Analysis to Recommend Countries using R

Kelvin Hendersen

2023-07-07

1. Business/Project Understanding

Objective To categorize countries using socioeconomic and health factors determine the development of the country as a whole.

About Organization: HELP International is an international humanitarian organization committed to combating poverty and provide basic facilities and assistance to people in countries underdeveloped during disasters and natural disasters.

Problem: HELP International has raised approx. $10 million. Nowadays, CEO need decide how to use this money strategically and effectively. So the CEO has to take the decision to select the country that needs the most assistance.

2. The Data

2.1. Dataset Understanding

Explanation of feature fields

- **Negara**: Country name
- **Kematian_anak**: Deaths of children under 5 years of age per 1000 births
- **Ekspor**: Export of goods and services per capita
- **Kesehatan**: Total health spending per capita
- **Impor**: Imports of goods and services per capita
- **Pendapatan**: Net income per person
- **Inflasi**: Measurement of the annual growth rate of Total GDP
- **Harapan_hidup**: The average number of years a newborn would live if current death patterns remained the same
- **Jumlah_fertiliti**: The number of children that would be born to each woman if the current age fertility rate remained the same
- **GDPperkapita**: GDP per capita Calculated as Total GDP divided by the total population

The data/file to be processed is named 'DATA_Negara_HELP csv' which consists of:

- 167 rows
- 10 columns

import the required libraries

```
library("tidyverse")
library("dplyr")
library("lares")
library("reshape2")
library("ggplot2")
library("ggpubr")
library("factoextra")
library("NbClust")
library("cluster")
library("skimr")
```

Read Data

```
df <- read_csv("Data_Negara_HELP.csv")
```

Displays the top 5 data

```
head(df,5) #Displays the top 5 data
```

```
## # A tibble: 5 × 10
##   Negara   Kematian_anak Ekspor Kesehatan Impor Pendapatan Inflasi Harapan_hidup
##   <chr>            <dbl>  <dbl>     <dbl> <dbl>      <dbl>   <dbl>         <dbl>
## 1 Afghani…          90.2   10        7.58  44.9       1610    9.44          56.2
## 2 Albania           16.6   28        6.55  48.6       9930    4.49          76.3
## 3 Algeria           27.3   38.4      4.17  31.4      12900   16.1           76.5
## 4 Angola           119     62.3      2.85  42.9       5900   22.4           60.1
## 5 Antigua…          10.3   45.5      6.03  58.9      19100    1.44          76.8
## # ℹ 2 more variables: Jumlah_fertiliti <dbl>, GDPperkapita <dbl>
```

2.2. EDA (Exploratory Data Analysis) - Part 1

(1)In statistics, exploratory data analysis (EDA) is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods.

Footnote: 1. link (https://en.wikipedia.org/wiki/Exploratory_data_analysis)

2.2.1. Find Missing Value

```
sum(is.null(df))
```

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

Because none of the datasets above have missing values such as “NaN”, “NULL”, etc., we proceed to the next stage, which is to perform a Multivariate Analysis Dataset to find correlations for each feature/column of data.

2.2.2. Multivariate analysis

Multivariate analysis is used to analyze more than 2 variables at the same time, the resulting trends can be naturally multidimensional, with this analysis it will help us understand which data has complex trends in combinations of attributes.

Creates a new dataframe for the backup, then retrieves the columns containing only numeric values

```
df_copy <- data.frame()
df_copy <- df
df_copy = subset(df_copy, select = -c(Negara))
```

Look for correlation relationships between columns with the cor() function to display matrices

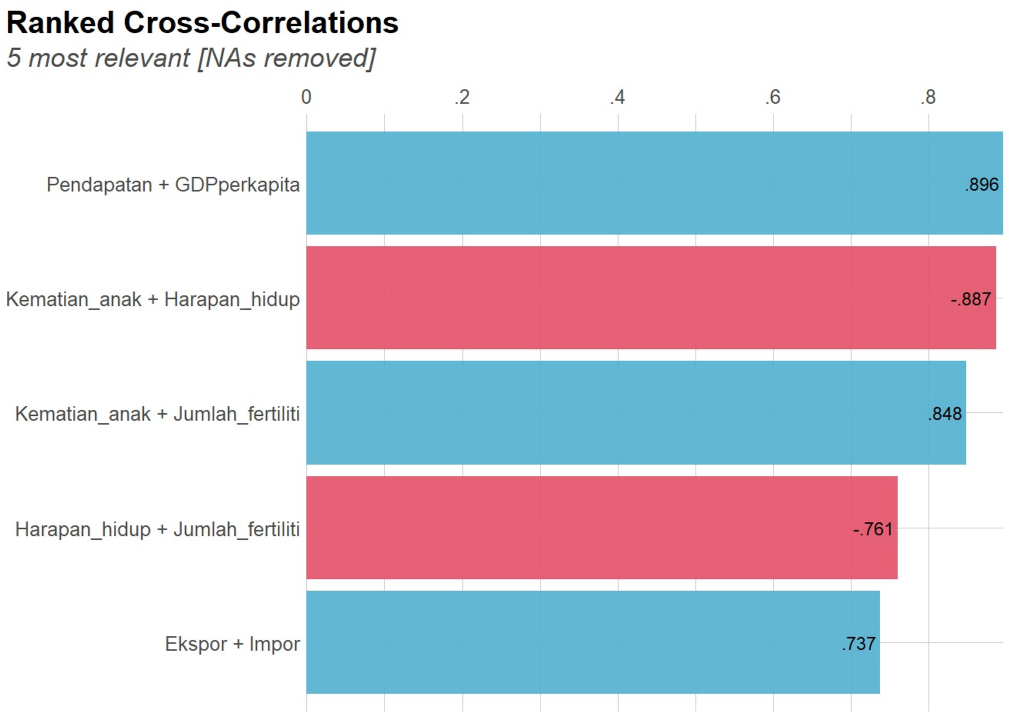
```
cor(df_copy)
```

```
##                  Kematian_anak     Ekspor   Kesehatan       Impor Pendapatan
## Kematian_anak        1.0000000 -0.3180932 -0.20040206 -0.12721092 -0.5243150
## Ekspor              -0.3180932  1.0000000 -0.11440840  0.73738083  0.5167836
## Kesehatan           -0.2004021 -0.1144084  1.00000000  0.09571668  0.1295786
## Impor               -0.1272109  0.7373808  0.09571668  1.00000000  0.1224062
## Pendapatan          -0.5243150  0.5167836  0.12957861  0.12240625  1.0000000
## Inflasi              0.2882762 -0.1072944 -0.25537579 -0.24699428 -0.1477560
## Harapan_hidup       -0.8866761  0.3163126  0.21069212  0.05439053  0.6119625
## Jumlah_fertiliti     0.8484781 -0.3200106 -0.19667399 -0.15904843 -0.5018401
## GDPperkapita        -0.4830322  0.4187248  0.34596553  0.11549817  0.8955714
##                     Inflasi Harapan_hidup Jumlah_fertiliti GDPperkapita
## Kematian_anak     0.2882762   -0.88667610        0.8484781   -0.4830322
## Ekspor           -0.1072944    0.31631260       -0.3200106    0.4187248
## Kesehatan        -0.2553758    0.21069212       -0.1966740    0.3459655
## Impor            -0.2469943    0.05439053       -0.1590484    0.1154982
## Pendapatan       -0.1477560    0.61196247       -0.5018401    0.8955714
## Inflasi           1.0000000   -0.23970496        0.3169211   -0.2216311
## Harapan_hidup    -0.2397050    1.00000000       -0.7608747    0.6000891
## Jumlah_fertiliti  0.3169211   -0.76087469        1.0000000   -0.4549103
## GDPperkapita     -0.2216311    0.60008913       -0.4549103    1.0000000
```

Since it is difficult to see these numbers in the table by default, we will define a function to take the best 5 strong positive correlation numbers and display them onto a bar graph (See Section 2.3).

2.3. Feature Selection

```
corr_cross(df_copy, top=5, rm.na=TRUE)
```



From the graph above there are 2 groups namely blue bars representing positive correlations and red for negative correlations (Correlation coefficient start from range -1 to 1). Of course we will take the positive correlation/top blue bar (Pendapatan & GDPperkapita) with a correlation value of +0.896.

But we will see how the relationships between columns are related by using a Heatmap plot. Code source reference:

1. link (https://www.geeksforgeeks.org/how-to-create-correlation-heatmap-in-r/)
2. link (http://www.sthda.com/english/wiki/ggplot2-quick-correlation-matrix-heatmap-r-software-and-data-visualization)

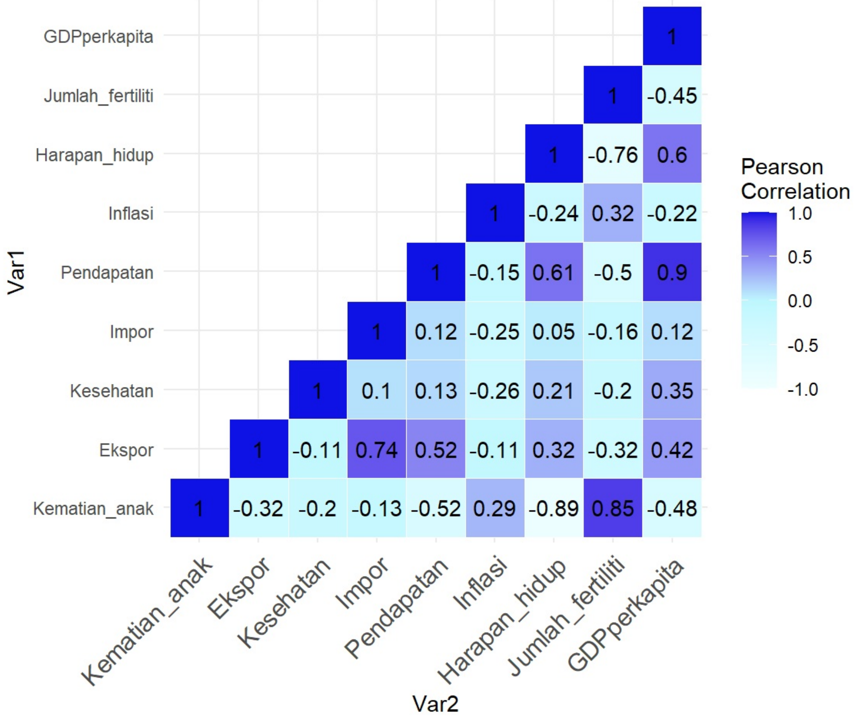
```
#Round off to 2 decimal places
corr_mat <- round(cor(df_copy),2)

# Get lower triangle of the correlation matrix
get_lower_tri<-function(corr_mat){
  corr_mat[upper.tri(corr_mat)] <- NA
  return(corr_mat)
}
# Get upper triangle of the correlation matrix
get_upper_tri <- function(corr_mat){
  corr_mat[lower.tri(corr_mat)]<- NA
  return(corr_mat)
}

upper_tri <- get_upper_tri(corr_mat)

melted_cormat <- melt(upper_tri, na.rm = TRUE)

ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "#FFFFFF")+
  scale_fill_gradient2(low = "#F1FFFF", mid = "#BEF7FF", high = "#0F12E4", midpoint = 0, limit = c(-1,1), space =
"Lab", name="Pearson\nCorrelation") +
  geom_text(aes(label = melted_cormat$"value"))+
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1))+
  coord_fixed()
```



On the heatmap, we can see the darkest color (most dark blue). For positive relations there are 3 strongest candidates, namely:

1. Pendapatan with GDPpercapita with a value of 0.9
2. Kematian_anak with Jumlah_fertiliti with a value of 0.85
3. Ekspor with Impor with a value of 0.74

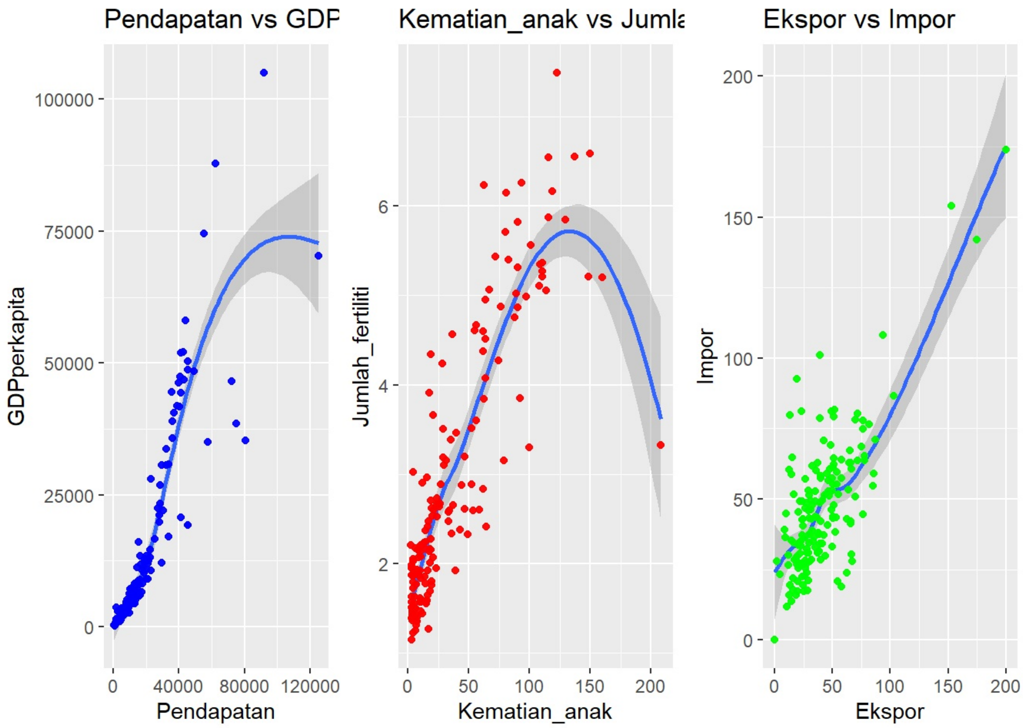
Of course we take the 2 variables with the highest positive correlation, namely Pendapatan with GDPperkapita. This can be proven by the plot as follows.

```
plot1 <- ggplot(data=df_copy)+
  geom_smooth(mapping=aes(x=Pendapatan,y=GDPperkapita),method="loess")+
  geom_jitter(mapping=aes(x=Pendapatan,y=GDPperkapita),color="blue")+
  labs(title="Pendapatan vs GDPperkapita")

plot2 <- ggplot(data=df_copy)+
  geom_smooth(mapping=aes(x=Kematian_anak,y=Jumlah_fertiliti),method="loess")+
  geom_jitter(mapping=aes(x=Kematian_anak,y=Jumlah_fertiliti),color="red")+
  labs(title="Kematian_anak vs Jumlah_fertiliti")

plot3 <- ggplot(data=df_copy)+
  geom_smooth(mapping=aes(x=Ekspor,y=Impor),method="loess")+
  geom_jitter(mapping=aes(x=Ekspor,y=Impor),color="green")+
  labs(title="Ekspor vs Impor")

ggarrange(plot1, plot2, plot3, ncol = 3, nrow = 1)
```



In the graph above it can be seen that the distribution of the blue data points is closer to the correlation line than the red and green dots which tend to spread away from the correlation line. This proves that there is a strong positive correlation between Pendapatan and GDPperkapita.

After we decided to link the correlation of Pendapatan with GDPperkapita. The next step is to look for outliers to be analyzed.

2.4. Data Cleaning

Generally, raw data contains outlier data, of course in this case we cannot use the data for analysis because it still contains outlier data. For that we need to examine the data between the lower limit (Q1), the middle / median (Q2) and the upper limit (Q3). Then examine outlier data that is outside Q1 and Q3

Interquartile Range is the percentile difference between the upper (Q3) and lower (Q1) quartiles

`IQR = Q3-Q1`

or in other words, IQR is the limits within the range of dividing the data into 4 equal parts marked with limits/symbols Q1, Q2 (median), Q3.

The formula for finding the lower (Q1) and upper (Q3) limits:

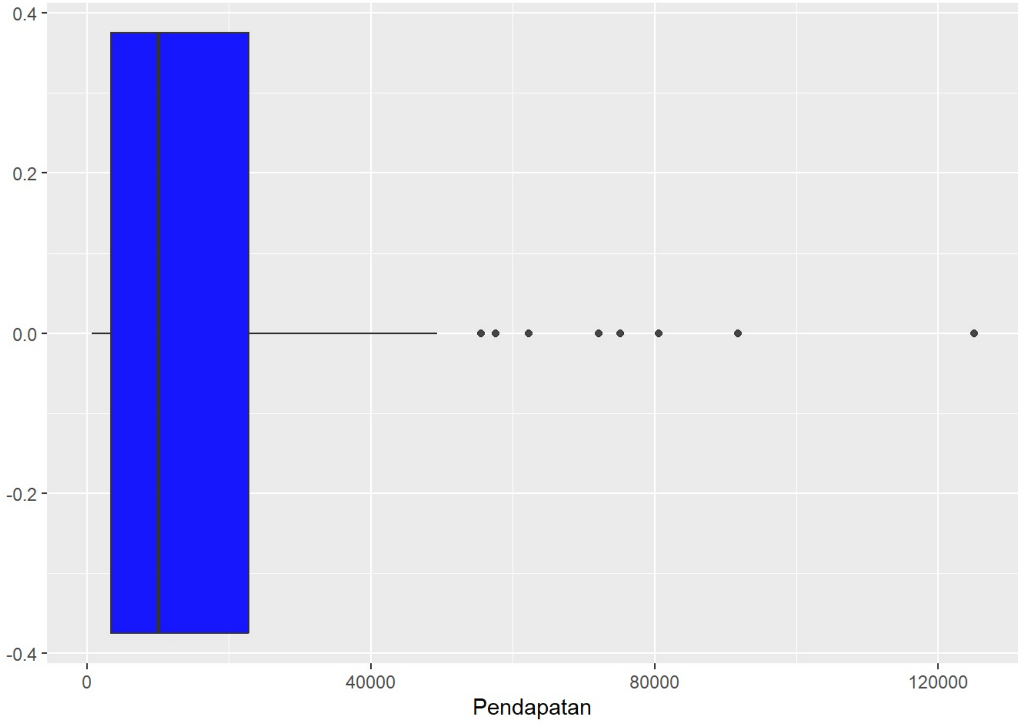
`Lower_bound = Q1 - (1.5*IQR)`

`Upper_bound = Q3 + (1.5*IQR)`

2.4.1. Finding Outlier

Outlier display of ‘Pendapatan’ column before analysis (using boxplot)

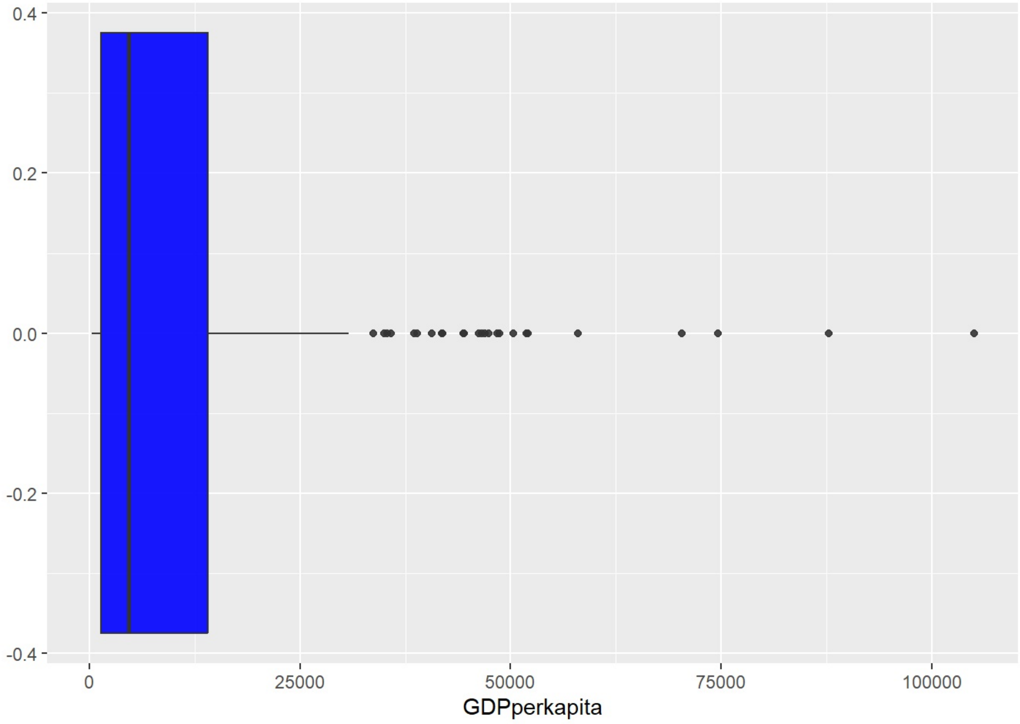
```
ggplot(data=df_copy)+
  geom_boxplot(mapping = aes(x=Pendapatan), fill='blue')
```



In the boxplot above, it can be seen that the ‘Pendapatan’ column has outliers that are beyond the upper quartile 3 (this can be seen from the Pendapatan range bounded by the line as the Inter Quartille Range (IQR))

Outlier display of column ‘GDPperkapita’ before analysis (using boxplot)

```
ggplot(data=df_copy)+
  geom_boxplot(mapping = aes(x=GDPperkapita), fill="blue")
```



In the boxplot above, it can be seen that the column ‘GDPperkapita’ also has outliers that are beyond the upper quartile 3 (this can be seen from the income range bounded by the line as the Inter Quartille Range (IQR))

We will now look at the individual outliers in the ‘Pendapatan’ and ‘GDPperkapita’ columns.

Find outliers for the ‘Pendapatan’ column using the zscore:

```
zscore_pendapatan <- (df_copy$"Pendapatan"-mean(df_copy$"Pendapatan"))/sd(df_copy$"Pendapatan")
zscore_gdp <- (df_copy$"GDPperkapita"-mean(df_copy$"GDPperkapita"))/sd(df_copy$"GDPperkapita")

variabel1 <- data.frame(df)
variabel1 <- mutate(variabel1, zscore_pendapatan = zscore_pendapatan, zscore_gdp = zscore_gdp)

hasilzscorependapatan <- variabel1 %>% filter(zscore_pendapatan > 3)
hasilzscorependapatan
```

```
##       Negara Kematian_anak Ekspor Kesehatan Impor Pendapatan Inflasi
## 1     Brunei          10.5   67.4      2.84  28.0      80600   16.70
## 2     Kuwait          10.8   66.7      2.63  30.4      75200   11.20
## 3 Luxembourg           2.8  175.0      7.77 142.0      91700    3.62
## 4      Qatar           9.0   62.3      1.81  23.8     125000    6.98
##   Harapan_hidup Jumlah_fertiliti GDPperkapita zscore_pendapatan zscore_gdp
## 1          77.1             1.84        35300          3.291580   1.218626
## 2          78.2             2.21        38500          3.011469   1.393216
## 3          81.3             1.63       105000          3.867364   5.021405
## 4          79.5             2.07        70300          5.594716   3.128199
```

The above shows that there are about 4 rows of ‘Pendapatan’ data whose outlier values are far from the distribution of other data, namely Brunei, Kuwait, Luxembourg, Qatar

```
hasilzscoregdp <- variabel1 %>% filter(zscore_gdp > 3)
hasilzscoregdp
```

```
##        Negara Kematian_anak Ekspor Kesehatan Impor Pendapatan Inflasi
## 1  Luxembourg           2.8  175.0      7.77 142.0      91700   3.620
## 2      Norway           3.2   39.7      9.48  28.5      62300   5.950
## 3       Qatar           9.0   62.3      1.81  23.8     125000   6.980
## 4 Switzerland           4.5   64.0     11.50  53.3      55500   0.317
##   Harapan_hidup Jumlah_fertiliti GDPperkapita zscore_pendapatan zscore_gdp
## 1          81.3             1.63       105000          3.867364   5.021405
## 2          81.0             1.95        87800          2.342315   4.082986
## 3          79.5             2.07        70300          5.594716   3.128199
## 4          82.2             1.52        74600          1.989583   3.362804
```

The above shows that there are about 4 rows of ‘GDP per capita’ data whose outlier values are far from the distribution of other data, namely Luxembourg, Norway, Qatar, Switzerland

2.4.2. Handling Outlier & Missing Value

Here we will take values that are within the range of the IQR, remove outlier, and then drop na value (Code source reference: link (https://www.geeksforgeeks.org/how-to-remove-outliers-from-multiple-columns-in-r-dataframe/))

```
df_copy <- data.frame(df)

# create detect outlier function
detect_outlier <- function(i) {
  Q1 <- quantile(i, probs=.25) # calculate first quantile
  Q3 <- quantile(i, probs=.75) # calculate third quantile
  IQR <- Q3-Q1 # calculate inter quartile range
  (i<Q1-(IQR*1.5)) | (i>Q3+(IQR*1.5)) # return TRUE or FALSE (Find & Get Outlier)
}

# create remove outlier function
remove_outlier <- function(dataframe, columns=names(dataframe)) {
  for (col in columns) { # for loop in columns vector
    dataframe <- dataframe[!detect_outlier(dataframe[[col]]), ] # "!" To take apart the outliers
  }
  return(dataframe)
}

df_copy <- remove_outlier(df_copy, c('Pendapatan', 'GDPperkapita'))
df_copy <- na.omit(df_copy)
```

Now we check whether the df_copy variable that has dropped the Outlier still has a missing value or not

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

Now we check again whether the Outliers still exist or not after going through the processes above. But before that, we must first know the upper & lower limits on the respective ‘Pendapatan’ & ‘GDPperkapita’ columns.

```
q1_pendapatan <- quantile(df_copy$"Pendapatan", .25)
q3_pendapatan <- quantile(df_copy$"Pendapatan", .75)

q1_gdp <- quantile(df_copy$"GDPperkapita", .25)
q3_gdp <- quantile(df_copy$"GDPperkapita", .75)

iqr_pendapatan <- q3_pendapatan - q1_pendapatan
iqr_gdp <- q3_gdp - q1_gdp

lowerboundpendapatan = q1_pendapatan-(1.5*iqr_pendapatan)
upperboundpendapatan = q3_pendapatan+(1.5*iqr_pendapatan)

lowerboundgdp = q1_gdp-(1.5*iqr_gdp)
upperboundgdp = q3_gdp+(1.5*iqr_gdp)
cat("Lower bound for Pendapatan: ",lowerboundpendapatan,"\n","Upper bound for Pendapatan: ",upperboundpendapatan,
"\n","Lower bound GDPperkapita: ",lowerboundgdp,"\n","Upper bound for GDPperkapita: ",upperboundgdp)
```

```
## Lower bound for Pendapatan:  -16787.5
##  Upper bound for Pendapatan:  35112.5
##  Lower bound GDPperkapita:  -9360
##  Upper bound for GDPperkapita:  18760
```

After we know the upper and lower limits of each column, now we will see if there are any outliers outside the upper & lower limits

```
bukti <- df_copy %>% filter(Pendapatan < lowerboundpendapatan) %>%
  filter(Pendapatan > upperboundpendapatan) %>%
  filter(GDPperkapita < lowerboundgdp) %>%
  filter(GDPperkapita > upperboundgdp)

bukti
```

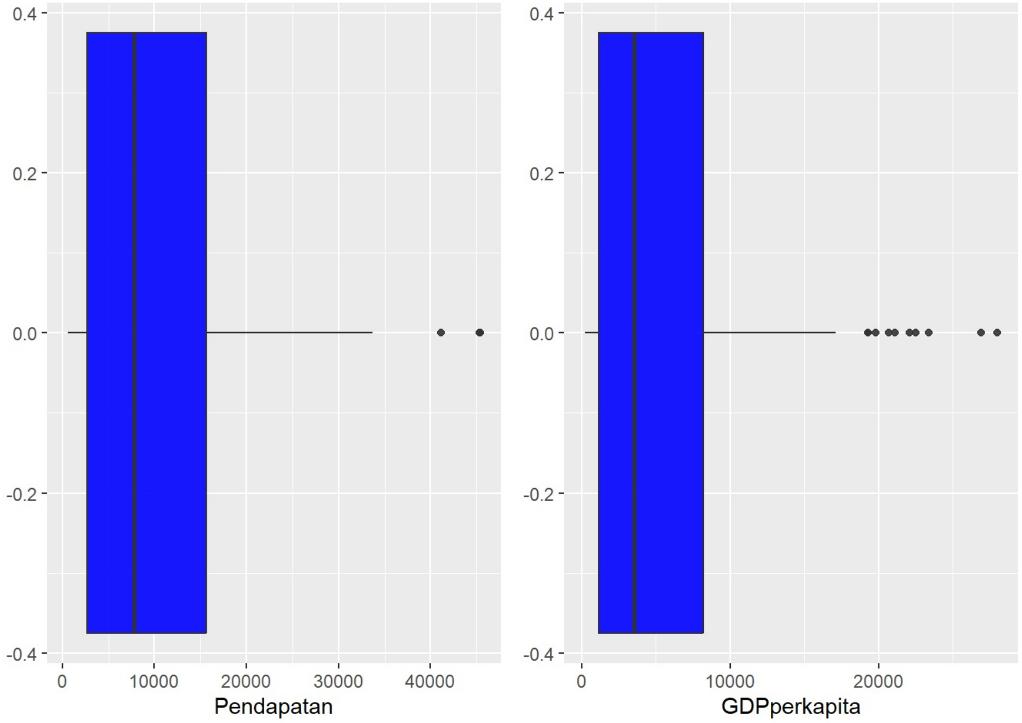
```
##  [1] Negara           Kematian_anak    Ekspor           Kesehatan
##  [5] Impor            Pendapatan       Inflasi          Harapan_hidup
##  [9] Jumlah_fertiliti GDPperkapita
## <0 rows> (or 0-length row.names)
```

Now we check again the outliers in the ‘Pendapatan’ column with a boxplot plot (after the data has been analyzed & processed)

```
plot4 <- ggplot(data=df_copy)+
  geom_boxplot(mapping = aes(x=Pendapatan), fill='blue')

plot5 <- ggplot(data=df_copy)+
  geom_boxplot(mapping = aes(x=GDPperkapita), fill='blue')

ggarrange(plot4, plot5, ncol = 2, nrow = 1)
```



If you pay attention, there are still data that are still considered Outliers in the ‘Pendapatan’ and ‘GDPperkapita’ columns due to differences in the previous quartile data so that the data is considered Outilers. This is commonplace sometimes.

2.5. EDA - Part 2

2.5.1.Univariate Analysis

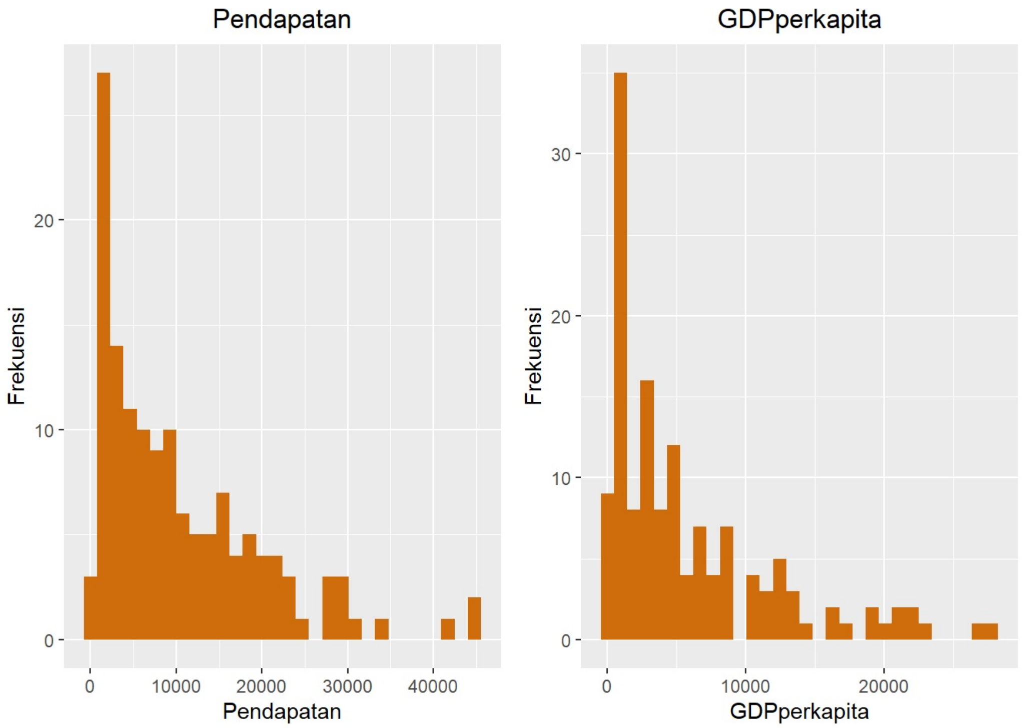
Univariate Analysis is a technique for understanding and exploring data. The prefix ‘Uni’ means ‘one’, so univariate analysis is a single feature data analysis.

Now we will try to analyze the features one by one using a histogram plot.

```
plot6 <- ggplot(df_copy, mapping=aes(x=Pendapatan))+
  geom_histogram()+
  theme(plot.title = element_text(hjust = 0.5))+
  labs(title="Pendapatan", x="Pendapatan",y="Frekuensi")

plot7 <- ggplot(df_copy, mapping=aes(x=GDPperkapita))+
  geom_histogram()+
  theme(plot.title = element_text(hjust = 0.5))+
  labs(title="GDPperkapita", x="GDPperkapita",y="Frekuensi")

ggarrange(plot6, plot7, ncol = 2, nrow = 1)
```



We need to know the income formula first:

```
Income = GDP/population
```

Information obtained from the information above:

1. Between Pendapatan & GDPperkapita is directly proportional
2. When Pendapatan decreases, it is automatically influenced by GDP which also decreases
3. The majority of individuals/individuals on average Pendapatan is in the range <= 10000 to >= 20000
4. Likewise with the majority of GDPperkapita on average in countries in the range 0 < x < 10000

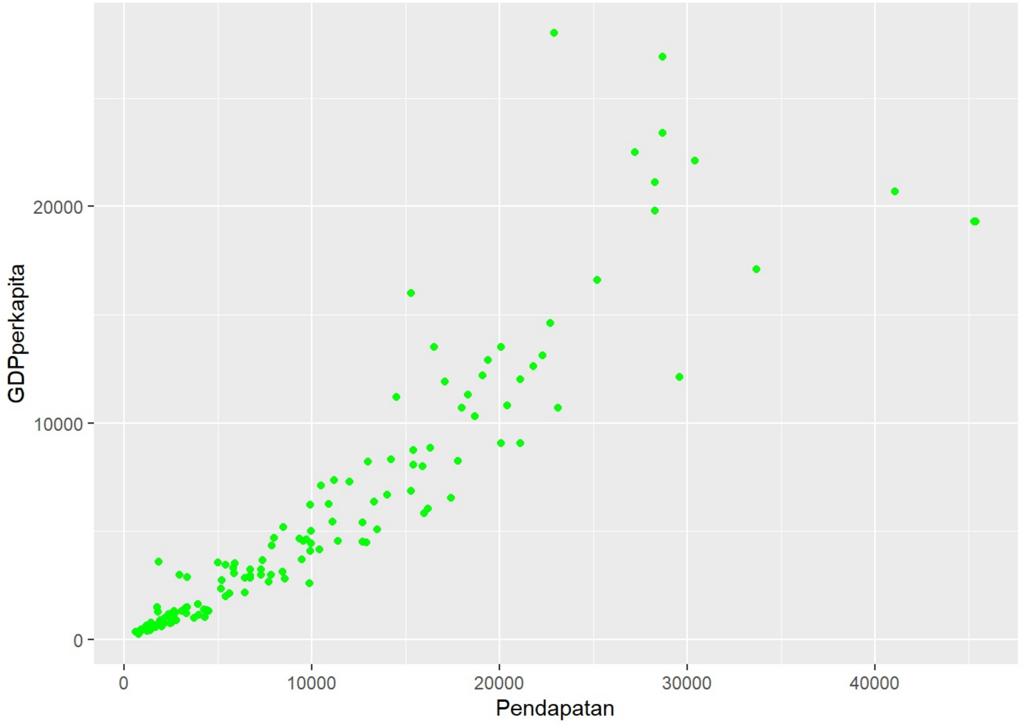
2.5.2. Bivariate Analysis

After we try Univariate, we will also try Bivariate Analysis.

Bivariate analysis is used to analyze 2 variables and find a relationship. Bivariate analysis is also a way to use the correlation coefficient in order to find out whether two variables have a relationship or not.

Now we will analyze using Scatter plots

```
ggplot(data=df_copy)+
  geom_point(mapping = aes(x=Pendapatan,y=GDPperkapita),color="green")
```



Information based on the Bivariate Analysis above:

1. Just like Univariate's assumption that the increase in coordinate points/data distribution between Pendapatan & GDP is directly proportional
2. When Pendapatan rises, GDP also tends to rise
3. The data distribution points tend to converge between the Pendapatan range of 10,000 and GDP per capita of 5,000

3. Clustering

3.1. Scale Data

The scale feature is used to normalize the distance between data variables between x and y.

If we look at the df_copy, the values between rows are less close together if we want to plot them. For that we will do data scaling.

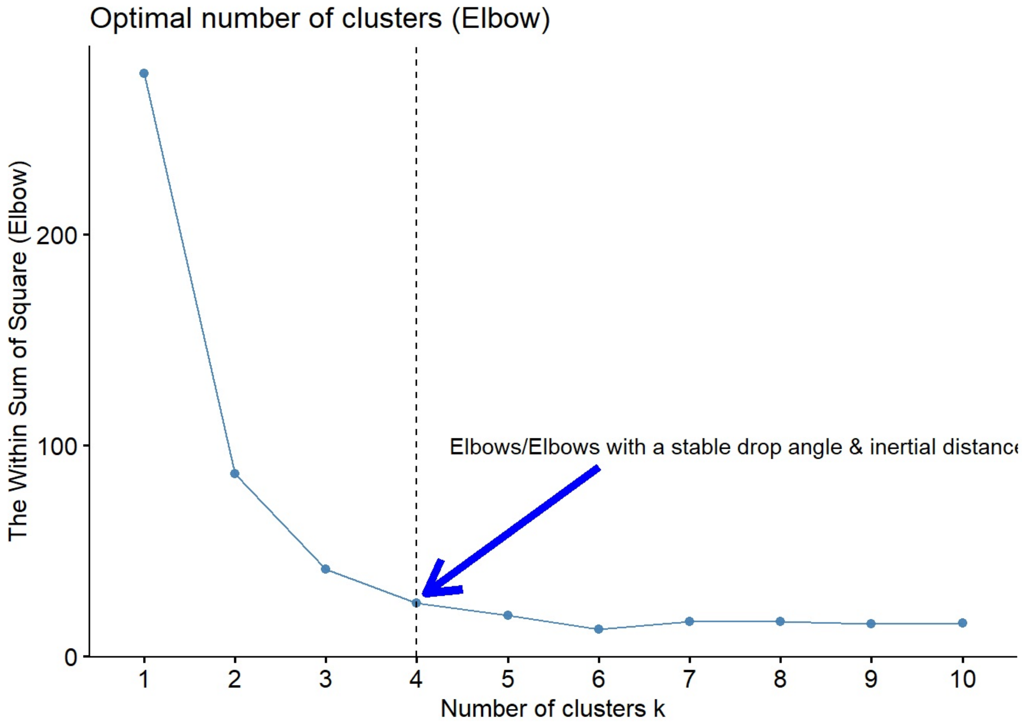
```
df_copy_sc <- select(df_copy, "Pendapatan", "GDPperkapita")
df_copy_sc <- mutate(df_copy_sc, Pendapatan=scale(df_copy$Pendapatan), GDPperkapita=scale(df_copy$GDPperkapita))
```

3.2. Choose the right number of clusters

As the name suggests, determining the value of K is one of the important things to do in the K-Means algorithm. To be able to determine this value we will use four methods of determining the best k value, namely the Elbow Method, Silhouette Method, Gap Statistic Method, and finally the combined function of several clustering methods.

Elbow Method

```
fviz_nbclust(df_copy_sc, kmeans, method = "wss")+
  geom_vline(xintercept = 4, linetype = 2)+
  labs(title="Optimal number of clusters (Elbow)", x="Number of clusters k",y="The Within Sum of Square (Elbow)")
+
  annotate("segment",x=6,xend=4.1,y=90,yend=30,color="blue",size=2,arrow=arrow())+
  annotate("text", x=8.5, y=100, label="Elbows/Elbows with a stable drop angle & inertial distance between cluste
rs", color="black")
```



Here we can see the inflection point on the Elbow indicating the number 4.

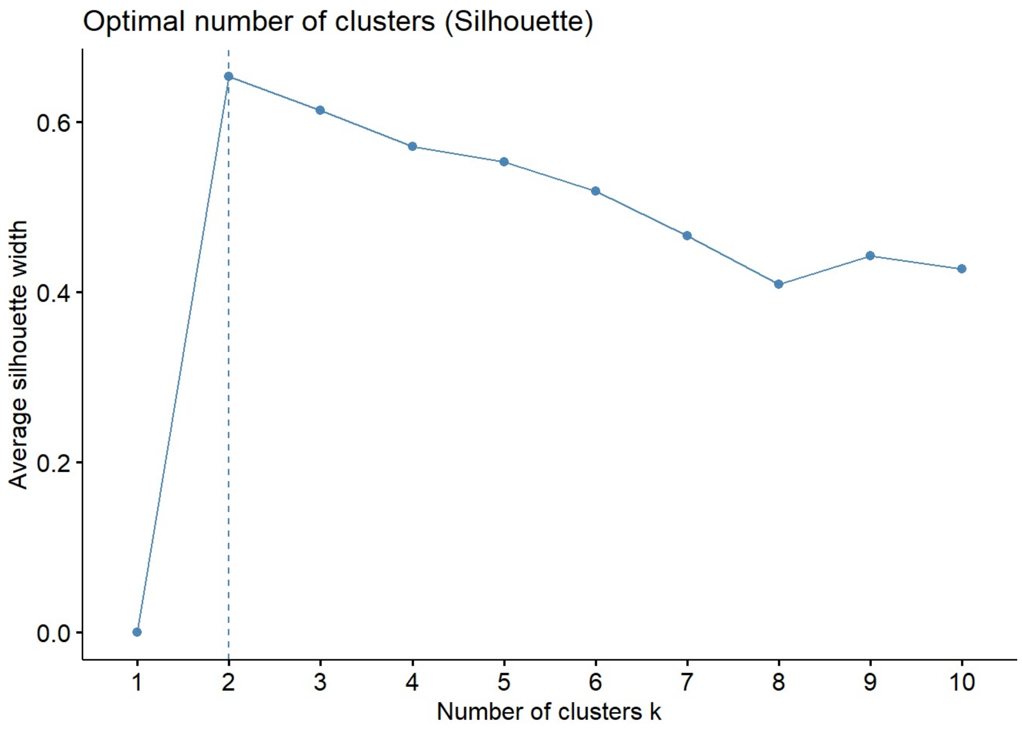
Silhouette Method

The Silhouette Method uses a coefficient value that is calculated from how close the relationships between objects in a cluster are, and how far a cluster is apart from other clusters. the equation used is:

```
Silhouette coefficient = (x-y)/ max(x,y)
```

Where x is the distance to other clusters and y is the distance between objects in the same cluster. The optimum K value is obtained from the peak value of the K plot against the Silhouette Coefficient.

```
fviz_nbclust(df_copy_sc, kmeans, method = "silhouette")+
  labs(title="Optimal number of clusters (Silhouette)")
```

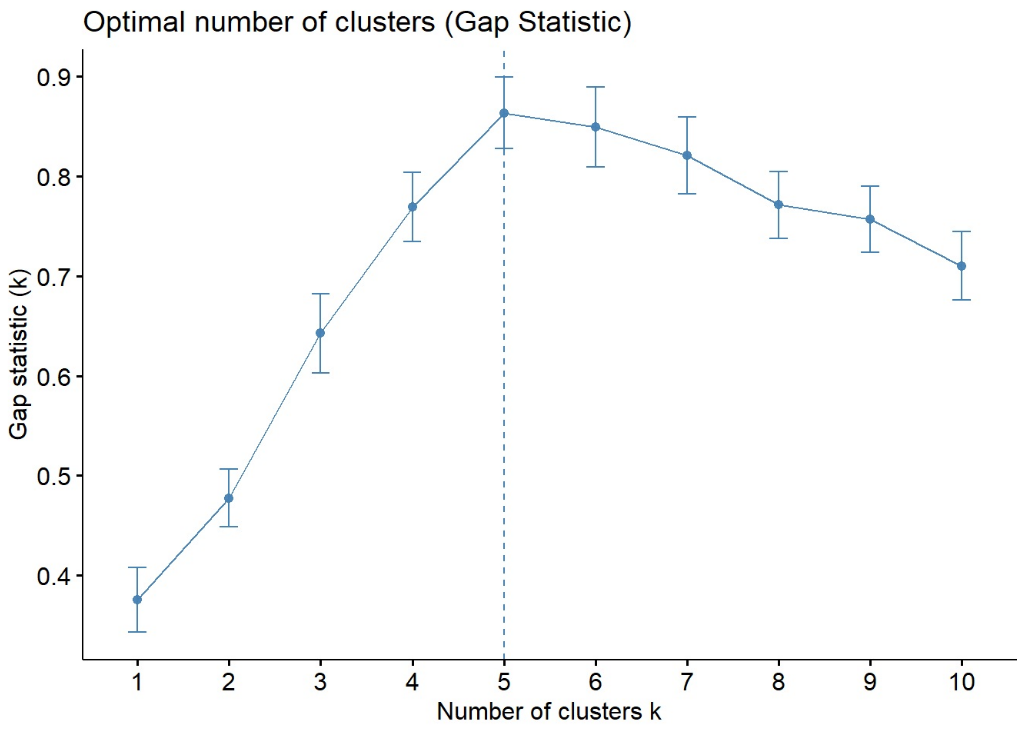


From these results we can see that the peak value with Silhouette Method is at a value of K = 2

Gap Statistic (2)Gap Statistics is a method to choose the number of K, where the biggest jump in within-cluster distance occurred, based on the overall behavior of uniformly drawn samples.

Footnote: 2. link (https://towardsdatascience.com/k-means-clustering-and-the-gap-statistics-4c5d414acd29)

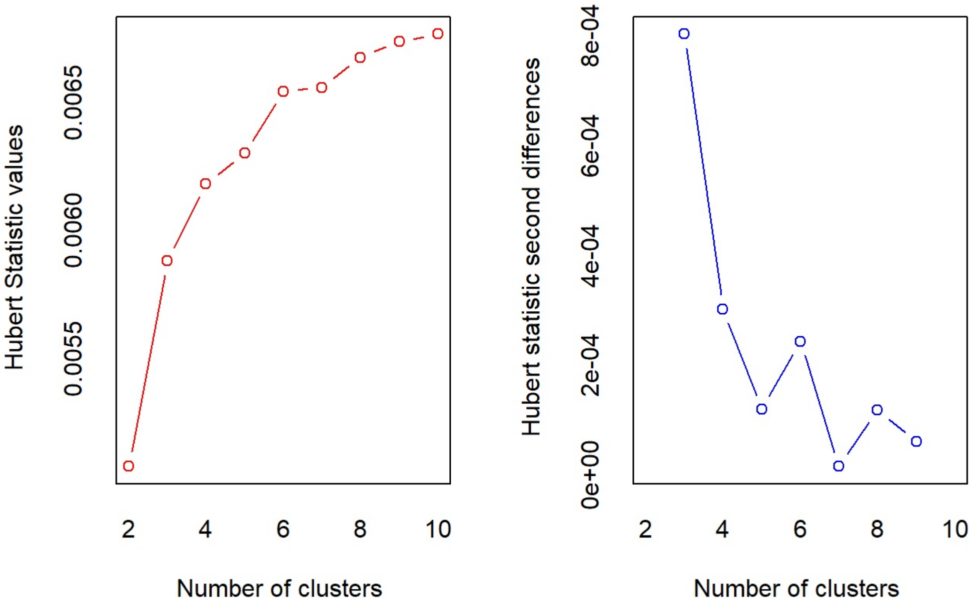
```
fviz_nbclust(df_copy_sc, kmeans, method = "gap_stat",nboot=50)+
  labs(title="Optimal number of clusters (Gap Statistic)")
```



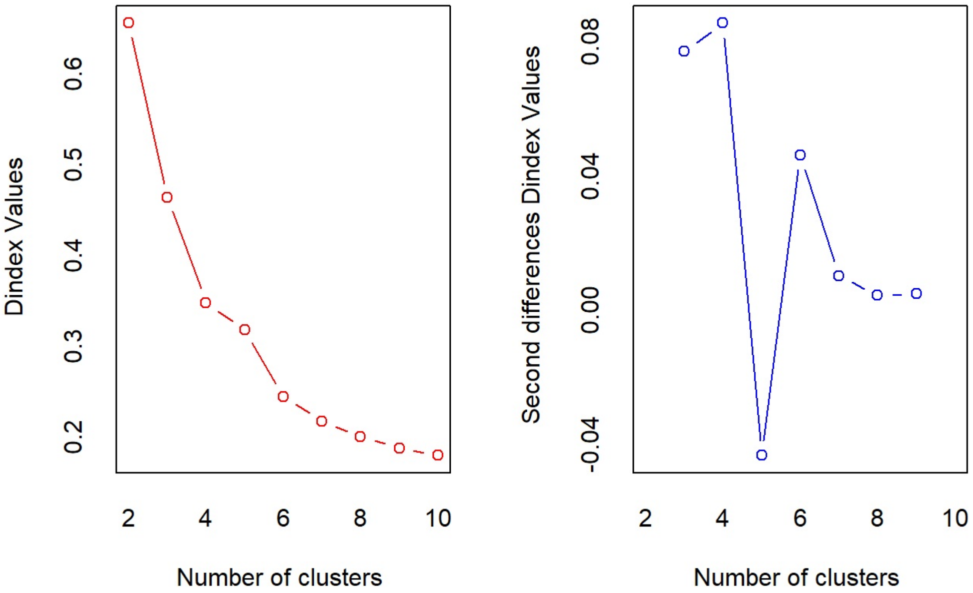
We can see in the Gap Statistics method, the peak point/optimum cluster value is 5.

Methods with combined functions

```
nb <- NbClust(df_copy_sc, distance = "euclidean", min.nc = 2, max.nc = 10, method = "kmeans", index="all")
```



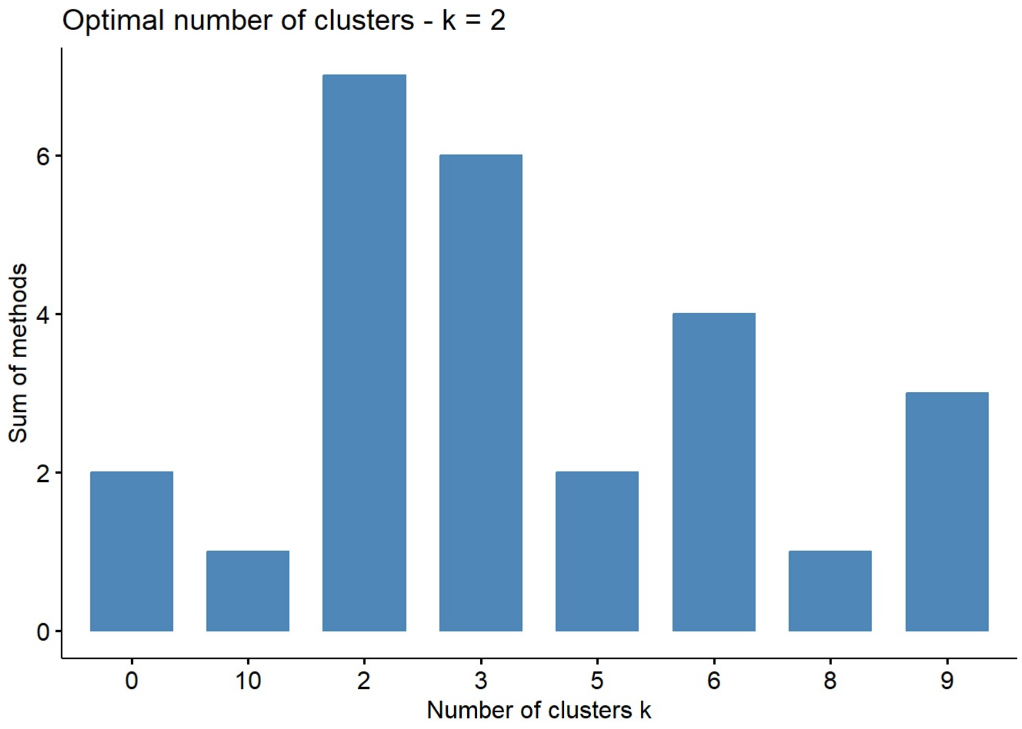
```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##       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



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 7 proposed 2 as the best number of clusters
## * 6 proposed 3 as the best number of clusters
## * 2 proposed 5 as the best number of clusters
## * 4 proposed 6 as the best number of clusters
## * 1 proposed 8 as the best number of clusters
## * 3 proposed 9 as the best number of clusters
## * 1 proposed 10 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

```
fviz_nbclust(nb)+ labs(x="Number of clusters k",y="Sum of methods")
```

```
## Among all indices:
## ===================
## * 2 proposed  0 as the best number of clusters
## * 7 proposed  2 as the best number of clusters
## * 6 proposed  3 as the best number of clusters
## * 2 proposed  5 as the best number of clusters
## * 4 proposed  6 as the best number of clusters
## * 1 proposed  8 as the best number of clusters
## * 3 proposed  9 as the best number of clusters
## * 1 proposed  10 as the best number of clusters
##
## Conclusion
## =========================
## * According to the majority rule, the best number of clusters is  2 .
```



Among all indices from above graph:

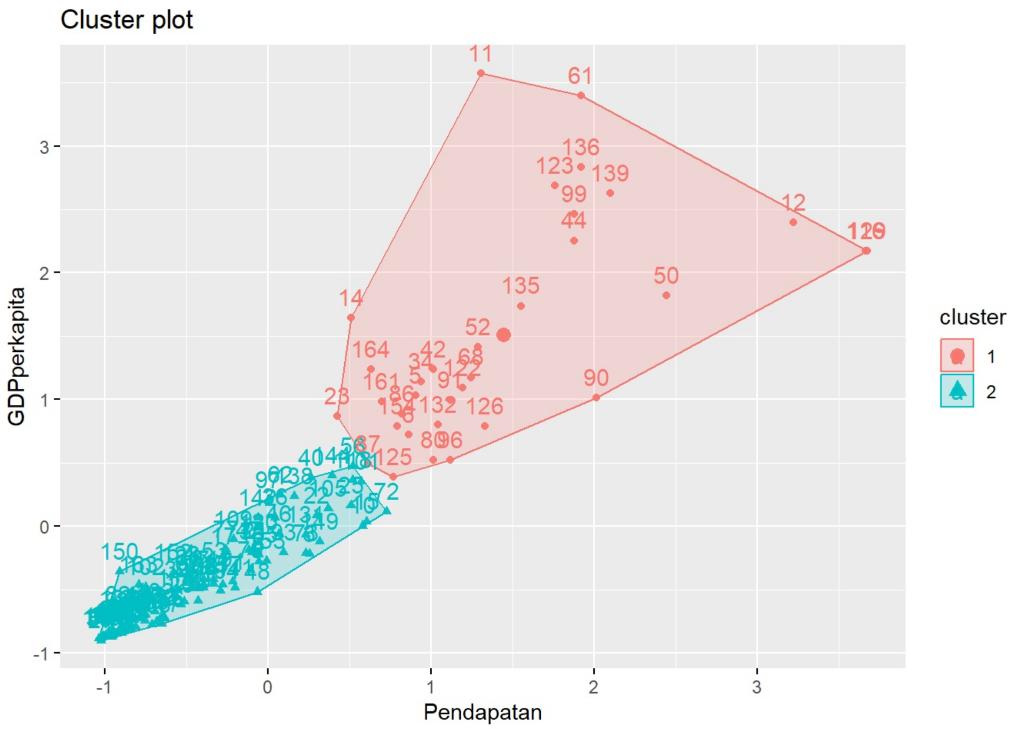
- 7 proposed 2 as the best number of clusters
- 6 proposed 3 as the best number of clusters
- 2 proposed 5 as the best number of clusters
- 4 proposed 6 as the best number of clusters
- 1 proposed 8 as the best number of clusters
- 3 proposed 9 as the best number of clusters
- 1 proposed 10 as the best number of clusters

Conclusion: According to the majority rule, the best number of clusters is 2

From the clustering charts above we will use the recommended number of clusters from the combined function method including the Silhouette Method where k=2

Clustering with KMeans (with combined functions, k=2)

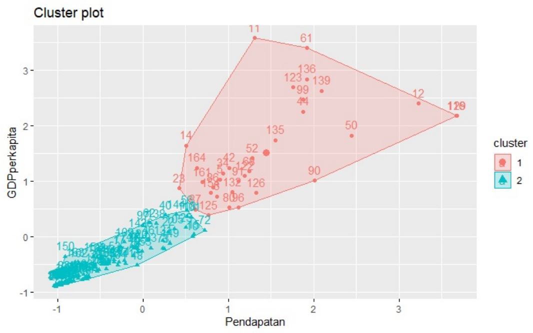
```
km <- kmeans(df_copy_sc, centers = 2, nstart = 25)
fviz_cluster(km, data = df_copy_sc)
```



Insight gained with Combined Functions, k=2:

- There are 2 clusters/groupings based on color, namely the red cluster and the blue cluster
- The blue cluster is countries with low and directly proportional Pendapatan and GDPperkapita
- The red cluster is countries with middle to high Pendapatan and GDPperkapita
- Our task is to find out which countries are included in the blue cluster with low Pendapatan and GDPperkapita

4. Recommendation



Because we use the Combined Functions where K=2. So we follow according to the insight that we have to look for which countries are included in the blue cluster where Pendapatan and GDPperkapita are low.

Here we will use 2 ways to take recommendations
1. The first way is to calculate the average from a dataset that is clean of outliers and then choose which country is less than that average 2. The second way is to find out which countries are based on the label kmeans on the cluster

4.1. Recommendation - Method 1

The steps used to find out which countries are eligible for assistance from HELP International:

1. Prepare a dataset that is clean from Missing Value and Outliers.
2. Find the average/mean of each column

3. Find which countries are less than the average Pendapatan and GDPperkapita
4. Sort the value of Pendapatan and GDPperkapita from smallest to largest as a priority
5. Take the top 5 countries with the smallest Pendapatan and GDPperkapita

```
df_copy <- cbind(df_copy, "cluster" = km$"cluster")

cara1 <- df_copy %>% drop_na() %>%
  filter(Pendapatan<mean(Pendapatan) & GDPperkapita<mean(GDPperkapita)) %>%
  arrange(Pendapatan,GDPperkapita)

head(cara1,5)
```

```
##                     Negara Kematian_anak Ekspor Kesehatan Impor Pendapatan
## 1         Congo, Dem. Rep.         116.0  41.10      7.91  49.6        609
## 2                  Liberia          89.3  19.10     11.80  92.6        700
## 3                  Burundi          93.6   8.92     11.60  39.2        764
## 4                    Niger         123.0  22.20      5.16  49.1        814
## 5 Central African Republic         149.0  11.80      3.98  26.5        888
##   Inflasi Harapan_hidup Jumlah_fertiliti GDPperkapita cluster
## 1   20.80          57.5             6.54          334       2
## 2    5.47          60.8             5.02          327       2
## 3   12.30          57.7             6.26          231       2
## 4    2.55          58.8             7.49          348       2
## 5    2.01          47.5             5.21          446       2
```

From the results above, recommendations for countries that are eligible for assistance can be given (in order from top to bottom as a priority scale):

1. Congo, Dem. Rep.
2. Liberia
3. Burundi
4. Niger
5. Central African Republic

4.2. Recommendation - Method 2

Now we will try the Combined functions method based on kmeans cluster label where our main focus is cluster = 2 which is a blue cluster

```
cara2 <- df_copy %>% drop_na() %>%
  filter(cluster==2 & Pendapatan<mean(Pendapatan) & GDPperkapita<mean(GDPperkapita)) %>%
  arrange(Pendapatan,GDPperkapita)

head(cara2,5)
```

```
##                     Negara Kematian_anak Ekspor Kesehatan Impor Pendapatan
## 1         Congo, Dem. Rep.         116.0  41.10      7.91  49.6        609
## 2                  Liberia          89.3  19.10     11.80  92.6        700
## 3                  Burundi          93.6   8.92     11.60  39.2        764
## 4                    Niger         123.0  22.20      5.16  49.1        814
## 5 Central African Republic         149.0  11.80      3.98  26.5        888
##   Inflasi Harapan_hidup Jumlah_fertiliti GDPperkapita cluster
## 1   20.80          57.5             6.54          334       2
## 2    5.47          60.8             5.02          327       2
## 3   12.30          57.7             6.26          231       2
## 4    2.55          58.8             7.49          348       2
## 5    2.01          47.5             5.21          446       2
```

From the results above, recommendations for countries that are eligible for assistance can be given (in order from top to bottom as a priority scale):

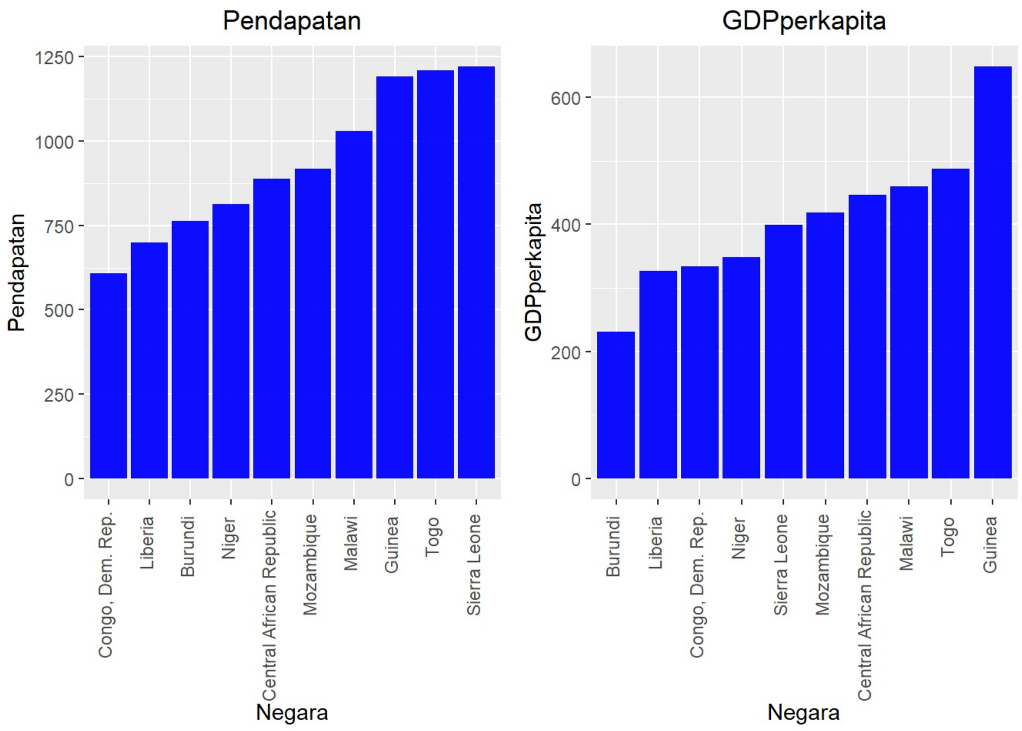
1. Congo, Dem. Rep.
2. Liberia
3. Burundi
4. Niger
5. Central African Republic

Following are the results of the Bar plot for each blue cluster according to Pendapatan and GDPperkapita

```
plot11 <- ggplot(head(cara2,10), mapping=aes(x=reorder(Negara,Pendapatan), y=Pendapatan))+
  geom_bar(stat='identity',fill="blue")+
  theme(plot.title = element_text(hjust = 0.5))+
  labs(title="Pendapatan",x="Negara",y="Pendapatan")+
  theme(axis.text.x = element_text(angle = 90, hjust = 0.95, vjust = 0.2))

plot12 <- ggplot(head(cara2,10), mapping=aes(x=reorder(Negara,GDPperkapita), y=GDPperkapita))+
  geom_bar(stat='identity',fill="blue")+
  labs(title="GDPperkapita",x="Negara",y="GDPperkapita")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(axis.text.x = element_text(angle = 90, hjust = 0.95, vjust = 0.2))

ggarrange(plot11, plot12, ncol = 2, nrow = 1)
```



5. Conclusion

Both Method 1 and 2 recommendations produce the same output. So in conclusion we can recommend the top 5 countries as priorities to get grants and notify the CEO of HELP International, namely:

1. Congo, Dem. Rep.
2. Liberia
3. Burundi
4. Niger
5. Central African Republic