Anàlisi multivariant_PAC2

Amelia Martínez Sequera

8/6/2020

Ejercicio 1

1. Componer un data.frame en R con estos datos de forma que las dos primeras columnas sean de texto y el resto numéricas. Sin embargo, la tercera columna (Location code) se debe convertir a factor con los nombres acortados de las áreas (Locality) de la Tabla 2. Comprobar que las frecuencias de felinos en cada área son correctas.

```
table(cleopard$`Location Code`) #Tabla de frecuencias
```

```
##
##  1  2  3  4  5  6
##  8 16 10 19  3  1
```

```
cleopard$`Location Code`<- factor(cleopard$`Location Code`, levels =
c(1,2,3,4,5,6), labels = c("Bor","Sum","Ind","Chi","Tai","Pal"))
cleopard$`Cloud Spots`<- as.numeric(cleopard$`Cloud Spots`)
cleopard$Lightness<- as.numeric(cleopard$Lightness)
cleopard$Brightness<- as.numeric(cleopard$Brightness)
cleopard$`Neck Stripes`<- as.numeric(cleopard$`Neck Stripes`)
cleopard$`Shoulder Patter`<- as.numeric(cleopard$`Shoulder Patter`)
str(cleopard)
```

```
## tibble [57 x 14] (S3: tbl_df/tbl/data.frame)
##  $ Register Number: chr [1:57] "1903.4.9.2" "40.374" "56134" "1212"
...
##  $ Locality       : chr [1:57] "Borneo, Sarawak, Baram" "West, Borneo,
Lawas River" "Borneo, British N., Borneo, Darvel Bay" "Borneo" ...
##  $ Location Code  : Factor w/ 6 levels "Bor","Sum","Ind",..: 1 1 1 1 1
1 1 1 2 2 ...
##  $ Cloud Size     : num [1:57] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Cloud Spots    : num [1:57] 3 3 2 1.5 3 2.5 2.5 2.5 2 3 ...
##  $ Lightness      : num [1:57] 1.5 1 2 2 2 2 2 2 2 2 ...
##  $ Brightness     : num [1:57] 1.5 1 2 2 2 3 3 2 2.5 2.5 ...
##  $ Coloration     : num [1:57] 2 2 1 2 2 3 6 2 2 2 ...
##  $ Dorsal Stripe  : num [1:57] 2 NA 3 3 2 NA 3 3 2 3 ...
##  $ Neck Stripes   : num [1:57] 2 NA NA 3 2 NA 2 2 2.5 2 ...
##  $ Shoulder Patter: num [1:57] 1.5 NA NA 2.5 1 NA 3 3 1 3 ...
##  $ Yellow         : num [1:57] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Tawny          : num [1:57] 0 0 0 0 0 1 1 0 0 0 ...
##  $ Grey           : num [1:57] 1 1 0 1 1 0 1 1 1 1 ...
```

```r
table(cleopard$`Location Code`)#Tabla de frecuencias
```

```
##
## Bor Sum Ind Chi Tai Pal
##   8  16  10  19   3   1
```

2. ¿Cual es la proporción de valores faltantes (missing) entre los datos de las 10 características?

```r
table(is.na(cleopard))
```

```
##
## FALSE  TRUE
##   772    26
```

¿Cuantas observaciones están completas y no tienen ningún dato faltante?

```r
cleopard[complete.cases(cleopard),]
```

```
## # A tibble: 44 x 14
##    `Register Numbe~ Locality `Location Code` `Cloud Size` `Cloud
Spots`
##    <chr>            <chr>    <fct>                  <dbl>
<dbl>
##  1 1903.4.9.2       Borneo,~ Bor                        1           3
##  2 1212             Borneo   Bor                        1
1.5
##  3 38148            Borneo,~ Bor                        1           3
##  4 196600           Borneo,~ Bor                        1
2.5
##  5 198705           Borneo,~ Bor                        1
2.5
##  6 d                Iles Ba~ Sum                        1           2
##  7 c                Iles Ba~ Sum                        1           3
##  8 1939.1656        Sumatra  Sum                        1           3
##  9 1938.11.30.23    Sumatra~ Sum                        1           3
## 10 1938.11.30.22    Sumatra~ Sum                        1
2.5
## # ... with 34 more rows, and 9 more variables: Lightness <dbl>,
## #   Brightness <dbl>, Coloration <dbl>, `Dorsal Stripe` <dbl>, `Neck
## #   Stripes` <dbl>, `Shoulder Patter` <dbl>, Yellow <dbl>, Tawny
<dbl>,
## #   Grey <dbl>
```

Hay 44 observaciones completas (de 57).

¿Cual es el argumento definitivo que los autores utilizan para justificar la imputación de los valores faltantes?

“Faltaban un total de 26 células de 570 (4,6%), debido al daño a las pieles. Para llevar a cabo un análisis multivariante, las celdas vacías se llenaron con un enésimo algoritmo vecino más cercano implementado por la función knn () en el paquete EMV

bajo R. Para cada individuo con un valor faltante para una característica particular, el método funciona al encontrar k individuos que han sido calificados para esa característica y que tienen la menor distancia euclidiana del individuo objetivo, medido a partir de los otros caracteres. El valor del carácter faltante se reemplaza por el promedio ponderado de los valores en el conjunto de k individuos, donde los pesos son inversamente proporcionales a la distancia euclidiana. En el análisis aquí, elegimos k = 2. De esta forma, usamos el 95% de los datos a costa de alguna aproximación, mientras que si excluimos las máscaras que tenían datos faltantes, se perdería una proporción mucho mayor."

Visto lo visto, los autores proponen rellenar los valores faltantes mediante una imputación por k-Nearest Neighbor con k = 2. Ellos utilizan la función knn() del paquete EMV, pero el paquete ya no existe. Una buena alternativa se encuentra en el paquete DMwR. Mejor si no modificamos los datos existentes y si utilizamos un estadístico robusto.

```
cleopard[is.na(cleopard$`Dorsal Stripe`), "Dorsal Stripe"] <-
mean(cleopard$`Dorsal Stripe`, na.rm=T)
cleopard[is.na(cleopard$`Neck Stripes`), "Neck Stripes"] <-
mean(cleopard$`Neck Stripes`, na.rm=T)
cleopard[is.na(cleopard$`Shoulder Patter`), "Shoulder Patter"] <-
mean(cleopard$`Shoulder Patter`, na.rm=T)
cleopard[is.na(cleopard$`Cloud Size` ), "Cloud Size"] <- 3
```

3. Calcular y mostrar en un gráfico las correlaciones entre las 10 características. Los autores recomiendan τ de Kendall ¿por qué? (ver Tabla S1). De las 45 correlaciones, ¿cuantas son significativas? (Escribir el código para responder a la pregunta.) Nota 1 : Para que el número de p-valores significativos coincida mejor con los detallados en la tabla S1, hay que considerar contrastes unilaterales en función del signo de tau. Nota 2 : La función rcorr() del paquete Hmisc solo sirve para las correlaciones de Pearson y de Spearman.

```
mvn(cleopard[,c(4:11)], univariateTest = "SW", desc = TRUE)
```

```
## $multivariateNormality
##              Test         Statistic           p value Result
## 1 Mardia Skewness  136.187680610301 0.148267180397585    YES
## 2 Mardia Kurtosis -1.10579794806902 0.268813983360912    YES
## 3             MVN              <NA>              <NA>    YES
##
## $univariateNormality
##           Test        Variable Statistic   p value Normality
## 1 Shapiro-Wilk   Cloud Size       0.6275  <0.001      NO
## 2 Shapiro-Wilk   Cloud Spots      0.8823  <0.001      NO
## 3 Shapiro-Wilk    Lightness       0.8555  <0.001      NO
## 4 Shapiro-Wilk   Brightness       0.8416  <0.001      NO
## 5 Shapiro-Wilk   Coloration       0.8342  <0.001      NO
## 6 Shapiro-Wilk  Dorsal Stripe     0.7800  <0.001      NO
## 7 Shapiro-Wilk  Neck Stripes      0.7923  <0.001      NO
```

```
## 8 Shapiro-Wilk Shoulder Patter    0.8116  <0.001      NO
##
## $Descriptives
##                  n     Mean   Std.Dev   Median Min Max 25th    75th
Skew
## Cloud Size      57 2.157895 0.9962335 3.000000 1.0   3  1.0 3.00000 -
0.31142228
## Cloud Spots     57 1.824561 0.8988645 2.000000 0.5   3  1.0 2.50000 -
0.05746295
## Lightness       57 1.991228 0.6975162 2.000000 1.0   3  1.5 2.50000
0.14692330
## Brightness      57 2.201754 0.7187517 2.000000 1.0   3  2.0 3.00000 -
0.39467954
## Coloration      57 2.877193 1.3505313 3.000000 1.0   7  2.0 3.00000
1.20176486
## Dorsal Stripe   57 2.469388 0.6006556 2.469388 1.0   3  2.0 3.00000 -
0.85150368
## Neck Stripes    57 2.063830 0.3966041 2.000000 1.0   3  2.0 2.06383
0.57680344
## Shoulder Patter 57 1.810000 0.8011710 1.810000 1.0   3  1.0 2.50000
0.37378792
##                    Kurtosis
## Cloud Size      -1.93606424
## Cloud Spots     -1.46352731
## Lightness       -1.09896859
## Brightness      -1.08417624
## Coloration       1.03235907
## Dorsal Stripe   -0.08055741
## Neck Stripes     1.26173779
## Shoulder Patter -1.42814880
```

Los autores probablemente lo usan porque los datos no siguen una distribución normal. Este método no pamétrico trabaja con rangos, por lo que requiere que las variables cuya relación se quiere estudiar sean ordinales o que se puedan transformar en rangos. Parece ser más aconsejable que el coeficiente de Spearman cuando el número de observaciones es pequeño o los valores se acumulan en una región, por lo que el número de ligaduras al generar los rangos es alto. Tau representa una probabilidad, es decir, es la diferencia entre la probabilidad de que las dos variables estén en el mismo orden en los datos observados y la probabilidad de que las dos variables estén en diferentes órdenes.

```
corleopard<- round(cor(cleopard[,c(4:14)], method = "kendall"),2)
corleopard
```

```
##                 Cloud Size Cloud Spots Lightness Brightness Coloration
## Cloud Size            1.00       -0.37      0.12       0.14       0.27
## Cloud Spots          -0.37        1.00     -0.21      -0.10      -0.13
## Lightness             0.12       -0.21      1.00       0.57      -0.29
## Brightness            0.14       -0.10      0.57       1.00      -0.18
## Coloration            0.27       -0.13     -0.29      -0.18       1.00
```

```
## Dorsal Stripe        -0.31        0.04      0.10       0.03      -0.04
## Neck Stripes         -0.16        0.07     -0.17      -0.12      -0.02
## Shoulder Patter      -0.05        0.18      0.02       0.00      -0.01
## Yellow                0.02        0.06      0.19       0.36      -0.52
## Tawny                 0.53       -0.09     -0.20       0.14       0.58
## Grey                 -0.50        0.06      0.00      -0.24      -0.09
##                 Dorsal Stripe Neck Stripes Shoulder Patter Yellow
Tawny  Grey
## Cloud Size              -0.31        -0.16           -0.05   0.02
0.53 -0.50
## Cloud Spots              0.04         0.07            0.18   0.06 -
0.09  0.06
## Lightness                0.10        -0.17            0.02   0.19 -
0.20  0.00
## Brightness               0.03        -0.12            0.00   0.36
0.14 -0.24
## Coloration              -0.04        -0.02           -0.01  -0.52
0.58 -0.09
## Dorsal Stripe            1.00        -0.09            0.09  -0.10 -
0.24  0.34
## Neck Stripes            -0.09         1.00           -0.20   0.15
0.10  0.01
## Shoulder Patter          0.09        -0.20            1.00   0.02 -
0.05  0.09
## Yellow                  -0.10         0.15            0.02   1.00
0.15 -0.23
## Tawny                   -0.24         0.10           -0.05   0.15
1.00 -0.66
## Grey                     0.34         0.01            0.09  -0.23 -
0.66  1.00
```

Los datos coinciden con la tabla S1 de los autores.

Obtenemos el p.value relacionando las variables de dos en dos (se podría hacer también mediante una función, pero lo haremos de manera "manual"):

```
CSP<-cor.test(cleopard$`Cloud Size`,cleopard$`Cloud Spots`,
method="kendall", use="pairwise")$p.value

LGT1<-cor.test(cleopard$`Cloud Size`,cleopard$Lightness,
method="kendall", use="pairwise")$p.value
LGT2<-cor.test(cleopard$`Cloud Spots`,cleopard$Lightness,
method="kendall", use="pairwise")$p.value

BRI1<-cor.test(cleopard$`Cloud Size`,cleopard$Brightness,
method="kendall", use="pairwise")$p.value
BRI2<-cor.test(cleopard$`Cloud Spots`,cleopard$Brightness,
method="kendall", use="pairwise")$p.value
BRI3<-cor.test(cleopard$Lightness,cleopard$Brightness, method="kendall",
use="pairwise")$p.value
```

```
DST1<-cor.test(cleopard$`Cloud Size`,cleopard$`Dorsal Stripe`,
method="kendall", use="pairwise")$p.value
DST2<-cor.test(cleopard$`Cloud Spots`,cleopard$`Dorsal Stripe`,
method="kendall", use="pairwise")$p.value
DST3<-cor.test(cleopard$Lightness, cleopard$`Dorsal Stripe`,
method="kendall", use="pairwise")$p.value
DST4<-cor.test(cleopard$Brightness,cleopard$`Dorsal Stripe`,
method="kendall", use="pairwise")$p.value

NST1<-cor.test(cleopard$`Cloud Size`,cleopard$`Neck Stripes`,
method="kendall", use="pairwise")$p.value
NST2<-cor.test(cleopard$`Cloud Spots`,cleopard$`Neck Stripes`,
method="kendall", use="pairwise")$p.value
NST3<-cor.test(cleopard$Lightness,cleopard$`Neck Stripes`,
method="kendall", use="pairwise")$p.value
NST4<-cor.test(cleopard$Brightness,cleopard$`Neck Stripes`,
method="kendall", use="pairwise")$p.value
NST5<-cor.test(cleopard$`Dorsal Stripe`,cleopard$`Neck Stripes`,
method="kendall", use="pairwise")$p.value

SHP1<-cor.test(cleopard$`Cloud Size`,cleopard$`Shoulder Patter`,
method="kendall", use="pairwise")$p.value
SHP2<-cor.test(cleopard$`Cloud Spots`,cleopard$`Shoulder Patter`,
method="kendall", use="pairwise")$p.value
SHP3<-cor.test(cleopard$Lightness,cleopard$`Shoulder Patter`,
method="kendall", use="pairwise")$p.value
SHP4<-cor.test(cleopard$Brightness,cleopard$`Shoulder Patter`,
method="kendall", use="pairwise")$p.value
SHP5<-cor.test(cleopard$`Dorsal Stripe`,cleopard$`Shoulder Patter`,
method="kendall", use="pairwise")$p.value
SHP6<-cor.test(cleopard$`Neck Stripes`,cleopard$`Shoulder Patter`,
method="kendall", use="pairwise")$p.value

YEL1<- cor.test(cleopard$`Cloud Size`,cleopard$Yellow, method="kendall",
use="pairwise")$p.value
YEL2<- cor.test(cleopard$`Cloud Spots`,cleopard$Yellow, method="kendall",
use="pairwise")$p.value
YEL3<- cor.test(cleopard$Lightness,cleopard$Yellow, method="kendall",
use="pairwise")$p.value
YEL4<- cor.test(cleopard$Brightness,cleopard$Yellow, method="kendall",
use="pairwise")$p.value
YEL5<- cor.test(cleopard$`Dorsal Stripe`,cleopard$Yellow,
method="kendall", use="pairwise")$p.value
YEL6<- cor.test(cleopard$`Neck Stripes`,cleopard$Yellow,
method="kendall", use="pairwise")$p.value
YEL7<- cor.test(cleopard$`Shoulder Patter`,cleopard$Yellow,
method="kendall", use="pairwise")$p.value

TAW1<- cor.test(cleopard$`Cloud Size`,cleopard$Tawny, method="kendall",
```

```
use="pairwise")$p.value
TAW2<- cor.test(cleopard$`Cloud Spots`,cleopard$Tawny, method="kendall",
use="pairwise")$p.value
TAW3<- cor.test(cleopard$Lightness,cleopard$Tawny, method="kendall",
use="pairwise")$p.value
TAW4<- cor.test(cleopard$Brightness,cleopard$Tawny, method="kendall",
use="pairwise")$p.value
TAW5<- cor.test(cleopard$`Dorsal Stripe`,cleopard$Tawny,
method="kendall", use="pairwise")$p.value
TAW6<- cor.test(cleopard$`Neck Stripes`,cleopard$Tawny, method="kendall",
use="pairwise")$p.value
TAW7<- cor.test(cleopard$`Shoulder Patter`,cleopard$Tawny,
method="kendall", use="pairwise")$p.value
TAW8<- cor.test(cleopard$Yellow, cleopard$Tawny, method="kendall",
use="pairwise")$p.value

GRY1<- cor.test(cleopard$`Cloud Size`,cleopard$Grey, method="kendall",
use="pairwise")$p.value
GRY2<- cor.test(cleopard$`Cloud Spots`,cleopard$Grey, method="kendall",
use="pairwise")$p.value
GRY3<- cor.test(cleopard$Lightness,cleopard$Grey, method="kendall",
use="pairwise")$p.value
GRY4<- cor.test(cleopard$Brightness,cleopard$Grey, method="kendall",
use="pairwise")$p.value
GRY5<- cor.test(cleopard$`Dorsal Stripe`,cleopard$Grey, method="kendall",
use="pairwise")$p.value
GRY6<- cor.test(cleopard$`Neck Stripes`,cleopard$Grey, method="kendall",
use="pairwise")$p.value
GRY7<- cor.test(cleopard$`Shoulder Patter`,cleopard$Grey,
method="kendall", use="pairwise")$p.value
GRY8<- cor.test(cleopard$Yellow, cleopard$Grey, method="kendall",
use="pairwise")$p.value
GRY9<- cor.test(cleopard$Tawny, cleopard$Grey, method="kendall",
use="pairwise")$p.value
```

Las correlaciones significativas ($p < 0.05$) son BRI3, CSP, DST1, GRY1, GRY4, GRY5, GRY9, TAW1, YEL4, que coinciden con los valores de la tabla S1 de los autores.

Podríamos haber especificado el contraste unilateral:

```
cor.test(cleopard$`Cloud Size`,cleopard$`Cloud Spots`, method="kendall",
alternative="less",use="pairwise")

##
##  Kendall's rank correlation tau
##
## data:  cleopard$`Cloud Size` and cleopard$`Cloud Spots`
## z = -3.0791, p-value = 0.001038
## alternative hypothesis: true tau is less than 0
## sample estimates:
```

```
##        tau
## -0.3653668
```

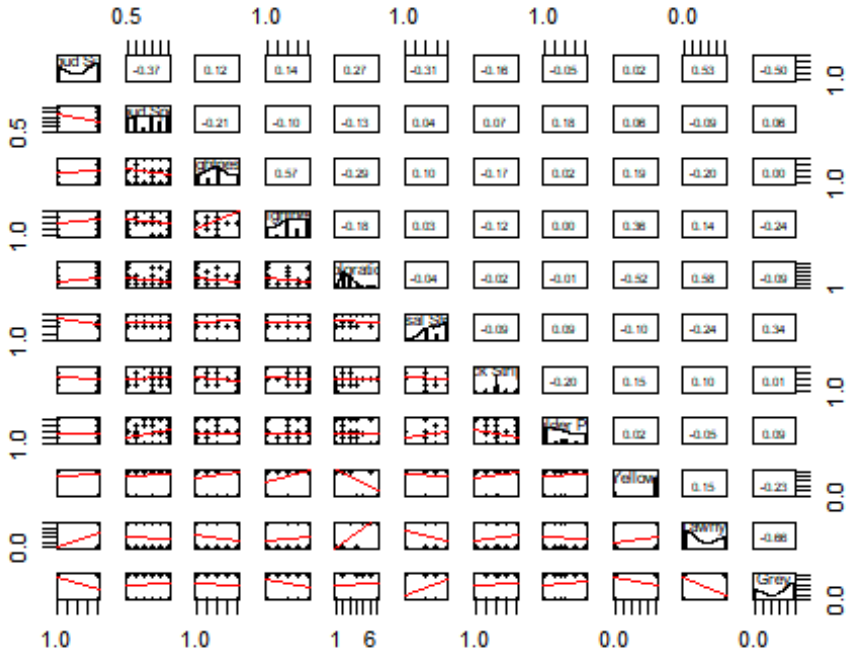
```
cor.test(cleopard$Lightness,cleopard$Brightness, method="kendall",
use="pairwise", alternative = "g")
```

```
##
##  Kendall's rank correlation tau
##
## data:  cleopard$Lightness and cleopard$Brightness
## z = 5.0464, p-value = 2.251e-07
## alternative hypothesis: true tau is greater than 0
## sample estimates:
##       tau
## 0.5670794
```

```
cor.test(cleopard$`Cloud Size`,cleopard$`Dorsal Stripe`,
method="kendall", use="pairwise",alternative="less")
```

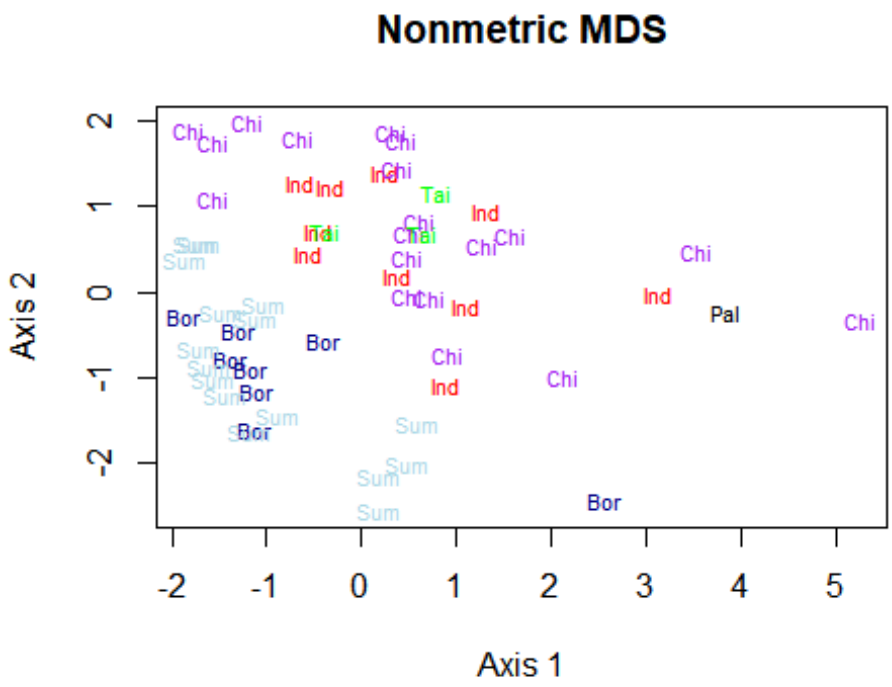
```
##
##  Kendall's rank correlation tau
##
## data:  cleopard$`Cloud Size` and cleopard$`Dorsal Stripe`
## z = -2.5106, p-value = 0.006026
## alternative hypothesis: true tau is less than 0
## sample estimates:
##       tau
## -0.3146205
```

```
pairs.panels(x = cleopard[,4:14], ellipses = FALSE, lm = TRUE, method =
"kendall")
```



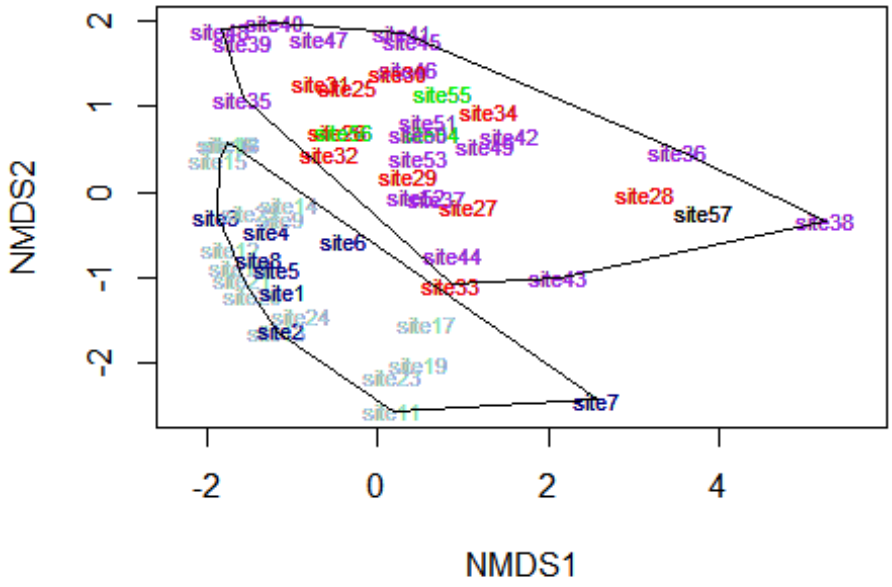
4. Realizar un escalado multidimensional isométrico de las 10 características con la función isoMDS del paquete MASS. A pesar de las posibles reticencias, la distancia elegida es la distancia euclídea.

```
cleopard<- dplyr::mutate(cleopard,
Group=c(rep("INDO",24),rep("CONT",33)))
D<- dist(cleopard[,c(4:14)])
D<- na.omit(D)
grups<-as.factor(cleopard$`Location Code`)
colores<- c("navyblue",
"lightblue","red","purple","green","black")[grups]
MDSnonmetric <- isoMDS(D, trace = T)

## initial  value 20.018595
## iter   5 value 17.175101
## iter   5 value 17.164200
## iter   5 value 17.163966
## final  value 17.163966
## converged

x <- MDSnonmetric$points[,1]
y <- MDSnonmetric$points[,2]
plot(x,y, xlab="Axis 1", ylab="Axis 2",
  main="Nonmetric MDS", type="n")
text(x, y, labels = grups, col=colores, cex=0.7)
```

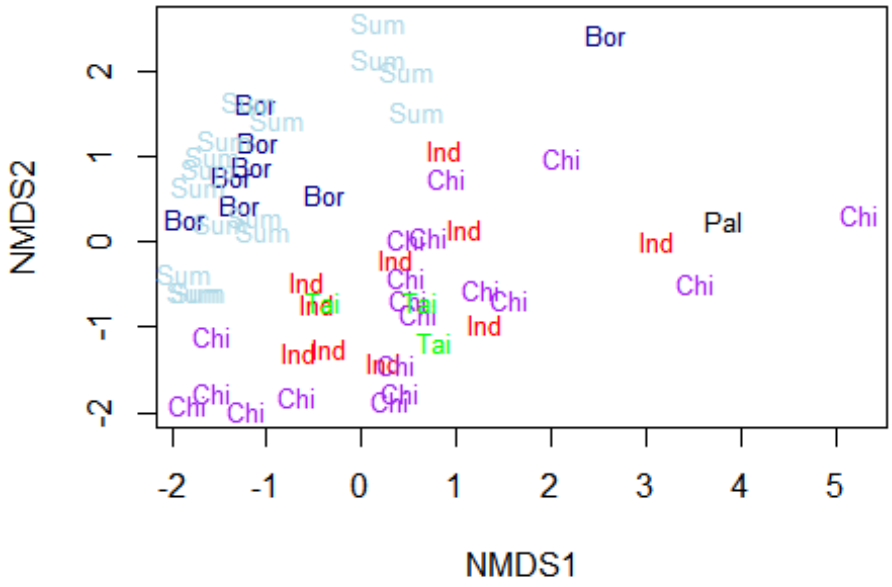


Mostrar en el mapa el área original de la pantera mediante colores o letras. Añadir las envolventes convexas (convex hulls) de los dos grupos que aparecen (continentales y indonesias). ¿Se obtiene la imagen A de la figura 2 del artículo? ¿Hay alguna diferencia notable?¿Es una buena representación en dos dimensiones de las distancias originales?

```
ordiplot(MDSnonmetric, type = "text",xlab="NMDS1", ylab="NMDS2")

## species scores not available

orditorp ( MDSnonmetric , display = "sites" , col = colores , air = 0.01
)
ordihull ( MDSnonmetric, groups = cleopard$Group , draw = "polygon" ,
label = F )
```



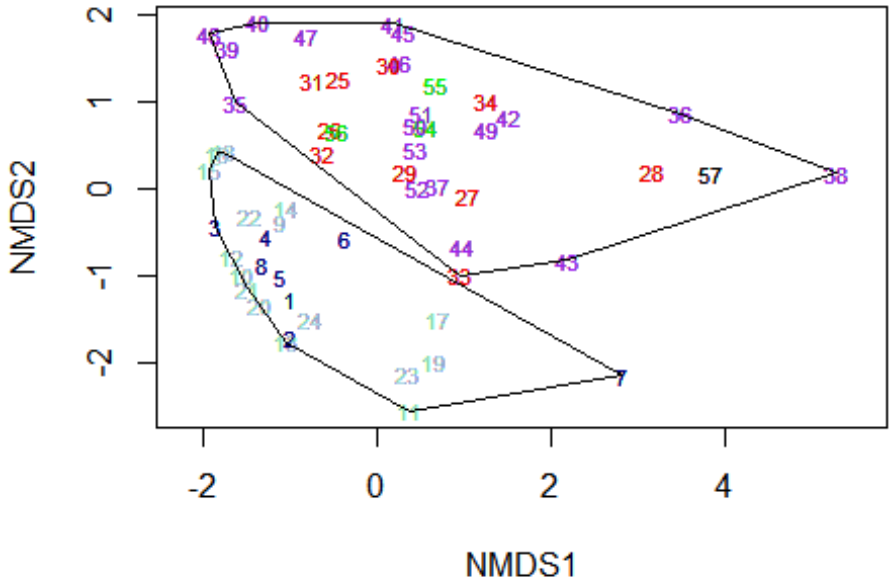

```r
plot(MDSnonmetric$points[,1], -MDSnonmetric$points[,2], type = "n",
xlab="NMDS1", ylab="NMDS2")
text(MDSnonmetric$points[,1], -MDSnonmetric$points[,2],labels = grups,
col=colores, cex=.8)
```



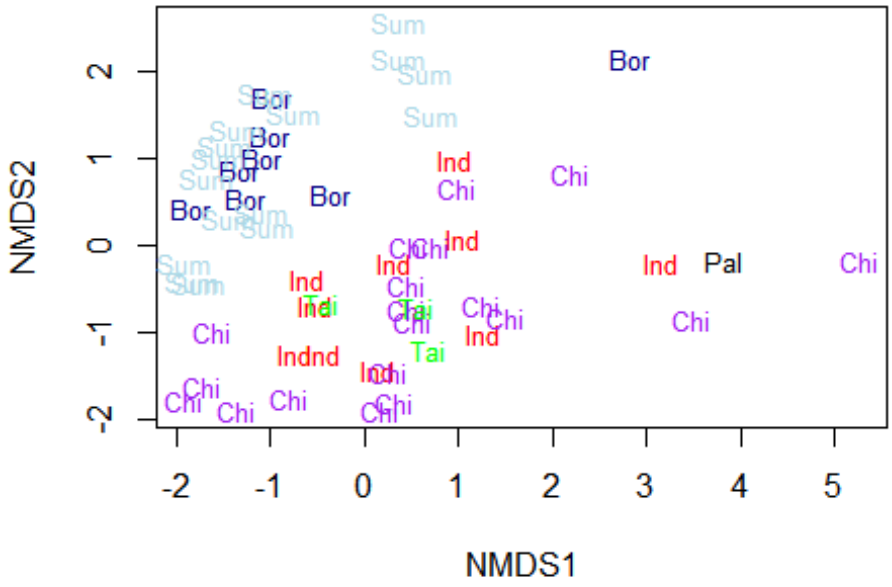
5. Como en el apartado anterior, realizar un escalado multidimensional no métrico con la función metaMDS() del paquete vegan, pero con el motor monoMDS.

```
#monoMDS en la función por defecto
MDSmeta<- vegan::metaMDS(comm = dist(cleopard[,c(4:14)]))

## Run 0 stress 0.1680045
## Run 1 stress 0.1680067
## ... Procrustes: rmse 0.0006410159  max resid 0.003284811
## ... Similar to previous best
## Run 2 stress 0.1680046
## ... Procrustes: rmse 0.000111282  max resid 0.0005458887
## ... Similar to previous best
## Run 3 stress 0.168005
## ... Procrustes: rmse 0.0003627294  max resid 0.001757393
## ... Similar to previous best
## Run 4 stress 0.1680044
## ... New best solution
## ... Procrustes: rmse 9.005191e-05  max resid 0.0004921633
## ... Similar to previous best
## Run 5 stress 0.1692298
## Run 6 stress 0.2302097
## Run 7 stress 0.1680045
## ... Procrustes: rmse 9.472465e-05  max resid 0.000501312
## ... Similar to previous best
## Run 8 stress 0.1680049
```

```
## ... Procrustes: rmse 0.0001796855  max resid 0.0009278674
## ... Similar to previous best
## Run 9 stress 0.1692294
## Run 10 stress 0.1690283
## Run 11 stress 0.1680044
## ... New best solution
## ... Procrustes: rmse 4.256737e-05  max resid 0.000240201
## ... Similar to previous best
## Run 12 stress 0.1690313
## Run 13 stress 0.1680061
## ... Procrustes: rmse 0.0005348501  max resid 0.002691802
## ... Similar to previous best
## Run 14 stress 0.1680064
## ... Procrustes: rmse 0.0005780446  max resid 0.002891012
## ... Similar to previous best
## Run 15 stress 0.1680053
## ... Procrustes: rmse 0.0003964792  max resid 0.002119837
## ... Similar to previous best
## Run 16 stress 0.1680049
## ... Procrustes: rmse 0.0002867222  max resid 0.001501424
## ... Similar to previous best
## Run 17 stress 0.169028
## Run 18 stress 0.1680055
## ... Procrustes: rmse 0.000429008  max resid 0.00224352
## ... Similar to previous best
## Run 19 stress 0.1690287
## Run 20 stress 0.1680046
## ... Procrustes: rmse 0.0001596663  max resid 0.0007629553
## ... Similar to previous best
## *** Solution reached

ordiplot(MDSmeta, type = "text",xlab="NMDS1", ylab="NMDS2")

## species scores not available

orditorp ( MDSmeta , display = "sites" , col = colores , air = 0.01 )
ordihull ( MDSmeta , groups = cleopard$Group , draw = "polygon" , label =
F )
```




```
plot(MDSmeta$points[,1], -MDSmeta$points[,2], type = "n", xlab="NMDS1",
ylab="NMDS2")
text(MDSmeta$points[,1], -MDSmeta$points[,2],labels = grups, col=colores,
cex=.8)
```

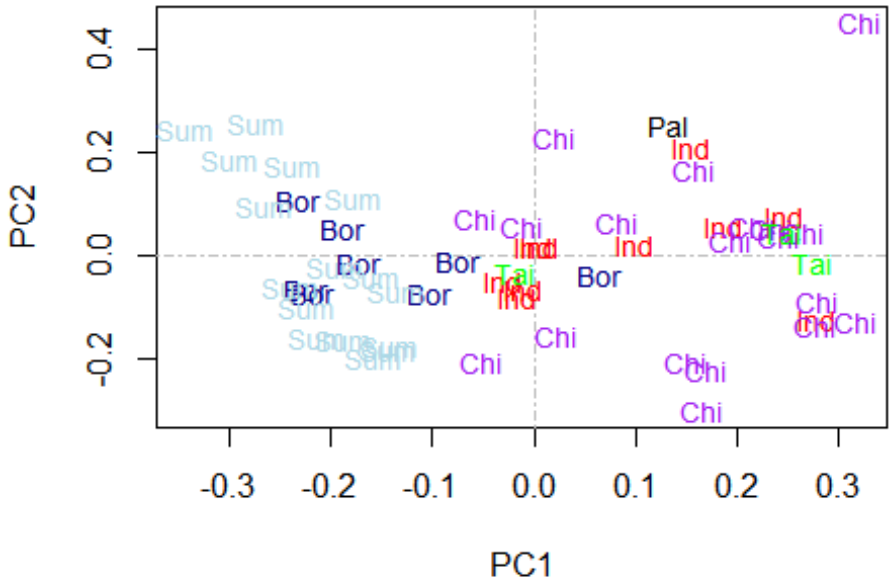


6. ¿Mejoramos el resultado del apartado anterior si utilizamos la distancia de Gower?

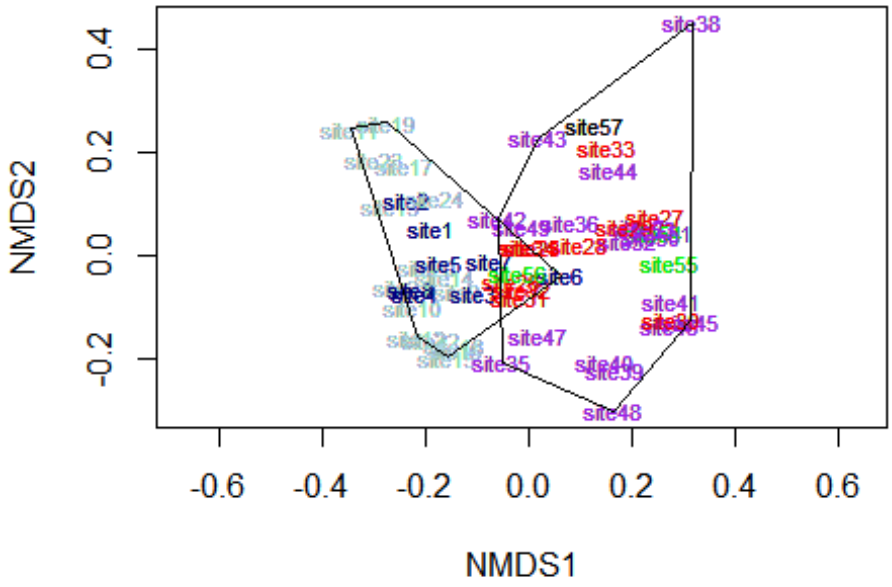
```
library(cluster)
G.dist <- daisy(cleopard[,c(4:14)],metric="gower",
type=list(asymm=c(9,10,11),ordratio=1))
attr(G.dist,"Types")

##  [1] "T" "I" "I" "I" "I" "I" "I" "I" "A" "A" "A"

mds <- cmdscale(G.dist,eig=T); mds

## $points
##                [,1]         [,2]
##  [1,] -0.1888483848  0.052600709
##  [2,] -0.2324310347  0.108474375
##  [3,] -0.1028824494 -0.073517854
##  [4,] -0.2170732690 -0.070223696
##  [5,] -0.1713021735 -0.014466364
##  [6,]  0.0637856302 -0.036588778
##  [7,] -0.0746660103 -0.009752036
##  [8,] -0.2249184925 -0.064300379
##  [9,] -0.1370501468 -0.066204294
## [10,] -0.2235544725 -0.100604820
## [11,] -0.3441899725  0.247223300
## [12,] -0.2144235966 -0.157226598
## [13,] -0.2654520105  0.095514339
```

```
## [14,] -0.1596565036 -0.038594851
## [15,] -0.1574653088 -0.195652187
## [16,] -0.1444221432 -0.183403585
## [17,] -0.2389198941  0.174030700
## [18,] -0.1420141746 -0.173153651
## [19,] -0.2731282330  0.258954494
## [20,] -0.1966984156 -0.018861888
## [21,] -0.2401968855 -0.059635614
## [22,] -0.1886169536 -0.160228834
## [23,] -0.2987965494  0.185617869
## [24,] -0.1782022104  0.111264205
## [25,]  0.0076869085  0.018885225
## [26,] -0.0285443887 -0.048686915
## [27,]  0.2476193075  0.077229614
## [28,]  0.0999212125  0.023778821
## [29,]  0.1877845579  0.056051263
## [30,]  0.2796454538 -0.123983960
## [31,] -0.0146207856 -0.080399466
## [32,] -0.0095029509 -0.065448817
## [33,]  0.1550266707  0.210477959
## [34,]  0.0006038973  0.018683623
## [35,] -0.0511886225 -0.205221588
## [36,]  0.0833190699  0.066203208
## [37,]  0.2144452624  0.059563636
## [38,]  0.3207568460  0.452802101
## [39,]  0.1703198161 -0.221929948
## [40,]  0.1498721006 -0.204301088
## [41,]  0.2786138923 -0.085799849
## [42,] -0.0565273077  0.073158115
## [43,]  0.0206353803  0.230380076
## [44,]  0.1581694324  0.167713077
## [45,]  0.3162598486 -0.125291806
## [46,]  0.2771314016 -0.133957250
## [47,]  0.0227969513 -0.154659071
## [48,]  0.1668463826 -0.300816353
## [49,] -0.0107840241  0.057509786
## [50,]  0.2418889903  0.039072907
## [51,]  0.2647199249  0.046845533
## [52,]  0.1942732033  0.031026494
## [53,]  0.2339555253  0.055712178
## [54,]  0.2434919903  0.046571506
## [55,]  0.2743144638 -0.014624300
## [56,] -0.0200059180 -0.031219282
## [57,]  0.1321991621  0.253410007
##
## $eig
##  [1]  2.121546e+00  1.167141e+00  8.363817e-01  6.375287e-01
4.259045e-01
##  [6]  2.770360e-01  2.593264e-01  2.226950e-01  1.628464e-01
1.235076e-01
```

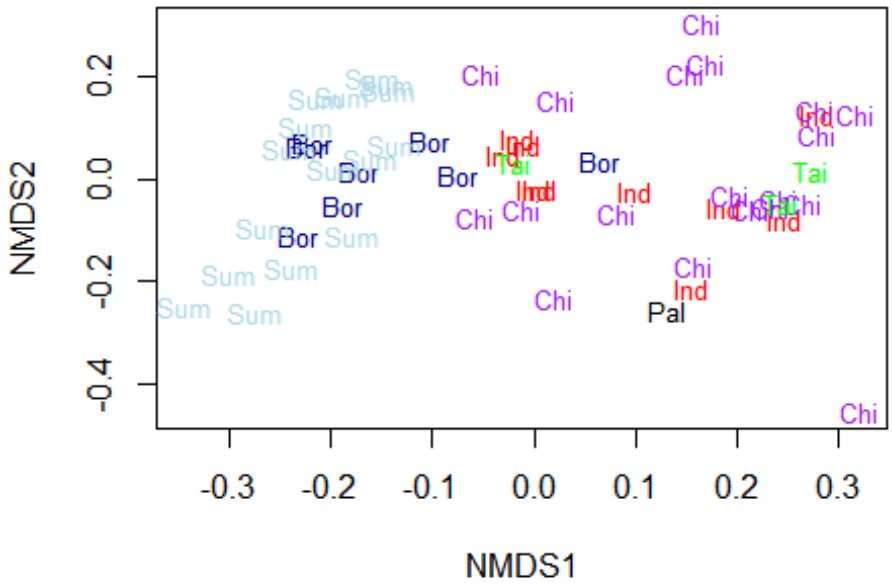
```
## [11]  1.122267e-01  9.819203e-02  7.160494e-02  3.768361e-02
2.964450e-02
## [16]  1.496613e-02  8.280422e-03  6.767015e-03  2.118455e-03 -
1.733368e-16
## [21] -1.320800e-04 -1.493652e-03 -4.192026e-03 -5.156493e-03 -
7.588330e-03
## [26] -8.445891e-03 -9.457785e-03 -9.741131e-03 -1.316939e-02 -
1.473439e-02
## [31] -1.513396e-02 -1.718681e-02 -1.841026e-02 -2.139833e-02 -
2.289237e-02
## [36] -2.603466e-02 -2.896289e-02 -3.149474e-02 -3.424779e-02 -
3.514900e-02
## [41] -4.094405e-02 -4.553400e-02 -5.071026e-02 -5.111158e-02 -
5.262961e-02
## [46] -5.846893e-02 -6.357493e-02 -6.709548e-02 -6.982250e-02 -
7.202896e-02
## [51] -8.670993e-02 -9.315882e-02 -1.043872e-01 -1.239612e-01 -
1.359429e-01
## [56] -1.924386e-01 -2.503008e-01
##
## $x
## NULL
##
## $ac
## [1] 0
##
## $GOF
## [1] 0.3869390 0.4971261

plot(mds$points,type="n",xlab="PC1",ylab="PC2")
abline(h=0,v=0,lty=4,col="gray")
text(mds$points[,1],mds$points[,2],labels = grups, col=colores, cex=.9)
```




```
ordiplot(mds, type = "text",xlab="NMDS1", ylab="NMDS2")

## species scores not available

orditorp ( mds , display = "sites" , col = colores , air = 0.01 )
ordihull ( mds , groups = cleopard$Group , draw = "polygon"  , label = F
)
```




```
plot(mds$points[,1], -mds$points[,2], type = "n", xlab="NMDS1",
ylab="NMDS2")
text(mds$points[,1], -mds$points[,2],labels = grups, col=colores, cex=.8)
```

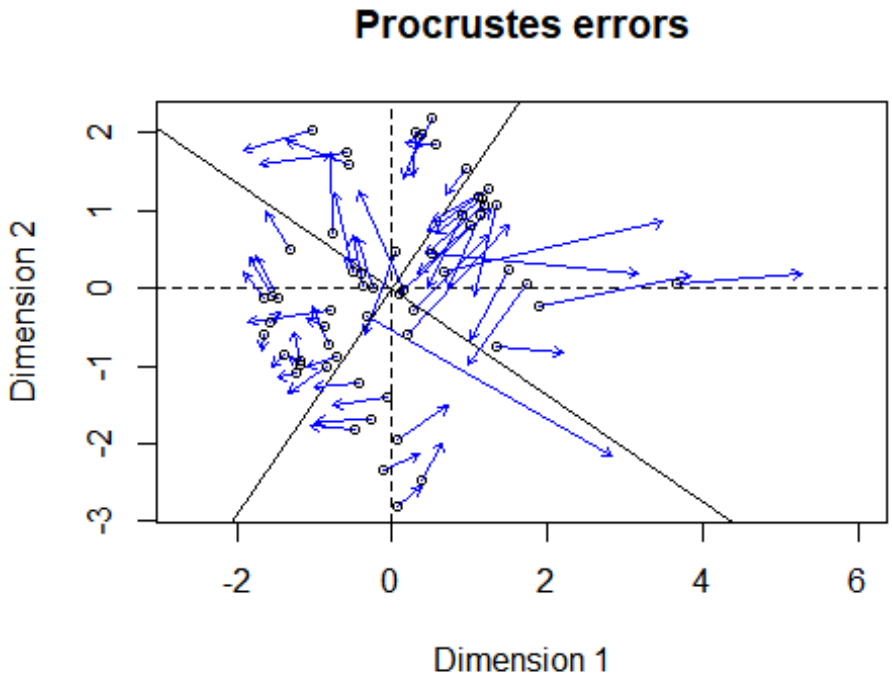


La gráfica se asemeja más a la de los autores del estudio, probablemente utilizaron las distancias de Gower.

Si las variables son mixtas, una mezcla de contínuas, binarias o cualitativas, es entonces adecuado utilizar la distancia de Gower, ya que la distancia no es euclídea y se obtiene mejores predicciones.

7. Comparar los resultados de los dos apartados anteriores con la función procrustes(). Identificar la pantera que tiene el cambio más notable entre los dos mapas.

```
pro<-procrustes(MDSmeta, mds)
plot(pro)
```



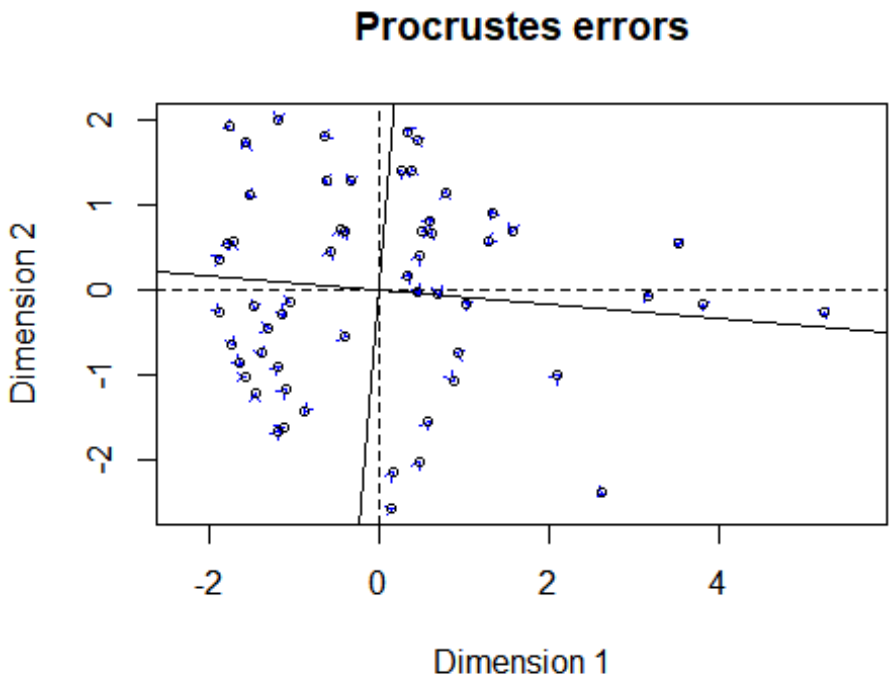

```
sort(residuals(pro), decreasing=T)
```

```
##         7        36        28        57        42        38        34
25
## 3.6277054 2.8838564 2.6805113 1.9950539 1.9203921 1.6002050 1.5896737
1.4010966
##        49        33        27        39         6         3        47
31
## 1.3679519 1.3004287 1.1854461 1.1459648 1.1432266 1.0766285 1.0543449
1.0452364
##        44        52        37        53        48        29        40
43
## 1.0408119 1.0275074 1.0209457 0.9679773 0.9397159 0.9209867 0.8684548
```

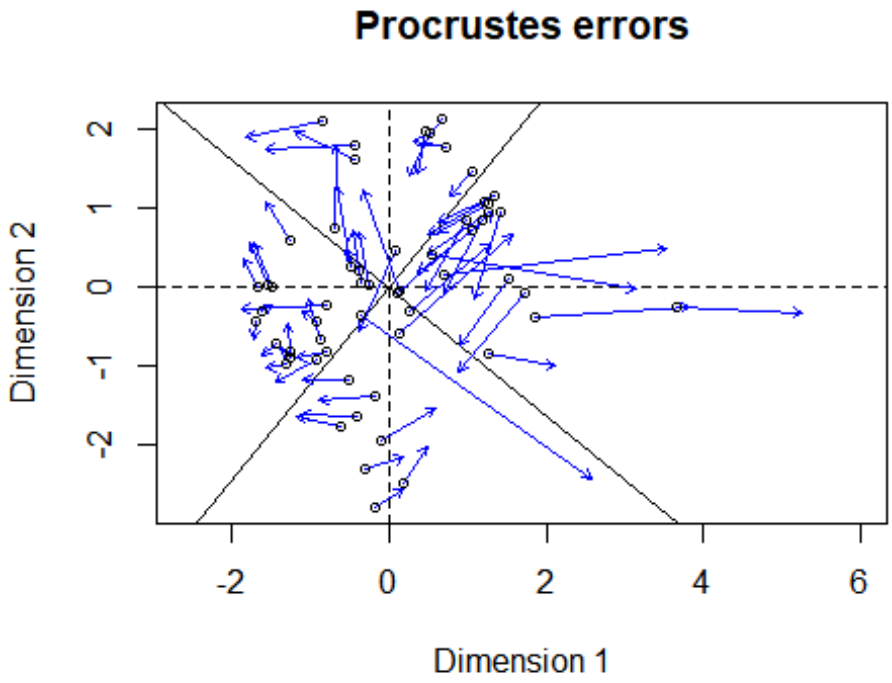
```
0.8620852
##        51        50        17        54         2        24        56
26
## 0.8314416 0.8023087 0.7921940 0.7552535 0.7111660 0.7090787 0.6890059
0.6669966
##        30        18        20        35         1        16        13
46
## 0.6165557 0.6158724 0.6050461 0.5859774 0.5808835 0.5793444 0.5777314
0.5762965
##        14        19        23        45        55        15         5
11
## 0.5450181 0.5369666 0.4998971 0.4410642 0.4406103 0.4199806 0.4084185
0.4001592
##        41         4        32        21         9        10        12
8
## 0.3938904 0.3726033 0.2937369 0.2623439 0.2607450 0.2329949 0.2093807
0.1760592
##        22
## 0.1758811

summary(pro)

##
## Call:
## procrustes(X = MDSmeta, Y = mds)
##
## Number of objects: 57    Number of dimensions: 2
##
## Procrustes sum of squares:
##  71.35258
## Procrustes root mean squared error:
##  1.118839
## Quantiles of Procrustes errors:
##       Min        1Q    Median        3Q       Max
## 0.1758811 0.4998971 0.7111660 1.0543449 3.6277054
##
## Rotation matrix:
##           [,1]       [,2]
## [1,] 0.5653149  0.8248752
## [2,] 0.8248752 -0.5653149
##
## Translation of averages:
##               [,1]         [,2]
## [1,] -5.467294e-17 1.030203e-16
##
## Scaling of target:
## [1] 6.618258
```

Procrustes rota una matriz para maximizar la similitud con otra matriz, minimizando la diferencia de los mínimos cuadrados. Es particularmente útil para comparar soluciones alternativas en el escalamiento multidimensional.

```
proc <- procrustes(MDSnonmetric,MDSmeta)
plot(proc)
```




```
proc2 <- procrustes(MDSnonmetric,mds)
plot(proc2)
```

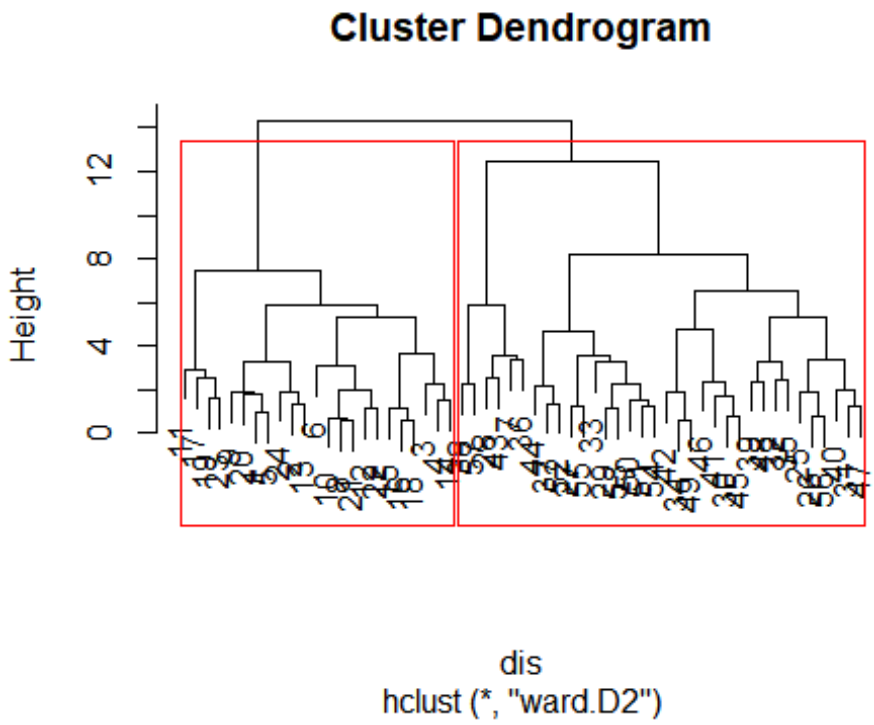


Las diferencias son mínimas entre los métodos isoMSD() y metaMSD. En cambio, hay muchas observaciones que difieren mucho entre éstos y los resultados con las distancias de Gower.

Ejercicio 2

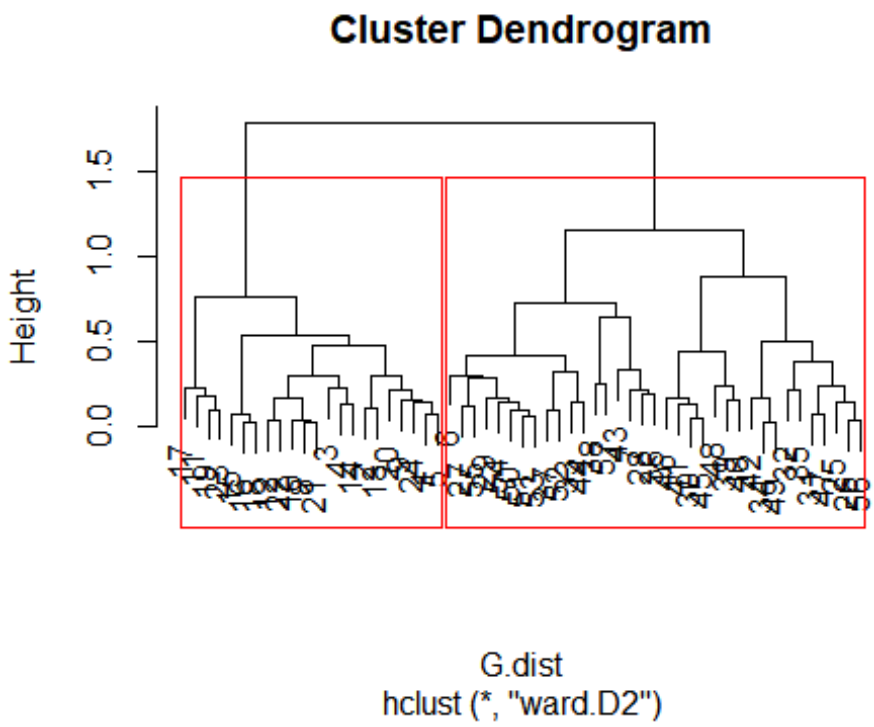
1. Realizar un análisis de conglomerados jerárquico con las distancias euclídeas y el método de Ward.

```
dis<- dist(cleopard[,-1])
clustl<- hclust(dis, method = "ward.D2")
plot(clustl)
rect.hclust(clustl, k=2, border="red")
```



Repetir el análisis con la distancia de Gower explicada en el apartado 1.6. Comparar ambos resultados y también con los dos grupos (indonesias y continentales) observados en el NMDS.

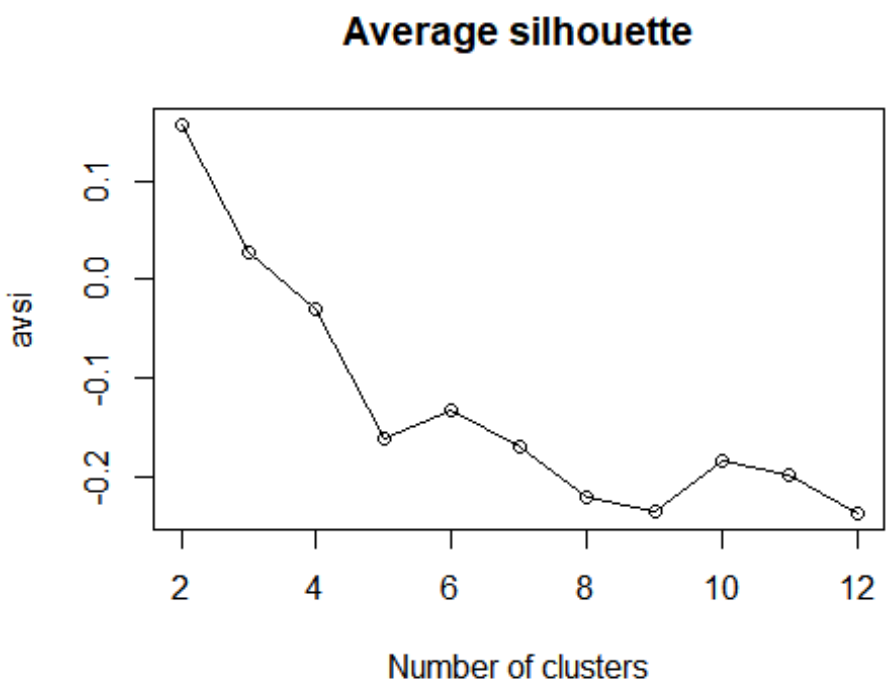
```
clustG<-hclust(G.dist, method = "ward.D2")
plot(clustG)
rect.hclust(clustG, k=2, border="red")
```



No se obtienen los mismos resultados.

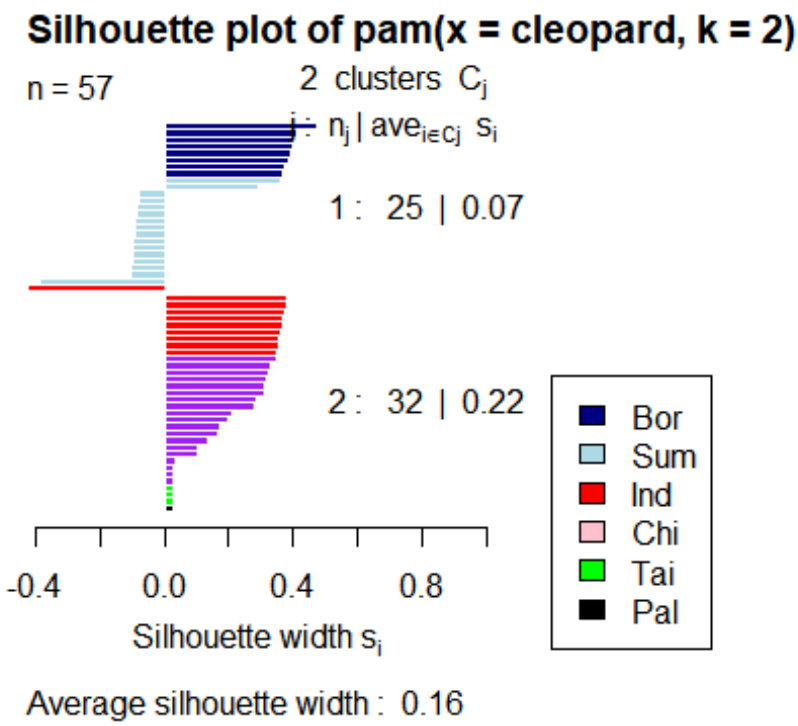
2. Realizar un particionado alrededor de los k-medoides con estos datos y la distancia euclídea. Estudiar el mejor número de conglomerados con ayuda de sus siluetas. Dibujar un gráfico bidimensional con los conglomerados hallados.

```
avsi <- c(0)
for(i in 2:12){
  si <- silhouette(pam(cleopard,k=i))
  avsi[i-1] <- mean(si[,"sil_width"])
  }
plot(2:12,avsi,type="o",xlab="Number of clusters")
title(main="Average silhouette")
```

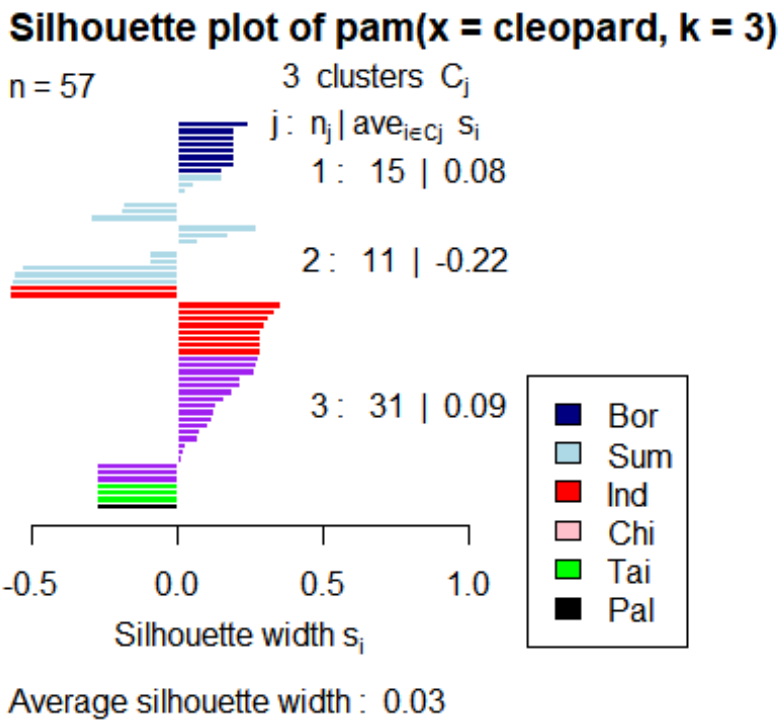


Según este criterio debemos hacer 2 clusters. Veamos las siluetas:

```
par(xpd= T, mar = par()$mar + c(0,0,0,7))

pam2 <- pam(cleopard,k=2)
si2 <- silhouette(pam2)
plot(si2,col=colores)
legend(1.2,20,
fill=c("navyblue","lightblue","red","pink","green","black"),
       legend=levels(grups))
```




```
pam3 <- pam(cleopard,k=3)
si3 <- silhouette(pam3)
plot(si3,col=colores)
legend(1.2,20,fill=c("navyblue","lightblue","red","pink","green","black")
,
       legend=levels(grups))
```

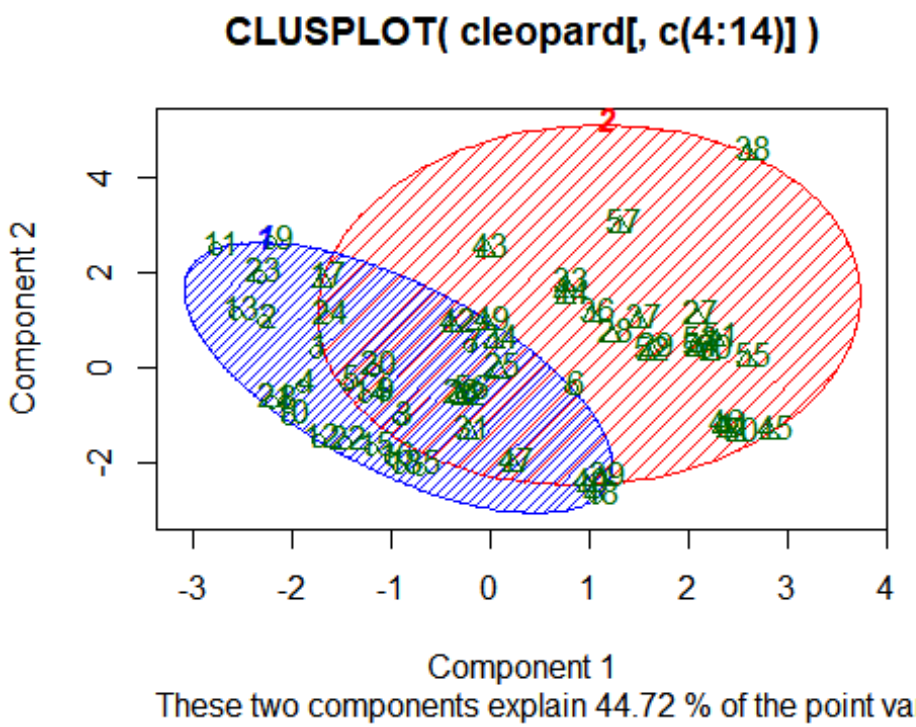



```
par(mar=c(5, 4, 4, 2) + 0.1)
```

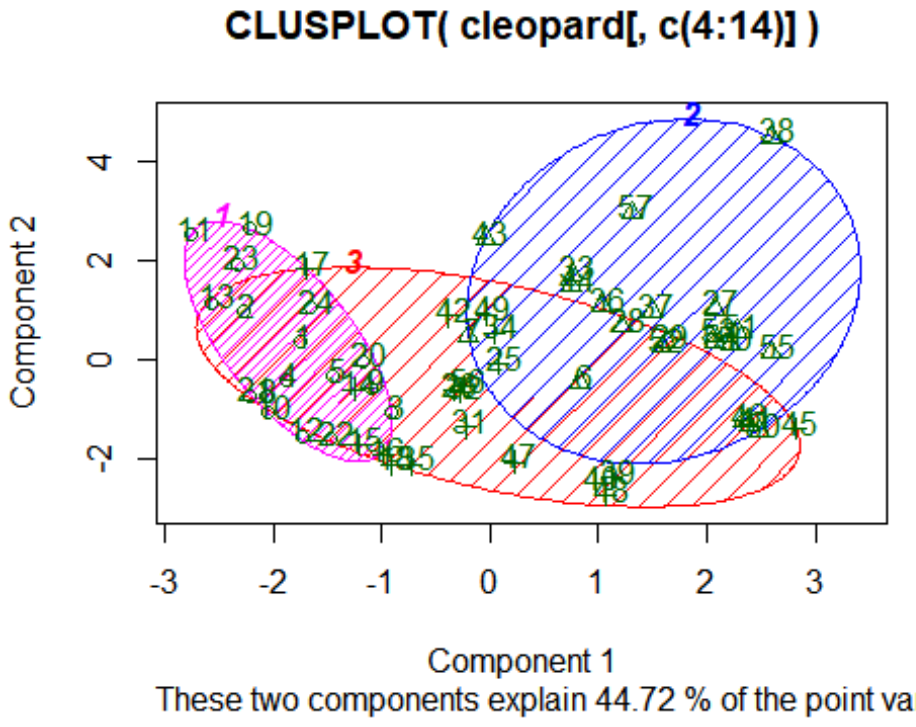
Muchas observaciones tienen coeficiente negativo. Parece que el algoritmo PAM no funciona muy bien con estos datos. En todo caso, es mejor con 2 conglomerados.

¿Tiene sentido aplicar aquí la función clusplot() del paquete cluster? Identificar cada conglomerado del mejor conjunto con su área de localización mayoritaria.

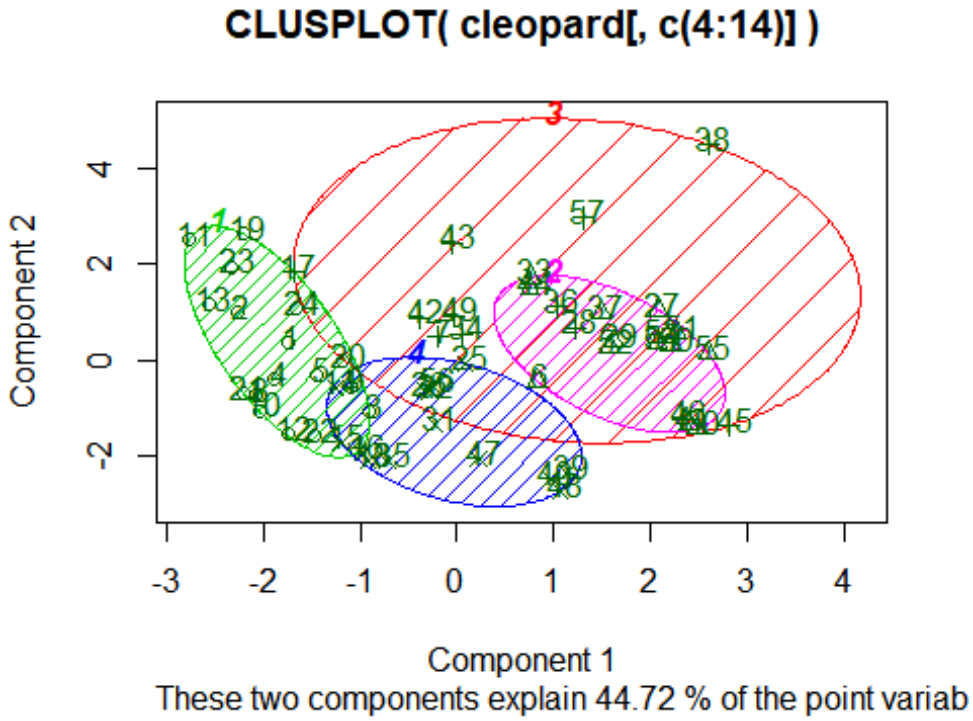
```
pam2 <- pam(cleopard[,c(4:14)],k=2, diss = F)
clusplot(cleopard[,c(4:14)], pam2$clustering, color=TRUE, shade=TRUE,
labels=2, lines=0)
```




```
pam3<- pam(cleopard[,c(4:14)],k=3, diss = F)
clusplot(cleopard[,c(4:14)], pam3$clustering, color=TRUE, shade=TRUE,
labels=2, lines=0)
```



```
pam3<- pam(cleopard[,c(4:14)],k=4, diss = F)
clusplot(cleopard[,c(4:14)], pam3$clustering, color=TRUE, shade=TRUE,
labels=2, lines=0)
```



Esta función representa las 2 primeras componentes principales. En este caso, estas 2 componentes explican menos del 50% de la variabilidad. Visualmente está claro que no son buenas representaciones.

Ejercicio 3.

Nota: Si el análisis da algún error, habrá que eliminar la variable responsable.

1. Realizar un análisis lineal discriminante respecto a los dos grupos (indonesias y continentales).

```
cleopard$`Location Code`<- as.numeric(cleopard$`Location Code`)
data.split<- split(cleopard, cleopard$Group)
g1 <- as.data.frame(data.split[1])
g2 <- as.data.frame(data.split[2])
g1.means <- colMeans(g1[,c(3:14)])
g2.means <- colMeans(g2[,c(3:14)])
g1.cov <- cov(g1[,c(3:14)])
g2.cov <- cov(g2[,c(3:14)])
n1 <- length(g1$CONT.Group)
n2 <- length(g2$INDO.Group)
n <- n1 + n2
```

```
#matriz de covarianzas común:
pooled.cov <- ((n1-1)*g1.cov + (n2-1)*g2.cov) / (n-2)
inv.pooled.cov <- Ginv(pooled.cov)
#establecemos las probabilidades a priori
#calculamos las puntuaciones y las probabilidades a posteriori.
prior.p <- c(n1,n2)/n
a0 <- (-1/2)*c(t(g1.means) %*% inv.pooled.cov %*% g1.means,
t(g2.means) %*% inv.pooled.cov %*% g2.means)
coef.fun <- function(x, means){
  t(means) %*% inv.pooled.cov %*% x
  }
scores <- matrix(numeric(n*2), ncol=2)
scores[,1] <- apply(cleopard[,c(3:14)],1,coef.fun,means=g1.means)
scores[,2] <- apply(cleopard[,c(3:14)],1,coef.fun,means=g2.means)
scores <- t(apply(scores,1,function(x){x+log(prior.p)+a0}))
posterior <- t(apply(scores,1,function(x) exp(x)/sum(exp(x))))
head(posterior)

##              [,1]      [,2]
## [1,] 5.480227e-06 0.9999945
## [2,] 1.903239e-06 0.9999981
## [3,] 1.440845e-05 0.9999856
## [4,] 9.123848e-07 0.9999991
## [5,] 7.412225e-06 0.9999926
## [6,] 2.902846e-04 0.9997097
```

La mayor puntuación a posteriori determina la predicción:

```
(pred <- apply(posterior,1,which.max))

##  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 2 1
1 1 1 1 1
## [39] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Con la función lda():

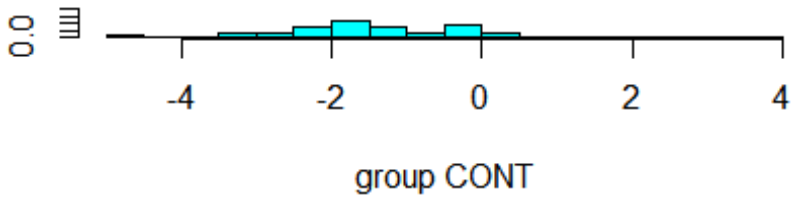
```
(leopard.lda <- lda(cleopard$Group~cleopard$`Location Code`+
                     cleopard$`Cloud Spots`+cleopard$Lightness+
                     cleopard$Brightness+cleopard$Coloration+
                     cleopard$`Dorsal Stripe`+cleopard$`Neck Stripes`+
                     cleopard$`Shoulder Patter`, data=cleopard))

## Call:
## lda(cleopard$Group ~ cleopard$`Location Code` + cleopard$`Cloud Spots`
+
##     cleopard$Lightness + cleopard$Brightness + cleopard$Coloration +
##     cleopard$`Dorsal Stripe` + cleopard$`Neck Stripes` +
cleopard$`Shoulder Patter`,
##     data = cleopard)
##
## Prior probabilities of groups:
##      CONT      INDO
```

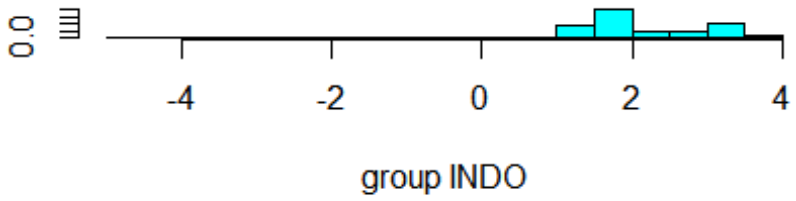
```
## 0.5789474 0.4210526
##
## Group means:
##      cleopard$`Location Code` cleopard$`Cloud Spots`
cleopard$Lightness
## CONT                 3.848485              1.515152
2.090909
## INDO                 1.666667              2.250000
1.854167
##      cleopard$Brightness cleopard$Coloration cleopard$`Dorsal Stripe`
## CONT             2.30303            3.151515                 2.313544
## INDO             2.06250            2.500000                 2.683673
##      cleopard$`Neck Stripes` cleopard$`Shoulder Patter`
## CONT                1.996454                   1.770606
## INDO                2.156472                   1.864167
##
## Coefficients of linear discriminants:
##                                      LD1
## cleopard$`Location Code`   -1.549058750
## cleopard$`Cloud Spots`      0.006439627
## cleopard$Lightness         -0.524943901
## cleopard$Brightness         0.170873891
## cleopard$Coloration        -0.096221236
## cleopard$`Dorsal Stripe`    0.351535895
## cleopard$`Neck Stripes`     0.365916520
## cleopard$`Shoulder Patter` -0.084129536
```

```r
#se elimina la variable Colud Size porque nos da un error
pred.lda <- predict(leopard.lda)
```

La función discriminante es: (-1.55)x Location Code + 0.006 x Cloud Spots - 0.525 x Lightness+...-0.084 x Shoulder Patter

```r
ldahist(data = pred.lda$x[,1], g= cleopard$Group)
```






```r
head(pred.lda$posterior)
```

```
##           CONT      INDO
## 1 4.823122e-05 0.9999518
## 2 1.369155e-05 0.9999863
## 3 1.827165e-05 0.9999817
## 4 9.193868e-06 0.9999908
## 5 7.958861e-05 0.9999204
## 6 3.908210e-05 0.9999609
```

```r
#los coeficientes del discriminador lineal de Fisher son proporcioneales
a <- as.numeric(inv.pooled.cov %*% (g2.means - g1.means))
a / as.vector(sqrt(t(a) %*% pooled.cov %*% a))
```

```
##  [1] -1.42729895  0.00000000  0.14550584 -0.86523764  0.61628720
0.27973340
##  [7]  0.25655262  0.61812735 -0.10518276  0.38764042 -1.82727877 -
0.05620862
```

```r
coef(leopard.lda)
```

```
##                                    LD1
## cleopard$`Location Code`    -1.549058750
## cleopard$`Cloud Spots`       0.006439627
## cleopard$Lightness          -0.524943901
## cleopard$Brightness          0.170873891
## cleopard$Coloration         -0.096221236
## cleopard$`Dorsal Stripe`     0.351535895
```

```
## cleopard$`Neck Stripes`      0.365916520
## cleopard$`Shoulder Patter` -0.084129536
```

Dibujar algún gráfico elegante de los scores y calcular la tabla de confusión. Evaluar de algún modo la clasificación.

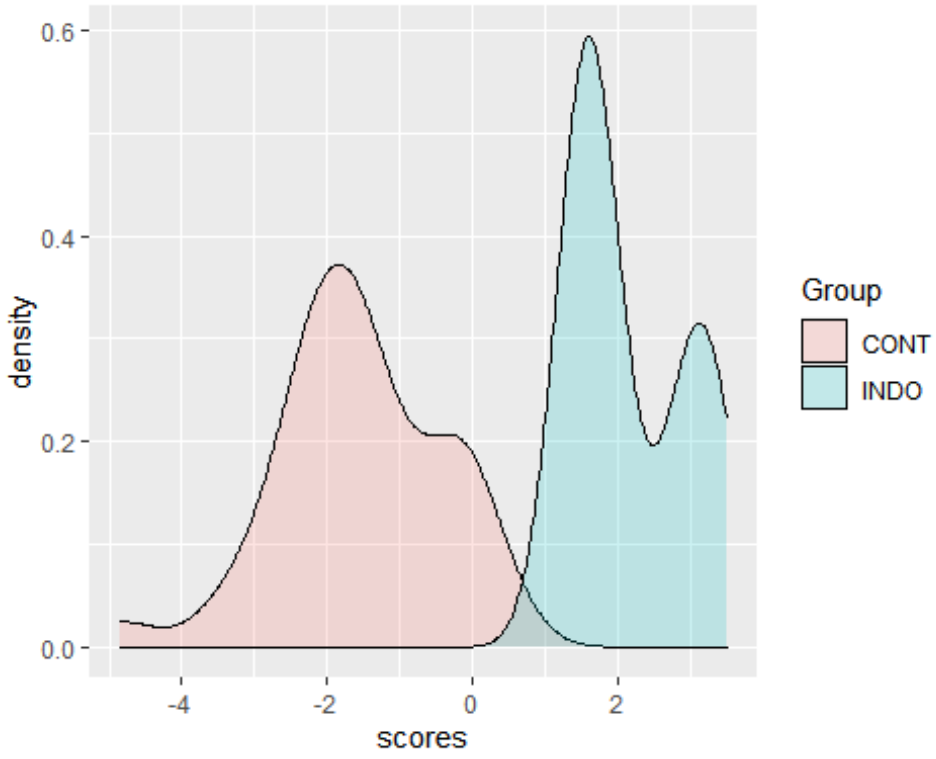
```
# tabla de confusión:
(conf<-table(pred.lda$class, real=cleopard$Group))
```

```
##       real
##        CONT INDO
##   CONT   32    0
##   INDO    1   24
```

```
# error de clasificación
1 - sum(diag(conf))/sum(conf)
```

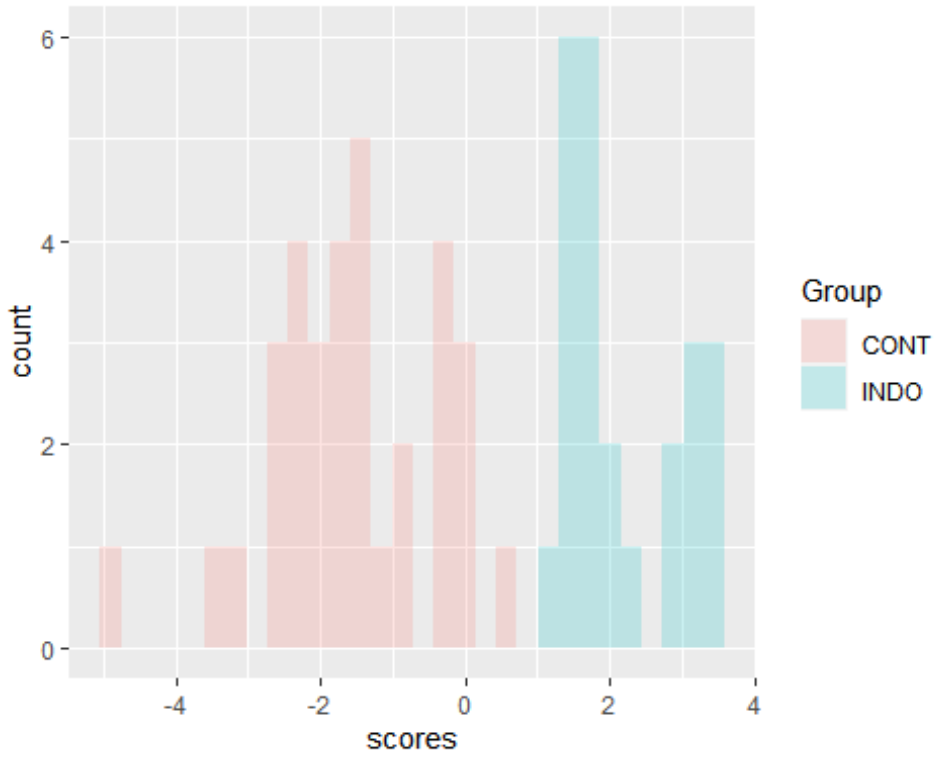
```
## [1] 0.01754386
```

```
ggplot(cleopard, aes(pred.lda$x, fill = Group)) +
geom_density(alpha = 0.2) + xlab("scores")
```



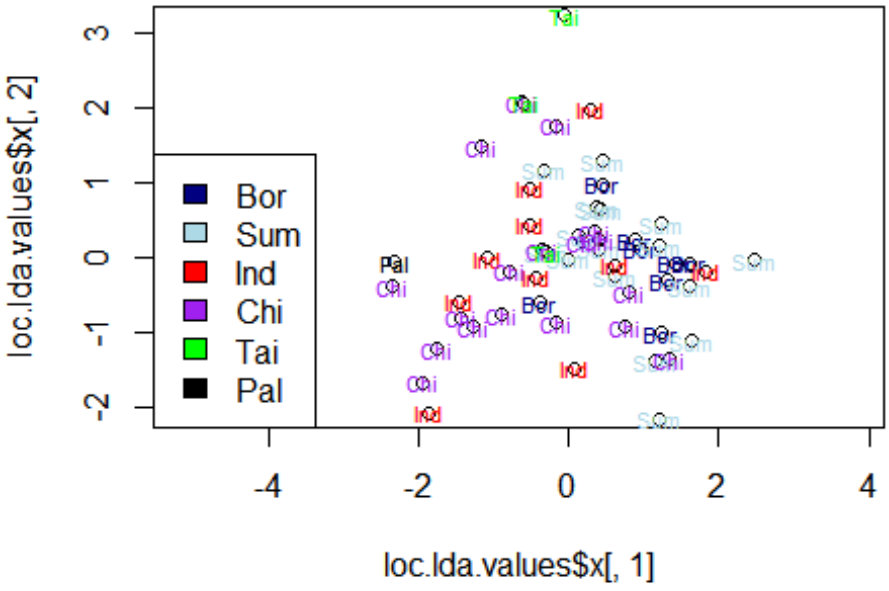

```
ggplot(cleopard, aes(pred.lda$x, fill = Group)) +
geom_histogram(alpha = 0.2) + xlab("scores")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



2. Dibujar el gráfico B de la figura 2 del trabajo donde se presenta el análisis lineal discriminante respecto a las 6 áreas de origen.

```
loc.lda<-lda(cleopard$`Location Code`~cleopard$`Cloud
Spots`+cleopard$Lightness+
                    cleopard$Brightness+cleopard$Coloration+
                    cleopard$`Dorsal Stripe`+cleopard$`Neck Stripes`+
                    cleopard$`Shoulder Patter`, data=cleopard)
loc.lda.values <- predict(loc.lda)

p<- eqscplot(loc.lda.values$x[,1],loc.lda.values$x[,2])
text(loc.lda.values$x[,1],loc.lda.values$x[,2], label=grups
,col=colores,cex=.7)
legend("bottomleft",fill=c("navyblue","lightblue","red","purple","green",
"black"),legend=levels(grups))
```


Bor
Sum
Ind
Chi
Tai
Pal
loc.lda.values$x[, 2]
loc.lda.values$x[, 1]
-4
-2
0
2
4
-2
-1
0
1
2
3