This Maple worksheet accompanies the papers:

Di Nardo E., G. Guarino, D. Senato (2007), *A new method for fast computing unbiased estimators of cumulants*. In press Statistics and Computing.
http://www.springer.com/statistics/computational/journal/11222 (download from http://www.unibas.it/utenti/dinardo/lavori.html)

Fast Maple algorithms for k-statistics, polykays and their multivariate generalization

E. Di Nardo*
elvira.dinardo@unibas.it
http://www.unibas.it/utenti/dinardo/home.html;
Tel: +39 0971205890, Fax: +39 0971205896

G. Guarino**
giuseppe.guarino@asl2.potenza.it

D. Senato*
domenico.senato@unibas.it

* Dipartimento di Matematica e Informatica, Università degli Studi della Basilicata,Viale dell'Ateneo Lucano n.10, 85100 Potenza, Italy

**Medical Scool, Università del Sacro Cuore (Rome branch), Largo Agostino Gemelli n.8, 00168 Roma, Italy

Introduction

***Abstract*:** We provide four algorithms to generate single and multivariate k-statistics and single and multivariate polykays. The computational times are very fast compared with the procedures available in the literature. Such speeding up is obtained through a symbolic method arising from the classical umbral calculus. The classical umbral calculus is a light syntax to manage sequences of numbers or polynomials, involving only elementary rules. The keystone of the procedures here introduced is the connection, achieved by a symbolic device, between cumulants of a random variable and a suitable compound Poisson random variable. Such a connection holds also for multivariate random variables.
Application Areas/Subject: Combinatorics & algebraic methods in statistics.
Keyword: umbral calculus, symmetric polynomials, set partitions, multiset, cumulants, k-statistics, polykays.
See Also: Maple algorithm [3], [6]
Remark: k-statistics, polykays and their multivariate generalization are commonly defined in terms of power sums, that are sums of the rth powers of the data points:

$$S_r = \sum_{i=i}^{n} X_i^r$$

Initialization

> *restart*

> *with*(*combinat*, *partition*, *multinomial*, *stirling2*)

[*partition*, *multinomial*, stirling2] **(2.1)**

k-statistics

The nth k-statistic is the unique symmetric unbiased estimator of the cumulant κ_n of a given statistical distribution.

k_n is defined so that E[k_n] = κ_n

> *makeTab_ss* := **proc**(*N*)

$$\left[seq\left(\left[\frac{N!}{mul\left(x!^{numboccur(y,\,x)} \cdot numboccur(y, x)!, x = \{op(y)\} \right)} \cdot mul\left(k_i, i = y\right), mul\left(S_i, i = y\right) \right], y = partition(N) \right) \right]$$

end proc:

makeK_s := **proc**(*N*)

$$\left[seq\left(k_i = add\left(\text{stirling2}(i, j) \cdot x^j \cdot (-1)^{j-1} \cdot (j-1)!, j = 1..i \right), i = 1..N \right) \right]$$

end proc:

fd := **proc**(*j*, *k*) *expand*(*mul*(*n* − *i* − *k*, *i* = 0 .. *j* − *k*)) **end proc** :

ks := **proc**(*N*)

local *u*, *v*;

v := *expand*(*eval*(*makeTab_ss*(*N*), *makeK_s*(*N*)));

$$u := \left[seq\left(x^i = (-1)^{i-1} \cdot (i-1)! \cdot fd(N-1, i), i = 1..N \right) \right];$$

$$\frac{add\left(x_1 \cdot x_2, x = expand(eval(v, u)) \right)}{mul((n - x), x = 0..N-1)}$$

end proc:

Example

> *ks*(3)

(3.1)

$$\frac{2\,S_1^3 - 3\,n\,S_1\,S_2 + n^2\,S_3}{n\,(n-1)\,(n-2)} \tag{3.1}$$

> $st := time(\,) : ks(21) : time(\,) - st$

$$0.250 \tag{3.2}$$

Example of k-statistics construction (k_3)

> $v := makeTab\_ss(3)$

$$v := \left[\left[k_1^3, S_1^3\right], \left[3\,k_1\,k_2, S_1\,S_2\right], \left[k_3, S_3\right]\right] \tag{3.1.1}$$

> $u := makeK\_s(3)$

$$u := \left[k_1 = x, k_2 = x - x^2, k_3 = x - 3\,x^2 + 2\,x^3\right] \tag{3.1.2}$$

> $v := expand(eval(v, u))$

$$v := \left[\left[x^3, S_1^3\right], \left[3\,x^2 - 3\,x^3, S_1\,S_2\right], \left[x - 3\,x^2 + 2\,x^3, S_3\right]\right] \tag{3.1.3}$$

> $u := \left[seq\left(x^i = (-1)^{i-1}\,(i-1)!\,fd(3-1, i), i = 1..3\right)\right]$

$$u := \left[x = n^2 - 3\,n + 2, x^2 = -n + 2, x^3 = 2\right] \tag{3.1.4}$$

> $\dfrac{add\left(x_1\,x_2, x = expand(eval(v, u))\right)}{mul(n - x, x = 0..3 - 1)}$

$$\frac{2\,S_1^3 - 3\,n\,S_1\,S_2 + n^2\,S_3}{n\,(n-1)\,(n-2)} \tag{3.1.5}$$

Test previouse result

> $ks(3)$

$$\frac{2\,S_1^3 - 3\,n\,S_1\,S_2 + n^2\,S_3}{n\,(n-1)\,(n-2)} \tag{3.1.6}$$

> $evalb(\% = `\%\%`)$

$$true \tag{3.1.7}$$

Note on "fd" function

fd(x, y): x is the lower factorial and y is the numbers of factors to delete from left of lower factorial expression.
Example: the decreasing factorial $(n)_3$ = n*(n-1)*(n-2)*(n-3)

> $fd(3, 0);\quad expand(n\,(n-1)\,(n-2)\,(n-3))$

$$n^4 - 6\,n^3 + 11\,n^2 - 6\,n$$

$$n^4 - 6\,n^3 + 11\,n^2 - 6\,n \tag{3.2.1}$$

Deleting "n" from $(n)_3$

> $fd(3, 1);\quad expand((n-1)\,(n-2)\,(n-3))$

$$n^3 - 6\,n^2 + 11\,n - 6$$

(3.2.2)

$$n^3 - 6\,n^2 + 11\,n - 6 \tag{3.2.2}$$

Deleting "n*(n-1)" from $(n)_3$

> $fd(3,2);\quad expand((n-2)\,(n-3))$

$$n^2 - 5\,n + 6$$

$$n^2 - 5\,n + 6 \tag{3.2.3}$$

Remark: if we want to calculate the following expression:

$$\frac{A}{n\,(n-1)} + \frac{B}{n\,(n-1)\,(n-2)}$$

we have to compute:

$$\frac{A\,(n-2)}{(n)_3} + \frac{B}{(n)_3} = \frac{A\,(n-2)+B}{(n)_3}$$

where $(n-2)$ is obtained from $(n)_3$ deleting the first two terms.

Example: compare the results of the expressions computed with and without "fd" function.
Without "fd" function

> $F := \frac{A}{n\,(n-1)} + \frac{B}{n\,(n-1)\,(n-2)}$

$$F := \frac{A}{n\,(n-1)} + \frac{B}{n\,(n-1)\,(n-2)} \tag{3.2.4}$$

> $simplify(F)$

$$\frac{A\,n - 2\,A + B}{n\,(n-1)\,(n-2)} \tag{3.2.5}$$

> $R1 := \frac{collect(numer(F), \{A, B\}, distributed)}{denom(F)}$

$$R1 := \frac{A\,(n-2)+B}{n\,(n-1)\,(n-2)} \tag{3.2.6}$$

With "fd" function. This metod is used in functions generating k-statistics and polykays.

> $\frac{A\,fd(3-1,2)}{`(n)`_3} + \frac{B}{`(n)`_3} = \frac{A\,fd(3-1,2)+B}{`(n)`_3}$

$$\frac{A\,(n-2)}{(n)_3} + \frac{B}{(n)_3} = \frac{A\,(n-2)+B}{(n)_3} \tag{3.2.7}$$

> $F := A\,fd(3-1,2) + B\,fd(3-1,3)$

$$F := A\,(n-2) + B \tag{3.2.8}$$

> $R2 := \frac{F}{fd(3-1,0)}$

$$R2 := \frac{A\,(n-2)+B}{n^3 - 3\,n^2 + 2\,n} \tag{3.2.9}$$

> $simplify(R1 - R2)$

(3.2.10)

Polykays

The symmetric statistic $k_{r, s, ..}$ is defined as

$$\mathrm{E}[k_{r, s, ..}] = \kappa_r \kappa_s ...$$

where κ_r is a cumulant. These statistics called polykays generalize k-statistics.

```
> makeCTR_s := proc(N)
    [seq(k[i] = add(mul(mu[k], k = v) * (-1)^(nops(v)-1) * (nops(v)-1)! i! / mul((x)!^numboccur(v, x) * numboccur(v, x)!, x = {op(v)}), v = partition(i)), i = 1..N)]
  end proc:

  makeMu := proc()
    local u, v, N, eu;
    N := add(i, i = args);
    eu := [seq(mu[i] = 1, i = 1..N)];
    if nargs = 1 then
      u := [seq([[x]], x = partition(args[1]))]
    else
      u := [seq([seq(x, x = partition(i))], i = args)]
    end if;
    u := op(add(mul((-1)^(nops(i)-1) * (nops(i)-1)! * mul(mu[k], k = i) * countP(i), i = v)
    * fd(N - 1, add(nops(k), k = v)) * countP([seq(op(i), i = v)])^(-1), v = `if`(nargs = 1,
    comb(u, 1, []), [comb(u, 1, [])]));
    seq(simplify(v / eval(v, eu)) = eval(v, eu), v = u)
  end proc:

  comb := proc(V, ptr, Y)
    if ptr = nops(V) + 1 then
      return Y
    end if;
    seq(comb(V, ptr + 1, [op(Y), L]), L = V[ptr])
  end proc:

  countP := proc(u)
    add(x, x = u)! / mul(x^numboccur(u, x) * numboccur(u, x)!, x = {op(u)})
  end proc:
```

```
ps := proc( )

    local u, v, N;
    N := add(x, x = args);
    u := expand(eval(makeTab_ss(N), makeCTR_s(N) ) );
    v := expand( eval( u, [makeMu(args), μ = 0] ) );
          add(x[1]·x[2], x = v)
    ------------------------------
    mul(n − x, x = 0 ..N − 1)

end proc:
```

```
> ps(2, 1)
```

$$\frac{-S_1^3 + (1+n)\,S_1\,S_2 - n\,S_3}{n\,(n-1)\,(n-2)} \tag{4.1}$$

```
> st := time( ) : ps(8, 7) : time( ) − st
```

$$0.297 \tag{4.2}$$

Example of polykays construction ($k_{2;1}$)

```
> v := makeTab_ss(3)
```

$$v := \left[\left[k_1^3, S_1^3\right], \left[3\,k_1\,k_2, S_1\,S_2\right], \left[k_3, S_3\right]\right] \tag{4.1.1}$$

```
> u := makeCTR_s(3)
```

$$u := \left[k_1 = \mu_1, k_2 = -\mu_1^2 + \mu_2, k_3 = 2\,\mu_1^3 - 3\,\mu_1\,\mu_2 + \mu_3\right] \tag{4.1.2}$$

```
> u := expand(eval(v, u))
```

$$u := \left[\left[\mu_1^3, S_1^3\right], \left[-3\,\mu_1^3 + 3\,\mu_1\,\mu_2, S_1\,S_2\right], \left[2\,\mu_1^3 - 3\,\mu_1\,\mu_2 + \mu_3, S_3\right]\right] \tag{4.1.3}$$

```
> vEval := [makeMu(2, 1), μ = 0]
```

$$vEval := \left[\mu_1^3 = -1, \mu_1\,\mu_2 = \frac{1}{3}\,n - \frac{2}{3}, \mu = 0\right] \tag{4.1.4}$$

```
> v := expand(eval(u, vEval))
```

$$v := \left[\left[-1, S_1^3\right], \left[1+n, S_1\,S_2\right], \left[-n, S_3\right]\right] \tag{4.1.5}$$

```
>   add(x[1] x[2], x = v)
    ------------------------------
    mul(n − x, x = 0 ..3 − 1)
```

$$\frac{-S_1^3 + (1+n)\,S_1\,S_2 - n\,S_3}{n\,(n-1)\,(n-2)} \tag{4.1.6}$$

Test previous result

```
> ps(2, 1)
```

$$\frac{-S_1^3 + (1+n)\,S_1\,S_2 - n\,S_3}{n\,(n-1)\,(n-2)} \tag{4.1.7}$$

```
> evalb(%= `%%`)
```

true **(4.1.8)**

Multiset subdivision

The following algorithm function is used for listing all subdivision of a multiset. This algorithm is fully discussed in [3]
Note: the algorithm is necessary for multivariate case. It is recalled only one time for every parameter input of the multivariate function. This speeds up the procedure.

```
> nRep :=proc(u) mul(x2!, x = convert(u, multiset)) end proc:
```

```
URv :=proc(u, v)
    local U, ou, i, ptr, vI;
    ou := NULL; U := [ ]; vI := indets(v);
    for ptr from nops(u) by -1 to 2 do
        if has(u[ptr], v) then break end if
    end do;
    for i from ptr to nops(u) do
        if not (u[i] = ou or has(u[i], vI)) then
          ou := u[i];
          U := [op(U), [op(u[1..i-1]), u[i]*v, op(u[i+1..-1])]]
        end if
    end do;
    op(U), [op(u), v]
end proc:
```

```
URV :=proc( )
    local U, V, i;
    U := [args[1, 1]]; V := args[2, 1];
    for i to nops(V) do U := [seq(URv(u, V[i]), u = U)] end do;
    seq([x, args[1, 2]*args[2, 2]], x = U)
end proc:
```

```
URmV :=proc( )
    local U, i;
    if nargs = 1 then args else
        U := URV(args[1], args[2]);
        for i from 3 to nargs do
            U := seq(URV(u, args[i]), u = [U])
        end do;
        seq([x[1], x[2]/nRep(x[1])], x = U)
    end if
end proc:
```

```
makeTab := proc( )
    local U;
    if add(x, x = args) = 0 then return 0 end if;
    U := [seq( `if`(args_i = 0, NULL, [seq( [[seq( (P||i)^z, z = y )], multinomial(args_i, seq(r, r
    = y)) ], y = partition(args_i) ) ]), i = 1 ..nargs ) ];
    if nops(U) = 1 then
        [ seq( [ x_1, x_2 / nRep(x_1) ], x = op(U) ) ]
    else
        [seq(URmV(op(x)), x = [comb(U, 1, [ ])])]
    end if
end proc:
```

Multivariate k-statistics

```
> makeTab_sm := proc( )
    local vP, U;
    U := makeTab(args);
    vP := sort( [seq( P||i, i = 1 ..nargs )] );
    [seq( [mul( k_{add(degree(x, vP_i), i = 1 ..nops(vP))}, x = y_1 ) ·y_2, mul( S_{seq(degree(x, vP_i), i = 1 ..nops(vP))}, x
    = y_1 ) ], y = U ) ]
end proc:
```

```
km := proc( )
    local u, v, N;
    N := add(x, x = args);
    v := expand(eval(makeTab_sm(args), makeK_s(N)));
    u := [seq( x^i = (-1)^(i-1) · (i - 1)! · fd(N - 1, i), i = 1 ..N )];
    add( x_1 · x_2, x = expand(eval(v, u)) ) / mul(n - x, x = 0 ..N - 1)
end proc:
```

```
> km(2, 1)
```

$$\frac{-2\,n\,S_{1,1}\,S_{1,0} + 2\,S_{1,0}^2\,S_{0,1} + n^2\,S_{2,1} - n\,S_{2,0}\,S_{0,1}}{n\,(n-1)\,(n-2)} \tag{6.1}$$

```
> km(1, 1, 1)
```

$$\frac{n^2\,S_{1,1,1} - n\,S_{1,1,0}\,S_{0,0,1} - n\,S_{1,0,1}\,S_{0,1,0} - n\,S_{1,0,0}\,S_{0,1,1} + 2\,S_{1,0,0}\,S_{0,1,0}\,S_{0,0,1}}{n\,(n-1)\,(n-2)} \tag{6.2}$$

```
> st := time( ) : km(8, 7) : time( ) - st
```

$$1.234 \tag{6.3}$$

Example of multivariate k-statistics construction ($k_{2,1}$)

> $v := makeTab\_sm(2, 1)$

$$v := \left[\left[2\,k_2\,k_1, S_{1,1}\,S_{1,0}\right], \left[k_1^3, S_{1,0}^2\,S_{0,1}\right], \left[k_3, S_{2,1}\right], \left[k_2\,k_1, S_{2,0}\,S_{0,1}\right]\right] \quad \textbf{(6.1.1)}$$

> $u := makeK\_s(3)$

$$u := \left[k_1 = x, k_2 = x - x^2, k_3 = x - 3\,x^2 + 2\,x^3\right] \quad \textbf{(6.1.2)}$$

> $v := expand(eval(v, u))$

$$v := \left[\left[2\,x^2 - 2\,x^3, S_{1,1}\,S_{1,0}\right], \left[x^3, S_{1,0}^2\,S_{0,1}\right], \left[x - 3\,x^2 + 2\,x^3, S_{2,1}\right], \left[x^2 - x^3, S_{2,0}\,S_{0,1}\right]\right] \quad \textbf{(6.1.3)}$$

> $u := \left[seq\left(x^i = (-1)^{i-1}\,(i-1)!\,fd(3-1, i), i = 1..3\right)\right]$

$$u := \left[x = n^2 - 3\,n + 2, x^2 = -n + 2, x^3 = 2\right] \quad \textbf{(6.1.4)}$$

> $\dfrac{add\left(x_1\,x_2, x = expand(eval(v, u))\right)}{mul(n - x, x = 0..3 - 1)}$

$$\frac{-2\,n\,S_{1,1}\,S_{1,0} + 2\,S_{1,0}^2\,S_{0,1} + n^2\,S_{2,1} - n\,S_{2,0}\,S_{0,1}}{n\,(n-1)\,(n-2)} \quad \textbf{(6.1.5)}$$

Test previous result

> $km(2, 1)$

$$\frac{-2\,n\,S_{1,1}\,S_{1,0} + 2\,S_{1,0}^2\,S_{0,1} + n^2\,S_{2,1} - n\,S_{2,0}\,S_{0,1}}{n\,(n-1)\,(n-2)} \quad \textbf{(6.1.6)}$$

> $evalb(\%= `\%\%`)$

$$true \quad \textbf{(6.1.7)}$$

Multivariate polykays

> $makeTab\_mm := \textbf{proc}()$
> $\quad \textbf{local}\ vP, U;$
> $\quad U := makeTab(args);$
> $\quad vP := sort([op(indets(U))]);$
> $\quad \left[seq\left(\left[mul\left(k_{seq\left(degree\left(x, vP_i\right), i = 1..nops(vP)\right)}, x = y_1\right)\cdot y_2, mul\left(S_{seq\left(degree\left(x, vP_i\right), i = 1..nops(vP)\right)}, x = y_1\right)\right], y = U\right)\right]$
> $\textbf{end proc}:$

> $ctr := \textbf{proc}()$
> $\quad \textbf{local}\ vP, U;$
> $\quad U := makeTab(args);$
> $\quad vP := [seq(P||i, i = 1..nargs)];$
> $\quad add\left(v_2\cdot(-1)^{nops\left(v_1\right)-1}\cdot\left(nops\left(v_1\right) - 1\right)!\cdot mul\left(\mu_{seq\left(degree\left(x, vP_i\right), i = 1..nops(vP)\right)}, x = v_1\right), v\right.$

$= makeTab(args)$]

end proc:

makeCTR_m :=**proc**( )

$[seq(\bar{k}_{op(i)} = ctr(op(i)), i = comb([seq([seq(x, x = 0..y)], y = args)], 1, [\,])\,)]$

end proc:

unionVects :=**proc**(*U*::*list*, *V*::*list*)

if $nops(U) = 0$ **then** V

elif $nops(V) = 0$ **then** U

else $[seq(seq([sort([op(v_1), op(u_1)])], v_2 \cdot u_2], u = U), v = V)]$

end if

end proc:

ricVtab :=**proc**(*v*, *V*, *N*)

local *u*, *vv*;

$vv := sort(v_1)$;

for *u* **in** *V* **do**

if $sort(u_1) = vv$ **then**

return $v_2 \cdot \dfrac{fd(N-1, nops(v_1))}{u_2}$

end if

end do;

return 0

end proc:

pm :=**proc**( )

local *N*, *M*, *u*, *v*, *i*, *vK*, *vTab*, *vP*, *vParts*, *vEval*;

$M := \max(seq(nops(x), x = args))$;

$vP := [seq(P||i, i = 1..M)]$;

$u := op(add(k, k = seq([op(h), seq(0, y = nops(h)..M-1)], h = args)))$;

$v := expand(eval(makeTab\_mm(u), makeCTR\_m(u)))$;

$N := add(h, h = u)$;

$vTab := [\,]$;

for *i* **to** *nargs* **do**

$vTab := unionVects(vTab, [seq([v_1, v_2 \cdot (-1)^{nops(v_1)-1} \cdot (nops(v_1)-1)!], v = makeTab(op(args_i)))])$

end do;

if $nops(vTab) > 1$ **then**

$vTab := [seq([op(i)_1, \dfrac{op(i)_2}{2}], i = mul(u_1^{2 \cdot u_2}, u = vTab))]$

end if;
$vParts := makeTab(u)$;
$vTab := [seq([x_1, ricVtab(x, vParts, N)], x = vTab)]$;
$vEval := \left[seq\left(mul\left(\mu_{seq(degree(x, vP_i), i=1..nops(vP))}, x = y_1\right) = y_2, y = vTab\right), \mu = 0\right]$;
$\frac{add(x_1 \cdot x_2, x = eval(v, vEval))}{mul(n - x, x = 0..N - 1)}$
end proc:

> $pm([1, 1], [1])$

$$\frac{n\, S_{1,1}\, S_{1,0} - S_{1,0}^2\, S_{0,1} - n\, S_{2,1} + S_{2,0}\, S_{0,1}}{n\,(n-1)\,(n-2)} \tag{7.1}$$

> $pm([1, 1], [1], [1])$

$$\frac{1}{n\,(n-1)\,(n-2)\,(n-3)}\left(n\, S_{1,1}\, S_{1,0}^2 - S_{1,0}^3\, S_{0,1} - n\, S_{1,1}\, S_{2,0} - 2\, n\, S_{1,0}\, S_{2,1} + 3\, S_{1,0}\, S_{2,0}\, S_{0,1} + 2\, n\, S_{3,1} - 2\, S_{3,0}\, S_{0,1}\right) \tag{7.2}$$

> $st := time() : pm([4, 3], [3]) : time() - st$

$$0.344 \tag{7.3}$$

Example of multivariate polykays construction ($k_{1,1;1,0}$)

M is the max order of elements in { [1,1],[1] }
N is (1 + 1) + (1)

> $M := 2; N := 3$

$$M := 2$$
$$N := 3 \tag{7.1.1}$$

> $vP := [seq(P||i, i = 1..M)]$

$$vP := [P1, P2] \tag{7.1.2}$$

If args = $[a_{1,1}, a_{2,1}], [a_{2,1}, a_{2,2}]$ then $u = a_{1,1} + a_{2,1}, a_{1,2} + a_{2,2}$

> $u := 2, 1;$

$$u := 2, 1 \tag{7.1.3}$$

> $v := makeTab\_mm(u)$

$$v := \left[[2\, k_{1,1}\, k_{1,0}, S_{1,1}\, S_{1,0}], \left[k_{1,0}^2\, k_{0,1}, S_{1,0}^2\, S_{0,1}\right], [k_{2,1}, S_{2,1}], [k_{2,0}\, k_{0,1}, S_{2,0}\, S_{0,1}]\right] \tag{7.1.4}$$

> $u := makeCTR\_m(u)$

$$u := \left[k_{0,0} = 0, k_{0,1} = \mu_{0,1}, k_{1,0} = \mu_{1,0}, k_{1,1} = \mu_{1,1} - \mu_{1,0}\, \mu_{0,1}, k_{2,0} = -\mu_{1,0}^2 + \mu_{2,0}, k_{2,1} = -2\, \mu_{1,1}\, \mu_{1,0} + 2\, \mu_{1,0}^2\, \mu_{0,1} + \mu_{2,1} - \mu_{2,0}\, \mu_{0,1}\right] \tag{7.1.5}$$

> $v := expand(eval(v, u))$

(7.1.6)

$$v := \left[\left[2\,\mu_{1,1}\,\mu_{1,0} - 2\,\mu_{1,0}^{2}\,\mu_{0,1}, S_{1,1}\,S_{1,0}\right], \left[\mu_{1,0}^{2}\,\mu_{0,1}, S_{1,0}^{2}\,S_{0,1}\right], \left[-2\,\mu_{1,1}\,\mu_{1,0} + 2\,\mu_{1,0}^{2}\,\mu_{0,1} + \mu_{2,1} - \mu_{2,0}\,\mu_{0,1}, S_{2,1}\right], \left[-\mu_{1,0}^{2}\,\mu_{0,1} + \mu_{2,0}\,\mu_{0,1}, S_{2,0}\,S_{0,1}\right]\right] \quad (7.1.6)$$

> $A1 := makeTab(1,1);\ A2 := seq\left(\left[v_1, v_2\,(-1)^{nops(v_1)-1}\,(nops(v_1)-1)!\right], v = A1\right)$

$$A1 := [[[P1\ P2], 1], [[P1, P2], 1]]$$

$$A2 := [[[P1\ P2], 1], [[P1, P2], -1]] \quad (7.1.7)$$

> $B1 := makeTab(1);\ B2 := seq\left(\left[v_1, v_2\,(-1)^{nops(v_1)-1}\,(nops(v_1)-1)!\right], v = B1\right)$

$$B1 := [[[P1], 1]]$$

$$B2 := [[P1], 1] \quad (7.1.8)$$

> $vTab := unionVects([A2], [B2])$

$$vTab := [[[P1\ P2, P1], 1], [[P1, P1, P2], -1]] \quad (7.1.9)$$

> $vParts := makeTab(2,1)$

$$vParts := [[[P1\ P2, P1], 2], [[P1, P1, P2], 1], [[P1^2\ P2], 1], [[P1^2, P2], 1]] \quad (7.1.10)$$

Note on function "ricVtab": for example with parameters [P1P2,P1] the function returns 1/2 [where 1 is in vTab e 2 in vParts] and computes fd(3-1, 2) [where 3-1 is N-1 and 2 is order of [P1P2, P1] block.

> $vTab := [seq([x_1, ricVtab(x, vParts, 3)], x = vTab)]$

$$vTab := \left[\left[[P1\ P2, P1], \frac{1}{2}\,n - 1\right], [[P1, P1, P2], -1]\right] \quad (7.1.11)$$

> $vEval := \left[seq\left(mul\left(\mu_{seq\left(degree\left(x, vP_i\right),\ i = 1..nops(vP)\right)}, x = y_1\right) = y_2, y = vTab\right), \mu = 0\right]$

$$vEval := \left[\mu_{1,1}\,\mu_{1,0} = \frac{1}{2}\,n - 1, \mu_{1,0}^{2}\,\mu_{0,1} = -1, \mu = 0\right] \quad (7.1.12)$$

> $\dfrac{add\left(x_1\,x_2, x = eval(v, vEval)\right)}{mul(n - x, x = 0..N-1)}$

$$\frac{n\,S_{1,1}\,S_{1,0} - S_{1,0}^{2}\,S_{0,1} - n\,S_{2,1} + S_{2,0}\,S_{0,1}}{n\,(n-1)\,(n-2)} \quad (7.1.13)$$

Test previous result

> $pm([1,1],[1])$

$$\frac{n\,S_{1,1}\,S_{1,0} - S_{1,0}^{2}\,S_{0,1} - n\,S_{2,1} + S_{2,0}\,S_{0,1}}{n\,(n-1)\,(n-2)} \quad (7.1.14)$$

> $evalb(\% = `\%\%`)$

$$true \quad (7.1.15)$$

Master function "polyk" for manage all cases

This function allows us to recall all functions for generate k-statistics, polykays and their multivariate generalizzations
The input is the following:
- for generate k-statistics k_r the parameter is: [r]
- for generate polykays $k_{r;s}$ the parameter is: [r], [s]
- for generate multivariate k-statistics $k_{r,\,s}$ the parameter is: [r, s]
- for generate multivariate polykays $k_{r,s;\,u,v}$ the parameter is: [r, s], [u, v]

```
> polyk:=proc( )
    if nargs=1 then
      if nops(args[1])=1 then print(KS); ks(op(args[1]))
                         else print(KM); km(op(args[1]))
      end if;
    elif add(`if`(nops(x)=1,0,1),x=args)=0 then
      print(PS); ps(seq(op(x),x=args))
    else print(PM); pm(args)
    end if
  end proc:
```

Example

```
> polyk([3])
```

$$KS$$

$$\frac{2\,S_1^3-3\,n\,S_1\,S_2+n^2\,S_3}{n\,(n-1)\,(n-2)} \tag{8.1}$$

```
> polyk([2],[1])
```

$$PS$$

$$\frac{-S_1^3+(1+n)\,S_1\,S_2-n\,S_3}{n\,(n-1)\,(n-2)} \tag{8.2}$$

```
> polyk([2,1])
```

$$KM$$

$$\frac{-2\,n\,S_{1,\,1}\,S_{1,\,0}+2\,S_{1,\,0}^2\,S_{0,\,1}+n^2\,S_{2,\,1}-n\,S_{2,\,0}\,S_{0,\,1}}{n\,(n-1)\,(n-2)} \tag{8.3}$$

```
> polyk([1,1],[1])
```

$$PM$$

$$\frac{n\,S_{1,\,1}\,S_{1,\,0}-S_{1,\,0}^2\,S_{0,\,1}-n\,S_{2,\,1}+S_{2,\,0}\,S_{0,\,1}}{n\,(n-1)\,(n-2)} \tag{8.4}$$

Replacing symbols with numerical data

Sums of the rth powers of the data points:

```
> powS := proc( )
```

```
    if nargs = 1 then
        Sum('X_i^args[1], i = 1 .. 'n')
    else
        Sum(mul('X_{i,j}^args[j], j = 1 .. nargs), i = 1 .. 'n')
    end if
end proc :
```

Example

```
> powS(5, 3, 1)
```

$$\sum_{i=1}^{n} X_{i,1}^{5} X_{i,2}^{3} X_{i,3} \tag{9.1}$$

```
> powS(9);
```

$$\sum_{i=1}^{n} X_i^9 \tag{9.2}$$

This function allows us to process a k-statistic or polykay replacing the simbols with numerical data. The parameter is the following:

- for generate k-statistics k_r the parameter is: [r], [[n1, n2, ...]]
- for generate polykays $k_{r;s}$ the parameter is: [[r], [s]], [[n1, n2, ...]]
- for generate multivariate k-statistics $k_{r,\,s}$ the parameter is: [[r , s]], [[n1a, n2a], [n1b, n2b] , ...]

- for generate multivariate polykays $k_{r,s;\,u,v}$ the parameter is: [[r , s], [u , v]], [[n1a, n2a], [n1b, n2b] , ...]

```
> npolyk := proc(V, data)
    local res, ind, N, vE, Si;
    Si := false;
    if nops(V) = 1 then
        if nops(op(V)) = 1 then
            Si := true; N := nops(op(data)); res := ks(op(op(V)))
        else
            N := nops(data); res := km(op(op(V)))
        end if;
    elif add(`if`(nops(x) = 1, 0, 1), x = V) = 0 then
        Si := true; N := nops(op(data)); res := ps(seq(op(x), x = V))
    else
        N := nops(data); res := pm(op(V))
    end if;
    ind := `minus`(indets(res), {n});
    vE := seq(y1 = Sum(mul('X'[`if`(Si, op([j, i]), op([i, j]))]^y2[j], j = 1 .. nops(y2)), i = 1
    .. N), y = seq([x, [op(x)]], x = ind));
    eval(res, [eval(evalf(vE), [X = data]), n = N])
```

end proc :

Examples: k-statistics and polykays

```
> data := [[16.34, 10.76, 11.84, 13.55, 15.85, 18.20, 7.51, 10.22, 12.52, 14.68, 16.08, 19.43,
      8.12, 11.20, 12.95, 14.77, 16.83, 19.80, 8.55, 11.58, 12.10, 15.02, 16.83, 16.98, 19.92,
      9.47, 11.68, 13.41, 15.35, 19.11]]
```

$$data := [[16.34, 10.76, 11.84, 13.55, 15.85, 18.20, 7.51, 10.22, 12.52, 14.68, 16.08, 19.43, 8.12, 11.20, 12.95, 14.77, 16.83, 19.80, 8.55, 11.58, 12.10, 15.02, 16.83, 16.98, 19.92, 9.47, 11.68, 13.41, 15.35, 19.11]] \quad (9.3)$$

The estimator for the mean is given by

```
> npolyk([[1]], data)
```

$$14.02166667 \quad (9.4)$$

The estimator for the variance is given by

```
> npolyk([[2]], data)
```

$$12.65006954 \quad (9.5)$$

The estimator for the skewness is given by $k_3 / \sqrt{k_2^3}$

```
> npolyk([[3]], data) / sqrt(npolyk([[2]], data)^3)
```

$$-0.03216240416 \quad (9.6)$$

The estimator for the kurtosis is given by k_4 / k_2^2

```
> npolyk([[4]], data) / npolyk([[2]], data)^2
```

$$-0.8852923202 \quad (9.7)$$

The estimator for the $\kappa_3\ \kappa_2$ is given by

```
> npolyk([[3], [2]], data)
```

$$-15.56090621 \quad (9.8)$$

Examples: multivariate k-ktatistics and multivariate polykays

```
> data := [[5.31, 11.16], [3.26, 3.26], [2.35, 2.35], [8.32, 14.34], [13.48, 49.45], [6.25,
      15.05], [7.01, 7.01], [8.52, 8.52], [0.45, 0.45], [12.08, 12.08], [19.39, 10.42]]
```

$$data := [[5.31, 11.16], [3.26, 3.26], [2.35, 2.35], [8.32, 14.34], [13.48, 49.45], [6.25, 15.05], [7.01, 7.01], [8.52, 8.52], [0.45, 0.45], [12.08, 12.08], [19.39, 10.42]] \quad (9.9)$$

The estimator for the $\kappa_{2,\,1}$ is given by

```
> npolyk([[2, 1]], data)
```

$$-23.73790506 \quad (9.10)$$

The estimator for the $\kappa_{2,\,1}\ \kappa_{1,\,0}$is given by

```
> npolyk([[1, 1], [1]], data)
```

$$294.2657624 \quad (9.11)$$

The estimator for the $\kappa_{2,1}\,\kappa_{2,0}\kappa_{1,0}$ is given by

```
> npolyk( [ [2, 1], [2], [1] ], data )
```

12369.47450 **(9.12)**

```
>
```

Conclusions

Tables 1 and 2 show computational times of three procedures, implementing algorithms to express single and multivariate k-statistics and single and multivariate polykays. The first one, which we call AS algorithms, has been implemented in Mathematica and refers to procedures explained in [7] - availables on the web page http://www.utstat.toronto.edu/david/trans.7.nb. The second one refers to the package MathStatica [8]. Note that in this package, there are no procedures devoted to multivariate polykays. The third procedure, named Fast algorithms has been implemented in Maple 10.x by using the results explained in [5]. The procedure to compute subdivisions of multisets have been described with a wealth of details in [3] and [4].
Above all, comparing our procedures with the more speed ones of MathStatica, it is evident the improvement in computational times. Let us remark that, for all the considered procedures, the results are in the same output form and have been evaluated on the same platform.

All tasks have been performed on a PC Pentium(R)4 Intel(R), CPU 3.00 Ghz, 480MB Ram.

$k_{t,\ldots,l}$	AS Algorithms	MathStatica	Fast-algorithms
k_5	0.06	0.01	0.01
k_7	0.31	0.02	0.01
k_9	1.47	0.04	0.01
k_{11}	14.55	0.16	0.01
k_{14}	396.34	0.61	0.05
k_{16}	58002.60	1.69	0.14
k_{18}	-	5.42	0.17
k_{20}	-	19.11	0.32
k_{22}	-	69.66	0.68
k_{24}	-	285.58	1.33
k_{26}	-	1551.48	2.52
k_{28}	-	6324.28	4.84
$k_{3,2}$	0.06	0.02	0.01
$k_{4,4}$	0.95	0.06	0.01
$k_{5,3}$	0.97	0.08	0.02
$k_{7,5}$	42.30	0.84	0.13
$k_{7,7}$	466.38	2.72	0.27
$k_{9,9}$	-	29.44	2.55
$k_{10,8}$	-	31.61	2.96
$k_{4,4,4}$	40.84	0.67	0.04

Tab 1
Comparison of computational times for k-statistics and polykays. Missed computational times "means greater than 20 houres".

$k_{t_1 \ldots t_r ; l_1 \ldots l_m}$	AS Algorithms	MathStatica	Fast-algorithms
$k_{3\,2}$	0.20	0.02	0.01
$k_{4\,4}$	18.20	0.14	0.02
$k_{5\,5}$	269.19	0.56	0.08
$k_{6\,5}$	1023.33	1.02	0.16
$k_{6\,6}$	-	2.26	0.33
$k_{7\,6}$	-	4.06	0.59
$k_{7\,7}$	-	8.66	1.23
$k_{8\,6}$	-	7.81	1.16
$k_{8\,7}$	-	15.89	2.59
$k_{8\,8}$	-	30.88	5.53
$k_{3\,3\,3}$	1211.05	0.92	0.44
$k_{4\,3\,3}$	-	2.09	0.34
$k_{4\,4\,3}$	-	4.98	1.02
$k_{4\,4\,4}$	-	13.97	2.78
$k_{1\,1;\,1\,1}$	0.05	-	0.01
$k_{2\,1;\,1\,1}$	0.20	-	0.01
$k_{2\,2;\,1\,1}$	1.30	-	0.03
$k_{2\,2;\,2\,1}$	6.50	-	0.06
$k_{2\,2;\,2\,2}$	34.31	-	0.11
$k_{2\,1;\,2\,1;\,2\,1}$	81.13	-	0.17
$k_{2\,2;\,1\,1;\,1\,1}$	30.34	-	0.14
$k_{2\,2;\,2\,1;\,1\,1}$	127.50	-	0.22
$k_{2\,2;\,2\,1;\,2\,1}$	406.78	-	0.47
$k_{2\,2;\,2\,2;\,1\,1}$	467.88	-	0.55
$k_{2\,2;\,2\,2;\,2\,1}$	1402.55	-	1.14
$k_{2\,2;\,2\,2;\,2\,2}$	3787.41	-	2.96

Tab 2
Comparison of computational times for multivariate k-statistics and multivariate polykays. For AS Algorithms, missed computational times means "greater than 20 houres". For MathStatica, missed computational times means "procedures not available".

References

[1] Di Nardo E., G. Guarino, D. Senato (2008) A Maple algorithm for polykays and their generalizations. Adv. Appl. Stat. Vol. 8, No. 1, 19 - 36, http://www.pphmj.com/journals/adas.htm.

[2] Di Nardo E., G. Guarino, D. Senato (2008) An unifying framework for k-statistics, polykays and their generalizations. Bernoulli. Vol. 14(2), 440-468. Official Journal of the Bernoulli Society for Mathematical Statistics and Probability, http://isi.cbs.nl/bernoulli/, (download from http://www.unibas.it/utenti/dinardo/lavori.html)

[3] Di Nardo E., G. Guarino, D. Senato*, Multiset Subdivision,* source Maple algorithm located in www.maplesoft.com *(submitted)*

[4] Di Nardo E., G. Guarino, D. Senato (2008) Symbolic computation of moments of sampling distributions. Comp. Stat. Data Analysis Vol. 52, no. 11, 4909-4922, (download from http://arxiv.org/PS_cache/arxiv/pdf/0806/0806.0129v1.pdf or http://www.unibas.it/utenti/dinardo/lavori.html)

[5] Di Nardo E., G. Guarino, D. Senato (2007), *A new method for fast computing unbiased estimators of cumulants.* In press Statistics and Computing. http://www.springer.com/statistics/computational/journal/11222 (download from http://www.unibas.it/utenti/dinardo/lavori.html)

[6] Di Nardo E., G. Guarino, D. Senato*, A Maple algorithm for k-statistics, polykays and their multivariate generalization,* source Maple algorithm located in www.maplesoft.com *(submitted)*

[7] D. F. Andrews and J. E. Stafford Symbolic computation for statistical inference :. Oxford Statistical Science Series, 21. Oxford University Press, Oxford, 2000.

[8] C. Rose and M. D. Smith, Mathematical Statistics with Mathematica:, Spinger Verlag, New York, 2002.

>