Anàlisi multivariant: Exercicis 1.1

Vicent Caselles Ballester

2024-03-16

Exercici 1

Carreguem les dades.

```
data(crabs, package = 'MASS')
```

Mirem els tipus de variables que tenim al dataset.

```
str(crabs)
```

```
## 'data.frame':    200 obs. of  8 variables:
##  $ sp   : Factor w/ 2 levels "B","O": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex  : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ index: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ FL   : num  8.1 8.8 9.2 9.6 9.8 10.8 11.1 11.6 11.8 11.8 ...
##  $ RW   : num  6.7 7.7 7.8 7.9 8 9 9.9 9.1 9.6 10.5 ...
##  $ CL   : num  16.1 18.1 19 20.1 20.3 23 23.8 24.5 24.2 25.2 ...
##  $ CW   : num  19 20.8 22.4 23.1 23 26.5 27.1 28.4 27.8 29.3 ...
##  $ BD   : num  7 7.4 7.7 8.2 8.2 9.8 9.8 10.4 9.7 10.3 ...
```

Veiem que tenim 8 variables, una l'índex de les diferents observacions, així que ens dóna poca informació. En podríem prescindir, en la meva opinió. Tenim 5 variables numèriques, dos factors (sexe, l'altra diria que és subespècie – confirmat fent `??crabs`).

Per a trobar les mitjanes, medianes, etc. es pot fer servir la funció `summary`.

```
summary(crabs[unlist(lapply(crabs, is.numeric))])
```

```
##      index            FL              RW              CL              CW
##  Min.   : 1.0   Min.   : 7.20   Min.   : 6.50   Min.   :14.70   Min.   :17.10
##  1st Qu.:13.0   1st Qu.:12.90   1st Qu.:11.00   1st Qu.:27.27   1st Qu.:31.50
##  Median :25.5   Median :15.55   Median :12.80   Median :32.10   Median :36.80
##  Mean   :25.5   Mean   :15.58   Mean   :12.74   Mean   :32.11   Mean   :36.41
##  3rd Qu.:38.0   3rd Qu.:18.05   3rd Qu.:14.30   3rd Qu.:37.23   3rd Qu.:42.00
##  Max.   :50.0   Max.   :23.10   Max.   :20.20   Max.   :47.60   Max.   :54.60
##        BD
##  Min.   : 6.10
##  1st Qu.:11.40
##  Median :13.90
##  Mean   :14.03
##  3rd Qu.:16.60
##  Max.   :21.60
```

Per a dur a terme el resum d'acord als factors, ho faig amb diferents funcions del paquet `dplyr`.

```
cat("Resum per als crancs blaus i femelles")
```

```
## Resum per als crancs blaus i femelles
```

```
require(dplyr, quietly = TRUE)
crabs %>%
  filter(sex == "F",
         sp == "B") %>%
  select(FL, RW, CL, CW, BD) %>% summary()
```

```
##        FL              RW              CL              CW
##  Min.   : 7.20   Min.   : 6.50   Min.   :14.70   Min.   :17.10
##  1st Qu.:11.53   1st Qu.:10.62   1st Qu.:23.93   1st Qu.:27.90
##  Median :13.15   Median :12.20   Median :27.90   Median :32.35
##  Mean   :13.27   Mean   :12.14   Mean   :28.10   Mean   :32.62
##  3rd Qu.:15.28   3rd Qu.:13.88   3rd Qu.:32.75   3rd Qu.:37.83
##  Max.   :19.20   Max.   :16.90   Max.   :40.90   Max.   :47.90
##        BD
##  Min.   : 6.10
##  1st Qu.:10.03
##  Median :11.60
##  Mean   :11.82
##  3rd Qu.:13.88
##  Max.   :18.10
```

```
cat("Resum per als crancs taronja i femelles")
```

```
## Resum per als crancs taronja i femelles
```

```
require(dplyr)
crabs %>%
  filter(sex == "F",
         sp == "O") %>%
  select(FL, RW, CL, CW, BD) %>% summary()
```

```
##        FL              RW              CL              CW
##  Min.   :10.70   Min.   : 9.20   Min.   :21.40   Min.   :24.00
##  1st Qu.:15.60   1st Qu.:13.60   1st Qu.:31.05   1st Qu.:34.88
##  Median :18.00   Median :14.65   Median :34.70   Median :39.55
##  Mean   :17.59   Mean   :14.84   Mean   :34.62   Mean   :39.04
##  3rd Qu.:19.90   3rd Qu.:16.68   3rd Qu.:39.70   3rd Qu.:44.05
##  Max.   :23.10   Max.   :20.20   Max.   :46.20   Max.   :52.50
##        BD
##  Min.   : 9.70
##  1st Qu.:13.80
##  Median :15.65
##  Mean   :15.63
##  3rd Qu.:17.85
##  Max.   :21.10
```

```
cat("Resum per als crancs blaus i mascles")
```

```
## Resum per als crancs blaus i mascles
```

```
require(dplyr)
crabs %>%
  filter(sex == "M",
```

```
        sp == "B") %>%
  select(FL, RW, CL, CW, BD) %>% summary()
```

```
##        FL              RW              CL              CW
##  Min.   : 8.10   Min.   : 6.70   Min.   :16.10   Min.   :19.00
##  1st Qu.:12.65   1st Qu.:10.65   1st Qu.:27.23   1st Qu.:31.52
##  Median :15.10   Median :11.70   Median :32.45   Median :37.10
##  Mean   :14.84   Mean   :11.72   Mean   :32.01   Mean   :36.81
##  3rd Qu.:17.05   3rd Qu.:13.28   3rd Qu.:37.15   3rd Qu.:42.52
##  Max.   :21.30   Max.   :15.80   Max.   :47.10   Max.   :54.60
##        BD
##  Min.   : 7.00
##  1st Qu.:11.00
##  Median :13.60
##  Mean   :13.35
##  3rd Qu.:15.60
##  Max.   :20.00
```

```
cat("Resum per als crancs taronja i mascles")
```

```
## Resum per als crancs taronja i mascles
```

```
require(dplyr)
crabs %>%
  filter(sex == "M",
         sp == "O") %>%
  select(FL, RW, CL, CW, BD) %>% summary()
```

```
##        FL              RW              CL              CW
##  Min.   : 9.10   Min.   : 6.90   Min.   :16.70   Min.   :18.60
##  1st Qu.:14.10   1st Qu.:10.78   1st Qu.:28.75   1st Qu.:31.73
##  Median :16.70   Median :12.10   Median :33.35   Median :36.30
##  Mean   :16.63   Mean   :12.26   Mean   :33.69   Mean   :37.19
##  3rd Qu.:18.80   3rd Qu.:13.70   3rd Qu.:39.02   3rd Qu.:43.08
##  Max.   :23.10   Max.   :16.80   Max.   :47.60   Max.   :52.80
##        BD
##  Min.   : 7.40
##  1st Qu.:12.95
##  Median :15.00
##  Mean   :15.32
##  3rd Qu.:17.77
##  Max.   :21.60
```

No coneixia la funció `fivenum`. Dóna la següent informació: minimum, lower-hinge, median, upper-hinge, maximum; d'acord a la documentació.

```
crabs %>%
  select(FL, RW, CL, CW, BD) %>% lapply(FUN = fivenum)
```

```
## $FL
## [1]  7.20 12.90 15.55 18.10 23.10
##
## $RW
## [1]  6.5 11.0 12.8 14.3 20.2
##
## $CL
```

```
## [1] 14.70 27.25 32.10 37.25 47.60
##
## $CW
## [1] 17.1 31.5 36.8 42.0 54.6
##
## $BD
## [1]  6.1 11.4 13.9 16.6 21.6
```

Utilitzo àlgebra lineal per a respondre aquesta pregunta. Aquesta es la matriu per a les dades generals.

```
# src: https://datascienceplus.com/understanding-the-covariance-matrix/
nums = crabs %>%
  select(FL, RW, CL, CW, BD) %>% scale(scale=F) %>% t()
nums %*% t(nums) / (200 - 1)
```

```
##           FL        RW       CL       CW        BD
## FL 12.217297  8.158045 24.35668 26.55080 11.822581
## RW  8.158045  6.622078 16.35466 18.23964  7.836659
## CL 24.356677 16.354662 50.67992 55.76138 23.971389
## CW 26.550801 18.239640 55.76138 61.96768 26.091867
## BD 11.822581  7.836659 23.97139 26.09187 11.729065
```

A continuació faig els càlculs per a cada una de les combinacions entre els dos factors (que cada un té dos nivells).

```
for (s in levels(crabs$sp)){
  for (x in levels(crabs$sex)){
    cat(paste('Showing results for sex', x, 'and sp\n', s, sep = ' '))
    nums = crabs %>%
      filter(sex == x,
             sp == s) %>%
      select(FL, RW, CL, CW, BD)  %>%
      scale(scale=F) %>% t()
    n = ncol(nums)
    print(nums %*% t(nums) / (n - 1))
  }
}
```

```
## Showing results for sex F and sp
##  B          FL        RW       CL       CW        BD
## FL  6.905408  6.278918 15.47333 17.79808  7.102939
## RW  6.278918  5.947302 14.25319 16.38927  6.555298
## CL 15.473327 14.253188 35.04224 40.21036 16.137927
## CW 17.798082 16.389273 40.21036 46.29084 18.528996
## BD  7.102939  6.555298 16.13793 18.52900  7.576065
## Showing results for sex M and sp
##  B          FL        RW       CL       CW        BD
## FL 10.255955  6.543106 23.29552 26.61182 10.170510
## RW  6.543106  4.459057 15.07893 17.26043  6.548061
## CL 23.295522 15.078927 53.41674 60.98231 23.238878
## CW 26.611816 17.260429 60.98231 69.78092 26.588878
## BD 10.170510  6.548061 23.23888 26.58888 10.239286
## Showing results for sex F and sp
##  O          FL        RW       CL       CW        BD
## FL  8.844657  6.725527 17.17195 19.25430  8.012033
## RW  6.725527  5.515004 13.39709 15.04582  6.211682
## CL 17.171947 13.397094 34.07253 38.07648 15.908188
```

```
## CW 19.254302 15.045820 38.07648 42.80072 17.763110
## BD  8.012033  6.211682 15.90819 17.76311  7.576914
## Showing results for sex M and sp
##  O          FL        RW       CL       CW        BD
## FL 12.355024  7.598151 26.61889 29.31460 12.324465
## RW  7.598151  4.820771 16.53423 18.23423  7.647053
## CL 26.618890 16.534229 57.93047 63.77230 26.763355
## CW 29.314604 18.234229 63.77230 70.34842 29.488253
## BD 12.324465  7.647053 26.76336 29.48825 12.441045
```

Per a calcular el que es demana a l'últim apartat, itero per les dues espècies i guardo els valors que es demanen en un dataframe, el qual utilitzo per a generar la taula final.

```
require(knitr)
```

```
## Loading required package: knitr
```

```
final_df = data.frame(tr = numeric(),
                      tr_p = numeric(),
                      det = numeric(),
                      dets2 = numeric())
for (s in levels(crabs$sp)){
  nums = crabs %>%
      filter(sp == s) %>%
      select(FL, RW, CL, CW, BD)  %>%
      scale(scale=F) %>% t()
    n = ncol(nums)
  sigma = (nums %*% t(nums) / (n - 1))
  trace = sum(diag(sigma))
  tracep = trace/5
  dets = det(sigma)
  dets2 = dets^(1/5)
  row = list(trace, tracep, dets, dets2)
  final_df[nrow(final_df) + 1, ] <- row
}
rownames(final_df) <- levels(crabs$sp)

kable(final_df)
```

	tr	tr_p	det	dets2
B	133.2470	26.64940	0.0770612	0.5989176
O	130.0708	26.01415	0.2457478	0.7552625

Exercici 6

```
source('chap1huswif.dat')

mat <- as.matrix(huswif)

mat = mat %>% scale(scale=F)
(t(mat) %*% mat) / (nrow(mat) - 1)
```

```
##               Hage    Hheight       Wage   Wheight     Hagefm
## Hage     130.23333 -192.18889  128.55556 -436.0000   28.03333
```

```
## Hheight -192.18889 4706.98889   25.88889  876.4444 -229.34444
## Wage     128.55556   25.88889  164.66667 -456.6667   21.66667
## Wheight -436.00000  876.44444 -456.66667 4173.3333   -8.00000
## Hagefm    28.03333 -229.34444   21.66667   -8.0000   29.87778
```

```
dg <- diag((t(mat) %*% mat) / (nrow(mat) - 1))

dgminus1 = sqrt(1 / dg)

d_1half <- matrix(0, ncol(mat), ncol(mat))
diag(d_1half) <- dgminus1

y = mat %*% d_1half
```

M'equivoco, he de repassar els apunts d'algebra lineal.

```
S <- (t(mat) %*% mat) / (nrow(mat) - 1)
S_1half <- 1/sqrt(S) # no, així no és!!!!
```

```
## Warning in sqrt(S): NaNs produced
```

Ara si, $S^{-1/2}$ es calcula tal que així:

```
lambda = eigen(S)$values
v = eigen(S)$vectors
D <- diag(lambda)

d_1half <- diag(sqrt(1/lambda))

S_1half <- v %*% d_1half %*% t(v)
huswif_scaled <- mat %*% S_1half
```

```
mah <- function(x, y, S){
  x <- as.numeric(x)
  y <- as.numeric(y)
  dm <- (x-y) %*% solve(S) %*% (x-y)
  dm <- sqrt(dm)
  dm
}

dm <- matrix(0, dim(huswif)[1], dim(huswif)[1])

for (i in 1:dim(huswif)[1]){
  for (j in 1:dim(huswif)[1]){
    if (i == j){
      dm[i, j] <- 0
    }
    else{
      x <- huswif[i, ]
      y <- huswif[j, ]

      dm[i, j] <- mah(x, y, S)
    }
  }
}
```

Exercici 6 revisat

```
source('chap1huswif.dat')

d <- diag(1/sqrt(diag(cov(huswif))))
xminusxhat <- scale(huswif, scale=F)
y<-xminusxhat %*% d
```

Let's think about this for a sec... S is a covariance matrix, and is therefore symmetric. We can perform diagonalization.

```
S = cov(huswif)
V <- eigen(S)$vectors; D <- diag(eigen(S)$values)
# comprovem que V és ortogonal
t(V)%*%V # t(V) == solve(V)
```

```
##               [,1]          [,2]          [,3]          [,4]          [,5]
## [1,]  1.000000e+00 -1.664576e-16 -2.408253e-16  2.804529e-16  9.147972e-17
## [2,] -1.664576e-16  1.000000e+00 -2.193809e-17  8.078724e-18 -5.077327e-19
## [3,] -2.408253e-16 -2.193809e-17  1.000000e+00 -2.992459e-16 -7.451430e-17
## [4,]  2.804529e-16  8.078724e-18 -2.992459e-16  1.000000e+00  1.633960e-17
## [5,]  9.147972e-17 -5.077327e-19 -7.451430e-17  1.633960e-17  1.000000e+00
```

```
y <- xminusxhat %*% V%*%diag(1/sqrt(diag(D)))%*%t(V)
max(abs(as.matrix(dist(y, diag=T, upper=T)) - dm))
```

```
## [1] 4.352074e-14
```

Exercici 7

```
sxyall <- diag(S)
sx <- sxyall[1]
sy <- sxyall[2]
r <- cor(huswif)[1, 2]

S_1_2 <- S[1:2, 1:2]
sqrt(det(S_1_2)) - unname(sqrt(sx) * sqrt(sy) * sqrt(1-r**2))
```

```
## [1] -2.273737e-13
```

Veiem que, efectivament, són coincidents els resultats.

Exercici 6 re-revisitat

Amb els coneixements que tenim ara de la matriu idempotent H, podem re-resoldre l'exercici 6.

```
n <- nrow(huswif)
I <- diag(1, n, n)
J <- matrix(rep(1, n*n), ncol=n)

H <- I - J/n

X <- as.matrix(huswif)

cov(X)
```

```
##               Hage    Hheight       Wage   Wheight     Hagefm
## Hage     130.23333 -192.18889  128.55556 -436.0000   28.03333
## Hheight -192.18889 4706.98889   25.88889  876.4444 -229.34444
## Wage     128.55556   25.88889  164.66667 -456.6667   21.66667
## Wheight -436.00000  876.44444 -456.66667 4173.3333   -8.00000
## Hagefm    28.03333 -229.34444   21.66667   -8.0000   29.87778
```

```
X_scaled <- H%*%X

sds <- apply(X, 2, sd)
X_scaled %*% diag(1/sds)
```

```
##              [,1]        [,2]       [,3]        [,4]       [,5]
##  [1,]  0.76235662  1.16751086  0.3896433  0.18575466 -0.3475997
##  [2,] -1.34069612  1.63393218 -0.7792865 -0.27863198 -1.4452832
##  [3,] -0.02628816 -1.01883907 -0.6234292  0.65014130  2.0307143
##  [4,]  1.02523821  0.73024088  1.4806444 -0.58822308 -0.1646525
##  [5,]  1.55100139 -1.64559271  1.0910011 -2.44576964  0.5671364
##  [6,] -0.72730574 -0.49411508 -0.8572152  1.26932348 -0.7134942
##  [7,]  0.23659343  0.01603323  1.0910011  0.49534575  1.1159781
##  [8,]  0.58710222  0.16178989  0.3896433  0.03095911 -0.1646525
##  [9,] -0.81493294 -0.63987175 -1.1689298  0.49534575 -0.1646525
## [10,] -1.25306892  0.08891156 -1.0130725  0.18575466 -0.7134942
```

```
X_scaled %*% diag(1/sds)
```

```
##              [,1]        [,2]       [,3]        [,4]       [,5]
##  [1,]  0.76235662  1.16751086  0.3896433  0.18575466 -0.3475997
##  [2,] -1.34069612  1.63393218 -0.7792865 -0.27863198 -1.4452832
##  [3,] -0.02628816 -1.01883907 -0.6234292  0.65014130  2.0307143
##  [4,]  1.02523821  0.73024088  1.4806444 -0.58822308 -0.1646525
##  [5,]  1.55100139 -1.64559271  1.0910011 -2.44576964  0.5671364
##  [6,] -0.72730574 -0.49411508 -0.8572152  1.26932348 -0.7134942
##  [7,]  0.23659343  0.01603323  1.0910011  0.49534575  1.1159781
##  [8,]  0.58710222  0.16178989  0.3896433  0.03095911 -0.1646525
##  [9,] -0.81493294 -0.63987175 -1.1689298  0.49534575 -0.1646525
## [10,] -1.25306892  0.08891156 -1.0130725  0.18575466 -0.7134942
```

```
t(diag(1/sds)) == diag(1/sds)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] TRUE TRUE TRUE TRUE TRUE
## [2,] TRUE TRUE TRUE TRUE TRUE
## [3,] TRUE TRUE TRUE TRUE TRUE
## [4,] TRUE TRUE TRUE TRUE TRUE
## [5,] TRUE TRUE TRUE TRUE TRUE
```

```
t(diag(1/sds) %*% t(X_scaled))
```

```
##              [,1]        [,2]       [,3]        [,4]       [,5]
##  [1,]  0.76235662  1.16751086  0.3896433  0.18575466 -0.3475997
##  [2,] -1.34069612  1.63393218 -0.7792865 -0.27863198 -1.4452832
##  [3,] -0.02628816 -1.01883907 -0.6234292  0.65014130  2.0307143
##  [4,]  1.02523821  0.73024088  1.4806444 -0.58822308 -0.1646525
##  [5,]  1.55100139 -1.64559271  1.0910011 -2.44576964  0.5671364
##  [6,] -0.72730574 -0.49411508 -0.8572152  1.26932348 -0.7134942
##  [7,]  0.23659343  0.01603323  1.0910011  0.49534575  1.1159781
```

```
##  [8,]  0.58710222  0.16178989  0.3896433  0.03095911 -0.1646525
##  [9,] -0.81493294 -0.63987175 -1.1689298  0.49534575 -0.1646525
## [10,] -1.25306892  0.08891156 -1.0130725  0.18575466 -0.7134942
```

```
A <- matrix(c(2,3,1,0), ncol=2)
B <- matrix(c(1,1,0,1), ncol=2)

A%*%B
```

```
##      [,1] [,2]
## [1,]    3    1
## [2,]    3    0
```

```
B%*%A
```

```
##      [,1] [,2]
## [1,]    2    1
## [2,]    5    1
```

```
sum(diag(A%*%B)) == sum(diag(B%*%A))
```

```
## [1] TRUE
```