EXPLORING WORLD HAPPINESS OF YEARS 2015,2016,2017

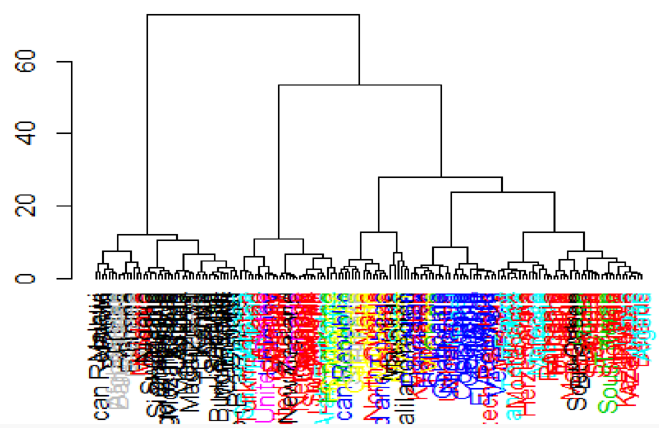
Vignesh J Muralidharan

September 29, 2018

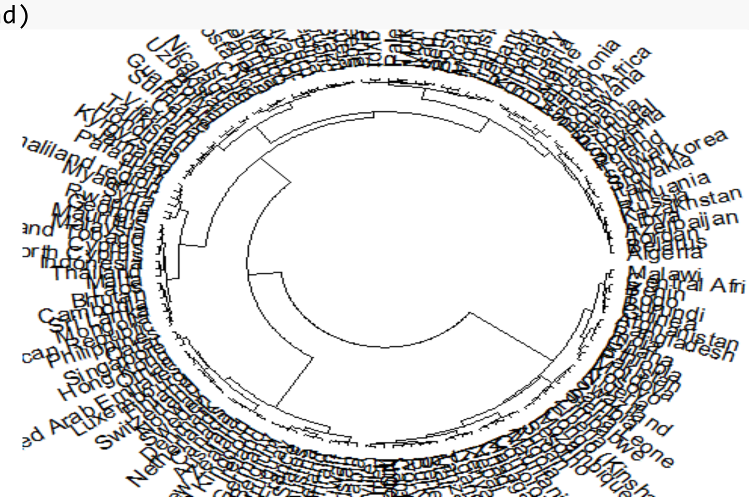
```
library(dendroextras); library(dendextend) ; library(cluster)
library(tidyverse) ; library(circlize) ; library(mclust)
library(factoextra) ; library(MVA) ; library(NbClust) ; library(seriation)
library(arulesCBA); library(arulesViz)
```

FOR DATASET HAPPY 2015

```
happy2015=read.csv("https://raw.githubusercontent.com/vigneshjmurali/Statistical-Predictive-Modelling/master/Datasets/World_Happiness_2015.csv")
dim(happy2015)
## [1] 158  12
# TAKING OUT HAPPINESS INFORMATION FROM THE GIVEN DATASET FOR THE CLUSTERING ANALYSIS
row.names(happy2015)<-happy2015$Country
happy2015cut<-happy2015[,6:12]
happy2015cut.s=scale(happy2015cut)
happy2015cut.d=dist(happy2015cut.s)
happy2015cut.hc.s=hclust(happy2015cut.d,method="ward.D")
happy2015cutdend=as.dendrogram(happy2015cut.hc.s)
labels_colors(happy2015cutdend)<-as.numeric(as.factor(happy2015$Region[happy2015cut.hc.s$order]))
dend=as.dendrogram(happy2015cut.hc.s)
plot(happy2015cutdend)
```

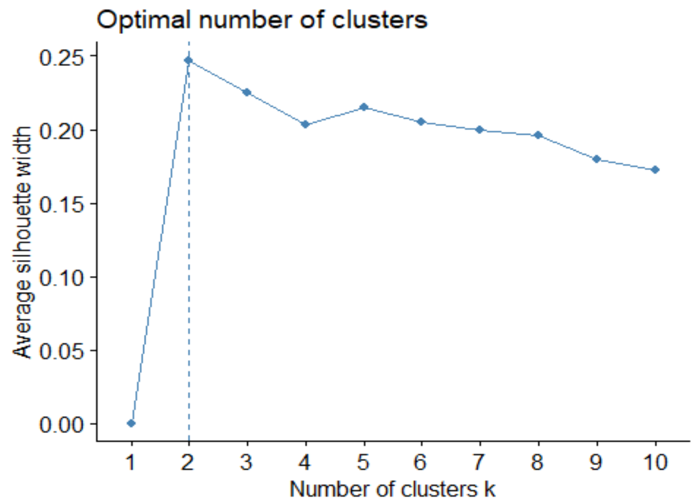



```
par(mar = rep(0,4))
dend=as.dendrogram(happy2015cut.hc.s)
circlize_dendrogram(dend)
```

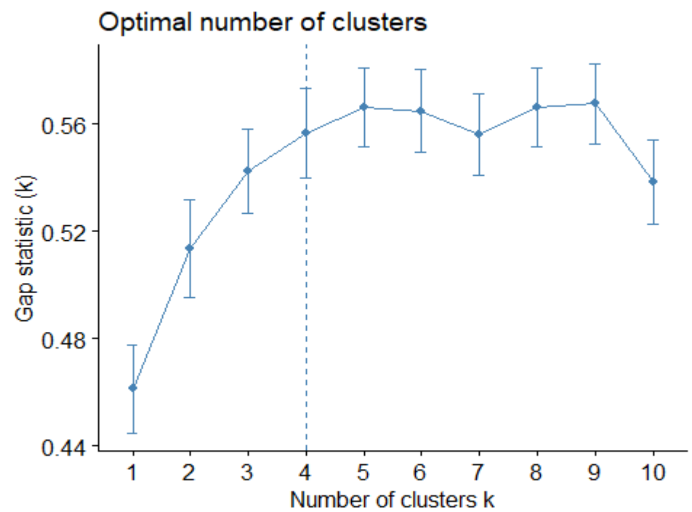


PARTITION CLUSTERING HAPPY 2015

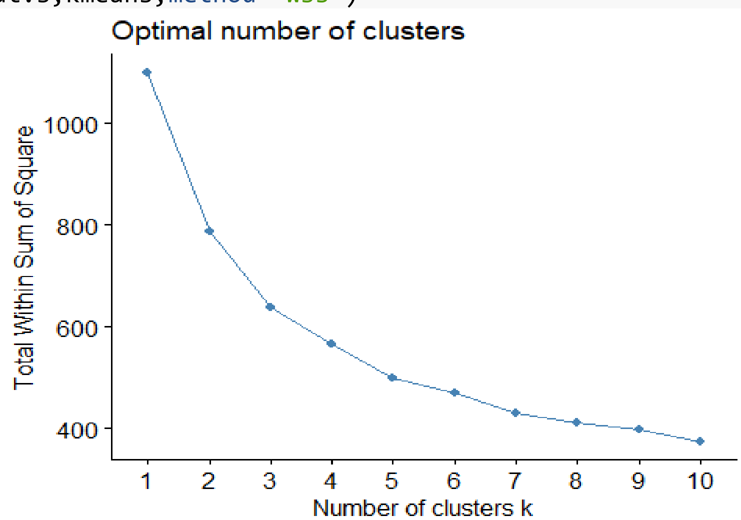
```
set.seed(123)
fviz_nbclust(happy2015cut.s,kmeans,method="silhouette")
```



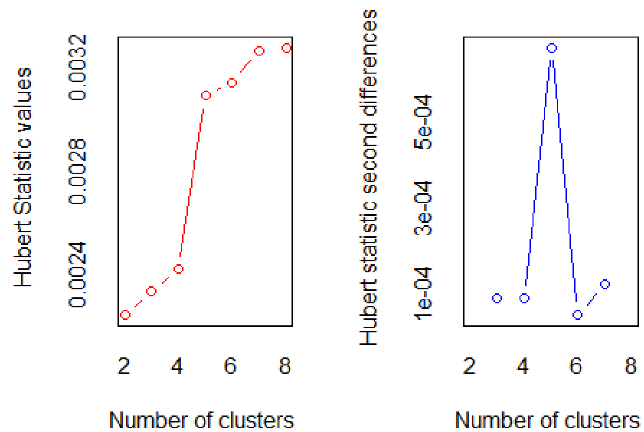

```
fviz_nbclust(happy2015cut.s,kmeans,method="gap_stat")
```



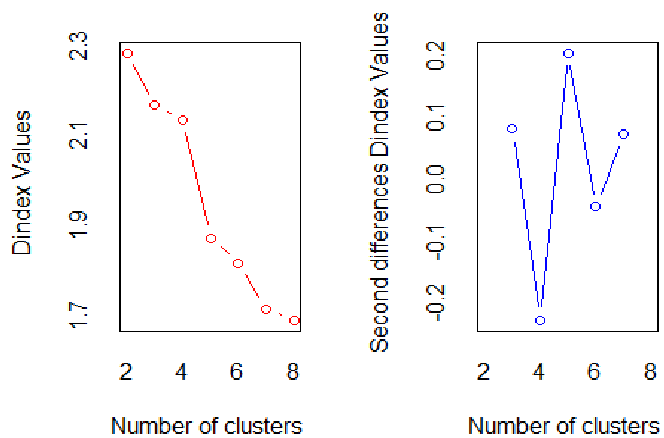

```
fviz_nbclust(happy2015cut.s,kmeans,method="wss")
```



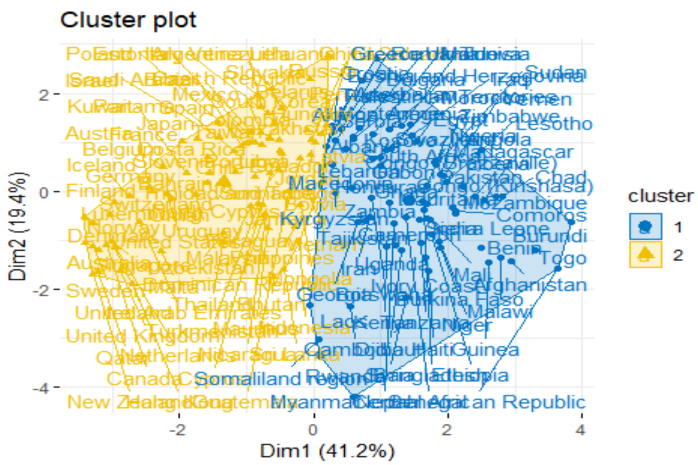

```
happy15.nbclust<-happy2015cut %>% #Using NbClust
  scale() %>% NbClust(distance="euclidean",min.nc=2,max.nc=8,method="complete",index="all")
```



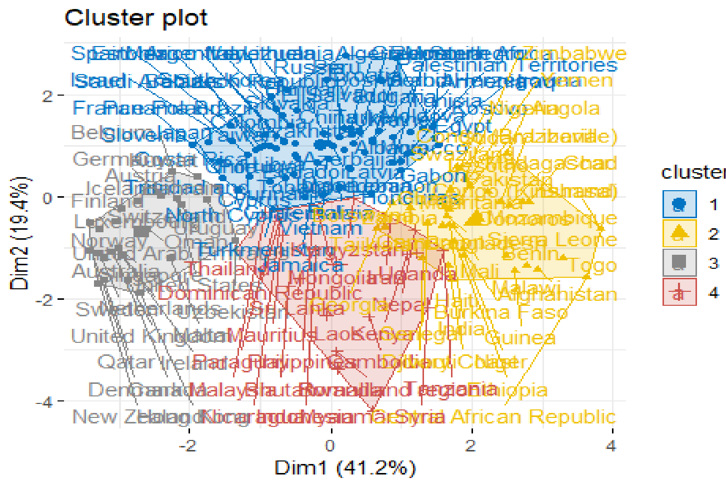

```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##            significant increase of the value of the measure i.e the significant peak in Hubert
##            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




```
## *** : The D index is a graphical method of determining the number of clusters.
##             In the plot of D index, we seek a significant knee (the significant peak in Dindex
##             second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
## *******************************************************************
## * Among all indices:
## * 5 proposed 2 as the best number of clusters
## * 3 proposed 3 as the best number of clusters
## * 2 proposed 4 as the best number of clusters
## * 10 proposed 5 as the best number of clusters
## * 1 proposed 7 as the best number of clusters
## * 2 proposed 8 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  5
## *******************************************************************
happy2015cut.k2sil<-kmeans(happy2015cut.s,centers=2,iter.max=100,nstart=25)
happy2015cut.k4gap<-kmeans(happy2015cut.s,centers=4,iter.max=100,nstart=25)
fviz_cluster(happy2015cut.k2sil,data=happy2015cut.s,ellipse.type="convex",palette="jco",repel=TRUE
,ggtheme=theme_minimal())
```

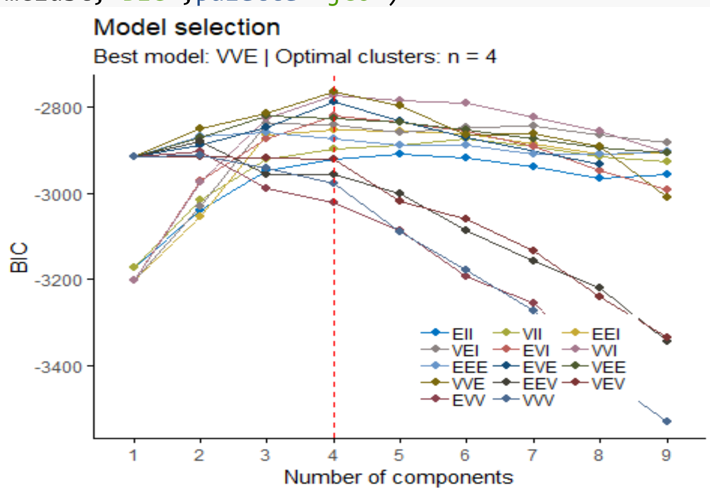



```
fviz_cluster(happy2015cut.k4gap,data=happy2015cut.s,ellipse.type="convex",palette="jco",repel=TRUE,ggtheme=theme_minimal())
```



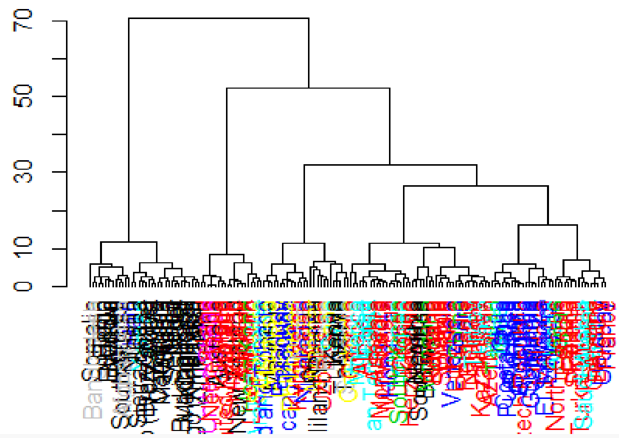
this clustering clearly show that most of the african countries and some of the asian countries are clustered in the yellow and most of the europe countries are clustered in the black. #**M-CLUST**

```
happy2015cut.mclust<-Mclust(happy2015cut.s) ; summary(happy2015cut.mclust)
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
## Mclust VVE (ellipsoidal, equal orientation) model with 4 components:
##  log.likelihood   n df       BIC       ICL
##       -1179.659 158 80 -2764.326 -2776.977
## Clustering table:
##  1  2  3  4
## 19 34 65 40
fviz_mclust(happy2015cut.mclust,"BIC",palette="jco")
```

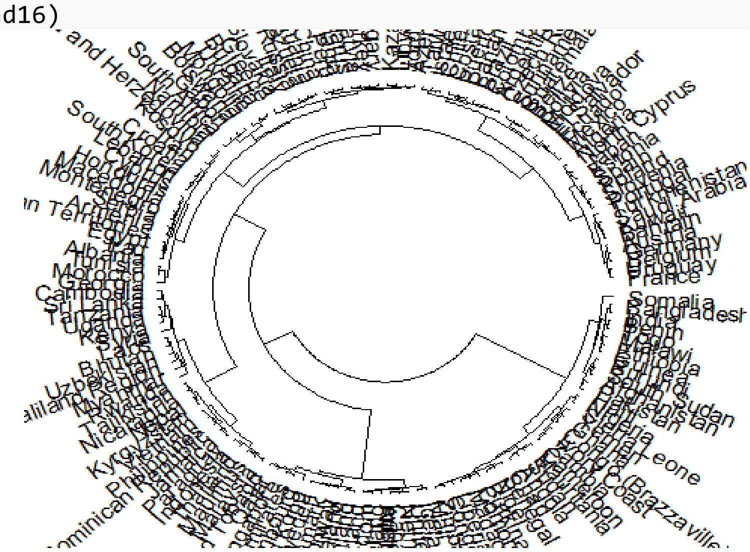


FOR DATASET HAPPY 2016

```
happy2016=read.csv("https://raw.githubusercontent.com/vigneshjmurali/Statistical-Predictive-Modelling/master/Datasets/World_Happiness_2016.csv")
dim(happy2016)
## [1] 157  13
# TAKING OUT HAPPINESS INFORMATION FROM THE GIVEN DATASET FOR THE CLUSTERING ANALYSIS
row.names(happy2016)<-happy2016$Country
happy2016cut<-happy2016[,7:13]
happy2016cut.s=scale(happy2016cut)
happy2016cut.d=dist(happy2016cut.s)
happy2016cut.hc.s=hclust(happy2016cut.d,method="ward.D")
happy2016cutdend=as.dendrogram(happy2016cut.hc.s)
labels_colors(happy2016cutdend)<-as.numeric(as.factor(happy2016$Region[happy2016cut.hc.s$order]))
dend16=as.dendrogram(happy2016cut.hc.s)
plot(happy2016cutdend)
```

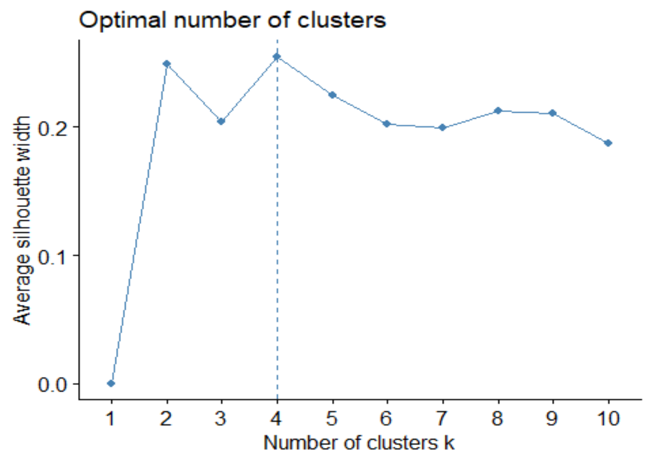



```
par(mar = rep(0,4))
dend16=as.dendrogram(happy2016cut.hc.s)
circlize_dendrogram(dend16)
```

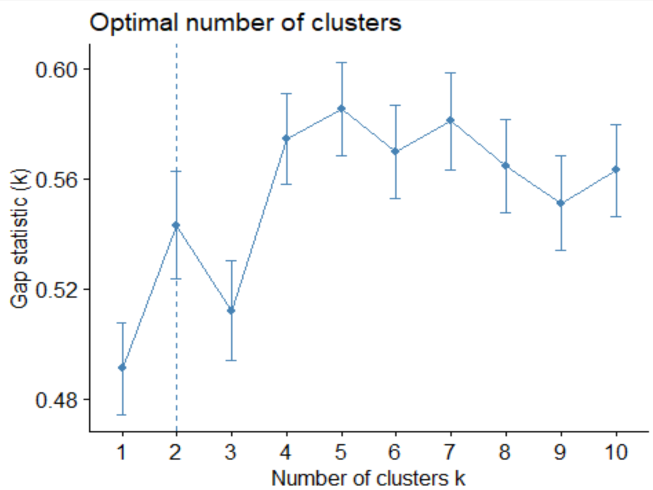


PARTITION CLUSTERING HAPPY 2016

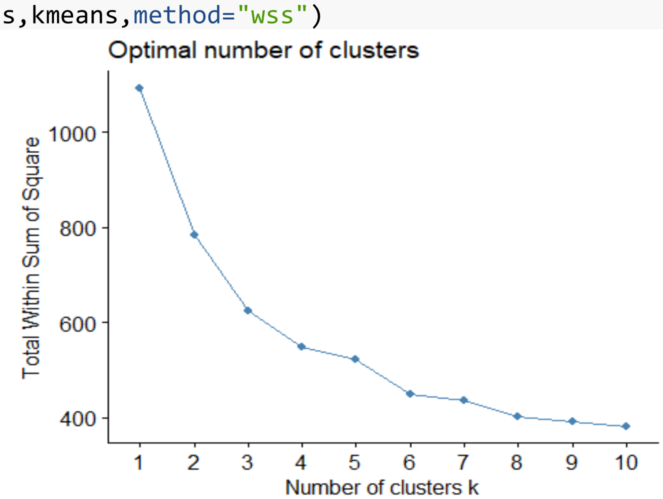
```
set.seed(123)
fviz_nbclust(happy2016cut.s,kmeans,method="silhouette")
```



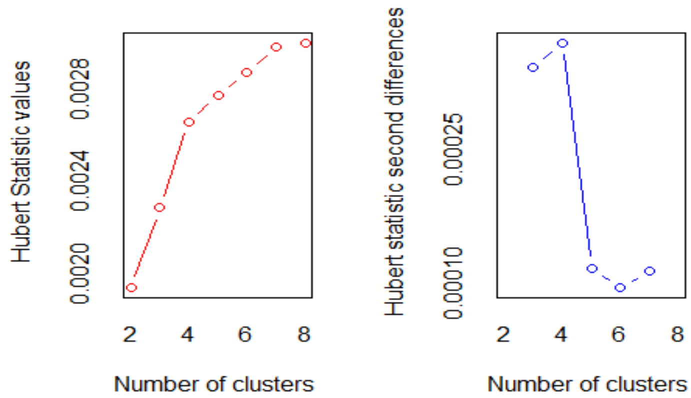

```r
fviz_nbclust(happy2016cut.s,kmeans,method="gap_stat")
```



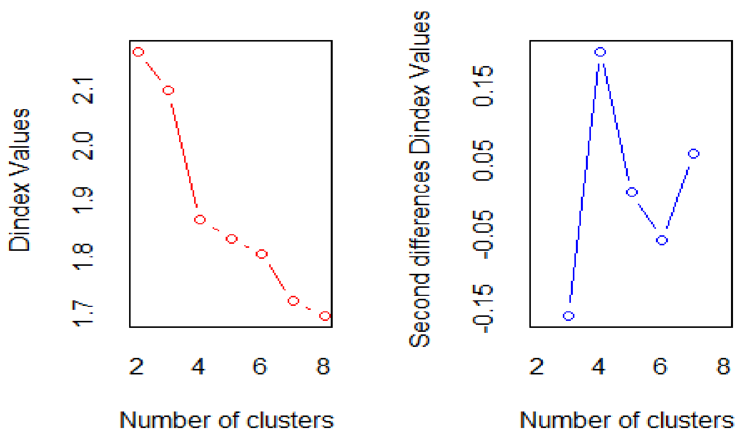

```r
fviz_nbclust(happy2016cut.s,kmeans,method="wss")
```



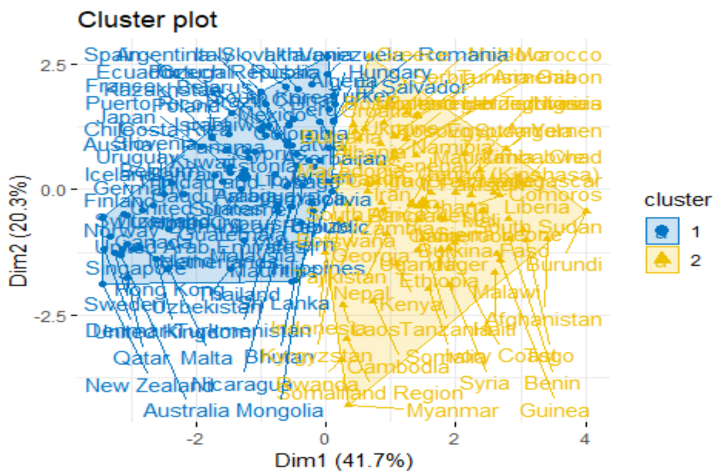

```r
happy16.nbclust<-happy2016cut %>% #Using NbClust
  scale() %>% NbClust(distance="euclidean",min.nc=2,max.nc=8,method="complete",index="all")
```



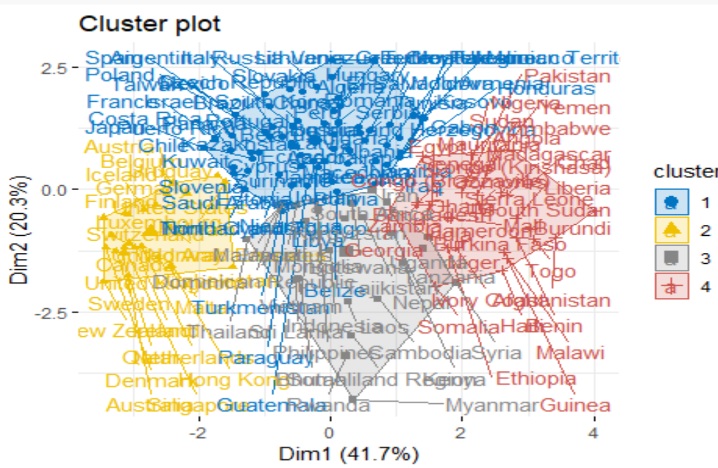

```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##            significant increase of the value of the measure i.e the significant peak in Hubert
##            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




```
## *** : The D index is a graphical method of determining the number of clusters.
##             In the plot of D index, we seek a significant knee (the significant peak in Dindex
##             second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
## *******************************************************************
## * Among all indices:
## * 7 proposed 2 as the best number of clusters
## * 1 proposed 3 as the best number of clusters
## * 9 proposed 4 as the best number of clusters
## * 2 proposed 6 as the best number of clusters
## * 1 proposed 7 as the best number of clusters
## * 3 proposed 8 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  4
## *******************************************************************
happy2016cut.k2<-kmeans(happy2016cut.s,centers=2,iter.max=100,nstart=25)
happy2016cut.k4<-kmeans(happy2016cut.s,centers=4,iter.max=100,nstart=25)
#pairs(happy2016cut[-1],pch=happy2016cut.k2$cluster,col=unclass(happy2016cut[,1]))
fviz_cluster(happy2016cut.k2,data=happy2016cut.s,ellipse.type="convex",palette="jco",repel=TRUE,gg
theme=theme_minimal())
```

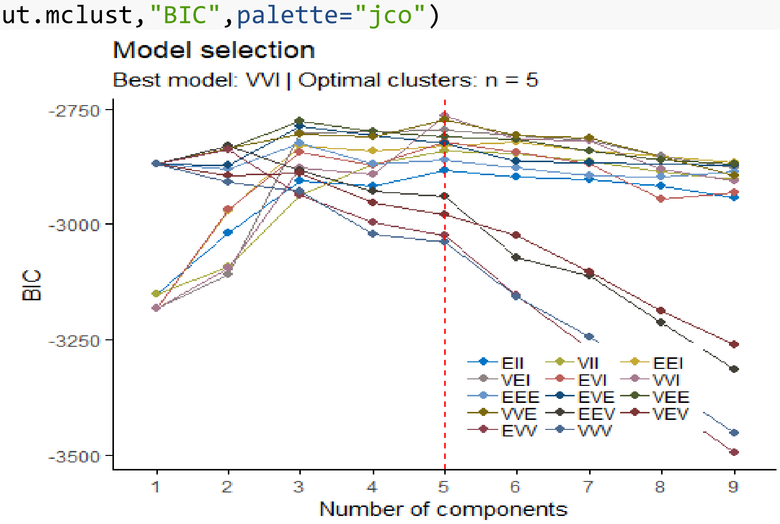



```r
fviz_cluster(happy2016cut.k4,data=happy2016cut.s,ellipse.type="convex",palette="jco",repel=TRUE,gg
theme=theme_minimal())
```



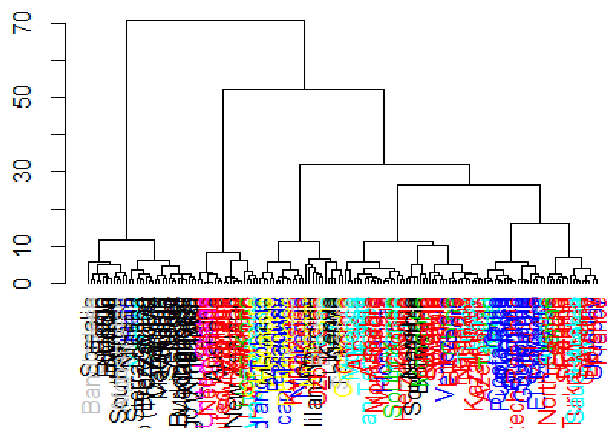
M-CLUST

```r
happy2016cut.mclust<-Mclust(happy2016cut.s)
summary(happy2016cut.mclust)
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
## Mclust VVI (diagonal, varying volume and shape) model with 5 components:
##  log.likelihood   n df       BIC       ICL
##       -1196.092 157 74 -2766.347 -2779.408
## Clustering table:
##  1  2  3  4  5
## 18 17 51 33 38
fviz_mclust(happy2016cut.mclust,"BIC",palette="jco")
```



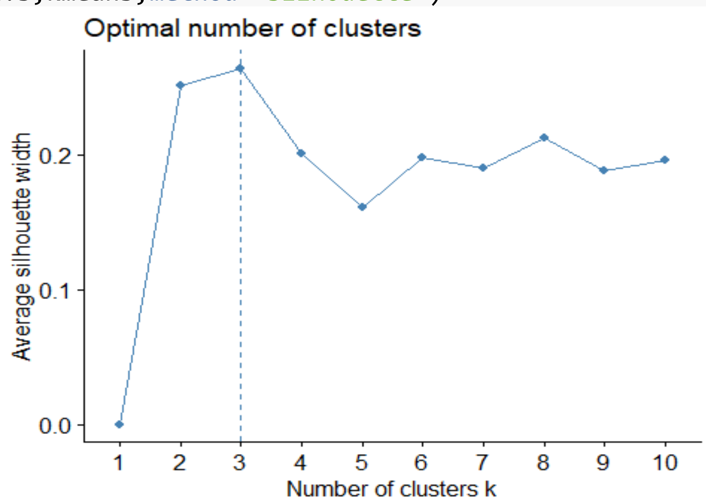
FOR DATASET HAPPY 2017

```
happy2017=read.csv("https://raw.githubusercontent.com/vigneshjmurali/Statistical-Predictive-Modelling/master/Datasets/World_Happiness_2017.csv")
dim(happy2017)
## [1] 155  12
# TAKING OUT HAPPINESS INFORMATION FROM THE GIVEN DATASET FOR THE CLUSTERING ANALYSIS
row.names(happy2017)<-happy2017$Country
happy2017cut<-happy2017[,6:12]
happy2017cut.s=scale(happy2017cut)
happy2017cut.d=dist(happy2017cut.s)
happy2017cut.hc.s=hclust(happy2017cut.d,method="ward.D")
happy2017cutdend=as.dendrogram(happy2017cut.hc.s)
labels_colors(happy2017cutdend)<-as.numeric(as.factor(happy2017$Region[happy2017cut.hc.s$order]))
dend17=as.dendrogram(happy2017cut.hc.s)
plot(happy2016cutdend)
```

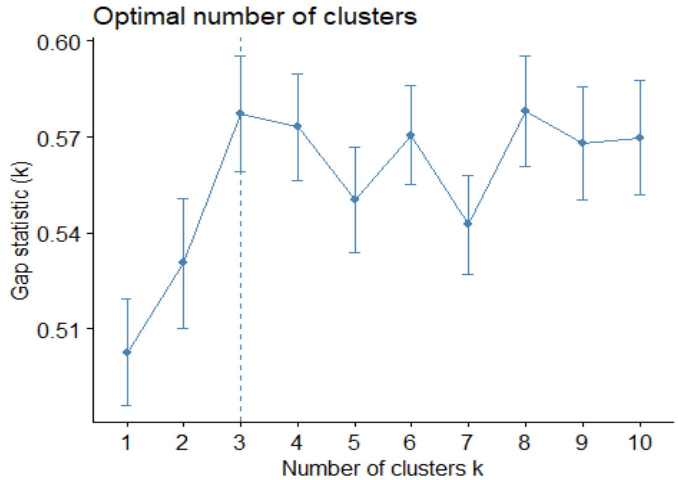


PARTITION CLUSTERING HAPPY 2017

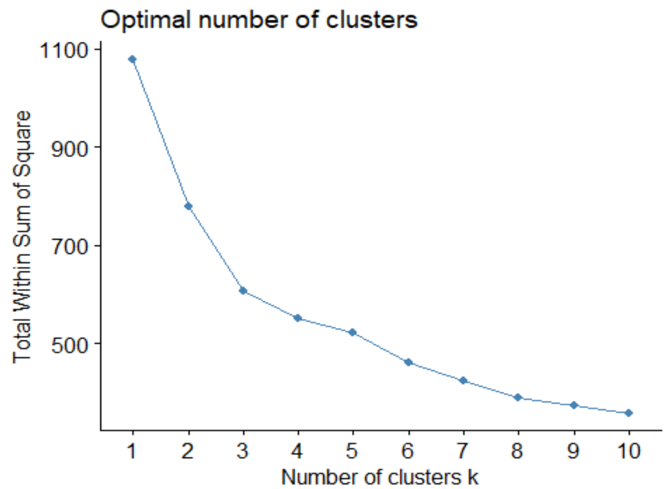
```
set.seed(123)
fviz_nbclust(happy2017cut.s,kmeans,method="silhouette")
```



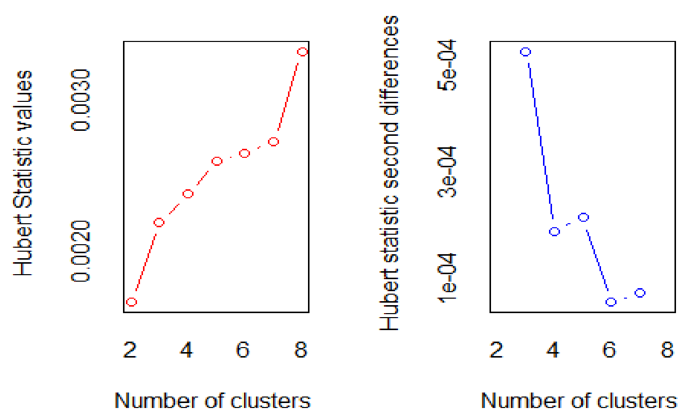

```
fviz_nbclust(happy2017cut.s,kmeans,method="gap_stat")
```



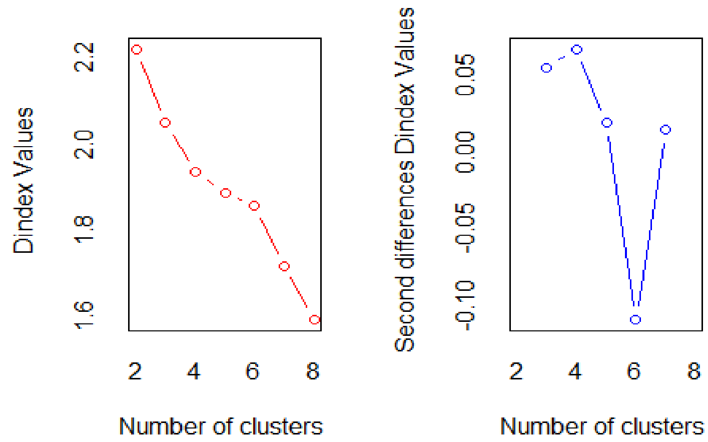

```
fviz_nbclust(happy2017cut.s,kmeans,method="wss")
```



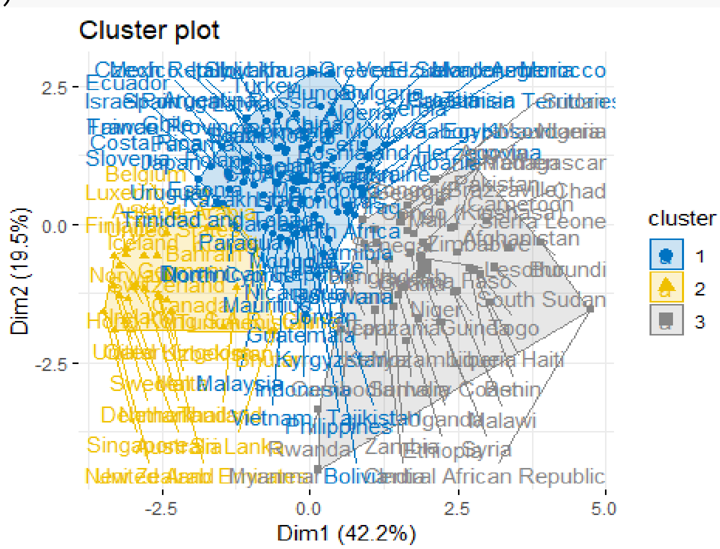

```
happy17.nbclust<-happy2017cut %>% #Using NbClust
  scale() %>% NbClust(distance="euclidean",min.nc=2,max.nc=8,method="complete",index="all")
```



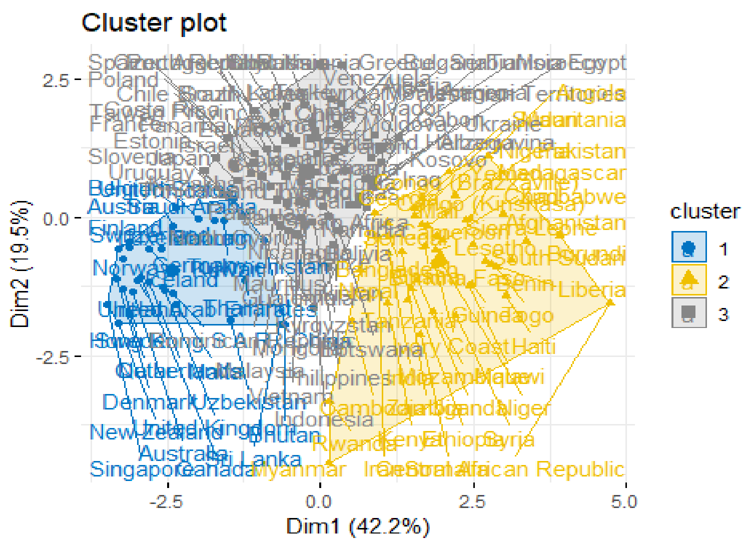

```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##               significant increase of the value of the measure i.e the significant peak in Hubert
##         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




```
## *** : The D index is a graphical method of determining the number of clusters.
##                In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                second differences plot) that corresponds to a significant increase of the value of
##                    the measure.
## *******************************************************************
## * Among all indices:
## * 7 proposed 2 as the best number of clusters
## * 4 proposed 3 as the best number of clusters
## * 3 proposed 4 as the best number of clusters
## * 4 proposed 6 as the best number of clusters
## * 2 proposed 7 as the best number of clusters
## * 3 proposed 8 as the best number of clusters
##                       ***** Conclusion *****
## * According to the majority rule, the best number of clusters is  2
## *******************************************************************
happy2017cut.k2sil<-kmeans(happy2017cut.s,centers=3,iter.max=100,nstart=25)
happy2017cut.k4gap<-kmeans(happy2017cut.s,centers=3,iter.max=100,nstart=25)
#pairs(happy2017cut[-1],pch=happy2017cut.k2sil$cluster,col=unclass(happy2017cut[,1]))
fviz_cluster(happy2017cut.k2sil,data=happy2017cut.s,ellipse.type="convex",palette="jco",repel=TRUE
,ggtheme=theme_minimal())
```

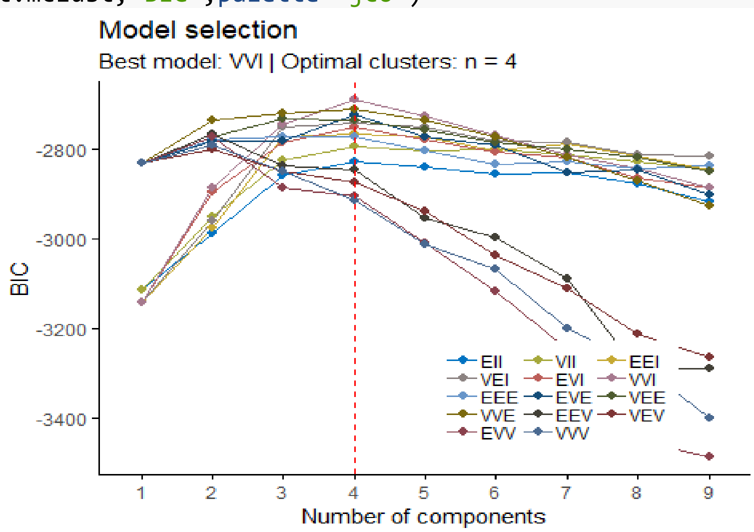



```
fviz_cluster(happy2017cut.k4gap,data=happy2017cut.s,ellipse.type="convex",palette="jco",repel=TRUE
,ggtheme=theme_minimal())
```



M-CLUST

```
happy2017cut.mclust<-Mclust(happy2017cut.s)
summary(happy2017cut.mclust)
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
## Mclust VVI (diagonal, varying volume and shape) model with 4 components:
##  log.likelihood   n df       BIC       ICL
##       -1196.501 155 59 -2690.564 -2708.876
## Clustering table:
##  1  2  3  4
## 17 24 70 44
fviz_mclust(happy2017cut.mclust,"BIC",palette="jco")
```

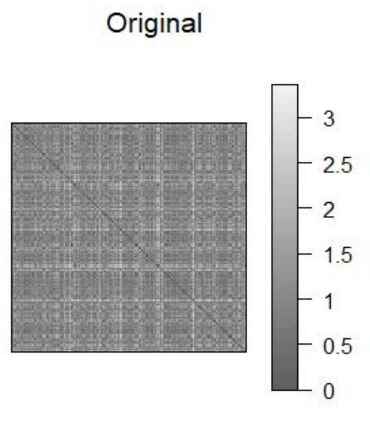


SUMMARY OF CLUSTER

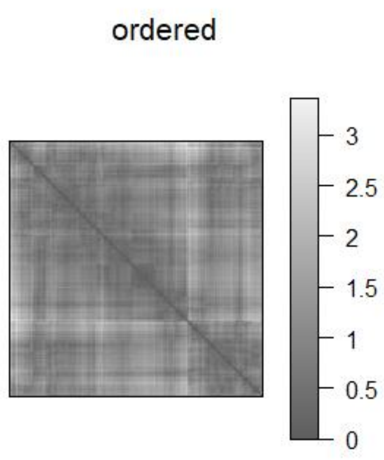
```
Cluster_Method<-c('Sulhouette','Gap-Stat','WSS','NBClust', ' MClust')
Happy_2015<-c(2,4,2,5,4)
Happy_2016<-c(4,2,4,4,5)
Happy_2017<-c(3,3,3,2,4)
results<-data.frame(Cluster_Method,Happy_2015,Happy_2016,Happy_2017) ; results
##   Cluster_Method Happy_2015 Happy_2016 Happy_2017
## 1     Sulhouette          2          4          3
## 2       Gap-Stat          4          2          3
## 3            WSS          2          4          3
## 4        NBClust          5          4          2
## 5         MClust          4          5          4
```

SERIATION ANALYSIS

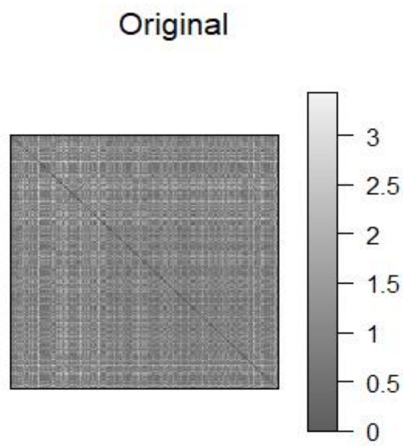
```
set.seed(34)
x2015<-as.matrix(happy2015cut)
x20151<-x2015[sample(seq_len(nrow(x2015))),]
d2015<-dist(x20151)
o2015<-seriate(d2015,method="OLO")
pimage(d2015,main="Original")
```



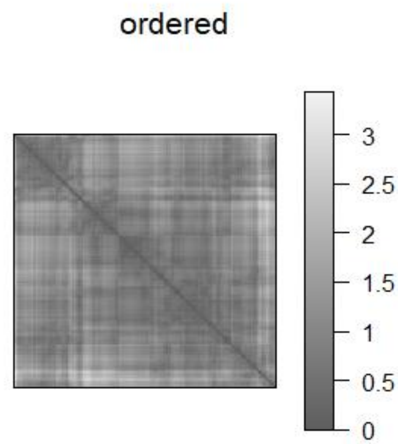

```
pimage(d2015,o2015,main="ordered")
```



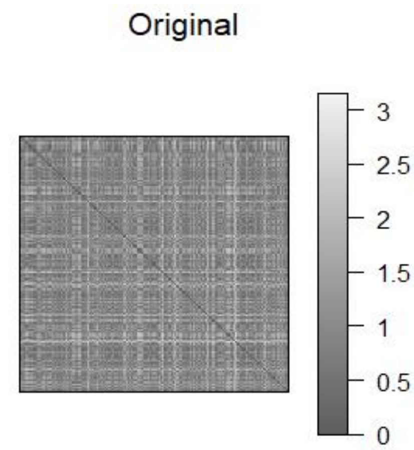

```
get_order(o2015)
##   [1] 123 102  36  74  40  31  54  23  59  83  72 117  27  26 124 110  44
##  [18] 119  12  34 114  52 146  68 151  50  13 108  58  97  80 116  28 144
##  [35] 135  82 111 143  77 129  10 101 118 145  49  67  78  38 109   1  66
##  [52]   9  19  11 139 112 138  71   4  61 128 152  15   5  73  39  93  55
##  [69]  56 141  95 106  75  33  91 125 132  53 157  47  14  70  41  18  37
##  [86]  85  42  76 105  84 127 104  24 148 107 120   7 133  35  25  21  92
## [103] 142  20  96  57 134  45 147  65  60  88 100  98  43  99 158 156 126
## [120]  51 136 131  46 103  79  48 113  62  81 140  89   3 155 115 121  29
## [137]  30 122 130  90  94  17   8  64 149  16  63 137   2 154 150  32 153
## [154]  22  87  86  69   6
data("happy2016cut")
## Warning in data("happy2016cut"): data set 'happy2016cut' not found
x2016<-as.matrix(happy2016cut)
x2016<-x2016[sample(seq_len(nrow(x2016))),]
d2016<-dist(x2016)
o2016<-seriate(d2016,method="OLO")
pimage(d2016,main="Original")
```



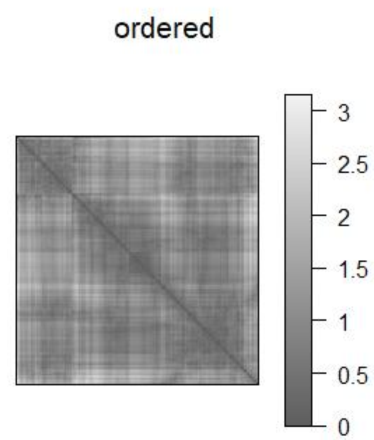

```
pimage(d2016,o2016,main="ordered")
```




```
get_order(o2016)
##   [1]  17  76 155 126  57  66  44  46  78 147  13  51   5 100  34  80 105
##  [18] 144  93  72  16 133  29  24 119  68 145  58   7  64  40  79  89  61
##  [35]  96 123  49   6  27   2 118 140 125   8  25 117 148  10  54 135  98
##  [52] 112  84  26  92 132  71 128 110 104 142  70 131  31  56  22  55  43
##  [69]  50 157  67  90 108  48  91 120 107 130  97  18 121  45  77  95 129
##  [86]  36 136 151  81 137 111  21  60  53 124  20  19 134 114  12 146  85
## [103]  82 154  35 102  62 106 101  87  86  69  63 103 152 149  15 141 138
## [120]  74  73  83 150  38  88   4 122  59 109  99 153  94 116   1  47  14
## [137] 115  65 139  42   3 127   9  39  23  37 113 143  32  33 156  52  41
## [154]  28  11  75  30
data("happy2017cut")
x2017<-as.matrix(happy2017cut)
x2017<-x2017[sample(seq_len(nrow(x2017))),]
d2017<-dist(x2017)
o2017<-seriate(d2017,method="OLO")
pimage(d2017,main="Original")
```




```
pimage(d2017,o2017,main="ordered")
```




```
get_order(o2017)
##   [1] 125  83  89 149  32  31 150 152  35  11 141  71  27  94  93  49  46
##  [18]  33  34 154 104  54 102 146 119  25  82 137  63  44 147  65 128 139
##  [35]  19  43 117  70  69  61  47  41  53  78 131 108 115 153 127  91   5
##  [52]   4  75 116  87 126  30  57  77 132 138  50  55 134  73  52  56  79
##  [69]  74  59 148  39  64 151 122  92   8 118 140  18   7 105  72 143   6
##  [86]  67 133  68  28  98  17 103  15 121  81  14 136   1 120  62  88 106
## [103] 144  13  22  10  76  97   2  96  42  29 100 114 110 111  21  66  26
## [120]   9 130   3  84 113 123  48  95 135  24  80 129 107  85 155  37  36
## [137]  51  38 145  20  58  23  16  12  60  86 112 101  99 109 142  45  40
## [154]  90 124
```

The Seriation analysis is used to compare the generated order of the happiness without the happiness score. when the order shows for every year the order gets changed each time the seriation is used to run and different order gets generated each time even though i set the seed it happens. I am not sure if that is the correct way to analyse the order based on the seriation to see the best happiness of the country. For example in 2015 once it shows kenya second time it runs it gives Nepal so I am not sure if this order makes good sence in this point of dataset.

1). Norway tops the global happiness rankings :
Based on the analysis of cluster or other association analysis we cant say that norway is topping the list. But when we see the cluster we can see that it does rank among one of the few top countries but clearly cannot be said in that way.

2. All top ten countries rank highly on all the main features found to support happiness

I really feel the countries are grouped based on there scores in each variable in that case if we just compare with the cluster analysis in the year 2015 the cluster of some countries which we can understand makes really as a close cluster and form a group according to the score . I could really belive the cluster how it has formed itself for each year. But, on the basis of seriation analysis I couldnt see the rankings based on the main features.

3) Happiness is both social and personal

I couldnt see any variable with this but, sensiblly seeing since the dataset is based on the happiness this should be related with social and personal for example “Dystopia” variable really explains the community or society that is undesirable or frightening sot this becomes a social issue in the happiness ranking while the family or freedom really comes with personal issues. So these variables in the dataset really helps in finding both social and personal of the citizens to figure out how the happiness is ranked in the world for each country.

4) Unemployment causes a major fall in happiness, and even for those in work the quality of work can cause major variations in happiness

Though we dont have any variable which says unemployment, some variables like Trust Government corruption or the Economic GDP and even genoricity will really helps us to explain few issues in each country happening regarding the job oppurtunities. For example if the countrys government is corrupted then unemployment will be a real factor and also economic growth is also a problem.

5) China are no happier than most countries, though richer and longer longevity

Based on all the cluster analysis and the seriation i feel china is in the middle which supports both the richer and the poor which makes the country having no happier

6) Much of Africa is struggling

In the hirachial clustering the african countries are listed in one color and also in other clustering it has no other combinations with other parts of the countries which stands appart. but based on the seriation analysis we cant really say that. When seeing the data with known facts our clustering methods makes real sence and helps us to say which country might group with what. So on that basis as a human being i feel african countries are little struggling.

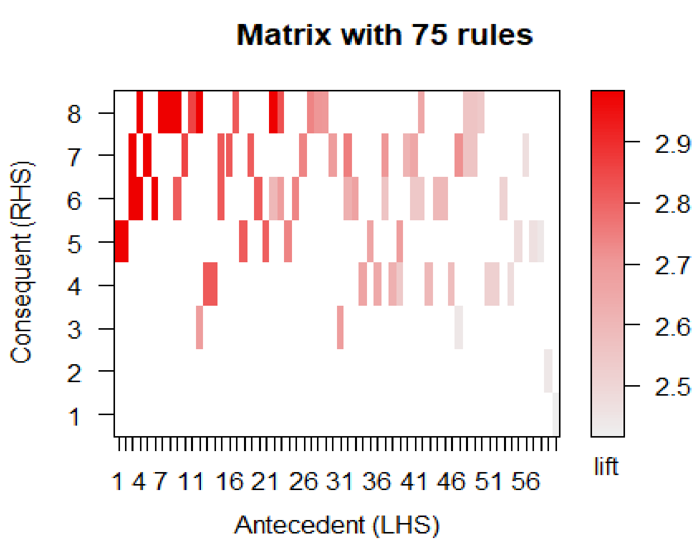
7) Happiness has fallen in America

I think that doesnt really feel in my analysis. Because when i see the united states in 2015 clusteing is stays in the same group where other european countries are available and until 2017 it stays with same group in this case i cant say that happiness has fallen. But if i could do the correct seriation analysis then I may be able to answer this.

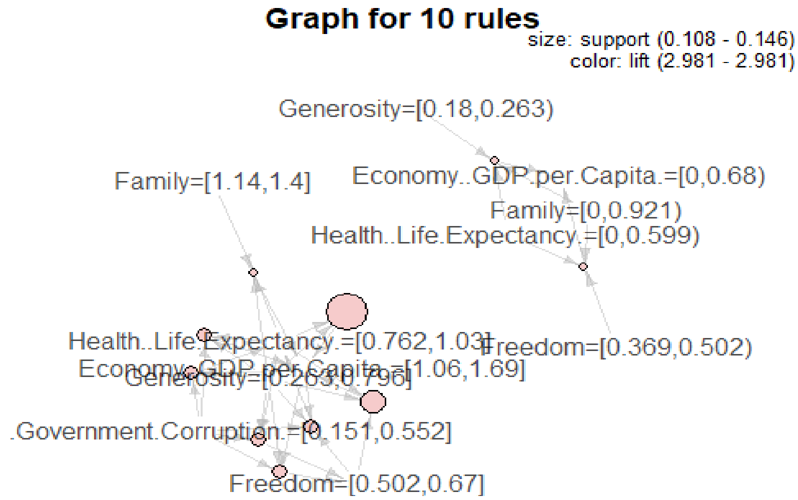
ASSOCIATION RULES

```
h2015<-discretizeDF(happy2015cut)
rules<-apriori(h2015)
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5     0.1      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 15
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[21 item(s), 158 transaction(s)] done [0.00s].
## sorting and recoding items ... [21 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [80 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
summary(rules)
## set of 80 rules
##
## rule length distribution (lhs + rhs):sizes
##  2  3  4  5
```

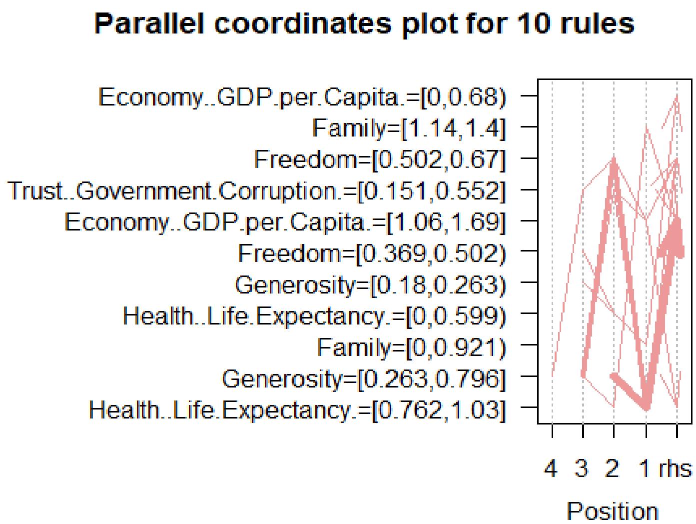
```
##  2 33 33 12
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   3.000   4.000   3.688   4.000   5.000
##
## summary of quality measures:
##     support         confidence          lift            count
##  Min.   :0.1013   Min.   :0.8000   Min.   :2.385   Min.   :16.0
##  1st Qu.:0.1076   1st Qu.:0.8558   1st Qu.:2.551   1st Qu.:17.0
##  Median :0.1139   Median :0.9000   Median :2.683   Median :18.0
##  Mean   :0.1285   Mean   :0.9007   Mean   :2.685   Mean   :20.3
##  3rd Qu.:0.1392   3rd Qu.:0.9444   3rd Qu.:2.816   3rd Qu.:22.0
##  Max.   :0.2785   Max.   :1.0000   Max.   :2.981   Max.   :44.0
##
## mining info:
##   data ntransactions support confidence
##  h2015           158     0.1        0.8
subrules<-rules[quality(rules)$confidence>0.8]
subrules
## set of 75 rules
plot(subrules,method="matrix",measure = "lift")
## Itemsets in Antecedent (LHS)
##  [1] "{Family=[0,0.921),Health..Life.Expectancy.=[0,0.599),Generosity=[0.18,0.263)}"
##  [2] "{Family=[0,0.921),Health..Life.Expectancy.=[0,0.599),Freedom=[0.369,0.502)}"
##  [3] "{Economy..GDP.per.Capita.=[1.06,1.69],Trust..Government.Corruption.=[0.151,0.552],Generos
ity=[0.263,0.796]}"
##  [4] "{Health..Life.Expectancy.=[0.762,1.03],Trust..Government.Corruption.=[0.151,0.552],Genero
sity=[0.263,0.796]}"
##  [5] "{Economy..GDP.per.Capita.=[1.06,1.69],Freedom=[0.502,0.67],Trust..Government.Corruption.=
[0.151,0.552],Generosity=[0.263,0.796]}"
##  [6] "{Economy..GDP.per.Capita.=[1.06,1.69],Health..Life.Expectancy.=[0.762,1.03],Trust..Govern
ment.Corruption.=[0.151,0.552],Generosity=[0.263,0.796]}"
##  [7] "{Health..Life.Expectancy.=[0.762,1.03],Freedom=[0.502,0.67],Trust..Government.Corruption.
=[0.151,0.552],Generosity=[0.263,0.796]}"
##  [8] "{Family=[1.14,1.4],Health..Life.Expectancy.=[0.762,1.03],Freedom=[0.502,0.67],Generosity=
[0.263,0.796]}"
##  [9] "{Family=[1.14,1.4],Health..Life.Expectancy.=[0.762,1.03],Generosity=[0.263,0.796]}"
```




```
subrules2<-head(sort(rules,by="lift"),10)
plot(subrules2,method = "graph")
```




```r
plot(subrules2, method="paracoord")
```




```r
#plot(subrules2, method = "grouped matrix", engine = "interactive")
```