Estadística Multivariante - PEC 2

Alejandro Keymer

Ejercicio 1 (70 pt.)

(a)

Después de reducir la base de datos original a las $p = 13$ especies destacadas, nos planteamos elegir una distancia adecuada para el análisis de conglomerados. Con este tipo de datos se suele utilizar la disimilaridad de Bray-Curtis con la siguiente fórmula según la Wikipedia:

$$d_{BC}^{(1)}(i,j) = 1 - \frac{2C_{ij}}{S_i + S_j}$$

donde C_{ij} es la suma del menor valor para cada una de las especies en común de las dos muestras. S_i y S_j son el total de especímenes en cada una de las muestras. También se puede calcular con la fórmula que utiliza la función `vegdist()` del paquete `vegan`

$$d_{BC}^{(2)}(i,j) = \frac{\sum_{k=1}^{p} |x_{ik} - x_{jk}|}{\sum_{k=1}^{p} (x_{ik} + x_{jk})}$$

Calcular la disimilaridad de Bray-Curtis entre las dos primeras muestras de la base de datos con ambas fórmulas (sin utilizar la función `vegdist()`). En la primera fórmula nos puede ayudar la función `pmin()` de R. Calcular a continuación la matriz de disimilaridades completa con la función `vegdist()` y comprobamos que el valor hallado para las dos primeras muestras coincide:

En primer lugar se calculan los datos y se crean nombres de fila con las coordenadas de cada muestra:

```
species <- c('carcar','corflo','faggra', 'ileopa','liqsty','maggra',
             'nyssyl','ostvir','oxyarb', 'pingla','quenig','quemic','symtin')

df <-
    read.table("wood.dat", header=T) %>%
    select(x,y,species) %>%
    unite(rowname, x, y, sep = "-") %>%
    column_to_rownames()
```

Con la base hacemos el cálculo *manual* con las fórmulas descritas utilizando las dos primeras muestras:

```
# formula 1
(d_bc_1 <- 1 - (2 * sum(pmin(df[1,], df[2,]))) / (sum(df[1,]) + sum(df[2,])))
```

```
## [1] 0.3181818
```

```
# formula 2
(d_bc_2 <- sum(abs(df[1,] - df[2,])) / sum(df[1,] + df[2,]))
```

```
## [1] 0.3181818
```

A continuación se realiza el cálculo de la matriz de disimilaridad utilizando la función `vegdist` que, por defecto, utiliza la distancia de Bray–Curtis. Finalmente revisamos que la primera distancia, entre las dos primeras muestras, coincide con las calculadas manualmente.

```r
df_dist <- vegdist(df)
df_dist[1]
```

```
## [1] 0.3181818
```

Se puede ver que las tres soluciones dan el mismo valor para la distancia entre las primeras dos muestras.

(b)

Sin embargo, la disimilaridad de Bray-Curtis no verifica la desigualdad triangular de forma que no es una distancia. Comprobar este hecho, tal vez con algún contra ejemplo entre dos muestras o con alguna función de R que podamos hallar. Para corregir este problema podemos transformar la disimilaridad de Bray-Curtis calculando la raíz cuadrada de sus valores. Comprobar que dicha raíz cuadrada verifica la desigualdad triangular.

Para verificar la desigualdad triangular se puede utilizar la función `tri.ineq` de la librería `fossil`. Con esta función podemos corroborar que la distancia de Bray-Curtis calculada **no** verifica la desigualdad triangular, pero la raíz cuadrada de las distancias, si lo hacen. Al examinar la función en sí, es posible modificar el procedimiento por el que se verifica la desigualdad triangular, y encontrar un contra ejemplo especifico. La fórmula busca si hay casos en que se cumpla que el "cateto" mas grande, sea mayor a la suma de los otros dos "catetos", lo que implica que no es un triángulo.

```r
tri.ineq(df_dist)
```

```
## [1] FALSE
```

```r
# función modificada para encontrar contra ejemplo.
# Devuelve el primer contra ejemplo que encuentra.
mat <- as.matrix(df_dist)
    n <- dim(mat)[1]
    for (i in 1:(n - 2)) {
        for (j in (i + 1):(n - 1)) {
            for (k in (j + 1):n) {
                sds <- c(mat[j, i], mat[k, i], mat[k, j])
                lng <- max(sds)
                if (lng > (sum(sds) - lng)){
                  return()}
            }
        }
    }


# no se satisface para las muestras:
rownames(mat)[j]
```

```
## [1] "2-7"
```

```r
rownames(mat)[i]
```

```
## [1] "1-2"
```

```r
rownames(mat)[k]
```

```
## [1] "5-2"
```

En cambio, para la distancia al cuadrado podemos ver que si se cumple.

```r
# cuadrado de las distancias
sq_dist <- sqrt(df_dist)
tri.ineq(sq_dist)
```

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

(c)

Muchos procedimientos de Análisis multivariante requieren que la distancia utilizada sea euclídea. Comprobar que la raíz cuadrada de la disimilaridad de Bray-Curtis es una distancia euclídea.

Es posible crear una función con las fórmulas del apartado 8.2 del libro de *C.M. Cuadras*, para poder valorar si la diferencia corresponde a una distancia *euclidea*.

```r
is_euclidean <- function(diss){
    mat <- as.matrix(diss)
    n <- dim(mat)[1]
    A <- mat * -(1/2)
    H <- diag(n) - (1/n) * (rep(1,n) %*% t(rep(1,n)))
    B <- H %*% A %*% H

    # prueba
    eig <- eigen(B)
    all(eig$values >= 0)
}


# probamos con la raíz cuadrada de la distancia
is_euclidean(sq_dist)
```

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

(d)

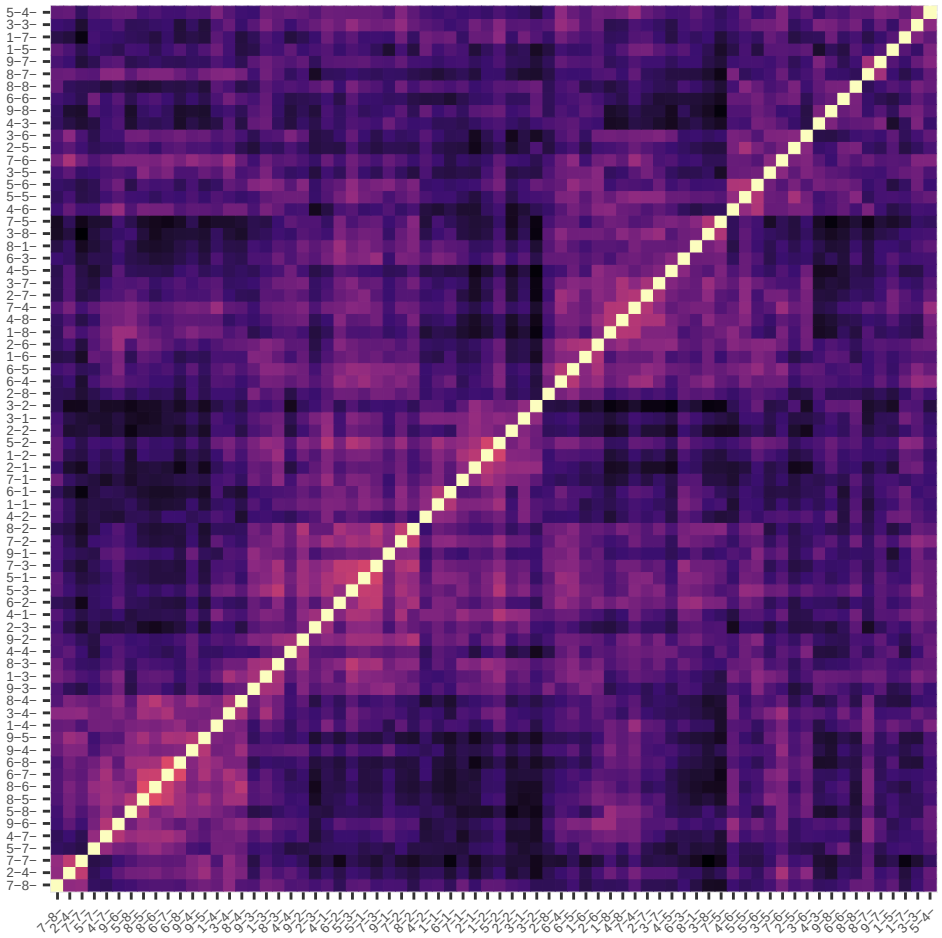
Evaluar la tendencia al agrupamiento con el estadístico de Hopkins que podéis calcular con la función `get_clust_tendency()` del paquete `factoextra`. La función `fviz_dist()` del mismo paquete permite realizar una imagen de la disimilaridad propuesta en el apartado anterior *(ordered dissimilarity image, ODI)* para evaluar gráficamente la tendencia al agrupamiento. También la misma función `get_clust_tendency()` lo hace con una imagen menos vistosa. ¿Cuantos conglomerados se intuyen? Para saber más sobre el análisis de la tendencia al agrupamiento, la fórmula del estadístico de Hopkins y un ejemplo se puede consultar la página https://www.datanovia.com/en/lessons/assessing-clustering-tendency/

Se realiza el cálculo del estadístico de Hopkins. En este caso resulta en aprox. 0.32, lo que se acera medianamente a 0. Los datos tienen una tendencia a agruparse moderada. Al mirar la gráfica de las distancias (la raíz de la disimilaridad) se corrobora lo anterior. Si bien los datos no son homogéneos, la forma en como se pueden agrupar no se ve tan claramente. Se podría intuir con el gráfico que hay 1 sector donde se aglomeran bien y luego unos 4 o quizás 5 sectores donde la los datos se aglomeran de manera menos estructurada, dependiendo del tamaño que se considere válido para una agrupación.

```r
set.seed(41)
get_clust_tendency(df, nrow(df)-1, graph = F)$hopkins_stat
```

```
## [1] 0.320416
```

```
fviz_dist(sq_dist, lab_size = 5) +
    # hay un 'bug' al compilar el pdf, que se ven los bordes de cada 'tile'
    # para arreglarlo hay que mapear 'color' a la variable de valor.
    aes(color = value) +
    scale_fill_viridis_c(direction = -1, option = "A") +
    scale_color_viridis_c(direction = -1, option = "A") +
    guides(fill = F, color = F)
```



(e)

Realizar un análisis de conglomerados jerárquico con el método de *Ward* de la distancia propuesta en el apartado (c). Dibujar el dendograma resultante.

Nota: Justamente el método de *Ward* exige que la distancia sea euclídea, pero no necesariamentela distancia euclídea.

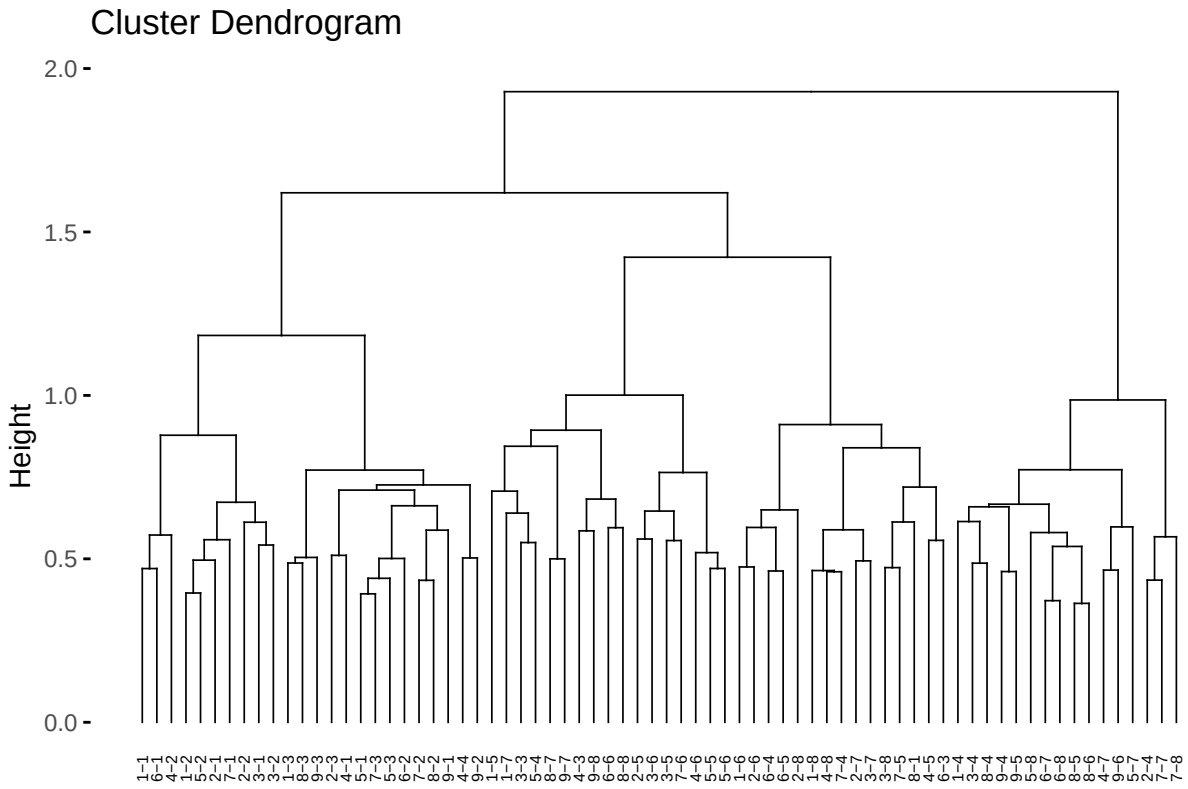
Dibujar también un *heatmap* de este análisis con la función `heatmap()`. Para ello, hay que elegir bien los parámetros `distfun=` y `hclustfun=` y una escala de colores. ¿Para qué sirve el heatmap?

Nota: No nos interesa re-ordenar las variables o columnas y tampoco un dendograma sobre ellas.

El dendograma permite detallar mejor el agrupamiento aglomerativo que se intuía en el apartado anterior.

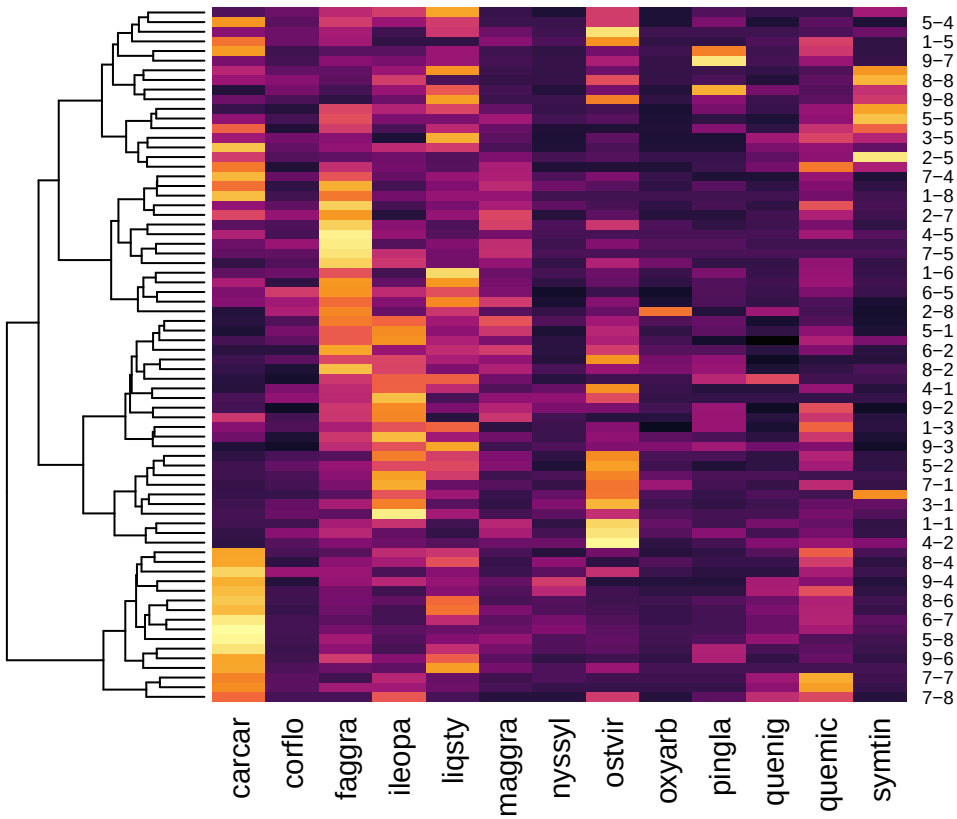
```
clust_1 <- agnes(sq_dist, diss = T, method = "ward")

fviz_dend(clust_1, cex = .4, lwd = .3)
```



La ventaja del "heatmap" es que incluye la información de las columnas. En este caso es posible ver la de manera gráfica la distribución de las especies en las diferentes muestra. Si ademas utilizamos un dendograma y ordenamos las muestras según el dendograma, obtenemos información visual de como se distribuyen las especies en las ramas del dendograma, y como se podrían construir los diferentes *clusters*.

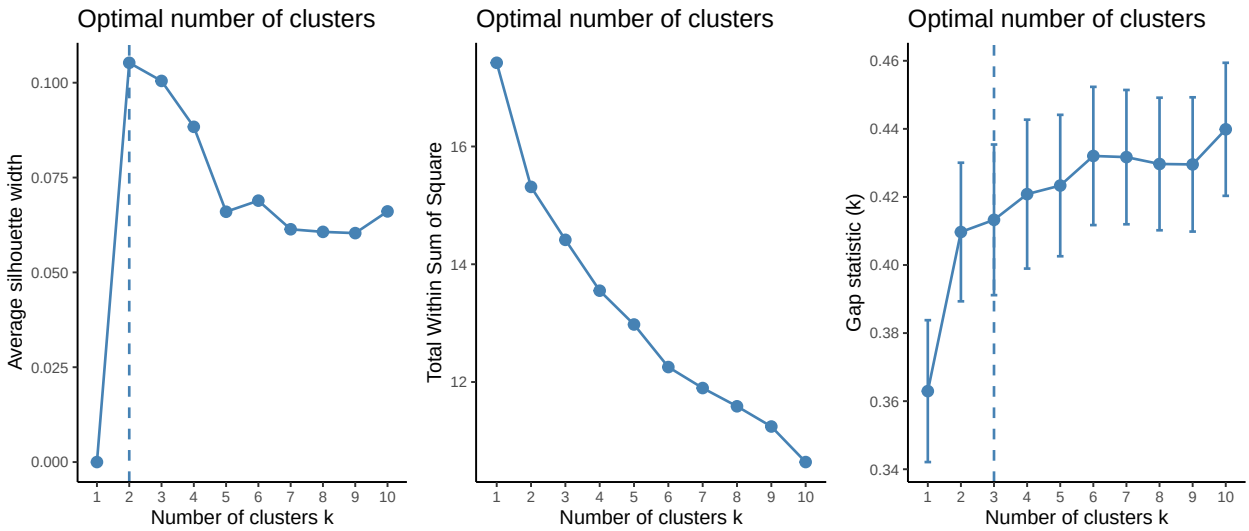
```
heatmap(as.matrix(df),
        Colv = NA,
        distfun = function(x) sqrt(vegdist(x)),
        hclustfun = hcut,
        col = hcl.colors(256, palette = "Inferno"))
```



(f)

El siguiente paso es estudiar por algún criterio el número óptimo de conglomerados para el análisis jerárquico. Con la distancia del apartado (c) en particular, lo más sencillo es utilizar el criterio de las siluetas. Sin embargo, el criterio de las siluetas y otros métodos como el de la suma de cuadrados dentro de los grupos o WSS o también el estadístico GAP se pueden aplicar con la función `fviz_nbclust()` del paquete `factoextra`. Habrá que especificar que se trata del método de clasificación jerárquico (que por defecto utiliza el método de Ward) y también la distancia raíz cuadrada de la disimilaridad de Bray-Curtis.

```
grid.arrange(
    fviz_nbclust(df, diss = sq_dist, FUNcluster = hcut, method = "silhouette") +
        theme_classic(base_size = 8),
    fviz_nbclust(df, diss = sq_dist, FUNcluster = hcut, method = "wss") +
        theme_classic(base_size = 8),
    fviz_nbclust(df, diss = sq_dist, FUNcluster = hcut, method = "gap_stat", verbose = F) +
        theme_classic(base_size = 8),
    nrow = 1)
```

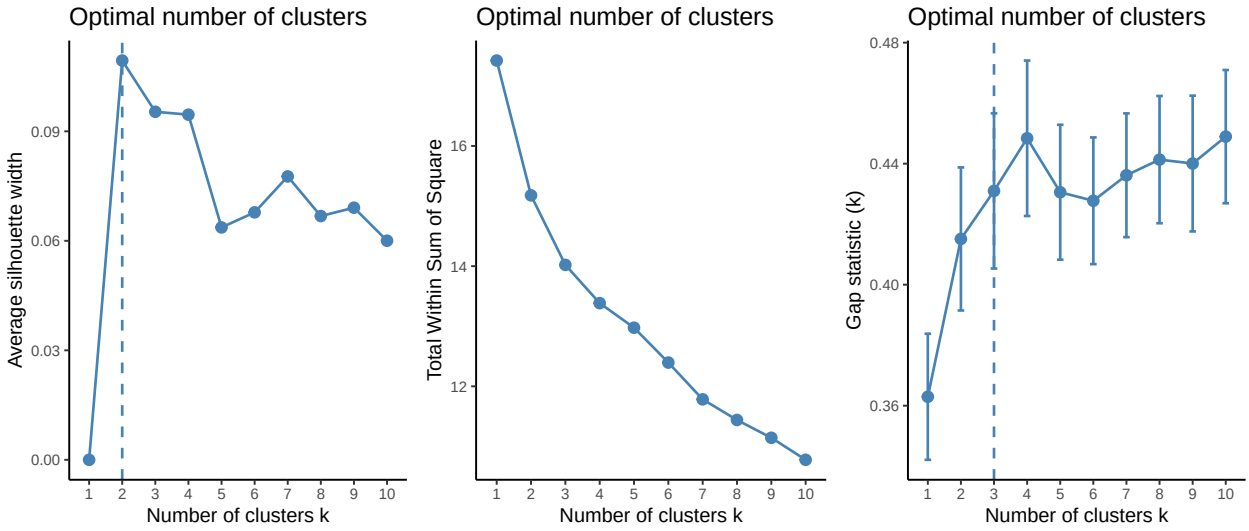


En este caso podemos observar que: Por el método de las siluetas la solución mas adecuada es las de 2 *clusters*. Por el método `gap_stat`, la solución óptima es la de 3 *clusters*. Por el método de WSS no hay una solución muy clara en la que se pueda establecer una reducción drástica o un *codo*. Quizás se podría decir que hay un *codo* en la solución de 2 *clusters*, y luego una en la de 6.

(g)

Estudiar con la misma distancia el número óptimo de conglomerados con el método PAM.

```
grid.arrange(
    fviz_nbclust(df, diss = sq_dist, FUNcluster = pam, method = "silhouette") +
        theme_classic(base_size = 8),
    fviz_nbclust(df, diss = sq_dist, FUNcluster = pam, method = "wss") +
        theme_classic(base_size = 8),
    fviz_nbclust(df, diss = sq_dist, FUNcluster = pam, method = "gap_stat", verbose = F) +
        theme_classic(base_size = 8),
    nrow = 1)
```



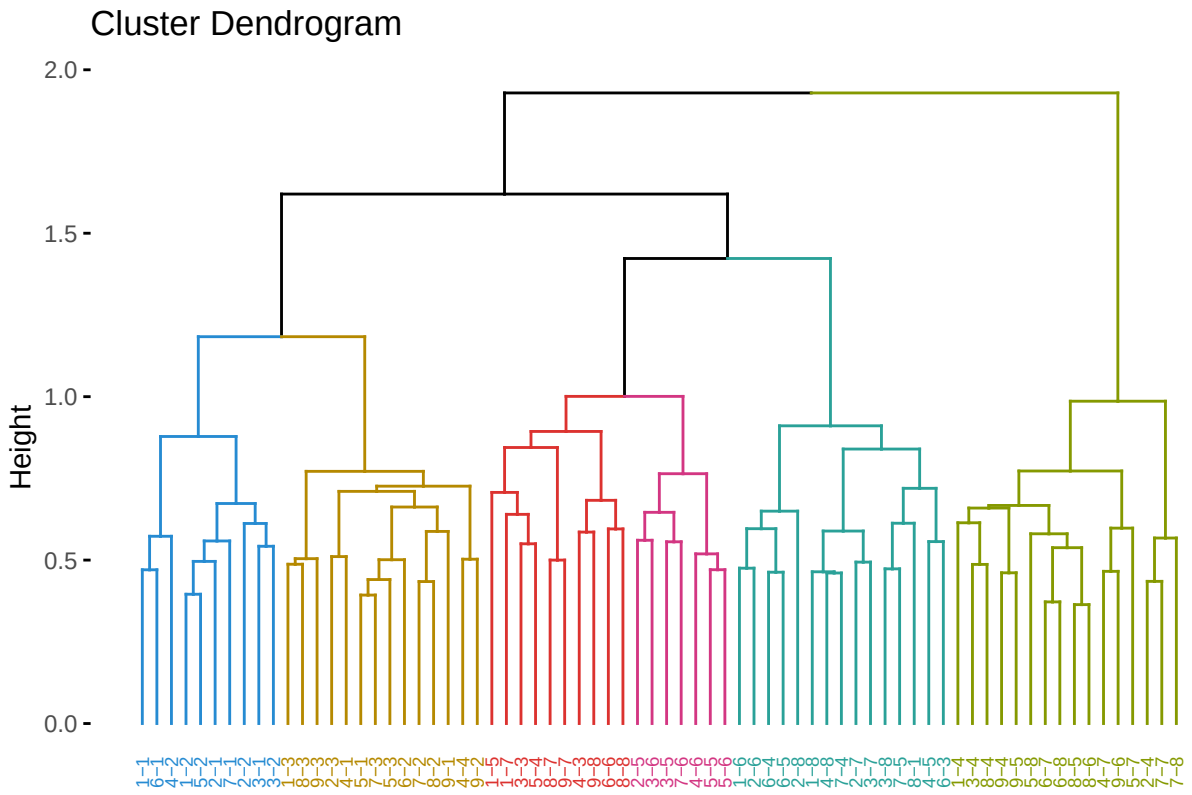
Pra el método de PAM, la solución no difiere mucho de la de `hcut`. La mejor solución por ancho de silueta es de 2 *clusters*, la de GAP es la de 3 *clusters*. POr WSS tampoco existe una solución muy clara. Se puede intuir un *codo* en los 3 y luego otro en los 7 *clusters*.

(h)

De los apartados anteriores se deduce que no hay un número claro de conglomerados. Un ecólogo experto propone que sean 6, ya que su experiencia así lo ha visto en otros bosques similares. Dibujar el dendograma del apartado (e) con esta partición. Con el objetivo de caracterizar los conglomerados, calcular una tabla con las medianas de las frecuencias de cada especie en cada uno de los seis conglomerados del dendograma anterior. Descartar de esta tabla las especies con una mediana inferior o igual a 4 en todos los conglomerados. Representar la tabla anterior con un análisis de correspondencias y un gráfico asimétrico.

```
clust_6 <- hcut(sq_dist, k = 6, hc_func = "agnes", hc_method = "ward.D2")

fviz_dend(clust_6,
          palette = solarized_pal()(6),
          lwd = .5,
          cex = .5)
```



Se puede observar como quedan definidos los 6 *clusters* en esta configuración.

```
dt <-
    df %>%
    mutate(cluster = factor(clust_6$cluster)) %>%
    group_by(cluster) %>%
    summarise_all(median) %>%
    # traslocamos el tibble para mejor visualización
```

```
    gather(especie, value, -cluster) %>%
    spread(cluster, value) %>%
    mutate(chk = `1` <= 4 & `2` <= 4 & `3` <= 4 & `4` <= 4 & `5` <= 4 & `6` <= 4 ) %>%
    filter(!chk) %>% select(-chk) %>%
    column_to_rownames('especie')

pander(dt)
```

	1	2	3	4	5	6
carcar	0	1	21	5.5	4	9
faggra	5	9.5	4.5	4.5	14	6
ileopa	16.5	11	3	4	3	4
liqsty	5.5	7	9	9	4	8
maggra	2.5	5	3	1	5	5
ostvir	18.5	7	1.5	7	3	2
quemic	4	5	9.5	3	5	8
symtin	0.5	0	0	4	0	16

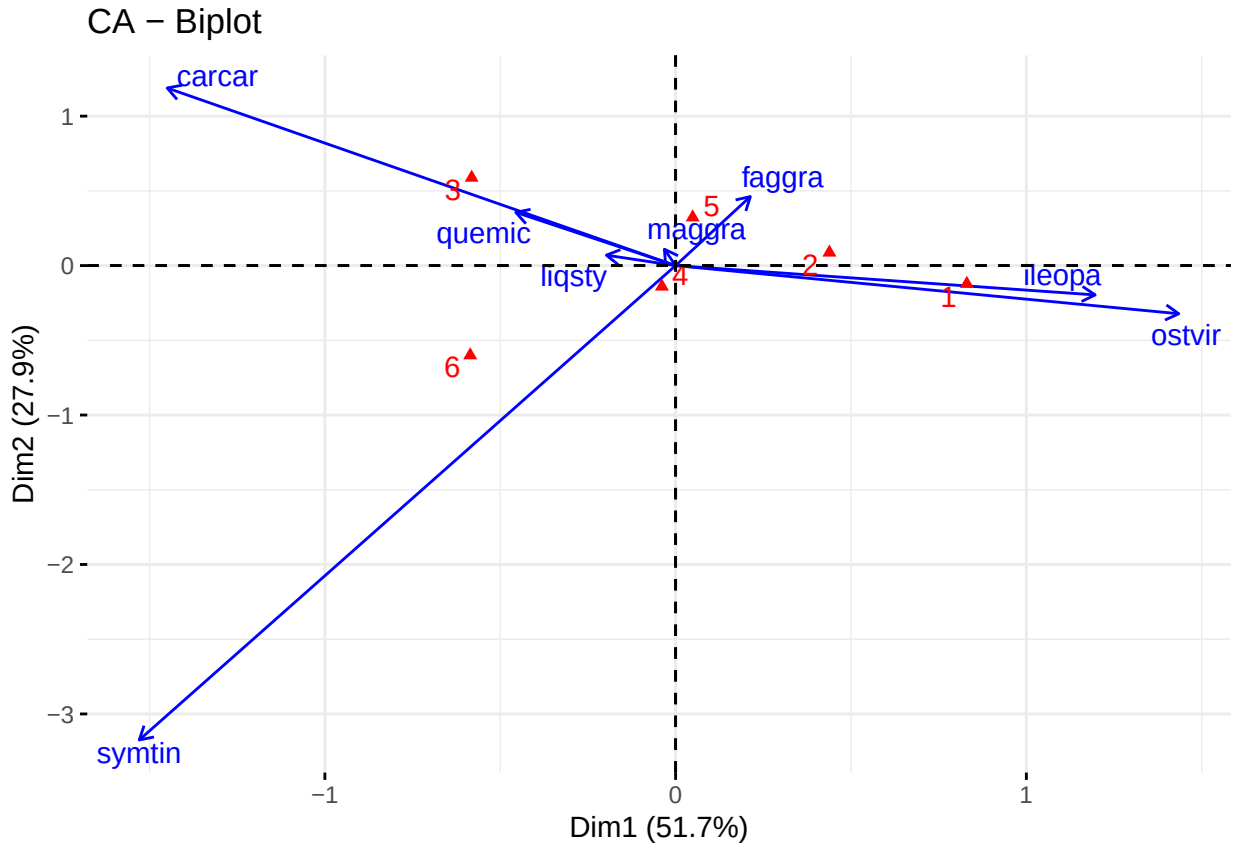
LA tabla permite observar la frecuencia de cada especie en los diferentes *clusters*, quitando las especies cuyas es menor de 4 en cada *cluster*.

```
ca_df <-
    FactoMineR::CA(dt, graph = F)

pander(ca_df$eig)
```

	eigenvalue	percentage of variance	cumulative percentage of variance
dim 1	0.2906	51.68	51.68
dim 2	0.1571	27.94	79.62
dim 3	0.09164	16.3	95.92
dim 4	0.0157	2.792	98.71
dim 5	0.007258	1.291	100

```
fviz_ca_biplot(ca_df, map ="colprincipal", arrows = c(T,F), repel = T)
```



En un gráfico asimétrico donde se represetan los conglomerados en la escala principal, podemos observar que los conglomerados que mas de diferencias son el 6 con el 1 y el 3. Los conglomerados 4 y 6, y en menor medida el 2, son menos diferentes o "idiotípicos" y dan menos infromacion diferencial.

Si miramos el angulo de los vectores de las especies, podemos observar una asociación entre *symtin* y el cluster 6, *ileopa* y *ostvir* con el cluster 1, y de *carcar* y *quemic* con el cluster 3.

Ejercicio 2 (30 pt.)

Es posible utilizar la regresión logística para estudiar la discriminación entre dos grupos dado un conjunto de variables explicativas. Para más detalles podéis consultar el apartado 8.10 del libro de Manly.

(a)

Realizar un análisis discriminante con ayuda de la regresión logística con los datos de los 49 gorriones hembra después de una tormenta. Reproducir con todo detalle la tabla 8.6 de la página 153 del libro de Manly.

```
load("gorriones.RData")

mod_lg <-
    glm(superviv ~ x1 + x2 + x3 + x4 + x5, gorriones, family = binomial(link = "logit"))

mod_NULL <-
    glm(superviv ~ 1, gorriones, family = binomial(link = "logit"))
```

```
# Para reporducir *con todo detalle* se puede utilizar la sintaxis de dplyr
# y crear la tabla
```

```r
tbl <-
    tidy(mod_lg) %>%
    mutate(
        chi_2 = (estimate / std.error) ^ 2,
        Variable = c("Constant", "Total length","Alar length", "Length beak and head",
                     "Length humerus", "Length keel of sternum"),
        chi_2 = round(chi_2, 2)) %>%
    mutate_at(vars(estimate:p.value), list( ~ round(., 3))) %>%
    select(
        Variable, '  estimate' = estimate, 'Standard error' = std.error,
        'Chi-squared' = chi_2, 'p-Value' = p.value)

tbl[1,4:5] <- "--"

pander(tbl, justify = "lcccc")
```

Variable	estimate	Standard error	Chi-squared	p-Value
Constant	13.58	15.87	–	–
Total length	-0.163	0.14	1.36	0.244
Alar length	-0.028	0.106	0.07	0.794
Length beak and head	-0.084	0.629	0.02	0.894
Length humerus	1.062	1.023	1.08	0.299
Length keel of sternum	0.072	0.417	0.03	0.864

(b)

Contrastar con un test χ^2 la significación de la regresión y explicar su resultado.

```r
anova(mod_NULL, mod_lg, test = "Chisq") %>%
    pander()
```

Table 4: Analysis of Deviance Table

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
48	66.92	NA	NA	NA
43	64.07	5	2.854	0.7225

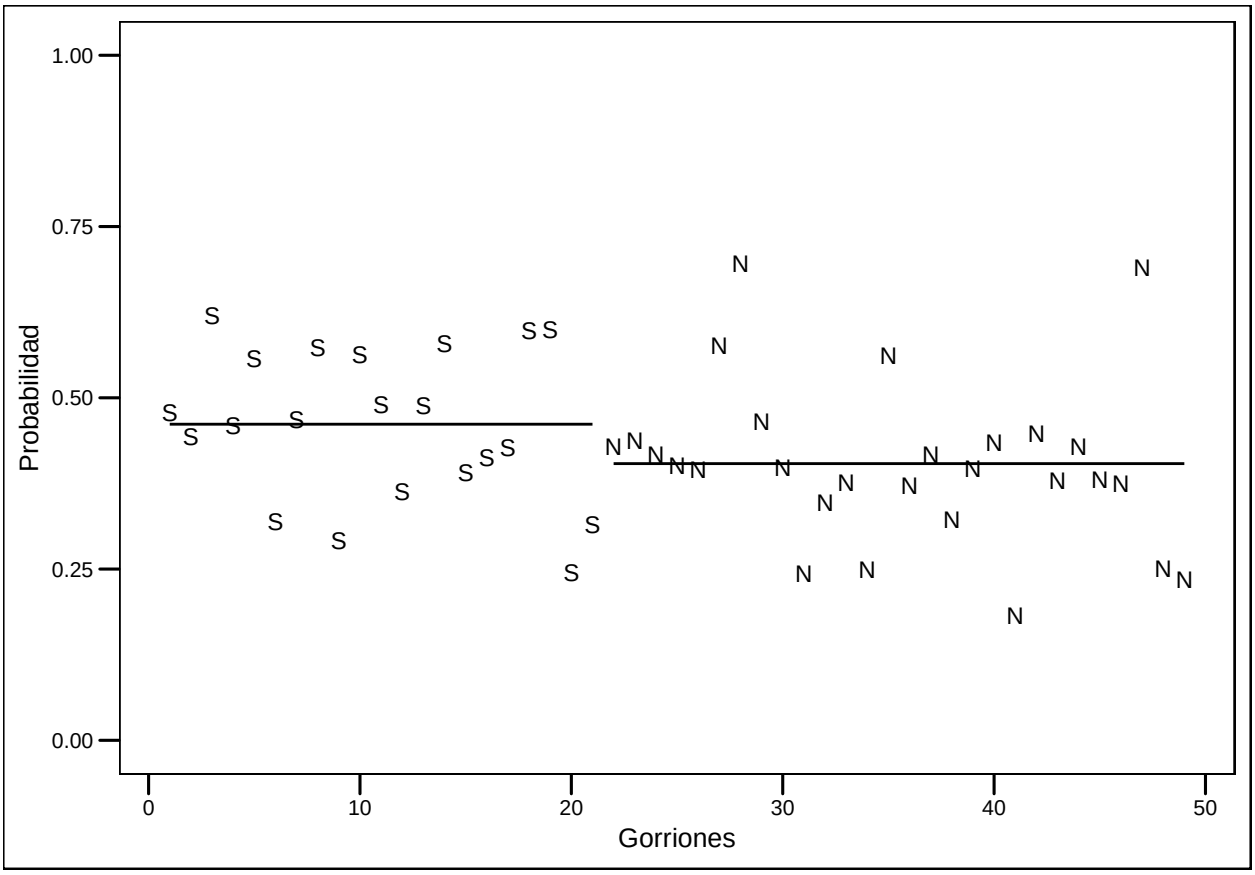
En este caso hacemos un contraste del modelo creado con el modelo nulo. De esta manera lo que se evalúa como $H0$ es que el modelo creado no difiere del modelo nulo, es decir no hay diferencias entre los grupos. El resultado del estadístico es mas bien bajo, y con un valor no significativo, por lo que no se puede rechazar la $H0$ y por ende, se establece que las diferencias entre los grupos no son significativas.

(c)

Realizar un gráfico como el de la figura 8.2 de la página 156 del libro de Manly pero con los datos de este ejercicio y valorar el resultado.

```r
tibble(
    fitted = mod_lg$fitted.values,
```

```
    class = augment(mod_lg)$superviv
) %>%
    rowid_to_column() %>%
    group_by(class) %>%
    mutate(mean = mean(fitted)) %>%
    ggplot(aes(rowid, fitted, shape = class)) +
    geom_point(size = 3) +
    geom_line(aes(y=mean)) +
    scale_shape_manual(values = c("N","S")) +
    theme_base(base_size = 10) +
    labs(x="Gorriones", y="Probabilidad") +
    scale_y_continuous(limits = c(0,1)) +
    guides(shape = F)
```



En el gráfico podemos observar que efectivamente, los dos grupos no presentan mayores diferencias en cuanto a lo predicho por el modelo según las variables cuantitativas que utiliza.

(d)

Calcular la tabla de confusión de la clasificación obtenida con la regresión logística.

```
ypred <-
    ifelse(
        predict.glm(mod_lg, type = "response") > 0.5,
        'S', 'N') %>%
```

```
    factor()


# para la matriz de confusión se puede utilizar la función de
# la librería `caret`
cm <- confusionMatrix(gorriones$superviv, ypred)


pander(cm$table)
```

	N	S
N	24	4
S	14	7

```
pander(cm$overall)
```

Table 6: Table continues below

Accuracy	Kappa	AccuracyLower	AccuracyUpper	AccuracyNull
0.6327	0.2025	0.4829	0.7658	0.7755

AccuracyPValue	McnemarPValue
0.9926	0.03389

Para el cálculo de la matriz de confusión, en primer lugar clasificamos la predicción del modelo; el modelo entrega una predicción probabilística acerca de la categoría, pero necesitamos utilizar una categoría determinada. Una vez que tenemos la predicción categórica, podemos utilizar la función `confusionMatrix` del paquete `caret`, que permite hacer la tabla de confusión y además calcular los parámetros de exactitud de la predicción.