CARLETON UNIVERSITY

STAT 5703- DATA MINING

PROJECT -WORLD HAPPINESS REPORT

Submitted To: -

Professor Dr. Shirley E. Mills
Associate Professor, Mathematics and Statistics
School of Mathematics and Statistics

Submitted By: -

Hemant Gupta (M.C.S.) - 101062246
(hemantgupta@cmail.carleton.ca)
Anurag Das (M. Eng. SYSC) – 101089268
(anuragdas@cmail.carleton.ca)
Manoj Kakarla (M. Eng. SYSC) -101071887
(manojkakarla@cmail.carleton.ca)

TABLE OF CONTENTS

1. INTRODUCTION

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. It is an essential process where intelligent methods are applied to extract data patterns.

In this project we are going to analyze the dataset using few methods based on our learning.

1.1. Dataset-World Happiness Report

The World Happiness Report is a landmark survey of the state of global happiness. The first report was published in 2012, the second in 2013, the third in 2015, and the fourth in the 2016 Update. The World Happiness 2017, which ranks 155 countries by their happiness levels, was released at the United Nations at an event celebrating International Day of Happiness on March 20th. Here we have data for last 3 years 2015 to 2017.

The attributes: GDP per Capita, Family, Life Expectancy, Freedom, Generosity, Trust Government Corruption describe the extent to which these factors contribute in evaluating the happiness in each country. The Dystopia Residual metric is the Dystopia Happiness Score (1.85) + the Residual value or the unexplained value for each country. If you add all these factors up, we get the happiness score, so it might be un-reliable to model them to predict Happiness Scores.

Dataset Link- https://www.kaggle.com/unsdsn/world-happiness

Dataset Parameters Description-

Attribute	**Description**	**Datatype**
Country	Name of the country	String
Region	Region the country belongs to	String
Happiness Rank	Rank of the country based on the Happiness Score.	Numeric
Happiness Score	A metric measured in 2015 by asking the sampled people the question: "How would you rate your happiness on a scale of 0 to 10 where 10 is the happiest"	Numeric
Economy (GDP per capita)	The extent to which GDP contributes to the calculation of the Happiness Score.	Numeric
Family	The extent to which Family contributes to the calculation of the Happiness Score	Numeric
Health (Life Expectancy)	The extent to which Life expectancy contributed to the calculation of the Happiness Score	Numeric
Freedom	The extent to which Freedom contributed to the calculation of the Happiness Score	Numeric

Trust (Government Corruption)	The extent to which Perception of Corruption contributes to Happiness Score	Numeric
Generosity	The extent to which Generosity contributed to the calculation of the Happiness Score	Numeric
Dystopia Residual	The extent to which Dystopia Residual contributed to the calculation of the Happiness Score.	Numeric
Year	Year data belongs to	Numeric

1.2. Overview

In this project, in section 2 we are going to perform data visualization on world happiness dataset. In section 3, we will perform dimension reduction and use that output in unsupervised learning. In section 4 we will perform data reduction and use the output for supervised learning. In section 5 we will perform, unsupervised learning and in section 6 we will perform supervised learning.

1.3. Contributions

Name	**Work Assignment**
Anurag Das	Data Selection, Programming and Analysis Writeup- Supervised Learning
Hemant Gupta	Introduction, Conclusion Programming and Analysis Writeup- Data visualization, Data Reduction and Unsupervised Learning Report Integration
Manoj Kakarla	Programming and Analysis Writeup -Dimension Reduction

1.4. Acknowledgment

This final project was supported by Professor. Dr. Shirley E. Mills. We thank her for providing insights and expertise that greatly assisted the final project. We also like to show our gratitude to her for sharing her pearls of wisdom with us during this project. We have implemented methods, we have learnt in class in completion of the project.

2. DATA VISUALIZATION

Data visualization is the concept of analyzing data before performing any mining operations. We have use many measures for visualization like GGobi (Scatterplot and Parallel plot), Coplot and boxplot methods for analysis of World Happiness Dataset.

We have Dataset in three formats –

1. World Happiness Dataset with combined 2015, 2016 and 2017. Here we use analyze different parameters participating in world happiness as Dataset for 2017 did not contain different Region with respect to country. We use plot for this analysis.
2. World Happiness Dataset with combined 2015 and 2016. Here we use colors for different regions and analyze the dataset using GGobi, Boxplot for data analysis with respect to region and Year.
3. In these datasets Happiness Rank and Happiness Score are inversely proportional. Higher the score lowers the rank.
4. Please find the analysis of for each method below.

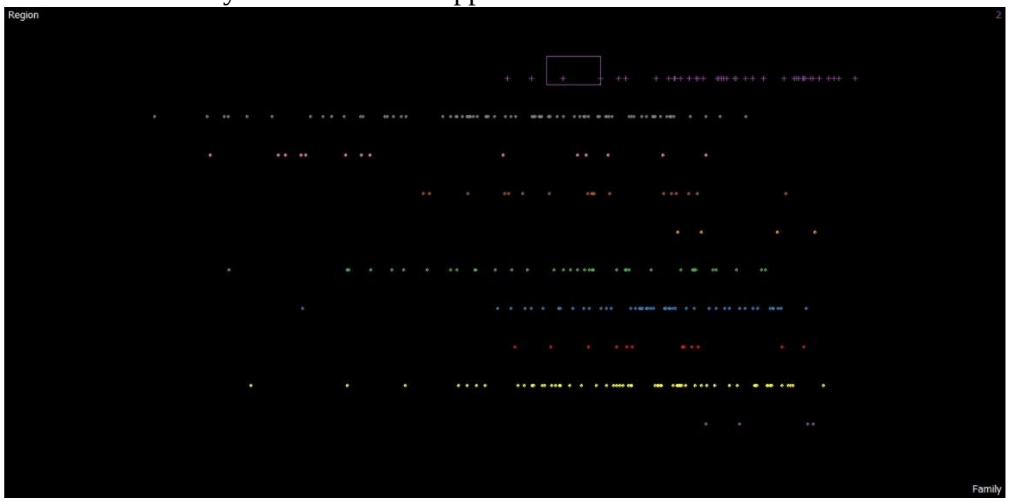
2.1. GGobi Analysis-(Dataset Year-2015 and 2016)-

For the GGobi Analysis we convert he Region attribute of dataset to the numeric.

[1]Australia and New Zealand(violet)[2]Central and Eastern Europe (yellow) [3] Eastern Asia(Red) [4] Latin America and Caribbean(blue) [5] Middle East and Northern Africa(Green) [7] North America(Orange) [8] Southeastern Asia(Dark Orange) [9] Southern Asia (Purple) [10] Sub-Saharan Africa (Grey) [11] Western Europe(Violet with Plus sign) [6] na

2.1.1. Scatterplot-

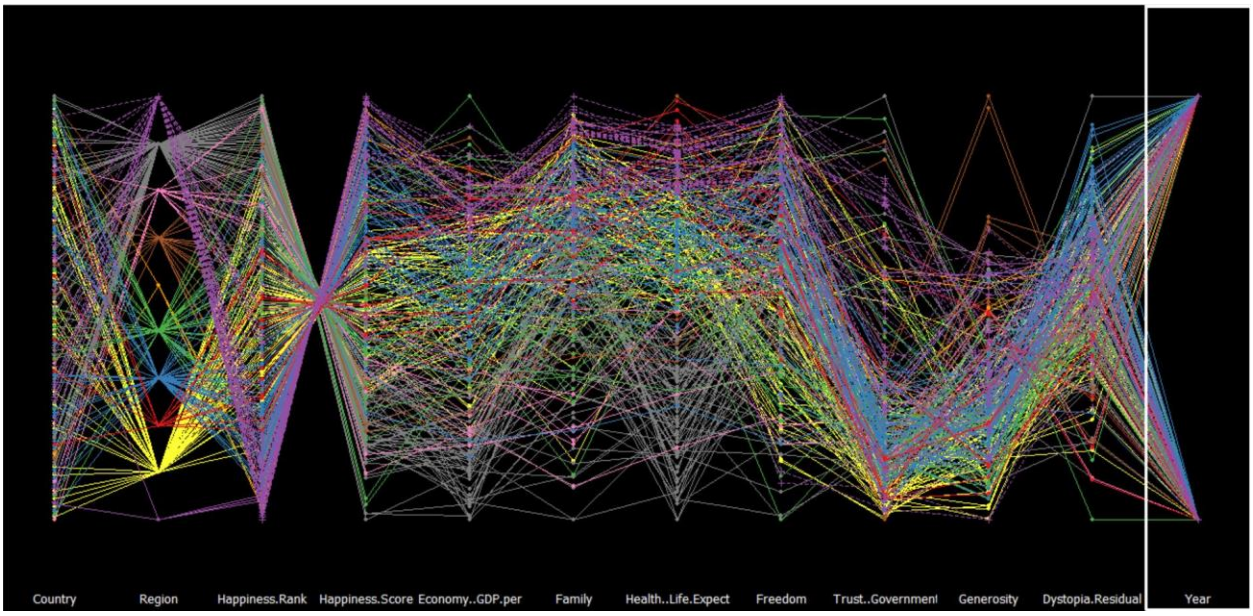
We found that there are 10 different regions available in the dataset and Western Europe has highest value of Family contribution to happiness score.



2.1.2. Parallel Plot-

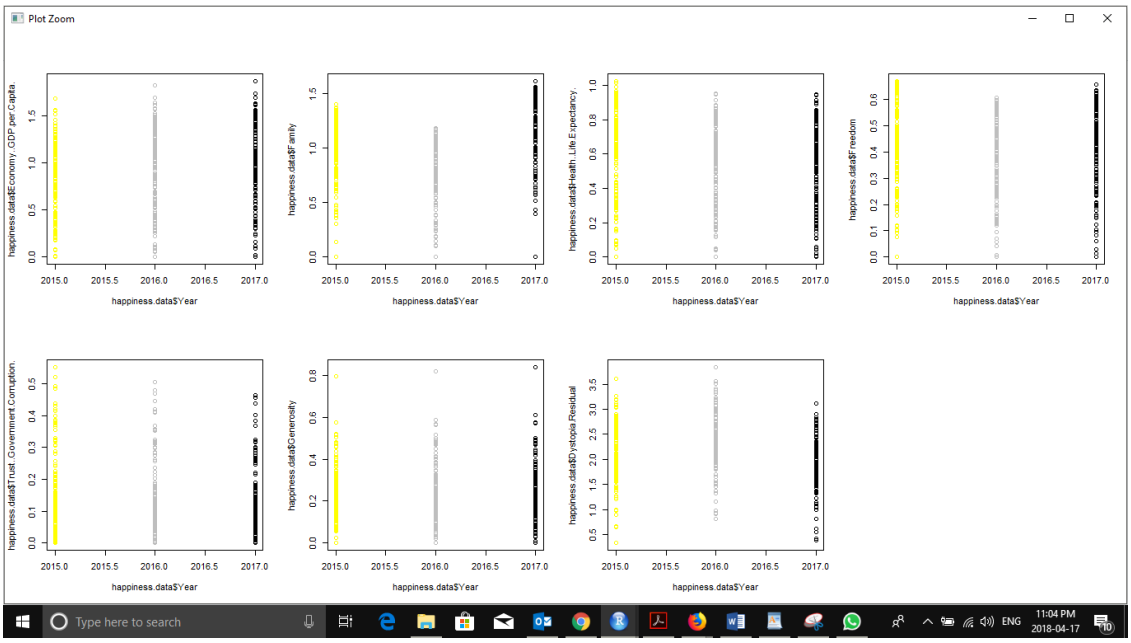
a. Countries in region Australia and New Zealand has highest happiness score.
b. One of the country of Middle East and Northern Africa has highest GDP per Capita.
c. One of the country of Western Europe has highest Family Contribution

d. One of the country of Southeastern Asia has high life expectancy and generosity value.
e. Sub-Saharan Africa has highest dystopia residual value.



2.2. Plot Analysis- (Attributes for Dataset Year-2015,2016,2017)-

a. Not significant change seen for highest GDP per capita, health life expectancy, freedom and generosity for all the three consecutive years.
b. Highest Value of Family contribution for overall happiness is high in 2017 and lowest in 2016.
c. Highest Value of trust over government was decreasing gradually year by year.
d. Highest value Dystopia residual is decrease in Year 2017.



2.3. Boxplot Analysis-(Dataset Year-2015 and 2016)

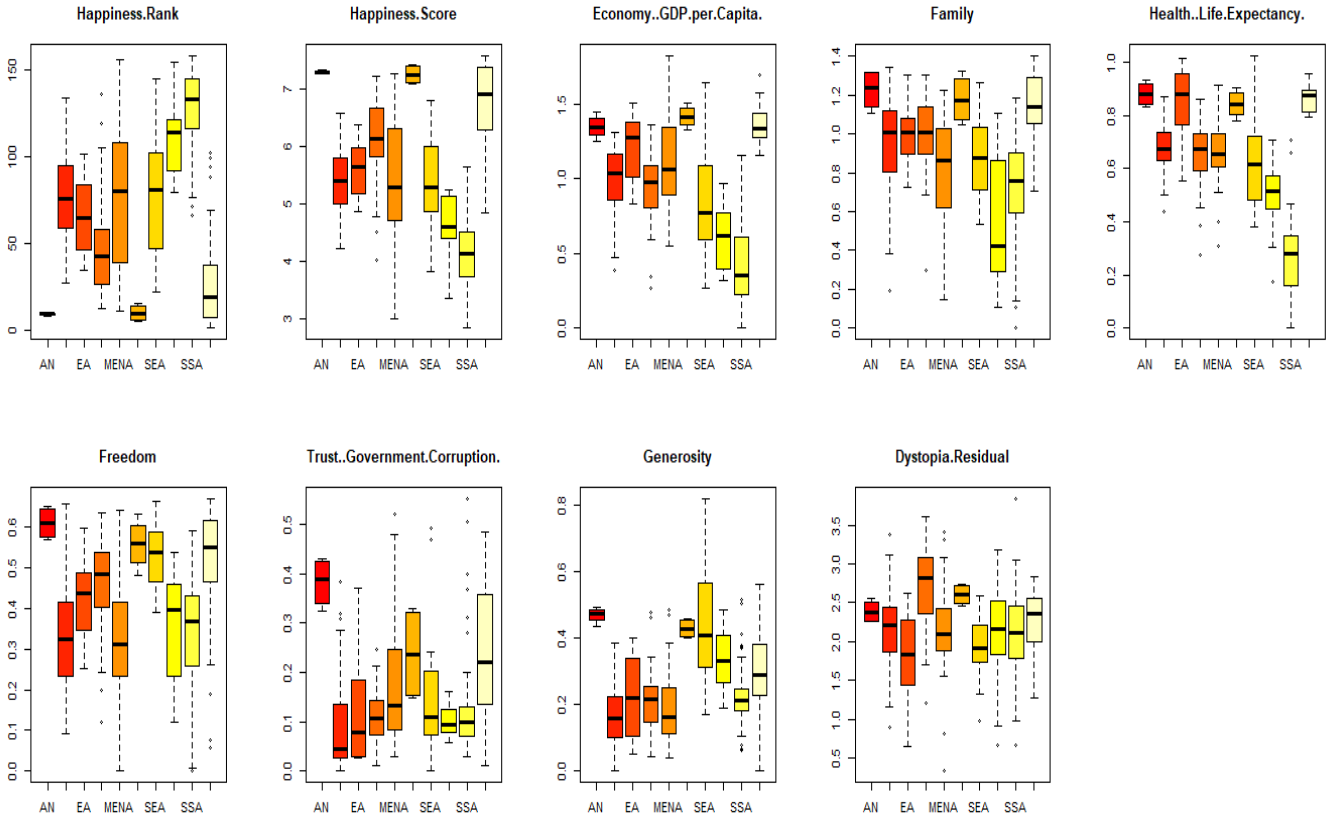
2.3.1. Region Analysis-

a. Countries in region Australia & New Zealand, North-America has happiness ranking below 25 and for Western Europe has happiness average raking below 25. All these Regions has high GDP and family contribution and higher life expectancy which participate in high happiness score.

b. Countries in Southern Asia and Sub-Suburban Africa has average happiness ranking above 100. These regions have below 1 GDP per Capita, Lowest average family contribution to happiness and lowest life expectancy. SSA has lowest health life expectancy in all regions.

c. Countries in region Australia & New Zealand have higher trust in government regarding corruption but even though countries in region North-America and Western Europe has rank less than 50 but have only average of approx. 50%(i.e. 0.25 out of 0.5) trust on government regarding corruption.

d. Countries in region Australia & New Zealand, North-America and Western Europe has average ranking below 25 and has high value of freedom. But countries in region Southeastern Asia has high freedom but ranks are between 50 to 100.

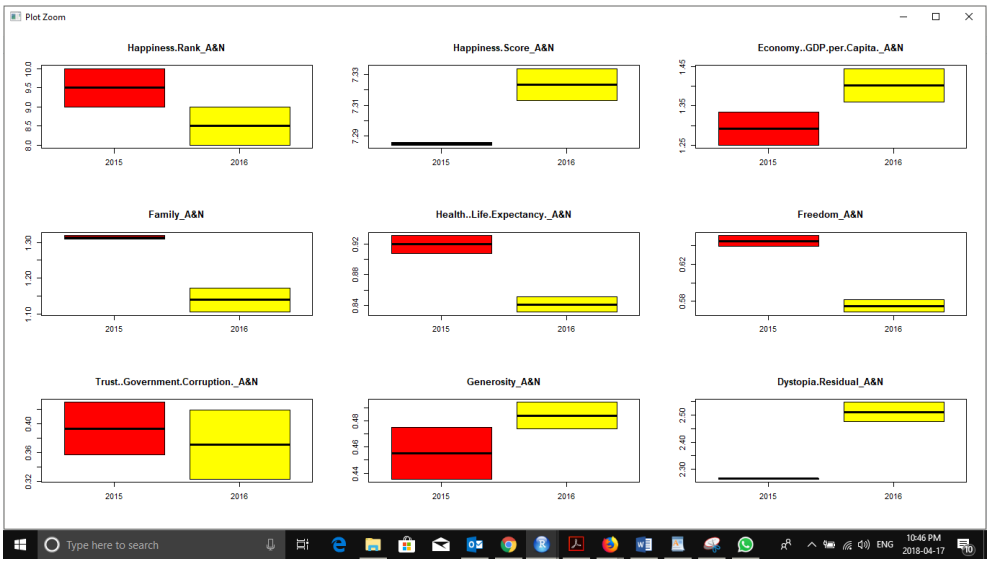


2.3.2. Region-Year Analysis

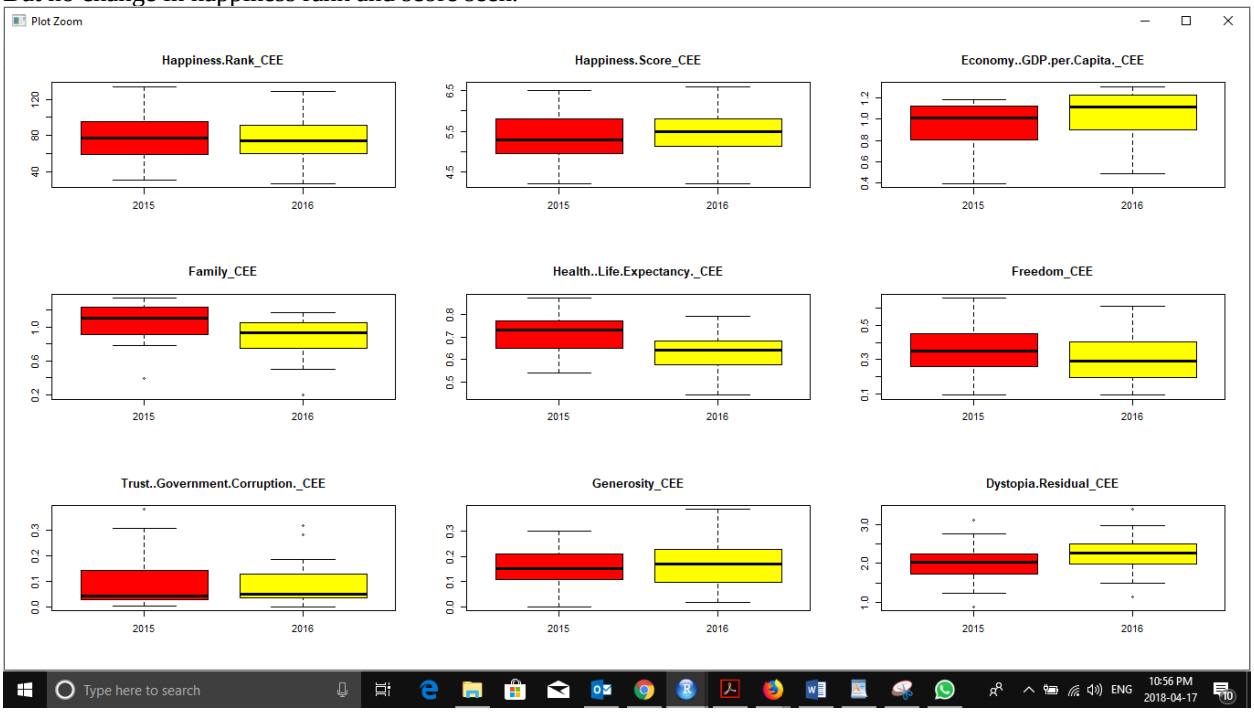
a. Australia and New Zealand-Increase in happiness score, generosity, GDP per capita and dystopia residual over year 2015 to 2016.

Decrease in life expectancy, family contribution, freedom and trust over the year 2015 to 2016.

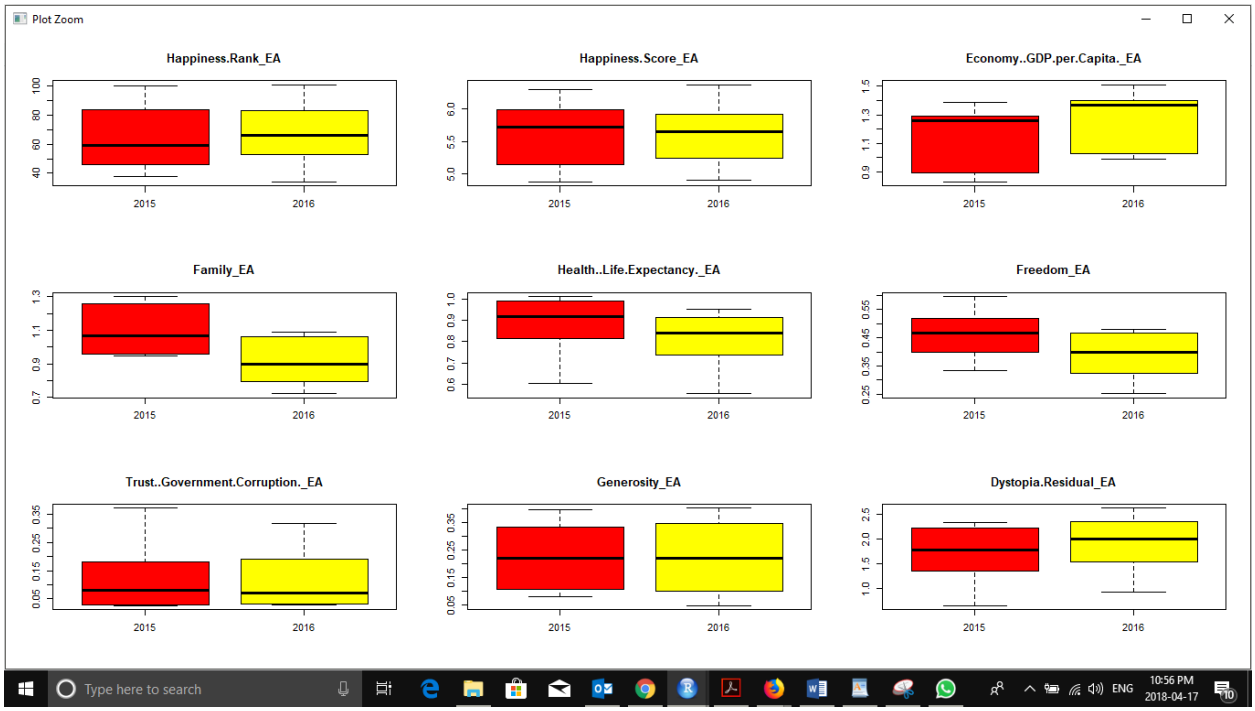
Leads to happiness rank for countries in Australia and New Zealand region improved.



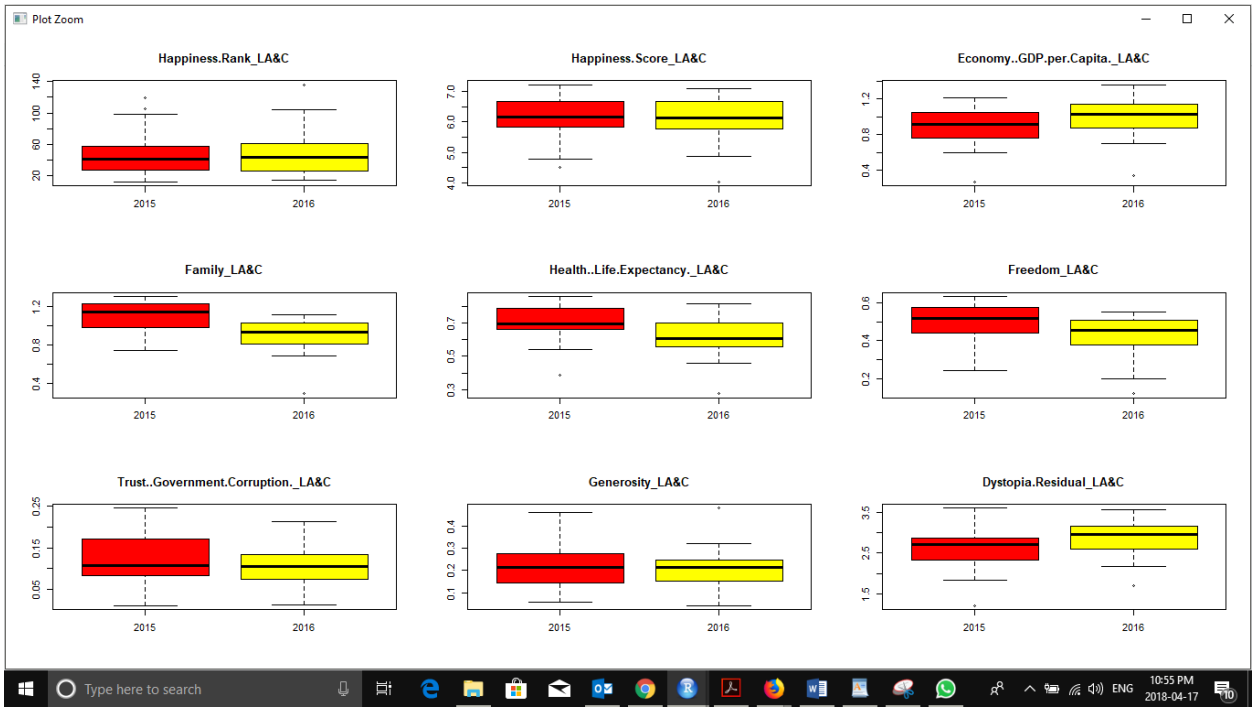
b. Central and Eastern Europe-There is not much change seen on all the attributes over year 2015 and 2016. Average value of GDP and dystopia residual increases and family contribution and life expectancy decreases. But no change in happiness rank and score seen.



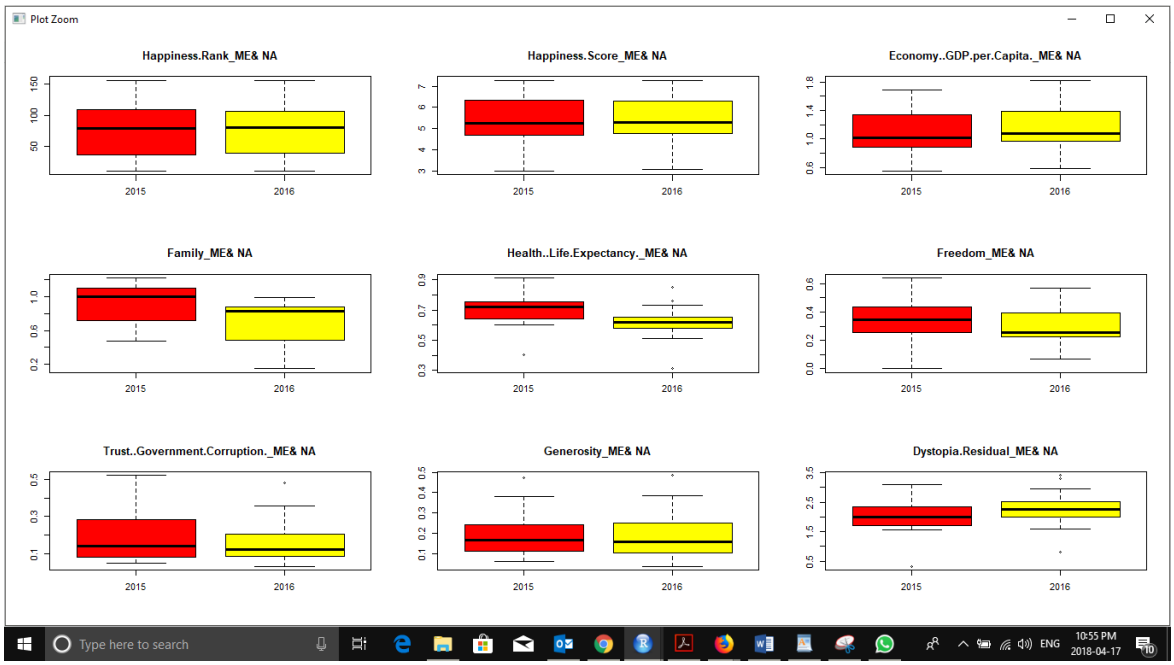
c. Eastern Asia-Average Value of Family contribution, life expectancy and freedom decrease over year 2015 to 2016 but GDP per capita increases.
Causes range for happiness rank decreases.



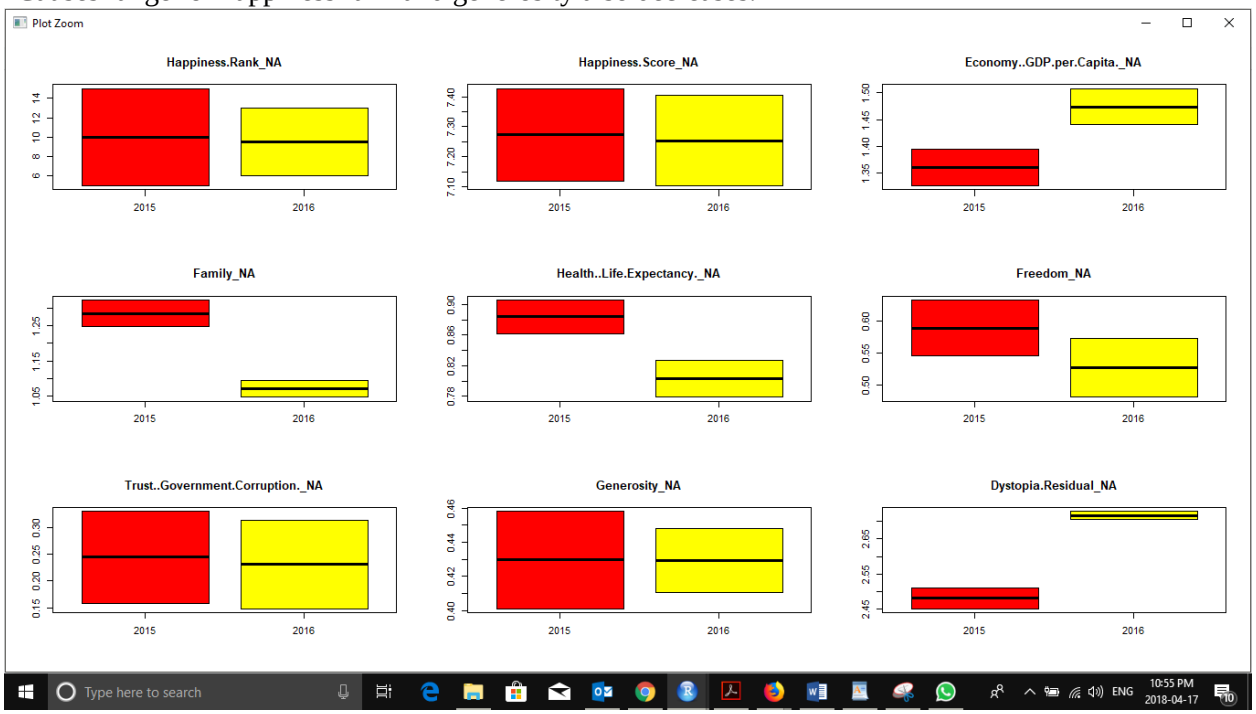
d. Latin America and Caribbean-Average value of GDP and dystopia residual increases and family contribution and life expectancy and freedom decreases. But no change seen for happiness rank and score.



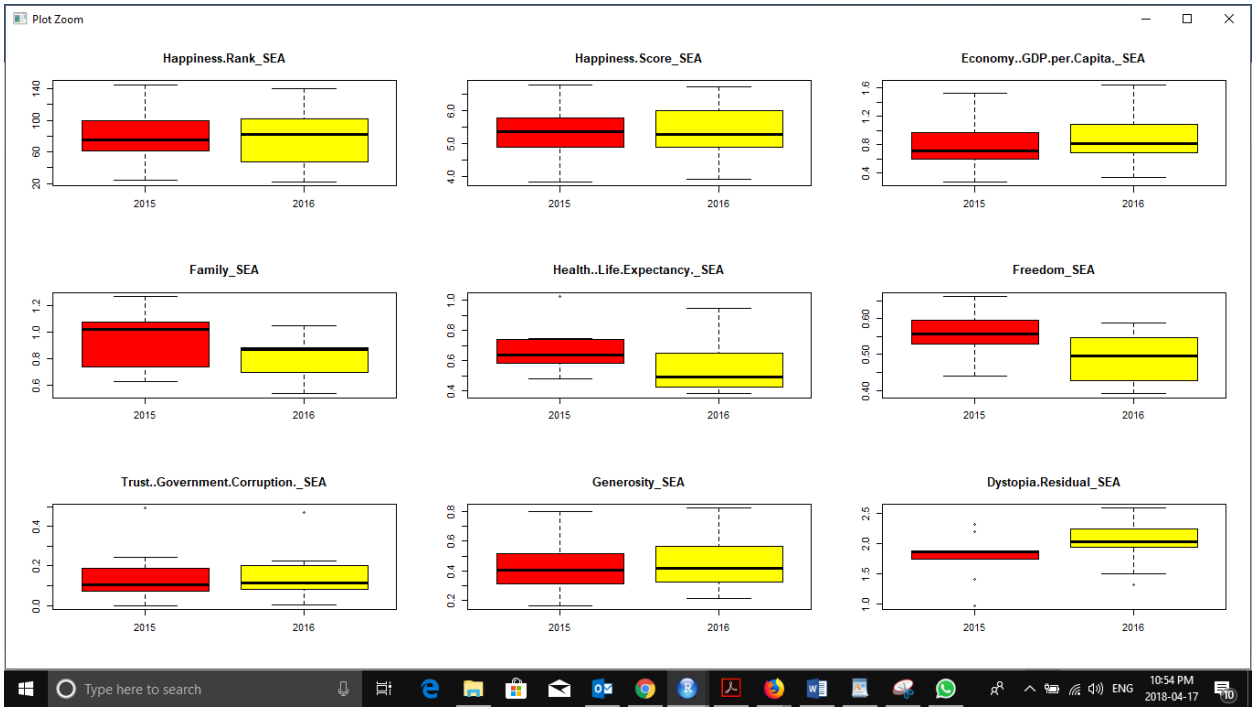
e. **Middle East and Northern Africa-** Not much change seen for happiness rank or score even though average value of family contribution and life expectancy and freedom decreases.



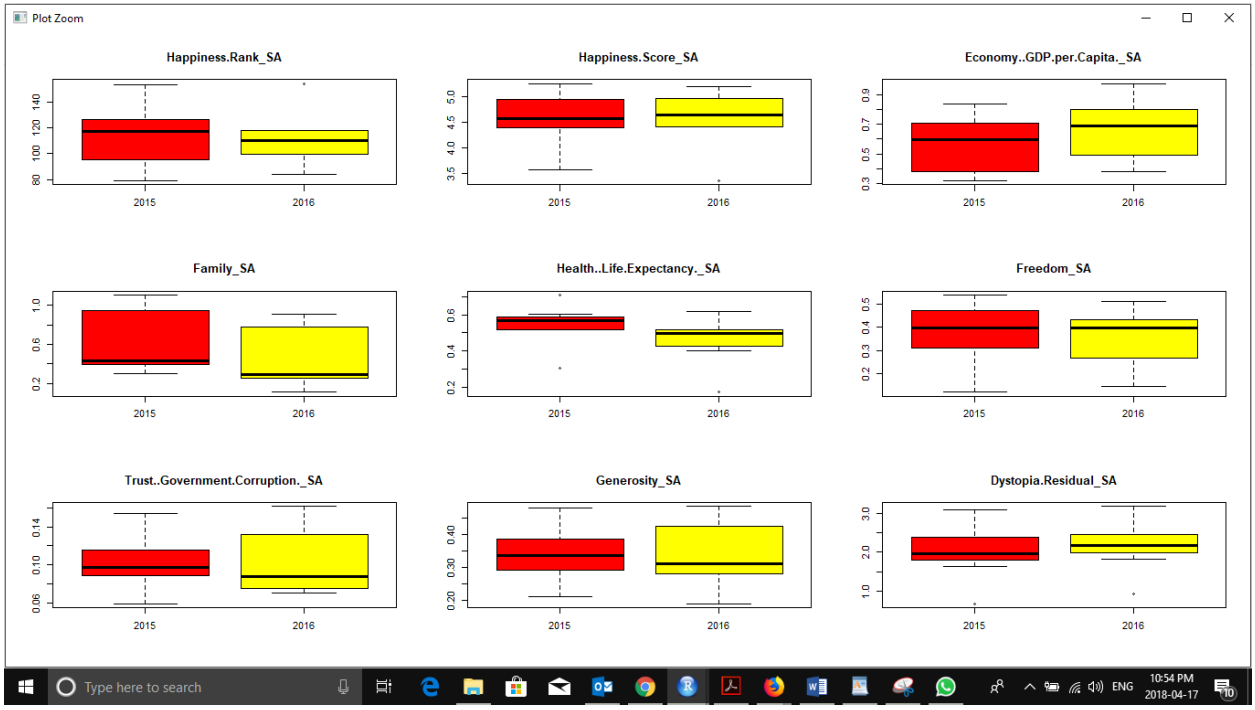
f. **North America**- Heavy increase in GDP and dystopia residual values over year 2015 to 2016 and decrease in the value of family contribution, life-expectancy and freedom.
Causes range for happiness rank and generosity also decreases.



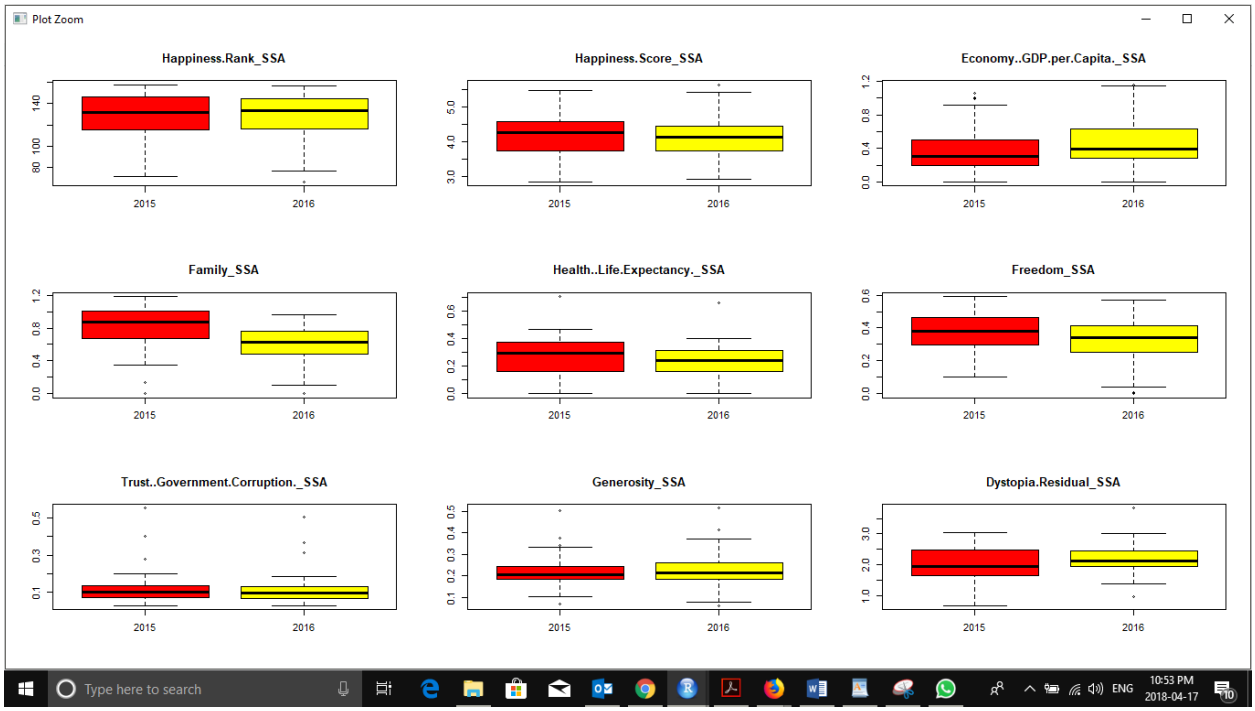
g. **Southeastern Asia**-Average value of family contribution to happiness, life expectancy, and freedom decrease due to which range for happiness rank and score increase.



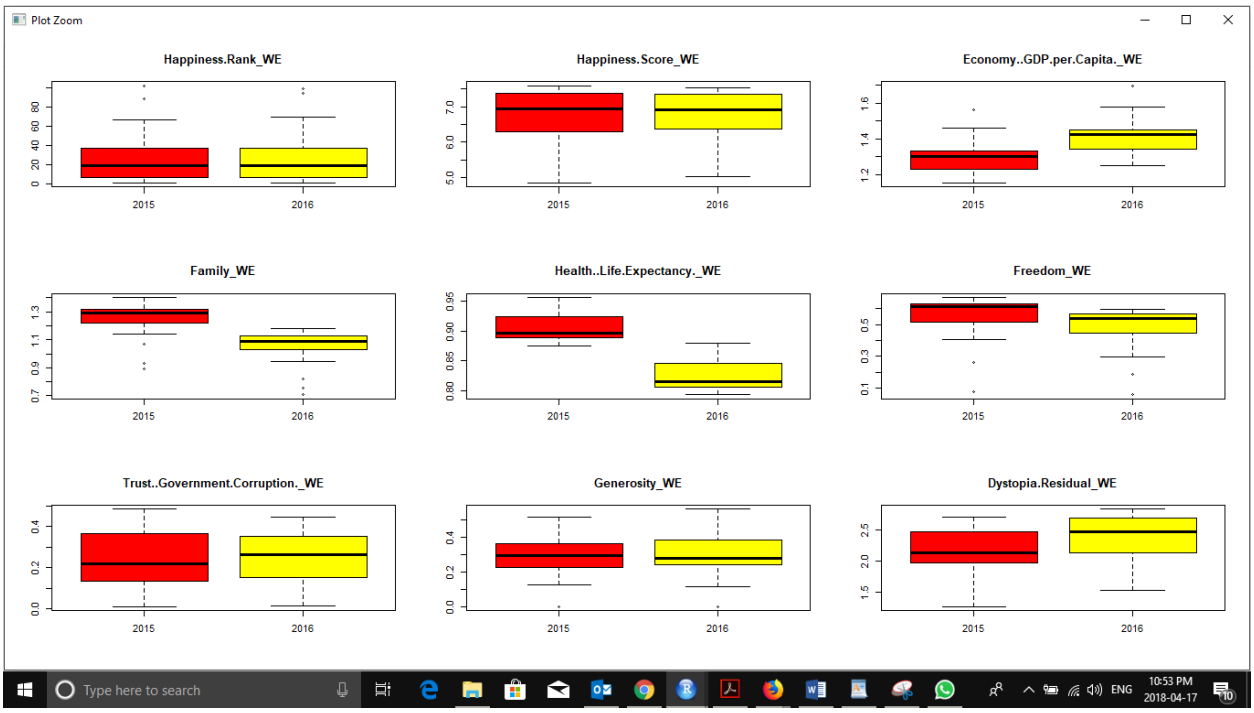
h. Southern Asia- Range for happiness rank decreases due to average GDP value increases and average contribution of Family, life expectancy and freedom and trust over government decrease over year from 2015 to 2016.



i. Sub-Saharan Africa- Family contribution and freedom decreases over the year from 2015 to 2016 but has not impact on overall happiness rank.



j. **Western Europe**- GDP increases and Family contribution and life expectancy decrease but no change seen with respect to happiness rank and score.



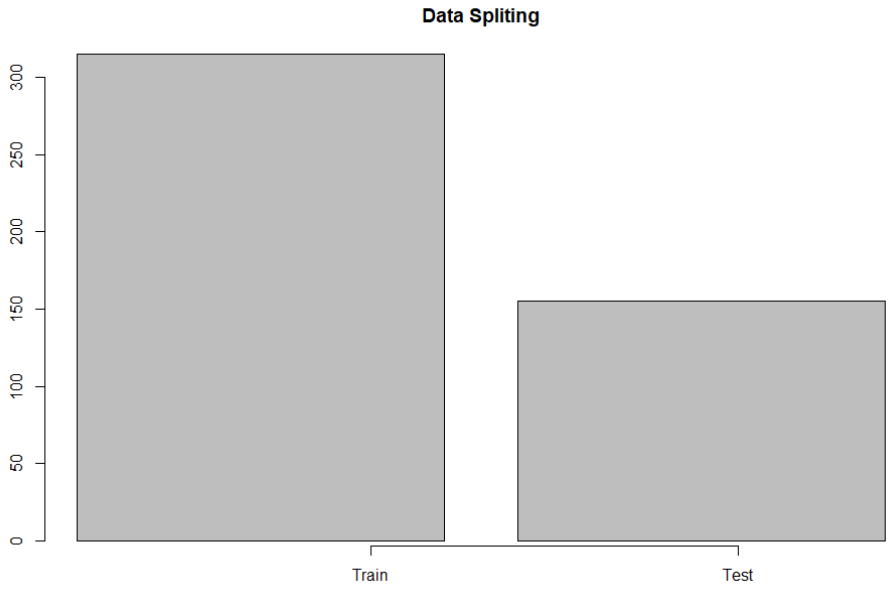
3. DIMENSION REDUCTION

Dimension reduction is the process of **reducing** the number of random variables under consideration by obtaining a set of principal variables. It can be divided into feature selection and feature extraction.

Analysis-

1. We have done dimension reduction using PCA and ICA.
2. With three components of the PCA for Raw whole, train and test dataset we get the approx. 74% of the cumulative proportion and after that there is not large difference in further components.
3. We have also performed the ICA using 3 components, so that we can compare the result with respect to PCA.
4. We will be using these three components derived from PCA and ICA for complete, train and test dataset for the unsupervised learning.

Training and Test Dataset- We have divided the dataset entries for year 2015 and 2016 in the training set. And use the data for Year 2017

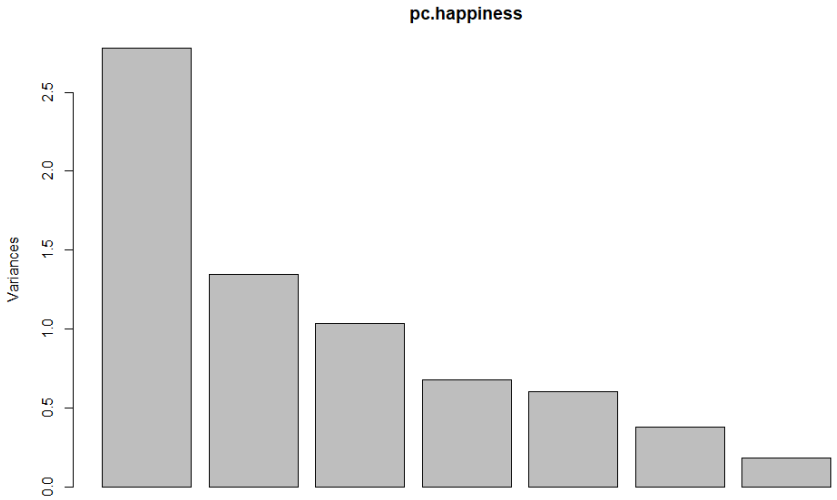


3.1. PCA-Dimension Reduction

3.1.1. Standard Complete Dataset

For whole dataset we took 3 components for the analysis.

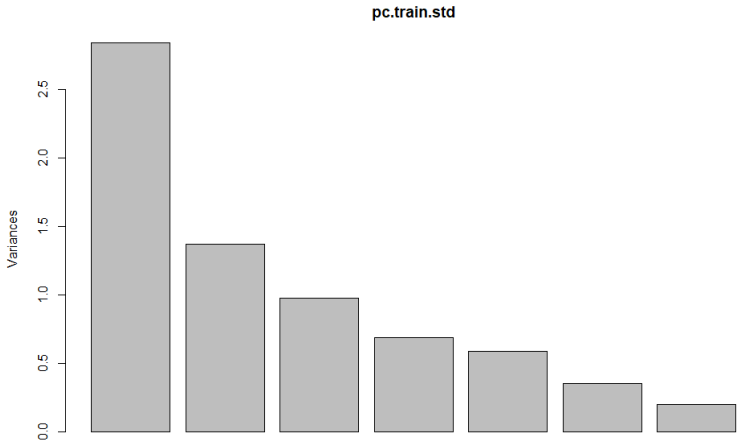
```
> summary(pc.happiness)
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
Standard deviation     1.667 1.1597 1.0160 0.82278 0.77829 0.61610 0.42612
Proportion of Variance 0.397 0.1921 0.1475 0.09671 0.08653 0.05423 0.02594
Cumulative Proportion  0.397 0.5891 0.7366 0.83330 0.91983 0.97406 1.00000
> plot(pc.happiness)
> # First  principal components
> happiness.pc <- data.frame(pc.happiness$x[,1:3])
> head(happiness.pc)
        PC1        PC2       PC3
1 -3.471232 -0.7178181 1.1912246
2 -2.671132 -0.2656504 0.7936938
3 -3.473580 -1.2833877 1.2555213
4 -3.317022 -0.8211403 0.9884043
5 -3.120203 -1.2579827 0.8422480
6 -3.027095 -0.4570216 1.3797201
>
```



3.1.2. Standard Train Dataset

For training dataset, we are taking 3 components for the analysis

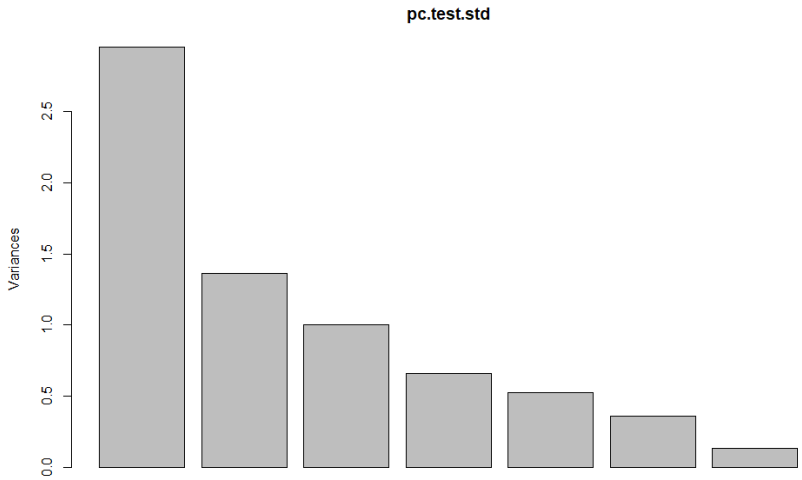
```
> summary(pc.train.std)
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
Standard deviation     1.6846 1.1697 0.9876 0.82691 0.76465 0.59522 0.44276
Proportion of Variance 0.4054 0.1955 0.1393 0.09768 0.08353 0.05061 0.02801
Cumulative Proportion  0.4054 0.6008 0.7402 0.83785 0.92138 0.97199 1.00000
> plot(pc.train.std)
>
> # First  principal components
> happiness.train.pc <- data.frame(pc.train.std$x[,1:3])
> head(happiness.train.pc)
        PC1        PC2       PC3
1 -3.664559 -0.5904899 0.7676974
2 -2.934885 -0.1469376 0.6967399
3 -3.660465 -1.1403405 0.8940800
4 -3.510354 -0.7059039 0.6585788
5 -3.322191 -1.1471292 0.6185194
6 -3.222442 -0.3138760 0.9484910
>
```



3.1.3. Standard Test Dataset

For Test Dataset, we took 3 components after analysis.

```
> summary(pc.test.std)
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
Standard deviation     1.7186 1.1688 1.0009 0.81342 0.72322 0.59949 0.3666
Proportion of Variance 0.4219 0.1951 0.1431 0.09452 0.07472 0.05134 0.0192
Cumulative Proportion  0.4219 0.6171 0.7602 0.85474 0.92946 0.98080 1.0000
> plot(pc.test.std)
>
> # First  principal components
> happiness.test.pc <- data.frame(pc.test.std$x[,1:3])
> head(happiness.test.pc)
          PC1        PC2       PC3
316 -3.208847 -0.7611688 0.9495370
317 -3.319268 -1.1255686 1.1506688
318 -2.802781 -0.5812645 0.8593260
319 -3.303680 -0.5700763 0.9615492
320 -3.107317 -0.4417074 1.3318639
321 -2.801775 -1.1687695 0.9216539
```



3.2. ICA-Dimension Reduction

3.2.1. Whitened Complete Dataset

For whole Data set we are taking 3 components after ICA for analysis after dimension reduction.

```
> happiness.ica<-happiness.white.ica$S
> #Estimated Source Matrix
> head(happiness.ica)
          [,1]       [,2]        [,3]
[1,] 0.5816709 -0.5456813 -0.05727165
[2,] 0.4543591 -1.2230504 -0.18501055
[3,] 0.5744976 -0.2765747  0.32143975
[4,] 0.9948967 -0.1894481 -0.07895725
[5,] 0.8742937 -0.8470915  0.34770381
[6,] 0.2154683 -0.3685356 -0.05321664
>
```

3.2.2. Whitened Train Dataset

For Training Data set we are taking 3 components after ICA for analysis after dimension reduction.

```
> #Estimated Source Matrix
> happiness.train.ica<-happiness.train.white.ica$S
> head(happiness.train.ica)
        [,1]      [,2]       [,3]
1 2.2436036 0.7455082 -0.8097095
2 0.1909327 0.7377022 -0.6876554
3 2.9638721 0.3705356 -0.4685667
4 1.6714765 0.2171257 -1.0487543
5 1.6924391 0.3935870 -0.6483589
6 2.2560864 0.8524120 -0.5521719
>
```

3.2.3. Whitened Test Dataset

For Test Data set we are taking 3 components after ICA for analysis after dimension reduction.

```
> #Estimated Source Matrix
> happiness.test.ica<-happiness.test.white.ica$S
> head(happiness.test.ica)
            [,1]       [,2]       [,3]
316 -0.42867010  0.3273619 0.38210165
317 -0.05027672  0.2469488 0.28640766
318 -0.51751469 -0.1829260 1.42726665
319  0.40585853  0.3658902 0.36256630
320  0.73903542  0.5330505 0.02178368
321 -0.82269813  0.3236262 1.25303289
>
```

4. DATA REDUCTION

Data reduction is the transformation of numerical or alphabetical digital information derived empirically or experimentally into a corrected, ordered, and simplified form. The basic concept is the reduction of multitudinous amounts of data down to the meaningful parts.
We are going to use the Reduced Dataset for Supervised Learning.

Analysis-

1. We have done clustering on complete Raw, Standardize and Whitened dataset. And we got 3 is the optimal cluster value for the raw and standardize dataset and whitened dataset it is 6.
2. Using histogram for the optimal cluster for Raw dataset we found that, Group 1 which is group is countries with medium happiness score is overlap by the Group 2 and Group 3. There is no proper separation between the happiness score based on clusters.
3. Using histogram for standardize dataset we found that there is all three groups of clusters are properly visible.
4. In Raw Dataset, Cluster with highest happiness has 20 countries which has happiness score below median value. And Cluster with least happiness has 10 countries which has happiness score above median.

```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest[happiest$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 178
> nrow(happiest[happiest$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 20
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest[least_happiest$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 10
> nrow(least_happiest[least_happiest$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 139
>
```

5. In Standardize Dataset, Cluster with highest happiness has 3 countries which has happiness score below median value. And Cluster with least happiness has 3 countries which has happiness score above median.

```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest.std[happiest.std$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 80
> nrow(happiest.std[happiest.std$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 3
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest.std[least_happiest.std$Happiness.Score > median(happiness.data$Happiness.Score), ]
)
[1] 3
> nrow(least_happiest.std[least_happiest.std$Happiness.Score < median(happiness.data$Happiness.Score), ]
)
[1] 148
>
```

6. We choose standardize dataset Clusters for data reduction of the complete Raw Dataset. We took 2/3 from each cluster. The median value of each cluster in standardize dataset is

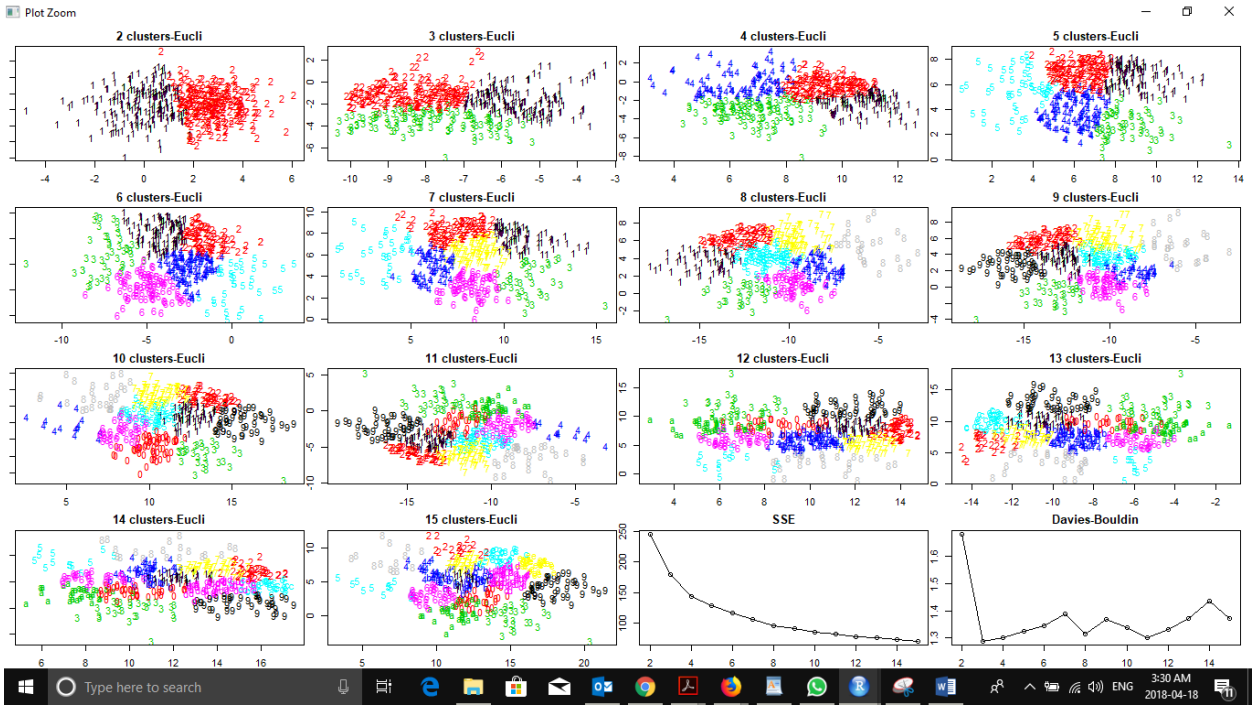
```
> med1<-median(happiness.std.new$Happiness.Score[happiness.std.new$cluster == 1])
> med2<-median(happiness.std.new$Happiness.Score[happiness.std.new$cluster == 2])
> med3<-median(happiness.std.new$Happiness.Score[happiness.std.new$cluster == 3])
> print(paste("Median Values-> Cluster1 = ", med1,", Cluster2 = ",med2,", Cluster3 = ",med3, sep=
"",))
[1] "Median Values-> Cluster1 = 5.6735, Cluster2 = 6.937, Cluster3 = 4.219"
>
```

4.1. Raw Dataset- Clustering

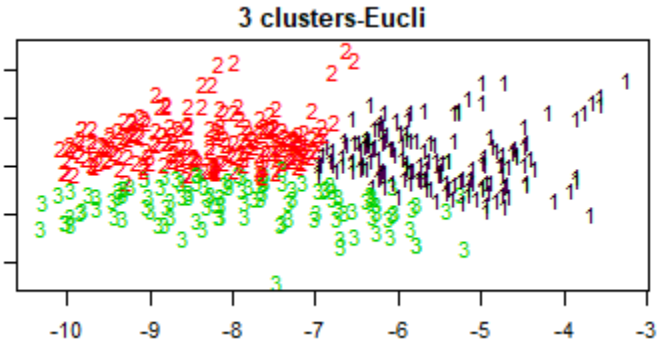
Cluster Analysis from range 2 to 15 for Raw Happiness Data Set.

In Cluster analysis, in this dataset we do not have any attributes for classification of data, therefore we have assigned numbers and colors based on the Clusters.

Optimal Value of Whole Raw Dataset from Davis Bouldin is 3.



- **Optimal Cluster Value is 3-**



- **Best Seed and Centroids for the Cluster-**

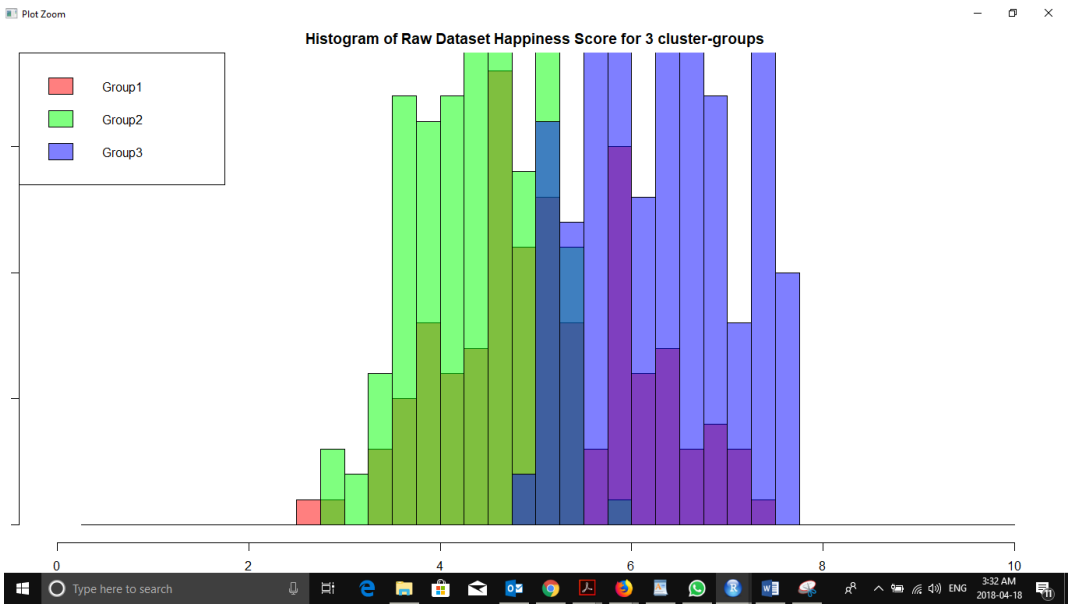
```
> km <- clustering_euclidean(happiness.data[, 5:11],happiness.data, 3)
[1] "Best Seed for Cluster Size  3 is  2"
[1] "Total Wrong in Cluster Size  3 is  0"
[1] "Centroids for Cluster Size  3 are :"
  Economy..GDP.per.Capita.    Family Health..Life.Expectancy.   Freedom
1                0.6167850 0.8247377                0.4091893 0.3390771
2                1.2880158 1.1969299                0.7621511 0.4639746
3                0.7718376 0.8829057                0.5199678 0.3917268
  Trust..Government.Corruption. Generosity Dystopia.Residual
1                     0.1074311  0.2500393          1.668095
2                     0.1796273  0.2543724          2.057470
3                     0.0994064  0.2113085          2.747418
K-means clustering with 3 clusters of sizes 163, 191, 116
```

- **Sum of Squares –**

```
Within cluster sum of squares by cluster:
[1] 73.65755 59.79762 46.50671
 (between_SS / total_SS =  45.5 %)
```

- **Cluster Analysis-**

• Distribution of High Score for Raw DataSet With optimal Value of Cluster Size from Davies Bouldin is 3- > There are overlap with respect to all cluster group.



- Happiest Cluster Group and Least Happiest Cluster Group- Country with Highest Happiness Score in Group G3 is Switzerland(7.587) and with Least Happiness Score in Cluster Group G2 is Tongo(2.839).

```
> #Cluster Size
> print(km$size)
[1] 123 149 198
>
> print(paste("RAW-Happiest Group is g", top, sep=""))
[1] "RAW-Happiest Group is g3"
>
> print(paste("RAW-Least Happiest Group is g", bottom, sep=""))
[1] "RAW-Least Happiest Group is g2"
>
>
> head(happiest[order(happiest$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
1   Switzerland           7.587
2       Iceland           7.561
316      Norway           7.537
3       Denmark           7.527
159     Denmark           7.526
4        Norway           7.522
>
> tail(least_happiest[order(least_happiest$Happiness.Score, decreasing=TRUE), ])
    Country Happiness.Score
155   Benin           3.340
314   Syria           3.069
156   Syria           3.006
157 Burundi           2.905
469 Burundi           2.905
158    Togo           2.839
>
```

- Countries above Median Value of Happiness Score in Least Happiness Cluster Group and Countries below Median values of Happiness Score of Happiest Cluster Group.

```
> print("Median Happiness Score for Whole Dataset-")
[1] "Median Happiness Score for Whole Dataset-"
> (median(happiness.test$Happiness.Score))
[1] 5.279
>
> head(happiest[least_happiest$Happiness.Score > median(happiness.data$Happiness.Score), ])
         Country Happiness.Score
1    Switzerland           7.587
2        Iceland           7.561
3        Denmark           7.527
68       Algeria           5.605
70  Turkmenistan           5.548
188        Malta           6.488
> head(happiest[happiest$Happiness.Score < median(happiness.data$Happiness.Score), ])
                  Country Happiness.Score
80              Azerbaijan           5.212
83              Montenegro           5.192
86                 Romania           5.124
87                  Serbia           5.123
96  Bosnia and Herzegovina           4.949
102                 Greece           4.857
>
```

- Number of Countries in Happiest Group and Least Happiest Cluster Group above and below Median Value-

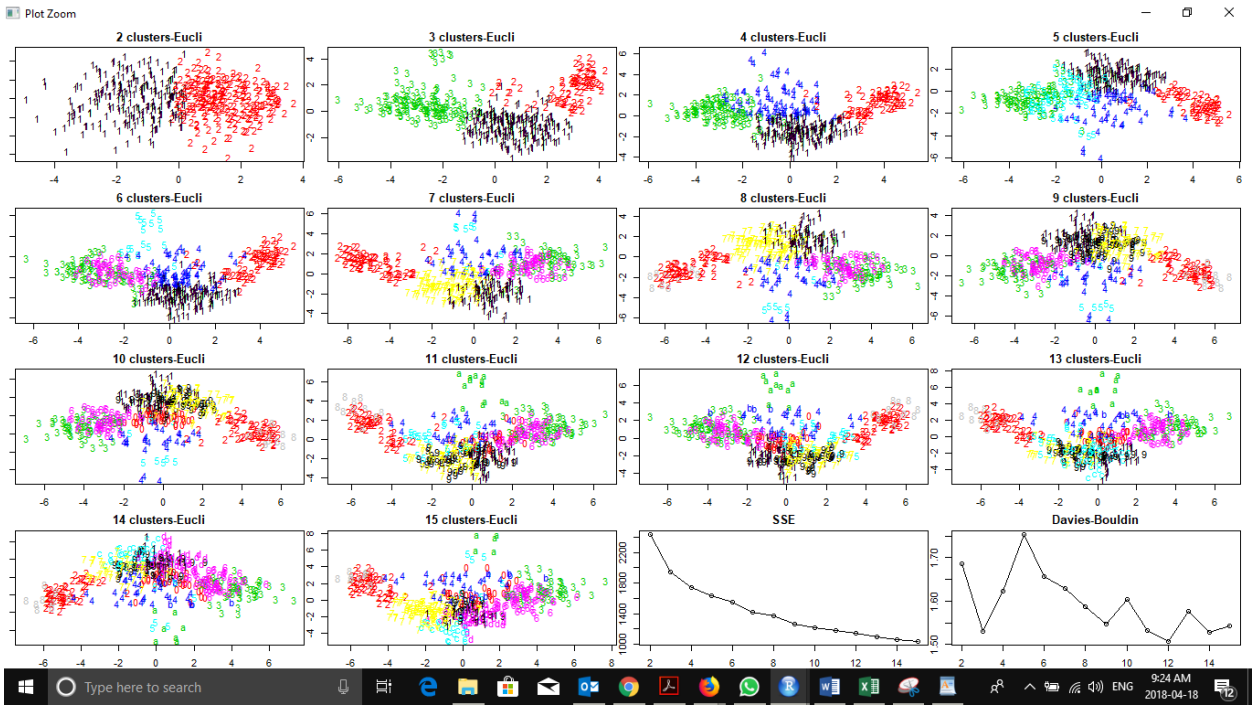
```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest[happiest$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 178
> nrow(happiest[happiest$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 20
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest[least_happiest$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 10
> nrow(least_happiest[least_happiest$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 139
>
```

4.2. Standard Dataset-Clustering

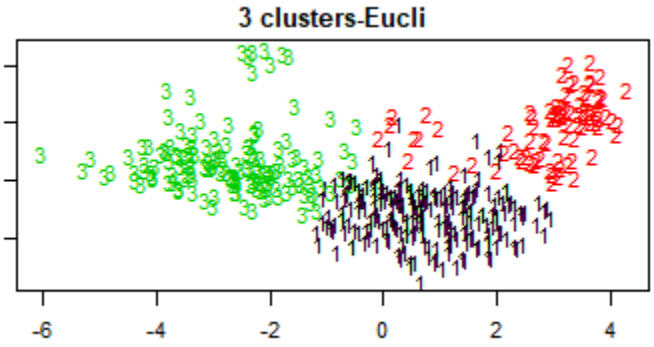
Cluster Analysis from range 2 to 15 for Standard Happiness Data Set.

In Cluster analysis, in this dataset we do not have any attributes for classification of data, therefore we have assigned numbers and colors based on the Clusters.

Optimal Value of Whole Standard Dataset from Davis Bouldin is 3.



- **Optimal Cluster Value is 3-**



- **Best Seed and Centroids for the Cluster-**

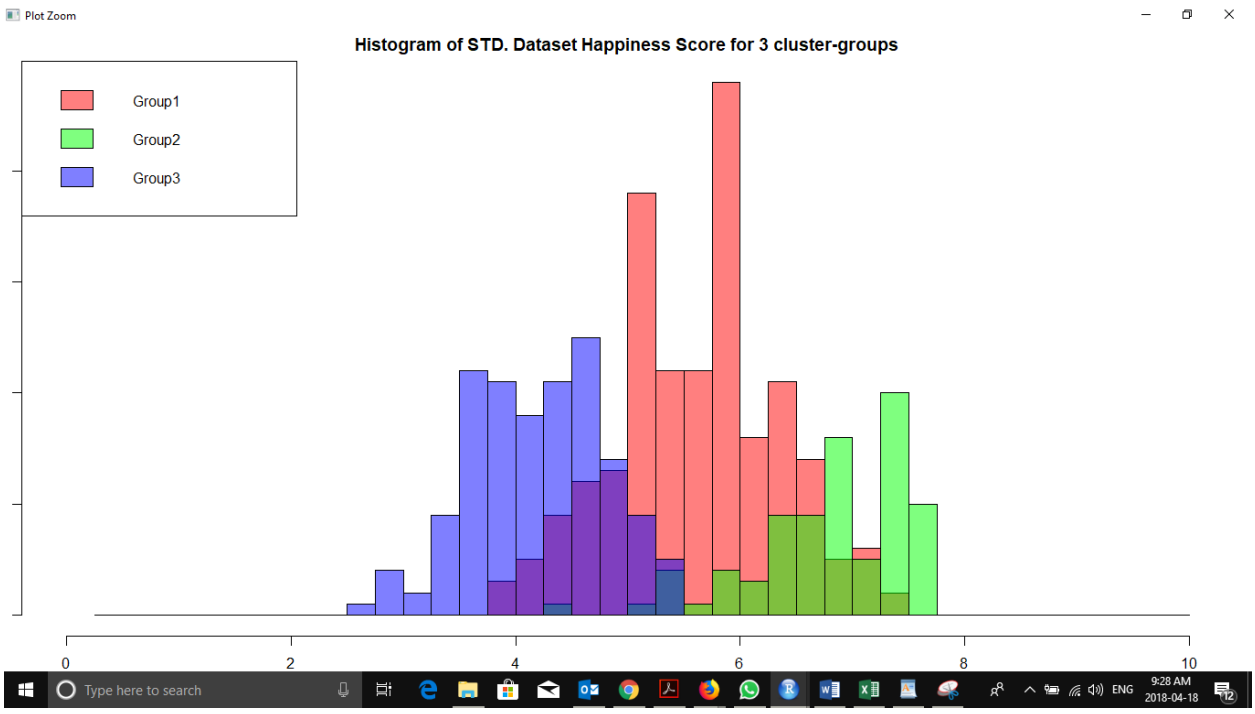
```
> km.std <- clustering_euclidean(happiness.std,happiness.data, 3)
[1] "Best Seed for Cluster Size  3 is  2"
[1] "Total Wrong in Cluster Size  3 is  0"
[1] "Centroids for Cluster Size  3 are :"
  Economy..GDP.per.Capita.     Family Health..Life.Expectancy.     Freedom
1                0.3415211  0.2489903                0.4063856 -0.1172141
2                1.1196415  0.8679609                0.9277679  1.1568463
3               -1.1492001 -0.8662416               -1.1451109 -0.4526868
  Trust..Government.Corruption. Generosity Dystopia.Residual
1                    -0.4403715 -0.4401195       0.096759426
2                     1.4948432  0.9418062       0.007615361
3                    -0.1334061  0.1701874      -0.155412580
K-means clustering with 3 clusters of sizes 236, 83, 151
```

- **Sum of Squares –**

```
Within cluster sum of squares by cluster:
[1] 896.9747 280.5620 766.8245
 (between_SS / total_SS =  40.8 %)
```

- **Cluster Analysis-**

• Distribution of High Score for Standard Dataset With optimal Value of Cluster Size from Davies Bouldin is 3-> There are overlap with respect to all cluster group.



- Happiest Cluster Group and Least Happiest Cluster Group- Country with Highest Happiness Score in Group G3 is Switzerland(7.587) and with Least Happiness Score in Cluster Group G2 is Tongo(2.839).

```
> #Cluster Size
> print(km.std$size)
[1] 236  83 151
>
> print(paste("Standard-Happiest Group is g", top, sep=""))
[1] "Standard-Happiest Group is g2"
>
> print(paste("Standard-Least Happiest Group is g", bottom, sep=""))
[1] "Standard-Least Happiest Group is g3"
>
> #Countries with highest and least Happiness score in clusters
> head(happiest.std[order(happiest.std$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
1   Switzerland           7.587
2       Iceland           7.561
316      Norway           7.537
3       Denmark           7.527
159     Denmark           7.526
4        Norway           7.522
>
> tail(least_happiest.std[order(least_happiest.std$Happiness.Score, decreasing=TRUE), ])
                    Country Happiness.Score
156                   Syria           3.006
157                 Burundi           2.905
315                 Burundi           2.905
469                 Burundi           2.905
158                    Togo           2.839
470 Central African Republic           2.693
>
```

- Countries above Median Value of Happiness Score in Least Happiness Cluster Group and Countries below Median values of Happiness Score of Happiest Cluster Group.

```
> print("Median Happiness Score for Whole Dataset-")
[1] "Median Happiness Score for Whole Dataset-"
> (median(happiness.data$Happiness.Score))
[1] 5.2825
>
> head(least_happiest.std[least_happiest.std$Happiness.Score > median(happiness.data$Happiness.Score), ]
)
      Country Happiness.Score
234   Somalia           5.440
235    Kosovo           5.401
237 Indonesia           5.314
> head(happiest.std[happiest.std$Happiness.Score < median(happiness.data$Happiness.Score), ])
      Country Happiness.Score
79     Bhutan           5.253
412    Bhutan           5.011
435 Sri Lanka           4.440
>
```

- Number of Countries in Happiest Group and Least Happiest Cluster Group above and below Median Value-

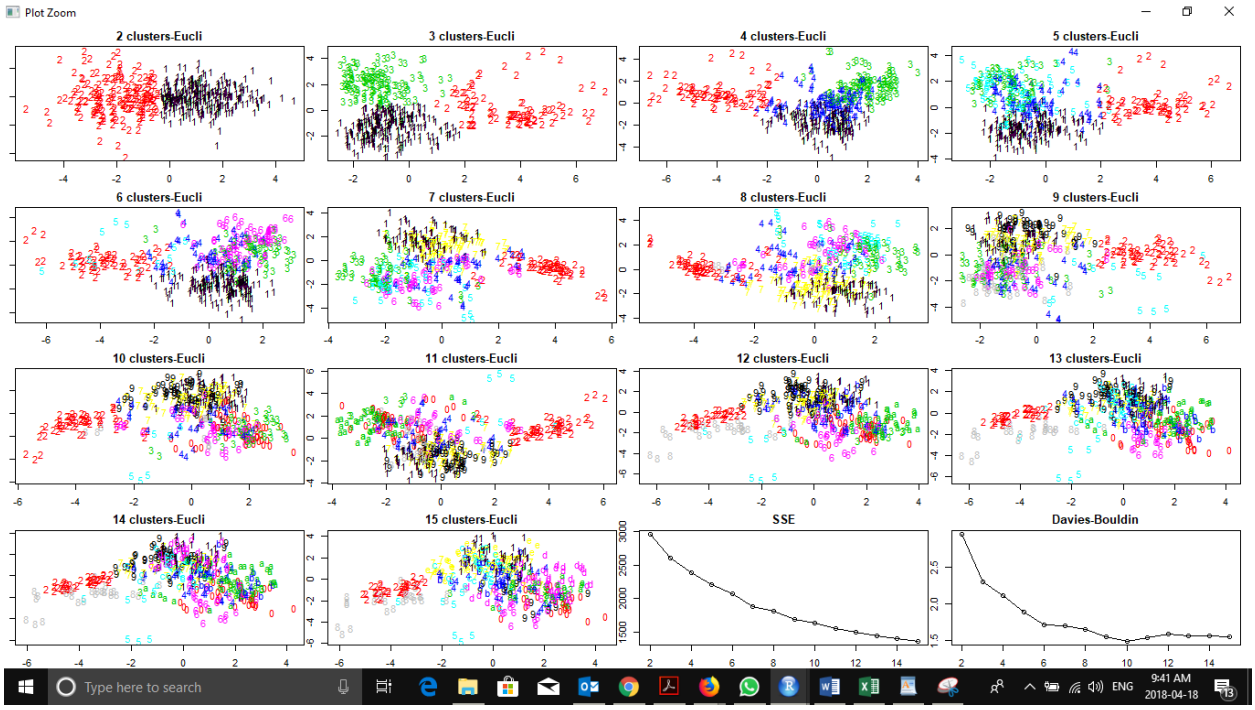
```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest.std[happiest.std$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 80
> nrow(happiest.std[happiest.std$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 3
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest.std[least_happiest.std$Happiness.Score > median(happiness.data$Happiness.Score), ]
)
[1] 3
> nrow(least_happiest.std[least_happiest.std$Happiness.Score < median(happiness.data$Happiness.Score), ]
)
[1] 148
>
```

4.3. Whitened Dataset-Clustering

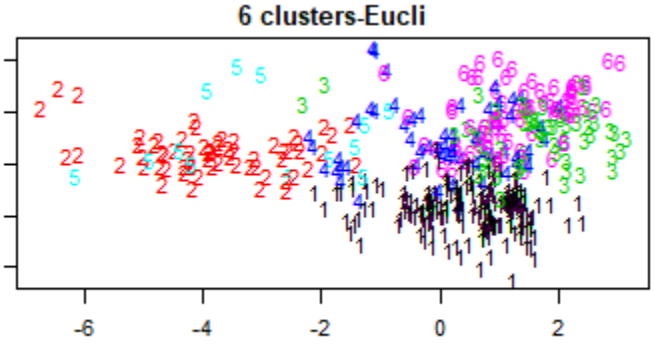
Cluster Analysis from range 2 to 15 for Whitened Happiness Data Set.

In Cluster analysis, in this dataset we do not have any attributes for classification of data, therefore we have assigned numbers and colors based on the Clusters.

Optimal Value of Whole Whitened Dataset from Davis Bouldin is 6.



- **Optimal Cluster Value is 6-**



- **Best Seed and Centroids for the Cluster-**

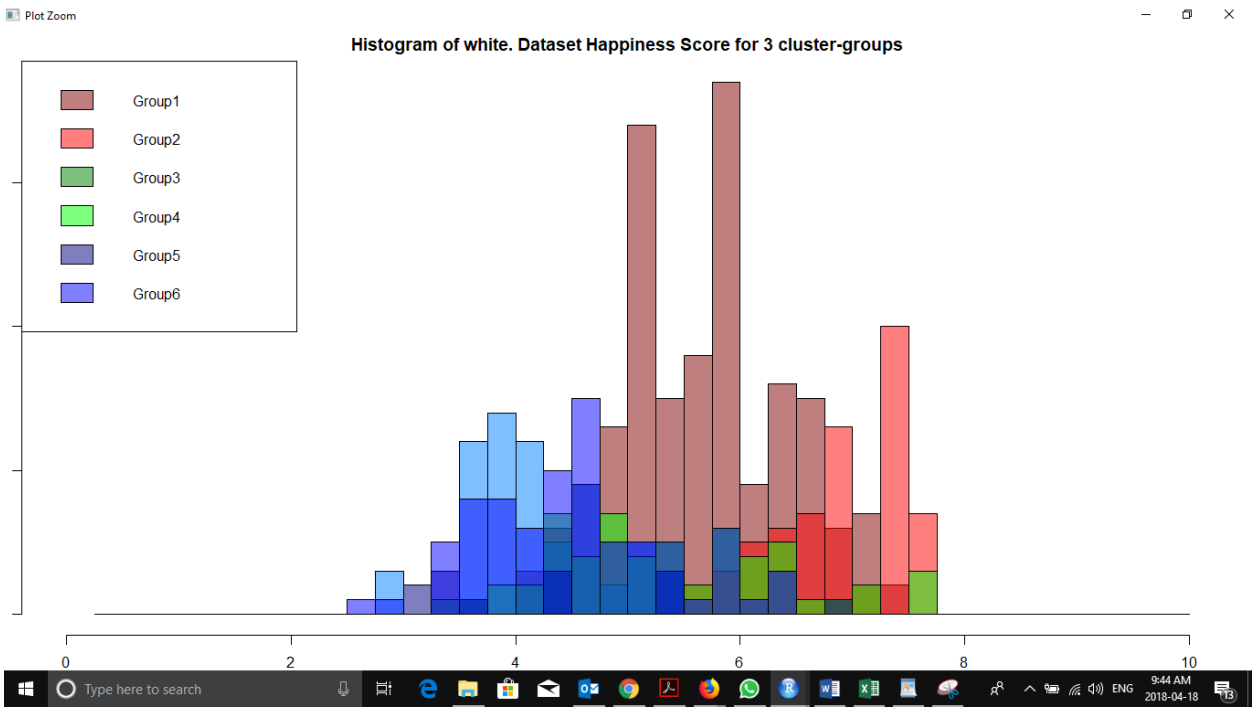
```
> km.white <- clustering_euclidean(happiness.white,happiness.data, 6)
[1] "Best Seed for Cluster Size  6 is  2"
[1] "Total Wrong in Cluster Size  6 is  0"
[1] "Centroids for Cluster Size  6 are :"
         [,1]       [,2]        [,3]       [,4]       [,5]        [,6]         [,7]
1  0.13299767  0.1793188  0.60773780 -0.2431635 -0.4227754 -0.52951202  0.214937113
2  1.08424555  0.3889339  0.09821661  0.4541148  1.5011301  0.46909949  0.181171605
3 -1.18641918  0.1164302 -0.67341411 -1.0465810  0.3942649  0.15013785 -0.009359432
4 -0.01511794  0.1850207  0.05795901  0.5180550 -0.8333960  1.47407425 -0.581043850
5 -0.11458152 -1.3284415  0.92809694 -0.1312500  2.0281762 -0.01386235 -1.939965988
6 -0.20346236 -0.7524459 -1.16814085  0.7111059 -0.2717008 -0.29356456  0.079165182
K-means clustering with 6 clusters of sizes 190, 65, 65, 57, 13, 80
```

- **Sum of Squares –**

```
Within cluster sum of squares by cluster:
[1] 761.0810 218.3659 319.3393 275.5933 110.7837 394.5346
 (between_SS / total_SS =  36.7 %)
```

- **Cluster Analysis-**

• Distribution of High Score for Raw DataSet With optimal Value of Cluster Size from Davies Bouldin is 6-> There are overlap with respect to all cluster group.



- Happiest Cluster Group and Least Happiest Cluster Group- Country with Highest Happiness Score in Group G2 is Switzerland(7.587) and with Least Happiness Score in Cluster Group G3 is Burundi(2.905).

```
> #Cluster Size
> print(km.white$size)
[1] 190  65  65  57  13  80
>
> print(paste("Whitened-Happiest Group is g", top, sep=""))
[1] "Whitened-Happiest Group is g2"
>
> print(paste("Whitened-Least Happiest Group is g", bottom, sep=""))
[1] "Whitened-Least Happiest Group is g3"
>
> #Countries with highest and least Happiness score in clusters
> head(happiest.white[order(happiest.white$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
1   Switzerland           7.587
316      Norway           7.537
3       Denmark           7.527
159     Denmark           7.526
4        Norway           7.522
317     Denmark           7.522
>
> tail(least_happiest.white[order(least_happiest.white$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
153 Afghanistan           3.575
463     Liberia           3.533
464      Guinea           3.507
157     Burundi           2.905
315     Burundi           2.905
469     Burundi           2.905
>
```

- Countries above Median Value of Happiness Score in Least Happiness Cluster Group and Countries below Median values of Happiness Score of Happiest Cluster Group.

```
> print("Median Happiness Score for Whole Dataset-")
[1] "Median Happiness Score for Whole Dataset-"
> (median(happiness.data$Happiness.Score))
[1] 5.2825
>
> head(least_happiest.white[least_happiest.white$Happiness.Score > median(happiness.data$Happiness.Score
), ])
    Country Happiness.Score
235  Kosovo           5.401
> head(happiest.white[happiest.white$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] Country         Happiness.Score
<0 rows> (or 0-length row.names)
```

- Number of Countries in Happiest Group and Least Happiest Cluster Group above and below Median Value-

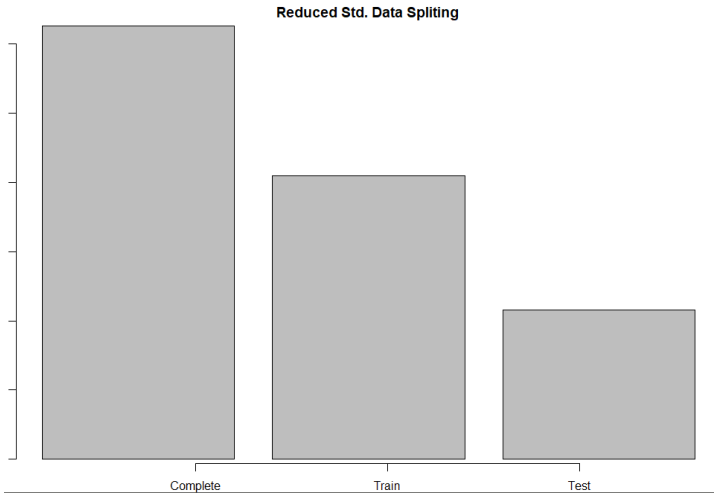
```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest.white[happiest.white$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 65
> nrow(happiest.white[happiest.white$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 0
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest.white[least_happiest.white$Happiness.Score > median(happiness.data$Happiness.Score
), ])
[1] 1
> nrow(least_happiest.white[least_happiest.white$Happiness.Score < median(happiness.data$Happiness.Score
), ])
[1] 64
>
```

4.4. Data Reduction -Using Standard Dataset Clustering-

- After analysis of Raw, Standard and Whitened dataset we found that Standard dataset is much better then Whitened and Raw.
- Now we reduce our dataset.
- We take samples of size 2/3 from Raw Dataset based on Standard Dataset cluster distribution. Using this we have a reduced dataset based on and standard dataset clustering with proper distribution of reduced dataset.
- Now we divide train and test dataset based on Year. Data belong to year 2015 and 2016 in training Set and data belong to year 2017 in test set.
- **We will use this reduced train and test dataset for Supervised Learning**

Our new Dataset has 313 rows out of 455.

```
> nrow(happiness.reduced.data[happiness.reduced.data$Year!=2017,])
[1] 205
> nrow(happiness.reduced.data[happiness.reduced.data$Year==2017,])
[1] 108
> nrow(happiness.std.reduced.data[happiness.std.reduced.data$Year!=2017,])
[1] 205
> nrow(happiness.std.reduced.data[happiness.std.reduced.data$Year==2017,])
[1] 108
```



5. UNSUPERVISED LEARNING

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data.

Analysis-

1. We have done the Cluster on 7 factors participating in Happiness score i.e. GDP per Capita, Family, Freedom, Life Expectancy, Trust (Government Corruption), Generosity and Dystopia Residual.

2. We have done the unsupervised learning for Complete and train Raw, PCA and ICA dataset, and applied the corresponding optimal value of the cluster on the test dataset.

3. We got optimal value of cluster for Raw and PCA dataset to be 3 and for ICA dataset we got 4.

4. In the raw train dataset clustering, we got number of data in the happiest cluster below median value is 6 and number of countries in the least happy group above median value is 3.
   When we apply the optimal cluster value 3 on test dataset, we got number of data in the happiest cluster below median value is 16 and number of countries in the least happy group above median value is 0.

5. In the PCA complete dataset clustering, we got number of data in the happiest cluster below median value is 10 and number of countries in the least happy group above median value is 3. Then we do clustering on PCA training dataset and found optimal value of cluster is 3 and got number of data in the happiest cluster below median value is 8 and number of countries in the least happy group above median value is 2.
   When we apply the optimal cluster value 3 on PCA test dataset, we got number of data in the happiest cluster below median value is 2 and number of countries in the least happy group above median value is 0.

6. In the ICA complete dataset clustering, we got number of data in the happiest cluster below median value is 32 and number of countries in the least happy group above median value is 13. Then we do clustering on ICA training dataset and found optimal value of cluster is 3 and got number of data in the happiest cluster below median value is 6 and number of countries in the least happy group above median value is 1.
   When we apply the optimal cluster value 3 on ICA test dataset, we got number of data in the happiest cluster below median value is 24 and number of countries in the least happy group above median value is 6.

7. With the analysis of the median values we found that PCA dataset with 3 components gives better result on modelling rather than Raw Dataset and ICA dataset.

8. Value of SSE for Raw train and test dataset with 3 clusters is approx. 48%, SSE for PCA complete, train and test dataset with 3 clusters is approx. 56%, and SSE for ICA complete, train and test dataset with 4 clusters is approx. 44%.

9. Data overlapping in histogram is higher is Raw complete, train, test dataset with 3 clusters, lower in ICA complete, train, test dataset with 4 clusters and lowest in PCA complete, train, test dataset.

10. After Complete analysis we found unsupervised learning using PCA dataset will gives the best result.

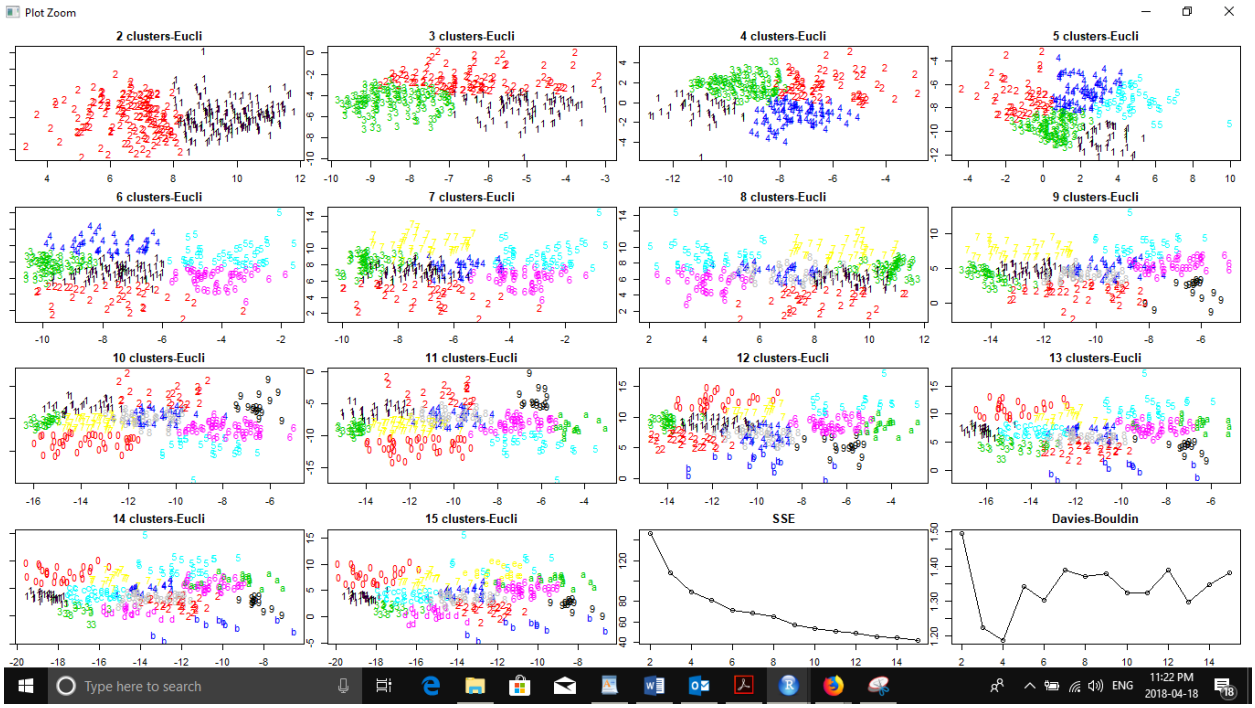
5.1. Raw Dataset-Clustering

5.1.1. Raw Train Dataset- Clustering

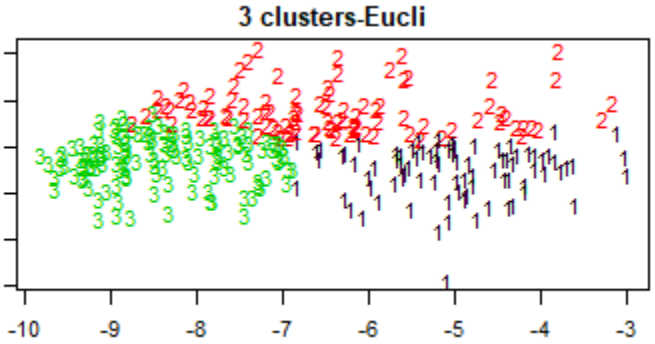
Cluster Analysis from range 2 to 15 for Train Raw Dataset

In Cluster analysis, in this dataset we do not have any attributes for classification of data, therefore we have assigned numbers and colors based on the Clusters.

Optimal Value of Train Dataset from Davis Bouldin is 3.



- **Optimal Cluster Value is 3-**



- **Best Seed and Centroids for the Cluster-**

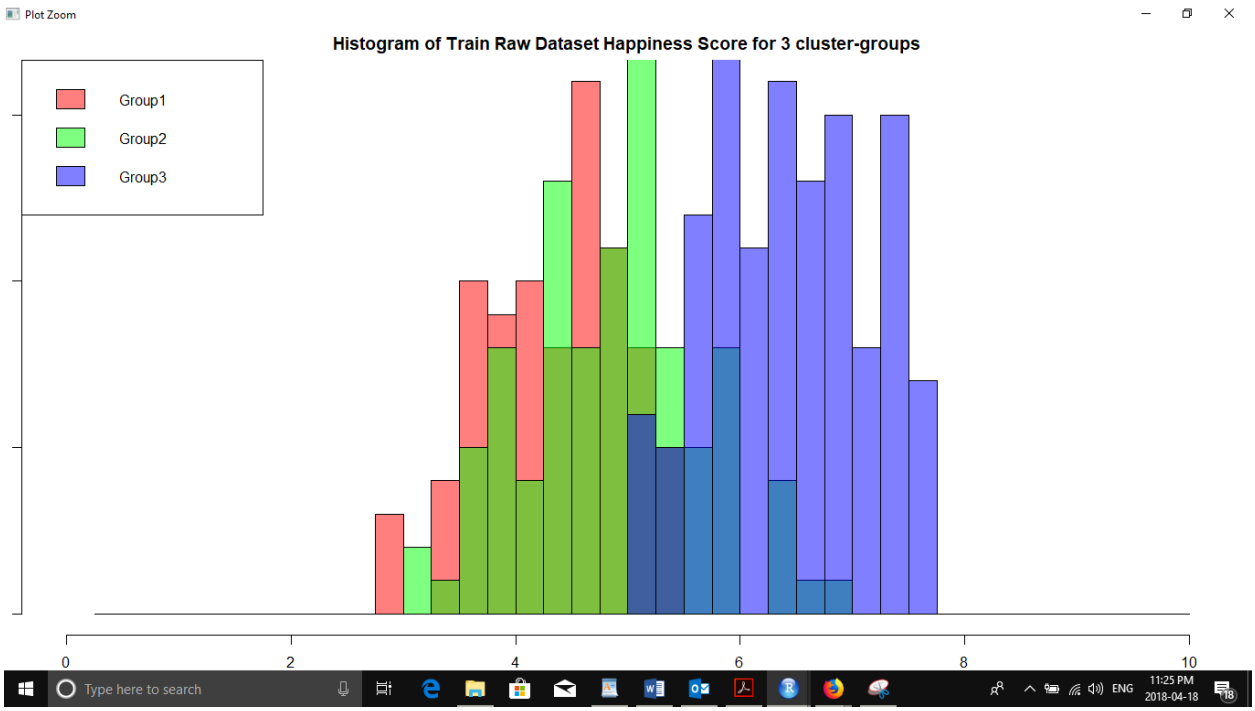
```
> km.train <- clustering_euclidean(happiness.train[, 5:11],happiness.train,3)
[1] "Best Seed for Cluster Size  3 is  2"
[1] "Total Wrong in Cluster Size  3 is  0"
[1] "Centroids for Cluster Size  3 are :"
  Economy..GDP.per.Capita.    Family Health..Life.Expectancy.    Freedom
1                0.4087787 0.6299252                0.3131764 0.3193463
2                0.9427228 0.9013507                0.6374669 0.3956524
3                1.1748890 1.0499287                0.7379520 0.4530332
  Trust..Government.Corruption. Generosity Dystopia.Residual
1                     0.1132198  0.2358189          2.313211
2                     0.1280064  0.2461880          1.612996
3                     0.1664334  0.2381002          2.575057
K-means clustering with 3 clusters of sizes 84, 96, 135
```

- **Sum of Squares –**

```
Within cluster sum of squares by cluster:
[1] 28.38773 39.37860 40.29551
 (between_SS / total_SS =  48.9 %)
```

- **Cluster Analysis-**

• Distribution of High Score for Train Dataset with optimal Value of Cluster Size from Davies Bouldin is 3- > There are less overlap with respect to all cluster group.



- Happiest Cluster Group and Least Happiest Cluster Group- Country with Highest Happiness Score in Group G3 is Switzerland(7.587) and with Least Happiness Score in Cluster Group G1 is Tongo(2.839).

```
> #Cluster Size
> print(km.train$size)
[1]  84  96 135
>
> print(paste("RAW-Train Happiest Group is g", top, sep=""))
[1] "RAW-Train Happiest Group is g3"
>
> print(paste("RAW-Train-Least Happiest Group is g", bottom, sep=""))
[1] "RAW-Train-Least Happiest Group is g1"
>
>
> head(happiest.train[order(happiest.train$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
1   Switzerland           7.587
2       Iceland           7.561
3       Denmark           7.527
159     Denmark           7.526
4        Norway           7.522
160 Switzerland           7.509
>
> tail(least_happiest.train[order(least_happiest.train$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
312 Afghanistan           3.360
155       Benin           3.340
313        Togo           3.303
157     Burundi           2.905
315     Burundi           2.905
158        Togo           2.839
```

- Countries above Median Value of Happiness Score in Least Happiness Cluster Group and Countries below Median values of Happiness Score of Happiest Cluster Group.

```
> print("Median Happiness Score for training Dataset-")
[1] "Median Happiness Score for training Dataset-"
> (median(happiness.train$Happiness.Score))
[1] 5.286
>
> head(least_happiest.train[least_happiest.train$Happiness.Score > median(happiness.train$Happine
ss.Score), ])
    Country Happiness.Score
75  Vietnam            5.36
234 Somalia            5.44
> head(happiest.train[happiest.train$Happiness.Score < median(happiness.train$Happiness.Score), ]
)
                  Country Happiness.Score
241                 China           5.245
244                Serbia           5.177
245 Bosnia and Herzegovina           5.163
246            Montenegro           5.161
256               Tunisia           5.045
257                Greece           5.033
```

- Number of Countries in Happiest Group and Least Happiest Cluster Group above and below Median Value-

```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest.train[happiest.train$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 129
> nrow(happiest.train[happiest.train$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 6
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest.train[least_happiest.train$Happiness.Score > median(happiness.data$Happines
s.Score), ])
[1] 3
> nrow(least_happiest.train[least_happiest.train$Happiness.Score < median(happiness.data$Happines
s.Score), ])
[1] 81
```

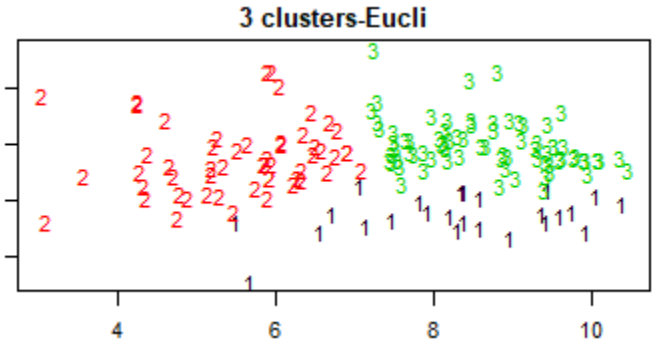
5.1.2. Raw Test Dataset- Clustering

Cluster Analysis for Test Raw Happiness Data Set with Optimal Value 3.

In Cluster analysis, in this dataset we do not have any attributes for classification of data, therefore we have assigned numbers and colors based on the Clusters.

Optimal Value of Test Raw Dataset from Davis Bouldin is 3.

- **Optimal Cluster Value is 3-**



- **Best Seed and Centroids for the Cluster-**

```
> km.test <- clustering_euclidean(happiness.test[, 5:11],happiness.test,3)
[1] "Best Seed for Cluster Size  3 is  2"
[1] "Total Wrong in Cluster Size  3 is  0"
[1] "Centroids for Cluster Size  3 are :"
  Economy..GDP.per.Capita.    Family Health..Life.Expectancy.   Freedom
1                0.8813445 1.1704651                0.5209162 0.4316554
2                0.5858999 0.9462608                0.3527142 0.3378009
3                1.3020936 1.3673611                0.7024783 0.4516999
  Trust..Government.Corruption. Generosity Dystopia.Residual
1                    0.09395736  0.2057365          2.588966
2                    0.09897280  0.2559743          1.566554
3                    0.14987064  0.2539594          1.808800
K-means clustering with 3 clusters of sizes 25, 54, 76
```

- **Sum of Squares –**

```
Within cluster sum of squares by cluster:
[1]  7.284975 22.733251 20.260476
 (between_SS / total_SS =  47.1 %)

Available components:
```

- **Cluster Analysis-**

• Distribution of High Score for Test Raw Dataset With optimal Value of Cluster Size from Davies Bouldin is 3-> There are less overlap with respect to all cluster group.

- Happiest Cluster Group and Least Happiest Cluster Group- Country with Highest Happiness Score in Group G3 is Norway(7.537) and with Least Happiness Score in Cluster Group G2 is Central African Republic(2.639).

```
> print(km.test$size)
[1] 25 54 76
>
> print(paste("RAW-Test Happiest Group is g", top, sep=""))
[1] "RAW-Test Happiest Group is g3"
>
> print(paste("RAW-Test Least Happiest Group is g", bottom, sep=""))
[1] "RAW-Test Least Happiest Group is g2"
>
> head(happiest.test[order(happiest.test$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
316     Norway           7.537
317    Denmark           7.522
318    Iceland           7.504
319 Switzerland           7.494
320    Finland           7.469
321 Netherlands           7.377
>
> tail(least_happiest.test[order(least_happiest.test$Happiness.Score, decreasing=TRUE), ])
                  Country Happiness.Score
465                  Togo           3.495
466                Rwanda           3.471
467                 Syria           3.462
468              Tanzania           3.349
469               Burundi           2.905
470 Central African Republic           2.693
>
```

- Countries above Median Value of Happiness Score in Least Happiness Cluster Group and Countries below Median values of Happiness Score of Happiest Cluster Group.

```
> print("Median Happiness Score for testing Dataset-")
[1] "Median Happiness Score for testing Dataset-"
> (median(happiness.test$Happiness.Score))
[1] 5.279
>
> head(least_happiest.test[least_happiest.test$Happiness.Score > median(happiness.test$Happiness.
Score), ])
[1] Country         Happiness.Score
<0 rows> (or 0-length row.names)
> head(happiest.test[happiest.test$Happiness.Score < median(happiness.test$Happiness.Score), ])
               Country Happiness.Score
394              China           5.273
396          Indonesia           5.262
397          Venezuela           5.250
398         Montenegro           5.237
400         Azerbaijan           5.234
401 Dominican Republic           5.230
>
```

- Number of Countries in Happiest Group and Least Happiest Cluster Group above and below Median Value-

```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest.test[happiest.test$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 60
> nrow(happiest.test[happiest.test$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 16
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest.test[least_happiest.test$Happiness.Score > median(happiness.data$Happiness.
Score), ])
[1] 0
> nrow(least_happiest.test[least_happiest.test$Happiness.Score < median(happiness.data$Happiness.
Score), ])
[1] 54
```

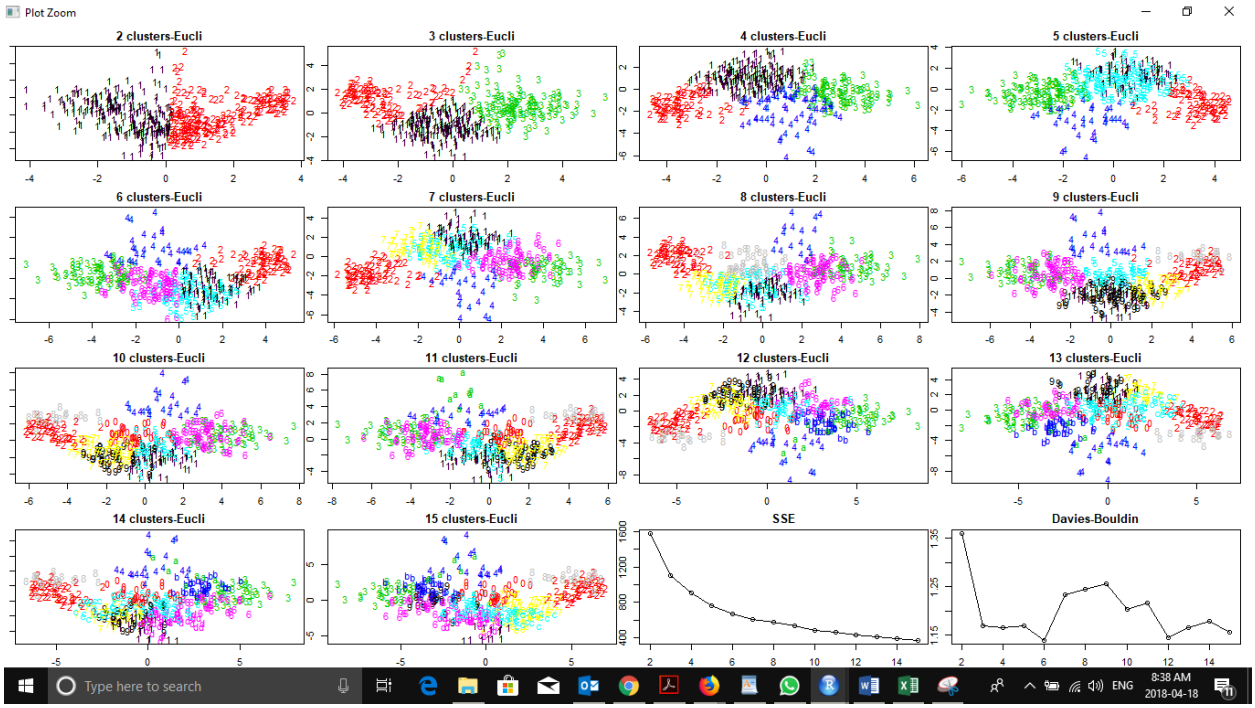
5.2. Standard Dataset After PCA-Clustering

5.2.1. Standard Complete Dataset After PCA- Clustering

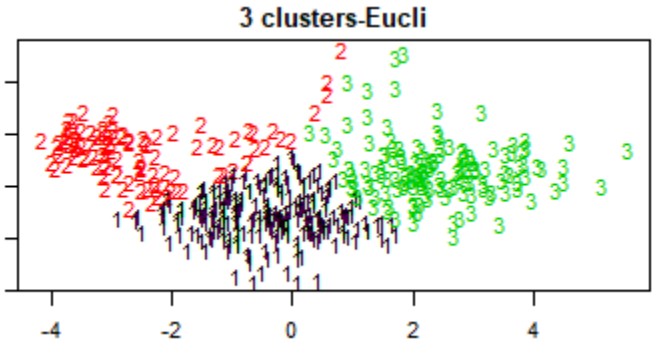
Cluster Analysis from range 2 to 15 for PCA Happiness Data Set.

In Cluster analysis, in this dataset we do not have any attributes for classification of data, therefore we have assigned numbers and colors based on the Clusters.

Optimal Value of Whole PCA Dataset from Davis Bouldin is 3.



- **Optimal Cluster Value is 3-**



- **Best Seed and Centroids for the Cluster-**

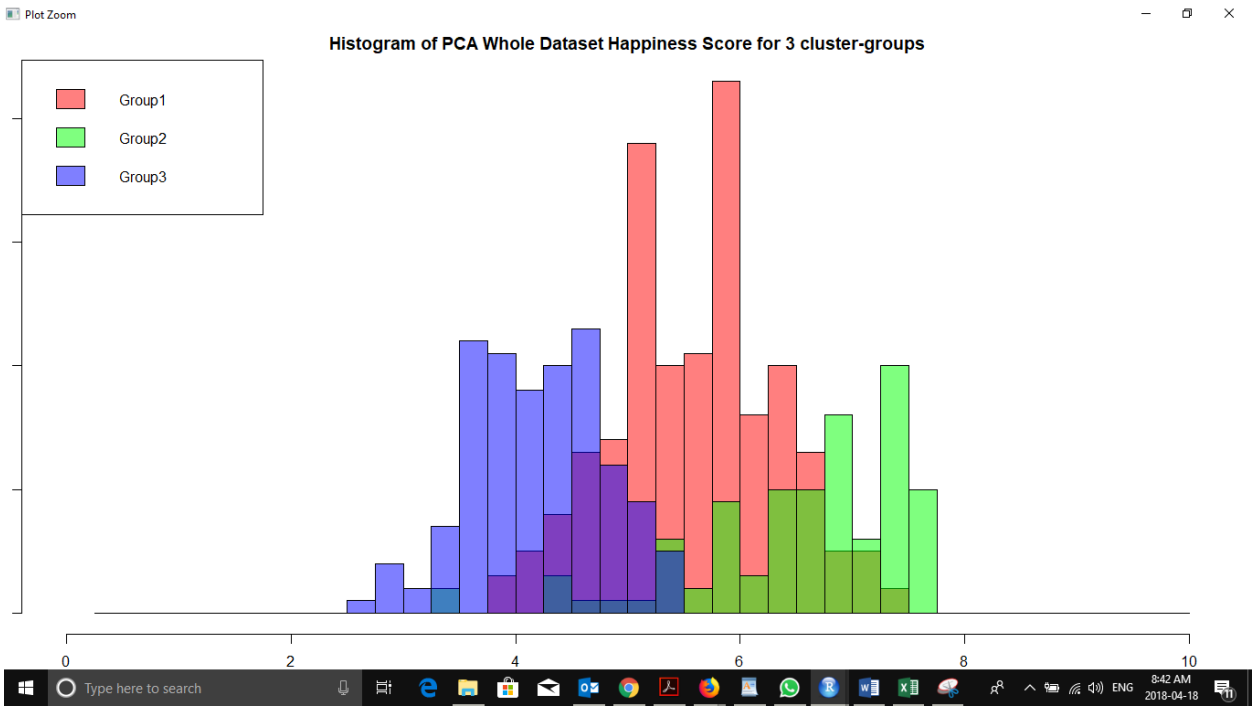
```
[1] "Best Seed for Cluster Size  3 is  2"
[1] "Total Wrong in Cluster Size  3 is  0"
[1] "Centroids for Cluster Size  3 are :"
         PC1        PC2         PC3
1 -0.1293305  0.8773469 -0.06370035
2 -2.2784090 -0.8443284  0.05364194
3  1.7852055 -0.7906081  0.06272281
K-means clustering with 3 clusters of sizes 226, 100, 144
```

- **Sum of Squares –**

```
within cluster sum of squares by cluster:
[1] 496.4332 254.0336 348.9140
 (between_SS / total_SS =  54.5 %)
```

- **Cluster Analysis-**

• Distribution of High Score for whole DataSet after PCA With optimal Value of Cluster Size from Davies Bouldin is 3-> There are less overlap with respect to all cluster group.



- Happiest Cluster Group and Least Happiest Cluster Group- Country with Highest Happiness Score in Group G2 is Switzerland(7.587) and with Least Happiness Score in Cluster Group G3 is Central African Republic(2.693).

```
> print(paste("PCA-Happiest Group is g", top, sep=""))
[1] "PCA-Happiest Group is g2"
>
> print(paste("PCA-Least Happiest Group is g", bottom, sep=""))
[1] "PCA-Least Happiest Group is g3"
>
> head(happiest.pc[order(happiest.pc$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
1   Switzerland           7.587
2       Iceland           7.561
316      Norway           7.537
3       Denmark           7.527
159     Denmark           7.526
4        Norway           7.522
>
> tail(least_happiest.pc[order(least_happiest.pc$Happiness.Score, decreasing=TRUE), ])
                     Country Happiness.Score
156                    Syria           3.006
157                  Burundi           2.905
315                  Burundi           2.905
469                  Burundi           2.905
158                     Togo           2.839
470 Central African Republic           2.693
>
```

- Countries above Median Value of Happiness Score in Least Happiness Cluster Group and Countries below Median values of Happiness Score of Happiest Cluster Group.

```
> print("Median Happiness Score for Whole Dataset-")
[1] "Median Happiness Score for Whole Dataset-"
> (median(happiness.data$Happiness.Score))
[1] 5.2825
>
> head(least_happiest.pc[least_happiest.pc$Happiness.Score > median(happiness.data$Happiness.Scor
e), ])
      Country Happiness.Score
234   Somalia           5.440
235    Kosovo           5.401
237 Indonesia           5.314
> head(happiest.pc[happiest.pc$Happiness.Score < median(happiness.data$Happiness.Score), ])
      Country Happiness.Score
79     Bhutan           5.253
99       Laos           4.876
132 Sri Lanka           4.271
154    Rwanda           3.465
275 Sri Lanka           4.415
396 Indonesia           5.262
```

- Number of Countries in Happiest Group and Least Happiest Cluster Group above and below Median Value-

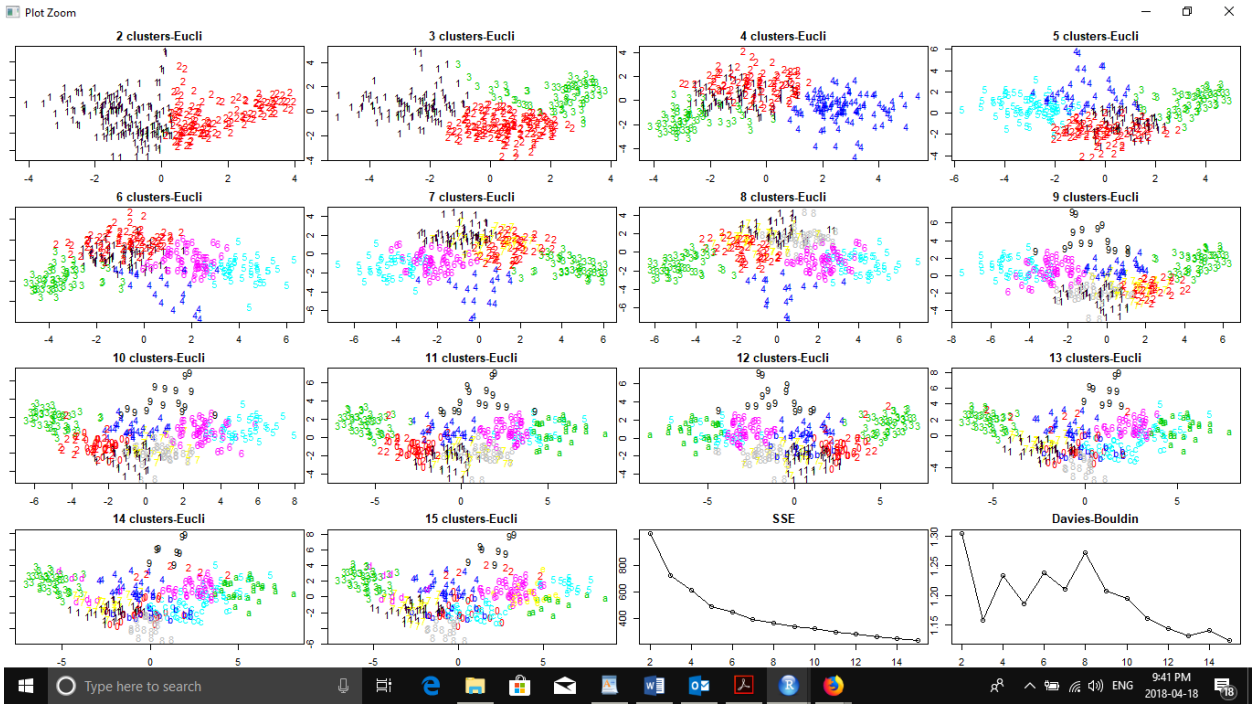
```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest.pc[happiest.pc$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 90
> nrow(happiest.pc[happiest.pc$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 10
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest.pc[least_happiest.pc$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 3
> nrow(least_happiest.pc[least_happiest.pc$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 141
>
```

5.2.2. Standard Train Dataset After PCA- Clustering

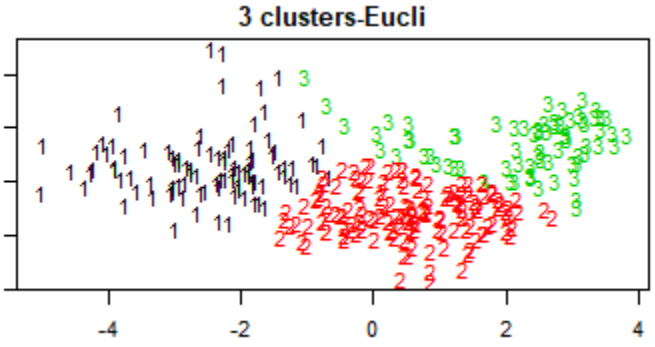
Cluster Analysis from range 2 to 15 for Train PCA Happiness Data Set.

In Cluster analysis, in this dataset we do not have any attributes for classification of data, therefore we have assigned numbers and colors based on the Clusters.

Optimal Value of Train PCA Dataset from Davis Bouldin is 3.



- **Optimal Cluster Value is 3-**



- **Best Seed and Centroids for the Cluster-**

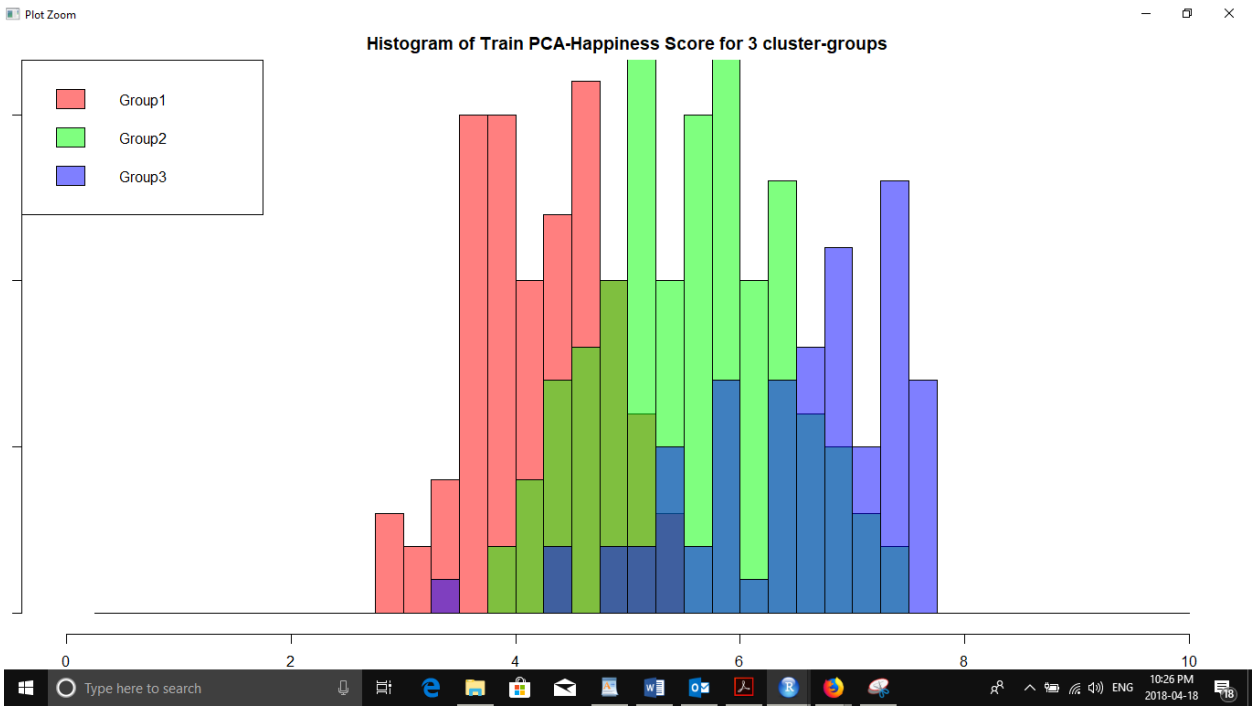
```
> km.train.pc <- clustering_euclidean(happiness.train.pc,happiness.train, 3)
[1] "Best Seed for Cluster Size  3 is  2"
[1] "Total Wrong in Cluster Size  3 is  0"
[1] "Centroids for Cluster Size  3 are :"
         PC1        PC2         PC3
1  1.77615644 -0.8475555  0.17958016
2 -0.06033927  0.9260440 -0.13085774
3 -2.21508884 -0.7374945  0.02555527
K-means clustering with 3 clusters of sizes 96, 146, 73
```

- **Sum of Squares –**

```
within cluster sum of squares by cluster:
[1] 244.8619 310.8659 170.0896
 (between_SS / total_SS =  55.4 %)
```

- **Cluster Analysis-**

• Distribution of High Score for Train DataSet after PCA With optimal Value of Cluster Size from Davies Bouldin is 3-> There are less overlap with respect to all cluster group.



- Happiest Cluster Group and Least Happiest Cluster Group- Country with Highest Happiness Score in Group G3 is Switzerland(7.587) and with Least Happiness Score in Cluster Group G1 is Tongo(2.839).

```
> #Cluster Size
> print(km.train.pc$size)
[1]  96 146  73
>
> print(paste("Train-PCA-Happiest Group is g", top, sep=""))
[1] "Train-PCA-Happiest Group is g3"
>
> print(paste("Train-PCA-Least Happiest Group is g", bottom, sep=""))
[1] "Train-PCA-Least Happiest Group is g1"
>
>
> head(happiest.train.pc[order(happiest.train.pc$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
1   Switzerland           7.587
2       Iceland           7.561
3       Denmark           7.527
159     Denmark           7.526
4        Norway           7.522
160 Switzerland           7.509
>
> tail(least_happiest.train.pc[order(least_happiest.train.pc$Happiness.Score, decreasing=TRUE), ]
)
    Country Happiness.Score
313    Togo           3.303
314   Syria           3.069
156   Syria           3.006
157 Burundi           2.905
315 Burundi           2.905
158    Togo           2.839
```

- Countries above Median Value of Happiness Score in Least Happiness Cluster Group and Countries below Median values of Happiness Score of Happiest Cluster Group.

```
> print("Median Happiness Score for testing Dataset-")
[1] "Median Happiness Score for testing Dataset-"
> (median(happiness.train$Happiness.Score))
[1] 5.286
>
> head(least_happiest.train.pc[least_happiest.train.pc$Happiness.Score > median(happiness.train$H
appiness.Score), ])
      Country Happiness.Score
234   Somalia           5.440
237 Indonesia           5.314
> head(happiest.train.pc[happiest.train.pc$Happiness.Score < median(happiness.train$Happiness.Sco
re), ])
               Country Happiness.Score
79              Bhutan           5.253
90         Philippines           5.073
98  Dominican Republic           4.885
99                Laos           4.876
132          Sri Lanka           4.271
154             Rwanda           3.465
```

- Number of Countries in Happiest Group and Least Happiest Cluster Group above and below Median Value-

```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest.train.pc[happiest.train.pc$Happiness.Score > median(happiness.data$Happiness.Scor
e), ])
[1] 65
> nrow(happiest.train.pc[happiest.train.pc$Happiness.Score < median(happiness.data$Happiness.Scor
e), ])
[1] 8
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest.train.pc[least_happiest.train.pc$Happiness.Score > median(happiness.data$Ha
ppiness.Score), ])
[1] 2
> nrow(least_happiest.train.pc[least_happiest.train.pc$Happiness.Score < median(happiness.data$Ha
ppiness.Score), ])
[1] 94
```

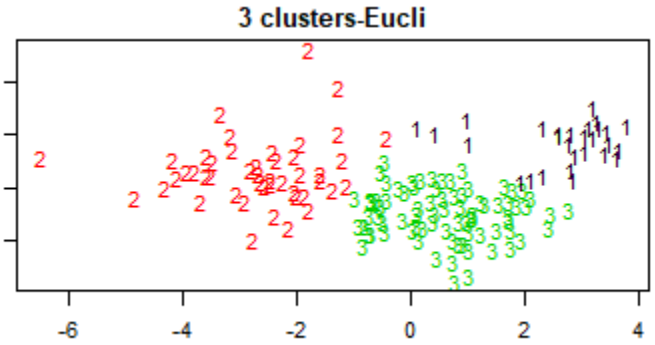
5.2.3. Standard Test Dataset After PCA- Clustering

Cluster Analysis for Test PCA Happiness Data Set with Optimal Value 3.

In Cluster analysis, in this dataset we do not have any attributes for classification of data, therefore we have assigned numbers and colors based on the Clusters.

Optimal Value of Train PCA Dataset from Davis Bouldin is 3 which we use on Test dataset.

- **Optimal Cluster Value is 3-**



- **Best Seed and Centroids for the Cluster-**

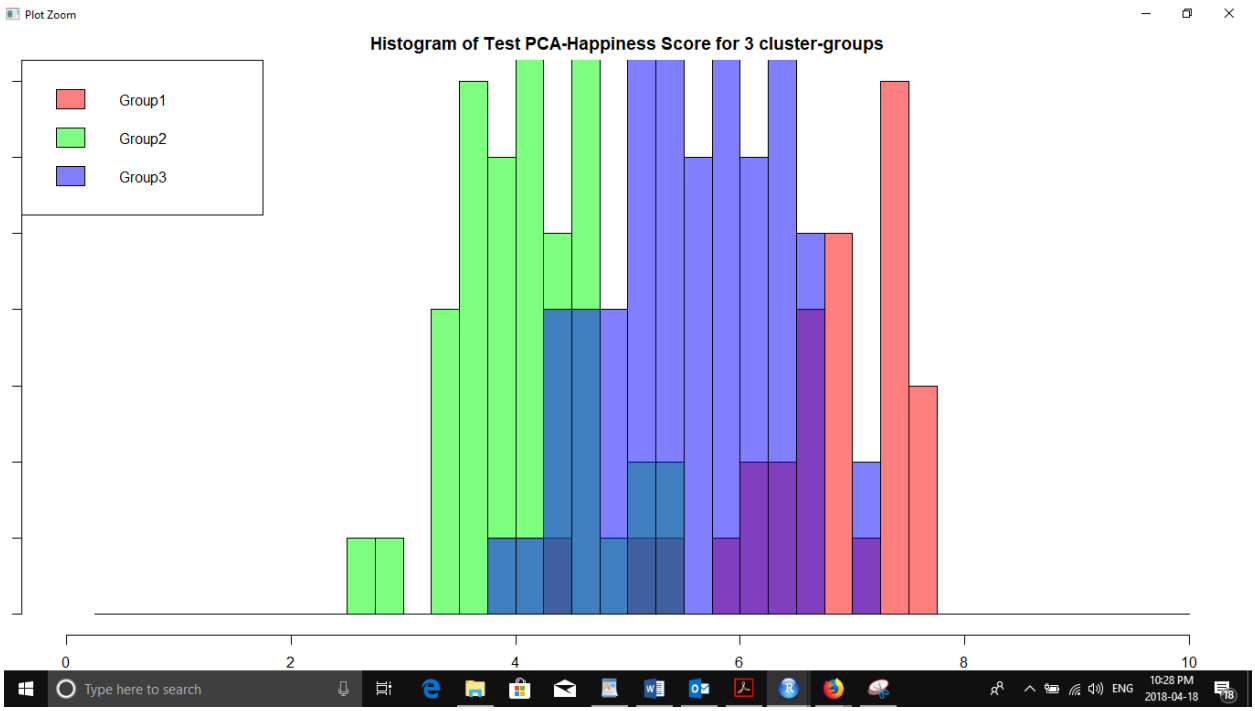
```
> km.test.pc <- clustering_euclidean(happiness.test.pc,happiness.test,3)
[1] "Best Seed for Cluster Size  3 is  2"
[1] "Total Wrong in Cluster Size  3 is  0"
[1] "Centroids for Cluster Size  3 are :"
         PC1         PC2         PC3
1 -2.5079708 -1.0179367  0.04413783
2  1.9244910 -0.8106594  0.13229307
3 -0.2259679  0.8122538 -0.09038692
K-means clustering with 3 clusters of sizes 28, 46, 81
```

- **Sum of Squares –**

```
Within cluster sum of squares by cluster:
[1]  55.15106 131.05551 168.47980
 (between_SS / total_SS =  56.7 %)
```

- **Cluster Analysis-**

• Distribution of High Score for Test DataSet after PCA With optimal Value of Cluster Size from Davies Bouldin is 3-> There are less overlap with respect to all cluster group.



- Happiest Cluster Group and Least Happiest Cluster Group- Country with Highest Happiness Score in Group G1 is Norway(7.537) and with Least Happiness Score in Cluster Group G2 is Central African Republic(2.639).

```
> print(km.test.pc$size)
[1] 28 46 81
>
> print(paste("Test-PCA-Happiest Group is g", top, sep=""))
[1] "Test-PCA-Happiest Group is g1"
>
> print(paste("Test-PCA-Least Happiest Group is g", bottom, sep=""))
[1] "Test-PCA-Least Happiest Group is g2"
>
> head(happiest.test.pc[order(happiest.test.pc$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
316     Norway           7.537
317    Denmark           7.522
318    Iceland           7.504
319 Switzerland           7.494
320    Finland           7.469
321 Netherlands           7.377
>
> tail(least_happiest.test.pc[order(least_happiest.test.pc$Happiness.Score, decreasing=TRUE), ])
                  Country Happiness.Score
465                  Togo           3.495
466                Rwanda           3.471
467                 Syria           3.462
468              Tanzania           3.349
469               Burundi           2.905
470 Central African Republic           2.693
>
```

- Countries above Median Value of Happiness Score in Least Happiness Cluster Group and Countries below Median values of Happiness Score of Happiest Cluster Group.

```
> #Identify who in the top group has lower happiness than the
> #median happiness for the entire set of countries
> print("Median Happiness Score for testing Dataset-")
[1] "Median Happiness Score for testing Dataset-"
> (median(happiness.test$Happiness.Score))
[1] 5.279
>
> head(least_happiest.test.pc[least_happiest.test.pc$Happiness.Score > median(happiness.test$Happ
iness.Score), ])
[1] Country         Happiness.Score
<0 rows> (or 0-length row.names)
> head(happiest.test.pc[happiest.test.pc$Happiness.Score < median(happiness.test$Happiness.Score)
, ])
      Country Happiness.Score
412    Bhutan           5.011
435 Sri Lanka           4.440
>
```

- Number of Countries in Happiest Group and Least Happiest Cluster Group above and below Median Value-

```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest.test.pc[happiest.test.pc$Happiness.Score > median(happiness.data$Happiness.Score)
, ])
[1] 26
> nrow(happiest.test.pc[happiest.test.pc$Happiness.Score < median(happiness.data$Happiness.Score)
, ])
[1] 2
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest.test.pc[least_happiest.test.pc$Happiness.Score > median(happiness.data$Happ
iness.Score), ])
[1] 0
> nrow(least_happiest.test.pc[least_happiest.test.pc$Happiness.Score < median(happiness.data$Happ
iness.Score), ])
[1] 46
```

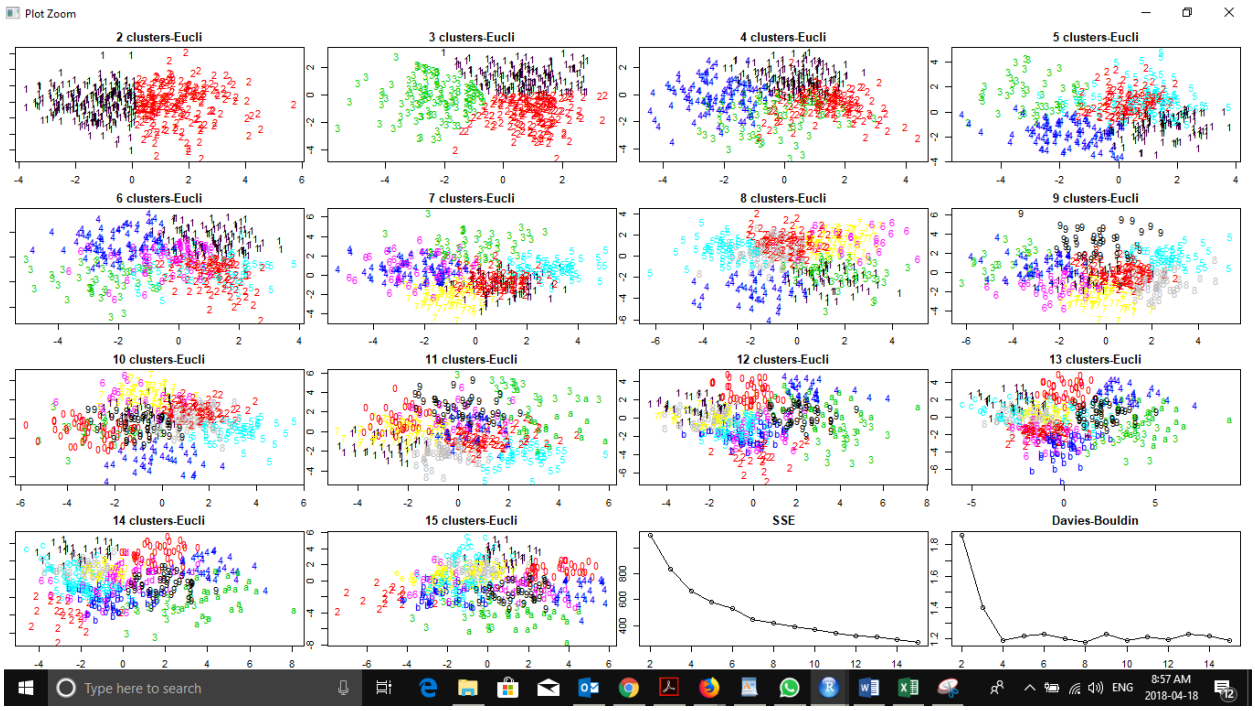
5.3. Whitened Dataset After ICA-Clustering

5.3.1. Whitened Dataset After ICA- Clustering

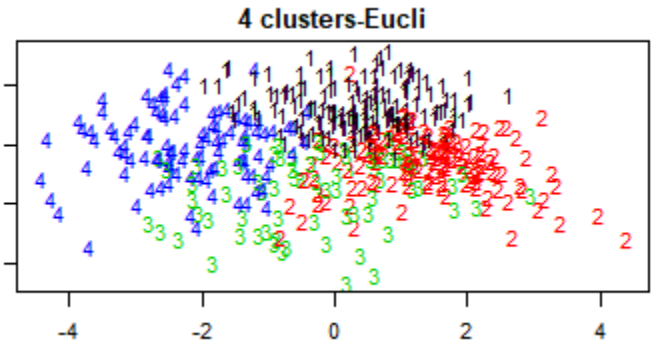
Cluster Analysis from range 2 to 15 for ICA Happiness Data Set.

In Cluster analysis, in this dataset we do not have any attributes for classification of data, therefore we have assigned numbers and colors based on the Clusters.

Optimal Value of ICA Dataset Set from Davis Bouldin is 4.



- **Optimal Cluster Value is 4-**



- **Best Seed and Centroids for the Cluster-**

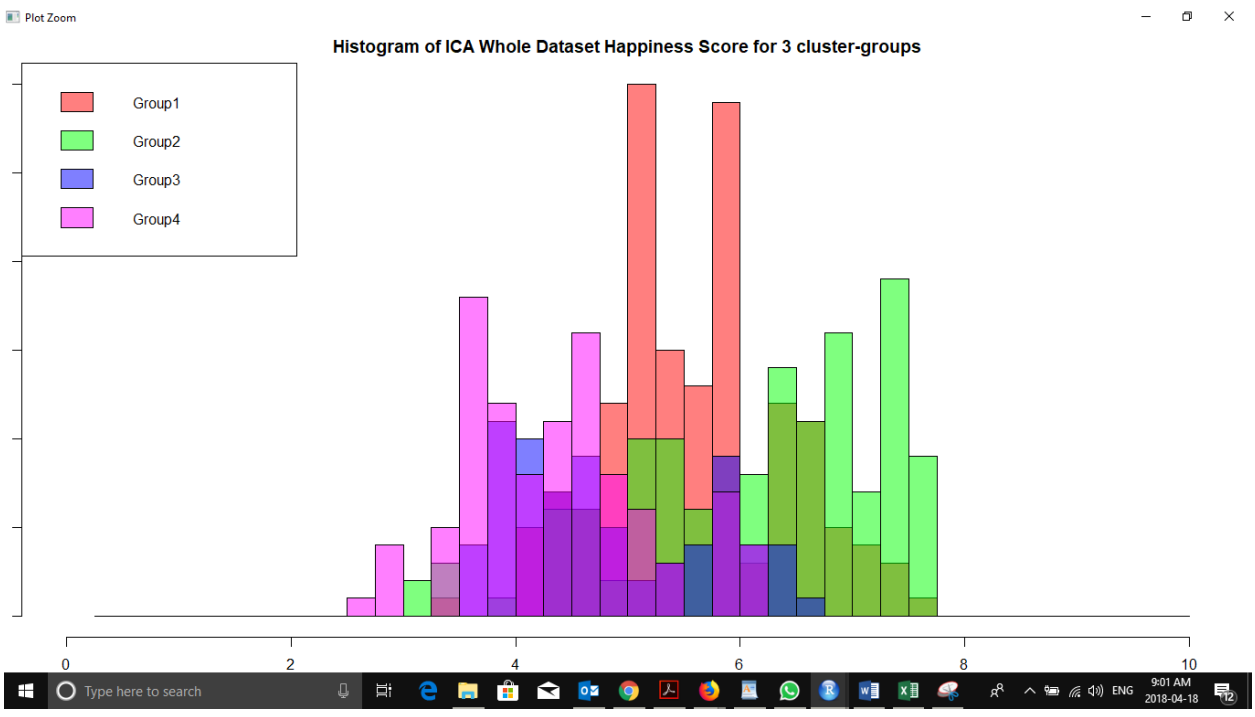
```
[1] "Best Seed for Cluster Size  4 is  2"
[1] "Total Wrong in Cluster Size  4 is  0"
[1] "Centroids for Cluster Size  4 are :"
        [,1]        [,2]       [,3]
1 -0.2442780 -0.56134525 -0.7947324
2  1.1229461 -0.04946334  0.2344608
3 -0.2458814  1.56013604 -0.2715028
4 -0.9470159 -0.18429367  1.0919569
K-means clustering with 4 clusters of sizes 157, 137, 73, 103
```

- **Sum of Squares –**

```
within cluster sum of squares by cluster:
[1] 172.2871 181.4966 130.5225 180.9027
 (between_SS / total_SS =  52.8 %)
```

- **Cluster Analysis-**

• Distribution of High Score for whole DataSet after ICA with optimal Value of Cluster Size from Davies Bouldin is 4-> There are less overlap with respect to all cluster group.



- Happiest Cluster Group and Least Happiest Cluster Group- Country with Highest Happiness Score in Group G2 is Switzerland(7.587) and with Least Happiness Score in Cluster Group G4 is Central African Republic(2.693).

```
> #Cluster Size
> print(km.ica$size)
[1] 157 137  73 103
>
> print(paste("ICA-Happiest Group is g", top, sep=""))
[1] "ICA-Happiest Group is g2"
>
> print(paste("ICA-Least Happiest Group is g", bottom, sep=""))
[1] "ICA-Least Happiest Group is g4"
>
> head(happiest.ica[order(happiest.ica$Happiness.Score, decreasing=TRUE), ])
        Country Happiness.Score
1   Switzerland           7.587
316      Norway           7.537
3       Denmark           7.527
159     Denmark           7.526
4        Norway           7.522
317     Denmark           7.522
>
> tail(least_happiest.ica[order(least_happiest.ica$Happiness.Score, decreasing=TRUE), ])
                    Country Happiness.Score
313                    Togo           3.303
157                 Burundi           2.905
315                 Burundi           2.905
469                 Burundi           2.905
158                    Togo           2.839
470 Central African Republic           2.693
>
```

- Countries above Median Value of Happiness Score in Least Happiness Cluster Group and Countries below Median values of Happiness Score of Happiest Cluster Group.

```
> print("Median Happiness Score for Whole Dataset-")
[1] "Median Happiness Score for Whole Dataset-"
> (median(happiness.data$Happiness.Score))
[1] 5.2825
>
> head(least_happiest.ica[least_happiest.ica$Happiness.Score > median(happiness.data$Happiness.Score), ]
)
       Country Happiness.Score
42 El Salvador           6.130
43   Guatemala           6.123
44  Uzbekistan           6.003
51     Bolivia           5.890
52     Moldova           5.889
57   Nicaragua           5.828
> head(happiest.ica[happiest.ica$Happiness.Score < median(happiness.data$Happiness.Score), ])
     Country Happiness.Score
81  Pakistan           5.194
110     Iran           4.686
112     Iraq           4.677
117    India           4.565
156    Syria           3.006
242   Bhutan           5.196
```

- Number of Countries in Happiest Group and Least Happiest Cluster Group above and below Median Value-

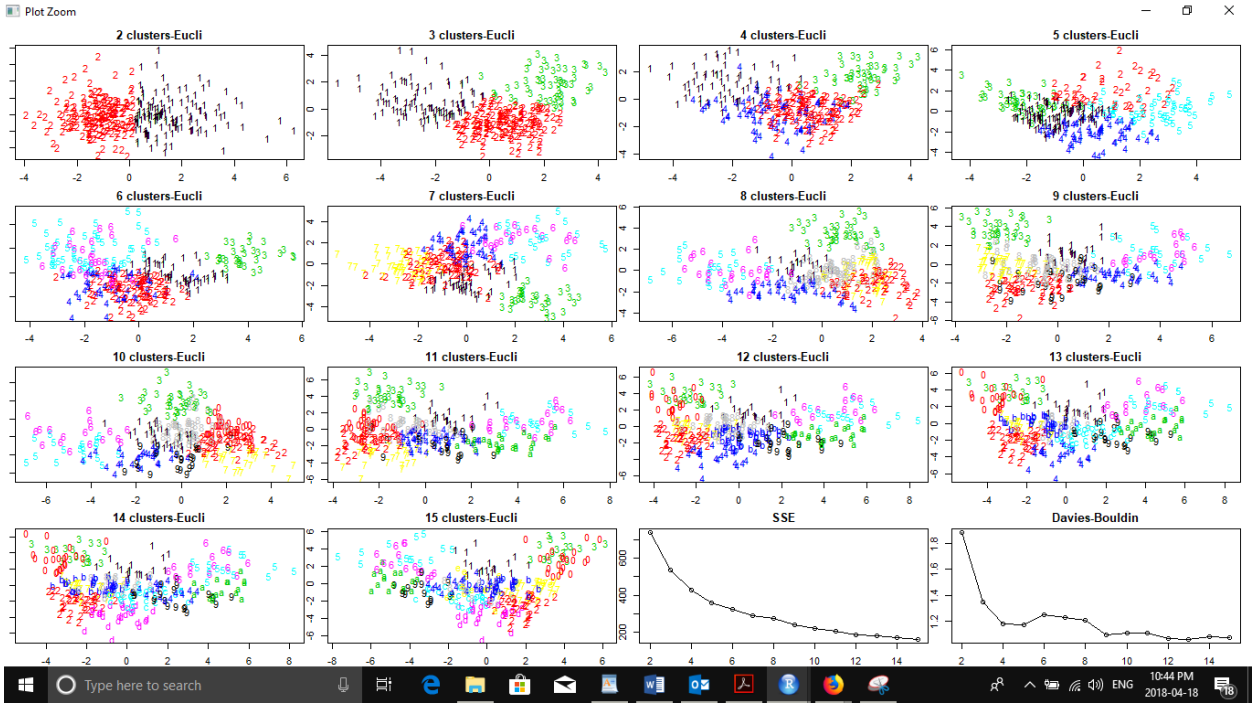
```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest.ica[happiest.ica$Happiness.Score > median(happiness.data$Happiness.Score), ])
[1] 105
> nrow(happiest.ica[happiest.ica$Happiness.Score < median(happiness.data$Happiness.Score), ])
[1] 32
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest.ica[least_happiest.ica$Happiness.Score > median(happiness.data$Happiness.Score), ]
)
[1] 13
> nrow(least_happiest.ica[least_happiest.ica$Happiness.Score < median(happiness.data$Happiness.Score), ]
)
[1] 90
```

5.3.2. Whitened Train Dataset After ICA- Clustering

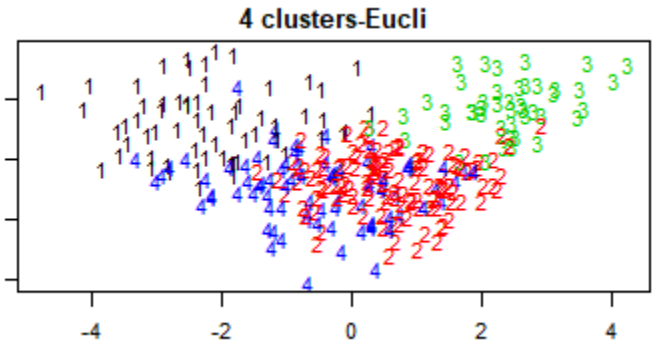
Cluster Analysis from range 2 to 15 for Train Dataset After ICA

In Cluster analysis, in this dataset we do not have any attributes for classification of data, therefore we have assigned numbers and colors based on the Clusters.

Optimal Value of Train Dataset after ICA from Davis Bouldin is 4.



- **Optimal Cluster Value is 4-**



- **Best Seed and Centroids for the Cluster-**

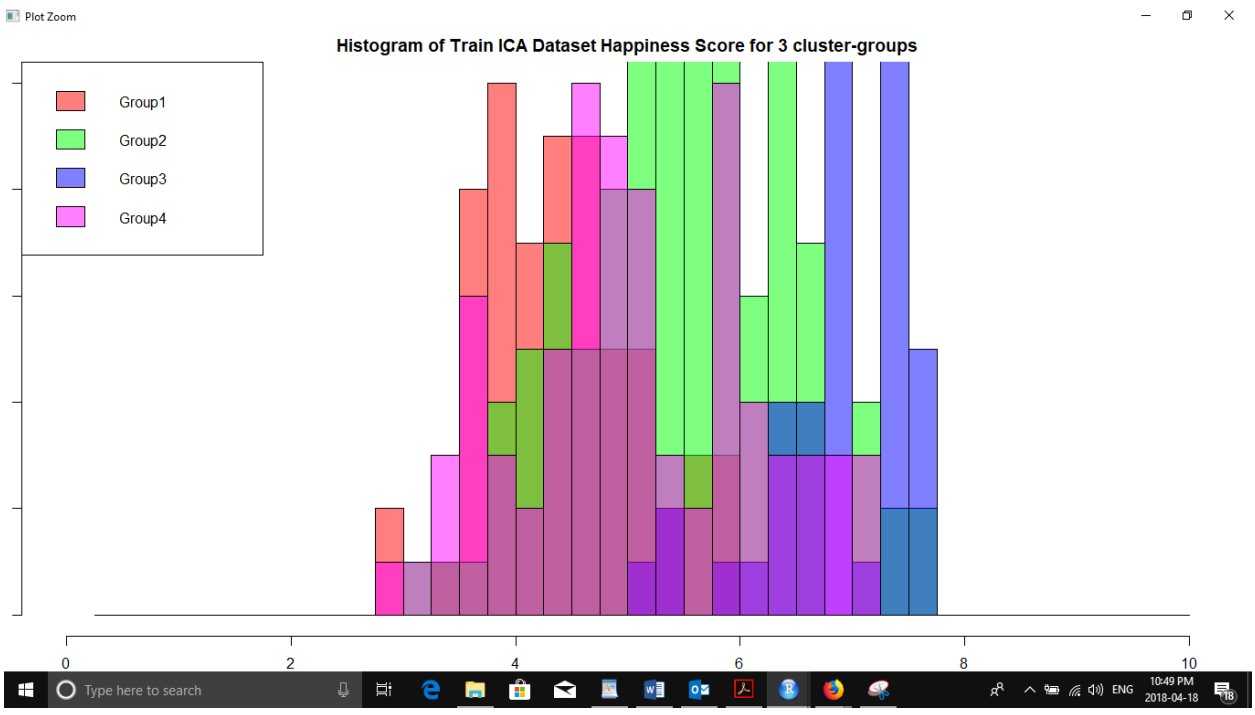
```
> km.train.ica <- clustering_euclidean(happiness.train.ica,happiness.train, 4)
[1] "Best Seed for Cluster Size  4 is  2"
[1] "Total Wrong in Cluster Size  4 is  0"
[1] "Centroids for Cluster Size  4 are :"
        [,1]       [,2]        [,3]
1 -1.2308766 -0.2834691 -0.90771377
2  0.5487566  0.6217961 -0.18281061
3  1.0690540 -1.5555059 -0.06201122
4 -0.4939673  0.1160972  1.06457131
K-means clustering with 4 clusters of sizes 64, 127, 45, 79
```

- **Sum of Squares –**

```
within cluster sum of squares by cluster:
[1] 141.16318 132.30965  46.91352 107.82732
 (between_SS / total_SS =  54.7 %)
```

- **Cluster Analysis-**

• Distribution of High Score for Train Dataset after ICA with optimal Value of Cluster Size from Davies Bouldin is 4-> There are less overlap with respect to all cluster group.



- Happiest Cluster Group and Least Happiest Cluster Group- Country with Highest Happiness Score in Group G3 is Switzerland(7.587) and with Least Happiness Score in Cluster Group G1 is Burundi(2.905).

```
> #Cluster Size
> print(km.train.ica$size)
[1]  64 127  45  79
>
> print(paste("Train-ICA-Happiest Group is g", top, sep=""))
[1] "Train-ICA-Happiest Group is g3"
>
> print(paste("Train-ICA-Least Happiest Group is g", bottom, sep=""))
[1] "Train-ICA-Least Happiest Group is g1"
>
> head(happiest.train.ica[order(happiest.train.ica$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
1   Switzerland           7.587
3       Denmark           7.527
159     Denmark           7.526
4        Norway           7.522
160 Switzerland           7.509
162      Norway           7.498
>
> tail(least_happiest.train.ica[order(least_happiest.train.ica$Happiness.Score, decreasing=TRUE),
 ])
        Country Happiness.Score
151  Ivory Coast           3.655
152 Burkina Faso           3.587
310       Rwanda           3.515
312  Afghanistan           3.360
157      Burundi           2.905
315      Burundi           2.905
> #
```

- Countries above Median Value of Happiness Score in Least Happiness Cluster Group and Countries below Median values of Happiness Score of Happiest Cluster Group.

```
> #Median happiness for the entire set of countries
> print("Median Happiness Score for train Dataset-")
[1] "Median Happiness Score for train Dataset-"
> (median(happiness.train$Happiness.Score))
[1] 5.286
>
> head(least_happiest.train.ica[least_happiest.train.ica$Happiness.Score > median(happiness.train
$Happiness.Score), ])
         Country Happiness.Score
59       Belarus           5.813
68       Algeria           5.605
70  Turkmenistan           5.548
207   Uzbekistan           5.987
219      Belarus           5.802
223 Turkmenistan           5.658
> head(happiest.train.ica[happiest.train.ica$Happiness.Score < median(happiness.train$Happiness.S
core), ])
     Country Happiness.Score
250 Pakistan           5.132
>
```

- Number of Countries in Happiest Group and Least Happiest Cluster Group above and below Median Value-

```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest.train.ica[happiest.train.ica$Happiness.Score > median(happiness.data$Happiness.Sc
ore), ])
[1] 44
> nrow(happiest.train.ica[happiest.train.ica$Happiness.Score < median(happiness.data$Happiness.Sc
ore), ])
[1] 1
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest.train.ica[least_happiest.train.ica$Happiness.Score > median(happiness.data$
Happiness.Score), ])
[1] 6
> nrow(least_happiest.train.ica[least_happiest.train.ica$Happiness.Score < median(happiness.data$
Happiness.Score), ])
[1] 58
```

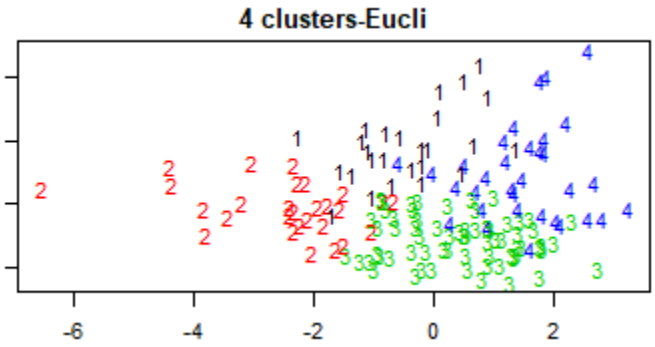
5.3.3. Whitened Test Dataset After ICA- Clustering

Cluster Analysis for Test ICA Happiness Data Set with Optimal Value 4.

In Cluster analysis, in this dataset we do not have any attributes for classification of data, therefore we have assigned numbers and colors based on the Clusters.

Optimal Value of Train ICA Dataset from Davis Bouldin is 4 which we use on Test dataset.

- **Optimal Cluster Value is 4-**



- **Best Seed and Centroids for the Cluster-**

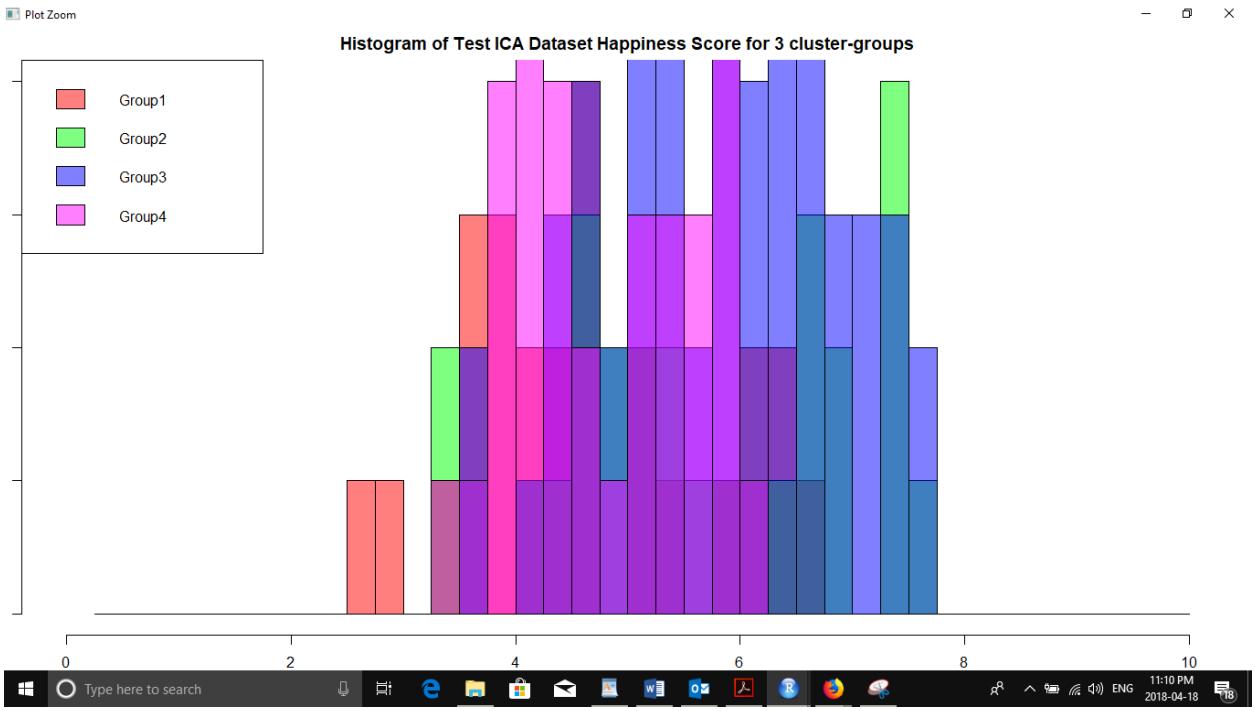
```
> km.test.ica <- clustering_euclidean(happiness.test.ica, happiness.test, 4)
[1] "Best Seed for Cluster Size  4 is  2"
[1] "Total Wrong in Cluster Size  4 is  0"
[1] "Centroids for Cluster Size  4 are :"
        [,1]       [,2]       [,3]
1  0.3928774  1.0685249 -1.0842319
2 -1.0468590 -0.8381992 -0.9354070
3 -0.3241695  0.2266597  0.7402824
4  1.2466898 -0.5977251  0.1224917
K-means clustering with 4 clusters of sizes 26, 28, 68, 33
```

- **Sum of Squares –**

```
Within cluster sum of squares by cluster:
[1] 42.35154 44.75530 74.15211 53.14106
 (between_SS / total_SS =  53.9 %)
```

- **Cluster Analysis-**

• Distribution of High Score for Test DataSet after ICA With optimal Value of Cluster Size from Davies Bouldin is 4-> There are less overlap with respect to all cluster group.



- Happiest Cluster Group and Least Happiest Cluster Group- Country with Highest Happiness Score in Group G3 is Norway(7.537) and with Least Happiness Score in Cluster Group G1 is Central African Republic(2.639).

```
> #Cluster Size
> print(km.test.ica$size)
[1] 26 28 68 33
>
> print(paste("Test-ICA-Happiest Group is g", top, sep=""))
[1] "Test-ICA-Happiest Group is g3"
>
> print(paste("Test-ICA-Least Happiest Group is g", bottom, sep=""))
[1] "Test-ICA-Least Happiest Group is g1"
>
> head(happiest.test.ica[order(happiest.test.ica$Happiness.Score, decreasing=TRUE), ])
       Country Happiness.Score
316     Norway           7.537
317     Denmark          7.522
319 Switzerland          7.494
320     Finland          7.469
324      Sweden          7.284
326      Israel          7.213
>
> tail(least_happiest.test.ica[order(least_happiest.test.ica$Happiness.Score, decreasing=TRUE), ]
)
                    Country Happiness.Score
458                   Benin           3.657
462             South Sudan           3.591
464                  Guinea           3.507
465                    Togo           3.495
469                 Burundi           2.905
470 Central African Republic           2.693
```

- Countries above Median Value of Happiness Score in Least Happiness Cluster Group and Countries below Median values of Happiness Score of Happiest Cluster Group.

```
> print("Median Happiness Score for testing Dataset-")
[1] "Median Happiness Score for testing Dataset-"
> (median(happiness.test$Happiness.Score))
[1] 5.279
>
> head(least_happiest.test.ica[least_happiest.test.ica$Happiness.Score > median(happiness.test$Ha
ppiness.Score), ])
                Country Happiness.Score
336 United Arab Emirates           6.648
350                Qatar           6.375
352         Saudi Arabia           6.344
353  Trinidad and Tobago           6.168
354               Kuwait           6.105
365               Belize           5.956
> head(happiest.test.ica[happiest.test.ica$Happiness.Score < median(happiness.test$Happiness.Scor
e), ])
       Country Happiness.Score
394      China           5.273
398 Montenegro           5.237
399    Morocco           5.235
402     Greece           5.227
403    Lebanon           5.225
404   Portugal           5.195
```

- Number of Countries in Happiest Group and Least Happiest Cluster Group above and below Median Value-

```
> print("Number of countries above and below Median Value in Happiest Group")
[1] "Number of countries above and below Median Value in Happiest Group"
>
> nrow(happiest.test.ica[happiest.test.ica$Happiness.Score > median(happiness.data$Happiness.Scor
e), ])
[1] 44
> nrow(happiest.test.ica[happiest.test.ica$Happiness.Score < median(happiness.data$Happiness.Scor
e), ])
[1] 24
>
> print("Number of countries above and below Median Value in Least Happiest Group")
[1] "Number of countries above and below Median Value in Least Happiest Group"
>
> nrow(least_happiest.test.ica[least_happiest.test.ica$Happiness.Score > median(happiness.data$Ha
ppiness.Score), ])
[1] 6
> nrow(least_happiest.test.ica[least_happiest.test.ica$Happiness.Score < median(happiness.data$Ha
ppiness.Score), ])
[1] 20
```

6. SUPERVISED LEARNING

Supervised learning is a type of system in which both input and desired output data are provided. Input and output data are labelled for classification to provide a learning basis for future data processing.

Analysis-

1. We first applied classification tree based on region to standardized reduced training dataset. We did visualization of tree.
2. We predicted using our model and tried to check it with test dataset.
3. Due to poor results we tried regression tree based on happiness score, will all other attributes be acting as predictors.
4. We found out SSE to be 8 and plotted residual graph.
5. We predicted using our model and prepared confusion matrix between predicted model and test dataset. Accuracy was 67.5%. We found happiness rank as most influential predictor for happiness score.
6. Then we did regression tree on score again but this time without happiness rank being a predictor.
7. We found out SSE to be 22.73 and plotted residual graph.
8. Based on model obtained in 7^{th} we predicted data and compared with test dataset. Accuracy was 45%
9. We applied recursive partitioning (rpart) on the standardized reduced training dataset. It was based on region with all other attributes acted as the predictors. Resulting tree was plotted for visualization.
10. Based on the model we obtained, we did prediction for the test dataset.
11. After that we applied rpart based on happiness score with all other attributes acted as the predictors. Resulting tree was plotted for visualization. We concluded that major predictor for determining nodes was happiness rank.
12. We found out SSE to be 2.56, least till now.
13. Based on the model we obtained in 11^{th} step, we did prediction for the test dataset. We obtained its accuracy though confusion matrix. It was 98.8%
14. To check how the other predictors, determine happiness score, we did rparts again based on score will all other attributes except happiness rank as our predictors. Resulting tree was plotted for visualization. There was no dominant factor and all predictors were effective in determining respective nodes.
15. We found SSE to be 35.33, highest till now.
16. Based on the model we obtained in 14^{th} step, we did prediction for the test dataset. We obtained its accuracy through confusion matrix just like step 4. It was only 55%.
17. We concluded that from our supervised learning that our model build using Rpart give better accuracy then classification trees.

	Tree with rank	**Tree without rank**	**Rpart with rank**	**Rpart without rank**
SSE	8	22.73	2.56	35.33
Accuracy (%)	67.5	45	98.8	55

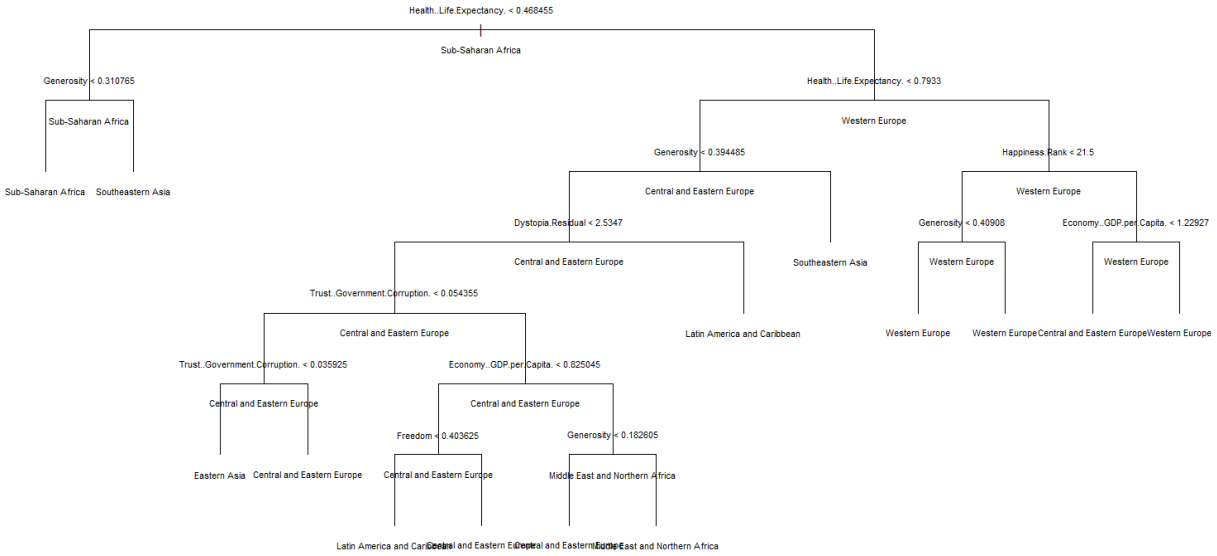
6.1. Classification tree based on region

- **Applying classification tree and tree visualisation**

First, we applied classification tree to reduced standardized train dataset.

```
region.tree <- tree(Region~Happiness.Rank+Happiness.Score+Economy..GDP.per.Capita.+
                      Family+Health..Life.Expectancy.+Freedom+Trust..Government.Corruption.
                    +Generosity+Dystopia.Residual ,method="class",data = happiness.std.reduced.train)
```

We choose target attribute as region. As we can see we selected all other attributes as predictors. Here is the tree we obtained:



From here we can conclude that generosity has low contribution is in sub-Saharan Africa for happiness. Health expectancy is high in Europe. Eastern Asia does not trust their government much. Central and Eastern Europe don't consider freedom highly for happiness. Generosity is high in Europe.

- **Prediction**

We predicted using our tree. As 2017 dataset (our test set) did not have region attribute, we just used head function on both to tally our predictions.

```
> head(test_pred)
    Australia and New Zealand Central and Eastern Europe Eastern Asia Latin America and Caribbean
372                         0                 0.3333333          0.5                   0.1666667
394                         0                 0.3333333          0.5                   0.1666667
409                         0                 0.5714286          0.0                   0.1428571
393                         0                 0.0000000          0.0                   0.1666667
406                         0                 0.0000000          0.0                   0.4000000
387                         0                 0.0000000          0.0                   0.0000000
    Middle East and Northern Africa na North America Southeastern Asia Southern Asia Sub-Saharan Africa Western Europe
372                      0.00000000  0             0          0.0000000          0.0          0.0000000              0
394                      0.00000000  0             0          0.0000000          0.0          0.0000000              0
409                      0.00000000  0             0          0.2857143          0.0          0.0000000              0
393                      0.83333333  0             0          0.0000000          0.0          0.0000000              0
406                      0.20000000  0             0          0.0000000          0.4          0.0000000              0
387                      0.02222222  0             0          0.0000000          0.0          0.9777778              0
```

The first and second entries has 33% probability of being in Central and Eastern Europe. Third entry has 57% of being in Central and Easter Europe. Fourth entry has 83% of belonging to Middle East and Norther Africa. Fifth entry has 40% probability for Latin America and Caribbean. Finally, last entry has 97% for Sub-Saharan Africa.

Let's trace these same entries in our actual test dataset.

```
> head(test)
        Country
372     Romania
394       China
409     Vietnam
393      Kosovo
406    Honduras
387 Philippines
```

As we can see here only Romania and Honduras were classified correctly. Low accuracy in determining region is since all the attributes are almost independent of geography.

6.2. Regression tree based on happiness score

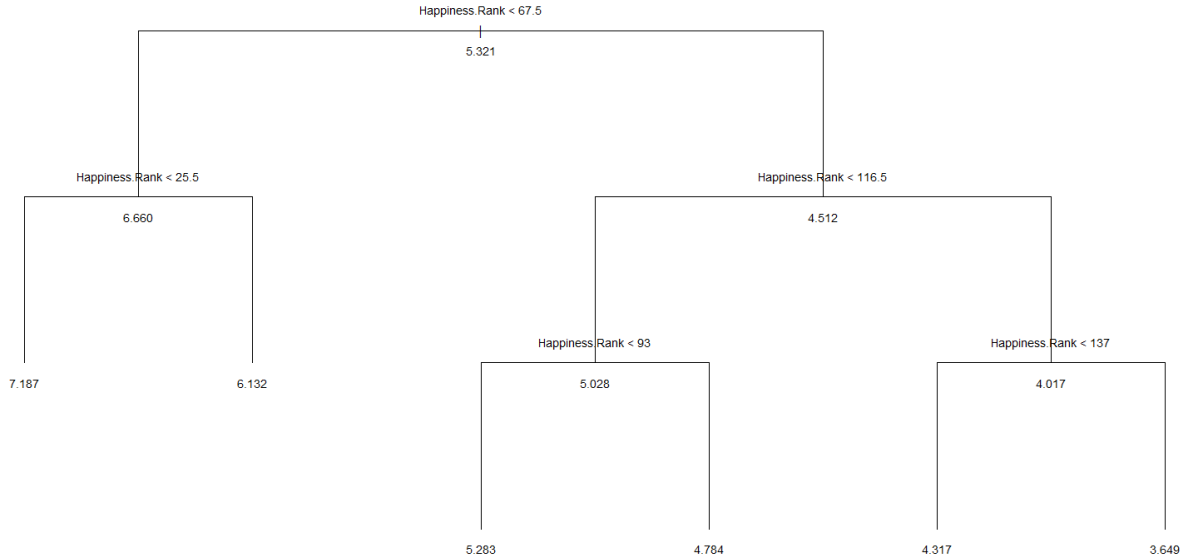
- **Applying regression tree and tree visualisation**

First, we applied regression tree to reduced standardized train dataset.

```
score.tree <- tree(Happiness.Score~Happiness.Rank+Region+Economy..GDP.per.Capita.+
                     Family+Health..Life.Expectancy.+Freedom+Trust..Government.Corruption.
                   +Generosity+Dystopia.Residual ,method="anova",data = happiness.std.reduced.train)
```

As we saw in previous section that region was not a attribute to predict, now we will make our model to predict happiness score. Here the target variable chosen was happiness score. All other attributes were used to predict the happiness score. We obtained the following tree:

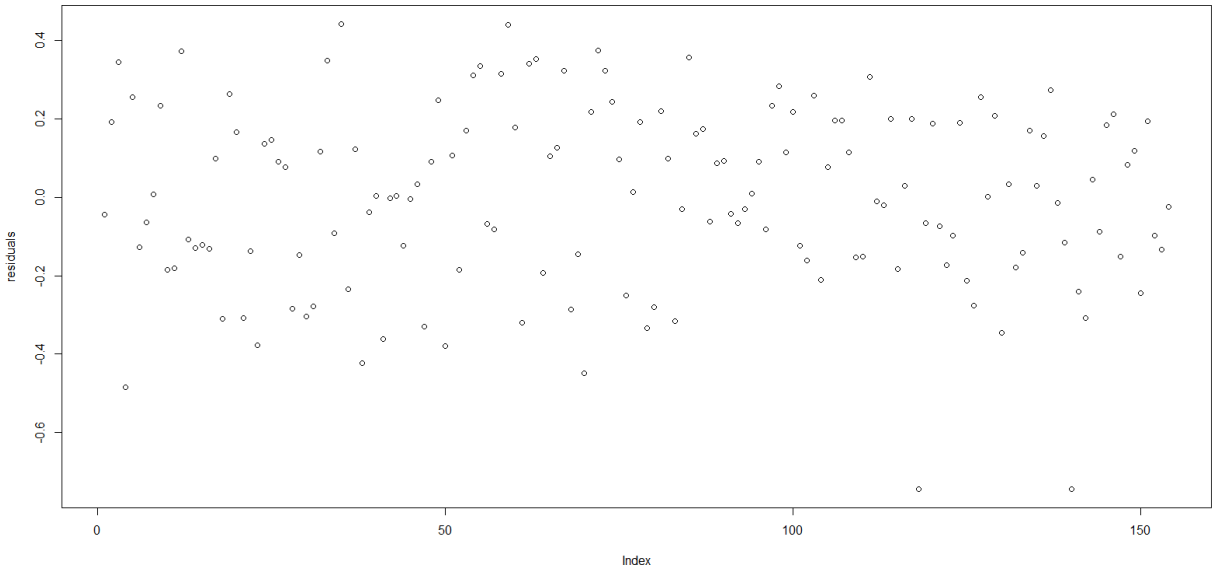
```
> (score.tree)
node), split, n, deviance, yval
      * denotes terminal node

 1) root 154 223.7000 5.321
   2) Happiness.Rank < 67.5 58  20.6200 6.660
     4) Happiness.Rank < 25.5 29   1.8890 7.187 *
     5) Happiness.Rank > 25.5 29   2.6180 6.132 *
   3) Happiness.Rank > 67.5 96  36.3500 4.512
     6) Happiness.Rank < 116.5 47   3.9170 5.028
      12) Happiness.Rank < 93 23   0.4778 5.283 *
      13) Happiness.Rank > 93 24   0.5114 4.784 *
     7) Happiness.Rank > 116.5 49   7.9100 4.017
      14) Happiness.Rank < 137 27   0.5268 4.317 *
      15) Happiness.Rank > 137 22   1.9630 3.649 *
```



We can see here that Happiness rank was most important parameter in determining the happiness score. Good Happiness ranking is an indication of high happiness score and vice versa.

- **Residual Analysis**

We plotted the residual of the above model:



As we can see, there is no pattern in this graph.

We squared and added them up to obtain SSE:

```
> r<-(residuals(score.tree))
> sum(r^2)
[1] 7.986807
```

SSE obtained was approximately 8.

- **Prediction**

Based on the model obtained above we tried to predict the test dataset.

```
model.2<-score.tree
test_pred <- predict(model.2,test)
Actual.Values<-round(test$Happiness.Score)
r.pred<-floor(test_pred)
Predicted.Values<-as.numeric(r.pred)
table(Predicted.Values,Actual.Values)
confusion(Predicted.Values, Actual.Values)
```

As the happiness score was in decimals from 1-10, it had more number of possibilities of different values then the total number of entries in test dataset itself. So, we rounded up the score for easier classification. Here is our confusion matrix:

```
                Actual.Values
Predicted.Values  3  4  5  6  7
               3  4  8  0  0  0
               4  0 11 17  0  0
               5  0  0 11  0  0
               6  0  0  0 14  1
               7  0  0  0  0 14
> confusion(Predicted.Values, Actual.Values)
Overall accuracy = 0.675
```

We can see lots of misclassifications here. Overall accuracy was 0.675 or 67.5%.

6.3. Regression Tree based on happiness score without happiness rank

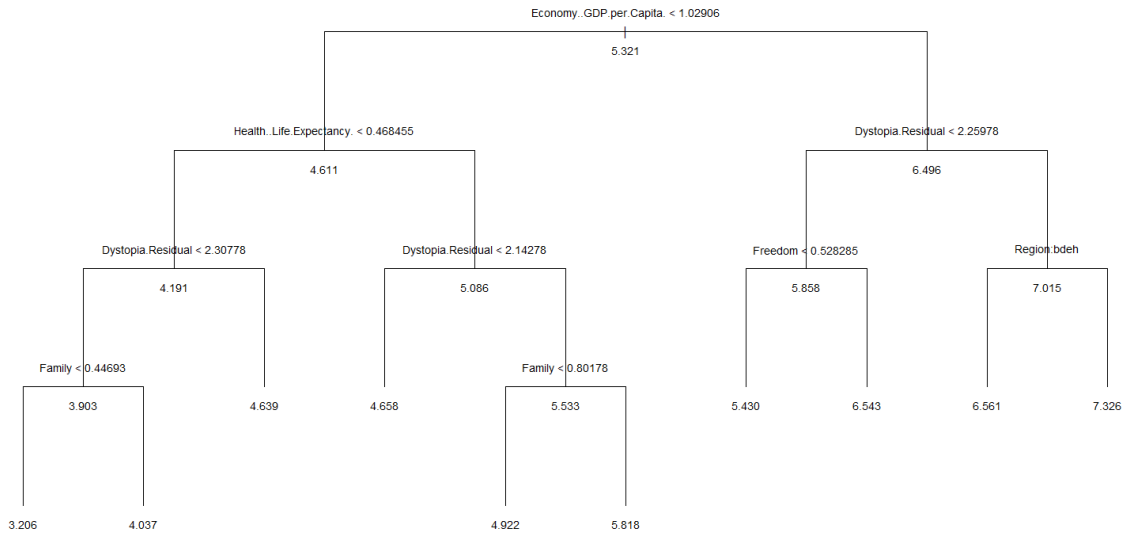
- **Applying regression tree and tree visualisation**

As happiness rank was involved at each level in tree, we wanted to see how other parameters interact with each to determine happiness score in the absence of happiness rank, the most dominant predictor. Hence, we modeled regression tree of reduced standardized train data based on happiness score. But this time we did not consider happiness rank as one of the predictors.

```
score.tree2 <- tree(Happiness.Score~Region+Economy..GDP.per.Capita.+
                      Family+Health..Life.Expectancy.+Freedom+Trust..Government.Corruption.
                    +Generosity+Dystopia.Residual ,method="anova",data = happiness.std.reduced.train)
```

Here the target variable chosen was happiness score again. All other attributes except happiness rank were used to predict the happiness score. We obtained the following tree:

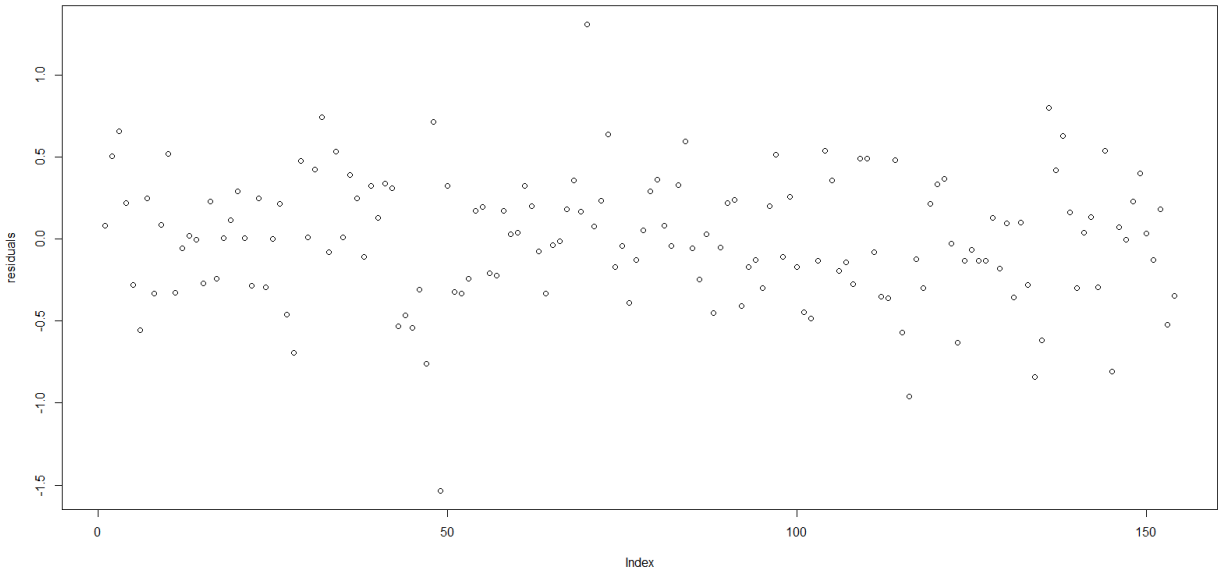
```
> (score.tree2)
node), split, n, deviance, yval
      * denotes terminal node

 1) root 154 223.7000 5.321
   2) Economy..GDP.per.Capita. < 1.02906 96  54.3600 4.611
     4) Health..Life.Expectancy. < 0.468455 51  16.4800 4.191
       8) Dystopia.Residual < 2.30778 31   5.7740 3.903
        16) Family < 0.44693 5   0.3447 3.206 *
        17) Family > 0.44693 26   2.5310 4.037 *
       9) Dystopia.Residual > 2.30778 20   4.1230 4.639 *
     5) Health..Life.Expectancy. > 0.468455 45  18.7700 5.086
      10) Dystopia.Residual < 2.14278 23   3.4700 4.658 *
      11) Dystopia.Residual > 2.14278 22   6.6800 5.533
        22) Family < 0.80178 7   0.2876 4.922 *
        23) Family > 0.80178 15   2.5590 5.818 *
   3) Economy..GDP.per.Capita. > 1.02906 58  40.7700 6.496
     6) Dystopia.Residual < 2.25978 26  14.2900 5.858
      12) Freedom < 0.528285 16   5.5580 5.430 *
      13) Freedom > 0.528285 10   1.1080 6.543 *
     7) Dystopia.Residual > 2.25978 32   7.2720 7.015
      14) Region: Central and Eastern Europe,Latin America and Caribbean,Middle East and Northern Africa,Southeastern Asia 13
   2.1320 6.561 *
      15) Region: Australia and New Zealand,North America,Western Europe 19   0.6187 7.326 *
```



As we can see high economy and GDP per capita lead to high happiness score. Low Dystopia residual lead to low score. High family score was a prerequisite for having high happiness score. In the end freedom was deciding factor between mediocre and high happiness score.

- **Residual Analysis**

We plotted the residual of the above model:



As we can see, there is no pattern in this graph.

We squared and added them up to obtain SSE:

```
> r<-(residuals(score.tree2))
> sum(r^2)
[1] 22.73211
>
```

SSE obtained was 22.73. It increased from our previous case.

- **Prediction**

Based on the model obtained above we tried to predict the test dataset.
We again rounded up the score for easier classification. Here is our confusion matrix:

```
                 Actual.Values
Predicted.Values  3  4  5  6  7
               3  2  1  0  0  0
               4  2 18 19  1  0
               5  0  0  8 10  2
               6  0  0  1  3  8
               7  0  0  0  0  5
> confusion(Predicted.Values, Actual.Values)
Overall accuracy = 0.45
```

As we can see here, there are lost of misclassifications compared to previous confusion matrix. Accuracy obtained was only 45%.

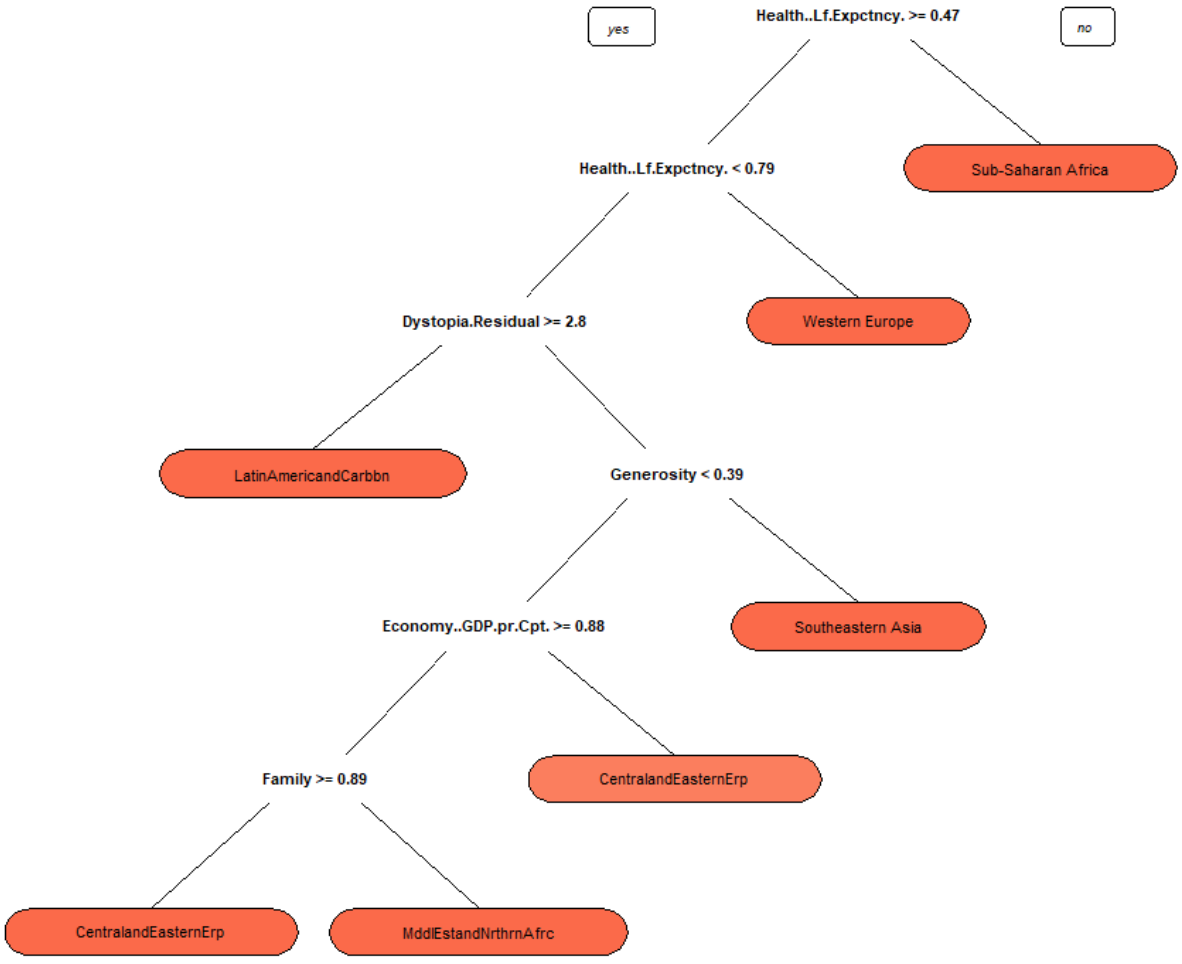
6.4. Recursive partitioning based on region

- **Applying rpart and tree visualisation**

First, we applied rpart to reduced standardized train dataset.

```
##########Applying rpart###################
region.rp.std <- rpart(Region~Happiness.Rank+Happiness.Score+Economy..GDP.per.Capita.+
                         Family+Health..Life.Expectancy.+Freedom+Trust..Government.Corruption.
                       +Generosity+Dystopia.Residual ,method="class",data = happiness.std.reduced.train)
```

We choose target attribute as region. As we can see we selected all other attributes as predictors. Here is the tree we obtained:



From here we can conclude that Sub-Saharan Africa has lowest life expectancy. Western Europe has highest life expectancy. Generosity is higher in Southeast Asia. Economy is intermediate in Central and Eastern Europe. It is high in Central Europe, Eastern Europe and Middle East. Though compared to Middle East, family is a more determining factor for happiness in Central and Eastern Europe.

- **Prediction**

We predicted using our tree. As 2017 dataset (our test set) did not have region attribute, we just used head function on both to tally our predictions.

```
> head(test_pred)
    Australia and New Zealand Central and Eastern Europe Eastern Asia Latin America and Caribbean
372                         0                 0.5384615   0.07692308                  0.15384615
394                         0                 0.5384615   0.07692308                  0.15384615
409                         0                 0.4761905   0.04761905                  0.19047619
393                         0                 0.5384615   0.07692308                  0.15384615
406                         0                 0.4761905   0.04761905                  0.19047619
387                         0                 0.0000000   0.00000000                  0.01923077
    Middle East and Northern Africa na North America Southeastern Asia Southern Asia Sub-Saharan Africa Western Europe
372                      0.23076923  0             0        0.00000000   0.00000000          0.0000000              0
394                      0.23076923  0             0        0.00000000   0.00000000          0.0000000              0
409                      0.04761905  0             0        0.09523810   0.14285714          0.0000000              0
393                      0.23076923  0             0        0.00000000   0.00000000          0.0000000              0
406                      0.04761905  0             0        0.09523810   0.14285714          0.0000000              0
387                      0.01923077  0             0        0.03846154   0.03846154          0.8846154              0
```

The first, second and fourth entries has 53% probability of being in Central and Eastern Europe. Third and fifth entries has 47% of being in Central and Easter Europe. Finally, last entry has 88% for Sub-Saharan Africa.

Let's trace these same entries in our actual test dataset.

```
        Country
372     Romania
394       China
409     Vietnam
393      Kosovo
406    Honduras
387 Philippines
```

As we can see here only Romania was classified correctly. Low accuracy in determining region is since all the attributes are almost independent of geography.

6.5. Recursive partitioning based on happiness score

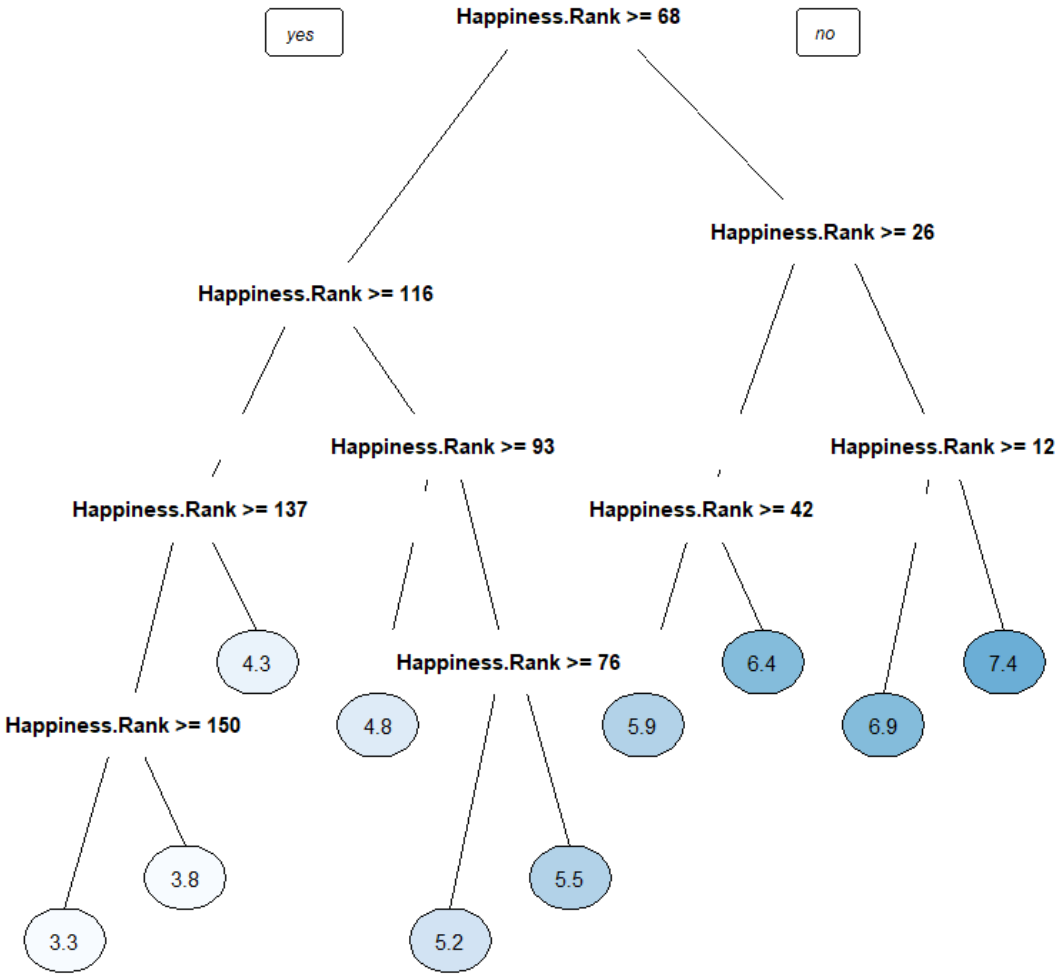
- **Applying rpart and tree visualisation**

First, we applied rpart to reduced standardized train dataset.

```
set.seed(123)
dtree_fit <- train(Happiness.Score ~., data = happiness.std.reduced.train, method = "rpart",
                   parms = list(split = "information"),
                   trControl=trctrl,
                   tuneLength = 10)
```

Here the target variable chosen was happiness score. All other attributes were used to predict the happiness score. We obtained the following tree:

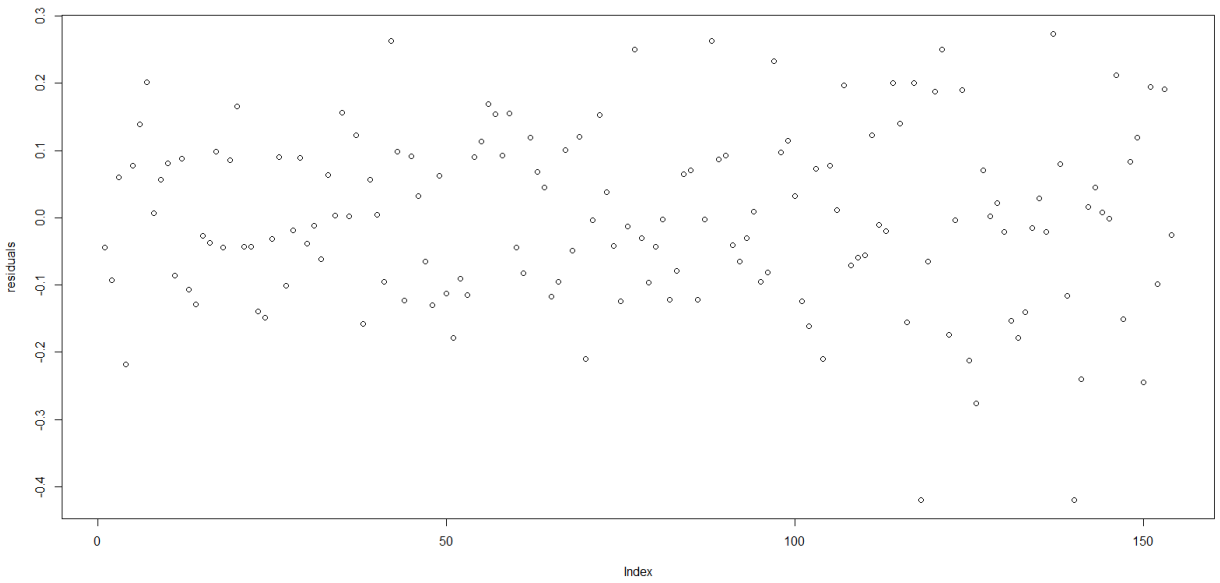
```
n= 154

node), split, n, deviance, yval
      * denotes terminal node

 1) root 154 223.73190000 5.320825
   2) Happiness.Rank>=67.5 96  36.35083000 4.511990
     4) Happiness.Rank>=116.5 49   7.90992300 4.016980
       8) Happiness.Rank>=137 22   1.96347100 3.648545
        16) Happiness.Rank>=150 8   0.54034990 3.324375 *
        17) Happiness.Rank< 150 14   0.10203440 3.833786 *
       9) Happiness.Rank< 137 27   0.52675810 4.317185 *
     5) Happiness.Rank< 116.5 47   3.91656900 5.028064
      10) Happiness.Rank>=93 24   0.51137250 4.783750 *
      11) Happiness.Rank< 93 23   0.47782800 5.283000
        22) Happiness.Rank>=76.5 15   0.05790773 5.188133 *
        23) Happiness.Rank< 76.5 8   0.03180888 5.460875 *
   3) Happiness.Rank< 67.5 58  20.62409000 6.659586
     6) Happiness.Rank>=25.5 29   2.61823300 6.132448
      12) Happiness.Rank>=41.5 15   0.25460760 5.866600 *
      13) Happiness.Rank< 41.5 14   0.16764290 6.417286 *
     7) Happiness.Rank< 25.5 29   1.88914400 7.186724
      14) Happiness.Rank>=11.5 14   0.21571350 6.949500 *
      15) Happiness.Rank< 11.5 15   0.15024570 7.408133 *
>
```



We can see here that Happiness rank was most important parameter in determining the happiness score. Good Happiness ranking is an indication of high happiness score and vice versa.

- **Residual Analysis**

We plotted the residual of the above model:



As we can see, there is no pattern in this graph.

We squared and added them up to obtain SSE:

```
> r<-(residuals(dtree_fit))
> sum(r^2)
[1] 2.558441
```

SSE obtained was 2.56. This is the least SSE we have seen till now.

- **Prediction**

Based on the model obtained above we tried to predict the test dataset.

```
model<-dtree_fit
test<-happiness.std.reduced.test
test_pred <- predict(model,test)
Actual.Values<-round(happiness.std.reduced.test$Happiness.Score)
r.pred<-round(test_pred)
Predicted.Values<-as.numeric(r.pred)
table(Predicted.Values,Actual.Values)
confusion(Predicted.Values, Actual.Values)
```

As the happiness score was in decimals from 1-10, it had more number of possibilities of different values then the total number of entries in test dataset itself. So, we rounded up the score for easier classification. Here is our confusion matrix:

```
> table(Predicted.Values,Actual.Values)
                Actual.Values
Predicted.Values  3  4  5  6  7
               3  4  0  0  0  0
               4  0 19  0  0  0
               5  0  0 28  0  0
               6  0  0  0 14  1
               7  0  0  0  0 14
> confusion(Predicted.Values, Actual.Values)
Overall accuracy = 0.988
```

As we can see here, almost all the actual values match predicted values except one. One entry having actual score near 7 was predicted to have score near 6 instead. So, we obtained 98.8% accuracy from this model.

6.6. Recursive partitioning based on happiness score without happiness rank

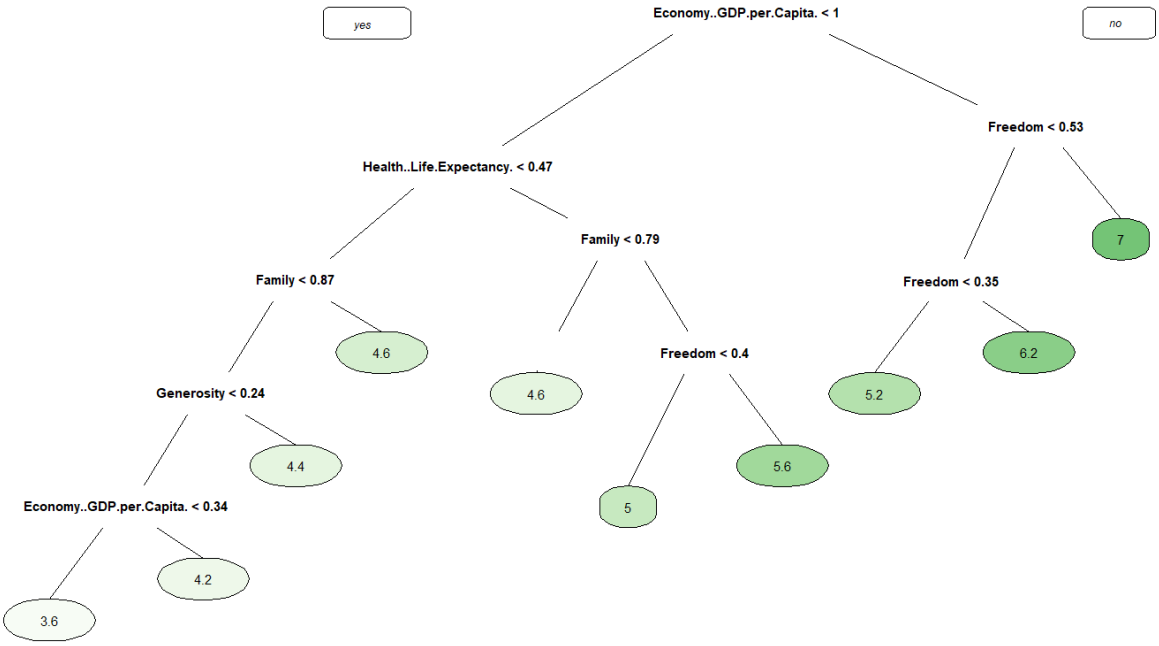
- **Applying rpart and tree visualisation**

As happiness rank was involved at each level in tree, we wanted to see how other parameters interact with each to determine happiness score in the absence of happiness rank, the most dominant predictor. Hence, we did rpart of reduced standardized train data based on happiness score. But this time we did not consider happiness rank as one of the predictors.

```
##########Analysis without rank##########
########Applying rpart############
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(123)
dtree_fit.2 <- train(Happiness.Score ~Region+Economy..GDP.per.Capita.+
                       Family+Health..Life.Expectancy.+Freedom+Trust..Government.Corruption.
                     +Generosity+Dystopia.Residual, data = happiness.std.reduced.train, method = "rpart",
                     parms = list(split = "information"),
                     trControl=trctrl,
                     tuneLength = 10)
```

Here the target variable chosen was happiness score again. All other attributes except happiness rank were used to predict the happiness score. We obtained the following tree:

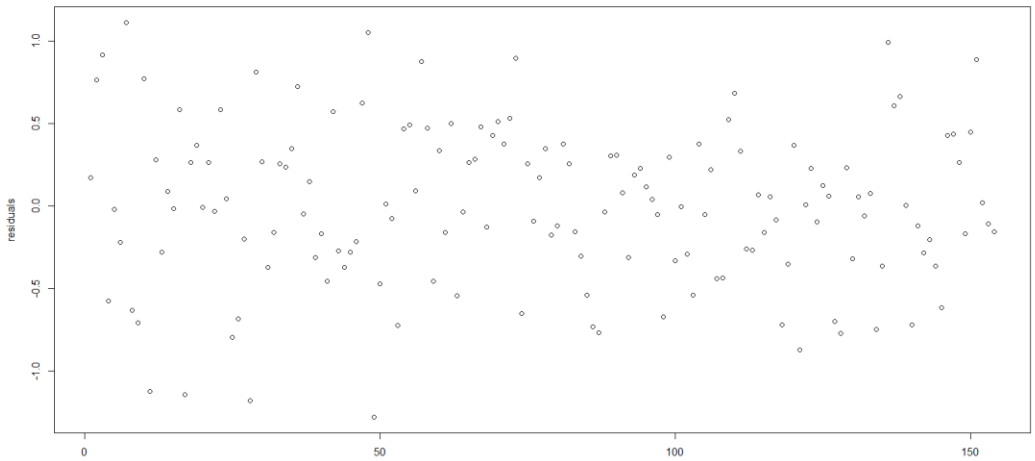
```
n= 154

node), split, n, deviance, yval
      * denotes terminal node

 1) root 154 223.7319000 5.320825
   2) Economy..GDP.per.Capita.< 1.02906 96  54.3572100 4.610521
     4) Health..Life.Expectancy.< 0.468455 51  16.4761800 4.191373
       8) Family< 0.866205 38  11.3970500 4.050526
        16) Generosity< 0.243575 26   5.2249920 3.866808
          32) Economy..GDP.per.Capita.< 0.339435 15   2.4785740 3.624000 *
          33) Economy..GDP.per.Capita.>=0.339435 11   0.6561749 4.197909 *
        17) Generosity>=0.243575 12   3.3931010 4.448583 *
       9) Family>=0.866205 13   2.1217910 4.603077 *
     5) Health..Life.Expectancy.>=0.468455 45  18.7664700 5.085556
      10) Family< 0.78546 16   1.8696040 4.566188 *
      11) Family>=0.78546 29  10.1997900 5.372103
        22) Freedom< 0.40281 9   2.0479840 4.956000 *
        23) Freedom>=0.40281 20   5.8923050 5.559350 *
   3) Economy..GDP.per.Capita.>=1.02906 58  40.7713100 6.496500
     6) Freedom< 0.5285 28  17.0834100 5.926357
      12) Freedom< 0.354725 8   2.9021990 5.176125 *
      13) Freedom>=0.354725 20   7.8773070 6.226450 *
     7) Freedom>=0.5285 30   6.0911690 7.028633 *
```



As we can see high economy and GDP per capita lead to high happiness score. Good life expectancy increases low score to higher one. High freedom and family contributed in intermediate happiness score. Countries having highest score has highest life expectancy as well.

- **Residual Analysis**

We plotted the residual of the above model:



As we can see, there is no pattern in this graph.

We squared and added them up to obtain SSE:

```
> r<-(residuals(dtree_fit.2))
> sum(r^2)
[1] 35.33021
```

SSE obtained was 35.33. This is the highest SSE we have seen till now.

- **Prediction**

Based on the model obtained above we tried to predict the test dataset.

```
########Prediction and Accuracy#########
model.2<-dtree_fit.2
test<-happiness.std.reduced.test
test_pred <- predict(model.2,test)
Actual.Values<-round(test$Happiness.Score)
r.pred<-floor(test_pred)
Predicted.Values<-as.numeric(r.pred)
table(Predicted.Values,Actual.Values)
confusion(Predicted.Values, Actual.Values)
```

We again rounded up the score for easier classification. Here is our confusion matrix:

```
                Actual.Values
Predicted.Values  3  4  5  6  7
               3  1  0  0  0  0
               4  3 19 17  1  0
               5  0  0  8  4  0
               6  0  0  2  6  5
               7  0  0  1  3 10
> confusion(Predicted.Values, Actual.Values)
Overall accuracy = 0.55
```

As we can see here, there are lost of misclassifications compared to previous confusion matrix. Accuracy obtained was only 55%. Hence, we can conclude that happiness rank is most important predictor in determining happiness score and without it we can obtain desirable accuracy in our model.

7. REFERENCES

[1] Kaggle.com. (2018). *World Happiness Report | Kaggle*. [online] Available at: https://www.kaggle.com/unsdsn/world-happines4

[2] People.math.carleton.ca. (2018). *Shirley Mills' Home Page*. [online] Available at: http://people.math.carleton.ca/~smills/.

[3] En.wikipedia.org. (2018). *Data reduction*. [online] Available at: https://en.wikipedia.org/wiki/Data_reduction.

[4] Towards Data Science. (2018). *Data Mining 101 — Dimensionality and Data reduction*. [online] Available at: https://towardsdatascience.com/data-mining-101-dimensionality-and-data-reduction-2a8fa427b092.

[5] Rpubs.com. (2018). *RPubs - Unsupervised Learning in R*. [online] Available at: https://rpubs.com/williamsurles/310847.

[6] Dataaspirant. (2018). *Decision Tree Classifier implementation in R*. [online] Available at: http://dataaspirant.com/2017/02/03/decision-tree-classifier-implementation-in-r/.

[7] Brownlee, J. (2018). *Non-Linear Classification in R with Decision Trees*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/non-linear-classification-in-r-with-decision-trees/.

APPENDIX-I: CODE

```
############################################################
#
# Final Project--STAT5703-HEMANT GUPTA-101062246
#                         ANURAG DAS-101089268
#                         Manoj Kakarla-101071887
#
#
############################################################
#packages:
install.packages("rggobi")
install.packages("cluster")
install.packages("stats")
install.packages("fpc")
install.packages("flexclust")
install.packages("plyr")
install.packages("fastICA")

install.packages("rpart")
install.packages("caret")
install.packages("rpart.plot")
install.packages("MASS")
install.packages("DAAG")
install.packages("tree")


## Setting the Path of Directory

## Setting the Path of Directory

drive="C:"
path.upto <- paste("STAT5703-HEMANT-101062246-Final-Project", sep="/" )
code.dir <- paste(drive, path.upto,"Code", sep="/")
data.dir <- paste(drive, path.upto,"Data", sep="/")
work.dir <- paste(drive, path.upto,"Work", sep="/")
setwd(work.dir)

##Reading the CSV format of Cars Data
happiness.file<- paste(data.dir,"AllCombined.csv",sep="/")

happiness.data <- read.csv(happiness.file, header=TRUE)

head(happiness.data)

############################################################
#
#               FUNCTIONS
#
############################################################
```

```
#
##Function: Not Contain in the Set
#

"%w/o%"<- function(x,y) x[!x %in% y]

#
## Function Set the indices for the training/test sets
#
get.train <- function (data.sz, train.sz)
{
  set.seed(123)
  # Take subsets of data for training/test samples
  # Return the indices
  train.ind <- sample(data.sz, train.sz)
  test.ind <- (data.sz) %w/o% train.ind
  list(train=train.ind, test=test.ind)
}

#
## Function Set the indices for the  sets
#
get.subset <- function (data, size)
{
  set.seed(123)
  # Take subsets of data for training/test samples
  # Return the indices
  data_subset <- sample(data, size)

}
#
#Function:========DBI Function=======#
#
Davies.Bouldin <- function(A, SS, m) {
  # A  - the centres of the clusters
  # SS - the within sum of squares
  # m  - the sizes of the clusters
  N <- nrow(A)   # number of clusters
  # intercluster distance
  S <- sqrt(SS/m)
  # Get the distances between centres
  M <- as.matrix(dist(A))
  # Get the ratio of intercluster/centre.dist
  R <- matrix(0, N, N)
  for (i in 1:(N-1)) {
    for (j in (i+1):N) {
      R[i,j] <- (S[i] + S[j])/M[i,j]
      R[j,i] <- R[i,j]
    }
  }
  return(mean(apply(R, 1, max)))
```

```
}

#
#Function for Data Standardization
#

###################################
# Standardize data
###################################
f.data.std <- function(data) {
  data <- as.matrix(data)
  bar <- apply(data, 2, mean)
  s <- apply(data, 2, sd)
  t((t(data) - bar)/s)
}

#####################################
#WhitenedData: Centre and sphere data
#####################################
Sphere.Data <- function(data) {
  data <- as.matrix(data)
  data <- t(t(data) - apply(data, 2, mean))
  data.svd <- svd(var(data))
  sphere.mat <- t(data.svd$v %*% (t(data.svd$u) * (1/sqrt(data.svd$d))))
  return(data %*% sphere.mat)
}


##############################
#
#FUNCTION: error_cal- For calculating wrong classification in Cluster
#
############################
error_cal <- function(tbl,cluster_size)
{
  wrong_data <- 0
  for(clust in 1:cluster_size)
  {
    wrong_data <- wrong_data + (sum(tbl[,clust])-max(tbl[,clust]))
  }
  return(wrong_data)
}
#################################################
#Function: Clustering_euclidean
# Cluster Size Varies 2 to 15
# SSE and DBI is determined at each iteration
#################################################

clustering_euclidean <- function(data_set,data_set.orig, limit)
{
  # set.seed(654321)
  oldpar <- par(mfrow = c(4,4))
```

```
  par(mar=c(2,1,2,1))

  errs <- rep(0, 7)
  DBI <- rep(0, 7)
  ##Package(cluster)
  library(cluster)
  library(stats)
  library(fpc)
  library(flexclust)


  #Loop for different CLuster Size
  for (i in limit)
  {
    min_error <- 450
    min_error_km <- 0
    best.seed <- 0
    #Loop for Seed
    for (j in 2:1000)
    {

      set.seed(j)
      #Clustering Using K means
      KM <- kmeans((data_set[,]), i, 25)
      ct.km <- table(KM$cluster, KM$cluster)

      #Calculating toal wrong data for each seed
      error <- error_cal(ct.km,i)
      if(min_error > error)
      {
        #Storing Error count and Kmeans output and best seed for min error
        min_error <- error
        min_error_km <-KM
        best.seed <- j
      }

    }
    print(paste("Best Seed for Cluster Size " , i ,"is " , best.seed))

    print(paste("Total Wrong in Cluster Size " , i ,"is " , min_error))

    print(paste("Centroids for Cluster Size " , i ,"are :"))

    print(min_error_km$centers)

    print(min_error_km)

    #Plotting the CLuster
    plotcluster(data_set, col=min_error_km$cluster,min_error_km$cluster,
main=paste(i,"clusters-Eucli"))

    if(length(limit) > 1)
```

```
    {
      #CLuster Analysis
      errs[i-1] <- sum(min_error_km$withinss)
      DBI[i-1] <- Davies.Bouldin(min_error_km$centers,
min_error_km$withinss, min_error_km$size)


    }

  }
  if(length(limit) > 1)
  {
    plot(2:15, errs, main = "SSE")
    lines(2:15, errs)
    #
    plot(2:15, DBI, main = "Davies-Bouldin")
    lines(2:15, DBI)
    #
  }
  else
  {
    return(min_error_km)
  }
  return(errs)
}



################################################################################
############
   #                 Training and Test Dataset

################################################################################
############
#Training Dataset of year 2015 and 2016
happiness.train <- happiness.data[happiness.data$Year!=2017,]

#Test Dataset
##Reading the CSV format of Cars Data
happiness.test <- happiness.data[happiness.data$Year==2017,]


out = c (nrow(happiness.train), nrow(happiness.test))

##Setting X axis name
x.names=c("Train","Test")

barplot(out, main="Data Spliting",xaxt="n")
axis(1,at = 1:2,labels=x.names)

################################################################################
############
```

```
#######                    DATA VISUALIZATION

##########################################################################
############
happiness.15.16<- happiness.train
happiness.15.16$Region = as.factor(happiness.15.16$Region)

happiness.15.16$Region<-as.numeric(happiness.train$Region)

#happiness.15.16a <- as.data.frame(happiness.15.16)

#happiness.15.16[is.na(happiness.15.16$Region),2] <- 0
## Factoring the Car Data ORigin Column


happiness.15.16 <- happiness.15.16[order(happiness.15.16$Region),]


happiness.col <- happiness.15.16$Region

## Cleaning the data for Column Year
#for (j in 1:length(happiness.15.16$Region)){
#  if(setequal(happiness.15.16[j,2],"North America")){
#    happiness.15.16[j,2] <- 2
#  }
#  else
#  {}
#}

#> unique(happiness.15.16$Region)
#[1] Australia and New Zealand       Central and Eastern Europe
#[3] Eastern Asia                    Latin America and Caribbean
#[5] Middle East and Northern Africa North America
#[7] Southeastern Asia               Southern Asia
#[9] Sub-Saharan Africa              Western Europe
## Ggobi PLot : Scatterplot and Parallel COordinate gives overall picture of
###              data but its hard to scale it.

library(rggobi)
g <- ggobi(happiness.15.16)

display(g[1], "Scatterplot Matrix")

display(g[1], "Parallel Coordinates Display")

## Plotting the Cars.dat with respect to origin Pairs Plot as it also
contain scale for each parameter


### CO-PLOT :-To get the concentation of points with yearly for each origin
```

```
par(mfrow=c(2,4))

plot(happiness.data$Year,happiness.data$Economy..GDP.per.Capita., col=
happiness.data$Year)
plot(happiness.data$Year,happiness.data$Family, col= happiness.data$Year)
plot(happiness.data$Year,happiness.data$Health..Life.Expectancy., col=
happiness.data$Year)
plot(happiness.data$Year,happiness.data$Freedom, col= happiness.data$Year)
plot(happiness.data$Year,happiness.data$Trust..Government.Corruption., col=
happiness.data$Year)
plot(happiness.data$Year,happiness.data$Generosity, col=
happiness.data$Year)
plot(happiness.data$Year,happiness.data$Dystopia.Residual, col=
happiness.data$Year)


#pairs(happiness.15.16[,3:12], col=happiness.col)


#Categorize data by Region
#Region - Australia and Newzealand
happiness.AN <- happiness.15.16[happiness.15.16$Region==1,-1]
#Region- Central_Eastern_Europe
happiness.CEE <-happiness.15.16[happiness.15.16$Region==2,-1]
#Region-Eastern_Asia
happiness.EA <-happiness.15.16[happiness.15.16$Region==3,-1]
#Region-Latin_America_Caribbean
happiness.LAC <- happiness.15.16[happiness.15.16$Region==4,-1]
#Region-Middle_East_Northern_Africa
happiness.MENA <- happiness.15.16[happiness.15.16$Region==5,-1]
#Region-North_America\
happiness.NA <- happiness.15.16[happiness.15.16$Region==7,-1]
#Region-Sourtheastern_Asia
happiness.SEA <- happiness.15.16[happiness.15.16$Region==8,-1]
#Region-Sourthern_Asia
happiness.SA <- happiness.15.16[happiness.15.16$Region==9,-1]
#Region-Sub_Saharan_Africa
happiness.SSA <- happiness.15.16[happiness.15.16$Region==10,-1]
#Region-Western_Europe
happiness.WE <- happiness.15.16[happiness.15.16$Region==11,-1]


par(mfrow=c(2,5))

##Setting X axis name
x.names=c("AN","CEE","EA","LAC","MENA","NA","SEA","SA","SSA","WE")

## PLotting the Box Plot of All 6 parameter based on Origin and keeping
na.rm =TRUE to avoid NA values
for(i in 2:10){
  plot.title=colnames(happiness.15.16)[i+1]
```

```
boxplot(happiness.AN[,i],happiness.CEE[,i],happiness.EA[,i],happiness.LAC[,i
],happiness.MENA[,i],happiness.NA[,i],happiness.SEA[,i],happiness.SA[,i],hap
piness.SSA[,i],happiness.WE[,i],main=plot.title,xaxt="n", col=
heat.colors(10), na.rm=TRUE)
  axis(1,at = 1:10,labels=x.names)
}



#####We use Box plot to get the Aveage value of the data per year
#####Coplot shows the distribition of data point but clearly didnot mention
#####average avlue concentartion . This helps in deduction Accurately.

#Categorize data by Year
#Region - Australia and Newzealand
happiness.AN.15=happiness.AN[happiness.AN$Year==2015,-1]
happiness.AN.16=happiness.AN[happiness.AN$Year==2016,-1]

##BoxPlot of data based on Yearly variation and keeping na.rm =TRUE to avoid
NA values
par(mfrow=c(3,3))
x.names=c("2015","2016")
for(i in 1:9){

  plot.title <- paste(colnames(happiness.AN)[i+1], "A&N",sep = '_')
  boxplot(happiness.AN.15[,i],happiness.AN.16[,i],main=plot.title, xaxt="n",
col= heat.colors(2),na.rm=TRUE)
  axis(1,at = 1:2,labels=x.names)
}


#Region - Central Eastern Europe
happiness.CEE.15=happiness.CEE[happiness.CEE$Year==2015,-1]
happiness.CEE.16=happiness.CEE[happiness.CEE$Year==2016,-1]

##BoxPlot of data based on Yearly variation CEEd keeping na.rm =TRUE to
avoid NA values
par(mfrow=c(3,3))
x.names=c("2015","2016")
for(i in 1:9){

  plot.title <- paste(colnames(happiness.CEE)[i+1], "CEE",sep = '_')
  boxplot(happiness.CEE.15[,i],happiness.CEE.16[,i],main=plot.title,
xaxt="n", col= heat.colors(2),na.rm=TRUE)
  axis(1,at = 1:2,labels=x.names)
}

#Region - Eastern ASia
happiness.EA.15=happiness.EA[happiness.EA$Year==2015,-1]
happiness.EA.16=happiness.EA[happiness.EA$Year==2016,-1]
```

```
##BoxPlot of data based on Yearly variation EAd keeping na.rm =TRUE to avoid
NA values
par(mfrow=c(3,3))
x.names=c("2015","2016")
for(i in 1:9){

  plot.title <- paste(colnames(happiness.EA)[i+1], "EA",sep = '_')
  boxplot(happiness.EA.15[,i],happiness.EA.16[,i],main=plot.title, xaxt="n",
col= heat.colors(2),na.rm=TRUE)
  axis(1,at = 1:2,labels=x.names)
}

#Region - Latin America and Carribean
happiness.LAC.15=happiness.LAC[happiness.LAC$Year==2015,-1]
happiness.LAC.16=happiness.LAC[happiness.LAC$Year==2016,-1]

##BoxPlot of data based on Yearly variation ead keeping na.rm =TRUE to avoid
NA values
par(mfrow=c(3,3))
x.names=c("2015","2016")
for(i in 1:9){

  plot.title <- paste(colnames(happiness.LAC)[i+1], "LA&C",sep = '_')
  boxplot(happiness.LAC.15[,i],happiness.LAC.16[,i],main=plot.title,
xaxt="n", col= heat.colors(2),na.rm=TRUE)
  axis(1,at = 1:2,labels=x.names)
}

#Region - Middle East and NOrthern Africa
happiness.MENA.15=happiness.MENA[happiness.MENA$Year==2015,-1]
happiness.MENA.16=happiness.MENA[happiness.MENA$Year==2016,-1]

##BoxPlot of data based on Yearly variation and keeping na.rm =TRUE to avoid
NA values
par(mfrow=c(3,3))
x.names=c("2015","2016")
for(i in 1:9){

  plot.title <- paste(colnames(happiness.MENA)[i+1], "ME& NA",sep = '_')
  boxplot(happiness.MENA.15[,i],happiness.MENA.16[,i],main=plot.title,
xaxt="n", col= heat.colors(2),na.rm=TRUE)
  axis(1,at = 1:2,labels=x.names)
}


#Region - North America
happiness.NA.15=happiness.NA[happiness.NA$Year==2015,-1]
happiness.NA.16=happiness.NA[happiness.NA$Year==2016,-1]

##BoxPlot of data based on Yearly variation and keeping na.rm =TRUE to avoid
NA values
par(mfrow=c(3,3))
```

```
x.names=c("2015","2016")
for(i in 1:9){

  plot.title <- paste(colnames(happiness.NA)[i+1], "NA",sep = '_')
  boxplot(happiness.NA.15[,i],happiness.NA.16[,i],main=plot.title, xaxt="n",
col= heat.colors(2),na.rm=TRUE)
  axis(1,at = 1:2,labels=x.names)
}

#Region - South Eastern Asia
happiness.SEA.15=happiness.SEA[happiness.SEA$Year==2015,-1]
happiness.SEA.16=happiness.SEA[happiness.SEA$Year==2016,-1]

##BoxPlot of data based on Yearly variation and keeping na.rm =TRUE to avoid
NA values
par(mfrow=c(3,3))
x.names=c("2015","2016")
for(i in 1:9){

  plot.title <- paste(colnames(happiness.SEA)[i+1], "SEA",sep = '_')
  boxplot(happiness.SEA.15[,i],happiness.SEA.16[,i],main=plot.title,
xaxt="n", col= heat.colors(2),na.rm=TRUE)
  axis(1,at = 1:2,labels=x.names)
}

#Region - Southern Asia
happiness.SA.15=happiness.SA[happiness.SA$Year==2015,-1]
happiness.SA.16=happiness.SA[happiness.SA$Year==2016,-1]

##BoxPlot of data based on Yearly variation and keeping na.rm =TRUE to avoid
NA values
par(mfrow=c(3,3))
x.names=c("2015","2016")
for(i in 1:9){

  plot.title <- paste(colnames(happiness.SA)[i+1], "SA",sep = '_')
  boxplot(happiness.SA.15[,i],happiness.SA.16[,i],main=plot.title, xaxt="n",
col= heat.colors(2),na.rm=TRUE)
  axis(1,at = 1:2,labels=x.names)
}

#Region - Sub Saharan Africa
happiness.SSA.15=happiness.SSA[happiness.SSA$Year==2015,-1]
happiness.SSA.16=happiness.SSA[happiness.SSA$Year==2016,-1]

##BoxPlot of data based on Yearly variation and keeping na.rm =TRUE to avoid
NA values
par(mfrow=c(3,3))
x.names=c("2015","2016")
for(i in 1:9){

  plot.title <- paste(colnames(happiness.SSA)[i+1], "SSA",sep = '_')
```

```
  boxplot(happiness.SSA.15[,i],happiness.SSA.16[,i],main=plot.title,
xaxt="n", col= heat.colors(2),na.rm=TRUE)
  axis(1,at = 1:2,labels=x.names)
}

#Region - Western Europe
happiness.WE.15=happiness.WE[happiness.WE$Year==2015,-1]
happiness.WE.16=happiness.WE[happiness.WE$Year==2016,-1]

##BoxPlot of data based on Yearly variation and keeping na.rm =TRUE to avoid
NA values
par(mfrow=c(3,3))
x.names=c("2015","2016")
for(i in 1:9){

  plot.title <- paste(colnames(happiness.WE)[i+1], "WE",sep = '_')
  boxplot(happiness.WE.15[,i],happiness.WE.16[,i],main=plot.title, xaxt="n",
col= heat.colors(2),na.rm=TRUE)
  axis(1,at = 1:2,labels=x.names)
}


#########################################################################
#########
# DIMENSION REDUCTION-PCA

#########################################################################
#########


happiness.std <- f.data.std(happiness.data[5:11])
# Get principal component vectors using prcomp
pc.happiness <- prcomp(happiness.std[,])
summary(pc.happiness)
plot(pc.happiness)
# First  principal components
happiness.pc <- data.frame(pc.happiness$x[,1:3])
head(happiness.pc)



happiness.train.std<- f.data.std(happiness.train[5:11])
# Get principal component vectors using prcomp
pc.train.std <- prcomp(happiness.train.std)
summary(pc.train.std)
plot(pc.train.std)

# First  principal components
happiness.train.pc <- data.frame(pc.train.std$x[,1:3])
head(happiness.train.pc)
```

```
happiness.test.std<- f.data.std(happiness.test[5:11])
# Get principal component vectors using prcomp
pc.test.std <- prcomp(happiness.test.std)
summary(pc.test.std)
plot(pc.test.std)

# First  principal components
happiness.test.pc <- data.frame(pc.test.std$x[,1:3])
head(happiness.test.pc)




################################################################################
#############

##########################DIMENSION REDUCTION-
ICA#########################################################

################################################################################
#############
library(fastICA)


#Whitening the whole happiness dataset
happiness.white <- Sphere.Data(happiness.data[,5:11])

##Taking number of components as 3 as want to compare it with result of PCA
3 components

happiness.white.ica <- fastICA(happiness.white, 3, alg.typ = "parallel", fun
= "logcosh", alpha = 1,
                     method = "R", row.norm = FALSE, maxit = 200, tol =
0.0001, verbose =
                       TRUE)

happiness.ica<-happiness.white.ica$S
#Estimated Source Matrix
head(happiness.ica)



#Whitening the Train Happiness dataset
happiness.train.white <- Sphere.Data(happiness.train[,5:11])

##Taking number of components as 3 as want to compare it with result of PCA
3 components
happiness.train.white.ica <- fastICA(happiness.train.white, 3, alg.typ =
"parallel", fun = "logcosh", alpha = 1,
                            method = "R", row.norm = FALSE, maxit = 200, tol =
0.0001, verbose =
```

```
                            TRUE)
#Estimated Source Matrix
happiness.train.ica<-happiness.train.white.ica$S
head(happiness.train.ica)



#Whitening the Test Happiness dataset
happiness.test.white <- Sphere.Data(happiness.test[,5:11])

##Taking number of components as 3 as want to compare it with result of PCA
3 components
happiness.test.white.ica <- fastICA(happiness.test.white, 3, alg.typ =
"parallel", fun = "logcosh", alpha = 1,
                                method = "R", row.norm = FALSE, maxit = 200,
tol = 0.0001, verbose =
                                  TRUE)
#Estimated Source Matrix
happiness.test.ica<-happiness.test.white.ica$S
head(happiness.test.ica)



##############################################################################
############

#  DATA REDUCTION

##############################################################################
############

###################################RAW
DATASET###################################
cluster_range <- 2:15

##Clustering on train and test Raw DataSet
happiness.ret <- clustering_euclidean(happiness.data[5:11],happiness.data,
cluster_range)

#Optimal Cluster Size is 3
# Create a K-Means cluster with 3 groups based on cols 5:11
# (GDP, Family, Life Expectancy, Freedom, Generosity, Corruption, Dystopia
Residual)
km <- clustering_euclidean(happiness.data[5:11],happiness.data, 3)

g1<-happiness.data[km$cluster == 1,]$Happiness.Score
g2<-happiness.data[km$cluster == 2,]$Happiness.Score
g3<-happiness.data[km$cluster == 3,]$Happiness.Score
# plot option "col=rgb(x,x,x,0.5)"" gives fill transparency

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

```
hist(g1, xlim=c(0,10), col=rgb(1,0,0,0.5), breaks=seq(0.25,10,0.25), main =
"Histogram of Raw Dataset Happiness Score for 3 cluster-groups", xlab =
"Country Happiness Score")
hist(g2, xlim=c(0,10), col=rgb(0,1,0,0.5), breaks=seq(0.25,10,0.25), add=T)
hist(g3, xlim=c(0,10), col=rgb(0,0,1,0.5), breaks=seq(0.25,10,0.25), add=T)
legend("topleft", c("Group1", "Group2", "Group3")
       , fill=c(rgb(1,0,0,0.5),rgb(0,1,0,0.5),rgb(0,0,1,0.5)) )


#Who is in the Happiest and least happiest Group?
top <- which.max(c(mean(g1),mean(g2), mean(g3))) # which is the top group
bottom <- which.min(c(mean(g1),mean(g2), mean(g3))) # which is the top group


happiest <- happiness.data[km$cluster == top, c(1,4)]
least_happiest <- happiness.data[km$cluster == bottom, c(1,4)]

#Cluster Size
print(km$size)

print(paste("RAW-Happiest Group is g", top, sep=""))

print(paste("RAW-Least Happiest Group is g", bottom, sep=""))

#Countries with highest and least Happiness score in clusters
head(happiest[order(happiest$Happiness.Score, decreasing=TRUE), ])

tail(least_happiest[order(least_happiest$Happiness.Score, decreasing=TRUE),
])

#Identify who in the top group has lower happiness than the
#median happiness for the entire set of countries

print("Median Happiness Score for Whole Dataset-")
(median(happiness.data$Happiness.Score))

head(least_happiest[least_happiest$Happiness.Score >
median(happiness.data$Happiness.Score), ])
head(happiest[happiest$Happiness.Score <
median(happiness.data$Happiness.Score), ])


print("Number of countries above and below Median Value in Happiest Group")

nrow(happiest[happiest$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(happiest[happiest$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Least Happiest
Group")
```

```
nrow(least_happiest[least_happiest$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(least_happiest[least_happiest$Happiness.Score <
median(happiness.data$Happiness.Score), ])


##################################Standard
DATASET#################################
cluster_range <- 2:15

##Clustering on train and test Raw DataSet
happiness.std.ret <- clustering_euclidean(happiness.std,happiness.data,
cluster_range)

#Optimal Cluster Size is 3
# Create a K-Means cluster with 3 groups based on cols 5:11
# (GDP, Family, Life Expectancy, Freedom, Generosity, Corruption, Dystopia
Residual)
km.std <- clustering_euclidean(happiness.std,happiness.data, 3)

g1.std<-happiness.data[km.std$cluster == 1,]$Happiness.Score
g2.std<-happiness.data[km.std$cluster == 2,]$Happiness.Score
g3.std<-happiness.data[km.std$cluster == 3,]$Happiness.Score
# plot option "col=rgb(x,x,x,0.5)"" gives fill transparency

par(mfrow=c(1,1))

hist(g1.std, xlim=c(0,10), col=rgb(1,0,0,0.5), breaks=seq(0.25,10,0.25),
main = "Histogram of STD. Dataset Happiness Score for 3 cluster-groups",
xlab = "Country Happiness Score")
hist(g2.std, xlim=c(0,10), col=rgb(0,1,0,0.5), breaks=seq(0.25,10,0.25),
add=T)
hist(g3.std, xlim=c(0,10), col=rgb(0,0,1,0.5), breaks=seq(0.25,10,0.25),
add=T)
legend("topleft", c("Group1", "Group2", "Group3")
       , fill=c(rgb(1,0,0,0.5),rgb(0,1,0,0.5),rgb(0,0,1,0.5)) )


#Who is in the Happiest and least happiest Group?
top <- which.max(c(mean(g1.std),mean(g2.std), mean(g3.std))) # which is the
top group
bottom <- which.min(c(mean(g1.std),mean(g2.std), mean(g3.std))) # which is
the top group


happiest.std <- happiness.data[km.std$cluster == top, c(1,4)]
least_happiest.std <- happiness.data[km.std$cluster == bottom, c(1,4)]

#Cluster Size
print(km.std$size)
```

```
print(paste("Standard-Happiest Group is g", top, sep=""))

print(paste("Standard-Least Happiest Group is g", bottom, sep=""))

#Countries with highest and least Happiness score in clusters
head(happiest.std[order(happiest.std$Happiness.Score, decreasing=TRUE), ])

tail(least_happiest.std[order(least_happiest.std$Happiness.Score,
decreasing=TRUE), ])

#Identify who in the top group has lower happiness than the
#median happiness for the entire set of countries

print("Median Happiness Score for Whole Dataset-")
(median(happiness.data$Happiness.Score))

head(least_happiest.std[least_happiest.std$Happiness.Score >
median(happiness.data$Happiness.Score), ])
head(happiest.std[happiest.std$Happiness.Score <
median(happiness.data$Happiness.Score), ])


print("Number of countries above and below Median Value in Happiest Group")

nrow(happiest.std[happiest.std$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(happiest.std[happiest.std$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Least Happiest
Group")

nrow(least_happiest.std[least_happiest.std$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(least_happiest.std[least_happiest.std$Happiness.Score <
median(happiness.data$Happiness.Score), ])

##################################Whitened
DATASET##################################
cluster_range <- 2:15

##Clustering on train and test Raw DataSet
happiness.white.ret <- clustering_euclidean(happiness.white,happiness.data,
cluster_range)

#Optimal Cluster Size is 6
# Create a K-Means cluster with 3 groups based on cols 5:11
# (GDP, Family, Life Expectancy, Freedom, Generosity, Corruption, Dystopia
Residual)
km.white <- clustering_euclidean(happiness.white,happiness.data, 6)

g1.white<-happiness.data[km.white$cluster == 1,]$Happiness.Score
```

```
g2.white<-happiness.data[km.white$cluster == 2,]$Happiness.Score
g3.white<-happiness.data[km.white$cluster == 3,]$Happiness.Score
g4.white<-happiness.data[km.white$cluster == 4,]$Happiness.Score
g5.white<-happiness.data[km.white$cluster == 5,]$Happiness.Score
g6.white<-happiness.data[km.white$cluster == 6,]$Happiness.Score

# plot option "col=rgb(x,x,x,0.5)"" gives fill transparency

par(mfrow=c(1,1))

hist(g1.white, xlim=c(0,10), col=rgb(0.5,0,0,0.5), breaks=seq(0.25,10,0.25),
main = "Histogram of white. Dataset Happiness Score for 3 cluster-groups",
xlab = "Country Happiness Score")
hist(g2.white, xlim=c(0,10), col=rgb(1,0,0,0.5), breaks=seq(0.25,10,0.25),
add=T)
hist(g3.white, xlim=c(0,10), col=rgb(0,0.5,1,0.5), breaks=seq(0.25,10,0.25),
add=T)
hist(g4.white, xlim=c(0,10), col=rgb(0,1,0,0.5), breaks=seq(0.25,10,0.25),
add=T)
hist(g5.white, xlim=c(0,10), col=rgb(0,0,0.5,0.5), breaks=seq(0.25,10,0.25),
add=T)
hist(g6.white, xlim=c(0,10), col=rgb(0,0,1,0.5), breaks=seq(0.25,10,0.25),
add=T)

legend("topleft", c("Group1", "Group2", "Group3","Group4","Group5","Group6")
       ,
fill=c(rgb(0.5,0,0,0.5),rgb(1,0,0,0.5),rgb(0,0.5,0,0.5),rgb(0,1,0,0.5),rgb(0
,0,0.5,0.5),rgb(0,0,1,0.5)) )


#Who is in the Happiest and least happiest Group?
top <- which.max(c(mean(g1.white),mean(g2.white),
mean(g3.white),mean(g4.white),mean(g5.white),mean(g6.white))) # which is the
top group
bottom <- which.min(c(mean(g1.white),mean(g2.white),
mean(g3.white),mean(g4.white),mean(g5.white),mean(g6.white))) # which is the
top group


happiest.white <- happiness.data[km.white$cluster == top, c(1,4)]
least_happiest.white <- happiness.data[km.white$cluster == bottom, c(1,4)]

#Cluster Size
print(km.white$size)

print(paste("Whitened-Happiest Group is g", top, sep=""))

print(paste("Whitened-Least Happiest Group is g", bottom, sep=""))

#Countries with highest and least Happiness score in clusters
head(happiest.white[order(happiest.white$Happiness.Score, decreasing=TRUE),
])
```

```
tail(least_happiest.white[order(least_happiest.white$Happiness.Score,
decreasing=TRUE), ])

#Identify who in the top group has lower happiness than the
#median happiness for the entire set of countries

print("Median Happiness Score for Whole Dataset-")
(median(happiness.data$Happiness.Score))

head(least_happiest.white[least_happiest.white$Happiness.Score >
median(happiness.data$Happiness.Score), ])
head(happiest.white[happiest.white$Happiness.Score <
median(happiness.data$Happiness.Score), ])


print("Number of countries above and below Median Value in Happiest Group")

nrow(happiest.white[happiest.white$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(happiest.white[happiest.white$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Least Happiest
Group")

nrow(least_happiest.white[least_happiest.white$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(least_happiest.white[least_happiest.white$Happiness.Score <
median(happiness.data$Happiness.Score), ])
##################Standard Data Reduction############
happiness.std.new <- happiness.data
happiness.std.new$cluster <- km.std$cluster

happiness.std.new <- happiness.data
happiness.std.new$cluster <- km.std$cluster

med1<-median(happiness.std.new$Happiness.Score[happiness.std.new$cluster ==
1])
med2<-median(happiness.std.new$Happiness.Score[happiness.std.new$cluster ==
2])
med3<-median(happiness.std.new$Happiness.Score[happiness.std.new$cluster ==
3])
print(paste("Median Values-> Cluster1 = ", med1,", Cluster2 = ",med2,",
Cluster3 = ",med3, sep=""))

T1.std.ind <- which(happiness.std.new$cluster == 1)
T2.std.ind <- which(happiness.std.new$cluster == 2)
T3.std.ind <- which(happiness.std.new$cluster == 3)

T1.std.size <- round((2*length(T1.std.ind))/3)
T2.std.size <- round((2*length(T2.std.ind))/3)
```

```
T3.std.size <- round((2*length(T3.std.ind))/3)

T1.std <- get.subset(T1.std.ind,T1.std.size)
T2.std <- get.subset(T2.std.ind,T2.std.size)
T3.std <- get.subset(T3.std.ind,T3.std.size)

happiness.std.reduced.ind <- c(T1.std,T2.std,T3.std)

happiness.std.reduced.data <- happiness.data[happiness.std.reduced.ind,]

head(happiness.std.reduced.data)

##DIvide this Reduced Dataset in Training and  Test Based on Year and Its
proper 2/3 train and
##1/3 Test Ratio
nrow(happiness.std.reduced.data[happiness.std.reduced.data$Year!=2017,])
nrow(happiness.std.reduced.data[happiness.std.reduced.data$Year==2017,])

#Train Dataset with year 2015 and 2016- Total Rows 205
happiness.std.reduced.train <-
happiness.std.reduced.data[happiness.std.reduced.data$Year!=2017,]
#Test Dataset with Year 2017 with Total Rows 108##Setting X axis name

happiness.std.reduced.test <-
happiness.std.reduced.data[happiness.std.reduced.data$Year==2017,]

out.std.reduced = c
(nrow(happiness.std.reduced.data),nrow(happiness.std.reduced.train),
nrow(happiness.std.reduced.test))

x.names=c("Complete","Train","Test")

barplot(out.std.reduced,main="Reduced Std. Data
Spliting",xaxt="n",width=c(1,1,1))
axis(1,at = 1:3,labels=x.names)




################################################################################
############

# UnSupervised Learning

################################################################################
############
################################################################################
############

# UnSupervised Learning
```

```
##########################################################################
###########
##########################################################################
######

## Train Raw data

##########################################################################
######

cluster_range <- 2:15

##Clustering on train and test Raw DataSet
happiness.train.ret <-
clustering_euclidean(happiness.train[5:11],happiness.train, cluster_range)

#Optimal Cluster Size is 3
# Create a K-Means cluster with 3 groups based on cols 5:10
# (GDP, Social Support, Life Expectancy, Freedom, Generosity, Corruption)
km.train <- clustering_euclidean(happiness.train[, 5:11],happiness.train,3)

g1.train<-happiness.train[km.train$cluster == 1,]$Happiness.Score
g2.train<-happiness.train[km.train$cluster == 2,]$Happiness.Score
g3.train<-happiness.train[km.train$cluster == 3,]$Happiness.Score
# plot option "col=rgb(x,x,x,0.5)"" gives fill transparency

par(mfrow=c(1,1))

hist(g1.train, xlim=c(0,10), col=rgb(1,0,0,0.5), breaks=seq(0.25,10,0.25),
main = "Histogram of Train Raw Dataset Happiness Score for 3 cluster-
groups", xlab = "Country Happiness Score")
hist(g2.train, xlim=c(0,10), col=rgb(0,1,0,0.5), breaks=seq(0.25,10,0.25),
add=T)
hist(g3.train, xlim=c(0,10), col=rgb(0,0,1,0.5), breaks=seq(0.25,10,0.25),
add=T)
legend("topleft", c("Group1", "Group2", "Group3")
       , fill=c(rgb(1,0,0,0.5),rgb(0,1,0,0.5),rgb(0,0,1,0.5)) )

#Who is in the Happiest and Least happy Group?
top <- which.max(c(mean(g1.train),mean(g2.train), mean(g3.train))) # which
is the top group
bottom <- which.min(c(mean(g1.train),mean(g2.train), mean(g3.train))) #
which is the top group

happiest.train <- happiness.train[km.train$cluster == top, c(1,4)]
least_happiest.train <- happiness.train[km.train$cluster == bottom, c(1,4)]
#Cluster Size
print(km.train$size)

print(paste("RAW-Train Happiest Group is g", top, sep=""))

print(paste("RAW-Train-Least Happiest Group is g", bottom, sep=""))
```

```
head(happiest.train[order(happiest.train$Happiness.Score, decreasing=TRUE),
])

tail(least_happiest.train[order(least_happiest.train$Happiness.Score,
decreasing=TRUE), ])

#Identify who in the top group has lower happiness than the
#median happiness for the entire set of countries

print("Median Happiness Score for training Dataset-")
(median(happiness.train$Happiness.Score))

head(least_happiest.train[least_happiest.train$Happiness.Score >
median(happiness.train$Happiness.Score), ])
head(happiest.train[happiest.train$Happiness.Score <
median(happiness.train$Happiness.Score), ])

print("Number of countries above and below Median Value in Happiest Group")

nrow(happiest.train[happiest.train$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(happiest.train[happiest.train$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Least Happiest
Group")

nrow(least_happiest.train[least_happiest.train$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(least_happiest.train[least_happiest.train$Happiness.Score <
median(happiness.data$Happiness.Score), ])


###########TEST Raw Dataset#######################################

#Optimal Cluster Size is 3
# Create a K-Means cluster with 3 groups based on cols 5:10
# (GDP, Social Support, Life Expectancy, Freedom, Generosity, Corruption)
km.test <- clustering_euclidean(happiness.test[, 5:11],happiness.test,3)

g1.test<-happiness.test[km.test$cluster == 1,]$Happiness.Score
g2.test<-happiness.test[km.test$cluster == 2,]$Happiness.Score
g3.test<-happiness.test[km.test$cluster == 3,]$Happiness.Score
# plot option "col=rgb(x,x,x,0.5)"" gives fill transparency

par(mfrow=c(1,1))

hist(g1.test, xlim=c(0,10), col=rgb(1,0,0,0.5), breaks=seq(0.25,10,0.25),
main = "Histogram of Test Raw Dataset Happiness Score for 3 cluster-groups",
xlab = "Country Happiness Score")
```

```
hist(g2.test, xlim=c(0,10), col=rgb(0,1,0,0.5), breaks=seq(0.25,10,0.25),
add=T)
hist(g3.test, xlim=c(0,10), col=rgb(0,0,1,0.5), breaks=seq(0.25,10,0.25),
add=T)
legend("topleft", c("Group1", "Group2", "Group3")
       , fill=c(rgb(1,0,0,0.5),rgb(0,1,0,0.5),rgb(0,0,1,0.5)) )

#Who is in the Happiest Group?
top <- which.max(c(mean(g1.test),mean(g2.test), mean(g3.test))) # which is
the top group
bottom <- which.min(c(mean(g1.test),mean(g2.test), mean(g3.test))) # which
is the top group

happiest.test <- happiness.test[km.test$cluster == top, c(1,4)]
least_happiest.test <- happiness.test[km.test$cluster == bottom, c(1,4)]


#Cluster Size
print(km.test$size)

print(paste("RAW-Test Happiest Group is g", top, sep=""))

print(paste("RAW-Test Least Happiest Group is g", bottom, sep=""))

head(happiest.test[order(happiest.test$Happiness.Score, decreasing=TRUE), ])

tail(least_happiest.test[order(least_happiest.test$Happiness.Score,
decreasing=TRUE), ])

#Identify who in the top group has lower happiness than the
#median happiness for the entire set of countries

print("Median Happiness Score for testing Dataset-")
(median(happiness.test$Happiness.Score))

head(least_happiest.test[least_happiest.test$Happiness.Score >
median(happiness.test$Happiness.Score), ])
head(happiest.test[happiest.test$Happiness.Score <
median(happiness.test$Happiness.Score), ])

print("Number of countries above and below Median Value in Happiest Group")

nrow(happiest.test[happiest.test$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(happiest.test[happiest.test$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Least Happiest
Group")

nrow(least_happiest.test[least_happiest.test$Happiness.Score >
median(happiness.data$Happiness.Score), ])
```

```
nrow(least_happiest.test[least_happiest.test$Happiness.Score <
median(happiness.data$Happiness.Score), ])

##############################PCA#############################################
###########

cluster_range <- 2:15
##Clustering on train and test Raw pcSet
happiness.pc.ret <- clustering_euclidean(happiness.pc,happiness.data,
cluster_range)

#Optimal Cluster Size is 3
# Create a K-Means cluster with 3 groups based on cols 5:10
# (GDP, Family, Life Expectancy, Freedom, Generosity, Corruption, Dystopia
Residual)
km.pc <- clustering_euclidean(happiness.pc,happiness.data, 3)

g1.pc<-happiness.data[km.pc$cluster == 1,]$Happiness.Score
g2.pc<-happiness.data[km.pc$cluster == 2,]$Happiness.Score
g3.pc<-happiness.data[km.pc$cluster == 3,]$Happiness.Score
# plot option "col=rgb(x,x,x,0.5)"" gives fill transparency

par(mfrow=c(1,1))

hist(g1.pc, xlim=c(0,10), col=rgb(1,0,0,0.5), breaks=seq(0.25,10,0.25), main
= "Histogram of PCA Whole Dataset Happiness Score for 3 cluster-groups",
xlab = "Country Happiness Score")
hist(g2.pc, xlim=c(0,10), col=rgb(0,1,0,0.5), breaks=seq(0.25,10,0.25),
add=T)
hist(g3.pc, xlim=c(0,10), col=rgb(0,0,1,0.5), breaks=seq(0.25,10,0.25),
add=T)
legend("topleft", c("Group1", "Group2", "Group3")
       , fill=c(rgb(1,0,0,0.5),rgb(0,1,0,0.5),rgb(0,0,1,0.5)) )

#Cluster Size
print(km.pc$size)

#Who is in the Happiest Group?
top <- which.max(c(mean(g1.pc),mean(g2.pc), mean(g3.pc))) # which is the top
group
bottom <- which.min(c(mean(g1.pc),mean(g2.pc), mean(g3.pc))) # which is the
top group

happiest.pc <- happiness.data[km.pc$cluster == top, c(1,4)]
least_happiest.pc <- happiness.data[km.pc$cluster == bottom, c(1,4)]

print(paste("PCA-Happiest Group is g", top, sep=""))

print(paste("PCA-Least Happiest Group is g", bottom, sep=""))

head(happiest.pc[order(happiest.pc$Happiness.Score, decreasing=TRUE), ])
```

```
tail(least_happiest.pc[order(least_happiest.pc$Happiness.Score,
decreasing=TRUE), ])
#Identify who in the top group has lower happiness than the
#median happiness for the entire set of countries
print("Median Happiness Score for Whole Dataset-")
(median(happiness.data$Happiness.Score))

head(least_happiest.pc[least_happiest.pc$Happiness.Score >
median(happiness.data$Happiness.Score), ])
head(happiest.pc[happiest.pc$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Happiest Group")

nrow(happiest.pc[happiest.pc$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(happiest.pc[happiest.pc$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Least Happiest
Group")

nrow(least_happiest.pc[least_happiest.pc$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(least_happiest.pc[least_happiest.pc$Happiness.Score <
median(happiness.data$Happiness.Score), ])


######################Train Dataset-After PCA####################


cluster_range <- 2:15
##Clustering on train and test Raw pcSet
happiest.train.pc.ret <-
clustering_euclidean(happiness.train.pc,happiness.train, cluster_range)

#Optimal Cluster Size is 3
# Create a K-Means cluster with 3 groups based on cols 5:10
# (GDP, Family, Life Expectancy, Freedom, Generosity, Corruption,Dystopia
Residual)
km.train.pc <- clustering_euclidean(happiness.train.pc,happiness.train, 3)

g1.train.pc<-happiness.train[km.train.pc$cluster == 1,]$Happiness.Score
g2.train.pc<-happiness.train[km.train.pc$cluster == 2,]$Happiness.Score
g3.train.pc<-happiness.train[km.train.pc$cluster == 3,]$Happiness.Score
# plot option "col=rgb(x,x,x,0.5)"" gives fill transparency

par(mfrow=c(1,1))

hist(g1.train.pc, xlim=c(0,10), col=rgb(1,0,0,0.5),
breaks=seq(0.25,10,0.25), main = "Histogram of Train PCA-Happiness Score for
3 cluster-groups", xlab = "Country Happiness Score")
```

```
hist(g2.train.pc, xlim=c(0,10), col=rgb(0,1,0,0.5),
breaks=seq(0.25,10,0.25), add=T)
hist(g3.train.pc, xlim=c(0,10), col=rgb(0,0,1,0.5),
breaks=seq(0.25,10,0.25), add=T)
legend("topleft", c("Group1", "Group2", "Group3")
       , fill=c(rgb(1,0,0,0.5),rgb(0,1,0,0.5),rgb(0,0,1,0.5)) )

#Who is in the Happiest and Least Happiest group Group?
top <- which.max(c(mean(g1.train.pc),mean(g2.train.pc), mean(g3.train.pc)))
# which is the top group
bottom <- which.min(c(mean(g1.train.pc),mean(g2.train.pc),
mean(g3.train.pc))) # which is the top group

happiest.train.pc <- happiness.train[km.train.pc$cluster == top, c(1,4)]
least_happiest.train.pc <- happiness.train[km.train.pc$cluster == bottom,
c(1,4)]

#Cluster Size
print(km.train.pc$size)

print(paste("Train-PCA-Happiest Group is g", top, sep=""))

print(paste("Train-PCA-Least Happiest Group is g", bottom, sep=""))


head(happiest.train.pc[order(happiest.train.pc$Happiness.Score,
decreasing=TRUE), ])

tail(least_happiest.train.pc[order(least_happiest.train.pc$Happiness.Score,
decreasing=TRUE), ])
#Identify who in the top group has lower happiness than the
#median happiness for the entire set of countries
print("Median Happiness Score for testing Dataset-")
(median(happiness.train$Happiness.Score))

head(least_happiest.train.pc[least_happiest.train.pc$Happiness.Score >
median(happiness.train$Happiness.Score), ])
head(happiest.train.pc[happiest.train.pc$Happiness.Score <
median(happiness.train$Happiness.Score), ])


print("Number of countries above and below Median Value in Happiest Group")

nrow(happiest.train.pc[happiest.train.pc$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(happiest.train.pc[happiest.train.pc$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Least Happiest
Group")
```

```
nrow(least_happiest.train.pc[least_happiest.train.pc$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(least_happiest.train.pc[least_happiest.train.pc$Happiness.Score <
median(happiness.data$Happiness.Score), ])

##########TEST DATASET AFTER PCA#############################################

#Optimal Cluster Size is 3
# Create a K-Means cluster with 3 groups based on cols 5:10
# (GDP, Social Support, Life Expectancy, Freedom, Generosity, Corruption)
km.test.pc <- clustering_euclidean(happiness.test.pc,happiness.test,3)

g1.test.pc<-happiness.test[km.test.pc$cluster == 1,]$Happiness.Score
g2.test.pc<-happiness.test[km.test.pc$cluster == 2,]$Happiness.Score
g3.test.pc<-happiness.test[km.test.pc$cluster == 3,]$Happiness.Score
# plot option "col=rgb(x,x,x,0.5)"" gives fill transparency

par(mfrow=c(1,1))

hist(g1.test.pc, xlim=c(0,10), col=rgb(1,0,0,0.5), breaks=seq(0.25,10,0.25),
main = "Histogram of Test PCA-Happiness Score for 3 cluster-groups", xlab =
"Country Happiness Score")
hist(g2.test.pc, xlim=c(0,10), col=rgb(0,1,0,0.5), breaks=seq(0.25,10,0.25),
add=T)
hist(g3.test.pc, xlim=c(0,10), col=rgb(0,0,1,0.5), breaks=seq(0.25,10,0.25),
add=T)
legend("topleft", c("Group1", "Group2", "Group3")
       , fill=c(rgb(1,0,0,0.5),rgb(0,1,0,0.5),rgb(0,0,1,0.5)) )

#Who is in the Happiest and least Happiest Group?
top <- which.max(c(mean(g1.test.pc),mean(g2.test.pc), mean(g3.test.pc))) #
which is the top group
bottom <- which.min(c(mean(g1.test.pc),mean(g2.test.pc), mean(g3.test.pc)))
# which is the top group

happiest.test.pc <- happiness.test[km.test.pc$cluster == top, c(1,4)]
least_happiest.test.pc <- happiness.test[km.test.pc$cluster == bottom,
c(1,4)]

#Cluster Size
print(km.test.pc$size)

print(paste("Test-PCA-Happiest Group is g", top, sep=""))

print(paste("Test-PCA-Least Happiest Group is g", bottom, sep=""))

head(happiest.test.pc[order(happiest.test.pc$Happiness.Score,
decreasing=TRUE), ])

tail(least_happiest.test.pc[order(least_happiest.test.pc$Happiness.Score,
decreasing=TRUE), ])
#Identify who in the top group has lower happiness than the
```

```
#median happiness for the entire set of countries
print("Median Happiness Score for testing Dataset-")
(median(happiness.test$Happiness.Score))

head(least_happiest.test.pc[least_happiest.test.pc$Happiness.Score >
median(happiness.test$Happiness.Score), ])
head(happiest.test.pc[happiest.test.pc$Happiness.Score <
median(happiness.test$Happiness.Score), ])

print("Number of countries above and below Median Value in Happiest Group")

nrow(happiest.test.pc[happiest.test.pc$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(happiest.test.pc[happiest.test.pc$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Least Happiest
Group")

nrow(least_happiest.test.pc[least_happiest.test.pc$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(least_happiest.test.pc[least_happiest.test.pc$Happiness.Score <
median(happiness.data$Happiness.Score), ])

#####################################ICA#########################################
################

cluster_range <- 2:15
##Clustering on train and test Raw icaSet
happiest.ica.ret <- clustering_euclidean(happiness.ica,happiness.data,
cluster_range)

#Optimal Cluster Size is 4
# Create a K-Means cluster with 4 groups based on cols 5:10
# (GDP, Family, Life Expectancy, Freedom, Generosity, Corruption,Dystopia
Residual)
km.ica <- clustering_euclidean(happiness.ica,happiness.data, 4)

g1.ica<-happiness.data[km.ica$cluster == 1,]$Happiness.Score
g2.ica<-happiness.data[km.ica$cluster == 2,]$Happiness.Score
g3.ica<-happiness.data[km.ica$cluster == 3,]$Happiness.Score
g4.ica<-happiness.data[km.ica$cluster == 4,]$Happiness.Score

# plot option "col=rgb(x,x,x,0.5)"" gives fill transparency

par(mfrow=c(1,1))

hist(g1.ica, xlim=c(0,10), col=rgb(1,0,0,0.5), breaks=seq(0.25,10,0.25),
main = "Histogram of ICA Whole Dataset Happiness Score for 3 cluster-
groups", xlab = "Country Happiness Score")
hist(g2.ica, xlim=c(0,10), col=rgb(0,1,0,0.5), breaks=seq(0.25,10,0.25),
add=T)
```

```
hist(g3.ica, xlim=c(0,10), col=rgb(0,0,1,0.5), breaks=seq(0.25,10,0.25),
add=T)
hist(g4.ica, xlim=c(0,10), col=rgb(1,0,1,0.5), breaks=seq(0.25,10,0.25),
add=T)
legend("topleft", c("Group1", "Group2", "Group3","Group4")
       , fill=c(rgb(1,0,0,0.5),rgb(0,1,0,0.5),rgb(0,0,1,0.5),rgb(1,0,1,0.5))
)


#Who is in the Happiest and least happiest Group?
top <- which.max(c(mean(g1.ica),mean(g2.ica), mean(g3.ica),mean(g4.ica))) #
which is the top group
bottom <- which.min(c(mean(g1.ica),mean(g2.ica), mean(g3.ica),mean(g4.ica)))
# which is the top group

happiest.ica <- happiness.data[km.ica$cluster == top, c(1,4)]
least_happiest.ica <- happiness.data[km.ica$cluster == bottom, c(1,4)]

#Cluster Size
print(km.ica$size)

print(paste("ICA-Happiest Group is g", top, sep=""))

print(paste("ICA-Least Happiest Group is g", bottom, sep=""))

head(happiest.ica[order(happiest.ica$Happiness.Score, decreasing=TRUE), ])

tail(least_happiest.ica[order(least_happiest.ica$Happiness.Score,
decreasing=TRUE), ])
#Identify who in the top group has lower happiness than the
#median happiness for the entire set of countries
print("Median Happiness Score for Whole Dataset-")
(median(happiness.data$Happiness.Score))

head(least_happiest.ica[least_happiest.ica$Happiness.Score >
median(happiness.data$Happiness.Score), ])
head(happiest.ica[happiest.ica$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Happiest Group")

nrow(happiest.ica[happiest.ica$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(happiest.ica[happiest.ica$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Least Happiest
Group")

nrow(least_happiest.ica[least_happiest.ica$Happiness.Score >
median(happiness.data$Happiness.Score), ])
```

```
nrow(least_happiest.ica[least_happiest.ica$Happiness.Score <
median(happiness.data$Happiness.Score), ])


###############Train Dataset After ICA###############
#####################################ICA#########################################
################

cluster_range <- 2:15
##Clustering on train and test Raw icaSet
happiness.train.ica.ret <-
clustering_euclidean(happiness.train.ica,happiness.train, cluster_range)

#Optimal Cluster Size is 4
# Create a K-Means cluster with 4 groups based on cols 5:10
# (GDP, Social Support, Life Expectancy, Freedom, Generosity, Corruption)
km.train.ica <- clustering_euclidean(happiness.train.ica,happiness.train, 4)

g1.train.ica<-happiness.train[km.train.ica$cluster == 1,]$Happiness.Score
g2.train.ica<-happiness.train[km.train.ica$cluster == 2,]$Happiness.Score
g3.train.ica<-happiness.train[km.train.ica$cluster == 3,]$Happiness.Score
g4.train.ica<-happiness.train[km.train.ica$cluster == 4,]$Happiness.Score

# plot option "col=rgb(x,x,x,0.5)"" gives fill transparency

par(mfrow=c(1,1))

hist(g1.train.ica, xlim=c(0,10), col=rgb(1,0,0,0.5),
breaks=seq(0.25,10,0.25), main = "Histogram of Train ICA Dataset Happiness
Score for 3 cluster-groups", xlab = "Country Happiness Score")
hist(g2.train.ica, xlim=c(0,10), col=rgb(0,1,0,0.5),
breaks=seq(0.25,10,0.25), add=T)
hist(g3.train.ica, xlim=c(0,10), col=rgb(0,0,1,0.5),
breaks=seq(0.25,10,0.25), add=T)
hist(g4.train.ica, xlim=c(0,10), col=rgb(1,0,1,0.5),
breaks=seq(0.25,10,0.25), add=T)
legend("topleft", c("Group1", "Group2", "Group3","Group4")
       , fill=c(rgb(1,0,0,0.5),rgb(0,1,0,0.5),rgb(0,0,1,0.5),rgb(1,0,1,0.5))
)

#Who is in the Happiest Group?
top <- which.max(c(mean(g1.train.ica),mean(g2.train.ica),
mean(g3.train.ica),mean(g4.train.ica))) # which is the top group
bottom <- which.min(c(mean(g1.train.ica),mean(g2.train.ica),
mean(g3.train.ica),mean(g4.train.ica))) # which is the top group

happiest.train.ica <- happiness.train[km.train.ica$cluster == top, c(1,4)]
least_happiest.train.ica <- happiness.train[km.train.ica$cluster == bottom,
c(1,4)]

#Cluster Size
print(km.train.ica$size)
```

```
print(paste("Train-ICA-Happiest Group is g", top, sep=""))

print(paste("Train-ICA-Least Happiest Group is g", bottom, sep=""))

head(happiest.train.ica[order(happiest.train.ica$Happiness.Score,
decreasing=TRUE), ])

tail(least_happiest.train.ica[order(least_happiest.train.ica$Happiness.Score
, decreasing=TRUE), ])
#Identify who in the top group has lower happiness than the
#median happiness for the entire set of countries
print("Median Happiness Score for train Dataset-")
(median(happiness.train$Happiness.Score))

head(least_happiest.train.ica[least_happiest.train.ica$Happiness.Score >
median(happiness.train$Happiness.Score), ])
head(happiest.train.ica[happiest.train.ica$Happiness.Score <
median(happiness.train$Happiness.Score), ])

print("Number of countries above and below Median Value in Happiest Group")

nrow(happiest.train.ica[happiest.train.ica$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(happiest.train.ica[happiest.train.ica$Happiness.Score <
median(happiness.data$Happiness.Score), ])

print("Number of countries above and below Median Value in Least Happiest
Group")

nrow(least_happiest.train.ica[least_happiest.train.ica$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(least_happiest.train.ica[least_happiest.train.ica$Happiness.Score <
median(happiness.data$Happiness.Score), ])


########TEST DATASET AFTER ICA#################################


#Optimal Cluster Size is 4
# Create a K-Means cluster with 4 groups based on cols 5:10
# (GDP, Social Support, Life Expectancy, Freedom, Generosity, Corruption)
km.test.ica <- clustering_euclidean(happiness.test.ica, happiness.test, 4)

g1.test.ica<-happiness.test[km.test.ica$cluster == 1,]$Happiness.Score
g2.test.ica<-happiness.test[km.test.ica$cluster == 2,]$Happiness.Score
g3.test.ica<-happiness.test[km.test.ica$cluster == 3,]$Happiness.Score
g4.test.ica<-happiness.test[km.test.ica$cluster == 4,]$Happiness.Score

# plot option "col=rgb(x,x,x,0.5)"" gives fill transparency

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

```
hist(g1.test.ica, xlim=c(0,10), col=rgb(1,0,0,0.5),
breaks=seq(0.25,10,0.25), main = "Histogram of Test ICA Dataset Happiness
Score for 3 cluster-groups", xlab = "Country Happiness Score")
hist(g2.test.ica, xlim=c(0,10), col=rgb(0,1,0,0.5),
breaks=seq(0.25,10,0.25), add=T)
hist(g3.test.ica, xlim=c(0,10), col=rgb(0,0,1,0.5),
breaks=seq(0.25,10,0.25), add=T)
hist(g4.test.ica, xlim=c(0,10), col=rgb(1,0,1,0.5),
breaks=seq(0.25,10,0.25), add=T)
legend("topleft", c("Group1", "Group2", "Group3","Group4")
       , fill=c(rgb(1,0,0,0.5),rgb(0,1,0,0.5),rgb(0,0,1,0.5),rgb(1,0,1,0.5))
)

#Who is in the Happiest Group?
top <- which.max(c(mean(g1.test.ica),mean(g2.test.ica),
mean(g3.test.ica),mean(g4.test.ica))) # which is the top group
bottom <- which.min(c(mean(g1.test.ica),mean(g2.test.ica),
mean(g3.test.ica),mean(g4.test.ica))) # which is the top group

happiest.test.ica <- happiness.test[km.test.ica$cluster == top, c(1,4)]
least_happiest.test.ica <- happiness.test[km.test.ica$cluster == bottom,
c(1,4)]

#Cluster Size
print(km.test.ica$size)

print(paste("Test-ICA-Happiest Group is g", top, sep=""))

print(paste("Test-ICA-Least Happiest Group is g", bottom, sep=""))

head(happiest.test.ica[order(happiest.test.ica$Happiness.Score,
decreasing=TRUE), ])

tail(least_happiest.test.ica[order(least_happiest.test.ica$Happiness.Score,
decreasing=TRUE), ])
#Identify who in the top group has lower happiness than the
#median happiness for the entire set of countries
print("Median Happiness Score for testing Dataset-")
(median(happiness.test$Happiness.Score))

head(least_happiest.test.ica[least_happiest.test.ica$Happiness.Score >
median(happiness.test$Happiness.Score), ])
head(happiest.test.ica[happiest.test.ica$Happiness.Score <
median(happiness.test$Happiness.Score), ])

print("Number of countries above and below Median Value in Happiest Group")

nrow(happiest.test.ica[happiest.test.ica$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(happiest.test.ica[happiest.test.ica$Happiness.Score <
median(happiness.data$Happiness.Score), ])
```

```
print("Number of countries above and below Median Value in Least Happiest
Group")

nrow(least_happiest.test.ica[least_happiest.test.ica$Happiness.Score >
median(happiness.data$Happiness.Score), ])
nrow(least_happiest.test.ica[least_happiest.test.ica$Happiness.Score <
median(happiness.data$Happiness.Score), ])



############################Supervised
Learning#####################################
############Classifications rparts#######
library(rpart)
library(caret)
library(rpart.plot)
library(MASS)
library(DAAG)
library(tree)

####################TREE################

library(tree)

region.tree <-
tree(Region~Happiness.Rank+Happiness.Score+Economy..GDP.per.Capita.+

Family+Health..Life.Expectancy.+Freedom+Trust..Government.Corruption.
                     +Generosity+Dystopia.Residual ,method="class",data =
happiness.std.reduced.train)

plot(region.tree, type = "uniform")
text(region.tree, all= T, cex=0.7)

########Prediction and Accuracy#########
model.2<-region.tree
test<-happiness.std.reduced.test
test_pred <- predict(model.2,test)
head(test_pred)
head(test)


###############Score tree############
score.tree <-
tree(Happiness.Score~Happiness.Rank+Region+Economy..GDP.per.Capita.+

Family+Health..Life.Expectancy.+Freedom+Trust..Government.Corruption.
                    +Generosity+Dystopia.Residual ,method="anova",data =
happiness.std.reduced.train)
```

```
(score.tree)

plot(score.tree, type = "uniform")
text(score.tree, all= T, cex=0.9)
##########Residual Analysis#########
plot(residuals(score.tree), ylab="residuals")
r<-(residuals(score.tree))
sum(r^2)



########Prediction and Accuracy#########
model.2<-score.tree
test_pred <- predict(model.2,test)
Actual.Values<-round(test$Happiness.Score)
r.pred<-floor(test_pred)
Predicted.Values<-as.numeric(r.pred)
table(Predicted.Values,Actual.Values)
confusion(Predicted.Values, Actual.Values)






############# Score tree without rank################

score.tree2 <- tree(Happiness.Score~Region+Economy..GDP.per.Capita.+

Family+Health..Life.Expectancy.+Freedom+Trust..Government.Corruption.
                    +Generosity+Dystopia.Residual ,method="anova",data =
happiness.std.reduced.train)



(score.tree2)

plot(score.tree2, type = "uniform")
text(score.tree2, all= T, cex=0.5)



##########Residual Analysis#########
plot(residuals(score.tree2), ylab="residuals")
r<-(residuals(score.tree2))
sum(r^2)



########Prediction and Accuracy#########
model.2<-score.tree2
test_pred <- predict(model.2,test)
```

```
Actual.Values<-round(test$Happiness.Score)
r.pred<-floor(test_pred)
Predicted.Values<-as.numeric(r.pred)
table(Predicted.Values,Actual.Values)
confusion(Predicted.Values, Actual.Values)



##########Applying rpart####################
region.rp.std <-
rpart(Region~Happiness.Rank+Happiness.Score+Economy..GDP.per.Capita.+

Family+Health..Life.Expectancy.+Freedom+Trust..Government.Corruption.
                        +Generosity+Dystopia.Residual ,method="class",data =
happiness.std.reduced.train)



########Applying rpart############
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(123)
dtree_fit <- train(Happiness.Score ~., data = happiness.std.reduced.train,
method = "rpart",
                   parms = list(split = "information"),
                   trControl=trctrl,
                   tuneLength = 10)



##########Plotting rpart############
prp(dtree_fit$finalModel, box.palette = "Blue", tweak = 1.2, varlen=0)


##########Residual Analysis#########
plot(residuals(dtree_fit), ylab="residuals")
r<-(residuals(dtree_fit))
sum(r^2)

########Prediction and Accuracy#########

model<-dtree_fit
test<-happiness.std.reduced.test
test_pred <- predict(model,test)
Actual.Values<-round(happiness.std.reduced.test$Happiness.Score)
r.pred<-round(test_pred)
Predicted.Values<-as.numeric(r.pred)
table(Predicted.Values,Actual.Values)
confusion(Predicted.Values, Actual.Values)
```

```
##########Analysis without rank##########
########Applying rpart############
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(123)
dtree_fit.2 <- train(Happiness.Score ~Economy..GDP.per.Capita.+

Family+Health..Life.Expectancy.+Freedom+Trust..Government.Corruption.
                     +Generosity, data = happiness.std.reduced.train, method
= "rpart",
                     parms = list(split = "information"),
                     trControl=trctrl,
                     tuneLength = 10)

##########Plotting rpart############
prp(dtree_fit.2$finalModel, box.palette = "Green", tweak = 1.2, varlen=0)




##########Residual Analysis#########
plot(residuals(dtree_fit.2), ylab="residuals")
r<-(residuals(dtree_fit.2))
sum(r^2)


########Prediction and Accuracy#########
model.2<-dtree_fit.2
test<-happiness.std.reduced.test
test_pred <- predict(model.2,test)
Actual.Values<-round(test$Happiness.Score)
r.pred<-floor(test_pred)
Predicted.Values<-as.numeric(r.pred)
table(Predicted.Values,Actual.Values)
confusion(Predicted.Values, Actual.Values)


####################################END#################################################
```