QMM: Exercise Sheet 5

QMM: Exercise Sheet 5 - Clustering

Library setup

```
library(FactoMineR)
library(ggplot2)
library(dplyr)
library(factoextra)
library(corrplot)
```

Exercise 1

Load the dataset

```
data <- read.csv("Cars.csv")
```

1a)

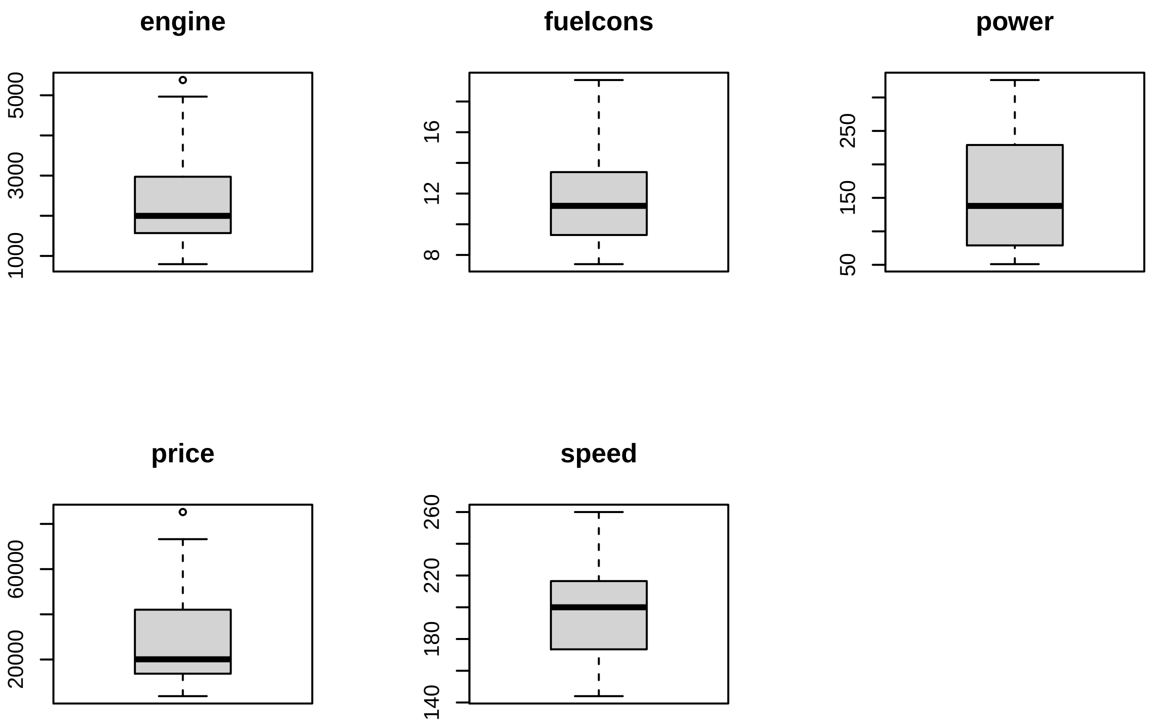
To get a first idea about the variables, their minimum and maximum values, their spread, one can simply use the `summary()` command:

```
summary(data)
```

```
##    carmodel             engine        fuelcons         power      
##  Length:32          Min.   : 796   Min.   : 7.40   Min.   : 51.0  
##  Class :character   1st Qu.:1575   1st Qu.: 9.35   1st Qu.: 79.5  
##  Mode  :character   Median :1998   Median :11.20   Median :138.0  
##                     Mean   :2410   Mean   :11.67   Mean   :161.3  
##                     3rd Qu.:2968   3rd Qu.:13.25   3rd Qu.:227.0  
##                     Max.   :5379   Max.   :19.40   Max.   :326.0  
##      price           speed      
##  Min.   : 3798   Min.   :144.0  
##  1st Qu.:14049   1st Qu.:175.2  
##  Median :20094   Median :200.0  
##  Mean   :28697   Mean   :198.8  
##  3rd Qu.:41354   3rd Qu.:215.2  
##  Max.   :85250   Max.   :260.0  
```

A visual approach, however, can bring some insights into many aspects of the dataset variables. One can first start to draw boxplots of the variables to describe their spread. You can use the following command, which allows to change the geometry of the plotting area and to display the individual boxplots:

```
par(mfrow=c(2,3))
for (i in 2:6){
        boxplot(data[,i], main=names(data[i]), type="l")

}
par(mfrow=c(1,1))
```

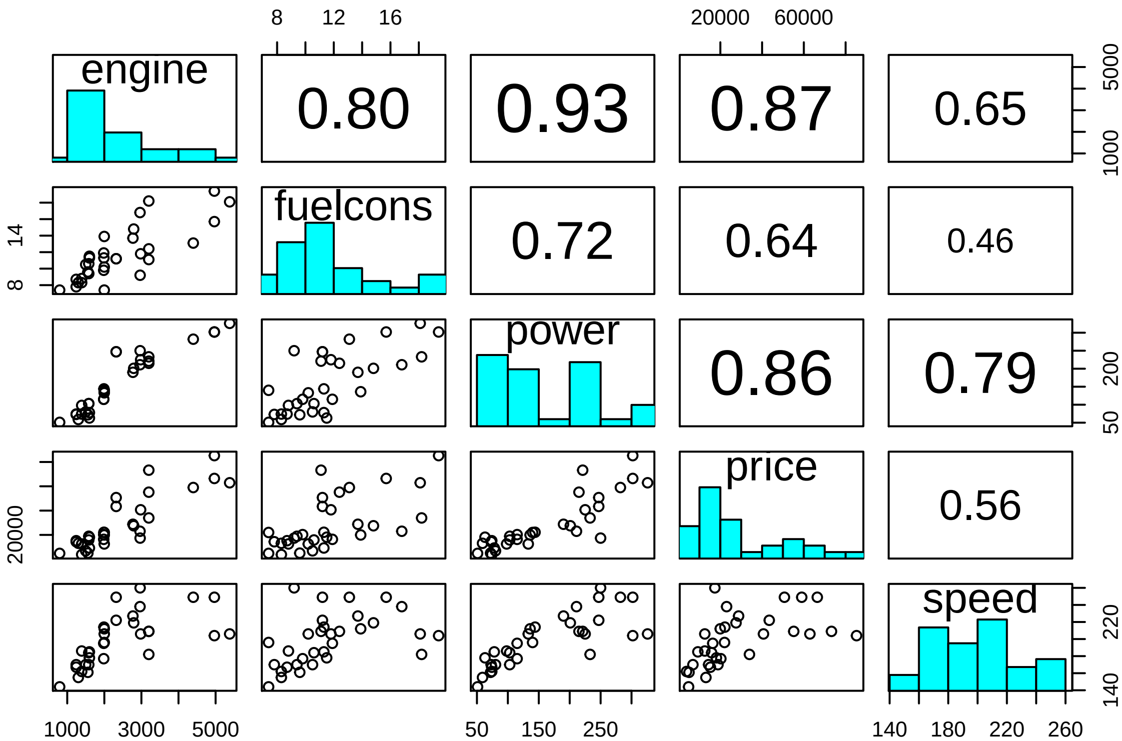


We remark, among other things, the large spread in the price of cars, in engine power and fuel consumption, signalling a very heterogenous dataset.

One can obtain more by drawing the pairwise correlations, the histogram of each of the variables and the displaying the Pearson's correlation coefficient:

```
panel.hist <- function(x, ...){
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(usr[1:2], 0, 1.5) )
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks; nB <- length(breaks)
    y <- h$counts; y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...){
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste0(prefix, txt)
    if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}
```

```r
pairs(data[,2:6], upper.panel=panel.cor, diag.panel=panel.hist)
```



In that way, we can for example describe, among other thing, the bimodal distribution in the power variable, the long-tailed behaviour of the price variable and the good correlation among the variables, especially high for the relations engine and power and engine and price.

1b)

To create a function that computes the Euclidean distance between two vectors of same length but that length could be arbitrary, one can use for example the following code and use it to compute the Euclidean distance between the first two observations of the dataset:

```r
# Here I create a function for which we can have any number of variables. Note that we need to feed it
eucdist2 <- function(data1, data2){
  nvar <- length(data1)
  sumsq <- 0
  for(i in 1:nvar){
    sumsq <- sumsq + (data1[i]-data2[i])^2
  }
  return(sqrt(sumsq))
}

as.numeric(eucdist2(data[1, -1], data[2, -1]))
```

```
## [1] 5406.275
```

We see that the distance between the Ford Mondeo LX and the Ford Galaxy LX is 5406.275.

1c)

One can use the `dist()` function (only on numerical variables) to get the distance table:

```
distmat <- dist(data[,-1], method="euclidean", diag=TRUE,upper=TRUE)
```

Displaying the (1,2) entry of this matrix, we should find the same distance as in b):

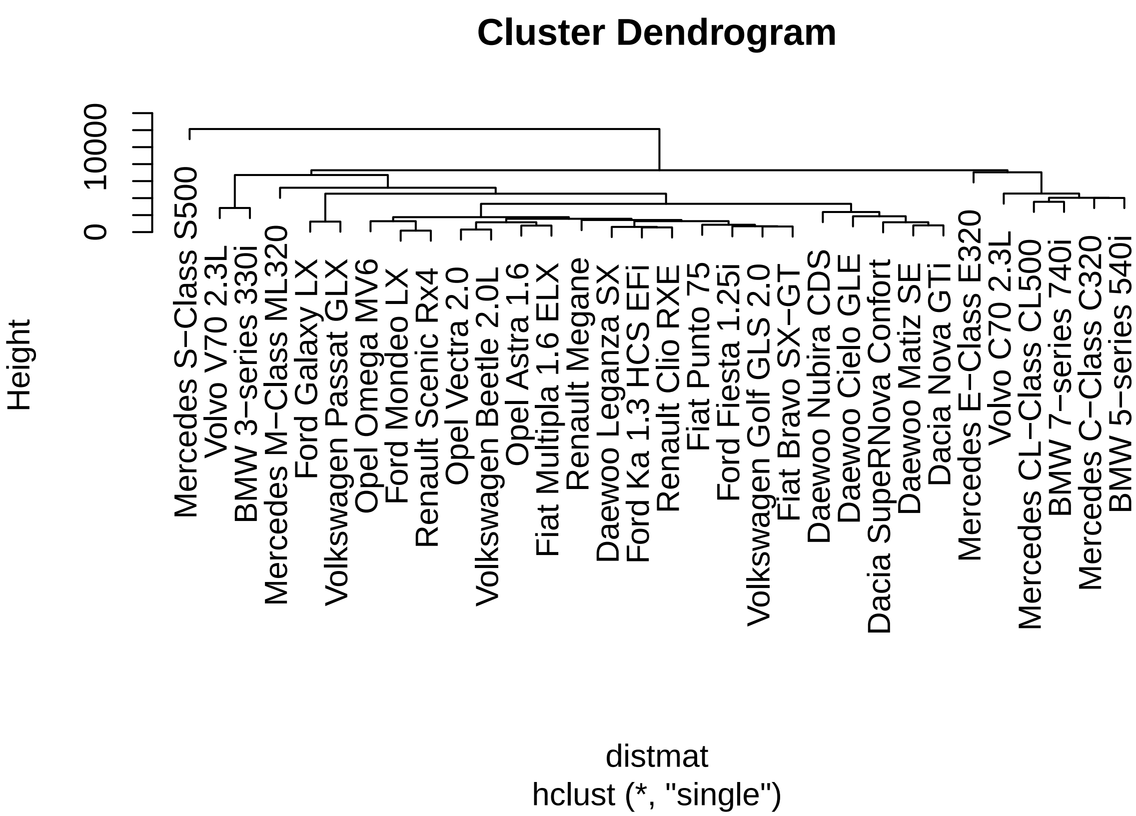
```
as.matrix(distmat)[1,2]
```

```
## [1] 5406.275
```

1d)

One can obtain a single-linkage clustering model using the following command, plotting the resulting dendrogram:

```
slclust <- hclust(distmat, method="single")
plot(slclust, labels=data[,1])
```

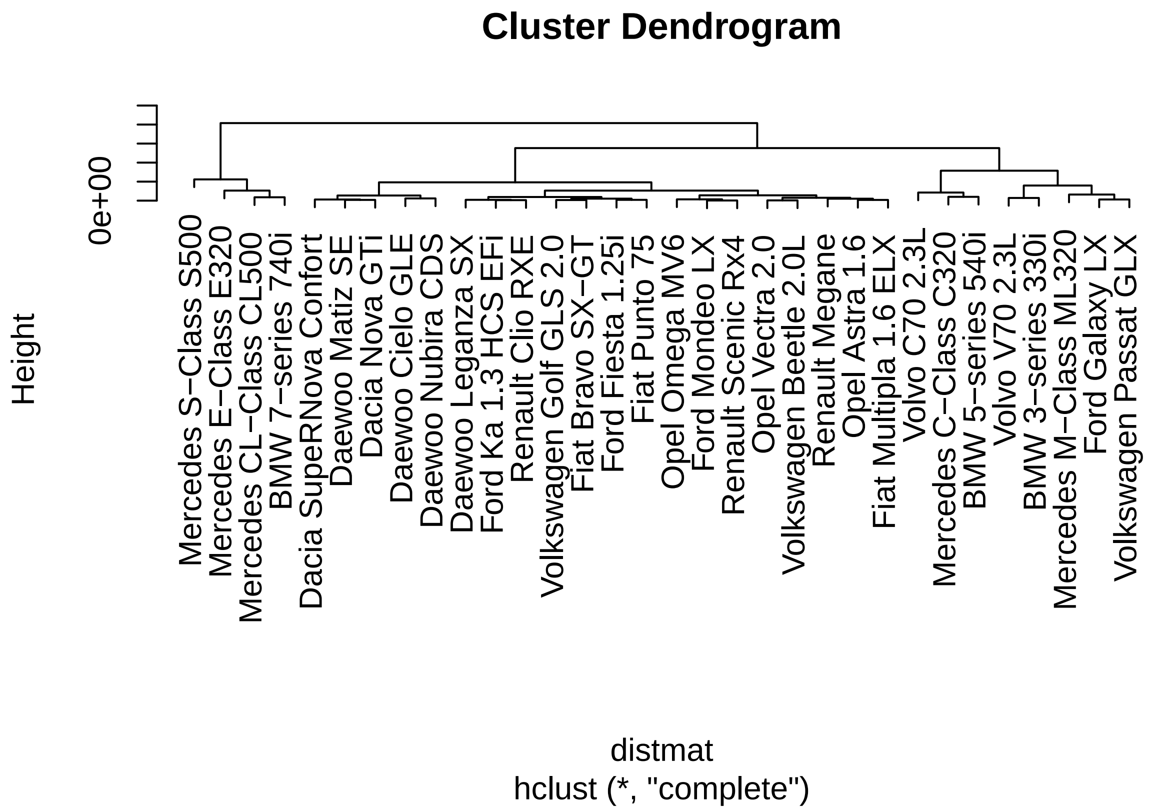


One sees, perhaps "intuitively", that cars in the same branding category are regrouped, e.g. the rightmost cluster of Mercedes, Volva snd BMW cars, that are targetting the same segment. However, one observation stands out in the form of the Mercedes S-Class S500, which is basically alone up to very high height of the dendrogram.

1e)

One uses the exact same function simply changing the method to get a complete-linkage clustering model:

```r
coclust <- hclust(distmat, method="complete")
plot(coclust, labels=data[,1])
```

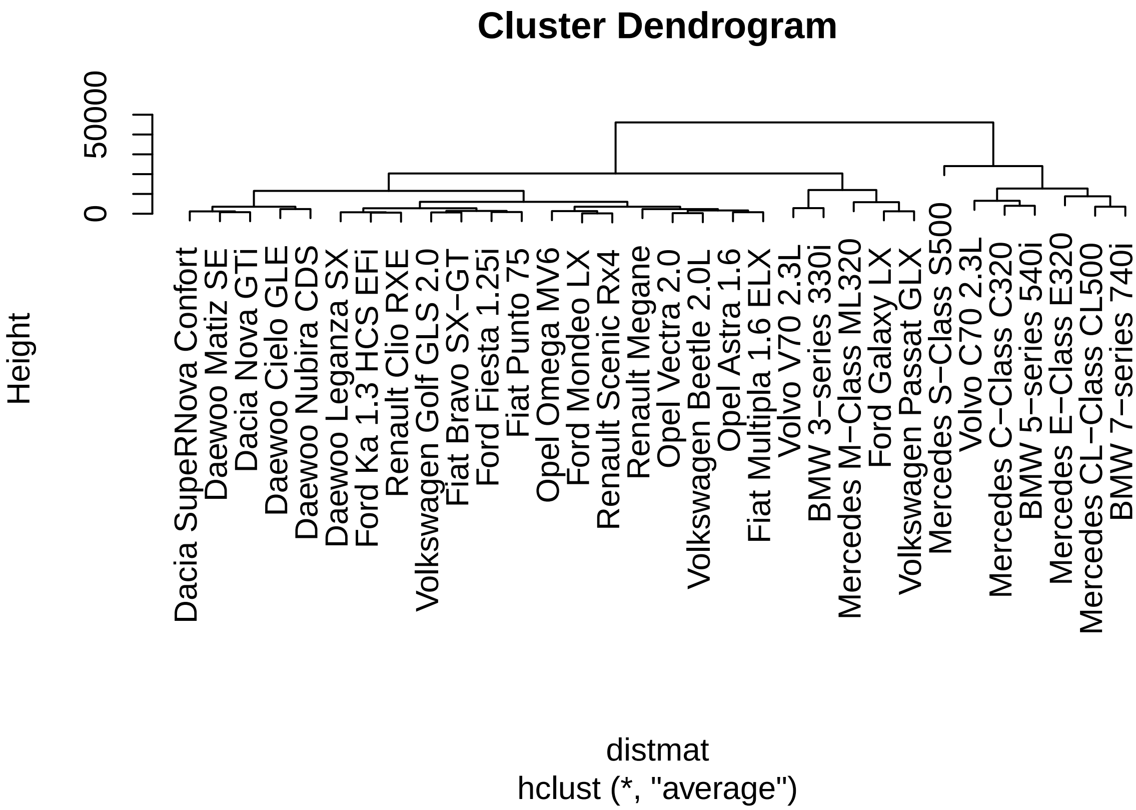


The Mercedes S-Class S500 is not isolated anymore and it seems that the clustering is more "intuitive". However, the fact that some BMW and Mercedes are split across the dendrogram could seem a bit off.

1f)

One uses the exact same function simply changing the method to get an average-linkage clustering model:

```r
avclust <- hclust(distmat, method="average")
plot(avclust, labels=data[,1])
```

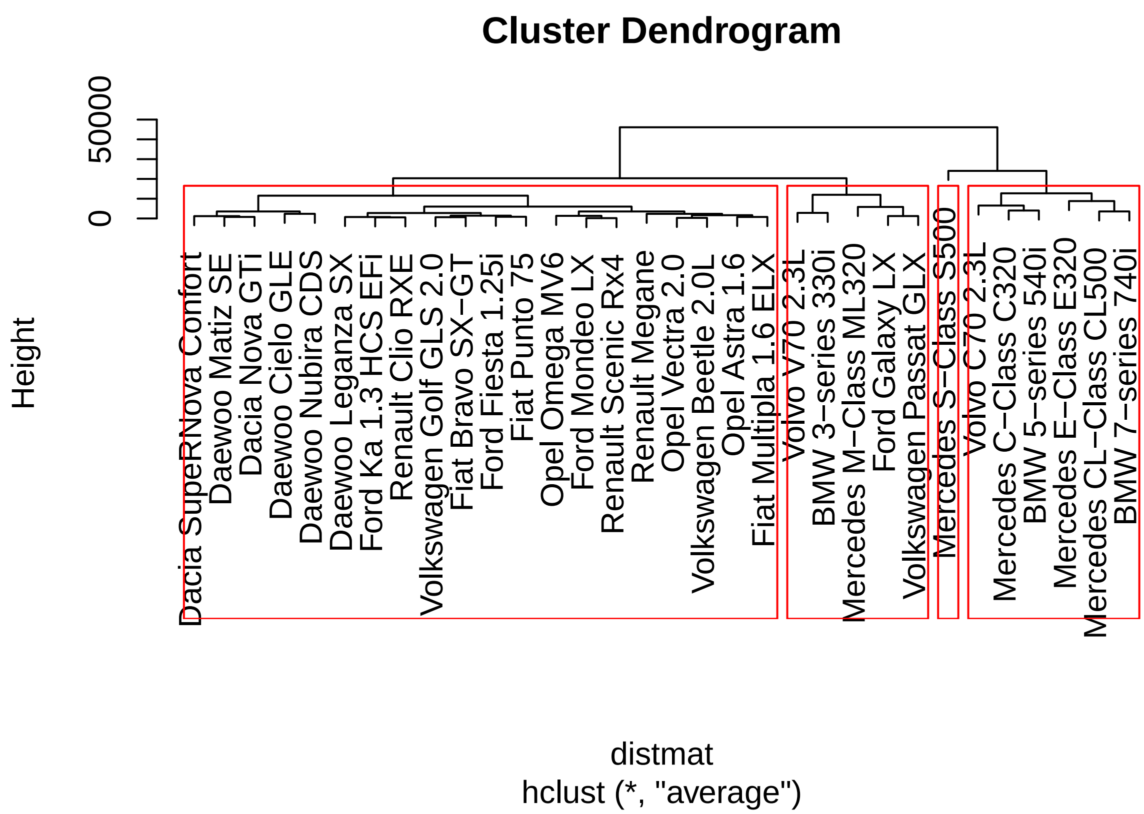


The picture is a bit different again and it seems that all "fancier" cars are clustered together, with some exceptions only explained by cars characteristics. But recall the discussion in the course: one prefers space-dilating methods (property completely fulfilled by complete-linkage) to space-contracting ones (property completely fulfilled by single-linkage). Average-linkage methods fall in between and could be preferable.

1g)

The following graph, using the `cutree()` function, shows you how to emphasise the clusters, when we select only k=4 of them:

```
ct1 <- cutree(avclust, k=4)
plot(avclust, labels=data[,1])
rect.hclust(avclust, k=4, border="red")
```

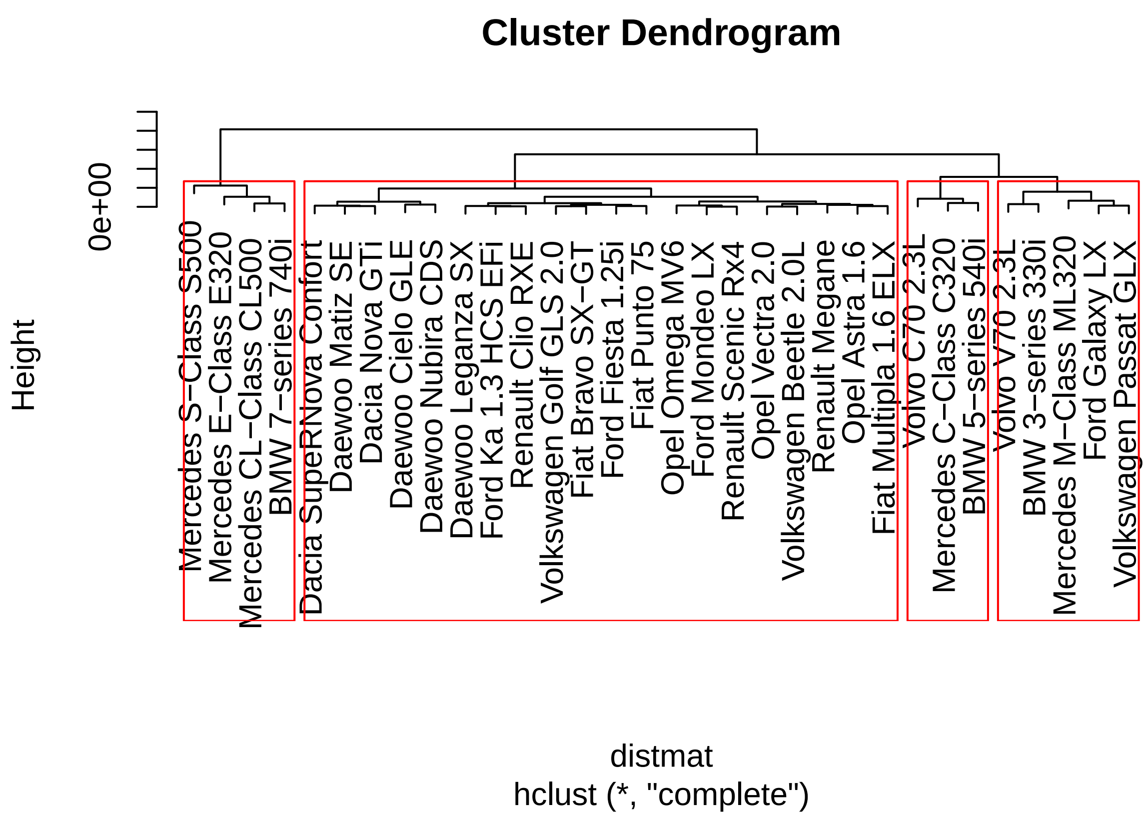



```
split(data, ct1)
```

```
## $`1`
##                   carmodel engine fuelcons power price speed
## 1             Ford Mondeo LX   1989     11.3   144 22170   214
## 3          Ford Fiesta 1.25i   1242      8.7    74 15232   167
## 4        Ford Ka 1.3 HCS EFi   1299      8.3    59 13085   155
## 5             Opel Astra 1.6   1598     11.5    63 18105   178
## 6            Opel Vectra 2.0   1998     13.9   136 19946   212
## 7             Opel Omega MV6   2962     16.8   211 22999   238
## 8  Volkswagen Golf GLS 2.0   1984     11.9   115 16350   195
## 10  Volkswagen Beetle 2.0L   1984      9.8   115 20243   177
## 11         Daewoo Leganza SX   1998     10.2   133 12545   206
## 12          Daewoo Cielo GLE   1498     10.5    80  6860   170
## 13         Daewoo Nubira CDS   1598     11.3    78  9222   185
## 14           Daewoo Matiz SE    796      7.4    51  4792   144
## 18          Renault Clio RXE   1390      8.8    98 12542   186
## 19            Renault Megane   2965      9.2   250 17370   260
## 20        Renault Scenic Rx4   1998      7.4   140 22000   196
## 21             Fiat Punto 75   1242      7.8    73 14370   170
## 22          Fiat Bravo SX-GT   1581     10.6   103 15830   184
## 23    Fiat Multipla 1.6 ELX   1581      9.4   103 18870   170
## 24            Dacia Nova GTi   1557      9.6    72  5000   161
## 25 Dacia SupeRNova Confort   1390      8.3    74  3798   162
##
## $`2`
##                 carmodel engine fuelcons power price speed
## 2           Ford Galaxy LX   2792     14.8   201 27516   219
```

```
## 9   Volkswagen Passat GLX   2771      13.7    190 28750    227
## 15 Mercedes M-Class ML320   3199      18.2    233 33950    182
## 16          Volvo V70 2.3L   2319      11.2    247 43422    222
## 17       BMW 3-series 330i   2979      11.8    225 40664    206
##
## $`3`
##                      carmodel engine fuelcons power price speed
## 26   Mercedes C-Class C320    3199      12.4    215 55150    209
## 27 Mercedes CL-Class CL500    4966      15.7    302 66440    249
## 28   Mercedes E-Class E320    3199      11.1    221 73250    209
## 30          Volvo C70 2.3L    2319      11.2    247 50700    249
## 31       BMW 5-series 540i    4398      13.1    282 58985    249
## 32       BMW 7-series 740i    5379      18.1    326 62900    206
##
## $`4`
##                    carmodel engine fuelcons power price speed
## 29 Mercedes S-Class S500    4966      19.4    302 85250    204
```

```
ct2 <- cutree(coclust, k=4)
plot(coclust, labels=data[,1])
rect.hclust(coclust, k=4, border="red")
```




```
split(data, ct2)
```

```
## $`1`
##                      carmodel engine fuelcons power price speed
## 1             Ford Mondeo LX    1989      11.3    144 22170    214
## 3          Ford Fiesta 1.25i    1242       8.7     74 15232    167
## 4        Ford Ka 1.3 HCS EFi    1299       8.3     59 13085    155
```

```
## 5           Opel Astra 1.6   1598      11.5    63 18105   178
## 6          Opel Vectra 2.0   1998      13.9   136 19946   212
## 7           Opel Omega MV6   2962      16.8   211 22999   238
## 8  Volkswagen Golf GLS 2.0   1984      11.9   115 16350   195
## 10  Volkswagen Beetle 2.0L   1984       9.8   115 20243   177
## 11       Daewoo Leganza SX   1998      10.2   133 12545   206
## 12        Daewoo Cielo GLE   1498      10.5    80  6860   170
## 13       Daewoo Nubira CDS   1598      11.3    78  9222   185
## 14         Daewoo Matiz SE    796       7.4    51  4792   144
## 18        Renault Clio RXE   1390       8.8    98 12542   186
## 19          Renault Megane   2965       9.2   250 17370   260
## 20      Renault Scenic Rx4   1998       7.4   140 22000   196
## 21           Fiat Punto 75   1242       7.8    73 14370   170
## 22        Fiat Bravo SX-GT   1581      10.6   103 15830   184
## 23   Fiat Multipla 1.6 ELX   1581       9.4   103 18870   170
## 24          Dacia Nova GTi   1557       9.6    72  5000   161
## 25 Dacia SupeRNova Confort   1390       8.3    74  3798   162
##
## $`2`
##                  carmodel engine fuelcons power price speed
## 2          Ford Galaxy LX   2792     14.8   201 27516   219
## 9   Volkswagen Passat GLX   2771     13.7   190 28750   227
## 15 Mercedes M-Class ML320   3199     18.2   233 33950   182
## 16          Volvo V70 2.3L   2319     11.2   247 43422   222
## 17       BMW 3-series 330i   2979     11.8   225 40664   206
##
## $`3`
##                 carmodel engine fuelcons power price speed
## 26 Mercedes C-Class C320   3199     12.4   215 55150   209
## 30        Volvo C70 2.3L   2319     11.2   247 50700   249
## 31     BMW 5-series 540i   4398     13.1   282 58985   249
##
## $`4`
##                   carmodel engine fuelcons power price speed
## 27 Mercedes CL-Class CL500   4966     15.7   302 66440   249
## 28   Mercedes E-Class E320   3199     11.1   221 73250   209
## 29   Mercedes S-Class S500   4966     19.4   302 85250   204
## 32       BMW 7-series 740i   5379     18.1   326 62900   206
```

The choice is made such that it cuts at a height leading to 4 clusters only. With the average-linkage method, the Mercedes S-Class is again alone in a single cluster. The three others are as follow: one is maded up of the most performing, expensive cars; another one is made up of entry-level cars of fancy manufacturers and the remaining ones, i.e. cars that are sometimes less performing but especially priced differently are clustered together. The picture is close under the complete-linkage model, even though we have three different clusters of fancy cars and one with all the other ones.

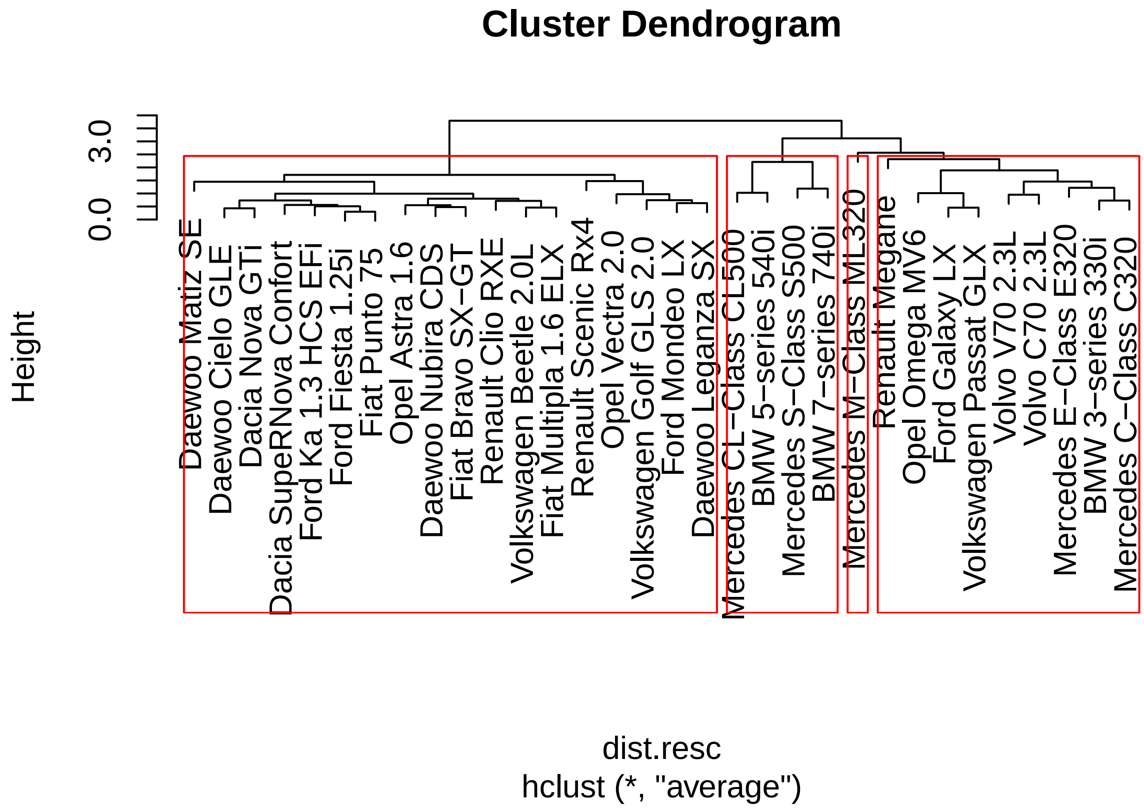
1h)

We start by rescaling the data and recomputing the distance matrix:

```
data.resc <- as.data.frame(cbind(data$carmodel, scale(data[,c(2:6)])))

dist.resc <- dist(data.resc[,-1], method="euclidean", diag=TRUE,upper=TRUE)
```

We then use an average-linkage model on the scaled data and use 4 clusters to show thee difference:

```
avclust.resc <- hclust(dist.resc, method="average")
plot(avclust.resc, labels=data.resc[,1])
ct3 <- cutree(avclust.resc, k=4)
rect.hclust(avclust.resc, k=4, border="red")
```




```
split(data.resc, ct3)
```

```
## $`1`
##                       V1             engine            fuelcons
## 1           Ford Mondeo LX -0.358382900507786  -0.114091994989392
## 3         Ford Fiesta 1.25i -0.993524126318737  -0.918537247795953
## 4       Ford Ka 1.3 HCS EFi -0.945059534790994   -1.04229805592004
## 5           Opel Astra 1.6  -0.69083299326827 -0.0522115909273487
## 6          Opel Vectra 2.0 -0.350730596582353   0.690353257817169
## 8  Volkswagen Golf GLS 2.0  -0.36263418046636  0.0715492171967377
## 10  Volkswagen Beetle 2.0L  -0.36263418046636  -0.578195025454715
## 11        Daewoo Leganza SX -0.350730596582353  -0.454434217330629
## 12         Daewoo Cielo GLE  -0.77585859243975  -0.361613611237564
## 13        Daewoo Nubira CDS  -0.69083299326827  -0.114091994989392
## 14          Daewoo Matiz SE  -1.37273829862354   -1.32075987419923
## 18         Renault Clio RXE -0.867686239544948  -0.887597045764931
## 20       Renault Scenic Rx4 -0.350730596582353   -1.32075987419923
## 21            Fiat Punto 75 -0.993524126318737   -1.19699906607515
## 22         Fiat Bravo SX-GT -0.705287345127422  -0.330673409206543
## 23    Fiat Multipla 1.6 ELX -0.705287345127422  -0.701955833578801
## 24           Dacia Nova GTi -0.725693488928577  -0.640075429516759
## 25 Dacia SupeRNova Confort -0.867686239544948   -1.04229805592004
```

```
##                power             price               speed
## 1  -0.206583167583111 -0.297198389660949   0.503076728681704
## 3   -1.04036027638702 -0.613108100571128    -1.0505729631813
## 4   -1.21902679970214 -0.710867995256073    -1.4472494802527
## 5   -1.17138239348477 -0.482291064041156  -0.686952822532511
## 6  -0.301871980017843  -0.39846434484461   0.436963975836469
## 8  -0.552005112659015 -0.562201923459917  -0.124994423348021
## 10 -0.552005112659015 -0.384940968635281  -0.720009198955128
## 11 -0.337605284680867 -0.735455952000308   0.238625717300767
## 12 -0.968893667060968 -0.994312496613217  -0.951403833913448
## 13 -0.992715870169651 -0.886762952483808  -0.455558187574191
## 14  -1.31431561213687  -1.08847526429299   -1.81086962090149
## 18  -0.75449383908282 -0.735592551759998  -0.422501811151574
## 20 -0.254227573800477  -0.30493904271006 -0.0919380469254037
## 21  -1.05227137794136 -0.652357764855442  -0.951403833913448
## 22 -0.694938331311113  -0.58587921513955  -0.488614563996809
## 23 -0.694938331311113 -0.447458125320158  -0.951403833913448
## 24   -1.0641824794957  -1.07900434762113     -1.248911221717
## 25  -1.04036027638702  -1.13373531800367   -1.21585484529438
##
## $`2`
##                       V1             engine            fuelcons
## 2         Ford Galaxy LX   0.324372660839194  0.968815076096363
## 7         Opel Omega MV6   0.468916179430709   1.58761911671679
## 9  Volkswagen Passat GLX   0.306517285013183  0.628472853755126
## 16        Volvo V70 2.3L -0.0777984232419037 -0.145032197020414
## 17     BMW 3-series 330i   0.483370531289861 0.0406090151657162
## 19        Renault Megane   0.471466947405853 -0.763836237640845
## 26 Mercedes C-Class C320   0.670426849467115  0.226250227351846
## 28 Mercedes E-Class E320   0.670426849467115 -0.175972399051435
## 30        Volvo C70 2.3L -0.0777984232419037 -0.145032197020414
##                power               price             speed
## 2  0.472349621014356 -0.0537776178930315 0.668358610794789
## 7  0.591460636557771  -0.259451322733227  1.29642976282451
## 9  0.341327503916599 0.00241041659286647 0.932809622175725
## 16  1.02026029251407   0.670474307984354  0.76752774006264
## 17 0.758216058318552   0.544893595575839 0.238625717300767
## 19  1.05599359717709  -0.515758005165253  2.02367004412209
## 26 0.639105042775137    1.20448830186653 0.337794846568618
## 28 0.710571652101186    2.02864018533068 0.337794846568618
## 30  1.02026029251407    1.00186532499275   1.6600499034733
##
## $`3`
##                        V1            engine        fuelcons             power
## 15 Mercedes M-Class ML320 0.670426849467115 2.0207819451511 0.853504870753284
##                price              speed
## 15 0.239183333389195 -0.554727316842043
##
## $`4`
##                         V1            engine          fuelcons             power
## 27 Mercedes CL-Class CL500 2.17282918682716  1.24727689437556 1.67537087800285
## 29   Mercedes S-Class S500 2.17282918682716  2.39206436952335 1.67537087800285
## 31       BMW 5-series 540i 1.68988378353315 0.442831641568996 1.43714884691602
## 32       BMW 7-series 740i 2.52398491140537  1.98984174312008 1.96123731530705
```

```
##                 price             speed
## 27 1.71855873083395   1.6600499034733
## 29 2.57503922409144 0.172512964455533
## 31 1.37910832800383   1.6600499034733
## 32 1.55737101439952 0.238625717300767
```

his time, it seems that the clustering is different. We have very fancy cars in a cluster (with the S-Class S500 for example), some good performing cars that have upper but not higher prices in the rightmost cluster and another cluster with the other cars, let alone the Mercedes M-Class which is alone. It could be that, when the data were not rescaled, the distance in the price variable was the most important feature in computing the various distances and impacted it. Therefore, by rescaling, we avoid such problems and all the variables are given the same importance in computing the distances on which the clustering is based.

Note that the rescaling of data happens in the following way: for the ith observation of the jth variable $x_{i,j}$, we compute

$$r_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}$$

where μ_j is the mean of the jth variable (approximated by the empirical mean in practice) and σ_j is the standard deviation of the jth variable (approximated by the empirical standard deviation in practice).

Exercise 2

Load the dataset

```
data2 <- read.csv("Country.csv")
```

2a)

We start by rescaling the data and create a new dataset for which we name the rows according to the country they are observed in:

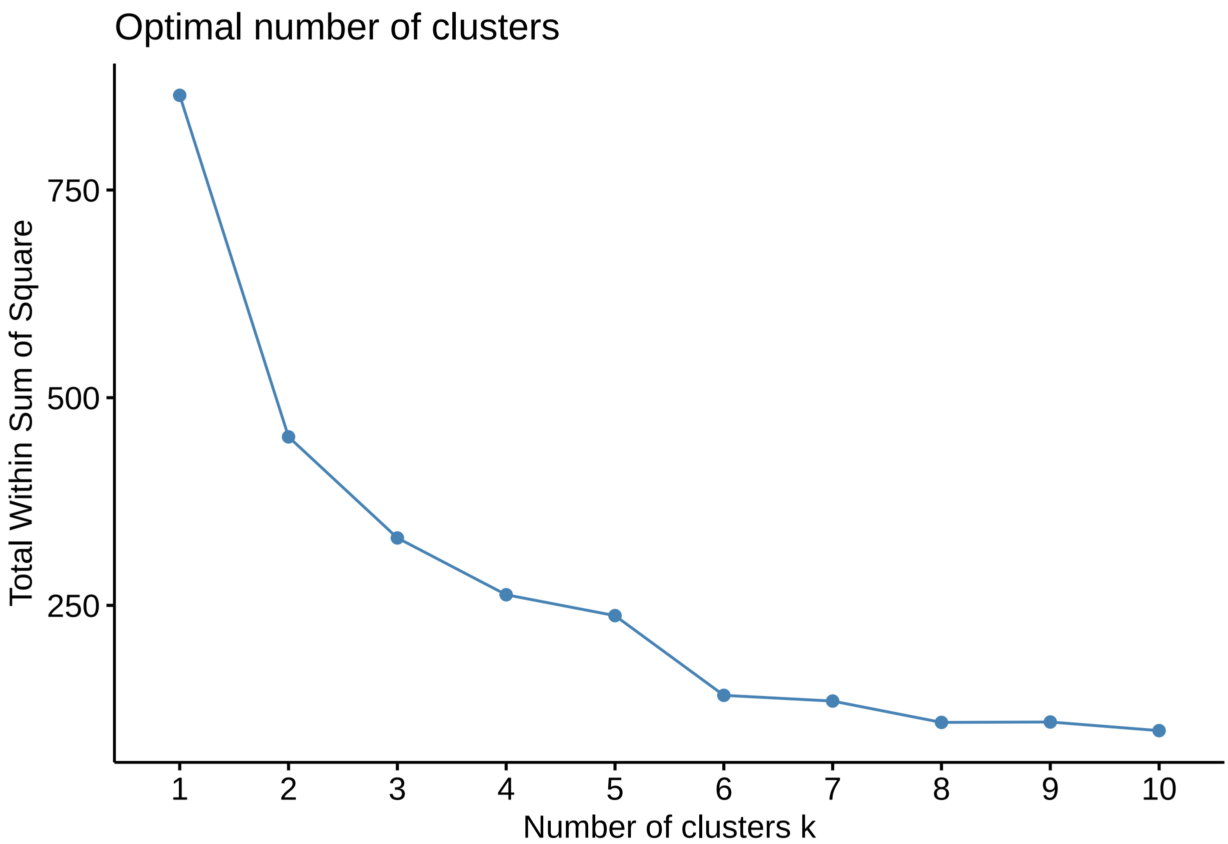
```
data2.sc <- scale(data2[,c(2:9)])
row.names(data2.sc) <- data2[,1]
```

This method will allow us to use the information of the country name on the graphs thereafter. If we were not to name the rows, then we would simply have the numbered observations appearing in the k-means graphs.

2b)

Recall that the aim of clustering is to come up with groups that are different from each other but, within a group, we want to minimise the variation. So we prefer the variation in the data to come from the difference Between Groups and not Within Groups. Each time we add a new cluster to the data, we switch information from the Within Groups to Between Groups. But there is a turning point at which adding a new group is not worth the reduce in the Within Groups variation. We use a graphical method to find such point, using the \texttt{fviz_nbclust()} command from the `factoextra()` package:

```
fviz_nbclust(data2.sc, kmeans, method="wss")
```



We see that a large jump occuprs in the reduction of the Within Sum of Square when going from 5 to 6 clusters but thereafter, it reduces only slightly. So we stop at 6 clusters.

2c)

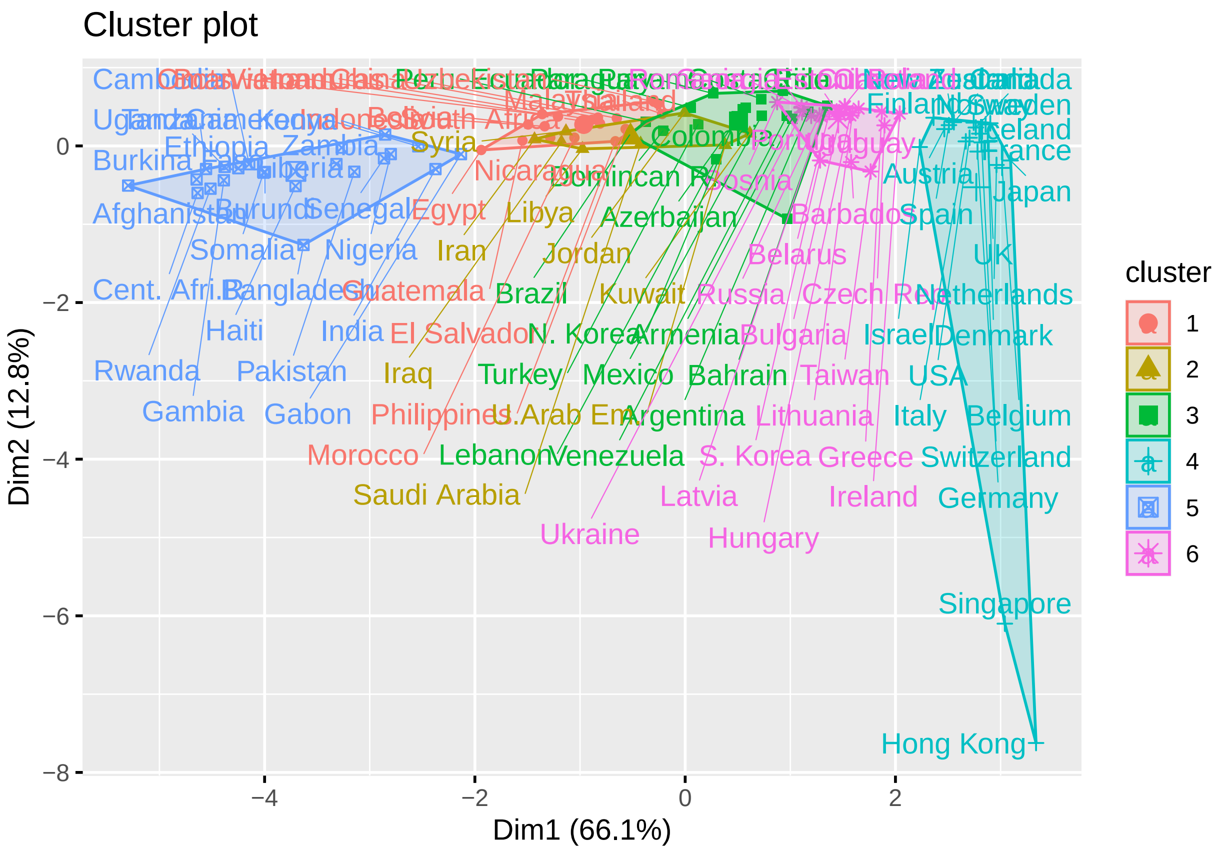
To run a k-means clustering model, use the following code, from which you will get an output describing the model and the plot along the first two principal components (if there are more than 2 variables in the dataset).

```
km.res <- kmeans(data2.sc, 6)
print(km.res)
```

```
## K-means clustering with 6 clusters of sizes 17, 8, 18, 22, 22, 22
##
## Cluster means:
##      density       flexp         gdp     infmort   literacy      mlexp
## 1 -0.1720163 -0.18167892 -0.65007717  0.20588864 -0.2204794 -0.1814065
## 2 -0.2384208  0.03254294 -0.12075640  0.04923031 -0.5596754  0.2245902
## 3 -0.1264904  0.35835800 -0.49706908 -0.23931223  0.4394809  0.3563982
## 4  0.5429666  0.96125918  1.77638760 -0.93833180  0.8114559  0.9843924
## 5 -0.1249148 -1.73459769 -0.80726368  1.66771248 -1.5667012 -1.6920721
## 6 -0.0949394  0.60869098 -0.01618725 -0.71057747  0.7695588  0.4745897
##      popincr      urban
## 1  0.4509805 -0.7964337
## 2  1.8428976  0.6673437
## 3  0.1756998  0.4657893
## 4 -0.9147120  0.9957505
```

```
## 5  0.8267361 -1.2763172
## 6 -1.0744081  0.2722219
##
## Clustering vector:
##  Afghanistan    Argentina      Armenia    Australia      Austria   Azerbaijan
##            5            3            3            4            4            3
##      Bahrain   Bangladesh     Barbados      Belarus      Belgium      Bolivia
##            3            5            6            6            4            1
##       Bosnia     Botswana       Brazil     Bulgaria Burkina Faso      Burundi
##            6            1            3            6            5            5
##     Cambodia     Cameroon       Canada Cent. Afri.R        Chile        China
##            5            5            4            5            3            1
##     Colombia   Costa Rica      Croatia         Cuba   Czech Rep.      Denmark
##            3            3            6            6            6            4
##  Domincan R.      Ecuador        Egypt  El Salvador      Estonia     Ethiopia
##            3            3            1            1            6            5
##      Finland       France        Gabon       Gambia      Georgia      Germany
##            4            4            5            5            6            4
##       Greece    Guatemala        Haiti     Honduras    Hong Kong      Hungary
##            6            1            5            1            4            6
##      Iceland        India    Indonesia         Iran         Iraq      Ireland
##            4            5            1            2            2            6
##       Israel        Italy        Japan       Jordan        Kenya       Kuwait
##            4            4            4            2            5            2
##       Latvia      Lebanon      Liberia        Libya    Lithuania     Malaysia
##            6            3            5            2            6            1
##       Mexico      Morocco     N. Korea  Netherlands  New Zealand    Nicaragua
##            3            1            3            4            4            1
##      Nigeria       Norway         Oman     Pakistan       Panama     Paraguay
##            5            4            1            5            3            3
##         Peru  Philippines       Poland     Portugal      Romania       Russia
##            3            1            6            6            6            6
##       Rwanda     S. Korea Saudi Arabia      Senegal    Singapore      Somalia
##            5            6            2            5            4            5
## South Africa        Spain       Sweden  Switzerland        Syria       Taiwan
##            1            4            4            4            2            6
##     Tanzania     Thailand       Turkey   U.Arab Em.           UK          USA
##            5            1            3            2            4            4
##       Uganda      Ukraine      Uruguay   Uzbekistan    Venezuela      Vietnam
##            5            6            6            1            3            1
##       Zambia
##            5
##
## Within cluster sum of squares by cluster:
## [1]  20.63726  13.39498  18.79707 109.21739  41.83624  16.94713
##  (between_SS / total_SS =  74.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
fviz_cluster(km.res, data=data2.sc, repel=TRUE)
```



We can see the 6 clusters and the individual observations along the two first principal components. However, running this code multiple times leads to different results. This is because, if you do not specify the `nstart` parameter in the `kmeans()` function, then there is only one choice of random set of rows chosen in the dataset as initial centers. We need to specify this parameter to obtain more robust results and interpret the model.

2d)

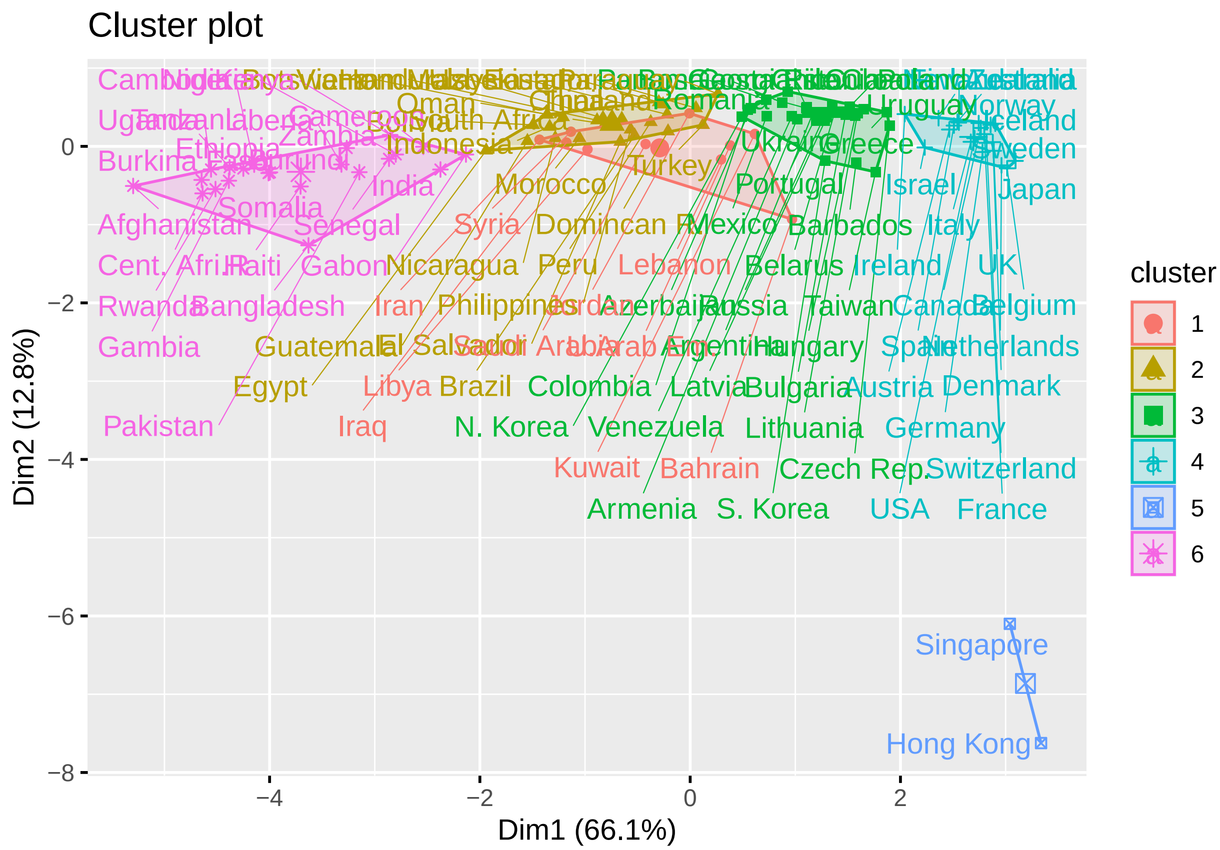
We repeat the experiment but we now set `nstart` to 25, to choose multiple random sets at the beginning of the algorithm to obtain the partition.

```r
set.seed(1)
km.res2 <- kmeans(data2.sc, 6, nstart = 25)
print(km.res2)
```

```
## K-means clustering with 6 clusters of sizes 10, 23, 31, 21, 2, 22
##
## Cluster means:
##       density       flexp        gdp     infmort    literacy      mlexp
## 1 -0.07764433  0.07037952 -0.1338894 -0.01346779 -0.44665308  0.2677274
## 2 -0.18482704 -0.11756961 -0.6439346  0.18737163 -0.09093573 -0.1130049
## 3 -0.12722933  0.58941207 -0.1982300 -0.62272255  0.71974571  0.4715853
## 4 -0.11445596  0.95818803  1.7757239 -0.93522823  0.87387395  0.9794905
## 5  7.06163778  0.88386616  1.3820594 -0.96017616  0.18100483  0.9794905
## 6 -0.12491481 -1.73459769 -0.8072637  1.66771248 -1.56670123 -1.6920721
##     popincr      urban
## 1  1.5605777  0.7552008
## 2  0.3949284 -0.5354664
```

```
## 3 -0.6966323  0.3524254
## 4 -0.9230408  0.8851406
## 5 -0.9489326  1.6647796
## 6  0.8267361 -1.2763172
##
## Clustering vector:
##  Afghanistan    Argentina      Armenia    Australia      Austria   Azerbaijan
##            6            3            3            4            4            3
##      Bahrain   Bangladesh     Barbados      Belarus      Belgium      Bolivia
##            1            6            3            3            4            2
##       Bosnia     Botswana       Brazil     Bulgaria Burkina Faso      Burundi
##            3            2            2            3            6            6
##     Cambodia     Cameroon       Canada Cent. Afri.R        Chile        China
##            6            6            4            6            3            2
##     Colombia   Costa Rica      Croatia         Cuba   Czech Rep.      Denmark
##            3            3            3            3            3            4
##  Domincan R.      Ecuador        Egypt  El Salvador      Estonia     Ethiopia
##            2            2            2            2            3            6
##      Finland       France        Gabon       Gambia      Georgia      Germany
##            4            4            6            6            3            4
##       Greece    Guatemala        Haiti     Honduras    Hong Kong      Hungary
##            3            2            6            2            5            3
##      Iceland        India    Indonesia         Iran         Iraq      Ireland
##            4            6            2            1            1            4
##       Israel        Italy        Japan       Jordan        Kenya       Kuwait
##            4            4            4            1            6            1
##       Latvia      Lebanon      Liberia        Libya    Lithuania     Malaysia
##            3            1            6            1            3            2
##       Mexico      Morocco     N. Korea  Netherlands  New Zealand    Nicaragua
##            3            2            3            4            4            2
##      Nigeria       Norway         Oman     Pakistan       Panama     Paraguay
##            6            4            2            6            3            2
##         Peru  Philippines       Poland     Portugal      Romania       Russia
##            2            2            3            3            3            3
##       Rwanda     S. Korea Saudi Arabia      Senegal    Singapore      Somalia
##            6            3            1            6            5            6
## South Africa        Spain       Sweden  Switzerland        Syria       Taiwan
##            2            4            4            4            1            3
##     Tanzania     Thailand       Turkey   U.Arab Em.           UK          USA
##            6            2            2            1            4            4
##       Uganda      Ukraine      Uruguay   Uzbekistan    Venezuela      Vietnam
##            6            3            3            2            3            2
##       Zambia
##            6
##
## Within cluster sum of squares by cluster:
## [1] 19.756374 31.550480 33.316333 13.177178  1.947703 41.836241
##  (between_SS / total_SS =  83.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
fviz_cluster(km.res2, data=data2.sc, repel=TRUE)
```



Now, we can actually interpret the model. First, from the model output, we see where each country is classified an, more importantly, the within cluster sum of squares is computed, along with the ration between_SS/total_SS, which is summing the 1-sum(within sum of squares)/total sum of square. We want, as we said earlier, the variation to come from Between Groups and not Within Groups. So a bigger ratio is better. In this case, it is 83.6%, which is a good level of explanation between clusters over the total variation in the data.

Looking at the plot along the two first principal components, we see that Singapore and Hong Kong form a distinct cluster, separated from the rest along the second principal component. The other countries are separated along the first principal component and we can roughly spot a progression from more deprived economies to OECD economies.

2e)

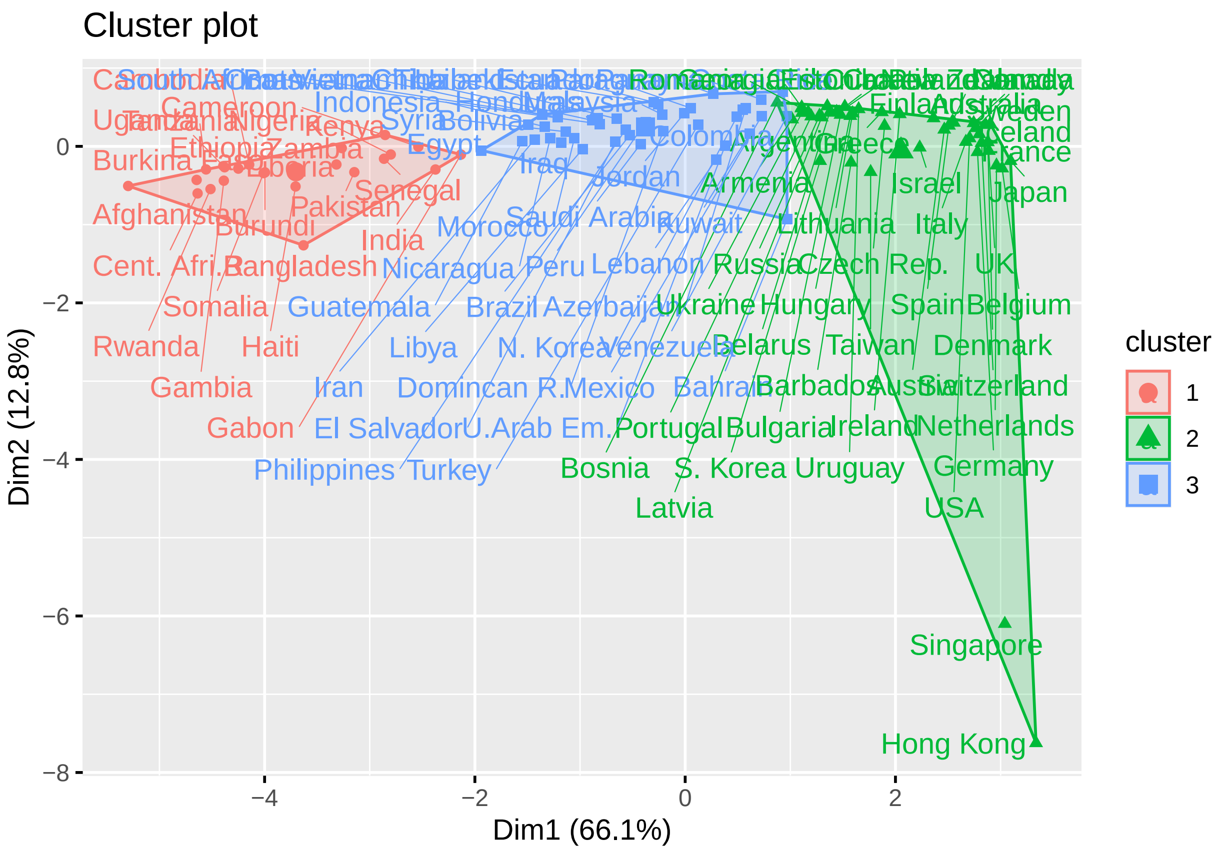
We now specify that we only want 3 clusters and repeat the outputs from before:

```r
km.res3 <- kmeans(data2.sc, 3, nstart = 25)
print(km.res3)
```

```
## K-means clustering with 3 clusters of sizes 22, 47, 40
##
## Cluster means:
##      density       flexp        gdp     infmort    literacy       mlexp
## 1 -0.1249148 -1.73459769 -0.8072637  1.66771248 -1.56670123 -1.69207209
## 2  0.1954116  0.77015514  0.8023150 -0.80520843  0.78755714  0.71103053
```

```
## 3 -0.1609055  0.04909644 -0.4987251  0.02887804 -0.06369397  0.09517877
##      popincr       urban
## 1  0.8267361 -1.27631720
## 2 -0.9427033  0.65403870
## 3  0.6529715 -0.06652101
##
## Clustering vector:
##  Afghanistan    Argentina      Armenia    Australia      Austria   Azerbaijan
##            1            2            2            2            2            3
##      Bahrain   Bangladesh     Barbados      Belarus      Belgium      Bolivia
##            3            1            2            2            2            3
##       Bosnia     Botswana       Brazil     Bulgaria Burkina Faso      Burundi
##            2            3            3            2            1            1
##     Cambodia     Cameroon       Canada Cent. Afri.R        Chile        China
##            1            1            2            1            2            3
##     Colombia   Costa Rica      Croatia         Cuba   Czech Rep.      Denmark
##            3            3            2            2            2            2
##  Domincan R.      Ecuador        Egypt  El Salvador      Estonia     Ethiopia
##            3            3            3            3            2            1
##      Finland       France        Gabon       Gambia      Georgia      Germany
##            2            2            1            1            2            2
##       Greece    Guatemala        Haiti     Honduras    Hong Kong      Hungary
##            2            3            1            3            2            2
##      Iceland        India    Indonesia         Iran         Iraq      Ireland
##            2            1            3            3            3            2
##       Israel        Italy        Japan       Jordan        Kenya       Kuwait
##            2            2            2            3            1            3
##       Latvia      Lebanon      Liberia        Libya    Lithuania     Malaysia
##            2            3            1            3            2            3
##       Mexico      Morocco     N. Korea  Netherlands  New Zealand    Nicaragua
##            3            3            3            2            2            3
##      Nigeria       Norway         Oman     Pakistan       Panama     Paraguay
##            1            2            3            1            3            3
##         Peru  Philippines       Poland     Portugal      Romania       Russia
##            3            3            2            2            2            2
##       Rwanda     S. Korea Saudi Arabia      Senegal    Singapore      Somalia
##            1            2            3            1            2            1
## South Africa        Spain       Sweden  Switzerland        Syria       Taiwan
##            3            2            2            2            3            2
##     Tanzania     Thailand       Turkey   U.Arab Em.           UK          USA
##            1            3            3            3            2            2
##       Uganda      Ukraine      Uruguay   Uzbekistan    Venezuela      Vietnam
##            1            2            2            3            3            3
##       Zambia
##            1
##
## Within cluster sum of squares by cluster:
## [1]  41.83624 185.06715  93.11430
##  (between_SS / total_SS =  63.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
fviz_cluster(km.res3, data=data2.sc, repel=TRUE)
```

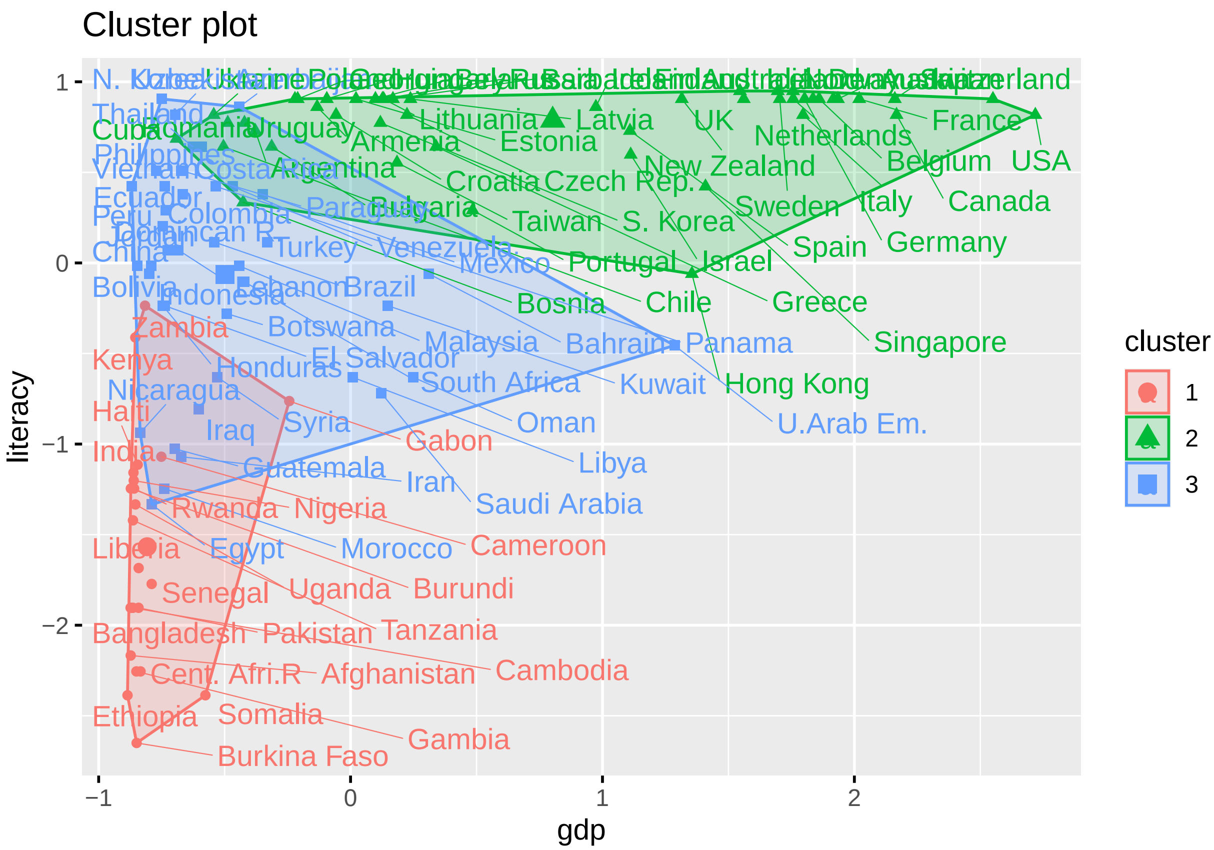


We have decreased the Between Sum of Square to Total Sum of Square ratio with only three clusters. We see that that Singapore and Hong Kong are now clustered with the rest of the more advanced economies.

2f)

We can plot specific variables as axes and emphasise the clusters according to the model in e) by using again the \texttt{fviz_cluster()} command simply changing the argument to display the gdp and literacy:

```r
fviz_cluster(km.res3, data=data2.sc, repel=TRUE, choose.vars=c(3,5))
```



Although there is some overlapping, note that these 2 variables are also distinctive features of the clusters.

Exercise 3

We first start to load the dataset and to obtain a rescaled version of it, just as before:

```
data3 <- read.csv("South.csv")
data3.sc <- scale(data3[,-1])
row.names(data3.sc) <- data3$state
```

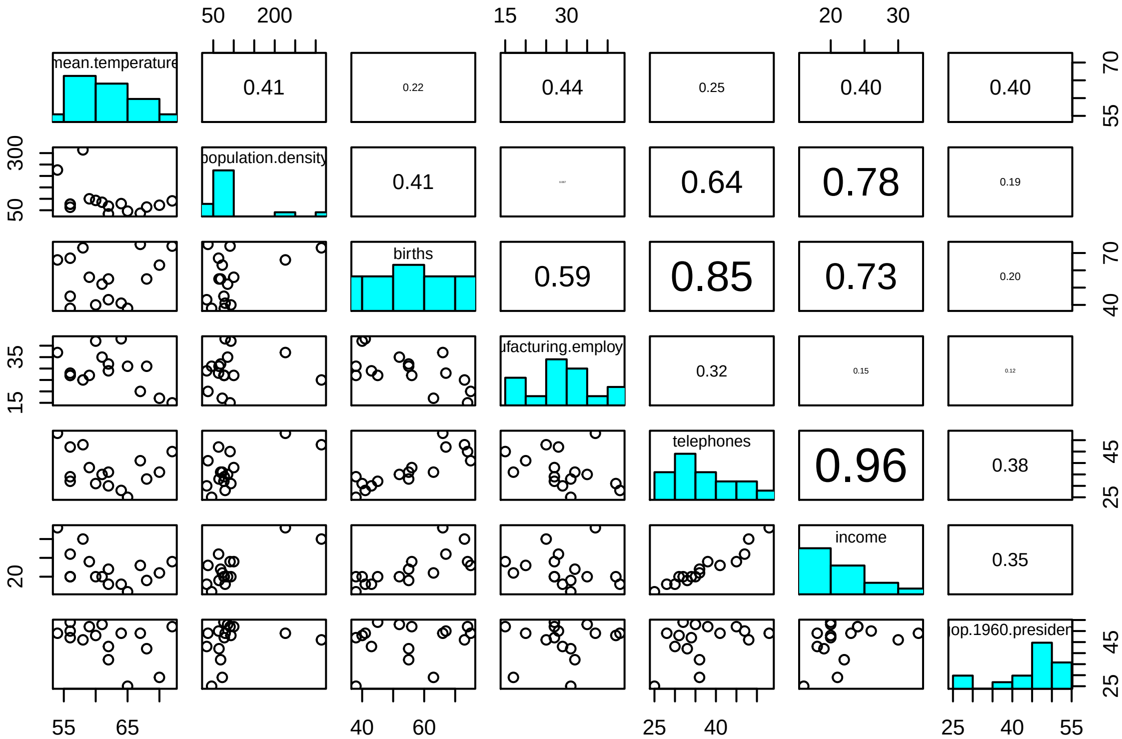
We can, as before, obtain a summary of the variables and also a visual to assess both the behaviour of individual variables and their relations to other variables. Here, since there are many variables to choose, we only display some of them, but feel free to explore the dataset:

```
summary(data3)
```

```
##     state           mean.altitude    mean.temperature mean.precipitation
##  Length:16          Min.   :  6.00   Min.   :54.00    Min.   :29
##  Class :character   1st Qu.: 33.75   1st Qu.:57.50    1st Qu.:44
##  Mode  :character   Median : 62.50   Median :61.50    Median :46
##                     Mean   : 64.44   Mean   :61.88    Mean   :47
##                     3rd Qu.: 82.50   3rd Qu.:65.50    3rd Qu.:49
##                     Max.   :170.00   Max.   :72.00    Max.   :68
##  population.density african.americans   median.age    urban.population
##  Min.   : 34.00     Min.   : 5.00      Min.   :23.00   Min.   :20.00
##  1st Qu.: 63.75     1st Qu.:13.50      1st Qu.:26.00   1st Qu.:21.00
##  Median : 76.50     Median :19.50      Median :27.50   Median :22.00
```

```
##  Mean   : 95.31    Mean   :20.94    Mean   :27.44   Mean   :22.00
##  3rd Qu.: 91.50    3rd Qu.:29.25    3rd Qu.:29.00   3rd Qu.:22.25
##  Max.   :314.00    Max.   :42.00    Max.   :32.00   Max.   :25.00
##      births      rural.population manufacturing.employment  automobiles
##  Min.   :38.00   Min.   : 2.00    Min.   :15.00            Min.   :29.00
##  1st Qu.:42.50   1st Qu.: 7.00    1st Qu.:26.50            1st Qu.:33.75
##  Median :55.00   Median :12.00    Median :28.50            Median :36.00
##  Mean   :55.06   Mean   :12.62    Mean   :29.12            Mean   :36.19
##  3rd Qu.:66.25   3rd Qu.:16.25    3rd Qu.:32.75            3rd Qu.:38.25
##  Max.   :75.00   Max.   :31.00    Max.   :43.00            Max.   :45.00
##    telephones        income      federal.revenue    lawyers
##  Min.   :25.00   Min.   :16.00   Min.   :11.00   Min.   : 72.00
##  1st Qu.:31.75   1st Qu.:19.75   1st Qu.:15.00   1st Qu.: 81.25
##  Median :35.50   Median :20.50   Median :17.50   Median :111.00
##  Mean   :37.00   Mean   :22.12   Mean   :17.75   Mean   :109.69
##  3rd Qu.:42.00   3rd Qu.:24.00   3rd Qu.:21.00   3rd Qu.:125.75
##  Max.   :53.00   Max.   :33.00   Max.   :24.00   Max.   :175.00
##     doctors      white.infant.mortality  school.years    education.expense
##  Min.   : 76.0   Min.   :20.00          Min.   : 10.00   Min.   :27.00
##  1st Qu.: 94.0   1st Qu.:22.00          1st Qu.: 85.75   1st Qu.:31.50
##  Median :105.5   Median :23.00          Median : 87.00   Median :33.00
##  Mean   :109.8   Mean   :23.19          Mean   : 85.81   Mean   :36.56
##  3rd Qu.:119.2   3rd Qu.:24.25          3rd Qu.: 92.25   3rd Qu.:41.25
##  Max.   :158.0   Max.   :26.00          Max.   :108.00   Max.   :54.00
##  sound.plumbing  gop.1960.president gop.1964.president gop.1962.1964.governor
##  Min.   :45.00   Min.   :25.00      Min.   :32.00      Min.   : 0.00
##  1st Qu.:54.00   1st Qu.:42.75      1st Qu.:36.75      1st Qu.:33.50
##  Median :57.50   Median :48.50      Median :44.50      Median :41.00
##  Mean   :61.50   Mean   :45.31      Mean   :48.06      Mean   :34.19
##  3rd Qu.:66.75   3rd Qu.:50.50      3rd Qu.:54.75      3rd Qu.:44.25
##  Max.   :81.00   Max.   :54.00      Max.   :87.00      Max.   :49.00
```

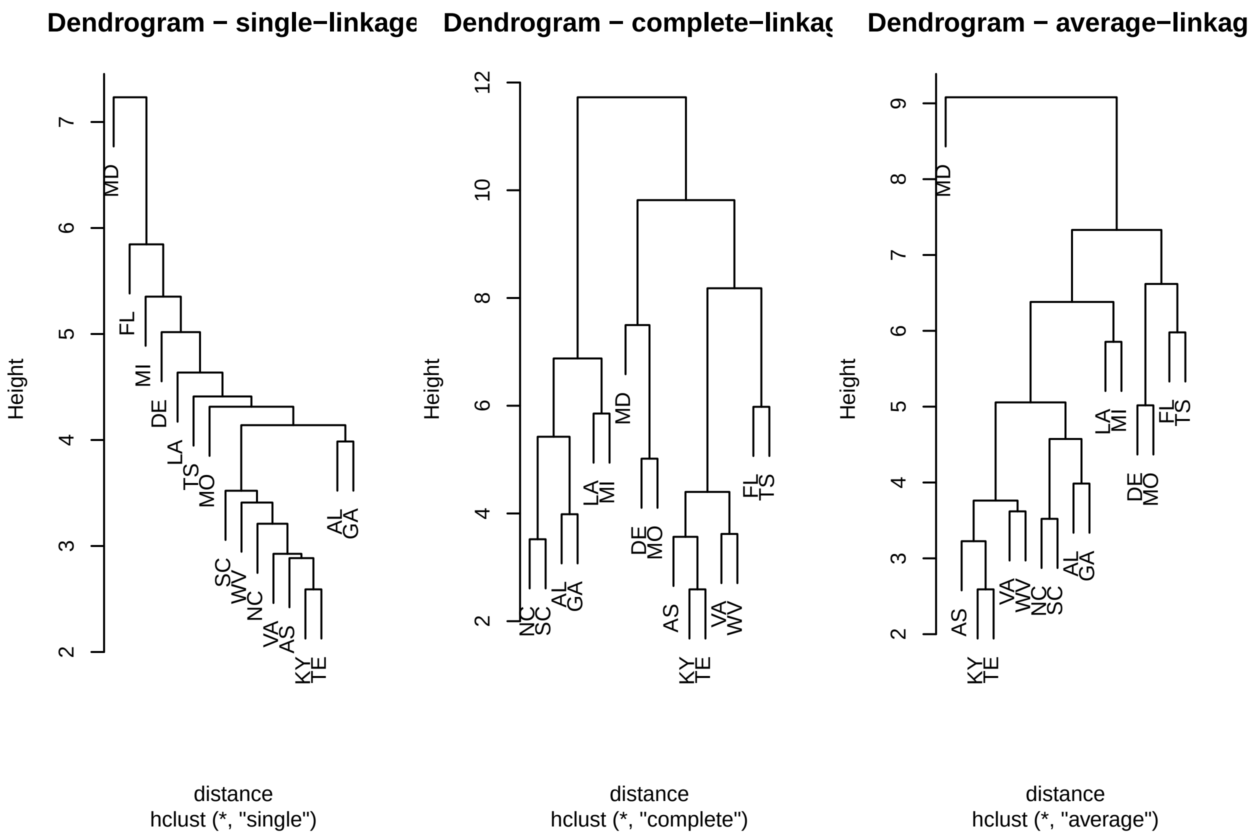
```
pairs(data3[,c(3,5,9,11,13,14,22)], upper.panel=panel.cor, diag.panel=panel.hist)
```



We can see, among other things, that the relations births and telephones and telephones and income are highly correlated. Also, the population density and income are positively associated.

We can now compute the distance matrix using all our quantitative information at hand and produce the three types of clustering we covered earlier:

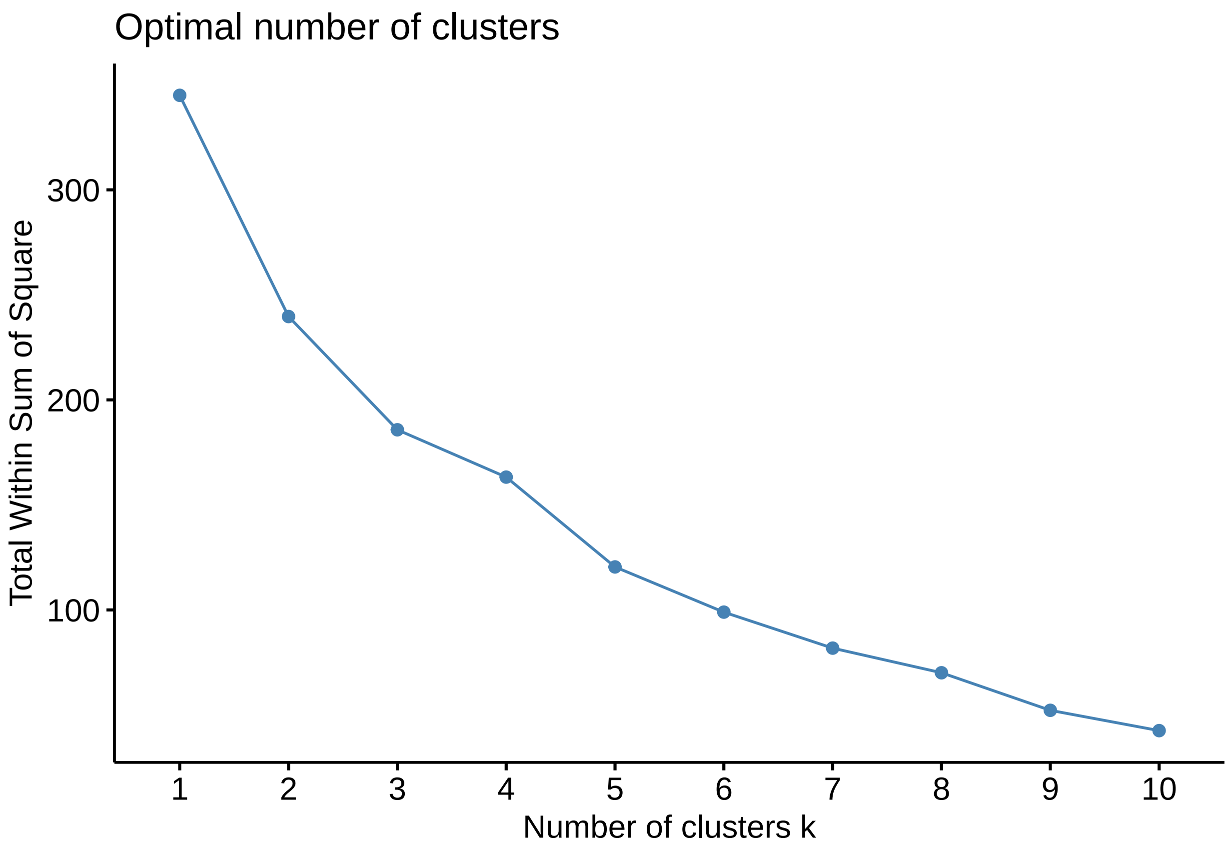
```
distance <- dist(data3.sc, diag=TRUE, upper=TRUE)
sing.clus <- hclust(distance, method="single")
co.clus <- hclust(distance, method="complete")
av.clus <- hclust(distance, method="average")

par(mfrow=c(1,3))
plot(sing.clus, labels=data3[,1], main="Dendrogram - single-linkage")
plot(co.clus, labels=data3[,1], main="Dendrogram - complete-linkage")
plot(av.clus, labels=data3[,1], main="Dendrogram - average-linkage")
```




```r
par(mfrow=c(1,1))
```

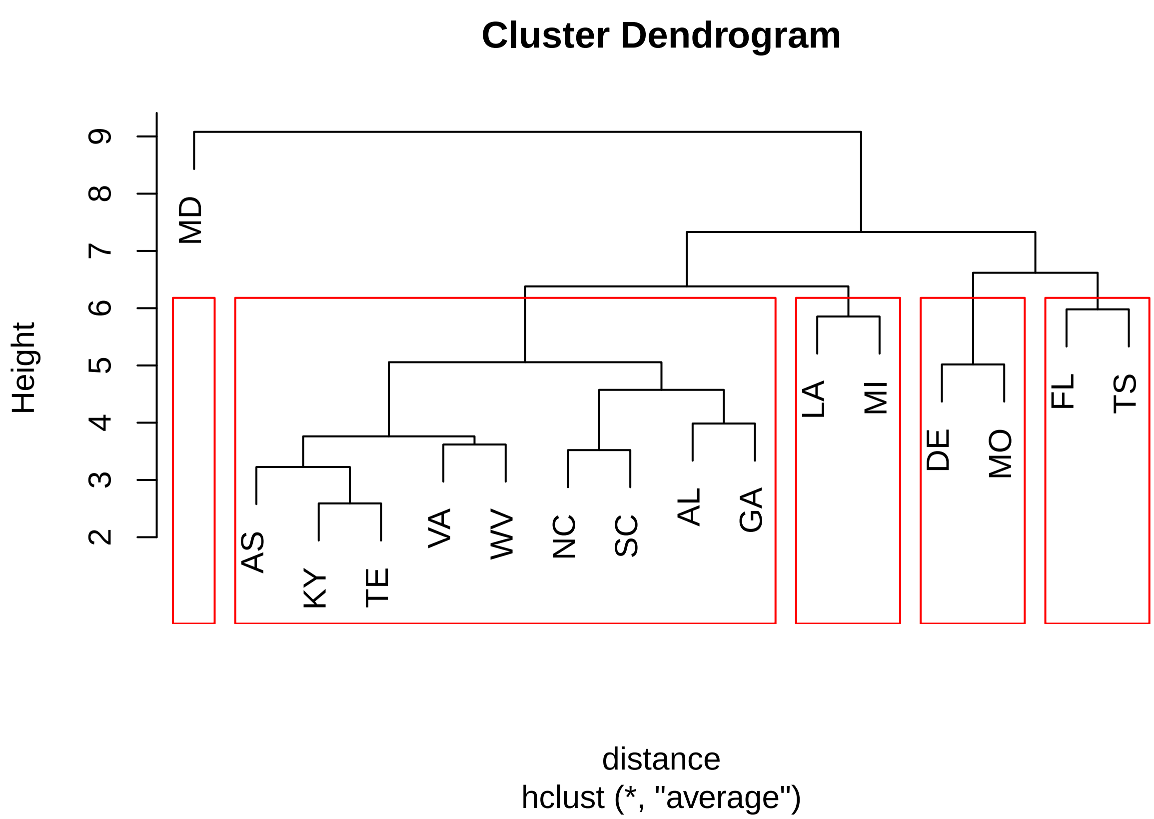
The clusters are quite different depending on the method. However, referring to the remarks made earlier, it could be wise to go with the average-linkage model. Let us now determine how many clusters are of interest with the elbow method:

```r
fviz_nbclust(data3.sc, kmeans, method="wss")
```



It seems that we should choose 5 clusters. Doing so, we can emphasise them on the Dendrogram above obtained with the average-linkage clustering:

```
cut <- cutree(av.clus, k=5)
plot(av.clus)
rect.hclust(av.clus, k=5, border="red")
```




```r
split(data3, cut)
```

```
## $`1`
##    state mean.altitude mean.temperature mean.precipitation population.density
## 1     AL            50               68                 68                 64
## 2     AS            65               62                 49                 34
## 5     GA            60               62                 47                 68
## 6     KY            75               56                 41                 76
## 11    NC            70               60                 44                 93
## 12    SC            35               64                 47                 79
## 13    TE            90               61                 47                 85
## 15    VA            95               59                 44                100
## 16    WV           150               56                 44                 77
##    african.americans median.age urban.population births rural.population
## 1                 30         26               22     55               12
## 2                 22         29               22     43               17
## 5                 29         26               24     55                9
## 6                  7         28               22     45               17
## 11                25         26               22     40               16
## 12                35         23               22     41               14
## 13                17         28               22     52               15
## 15                21         27               21     56                9
## 16                 5         29               20     38                7
##    manufacturing.employment automobiles telephones income federal.revenue
## 1                        31          39         33     19              22
## 2                        29          33         30     18              24
## 5                        32          37         36     22              19
## 6                        27          38         32     20              21
```

```
## 11                      42          35         31     20              15
## 12                      43          35         28     18              16
## 13                      35          35         35     20              22
## 15                      27          34         38     24              16
## 16                      27          31         34     20              18
##    lawyers doctors white.infant.mortality school.years education.expense
## 1       82      79                     25           89                28
## 2       79      91                     23           87                32
## 5      125     102                     23           88                33
## 6      108      95                     26           85                32
## 11      77     100                     22           85                32
## 12      72      80                     22           84                28
## 13     116     113                     24           86                30
## 15     114     108                     24           92                38
## 16      97     103                     26           87                33
##    sound.plumbing gop.1960.president gop.1964.president gop.1962.1964.governor
## 1              54                 42                 70                      4
## 2              48                 43                 43                     43
## 5              58                 37                 54                      0
## 6              53                 54                 36                     49
## 11             57                 48                 44                     43
## 12             54                 49                 59                      0
## 13             57                 53                 45                     49
## 15             66                 52                 46                     36
## 16             57                 47                 32                     45
##
## $`2`
##    state mean.altitude mean.temperature mean.precipitation population.density
## 3     DE             6               54                 45                226
## 10    MO            80               56                 35                 63
##    african.americans median.age urban.population births rural.population
## 3                 14         29               23     66                4
## 10                 9         32               21     67               12
##    manufacturing.employment automobiles telephones income federal.revenue
## 3                        37          40         53     33              15
## 10                       28          37         47     26              17
##    lawyers doctors white.infant.mortality school.years education.expense
## 3      115     135                     20          108                54
## 10      72     149                     21           93                45
##    sound.plumbing gop.1960.president gop.1964.president gop.1962.1964.governor
## 3              80                 49                 39                     49
## 10             66                 50                 36                     38
##
## $`3`
##    state mean.altitude mean.temperature mean.precipitation population.density
## 4     FL            10               72                 56                 91
## 14    TS           170               67                 29                 37
##    african.americans median.age urban.population births rural.population
## 4                 18         31               20     74                2
## 14                12         27               22     75                7
##    manufacturing.employment automobiles telephones income federal.revenue
## 4                        15          45         45     24              11
## 14                       20          42         41     23              15
##    lawyers doctors white.infant.mortality school.years education.expense
```

```
## 4      150     142                    23          106                41
## 14     144     111                    26          101                40
##    sound.plumbing gop.1960.president gop.1964.president gop.1962.1964.governor
## 4              78                 52                 49                     44
## 14             69                 49                 37                     26
##
## $`4`
##   state mean.altitude mean.temperature mean.precipitation population.density
## 7    LA            10               70                 63                 72
## 9    MI            30               65                 49                 46
##   african.americans median.age urban.population births rural.population
## 7                32         25               25     63                7
## 9                42         24               24     38               23
##   manufacturing.employment automobiles telephones income federal.revenue
## 7                       17          32         36     21              21
## 9                       31          29         25     16              21
##   lawyers doctors white.infant.mortality school.years education.expense
## 7     128     114                     21           86                42
## 9     101      76                     23           86                27
##   sound.plumbing gop.1960.president gop.1964.president gop.1962.1964.governor
## 7             61                 29                 57                     39
## 9             45                 25                 87                     38
##
## $`5`
##   state mean.altitude mean.temperature mean.precipitation population.density
## 8    MD            35               58                 44                314
##   african.americans median.age urban.population births rural.population
## 8                17         29               20     73               31
##   manufacturing.employment automobiles telephones income federal.revenue
## 8                       25          37         48     30              11
##   lawyers doctors white.infant.mortality school.years education.expense
## 8     175     158                     22           10                50
##   sound.plumbing gop.1960.president gop.1964.president gop.1962.1964.governor
## 8             81                 46                 35                     44
```

Now, we see that Maryland (MD) is in its own cluster. This makes sense, since Maryland can be considered a Northern state with respect to its characteristics, geographically since it is much more of a mild climate, with cold winters, and much warmer than the other Southern states.

For the rest, we distinguish a large cluster of mainly Southern states among which neighours tend to cluster together, for geographical, climatic and industrial reasons mainly. Political alikeness is also a reason for them to cluster together.

Louisiana and Mississippi, being both Deep South states, with very similar geographical feature and political inclination, as well as similar industries (mostly farming) are clustered together. Florida and Texas, being more populated state, with industries relying on oil, are are clustered together. Delaware and Missouri also cluster together, which makes sense when looking at the type of industry and population dynamics there.

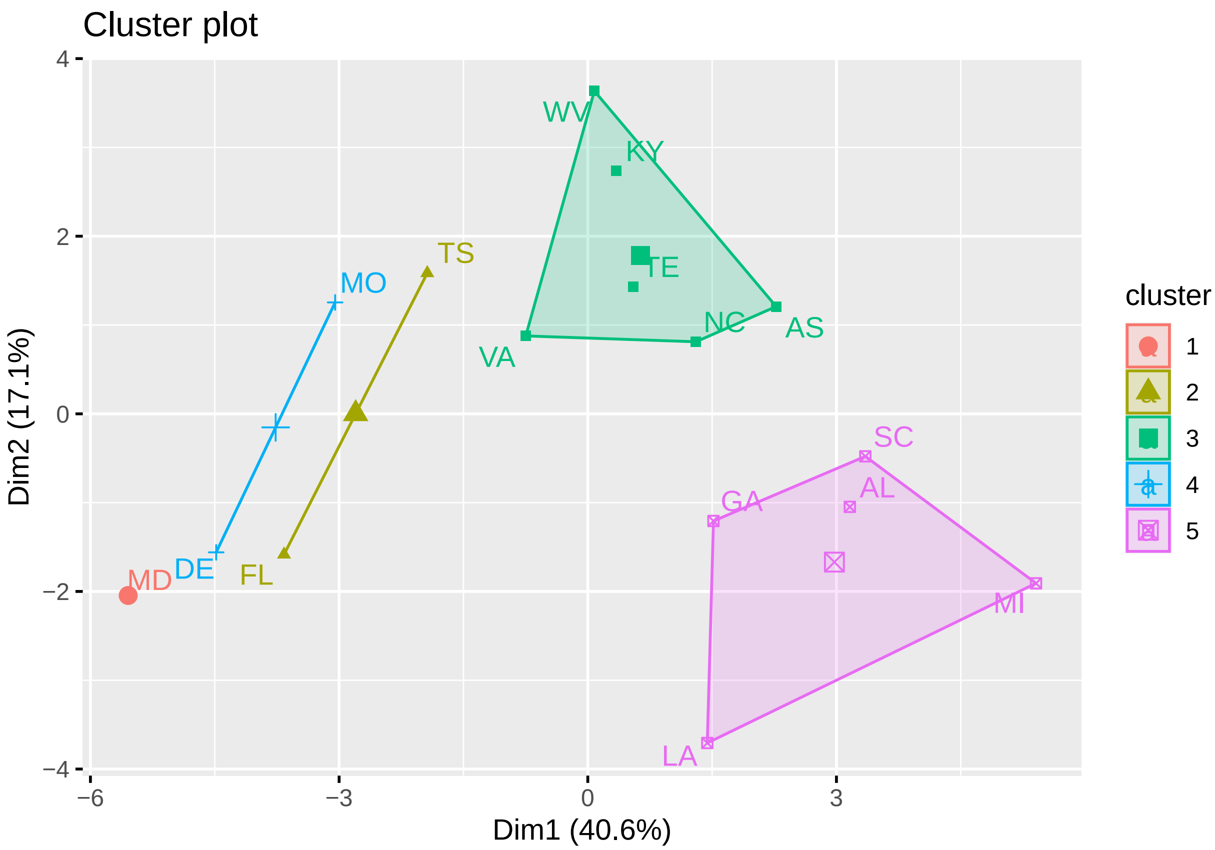
We now switch to non-hierarchical method, using again 5 clusters for our data:

```
km <- kmeans(data3.sc, 5, nstart = 25)
print(km)
```

```
## K-means clustering with 5 clusters of sizes 1, 2, 6, 2, 5
##
## Cluster means:
##   mean.altitude mean.temperature mean.precipitation population.density
```

```
## 1    -0.6283151       -0.7156726         -0.3185965          3.0146270
## 2     0.5456069        1.4082591         -0.4778948         -0.4316456
## 3     0.5633936       -0.5309829         -0.2300975         -0.2455469
## 4    -0.4575628       -1.2697418         -0.7433919          0.6780541
## 5    -0.5856270        0.7249071          0.8283510         -0.4068325
##   african.americans median.age urban.population     births rural.population
## 1        -0.3712929  0.6453252       -1.3693064  1.3452442         2.497824
## 2        -0.5598862  0.6453252       -0.6846532  1.4577385        -1.104480
## 3        -0.4498735  0.1634824       -0.3423266 -0.7046517         0.118944
## 4        -0.8899244  1.2648373        0.0000000  0.8577689        -0.628704
## 5         1.1940310 -1.0893089        0.9585145 -0.3496697         0.050976
##   manufacturing.employment automobiles telephones     income federal.revenue
## 1               -0.5233833   0.1972779  1.4015298  1.7500000      -1.7275541
## 2               -1.4749894   1.7755011  0.7644708  0.3055556      -1.2156862
## 3                0.2590483  -0.4501983 -0.4671766 -0.3981481       0.4052287
## 4                0.4282227   0.5614833  1.6563534  1.6388889      -0.4478844
## 5                0.2125254  -0.4340114 -0.6880237 -0.6500000       0.5246646
##      lawyers    doctors white.infant.mortality school.years education.expense
## 1  2.1705464  1.9357902             -0.6351236  -3.51183635         1.6557676
## 2  1.2400155  0.6720101              0.7019787   0.81933197         0.4851784
## 3 -0.3717969 -0.3243034              0.5236984   0.05500815        -0.4595076
## 4 -0.5379632  1.2938701             -1.4373850   0.68036401         1.5941576
## 5 -0.2687739 -0.7843461             -0.2072509   0.03647909        -0.6114788
##   sound.plumbing gop.1960.president gop.1964.president gop.1962.1964.governor
## 1      1.7781237         0.08172805         -0.8925258             0.56687542
## 2      1.0942300         0.61667526         -0.3459071             0.04693873
## 3     -0.4711268         0.49779810         -0.4825618             0.57650388
## 4      1.0486371         0.49779810         -0.7217075             0.53799005
## 5     -0.6474194        -1.05949268          1.1846252            -1.03915125
##
## Clustering vector:
## AL AS DE FL GA KY LA MD MI MO NC SC TE TS VA WV
##  5  3  4  2  5  3  5  1  5  4  3  5  3  2  3  3
##
## Within cluster sum of squares by cluster:
## [1]  0.00000 17.87878 34.06851 12.58765 55.92838
##  (between_SS / total_SS =  65.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
fviz_cluster(km, data=data3.sc, repel=TRUE)
```



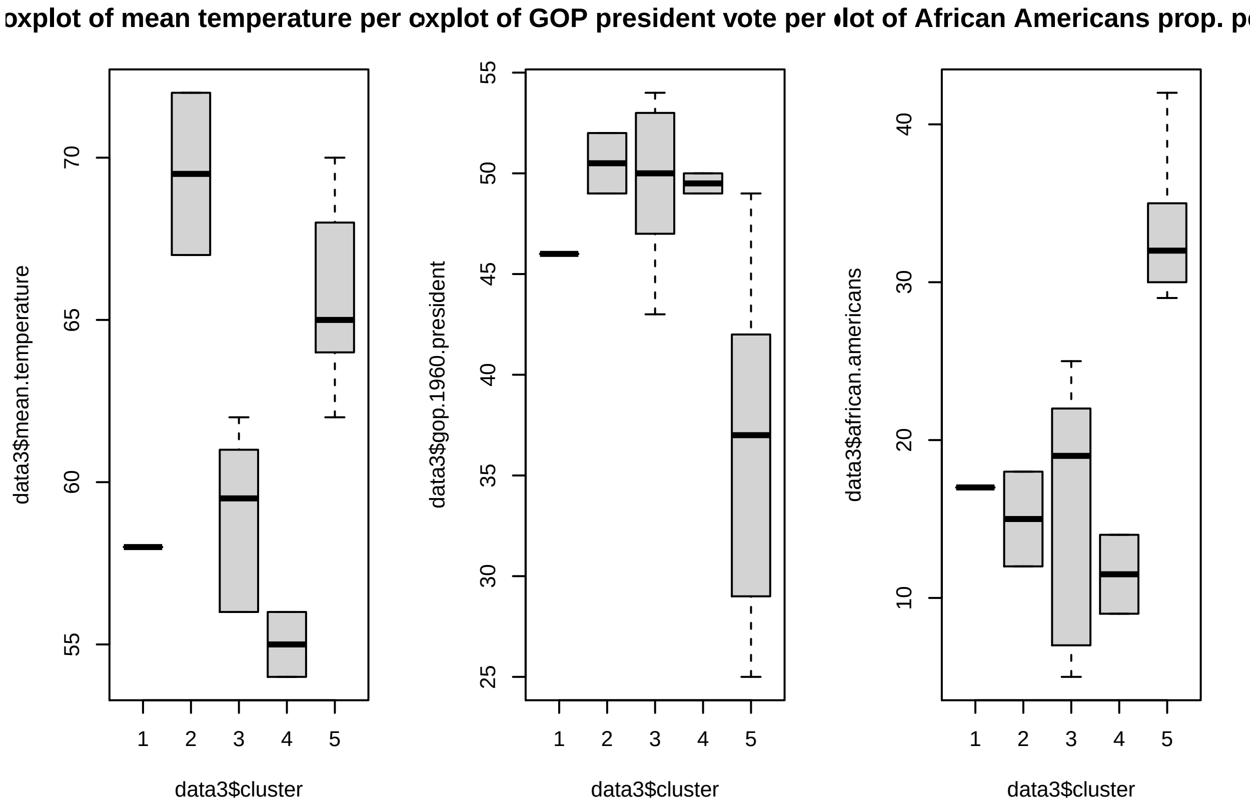
The picture is very similar to the above except that the Deep South states (South Carolina, Alabama, Mississippi, Georgia and Louisiana) are clustered together. The (less) Southern, more industrial states like West Virginia, Kentucky, Tennessee, Virginia, North Carolina and Arkansas are clustered together. Texas and Florida are still grouped together, same with Missouri and Delaware and Maryland is still its own cluster. The clear patterns are that the pink cluster is clearly positively associated with the first principle component and negatively with the second one, while the green cluster is positively associated with the second principal component.

We can now add a new column to the initial dataset with the cluster membership:

```
data3 <- mutate(data3, cluster=as.vector(km$cluster))
```

This allows us to then assess the individual variables and their variation across clusters, to see if they can be considered distinctive features of these clusters. For example, we could obtain boxplots of the mean temperature, GOP presidential vote share in 1960 and proportion of African Americans per cluster:

```
par(mfrow=c(1,3))
boxplot(data3$mean.temperature~data3$cluster, main="Boxplot of mean temperature per cluster")
boxplot(data3$gop.1960.president~data3$cluster, main="Boxplot of GOP president vote per cluster")
boxplot(data3$african.americans~data3$cluster, main="Boxplot of African Americans prop. per cluster")
```




```r
par(mfrow=c(1,1))
```

We see that the mean temperature is a very distinctive feature, since clusters 2 followed by cluster 5 have the highest temperatures where the other have lower temperatures. Observe however that we have very few observations in most of the clusters.

The GOP support does not really separate the clusters well but we have a sense of variation per cluster.

Finally, the share of African Americans is higher in the fifth cluster than it is in the other clusters that have similar proportions, although cluster 3 has quite some spread in this variable.