Prova d'avaluació continuada 2

Vicent Caselles Ballester

2024-06-28

Contents

Exercici 1. Inferència multivariant

Carrego les dades a través del paquet `GGally`.

```
# install.packages('GGally')
require(GGally)
data(flea)
str(flea) # observem les variables que hi han al dataset
```

```
## 'data.frame':    74 obs. of  7 variables:
##  $ species: Factor w/ 3 levels "Concinna","Heikert.",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ tars1  : int  191 185 200 173 171 160 188 186 174 163 ...
##  $ tars2  : int  131 134 137 127 118 118 134 129 131 115 ...
##  $ head   : int  53 50 52 50 49 47 54 51 52 47 ...
##  $ aede1  : int  150 147 144 144 153 140 151 143 144 142 ...
```

```
##  $ aede2  : int  15 13 14 16 13 15 14 14 14 15 ...
##  $ aede3  : int  104 105 102 97 106 99 98 110 116 95 ...
```

A l'enunciat s'estipula que només s'utilitzaran les tres darreres variables, així que em desfaig de les restants.

```
flea <- flea[,
             colnames(flea) %in% c("species", paste("aede", seq(1, 3), sep=""))]
```

a) Normalitat univariant de les tres variables a les tres espècies

Per a dur a terme aquest exercici, em baso en l'exemple 6.8 del document sobre normalitat multivariant proporcionat al campus virtual de l'assignatura. En aquest exemple, per a estudiar la normalitat univariant de múltiples variables es fa servir el test de Shapiro-Wilk. En primer lloc, anem a realitzar els tests per a l'espècie *Chaetocnema concinna*.

```
# install.packages('MVN')
require(MVN)
species <- character(length = nlevels(flea$species))
for (l in 1:nlevels(flea$species)){
  species[l] <- levels(flea$species)[l]
}
species1 <- species[1]
mvn(flea[flea$species==species1, -1], univariateTest = 'SW')$univariateNormality
```

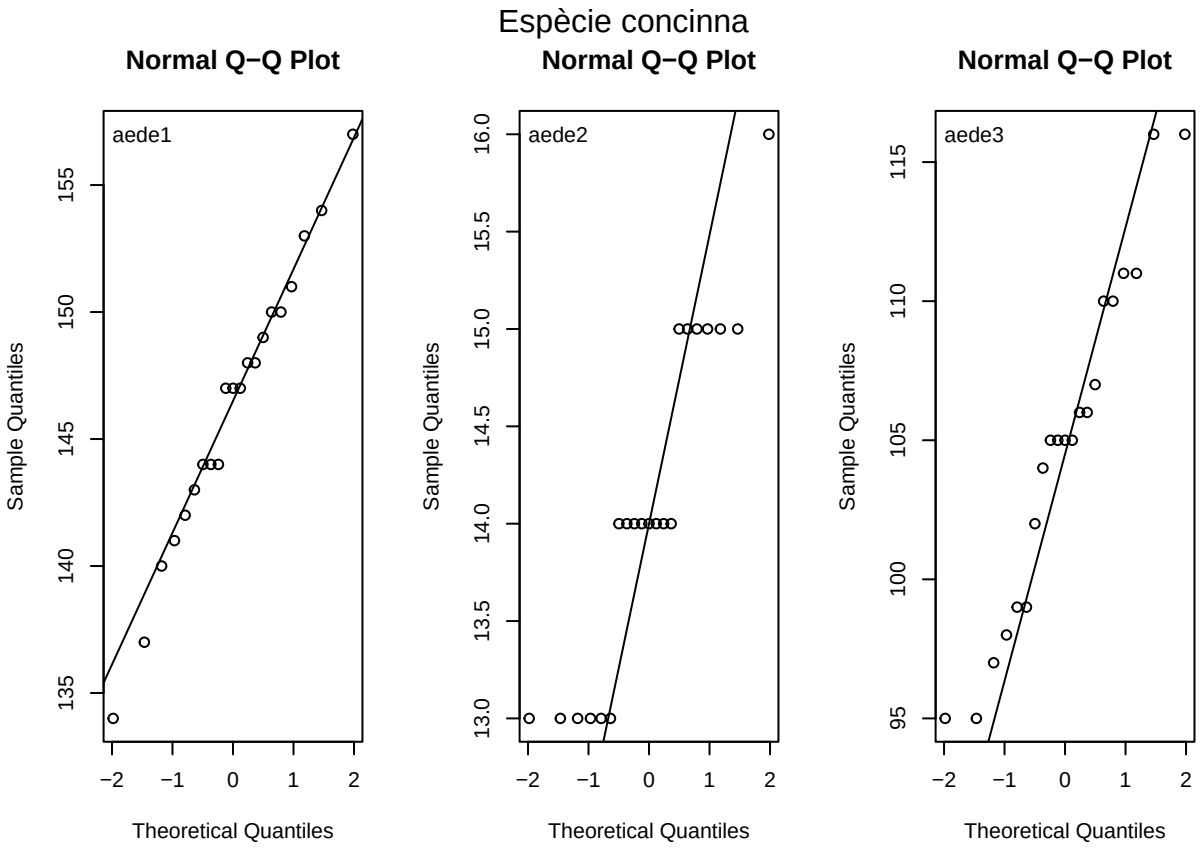
```
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk   aede1      0.9881    0.9936    YES
## 2 Shapiro-Wilk   aede2      0.8669    0.0084    NO
## 3 Shapiro-Wilk   aede3      0.9536    0.3977    YES
```

Veiem que, segons el test de Shapiro-Wilk per a normalitat univariant, la variant `aede2` no segueix una distribució normal. Aquest test estadístic estudia la següent hipòtesi:

$$H_0 : X \sim N(\mu, \sigma^2)$$
$$H_1 : X \nsim N(\mu, \sigma^2)$$

Aquest fet també ho podem observar dibuixant els `qqplots` corresponents per a cada una de les variables.

```
par(mfrow=c(1,3))
for (i in 1:3){
  qqnorm(flea[flea$species==species1, i+1])
  qqline(flea[flea$species==species1, i+1])
  text(
    x = -1.5,
    y = quantile(flea[flea$species==species1, i+1], probs = 0.9999),
    labels = colnames(flea)[i+1]
  )
}
mtext(paste("Espècie",tolower(species1)), side = 3, line = -1.2, outer=TRUE)
```



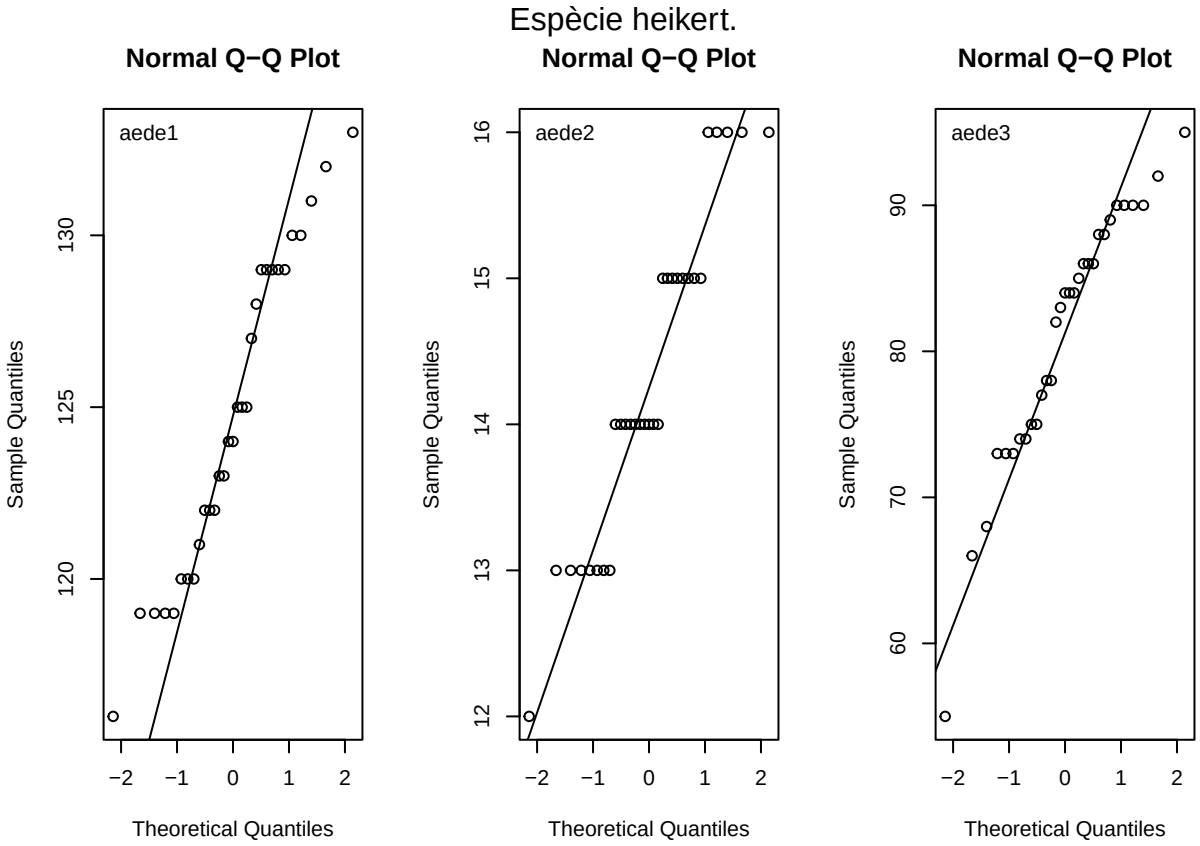
Com podem veure, la variable `aede2`, donat que mesura l'angle entre els *aedeagus* (orgàn reproductor dels artròpodes mascle), només presenta nombres enters, fet que clarament fa que no presenti una distribució normal. Ara anem a realitzar el mateix per a la segona espècie, *Chaetocnema heikertingeri*.

```
species2<- species[2]
mvn(flea[flea$species==species2, -1], univariateTest = 'SW')$univariateNormality
```

```
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk   aede1      0.9494    0.1499    YES
## 2 Shapiro-Wilk   aede2      0.9112    0.0139    NO
## 3 Shapiro-Wilk   aede3      0.9355    0.0622    YES
```

Observem que la variable `aede2` tampoc sembla seguir una distribució normal univariant. Per altra banda, la variable `aede3` presenta un p-valor superior a 0.05, però només lleugerament.

```
par(mfrow=c(1,3))
for (i in 1:3){
  qqnorm(flea[flea$species==species2, i+1])
  qqline(flea[flea$species==species2, i+1])
  text(
    x = -1.5,
    y = quantile(flea[flea$species==species2, i+1], probs = 0.9999),
    labels = colnames(flea)[i+1]
  )
}
mtext(paste("Espècie",tolower(species2)), side = 3, line = -1.2, outer=TRUE)
```

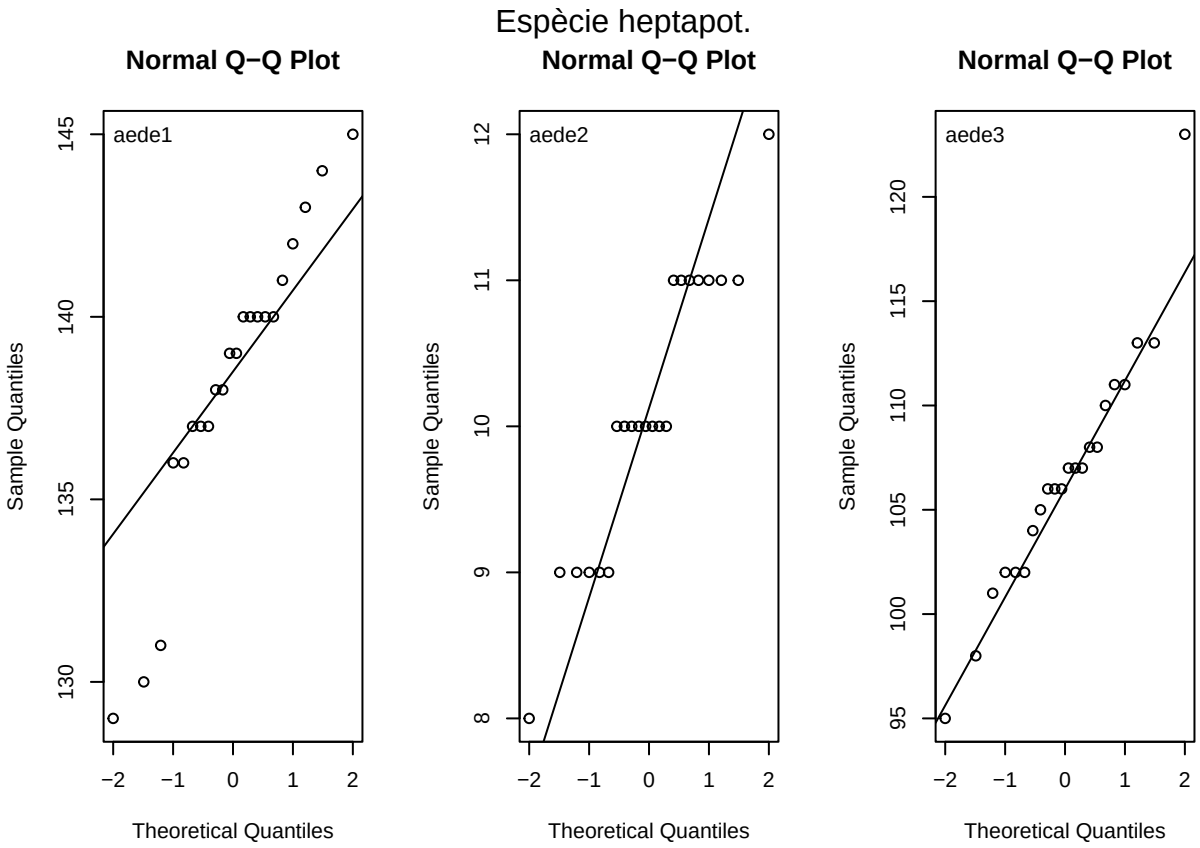


Finalment, la tercera espècie, *Chaetocnema heptapotamica*, presenta els tres p-valors superiors a 0.05. Tot i així, tant `aede1` com `aede2` presenten un p-valor molt proper a aquest tall del 5% de confiança.

```
species3<- species[3]
mvn(flea[flea$species==species3, -1], univariateTest = 'SW')$univariateNormality
```

```
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk   aede1      0.9245    0.0941    YES
## 2 Shapiro-Wilk   aede2      0.9113    0.0504    YES
## 3 Shapiro-Wilk   aede3      0.9523    0.3499    YES
```

```
par(mfrow=c(1,3))
for (i in 1:3){
  qqnorm(flea[flea$species==species3, i+1])
  qqline(flea[flea$species==species3, i+1])
  text(
    x = -1.5,
    y = quantile(flea[flea$species==species3, i+1], probs = 0.9999),
    labels = colnames(flea)[i+1]
  )
}
mtext(paste("Espècie",tolower(species3)), side = 3, line = -1.2, outer=TRUE)
```



En resum, es pot observar que la primera espècie (*Chaetocnema concinna*) presenta la primera (`aede1`) i la tercera variable (`aede3`) aparentment amb normalitat univariant, mentre que la segona (`aede2`) no ho és. La segona espècie (*Chaetocnema heikertingeri*) presenta el mateix patró que *concinna*. Finalment, la tercera espècie (*Chaetocnema heptapotamica*) presenta els tres p-valors del test de Shapiro-Wilk de normalitat univariant superiors a 0.05, encara que la segona variable presenta un p-valor de 0.0504 (quasi significatiu). Així doncs, sembla evident que podem rebutjar la hipòtesi de normalitat multivariant respecte a les dues primeres espècies ja que, si les distribucións marginals no són normals, la multivariant tampoc ho serà.

b) Test de Mardia manual

En el següent *chunk*, realitzo un *for loop* en el qual calculo els estadístics que es demanen per a cada una de les espècies. Primer de tot, calculo la següent matriu, a la qual anomeno `res_`.

$$\text{res\_} = [(\mathbf{x}_i - \bar{\mathbf{x}})' S^{-1} (\mathbf{x}_i - \bar{\mathbf{x}})]$$

Per a dur a terme el càlcul d'aquesta matriu, utilitzo la matriu de covariàncies mostral sesgada, tal i com s'indica a l'apèndix. Un cop calculada `res_`, la utilitzo per a calcular $b_{1,p}$ i $b_{2,p}$, a partir dels quals calculo $\gamma_{1,p}$ i $\gamma_{2,p}$.

```
p <- ncol(flea[, -1]) # number of variables
stopifnot(p == 3)
s <- c()
k <- c()
kmejorada <- c()
for (s_ in species){
  X <- flea[flea$species == s_, -1] # només variables numèriques
  n <- nrow(X)
```

```
  mu_hat <- colMeans(X)  # vector de mitjanes
  S <- cov(X) * (n-1)/n # matriu de covariances sesgada (dividida per n)
  S_inv <- solve(S)

  # càlcul res_
  res_ <- matrix(0, n, n)
  for (i in 1:n){
    for (j in i:n){
      xi <- as.numeric(as.vector(X[i, ])) - mu_hat
      xj <- as.numeric(as.vector(X[j, ])) - mu_hat
      res_[i, j] <- t(xi) %*% S_inv %*% xj
      res_[j, i] <- t(xj) %*% S_inv %*% xi
    }
  }

  # càlcul assimetria
  b_1p <- (1/n**2) * sum(res_**3)
  s <- append(s, n/6 * b_1p)

  # càlcul kurtosi
  b_2p <- (1/n) * sum(diag(res_)**2)
  k_ <- (b_2p - p * (p + 2)) * (sqrt(n/(8 * p * (p + 2))))
  k <- append(k, k_)

  # càlcul kurtosi millorada
  kmejorada_ <- ((b_2p - p*(p+2)*(n-1)*((n+1)**(-1))) / (sqrt(8*p*(p+2)*(n**(-1)))))
  kmejorada <- append(kmejorada, kmejorada_)
}
```

Així doncs, si les dades a partir de les que ha sigut calculada l'assimetria provenen d'una distribució normal multivariant, $\gamma_{1,p} = \frac{n}{6} b_{1,p} \sim \chi^2_f$. Anem a comprovar-ho.

```
f <- p*(p+1)*(p+2) / 6 # degrees of freedom
pchisq(s, df=f, lower.tail=F)
```

```
## [1] 0.9472739 0.8090574 0.3161903
```

Com podem veure, les tres espècies presenten un p-valor superior a 0.05, així que no podem afirmar que cap d'aquestes espècies no segueixin una distribució normal multivariant (**només tenint en compte aquest apartat**).

A continuació, comprovo la kurtosis. En aquest cas, si $\mathbf{X}$ segueix una distribució normal multivariant, $b_{2,p} \sim N(0, 1)$. Així doncs, calculem amb una confiança del 5% si l'estadístic observat pertany a una distribució normal estàndard:

```
2 * pnorm(abs(k), lower.tail = F)
```

```
## [1] 0.2681263 0.2341694 0.7894975
```

```
2 * pnorm(abs(kmejorada), lower.tail = F)
```

```
## [1] 0.5913105 0.4757290 0.4091152
```

Com podem veure, tots els p-valor són superiors a 0.05. D'aquesta manera, donat que tant els estadístics de kurtosi i de assimetria semblen estar d'acord amb les seves corresponents distribucions en cas de que les dades subjacents provinguessin d'una distribució normal multivariant, podem dir que si que ens trobem en aquesta situació (**només tenint en compte aquest apartat**).

Tot i això, tenint en compte els resultats de l'apartat anterior, on a les espècies *Concinna* i *Heikert.* la variable `aede2` no presenta normalitat univariant, llavors no podríem realitzar aquesta afirmació de que hi ha normalitat multivariant, ja que les distribucions marginals de dues de les tres variables no són normals. Finalment, l'espècie *Heptapot.*, com que tots els tests de normalitat univariant de l'apartat anterior han resultat no significatius (per tant no podem rebutjar H_0), si que podríem realitzar aquesta afirmació. Tot i això, em sembla que el p-valor del test de Shapiro-Wilk i el `qqplot` corresponents a la variable `aede2` són massa ajustats per a estar-ne molt convençut.

Gathering further evidence:

De l'apartat 2.2.8 del document de Normal Multivariant distribuït al campus virtual de l'assignatura, tenim que:

$$\text{Si } \mathbf{x} \sim N(\mu, \Sigma), \text{ llavors } (\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu) \sim \chi^2.$$

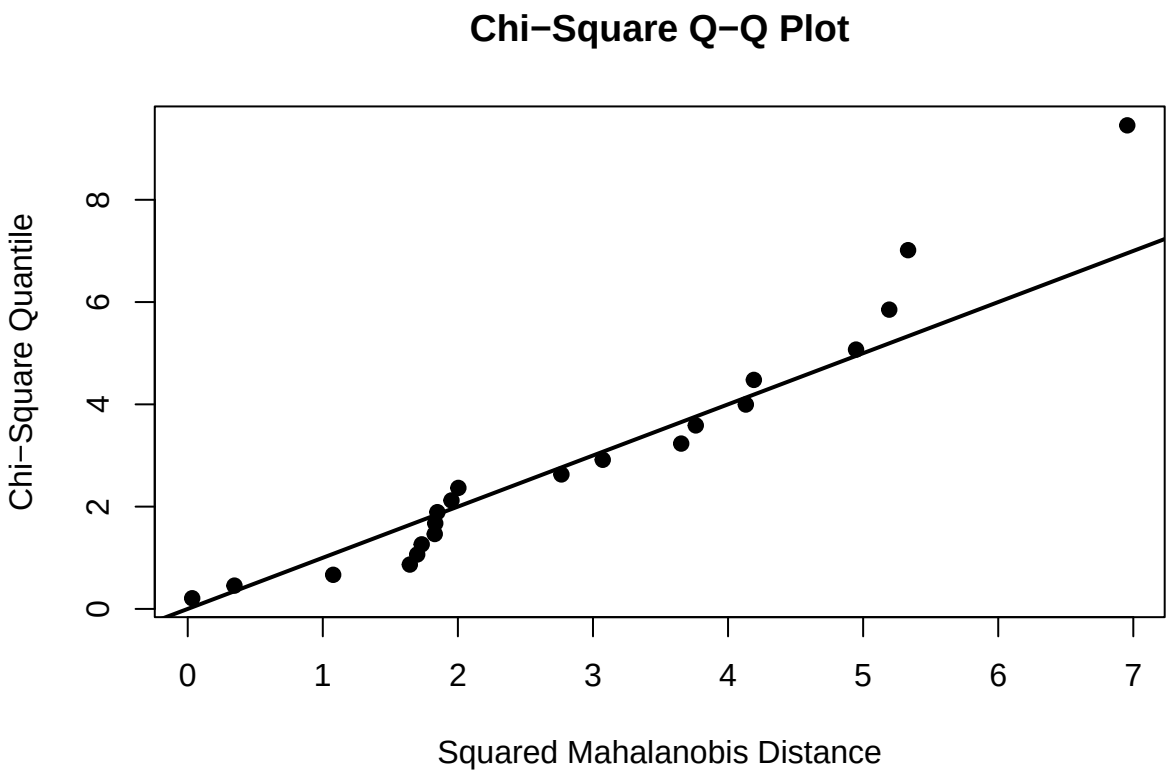
Anem a comprovar-ho. Hem de calcular la distància de mahalanobis (D^2) de cada observació.

```
X <- flea[flea$species==species1, -1]
it <- nrow(X)
muhat <- colMeans(X)
sigma_minus_1 <- solve(cov(X))
ds <- numeric(it)
for (i in 1:it){
  xi <- as.numeric(as.vector(X[i,]))
  ds[i] = t(xi - muhat) %*% sigma_minus_1 %*% (xi - muhat)
}
```

Ara vaig a realitzar el gràfic dels D^2 observats versus els quantils χ^2 corresponents. Copio el codi de la funció `MVN::mvn`.

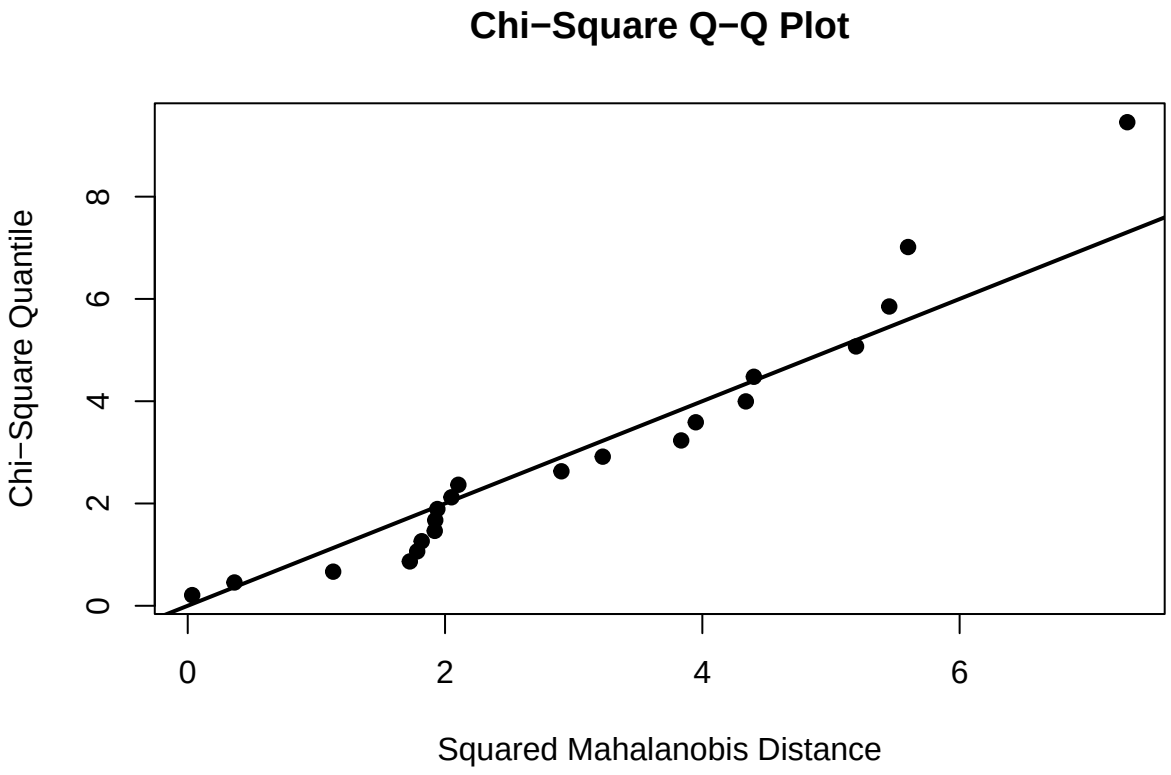
```
r <- rank(ds)
p <- ncol(X)
chi2q <- qchisq((r - 0.5)/it, p)

plot(ds, chi2q, pch = 19, main = "Chi-Square Q-Q Plot",
          xlab = "Squared Mahalanobis Distance", ylab = "Chi-Square Quantile")
abline(0, 1, lwd = 2, col = "black")
```



Comprovo que obtenim el mateix que utilitzant la funció `MVN::mvn` directament, seguint l'exemple 6.9 del document de Normal Multivariant que porto mencionant tota la PAC.

```
mvn(flea[flea$species==species1, -1], multivariatePlot = "qq")
```



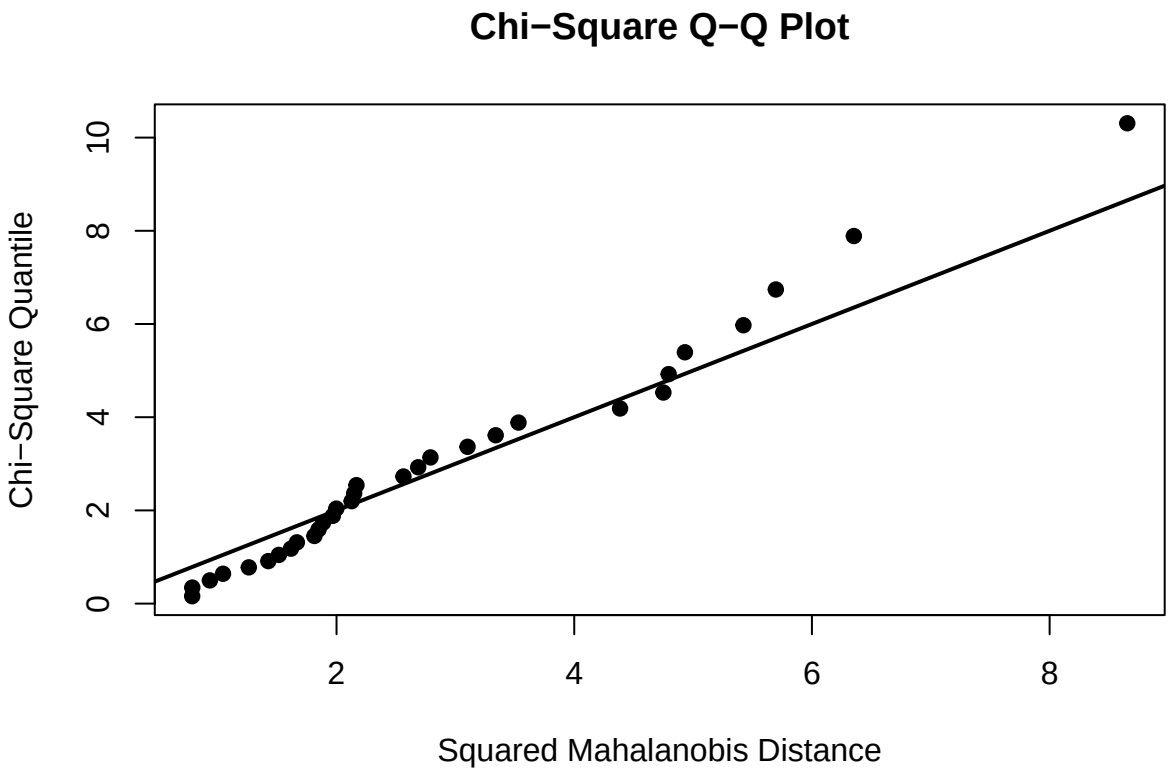
Veiem que obtenim el mateix. Com podem observar, força distàncies s'allunyen del seu quantil esperat, però

no sembla molt greu. L'última distància sembla un *outlier*. Anem a dur a terme el mateix però per a la espècie *Heikert.*.

```
X <- flea[flea$species==species2, -1]
it <- nrow(X)
muhat <- colMeans(X)
sigma_minus_1 <- solve(cov(X))
ds <- numeric(it)
for (i in 1:it){
  xi <- as.numeric(as.vector(X[i,]))
  ds[i] = t(xi - muhat) %*% sigma_minus_1 %*% (xi - muhat)
}
```

```
r <- rank(ds)
p <- ncol(X)
chi2q <- qchisq((r - 0.5)/it, p)

plot(ds, chi2q, pch = 19, main = "Chi-Square Q-Q Plot",
          xlab = "Squared Mahalanobis Distance", ylab = "Chi-Square Quantile")
abline(0, 1, lwd = 2, col = "black")
```



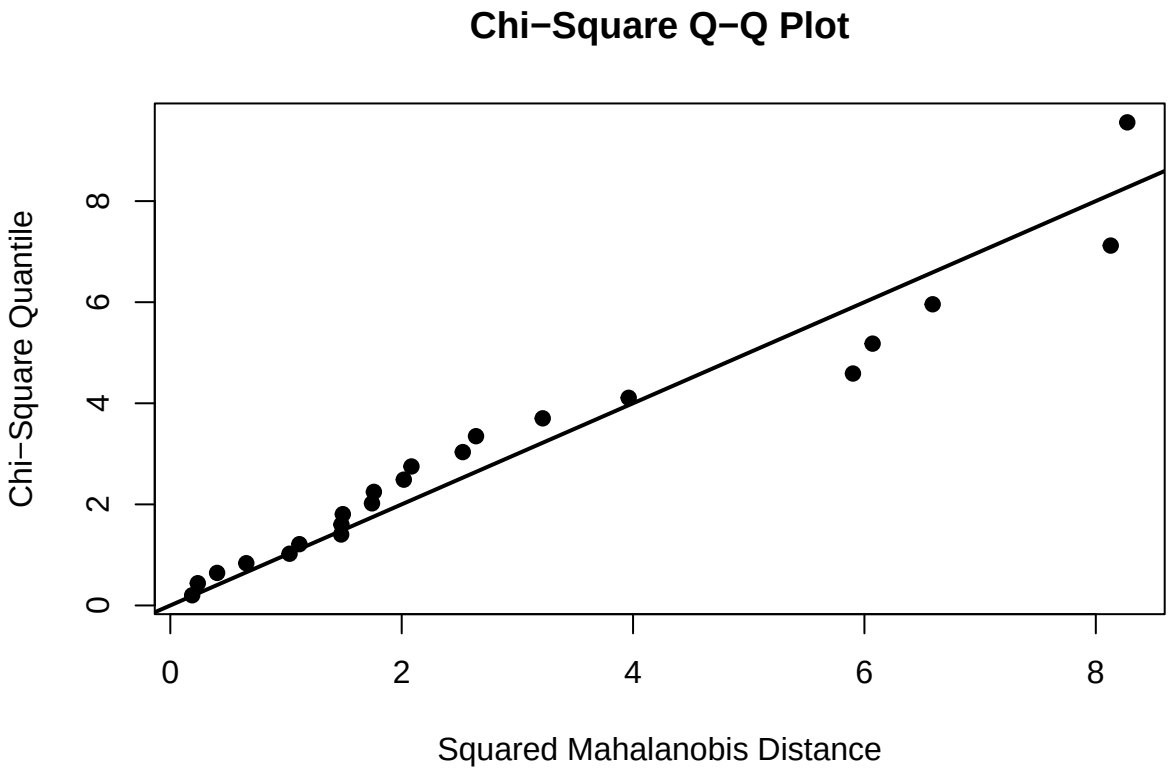
Sembla que aquesta espècie té un millor *fit* a la recta, en quant a les distàncies. Finalment, duc a terme el mateix procediment per a la tercera espècie, *Heptapot.*.

```
X <- flea[flea$species==species3, -1]
it <- nrow(X)
muhat <- colMeans(X)
sigma_minus_1 <- solve(cov(X))
ds <- numeric(it)
for (i in 1:it){
  xi <- as.numeric(as.vector(X[i,]))
```

```
  ds[i] = t(xi - muhat) %*% sigma_minus_1 %*% (xi - muhat)
}
```

```
r <- rank(ds)
p <- ncol(X)
chi2q <- qchisq((r - 0.5)/it, p)

plot(ds, chi2q, pch = 19, main = "Chi-Square Q-Q Plot",
          xlab = "Squared Mahalanobis Distance", ylab = "Chi-Square Quantile")
abline(0, 1, lwd = 2, col = "black")
```



Veiem que el millor ajustament a la recta teòrica correspon a les dues espècies que comencen per H, *Heikert.* i *Heptapod.*.

c) Utilitzant el paquet `MVN`

Investigant sobre el test de Mardia, he trobat que no només el paquet `MVN` l'implementa, sino també el paquet `mvnormalTest`. La diferència és que el primer implementa la versió *no millorada* de la kurtosi, mentre que el segon implementa la versió *milorada* (MK).

```
mvn(flea[flea$species==species1, -1], mvnTest='mardia')$multivariateNormality
```

```
##              Test         Statistic           p value Result
## 1 Mardia Skewness  4.00161895575667 0.947273919151317    YES
## 2 Mardia Kurtosis -1.10738786545985 0.268126270031477    YES
## 3             MVN              <NA>              <NA>    YES
```

```
c(assym=s[1], pval=pchisq(s, df=f, lower.tail=F)[1]) # assimetria espècie Concinna
```

```
##     assym      pval
## 4.0016190 0.9472739
```

```r
c(kurt=k[1], pval=2 * pnorm(abs(k), lower.tail = F)[1]) # kurtosi espècie Concinna
```

```
##       kurt       pval
## -1.1073879  0.2681263
```

Veiem que la assimetria i la kurtosi *no millorada* coincideixen. Anem a comprovar ara la kurtosi *millorada*.

```r
mvnormalTest::mardia(X=flea[flea$species==species1, -1])$mv.test
```

```
##           Test Statistic p-value Result
## 1     Skewness    4.0016  0.9473    YES
## 2     Kurtosis   -0.5369  0.5913    YES
## 3 MV Normality      <NA>    <NA>    YES
```

```r
c(kurt_mej=kmejorada[1],  pval=2 * pnorm(abs(kmejorada), lower.tail = F)[1]) # espècie Concinna
```

```
##   kurt_mej       pval
## -0.5369378  0.5913105
```

Vaig a repetir el mateix per a les altres espècies.

```r
mvn(flea[flea$species==species2, -1], mvnTest='mardia')$multivariateNormality
```

```
##              Test        Statistic           p value Result
## 1 Mardia Skewness 6.07342106102113 0.809057445916978    YES
## 2 Mardia Kurtosis -1.1896871377758 0.234169382618609    YES
## 3             MVN             <NA>              <NA>    YES
```

```r
c(assym=s[2], pval=pchisq(s, df=f, lower.tail=F)[2]) # assimetria espècie Heikert.
```

```
##     assym      pval
## 6.0734211 0.8090574
```

```r
c(kurt=k[2], pval=2*pnorm(abs(k), lower.tail = F)[2]) # kurtosi espècie Heikert.
```

```
##       kurt       pval
## -1.1896871  0.2341694
```

```r
mvnormalTest::mardia(X=flea[flea$species==species2, -1])$mv.test
```

```
##           Test Statistic p-value Result
## 1     Skewness    6.0734  0.8091    YES
## 2     Kurtosis   -0.7132  0.4757    YES
## 3 MV Normality      <NA>    <NA>    YES
```

```r
c(kurt_mej=kmejorada[2],  pval=2*pnorm(abs(kmejorada), lower.tail = F)[2]) # espècie Heikert.
```

```
##   kurt_mej       pval
## -0.7131887  0.4757290
```

Per a l'espècie *Heikert.* també coincideix, estic content. Falta l'última espècie.

```r
mvn(flea[flea$species==species3, -1], mvnTest='mardia')$multivariateNormality
```

```
##              Test        Statistic           p value Result
## 1 Mardia Skewness  11.551540302658 0.316190290178933    YES
## 2 Mardia Kurtosis 0.26696325672142 0.789497451542429    YES
## 3             MVN             <NA>              <NA>    YES
```

```r
c(assym=s[3], pval=pchisq(s, df=f, lower.tail=F)[3]) # assimetria espècie Heptapot.
```

```
##      assym       pval
```

```
## 11.5515403  0.3161903
```

```
c(kurt=k[3], pval=2*pnorm(abs(k), lower.tail = F)[3]) # kurtosi espècie Heptapot.
```

```
##      kurt      pval
## 0.2669633 0.7894975
```

```
mvnormalTest::mardia(X=flea[flea$species==species3, -1])$mv.test
```

```
##           Test Statistic p-value Result
## 1     Skewness   11.5515  0.3162    YES
## 2     Kurtosis    0.8255  0.4091    YES
## 3 MV Normality      <NA>    <NA>    YES
```

```
c(kurt_mej=kmejorada[3],  pval=2*pnorm(abs(kmejorada), lower.tail = F)[3]) # espècie Heptapod.
```

```
##  kurt_mej      pval
## 0.8254516 0.4091152
```

Efectivament, tots els resultats coincideixen en les tres espècies, tant en el cas de l'assimetria, com en el de la kurtosi *millorada* i la *no millorada*.

d) Evaluant `psych::mardia`

Recalculo els resultats de l'apartat b) amb la matriu de covariances insesgada.

```
p <- ncol(flea[, -1])
stopifnot(p == 3)
s <- c()
k <- c()
kmejorada <- c()
for (s_ in species){
  X <- flea[flea$species == s_, -1]
  n <- nrow(X)

  mu_hat <- colMeans(X)
  S <- cov(X) # unbiased
  S_inv <- solve(S)

  res_ <- matrix(0, n, n)
  for (i in 1:(n)){
    for (j in (i):n){
      xi <- as.numeric(as.vector(X[i, ])) - mu_hat
      xj <- as.numeric(as.vector(X[j, ])) - mu_hat
      res_[i, j] <- t(xi) %*% S_inv %*% xj
      res_[j, i] <- t(xj) %*% S_inv %*% xi
    }
  }

  # skewness
  b_1p <- (1/n**2) * sum(res_**3)
  s <- append(s, n/6 * b_1p)

  # kurtosis
  b_2p <- (1/n) * sum(diag(res_)**2)
  k_ <- (b_2p - p * (p + 2)) * (sqrt(n/(8 * p * (p + 2))))
  k <- append(k, k_)
```

```
  # kurtosis millorada
  kmejorada_ <- ((b_2p - p*(p+2)*(n-1)*((n+1)**(-1))) / (sqrt(8*p*(p+2)*(n**(-1)))))
  kmejorada <- append(kmejorada, kmejorada_)
}
```

Ara amb la funció `mardia` del paquet `psych`.

```
c(true=psych::mardia(flea[flea$species==species1, -1], plot=F)$skew, mine=s[1])
```

```
##     true     mine
## 3.456749 3.456749
```

Veiem que coincideix la *skewness* entre ambdós mètodes per a la espècie *Concinna*.

En el cas de la espècie *Heikert.*, ho realitzo a continuació.

```
c(true=psych::mardia(flea[flea$species==species2, -1], plot=F)$skew, mine=s[2])
```

```
##     true     mine
## 5.504426 5.504426
```

Finalment, realitzem el mateix per a la tercera i última espècie.

```
c(true=psych::mardia(flea[flea$species==species3, -1], plot=F)$skew, mine=s[3])
```

```
##     true     mine
## 10.04685 10.04685
```

Veiem que és així el que diu l'enunciat. No mostro els resultats complerts de cada test ja que són els mateixos que abans, cap p-valor dona significatiu (< 0.05).

e) Comparació de matrius de covariances

Per a dur a terme aquest exercici, em baso en l'exercici 18 del tema R5. Primer de tot, calculem els diferents nombres d'observacions de cada grup.

```
p <- 3 # número de variables
n1 <- nrow(flea[flea$species==species1, ])
n2 <- nrow(flea[flea$species==species2, ])
n3 <- nrow(flea[flea$species==species3, ])
n <- nrow(flea)

stopifnot((n1+n2+n3) == n)
```

Ara calculo les matrius de variança-covariança insesgades, a partir de les quals calculo les sesgades (de màxima verisimilitud), necessàries per a realitzar el test de raó de verosimilituds.

```
# càlcul S insesgada; càlcul S sesgada
S1 <- cov(flea[flea$species == species1, -1]); S1 <- (n1-1)*S1/n1
S2 <- cov(flea[flea$species == species2, -1]); S2 <- (n2-1)*S2/n2
S3 <- cov(flea[flea$species == species3, -1]); S3 <- (n3-1)*S3/n3
```

Ara calculem la matriu de covariances comú.

```
# matriu cov comú
S <- (n1*S1 + n2*S2 + n3*S3) / n
```

Ara calculem el test de *Bartlett* multivariant, que modela:

$$n \log |\mathbf{S}| - n_1 \log |\mathbf{S}_1| - n_2 \log \ |\mathbf{S}_2| - n_3 \log |\mathbf{S}_3| \sim \chi^2$$

```r
llr <- n * log(det(S)) - n1 * log(det(S1)) - n2 * log(det(S2)) - n3 * log(det(S3))
pchisq(llr, df=(3-1)*p*(p+1)/2, lower.tail=F)
```

```
## [1] 0.3093875
```

Com podem veure, el p-valor és superior a 0.05, així que no podem rebutjar la hipòtesi nula (homogeneitat de matriu de covariances amb el factor espècie). Utilitzant el test M de Box:

```r
heplots::boxM(flea[,-1], flea$species)
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  flea[, -1]
## Chi-Sq (approx.) = 12.177, df = 12, p-value = 0.4316
```

Aquest test reforça les nostres conclusions extretes a partir del test de raó de verosimilituds.

f) MANOVA d'un factor

Per a poder realitzar un anàlisi de variança multivariant robust, s'han de complir les següents condicions:

- Les observacions han de ser independents
- Les dades han de provenir d'una distribució amb matriu de covariances $\mathbf{\Sigma}$ comú.
- Les dades segueixen una distribució normal multivariant.

La assumpció amb la que més dubtes tinc és la final, ja que hem vist que no en tots els grups (espècies) es compleix que les variables provinguin d'una distribució normal multivariant. Tot i això, vaig a observar els resultats.

```r
Y <- as.matrix(flea[, -1])
g <- manova(cbind(aede1, aede2, aede3) ~ species, data=flea)

summary(g)
```

```
##           Df Pillai approx F num Df den Df    Pr(>F)
## species    2 1.6043   94.595      6    140 < 2.2e-16 ***
## Residuals 71
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Així doncs, veiem que amb aquest anàlisi d'anova multivariant hi ha alguna espècie amb un vector de mitjanes μ_i diferent a les altres, i.e. $\exists\, i, j \colon \mu_i \neq \mu_j$, on i i j indiquen les espècies.

Aquest apartat també es podria dur a terme amb el test de Wilks. Vaig a realitzar-lo manualment i a comprovar els resultats, tal i com es fa a l'apartat c) de l'exercici 16 dels exercicis d'Inferència Multivariant. Per a això necessitaré una sèrie de funcions. Aquestes van ser escrites durant l'estudi de l'apartat `R7` de l'assignatura, i les he adaptat per a poder-les utilitzar per a aquest apartat i a l'apartat 3d) de la present PAC. Calculen les matrius de *sums of squares* i *cross-products* intra-grups (*within*, `W`) i total (a partir de les quals es pot obtenir la inter-groups (*between*, `B`)).

```r
## Funció que separa el dataframe
separate_df <- function(df, f){
  n <- nlevels(f)
  l <- levels(f)
  splitted <- list()
  for (i in 1:n){
    splitted[[l[i]]] <- data.matrix(df[f == l[i], ])
```

```
  }
  splitted
}

total_variance_helper <- function(x, colmeans){
  sweep(x, 2, STATS=colmeans, FUN = "-")
}

within_variance_helper <- function(x){
  colmeans <- colMeans(x)
  sweep(x, 2, STATS=colmeans, FUN='-')
}

sum_of_squares <- function(df, f, mode = 'total', sum_ = TRUE){
  mode <- tolower(mode)
  f <- as.factor(f)
  separated <- separate_df(df, f)
  if (mode == 'total'){
    substracted <- lapply(separated, FUN = total_variance_helper,
                          colmeans=colMeans(df))
  }
  else if (mode == 'within'){
    substracted <- lapply(separated, FUN = within_variance_helper)
  }
  else if (mode == 'all'){
    if (!sum_) stop('Not implemented') # fa mandra
    else{
      W = sum_of_squares(df, f, mode='within', sum_=sum_)
      T_ = sum_of_squares(df, f, mode='total', sum_=sum_)
      return(list(
          W = W,
          T_ = T_,
          B = T_ - W
          ))
    }
  }
  else {stop(paste('Non-implemented mode:', mode))}

  k <- nlevels(f)
  res <- if (sum_) matrix(0, ncol(df), ncol(df)) else list()
  for (j in 1:k){
    tmp <- substracted[[j]]
    if (sum_){
      res <- res + t(tmp)%*%tmp
    }
    else res[[j]] <- t(tmp)%*%tmp
  }
  res
}
```

Ara, utilitzant aquestes funcions, calculo les matrius de *sums of squares* i *cross-products* requerides, a partir de les quals calculo la lambda de Wilks (Λ) i l'estadístic de Fisher.

```r
tmp <- sum_of_squares(df=Y, f=flea$species, mode = 'all')
W_ <- tmp$W; T_ <- tmp$T_; B_ <- tmp$B
```

```r
# source: https://online.stat.psu.edu/stat505/lesson/8/8.3
lambda <- det(W_) / det(W_+B_)
N <- nrow(flea)
g <- nlevels(flea$species)
a <- N - g  - (p-g+2)*1/2
b <- ((p**2)*((g-1)**2) - 4)/((p**2) + ((g-1)**2) - 5)
b <- sqrt(b)
c <- (p*(g-1) - 2) * 1/2
df1 <- p * (g-1)
df2 <- a*b - c
F_ <- ((1 - (lambda**(1/b)))/(lambda**(1/b))) * ((a*b - c)/(p * (g-1)))
p.val <- pf(F_,df1,df2,lower.tail=FALSE)

c(lambda_wilks=lambda, F_stat=F_, pval=p.val)
```

```
## lambda_wilks       F_stat         pval
## 3.641218e-02 9.753259e+01 3.778806e-47
```

```r
maov1 <- manova(cbind(aede1, aede2, aede3) ~ species, data=flea)
summary(maov1, test="Wilks")
```

```
##           Df    Wilks approx F num Df den Df    Pr(>F)
## species    2 0.036412   97.533      6    138 < 2.2e-16 ***
## Residuals 71
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Així doncs, ens dóna el mateix. En resum, hi ha diferències entre les espècies per a les variables `aede1`, `aede2`, `aede3`.

g) *Pairwise comparisons*

Responent a la pregunta de si és necessari fer ajustaments per a comparacions múltiples, jo diria que si, ja que estem intentant fer inferència múltiples vegades, fet que pot inflar l'error de tipus I (rebutjar alguna H_0 sent aquesta certa).

Per a dur a terme les *pairwise comparisons*, he fet una mica de recerca a internet i he trobat una funció del paquet `biotools` que permet dur a terme aquest anàlisi de manera fàcil, amb ajustament de p-valors implementat.

```r
g <- manova(cbind(aede1, aede2, aede3) ~ species, data=flea)
biotools::mvpaircomp(g, factor1='species', test="Pillai", adjust='bonferroni')
```

```
##
##             Multivariate Pairwise Comparisons
##
##                         Pillai approx F num DF den DF    Pr(>F)
## Concinna - Heikert.  0.80657   95.904      3     69 < 2.2e-16 ***
## Concinna - Heptapot. 0.76396   74.442      3     69 < 2.2e-16 ***
## Heikert. - Heptapot. 0.84995  130.282      3     69 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## With bonferroni p-value adjustment for multiple comparisons
```

A més a més, per a poder prendre conclusions més segures, vaig a fer comparacions *two-vs-two* amb l'estadístic T^2 de Hotelling, que modela les distàncies mahalanobis entre les mitjanes mostrals i la covariança comú amb aquesta distribució. Primer de tot, comparo les mitjanes de l'espècie *Concinna* i *Heikert.*.

```
# separo les dades
X_c <- flea[flea$species==species1, -1]
X_hei <- flea[flea$species==species2, -1]
n_c <- nrow(X_c)
n_hei <- nrow(X_hei)
p <- ncol(X_hei); stopifnot(p == ncol(X_c))

# vectors de mitjanes
mu_hat_c <- colMeans(X_c)
mu_hat_hei <- colMeans(X_hei)
S_c <- cov(X_c) # covariances
S_hei <- cov(X_hei) # covariances
S_chei <- ((n_c-1) * S_c + (n_hei - 1)*S_hei)/(n_c+n_hei-2) # cov comú

# diferències de les mitjanes (dist mahalanobis)
D2_chei <- t(mu_hat_c - mu_hat_hei) %*% solve(S_chei) %*% (mu_hat_c - mu_hat_hei)

T2_chei <- n_c*n_hei/(n_c+n_hei) * as.numeric(D2_chei) # hotelling stat
F_chei <- (n_c + n_hei - p - 1) / ((n_c + n_hei - 2)*p) * T2_chei # fstat
p.val_chei <- pf(F_chei, p,  # guardo el pvalor
                 n_c+n_hei-p-1,
                 lower.tail=FALSE)
```

En segon lloc, calculo *Concinna* i *Heptapot.*

```
# idem
X_hepta <- flea[flea$species == species3, -1]; stopifnot(p == ncol(X_hepta))
n_hepta <- nrow(X_hepta)
mu_hat_hepta <- colMeans(X_hepta)
S_hepta <- cov(X_hepta)
S_chepta <- ((n_c - 1)*S_c + (n_hepta - 1)*S_hepta)/(n_c + n_hepta - 2)

# diferències de les mitjanes (dist mahalanobis)
D2_chepta <- t(mu_hat_hepta - mu_hat_c) %*% solve(S_chepta) %*% (mu_hat_hepta-mu_hat_c)
T2_chepta <- n_c*n_hepta/(n_c+n_hepta) * as.numeric(D2_chepta) # hotelling stat
F_chepta <- (n_c + n_hepta - p - 1) / ((n_c + n_hepta - 2) * p) * T2_chepta # fstat
p.val_chepta <- pf(F_chepta, p, n_c + n_hei - p - 1, lower.tail=FALSE)
```

Lastly, entre *Heikert.* i *Heptapot.*.

```
S_heptahei <- ((n_hepta-1)*S_hepta + (n_hei-1)*S_hei)/(n_hepta + n_hei - 2)

# mahalanobis distances between sample means
D2_heptahei <- t(mu_hat_hepta - mu_hat_hei) %*% solve(S_heptahei) %*% (mu_hat_hepta-mu_hat_hei)
T2_heptahei <- n_hepta*n_hei/(n_hepta+n_hei) * as.numeric(D2_heptahei) # hotelling stat
F_heptahei <- (n_hepta+n_hei - p - 1) / ((n_hepta + n_hei - 2)*p)*T2_heptahei # f stat
p.val_heptahei <- pf(F_chepta, p, n_hepta+n_hei-p-1, lower.tail=FALSE)
```

Realitzo ajustament per a comparacions múltiples. S'utilitza l'ajustament de Bonferroni (*default*), que bàsicament múltiplica `n * p`.

```
p.adjust(c(p.val_chei, p.val_chepta, p.val_heptahei))
```

```
## [1] 1.072275e-18 1.072275e-18 1.072275e-18
```

Comprovo que ho hagi fet bé amb el la funció `hotelling.test` del paquet `Hotelling`.

```
t=Hotelling::hotelling.test(x=as.matrix(X_c), y=as.matrix(X_hepta))
stopifnot(abs(t$stats[1]$statistic - T2_chepta) < 1e-10)

t=Hotelling::hotelling.test(x=as.matrix(X_hei), y=as.matrix(X_hepta))
stopifnot(abs(t$stats[1]$statistic - T2_heptahei) < 1e-10)

t=Hotelling::hotelling.test(x=as.matrix(X_hei), y=as.matrix(X_c))
stopifnot(abs(t$stats[1]$statistic - T2_chei) < 1e-10)
```

Veiem que els resultats son gairebé els mateixos, en quant a l'estadístic. En definitiva, hi han diferències entre els vectors de mitjanes de totes les espècies.

Exercici 2. Anàlisi discriminant

a) Reducció del conjunt de dades i gràfic `pairs`

Recarrego el conjunt de dades i faig el *subset* tal i com es demana.

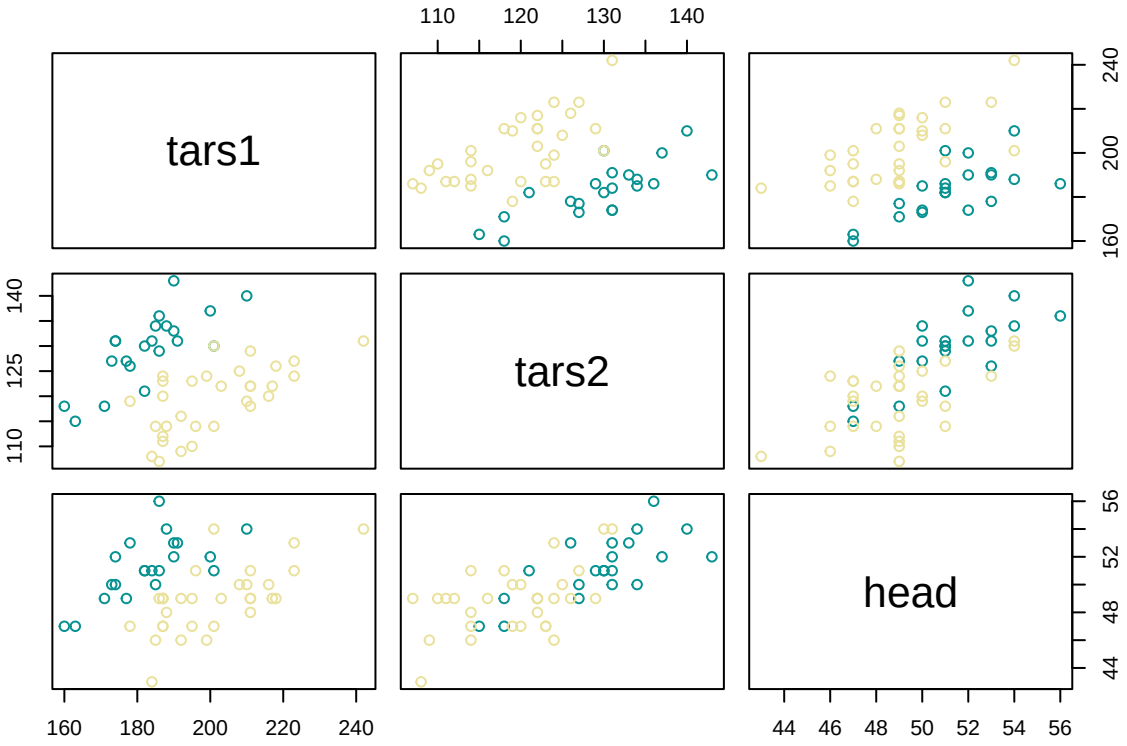
```
data(flea, package='GGally')
flea <- flea[(flea$species == species1 | flea$species == species2), 1:4]

# agafem els 10 primers individus de cada espècie Concinna i Heikert. per al conjunt training
w <- which(flea$species == "Concinna")[1]
w2 <- which(flea$species=="Heikert.")[1]
training <- flea[c(w:(w+9), w2:(w2+9)), ]

# la resta serà test
test <- flea[c((w+10):(w2-1), (w2+10):nrow(flea)) , ]
stopifnot(nrow(flea) == nrow(test) + nrow(training)) # check
```

Ara realitzo el gràfic amb `pairs` tal i com es demana.

```
pairs(flea[,-1], col = hcl.colors(3, "Temps")[flea$species])
```



Veiem que hi ha una certa correlació positiva entre les variables. Això té sentit, ja que les tres variables són mesures de tamany, les dues primeres referents a les cames i la tercera al cap. Respecte al gràfic, sembla que ha de ser força *fàcil* separar entre les dues classes (cian: *Concinna*; verd: *Heikert.*).

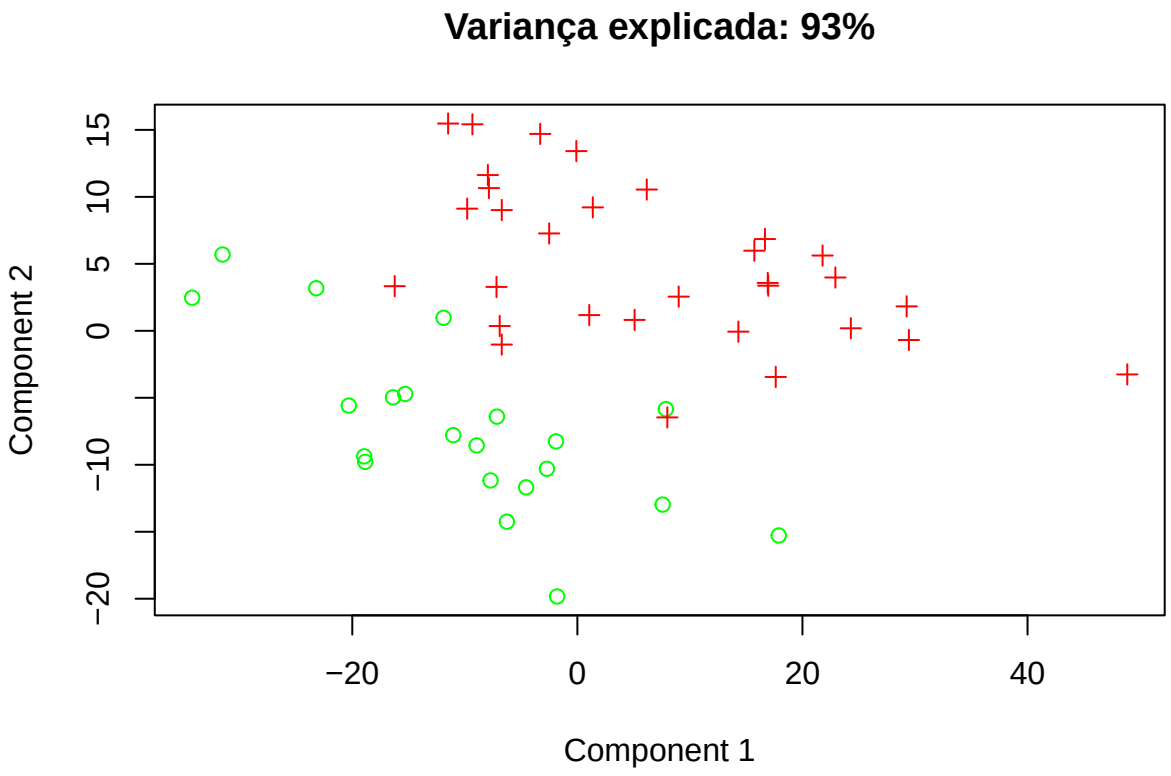
A continuació il · lustro aquesta correlació, i lo relativament fàcil que hauria de ser (aparentment[1]) aquesta

[1]He de dir que he provat de realitzar un anàlisi discriminant a partir de les dues primeres components principals (les que mostro en el gràfic de la següent pàgina), tant amb la descomposició a partir de la matriu de covariances, com a partir de la matriu de correlacions, i cap de les dues aproximacions han millorat els resultats obtinguts amb les dades *raw*. És més, les prediccions del conjunt *test* obtingudes després de dur a terme l'anàlisi discriminant a partir de les components principals extretes de la matriu de covariances resulten en exactament la mateixa taula de confusió (matriu de confusió) que utilitzant les dades *raw* – això deu ser degut a les propietats d'anàlisi discriminant lineal i PCA que ara mateix se'm escapen (deu estar relacionat amb que PCA és una transformació lineal de les dades, el mateix tipus de transformació que utilitza LDA), però

separació entre les dues espècies, mitjançant una descomposició en les dues primeres components principals, que expliquen més d'un 90% de la variança de les dades.

```
pca_ <- princomp(flea[, -1], scores = TRUE)
y <- pca_$scores[, 1:2]

plot(y[, 1], y[, 2],
     xlab = "Component 1",
     ylab = "Component 2",
     main = paste("Variança explicada: ",
                  round(cumsum(pca_$sdev)/sum(pca_$sdev), 2)[2] * 100, "%", sep=""),
     pch = c(1, 3)[as.numeric(flea$species)],
     col = c('green', 'red')[as.numeric(flea$species)])
```



b) Contrast d'homogeneitat de variances

Com que estic cansat de calcular el test `llr`, vaig a escriure una funció que ho faci.

```
cov_equality_check <- function(X, group){

  X <- as.matrix(X)
  group <- as.factor(as.character(group))

  p <- ncol(X)
  n <- nrow(X)
  nlev <- nlevels(group)
  lev <- levels(group)

  covs <- list()
```

m'agradaria entendre-ho, si pogués ser respost (vcasellesb@uoc.edu).

```
  cov_comm <- 0
  llr_tmp <- 0

  for (l in 1:nlev){
    tmp <- cov(X[group == lev[l], ])
    n_l <- sum(group==lev[l])
    covs[[l]] <- tmp * (n_l-1) / n_l
    cov_comm <- cov_comm + n_l * covs[[l]]
    llr_tmp <- llr_tmp + n_l * log(det(covs[[l]]))
  }

  cov_comm <- cov_comm / n

  llr <- n * log(det(cov_comm)) - llr_tmp
  dfs <- (nlev - 1) * p * (p + 1) / 2
  p_val <- pchisq(llr, df = dfs, lower.tail=FALSE)

  return(list(
    llr=llr,
    df=dfs,
    pval=p_val)
  )
}

res <- cov_equality_check(flea[, -1], flea$species)
res
```

```
## $llr
## [1] 5.627243
##
## $df
## [1] 6
##
## $pval
## [1] 0.466213
```

M'ha anat molt bé escriure la funció, ja que anteriorment em donava resultats excessivament diferents entre `boxM` de `heplots` i calculant-ho amb la *log-likelihood*.

```
heplots::boxM(flea[, -1], droplevels(flea$species))
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  flea[, -1]
## Chi-Sq (approx.) = 4.8599, df = 6, p-value = 0.5619
```

Així doncs, sembla ser que no hi ha diferències entre les matrius de covariances d'ambdós grups. No ho mostro, però també ho he calculat amb el conjunt de dades `training` que he separat anteriorment i també resulta un p-valor superior a 0.05. Tot i això, per dur a terme anàlisi discriminant, tinc entès que s'ha de complir la assumpció de normalitat multivariant.

```
MVN::mvn(data=flea[flea$species=='Concinna', -1], univariateTest = 'SW')$univariateNormality
```

```
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk   tars1      0.9802    0.9287    YES
```

```
## 2 Shapiro-Wilk   tars2      0.9587    0.4911    YES
## 3 Shapiro-Wilk   head       0.9699    0.7303    YES
```

```
MVN::mvn(data=flea[flea$species=='Concinna', -1], mvnTest = 'mardia')$multivariateNormality
```

```
##              Test          Statistic           p value Result
## 1 Mardia Skewness   9.37117534930242 0.497276834142472    YES
## 2 Mardia Kurtosis -0.531020902210195  0.59540429468735    YES
## 3             MVN               <NA>              <NA>    YES
```

Veiem que, per a la espècie *Concinna*, si que es compleix la normalitat univariant i multivariant. Anem a comprovar-ho per a l'altra espècie d'aquest apartat, *Heikert.*.

```
MVN::mvn(data=flea[flea$species=='Heikert.', -1], univariateTest = 'SW')$univariateNormality
```

```
##           Test  Variable Statistic   p value Normality
## 1 Shapiro-Wilk   tars1      0.9357    0.0627    YES
## 2 Shapiro-Wilk   tars2      0.9635    0.3597    YES
## 3 Shapiro-Wilk   head       0.9397    0.0808    YES
```

```
MVN::mvn(data=flea[flea$species=='Heikert.', -1], mvnTest = 'mardia')$multivariateNormality
```

```
##              Test          Statistic           p value Result
## 1 Mardia Skewness   8.60173322651139 0.570270609830045    YES
## 2 Mardia Kurtosis -0.252210761923696 0.800878160096768    YES
## 3             MVN               <NA>              <NA>    YES
```

També es compleix. Així doncs, entenc que podriem procedir a dur a terme anàlisi discriminant sense problemes. Finalment, per a determinar si està justificat l'anàlisi discriminant, duc a terme un anàlisi `MANOVA` d'acord a l'espècie.

```
m <- manova(cbind(tars1, tars2, head) ~ species, data=flea)
summary(m, test="Wilks")
```

```
##           Df   Wilks approx F num Df den Df    Pr(>F)
## species    1 0.20947   60.383      3     48 2.556e-16 ***
## Residuals 50
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Veiem que el resum del test `MANOVA` indica que hi ha diferències en les mitjanes de les tres variables d'acord a la espècie, així que està justificat dur a terme un anàlisi discriminant.

c) Anàlisi discriminant

Duc a terme l'anàlisi discriminant amb la funció `lda` del paquet `MASS`, amb les probabilitats *prior* com a 0.5 per a les dues espècies, tal i com es demana a l'enunciat.

```
lda_ <- MASS::lda(training[, -1],
                  grouping=droplevels(training$species),
                  prior=c(0.5,0.5))
```

d) Probabilitats *a posteriori*

Per a obtenir les probabilitats *a posteriori*, utilitzo la funció `predict` amb l'objecte generat a l'apartat anterior com a argument (predicció de les dades training) .

```
post <- predict(lda_)$posterior
post
```

```
##        Concinna     Heikert.
## 1  1.000000e+00 1.626150e-09
## 2  1.000000e+00 5.367063e-11
## 3  9.999999e-01 8.193337e-08
## 4  1.000000e+00 1.670247e-12
## 5  9.999997e-01 3.020108e-07
## 6  1.000000e+00 2.601439e-10
## 7  1.000000e+00 1.256784e-13
## 8  1.000000e+00 1.155865e-08
## 9  1.000000e+00 2.380094e-16
## 10 9.999997e-01 3.339652e-07
## 44 1.896528e-07 9.999998e-01
## 45 2.388333e-10 1.000000e+00
## 46 2.404398e-12 1.000000e+00
## 47 1.898086e-14 1.000000e+00
## 48 5.684684e-12 1.000000e+00
## 49 1.214784e-10 1.000000e+00
## 50 4.090246e-13 1.000000e+00
## 51 5.342607e-11 1.000000e+00
## 52 3.090247e-06 9.999969e-01
## 53 1.123980e-06 9.999989e-01
```

Podem veure si el conjunt de dades *training* ha sigut predit efectivament.

```
pred_tr <- apply(post, MARGIN=1, FUN=which.max)
# 1 == TRUE
cbind(pred=pred_tr,
      true=as.numeric(training$species),
      match=pred_tr == as.numeric(training$species))
```

```
##    pred true match
## 1     1    1     1
## 2     1    1     1
## 3     1    1     1
## 4     1    1     1
## 5     1    1     1
## 6     1    1     1
## 7     1    1     1
## 8     1    1     1
## 9     1    1     1
## 10    1    1     1
## 44    2    2     1
## 45    2    2     1
## 46    2    2     1
## 47    2    2     1
## 48    2    2     1
## 49    2    2     1
## 50    2    2     1
## 51    2    2     1
## 52    2    2     1
## 53    2    2     1
```

Efectivament, les prediccions són 100% acertades (menys mal, és el conjunt amb el qual hem dut a terme l'anàlisi).

e) Prediccions

Predim les espècies de les dades *test* a continuació.

```
pred <- predict(lda_, newdata=as.matrix(test[, -1]))

pred_c <- pred$class
table(predicció=pred_c, true=droplevels(test$species))
```

```
##           true
## predicció Concinna Heikert.
##   Concinna       11        4
##   Heikert.        0       17
```

Com veiem, hi han 4 subjectes que han sigut erròniament classificats com a *Concinna*, quan eren realment *Heikert.*. En canvi, tots els *Heikert.* han sigut correctament classificats. Així doncs, l'*error rate* és de:

```
t <- table(predicció=pred_c, true=droplevels(test$species))
(1- sum(diag(t)) / sum(t)) * 100
```

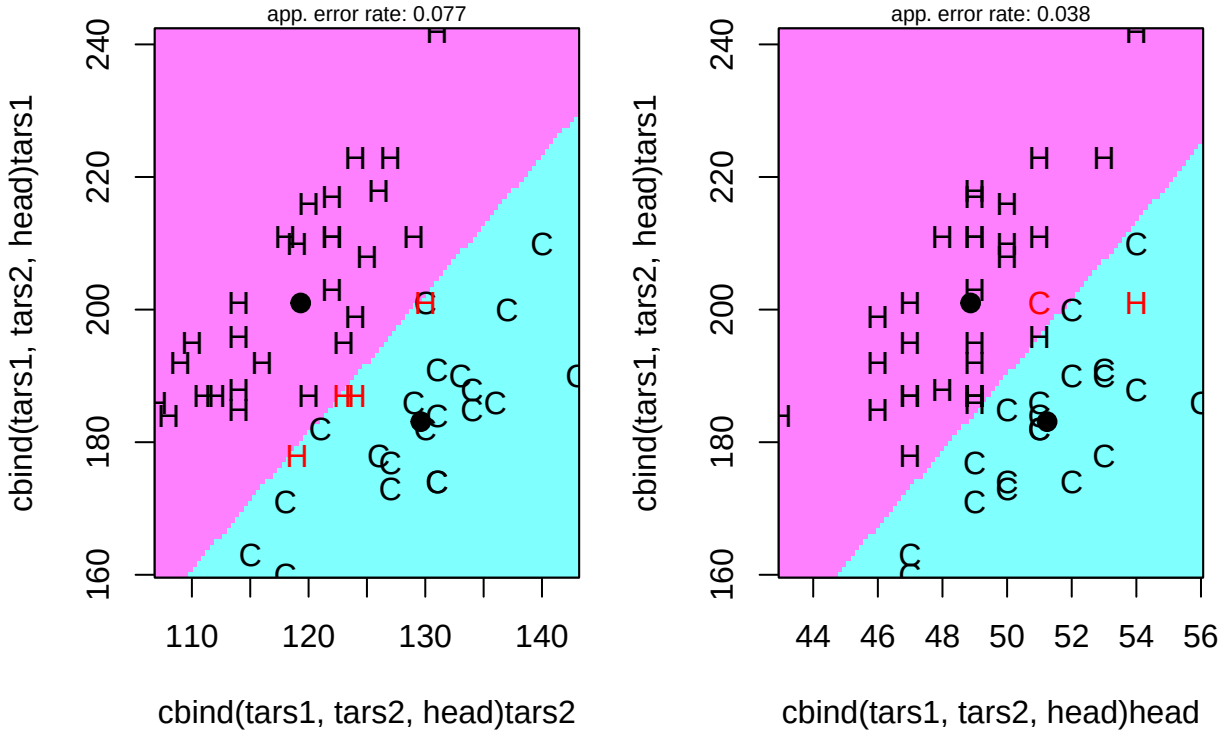
```
## [1] 12.5
```

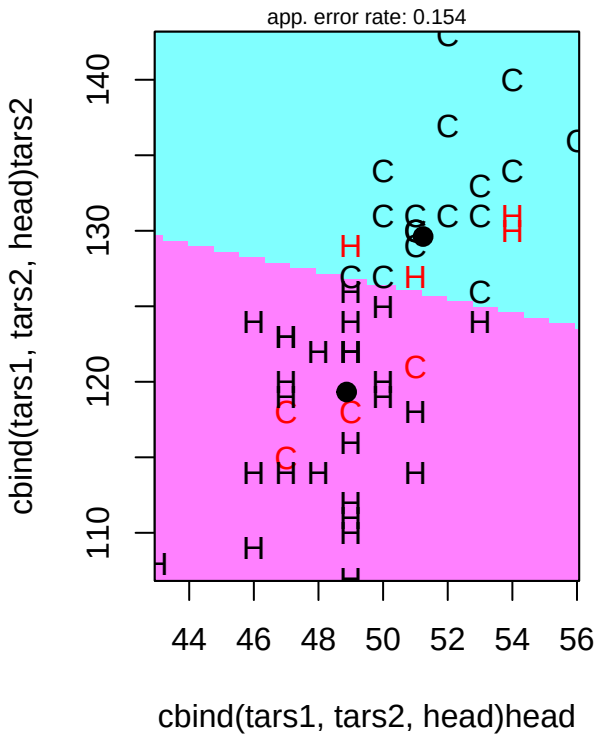
12.5%. Aquest resulta força alt.

f) `partimat`

A continuació duc a terme els gràfics resultats de classificació respecte a tot el *dataset* mitjançant anàlisi discriminant lineal.

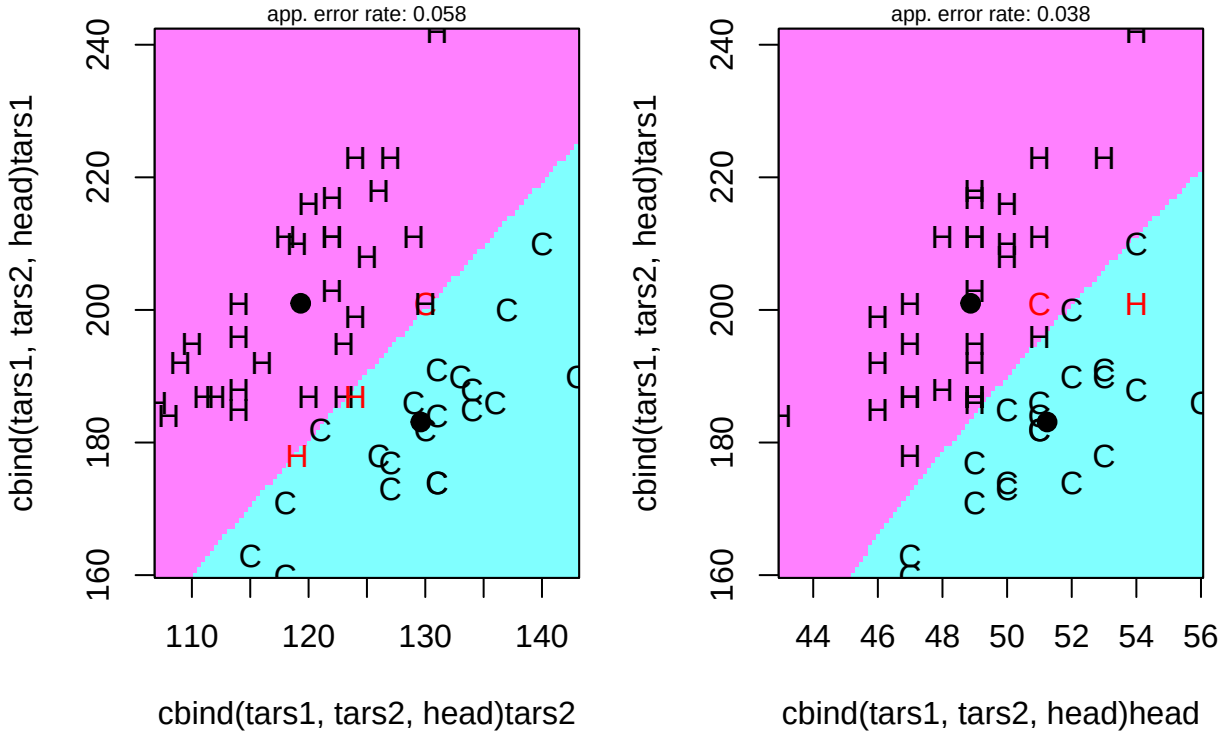
```
klaR::partimat(droplevels(species) ~ cbind(tars1, tars2, head), data=flea, method='lda')
```

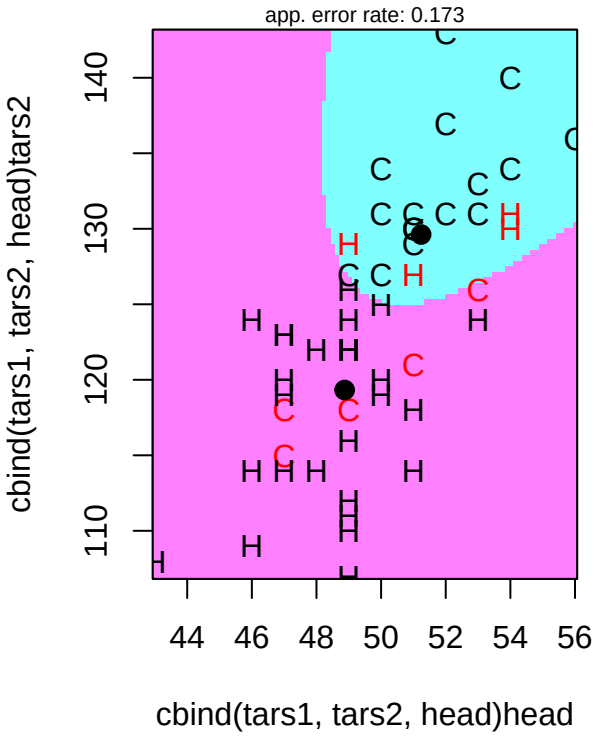




Ara ho duc a terme mitjançant anàlisi discriminant quadràtic.

```
klaR::partimat(droplevels(species) ~ cbind(tars1, tars2, head), data=flea, method='qda')
```





Veiem que, en ambdós casos, els millors resultats s'obtenen utilitzant les variables `tars1` i `head`. Jo triaria el mètode lineal, ja que utilitzant mètodes no lineals augmenta bastant la probabilitat de fer *overfitting*, i també augmenta la complexitat del model. A més a més, els resultats del model quadràtic no milloren els del lineal (el mínim de *error rate* és el mateix en els dos casos, 0.038).

Exercici 3. Anàlisi de conglomerats

Llegeixo les dades.

```
wood <- read.table('wood.txt')
```

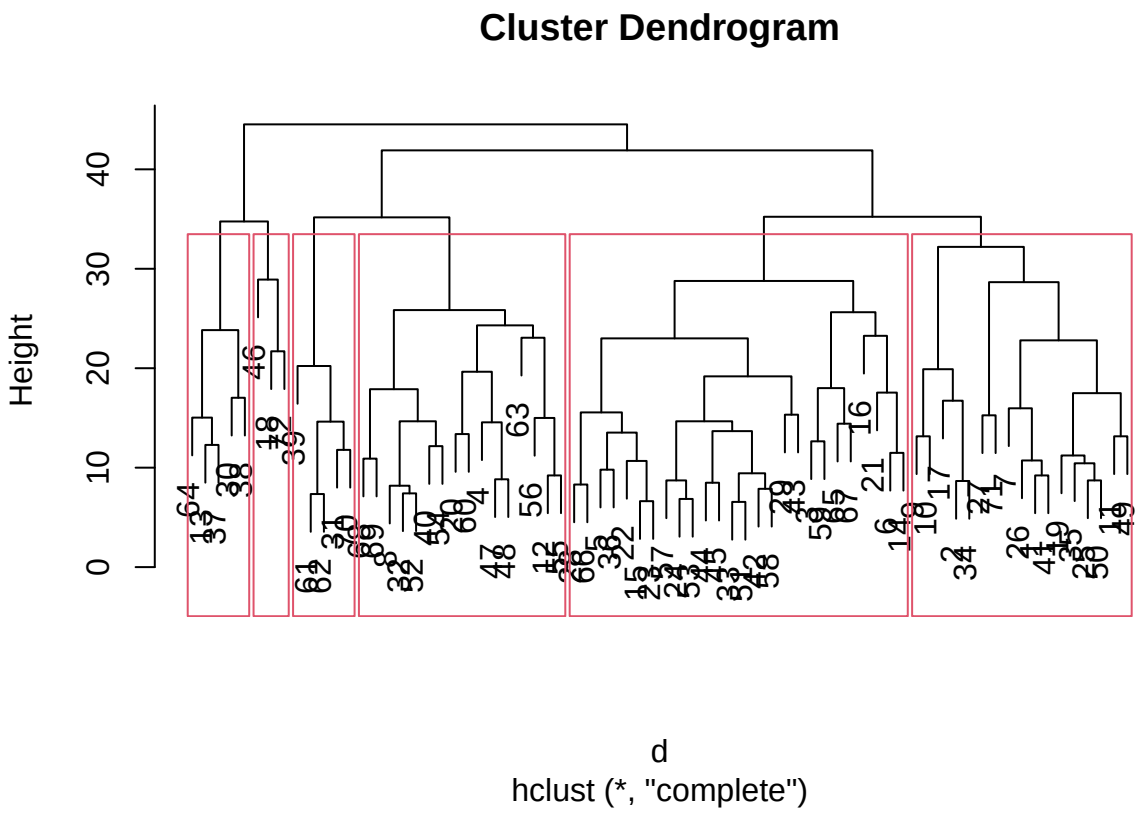
a) Anàlisi jeràrquic aglomeratiu amb distàncies euclídies i *complete linkage*

Primer de tot calculo les distàncies euclídies, i genero l'objecte `hclust` amb la funció homònima.

```
d <- dist(wood)
clust <- hclust(d=d, method = 'complete')
```

Utlitzant `plot` i `rect.hclust`, puc graficar el dendograma generat per `hclust` i marcar els 6 *clusters* que es demanen a l'enunciat.

```
plot(clust)
x <- rect.hclust(clust, k=6)
```



Guardo els resultats del *call* a `rect.hclust`, que bàsicament són una llista amb els indexos (*sites* de mostreig del *dataset*) i el *cluster* al que correspon cada un, per a dur a terme els apartats subseqüents.

b) Anàlisi de variança amb el factor *cluster*

Vaig a *parsejar* a quin d'aquests 6 *clusters* pertany cada observació. Recordem, `x` conté una llista amb els clusters i els índexos (observacions, *sites*) que pertanyen a cada un d'aquests.

```
cl <- 1
sixclusters <- numeric(length=nrow(wood))
for (l in x){
  indexes <- unname(l)
```

```
  sixclusters[indexes] <- cl
  cl <- cl + 1
}
```

Ara tinc un vector que conté el *cluster* obtingut anteriorment al que pertany cada una de les observacions. Vaig a afegir-lo al *dataframe* `wood`.

```
wood$sixclust <- as.factor(sixclusters)
```

```
species <- pval <- F_stat <- numeric(ncol(wood) - 1) # no m'interessa sixclust
for (i in 1:(ncol(wood)-1)){
  aov_ <- summary(aov(wood[, i] ~ wood$sixclust))
  species[i] <- colnames(wood)[i]
  pval[i] <- aov_[[1]][["Pr(>F)"]][1] # extret de stackoverflow (no se quina entrada exactament)
  F_stat[i] <- aov_[[1]][["F value"]][1]
}

pval <- p.adjust(pval, method='bonferroni') # ajustem els p-valors
```

Genero la taula que es demana a continuació.

```
require(knitr, quietly=T, warn.conflicts=F)
parsed_results <- data.frame(species = species,
                             Fstats = F_stat,
                             pvalues = pval,
                             significative = ifelse(pval<0.05, "*", ""))

kable(parsed_results,
      caption = paste('Number of significant species: ', sum(pval<0.05)))
```

Table 1: Number of significant species: 7

species	Fstats	pvalues	significative
carcar	62.9352806	0.0000000	*
corflo	1.5486533	1.0000000	
faggra	7.1065090	0.0002953	*
ileopa	3.4226847	0.1071796	
liqsty	5.8660589	0.0019818	*
maggra	3.9713488	0.0427378	*
nyssyl	1.6598089	1.0000000	
ostvir	17.7020897	0.0000000	*
oxyarb	1.4179141	1.0000000	
pingla	0.4323609	1.0000000	
quenig	2.2327317	0.7959025	
quemic	4.1225263	0.0332358	*
symtin	75.5714240	0.0000000	*

A continuació, agafo aquelles espècies on les diferències són significatives d'acord als anàlisi de variança (ANOVA) anteriorment computats.

```
sig_sp <- species[pval < 0.05]
```

Amb l'operador `%in%` *fàcilment*[2] puc computar les mitjanes que se'm demanen.

[2] m'ha costat més del que pensava

```r
require(dplyr,
        warn.conflicts = FALSE,
        quietly=TRUE)

means_ <- wood[, colnames(wood)%in%c(sig_sp, "sixclust")] %>%
            group_by(sixclust) %>%
            summarise(across(everything(), list(mean)))
```

Genero la taula, basant-me en [3].

```r
require(knitr); require(kableExtra)
means_ <- as.matrix(sapply(means_, as.numeric))[, -1] # trec sixclust
rownames(means_) <- 1:6
colnames(means_) <- sig_sp
means_ <- round(means_, 4)

maxes <- apply(means_, 2, which.max) # màxim de cada espècie
rows <- seq_len(nrow(means_))

for (c in 1:ncol(means_)){
  # en negreta els màxims de cada espècie
  means_[,c] <- means_[,c] %>% cell_spec(bold = rows == maxes[c])
}
```

A continuació mostro el resultat.

```r
means_ %>% kable(booktabs = TRUE, escape = FALSE, row.names = TRUE)
```

	carcar	faggra	liqsty	maggra	ostvir	quemic	symtin
1	8.2	8.6	6.6	4.6	3.6	7	18
2	6	2.6667	**18**	0.6667	**14**	2.3333	**20**
3	**24.4**	6.4	17.4	3.8	2.8	5.2	0
4	18.5	5.9375	6.4375	2.75	2.875	**9.375**	0.6875
5	3.8462	**11.3846**	7.1923	**5.2692**	4.2692	5.2692	0.9231
6	1.2353	5.9412	6.7647	3.2353	13.8235	4.1176	2

En negreta podem observar el màxim de cada espècie d'acord al *cluster*. He remarcat el màxim **per columna**, enlloc del màxim per fila, encara que trobo que la interpretació de l'enunciat podia anar pels dos costats. Espero no haver-me equivocat.

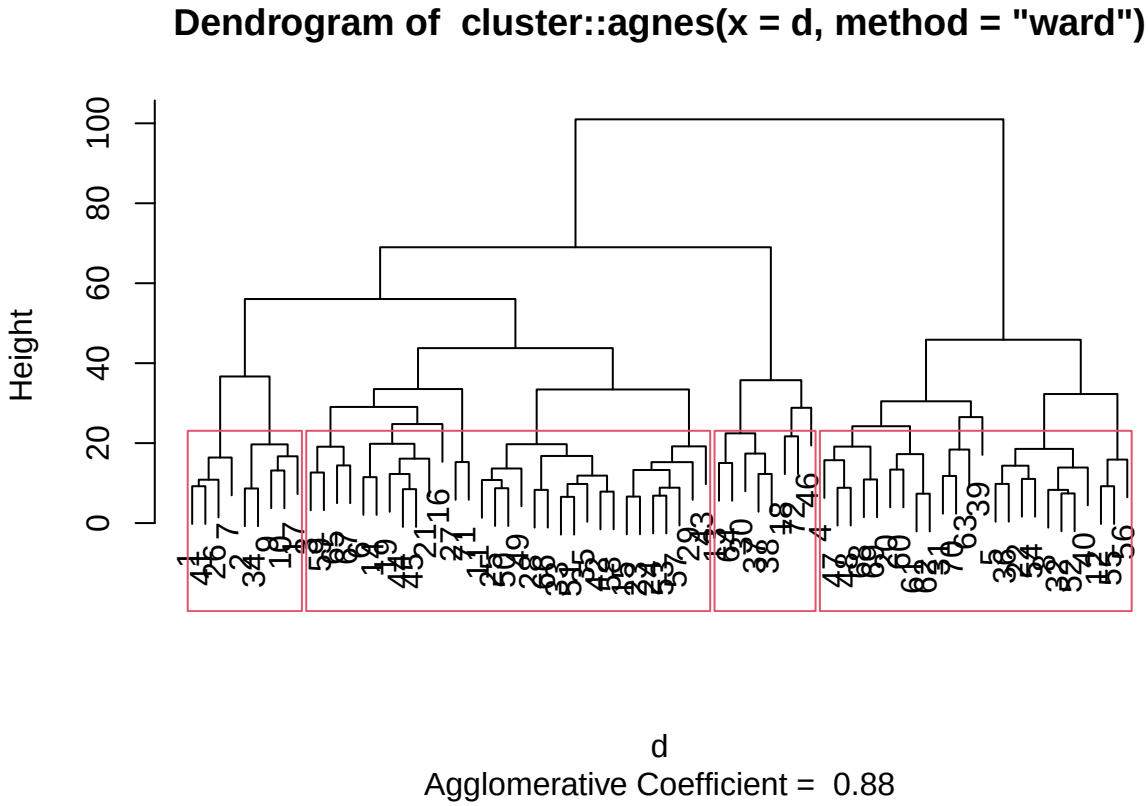
c) Mètode de Ward amb $k = 4$

Repeteixo el mateix procediment que a l'apartat anterior, només canviant el mètode (i la funció) utilitzat per a realitzar el *clustering*. Limito els comentaris d'aquest apartat per a no fer-me repetitiu.

```r
d <- dist(wood[, -ncol(wood)])
clust <- cluster::agnes(d, method="ward")

plot(clust, which.plots = 2) # trio que es grafiqui el dendograma
x <- rect.hclust(clust, k=4)
```

[3] https://stackoverflow.com/questions/61391380/kableextra-how-can-i-set-to-bold-the-biggest-value-of-the-row



Veiem que la funció `rect.hclust` no funciona tan bé com quan l'objecte `hclust` provenia de la funció homònima (el límit superior dels rectangles no es correspon a l'alçada que dóna els 4 *clusters*).

```
cl <- 1
fourclusters <- numeric(length=nrow(wood))
for (l in x){
  indexes <- unname(l)
  fourclusters[indexes] <- cl
  cl <- cl + 1
}

names(fourclusters) <- rownames(wood)

wood$fourclust <- as.factor(fourclusters)
```

```
species <- pval <- F_stat <- numeric(ncol(wood) - 2) # fora sixclust i fourclust
for (i in 1:(ncol(wood)-2)){
aov_ <- summary(aov(wood[, i] ~ wood$fourclust))
species[i] <- colnames(wood)[i]
pval[i] <- aov_[[1]][["Pr(>F)"]][1]
F_stat[i] <- aov_[[1]][["F value"]][1]
}

pval <- p.adjust(pval, method='bonferroni')
```

```
parsed_results <- data.frame(species = species,
                             Fstats = F_stat,
                             pvalues = pval,
                             significative = ifelse(pval<0.05, "*", ""))
```

```r
kable(parsed_results,
      caption=paste('Number of significant species: ', sum(pval<0.05)))
```

Table 3: Number of significant species: 5

species	Fstats	pvalues	significative
carcar	67.4184552	0.0000000	*
corflo	2.3136244	1.0000000	
faggra	7.1282769	0.0040475	*
ileopa	5.3770140	0.0287853	*
liqsty	0.7628793	1.0000000	
maggra	2.7492843	0.6423637	
nyssyl	1.3592387	1.0000000	
ostvir	32.9108904	0.0000000	*
oxyarb	3.1530369	0.3947666	
pingla	1.0322235	1.0000000	
quenig	2.3944723	0.9864573	
quemic	3.4385131	0.2802128	
symtin	120.9470720	0.0000000	*

```r
sig_sp <- species[pval < 0.05]

means_ <- wood[, colnames(wood)%in%c(sig_sp, "fourclust")] %>%
  group_by(fourclust) %>%
  summarise(across(everything(), list(mean)))
```

```r
means_ <- as.matrix(sapply(means_, as.numeric))[, -1] # trec fourclust
rownames(means_) <- 1:4
colnames(means_) <- sig_sp
means_ <- round(means_, 4)
maxes <- apply(means_, 2, which.max)
rows <- seq_len(nrow(means_))
for (c in 1:ncol(means_)){
means_[,c] <- means_[,c] %>% cell_spec(bold = rows == maxes[c])
}

means_ %>% kable(booktabs = TRUE, escape = FALSE, row.names = T)
```

	carcar	faggra	ileopa	ostvir	symtin
1	1	5.8889	**12.3333**	**18.3333**	1.4444
2	2.7742	**10.5806**	7.5484	5.3548	1.2903
3	7.375	6.375	7.875	7.5	**18.75**
4	**18.5**	5.9583	4.2917	3.125	0.6667

Veiem que el número d'espècies significatives s'ha reduit de 7 (amb $k = 6$ i mètode `complete`) a 5 (amb $k = 4$ i mètode de `ward`). A continuació, seguint la solució de l'exercici 3 dels exercicis d'anàlisi de conglomerats, on s'avaluen diferents valors de k mitjançant la funció `silhouette` del paquet `cluster`, vaig a utilitzar la *average silhouette width* com a mesura comparativa entre els resultats que hem obtingut fins ara.

```
sil <- cluster::silhouette(fourclusters, d)
paste('Ward method with k = 4 ->', round(mean(sil[, "sil_width"]), 3))
```

```
## [1] "Ward method with k = 4 -> 0.187"
```

```
sil <- cluster::silhouette(sixclusters, d)
paste('Complete method with k = 6 ->', round(mean(sil[, "sil_width"]), 3))
```

```
## [1] "Complete method with k = 6 -> 0.139"
```

Veiem que, amb 4 clusters i el mètode de Ward, obtenim millors resultats. Tot i això, els *average silhouette width* és molt baix, per sota del *threshold* de 0.5 que recomana Wikipedia[4] com a indicador d'un *clustering* raonablement fort.

d) *k-means* amb l'algorisme de Hartigan-Wong

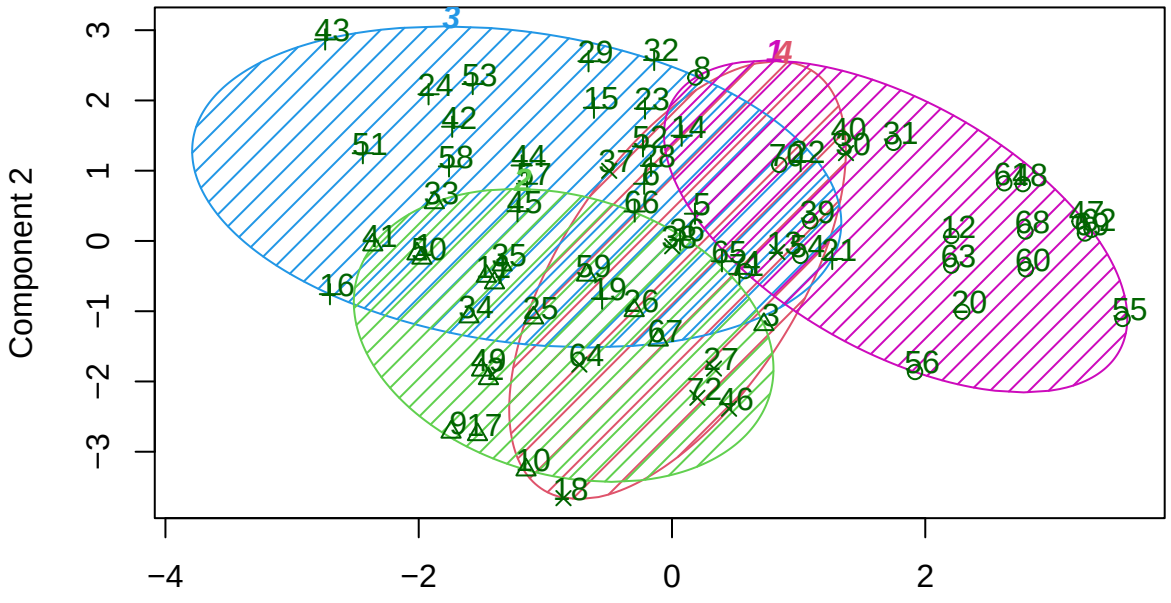
Fixo la *llavor* per a garantir la reproducibilitat tal i com es fa als exercicis del tema `R6`.

```
x <- wood[, -c(ncol(wood), ncol(wood)-1)] # trec sixclust i fourclust
set.seed(123)
clust_kmeans <- kmeans(x,
                       centers=4,
                       algorithm="Hartigan-Wong")
```

Realitzo el gràfic que es demana amb la funció `clusplot` del paquet `cluster`.

```
cluster::clusplot(x,
                  clust_kmeans$cluster,
                  color=TRUE,
                  shade=TRUE,
                  labels=2,
                  lines=0)
```

[4] https://en.wikipedia.org/wiki/Silhouette_(clustering)



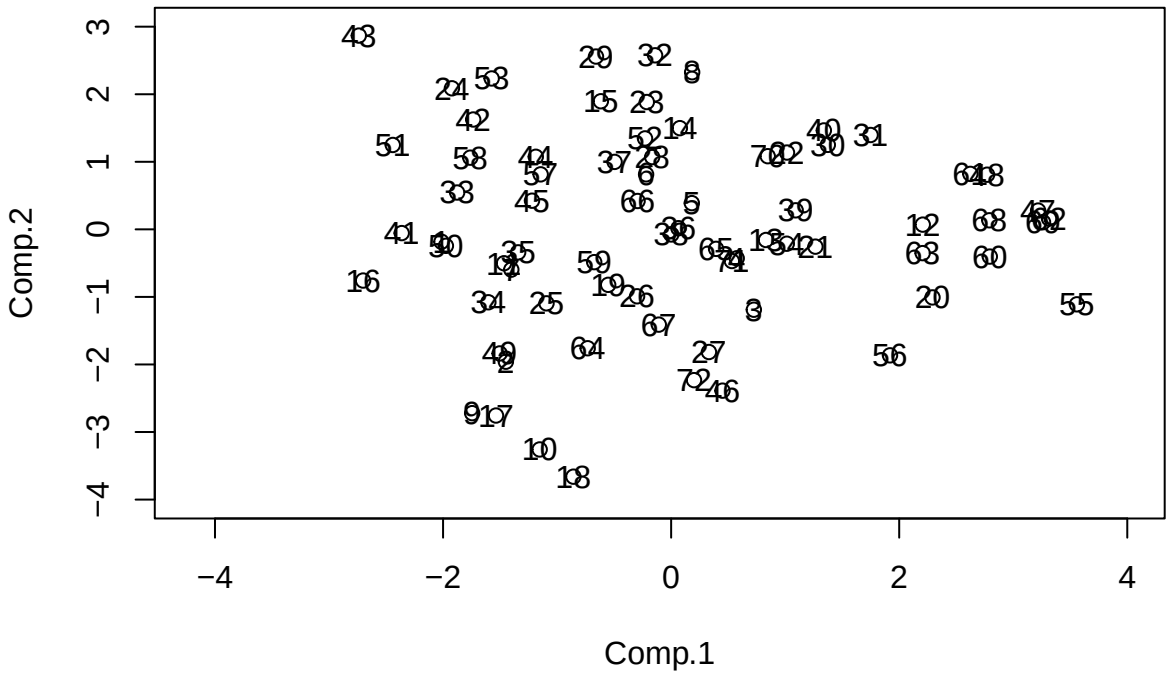
Veiem que la descomposició en els dos components principals només explica un 35.63% de la variabilitat total del conjunt de dades. Això em sembla força poc. Per altra banda, cal destacar que els clusters s'*overlapegen*.

Per a obtenir els *centers* de cada un dels *clusters* de manera que es puguin graficar en un gràfic bidimensional, he de portar-los a aquest espai inferior. Buscant a internet, he trobat que `clusplot` utilitza `princomp` a partir de la matriu de correlacions per a obtenir la descomposició de les dades en components principals. Per tant, per a poder graficar les mitjanes de cada un dels *clusters* en el gràfic anterior, necessito extreure les mitjanes, per *cluster*, d'aquestes components.

```
pr <- princomp(x, scores=TRUE, cor=T)
wood2d <- pr$scores[, 1:2] # seleccionem els components que es mostren al gràfic clusplot
wood2d <- as.data.frame(wood2d)
```

Comprovo ràpidament que és amb aquests components principals amb els que realitza el gràfic `clusplot`.

```
plot(wood2d[, 1:2], xlim=c(-4.2, 4), ylim=c(-4, 3))
text(wood2d[, 1:2], labels=rownames(wood))
```



Efectivament, podem procedir a extreure les mitjanes *across clusters* d'aquestes dues components principals.

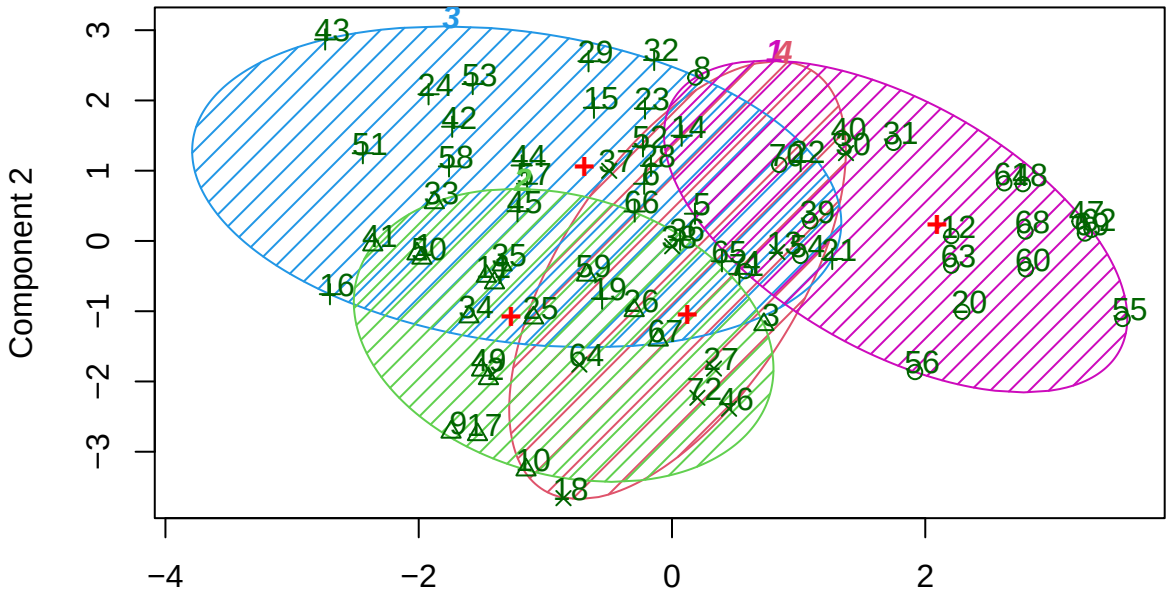
```
wood2d$clust <- as.factor(clust_kmeans$cluster)

means_by_cluster2d <- wood2d %>%
                        group_by(clust) %>%
                        summarise(across(1:2, list(mean)))

means_by_cluster2d <- as.matrix(means_by_cluster2d[, -1])

cluster::clusplot(x, clust_kmeans$cluster,
                  color=TRUE, shade=TRUE, labels=2, lines=0)

text(means_by_cluster2d, labels="+",col="red",font=2,cex=1.2)
```



Veiem que les mitjanes que he obtengut no es corresponen exactament al que, visualment, es pot observar com el centre dels el · lipses. Això, pensant-hi, és possible que es degui al sol · lapament que exhibeixen les el · lipses. És possible que, en la realització del gràfic, `clusplot` tingui en compte el sol · lapament, mentre que jo computant les mitjanes, no ho tinc en compte.

Per a trobar les mitjanes a cada *cluster* de les dades en la seva dimensió original, es faria a partir de l'objecte que he generat al principi de l'apartat.

```
round(clust_kmeans$centers, 2)
```

```
##   carcar corflo faggra ileopa liqsty maggra nyssyl ostvir oxyarb pingla quenig
## 1  20.74   1.26   5.47   4.84   9.53   2.63   2.58   2.84   0.26   2.47   3.58
## 2   1.56   2.72   7.17  12.67   7.28   3.83   1.78  13.56   2.22   2.22   1.17
## 3   4.50   2.31  11.12   4.65   6.42   5.15   0.96   3.96   1.31   2.69   1.35
## 4   6.56   2.44   5.89   7.78  11.11   3.00   1.22   7.22   0.33   4.56   1.56
##   quemic symtin
## 1   8.79   0.58
## 2   5.39   1.17
## 3   4.46   1.04
## 4   5.00  17.78
```

I per a trobar el tamany de cada un dels conglomerats o *clusters*, ho podem fer de dues maneres simples.

```
rbind(directe=clust_kmeans$size, table=table(clust_kmeans$cluster))
```

```
##          1  2  3 4
## directe 19 18 26 9
## table   19 18 26 9
```

Veiem que el conglomerat amb més observacions és el tercer. Per a calcular la *sum of squares* de cada un dels grups, l'objecte generat a l'inici també conté la clau.

```
clust_kmeans$withinss
```

```
## [1] 3277.789 2773.722 3616.615 2256.889
```

Veiem que, el cluster amb més observacions és al que correspon una *sum of squares* major, i el que té menys observacions és el que té una *sum of squares* menor.

Com a comprovació, utilitzo la funció definida a l'apartat 1f per a calcular les matrius de *within-class sums of squares and cross-products* de a cada grup. La suma de les diagonals em permet obtenir les *sums of squares* de cada un dels grups.

```
ww <- sum_of_squares(df = x, f=clust_kmeans$cluster, mode='within', sum=FALSE)
ss <- lapply(ww, FUN=function(x) sum(diag(x)))

rbind(ss_vicent=ss, ss_true=clust_kmeans$withinss)
```

```
##           [,1]     [,2]     [,3]     [,4]
## ss_vicent 3277.789 2773.722 3616.615 2256.889
## ss_true   3277.789 2773.722 3616.615 2256.889
```

Veiem que els resultats són els mateixos. De la mateixa manera que he fet abans, vaig a comprovar quin és el *average silhouette width* per a aquest *clustering method*.

```
sil <- cluster::silhouette(clust_kmeans$cluster, d)
paste('k-means method with k = 4 ->', round(mean(sil[, "sil_width"]), 3))
```

```
## [1] "k-means method with k = 4 -> 0.186"
```

Veiem que, amb aquesta mètrica, obtenim un promig molt similar a l'obtingut amb el mètode de Ward amb $k = 4$.

e) k-medoides

Utilitzo la funció `pam` del paquet `cluster`, amb un valor de $k = 4$.

```
pam4 <- cluster::pam(x=x, k=4, diss=F)
```

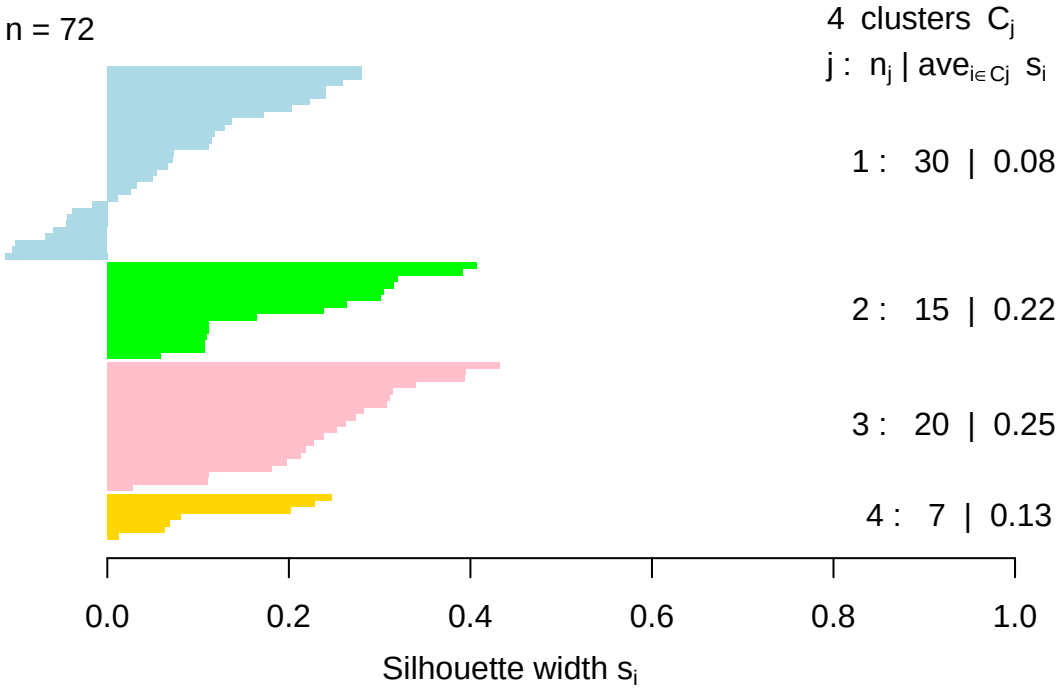
A partir de l'objecte generat (`pam4`) podem obtenir els *medoids*.

```
pam4$medoids
```

```
##    carcar corflo faggra ileopa liqsty maggra nyssyl ostvir oxyarb pingla quenig
## 50      2      3      8      8      6      5      1     11      4      5      0
## 47     27      1      3      1     11      3      5      2      0      1      4
## 15      7      4     10      0      4      7      0      2      0      0      1
## 37      6      2     11      5      5      7      2      3      0      1      0
##    quemic symtin
## 50      1      1
## 47     10      3
## 15      5      1
## 37      6     17
```

Aquí veiem els valors de les 13 espècies corresponents als 4 medoides que es generen. Ara vaig a generar el gràfic *silueta* que es demana.

```
plot(cluster::silhouette(pam4), col= c("lightblue","green","pink","gold"))
```



Average silhouette width : 0.16

D'acord a l'entrada de Wikipedia que es dóna com a referència a l'enunciat de l'exercici, un *average silhouette width* inferior a 0.25 indica un *clustering* pobre, on els objectes dins del propi *cluster* s'assemblen relativament poc entre ells comparat amb la semblança amb els objectes d'altres *clusters*. Amb el *clustering* generat en aquest apartat obtenim un *average silhouette width* de 0.16, indicant que aquest *clustering* és pobre (tampoc millora cap dels *widths* anteriorment observats). Això, tal i com es menciona a l'article de la Wikipedia mencionat, és bastant probable que sigui degut a que les dades són d'una dimensió alta (hi ha moltes variables), fet que dificulta trobar una agrupació al voltant de medoides (o mitjanes) que sigui relevant o robusta.

Buscant a Internet, he trobat que això es podria solventar o bé utilitzant pesos que reflecteixin que algunes dimensions poden ser menys rellevants que d'altres a l'hora de descriure les dades, utiltzant *feature selection* per a reduir el número de dimensions, o bé utilitzant alguna tècnica de reducció de la dimensionalitat com PCA.

Com a comentari extra, he de dir que m'he estat debatent força sobre com tractar les dades d'aquest exercici, donat que són *counts*. He pensat que potser s'havien de convertir en freqüències relatives (i que aquesta era la trampa de l'exercici), però no m'he atrevit a fer-ho ja que enlloc de l'enunciat de cap dels apartats sembla haver cap indicació de que això és el que s'hauria d'haver fet. De totes formes, he buscat a Internet i he trobat el següent fòrum [5] on es comenta aquesta casuística. Bàsicament, un dels usuaris comenta que la millor manera de procedir a l'hora de voler dur a terme *clustering* amb dades d'aquest tipus, és computar distàncies respecte a aquestes i utilitzar-les com a *input* per al *clustering* (és el que hem fet). Tot i això, l'usuari argumenta que la millor manera de dur-ho a terme és amb distàncies Chi-quadrat. Nosaltres hem fet servir, generalment, distàncies euclídies.

En altres entrades del mateix portal de fòrum es recomana utilitzar mètodes de *linkage complete* o *average*[6], com s'ha indicat a l'enunciat d'aquest exercici. Cal dir que, encara que citar entrades d'`Stack Exchange` pot resultar una metodologia de citació poc autoritativa, aquestes han sigut realitzades totes pel mateix usuari, que és desenvolupador del famós programari d'anàlisi estadística `SPSS`.

[5] https://stats.stackexchange.com/questions/173636/clustering-of-very-skewed-count-data-any-suggestions-to-go-about-transform-et.

[6] https://stats.stackexchange.com/questions/304182/choice-of-clustering-method-with-frequency-data.

Buscant més informació respecte al tema, he trobat que l'anàlisi de *count data* és un àmbit amb força potència, per exemple en el cas de l'anàlisi de patrons d'expressió gènica (on s'analitza el recompte de molècules de DNA, RNA o proteïnes), o, com ens ocupa en aquesta PAC, l'anàlisi ecològic d'abundància de diferents espècies en diferents llocs de mostreig. En aquest sentit, sembla que un dels models que s'ha proposat per a realitzar *clustering* d'aquest tipus de dades passa per el seu model · litzat mitjançant la *multivariate Poisson-log normal distribution* (Aitchison, 1989)[7], que és un model mixte (Poisson i normal, tal i com indica el nom). Encara que no em correspon entrar-hi en aquest treball, seria una possibilitat estudiar si aplicar aquest *framework* per a realitzar un anàlisi potencialment més potent del *dataset* `wood.txt`.

[7]Exemple d'aplicació en seqüenciació de transcriptoma: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-2916-0.

Apèndix

Funció per a comprovar que `rect.hclust` i `cutree` funcionen de la mateixa manera. La primera funció retorna una llista amb el mateix número d'entrades que de *clusters*, i a cada entrada hi ha els seus membres. Per altra banda, `cutree` retorna un vector amb els membres als noms del vector i els elements que corresponen al *clúster* al que pertany cada un. El problema és que `rect.hclust` assigna uns números/ordre als *clústers* diferents que els que assigna `cutree`.

```
check_OCD <- function(vector_cutree, list_rect){

  if (!is.list(list_rect) | !is.vector(vector_cutree))
    stop(
      "wrong args. Got: '", class(vector_cutree), "' (expected vector); and '",
         class(list_rect), "' (expected list)"
      )

  for (l in list_rect){
    # get first one, determines the cluster we'll check from vector_cutree
    bossman <- l[1]

    # get the cluster as numbered in vector
    which_clust <- vector_cutree[names(vector_cutree) == bossman]

    # get all his friends
    friends <- names(vector_cutree)[vector_cutree == which_clust]

    # check if they match from list
    stopifnot(all(friends == l))
  }
}
```

```
# test 1
wood <- read.table('wood.txt')
d <- dist(wood)
clust <- hclust(d, method='complete')
vector_cutree <- cutree(clust, k=6)

plot(clust)
list_rect <- rect.hclust(clust, k=6)
```

```
check_OCD(vector_cutree = vector_cutree, list_rect = list_rect)
```

Test 1 passa.

```
# test 2
clust <- hclust(d, method='ward.D2')
vector_cutree <- cutree(clust, k=4)
plot(clust)
list_rect <- rect.hclust(clust, k=4)
```

```
check_OCD(vector_cutree = vector_cutree, list_rect = list_rect)
```

Veiem que si que funcionen equivalentment (la funció no falla en ambdós casos).