Pràctica 2: Descomposicions de matrius

Lliurament d'un fitxer `Cognom1Cognom2Nom_pr2.html` fins el dia **8 de març les 23:50**. El document lliurat ha de contenir: els enunciats (mode resumit), el codi R, els resultats i les interpretacions demanades.
Nota: Fixa la llavor `set.seed(1234)`.

```
library(mnormt); library(xtable); library(fBasics)
```

Indicacions: Funció `rk()` de la llibreria `fBasics` per calcular rangs, la funció `solve()` per invertir, `t()` per transposar i l'operador `%*%` per al producte de matrius (recorda que `*` dóna el producte element per element, si són de la mateixa mida). La funció `all.equal()` compara objectes, pot ser convenient l'opció `check.attributes=F`.

1. Genera una mostra Gaussiana multivariant `data` $\sim MN(\mu, \Sigma)$ de mida 2000 *data* $\sim MN(\mu, \Sigma)$ amb

$$\mu = \begin{pmatrix} 5 \\ 1 \\ -1 \\ -4 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 3 & 1.385 & -2.939 & 1 \\ 1.385 & 1 & -1.979 & 1 \\ -2.939 & -1.979 & 8 & -3 \\ 1 & 1 & -3 & 8 \end{pmatrix}$$

   Assigna noms X1, X2, X3 i X4 a les columnes de data i, quan convingui, tracta-la com a matriu o com a data frame. Calcula el rang i la dimensió de `data`.

2. Exercici amb **SD**. Parteix de la matriu $A = \mathbf{X^t X}$, on $\mathbf{X}$ és el fitxer de dades centrades de `data`.

   (a) Calcula la descomposició espectral de A usant la funció `eigen()` i guarda el resultat en un objecte anomenat `sdA`. Guarda els valors propis en una matriu diagonal `Lambda` i els vectors propis en una matriu V. Dóna la classe i la dimensió de Λ i de V. [*Atenció:* El resultat de la funció `eigen()` és una llista d'elements heterogenis; els elements d'una llista es criden amb el `nomllista$nomobjecte`.]

   (b) Comprova l'ortogonalitat de la matriu V i la igualtat del teorema espectral: $A = V\Lambda V^t$.

   (c) Quina és la relació entre la descomposició espectral de A i de S =`cov(data)`? Dedueix teòricament el resultat i comprova-ho en aquest exemple (cal vigilar els signes!).

   (d) Calcula la matriu $B = \mathbf{XX^t}$, i comprova que $B = U\Lambda U^t$, on Λ és la mateixa matriu diagonal de valors propis de A i $U = XV\Lambda^{-\frac{1}{2}}$, on V és la matriu de vectors propis de A. Comprova la igualtat.

   (e) Calcula la matriu Q que factoritza A.

   (f) Calcula totes les aproximacions de rang 2 de A del tipus

$$A^{su} = q_s q_s^t + q_u q_u^t \qquad s, u = 1, \ldots, p, \ s < u,$$

   on q_s és el $s-$èssim vector columna de Q i p el nombre de columnes de A. Quina de les aproximacions assoleix el mínim $\left(\sum_{i,j}(a_{ij} - a_{ij}^{su})^2\right)^{1/2}$?

   (g) *Exercici opcional (sense cap ajuda). Crea una funció que pots anomenar `Arep()` que calculi les aproximacions d'una matriu donada A (simètrica i semi-definida positva) d'un rang fixat, a partir de la seva descomposició espectral. Veure la pàg. següent: cal crear una funció de manera que produeixi els outputs com els que es mostren (llista i noms), i de manera que l'execució de 1000 rèpliques de la funció aplicada a la matriu $A = \mathbf{X^t X}$ trigui un temps similar al que figura al final de la pàg. 3 (temps amb el meu pc !). S'obrirà un lliurament especial per a aquest exercici al campus virtual.

3. Exercici amb **SVD**. Amb la funció `svd()`, calcula la descomposició singular de les dades centrades $\mathbf{X}$, i.e. les matrius $U,\ D,\ V$ tals que $X = UDV^t$.

   (a) Comprova que (llevat de signes, eventualment) les matriu U, V són les matrius que has obtingut en apartats anteriors, i que $D = \Lambda^{1/2}$

   (b) Comprova que: $\mathbf{X} = \sum d_j u_j v_j^t$.

   (c) Dóna la millor aproximació de $\mathbf{X}$ de rang 3, digues-li $\tilde{\mathbf{X}}$. Seguidament, calcula $\left(\sum_{ij}(x_{ij} - \tilde{x}_{ij})^2\right)^{1/2}$ per quantificar l'error d'aquesta aproximació.

OPCIONAL: **Funció per calcular totes les aproximacions de rang r de A.**

```
Arep<-function(A,r){
         # A matriu simètrica i semi-definida positiva
         # r = rang de l'aproximació  ( r< ncol(A))
       sdA<-eigen(A); V<-sdA$vectors; lambda<-sdA$values; Lambda<-diag(l)

                            ...........

       list(Aaprox=matrius,dif=dif,difmin=difmin)
       }

 Arep(A,2)
 Arep(A,3)
```

```
## $Aaprox
## $Aaprox$`12`
##           [,1]       [,2]      [,3]       [,4]
## [1,]  1875.978  1104.4439 -3670.845  1225.1305
## [2,]  1104.444   659.7963 -2173.983   978.1987
## [3,] -3670.845 -2173.9825  7200.209 -2741.9320
## [4,]  1225.130   978.1987 -2741.932  7692.4965
##
## $Aaprox$`13`
##           [,1]       [,2]      [,3]      [,4]
## [1,]  2637.090  1320.1672 -2167.373  2476.806
## [2,]  1320.167   712.6238 -1568.434  1548.456
## [3,] -2167.373 -1568.4345  6298.985 -4918.945
## [4,]  2476.806  1548.4563 -4918.945  4166.475
##
## $Aaprox$`14`
##           [,1]       [,2]      [,3]      [,4]
## [1,]  1479.030   827.6520 -2913.708  2457.847
## [2,]   827.652   771.7113 -1807.073  1534.321
## [3,] -2913.708 -1807.0734  5841.101 -4932.945
## [4,]  2457.847  1534.3210 -4932.945  4166.302
##
## $Aaprox$`23`
##             [,1]       [,2]       [,3]       [,4]
## [1,]  1619.23697  608.33086  -24.64011 -1208.4479
## [2,]   608.33086  232.45233  -93.59034  -555.2925
## [3,]   -24.64011  -93.59034 1819.97271  2203.8689
## [4,] -1208.44795 -555.29252 2203.86886  3526.8060
##
## $Aaprox$`24`
##           [,1]      [,2]      [,3]       [,4]
## [1,]   461.1769  115.8157 -770.9747 -1227.4075
## [2,]   115.8157  291.5399 -332.2292  -569.4278
## [3,]  -770.9747 -332.2292 1362.0885  2189.8690
## [4,] -1227.4075 -569.4278 2189.8690  3526.6329
##
## $Aaprox$`34`
##           [,1]        [,2]      [,3]       [,4]
## [1,] 1222.28861 331.5390574 732.49746 24.2683419
## [2,]  331.53906 344.3673488 273.31881  0.8298069
## [3,]  732.49746 273.3188050 460.86523 12.8561749
## [4,]   24.26834   0.8298069  12.85617  0.6119153
##
##
## $dif
## $dif$`12`
## [1] 1807.991
##
## $dif$`13`
## [1] 5411.016
##
## $dif$`14`
## [1] 5695.345
##
## $dif$`23`
```

```
## [1] 12024.9
##
## $dif$`24`
## [1] 12155.5
##
## $dif$`34`
## [1] 13182.05
##
##
## $difmin
## $difmin$`12`
## [1] 1807.991
```

```
## $Aaprox
## $Aaprox$`123`
##           [,1]       [,2]      [,3]       [,4]
## [1,]  3066.152  1516.4710 -2931.429  1246.7444
## [2,]  1516.471   802.4362 -1918.004   985.6813
## [3,] -2931.429 -1918.0037  7659.583 -2728.5040
## [4,]  1246.744   985.6813 -2728.504  7692.8890
##
## $Aaprox$`124`
##           [,1]       [,2]      [,3]      [,4]
## [1,]  1908.092  1023.9558 -3677.764  1227.785
## [2,]  1023.956   861.5238 -2156.643   971.546
## [3,] -3677.764 -2156.6425  7201.699 -2742.504
## [4,]  1227.785   971.5460 -2742.504  7692.716
##
## $Aaprox$`134`
##           [,1]       [,2]      [,3]      [,4]
## [1,]  2669.204  1239.6792 -2174.292  2479.461
## [2,]  1239.679   914.3512 -1551.095  1541.804
## [3,] -2174.292 -1551.0945  6300.476 -4919.517
## [4,]  2479.461  1541.8036 -4919.517  4166.695
##
## $Aaprox$`234`
##             [,1]       [,2]       [,3]       [,4]
## [1,]  1651.35124  527.84280  -31.55866 -1205.7935
## [2,]   527.84280  434.17978  -76.25037  -561.9453
## [3,]   -31.55866  -76.25037 1821.46320  2203.2970
## [4,] -1205.79355 -561.94525 2203.29701  3527.0254
##
##
## $dif
## $dif$`123`
## [1] 235.5516
##
## $dif$`124`
## [1] 1792.581
##
## $dif$`134`
## [1] 5405.887
##
## $dif$`234`
## [1] 12022.59
##
##
## $difmin
## $difmin$`123`
## [1] 235.5516
```

Temps d'execució.

```
t0<-proc.time()
B<-replicate(10000, Arep(A,2)$difmin)
print(proc.time()-t0)
```

```
##    user  system elapsed
##   10.93    0.01   10.98
```