prova2_est_mvt

Gustavo Alovisi

17/01/2021

Exercício 1: O arquivo "BD1" trás um banco de dados com medidas da mandíbula de 77 cães tailandeses de 5 raças distintas. As nove variáveis estão descritas no arquivo R "Arquivo prova2 2020":

```
##############################################################################
#################################QUESTAO 1#####################################
#x1: Comprimento da mandíbula
#x2: Largura da mandíbula abaixo do primeiro molar
#x3: Largura do côndilo articular
#x4: Altura da mandíbula abaixo do primeiro molar
#x5: Comprimento do primeiro molar
#x6: Largura do primeiro molar
#x7: Comprimento do primeiro ao terceiro molar
#x8: Comprimento do primeiro ao quarto premolar
#x9: Largura do canino inferior


dados_full <- read.table("BD1.txt", header=T,sep="")

#dados <- dados[,-1]
dados <- dados_full[,2:10]

#Analise da matriz de covariancia:
cov(dados)
```

```
##           X1        X2        X3        X4        X5         X6        X7
## X1 487.65960 20.469788 57.892174 49.387560 43.044087 16.2421053 60.512474
## X2  20.46979  1.968462  3.953213  4.235048  2.869481  1.2147368  3.277085
## X3  57.89217  3.953213 12.839371  9.379187  6.947027  2.6407895  7.145762
## X4  49.38756  4.235048  9.379187 11.330144  6.285885  2.8118421  6.655502
## X5  43.04409  2.869481  6.947027  6.285885  6.200615  2.1763158  7.713944
## X6  16.24211  1.214737  2.640789  2.811842  2.176316  1.0478947  2.757895
## X7  60.51247  3.277085  7.145762  6.655502  7.713944  2.7578947 17.410800
## X8  78.13500  4.610629 11.468558 10.656699  9.627649  3.6000000 14.459159
## X9  16.84556  1.267498  2.727717  2.831818  2.236688  0.9314474  2.751777
##           X8         X9
## X1 78.134997 16.8455571
## X2  4.610629  1.2674983
## X3 11.468558  2.7277170
## X4 10.656699  2.8318182
## X5  9.627649  2.2366883
```

```
## X6  3.600000  0.9314474
## X7 14.459159  2.7517772
## X8 19.401572  3.7569036
## X9  3.756904  1.0357621
```

a) Realize a ACP (sem rotação) e apresente os 2 maiores autovalores, os autovetores associados e opercentual acumulado da variação explicada. Interprete a estrutura de correlação dos dados usandoos 2 "primeiros" autovetores:

Vamos rodar um PCA com SCALE = TRUE (na matriz de correlação) dado as diferenças de escala entre as variáveis:

```
pc<-prcomp(dados, scale.= TRUE)
```

Dois maiores autovalores (variâncias):

```
eigenvalues <- pc$sdev^2
head(eigenvalues,2)
```

```
## [1] 7.0507296 0.7417013
```

Vamos obter os autovetores associados a estes autovalores:

```
pc$rotation[,c(1,2)]
```

```
##          PC1         PC2
## X1 0.3195359  0.21017335
## X2 0.3424583 -0.32513275
## X3 0.3198482 -0.28580628
## X4 0.3291507 -0.42690620
## X5 0.3549591  0.12508534
## X6 0.3463203 -0.08150759
## X7 0.2808105  0.69137239
## X8 0.3450501  0.27995646
## X9 0.3551455 -0.08672077
```

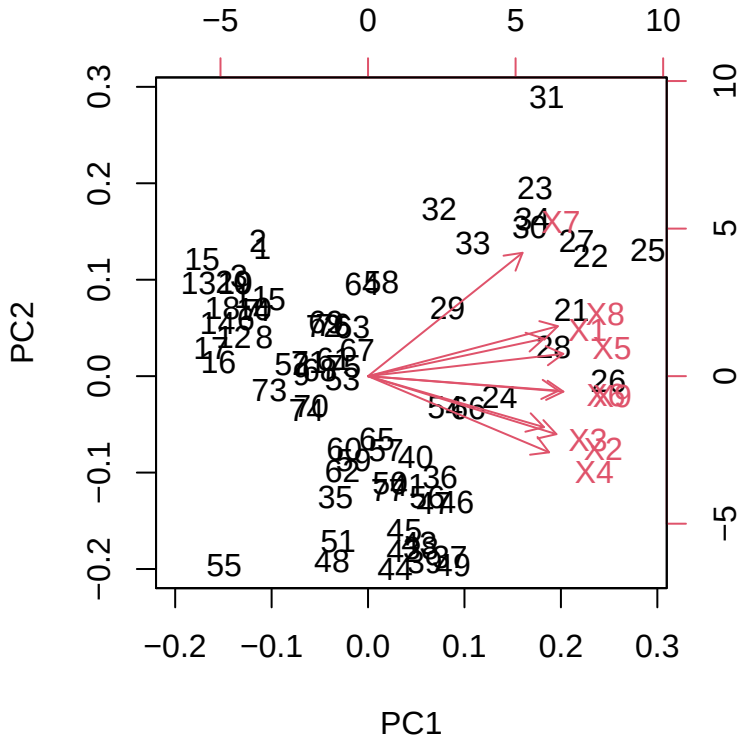
O percentual da variância explicada pelos dois primeiros componentes é obtida através da função summary:

```
summary(pc)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.6553 0.86122 0.62915 0.47236 0.42505 0.37042 0.34688
## Proportion of Variance 0.7834 0.08241 0.04398 0.02479 0.02007 0.01525 0.01337
## Cumulative Proportion  0.7834 0.86583 0.90981 0.93460 0.95467 0.96992 0.98329
##                           PC8     PC9
## Standard deviation     0.29497 0.25180
## Proportion of Variance 0.00967 0.00704
## Cumulative Proportion  0.99296 1.00000
```

Podemos notar que a proporção cumulativa da variância é de 0.8658 para PC1 e PC2.

```
biplot(pc)
```



Através da análise dos dois primeiros componentes, podemos notar que o PC1 é a média ponderada das variáveis X1...X9. Porém, o PC2 faz uma distinção entre as variáveis de Largura e as variáveis de Comprimento.

b) 'Ordene os cães baseados nos escores do primeiro CPs e apresente a raça dos 20 com asmaiores medidas. Esse ordenamento faz algum sentido? Justifique!

Vamos ordenar os cães com base no Primeiro Componente, que é uma média ponderada das variáveis X1..X9. Os valores calculados do primeiro componente são obtidos através de pc$x[,1]:

```
n <- nrow(dados_full)
ss <- data.frame(ord=seq(1,n,by=1), y=pc$x[,1], raca = dados_full[,11])
ss_ord = ss[order(ss[,2], decreasing=T),]
head(ss_ord, 20)
```

```
##    ord        y                raca
## 25  25 6.760645 LobosIndianos
## 26  26 5.807686 LobosIndianos
## 22  22 5.380140 LobosIndianos
## 27  27 5.041421 LobosIndianos
## 21  21 4.921569 LobosIndianos
## 28  28 4.480623 LobosIndianos
## 31  31 4.321046 LobosIndianos
## 23  23 4.034656 LobosIndianos
## 34  34 3.974399 LobosIndianos
## 30  30 3.909737 LobosIndianos
## 24  24 3.180130 LobosIndianos
## 33  33 2.533067 LobosIndianos
```

```
## 66  66 2.420571 cãesmodernosdaTailândia
## 46  46 2.135198                   Cuons
## 49  49 2.054242                   Cuons
## 37  37 1.976586                   Cuons
## 29  29 1.919607           LobosIndianos
## 54  54 1.867634    cãespréhisttailandes
## 36  36 1.745890                   Cuons
## 32  32 1.719157           LobosIndianos
```

Através do ordenamento utilizando o PC1, percebemos que a raça mais presente é a de Lobos Indianos (14/20 obs). Como o PC1 é uma média ponderada das variáveis X1...X9 e seus valores, esta raça tem uma predominância de observações com valores altos para as variáveis mencionadas em relação à outras raças.

Para checar se esta análise faz sentido, podemos comparar a média geral das observações para X1..X9 com a média dos Lobos Indianos:

```
#média geral de X1..X9
dados_full %>% summarize_at(vars(X1:X9), mean) %>% round(2)
```

```
##      X1   X2    X3    X4    X5 X6    X7   X8   X9
## 1 127.7 9.96 21.95 21.45 20.49  8 32.52 37.4 6.08
```

```
#média dos Lobos Indianos para X1..X9:
dados_full %>% filter(Raça == 'LobosIndianos') %>% summarize_at(vars(X1:X9), mean) %>% round(2)
```

```
##       X1    X2    X3    X4    X5   X6    X7    X8   X9
## 1 157.64 11.58 26.21 24.71 24.71 9.34 40.21 44.79 7.41
```

De fato, podemos perceber que as médias de X1..X9 para Lobos Indianos são maiores que a média geral, corroborando nossa hipótese.

c) Realize a AF com 2 fatores (utilizando extração das cargas via máxima verossimilhança e rotação varimax) e apresente as cargas fatoriais, as variâncias dos fatores, o percentual acumulado da variação explicada, as comunalidades e a variância não explicada de cada variável. Interprete a estrutura de correlação dos dados usando os 2 primeiros fatores.

```
facAnalysis <- factanal(dados_full[2:10], factors = 2, rotation = "varimax", scores = "regression")
facAnalysis
```

```
##
## Call:
## factanal(x = dados_full[2:10], factors = 2, scores = "regression",     rotation = "varimax")
##
## Uniquenesses:
##    X1    X2    X3    X4    X5    X6    X7    X8    X9
## 0.300 0.076 0.310 0.134 0.103 0.162 0.282 0.110 0.103
##
## Loadings:
##    Factor1 Factor2
## X1 0.476   0.688
## X2 0.880   0.387
## X3 0.695   0.455
## X4 0.871   0.327
## X5 0.611   0.723
## X6 0.723   0.561
```

```
## X7 0.256   0.807
## X8 0.505   0.797
## X9 0.748   0.581
##
##                Factor1 Factor2
## SS loadings      4.019   3.400
## Proportion Var   0.447   0.378
## Cumulative Var   0.447   0.824
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 35.92 on 19 degrees of freedom.
## The p-value is 0.0108
```

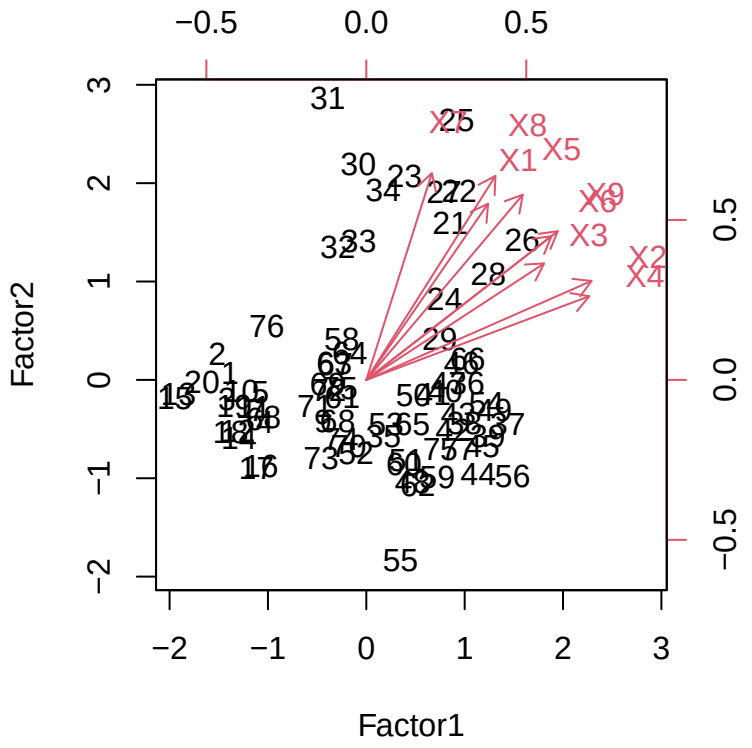
Acima, as cargas fatoriais do F1 e F2 são dadas pela lista Loadings. Proportional Var é a variância de cada fator (0.447, 0.378), com Variância Cumulativa dos Fatores de 0.824. A variância de cada variável não explicada pelos fatores é dada pela lista Uniquenesses. A comunalidade é simplesmente 1 - Uniquenesses:

```
Comunalidade <- 1 - facAnalysis$uniquenesses
Comunalidade
```

```
##        X1        X2        X3        X4        X5        X6        X7        X8
## 0.6999506 0.9239858 0.6896857 0.8658043 0.8967965 0.8379777 0.7176077 0.8895776
##        X9
## 0.8969007
```

Vamos agora plotar o biplot dos fatores 1 e 2 e realizarmos uma análise dos resultados. A rotação varimax busca criar fatores ortogonais não correlacionados entre si.

```
x = facAnalysis$scores[,1:2]
    # Get the loadings on the first two factors
y = facAnalysis$loadings[,1:2]
biplot(x,y)
```



Analisando as cargas fatoriais e o plot podemos perceber que o Fator 1 fez uma maior distinção entre as variáveis de altura x comprimento do que o PCA. O Fator 2 se assemelha de certa forma ao PC2 do PCA, que também busca realizar uma distinção entre essas variáveis.

Exercício 2. O banco "nacoes2" trás um banco de dados contendo variáveis sócio demográficasde 20 países em um determinado período.

```
dadosFull <- read.table("nacoes2.txt", header=T,sep="")
glimpse(dadosFull)
```

```
## Rows: 20
## Columns: 9
## $ Country   <chr> "Argentina", "Australia", "Brazil", "Canada", "Chile", "D...
## $ popu      <int> 33900, 17800, 156600, 29100, 14000, 5200, 58000, 81200, 5...
## $ density   <dbl> 12.0, 2.3, 18.0, 2.8, 18.0, 120.0, 105.0, 227.0, 188.0, 3...
## $ urban     <int> 86, 85, 75, 77, 85, 85, 73, 85, 69, 77, 89, 77, 49, 78, 8...
## $ lifeexpf  <int> 75, 80, 67, 81, 78, 79, 82, 79, 81, 82, 81, 70, 68, 81, 8...
## $ lifeexpem <int> 68, 74, 57, 74, 71, 73, 74, 73, 74, 76, 75, 66, 62, 74, 7...
## $ literacy  <int> 95, 100, 81, 97, 93, 99, 99, 99, 97, 99, 99, 62, 76, 95, ...
## $ pop.incr  <dbl> 1.3, 1.4, 1.3, 0.7, 1.7, 0.1, 0.5, 0.4, 0.2, 0.3, 0.6, 3....
## $ babymort  <dbl> 25.6, 7.3, 66.0, 6.8, 14.6, 6.6, 6.7, 6.5, 7.6, 4.4, 6.3,...
```

Vamos primeiro padronizar os dados para trabalhar com a clusterização:

```
dados <- scale(dadosFull[,2:9])
round(head(dados),2)
```

```
##       popu density urban lifeexpf lifeexpem literacy pop.incr babymort
## [1,] -0.46   -0.81  0.78    -0.28     -0.36     0.24     0.30     0.27
## [2,] -0.70   -0.89  0.70     0.48      0.58     0.60     0.40    -0.56
## [3,]  1.41   -0.75 -0.03    -1.49     -2.08    -0.79     0.30     2.09
## [4,] -0.53   -0.89  0.11     0.63      0.58     0.38    -0.33    -0.58
## [5,] -0.76   -0.75  0.70     0.17      0.11     0.09     0.71    -0.23
## [6,] -0.89    0.14  0.70     0.33      0.42     0.53    -0.95    -0.59
```

a) Proceda a uma Análise de Cluster dos países utilizando dois métodos hierárquicos (o complete linkage e o de Ward) e proponha um agrupamento. Descreva os clusters.

(i) Complete Linkage:

Matriz de Distâncias:

```
dist <- dist(dados, method = "euclidean")
matriz_dist <- as.dist(dist, diag = T)
```

Gerando a árvore:

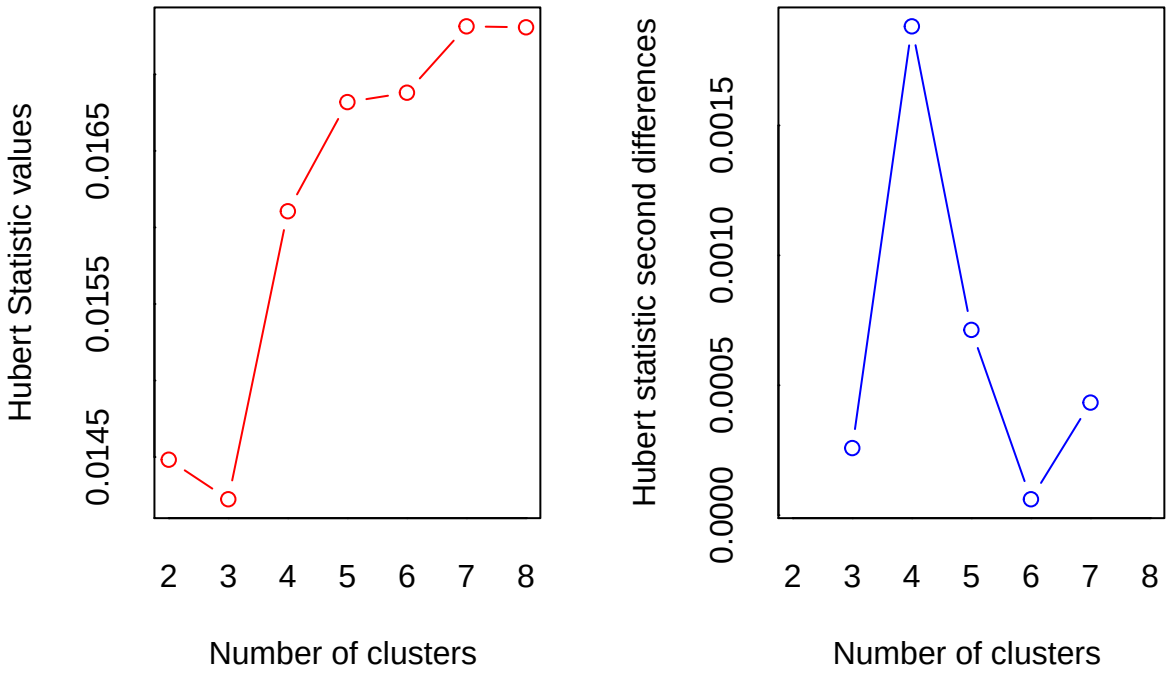
```
Complete_CL <- hclust(matriz_dist, method = "complete")
Complete_CL$height
```

```
##  [1] 0.5540004 0.6560413 0.6882722 0.8953409 1.1137463 1.3569840 1.4394765
##  [8] 1.5332680 1.6471533 1.9612808 2.1128524 2.5380001 2.6285577 2.9653131
## [15] 3.5925889 3.8719186 4.5857683 4.9377872 8.6884554
```

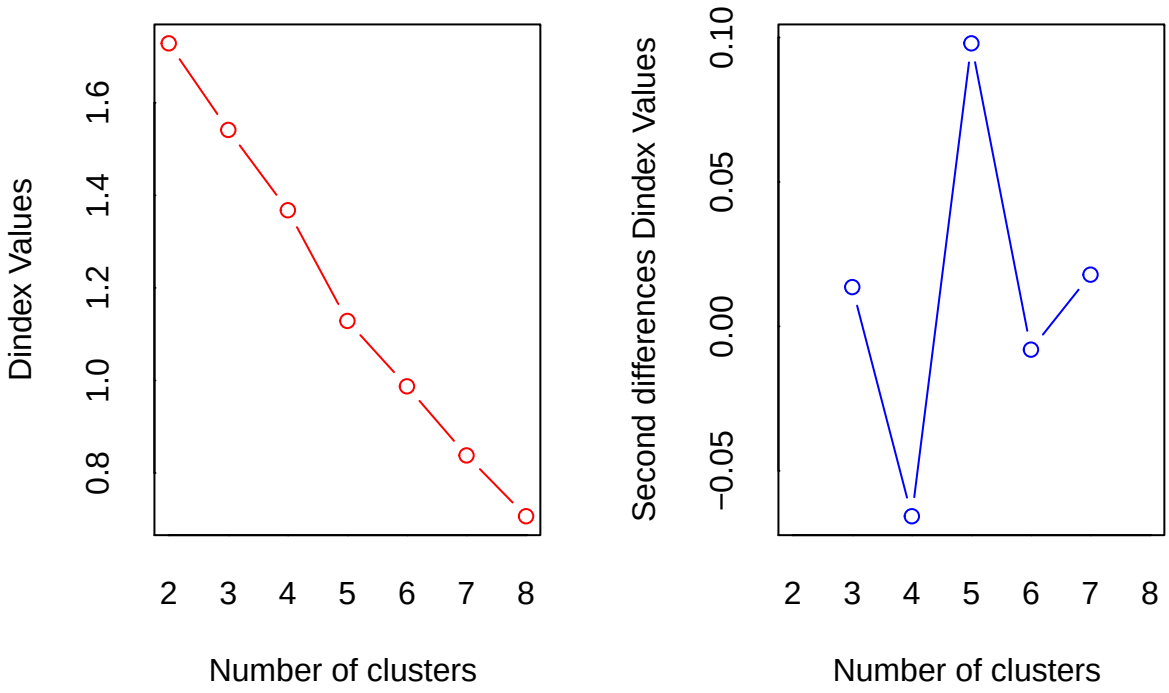
```
cluster_an <- NbClust::NbClust(dados, distance = "euclidean", min.nc = 2,
                               max.nc = 8, method = "complete",
                               index =  "all", alphaBeale = 0.1)
```

```
## Warning in pf(beale, pp, df2): NaNs produzidos
```

```
## Warning in pf(beale, pp, df2): NaNs produzidos
```



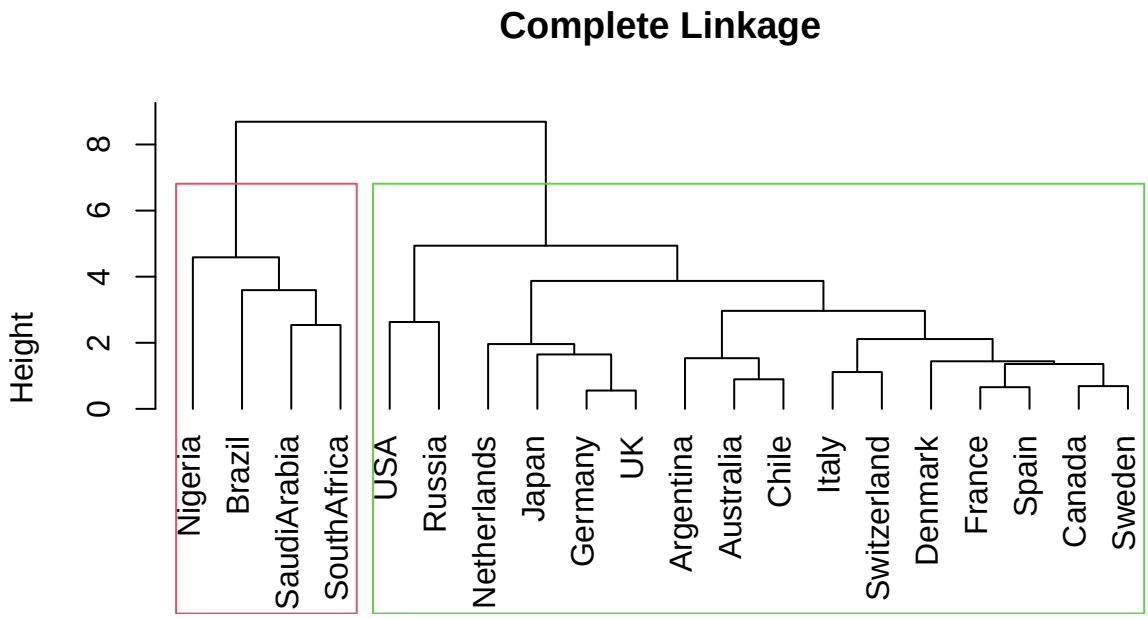

```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 9 proposed 2 as the best number of clusters
## * 3 proposed 3 as the best number of clusters
## * 3 proposed 4 as the best number of clusters
## * 3 proposed 5 as the best number of clusters
## * 1 proposed 6 as the best number of clusters
## * 1 proposed 7 as the best number of clusters
## * 4 proposed 8 as the best number of clusters
##
##                   ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

Para tomar a decisão do número de clusters a serem utilizados, analisamos o Hubert Index e o D Index. O Hubert Index procura um ponto de grande variação na sua medida, assim como D Index. Para o método de Complete Linkage, o número de clusters sugeridos foi de k=2. Assim, vamos plotar a árvore final com 2 clusters:

```
plot(Complete_CL, labels=dadosFull$Country, main = "Complete Linkage", hang=-1)
rect.hclust(Complete_CL, k=2, border = 2:4)
```



Podemos ver que o primeiro cluster (cluster 2) é constituido dos países Brazil, Nigéria, Saudi Arabia e South Africa. O restante dos países foi classificado como pertecendo ao segundo cluster.

Ainda, é possível comparar as médias das observações de cada cluster para ter uma noção do que está sendo diferenciado para cada grupo.

Atribuindo os clusters estimados aos dados:

```
clusters <- cutree(Complete_CL, k=2) #vamos cortar em k = 2, nosso k escolhido.
dadosFull$clusters <- clusters
```

Calculando médias:

```
dadosFull %>% group_by(clusters) %>% summarize_at(vars(popu:babymort), mean)
```

```
## # A tibble: 2 x 9
##   clusters  popu density urban lifeexpf lifeexpem literacy pop.incr babymort
##      <int> <dbl>   <dbl> <dbl>    <dbl>     <dbl>    <dbl>    <dbl>    <dbl>
## 1        1 60100   119.   79.6     79.7      72.9     97.8    0.631     9.59
## 2        2 79150    40.7  59       65.5      59.8     67.5    2.55     60.0
```

Percebemos que o cluster 1 (demais países) apresenta uma média menor de população (popu), com maior densidade, população urbana, expectativa de vida e alfabetização (literacy). O cluster 2 (Brazil, Nigéria, Saudi Arabia e South Africa) apresenta níveis maiores de mortalidade infantil (babymort), crescimento populacional (pop.incr) em relação ao cluster 1.

(ii) Ward

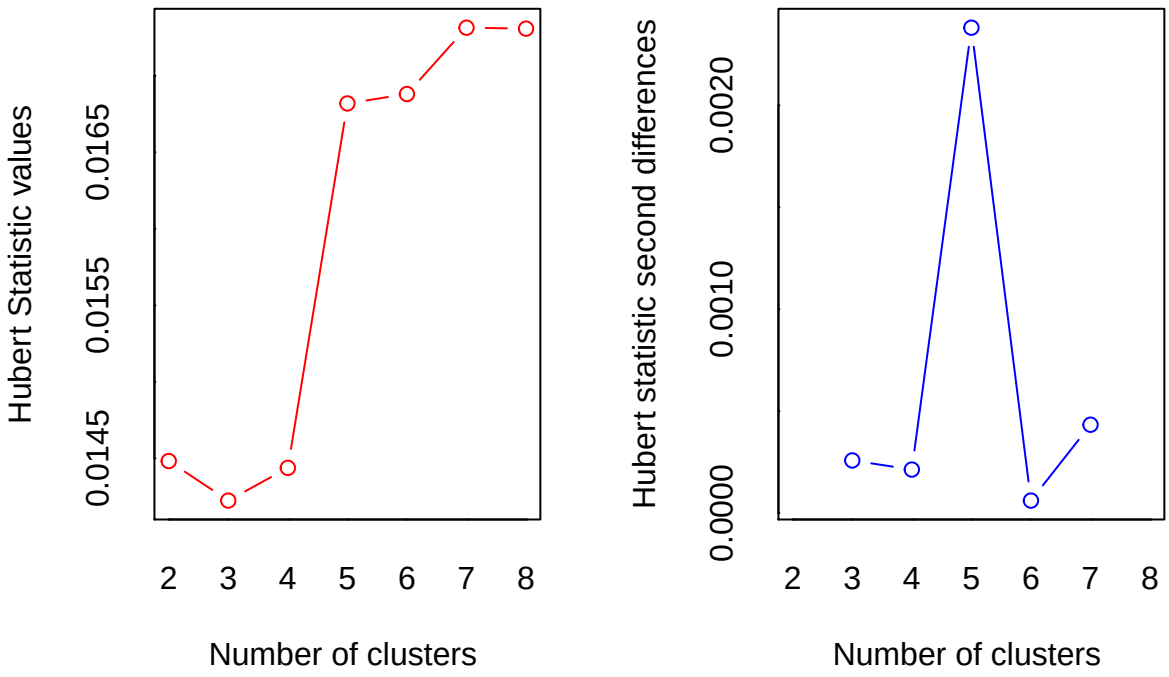
```
Ward_CL <- hclust(matriz_dist, method = "ward.D2")
Ward_CL$height
```

```
##  [1]  0.5540004  0.6560413  0.6882722  0.8953409  1.1137463  1.3059169
##  [7]  1.3945440  1.3978915  1.7527933  1.8629419  2.4404806  2.5380001
## [13]  2.6285577  3.5382221  3.6194942  4.6781677  5.1255341  5.5168259
## [19] 12.8484092
```

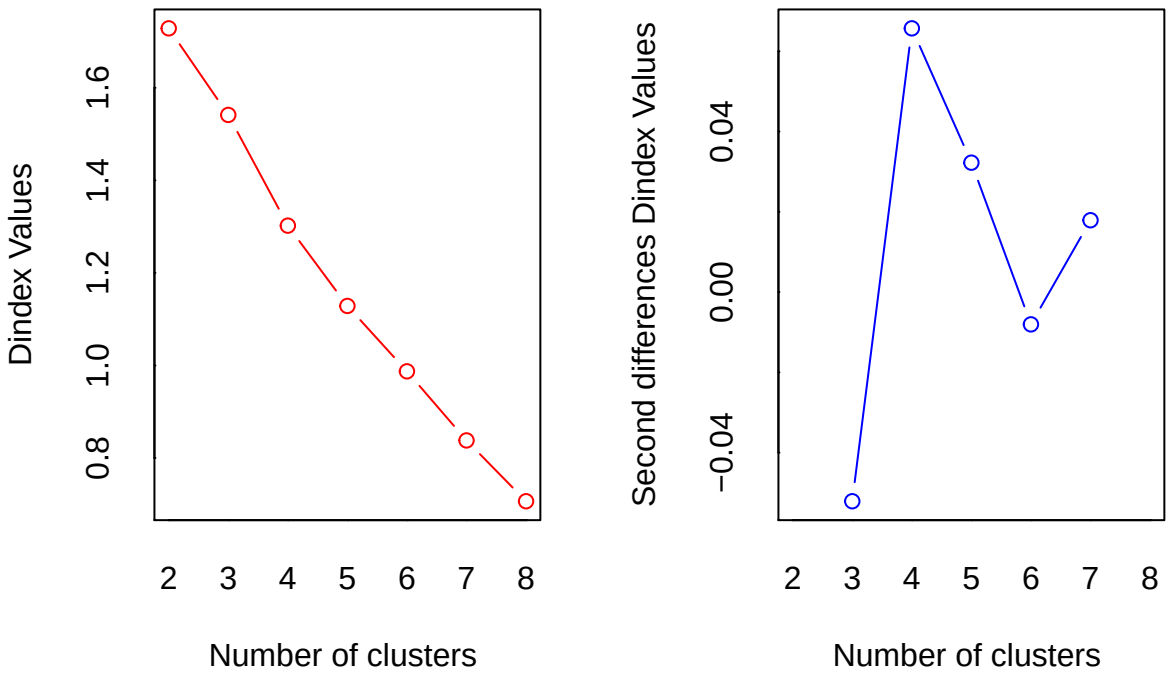
```
cluster_an <- NbClust::NbClust(dados, distance = "euclidean", min.nc = 2,
                               max.nc = 8, method = "ward.D2",
                               index =  "all", alphaBeale = 0.1)
```

```
## Warning in pf(beale, pp, df2): NaNs produzidos
```

```
## Warning in pf(beale, pp, df2): NaNs produzidos
```



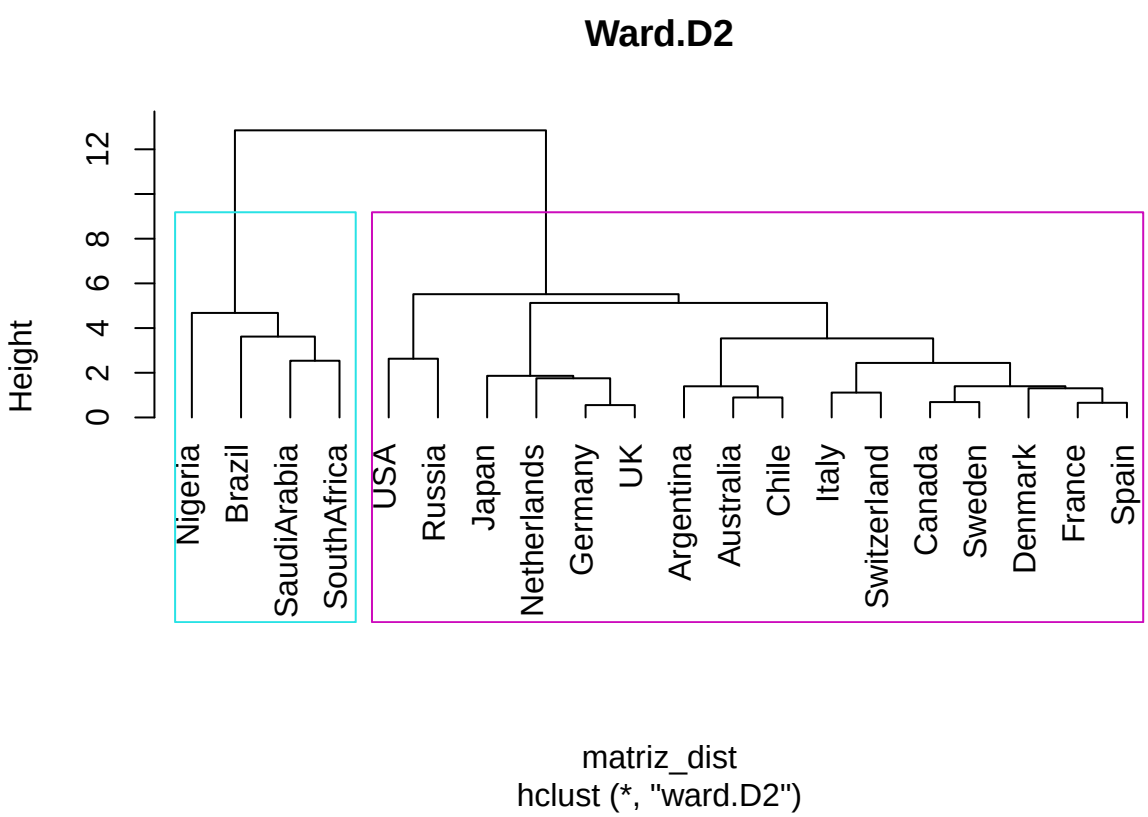

```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 9 proposed 2 as the best number of clusters
## * 4 proposed 3 as the best number of clusters
## * 2 proposed 4 as the best number of clusters
## * 1 proposed 5 as the best number of clusters
## * 2 proposed 6 as the best number of clusters
## * 2 proposed 7 as the best number of clusters
## * 4 proposed 8 as the best number of clusters
##
##                   ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

No caso do método de Ward, o número sugerido de clusters também foi 2. Vamos checar o plot da árvore:

```
plot(Ward_CL, labels=dadosFull$Country, main = "Ward.D2", hang=-1)
rect.hclust(Ward_CL, k=2, border = 5:6)
```



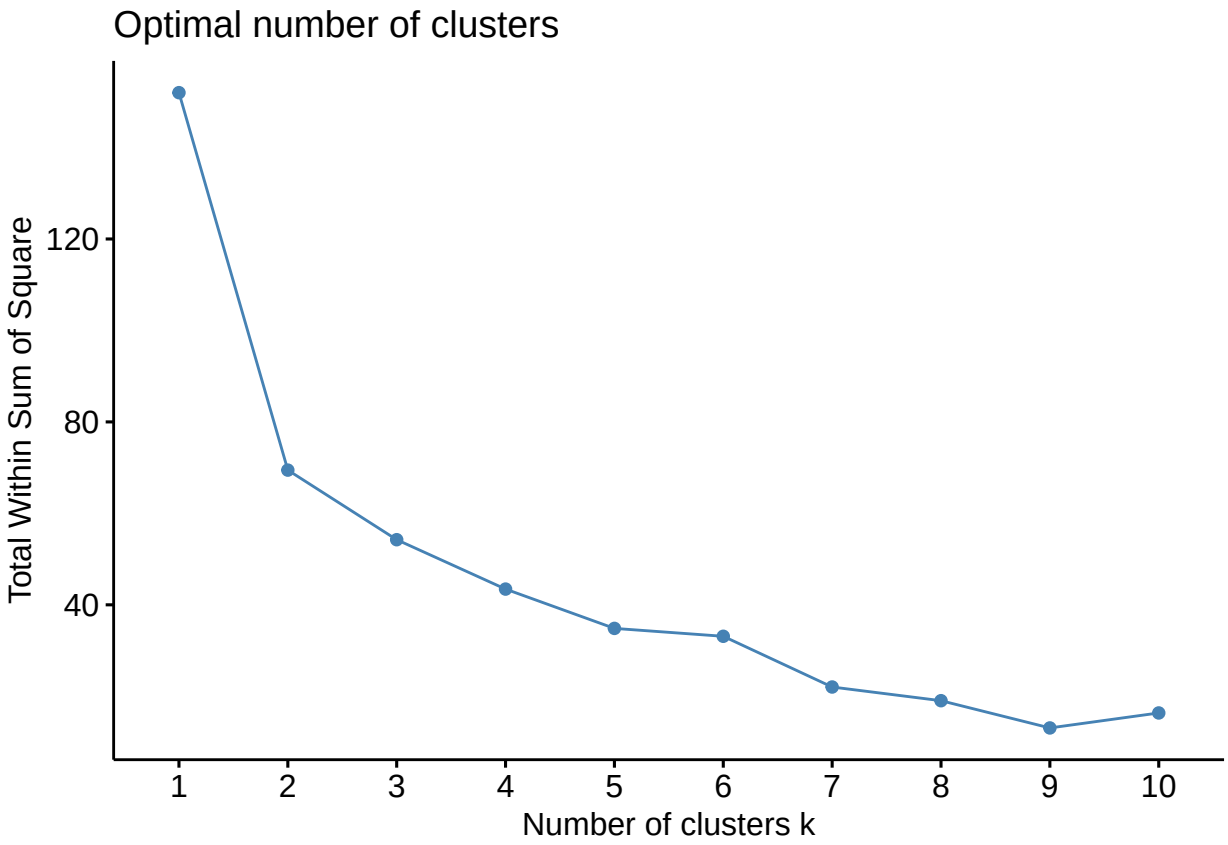
Pelo método de Ward, percebemos a mesma classificação dos países como a classificação do Complete Linkage. Assim, a mesma análise acima em relação aos clusters se aplica.

b) Proceda uma Análise de Cluster dos países utilizando o método k-means [com centroidesiniciais aleatórios e semente set.seed(1)]. Defina um número k de grupos a priori e justifique aescolha de k. Descreva os clusters.

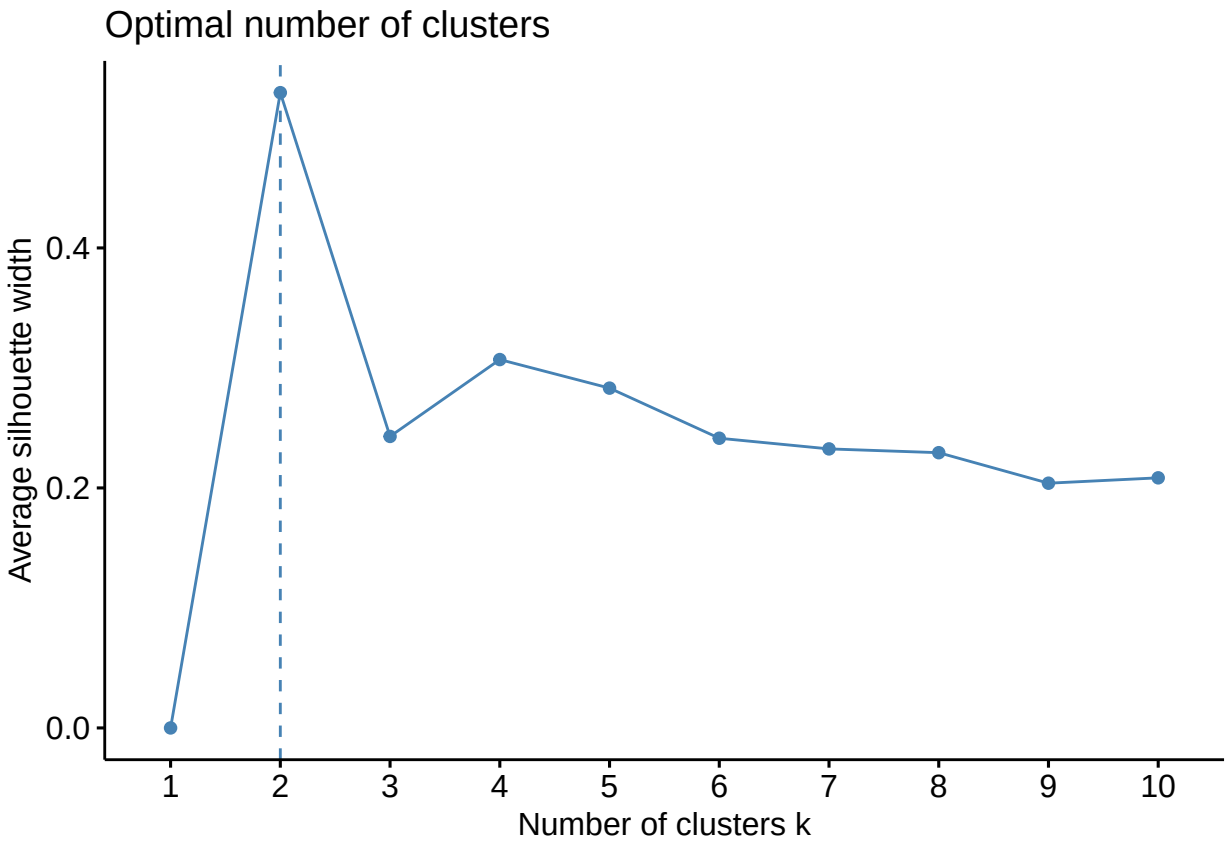
Podemos determinar o k ótimo de diversas formas. Formas famosas na literatura incluem o 'elbow' que apresenta a maior decaída do WSS: Withing Sum of Squares do kmeans, e o valor que maximiza a largura média do Silhouete.

Abaixo, vamos rodar estas análises com a library factoextra.

```
factoextra::fviz_nbclust(dados, kmeans, method = "wss")
```




```
factoextra::fviz_nbclust(dados, kmeans, method = "silhouette")
```



Em ambas as métricas, podemos ver que k = 2 parece ser o número ótimo de clusters do k-means: um decaimento rápido no WSS entre k=1 e k=2 e a maior largura da Silhouette para k=2. Ainda, k=2 foi o número ótimo de clusters encontrados para nossos clusters hierárquicos.

Assim, vamos rodar o k-means com k=2:

```
set.seed(1)
km <- kmeans(dados, 2)
km$cluster
```

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

```
km$centers
```

```
##          popu    density      urban   lifeexpf lifeexpem   literacy   pop.incr
## 1  0.23210135 -0.5537533 -1.2125454 -1.7187523  -1.64848 -1.7726498  1.5973582
## 2 -0.05802534  0.1384383  0.3031363  0.4296881   0.41212  0.4431624 -0.3993395
##     babymort
## 1  1.8198753
## 2 -0.4549688
```

```
km
```

```
## K-means clustering with 2 clusters of sizes 4, 16
##
## Cluster means:
##          popu    density      urban   lifeexpf lifeexpem   literacy   pop.incr
## 1  0.23210135 -0.5537533 -1.2125454 -1.7187523  -1.64848 -1.7726498  1.5973582
## 2 -0.05802534  0.1384383  0.3031363  0.4296881   0.41212  0.4431624 -0.3993395
```

```
##     babymort
## 1  1.8198753
## 2 -0.4549688
##
## Clustering vector:
##  [1] 2 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 1
##
## Within cluster sum of squares by cluster:
## [1] 20.71372 48.74547
##  (between_SS / total_SS =  54.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

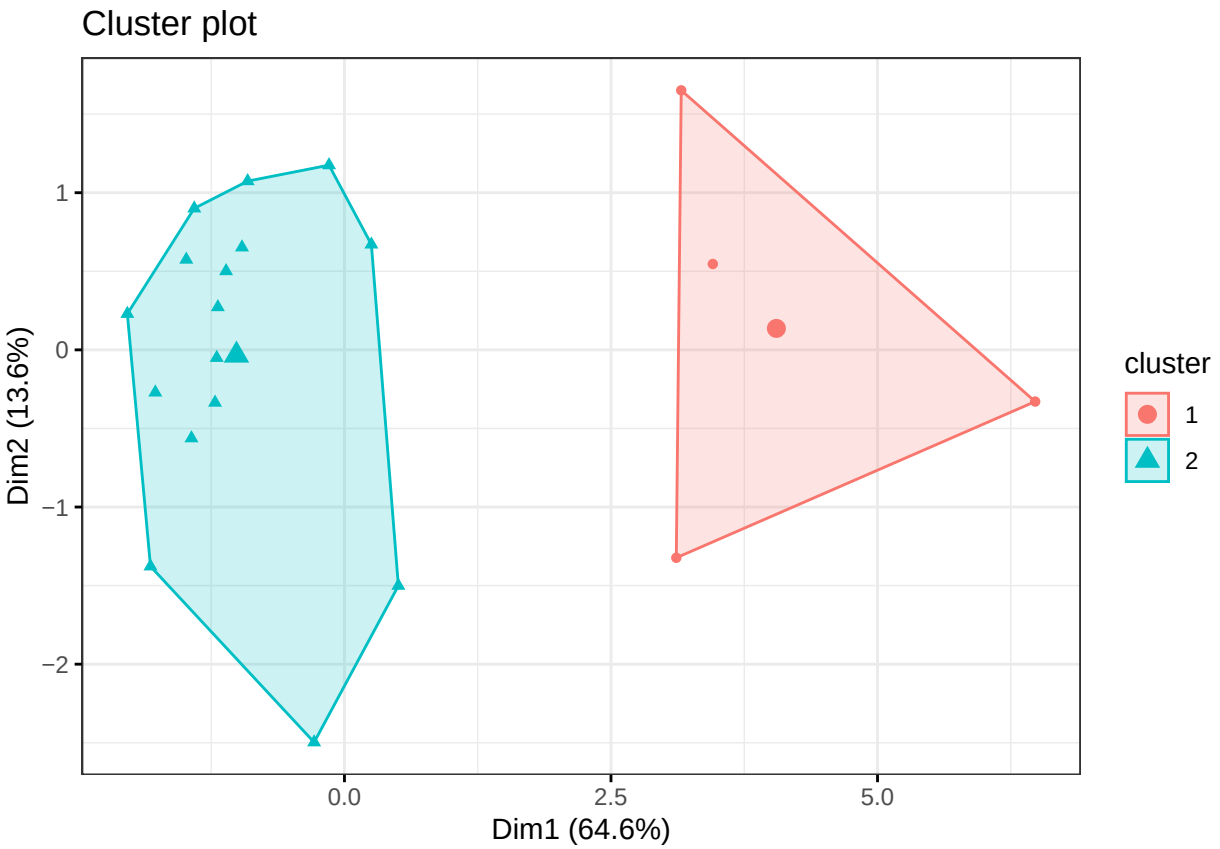
Vamos analisar os clusters atribuídos a nossos dados:

```
dadosFull1 <- cbind(dadosFull, km$cluster)
glimpse(dadosFull1)
```

```
## Rows: 20
## Columns: 11
## $ Country      <chr> "Argentina", "Australia", "Brazil", "Canada", "Chile",...
## $ popu         <int> 33900, 17800, 156600, 29100, 14000, 5200, 58000, 81200...
## $ density      <dbl> 12.0, 2.3, 18.0, 2.8, 18.0, 120.0, 105.0, 227.0, 188.0...
## $ urban        <int> 86, 85, 75, 77, 85, 85, 73, 85, 69, 77, 89, 77, 49, 78...
## $ lifeexpf     <int> 75, 80, 67, 81, 78, 79, 82, 79, 81, 82, 81, 70, 68, 81...
## $ lifeexpem    <int> 68, 74, 57, 74, 71, 73, 74, 73, 74, 76, 75, 66, 62, 74...
## $ literacy     <int> 95, 100, 81, 97, 93, 99, 99, 99, 97, 99, 99, 62, 76, 9...
## $ pop.incr     <dbl> 1.3, 1.4, 1.3, 0.7, 1.7, 0.1, 0.5, 0.4, 0.2, 0.3, 0.6,...
## $ babymort     <dbl> 25.6, 7.3, 66.0, 6.8, 14.6, 6.6, 6.7, 6.5, 7.6, 4.4, 6...
## $ clusters     <int> 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, ...
## $ `km$cluster` <int> 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, ...
```

Podemos também plotar os clusters em plano 2d:

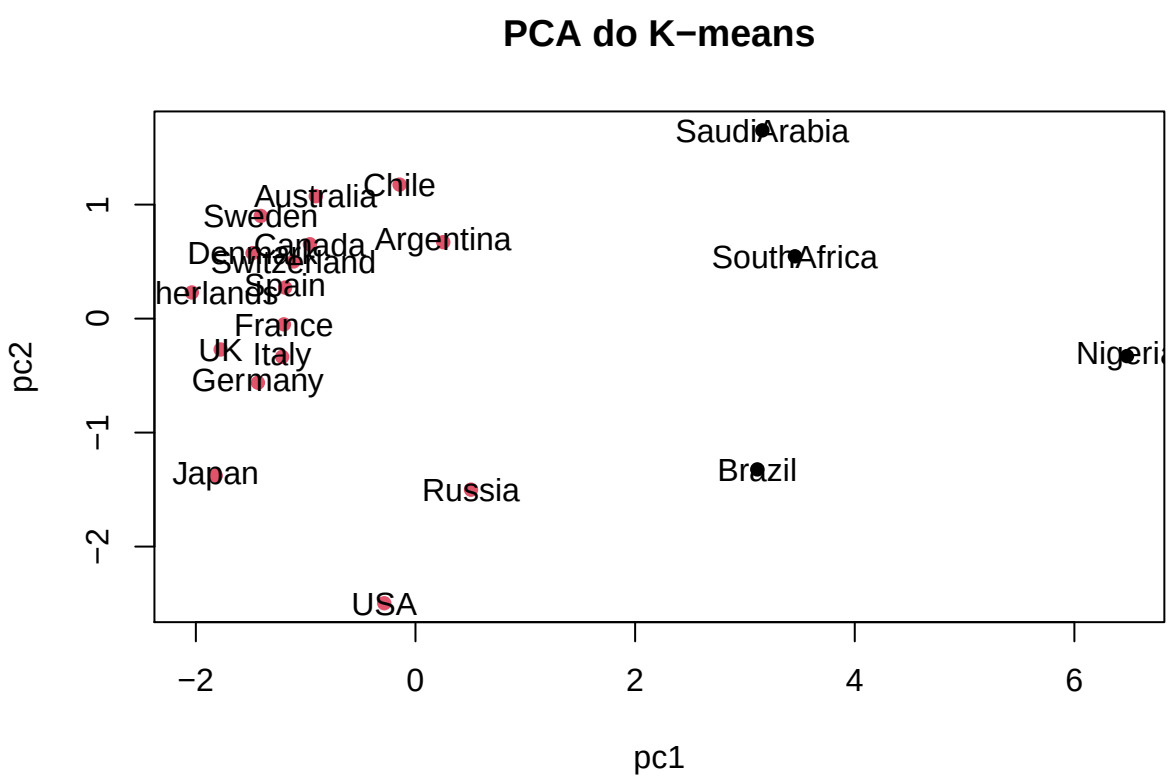
```
factoextra:: fviz_cluster(km, data = dados,
           #  palette = c("#2E9FDF", "#00AFBB", "#E7B800"),
                         geom = "point",
                         ellipse.type = "convex",
                         ggtheme = theme_bw())
```



No plot, é possível perceber que o algoritmo do k-means fez um bom trabalho de classificação com k = 2 grupos.

Da mesma forma, vamos rodar um PCA para visualizarmos:

```
pc<-prcomp(dados, scale.= T)
z<-pc$x
plot(z[,1],z[,2],pch=16,col=km$cluster,xlab="pc1", ylab="pc2",main="PCA do K-means")
text(z,labels=dadosFull[,1])
```



Os clusters tanto do k-means quanto dos hierárquicos apresentaram a mesma classificação para os países: um cluster composto de Brazil, Saudi Arabia, Nigeria e South Africa e um cluster para os demais.

Exercício 3

Sejam $\boldsymbol{X_1} \sim N_3(\boldsymbol{\mu_1}, \Sigma)$ e $\boldsymbol{X_2} \sim N_3(\boldsymbol{\mu_2}, \Sigma)$ vetores aleatorios e considere $\boldsymbol{\mu_1'} = [-3, 1, 4]$ e $\boldsymbol{\mu_2'} = [-1, 0, 3]$. Seja ainda

$$\Sigma = \begin{bmatrix} 1 & -2 & 0 \\ -2 & 5 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

.

a) Admitindo os custos de classificação errada c(2/1) = 100,00 e c(1/2) = 50,00 e probabilidades a priori p1 = 0.7 e p2 = 0.3, encontre e mostre os coeficientes do vetor discriminante e a constante discriminante:

Informações do problema:

```
mu1 <- c(-3, 1, 4)
mu2 <- c(-1, 0, 3)
Sigma <- matrix(c(1,-2,0,-2,5,0,0,0,2), ncol=3)

c2_1 <- 100
c1_2 <- 50
p1 <- 0.7
p2 <- 0.3
```

Os coeficientes do vetor discriminante (a) são dados por $\boldsymbol{a} = (\boldsymbol{\mu_1} - \boldsymbol{\mu_2})^{'}\Sigma^{-1}$:

```
a_transpose <- t(mu1-mu2)%*%solve(Sigma)
a_transpose
```

```
##      [,1] [,2] [,3]
## [1,]   -8   -3  0.5
```

A Constante Discriminante 'm' é encontrada através de $m = \frac{1}{2}\boldsymbol{a}^{'}(\boldsymbol{\mu_1} + \boldsymbol{\mu_2})$:

```
m = 0.5*(a_transpose%*%((mu1+mu2))) #constante discriminante 'm'
m
```

```
##       [,1]
## [1,] 16.25
```

Decide-se que a observação $\boldsymbol{x}_0$ pertence a população 1 se $y \geq m + \ln(\frac{c1|2}{c2|1}\frac{p2}{p1})$:

```
regra <- m + log((c1_2/c2_1)*(p2/p1))
regra
```

```
##          [,1]
## [1,] 14.70955
```

b) Considerando c(2|1) = c(1|2) e p1 = p2, calcule o TOE desta regra. Classifique uma nova observação $\boldsymbol{x}^{'} = [0, 1, 4]$:

Se p1 = p2 e c2_1 = c1_2, a regra para classificar $\boldsymbol{x}$ no grupo 1 torna-se apenas $y \geq m = 16.25$.

Demonstra-se que com essas hipóteses, $PTCI = TOE = \phi(-\Delta/2)$, pois o PTCI foi achado de maneira ótima.

Assim, o TOE é dado por:

```
deltaSquared <- sqrt(a_transpose%*%(mu1-mu2))
pnorm(-deltaSquared/2)
```

```
##            [,1]
## [1,] 0.03309629
```

Seja a nova observação $x_{new} = [0, 1, 4]$. Achamos y através de $y = \boldsymbol{a}^{'}\boldsymbol{x}$:

```
x_new <- c(0,1,4)
y = a_transpose %*% x_new
y
```

```
##      [,1]
## [1,]   -1
```

Assim, como $y = -1 \leq m = 16.25$, classificamos $\boldsymbol{x}_{new}$ como sendo da população 2.

c) Usando a semente set.seed(1), simule uma amostra de tamanho 100 de cada uma das populações, ajuste a função de Ficher aos dados (usando função lda do R), obtenha a matriz de confusão e o TAE. Mostre os coeficientes do vetor discriminante:

Gerando dados:

```
Sigma <- matrix(c(1,-2,0,-2,5,0,0,0,2), ncol=3)
#Sigma # covariancia
mu_1 <- c(-3,1,4) ##vetor de medias da população 1
mu_2 <- c(-1,0,3) ##vetor de medias da população 2
```

```r
###funcao para gerar dados normais multivariados (mvtnorm)

### Simula Variaveis X - Normal Multivariada
n <- 100
set.seed(1)
#Ex: Gerar uma normal 3 variada com media mu_1=c(-3,1,4) e covariancia Sigma
x_1 <- mvtnorm::rmvnorm(n, mean = mu_1, sigma = Sigma, method = "svd")
x_2 <- mvtnorm::rmvnorm(n, mean = mu_2, sigma = Sigma, method = 'svd')

f <- rep(c("g1","g2"), rep(100,2)) #######variavel fator de grupo

##Continuar daqui pra frente


dadosSim <- rbind(x_1, x_2)

colnames(dadosSim) <- c("X1","X2","X3")

dadosSim <- as.data.frame(dadosSim)
dadosSim$f <- f
head(dadosSim)
```

```
##          X1         X2       X3  f
## 1 -3.572825  1.8325361 2.818243 g1
## 2 -2.104963  0.5709577 2.839682 g1
## 3 -3.177410  2.2215588 4.814278 g1
## 4 -4.284933  4.4229143 4.551322 g1
## 5 -1.873254 -3.2588045 5.590893 g1
## 6 -3.020325  0.9974281 5.334786 g1
```

Rodando a LDA para os dados simulados, com prior = 0.3 e 0.7:

```r
dc <- MASS::lda(dadosSim$f ~., dadosSim[,c("X1","X2","X3")], prior = c(0.3,0.7))
dc
```

```
## Call:
## lda(dadosSim$f ~ ., data = dadosSim[, c("X1", "X2", "X3")], prior = c(0.3,
##     0.7))
##
## Prior probabilities of groups:
##  g1  g2
## 0.3 0.7
##
## Group means:
##            X1           X2       X3
## g1 -3.0244737 1.1927380582 3.854905
## g2 -0.9951758 0.0004099852 2.934555
##
## Coefficients of linear discriminants:
##            LD1
## X1  2.09545808
## X2  0.79273068
## X3 -0.09947053
```

Coeficientes do Vetor Discriminante:

```r
dc$scaling  ####coeficientes do vetor discriminante
```

```
##            LD1
## X1  2.09545808
## X2  0.79273068
## X3 -0.09947053
```

Obtendo a TAE (Taxa Aparente de Erro)

```r
pred <- predict(dc)$class ##classificando as observacoes
y <- predict(dc)$x ########gerando os escores discriminantes y-m

tc <- table(dadosSim$f,pred) # Tabela de classificação
TAE <- (tc[1,2]+tc[2,1])/nrow(dadosSim)  ########percentual empirico de erro classificacao
TAE
```

```
## [1] 0.07
```

Avaliando o ajuste: Matriz de Confusão.

```r
caret::confusionMatrix(as.factor(dadosSim$f), pred)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction g1 g2
##         g1 88 12
##         g2  2 98
##
##                Accuracy : 0.93
##                  95% CI : (0.8853, 0.9612)
##     No Information Rate : 0.55
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.86
##
##  Mcnemar's Test P-Value : 0.01616
##
##             Sensitivity : 0.9778
##             Specificity : 0.8909
##          Pos Pred Value : 0.8800
##          Neg Pred Value : 0.9800
##              Prevalence : 0.4500
##          Detection Rate : 0.4400
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.9343
##
##        'Positive' Class : g1
##
```