matlab. In particolare GNU Octave usa una sintassi talmente compatibile con Matlab che è stato definito il suo clone.

In pratica chi impara a usare Matlab, sa usare anche GNU Octave e altri software simili. Inoltre le differenze tra Matlab e R, ai nostri fini, sono solo in termini di sintassi e non di logica e anche le strutture usate sono infatti le stesse, cambia al più il nome.

Usare Matlab, R o un altro software non fa dunque molta differenza in psicometria computazionale, va comunque capita la logica e poi implementata nel software che si utilizza.

Matlab è anche disponibile con una licenza studenti a \$ 89,00 inclusi dodici prodotti e funziona su Windows, Mac e Linux.

In questo volume useremo perlopiù Matlab, con la considerazione che è già disponibile in oltre 5000 Università nel mondo. Qualora non fosse disponibile al lettore, suggerisco il download gratuito di GNU Octave che consente l'uso di tutti i dati e codici disponibili a nel materiale supplementare al volume.

Chi non ha mai aperto nessuno dei software di cui sopra potrebbe sentirsi assai disorientato: sono infatti software molto flessibili ma alla loro apertura l'elemento comune è il cursore lampeggiante in attesa di comandi e capisco perfettamente che questo possa preoccupare il lettore che non abbia mai programmato.



Figura 7.11. Matlab è tra i software più usati al mondo in ambito scientifico.

Tuttavia la programmazione di questi software è molto semplice e permette di effettuare operazioni sui file che altrimenti non sarebbero possibili.

Per meglio orientarsi su questi software può essere utile avere una minima interfaccia grafica, come quella che già Matlab fornisce di default. In particolare GNU Octave, una volta installato si apre come finestra con solo il cursore e, così com'è, può essere scomodo anche per un esperto che voglia rapidamente operare sui dati. Per fortuna la comunità di Octave è molto attiva e ha creato DomainMath IDE (<http://sourceforge.net/projects/domainmath/>), un interfaccia grafica per GNU Octave, disponibile come open source totalmente gratuita. Questa interfaccia trasforma globalmente il programma per renderlo totalmente uguale a Matlab

anche nella grafica. In Figura 7.11 vediamo l'effettiva trasformazione estetica.

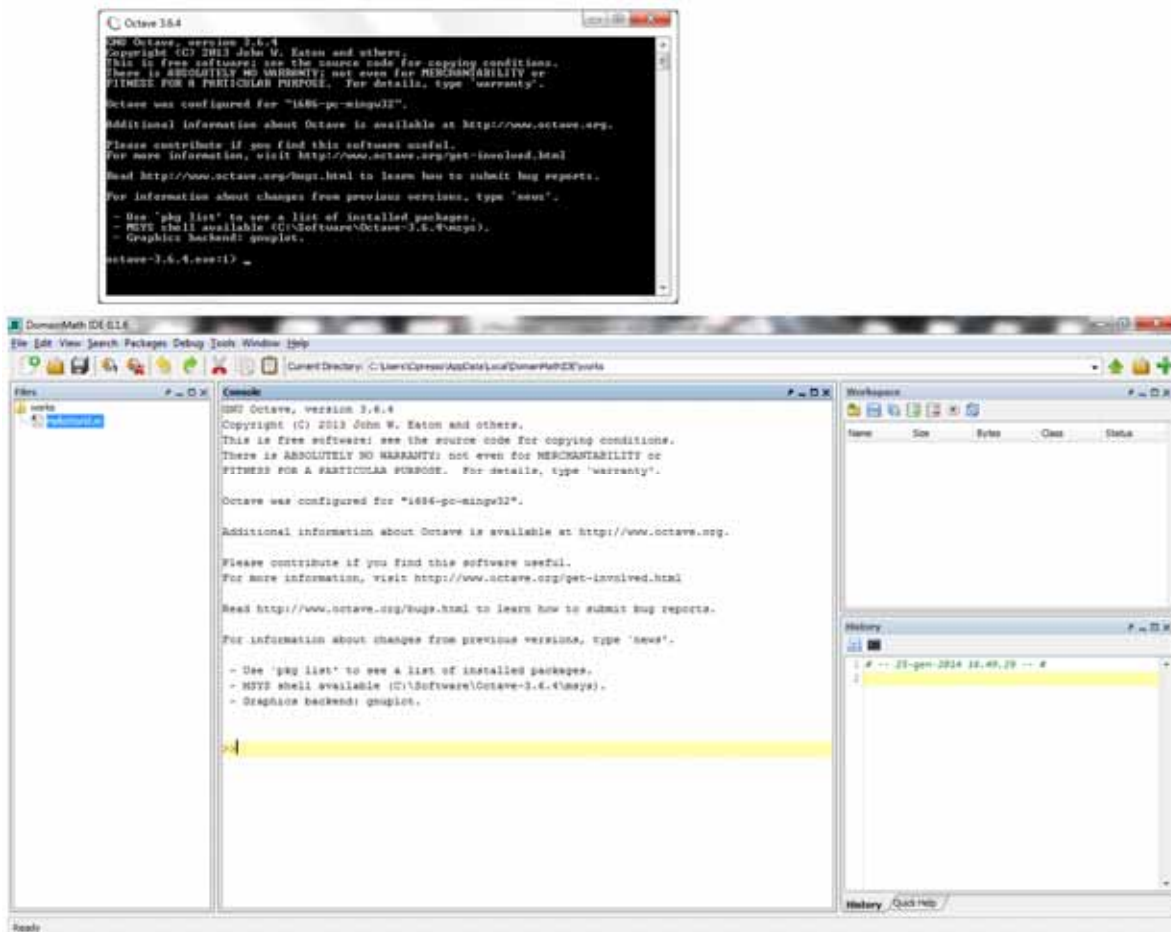
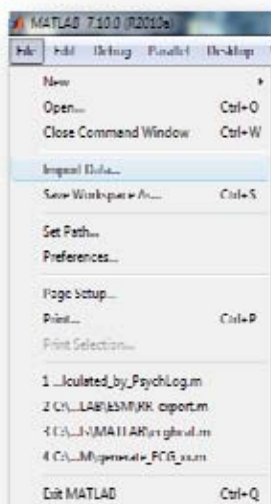


Figura 7.12. Interfaccia di Octave prima (sopra) e dopo (sotto) il trattamento estetico.

L'interfaccia grafica non cambia i comandi, che restano identici.

Per completezza segnalo che anche per R esiste una simile interfaccia grafica, RStudio (<http://www.rstudio.com/>), che consiglio vivamente.



In matlab useremo un solo "wizard": quello di importazione dei dati. Per il momento serve solo per dare l'illusione che tutto andrà bene, ma alla fine di questa seconda parte del volume vorrete fare anche questa operazione tramite codice e non per virtuosismo, ma semplicemente perché vi sarà chiaro come scrivere tre righe di semplicissimo codice vi può semplificare la vita.

Come prima abbiamo imparato a importare correttamente il file "1.txt" in Excel, adesso vedremo come importarlo in Matlab (Figura 7.13).

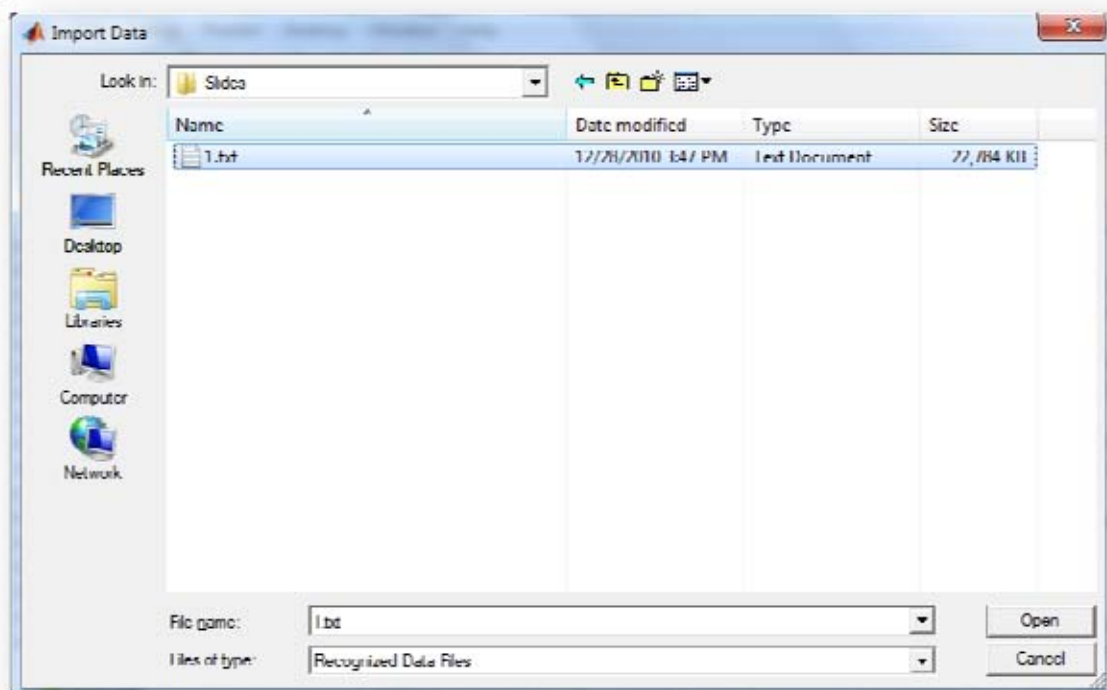


Figura 7.13. Importazione di un file di testo con Matlab.

Così come abbiamo fatto per Excel, anche in questa importazione dobbiamo indicare il delimitatore di dati (virgola, spazio, ...) e il numero di righe di intestazione, considerando peraltro che in Matlab importeremo solo i dati numerici (Figura 7.14).

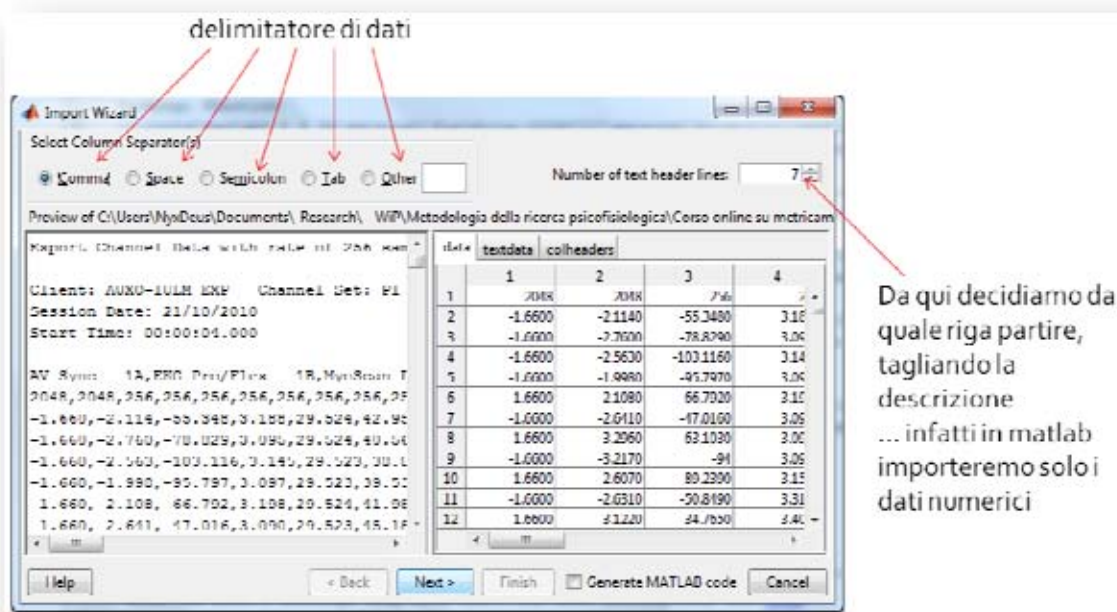



Figura 7.14. Configurazione dell'importazione in Matlab

Appena abbiamo verificato che i dati vengono importati in maniera corretta possiamo continuare la procedura andando al passo successivo (premendo il pulsante "Next"). Il successivo passo è abbastanza importante in quanto, come possiamo vedere dalla Figura 7.15, abbiamo per la prima volta l'evidenza delle strutture in Matlab, con in particolare la struttura con il simbolo  che

indica una matrice di nome (Name) "data" e di dimensioni (Size) 235025x14, ovvero oltre 235 mila righe in 14 colonne.

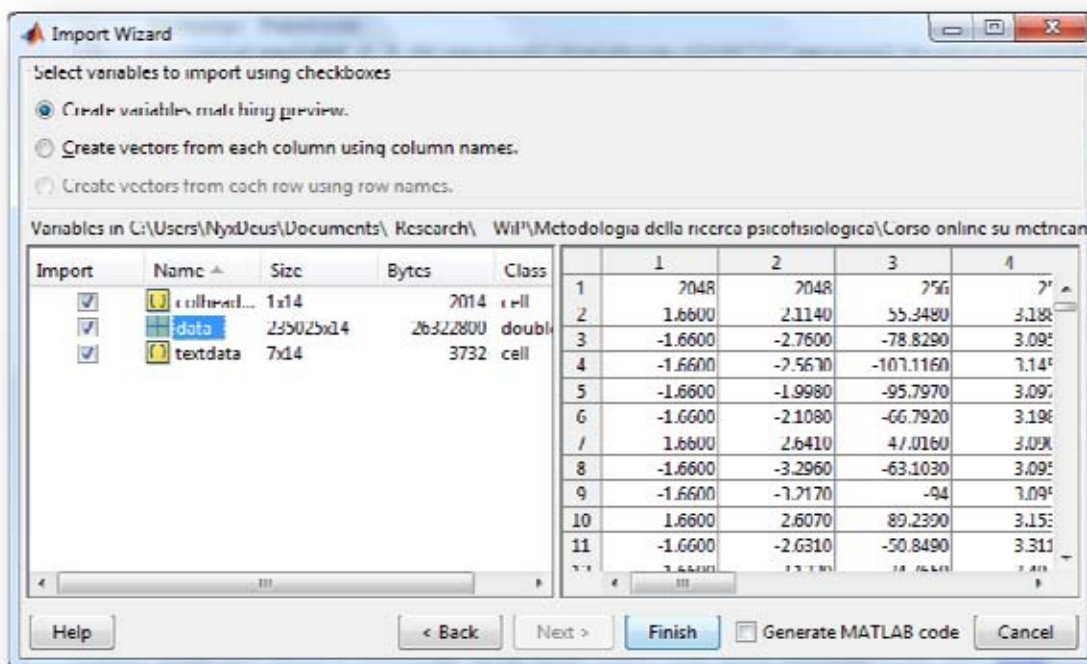


Figura 7.15. L'ultimo passo dell'importazione in Matlab, con l'indicazione della matrice "data".

Ci sono delle notazioni importanti in Matlab, che di volta in volta verranno evidenziate con cura. In generale nella psicometria computazionale bisogna essere piuttosto attenti e accurati: una piccola distrazione può bloccare un intero processo.

Fatta questa premessa la prima notazione da fare è sul nome delle variabili (anche la matrice di dati "data" è una variabile per

Matlab), che devono seguire delle precise regole, secondo quanto segue:

- devono iniziare con una lettera;
- possono contenere solo caratteri alfanumerici e il trattino basso (underscore, ovvero il simbolo "_");
- non possono essere una parola chiave (keyword) di Matlab.

Inoltre i nomi sono "case sensitive" quindi, ad esempio, "data" e "Data" sono due nomi diversi per Matlab e potranno anche essere due diverse variabili.

Non appena finita l'importazione le variabili saranno visibili sul pannello di destra, detto Workspace, e potranno essere visualizzate cliccando due volte sul loro nome (Figura 7.16). Nel caso specifico, la matrice "data" è una variabile di tipo "double".

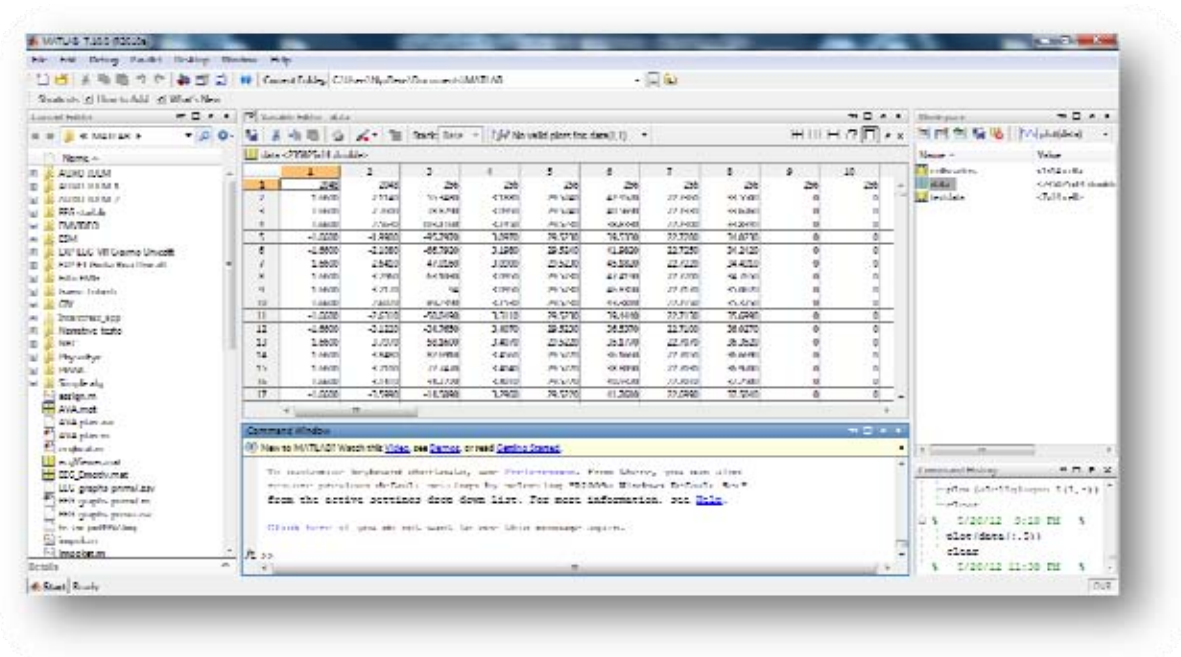


Figura 7.16. Visualizzazione delle variabili in Matlab.

Avere i dati in Matlab consente una loro gestione computazionale di alto livello. Fare un grafico della colonna 5 (Skin Conductance, nel nostro solito esempio), ci darà un risultato immediato e non avremo problemi di dimensione della matrice. Possiamo dunque graficare tutti gli oltre 235 mila punti che costituiscono l'intero segnale, come da Figura 7.17.

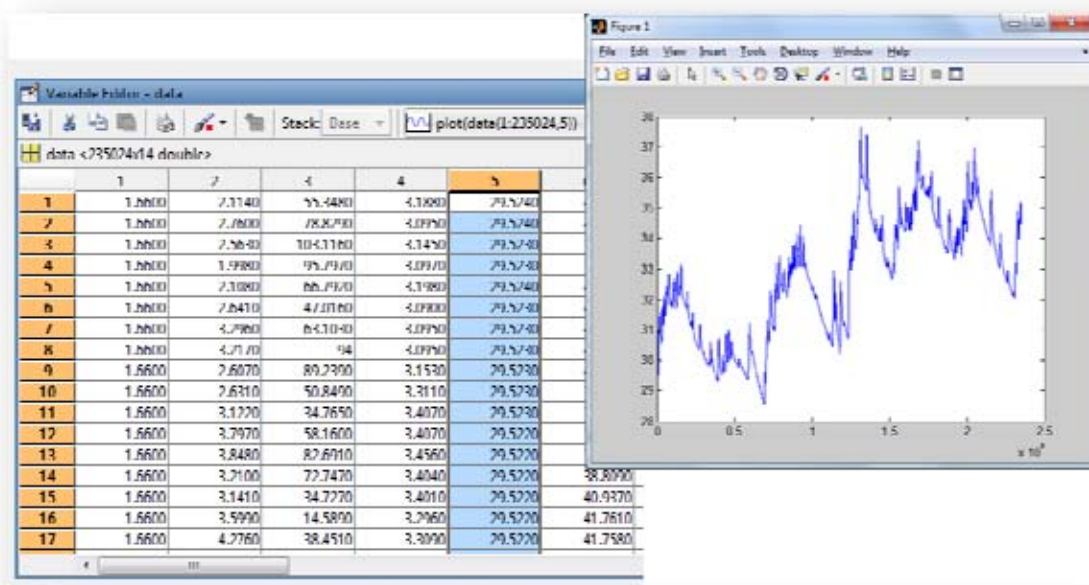
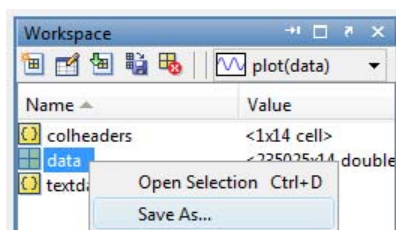


Figura 7.17. Grafico dell'intero segnale in colonna 5, con Matlab.



L'esecuzione e la manipolazione dei dati sarà anche molto più veloce, rispetto ad Excel e sarà anche possibile salvarli in locale sul disco semplicemente cliccando con il tasto destro del mouse sul nome della variabile nel Workspace e selezionando "Save as...".

Nel materiale supplementare al volume potete trovare il file di esempio "1.txt", contenente alcuni dati psicofisiologici, da usare per eseguire l'esercitazione seguente (Figura 7.18).



Figura 7.18. Esercitazione sull'import dei dati e prima visualizzazione di un segnale psicofisiologico (Skin Conductance).

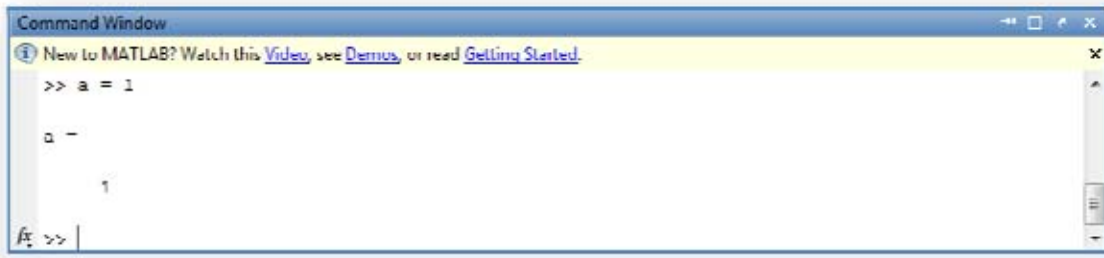
Appena conclusa l'esercitazione ricordiamoci di salvare, su una cartella di lavoro del computer, la matrice dei dati con il nome "data". Considerazione non banale è quella di essere molto ordinati con i dati. Personalmente tengo sempre una cartella "DATA" direttamente sotto il disco principale C (abituamoci anche a

pensare e scrivere questo percorso in formato "C:\DATA"). Sotto questa cartella ho delle cartelle con i nomi delle sperimentazioni correnti (ad esempio "C:\DATA\Exp Arte") e dentro questa cartella salvo tutte le variabili e i programmi che ho usato. Vi consiglio vivamente di trovare una vostra organizzazione coerente e cominciare ad archiviare anche le vostre prove in modo coerente. Quando diventate pratici e iniziate delle vostre sperimentazioni, siate funzionali non cronologici, create dunque cartelle che rappresentano sperimentazioni e poi all'interno parti dell'elaborazione, come ad esempio "Analisi Skin Conductance", "Analisi EMG", "Matrici finali con dati di sintesi", "Analisi statistiche" e così via. Una tale organizzazione vi permetterà a colpo d'occhio di avere chiara la situazione delle vostre sperimentazioni.

Scalari, Vettori e Matrici

In questo capitolo iniziamo a lavorare con le strutture di base dei dati. Siano essi di una certa complessità o meno, i dati vengono raccolti, con i metodi visti nella prima parte del volume, in determinate strutture. Queste strutture, come abbiamo visto nel capitolo precedente, vengono gestite con dei programmi capaci di grandi elaborazioni, senza problematiche particolari legate alla dimensione dei dati stessi. All'interno di questi programmi i nostri dati trovano una razionale organizzazione in strutture coerenti con la raccolta effettuata. Avere acquisito alcune nozioni di base dell'algebra lineare, può certamente essere d'aiuto per l'apprendimento dei contenuti di questo capitolo, tuttavia non è indispensabile e descriverò le strutture prescindendo da tali conoscenze. La struttura di base per eccellenza nei dati è lo "scalare": altro non è che semplicemente un numero reale associato ad un'unità di misura. In pratica se a una variabile assegniamo un numero, questa variabile è identificabile come scalare (Figura 8.1).

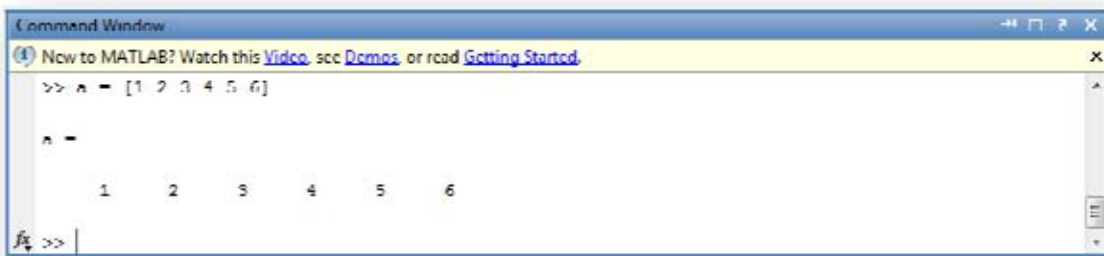
Se a una variabile assegniamo la media di un segnale, questa sarà un numero e ancora una volta uno scalare.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> a = 1
a =
    1
>>
```

Figura 8.1. Assegnazione di un numero (1) a uno scalare (a).

Lo scalare è a tutti gli effetti una variabile in Matlab. Se invece di un solo numero considero una successione di numeri, la variabile associata sarà un vettore. Basterà frapporre degli spazi tra un numero e l'altro e racchiudere la serie tra parentesi quadre, per creare un vettore, come si può vedere in Figura 8.2. In alternativa agli spazi si possono inserire delle virgole ($a = [1,2,3,4,5,6]$).



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> a = [1 2 3 4 5 6]
a =
    1     2     3     4     5     6
>>
```

Figura 8.2. Assegnazione di una successione di numeri a un vettore (a).

Un vettore come quello appena visto viene definito più precisamente vettore riga, in quanto tutti gli elementi sono ordinati in una riga. Esistono anche i vettori colonna, in cui gli elementi saranno uno sotto l'altro. Per creare un vettore colonna



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> a = [1; 2; 3; 4; 5; 6]

a =

     1
     2
     3
     4
     5
     6

f1 >> |
```

Figura 8.3. Assegnazione di una successione di numeri a un vettore colonna (a).

Un vettore riga è sempre convertibile in un vettore colonna e viceversa. Per effettuare questa conversione in algebra lineare si usa l'operazione detta trasposta. In Matlab, basterà mettere un apostrofo dopo la variabile (a') per convertirla temporaneamente in trasposta, per ogni utile uso. Se invece si vuole rendere l'operazione definitiva basta scrivere $a = a'$ e la variabile a verrà riassegnata con la sua trasposta. Questo tipo di assegnazione riflessiva è sempre eseguibile in Matlab, anche con altri operatori algebrici. Ad esempio posso scrivere $a = a + 1$ per aggiungere 1 a tutti gli elementi del vettore a e fare in modo che questo nuovo

vettore venga definito come a al posto del vecchio, come in figura 8.4.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
a =
     1     2     3     4     5     6

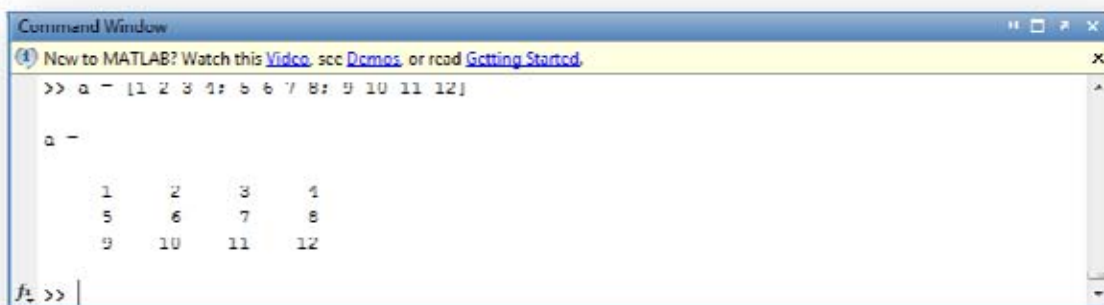
>> a = a + 1

a =
     2     3     4     5     6     7

fx >> |
```

Figura 8.4. Nuova definizione della variabile a come somma di se stessa con il valore 1.

Visti scalari, vettori riga e vettori colonna, la naturale estensione di queste strutture è la matrice, che può essere vista come un accostamento di più vettori riga o se si preferisce di più vettori colonna. E' di fatto un insieme di numeri ordinati riga x colonna.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> a = [1 2 3 4; 5 6 7 8; 9 10 11 12]

a =

     1     2     3     4
     5     6     7     8
     9    10    11    12

fx >> |
```

Figura 8.5. Assegnazione di una matrice righe x colonne.

La matrice in Figura 8.5, ad esempio, si definisce una matrice 3x4, ovvero una matrice con tre righe e quattro colonne.

In Figura 8.6 si può vedere la descrizione completa della sintassi usata per creare una matrice.

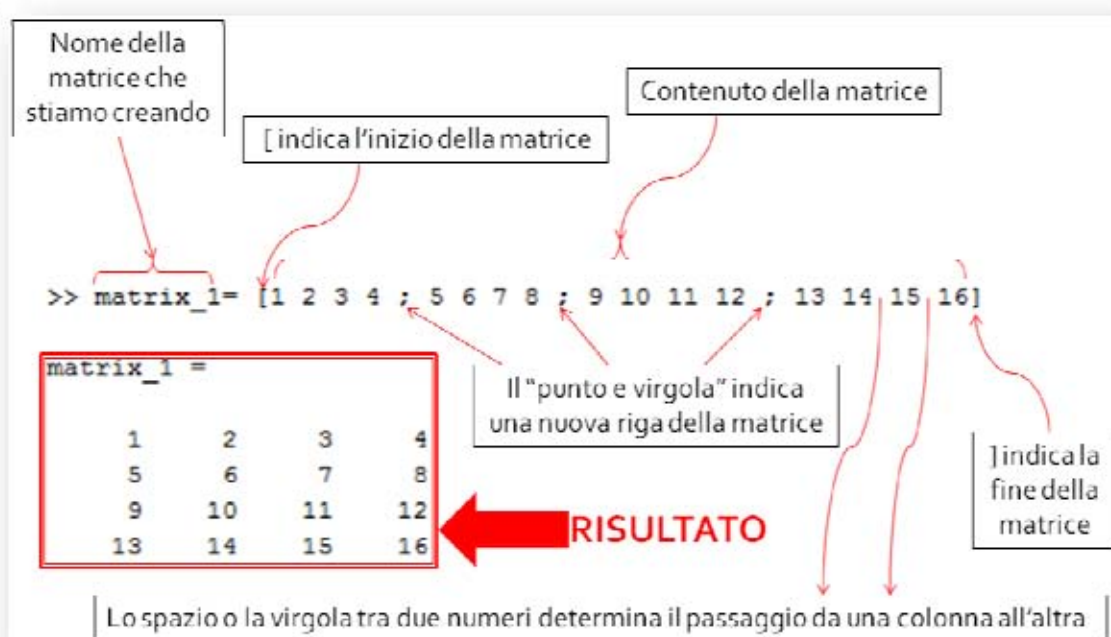
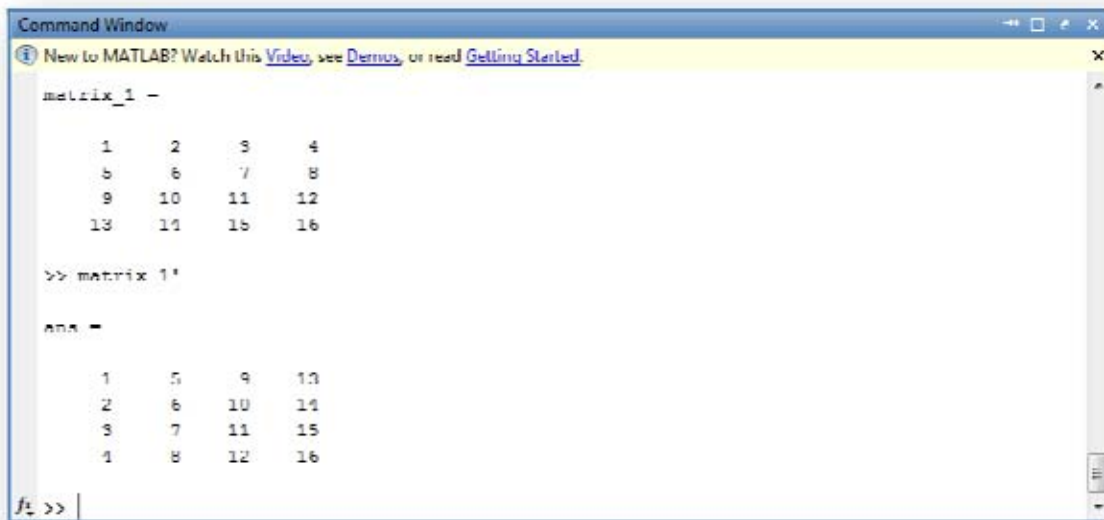


Figura 8.6. Sintassi di creazione di una matrice in righe x colonne.

La sintassi di Matlab va usata in modo esatto. Se è vero che inserire uno spazio in più per questioni estetiche non comporta generalmente problemi ("b=1" è uguale a "b = 1"), è altrettanto vero che sbagliare una virgola in Matlab comporta conseguenze catastrofiche sull'accesso e la manipolazione dei dati, generando errori a catena. E' dunque indispensabile prendere una certa dimestichezza con la sintassi per non sbagliarla frequentemente e,

soprattutto, controllare sempre i risultati prodotti, anche quelli intermedi, per esser certi che si sta procedendo nel modo corretto. L'operazione di trasposizione prima vista per i vettori, vale anche per le matrici e a volte può essere molto utile (Figura 8.7). In pratica si effettua una rotazione sulla diagonale principale (quella costituita, nell'esempio, da 1, 6, 11 e 16).



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
matrix_1 =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16

>> matrix_1'

ans =
     1     5     9    13
     2     6    10    14
     3     7    11    15
     4     8    12    16

f1 >> |
```

Figura 8.7. Trasposta di una matrice di ordine 4 x 4.

La creazione di una matrice è un'operazione assai utile, tuttavia spesso è ancor più necessario orientarsi in una tale struttura, andando a identificare intere righe o colonne o anche solo alcuni numeri dalle coordinate specifiche (ad esempio: riga 2, colonna 3). Per poter effettuare queste semplici operazioni bisogna

innanzitutto aver ben chiaro come è strutturata una matrice e come poter richiamare gli elementi in essa contenuti (Figura 8.8).

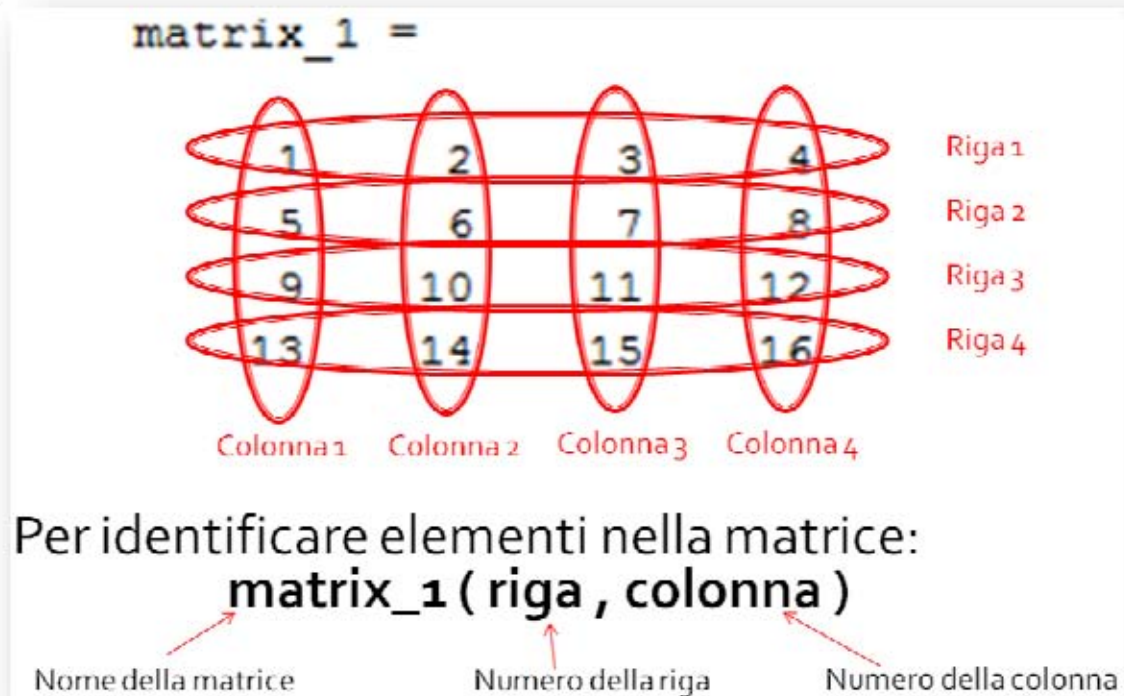


Figura 8.8. Orientarsi in una matrice righe x colonne.

In Matlab scalari, vettori e matrici sono tutte strutture considerate variabili. Inoltre scalari e vettori possono essere viste come delle matrici, infatti:

- lo scalare è una matrice 1x1;
- il vettore riga con n scalari è una matrice 1xn;
- il vettore colonna con n scalari è una matrice nx1.

Possiamo considerare dunque, da qua in avanti, solo matrici.

Seguendo la struttura della matrice, definita in Figura 8.8, possiamo richiamare elementi da qualsiasi matrice, semplicemente dandone le coordinate di riga e di colonna (Figura 8.9).

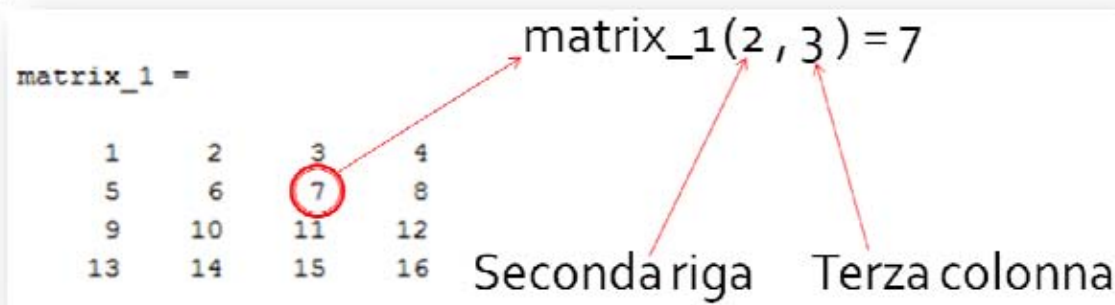
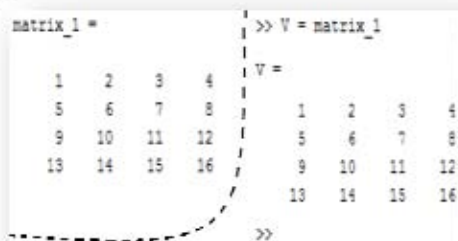


Figura 8.9. Sistema di individuazione di elementi in una matrice per mezzo delle coordinate di riga e colonna.

La sintassi è dunque molto semplice: basta mettere il nome della matrice (ricordando che maiuscole e minuscole sono caratteri diversi per Matlab) e tra parentesi tonde indicare la posizione di riga e di colonna.

Inoltre, se al posto di un numero inseriamo i due punti, indicheremo a Matlab che vogliamo prendere tutti i valori. Questo ci permette anche di trasformare delle matrici in dei vettori riga o



colonna. Possiamo innanzitutto assegnare una nuova variabile (ad esempio V) alla nostra matrice. Poi effettueremo le trasformazioni.

L'assegnazione a una nuova variabile è importante per non perdere la matrice originale (nell'esempio avevamo `matrix_1`). Non abbiamo limiti di variabili in Matlab, dunque è sempre meglio creare una nuova variabile prima di intervenire. In Figura 8.10, possiamo vedere due tipi di trasformazioni della matrice `V` in vettore. Riassegnando la stessa variabile `V` a una parte di se stessa, in pratica, perdiamo l'originale matrice `V` ed è per questo che parliamo di trasformazione e non creazione di una nuova variabile.

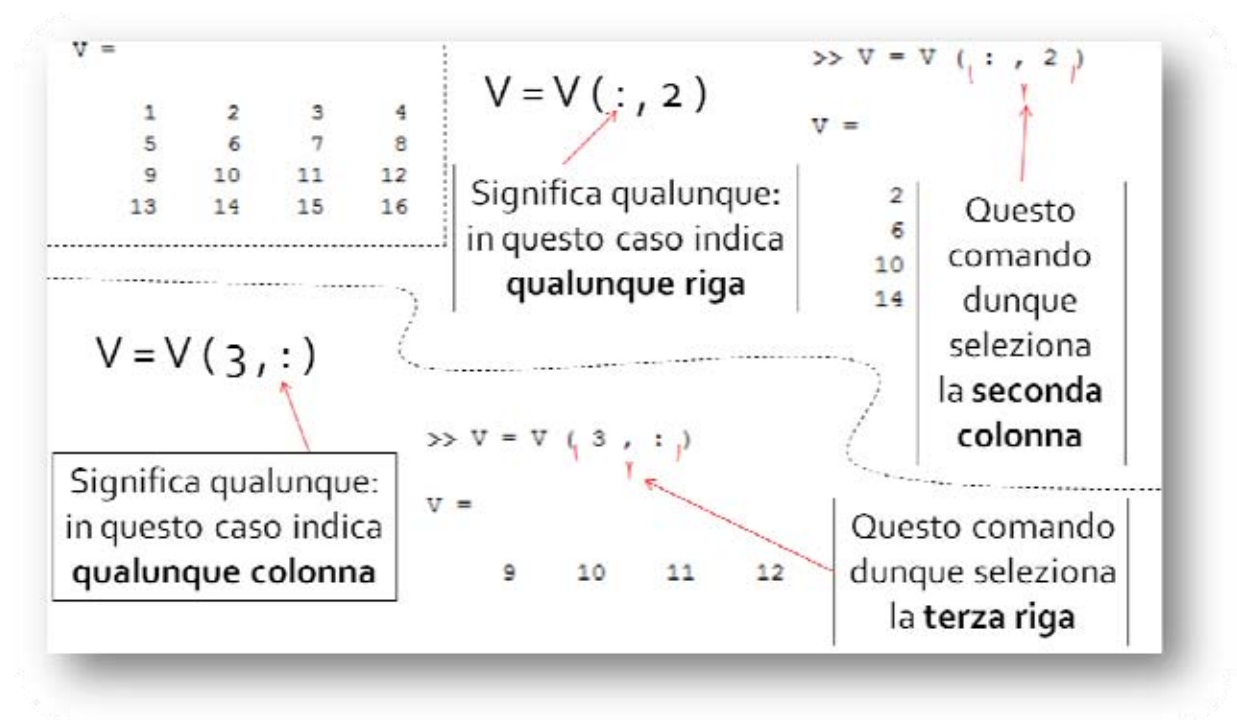


Figura 8.10. Trasformazione della matrice `V` in vettore.

Le due operazioni in Figura 8.10 non possono essere effettuate entrambe, perché una volta trasformata la matrice in vettore riga se prendo il vettore colonna di un vettore riga otterrò uno scalare.

Sembra molto complesso, ma se si fa un po' di pratica diventa tutto molto più chiaro. In particolare se ci si vuole impratichire, suggerisco di creare sempre nuove variabili e passare alle trasformazioni quando si padroneggiano le assegnazioni.

Come abbiamo appena visto, i due punti indicano a Matlab che vogliamo prendere tutti i numeri associati a quella posizione (riga o colonna). Un'altra importante funzione dei due punti nella nostra sintassi la si può dare inserendoli tra due numeri (ad esempio 1:3). In questo caso i due punti indicano tutti i numeri tra il numero che viene prima dei due punti e quello che viene dopo. Dunque se indichiamo 1:3 è come dire 1,2 e 3. Questa notazione può essere inserita sia in riga (ad esempio da riga 1 a riga 3) sia in colonna (da colonna 1 a colonna 3). Vediamo in Figura 8.11 un esempio.

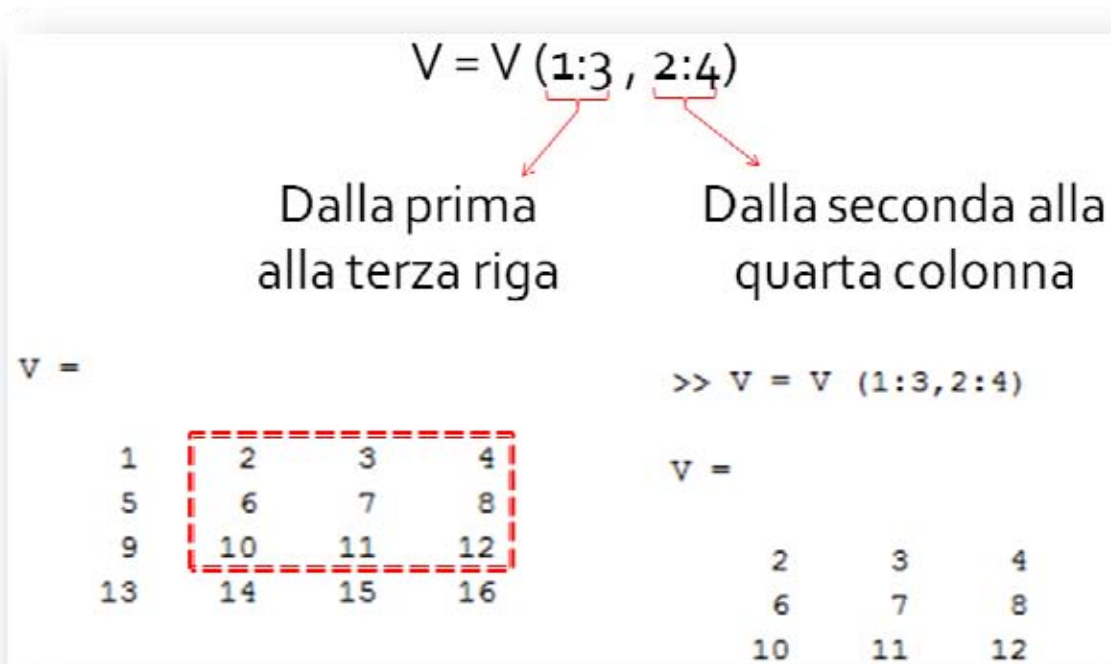
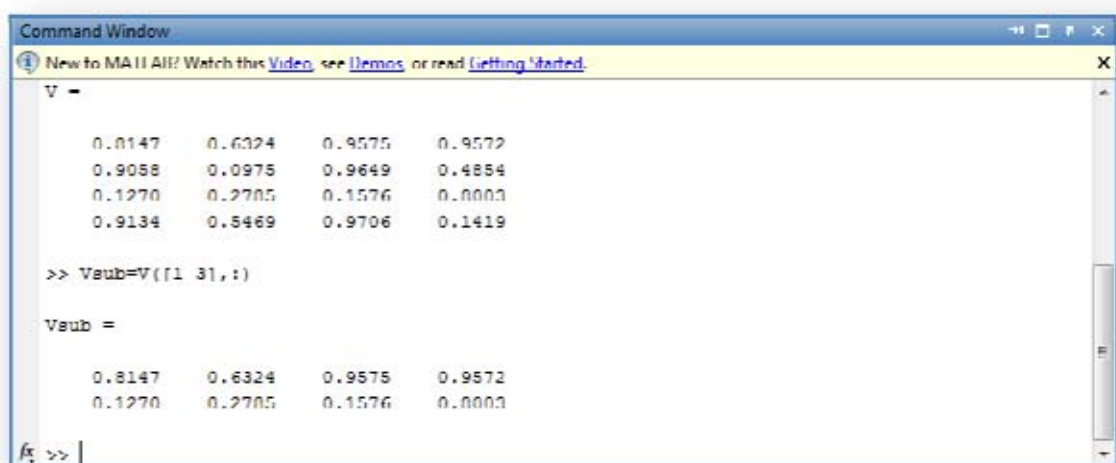


Figura 8.11. Un esempio di selezione di una porzione di matrice.

In figura 8.11, in particolare, abbiamo tagliato la nostra matrice selezionando tre righe (dalla prima alla terza) e tre colonne (dalla seconda alla quarta), costruendo di fatto una sottomatrice, ovvero una matrice che nasce come porzione di dati contigui di una più grande matrice.

Altre importanti operazioni che si possono fare sulle matrici riguardano le selezioni di righe o colonne non contigue. Usando i due punti, tra due numeri, prendiamo tutti gli indicatori di riga o colonna inclusi tra di essi, ma se vogliamo saltare (ad esempio prendere la prima e la terza riga) non possiamo usare i due punti. Per fare questo tipo di selezioni possiamo usare le parentesi quadre, mettendo al loro interno gli indici di riga o di colonna che vogliamo selezionare. Dunque `[1 3]` può essere la prima e la terza riga o la prima e la terza colonna, in base a dove lo posizioniamo. In figura 8.12 creiamo una sottomatrice con le righe 1 e 3.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
V =
    0.8147    0.6324    0.9575    0.9572
    0.9058    0.0975    0.9649    0.4854
    0.1270    0.2785    0.1576    0.0003
    0.9134    0.5469    0.9706    0.1419

>> Vsub=V([1 3],:)

Vsub =
    0.8147    0.6324    0.9575    0.9572
    0.1270    0.2785    0.1576    0.0003
```

Figura 8.12. Selezione di righe non contigue.

Esercizi con le sottomatrici

- $V(1:2, 3:4)$
- $V(1:3, 2:4)$
- $V(:, 1:3)$
- $V(:, 2)$
- $V(:, :)$
- $V(3, :)$
- $V([1\ 3], :)$
- $V([1\ 4], 3:4)$
- $V([1\ 4], [2\ 4])$
- $V(:, [1\ 3\ 2\ 4])$

Definire una matrice V con il comando $V = \text{rand}(4, 4)$ e provare gli esercizi qui a fianco per fare un po' di pratica con le sottomatrici

Figura 8.13. Esercitazione sulla creazione di sottomatrici.

Fare esercizio con le matrici e le operazioni a esse legate è molto importante, sia per acquisire una certa pratica, sia per comprendere operazioni con matrici più complesse. Le operazioni che vediamo qui sono infatti valide sia per una matrice con poche righe e poche colonne, sia per matrici con migliaia o milioni di righe e colonne. Se si è abituati a ragionare su piccole matrici, dove la visualizzazione dei dati è molto semplice, si sarà abituati anche a ragionare su matrici di ordine superiore, perché quel che conta è la

logica di struttura che si segue e non la dimensione della matrice stessa.

In algebra lineare si è soliti studiare alcune matrici canoniche, come le matrici contenenti tutti zero o tutti uno, che potrebbero essere utili anche nella pratica computazionale. In Figura 8.14 vediamo come creare alcune di queste matrici.

- Creare una matrice di zero
 $Z = \text{zeros}(2, 3)$

```
Z =  
  0  0  0  
  0  0  0
```

- Creare una matrice di uno
 $O = \text{ones}(4, 3)$

```
O =  
  1  1  1  
  1  1  1  
  1  1  1  
  1  1  1
```

- Creare una matrice di cinque
 $F = 5 * \text{ones}(4, 3)$

```
F =  
  5  5  5  
  5  5  5  
  5  5  5  
  5  5  5
```

Figura 8.14. Creare le matrici canoniche a valori predefiniti.

Viste le operazioni di base sulle matrici e le loro trasformazioni in sottomatrici, vediamo adesso alcune importanti operazioni che combinano più matrici e vettori con alcuni semplici comandi.

Queste operazioni, seppur semplici, sono essenziali per la psicometria computazionale in quanto permettono una efficace

manipolazione dei dati sperimentali, secondo logiche di coerenza e necessità di disegno sperimentale. E' dunque essenziale impraticarsi con questo tipo di operazioni perché poi le trasleremo su matrici di dati assai più complesse.

La prima operazione che vediamo è l'accostamento di due matrici. Il primo vettore che abbiamo creato (vedi Figura 8.2) era costituito da un insieme di numeri separati da spazi (vi ricordo che, come abbiamo già visto, si possono anche usare le virgole). Come già detto in Matlab i numeri (gli scalari) sono matrici 1x1 e dunque in realtà abbiamo già fatto un accostamento di matrici in orizzontale. Se si usano matrici di tipo $n \times m$ (n righe, m colonne) si effettuerà la stessa operazione, accostando tra parentesi quadre i nomi delle due matrici, separate da spazio o da virgola (Figura 8.15).

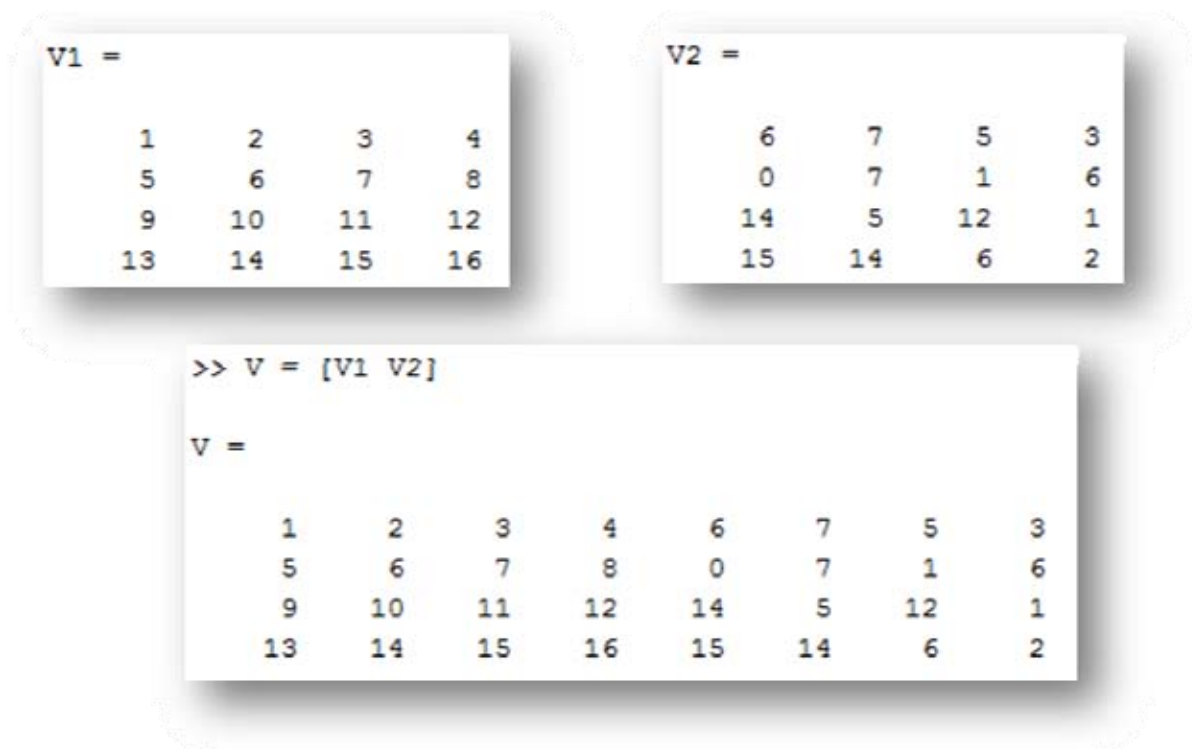


Figura 8.15. Generare una matrice per accostamento di altre due.

Per accostare matrici in verticale, invece, così come abbiamo visto per gli scalari (in Figura 8.3), basterà separare i nomi delle matrici con un punto e virgola (Figura 8.16).

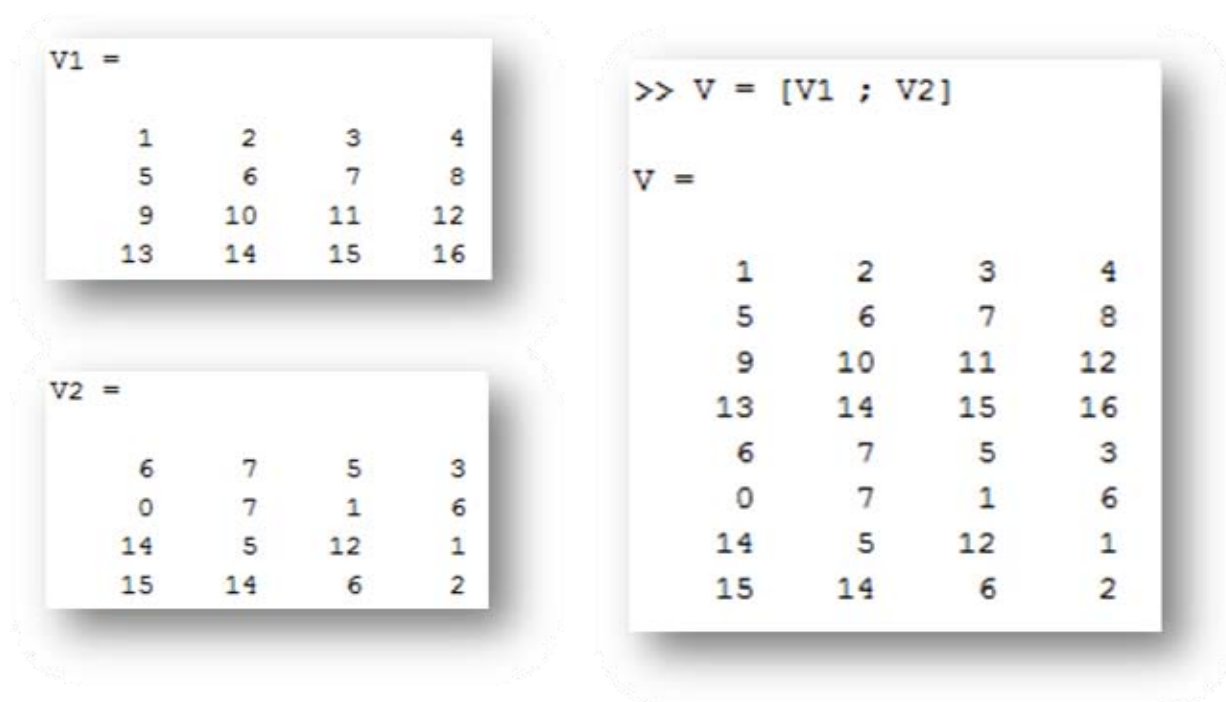


Figura 8.16. Accostamento di due matrici in verticale.

Quanto qui visto per due matrici può essere fatto anche per più matrici e seguendo l'ordine necessario. Si può inoltre anche accostare un vettore (che comunque è una matrice!) a un'altra matrice. L'ordine necessario di cui parlo è l'esigenza del ricercatore, per cui se devo accostare orizzontalmente un vettore Vett1 5x1 (quindi vettore colonna con cinque righe) alla matrice Mat1 5x3 (quindi matrice con cinque righe e tre colonne) potrò scrivere [Vett1, Mat1] o [Mat1, Vett1] in base al fatto che io

necessiti della colonna del vettore a sinistra o a destra della matrice rispettivamente. La matrice che ne risulta sarà dunque diversa e questo aspetto è sempre da tenere in forte considerazione perché potrebbe cambiare il numero della colonna a cui è associata una certa variabile. Quanto detto per un vettore vale anche per le matrici. Dunque nell'esempio di Figura 8.15 si potrà considerare sia $V=[V1, V2]$, sia $V=[V2, V1]$. Così come nell'esempio di Figura 8.16 si potrà considerare sia $V=[V1; V2]$, sia $V=[V2; V1]$. I risultati prodotti, ripeto, saranno differenti: l'ordine conta.

Un altro importante aspetto da considerare sempre, sarà la coerenza di dimensione nell'accostamento. In pratica non posso, ad esempio, accostare una matrice di ordine 3×3 con una di ordine 4×4 . E' sempre necessario verificare che se accosto in orizzontale le righe delle matrici che accosto sono di pari numerosità e che lo siano le colonne se invece accosto le matrici in verticale. In caso contrario Matlab darà errore, emettendo un suono al quale non farete sicuramente mai l'abitudine. Questa operazione di accostamento (orizzontale o verticale) può sembrare molto semplice, così come verificare la coerenza delle dimensioni delle matrici, tuttavia vedremo come questa operazione diventerà assai più complessa quando usiamo migliaia o milioni di dati e diventa impossibile verificare le dimensioni "a occhio" dovendoci invece affidare alla logica di costruzione e alla dimensione delle variabili riportate sul Workspace. Questo è anche uno dei motivi per cui ho spiegato nel capitolo precedente che un'interfaccia grafica, seppur minimale, è utile anche ai più esperti. Chi usa GNU Octave con

l'interfaccia grafica DomaniMathIDE, ha comunque il Workspace a disposizione.

Come abbiamo visto è possibile selezionare intere righe o colonne da una matrice V indicando ad esempio $V(3,:)$ per la terza riga o $V(:,5)$ per la quinta colonna. Nel caso vogliamo eliminare queste righe o queste colonne, basterà assegnare un elemento vuoto (parentesi quadre aperte e chiuse senza nulla in mezzo) in loro corrispondenza. Dunque $V(3,:)=[]$ eliminerà la terza riga e $V(:,5)=[]$ eliminerà la quinta colonna. Questa operazione è molto utile in quanto capita spesso, quando si voglia ripulire una matrice di dati senza creare una nuova variabile, avere la necessità di rimuovere righe o colonne intere, con un semplice comando.

All'inizio del capitolo abbiamo visto come creare un vettore a partire da un insieme di elementi (numeri) dati. A volte però l'esigenza è quella di creare una successione ordinata di numeri seguendo certi criteri aritmetici di base. Ad esempio si può voler creare una successione di numeri naturali da 1 a 5. L'operazione è semplice e basterà scrivere $1:5$ (da 1 a 5), come da Figura 8.17.

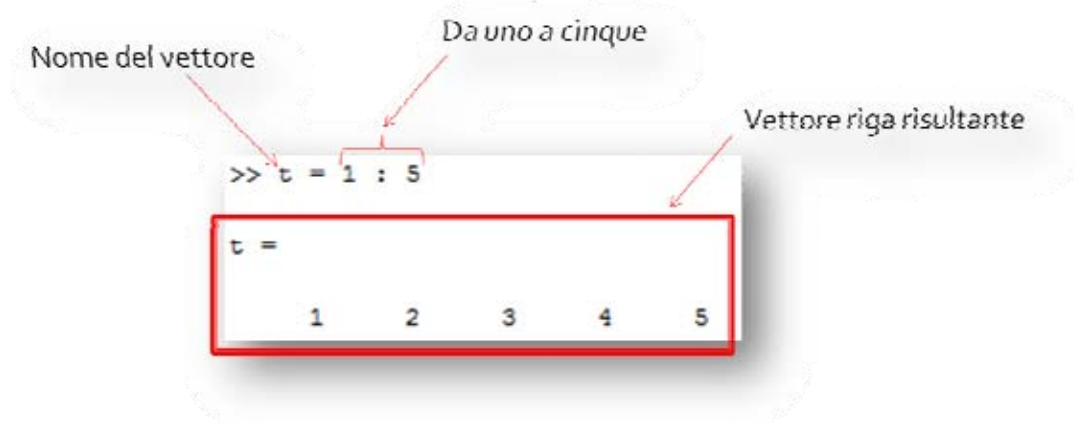


Figura 8.17. Vettore contenente una successione di cinque numeri.

Matlab calcola automaticamente una successione a incremento costante di uno, se non indicato diversamente. Se invece l'esigenza è più complessa e la necessità è ad esempio avanzare di due in due, basterà indicare `2:2:10` per dire da 2 a 10 (valori estremi) avanzando a blocchi di 2 (valore intermedio).

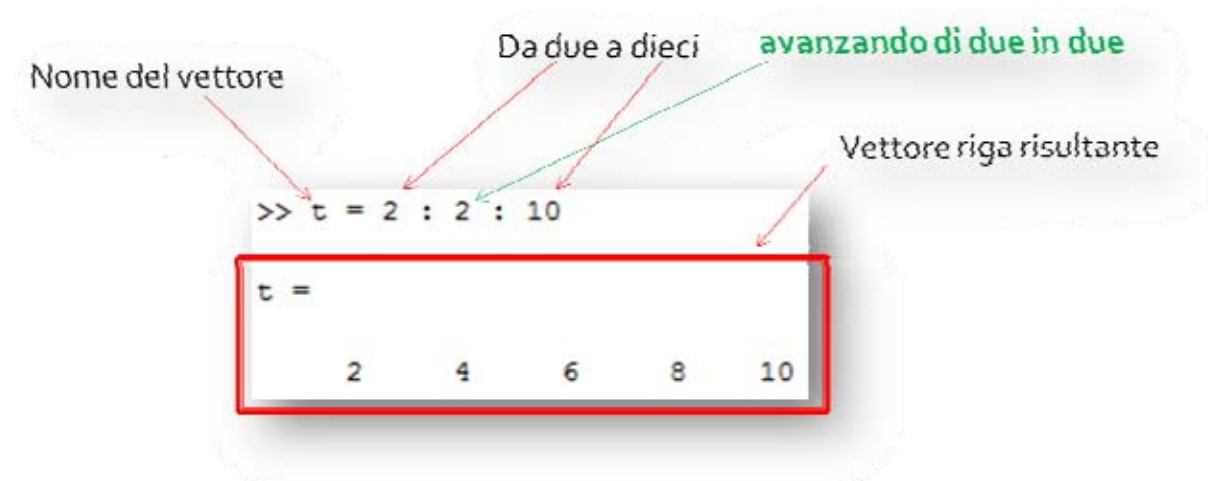
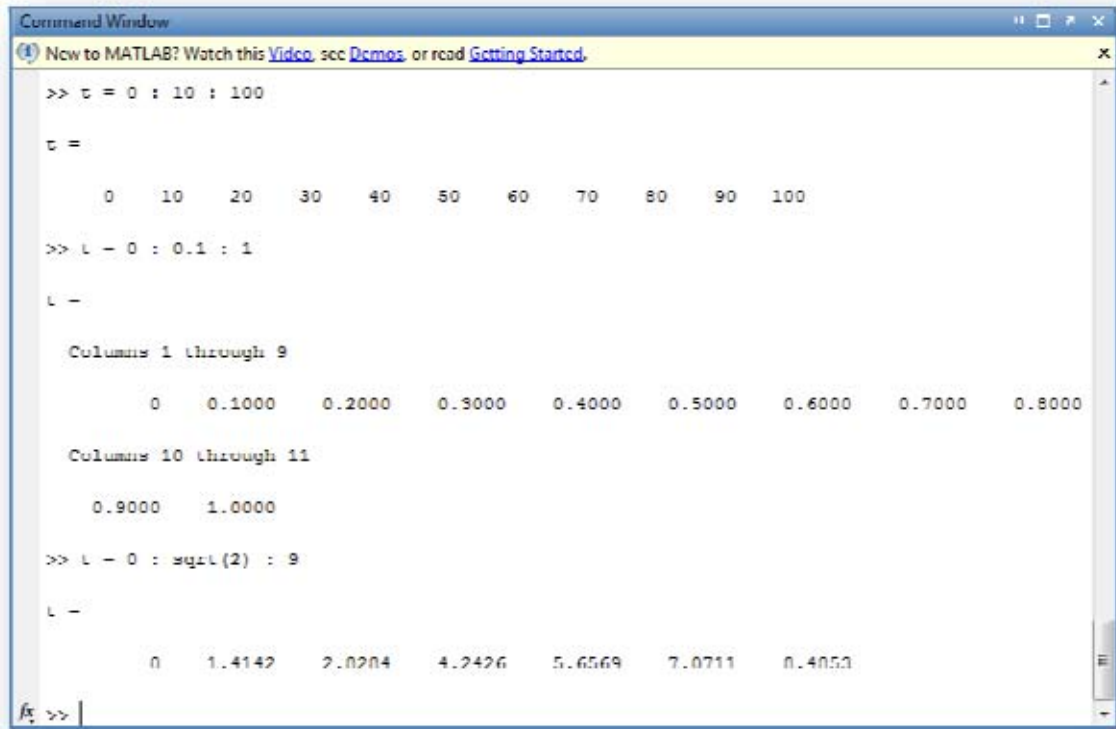


Figura 8.18. Vettore con successione di 5 numeri, da 2 a 10, che avanzano a blocchi di 2.

Da notare che come abbiamo scelto di avanzare a blocchi da 1 (default di Matlab) o blocchi da 2 (con nostra indicazione), è anche possibile scegliere di avanzare ad altri blocchi di interi, razionali o persino per numeri irrazionali, come la radice quadrata di due (vedi Figura 8.19). La generazione di queste sequenze di numeri è importantissima per la gestione dei dati computazionali in quanto, come vedremo, ci permetterà l'aggiunta di vettori che devono seguire regole predefinite, come ad esempio quando vogliamo inserire una colonna dei tempi su milioni di dati.



```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> t = 0 : 10 : 100
t =
    0    10    20    30    40    50    60    70    80    90   100
>> L = 0 : 0.1 : 1
L =
Columns 1 through 9
    0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000    0.7000    0.8000
Columns 10 through 11
    0.9000    1.0000
>> L = 0 : sqrt(2) : 9
L =
    0    1.4142    2.8284    4.2426    5.6569    7.0711    8.4853
fx >> |

```

Figura 8.19. Altre successioni ad avanzamento differenziato.

In Figura 8.19 è anche possibile vedere che quando i dati superano la visualizzazione orizzontalmente disponibile, Matlab inserisce dei commenti ("Columns 1 through 9" e così via). Se però i dati diventano troppi (come scrivendo "t=1:0.1:100"), la visualizzazione diventa scomoda, oltre che inutile. Se si vuole dunque fare in modo che Matlab non visualizzi il risultato di un'espressione, basterà inserire un punto e virgola alla fine della riga e automaticamente il risultato non verrà visualizzato. Se dunque scrivo "t=1:0.1:100;" niente verrà visualizzato e mi risponderà nuovamente il cursore in attesa di altre istruzioni. Se

avete dimenticato il punto e virgola e Matlab sta scorrendo migliaia o milioni di dati, non aspettate che finisca di visualizzare, potete interrompere premendo CTRL+C.

Se in seguito abbiamo necessità di visualizzare i dati in t, basterà cliccare due volte sulla variabile t, che comunque sarà sempre visibile nel Workspace.

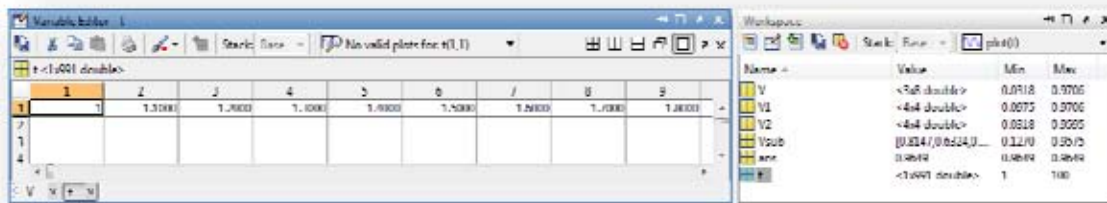


Figura 8.20. Visualizzazione sul Workspace e nell'editor dei dati.

Una volta concluse le varie operazioni sulle variabili ricordiamoci sempre di salvarle sul disco locale, per averle sempre a disposizione per usi futuri. Invece per cancellare tutte le variabili dal Workspace basterà inviare il comando `clear` e tutto verrà cancellato e non sarà più disponibile, a meno di averlo salvato.

8.1 Ristrutturazione dei dati complessi

Finora abbiamo esaminato scalari, vettori e matrici di esempio, per capirne la logica e il funzionamento. Adesso andremo invece a

esaminare dati molto più complessi, basandoci comunque sulle regole e sulle espressioni viste con gli esempi.

Per semplicità, useremo ancora l'esempio con la matrice di dati "data" che abbiamo imparato a importare in Matlab nel Capitolo 7. La matrice è anche disponibile nei materiali supplementari al volume con il nome "1.mat" (1.txt era infatti il nome dal quale abbiamo importato i dati). Innanzitutto cominciamo adesso a spiegare il nome che abbiamo dato al file (1.txt), in quanto anche questo è di una certa importanza per seguire la logica di processo di cui abbiamo parlato. Dal momento che il file contiene dati psicofisiologici di un soggetto, il file "1.txt" sta ad indicare che questi dati sono relativi al primo soggetto di una sperimentazione. In particolare i dati sono relativi alla sperimentazione di cui abbiamo parlato al paragrafo 2.3, nel capitolo 2 e la cui sperimentazione è spiegata sinteticamente in Figura 2.6. Spero che questa continuità nel volume aiuti il lettore a seguire la logica di processo che va dalla raccolta dati alla loro analisi.

Una volta aperto il file "1.mat" dovrete avere nel Workspace la matrice "data". Se avete cambiato il nome (ad esempio in "nome") basterà scrivere "data=nome;" per assegnare alla matrice data gli stessi valori della matrice "nome".

Un'altra notazione importante riguarda la prima riga dei dati. Se osservate bene la Figura 7.15, vedrete che la prima riga contiene dei dati assai diversi dalle altre. In effetti quei numeri rappresentano la descrizione della frequenza di campionamento delle rispettive colonne (che rappresentano i segnali

psicofisiologici raccolti). Dunque, in effetti, si potrebbe rifare l'importazione indicando che le righe di descrizione (di cui in Figura 7.14) sono in realtà 8 e non 7 come in precedenza indicato. Tuttavia abbiamo visto in questo capitolo come rimuovere una riga da una matrice e dunque questa è già una buona occasione per entrare nella pratica computazionale, evitando la lunga procedura di importazione dei dati. Per rimuovere la prima riga della matrice "data" basterà infatti scrivere `data(1,:)=[];` e vedremo nel Workspace che la matrice passerà dalla dimensione 235025x14 a 235024x14 (stesse colonne, ma una riga in meno).

La prima esigenza che vogliamo adesso esplorare su questa matrice di dati è un'operazione che può sembrare molto semplice, ma che in realtà richiede anch'essa una certa riflessione: inserire come prima colonna dei dati il tempo espresso in secondi.

So bene che la tentazione di fare l'operazione su Excel o simili è assai forte, ma proviamo a pensare anche ai limiti discussi in precedenza: oltre 235 mila righe di dati comporta problemi anche solo in apertura, per non considerare poi se si parla di milioni di righe.

La prima operazione che bisogna fare per inserire il tempo e creare un vettore dei tempi. Possiamo cominciare scrivendo

```
t=1: 235024;
```

in modo da avere l'indicatore per ogni riga. Per semplificarci la vita possiamo anche usare un'espressione in Matlab che estrae

automaticamente la lunghezza di una matrice: `length`. Possiamo dunque scrivere in effetti

```
t=1:length(data);
```

dove `length(data)` è il numero che identifica il numero di righe nella matrice `data`. Il vettore (riga) `t` sarà della stessa lunghezza della matrice `data` e conterrà la sequenza: 1 2 3 ... 235024 (numero di righe di `data`).

Per inserire il tempo come colonna nella matrice `data` dobbiamo trasformare il vettore riga `t` in un vettore colonna. Basterà fare la trasposta "`t=t'`";" come visto in precedenza. Adesso è sufficiente accostare la colonna tempo (data dal vettore `t`) alla matrice dei dati (`data`):

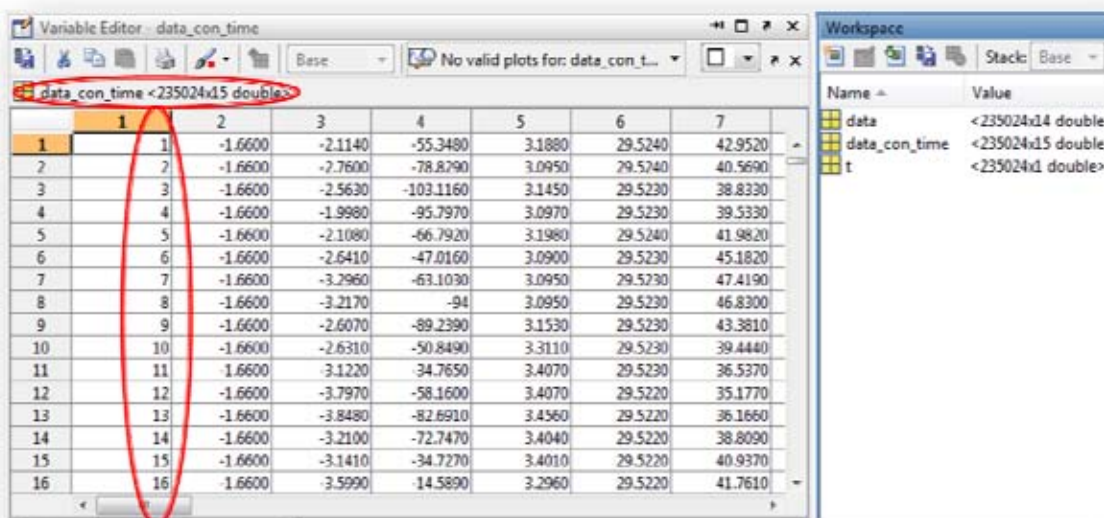
```
data_con_time = [t data];
```

con `t` a sinistra della matrice `data`, in modo da avere il tempo come prima colonna. Notare che se la `skin conductance` negli esercizi precedenti era la quinta colonna, adesso sarà dunque la sesta.

Come si può vedere dall'espressione usata, cercheremo sempre di nominare le nuove matrici con nomi che ci possano aiutare anche a distanza di tempo a identificare le differenze tra le varie variabili.

Cercheremo anche di creare nuove matrici ogni volta che le modifiche sono tali da cambiare il senso dei dati ed ha un qualche senso tenere la variabile originale.

Il risultato dell'inserimento della colonna è sempre visibile nel Workspace, dove si possono anche vedere le tre variabili finora create (le due matrici e il vettore dei tempi).



Variable Editor: data_con_time

Workspace

Name	Value
data	<235024x24 double>
data_con_time	<235024x15 double>
t	<235024x1 double>

	1	2	3	4	5	6	7
1	1	-1.6600	-2.1140	-55.3480	3.1880	29.5240	42.9520
2	2	-1.6600	-2.7600	-78.8790	3.0950	29.5240	40.5690
3	3	-1.6600	-2.5630	-103.1160	3.1450	29.5230	38.8330
4	4	-1.6600	-1.9980	-95.7970	3.0970	29.5230	39.5330
5	5	-1.6600	-2.1080	-66.7920	3.1980	29.5240	41.9820
6	6	-1.6600	-2.6410	-47.0160	3.0900	29.5230	45.1820
7	7	-1.6600	-3.2960	-63.1030	3.0950	29.5230	47.4190
8	8	-1.6600	-3.2170	-94	3.0950	29.5230	46.8300
9	9	-1.6600	-2.6070	-89.2390	3.1530	29.5230	43.3810
10	10	-1.6600	-2.6310	-50.8490	3.3110	29.5230	39.4440
11	11	-1.6600	-3.1220	-34.7650	3.4070	29.5230	36.5370
12	12	-1.6600	-3.7970	-58.1600	3.4070	29.5220	35.1770
13	13	-1.6600	-3.8480	-82.6910	3.4560	29.5220	36.1660
14	14	-1.6600	-3.2100	-72.7470	3.4040	29.5220	38.8090
15	15	-1.6600	-3.1410	-34.7270	3.4010	29.5220	40.9370
16	16	-1.6600	-3.5990	-14.5890	3.2960	29.5220	41.7610

Figura 8.21a. La nuova matrice "data_con_time" contenente la nuova colonna dei tempi.

Adesso però dobbiamo interrogarci sul dato appena inserito e verificare se rappresenta veramente quello che volevamo. In effetti la prima colonna, così come è adesso, rappresenta l'evoluzione dei vari segnali (uno per colonna) passo dopo passo (Figura 8.21a). Ma un passo non è un secondo, in quanto i dati psicofisiologici raccolti (si veda il capitolo 2), erano campionati a 256 Hz e proprio per questo generavano un così ampio numero di righe, pur essendo relativi a soli circa quindici minuti di registrazione di dati.

E' dunque necessario, per esprimere il tempo in secondi, considerare che un secondo corrisponde a 256 righe e quindi una riga corrisponde a un duecentocinquantesimo di secondo. Il tempo in secondi che esprime dunque il passaggio da una riga all'altra è $1/256$. Bisognerà pertanto dividere per 256 ogni elemento della prima riga, in modo che la riga 1 sarà in effetti $1/256$, la riga 2 sarà $2/256$, la riga 256 sarà 1 ($=256/256$) e così via. In tal modo il tempo espresso dalla prima colonna sarà in secondi. Per modificare la prima colonna della matrice "data_con_time" useremo "data_con_time(:,1)" e con la seguente espressione cambieremo tutti i valori della prima colonna:

```
data_con_time(:,1)=data_con_time(:,1)/256;
```

A questo punto la prima colonna avrà i tempi espressi in secondi da 0.0039 ($=1/256$) a 918.0625 ($=235024/256$) (Figura 8.21b).

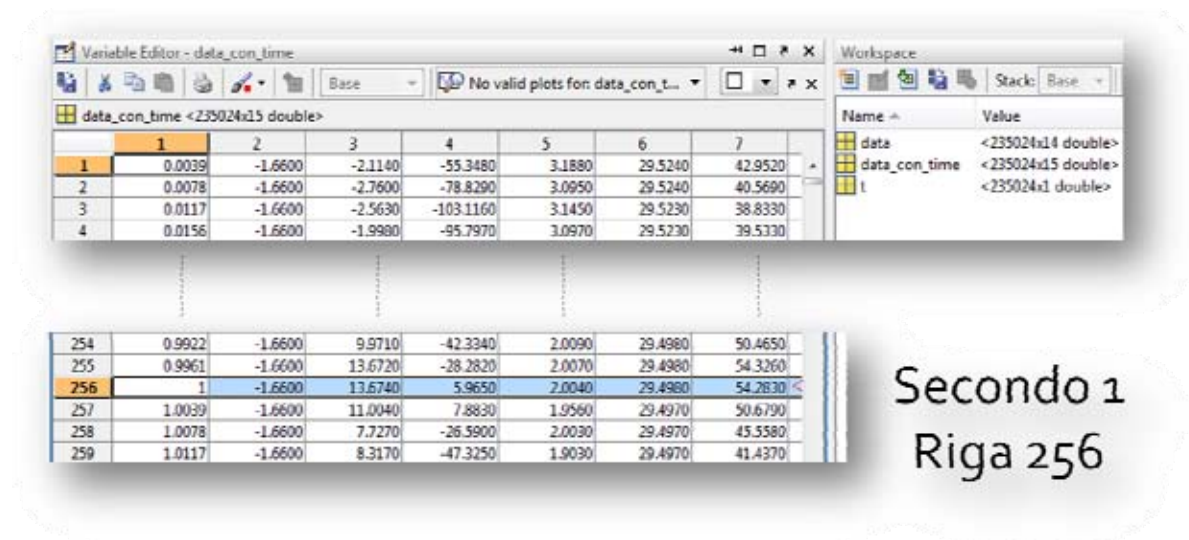


Figura 8.21b. La matrice con il tempo espresso in secondi.

Un'alternativa a questi passaggi poteva essere creare il vettore dei tempi già in secondi. Invece di scrivere "t=1:length(data);" potevamo scrivere l'espressione (commentata) in Figura 8.22.

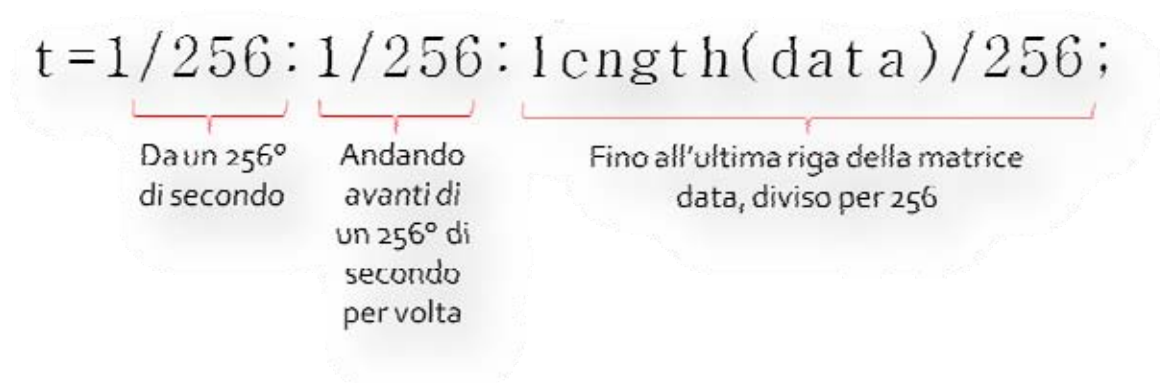


Figura 8.22. Un'alternativa per creare i tempi già in secondi.

Inoltre, se creiamo le seguenti due nuove variabili:

```
L=length(data);
s=256;
```

allora la precedente espressione

```
t=1/256:1/256:length(data)/256;
```

Può essere riscritta come segue:

```
t=1/s:1/s:L/s;
```

Questo modo di procedere per sintetizzare i dati e renderli generalizzabili può essere molto utile. Tuttavia il numero di espressioni da eseguire in sequenza sta diventando elevato e può essere comodo dunque avere tutto in un unico file (detto "editor") in cui le varie espressioni vengono eseguite tutte insieme. Vediamo come creare questo file in Matlab.

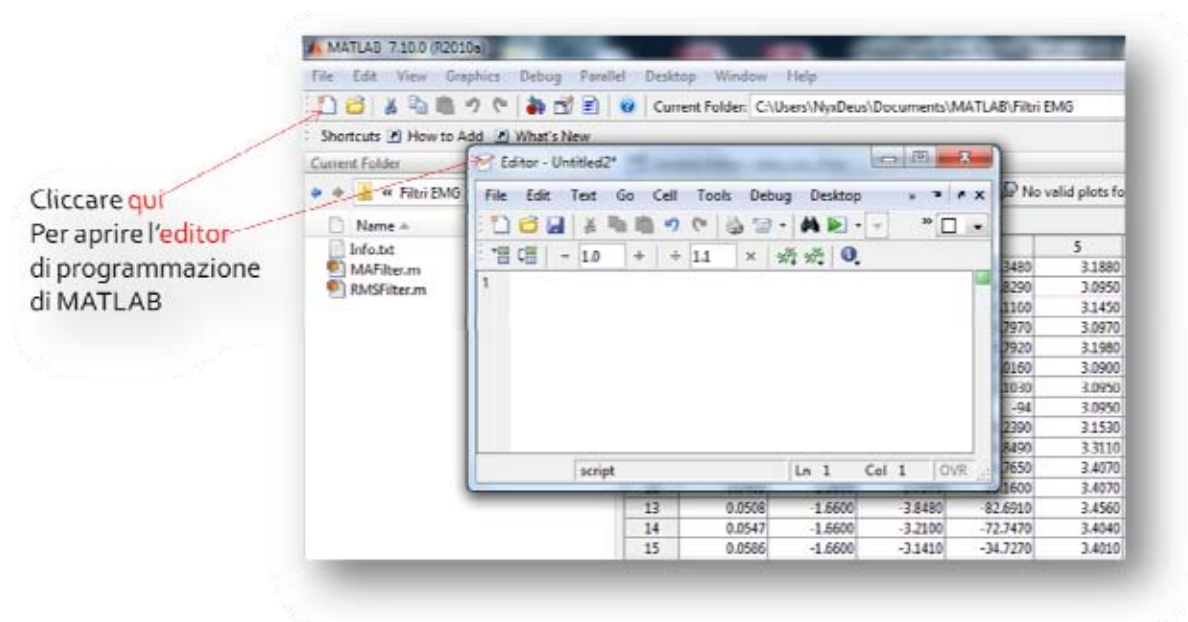


Figura 8.23. Creare un nuovo file per contenere espressioni in sequenza, nell'editor di Matlab.

Nell'editor possiamo inserire il codice in sequenza, come se si dovesse eseguire un'operazione dopo l'altra. Inoltre è possibile commentare le righe usando il simbolo %. Tutto ciò che si scrive dopo % non sarà eseguito e verrà visualizzato nell'editor in verde (proprio per indicare che è un commento).

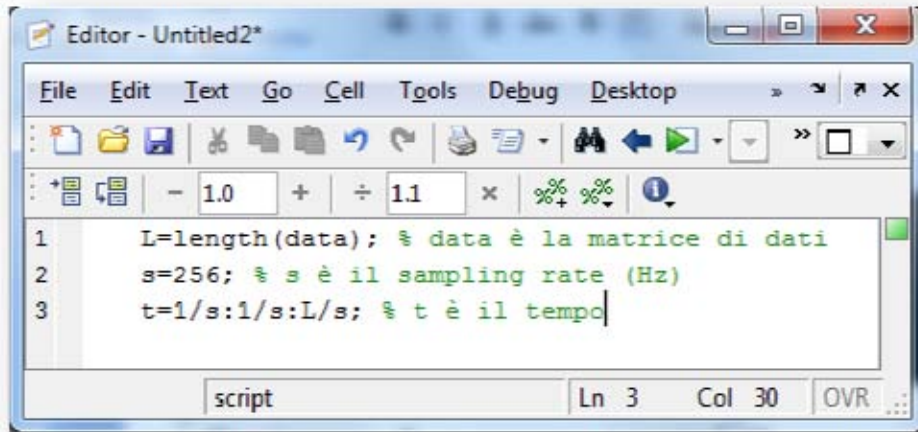


Figura 8.24. Creazione della colonna dei tempi con l'editor.

Mettere le varie espressioni nell'editor presenta molti vantaggi e più avanti si va con la psicometria computazionale, più ci si rende conto che questo è l'approccio corretto.

Tra i vantaggi nell'usare l'editor, vale la pena ricordare i seguenti:

- nell'editor è possibile inserire quante espressioni vogliamo in sequenza assegnando variabili man mano che procediamo con la scrittura del codice;
- tutto ciò che scriviamo nell'editor può essere salvato e resta disponibile anche per future elaborazioni;
- è possibile commentare le righe e tenere traccia di ciò che si sta facendo e di ciò che si vuole ricordare;
- il file "nome.m" che andiamo a creare diventa un'istruzione eseguibile di Matlab e basterà dunque digitare nome come espressione per avviare il codice;
- i file .m sono apribili ed eseguibili anche con GNU Octave.

Salvare il codice in un file .m è un'operazione molto semplice, come si può vedere in Figura 8.25.

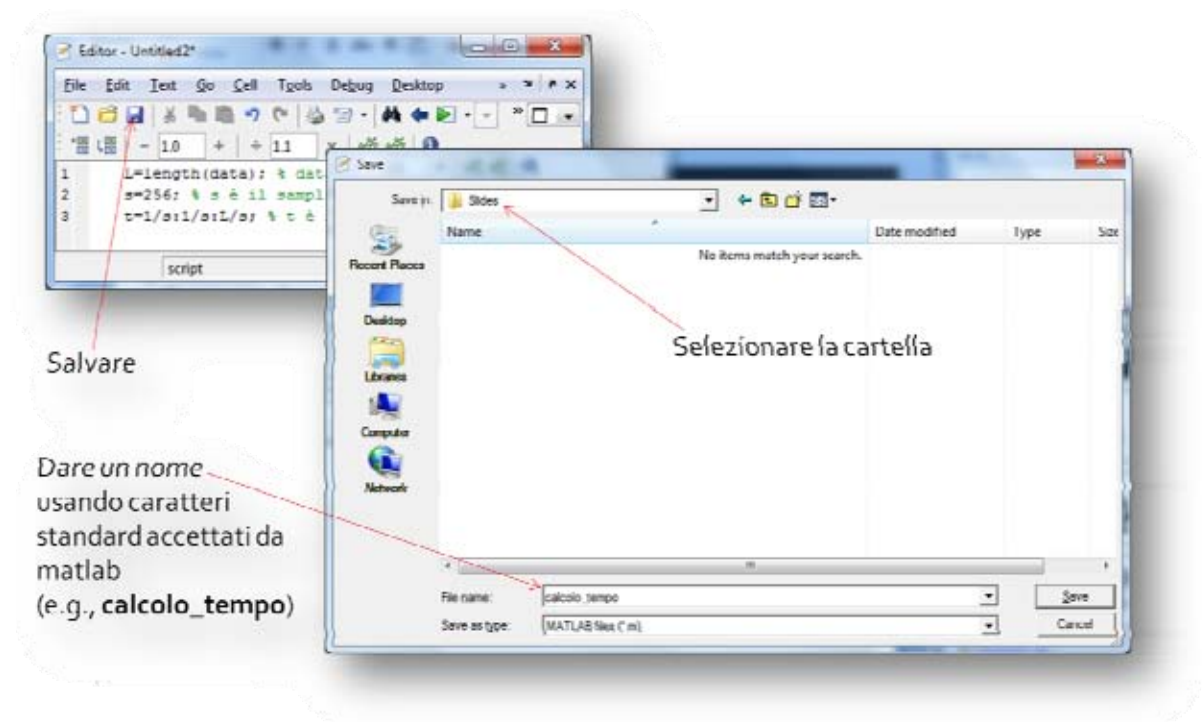
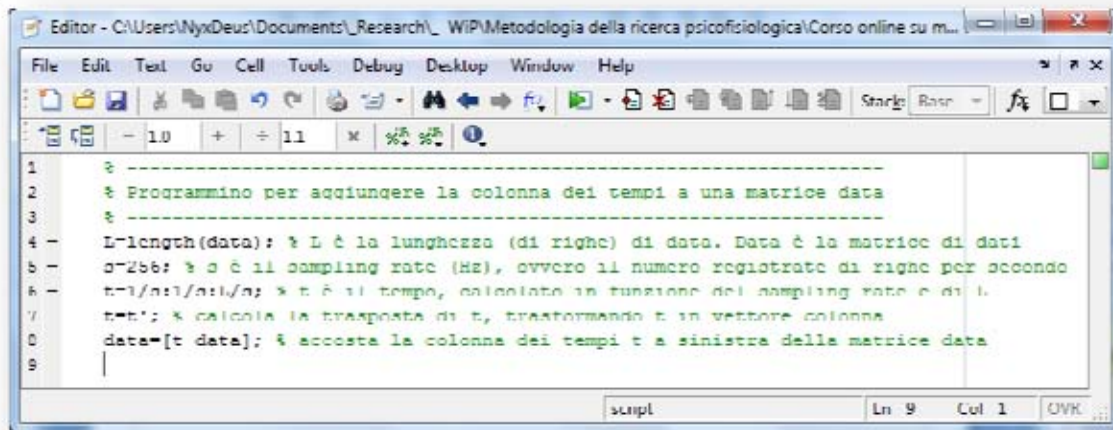


Figura 8.25. Salvataggio come file "calcolo_tempo.m".

Una volta salvato, il file .m, può essere sempre integrato e/o modificato in base alle esigenze specifiche.

Inoltre se si ha avuto l'accortezza di creare degli scalari per le misure che potrebbe essere importante cambiare in futuro, avremo già pronto un determinato codice per sempre. Ad esempio sapendo che la frequenza di campionamento cambia in base ai dati che devo importare, posso impostare un parametro s (sta per sampling rate) a 256 e scrivere tutto il codice usando s quando serve. Allo stesso modo la lunghezza della matrice $data$, è naturalmente variabile e

dunque posso assegnare un parametro L (length) e inserire un'istruzione " $L=\text{Length}(\text{data})$ ". In questo modo potrò fare un codice molto flessibile, che sia usabile anche con altre matrici data a meno di cambiare solo qualche parametro di riferimento. In Figura 8.26, il programma completo che prende la matrice "data" e vi aggiunge la prima colonna di tempi espressi in secondi.



```

1 % -----
2 % Programmino per aggiungere la colonna dei tempi a una matrice data
3 % -----
4 L=length(data); % L è la lunghezza (di righe) di data. Data è la matrice di dati
5 s=256; % s è il sampling rate (Hz), ovvero il numero registrate di righe per secondo
6 t=1/s:1/s:1/s; % t è il tempo, calcolato in funzione del sampling rate e di L.
7 t=t'; % calcola la trasposta di t, trasformando t in vettore colonna
8 data=[t data]; % accosta la colonna dei tempi t a sinistra della matrice data
9

```

Figura 8.26. Integrazione dei dati mancanti nel programma e salvataggio come "add_time_to_data.m".

Basterà importare la matrice "data" e scrivere "add_time_to_data", ovvero il nome del file, che trovate ovviamente nel materiale supplementare (Figura 8.27).

Se importo sul Workspace un'altra matrice che chiamo data2 contenente parametri psicofisiologici campionati a 512 Hz, basterà cambiare nel nostro programma il nome della matrice (da data a data2) e il parametro s , da 256 a 512.

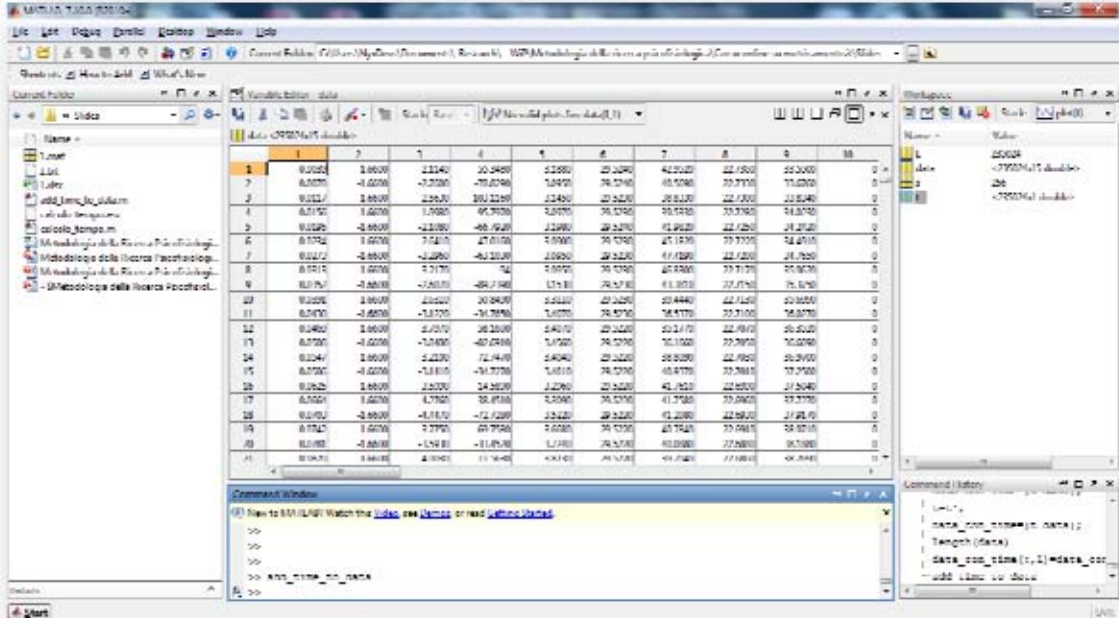


Figura 8.27. Esecuzione del file "add_time_to_data.m" in Matlab.

E' importante che si inserisca il codice in una cartella (possibilmente la stessa dove stanno i dati, ad esempio c:\DATA\) e che si aggiunga questa cartella e tutte le sottocartelle al path (percorso) di Matlab (basta andare su "File > Set Path..."), o altrimenti si dovrà essere sullo stessa cartella del file .m (come si può vedere nel riquadro di sinistra in figura 8.27).

Chiudiamo questo paragrafo con una selezione dei dati che corrisponde, in pratica, alla creazione di una sottomatrice, che abbiamo visto all'inizio di questo capitolo.

Ipotizziamo in particolare di volere selezionare i dati sulla base di due tempi sperimentali (praticamente: inizio e fine della sperimentazione che abbiamo fatto).

Definiamo i tempi come segue:

```
Start=50; % partenza a 50 secondi  
End=850; % fine a 850 secondi
```

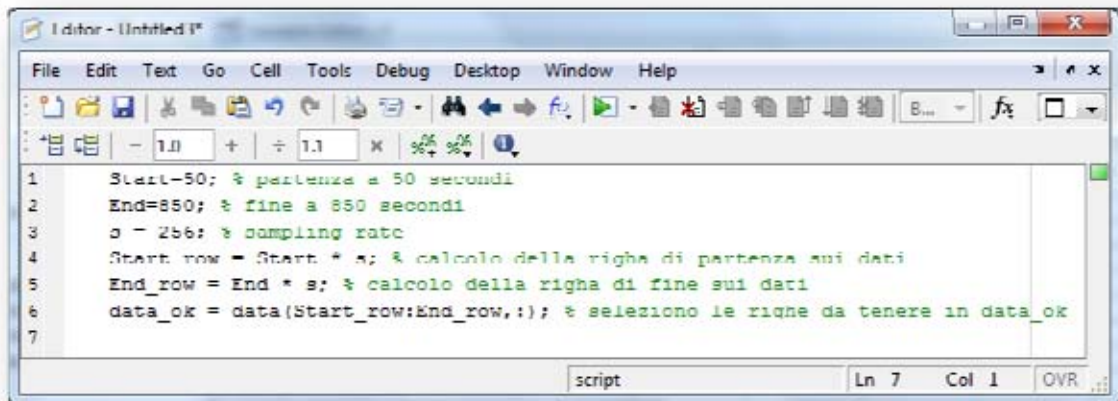
Vogliamo creare una matrice "data_ok" che consideri solo i dati della solita matrice "data" tra il tempo di inizio (start) e il tempo di fine (end).

In pratica abbiamo fatto una sperimentazione e abbiamo registrato I dati psicofisiologici per più di novecento secondi. Abbiamo però accuratamente misurato quanti secondi sono passati dall'inizio della registrazione all'inizio della sperimentazione (50 secondi esatti). Sappiamo anche che la nostra sperimentazione dura esattamente 800 secondi e dunque si è terminata esattamente 850 secondi dopo l'inizio della registrazione.

Per prima cosa convertiamo i secondi in righe della matrice data (considerando il sampling rate) e definiamo le variabili "Start_row" ed "End_row". Per effettuare questa operazione dobbiamo in pratica moltiplicare i secondi per il numero di righe, in modo che così otteniamo l'esatto numero di riga dal quale partire (nell'esempio $50 \times 256 = 12800$) e l'esatta riga a cui dobbiamo arrivare (nell'esempio $850 \times 256 = 217600$).

Dopo definiamo la nuova matrice "data_ok" tagliandola (con l'istruzione vista per creare le sottomatrici) da "Start_row" ad "End_row". Il codice per generare la matrice "data_ok" può essere

visto in Figura 8.28 ed è disponibile nei materiali supplementari al volume con il nome "genera_data_ok.m".



```

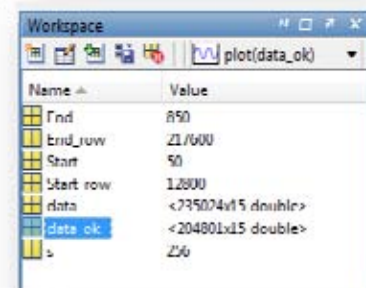
1 Start=50; % partenza a 50 secondi
2 End=850; % fine a 850 secondi
3 s = 256; % sampling rate
4 Start_row = Start * s; % calcolo della righe di partenza sui dati
5 End_row = End * s; % calcolo della righe di fine sui dati
6 data_ok = data(Start_row:End_row,:); % selezione le righe da tenere in data_ok
7

```

Figura 8.28. Taglio della matrice "data" sui tempi sperimentali.

In pratica, quando abbiamo registrato i segnali psicofisiologici abbiamo preso il tempo intercorso prima della sperimentazione (50 secondi) e abbiamo continuato a registrare dei dati anche dopo gli 850 secondi passati dall'inizio della registrazione, ovvero a sperimentazione finita.

Sul Workspace di Matlab appariranno le nuove variabili definite (Start_row, Start, End_row, End) e la nuova matrice "data_ok" che sarà ovviamente più corta di "data". Infatti "data_ok" sarà 204801 righe (800x256+1), ovvero gli 800 secondi della sperimentazione.



Name	Value
End	850
End_row	21/000
Start	50
Start_row	12800
data	<735074x15 double>
data_ok	<204801x15 double>
s	256

8.2 Definizione delle epoche

Riprendiamo adesso i dati della sperimentazione descritta nel capitolo 2, al paragrafo 2.3, con due sessioni sperimentali: una di relax e una di stress. Abbiamo registrato anche una baseline di 200 secondi. La sessione di relax dura 300 secondi, la sessione di stress dura anch'essa 300 secondi (Figura 8.29).

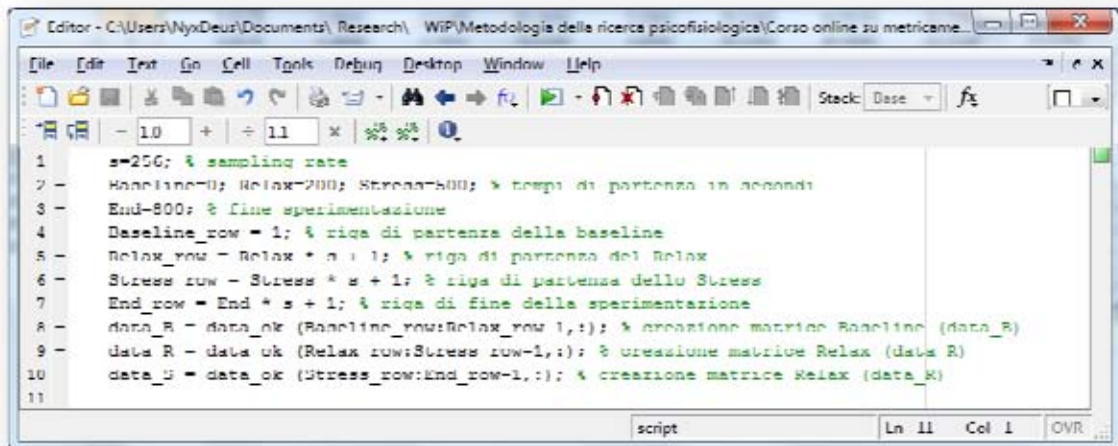


Figura 8.29. Definizione delle epoche sperimentali.

Il totale della sessione sperimentale è dunque 800 secondi, esattamente il periodo che abbiamo tagliato al paragrafo precedente ottenendo la matrice "data_ok".

Una volta che sono state chiaramente definite le epoche (nell'esempio corrente: Baseline, Relax e Stress) si possono creare tre matrici che chiameremo "data_B", "data_R" e "data_S", facendo riferimento (anche nel nome) alle tre epoche sperimentali. L'operazione di creazione di queste matrici consiste in tagli esatti della matrice "data_ok", come possiamo vedere dal codice in

Figura 8.30 e nel programma "genera_epoche.m" nel materiale di supporto del volume.



```

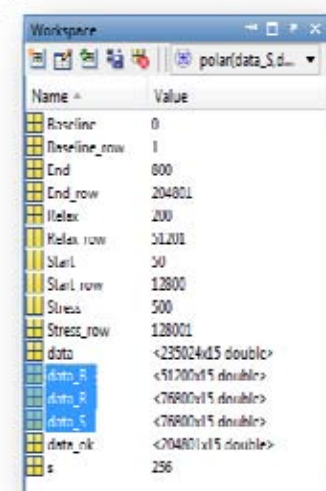
1  s=256; % sampling rate
2  - Baseline=0; Relax=200; Stress=500; % tempi di partenza in secondi
3  - End=800; % fine sperimentazione
4  Baseline_row = 1; % riga di partenza della baseline
5  Relax_row = Relax + s + 1; % riga di partenza del Relax
6  Stress_row = Stress * s + 1; % riga di partenza dello Stress
7  End_row = End * s + 1; % riga di fine della sperimentazione
8  data_B = data_ok (Baseline_row:Relax_row 1,:); % creazione matrice Baseline (data_B)
9  data_R = data_ok (Relax_row:Stress_row-1,:); % creazione matrice Relax (data_R)
10 data_S = data_ok (Stress_row:End_row-1,:); % creazione matrice Relax (data_R)
11

```

Figura 8.30. Programma "genera_epoche.m" per la generazione delle tre matrici ciascuna contenente l'epoca sperimentale associata (Baseline, Relax o Stress).

Sul Workspace di Matlab appariranno le nuove variabili definite e le matrici "data_B", "data_R" e "data_S",

Abbiamo così creato una matrice per ogni epoca definita. Questa operazione è molto utile in quanto ci consente di avere tre diverse strutture per esaminare indipendentemente l'una dall'altra le tre epoche sperimentali ed è anche utile per definire dei valori di sintesi separati per ogni matrice considerata.



Name	Value
Baseline	0
Baseline_row	1
End	800
End_row	204801
Relax	200
Relax_row	21201
Stress	500
Stress_row	128001
data	<235024x15 double>
data_B	<51700x15 double>
data_R	<76800x15 double>
data_S	<76800x15 double>
data_ok	<704801x15 double>
s	256

La dimensione delle matrici deve corrispondere ai secondi delle Epoche, ovvero $51200 / 256 = 200$ secondi per la sessione di Baseline e $76800 / 256 = 300$ secondi per la sessione di Relax e per quella di Stress, come è sempre possibile vedere dal Workspace. le tre matrici sono anche riportate in Figura 8.31, come da visualizzazione in formato dati nel variable editor di Matlab. Avere inserito il tempo, torna adesso molto utile come elemento di verifica, infatti le tre matrici partono sempre dalla riga 1, ma è possibile vedere il tempo di partenza effettiva dalla prima colonna.

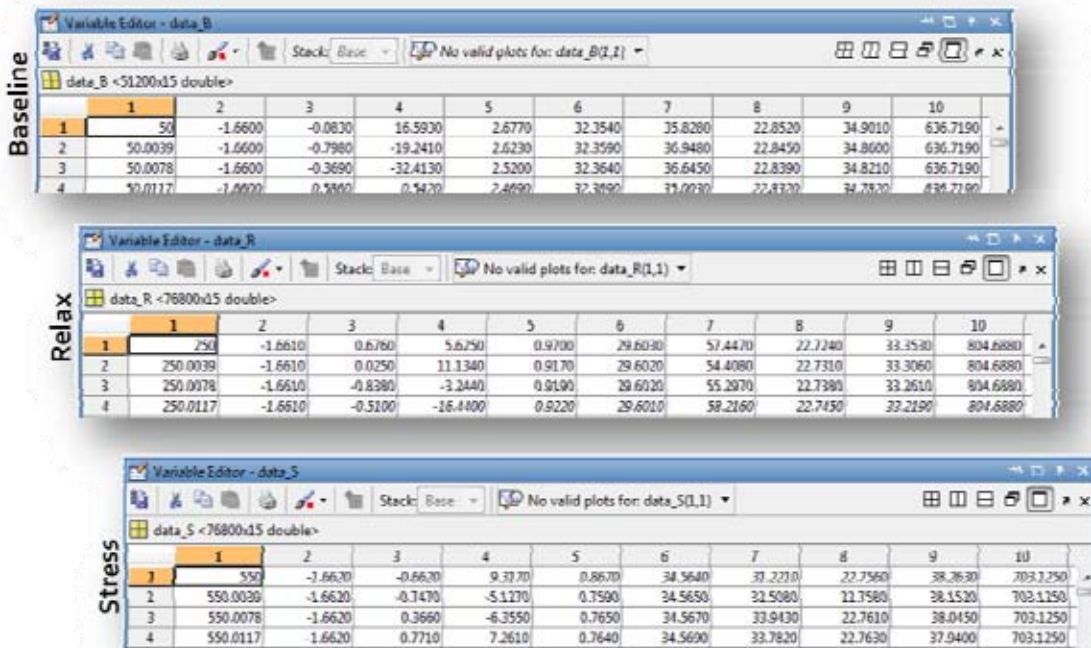
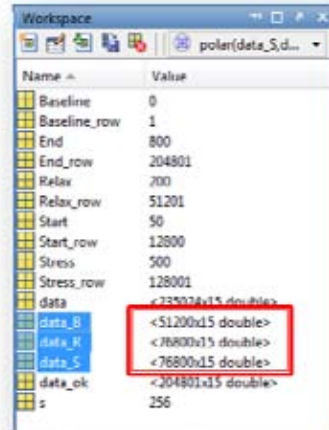
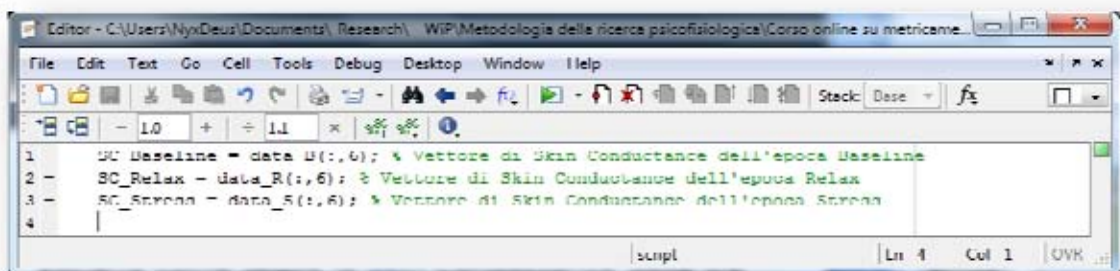


Figura 8.31. Le tre matrici con la colonna dei tempi (in secondi) a indicare l'effettiva partenza della sperimentazione.

Ogni colonna della matrice `data` rappresenta un segnale estratto dall'equipment usato per la registrazione dei segnali psicofisiologici. Sapendo ad esempio che la sesta colonna corrisponde al canale che ha convogliato i dati dal sensore di Skin Conductance (era la quinta, ma come abbiamo visto è diventata la sesta dopo aver inserito il tempo come prima colonna), possiamo estrarre le relative informazioni di sintesi dalla nostra matrice.

Ad esempio nel nostro caso la matrice `data` ha 15 colonne, di cui la prima l'abbiamo precedentemente calcolata noi (è il tempo). La sesta colonna è la Skin Conductance e corrisponde al quinto canale estratto dall'equipment (Canale E). E' dunque indispensabile tenere sempre traccia scritta delle colonne e dei corrispondenti canali.

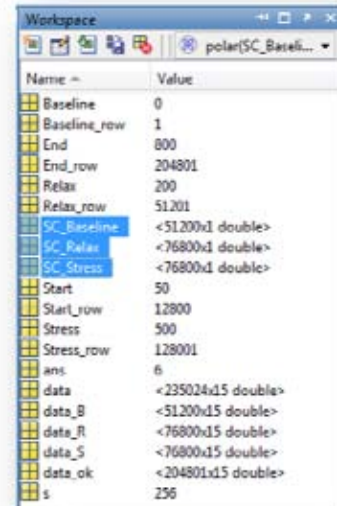
Dal momento che abbiamo già diviso il segnale in epoche e che siamo dunque intervenuti solo sulle righe della matrice e non sulle colonne, possiamo estrarre le Skin Conductance delle tre epoche come da programma in Figura 8.32, riportato nel materiale supplementare al volume come file "`genera_vettori_SCs.m`".



```
1 SC_Baseline = data(:,6); % Vettore di Skin Conductance dell'epoca Baseline
2 SC_Relax = data_R(:,6); % Vettore di Skin Conductance dell'epoca Relax
3 SC_Screen = data_5(:,6); % Vettore di Skin Conductance dell'epoca Screen
4
```

Figura 8.32. Generazione dei vettori di Skin Conductance.

Sul Workspace di Matlab appariranno le nuove variabili definite e i vettori SC_Baseline, SC_Relax e SC_Stress. Abbiamo così creato un vettore di Skin Conductance per ogni epoca. E' anche vero che potremmo ogni volta indicare la colonna 6 dalla matrice che ci interessa, come ad esempio `data_B(:,6)`, sapendo che questa corrisponde alla Skin

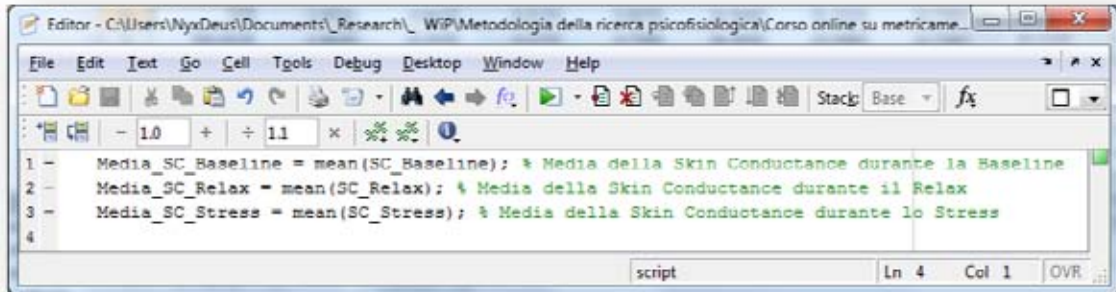


Conductance, tuttavia avere dei vettori di Skin Conductance divisi per epoche è visivamente più immediato ed evita di complicare ulteriormente delle espressioni che cominciano ad essere corpose.

Volendo cominciare a calcolare le medie della Skin Conductance, con Matlab, definiamo come calcolare la media di un generico vettore X. In Matlab è sufficiente usare "`mean(X)`" per ottenere lo scalare media dei valori dell'intero vettore considerato. Dato ad esempio il vettore `[2 4 6 8 10]`, la media si calcolerà come `mean([2 4 6 8 10])`.

Ipotizziamo adesso di voler sapere la media della Skin Conductance durante la Baseline, il Relax e lo Stress.

Sarà sufficiente calcolare la media dei tre vettori di Skin Conductance, assegnando il risultato a tre scalari, con i nomi coerenti rispetto al calcolo (ad esempio `Media_SC_Epoca`). Il programma per eseguire questo semplice calcolo può essere visto in Figura 8.33 e disponibile nel materiale supplementare del volume con il nome "`calcolo_medie_SCs.m`".



```

1 - Media_SC_Baseline = mean(SC_Baseline); % Media della Skin Conductance durante la Baseline
2 - Media_SC_Relax = mean(SC_Relax); % Media della Skin Conductance durante il Relax
3 - Media_SC_Stress = mean(SC_Stress); % Media della Skin Conductance durante lo Stress
4

```

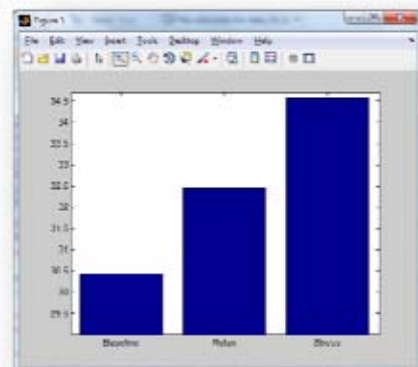
Figura 8.33. Calcolo delle medie per le tre epoche considerate.

Sul Workspace di Matlab appariranno le tre variabili definite per le medie delle tre Skin Conductance:

- Media_SC_Baseline (30.4390);
- Media_SC_Relax (32.4476);
- Media_SC_Stress (34.5733).

Abbiamo così calcolato le tre medie della Skin Conductance per ogni epoca.

Infine possiamo anche graficare le tre medie per un confronto visuale più immediato. L'espressione per graficare le medie è abbastanza semplice, tuttavia viene qui indicata per completezza, ma sarà oggetto di studio del capitolo 10.



```

bar([Media_SC_Baseline Media_SC_Relax Media_SC_Stress]);
set(gca,'XTickLabel',{'Baseline','Relax','Stress'});

```

Capisco pienamente che questo capitolo è piuttosto denso, soprattutto per chi non ha mai studiato né algebra lineare, né programmazione.

Le strutture dei dati registrati con gli strumenti visti nella prima parte del volume sono quasi sempre molto complesse. Abbiamo fatto alcuni esempi sui segnali psicofisiologici, che contengono una elevata quantità di dati, tuttavia le altre tipologie di dati, seppur a volte meno numerose, richiedono una continua manipolazione delle variabili e richiedono un uso considerevole delle espressioni qui viste. Dunque sarà innanzitutto necessario fare molta pratica. In Figura 8.34, la solita esercitazione.



Figura 8.34. Esercitazione su matrici e definizione epoche.

Nel materiale supplementare al volume si trovano tutti i codici usati in questo capitolo, che funzionano sia in Matlab, sia in GNU Octave. Tuttavia suggerisco vivamente di provare a crearseli nuovamente durante l'esercitazione, invece di modificare quelli già forniti. Fare e creare passo dopo passo è in questa fase dell'apprendimento indispensabile e aiuta molto anche a comprendere la logica di processo e la relativa implementazione.

Operazioni multiple generando codice

Finora abbiamo considerato le varie operazioni sulle variabili in modo sequenziale, ovvero processandole una dopo l'altra. Questa è la logica di processo corretta per un ottimo apprendimento di strutture tanto complesse. Il multitasking con la psicomatria computazionale non funziona, sarebbe anzi disastroso. Tuttavia, sempre mantenendo la logica della sequenzialità, ci sono processi che possono essere automatizzati in modo ottimale, per poter essere eseguiti con gruppi di sequenze multiple e farci pertanto risparmiare molto tempo. Il caso tipico è l'importazione, che nei capitoli precedenti abbiamo visto per un solo file, ma in generale potrebbe essere necessaria per decine, centinaia o anche migliaia di file. In casi come questi bisogna automatizzare il processo e anche il wizard dell'importazione, deve essere trasformato in codice per accelerare i nostri processi di codifica. Inizieremo il

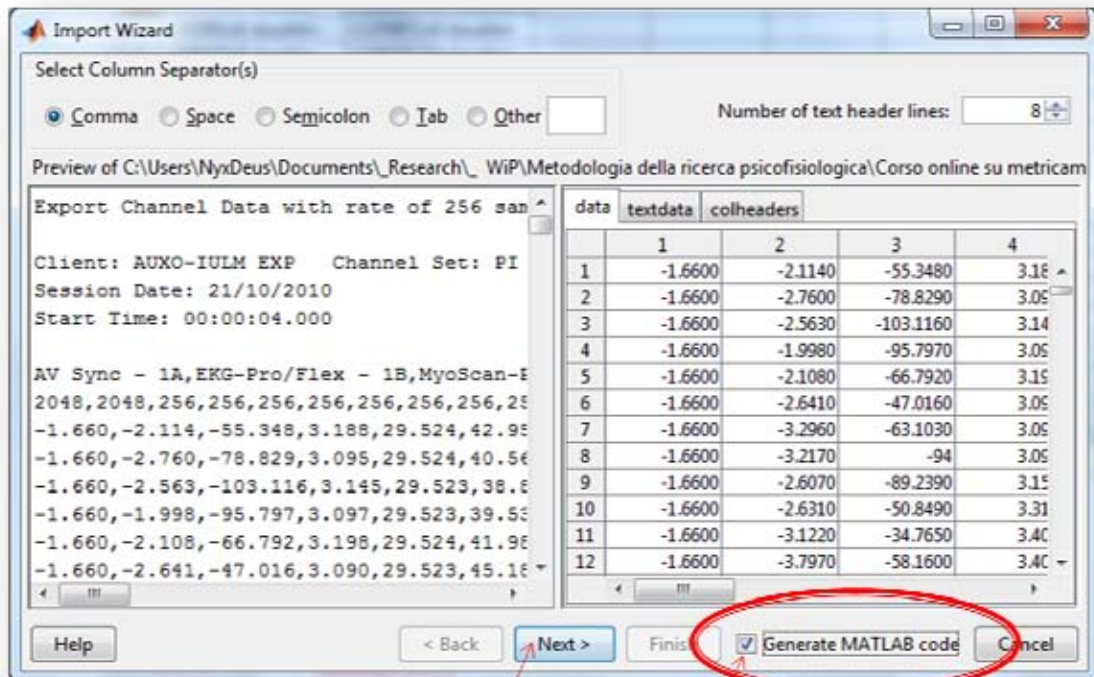
capitolo trattando le funzioni di importazioni disponibili, per imparare a lavorare con più soggetti e dunque con più matrici di dati. Vedremo dopo un metodo piuttosto fantasioso per generare un codice: non sarà elegante, ma è certamente semplice e pratico. Nella visione della logica di processo questa generazione fantasiosa ci aiuterà a ragionare e ci guiderà verso i metodi più evoluti dei cicli, che vedremo più in dettaglio nel capitolo 11.

9.1 Funzioni di importazione

Come abbiamo visto, già dalla prima lezione, Matlab ha un wizard per l'importazione e avevamo appunto affermato che sarebbe servito solo per dare l'illusione tutto sarebbe andato per il meglio. In effetti l'idea di importare i dati con un wizard può andare bene quando questi sono pochi, ma diventa davvero impensabile già sulle decine o se addirittura si va alle centinaia.

Le funzioni di importazioni in Matlab possono essere abbastanza complesse, ma esiste un modo molto semplice per generarle proprio a partire dal wizard che abbiamo già visto al capitolo 7, nel paragrafo 7.2 per importare la matrice "data" dal file "1.txt".

In effetti, all'interno del wizard, esiste una opzione "Generate MATLAB Code" che basta selezionare per ottenere il codice di importazione con i settaggi impostati rispetto a quella specifica tipologia di dati (ad esempio partire dalla riga 8 ed essere delimitati da virgola). In Figura 9.1 vediamo questa opzione.



Poi
cliccare qui

Cliccare qui (flag sulla casella)

Figura 9.1. Wizard di importazione con generazione di codice.

Prima di approfondire la funzione di importazione generata da Matlab, dobbiamo però capire a cosa serve una funzione generica di Matlab. Scriviamo, a tal proposito, una semplice funzione di esempio (che chiameremo appunto "sf"):

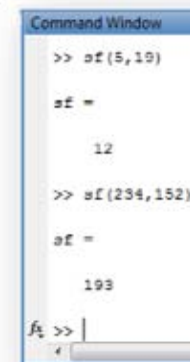
```
function sf(a,b) %funzione sf con richiesta di input: a e b
    sf=(a+b)/2 %funzionalità operativa della funzione sf
```

e salviamola in un file che chiameremo "sf.m" naturalmente disponibile tra i materiali supplementari al volume.

"sf" identificherà la nostra funzione, mentre "a" e "b" saranno i valori di input della nostra funzione. Il nome del file deve essere uguale al nome della funzione con estensione .m (in questo caso dunque "sf.m").

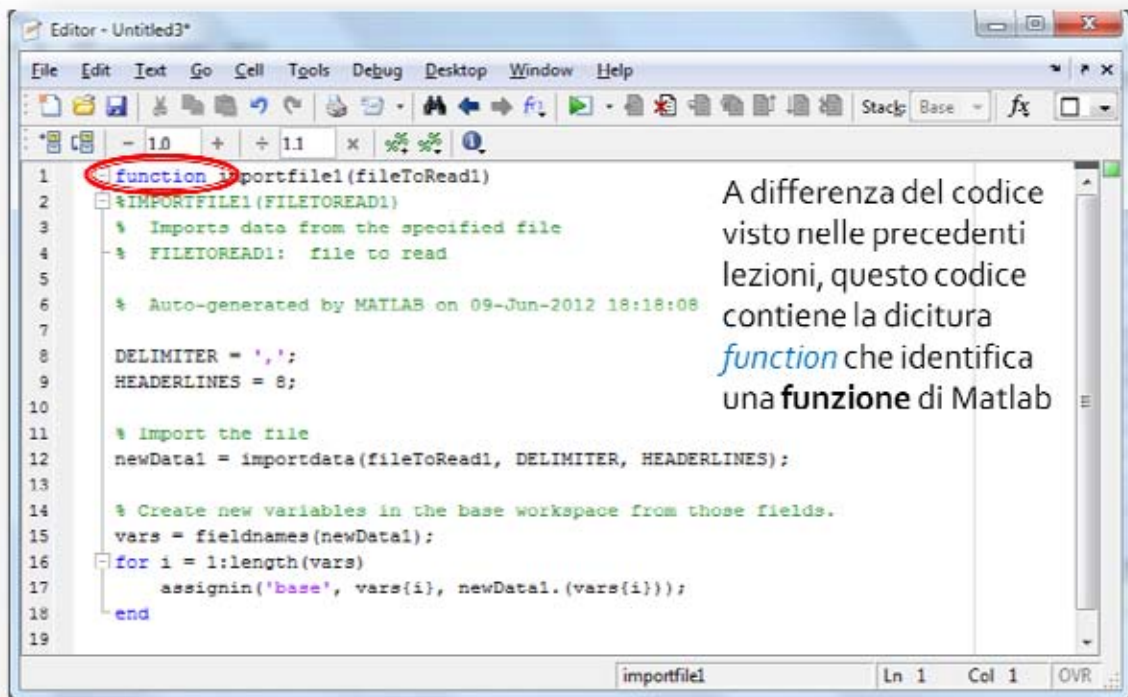
A questo punto se nella Command Window o anche all'interno di un altro file .m che abbiamo creato, scriviamo: `sf(numero1,numero2)`, otterremo il calcolo definito all'interno della funzione stessa. In questo caso sarà: $(\text{numero1} + \text{numero2}) / 2$. Questa definizione è un'operazione molto importante, in quanto ci permette di chiedere un input, definito richiamando la funzione all'occorrenza. Questo meccanismo può essere usato anche con le espressioni viste in precedenza e capiamo dunque adesso perché nei file .m che abbiamo finora creato abbiamo sempre dichiarato dei parametri soggetti a cambiamenti (come il sampling rate). Tali valori possono essere dati come input delle funzioni che andiamo a definire e dunque servono ad avere un codice maggiormente flessibile rispetto alle esigenze presenti e future del suo programmatore.

Chiarito il modo d'uso e la stessa utilità delle funzioni, torniamo all'importazione e in particolare al codice che Matlab automaticamente genera usando l'apposita funzione nel wizard (come da Figura 9.1). Una volta terminata la prima importazione,



```
Command Window
>> sf(5,19)
sf =
    12
>> sf(234,152)
sf =
    193
f1 >> |
```

Matlab aprirà una finestra di Editor contenente il codice appena generato (Figura 9.2).



```

1 function importfile1(fileToRead1)
2 %IMPORTFILE1 (FILETOREAD1)
3 % Imports data from the specified file
4 % FILETOREAD1: file to read
5
6 % Auto-generated by MATLAB on 09-Jun-2012 18:18:08
7
8 DELIMITER = ',';
9 HEADERLINES = 8;
10
11 % Import the file
12 newData1 = importdata(fileToRead1, DELIMITER, HEADERLINES);
13
14 % Create new variables in the base workspace from those fields.
15 vars = fieldnames(newData1);
16 for i = 1:length(vars)
17     assignin('base', vars{i}, newData1.(vars{i}));
18 end
19

```

A differenza del codice visto nelle precedenti lezioni, questo codice contiene la dicitura *function* che identifica una **funzione** di Matlab

Figura 9.2. Funzione di importazione generata da Matlab.

Nel codice ci sono due variabili che chiaramente identificano il tipo di delimitatore di testo e il numero di righe di intestazione da non considerare nell'importazione. Il resto del codice è più complesso, ma non è necessario per il momento approfondirlo.

Bisogna invece definire la funzione, dandole innanzitutto un nome semplice, che ci aiuti a rievocare semplicemente la sua funzionalità. Come si può vedere in Figura 9.3, possiamo indicare ad esempio il nome `imp` per la funzione di importazione, che

implicherà di dover chiamare il file .m della funzione imp (di importazione) come "imp.m", come da Figura 9.3 e 9.4.

The screenshot shows a MATLAB editor window with the following code:

```

1 function imp(fileToRead1)
2 %IMPORTFILE(FILETOREAD1)
3 % Imports data from the specified file
4 % FILETOREAD1: file to read
5
6 % Auto-generated by MATLAB on 09-Jun-2012 18:18:08
7
8 - DELIMITER - ',';
9 - HEADERLINES = 8;
10
11 % Import the file
12 newData1 = importdata(fileToRead1, DELIMITER, HEADERLINES);
13
14 % Create new variables in the base workspace from those fields.
15 vars = fieldnames(newData1);
16 for i = 1:length(vars)
17     assignin('base', vars(i), newData1.(vars(i)));
18 end
19

```

On the right side of the editor, there is a text box that reads: "Per semplicità chiamiamo la funzione imp e di conseguenza salviamo il file chiamandolo imp.m". The word "imp" in the function definition on line 1 is circled in red.

Figura 9.3. Nominare funzione e file .m in modo coerente.

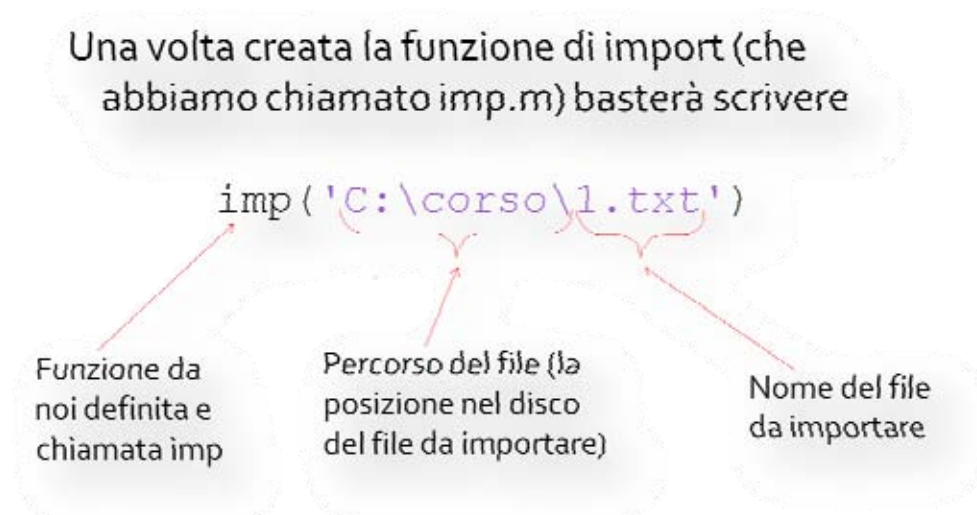


Figura 9.4. Sintassi della funzione di importazione creata.

Percorso e nome del file vanno sempre messi tra apici (come in Figura 9.4. L'editor di Matlab provvederà a indicarli in viola per darne maggiore evidenza.

Una funzione di importazione presenta una serie di vantaggi che la rende più appetibile.

- è più semplice se devo ripetere l'operazione più volte (esempio: importare molti file con le stesse caratteristiche)
- l'esecuzione è molto più veloce
- può essere usata dentro altro codice
- ci permetterà di automatizzare processi complessi (import multipli)

Quando facciamo una ricerca con misure computazionali usiamo la stessa strumentazione per tutti i soggetti, quindi i file estratti (i vari .txt) saranno tutti con le stesse caratteristiche. Ci chiediamo dunque come fare un import per tutti i soggetti, in modo automatico. Se ad esempio abbiamo otto soggetti possiamo scrivere:

```
imp('C:\corso\1.txt')  
imp('C:\corso\2.txt')  
imp('C:\corso\3.txt')  
imp('C:\corso\4.txt')  
imp('C:\corso\5.txt')  
imp('C:\corso\6.txt')  
imp('C:\corso\7.txt')  
imp('C:\corso\8.txt')
```


Tuttavia facendo in questo modo avremo un piccolo problema. Quando importiamo i segnali del primo soggetto useremo, come visto prima:

```
imp('C:\corso\1.txt')
```

Nel workspace spunterà una matrice data. Quando importiamo i segnali del secondo soggetto useremo, come visto prima:

```
imp('C:\corso\2.txt')
```

Nel workspace spunterà una matrice data che andrà a cancellare quella precedente. Così continuando per gli otto soggetti, alla fine nel Workspace avremo una sola matrice "data" relativa all'ottavo soggetto.

Dunque, arrivati all'ottavo soggetto (l'ultimo) avremo un'unica matrice data e non 8 come vorremmo. Quindi manca ancora qualcosa nel nostro programma

Per risolvere il problema basterà fare un assegnamento dopo ogni import:

```
imp('C:\corso\1.txt')
```

```
Sbj1=data;
```

```
imp('C:\corso\2.txt')
```

```
Sbj2=data;
```

e così via, per tutti gli otto soggetti. Inseriamo queste espressioni in un apposito file "impall.m" (sta per importa tutto, import all), come in Figura 9.5.

```

1 - imp('C:\corso\1.txt') % importo i segnali del primo soggetto
2 - Sbj1=data; % Creo una matrice Sbj1 contenente i segnali appena importati
3 - imp('C:\corso\2.txt') % importo i segnali del secondo soggetto
4 - Sbj2=data; % Creo una matrice Sbj2 contenente i segnali appena importati
5 - imp('C:\corso\3.txt')
6 - Sbj3=data;
7 - imp('C:\corso\4.txt')
8 - Sbj4=data;
9 - imp('C:\corso\5.txt')
10 - Sbj5=data;
11 - imp('C:\corso\6.txt')
12 - Sbj6=data;
13 - imp('C:\corso\7.txt')
14 - Sbj7=data;
15 - imp('C:\corso\8.txt') % importo i segnali dell'ottavo soggetto
16 - Sbj8=data; % Creo una matrice Sbj8 contenente i segnali appena importati

```

Salvato con il nome impall.m lo trovate come sempre nel materiale supplementare al volume

Figura 9.5. File "impall.m" per importare gli otto file txt.

Il problema di questo processo è rispetto al numero di matrici da importare. Fino a quando sono otto va anche bene, ma se diventano 80, 800 o 8 mila, questo processo è comunque problematico.

Una possibile soluzione è il caro vecchio e sempre buono "copia-incolla", ma comunque è sempre necessario cambiare i numeri stando peraltro molto attenti a non sbagliare.

Inoltre se vogliamo anche fare delle operazioni su tutte le matrici importate, può essere molto complesso cambiare i numeri su centinaia di righe di codice, se non migliaia.

Ipotizziamo di avere fatto una sperimentazione con venti partecipanti e di aver messo i dati sulla cartella "C:\corso\" del nostro disco.

Vogliamo importare tutti le matrici di dati dei segnali psicofisiologici dei venti partecipanti e aggiungere una colonna dei tempi all'inizio di ogni matrice di dati. I file da importare si chiamano 1.txt, 2.txt, ... , 20.txt. Le matrici hanno tutte dimensione diversa.

Innanzitutto iniziamo con lo scrivere il file .m per il primo partecipante, con file da importare "1.txt".

```

1  % Importazione dei dati e aggiunta del tempo sulla prima colonna PER IL SOGGETTO 1
2  imp('C:\corso\1.txt') % importo i segnali del primo soggetto come matrice "data"
3
4  % Programmino add_time_to_data <-- vedi codice del Capitolo 8 !
5
6  L=length(data); % L è la lunghezza (di righe) di data. Data è la matrice di dati
7  s=256; % s è il sampling rate (Hz), ovvero il numero di righe registrate per sec.
8  t=1/s:1/s:L/s; % t è il tempo, calcolato in funzione del sampling rate e di L
9  t=t'; % calcola la trasposta di t, trasformando t in vettore colonna
10 data=[t data]; % accosta la colonna dei tempi t a sinistra della matrice data
11
12 Sb1=data; % Creo una matrice Sb1 contenente i segnali appena importati
13 % La nostra matrice Sb1 avrà come prima colonna il tempo

```

Figura 9.6. File .m per importare il file 1.txt relativo al primo partecipante.

Ci chiediamo adesso se sia possibile replicare in automatico questo processo (e in particolare il relativo codice) su tutti i partecipanti, senza dover copiare, incollare e modificare ogni numero di riferimento, diminuendo così anche la probabilità di commettere un errore.

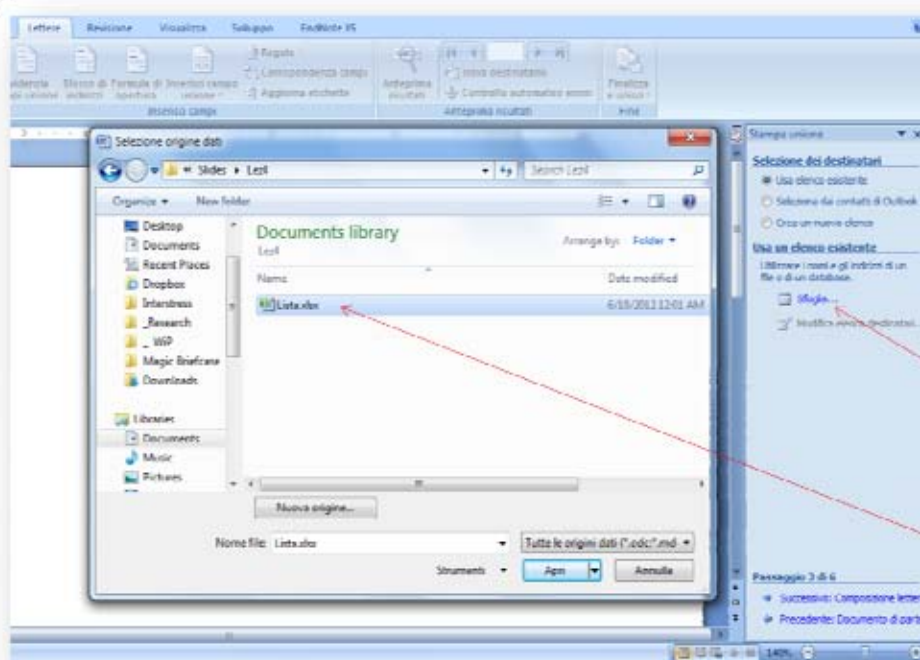
Una prima soluzione fantasiosa è quella di usare la "stampa unione" in Microsoft Word per creare un generatore di codice basato su una lista fatta in Microsoft Excel.

La stampa unione è uno strumento di Microsoft Word, molto semplice e molto potente. Per prima cosa creare una lista in Microsoft Excel con il nome dei file da importare (relativo al nome dei partecipanti). Nel nostro caso i nomi dei file da importare sono dei numeri da uno a venti !!



Figura 9.7. Creare una lista in Excel, in modo rapido.

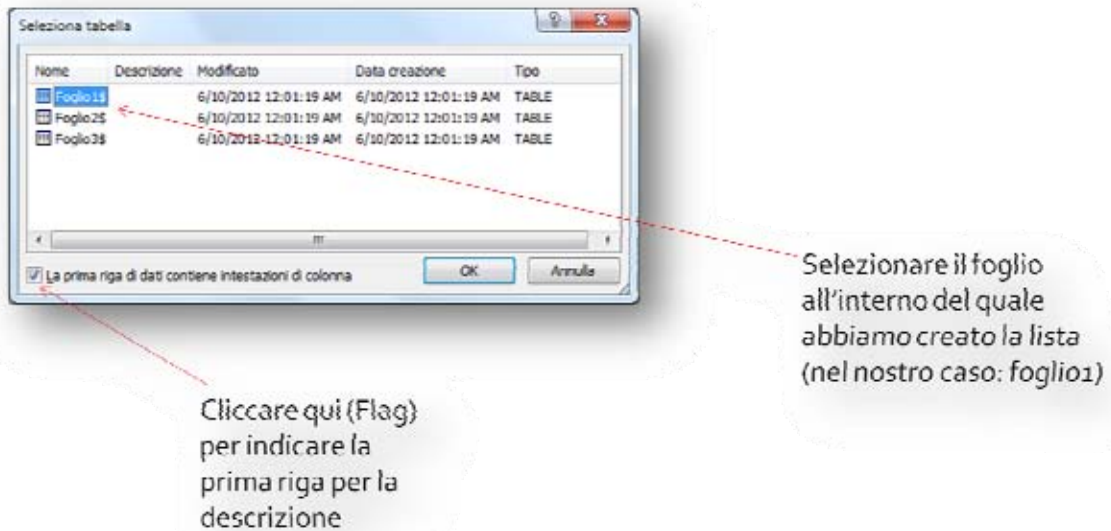
Una volta creata la lista in Excel, aprire Word e selezionare "Lettere > Inizia stampa unione > Creazione guidata stampa unione". Sulla parte destra della pagina si apre il wizard della stampa unione. Cliccare su "lettere" e procedere cliccando su "Successivo ...". Cliccare su "usa il documento corrente per usare il file vuoto appena aperto e quindi cliccare su "Successivo ..." per procedere. A questo punto, come mostrato in Figura 9.8, bisogna selezionare la lista Excel appena creata (il file della lista deve essere chiuso per poterlo selezionare in Word).



Cliccare su
Sfogliare
e scegliere il
file
Lista.xlsx
prima
creato

Figura 9.8. Selezione della lista Excel nella stampa unione Word.

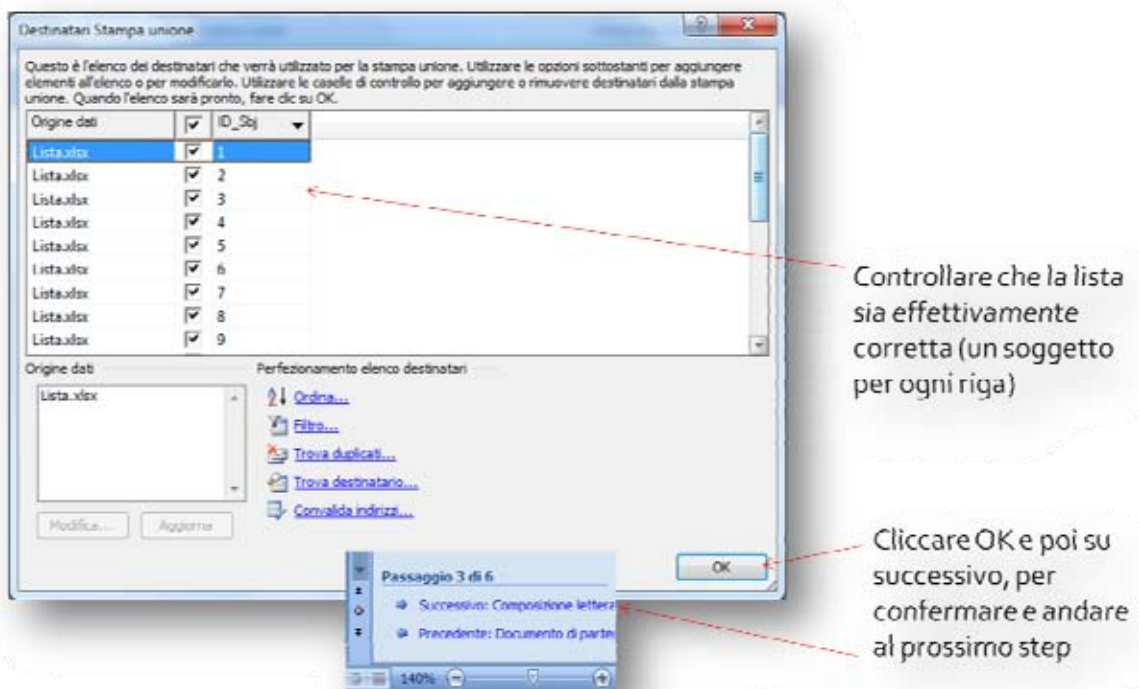
Una volta selezionata e aperto il file, si potrà scegliere il foglio di lavoro Excel (Figura 9.9) e controllare la lista (Figura 9.10).



Cliccare qui (Flag) per indicare la prima riga per la descrizione

Selezionare il foglio all'interno del quale abbiamo creato la lista (nel nostro caso: foglio1)

Figura 9.9. Selezione del foglio di lavoro che contiene la lista.



Controllare che la lista sia effettivamente corretta (un soggetto per ogni riga)

Cliccare OK e poi su successivo, per confermare e andare al prossimo step

Figura 9.10. Visualizzazione e conferma della lista.

A questo punto possiamo Incollare nel documento Word vuoto, il contenuto di "imp_e_time_Sbj1.m" (file di importazione del primo partecipante). Nello step successivo andremo a inserire i riferimenti della stampa unione al posto del numero del partecipante nel codice, come in Figura 9.11.

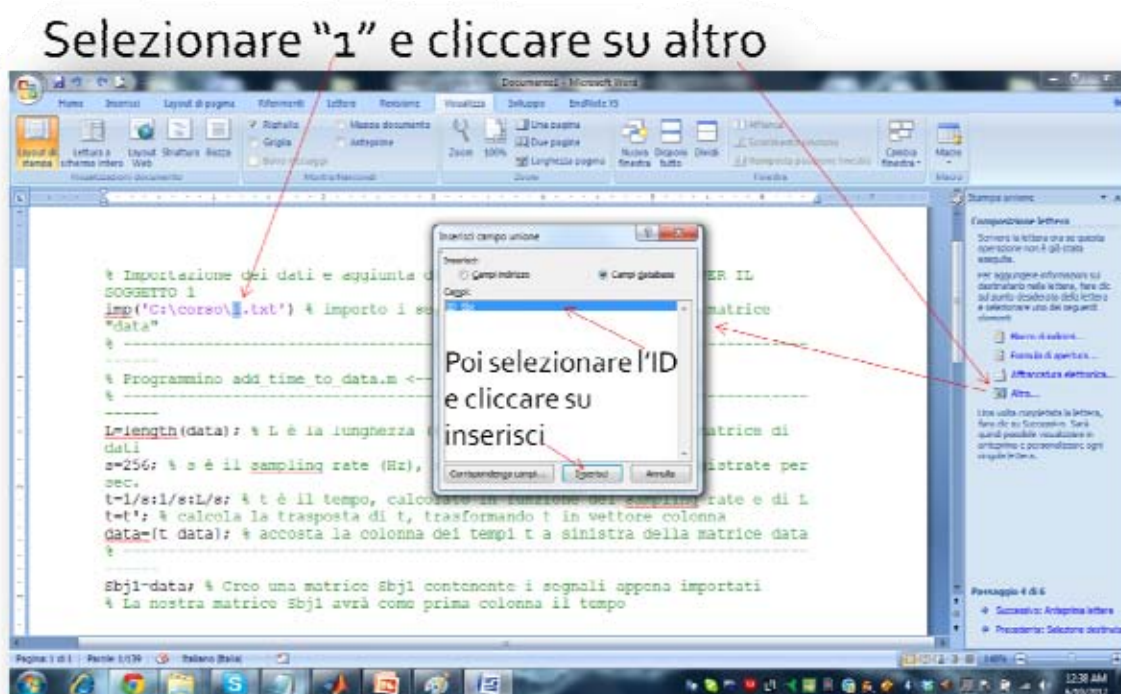


Figura 9.11. Inserimento dei riferimenti di stampa unione (legati alla lista selezionata) nel codice.

A questo punto si può cliccare "Chiudi" e ripetere l'operazione appena vista per tutti gli "1" (nome soggetto) presenti nel codice (Figura 9.12 e 9.13, nelle quali si può vedere come al posto di 1 spunti prima <<ID_Sbj>> e dopo il campo chiave della lista).

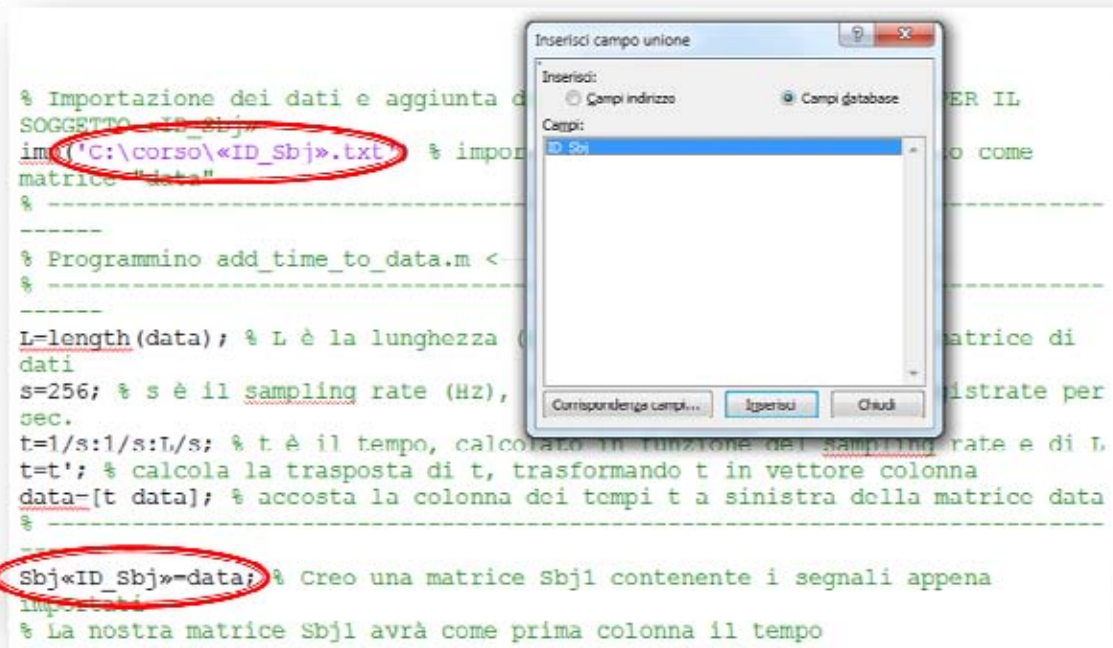


Figura 9.12. Inserimento dei campi unione al posto dei vari 1.

Controllare che cambiando il Destinatario
anche i numeri settati cambino

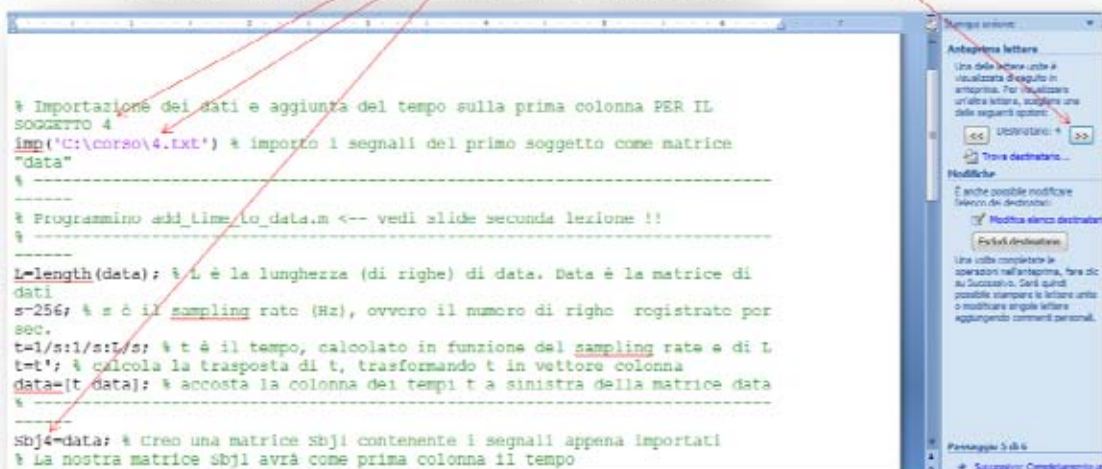
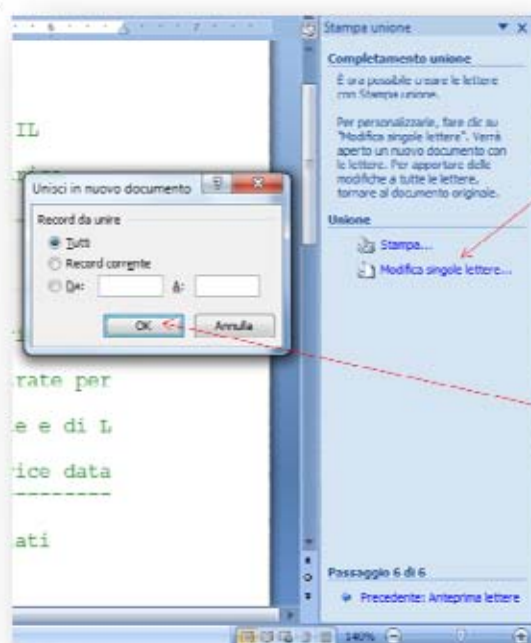


Figura 9.13. Anteprima del codice per ogni elemento della lista.

Infine, quindi, tramite la stampa unione, avrò la possibilità di generare venti pagine di codice diverso, personalizzato per ciascun partecipante (Figura 9.14).



Cliccando su "Modifica singole lettere" è possibile filtrare (o selezionare tutti) i numeri (nomi soggetti) da generare.

Cliccando ok verrà generato un file di 20 pagine (una per numero) con il codice.

Figura 9.14. Generazione finale di tutte le pagine (una per ciascun codice generato).

Una volta ottenuto il file Word unico con le venti pagine di codice, si potrà copiare per intero e incollare al pari in una finestra di editor di Matlab.

Il codice così incollato importa 20 soggetti (1.txt, 2.txt, ... , 20.txt) creando 20 matrici (Sbj1, Sbj2, ..., Sbj20) con i tempi sulla prima colonna. In Figura 9.15 si può vedere l'editor di Matlab con una parte del codice generato, relativa ai partecipanti 19 e 20.

```

249
250 Sbj18=data; % Creo una matrice Sbj18 contenente i segnali appena importati
251 % La nostra matrice Sbj18 avrà come prima colonna il tempo
252
253 % Importazione dei dati e aggiunta del tempo sulla prima colonna PER IL SOGGETTO 19
254 imp('C:\corso\19.txt') % importo i segnali del primo soggetto come matrice "data"
255 % -----
256 % Programmino add_time_to_data.m <-- vedi slide seconda lezione !!
257 % -----
258 L=length(data); % L è la lunghezza (di righe) di data. Data è la matrice di dati
259 s=256; % s è il sampling rate (Hz), ovvero il numero di righe registrate per sec.
260 t=1/s:1/s:L/s; % t è il tempo, calcolato in funzione del sampling rate e di L
261 t=t'; % calcola la trasposta di t, trasformando t in vettore colonna
262 data=[t data]; % accosta la colonna dei tempi t a sinistra della matrice data
263 % -----
264 Sbj19=data; % Creo una matrice Sbj19 contenente i segnali appena importati
265 % La nostra matrice Sbj19 avrà come prima colonna il tempo
266
267 % Importazione dei dati e aggiunta del tempo sulla prima colonna PER IL SOGGETTO 20
268 imp('C:\corso\20.txt') % importo i segnali del primo soggetto come matrice "data"
269 % -----
270 % Programmino add_time_to_data.m <-- vedi slide seconda lezione !!
271 % -----
272 L=length(data); % L è la lunghezza (di righe) di data. Data è la matrice di dati
273 s=256; % s è il sampling rate (Hz), ovvero il numero di righe registrate per sec.
274 t=1/s:1/s:L/s; % t è il tempo, calcolato in funzione del sampling rate e di L
275 t=t'; % calcola la trasposta di t, trasformando t in vettore colonna
276 data=[t data]; % accosta la colonna dei tempi t a sinistra della matrice data
277 % -----
278 Sbj20=data; % Creo una matrice Sbj20 contenente i segnali appena importati
279 % La nostra matrice Sbj20 avrà come prima colonna il tempo
280
281

```

Figura 9.15. Codice incollato in Matlab dalla stampa unione e salvato come "impall_con_time_20_Sbjs.m".

Questa "generazione di codice" richiede uno sforzo esattamente pari sia se fatta su 20 soggetti così come se viene fatta su 2000. Si riduce notevolmente la probabilità di commettere errori: se si controlla bene il codice del primo soggetto e la sua estensione (individuazione e sostituzione degli "1"), è praticamente impossibile sbagliare.

Questa procedura richiede che i file ".txt", contenenti i dati dei partecipanti, siano ordinati e codificati dal numero 1.txt (primo soggetto) al numero x.txt (x-esimo / ultimo soggetto). Però nulla vieta che sia adattata usando una lista dei nomi dei file ".txt" relativi ai soggetti. Ovviamente se un nome è anche lievemente diverso, Matlab darà errore. E' dunque generalmente consigliato, per evitare errori, l'uso di sequenze di numeri (1.txt, 2.txt, ...).

Per evitare di rinominare centinaia o migliaia di file manualmente e dunque evitare di commettere errori umani, suggerisco Renamer (<http://www.den4b.com/?x=downloads&product=renamer>), un bel software gratuitamente scaricabile che permette di rinominare i file con una vasta gamma di opzioni (Figura 9.16), oltre ad advanced renamer già visto al precedente capitolo 2.

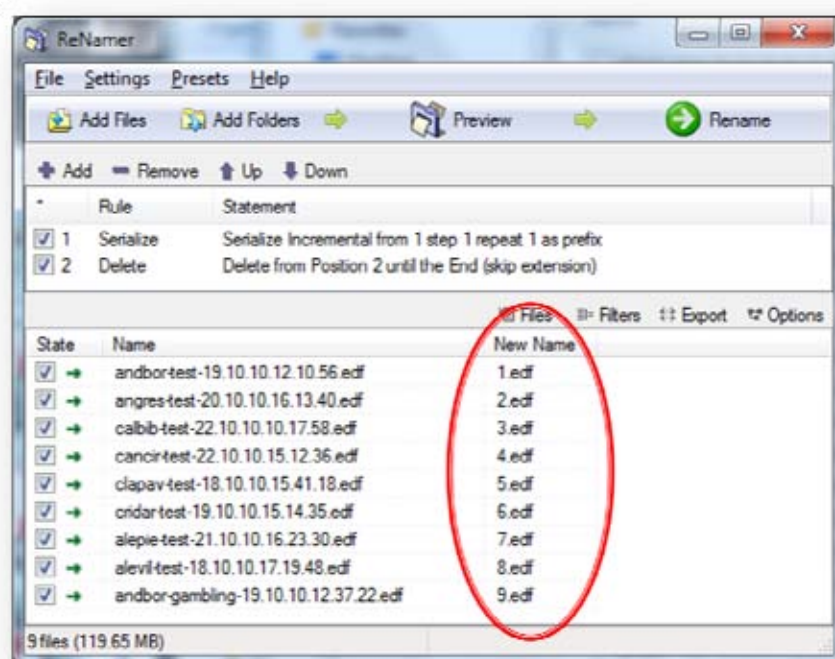


Figura 9.16. Uso di renamer per serializzare una lista di nomi.

Tramite questa procedura, dunque, Importare centinaia o migliaia di file ".txt" contenenti, ad esempio, segnali psicofisiologici della durata di 50 minuti l'uno campionati a 256 Hz, è un'operazione affidabile e automatizzabile, seppur richiede un certo tempo di computazione.


9.2 Codice Visual Basic for Application

Nel precedente paragrafo abbiamo visto come importare più matrici di dati usando Matlab. Come abbiamo detto, l'uso di questi software avanzati, per i dati complessi, è sempre consigliabile in quanto consentono una manipolazione delle matrici importate con routine automatizzate. Tuttavia a volte può essere utile serializzare delle operazioni anche in Microsoft Excel, in particolare questo potrebbe essere utile per matrici meno numerose rispetto a quelle viste finora e soprattutto quando si vogliono fare dei semplici calcoli automatizzati più di ampio utilizzo, anche da parte di chi non ha o non vuole installare né Matlab, né GNU Octave.

Vedremo adesso un esempio di import multiple, per continuità con il precedente paragrafo, ma quello che importa è la logica di processo. Le operazioni di serializzazione viste in questo paragrafo valgono dunque anche per altre operazioni.

In Microsoft Excel esiste un motore di programmazione basato sul linguaggio Basic. In particolare esiste un vero e proprio ambiente di sviluppo con un linguaggio detto Visual Basic for Application.

L'accesso a questo ambiente non è normalmente visibile quando si installi Excel, tuttavia basta attivare la visualizzazione per poterci lavorare.

Aprire Microsoft Excel, cliccare sul pulsante Office  e selezionare "Opzioni di Excel". Apparirà la finestra visibile in Figura 9.17, in cui basterà selezionare "Mostra scheda Sviluppo sulla barra multifunzione" e cliccare su "Ok".

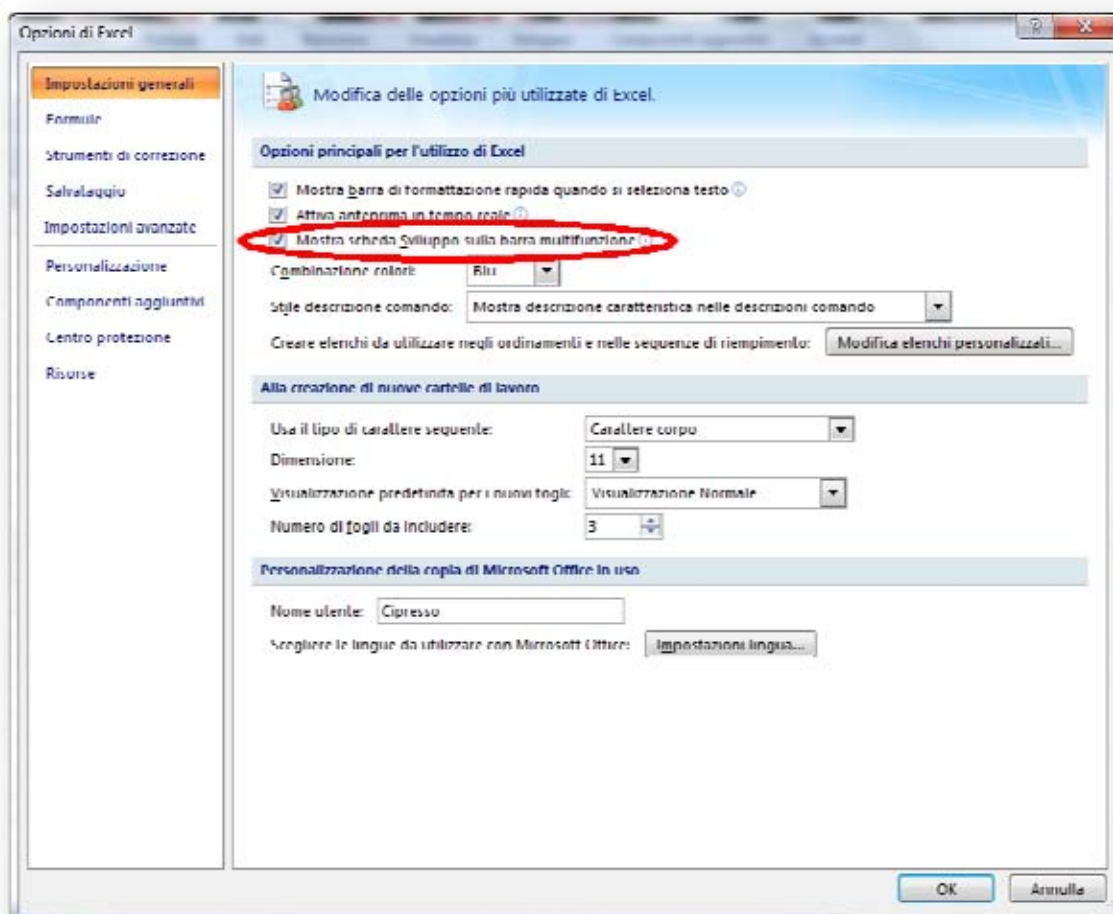


Figura 9.17. Opzioni di Excel per l'attivazione della scheda Sviluppo sulla barra multifunzione.

Come si può vedere in Figura 9.18, sulla barra multifunzione apparirà un nuovo tab "Sviluppo", con molte funzioni attive, tra cui il pulsante Visual Basic, premendo il quale si avrà accesso all'editor di programmazione.

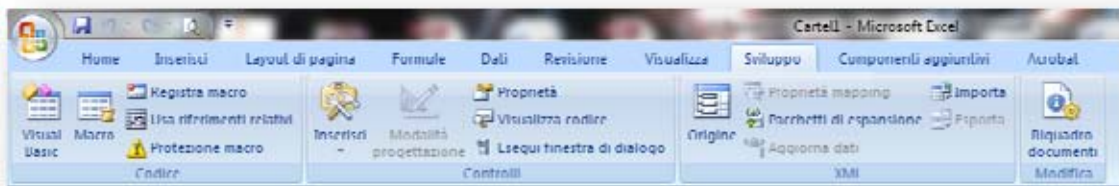


Figura 9.18. Barra multifunzione con il tab di sviluppo.

Cliccando sul pulsante "Visual Basic" abbiamo la possibilità di creare un nuovo Modulo come in Figura 9.19. Sulla parte destra avremo così l'editor del Modulo (rinominabile in base alle nostre esigenze).

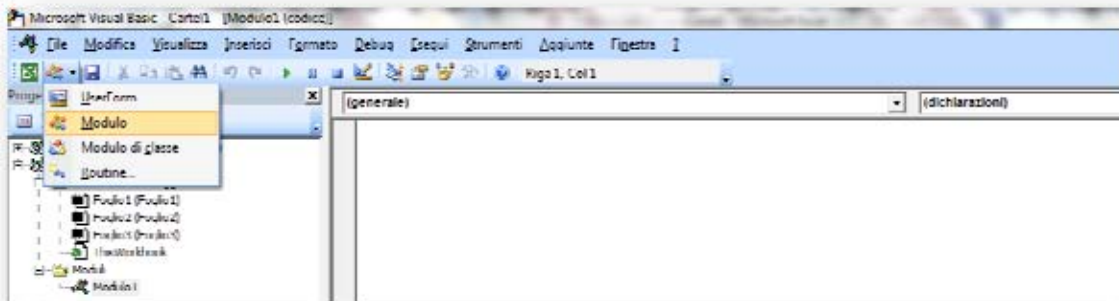


Figura 9.19. Creazione di un nuovo Modulo nell'ambiente di sviluppo di Visual Basic.

Di seguito indico, per cominciare a fare qualche esercizio, il codice Visual Basic for Application (VBA) per l'import di un file "1.txt" contenuto nella cartella "C:\Corso\Capitolo 9\". L'apostrofo precede i commenti come la % li precede in Matlab.

```
Sub Import1()
  Workbooks.Add
  With ActiveSheet.QueryTables.Add(Connection:="TEXT;C:\Corso\Capitolo
9\1.txt" _
    , Destination:=Range("$A$1"))
    .Name = "1"
    .FieldNames = True
    .RowNumbers = False
    .FillAdjacentFormulas = False
    .PreserveFormatting = True
    .RefreshOnFileOpen = False
    .RefreshStyle = xlInsertDeleteCells
    .SavePassword = False
    .SaveData = True
    .AdjustColumnWidth = True
    .RefreshPeriod = 0
    .TextFilePromptOnRefresh = False
    .TextFilePlatform = 437
    .TextFileStartRow = 1
    .TextFileParseType = xlDelimited
    .TextFileTextQualifier = xlTextQualifierDoubleQuote
    .TextFileConsecutiveDelimiter = False
    .TextFileTabDelimiter = True
    .TextFileSemicolonDelimiter = False
    .TextFileCommaDelimiter = False
    .TextFileSpaceDelimiter = False
    .TextFileColumnDataTypes = Array(1, 1, 1, 1, 1, 1)
    .TextFileTrailingMinusNumbers = True
    .Refresh BackgroundQuery:=False
  End With
  ActiveWorkbook.SaveAs Filename:="C:\Corso\Capitolo 9\1.xlsx", FileFormat:= _
  xlOpenXMLWorkbook, CreateBackup:=False
  ActiveWindow.Close
End Sub
```

Il nome di questo programma sarà quello assegnato dopo Sub, nel nostro caso "Sub Import1()" indica il nome Import1 (ovvero importazione del file "1.txt" in Excel. L'istruzione "End Sub" chiude invece il programma.

Per eseguire questo programma basterà cliccare sul pulsante "Macro" sulla barra multifunzione con il tab di sviluppo e scegliere il nome corrispondente (Figura 9.20).

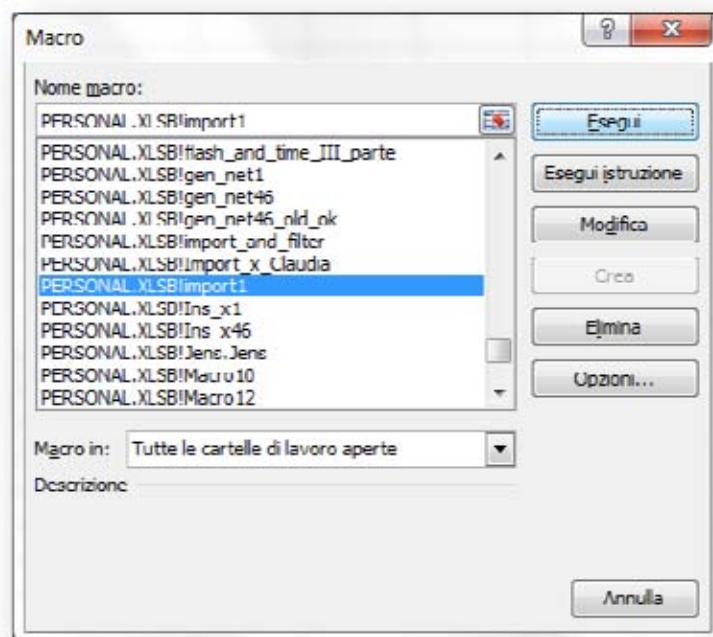


Figura 9.20. Esecuzione del programma VBA prima scritto.

Il codice sopra riportato può essere eseguito per importare il file "1.txt", oppure può essere opportunamente "replicato" per importare gli n file disponibili (1.txt, 2.txt, ...).

La procedura da seguire è esattamente la stessa dello scorso paragrafo. Dunque usiamo la stampa unione di word dopo aver creato una lista con prima riga ID e sotto i numeri in sequenza (come già fatto in Figura 9.7), poi associamo la lista al file Word, identificando i nomi da associare agli elementi della lista (quindi "1.txt", o meglio "1" da trasformare in <<ID>>), come in Figura 9.21.

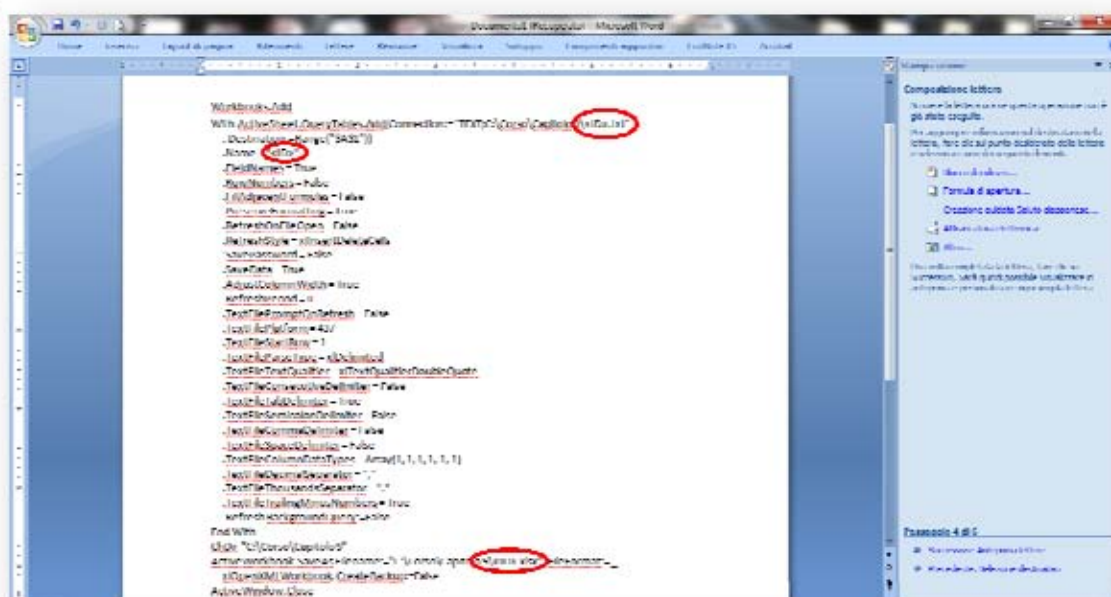


Figura 9.21. Individuazione degli ID di file.

Finalizzando la stampa unione per tutti gli elementi della lista, avremo il programma visto opportunamente replicato per ogni elemento della lista (1, 2, ...). Selezioniamo tutto il codice generato e lo incolliamo nell'editor di VBA, come in Figura 9.22.

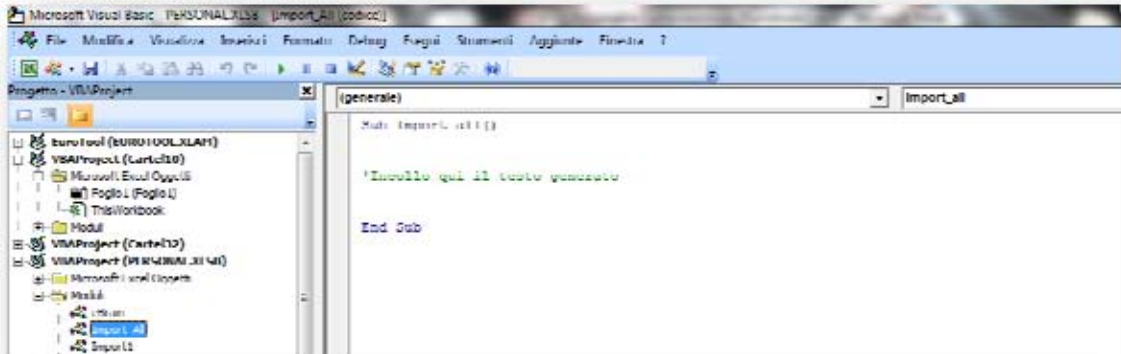


Figura 9.22. Copia del codice di importazione per tutti i soggetti nell'editor di Visual Basic for Application.

Come risultato il programma generato creerà dei file .xlsx contenenti i dati importati e mantenendo la stessa numerazione (Figura 9.23).

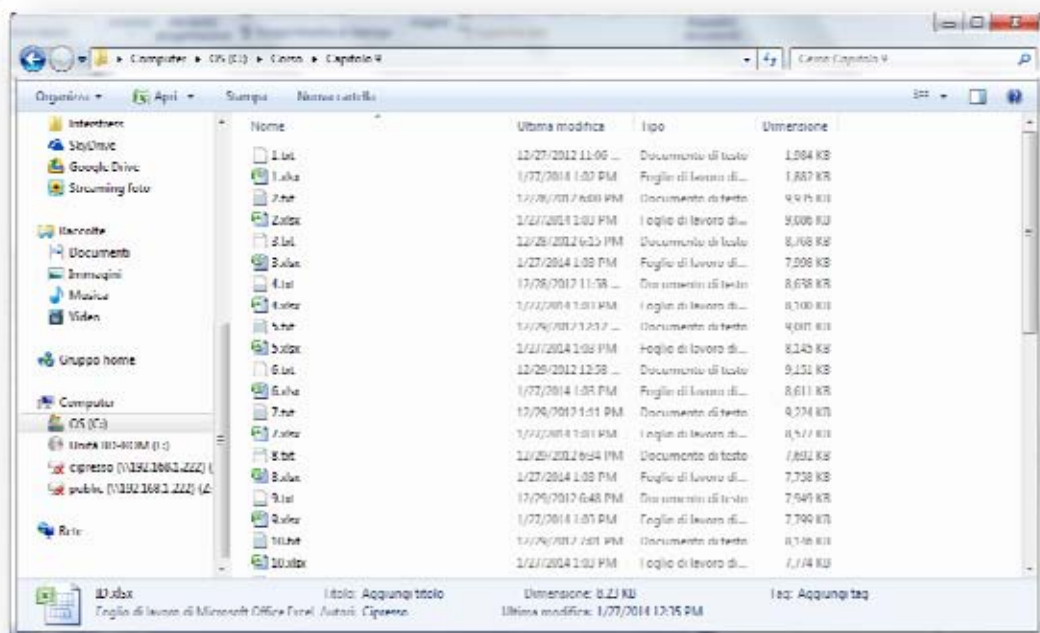


Figura 9.23. Risultato prodotto dal programma "Import_All".

I file "Import1.bas" e "Import_All.bas", disponibili nel materiale supplementare al volume, possono essere importati nell'editor di Visual Basic for Application e modificati in base alle proprie esigenze. Anche i file Word della stampa unione (ID, generatore e generato) sono disponibili nel materiale.

Creare del codice personalizzato in Excel, come dicevamo all'inizio può essere molto utile per creare delle interfacce grafiche a più ampia diffusione. La creazione di pulsanti e altri elementi è assai semplice, come si può vedere in Figura 9.24.

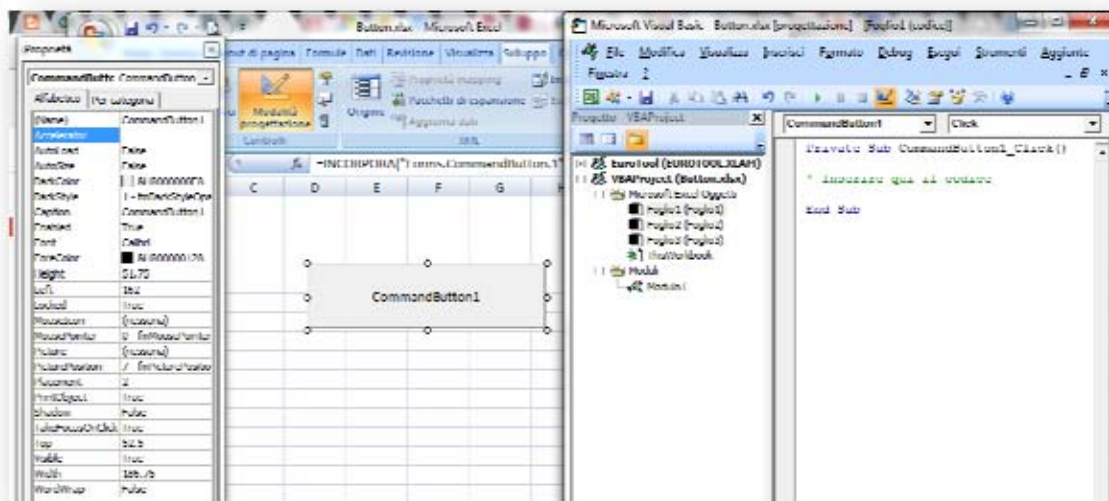


Figura 9.24. Creazione di un'interfaccia grafica in Excel con Visual Basic for Application.

Le possibilità che si aprono sono molte perché in questo modo si possono creare dei fogli di calcolo interattivi per l'inserimento di dati e il calcolo, anche molto complesso, tramite VBA.

Il vantaggio di queste interfacce grafiche sta nel fatto che Excel è disponibile a molti utenti, quindi a volte questa soluzione può essere utile per la condivisione e il calcolo automatico di dati di ricerca o altre informazioni.

Si possono peraltro realizzare interfacce grafiche anche solo usando le normali macro di Excel (comandi registrati e ripetuti), anche se imparare il VBA può essere molto utile per operazioni avanzate. Inoltre quando creiamo una macro, più o meno complessa, Excel la convertirà sempre in codice VBA e sarà possibile modificarla seguendo le procedure fin qui viste.

Plotting dei dati

Nei capitoli precedenti abbiamo esaminato modalità di intervento sui dati, tramite strutture molto flessibili che ci hanno permesso una efficace ristrutturazione dei dati secondo epoche sperimentali o altri criteri. Sebbene questo processo sia indispensabile per poter fare dei ragionamenti sui dati e per poter arrivare a misure di sintesi di questi dati, vi è un passaggio obbligato precursore delle analisi e spesso anche loro successore. Questo passaggio è l'operazione di plotting dei dati. Lo scopo non è solamente quello di presentare dei risultati (seppur anch'esso importante), ma quello di poterli esaminare per individuare possibili incongruenze tra quanto previsto e quanto registrato. Queste incongruenze sono particolarmente evidente nel caso dei segnali psicofisiologici, dove la presenza di artefatti da movimenti o interferenze di diverso tipo, produce un'evidente rottura o stortura del segnale e una modifica del corretto risultato nel caso in cui non si intervenga opportunamente per ripristinare la reale sequenza dei dati.

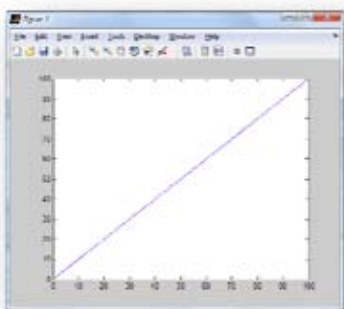
Il plotting è dunque un'operazione assai delicata che va sempre effettuata prima di trarre informazioni. Inoltre, errori nel segnale a parte, il plotting consente anche una più approfondita analisi dei dati, aiutandoci a individuare gli andamenti durante tutta la sessione sperimentale e non solo gli effetti pre-post sperimentazione.

10.1 Primi passi nel plotting

Per iniziare a capire come plottare i dati è necessario avere acquisito le capacità di manipolazione algebrica viste nei precedenti capitoli. In particolare è necessario avere una certa dimestichezza con le matrici e il codice .m di Matlab o GNU Octave.

Per plottare i dati di un vettore X (riga o colonna), è sufficiente digitare "plot(X)". Ad esempio:

```
X=[1:100]; % creazione di un vettore riga  
plot(X); % plot del vettore riga (sarà un segmento)
```



Il plot è normalmente prodotto in una nuova finestra, sia in Matlab, sia in GNU Octave. E' anche assai semplice salvarlo (File > Save As) nel formato che riteniamo più adatto.

Proviamo con un esempio un po' più complesso:

```
X=[1:100]; % creazione di un vettore riga  
plot(sin(X)); % possiamo plottare funzioni del nostro vettore
```

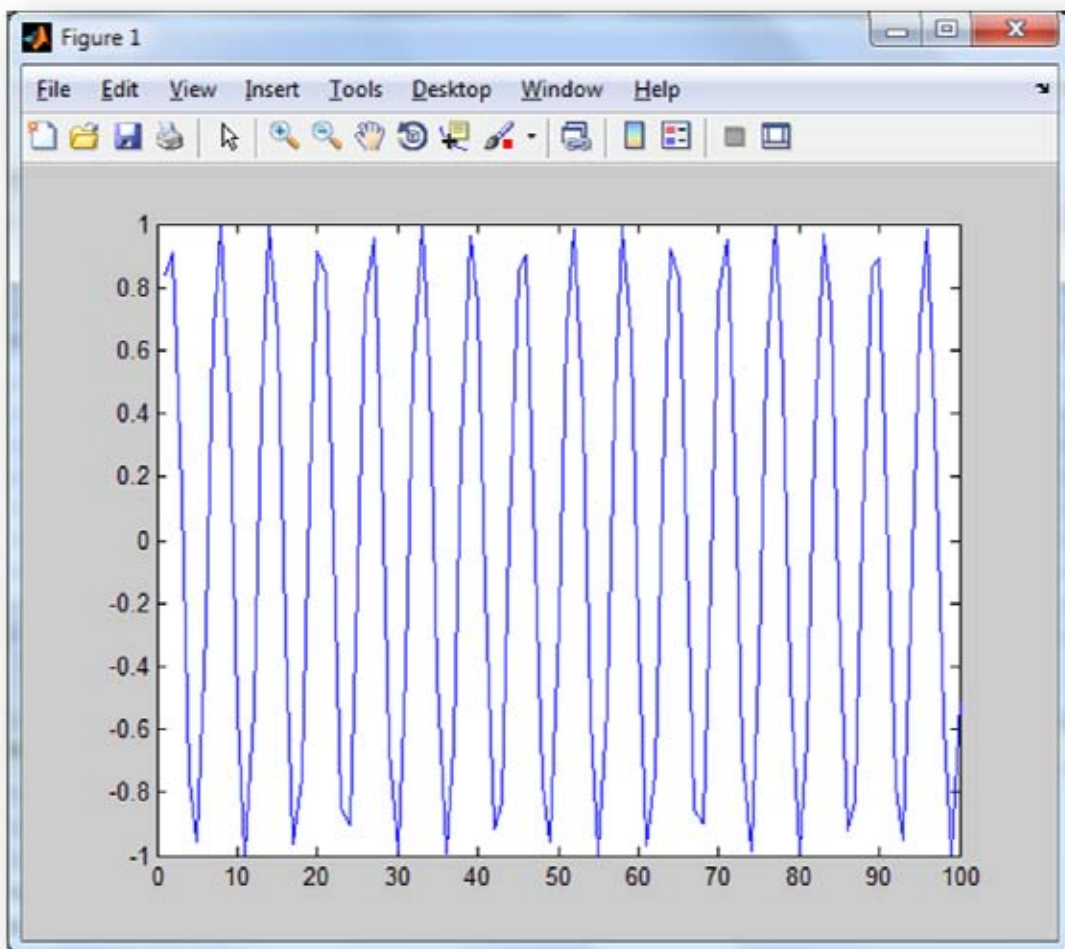


Figura 10.1. Plot di una funzione seno del vettore creato.

Il plot viene automaticamente graficato con le corrette scale sugli assi cartesiani (Figura 10.1).

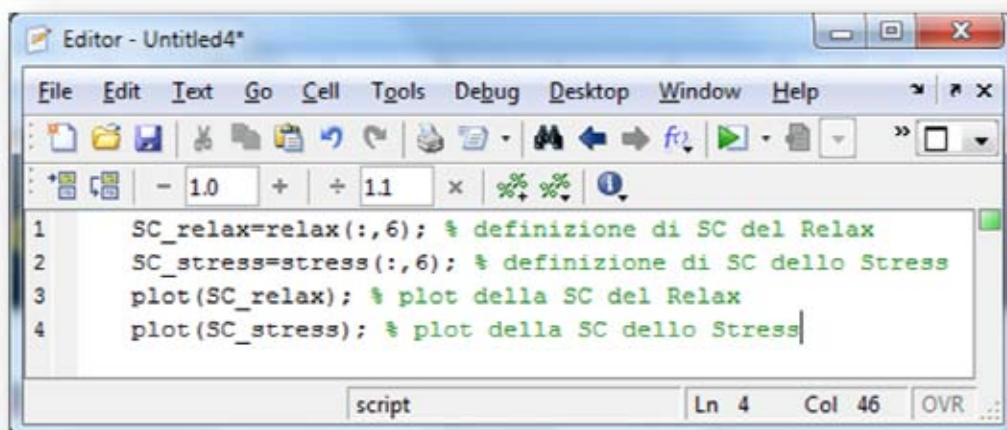
Riprendiamo adesso la sperimentazione vista nel capitolo 2 e usata nel capitolo 8, con le due sessioni: una di relax e una di stress.

La sessione di relax è registrata nella matrice relax, la sessione di stress è registrata nella matrice stress. Per comodità, entrambe le matrici sono disponibili nei materiali supplementari al volume.

Sappiamo di avere registrato la skin conductance nella sesta colonna, ovvero nel linguaggio di Matlab (:,6)

Quindi, la skin conductance del relax sarà data da "SC_relax=relax(:,6)", mentre la skin conductance dello stress sarà data da "SC_stress=stress(:,6)".

Dunque il primo programma per plottare questi dati relativi alle due matrici può sembrare molto semplice, come mostrato in Figura 10.2. In realtà però è totalmente sbagliato in quanto Matlab e gli altri programmi simili (come GNU Octave) sono sequenziali e dunque il secondo comando di plot sovrascrive il primo.



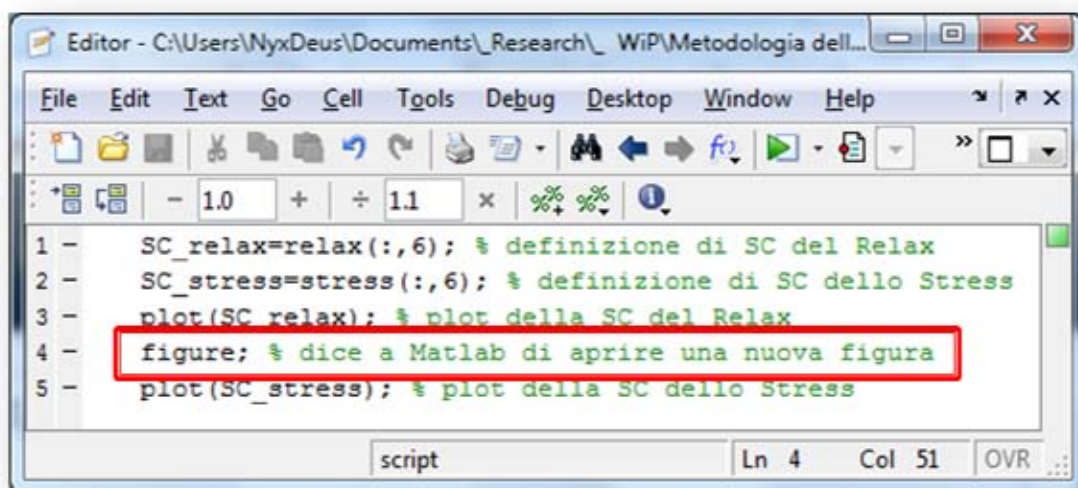
```
1 SC_relax=relax(:,6); % definizione di SC del Relax
2 SC_stress=stress(:,6); % definizione di SC dello Stress
3 plot(SC_relax); % plot della SC del Relax
4 plot(SC_stress); % plot della SC dello Stress
```

Figura 10.2. Plotting "errato" dei due grafici. L'istruzione in riga 4 sovrascrive l'istruzione in riga 3.

Ci sono due modi per risolvere il problema. Dipende da cosa vogliamo ottenere, ovvero:

1. vogliamo due figure separate;
2. oppure vogliamo avere i due grafici nella stessa figura

Se vogliamo avere due figure separate, allora basterà mettere tra un comando e l'altro l'espressione "figure;", come da Figura 10.3.



```
1 - SC_relax=relax(:,6); % definizione di SC del Relax
2 - SC_stress=stress(:,6); % definizione di SC dello Stress
3 - plot(SC_relax); % plot della SC del Relax
4 - figure; % dice a Matlab di aprire una nuova figura
5 - plot(SC_stress); % plot della SC dello Stress
```

Figura 10.3. Il comando "figure;" apre una nuova finestra.

In questo primo caso il programma farà il normale plot della Skin Conductance del Relax nella finestra (riga di codice 3), poi aprirà una nuova finestra ("figure;", riga di codice 4) che a questo punto sarà la nuova finestra di riferimento e infine plotterà nella nuova finestra di riferimento la Skin Conductance dello Stress (riga di codice 5). Il risultato sarà dunque quello di avere due diverse

finestre, una per ogni grafico (Figura 10.4), che potranno essere salvate l'una indipendentemente dall'altra.

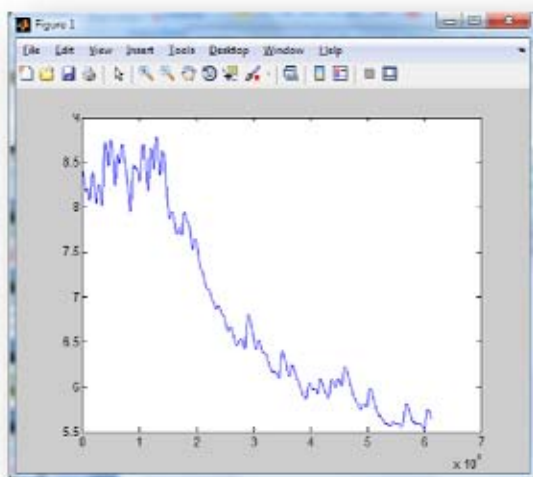


Figure 1
Grafico della skin
conductance per il Relax

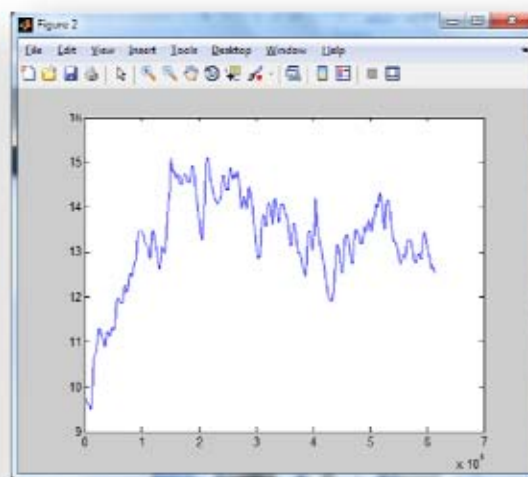
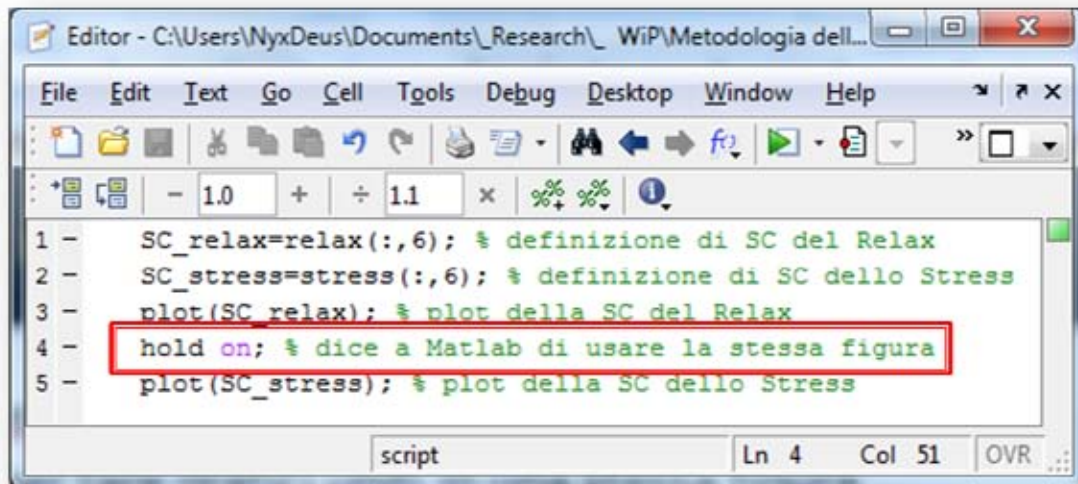


Figure 2
Grafico della skin
conductance per lo Stress

Figura 10.4. Il risultato del programma in Figura 10.3, con le due figure separate.

Se invece vogliamo avere i due grafici nella stessa figura allora basterà mettere tra un comando e l'altro l'espressione "hold on;" (Figura 10.5). In questo secondo caso il programma farà il normale plot della Skin Conductance del Relax nella finestra (riga di codice 3), poi attiverà la stessa finestra ("hold on;", riga di codice 4) che a questo punto resterà la nuova finestra di riferimento per la scrittura del grafico, ma senza sovrascrittura e infine plotterà in

questa stessa finestra di riferimento la Skin Conductance dello Stress (riga di codice 5).



```
1 - SC_relax=relax(:,6); % definizione di SC del Relax
2 - SC_stress=stress(:,6); % definizione di SC dello Stress
3 - plot(SC_relax); % plot della SC del Relax
4 - hold on; % dice a Matlab di usare la stessa figura
5 - plot(SC_stress); % plot della SC dello Stress
```

Figura 10.5. Il comando "hold on;" tiene come riferimento la stessa finestra, evitando la sovrascrittura.

Il risultato sarà dunque quello di avere una stessa finestra con due diversi grafici all'interno (Figura 10.6), che potranno essere salvate l'una indipendentemente dall'altra.

Non esiste un modo giusto e uno sbagliato per scegliere tra queste due modalità. Presentano caratteristiche molto diverse e bisognerà scegliere di volta in volta quello più adatto per rappresentare le proprie esigenze.

Con questa seconda soluzione, ovviamente, non si potranno effettuare due salvataggi differenziati. Il salvataggio è sempre della figura, che in questo caso è comunque unica, seppur

contenente due diversi grafici. Questa distinzione è necessaria, perché la modalità di salvataggio delle figure rappresenta anche il loro futuro recupero. Anche questo va calibrato sulle specifiche necessità.

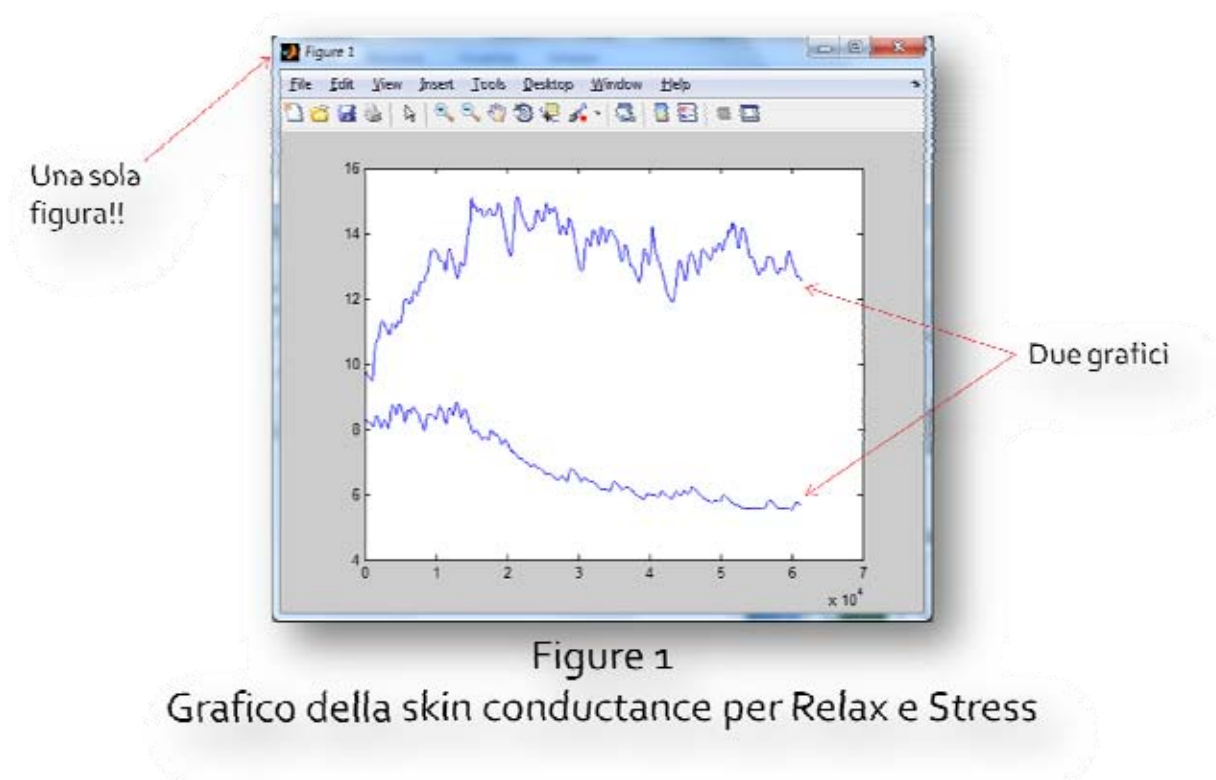


Figura 10.6. Il risultato del programma in Figura 10.5, con i due grafici all'interno della stessa figura.

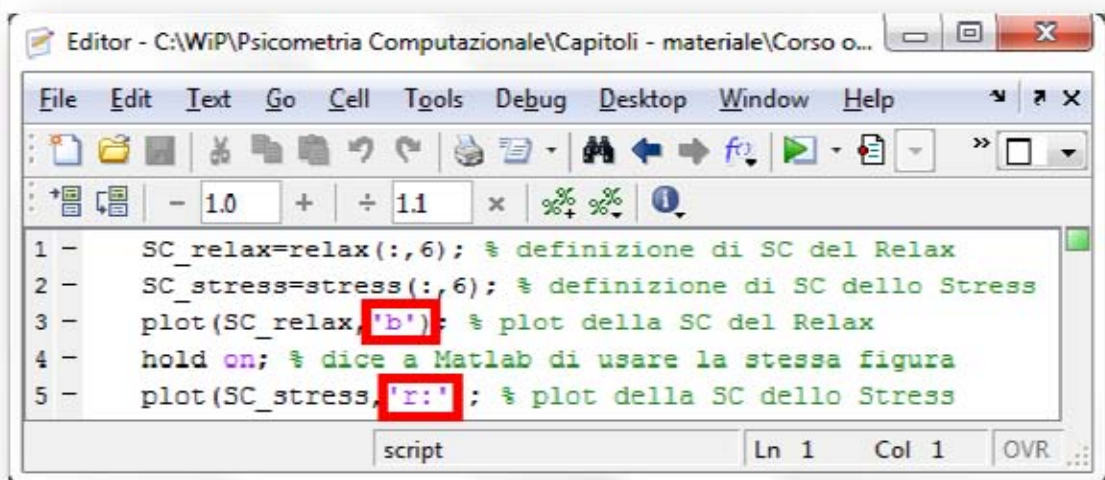
Questa seconda soluzione presenta anche un'altra difficoltà, ovvero il problema a distinguere i due grafici ovvero lo Stress dal Relax. Le due linee sono separate, ma non ho indicazione alcuna in merito a cosa ciascuna linea rappresenti. Una prima soluzione è usare i

colori, ad esempio una linea rossa per lo Stress e una linea blu per il Relax.

In Matlab la linea di un grafico è facilmente personalizzabile con una piccola opzione nell'espressione "plot(X);", basterà infatti usare le espressioni:

```
plot(X,'b'); % traccerà la linea del grafico in blu  
plot(X,'r'); % traccerà la linea del grafico in rosso
```

Ad esempio il codice precedente, diventa come in Figura 10.7.



```
Editor - C:\WIP\Psicometria Computazionale\Capitoli - materiale\Corso o...  
File Edit Text Go Cell Tools Debug Desktop Window Help  
- 1.0 + ÷ 1.1 × % % ?  
1 - SC_relax=relax(:,6); % definizione di SC del Relax  
2 - SC_stress=stress(:,6); % definizione di SC dello Stress  
3 - plot(SC_relax,'b'); % plot della SC del Relax  
4 - hold on; % dice a Matlab di usare la stessa figura  
5 - plot(SC_stress,'r'); % plot della SC dello Stress  
script Ln 1 Col 1 OVR
```

Figura 10.7. Aggiunta delle opzioni colore alla figura precedente.

E' possibile usare diversi tipi di colore, che di volta in volta possiamo usare per indicare alla funzione plot anche questa informazione. Inoltre usando la stessa sintassi, possiamo

aggiungere dopo la lettera del colore un simbolo per cambiare lo stile della linea. In Figura 10.8 alcune opzioni di colore e di stile di linea.

RGB Value	Short Name	Long Name
[1 1 0]	y	yellow
[1 0 1]	m	magenta
[0 1 1]	c	cyan
[1 0 0]	r	red
[0 1 0]	g	green
[0 0 1]	b	blue
[1 1 1]	w	white
[0 0 0]	k	black

Specifier String	Line Style
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dot line
none	No line

Figura 10.8. Simboli di colore e stile di linea per il comando plot.

Usando in modo corretto tutte queste opzioni, possiamo ottenere dei grafici differenziati per colore e per stile di linea. Questa operazione è molto utile soprattutto quando è necessario distinguere tra diverse sessioni sperimentali.

Nel nostro esempio avevamo necessità di distinguere la sessione di Relax da quella di Stress. Usando il codice riportato in Figura 10.7 rendiamo la linea del relax blu (con linea continua, lo stile standard) e la linea dello stress rossa (con linea tratteggiata).

La Figura 10.9, rispetto alla Figura 10.6 ha una più chiara identificazione delle due diverse linee e permette una più rapida individuazione della sperimentazione.

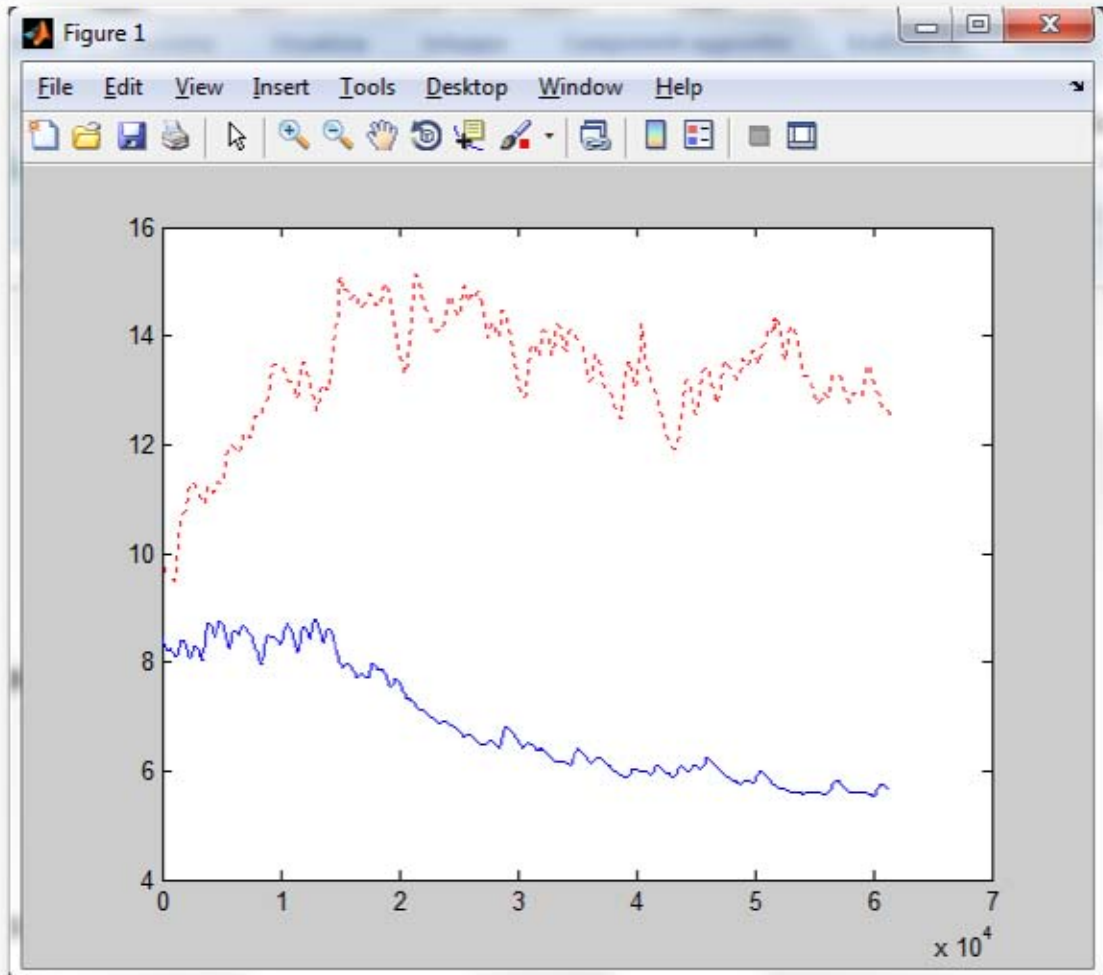
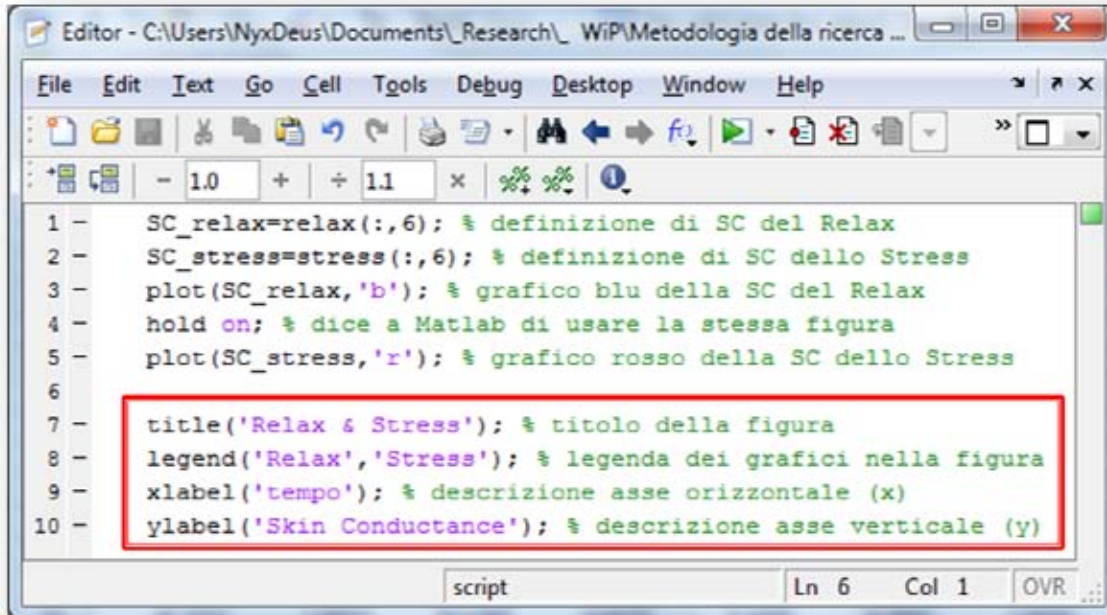


Figura 10.9. Risultato del codice con colori e stili di linea.

La figura è certamente molto più chiara adesso, tuttavia mancano ancora diversi elementi. Matlab permette di aggiungere con semplici espressioni il titolo, la legenda e le etichette per ciascun asse cartesiano, come si può vedere dal codice in Figura 10.10.

Questi elementi permettono una migliore descrizione del nostro grafico e una più precisa identificazione dei vari elementi.



```

1 - SC_relax=relax(:,6); % definizione di SC del Relax
2 - SC_stress=stress(:,6); % definizione di SC dello Stress
3 - plot(SC_relax,'b'); % grafico blu della SC del Relax
4 - hold on; % dice a Matlab di usare la stessa figura
5 - plot(SC_stress,'r'); % grafico rosso della SC dello Stress
6
7 - title('Relax & Stress'); % titolo della figura
8 - legend('Relax','Stress'); % legenda dei grafici nella figura
9 - xlabel('tempo'); % descrizione asse orizzontale (x)
10 - ylabel('Skin Conductance'); % descrizione asse verticale (y)

```

Figura 10.10. Aggiunta di titolo, legenda ed etichette degli assi.

Adesso che abbiamo arricchito ulteriormente la figura, possiamo definire in modo più preciso le informazioni in essa contenute, prima fra tutte la definizione e visualizzazione corretta del tempo nell'asse delle ascisse (orizzontale).

Nella nostra figura il tempo non è infatti specificato e viene considerato in automatico da Matlab il numero di righe.

Ma se abbiamo un vettore (riga o colonna) dei tempi, allora basterà plottare invece di, per esempio, "plot(X)", una nuova istruzione "plot(t,X)", dove t è un vettore dei tempi, ad esempio:

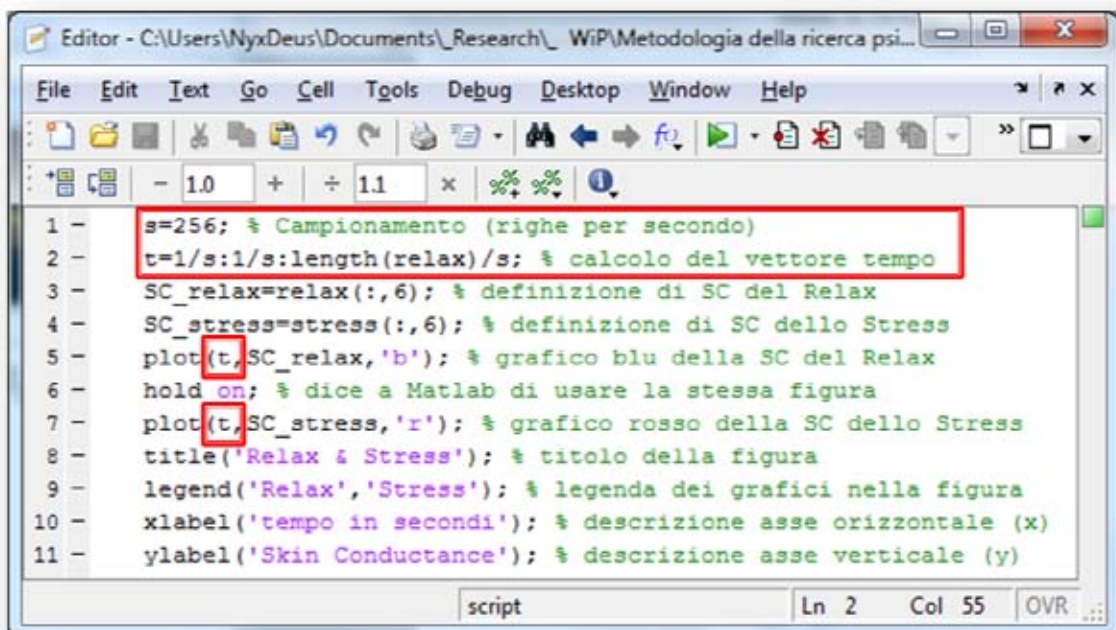
```
t=1/256:1/256:length(X)/256 % definizione del tempo come vettore
```

Nel nostro caso avremo un vettore dei tempi unico, perchè le matrici "relax" e "stress" sono della stessa dimensione.

Basterà dunque scrivere (vedi capitolo 8):

```
s=256; % Sampling rate
t=1/s:1/s:length(relax)/s; % Vettore dei tempi su relax e stress
```

Il nostro codice quindi si trasforma livemente per includere un asse orizzontale del tempo basato sulle matrici (Figura 10.11).



```
Editor - C:\Users\NyxDeus\Documents\Research\ WiP\Metodologia della ricerca psi...
File Edit Text Go Cell Tools Debug Desktop Window Help
- 1.0 + ÷ 1.1 x
1 - s=256; % Campionamento (righe per secondo)
2 - t=1/s:1/s:length(relax)/s; % calcolo del vettore tempo
3 - SC_relax=relax(:,6); % definizione di SC del Relax
4 - SC_stress=stress(:,6); % definizione di SC dello Stress
5 - plot(t, SC_relax, 'b'); % grafico blu della SC del Relax
6 - hold on; % dice a Matlab di usare la stessa figura
7 - plot(t, SC_stress, 'r'); % grafico rosso della SC dello Stress
8 - title('Relax & Stress'); % titolo della figura
9 - legend('Relax', 'Stress'); % legenda dei grafici nella figura
10 - xlabel('tempo in secondi'); % descrizione asse orizzontale (x)
11 - ylabel('Skin Conductance'); % descrizione asse verticale (y)
script Ln 2 Col 55 OVR
```

Figura 10.11. Codice con la definizione del tempo.

A differenza delle modifiche di cui prima, questa nuova informazione è sostanziale, perché permette una migliore

leggibilità dei dati, esprimendo il tempo in secondi e questo permette una migliore comprensione del grafico e quindi della sperimentazione (Figura 10.12).

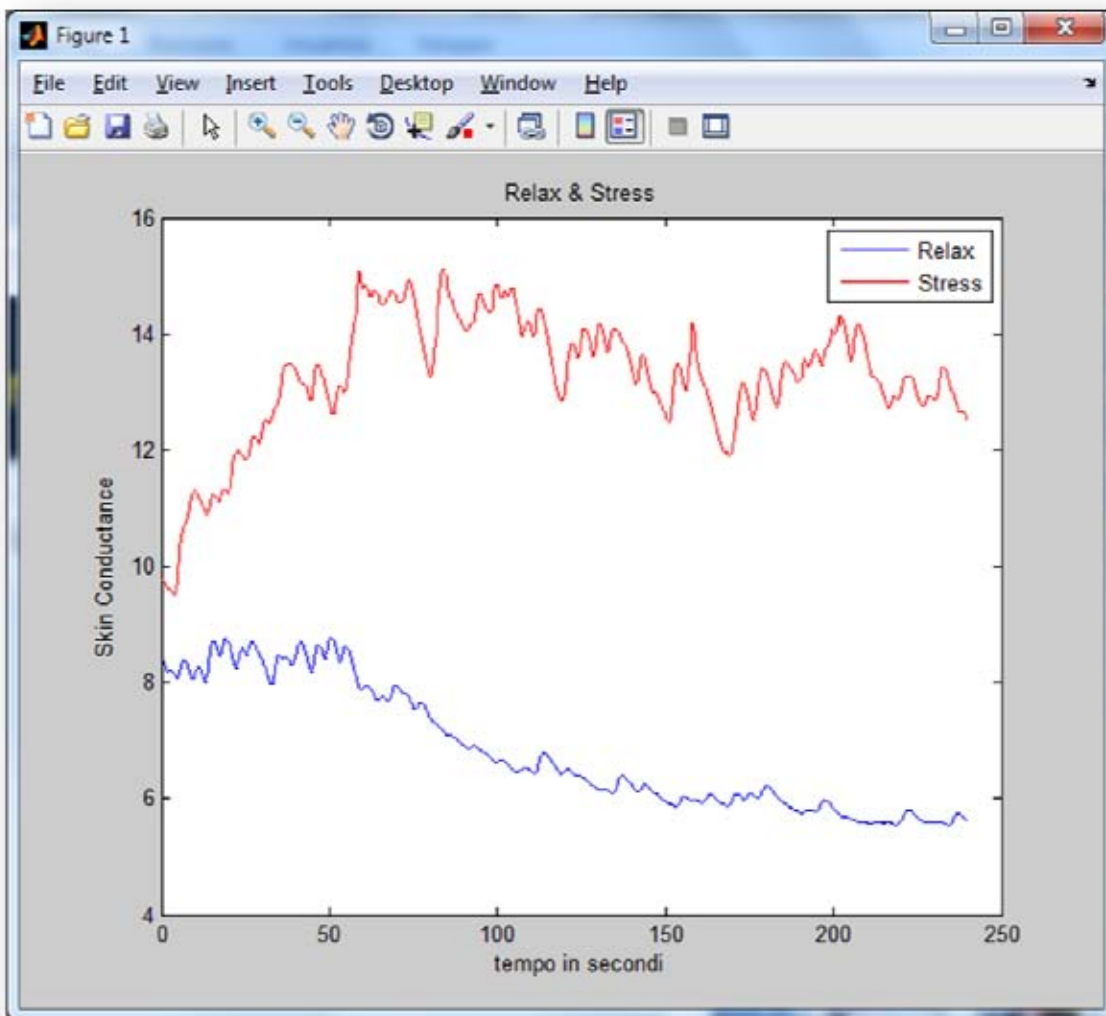
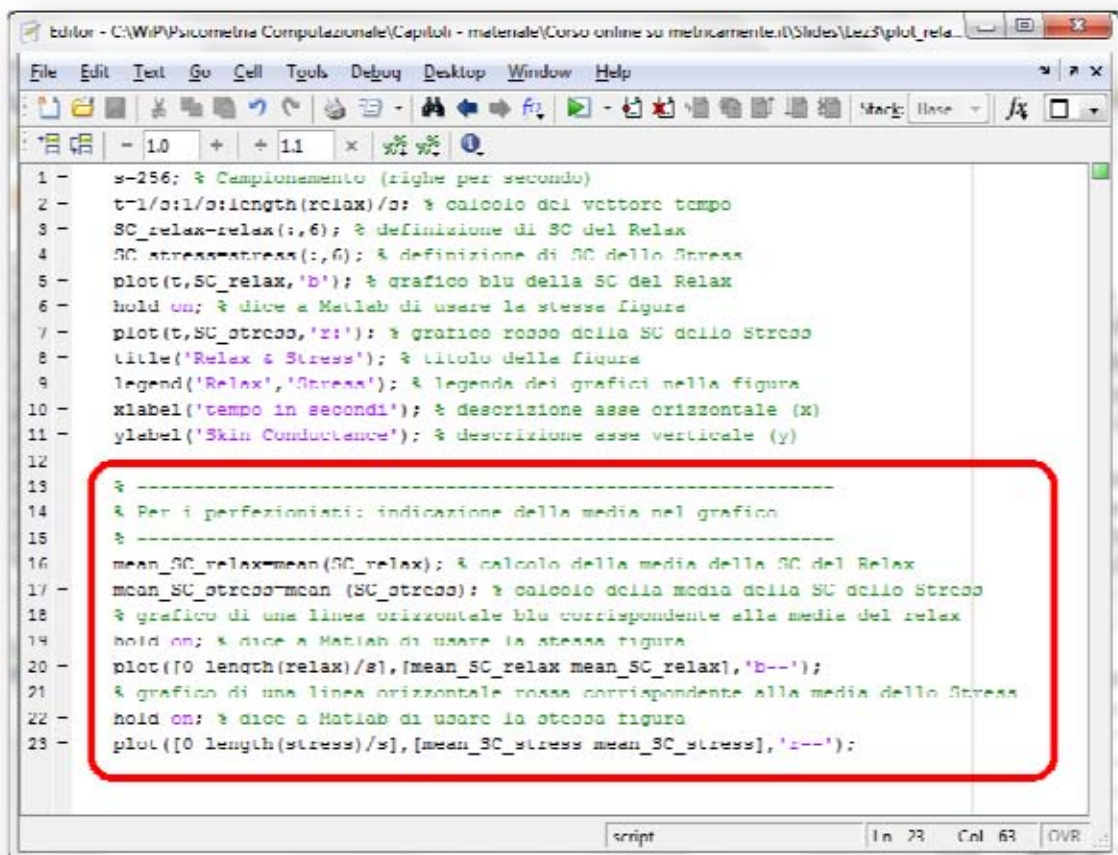


Figura 10.12. Indicazione del tempo in secondi (asse orizzontale).

A questo punto il grafico risulta molto completo, sia graficamente, sia nella sostanza dei dati e nella loro rappresentazione.

Per completezza, dal momento che al meglio non c'è mai fine, lascio anche qualche spunto di riflessione con l'espansione del codice per includere due linee orizzontali tratteggiate in corrispondenza della media delle due sessioni.



```

1 - s=256; % Campionamento (righe per secondo)
2 - t=1/s:1/s:length(rclax)/s; % calcolo del vettore tempo
3 - SC_relax=relax(:,6); % definizione di SC del Relax
4 - SC_stress=stress(:,6); % definizione di SC dello Stress
5 - plot(t,SC_relax,'b'); % grafico blu della SC del Relax
6 - hold on; % dice a Matlab di usare la stessa figura
7 - plot(t,SC_stress,'r'); % grafico rosso della SC dello Stress
8 - title('Relax & Stress'); % titolo della figura
9 - legend('Relax','Stress'); % legenda dei grafici nella figura
10 - xlabel('tempo in secondi'); % descrizione asse orizzontale (x)
11 - ylabel('Skin Conductance'); % descrizione asse verticale (y)
12
13 % -----
14 % Per i perfezionisti: indicazione della media nel grafico
15 % -----
16 mean_SC_relax=mean(SC_relax); % calcolo della media della SC del Relax
17 mean_SC_stress=mean(SC_stress); % calcolo della media della SC dello Stress
18 % grafico di una linea orizzontale blu corrispondente alla media del relax
19 hold on; % dice a Matlab di usare la stessa figura
20 plot([0 length(relax)/s],[mean_SC_relax mean_SC_relax],'b--');
21 % grafico di una linea orizzontale rossa corrispondente alla media dello Stress
22 hold on; % dice a Matlab di usare la stessa figura
23 plot([0 length(stress)/s],[mean_SC_stress mean_SC_stress],'r--');

```

Figura 10.13. Codice per l'indicazione delle medie nel grafico.

Non è un codice di immediata comprensione ma se lo esaminate bene, vedrete che avete già, dalle lezioni precedenti, tutte le competenze per poterlo capire e anche realizzare autonomamente.

La figura risultante, comprenderà anche le medie per sessione, in tratteggio, come da Figura 10.14.

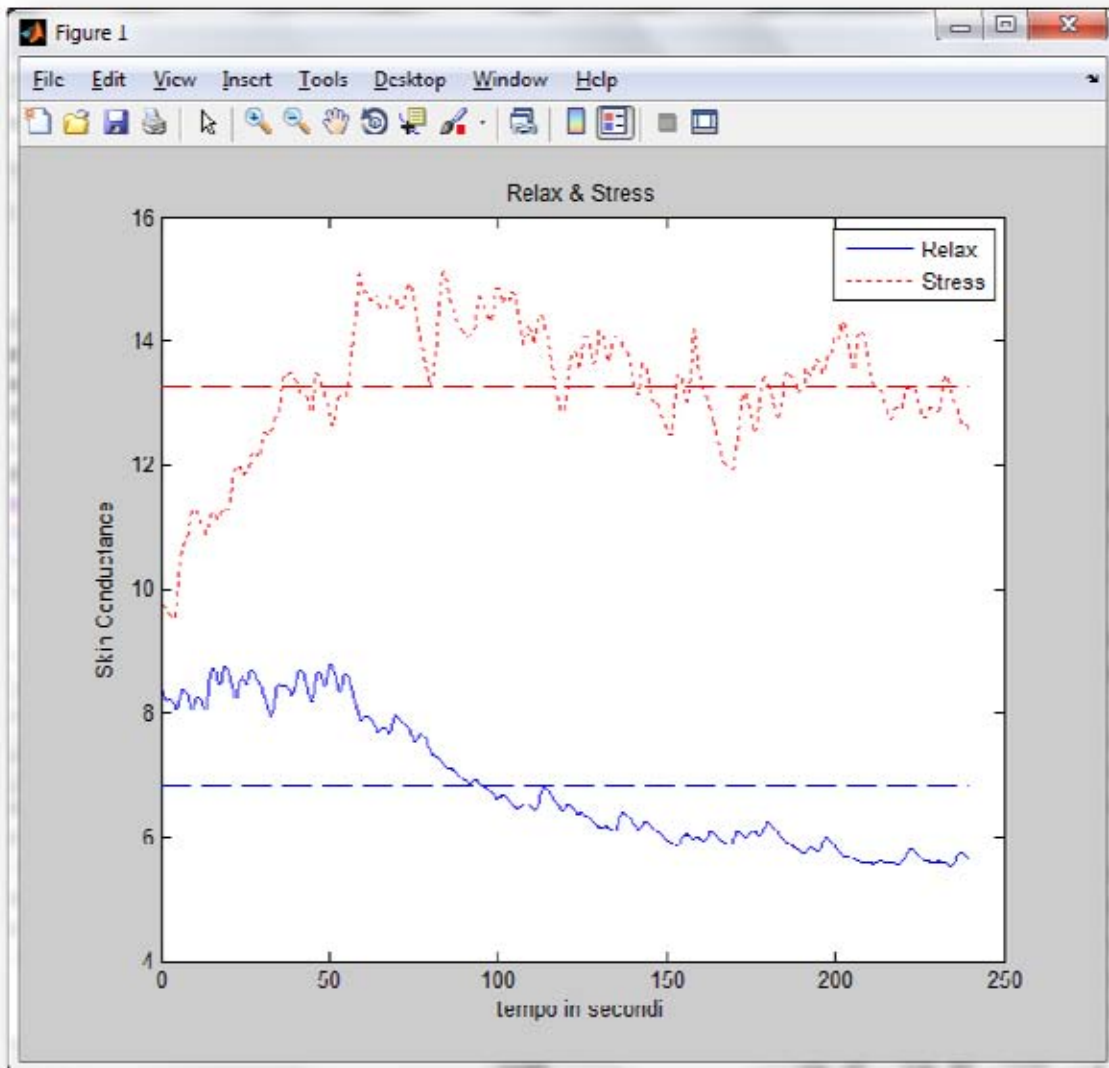


Figura 10.14. Inclusione delle medie per sessione con tratteggio.

Proviamo adesso a pensare se si possono fare gli stessi perfezionamenti quando si lavora a figure separate.

Ovviamente, la risposta è positiva. Bisognerà però ripetere e dividere le specifiche (titoli, legenda, descrizione assi) per ognuna delle figure generate, ovvero, dopo ogni comando "plot" e ovviamente bisogna dare il comando "figure;" tra i due plot, come da Figura 10.15.

```

1 - s=256; % Campionamento (righe per secondo)
2 - t=1/s:1/s:length(relax)/s; % calcolo del vettore tempo
3 - SC_relax=relax(:,6); % definizione di SC del Relax
4 - SC_stress=stress(:,6); % definizione di SC dello Stress
5
6 - plot(t,SC_relax,'b'); % grafico blu della SC del Relax
7 - title('Relax'); % titolo della figura
8 - xlabel('tempo in secondi'); % descrizione asse orizzontale (x)
9 - ylabel('Skin Conductance'); % descrizione asse verticale (y)
10 - legend('Relax'); % legenda del grafico nella figura
11
12 - figure; % dice a Matlab di usare un'altra figura
13
14 - plot(t,SC_stress,'r'); % grafico rosso della SC dello Stress
15 - title('Stress'); % titolo della figura
16 - legend('Stress'); % legenda del grafico nella figura
17 - xlabel('tempo in secondi'); % descrizione asse orizzontale (x)
18 - ylabel('Skin Conductance'); % descrizione asse verticale (y)

```

Come prima

Primo plot

Fa cambiare figura

Secondo plot

Figura 10.15. Codice completo per le figure separate.

In pratica il codice non cambia considerevolmente, basterà un minimo di accortezza nel cambiare i parametri principali, per poter usare di fatto sempre un codice simile. Nel materiale supplementare al volume ci sono tutti i file .m per realizzare plot completi, sebbene suggerisco di fare alcune prove in autonomia.

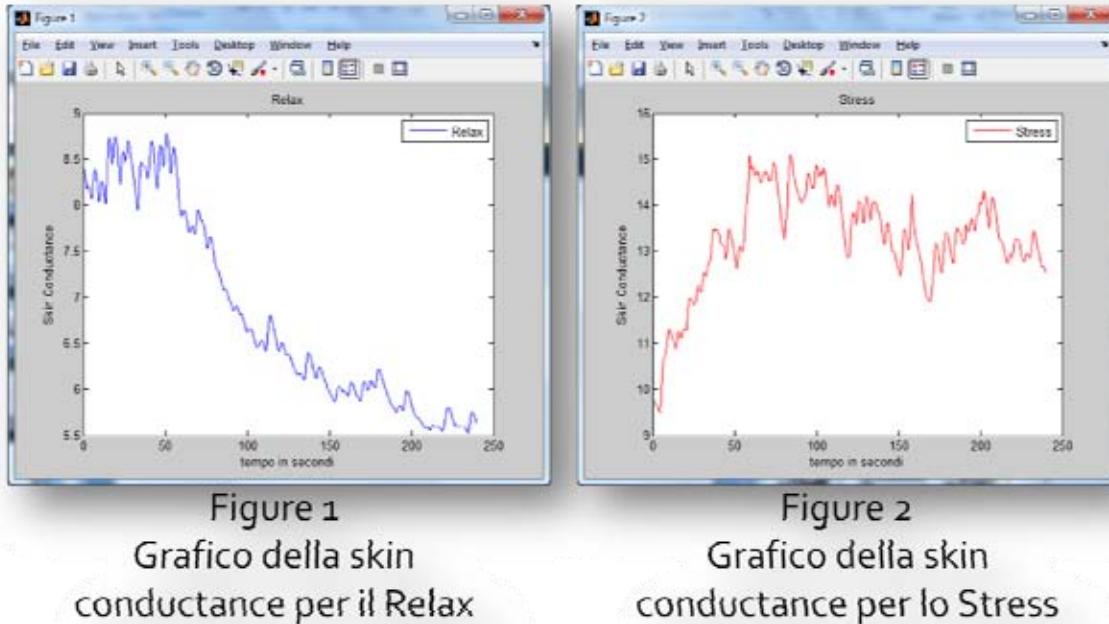


Figura 10.16. Figure separate complete in tutti i dettagli.

10.2 Subplotting

Il subplot sono delle figure che contengono al loro interno delle altre figure totalmente indipendenti fra loro, secondo una struttura predefinita.

Il subplot e può essere fatto antepoendo un semplice comando a ciascun plot:

```
subplot(2,1,1) % per il primo plot
```

...

```
subplot(2,1,2) % per il secondo plot
```

...

dove i tre parametri contenuti nella sintassi del subplot sono spiegati in Figura 10.17.

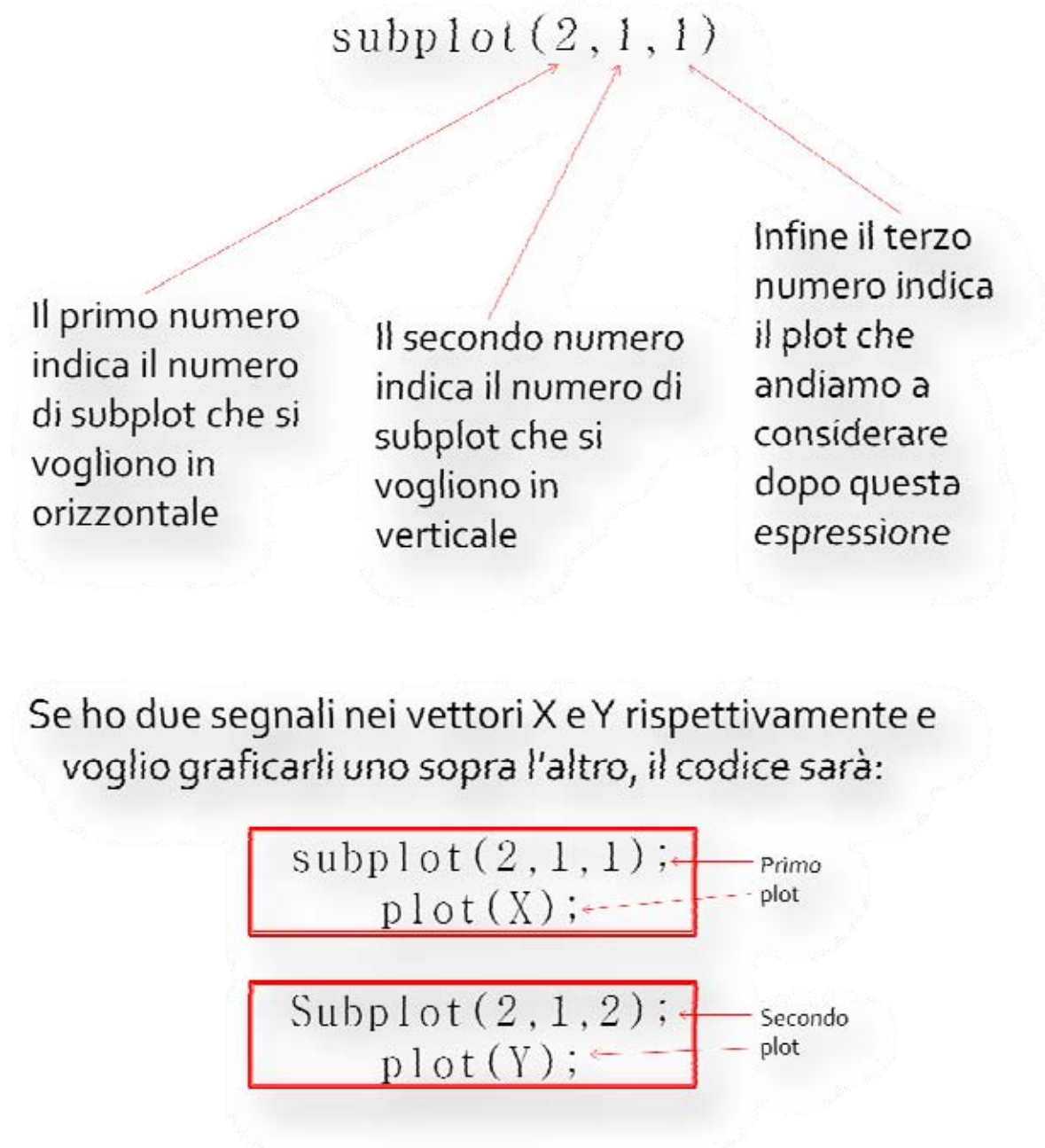
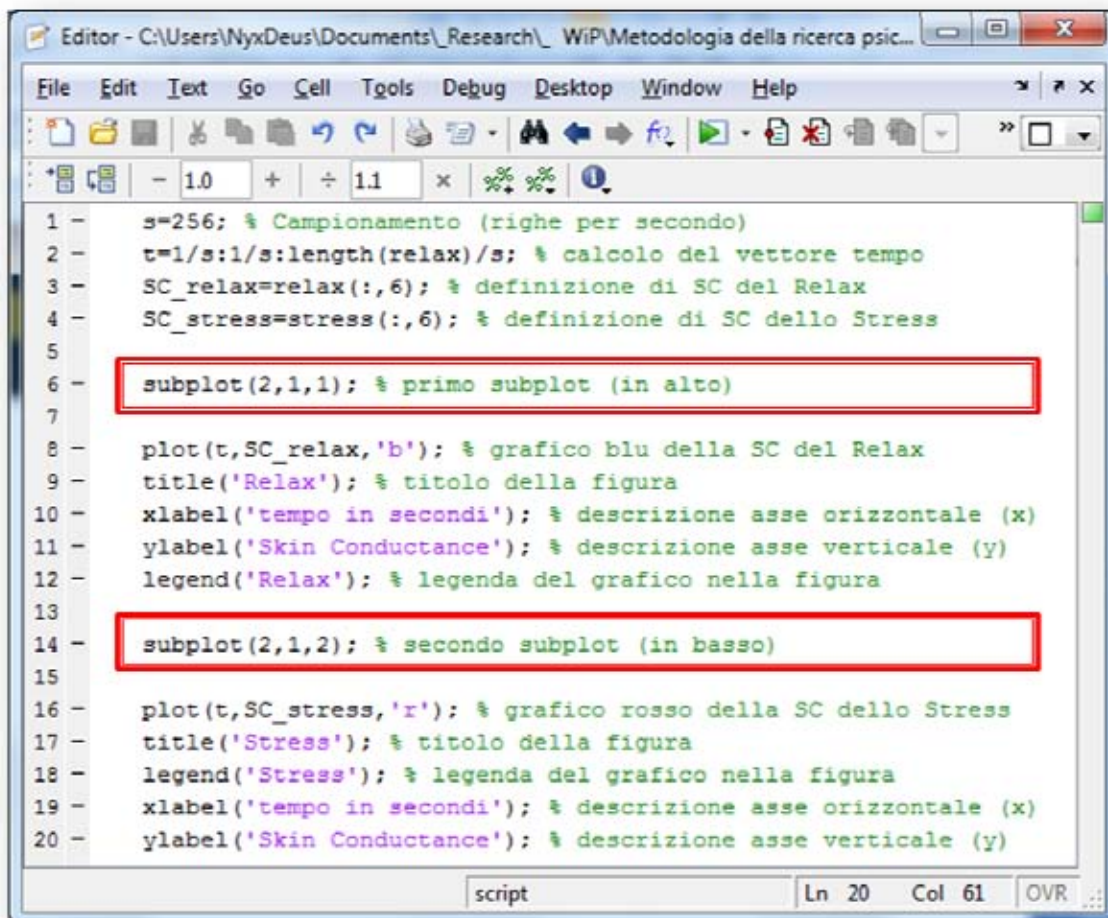


Figura 10.17. Sintassi del subplot, tramite la quale possiamo scomporre una figura in più sotto-figure rappresentanti diversi segnali, coerenti con la sperimentazione.

Il subplot è dunque molto facile da fare e basterà aggiungere un paio di righe al precedente codice (Figura 10.18) per ottenere questo nuovo tipo di visualizzazione (Figura 10.19).



```
1 - s=256; % Campionamento (righe per secondo)
2 - t=1/s:1/s:length(relax)/s; % calcolo del vettore tempo
3 - SC_relax=relax(:,6); % definizione di SC del Relax
4 - SC_stress=stress(:,6); % definizione di SC dello Stress
5
6 - subplot(2,1,1); % primo subplot (in alto)
7
8 - plot(t,SC_relax,'b'); % grafico blu della SC del Relax
9 - title('Relax'); % titolo della figura
10 - xlabel('tempo in secondi'); % descrizione asse orizzontale (x)
11 - ylabel('Skin Conductance'); % descrizione asse verticale (y)
12 - legend('Relax'); % legenda del grafico nella figura
13
14 - subplot(2,1,2); % secondo subplot (in basso)
15
16 - plot(t,SC_stress,'r'); % grafico rosso della SC dello Stress
17 - title('Stress'); % titolo della figura
18 - legend('Stress'); % legenda del grafico nella figura
19 - xlabel('tempo in secondi'); % descrizione asse orizzontale (x)
20 - ylabel('Skin Conductance'); % descrizione asse verticale (y)
```

Figura 10.18. Codice a figure separate adattato con il subplot.

I subplot sono molto utili:

- quando si vogliono graficare diversi tipi di segnale (non solo SC) in un'unica figura;

- quando si devono ispezionare visivamente i segnali e si vogliono vedere parti diverse dello stesso segnale (diversi momenti della sperimentazione);
- per altre motivi estetici e/o funzionali, che possano migliorare la comprensione e l'interpretazione dei dati, o la loro analisi.

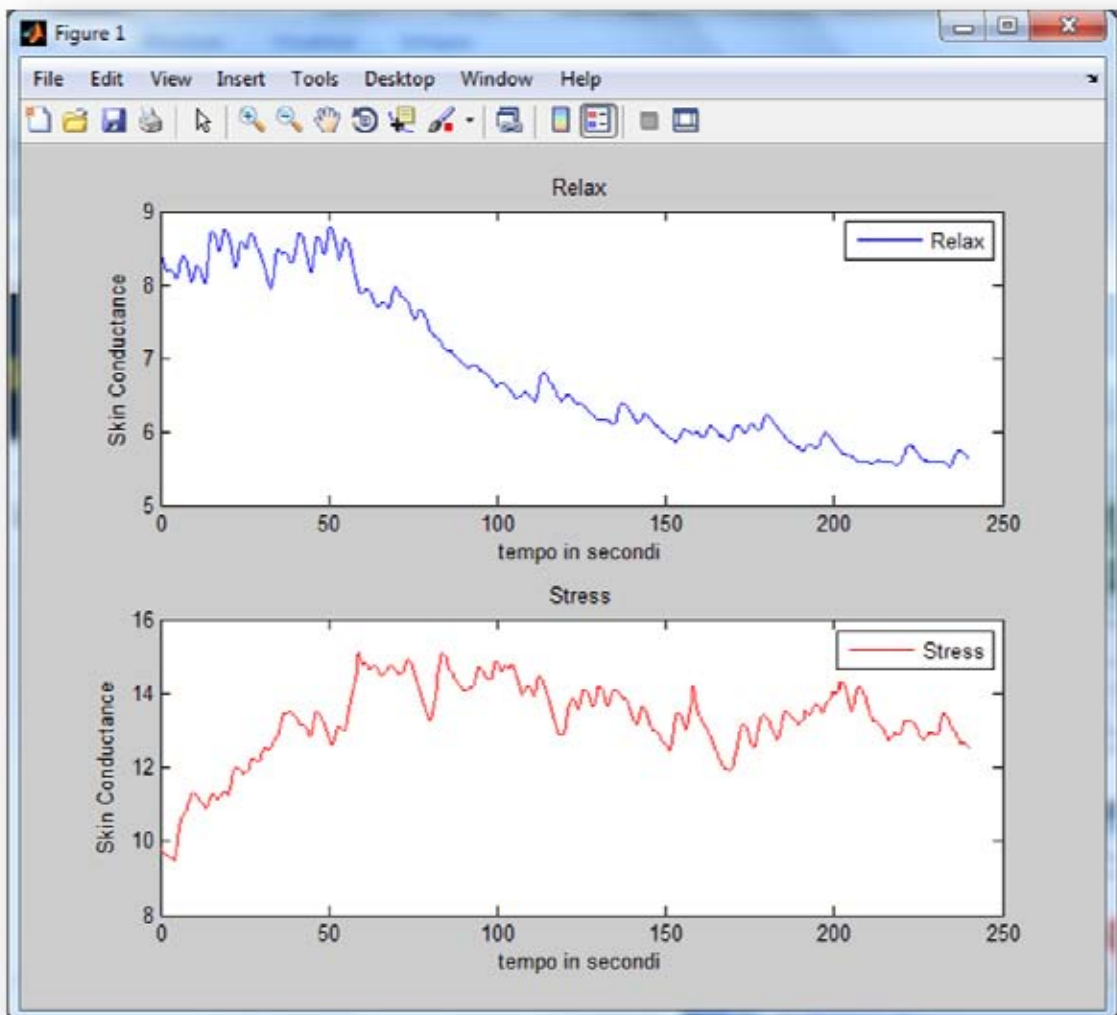


Figura 10.19. figure separate con due subplot, uno per la sessione di Relax e uno per quella di Stress.

In conclusione del paragrafo consideriamo un'altra importantissima funzione per la gestione dei plot: il salvataggio. Per ogni figura, si può sempre procedere cliccando su "File > Save As...", tuttavia può essere molto utile effettuare questa operazione tramite codice Matlab. In figura 10.20, indichiamo la spiegazione della sintassi relativa al salvataggio di una figura nella cartella "C:\figs", con nome "data1_vs_data2" ed estensione ".fig" (il formato Matlab).

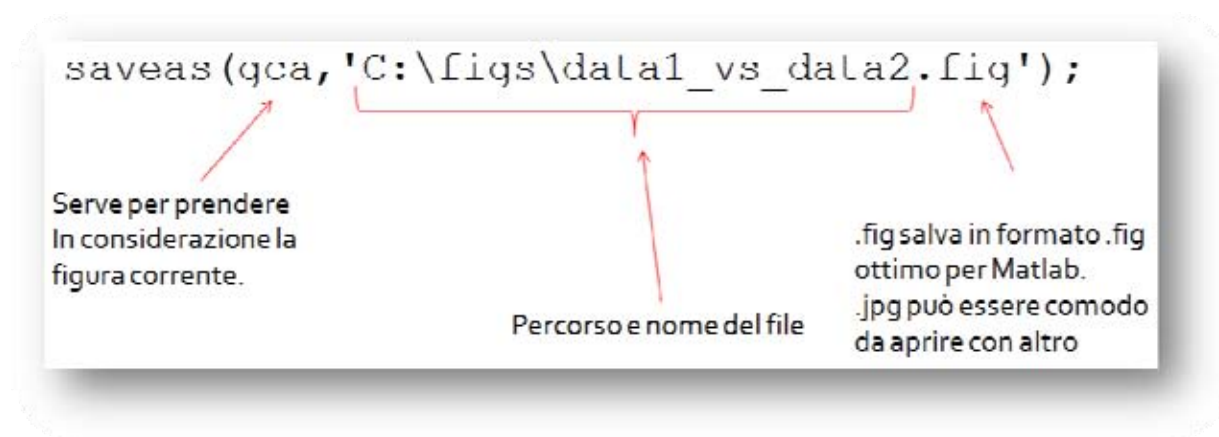


Figura 10.20. Sintassi del comando saveas.

Sebbene in questo momento può non sembrarci critica la scelta di salvare con il menu piuttosto che tramite codice, più avanti vedremo come la possibilità di inserire salvataggi nel codice che creiamo, può essere molto utile e farci risparmiare molto tempo e fatica. In particolare già adesso, con le conoscenze acquisite nei precedenti capitoli, si può immaginare di usare una stampa unione per creare diverse centinaia di grafici e salvarle in locale, secondo le indicazioni fornite.

In generale, come possiamo vedere in Figura 10.21, una volta inserito il codice in un file .m è possibile avviarlo e ritrovare i file delle immagini nella cartella creata, con i rispettivi nomi indicati.

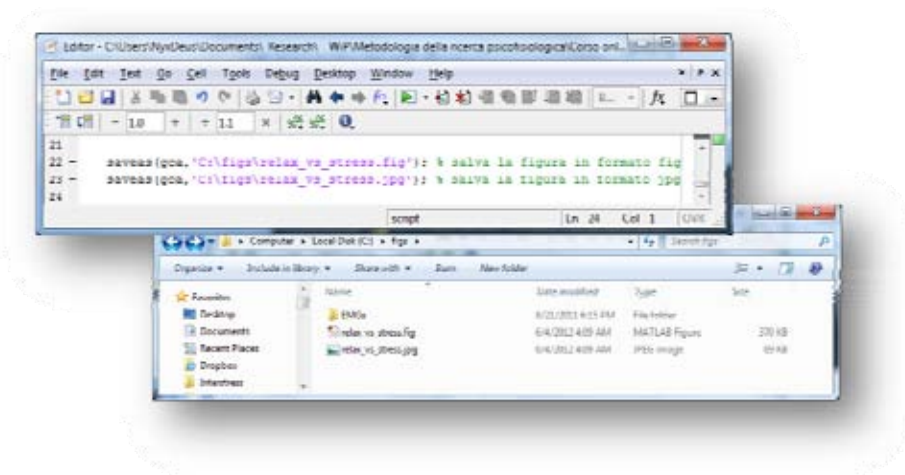


Figura 10.21. Salvataggio della figura corrente in formato .fig e .jpg e loro visualizzazione nella cartella "C:\figs\".

Gestione di dati complessi

In questa seconda parte del volume abbiamo visto come gestire e manipolare dati complessi con software professionali. In particolare nel capitolo 9 abbiamo visto come eseguire anche operazioni multiple su più dati. Tuttavia l'organizzazione di dati complessi a volte può richiedere strumenti e strutture più avanzate di quelle che abbiamo fin qui visto. In questo capitolo approfondiamo ulteriormente le nostre competenze di manipolazione dei dati, usando strutture logiche molto più complesse di quelle finora viste, ma che tuttavia ci semplificheranno molto il lavoro e ci permetteranno di fare molte più operazioni complesse, con molto meno codice rispetto a quello che avremmo dovuto usare con le competenze finora acquisite. Obiettivo di questo capitolo è dunque l'approfondimento e la semplificazione di operazioni di intervento avanzato sui dati.

11.1 Condizioni e iterazioni

Gli strumenti finora usati sulle matrici eseguono alcuni comandi in sequenza, senza condizioni che fermino questo processo o lo vincolino a determinate esigenze. Tuttavia nella pratica sperimentale succede spesso di dover sottoporre condizioni di varia natura e specie, quindi in questo paragrafo approfondiamo questa tipologia di istruzioni, atte a contenere i comandi già visti. Innanzitutto cerchiamo di capire l'idea che sta alla base della condizione, aiutandoci con dei diagrammi a blocco (Figura 11.1).

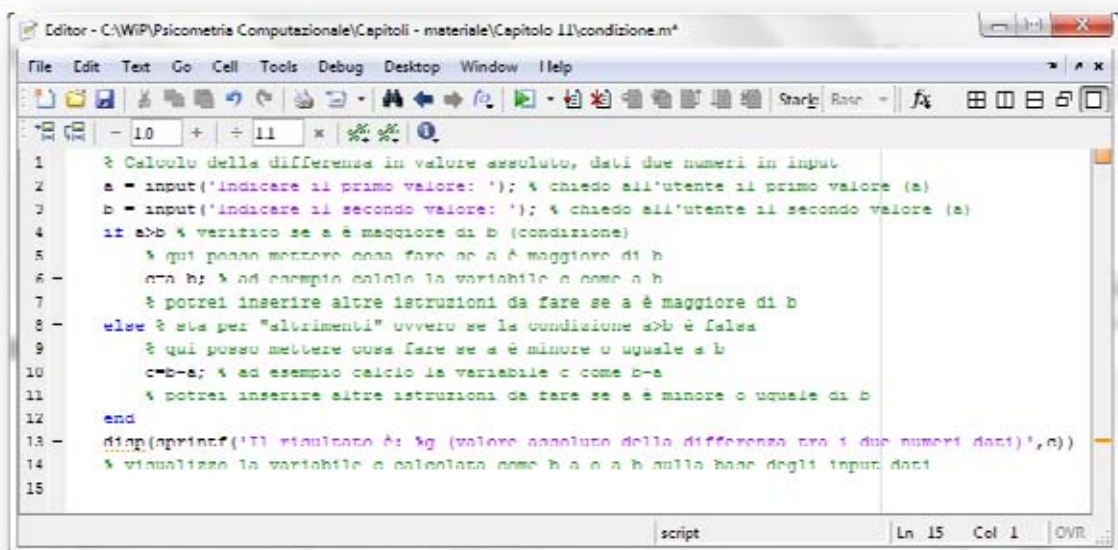


Figura 11.1. Diagramma a blocchi di una condizione.

Abbiamo qui definito delle variabili di input (che magari sono già disponibili nel Workplace di Matlab), una condizione di confronto

tra queste variabili, una serie di comandi ("Fai qualcosa") e una fine, che potrebbe anche essere l'inizio di una nuova procedura.

Implementare l'algoritmo in Figura 11.1 in Matlab è abbastanza semplice: per chiedere le variabili useremo il comando "input", invece per la condizione il comando "if" seguito dalla condizione e dalle istruzioni se la condizione è vera ed "else" seguito da altre istruzioni se la condizione è falsa. In Figura 11.2 un piccolo esempio d'uso della condizione "if".



```

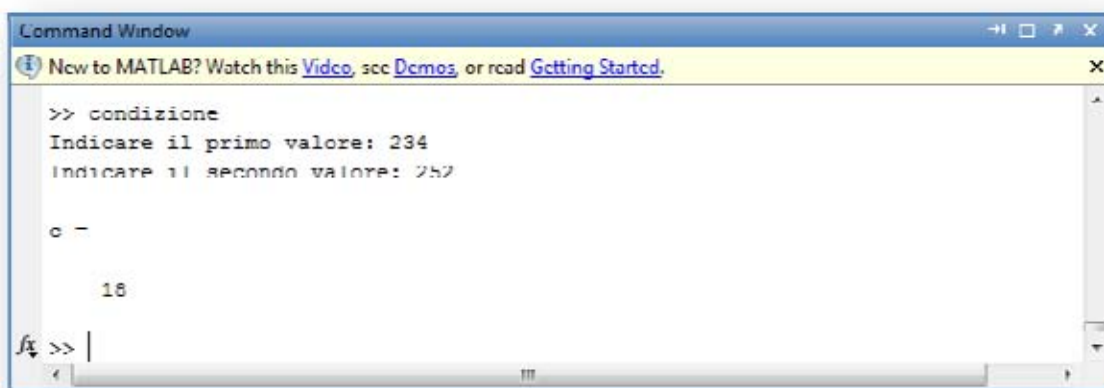
1  % Calcolo della differenza in valore assoluto, dati due numeri in input.
2  a = input('indicare il primo valore: '); % chiedo all'utente il primo valore (a)
3  b = input('indicare il secondo valore: '); % chiedo all'utente il secondo valore (a)
4  if a>b % verifico se a è maggiore di b (condizione)
5      % qui posso mettere cosa fare se a è maggiore di b
6      c=a-b; % ad esempio calcolo la variabile c come a-b
7      % potrei inserire altre istruzioni da fare se a è maggiore di b
8  else % sta per "altrimenti" ovvero se la condizione a>b è falsa
9      % qui posso mettere cosa fare se a è minore o uguale a b
10     c=b-a; % ad esempio calcolo la variabile c come b-a
11     % potrei inserire altre istruzioni da fare se a è minore o uguale di b
12 end
13 disp(sprintf('Il risultato è: %g (valore assoluto della differenza tra i due numeri dati)',c))
14 % visualizzo la variabile c calcolata come b-a o a-b sulla base degli input dati
15

```

Figura 11.2. Uso della condizione "if" per il calcolo della differenza in valore assoluto, dati due numeri in input.

In Matlab in realtà esiste già una funzione per calcolare il valore assoluto (abs) senza dovere controllare quale dei due numeri è maggiore per non ottenere un valore negativo.

Ho cercato comunque di fornire un piccolo esempio, facile da capire, per spiegare la condizione "if". Chi invece volesse calcolare il valore assoluto per altre esigenze può usare le righe 2 e 3 del codice per ricevere l'input, aggiungere la semplice istruzione "c=abs(b-a)" e infine inserire "c" alla successiva riga, per visualizzare il risultato. Che si esegui così o tramite il file .m di cui sopra, il risultato sarà sempre quello in Figura 11.3.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> condizione
Indicare il primo valore: 234
Indicare il secondo valore: 252

c =

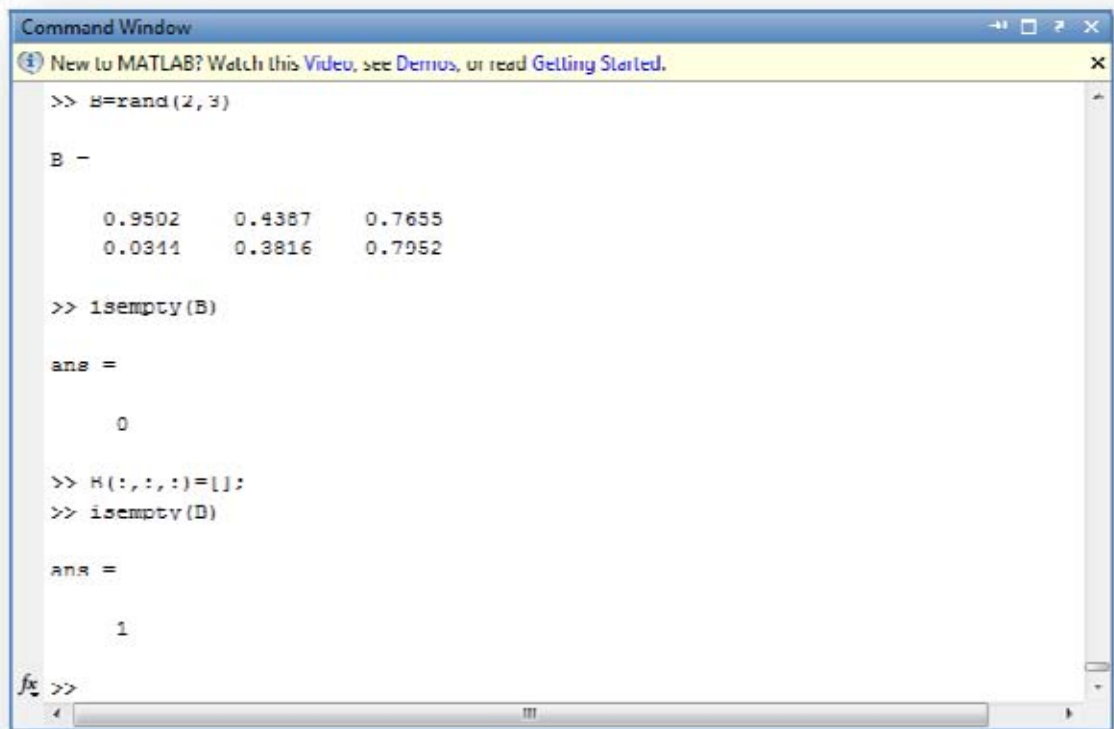
    18

fx >> |
```

Figura 11.3. Risultato del programma con la richiesta dei due valori all'utente e calcolo del valore assoluto della differenza tra i due numeri dati.

Una condizione che ci potrà tornare molte volte assai utile in psicometria computazionale è quella che usa l'istruzione "isempty", che ci servirà per capire se una determinata variabile è vuota o no. In pratica, "isempty(X)" restituirà 0 se la variabile X contiene dei valori, oppure 1 se la variabile X è vuota. Se la variabile è vuota eviteremo di effettuare operazioni su di essa.

Per far capire meglio il funzionamento dell'istruzione `isempty`, come da Figura 11.4, ho prima creato una matrice casuale `B`, di dimensioni `2x3` dunque piena e poi ne ho cancellato le tre colonne rendendo `B` una struttura vuota, sulla quale non avrebbe senso effettuare manipolazioni dei dati.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> B=rand(2,3)

B =
    0.9502    0.4387    0.7655
    0.0314    0.3816    0.7952

>> isempty(B)

ans =

     0

>> B(:, :, :) = [];
>> isempty(B)

ans =

     1

fx >>
```

Figura 11.4. Esempio d'uso del comando `isempty`.

Il comando `isempty` può sembrare inutile e invece è spesso davvero indispensabile in quanto previene interruzioni del programma da errori. Quando ad esempio importiamo in Matlab una lunga serie di dati, può capitare che alcuni di questi contengano strutture vuote

ed è indispensabile in tal caso inserire una condizione per evitare l'elaborazione su questi dati vuoti, usando "if isempty(...)==0" dove al posto dei tre punti di sospensione ci sia la struttura che si vuole interrogare circa la presenza di dati.

Un'altra struttura che useremo molto in psicometria computazionale è il ciclo, che ci consente di eseguire una serie di iterazioni (da iter, da non confondere con interazioni!) sulla base delle specifiche esigenze. In Figura 11.5 vediamo l'istruzione di iterazione con i diagrammi a blocchi.

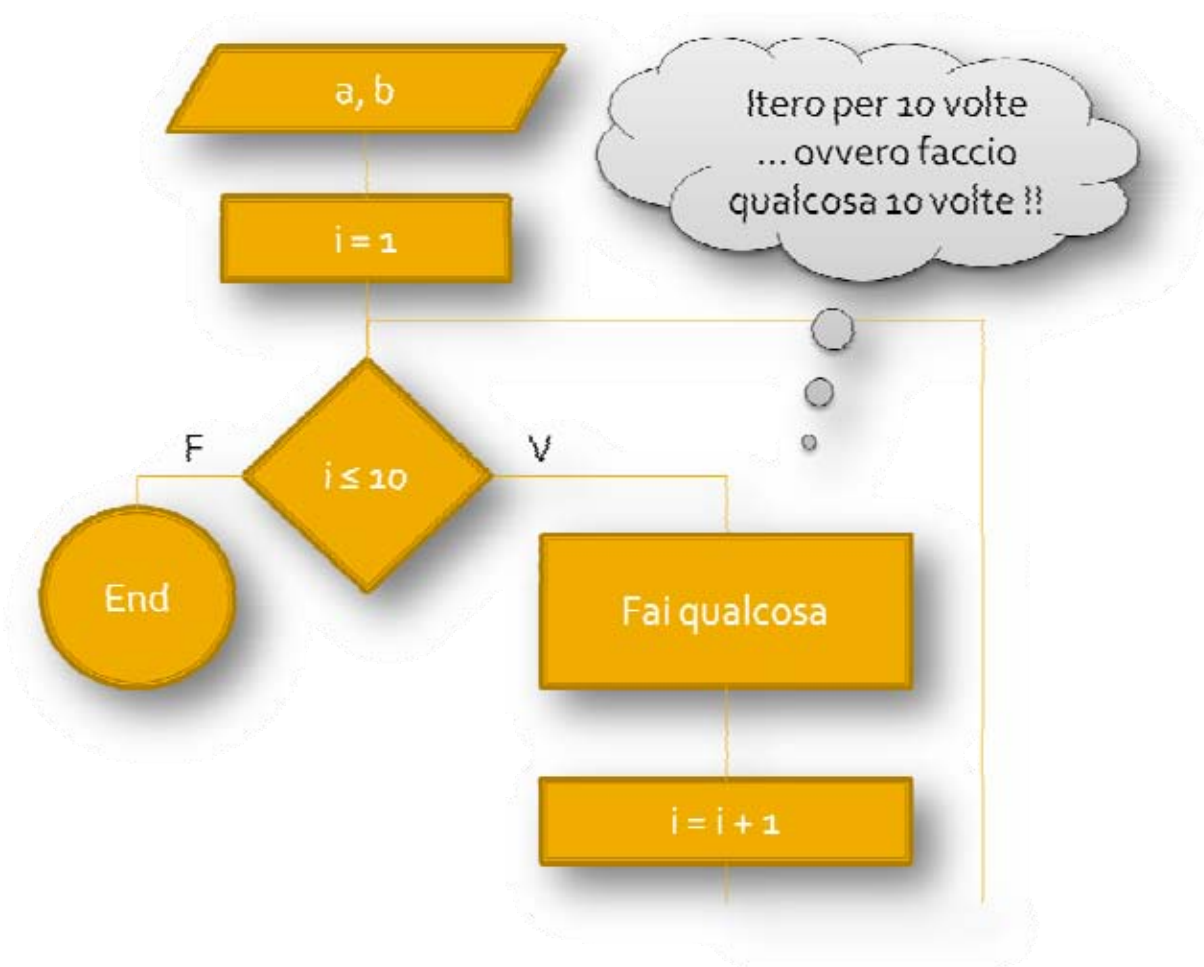
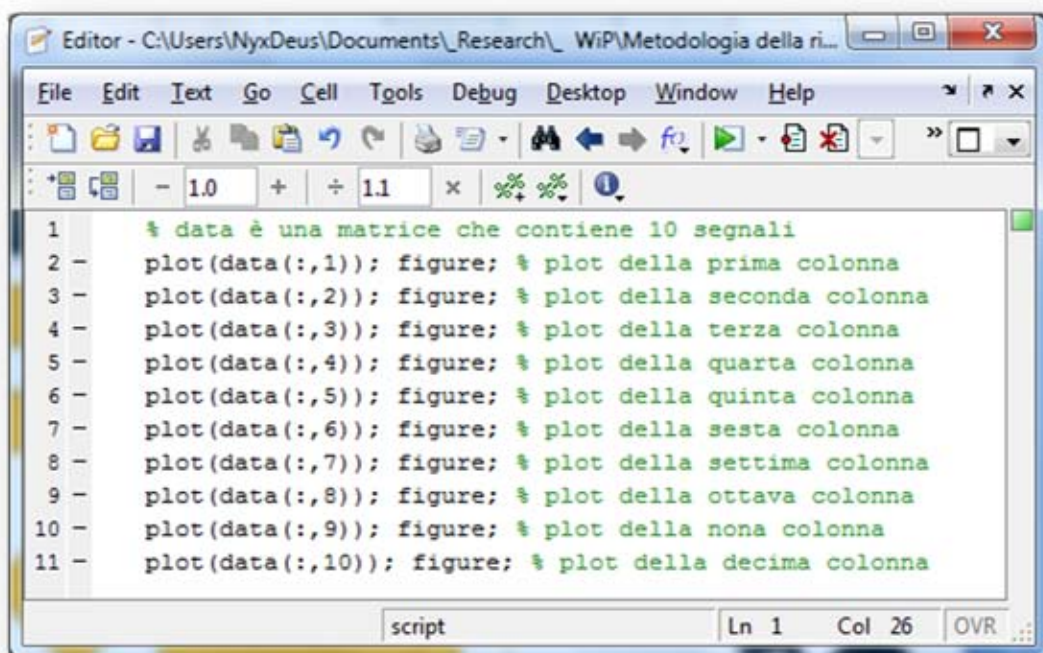


Figura 11.5. Diagramma a blocchi di una iterazione.

Questo significa che possiamo inserire all'interno del ciclo tutte le espressioni di cui necessitiamo. Il ciclo è appunto la parte che viene iterata (ripetuta) diverse volte. Di base, l'iterazione è una condizione con una variabile indice incrementale e un ritorno, tuttavia tutti i moderni linguaggi di programmazione hanno al loro interno un'istruzione dedicata al ciclo, che ci semplifica la creazione e l'uso di questa condizione.

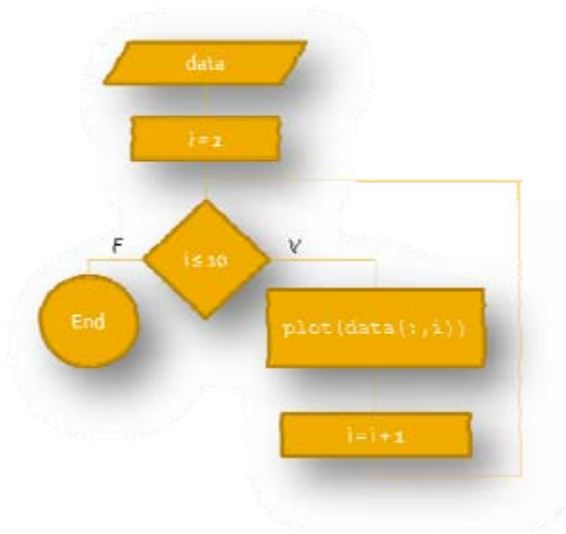
Cerchiamo comunque di capire meglio quale possa essere il senso di usare un'iterazione con i nostri dati. Ipotizziamo, in particolare, di dover effettuare il plotting di 10 segnali, uno per colonna da una matrice "data" relativa a un singolo partecipante. In Figura 11.6 riportiamo la soluzione, come da conoscenze, acquisite in precedenza, al Capitolo 10.



```
1 % data è una matrice che contiene 10 segnali
2 - plot(data(:,1)); figure; % plot della prima colonna
3 - plot(data(:,2)); figure; % plot della seconda colonna
4 - plot(data(:,3)); figure; % plot della terza colonna
5 - plot(data(:,4)); figure; % plot della quarta colonna
6 - plot(data(:,5)); figure; % plot della quinta colonna
7 - plot(data(:,6)); figure; % plot della sesta colonna
8 - plot(data(:,7)); figure; % plot della settima colonna
9 - plot(data(:,8)); figure; % plot della ottava colonna
10 - plot(data(:,9)); figure; % plot della nona colonna
11 - plot(data(:,10)); figure; % plot della decima colonna
```

Figura 11.6. Plotting di 10 segnali (colonne) da una matrice.

Effettivamente il plotting fatto eseguendo il programma in Figura 11.6 funziona perfettamente. Peraltro, come abbiamo visto al capitolo 9, è possibile effettuare una stampa unione che ci permetta di generare questo codice in modo automatico, anche se dovessimo farlo per 100 o 1000 colonne. Tuttavia dobbiamo anche ammettere che non è certamente il metodo più comodo e veloce possibile e richiederà comunque l'apertura di altri due programmi e dei copia-incolla ragionati. In Figura 11.7 presentiamo, invece, il costrutto "for" in Matlab, per il plotting di 10 segnali con iterazione.



```

1 % data è una matrice che contiene 10 segnali
2 for i=1:10 % questo vuol dire: per i che va da 1 a 10 !!
3     plot(data(:,i)); figure; % plot della i-esima colonna
4 end % chiude il ciclo "for, next"
5 % notare che dentro un ciclo for l'incremento di i è automatico
  
```

Figura 11.7. Plotting di dieci segnali (colonne) con il ciclo "for".

L'uso del ciclo "for" semplifica immensamente il codice e ci permette di effettuare l'operazione di plotting di dieci segnali con tre righe di codice. Se la matrice data contenesse cento o mille segnali, il programma sarebbe comunque uguale, con l'unica differenza sulla prima riga per indicare "for i=1:n" con n (numero delle colonne) grande quanto serve. Inoltre l'istruzione "1:n", che leggeremo come "da 1 a n" usa le stesse proprietà viste già nei capitoli precedenti per le matrici. Questo vuol dire che possiamo anche indicare un vettore di numeri a piacimento, come ad esempio:

```
for i=1:(n-1) % per i da 1 a n meno 1
for i=[1 4 6 9] % per i uguale a 1, 4, 6 e 9
for i=[1:3 8:15 33:52] % per i da 1 a 3, da 8 a 15 e da 33 a 52
```

Tutte queste espressioni per il nostro "for" saranno totalmente comprensibili per Matlab e daranno seguito a selezioni accurate, secondo le nostre esigenze, che in questo caso potrebbero per esempio essere di fare il plotting solo di alcuni segnali.

11.2 Dalle matrici alle celle

Acquisita una certa pratica e capacità d'uso avanzato delle matrici, possiamo esaminare la cella, una struttura molto più complessa, sempre con lo scopo di semplificare i dati e il nostro intervento.

Una cella è una struttura in grado di contenere differenti tipologie di dati. In particolare una cella può contenere molte matrici contenenti a loro volta molti altri dati (o segnali).

Quindi una cella è come una borsa contenente tutti i dati di cui Necessitiamo. Ad esempio si possono inserire tutte le matrici di tutti i partecipanti a una sperimentazione.

Da un punto di vista strutturale una cella è un contenitore simile a una matrice e avremo quindi righe e colonne. Ecco come creare la nostra prima cella (Figura 11.8):

`primacella=cell(8,5); % Creazione di una cella 8x5`

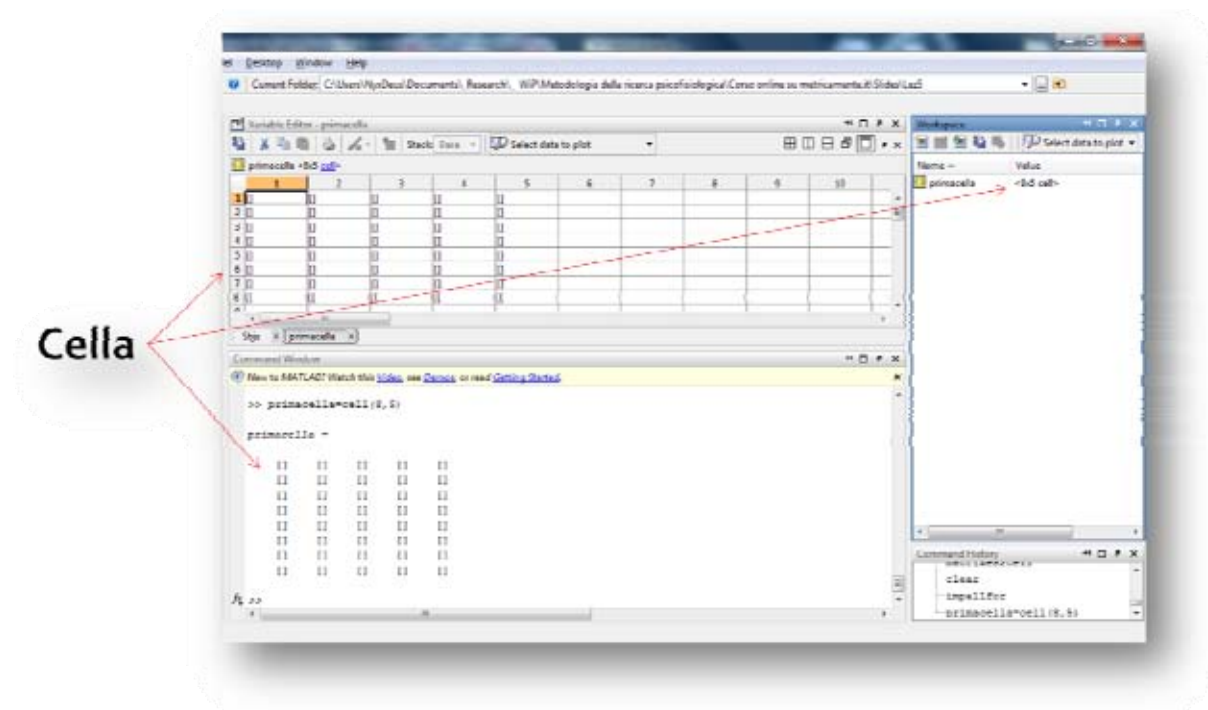



Figura 11.8. Creazione di una Cella vuota, pronta a contenere 40 elementi (ad esempio matrici). E' possibile identificare le celle nel Workplace, anche grazie alla tipica icona .

Adesso la nostra "cella" (primacella) esiste, con otto righe e cinque colonne, per un totale di 40 elementi (5x8), che possono ad esempio essere matrici contenenti milioni di dati ciascuna.

Come abbiamo visto nelle precedenti lezioni, per interrogare una matrice dobbiamo indicare riga e colonna.

Con una cella vale tutto allo stesso modo. L'unica differenza è l'uso di parentesi graffe:

```
primacella{3,4}; % identificazione della cella in posizione (3,4)
```

che indicherà la cella in corrispondenza della terza riga e della quarta colonna, nella struttura generale.

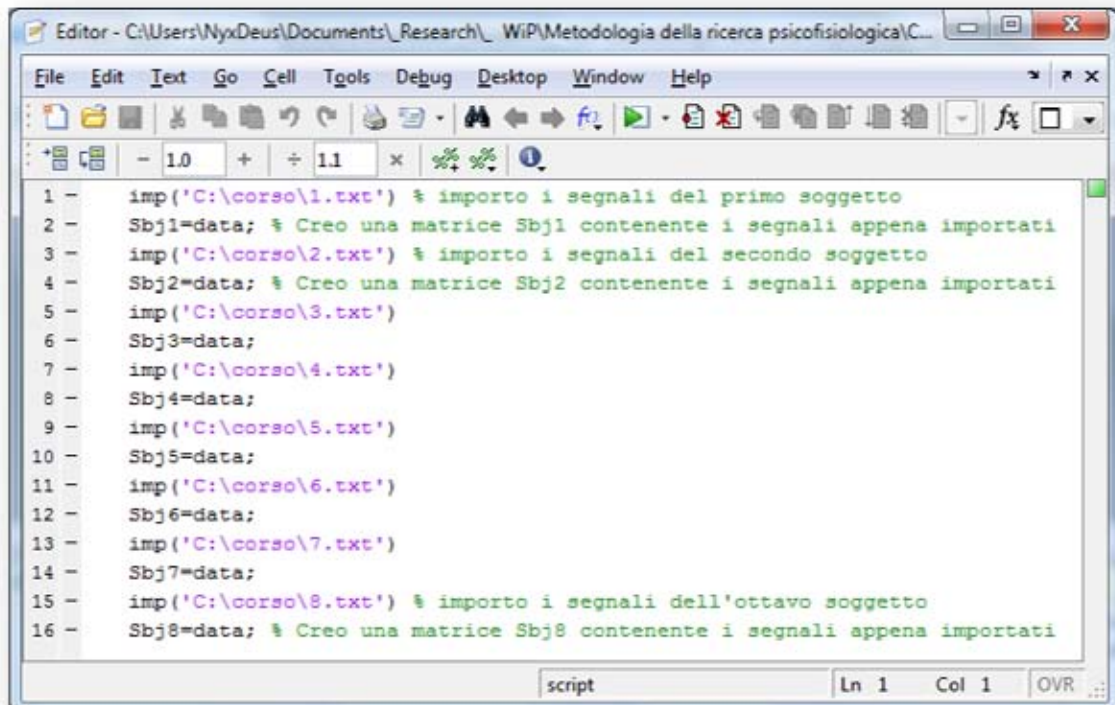
Ipotizziamo, in particolare, di avere importato il file 1.xls contenente dati psicofisiologici di un soggetto e di aver rinominato la matrice come Sbj1. Vogliamo adesso inserire questa matrice (Sbj1) nella nostra cella (primacella), in particolare la vogliamo inserire in posizione terza riga e quarta colonna.

La logica di inserimento di una matrice in una cella, funziona esattamente come l'inserimento di un numero in una matrice. Basterà dunque fare un assegnamento, specificando la posizione della cella è specificare che in quella posizione va messa la nostra matrice. Come possiamo vedere in Figura 11.9, basterà dunque dare il comando

```
"primacella{3,4}=Sbj1;" % Inserimento di una matrice nella cella
```


Abbiamo già visto il caso in cui volevamo fare il plotting di più segnali (colonne) di uno stesso soggetto. adesso ci chiediamo se sia possibile usare il ciclo "for" per gestire più di un soggetto.

Ad esempio, avevamo visto come importare 8 soggetti in Matlab, come da Figura 11.10.



```
Editor - C:\Users\NyxDeus\Documents\Research\WIP\Metodologia della ricerca psicofisiologica\C...
File Edit Text Go Cell Tools Debug Desktop Window Help
- 1.0 + ÷ 1.1 ×
1 - imp('C:\corso\1.txt') % importo i segnali del primo soggetto
2 - Sbj1=data; % Creo una matrice Sbj1 contenente i segnali appena importati
3 - imp('C:\corso\2.txt') % importo i segnali del secondo soggetto
4 - Sbj2=data; % Creo una matrice Sbj2 contenente i segnali appena importati
5 - imp('C:\corso\3.txt')
6 - Sbj3=data;
7 - imp('C:\corso\4.txt')
8 - Sbj4=data;
9 - imp('C:\corso\5.txt')
10 - Sbj5=data;
11 - imp('C:\corso\6.txt')
12 - Sbj6=data;
13 - imp('C:\corso\7.txt')
14 - Sbj7=data;
15 - imp('C:\corso\8.txt') % importo i segnali dell'ottavo soggetto
16 - Sbj8=data; % Creo una matrice Sbj8 contenente i segnali appena importati
script Ln 1 Col 1 OVR
```

Figura 11.10. Import di 10 soggetti in matrici Matlab.

Ci chiediamo dunque se sia possibile iterare il processo di importazione. Supponiamo di avere otto file ".xls" nella cartella "c:\corso\" contenenti i segnali psicofisiologici di otto soggetti e proviamo a scrivere un programma con un ciclo "for" per l'importazione automatica in Matlab.

Quello che vogliamo ottenere è una cella con tutti i soggetti (che chiameremo "Sbjs") di otto righe e una colonna. La prima riga conterrà la matrice "Sbj1" con i dati psicofisiologici del primo soggetto. La seconda riga conterrà la matrice "Sbj2" per il secondo soggetto e così via fino all'ottava riga che conterrà la matrice "Sbj8" con i dati dell'ottavo soggetto.

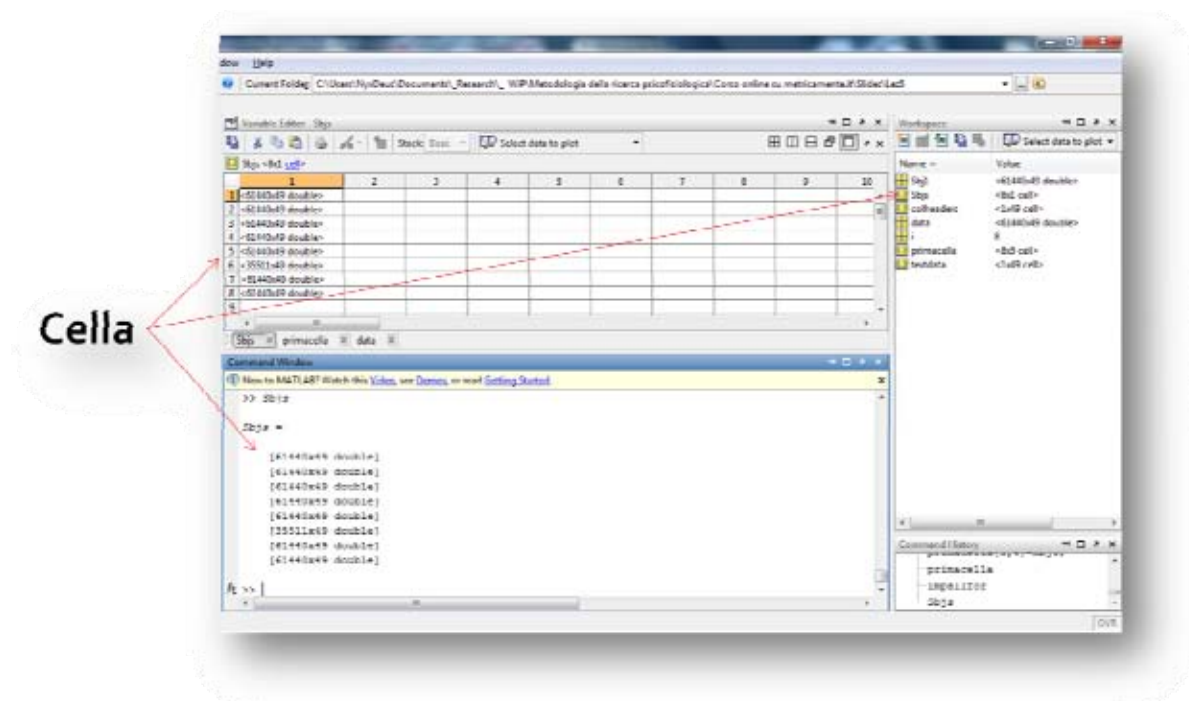


Figura 11.11. Cella contenente una matrice (soggetto) per ogni riga, per un totale di otto righe.

Come abbiamo già visto, ci sono diversi modi per ottenere questo risultato. Si potrebbe usare ad esempio la stampa unione per automatizzare otto inserimenti con la formula già vista in Figura 11.9. Tuttavia visto che questa operazione è di fatto un'iterazione,

possiamo creare un apposito codice usando il "for" per inserire tutte le celle, siano esse otto o ottomila. Per procedere con l'iterazione, vediamo prima lo pseudo codice da usare, ovvero un codice intermedio tra quello che andremo a eseguire e una sua descrizione colloquiale (Figura 11.12).

Pseudo codice

```
Creo una cella di 8 righe e una colonna  
  
for i=1:8  
  importa l'i esimo soggetto  
  inserisco la matrice data nella i-esima riga della cella  
end
```

Come implementare queste due righe in Matlab?

Figura 11.12. Pseudo codice per l'import con ciclo for.

Abbiamo già visto come implementare entrambe le operazioni selezionate con la parentesi graffa in Figura 11.12, tuttavia avremo qui qualche piccolo adattamento che vale la pena discutere in forma più estesa.

Per l'importazione di un singolo soggetto, abbiamo già visto che si può facilmente creare un programmino "imp", che importi in una matrice "data" tutti i dati contenuti in "1.xls".

Basterà poi scrivere la seguente riga di codice:

```
data = imp('C:\corso\1.xls'); % importazione del file con "imp.m"
```

Dove l'espressioni tra apici è una stringa (ovvero una sequenza di caratteri a cui non è associata necessariamente una variabile) con il percorso e il nome del file da importare.

Dunque, potremmo essere indotti a credere che se vogliamo importare l'i-esimo soggetto (con i da 1 a 8), basterà assegnare alla matrice "data" la seguente espressione:

```
imp('C:\corso\i.xls') % Modo errato per importare l'iesimo soggetto
```

Tuttavia il problema è che l'espressione tra apici è una stringa (con il percorso e il nome del file da importare) e Matlab la leggerà sempre con "C:\corso\i.xls" senza sostituire la "i" con i valori da 1 a 8, che servono per importare gli otto soggetti.



Figura 11.13. Lettura da parte del programma "imp.m" della stringa "C:\corso\i.xls".

In pratica il programma "imp.m" con l'input in Figura 11.13, andrà a cercare sul disco (sotto "C:\corso") il file "i.xls", che ovviamente non esiste.

Quindi dobbiamo trovare un modo per fare capire al programma "imp.m" che la "i" non è una stringa qualsiasi, ma deve cambiare (da 1 a 8) all'interno di un'interazione con ciclo "for".

Una prima cosa che possiamo fare è separare la stringa in tre parti, come in Figura 11.14.

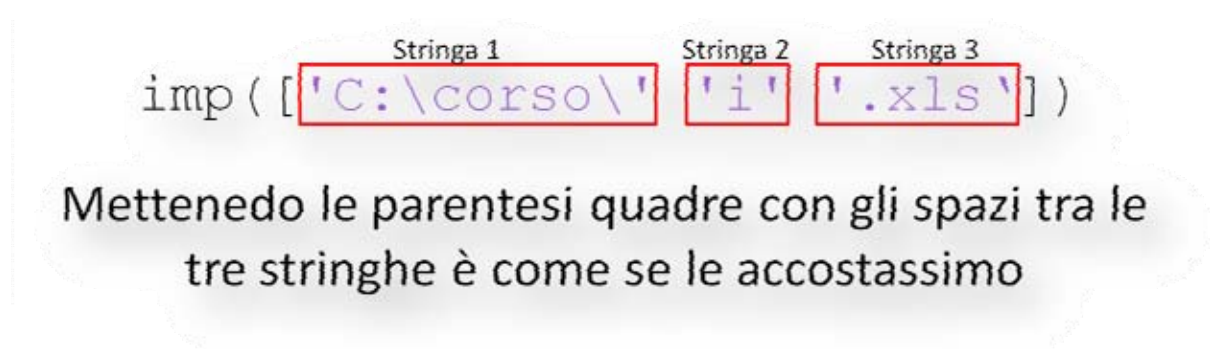


Figura 11.14. Accostamento di tre stringhe separate in una unica, dentro il programma "imp.m".

In pratica scrivere ['a' 'b' 'c'] o scrivere 'abc', per il programma "imp.m" è la stessa cosa, in quanto il programma .m eredita le caratteristiche di Matlab, inclusa l'operazione di accostamento tra elementi per mezzo di parentesi quadre.

Il secondo step è spiegare al programma "imp.m" che deve prendere in considerazione la "i" come variabile e poi trasformare il numero corrispondente all'iterazione nel ciclo (il valore che la

variabile "i" assume in quel momento) in una stringa da accostare tra le altre due.

Se mettessimo la sola "i" senza apici, il programma non la accosterebbe, in quanto non può accostare una (o due) stringhe con una variabile, quindi bisogna trasformare la variabile "i" in stringa, ma fare in modo che il programma la legga comunque come variabile (in modo che può così variare tra 1 e 8). Farlo è più semplice che spiegarlo, infatti basterà usare l'istruzione "int2str" che trasforma una variabile in una stringa di testo. L'espressione di cui sopra diventerà dunque ['C:\corso\' int2str(i) '.xls']: quando i è uguale a 1 si avrà ['C:\corso\' int2str(1) '.xls'] e quindi "C:\corso\1.xls", quando i è uguale a 2 si avrà ['C:\corso\' int2str(2) '.xls'] e quindi "C:\corso\2.xls" e così via per qualunque i da 1 a n (da 1 a 8 nel caso sopra).

La nostra assegnazione alla matrice "data" diventa dunque:

```
data = imp(['C:\corso\' int2str(i) '.xls']) % import con i dinamico
```

che inserita in un ciclo "for" consentirà al programma "imp.m" di leggere la "i" come 1, poi come 2, e così via e infine come 8 e di volta in volta la trasformerà in stringa da accostare alle altre tra le altre due stringhe.

A questo punto possiamo trasformare lo pseudo-codice visto in Figura 11.12, includendo l'import "dinamico" delle otto matrici.

Come possiamo vedere dalla Figura 11.15, il programma adesso, ad ogni step (ovvero per ogni "i") importa la matrice

corrispondente a i (la prima, poi la seconda, ...) e la assegna alla matrice "data". L'unica cosa che manca a questo punto è l'inserimento della matrice "data" in una cella.

```
Sbjs=cell(8,1); %Creo una cella di 8 righe e una colonna

for i=1:8

    %importo l'i-esimo soggetto ... ovvero:
    data = imp(['C:\corso\' int2str(i) '.xls'])
    %inserisco la matrice data nella i-esima riga della cella
    %qui dovrò inserire l'inserimento della matrice in cella
end
```

Quando $i=1$ sarà `imp(['C:\corso\' '1' '.xls'])`
 Quando $i=2$ sarà `imp(['C:\corso\' '2' '.xls'])` ... ecc. ...
 ... Quando $i=8$ sarà `imp(['C:\corso\' '8' '.xls'])`




Figura 11.15. Codice Matlab per l'importazione multipla di otto matrici di dati (1.xls, 2.xls, ..., 8.xls).

In particolare, la matrice "data" corrispondente all' i -esima importazione (e quindi all' i -esimo soggetto), andrà inserita all' i -esima riga della cella "Sbjs" che abbiamo appositamente creato con 8 righe (infatti una per soggetto) e una colonna unica.

Precedentemente abbiamo già visto come assegnare una matrice in una cella. L'operazione è qui identica, con la differenza che la colonna assegnata sarà sempre 1, mentre la riga sarà semplicemente " i ", che ovviamente inserita nel ciclo "for" verrà

iterata n volte (nel nostro caso 8 volte), assegnando di volta in volta una riga diversa, come da Figura 11.16.

Ultimo step sul codice

```
Sbjs=cell(8,1); %Creo una cella di 8 righe e una colonna

for i=1:8
    %importo l'i-esimo soggetto ... ovvero:
    imp(['C:\corso\' int2str(i) '.xls'])

    %inserisco la matrice data nella i-esima riga della cella
    Sbjs(i,1)=data;
end
```

Cella (Sbjs)

i-esima riga della Cella

Matrice appena importata

Figura 11.16. Ultimo passaggio dello pseudo-codice: assegnazione della matrice "data" alla corrispondente posizione in cella.

Ecco il risultato che salviamo nel file "impallfor.m", come sempre disponibile nei materiali supplementari al volume.

```

1 - Sbjs=cell(8,1); % Creo una cella di 8 righe e una colonna
2 - for i=1:8
3 -     imp(['C:\corso\' int2str(i) '.xls']) % importo i segnali dell'i-esimo soggetto
4 -     Sbjs{i,1}=data; % Inserisco la matrice data nella i-esima riga della cella
5 - end

```

Figura 11.17. File "impallfor", ovvero importa tutto con il "for".

Il grande vantaggio di questo file è che se volessimo usarlo per importare ottocento soggetti invece che soli otto in una cella, basterebbe sostituire 8 con 800 nelle prime due righe di codice. Ovviamente non bisogna dimenticare che per fare questo bisognerà aver rinominato tutti i file con una sequenza da "1.xls", "2.xls", ..., "800.xls": operazione abbastanza semplice se si usa il programma renamer, descritto nella prima parte del volume.

Abbiamo dunque anche scoperto, adesso, perché è meglio usare il codice (parliamo comunque di cinque righe di codice) rispetto al wizard automatico di importazione. D'altra parte se qualcuno è ancora dell'idea che sia meglio eseguire per ottocento volte (o anche solo per dieci volte) il wizard, non gli si può dare certo torto: funziona anche quello, solo che richiederà molto più tempo (probabilmente ha anche sbagliato libro da leggere, d'altra parte se ama perdere tempo ...). Bisogna inoltre anche considerare che un'importazione fatta tramite codice è molto più rapida di un'importazione fatta tramite wizard, anche per una sola matrice.

Il motivo per cui ho voluto spiegare in precedenza l'importazione tramite wizard è la sua utilità per generare in automatico il codice che abbiamo chiamato "imp.m", che viene generato specificamente sul formato di dati che dobbiamo importare. Poi, una volta creato questo piccolo programmino, possiamo procedere con l'iterazione. Una volta che ho tutti i soggetti in una cella bisogna però capire come faccio a lavorare con le singole matrici in essa contenute. Una possibilità è creare delle singole matrici per ogni soggetto, ad esempio: $Sbj1=Sbjs\{1,1\}$; $Sbj2=Sbjs\{2,1\}$; ...; $Sbj8=Sbjs\{8,1\}$. A questo punto si può lavorare con le singole matrici e non considerare più la cella creata. Tuttavia, come vedremo nel prossimo paragrafo, esistono molti vantaggi a lavorare con una struttura come la cella.

11.3 Manipolare e organizzare le celle

Lavorare con le celle può essere ampiamente vantaggioso. Immaginiamo di voler fare il plotting delle skin conductance di tutti i soggetti importati. Ipotizziamo che la skin conductance è nella sesta colonna di ogni singola matrice. Per identificare l'*i*-esima matrice nella cella precedente, dobbiamo scrivere:

```
Sbjs{i,1} % i-esima riga, quindi i-esima matrice, della cella
```

Per identificare la sesta colonna di questa matrice, basterà scrivere la seguente espressione:

```
Sbjs{i,1}(:,6) % sesta colonna della i-esima matrice
```

in pratica Matlab vede "Sbjs{i,1}" come una matrice e ne eredita la sintassi senza problemi. Quindi, così come per una matrice "A" scriviamo "A(:,6)" per identificare la sesta colonna, per una matrice "Sbjs{i,1}" scriveremo "Sbjs{i,1}(:,6)".

Per plottarla allora basterà inserire l'espressione precedente nella funzione plot():

```
plot(Sbjs{i,1}(:,6)) % plot della colonna 6, della i-esima matrice
```

A questo punto possiamo inserire questa riga all'interno di un ciclo "for" per fare il plotting della sesta colonna di tutti i soggetti, come da Figura 11.18 in cui aggiungiamo dentro il ciclo "for" la solita riga "figure;" altrimenti sovrascriverà la figura ogni volta invece di aprirne una nuova.

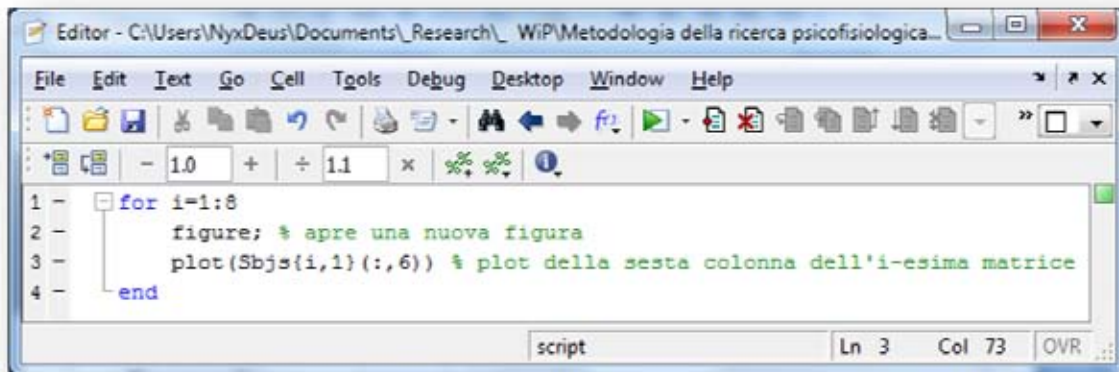


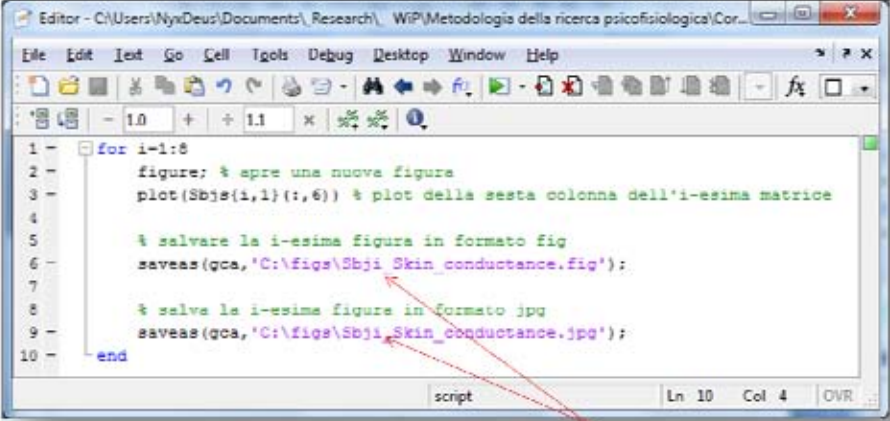
Figura 11.18. Programma per fare il plotting di tutti gli 8 soggetti.

Spesso quando si generano una serie di plot è anche utile salvarli in una cartella, soprattutto quando questi siano relativi a più soggetti. Nel Capitolo precedente abbiamo esaminato l'espressione per il salvataggio delle figure generate sia in formato ".fig" (standard Matlab), sia in formato ".jpg".

Le istruzioni per salvare nei due formati sono:

```
saveas(gca,'C:\figs\nome.fig'); % Salva la figura corrente in ".fig"  
saveas(gca,'C:\figs\nome.jpg'); % Salva la figura corrente in ".jpg"
```

Possiamo dunque aggiungere le espressioni del salvataggio nel precedente ciclo "for", tuttavia dobbiamo stare attenti a non inserire la nostra "i" direttamente nella stringa di testo, come in Figura 11.19, altrimenti avremo nuovamente il problema di identificazione delle variabili all'interno di una stringa in Matlab, di cui abbiamo prima discusso.



```

1 - for i=1:8
2 -     figure; % apre una nuova figura
3 -     plot(Sbj{i,1}(:,6)) % plot della sesta colonna dell'i-esima matrice
4 -
5 -     % salvare la i-esima figura in formato fig
6 -     saveas(gca, 'C:\figs\Sbj{i}_Skin_conductance.fig');
7 -
8 -     % salve la i-esima figure in formato jpg
9 -     saveas(gca, 'C:\figs\Sbj{i}_Skin_conductance.jpg');
10 - end

```

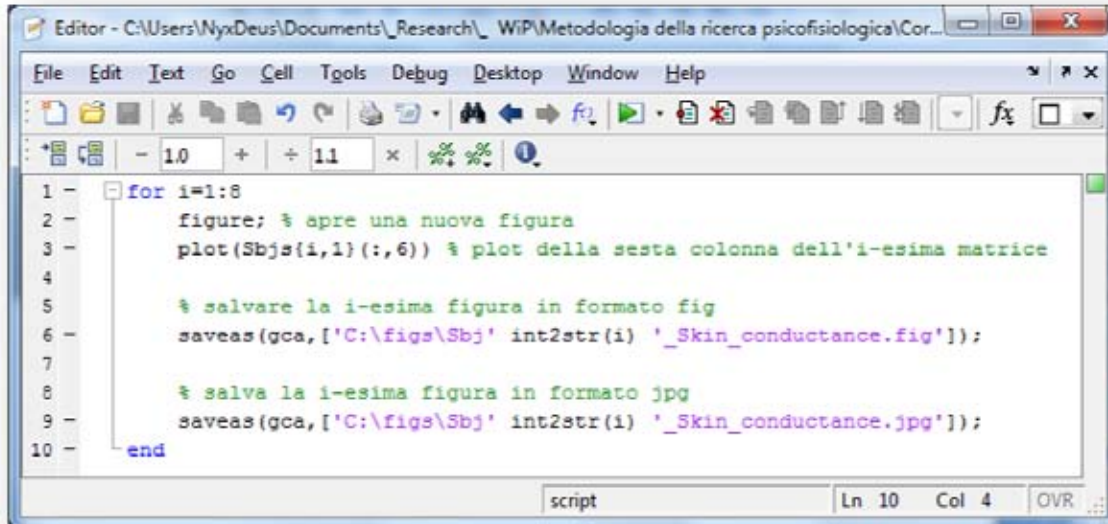
Il problema è che anche qua la "i" non può essere inserita in questo modo

Figura 11.19. Inserimento errato del salvataggio immagini.

Per risolvere questo problema cominciamo, come abbiamo visto prima, a spezzare in tre stringhe usando virgolette e parentesi quadre come segue

```
saveas(gca,['C:\figs\Sbj' 'i' '_Skin_conductance.fig']);
```

per poi sostituire la nostra "i" tra apici con "intstr(i)", come possiamo vedere in Figura 11.20.



```
1 - for i=1:8
2 -     figure; % apre una nuova figura
3 -     plot(Sbjs(i,1)(:,6)) % plot della sesta colonna dell'i-esima matrice
4 -
5 -     % salvare la i-esima figura in formato fig
6 -     saveas(gca,['C:\figs\Sbj' int2str(i) '_Skin_conductance.fig']);
7 -
8 -     % salva la i-esima figura in formato jpg
9 -     saveas(gca,['C:\figs\Sbj' int2str(i) '_Skin_conductance.jpg']);
10 - end
```

Figura 11.20. Programma corretto per il plotting con salvataggio coerente dentro un ciclo "for".

Come risultato di questo programma otterremo una serie ordinata di figure, sia in formato ".fig", sia in formato ".jpg" contenenti i plot della skin conductance relativi a ciascun soggetto (da 1 a 8) e con dei nomi di file coerenti rispetto al numero del soggetto. Dunque il file "Sbj5_Skin_conductance.fig" contiene il plot della skin conductance del soggetto 5. In Figura 11.21 possiamo vedere la sequenza ordinata per gli 8 soggetti.

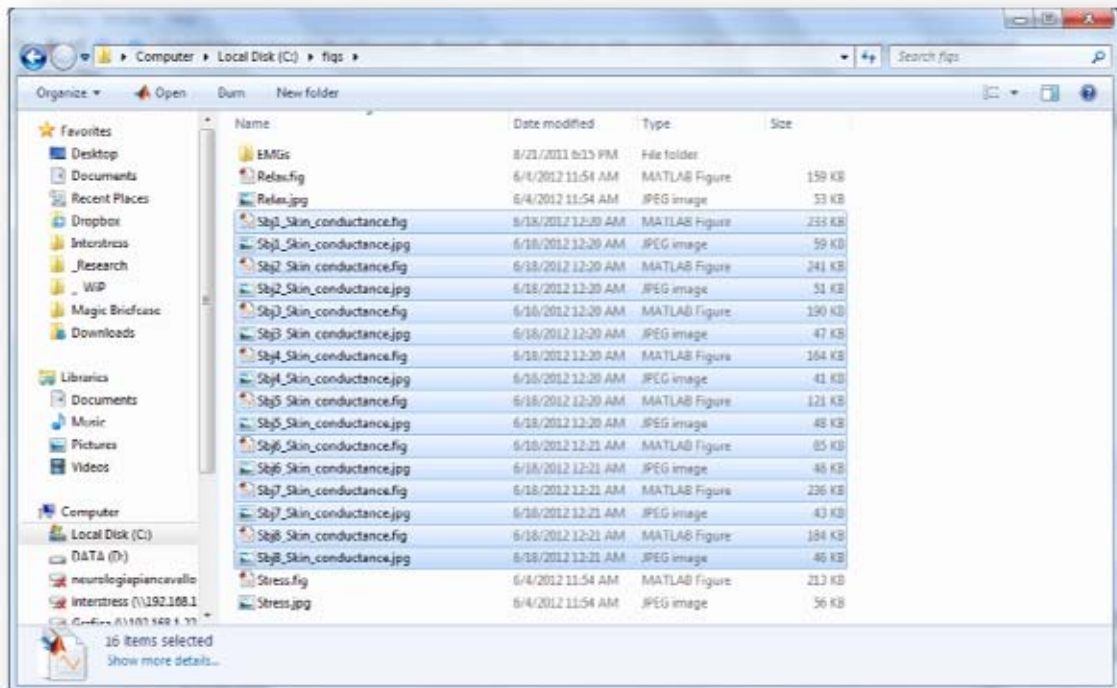


Figura 11.21. File creati dal programma per il plotting e salvataggio dei dati di skin conductance degli 8 soggetti.

Il grande vantaggio di questo programma è che se volessimo usarlo per plottare la skin conductance di ottocento soggetti contenuti in una cella invece che di soli otto, basterebbe sostituire 8 con 800 alla prima riga del programma. Inoltre, se invece di fare il plotting della skin conductance fossimo interessati a fare il plotting di un altro segnale, contenuto in un'altra colonna, basterebbe cambiare il numero di colonna indicato alla terza riga del codice.

Naturalmente dentro il ciclo "for" possiamo inserire (adattandolo) tutto ciò che riteniamo necessario, come le opzioni di tempo, il

titolo, gli assi e la legenda, viste al capitolo precedente (Figura 11.22).

```

1 for i=1:0
2     figure: % apre una nuova figura
3
4     s=256; % Campionamento (righe per secondo)
5     t=1/s:1/s:length(Sb{i,1})/s; % calcolo del vettore tempo
6
7     plot(t,Sb{i,1}(:,6)); % grafico della
8     title(['Skin Conductance del soggetto ' int2str(i)]); % titolo della figura
9     xlabel('tempo in secondi'); % descrizione asse orizzontale (x)
10    ylabel('SC'); % descrizione asse verticale (y)
11    legend('Skin Conductance'); % legenda del grafico nella figura
12
13    % salvare la i-esima figura in formato fig
14    saveas(gca,['C:\figs\Sbj' int2str(i) '_Skin_conductance.fig']);
15
16    % salva la i-esima figura in formato jpg
17    saveas(gca,['C:\figs\Sbj' int2str(i) '_Skin_conductance.jpg']);
18 end

```

Figura 11.22. Adattamento del programma con l'aggiunta del tempo e di altre informazioni di labeling.

L'inserimento del tempo e di tutte le altre informazioni non è solo una questione estetica, bensì una caratteristica funzionale importante che ci serve per un'efficace ispezione visiva dei segnali e per potere intervenire efficientemente nel caso in cui dal grafico ci rendiamo conto che ci siano problemi di interferenza o altri artefatti del segnale. Inoltre le informazioni sono personalizzabili con il numero del soggetto, trasformando la variabile "i" dentro

una stringa, quando serve, come ad esempio nel titolo (Skin Conductance del soggetto 8) (Figura 11.23).

A titolo di esempio
riportiamo il grafico
del soggetto 8

Ma il programma
salva i grafici di
tutti i soggetti

... che siano 8 o 800!

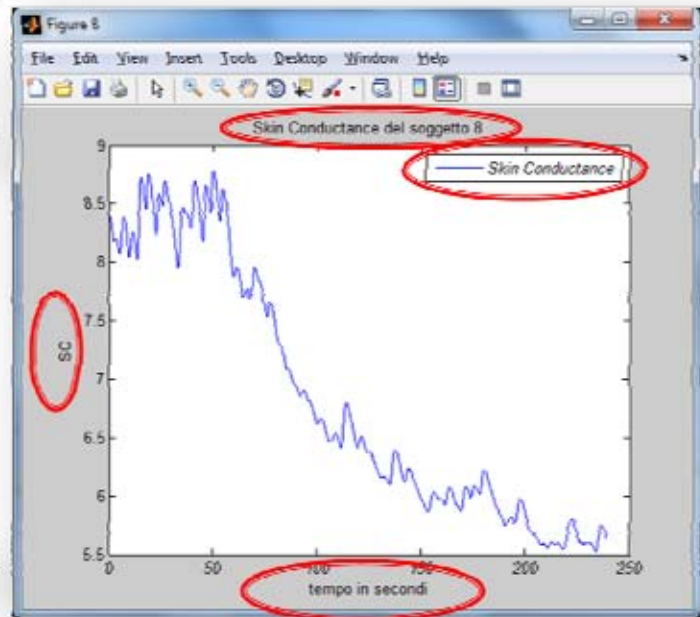


Figura 11.23. Grafico personalizzato e arricchito con varie utili informazioni, per il soggetto 8.

Usare l'iterazione con il ciclo "for" può essere molto utile anche per generare le epoche per tutti i soggetti con un'unica operazione. Nel secondo paragrafo del capitolo 8 avevamo realizzato il programma "genera_epoche.m" (Figura 11.24).

```

1 - s=256; % sampling rate
2 - Baseline=50; Relax=100; Stress=150; % tempi di partenza in secondi
3 - End=200; % fine sperimentazione
4 - Baseline_row = 1; % riga di partenza della baseline
5 - Relax_row = Relax * s ; % riga di partenza del Relax
6 - Stress_row = Stress * s ; % riga di partenza dello Stress
7 - End_row = End * s ; % riga di fine della sperimentazione
8 - data_B = data_ok (Baseline_row:Relax_row, :); % creazione matrice Baseline (data_B)
9 - data_R = data_ok (Relax_row:Stress_row-1, :); % creazione matrice Relax (data_R)
10 - data_S = data_ok (Stress_row:End_row-1, :); % creazione matrice Stress (data_R)

```

Figura 11.24. Programma "genera_epoche.m" per la generazione di tre matrici ciascuna contenente l'epoca sperimentale associata (Baseline, Relax o Stress), cambiate su altri tempi.

Ipotizziamo di voler mettere le tre epoche (Baseline, Relax e Stress) in una cella "Sbjs" con la seguente disposizione:

- Colonna 1: dati completi
- Colonna 2: Baseline
- Colonna 3: Relax
- Colonna 4: Stress

Iniziamo cambiando le precedenti matrici "data_B", "data_R" e "data_S" con la seconda, terza e quarta colonna della cella (la prima la ipotizziamo già assegnata al valore completo tramite import), ovviamente ragionando in termini di iterazione con la colonna che ogni volta ha in riga una matrice diversa. Poi

cambiamo "data_ok" (la matrice da dividere) con "Sbjs{i,1}" (la matrice i-esima da tagliare, seguita dai tagli secondo i quali tagliare. Per la Baseline in seconda colonna (baseline) i tempi saranno da "Baseline_row" a "Relax_row", per la terza colonna (relax) i tempi saranno da "Relax_row" a "Stress_row" e per la quarta colonna (stress) i tempi saranno da "Stress_row" a "End_row". Per una maggiore specificità si guardi la Figura 11.25.

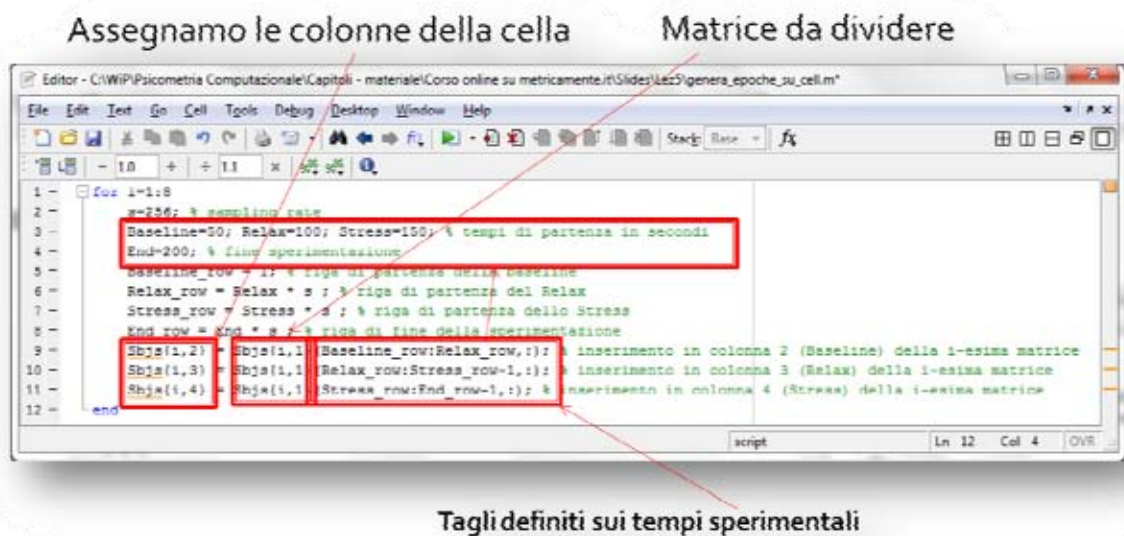


Figura 11.25. Programma "genera_epoche_su_cell" per assegnare le matrici associate alle epoche sperimentali in una cella "Sbjs".

La cella così generata conterrà 4 colonne (la prima con tutto il segnale e le altre tre con le epoche sperimentali) e 8 righe (una per ogni soggetto). Nel Workplace avremo un'unica variabile (assai complessa) "Sbjs" per identificare la cella, che sarà visibile nel dettaglio sul Variable editor (Figura 11.26).

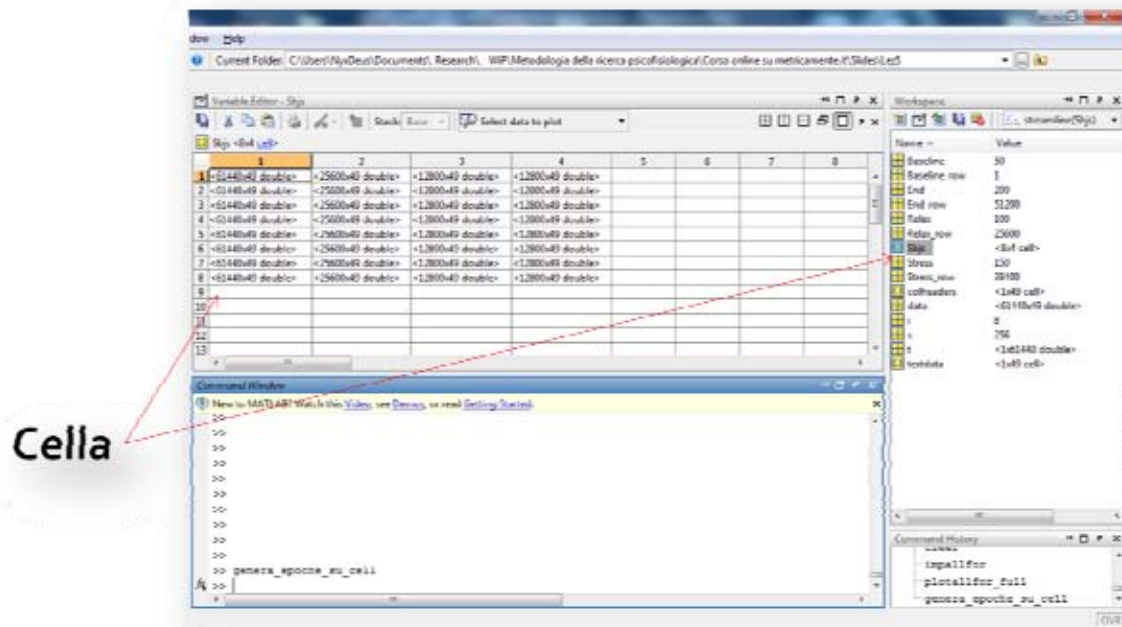


Figura 11.26. Cella generata su Workplace e Variable Editor.

A partire dalla cella "Sbjs" di dimensioni 8x4, che abbiamo appena generato, si possono effettuare diverse operazioni sulle matrici al suo interno che altrimenti sarebbero complesse o quanto meno andrebbero fatte manualmente. Quindi la cella è una struttura che ci permette operazioni multiple su strutture complesse. In particolare tramite le celle si può effettuare un'elaborazione per tutte le righe o per tutte le colonne, usando un'interazione tramite il ciclo "for".

Se ad esempio si vogliono graficare le SC della baseline basterà effettuare piccoli cambiamenti sul codice precedente realizzato per tutto il segnale (in Figura 11.22) per adattarlo alle specifiche esigenze, come da Figura 11.27. In particolare basterà indicare la

seconda colonna invece della prima e le descrizioni nel titolo, nella legenda e soprattutto nel nome del file per il salvataggio dell'immagine.

```

1 - for i=1:8
2 -     figure; % apre una nuova figura
3 -     title('Seconda colonna, invece della prima'); % titolo della figura
4 -     s=256; % Campionamento (sample per secondo)
5 -     t=1/s:1/s:length(Sbjs(:,2))/s; % calcolo del vettore tempo
6 -
7 -     plot(t,Sbjs(:,2) (:,6)); % grafico della SC del Relax
8 -     title(['Skin conductance del soggetto ' int2str(i) ' in baseline']); % titolo della figura
9 -     xlabel('tempo in secondi'); % descrizione asse orizzontale (x)
10 -    ylabel('SC'); % descrizione asse verticale (y)
11 -    legend('Baseline'); % legenda del grafico nella figura
12 -
13 -    % salvare la i-esima figura in formato fig
14 -    saveas(gcf,['C:\figs\Sbj' int2str(i) '_Skin_conductance_Baseline.fig']);
15 -
16 -    % salva la i-esima figura in formato jpg
17 -    saveas(gcf,['C:\figs\Sbj' int2str(i) '_Skin_conductance_Baseline.jpg']);
18 - end

```

Figura 11.27. Adattamento del codice per fare il plotting di una specifica epoca (baseline)

Chiaramente questi adattamenti richiedono grande attenzione e qualche prova, tuttavia bisogna tenere presente che per fare queste operazioni ci vogliono normalmente pochi secondi, infatti una volta che i dati sono organizzati in celle, sono ottimizzati per l'accesso diretto e immediato da Matlab. A volte però le celle sono talmente grandi e complesse che le operazioni richiedono più computer in cluster o una programmazione parallela, ma questo

rientra nella psicomatria computazionale più avanzata. Per le celle con decine di matrici contenenti qualche milione di dati, non abbiamo questo problema.

In Figura 11.28 vediamo la cartella di output del programma, dove sono state generate anche le Figure relative all'epoca di baseline, oltre quelle prima calcolate per tutta la sessione.

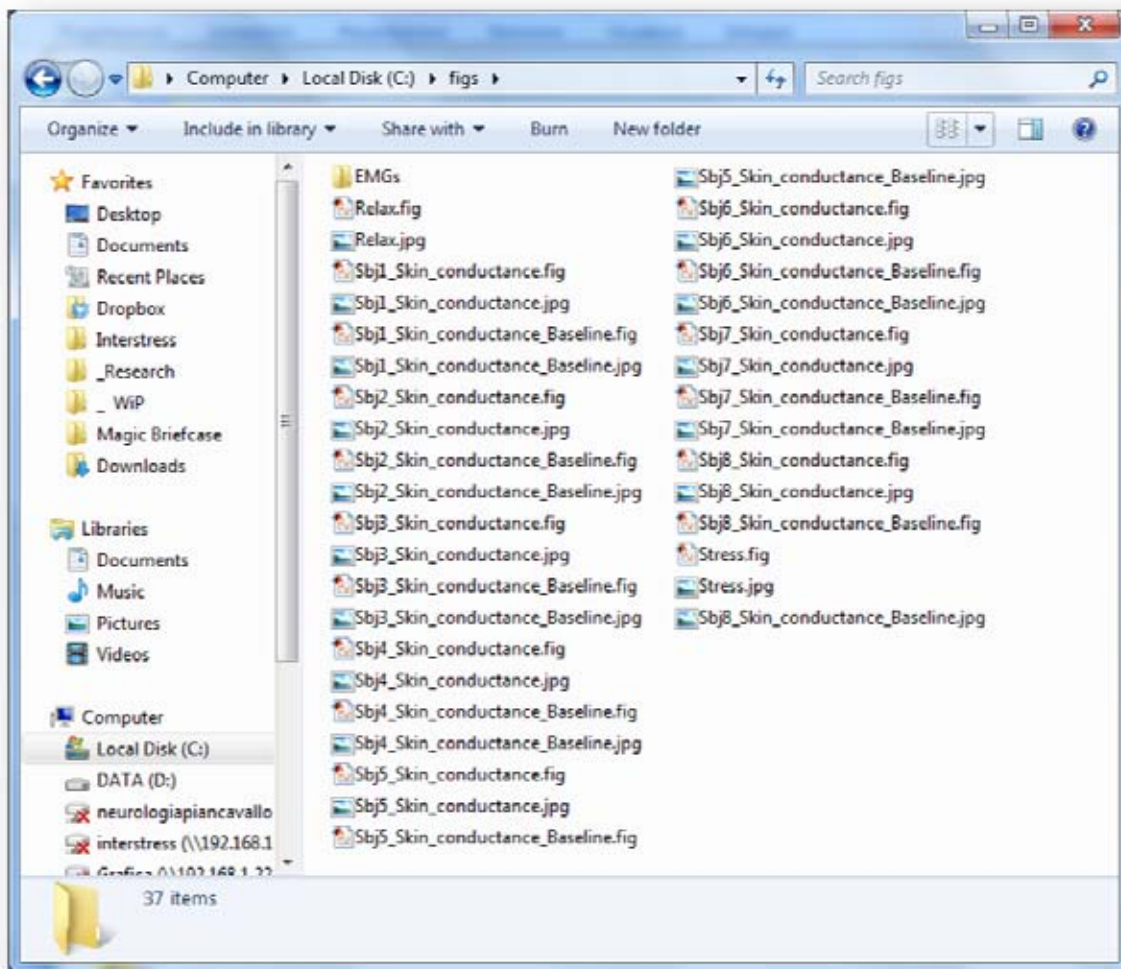
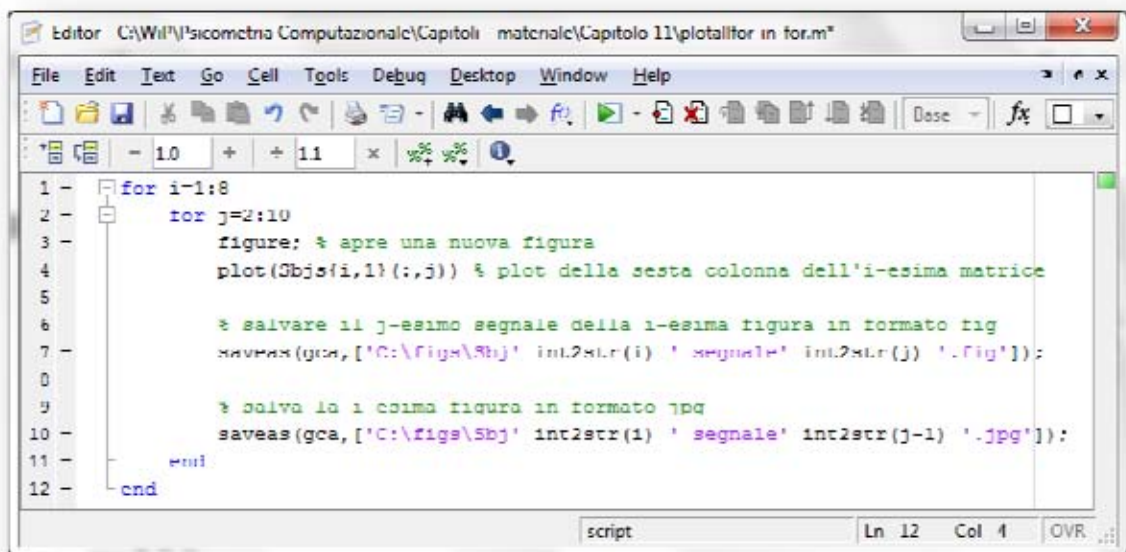


Figura 11.28. File creati dal programma per il plotting e il salvataggio dei dati di skin conductance degli 8 soggetti, sia quelli per tutta la sessione, sia quelli relativi alla sola epoca di baseline.

Per concludere vediamo anche un caso un po' più avanzato di iterazione. Immaginiamo di voler fare il plotting e salvare sul disco non solo tutti i segnali di skin conductance (colonna 6) di tutti i soggetti, ma anche tutti gli altri segnali (dalla seconda colonna alla decima), dove la prima colonna è il tempo, la seconda colonna è il segnale 1, la terza è il segnale 2, ..., la decima colonna è il segnale 9. Per fare questa operazione, dovremmo ricorrere a un'altro ciclo "for" che ci consideri anche le colonne delle matrici all'interno delle celle. Vediamo come realizzarlo in Figura 11.29.



```

1 - for i=1:8
2 -     for j=2:10
3 -         figure; % apre una nuova figura
4 -         plot(Sbjs{i,1}(:,j)) % plot della sesta colonna dell'i-esima matrice
5 -
6 -         % salvare il j-esimo segnale della i-esima figura in formato fig
7 -         saveas(gcf, ['C:\figs\5bj' int2str(i) ' segnale' int2str(j) '.fig']):
8 -
9 -         % salva la 1 esima figura in formato jpg
10 -        saveas(gcf, ['C:\figs\5bj' int2str(i) ' segnale' int2str(j-1) '.jpg']):
11 -
12 -     end

```

Figura 11.29. Inclusione di un ciclo iterativo dentro un altro.

In pratica, all'inizio la "i" è 1 e il programma farà scorrere la "j" da 2 a 10 plottando e salvando tutti i segnali da 1 a 9 (ovvero dalla colonna 2 alla 10) per il soggetto 1. Poi il programma esce dal

primo "for" (ovvero la "j" arriva a 10) e passa alla "i" successiva che è 2, plottando e salvando tutti i segnali da 1 a 9 per il soggetto 2. Il programma continua così fino ad arrivare al soggetto 8 ("i" uguale ad 8), plottando e salvando tutti i segnali anche di questo ultimo soggetto.

Considerate che si possono mettere anche altri cicli all'interno o all'esterno, qualora questo abbia un senso logico. Si potrebbe mettere ad esempio un ciclo "for h=1:4" prima del ciclo "for i=1:8" per plottare e salvare tutti i segnali di tutti i soggetti per tutte le epoche, dove epoca1 è tutto il segnale, epoca2 è la baseline, epoca3 è il relax ed epoca4 è lo stress, così come le abbiamo definite in precedenza. Includere anche questo ciclo "for" non è troppo complesso e lo lascio come esercizio. Scrivetemi.

Organizzazione strutturale dei dati complessi

Nei precedenti capitoli abbiamo visto come gestire dati complessi usando Matlab, Octave o altri software per l'implementazione di strutture logiche atte a intervenire efficacemente sulla loro manipolazione informatica. Adesso però tratteremo, in questo capitolo, un altro tipo di complessità, ovvero quella riferita alla complessità strutturale entro cui i dati vengono gestiti. Facciamo dunque riferimento al "contenitore" dei dati in senso più ampio di quanto visto prima (cella, matrice, variabile, ...), riferendoci anche alla struttura fisica che si occupa di archiviare i dati. Infatti in un momento in cui si sente tanto parlare di "big data" è d'obbligo trattare in un volume come questo di infrastrutture di raccolta dei dati che vadano oltre il personal computer dello psicologo. Ci interessiamo innanzitutto ai database relazionali, ovvero complesse strutture che includono sia varie tabelle di dati, sia l'indicazione

delle relazioni esistenti tra queste tabelle. Non ci riferiamo dunque ai semplici dataset (singole tabelle casi per variabili) che vengono invece usati per le analisi statistiche. Accenneremo poi anche alla gestione degli utenti e allo sviluppo di contenuti e interfacce dei dati.

12.1 Database relazionali

Non c'è abbastanza spazio in questo volume per trattare, seppur sommariamente dei database relazionali: già per farne la storia non basterebbe un capitolo. Dunque consiglio vivamente, al lettore interessato, di approfondire l'argomento con uno dei molti volumi aggiornati attualmente disponibili. In questo paragrafo ci limiteremo perlopiù a esaminare alcuni aspetti pratici legati all'accesso ai dati e alle applicazioni possibili. Darò anche qualche indicazione per la gestione di un database MySQL per applicazioni web usate in psicomatria computazionale.

Come abbiamo prima detto, un database relazionale è costituito da più tabelle e dalle relazioni tra queste tabelle. In un database aziendale, ad esempio, avremo sicuramente come tabella i dipendenti, con tutte le caratteristiche atte a identificarli: Società (potrebbero infatti anche essere consulenti esterni), Cognome, Nome, Indirizzo postale, Posizione, Telefono ufficio e così via.

Se definiamo anche una tabella progetti (in cui sono coinvolti i dipendenti), possiamo anche definire la relazione tra questa tabella

e la precedente. Creando altre tabelle possiamo contemporaneamente definire di volta in volta le relazioni intercorrenti, strutturando in tal modo un database relazionale, come quello d'esempio mostrato in Figura 12.1.

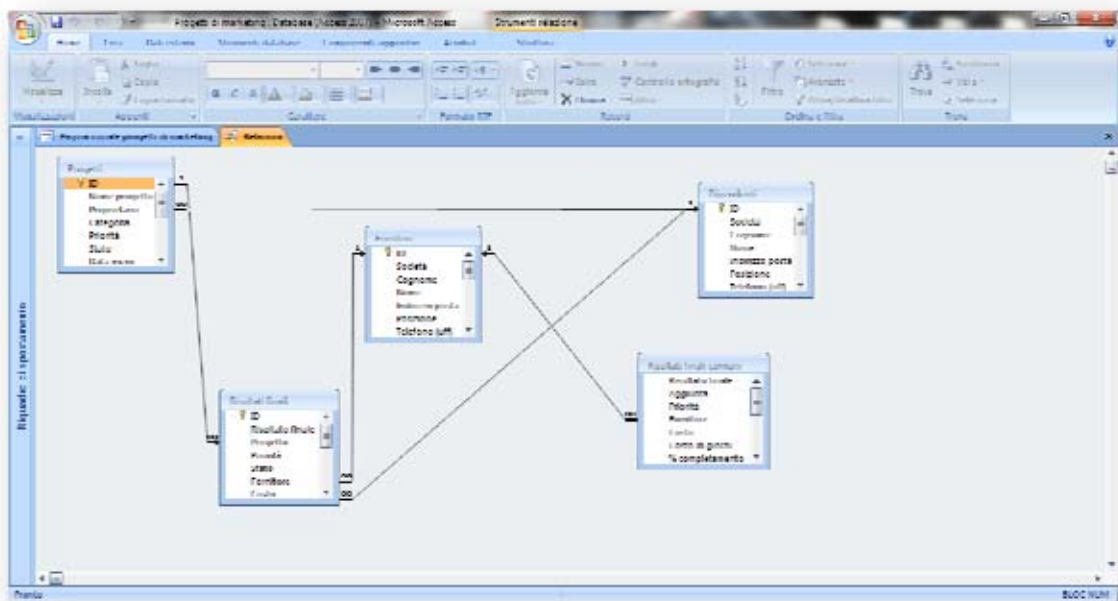


Figura 12.1. Le relazioni tra tabelle in un database di esempio in Microsoft Access

Per chi volesse far pratica, strutturando dei semplici database sul proprio computer, in Microsoft Office è incluso Microsoft Access il cui scopo è proprio quello di crearsi dei database a uso casalingo o aziendale. E' possibile, ad esempio, creare un database dei film collezionati e generare una scheda di dettaglio per ogni film. Per fare pratica in Microsoft Access e quindi con un primo esempio di

database, è possibile usare (sin già dalla versione 2007) il database aziendale di esempio Northwind 2007, la cui struttura delle relazioni è mostrata in Figura 12.2.

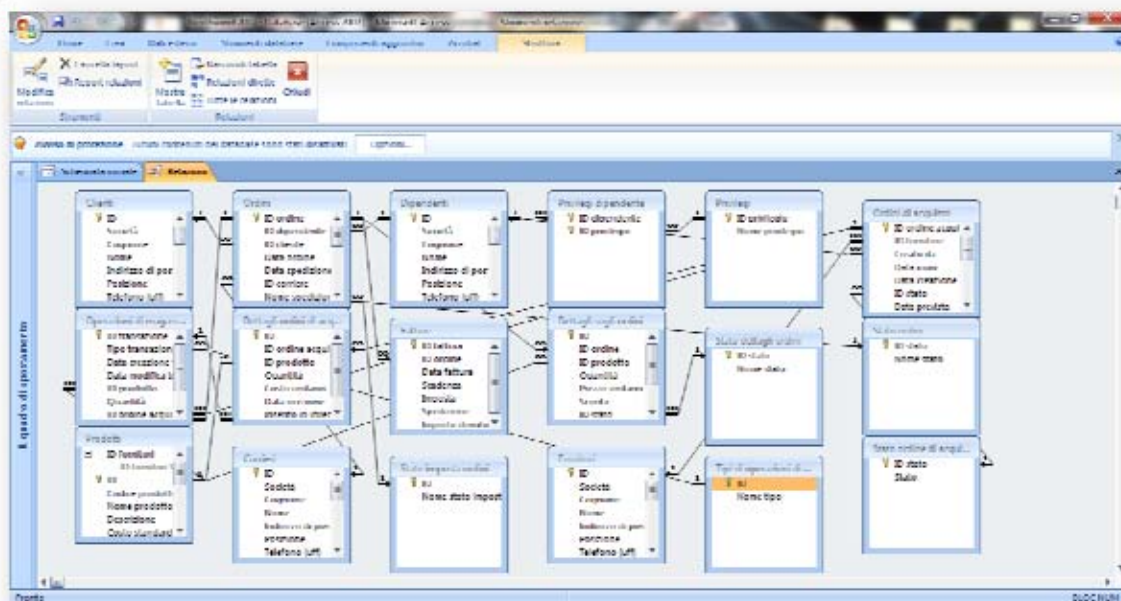


Figura 12.2. Relazioni del database di esempio Northwind 2007 in Microsoft Access.

Fare pratica con un database relazionale locale è sicuramente un ottimo esercizio e serve soprattutto per acquisire la logica di funzionamento di queste strutture, che non sono affatto semplici.

Vedremo adesso, nella filosofia open di questo volume, MySQL (<http://www.mysql.it/>), Il database open source più diffuso al mondo. In particolare MySQL rientra tra i Relational database management system (RDBMS).

Il vantaggio di MySQL per il mondo della psicologia è considerevole e merita un minimo dettaglio. Innanzitutto è un sistema assai diffuso e può essere configurato senza molte difficoltà, ma soprattutto vorrei enfatizzare l'utilità di questo sistema. In pratica quando decidiamo di costruire una piattaforma web di qualche tipo e dobbiamo dunque gestire una certa quantità di dati e di utenti, è indispensabile avere un database di questo tipo. Fin quando si realizza un sito "vetrina" per una questione di visibilità, è necessario solo uno spazio web e un dominio. Tuttavia non appena si vuole gestire una complessità di contenuti e/o utenti, diventa necessario attivare un database. Ad esempio, se si decide di fare una piattaforma web per la somministrazione di questionari, magari usando LimeSurvey (visto nel primo capitolo di questo volume) è indispensabile avere un database MySQL. Come già accennato proprio per questa piattaforma di raccolta dati, i costi legati al dominio, allo spazio web e al database MySQL (di circa 100 Megabyte), sono nell'ordine delle cinquanta euro. Tuttavia per fare un po' di pratica prima di passare a una piattaforma disponibile sul web, possiamo installare tutto quel che ci serve sul nostro computer in locale e fare tutta la pratica che vogliamo, ovviamente a costo zero.

XAMPP (<http://www.apachefriends.org/>), cito letteralmente dal sito, è una distribuzione Apache completamente gratuita e facile da installare che contiene MySQL, PHP, e Perl. XAMPP è il più popolare ambiente di sviluppo PHP, il linguaggio di programmazione in cui sono implementate la maggior parte delle

piattaforme web open source (tra cui anche LimeSurvey). L'installazione e la gestione del programma è assai semplice. Basterà avviare (con Start) i moduli necessari, per averli disponibili in locale (con indirizzo "localhost").

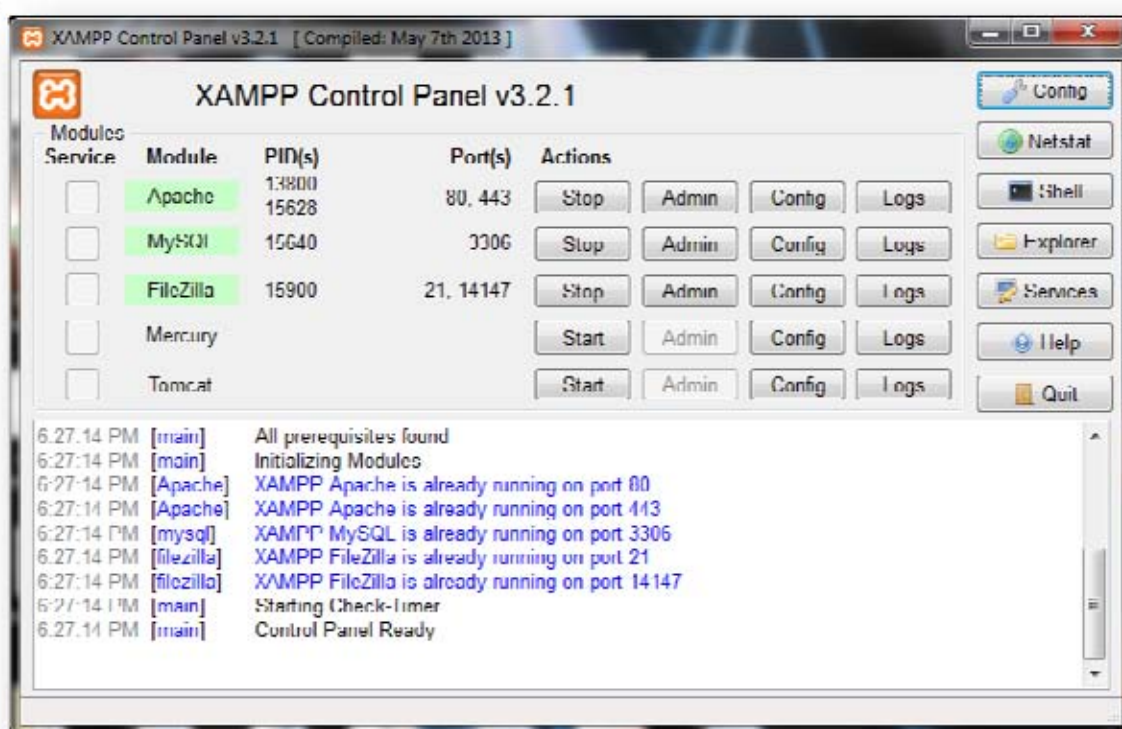


Figura 12.3. Interfaccia del pannello di controllo di XAMPP per l'avvio e la configurazione dei moduli necessari.

A questo punto, per installare una piattaforma php usando XAMPP basterà creare una cartella dentro "C:\xampp\htdocs". Ad esempio, per installare una piattaforma LimeSurvey in locale, basterà creare la cartella "C:\xampp\htdocs\Lime" e inserire all'interno di questa

cartella il contenuto del file zip scaricato dal sito istituzionale (<http://www.limesurvey.org/en/stable-release>), in pratica il codice della piattaforma.

A questo punto basterà scrivere in un browser il percorso locale prima stabilito (<http://localhost/lime>) per avviare la configurazione della piattaforma, come da Figura 12.4.



Figura 12.4. Avvio della configurazione della piattaforma LimeSurvey in locale, usando XAMPP.

A questo punto la procedura di configurazione non presenta troppi problemi, basterà infatti attenersi alle istruzioni e seguire il processo di configurazione passo dopo passo.

A un certo punto viene anche fatta una verifica della piattaforma e dell'adeguatezza del sistema rispetto alle caratteristiche obbligatorie, a quelle raccomandate, per un uso ottimale, nonché rispetto alle caratteristiche correnti. E' indispensabile, ai fini del funzionamento della piattaforma, che almeno le caratteristiche

obbligatorie siano verificate pienamente, come è possibile vedere dalla Figura 12.5.

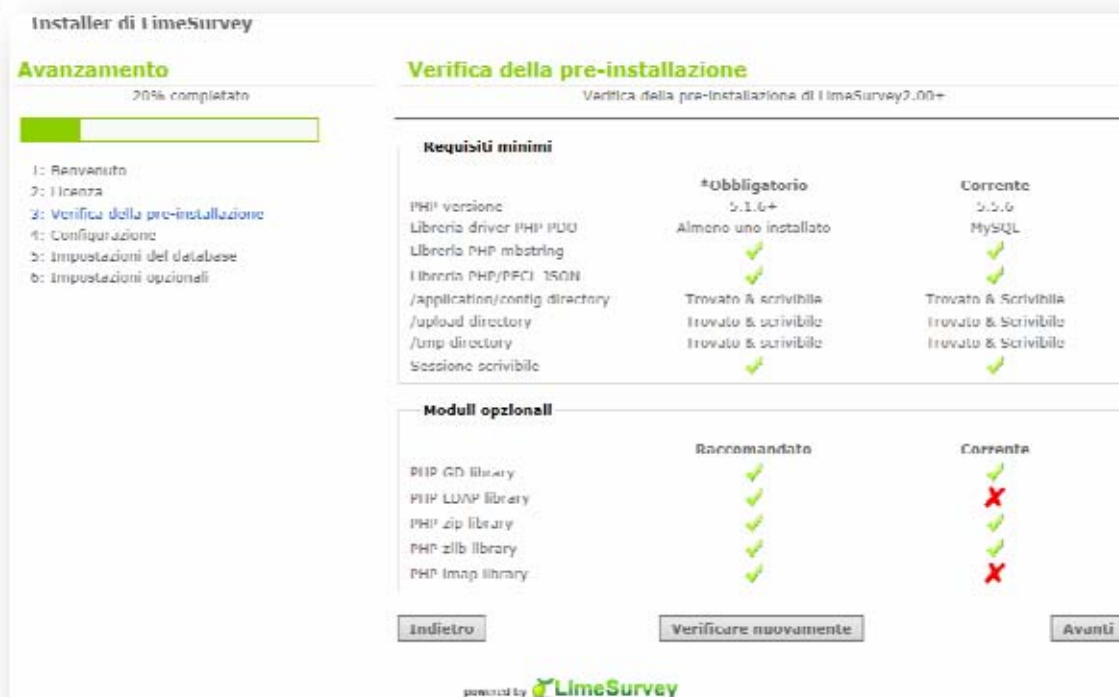


Figura 12.5. Requisiti minimi per l'installazione della piattaforma LimeSurvey nel computer locale.

Andando avanti con la configurazione della piattaforma, si arriverà a dover definire il database MySQL, indispensabile per poter installare correttamente la piattaforma. Per il database bisogna considerare alcune importanti informazioni:

- Tipo di database, ovvero la tipologia del database management system (nel nostro caso creeremo un MySQL)

- Posizione del database, impostare questo alla posizione IP/net del server del database. Nella maggior parte dei casi (come nel nostro), "localhost" funzionerà.
- Database degli utenti, user name del database server. Nella maggior parte dei casi "root" è la scelta corretta.
- Password del database, Password del database server.
- Nome database, se il database non esiste, sarà creato (assicurati che l'utente del database abbia i permessi necessari). Se invece ci sono tabelle esistenti di LimeSurvey nel database, queste saranno aggiornate automaticamente dopo l'installazione.
- Prefisso di tabella, se il database è condiviso, è consigliato il prefisso "lime_", altrimenti lasciare vuota questa impostazione.

Per creare un database per questa piattaforma in locale usando XAMPP, possiamo andare su <http://localhost/phpmyadmin/> e creare un nuovo database (limedb) come da Figura 12.6.

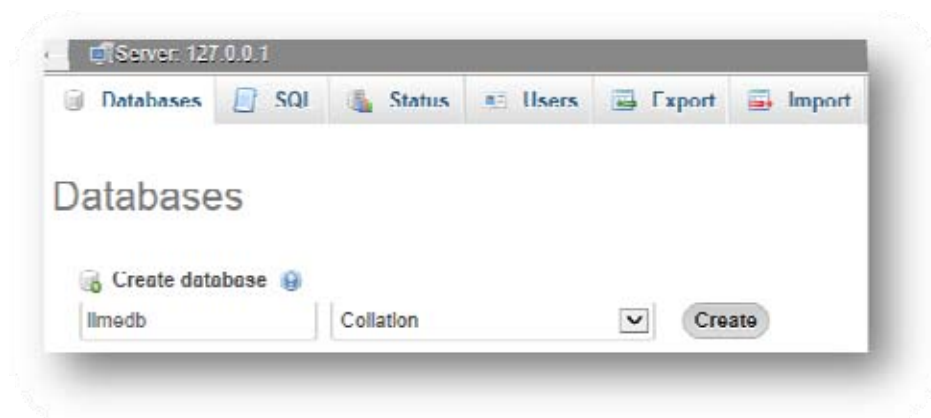


Figura 12.6. Creazione di un database MySQL in locale.

Sarà anche necessario aggiungere un nuovo utente (Check Privileges > Add user), come da Figura 12.7, ad esempio "admin" con una password e poi inserire i dati in LimeSurvey (Figura 12.8).

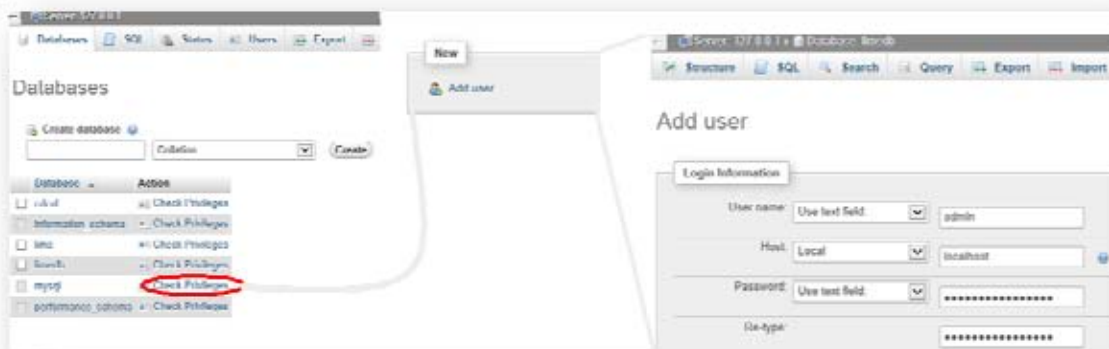


Figura 12.7. Creazione utente nel database MySQL.

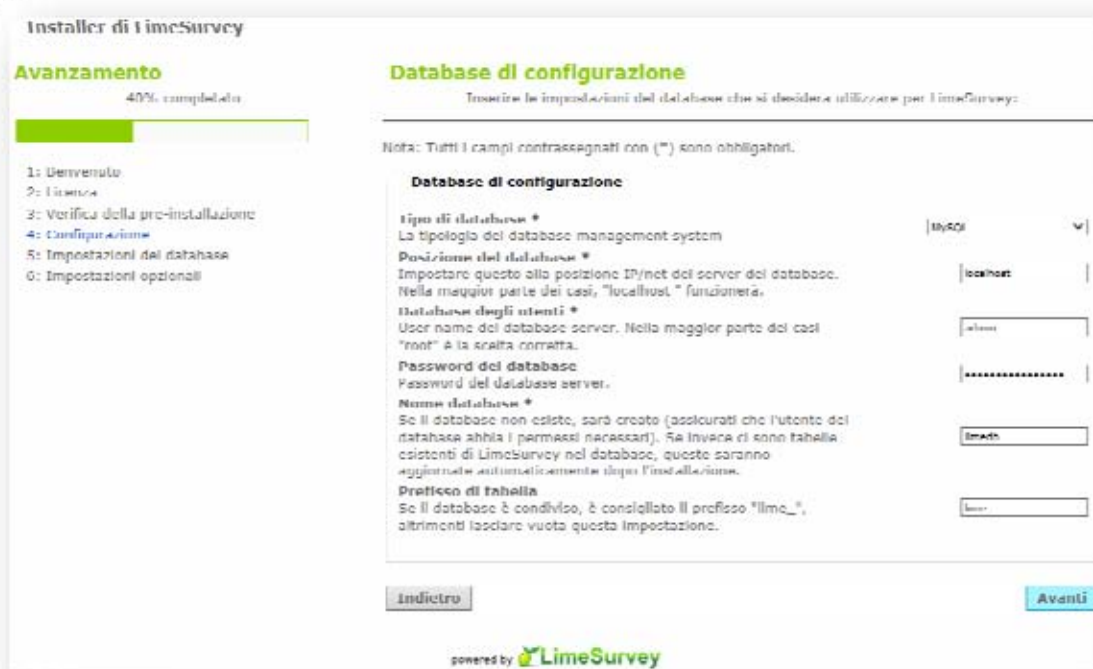


Figura 12.8. Inserimento dei dati di MySQL in LimeSurvey.

Una volta effettuata la configurazione del database MySQL nella piattaforma LimeSurvey, verrà chiesto di popolare il database (automaticamente, con i dati necessari alla piattaforma), come da Figura 12.9.



Figura 12.9. Popolare il database della piattaforma.

A questo punto basta finalizzare il processo scegliendo le impostazioni d'accesso alla piattaforma appena configurata.

Finita la procedura si potrà accedere all'amministrazione di LimeSurvey andando su <http://localhost/lime/admin> e procedere con l'uso della piattaforma come visto al Capitolo 1, dopo avere effettuato il login.

Con una procedura molto simile si possono installare anche altre piattaforme gratuite e open source per la creazione di portali utili in psicologia, come ad esempio Joomla (<http://www.joomla.it>), per piattaforme web avanzate o Moodle (<https://moodle.org>), acronimo

di Modular Object-Oriented Dynamic Learning Environment, che serve per creare piattaforme di eLearning, come quella usata per i materiali di supporto al volume.

La programmazione delle piattaforme presentate è in PHP, acronimo ricorsivo di "PHP: Hypertext Preprocessor", un linguaggio che può essere usato anche per aggiungere elementi alle nostre piattaforme. A titolo d'esempio con la macchina locale XAMPP vediamo, di seguito, il codice per la creazione di una rubrica sul web.

```
<?php include("langsettings.php"); ?>
<html>
<head>
<title>XAMPP phonebook</title>
<link href="xampp.css" rel="stylesheet" type="text/css">
</head>
<body>
&nbsp;<p>
<h1><?php print $TEXT['phonebook-head']?></h1>
<?php print $TEXT['phonebook-text1']?><p>
<?php print @$TEXT['phonebook-text2']?><p>
<?php
// Copyright (C) 2003-2004 Kai Seidler, oswald@apachefriends.org
//
// This program is free software; you can redistribute it and/or
modify
// it under the terms of the GNU General Public License as published
by
// the Free Software Foundation; either version 2 of the License, or
// (at your option) any later version.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied
warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE. See the
```

```
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public
// License
// along with this program; if not, write to the Free Software
// Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

    if(!($db=new SQLite3('sqlite/phonebook.sqlite3', '0666')))
    {
        echo "<h2>".$TEXT['phonebook-error']."</h2>";
        die();
    }
?>
<h2><?php print $TEXT['phonebook-head1']?></h2>
<table border=0 cellpadding=0 cellspacing=0>
<tr bgcolor=#f87820>
<td><img src=img/blank.gif width=10 height=25></td>
<td class=tabhead><img src=img/blank.gif width=150
height=6><br><b><?php print $TEXT['phonebook-
attrib1']?></b></td>
<td class=tabhead><img src=img/blank.gif width=150
height=6><br><b><?php print $TEXT['phonebook-
attrib2']?></b></td>
<td class=tabhead><img src=img/blank.gif width=150
height=6><br><b><?php print $TEXT['phonebook-
attrib3']?></b></td>
<td class=tabhead><img src=img/blank.gif width=50
height=6><br><b><?php print $TEXT['phonebook-
attrib4']?></b></td>
<td><img src=img/blank.gif width=10 height=25></td>
</tr>
<?php
    $firstname=$db->escapeString(@$_REQUEST['firstname']);
    $lastname=$db->escapeString(@$_REQUEST['lastname']);
    $phone=$db->escapeString(@$_REQUEST['phone']);

    if($firstname!="")
    {
        $db->query("INSERT INTO users
(firstname,lastname,phone)
VALUES('$firstname','$lastname','$phone')");
    }
}
```

```

if(@$_REQUEST['action']=="del")
{
    $db->query("DELETE FROM users WHERE
id=".$_REQUEST['id']);
}
$result=$db->query("SELECT id,firstname,lastname,phone
FROM users ORDER BY lastname;");
$i=0;
while( $row=$result->fetchArray(SQLITE3_ASSOC) )
{
    if($i>0)
    {
        echo "<tr valign=bottom>";
        echo "<td bgcolor=#ffffff
background='img/strichel.gif' colspan=6><img src=img/blank.gif
width=1 height=1></td>";
        echo "</tr>";
    }
    echo "<tr valign=center>";
    echo "<td class=tabval><img src=img/blank.gif width=10
height=20></td>";
    echo "<td
class=tabval><b>".$_htmlspecialchars($row['lastname'])."</b></td>";
    echo "<td
class=tabval>".$_htmlspecialchars($row['firstname'])."&nbsp;</td>";
    echo "<td
class=tabval>".$_htmlspecialchars($row['phone'])."&nbsp;</td>";

    echo "<td class=tabval><a onclick=\"return
confirm(\"".$_TEXT['phonebook-sure'].\"");\"
href=phonebook.php?action=del&id=".$_row['id'].\"><span
class=red>".$_TEXT['phonebook-button1'].\"</span></a></td>";
    echo "<td class=tabval></td>";
    echo "</tr>";
    $i++;
}
echo "<tr valign=bottom>";
echo "<td bgcolor=#fb7922 colspan=6><img src=img/blank.gif
width=1 height=8></td>";
echo "</tr>";
$db->close();

```

```

?>
</table>
<h2><?php print $TEXT['phonebook-head2']?></h2>
<form action=phonebook.php method=get>
<table border=0 cellpadding=0 cellspacing=0>
<tr><td><?php print $TEXT['phonebook-attrib1']?>:</td><td><input
type=text size=20 name=lastname></td></tr>
<tr><td><?php print $TEXT['phonebook-attrib2']?>:</td><td> <input
type=text size=20 name=firstname></td></tr>
<tr><td><?php print $TEXT['phonebook-attrib3']?>:</td><td> <input
type=text size=20 name=phone></td></tr>
<tr><td></td><td><input type=submit border=0 value="<?php print
$TEXT['phonebook-button2']?>"></td></tr>
</table>
</form>
<?php include("showcode.php"); ?>
</body>
</html>

```

Il risultato di questo programma PHP è riportato in Figura 12.10, dove si può vedere come permette di aggiungere elementi con un semplice modulo web.

Rubrica (Esempio per PHP+SQLite)
 Uno script per la rubrica molto semplice. Ma implementato con una tecnologia molto moderna: SQLite, il database SQL senza server.

T miei numeri telefonici

Cognome	Nome	Numero telefonico	Comando
Cipressu	Pietro	+39 366 7175081	<input type="button" value="CANCELLA"/>

Aggiungi voce

Cognome:

Nome:

Numero telefonico:

Figura 12.10. Risultato del programma PHP per creare una Rubrica sul web, configurabile dai lettori.

Un database MySQL, può servire anche per creare delle tabelle da poter anche collegare con delle relazione, come prima spiegare.

Per creare una tabella possiamo usare il linguaggio SQL, come possiamo ad esempio vedere nel seguente codice e in Figura 12.11.

```
query SQL:CREATE TABLE `Questionario` (
`Nome` VARCHAR( 30 ) NOT NULL ,
`Cognome` VARCHAR( 30 ) NOT NULL ,
`Pre1` INT( 30 ) NOT NULL ,
`Post1` INT( 30 ) NOT NULL ,
`Pre2` INT( 30 ) NOT NULL ,
`Post2` INT( 30 ) NOT NULL
) ENGINE = MYISAM ;
```



The screenshot shows a MySQL database interface with the following table structure:

Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra	Azione
<input type="checkbox"/> Nome	varchar(30)	latin1_swedish_ci		No			<input type="checkbox"/> Naviga tra i valori DISTINCT / Modifica / Elimina / Primaria
<input type="checkbox"/> Cognome	varchar(30)	latin1_swedish_ci		No			<input type="checkbox"/> Naviga tra i valori DISTINCT / Modifica / Elimina / Primaria
<input type="checkbox"/> Pre1	int(30)			No			<input type="checkbox"/> Naviga tra i valori DISTINCT / Modifica / Elimina / Primaria
<input type="checkbox"/> Post1	int(30)			No			<input type="checkbox"/> Naviga tra i valori DISTINCT / Modifica / Elimina / Primaria
<input type="checkbox"/> Pre2	int(30)			No			<input type="checkbox"/> Naviga tra i valori DISTINCT / Modifica / Elimina / Primaria
<input type="checkbox"/> Post2	int(30)			No			<input type="checkbox"/> Naviga tra i valori DISTINCT / Modifica / Elimina / Primaria

Figura 12.11. Tabella creata direttamente con un codice SQL.

Per gestire i database MySQL esistono sia interfacce web, come quella che abbiamo visto, sia dei veri e propri client da installare localmente sul proprio computer, come HeidiSQL (<http://www.heidisql.com>), che possono essere molto comodi per accedere al database ed effettuare una vasta serie di operazioni e, in particolare delle QUERY, vere e proprie interrogazioni di un database relazionale, che si usano per estrarre informazioni, anche

legate con altre tabelle in modo strutturato e codificato (Figura 12.12).

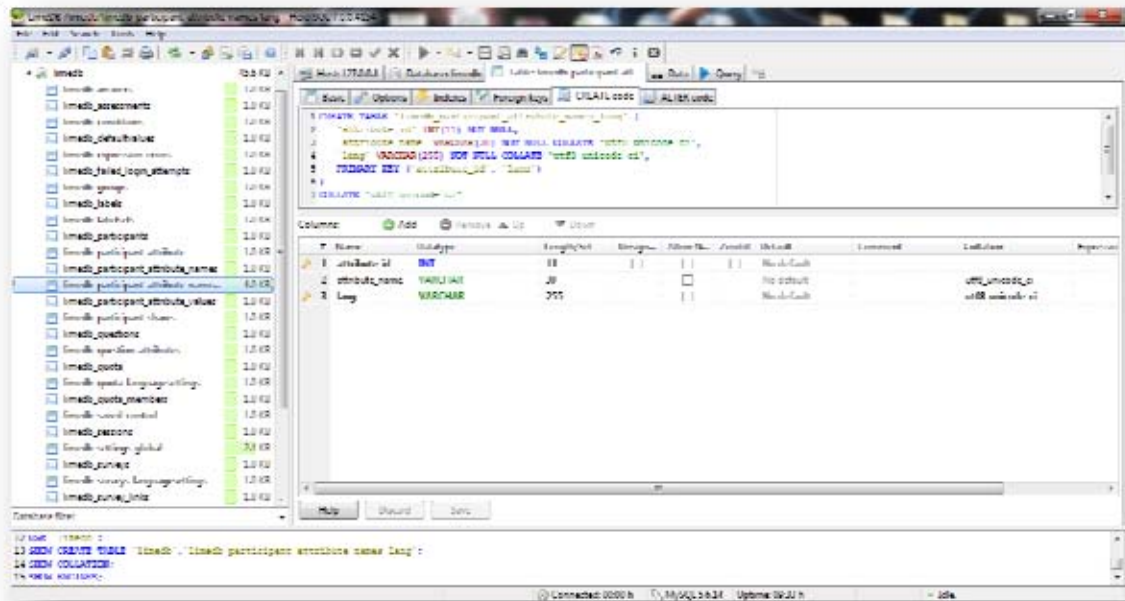


Figura 12.12. Il database MySql, nel client HeidiSQL.

I database sono degli utilissimi strumenti e permettono di poter spostare i dati in un server remoto. Questo ha diversi vantaggi quando il server è ospitato in una sala server professionale (come in molte delle offerte, anche quelle economiche), in quanto da un lato si ha una maggiore sicurezza dei dati (soprattutto se si dispone anche di un backup automatico) rispetto alla memorizzazione sull'hard disco del proprio computer, dall'altro lato si ha la possibilità di accedere remotamente ai dati, tramite world wide web o tramite un apposito client remoto (come HeidiSQL), da

qualsiasi computer dotato di una connessione internet. E' importante tuttavia avere un pieno controllo delle informazioni ospitate nel database e a questo scopo bisogna sempre avere uno schema concettuale del proprio database, con anche le relazioni tra le varie tabelle (in Figura 12.13).

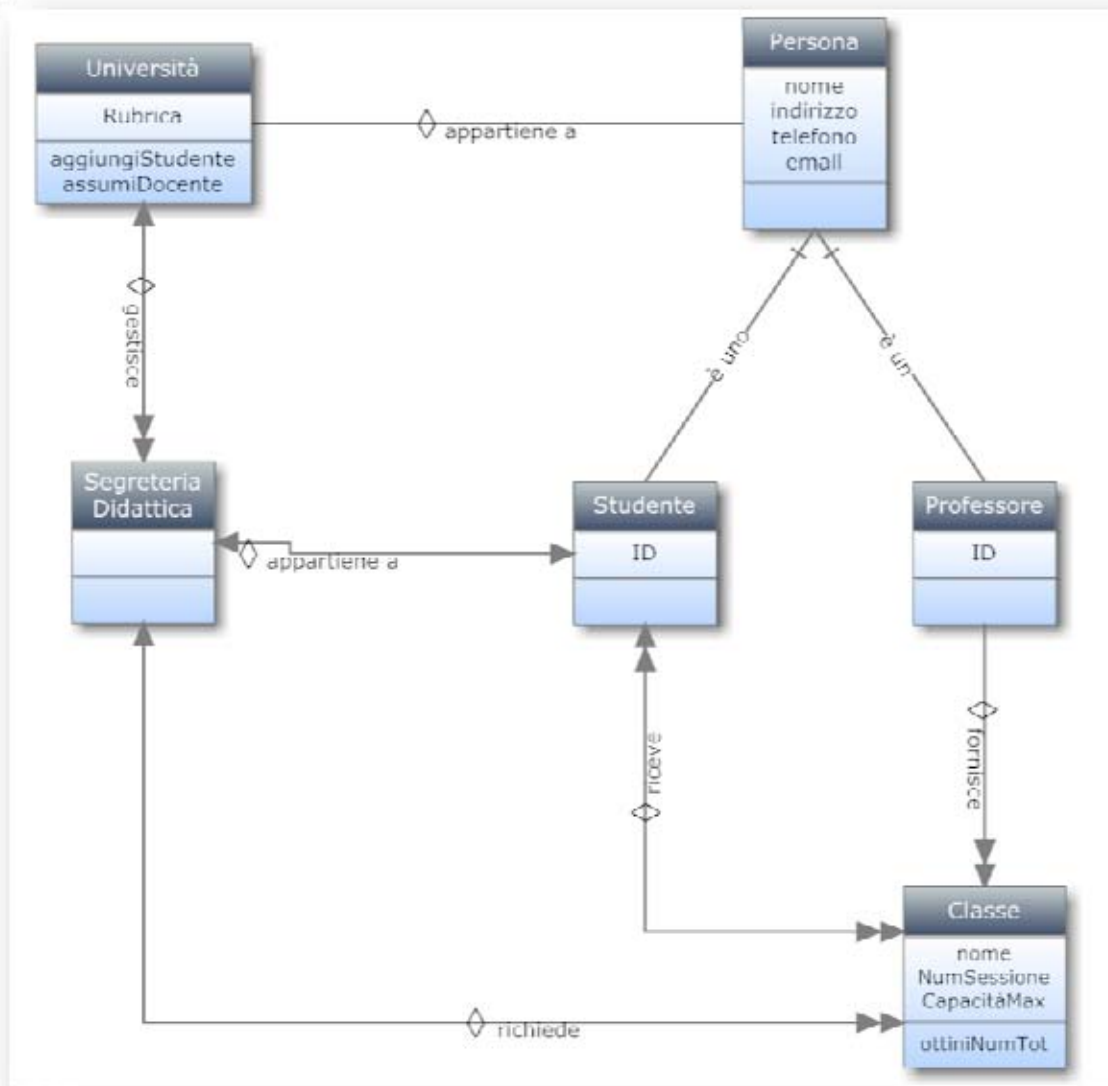


Figura 12.13. Un esempio di schema di database con indicazione delle tabelle e delle relative relazioni.

Importante è anche lo schema di creazione di database e record che potrebbe incorporare una certa complessità, come nell'esempio in Figura 12.14.

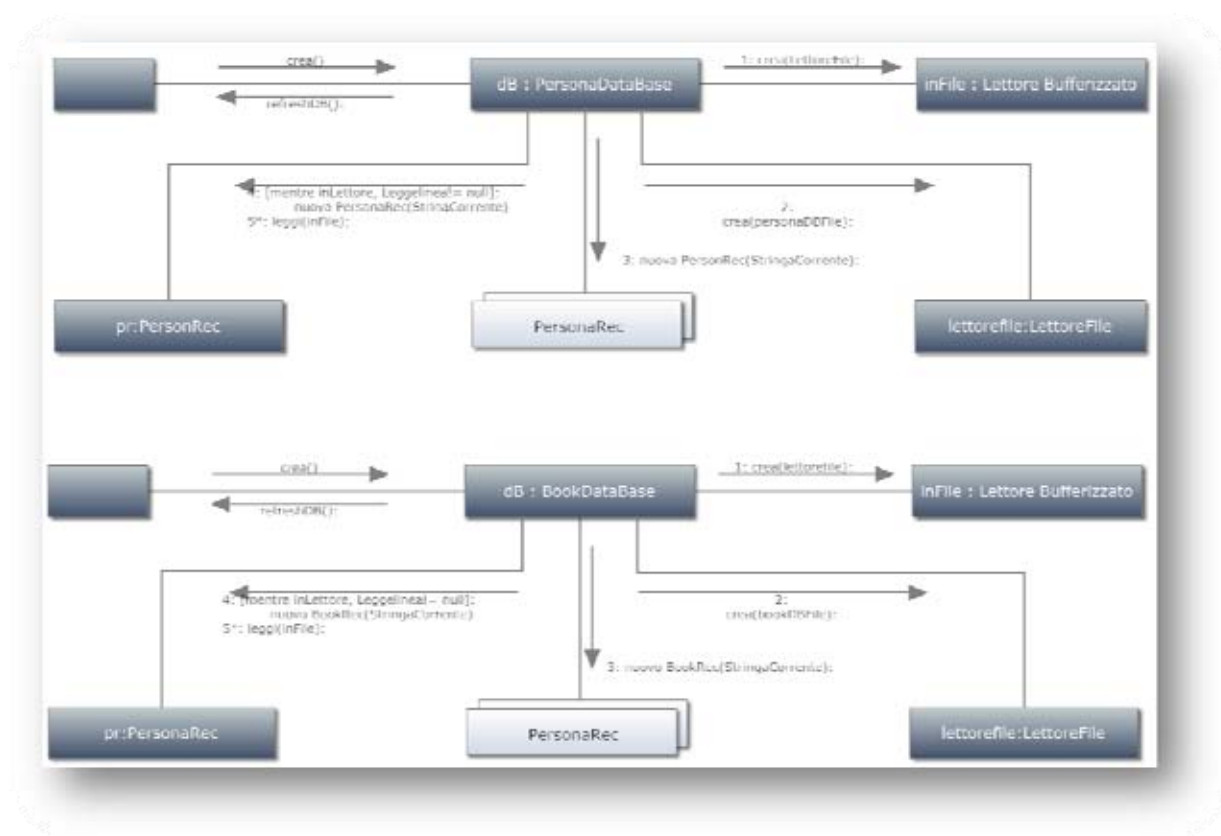


Figura 12.14. Schema di creazione database e record.

Per concludere questo paragrafo, vorrei sottolineare come l'accesso al database può essere gestito anche direttamente tramite codice. Per quanto questo è ovviamente più complesso rispetto all'uso di un semplice client, valgono le considerazioni più volte fatte in questo volume in merito al codice, ovvero man mano che le dimensioni e la complessità dei dati che trattiamo salgono, diventa

più conveniente scrivere qualche riga di codice (magari anche molto semplice) piuttosto che fare un lavoro a mano.

Peraltro se sto già utilizzando un certo codice per una piattaforma, diventa indispensabile avere l'accesso al database già dallo stesso codice della piattaforma, che lo può usare per estrarre le informazioni necessarie a interagire con gli utenti della piattaforma stessa.

Di seguito fornisco il semplice codice d'accesso a un database MySQL tramite PHP, PERL e Python.

Codice PHP

```
<?php
$link = mysql_connect('domain.hostmysql.com', 'username', '*password*');
if (!$link) {
    die('Could not connect: ' . mysql_error());
}
echo 'Connessione avvenuta con successo';
mysql_select_db(database);
?>
```

Codice PERL

```
#!/usr/bin/perl
use DBI;
print "content-type:text/html\n\n";
$db = DBI-
>connect('dbi:mysql:database:domain.hostmysql.com:3306','username','*password*')
or die("Connessione non riuscita");
```

Codice Python

```
#!/usr/bin/python
print "Content-type:text/html\n\n"
import MySQLdb
try:
```

```
conn = MySQLdb.connect (  
    host = "domain.hostmysql.com",  
    user = "username",  
    passwd = "*password*"  
    db = "database")  
except MySQLdb.Error, e:  
    print "Error %d: %s" % (e.args[0], e.args[1])  
    sys.exit (1)  
print "connesso al database"
```

Questi tre programmi permettono di essere facilmente adeguati per le proprie esigenze, indicando i dati del database MySql forniti dal provider del servizio.

12.2 Gestione utenti, contenuti e interfacce

Essendo sul web, un database, ha una posizione privilegiata verso tutti i device che comunicano con un protocollo Internet. In pratica qualunque raccolta è più efficace se posso inserire direttamente i dati acquisiti online, con un'archiviazione automatica. Questo inoltre presenta un altro interessante aspetto, ovvero la possibilità di creare complesse piattaforme che da un lato acquisiscono i dati online direttamente dagli utenti e dall'altro possono fornire un feedback di qualche tipo, basato sui dati, interfacciandosi direttamente all'utente. Vedremo nel capitolo 14 qualche caso concreto, tuttavia mostro qui in Figura 12.15 un'architettura generale di interfaccia tra un database e diversi tipi di utente, con soluzioni fisse o mobili, intermedie da applicazioni, infrastrutture e dei firewall di sicurezza dati.

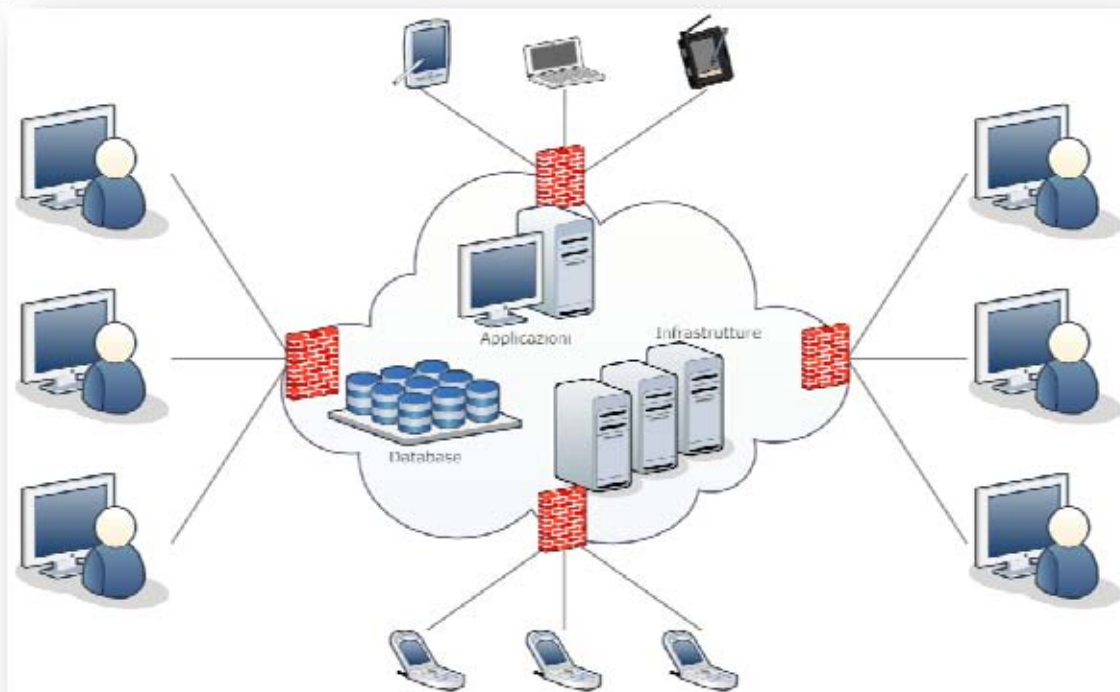


Figura 12.15. Esempio di una piattaforma di interfaccia sicura tra i dati e gli utenti.

Tra le varie piattaforme possibili, in psicologia, assumono particolare rilievo le cosiddette piattaforme di servizio.

In pratica, l'idea è quella di fornire dei feedback all'utente, sia per attivare dei percorsi personalizzati, sia per migliorarne il coinvolgimento. In questi casi bisogna definire delle applicazioni web intermedie da portali specifici, come si può vedere nello schema in Figura 12.16.

Come si può iniziare a intuire le piattaforme di raccolta e gestione dati complessi online, possono essere molto complesse e richiedere sistemi dedicati e personale altamente qualificato.

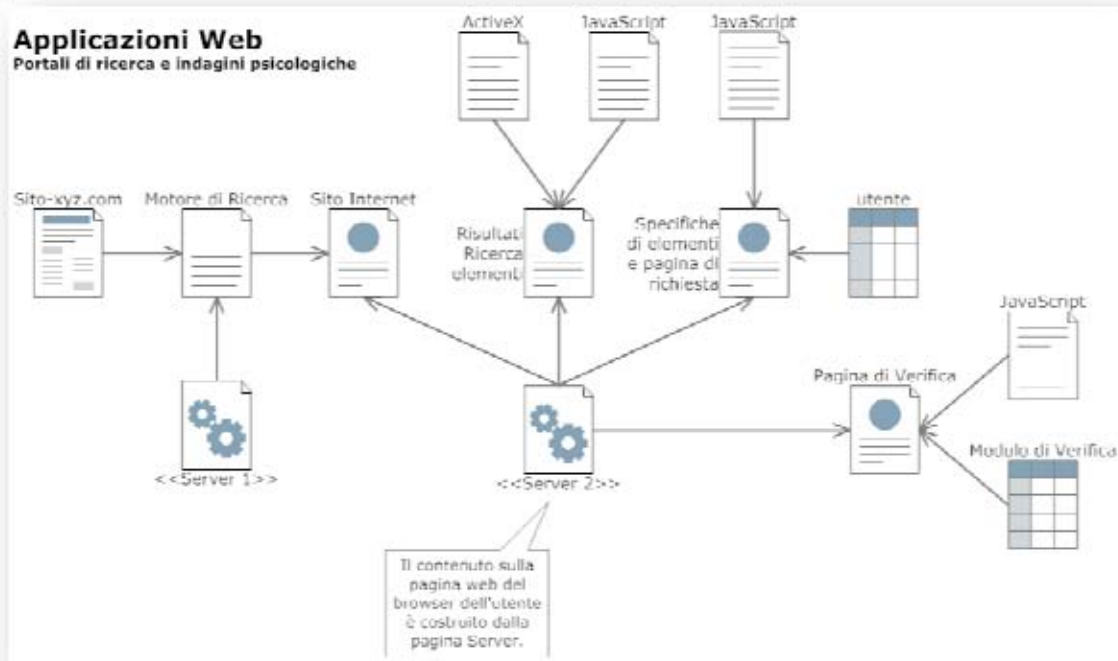


Figura 12.16. Applicazioni web su portali personalizzati.

Ci sono altri due aspetti peraltro che vanno ancora considerati: la gestione utenti e la gestione dei contenuti.

Per quanto riguarda la gestione utenti, va sottolineato come questa debba essere strettamente interconnessa con la gestione delle applicazioni e quindi con il database dati. L'utente si deve interfacciare in modo trasparente a questa complessità, avendo a disposizione un browser e ricevendo le informazioni appositamente filtrate da un "Server di applicazioni" che si preoccupi autonomamente di reperire quanto necessario dal database, elaborandolo per quanto necessario.

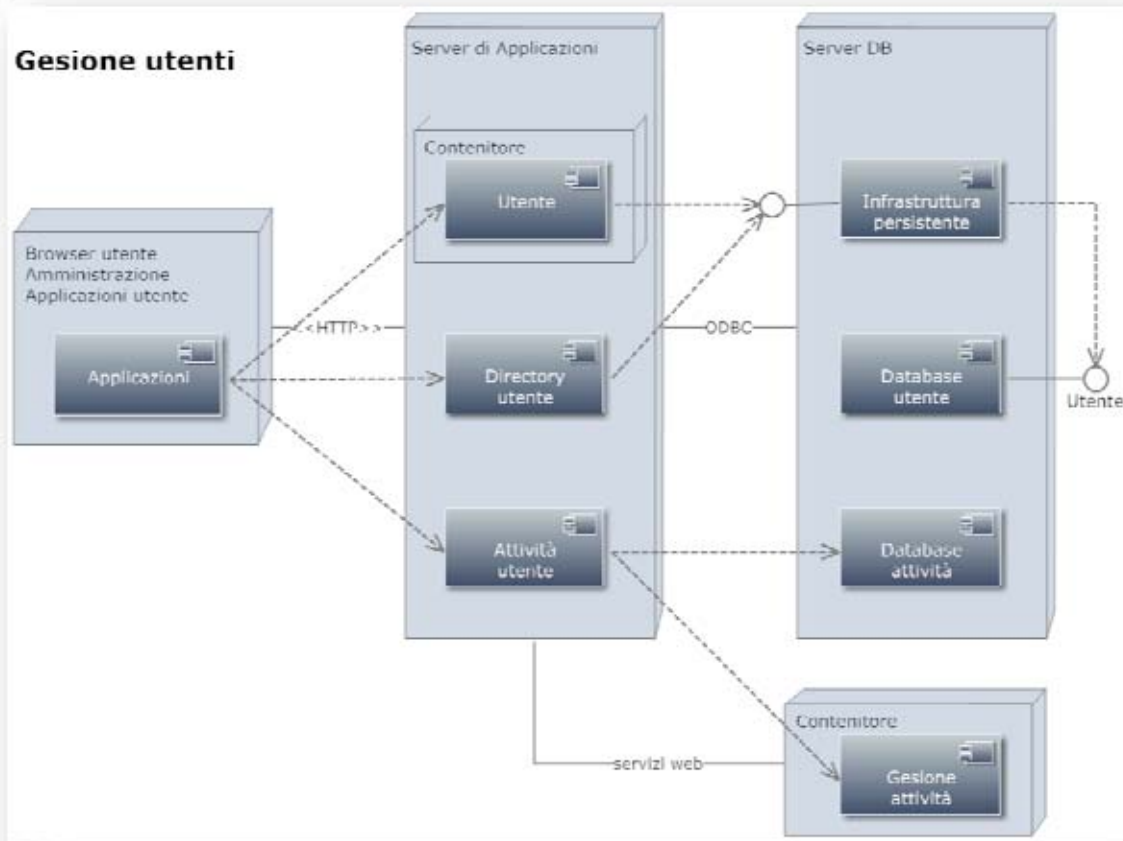


Figura 12.17. Schema di gestione utenti, con un server di intermediazione tra le applicazioni che l'utente riceve in visualizzazione e il database.

Per quanto attiene la gestione dei contenuti il discorso è anche più complesso. Negli ultimi anni c'è stata una forte tendenza verso architetture di tipo Cloud Computing. In questo caso i dati sono "distribuiti" e resi disponibili a diverse tipologie di device (dal computer allo smartphone) che si interfacciano a più livelli con la "nuvola informatica" (il Cloud).

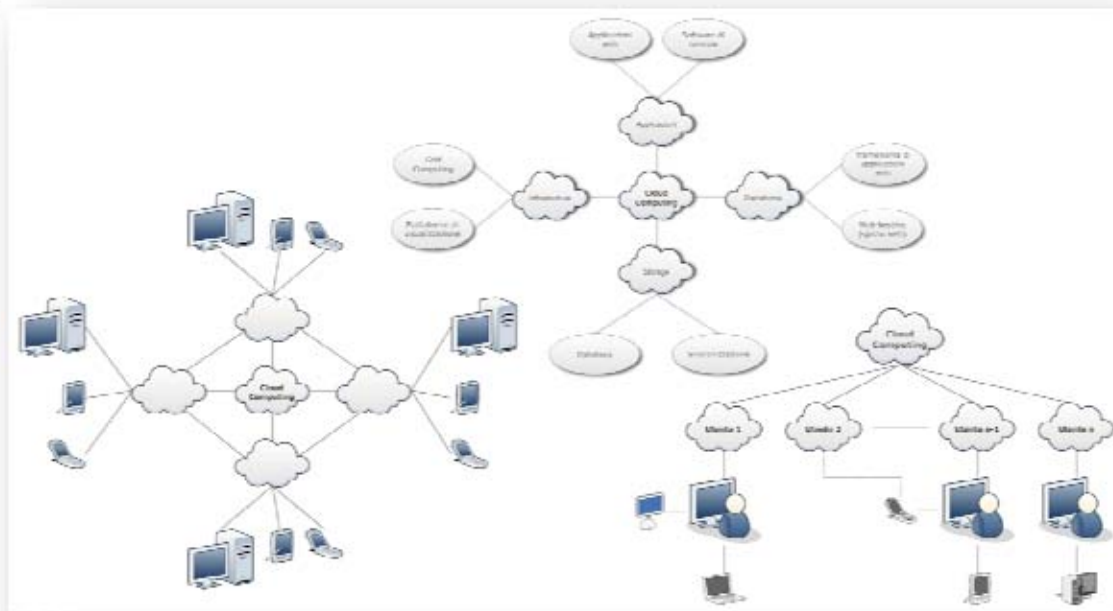


Figura 12.18. Architetture di "Cloud Computing".

Questo tipo di architettura agevola le tecnologie che accedono da ovunque ai dati, tanto che si parla sempre più di "pervasive and ubiquitous computing", ambito in cui la psicomatria computazionale gioca un ruolo fondamentale nel fare da ponte tra tecnici e psicologi, per mezzo di un linguaggio condiviso.

Gestire e organizzare i dati relazionali

Come abbiamo visto nel secondo paragrafo del quinto capitolo del volume, da un punto di vista strutturale i dati relazionali sono facilmente implementabili in forma di matrici quadrate, ovvero matrici aventi un numero di righe uguale al numero di colonne.

In termini di algebra lineare (così come nei capitoli precedenti), si rappresentano le relazioni tra individui indicando sulla matrice di relazioni (ad esempio "MatRel") ogni rapporto dall' i -esimo individuo al j -esimo tramite la cella della matrice "MatRel(i,j)", in cui il caso " $i=j$ " indica la relazione con se stesso, che può avere senso in base a quello che indica la rete: se parliamo di sentimenti amorosi questo caso potrebbe ad esempio indicare l'amor proprio.

Le varie operazioni sulle matrici, che abbiamo approfondito in questa seconda parte del volume, possono dunque essere usate per organizzare le informazioni dei dati relazionali. Quando le matrici

di relazioni riguardano pochi individui, si potrà anche usare un normale foglio di calcolo, come Microsoft Excel. Se invece, il numero di individui e/o il numero di relazioni crescono, avrà maggiormente senso usare programmi per gestire questo tipo di complessità tramite le varie strutture in precedenza esaminate (Celle, Matrici, Vettori, scalari, ...) e facendo uso dei più adeguati strumenti computazionali.

13.1 Dati relazionali di rete tramite matrici quadrate

Come abbiamo visto, i dati relazionali possono agevolmente essere rappresentati in matrici quadrate di ordine n , dove n è il numero di individui (detti nodi) sulla rete. Gli individui sulla rete sono tra di loro connessi secondo un qualche tipo di relazione (amicale, familiare, ecc.), tuttavia le reti possono rappresentare, anche in psicologia, una vasta serie di fenomeni. Possiamo ad esempio pensare ad una rete di individui dove i legami nella rete indicano la comunicazione da uno all'altro, oppure i sorrisi scambiati o le email scambiate o l'amicizia su Facebook. Le reti dunque possono rappresentare anche cose assai diverse tra loro e bisogna prestare molta attenzione al tipo di interpretazioni che si fanno a fare tramite questo strumento.

Da un punto di vista più formale rappresentiamo le reti come visto prima, con una matrice, ad esempio "MatRel" le cui celle sono identificate tramite la notazione "MatRel(i,j)".

Quando per ogni i e per ogni j da uno a n , succede che

$$\text{MatRel}(i,j) = \text{MatRel}(j,i)$$

allora possiamo affermare che la matrice è simmetrica e, in pratica, la nostra rete sarà una rete non-diretta in quanto non conta la direzione da i a j o da j a i , ma conta il legame (i,j) che è uguale al legame (j,i) .

Se la matrice $\text{MatRel}(i,j)$, per ogni i e per ogni j è composta tutta da valori zero o valori uno, la rete non è pesata quindi consideriamo l'esistenza o meno dei legami e non il peso che questi legami hanno. Ad esempio, una rete sugli scambi di comunicazione via email sarà pesata se conto il numero di email scambiate da i a j e assegno al legame $\text{MatRel}(i,j)$ un valore pari a questo numero, in caso contrario, ovvero se l'esistenza di una o più email genera il valore 1 se esistente (anche con più email) o zero se non esistente, allora la rete è non pesata.

Operativamente la notazione in matrice è anche molto utile quando si voglia lavorare sui dati di rete per effettuare una ristrutturazione. Questa strutturazione serve quando si vuole usare un programma per l'elaborazione di indici di rete che richiede i dati in un modo differente. In particolare alcuni programmi invece di chiedere la matrice delle relazioni, chiedono l'indicazione in sequenza di ciascuna relazione (se conta l'ordine sarà una rete diretta) con accanto il peso della relazione (se non è sempre 0 o 1, sarà una rete pesata). Questo modo di strutturare le reti è

comunque corretto e per certi versi può essere anche più immediato e decisamente più intuitivo, in quanto mette in relazione ogni individuo con tutti gli altri della rete e a ciascuno vi assegna un eventuale peso.

Consideriamo adesso l'esempio fatto al capitolo cinque per costruire una matrice quadrata tra tre individui in rete.

Secondo quanto acquisito in questa seconda parte del volume, possiamo direttamente imputare la matrice in Matlab tramite il seguente comando per creare la matrice (A):

$$A=[0,7,3;3,0,1;1,6,0];$$

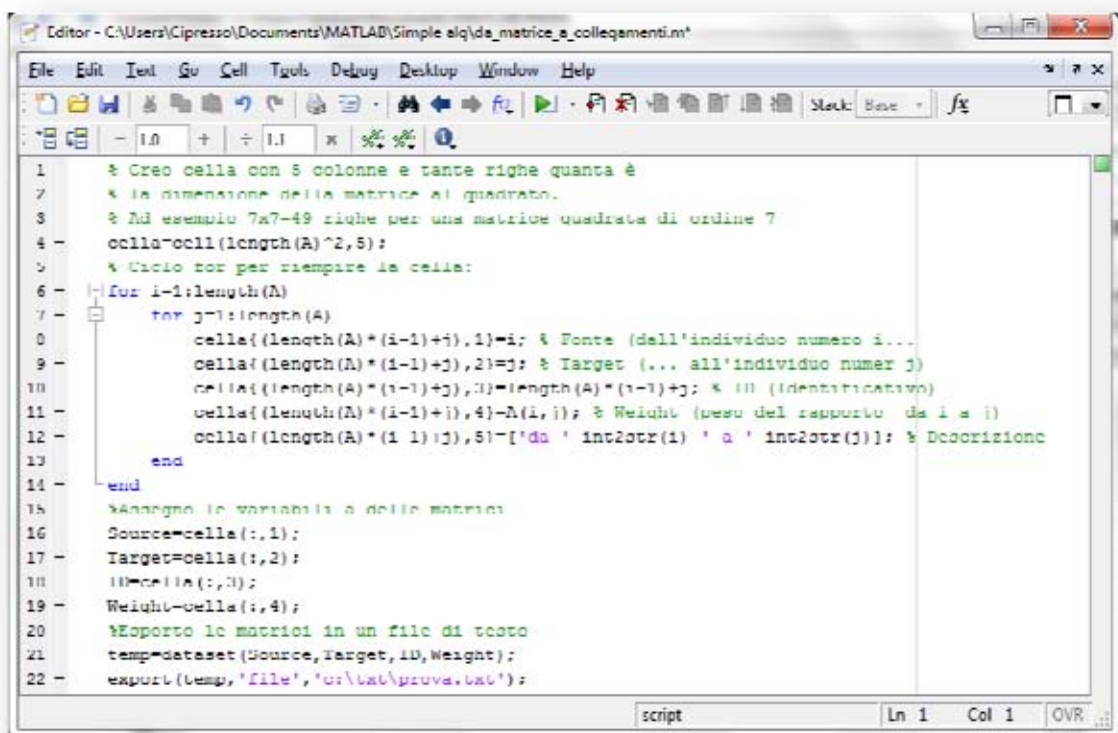
La matrice A rappresenta dunque una matrice quadrata di ordine 3x3 ed è identica a quella vista in figura 5.5.

Per quanto è possibile ed anche semplice descrivere la matrice A in termini di relazioni, quello che adesso faremo è automatizzare il processo, in modo che se invece di avere una matrice quadrata di ordine tre ne abbiamo una di ordine trecento, metteremo esattamente lo stesso sforzo e impiegheremo circa lo stesso tempo. Abbiamo visto nei capitoli precedenti come manipolare le matrici per ottenere i dati in forme e strutture diverse. Tuttavia il programma per eseguire questa operazione è un tantino più complesso e richiede un ragionamento in termini di righe da implementare nella nuova matrice passo dopo passo. Infatti l'identificazione della riga che di volta a volta andiamo a scrivere nella nuova matrice non è di immediata comprensione, ma richiede

una certa esperienza in questo tipo di calcoli. In particolare bisognerà, soprattutto all'inizio, scrivere una sequenza di incrementi delle due variabili che fungono da indicatori ("i" e "j") simulando i due cicli "for". Dopo un po' di tentativi arriverete a definire la riga come segue:

$$\text{length}(A)*(i-1)+j$$

e sarà a quel punto più semplice definire il codice in Figura 13.1.



```

1  % Creo cella con 5 colonne e tante righe quanta è
2  % la dimensione della matrice al quadrato.
3  % Ad esempio 7x7-49 righe per una matrice quadrata di ordine 7
4  cella=cell(length(A)^2,5);
5  % Ciclo for per riempire la cella:
6  for i=1:length(A)
7      for j=1:length(A)
8          cella{(length(A)*(i-1)+j),1}=i; % Fonte (dell'individuo numero i...)
9          cella{(length(A)*(i-1)+j),2}=j; % Target (... all'individuo numer j)
10         cella{(length(A)*(i-1)+j),3}=length(A)*(i-1)+j; % ID (identificativo)
11         cella{(length(A)*(i-1)+j),4}=A(i,j); % Weight (peso del rapporto da i a j)
12         cella{(length(A)*(i-1)+j),5}=[ 'da ' int2str(i) ' a ' int2str(j)]; % Descrizione
13     end
14 end
15 %Assegno le variabili a delle matrici
16 Source=cella(:,1);
17 Target=cella(:,2);
18 ID=cella(:,3);
19 Weight=cella(:,4);
20 %Esporto le matrici in un file di testo
21 temp=dataset(Source,Target,ID,Weight);
22 export(temp,'file','c:\lab\pruva.txt');
  
```

Figura 13.1. Programma "da_matrice_a_collegamenti.m" che trasforma una matrice quadrata di qualunque ordine in un file di testo dei collegamenti elencati in sequenza.

il codice `length(A)*(i-1)+j` rappresenta dunque la dimensione della matrice quadrata moltiplicata per l'indice "i" (del ciclo "for" più "esterno") tolto 1. Così facendo al primo giro di "j" (del ciclo "for" più "interno") si hanno gli stessi valori di "j", al secondo ciclo di "j" ("i=2") gli si sommeranno i valori di `length(A)*1`, ovvero a "j" si sommerà la dimensione della matrice (in una matrice di ordine 7 si sommerà 7), al terzo giro si sommeranno ai valori "j" `length(A)*2` (in una matrice di ordine 7 si sommerà 14) e così via fino all'ultima riga che sarà al valore `length(A)*length(A) = length(A)^2` ovvero la dimensione della nuova matrice.

Capisco che può sembrare un po' complesso, ma una volta acquisita una certa pratica sarete in grado di effettuare questi ragionamenti in piena autonomia. Per il momento ci si può affidare al programma `da_matrice_a_collegamenti.m` che ho realizzato per qualunque vostro uso futuro: funziona infatti con matrici quadrate di qualsiasi dimensione, ma ricordate di creare la cartella "txt" sotto il disco c. In Figura 13.2 le variabili in Matlab.

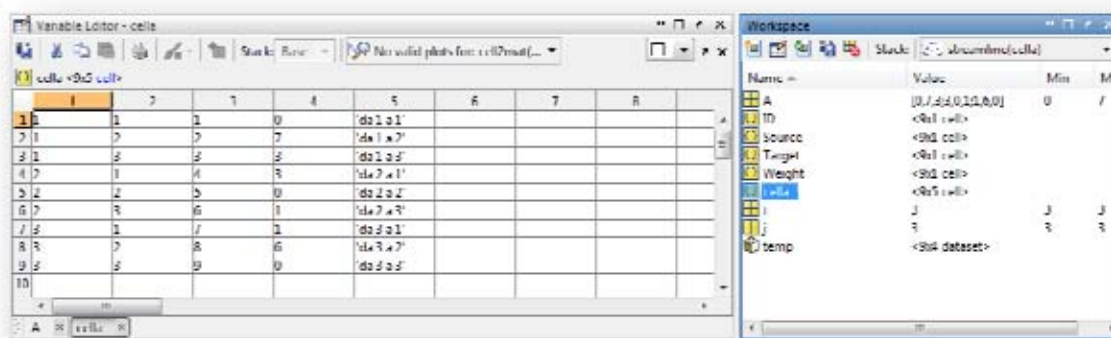


Figura 13.2. La cella creata dal programma, vista in Matlab.

In Figura 13.3 si può invece vedere l'output generato per la matrice di relazioni di rete, in un file txt delimitato da tab (tabulazioni) pronto per essere importato nei programmi che richiedano tale formato, come vedremo nel prossimo paragrafo.

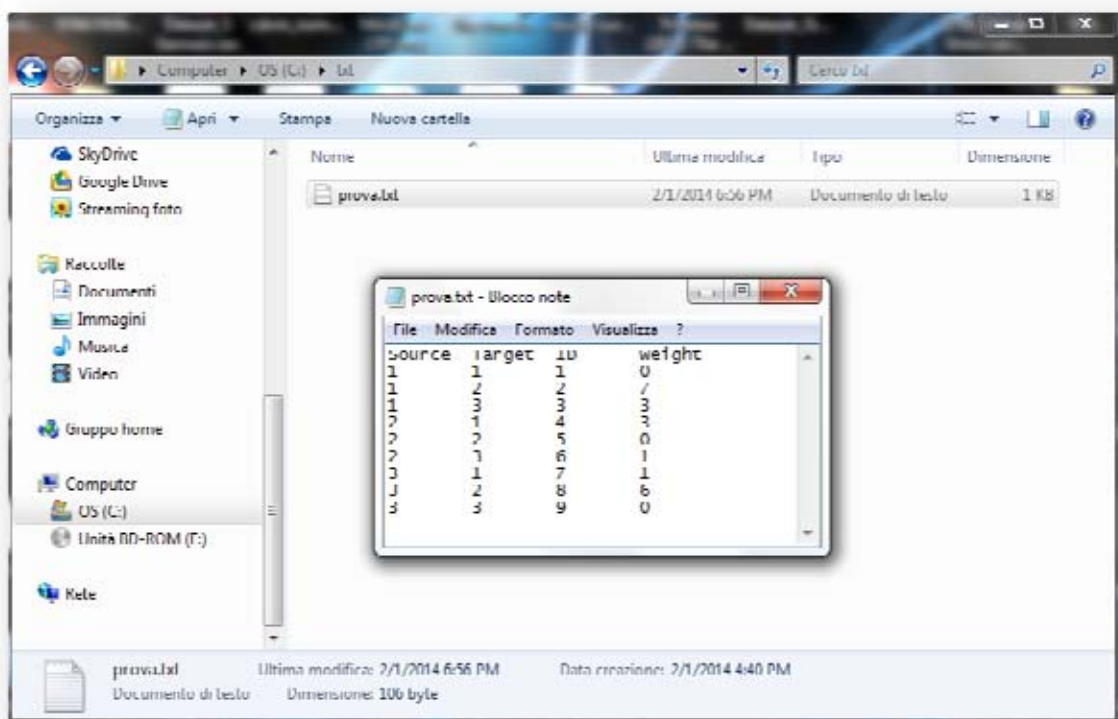


Figura 13.3. Il file "prova.txt" generato dal programma Matlab.

Nel caso in cui il file di testo dovesse essere adattato per esigenze particolari del programma di elaborazione dei dati di rete, è sempre possibile modificare il codice tramite piccoli aggiustamenti. Basterà ad esempio cambiare il nome delle variabili e/o selezionarne solo alcune dalla penultima riga del codice, per ottenere un output adeguato alle proprie esigenze.

13.2 Analizzare i dati relazionali di rete

L'analisi dei dati di rete, risolti i problemi relativi al formato di tali dati sperabilmente risolti dopo la lettura del precedente paragrafo, può essere effettuata per mezzo di programmi nati e da sempre usati proprio a tale scopo. Esistono attualmente centinaia di programmi che si occupano di Social Network Analysis e sarebbe impossibile nel breve spazio di questo volume esaminarne le caratteristiche anche solo di alcuni. Dal momento che lo scopo di questo volume è quello di fornire le basi della psicomatria computazionale, era assolutamente indispensabile dare qualche indicazione su come gestire i dati relazionali per essere pronti per questi programmi, che il lettore interessato potrà adesso esplorare con la certezza, costruita tramite lo studio del volume, di avere molti strumenti per potere fornire i dati necessari nel formato che occorre a un qualunque programma di questo tipo.

Detto ciò, sento comunque di dover dare alcuni dettagli su almeno un paio di strumenti, ovviamente gratuiti, per potere cominciare il divertimento con il fantastico mondo dell'analisi delle reti sociali.

Il primo semplicissimo software che presento si chiama NodeXL (<http://nodexl.codeplex.com/>), un software gratuito e open source che si configura come template di Microsoft Excel per le versioni 2007, 2010 e 2013. NodeXL rende possibile l'esplorazione grafica delle reti in modo molto semplice: basterà fornire la lista dei collegamenti (da un individuo a un altro) e premere un bottone per vederne la visualizzazione in formato di rete.

Una volta installato, NodeXL si aprirà all'interno di Excel, in particolare con un menu dedicato, al cui interno è possibile trovare il pulsante "Sample workbook", un semplice esempio (Figura 13.4) per poter iniziare subito a esplorare le varie caratteristiche e funzioni disponibili.

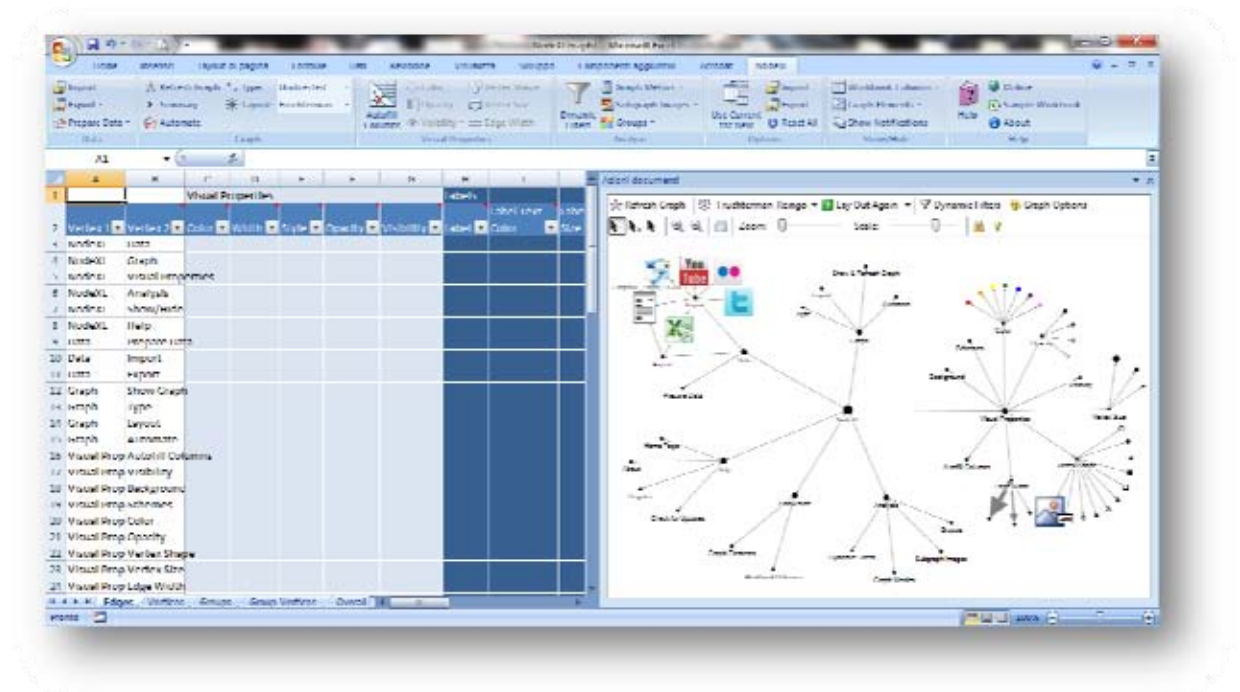


Figura 13.4. Il "Sample workbook" di NodeXL.

La struttura di acquisizione dei dati di NodeXL è basata su due legami e può essere usato il programma Matlab prima esposto per la trasformazione dei dati da matrice e elenco di collegamenti, includendo le prime due colonne e modificando i numeri in nomi a piacimento, una volta importati.

NodeXL è un software molto semplice e l'apprendimento è agevolato dal potersi muovere all'interno di Excel, un programma generalmente già conosciuto dai potenziali utenti.

In Figura 13.5, possiamo vedere la rete di personaggi del romanzo "Les Misérables", che non ha certo bisogno di presentazioni.

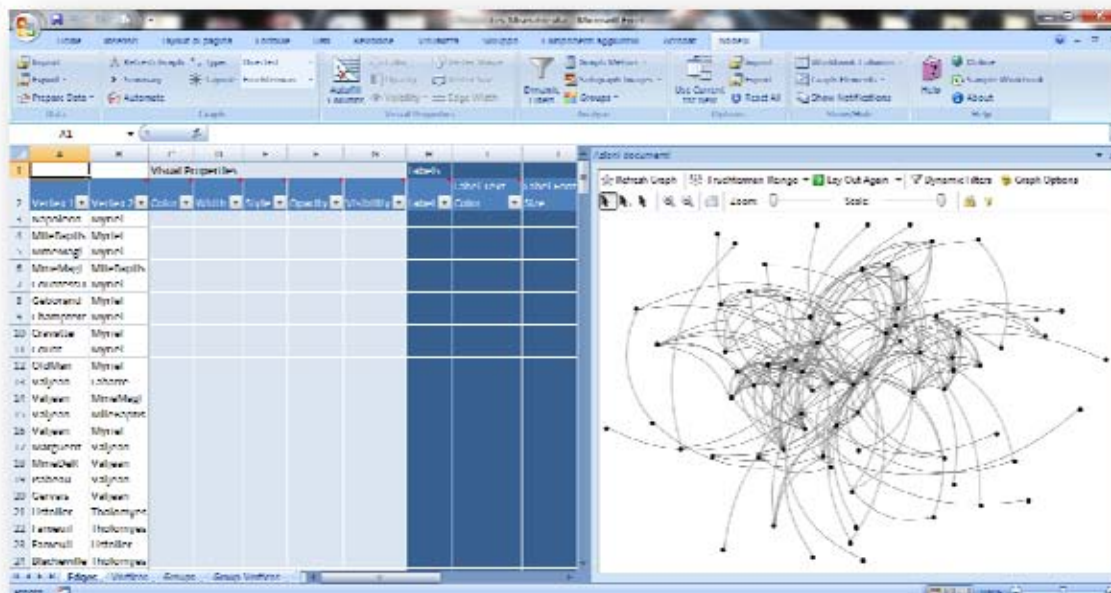


Figura 13.5. Rete dei personaggi del romanzo "Les Misérables".

Se usato con devozione NodeXL è anche capace di produrre risultati assai convincenti per la visualizzazione (si veda ad esempio la Figura 13.6) e ritengo che, se ulteriormente semplificato, può diventare un primo strumento di visualizzazione per chi inizia ad avvicinarsi a questo mondo. Va tuttavia sottolineato che i limiti di Excel, già discussi al secondo capitolo,

Vediamo adesso un altro software che permette il calcolo di indici più avanzati e la gestione di reti dirette e/o pesate.

L'Open Graph Viz Platform Gephi (<http://gephi.org/>) è una piattaforma di visualizzazione ed esplorazione per tutti i tipi di reti e sistemi complessi (di cui parleremo nella terza parte del volume), siano essi grafi dinamici o gerarchici. Funziona sotto Windows, Linux e Mac OS X. Inoltre, Gephi è open source e ovviamente gratuito.

Al paragrafo precedente, abbiamo convertito la matrice dei dati relazionali di rete, in formato testo con i dati delle relazioni in forma di lista e includendo il peso di ciascuna relazione indicata. Adesso, come da Figura 13.7, possiamo importare il file "prova.txt" in Gephi prima creato.

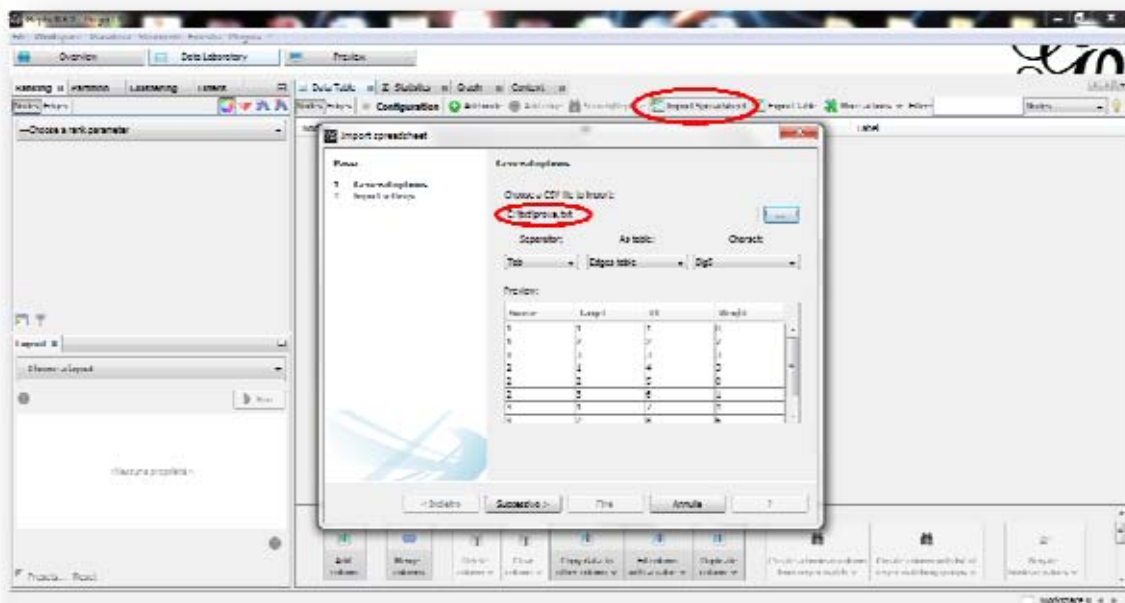


Figura 13.7. Importazione dei dati di "prova.txt" in Gephi.

Il programma per Matlab "da_matrice_a_collegamenti.m" è stato da me appositamente creato per il formato di importazione di Gephi che richiede specificamente l'indicazione di almeno tre colonne (Source, Target, ID) con l'eventuale aggiunta del peso (Weight). Una volta importati i dati in Gephi possiamo dare un nome a ciascun nodo (in questo caso individuo) della rete, come in Figura 13.8. In Gephi i nodi sono chiamati "nodes" e i collegamenti (ad esempio le relazioni) tra nodi "edges".

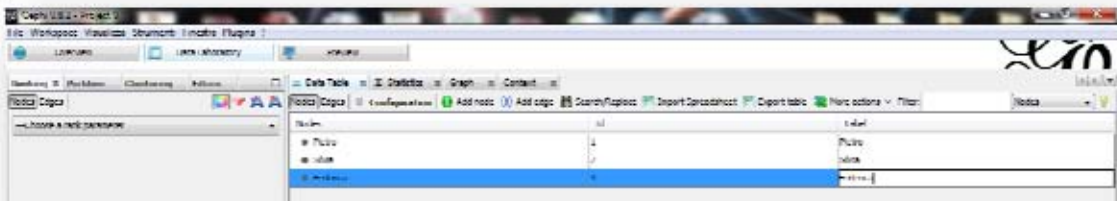


Figura 13.8. Labeling dei nomi ai nodi (individui) della rete.

Gli "edges" (Figura 13.9) a volte sono anche detti "links". Nel gergo tecnico della Network Analysis sono più spesso detti "ties".

Source	Target	Type	ID	Label	Weight
1	2	Directed	1		1
2	1	Directed	2		1
3	1	Directed	3		1
2	4	Directed	4		1
1	3	Directed	5		1
2	4	Directed	6		1
3	4	Directed	7		1
4	2	Directed	8		1
1	3	Directed	9		1

Figura 13.9. Edges, ovvero i collegamenti tra i nodi e il loro peso.

In Gephi esiste una serie preconfezionata di algoritmi per il calcolo automatico dei principali indici usati nell'analisi delle reti. Per calcolarli sulla rete appena importata è sufficiente cliccare sul tab "statistics" e poi sui vari "Run" delle funzioni (Figura 13.10).

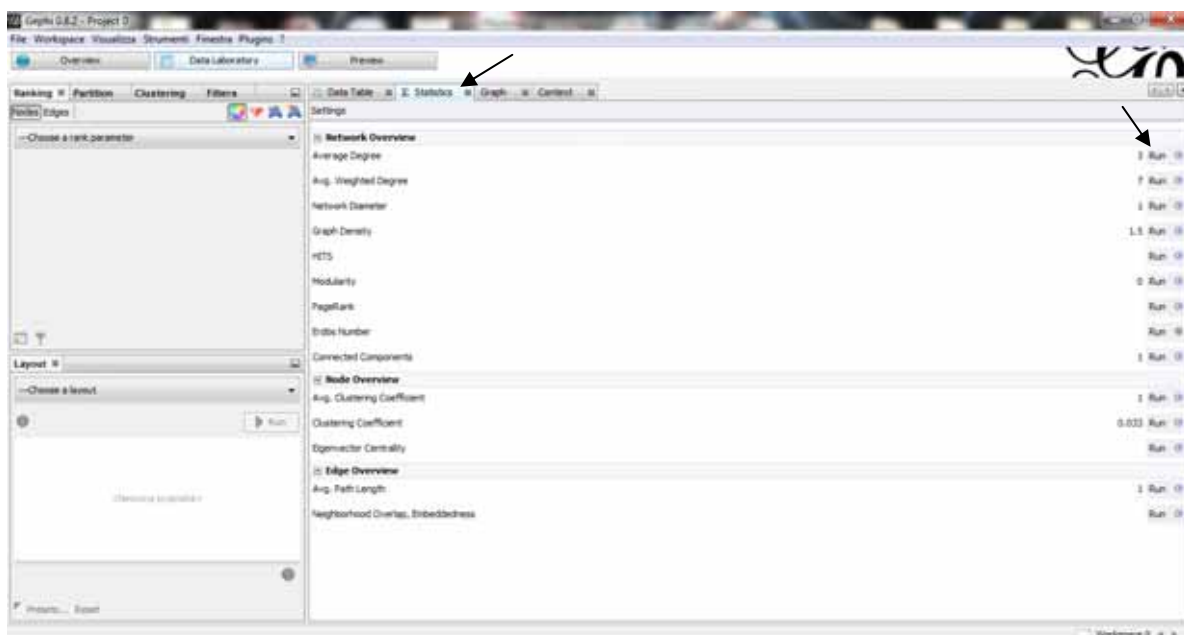


Figura 13.10. Calcolo dei principali indici usati per le reti.

Una volta calcolati gli indici, i valori corrispondenti a ciascun nodo saranno automaticamente disponibili tra i dati (Figura 13.11).

Node	Label
1
2
3

Figura 13.11. Gli indici relativi ai nodi sono riportati in tabella.

A volte, come abbiamo visto nel capitolo precedente, i dati o per raccolta (come per i questionari) o per intrinseca complessità strutturale (come per i Big Data), sono archiviati in un database relazionale. Abbiamo visto come sia possibile esportare tali dati per renderli disponibili all'elaborazione tramite Matlab o GNU Octave. Tuttavia, se possibile, potrebbe essere più semplice e meno complesso importare tali dati, archiviati in un database, direttamente dentro Gephi per la visualizzazione della rete e per l'elaborazione dei relativi indici (Figura 13.12).

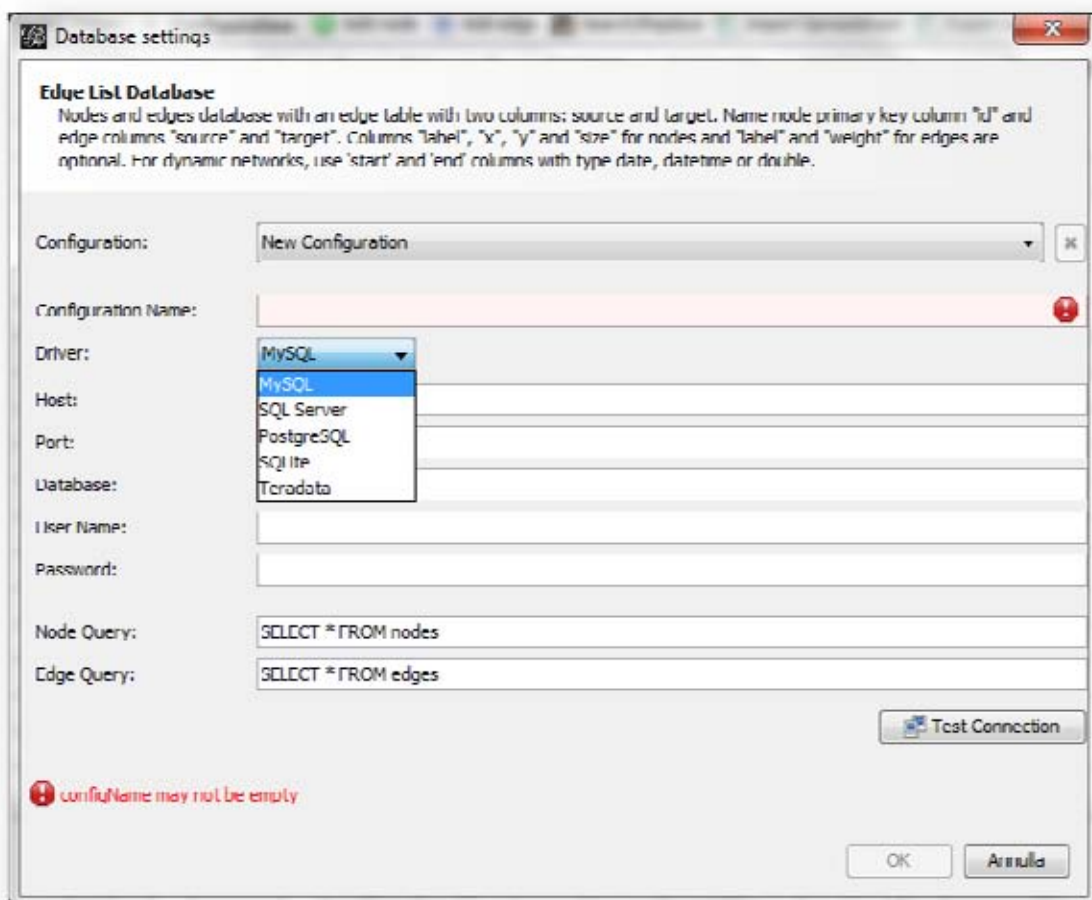


Figura 13.12. Importazione degli edges da un database.

Gephi mette a disposizione anche la possibilità di generare alcune reti tipicamente usate per la descrizione dei sistemi complessi (Figura 13.13), ma di questo avremo modo di parlare in modo più approfondito nella terza parte del volume.

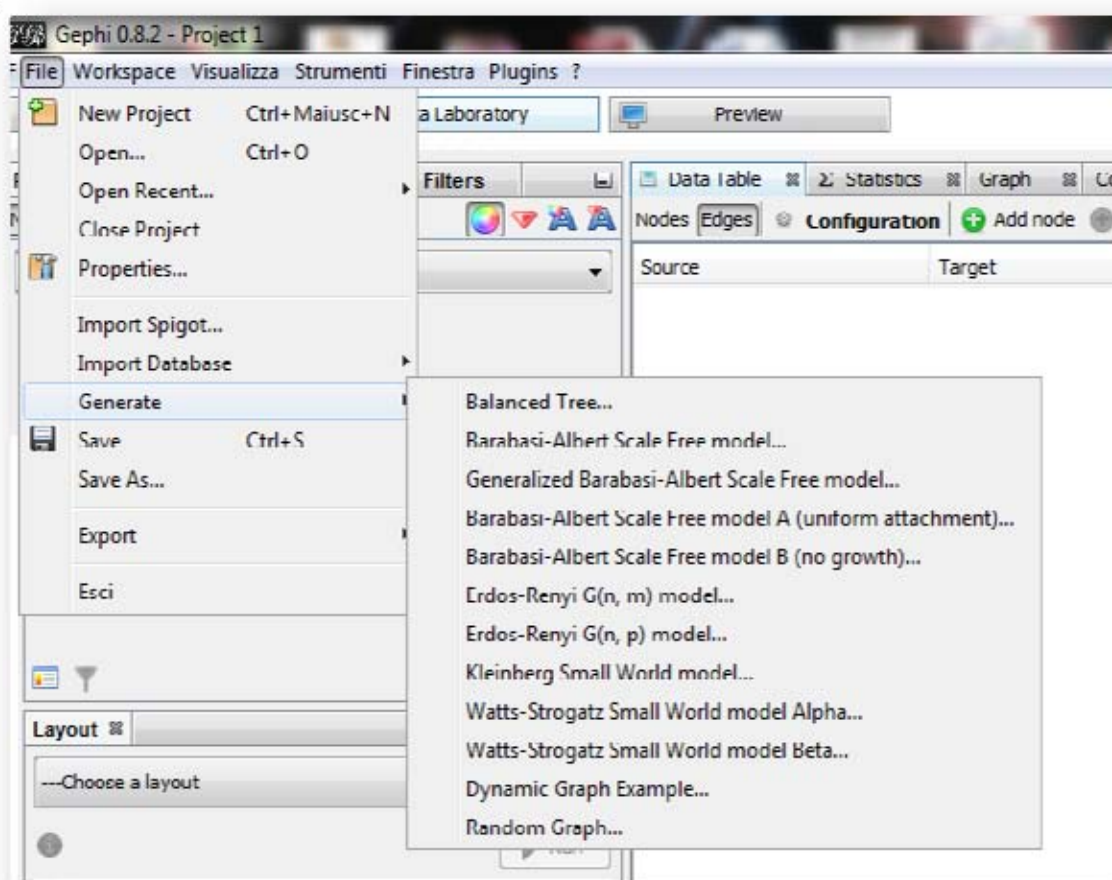


Figura 13.13. Generazione di reti di esempio, secondo i classici modelli studiati in letteratura.

Il lettore che non abbia mai fatto analisi delle reti, probabilmente, sarà maggiormente interessato a usare delle reti reali per

cominciare a fare alcune analisi e sperimentare magari su una propria rete sociale. Ci sono tre modalità che posso a tal proposito suggerire. La prima, in pratica già vista, è di crearsi una propria matrice di collegamenti, costruendo diverse matrici sulla base di relazioni, contatti, comunicazione e qualunque altra cosa venga in mente, con l'accortezza di stare assai attenti nell'interpretazione. Le altre due modalità sono le email e Facebook, uno dei siti di social networking più usati al mondo.

Per quanto riguarda l'estrazione dei dati di email, esiste in Gephi un semplicissimo modulo che vi consente l'importazione automatica dal server di posta elettronica. In figura 13.14, mostro l'importazione da Gmail (ricordate di indicare la porta 993 con l'uso di SSL).

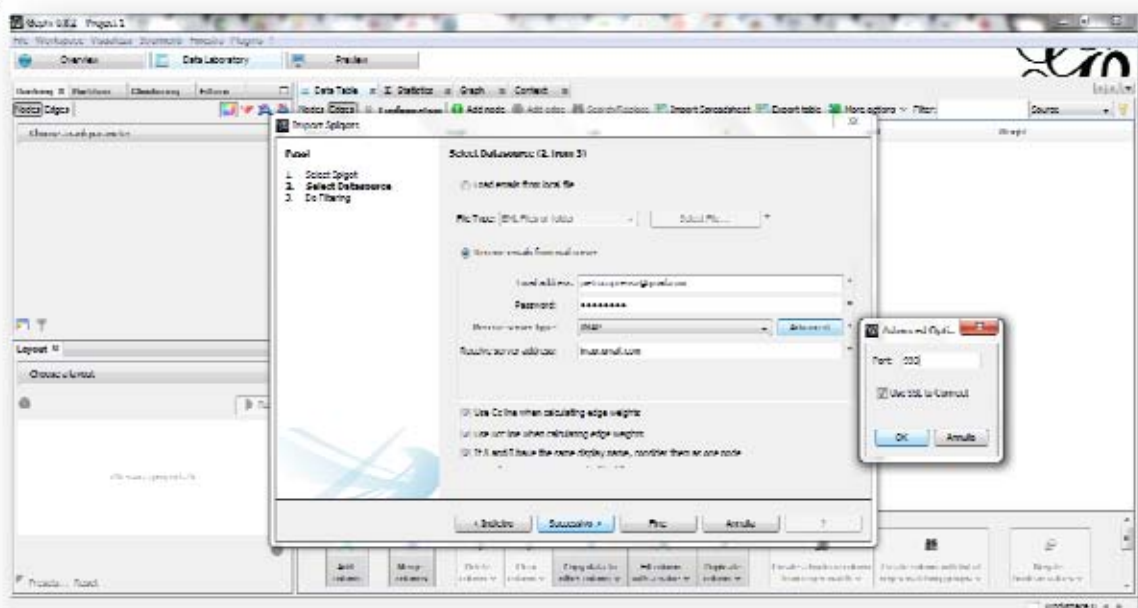


Figura 13.14. Importazione dei propri dati di email, dal server.

Sebbene l'operazione di import dei dati di posta elettronica dal server è molto semplice, considerate che questa può richiedere anche ore (Figura 13.15). Nella mia casella al momento dell'importazione erano presenti 15.841 messaggi e il processo è durato meno di un'ora, con una connessione veloce e un computer più che professionale.

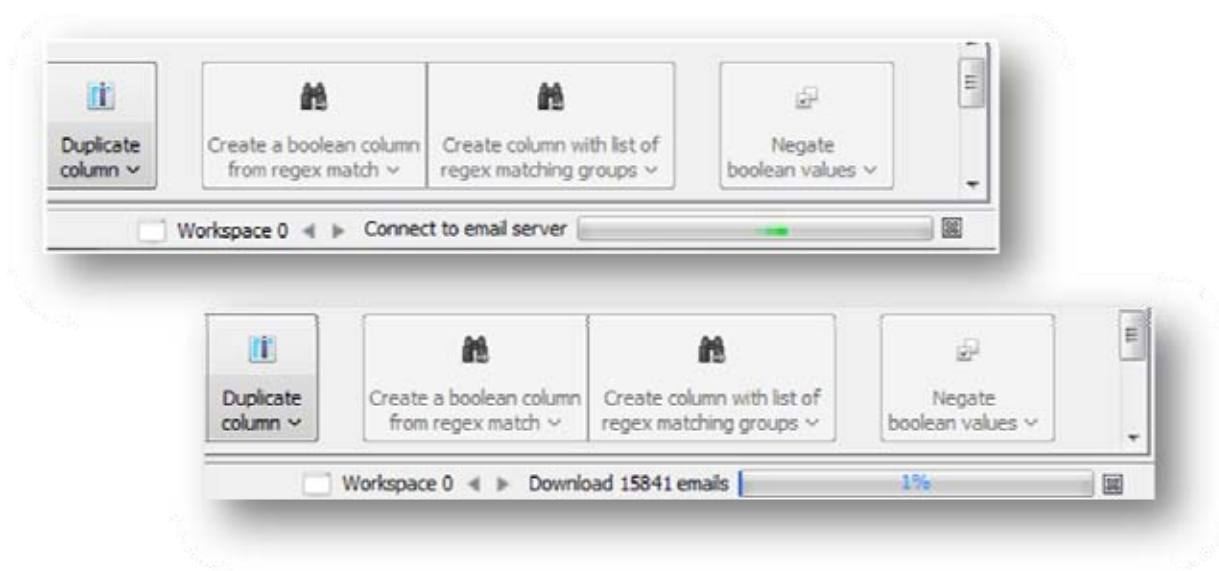


Figura 13.15. L'importazione dei messaggi email può richiedere diverse ore: dipende soprattutto dal numero di messaggi contenuti.

Considerate sempre che i dati scaricati da Gephi sono sempre solo quelli relativi agli individui coinvolti nella comunicazione e mai quelli relativi al contenuto del messaggio. Dunque questo sistema è utile se si vuole esaminare la propria comunicazione via email con gli altri in termini di flussi e non in termini di contenuti dei messaggi, che appunto non vengono neanche scaricati.

In Figura 13.16 è possibile vedere la rappresentazione grafica della mia comunicazione via Gmail. La figura rende subito chiaro che esistono dei cluster di comunicazione, ma valgono anche qui le considerazioni sulla cautela già fatte al quinto capitolo circa le interpretazioni che si possono trarre da questi grafici.

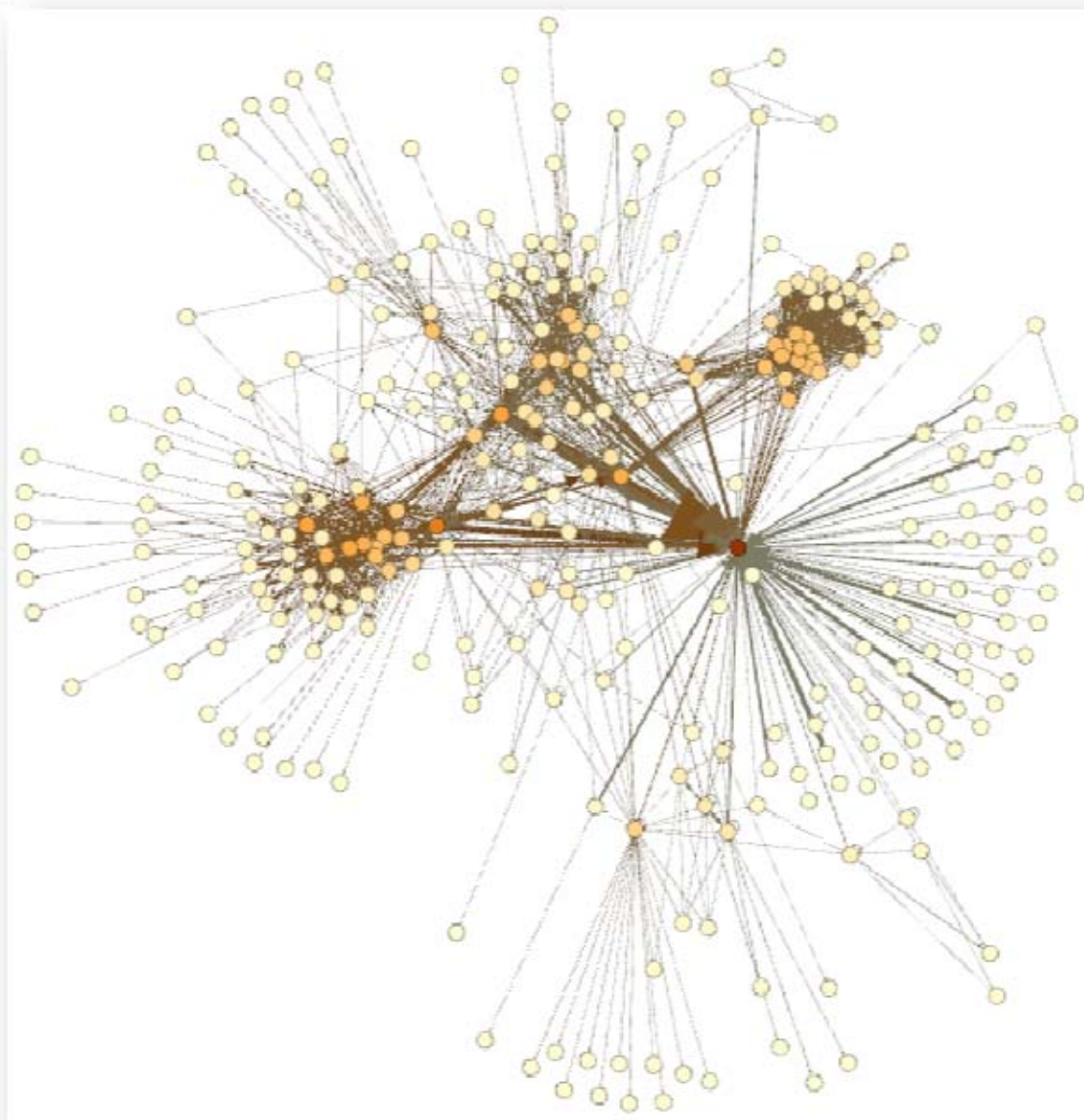


Figura 13.16. Rete relativa alla mia comunicazione via Gmail.

Un aspetto interessante di Gephi e più in generale della social network analysis, è la possibilità di rappresentare graficamente alcuni indici significativi, che invece ci permettono qualche interpretazione in più rispetto al mero grafico lineare.

Nell'esempio in Figura 13.16, ho applicato generato una maggiore grandezza ai nodi che hanno una più alta "Betweenness Centrality", un indice di centralità che affida maggiore valore ai nodi che agiscono da brokers (già spiegati al quinto capitolo).

In pratica dunque un nodo con alta "Betweenness Centrality" potrebbe avere anche molti meno nodi degli altri, ma certamente ha un ruolo di collegamento tra diversi gruppi di nodi e questo lo rende molto importante all'interno della rete raffigurata.

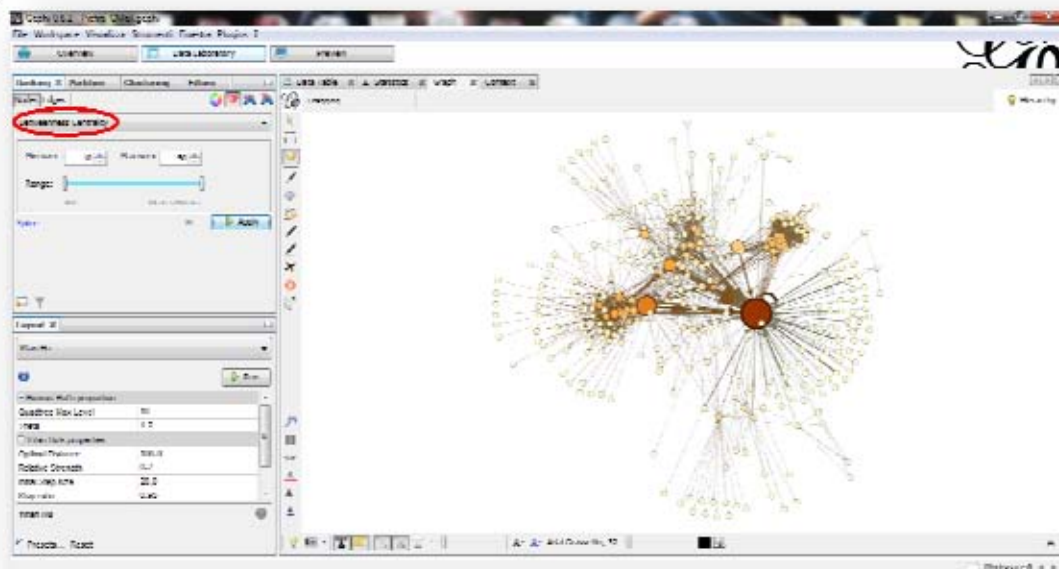


Figura 13.17. Rete relativa alla mia comunicazione via Gmail con maggiore dimensione ai nodi con più alto indice di "Betweenness Centrality", per l'identificazione visiva dei brokers.

Nel quinto capitolo avevamo visto come poter estrarre i propri dati di rete tramite l'applicazione gratuita NameGenWeb. Questi dati possono essere anch'essi importati, infatti l'applicazione permette di scaricare i dati in un formato già compatibile con Gephi. Basterà aprire il file (File > Open) e si potranno già visualizzare i propri contatti e i relativi collegamenti, come da Figura 13.18.

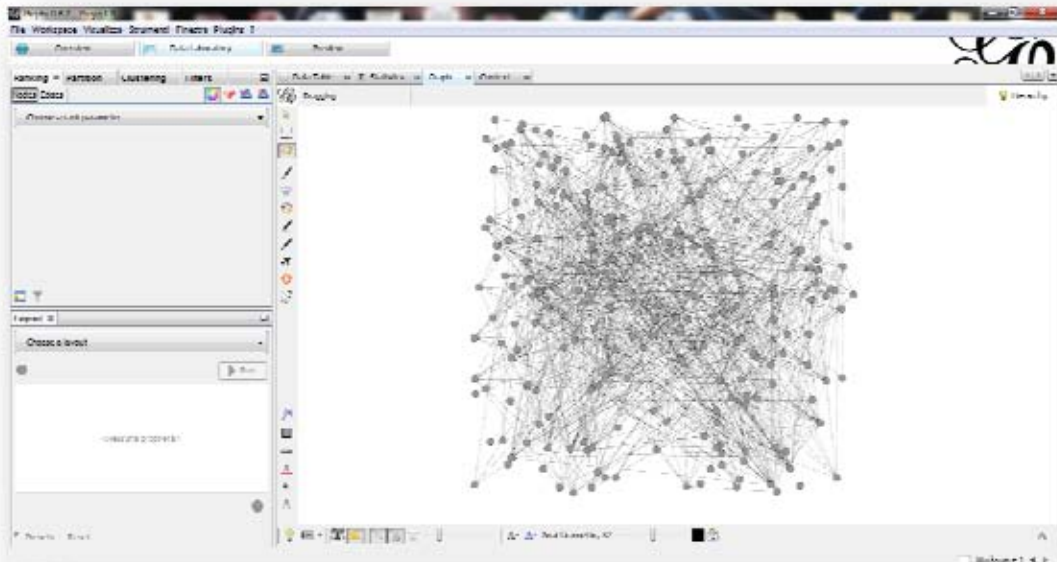


Figura 13.18. Rete relativa ai miei amici Facebook, visualizzati con Gephi e tracciati senza alcuna specifica.

Come si può però notare dalla figura i dati vengono rappresentati insieme ai collegamenti, senza darci alcuna informazione utile, almeno visivamente. Una prima elaborazione utile è applicare un tipo di layout alla rete, in modo da raggruppare i nodi sulla base delle varie connessioni presenti. Uno dei grandi vantaggi di Gephi

è la possibilità di programmare tramite Python e altri linguaggi di programmazione e crearsi anche dei plugin avanzati per la visualizzazione dei dati e per il calcolo di utili indici che non siano già presenti nel programma originale. Un'altra possibilità, come visto anche prima è dare una dimensione ai nodi sulla base della "Betweenness Centrality" e un colore sulla base del genere (rosso per i maschi, blu per le femmine e verde quando non indicato). Queste modifiche permettono agevolmente di identificare, anche visivamente, i gruppi tra gli amici in rete, così come ho fatto io per i miei amici Facebook (Figura 13.19).

Dalla figura risulta anche chiara l'identificazione dei broker nella rete sociale. Restano valide tutte le considerazioni fatte in precedenza circa l'interpretazione dei dati.

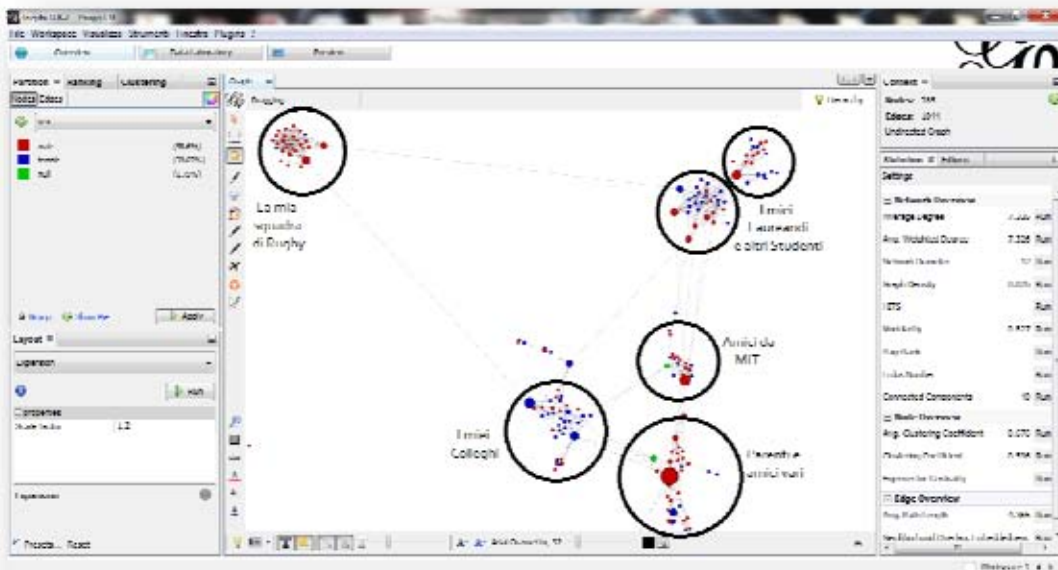


Figura 13.19. Rete relativa ai miei amici Facebook, visualizzati con Gephi e indicando il gruppo di appartenenza per i principali.

Una volta calcolati gli indici statistici e ispezionata visivamente la nostra rete sociale, possiamo estrarre la matrice dei risultati, pronta per essere importata in un programma di analisi statistica che ne consenta un'approfondita elaborazione.

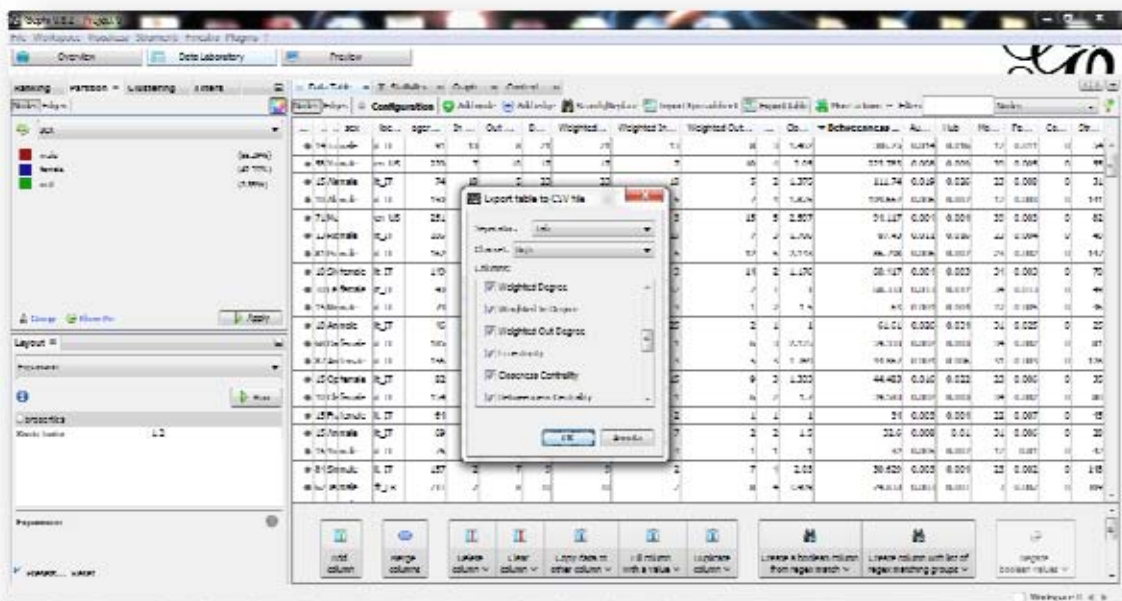


Figura 13.20. Esportazione della matrice dati per la successiva elaborazione statistica.

Un'ultima considerazione è doverosa in merito al salvataggio delle matrici di dati ottenute. Se si sta lavorando su un'unica rete in cui sono presenti molti dati, sarà probabilmente il caso di effettuare analisi con modelli gerarchici per prendere in considerazione i diversi livelli di analisi ed, eventualmente, la dinamica longitudinale. Se invece state lavorando con più matrici (magari

oltre cento) vi converrà curare particolarmente il salvataggio dei dati, in quanto potrebbe essere utile un'ulteriore manipolazione delle matrici di output per mezzo di un programma come Matlab. Sarà dunque utile, come già spiegato sin dai primi capitoli, avere un codice di numerazione (o simili) per ogni matrice generata in output (vedi Figura 13.21), in modo da poterle poi importare con un ciclo "for" unico.

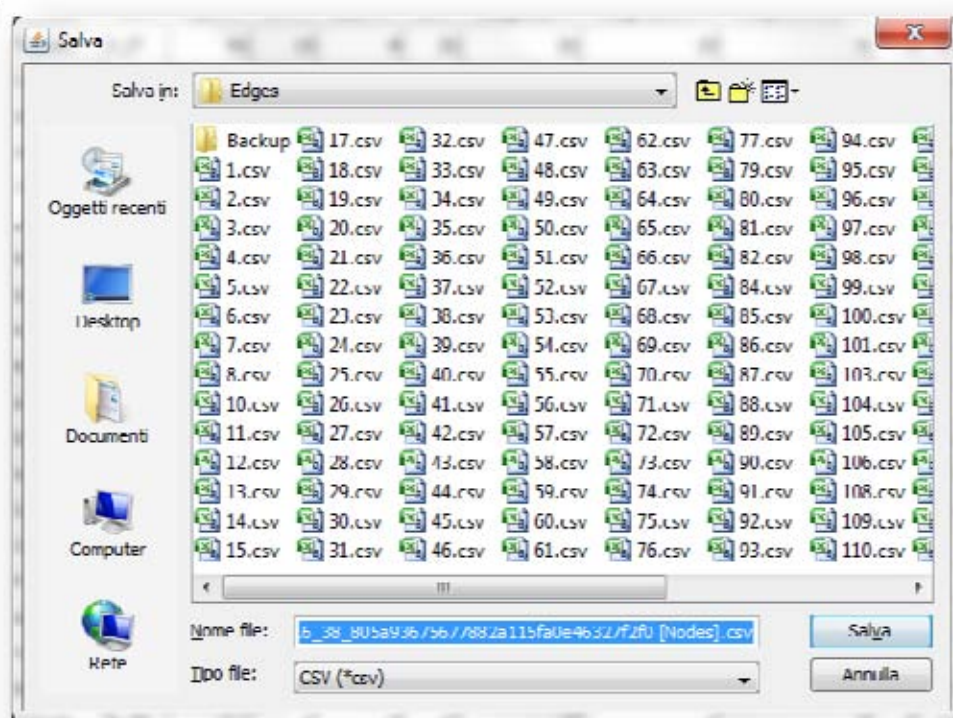


Figura 13.21. Salvataggio della matrice di output, contenente i risultati a livello di nodi o a livello di collegamenti (edges).

In questo capitolo abbiamo visto come la preparazione dei dati per l'elaborazione e la correzione delle matrici per la visualizzazione

di dati relazionali è spesso molto complessa. Tuttavia un uso di programmi professionali per la manipolazione delle matrici dei dati, come Matlab o GNU Octave, combinato a un buon programma di visualizzazione e calcolo degli indici di Social Network Analysis, consente una buona elaborazione ed è certamente un buon inizio per un'efficace interpretazione di questi dati, tanto interessanti quanto complessi.

Computazione dati in realtime

In questa seconda parte del volume abbiamo imparato a gestire dati di una certa complessità e dimensione, tuttavia abbiamo sempre considerato i dati come registrati e importabili in software atti alla loro manipolazione. Chiudiamo questa parte del volume con un breve cenno alla manipolazione dati in realtime, identificando sotto tale dicitura tutti i dati che vengono processati, per diversi motivi, contemporaneamente alla loro acquisizione. Effettivamente riconosco che può sembrare un'esigenza un po' bizzarra o quanto meno un caso tanto raro da non meritare un intero capitolo, anche perché è certamente vero che un segnale elaborato in realtime potrebbe essere meno raffinato rispetto a uno elaborato a posteriori, con tutti i controlli di qualità (ad esempio, rimozione degli artefatti e accurate ispezioni visive). Invece vi sono ambiti in cui questo modo di elaborare dati ha dato spunto a filoni di ricerca del tutto nuovi e innovativi, con applicazioni che vanno dalla clinica all'entertainment. E' dunque, questa, una parte della

psicometria computazionale assai importante e che giorno dopo giorno sta diventando sempre più interessante per il mondo della psicologia.

15.1 Brain Computer Interface

La Brain Computer Interface, anche detta BCI, è una tecnologia per certi versi sorprendente. In particolare, le applicazioni della BCI sono molto affascinanti e rientrano in ambiti che fino a qualche anno fa erano considerati fantascienza. Infatti tramite un sistema BCI, pazienti totalmente paralizzati e che hanno anche perso l'uso della parola, riescono a comunicare. E' una comunicazione lenta, ma è pur sempre comunicazione. Senza entrare troppo in tecnicismi, che esulano dalla trattazione di questo volume, una piattaforma BCI, tramite un sofisticato sincronismo tra l'illuminazione controllata di alcuni stimoli (tipicamente lettere) e le onde corticali prevalentemente in area motoria e/o occipitale, rilevate tramite un elettroencefalogramma (EEG), riesce ad associare l'attenzione rivolta dal paziente verso lo stimolo con il momento in cui un particolare stimolo si è illuminato. Il risultato dunque è che il sistema di BCI riesce a rilevare lo stimolo verso cui il paziente ha rivolto l'attenzione. Non è lettura del pensiero, ci tengo a sottolinearlo chiaramente. Il processo dipende dal fatto che quando il paziente riconosce uno stimolo rilevante tra una serie di diversi stimoli frequenti, automaticamente produce un potenziale

evento correlato (detto anche ERP, event related potential) a distanza di circa 300 millisecondi. Questo specifico ERP viene chiamato P300 e non è l'unico ERP rilevabile tramite un EEG.

In Figura 14.1 possiamo vedere Un'onda che mostra i diversi componenti dell'ERP, incluse la N100 e la P300. Da notare che l'ERP è plottata con voltaggi negativi capovolti (P infatti sta per Positive e N per Negative), una pratica comune, anche se non universale, nella ricerca sugli ERP.

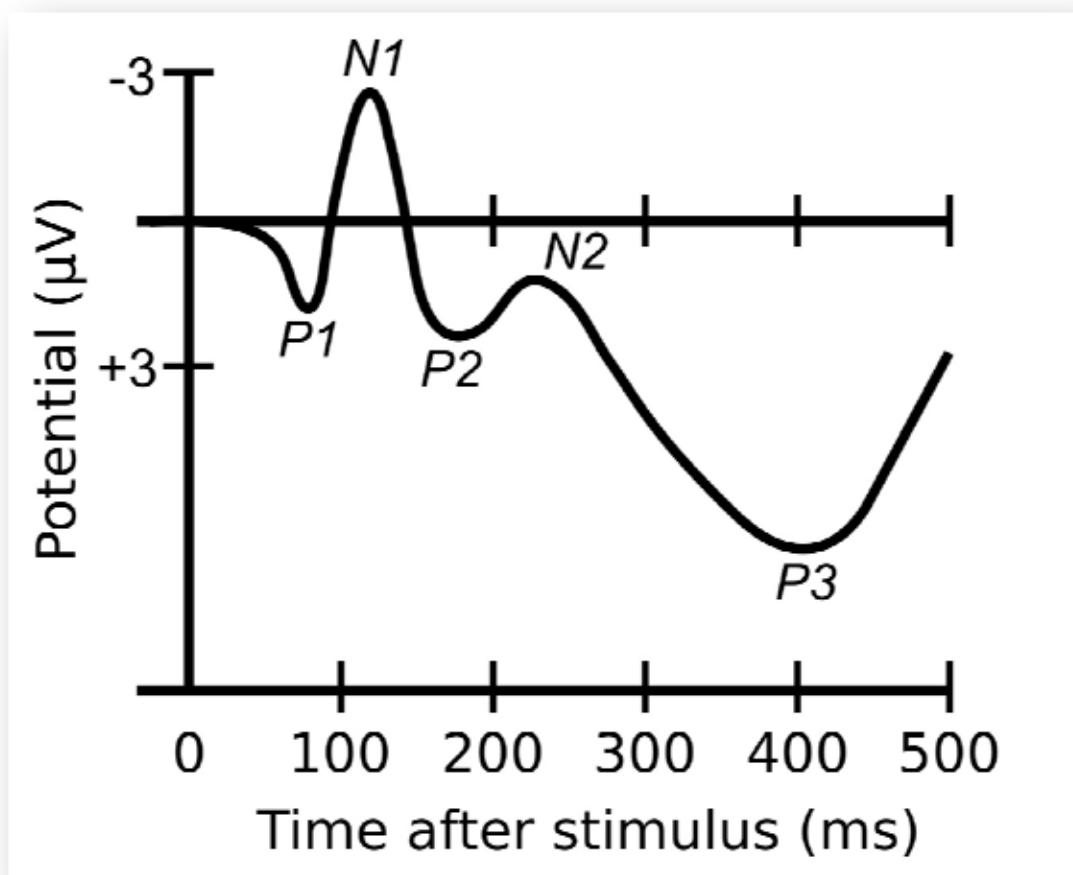


Figura 14.1. Un'onda che mostra i diversi componenti dell'ERP, inclusa la P300.

In Figura 14.2, possiamo vedere un gentile volontario durante un test di controllo per un protocollo di BCI con lettere che si illuminano a blocchi di righe o colonne. La piattaforma interseca in tempo reale le varie informazioni ricevute e scrive la lettera cercata dal soggetto nella parte alta del monitor.



Figura 14.2. Test di controllo per un protocollo di BCI.

La BCI basata su ERP P300 non è l'unica possibile BCI che si può trovare. Tuttavia è certamente tra le più importanti e usate "forme" di BCI in uso nel mondo della psicologia, in particolare nella psicologia clinica. Il sistema descritto è comunemente chiamato di Augmentative and Alternative Communication (AAC,

comunicazione aumentativa e alternativa) e, oltre che in psicologia clinica, è usato per molte applicazioni industriali, militari, di marketing e molto altro. Per i pazienti in cui la paralisi è totale e include anche la paralisi dei movimenti oculari (come per i pazienti in stato avanzato di SLA, sclerosi laterale amiotrofica), la BCI è di fatto l'unico modo per avere una forma di comunicazione con il resto del mondo.

Un'altro tipo di BCI associato a usi differenti è quella basata sulla rilevazione sull'immaginazione motoria (motor imagery based BCI). In pratica il soggetto pensa di chiudere e aprire continuamente la mano sinistra e questo produce un'attivazione corticale nell'area motoria destra rilevata anch'essa tramite EEG. Allo stesso modo viene rilevata l'area opposta. Si "costruiscono" così due comandi (sinistra vs. destra) per muoversi tramite una carrozzina appositamente modificata per comunicare con la piattaforma. Per andare avanti il partecipante immagina di agitare i piedi, attivando un'altra area motoria che fa muovere in avanti la carrozzina. Questo paradigma di BCI è diverso dal precedente basato su ERP P300 e viene maggiormente usato per interagire tramite semplici comandi (avanti, sinistra, destra).

In realtà esistono anche altri tipi di piattaforme BCI, ma l'argomentazione è davvero ampia e in continua evoluzione e lo spazio in questo volume è limitato per poterne discutere più in dettaglio. Ci concentriamo dunque, come sempre, sugli aspetti metodologici pratici, lasciando l'approfondimento per i paradigmi a volumi più dettagliati sui presupposti teorici della trattazione.

In Figura 14.3, possiamo vedere uno schema di funzionamento dei due tipi di BCI descritti.

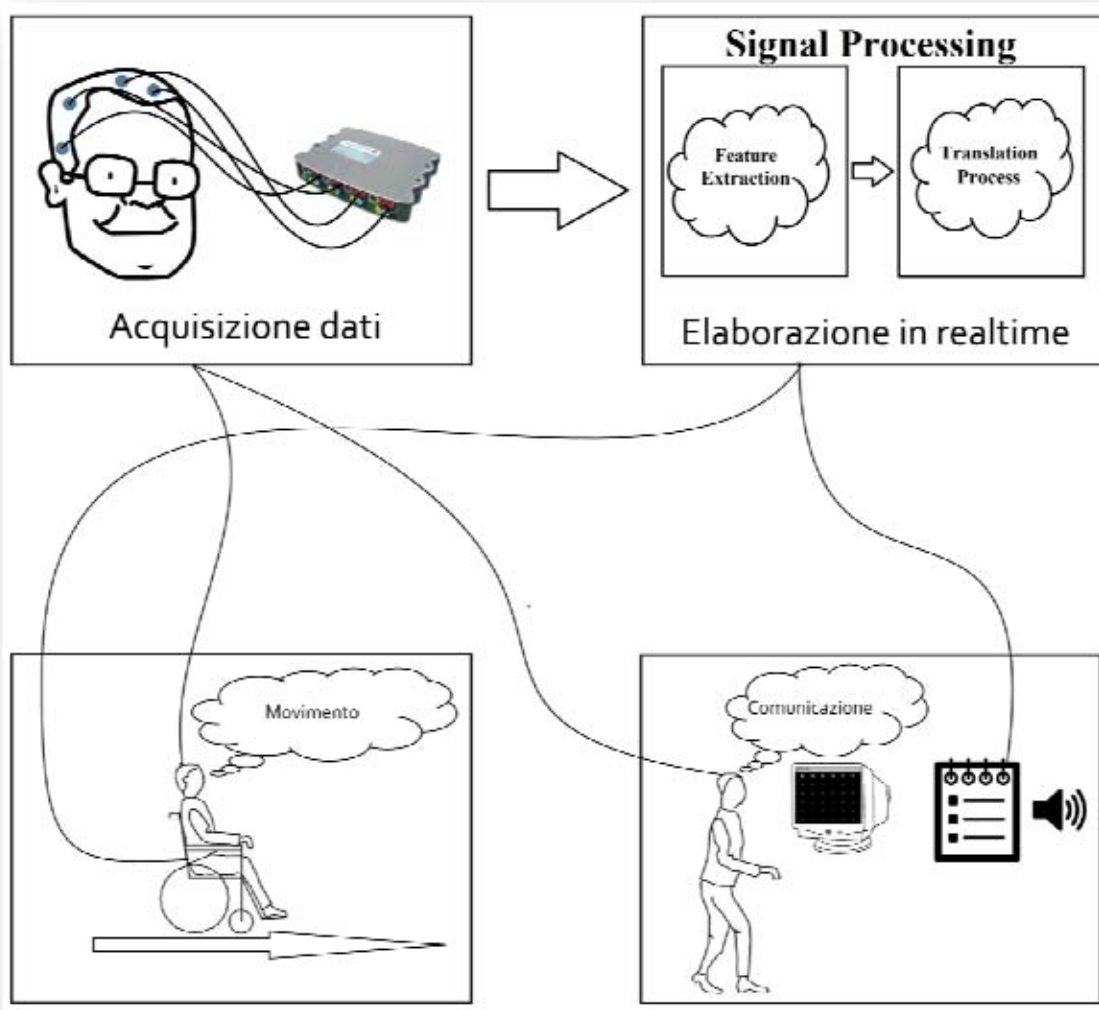


Figura 14.3. BCI per la comunicazione e/o per la motorizzazione controllata tramite la piattaforma.

Esistono diversi schemi di canali EEG che si possono configurare per la BCI basata su P300. Personalmente ho usato uno schema a

sedici canali EEG che ha sempre funzionato molto bene e dunque suggerisco: FZ, C3, C4, CZ, CPZ, P3, P4, PZ, PO3, PO4, POZ, PO7, PO8, O1, O2, OZ (con classico schema internazionale 10-20). Ho sviluppato, una simulazione grafica dettagliata per queste coordinate, che potete visualizzare in Figura 15.4

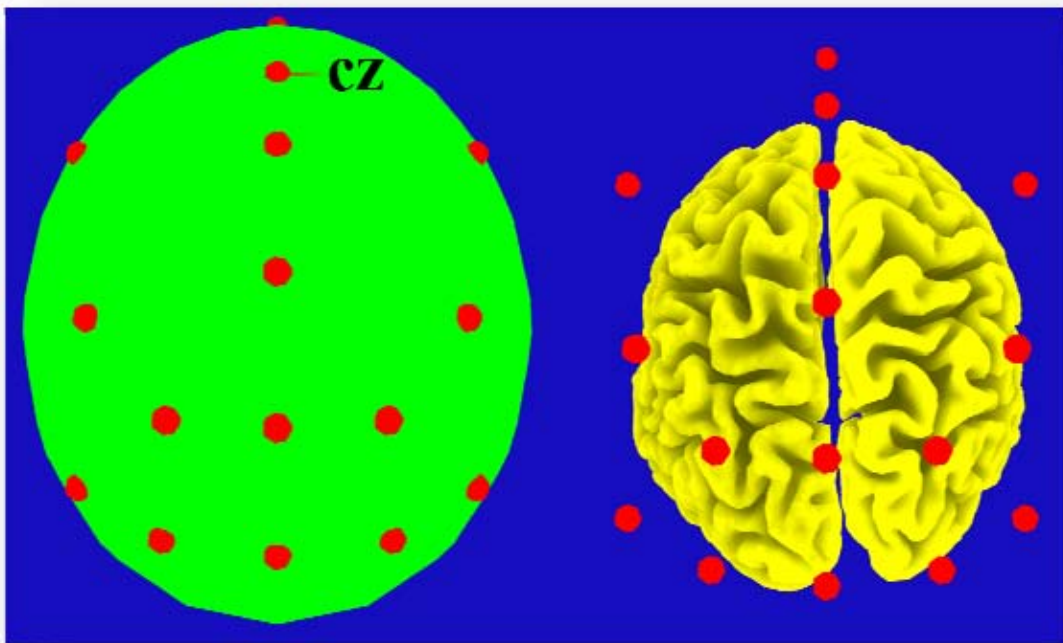


Figura 14.4. Simulazione dell'area corticale e dello scalpo, con canali sovrainposti.

Per chi volesse un maggiore dettaglio, delle possibili aree coinvolte ho usato un'algoritmo inverso per il calcolo delle zone da coordinate di Talairach per lo studio stereotassico. Le immagini relative sono riportate in Figura 14.5. Nel materiale supplementare al volume, trovate anche il file ".xyz" con le coordinate.

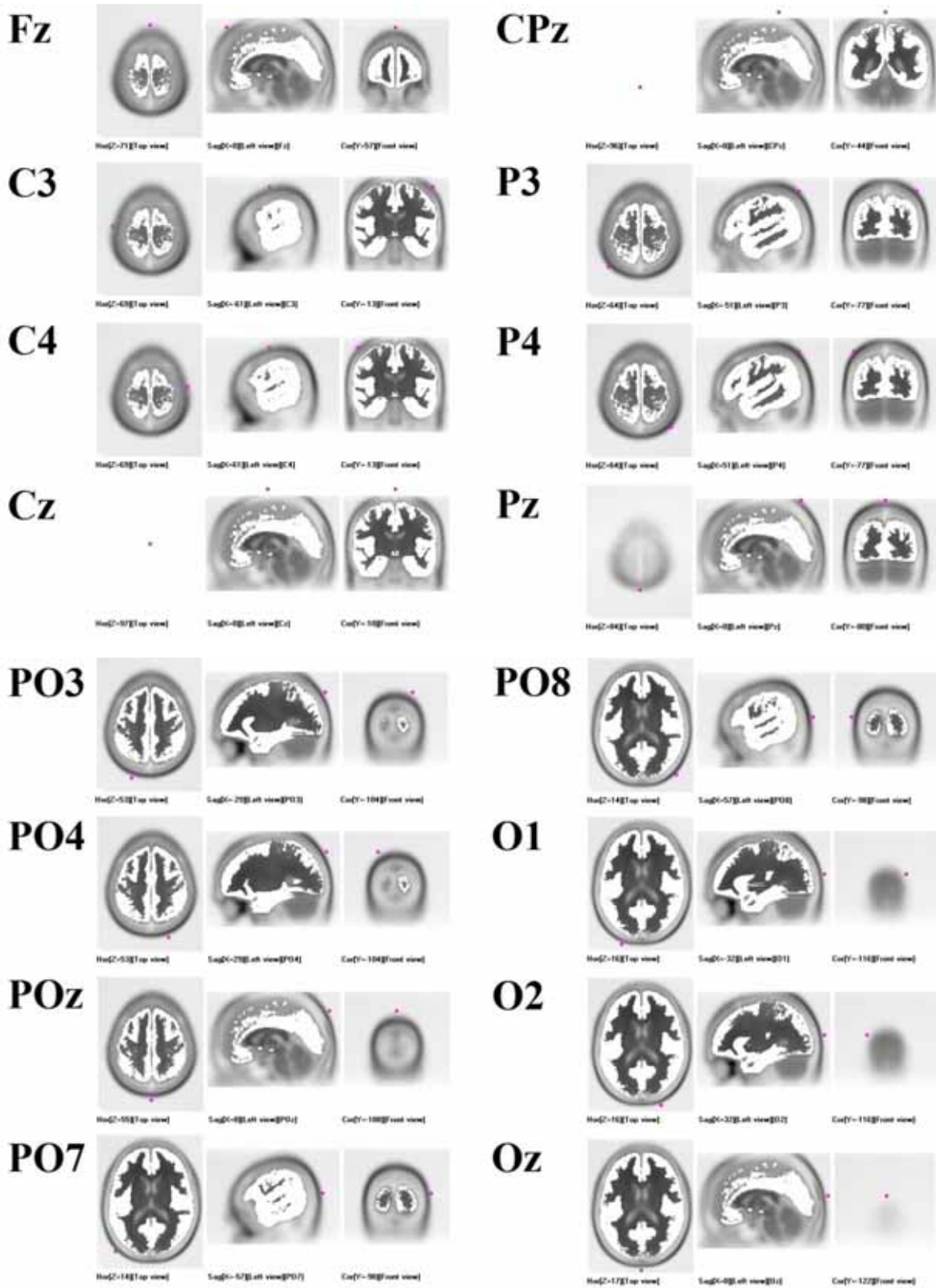


Figura 14.5. Dettaglio specifico delle coordinate di Talairach.

Per chi si volesse dilettere con l'uso della BCI, suggerisco il programma gratuito BCI2000 (<http://www.bci2000.org/>), ormai considerato uno standard nel campo (Figure 14.6 e 14.7).

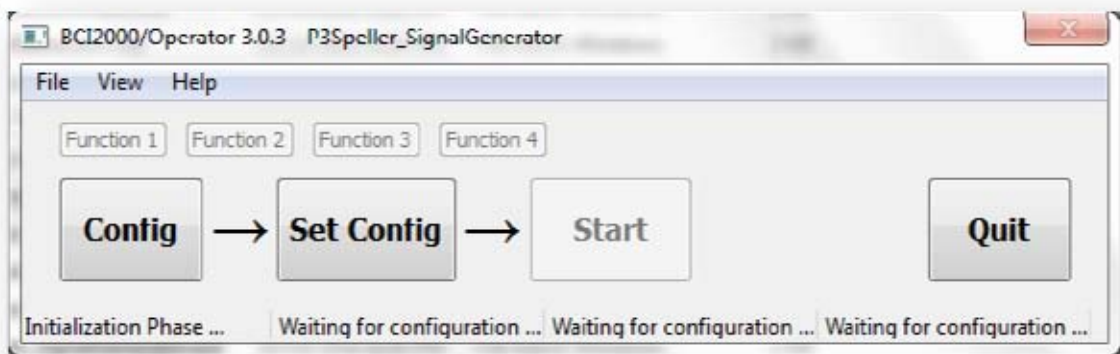


Figura 14.6. Interfaccia iniziale di configurazione e avvio di BCI2000.

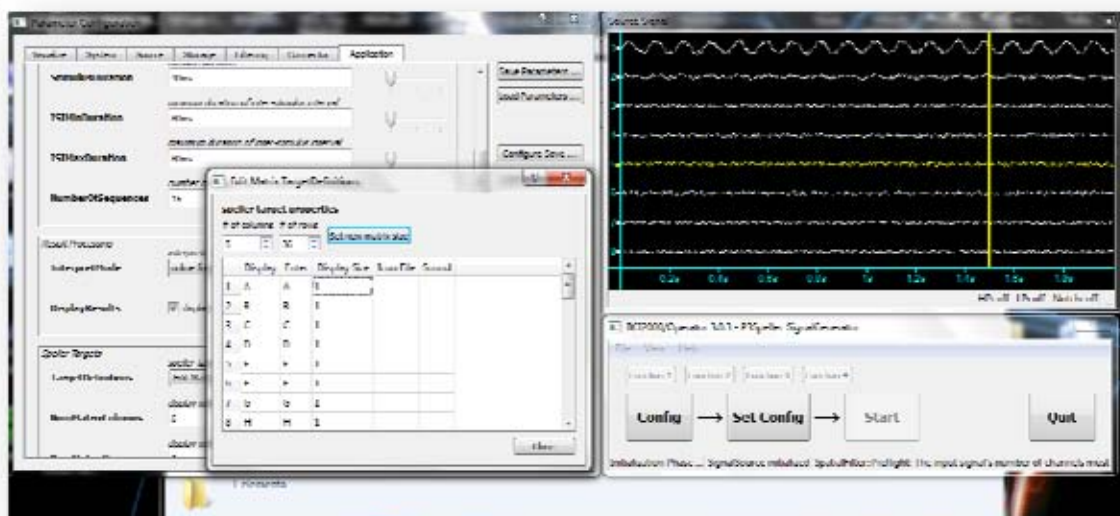


Figura 14.7. Interfaccia e visualizzazione del segnale di BCI2000, con configurazione della matrice di lettere.

14.2 Altre elaborazioni in realtime

Nel precedente paragrafo abbiamo presentato la BCI che per quanto riconosciuta dal mondo scientifico, non è spesso nota al grande pubblico ed è troppo spesso mediaticamente disseminata come lettura del pensiero. Era dunque doveroso fare un po' di chiarezza. Fatto ciò riflettiamo adesso sull'eye-tracker, una tecnologia già esaminata nei precedenti paragrafi, ma vista qui in una modalità differente. In pratica gli eye-tracker più evoluti permettono una programmazione che consente un intervento sui dati in realtime per gestire una modalità d'uso attivo dell'eye-tracker, ovvero una risposta sulla base delle fissazioni. Se dunque voglio attivare determinate risposte a seguito della fissazione di un particolare stimolo o di parte di esso, potrò farlo anche se non è molto semplice e richiede una certa dimestichezza con l'uso dello strumento. Tuttavia, il programma OpenSesame, che abbiamo visto al secondo paragrafo del quarto capitolo del volume, permette un agevole intervento anche tramite programmazione in codice Python. Inoltre, per chi volesse spingersi oltre il gruppo di ricercatori del progetto eBrain (tra cui il sottoscritto) ha sviluppato il software eBrain Engine (<http://ebrainengine.codeplex.com/>), gratuito e open source, per la gestione contemporanea di piattaforme BCI e Eye-trackers, che può essere usato sia per l'AAC, sia per effettuare una batteria neuropsicologica includente vari test da essere effettuati con Eye-tracker e/o BCI, per la valutazione di varie funzioni cognitive.

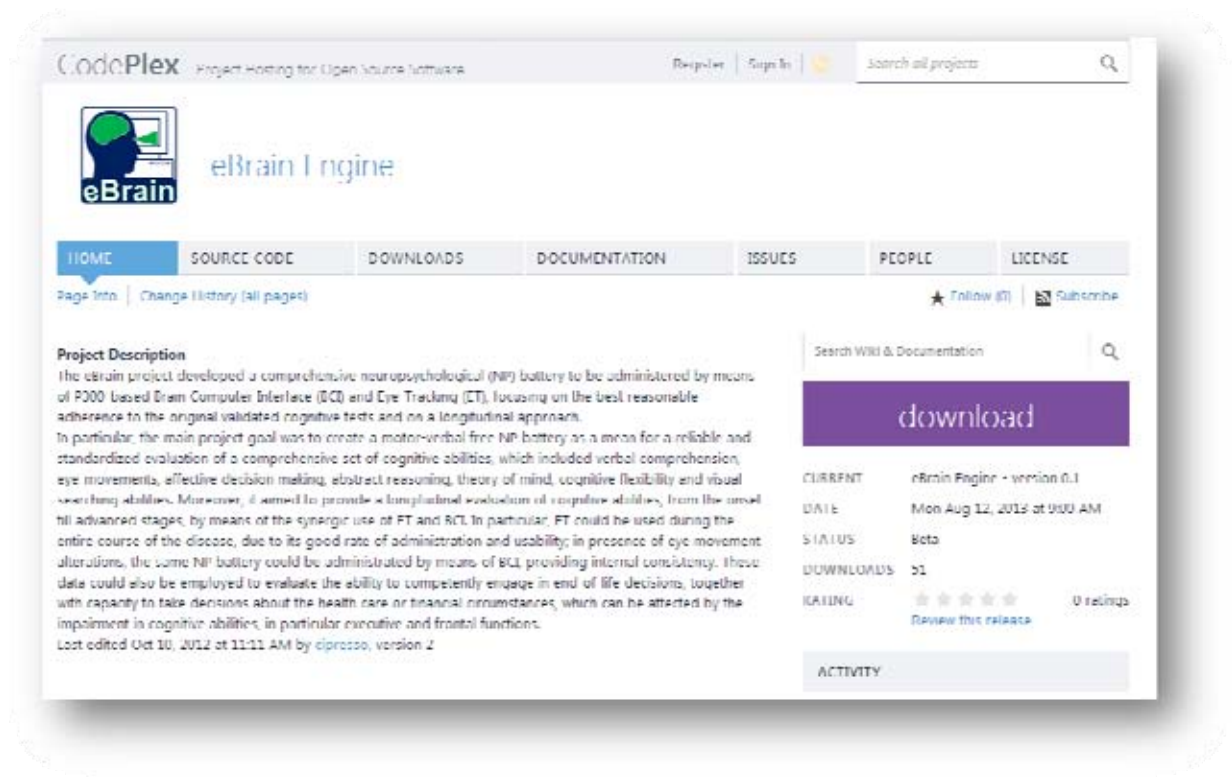
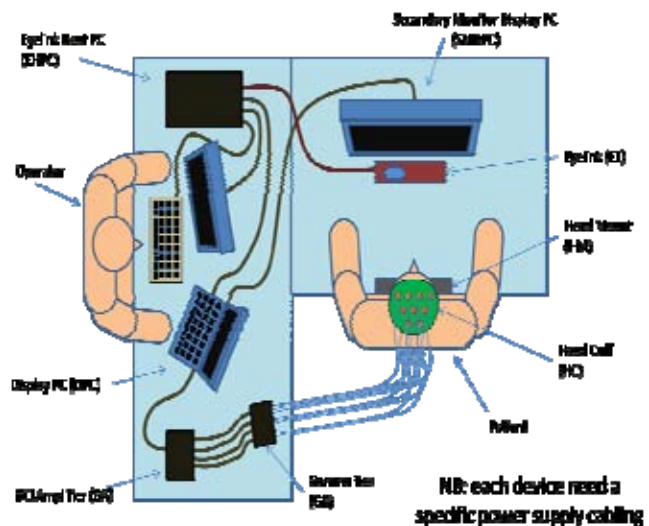


Figura 14.8. Piattaforma eBrain Engine per BCI e Eye-trackers.

L'uso attivo dell'eye-tracker permette interessanti modalità di interazione, in quanto il risultato che si ottiene è continuo, ovvero si può anche creare un'interazione continua tra lo stimolo e l'utente che può, ad esempio, trascinare degli stimoli e sostanzialmente fare più o meno quello che fa con altre interfacce come (il mouse).



Tramite l'uso attivo dell'eye-tracker, l'utente può anche controllare abbastanza rapidamente gli stimoli. In particolare se si configura una tastiera, sarà possibile scrivere frasi in pochi secondi, con tempi considerevolmente ridotti (Figura 14.9).

Inoltre l'uso combinato proposto tramite eBrain Engine, apre nuovi fronti di ricerca clinica e sperimentale.

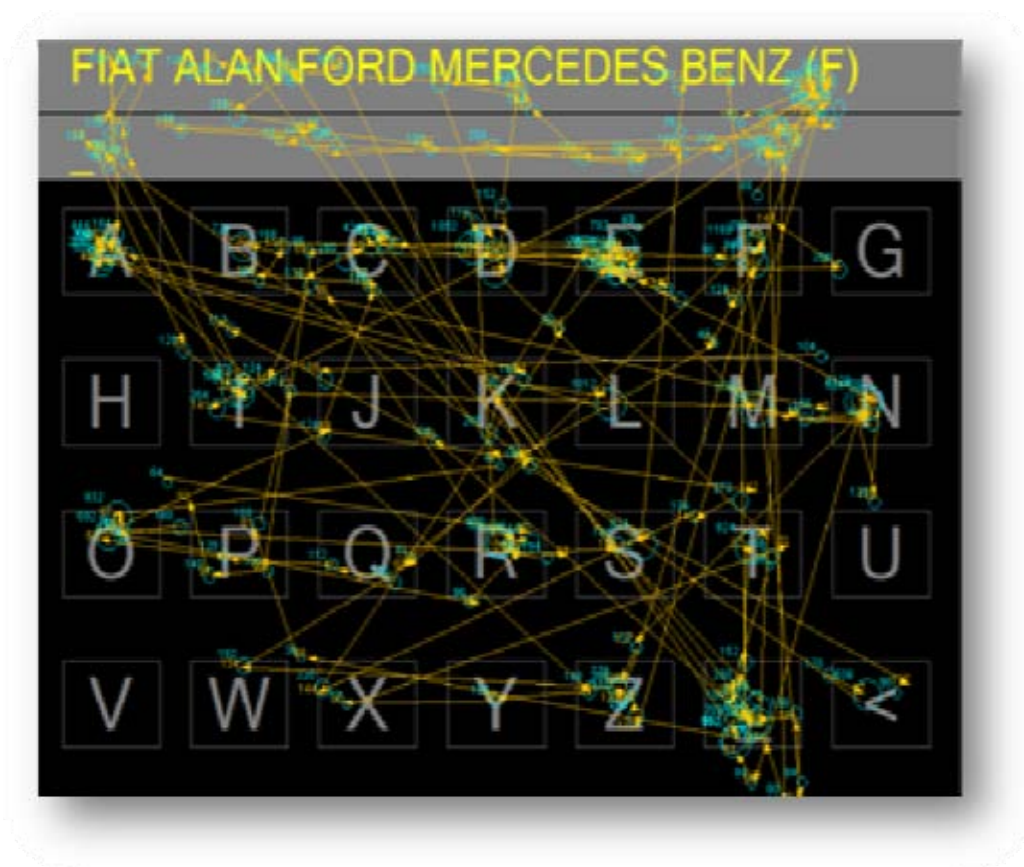


Figura 14.9. Tracciati dei movimenti oculari sugli stimoli (lettere) per la scrittura di una serie di parole (in alto).

Dopo aver visto anche l'uso combinato di eye-tracker e BCI, esaminiamo adesso quali altre tecnologie elaborano in realtime.

In realtà l'elenco dei possibili strumenti che permettono acquisizione ed elaborazione di dati in realtime è molto ampio, sia nell'ambito d'uso della psicologia, sia in altri ambiti.

Per quanto attiene le tematiche maggiormente legate alla psicomетria computazionale, vi sono altri tre filoni molto importanti e in buona parte anche legati tra di loro:

- 1) il Biofeedback;
- 2) l'Affective Computing;
- 3) e il monitoraggio continuo.

Il biofeedback è una tecnica usata soprattutto a scopi clinici e per lo sport. In pratica per fare il biofeedback un soggetto indossa dei biosensori per la rilevazione di parametri psicofisiologici periferici o centrali (tramite l'EEG nel qual caso è più comunemente detto neurofeedback). Questi sensori trasmettono il segnale a un computer che lo riceve, lo elabora in tempo reale, ne estrae gli opportuni indici e lega la variazione di questi indici a degli oggetti da fare visualizzare al soggetto stesso. Dunque chi fa una sessione di biofeedback ottiene una visualizzazione in tempo reale dei propri segnali psicofisiologici, imparando in modo implicito a gestirli opportunamente per migliorare il benessere.

Di recente una App per iPad, Positive Technology App (<http://positiveapp.info/>), sviluppata da un gruppo di italiani (tra cui il sottoscritto) nel contesto del progetto Interstress (<http://interstress.eu/>) finanziato dalla Commissione Europea, permette di effettuare il biofeedback tramite l'uso di un sensore

commerciale con Bluetooth Smart, in un ambiente virtuale dentro il quale l'utente gestisce il flusso di una cascata tramite il proprio battito cardiaco, come si può vedere in Figura 14.10.



Figura 14.10. Biofeedback in realtà virtuale tramite un App per dispositivi mobili con sistema operativo iOS.

I dati raccolti dalla Positive Technology App, vengono automaticamente inviati a un database MySQL per l'elaborazione computazionale e per la successiva analisi statistica. Inoltre i dati

vengono resi disponibili in modo sicuro e protetto all'utente che si registra su un'apposita piattaforma web per poter visualizzare solo ed esclusivamente le proprie informazioni, tracciando e monitorando in tal modo l'andamento relativo al proprio stress.

L'Affective Computing è un altro campo di ricerca molto attivo in psicologia. In pratica, i ricercatori stanno cercando di ottimizzare e automatizzare il processo di riconoscimento delle emozioni (più precisamente degli stati affettivi) di un qualsiasi individuo e contemporaneamente di creare della tecnologia capace di esprimere emozioni verso gli individui. L'Affective Computing ha obiettivi molto ambiziosi, tuttavia la tecnologia oggi disponibile offre sempre più possibilità di ricerca e sperimentazione e gli algoritmi di classificazione diventano ogni giorno più efficaci. Tra le grandi sfide per l'Affective Computing è certamente da considerarsi la gestione dei Big Data per una più corretta classificazione degli stati affettivi.

Legato al biofeedback e all'Affective Computing è certamente il macrotema del monitoraggio continuo. Oggi esistono moltissimi sensori indossabili e ciascuno di noi ha sempre (o quasi) con sé uno smartphone pronto a collezionare moltissimi dati, come visto nella prima parte di questo volume. Il monitoraggio continuo è oggi una realtà sia in telemedicina, sia in altri ambiti e soprattutto in psicologia. La sfida dei Big Data, anche per il monitoraggio continuo, sarà un'opportunità per sintetizzare in modo coerente una simile mole di dati.

Abbiamo già parlato, alla fine del capitolo 4 della raccolta di dati comportamentali in contesti ecologici usando Purple Robot (https://play.google.com/store/apps/details?id=edu.northwestern.cb.its.purple_robot_manager). Tuttavia questa applicazione, per quanto innovativa e potente, non prevede la connessione con biosensori e, più in generale, l'acquisizione di dati psicofisiologici. Alla luce di quanto abbiamo visto nei precedenti capitoli, un monitoraggio dello stato individuale è certamente più accurato se possiamo legare dati psicologici, dati comportamentali, e dati psicofisiologici, rendendo questa raccolta dati contestuale alle situazioni di vita quotidiana. Con questa ambizione nasce il software Psychlog (<http://sourceforge.net/projects/psychlog/>), sviluppata da un team di ricercatori italiani del CNR di Pisa, dell'Istituto Auxologico Italiano e dell'Università Cattolica di Milano. Lo sviluppo di questa App, a cui ho personalmente partecipato, è avvenuto nell'ambito del progetto Interstress (<http://interstress.eu/>), finanziato dalla Commissione Europea.

Psychlog è un App gratuita e open source nata per funzionare su smartphone Windows mobile 6.5. L'App si collega a sensori ECG della Shimmer (<http://www.shimmersensing.com/>) e registra sia il dato grezzo dell'elettrocardiogramma, sia il relativo indice R-R, di cui abbiamo già parlato al capitolo 2. Ma Psychlog fa molto di più: essendo un'App nata per soddisfare le esigenze degli psicologi, Psychlog usa la metodologia dell'Experience Sampling Method (ESM, metodo di campionamento dell'esperienza) per interrogare il partecipante con appositi questionari (personalizzabili) ad

intervalli di tempo randomizzati (o predefiniti) e acquisendo contemporaneamente i dati psicofisiologici e di contesto per un periodo di tempo precedente, contemporaneo e successivo alla richiesta del questionario. Psychlog permette dunque approfondite e accurate indagini sul rapporto tra la sfera psicologica e i relativi correlati psicofisiologici legati alle risposte date.

La piattaforma è stata testata scientificamente da Gaggioli e colleghi (Gaggioli, 2013) con una sperimentazione schematicamente descritta in Figura 14.11.

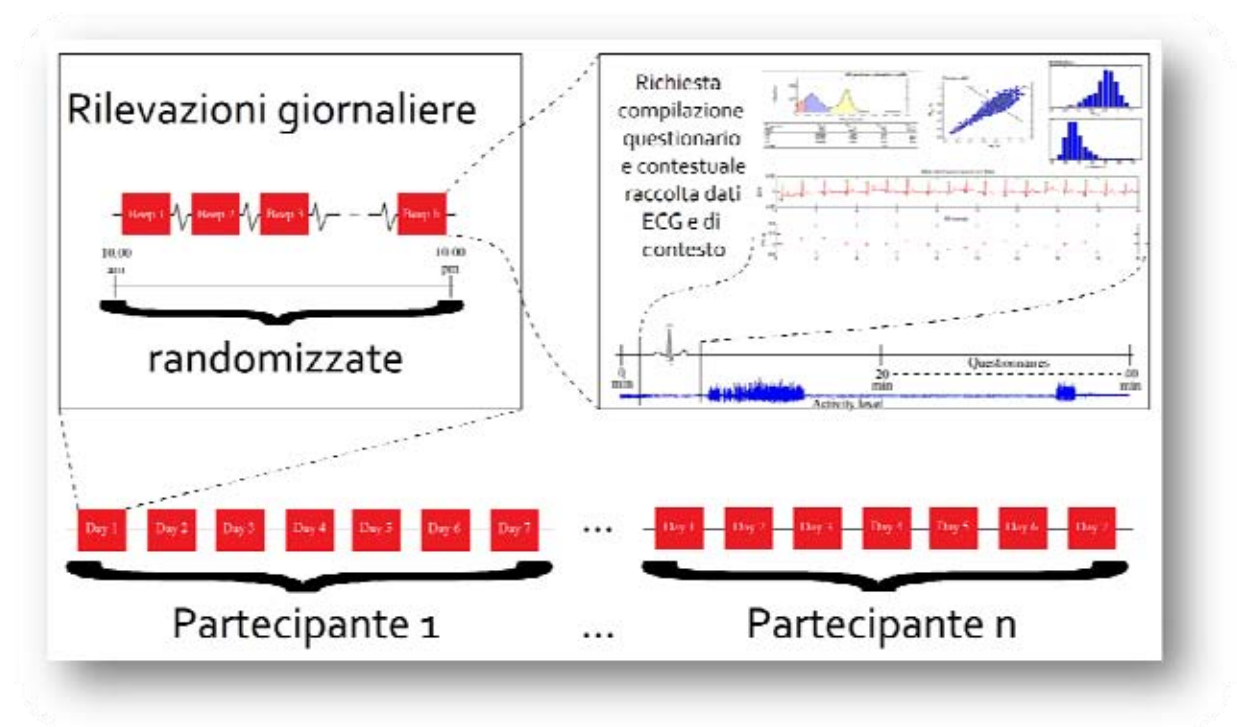


Figura 14.11. Procedura di test della piattaforma Psychlog.

Come si può vedere in Figura, i dati dell'accelerometro, possono essere usati per prendere il segnale ECG solo in situazioni di stasi

del soggetto, evitando errori legati al movimento eccessivo o alle situazioni contestuali (come quando il dato è rilevato durante una corsa).

Grazie al codice aperto i ricercatori possono personalizzare questionari mediante la configurazione di un file "xml", inserendo le domande più adatte alla propria sperimentazione.

I dati raccolti con Psychlog possono essere importati in Matlab per l'elaborazione avanzata del segnale cardiaco. Tuttavia il codice per l'importazione dei dati, per quanto semplice, è più complesso di quello visto nei capitoli precedenti e vi tolgo dunque l'imbarazzo indicandolo di seguito e rendendolo ovviamente disponibile nei materiali supplementari al volume.

```
Sub Import1()
[FileName,PathName] = uigetfile('*.csv','Scegli il file .csv','c:\');
if FileName==0
    return
end
clear questionario_1 questionario_2 questionario_3 questionario_4 questionario_5
questionario_6 questionario_7 questionario_8 questionario_9 questionario_10;
pack;
filecsv = strcat(PathName,FileName);
fid = fopen(filecsv);
num_of_signals = 8;
counter = 1;
posarray = zeros(11,1);
linecounter = 0;
if(fid > 0)
    % Faccio un primo giro per cercare le posizioni di inizio dei
    % questionari
    while (~feof(fid))
        clear C;
        linecounter = linecounter + 1;
        line = fgetl(fid);
```

```

if (strmatch('Questionario', line))
    C = sscanf( line, 'Questionario %d');
    posarray(counter) = linecounter;
    counter = counter +1;
end
if feof(fid)
    posarray(counter) = linecounter;
end
%C = textscan(fid,'Questionario %d','delimiter','\n'); temp_string =
%cell2mat(C{2});
%= ftell(fid);
end
for i=1:counter-1
    switch i
        case 1
            questionario_1 = zeros(num_of_signals, posarray(2) - posarray(1));
        case 2
            questionario_2 = zeros(num_of_signals, posarray(3) - posarray(2));
        case 3
            questionario_3 = zeros(num_of_signals, posarray(4) - posarray(3));
        case 4
            questionario_4 = zeros(num_of_signals, posarray(5) - posarray(4));
        case 5
            questionario_5 = zeros(num_of_signals, posarray(6) - posarray(5));
        case 6
            questionario_6 = zeros(num_of_signals, posarray(7) - posarray(6));
        case 7
            questionario_7 = zeros(num_of_signals, posarray(8) - posarray(7));
        case 8
            questionario_8 = zeros(num_of_signals, posarray(9) - posarray(8));
        case 9
            questionario_9 = zeros(num_of_signals, posarray(10) - posarray(9));
        case 10
            questionario_10 = zeros(num_of_signals, posarray(11) - posarray(10));
    end
end
tempcounter = 0;
line = "";
%mi metto all'inizio del file
fseek(fid, 0, 'bof');
while (~feof(fid))
    %primo passo, cerco il termine Questionario

```



```
linecounter = linecounter + 1;
line = fgetl(fid);
if (strmatch('Questionario', line))
    tempcounter = tempcounter + 1;
    arraylinecounter = 1;
else
    line = regexprep(line, ',', '!');
    %line = regexprep(line, ';', '!');
    a = sscanf(line, '%f;%f;%f;%f;%f;%f;%f;%f');
    switch tempcounter
        case 1
            questionario_1(:, arraylinecounter) = a;
        case 2
            questionario_2(:, arraylinecounter) = a;
        case 3
            questionario_3(:, arraylinecounter) = a;
        case 4
            questionario_4(:, arraylinecounter) = a;
        case 5
            questionario_5(:, arraylinecounter) = a;
    end
    arraylinecounter = arraylinecounter + 1;
end
end
end
```


Cenni di Signal Processing

Nello studio dei comportamenti umani è utile capire i meccanismi alla base del comportamento ma è ancor più utile analizzare a fondo la sfera decisionale degli individui, comunque ampiamente considerando le dinamiche alla base della decisione.

Per comprendere i fenomeni umani è dunque necessario uno studio integrato che definisca un percorso completo che va dalle basi emotive dell'agire umano al comportamento, per sua natura osservabile e misurabile (ovviamente entro certi limiti), passando per le analisi di correlati psicofisiologici (misurabili tramite elettroencefalogramma, elettrocardiogramma, pletismografia, conduttanza cutanea, respirazione, ...) e per un approfondito studio del rapporto tra questi indicatori e le decisioni prese dall'individuo.

L'analisi dei correlati psicofisiologici richiede strumenti tecnici di una certa complessità e competenze specifiche nel campo della

teoria dei segnali, per poter interpretare correttamente i risultati delle analisi.

In questa ottica mi sembra importante dedicare un capitolo atto a dare qualche cenno di signal processing e in particolare agli aspetti stocastici di base della teoria dei segnali.

Lo scopo di questo lavoro è quello di approfondire l'analisi matematico-statistica di un segnale, che può servire, ad esempio, per estrarre informazioni dai segnali psicofisiologici di un campione di soggetti. Dunque ci concentreremo sugli aspetti di estrazione delle informazioni da questi segnali piuttosto che sulla mera descrizione dei segnali e dei dispositivi per l'acquisizione già visti nei capitoli precedenti (elettroencefalogramma, elettrocardiogramma, elettromiografo, eye-tracker, rilevatore di conduttanza cutanea, temperatura, ecc.).

15.1 Processi stocastici e principali proprietà

Lo strumento che utilizziamo per far fronte all'esigenza di trovare una metafora probabilistica per i segnali raccolti è il processo stocastico.

I processi stocastici sono una naturale generalizzazione di una variabile aleatoria. Così come, per definire una variabile aleatoria è necessario associare ad ogni evento elementare un numero reale, per definire un processo stocastico si deve associare ad ogni

evento elementare una funzione reale nel dominio del tempo, che descrive un'evoluzione temporale dell'esperimento casuale.

Più formalmente, associamo ad ogni evento semplice $\omega \in \Omega$ una funzione numerica di $t \in T$, $\{X(t, \omega); t \in T\}$. Se t è una variabile indicante il tempo e T è l'insieme reale, allora $\{X(t, \omega); t \in T\}$ è una funzione del tempo. Per t fissato e ω variabile in Ω , $X(t, \omega)$ rappresenta invece una variabile aleatoria. Se entrambi t e ω sono variabili, allora la famiglia

$$\{X(t, \omega); t \in T, \omega \in \Omega\}$$

è detta processo stocastico.

Così scritto il processo stocastico evidenzia la dipendenza sia da t sia da ω . In effetti, si possono dare due differenti interpretazioni di un processo stocastico. Ovvero per $t \in T$ variabile e $\omega = \omega_0$ fissato, $X(t, \omega_0)$ è una singola funzione di t detta realizzazione del processo stocastico. Per $t = t_0$ fissato e $\omega \in \Omega$ variabile, $X(t_0, \omega)$ è una funzione misurabile dell'insieme $\omega \in \Omega$, quindi è una variabile casuale. Nel primo caso avremo una famiglia di funzioni temporali, $X(t, \cdot)$ i cui membri sono ottenuti al variare di $\omega \in \Omega$. Ogni membro di tale famiglia è detto realizzazione del processo stocastico. Nel secondo caso si avrà una famiglia di variabili aleatorie, $X(\cdot, \omega)$ i cui membri sono ottenuti al variare di $t \in T$. Nel seguito verrà adottata la seconda interpretazione, salvo che in taluni casi che verranno debitamente esplicitati.

In particolare prenderemo in considerazione i processi continui a parametro discreto, in tal caso T è discreto mentre Ω è uno spazio di probabilità continuo per cui $X(t_0, \omega)$ definisce una variabile casuale continua per qualsiasi $t_0 \in T$. I segnali campionati spesso costituiscono una realizzazione di tali processi.

Resta adesso da definire come può essere caratterizzato un processo stocastico. Se $X(t)$ è una successione di variabili casuali, essa è nota se sono note tutte le funzioni di ripartizione semplici $F(x_1; t_1)$ per ogni x_1 reale e per qualunque $t_1 \in T$; inoltre, occorre conoscere tutte le funzioni di ripartizione doppie $F(x_1, x_2; t_1, t_2)$ e in generale occorre conoscere tutte le funzioni di ripartizione n-uple

$$F_n(x_1, x_2, \dots, x_n; t_1, t_2, \dots, t_n) = P\{X(t_1) \leq x_1, X(t_2) \leq x_2, \dots, X(t_n) \leq x_n\}$$

per qualsiasi n-upla di istanti temporali $t_1, t_2, \dots, t_n \in T$ e per ogni $n = 1, 2, 3, \dots$. Solo in questo caso, infatti, sarà nota la probabilità di qualunque evento riguardante $X(t)$. In effetti, dal teorema di Kolmogorov sappiamo che devono essere soddisfatte le due seguenti condizioni dette di simmetria e di compatibilità

a) $F_n(x_{i_1}, x_{i_2}, \dots, x_{i_n}; t_{i_1}, t_{i_2}, \dots, t_{i_n}) = F_n(x_1, x_2, \dots, x_n; t_1, t_2, \dots, t_n)$

essendo (i_1, i_2, \dots, i_n) una permutazione di $(1, 2, \dots, n)$;

b) $F_n(x_1, x_2, \dots, x_m, +\infty, \dots, +\infty; t_1, t_2, \dots, t_m, t_{m+1}, \dots, t_n) = F_m(x_1, x_2, \dots, x_m; t_1, t_2, \dots, t_m)$.

La condizione di simmetria esprime l'invarianza della funzione di ripartizione rispetto a ogni permutazione contemporanea di x_i e t_i .

La condizione di compatibilità esprime invece il fatto che quando

si passa dalla distribuzione F_n a quella F_m con $m < n$ considerando il limite a $+\infty$ delle variabili irrilevanti, quest'ultima funzione non deve dipendere dai parametri t_{m+1}, \dots, t_n .

Mediante la famiglia delle funzioni di ripartizioni possiamo, pertanto, caratterizzare un processo stocastico in modo soddisfacente. L'unico inconveniente sta nel fatto che, praticamente sempre, queste famiglie risultano analiticamente parecchio complicate. Per tal motivo si ricorre spesso a caratterizzazioni sintetiche dei processi stocastici. Vediamone alcune.

Chiameremo valore atteso di un processo stocastico $X(t)$, la funzione ordinaria di t intorno alla quale oscillano le possibili realizzazioni del processo, i.e.

$$m(t) = E[X(t)] = \int_{-\infty}^{+\infty} x dF_1(x; t)$$

Chiameremo, invece, varianza la misura della dispersione delle varie realizzazioni intorno alla funzione valore atteso, i.e.

$$\sigma^2(t) = V[X(t)] = \int_{-\infty}^{+\infty} [x - m(t)]^2 dF_1(x; t)$$

Parliamo adesso della covarianza, per mettere in evidenza la dipendenza che esiste tra $X(t_1)$ e $X(t_2)$ per ogni coppia di t_1 e t_2 , ovvero

$$\begin{aligned}
\text{Cov}[X(t_1), X(t_2)] &= E[X(t_1) \cdot X(t_2)] - m(t_1) \cdot m(t_2) = \\
&= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [x_1 - m(t_1)][x_2 - m(t_2)] d x_1 x_2 F_2(x_1, x_2; t_1, t_2) = \\
&= K(t_1, t_2)
\end{aligned}$$

La funzione $K(t_1, t_2)$ viene a volte chiamata autocovarianza ma noi in seguito la chiameremo sempre funzione di covarianza.

Visto che ne farò uso in seguito definiamo l'operazione di convoluzione (anche detto integrale di convoluzione, se ne capirà tra breve il motivo) tra due processi stocastici $X(t)$ e $Y(t)$. Il processo $Z(t)$ ricavato dalla loro convoluzione, e indicato con il simbolo $*$, è definito nel modo seguente

$$Z(t) = X(t) * Y(t) = \int_{-\infty}^{+\infty} X(\tau) Y(t - \tau) d\tau$$

Se $X(t) = Y(t) = 0$ per $t < 0$ l'espressione precedente diventa

$$Z(t) = X(t) * Y(t) = \int_0^t X(\tau) Y(t - \tau) d\tau$$

L'integrale di convoluzione gode di numerose proprietà, ricaviamone adesso in dettaglio le principali.

Dato il processo $Z(t)$ ricavato dalla convoluzione di $X(t)$ e $Y(t)$ come sopra visto, ponendo $t - \tau = h$, si avrà

$$\begin{aligned}
 Z(t) = X(t) * Y(t) &= \int_{-\infty}^{+\infty} X(\tau)Y(t-\tau)d\tau = \int_{-\infty}^{+\infty} X(t-h)Y(h)dh = \\
 &= \int_{-\infty}^{+\infty} Y(h)X(t-h)dh = Y(t) * X(t)
 \end{aligned}$$

Pertanto l'integrale di convoluzione gode della proprietà commutativa. Proviamo adesso a vedere se gode anche della proprietà associativa.

Dati i tre processi stocastici $X(t)$, $Y(t)$ e $Z(t)$, poniamo

$$\begin{aligned}
 R(t) &= X(t) * Y(t) \\
 S(t) &= Y(t) * Z(t)
 \end{aligned}$$

Affinché valga la proprietà associativa, dovrà quindi essere

$$R(t) * Z(t) = X(t) * S(t)$$

inoltre per la convoluzione sappiamo che

$$R(t) * Z(t) = \int_{-\infty}^{+\infty} R(\tau)Z(t-\tau)d\tau$$

da cui, sostituendo

$$R(t) = \int_{-\infty}^{+\infty} X(h)Y(t-h)dh$$

e ponendo $j = \tau - h$ e scambiando l'ordine di integrazione, si ha

$$R(t) * Z(t) = \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} X(h)Y(\tau - h)dh \right] Z(t - \tau)d\tau = \int_{-\infty}^{+\infty} X(h) \left[\int_{-\infty}^{+\infty} Y(j)Z(t - h - j)dj \right] dh$$

L'espressione tra parentesi, in questa espressione, è chiaramente $S(t - h)$ e pertanto possiamo scrivere

$$R(t) * Z(t) = \int_{-\infty}^{+\infty} X(h)S(t - h)dh = X(t) * S(t)$$

come volevasi dimostrare.

Vediamo adesso la convoluzione in tempo. Dati due processi stocastici $X(t)$ e $Y(t)$, se

$$F(\omega) = f[X(t)] \text{ e } G(\omega) = f[Y(t)]$$

Allora si ha

$$f[X(t) * Y(t)] = F(\omega) \cdot G(\omega)$$

La dimostrazione di questa affermazione deriva direttamente ricordando la proprietà di traslazione in tempo della trasformata di Fourier.

Adesso approfondiamo invece la convoluzione in frequenza. Dati due processi stocastici $X(t)$ e $Y(t)$, se

$$X(t) = f^{-1}[F(\omega)] \text{ e } Y(t) = f^{-1}[G(\omega)]$$

Allora si ha

$$f^{-1}[F(\omega) * G(\omega)] = 2\pi \cdot X(t) \cdot Y(t)$$

oppure

$$f[X(t) \cdot Y(t)] = \frac{1}{2\pi} [F(\omega) * G(\omega)] = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) G(\omega - s) ds$$

Questa affermazione indica che la convoluzione in frequenza corrisponde alla moltiplicazione in tempo.

Quando abbiamo introdotto i processi stocastici, abbiamo detto che, per caratterizzare un generico processo stocastico è necessario conoscere la funzione di distribuzione congiunta $F_n(x_1, x_2, \dots, x_n; t_1, t_2, \dots, t_n)$ dove abbiamo n variabili casuali estratte dal processo a diversi istanti ed $n \geq 1$ arbitrario.

Una classe molto importante di processi stocastici è quella dei processi stazionari che, da un punto di vista intuitivo, rappresentano quei processi invarianti ad un cambiamento dell'istante di riferimento. Formalmente un processo stocastico di

dice stazionario in senso stretto quando, fissati n istanti di tempo t_1, t_2, \dots, t_n , risulta

$$F_n(x_1, x_2, \dots, x_n; t_1, t_2, \dots, t_n) = F_n(x_1, x_2, \dots, x_n; t_1 + \tau, t_2 + \tau, \dots, t_n + \tau)$$

per ogni $n \geq 1$, $t_1, t_2, \dots, t_n \in T$ e $t_1 + \tau, t_2 + \tau, \dots, t_n + \tau \in T$.

In altri termini, quando un processo stocastico è stazionario in senso stretto le caratteristiche distribuzionali di tutte le marginali rimangono costanti al passare del tempo.

Questa relazione vale per qualsiasi valore di n . Proviamo, allora, a prendere qualche valore particolare di n e vediamo cosa implica.

Iniziamo prendendo $n=1$: in questo caso la relazione diventa

$$F_1(x_1; t_1) = P\{X(t_1) \leq x_1\} = F_1(x_1; t_1 + \tau) = P\{X(t_1 + \tau) \leq x_1\}$$

Questa relazione dice in pratica che, dato un processo stocastico stazionario in senso stretto, dati due istanti qualsiasi di tempo e date le corrispondenti variabili aleatorie estratte dal processo, tali due variabili sono identiche dal punto di vista statistico. Essendo la distribuzione di $X(t)$ indipendente da t , abbiamo

$$\begin{aligned} E[X(t)] &= m(t) = m \\ V[X(t)] &= \sigma^2(t) = \sigma^2 \end{aligned}$$

Passiamo adesso ad $n=2$: in questo caso la relazione diventa

$$\begin{aligned} F_2(x_1, x_2; t_1, t_2) &= P\{X(t_1) \leq x_1, X(t_2) \leq x_2\} = \\ &= F_2(x_1, x_2; t_1 + \tau, t_2 + \tau) = P\{X(t_1 + \tau) \leq x_1, X(t_2 + \tau) \leq x_2\} \end{aligned}$$

Se ad esempio prendiamo $\tau = -t_1$, questa diventa

$$\begin{aligned} F_2(x_1, x_2; t_1, t_2) &= P\{X(t_1) \leq x_1, X(t_2) \leq x_2\} = \\ &= F_2(x_1, x_2; 0, t_2 - t_1) = P\{X(0) \leq x_1, X(t_2 - t_1) \leq x_2\} \end{aligned}$$

Questa relazione indica che la distribuzione congiunta delle variabili casuali $X(t_1)$ e $X(t_2)$ non dipende da t_1 e t_2 separatamente ma solo dalla loro differenza $t_2 - t_1$.

Per approfondire ulteriormente il concetto, valutiamo quanto vale la funzione di covarianza $K(t_1, t_2)$ che dipende da $\tau = |t_2 - t_1|$ e non da t_1 e t_2 isolatamente. Si avrà

$$\begin{aligned} K(t_1, t_1 + \tau) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x_1 - m)(x_2 - m) d x_1 x_2 F_2(x_1, x_2; t_1, t_1 + \tau) = \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x_1 - m)(x_2 - m) d x_1 x_2 F_2(x_1, x_2; 0, \tau) = \\ &= K(0, \tau) = K(\tau) \end{aligned}$$

Dunque l'esistenza della condizione per la stazionarietà stretta implica le condizioni su valore atteso e varianza, che abbiamo ottenuto per $N=1$, ed inoltre la condizione sulla covarianza appena vista, ottenuta per $N=2$. In generale non vale il viceversa.

Le implicazioni appena viste sono importanti in quanto definiscono un processo stocastico in senso lato, meglio detto a covarianza stazionaria. La definizione non impone alcuna restrizione per

quanto riguarda il valore atteso, il quale pertanto potrebbe essere rappresentato da una qualunque funzione ordinaria del tempo. La covarianza dipende solo da τ , ossia il lag che separa le due variabili e non da t . Per queste ragioni usiamo questa accezione nel definire la stazionarietà in senso lato.

Riepilogando, affinché il processo sia a covarianza stazionaria devono verificarsi le seguenti condizioni

- Il valore medio del processo stocastico $E[X(t)]$ è indipendente dall'istante di tempo considerato;
- La funzione di covarianza delle due variabili aleatorie estratte, $X(t_1)$ e $X(t_2)$ dipende soltanto dalla differenza degli istanti di tempo $\tau = |t_2 - t_1|$ e non dai due istanti di tempo indipendentemente.

Appare ovvio a questo punto che un processo stocastico stazionario in senso stretto è anche un processo stocastico a covarianza stazionaria. In generale, invece non vale l'implicazione inversa. E' anche sbagliato dire che i processi stocastici a covarianza stazionaria includono strettamente l'insieme dei processi stocastici stazionari in senso stretto. Infatti, una definizione non implica strettamente l'altra; ad esempio un processo stocastico può essere stazionario in senso stretto ma non possedere momenti; viceversa, la costanza nel tempo dei momenti non implica che le varie marginali abbiano la stessa distribuzione. In un caso, tuttavia, le due definizioni coincidono: questo caso, che è particolarmente importante per le applicazioni pratiche, è

quello in cui il processo è gaussiano, ossia quando la distribuzione congiunta di un qualunque sottoinsieme di elementi del processo è una normale multivariata. Se un processo è gaussiano, stabilire che è a covarianza stazionaria equivale a stabilire la stazionarietà in senso stretto. Data la prevalenza dei processi gaussiani nelle applicazioni ai dati, da un punto di vista operativo si adotta generalmente la definizione di stazionarietà in covarianza, e quando si parla di stazionarietà senza aggettivi, è appunto a questa che ci si riferisce.

Il più semplice processo stocastico è il processo White Noise (rumore bianco). Esso è definito da

$$\begin{aligned} E(X_t) &= 0 \\ \text{Var}(X_t) &= E(X_t)^2 = \sigma^2 < +\infty \\ \text{Cov}(X_t, X_{t-k}) &= E(X_t X_{t-k}) = \gamma(k) = 0 \quad \text{per } |k| > 0 \end{aligned}$$

Un white noise è quindi, in sostanza, un processo composto di un numero infinito di variabili aleatorie a media zero e varianza costante; queste variabili aleatorie, inoltre, sono tutte non correlate l'una all'altra.

Anticipiamo adesso un risultato, che sarà ripreso, dicendo che se un processo stocastico è stazionario, allora la funzione di covarianza $k(\tau)$, che dipende soltanto dal tempo, è dotata di trasformata di Fourier definita come densità spettrale del processo. Abbiamo fatto ricorso, in precedenza, a caratterizzazioni sintetiche dei processi stocastici, come valore medio e funzione di

covarianza. Nella realtà che ci apprestiamo a studiare questi sono ottimi strumenti ma, purtroppo, non sempre disponibili. Vanno pertanto stimati attraverso i dati osservati del segnale, cioè sulla base di una realizzazione del processo considerato. Nei classici problemi di stima si ha un certo numero di osservazioni sulla stessa variabile, noi invece abbiamo alle varie epoche di osservazione t_1, t_2, \dots, t_N un'unica osservazione su ogni variabile $X(t_i)$ con $i=1, 2, \dots, N$. Spesso non è possibile replicare l'esperimento per ottenere più repliche della realizzazione. Ma ciò non deve far pensare che le stime su di un'unica realizzazione non siano consistenti o a basso contenuto informativo. Infatti, se il processo stocastico che genera il segnale è ergodico, allora si possono dedurre stime consistenti anche con una sola realizzazione. L'ergodicità, quindi, ci permette di inferire da una successione di dati, su alcune funzioni-momento del processo. Essa è definita con riguardo a una funzione. Si parla, pertanto, di ergodicità rispetto alla media, alla varianza, all'autocorrelazione, alla funzione di covarianza e così via. Poiché il nostro interesse è rivolto solo alla media e alla funzione di covarianza, ci occuperemo solo dell'ergodicità rispetto a questi due parametri.

In generale un processo stocastico si dice ergodico se osservandolo per un lasso di tempo abbastanza lungo, è possibile osservare quasi tutte le sottosequenze che il processo è in grado di generare. Il fatto che, in un sistema ergodico, se qualcosa può succedere allora prima o poi deve succedere fa sì che eventi molto lontani fra loro possono essere considerati virtualmente indipendenti.

Di conseguenza, se un processo è ergodico, è possibile (almeno in linea di principio) usare le informazioni contenute nel suo svolgimento nel tempo per inferirne le caratteristiche. Esiste un teorema (detto appunto teorema ergodico) che dice che, se un processo è ergodico, l'osservazione di una sua realizzazione abbastanza lunga è equivalente, ai fini inferenziali, all'osservazione di un gran numero di realizzazioni.

Una successione aleatoria si dice ergodica rispetto alla media se, definita la successione di variabili

$$\bar{X}_n = \frac{1}{N} \sum_{t=0}^{N-1} X(t) \quad E[X(t)] = m \quad \text{per ogni } t,$$

essa converge in media quadratica verso la costante m , cioè se e solo se

$$\lim_{N \rightarrow +\infty} E \left[\left| \bar{X}_n - m \right|^2 \right] = 0$$

Si dimostra che condizione necessaria e sufficiente affinché la successione stazionaria $\{X(t); t \in N\}$, $N = 0, 1, \dots$ sia ergodica rispetto alla media, è che la successione $\text{Re}[K(\tau)]$ converga a zero nel senso di Césaro, ossia che

$$\lim_{N \rightarrow +\infty} \frac{1}{N} \cdot \sum_{\tau=0}^N \text{Re}[K(\tau)] = 0$$

dove $\operatorname{Re}[K(\tau)]$ indica la parte reale della funzione di covarianza $K(\tau)$.

Questa condizione non risulta sempre facilmente verificabile e pertanto è utile la ricerca di una condizione solo sufficiente che assicuri l'ergodicità. Ricordando che, se una successione converge nel senso ordinario allora converge allo stesso valore anche nel senso di Césaro, scriveremo

$$\lim_{t \rightarrow +\infty} \operatorname{Re}[K(t)] = 0$$

Da questa discussione si capisce chiaramente che non tutti i processi stazionari sono ergodici rispetto alla media. Del resto si possono avere processi ergodici senza che essi siano stazionari.

Per l'ergodicità rispetto alla funzione di covarianza analizziamo la variabile aleatoria

$$C_N(\tau) = \frac{1}{N - \tau_0} \sum_{t=0}^{N-\tau_0-1} Z(t)$$

con $Z(t) = [(X(t) - m)\overline{(X(t + \tau_0) - m)}]$ successione stazionaria. Indichiamo adesso con $K_Z(\tau)$ la funzione di covarianza della successione $Z(t)$.

Condizione necessaria e sufficiente affinché la successione $X(t)$ sia ergodica rispetto alla funzione di covarianza è che

$$\lim_{N \rightarrow +\infty} \frac{1}{N - \tau_0} \sum_{t=0}^{N-\tau_0-1} K_Z(t) = 0$$

E' chiaro che vi sono processi ergodici rispetto alla media ma non rispetto alla covarianza. Comunque se il processo è gaussiano e l'ergodicità in media deriva dalla condizione

$$\lim_{\tau \rightarrow \infty} K(\tau) = 0$$

allora esso è ergodico anche rispetto alla funzione di covarianza, oltre che stazionario.

Una funzione $k(\tau)$ definita per $\tau = 0, \pm 1, \pm 2, \dots$ è la funzione di covarianza di un processo stocastico se e solo se essa può rappresentarsi come

$$k(\tau) = \int_{-\pi}^{+\pi} e^{i\lambda\tau} dF(\lambda)$$

dove $F(\lambda)$ è una funzione non decrescente definita in $|\lambda| \leq \pi$ con variazione totale data da $F(\pi) - F(-\pi) = k(0)$.

La funzione $F(\lambda)$ è detta funzione spettrale. Assumeremo che la funzione $F(\lambda)$ sia differenziabile e affinché ciò sia basta che

$$\sum_{\tau=-\infty}^{+\infty} |k(\tau)| < \infty$$

Posto, dunque, che esista (finita) la derivata di $F(\lambda)$, diremo $F'(\lambda) = f(\lambda)$ densità spettrale del processo. Scriveremo, pertanto,

$$(15.1) \quad k(\tau) = \int_{-\pi}^{+\pi} e^{i\lambda\tau} f(\lambda) d\lambda \quad \tau = 0, \pm 1, \pm 2, \dots$$

D'altra parte sappiamo che

$$(15.2) \quad f(\lambda) = \frac{1}{2\pi} \sum_{\tau=-\infty}^{+\infty} e^{-i\lambda\tau} k(\tau) \quad (-\pi < \lambda \leq \pi)$$

Si nota dunque che per una successione stazionaria, la densità spettrale, $f(\lambda)$, e la funzione di covarianza, $k(\tau)$, sono le trasformate di Fourier una dell'altra.

Qualche autore in luogo di $1/2\pi$ nella (15.2) mette $1/\pi$ considerando così solo le frequenze positive, cioè da 0 a π invece che da $-\pi$ a π . Non è errato, ma $1/2\pi$ è più diffuso, quindi mi adeguo a questo standard.

Nel caso di successioni stazionarie reali, la densità spettrale potrà essere scritta come

$$f(\lambda) = \frac{1}{2\pi} \left[\sum_{\tau=-\infty}^1 k(\tau) \cos \lambda\tau + k(0) \cos 0\lambda + \sum_{\tau=1}^{+\infty} k(\tau) \cos \lambda\tau \right]$$

quindi, essendo $k(-\tau) = k(\tau)$, avremo

$$(15.3) \quad f(\lambda) = \frac{1}{2\pi} k(0) + \frac{1}{\pi} \sum_{\tau=1}^{+\infty} k(\tau) \cos \lambda \tau$$

che è la trasformata coseno di Fourier della funzione di covarianza della successione stazionaria reale. Si ricorda che $e^{-i\lambda\tau} = \cos \lambda\tau - i \sin \lambda\tau$, ed è proprio per questo che parlando di successioni stazionarie reali la trasformata di Fourier (esponenziale) diventa trasformata coseno di Fourier. Mentre si avrà funzione spettrale data da

$$F(\lambda) = \frac{1}{2\pi} k(0)(\lambda + \pi) + \frac{1}{\pi} \sum_{\tau=1}^{+\infty} k(\tau) \frac{\sin \lambda \tau}{\tau}$$

Le relazioni (15.1) e (15.2) motivano già sufficientemente il punto di vista spettrale nella trattazione del signal processing. La cosa che più, però, ci convince a passare sul dominio frequenza è la possibilità di vedere la densità spettrale come una vera e propria densità di probabilità. In effetti si può dimostrare che

$$\int_{-\infty}^{+\infty} f(\lambda) d\lambda = \sigma_j^2$$

ma l'integrale a primo membro dovrebbe risultare uno per potere parlare di densità di probabilità per $f(\lambda)$. Definiamo quindi una funzione $\varphi(\lambda)$ t.

$$\varphi(\lambda) = \frac{f(\lambda)}{\sigma_j^2} = \frac{1}{2\pi} \sum_{\tau=-\infty}^{+\infty} e^{-i\lambda\tau} \frac{k(\tau)}{\sigma_j^2}$$

Essendo $\frac{k(\tau)}{\sigma_j^2}$ la funzione di correlazione.

Qualche autore chiama la funzione $\varphi(\lambda)$ densità spettrale normalizzata (o effettiva), in quanto questa è una densità di probabilità, essendo

$$\int_{-\infty}^{+\infty} \varphi(\lambda) d\lambda = 1$$

Quindi la densità spettrale $f(\lambda)$, definita nell'intervallo $(-\pi, \pi]$, a meno di una costante di normalizzazione, equivale ad una funzione di densità di probabilità, con i connessi vantaggi interpretativi.

E' anche possibile a questo punto trovare la funzione di distribuzione (detta spettro integrato o cumulato) di $\varphi(\lambda)$ data da

$$\Phi(\lambda) = \int_{-\infty}^{\lambda} \varphi(\omega) d\omega$$

La funzione di covarianza è invece definita per $\tau = 0, \pm 1, \pm 2, \dots$ e, a meno della solita costante di normalizzazione, rappresenta una funzione caratteristica il cui andamento temporale non è facilmente e direttamente interpretabile.

Si noti anche che l'area sottesa alla densità spettrale equivale a $k(0)$. Infatti, dalla (15.1) si ottiene

$$k(0) = \int_{-\pi}^{\pi} f(\lambda) d\lambda$$

15.2 Analisi armonica

Lo studio dell'analisi dei segnali può essere effettuato sia sul dominio temporale, sia sul dominio frequenziale.

L'aspetto frequenziale si basa, in generale, sull'analisi di Fourier e sull'analisi spettrale.

Lo studio dell'analisi dei segnali psicofisiologici nel dominio frequenziale comporta, in generale, maggiori difficoltà sia dal punto di vista analitico sia statistico, che non quello del dominio temporale. Esso richiede, di norma, l'impiego delle quantità complesse e, forse per questo motivo, viene trascurato in alcuni testi che peraltro sono ottimi per lo studio nel dominio temporale.

Fra le infinite funzioni che si conoscono hanno particolare interesse, nello studio dell'analisi dei segnali psicofisiologici nel dominio frequenziale, le funzioni periodiche, cioè quelle funzioni che si ripetono ad intervalli uguali di tempo.

Si dice che la funzione $f(t)$ è periodica con periodo P , se, per ogni $t \geq 0$,

$$f(t) = f(t + KP) \quad \text{per ogni intero } K.$$

Chiaramente il periodo di una funzione periodica non è unico in quanto se $f(t)$ ha periodo P , allora avrà anche periodo $2P, 3P, \dots$

Le funzioni periodiche più usate sono senz'altro le funzioni trigonometriche $\sin \alpha$ e $\cos \alpha$ che hanno un periodo uguale a 2π . Introduciamo adesso degli opportuni parametri e naturalmente il tempo, indispensabile per la nostra indagine, scriveremo l'espressione generale delle funzioni trigonometriche sotto la forma $f(t) = A \cos \lambda t$ oppure $f(t) = A \sin \lambda t$.

La funzione $f(t)$ è detta armonica reale di frequenza λ e di ampiezza A con λ e A costanti positive. Un'armonica di frequenza λ ha periodo $P = \frac{2\pi}{\lambda}$. La frequenza $\lambda = \frac{2\pi}{P}$ rappresenta invece il numero di cicli completi che l'armonica compie in 2π unità di tempo. Chiaramente $\frac{\lambda}{2\pi} = \frac{1}{P}$ non è altro che il numero di cicli in un'unità di tempo. Si dice angolo di fase φ , di un'armonica, quello definito da

$$S(t) = \cos(\lambda t + \varphi)$$

Tale angolo ha lo stesso valore che l'armonica assume nell'origine, dal momento che $s(0) = \cos \varphi$.

In luogo di introdurre la fase φ nella funzione generale per seno o coseno si può scrivere $S(t)$ come somma ponderata delle due funzioni seno e coseno, ottenendo quindi la seguente forma

$$f(t) = A e^{i\lambda t} = A \cos \lambda t + i A \sin \lambda t$$

che chiameremo armonica complessa di ampiezza A , frequenza λ e periodo $P = \frac{2\pi}{\lambda}$.

Passiamo ora ad esaminare il modello dato da

$$X(t) = \sum_{j=1}^n (a_j \cos \lambda_j t + b_j \sin \lambda_j t)$$

in cui a_j e b_j sono variabili aleatorie mentre λ_j sono parametri non aleatori.

Sostituendo $a_j = A_j \cos \varphi_j$ e $b_j = A_j \sin \varphi_j$ (con $A_j > 0$), l'espressione $X(t)$ può essere scritta nel modo seguente:

$$X(t) = \sum_{j=1}^n A_j \cos(\lambda_j t - \varphi_j)$$

in cui

$$A_j = (a_j^2 + b_j^2)^{1/2}, \quad \varphi_j = \arctan\left(\frac{b_j}{a_j}\right) \text{ e } \left| \arctan\left(\frac{b_j}{a_j}\right) \right| < \frac{\pi}{2}$$

Scritta in questa forma $X(t)$ appare come la somma di n armoniche le cui ampiezze A_j e fasi φ_j sono variabili aleatorie, mentre le frequenze λ_j sono fissate.

Ciò posto si può dimostrare che le quantità A_j^2 sono legate alla varianza della serie osservata σ_X^2 dalla seguente relazione

$$\sigma_X^2 = V[X(t)] = \sum_{j=1}^{(N/2)-1} \frac{A_j^2}{2} + a_{N/2}^2 = \sum_{j=1}^n \sigma_j^2$$

nota sotto il nome di teorema di Parseval.

L'importanza di questo teorema è notevole in quanto mette in evidenza il peso che le varie armoniche hanno nel determinare il valore della varianza della serie σ_X^2 . Ovvero le quantità A_j^2 rappresentano il contributo che la j -esima armonica porta alla varianza totale e permette quindi di giudicare l'importanza relativa delle varie armoniche sulla varianza totale.

Nell'analisi matematica accade spesso di dover trasformare una funzione $f(x)$ in un'altra $F(p)$. Si chiama trasformata d'integrali la seguente relazione

$$F(p) = \int_a^b k(p, x) f(x) dx$$

che effettua appunto una trasformazione dalla funzione $f(x)$ alla $F(p)$, dove $k(p, x)$ è nota sotto il nome di nucleo di trasformazione. Tale cambiamento di funzione è spesso giustificato dal fatto che la $F(p)$ è più maneggevole o semplice da usare della $f(x)$. Chiaramente a seconda del nucleo che si sceglie si ottiene, ad

esempio, la trasformata di Laplace, utilizzata per la risoluzione delle equazioni differenziali, o la trasformata di Fourier utilizzata nello studio dell'analisi dei segnali psicofisiologici sul dominio frequenziale, che può essere considerata nel continuo oppure nel discreto, come ora vedremo.

Consideriamo una funzione del tempo $f(t)$ che si estenda sull'intervallo $(-\infty, +\infty)$ e poniamo il nucleo di trasformazione

$$k(p, x) = k(\lambda, t) = e^{-i\lambda t}$$

dove $\frac{1}{i} = -i$ è l'unità immaginaria e λ è la frequenza angolare.

Eseguendo la trasformazione, di cui sopra, otteniamo

$$F(x) = \int_{-\infty}^{\infty} f(x) e^{-i\lambda t} dt$$

che si chiama, ammesso che l'integrale esista per ogni valore di λ , trasformata di Fourier.

Chiaramente ad ogni trasformata diretta corrisponde una trasformata inversa $F(t) \rightarrow f(t)$. Questa si ottiene mediante la cosiddetta formula d'inversione che, nel caso della trasformata di Fourier, è rappresentata dall'integrale

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\lambda) e^{i\lambda t} d\lambda$$

che si chiama appunto trasformata inversa di Fourier.

Nelle nostre applicazioni interessa il caso discreto, nel quale $f(t)$ è definita solamente in corrispondenza a valori interi di t . La trasformata di Fourier diventa in questo caso

$$F(\lambda) = \frac{1}{2\pi} \sum_{t=-\infty}^{\infty} f(t)e^{-i\lambda t}$$

e la sua inversa

$$f(t) = \int_{-\pi}^{\pi} F(\lambda)e^{i\lambda t} d\lambda$$

La Trasformata discreta di Fourier, è caratterizzata da un campo applicativo vastissimo, che spazia dall'interpolazione trigonometrica alla risoluzione di sistemi lineari, dai problemi differenziali alle derivate parziali alle antenne, dall'ottica all'elaborazione dei segnali; in quest'ultimo campo, in particolare, riveste un ruolo fondamentale per le sue applicazioni, soprattutto per quanto riguarda il "filtering" dei segnali, cioè quel processo computazionale che trasforma un segnale campionato in un altro e che prende il nome di filtro digitale.

Sebbene la trasformata discreta di Fourier risulti una potentissima procedura di risoluzioni di molti problemi matematici, essa è caratterizzata da una complessità computazionale che è eccessiva quando si utilizza la trasformata per risolvere problemi di grandi

dimensioni. Tutto ciò ha portato alla ricerca di un algoritmo per il calcolo della Trasformata discreta di Fourier che avesse una complessità inferiore.

Nel 1965 l'ingegnere dell'IBM Cooley e il matematico Tukey giunsero alla determinazione dell'algoritmo noto come "Decimation in time", detto anche Fast Fourier Transform (FFT), che riduceva notevolmente la complessità computazionale della Trasformata e ne aumentava inoltre la stabilità. L'FFT è semplicemente un algoritmo che consente di calcolare in modo molto rapido la trasformata discreta di Fourier.

La strategia su cui si basano gli algoritmi FFT, detti "divide and conquer", è quella di ricondurre il calcolo di una trasformata discreta di lunghezza N , al calcolo di una trasformata discreta di lunghezza inferiore.

Una variante all'algoritmo decimation in time è il "decimation in frequency". La differenza principale fra i due algoritmi è nell'ordine con cui sono svolte alcune operazioni. Nel decimation in time, anche detto algoritmo di Cooley-Tukey, gli elementi in input vengono riarrangiati mediante un'operazione di bit-reverse, e quindi viene calcolato l'output della FFT, mediante $\log_2 N$ iterazioni. Nel decimation in frequency o algoritmo FFT di Sande-Tukey vengono preliminarmente eseguite $\log_2 N$ iterazioni sui dati di ingresso e quindi viene applicato il bit-reverse.

Una classe di algoritmi FFT è quella detta radix- r , in cui il calcolo di una trasformata discreta di lunghezza $N = r^s$, è ricondotto al calcolo di una trasformata discreta di lunghezza r . Se r è uguale a

due, allora l'algoritmo è di tipo particolarmente agevole (FFT radix-2). In generale esistono algoritmi per $N = r_1 r_2 \dots r_k$ detti mixed-radix.

Per quanto riguarda il vantaggio computazionale, mentre il calcolo di una trasformata discreta di lunghezza N con il metodo tradizionale ha una complessità di tempo di $O(n^2)$ e quelli di tipo mixed-radix una complessità di tempo $O[n(r_1 + r_2 + \dots + r_k)]$, gli algoritmi FFT ne hanno una di $O(n \log_2 N)$, da cui segue la convenienza a adottare tali algoritmi.

15.3 La stima dei segnali

Dato un campione X_1, \dots, X_N di ampiezza N , la funzione $I(\lambda)$, chiamata il periodogramma, è definita per tutti i λ nell'intervallo $(-\pi < \lambda \leq \pi)$ da

$$I(\lambda) = \left\{ [A(\lambda)]^2 + [B(\lambda)]^2 \right\}$$

Dove

$$A(\lambda) = \frac{1}{\sqrt{2\pi N}} \left\{ \sum_{t=1}^N \cos \lambda t X(t) \right\}$$

$$B(\lambda) = \frac{1}{\sqrt{2\pi N}} \left\{ \sum_{t=1}^N \sin \lambda t X(t) \right\}$$

In alternativa $I(\lambda)$ può essere scritta nella forma

$$I(\lambda) = \frac{1}{2\pi N} \left| \sum_{t=1}^N e^{-i\lambda t} X(t) \right|^2$$

Proviamo adesso a trasformare quest'ultima espressione in qualcosa di più familiare

$$I(\lambda) = \frac{1}{2\pi N} \left| \sum_{t=1}^N e^{-i\lambda t} X(t) \right|^2 =$$

$$I(\lambda) = \frac{1}{2\pi N} \left[\sum_{t=1}^N e^{-i\lambda t} X(t) \right] \left[\sum_{s=1}^N e^{i\lambda s} X(s) \right]$$

$$I(\lambda) = \frac{1}{2\pi N} \left[\sum_{t=1}^N \sum_{s=1}^N X(t) X(s) \cos(t-s)\lambda \right]$$

I termini in seno, chiaramente, se ne vanno per la simmetria del fattore $[X(t) \cdot X(s)]$

$$I(\lambda) = \frac{1}{2\pi} \sum_{t-s=-(N-1)}^{N-1} \left[\frac{1}{N} \sum_{t=1}^{N-|t-s|} X(t) X(t+|t-s|) \right] \cos(t-s)\lambda$$

$$I(\lambda) = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \left[\frac{1}{N} \sum_{t=1}^{N-|r|} X(t) X(t+|r|) \right] \cos r\lambda$$

$$I(\lambda) = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \hat{k}(r) \cos r\lambda$$

Quindi il periodogramma può essere espresso come

$$I(\lambda) = \frac{1}{2\pi} \left[\hat{k}(0) + 2 \sum_{r=1}^{N-1} \hat{k}(r) \cos r\lambda \right]$$

Con $\hat{k}(r) = \frac{1}{N} \sum_{t=1}^{N-|r|} X(t)X(t+|r|)$.

La funzione $\hat{k}(r)$ viene chiamata funzione di covarianza campionaria di $X(t)$. Confrontando il periodogramma espresso in quest'ultima forma con la densità spettrale per successioni stazionarie reali, rileviamo subito che il periodogramma costituisce un'ovvia stima della densità spettrale $f(\lambda)$.

Passiamo adesso ad esaminare qualche proprietà del periodogramma al fine di poter dare un giudizio su questa stima.

$$E[I(\lambda)] = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} E[\hat{k}(r)] \cos r\lambda$$

$$E[I(\lambda)] = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \left[\frac{1}{N} \sum_{t=1}^{N-|r|} k(r) \right] \cos r\lambda$$

$$E[I(\lambda)] = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \left[k(r) - \frac{|r|}{N} k(r) \right] \cos r\lambda$$

$$E[I(\lambda)] = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \left(1 - \frac{|r|}{N} \right) k(r) \cos r\lambda$$

Quando $N \rightarrow \infty$ il fattore $\left(1 - \frac{|r|}{N} \right) \rightarrow 1$ per ogni r . Pertanto, per $N \rightarrow \infty$

$$E[I(\lambda)] \rightarrow \frac{1}{2\pi} \sum_{r=-\infty}^{+\infty} k(r) \cos r\lambda$$

ovvero $E[I(\lambda)] \rightarrow f(\lambda)$.

Questo dimostra che, per ogni λ , $I(\lambda)$ è uno stimatore asintoticamente non distorto per $f(\lambda)$, cioè per la densità spettrale. Ricordiamo che facendo tendere N all'infinito entriamo automaticamente in una casistica di grandi campioni.

Possiamo dimostrare in maniera più rigorosa questo risultato sostituendo $k(r) = \int_{-\pi}^{\pi} f(\tau) \cos r\tau d\tau$ nella precedente.

L'aver scoperto che $I(\lambda)$ è uno stimatore asintoticamente non distorto per $f(\lambda)$ è un'ottima cosa, ma non è sufficiente per dare un giudizio positivo allo stimatore. Si può dimostrare che

$$\text{Var}[I(\lambda)] = \frac{\hat{k}(0)}{\pi^2}$$

Quindi, come facilmente constatabile, la varianza di $I(\lambda)$ non dipende da N , cioè non dipende dall'ampiezza del campione. Quest'ultima cosa è notevolmente preoccupante perché in questo modo il periodogramma viene a mancare di un importante requisito statistico, ossia la consistenza. Per essere tale, quando il campione si allarga, la varianza dovrebbe tendere a zero.

La mancanza della consistenza fa rilevare varianze importanti in corrispondenza di parecchie frequenze, questo lascia presupporre l'esistenza di cicli di periodo differente anche per fenomeni sui quali a priori si poteva scartare l'ipotesi di componenti periodiche; pertanto possiamo dire che il periodogramma è una stima asintoticamente corretta, ma non consistente.

Vi è un'altra cosa importante da aggiungere fra le proprietà del periodogramma che rendono tale strumento inadeguato al nostro scopo. Si può infatti vedere che il periodogramma, come funzione di λ , ha un comportamento fortemente oscillatorio ed erratico. Si dimostra, infatti, che prese λ_1 e λ_2 successive, quando i dati sono normali, $I(\lambda_1)$ e $I(\lambda_2)$ sono indipendenti fra di loro. In altri termini $\text{Cov}[I(\lambda_1), I(\lambda_2)]$ decresce quando N cresce.

Si verificano quindi delle forti oscillazioni, nei valori campionari di $I(\lambda)$, che rendono complessa la stima.

Queste oscillazioni devianti dell'effettivo andamento della densità spettrale le possiamo vedere chiaramente se confrontiamo

l'andamento teorico di un rumore bianco con l'andamento della stima tramite periodogramma.

Il rumore bianco ha densità teorica fissa (costante) pari a $1/2\pi$, mentre l'andamento che si ottiene come stima attraverso il periodogramma ha un andamento fortemente oscillatorio.

Per dovere di cronaca, faccio notare che stiamo parlando di densità

spettrale normalizzata, infatti
$$\int_{-\infty}^{+\infty} \varphi(\lambda) d\lambda = \int_{-\pi}^{+\pi} \frac{1}{2\pi} d\lambda.$$

Chiaramente la densità spettrale non normalizzata sarà data da

$$f(\lambda) = \sigma_j^2 \varphi(\lambda) = \frac{\sigma_j^2}{2\pi}.$$

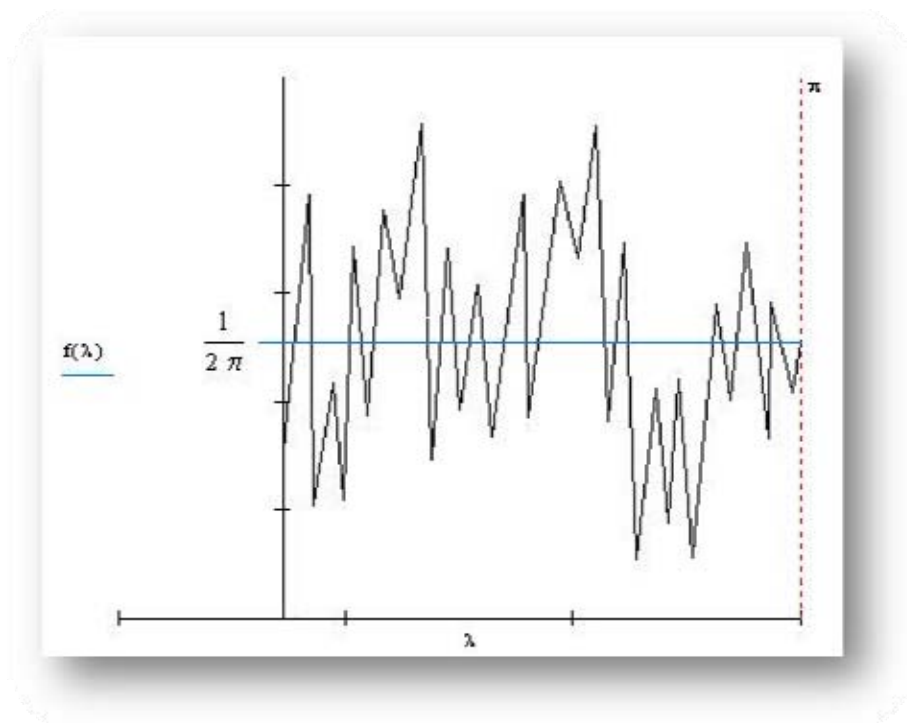


Figura 15.1. Rumore bianco con densità teorica fissa (costante) pari a $1/2\pi$.

Abbiamo constatato che il periodogramma è uno strumento non preciso, anche a causa della sua elevata varianza. Questa non tende a zero quando N tende a infinito. Potremmo notare che la varianza del periodogramma non tende a zero semplicemente perché questa contiene troppe funzioni di covarianza campionarie.

Un metodo per rendere utilizzabile questo strumento è ridurre la varianza. Per effettuare questa riduzione possiamo semplicemente omettere "alcuni" termini della funzione di covarianza campionaria. Facendolo, sicuramente ridurremo la varianza, ma, in questo modo, i termini omessi influenzeranno il valore della nuova funzione e di conseguenza ci sarà un aumento della distorsione ed il pericolo di un aumento della tendenziosità della stima.

Comunque noi sappiamo che $k(\tau) \rightarrow 0$ quando $|\tau| \rightarrow \infty$ e quindi omettiamo solo quei termini che corrispondono alla coda della funzione di covarianza campionaria. Si può allora pensare di stimare la densità spettrale con

$$\hat{f}_0(\lambda) = \frac{1}{2\pi} \sum_{r=-M}^{+M} k(r) \cos \lambda r$$

dove M è un intero minore di $N-1$. Questa stima si chiama periodogramma troncato ed M si chiama punto di troncamento.

Come già fatto in precedenza per il periodogramma, calcoliamo il valore atteso di questo stimatore.

$$\begin{aligned}
 E[\hat{f}_0(\lambda)] &= \frac{1}{2\pi} \sum_{r=-M}^{+M} E[\hat{k}(r)] \cos \lambda r \\
 &= \frac{1}{2\pi} \sum_{r=-M}^{+M} \left(1 - \frac{|r|}{N}\right) k(r) \cos \lambda r
 \end{aligned}$$

Quando $M \rightarrow \infty$ si avrà

$$E[\hat{f}_0(\lambda)] \rightarrow f(\lambda)$$

Quindi se la scelta di M dipende da N di modo che $M \rightarrow \infty$ quando $N \rightarrow \infty$, allora $\hat{f}_0(\lambda)$ sarà uno stimatore asintoticamente non distorto per la densità spettrale $f(\lambda)$.

Peraltro, se scegliamo M ed N , con $M \rightarrow \infty$ ed $N \rightarrow \infty$, in modo che (Si precisa che bisogna fare in modo, chiaramente, che M vada all'infinito più lentamente rispetto ad N ; questo è possibile, dal momento che scegliamo noi il valore di M) $M/N \rightarrow 0$ quando $N \rightarrow \infty$, allora sia la tendenziosità che la varianza, di $\hat{f}_0(\lambda)$, tendono a zero per $N \rightarrow \infty$. Questo risultato è importantissimo in quanto dimostra che il periodogramma troncato da una stima consistente della funzione di densità spettrale. Inoltre, $\hat{f}_0(\lambda)$ è stimatore asintoticamente non distorto di $f(\lambda)$.

E' abbastanza facile trovare dei valori di M che soddisfano le condizioni di cui sopra. Potremmo ad esempio prendere $M = N^{1/2}$ o più in generale $M = N^\alpha$ con $0 < \alpha < 1$.

Il periodogramma troncato potrebbe essere considerato come un caso particolare del seguente stimatore

$$\hat{f}(\lambda) = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) \hat{k}(r) \cos \lambda r$$

che nel caso in cui

$$\theta(r) = \begin{cases} 1 & |r| \leq M \\ 0 & |r| > M \end{cases}$$

sarà esattamente $\hat{f}(\lambda)$. Esistono diversi tipi di funzioni $\theta(r)$, generalmente chiamate pesi, per ovvi motivi. La funzione dei pesi deve soddisfare le seguenti condizioni

$$\begin{aligned} |\theta(r)| &\leq 1 \\ \theta(0) &= 1 \\ \theta(r) &= \theta(-r) \\ \theta(r) &= 0 \quad \text{per } |r| > M \end{aligned}$$

La prima condizione è alquanto scontata, essendo $\theta(r)$ un peso. La seconda indica che per $r=0$ la finestra deve avere peso unitario. La terza definisce la simmetria della finestra rispetto ad r . La quarta, infine, precisa che oltre il punto di troncamento il peso deve portare all'annullamento della funzione $\hat{f}(\lambda)$.

Parecchi autori chiamano la funzione $\theta(r)$, "lag window" o finestra del ritardo. E' importante far rilevare che la funzione $\hat{f}(\lambda)$ può essere riscritta in una forma alternativa. Siccome sia $\hat{k}(r)$, sia $\cos \lambda r$ sono funzioni pari di r , possiamo scegliere $\theta(r)$ in modo che sia anch'essa funzione pari di r . Avremo dunque

$$\hat{f}(\lambda) = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) \hat{k}(r) e^{-i\lambda r}$$

che ci tornerà utile tra poco. Oltre alla finestra dei ritardi, $\theta(r)$, è molto utile anche un'altra funzione che generalmente viene chiamata finestra spettrale. Indicheremo questa funzione con $W(\lambda)$ definita come

$$W(\lambda) = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) e^{-i\lambda r}$$

che, come si può immediatamente constatare, è la trasformata di Fourier dei pesi. La finestra dei ritardi e la finestra spettrale sono strettamente collegate. Per vedere in che modo dobbiamo partire dalla funzione di covarianza espressa come trasformata di Fourier della densità spettrale. In pratica calcoliamo la trasformata di Fourier inversa di $\hat{f}(\lambda)$; cioè dalla (15.1)

$$\hat{k}(r) = \int_{-\pi}^{+\pi} \hat{f}(\lambda_1) e^{i\lambda_1 r} d\lambda \quad \text{per } r = 0, \pm 1, \dots, \pm(N-1)$$

Quindi

$$\begin{aligned} \hat{f}(\lambda) &= \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) \hat{k}(r) e^{-i\lambda_2 r} \\ &= \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) \int_{-\pi}^{+\pi} \hat{f}(\lambda_1) e^{i\lambda_1 r} e^{-i\lambda_2 r} \\ &= \int_{-\pi}^{+\pi} \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) e^{-i(\lambda_2 - \lambda_1)r} \hat{f}(\lambda_1) d\lambda \\ &= \int_{-\pi}^{+\pi} W(\lambda_2 - \lambda_1) \hat{f}(\lambda_1) d\lambda \\ &= \int_{-\pi}^{+\pi} W(\lambda_1) \hat{f}(\lambda_2 - \lambda_1) d\lambda \end{aligned}$$

Da quest'ultima espressione è chiaro, dunque, che la finestra spettrale, $W(\lambda)$, è la trasformata di Fourier della finestra del ritardo e che la finestra del ritardo è la trasformata inversa di Fourier della finestra spettrale, cioè

$$\theta(r) = \int_{-\pi}^{+\pi} W(\lambda_2) e^{i\lambda_2 r} d\lambda_2 \quad \text{per } r = 0, \pm 1, \dots, \pm(N-1)$$

Una cosa molto importante la possiamo notare ricordando che il periodogramma è un caso particolare dello stimatore $\hat{f}(\lambda_1)$, quindi, eseguendo gli stessi calcoli con $I(\lambda)$ al posto di $\hat{f}(\lambda_1)$ otterremo che

$$\hat{f}(\lambda) = \int_{-\pi}^{+\pi} W(\lambda_2 - \tau) I(\tau) d\tau$$

Quindi la stima della densità spettrale è un integrale ponderato del periodogramma $I(\tau)$, dove i pesi $W(\tau)$ sono forniti dalla finestra spettrale. I termini finestra spettrale e finestra del ritardo sono stati dati da Blackman e Tukey nel 1958; funzione ponderata e sequenza ponderata erano i consueti termini prima usati dalla letteratura.

Analizziamo adesso i diversi tipi di finestra spettrale, cogliendone di volta in volta gli aspetti positivi e quelli negativi che ne hanno reso una maggiore diffusione.

La finestra rettangolare

La finestra del ritardo più semplice che possiamo immaginare è quella esprimibile come

$$\theta(r) = \begin{cases} 1 & |r| \leq M \\ 0 & |r| > M \end{cases}$$

Tramite questa espressione di $\theta(r)$, abbiamo già visto che si arrivava a esprimere $\hat{f}_0(\lambda)$, ovvero il periodogramma troncato. E' per questa ragione che qualche autore chiama questa finestra del ritardo e la finestra spettrale corrispondente, rispettivamente finestra del ritardo del periodogramma troncato e finestra del periodogramma troncato. Dalla finestra del ritardo rettangolare deriviamo adesso la finestra spettrale rettangolare⁸. Ovvero

$$\begin{aligned}
 W(\lambda) &= \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) e^{-i\lambda r} \\
 &= \frac{1}{2\pi} \sum_{r=-M}^{+M} e^{-i\lambda r} \\
 &= \frac{1}{2\pi} \left(1 + 2 \sum_{r=1}^{+M} \cos \lambda r \right) \\
 &= \frac{1}{2\pi} \left\{ 1 + 2 \frac{\cos[\lambda(M+1)/2] \sin(\lambda M/2)}{\sin(\lambda/2)} \right\} \\
 &= \frac{1}{2\pi} \frac{\sin(\lambda/2) + \sin[\lambda(M+1/2)] - \sin(\lambda/2)}{\sin(\lambda/2)} \\
 &= \frac{1}{2\pi} \frac{\sin[\lambda(M+1/2)]}{\sin(\lambda/2)}
 \end{aligned}$$

La finestra dei ritardi e la finestra spettrale sono rappresentate in Figura 15.2.

⁸ Si ricorda che per passare dalla finestra del ritardo alla finestra spettrale basta fare la trasformata di

$$\text{Fourier della prima. Ovvero } W(\lambda) = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) e^{-i\lambda r}$$

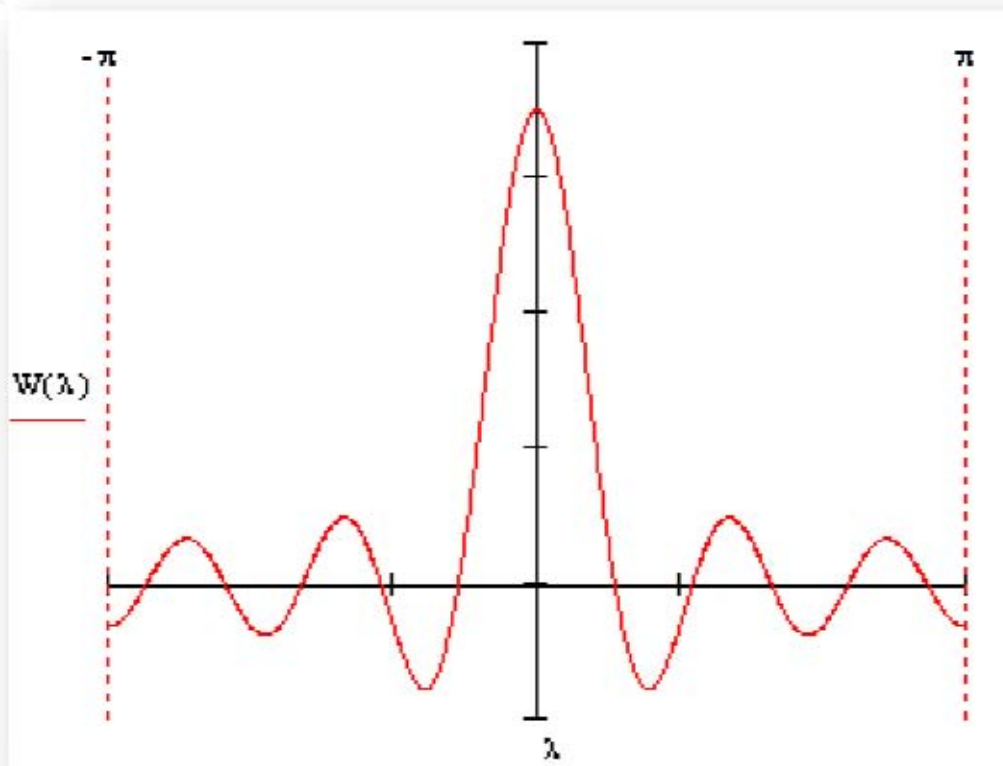
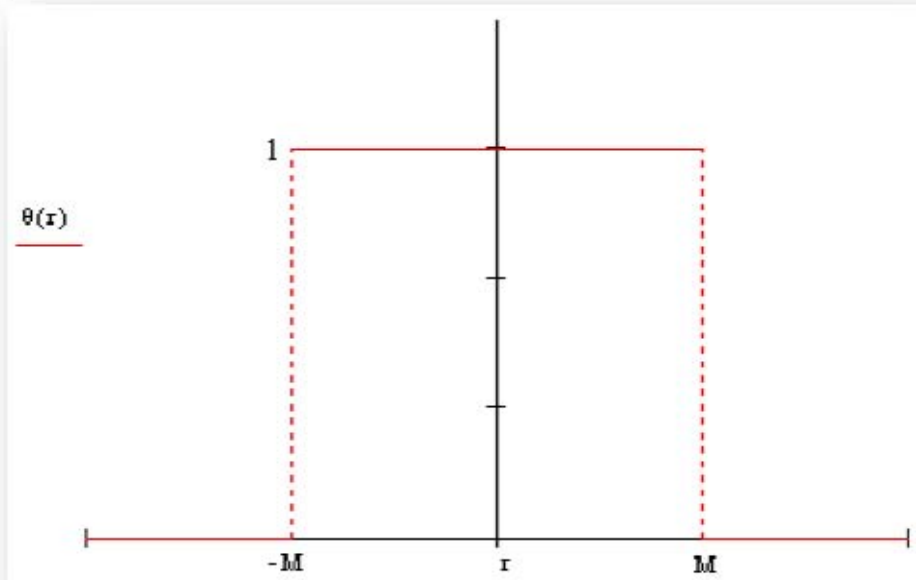


Figura 15.2. Finestra rettangolare dei ritardi e la finestra spettrale.

Come si può vedere dalla figura 15.2, la finestra spettrale ha un “lobo” (o picco) principale in corrispondenza di $\lambda = 0$ con un'altezza di $(2M+1)/2\pi$ e dei lobi (picchi) laterali a magnitudine (grandezza) decrescente per $\lambda = \pm \frac{(4j+1)\pi}{2M+1}$ approssimativamente e tocca lo zero per $\lambda = \pm \frac{2j\pi}{2M+1}$ con $j = 1, 2, \dots$.

La finestra spettrale rettangolare è anche conosciuta come “kernel di Dirichlet” e gioca un importante ruolo nell'analisi di Fourier. E' importante vedere un aspetto parecchio negativo di questa finestra. Vi sono valori di λ che danno luogo a valori negativi di $W(\lambda)$. Questo, chiaramente, produce a sua volta valori negativi nella densità spettrale che, come abbiamo visto, deve essere sempre positiva (anche se non normalizzata). Proprio per evitare questo inconveniente Barlett nel 1950 circa, propose una nuova finestra.

La finestra di Barlett

La finestra dei ritardi, proposta da Barlett, ha la seguente forma

$$\theta(r) = \begin{cases} 1 - \frac{|r|}{M} & |r| \leq M \\ 0 & |r| > M \end{cases}$$

Esattamente come nel caso precedente, M determina il punto nel quale la funzione di covarianza campionaria viene troncata.

Applicando, come di consueto, la trasformata di Fourier, deriviamo la corrispondente finestra spettrale, data da

$$\begin{aligned}
 W(\lambda) &= \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) e^{-i\lambda r} \\
 &= \frac{1}{2\pi} \sum_{r=-M}^M \left(1 - \frac{|r|}{M}\right) e^{-i\lambda r} \\
 &= \frac{1}{2\pi M} \sum_{r=-M}^M (M - |r|) e^{-i\lambda r} \\
 &= \frac{1}{2\pi M} \sum_{s=0}^{M-1} \sum_{r=-s}^{+s} e^{-i\lambda r}
 \end{aligned}$$

Adesso, per i calcoli fatti con la finestra rettangolare, sappiamo

che $\sum_{r=-M}^{+M} e^{-i\lambda r} = \frac{\sin[\lambda(M+1/2)]}{\sin(\lambda/2)}$.

Quindi,

$$\begin{aligned}
 W(\lambda) &= \frac{1}{2\pi M} \sum_{s=0}^{M-1} \frac{\sin[\lambda(s+1/2)]}{\sin(\lambda/2)} \\
 &= \frac{1}{2\pi M \sin(\lambda/2)} \sum_{s=0}^{M-1} \sin[\lambda(s+1/2)] \\
 &= \frac{1}{2\pi M \sin(\lambda/2)} \left\{ \sin(\lambda/2) + \sum_{s=1}^{M-1} \sin[\lambda(s+1/2)] \right\} \\
 &= \frac{1}{2\pi M \sin(\lambda/2)} \left\{ \sin(\lambda/2) + \sum_{s=1}^{M-1} \sin(\lambda s) \cos(\lambda/2) + \cos(\lambda s) \sin(\lambda/2) \right\} \\
 &= \frac{1}{2\pi M \sin(\lambda/2)} \left\{ \sin(\lambda/2) + \cos(\lambda/2) \sum_{s=1}^{M-1} \sin(\lambda s) + \sin(\lambda/2) \sum_{s=1}^{M-1} \cos(\lambda s) \right\}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2\pi M \sin(\lambda/2)} \left\{ \sin(\lambda/2) + \cos(\lambda/2) \sum_{s=1}^{M-1} \sin(\lambda s) + \sin(\lambda/2) \sum_{s=1}^{M-1} \cos(\lambda s) \right\} \\
&= \frac{1}{2\pi M \sin(\lambda/2)} \left\{ \sin(\lambda/2) + \cos(\lambda/2) \frac{\sin(\lambda M/2) \sin \frac{\lambda(M-1)}{2}}{\sin(\lambda/2)} + \right. \\
&\quad \left. + \sin(\lambda/2) \frac{\cos(\lambda M/2) \sin \frac{\lambda(M-1)}{2}}{\sin(\lambda/2)} \right\} \\
&= \frac{1}{2\pi M \sin(\lambda/2)} \left\{ \sin(\lambda/2) + \frac{\sin \frac{\lambda(M-1)}{2}}{\sin(\lambda/2)} [\cos(\lambda/2) \sin(\lambda M/2) + \right. \\
&\quad \left. + \sin(\lambda/2) \cos(\lambda M/2)] \right\} \\
&= \frac{1}{2\pi M \sin(\lambda/2)} \left\{ \sin(\lambda/2) + \frac{\sin \frac{\lambda(M-1)}{2}}{\sin(\lambda/2)} \sin \left[\lambda \left(\frac{M+1}{2} \right) \right] \right\} \\
&= \frac{1}{2\pi M \sin^2(\lambda/2)} \left\{ \sin^2(\lambda/2) + \sin \frac{\lambda(M-1)}{2} \sin \frac{\lambda(M+1)}{2} \right\} \\
&= \frac{1}{2\pi M \sin^2(\lambda/2)} \left\{ \frac{1}{2} (1 - \cos \lambda) + \frac{1}{2} (\cos \lambda - \cos \lambda M) \right\} \\
&= \frac{1}{2\pi M \sin^2(\lambda/2)} \left\{ \frac{1}{2} [1 - \cos(\lambda M)] \right\} \\
&= \frac{1}{2\pi M \sin^2(\lambda/2)} \sin^2(\lambda M/2) = \frac{1}{2\pi M} \left[\frac{\sin(\lambda M/2)}{\sin(\lambda/2)} \right]^2
\end{aligned}$$

In Figura 15.3, la finestra dei ritardi e la finestra spettrale.

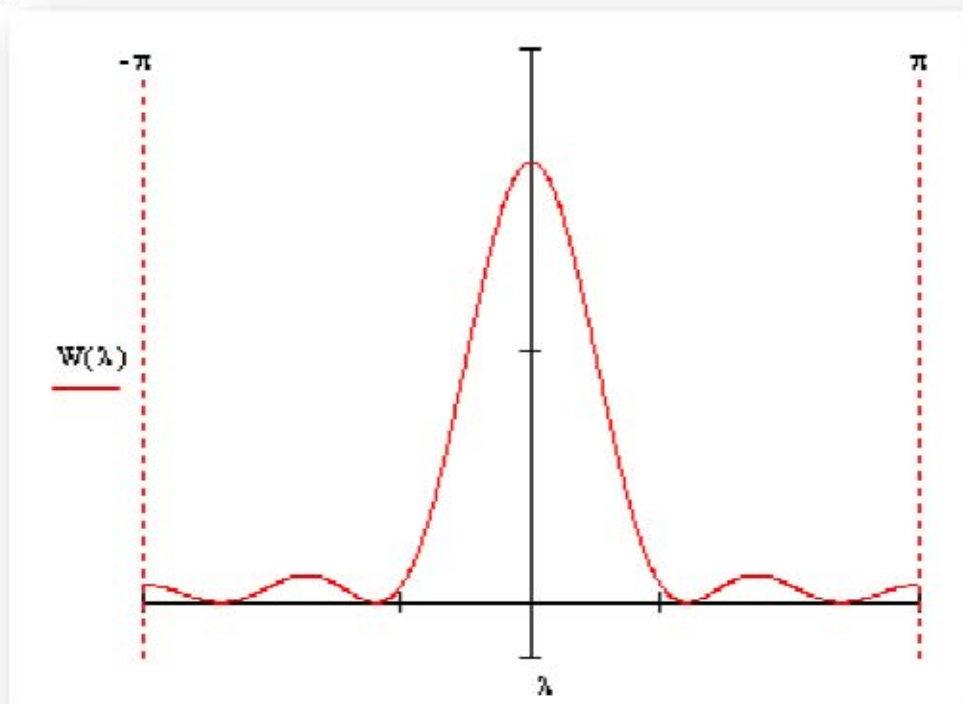
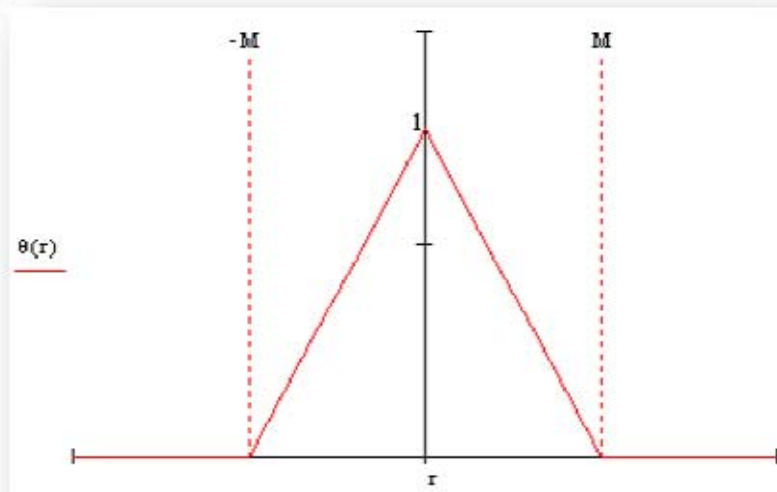


Figura 15.3. Finestra di Barrett dei ritardi e la finestra spettrale.

Come si rileva immediatamente, guardando la figura 15.3, la finestra spettrale di Barlett non assume valori negativi, per ogni valore di λ . Da un'attenta analisi comparativa, eseguita al computer, delle Figure 15.2 e 15.3, si nota che i lobi (picchi) della finestra di Barlett sono più piccoli di quelli della finestra rettangolare. L'effetto dei lobi (picchi) grandi è dato dal fatto che $\hat{f}(\lambda)$ dà più importanza alle frequenze lontane da λ . Quindi la stima spettrale può avere componenti spettrali significative ad altre frequenze diverse da λ . Questo fenomeno è noto come dispersione.

La finestra di Blackman–Tukey generale

Nel 1959 Blackman e Tukey suggerirono la seguente finestra dei ritardi

$$\theta(r) = \begin{cases} 1 - 2a + 2a \cos(\pi r / M) & |r| \leq M \\ 0 & |r| > M \end{cases}$$

Come nei casi precedenti M è il punto di troncamento, mentre a è un parametro scelto nell'intervallo $0 < a \leq \frac{1}{4}$ per fare in modo che $\theta(r)$ sia positiva per ogni r .

La finestra spettrale corrispondente è una combinazione lineare del kernel di Dirichlet e può essere derivata come segue.

$$\begin{aligned}
W(\lambda) &= \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) e^{-i\lambda r} \\
&= \frac{1}{2\pi} \sum_{r=-M}^M [1 - 2a + 2a \cos(\pi r / M)] e^{-i\lambda r} \\
&= \frac{1}{2\pi} \sum_{r=-M}^M \left[1 - 2a + 2a \frac{e^{i\pi r / M} + e^{-i\pi r / M}}{2} \right] e^{-i\lambda r} \\
&= \frac{1}{2\pi} \sum_{r=-M}^{+M} (1 - 2a) e^{-i\lambda r} + \frac{1}{2\pi} \sum_{r=-M}^{+M} a e^{-i r(\lambda - \pi / M)} + \frac{1}{2\pi} \sum_{r=-M}^{+M} a e^{-i r(\lambda + \pi / M)} \\
&= (1 - 2a) \frac{1}{2\pi} \sum_{r=-M}^{+M} e^{-i\lambda r} + a \frac{1}{2\pi} \sum_{r=-M}^{+M} e^{-i r(\lambda - \pi / M)} + a \frac{1}{2\pi} \sum_{r=-M}^{+M} e^{-i r(\lambda + \pi / M)}
\end{aligned}$$

Adesso, per i calcoli fatti con la finestra rettangolare, sappiamo

$$\text{che } \sum_{r=-M}^{+M} e^{-i\lambda r} = \frac{\sin[\lambda(M + 1/2)]}{\sin(\lambda/2)},$$

Quindi

$$\begin{aligned}
W(\lambda) &= (1 - 2a) \frac{1}{2\pi} \frac{\sin[\lambda(M + 1/2)]}{\sin(\lambda/2)} + \\
&\quad + a \frac{1}{2\pi} \frac{\sin[(\lambda - \pi / M)(M + 1/2)]}{\sin[(\lambda - \pi / M) / 2]} + \\
&\quad + a \frac{1}{2\pi} \frac{\sin[(\lambda + \pi / M)(M + 1/2)]}{\sin[(\lambda + \pi / M) / 2]}
\end{aligned}$$

Quindi questa finestra spettrale è la combinazione lineare ponderata della finestra rettangolare. Qualche autore preferisce dire che la finestra di Blackman–Tukey è la combinazione lineare ponderata del kernel di Dirichlet. Non cambia nulla. E' molto più

interessante, invece, far notare che lo stimatore di Blackman–Tukey, $\hat{f}_{BT}(\lambda)$ che deriva da questa finestra, può essere elegantemente espresso come

$$\hat{f}_{BT}(\lambda) = (1 - 2a)\hat{f}_0(\lambda) + a\hat{f}_0(\lambda - \pi/M) + a\hat{f}_0(\lambda + \pi/M)$$

dove, come abbiamo visto, $\hat{f}_0(\lambda)$ è lo stimatore della densità spettrale tramite periodogramma troncato. Quindi lo stimatore Blackman–Tukey è una combinazione lineare ponderata del valore che assume il periodogramma troncato alle frequenze $(\lambda - \pi/2)$ e $(\lambda + \pi/2)$.

La finestra di Tukey–Hamming

Originariamente Tukey suggerì di prendere $a = 0.23$, come valore del parametro, nel qual caso la finestra dei ritardi diviene

$$\theta(r) = \begin{cases} 0.54 + 0.46 \cos(\pi r / M) & |r| \leq M \\ 0 & |r| > M \end{cases}$$

Questo valore di a , sembra sia stato da lui scelto per ridurre la magnitudine del lobo (picco) principale. La finestra spettrale che deriva da questa finestra del ritardo, viene chiamata in letteratura, finestra di Tukey–Hamming.

La finestra di Tukey–Hanning

Questa finestra è quella che otteniamo dando $a = 0.25$. E' una tra le più usate fra quelle viste. Per questo motivo la letteratura si riferisce a questa come la finestra di Tukey, dando per scontato questo valore di a . La finestra dei ritardi può essere espressa come segue

$$\theta(r) = \begin{cases} \frac{1}{2} [1 + \cos(\pi r / M)] & |r| \leq M \\ 0 & |r| > M \end{cases}$$

La cosa più interessante è lo stimatore di Tukey–Hanning, $\hat{f}_{TH}(\lambda)$, che viene, chiaramente, espresso in funzione di $\hat{f}_0(\lambda)$.

Quindi

$$\hat{f}_{BT}(\lambda) = \frac{1}{2} \hat{f}_0(\lambda) + \frac{1}{4} \hat{f}_0(\lambda - \pi / M) + \frac{1}{4} \hat{f}_0(\lambda + \pi / M)$$

La finestra dei ritardi e la finestra spettrale di Tukey-Hanning sono rappresentate in Figura 15.4.

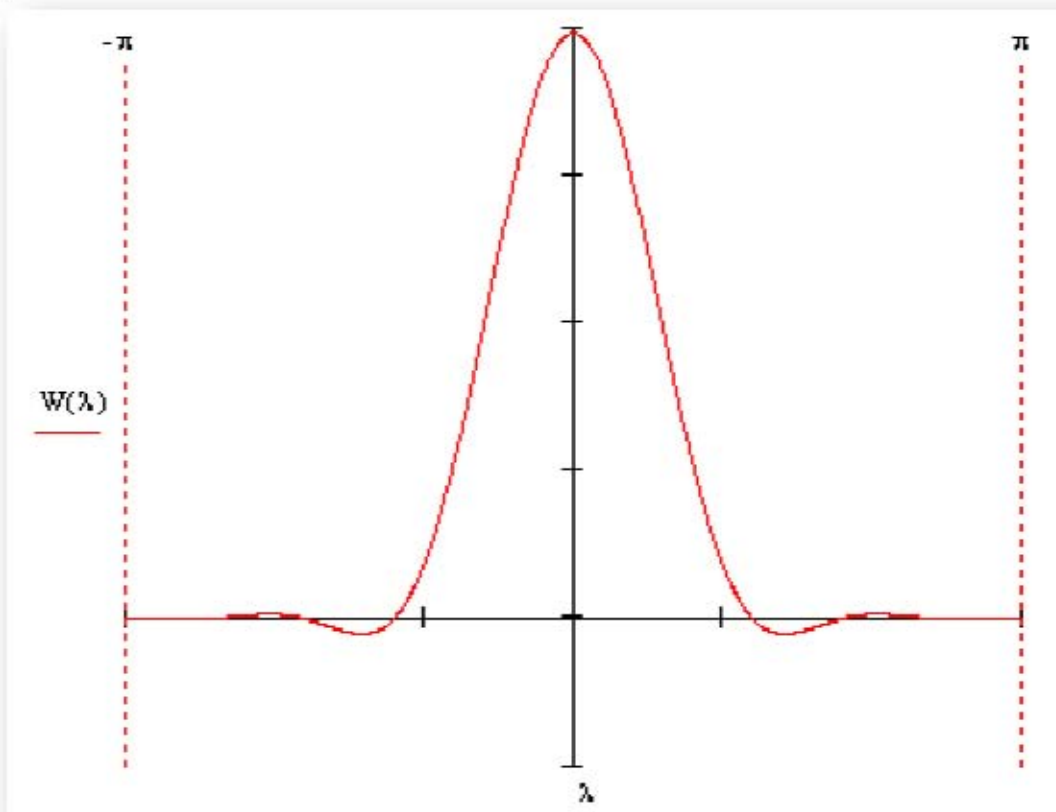
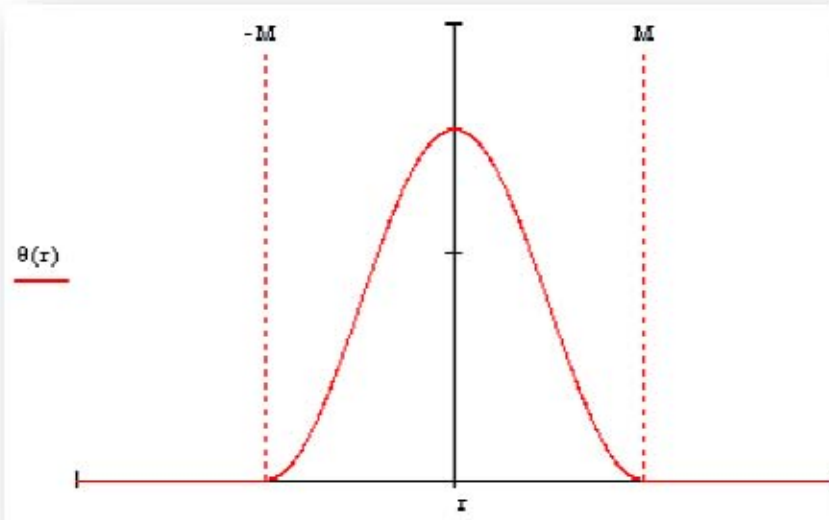


Figura 15.4. Finestra di Tukey dei ritardi e la finestra spettrale.

La finestra di Parzen

Nel 1961, Parzen, suggerì la seguente finestra dei ritardi

$$\theta(r) = \begin{cases} 1 - 6(r/M)^2 + 6(r/M)^3 & |r| \leq M/2 \\ 2(1 - |r|/M)^3 & M/2 < |r| \leq M \\ 0 & |r| > M \end{cases}$$

La corrispondente finestra spettrale è data da

$$\begin{aligned} W(\lambda) &= \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) e^{-i\lambda r} \\ &= \frac{1}{2\pi} \left\{ \sum_{r=-M/2}^{+M/2} [1 - 6(|r|/M)^2 + 6(|r|/M)^3] e^{-i\lambda r} + \right. \\ &\quad \left. + \sum_{r=+M/2}^{+M} [2(1 - |r|/M)^3] e^{-i\lambda r} \right\} \\ &= \frac{3}{8\pi M^3} \left[\frac{\sin(\lambda M/4)}{\frac{1}{2} \sin(\lambda/2)} \right]^4 \left[1 - \frac{2}{3} \sin^2(\lambda/2) \right] \end{aligned}$$

Per M grande quest'espressione è approssimabile come segue

$$W(\lambda) \sim \frac{3}{8\pi M^3} \left[\frac{\sin(\lambda M/4)}{\frac{1}{2} \sin(\lambda/2)} \right]^4$$

La finestra dei ritardi di Parzen e la finestra spettrale di Parzen sono rappresentate in Figura 15.5.

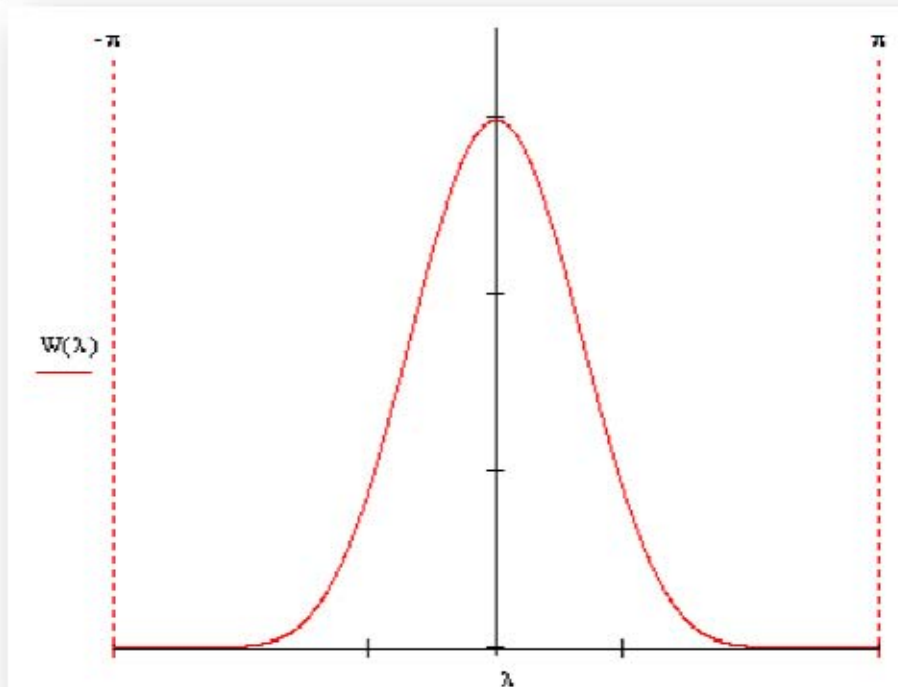
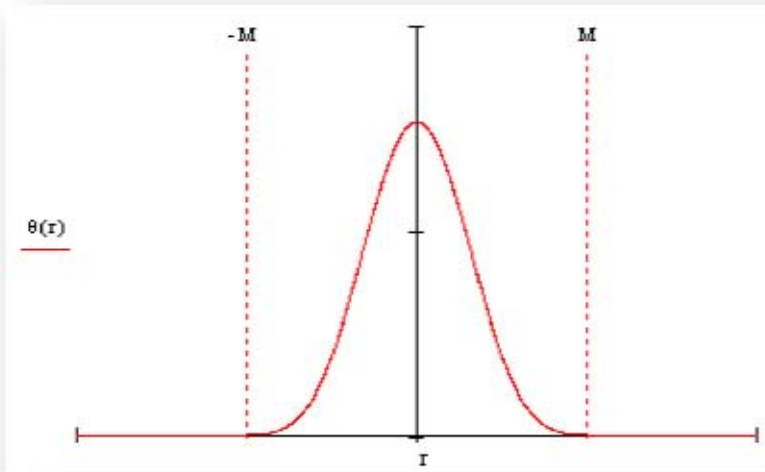


Figura 15.5. Finestra di Parzen dei ritardi e la finestra spettrale.

E' importante sottolineare che l'espressione che abbiamo visto, della finestra di Parzen, vale per M pari.

Parzen suggerì anche un'altra finestra dei ritardi che deriva dalla finestra dei ritardi di Barlett; ovvero

$$\theta(r) = \begin{cases} \left(1 - \frac{|r|}{M}\right)^q & |r| \leq M \\ 0 & |r| > M \end{cases}$$

In particolare, Parzen, suggerì il caso $q=2$; per questo valore la stima della densità spettrale non assume valori negativi.

La finestra dei ritardi di Parzen può essere costruita partendo dalla finestra dei ritardi di Barlett con parametro $M/2$ (M pari, come già detto). Questo spiega perché la finestra spettrale di Parzen ha la stessa forma della finestra spettrale di Barlett al quadrato con parametro $M/2$. Ciò, peraltro assicura che la stima della densità spettrale assuma valori non negativi.

Le finestre che abbiamo esaminato, si chiamano finestre con parametro di scala. Questo perché, come è facilmente verificabile, il parametro M agisce alle variazioni comprimendo o espandendo la finestra dei ritardi $\theta(r)$; quindi M si comporta da parametro di scala. Le finestre con parametro di scala possono essere rappresentate come $\theta(r) = \xi(r/M)$, dove $\xi(u)$ è una determinata funzione continua e pari di u , con $\xi(0)=1$. La specifica forma della

funzione $\xi(u)$ per le finestre da noi esaminate è sufficientemente ovvia.

La funzione $\xi(u)$, è stata inventata e chiamata da Parzen “generatore di finestra dei ritardi”. Come è facilmente intuibile, la trasformata di Fourier di questa funzione è chiamata “generatore di finestra spettrale” ed è esprimibile come

$$\Xi(\lambda) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \xi(u) e^{-i\lambda u} du$$

La finestra spettrale $W(\lambda)$ che corrisponde alla finestra dei ritardi $\theta(r)$, può essere espressa in funzione di $\Xi(\lambda)$ scrivendo

$$\begin{aligned} W(\lambda) &= \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \theta(r) e^{-i\lambda r} \\ &= M \left\{ \frac{1}{2\pi M} \sum_{r=-\infty}^{+\infty} \xi(r/M) e^{-i(r/M)M\lambda} \right\} \end{aligned}$$

Per M grande la sommatoria qui sopra può essere approssimata dall'integrale in $\Xi(\lambda)$, pertanto possiamo scrivere

$$W(\lambda) \sim M \cdot \Xi(M\lambda)$$

Da quest'ultima espressione possiamo derivare la forma di $\Xi(\lambda)$, conoscendo la finestra spettrale. Per esempio la forma della

finestra di Barlett, che abbiamo in precedenza derivato, è esprimibile come

$$W(\lambda) = \frac{1}{2\pi M} \left[\frac{\sin(\lambda M / 2)}{\sin(\lambda / 2)} \right]^2 = M \tilde{\Xi}(M\lambda)$$

con

$$\tilde{\Xi}(\lambda) = \frac{1}{2\pi M^2} \left[\frac{\sin(\lambda / 2)}{\sin(\lambda / 2M)} \right]^2 \sim \frac{1}{2\pi} \left[\frac{\sin(\lambda / 2)}{\lambda / 2} \right]^2 \quad \text{per } M \text{ grande.}$$

Punto di troncamento, larghezza di banda e varianza

La qualità di una stima spettrale è data dall'aspetto della finestra spettrale e dalla larghezza di banda della stessa. Quest'ultima è determinata dalla scelta del punto di troncamento. Un M piccolo, genera una stima a bassa varianza, ma aumenta la tendenziosità, ovvero l'errore sistematico. Del rapporto tra M e la varianza ne parleremo fra poco. E', invece, importante adesso far rilevare che se M è troppo piccolo, le principali caratteristiche della densità spettrale possono venire troppo spianate e quindi eliminate, mentre se M è troppo grande l'andamento dello stimatore $\hat{f}(\lambda)$ tende a diventare simile a quello del periodogramma e quindi a presentare oscillazioni erratiche.

Come dicevamo prima, per effettuare una buona stima spettrale non dobbiamo solamente preoccuparci di trovare la finestra spettrale con l'aspetto più desiderabile (la "window carpentry")

come l' ha chiamata Tukey), ma dobbiamo anche interessarci alla ricerca della larghezza di banda per quella finestra. Il problema è che per ogni data finestra non esiste un singolo criterio di scelta della larghezza di banda ottimale. Vi sono molte definizioni di larghezza di banda, le più famose di Parzen, Jenkins, Grenander e Brillinger. Importanti considerazioni possiamo trarre dalla seguente identità:

$$\text{Varianza} \times \text{Larghezza di banda} = \text{costante}$$

Quindi, in generale, l'effetto di incrementare il parametro M è di aumentare la varianza e ridurre la distorsione, mentre decrementando M diminuisce la varianza e cresce la distorsione.

Da non dimenticare il principio di incertezza di Grenander, secondo il quale "attendibilità e risolubilità sono antagoniste". E' cioè necessario un compromesso tra stabilità ed elevata capacità risolutiva. Affinché ciò avvenga possiamo procedere come segue. Innanzi tutto scegliamo una finestra spettrale adatta al modello. Poi, calcoliamo inizialmente la stima spettrale usando un'elevata larghezza di banda, successivamente ricalcoliamo la stima usando gradualmente larghezze di banda più piccole e continuiamo fino a quando non otteniamo l'attendibilità e la risolubilità che ci siamo prefissati. Questa procedura è conosciuta come chiusura della finestra.

E' molto importante sottolineare che alcuni segnali non hanno, ovviamente, un'interpretazione "fisica" del concetto di larghezza

di banda (Cosa che spesso accade, invece, nei fenomeni deterministici), quindi spesso bisogna adottare una procedura basata su prove ripetute e verifica empirica degli errori commessi. Ciò può sembrare poco scientifico, ma bisogna considerare che l'analisi spettrale è lo standard per molte elaborazioni dei segnali raccolti ed è certamente uno dei maggiori strumenti di analisi di questo tipo di dati.

Nell'uso delle finestre spettrali giocano un importante ruolo i cosiddetti gradi di libertà equivalenti calcolati secondo la relazione

$$v = \frac{2N}{M \int_{-\infty}^{+\infty} \xi^2(u) du}$$

Da questa relazione troviamo i gradi di libertà equivalenti, per i differenti tipi di finestra spettrale, esposti nella Tabella 15.1.

Stima	Gradi di libertà equivalente (v)
Periodogramma troncato	N/M
Finestra di Barlett	$3N/M$
Finestra di Tukey - Hamming	$2 \cdot 5164(N/M)$
Finestra di Tukey - Hanning	$8N/3M$
Finestra di Parzen	$3 \cdot 708614(N/M)$

Tabella 15.1. Calcolo dei gradi di libertà equivalente.

Jenkins e Watts, hanno fornito un'importante risultato per la stima della densità spettrale, $f(\lambda)$. Hanno, cioè dimostrato che

$$v \cdot \frac{\hat{f}(\lambda)}{f(\lambda)} \sim \chi_v^2$$

dove v sono gli equivalenti gradi di libertà, di cui sopra. Possiamo, pertanto, scrivere

$$\Pr \left\{ \chi^2(v, 1-\alpha/2) < v \cdot \frac{\hat{f}(\lambda)}{f(\lambda)} < \chi^2(v, \alpha/2) \right\} = 1-\alpha$$

ovvero, l'intervallo di confidenza per la densità spettrale $\hat{f}(\lambda)$ al livello $100(1-\alpha)\%$ è

$$\frac{v \cdot \hat{f}(\lambda)}{\chi^2(v, \alpha/2)} < f(\lambda) < \frac{v \cdot \hat{f}(\lambda)}{\chi^2(v, 1-\alpha/2)}$$

Consideriamo adesso un'altra interessante metodologia: il metodo Multitaper. Esso offre diverse caratteristiche interessanti tra cui un'alta risoluzione e varianza delle stime, che sono statisticamente indipendenti dalla potenza spettrale (oscillazioni piccole in ampiezza possono avere un livello di significatività alto). Alcune cautele sono sempre necessarie, infatti il livello di confidenza

statistica valutato per questo metodo, è spesso più ottimistico rispetto a quello dato per i metodi classici.

Lo scopo di questo metodo spettrale non parametrico (anche detto metodo di Thomson) è quello di calcolare un insieme di stime indipendenti e significative dello spettro di potenza, per ottenere una migliore e più affidabile stima di quanto non sia possibile con i metodi della singola finestra, in presenza di una serie cronologica finita. Un insieme di finestre rastremate ottimali è calcolato in modo che questa tecnica di stima spettrale sia meno euristica di quelle tradizionali (e.g. Blackman e Tukey).

Il metodo Multitaper (MTM) di Thomson è stato applicato a vari campi della geofisica, compresa la scienza della Terra (Lindberg, 1986; Park et al., 1987), la geofisica (Lanzerotti et al., 1986), la climatologia sulle scale temporali interdecennali e secolari (Kuo et al., 1990; Ghil e Vautard, 1991; Mann et al., 1995) e la paleoclimatologia, su dati relativi agli anelli di accrescimento degli alberi (Thomson, 1990), dati ricavati da carote marine (Thomson, 1990; Berger et al., 1991; Park e Maasch, 1993) e dati ricavati da carote di ghiaccio (Yiou et al., 1991, 1994).

Supponiamo che $X(t)$, sia un segnale sinusoidale di frequenza angolare $\hat{\lambda}$, per esempio $x(t) = \eta e^{i\hat{\lambda}t} + \varepsilon_t$, per $t = 0, \dots, N-1$. Supponiamo poi che sia $\lambda(t), t = 0, \dots, N-1$ una finestra di lunghezza N e $y(\lambda)$ la trasformata di Fourier discreta della serie finestrata $X(t)\lambda(t)$ dato da

$$y(\lambda) = \sum_{t=0}^{N-1} x(t)\lambda(t)e^{-i\lambda t}$$

dove, in questo caso, $\lambda = \frac{k}{N}$ è la frequenza angolare, con $k = 0, \dots, N-1$.

Lo scopo primario della finestra rastremata è di minimizzare le perdite spettrali, ovvero di minimizzare il contributo al di fuori di una determinata larghezza di banda $(\hat{\lambda} - \Lambda, \hat{\lambda} + \Lambda)$ della serie finestrata con frequenza angolare $\hat{\lambda}$, dove $|y(\lambda)|^2$ è una stima della potenza del segnale. Pertanto, la serie finestrata $X(t)\lambda(t)$ dovrebbe mantenere quanto più possibile nell'intervallo $(\hat{\lambda} - \Lambda, \hat{\lambda} + \Lambda)$, relativamente alla potenza totale che copre l'intera banda $(-\pi, \pi)$.

Si può scegliere una finestra $\lambda(t)$ tale da massimizzare il funzionale

$$f(\lambda) = \frac{\int_{\bar{\lambda}-\Lambda}^{\bar{\lambda}+\Lambda} |y(\lambda)|^2 d\lambda}{\int_{-\pi}^{\pi} |y(\lambda)|^2 d\lambda}$$

Nel massimizzare $f(\lambda)$ rispetto a $\lambda(t)$ sorge un classico problema Rayleigh-Ritz. Questo è equivalente ad un problema agli auto vettori come segue

$$A \cdot \lambda = \omega \lambda.$$

Dove A è una matrice Toeplitz. Le finestre ottenute da questi auto vettori sono chiamate successioni sferoidali prolate discrete (Discrete Prolate Spheroidal Sequences, DPSS) (Slepian, 1978). Per costruzione, esse sono ortogonali; quindi gli spettri associati forniscono stime indipendenti dello spettro reale. Se ordiniamo gli auto vettori mediante i loro corrispondenti auto valori

$$1 > \omega_1 > \dots > \omega_N > 0$$

allora l'espressione $1 - \omega_k$ dà la frazione del valore della k -esima finestra al di fuori dell'intervallo $(\hat{\lambda} - \Lambda, \hat{\lambda} + \Lambda)$. È evidente che i primi $K = \text{Int}(2N)\Lambda$ auto vettori (dove Int identifica la parte intera) hanno le migliori proprietà spettrali, dato che le relative frazioni di perdita $1 - \omega_k$, $k = 1, \dots, K$ sono molto vicine ad uno. Pertanto, si devono prendere al massimo $K = \text{Int}(2N)\Lambda$ finestre, per ottenere una stima spettrale efficiente. Slepian (1978), Thomson (1990) e Røgnvaldsson (1993) forniscono algoritmi efficienti per la computazione delle DPSS. Di seguito, indicheremo con $y_k(\lambda)$ la trasformata discreta di Fourier della serie finestrata $X(t)\lambda_k(t)$.

Una volta calcolate la finestre, si può stimare lo spettro di potenza totale, facendo la media dei singoli spettri di ciascuna serie finestrata. Quindi, se definiamo $\hat{S}_k(\lambda) = |y_k(\lambda)|^2$ il k -esimo auto spettro del segnale $X(t)$, la stima dello spettro totale ottenuta dalle prime K finestre è:

$$\hat{S}(\lambda) = \frac{1}{K} \sum_{k=0}^{K-1} \hat{S}_k(\lambda)$$

La risoluzione è quindi $\pm\lambda$, il che significa che le linee spettrali saranno rilevate mediante picchi o gobbe di larghezza 2λ . Il vantaggio di questa stima multi taper è che il processo di media su diverse stime indipendenti S_k dello spettro reale, attenuano le irregolarità spurie e quindi riducono la varianza della stima. La varianza può essere ulteriormente ridotta e le barre di errore possono essere calcolate applicando le cosiddette procedure «jackknifing» alle K stime.

Lo scopo dell'analisi armonica è di determinare le linee spettrali di un segnale periodico o quasi periodico e le relative ampiezze. La trasformata di Fourier di una serie periodica infinita fornisce una funzione del tipo di Dirac alla frequenza della serie, cioè un picco di altezza infinita. Una stima spettrale semplice (con il metodo di massima entropia, o con la trasformata di Fourier) fornisce informazioni indirette sull'ampiezza di una serie ad una data frequenza, attraverso l'area sotto il picco, che è approssimativamente costante, anche se l'altezza del picco è proporzionale al numero di punti nel segnale e tende ad infinito al crescere di questo numero. Quindi, lo scopo dell'analisi armonica è di definire un modo per avere un accesso diretto all'ampiezza di un'oscillazione periodica.

Una regressione nel senso dei minimi quadrati nel dominio delle frequenze, fornisce una stima $\hat{\eta}$ dell'ampiezza η

$$\hat{\eta}(\hat{\lambda}) = \frac{\sum_{k=0}^{K-1} U_k^*(0) y_k(\hat{\lambda})}{\sum_{k=0}^{K-1} |U_k(0)|^2}$$

dove l'asterisco indica una coniugazione complessa e $U_k(\lambda)$ è la trasformata discreta di Fourier di $\lambda_k(t)$ considerando le prime K successioni sferoidali prolate discrete.

Una misura di affidabilità statistica può essere ottenuta mediante un test di Fisher. Questo test è approssimativamente basato sul rapporto fra la varianza "spiegata" dal modello, e la varianza residua. Se si espande la varianza del modello, si trova che è la somma di due termini

$$\theta = |\hat{\eta}(\lambda)|^2 \sum_{k=0}^{K-1} |U_k(0)|^2$$

e

$$\psi = \sum_{k=0}^{K-1} |y_k(\hat{\lambda}) - \hat{\eta}(\hat{\lambda}) U_k(0)|^2$$

che rispettivamente sono il contributo alla varianza "spiegato" e residuo.

La variabile casuale

$$F(\hat{\lambda}) = (K-1) \frac{\theta}{\psi}$$

segue una legge Fisher-Snedecor con 2 e $2K-2$ gradi di libertà se il processo è un puro rumore bianco. L'ipotesi nulla corrisponde all'annullamento dell'ampiezza della corrispondente frequenza ($\eta=0$) e l'ipotesi alternativa è il non annullamento di tale ampiezza.

Con questo metodo si possono rilevare oscillazioni di piccola ampiezza in un segnale relativamente breve, con un alto grado di significatività statistica, oppure scartare un picco di grande ampiezza se fallisce al test F , perché il valore F non dipende in prima approssimazione dal valore di η . Questo significa in pratica che questo test è robusto nel caso di rumore bianco e dà ancora risultati ragionevolmente buoni con il rumore colorato, come quello presente in molti segnali campionati.

Una delle assunzioni principali di questa tecnica di analisi armonica è che il segnale deve fornire componenti periodiche separate. In caso contrario, uno spettro continuo (nel caso di rumore rosso o di un sistema caotico) sarà spezzato in linee spurie con frequenze arbitrarie e possibilmente con valori F alti. Questo è uno degli inconvenienti del metodo, che può essere parzialmente evitato se viene calcolato uno spettro grezzo di potenza e rilevate evidenze di linee spettrali.

È molto importante anche variare il parametro larghezza di banda ΔN ed il numero di finestre, per assicurarsi della stabilità delle stime di frequenza e di ampiezza. $\Delta N=4$ viene generalmente presa come scelta ΔN iniziale, che implica $K \leq 7$ finestre (Lindberg,

1986; Park et al., 1987). Successivamente la larghezza di banda può essere modificata in funzione della lunghezza delle serie temporali, e delle frequenze che devono essere investigate.

Per poter eseguire analisi di coerenza è utile approfondire il caso bivariato, eventualmente estendibile al caso multivariato.

$X_1(t), X_2(t), X_1(t+u), X_2(t+u)$ hanno una funzione di densità di probabilità congiunta. Con riguardo ai momenti otteniamo:

$$E[X_i(t)] = \mu_i, \quad i = 1, 2.$$

Le funzioni di auto covarianza (momento secondo) sono:

$$\gamma_{X_i X_i}(u) = E[(X_i(t) - \mu_i) \cdot (X_i(t) - \mu_i)]$$

e le funzioni di covarianza incrociata:

$$\gamma_{X_i X_j}(u) = E[(X_i(t) - \mu_i) \cdot (X_j(t+u) - \mu_j)] \quad i, j = 1, 2 \quad \text{con } i \neq j$$

Chiameremo la funzione di covarianza incrociata $C_{X_1 X_2}(u)$.

Le proprietà di queste funzioni inducono a:

$$\begin{cases} \gamma_{X_i X_i}(0) = \sigma_{X_i}^2 \\ \gamma_{X_i X_i}(u) = \gamma_{X_i X_i}(-u) \\ \gamma_{X_1 X_2}(u) = \gamma_{X_1 X_2}(-u) \end{cases}$$

La funzione di correlazione incrociata viene calcolata come:

$$\rho_{12}(u) = \frac{\gamma_{12}(u)}{\sqrt{\gamma_{11}(0)\gamma_{22}(0)}}, \quad \text{dove } \gamma_{ij} = \gamma_{x_i x_j}$$

che ha le seguenti proprietà:

$$|\rho_{12}(u)| \leq 1 \text{ e } \rho_{12}(u) = \rho_{21}(-u)$$

Lo stimatore delle funzioni di covarianza incrociata si chiama covarianza campionaria incrociata e può essere ottenuto come segue:

$$\begin{cases} c_{ij}(k) = \frac{1}{N} \sum_{t=1}^{N-k} (x_i(t) - \bar{x}_i)(x_j(t+k) - \bar{x}_j), & k \geq 0 \\ c_{ij}(-k) = c_{ij}(k) & k = 0 \end{cases}$$

$$\text{dove } \bar{x}_i = \frac{1}{N} \sum_{t=1}^N x_i(t).$$

15.4 Correlazione campionaria incrociata

Lo stimatore della funzione di correlazione incrociata si chiama correlazione campionaria incrociata e può essere ottenuto come segue:

$$r_{12}(k) = \frac{c_{12}(k)}{\sqrt{c_{11}(0) c_{22}(0)}}$$

Con le quattro covarianze, cioè le due autocovarianze e le due covarianze incrociate, si può costruire la matrice delle covarianze di intervallo u che indicheremo $\underline{\gamma}(u)$ e che risulta:

$$\underline{\gamma}(u) = \begin{bmatrix} \gamma_{X_1X_1}(u) & \gamma_{X_1X_2}(u) \\ \gamma_{X_2X_1}(u) & \gamma_{X_2X_2}(u) \end{bmatrix}$$

Analogamente le quattro correlazioni, cioè le due autocorrelazioni e le due correlazioni incrociate, danno luogo alla seguente matrice delle correlazioni di intervallo u

$$\underline{R}(u) = \begin{bmatrix} \rho_{X_1X_1}(u) & \rho_{X_1X_2}(u) \\ \rho_{X_2X_1}(u) & \rho_{X_2X_2}(u) \end{bmatrix}$$

Inoltre, con le quattro correlazioni possiamo scrivere quattro densità spettrali normalizzate espresse da:

- due autospettri

$$h_{X_1X_2}(w) = \frac{1}{2\pi} \sum_{-\infty}^{\infty} \rho_{X_1X_2}(u) e^{-iwu}$$

- due spettri incrociati

$$h_{X_2X_1}(w) = \frac{1}{2\pi} \sum_{-\infty}^{\infty} \rho_{X_2X_1}(u) e^{-i w u}$$

dove abbiamo usato la trasformata esponenziale di Fourier, in quanto si tratta di grandezze complesse.

Anche questi quattro spettri si possono riunire in una matrice spettrale del tipo

$$\underline{h}(w) = \begin{bmatrix} h_{X_1X_1}(w) & h_{X_1X_2}(w) \\ h_{X_2X_1}(w) & h_{X_2X_2}(w) \end{bmatrix}$$

Se invece di usare le correlazioni utilizzassimo le covarianze $\gamma(u)$, così come abbiamo fatto nel caso univariato, otterremmo quattro densità spettrali non normalizzate, come ad esempio la

$$f_{X_1X_2}(w) = \frac{1}{2\pi} \sum_{-\infty}^{\infty} \gamma_{X_1X_2}(u) e^{-i w u}$$

In modo analogo si calcolano le altre tre densità spettrali non normalizzate e l'insieme delle quattro densità così ottenute danno luogo, analogamente a prima, ad una matrice delle densità spettrali non normalizzate, che indicheremo con $\underline{f}(w)$ e chiameremo densità

spettrale incrociata che da qui in avanti sarà intesa non normalizzata

$$\underline{f}(w) = \begin{bmatrix} f_{X_1X_1}(w) & f_{X_1X_2}(w) \\ f_{X_2X_1}(w) & f_{X_2X_2}(w) \end{bmatrix}$$

La funzione di densità spettrale incrociata non è, in generale, reale. Questo a causa della funzione di covarianza incrociata che, di norma, non è simmetrica.

Ne consegue che la densità spettrale incrociata si può scrivere:

$$f_{X_1X_2}(w) = co_{X_1X_2}(w) - i q_{X_1X_2}(w)$$

dove

- $co_{X_1X_2}(w)$ indica la parte reale e si chiama co-spettro
- $q_{X_1X_2}(w)$ indica la parte immaginaria e si chiama spettro di quadratura.

E' facile dimostrare che $co_{X_1X_2}(w)$ è una funzione pari di w , mentre $q_{X_1X_2}(w)$ è una funzione dispari.

Come è noto, le grandezze complesse, come le densità spettrali incrociate, si possono anche esprimere in forma polare e, nel caso che stiamo considerando, acquistano particolare interesse.

La densità spettrale incrociata, esposta in forma polare, si scrive

$$f_{X_1X_2}(w) = \alpha_{X_1X_2} \cdot e^{i\phi_{X_1X_2}(w)}$$

dove

$$\alpha_{X_1X_2}(w) = |f_{X_1X_2}(w)| = \sqrt{co_{X_1X_2}^2(w) + q_{X_1X_2}^2}$$

è nota sotto il nome di spettro dell'ampiezza di fase, mentre

$$\tan[\phi_{X_1X_2}(w)] = -\frac{q_{X_1X_2}(w)}{co_{X_1X_2}(w)}$$

si chiama spettro di fase. Si può dimostrare che

$$\alpha_{X_1X_2}(w) \leq \sqrt{f_{X_1X_1}(w) \cdot f_{X_2X_2}(w)}$$

per cui, così come si fa per il calcolo del coefficiente di correlazione lineare, si può standardizzare l'ampiezza incrociata in modo che essa assuma solamente valori compresi nell'intervallo $[0,1]$. Ciò si ottiene ponendo

$$p_{X_1X_2} = \frac{\alpha_{X_1X_2}(w)}{\sqrt{f_{X_1X_1}(w) \cdot f_{X_2X_2}(w)}}$$

e questa nuova funzione si chiama coerenza spettrale e soddisfa alle condizioni

$$p_{X_1X_2}(w) = p_{X_2X_1}(w) = p_{X_1X_2}(-X_1)$$

Questa funzione, sicuramente la più interessante tra quelle viste finora, si può interpretare come un coefficiente di correlazione lineare nel dominio frequenziale: essa, infatti, rappresenta i valori assoluti della correlazione fra le ampiezze delle sinusoidi presenti nei due processi, per la frequenza w .

E' inoltre interessante rilevare che la coerenza $p_{X_1X_2}(w)$ è insensibile alle trasformazioni lineari che si possono effettuare sui due processi $X_1(t)$ e $X_2(t)$ e questa proprietà è importante nello studio delle interrelazioni fra due segnali psicofisiologici. Può infatti accadere che il correlogramma stimato, relativo a due realizzazioni di processi stocastici fra loro indipendenti, assuma, invece, valori elevati a causa della struttura interna di ciascuno dei due processi, come dimostrato da Jenkins e Watts (1968, pag. 338). E' allora consigliabile, volendo studiare le interrelazioni fra i due segnali del dominio temporale, filtrarle in modo che ciascuna diventi la realizzazione di un processo a rumore bianco.

Una volta stimata la funzione di covarianza incrociata $C_{X_1X_2}(u)$ come visto in precedenza, possiamo stimare la funzione della densità spettrale incrociata $f_{X_1X_2}(w)$ utilizzando le finestre spettrali di cui si è parlato nel caso univariato.

Si ha allora come stima di $f_{X_1X_2}(w)$ la relazione:

$$\hat{f}_{X_1 X_2}(w) = \frac{1}{2\pi} \sum_{u=-M+1}^{M-1} W(u) C_{X_1 X_2}(u) e^{-iwu}$$

Se da questa stima, espressa in forma complessa, si separano la parte reale da quella immaginaria otteniamo la stima del co-spettro

$$\hat{c}_{X_1 X_2}(w) = \frac{1}{2\pi} \left\{ C_{X_1 X_2}(0) + \sum_{u=1}^{M-1} W(u) [C_{X_1 X_2}(u) + C_{X_1 X_2}(-u)] \cdot \cos wu \right\}$$

e quella dello spettro di quadratura

$$\hat{q}_{X_1 X_2}(w) = \frac{1}{2\pi} \sum_{u=1}^{M-1} W(u) [C_{X_1 X_2}(u) - C_{X_1 X_2}(-u)] \cdot \sin wu$$

Queste due stime consentono così di stimare

- $\hat{\alpha}_{X_1 X_2}(w) = \sqrt{\hat{C}_{X_1 X_2}^2(w) + \hat{q}_{X_1 X_2}^2(w)}$
- $\tan[\hat{\phi}_{X_1 X_2}(w)] = -\frac{\hat{q}_{X_1 X_2}(w)}{\hat{C}_{X_1 X_2}(w)}$

Da tutte le relazioni precedenti, si ottiene la stima della funzione di coerenza spettrale che risulta:

$$\hat{p}_{X_1 X_2} = \frac{\hat{\alpha}_{X_1 X_2}(w)}{\sqrt{\hat{f}_{X_1 X_1}(w) \cdot \hat{f}_{X_2 X_2}(w)}}$$

dove $\hat{f}_{X_1 X_1}(w)$ e $\hat{f}_{X_2 X_2}(w)$ sono state ottenute anch'esse col metodo delle finestre spettrali.

Qualche relazione trigonometrica

Per lo studio della serie di Fourier e delle trasformate di Fourier si utilizzano diverse relazioni che legano tra di loro le funzioni trigonometriche seno e coseno ed altre che esprimono queste due funzioni mediante l'unità immaginaria i . Nella tabella seguente ne presento alcune.

$\sin(x + y) = \sin x \cos y + \cos x \sin y$	$\cos(x + y) = \cos x \cos y - \sin x \sin y$
$\sin(x - y) = \sin x \cos y - \cos x \sin y$	$\cos(x - y) = \cos x \cos y + \sin x \sin y$
$\sin 2x = 2 \sin x \cos x$	$\cos 2x = \cos^2 x - \sin^2 x$
$\sin^2 x + \cos^2 x = 1$	$e^{\pm ix} = \cos x \pm i \sin x$
$\sin x = \frac{e^{ix} - e^{-ix}}{2i}$	$\cos x = \frac{e^{ix} + e^{-ix}}{2}$
$\sum_{t=1}^n \cos xt = \frac{\cos[x(n+1)/2] \sin(xn/2)}{\sin(x/2)}$	$\sum_{t=1}^n \sin xt = \frac{\sin[x(n+1)/2] \sin(xn/2)}{\sin(x/2)}$
$\sum_{t=1}^n e^{ixt} = \sum_{t=1}^n \cos xt + i \sum_{t=1}^n \sin xt$	$\sin x \sin y = \frac{1}{2} [\cos(x - y) - \cos(x + y)]$
$\cos x \cos y = \frac{1}{2} [\cos(x - y) + \cos(x + y)]$	$\sin x \cos y = \frac{1}{2} [\sin(x - y) + \sin(x + y)]$

Tabella 15.2. Relazioni trigonometriche.

PARTE III

MODELLI COMPUTAZIONALI

INTRODUZIONE AI SISTEMI COMPLESSI

Intorno a noi esistono molti sistemi complicati, costituiti da parecchi elementi che sono difficili da mappare e quasi impossibili da capire. È comunque necessario provare a specificare quali sono le differenze tra sistemi complicati e sistemi semplici.

Un sistema complicato è composto da elementi indipendenti l'uno dall'altro. Così, se rimuoviamo un elemento da questo sistema, rendiamo questo sistema più semplice ma, in sostanza, non cambiamo il comportamento generale del sistema stesso.

In un sistema complesso le interdipendenze tra elementi sono essenziali per formare "l'intero" e questo non esisterà più rimuovendo elementi da esso.

Pertanto, la complessità è una proprietà del sistema, la "comlicatezza" non lo è. Un sistema complesso non esiste più rimuovendo un elemento che lo costituisce, invece un sistema

complicato continua a esistere, anche se lievemente differente. I mondi complessi sono irriducibili.

Il comportamento nella maggior parte dei sistemi complessi deriva dall'interazione delle parti che lo compongono a livello microscopico. Come vedremo successivamente, questa proprietà è chiamata "emergenza" ed è considerata una sorta di potente forza organizzatrice.

In tal senso potremmo dire che un sistema complesso si compone di parti interconnesse e "intrecciate".

Per capire il comportamento dei sistemi complessi dobbiamo analizzare sia il comportamento dei singoli elementi, sia le connessioni tra essi, nel processo di formazione del sistema stesso.

Il problema è che è possibile capire "l'intero" solo analizzando ogni singolo elemento, ma questi singoli elementi possono essere descritti solo in relazione ad altri elementi.

Per questo motivo un sistema complesso è difficile da analizzare e comprendere.

Potremmo definire un sistema complesso come "non facile da comprendere o analizzare".

È possibile articolare questi concetti sulla natura dei sistemi complessi da un punto di vista quantitativo.

Ad oggi non esiste un'unica accettata definizione di sistema complesso.

Un numero speciale sui sistemi complessi della rivista *Science* [Vol 284, Issue 5411, Pages 1-212], contiene le seguenti definizioni:

- *A complex system is one in which there are multiple interactions between many different components (D. Rind);*
- *Complex systems are systems in process that constantly evolve and unfold over time (W. Brian Arthur);*
- *A complex system is a highly structured system, which shows structure with variations (N. Goldenfeld and Kadanoff);*
- *A complex system is one whose evolution is very sensitive to initial conditions or to small perturbations, one in which the number of independent interacting components is large, or one in which there are multiple pathways by which the system can evolve (Whitesides and Ismagilov);*
- *A complex system is one that by design or function or both is difficult to understand and verify (Weng, Bhalla and Iyengar).*

Possiamo provare a definire le proprietà di un sistema complesso secondo le seguenti caratteristiche degli elementi che lo compongono:

- numerosità;
- interazioni e loro forza;
- formazione/operatività e loro scala temporale;
- diversità/variabilità;
- ambiente e loro interazioni con esso;
- attività e loro obiettivi;

In tal modo è possibile quantificare le proprietà di un sistema complesso.

Per quantificare le ultime tre caratteristiche della lista, abbiamo bisogno di metodi per conteggiare le varie possibilità. Questa problematica è fondamentale per una "complessità quantitativa".

16.1 Emergenza e complessità

Un tema centrale negli studi sui sistemi complessi e sulle loro applicazioni è la comprensione dei comportamenti emergenti nello stesso.

Possiamo organizzare le proprietà dei sistemi complessi seguendo due differenti approcci.

Il primo approccio consiste nel considerare le relazioni tra le parti e "l'intero". Dal momento che sappiamo che un sistema complesso ha la proprietà di essere – per l'appunto – "complesso", possiamo analizzare quali siano le relazioni tra la "complessità dell'intero" e la "complessità delle parti".

Il secondo approccio consiste nel provare a comprendere la descrizione delle relazioni del sistema. In tal modo è possibile definire quantitativamente l'idea di complessità. Proviamo a comprendere in cosa un sistema risulta complesso e cosa significa che un sistema è più complesso di un altro. Proveremo a riconoscere la complessità di un sistema, anche comparandolo con altri sistemi. Seguendo questa linea di pensiero possiamo

affermare che la complessità misura l'ammontare di informazione necessaria a descrivere il sistema.

Un'importante proprietà dei sistemi complessi è la non-linearità. Un'interazione non-lineare a livello micro, probabilmente causerà un'emergenza forte a livello macro (di sistema). Interazioni lineari causeranno un comportamento aggregato che non ha senso chiamare "emergenza". Comunque, essendo la complessità sull'"orlo del caos", potremmo avere un'emergenza debole generata da semplici (più lineari) interazioni.

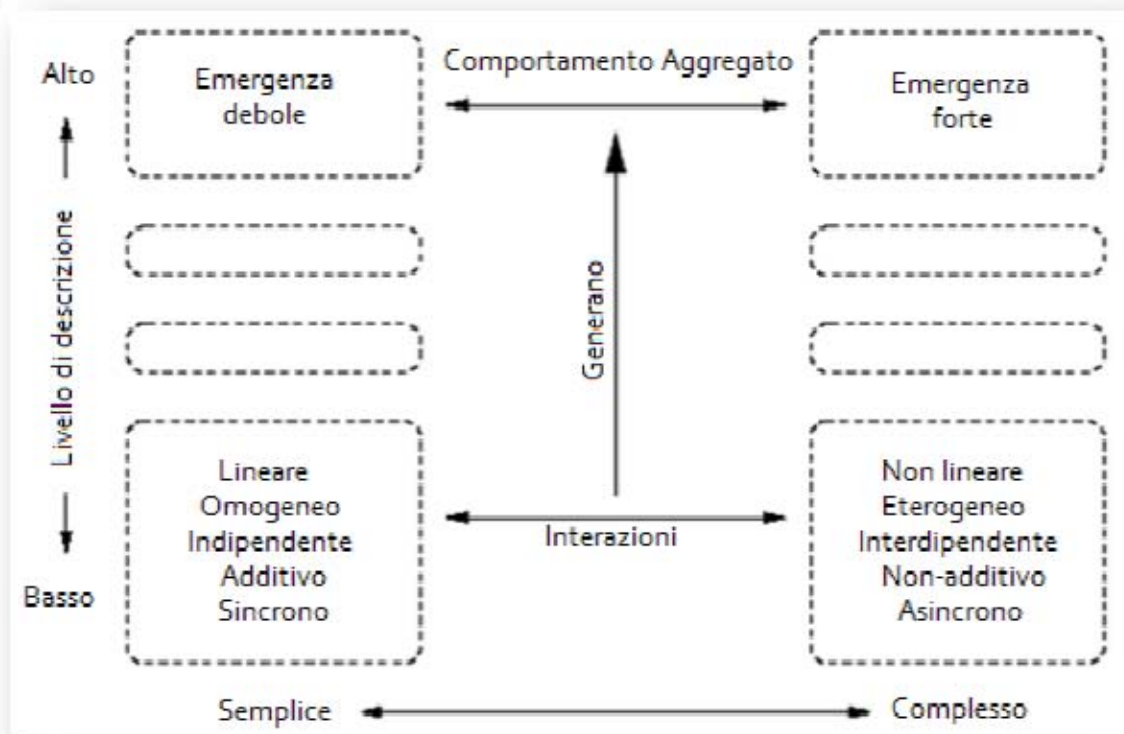


Figura 16.1. Emergenza debole ed emergenza forte.

Complessità emergente, semplicità emergente

Le parti sono spesso a loro volta sistemi complessi che interagendo formano altri sistemi complessi. Risulta abbastanza intuitivo che un insieme di sistemi complessi formi un sistema complesso, ma è possibile vedere che talvolta può accadere qualcosa di totalmente differente, lasciandoli interagire.

Ad esempio potremmo descrivere parti semplici che interagiscono, con un comportamento collettivo complesso. In tal caso diremo che abbiamo una *complessità emergente* e chiameremo, nella descrizione "dell'intero", *elementi* quelle parti semplici del sistema complesso.

Un esempio di complessità emergente è riscontrabile in ogni sistema complesso formato dall'interazione tra atomi (parti semplici).

Una differente situazione è quella in cui le parti che interagiscono sono sistemi complessi. In tal caso se accade che il comportamento collettivo emergente è semplice, avremo una *semplicità emergente*. In questo sistema "l'intero" ha un comportamento con complessità differente rispetto alle sue parti. Si pensi ad esempio alla Terra che orbita intorno al Sole; il suo comportamento è effettivamente semplice, anche se composto da sistemi complessi (quali gli esseri viventi): a livello micro il sistema si comporta in vari modi complessi, ma a livello macro tutti i sistemi complessi che lo compongono diventano irrilevanti.

Il comportamento collettivo di un sistema complesso potrebbe apparire solo in qualche piccola parte del sistema: in questo caso diremo che il sistema ha un'emergenza (o delle emergenze) locale (o locali). Se invece, il sistema, nel suo intero, ha comportamenti collettivi, allora diremo che il sistema ha un'emergenza globale. Quest'ultima particolare forma di emergenza è molto significativa nello studio di un sistema complesso.

Se consideriamo un intero sistema con proprietà emergenti locali, rimuovendo una piccola parte avremo modesti cambiamenti nelle proprietà sia dell'intero sistema, sia delle parti più piccole. D'altra parte in un sistema con proprietà emergenti globali avremo due differenti situazioni: una in cui le piccole parti risultano isolate e l'altra in cui le piccole parti rappresentano una parte "dell'intero". Pensando "all'intero" è utile riconoscere quei sistemi che mostrano proprietà di emergenza globale derivanti da parti interdipendenti.

Parliamo di parti interdipendenti per sottolineare l'influenza che ogni elemento ha sugli altri. Questa è una proprietà rilevante dei sistemi dinamici. Anche considerando interazioni forti non riusciamo a calcolare qualcosa come l'interdipendenza nel comportamento. Potremmo, ad esempio, pensare ai solidi semplici, in termini di proprietà macroscopiche.

Pertanto l'idea è di descrivere i sistemi complessi tramite un processo di estrazione di una parte di esso.

Rimuovendo una parte dal sistema, si potranno avere due casi: o le proprietà degli elementi di quella parte ne sono influenzate, ma gli altri elementi non lo sono, o le proprietà degli altri elementi, che

non sono nella parte rimossa, ne sono influenzate. In quest'ultimo caso avremo probabilmente un sistema complesso. In tale sistema il comportamento delle parti interdipendenti genera un comportamento collettivo. Questa idea generale sarà certamente più chiara introducendo una misura quantitativa della complessità.

16.2 Misurare la complessità tramite l'entropia

Come già abbiamo intuito è necessario trovare una misura quantitativa della complessità, così da potere almeno comparare due sistemi e definire quale sia più complesso tra i due sistemi stessi.

Sono stati proposti differenti approcci, come segue:

1. tramite osservazione empirica, è necessaria una scala di valutazione soggettiva;
2. contando i diversi elementi del sistema, definendo come distinguere le varie parti;
3. con una misura di dimensionalità del sistema;
4. considerando il numero di parametri necessari a controllare il sistema;
5. usando una descrizione minima, formalizzata in un linguaggio;
6. misurando l'informazione contenuta nel sistema, è comunque necessario identificarla;
7. con un metodo "minimal generator";

8. con un algoritmo "minimum energy".

Ovviamente non troveremo una misura universale, ma possiamo cercare un metodo di misurazione utile in particolari casi, anche se con qualche limitazione.

A tale scopo andiamo a definire una misura dell'informazione in termini di probabilità p , ovvero avremo una misura definita $I(p)$ con molte proprietà:

- $I(p) \geq 0$, cioè l'informazione è non-negativa;
- $I(p_1 * p_2) = I(p_1) + I(p_2)$, ovvero se si verifica la probabilità congiunta di due eventi indipendenti, allora l'informazione ottenuta è la somma di queste due informazioni;
- $I(p)$, la misura dell'informazione, deve essere una funzione monotona e continua della probabilità p ;
- $I(1) = 0$, ovvero per un evento a probabilità 1 (un evento certo) non otterremo informazioni.

Perciò,

$$I(p^2) = I(p * p) = I(p) + I(p) = 2 * I(p)$$

per induzione,

$$I(p^n) = n * I(p)$$

$$I(p) = I\left(\left(p^{\frac{1}{m}}\right)^m\right) = m * I\left(p^{\frac{1}{m}}\right)$$

perciò,

$$I\left(p^{\frac{1}{m}}\right) = \frac{1}{m} * I(p)$$

in termini generali,

$$I\left(p^{\frac{n}{m}}\right) = \frac{n}{m} * I(p)$$

e grazie alla continuità,

$$I(p^a) = a * I(p)$$

per $0 < p \leq 1$ e $a > 0, a \in \mathbb{R}$.

Dunque otteniamo,

$$I(p) = -\log_h(p) = \log_h\left(\frac{1}{p}\right)$$

per una costante $h > 0$. Questa base (h) definisce le unità che stiamo usando. Per differenti valori di h , nella base del logaritmo della misura di informazione $I(p)$, otterremo:

- \log_2 unità sono dette bits, derivando il termine da binary (binario);

- \log_3 unità sono dette trits, derivando il termine da trinary (trinario);
- $\log_e (= \ln)$ nats, derivando il termine da logaritmo naturale;
- \log_{10} unità sono dette Hartleys, derivando il termine dal nome di un Ricercatore in questo campo.

Genericamente parlando, l'ammontare di informazione necessaria a descrivere un sistema, definisce la complessità del sistema stesso. Supponiamo di avere una sequenza $\{s_1, s_2, \dots, s_n\}$ fornita indipendentemente da una certa fonte, con probabilità $\{p_1, p_2, \dots, p_n\}$ rispettivamente. Noi siamo interessati alla quantità media pesata di informazione, ottenuta da ciascun s_i , per $i=1, \dots, n$. Considerando un lungo periodo (assumiamo N) avremo N osservazioni, certamente indipendenti e un'informazione totale, I , data da:

$$I = \sum_{i=1}^n (N * p_i) * \log\left(\frac{1}{p_i}\right)$$

ovvero, per ogni s_i ($i=1, \dots, n$),

$$\frac{I}{N} = \left(\frac{I}{N}\right) \sum_{i=1}^n (N * p_i) * \log\left(\frac{1}{p_i}\right) = \sum_{i=1}^n p_i * \log\left(\frac{1}{p_i}\right)$$

Siccome $\lim_{n \rightarrow 0} n * \log\left(\frac{1}{n}\right) = 0$, definiremo $p_i * \log\left(\frac{1}{p_i}\right) = 0$, se $p_i = 0$.

Abbiamo definito l'informazione come funzione della probabilità di eventi $\{p_1, p_2, \dots, p_n\}$. Chiamiamo questo insieme P e definiamo l'entropia di questa distribuzione come:

$$S(P) = \sum_{i=1}^n p_i * \log\left(\frac{1}{p_i}\right)$$

Generalizzando $S(P)$ nel continuo invece che nel discreto, otterremo:

$$S(P) = \int P(x) * \log\left(\frac{1}{P(x)}\right) dx.$$

Adesso proviamo a pensare all'entropia in termini di "valore atteso". Dal momento che stiamo parlando di probabilità, considerando la distribuzione discreta $\{p_1, p_2, \dots, p_n\}$, abbiamo $p_i \geq 0$ e $\sum_{i=1}^n p_i = 1$, dove considerando una distribuzione continua $P(x)$ avremo $P(x) \geq 0$ e $\int P(x)dx = 1$.

Perciò, definendo l'insieme $\{h_1, h_2, \dots, h_n\}$ come H , il valore atteso di H sarà:

$$E[H] = \sum_{i=1}^n h_i * p_i$$

per distribuzioni discrete $\{p_1, p_2, \dots, p_n\}$, o

$$E[H(x)] = \int H(x) P(x) dx$$

per distribuzioni continue $P(x)$.

Considerando i precedenti risultati sull'entropia e l'informazione, otteniamo:

$$S(P) = E[I(p)],$$

Ovvero, il valore atteso dell'informazione della nostra distribuzione è l'entropia della distribuzione di probabilità.

Un altro approccio per misurare la complessità è stato definito da A. Rényi. Prendiamo una distribuzione di probabilità, considerando momenti di ordine $\alpha \in \mathbb{R}$ della distribuzione:

$$S_\alpha = \frac{1}{\alpha - 1} \log \sum_i p_i^\alpha.$$

per $\alpha \rightarrow 1$,

$$S_1 = \sum_i p_i \log \left(\frac{1}{p_i} \right),$$

è l'entropia, come in precedenza definite. Possiamo perciò dire che, $\forall \alpha \in \mathbb{R}$, S_α rappresenta l'entropia generalizzata.

Consideriamo la dimensione di Hausdorff D (la cosiddetta "fractal dimension", dimensione frattale), definita:

$$D = \lim_{\beta \rightarrow 0} \frac{\log \sum_i p_i^0}{\log \left(\frac{1}{\beta} \right)} = \lim_{\beta \rightarrow 0} \frac{\log(N_\beta)}{\log \left(\frac{1}{\beta} \right)}.$$

Possiamo generalizzare la dimensione frattale, in questo modo:

$$D_\alpha = \lim_{\beta \rightarrow 0} \frac{1}{\alpha - 1} \frac{\log \sum_i p_i^\alpha}{\log(\beta)}.$$

Questa rappresenta la dimensione legata a un insieme di dati. Possiamo immaginarla come distribuita tra scatole di diametro β , con probabilità p_i di cadere nell' i -esima scatola. Esiste una dimensione $\forall \beta$.

Per $\beta = 0$ (con la convenzione che quando $p_i = 0$, $p_i^0 = 0$) otterremo $D_0 = D$, ovvero la dimensione frattale sopra spiegata.

Per esempio, considerando un intervallo $[0,1]$; diciamo $\beta_\varepsilon = \frac{1}{2^\varepsilon}$; perciò $N_{\beta_\varepsilon} = 2^\varepsilon$, e

$$D_0 = \lim_{\varepsilon \rightarrow 0} \frac{\log(2^\varepsilon)}{\log(2^\varepsilon)} = 1,$$

considerando $[0,1] \times [0,1]$, diciamo $\beta_\varepsilon = \frac{1}{2^\varepsilon}$; perciò $N_{\beta_\varepsilon} = 2^{2\varepsilon}$, e

$$D_0 = \lim_{\varepsilon \rightarrow 0} \frac{\log(2^{2\varepsilon})}{\log(2^\varepsilon)} = 2.$$

Possiamo fare alcune considerazioni. Prima di tutto, $\alpha_1 \leq \alpha_2 \Rightarrow D_{\alpha_1} \leq D_{\alpha_2}$.

Poi, quando un insieme con uguali probabilità $p_i=1/n$ è strettamente autosimilare, non prendiamo il limite per $\beta \rightarrow 0$,

$$D_\alpha = \frac{1}{\alpha - 1} \frac{\log\left(N * \left(\frac{1}{N}\right)^\alpha\right)}{\log(\beta)} = \frac{\log(N)}{\log\left(\frac{1}{\beta}\right)} = D_0, \quad \forall \alpha.$$

Questo accade, ad esempio, con l'insieme di Cantor.

Avremo $\beta_\varepsilon = \frac{1}{3^\varepsilon}$; perciò $N_{\beta_\varepsilon} = 2^\varepsilon$, e

$$D_0 = \lim_{\varepsilon \rightarrow \infty} \frac{\log(2^\varepsilon)}{\log(3^\varepsilon)} = \frac{\log(2)}{\log(3)} \approx 0.6309.$$

L'insieme di Cantor rappresenta un esempio di frattale.

Una considerazione finale:

- D_1 è detta "dimensione dell'informazione" (information dimension), infatti il numeratore è l'entropia della distribuzione

$$D_1 = \lim_{\beta \rightarrow 0} \frac{\sum_i p_i * \log\left(\frac{1}{p_i}\right)}{\log(\beta)}$$

- D_2 è detta "dimensione della correlazione" (correlation dimension), rappresenta la probabilità di prendere due elementi da un insieme avente una distanza massima di β .

$$D_2 = \lim_{\beta \rightarrow 0} \frac{\log \sum_i p_i^2}{\log(\beta)}.$$

16.3 Caratterizzazione

Consideriamo due differenti distribuzioni di probabilità:

$$P = \{p_1, p_2, \dots, p_n\} \text{ e } G = \{g_1, g_2, \dots, g_n\},$$

Con $p_i, g_i \geq 0$; $\sum_{i=1}^n p_i = 1$ e $\sum_{i=1}^n g_i = 1$. Perciò,

$$\sum_{i=1}^n g_i \ln\left(\frac{p_i}{g_i}\right) \leq \sum_{i=1}^n g_i \left(\frac{p_i}{g_i} - 1\right) = \sum_{i=1}^n (p_i - g_i) = \sum_{i=1}^n g_i - \sum_{i=1}^n p_i = 0,$$

con il segno di uguaglianza se $\forall i, p_i = g_i$.

Ovviamente ciò accade in quanto $\frac{\partial \ln(x)}{\partial x} = \frac{1}{x}$, perciò $y = x - 1$ rappresenta la tangente della funzione $\ln(x)$ per $x = 1$; essendo $\ln(x)$ una funzione concava verso il basso, $\ln(x) \leq x - 1$ con segno di uguaglianza solo se $x = 1$.

Consideriamo adesso la distribuzione di probabilità $F = \{f_1, f_2, \dots, f_n\}$.

$$\begin{aligned}
 S(F) - \log(n) &= \sum_{i=1}^n f_i \log\left(\frac{1}{f_i}\right) - \log(n) = \sum_{i=1}^n f_i \log\left(\frac{1}{f_i}\right) - \log(n) \sum_{i=1}^n f_i \\
 &= \sum_{i=1}^n f_i \log\left(\frac{1}{f_i}\right) - \sum_{i=1}^n f_i \log(n) = \sum_{i=1}^n f_i \left(\log\left(\frac{1}{f_i}\right) - \log(n)\right) \\
 &= \sum_{i=1}^n f_i \left(\log\left(\frac{1}{f_i}\right) + \log\left(\frac{1}{n}\right)\right) = \sum_{i=1}^n f_i \log\left(\frac{1/n}{f_i}\right) \leq 0,
 \end{aligned}$$

Con segno di uguaglianza solo se $f_i = \frac{1}{n}, \forall i$.

Perciò, usando la disuguaglianza di Gibbs troviamo la distribuzione f_i che massimizza l'entropia $S(F)$.

Dunque abbiamo

$$0 \leq S(F) \leq \log(n)$$

Avremo $S(F) = 0$ se $\exists! i: p_i = 1$ e $p_j = 0, \forall j \neq i$. Allora avremo $S(F) = \log(n)$ se $p_i = \frac{1}{n}, \forall i$.

Nel 1948 Claude Shannon crea le fondamenta della teoria della comunicazione, sviluppando una teoria matematica per i sistemi di comunicazione.

Un modello di base di Shannon si compone di tre parti: una fonte (che spedisce un messaggio), un canale (dove passa il messaggio), e un ricevente (che riceve il messaggio). In un modello più generale, Shannon, considera anche il rumore (disturbo) sul canale e un sistema di codifica e decodifica.

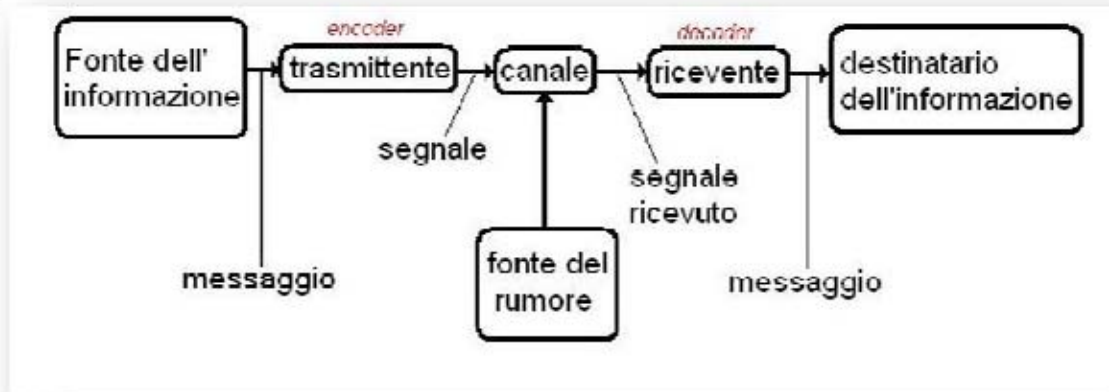


Figura 16.2. Il classico modello di Shannon.

Shannon, nella prima parte del suo lavoro, definisce un modello discreto, assumendo una sequenza di lettere $\{h_1, h_2, \dots, h_n\}$, spedite tramite un canale (disturbato da rumore) e poi codificate, dalla fonte. Successivamente il ricevente decodifica i simboli in informazioni.

È importante evidenziare che nella teoria di Shannon il processo di "spedizione delle informazioni" può essere visto come un processo di "trasmissione delle informazioni" del segnale da un punto a un altro o anche come "archiviazione delle informazioni" da un tempo t_0 a un tempo t_1 , con $t_1 > t_0$.

Assumendo che gli elementi attraverso il canale siano lettere, potremmo definire un alfabeto su questo canale. In generale possiamo considerare un insieme di simboli $\{r_1, r_2, \dots, r_n\}$, chiamato R ; consideriamo questo alfabeto A , la codifica è una funzione $e: A \rightarrow R^+$, dove R^+ rappresenta tutte le stringhe finite che

possono essere ottenute da R . Consideriamo adesso la disuguaglianza di McMillan Kraft:

$$K = \sum_{i=1}^n \frac{1}{\beta^{l_i}} \leq 1 \Leftrightarrow \exists! u: \exists \mathcal{L}(u) \in \{l_1, l_2, \dots, l_n\},$$

ovvero, abbiamo solo un codice con lunghezza $\{l_1, l_2, \dots, l_n\}$ sse $K \leq 1$ dove $l_i = |e(a_i)|, i = 1, 2, \dots, n;$ e $a_i \in A$.

Consideriamo adesso K^n :

$$k^n = \left[\sum_{i=1}^n \frac{1}{\beta^{l_i}} \right]^n = \sum_{i=1}^{nl} \frac{N_k}{\beta^k}$$

con $l = \text{Max} \{l_1, l_2, \dots, l_n\}$.

Ovviamente $N_k \leq \beta^k$, perciò

$$K^n \leq \sum_{k=n}^{nl} \frac{\beta^k}{\beta^k} = nl - n + 1 \leq nl,$$

Quindi, $K \leq 1$.

Adesso definiamo, per $0 < G_i \leq 1$, e $\sum_{i=1}^n G_i = 1$,

$$G_i = \frac{\beta^{-l_i}}{K}.$$

Applicando la disuguaglianza di Gibbs, data p_i la probabilità di osservare h_i , otterremo:

$$\sum_{i=1}^n p_i \log\left(\frac{G_i}{p_i}\right) \leq 0$$

O anche

$$\sum_{i=1}^n p_i \log\left(\frac{1}{p_i}\right) \leq \sum_{i=1}^n p_i \log\left(\frac{1}{G_i}\right).$$

Da notare che $S(B) = \sum_{i=1}^n p_i \log\left(\frac{1}{p_i}\right)$ rappresenta l'entropia della fonte, quindi:

$$\begin{aligned} S(B) &\leq \sum_{i=1}^n p_i [\log(K) - \log(\beta^{-l_i})] = \log(K) + \sum_{i=1}^n p_i l_i \log(\beta) \\ &\leq \log(\beta) \sum_{i=1}^n p_i l_i. \end{aligned}$$

Perciò, $S(B) \leq L \log(\beta)$, dove $L = \sum_{i=1}^n p_i l_i$.

Quindi, l'entropia fornisce un limite inferiore dell'efficienza del sistema di

codifica. Abbiamo dunque bisogno di creare un miglior modello di probabilità per migliorare questo schema.

Consideriamo $F = \{f_1, f_2, \dots, f_n\}$, contenente simboli in input e $G = \{g_1, g_2, \dots, g_m\}$, contenente simboli in output, dove può anche essere $m \neq n$.

Dal momento che non sappiamo quale simbolo f_i ha generato il simbolo g_j , la caratterizzazione del canale è data dall'insieme $\{P(f_i|g_j)\}$.

Adesso possiamo definire la *mutual information*:

$$I(f_i; g_j) = \log\left(\frac{1}{P(f_i)}\right) - \log\left(\frac{1}{P(f_i|g_j)}\right) = \log\left(\frac{P(f_i|g_j)}{P(f_i)}\right),$$

dove $P(f_i)$ è uno stimatore "a priori". Dunque:

- $I(f_i; g_j) = I(g_j; f_i)$;
- $I(f_i; g_j) = \log(P(f_i|g_j)) + I(f_i)$;
- $I(f_i; g_j) \leq I(f_i)$;
- $P(f_i; g_j) = P(f_i) * P(g_j) \Rightarrow I(f_i; g_j) = 0$.

Adesso siccome:

$$I(F; g_j) = \sum_i P(f_i|g_j) * I(f_i; g_j) = \sum_i P(f_i|g_j) * \log\left(\frac{P(f_i|g_j)}{P(f_i)}\right),$$

e

$$I(f_i; G) = \sum_j P(f_i|g_j) * \log\left(\frac{P(g_j|f_i)}{P(g_j)}\right),$$

avremo:

$$\begin{aligned}
 I(F; G) &= \sum_i P(f_i) * I(f_i; G) = \\
 &= \sum_i \sum_j P(f_i; g_j) * \log\left(\frac{P(f_i|g_j)}{P(f_i)P(g_j)}\right) = I(G; F).
 \end{aligned}$$

Dunque, $I(F; G) \geq 0$, $I(F; G) = 0 \Leftrightarrow P(F, G) = P(F) * P(G)$, e I è simmetrico in F e G ($I(F; G) = I(G; F)$). Inoltre,

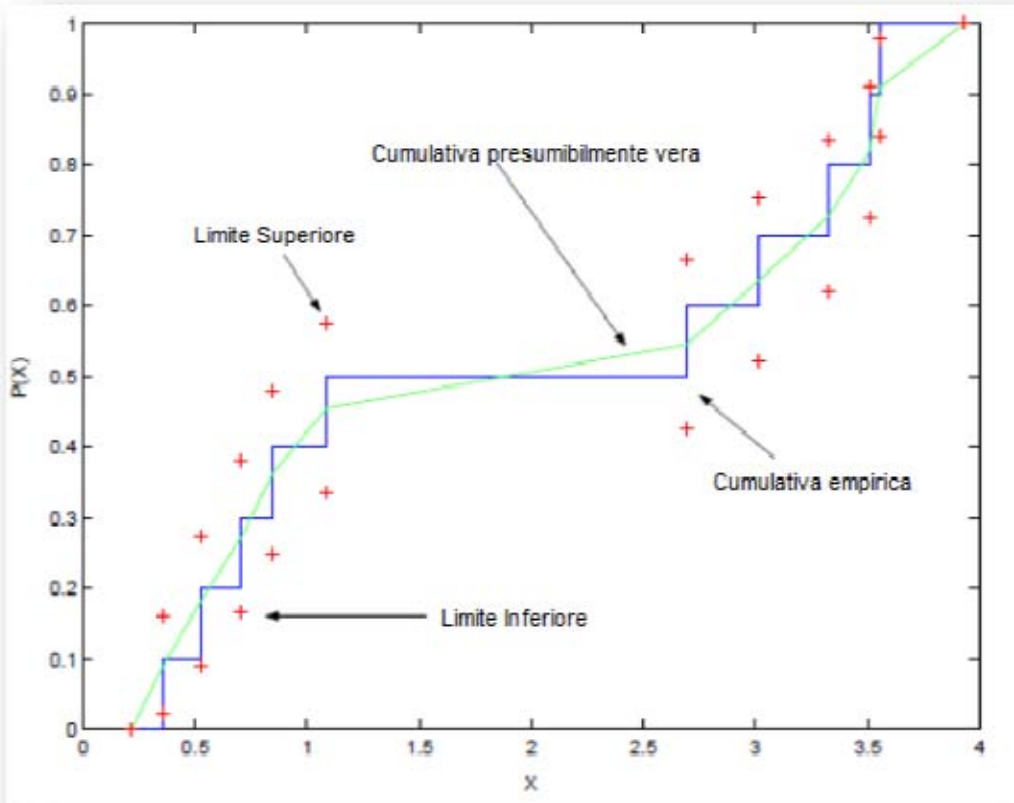


Figura 16.3. Limite inferiore e limite superiore, come descritto.

$$S(F) = \sum_{i=1}^n P(f_i) * \log\left(\frac{1}{P(f_i)}\right)$$

$$S(G) = \sum_{j=1}^m P(g_j) * \log\left(\frac{1}{P(g_j)}\right)$$

$$S(F|G) = \sum_{i=1}^n \sum_{j=1}^m P(f_i|g_j) * \log\left(\frac{1}{P(f_i|g_j)}\right)$$

$$S(F, G) = \sum_{i=1}^n \sum_{j=1}^m P(f_i, g_j) * \log\left(\frac{1}{P(f_i, g_j)}\right)$$

$$S(F, G) = S(F) + S(G|F) = S(G) + S(F|G)$$

$$I(F, G) = I(F) + I(G) - I(F, G) = I(F) - I(F|G) = I(G) - I(G|F) \geq 0,$$

Perciò, possiamo scrivere la mutual information $I(F, G)$ come una differenza tra l'entropia marginale e l'entropia condizionale.

Dunque, data la conoscenza di G , la diminuita incertezza di F rappresenta la mutual information. Per questo motivo chiamiamo $I(F, G)$ mutual information.

Adesso possiamo definire la "capacità del canale" (channel capacity):

$$C^{Max} = \text{Max}_{P(f)} I(F; G).$$

Adesso useremo un classico principio di massima entropia (maximum entropy principle, MEP).

Chiamiamo s_r ($r = 1, 2, \dots, K$) alcune caratteristiche a livello macroscopico. Assumiamo che queste siano associate a caratteristiche a livello microscopico tramite:

$$\sum_i f_i * s_i^{(r)} = s_r,$$

Vincolate da $f_i \geq 0$ e $\sum_i f_i = 1$: soggetti a tali vincoli dobbiamo massimizzare $\sum_i f_i * \log\left(\frac{1}{f_i}\right)$, l'entropia.

Quindi, una soluzione generale è:

$$f_i = \exp\left(-\lambda - \sum_r \lambda_r * s_i^{(r)}\right).$$

dove λ_r rappresenta il classico moltiplicatore di Lagrange.

Possiamo definire:

$$H(\lambda_1, \lambda_2, \dots, \lambda_K) = \sum_i \exp\left(-\sum_r \lambda_r * s_i^{(r)}\right)$$

Quindi $e^\lambda = H$ o $\lambda = \ln(H)$.

Un interessante applicazione all'economia è stata discussa da Adrian Dragulescu e Victor M. Yakovenko e Anirban Chakraborti e Bikas K. Chakraborti.

Consideriamo una semplice (e non realistica) economia con M (quantità di moneta) fissata e N (numero degli individui) fissato. Supponiamo che ognuno trasferisce un dollaro a un altro, selezionato casualmente. Non esistono situazioni debitorie (quando gli individui non hanno soldi). Esaminiamo la distribuzione di moneta nel lungo periodo, in termini di stabilità.

Definiamo $\{p_i\}$ la probabilità che un individuo ha una specifica quantità di moneta i , e n_i l'individuo che detiene i (unità di valuta). Quindi dobbiamo considerare i vincoli $\sum_i n_i * i = M$ e $\sum_i n_i = N$, e usando probabilità $p_i = \frac{n_i}{N}$, otterremo:

$$\sum_i p_i * i = \frac{M}{N}$$

e

$$\sum_i p_i = 1,$$

che è coerente con la definizione di probabilità. Applicando i moltiplicatori di Lagrange prima definiti, avremo:

$$L = \sum_i p_i \ln\left(\frac{1}{p_i}\right) - \lambda \left(\sum_i p_i * i - \frac{M}{N} \right) - \mu \left(\sum_i p_i - 1 \right)$$

Quindi,

$$\frac{\partial L}{\partial p_i} = -(1 + \ln(p_i)) - \lambda_i - \mu = 0.$$

Risolvendo per p_i , $\ln(p_i) = -\lambda_i - (1 + \mu)$; perciò $p_i = e^{-\lambda_0} e^{-\lambda_i}$, con $\lambda_0 \equiv 1 + \mu$.

Considerando i vincoli sopra, avremo:

$$1 = \sum_i p_i = \sum_i e^{-\lambda_0} e^{-\lambda_i} = e^{-\lambda_0} \sum_{i=0}^M e^{-\lambda_i}$$

e

$$\frac{M}{N} = \sum_i p_i * i = \sum_i e^{-\lambda_0} e^{-\lambda_i} * i = e^{-\lambda_0} \sum_{i=0}^M e^{-\lambda_i} * i.$$

Approssimativamente:

$$\sum_{i=0}^M e^{-\lambda_i} \approx \int_0^M e^{-\lambda x} dx \approx \frac{1}{\lambda} \quad \text{and} \quad \sum_{i=0}^M e^{-\lambda_i} * i \approx \int_0^M x e^{-\lambda x} dx \approx \frac{1}{\lambda^2},$$

per M grande. Quindi,

$$e^{\lambda_0} = \frac{1}{\lambda} \quad \text{and} \quad \frac{e^{\lambda_0 M}}{N} = \frac{1}{\lambda^2} \quad \therefore \quad \lambda = \frac{M}{N} = e^{-\lambda_0}.$$

Fissando $T = \frac{M}{N}$, otterremo:

$$p_i = e^{-\lambda_0} e^{-\lambda_i} = \frac{1}{T} e^{-\frac{i}{T}}.$$

Questa viene chiamata distribuzione di Boltzmann-Gibbs.

16.4 Sistemi adattativi complessi

Per dare una buona definizione di sistemi adattativi complessi, possiamo citare uno dei più importanti studiosi nel campo, considerato il padre degli algoritmi genetici, ovvero John Henry Holland:

"A complex adaptive system (CAS) is a dynamic network of many agents (which may represent cells, species, individuals, firms, nations) acting in parallel, constantly acting and reacting to what the other agents are doing. The control of a CAS tends to be highly dispersed and decentralized. If there is to be any coherent behavior in the system, it has to arise from competition and cooperation among the agents themselves. The overall behavior of

the system is the result of a huge number of decisions made every moment by many individual agents."

I sistemi sociali, gli esseri umani, I sistemi ecologici, il cervello, i mercati azionari, le imprese, sono solo alcuni esempi di sistemi adattativi complessi.

A livello macro un sistema adattativo complesso manifesta la complessità e le proprietà emergenti, come già esaminato nei paragrafi precedenti, anche grazie a caratteristiche quali adaptivity, Self-organization (processi di attrazione e repulsione), stigmergy, autocatalysis, syntropy, self-similarity e così via. Quindi un sistema adattativo complesso ha la capacità di adattarsi all'ambiente e ai suoi cambiamenti. Quando si presenta una perturbazione del sistema, un sistema adattativo complesso si riconfigura senza grosse perdite delle sue funzionalità intrinseche; ovviamente questo dipende dal grado di resilienza del sistema stesso.

Un'altra importante proprietà dei sistemi adattativi complessi è l'omeostasi, ovvero la capacità di un sistema di regolare le proprie variabili interne per tendere verso un equilibrio stabile. Per attuare tale regolazione, dove le interazioni non sono indipendenti, è necessario introdurre il concetto di feedback: di base, questa caratteristica, interviene nei sistemi modificando la dinamica dei sistemi stessi.

Possiamo avere due differenti tipi di feedback:

- feedback positivo, tramite il quale le variazioni del sistema vengono ampliate, portando il sistema a essere instabile;

- feedback negativo, tramite il quale le variazioni del sistema lo conducono verso l'omeostasi (in economia si dice lo stato stazionario).

Spesso nel campo dei sistemi adattativi complessi si parla di "orlo del caos" (edge of the chaos). Questa idea è stata originariamente sviluppata da Packard and Langton tramite degli esperimenti computazionali. L'idea nasce dal fatto che i sistemi semplici sono statici, mentre i sistemi troppo attivi sono caotici; quindi, la complessità risiede sull'orlo tra queste due tipologie, dove i sistemi hanno la capacità di emergenza computazionale.

Già nei primi anni '50, Friedrich Von Hayek, premio Nobel per l'economia, aveva introdotto l'idea di "ordine spontaneo" (spontaneous order). Questa emergenza spontanea di ordine generato dal caos e altre idee relative ai sistemi adattativi complessi, hanno portato a una vivida evoluzione di tecniche e modelli matematici e computazionali, che adesso esploreremo brevemente.

Un primo interessante modello, assai utile in molte applicazioni, è quello delle reti neurali artificiali (artificial neural network, ANN) che, usando un approccio connessionista, replica (metaforicamente) il "comportamento" delle reti neurali biologiche.

Le reti neurali artificiali sviluppano un'analogia con il cervello, dal momento che operano una sorta di processo parallelo distribuito.

Un singolo neurone artificiale è composto da:

- sinapsi, ovvero collegamenti che connettono i valori in input ai segnali x_j ($j = 1, \dots, m$), con pesi w_{kj} ;
- un *adder* $v = w_0 + \sum_{j=1}^m w_j x_j$, ovvero una variabile ottenuta dalla somma di valori di input pesati più una distorsione, che è il peso del valore di input x_0 ; la distorsione è fissata uguale a 1. Perciò, $v = \sum_{j=0}^m w_{kj} x_j$;
- la funzione (la cosiddetta *squashing function*) di attivazione (monotona), tramite la quale deriviamo l'output del k-esimo neurone artificiale, cioè:

$$y_k = \varphi \left(\sum_{j=0}^m w_{kj} x_j \right).$$

Tra le reti neurali artificiali più usate ci sono le *multilayer feedforward networks*, che hanno la struttura mostrata in Figura 16.4.

Un altro argomento di vivido interesse nei sistemi adattativi complessi è l'*Artificial Life* (ALife), che tradizionalmente usa i seguenti strumenti:

- Ant colony optimization;
- Evolutionary algorithm;
- Genetic algorithm;
- Genetic programming;
- Swarm intelligence.

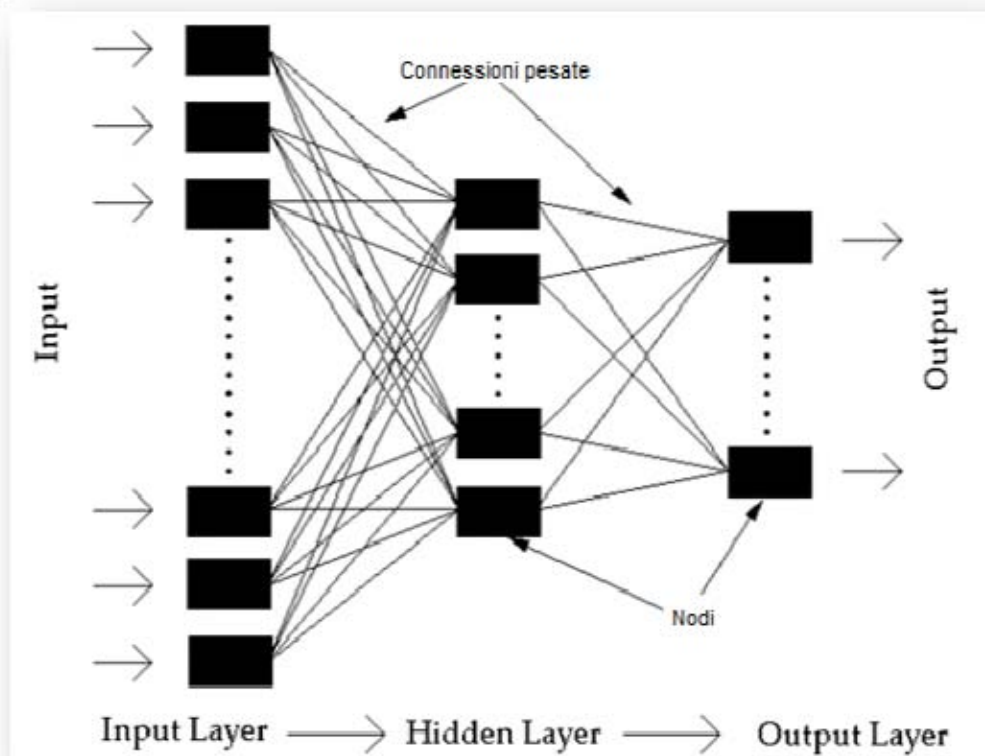


Figura 16.4. Un esempio di feed-forward artificial neural network.

È certamente utile concludere la discussione sui sistemi adattativi complessi con una riflessione di Murray Gell-Mann, premio Nobel per la fisica e uno dei più importanti studiosi e divulgatori nel campo dei sistemi complessi.

"[CAS] receives a stream of data about itself and its surroundings. In that stream it identifies particular irregularities and compresses them into a concise schema, one of many possible ones related by mutation and substitution. In the presence of further data from the stream, the schema can supply descriptions of

certain aspects of the real world, predictions of events that are about to happen in the real world, and prescriptions for behavior of the complex adaptive system in the real world. In all these cases, there are real world consequences: the description can turn out to be more accurate or less accurate, the predictions can turn out to be more or less reliable, and the prescriptions for behavior can turn out to lead to favorable or unfavorable outcomes. All of these consequences then feed back to exert 'selection pressures' on the competition among the various schemata, so that there is a strong tendency for more successful schemata to survive and for less successful ones to disappear or at least be demoted in some sense...A complex adaptive system (CAS) may be an integral part of another CAS, or it may be a loose aggregation of complex adaptive systems forming a composite CAS. Thus, a CAS has a tendency to give rise to others."

La misura di Kullback-Leibler

Supponiamo di avere due distribuzioni, $F = f(x)$ e $G = g(x)$. Definiamo la misura di Kullback-Leibler, la cosiddetta entropia relativa (relative entropy)¹:

$$S^*(F; G) = E_F \left[\log \left(\frac{f(x)}{g(x)} \right) \right] = \int_{-\infty}^{\infty} \log \left(\frac{f(x)}{g(x)} \right) f(x) dx$$

¹ Usando $-S^*(F; G)$, minimizzando la misura di Kullback-Leibler, sarebbe come massimizzare l'entropia relativa.

Come è semplice verificare:

$$S^*(F; G) \geq 0 \text{ e } S^*(F; G) = 0 \iff f(x) = g(x) \text{ q.o.}^2.$$

La misura di Kullback-Leibler potrebbe essere considerata come misura della distanza tra queste distribuzioni.

Pensando in termini di entropia, otteniamo:

$$\begin{aligned} 0 \leq S^*(F; G) &= \sum_x f(x) \log\left(\frac{f(x)}{g(x)}\right) \\ &= \sum_x f(x) \log\left(\frac{1}{g(x)}\right) - \sum_x f(x) \log\left(\frac{1}{f(x)}\right) = S(F; G) - S(F) \end{aligned}$$

dove $S(F; G)$ rappresenta l'entropia incrociata (cross entropy) tra F e G .

Dunque, $S^*(F; G)$ ci da l'idea di un uso inefficiente (in termini di codifica dei dati) della distribuzione G piuttosto che della F .

Predizione and retrodizione

In alcuni lavori del 2009 (ancora in corso di stampa), Christopher J. Ellison, John R. Mahoney e James P. Crutchfield e James P. Crutch, Christopher J. Ellison e John R. Mahoney, considerano sia modelli predittivi sia "modelli retrodittivi" (retrodictive models),

² Quasi ovunque.

scoprendo nuove proprietà dei sistemi dinamici stocastici stazionari.

In particolare ottengono una relazione tra la complessità statistica (informazione negli stati interni) e l'informazione osservata, chiamata *excess entropy*.

Brevemente, per averne un'idea intuitiva, si consideri che l'*excess entropy* è la mutual information (definita al paragrafo 2.4) tra due distribuzioni, dove le due distribuzioni sono, qui, la configurazione predittiva e quella retrodittiva:

$$ES = I(\vec{T}; \overleftarrow{T}).$$

Dal momento che possiamo scrivere la mutual information $I(F, G)$ come una differenza tra l'entropia marginale e l'entropia condizionale (ibidem), avremo:

$$ES = I(\vec{T}; \overleftarrow{T}) = S(\overleftarrow{T}) - S(\overleftarrow{T} | \vec{T}),$$

quindi, ES rappresenta la riduzione media di \overleftarrow{T} , data la conoscenza di \vec{T} . In altre parole, l'*excess entropy* può essere vista come l'informazione che attraversa il canale tra il passato e il futuro.

Più specificamente, Christopher J. Ellison, John R. Mahoney e James P. Crutchfield dimostrano perché l'informazione archiviata nel presente non è *excess entropy*, ma complessità statistica.

Dunque, usando il futuro per predire il passato potremmo retrodiredo, scandendo le variabili di misurazione nella direzione inversa del

tempo (indietro nel tempo), invece che nella classica direzione temporale in avanti (ovvero, usare il passato per predire il futuro).

Modellizzazione di Sistemi Complessi

Supponiamo di voler costruire una mappa topografica di un qualunque territorio. Dobbiamo scegliere quanti elementi vogliamo considerare e quali dettagli vogliamo considerare. Se aggiungiamo sempre più dettagli gradualmente complichiamo la comprensione e l'utilità della mappa stessa. Alla fine questa mappa diventerebbe talmente complessa che sarebbe meglio non considerarla e usare il territorio stesso come mappa! Lo scopo di una mappa è trasmettere idee nello spazio e nel tempo. Ciò non significa solamente semplificare ("Make everything as simple as possible, but not simpler" come diceva Albert Einstein), è anche necessario ottenere una comune comprensione della mappa. Dunque, una mappa può essere considerata un modello. La mappa ci consente di capire e usare le caratteristiche chiave del territorio che rappresenta;

inoltre, così come fa un modello, ci consente di notare caratteristiche altrimenti non osservabili.

In questo volume esploreremo un'ampia varietà di tecniche di modellazione e sarà cruciale e utile tenere ben in mente quanto dicevamo riguardo la mappa e il territorio, onde evitare di cadere nella tentazione di complessificare troppo i nostri "mondi".

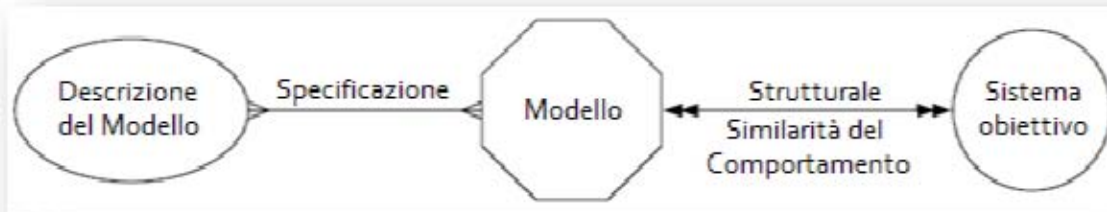


Figura 17.1. Secondo Michael Weisberg, "A concrete model and its relations to a model description and a real-world target. The connection to the target system is optional, as some models are studied for their intrinsic interest or without knowing whether or not they are similar to a real world target". [Riprodotta dietro autorizzazione dell'Autore].

17.1 Modelli di sistemi dinamici e mappe iterative

Per modellare una struttura complessa è necessario capire le caratteristiche interessanti dei sistemi semplici e di quelli complessi, da un punto di vista dinamico. Adesso discuteremo semplici modelli, che non sono generalmente presi in

considerazione nell'analisi scientifica dei "sistemi semplici" e che invece sono assai utili nello studio dei sistemi complessi.

Anche se i modelli matematici sono spesso sviluppati nel tempo continuo (con equazioni differenziali), noi useremo generalmente il tempo discreto (ed equazioni alle differenze o – più generalmente – relazioni di ricorrenza³), certamente più adatte per le simulazioni agent-based. Inoltre, usare uno spazio-tempo continuo a livello micro poco si adatta alla prospettiva dei sistemi complessi.

Consideriamo una funzione f , tale da poter definire una mappa iterativa come segue:

$$s(t) = f(s(t - \delta t)).$$

Dove $s(t)$ rappresenta lo "stato" del sistema al tempo t . Chiameremo la funzione f la *mapping function* di questa mappa iterativa nel tempo discreto.

Consideriamo, ad esempio, $s \in \mathbb{R}$; otterremo

$$s(t) = s_0,$$

e chiameremo questa mappa iterativa mappa costante.

Graficamente, come da figura 3.2, mostriamo sulla parte sinistra la mappa iterativa come funzione del tempo e sulla parte destra la

³ Le relazioni di ricorrenza (recurrence relations) potrebbero avere un comportamento caotico. Esiste un'ampia letteratura in fisica e in matematica nel campo dell'analisi non lineare.

relazione tra $s(t)$ e $s(t-1)$, ottenendo i valori consecutivi per s . Definiamo il valore iniziale $s(0)$ sull'ascissa, trovando in tal modo il corrispondente valore $s(1)$, ovvero il punto $(s(0), s(1))$, continuando in sequenza con la stessa procedura.

Una mappa iterativa ha punti fissi quando $s_0 = f(s_0)$. Questi punti fissi possono essere stabili o instabili, come può essere visto nell'esempio seguente.

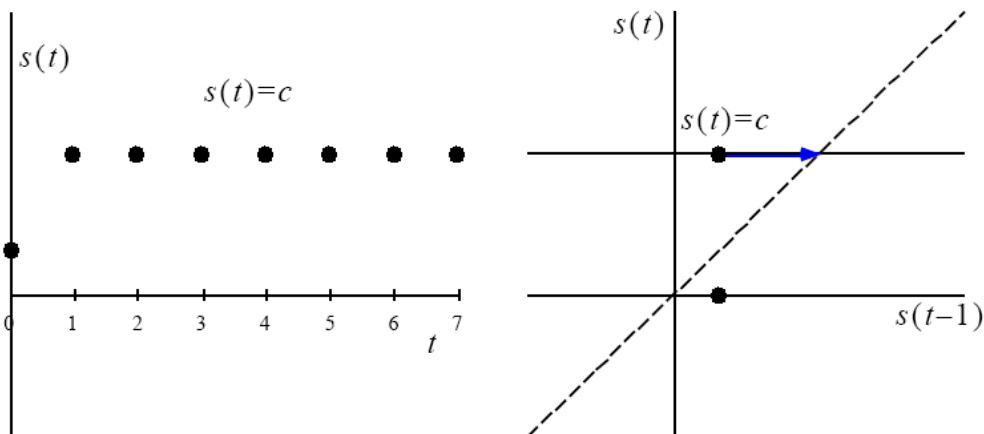


Figura 17.2. Mappa iterativa costante $s(t) = s_0$.

Un'altra classe di mappe iterative può essere ottenuta considerando:

$$s(t) = s(t-1) + v,$$

la cosiddetta mappa iterativa lineare, ottenuta aggiungendo v al precedente valore di s . In fisica se v è considerata la velocità, questo modello rappresenta la propagazione nello spazio.

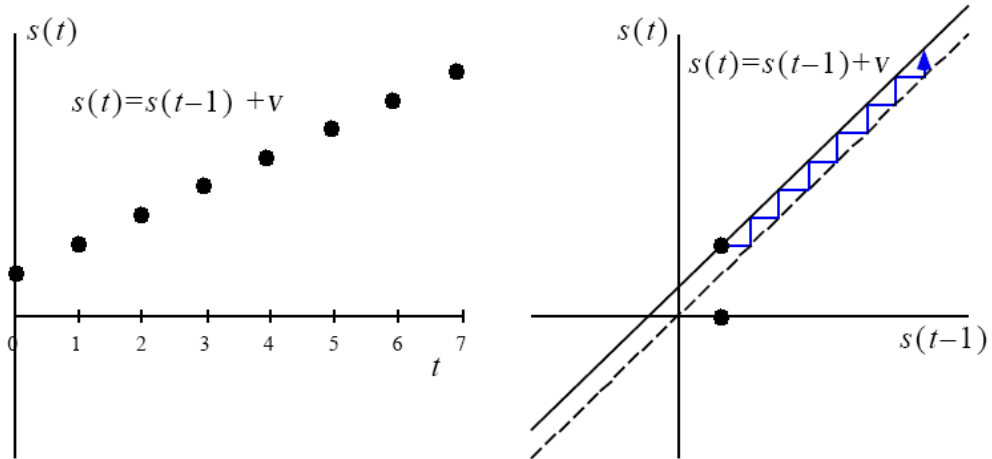


Figura 17.3. Mappa iterativa lineare.

Generalizzando la precedente equazione tramite un parametro g con una velocità $v = 0$, nel modo seguente:

$$s(t) = gs(t - 1),$$

e riscrivendo come segue,

$$s(t) = g^t s_0 = e^{\ln(g)t} s_0,$$

otterremo l'espressione come una crescita esponenziale (o decadenza).

Graficamente possiamo vedere che il punto fisso è stabile per $-1 < g < 1$, come nelle seguenti due figure. Inoltre, c'è un comportamento chiaramente differente se $1 > g > 0$ o $-1 < g < 0$.

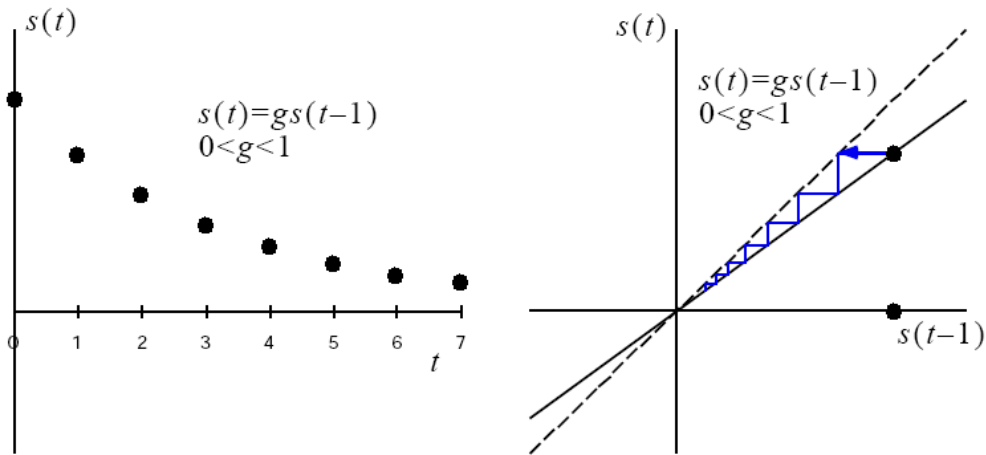


Figura 17.4. Mappa lineare iterativa generalizzata con punto fisso stabile e g positivo.

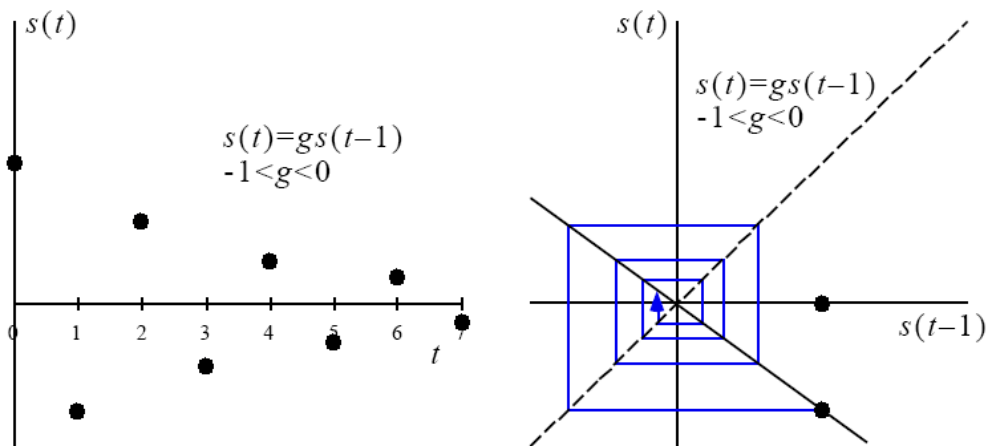


Figura 17.5. Mappa lineare iterativa generalizzata con punto fisso stabile e g negativo.

Nelle prossime due figure possiamo vedere che il punto fisso è instabile per $g > 1$ e per $g < -1$. Inoltre, c'è un comportamento chiaramente differente se $g > 1 > 0$ o $g < -1 < 0$.

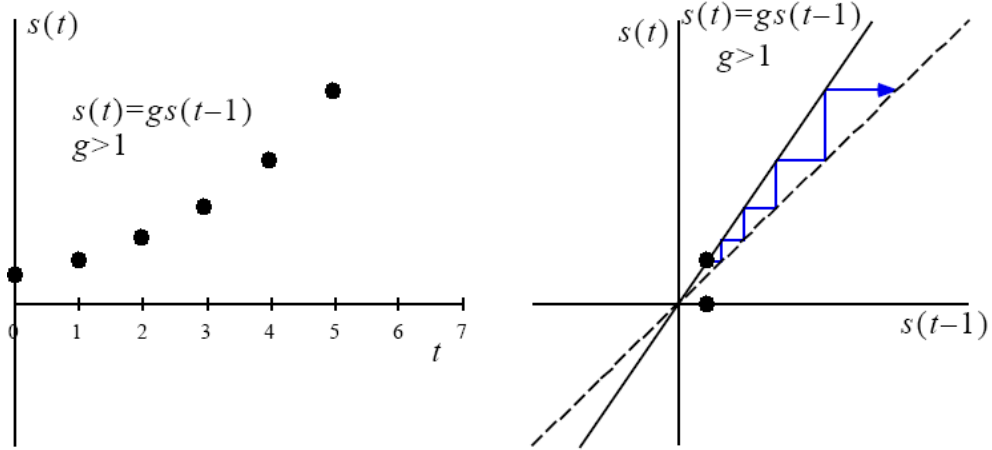


Figura 17.6. Mappa lineare iterativa generalizzata con punto fisso instabile e g positivo.

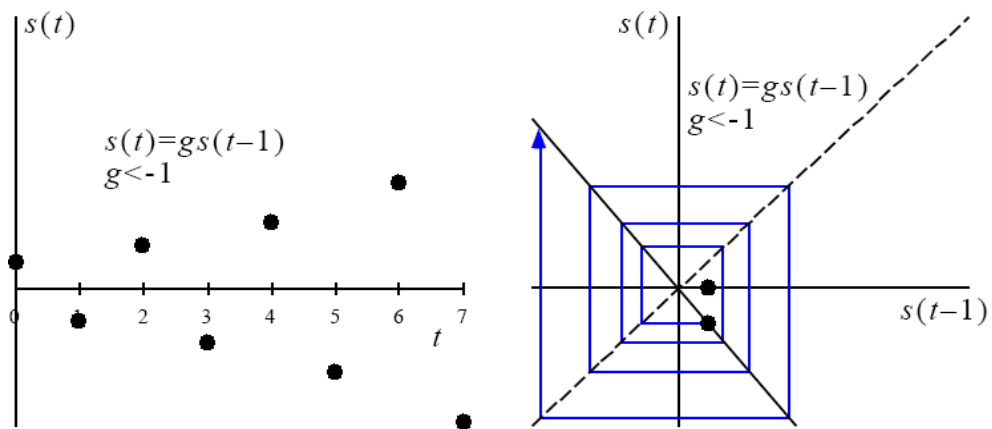


Figura 17.7. Mappa lineare iterativa generalizzata con punto fisso instabile e g negativo.

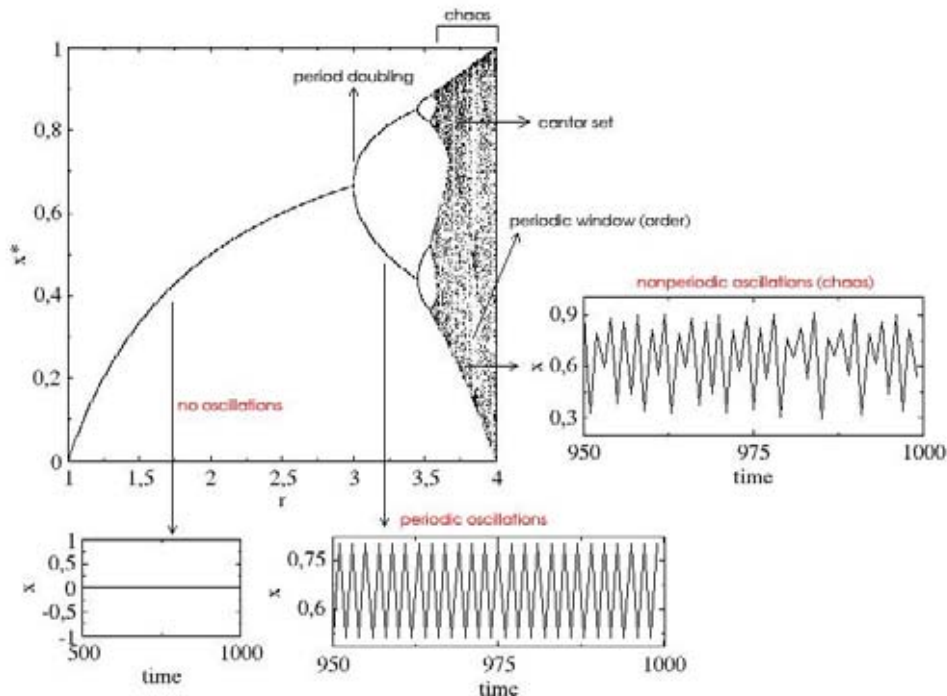


Figura 17.8. Un diagramma di biforcazione logistica dopo molte iterazioni. Oltre cicli e punti stabili, ci saranno comportamenti caotici.

17.2 Automi cellulari (cellular automata, CA)

Ho sempre pensato che il primo libro da leggere per iniziare lo studio dei sistemi complessi è *Flatland* di Abbott. Non è dunque stato sorprendente per me trovare in un recente e famoso libro sui sistemi adattativi complessi (v. paragrafo 2.4) di John H. Miller and Scott E. Page, parlando di CAS a una dimensione, la seguente affermazione: *"In Abbott's Flatland, geometric figures confined to living in a two-dimensional world gain insight into the third-dimension when a sphere slowly passes through their plane. The*

sphere begin as a point, grows into ever larger circles, eventually reserves its course and returns to a point, and disappears. After seeing this amazing sequence of activity, the figures confined to Flatland begin to glimpse the third dimension. Here we explore some simple models with a similar motivation to Abbott's sphere, namely, to provide some useful glimpse into the behavior of complex adaptive social systems".

Gli automi cellulari rappresentano l'idea espansa di mappa iterativa (v. paragrafo 3.1) per modellare sistemi distribuiti spaziali, su uno spazio reticolare (*lattice*), dinamicamente. In un automa cellulare operiamo in uno spazio-tempo discreto e omogeneo (in termini di influenze di variabili). Avremo un ridotto numero di stati di variabili semplici. Costruiamo un modello distribuito nello spazio, dando maggior importanza alle regioni vicine (questo ha senso in molti sistemi, soprattutto pensando alla comunicazione). Immaginiamo di divider questo spazio in celle. Poi, usando qualche variabile, definiamo degli stati al tempo t in una specifica cella. Quindi,

$$s(l, m, n; t) = s(x_l, y_m, z_n; t); \quad l, m, n \in \mathbb{Z}.$$

Questa espressione è definita in uno spazio 3D, ma potrebbe anche essere definite in 1D, 2D o a più dimensioni.

Adesso, diamo una cosiddetta regola iterativa:

$$s(l, m, n; t) = R_{l', m', n' \in \mathbb{Z}}(s(l' - l, m' - m, n' - n; t - 1))$$

dove R è una funzione di l, m, n al tempo $t - 1$ e nella posizione della cella nello stato $s(l, m, n; t - 1)$. La regola non cambia nel tempo e nello spazio.

Gli automi cellulari possono avere, semplificando, celle binarie (variabili). Ad esempio on/off o 0/1, come in Figura 20.9.

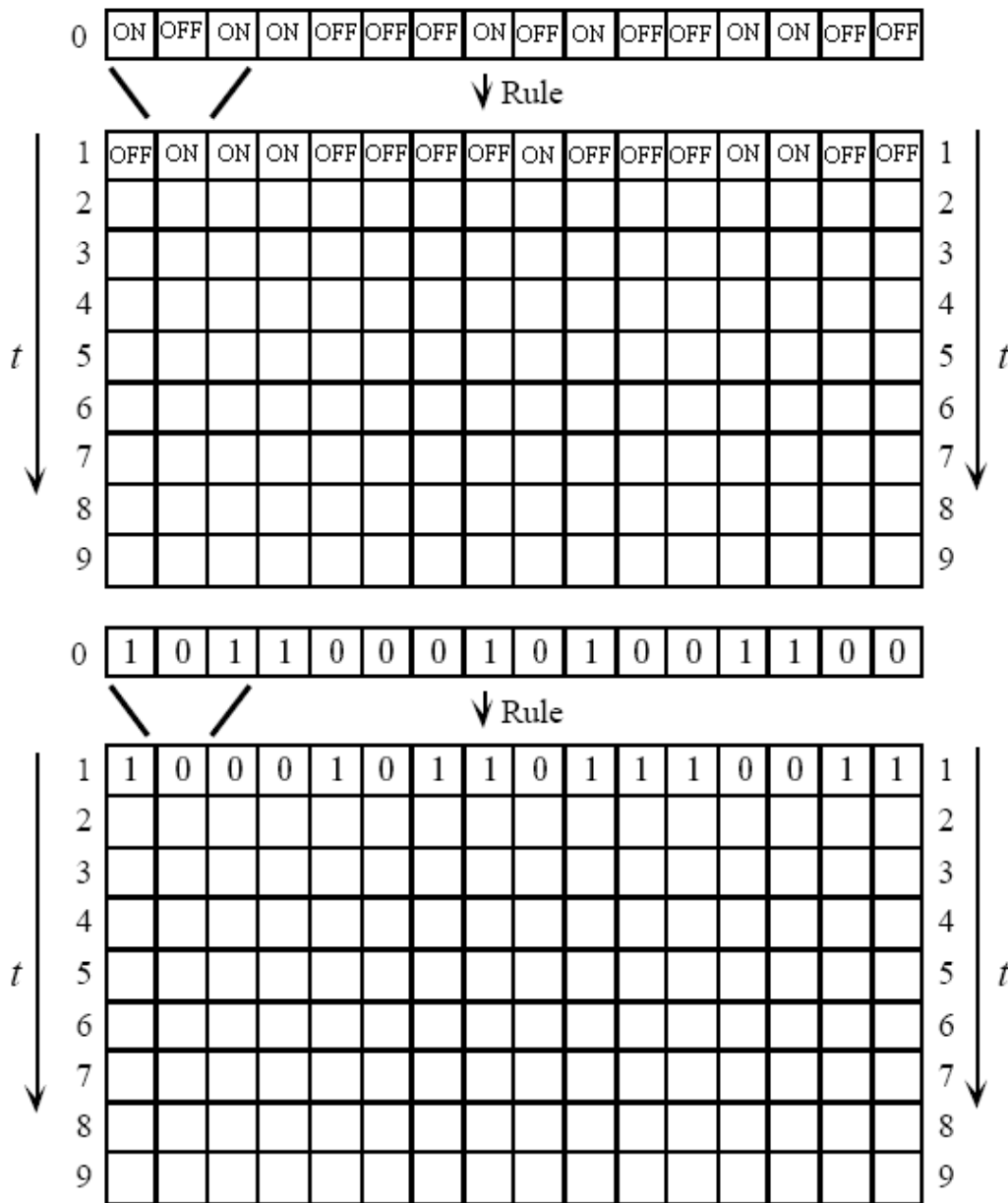


Figure 17.9. Due differenti automi cellulari uno-dimensionali.

Wolfram ha classificato gli automi cellulari in quattro classi:

- a) Classe 1: le regole evolvono velocemente verso lo stato omogeneo (e fisso).
- b) Classe 2: le regole evolvono verso uno stato disomogeneo (e fisso) semplicemente stabile o strutture periodiche (una specie di regola "fai l'opposto").
- c) Classe 3: le regole evolvono verso configurazioni caotiche.
- d) Classe 4: le regole evolvono verso strutture complesse.

Un esempio di questa classificazione può essere trovato nella seguente figura.

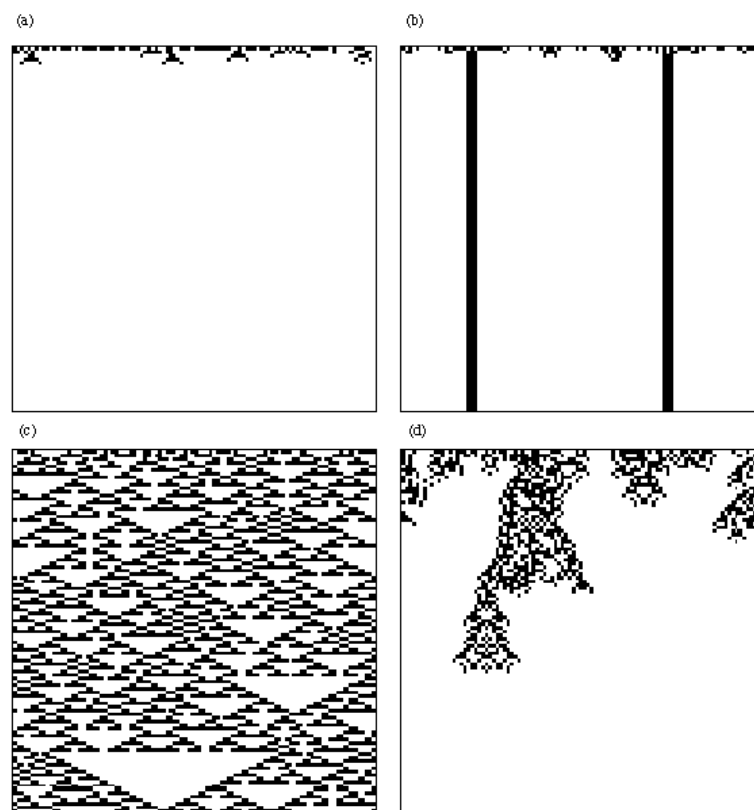


Figura 17.10. Quattro automi cellulari in uno spazio con 100 celle (periodico). Il tempo evolve dall'alto (condizione iniziale), a scendere.

La dinamica di un automa cellulare è definita, come per le mappe iterative, con attrattori e cicli. Usando uno spazio finito dobbiamo definire delle condizioni al contorno, come nella precedente figura, dove abbiamo usato condizioni al contorno periodiche. In questo caso lo spazio è un cerchio per una dimensione, un toro (forma geometrica "a ciambella") per due dimensioni e uno spazio toroidale per più dimensioni.

Modelli basati su agenti

Per creare modelli usando oggetti agent-based⁴ si procede secondo una logica bottom-up (dal particolare al generale) nella comprensione dei sistemi complessi.

In pratica si crea un programma per computer contenente parti di programma che rappresentano agenti artificiali, modellando questi agenti in un ambiente, dotandoli di alcune regole e lasciandoli interagire gli uni con gli altri nel tempo nella cosiddetta simulazione agent-based, costruendo in tal modo un sorta di laboratorio artificiale per sperimentazioni sociali, di fenomeni economici o qualsiasi altra cosa siamo interessati a osservare.

Questo nuovo approccio, a differenza della maggior parte dei modelli matematici e statistici, ci permette – ad esempio – di costruire agenti eterogenei, sperimentare in situazioni "lontane

⁴ Secondo John H. Miller and Scott E. Page, questo potrebbe essere un miglior nome per identificare i modelli agent-based, è inoltre utile distinguere tra modelli bottom-up (ovvero, artificial adaptive agents) and simulazioni bottom-up (ovvero, artificial life).

dall'equilibrio" e considerare nel modello le conseguenze dell'interazione tra agenti.

I modelli agent-based sono diventati molto usati in economia e nelle scienze sociali; Kenneth J. Arrow (premio Nobel per l'economia), a proposito del volume di Miller, J. H. e Page, S.E. "*Complex Adaptive Social Systems: An Introduction to Computational Models of Social Life*", Princeton University Press, 2007, afferma: "*the use of computational, especially agent-based, models has already shown its value in illuminating the study of economic and other social processes*".

Operando con paradigmi *agent-based* è possibile costruire ambienti di simulazioni che – grazie a un continuo adattamento degli agenti artificiali, delle regole e dell'ambiente stesso – riescano a:

- considerare, come parte del modello costruito, caratteristiche altrimenti non includibili;
- manipolare e gestire in modo continuativo le caratteristiche strutturali del sistema;
- fornire scenari di sviluppo secondo diversi gradi di probabilità;
- includere la caratteristica di autoapprendimento e aggiornamento automatico, così da poter perfezionare previsioni, valutazioni e scenari.

In tal modo una simulazione *agent-based* è in grado di sostituire vantaggiosamente i metodi classici con scenari di sviluppo

adattabili, attendibili e in grado di adeguarsi ai diversi contesti, perfezionando così la qualità delle previsioni stesse.

Grazie ai modelli di simulazione *agent-based* è inoltre possibile effettuare un gran numero di sperimentazioni artificiali che, oltre essere a basso costo, danno la possibilità di condurre test altrimenti non fattibili (ad esempio test su esseri viventi non fattibili per motivazioni etiche o per costi troppo elevati).

Come sottolineato da Gilbert, una ragione della diffusione delle metodologie agent-based nelle scienze sociali è da attribuire al fatto che queste simulazioni consentono di costruire modelli dove le entità individuali e le loro interazioni sono direttamente rappresentate.

Naturalmente l'aggiunta di simulazioni, mentre riduce i costi di indagine e introduce maggiore precisione e attendibilità delle previsioni, comporta una ulteriore complessificazione del sistema e richiede, per poter dare buoni risultati, l'introduzione di altre metodologie e tecnologie (modelli matematici, metodologie statistiche, software, algoritmi, competenze e soprattutto considerazioni di natura psicologica).

Questo approccio bottom-up è utile per la comprensione dell'emergenza di un sistema complesso: creando gli elementi interagenti a livello micro ed eseguendo simulazioni nel tempo, è possibile manipolare le variabili del modello, per cercare di capire quale combinazione di ambiente, regole e agenti genera un'emergenza sistemica.

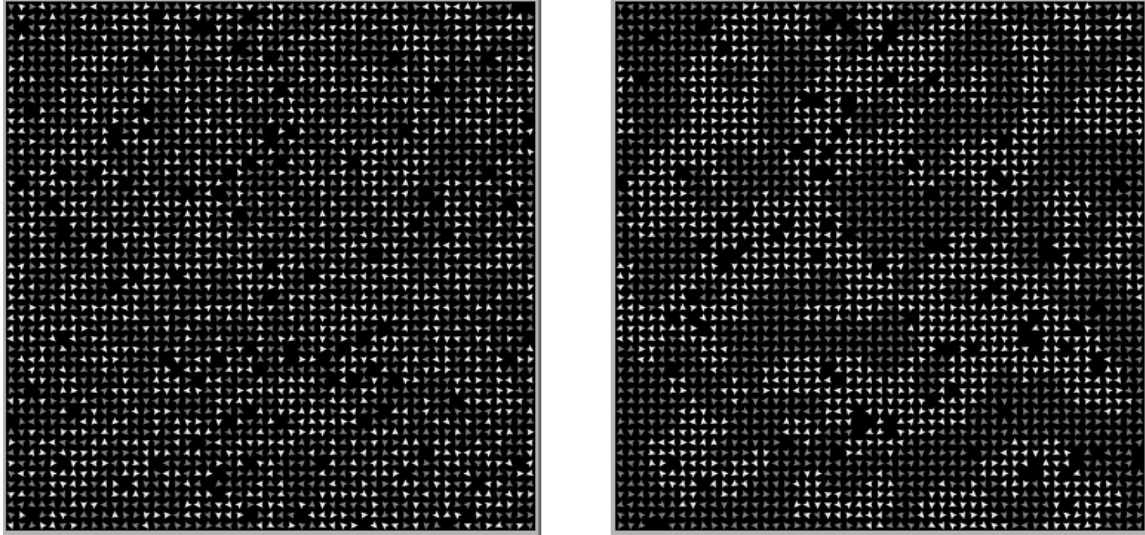


Figura 18.1. Il modello di segregazione di Thomas Shelling (premio Nobel per l'economia) all'inizio della simulazione (a sinistra) e dopo che l'equilibrio è stato raggiunto (a destra). Simulazione eseguita con NetLogo (v. paragrafo 3.5) con 2.500 agenti e variabile "%-similar-wanted" (tolleranza uniforme) del 30%.

In una simulazione agent-based rappresentiamo una sorta di "realtà sociale" e, come visto nella precedente discussione sulla mappa e il territorio, non ci interessa inserire tutte le caratteristiche di un "sistema reale" nel nostro modello, ma siamo piuttosto interessati a poche caratteristiche, elementi e interazioni: tanto quanto basta per consentirci di osservare i fenomeni emergenti del sistema che stiamo cercando di capire. In pratica costruiamo il nostro modello partendo dal particolare e andando al generale (bottom-up approach).

Secondo questa costruzione possiamo pensare alla simulazione agent-based come a esperimenti sui sistemi complessi, ma sono necessarie alcune considerazioni. Ad esempio, molti sistemi complessi sono studiati tramite metodologie agent-based (o similari) solo perché non è possibile fare altrimenti e non perché questo sia il miglior modo di procedere; si pensi ai disastri naturali, anche se possibile pianificare simulazioni "dal vivo" con esseri umani che attuano piani di evacuazione, non è certamente possibile inserire in tali piani elementi che uccidano effettivamente le persone (per ovvie ragioni etiche e morali), inoltre il costo di tali "prove tecniche" può essere – in certi casi – parecchio elevato (si pensi a una evacuazione di prova di una intera città!). In questo caso una simulazione agent-based, è adatta a rappresentare tali fenomeni (anche se non fosse il migliore strumento, sarebbe comunque il più "attuabile") dal momento che offre strumenti pratici per la gestione di scenari probabili, sulla base dei quali decidere le migliori strategie.

Come visto, dunque, le simulazioni agent-based danno la possibilità di cambiare condizioni iniziali, input, comportamenti, interazioni, strutture, ambiente e altri "parametri", per avere una vasta serie di scenari e non un'unica soluzione (o previsione). Questo è contemporaneamente un grande limite e tra le più grandi risorse dei modelli agent-based.

Secondo quanto affermano John H Miller e Scott E. Page, *"no single theoretical tool is suitable for all needs, and we are certainly not claiming that agent-based object modeling is an*

exceptions. We do, however, suggest that the constellation of features offered by such models represents a very appropriate set from which to gain new insights into complex adaptive social systems".

Strumenti Tradizionali	Strumenti Agent-based
Precisi	Flessibili
Piccoli processi	Orientati ai processi
Senza tempi	Con tempi
Ottimizzanti	Adattativi
Statici	Dinamici
1, 2, ... o ∞	1, 2, ..., N agenti
Vacui	Spaziali/Reticolari
Omogenei	Eterogenei

Tabella 18.1. Potenziale dei modelli. Adattato da John H Miller e Scott E. Page.

Come è possibile vedere in letteratura e dalle precedenti argomentazioni, i modelli agent-based presentano molte criticità, come sottolineato – tra gli altri – da Gilbert, che inoltre fa anche notare: *"you should assume that, no matter how carefully you have designed and built your simulation, it will contain bugs (code that does something different to what you wanted and expected)".* Dal momento che i modelli agent-based spesso si traducono in codice

per eseguire simulazioni, dobbiamo necessariamente considerare problemi di "coding" oltre ad eventuali errori nel nostro modello.

È abbastanza importante tenere a mente anche Axelrod quando sottolinea: *"Achieving internal validity is harder than it might seem. The problem is knowing whether an unexpected result is a reflection of a mistake in the programming, or a surprising consequence of the model itself. [...] As is often the case, confirming that the model was correctly programmed was substantially more work than programming the model in the first place"*.

Gli errori sono inevitabili, ma grazie alla flessibilità dei modelli agent-based le applicazioni sono in fortissima crescita nelle scienze sociali e in molti altri campi. Infatti il campo dell'agent-based modeling è composto da vari tipi di scienziati, quali scienziati sociali, ingegneri, filosofi, economisti, psicologi, matematici biologi, fisici, informatici, sociologi, medici e così via. Ciò non è sorprendente parlando di sistemi complessi. Sicuramente la scienza è disciplinare ma dobbiamo sempre ricordare che la realtà è comunque unica.

Un modello agent-based si compone di molti agenti interagenti, dotati di regole in un ambiente. Gli agenti possono rappresentare persone, imprese, nazioni e molti altri aggregati. Questi agenti scambiano informazioni, regole, comportamenti e così via. In particolare i modelli agent-based differiscono da altri modelli computazionali soprattutto per questa possibilità di modellare l'interazione tra agenti.

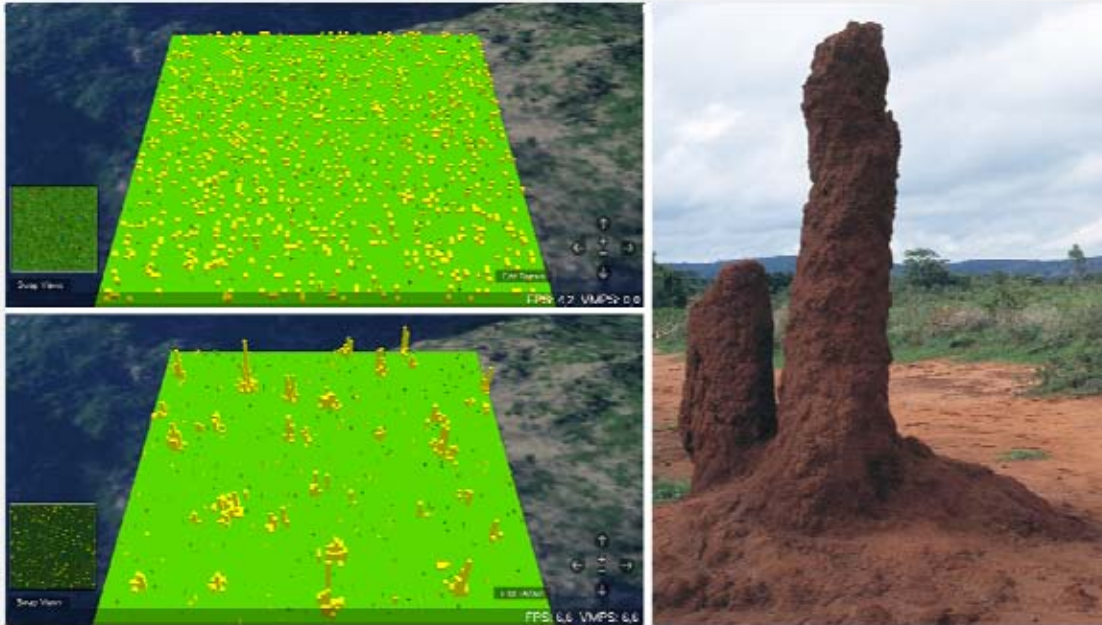


Figura 18.2. Un noto esempio di organizzazione emergente. Sulla parte sinistra una simulazione in StarLogoTNG, che rappresenta il comportamento di termiti che assemblano bastoncini di legno incolonnandoli in pile, seguendo un limitatissimo insieme di semplicissime regole dalle quali emerge il termitaio. Sulla destra un vero termitaio, assemblato con le stesse regole da termiti reali.

18.1 Strumenti per la modellizzazione

Grazie alle attuali capacità computazionali e all'ampio interesse che i sistemi adattativi complessi hanno suscitato in vari campi scientifici, negli ultimi anni sono stati sviluppati molti

strumenti che consentono a svariate tipologie di individui di creare modelli per un ampio numero di applicazioni. In questo volume, come sempre, a seconda dell'uso e delle caratteristiche che sarà necessario sviluppare, si farà un ampio uso di toolkits, framing, ambienti di sviluppo, linguaggi di programmazione, e librerie. Ovviamente sarebbe impossibile passare in rassegna tutti gli strumenti usati per la creazione di modelli di sistemi complessi e anche per quelli qui citati daremo solo alcune informazioni, rimandando al sito web ufficiale per ulteriori esplorazioni.

Creare modelli di sistemi complessi, soprattutto per i modelli agent-based, richiede un computer; dunque un primo passo necessario è la scelta del sistema operativo in cui andremo ad operare. Sia i client, sia i server sono adatti per la modellazione agent-based; ovviamente dove sia necessaria un'alta capacità computazionale si rende indispensabile la pianificazione dell'uso di più computer in cluster e in tal caso un sistema server-based rappresenta una scelta più adatta.

Le configurazioni standard per la creazione di modelli di sistemi complessi richiedono un singolo computer per l'elaborazione.

È oggi disponibile una vasta scelta di sistemi operativi. Vediamone brevemente alcuni.

La maggior parte dei tools "girano" sotto Windows (versioni 3.1, 95, 98, 98se, NT, 2000, XP, Vista, 7 e corrispondenti versioni Server, ad esempio 2003 Server, 2008 Server, ecc.) e Linux (x86 or x86_64 Linux versions, SuSE Open Linux 10.2 o successivi, Ubuntu Linux 7.04 o successivi, ecc.). Molti tools rendono

disponibili versioni per Macintosh (generalmente la versione OS X, basata su piattaforma Unix FreeBSD con XNU microkernel) per entrambe le tipologie di processori (PowerPC e Intel).

Altri sistemi operativi molto usati per creare modelli di sistemi complessi sono Unix, OS/2, OS/370, OS/400, HPUX, SunOS, DOS, AIX, BSD e molti altri.

Una discussione a parte meritano le applicazioni eseguite su Java Virtual Machine (JVM), che consente a un programma – entro alcuni limiti – di essere indipendente dalla piattaforma utilizzata, così come sottolineato dallo slogan: *"write once, run everywhere"*.

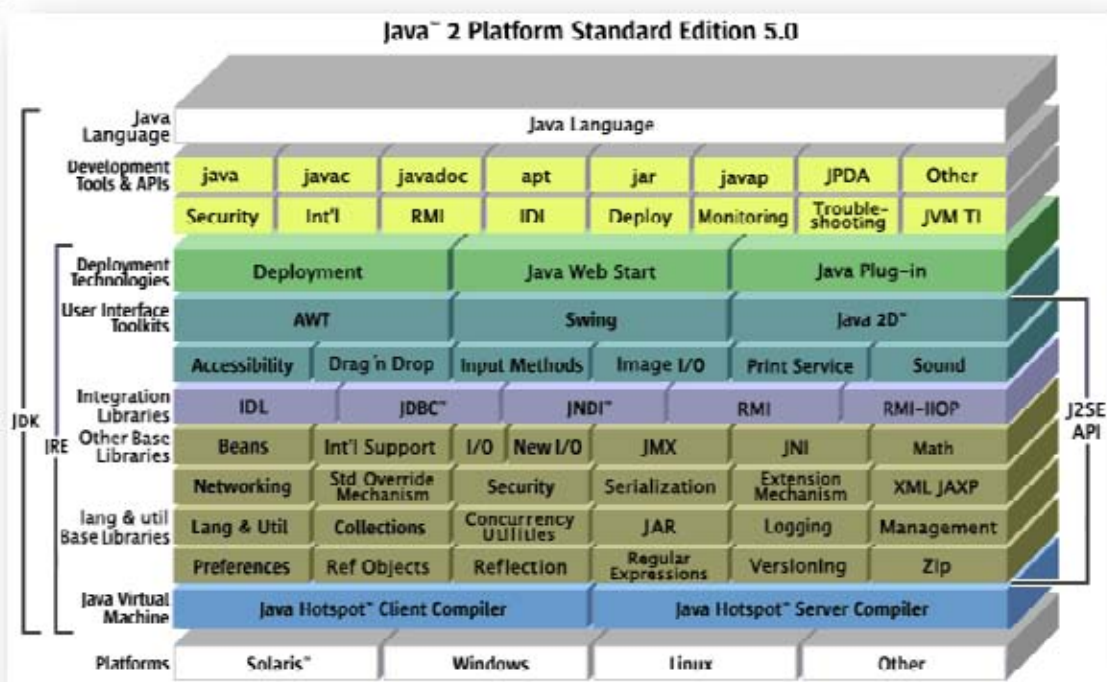


Figura 18.3. Una piattaforma Java™. <http://java.sun.com>.

La maggior parte dei sistemi operativi include (o ha la possibilità di integrare) una versione di Java Virtual Machine (JVM): Java 2 JVM, Bea's JRockit JVM, SDK versione 1.4.1 o successiva, Java 2 SDK, Java Runtime Environment(JRE), IBM's JVM, Java Development Kit (JDK), Java-1.4-capable PDAs e molte altre.

Per la creazione di modelli di sistemi complessi l'uso intensivo di un linguaggio di programmazione potrebbe essere un'ottima scelta. Generalmente programmare dà maggiore flessibilità rispetto ai più semplici toolkit, ma spesso richiede un'ampia esperienza e la capacità di trasformare idee in diagrammi a blocchi, pseudo-codice e infine codice eseguibile.

D'altra parte l'ampia disponibilità di toolkit facili da usare offre l'opportunità anche ai non esperti di informatica di creare modelli concentrandosi su emergenza sistemica e complessità, senza doversi preoccupare di virtuosismi informatici. Il solo problema di questi toolkit è che spesso non hanno la flessibilità che serve per il modello che stiamo creando: restano comunque molto utili per una prima analisi del fenomeno che stiamo studiando che potrà comunque essere approfondito successivamente con strumenti più flessibili.

Queste considerazioni sono passate al vaglio di molti scienziati della complessità e in effetti negli ultimi anni sono stati sviluppati sempre più tool facili da usare con capacità e flessibilità sempre più elevate. Comunque non esistono alternative migliori di altre in questo momento (per le mie ricerche, molte volte, "python" è stato il linguaggio di programmazione più adatto); dipende da

metodologie, preferenze, conoscenze, costi (in termini monetari, ma anche di tempo, forza, capacità, ecc.), requisiti del modello e molti altri fattori. Inoltre stanno nascendo sempre più tools per applicazioni specifiche, adatti ad analizzare particolari fenomeni o situazioni, tramite modelli. Un esempio potrebbe essere il jES (Java Enterprise Simulator, <http://web.econ.unito.it/terna/jes/>) sviluppato da Pietro Terna; descritto nel sito web ufficiale "both to simulate the activities - and the consistent emerging results - of an actual enterprise and to build virtual or hypothetical enterprises." jES è stato scritto in Java ed è programmato con le librerie di Swarm.

Swarm (www.swarm.org) è un pacchetto software per i modelli agent-based (e individual-based) per lo studio dei sistemi complessi. Swarm può essere usato con molti sistemi operativi, tra cui Linux, Mac e Windows (usando Cgiwin per avere l'API Linux emulation layer che fornisce, sostanzialmente, funzionalità Linux API) e può essere programmato usando Java o ObjectiveC. Usando Swarm è possibile creare molte simulazioni e usare il computer come un vero e proprio laboratorio per creare modelli per lo studio dei sistemi complessi. Swarm è abbastanza datato (è stato il mio primo simulatore agent-based, se escludiamo i LEGO), ma ha ancora una vasta comunità. Inoltre Swarm ha ispirato molti modelli (ad esempio, HeatBugs) e tools per creare modelli per lo studio dei sistemi complessi.

Uno dei "figli di Swarm", noto per essere veloce e flessibile, è Mason (Multi-Agent Simulator Of Neighborhoods... or Networks... or something..., come riporta il sito web ufficiale).

Mason è adatto a creare modelli ad uso generale (general purpose) per lo studio dei sistemi complessi; può essere programmato in Java e può implementare machine learning, altri algoritmi AI (Artificial Intelligence) e reti complesse.

Da un punto di vista grafico, Mason, può gestire visualizzazioni 2D e 3D e generare molti tipi di output (immagini, video, grafici, ecc.). Mason è soprattutto ottimo per simulazione di sciami (*swarm*) anche usando milioni di agenti artificiali.

In Figura 18.4 e 18.5, riporto alcune visualizzazioni di Mason.

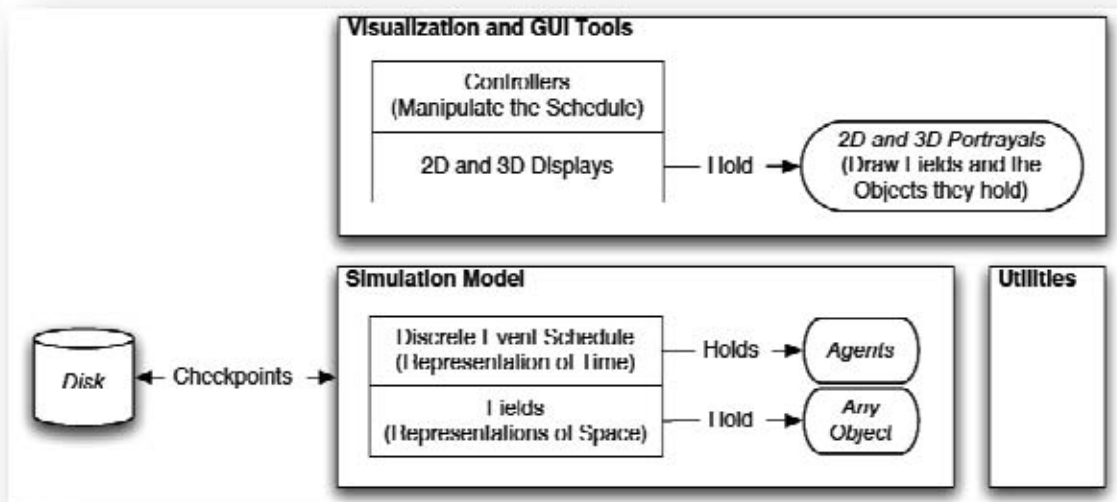


Figura 18.4. Elementi di base dei modelli di Mason e delle strutture di visualizzazione.

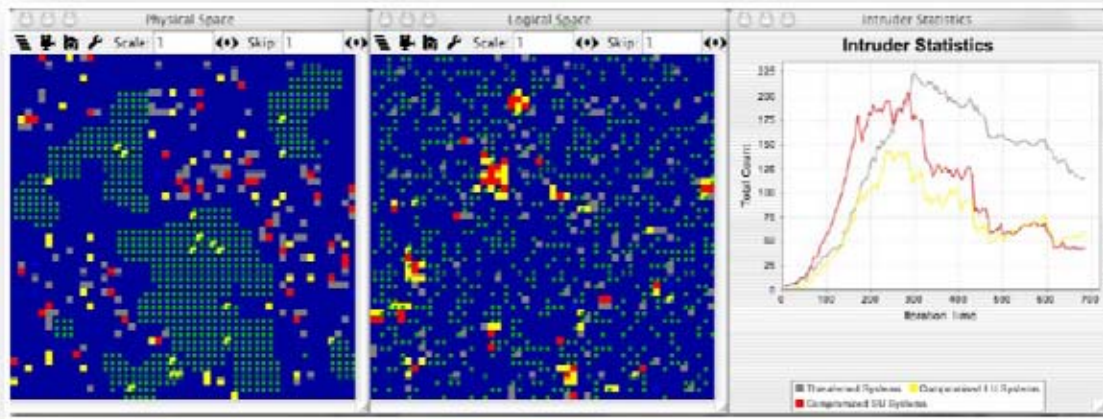


Figura 18.5. Network Intrusion model: lo spazio fisico (a sinistra) and logico (al centro), insieme ad alcune statistiche sull'intrusione e sui sistemi compromessi (sulla destra).

Molti altri tool sono paragonabili a Mason. Nessuno (per il momento) è il migliore in assoluto; come già detto, dipende dal tipo di modello che dobbiamo costruire.

Un altro tool, programmabile con Java, è RePast (Recursive Porous Agent Simulation Toolkit, veramente ottimo, soprattutto per la sua capacità di integrazione con gli altri tool (anche di rete e matematici). RePast può essere programmato in Java, come già detto, è Python (RePastPy), Visual Basic .Net, C++, J#, C#. RePast è particolarmente indicato per le scienze sociali.

Molti altri tool per lo studio dei sistemi complessi sono stati ispirati da Swarm, nel sito di Ascape, ad esempio, si legge *"This work benefits greatly from the work of many software developers in the agent based modeling world. The Swarm community,*

especially Roger Burkhardt, Marcus Daniels and Glen Ropella provided an inspiring and helpful environment in which to explore and discuss modeling issues, and Swarm itself has inspired some Ascape features. While Ascape and Repast were developed separately, related continuing work has benefited greatly from work with Nick Collier, Tom Howe and everyone else at the Repast team".

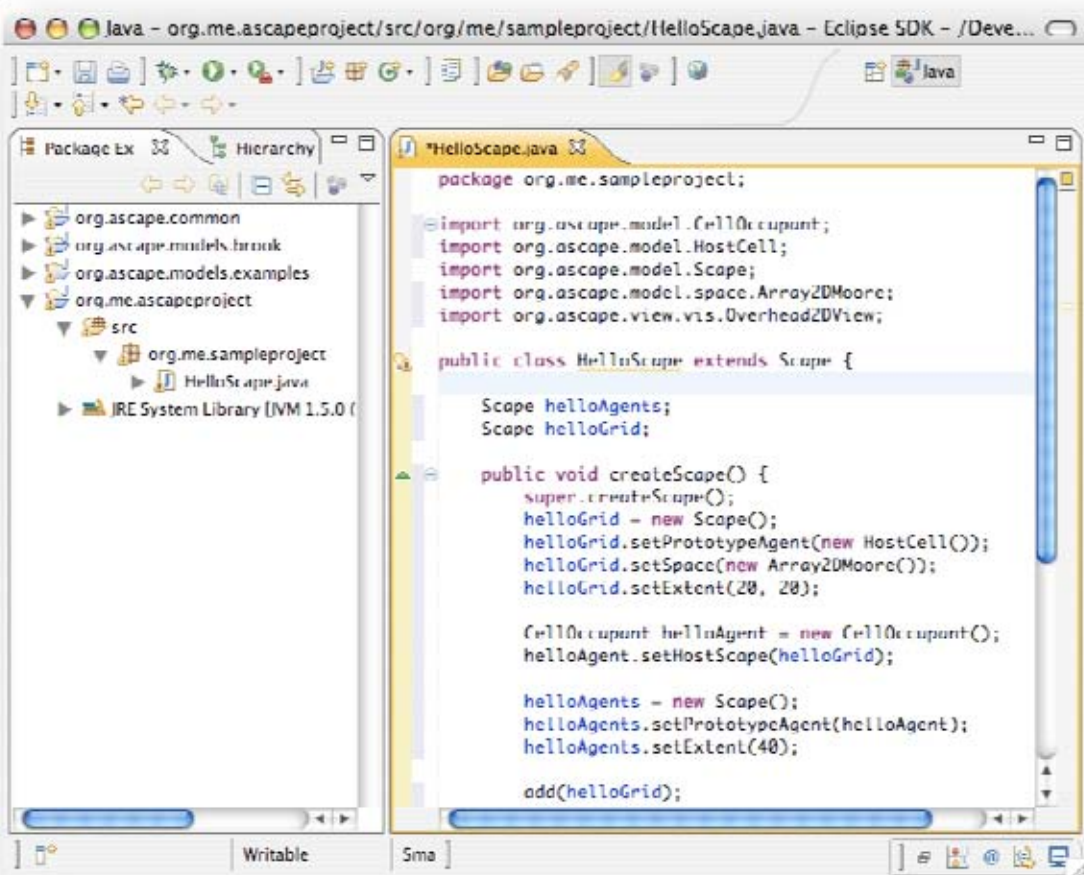


Figura 18.6. Costruzione di modelli Ascape con Eclipse, una piattaforma open source di tipo Integrated Development Environment (IDE), per sviluppare software.

Talvolta è utile, oltre che spettacolare, visualizzare una simulazione in 3D. Come precedentemente visto, molti tool danno questa possibilità. Un ulteriore esempio è dato da Breve, un ambiente 3D per simulazioni multi-agent e artificial life (ALife), dove il comportamento degli agenti può anche essere programmato con Python. Uno scopo differente di simulazione 3D ha invece StarLogo TNG (the next generation), dove l'idea di base è quella di insegnare i sistemi complessi e le basi della programmazione per computer, tramite semplici blocchi, agli studenti K-12 (quelli fino alle classi medie superiori, nel nostro ordinamento), così come fa Open StarLogo (una versione open source della serie StarLogo sviluppata all'MIT) e AgentSheets.

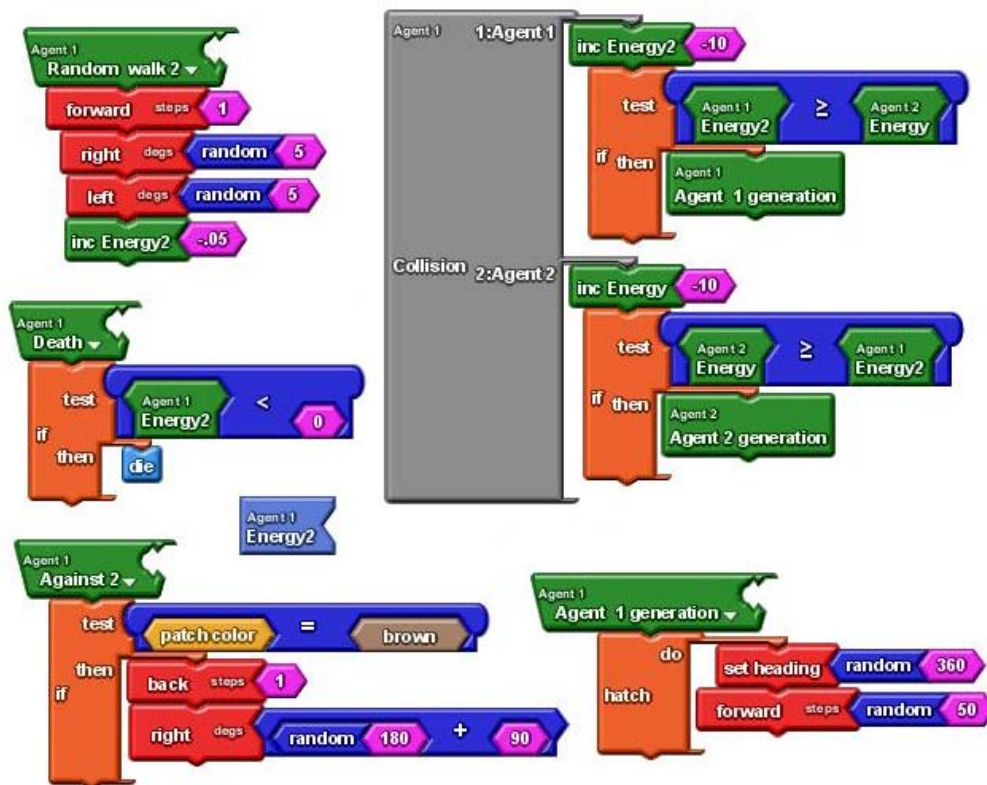


Figura 18.7. Programmare a "blocchi" con StarLogoTNG.

Tra i più popolari tool di simulazione agent-based c'è NetLogo. È abbastanza flessibile e può essere programmato con un linguaggio integrato. Gira su qualsiasi JVM (si installa con la versione 1.4.1 o successive, ma è preferibile la versione 1.5.0_12 o successive). È gratuito se usato a scopi educativi o di ricerca, ma non è open source. Potrebbe non essere il più adatto per i modelli più complessi. Di seguito riportiamo il codice per generare la simulazione del modello di segregazione di Thomas Shelling, già visto prima (in Figura 18.1).

```
globals [
  percent-similar ;; on the average, what percent of a turtle's
  neighbors
                    ;; are the same color as that turtle?
  percent-unhappy ;; what percent of the turtles are unhappy?
]

turtles-own [
  happy?           ;; for each turtle, indicates whether at least %-similar-
  wanted percent of
                    ;; that turtles' neighbors are the same color as the turtle
  similar-nearby  ;; how many neighboring patches have a turtle with
  my color?
  other-nearby   ;; how many have a turtle of another color?
  total-nearby  ;; sum of previous two variables
]

to setup
  clear-all
  if number > count patches
  [ user-message (word "This pond only has room for " count patches
  " turtles.")
    stop ]

  ;; create turtles on random patches.
```

```
ask n-of number patches
  [ sprout 1
    [ set color red ] ]
;; turn half the turtles green
ask n-of (number / 2) turtles
  [ set color green ]
update-variables
do-plots
end

to go
  if all? turtles [happy?] [ stop ]
  move-unhappy-turtles
  update-variables
  tick
  do-plots
end

to move-unhappy-turtles
  ask turtles with [ not happy? ]
  [ find-new-spot ]
end

to find-new-spot
  rt random-float 360
  fd random-float 10
  if any? other turtles-here
    [ find-new-spot ] ;; keep going until we find an unoccupied
patch
  move-to patch-here ;; move to center of patch
end

to update-variables
  update-turtles
  update-globals
end

to update-turtles
  ask turtles [
    ;; in next two lines, we use "neighbors" to test the eight patches
    ;; surrounding the current patch
    set similar-nearby count (turtles-on neighbors)
```

```

    with [color = [color] of myself]
    set other-nearby count (turtles-on neighbors)
    with [color != [color] of myself]
    set total-nearby similar-nearby + other-nearby
    set happy? similar-nearby >= ( %-similar-wanted * total-nearby /
100 )
  ]
end

to update-globals
  let similar-neighbors sum [similar-nearby] of turtles
  let total-neighbors sum [total-nearby] of turtles
  set percent-similar (similar-neighbors / total-neighbors) * 100
  set percent-unhappy (count turtles with [not happy?]) / (count
turtles) * 100
end

to do-plots
  set-current-plot "Percent Similar"
  plot percent-similar
  set-current-plot "Percent Unhappy"
  plot percent-unhappy
end

```

La maggior parte dei tool che abbiamo visto ha qualche funzione (ad esempio moduli o librerie) per l'analisi delle reti complesse. Comunque, talvolta, uno specifico tool potrebbe essere una scelta migliore, anche per beneficiare della comunità di utenti nello specifico campo delle complex networks.

Uno dei primi tool (sicuramente tra i più conosciuti) è UCINET (<http://www.analytictech.com/ucinet/ucinet.htm>), in figura 18.8 è possibile vederne un esempio. Tuttavia, come abbiamo visto al capitolo 13, Gephi ha una visualizzazione e un uso più semplice e avanzato.

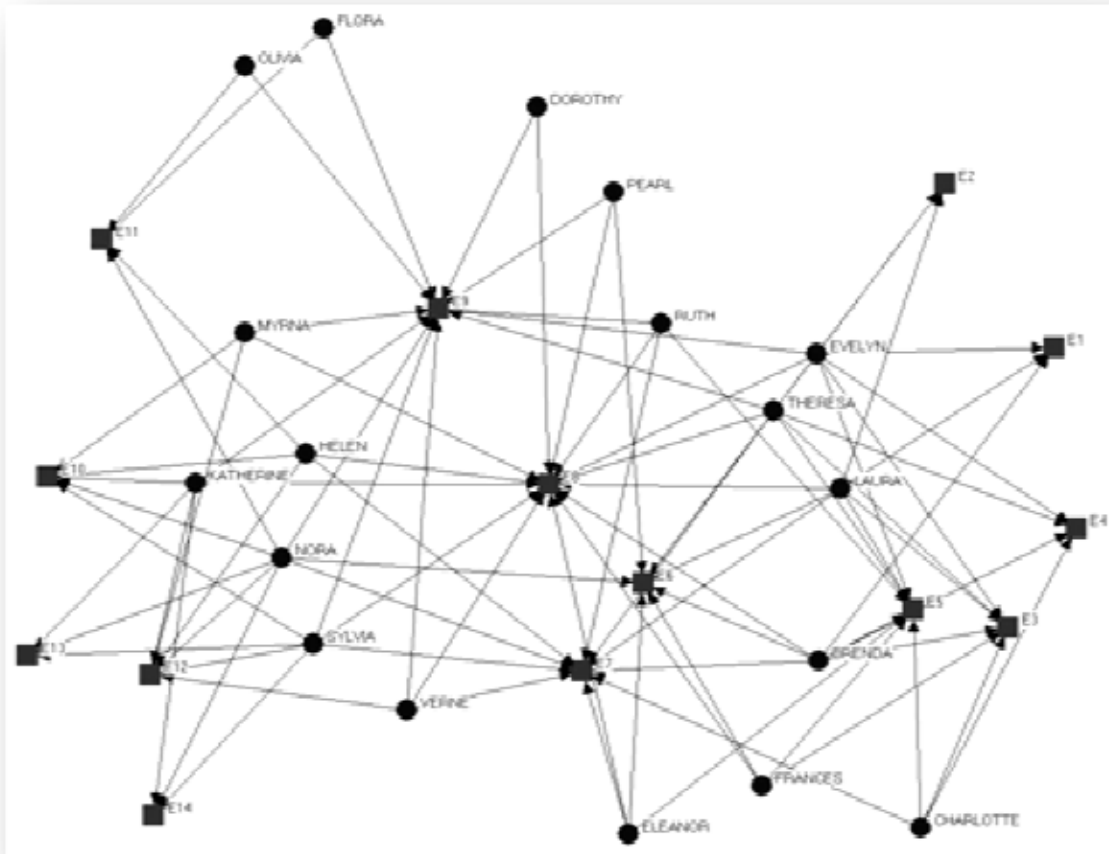


Figura 18.8. Davis data. Rappresentazione 2-Mode. Generazione automatica in UCINet.

Altri tool, oltre a quelli già visti al capitolo 13, sono: Pajek (<http://vlado.fmf.uni-lj.si/pub/networks/pajek/>) di cui un esempio può essere visto di seguito in Figura 18.9), ORA (<http://www.casos.cs.cmu.edu/projects/ora/>), NetworkX in Python (<http://networkx.github.io/>), Statnet (

project.org/web/packages/statnet/) suite di pacchetti in R, SocNetV (<http://socnetv.sourceforge.net/>), e molti altri. Per la visualizzazione: SoNIA (Social Network Image Animator, <http://www.stanford.edu/group/sonia/>), iGraph e SocioMetrica (<http://www.mdlogix.com/solutions/additional.html#row1>).

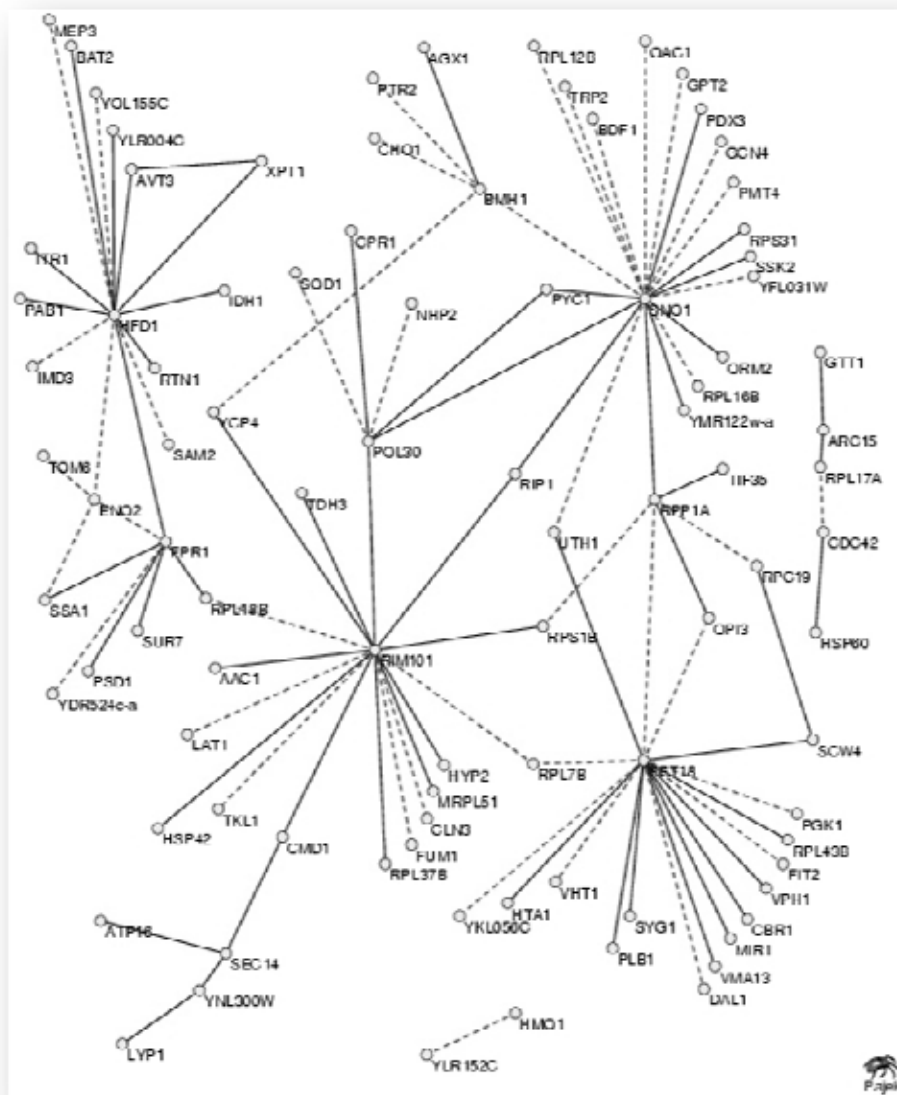


Figura 18.9. I nodi sono identificati dai nomi dei geni. Le linee intere denotano coppie positive, le tratteggiate coppie negative.

Nella creazione di modelli per lo studio di sistemi complessi, spesso, si fa largo uso di Matlab (già ampiamente visto) e Mathematica, due software molto utili che hanno parecchi moduli e specifiche per la creazione di modelli di reti complesse, simulazioni agent-based, PDE (partially differentially equations, ovvero equazioni differenziali alle derivate parziali) e altri. In generale consentono un ampio sviluppo con molte tecniche matematiche e statistiche. In letteratura, così come in questo volume, se ne fa largo uso.

Infine, una speciale menzione merita Python (in onore alla mitica serie televisiva *"Monty Python's Flying Circus"*), un linguaggio di programmazione liberamente usabile e distribuibile, anche a uso commerciale.

Sviluppato da Guido van Rossum, più di dieci anni fa, la popolarità di Python è cresciuta nel tempo e sta diventando sempre più una buona alternativa ai ben conosciuti C++ e Java.

Per comprendere la potenza di Python, si pensi che Peter Norvig, Direttore della Ricerca di Google, ha dichiarato che *"Python has been an important part of Google since the beginning, and remains so as the system grows and evolves. Today dozens of Google engineers use Python, and we're looking for more people with skills in this language"*.

Python gira sulla maggior parte di sistemi operative, anche perché ci sono versioni per .NET (IronPython), JVM (Jython), e cellular Nokia Serie 60; è anche supportato da Internet Communications

Engine (ICE). È inoltre possibile scrivere moduli di estensione in C o C++.

Un utile insieme di librerie di Python, che sarà ampiamente usato in questo volume, è NetworkX, *"a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks"* come si può leggere sul sito web ufficiale (sotto lo slogan *"high productivity software for complex networks"*).

18.2 Usare Python per creare modelli

Dal momento che in questo volume verrà fatto un ampio uso di Python, vedremo adesso qualche esempio di uso, su un Mac (il mio).

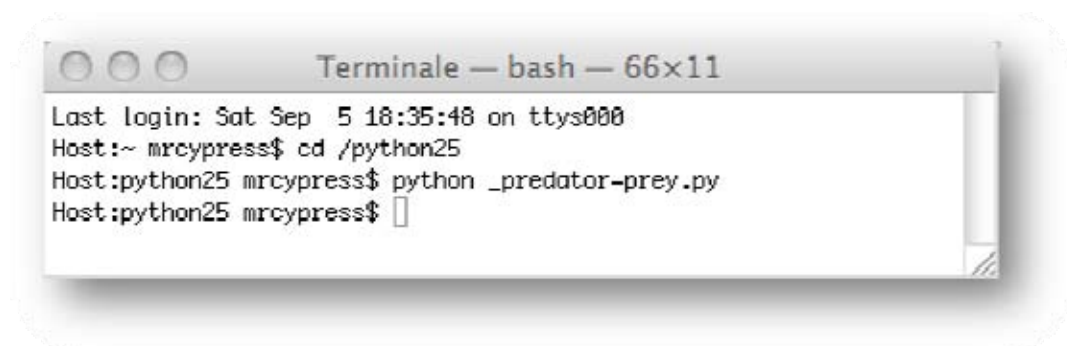


Figura 18.10. Il codice su un Mac può essere eseguito semplicemente dal terminale. Anche se non si dispone di un'interfaccia GUI (Grafic Unit Interface, o qualcosa come un'esecuzione con doppio click), è abbastanza facile capire la sintassi.

Un primo esempio è la creazione di un grafico con Python sul diagramma di biforcazione che abbiamo visto al capitolo precedente. Possiamo vedere il codice qui di seguito e il risultato della sua esecuzione in Figura 18.11.

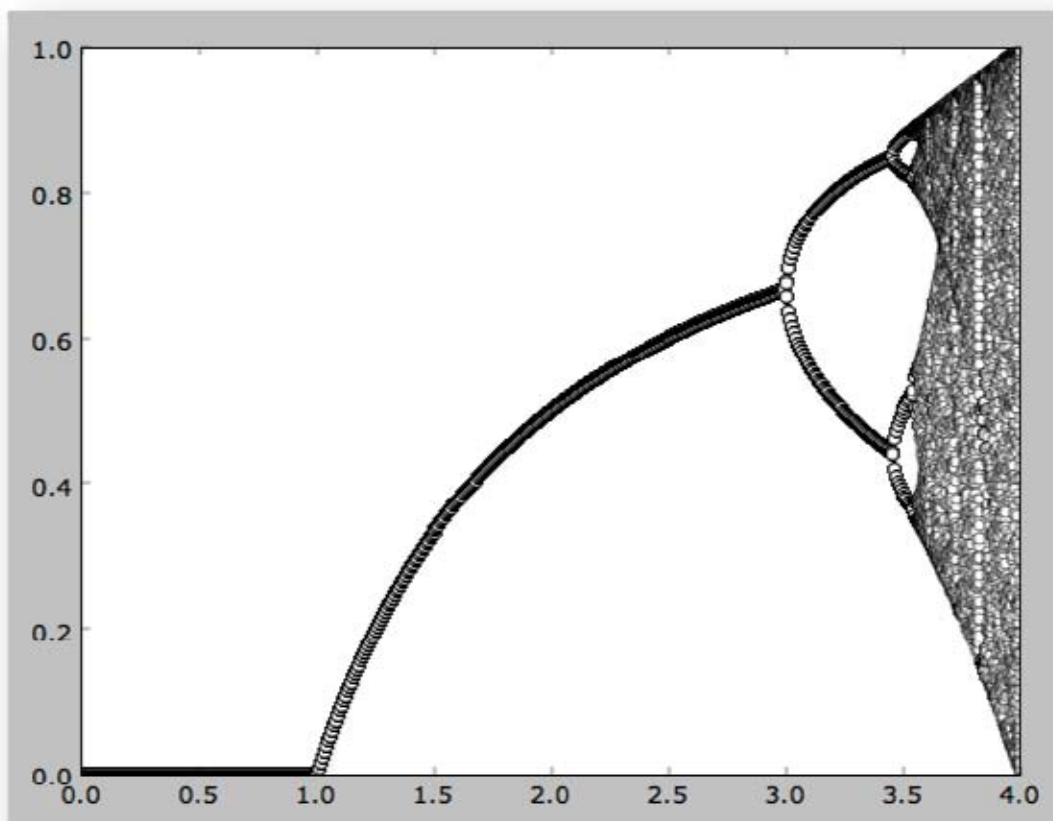


Figura 18.11. Diagramma di biforcazione ottenuto tramite codice.

```
import pylab as PL
def f(s): return a*s*(1-s)
s0 = 0.05
samplingStartTime = 500
```

```

sampleNumber = 100
A = []
S = []
for a in map(lambda i: 4.0*i/500.0, range(500 + 1)):
    s = s0
    for t in range(samplingStartTime):
        s = f(s)
    for t in range(sampleNumber):
        s = f(s)
        A.append(a)
        S.append(s)
PL.plot(A, S, 'wh')
PL.show()

```

Adesso vedremo un esempio un po' più complesso: un classico modello preda-predatore. Possiamo vedere, qui di seguito, il codice in Python. In Figura 18.12, l'esecuzione del codice al tempo $t=1$ (non rappresentiamo al tempo $t=0$ per vedere almeno un passo dalle condizioni iniziali), procedendo in Figura 18.13 al tempo $t=150$, poi in Figura 18.14 al tempo $t=300$ e infine in Figura 18.15 al tempo $t=500$.

Fisseremo le condizioni iniziali a 30 predatori e 100 prede; è ovviamente possibile cambiare ciascun parametro del modello per osservarne le differenze.

```

import random as RD
import pylab as PL
import scipy as SP
RD.seed()
width = 100
height = 100
rabbit = 1
fox = 0

```

```
initialRabbitPopulation = 100
rabbitReproductionRate = 0.1
rabbitPopulationLimit = 500
rabbitNoiseLevel = 2
initialFoxPopulation = 30
foxReproductionRate = 0.5
foxPopulationLimit = 500
foxNoiseLevel = 3
foxHungerLimit = 10
collisionDistance = 2
maxTime = 500
agents = []
for i in range(initialRabbitPopulation):
    newAgent = [RD.uniform(0, width), RD.uniform(0, height), rabbit,
0]
    agents.append(newAgent)
for i in range(initialFoxPopulation):
    newAgent = [RD.uniform(0, width), RD.uniform(0, height), fox, 0]
    agents.append(newAgent)
PL.ion()
PL.figure()
PL.hold(False)
PL.ioff()
PL.bone()
rabbitData = [initialRabbitPopulation]
foxData = [initialFoxPopulation]
def visualize(data, time):
    PL.subplot(1, 2, 1)
    if data == []:
        PL.cla()
    else:
        x = map(lambda a:a[0], data)
        y = map(lambda a:a[1], data)
        s = map(lambda a:a[2], data)
        PL.scatter(x, y, c = s, vmin = 0, vmax = 1)
    PL.axis('scaled')
    PL.axis([0, width, 0, height])
    PL.title('t = ' + str(time))
    PL.subplot(1, 2, 2)
    PL.plot(rabbitData)
    PL.hold(True)
    PL.plot(foxData)
```

```

PL.hold(False)
PL.draw()
PL.savefig('Project_images/predatorprey'+str(time)+'.png')
visualize(agents, 0)
CDsquared = collisionDistance ** 2
def clip(a, amin, amax):
    if a < amin: return amin
    elif a > amax: return amax
    else: return a
for t in range(maxTime):
    survivors = []
    newborns = []
    IDs = range(len(agents))
    for i in IDs:
        if agents[i][2] == rabbit:
            agents[i][0] += RD.gauss(0, rabbitNoiseLevel)
            agents[i][1] += RD.gauss(0, rabbitNoiseLevel)
        else:
            agents[i][0] += RD.gauss(0, foxNoiseLevel)
            agents[i][1] += RD.gauss(0, foxNoiseLevel)
            agents[i][0] = clip(agents[i][0], 0, width)
            agents[i][1] = clip(agents[i][1], 0, height)
    for i in IDs:
        if agents[i][2] == rabbit:    # rabbit's survival criteria
            for j in IDs:
                if agents[j][2] == fox:
                    if (agents[i][0]-agents[j][0])**2 + (agents[i][1]-
agents[j][1])**2 < CDsquared:
                        break
            else:
                survivors.append(agents[i])

        else:    # fox's survival criteria
            for j in IDs:
                if agents[j][2] == rabbit:
                    if (agents[i][0]-agents[j][0])**2 + (agents[i][1]-
agents[j][1])**2 < CDsquared:
                        agents[i][3] = 0    # yum!
                        survivors.append(agents[i])
                        break
            else:
                agents[i][3] += 1    # starving...

```

```

        if (agents[i][3] <= foxHungerLimit):
            survivors.append(agents[i])
    survivorsSpecies = map(lambda a:a[2], survivors)
    rabbitPopulation = survivorsSpecies.count(rabbit)
    foxPopulation = survivorsSpecies.count(fox)
    for i in range(len(survivors)):
        if survivors[i][2] == rabbit:
            if RD.random() < rabbitReproductionRate * (1.0 -
float(rabbitPopulation) / float(rabbitPopulationLimit)):
                newborns.append([survivors[i][0],
                                survivors[i][1],
                                survivors[i][2],
                                0])
        else:
            if survivors[i][3] == 0 and RD.random() <
foxReproductionRate * (1.0 - float(foxPopulation) /
float(foxPopulationLimit)):
                newborns.append([survivors[i][0],
                                survivors[i][1],
                                survivors[i][2],
                                0])
    agents = survivors + newborns
    agentSpecies = map(lambda a:a[2], agents)
    rabbitData.append(agentSpecies.count(rabbit))
    foxData.append(agentSpecies.count(fox))
    visualize(agents, t + 1)
PL.show()

```

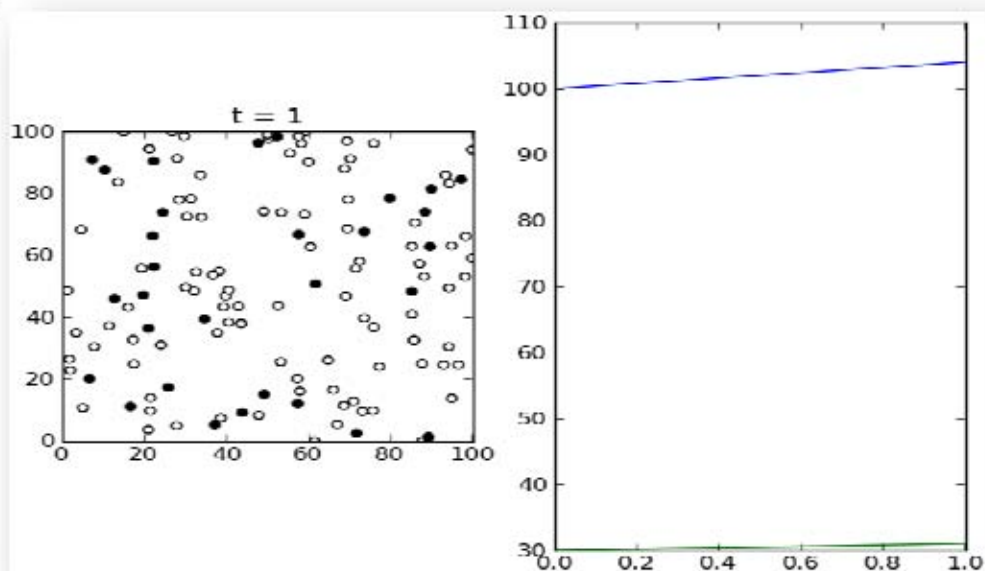


Figura 18.12. Esecuzione del codice in figura 3.28, al tempo $t=1$, è possibile vedere le condizioni iniziali e il primo step (passo).

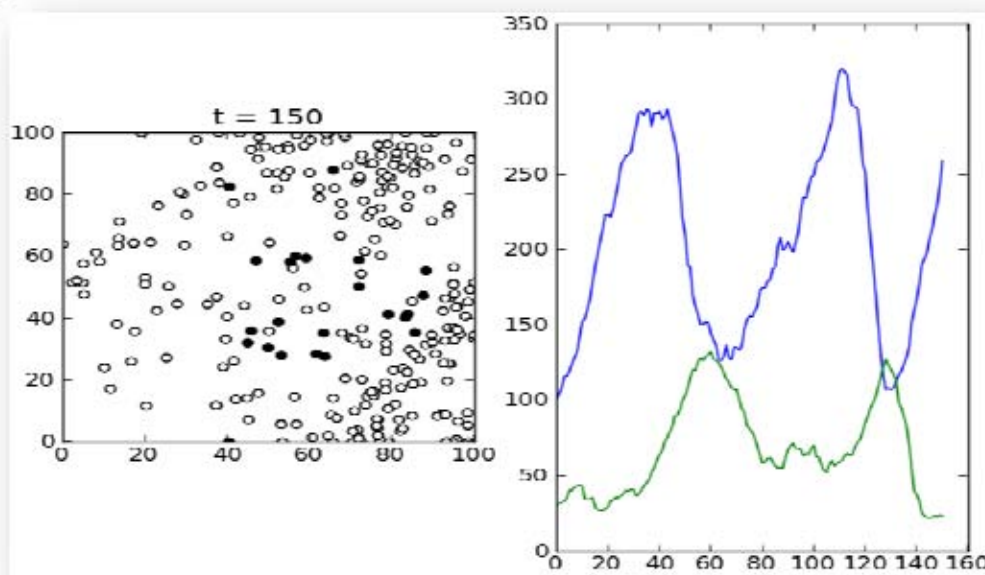


Figura 18.13. Execution of the code in figure 3.28, at time $t=150$.

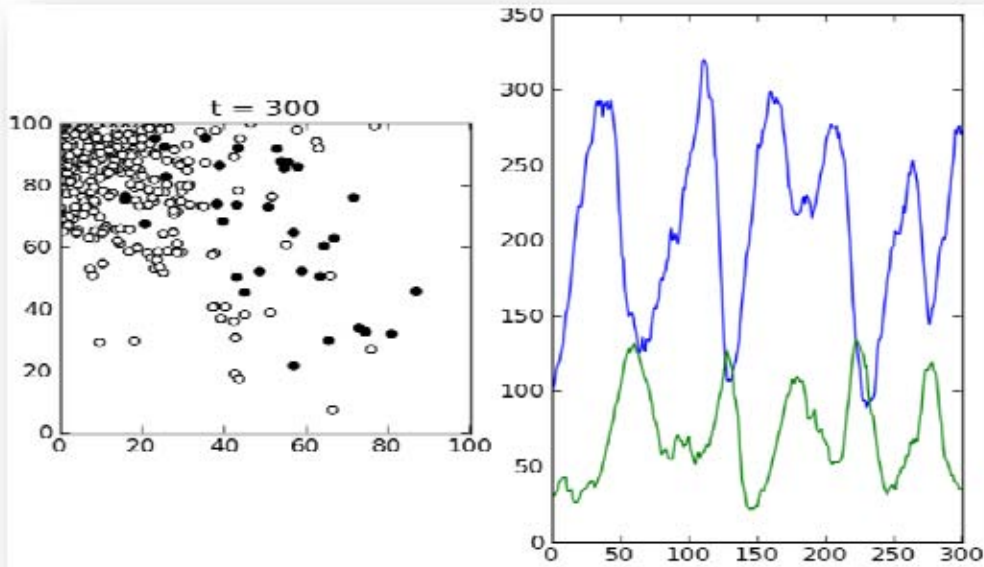


Figura 18.14. Esecuzione del codice in figura 3.28, al tempo $t=300$.

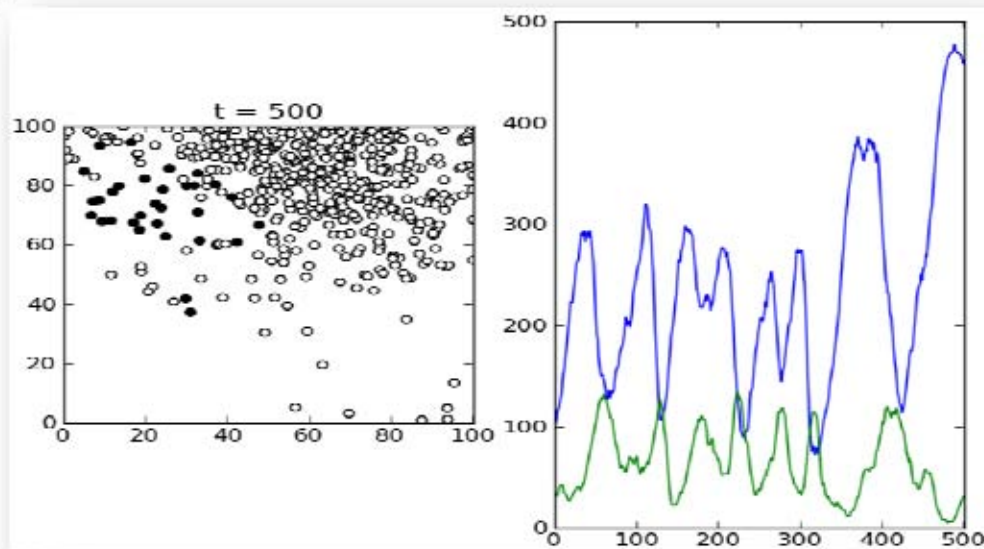


Figura 18.15. Esecuzione del codice in figura 3.28, al tempo $t=500$.

E' importante sottolineare che il codice eseguito è una simulazione basata su agenti e quindi non genererà sempre le stesse figure. Ciò significa che se rilanciate il codice con gli stessi parametri, ogni volta otterrete delle figure diverse, ma che comunque avranno proprietà emergenti simili. Proprio questa caratteristica rende i modelli agent-based molto adatti ai fenomeni psicologici, tipicamente complessi per loro natura e specie.

Microanalisi di simulazione tra agenti

In questo capitolo consideriamo le possibili interazioni di un sistema agent-based popolato da "affective agents", ovvero prendiamo in considerazione architetture per la creazione di agenti artificiali che incorporino regole e meccanismi automatici per considerare reazioni istantanee legate a strategie di tipo "fight or flight", aventi come meccanismo induttore le emozioni, in particolare rabbia e paura. Partendo da ipotesi sulla storia evolutiva delle emozioni, si tenta di definire queste in modo operativo, prendendo in considerazione il grande problema di mettere in rilievo le emozioni fondamentali per la sopravvivenza degli individui e delle specie, rispetto alle altre emozioni. Nasce qui il problema, psicologico e semantico, del reale significato operativo di ciò che definiamo "emozioni".

Inizialmente definiamo l'emozione come un elemento induttore o facilitatore di shortcuts comportamentali.

Questo è ottenuto, soprattutto per le emozioni collegate alla sopravvivenza immediata - paura, quindi fuga e ira quindi attacco - mediante circuiti brevi che collegano organi di senso, aree di genesi/smistamento e apparati motori.

Per emozioni non direttamente collegate alla sopravvivenza la questione si presenta assai più complessa in quanto si definiscono come "emozioni" molti stati e situazioni con caratteristiche differenti e talvolta lontane, sia sul piano neuropsicologico che su quello operativo (ossia dei risultati che ne possono derivare).

Ne consegue che è assai difficile modellizzare e rappresentare mediante agenti, situazioni, comportamenti e soprattutto le loro cause e conseguenze se non se ne ha una chiara visione per quanto attiene al "reale".

Una delle criticità maggiori è pertanto costituita dal trovare delle definizioni operative adeguate che ci guidino verso una rappresentazione puntuale di ciò che chiamiamo "emozioni", certamente lo sviluppo delle scienze (e.g. le scienze matematiche, dal calcolo infinitesimale in poi) è stato tale da consentire di rappresentare con conveniente approssimazione e da trattare con sufficiente accuratezza anche situazioni sfumate (fuzzy) e proprio qui sta l'abilità del ricercatore:

- 1) definire al meglio l'oggetto dell'esperimento;
- 2) scegliere le metodologie più adatte per trattare quell'oggetto;

3) essere intellettualmente onesto nell'interpretare i risultati.

19.1 Simulare le emozioni

Non è mia intenzione in questo volume, creare un modello per una corretta e operativa simulazione di un processo psicobiologico talmente complesso come le emozioni, quanto cercare di fornire un esempio e uno spunto di riflessione sull'operare, in termini di modelli, con strutture psicologiche complesse e non sempre ben gestibili con strumenti informatici e/o matematici.

Molti studiosi hanno analizzato i processi decisionali, soprattutto in economia. Vale la pena citare Daniel Kahneman "per aver integrato argomenti della ricerca psicologica con le scienze dell'economia, con particolare riguardo ai processi decisionali e di giudizio nelle incertezze" e Herbert Simon che al "modello della razionalità olimpica" (come egli stesso la definì) contrapponeva il concetto di razionalità limitata.

Prima di parlare di emozioni, siano esse umane o artificiali, bisogna capire il contesto in cui queste operano. In termini temporali (ma non è l'unico asse da considerare) parliamo di antecedenti e conseguenti:

- antecedenti sono l'ambiente e i suoi accadimenti, ma anche l'emotional elicitation e sicuramente anche i processi di attenzione (fortemente influenzati dalla sfera emotiva);

- conseguenti sono i processi di shortcut comportamentali con un effetto diretto sull'azione: è questa azione che noi analizziamo, studiamo e cerchiamo di comprendere ed è partendo da questa che molti studiosi hanno ipotizzato le loro teorie delle emozioni (e.g., le teorie top-down; al contrario delle bottom-up che analizzano le basi biologiche delle emozioni).

In termini non temporali ci sembra, come sopra accennato, utile suddividere le emozioni come segue:

- emozioni necessarie alla sopravvivenza del soggetto e della specie, qualunque essa sia: umana, animale o artificiale;
- emozioni non necessarie alla sopravvivenza dell'individuo e della specie.

Le emozioni necessarie alla sopravvivenza, potrebbero essere inserite in agenti artificiali. Nel caso di esseri umani, esempi di tali emozioni sono la paura e l'ira, strettamente correlate alla difesa e all'attacco. In agenti artificiali, con gli strumenti e le conoscenze oggi disponibili, è possibile considerare strategie di difesa e di attacco che molto somigliano agli effetti provocati dalle emozioni (così come già detto prima).

Consideriamo come esempio il mercato di borsa e le sue fluttuazioni. In particolare consideriamo le simulazioni di Ferraris-Cappellini e di Cipresso, presentate anni fa al Convegno WIVACE 2007. Nel primo caso vengono date agli agenti particolari

caratteristiche di memoria (ovvero gli agenti ricordano meno le informazioni ricevute – temporalmente – nei periodi centrali, la memoria segue l'andamento di una gaussiana rovesciata, quindi si ricordano meglio le code). Gli agenti stessi sono in grado di trasferire l'un l'altro le informazioni fino a giungere a conseguenze operative. Nel secondo caso vengono analizzati mercati in una situazione di iperinflazione. Si ipotizza che vi sia l'annuncio di un aumento dell'offerta di moneta, a partire da un certo momento successivo a quello dell'annuncio stesso. Effettivamente la Banca Centrale non cambia nulla, al momento dall'annuncio, ma comunque il livello dei prezzi subisce una variazione; questo accade probabilmente perché cambia la percezione degli agenti nei confronti della politica monetaria, quindi la loro attesa sul livello dei prezzi.

Ovviamente entrambe le simulazioni, pur non considerando le emozioni negli agenti, si prestano a un modello di questo tipo. Meccanismi di difesa e di attacco tra gli agenti, sulla base delle informazioni che "girano" sul mercato, potrebbero ben rappresentare un tessuto emotivo, certamente da ampliare, anche con meccanismi di selezione (così come fanno gli algoritmi genetici).

Includere emozioni non significa soltanto fare prendere, agli agenti, decisioni in modo rapido. Ad esempio, quando un agente incontra un ostacolo e "decide" che direzione prendere, come conseguenza dello scontro, sulla base di un criterio random o sulla base di un apprendimento avvenuto nell'ambiente, non viene usato

un criterio basato su intelligenza; e naturalmente per inserire emozioni in questo agente non è sufficiente fargli prendere questa decisione più in fretta.

Per considerare le emozioni a livello di agente bisogna innanzitutto dare un tessuto emotivo allo stesso. A livello computazionale questo potrebbe essere fatto con modelli che includano le seguenti caratteristiche degli agenti:

- agente con avversione al rischio (caratteristica emotiva: paura): se preferisce sempre ottenere con certezza il valore atteso di una data quantità aleatoria rispetto alla quantità aleatoria stessa;
- agente con neutralità al rischio (caratteristica emotiva neutra): se un agente è sempre indifferente tra valore atteso di una quantità aleatoria e la quantità aleatoria stessa;
- agente con propensione al rischio (caratteristica emotiva: ira/aggressività): se un agente preferisce sempre una data quantità aleatoria rispetto a ottenere il suo valore atteso con sicurezza.

La suddivisione non è certamente esaustiva, né (neanche lontanamente) esatta. Comunque ci serve per costruire, in modo "semplice", un "tessuto artificiale" che differenzi gli agenti con, ai due estremi della gaussiana, l'agente con massima avversione al rischio e quello con massima propensione al rischio. I passaggi "da difesa a paura" e "da attacco a ira" sono forzature che possono

essere fatte per nostra comodità (o meglio, per comodità di modellazione/simulazione).

Il processo di scelta/comportamento negli agenti artificiali

Nel processo di scelta/comportamento degli agenti intervengono fattori ambientali (e non solo) che influiscono, tramite un processo emozionale, che potrebbe – ad esempio – agire seguendo paradigmi di attacco e difesa, influenzando altri agenti che, a loro volta, adotteranno meccanismi simili.

Dall'elicitazione delle emozioni scaturiscono fenomeni emotivi che, sulla base di avversità/neutralità/propensione al rischio degli agenti, implicheranno comportamenti, i quali, essendo azioni all'interno di un ambiente, genereranno nuovi stimoli, emozioni, risposte.

Per quanto riguarda i rapporti tra simulazioni ed emozioni si possono considerare tre differenti prospettive:

- 1) *Recognition of Human Emotion*
- 2) *Expression of Emotional-like behavior*
- 3) *Modeling and Simulation of Human Behavior*

Le categorie qui elencate dovrebbero essere viste solo come riferimento generale di campi più specifici di ricerca.

Alcune considerazioni generali

- In questo volume, non consideriamo architetture che interagiscano con esseri umani: gli agenti sono intesi autonomi nell'ambiente simulato e la sola interazione che essi possono avere con l'utente è attraverso la visualizzazione dell'applicazione o anche attraverso un command-line shell. Ciò non di meno consideriamo che potrebbe essere molto interessante avere meccanismi di riconoscimento delle emozioni, nel caso in cui utenti umani avessero un ruolo di intervento nella simulazione.
- Il focus principale è orientato allo sviluppo di architetture per agenti che massimizzino la loro autonomia e adattabilità nell'ambiente: come sopra richiamato, le interazioni, con l'utente umano, allo stato attuale non sono considerate.
- Non siamo interessati neppure a creare *Believable Agents* o agenti che, in qualche modo, esprimano emozioni. Le architetture che proponiamo sono pensate per usare meccanismi emotivi, a livello di agente, correlati con le performance di altri agenti nell'ambiente. La rappresentazione visuale dell'agente è un aspetto secondario.
- Infine, nonostante l'utilizzo di scenari antropomorfici (almeno nelle intenzioni, con quante più caratteristiche dell'essere umano ci siano utili), in questo caso non si prendono in considerazione modelli di simulazione dell'essere umano.

Aspetti metodologici

Le fasi dello sviluppo metodologico possono essere definite genericamente come segue:

- studio dei fenomeni emotivi da un punto di vista funzionale, provando, in ogni caso, a esplorare possibili applicazioni pratiche all'interno di architetture Autonomous Agent;
- sviluppo di un modello basato su meccanismi emozionali, compatibile con varie teorie e modelli delle emozioni, tra intelligenza artificiale, psicologia e neuroscienze;
- sviluppo di un'architettura ad agenti ove meccanismi basati su emozioni forniscano un vantaggio operativo, in termini di performance, per agenti a risorse limitate che operano in ambienti complessi e dinamici.

Architettura emotional-agent-based

Innanzitutto è importante strutturare un'architettura in un contesto più generale, cioè all'interno dello schema Soggetti-Regole-Ambiente.

Questo è essenziale per la corretta implementazione-gestione di un'architettura basata su emozioni e agenti artificiali ed è il cardine stesso della struttura, rappresentando in modo ben strutturato le interazioni tra gli agenti e tra questi e l'ambiente.

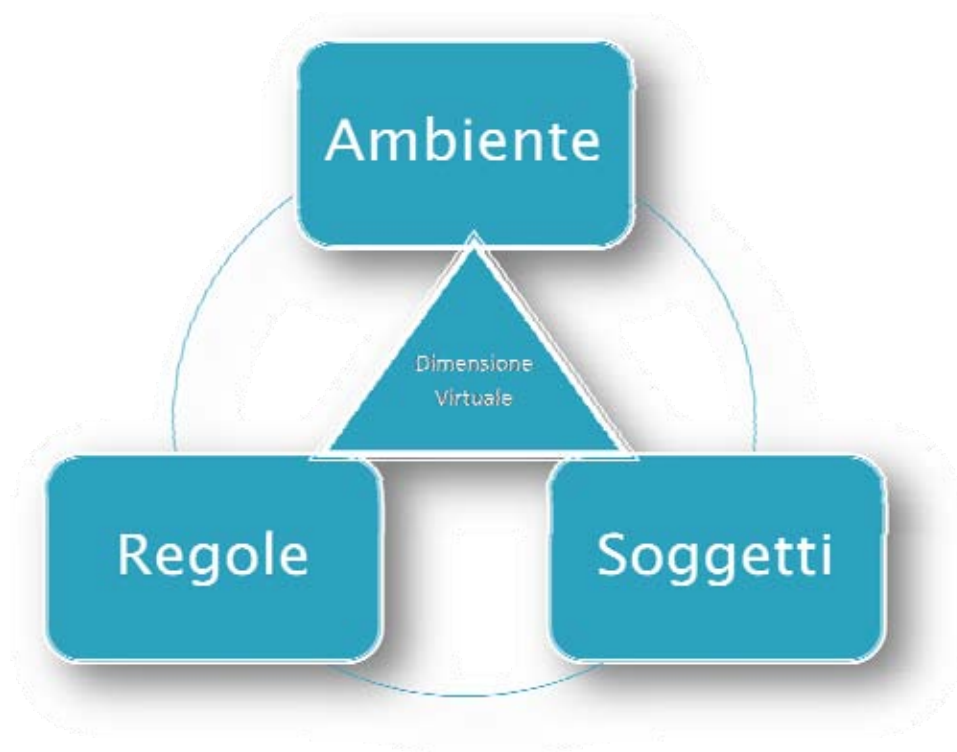


Figura 19.1. Schema concettuale per dei soggetti (anche artificiali) in un ambiente con delle regole. I tre elementi coesistono nel modello.

E' dunque essenziale creare un'architettura secondo alcune fasi:

- creare una struttura base degli agenti: avversità / neutralità / propensione al rischio, quindi sulla base di tali caratteristiche generare un set di azioni probabili (aggressività / difesa);
- creare un ambiente come contenitore di stimoli emotivi;
- creare risorse (anche per la sopravvivenza) inserendo una dinamica con predatori e prede;
- implementare condizioni (anche ambientali) tali da generare differenti caratteristiche tra gli agenti;

- operare una selezione (sulla base anche di algoritmi genetici ed evolutivi);
- strutturare meccanismi emotivi, strettamente correlati con il processo di selezione, facilitatori di euristiche comportamentali;
- strutturare processi di decision-making.

L'ambiente

Per sviluppare un sistema che stimoli emozioni, è necessario un ambiente, popolato da agenti artificiali, che possano includere emozioni. Per modellare diverse tipologie di stati emotivi è necessario soddisfare molte condizioni, sia a livello di sistema, sia a livello di agente. Gli agenti devono avere una rappresentazione esplicita o implicita di *obiettivi*, *standard* e *attitudini*. Inoltre, gli agenti dovrebbero anche essere in grado di "traslare" le osservazioni in termini di obiettivi, standard e attitudini.

In pratica quest'ultimo requisito ha le seguenti conseguenze sugli agenti:

- dovrebbero essere in grado di "verificare" se un evento soddisfi un particolare obiettivo o se ha un effetto positivo o negativo sulla probabilità che un particolare obiettivo possa essere soddisfatto; qui un ruolo fondamentale può essere svolto dal regret;

- dovrebbero poter fare delle inferenze e cambiare le loro aspettative sul futuro, come conseguenza dei predetti eventi. Le aspettative sugli obiettivi sono molto importanti per le emozioni;
- dovrebbero avere un qualche tipo di memoria sulle aspettative precedenti e dovrebbero essere in grado di comparare i nuovi eventi con quelli precedenti;
- la capacità di fare inferenza a livello di agente dovrebbe poter essere trasferita agli altri agenti tramite processi di trasferimento delle informazioni;
- dovrebbero poter comparare le proprie azioni e quelle degli altri agenti, seguendo e creando degli standard di comportamento condiviso;
- infine, dovrebbero poter comparare punti di vista esterni (degli altri agenti e dell'ambiente) e attitudini proprie (punti di vista interni).

Emozioni e apprendimento degli agenti

Abbiamo esaminato come l'ambiente abbia grande influenza sugli agenti e, conseguentemente, sulle emozioni degli stessi e sul comportamento.

È comunque emerso un fondamentale problema da considerare nella costruzione degli agenti: l'apprendimento.

Da molti anni gli studiosi di intelligenza artificiale (e non solo) si occupano di implementare algoritmi e modelli (reti neurali) per l'apprendimento dell'agente.

È naturalmente nostra intenzione considerare (così come accade nell'essere umano) le emozioni anche a livello di apprendimento degli agenti.

Gli strumenti sono molti e per la maggior parte legati alle reti neurali, ma non solo. Comunque, qualunque strumento informatico/algoritmico/ matematico si utilizzi, le emozioni possono (e devono) entrare a pieno titolo nella fase di modellazione del processo di apprendimento.

Le fasi da considerare nel processo di apprendimento sono:

- fase di apprendimento ambientale: è la fase in cui l'agente recepisce informazioni dall'ambiente e sull'ambiente, seguendo determinate regole;
- fase di interazione tra agenti: entrando in contatto tra loro, gli agenti, apprendono, modificano, ri-apprendono,... le regole
- fase di interazione avanzata: tramite stimoli, ambientali e non, gli agenti apprendono e sviluppano strategie legate all'ambiente, alle risorse e agli altri agenti, creando così regole di sopravvivenza e, soprattutto, ruoli tra agenti

Da un punto di vista funzionale, il principale ruolo del *sistema* è quello di convertire un evento con particolari proprietà in un nuovo stato emotivo.

Per poter effettuare tale distinzione sono necessarie due fasi:

- 1) una fase consiste nel calcolo dello stato emotivo, ovvero nell'uso di una rappresentazione numerica per il significato emotivo di un evento. Il calcolo è basato, in prima analisi, sull'apprendimento effettuato dall'agente nel passato. Successivamente questo calcolo va normalizzato (con una rete neurale) per essere confrontato con gli altri eventi;
- 2) un'altra fase è la valutazione del significato emotivo dell'evento; ad esempio, se un agente "vede" una risorsa dovrà effettuare una valutazione sulla base di un "discriminatore", estremamente variabile nello spazio e nel tempo e fortemente influenzato dagli stati emotivi dell'agente stesso.

Criticità

Da molti anni gli studiosi si occupano di "emozioni e AI". Già, nel 1988, Pfeifer, scrive *Artificial Intelligence Models of Emotion* passando in rassegna molti modelli AI delle emozioni. Successivamente vedono la luce molti modelli basati sulle tecniche più innovative e, spesso, su quelle più in voga. Molti modelli, davvero eleganti e ben costruiti, sono caduti nell'oblio, probabilmente solo perché non hanno avuto la fortuna di trovare l'interesse dei molti. Uno su tutti, secondo me da considerare e aggiornare, è quello di El-Nasr & Yen (1988), dove viene

utilizzata la logica fuzzy, una tecnica molto utile nel dominio delle emozioni.

Oggi le discipline *Recognition of Human Emotion*, *Modeling and Simulation of Human Behavior* e *Expression of Emotional-like behavior*, offrono molti contributi, aggiornandosi con le moderne teorie delle emozioni.

Attualmente, i contributi che considerano le emozioni nei modelli e nelle simulazioni agent-based sono molto pochi. C'è un vivo interesse, per questo campo, ma si sono presentate molte difficoltà, legate alla definizione stessa di emozioni e all'oggettivare le emozioni. Per quel che concerne quest'ultimo punto, comunque, va detto che i grossi avanzamenti fatti dall'Affective Computing, che abbiamo già visto nel capitolo 14, grazie a rilevatori sempre più portatili e a tecnologie sempre più sofisticate (capaci di registrare i segnali EEG, ECG, EMG, GSR, BVP, Respiration, ecc., che abbiamo visto al capitolo 2), hanno permesso di meglio comprendere complessi processi emotivi.

Le difficoltà di passaggio dalla rappresentazione del modello alla sua architettura e simulazione, sono strettamente legate alla natura stessa del problema. Inoltre, le emozioni, sono ovviamente difficili da rappresentare, qualunque approccio si segua. Comunque bisogna sempre ricordare che le emozioni sono state e continuano a essere essenziali per la sopravvivenza delle specie e potrebbero rappresentare un importante paradigma per l'agent-based.

Un modello Computazionale in Reti Complesse

Nei capitoli precedenti di questa terza parte abbiamo tracciato le varie teorie e gli strumenti per potere realizzare un primo modello computazionale, che non è certamente un modello definitivo o effettivo di funzionamento reale, ma traccia comunque un percorso per poter effettuare modelli meno generici e più sofisticati.

Purtroppo non si trovano in letteratura molti modelli computazionali legati alla psicologia, se non alcuni più inerenti alle neuroscienze che comunque esulano da una prima trattazione in psicomatria computazionale. Esistono anche pochi modelli matematici (non necessariamente computazionali) che affrontano le complesse dinamiche della psicologia. In questo capitolo creeremo un modello delle emozioni nel framework concettuale della psicologia della complessità. Il semplice modello, in particolare, simulerà la diffusione di un informazione in merito ai

prezzi in un contesto di iperinflazione. Ho scelto il contesto dell'iperinflazione per due motivi. Innanzitutto ha più senso parlare di dinamiche comunicazione influenzate da processi emotivi se si considera una popolazione più propensa al panico, come la storia ci ha tristemente insegnato con i fenomeni della tesaurizzazione, della corsa agli sportelli bancari e altre dinamiche sociali fortemente psicologiche, tipiche in situazioni di crisi economiche. La seconda motivazione di questa scelta è la disponibilità di un modello matematico per le economie in iperinflazione (modello di Cagan) che possiamo didatticamente modificare per includere agenti artificiali, eseguire alcune simulazioni e trarre alcune prime semplici interpretazioni.

In regime economico neoclassico, nel caso di agente rappresentativo, ogni singolo agente dell'economia si trova nella stessa situazione. Questo, però, non è ciò che accade nella realtà. In regime di agenti eterogenei, ogni singolo agente ha una situazione differente e l'equilibrio totale è dato dalla somma totale degli effetti. Pertanto il risultato basato su aggregati è diverso dalla somma dei singoli risultati.

20.1 Il modello matematico classico

In attuali lavori sui modelli di politica monetaria, tra cui Howitt, P. e Leijonhufvud, si è provato a concentrare l'attenzione sul Consumption Capital Asset Pricing Models (CCAPM), lavorando

sul Santa Fe Artificial Stock Market, SF-ASM: *"The basic objective of the SF-ASM is to understand the behavior of an environment of evolving trader behavior, where prediction strategies compete against each other"* (LeBaron).

Il nostro lavoro concentra l'attenzione sullo stesso tipo di metodologia e ambiente di lavoro. Tuttavia il modello di riferimento per noi è basato sulle economie in situazioni di iperinflazione e, in particolare, sulla formazione del prezzo basato su una prima componente detta "fondamentale" in quanto associata ad un fondamentale dell'economia (il saldo monetario) e una seconda componente detta "bolla" dipendente dalle aspettative sul livello dei prezzi futuri degli agenti.

Il Modello analitico

Consideriamo adesso economie in situazione di iperinflazione. Partiamo dall'equazione che descrive la domanda di moneta (Barro):

$$\frac{M_t^d}{P_t} = (1 + i_{t+1})^{-\eta} Y_t^\beta \quad (1)$$

Con M_t^d domanda di moneta al tempo t , P_t Prezzo al tempo t e Y_t reddito al tempo t .

$\eta > 0$ è l'elasticità della domanda di moneta al tasso d'interesse.

$\beta > 0$ è l'elasticità della domanda di moneta al reddito.

Passiamo ai logaritmi

$$\ln M_t^d - \ln P_t = -\eta \ln(1 + i_{t+1}) + \beta \ln Y_t \quad (2)$$

Assumendo che $(1 + i_{t+1}) = (1 + r_{t+1})(1 + \pi_{t+1})$

dove $(1 + \pi_{t+1}) = 1 + \frac{P_{t+1} - P_t}{P_t} = \frac{P_{t+1}}{P_t}$, π rappresenta la variazione di prezzi.

Allora varrà:

$$\ln M_t^d - \ln P_t = -\eta \left(\ln(1 + r_{t+1}) + \ln \frac{P_{t+1}}{P_t} \right) + \beta \ln Y_t \quad (3)$$

$$\ln M_t^d - \ln P_t = -\eta \ln(1 + r_{t+1}) - \eta \ln P_{t+1} + \eta \ln P_t + \beta \ln Y_t \quad (4)$$

Sostituendo i log delle lettere maiuscole con le lettere minuscole:

$$m_t^d - p_t = \underbrace{-\eta \ln(1 + r_t) + \beta y_t}_{\text{parte reale}} - \underbrace{\eta(p_{t+1} - p_t)}_{\text{parte inf lazionistica}} \quad (5)$$

Abbiamo dunque diviso la domanda di moneta in due componenti: una di natura reale e una di natura inflazionistica.

A questo punto sono d'obbligo due considerazioni:

- 1) In periodi di iperinflazione le variazioni di prezzi sono molto più grandi di quelle delle variabili reali. Quindi la seconda componente influenzerà in maniera significativa la domanda di moneta.
- 2) Il tasso di crescita dei prezzi in $t+1$ non è noto in t . Dovremo quindi considerare le aspettative che su di esso viene formulata in t .

Pertanto, volendo adottare l'equazione di cui sopra a situazioni di iperinflazione e prendendo in considerazione le aspettative (Mankiw e Ricardo), scriviamo:

$$m_t^d - p_t = -\eta E_t(p_{t+1} - p_t) \quad (6)$$

Quest'ultima è l'equazione che descrive la domanda di moneta nel modello (detto appunto modello di Cagan).

Per trovare l'equilibrio all'interno del modello, è necessario considerare l'offerta di moneta.

Ipotizziamo che essa sia fissata esogenamente dalla Banca Centrale, ossia $m_t^s = m_t$.

Condizione d'equilibrio nel mercato della moneta:

$$\begin{aligned}m_t^s - p_t &= m_t^d - p_t \Rightarrow m_t - p_t = -\eta E_t(p_{t+1} - p_t) & (7) \\m_t - p_t &= -\eta E_t p_{t+1} + \eta p_t \\(1 + \eta)p_t &= m_t + \eta E_t p_{t+1}\end{aligned}$$

Quindi:

$$p_t = \frac{1}{1 + \eta} m_t + \frac{\eta}{1 + \eta} E_t p_{t+1} \quad (8)$$

Per risolvere questa equazione alle differenze stocastica è necessario specificare il modo con cui gli agenti formulano le aspettative (Sargent).

Nel caso di agenti eterogenei avremmo bisogno di un'equazione di prezzo per ogni agente: infatti ognuno di questi genera le proprie aspettative sui prezzi basandosi sul proprio comportamento / apprendimento.

Dobbiamo dunque ricorrere alle simulazioni e, onde evitare la complessità della simulazione ad equazioni, sarà necessario usare tecniche basate su agenti, ovvero utilizziamo un approccio bottom-up (ACE) (Tesfatsion). Ad una maggiore eterogeneità degli agenti corrisponderà una maggiore emergenza delle proprietà macroeconomiche.

Ipotizziamo che le aspettative siano razionali. Quindi gli agenti non commettono errori sistematici ossia ragionano in maniera

coerente col modello – nella formalizzazione del processo di formazione delle aspettative si utilizzano le stesse equazioni del modello. Quindi gli agenti utilizzano nella previsione tutte le informazioni a loro disponibili. Nel nostro caso specifico:

$$E_t(p_{t+1}|\Omega) = E_t p_{t+1} \quad (9)$$

Calcoliamo, dunque, la soluzione al modello (8). Portando avanti di un periodo l'espressione, avremo:

$$p_{t+1} = \frac{1}{1+\eta} m_{t+1} + \frac{\eta}{1+\eta} E_{t+1} p_{t+2} \quad (10)$$

Per la legge delle aspettative iterate:

$$E[E(p_{t+1}|\Omega_{t+1})|\Omega_t] = E(p_{t+1}|\Omega_t) = E_t p_{t+1} \quad (11)$$

Quindi

$$E_t p_{t+1} = \frac{1}{1+\eta} E_t m_{t+1} + \frac{\eta}{1+\eta} E_t p_{t+2} \quad (12)$$

$$p_t = \frac{1}{1+\eta} m_t + \frac{\eta}{1+\eta} [E_t p_{t+1}] \quad (13)$$

$$p_t = \frac{1}{1+\eta} m_t + \frac{\eta}{1+\eta} \left[\frac{1}{1+\eta} E_t m_{t+1} + \frac{\eta}{1+\eta} E_t p_{t+2} \right] \quad (14)$$

$$p_t = \frac{1}{1+\eta} m_t + \frac{1}{1+\eta} \frac{\eta}{1+\eta} E_t m_{t+1} + \left(\frac{\eta}{1+\eta} \right)^2 E_t p_{t+2} \quad (15)$$

Iterando, quest'ultima per t periodi:

$$p_t = \frac{1}{1+\eta} \sum_{s=t}^T \left(\frac{\eta}{1+\eta} \right)^{s-t} E_t m_s + \left(\frac{\eta}{1+\eta} \right)^T E_t p_{t+T} \quad (16)$$

Per $T \rightarrow \infty$:

$$p_t = \underbrace{\left[\frac{1}{1+\eta} \sum_{s=t}^{+\infty} \left(\frac{\eta}{1+\eta} \right)^{s-t} E_t m_s \right]}_{p_t^f: \text{fondamentale}} + \underbrace{\left[\lim_{T \rightarrow \infty} \left(\frac{\eta}{1+\eta} \right)^T E_t p_{t+T} \right]}_{b_t: \text{bolla}} \quad (17)$$

Abbiamo quindi ottenuto una soluzione generale scomponibile in una prima componente detta "fondamentale" in quanto associata ad un fondamentale dell'economia: il saldo monetario; e una seconda componente detta "bolla" dipendente dalle aspettative sul livello dei prezzi futuri degli agenti.

Prezzo generale = Prezzo fondamentale + bolla ovvero: $p_t^g = p_t^f + b_t$.

In tale situazione, pertanto, il prezzo generale dipende dalle aspettative degli agenti sui prezzi e sulla politica monetaria e, considerando agenti eterogenei, va considerata sia l'emergenza scaturente sia le relazioni tra gli agenti e le modificazioni intervenute negli agenti tramite cambi di fase, implementando così anche un approccio di tipo top-down.

Ci chiediamo ora quando siano ammissibili dei prezzi diversi da quello fondamentale. Ciò equivale a determinare quale tipo di "bolla" sia compatibile con la condizione d'equilibrio (17) - sopra enunciata.

Per ogni prezzo deve valere:

$$p_t = \frac{1}{1+\eta} m_t + \frac{\eta}{1+\eta} E_t p_{t+1} \quad (18)$$

Quindi per il prezzo fondamentale vale:

$$p_t^f = \frac{1}{1+\eta} m_t + \frac{\eta}{1+\eta} E_t p_{t+1}^f \quad (19)$$

Mentre per il prezzo generale:

$$p_t^g = \frac{1}{1+\eta} m_t + \frac{\eta}{1+\eta} E_t p_{t+1}^g \quad (20)$$

Ossia:

$$p_t^f + b_t = \frac{1}{1+\eta} m_t + \frac{\eta}{1+\eta} E_t [p_{t+1}^f + b_{t+1}] \quad (21)$$

Quindi, sottraendo p_t^f ad entrambi i membri:

$$[p_t^f + b_t] - p_t^f = \frac{1}{1+\eta} m_t + \frac{\eta}{1+\eta} [E_t p_{t+1}^f + E_t b_{t+1}] + \quad (22)$$

$$-\underbrace{\frac{1}{1+\eta} m_t + \frac{\eta}{1+\eta} E_t p_{t+1}^f}_{p_t^f}$$

Ovvero

$$b_t = \left(\frac{\eta}{1+\eta} \right) E_t b_{t+1} \Rightarrow E_t b_{t+1} = \left(\frac{1+\eta}{\eta} \right) b_t \quad (23)$$

La soluzione di questa equazione alle differenze omogenea è

$$b_t = \left(\frac{1+\eta}{\eta} \right)^t b_0 \quad (24)$$

Lo scenario simulativo in assenza di bolle

Ipotizziamo ora che in t_0 vi sia l'annuncio di un aumento dell'offerta di moneta a partire da T , con $T > t_0$.

$$m_t = \begin{cases} \bar{m} & \forall t < T \\ \bar{m}' & \forall t \geq T \end{cases} \quad (25)$$

$$\text{dove } \bar{m}' = \bar{m} + \Delta m$$

la percezione degli agenti nei confronti della politica monetaria, e dunque la loro attesa riguardo il livello dei prezzi, cambia a seconda del momento in cui essi si pongono. Avendo considerato lo scenario in assenza di bolle, avremo:

$$p_t = \frac{1}{1+\eta} \sum_{s=t}^{\infty} \left(\frac{\eta}{1+\eta} \right)^{s-t} E_t m_s \quad (26)$$

Prima di t_0 vale $E_t m_s = \bar{m}$ per ogni s . Prima dell'annuncio gli agenti credono che la politica monetaria non subirà variazioni e che lo stock di moneta emesso rimanga sempre \bar{m} . Dunque, $\forall t < t_0$:

$$\begin{aligned}
p_t &= \frac{1}{1+\eta} \sum_{s=t}^{\infty} \left(\frac{\eta}{1+\eta} \right)^{s-t} \bar{m} = \bar{m} \frac{1}{1+\eta} \sum_{s=0}^{\infty} \left(\frac{\eta}{1+\eta} \right)^s = & (27) \\
&= \bar{m} \left(\frac{1}{1+\eta} \right) (1+\eta) = \bar{m}
\end{aligned}$$

Dal momento in cui viene applicata la nuova politica monetaria, invece, gli agenti credono che lo stock di moneta emesso sarà sempre \bar{m}' . Quindi $E_t m_s = \bar{m}'$ per ogni s . E $\forall t \geq T: p_t = \bar{m}'$.

Tra t_0 e T , invece, sarà:

$$\begin{aligned}
p_t &= \frac{1}{1+\eta} \left[\sum_{s=t}^{T-1} \left(\frac{\eta}{1+\eta} \right)^{s-t} \bar{m} + \sum_{s=T}^{\infty} \left(\frac{\eta}{1+\eta} \right)^{s-t} \bar{m}' \right] = & (28) \\
&= p_t = \frac{1}{1+\eta} \left[\sum_{s=t}^{\infty} \left(\frac{\eta}{1+\eta} \right)^{s-t} \bar{m} - \sum_{s=T}^{\infty} \left(\frac{\eta}{1+\eta} \right)^{s-t} \bar{m} + \right. \\
&\quad \left. + \sum_{s=T}^{\infty} \left(\frac{\eta}{1+\eta} \right)^{s-t} \bar{m}' \right]
\end{aligned}$$

$$p_t = \frac{1}{1+\eta} \left[\sum_{s=t}^{\infty} \left(\frac{\eta}{1+\eta} \right)^{s-t} \bar{m} + (\bar{m}' - \bar{m}) \sum_{s=T}^{\infty} \left(\frac{\eta}{1+\eta} \right)^{s-t} \right] \quad (29)$$

$$p_t = \frac{1}{1+\eta} \left[\bar{m}(1+\eta) + (\bar{m}' - \bar{m}) \left(\frac{\eta}{1+\eta} \right)^{T-t} \underbrace{\sum_{s=0}^{\infty} \left(\frac{\eta}{1+\eta} \right)^s}_{1+\eta} \right] \quad (30)$$

$$p_t = \bar{m} + (\bar{m}' - \bar{m}) \left(\frac{\eta}{1+\eta} \right)^{T-t} \quad (31)$$

Al crescere di t , p_t sale, per ogni $t_0 \leq t < T$.

Stiamo considerando un intervallo $[t_0, T)$. Valutiamo ora qual è il livello dei prezzi all'inizio dell'intervallo, cioè in t_0 :

$$p_0 = \bar{m} + (\bar{m}' - \bar{m}) \left(\frac{\eta}{1+\eta} \right)^T \quad (32)$$

Tale espressione segnala che, al momento dell'annuncio, il livello di prezzi subisce un salto discreto.

Essendo $\frac{\eta}{1+\eta} < 1$, possiamo fare un'altra considerazione. Tanto maggiore è T , ossia con quanto più anticipo è dato l'annuncio di un futuro aumento di m , tanto minore è p_0 , ovvero è più piccolo il salto immediato di prezzi, come si può vedere in Figura 20.1.

Nella simulazione oltre ad influire le scelte di T , che regola l'annuncio un futuro aumento di m , viene considerato lo scambio di informazioni tra agenti eterogenei. Si possono avere tanti equilibri di prezzo quanti sono gli agenti e questo implica che in molti casi l'effetto che ha l'annuncio di aumento di m e il tempo T in cui è effettuato tale annuncio contrasta con l'effetto che si ottiene con un modello che si basa sull'agente rappresentativo.

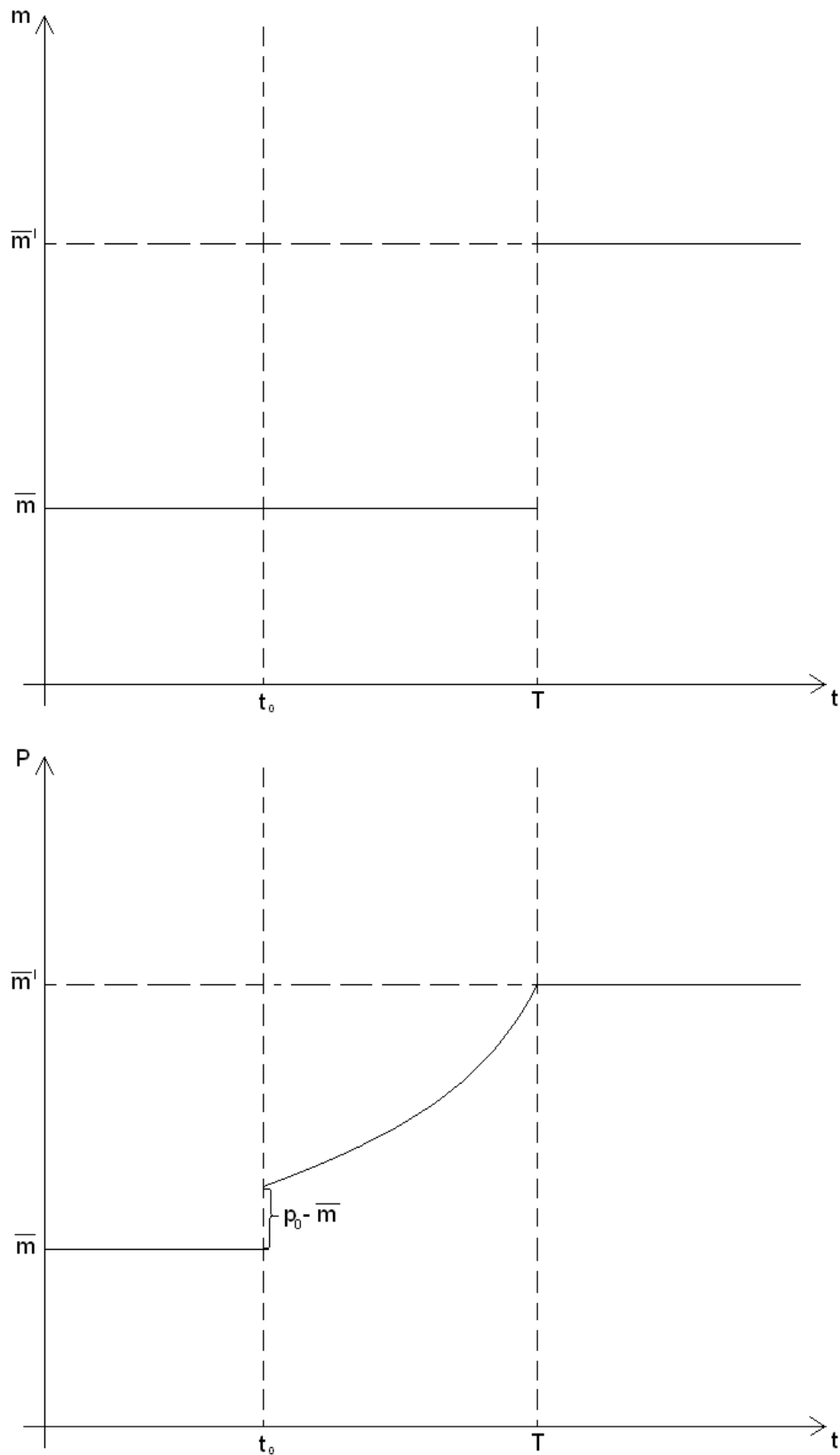


Figura 20.1. Una rappresentazione grafica del "salto discreto".

Complex networks

Le complex networks, sono la colonna vertebrale dei sistemi complessi; l'anatomia delle reti è importante per capire come la struttura viene influenzata. La letteratura fornisce svariati esempi:

- networks di reti interpersonali: explaining patterns of interactions among people of groups Friendship (Rapoport & Horvath, 1961; Milgram, 1967); Marriage (Padgett & Ansell, 1993); Job-market interactions (Granovetter, 1974);
- analisi statistica della topologia di rete: Albert & Barabási (2002), Newman (2003), Pastor-Satorras & Vespignani (2004), Dorogovtsev & Mendes (2003); properties of real-world technological, biological and information networks; WWW and the Internet, peer-to-peer networks, power grids, train routes, airline connections, electronic circuits, metabolism, protein; interactions, neural networks.
- analisi empirica di reti sociali ed economiche: scientific co-authorship (Newman, 2001) and citation (Redner, 1998); telephone calls (Aiello et al., 2000); email exchanges (Kossinets and Watts, 2006); sexual relationships (Liljeros et al., 2001); knowledge spill-overs among firms in industrial clusters (Giuliani and Bell, 2001); market investment (Battiston and Catanzaro, 2004); patent citation and innovation networks (Breschi and Lissoni, 2001; Ahuja, 2000); firm alliance formation (Gulati, 1998; Garcia Pont and Nohria, 2002); R&D teams and other within-firm networks

(Reagans and co-authors, 2001, 2004); social capital (Walker, Kogut, and Shan, 1997); company ownership and control (Garlaschelli et al., 2005); financial networks (Kullman et al., 2001); bank-firm relationships (De Masi et al., 2007).

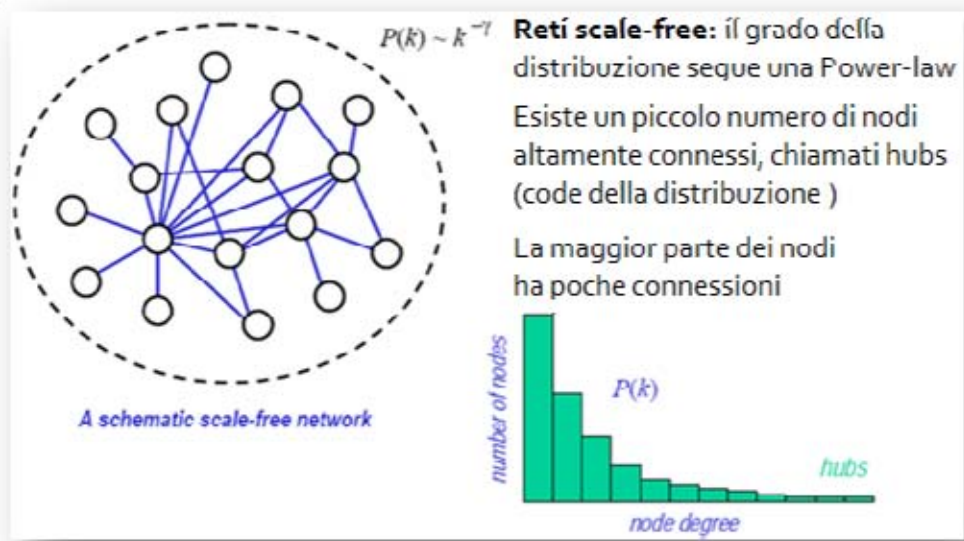
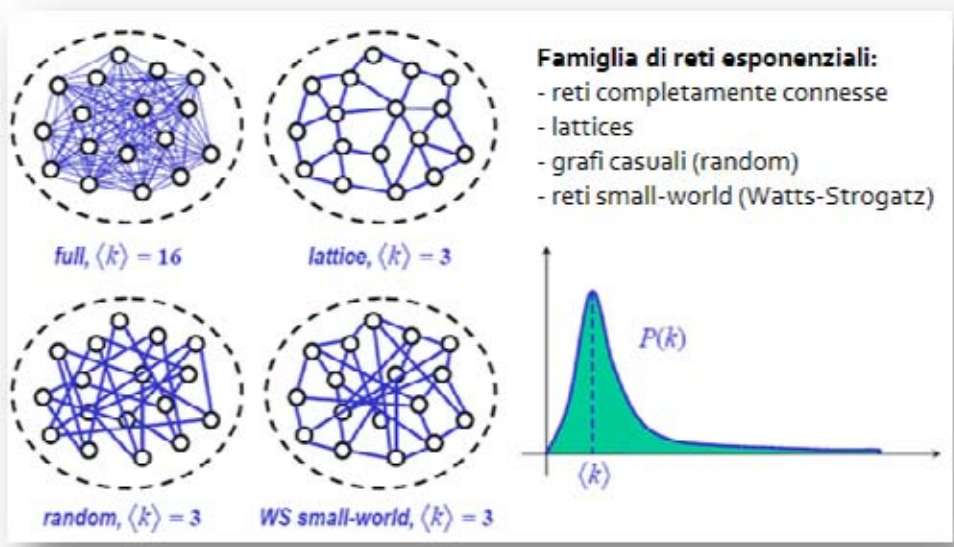


Figura 20.2. Reti esponenziali e reti Scale-free.

Per poter condurre le simulazioni di questo modello di comunicazione ho usato i dati di circa 60 studenti, allo scopo di poter inserire negli agenti degli indici di variabilità che dividessero la popolazione simulata su due livelli di arousal, come modello di esempio. In particolare sono stati usati i seguenti sensori:

- due canali EEG (Electroencephalography), posizionati sullo scalpo in corrispondenza della corteccia orbitofrontale, la quale viene considerata coinvolta nei processi di decision-making;
- un canale Blood Volume Pulse (BVP);
- un canale Galvanic Skin Response (GSR);
- un canale di respirazione toracica.

Questi sensori hanno fornito indici coerenti con le situazioni di relax, stress ed engagement dei soggetti. Successivamente i parametri sono stati adattati per essere dati in input agli agenti artificiali, secondo ben definite regole di comportamento.

20.2 Il Modello computazionale

Nel modello agent-based, definiamo l'interazione tra agenti. Ogni agente ha una variabile "self state" che fornisce informazioni agli altri agenti. Inoltre, gli agenti hanno una lista dei vicini che rappresenta la lista degli agenti a cui sono connessi tramite la rete.

Definiamo inoltre un'altra variabile che contiene l'"emotional state" dell'agente.

```
class agentClass:
    def __init__(self, id, state_agent, neighbors, emotion,
emotion_degree):
        self.id = id
self.state = state_agent # knows or not
self.neighbors = neighbors # list of neighbors
self.emotion_degree = emotion_degree # relax or stress
```

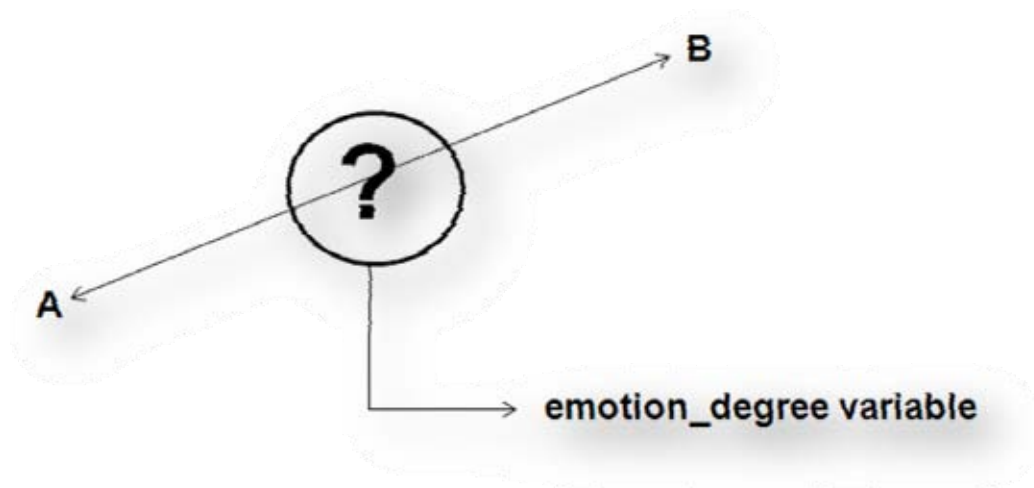


Figura 20.3. Interazione tra due agenti considerando una variabile di valore delle emozioni.

Ogni agente deve decidere se diffondere l'informazione ai vicini. Questa scelta dipende dalla variabile *emotion_degree*.

Partendo da una situazione omogenea di "stati emotivi", è possibile definire delle funzioni per la relazione tra emozioni e il rapporto con la diffusione delle informazioni che gli agenti contengono.

Si vengono a creare diverse situazioni di incroci tra agenti, come spiego in Figura 20.4 (valori relativizzati tra -100 e 100).

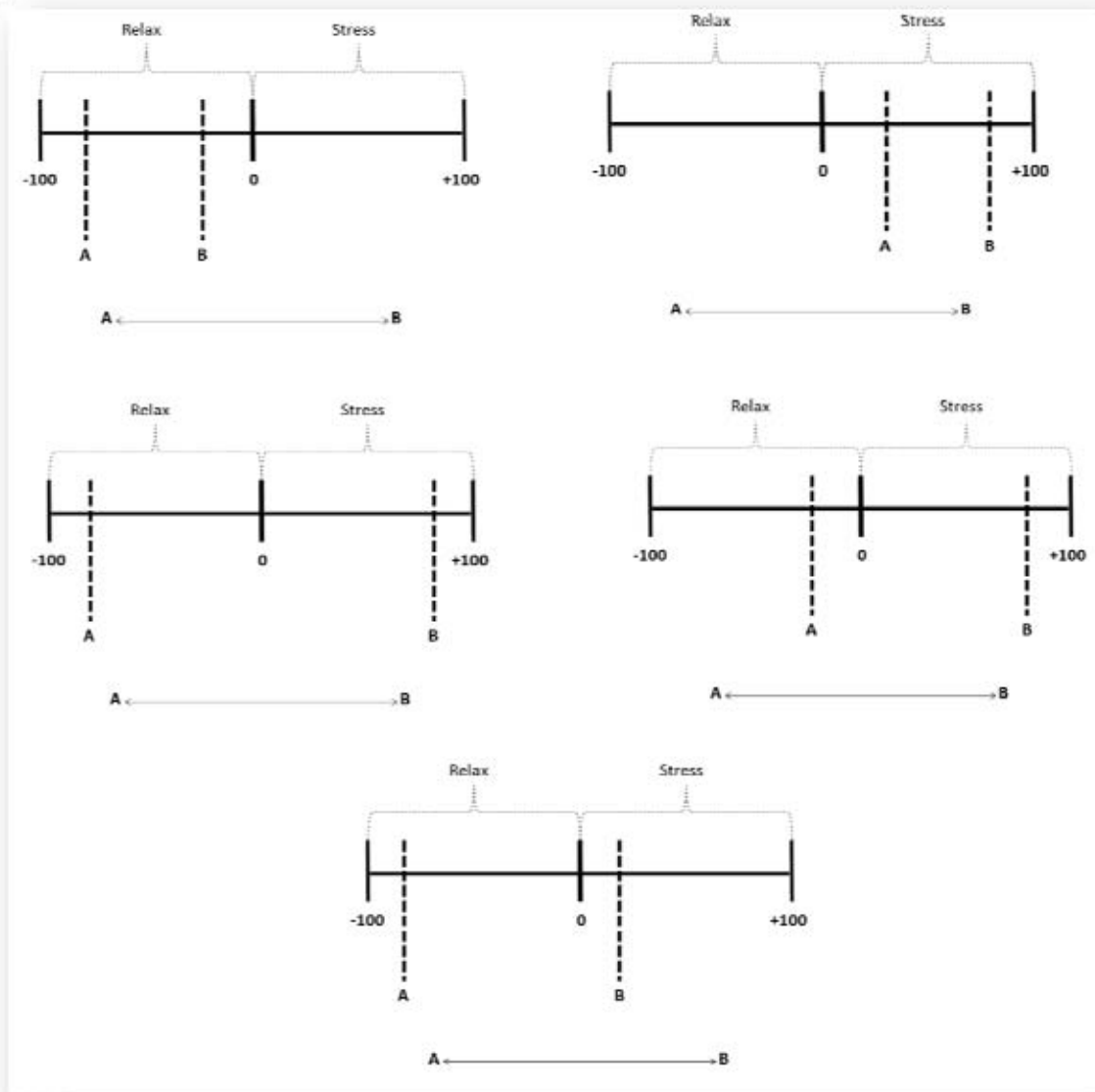


Figura 20.4. Configurazioni possibili nell'interazione tra agenti.

Dunque da un punto di vista "algoritmico/meccanico", definiamo la diffusione tra agenti come segue:

entrambi sono rilassati

if agent.emotion_degree < 0 and neighbor.emotion_degree < 0:

*neighbor.emotion_degree = (neighbor.emotion_degree +
agent.emotion_degree) / 2*

il successivo è in stress

if agent.emotion_degree < 0 and neighbor.emotion_degree >= 0:

*neighbor.emotion_degree = neighbor.emotion_degree +
agent.emotion_degree*

entrambi sono stressati

if agent.emotion_degree >= 0 and neighbor.emotion_degree >= 0:

*neighbor.emotion_degree = (neighbor.emotion_degree +
agent.emotion_degree) / 2*

il successivo è in relax

if agent.emotion_degree >= 0 and neighbor.emotion_degree < 0:

*neighbor.emotion_degree = neighbor.emotion_degree +
agent.emotion_degree*

Diffusione dell'informazione tra agenti

La relazione tra agenti va ovviamente definita "psicologicamente" e in tal senso questo modello garantisce una elevata flessibilità per essere adattato a situazioni (principalmente panic-driven) definite da attenti osservatori della realtà che si intende simulare. A titolo di esempio si consideri lo schema in Figura 20.5.

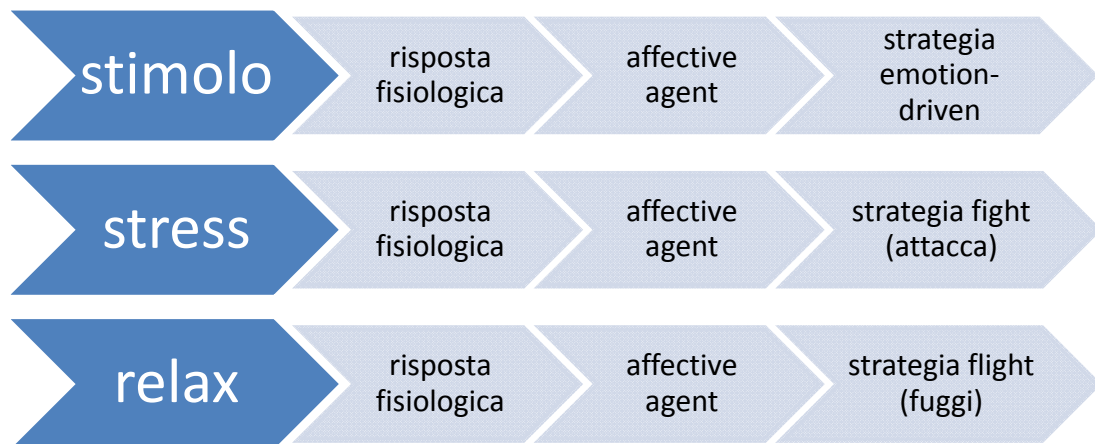


Figura 20.5. Schema stimolo > risposta > agente > strategia.

Schema ERA e affective agents

Volendo definire uno schema entro cui delineare il nostro modello agent-based, ho modificato lo schema ERA (Environment, Rules, Agent) (Gilbert, Terna, 2001), trasposizione pratico/informatica del più ampio modello Soggetti-Regole-Ambiente.

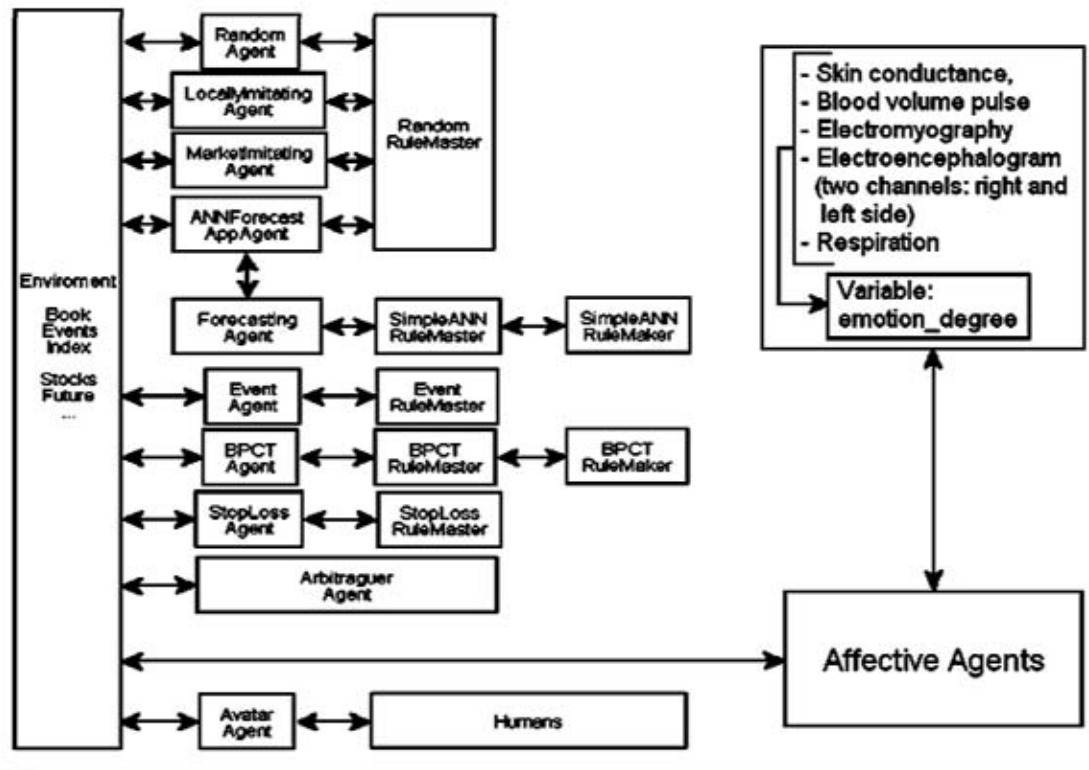


Figura 20.6. Adattamento dello schema ERA (Environment, Rules, Agent).

Il modello computazionale da me delineato si basa su reti Barabási-Albert e analizza la diffusione delle informazioni sotto influenze emotional-based.

Per cominciare inizializzo la variabile chiamata "emotion_degree" per ogni agente. Questa variabile definisce due tipi di agente nel seguente modo: per ogni agente, con assegnamento casuale, stabiliamo il grado dell'emozione individuale tra -100% e +100%. Possiamo considerarlo come un imprinting.

Nella simulazione ho implementato reti Barabàsi-Albert a 1, 2 e 3 out-degree (grado di connessioni in uscita). Un esempio è riportato in Figura 20.7.

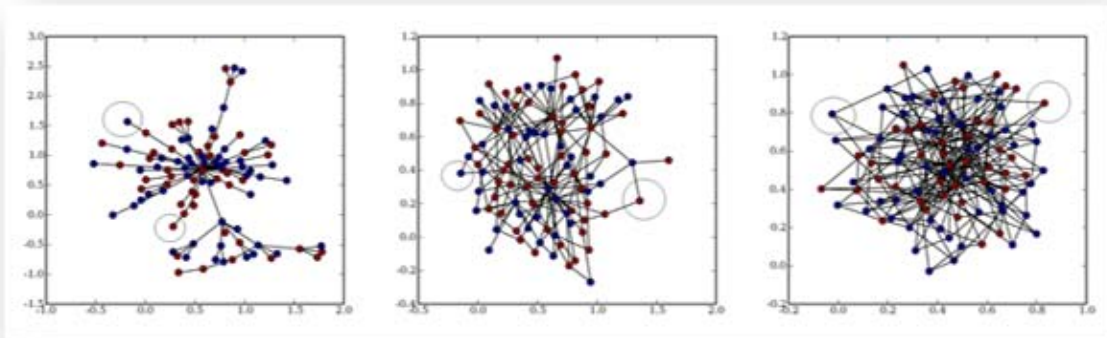


Figura 20.7. Reti Barabási-Albert a 1, 2 e 3 out-degree.

In Figura 20.8 possiamo vedere il grado di distribuzione della rete.

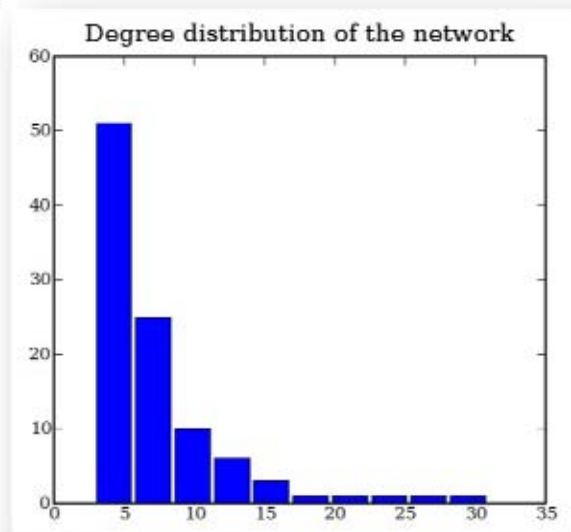


Figure 20.8. Grado di distribuzione della rete.

Il modello considera una situazione in cui la Banca Centrale comunica (da l'informazione) a qualche agente (nella simulazione questi agenti potrebbero essere le banche, ma non necessariamente).

Al tempo $t=0$ (Figura 20.9), inizializziamo gli agenti con la procedura dell'imprinting, vista in precedenza.

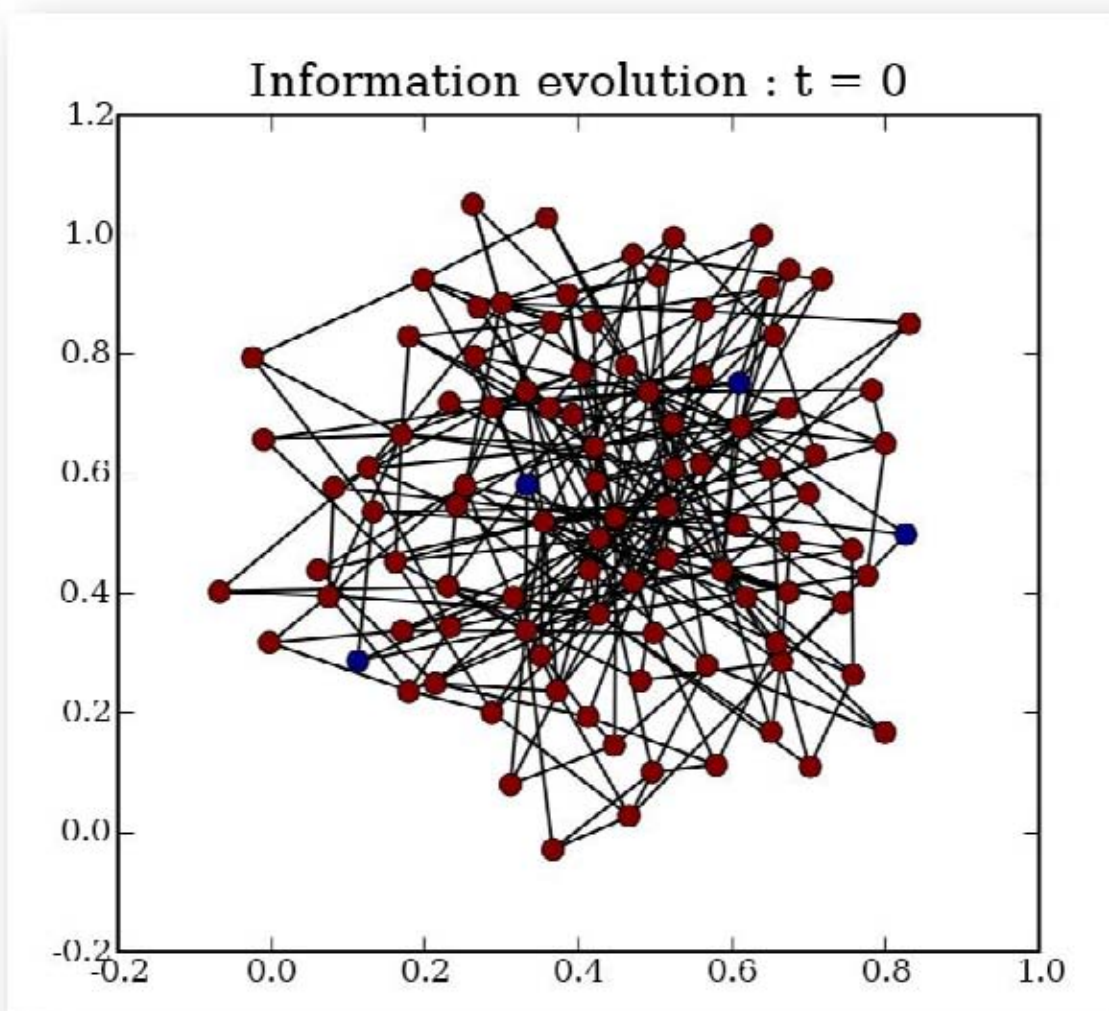


Figura 20.9. Situazione iniziale, con pochi agenti che detengono l'informazione (rete dell'informazione).

Usando la procedura prima descritta, sulla rete delle emozioni, una metà degli agenti avrà una fight strategy e l'altra metà una flight strategy (Figura 20.10).

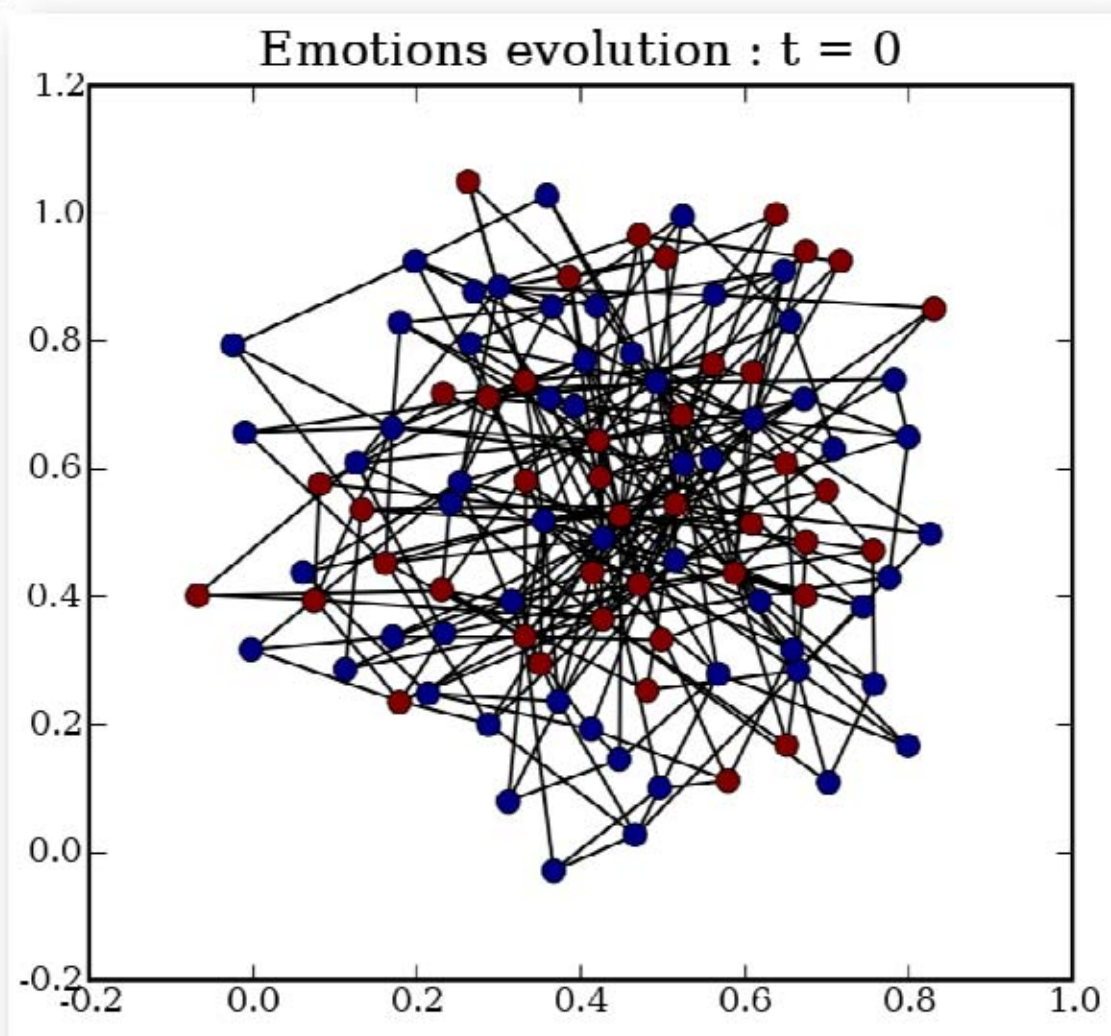


Figura 20.10. Situazione iniziale sulla rete delle emozioni. Il colore di ciascun nodo denota l'appartenenza a ciascun gruppo (avente fight strategy o flight strategy). A ogni nodo è anche assegnata una intensità.

Se gli agenti ottengono l'informazione e hanno una fight strategy, allora acquisiranno una maggiore propensione a diffondere l'informazione stessa.

Chiaramente questo protocollo va definito con criteri psicologici, pur sempre tenendo in mente che si tratta di un modello, che ci aiuta a descrivere la realtà che si sta esaminando, cercando emergenze sistemiche.

Il punto di forza del modello è infatti che è possibile esaminare sia la situazione delle emozioni, sia la situazione della diffusione dell'informazione, onde poter studiare le dinamiche di reciproca modificazione e trarre delle conclusioni su "scenari possibili" (Figura 20.11).

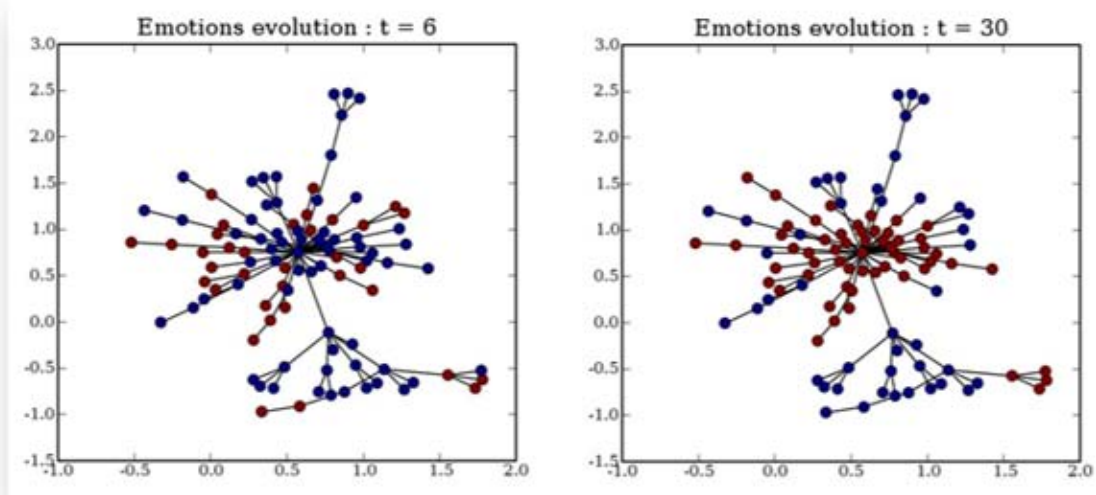


Figura 20.11. Evoluzione della rete delle emozioni.

Un'ultima considerazione circa le emergenze sistemiche di questo primo modello agent-based.

Come è possibile notare in Figura 20.12, nel lungo periodo, ovvero quando quasi tutti gli agenti detengono l'informazione, la migliore strategia degli agenti cambia (autonomamente, come emergenza sistemica appunto). Il sistema torna in equilibrio. Quindi il sistema ha "sviluppato" un feedback negativo per tornare in omeostasi.

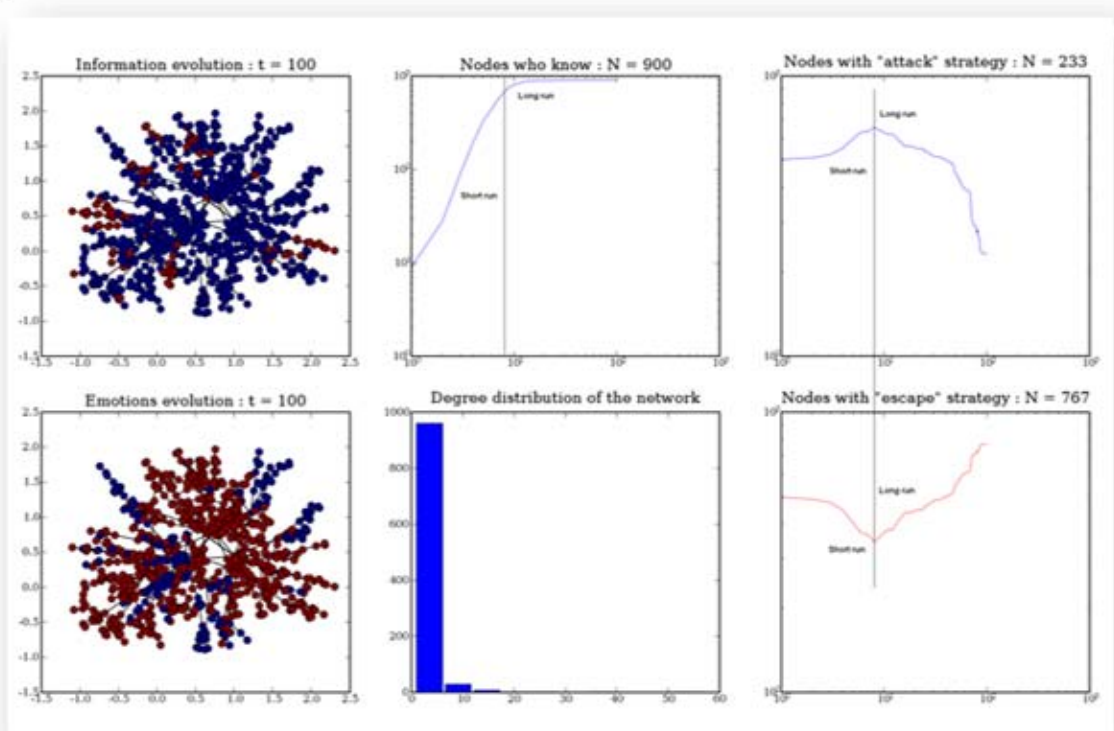


Figura 20.12. Una sintesi di una tra le tante (circa 2000) simulazioni eseguite. E' possibile vedere la strategia di lungo periodo (long run) e quella di breve periodo (short run).

Comunque, gli scenari potrebbero essere talmente rapidi da non consentire un feedback negativo. In questo caso si instaura un circolo vizioso che aumenta continuamente le variazioni del sistema portandolo al collasso, dopo il quale potrebbe anche "ricostruire" l'equilibrio. Un esempio è fornito in Figura 20.13.

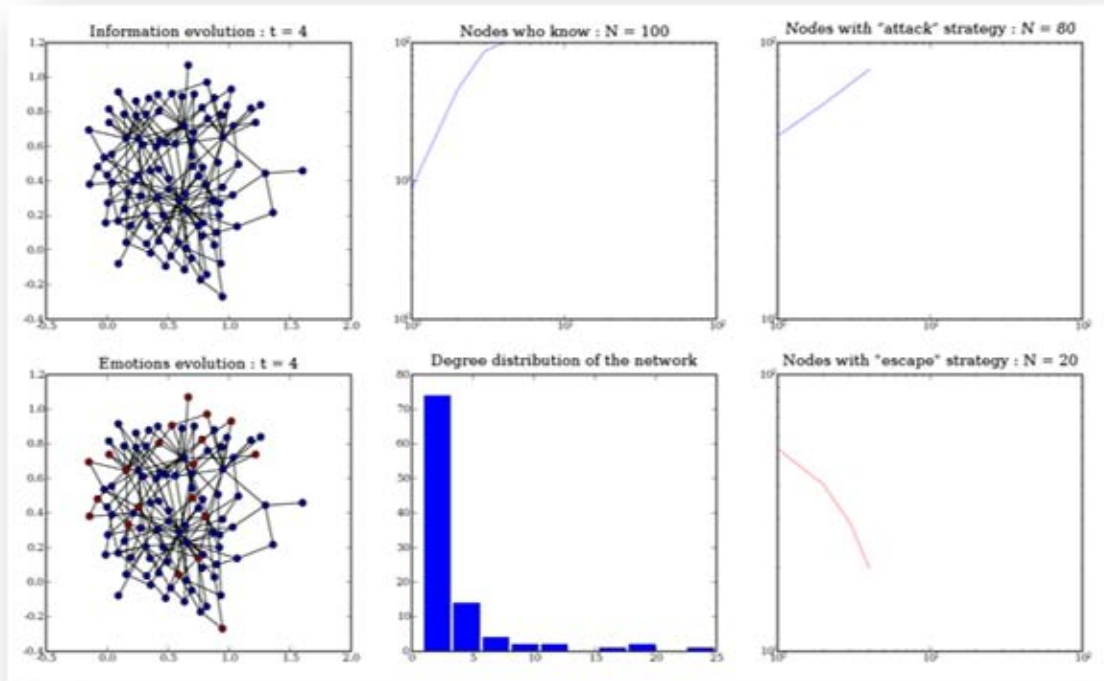


Figura 20.13. Rapida diffusione dell'informazione e collasso del sistema.

Ovviamente un sistema con un out-degree più elevato tende a far collassare il sistema in quanto accelera le dinamiche di diffusione (Figura 20.14).

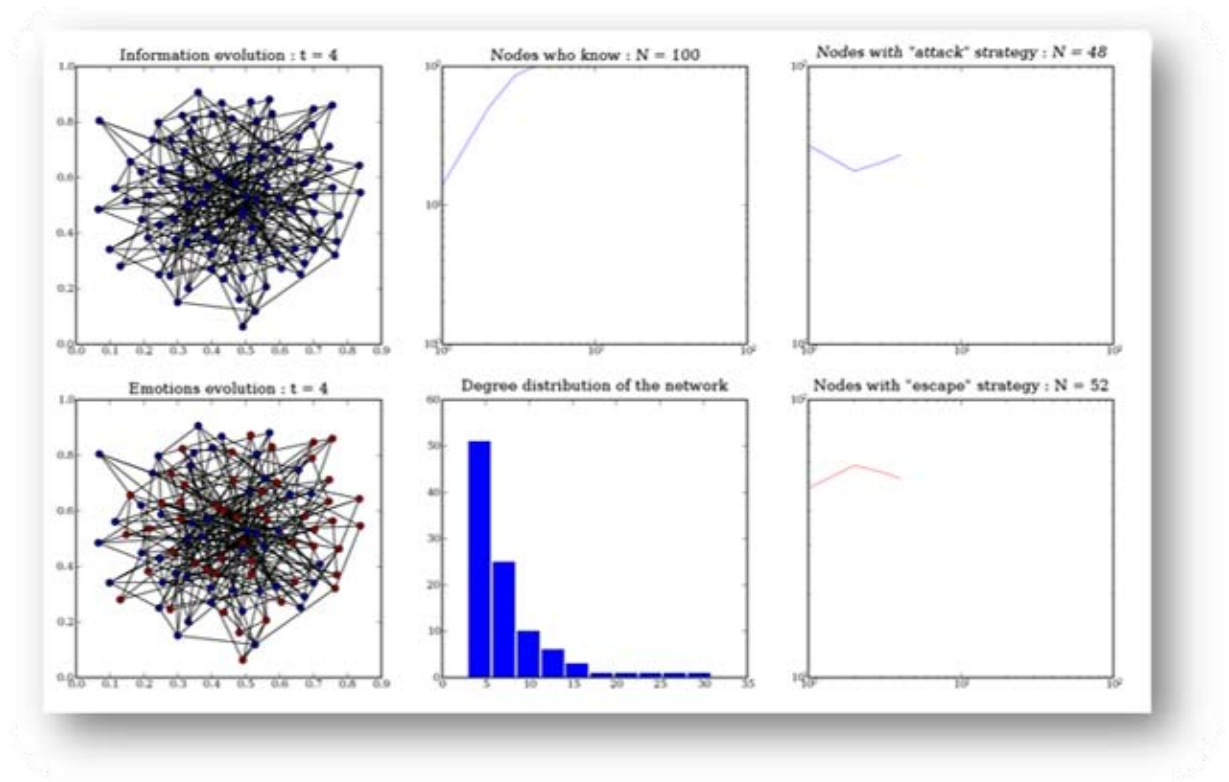


Figura 20.14. Rapida diffusione delle emozioni in una rete con più elevato out-degree.

Di seguito riporto il codice Python che ho creato per la diffusione delle emozioni in assenza di dinamiche di feedback.

```
while number_that_know < population_size:
    plot_and_save()
    time += 1
    for agent in list_of_agents_that_know:
        for j in agent.neighbors :
            neighbor = list_of_agents[j]
            if neighbor not in list_of_agents_that_know:
                ##### Without feedback
```

```

##### both are diffuser
    if agent.emotion_degree >= 0 and neighbor.emotion_degree
>= 0:
        neighbor.emotion_degree = (neighbor.emotion_degree +
agent.emotion_degree) / 2
        neighbor.state = know
        state_for_color[j] = know
        if neighbor not in list_of_agents_that_know_next:
            list_of_agents_that_know_next.append(neighbor)
            number_that_know += 1

##### the next one is not a diffuser
    if agent.emotion_degree >= 0 and neighbor.emotion_degree
< 0:
        neighbor.emotion_degree = neighbor.emotion_degree +
agent.emotion_degree
        if neighbor.emotion_degree < 0:
            neighbor.state = know
            state_for_color[j] = know
            state_for_color_bis[j] = unhappy
            if neighbor not in list_of_agents_that_know_next:
                list_of_agents_that_know_next.append(neighbor)
                number_that_know += 1
        else :
            neighbor.state = know
            state_for_color[j] = know
            state_for_color_bis[j] = happy
            if neighbor not in list_of_agents_that_know_next:
                list_of_agents_that_know_next.append(neighbor)
                number_that_know += 1

##### both are not diffuser
    if agent.emotion_degree < 0 and neighbor.emotion_degree <
0:
        neighbor.emotion_degree = (neighbor.emotion_degree +
agent.emotion_degree) / 2

##### the next one is a diffuser
    if agent.emotion_degree < 0 and neighbor.emotion_degree
>= 0:
        neighbor.emotion_degree = neighbor.emotion_degree +

```

```

agent.emotion_degree
    if neighbor.emotion_degree < 0:
        state_for_color_bis[j] = unhappy

        #####
    plot_number[time] = number_that_know
    list_of_agents_that_know =
copy.deepcopy(list_of_agents_that_know_next)

plot_and_save()
PL.show()

```

Di seguito riporto anche il codice Python per la diffusione delle emozioni in presenza di dinamiche di feedback.

```

while number_that_know < population_size:
    plot_and_save()
    time += 1
    for agent in list_of_agents_that_know:
        for j in agent.neighbors :
            neighbor = list_of_agents[j]

            ##### both are diffuser
            if agent.emotion_degree >= 0 and neighbor.emotion_degree >=
0:
                neighbor.emotion_degree = (neighbor.emotion_degree +
agent.emotion_degree) / 2
                neighbor.state = know
                state_for_color[j] = know
                if neighbor not in list_of_agents_that_know_next:
                    list_of_agents_that_know_next.append(neighbor)
                    number_that_know += 1

            ##### the next one is not a diffuser
            if agent.emotion_degree >= 0 and neighbor.emotion_degree <
0:
                neighbor.emotion_degree = neighbor.emotion_degree +
agent.emotion_degree

```



```

    if neighbor.emotion_degree < 0:
        neighbor.state = know
        state_for_color[j] = know
        state_for_color_bis[j] = unhappy
        if neighbor not in list_of_agents_that_know_next:
            list_of_agents_that_know_next.append(neighbor)
            number_that_know += 1
    else :
        neighbor.state = know
        state_for_color[j] = know
        state_for_color_bis[j] = happy
        if neighbor not in list_of_agents_that_know_next:
            list_of_agents_that_know_next.append(neighbor)
            number_that_know += 1

    ##### both are not diffuser
    if agent.emotion_degree < 0 and neighbor.emotion_degree < 0:
        neighbor.emotion_degree = (neighbor.emotion_degree +
agent.emotion_degree) / 2

    ##### the next one is a diffuser
    if agent.emotion_degree < 0 and neighbor.emotion_degree >=
0:
        neighbor.emotion_degree = neighbor.emotion_degree +
agent.emotion_degree
        if neighbor.emotion_degree < 0:
            state_for_color_bis[j] = unhappy

    #####
    plot_number[time] = number_that_know
    list_of_agents_that_know =
copy.deepcopy(list_of_agents_that_know_next)

plot_and_save()
PL.show()

```

Il programma qui descritto è una simulazione ad agenti e può essere arricchito a più livelli, sia considerando un ambiente più ricco di regole, sia dotando gli agenti di capacità evolutive, mediante l'uso di semplici algoritmi genetici e algoritmi di intelligenza artificiale, magari considerando fattori esterni, come indici reali di qualsiasi tipo. Tuttavia bisogna sempre ricordare che si tratta di una simulazione in cui lo scopo non è rappresentare in rapporto uno a uno la realtà, bensì cercare le semplici regole che possono fare emergere sistematicamente delle proprietà più profonde da interpretare con profonda cura.

Bibliografia

- Aarts E. (2005), "Ambient intelligence drives open innovation", ACM Interactions, Vol. 12, No. 4, pp. 66-68.
- Adams, J. & Moody,-. (2007). To Tell the Truth? Measuring Concordance in Multiply-Reported Network Data.. Social Networks, 29, 44-58
- Albani G, Cipresso P, Gaggioli A, Serino S, Vigna C, Priano L, Mauro A, Riva G (2012). Virtual help for real surgery: The case of awake surgery. STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS, vol. 173, p. 13-15, ISSN: 0926-9630, doi: 10.3233/978-1-61499-022-2-13
- Albani G, Franzini A, Vigna C, Cipresso P, Mauro A, Riva G (2011). Virtual Simulator of Deep Brain Stimulation in Parkinson's Disease. JOURNAL OF CYBERTHERAPY AND REHABILITATION, vol. 4, p. 281-282, ISSN: 1784-9934
- Albani G, Priano L, Cipresso P, Pignatti R, Riva G, Mauro A (2012). Sleep Deconstruction in Early Executive Dysfunctions: A Virtual and Neuropsychological Study in Parkinson's Disease. PARKINSONISM & RELATED DISORDERS, vol. 18, p. S23-S24, ISSN: 1353-8020, doi: 10.1016/S1353-8020(11)70170-5

- Albani G, Priano L, Cipresso P, Raspelli S, Pignatti R, Ferronato P, Liuzzi A, Riva G, Mauro A (2012). Diurnal sleepiness and executive dysfunctions: A virtual and neuropsychological study in Parkinson's disease and sleep apnea syndrome. *MOVEMENT DISORDERS*, vol. 27, p. S9, ISSN: 0885-3185
- Arnold M. (1960), *Emotion and Personality*, Columbia University Press, New York.
- Ashby F. G., Isen A. M. e Turken U. (1999), "A neuropsychological theory of positive affect and its influence on cognition", *Psychological Review*, 106, No 3, pp. 529-550.
- Atkin D. (2004), *The Culting of Brands: When Customers Become True Believers*, Portfolio Publisher, New York.
- Atlan H. (2007), "Complessità, disordine e autocreazione del significato", in Bocchi G., Ceruti M., *La Sfida della Complessità*, Mondadori, Milano.
- Baars B.J. (2002), "The conscious access hypothesis: origins and recent evidence", *Trends in Cognitive Sciences*, 6, 1, pp. 47-52.
- Balgera A, Cipresso P, Bracco F, Villamira M (2009). Entropia. In: Villamira M.; Bracco F.; *Comunicare. Elementi di psicologia della comunicazione*. Milano: FrancoAngeli, ISBN/ISSN: 978-88-568-0394-5.
- Balgera A, Cipresso P, Bracco F, Villamira M (2009). Sistemi. In: Villamira M.; Bracco F.; *Comunicare. Elementi di psicologia della comunicazione*. Milano: FrancoAngeli, ISBN/ISSN: 978-88-568-0394-5.

- Barnett, G. (1988). Precise procedures for longitudinal network analysis.. Barnett, G. & Woelfel, J., (Eds.). Readings in the Galileo System: Theory, methods and applications. (ed.). (pp. 333-368) Dubuque, IA: Kendall/Hunt
- Baron J. (1994), Thinking and Deciding, Cambridge University Press, Cambridge, Massachusetts.
- Bar-On R. e Parker J. D. A. (2000), The handbook of emotional intelligence, Jossey–Bass, San Francisco.
- Barro, R.J. (1977), Unanticipated Money Growth and Unemployment in the United States, in American Economic Review 67, 101–115
- Bateson G. (1976), Verso un'ecologia della mente. Adelphi, Milano.
- Bateson G. (1984), Mente e Natura, un'unità necessaria, Adelphi, Milano.
- Bertalanffy von L. (1983), Teoria generale dei sistemi, Mondadori Milano.
- Bexton W.H., Heron W. e Scott T.H. (1954), "Effects of decreased variation in the sensory environment", Canadian Journal of Psychology, 8, pp. 70-76.
- Birke, D. & Swann,P. Social Networks and Choice of Mobile Phone Operator. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=944354..
- Birke, D. (2008). The economics of networks - A survey of the empirical literature. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1264662..

- Bocchi G., Ceruti M. (2007), *La Sfida della Complessità*, Mondadori, Milano.
- Bonaiuto P., Umiltà C. e Canestrari R. (1965), "Capacità di riconoscimento di figure mascherate dopo privazione sensorimotoria. " *Bollettino della Società Italiana di Biologia Sperimentale*, 41, pp. 523-526.
- Boyatzis R. E., Goleman D. e Rhee K. S. (2000), "Clustering competence in emotional intelligence", in Bar-On R. e Parker J. D. A., *The handbook of emotional intelligence*, Jossey-Bass, San Francisco.
- Bracco F, Cipresso P, Villamira M (2009). *L'errore. Umano, troppo umano*. In: Villamira M.; Bracco F.; *Comunicare. Elementi di psicologia della comunicazione*. Milano: FrancoAngeli, ISBN/ISSN: 978-88-568-0394-5.
- Bracco F, Cipresso P, Villamira M (2009). *Sistemi emotivi*. In: Villamira M.; Bracco F.; *Comunicare. Elementi di psicologia della comunicazione*. Milano: FrancoAngeli, ISBN/ISSN: 978-88-568-0394-5.
- Bracco F, Cipresso P, Villamira M (2009). *Sistemi resilienti*. In: Villamira M.; Bracco F.; *Comunicare. Elementi di psicologia della comunicazione*. Milano: FrancoAngeli, ISBN/ISSN: 978-88-568-0394-5.
- Bracco F, Cipresso P, Villamira M (2009). *Un vecchio alfabeto per nuovi linguaggi*. In: Villamira M.; Bracco F.; *Comunicare. Elementi di psicologia della comunicazione*. Milano: FrancoAngeli, ISBN/ISSN: 978-88-568-0394-5.

- Bracco F. (2005), "Approccio sistemico alla sicurezza nelle organizzazioni complesse: i sistemi ad alta affidabilità", Ticonzero, 61.
- Bracco F. e Spinelli G., (2005), "Elaborazione implicita dell'informazione: dalla "percezione subliminale" alla cognizione inconscia", *Giornale Italiano di Psicologia*, XXXII, 2, pp. 247-286.
- Bracco F., Gianatti R., e Pisano L. (2008), "Cognitive resilience in Emergency Rooms operations: a theoretical framework", *Third Symposium on Resilience Engineering*, Antibes/Luans-les-Pins.
- Braitenberg V. (2008), *I veicoli pensanti. Saggio di psicologia sintetica*, Mimesis, Milano.
- Braman S. (1989), "Defining Information. An approach for policy-makers", *Telecommunications Policy*, 13 (3), pp. 233-242.
- Broadbent D. E., Cooper P. F., FitzGerald P. e Parkes K.R. (1982), "The cognitive failures questionnaire (CFQ) and its correlates", *British Journal of Clinical Psychology*, 21, pp. 1-16.
- Brockman J. (1995), *La terza cultura*, Garzanti, Milano.
- Brown R. e Kulik J. (1977), "Flashbulb memories", *Cognition*, 5, pp. 73-99
- Buck R., (1999), "The biological affect: a typology", *Psychological Review*, vol. 106, 2, pp. 301-336.

- Buckland M. (1991), "Information as thing", *Journal of the American Society of Information Science*, Vol. 42 (5), pp. 351-360.
- Burghouts G.J., R. op den Akker, Heylen D., Poel M. and Nijholt A. (2003), *An Action Selection Architecture for an Emotional Agent*
- Cagan, P (1956), *The monetary Dynamics of Hyperinflation*, in *Studies in the Quantity Theory of Money*, ed. Milton Friedman, Chicago, University of Chicago Press
- Caillois R. (2000), *I giochi e gli uomini. La maschera e la vertigine*, Bompiani, Milano.
- Callaos N. e Callaos B. (2002), "Toward a systemic notion of information: practical consequences", *Informing Science. The International Journal of an Emerging Transdiscipline*, 5(1), pp. 1-11.
- Camurri A., Langerlöf I. e Volpe G. (2003), "Recognizing emotion from dance movement: comparison of spectator recognition and automated techniques", *International Journal of Human-Computer Studies*, 59 pp. 213–225.
- Cannon W. B. (1931), *Again the James-Lange and the thalamic theories of emotions*, *Psychological Review*, Vol. 38
- Capra F. (1997), *La rete della vita*, Rizzoli, Milano.
- Capucci P.L. (1993), *Realtà del virtuale*, Clueb, Bologna
- Carbonaro N, Tognetti A, Anania G, De Rossi D, Cipresso P, Gaggioli A, Riva G (2013). *A mobile biosensor to detect cardiorespiratory activity for stress tracking*. In: *Proceedings*

- of the 2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, PervasiveHealth 2013. p. 440-445, ISBN: 978-193696880-0, 5-8 Maggio 2013, doi: 10.4108/icst.pervasivehealth.2013.252357
- Carroll, C.D. (2003), Macroeconomic Expectations of Households and Professional Forecasters, *Quarterly Journal of Economics*
- Castellano G., Bresin R., Camurri A. e Volpe G. (2007), "Expressive control of music and visual media by full-body movement", in *Proc. of the 2007 Conference on New Interfaces for Musical Expression (NIME07)*, New York, NY.
- Cavelzani A, Villamira M, Cipresso P, Balgera A (2008). Colori ed emozioni. In: Villamira M.A.; Cavelzani A.; *Ambienti dell'ospitalità*. Firenze: Alinea Editrice, ISBN/ISSN: 978-88-6055-329-4.
- Ceruti M. (1986), *Il vincolo e le possibilità*, Feltrinelli, Milano.
- Cherniss C. (2004), "Intelligence, emotional", in C.D. Spielberger, *Encyclopedia of Applied Psychology*, vol. 2, Elsevier Academic Press, New York.
- Chmiel N. (1998), *Uomo, tecnologia, organizzazione*, il Mulino, Bologna.
- Churcland P. (1992), *La struttura della Mente e la Natura della Scienza*, il Mulino, Bologna.
- Ciarrochi J., Mayer J.D. (2007), *Applying Emotional Intelligence*, Psychology Press, New York

- Cipresso P (2010). Modeling Emotions At the Edge of Chaos. From psychophysiology to networked emotions. p. 1-204, VDM Verlag Dr. Müller Editor, ISBN: 978-3-639-30169-4
- Cipresso P (2010). Verso una teoria computazionale della comunicazione. *GIORNALE DI PSICOLOGIA*, vol. 4:1, p. 132-138, ISSN: 1971-9450
- Cipresso P (2011). Synchronizing Physiological Signals Acquired from Biofeedback Equipment and Eye-Tracker Systems. *APPLIED PSYCHOPHYSIOLOGY AND BIOFEEDBACK*, vol. 36, p. 58, ISSN: 1090-0586
- Cipresso P (2012). Do affects affect you or do you affect affects? A closed-loop in positive technology. *JOURNAL OF CYBERTHERAPY AND REHABILITATION*, vol. 5, p. 17-20, ISSN: 1784-9934
- Cipresso P (2012). Welcome to the Contactless Era. *CONTACTLESS BIO-BEHAVIORAL RESEARCH METHODS*, ISSN: 2281-5929
- Cipresso P, Balgera A, Bracco F, Villamira M (2009). Da un Nobel all'altro. In: Villamira M.; Bracco F.; *Comunicare. Elementi di psicologia della comunicazione*. Milano: FrancoAngeli, ISBN/ISSN: 978-88-568-0394-5.
- Cipresso P, Balgera A, Villamira M (2008). Agent-based Computational Economics and emotions for decision-making processes. In: "Economics and psychology: methods and synergies". Selected Proceedings of the IAREP/SABE 2008 World Conference at LUISS in Rome; International

- Association for Research in Economic Psychology and the Society for Advancement of Behavioral Economics. Roma, September 3 - 6, 2008, Roma: Luiss University Press, ISBN/ISSN: 9788861050273.
- Cipresso P, Carelli L, Solca F, Meazzi D, Meriggi P, Poletti B, Lulé D, Ludolph A C, Silani V, Riva G (2012). The use of P300 based BCIs in Amyotrophic Lateral Sclerosis: From Augmentative and Alternative Communication to Cognitive Assessment. *BRAIN AND BEHAVIOR*, vol. 2, p. 479-498, ISSN: 2162-3279, doi: 10.1002/brb3.57
- Cipresso P, Dembele J M, Mauri M, Balgera A, Romanò E, Villamira M (2009) Physiologically driven affective agents in simulated networks in Emergence and Networks, a satellite meeting of the European Conference on Complex Systems (ECCS 2009).
- Cipresso P, Dembele J M, Villamira M (2009). An emotional perspective for Agent-based Computational Economics. In: Vallverdu' J.; Casacuberta D. (Eds.); *Handbook of Research on Synthetic Emotions and Sociable Robotics: New Applications in Affective Computing and Artificial Intelligence*. Hershey, PA 17033 – USA: IGI Global, ISBN: 978-1-60566-354-8.
- Cipresso P, Foresti R, Balgera A, Villamira M (2010). Emozioni e e-learning nella prospettiva dei sistemi complessi. In: *Atti Moodlemoot*.
- Cipresso P, Foresti R, Balgera A, Villamira M (in press). Emozioni e e-learning nella prospettiva dei sistemi complessi. In:

MoodleMoot. Reggio Emilia, Università di Modena e Reggio Emilia

Cipresso P, Gaggioli A, Grassi A, Pallavicini F, Raspelli S, Vigna C, Riva G (2011). Neurophysiological Aspects of Stress-related Disorders. *APPLIED PSYCHOPHYSIOLOGY AND BIOFEEDBACK*, vol. 36, p. 225-226, ISSN: 1090-0586

Cipresso P, Gaggioli A, Serino S, Cipresso S, Riva G (2012). How to create memorable and strong passwords. *JMIR. JOURNAL OF MEDICAL INTERNET RESEARCH*, vol. 14, ISSN: 1438-8871, doi: 10.2196/jmir.1906

Cipresso P, Gaggioli A, Serino S, Pallavicini F, Raspelli S, Grassi A, Sellitti L, Riva G (2012). EEG alpha asymmetry in virtual environments for the assessment of stress-related disorders. *STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS*, vol. 173, p. 102-104, ISSN: 0926-9630, doi: 10.3233/978-1-61499-022-2-102

Cipresso P, Gaggioli A, Serino S, Raspelli S, Vigna C, Pallavicini F, Riva G (2012). Inter-reality in the Evaluation and Treatment of Psychological Stress Disorders: the INTERSTRESS Project. *STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS*, vol. 181, p. 8-11, ISSN: 0926-9630, doi: 10.3233/978-1-61499-121-2-8

Cipresso P, Gaggioli A, Serino S, Riva G (2012). Stress diffusion through complex networks. *INTERNATIONAL JOURNAL OF ADAPTIVE, RESILIENT AND AUTONOMIC SYSTEMS*, vol. 3, p. 46-64, ISSN: 1947-9220, doi: 10.4018/jaras.2012010103

- Cipresso P, Hoogendoorn M, Klein M, Matic A (2012). Computer Paradigms for Mental Health. p. 1-120, SciTePress – Science and Technology Publications, ISBN: 978-989-8425-92-8
- Cipresso P, Hoogendoorn M, Klein M, Matic A (2012). Foreword. In: (a cura di): Cipresso, P, Hoogendoorn M, Klein M, Matic A, Proceedings of the 2nd International Workshop on Computing Paradigms for Mental Health, MindCare 2012, in Conjunction with BIOSTEC 2012. p. iii-iv
- Cipresso P, La Paglia F, La Cascia C, Riva G, Albani G, La Barbera D (2013). Break in volition: A virtual reality study in patients with obsessive-compulsive disorder. EXPERIMENTAL BRAIN RESEARCH, vol. 229, p. 443-449, ISSN: 0014-4819, doi: 10.1007/s00221-013-3471-y
- Cipresso P, Mauri M, Balgera A, Roman E, Villamira M (2010). Synchronization of a Biofeedback System with an Eye Tracker through an Audiovisual Stimulus Marker. APPLIED PSYCHOPHYSIOLOGY AND BIOFEEDBACK, vol. 35, p. 332, ISSN: 1090-0586
- Cipresso P, Meriggi P, Carelli L, Solca F, Meazzi D, Poletti B, Lulé D, Ludolph A C, Riva G, Silani V (2011). The combined use of Brain Computer Interface and Eye-Tracking technology for cognitive assessment in Amyotrophic Lateral Sclerosis. In: 2011 5th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, PervasiveHealth 2011. p. 320-324, ISBN: 978-193696815-2, Dublin, Ireland, May 2011, doi: 10.4108/icst.pervasivehealth.2011.246018

- Cipresso P, Meriggi P, Carelli L, Solca F, Poletti B, Lulé D, Ludolph A C, Silani V, Riva G (2012). Brain Computer Interface and Eye-Tracking for Neuropsychological Assessment of Executive Functions: A Pilot Study. In: Proceedings of the 2nd International Workshop on Computing Paradigms for Mental Health. p. 79-88, Lisbona:SciTePress, Villamoura, Algarve, Portugal, 1 Febbraio 2012, doi: 10.5220/0003893100790088
- Cipresso P, Meriggi P, Pedroli E, Solca F, Carelli L, Poletti B, Lulé D, Ludolph A C, Silani V, Riva G (2012). eBrain Engine (Open Source software)
- Cipresso P, Raspelli S, Pallavicini F, Grassi A, Balgera A, Gaggioli A, Villamira MA, Albani G, Mauro A, Riva G (2011). Investigations of executive functions using Virtual Multiple Errands Test and Psychophysiological measures. JOURNAL OF CYBERTHERAPY AND REHABILITATION, vol. 4, p. 259-260, ISSN: 1784-9934
- Cipresso P, Sellitti L, Assawy N, Galli F, Balgera A, Dembele J, Villamira MA, Riva G (2012). Real Emotions for Simulated Social Networks. In: (a cura di): Abraham Ajith;Hassanien Aboul-Ella, Computational Social Networks: Tools, Perspectives and Applications. p. 391-415, ISBN: 978-1-4471-4048-1, doi: 10.1007/978-1-4471-4048-1_16
- Cipresso P, Serino S, Gaggioli A, Albani G, Riva G (2013). Contactless Bio-behavioral Technologies for Virtual Reality. STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS,

- vol. 191, p. 149-153, ISSN: 0926-9630, doi: 10.3233/978-1-61499-282-0-149
- Cipresso P, Serino S, Gaggioli A, Riva G (2012). Modeling the social networking experience objectifying the subjective. *STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS*, vol. 181, p. 12-16, ISSN: 0926-9630, doi: 10.3233/978-1-61499-121-2-12
- Cipresso P, Serino S, Pedroli E, Albani G, Riva G (2013). Psychometric reliability of the NeuroVR-based virtual version of the Multiple Errands Test. In: *Proceedings of the 2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, PervasiveHealth 2013*. p. 446-449, ISBN: 978-193696880-0, 5-8 Maggio 2013, doi: 10.4108/icst.pervasivehealth.2013.252361
- Cipresso P, Serino S, Riva G (2014). The pursuit of happiness measurement: A psychometric model based on psychophysiological correlates. *THE SCIENTIFIC WORLD JOURNAL*, Article ID 139128, 15 pages, ISSN: 1537-744X, doi: 10.1155/2014/139128
- Cipresso P, Serino S, Villani D, Repetto C, Sellitti L, Albani G, Mauro A, Gaggioli A, Riva G (2012). Is your phone so smart to affect your state? An exploratory study based on psychophysiological measures. *NEUROCOMPUTING*, vol. 84, p. 23-30, ISSN: 0925-2312, doi: 10.1016/j.neucom.2011.12.027

- Cipresso P, Villamira M (2007). Aspettative razionali e microfondazione per economie in situazione di iperinflazione. In: Workshop italiano di Vita Artificiale e Computazione Evolutiva. Sampieri (RG), Italia, 5 - 7 Settembre, ISBN/ISSN: 1973-0543.
- Cipresso P, Villamira M (2007). Shaping the "post-carbon" society: changes at systemic level in transport, housing and consumer behaviour, an Agent-based Computational Economics approach. In: Values and Economy. Ljubljana, Slovenia, 9-12 Settembre, ISBN/ISSN: 978-961-237-206-4.
- Cipresso P, Villamira M (2008). Ambienti virtuali. In: Villamira M.; Cavelzani A.; Ambienti dell'ospitalità. Firenze: Alinea Editrice, ISBN/ISSN: 978-88-6055-329-4.
- Cipresso P, Villamira M (2009). Casi chiusi, casi aperti. In: Villamira M.; Bracco F.; Comunicare. Elementi di psicologia della comunicazione. Milano: FrancoAngeli, ISBN/ISSN: 978-88-568-0394-5.
- Cipresso P, Villamira M (2009). Scenari futuri. In: Villamira M.; Bracco F.; Comunicare. Elementi di psicologia della comunicazione. Milano: FrancoAngeli, ISBN/ISSN: 978-88-568-0394-5.
- Cipresso P, Villamira M (2010). Il mezzo e il messaggio: Primi risultati di un'indagine pilota sulla fruizione di notizie tramite giornali, web e tv. *PROBLEMI DELL'INFORMAZIONE*, vol. 35:4, p. 441-446, ISSN: 0390-5195, doi: 10.1445/33849

- Cipresso P, Villamira M (2010). Physiological correlates for an agent based computational model. *APPLIED PSYCHOPHYSIOLOGY AND BIOFEEDBACK*, vol. 35, p. 331-332, ISSN: 1090-0586
- Cipresso P, Villamira M (2010), Physiological correlates for an agent-based computational model [Abstract], *International Journal of Applied Psychophysiology and Biofeedback*, Springer Netherlands, 1090-0586 (Print) 1573-3270 (Online).
- Cipresso P., Mauri M., Balgera A., Romanò E., Villamira M. (2011), Synchronization of a biofeedback system with an eye tracker through an audiovisual stimulus marker [Abstract], *International Journal of Applied Psychophysiology and Biofeedback*, Springer Netherlands, 1090-0586 (Print) 1573-3270 (Online).
- Clifton, A., Pilkonis, P., & Christopher, M. (2007). Social networks in borderline personality disorder. *Journal of Personality Disorders*, 21, 434-441
- Clifton, A., Turkheimer, E. F., & Oltmanns, T. F. (2007). Improving assessment of personality disorder traits through social network analysis. *Journal of Personality*, 75, 1007-1031
- Clifton, A., Turkheimer, E. F., & Oltmanns, T. F. (2009). Personality disorder in social networks: Network position as a marker of interpersonal dysfunction. *Social Networks*, 31, 26-32
- Csikszentmihalyi M. (2008), *Flow. The psychology of optimal experience*, Harper Collins, New York.

- D'Urso V. e Trentin R. (2006), *Introduzione alla psicologia delle emozioni*, Laterza, Roma-Bari.
- Dai D. Y. e Sternberg R. J. (2004), "Beyond Cognitivism: Toward an Integrated Understanding of Intellectual Functioning and Development", in Sternberg R. J. e Dai D. Y., *Motivation, emotion and cognition*, Lawrence Erlbaum Associates, Mahwah, New Jersey.
- Damasio A. R. (1995), *L'errore di Cartesio*, Adelphi, Milano.
- Damasio A. R. (2000), *Emozione e coscienza*, Adelphi, Milano.
- Damasio A. R., Grabowski T. J., Bechara A., Damasio H., Ponto L. L. B., Parvizi J. e Hichwa R. D. (2000), "Subcortical and cortical brain activity during the feeling of self-generated emotions", *Nature Neuroscience*, 3, pp 1049-1056.
- Darwin C. (1872), *The expression of the emotions in man and animals*, London
- De Sousa R. (1987), *The rationality of emotion*, MIT Press, Cambridge, MA.
- De Toni A. e Comello L. (2005), *Prede o ragni? Uomini e organizzazioni nella ragnatela della complessità*, UTET Università, Torino.
- Dekker S. W. A. (2005), *Ten Questions about Human Error: A new view of human factors and system safety*, Lawrence Erlbaum Associates, Hillsdale, New Jersey.
- Dennett D. C. (2000), *La mente e le menti*, Rizzoli, Milano.
- Dervin B. (1981), "Mass communicating: Changing Conceptions of the audience", in Rice R. E. e Paisley W. J., *Public*

- Communication Campaigns, pp.71-78, Sage Publications, Beverly Hills.
- Di Nocera F. (2004), *Cos'è l'ergonomia cognitiva*, Carocci, Milano.
- Dini A. (2007), *Emozione Apple. Fabbricare sogni nel XXI secolo*, Il Sole
- Downer J. D. C. (1961), "Changes in visual gnostic function and emotional behavior following unilateral temporal lobe damage in the split-brain monkey", *Nature*, pp. 50-51.
- Edwards E. (1988), "Introductory Overview", in Wiener E. L. e Nagel D. C., *Human Factors in Aviation* Academic Press, San Diego.
- Ekman P. (2003), *Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life*, Times Book
- Ekman P. e Friesen W. (1977), *Facial Action Coding System*, Consulting Psychologists Press.
- Ekman P.; Levenson R. W.; Friesen W. (1983), Autonomic nervous system activity distinguishes among emotions, *Science*, 221
- Ekman P.; Scherer K. R. (a cura di), (1984), *Approaches to emotions*, Erlbaum, Hillsdale
- Elbirt, B. S. *The Nature of Networks: A Structural Census of Degree Centrality across Multiple Network Sizes and Edge Densities*. (2008). Proceedings from SUNBELT 2008. St. Petersburg Beach, Florida, USA:...

- Elman J.L. (1990), Finding structure in time. *Cognitive Science*, 14
- Fischhoff B. (1975), "Hindsight \neq foresight: The effect of outcome knowledge on judgment under uncertainty" *Journal of Experimental Psychology: Human Perception & Performance*, 1, pp. 288-299.
- Fisher, I. (1930), *The Theory of interest*. The Macmillan Company. ISBN13 978-0879918644
- Floridi L. (1999). *Philosophy and Computing, An Introduction*, Routledge, Taylor and Francis Group, London.
- Foerster Heinz von (1987), *Sistemi che Osservano*, Astrolabio, Roma.
- Frijda N. H. (1986), *Emozioni*, il Mulino, Bologna.
- Frijda N. H. (1993), *Appraisal and Beyond: The Issue of Cognitive Determinants of Emotion*, Taylor & Francis
- Gaggioli A, Cipresso P, Serino S, Pioggia G, Tartarisco G, Baldus G, Corda D, Riva G (2012). An Open Source Mobile Platform for Psychophysiological Self Tracking. *STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS*, vol. 173, p. 136-138, ISSN: 0926-9630, doi: 10.3233/978-1-61499-022-2-136
- Gaggioli A, Cipresso P, Serino S, Riva G (2013). Psychophysiological correlates of flow during daily activities. *STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS*, vol. 191, p. 65-69, ISSN: 0926-9630, doi: 10.3233/978-1-61499-282-0-65

- Gaggioli A, Morganti L, Bonfiglio S, Scaratti C, Cipresso P, Serino S, Riva G (2013). Intergenerational Group Reminiscence: A Potentially Effective Intervention to Enhance Elderly Psychosocial Wellbeing and to Improve Children's Perception of Aging. *EDUCATIONAL GERONTOLOGY*, ISSN: 0360-1277, doi: 10.1080/03601277.2013.844042
- Gaggioli A, Pioggia G, Tartarisco G, Baldus G, Corda D, Cipresso P, Riva G (2011). PsychLog (Open Source: <http://psychlog.sourceforge.net/>).
- Gaggioli A, Pioggia G, Tartarisco G, Baldus G, Corda D, Cipresso P, Riva G (2013). A Mobile Data Collection Platform for Mental Health Research. *PERSONAL AND UBIQUITOUS COMPUTING*, vol. 17, p. 241-251, ISSN: 1617-4909, doi: 10.1007/s00779-011-0465-2
- Gaggioli A, Pioggia G, Tartarisco G, Baldus G, Ferro M, Cipresso P, Serino S, Popleteev A, Gabrielli S, Maimone R, Riva G (2012). A System for Automatic Detection of Momentary Stress in Naturalistic Settings. *STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS*, vol. 181, p. 182-186, ISSN: 0926-9630, doi: 10.3233/978-1-61499-121-2-182
- Gaggioli A, Pioggia G, Tartarisco G, Cipresso P, Riva G (2011). PsychLog: A Personal Data Collection Platform for Psychophysiological Research. *ANNUAL REVIEW OF CYBERTHERAPY AND TELEMEDICINE*, vol. 9, p. 82-84, ISSN: 1554-8716

- Gaggioli A, Raspelli S, Grassi A, Pallavicini F, Cipresso P, Wiederhold B K, Riva G (2011). Ubiquitous Health in Practice: The Interreality Paradigm. *STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS*, vol. 163, p. 185-191, ISSN: 0926-9630, doi: 10.3233/978-1-60750-706-2-185
- Gardner H. (1987), *Formae mentis. Saggio sulla pluralità delle intelligenze*, Feltrinelli, Milano.
- Gell-Mann M. (1996), *Il quark e il giaguaro, avventure nel semplice e nel complesso*, Bollati Boringhieri, Torino.
- Giakoumis D, Drosou A, Cipresso P, Tzovaras D, Hassapis G, Gaggioli A, Riva G (2012). Real-time Monitoring of Behavioural Parameters Related to Psychological Stress. *STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS*, vol. 181, p. 287-291, ISSN: 0926-9630, doi: 10.3233/978-1-61499-121-2-287
- Giakoumis D, Drosou A, Cipresso P, Tzovaras D, Hassapis G, Gaggioli A, Riva G (2012). Using Activity-Related Behavioural Features towards more Effective Automatic Stress Detection. *PLOS ONE*, vol. 7, e43571, ISSN: 1932-6203, doi: 10.1371/journal.pone.0043571
- Gibson J. J. (1999), *Un approccio ecologico alla percezione visiva*, il Mulino, Bologna.
- Gilbert N. (2008) *Agent-Based Models. Quantitative Applications in the Social Sciences*. London: SAGE Publications.
- Gillies J. e Cailliau R. (2002), *Com'è nato il web*, Baldini e Castoldi, Milano.

- Giroto V. (1994), *Il ragionamento*, Il Mulino, Bologna.
- Gleick J. (1996), *Caos. La nascita di una nuova scienza*, Sansoni, Milano.
- Goldstein K. (1995), *The organism: A holistic approach to biology derived from pathological data in man*, Zone Books, New York.
- Goleman D. (1996), *Intelligenza emotiva. Che cos'è, perché può renderci felici*, Rizzoli, Milano.
- Goleman D. (1998), *Lavorare con intelligenza emotiva*, Rizzoli, Milano.
- Gordon T. (1999), *Leader efficaci. Essere una guida responsabile favorendo la partecipazione*, Edizioni la Meridiana, Molfetta, Bari.
- Gordon T. (2005), *Relazioni efficaci. Come costruirle, come non pregiudicarle*, Edizioni la Meridiana, Molfetta, Bari.
- Gould S. J. e Vrba E. (1982), "Exaptation - a missing term in the science of form", *Paleobiology* 8 (1), pp. 4-15.
- Grassi A, Pallavicini F, Raspelli S, Serino S, Spagnoli G, Cipresso P, Villamira MA, Gaggioli A, Riva G (2011). Using Video Clips to Induce Negative Emotional Responses. *JOURNAL OF CYBERTHERAPY AND REHABILITATION*, vol. 4, p. 285-287, ISSN: 1784-9934
- Grimes J. (1996). "On the failure to detect changes in scenes across saccades", in Akins K., *Perception. Vancouver Studies in Cognitive Science*, Vol. 2, pp. 89-110, Oxford University Press, New York.

- Handcock, M. S. & Jones, J. (2004). Likelihood-Based Inference for Stochastic Models of Sexual Network Evolution. *Theoretical Population Biology*, 65, 413-422.
- Handcock, M. S. & Jones, J. (2006). Interval Estimates for Epidemic Thresholds in Two-Sex Network Models. *Theoretical Population Biology*, 70, 125-134.
- Hasson U. et al. (2001), Vase or Face? A Neural Correlate of Shape-Selective Grouping Processes in the Human Brain, *Journal of cognitive neuroscience*, Vol. 13, No. 6, Pages 744-753
- Heims S. J. (1994), *I cibernetici. Un gruppo e un'idea*, Editori Riuniti, Roma, 1994.
- Heinrich H. W. (1931), *Industrial Accident Prevention. A Scientific Approach*, McGraw-Hill, New York.
- Henderson J. M. e Hollingworth, A. (1999), "The role of fixation position in detecting scene changes across saccades, *Psychological Science*, 10(5), pp. 438-443.
- Hoffman D. L., Novak, T. P. e Venkatesh, A. (2004), "Has the Internet become indispensable?" *Communications of the ACM*, 47(7), pp. 37-42.
- Hollnagel E., Leveson N. e Woods J. (2006), *Resilience Engineering-Concepts and Precepts*, Ashgate Press, London.
- Howitt, P. (2006), The Microfoundations of the Keynesian Multiplier Process, *Journal of Economic Interaction and Coordination*, 33-44
- <http://ebrainengine.codeplex.com/>).

- Huizinga J. (1946), *Homo ludens*, Einaudi, Torino.
- Infante C. (2000), *Imparare giocando. L'interattività tra teatro e ipermedia*, Bollati Boringhieri, Torino.
- Isen A M. (2000), "Positive affect and decision making", in Lewis M. e Haviland J., *Handbook of emotions*, Guilford, New York.
- Izard C. E. (1977), *Human emotions*, Plenum Press, New York
- James W. (1884), "What is an emotion?", *Mind*, 9, 188-205.
- James W. (1884), *What is an emotion?*, *Mind*
- Johnson S. (2003), *Emotions and the brain: Fear*, *Discover Magazine*
- Johnson S. (2004), *La nuova scienza dei sistemi emergenti*, Garzanti, Milano.
- Johnson-Laird P. (1988), *Modelli mentali*, Il Mulino, Bologna
- Jones B. e Mishkin M. (1972), "Limbic lesions and the problem of stimulus-reinforcement associations", *Experimental Neurology*, 36; pp. 362- 377.
- Jones, J. H. & Hancock, M. (2003). *Sexual Contacts and Epidemic Thresholds*. *Nature*, 425, 605-606.
- Kagan J. (1984), *The Nature of the Child*, Basic Books, New York.
- Kahneman D. e Tversky A. (2000), *Choices, Values and Frames*, Cambridge University Press and the Russell Sage Foundation, New York.
- Kahneman D. e Tversky A., (1986), "Choices, Values and Frames", AA.VV., *Behavioral and Social Science: 50 Years of Discovery*, The National Academy of Sciences.

- Kahneman D., Slovic P. e Tversky A. (1981), *Judgement Under Uncertainty: Heuristics and Biases*, Cambridge University Press, New York.
- Kahney L. (2004), *The Cult of Mac*, No Starch Press, San Francisco.
- Kahney L. (2005), *The Cult of iPod*, No Starch Press, San Francisco.
- Kauffman S. (1999), "L'ordine gratuito" in J. Brockman (a cura di), *La terza cultura*, Garzanti, Milano.
- Kauffman S. (2001), *A casa nell'Universo. Le leggi del caos e della complessità*, Editori Riuniti, Roma.
- Kleinginna Jr, P R e Kleinginna A M. (1981), "A categorized list of emotion definitions, with suggestions for a consensual definition", *Motivation and Emotion*, 5, pp 345-379.
- La France M. (1982), "Posture Mirroring and Rapport", in David M., *Interaction rhythms: periodicity in communicative behaviour*, Human Sciences Press, New York.
- Labouvie-Vief G. (2000), "Reason and emotion across the life span: A personal view" in Birren J. E. e Schroots J. J. F., *A history of geropsychology in autobiography* (pp. 173-184), American Psychological Association Washington.
- Lane R., Reiman E. M., Ahern G. L., Schwartz G. E. e Davidson R. J. (1997), "Neuroanatomical correlates of happiness, sadness and disgust", *American Journal of Psychiatry*, 154, pp 926-933.

- Lanz, M., Tagliabue, S., Barni, D., Marabelli, C. (2010), Elementi di metodologia della ricerca. Esperienze pratiche in psicologia, Franco Angeli, Milano.
- Lazarus R. (1966), Psychological stress and the coping process, McGraw-Hill, New York
- Lazarus R. e Folkman S. (1984), Stress, appraisal, and coping, Springer Publisher, New York.
- Lazarus R. S. (1968), "Emotion as coping process" in: Arnold M. B., The nature of emotion, Penguin books, London.
- Lazarus R. S. (1991), Emotion & Adaptation, Oxford University Press, New York.
- LeBaron, B. (2006), Agent Based Computational Finance: Suggested Readings and Early Research, Journal of Economic Dynamics and Control
- LeDoux J. (1996), The emotional brain – The mysterious underpinnings of emotional life, Simon & Schuster
- LeDoux J. (1998), Fear and the brain: Where have we been, and where we are going?, «Biological- Psichiatria», 44, 12, pp. 229-1238
- LeDoux J. (1998), Il cervello emotivo (I ed. 1996), Baldini e Castoldi, Milano
- LeDoux J. (2000), Emotion Circuits in the Brain, Annual Review of Neuroscience, Vol. 23: 155-184
- LeDoux J. (2008), Amygdala, Scholarpedia

- Leeper R. W. (1948), "A Motivational Theory of Emotion to Replace 'emotion as disorganized response' ", *Psychological Review*, 55, pp.5-21.
- Leijonhufvud, A. (2006), Agent-Based Macro, in *Handbook of Computational Economics*, edited by Leigh Tesfatsion and Kenneth L. Judd, 1625-37, Vol. 2, Amsterdam: North-Holland
- Levenson R. W. (1992), Autonomic nervous system differences among emotions, *Psychological Science*
- Levenson R. W.; Ascher E.; Goodkind M.; McCarthy M.; Sturm V.; Werner K. (2008), Laboratory testing of emotion and frontal cortex, *Handbook of Clinical Neurology*, Vol. 88, Chapter 25
- Levin D. T. e Simons D. J. (1997), "Failure to detect changes to attended objects in motion pictures" *Psychonomic Bulletin and Review*, 4, pp. 501-506.
- Levin D. T., Momen N., Drivdahl S.B. e Simons D. J. (2000), "Change blindness blindness: The metacognitive error of overestimating change-detection ability", *Visual Cognition*, 7(1/2/3), pp. 397-412.
- Levin, D. T., Simons, D. J., Angelone, B. L., Chabris, C. F. (2002), "Memory for centrally attended changing objects in an incidental real-world change detection paradigm", *British Journal of Psychology*, 93, pp. 289-302.
- Lewis M, Haviland-Jones J.M. & L. Feldman Barrett (Eds.), (2008), *Handbook of Emotions*. Third Edition, Psychology Press, New York

- Liu K. K. e Picard R. W. (2003), 'Subtle Expressivity in a Robotic Computer', Proceedings of the CHI2003 Workshop on Subtle Expressivity for Characters and Robot, April 2003, Ft. Lauderdale, FL.
- Losee R. M. (1997), "A discipline independent definition of information", Journal of the American Society for Information Science, 48(3), pp. 254-269.
- Luccio, R. (2005). Ricerca e analisi dei dati in psicologia: Il Mulino.
- Mack A. e Rock I. (1998), Inattention blindness, MIT Press, Cambridge, MA.
- Magagnin V, Mauri M, Cipresso P, Mainardi L, Brown En, Cerutti S, Villamira M, Barbieri R (2010). Heart Rate Variability and Respiratory Sinus Arrhythmia Assessment of Affective States by Bivariate Autoregressive Spectral Analysis. COMPUTERS IN CARDIOLOGY, vol. 37, p. 145-148, ISSN: 0276-6574
- Maistrello S. (2007), La parte abitata della Rete, Tecniche nuove, Milano.
- Mandelbrot B. B. (1990), La geometria della natura, Theoria, Roma.
- Mankiw, N.G. and Ricardo, R (2001), Sticky Information: A Model of Monetary Nonneutrality and Structural Slumps, NBER Working Paper Number 8614
- Mann, H. (1920). A method of analyzing the electrocardiogram. Archives of Internal Medicine, 25(3), 283.

- Marabelli, C., Piroli, V., Tagliabue, S. (2010), *Psicologia ed excel*, EDUCatt.
- Maragliano R. (2004), *Nuovo manuale di didattica multimediale*, Laterza, Roma-Bari.
- Marmocchi P., Dall'Aglio C. e Zannini M. (2004), *Educare le life skills. Come promuovere le abilità psico-sociali e affettive secondo l'Organizzazione Mondiale della Sanità*, Centro Studi Erickson, Trento.
- Maslow A. H. (1973), *Motivazione e personalità*, Armando Editore, Roma.
- Matic A, Cipresso P, Osmani V, Serino S, Popleteev A, Gaggioli A, Mayora O, Riva G (2012). *Sedentary Work Style and Heart Rate Variability: a Short Term Analysis*. In: *Proceedings of the 2nd International Workshop on Computing Paradigms for Mental Health*. p. 96-101, Lisbona:SciTePress, Villamoura, Algarve, Portugal, 1 Febbraio 2012, doi: 10.5220/0003893000960101
- Maturana H. R. e Varela, F. J. (1985), *Autopoiesi e cognizione. L'organizzazione del vivente*, Marsilio, Venezia.
- Mauri M, Cipresso P, Balgera A, Villamira MA, Riva G (2011). *Why Is Facebook So Successful? Psychophysiological Measures Describe a Core Flow State While Using Facebook*. *CYBERPSYCHOLOGY, BEHAVIOR AND SOCIAL NETWORKING*, vol. 14, p. 723-731, ISSN: 2152-2715, doi: 10.1089/cyber.2010.0377

- Mauri M, Magagnin V, Cipresso P, Mainardi L, Brown EN, Cerutti S, Villamira M, Barbieri R (2010). Psychophysiological signals associated with affective states. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference. IEEE ENGINEERING IN MEDICINE AND BIOLOGY ... ANNUAL CONFERENCE, ISSN: 1557-170X
- Mauri M, Magagnin V, Cipresso P, Mainardi L, Brown En, Cerutti S, Villamira M, Barbieri R (2010). Psychophysiological signals associated with affective states. In: Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE. PROCEEDINGS OF THE ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY, p. 3563-3566, ISSN: 1094-687X, Buenos Aires, Aug. 31 - Sept. 4 2010
- Mayer J. D. e Salovey P. (1997), "What is emotional intelligence?" in Salovey P. e Sluyter D., Emotional development and emotional intelligence: Educational implications, Basic Books, New York.
- Mayer J. D., DiPaolo M. T. e Salovey P. (1990), "Perceiving affective content in ambiguous visual stimuli: A component of emotional intelligence.", *Journal of Personality Assessment*, 54, pp. 772-781.

- Mayer J. D., Salovey P. e Caruso D. R. (2002), Mayer-Salovey-Caruso Emotional Intelligence Test (MSCEIT), Multi-Health Systems, Toronto, Canada.
- McBurney, D. H. (2001). Metodologia della ricerca in psicologia: Il mulino.
- Meadows M. S. (2007), I, Avatar; The Culture and Consequences Of Having A Second Life, New Riders, Indianapolis.
- Metzger W. (1930), "Optische Untersuchungen am Ganzfeld: II zur Phänomenologie des homogenen Ganzfelds" Psychologische Forschung, 13.
- Miller G. A. (1983), "Informavores" in Machlup F. e Mansfield U., The study of information: interdisciplinary messages, Wiley, New York.
- Mininni G. (2004), Psicologia e media, Editori Laterza, Bari
- Minsky M. (1985), La società della mente, Adelphi, Milano.
- Morgan G. (1986), Images. Le metafore dell'organizzazione, Franco Angeli, Milano.
- Morin E. (1983), Il Metodo, ordine, disordine, organizzazione, Feltrinelli, Milano.
- Muth J.F. (1961), Rational Expectations and the Theory of Price Movements, *Econometrica* 29, 315-335
- NATO Advanced Study Institute in Information Science, Champion, Pa (1972), Information Science: Search for Identity, Marcel Dekker, New York.

- Norman D. A. (1981), "Twelve issues for cognitive science", in Norman D. A., *Perspectives on cognitive science*, Ablex Publishing, New Jersey.
- Norman D. A. (1990), *La caffettiera del masochista*, Giunti Editore, Firenze.
- Norman D. A. (1993), *Le cose che ci fanno intelligenti*, Feltrinelli, Milano.
- Norman D. A. (2000), *Il computer invisibile*, Apogeo, Milano.
- Norman D. A. (2003), *Emotional Design: Why We Love (or Hate) Everyday*
- Norman D. A. (2007), *The Design of Future Things*, Perseus Publishing.
- O'Regan J. K., Deubel H., Clark J. J. e Rensink R. A. (2000), "Picture change during blinks: looking without seeing and seeing without looking" *Visual Cognition* 7, pp. 191-212.
- O'Regan J. K., Rensink R. A. e Clark J. J. (1996), "'mud splashes' render picture change invisible" *Investigative Ophthalmology and Visual Science*, 37, S 213.
- Oatley K. (1992), *Psicologia ed emozioni*, il Mulino, Bologna.
- Ostrovsky Y, Mauri M, Citi L, Preda C E, Cipresso P, Barbieri R, Saccani M, Lenti C (2010). Gaze and Emotional Response in Visual Search Tasks Involving Faces within Scenes in Autistic Individuals: A Study Combining Psychophysiological and Eyetracking Measures. In: 57th Annual Meeting American Academy of Child and Adolescent Psychiatry Proceedings Abstracts. New York City, 26-31 Ottobre 2010

- Pallavicini F, Cipresso P, Raspelli S, Grassi A, Serino S, Vigna C, Triberti S, Villamira M, Gaggioli A, Riva G (2013). Is Virtual Reality Always an Effective Stressors for Exposure Treatments? Some Insights from a Controlled Trial. *BMC PSYCHIATRY*, vol. 13, 52, ISSN: 1471-244X, doi: 10.1186/1471-244X-13-52
- Pallavicini F, Gaggioli A, Raspelli S, Cipresso P, Serino S, Vigna C, Grassi A, Morganti L, Baruffi M, Wiederhold B, Riva G (2013). Interreality for the management and training of psychological stress: Study protocol for a randomized controlled trial. *TRIALS*, vol. 14, 191, ISSN: 1745-6215, doi: 10.1186/1745-6215-14-191
- Pantic M. (2005), "Affective Computing" in Pagani M., *Encyclopedia of Multimedia Technology and Networking*, vol. 1, Ed. Hershey, PA.
- Pedon, A., & Gnisci, A. (2004). *Metodologia della ricerca psicologica: il Mulino*.
- Pedroli E, Cipresso P, Serino S, Albani G, Riva G (2013). A Virtual Reality Test for the assessment of cognitive deficits: Usability and perspectives. In: *Proceedings of the 2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, PervasiveHealth 2013*. p. 453-458, ISBN: 978-193696880-0, 5-8 Maggio 2013, doi: 10.4108/icst.pervasivehealth.2013.252359
- Pedroli E, Cipresso P, Serino S, Pallavicini F, Albani G, Riva G (2013). Virtual multiple errands test: reliability, usability and

- possible applications. *STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS*, vol. 191, p. 38-42, ISSN: 0926-9630, doi: 10.3233/978-1-61499-282-0-38
- Perrow C. (1999), *Normal Accidents: Living with High-Risk Technologies*, Princeton University Press, New York.
- Piaget J. (1974), *Psicologia ed epistemologia*, Loescher, Torino.
- Piaget J. (1983), *Biologia e conoscenza*. Einaudi, Torino.
- Picard R. (1997), *Affective Computing*, MIT Press, Cambridge.
- Picard R. W. (1995), *Affective computing*, MIT Media Laboratory, Cambridge
- Picard R. W. (1997), *Affective computing*, MIT Press
- Picard R. W. (1999), *Affective computing for HCI*, MIT Media Laboratory, Cambridge
- Picard R. W. (2001), *Toward Machine Emotional Intelligence: Analysis of Affective Physiological State*, *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 23, No. 10
- Picard R. W. (2003), *Affective computing: Challenges*, *International Journal of Human-Computer Studies*, MIT Media Laboratory, Cambridge
- Picard R. W., Papert S., Bender W., Blumberg B., Breazeal C., Cavallo D., Machover T., Resnick M., Roy D. e Strohecker C. (2004), "Affective Learning - A Manifesto," *BT Technical Journal*, Volume 22, No. 4, pp. 253-269.
- Picard R. W., Vyzas E. e Healey J. (2001), "Toward Machine Emotional Intelligence: Analysis of Affective Physiological

- State," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 10.
- Picard R.W. (2003), "Affective Computing: Challenges," International Journal of Human-Computer Studies, Volume 59, Issues 1-2, pp. 55-64.
- Picard. R. (1997), Affective Computing. MIT, Press
- Pierantoni R. (2003), "Di come la "Gestalt" sfuggì alla rete. Uno studio sull'eclissi delle "forme"", in Antonucci R. e Pedemonte O. (a cura di), Il Tao del Web, Il Melangolo, Genova.
- Pilia M. e Coletta V. (2006), "Cultura ed errore nelle organizzazioni complesse", Rivista di sicurezza del Volo, 256, pp. 11-19.
- Pirolli P. (2003), "Exploring and Finding Information", in Carroll J. M., Toward a Multidisciplinary Science of Human-Computer Interaction, Morgan Kaufmann, Los Altos.
- Pirolli P. (2007), Information foraging; A theory of adaptive interaction with information, Oxford University Press, Cambridge.
- Pirolli P. e Card S. K. (1999), "Information foraging", Psychological Review 106(4), pp. 643-675.
- Plutchick R. (1980), Emotion: A Psychoevolutionary Synthesis, Harper and Row, New York
- Plutchick R. (2002), Emotions and life: perspectives for psychology, biology, and evolution, American Psychological Association (APA), Whashington, D.C., 592 pp.

- Plutchick R.; Conte H. R. (1997), *Circumplex Models of Personality and Emotions*, American Psychological Association (APA), Washington, D.C.
- Plutchick R.; Kellerman H. (1989), *Emotions – Theory, Research and experience*, Academic Press
- Plutchik R. (2001), *The nature of emotions*, Scienceweek
- Plutchik R. (1995), *Psicologia e biologia delle emozioni*, Bollati Boringhieri, Torino.
- Plutchik R. e Kellerman H. (a cura di) (1980), *Emotion: Theory, Research and Experience*, vol. 1-5, Academic Press, New York.
- Poletti B, Carelli L, Solca F, Meriggi P, Cipresso P, Pedroli E, Messina S, Morelli C, Ticozzi N, Lule D, Ludolph AC, Riva G, Silani V (2012). *Brain Computer Interface and Eye-Tracking Technology for Neuropsychological Assessment of ALS Patients. DEMENTIA AND GERIATRIC COGNITIVE DISORDERS*, vol. 33, p. 186-187, ISSN: 1420-8008
- Poletti B, Meriggi P, Lafronza A, Solca F, Carelli L, Ticozzi N, Messina S, Morelli C, Meazzi D, Cipresso P, Lombardi C, Riva G, Silani V (2011). *Anti-Saccade Paradigm and Cognitive Assessment in Amyotrophic Lateral Sclerosis. EUROPEAN JOURNAL OF NEUROLOGY*, vol. 18, p. 425, ISSN: 1351-5101
- Poletti B, Solca F, Annalisa L, Carelli L, Meriggi P, Cipresso P, Pedroli E, Ticozzi N, Morelli C, Messina S, Riva G, Silani V (2012). *Computerized Neuropsychological Assessment of*

Attention and Frontal Abilities in Amyotrophic Lateral Sclerosis. *DEMENTIA AND GERIATRIC COGNITIVE DISORDERS*, vol. 33, p. 185-186, ISSN: 1420-8008

Power M., Dalgleish T. (2008), *Cognition and emotion. From order to disorder*, Psychology Press, New York
powerHouse Books, New York.

Preece J.; Rogers Y. (1994), *Human-Computer Interaction*, Addison-Wesley

Prigogine I. (2003), *Le leggi del caos*, Laterza, Roma-Bari.

Rasmussen J. (1986), *Information Processing and Human-Machine Interaction. An Approach to Cognitive Engineering*, Elsevier, Amsterdam.

Raspelli S, Pallavicini F, Carelli L, Morganti F, Pedroli E, Cipresso P, Poletti B, Corra B, Sangalli D, Silani V, Riva G (2012). Validating the Neuro VR-based virtual version of the Multiple Errands Test: preliminary results. *PRESENCE-TELEOPERATORS AND VIRTUAL ENVIRONMENTS*, vol. 21, p. 31-42, ISSN: 1054-7460, doi: 10.1162/PRES_a_00077

Raspelli S, Pallavicini S, Grassi A, Cipresso P, Balgera A, Meazzi D, Gaggioli A, Villamira MA, Riva G (2011). Validation of a Narrative as an Emotional-Induction Technique through Different Non-Invasive Psychophysiological Monitoring Devices: Preliminary Results. *JOURNAL OF CYBERTHERAPY AND REHABILITATION*, vol. 4, p. 261-262, ISSN: 1784-9934

- Reason J. (1990), *Human Error*, Cambridge University Press, Cambridge.
- Reason J. (1994), *L'errore umano*, Il Mulino, Bologna.
- Reason J. (1997), *Managing the risks of organizational accidents*, Ashgate, Aldershot.
- Rensink R. A. (2002), "Change detection", *Annual Review of Psychology*, 53, pp. 245-277.
- Rensink R. A., O'Regan J. K. e Clark J. J., (1997), "To see or not to see. The need for attention to perceive changes in the scenes", *Psychological Science*, 8, pp. 368-373.
- Repetto C, Colombo B, Cipresso P, Riva G (2013). The effects of rTMS over the primary motor cortex: the link between action and language. *NEUROPSYCHOLOGIA*, vol. 51, p. 8-13, ISSN: 0028-3932, doi: 10.1016/j.neuropsychologia.2012.11.001
- Repetto C, Gaggioli A, Pallavicini F, Cipresso P, Raspelli S, Riva G (2013). Virtual reality and mobile phones in the treatment of generalized anxiety disorders: a phase-2 clinical trial. *PERSONAL AND UBIQUITOUS COMPUTING*, vol. 17, p. 253-260, ISSN: 1617-4909, doi: 10.1007/s00779-011-0467-0
- Repetto C, Gaggioli A, Pallavicini F, Cipresso P, Raspelli S, Riva G (2013). Virtual reality and mobile phones in the treatment of generalized anxiety disorders: a phase-2 clinical trial. *PERSONAL AND UBIQUITOUS COMPUTING*, vol. 17, p. 253-260, ISSN: 1617-4909, doi: 10.1007/s00779-011-0467-0

- Riera A, Soria-Frischa A, Albajes-Eizagirrea A, Cipresso P, Graua c, Dunne S, Ruffini G (2012). Electro-Physiological Data Fusion for Stress Detection,. STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS, vol. 181, p. 228-232, ISSN: 0926-9630, doi: 10.3233/978-1-61499-121-2-228
- Riva G, Cipresso P, Mantovani F, Dakanalis A, Gaggioli A (2013). New Technologies for Improving the Psychological Treatment. In: (a cura di): Capodaglio P, Faintuch J, Liuzzi A, Disabling Obesity. From Determinants to Health Care Models.. p. 269-284, ISBN: 978-3-642-35971-2, doi: 10.1007/978-3-642-35972-9_15
- Riva G, Gaggioli A, Grassi A, Raspelli S, Cipresso P, Pallavicini F, Vigna C, Gagliati A, Gasco S, Donvito G (2011). NeuroVR 2 - A Free Virtual Reality Platform for the Assessment and Treatment in Behavioral Health Care. STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS, vol. 163, p. 493-495, ISSN: 0926-9630, doi: 10.3233/978-1-60750-706-2-493
- Riva G, Vigna C, Grassi A, Raspelli S, Cipresso P, Pallavicini F, Gaggioli A (2012). Learning Island: The development of a virtual reality system for the experiential training of stress management. STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS, vol. 173, p. 369-371, ISSN: 0926-9630, doi: 10.3233/978-1-61499-022-2-369
- Roberts K. (1990), "Some characteristics of high reliability organizations", *Organization Science*, 1, pp. 160-177.

- Roberts K. (2005), *Lovemarks: The Future Beyond Brands* (Expanded edition),
- Roberts K. (2006), *The Lovemarks Effect: Winning in the Consumer Revolution*, powerHouse Books, New York.
- Roberts K.L., Hall D.A. (June, 2008), Examining a supramodal network for conflict processing: a systematic review and novel functional magnetic resonance imaging data for related visual and auditory stroop tasks, *Journal of cognitive neuroscience* 20 (6): 1063–78.
- Robins, G. & Kashima, Y. (2008). Social psychology and social networks.. *Asian Journal of Social Psychology*, 11, 1-12
- Rogers C. R. (1970), *La terapia centrata sul cliente*, Martinelli, Firenze.
- Rogers C. R. (1971), *Psicoterapia di consultazione*. Astrolabio, Roma.
- Rogers C. R. (1978), *Potere personale: la forza interiore e il suo effetto rivoluzionario*. Astrolabio, Roma.
- Rogers C. R. (2007), "Le condizioni essenziali di una relazione terapeutica facilitante", *Rivista di studi rogersiani*, 5-9.
- Roseman I. J. (1984), "Cognitive determinants of emotion: a structural theory" in Shaver P., *Review of Personality and Social Psychology: Emotions, Relationships, and Health*, Vol. 5, Sage, Beverly Hills.
- Ruelle D. (1992), *Caso e Caos*, Bollati Boringhieri, Torino.
- Rumelhart D. E. e McClelland J. L. (1991), *PDP. Microstruttura dei processi cognitivi*, Il Mulino, Bologna

- Rumiati R. e Bonini N. (1996), *Le decisioni degli esperti: psicologia cognitiva delle decisioni manageriali*, il Mulino, Bologna.
- Russell J. A. (1980), A circumplex model of affect, *Journal of Personality and social psychology*, 39, 1161 – 1178
- Salovey P. e Mayer J. (1990), "Emotional intelligence", *Imagination, Cognition and Personality*, 9, pp. 185-211.
- Salovey P.; Mayer J. D. (1990), *Emotional intelligence*, Baywood Publishing Co., Inc.
- Sargent, T.J.: *Bounded Rationality in Macroeconomics*, Oxford University, Press, Oxford, 1993
- Schachter S. e Singer J. E. (1962), "Cognitive, social and physiological determinants of emotional state", *Psychological Review*, 69, pp. 379-399.
- Schrödinger E. (1945), *What is Life?* tr. i.: *Che cos'è la vita?* (1992), Adelphi, Milano.
- Senge P. (2006), *La quinta disciplina. L'arte e la pratica dell'apprendimento organizzativo*, Sperling e Kupfer Editori, Milano.
- Serino S, Cipresso P, Gaggioli A, Pioggia G, Tartarisco G, Baldus G, Corda D, Riva G (2012). *Beyond the Lab: Towards the Mobile Assessment*. *APPLIED PSYCHOPHYSIOLOGY AND BIOFEEDBACK*, ISSN: 1090-0586
- Serino S, Cipresso P, Gaggioli A, Riva G (2013). *The Potential of Pervasive Sensors and Computing for Positive Technology: The Interreality Paradigm*. In: (a cura di): Mukhopadhyay S C

- et al., *Pervasive and Mobile Sensing and Computing for Healthcare*, SSMI 2. vol. 2, p. 207-232, Springer, ISBN: 978-3-642-32537-3, doi: 10.1007/978-3-642-32538-0_10
- Serino S, Cipresso P, Tartarisco G, Baldus G, Corda D, Pioggia G, Gaggioli A, Riva G (2012). An Event-Driven Psychophysiological Assessment for Health Care. In: *Proceedings of the 2nd International Workshop on Computing Paradigms for Mental Health*. p. 25-34, Lisbona:SciTePress, Villamoura, Algarve, Portugal, 1 Febbraio 2012, doi: 10.5220/0003884200250034
- Serino S, Cipresso P, Vigna C, Gaggioli A, Riva G (2012). An ubiquitous platform for fear of flying research. *JOURNAL OF CYBERTHERAPY AND REHABILITATION*, vol. 5, p. 138-139, ISSN: 1784-9934
- Serino S, Triberti S, Villani D, Cipresso P, Gaggioli A, Riva G (2013). Toward a validation of cyber-interventions for stress disorders based on stress inoculation training: a systematic review. *VIRTUAL REALITY*, ISSN: 1359-4338, doi: 10.1007/s10055-013-0237-6
- Shannon C. E. (1948), "A mathematical theory of communication", *Bell Sys. Tech. Journal*, 27, 379-423, 623-656.
- Shannon C. E. e Weaver W. (1949), *The Mathematical Theory of Communication*, University of Illinois Press, Urbana.
- Shepard R. N. (1984), "Ecological constraints on internal representation: resonant kinematics of perceiving, imagining,

- thinking and dreaming", *Psychological Review*, 91(4), pp. 417-447.
- Shore D. I. e Klein R. M. (2000), "The effects of scene inversion on change blindness", *Journal of General Psychology*, 127(1), pp. 27-43.
- Simon H. (1997), *Models of Bounded Rationality*, Vol. 3. MIT Press
- Simon H. A. (1955), "A behavioral model of rational choice", *Quarterly Journal of Economics*, 69, pp. 99-118.
- Simon H. A. (1985), *Causalità, razionalità, organizzazione*, il Mulino, Bologna.
- Simon H. A. (1987), "Satisficing", in Eatwell J., Milgate M. e Newman P., *The New Palgrave, a new dictionary of economics*, vol. 4, MacMillan, London.
- Simons D. J. (2000), *Change blindness and visual memory: a special issue of visual cognition* Psychology Press, Philadelphia, PA.
- Simons D. J. e Levin D. T. (1998), "Failure to detect changes to people during a real-world interaction", *Psychonomic Bulletin and Review*, 5(4), pp. 644-649.
- Smith B. (1988), *Foundations of Gestalt Theory*, Philosophia Resources Library
- Smith C. A. e Ellsworth P. C. (1985), "Patterns of cognitive appraisal in emotion", *Journal of Personality and Social Psychology*, 48(4), pp. 813-838.

- Snowdon B. and Vane H. (2002), *An Encyclopedia of Macroeconomics*, Edward Elgar Publishing, ISBN 1840643870, 9781840643879
- Soro A. (2008), *Human computer interaction – Fondamenti e prospettive*, Polimetrica s.a.s.
- Sternberg R. J. (2000), *Psicologia cognitiva*, Piccin, Padova.
- Sternberg R. J. e Spear-Swerling L. (1997), *Le tre intelligenze. Come potenziare le capacità analitiche, creative e pratiche*, Erickson, Trento.
- Stewart I. (1993), *Dio gioca a dadi?*, Bollati Boringhieri, Torino.
- Strack F., Martin L. L. e Stepper S. (1988), "Inhibiting and facilitating conditions of the human smile: A nonobtrusive test of the facial feedback hypothesis", *Journal of Personality & Social Psychology*, 54(5), pp. 768-777
- Stroop, J.R. (1935), *Studies of interference in serial verbal reactions*, *Journal of Experimental Psychology* 18: 643-662; <http://psychclassics.yorku.ca/Stroop/>
- Tanoni I. (2003), *Videogiocando s'impara. Dal divertimento puro all'insegnamento-apprendimento*, Erickson, Milano.
- Terna P., Boero R., Morini M., Sonnessa M., *Modelli per la complessità. La simulazione ad agenti in economia*, Bologna, Il Mulino, 2006
- Tesfatsion, L. (2006), *Agent Based Computational Economics: A Constructive Approach to Economic Theory*, in *Handbook of Computational Economics*, Tesfatsion L. and Judd K.L. (Eds.), 831-80, Vol. 2, North-Holland, Amsterdam

Things, Basic Books, New York,

Thorndike E. L. (1920), "Intelligence and its use", Harper's Magazine, 140, pp. 227-235.

Tomkins S. (1984), Affect theory, in Ekman P.; Scherer K. R. (a cura di), (1984), Approaches to emotions, Erlbaum, Hillsdale, pp.129-157

Tomkins S. S. (1981), "The role of facial response in the experience of emotion", Journal of Personality and Social Psychology, 40, pp. 351-357.

Turkle S. (1997), La vita sullo schermo. Nuove identità e relazioni sociali nell'epoca di Internet, Apogeo, Milano.

Tutzauer, F. & Elbirt, B. S. The Distribution of Entropy-Based Centralization in Sparse, Directed Networks. (2007). Proceedings from International Communication Association. San Francisco, CA:.

Van Rooy D. L. e Viswesvaran C. (2004), "Emotional intelligence: A meta-analytic investigation of predictive validity and nomological net", Journal of Vocational Behavior, 65(1), pp. 71-95.

Villamira M, Cipresso P (2010). Bio-Inspired ICT for Evolutionary Emotional Intelligence. In: (a cura di): Serra R, Villani M, Poli I, Artificial Life and Evolutionary Computation. ISBN: 978-981-4287-44-9

Villamira M. A. (1995), Comunicazione e interazione. Aspetti del comportamento interpersonale e sociale, Franco Angeli, Milano.

- Villamira M. A. e Roggeroni L. D. (1999), Inter Actio, Franco Angeli, Milano.
- Villamira M. A., Baraldi D., Balgera A. e Cavelzani A. (2007), "Psicologia dell'alimentazione: il cibo come varietà neghentropica" in Stroppa C. (a cura di), Cibo, società e scienza dell'alimentazione, Aracne editrice, Roma.
- Villamira M., Cipresso P. (2008), Bio-inspired ICT for evolutionary emotional intelligence, In: Proceedings of Italian Workshop in Artificial Life and Evolutionary Computation – WIVACE 2008, Venezia, Università di Modena e Reggio Emilia, ISBN/ISSN: 978-88-903581-0-4
- Villamira M.A., Bracco F. (2009), Comunicare. Elementi di psicologia della comunicazione, FrancoAngeli, Milano
- Villani D, Grassi A, Cognetta C, Cipresso P, Toniolo D, Riva G (2012). The Effects of a Mobile Stress Management Protocol on Nurses Working with Cancer Patients: a Preliminary Controlled Study. STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS, vol. 173, p. 524-528, ISSN: 0926-9630, doi: 10.3233/978-1-61499-022-2-524
- Villani D, Grassi A, Cognetta C, Toniolo D, Cipresso P, Riva G (2013). Self-Help Stress Management Training Through Mobile Phones: An experience with oncology nurses. PSYCHOLOGICAL SERVICES, vol. 10, p. 315-322, ISSN: 1541-1559, doi: 10.1037/a0026459
- Villani D, Repetto C, Cipresso P, Riva G (2012). May I experience more presence in doing the same thing in virtual reality than

in reality? An answer from a simulated job interview. INTERACTING WITH COMPUTERS, vol. 24, p. 265-272, ISSN: 0953-5438, doi: 10.1016/j.intcom.2012.04.008

Waldrop M. M. (1995), *Complessità: uomini e idee al confine tra ordine e caos*, Instar Libri, Torino.

Watson J. B. e Rayner R. (1920), "Conditioned emotional reactions", *Journal of Experimental Psychology*, 3, pp. 1-14.

Watzlawick P., Beavin J. H., Jackson D. D. (1967), *Pragmatica della Comunicazione Umana*, Astrolabio, Roma.

Watzlawick P., Weakland J. H. e Fish R., (1974), *Change. Sulla formazione e la soluzione dei problemi*, Astrolabio, Roma.

Wechsler D. (1943), "The non-intellective factors in general intelligence", *Journal of Abnormal and Social Psychology*, 38, pp. 100-104.

Weick K. E. e Sutcliffe K. M. (2007), *Managing the Unexpected: Resilient Performance in an Age of Uncertainty*, Jossey-Bass, San Francisco.

Weiser M. (1991), "The Computer for the Twenty-First Century" *Scientific American*, Vol. 265, No. 3 pp. 94-104.

Weppner J, Lukowicz P, Serino S, Cipresso P, Gaggioli A, Riva G (2013). Smartphone based experience sampling of stress-related events. In: *Proceedings of the 2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*, PervasiveHealth 2013. p. 464-467, ISBN: 978-193696880-0, 5-8 Maggio 2013, doi: 10.4108/icst.pervasivehealth.2013.252358

- Wiener N. (1948), *Cybernetics, or Control and Communication in the Animal and the Machine*, MIT Press and Wiley, New York
- Wienhardt, T. (2005). Netzwerke als Solidaritätspotential. Gad, G., Hiß, S., & Wienhardt, T., (Eds.). *Wirtschaft, Ethik und Entwicklung - Wie passt das zusammen?* (ed.). (pp. 120-140) Berlin: WVB
- Wienhardt, T. (2006). Netzwerke als Potential katholischer Entwicklungszusammenarbeit. *Soziales Kapital: Faktor solidarischer Institutionengestaltung* (Doctoral dissertation, 2006).
- Wienhardt, T. (2006). Netzwerkfaktoren gegen Arbeitslosigkeit. Kurzke-Maasmeier, S., Mandry, C., & Oberer, C., (Eds.). *Baustelle Sozialstaat! Sozialethische Sondierungen in unübersichtlichem Gelände* (ed.). (pp. 165-^180) Münster: Aschendorff Verlag
- Wienhardt, T., Gad, G., & Hiss, S. (Eds.). (2005). *Wirtschaft, Ethik und Entwicklung - Wie passt das zusammen?* (ed.). Berlin: WVB
- Woolf S. H., Kuzel A. J., Dovey S. M. e Phillips R. L. (2004), "A string of mistakes: the importance of cascade analysis in describing, counting and preventing medical errors", *Annals of Family Medicine*, 2(4), pp. 317-326.
- Wreathall, J. (2006), "Properties of Resilient Organizations: An Initial View", in Hollnagel E., Woods D. D. e Levenson N. G., *Resilience Engineering-Concepts and Precepts*, Ashgate Press, London.

- Zajonc R. B., Murphy S. T. e Inglehart M. (1989), "Feeling and facial efference: Implications of the vascular theory of emotion", *Psychological Review* 96, pp. 395-416.
- Zammuner, V. L. (1996). *Interviste e questionari: processi psicologici e qualità dei dati*: Borla.
- Zammuner, V. L. (1998). *Tecniche dell'intervista e del questionario*: il Mulino.
- Zechmeister, J. S. E. S. J. (2012). *Metodologia della ricerca in psicologia* (edizione italiana a cura di M. Lanz, G. Amoretti, S. Tagliabue). McGrawHill, Milano.

INDICE ANALITICO

- .m; 236; 237; 239; 254; 256;
260; 280; 295; 301; 306; 319
- AAC; 178; 394; 395; 400
- Accedere remotamente ai dati;
355
- Accelerazione; 128
- Accelerometri; 91; 128
- Accostare matrici in verticale;
223
- Activity of Daily Living
(ADL); 173
- Adobe Flash; 35
- Aereo virtuale; 162
- Aeroporto virtuale; 161
- Affective Computing; 403;
405; 585
- Affidabilità statistica; 475
- Algebra lineare; II; 209; 211;
221; 248; 365
- Algoritmi di classificazione;
405
- Algoritmo FFT di Sande-
Tukey; 439
- Algoritmo per il calcolo della
Trasformata discreta di
Fourier; 439
- Alpha (EEG); 66
- Alternative and Augmented
Communication (AAC); 178;
394; 395; 400
- Ambiente controllato; 160;
169
- Ambiente reale; 157
- Ambienti 3D; 156
- Ambienti tridimensionali; 156;
158
- Ambito clinico; 64; 156; 164;
171; 178
- Ammiccamento; 65; 86; 88

- Ampiezza; 434
- Ampiezza di fase; 482
- Ampiezza di un'oscillazione periodica; 474
- Analisi armonica; 433; 474; 476
- Analisi dei comportamento; 91
- Analisi dei segnali; 59; 433; 437
- Analisi qualitativa; 106; 107; 112; 126
- Analizzare i dati relazionali di rete; 372
- Android; 131; 133; 180
- Angolo di fase; 434
- Annotazione; 108; 109; 110; 111
- Anonymise Network; 152
- Antisaccadico; 77
- AOI. *Vedi* aree di interesse
- Apache; 30; 343
- API; 119; 134; 550
- App; 129; 130; 131; 132; 133; 134; 135; 180; 403; 404
- Applicazioni web; 340; 360
- Appunti vocali; 130
- Architetture di "Cloud Computing"; 363
- Aree di interesse; 81
- Armonica di frequenza; 434
- Armonica reale; 434
- Arousal; 57; 603
- Array; 36; 39; 272
- Artefatti; 65; 67; 279; 330; 391
- ASCII; 70
- Aspetti stocastici; 414
- Assegnazioni; 218
- Assi cartesiani; 128; 281
- Attenzione visiva; 178
- Attivazione simpatica; 63
- Attività di vita quotidiana; 172; 174
- Atto respiratorio; 63
- Audacity (software); 101
- Augmentative and Alternative Communication (AAC); 178; 394
- Auto valori; 473
- Auto vettori; 473
- Autocorrelazioni; 479
- Autospettri; 479

- Barometri; 129
- Baseline; 70; 100; 242; 332;
333; 334; 335; 337
- Basickeylogger (software); 94;
98; 100
- Batteria neuropsicologica; 400
- Battito cardiaco; 65; 404
- BCI; 178; 392; 394; 395; 396;
399; 400; 401; 402
- BCI basata su ERP P300; 394
- BCI per la comunicazione e/o
per la motorizzazione
controllata; 396
- BCI2000 (software); 399
- Behavioral experiment
software; 106; 112
- Beta (EEG); 66
- Betweenness Centrality; 384;
386
- Big data; 20; 339; 379; 405
- Biofeedback; 403; 404; 405
- Biosensori; 57; 59; 60; 62; 67;
68; 69; 185; 403
- Bit-reverse; 439
- Blinking; 65; 86
- Blood Volume Pulse; 64; 585;
603
- Bluetooth; 69
- Bluetooth Smart; 404
- Brain Computer Interface
(BCI); 178; 392; 394; 395;
396; 399; 400; 401; 402
- Broker; 144; 384; 386
- Buco strutturale (reti); 144
- BVP; 64; 585; 603
- Calcolo automatico; 69
- Calcolo dei gradi di libertà
equivalente; 469
- Campi unione; 265
- Campionamento; 59; 61; 193
- Campioni di saliva; 127
- Canali; 245; 396; 397; 603
- CAQDAS; 106
- Caratterizzare un processo
stocastico; 417
- Caratterizzazioni sintetiche
dei processi stocastici; 417;
425
- Carico cognitivo; 88
- Casco di realtà virtuale; 157
- Caso multivariato; 477

- Celle; 195; 311; 312; 317;
324; 333; 334; 336; 366;
533; 534; 535
- Chiusura della finestra; 468
- Chroma key; 168
- Cicli di periodo; 444
- Ciclo "for"; 310; 311; 314;
315; 319; 320; 321; 325;
326; 327; 329; 330; 333;
336; 337; 370; 388
- Closeness; 153
- Cloud Computing; 362
- Codice aperto; 29; 45; 94; 98;
124 *Vedi anche* Open source
- Codice generato; 266; 274
- Codici a barre; 103
- Coefficiente di correlazione
lineare; 482
- Coefficiente di correlazione
lineare nel dominio
frequenziale; 483
- Coerenza; 222; 224; 477; 482;
483
- Coerenza spettrale; 482
- Comandi interattivi; 164
- Comando "if"; 305
- Command Window; 254
- Compilare; 21; 33; 34
- Complessità; 58; 59; 83; 87;
121; 124; 138; 158; 163;
178; 180; 185; 186; 194;
196; 197; 209; 339; 343;
357; 361; 366; 379; 391;
413; 440; 489; 492; 493;
494; 496; 499; 501; 516;
517; 522; 549; 587; 592
- Complessità computazionale;
158; 438; 439
- Complessità dei dati; 58
- Compliance del paziente; 172
- Componenti dell'ERP; 393
- Componenti periodiche
separate; 476
- Computazione continua; 155
- Computazione dati in realtime;
391
- Comunicazione aumentativa e
alternativa; 178; 394; 395;
400
- Comunicazione via Gmail;
383; 384
- Condizione "if"; 305; 306

- Condizione di compatibilità; 416
- Condizione di simmetria; 416
- Condizioni; 304
- Conduttanza cutanea; 57; 63; 67; 193; 413; 414
- Conduttanza elettrica della pelle *vedi* Conduttanza cutanea
- Configurazione della matrice di lettere; 399
- Coniugazione complessa; 475
- Contatti sociali; 160
- Contenuti 3D clinicamente testati; 180
- Contesti ecologici; 126; 134
- Contesto; 20; 28; 43; 97; 126; 128; 133; 138; 139; 145; 157; 573; 579; 588
- Convergenza a zero nel senso di Césaro; 427
- Convoluzione in frequenza; 421
- Convoluzione in tempo; 420
- Coordinate di posizione; 169
- Coordinate di Talairach; 397; 398
- Co-presenza; 179
- Correlati psicofisiologici; 413
- Correlazione campionaria incrociata; 478
- Correlazioni di intervallo; 479
- Correlazioni incrociate; 479
- Correlogramma; 483
- Coseno; 431; 434; 485; *Vedi* funzioni trigonometriche
- Co-spettro; 481
- Costanza nel tempo dei momenti; 424
- Covarianza; 417; 423; 425; 426; 428; 429; 430; 432; 442; 446; 449; 454; 477; 478; 481
- Covarianza campionaria incrociata; 478
- Covarianza stazionaria; 424
- Creazione di una matrice; 214
- Creazione di una sottomatrice; 239
- csv (formato); 47; 51
- Data Collection; 131; 132

- Database; 20; 30; 45; 149;
339; 340; 341; 342; 343;
346; 347; 348; 349; 354;
355; 356; 357; 358; 359;
359; 361; 362; 379; 404
- Dati complessi; 187; 198; 269;
303
- Dati comportamentali; 91
- Dati di navigazione; 169
- Dati empirici; 138
- Dati relazionali; 149; 365;
372; 389
- Dato di relazione; 146
- Decimation in frequency; 439
- Decimation in time; 439
- Definizione delle epoche; 242
- Degree; 153; 604; 606; 608;
609; 614; 615; 616; 617; 618
- Delimitatore; 190; 202; 255
- Delta; 66
- Densità di probabilità; 431;
432
- Densità spettrale; 425; 430;
431; 432; 442; 443; 444;
445; 446; 447; 451; 454;
460; 465; 467; 470; 481; 483
- Densità spettrale del processo;
425; 430
- Densità spettrali non
normalizzate; 480
- Densità spettrali normalizzate;
479
- Densità teorica fissa; 445
- Derivazioni (ECG); 64
- Descrizione assi; 295
- Diagonale principale; 146; 214
- Diagrammi a blocco; 304
- Diametro della pupilla; 87
- Differenziabile; 429
- Digit; 59
- Dimensione della matrice
quadrata; 370
- Direzione dei legami; 146;
147
- Discrete Prolate Spheroidal
Sequences; 473
- Dispersione; 417; 458
- Dispositivi di tipo uditivo;
156
- Dispositivi interattivi; 177
- Distanza R-R; 64
- Disturbi elettrici; 61

- Domainmath IDE (software);
199
- Dominio frequenziale; 433;
437
- Dominio temporale; 433; 483
- DPSS. *Vedi* Discrete Prolate
Spheroidal Sequences
- Ebrain Engine; 400; 401; 402
- ECG; 64; 65; 73; 585
- EDA; 57; 63
- Edges; 377
- EEG; 65; 73; 116; 392; 395;
396; 403; 585; 603; 630
- Efficacia dell'uso della RV;
156
- Ego; 152
- EKG. *Vedi* ECG
- Elaborazione grafica; 155
- ELAN (software); 106; 107;
108; 109; 111
- Elearning; 30; 350
- Elementi in una matrice; 216
- Elementi naturali 3D; 167
- Elettrocardiogramma; 64; 69;
413; 414
- Elettrodi; 63; 64; 66
- Elettroencefalografia; 66
- Elettroencefalogramma; 69;
392; 413; 414
- Elettromiografia; 65
- Elettrooculografia; 65
- EMG; 65; 73; 207; 585
- Emoticon; 34
- Epoca; 71; 73; 243; 246; 247;
331; 334; 335; 664
- E-Prime; 78; 112
- Equazioni differenziali; 437;
527; 560
- Ergodicità rispetto alla
funzione di covarianza; 428
- Ergodicità rispetto alla media;
426
- Ergodico; 426; 427; 429
- ERP; 393; 395
- Esercizi riabilitativi; 172;
173; 177
- Esplorazione di dati; 193
- Esportazione; 42; 45; 46; 72;
73
- Esprimere emozioni; 405
- Estrazione; 60; 67; 70; 78; 82;
86; 87; 150; 381; 414; 495

- Event related potential; 393; 395
- Eventi; 93; 94; 95; 102; 169; 171; 426; 497; 500; 582; 584
- Evoluzione temporale dell'esperimento casuale; 415
- Experimental builder; 80
- Eye-tracker; 77; 78; 80; 83; 87; 88; 112; 119; 178; 185; 400; 401; 402; 414
- Facebook; 93; 149; 150; 151; 152; 366; 381; 385; 386; 648
- Famiglia di variabili aleatorie; 415
- Fase. *Vedi* angolo di fase
- Fast Fourier Transform; 439
- Fenomeni deterministici; 469
- FFT; 439; 440
- FFT radix-2; 440
- Fibre muscolari; 65
- Figure separate; 283; 284; 294; 295; 298; 299
- Figure totalmente indipendenti fra loro; 296
- File di log; 169
- File multimediale; 109
- Filtering; 438
- Filtro digitale; 438
- Finestra dei ritardi; 449; 452; 454; 457; 458; 460; 461; 463; 465; 466
- Finestra dei ritardi di Barlett; 465
- Finestra dei ritardi di Parzen; 464
- Finestra del periodogramma troncato; 452
- Finestra del ritardo; 449; 450; 451; 452; 460
- Finestra di Barlett; 454; 458; 467
- Finestra di Blackman–Tukey generale; 458
- Finestra di Parzen; 463; 465
- Finestra di Tukey; 460; 461
- Finestra di Tukey–Hamming; 460
- Finestra di Tukey–Hanning; 461
- Finestra rastremata; 472

- Finestra rettangolare; 451;
455; 458; 459
- Finestra spettrale; 449; 450;
451; 452; 453; 454; 455;
457; 458; 459; 460; 462;
463; 464; 465; 466; 467; 469
- Finestra spettrale adatta al
modello; 468
- Finestra spettrale di Parzen;
464
- Finestra spettrale di Tukey-
Hanning; 461
- Finestrata; 471; 472; 473
- Fissazioni; 82; 83; 84; 85; 86;
400
- Flash; 35; 69
- Fogli di calcolo; 194
- Foglio di calcolo; 27; 28; 186;
366
- Forma polare; 481
- Formati; 28; 42; 107; 326
- Forze di rotazione; 128
- Fotometri; 129
- Frequenza; 434
- Frequenza angolare; 437; 471;
472
- Frequenza di campionamento;
59; 132; 229; 237
- Frequenze; 66; 430; 435; 444;
458; 460; 477
- Funzione del tempo; 415
- Funzione di correlazione
incrociata; 478
- Funzione di covarianza; 418;
424; 426
- Funzione di covarianza della
successione stazionaria
reale; 431
- Funzione di covarianza
incrociata; 477; 483
- Funzione di densità di
probabilità congiunta; 477
- Funzione di distribuzione; 432
- Funzione generica di Matlab;
253
- Funzione non decrescente; 429
- Funzione numerica; 415
- Funzione periodica; 433
- Funzione reale nel dominio
del tempo; 415
- Funzione spettrale; 429; 431

- Funzioni cognitive; 112; 171; 400
- Funzioni di auto covarianza (momento secondo); 477
- Funzioni di covarianza campionarie; 446
- Funzioni di covarianza incrociata; 477
- Funzioni di ripartizione doppie; 416
- Funzioni di ripartizione n-uple; 416
- Funzioni di ripartizione semplici; 416
- Funzioni esecutive; 173
- Funzioni temporali; 415
- Funzioni trigonometriche; 434; 485
- Gaussiano; 425; 429
- Gazeplot; 84
- Generatore di finestra spettrale; 466
- Generazione di reti di esempio; 380
- Gephi (software); 376; 377; 378; 379; 380; 381; 382; 384; 385; 386; 557
- Gesti; 109; 121; 124
- Gestire la fobia sociale; 179
- Gesture analysis; 121
- Giroscopi; 128
- Global Positioning System; 127
- GNU Octave; 197; 198; 199; 224; 236; 249; 269; 280; 282; 379; 389
- Go/no go task; 112
- Google Drive; 22; 23; 27; 28
- GPS; 91; 127
- Gradi di libertà equivalenti; 469
- Grafica tridimensionale; 155
- Graphml (formato file); 152
- GSR; 63; 585; 603
- Head mounted display; 163
- Head-tracker; 157
- Heatmap; 84
- HeidiSQL; 354; 355
- Hertz; 59

- Hz; 59; 61; 83; 87; 132; 192;
193; 232; 238; 269
- IBI; 65
- Imaginazione motoria; 395
- Immersivo; 157
- Implementazione; 249; 339;
579
- Import dei dati di posta
elettronica; 382
- Importazione; 45; 47; 50; 51;
70; 107; 167; 187; 188; 189;
190; 191; 192; 193; 201;
202; 203; 204; 206; 230;
251; 252; 253; 254; 255;
256; 257; 264; 273; 275;
315; 317; 318; 321; 323;
377; 381; 382
- Indagini; 20; 33; 135
- Indici di rete; 367
- Input; 118; 119; 121; 176;
177; 178; 253; 254; 304;
305; 306; 319; 353; 439;
509; 518; 543; 603
- Inquisit (software); 112
- Integrale di convoluzione;
418; 419
- Intensità emotiva; 88
- Inter-Beat Interval; 65
- Interferenze; 61; 279
- Interferenze elettriche; 61
- Interrelazioni fra due segnali
psicofisiologici; 483
- Interstress; 403; 406
- Intervallo RR; 64; 65
- Intervento sui dati in realtime;
400
- iOS; 131; 404
- iPad; 176; 180; 403
- isEmpty; 306; 307
- Ispezionare visivamente i
segnali; 299
- Ispezione visiva; 61; 83; 148;
187; 329
- ISPY; 102; 103
- Iterazioni; 304; 308; 439; 532
- Jack-knifing; 474
- Joomla; 30; 349
- Kernel di Dirichlet; 454; 458;
459
- K-esimo auto spettro del
segnale; 473

- Kinect; 120; 121; 122; 123;
124; 125; 126; 176; 177
- Labeling; 329
- Lag window; 449
- Larghezza di banda; 467; 468;
469; 472; 476
- Legame; 139; 146; 148; 367
- Legenda; 289; 290; 295; 329;
334
- Legge Fisher-Snedecor; 476
- Libreria; 30; 166; 167
- Librerie; 30; 122; 547; 550;
557; 561
- Limesurvey; 28; 29; 30; 31;
32; 33; 34; 36; 43; 45; 46;
48; 51; 52; 53; 127; 343;
344; 345; 346; 347; 348; 349
- Limiti di spazio; 35
- Linden; 158
- Linden Dollar; 158
- Linea continua; 61; 288
- Linee spettrali di un segnale
periodico o quasi periodico;
474
- Linee spurie con frequenze
arbitrarie; 476
- Linux; 106; 198; 376; 547;
550
- Logging sui percorsi
effettuati; 163
- Logica di indagine; 40
- Logica di processo; 197; 229;
249; 251; 269
- Logica di struttura; 221
- Loop; 116; 117; 628
- Luce riflessa; 64
- Luminosità; 89
- Mac OS X; 106; 376
- Magnetometri; 129
- Mancanza della consistenza;
444
- Manipolazione di ambienti
virtuali; 165
- Manipolazione dell'audio; 101
- Manipolazione delle matrici;
269; 388; 389
- Marker; 82; 93; 167; 179; 635
- Marketplace; 158; 159
- Matrice delle densità spettrali
non normalizzate; 480
- Matrice spettrale; 480
- Matrice Toeplitz; 473

- Matrici e le loro
trasformazioni; 221
- Media quadratica; 427
- Memoria locale; 69; 127
- Metafora probabilistica per i
segnali raccolti; 414
- Metodo di massima entropia;
474
- Metodo di Thomson; 471
- Metodo Multitaper; 470
- Microfono; 92; 98; 99
- Micro-Siemens; 63
- Microsoft SQL Server; 30
- Microvolt; 64; 65; 66
- Miliardi di dati; 58; 86
- Minimizzare le perdite
spettrali; 472
- Misurazione dei tempi di
reazione; 91
- Misure psicofisiologiche; 56
- Mixed-radix; 440
- Modellazione; 138; 142; 526;
547; 577; 583
- Moduli; 23; 30; 344; 557; 560;
561
- Moltiplicazione in tempo; 421
- Mondi virtuali; 160
- Monitor; 21; 82; 94; 99; 104;
157; 394
- Monitoraggio a distanza; 77
- Monitoraggio continuo; 403;
405
- Motor imagery based BCI; 395
- Mouse; 31; 92; 93; 94; 97; 98;
100; 108; 112; 118; 119;
166; 205; 401
- Movie Maker; 102
- Movimenti oculari; 65; 77; 78;
83; 84; 86; 115; 178; 402
- MSSQL; 30
- MTM. *Vedi* metodo Multitaper
- Multitasking; 251
- Muscoli facciali; 65
- MySQL; 30; 45; 340; 342;
343; 346; 347; 348; 349;
354; 355; 358; 359; 404
- Namegenweb (software); 150;
385
- Navigazione in ambienti
virtuali; 165
- Network Analysis; 377
- Neuroplasticità; 171

- Neuropsicologia; 112; 156
- Neuroriabilitazione; 156; 171
- Neuroscienze; 91; 120; 579; 587
- Neurovirtual 3D; 175; 176; 177; 179; 180
- NeuroVR (software); 164; 165; 166; 167; 168; 169; 171; 176; 177; 178; 179; 633; 658
- NodeXL (software); 372; 373; 374; 375
- Normale multivariata; 425
- Nucleo di trasformazione; 436; 437
- Numero di utenti; 33
- N-upla di istanti temporali; 416
- Octave. Vedi GNU Octave
- Oggetti 3D; 162; 167; 168
- Oggetti virtuali; 166
- On screen capture recorder to video free; 104
- Onde; 66; 75; 77; 98; 166; 392; 526; 592; 612
- Open Graph Viz Platform
Gephi; 376
- Open source; 29; 94; 101; 102; 104; 105; 106; 113; 122; 198; 199; 342; 344; 349; 372; 376; 400; 553; 554; 555
vedi anche Codice aperto
- OpenKinect (software); 122
- OpenNI (software); 122
- Opensesame (software); 113; 114; 115; 116; 117; 119; 400
- Operazione di convoluzione; 418
- Opzioni di colore; 288
- Organizzazione strutturale dei dati complessi; 339
- Orientarsi in una matrice; 215
- Orologio di sistema; 93
- Ortogonalità; 473
- P300; 393; 395; 396; 629
- Pagina web; 24; 32
- Pannello di amministrazione; 31
- Paradigm (software); 112; 640; 655; 660

- Paralisi dei movimenti oculari;
395
- Parametri ambientali; 129
- Parametri non aleatori; 435
- Parametro psicofisiologico; 58
- Patch adesiva; 63
- Paura di parlare in pubblico;
179
- Paura di Volare 3D (App); 162
- PDO (database driver); 30
- Percezione; 178; 575; 597;
625; 640
- Percorso (nel senso di tragitto
compiuto); 97; 128; 169; 170
- Performance osservabili; 172
- Periodo di una funzione
periodica; 433
- Periodogramma; 440; 442;
444; 445; 446; 447; 448;
451; 452; 460; 467
- Periodogramma troncato; 446;
452; 460
- PERL; 30; 343; 358
- Pervasive and ubiquitous
computing; 363
- Pervasività; 130; 363
- Peso della relazione; 367
- PHP (linguaggio); 30; 343;
350; 353; 358
- Piattaforma web; 343; 405
- Player; 165
- Pletismografia; 413
- Plotting; 279; 280; 309; 310;
311; 314; 315; 324; 325;
327; 328; 334; 335; 336
- Plug-in; 114
- Pool dei file; 116
- Pose; 124; 126
- Positive Technology App; 403;
404
- Posizione di collegamento;
144
- Postgres; 30
- Potenziale elettrico; 65
- Precisione temporale; 83
- Presentation (software); 78;
112; 115
- Pressione atmosferica; 129
- Pressione dei tasti; 92; 95;
100
- Principio di incertezza di
Grenander; 468

- Probabilità di emergere, dato un certo tipo di comportamento.; 144
- Problema agli auto vettori; 472
- Problema Rayleigh-Ritz; 472
- Processamento; 121; 155; 158
- Processi attentivi; 77
- Processi continui a parametro discreto; 416
- Processi gaussiani; 425
- Processi stazionari; 421; 428
- Processo automatizzato; 150
- Processo computazionale; 438
- Processo stocastico; 143; 414; 415; 416; 417; 421; 422; 423; 424; 425; 426; 429; Vedi Processi stocastici
- Processo stocastico a covarianza stazionaria; 424
- Processo stocastico stazionario in senso stretto; 422; 424
- Processo White Noise *Vedi* Rumore bianco
- Programmazione; 20; 80; 121; 163; 164; 199; 248; 269; 271; 309; 343; 350; 386; 400; 547; 549; 554; 560
- Programmazione parallela; 334
- Prospettiva teorica della complessità. *Vedi* Complessità
- Provider; 30; 359
- Pseudo-codice; 320; 322; 549
- Psicofisiologia; 55; 56; 58; 60; 61; 64; 65
- Psicoterapia; 156
- Psychlog; 406; 407; 408
- Pulsantiera; 91; 92; 114
- Pulsazione; 65
- Punto di troncamento; 446; 448; 458; 467
- Punto di troncamento; 467
- Pupillometria; 87
- Purple Robot (software/App); 133; 134
- Pylink; 119
- Python (linguaggio); 95; 113; 114; 117; 358; 386; 400;

- 552; 554; 558; 560; 561;
562; 563; 615; 617
- Qualche relazione
trigonometrica; 485
- Qualtrics (software); 29
- Questionari elettronici; 20;
21; 22; 23; 24; 25; 26; 27;
28; 29; 31; 32; 33; 36; 38;
39; 40; 41; 42; 43; 45; 127;
137; 148; 155; 185
- Radix-r; 439
- Random-graph models; 143
- RDBMS; 342
- Realizzazione del processo
stocastico; 415
- Realtà virtuale (RV); 120;
126; 155; 156; 157; 158;
163; 165; 166; 169; 171;
172; 173; 174; 175; 177;
179; 181; 391; 404
- Realtime; 63; 158; 171; 400;
402; 403
- Registrazione del segnale; 67
- Registrazione video; 103
- Regressione nel senso dei
minimi quadrati nel dominio
delle frequenze; 474
- Relax; 48; 52; 70; 72; 73; 242;
282; 288; 291; 332; 337;
603; 604; 606
- Relazioni di matrimonio; 140;
142
- Relazioni di rete; 371
- Relazioni politiche; 141; 142
- Renamer (software); 74; 268
- Repository; 176; 180
- Resize; 167
- Respirazione; 63
- Rete diretta; 146; 367
- Rete pesata; 146; 367
- Rete sociale; 138
- Reti amicali; 137
- Reti complesse; 142; 144;
551; 557; 560
- Reti familiari; 137
- Reti sociali; 137; 139; 140;
145; 372; 601
- RGB; 121
- Riabilitazione; 120; 121; 126;
171; 173; 180

- Ricerca comportamentale;
106; 112
- Ricerca di informazioni; 178
- Riconoscere il movimento;
103
- Riconoscimento delle
emozioni; 405; 578
- Riconoscimento delle pose;
123
- Riflesso dell'infrarosso; 64
- Rilevazioni al secondo; 58; 83
- Rinascimento; 140
- Risposta elettrodermale *Vedi*
Risposta galvanica cutanea
- Risposta galvanica cutanea;
63; 585; 603 *Vedi* GSR
- RQDA (software); 111
- Rstudio (software); 111; 200
- Rumore bianco; 445; 476; 483
- Rumore rosso; 476
- Ruoli chiave; 144
- Ruotation; 167
- Saccadico; 77
- Salivette; 127
- Salvataggio dei dati; 328; 388
- Sampling rate; 237; 240; 254;
Vedi Frequenza di
campionamento
- Scalari; 209
- Scanner 3D; 123
- Schema internazionale 10-20;
397
- Sclerosi laterale amiotrofica
(SLA); 395
- SD card; 69
- Sdk; 80
- Second Life; 158; 159; 160;
161; 162; 163; 164; 179; 650
- Segnale analogico; 59
- Segnale grezzo; 66; 67
- Segnale in epoche; 245
- Segnale sinusoidale; 471
- Segnali alterati; 61; 62
- Segnali campionati; 68; 416
- Segnali elettrici; 65
- Segnali psicofisiologici; 60;
61; 63; 67; 70; 72; 86; 194;
206; 230; 241; 245; 248;
260; 269; 279; 315; 403;
414; 433; 437
- Segnali raccolti; 67; 70; 469

- Seno; 281; 434; 441; 485;
Vedi funzioni
trigonometriche
- Senso di Presenza; 172
- Sensor Data; 132
- Sensori ambientali; 129
- Sensori dei vettori di
rotazione; 128
- Sensori di gravità; 128
- Sensori di movimento; 128
- Sensori di orientamento; 129
- Sensori di posizione; 129
- Sequenza numerica; 75
- Serie osservata; 436
- Sessioni sperimentali; 242;
288
- Signal processing; 414; 431
- Simbolo "%"; 235
- Simulare esperienze; 156
- Simulazione dell'area corticale
e dello scalpo; 397
- Simulazioni 3D interattive;
180
- Sintassi; 47; 50; 51; 197; 198;
213; 216; 218; 287; 297;
300; 324; 561
- Sistema caotico; 476
- Sistema Internazionale della
frequenza; 59
- Sistema nervoso simpatico; 57
- Sito web; 25; 30; 36; 547;
550; 551; 561
- Situazioni cliniche; 169
- Situazioni reali; 169
- Smartphone; 21; 33; 91; 103;
127; 128; 129; 130; 131;
135; 174; 180; 362; 405
- Smooth pursuit; 77; 178
- SMS; 22
- Social Network Analysis; 139;
144; 148; 149; 153; 372; 389
- Social Network Sites (SNSs);
149
- Social networks; 21; 145; 635;
659
- Software development kit
(sdk); 80
- Sonno; 65
- Sotto-domande (di un
questionario); 36; 37
- Sottomatrici; 220; 221; 240

- Spazio di probabilità continuo;
416
- Spazio web; 29; 30; 33; 343
- Spettri; 473; 480
- Spettri incrociati; 480
- Spettro continuo; 476
- Spettro di quadratura; 481;
484
- Spettro grezzo di potenza; 476
- Spettro integrato o cumulato;
432
- SPSS (software); 47; 48; 50
- SQL; 30; 45; 354
- SR Research; 80
- Stabilità delle stime di
frequenza e di ampiezza; 476
- Stati affettivi; 405
- Statistica fisica; 142
- Stato termico; 63
- Stazionarietà in senso lato;
424
- Stereo-camere; 102
- Stile di linea; 288
- Stima asintoticamente
corretta; 444
- Stima dei segnali; 440
- Stima del co-spettro; 484
- Stima della funzione di
coerenza spettrale; 484
- Stima della potenza del
segnale; 472
- Stima dello spettro totale; 473
- Stima spettrale semplice; 474
- Stimatore asintoticamente non
distorto; 443; 447
- Stimatore Blackman–Tukey;
460
- Stimatore di Tukey–Hanning;
461
- Stime indipendenti dello
spettro reale; 473
- Stimolo; 78; 81; 82; 84; 85;
392; 400; 401; 607
- Strategic models of network
formation; 143
- Streaming; 132; 133
- Stress; 48; 52; 70; 72; 123;
242; 282; 288; 291; 332;
337; 405; 603; 604; 606;
626; 630; 645; 652; 658;
661; 666
- Strutturare le reti; 367

- Strutture di base dei dati; 209
- Strutture di rete; 144
- Studio del comportamento;
144
- Studio delle connessioni; 142
- Studio longitudinale; 145
- Studio stereotassico; 397
- Subplotting; 296; 297; 298;
299; 564
- Successione; 65; 78; 210; 211;
225; 226; 426; 427; 428; 430
- Successione di numeri; 210;
211; 225
- Successione di variabili
casuali; 416
- Successione stazionaria; 428
- Successioni sferoidali prolate
discrete; 473; 475
- Successioni stazionarie reali;
430; 431; 442
- Superfici 3D; 162
- Surveymonkey (software); 29
- Tablet; 21; 33
- Tabulazioni; 190; 371
- Tastiera; 92; 93; 94; 98; 100;
112; 119; 166; 402
- Tattile; 156
- Telemedicina; 180; 405
- Temperatura; 63; 129; 414
- Temperatura periferica; 63
- Tempi di risposta; 113; 115
- Tempi morti; 93
- Tempi sperimentali; 239; 241
- Tendenziosità della stima; 446
- Teorema del campionamento
di Nyquist-Shannon; 59; 505
- Teorema di Kolmogorov; 416
- Teorema di Parseval; 436
- Teoria dei giochi; 143
- Teoria dei grafi; 142
- Teoria dei segnali; 414
- Terapia virtuale; 160
- Termistore; 63
- Test di Fisher; 475
- Theta (EEG); 66
- Ties; 151; 377
- Timeline; 78; 79; 100; 107;
108; 109
- Timestamp; 95; 100; 119; 169
- Tobii Studio (software); 79
- Tracciare il movimento della
testa; 157

- Tracciato oculare; 179
- Training personalizzato; 171
- Transfer of training; 174
- Translation; 167
- Trasformata coseno di Fourier; 431
- Trasformata d'integrali; 436
- Trasformata di Fourier; 420; 425; 430; 431; 437; 438; 449; 450; 452; 455; 466; 471; 485
- Trasformata di Fourier di una serie periodica infinita; 474
- Trasformata di Laplace; 437
- Trasformata discreta di Fourier; 438
- Trasformata discreta di Fourier della serie finestrata; 473
- Trasformata esponenziale di Fourier; 480
- Trasformata inversa di Fourier; 438; 450
- Trasformazioni lineari; 483
- Trasposta; 211; 214; 231
- Triangolo di Einthoven; 64
- Triggering; 163
- Unità immaginaria; 437; 485
- Validità ecologica; 179
- Valore atteso; 417; 423; 424; 446; 500; 501; 576
- Valutazione di stimoli ambientali; 178
- Variabile aleatoria; 414; 415; 425; 428; 435
- Variabile casuale; 415; 475
- Variabile casuale continua; 416
- Varianza "spiegata" dal modello; 475
- Varianza residua; 475
- Variazioni della pupilla. *Vedi* diametro della pupilla
- VBA *Vedi* Visual Basic for Application
- Vedere parti diverse dello stesso segnale; 299
- Versione pdf; 42
- Vettore colonna; 211; 215; 217; 223; 231
- Vettore di direzione anteriore; 169

- Vettore riga; 211; 215; 217;
231; 280; 281
- Visual Basic for Application
(linguaggio); 269; 272; 273;
274; 275; 276; 277
- Webcam; 102; 104
- Wii; 177
- Windows; 75; 102; 106; 120;
198; 376; 547; 550
- Wizard; 201; 251; 252; 254;
262; 323
- Workspace; 204; 205; 224;
228; 229; 230; 232; 238;
241; 243; 244; 246; 247; 258
- WYSIWYG; 34; 37; 38; 43
- XAMPP; 343; 344; 345; 347;
350
- Xtion; 120
- Youtube Video Downloader
(software); 104; 105

Note di copyrights

Tutti i marchi riportati appartengono ai legittimi proprietari; marchi di terzi, nomi di prodotti, nomi commerciali, nomi corporativi e società citati possono essere marchi di proprietà dei rispettivi titolari o marchi registrati d'altre società e sono stati utilizzati a puro scopo esplicativo ed a beneficio del possessore, senza alcun fine di violazione dei diritti di Copyright vigenti.

Alcune immagini contenute nel volume potrebbero essere *screenshots* relativi a un'opera dell'ingegno protetta da copyright. Questi possono essere riprodotti sul volume, in osservanza dell'articolo 70 comma 1 della legge 22 aprile 1941 n. 633 sulla Protezione del diritto d'autore e di altri diritti connessi al suo esercizio, modificata dalla legge 22 maggio 2004 n. 128, poiché trattasi di «*riassunto, [...] citazione o [...] riproduzione di brani o di parti di opera [...]*» utilizzati «*per uso di critica o di discussione*», o per mere finalità illustrative e per fini non commerciali, e in quanto la presenza nel volume non costituisce «*concorrenza all'utilizzazione economica dell'opera*». Queste immagini non possono essere utilizzate per altri scopi senza il previo assenso dell'azienda o dell'autore titolare dei diritti.

Informazioni sul materiale supplementare al volume

Tutto il materiale supplementare al volume si trova al sito web:

www.metricamente.it

