UNIVERSITAT DE BARCELONA

Universitat Oberta de Catalunya

Estadística Multivariante: Solución PEC2

Francesc Carmona

15 de junio de 2024

Ejercicio 1 (40 pt.)

Apartado (a)

En primer lugar vamos a cargar los datos y ver su contenido

```
library(GGally)
data(flea)
str(flea)
```

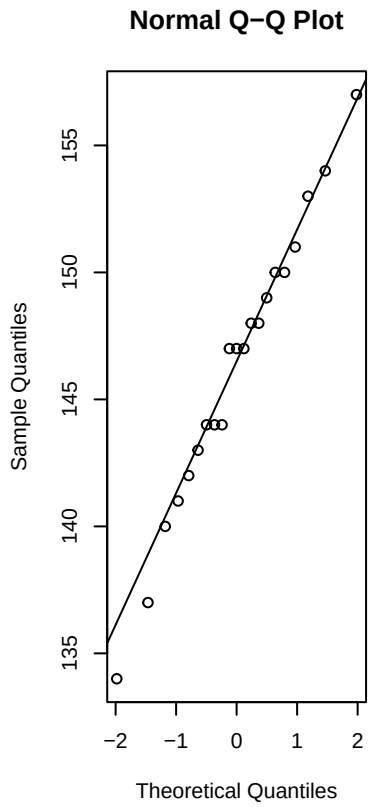
```
## 'data.frame':    74 obs. of  7 variables:
##  $ species: Factor w/ 3 levels "Concinna","Heikert.",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ tars1  : int  191 185 200 173 171 160 188 186 174 163 ...
##  $ tars2  : int  131 134 137 127 118 118 134 129 131 115 ...
##  $ head   : int  53 50 52 50 49 47 54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

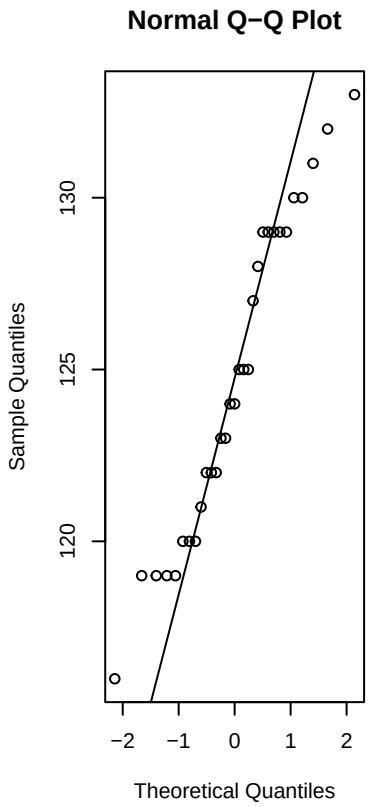
```
table(flea$species)
```

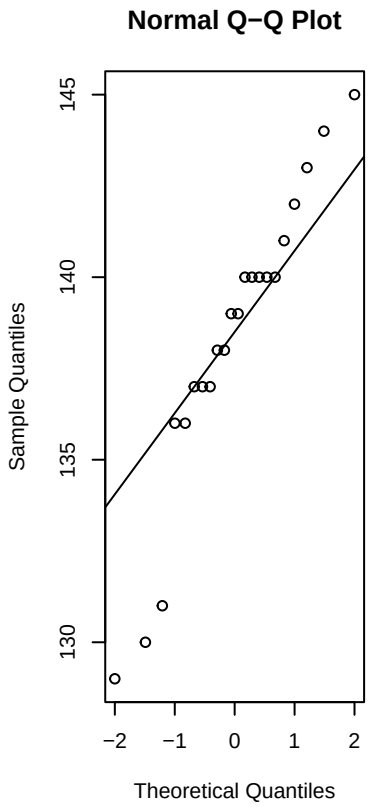
```
##
##  Concinna  Heikert. Heptapot.
##        21        31        22
```

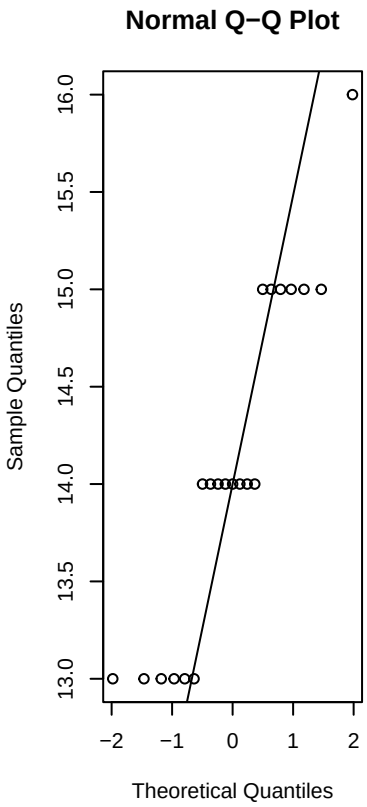
Veamos la normalidad univariante de las tres variables consideradas, separadas por especie.

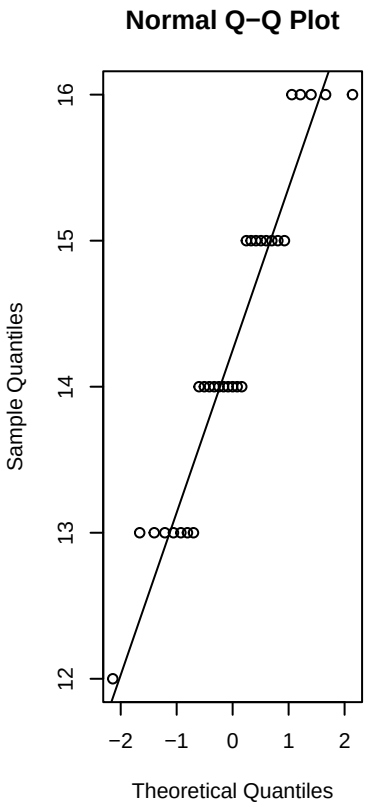
```
par(mfrow=c(1,3))
for(i in 5:7){
  for(esp in levels(flea$species)){
    qqnorm(flea[flea$species==esp,i])
    qqline(flea[flea$species==esp,i])
  }
}
```

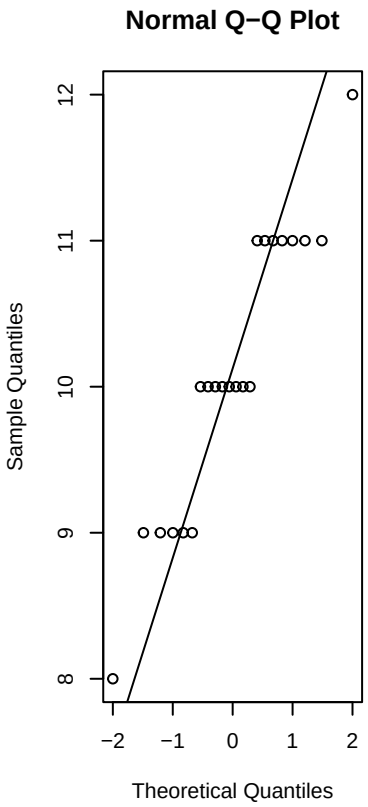

Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
135
140
145
150
155
−2
−1
0
1
2

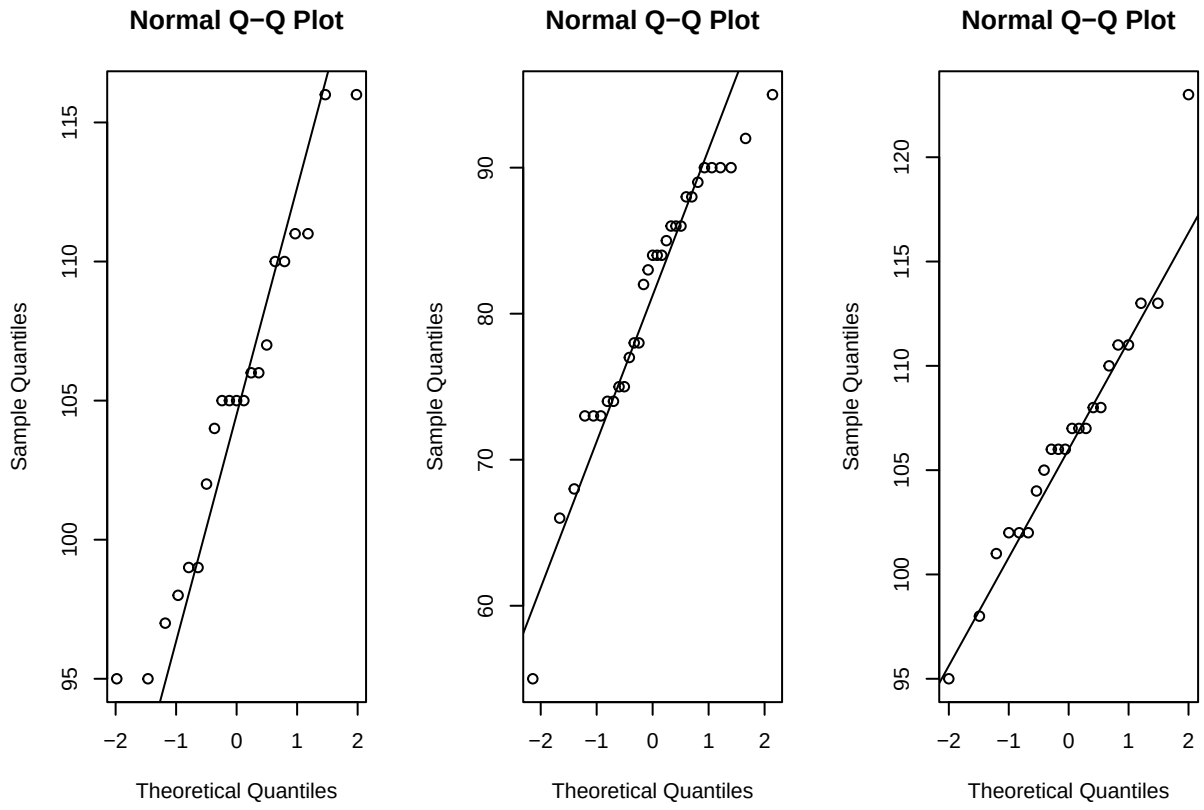

Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
120
125
130
−2
−1
0
1
2


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
130
135
140
145
−2
−1
0
1
2


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
13.0
13.5
14.0
14.5
15.0
15.5
16.0
−2
−1
0
1
2


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
12
13
14
15
16
−2
−1
0
1
2


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
8
9
10
11
12
−2
−1
0
1
2



Shapiro-Wilk normality test

```
mm <- matrix(numeric(), ncol=3, nrow=3)
rownames(mm) <- colnames(flea)[5:7]
colnames(mm) <- levels(flea$species)
for(i in 1:3){
  for(j in 1:3){
    ll <- levels(flea$species)[j]
  mm[i,j] <- shapiro.test(flea[flea$species==ll,i+4])$p.value
  }
}
# p values
round(mm,3)
```

```
##       Concinna Heikert. Heptapot.
## aede1    0.994    0.150     0.094
## aede2    0.008    0.014     0.050
## aede3    0.398    0.062     0.350
```

Es evidente que la segunda variable `aede2` tiene alguna dificultad para ser normal, ya que sus valores son enteros y muchos repetidos. Parece más bien una variable con distribución normal pero con valores redondeados.

Apartado (b)

Vamos a estudiar la normalidad multivariante con el test de Mardia.

Para los individuos de cada especie por separado

```
flea.conc <- flea[flea$species=="Concinna",5:7]
flea.heik <- flea[flea$species=="Heikert.",5:7]
flea.hept <- flea[flea$species=="Heptapot.",5:7]
skew.and.kurt.Mardia <- function(subsetData){
  X <- scale(subsetData, center = T, scale = F) # centered data
  n <- dim(X)[1]
  p <- dim(X)[2]
  Sinv <- solve(cov(X)*(n-1)/n)
  M <- X %*% Sinv %*% t(X)
  b1p <- sum(M^3)/n^2
  b2p <- sum(diag(M)^2)/n
  g1p <- b1p*n/6         # chi.df=p*(p+1)*(p+2)/6
  g2p <- b2p-p*(p+2)     # N(0,1)
  M.skew <- g1p
  M.kurt <- g2p/sqrt(8*p*(p+2)/n) # Normalized Multivariate Kutosis
  c(b1p=b1p,b2p=b2p,g1p=g1p,g2p=g2p,M.skew=M.skew,M.kurt=M.kurt)
}
skew.and.kurt.Mardia(flea.conc)
```

```
##       b1p       b2p       g1p       g2p    M.skew    M.kurt
##  1.143320 12.352837  4.001619 -2.647163  4.001619 -1.107388
```

```
skew.and.kurt.Mardia(flea.heik)
```

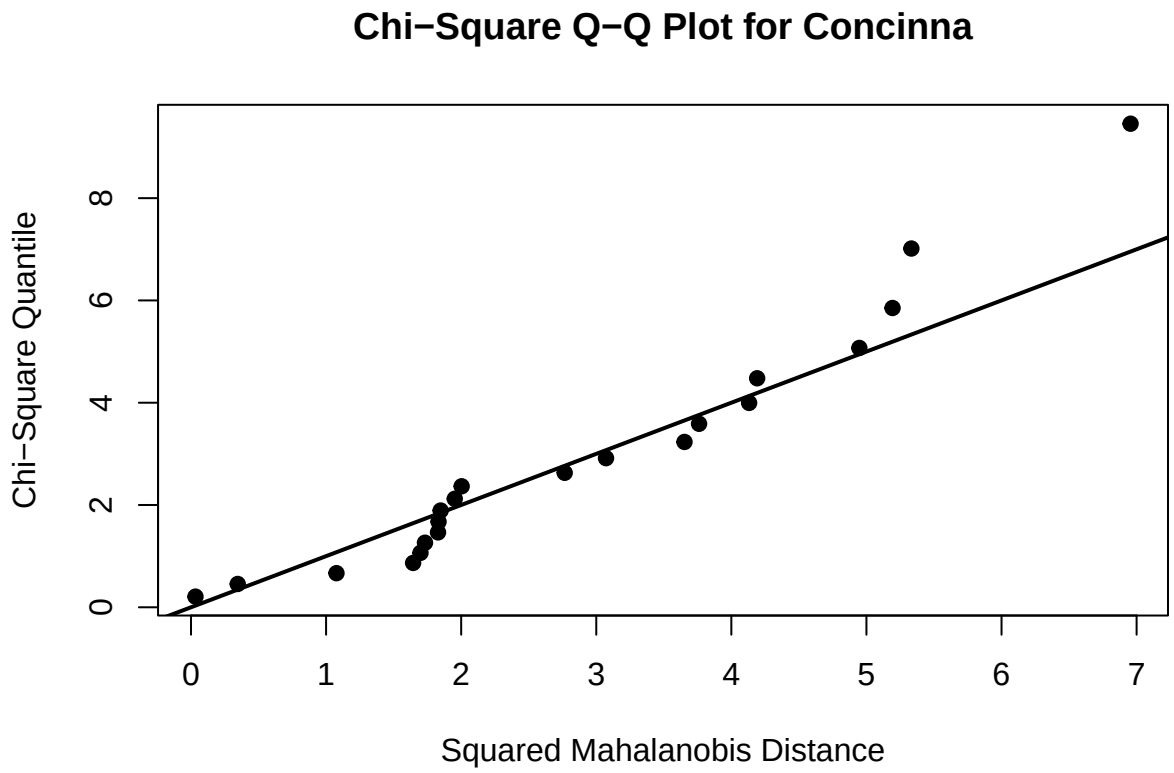
```
##       b1p       b2p       g1p       g2p    M.skew    M.kurt
##  1.175501 12.659317  6.073421 -2.340683  6.073421 -1.189687
```

```
skew.and.kurt.Mardia(flea.hept)
```

```
##        b1p        b2p        g1p        g2p     M.skew     M.kurt
##  3.1504201 15.6234918 11.5515403  0.6234918 11.5515403  0.2669633
```

Estos tests se pueden acompañar de gráficos qq-plot de ajuste a la distribución ji-cuadrado de las distancias de Mahalanobis (al cuadrado) de los datos a la media.

```
subsetData <- flea.conc
n <- dim(subsetData)[1]
p <- dim(subsetData)[2]
S <- cov(subsetData)
dif <- scale(subsetData, scale = FALSE)
D2 <- diag(dif %*% solve(S) %*% t(dif))
r <- rank(D2)
chi2q <- qchisq((r - 0.5)/n, p)
plot(D2, chi2q, pch = 19,
     main = "Chi-Square Q-Q Plot for Concinna",
     xlab = "Squared Mahalanobis Distance",
     ylab = "Chi-Square Quantile")
abline(0, 1, lwd = 2, col = "black")
```

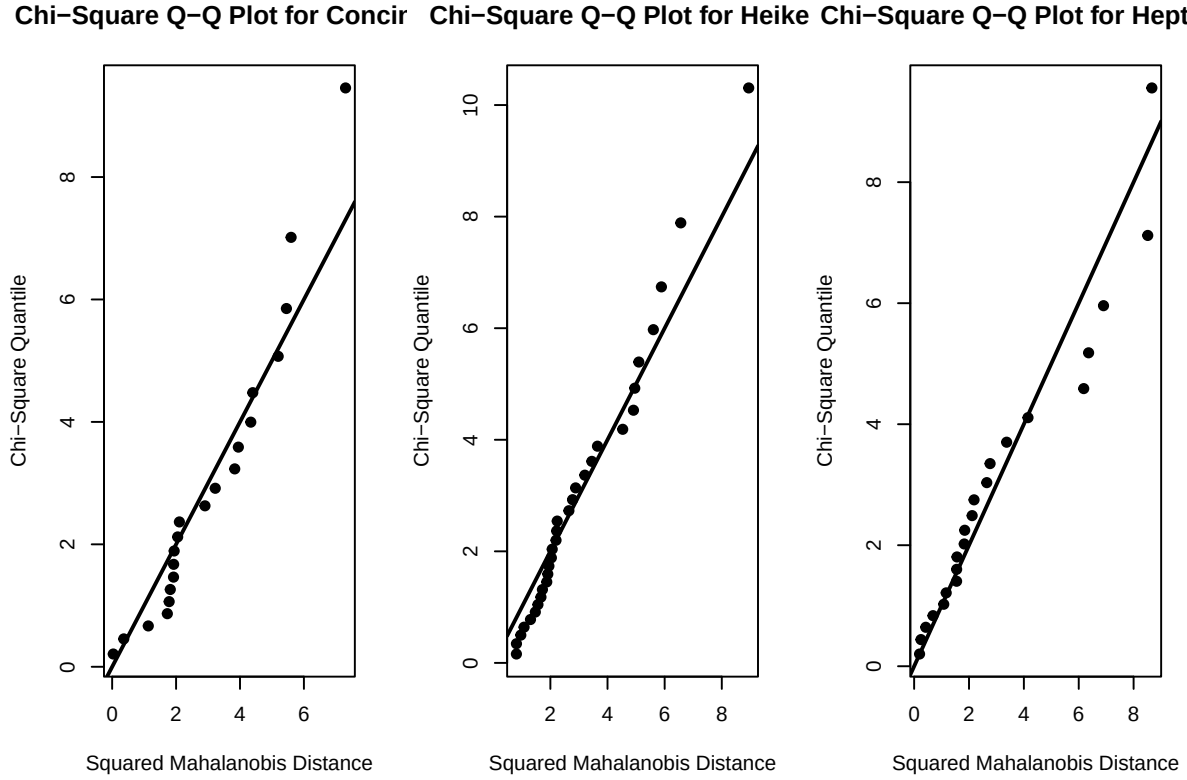


Del mismo modo para las otras dos especies.

Apartado (c)

Ahora repetimos el estudio de la normalidad multivariante con la función `mvn()` del paquete `MVN`. Los resultados son idénticos a los calculados en los apartados anteriores.

```
library(MVN)
par(mfrow=c(1,3))
mvn(flea[c(1,5:7)],"species",mvnTest = "mardia",
                             univariateTest = "SW",
                             univariatePlot = "qq",
                             multivariatePlot = "qq")
```



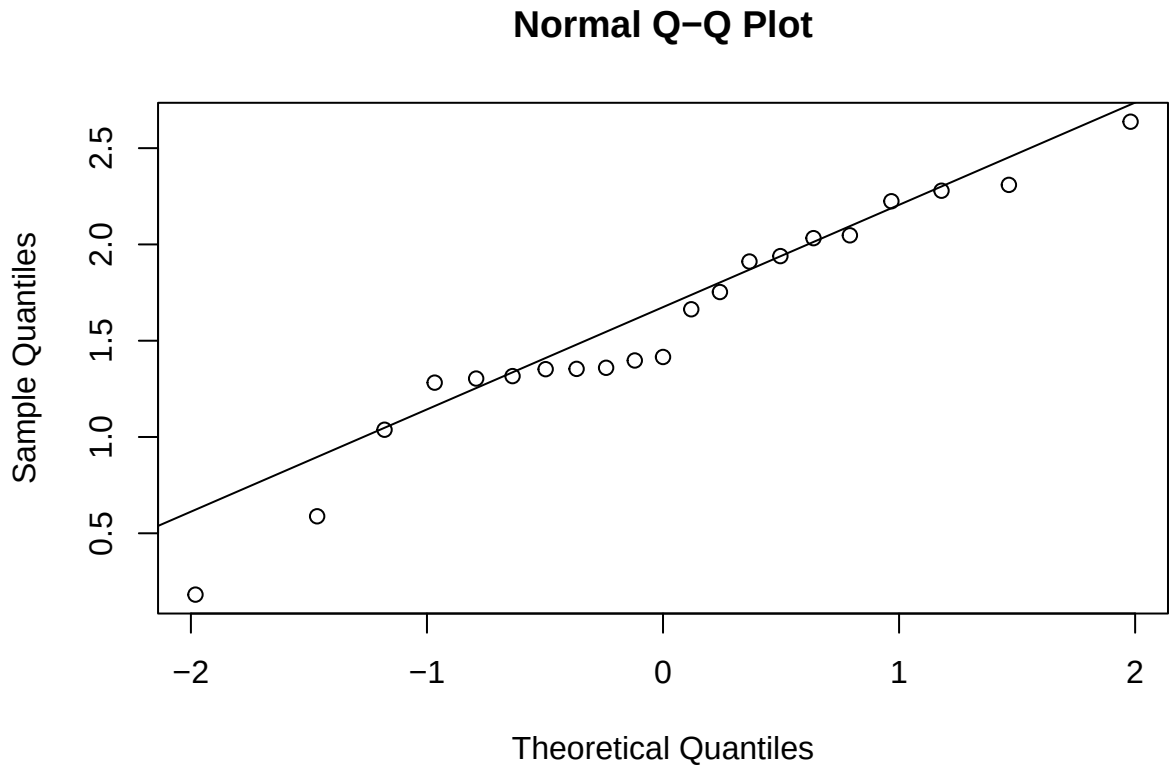

```
## $multivariateNormality
## $multivariateNormality$Concinna
##              Test         Statistic           p value Result
## 1 Mardia Skewness  4.00161895575667 0.947273919151317    YES
## 2 Mardia Kurtosis -1.10738786545985 0.268126270031477    YES
## 3             MVN              <NA>              <NA>    YES
##
## $multivariateNormality$Heikert.
##              Test        Statistic           p value Result
## 1 Mardia Skewness 6.07342106102112 0.809057445916979    YES
## 2 Mardia Kurtosis -1.1896871377758 0.234169382618608    YES
## 3             MVN             <NA>              <NA>    YES
##
## $multivariateNormality$Heptapot.
##              Test         Statistic           p value Result
## 1 Mardia Skewness   11.551540302658 0.316190290178933    YES
## 2 Mardia Kurtosis 0.266963256721424 0.789497451542426    YES
## 3             MVN              <NA>              <NA>    YES
##
##
## $univariateNormality
## $univariateNormality$Concinna
##           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
##
## $univariateNormality$Heikert.
##           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
##
## $univariateNormality$Heptapot.
##           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
##
##
## $Descriptives
## $Descriptives$Concinna
##        n      Mean   Std.Dev Median Min Max 25th 75th        Skew   Kurtosis
## aede1 21 146.19048 5.6268912    147 134 157  143  150 -0.21115421 -0.5081691
## aede2 21  14.09524 0.8890873     14  13  16   13   15  0.23476442 -1.0324544
## aede3 21 104.85714 6.1829258    105  95 116   99  110  0.07388403 -0.9658149
##
## $Descriptives$Heikert.
##        n      Mean  Std.Dev Median Min Max  25th 75th        Skew   Kurtosis
## aede1 31 124.64516 4.622758    124 116 133 120.5  129  0.07472608 -1.2755339
## aede2 31  14.29032 1.101319     14  12  16  13.5   15  0.01326813 -0.9883873
## aede3 31  81.00000 8.929352     84  55  95  74.5   88 -0.77939456  0.2821875
##
## $Descriptives$Heptapot.
##        n      Mean  Std.Dev Median Min Max   25th  75th       Skew    Kurtosis
## aede1 22 138.27273 4.142484  139.0 129 145 137.00 140.0 -0.6914117 -0.08991125
## aede2 22  10.09091 0.971454   10.0   8  12   9.25  11.0 -0.1696389 -0.74108828
## aede3 22 106.59091 5.852627  106.5  95 123 102.50 109.5  0.5630603  0.97608161
```

Apartado (d)

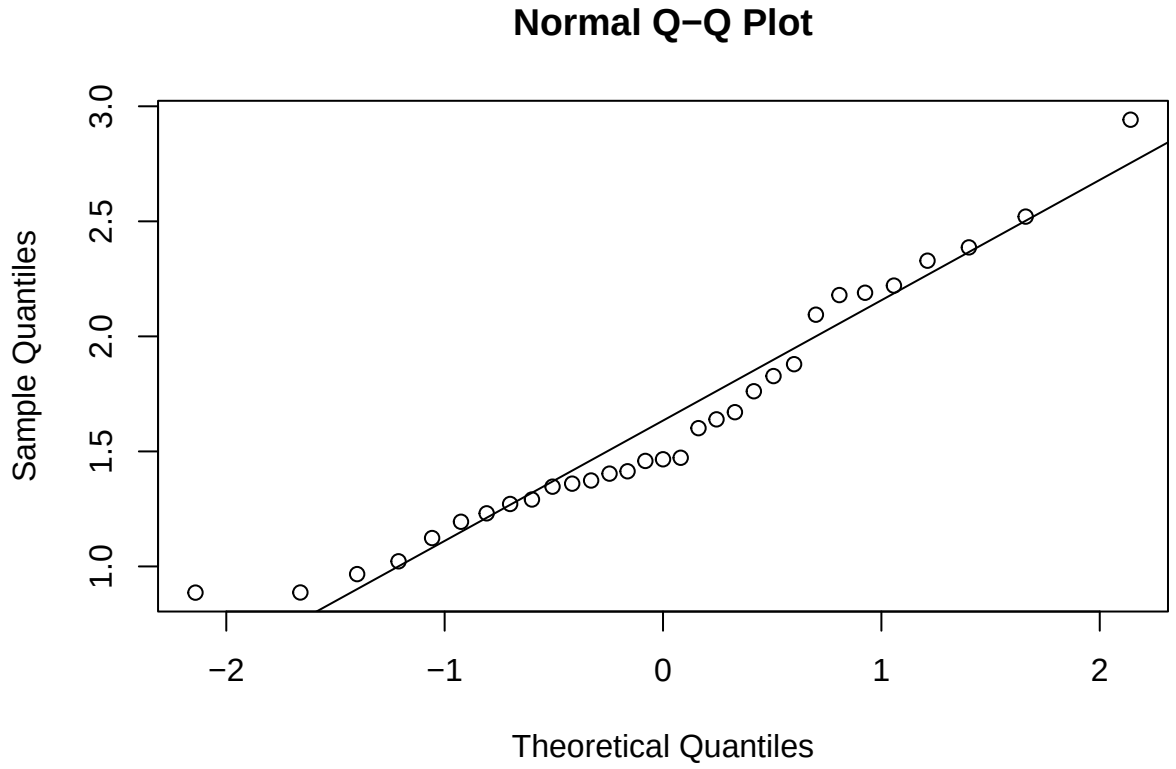
Repetimos el estudio de la normalidad multivariante con la función `mardia()` del paquete `psych`. Los resultados no son los mismos ya que esta función utiliza la matriz de covarianzas insesgada.

```
library(psych)
mardia(flea.conc)
```



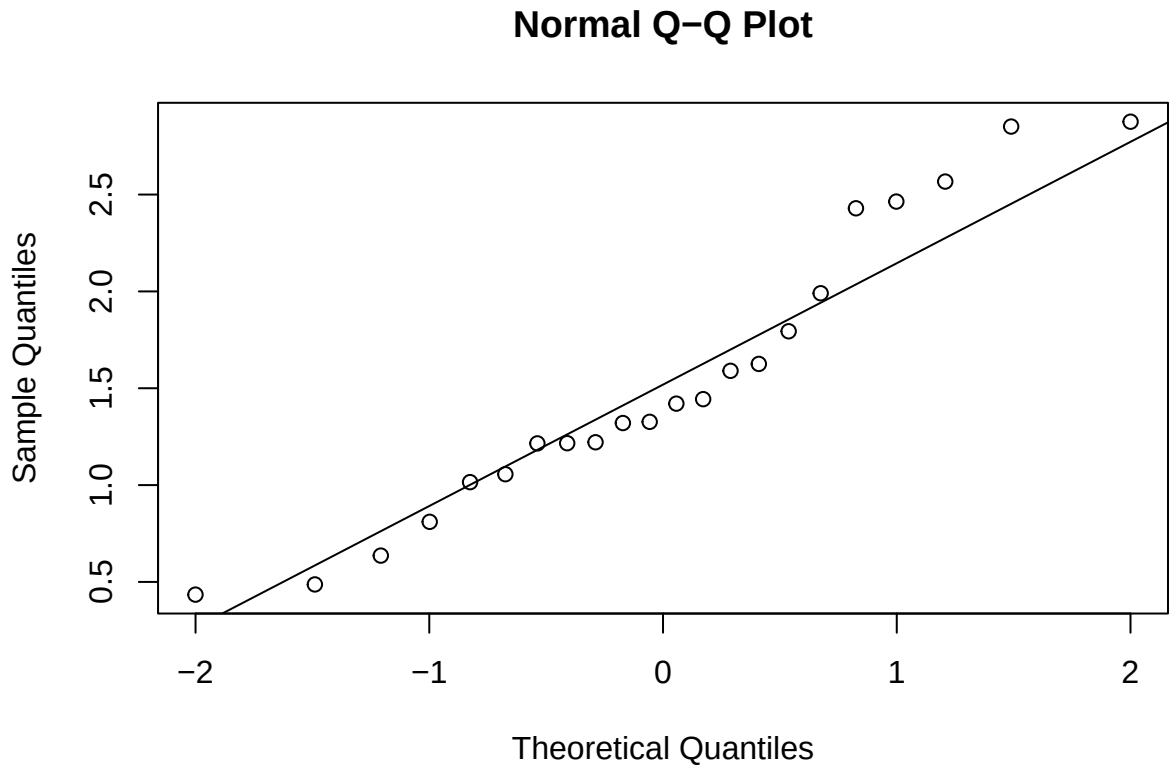

```
## Call: mardia(x = flea.conc)
##
## Mardia tests of multivariate skew and kurtosis
## Use describe(x) the to get univariate tests
## n.obs = 21   num.vars =  3
## b1p =  0.99   skew =  3.46  with probability  <=  0.97
##  small sample skew =  4.24  with probability <=  0.94
## b2p =  11.2   kurtosis =  -1.59  with probability <=  0.11
```

```
mardia(flea.heik)
```




```
## Call: mardia(x = flea.heik)
##
## Mardia tests of multivariate skew and kurtosis
## Use describe(x) the to get univariate tests
## n.obs = 31   num.vars =  3
## b1p =  1.07   skew =  5.5  with probability  <=  0.86
##  small sample skew =  6.33  with probability <=  0.79
## b2p =  11.86   kurtosis =  -1.6  with probability <=  0.11
```

```
mardia(flea.hept)
```




```
## Call: mardia(x = flea.hept)
##
## Mardia tests of multivariate skew and kurtosis
## Use describe(x) the to get univariate tests
## n.obs = 22   num.vars =  3
## b1p =  2.74   skew =  10.05  with probability  <=  0.44
##  small sample skew =  12.21  with probability <=  0.27
## b2p =  14.24   kurtosis =  -0.33  with probability <=  0.74
```

Comprobamos que con la función `mvn()` y la matriz de covarianzas insesgada se obtiene el mismo resultado.

```
mvn(flea[c(1,5:7)],"species",mvnTest = "mardia",
                             covariance = FALSE)$multivariateNormality
```

```
## $Concinna
##              Test         Statistic           p value Result
## 1 Mardia Skewness  3.45674890897888  0.96854656803215    YES
## 2 Mardia Kurtosis -1.58781883071013 0.112327312085361    YES
## 3             MVN              <NA>              <NA>    YES
##
## $Heikert.
##              Test         Statistic           p value Result
## 1 Mardia Skewness  5.50442645925181 0.855041176563168    YES
## 2 Mardia Kurtosis -1.59810709656453 0.110019144550061    YES
## 3             MVN              <NA>              <NA>    YES
##
```

```
## $Heptapot.
##              Test          Statistic           p value Result
## 1 Mardia Skewness   10.0468458624076 0.436392996429239    YES
## 2 Mardia Kurtosis -0.327358893031306 0.743396453707267    YES
## 3             MVN               <NA>              <NA>    YES
```

Apartado (e)

Comparamos las matrices de covarianzas de las tres especies con el test de la razón de verosimilitudes.

```
# Dimensiones
p <- 3 # variables
k <- 3 # grupos
tt <- as.numeric(table(flea$species))
n1 <- tt[1]; n2 <- tt[2]; n3 <- tt[3]
n <- sum(tt)
```

Estadísticos por grupo

```
medias1 <- colMeans(flea.conc); round(medias1,2)
```

```
##  aede1  aede2  aede3
## 146.19  14.10 104.86
```

```
medias2 <- colMeans(flea.heik); round(medias2,2)
```

```
##  aede1  aede2  aede3
## 124.65  14.29  81.00
```

```
medias3 <- colMeans(flea.hept); round(medias3,2)
```

```
##  aede1  aede2  aede3
## 138.27  10.09 106.59
```

```
S1 <- cov(flea.conc); round(S1,2)
```

```
##       aede1 aede2 aede3
## aede1 31.66 -0.97 15.63
## aede2 -0.97  0.79 -1.99
## aede3 15.63 -1.99 38.23
```

```
S2 <- cov(flea.heik); round(S2,2)
```

```
##       aede1 aede2 aede3
## aede1 21.37 -0.33 11.70
## aede2 -0.33  1.21  1.27
## aede3 11.70  1.27 79.73
```

```
S3 <- cov(flea.hept); round(S3,2)
```

```
##       aede1 aede2 aede3
## aede1 17.16 -0.50  7.93
## aede2 -0.50  0.94  0.28
## aede3  7.93  0.28 34.25
```

```r
# Matriz de varianzas común
S <- ((n1-1)*S1+(n2-1)*S2+(n3-1)*S3)/(n-3); round(S,2)
```

```
##       aede1 aede2 aede3
## aede1 23.02 -0.56 11.69
## aede2 -0.56  1.01  0.06
## aede3 11.69  0.06 54.59
```

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

```
##        llr    p.valor
## 13.8593416  0.3097806
```

Podemos aceptar la igualdad de varianzas.

Para contrastar la homogeneidad de las matrices de covarianzas, también podemos utilizar el test mejorado M de Box:

```r
library(heplots)
Y <- as.matrix(flea[,5:7])
boxM(Y ~ species, data=flea)
```

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

Apartado (f)

El MANOVA de un factor para comparar las medias de las tres poblaciones es

```r
g <- manova( Y ~ 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
```

El resultado del MANOVA nos indica que debemos rechazar la hipótesis nula y concluir que hay diferencias significativas entre los vectores de medias de los 3 grupos.

También se pueden utilizar otros test como el de Wilks:

```r
summary(g, 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
```

Apartado (g)

Comparación de medias dos a dos con el test T^2 de Hotelling.

Para las dos primeras poblaciones:

```
S <- ((n1-1)*S1+(n2-1)*S2)/(n1+n2-2)
D2 <- t(medias1-medias2) %*% solve(S) %*% (medias1-medias2)
D2 <- as.numeric(D2)
T2 <- n1*n2/(n1+n2) * D2
F <- (n1+n2-1-p)/((n1+n2-2)*p) * T2
p.valor <- pf(F,p,n1+n2-1-p,lower.tail=FALSE)
c(D2,T2,F,p.valor)
```

```
## [1] 2.110933e+01 2.642726e+02 8.456723e+01 3.574250e-19
```

También podemos utilizar una función específica del paquete `Hotelling`:

```
library(Hotelling)
h1 <- hotelling.test(flea.conc,flea.heik)
c(T2=h1$stats$statistic,p.valor=h1$pval)
```

```
##       T2  p.valor
## 264.2726   0.0000
```

```
h2 <- hotelling.test(flea.conc,flea.hept)
c(T2=h2$stats$statistic,p.valor=h2$pval)
```

```
##      T2 p.valor
## 259.715   0.000
```

```
h3 <- hotelling.test(flea.heik,flea.hept)
c(T2=h3$stats$statistic,p.valor=h3$pval)
```

```
##       T2  p.valor
## 420.3363   0.0000
```

Según los p-valores obtenidos, las tres medias son distintas entre sí.

Cuando se hacen comparaciones múltiples siempre deberíamos hacer alguna corrección de los p-valores, como la de Bonferroni.

```
p <- c(h1$pval,h2$pval,h3$pval)
p.adjust(p,method = "bonferroni")
```

```
## [1] 0 0 0
```

En este caso las diferencias son tan grandes que el ajuste no modifica la significación de ninguno de los contrastes.

Ejercicio 2 (25 pt.)

Apartado (a)

En primer lugar vamos a reducir los datos a las dos especies consideradas y a las tres primeras variables numéricas (y la especie):

```
insect <- droplevels(flea[flea$species %in% c("Concinna","Heikert."),1:4])
table(insect$species)
```

```
##
## Concinna Heikert.
##       21       31
```

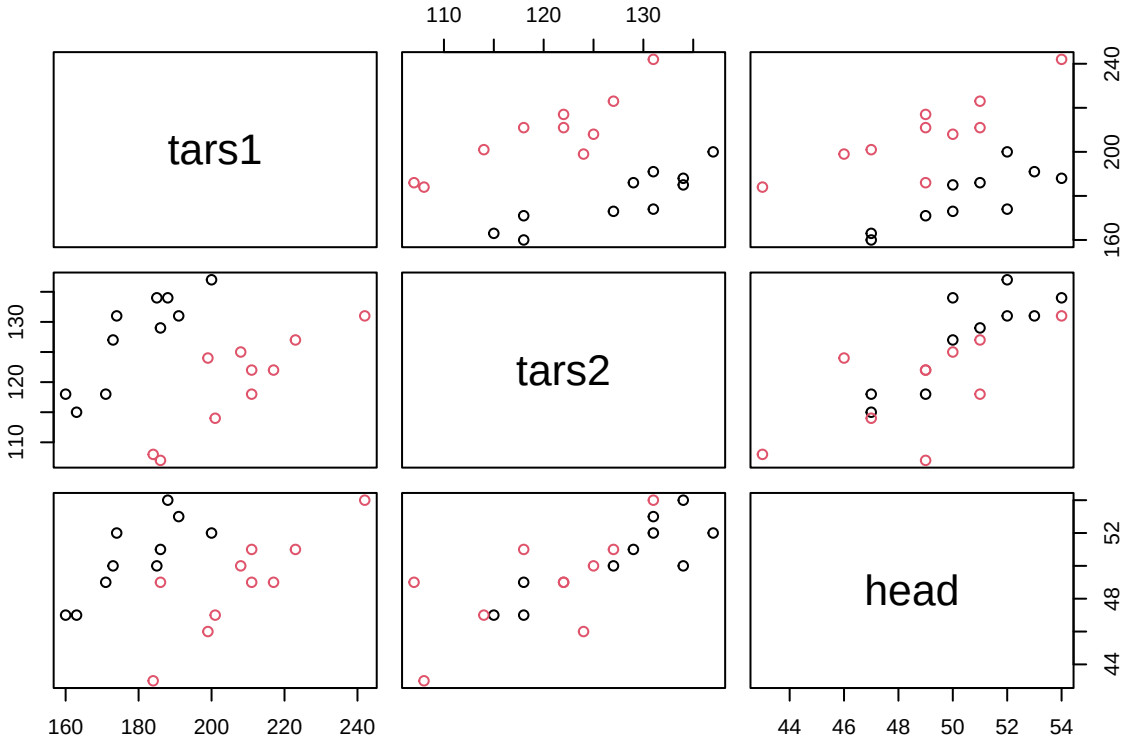
Para no tener dificultades más adelante, hemos eliminado del factor `species` la especie que no interviene en este ejercicio.

Ahora seleccionamos los 10 primeros insectos de cada una de las dos especies.

```
idx <- c(rep(TRUE,10),rep(FALSE,11),rep(TRUE,10),rep(FALSE,21))
insect20 <- insect[idx,-1]
insect20$especie <- factor(c(rep("a",10),rep("b",10)))
```

También hemos renombrado las dos especies como “a” y “b”.

```
pairs(insect20[,1:3], col=insect20$especie)
```



Apartado (b)

Test para contrastar la homogeneidad de las matrices de varianzas-covarianzas de las dos especies (test de Bartlett):

```
# Matrices de varianzas-covarianzas
attach(insect20)
n.i <- table(especie)
n <- sum(n.i)
p <- ncol(insect20)-1
S.a <- cov(as.matrix(insect20[especie=="a",-4]))
S.b <- cov(as.matrix(insect20[especie=="b",-4]))
S <- ((n.i[1]-1)*S.a+(n.i[2]-1)*S.b)/(n-2)
# Comparación de matrices de covarianzas
llr <- n*log(det(S))-n.i[1]*log(det(S.a))-n.i[2]*log(det(S.b))
llr <- as.numeric(llr)
alpha <- 0.05
chi2.cv <- qchisq(1-alpha,p*(p+1)/2)
p.val <- pchisq(llr, p*(p+1)/2, lower.tail=FALSE)
cbind(llr,chi2.cv,p.val)
```

```
##           llr  chi2.cv     p.val
## [1,] 5.767907 12.59159 0.4496852
```

Entonces el análisis lineal discriminante está justificado ya que no hay motivos para rechazar la igualdad de las matrices de varianzas-covarianzas de los dos grupos.

Apartado (c)

Vamos a calcular el análisis LDA:

```
library(MASS)
g <- lda(especie ~ .,data=insect20); g
```

```
## Call:
## lda(especie ~ ., data = insect20)
##
## Prior probabilities of groups:
##   a   b
## 0.5 0.5
##
## Group means:
##   tars1 tars2 head
## a 179.1 127.4 50.5
## b 208.2 119.8 48.9
##
## Coefficients of linear discriminants:
##              LD1
## tars1  0.1588694
## tars2 -0.1981674
## head  -0.3457517
```

```
cls <- predict(g)
table(cls$class,insect20$especie)
```

```
##
##      a  b
##   a 10  0
##   b  0 10
```

Parece un discriminador perfecto (pocos datos).

Apartado (d)

Las probabilidades a posteriori son:

```
cls$posterior
```

```
##               a            b
## 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
```

Muestran valores de clasificación muy elevados para cada clase.

Apartado (e)

```
predict(g, newdata = insect[!idx,-1])$class
```

```
##  [1] a a a a a a a a a a b b b b b b a b b b b b a b b a b b b b a
## Levels: a b
```

La tabla de validación es:

```
table(insect$species[!idx], predict(g, newdata = insect[!idx,-1])$class)
```

```
##
```

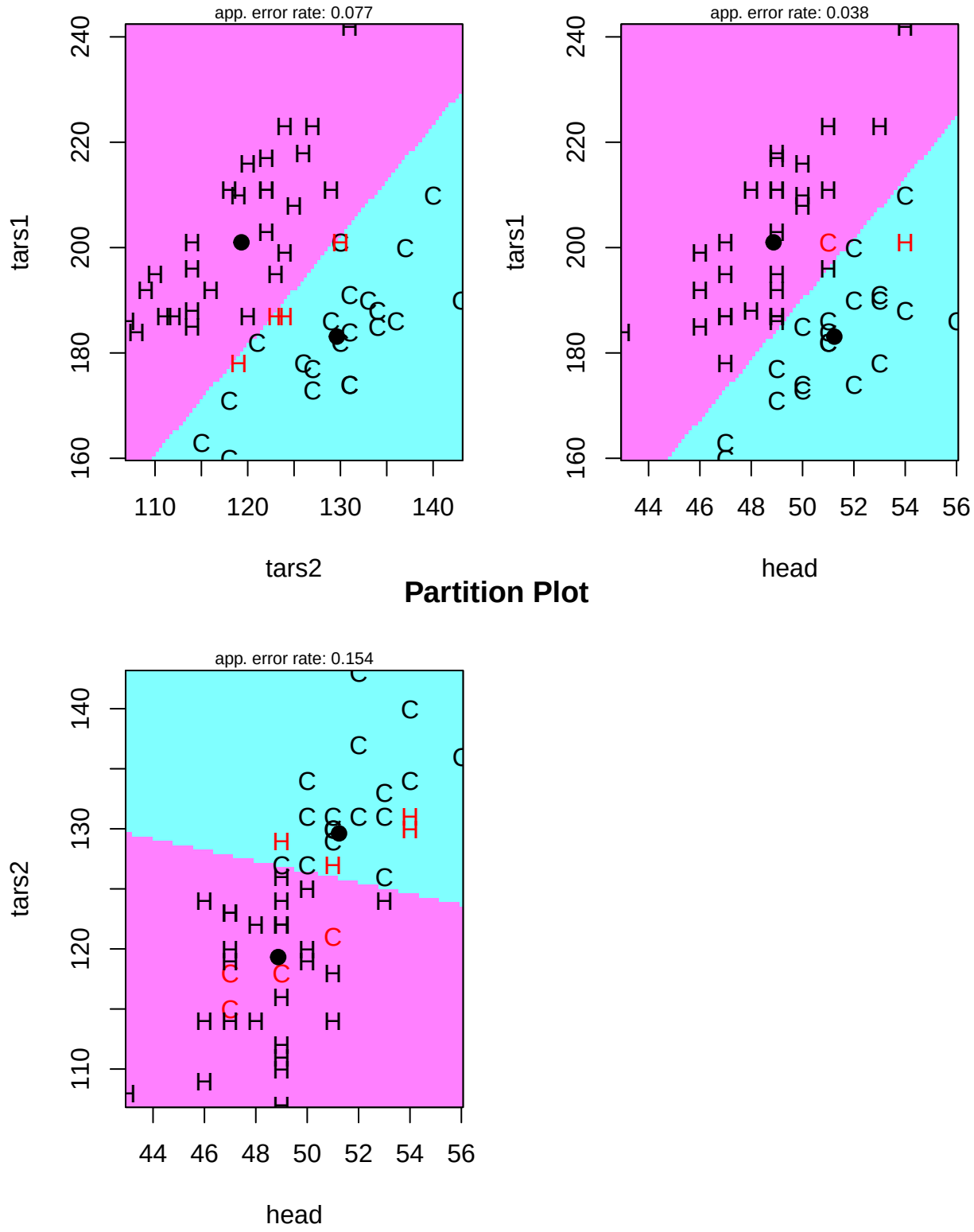
```
##             a  b
##   Concinna 11  0
##   Heikert.  4 17
```

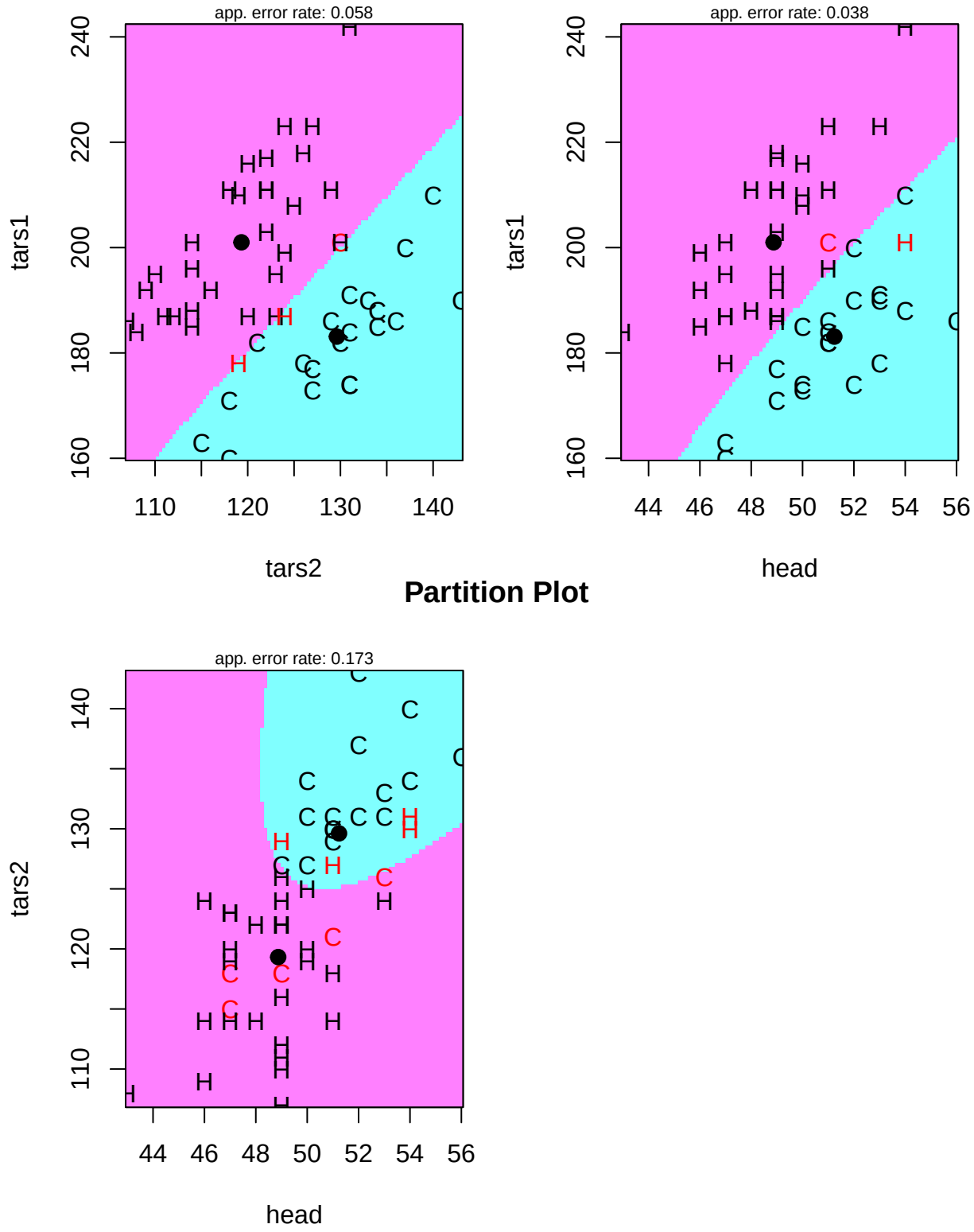
Únicamente hay 4 mal clasificados.

```
# detach(insect20)
```

Apartado (f)

```
library(klaR)
partimat(species ~ ., data=insect)
partimat(species ~ ., method="qda", data=insect)
```


app. error rate: 0.077
app. error rate: 0.038
app. error rate: 0.154
tars1
tars2
head
Partition Plot


app. error rate: 0.058
app. error rate: 0.038
app. error rate: 0.173
tars1
tars2
head
Partition Plot

Ejercicio 3 (35 pt.)

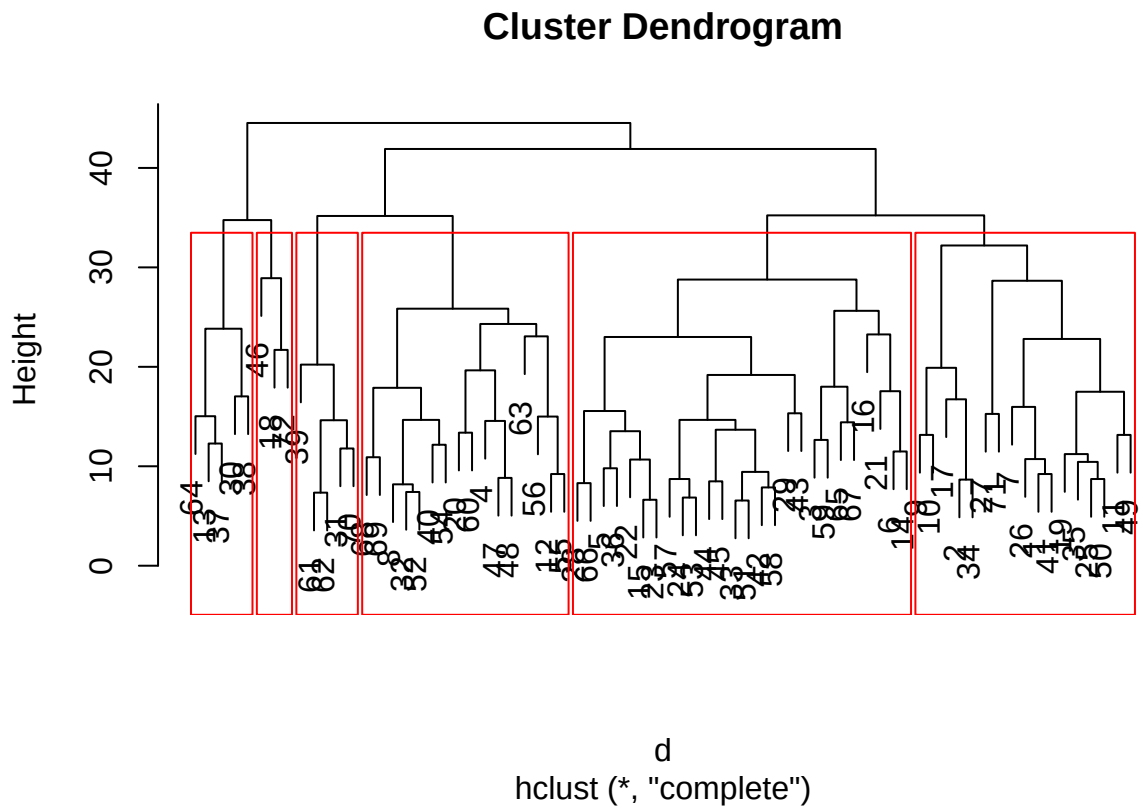
Los datos son:

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

Apartado (a)

Análisis jerárquico aglomerativo con la distancia euclídea y *complete linkage*:

```
d <- dist(wood, method = "euclidean") # distance matrix
fit <- hclust(d, method="complete")
plot(fit) # display dendogram
groups <- cutree(fit, k=6) # cut tree into 5 clusters
# draw dendogram with red borders around the 5 clusters
rect.hclust(fit, k=6, border="red")
```



Apartado (b)

ANOVA para cada especie:

```
mi.tabla <- data.frame(F.est=numeric(13), p.value=numeric(13), signif=logical(13))
rownames(mi.tabla) <- colnames(wood)
for(i in 1:13){
g <- aov(wood[,i] ~ factor(groups))
resumen <- summary(g)
```

```
mi.tabla[i,1] <- resumen[[1]][1,4]           # F.est
mi.tabla[i,2] <- resumen[[1]][1,5]           # p.value
mi.tabla[i,3] <- resumen[[1]][1,5] < 0.05/13 # Bonferroni correction
}
mi.tabla
```

```
##             F.est      p.value signif
## carcar 62.9352806 8.808529e-24   TRUE
## corflo  1.5486533 1.869870e-01  FALSE
## faggra  7.1065090 2.271904e-05   TRUE
## ileopa  3.4226847 8.244588e-03  FALSE
## liqsty  5.8660589 1.524447e-04   TRUE
## maggra  3.9713488 3.287522e-03   TRUE
## nyssyl  1.6598089 1.566971e-01  FALSE
## ostvir 17.7020897 4.354362e-11   TRUE
## oxyarb  1.4179141 2.293756e-01  FALSE
## pingla  0.4323609 8.244457e-01  FALSE
## quenig  2.2327317 6.122327e-02  FALSE
## quemic  4.1225263 2.556603e-03   TRUE
## symtin 75.5714240 5.790540e-26   TRUE
```

Medias para cada cluster de cada una de las especies con diferencias significativas:

```
wood.sig <- wood[,mi.tabla$signif]
mi.tabla2 <- matrix(numeric(7*6), ncol=6)
colnames(mi.tabla2) <- as.character(1:6)
row.names(mi.tabla2) <- names(wood.sig)
for(i in 1:7){
mi.tabla2[i,] <- tapply(wood.sig[,i], factor(groups), mean)
}
round(mi.tabla2,2)
```

```
##            1     2     3    4     5    6
## carcar  1.24  3.85 18.50  8.2  6.00 24.4
## faggra  5.94 11.38  5.94  8.6  2.67  6.4
## liqsty  6.76  7.19  6.44  6.6 18.00 17.4
## maggra  3.24  5.27  2.75  4.6  0.67  3.8
## ostvir 13.82  4.27  2.88  3.6 14.00  2.8
## quemic  4.12  5.27  9.38  7.0  2.33  5.2
## symtin  2.00  0.92  0.69 18.0 20.00  0.0
```

Observamos que en cada cluster hay una o dos especies con mayor media. Eso caracteriza los clusters:

Cluster 1: principalmente `ostvir`.

Cluster 2: principalmente `faggra`.

Cluster 3: principalmente `carcar`.

Cluster 4: principalmente `symtin`.

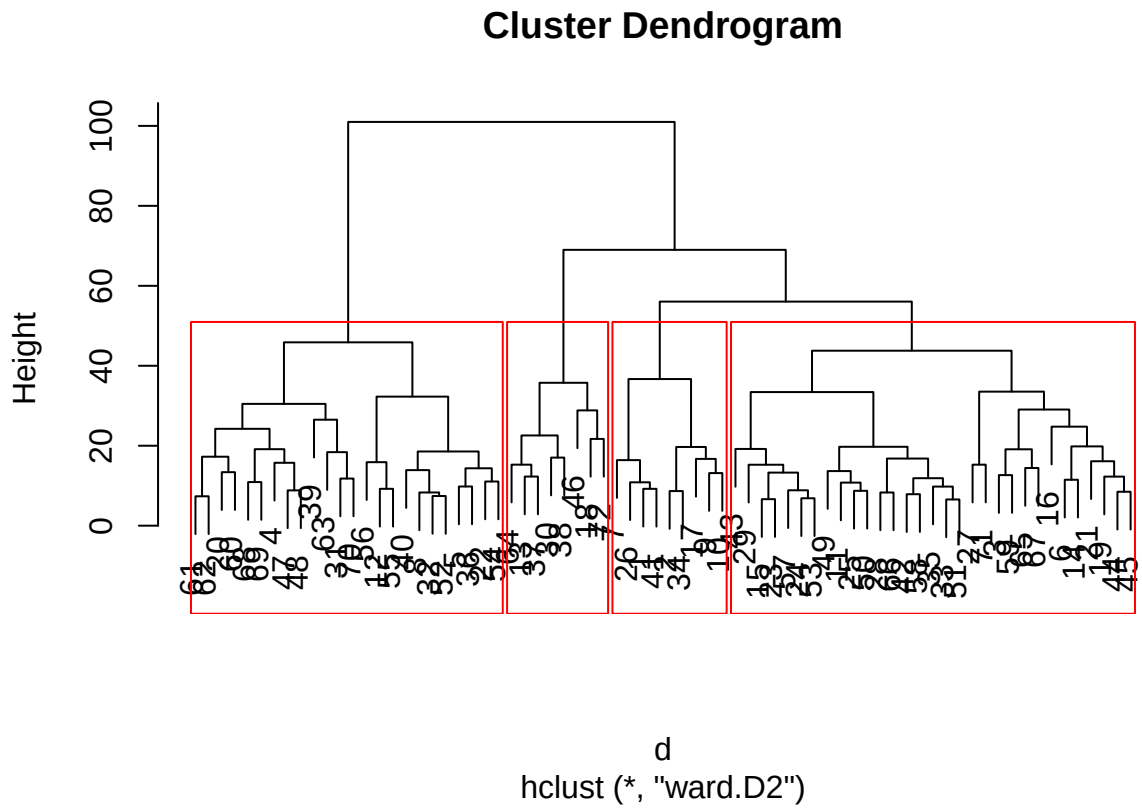
Cluster 5: `liqsty`, `ostvir` y `symtin`.

Cluster 6: `carcar` y `liqsty`.

Apartado (c)

Análisis jerárquico aglomerativo con la distancia euclídea y el método de Ward:

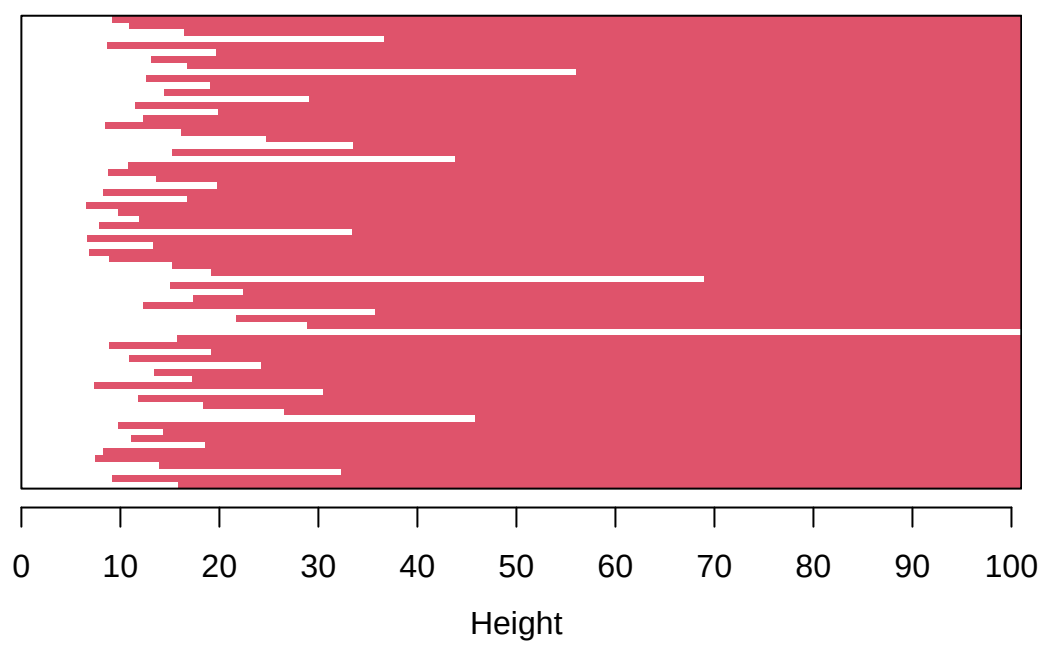
```
# Ward Hierarchical Clustering
fit <- hclust(d, method="ward.D2")
plot(fit) # display dendogram
# draw dendogram with red borders around the 4 clusters
rect.hclust(fit, k=4, border="red")
```



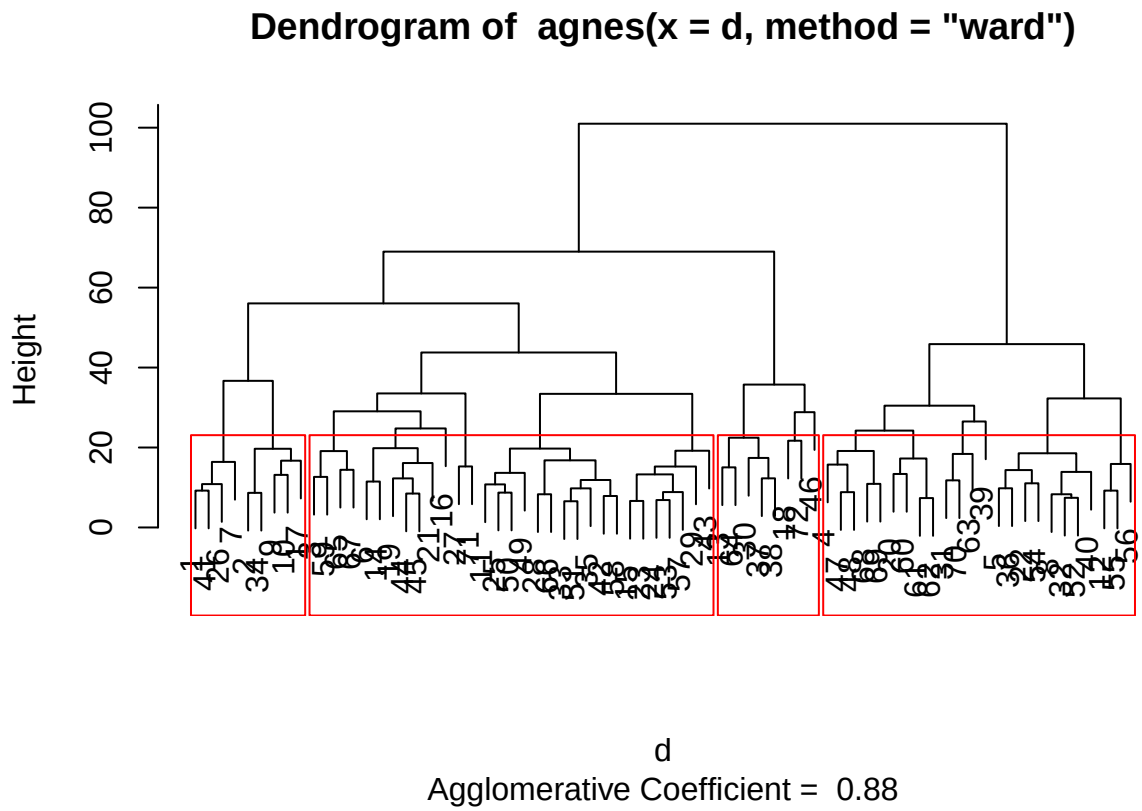

```
groups <- cutree(fit, k=4) # cut tree into 4 clusters
```

Con la función `agnes()`:

```
library(cluster)
fit <- agnes(d, method = "ward")
plot(fit)
```




```
rect.hclust(fit, k=4, border="red")
```




```
groups <- cutree(fit, k=4) # cut tree into 4 clusters
```

ANOVA para cada especie:

```
mi.tabla <- data.frame(F.est=numeric(13), p.value=numeric(13), signif=logical(13))
rownames(mi.tabla) <- colnames(wood)
for(i in 1:13){
g <- aov(wood[,i] ~ factor(groups))
resumen <- summary(g)
mi.tabla[i,1] <- resumen[[1]][1,4]           # F.est
mi.tabla[i,2] <- resumen[[1]][1,5]           # p.value
mi.tabla[i,3] <- resumen[[1]][1,5] < 0.05/13 # Bonferroni correction
}
mi.tabla
```

```
##              F.est      p.value signif
## carcar  67.4184552 2.438082e-20   TRUE
## corflo   2.3136244 8.367428e-02  FALSE
## faggra   7.1282769 3.113445e-04   TRUE
## ileopa   5.3770140 2.214255e-03   TRUE
## liqsty   0.7628793 5.188003e-01  FALSE
## maggra   2.7492843 4.941259e-02  FALSE
## nyssyl   1.3592387 2.626618e-01  FALSE
## ostvir  32.9108904 2.951004e-13   TRUE
## oxyarb   3.1530369 3.036667e-02  FALSE
```

```
## pingla   1.0322235 3.839379e-01  FALSE
## quenig   2.3944723 7.588133e-02  FALSE
## quemic   3.4385131 2.155483e-02  FALSE
## symtin 120.9470720 3.352067e-27   TRUE
```

Medias para cada cluster de cada una de las especies con diferencias significativas:

```
wood.sig <- wood[,mi.tabla$signif]
mi.tabla2 <- matrix(numeric(5*4), ncol=4)
colnames(mi.tabla2) <- as.character(1:4)
row.names(mi.tabla2) <- names(wood.sig)
for(i in 1:5){
mi.tabla2[i,] <- tapply(wood.sig[,i], factor(groups), mean)
}
round(mi.tabla2,2)
```

```
##            1     2     3     4
## carcar  1.00  2.77 18.50  7.38
## faggra  5.89 10.58  5.96  6.38
## ileopa 12.33  7.55  4.29  7.88
## ostvir 18.33  5.35  3.12  7.50
## symtin  1.44  1.29  0.67 18.75
```

También con esta tabla podemos caracterizar los clusters con la abundancia de las especies en cada uno de ellos.

Apartado (d)

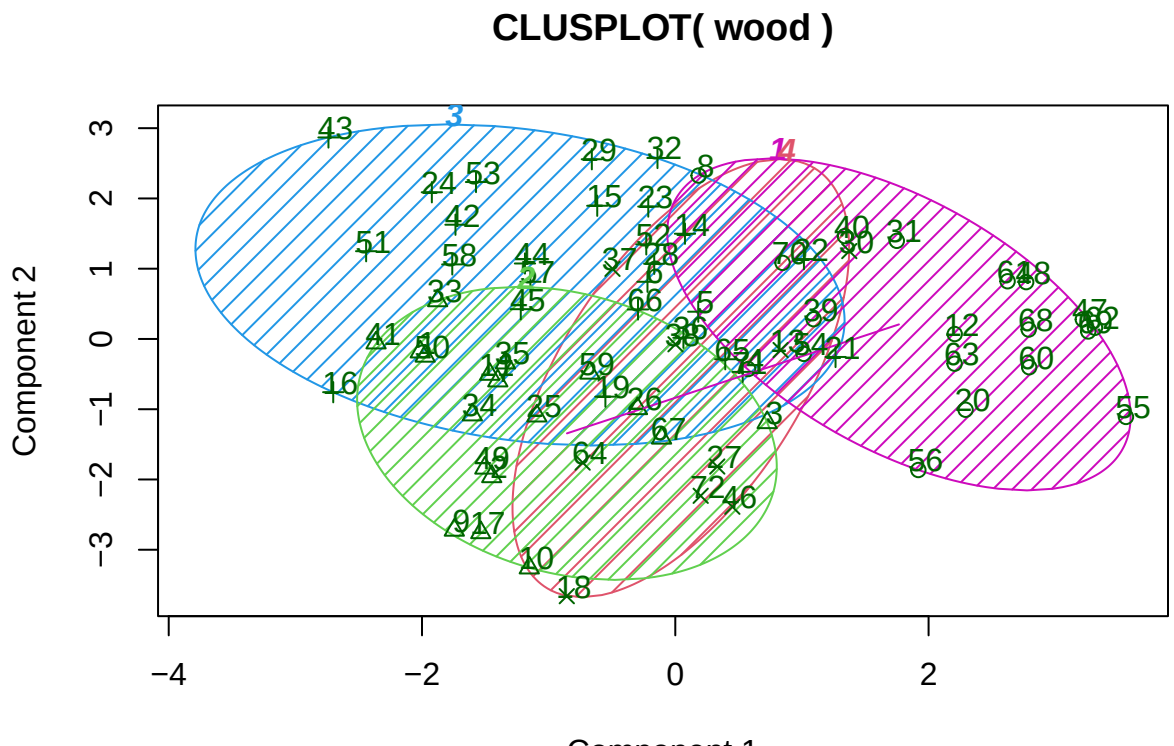
Particionado de las k-medias con el algoritmo de Hartigan-Wong:

```
# Inicializar la semilla aleatória
set.seed(123)
# Formar 4 grupos para k-means
km.4 <- kmeans(wood, centers=4, algorithm="Hartigan-Wong"); km.4
```

```
## K-means clustering with 4 clusters of sizes 19, 18, 26, 9
##
## Cluster means:
##      carcar   corflo    faggra    ileopa    liqsty   maggra    nyssyl    ostvir
## 1 20.736842 1.263158  5.473684  4.842105  9.526316 2.631579 2.5789474  2.842105
## 2  1.555556 2.722222  7.166667 12.666667  7.277778 3.833333 1.7777778 13.555556
## 3  4.500000 2.307692 11.115385  4.653846  6.423077 5.153846 0.9615385  3.961538
## 4  6.555556 2.444444  5.888889  7.777778 11.111111 3.000000 1.2222222  7.222222
##      oxyarb   pingla   quenig   quemic     symtin
## 1 0.2631579 2.473684 3.578947 8.789474  0.5789474
## 2 2.2222222 2.222222 1.166667 5.388889  1.1666667
## 3 1.3076923 2.692308 1.346154 4.461538  1.0384615
## 4 0.3333333 4.555556 1.555556 5.000000 17.7777778
##
## Clustering vector:
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  2  2  2  1  3  3  2  1  2  2  2  1  4  3  3  3  2  4  3  1  3  3  3  3  2  2
```

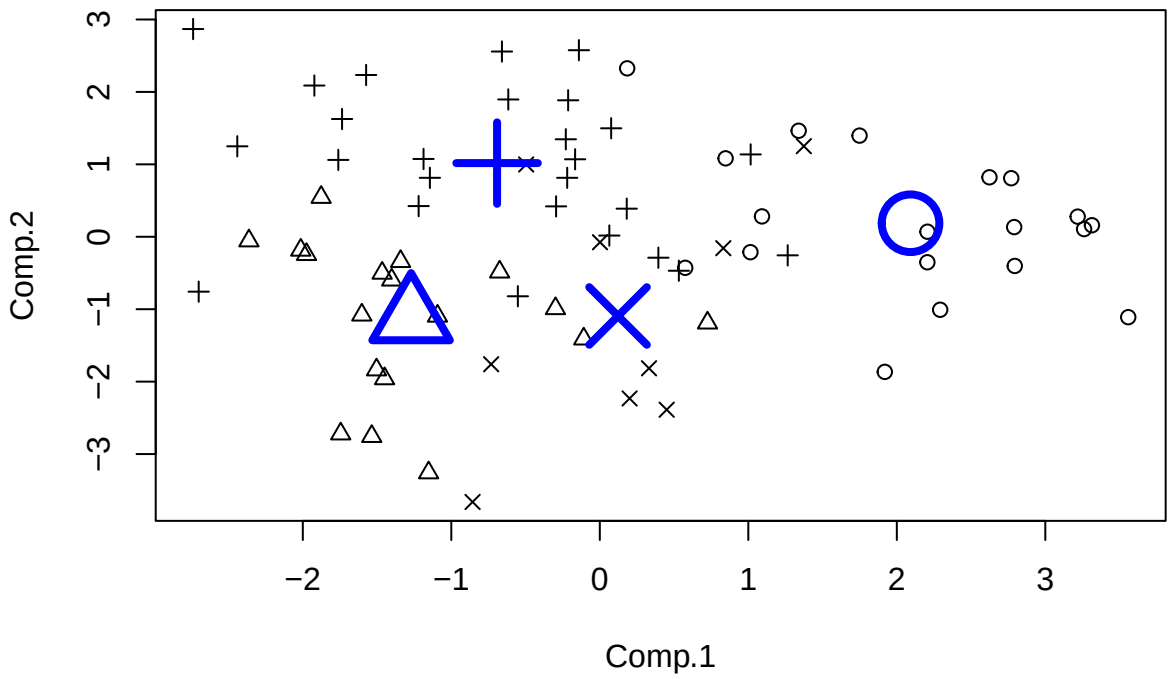
```
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
##  4  3  3  4  1  3  2  2  2  3  4  4  1  1  2  3  3  3  3  4  1  1  2  2  3  3
## 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
##  3  1  1  1  3  3  2  1  1  1  1  4  3  3  2  1  1  1  3  4
##
## Within cluster sum of squares by cluster:
## [1] 3277.789 2773.722 3616.615 2256.889
##  (between_SS / total_SS =  44.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
clusplot(wood, km.4$cluster, color=TRUE, shade=TRUE,
    labels=2, lines=1)
```



These two components explain 35.63 % of the point variability.

```
# Gráfico equivalente con los puntos centrales de cada cluster
pca.km <- princomp(wood, cor=T)
plot(pca.km$scores[,1:2], pch=km.4$cluster)
points(predict(pca.km, km.4$centers)[,1:2], pch = 1:4, cex = 4, lwd = 4,  col = "blue")
```



Las sumas de cuadrados dentro de cada grupo:

```
km.4$withinss
```

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

Los tamaños de los conglomerados:

```
km.4$size
```

```
## [1] 19 18 26  9
```

Apartado (e)

Particionado con los k-medoides o PAM:

```
pam.4 <- pam(wood,4,diss=FALSE); pam.4
```

```
## Medoids:
##    ID carcar corflo faggra ileopa liqsty maggra nyssyl ostvir oxyarb pingla
## 50 50      2      3      8      8      6      5      1     11      4      5
## 47 47     27      1      3      1     11      3      5      2      0      1
## 15 15      7      4     10      0      4      7      0      2      0      0
## 37 37      6      2     11      5      5      7      2      3      0      1
##    quenig quemic symtin
## 50      0      1      1
## 47      4     10      3
## 15      1      5      1
## 37      0      6     17
## Clustering vector:
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
```

```
##  1  1  1  2  3  3  1  3  1  1  1  3  4  3  3  1  1  4  1  2  3  3  3  3  1  1
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
##  1  3  3  4  2  3  1  1  1  3  4  4  2  2  1  1  1  3  3  4  2  2  1  1  1  3
## 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
##  3  3  2  1  3  1  1  2  2  2  2  4  1  1  1  2  2  2  1  1
## Objective function:
##    build     swap
## 13.95399 13.51838
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

```
table(pam.4$clustering, km.4$clus)
```

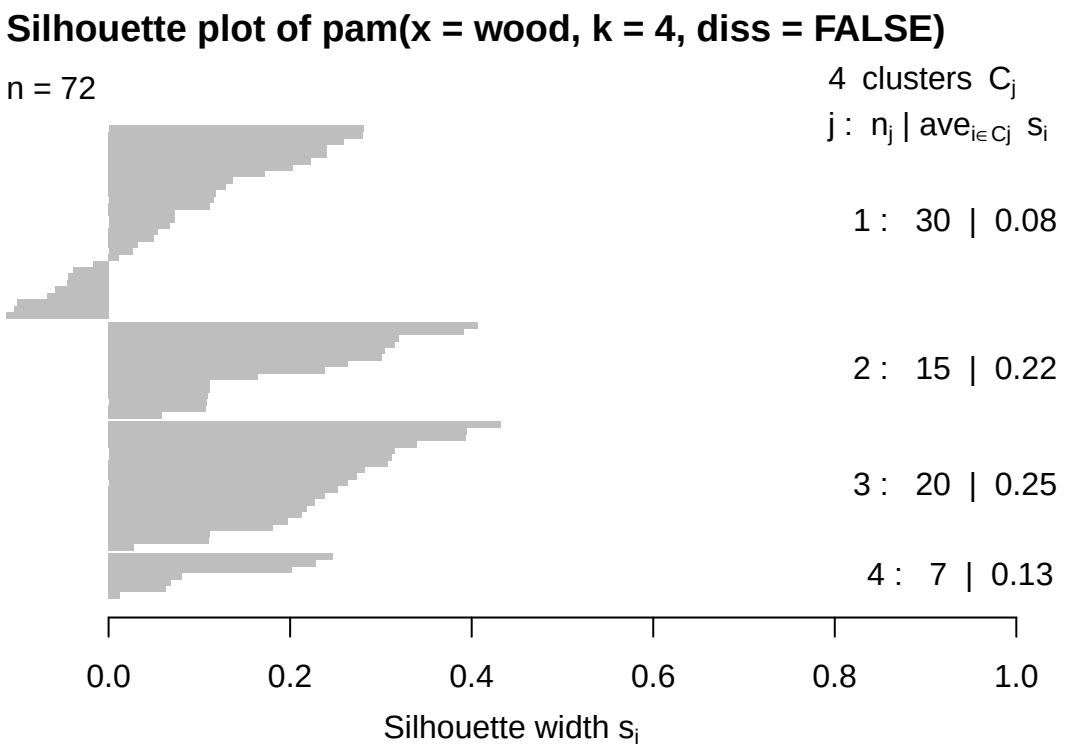
```
##
##      1  2  3  4
##   1  1 18  9  2
##   2 15  0  0  0
##   3  3  0 17  0
##   4  0  0  0  7
```

```
pam.4$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
```

Los medoides son las observaciones 50, 47, 15 y 37.

```
plot(silhouette(pam.4))
```



Average silhouette width : 0.16

Algunos elementos del primer cluster aparecen allí con poca convicción (valor de silueta negativo).