Orthogonal Polynomials in R

```
> ### Orthogonal polynomials
>
> # R generates orthogonal polynomial contrasts using contr.poly and
> # the resulted contrasts are each normalized (by dividing each
> # contrast by its length, i.e., the square root of L_i, see class
> # notes for L_i)
> contr.poly(5) # the 4 contrasts for k=5

                .L         .Q            .C         ^4
[1,] -6.324555e-01  0.5345225 -3.162278e-01  0.1195229
[2,] -3.162278e-01 -0.2672612  6.324555e-01 -0.4780914
[3,] -3.287978e-17 -0.5345225  1.595204e-16  0.7171372
[4,]  3.162278e-01 -0.2672612 -6.324555e-01 -0.4780914
[5,]  6.324555e-01  0.5345225  3.162278e-01  0.1195229
> # To show that it's actually the same as reported in the class
> # notes, each contrast above is multiplied by respective value
> # of the square root of L_i (or equivalently scaled by the reciprocal
> # of the square root of L_i):
> x <- scale(contr.poly(5),scale=1/sqrt(scan()))
1: 10 14 10 70
5:

Read 4 items
> x # displayed the `integer'-orthogonal polynomial contrasts

               .L .Q            .C ^4
[1,] -2.00000e+00  2 -1.000000e+00  1
[2,] -1.00000e+00 -1  2.000000e+00 -4
[3,] -8.31637e-17 -2  5.791043e-16  6
[4,]  1.00000e+00 -1 -2.000000e+00 -4
[5,]  2.00000e+00  2  1.000000e+00  1
attr(,"scaled:center")
           .L            .Q            .C            ^4
-6.581107e-18 -2.220446e-17 -2.360850e-17 -8.326673e-18
attr(,"scaled:scale")
[1] 0.3162278 0.2672612 0.3162278 0.1195229
> zapsmall(x, digits=15) # fuzz print of the values

     .L .Q .C ^4
[1,] -2  2 -1  1
[2,] -1 -1  2 -4
[3,]  0 -2  0  6
[4,]  1 -1 -2 -4
[5,]  2  2  1  1
attr(,"scaled:center")
           .L            .Q            .C            ^4
-6.581107e-18 -2.220446e-17 -2.360850e-17 -8.326673e-18
attr(,"scaled:scale")
[1] 0.3162278 0.2672612 0.3162278 0.1195229
> # Example below shows OPC for k=4
> zapsmall(scale(contr.poly(4),scale=1/sqrt(scan())), digits=15)
1: 20 4 20
4:
```

```
Read 3 items
     .L .Q .C
[1,] -3  1 -1
[2,] -1 -1  3
[3,]  1 -1 -3
[4,]  3  1  1
attr(,"scaled:center")
           .L            .Q            .C
 0.000000e+00  0.000000e+00 -6.938894e-18
attr(,"scaled:scale")
[1] 0.2236068 0.5000000 0.2236068
> # Example below shows OPC for k=6
> zapsmall(scale(contr.poly(6),scale=1/sqrt(scan())), digits=15)
1: 70 84 180 28 252
6:

Read 5 items
     .L .Q .C ^4  ^5
[1,] -5  5 -5  1  -1
[2,] -3 -1  7 -3   5
[3,] -1 -4  4  2 -10
[4,]  1 -4 -4  2  10
[5,]  3 -1 -7 -3  -5
[6,]  5  5  5  1   1
attr(,"scaled:center")
           .L            .Q            .C            ^4            ^5
-4.394633e-17  2.775558e-17 -1.850372e-17  0.000000e+00  1.387779e-17
attr(,"scaled:scale")
[1] 0.11952286 0.10910895 0.07453560 0.18898224 0.06299408
> # Example below shows OPC for k=3
> zapsmall(scale(contr.poly(3),scale=1/sqrt(c(2,6))), digits=15)

     .L .Q
[1,] -1  1
[2,]  0 -2
[3,]  1  1
attr(,"scaled:center")
           .L            .Q
-6.725667e-17  5.551115e-17
attr(,"scaled:scale")
[1] 0.7071068 0.4082483
> # Can customize the calculation with a function
> mypoly <- function(n, L, digits=8){
+ x <- scale(contr.poly(n),scale=1/sqrt(L))
+ attributes(x) <- attributes(x)[-(3:4)]
+ zapsmall(x, digits)
+ }
> mypoly(3, c(2,6)) # now apply the function for k=3

     .L .Q
[1,] -1  1
[2,]  0 -2
[3,]  1  1
> mypoly(7,scan()) # for k=7
1: 28 84 6 154 84 924
7:
```

```
Read 6 items
     .L .Q .C ^4 ^5  ^6
[1,] -3  5 -1  3 -1   1
[2,] -2  0  1 -7  4  -6
[3,] -1 -3  1  1 -5  15
[4,]  0 -4  0  6  0 -20
[5,]  1 -3 -1  1  5  15
[6,]  2  0 -1 -7 -4  -6
[7,]  3  5  1  3  1   1
> mypoly(8, scan()) # for k=8
1: 168 168 264 616 2184 264 3432
8:

Read 7 items
     .L .Q .C  ^4  ^5 ^6  ^7
[1,] -7  7 -7   7  -7  1  -1
[2,] -5  1  5 -13  23 -5   7
[3,] -3 -3  7  -3 -17  9 -21
[4,] -1 -5  3   9 -15 -5  35
[5,]  1 -5 -3   9  15 -5 -35
[6,]  3 -3 -7  -3  17  9  21
[7,]  5  1 -5 -13 -23 -5  -7
[8,]  7  7  7   7   7  1   1
> mypoly(9,scan()) # for k=9
1: 60 2772 990 2002 468 1980 858 12870
9:

Read 8 items
      .L  .Q  .C  ^4  ^5  ^6  ^7  ^8
 [1,] -4  28 -14  14  -4   4  -1   1
 [2,] -3   7   7 -21  11 -17   6  -8
 [3,] -2  -8  13 -11  -4  22 -14  28
 [4,] -1 -17   9   9  -9   1  14 -56
 [5,]  0 -20   0  18   0 -20   0  70
 [6,]  1 -17  -9   9   9   1 -14 -56
 [7,]  2  -8 -13 -11   4  22  14  28
 [8,]  3   7  -7 -21 -11 -17  -6  -8
 [9,]  4  28  14  14   4   4   1   1
```