PROJECT 1: UNSUPERVISED LEARNING

Statistical Learning. Bachelor in Data Science and Engineering

Ángela María Durán Pinto

05/11/2023

1 Main Objective

Have you ever heard: “Money doesn’t give happiness”? Well, during this project we are going to answer that question, and we are going to identify groups of countries exhibit similar characteristics or trends in their happiness scores and the associated factors.This can be valuable for policy analysis, international comparisons, and understanding the drivers of well-being in different regions.

2 Analyze Dataset

The first thing that we have to do before starting the project, is understand our variables for future decisions.

Our dataset is “The World Happiness Report”, which contains a variety of data related to happiness scores, as well as factors that contribute to overall well-being in different countries.

Let’s define our variables:

- Ladder score: Happiness score or subjective well-being. National average response to the question of life evaluations.

-Standard error: indicates the level of uncertainty or margin of error associated with the reported “Ladder score.”

- GDP: Gross Domestic Product

-Healthy Life Expectancy (HLE). They are based on the data extracted from the World Health Organization’s (WHO) Global Health Observatory data repository

-Social support (or having someone to count on in times of trouble) is the national average of the binary responses (either 0 or 1) to the GWP question “If you were in trouble, do you have relatives or friends you can count on to help you whenever you need them, or not?” -Freedom to make life choices is the national average of responses to the GWP question

-Generosity: the residual of regressing national average of response to the GWP question “Have you donated money to a charity in the past month?” on GDP per capita.

-Corruption Perception: The measure is the national average of the survey responses to two questions

-Upperwhisker: Positive affect defined as the average of three positive affect measures in GWP: laugh, enjoyment and doing interesting things

-Lowerwhisker: Negative affect defined as the average of three negative affect measures in GWP. They are worry, sadness and anger.

-Positive affect is defined as the average of three positive affect measures in GWP: laugh, enjoyment and doing interesting things in the Gallup World Poll waves 3-7.

-Negative affect is defined as the average of three negative affect measures in GWP. They are worry, sadness and anger,

Load necessary libraries

```
#install.packages("readxl")
library(readxl)

rm(list=ls())

library(tidyverse)
library(countrycode)
library(rworldmap)
library(ggplot2)

# Dealing with missing values
library(mice)

#GGally: collection of functions for creating plots and visualizations to explore and understa
nd relationships between variables in your data. : heatmap-like graphs, and ggcorr

library(GGally)
# Visualisation and Imputation of Missing Data

library(VIM)
# extracting and visualizing information from multivariate data analyses, particularly factor
analysis and PCA)
library(factoextra)
```

```r
# for clustering
library(cluster)
library(mclust)

# kernel k-means
library(kernlab)
```

Load dataset

```r
data <- read_excel("World_Happiness.xls")

# Let's change the names of the columns as they are long and with spaces
new_column_names <- c("Country", "Score", "StandardError", "Upperwhisker", "Lowerwhisker", "Lo
gGDP", "SocialSupport", "HLE", "Freedom", "Generosity", "PerceptionsCorruption","ScoreDystopia
", "ExLogGDP", "ExSocialSupport", "ExHLE", "ExFreedom", "ExGenerosity", "ExPerceptionsCorrupti
on", "DystopiaResidual"   )

colnames(data) <- new_column_names

summary(data)
```

```
##    Country              Score       StandardError      Upperwhisker
##  Length:139         Min.   :1.859   Min.   :0.01350   Min.   :1.923
##  Class :character   1st Qu.:4.790   1st Qu.:0.04651   1st Qu.:4.982
##  Mode  :character   Median :5.684   Median :0.06013   Median :5.851
##                     Mean   :5.543   Mean   :0.06444   Mean   :5.678
##                     3rd Qu.:6.332   3rd Qu.:0.07727   3rd Qu.:6.463
##                     Max.   :7.804   Max.   :0.14654   Max.   :7.875
##
##   Lowerwhisker       LogGDP        SocialSupport         HLE
##  Min.   :1.795   Min.   : 5.527   Min.   :0.3413   Min.   :51.53
##  1st Qu.:4.611   1st Qu.: 8.593   1st Qu.:0.7241   1st Qu.:60.72
##  Median :5.572   Median : 9.582   Median :0.8350   Median :65.88
##  Mean   :5.421   Mean   : 9.463   Mean   :0.8003   Mean   :65.04
##  3rd Qu.:6.231   3rd Qu.:10.497   3rd Qu.:0.8961   3rd Qu.:69.54
##  Max.   :7.733   Max.   :11.660   Max.   :0.9825   Max.   :77.28
##                                                    NA's   :1
##     Freedom         Generosity         PerceptionsCorruption ScoreDystopia
##  Min.   :0.3816   Min.   :-0.2542763   Min.   :0.1461        Min.   :1.778
##  1st Qu.:0.7255   1st Qu.:-0.0785508   1st Qu.:0.6744        1st Qu.:1.778
##  Median :0.8013   Median :-0.0004827   Median :0.7749        Median :1.778
##  Mean   :0.7887   Mean   : 0.0197509   Mean   :0.7276        Mean   :1.778
##  3rd Qu.:0.8756   3rd Qu.: 0.1153806   3rd Qu.:0.8462        3rd Qu.:1.778
##  Max.   :0.9614   Max.   : 0.5313863   Max.   :0.9291        Max.   :1.778
##
##     ExLogGDP     ExSocialSupport      ExHLE          ExFreedom
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.100   1st Qu.:0.9668   1st Qu.:0.2503   1st Qu.:0.4576
##  Median :1.454   Median :1.2469   Median :0.3908   Median :0.5585
##  Mean   :1.412   Mean   :1.1594   Mean   :0.3682   Mean   :0.5417
##  3rd Qu.:1.782   3rd Qu.:1.4014   3rd Qu.:0.4906   3rd Qu.:0.6573
##  Max.   :2.200   Max.   :1.6197   Max.   :0.7016   Max.   :0.7715
##                         NA's   :1
##   ExGenerosity     ExPerceptionsCorruption DystopiaResidual
##  Min.   :0.00000   Min.   :0.00000        Min.   :-0.1098
##  1st Qu.:0.09429   1st Qu.:0.05938        1st Qu.: 1.5563
##  Median :0.13617   Median :0.11044        Median : 1.8483
##  Mean   :0.14703   Mean   :0.14434        Mean   : 1.7774
##  3rd Qu.:0.19834   3rd Qu.:0.18242        3rd Qu.: 2.0758
##  Max.   :0.42155   Max.   :0.56082        Max.   : 2.9546
##                                           NA's   :1
```

```
head(data)
```

```
## # A tibble: 6 × 19
##   Country     Score StandardError Upperwhisker Lowerwhisker LogGDP SocialSupport
##   <chr>       <dbl>         <dbl>        <dbl>        <dbl>  <dbl>         <dbl>
## 1 Finland      7.80        0.0362         7.88         7.73   10.8         0.969
## 2 Denmark      7.59        0.0410         7.67         7.51   11.0         0.954
## 3 Iceland      7.53        0.0486         7.62         7.43   10.9         0.983
## 4 Israel       7.47        0.0316         7.53         7.41   10.6         0.943
## 5 Netherlands  7.40        0.0293         7.46         7.35   10.9         0.930
## 6 Sweden       7.40        0.0374         7.47         7.32   10.9         0.939
## # ℹ 12 more variables: HLE <dbl>, Freedom <dbl>, Generosity <dbl>,
## #   PerceptionsCorruption <dbl>, ScoreDystopia <dbl>, ExLogGDP <dbl>,
## #   ExSocialSupport <dbl>, ExHLE <dbl>, ExFreedom <dbl>, ExGenerosity <dbl>,
## #   ExPerceptionsCorruption <dbl>, DystopiaResidual <dbl>
```

```
dim(data)
```

```
## [1] 139  19
```

```
str(data)
```

```
## tibble [139 × 19] (S3: tbl_df/tbl/data.frame)
##  $ Country                : chr [1:139] "Finland" "Denmark" "Iceland" "Israel" ...
##  $ Score                  : num [1:139] 7.8 7.59 7.53 7.47 7.4 ...
##  $ StandardError          : num [1:139] 0.0362 0.041 0.0486 0.0316 0.0293 ...
##  $ Upperwhisker           : num [1:139] 7.88 7.67 7.62 7.53 7.46 ...
##  $ Lowerwhisker           : num [1:139] 7.73 7.51 7.43 7.41 7.35 ...
##  $ LogGDP                 : num [1:139] 10.8 11 10.9 10.6 10.9 ...
##  $ SocialSupport          : num [1:139] 0.969 0.954 0.983 0.943 0.93 ...
##  $ HLE                    : num [1:139] 71.1 71.3 72.1 72.7 71.6 ...
##  $ Freedom                : num [1:139] 0.961 0.934 0.936 0.809 0.887 ...
##  $ Generosity             : num [1:139] -0.0188 0.1342 0.211 -0.0231 0.2127 ...
##  $ PerceptionsCorruption  : num [1:139] 0.182 0.196 0.668 0.708 0.379 ...
##  $ ScoreDystopia          : num [1:139] 1.78 1.78 1.78 1.78 1.78 ...
##  $ ExLogGDP               : num [1:139] 1.89 1.95 1.93 1.83 1.94 ...
##  $ ExSocialSupport        : num [1:139] 1.58 1.55 1.62 1.52 1.49 ...
##  $ ExHLE                  : num [1:139] 0.535 0.537 0.559 0.577 0.545 ...
## $ ExFreedom             : num [1:139] 0.772 0.734 0.738 0.569 0.672 ...
## $ ExGenerosity          : num [1:139] 0.126 0.208 0.25 0.124 0.251 ...
## $ ExPerceptionsCorruption: num [1:139] 0.535 0.525 0.187 0.158 0.394 ...
## $ DystopiaResidual      : num [1:139] 2.36 2.08 2.25 2.69 2.11 ...
```

```r
tail(data)
```

```
## # A tibble: 6 × 19
##   Country     Score StandardError Upperwhisker Lowerwhisker LogGDP SocialSupport
##   <chr>       <dbl>         <dbl>        <dbl>        <dbl>  <dbl>         <dbl>
## 1 Botswana     3.44        0.136          3.70         3.17   9.63         0.753
## 2 Congo (Kin…  3.21        0.0954         3.39         3.02   7.01         0.652
## 3 Zimbabwe     3.20        0.0609         3.32         3.08   7.64         0.690
## 4 Sierra Leo…  3.14        0.0824         3.30         2.98   7.39         0.555
## 5 Lebanon      2.39        0.0445         2.48         2.30   9.48         0.530
## 6 Afghanistan  1.86        0.0325         1.92         1.80   7.32         0.341
## # ℹ 12 more variables: HLE <dbl>, Freedom <dbl>, Generosity <dbl>,
## #   PerceptionsCorruption <dbl>, ScoreDystopia <dbl>, ExLogGDP <dbl>,
## #   ExSocialSupport <dbl>, ExHLE <dbl>, ExFreedom <dbl>, ExGenerosity <dbl>,
## #   ExPerceptionsCorruption <dbl>, DystopiaResidual <dbl>
```

2.1 Feature Extraction

We want to add to our “data” 2 columns which are goin to be obtained from another source: “additional”. In additional we have data for different years for each country, so we get the vale of PositiveAffect and NegativeAffect only for year 2021. We will get some missing values as there some countries in “data” which are not in “additional”. We will deal later with missing values.

```r
additional <- read_excel("additional.xls")
idx = 1
# Initialize our new columns with NAs
data$PositiveAffect <- NA
data$NegativeAffect <- NA

# for each country
for (i in data$Country) {
  # Compute the index of the row with country name= i and for year 2021
  pos_idx <- which(additional$`Country name` == i & additional$year == 2021)

  # Only if there exists the country i in "additional"
  if (length(pos_idx) > 0) {
    # Assign the Positive Affect value if it's found, else keep it as 0
    data$PositiveAffect[idx] <- additional$`Positive affect`[pos_idx]
    data$NegativeAffect[idx] <- additional$`Negative affect`[pos_idx]
  }
  idx = idx + 1
}
```

2.2 Duplicated Observations

Check duplicated observations (lab clustering class)

```
# 1° search for exactly duplicated rows. We obtain 0. This is usual as 1 one the values can ch
ange, but we still consider it a duplicated row
sum(duplicated(data))
```

```
## [1] 0
```

```
# 2° search for observations with duplicated country, as all the other variables are numeric
sum(duplicated(data$Country))  # Obtain 2 duplicated rows
```

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

```
duplicated_rows <- which(duplicated(data$Country))

# We see that Panama and Portugal have a duplicated observation
data$Country[duplicated(data$Country)]
```

```
## [1] "Panama"   "Portugal"
```

```
# Remove them
data = data[-duplicated_rows,]

# Another way is using dplyr package to keep only the unique rows based on the "Country" colum
n.
data_cleaned <- data %>%
  distinct(Country, .keep_all = TRUE)
```

```
country_names <- data$Country
```

2.3 Feature Selection

It is important to remember that we should only use continuous variables in our study. The reason is that if the different classes of our categorical variables are unbalanced, the one with more observations is going to take too much importance. We are not taking the variable “Country name”. Also, we remove “Standard Error of Ladder Score” as this variable represents the uncertainty associated with the reported “Ladder Score.” It doesn’t provide direct information about the well-being of a country

Explained by Factors (e.g., Log GDP per capita, Social Support, etc.): These variables represent the portion of the “Ladder Score” explained by specific factors. While they are important for understanding the contributions of individual factors to happiness, they are not typically used for clustering because they already represent components of the “Ladder Score.”

Upper Whisker and Lower Whisker: These variables are related to data visualization and do not represent the inherent characteristics of a country’s well-being.

```
data2 =  data %>% dplyr::select(-Country, -StandardError, -Upperwhisker, -Lowerwhisker, -ExLog
GDP, -ExSocialSupport, -ExHLE, -ExFreedom, -ExGenerosity, -ExPerceptionsCorruption,-DystopiaRe
sidual)
```

We have reduced our considerably our dataset variables: from 19 to 10. The other variables were not useful for the main objective of this work.

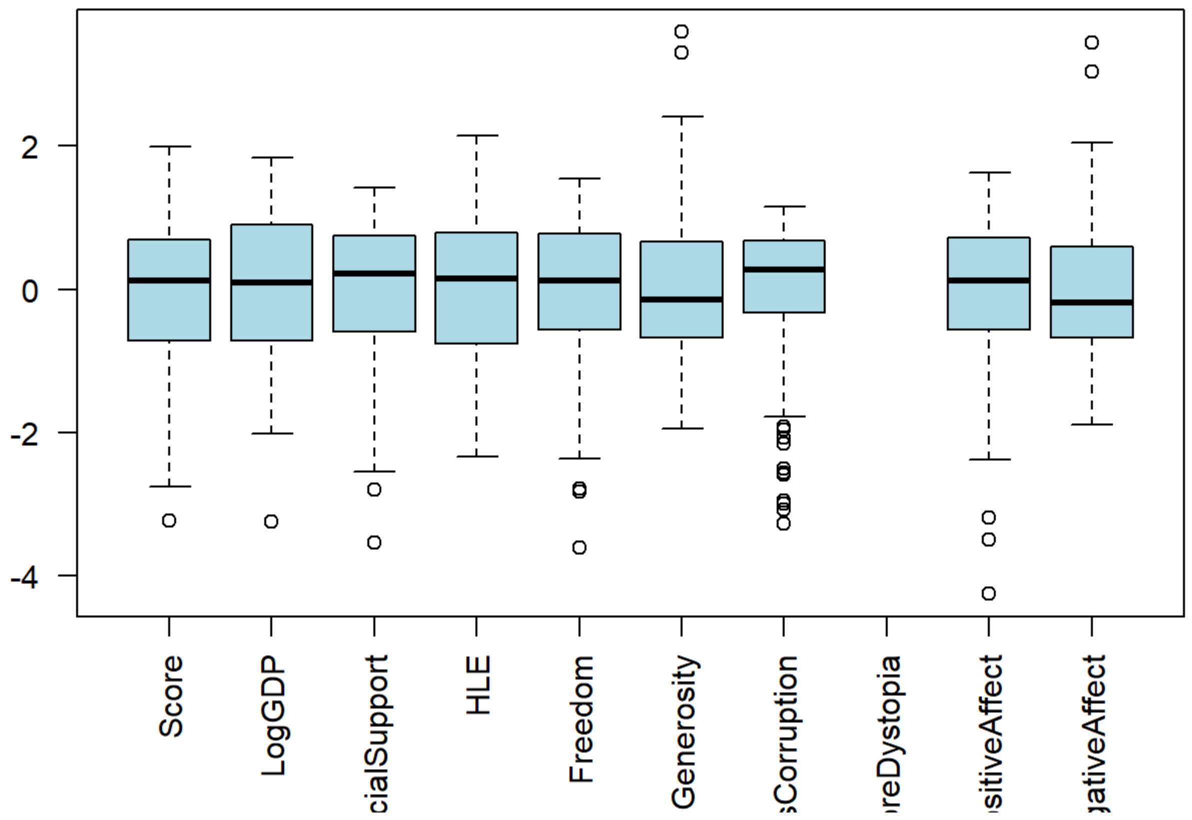
2.4 Outliers

An outlier is an observation or data point that significantly differs from other observations in a dataset. It is an unusual or rare value that falls outside the typical range of values in a dataset.

- 1º We make a general boxplot for the entire dataset

```
# Don't forget to scale the data
boxplot(scale(data2), col = "lightblue", las = 2)
```

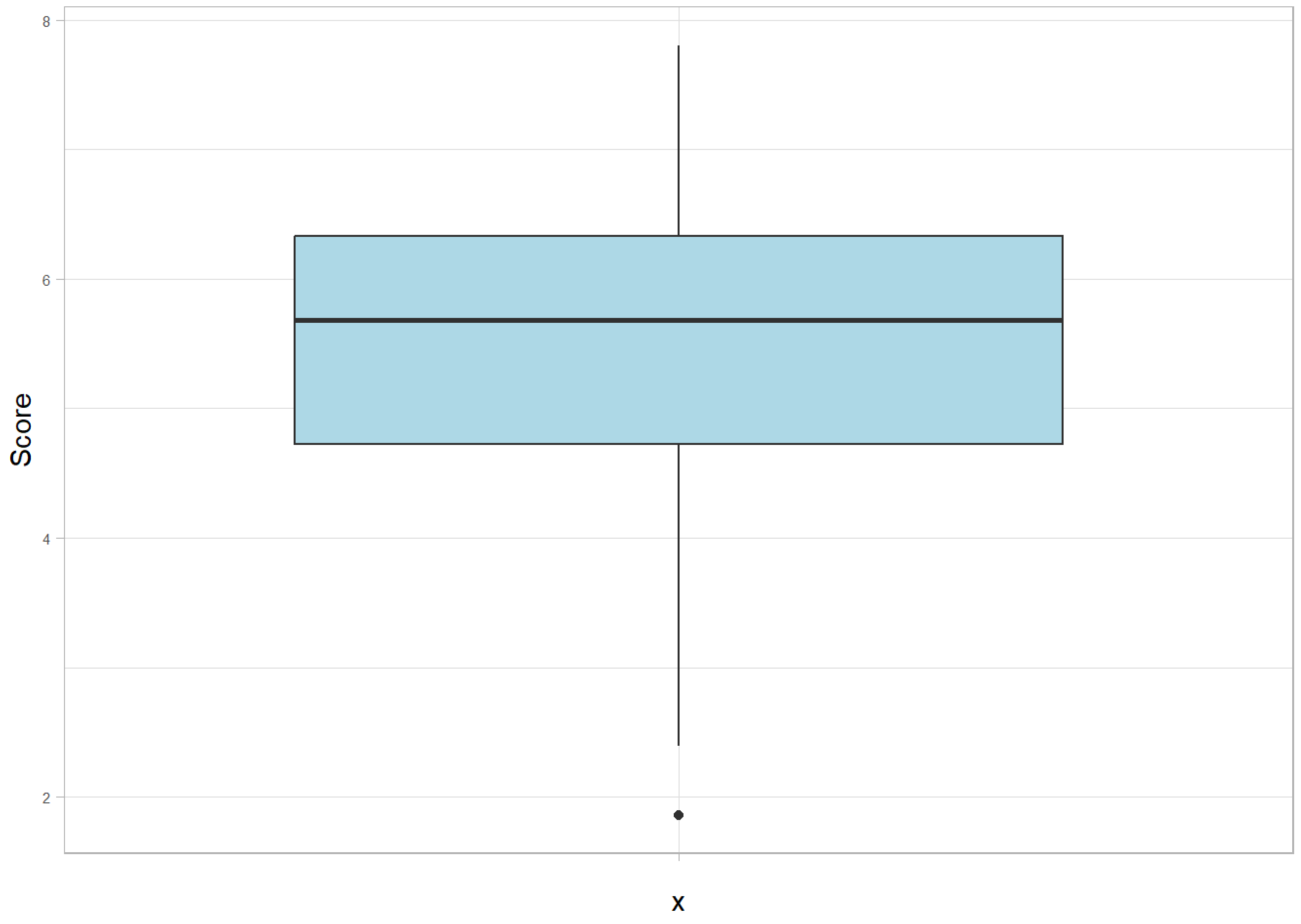
```
## Error : The fig.showtext code chunk option must be TRUE
```



We can see that variables as “Score”, “LogGDP”, “lowerwisker may contain outliers”. Therefore, we have to look more carefully to those variables an make an individual boxplot

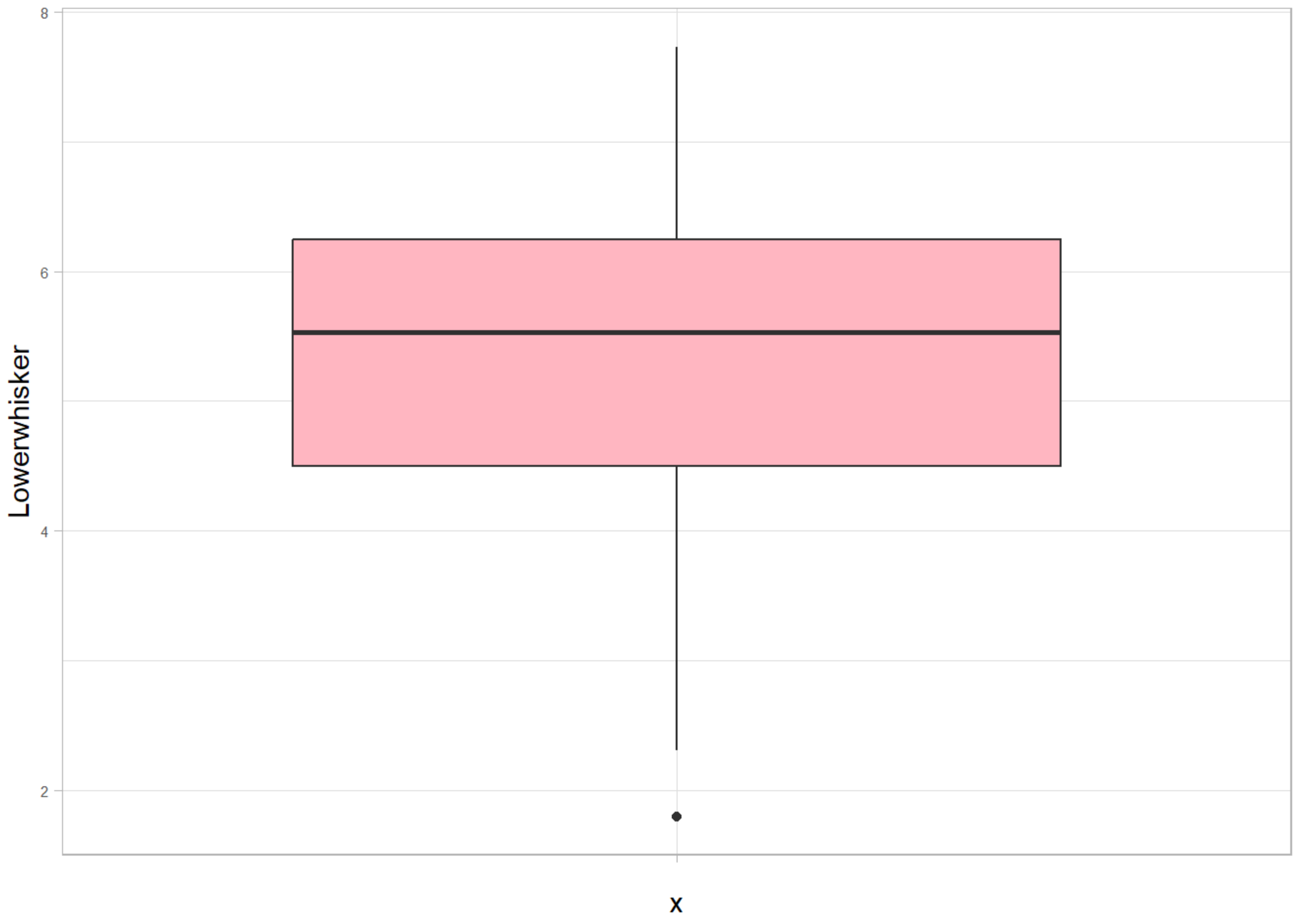
```r
par(mfrow=c(1,3))
# For "Score"
ggplot(data) +
  aes(x = "", y = Score) +
  geom_boxplot(fill = "lightblue") +
  labs(y = "Score") +
  theme_light()
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



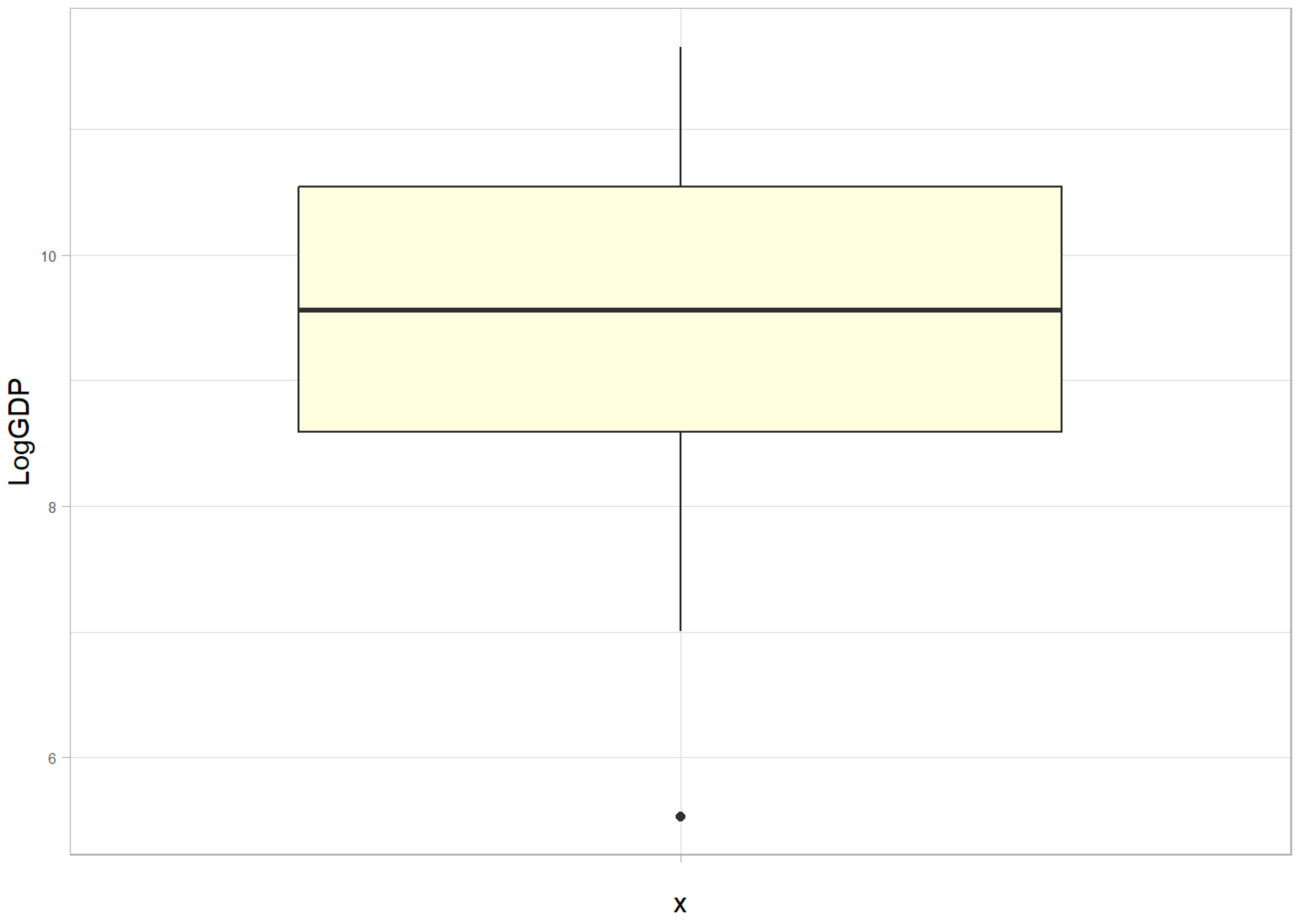
```r
# For "Lowerwhisker"
ggplot(data) +
  aes(x = "", y = Lowerwhisker) +
  geom_boxplot(fill = "lightpink") +
  labs(y = "Lowerwhisker") +
  theme_light()
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



```
# For "LogGDP"

ggplot(data) +
  aes(x = "", y = LogGDP) +
  geom_boxplot(fill = "lightyellow") +
  labs(y = "LogGDP") +
  theme_light()
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



We see that could be an outlier with score <2, Lowershisker > 2, and LogGDP < 5.5

2.5 Missing values

(lab1 data preprocessing)

Missing values, often denoted as NA (Not Available) or NaN (Not-a-Number), are placeholders used in data to represent the absence of a value or an unknown value for a particular observation or variable. They can have a significant impact on data analysis and modeling.

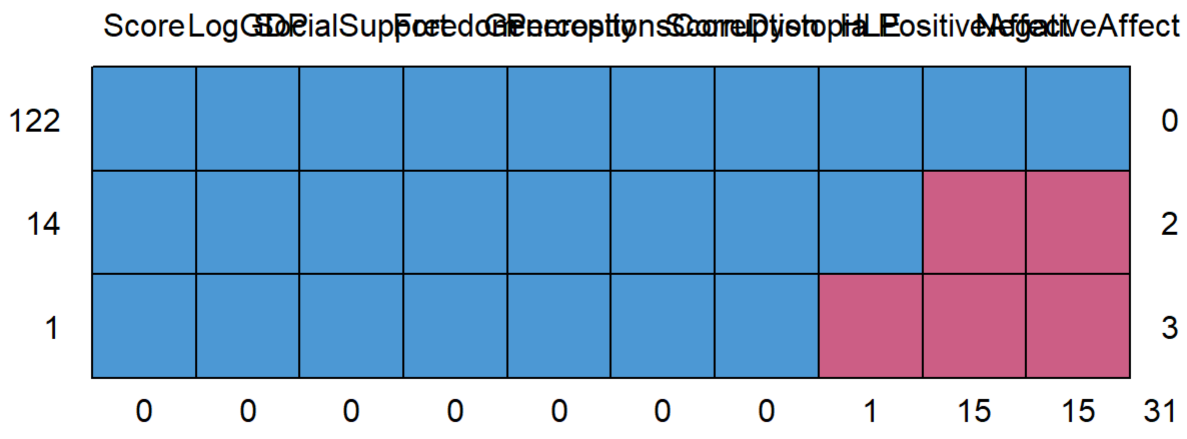
We can see the distribution of the NAs with the mice package

```
sum(is.na(data2)) ## 31 missing values
```

```
## [1] 31
```

```
md.pattern(data2)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



```
##     Score LogGDP SocialSupport Freedom Generosity PerceptionsCorruption
## 122     1      1             1       1          1                     1
## 14      1      1             1       1          1                     1
## 1       1      1             1       1          1                     1
##         0      0             0       0          0                     0
##     ScoreDystopia HLE PositiveAffect NegativeAffect
## 122             1   1              1              1  0
## 14              1   1              0              0  2
## 1               1   0              0              0  3
##                 0   1             15             15 31
```

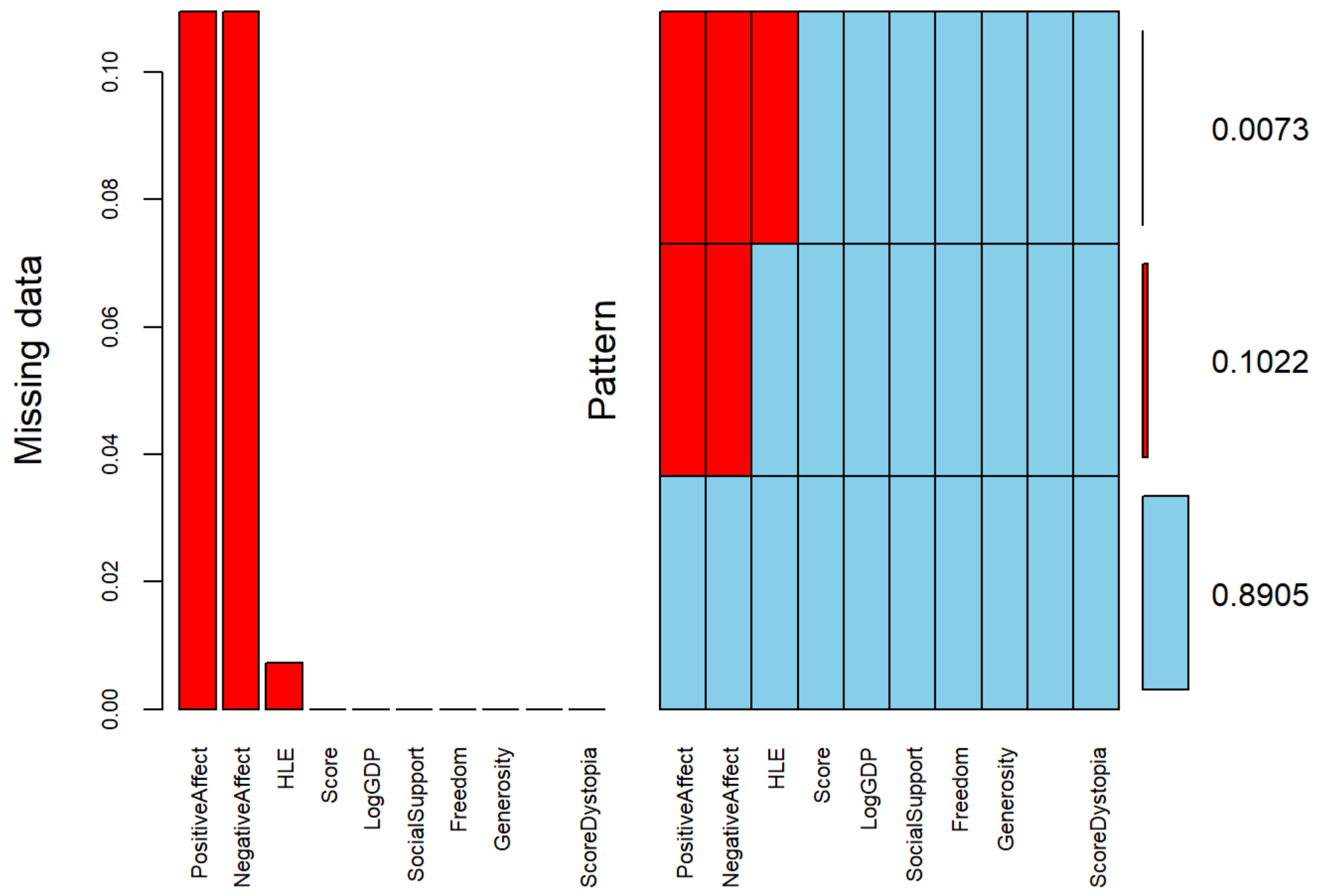
```
aggr(data2, number = TRUE, sortVars = TRUE, labels = names(data2),
     cex.axis = .7, gap = 1, ylab= c('Missing data','Pattern'))
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



```
##
## Variables sorted by number of missings:
##              Variable      Count
##        PositiveAffect 0.10948905
##        NegativeAffect 0.10948905
##                   HLE 0.00729927
##                 Score 0.00000000
##                LogGDP 0.00000000
##         SocialSupport 0.00000000
##               Freedom 0.00000000
##            Generosity 0.00000000
##  PerceptionsCorruption 0.00000000
##         ScoreDystopia 0.00000000
```

We can see that we have 1 observation with 3 missing values and 14 observations with 2 missing values. We could think of removing the observation with 3 missing values, but it has more than the half of the values, we are following a different aproach. Additionally, we see that both variables “PositiveAffect” and “NegativeAffect” have 15 NAs

The “mice” package in R is a powerful tool which is an iterative imputation method that replaces missing values with multiple imputations using a regression model. The imputed values are then used to estimate the missing values in the subsequent iteration until the convergence criteria are met. We are going to use Random Forest as method

```
# If we want to remove the observation:   data[-which(rowSums(is.na(data)) ==5,)

imp <- mice(data2, method = 'rf', m = 7)
```

```
##
##  iter imp variable
##   1   1  HLE  PositiveAffect  NegativeAffect
##   1   2  HLE  PositiveAffect  NegativeAffect
##   1   3  HLE  PositiveAffect  NegativeAffect
##   1   4  HLE  PositiveAffect  NegativeAffect
##   1   5  HLE  PositiveAffect  NegativeAffect
##   1   6  HLE  PositiveAffect  NegativeAffect
##   1   7  HLE  PositiveAffect  NegativeAffect
##   2   1  HLE  PositiveAffect  NegativeAffect
##   2   2  HLE  PositiveAffect  NegativeAffect
##   2   3  HLE  PositiveAffect  NegativeAffect
##   2   4  HLE  PositiveAffect  NegativeAffect
##   2   5  HLE  PositiveAffect  NegativeAffect
##   2   6  HLE  PositiveAffect  NegativeAffect
##   2   7  HLE  PositiveAffect  NegativeAffect
##   3   1  HLE  PositiveAffect  NegativeAffect
##   3   2  HLE  PositiveAffect  NegativeAffect
##   3   3  HLE  PositiveAffect  NegativeAffect
##   3   4  HLE  PositiveAffect  NegativeAffect
##   3   5  HLE  PositiveAffect  NegativeAffect
##   3   6  HLE  PositiveAffect  NegativeAffect
##   3   7  HLE  PositiveAffect  NegativeAffect
##   4   1  HLE  PositiveAffect  NegativeAffect
##   4   2  HLE  PositiveAffect  NegativeAffect
##   4   3  HLE  PositiveAffect  NegativeAffect
##   4   4  HLE  PositiveAffect  NegativeAffect
##   4   5  HLE  PositiveAffect  NegativeAffect
##   4   6  HLE  PositiveAffect  NegativeAffect
##   4   7  HLE  PositiveAffect  NegativeAffect
##   5   1  HLE  PositiveAffect  NegativeAffect
##   5   2  HLE  PositiveAffect  NegativeAffect
##   5   3  HLE  PositiveAffect  NegativeAffect
##   5   4  HLE  PositiveAffect  NegativeAffect
##   5   5  HLE  PositiveAffect  NegativeAffect
##   5   6  HLE  PositiveAffect  NegativeAffect
##   5   7  HLE  PositiveAffect  NegativeAffect
```

```
data2 = complete(imp)
summary(data2)
```

```
##      Score          LogGDP       SocialSupport        HLE
##  Min.   :1.859   Min.   : 5.527   Min.   :0.3413   Min.   :51.53
##  1st Qu.:4.724   1st Qu.: 8.591   1st Qu.:0.7220   1st Qu.:60.50
##  Median :5.684   Median : 9.567   Median :0.8271   Median :65.83
##  Mean   :5.540   Mean   : 9.450   Mean   :0.7990   Mean   :64.91
##  3rd Qu.:6.334   3rd Qu.:10.540   3rd Qu.:0.8960   3rd Qu.:69.35
##  Max.   :7.804   Max.   :11.660   Max.   :0.9825   Max.   :77.28
##     Freedom         Generosity       PerceptionsCorruption ScoreDystopia
##  Min.   :0.3816   Min.   :-0.254276   Min.   :0.1461        Min.   :1.778
##  1st Qu.:0.7239   1st Qu.:-0.073543   1st Qu.:0.6678        1st Qu.:1.778
## Median :0.8005   Median : 0.001419   Median :0.7736       Median :1.778
## Mean   :0.7874   Mean   : 0.022444   Mean   :0.7254       Mean   :1.778
## 3rd Qu.:0.8745   3rd Qu.: 0.117025   3rd Qu.:0.8459       3rd Qu.:1.778
## Max.   :0.9614   Max.   : 0.531386   Max.   :0.9291       Max.   :1.778
## PositiveAffect   NegativeAffect
## Min.   :0.1789   Min.   :0.1161
## 1st Qu.:0.5901   1st Qu.:0.2327
## Median :0.6628   Median :0.2747
## Mean   :0.6513   Mean   :0.2924
## 3rd Qu.:0.7284   3rd Qu.:0.3465
## Max.   :0.8344   Max.   :0.6067
```

```r
sum(is.na(data2))
```

```
## [1] 0
```

Now, we don't have missing values anymore.

3 Visualization

Visualizations are a crucial part of EDA. Effective visualizations help you understand data distributions, trends, and anomalies. Along this section we are going to make some questions and use a plot to answer them.

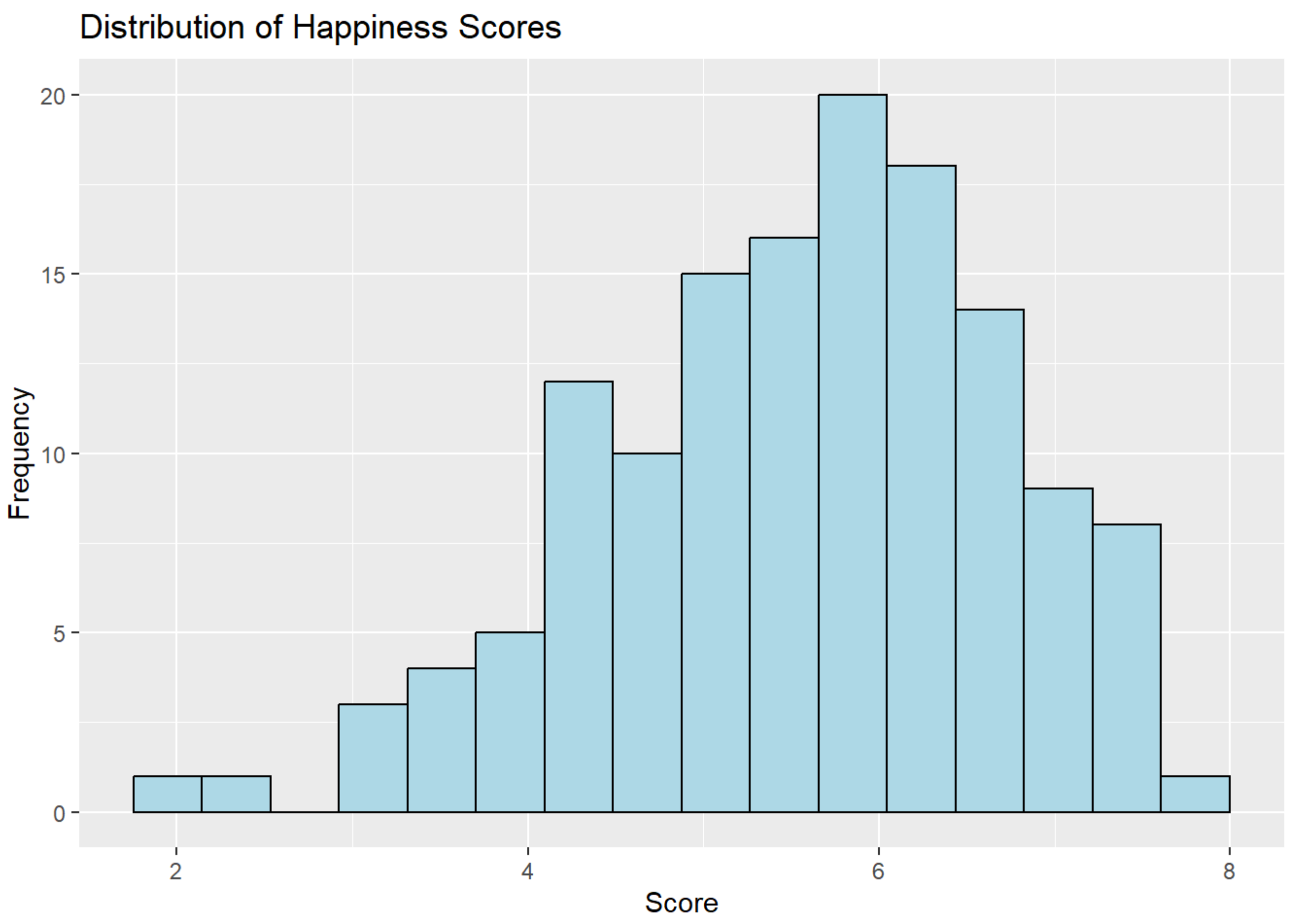
3.1 Univariate: Dimension 1

We have already seen the boxplots.

1) What is the distribution of the "Score"? In this case we are going to generate a histogram that shows the distribution of happiness scores.

```r
ggplot(data, aes(x = Score)) +
  geom_histogram(binwidth = 0.39, fill = "lightblue", color = "black") +
  labs(title = "Distribution of Happiness Scores",
       x = "Score",
       y = "Frequency")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



It is slightly skewed to the right (positively skewed), which means that higher happiness scores are more common. As the histogram is spread, it indicates than Score has variability. See possible outliers socre < 2

```
sum(which(data2$Score < 2))
```

```
## [1] 137
```

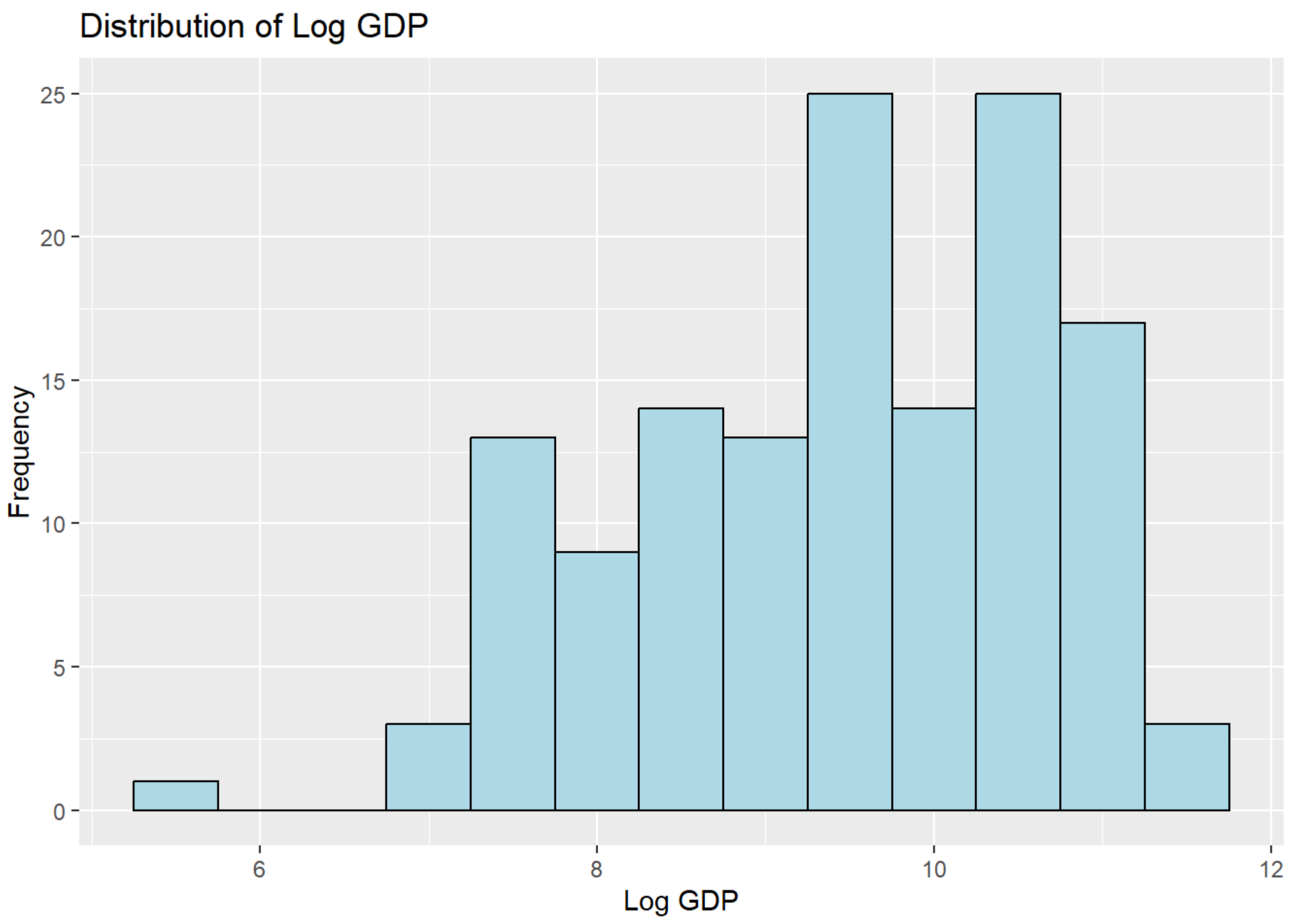
```
# We have 2 observations with a really low score
r <- which(data2$Score < 3)
n <- data$Country[r]
print(n)
```

```
## [1] "Lebanon"     "Afghanistan"
```

Now, I think that these observations are not outliers as in “Lebanon” and “Afghanistan” is understandable a low score because of their bad situation.

```
# Create a histogram for the "LogGDP" variable
ggplot(data, aes(x = LogGDP)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
  labs(title = "Distribution of Log GDP",
       x = "Log GDP",
       y = "Frequency")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

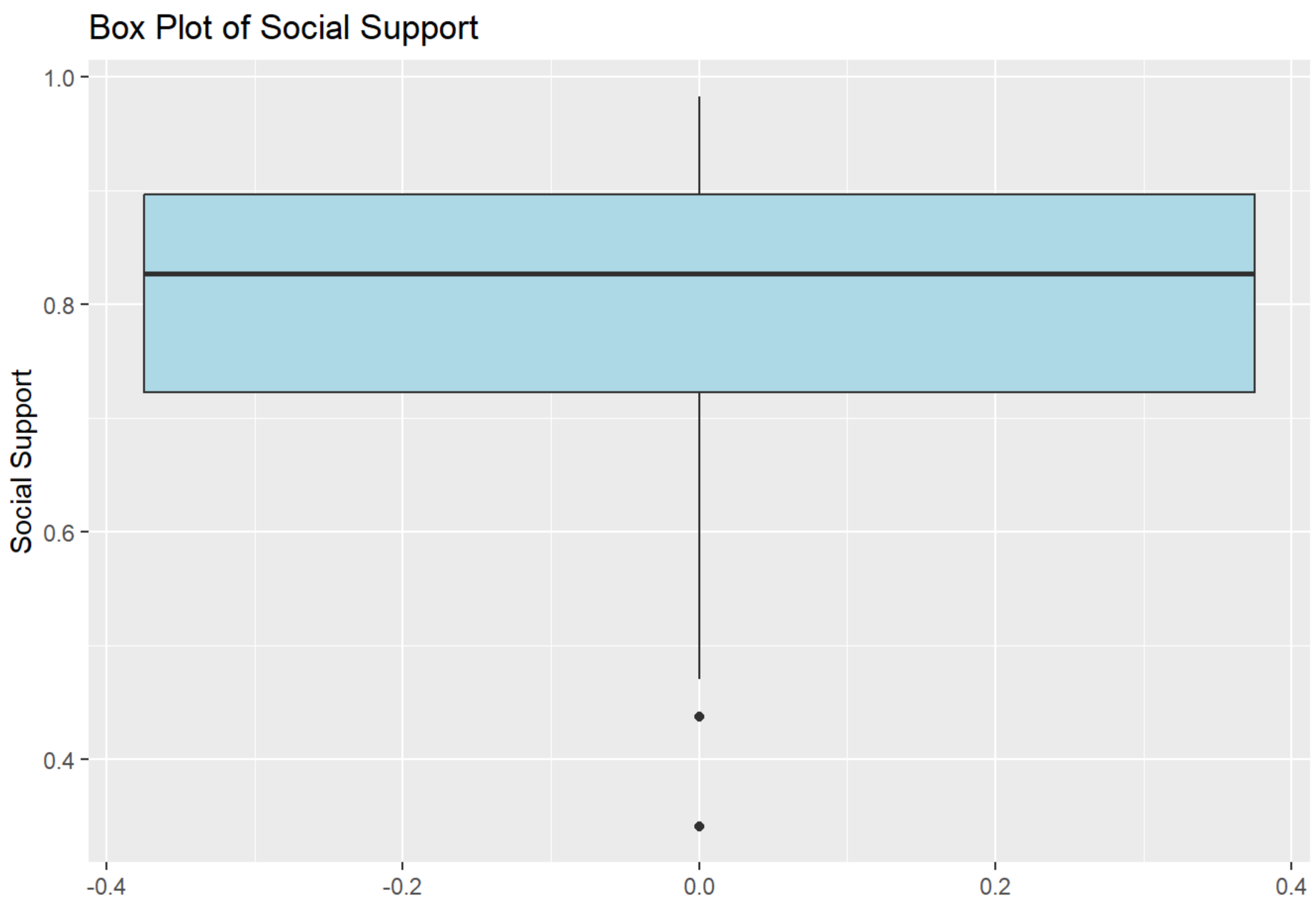


```r
print(data$Country[which(data2$LogGDP < 6)])   # Venezuela
```

```
## [1] "Venezuela"
```

```r
ggplot(data, aes(y = SocialSupport)) +
  geom_boxplot(fill = "lightblue") +
  labs(title = "Box Plot of Social Support",
       x = "",
       y = "Social Support")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

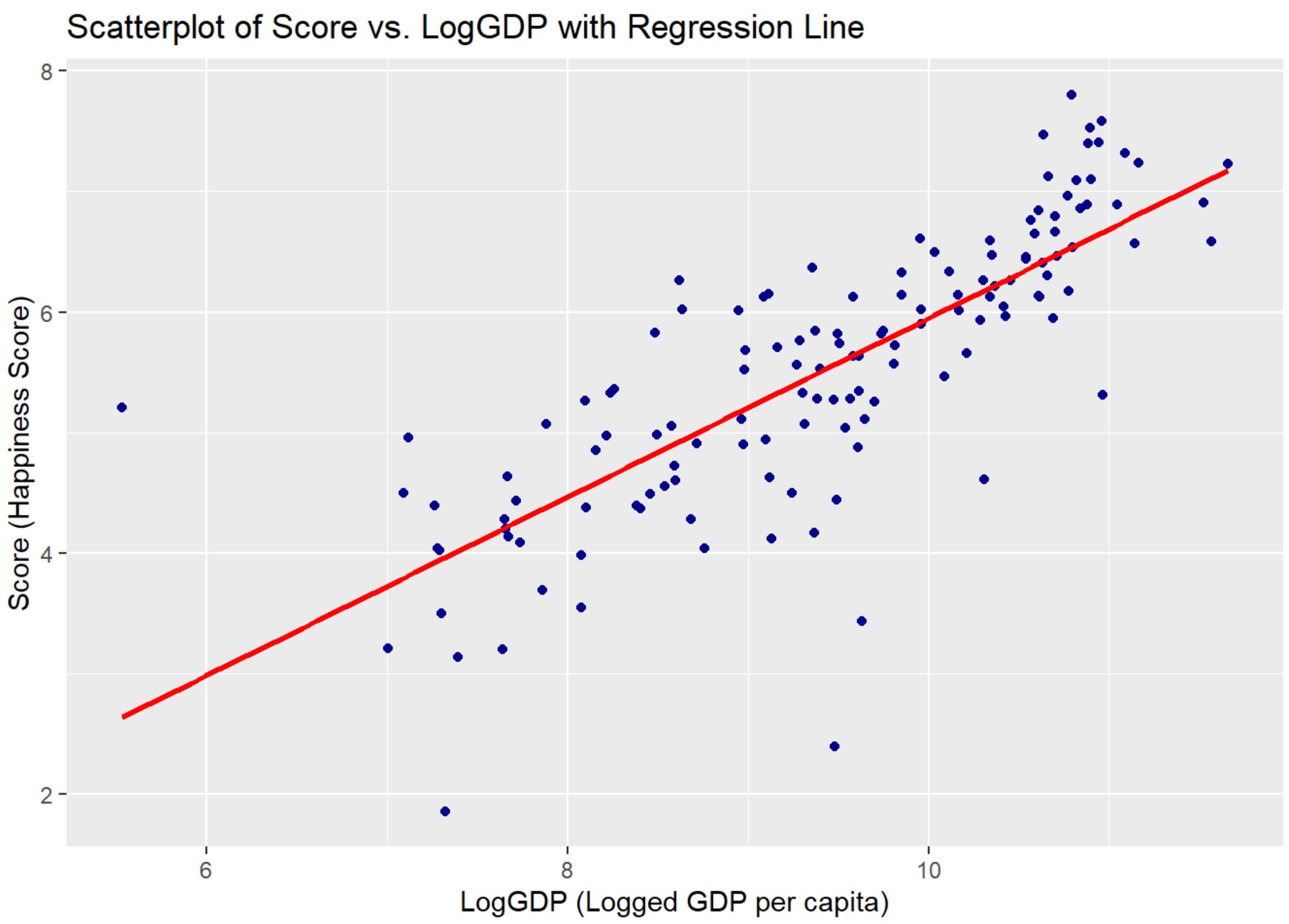


##Dimension 2: bivariate analysis (scatter plots)

As we want to know weather the wealthy a contry is has an impact on the level of happines, we are going to make an scatterplot to visualize the relationship

```
ggplot(data2, aes(x = LogGDP, y = Score)) +
  geom_point(color = "darkblue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Scatterplot of Score vs. LogGDP with Regression Line",
       x = "LogGDP (Logged GDP per capita)",
       y = "Score (Happiness Score)")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

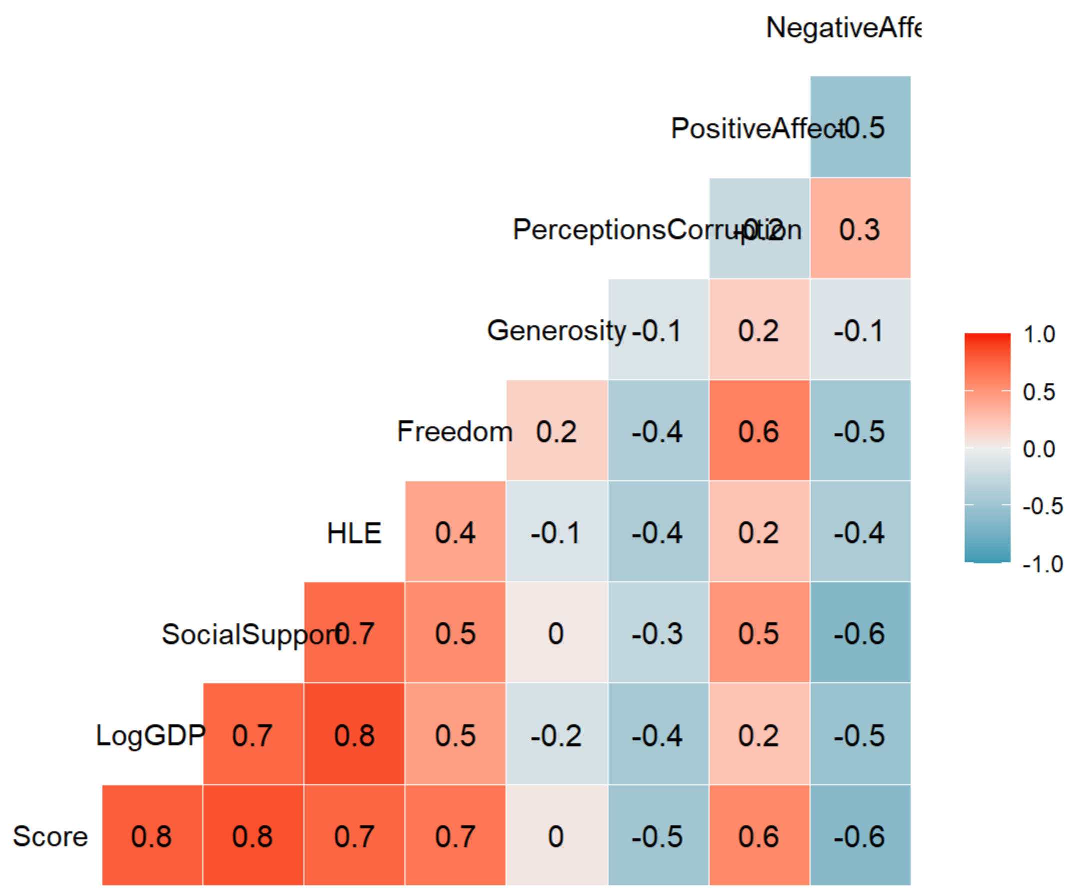


Without any doubt, we can see a strong positive linear relationship between GDP and Score.

Multiple scatterplots:

```
# When applying ggcorr() -> obtain Warning: the standard deviation is zero.
# a standard deviation of zero means that one or more variables in our dataset have no variabi
lity because all of # their values are the same. So, we check this
vars_0sd = colnames(data2)[apply(data2, 2, sd) == 0]
# We realise that "ScoreDystopia" takes always the value 1.777825. Therefore,I am going to exc
lude it as it won't provide meaningful correlation values.

data2 =  data2 %>% dplyr::select(-ScoreDystopia)

ggcorr(data2, label = T)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

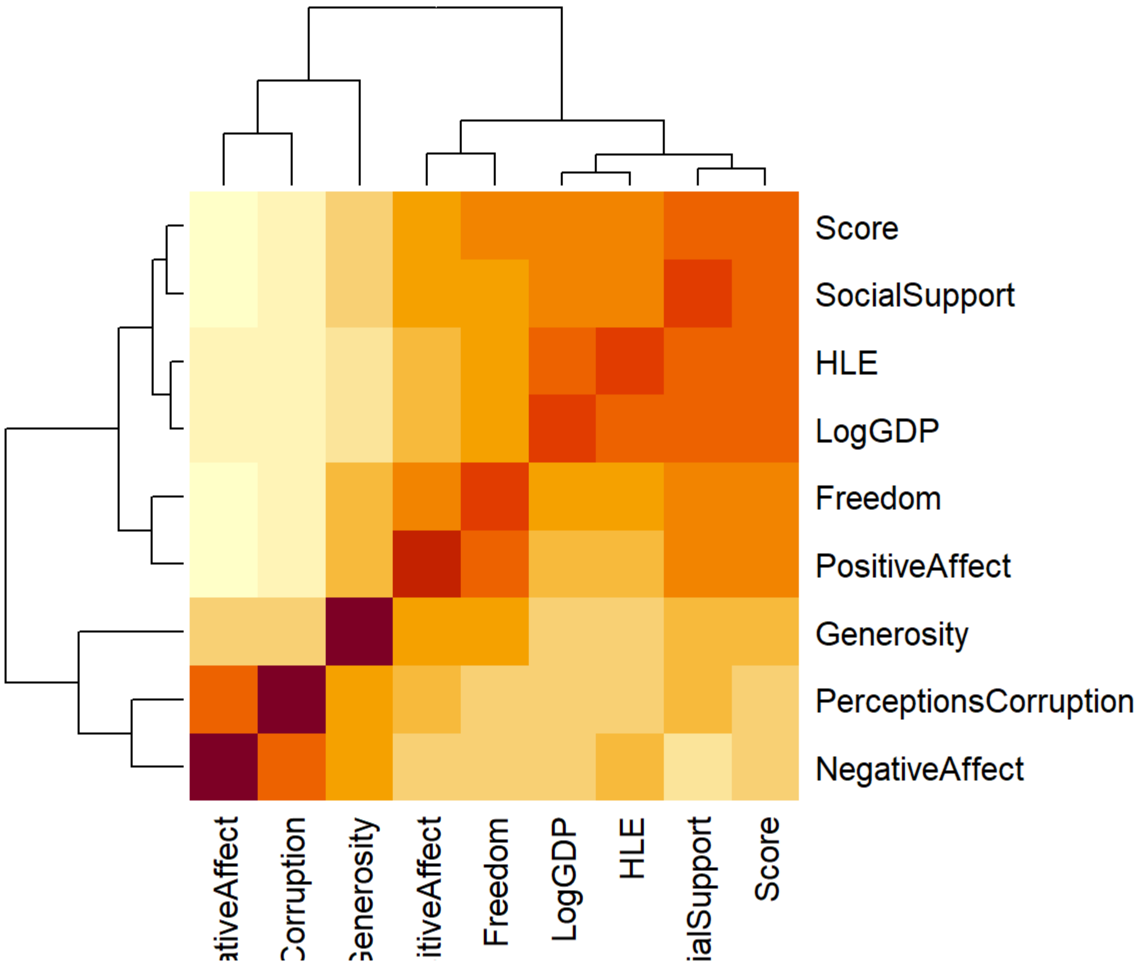


```r
heatmap(cor(data2))
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



We can see that the hapiness score is highly positive related with GDP and SocialSupport, and negative related with PerceptionsCorruption and NegativeAffect. These are expected results. However, I find it surprising that Score and Generosity are independent.

Let’s see the relation between each pair of our variables

```
pairs(data2, pch = 19, col = "lightblue")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

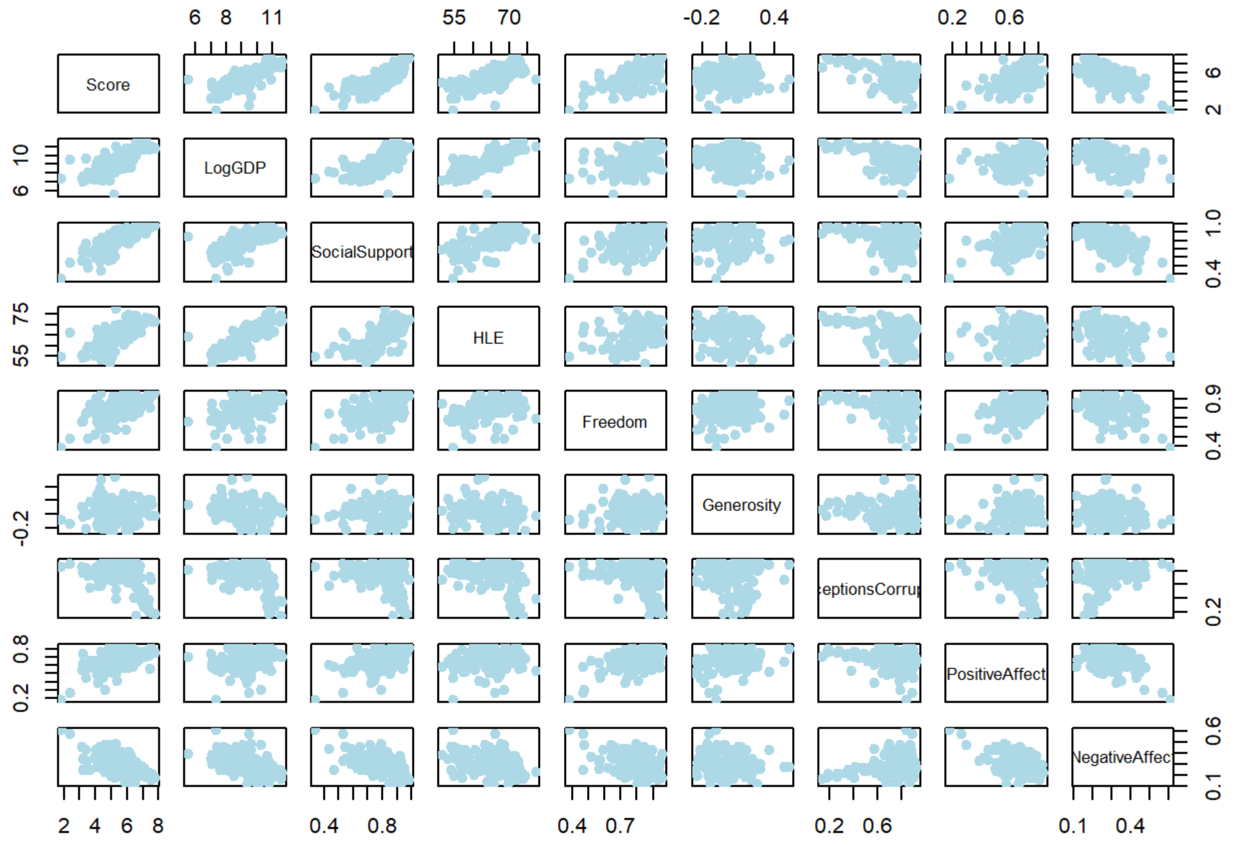
```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



4 PCA

(lab PCA)

Principal Component Analysis (PCA) is a widely used statistical technique for dimensionality reduction and data transformation. PCA is employed when dealing with high-dimensional data, where each observation is described by multiple variables. It allows to identify and highlight the most important patterns or directions in the data. It creates new variables (principal components) that are linear combinations of the original variables, capturing the maximum variance in the data.

It is important to remember that PCA is sensitive to the scaling of the data. When variables have different scales, those with larger scales will contribute more to the overall variance. This can lead to the PCA being dominated by variables with large scales, while variables with smaller scales may have limited influence on the principal components.

```
pca = prcomp(data2, scale=T)
# pca = princomp(nba, cor=T) # the same, but using SVD instead of eigen decomposition
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.1815 1.1750 0.93059 0.81871 0.73010 0.58083 0.43591
## Proportion of Variance 0.5288 0.1534 0.09622 0.07448 0.05923 0.03748 0.02111
## Cumulative Proportion  0.5288 0.6822 0.77841 0.85288 0.91211 0.94960 0.97071
##                           PC8     PC9
## Standard deviation     0.38588 0.33869
## Proportion of Variance 0.01655 0.01275
## Cumulative Proportion  0.98725 1.00000
```

We see that PC1 explains 52.9% of the total variance, it is the most important component. PC2 is the second component which explains more variance, 15.6%. PC3 explains 9.76%. With these 3 component we can explain 78.26% of the variance. But, is this enough or should we take more components?

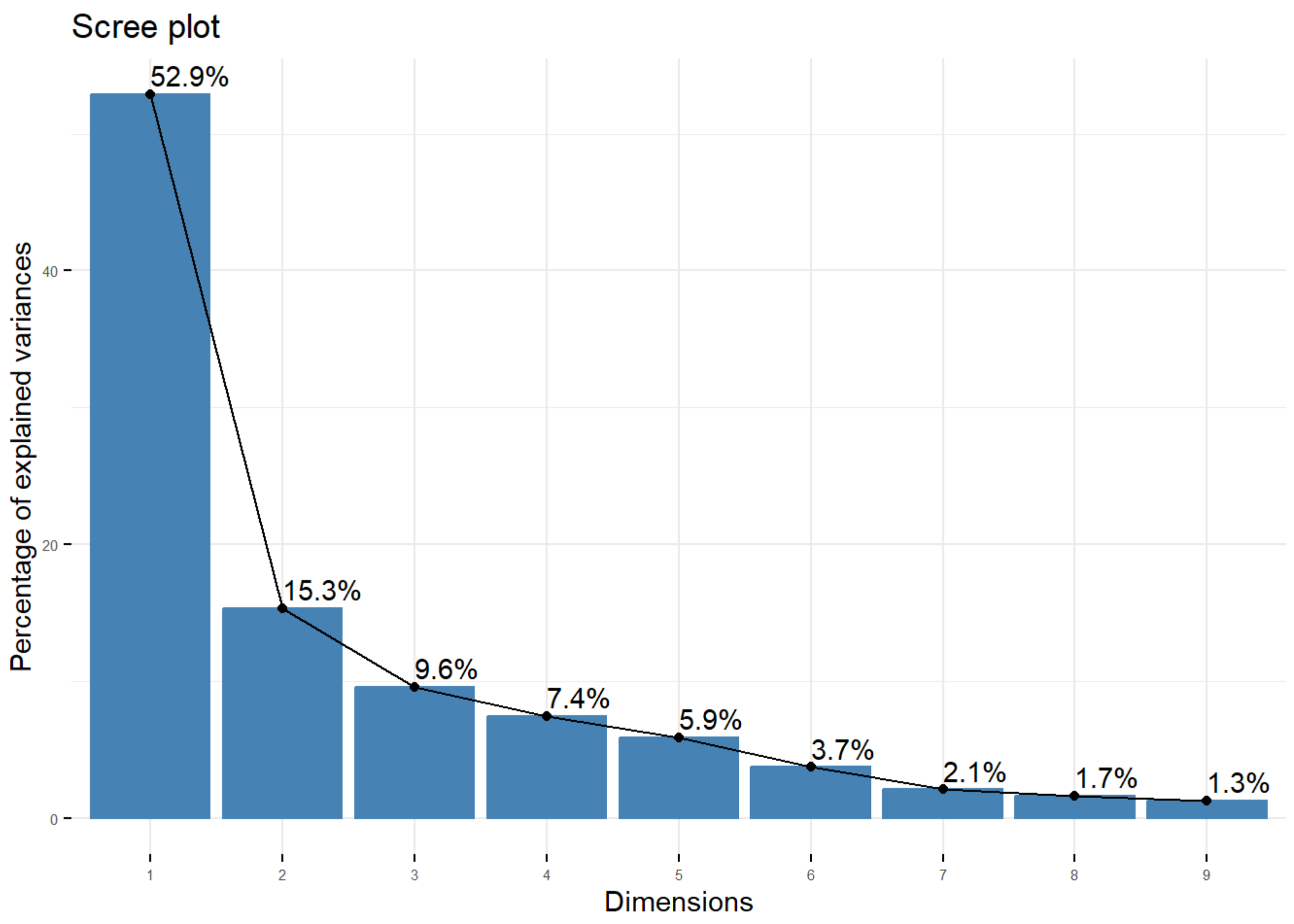
4.1 Number of PCA components

We are using **screeplot** to generate a PCA scree plot that visually displays the explained variance for each principal component. This visualization aids in pinpointing the 'elbow point,' where the explained variance curve starts to stabilize, providing insights into the dimensionality of the data.

For that, we use fviz_screeplot() function from factoextra package:

```
fviz_screeplot(pca, addlabels = TRUE)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



We decide to take 3 components.

4.2 Interpretation of the Components

Interpreting the components in a Principal Component Analysis (PCA) involves understanding the relationship between the original variables and the principal components, which are linear combinations of the original variables.

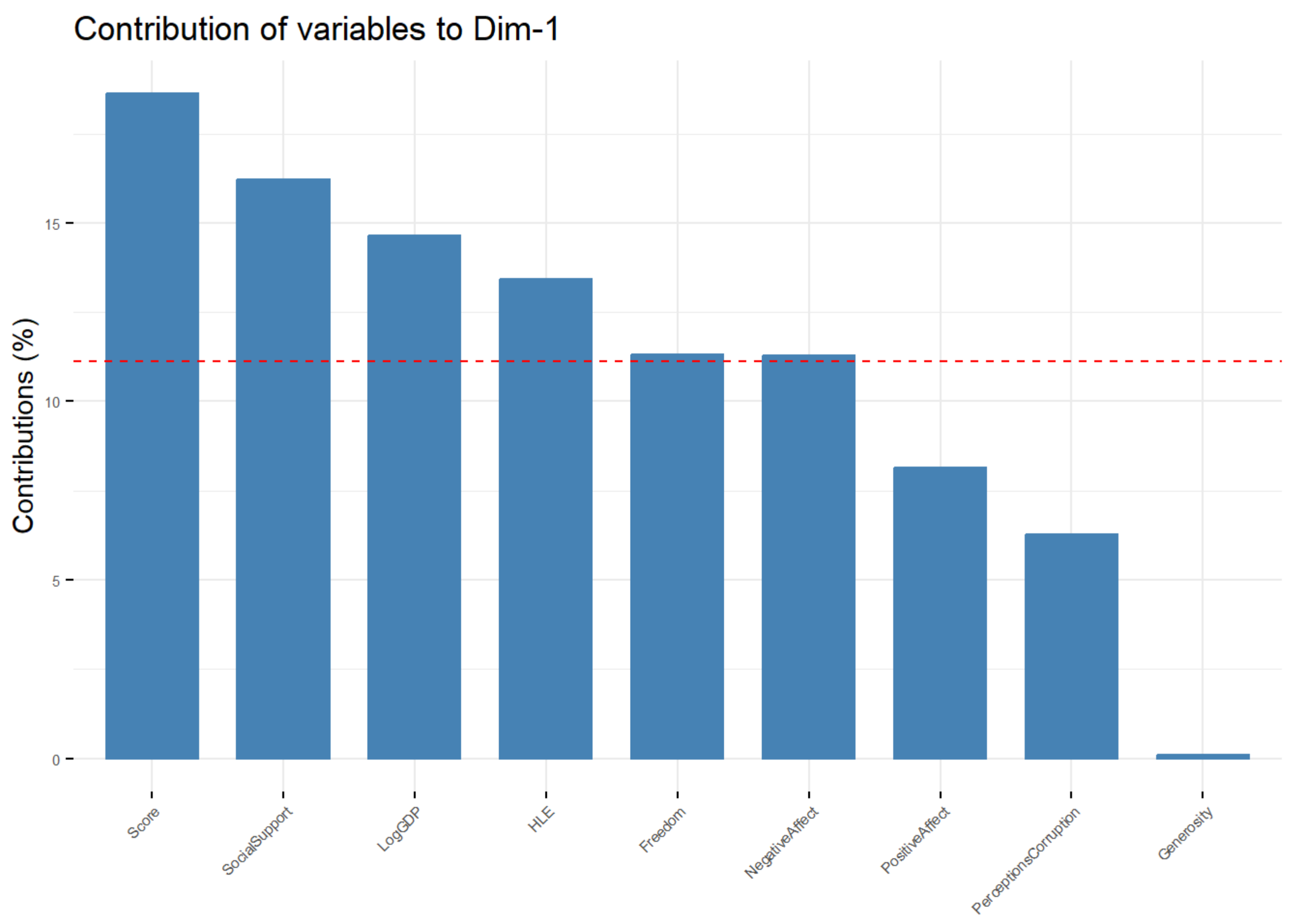
###1º Component

We are creating a bar plot for the first principal component's loadings in a PCA analysis. We need to extract the loadings which represent the strength and direction of each original variable's contribution to the first component.

First, we are going to plot the **square loadings** which indicate the proportion of variance in each original variable that is explained by the associated principal component. Higher squared loadings indicate that the variable contributes more to that component.

```
fviz_contrib(pca, choice = "var", axes = 1)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

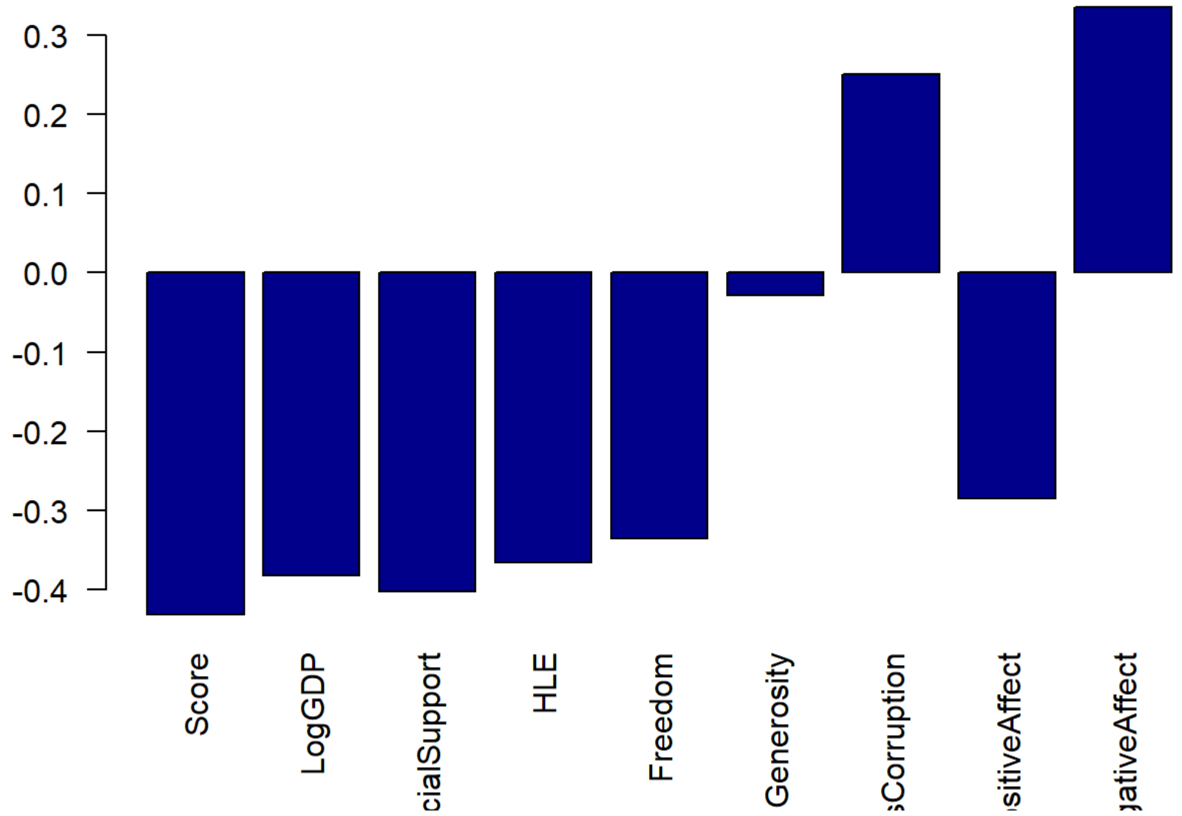


We see that "Score" is the one which most contributes as expected. On the other hand "Generosity" is the one which contributes less as it was independent to Score.

Now, let's analyze the Direction of Component Loadings. For that, we need to examine the signs of the component loadings. Positive loadings indicate a positive relationship with the component, while negative loadings indicate a negative relationship.

```
barplot(pca$rotation[,1], las=2, col="darkblue")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



We see that “PerceptionsCorruption” and “NegativeAffect” have a positive effect in PC1 whereas the other variables a negative effect. Therefore, we can interpret this PC1 has the “unhapiness of the countries”

pca$x[,1] gives you a vector of the PC1 scores for all the data points in your analysis. These scores represent how each observation contributes to PC1 and can be used for various purposes, including ranking or analyzing the data based on this component. Now we can rank the Countries by their PC1 scores: level of unhappiness:

```
# Take the first 5 countries which have a high value of the scores but with NEGATIVE sign whic
h  indicates a negative association with PC1. These countries have less level of unhappiness.
rows_first = order(pca$x[,1])[1:5]
country_names[rows_first]
```

```
## [1] "Finland" "Denmark" "Sweden"  "Norway"  "Iceland"
```

```
# Take the first 5 countries which have a high value of the scores but with POSITIVE sign whic
h  indicates a positive association with PC1. These countries have higer level of unhappiness.
rows_last <- tail(order(pca$x[,1]), 5)
country_names[rows_last]
```

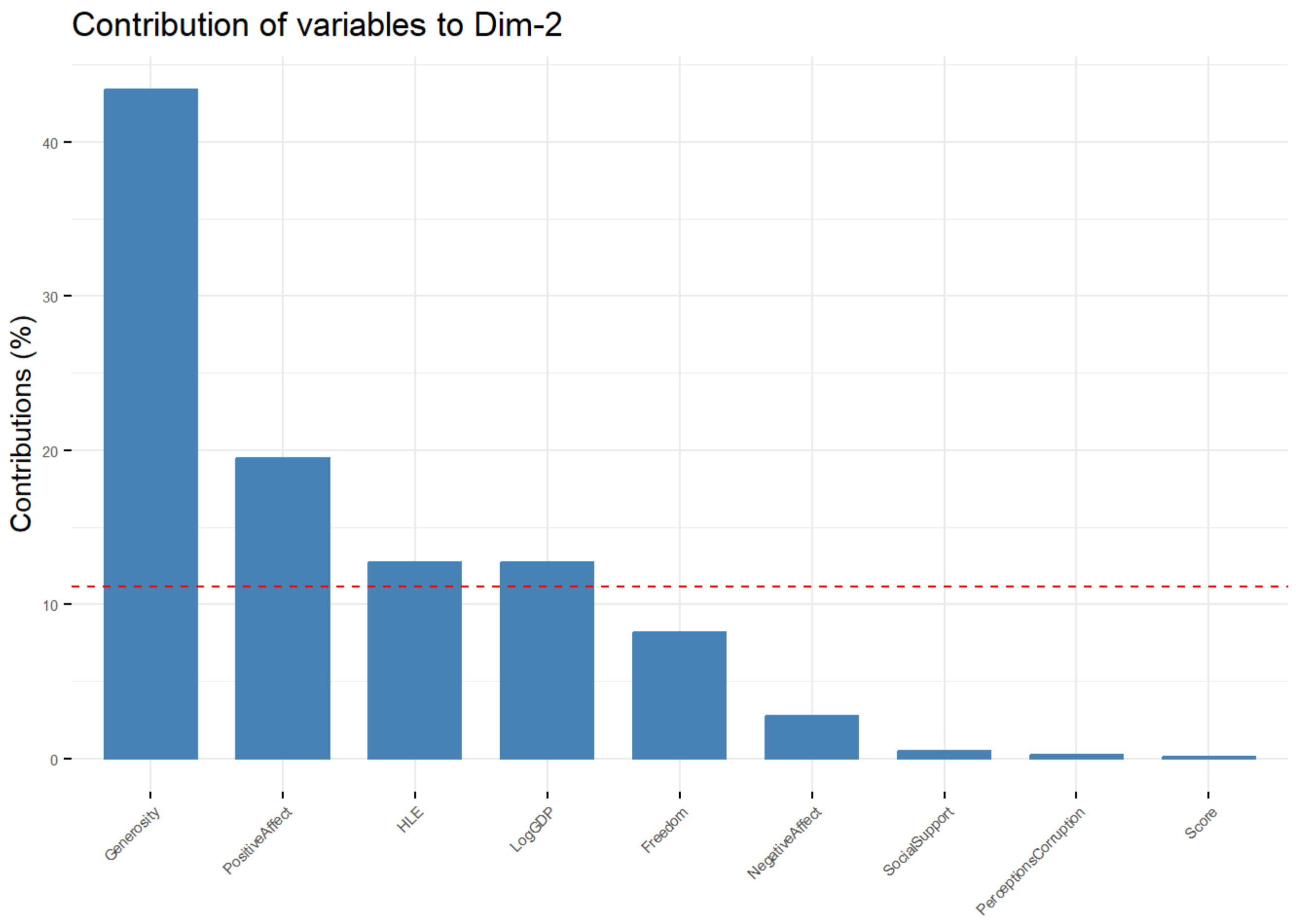
```
## [1] "Comoros"      "Gambia"       "Sierra Leone" "Lebanon"      "Afghanistan"
```

###2º Component

We do the same for PC2

```
fviz_contrib(pca, choice = "var", axes = 2)
```

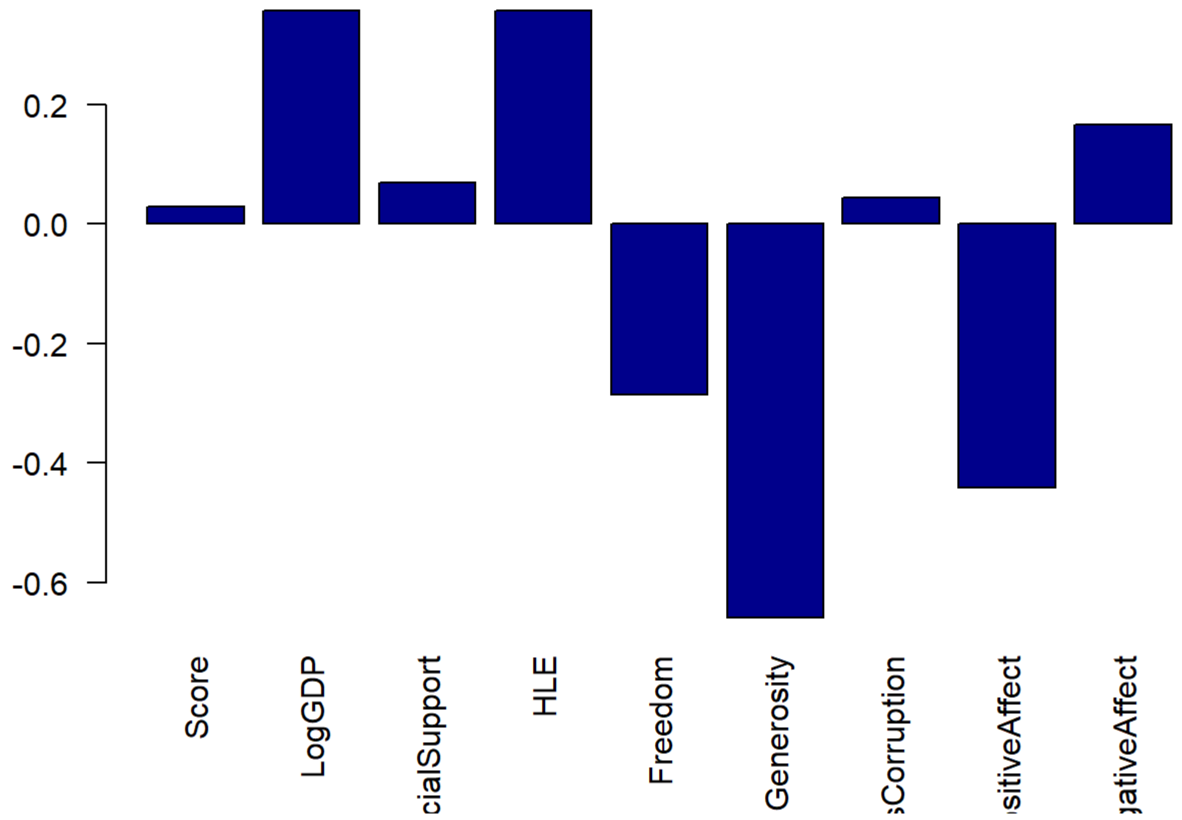
```
## Error : The fig.showtext code chunk option must be TRUE
```



In this case, we see that the contribution of “Score” is really low, so we are not interpreting this component relating it with level of hapiness. Now, we focus on “Generosity” which is the variable that contribute most.

```
barplot(pca$rotation[,2], las=2, col="darkblue")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

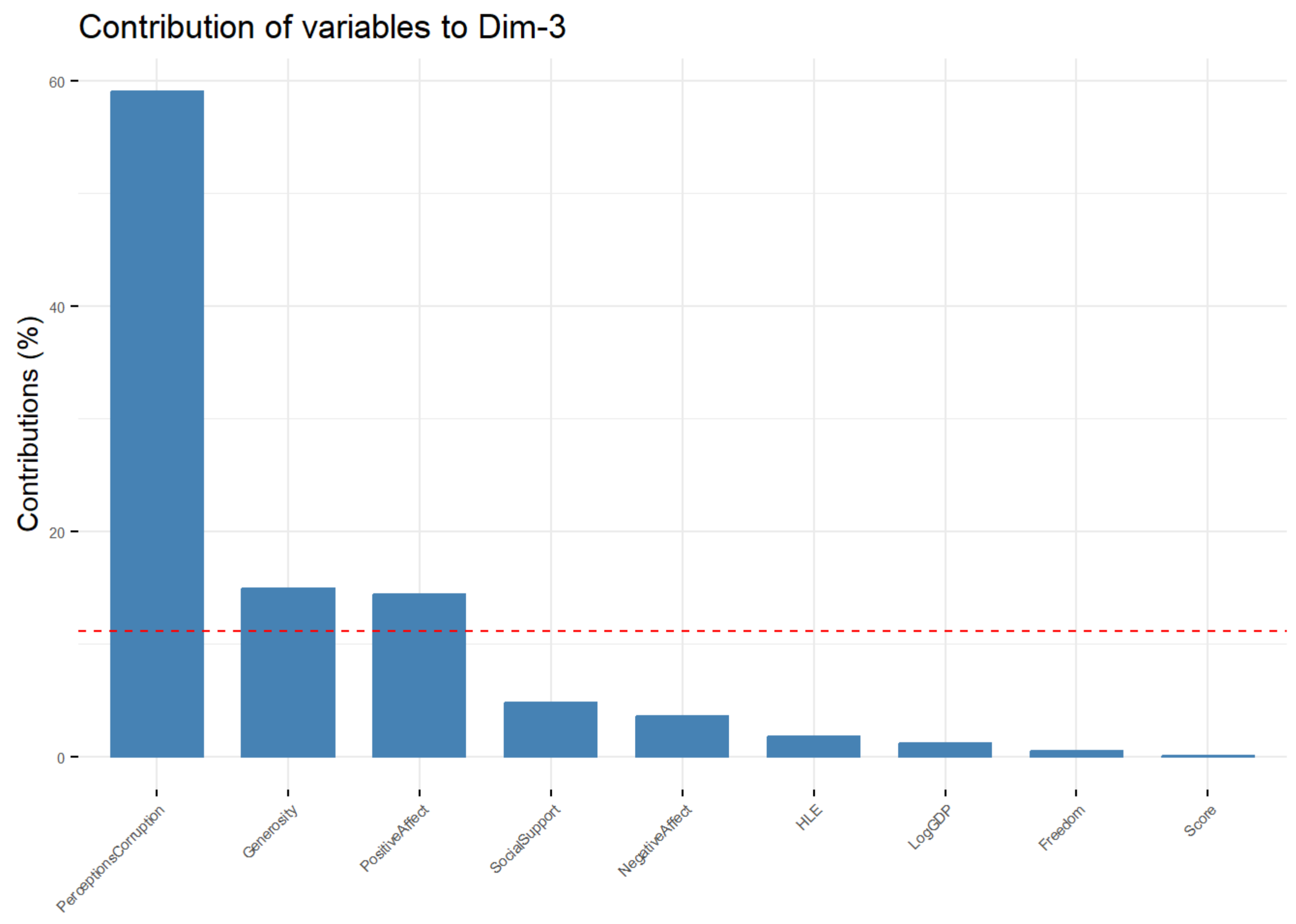


We can see that when PC2 decreases with “Generosity” and “PositiveAffect” while it increases with “LogGDP” and ” HLE”. We could interpret this as the countries with higher GDP and Healthy Life Expectancy tend to be less generous and have less laugh, enjoyment (positiveAffect)

4.2.1 **3º Component**

```
fviz_contrib(pca, choice = "var", axes = 3)
```

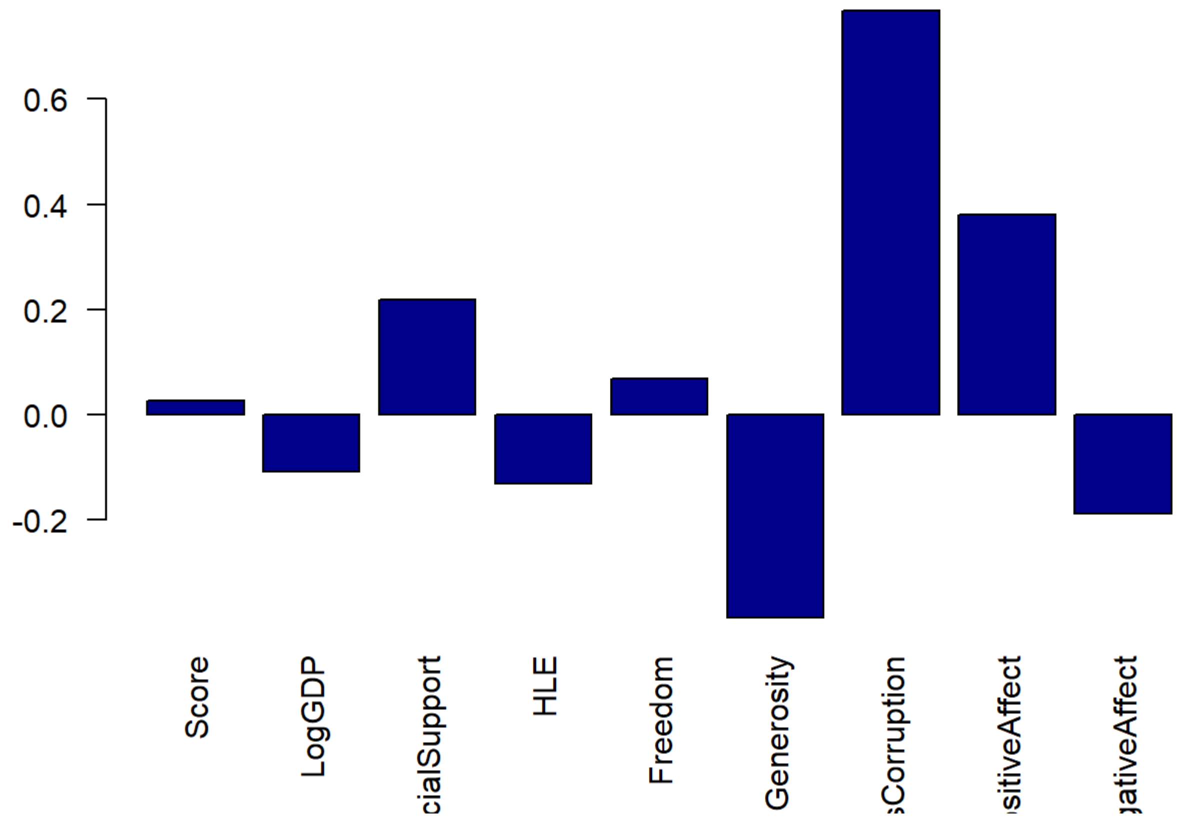
```
## Error : The fig.showtext code chunk option must be TRUE
```



Here, “PerceptionsCorruption” is now our target.

```
barplot(pca$rotation[,3], las=2, col="darkblue")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



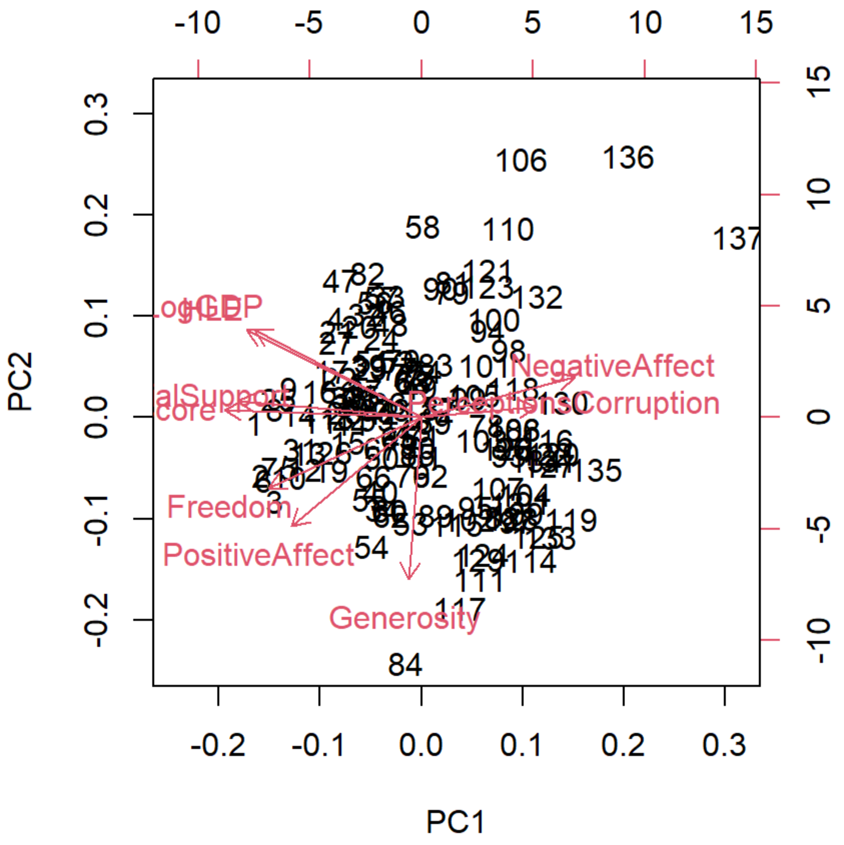
4.2.2 Biplot

A biplot displays both the observations (data points) and the variables (original features), allowing you to visualize the relationships and associations between them in a reduced-dimensional space. The variables are represented by arrows that point in the direction of increasing values. The length of these arrows represents the strength of the relationship between the variables and the principal components. Longer arrows indicate stronger relationships.

```r
biplot(pca)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

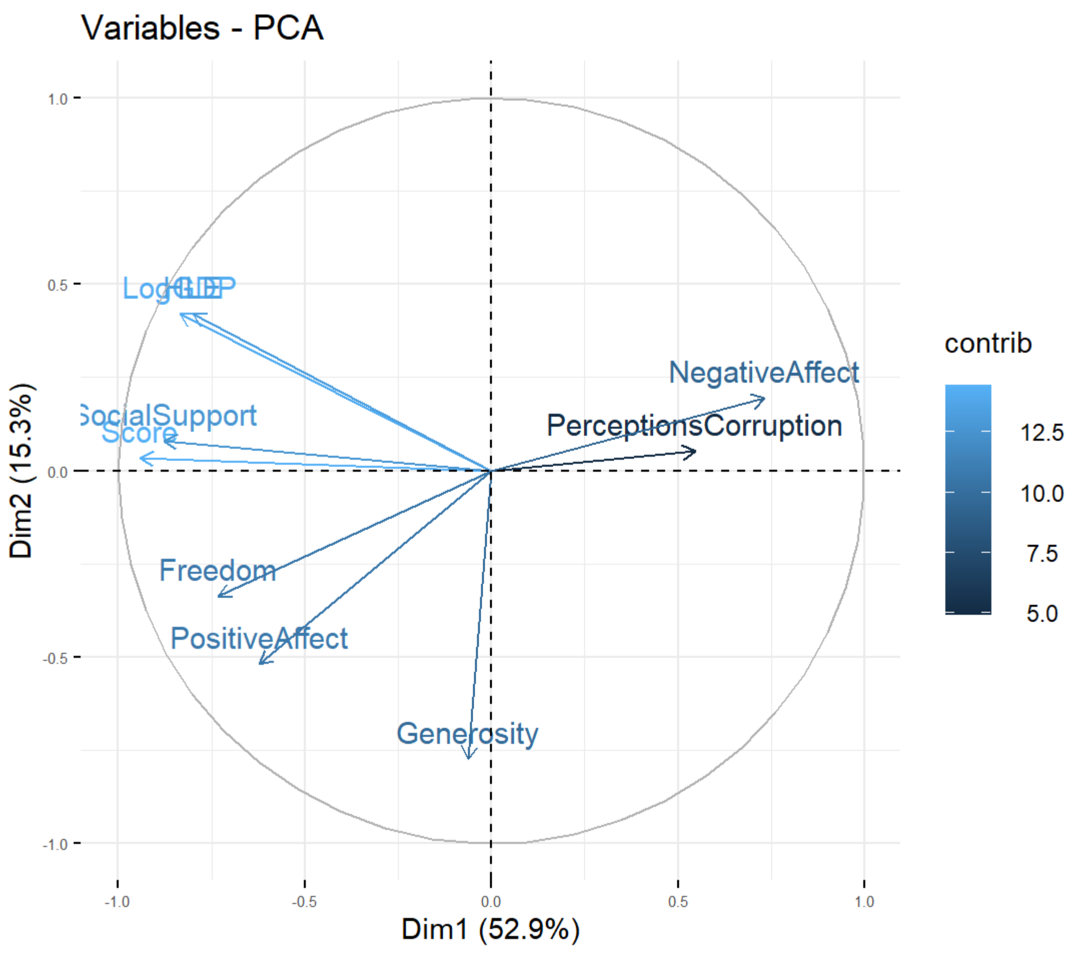


In this case, it is not very useful because of the high amount of countries

We can make a plot that displays the contributions of each variable to the principal components. Variables that make a stronger contribution to a component will be highlighted in the plot, helping you identify which variables are most influential in shaping the PCA results.

```
fviz_pca_var(pca, col.var = "contrib")
```

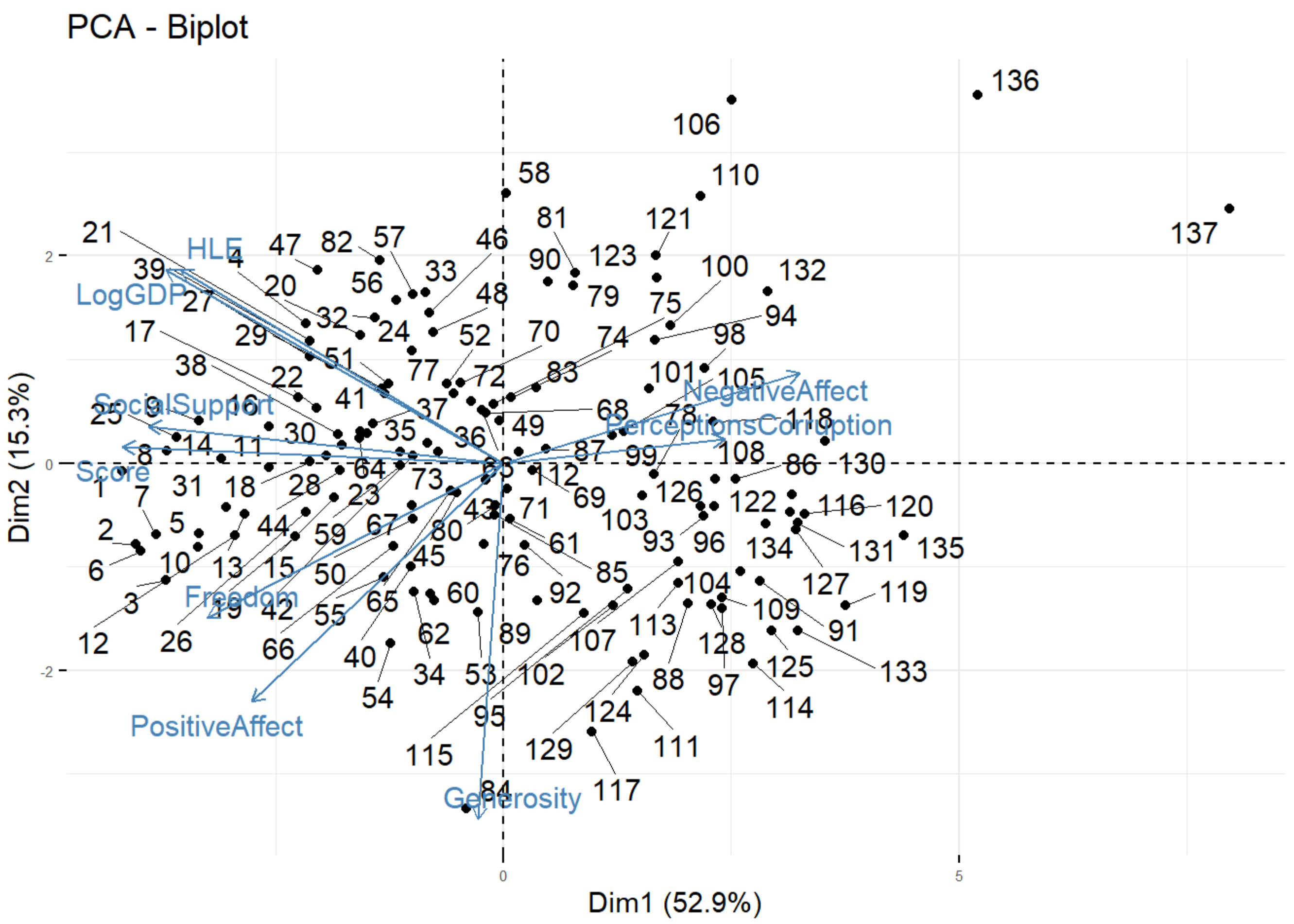
```
## Error : The fig.showtext code chunk option must be TRUE
```



As we have said previously, we see we can see a positive contribution of "PerceptionsCorruption" and "NegativeAffect" on PC1 (Dim1), as well as "LogGDP" and "HLE" on PC2 (Dim2).

```
fviz_pca_biplot(pca, repel = TRUE)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



With repel = TRUE, the labels of observations and variables in the biplot will automatically adjust their positions to avoid overlapping, making it easier to interpret and visualize the relationships between observations and variables in the reduced-dimensional space when we have a large dataset.

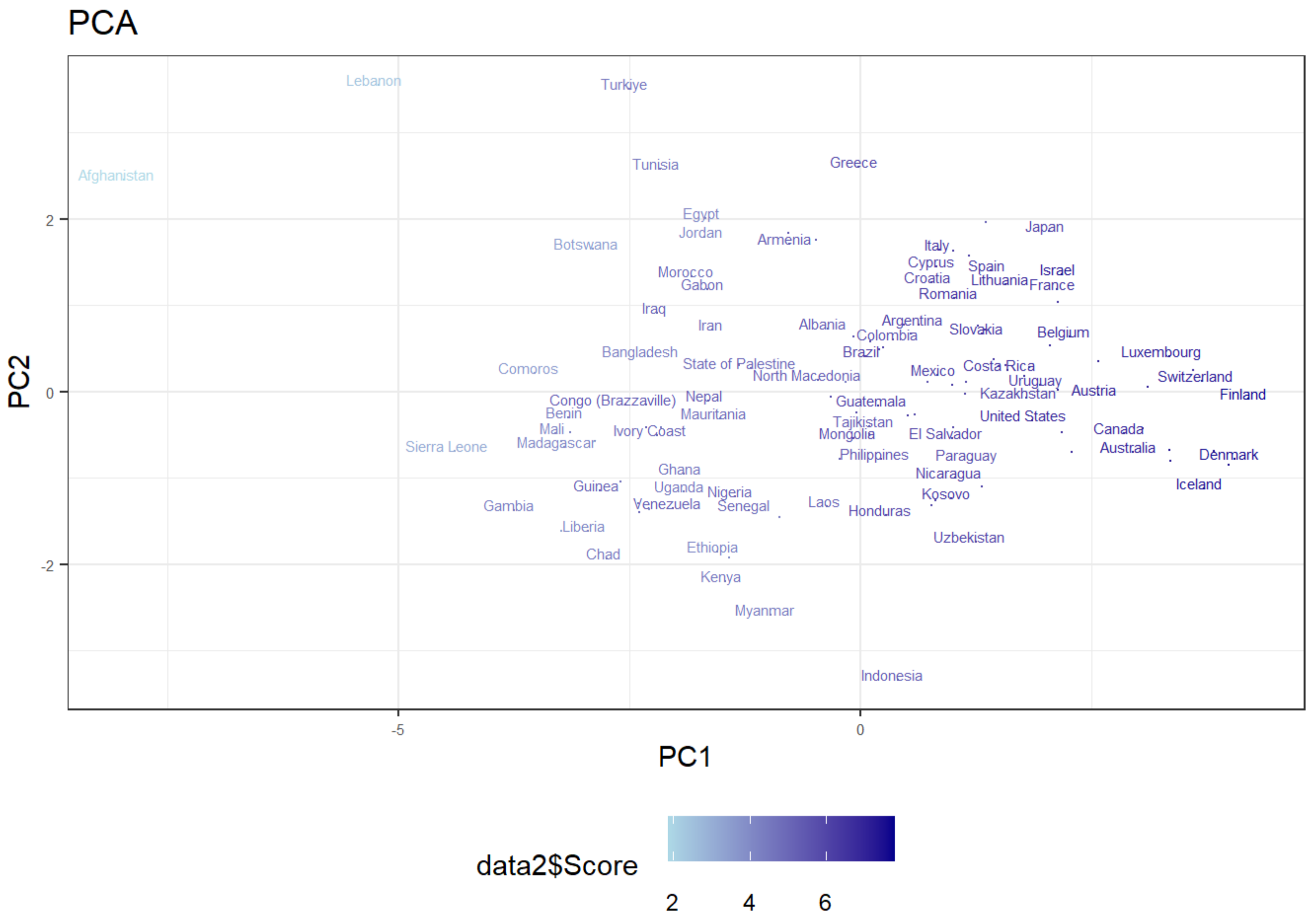
4.3 The Scores

Remember, for the \(j\)-th principal component: \(Z_j = X a_j\), \(a_j\) denotes the loadings, and \(Z_j\) denotes the scores

Let’s plot the first two scores, using colors for minutes played:

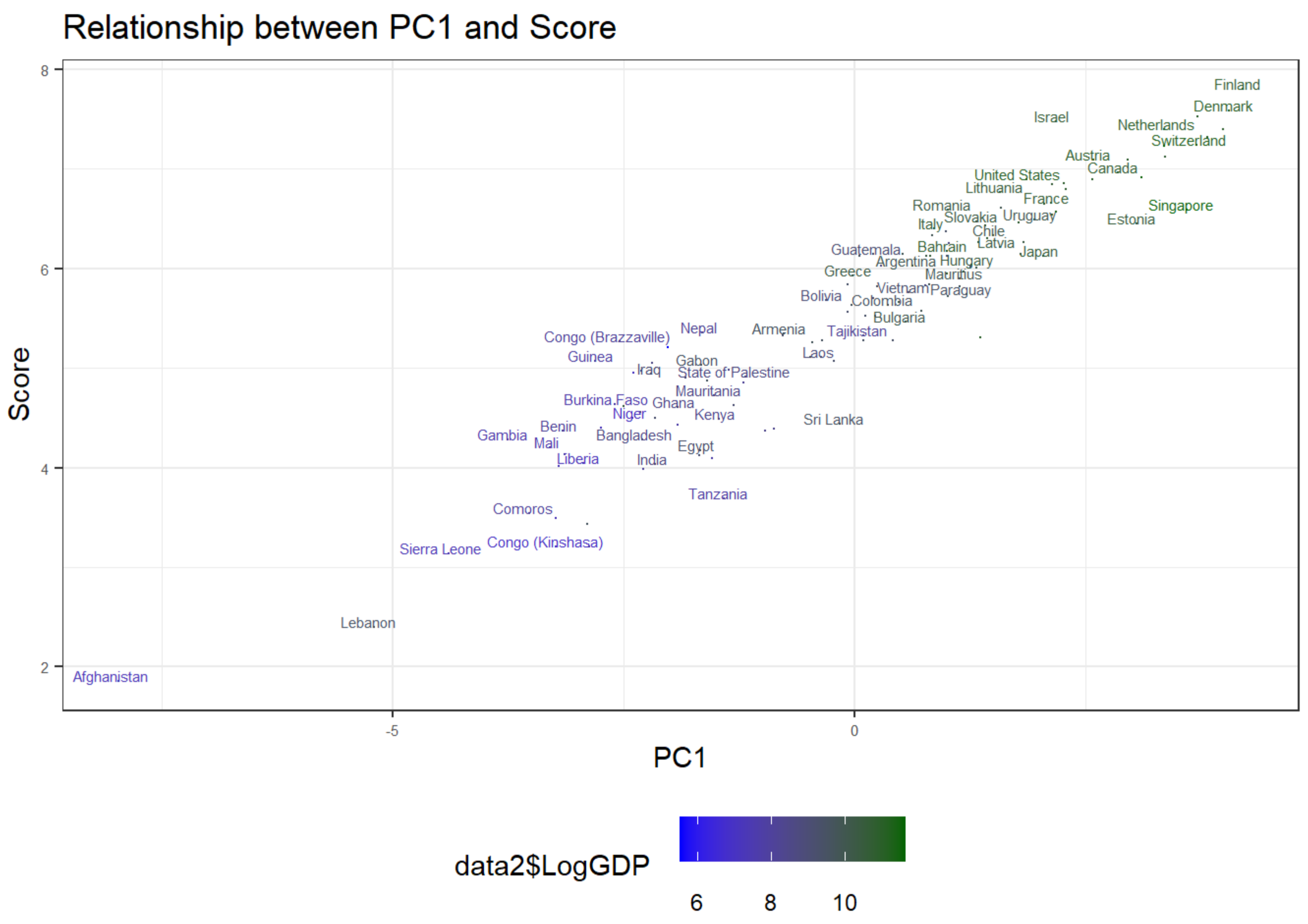
```
data.frame(z1=-pca$x[,1],z2=pca$x[,2]) %>%
  ggplot(aes(z1,z2,label=country_names,color=data2$Score)) + geom_point(size=0) +
  labs(title="PCA", x="PC1", y="PC2") +
  theme_bw() + scale_color_gradient(low="lightblue", high="darkblue")+theme(legend.position="b
ottom") + geom_text(size=2, hjust=0.6, vjust=0, check_overlap = TRUE)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



```r
data.frame(z1=-pca$x[,1],z2=data2$Score) %>%
  ggplot(aes(z1,z2,label=country_names,color=data2$LogGDP)) + geom_point(size=0) +
  labs(title="Relationship between PC1 and Score", x="PC1", y="Score") +
  theme_bw() + scale_color_gradient(low="blue", high="darkgreen")+theme(legend.position="botto
m") + geom_text(size=2, hjust=0.6, vjust=0, check_overlap = TRUE)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



We can see a clear linear relationship

5 Factor Analysis

(Based on lab FA)

Factor Analysis is a statistical method used for data reduction and dimensionality reduction. The goal of Factor Analysis is to identify a smaller number of latent factors that can explain the correlations or covariations between the observed variables. It's common to perform factor rotation to make the factors more interpretable. In our case we are using "varimax"

```
fa <- factanal(data2, factors = 4, rotation="varimax", scores="Bartlett", lower = 0.01)
fa
```

```
##
## Call:
## factanal(x = data2, factors = 4, scores = "Bartlett", rotation = "varimax",     lower = 0.01)
##
## Uniquenesses:
##              Score             LogGDP       SocialSupport
##              0.105              0.092              0.010
##                HLE            Freedom          Generosity
##              0.211              0.370              0.847
## PerceptionsCorruption      PositiveAffect      NegativeAffect
##              0.624               0.344               0.010
##
## Loadings:
##                     Factor1 Factor2 Factor3 Factor4
## Score                0.731   0.476   0.304   0.203
## LogGDP               0.757   0.504  -0.177   0.222
## SocialSupport        0.915           0.265   0.281
## HLE                  0.756   0.437  -0.122   0.104
## Freedom              0.381   0.447   0.526
## Generosity                   0.377
## PerceptionsCorruption -0.197  -0.559          -0.125
## PositiveAffect       0.269   0.241   0.699   0.190
## NegativeAffect      -0.335  -0.235  -0.295  -0.858
##
##                Factor1 Factor2 Factor3 Factor4
## SS loadings      2.893   1.303   1.212   0.980
## Proportion Var   0.321   0.145   0.135   0.109
## Cumulative Var   0.321   0.466   0.601   0.710
##
## Test of the hypothesis that 4 factors are sufficient.
## The chi square statistic is 7.62 on 6 degrees of freedom.
## The p-value is 0.267
```

When the loading is close to 0, the factor does not explain too much. Therefore, if we look to SS loadings, we see that Factor4 is not very good (0.384).

Let's try to perform Factor Analysis with 3 factors

```r
fa2 <- factanal(data2, factors = 3, rotation="varimax", scores="Bartlett", lower = 0.01)
fa2
```

```
##
## Call:
## factanal(x = data2, factors = 3, scores = "Bartlett", rotation = "varimax",     lower = 0.01)
##
## Uniquenesses:
##              Score              LogGDP       SocialSupport
##              0.107               0.105               0.010
##                HLE             Freedom          Generosity
##              0.213               0.383               0.848
## PerceptionsCorruption      PositiveAffect       NegativeAffect
##              0.623               0.332               0.511
##
## Loadings:
##                     Factor1 Factor2 Factor3
## Score                0.738   0.517   0.284
## LogGDP               0.754   0.543  -0.179
## SocialSupport        0.957   0.112   0.247
## HLE                  0.730   0.479  -0.155
## Freedom              0.381   0.462   0.508
## Generosity                   0.383
## PerceptionsCorruption -0.193  -0.575
## PositiveAffect       0.308   0.259   0.712
## NegativeAffect      -0.546  -0.248  -0.360
##
##                Factor1 Factor2 Factor3
## SS loadings      3.141   1.478   1.249
## Proportion Var   0.349   0.164   0.139
## Cumulative Var   0.349   0.513   0.652
##
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 16.33 on 12 degrees of freedom.
## The p-value is 0.177
```

Now, all factor SS loadings are > 1, and we can say that they explain better If we look at the Cumulative Var, we notice that the 3 factors explain 65.6% of the Variance, which is actually not really good.

5.1 Interpretation

Now, we create a dataframe that combines the factor loadings and uniquenesses into a single data frame. Factor loading is a measure of the relationship between each observed variable and the extracted factors. High factor loadings indicate a strong relationship between the variable and the factor.

Communality represents the proportion of variance in an observed variable that is explained by the retained factors. High communality values suggest that the factor model explains a large portion of the variable’s variance. The uniqueness is complementary to communality (= 1-Communality) and it refers to the proportion of variance in an observed variable that is unique or specific to that variable and not explained by the underlying factors extracted by the analysis.

```r
cbind(fa2$loadings, fa2$uniquenesses)
```

```
##                         Factor1     Factor2     Factor3
## Score                0.73810749  0.51679527  0.28400635 0.1074599
## LogGDP               0.75368775  0.54318352 -0.17890222 0.1049006
## SocialSupport        0.95743063  0.11182498  0.24663961 0.0100000
## HLE                  0.73046279  0.47858162 -0.15544354 0.2132181
## Freedom              0.38076746  0.46247854  0.50842715 0.3826284
## Generosity          -0.05727022 -0.04386745  0.38320639 0.8479409
## PerceptionsCorruption -0.19277640 -0.57452666 -0.09843363 0.6230780
## PositiveAffect       0.30826212  0.25878534  0.71157944 0.3316556
## NegativeAffect      -0.54579368 -0.24770690 -0.36004049 0.5111276
```

```r
par(mfrow=c(3,1))
barplot(fa2$loadings[,1], names=F, las=2, col="darkblue", ylim = c(-1, 1))
```

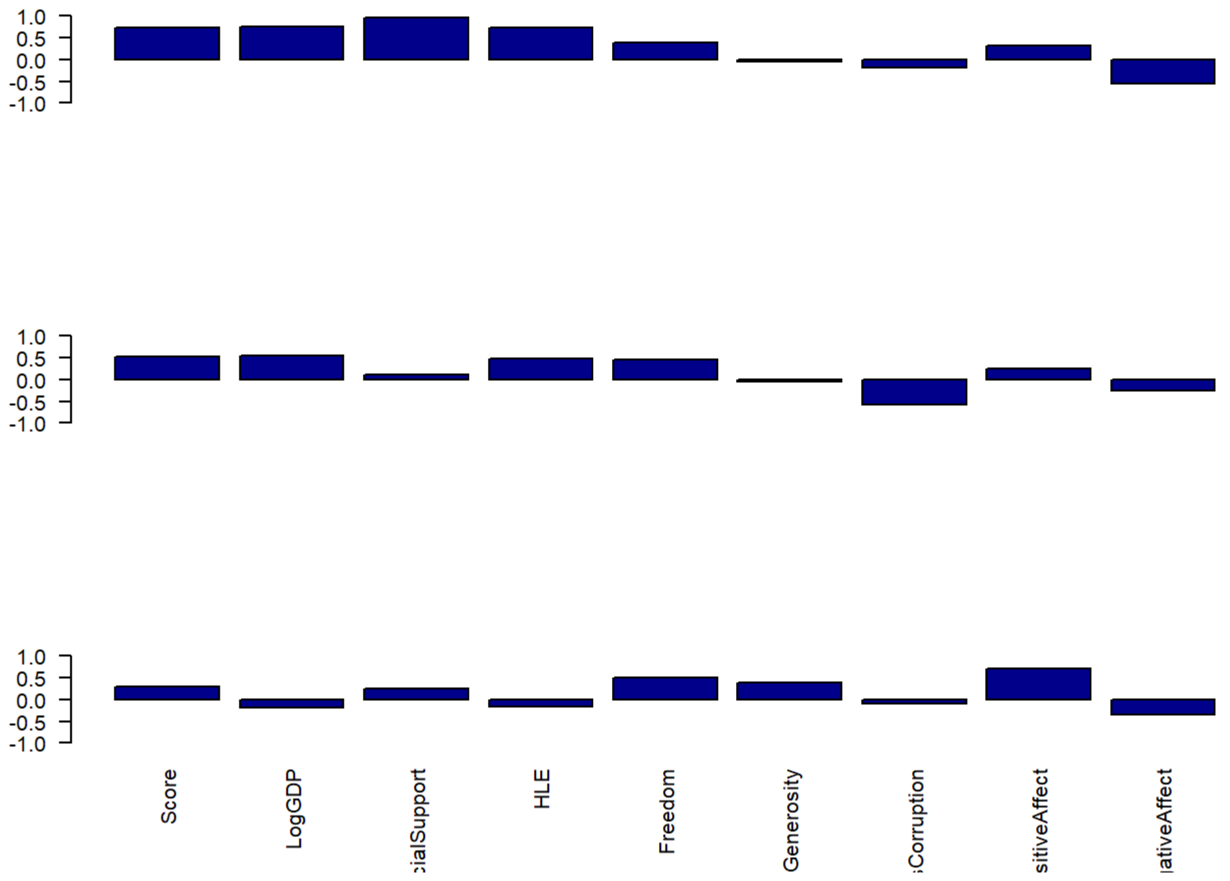
```
## Error : The fig.showtext code chunk option must be TRUE
```

```r
barplot(fa2$loadings[,2], names=F, las=2, col="darkblue", ylim = c(-1, 1))
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
barplot(fa2$loadings[,3], las=2, col="darkblue", ylim = c(-1, 1))
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



Using the previous plots and dataframe we can associate to the original variavles to the factors Factor1: “Score”, “LogGDP”, “SocialSupport”, “HLE” and negatively “NegativeAffect” Factor2: Negative “PerceptionsCorruption” Factor3: “Freedom”, “Generosity”, “PositiveAffect”

With this, we can interpret: - Factor 1 appears to represent a factor related to overall well-being, happiness, and life quality. - Factor 2 appears to represent a factor related to the absence of perceptions of corruption. - Factor 3 appears to represent a factor related to positive social attributes or attitudes.

In addition to that if we look at uniqueness, we see that “SocialSupport” has the lowest value: 0.01 therefore its variance is really well explained by the factors. On the other hand, “Generosity” has the highest value: 0.834, which means that the factors don’t make a good job explaing its variance.

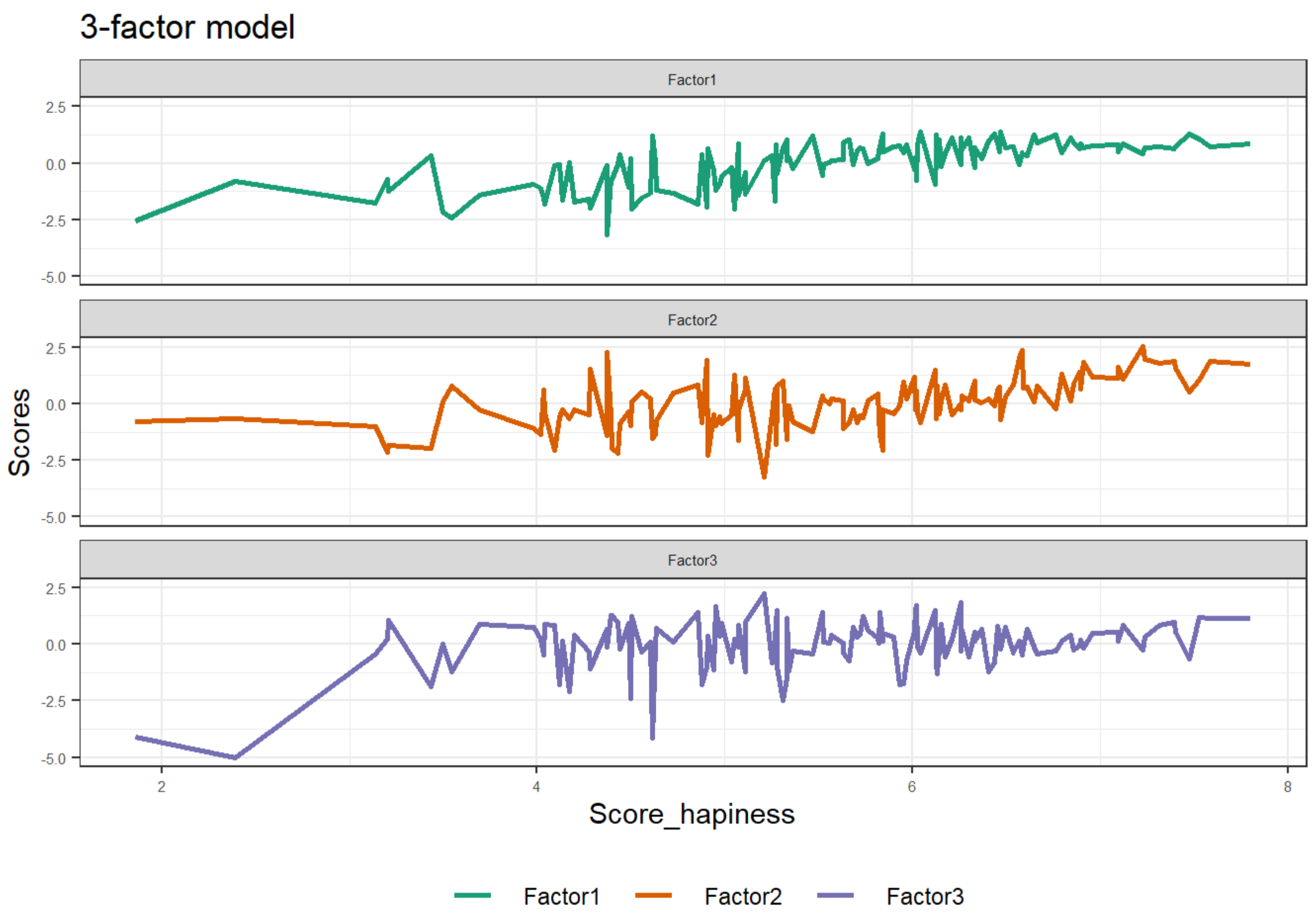
5.2 Scores

Factor scores are values that represent the degree to which each observation in a dataset exhibits certain characteristics or behaviors associated with underlying latent factors.

```
factor.df <- data.frame(Score_hapiness = data2$Score, fa2$scores)

# Reshape the data to long format using gather
```

```r
factor.df <- factor.df %>%
  gather("factor", "score", -Score_hapiness)

# Create the ggplot visualization
factor.df %>%
  ggplot(aes(x = Score_hapiness, y = score, color = factor)) +
  geom_line(size = 1) +
  theme_bw() +
  theme(legend.position = "bottom") +
  scale_color_brewer(palette = "Dark2") +
  facet_wrap(~factor, ncol = 1) +
  labs(title = "3-factor model", x = "Score_hapiness", y = "Scores", col = "")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



Factor 1 is associated with well-being. We see a positive trend for Factor 1, this suggests that higher “Score_happiness” values are associated with higher well-being.

Factor 2 is associated absence of perceptions of corruption. We see a positive trend for Factor 2, this suggests that higher “Score_happiness” values are associated no perception of corruption.

Factor 3 is associated with positive social attributes. We see a positive trend for Factor 3, this suggests that higher “Score_happiness” values are associated with higher positive social attributes.

6 Clustering

(Based on labs of clustering)

Clustering is a machine learning and data analysis technique that involves grouping similar data points or objects together based on certain characteristics or features. The goal of clustering is to discover underlying patterns or structures in data, which can be helpful for various purposes, such as data exploration, pattern recognition, and decision-making.

In our case, clustering can help segment countries into distinct groups based on their happiness-related characteristics.

6.1 K-Means

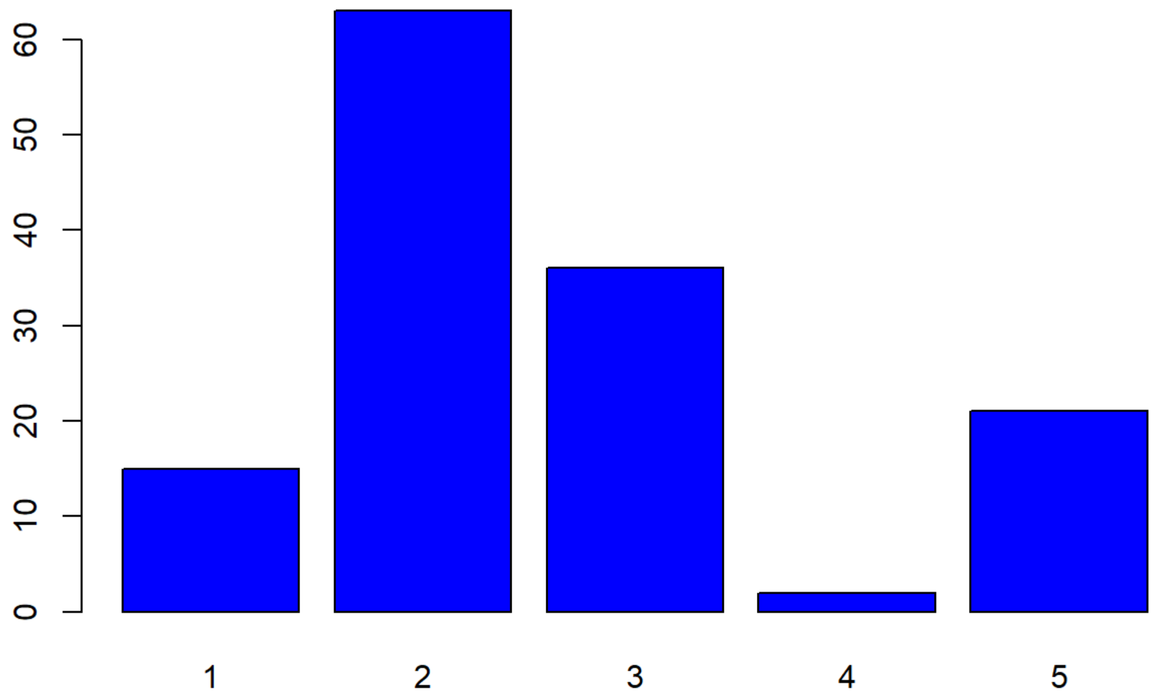
We are going to try the number of clusters as geographical regions as they might naturally lead to clusters. It could be a possibility that countries within the same continent tend to have similar happiness profiles. However, we will compute the number of clusters in a more sophiticated way later.

Here, we apply the K-Means clustering algorithm to our data. We scales the data. centers=k specifies the number of clusters, and nstart=1000 sets the number of times the algorithm will attempt to find the optimal cluster centers. The results are stored in the fit object. “groups” indicates which cluster each data point belongs to.

Then, we create a bar plot showing the distribution of data points in each cluster. The table(groups) function counts the number of data points in each cluster,

```r
set.seed(123) # In order to obtain the same clusters for explanations
k = 5
# nstart: make the random process nstart times,
fit = kmeans(scale(data2), centers=k, nstart=1000)
groups = fit$cluster

barplot(table(groups), col="blue")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



```
# Heuristic because we select the first centroids randomly
```

We notice an imbalance in the sizes of the clusters

6.1.1 Interpretation of centers

1. We are visualizing the cluster centers, which represent the mean values of the observations within each cluster. In order to compare and analyze the cluters it is better to plot them together.

```
centers=fit$centers
# Centers are the mean value of the observations in the group. It contains the mean for each f
eature
par(mfrow=c(2,3))

barplot(centers[1,], las=2, col="lightblue", main = "Cluster 1")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
barplot(centers[2,], las=2, col="lightblue",main = "Cluster 2")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```r
barplot(centers[3,], las=2, col="lightblue", main = "Cluster 3")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

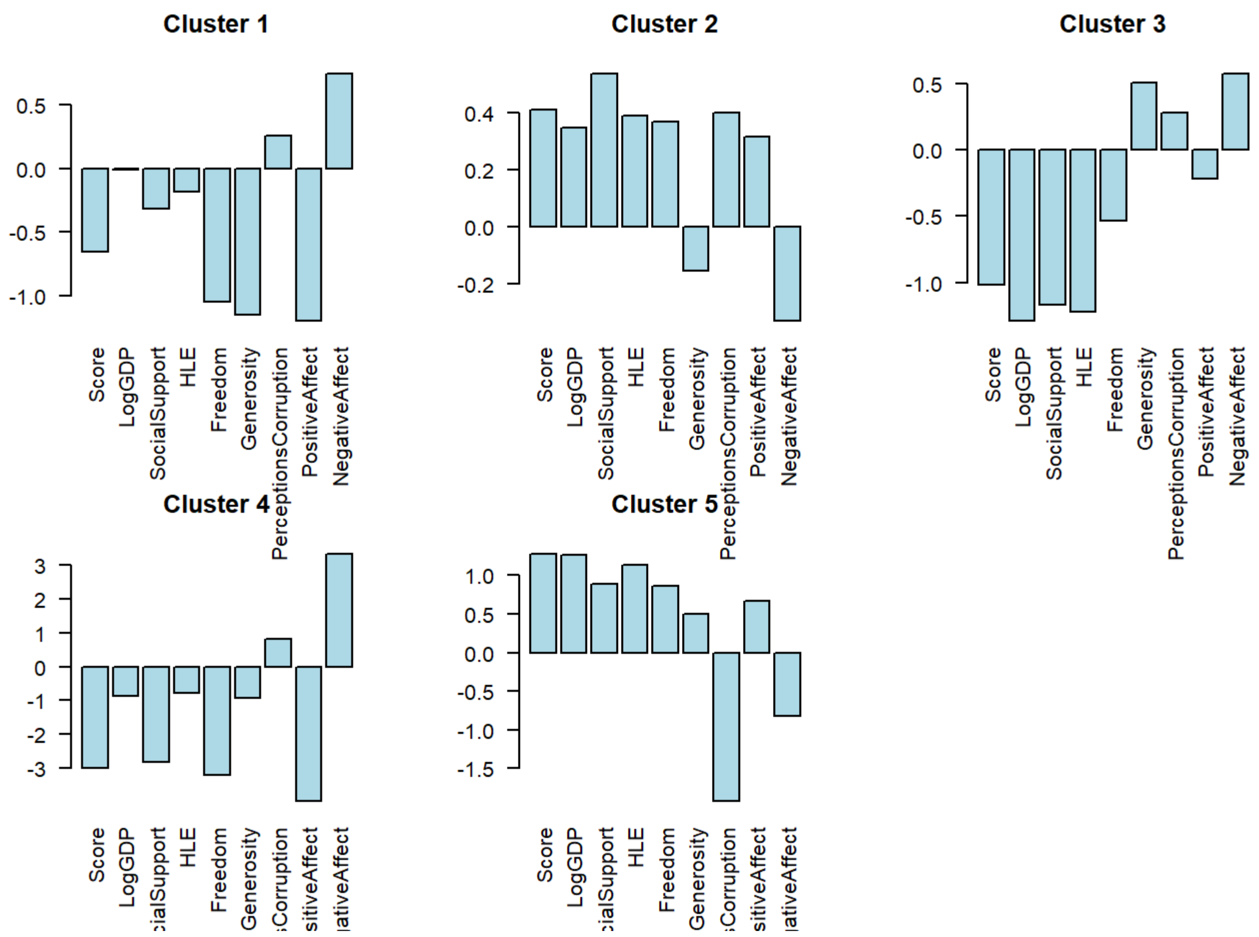
```r
barplot(centers[4,], las=2, col="lightblue", main = "Cluster 4")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```r
barplot(centers[5,], las=2, col="lightblue", main = "Cluster 5")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```r
# Difficult to understand as I could not say in which group is Spain
```



These visualizations can provide insights into how each cluster differs in terms of the feature values. We can compare the bar plots to understand the characteristics and differences of the clusters. However, it is not very easy to understand as we cannot see which cluster corresponds to each country.

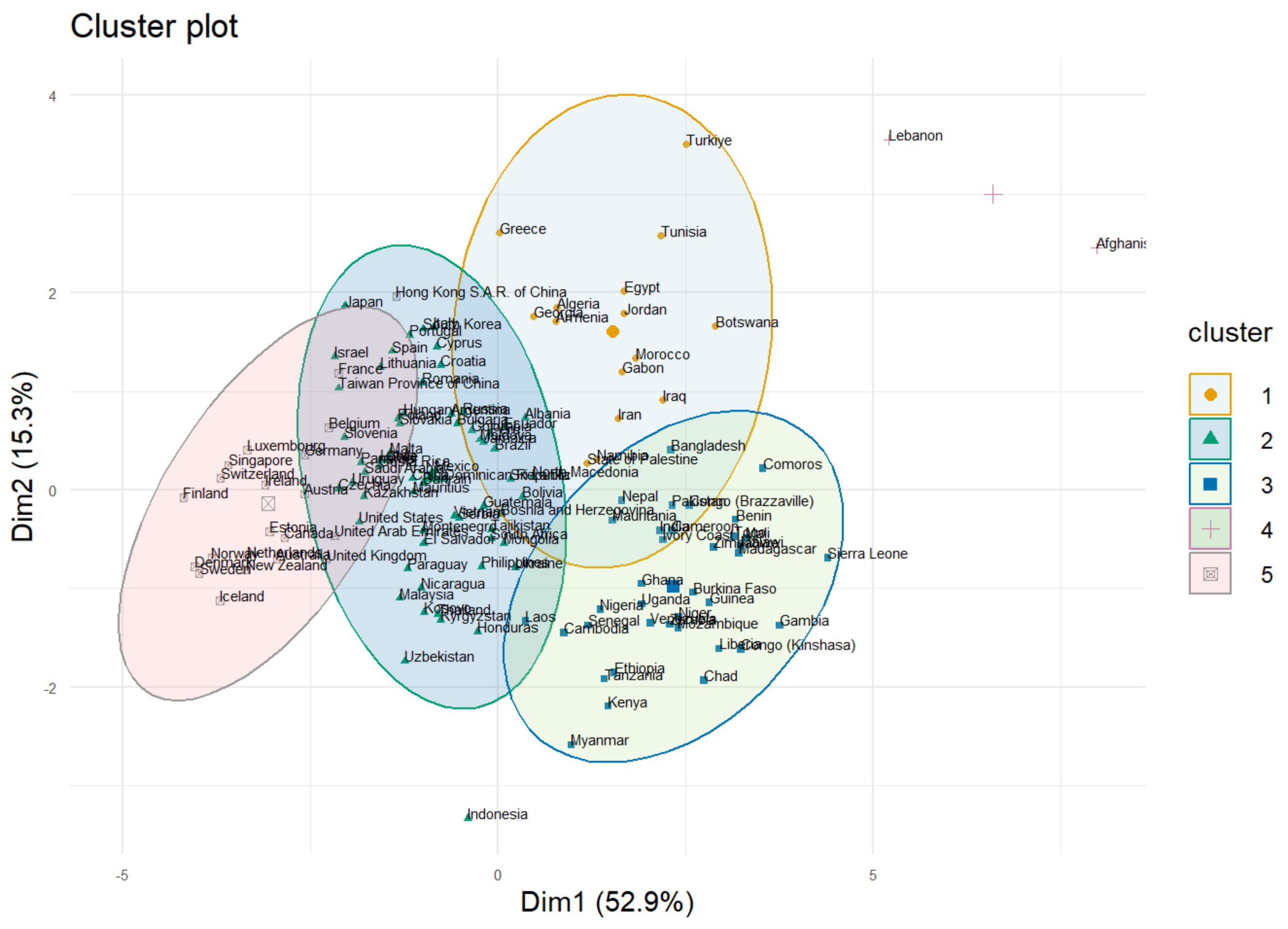
(NOOO: We can divide our 5 clusters in 2 types:- “NegativeAffect” and “PerceptionsCorruption” affect positively: Cluster1:)

6.1.2 Clusplot

For this reason, we make a plot to visualize our clustered data in a 2D PCA space, with points representing countries and colors indicating the clusters.

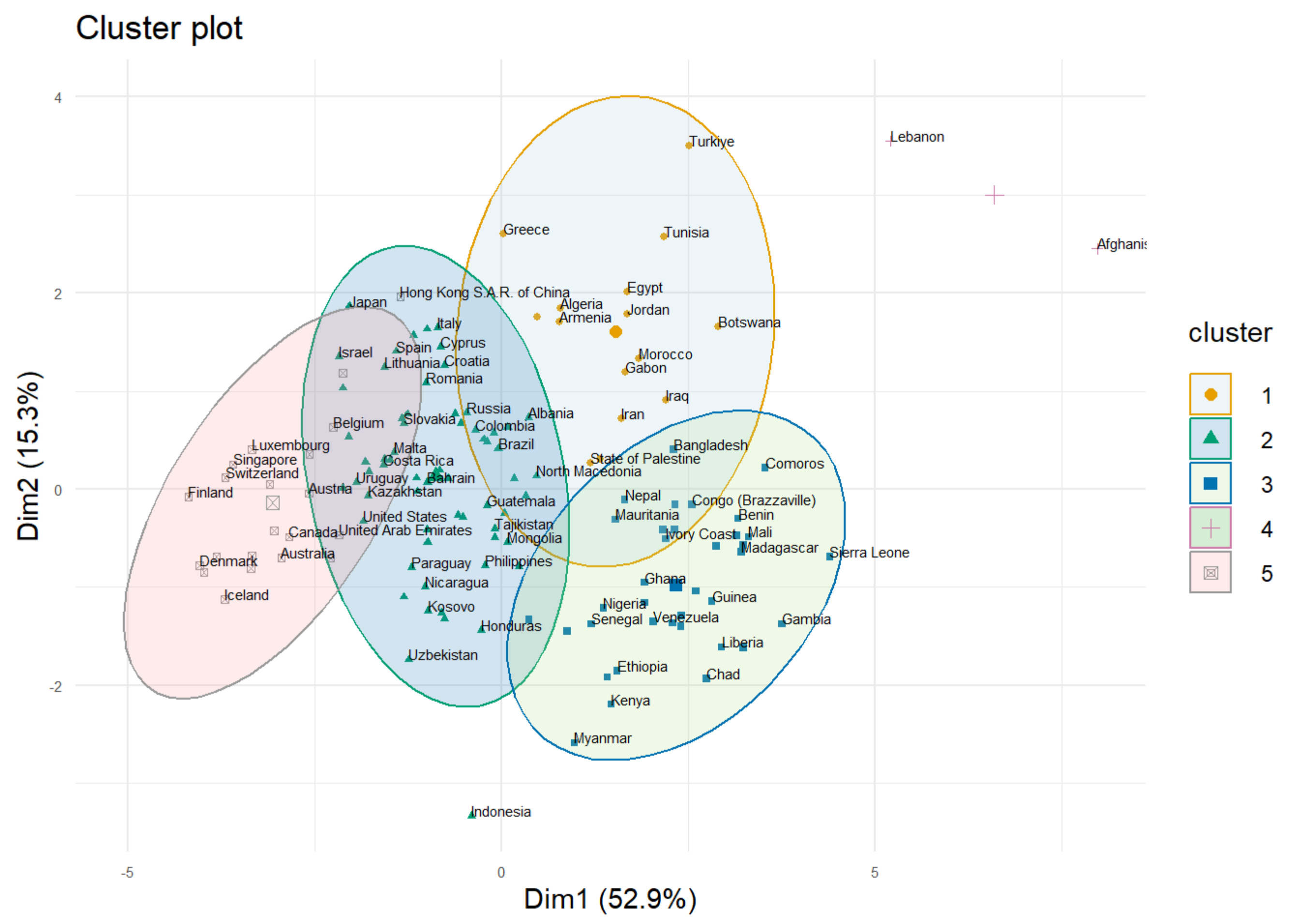
```
# clusplot
#options(repr.plot.width = 15, repr.plot.height = 6)


fviz_cluster(fit, data = data2, geom = c("point"),ellipse.type = 'norm', pointsize=1)+
  theme_minimal()+geom_text(label=country_names,hjust=0, vjust=0,size=2,check_overlap = F)+sca
le_fill_brewer(palette="Paired")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



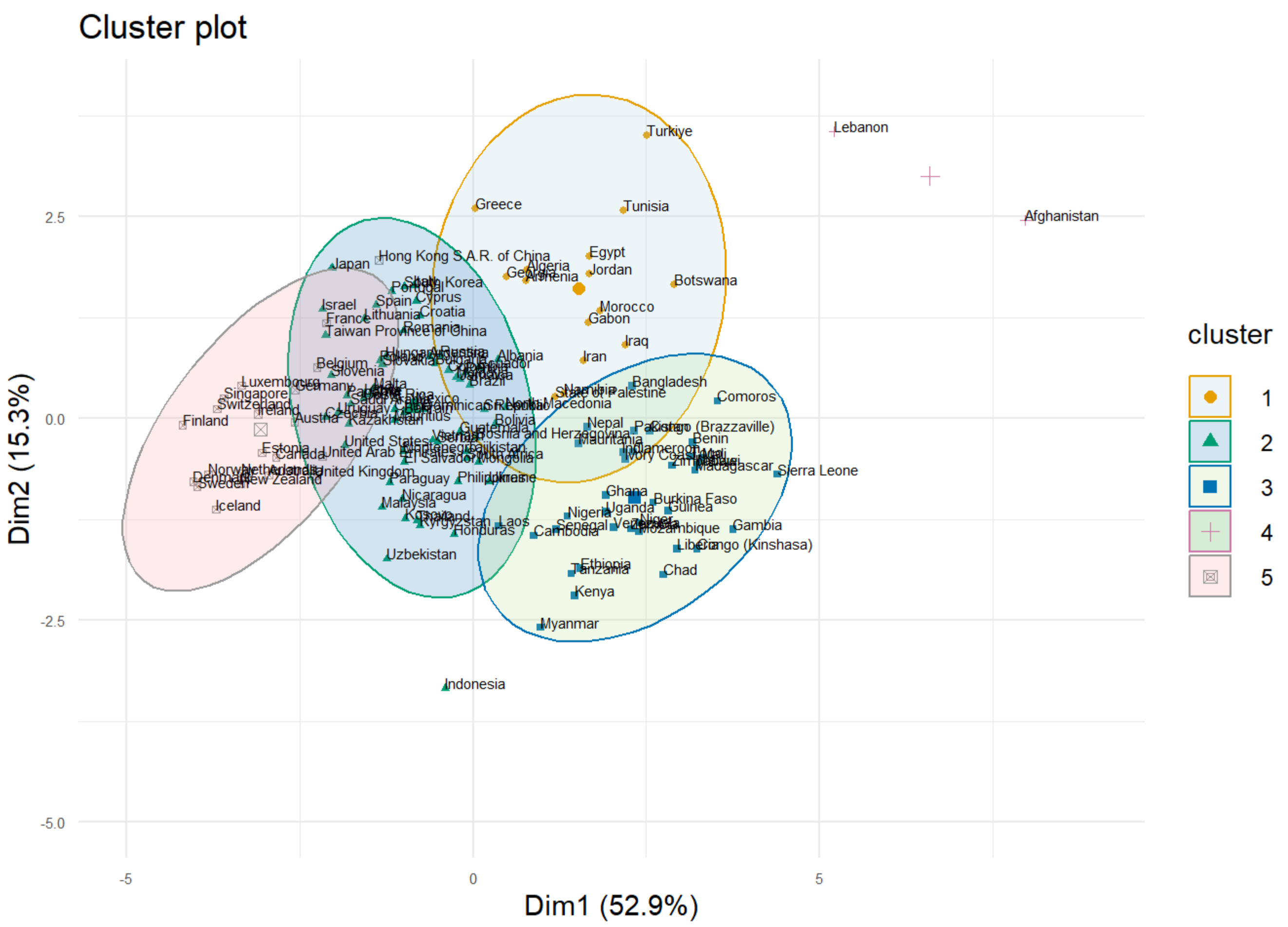
```
# Let's make another plot with check_overlap = T, which is not going to display all countries,
 but it will more understandable

fviz_cluster(fit, data = data2, geom = c("point"),ellipse.type = 'norm', pointsize=1)+
  theme_minimal()+geom_text(label=country_names,hjust=0, vjust=0,size=2,check_overlap = T)+sca
le_fill_brewer(palette="Paired")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



```
# Make a little zoom
fviz_cluster(fit, data = data2, geom = c("point"),ellipse.type = 'norm', pointsize=1)+
  theme_minimal()+geom_text(label=country_names,hjust=0, vjust=0,size=2,check_overlap = F)+sca
le_fill_brewer(palette="Paired")  + coord_cartesian(xlim = c(-5, 9), ylim = c(-5, 4))
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



We need to remember the interpretation of the PCs. PC1 represent the “unhapiness of the countries” PC2 decreases with “Generosity” and “PositiveAffect” while it increases with “LogGDP” and ” HLE”.

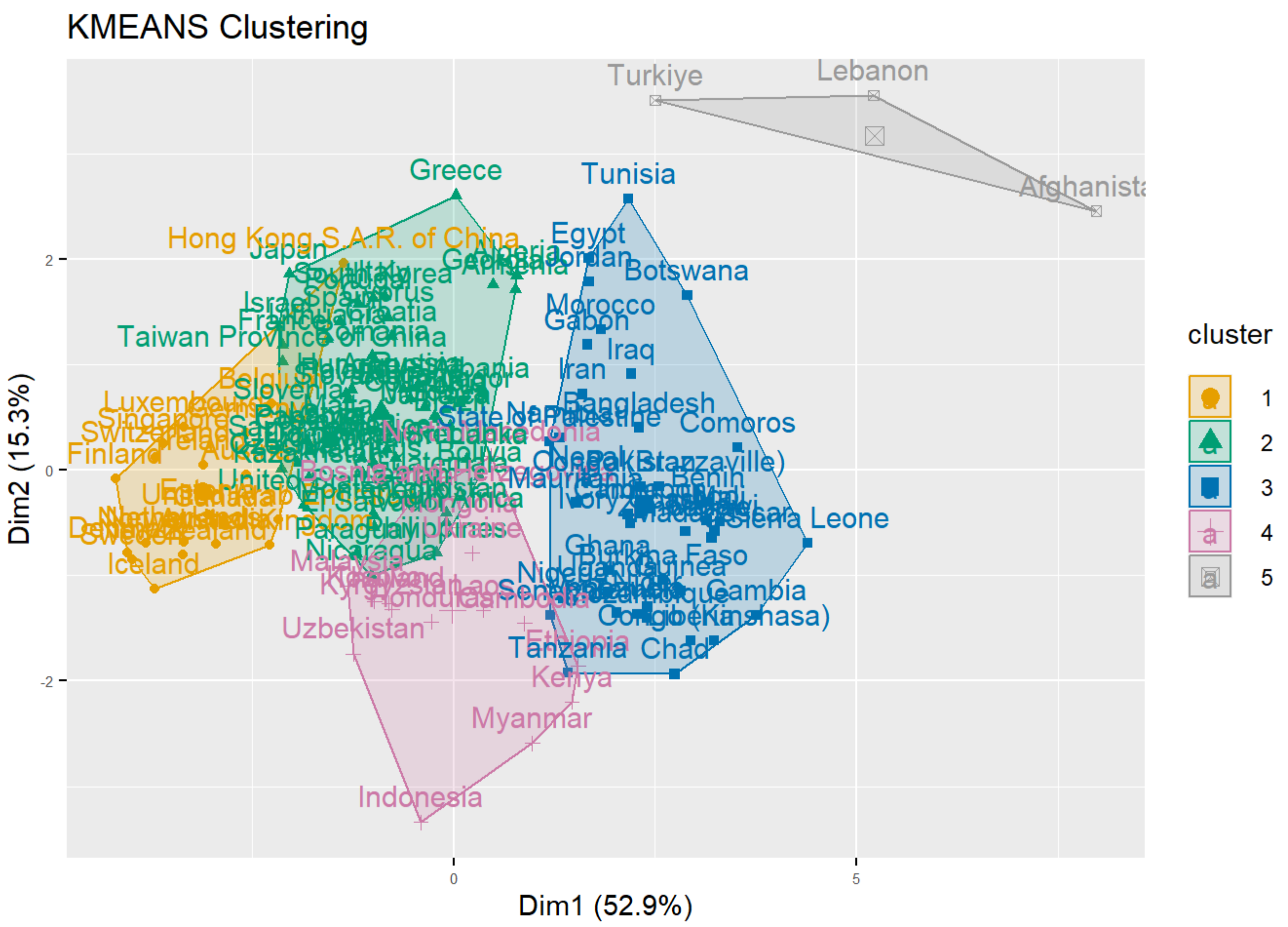
In cluster 5, the red one, all variables positively contribute to the cluster except from “NegativeAffect” and “PerceptionsCorruption”. So, in this cluster we have the happiest and with better situation countries as Finland.

On the other hand, cluster 4 is the oppostite, as all variables negatively contribute to the cluster except from “NegativeAffect” and “PerceptionsCorruption”. So, in this cluster we have the unhappiest and with worse situation countries as Afghanistan

An alternative approach is to leverage the eclust function provided by the factoextra package. This function offers a wide range of clustering methods (such as k-means, pam, hierarchical clustering, and more), as well as various distance metrics and linkage methods to suit your specific clustering needs.

```
data_name = data2
rownames(data_name) <- country_names
fit.kmeans <- eclust(data_name, "kmeans", stand=TRUE, k=5)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



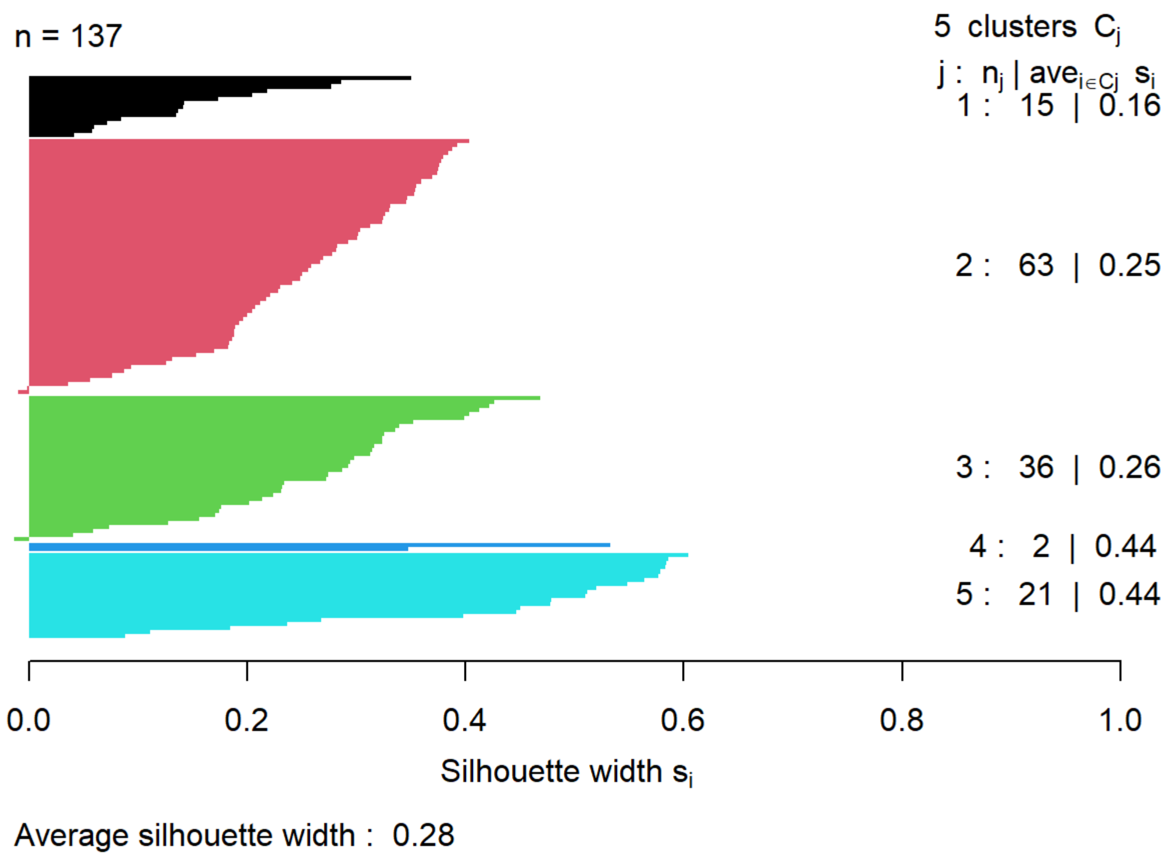
6.1.3 Silhouette plot

The silhouette plot is a graphical representation of the silhouette coefficients for each data point in the clusters (measure of how similar a data point is to its own cluster compared to other clusters). It can help you assess the quality of the clustering by visualizing how well-separated and consistent the clusters are.

A high silhouette coefficient suggests a good cluster assignment. A silhouette coefficient near 0 suggests that the data point is on or very close to the decision boundary between two neighboring clusters. A negative silhouette coefficient indicates that the data point may have been assigned to the wrong cluster.

```
# Distances between the points and the clusters
d <- dist(scale(data2), method="euclidean")
sil = silhouette(groups, d)
plot(sil, col=1:5, main="", border=NA)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



```
summary(sil)
```

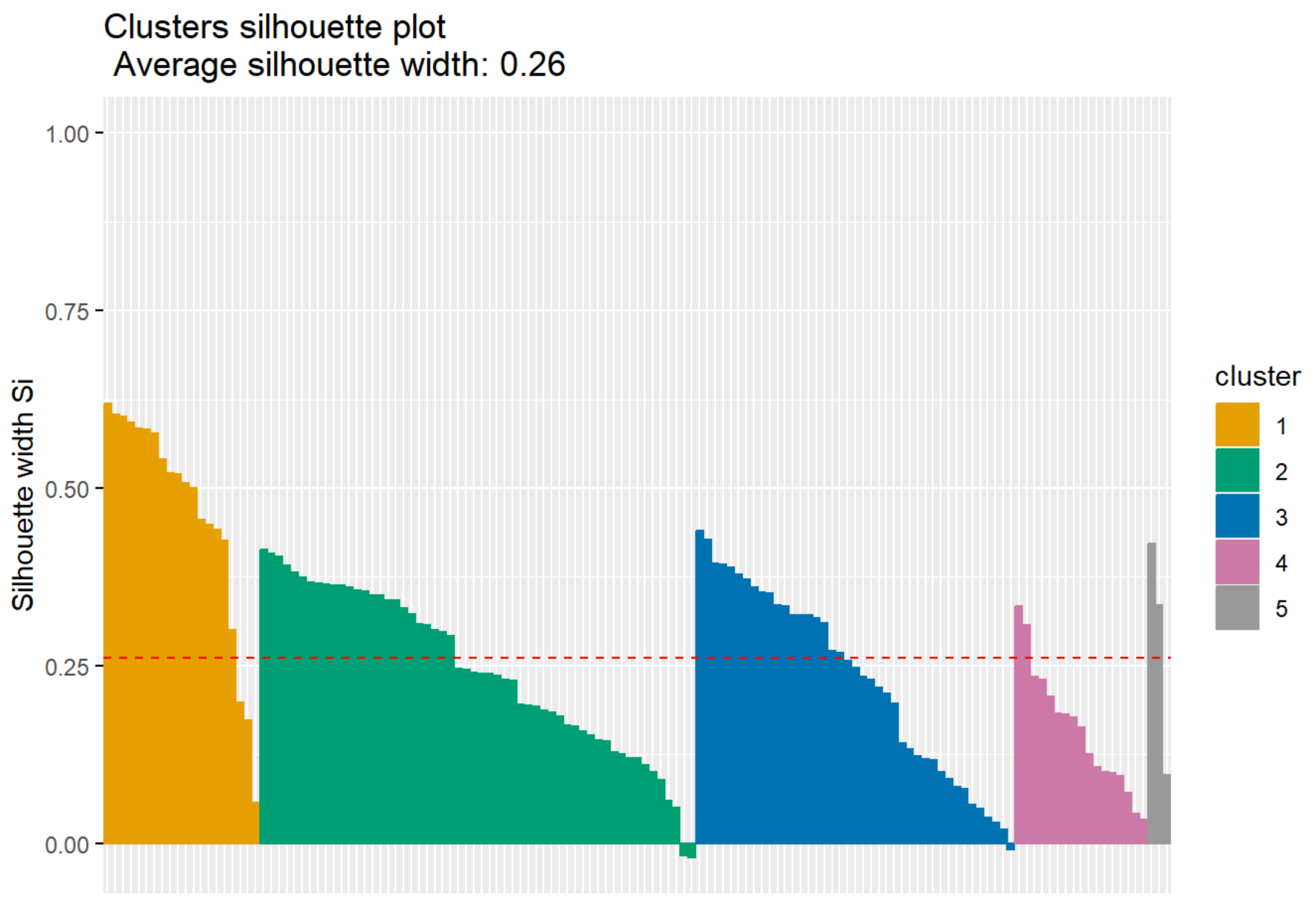
```
## Silhouette of 137 units in 5 clusters from silhouette.default(x = groups, dist = d) :
##  Cluster sizes and average silhouette widths:
##        15        63        36         2        21 
## 0.1586010 0.2516555 0.2636962 0.4403050 0.4429850 
## Individual silhouette widths:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## -0.01365  0.18472  0.27654  0.27671  0.35510  0.60388
```

The cluster with the highest average silhouette width (0.44 C5) is the most well-separated and cohesive, while the cluster with the lowest average silhouette width (0.15 C1) is relatively less well-separated. The clusters with smaller average silhouette width (closer to 0) suggests that the data points within the cluster may be closer to the decision boundary between clusters.

```
# the same with factoextra
fviz_silhouette(fit.kmeans)
```

```
##   cluster size ave.sil.width
## 1       1   20          0.46
## 2       2   56          0.24
## 3       3   41          0.23
## 4       4   17          0.16
## 5      5   3        0.28
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



The negative values (close to -1) means that a data point is more similar to a neighboring cluster than its own, suggesting a potential misclassification.

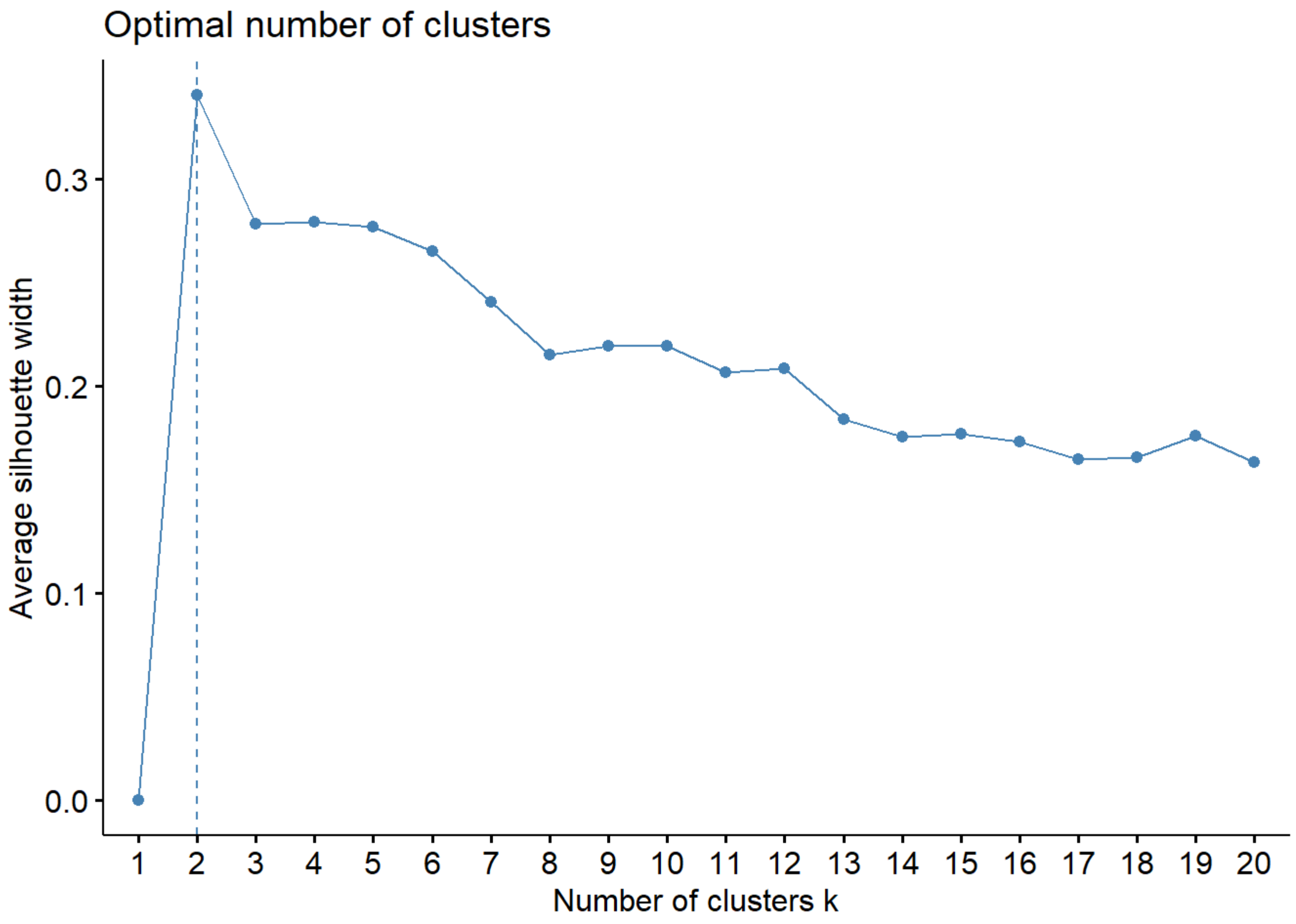
6.1.4 Number of clusters

The choice of the number of clusters (k) is a crucial decision in clustering analysis, as it significantly impacts the quality and interpretability of the results.

The function **fviz_nbclust** will generate a plot or print a table showing the silhouette score for a range of values of k (the number of clusters). The optimal number of clusters is often associated with the peak or plateau in the silhouette score, where the clusters are well-separated and internally cohesive.

```
fviz_nbclust(scale(data2), kmeans, method = 'silhouette', k.max = 20, nstart = 1000)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

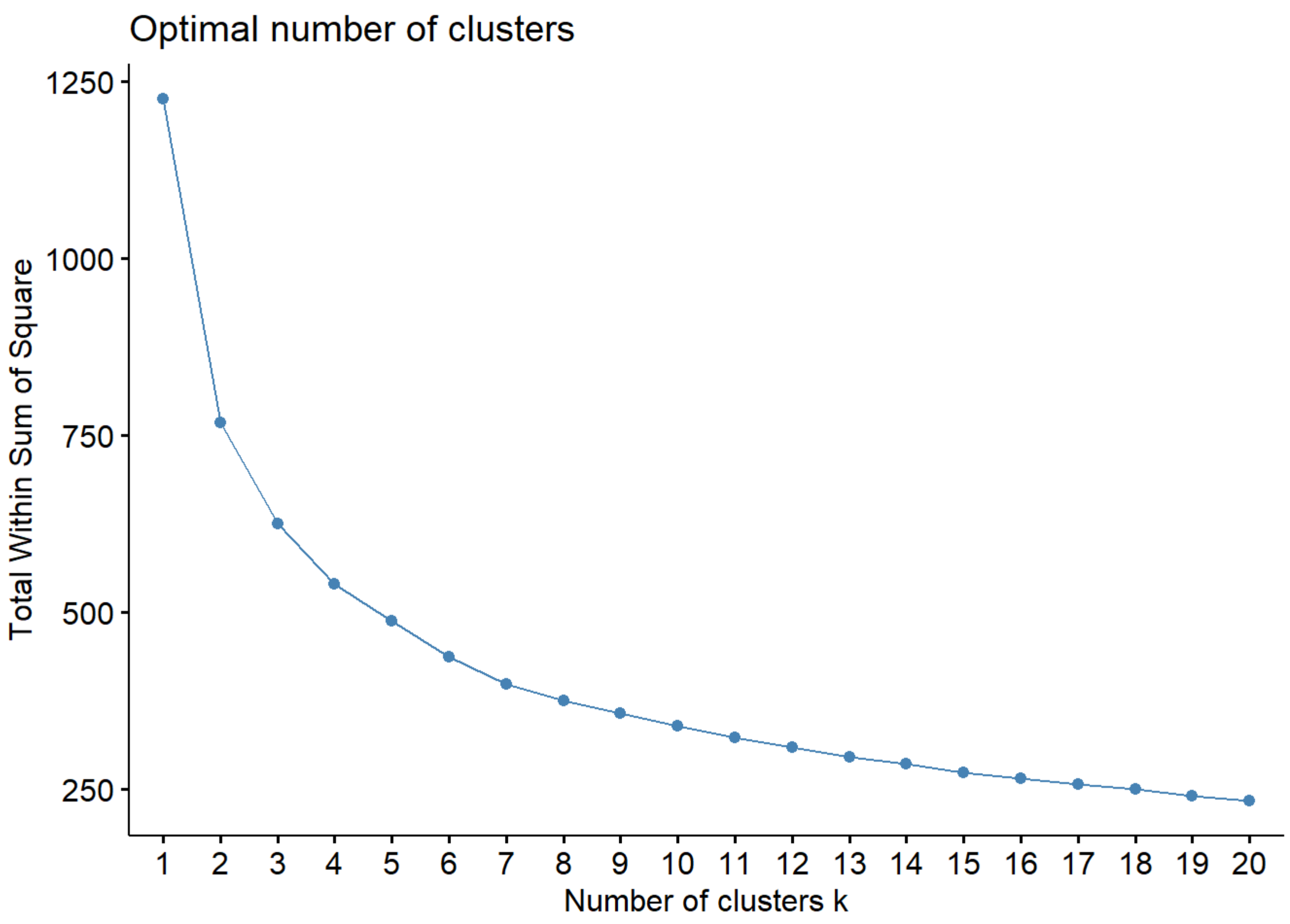


(2 clusters)

Within-Cluster Sum of Squares (WSS) criterion. The WSS measures the total variation within each cluster, and finding the "elbow point" in the plot of the WSS can help determine the optimal number of clusters.

```
fviz_nbclust(scale(data2), kmeans, method = 'wss', k.max = 20, nstart = 1000)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

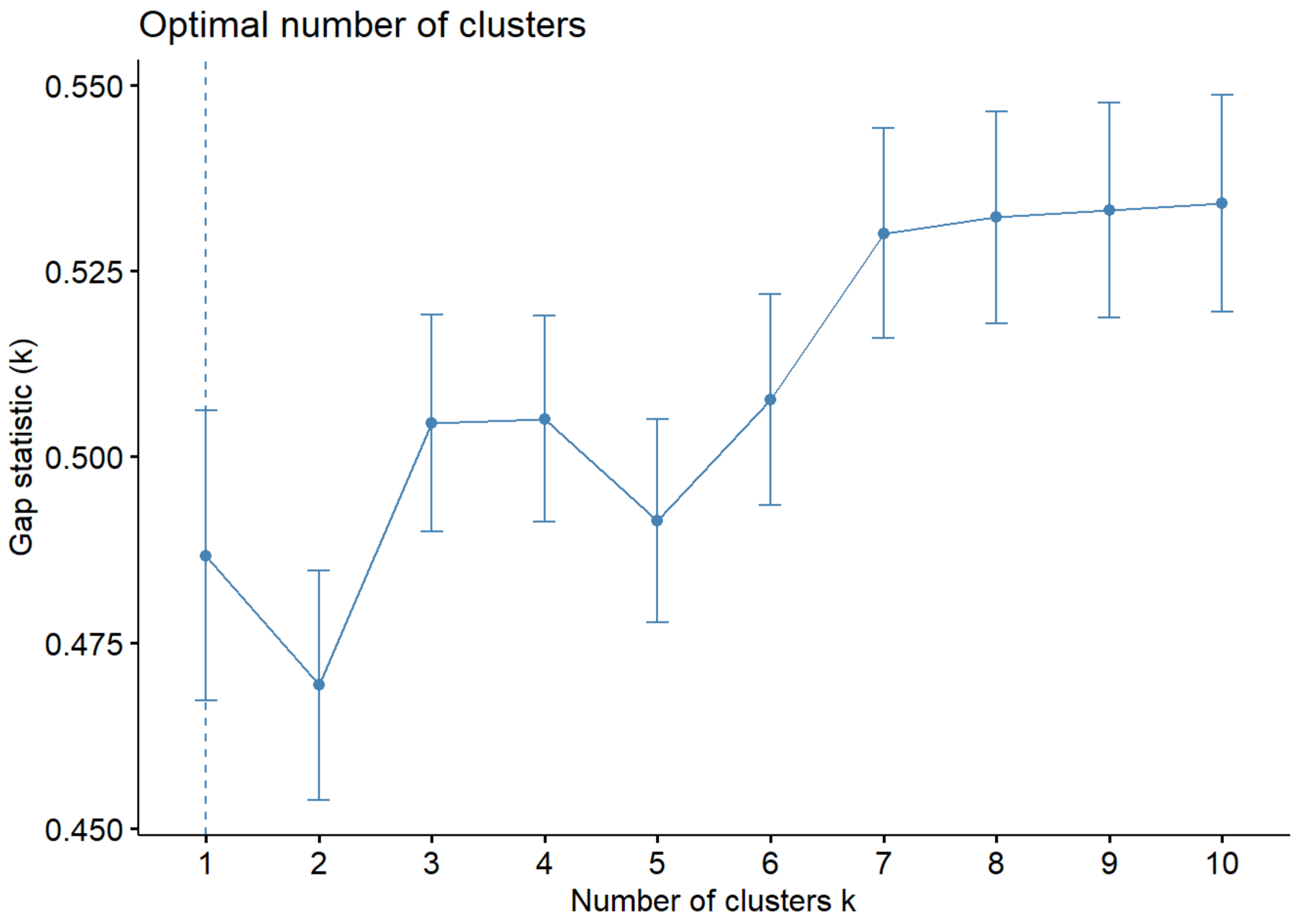


(2 or 3 clusters)

Gap Statistic and the reference distribution for a range of values of k (the number of clusters). The optimal number of clusters is often associated with the value of k that maximizes the Gap Statistic while taking into account the reference distribution.

```
fviz_nbclust(scale(data2), kmeans, method = 'gap_stat', k.max = 10, nstart = 100, nboot = 500)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

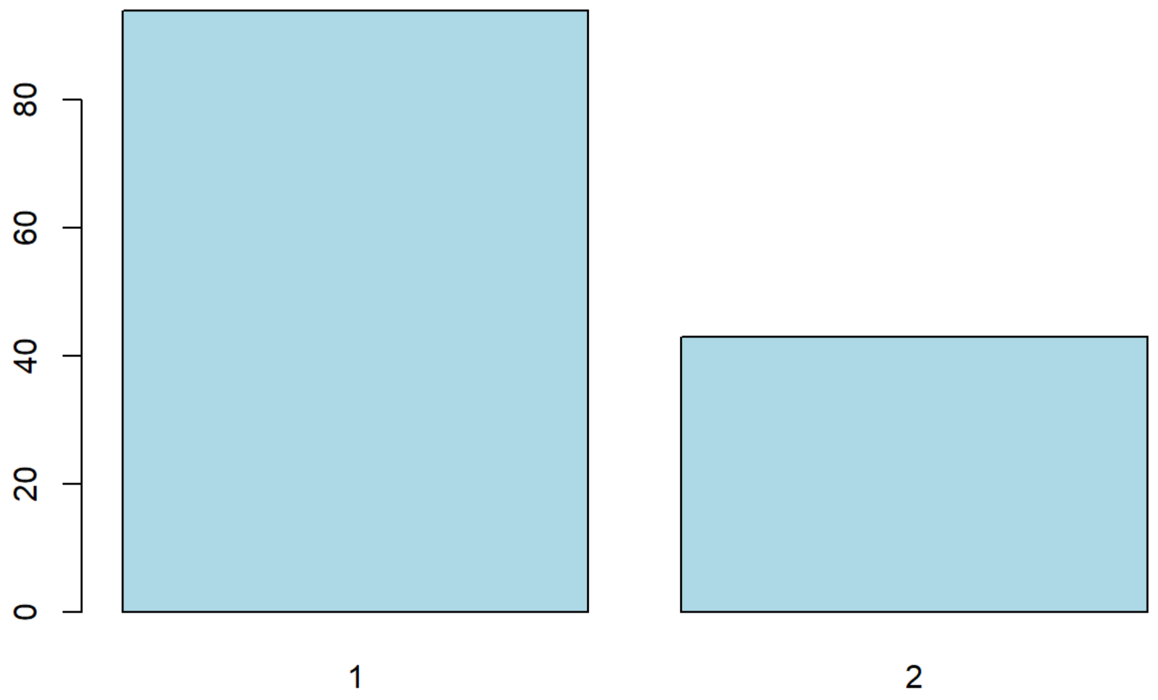


Maybe 2 or 3 clusters

Perform clustering with the best number of clusters. k = 2

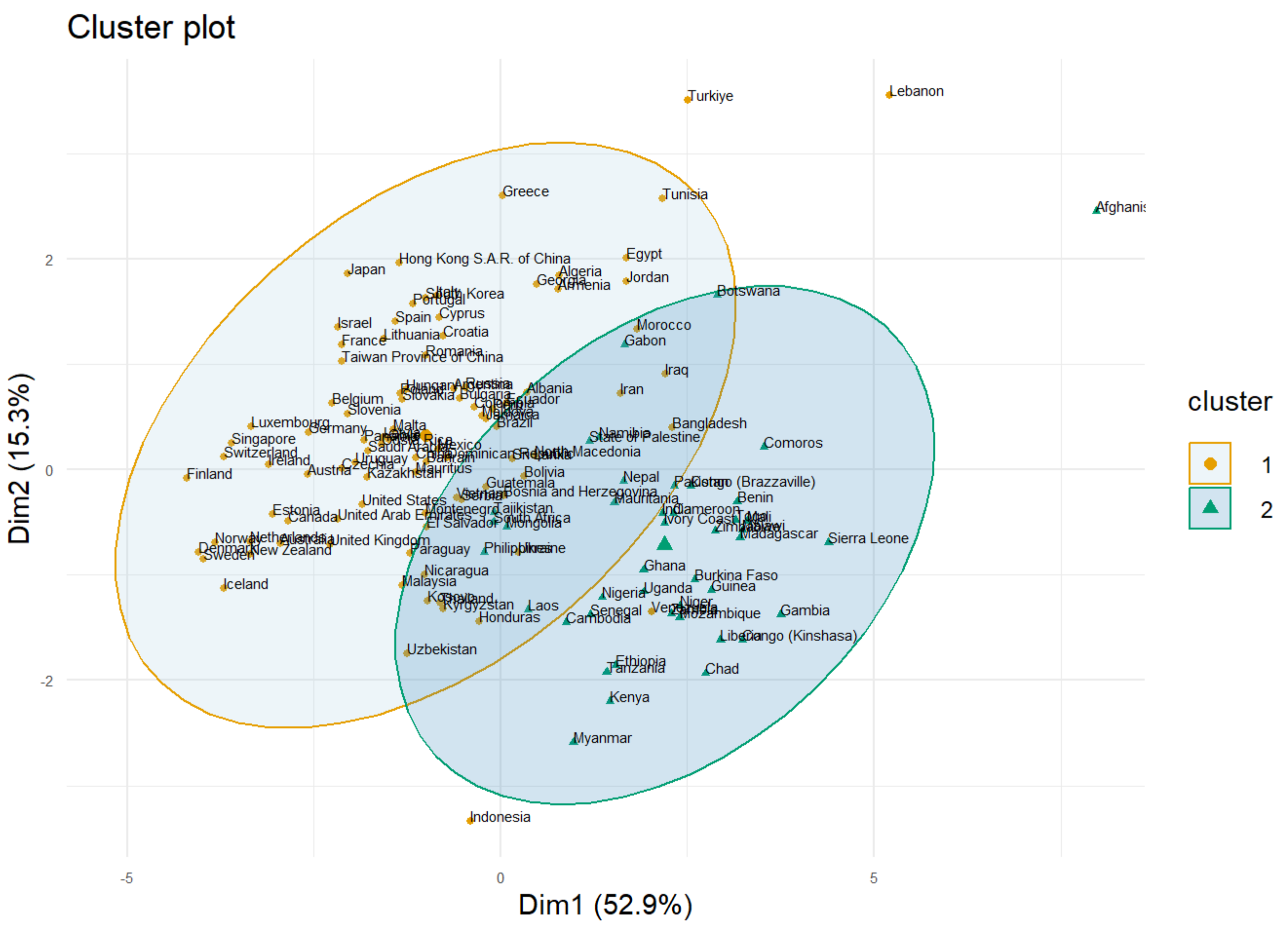
```r
fit2 = kmeans(data2, centers = 2, nstart = 25)

# Is the classification more balanced than before?
groups2 = fit2$cluster
barplot(table(groups2), col="#ADD8E6")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



```
# The clusplot can also be checked, observing that, indeed, a more balanced
# classification is obtained with the number of clusters equal to 2.
fviz_cluster(fit2, data = data2, geom = c("point"),ellipse.type = 'norm', pointsize=1)+
  theme_minimal()+geom_text(label=country_names,hjust=0, vjust=0,size=2,check_overlap = F)+sca
le_fill_brewer(palette="Paired")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

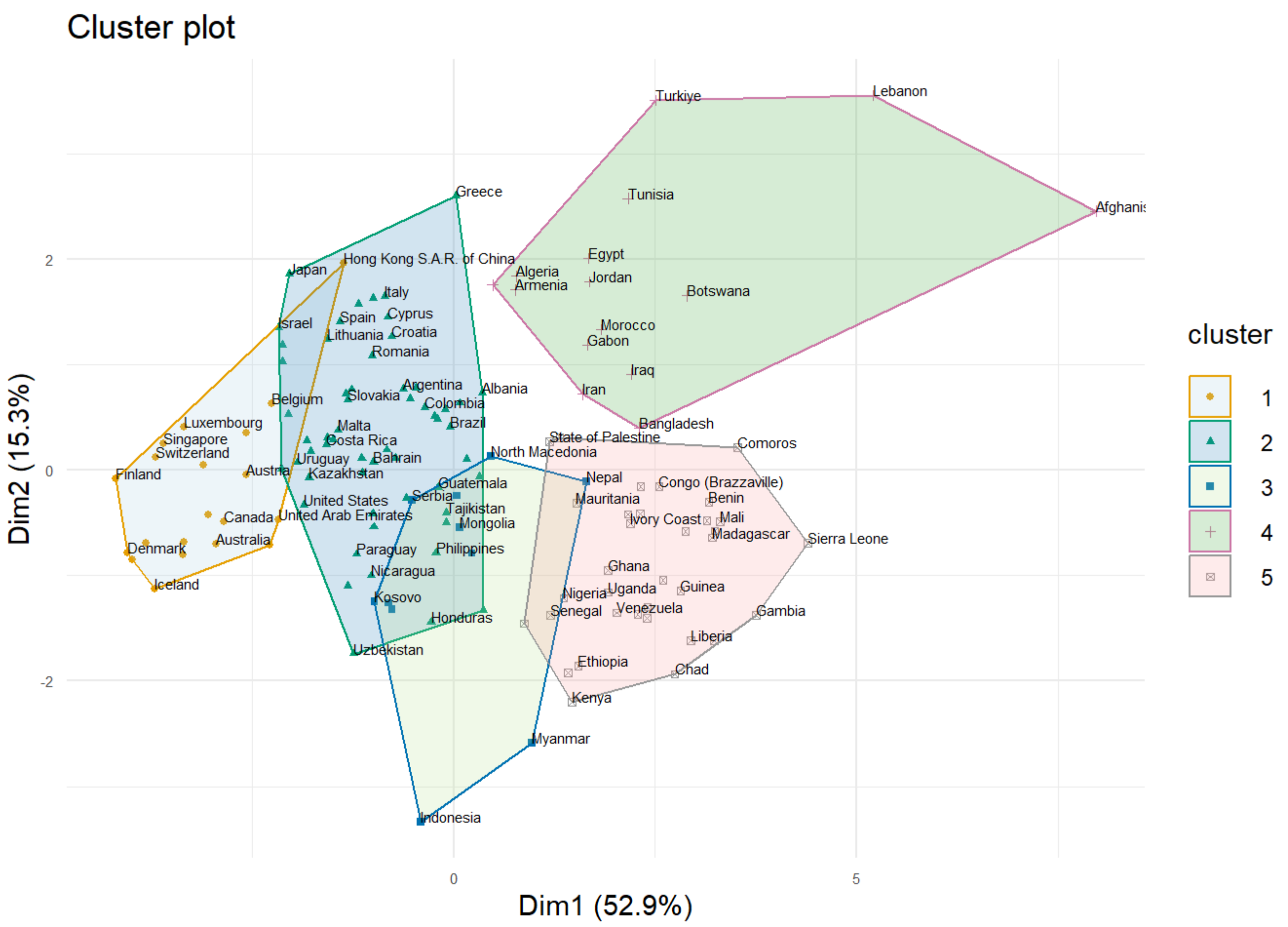


6.2 PAM

Partitioning Around Medoids. The key difference between PAM and k-means is that PAM uses medoids as cluster centers, which are actual data points from the dataset (countries in our case), rather than the means (centroids) as in k-means. This makes PAM more robust to outliers and noise in the data.

```
fit.pam <- eclust(data2, "pam", stand=TRUE, k=5, graph=F)

fviz_cluster(fit.pam, data = data2, geom = c("point"), pointsize=1)+
  theme_minimal()+geom_text(label=country_names,hjust=0, vjust=0,size=2,check_overlap = T)+sca
le_fill_brewer(palette="Paired")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

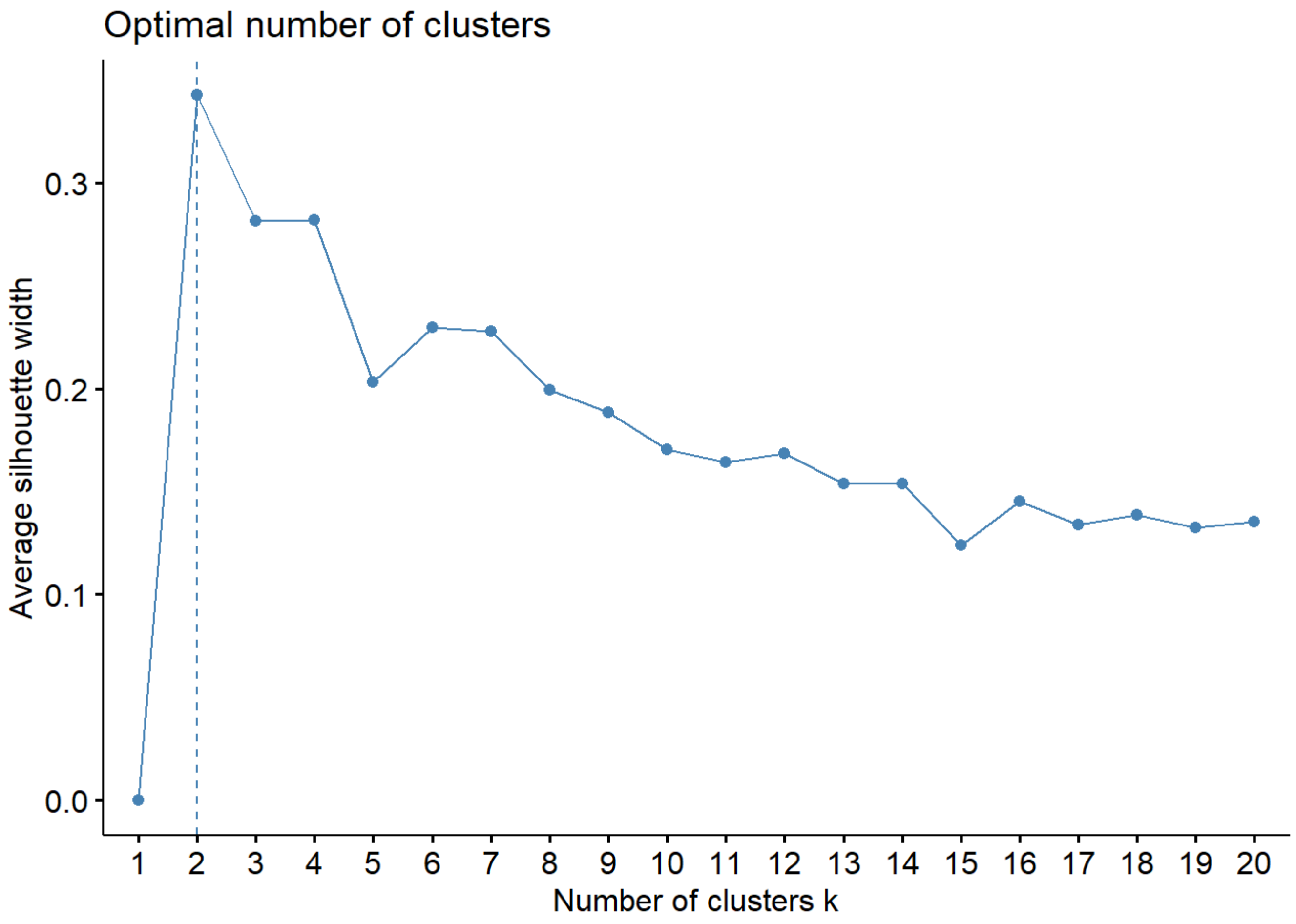


We obtain different cluster that the ones from k-means

6.2.1 Number clusters PAM

```
fviz_nbclust(scale(data2), pam, method = 'silhouette', k.max = 20, nstart = 100)
```

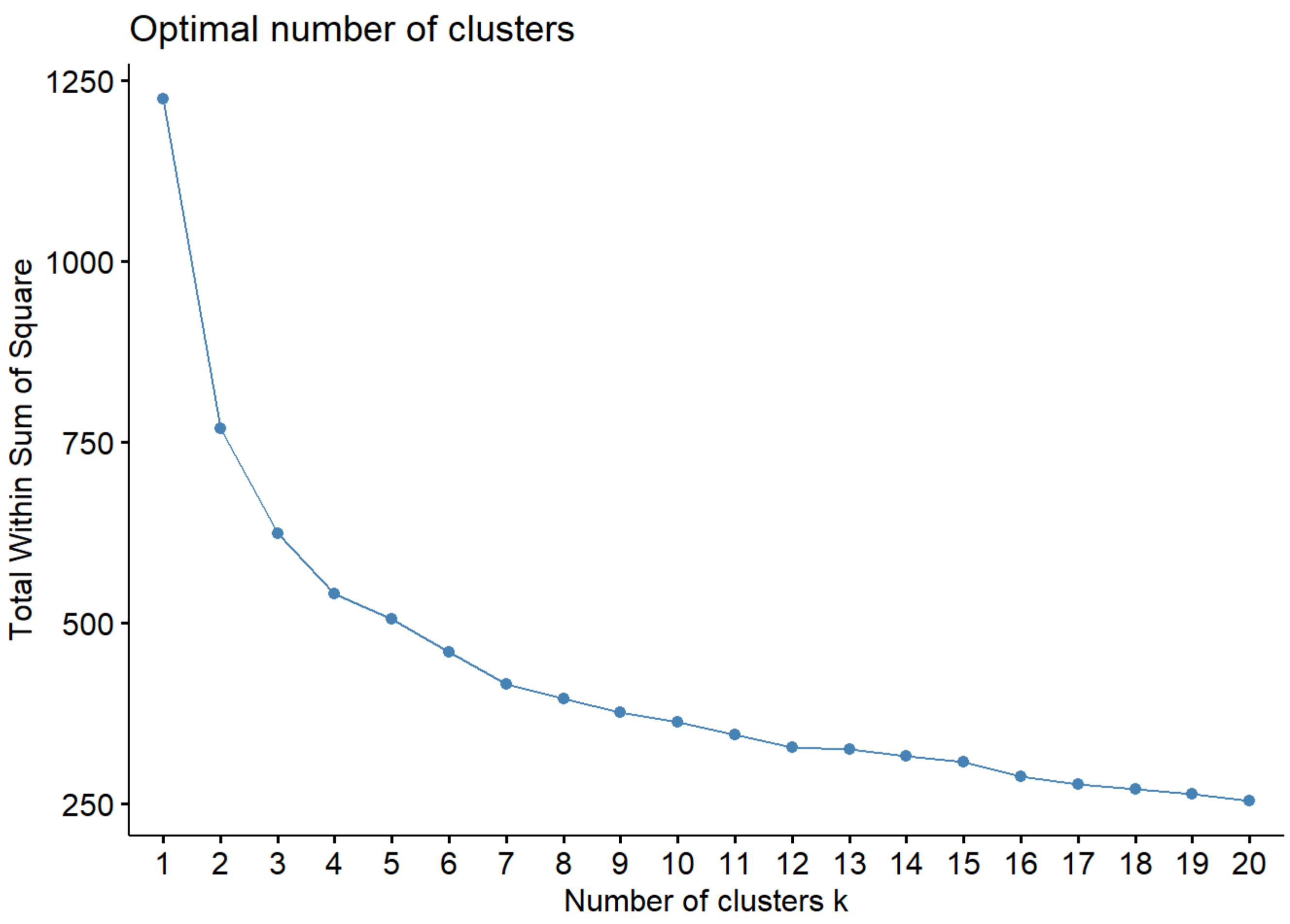
```
## Error : The fig.showtext code chunk option must be TRUE
```



(2 clusters)

```
fviz_nbclust(scale(data2), pam, method = 'wss', k.max = 20, nstart = 100)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

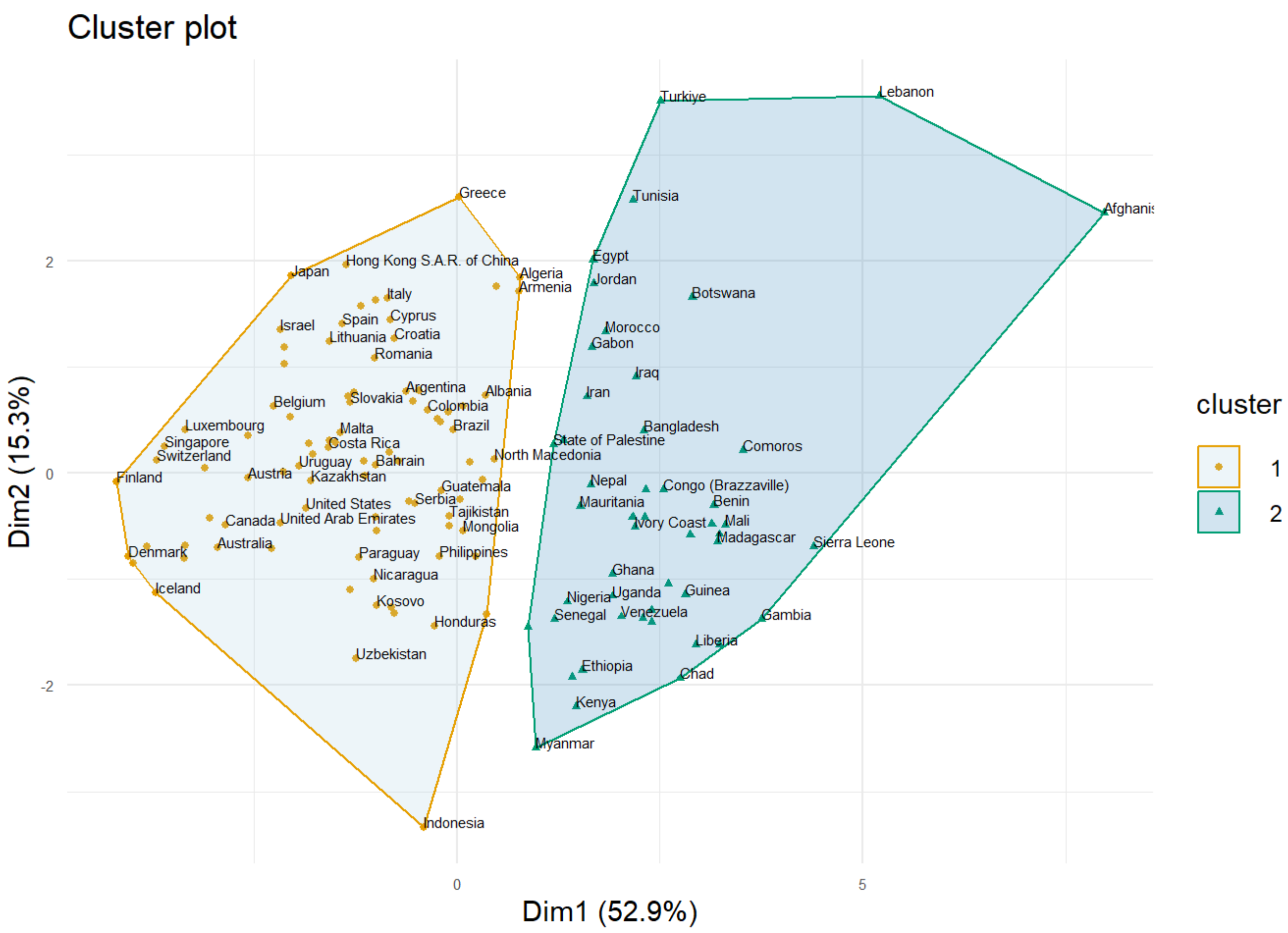


(2 or 3 clusters)

Perform clustering with the best number of clusters. k = 2 PAM

```
fit.pam2 <- eclust(data2, "pam", stand=TRUE, k=2, graph=F)

fviz_cluster(fit.pam2, data = data2, geom = c("point"), pointsize=1)+
  theme_minimal()+geom_text(label=country_names,hjust=0, vjust=0,size=2,check_overlap = T)+sca
le_fill_brewer(palette="Paired")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



6.2.1.1 Similarity between clusters PAM - k-means

```
# Computes the adjusted Rand index comparing two classifications.
# The closer to 1 the more agreement

# Similarity between for k = 5
adjustedRandIndex(fit.kmeans$cluster, fit.pam$clustering)
```

```
## [1] 0.7497828
```

```
# Similarity between for k = 2
adjustedRandIndex(fit2$cluster, fit.pam2$clustering)
```

```
## [1] 0.6027743
```

They quite similar

6.2.2 Maps for clustering

```
# Select here your favorite clustering tool
map = data.frame(country=country_names, value=fit.pam$clustering)
#map = data.frame(country=names, value=fit.kmeans$cluster)
```

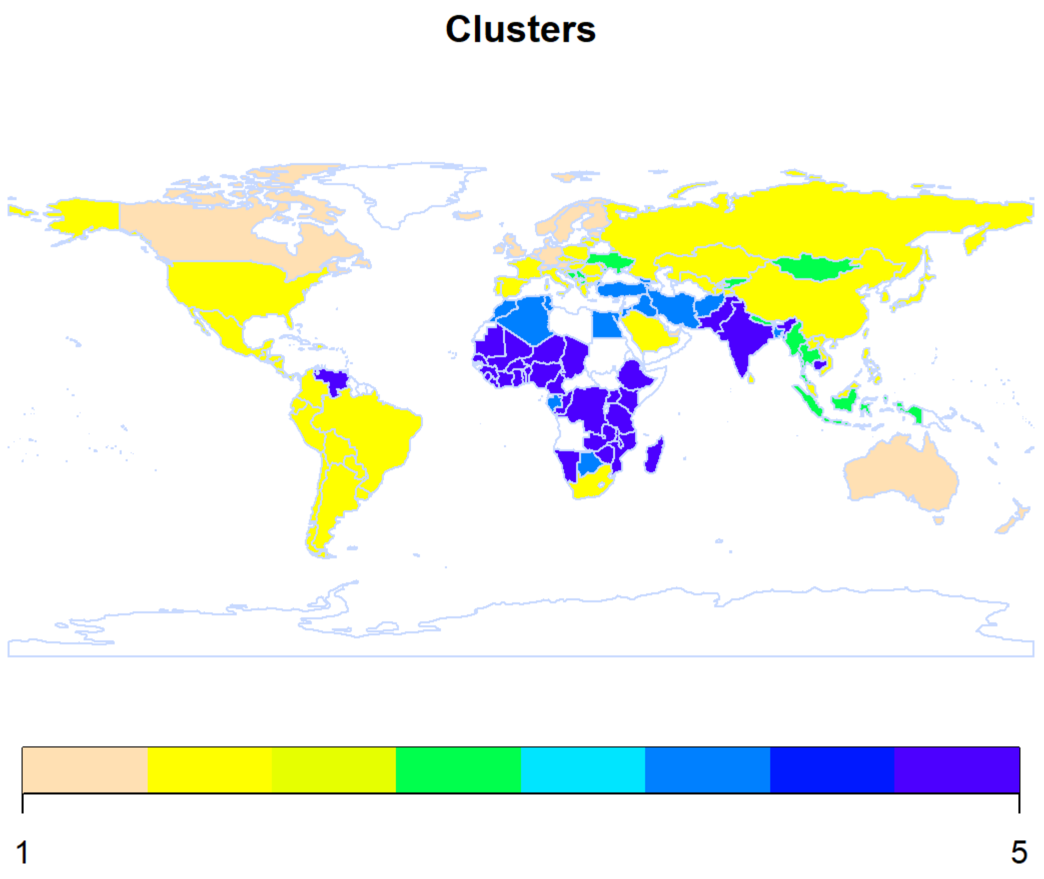
```
#Convert the country code into iso3c using the function countrycode()
map$country = countrycode(map$country, 'country.name', 'iso3c')
#Create data object supporting the map
matched <- joinCountryData2Map(map, joinCode = "ISO3",
                               nameJoinColumn = "country")
```

```
## 136 codes from your data successfully matched countries in the map
## 1 codes from your data failed to match with a country code in the map
## 107 codes from the map weren't represented in your data
```

```
#Draw the map
mapCountryData(matched,nameColumnToPlot="value",missingCountryCol = "white",
               borderCol = "#C7D9FF",
               catMethod = "pretty", colourPalette = "topo",
               mapTitle = c("Clusters"), lwd=1)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



6.3 Kernel k-means

Kernel k-means is a variant of the traditional k-means clustering algorithm that uses kernel functions to perform clustering in a high-dimensional feature space. It is particularly useful when dealing with data that is not linearly separable in the original feature space. Kernel k-means allows for non-linear separation of clusters by implicitly mapping the data into a higher-dimensional space where linear separation may be possible.

```
fit.ker <- kkmeans(as.matrix(data2), centers=5, kernel="rbfdot") # Radial Basis kernel (Gaussi
an)
```

```
## Using automatic sigma estimation (sigest) for RBF or laplace kernel
```

Retrieve the cluster centers from the kernel k-means clustering results

```
centers(fit.ker)
```

```
##          [,1]      [,2]      [,3]     [,4]      [,5]         [,6]      [,7]
## [1,] 6.556867 10.796540 0.8883605 72.98875 0.7921294 -0.057399183 0.6029250
## [2,] 7.044259 10.894483 0.9208701 71.32505 0.8771523  0.093573462 0.4553478
## [3,] 6.069650 10.121002 0.8729529 69.39656 0.8136221 -0.033325582 0.7774809
## [4,] 4.496704  8.300552 0.6804989 59.05095 0.7253140  0.046531169 0.7756900
## [5,] 5.815083  9.753453 0.8509881 66.10464 0.8187466  0.003359769 0.7775715
##           [,8]      [,9]
## [1,] 0.6490884 0.2394319
## [2,] 0.7191553 0.2409622
## [3,] 0.6910953 0.2755464
## [4,] 0.6121222 0.3413712
## [5,] 0.6568087 0.2689462
```

Obtain the number of data points in each cluster,

```
size(fit.ker)
```

```
## [1]  9 17 18 53 40
```

We see again unbalanced clusters

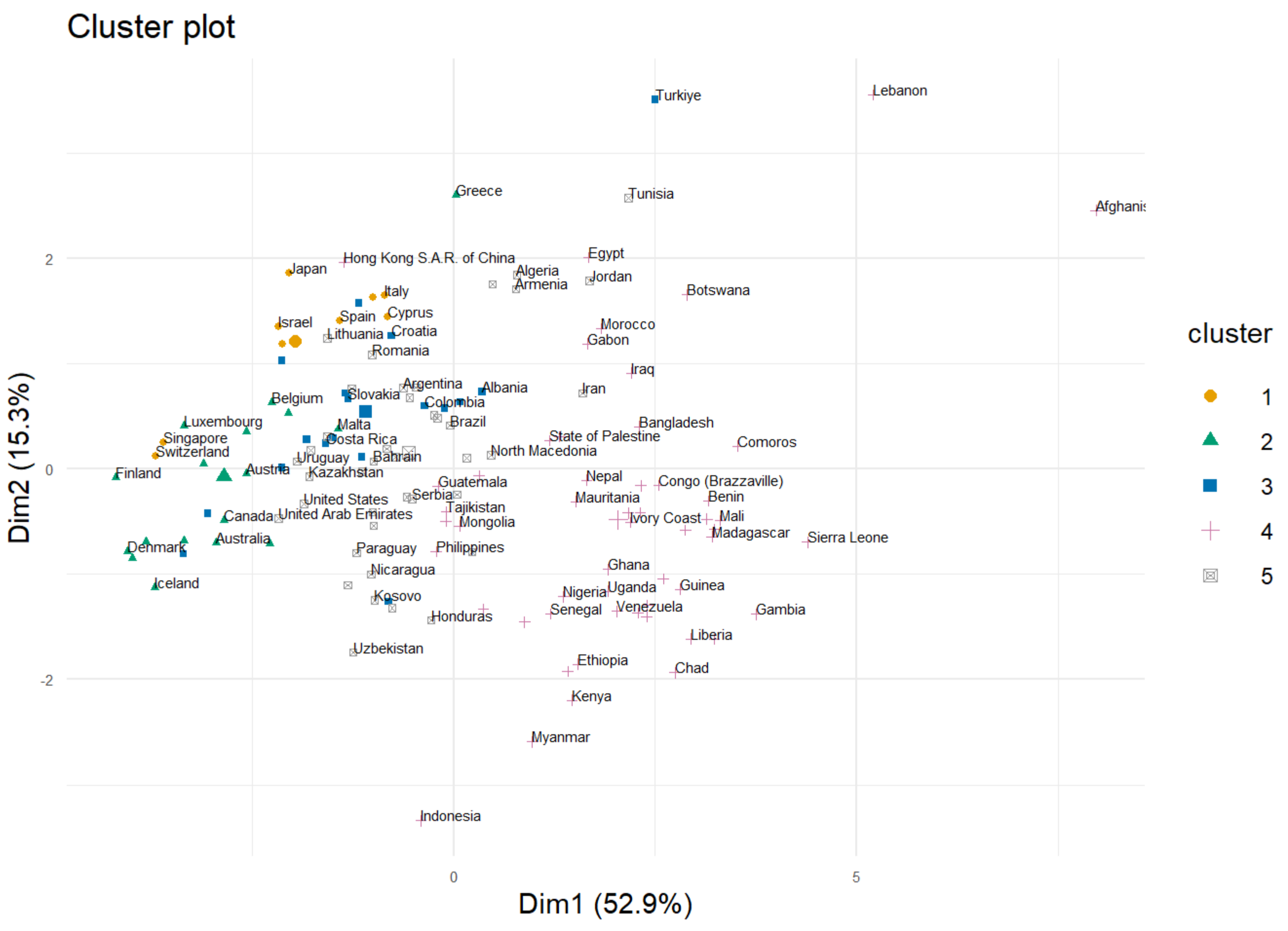
Obtain a WSS (within-cluster sum of squares for each cluster) for each cluster in the kernel k-means. Lower WSS values indicate that the data points within a cluster are closer to the cluster center, suggesting a more compact cluster.

```
withinss(fit.ker)
```

```
## [1]  81128.38 144524.93 146794.68 310564.68 292806.16
```

```
object.ker = list(data = data2, cluster = fit.ker@.Data)
fviz_cluster(object.ker, geom = c("point"), ellipse=F,pointsize=1)+
  theme_minimal()+geom_text(label=country_names,hjust=0, vjust=0,size=2,check_overlap = T)+sca
le_fill_brewer(palette="Paired")
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



Again, we obtain very different clusters

6.4 Hierarchical clustering

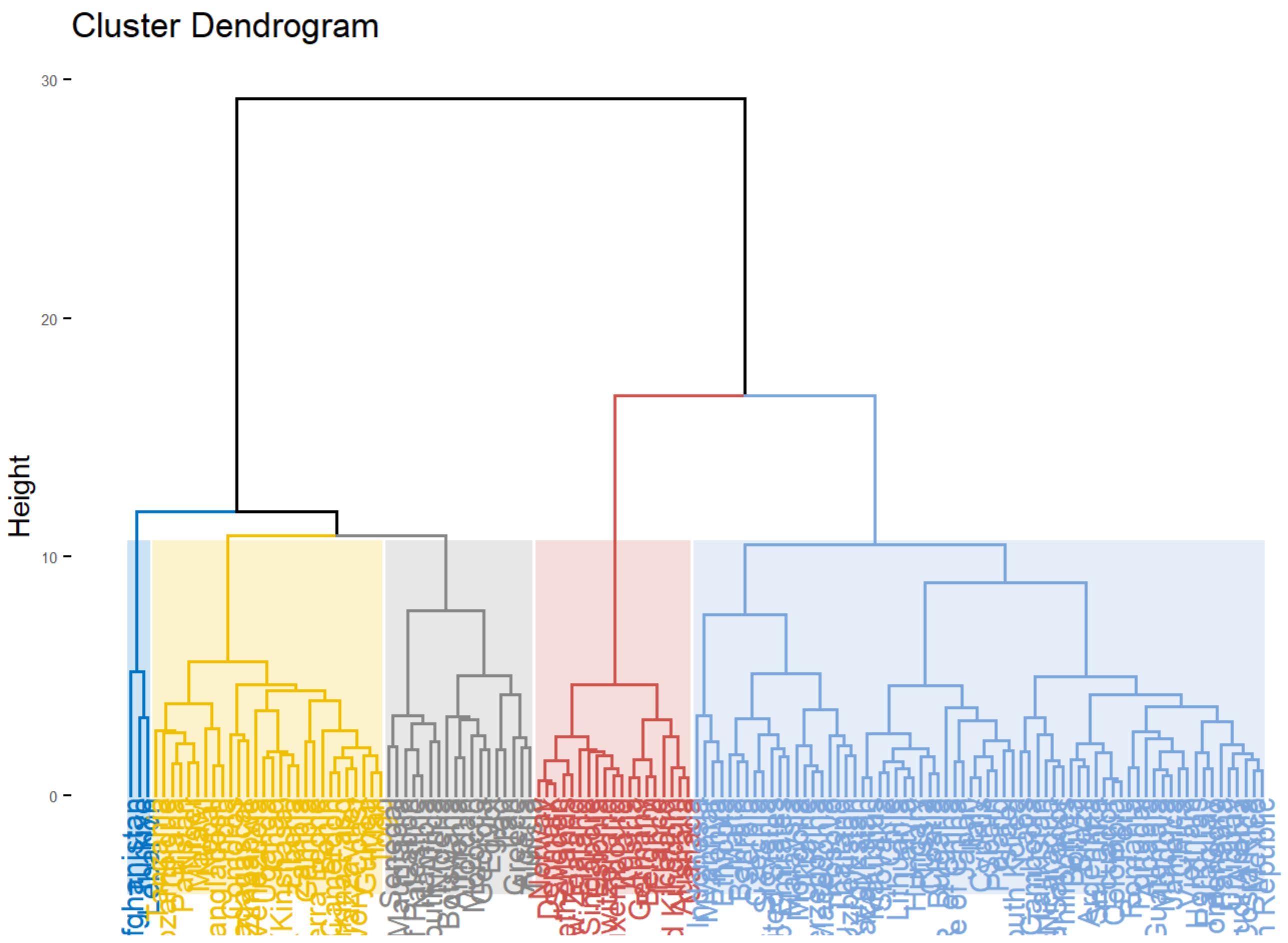
```r
d = dist(scale(data2), method = "euclidean")
hc <- hclust(d, method = "ward.D2")
```

6.4.1 Plots

####Dendogram

```r
hc$labels <- country_names

fviz_dend(x = hc,
          k=5,
          palette = "jco",
          rect = TRUE, rect_fill = TRUE,
          rect_border = "jco"   )
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



It is not clear as we have a lot of observations (countries)

####Geographical map

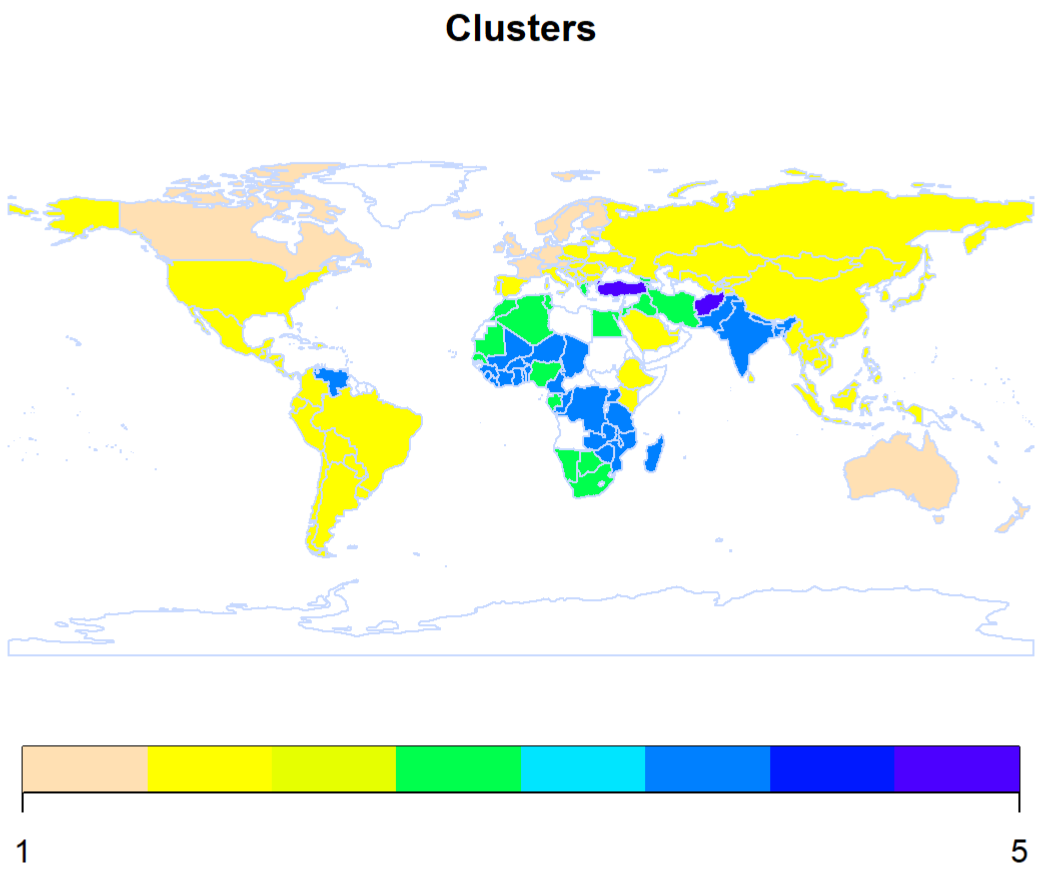
```
groups.hc = cutree(hc, k = 5)

# Map our PCA index in a map:
map = data.frame(country=country_names, value=groups.hc)
#Convert the country code into iso3c using the function countrycode()
map$country = countrycode(map$country, 'country.name', 'iso3c')
#Create data object supporting the map
matched <- joinCountryData2Map(map, joinCode = "ISO3",
                               nameJoinColumn = "country")
```

```
## 136 codes from your data successfully matched countries in the map
## 1 codes from your data failed to match with a country code in the map
## 107 codes from the map weren't represented in your data
```

```
#Draw the map

mapCountryData(matched,nameColumnToPlot="value",missingCountryCol = "white",
               borderCol = "#C7D9FF",
               catMethod = "pretty", colourPalette = "topo",
               mapTitle = c("Clusters"), lwd=1)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



7 Heatmaps

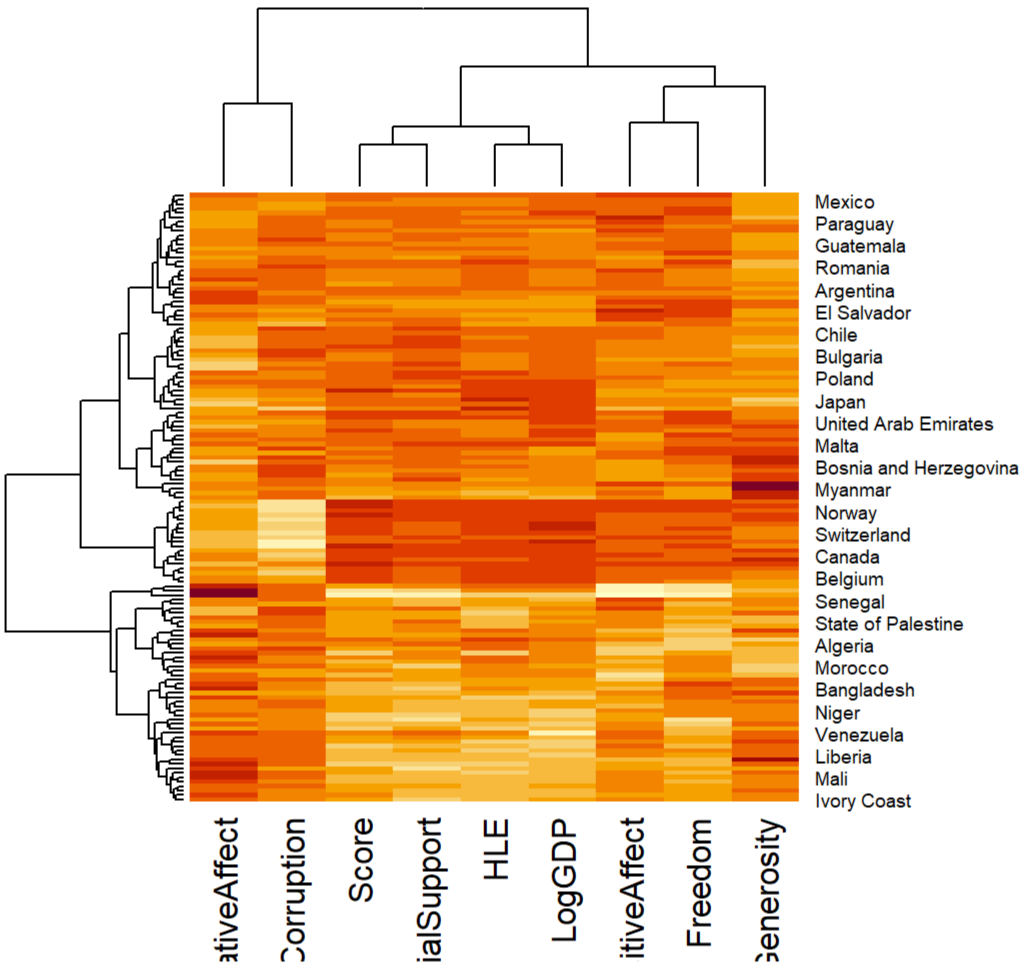
The heatmap provides a visual representation of the similarities and differences between the rows and columns of your data. The color intensity in the heatmap represents the values of your scaled data, with darker colors indicating higher values.

```r
heatmap(scale(data_name), scale = "none",
        distfun = function(x){dist(x, method = "euclidean")},
        hclustfun = function(x){hclust(x, method = "ward.D2")},
        cexRow = 0.7)
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```

```
## Error : The fig.showtext code chunk option must be TRUE
```



We see that we have 2 main groups of countries, Group 1 which has high values for "PerceptionsCorruption" and "NegativeAffect". In this group we have countries with lower happiness as Iraq.

Group 2 with high values in the other variables. We have countries with higher happines as Spain and New Zealand.

```
write.table(data2, "data_final.xls", sep=",")
```

8 CONCLUSIONS

Throughout this project, I have gained valuable insights into the significance of data cleaning techniques. By addressing issues such as missing values, duplicates, and irrelevant data points, I have ensured the data's quality and reliability for subsequent analysis.

The utilization of dimensionality reduction methods, specifically Principal Component Analysis (PCA) and Factor Analysis (FA), has provided a deeper understanding of our dataset. These techniques have enabled me to uncover the relationships between variables, emphasizing the strong positive correlation between happiness levels and factors like GDP and Social Support, while revealing negative associations with Perceptions of Corruption and Negative Affect.

Clustering methodologies, including k-means, PAM, and kernel k-means, have proven instrumental in categorizing countries based on their unique characteristics. By forming these clusters, I have identified commonalities and disparities among countries, offering valuable insights into the potential drivers of happiness.

A notable outcome of this analysis is the identification of two primary country groups characterized by their "Perceptions of Corruption" and "Negative Affect" values. These groupings correspond to variations in happiness levels and have provided valuable geographical insights into the distribution of happiness across countries.

In summary, this project has highlighted the significance of data cleaning, dimensionality reduction, and clustering techniques in uncovering patterns and associations within the World Happiness dataset. These findings contribute to a deeper understanding of the factors influencing happiness and the geographic distribution of well-being, facilitating informed decision-making and further research in this area.