

Unsupervised classification

Andrea Huerfano

September 8, 2019

```
packages<-c("FactoMineR", "factoextra", "xtable", "NbClust", "plyr", "GGally", "ggcorrplot")
lapply(packages, library, character.only=TRUE)
```

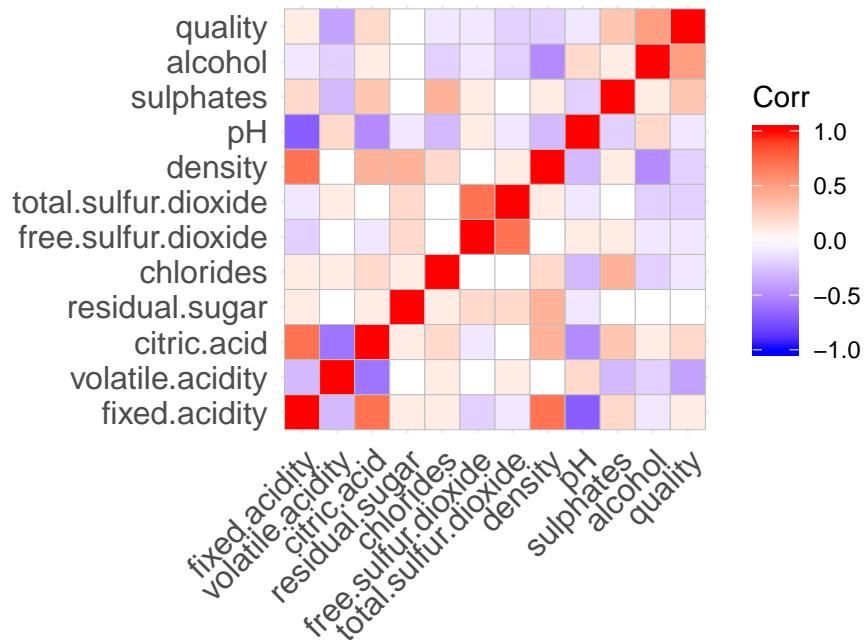
This dataset is a subset of the results of a chemical analysis of Italian wine. The analysis determined the quantities of 13 constituents of red wines. These are the variables:

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free.sulfur.dioxide
- total.sulfur.dioxide
- density
- ph
- sulphates
- alcohol
- quality

```
winequality.red <- read.csv("C:/Users/Andrea/Desktop/wine/winequality-red.csv", sep=";")
```

Looking into the data structure we can see a strong direct/inverse relation between some variables like fixed-volatile acidity, free-total sulfur dioxide and alcohol-quality.

```
corr <- round(cor(winequality.red), 1)
ggcorrplot(corr)
```



Kmeans

Number of cluster

The data is standardized before clustering to obtain a better quality, efficient and accurate cluster result. The library *NbClust* is used to calculate the optimum number of clusters, we are going to use the Friedman index.

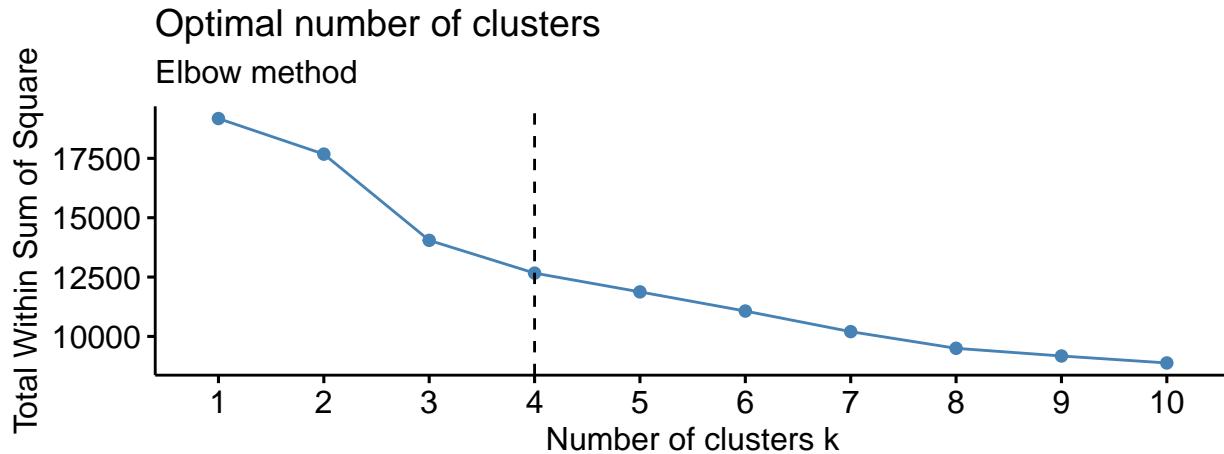
```
df<-scale(winequality.red)
res.wine<-NbClust(df, distance = "canberra", min.nc = 2, max.nc = 10, method = "kmeans", index = "friedman")
res.wine$Best.nc
```



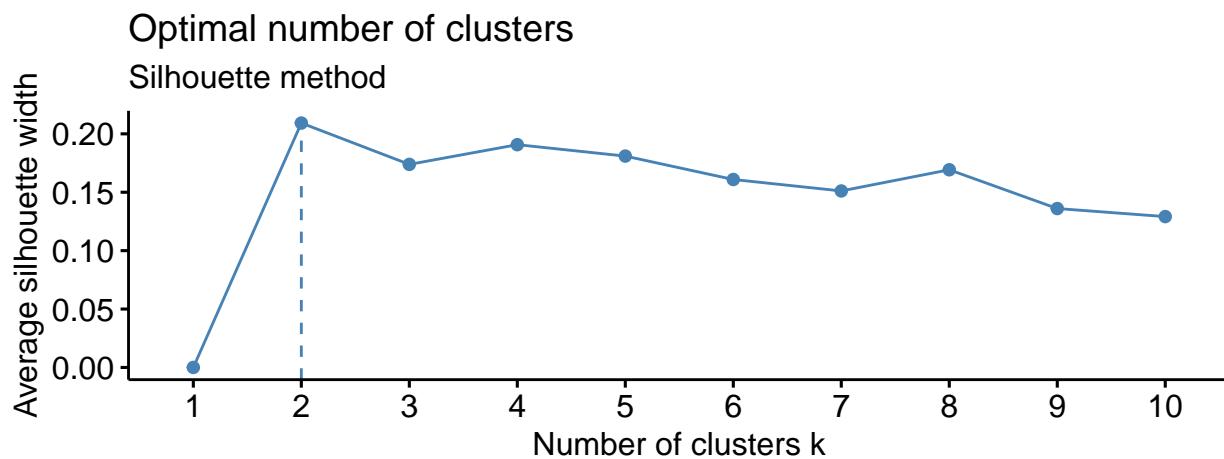
```
## Number_clusters      Value_Index
##             4.0000          3.7614
```

Another methods to select the number of cluster are plotted below.

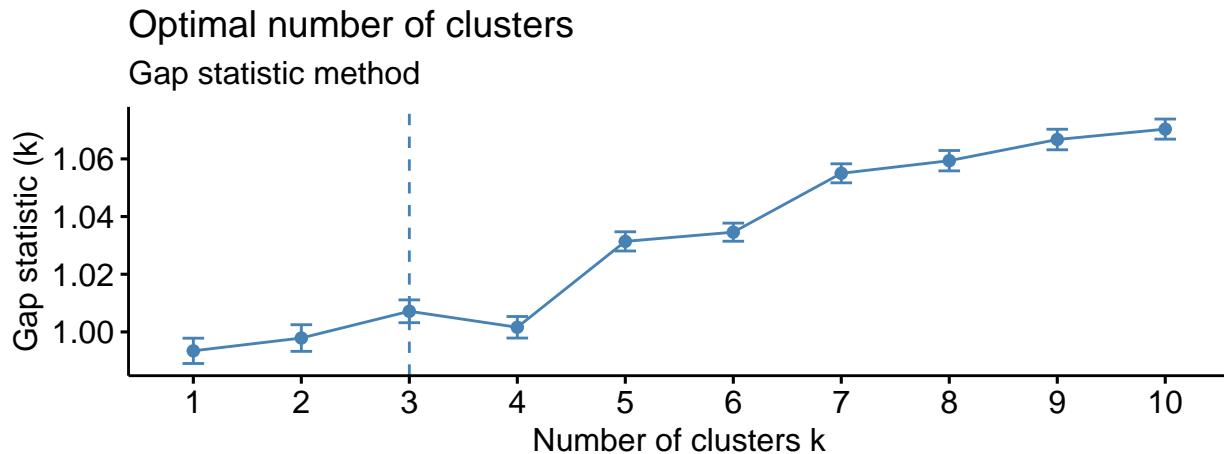
```
set.seed(21)
elbow<-fviz_nbclust(df, kmeans, method = "wss") +geom_vline(xintercept = 4, linetype = 2)+ labs(subtitle = "Elbow method")
```



```
Silhouette<-fviz_nbclust(df, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method");Silhouette
```



```
gap<-fviz_nbclust(df, kmeans, nstart = 25, method = "gap_stat",
  nboot = 50)+labs(subtitle = "Gap statistic method"); gap
```



After all of the results, I decided to use the mode and in this way, obtaining that the better number of clusters is 4.

Kmeans algorithm

No forget to set a seed to obtain every run the same results.

```
set.seed(10)
n_cluster<-4
kmedia2<-kmeans(df, n_cluster, nstart = 1000)
```

Let see the centroids

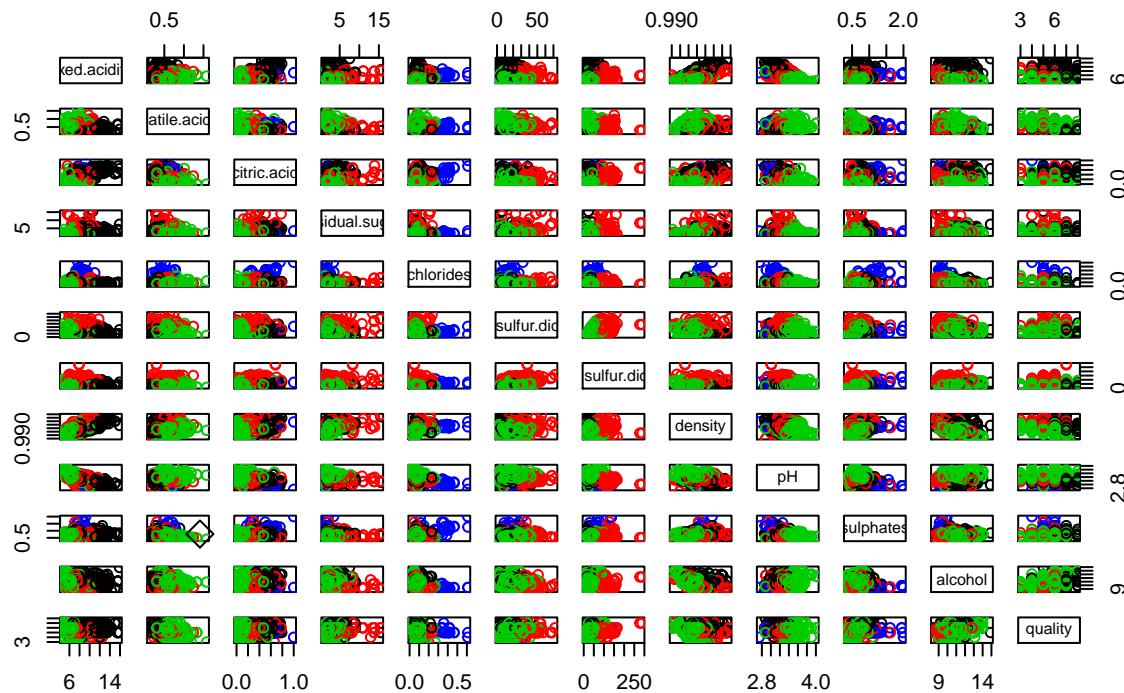
```
round(kmedia2$centers,2)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1      0.99        -0.78     1.01      0.06     -0.09
## 2     -0.05        0.03     0.12      0.37      0.00
## 3     -0.65        0.51     -0.80     -0.23     -0.17
## 4      0.08        0.02     1.14     -0.40      5.60
##   free.sulfur.dioxide total.sulfur.dioxide density      pH sulphates alcohol
## 1      -0.46        -0.53     0.36    -0.66      0.38     0.47
## 2       1.00         1.26     0.32    -0.17     -0.20     -0.56
## 3     -0.24        -0.36     -0.43     0.61     -0.30     0.03
## 4     -0.07        0.47     0.19    -1.69      3.72     -0.88
##   quality
## 1      0.63
## 2     -0.45
## 3     -0.16
## 4     -0.36
```

And the clusters over all of the variables are painted in the scatter plots when each color represents a different cluster.

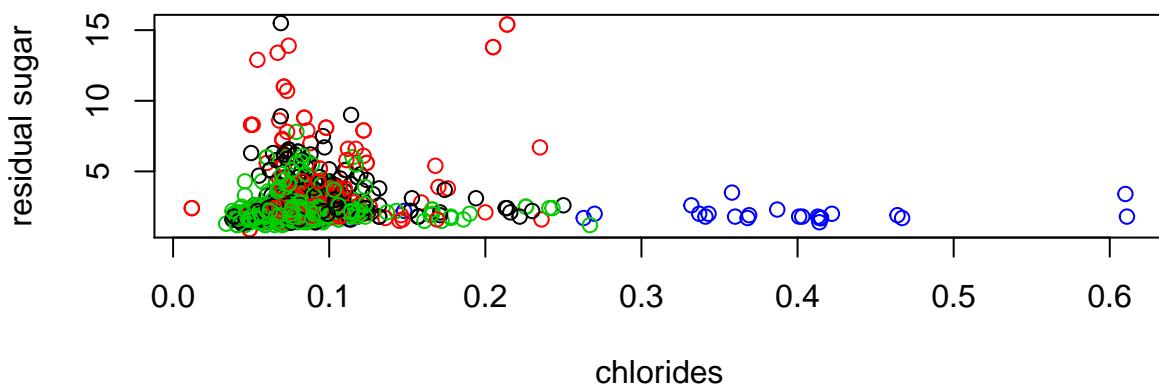
```
plot(winequality.red, col=kmedia2$cluster,
      main="Posición de los grupos respecto a las variables")
points(kmedia2$centers, col = kmedia2$cluster, pch = 23, cex = 1.5)
```

Posición de los grupos respecto a las variables



To see in detail one of the scatterplots

```
plot(winequality.red$chlorides, winequality.red$residual.sugar, col=kmedia2$cluster, xlab = "chlorides"
legend(0,100, legend=c('1', '2', '3', '4'),col=c("black","blue", "red", "green" ),
pch = 18, cex = 1)
```



It is really important look into the withinss and betweens sum of squares.

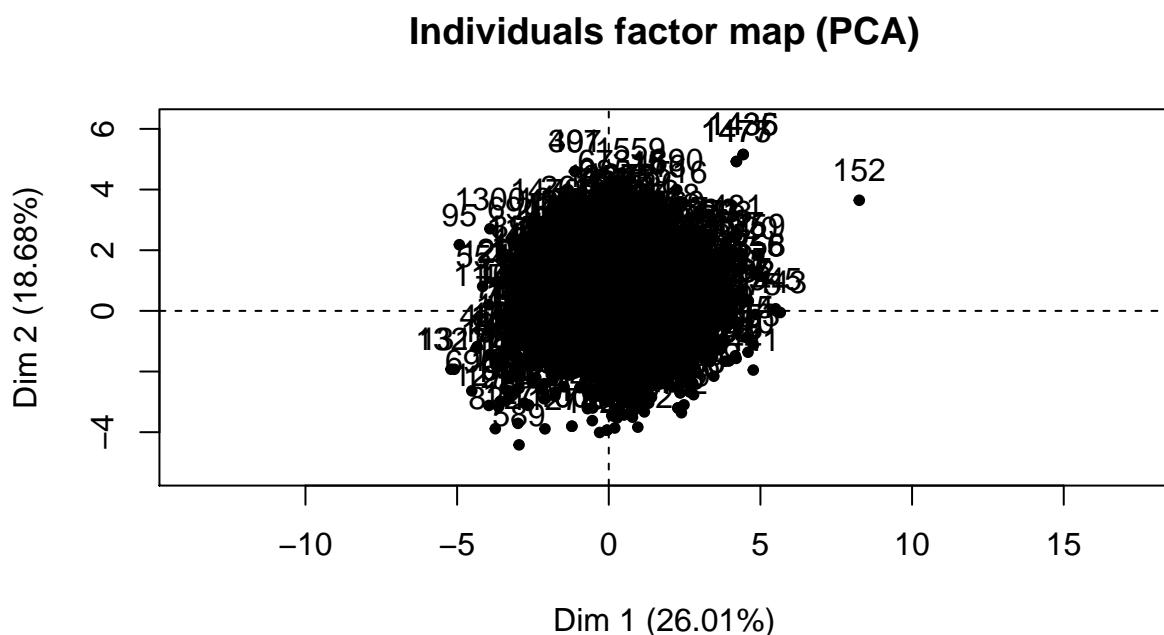
```
print(round(kmedia2$withinss,2))  
## [1] 3749.95 3728.35 4704.79 478.21  
  
print(kmedia2$betweenss)  
## [1] 6514.699
```

In this case, the sum of squares over the between is greater than the within, which means that the segmentation creates clusters different between them but internally the observations are similar, homogeneity inside and heterogeneity outside the clusters.

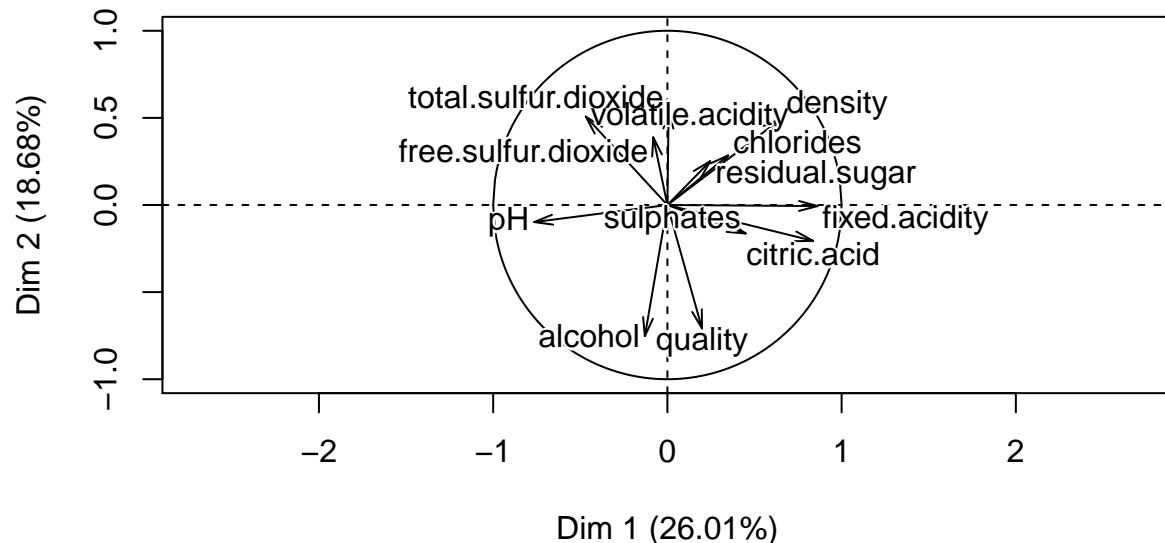
Hierarchichal Clustering

Principal Component Analysis

```
res.PCA<-PCA(df, ncp = 2)
```

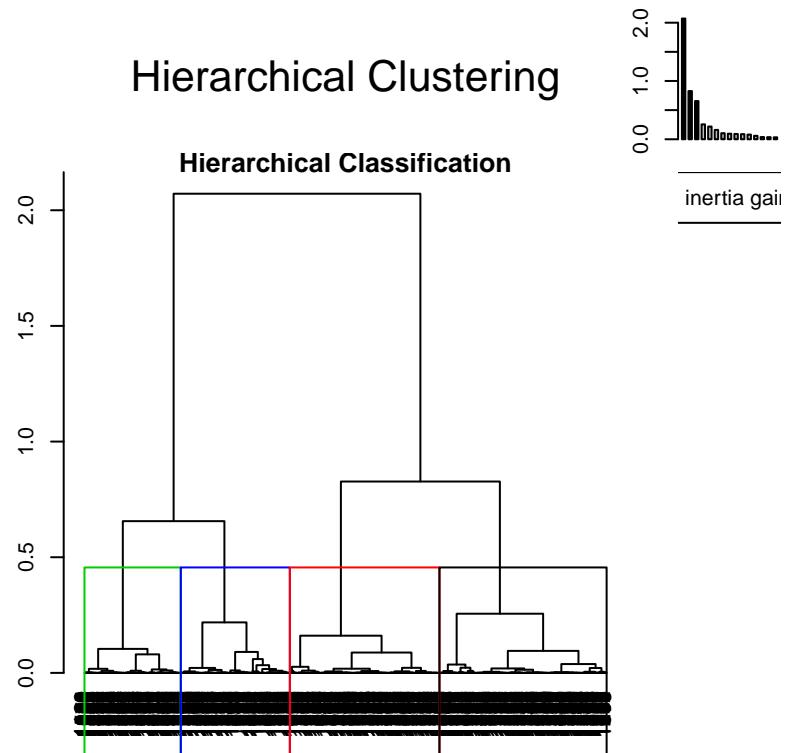


Variables factor map (PCA)

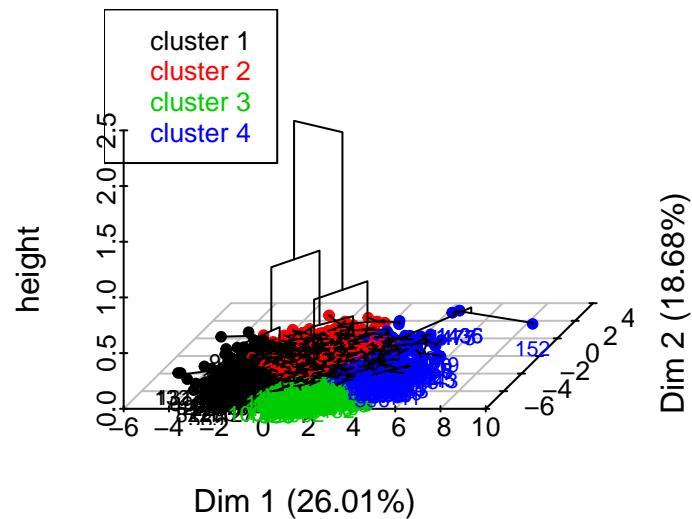


Ahora, utilicemos la función HPC sobre res.PCA

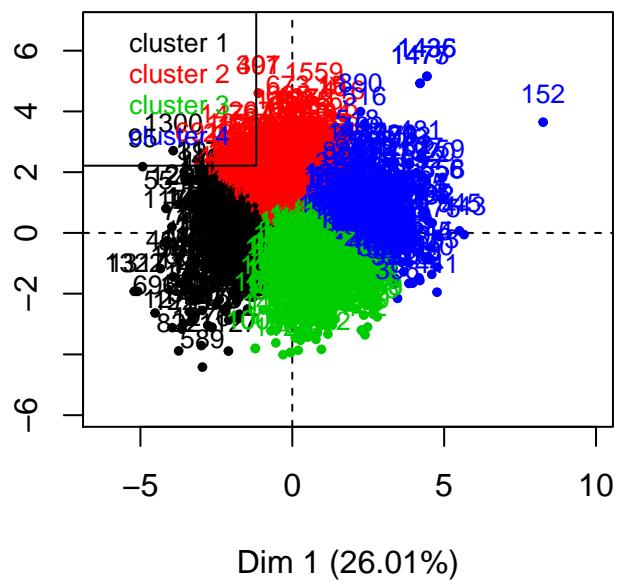
```
set.seed(10)
res.hpc<-HCPC(res.PCA, nb.clust = -1,min = 3, max = 6)
```



Hierarchical clustering on the factor map



Factor map



```
#gmfd_kmeans(df, n.cl = 4, metric, p = NULL, k_trunc = NULL)
```

References

- Sebastien Le, Julie Josse, Francois Husson (2008). FactoMineR: An R Package for Multivariate Analysis. *Journal of Statistical Software*, 25(1), 1-18. 10.18637/jss.v025.i01
- Alboukadel Kassambara and Fabian Mundt (2017). factoextra: Extract and Visualize the Results of Multivariate Data Analyses. R package version 1.0.5. <https://CRAN.R-project.org/package=factoextra>
- David B. Dahl, David Scott, Charles Roosen, Arni Magnusson and Jonathan Swinton (2019). xtable: Export Tables to LaTeX or HTML. R package version 1.8-4. <https://CRAN.R-project.org/package=xtable>
- Malika Charrad, Nadia Ghazzali, Veronique Boiteau, Azam Niknafs (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *Journal of Statistical Software*, 61(6), 1-36. URL <http://www.jstatsoft.org/v61/i06/>.
- Hadley Wickham (2011). The Split-Apply-Combine Strategy for Data Analysis. *Journal of Statistical Software*, 40(1), 1-29. URL <http://www.jstatsoft.org/v40/i01/>.
- Barret Schloerke, Jason Crowley, Di Cook, Francois Briatte, Moritz Marbach, Edwin Thoen, Amos Elberg and Joseph Larmarange (2018). GGally: Extension to ‘ggplot2’. R package version 1.4.0. <https://CRAN.R-project.org/package=GGally>
- Alboukadel Kassambara (2019). ggcormrplot: Visualization of a Correlation Matrix using ‘ggplot2’. R package version 0.1.3. <https://CRAN.R-project.org/package=ggcormrplot>