MULTIVARIATE ANALYSIS
CLUSTERING AND PROFILING

Authors

Goktug Cengiz (Y6545603R)
Henry Qiu (X8281348Z)

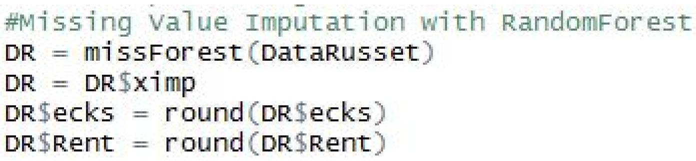
Date
11/04/2019

This homework consist in the practice of Clustering and Profiling, there are some requirements that are splitted into different main parts in our report. These parts are: the imputation of the missing values, performing the Principal Component Analysis, performing a hierarchical clustering, and representing the clusters in the first factorial display.

For each one of these parts we will provide a brief explanation of the procedure (the detailed implementation will be provided in the the submitted script), the results obtained and the answers to the corresponding questions in the statement.

1. Imputation of the Missing Values (Exercise 1)

Before starting, we have to threat all the missing values of the dataset so that the the PCA analysis make sense and gives us significant results, to deal with this problem we will do the imputation of missing values. We used Random Forest in this project for imputation. In Random Forest, the basic idea is to do a quick replacement of missing data and then iteratively improve the missing imputation using proximity. In order to do that, we used "missForest" package in R.

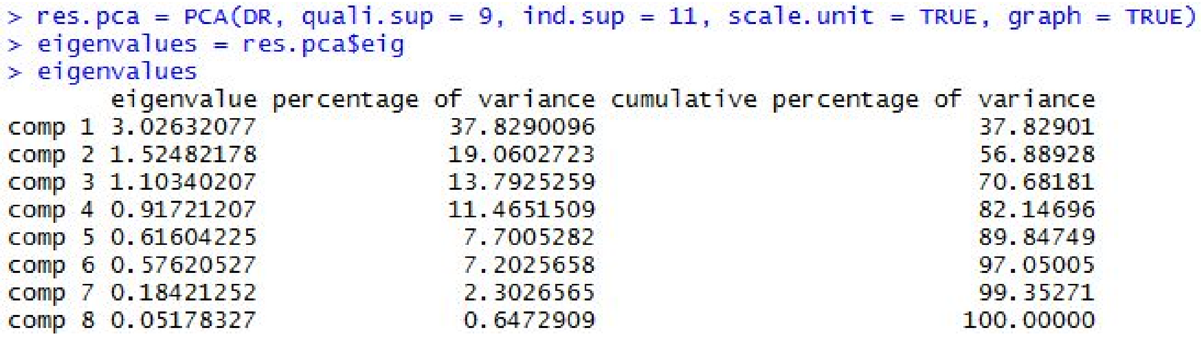


```
#Missing Value Imputation with RandomForest
DR = missForest(DataRusset)
DR = DR$ximp
DR$ecks = round(DR$ecks)
DR$Rent = round(DR$Rent)
```

We can see above the instructions performed for the imputation of missing values.

2. Principal Component Analysis (Exercise 1, 2, 3)

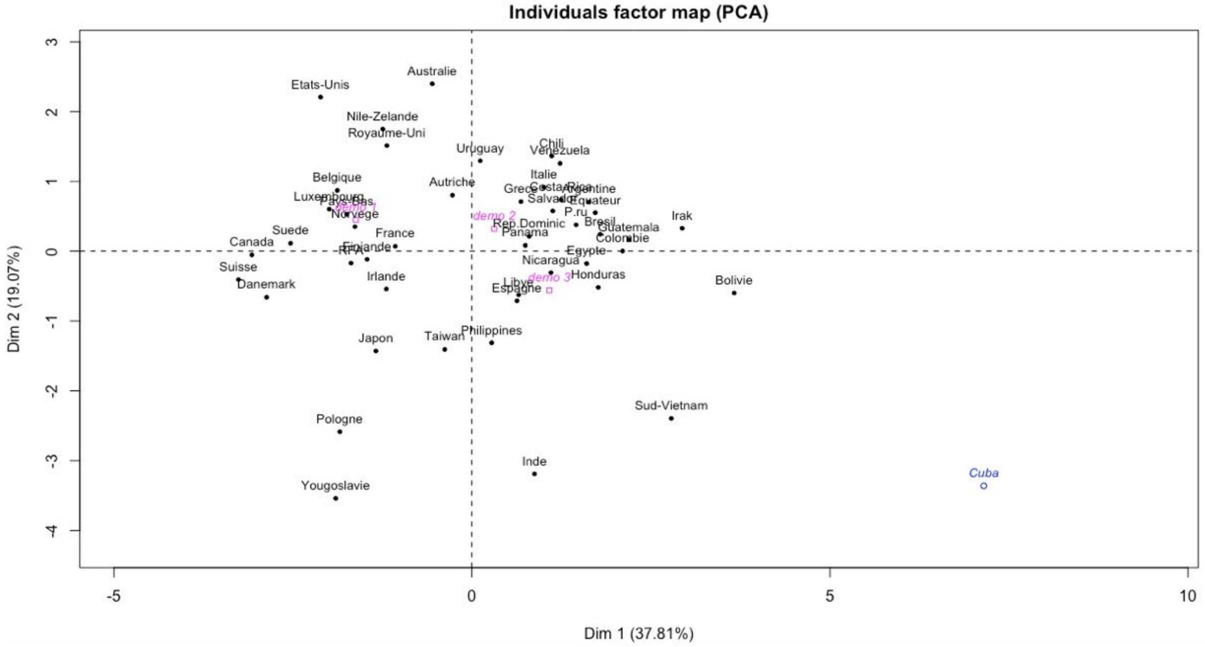
As it is not specified as in the previous homework did, we have computed the Principal Component Analysis with the FactoMiner package of R. We also considered that we can take into account that we will assign uniform weights for the instances. However, as we proved in the previous homework, we saw that for the Russet data set, the performance of using standardized values were better for the showing of the information, for that reason in this homework we will also use standardized values for the computing of the PCA.



```
> res.pca = PCA(DR, quali.sup = 9, ind.sup = 11, scale.unit = TRUE, graph = TRUE)
> eigenvalues = res.pca$eig
> eigenvalues
       eigenvalue percentage of variance cumulative percentage of variance
comp 1 3.02632077             37.8290096                          37.82901
comp 2 1.52482178             19.0602723                          56.88928
comp 3 1.10340207             13.7925259                          70.68181
comp 4 0.91721207             11.4651509                          82.14696
comp 5 0.61604225              7.7005282                          89.84749
comp 6 0.57620527              7.2025658                          97.05005
comp 7 0.18421252              2.3026565                          99.35271
comp 8 0.05178327              0.6472909                         100.00000
```

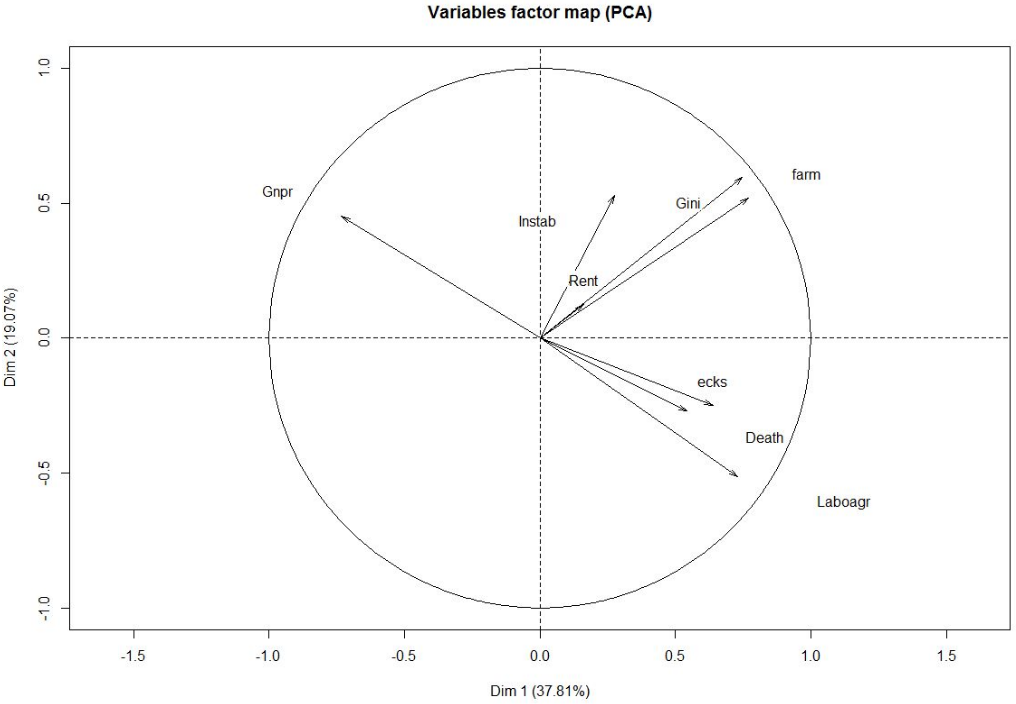
We can see above the computation of the PCA using FactoMineR and the eigenvalues computed.

- <u>Interpretation</u>



We can see above the individuals factor map figure, where the axis represents the dimensions. The dimension 1 and 2 represent the dimensions where the cloud of projection of individuals have more variance.

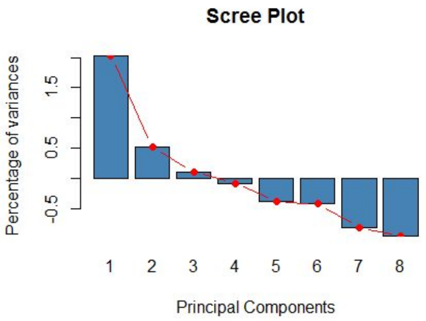
In the individuals factor map figure we can also see that Cuba is clearly an outlier. With the variable marked clearly we can also see where are the different categories located: demo 1 is in the second quadrant, were countries as Luxembourg, Norvege, Belgique, etc are projected near this category of the variable, this makes sense because analyzing the history context, they are countries that politically have been stable; demo 2 is in the first quadrant, where the countries politically more unstable have been represented, like Republica Dominicana, Panamá, etc; and finally, demo 3 represented by the dictatorship countries as Spain during that time.

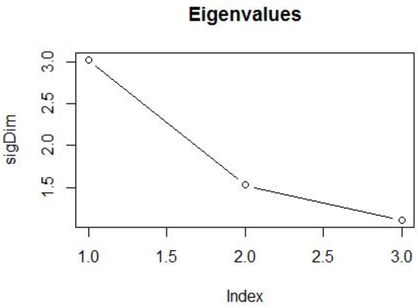


In the factor map image can observe that the variables Ecks and Death are very correlated, which completely make sense because they indicate the index of violent conflicts and the deaths, it is natural to think that higher index of violent conflicts produce more deaths. These variables and Laboagr that indicates the active population in agriculture explain more the second dimension. We can also conclude that these three variables are completely not related to Gnpr that indicates the GNP per capita, we can observe this from the completely opposited directions of the arrows in the map. A similar behaviour happen with the variables Rent, Gini and Farm, that are very correlated between them but seems not related to the other variables.

What is more, we can observe that variable Instab seems more significant for dimension 2, because the projection is much bigger in the second dimension (y axis) than in the first dimension. However, for other attributes like specially Laboagr, Ecks, Gini, Farm, and so on, seem much more significant for dimension one. Taking into account that some variables as Rent that has small value of projection either in dimension 1 or dimension 2, this makes us think that there might be another significant component.

- Significant dimensions





One of the ways to determine which dimensions are relevant to consider is the Kaiser rule. This consists in the difference between the eigenvalues of each component obtained with the mean of the eigenvalues of all the components, the components which its difference is higher than 0, which means that their eigenvalues are above the mean, are considered significant.

In the figures above we can see the Kaiser rule applied to first all the principal components, where it is clearly shown that the components that has eigenvalues smaller than the mean are represented with a bar growing under the axis 0. In the second figure we can see the components we are taking into account for our future calculations. In this case the number of significant dimensions we retain is 3.

3. Hierarchical Clustering (Exercise 4, 5)

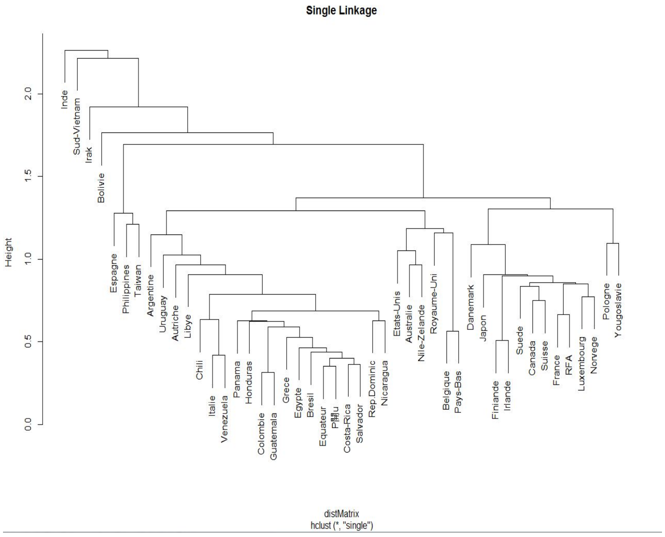
The hierarchical clustering works in the following way: it starts by calculating the distance between every pair of observation points and store it in a distance matrix; it then starts assigning clusters, each point belongs to its own cluster, so as many clusters as points we have are created; then it starts merging the closest pairs of points based on the distances from the distance matrix, as a result the number of clusters decrease in each iteration; then it computes newly the distance between the new cluster and the old ones and stores them in a new distance matrix. until all the clusters are merged into one single cluster.

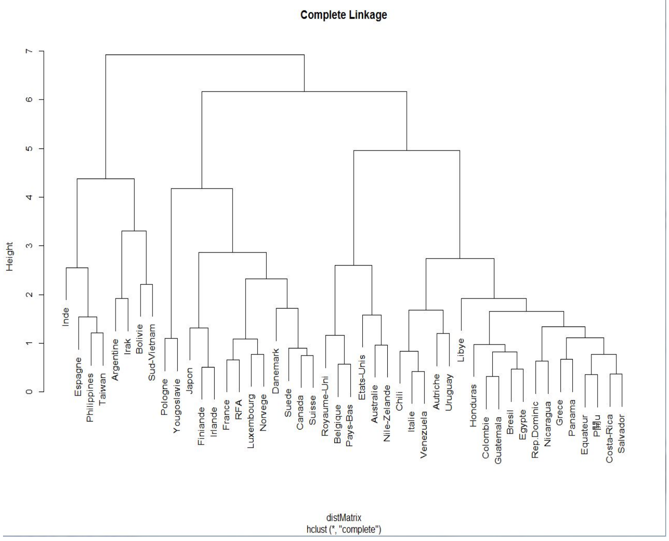
We want to do the clustering based on the projections of the individuals over the principal components. For this reason, we are going to compute the distance matrix with the projection of the individuals on the three dimensions. Taking into account that all the values we are treating are numerical continuous values, we can compute the euclidean distance for the construction of the matrix.

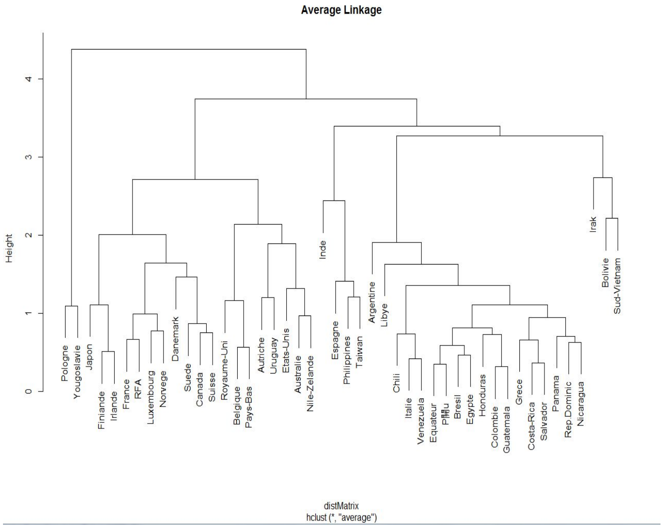
The distance between clusters clusters can be computed in different ways, they are called Linkage Methods, and each method can determine a different clustering decision, so that it will affect in how the clusters are assigned in the result. To see the performance of the algorithm with different linkage methods we can generate dendograms.

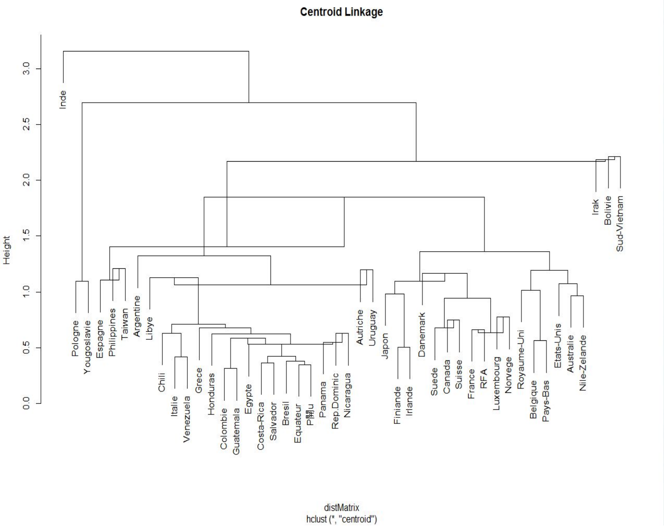
Dendograms show the evolution of the hierarchical clustering algorithm, on the horizontal axis we can see the instances that need to be classified, on the vertical axis we can see the height, that represents the distance of merge of the clusters that are merged in each iteration. We can observe that at the beginning we have so many clusters as instances we want to classify, and at the end we have only one cluster.

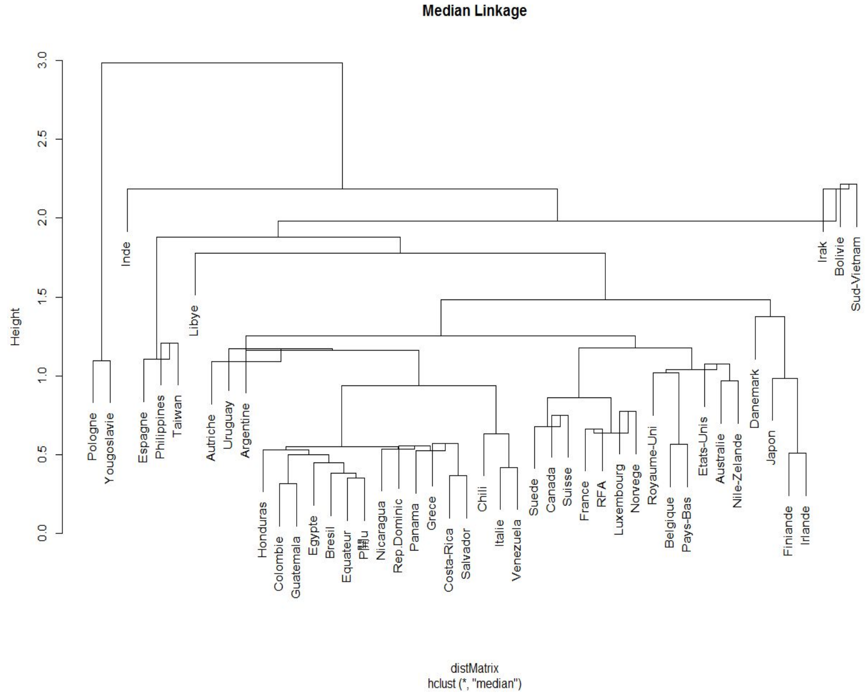
The linkage methods we have computed in this homework are: Single Linkage, Complete Linkage, Average Linkage, Centroid Linkage, Median Linkage and Ward’s Method.

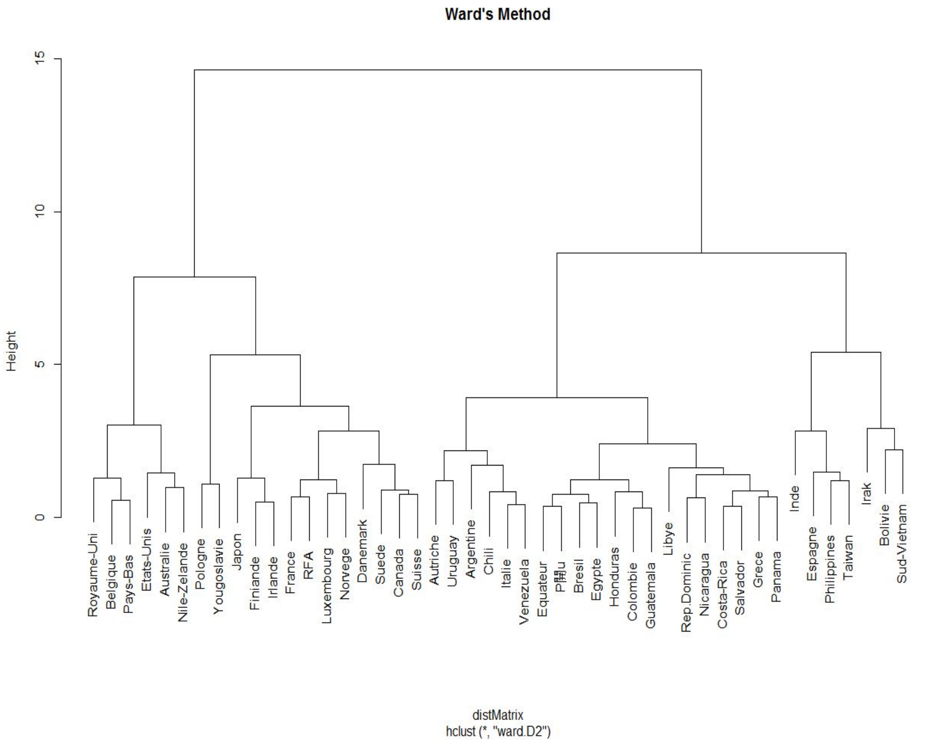

Single Linkage
Height
0.0
0.5
1.0
1.5
2.0
Inde
Sud-Vietnam
Irak
Bolivie
Espagne
Philippines
Taiwan
Argentine
Uruguay
Autriche
Libye
Chili
Italie
Venezuela
Panama
Honduras
Colombie
Guatemala
Grece
Egypte
Bresil
Equateur
P▯u
Costa-Rica
Salvador
Rep.Dominic
Nicaragua
Etats-Unis
Australie
Nlle-Zelande
Royaume-Uni
Belgique
Pays-Bas
Danemark
Japon
Finlande
Irlande
Suede
Canada
Suisse
France
RFA
Luxembourg
Norvege
Pologne
Yougoslavie
distMatrix
hclust (*, "single")


Complete Linkage
Height
0
1
2
3
4
5
6
7
Inde
Espagne
Philippines
Taiwan
Argentine
Irak
Bolivie
Sud-Vietnam
Pologne
Yougoslavie
Japon
Finlande
Irlande
France
RFA
Luxembourg
Norvege
Danemark
Suede
Canada
Suisse
Royaume-Uni
Belgique
Pays-Bas
Etats-Unis
Australie
Nlle-Zelande
Chili
Italie
Venezuela
Autriche
Uruguay
Libye
Honduras
Colombie
Guatemala
Bresil
Egypte
Rep.Dominic
Nicaragua
Grece
Panama
Equateur
P▯u
Costa-Rica
Salvador
distMatrix
hclust (*, "complete")


Average Linkage
Height
0
1
2
3
4
Pologne
Yougoslavie
Japon
Finlande
Irlande
France
RFA
Luxembourg
Norvege
Danemark
Suede
Canada
Suisse
Royaume-Uni
Belgique
Pays-Bas
Autriche
Uruguay
Etats-Unis
Australie
Nlle-Zelande
Inde
Espagne
Philippines
Taiwan
Argentine
Libye
Chili
Italie
Venezuela
Equateur
P▯u
Bresil
Egypte
Honduras
Colombie
Guatemala
Grece
Costa-Rica
Salvador
Panama
Rep.Dominic
Nicaragua
Irak
Bolivie
Sud-Vietnam
distMatrix
hclust (*, "average")


Centroid Linkage
Height
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Inde
Pologne
Yougoslavie
Espagne
Philippines
Taiwan
Argentine
Libye
Chili
Italie
Venezuela
Grece
Honduras
Colombie
Guatemala
Egypte
Costa-Rica
Salvador
Bresil
Equateur
P▯u
Panama
Rep.Dominic
Nicaragua
Autriche
Uruguay
Japon
Finlande
Irlande
Danemark
Suede
Canada
Suisse
France
RFA
Luxembourg
Norvege
Royaume-Uni
Belgique
Pays-Bas
Etats-Unis
Australie
Nlle-Zelande
Irak
Bolivie
Sud-Vietnam
distMatrix
hclust (*, "centroid")


Median Linkage
Height
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Pologne
Yougoslavie
Inde
Espagne
Philippines
Taiwan
Libye
Autriche
Uruguay
Argentine
Honduras
Colombie
Guatemala
Egypte
Bresil
Equateur
P▯u
Nicaragua
Rep.Dominic
Panama
Grece
Costa-Rica
Salvador
Chili
Italie
Venezuela
Suede
Canada
Suisse
France
RFA
Luxembourg
Norvege
Royaume-Uni
Belgique
Pays-Bas
Etats-Unis
Australie
Nlle-Zelande
Danemark
Japon
Finlande
Irlande
Irak
Bolivie
Sud-Vietnam
distMatrix
hclust (*, "median")



In the figures above we can the result of the dendograms generated performing the different linkage methods respectively: Single Linkage, Complete Linkage, Average Linkage, Centroid Linkage, Median Linkage and Ward’s Method.

Single-linkage calculates the minimum distance between the clusters before merging, it is robust to small changes between a pair of objects, and it is sensitive to noise and outliers. In our case we can see that the clusters we have generated are very unbalanced, if we perform a cut in any height the number of instances belonging to a clusters is very small, maybe it is not suitable to use this linkage method.

Complete-linkage calculates the maximum distance between the clusters before merging, it is robust to outliers and noise and it can break large clusters. In our case we can see that the structure of the dendogram generated is quite good, clusters seem well structured and balanced, this can be a suitable method to consider.

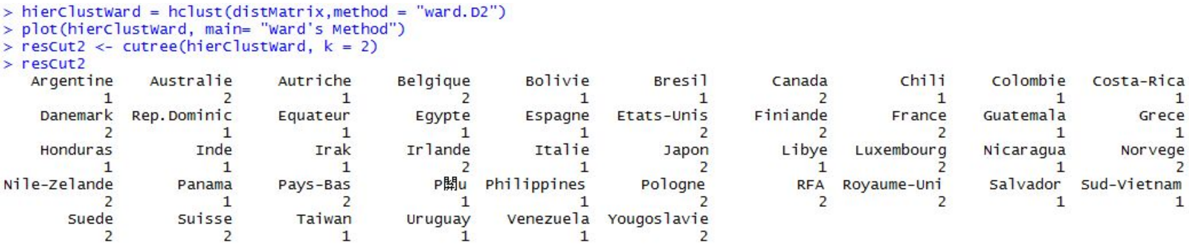
Average-linkage calculates the average distance between the clusters before merging. In our case we see that the dendogram generated is also bad balanced, we can see that the clusters in the extremes of the figure contains very few instances. We can also discard this method.

Centroid-linkage consists in finding the centroid of the two clusters and calculate the distance between these centroid before merging. This method has a problem, it produces frequently inversions in the dendogram, that is exactly what we can observe in our execution.

Median-linkage calculates the median distance between the clusters. There is a problem in this method, that is the median representativeness do not depend on the size of the clusters. In our case we can observe that the dendogram generated also presents inversions. This is why we can also discard this method.

Ward’s method consists in calculating the distance between the clusters as the sum of squared deviations from points to centroids. The aim of the Ward’s linkage is to minimize the sum of squares within clusters. We can observe that the dendogram generated is very well balanced.

Finally, we have chosen the Ward’s method, firstly because the clusters are even better balanced that the complete linkage, what is more, the heights of the final merges between merges are very high , so performing cuts within the final merges can generate small amount of clusters, that is our aim in the clustering process.



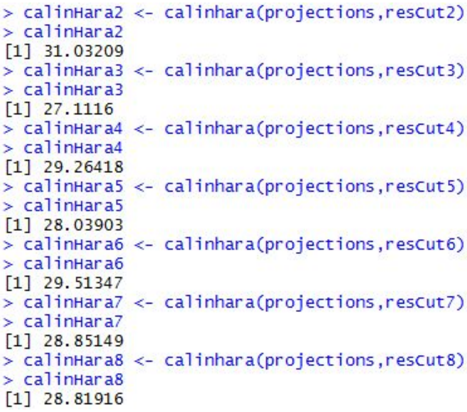
```
> hierClustWard = hclust(distMatrix,method = "ward.D2")
> plot(hierClustWard, main= "Ward's Method")
> resCut2 <- cutree(hierClustWard, k = 2)
> resCut2
   Argentine    Australie     Autriche     Belgique      Bolivie       Bresil       Canada        Chili     Colombie   Costa-Rica 
           1            2            1            2            1            1            2            1            1            1 
    Danemark  Rep.Dominic     Equateur       Egypte      Espagne   Etats-Unis     Finiande       France    Guatemala        Grece 
           2            1            1            1            1            2            2            2            1            1 
    Honduras         Inde         Irak      Irlande       Italie        Japon        Libye   Luxembourg    Nicaragua      Norvege 
           1            1            1            2            1            2            1            2            1            2 
Nile-Zelande       Panama     Pays-Bas         P關u  Philippines      Pologne          RFA  Royaume-Uni     Salvador  Sud-Vietnam 
           2            1            2            1            1            2            2            2            1            1 
       Suede       Suisse       Taiwan      Uruguay    Venezuela  Yougoslavie 
           2            2            1            1            1            2 
```

We can see above the instructions used to compute hierarchical clustering.

- <u>Decision of final clusters</u>

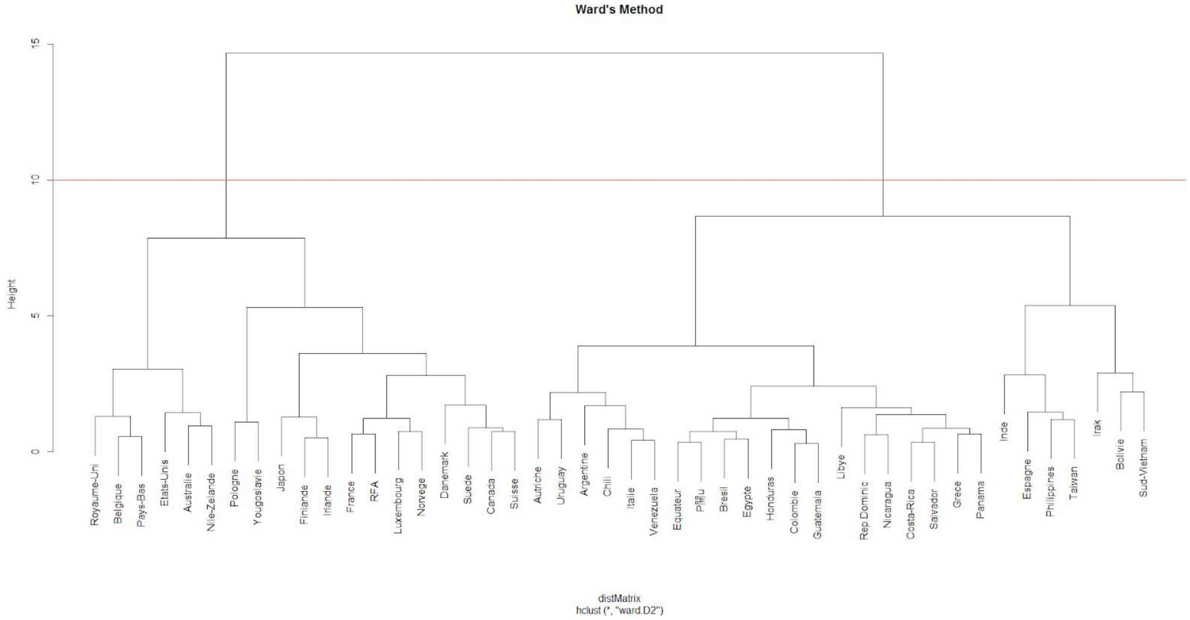
Usually to define the number of clusters is required the knowledge of an expert, but many times this knowledge is not provided to us, for the reason there are different methods that helps in the decision of the number of clusters. One of the method is to perform a cut in the tree of the dendogram to decide the clustering, it does not exist a rule of cutting that guarantees the obtaintion of the best clusters, usually the best cuts are performed within the merges with the highest height, in our case we should perform the cut in the height 10. In that case, the number of clusters generated will be 2.

To know how good is our decision of clusters, we can perform the calculation of the Calinski-Harabasz index. The Calinski Harabasz index consists in the ratio between the between classes variability(indicates the distinguishability between the classes) and the within classes variability (indicates the homogeneity of a cluster). The value mainly indicates how much the different classes more variable are between them respect to the variability inside the classes, that means that if two classes are more variated between them, this means that they are more different, that indicates that they should belong to different class, and that the clustering decision taken is correct. This means that a higher value of the Calinski-Harabasz index indicates that the clustering is better performed.

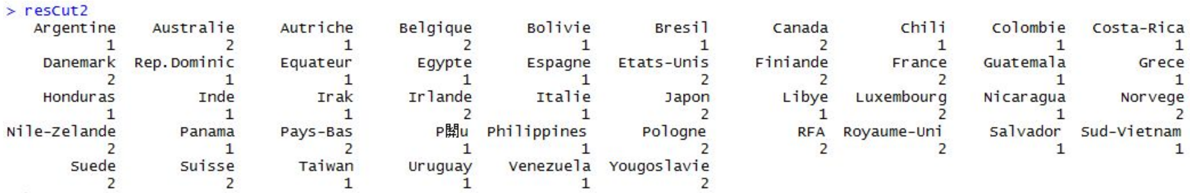


```
> calinHara2 <- calinhara(projections,resCut2)
> calinHara2
[1] 31.03209
> calinHara3 <- calinhara(projections,resCut3)
> calinHara3
[1] 27.1116
> calinHara4 <- calinhara(projections,resCut4)
> calinHara4
[1] 29.26418
> calinHara5 <- calinhara(projections,resCut5)
> calinHara5
[1] 28.03903
> calinHara6 <- calinhara(projections,resCut6)
> calinHara6
[1] 29.51347
> calinHara7 <- calinhara(projections,resCut7)
> calinHara7
[1] 28.85149
> calinHara8 <- calinhara(projections,resCut8)
> calinHara8
[1] 28.81916
```

In the figure above we have the Calinski-Harabasz index for considering 2, 3, 4, 5, 6, 7 and 8 clusters respectively. Taking into account the Calinski-Harabasz indexes obtained, the most appropriate number of clusters should be 2, because it has the highest Calinski-Harabasz index: 30.98835.



In the figure we can see how the cut of the tree is performed, the cut is represented by the red line. With this cut we have splitted 4 clusters.

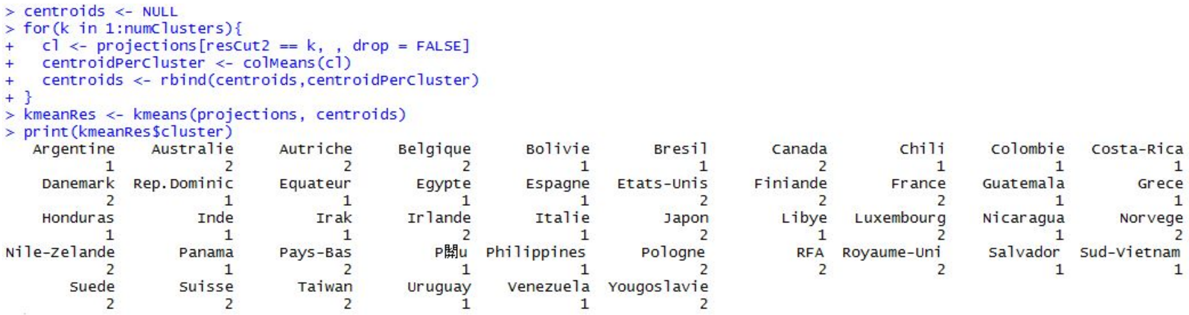


```
> resCut2
  Argentine    Australie     Autriche     Belgique      Bolivie       Bresil       Canada        Chili     Colombie   Costa-Rica 
          1            2            1            2            1            1            2            1            1            1 
   Danemark  Rep.Dominic     Equateur       Egypte      Espagne   Etats-Unis     Finiande       France    Guatemala        Grece 
          2            1            1            1            1            2            2            2            1            1 
   Honduras         Inde         Irak      Irlande       Italie        Japon        Libye   Luxembourg    Nicaragua      Norvege 
          1            1            1            2            1            2            1            2            1            2 
Nile-Zelande       Panama     Pays-Bas         P閣u  Philippines      Pologne          RFA  Royaume-Uni     Salvador  Sud-Vietnam 
          2            1            2            1            1            2            2            2            1            1 
      Suede       Suisse       Taiwan      Uruguay    Venezuela  Yougoslavie 
          2            2            1            1            1            2 
```

In the figure above we can see the results of the assignment of clusters to the individuals.

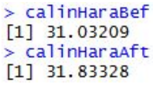
- <u>Consolidation operation</u>

Many times the results obtained by the hierarchical clustering is not good enough, we can perform a consolidation operation that consists in combining the k-means algorithm with hierarchical clustering. After deciding the number of classes present in our data, we can calculate the centroids of the resulting clusters and then perform a k-means algorithm taking as seeds the centroids previously calculated, in this way, some instances that was previously classified as part of one cluster can be changed to another one.



```
> centroids <- NULL
> for(k in 1:numClusters){
+   cl <- projections[resCut2 == k, , drop = FALSE]
+   centroidPerCluster <- colMeans(cl)
+   centroids <- rbind(centroids,centroidPerCluster)
+ }
> kmeanRes <- kmeans(projections, centroids)
> print(kmeanRes$cluster)
   Argentine    Australie     Autriche     Belgique      Bolivie       Bresil       Canada        Chili     Colombie   Costa-Rica 
           1            2            2            2            1            1            2            1            1            1 
    Danemark  Rep.Dominic     Equateur       Egypte      Espagne   Etats-Unis     Finiande       France    Guatemala        Grece 
           2            1            1            1            1            2            2            2            1            1 
    Honduras         Inde         Irak      Irlande       Italie        Japon        Libye   Luxembourg    Nicaragua      Norvege 
           1            1            1            2            1            2            1            2            1            2 
Nile-Zelande       Panama     Pays-Bas         P閥u  Philippines      Pologne          RFA  Royaume-Uni     Salvador  Sud-Vietnam 
           2            1            2            1            1            2            2            2            1            1 
       Suede       Suisse       Taiwan      Uruguay    Venezuela  Yougoslavie 
           2            2            2            1            1            2 
```

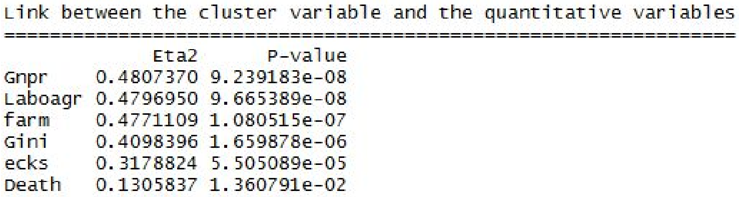
In the figure above we can see the implementation of the consolidation process, where the final assignment of clusters of each individual is also shown.



```
> calinHaraBef
[1] 31.03209
> calinHaraAft
[1] 31.83328
```

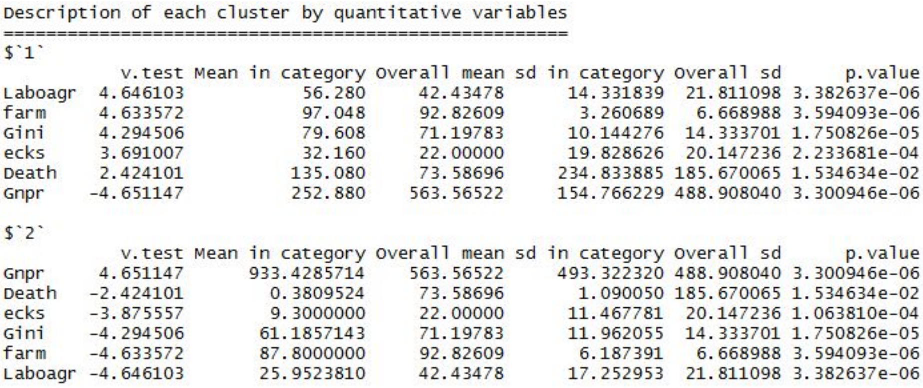
In the figure above we can see that after performing a consolidation process the Calinski-Harabasz index is higher, this means that the consolidation process has made better the decision of clusters of the data.

4. Function catdes and representation of the first factorial display (Exercise 6, 7)



```
Link between the cluster variable and the quantitative variables
================================================================
             Eta2      P-value
Gnpr    0.4807370 9.239183e-08
Laboagr 0.4796950 9.665389e-08
farm    0.4771109 1.080515e-07
Gini    0.4098396 1.659878e-06
ecks    0.3178824 5.505089e-05
Death   0.1305837 1.360791e-02
```

In the figure above we can see the correlation coefficient and p-values. We can observe that the variables are ordered in ascending order of their correlation value and in descending order of p-value. We can conclude that Gnpr, Laboagr, famr and Gini are the most correlated variables respect to the cluster variable, this means that they determine mainly which cluster an individual belongs to. By the other hand, we can say that the death variable is one of the variables that determine less the cluster of an individual, together with other variables that do not appear in the list (because they have even smaller correlation value).



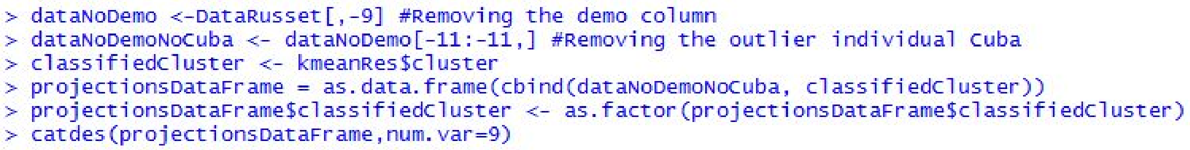
```
Description of each cluster by quantitative variables
=====================================================
$`1`
           v.test Mean in category Overall mean sd in category Overall sd      p.value
Laboagr  4.646103           56.280     42.43478      14.331839  21.811098 3.382637e-06
farm     4.633572           97.048     92.82609       3.260689   6.668988 3.594093e-06
Gini     4.294506           79.608     71.19783      10.144276  14.333701 1.750826e-05
ecks     3.691007           32.160     22.00000      19.828626  20.147236 2.233681e-04
Death    2.424101          135.080     73.58696     234.833885 185.670065 1.534634e-02
Gnpr    -4.651147          252.880    563.56522     154.766229 488.908040 3.300946e-06

$`2`
           v.test Mean in category Overall mean sd in category Overall sd      p.value
Gnpr     4.651147      933.4285714    563.56522     493.322320 488.908040 3.300946e-06
Death   -2.424101        0.3809524     73.58696       1.090050 185.670065 1.534634e-02
ecks    -3.875557        9.3000000     22.00000      11.467781  20.147236 1.063810e-04
Gini    -4.294506       61.1857143     71.19783      11.962055  14.333701 1.750826e-05
farm    -4.633572       87.8000000     92.82609       6.187391   6.668988 3.594093e-06
Laboagr -4.646103       25.9523810     42.43478      17.252953  21.811098 3.382637e-06
```

In the figure above we can see a description about the association between one categorical variable and numerical variables. In our case, we want to see the associations between the categorical variable that is the variable demo with the projections of individuals in the significant dimensions. In other words, how the dimensions determine in the decision of the cluster that an instance belongs to.

The function catdes is used to describe the clusters and help us in the interpretation in terms of the variables. Concretely, a categorical description based on the information of the clusters are performed.

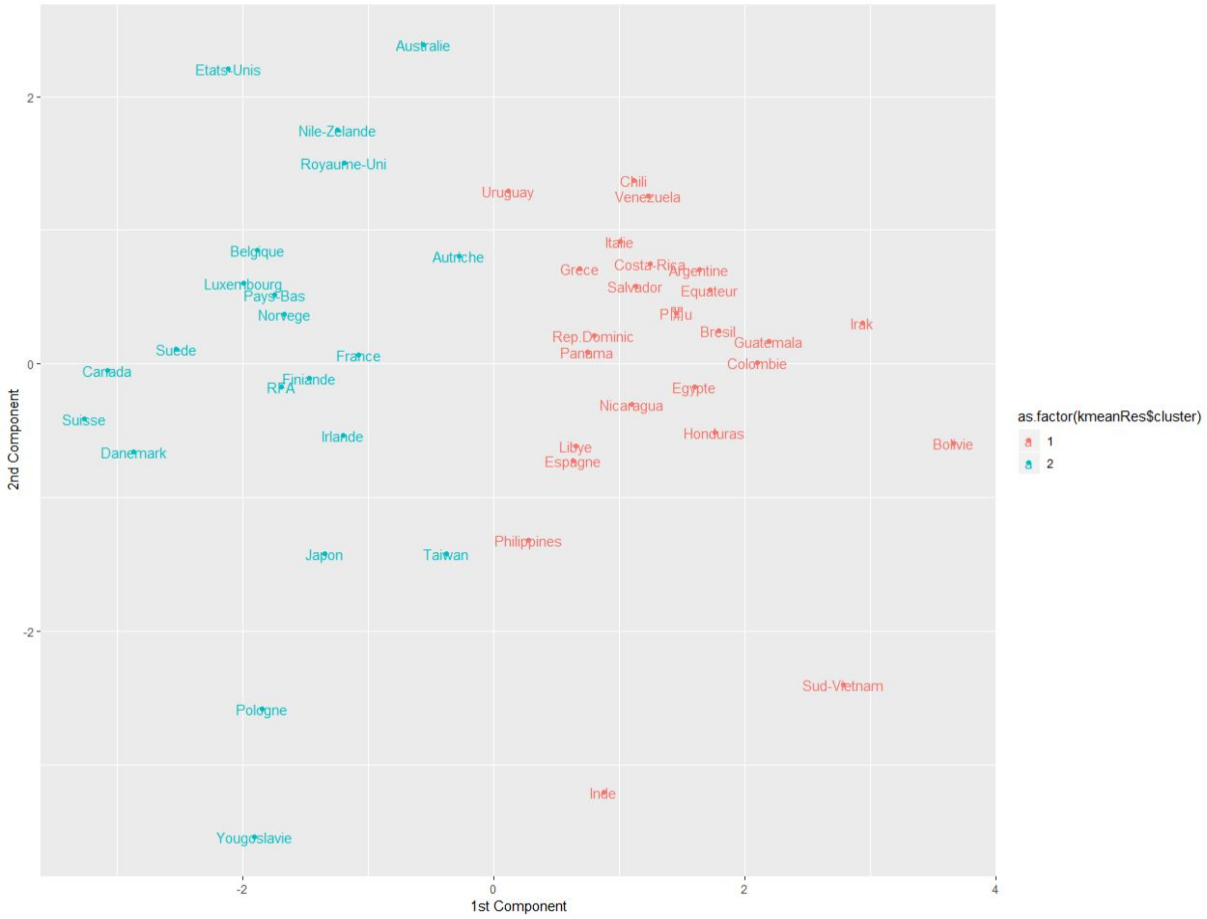
Analyzing the results, it indicates us that the individuals in cluster 1 are mainly determined by the % of farmers with 50% of land and the % of active population in agriculture, because Laboagr and Farm variables have high values of v.test (that indicates the standard deviation below or above the overall mean values). By the other hand, the individuals in cluster 2 are mainly determined by their GNP per capita.



```
> dataNoDemo <-DataRusset[,-9] #Removing the demo column
> dataNoDemoNoCuba <- dataNoDemo[-11:-11,] #Removing the outlier individual Cuba
> classifiedCluster <- kmeanRes$cluster
> projectionsDataFrame = as.data.frame(cbind(dataNoDemoNoCuba, classifiedCluster))
> projectionsDataFrame$classifiedCluster <- as.factor(projectionsDataFrame$classifiedCluster)
> catdes(projectionsDataFrame,num.var=9)
```

In the figure above, we can see the code of the catdes function.

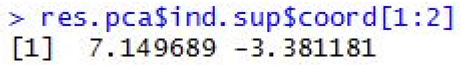
- <u>Representation of the clusters obtained</u>



In the figure above we can see the representation of the different clusters in different colors and painting the instances in the color of the cluster they belong to.

We can observe that our clusters are separated clearly. At the left hand side marked with a blue color we can observe that the countries have more stability politically, while at the right hand side marked with a red color we can observe countries with greater instability.

- <u>Plausible profile for Cuba</u>



```
> res.pca$ind.sup$coord[1:2]
[1]  7.149689 -3.381181
```

In the figure above we can see the the coordinates of the supplementary individual, with this we can observe where Cuba is projected.

The most plausible profile for cuba should be cluster 2. Because the coordinates of Cuba is located at the right side. Which make sense because Cuba has been an politically inestable country during 1969.

5. Annex: R script

```
################################
# Henry Qiu and Goktug Cengiz #
################################

library(FactoMineR)
library(missForest)
library(mice)
library(fpc)
library(ggplot2)
library(gridExtra)

#Reading the data
DataRusset = read.delim("C:/Users/Henry/Desktop/MASTER/2n
Semester/MVA/Homework/H3/Russet_ineqdata.txt", header = TRUE, sep="\t", dec =".")
#Data Understanding
str(DataRusset)
summary(DataRusset)
apply(DataRusset, 2, function(x) sum(is.na(x))) #Missing Value Detection
md.pattern(DataRusset) #Missing Value Visualization
#Data Preprocessing
#Missing Value Imputation with RandomForest
DR = missForest(DataRusset)
DR = DR$ximp
DR$ecks = round(DR$ecks)
DR$Rent = round(DR$Rent)
sum(is.na(DR))

#1.
#Perform the Principal Components Analysis.
#Take the democracy index as supplementary variable,
#whereas the remaining ones are active
#and CUBA as supplementary individual
res.pca = PCA(DR, quali.sup = 9, ind.sup = 11, scale.unit = TRUE, graph = TRUE)
print(res.pca)
eigenvalues = res.pca$eig

#2.
#Interpret the first two obtained factors.
summary(res.pca, ncp = 2)
dimdesc(res.pca, axes = 1:2) #correlation between each variable and the principal
component of rank s is calculated. correlation coefficients are sorted and significant ones are
output
head(eigenvalues[, 1:2])
plot(res.pca, cex = 0.8)
barplot(eigenvalues[,1],main="Eigenvalues",names.arg=1:nrow(eigenvalues))

#3.
#Decide the number of significant dimensions that you retain
#(by subtracting the average eigenvalue
# and represent the new obtained eigenvalues in a new screeplot).
newEigenvalues = eigenvalues[ ,1] - mean(eigenvalues[,1])
barplot(newEigenvalues, names.arg=1:length(newEigenvalues),
        main = "Scree Plot",
        xlab = "Principal Components",
        ylab = "Percentage of variances",
        col ="steelblue")
# Add connected line segments to the plot
lines(x = seq(0.7, 9.5, 1.2), newEigenvalues,
      type="b", pch=19, col = "red")

sigDim = eigenvalues[ ,1][newEigenvalues > 0]
plot(sigDim, type="b", main="Eigenvalues")
numSigDim = length(sigDim)

#4.
#Perform a hierarchical clustering with the significant factors,
#decide the number of final classes to obtain and
#perform a consolidation operation of the clustering.
projections = res.pca$ind$coord[,1:numSigDim]
distMatrix = dist(projections)

hierClustSingl = hclust(distMatrix,method = "single")
plot(hierClustSingl, main= "Single Linkage")

hierClustComp = hclust(distMatrix,method = "complete")
plot(hierClustComp, main= "Complete Linkage")

hierClustAver = hclust(distMatrix,method = "average")
plot(hierClustAver, main= "Average Linkage")

hierClustCentr = hclust(distMatrix,method = "centroid")
plot(hierClustCentr, main= "Centroid Linkage")

hierClustMed = hclust(distMatrix,method = "median")
plot(hierClustMed, main= "Median Linkage")

hierClustWard = hclust(distMatrix,method = "ward.D2")
plot(hierClustWard, main= "Ward's Method")

#Choose Ward's Method
resCut2 <- cutree(hierClustWard, k = 2)
resCut3 <- cutree(hierClustWard, k = 3)
resCut4 <- cutree(hierClustWard, k = 4)
resCut5 <- cutree(hierClustWard, k = 5)
resCut6 <- cutree(hierClustWard, k = 6)
resCut7 <- cutree(hierClustWard, k = 7)
resCut8 <- cutree(hierClustWard, k = 8)
calinHara2 <- calinhara(projections,resCut2)
calinHara3 <- calinhara(projections,resCut3)
calinHara4 <- calinhara(projections,resCut4)
```

```r
calinHara5 <- calinhara(projections,resCut5)
calinHara6 <- calinhara(projections,resCut6)
calinHara7 <- calinhara(projections,resCut7)
calinHara8 <- calinhara(projections,resCut8)

numClusters = 4
plot(hierClustWard, main= "Ward's Method")
abline(h = 6, col = 'red')

centroids <- NULL
for(k in 1:numClusters){
  prueba <- projections[resCut4 == k, , drop = FALSE]
  centroidPerCluster <- colMeans(prueba)
  centroids <- rbind(centroids,centroidPerCluster)
}
print(centroids)

kmeanRes <- kmeans(projections, centroids)
print("K-means Clustering after consolidation")
print(kmeanRes$cluster)

#5.
#Compute the Calinski-Harabassz index and
#compare before and after the consolidation step.
calinHaraBef <- calinhara(projections,resCut6)
calinHaraAft <- calinhara(projections,kmeanRes$cluster)

#6.
#Using the function catdes interpret and name the obtained clusters and
#represent them in the first factorial display.

demoRow <-DataRusset$demo
demoRow <- demoRow[-11:-11]#Removing the instance of the outlier Cuba
demoColumn  <- matrix(demoRow,ncol=1)

projectionsDataFrame = as.data.frame(cbind(projections, demoColumn))
projectionsDataFrame$V4 <- as.factor(projectionsDataFrame$V4)
catdes(projectionsDataFrame,num.var=4)

plot(x = projections[,1], y = projections[,2])

g <- ggplot(as.data.frame(projections), aes(x=projections[,1], y=projections[,2],
            color = as.factor(kmeanRes$cluster))) +  geom_point() +
            labs(x = '1st Component', y = '2nd Component') + geom_text(aes(label=labls))
grid.arrange(g, ncol = 1)


labls = row.names(projections)
text(as.data.frame(projections),labels = labls)
```