Stat 517 Project #2

Name - **ABHISHEK KUMAR THAKUR**

Major - Materials Science & Engineering

Question 0 [from last time] : Predicting Job Salary

The given dataset has been imported which is of 10000 rows × 12 columns size.

Cleaning Dataset and taking care of missing values

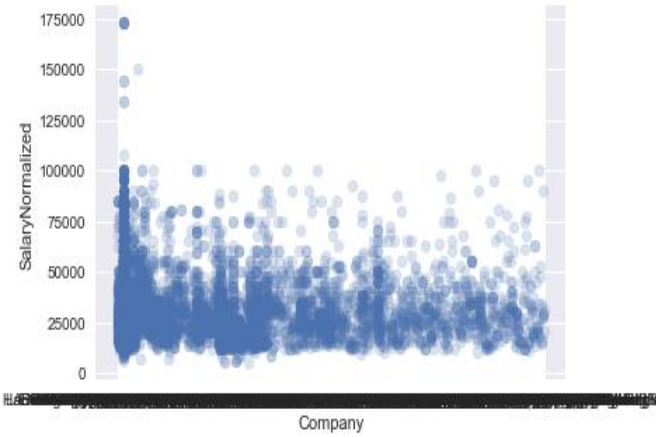
```
print (salary.shape)

print (salary.columns)

print (salary.isnull().sum())
```

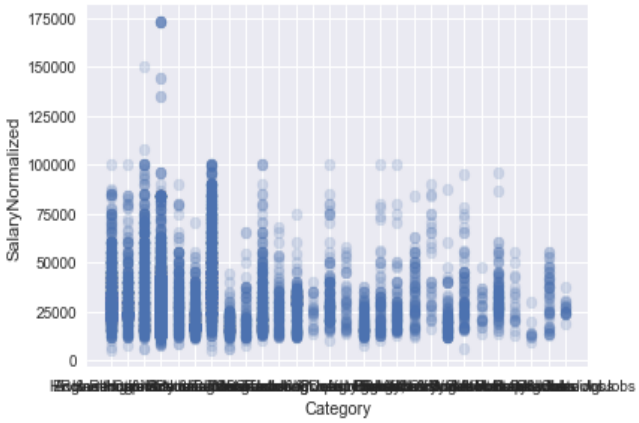
This reveals that ContractType, ContractTime and Company has null values 6444 5263 and 4049 respectively.

```
print (salary["Title"].value_counts()) #The most frequent value in this column is "Staff Nurse"

print (salary["ContractType"].value_counts()) #The most frequent value in this column is "full_time"

print (salary["ContractTime"].value_counts()) #The most frequent value in this column is "permanent"

print (salary["Company"].value_counts()) #The most frequent value in this column is "JOBG8"

salary["Title"].fillna(value = 'Staff Nurse', inplace=True)

salary["ContractType"].fillna(value = 'full_time', inplace=True)

salary["ContractTime"].fillna(value = 'permanent', inplace=True)

salary["Company"].fillna(value = 'JOBG8', inplace=True)
```

General Curves

```
plt.scatter(salary['Company'], salary['SalaryNormalized'], alpha=0.2)

plt.xlabel('Company')

plt.ylabel('SalaryNormalized');
```





Splitting into Training and Test datasets

```
from sklearn.cross_validation import train_test_split

xtrain, xtest, ytrain, ytest = train_test_split(x_salary, y_salary, random_state = 1, test_size = 0.25)
```

Multiple Linear Regression

```
reg = linear_model.LinearRegression()

model = reg.fit(xtrain,ytrain)

predictions = reg.predict(xtest)

print(predictions)[0:10]
```

```
[[24940.53877159] [18389.85424668] [29088.74314462] [26732.17099081] [41215.95025693] [28485.94838764]
[22269.85588113] [40892.98752593] [28450.71371015] [28660.55953689]]
```

```
from sklearn.metrics import mean_squared_error, r2_score

print("Mean squared error: %.2f"

      % mean_squared_error(ytest, predictions))

print('Variance score: %.2f' % r2_score(ytest, predictions))
```

```
Mean squared error: 109195132.97 Variance score: 0.58
```

Ridge Regression

```
from sklearn.linear_model import Ridge

ridge = Ridge().fit(xtrain, ytrain)

acc_rr1 = ridge.score(xtrain, ytrain) * 100

acc_rr = ridge.score(xtest, ytest) * 100
```

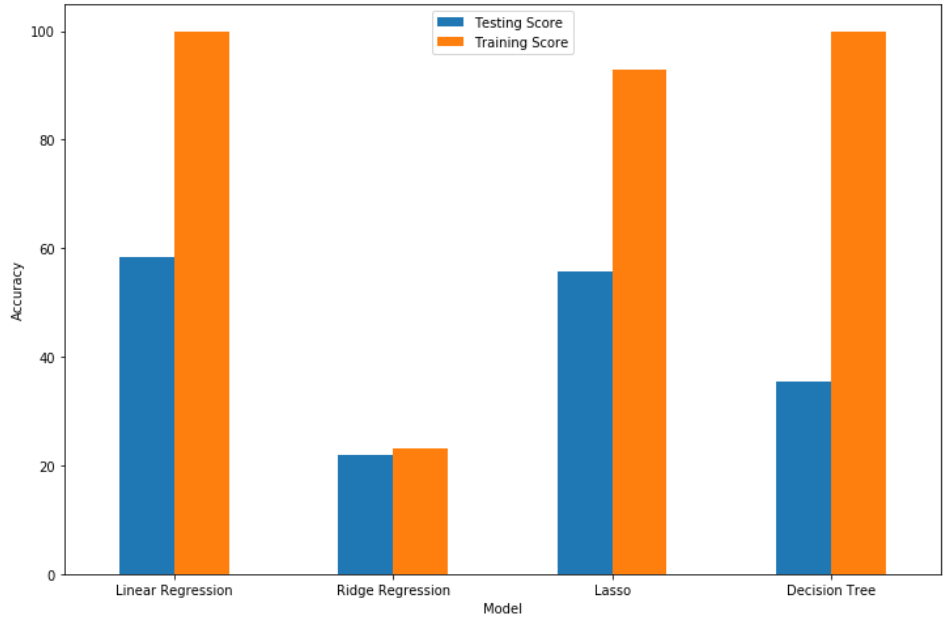
```
print("Training set score: {:.2f}".format(ridge.score(xtrain, ytrain) * 100, 2))

print("Test set score: {:.2f}".format(ridge.score(xtest, ytest) * 100, 2))
```

```
Training set score: 23.06 Test set score: 21.90
```

Lasso

```
from sklearn.linear_model import Lasso

lasso = Lasso().fit(xtrain, ytrain)

acc_l1 = lasso.score(xtrain, ytrain) * 100

acc_l = lasso.score(xtest, ytest) * 100

print("Training set score: {:.2f}".format(lasso.score(xtrain, ytrain) * 100, 2))

print("Test set score: {:.2f}".format(lasso.score(xtest, ytest) * 100, 2))

print("Number of features used: {}".format(np.sum(lasso.coef_ != 0)))
```

```
Training set score: 92.90 Test set score: 55.68
```



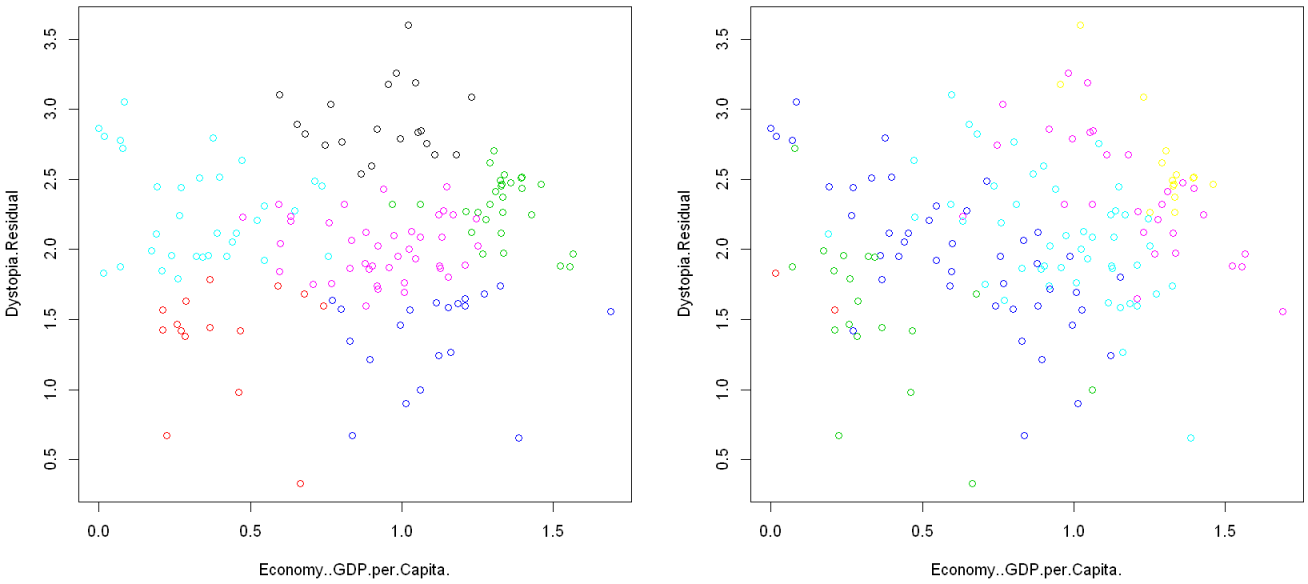
Discussion

In this work, some graphs has been presented which can reveal sensible information about the given dataset. The developed graphs can be used to study the dependency of various non-target variable on the target variable. Further, we have developed 3 feasible models which can be used for analysis of the given dataset. Similar other methods can also be used for the purpose. The best model based on their performance of training and test data is the Multiple Linear Regression.

Question 1: Exploring World Happiness

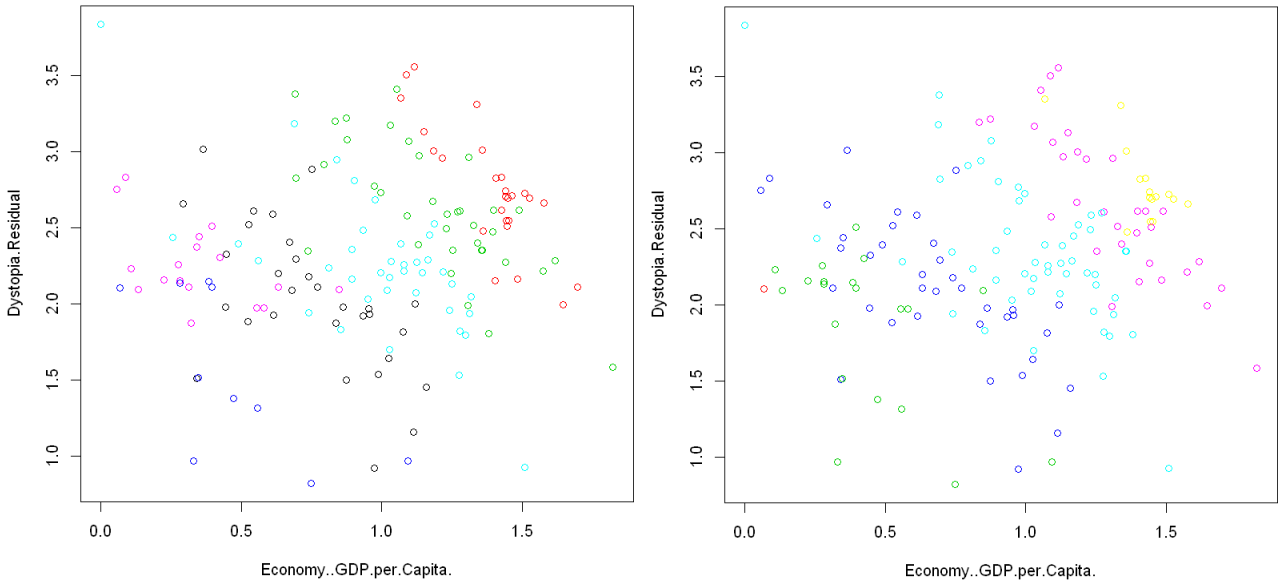
K-Means Clustering on Happiness 2015 Dataset

```
result_15 <- kmeans(happy_15, 6)
plot(happy_15[c("Economy..GDP.per.Capita.","Dystopia.Residual")], col = result_15$cluster)
plot(happy_15[c("Economy..GDP.per.Capita.","Dystopia.Residual")], col =
original_happy_15$Happiness.Score)
```



K-Means clustering on Happiness 2016 Dataset.

```
result_16 <- kmeans(happy_16, 6)
plot(happy_16[c("Economy..GDP.per.Capita.","Dystopia.Residual")], col = result_16$cluster)
plot(happy_16[c("Economy..GDP.per.Capita.","Dystopia.Residual")], col =
original_happy_16$Happiness.Score)
```

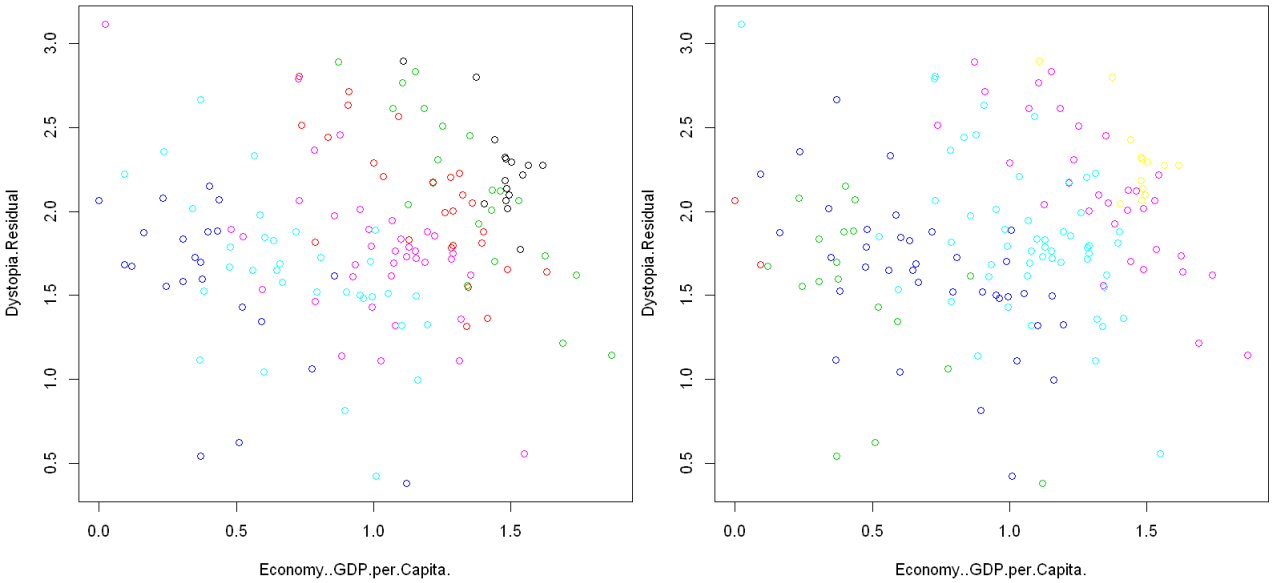


K-Means clustering on Happiness 2017 Dataset.

result_17 <- kmeans(happy_17, 6)

plot(happy_17[c("Economy..GDP.per.Capita.","Dystopia.Residual")], col = result_17$cluster)

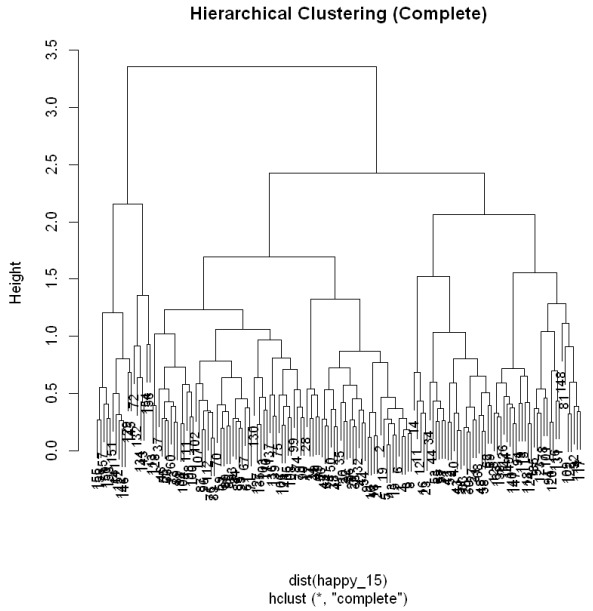
plot(happy_17[c("Economy..GDP.per.Capita.","Dystopia.Residual")], col = original_happy_17$Happiness.Score)



Hierarchical Clustering on Happiness 2015 Dataset.

happy_15_complete = hclust(dist(happy_15), method = "complete")

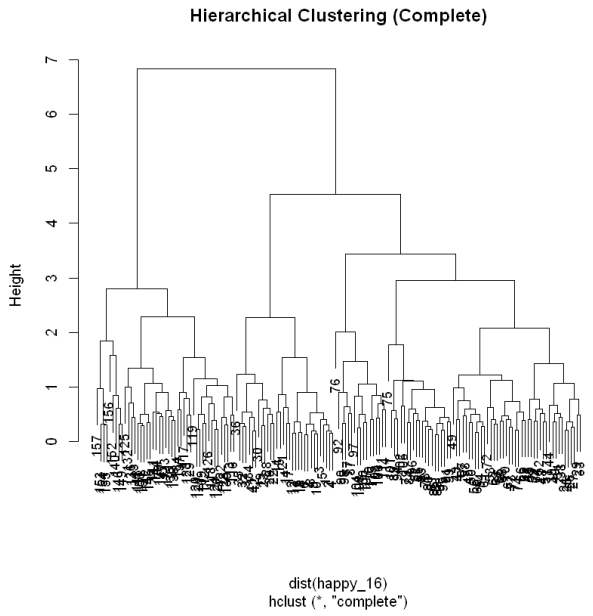
plot(happy_15_complete, main = "Hierarchical Clustering (Complete)", cex = 0.9)



Hierarchical Clustering on Happiness 2016 Dataset.

happy_16_complete = hclust(dist(happy_16), method = "complete")

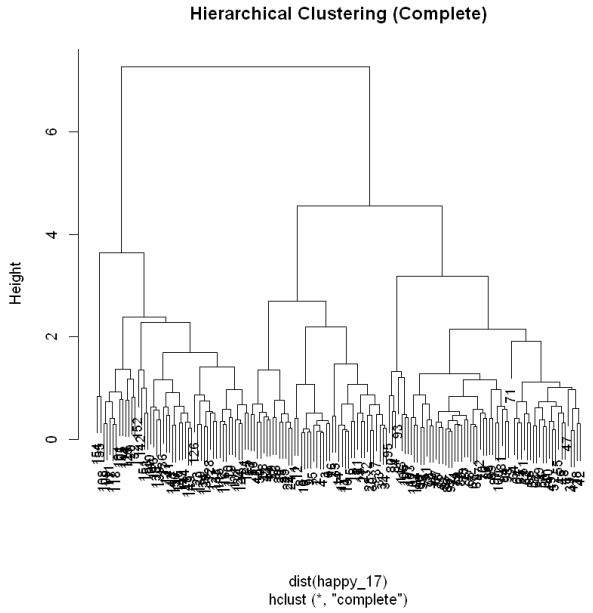
plot(happy_16_complete, main = "Hierarchical Clustering (Complete)", cex = 0.9)



Hierarchical Clustering on Happiness 2017 Dataset.

happy_17_complete = hclust(dist(happy_17), method = "complete")

plot(happy_17_complete, main = "Hierarchical Clustering (Complete)", cex = 0.9)

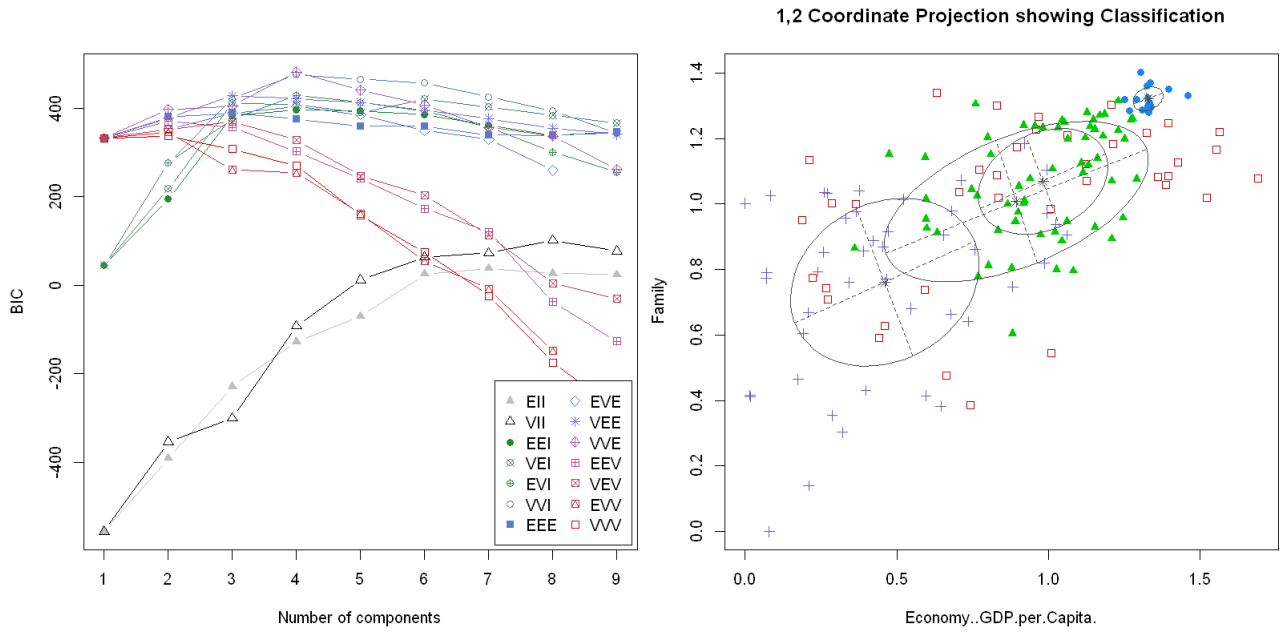


Using Gaussian Mixture on Happiness 2015 dataset.

```
cluster_happy_15 <- Mclust(happy_15)
cluster_happy_15$classification
```

```
1111111111331323211212323132333323223232423323333234332333333233333332222323334244333433332243433422243333433423444444424344422342242434444424424424244444424244
```

Cluster_1	Cluster_2	Cluster_3	Cluster_4
Switzerland	United States	Israel	Togo
Iceland	Luxembourg	Costa Rica	Burundi
Denmark	United Arab Emirates	Mexico	Benin
Norway	Oman	Brazil	Afghanistan
Canada	Singapore	Venezuela	Burkina Faso

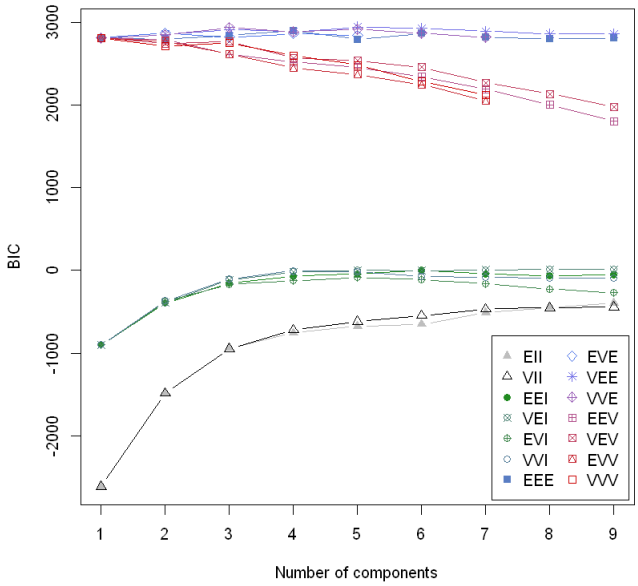


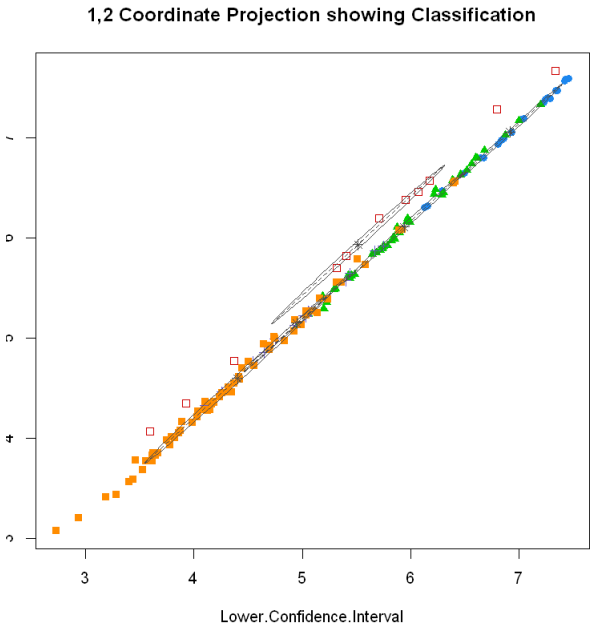
Using Gaussian Mixture on Happiness 2016 dataset.

```
cluster_happy_16 <- Mclust(happy_16)
cluster_happy_16$classification
```

```
1121111111311321311131133331353351323332112333535332333334333433552343332455335335355444533544455345555455544555255555544555555545552555555555255555555555555
```

Cluster_1	Cluster_2	Cluster_3	Cluster_4	cluster_5
Denmark	Iceland	Israel	South Korea	Malta
Switzerland	Puerto Rico	Costa Rica	North Cyprus	Thailand
Norway	Qatar	Brazil	Cyprus	Malaysia
Finland	Suriname	Mexico	Croatia	Uzbekistan
Canada	Trinidad & Tobago	Chile	Serbia	Turkmenistan





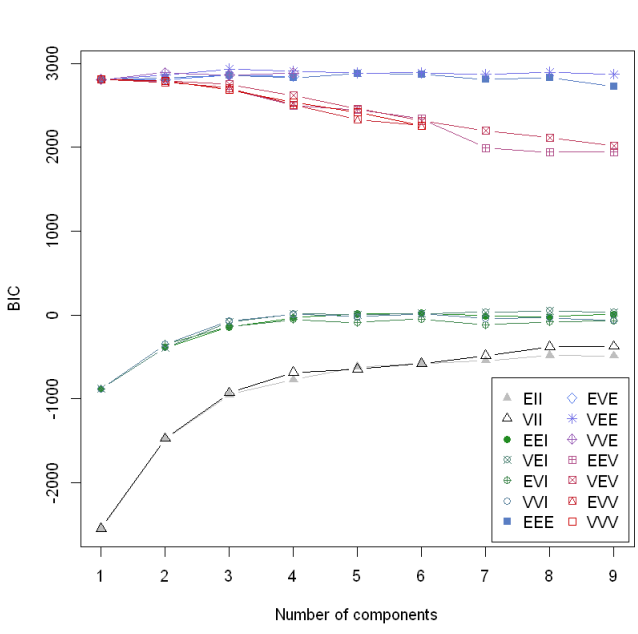
Using Gaussian Mixture on Happiness 2017 dataset.

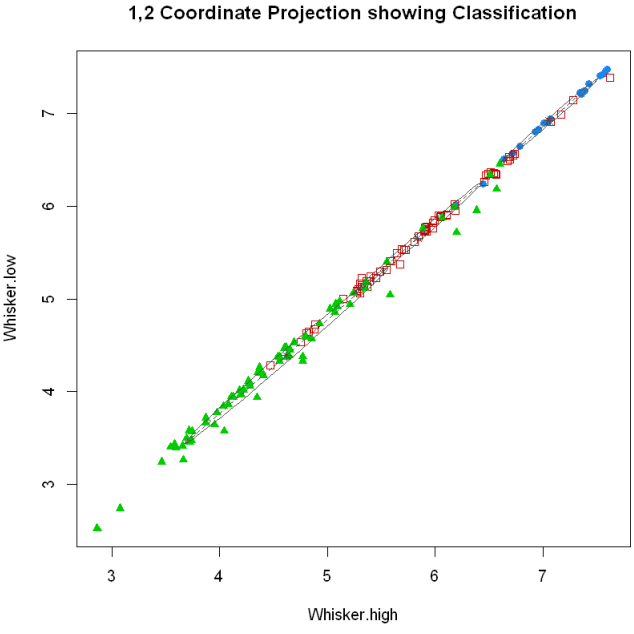
cluster_happy_17 <- Mclust(happy_17)

cluster_happy_17$classification

1121111111221211111212222132222322321312132222322322222222232222222222232222322233222232222222323333333222233323333333333323333333333333333333333333333333333333333333333

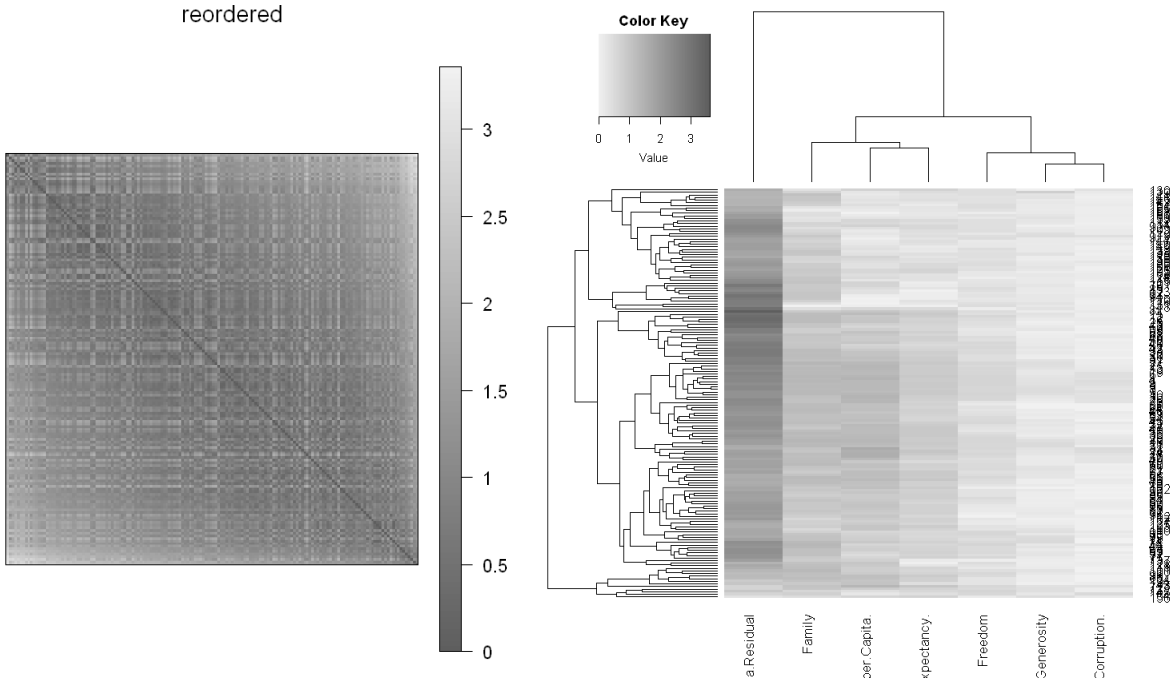
Cluster_1	Cluster_2	Cluster_3
Norway	Iceland	Malta
Denmark	Israel	Thailand
Switzerland	Costa Rica	Qatar
Finland	United States	Trinidad & Tobago
Netherlands	Chile	Malaysia





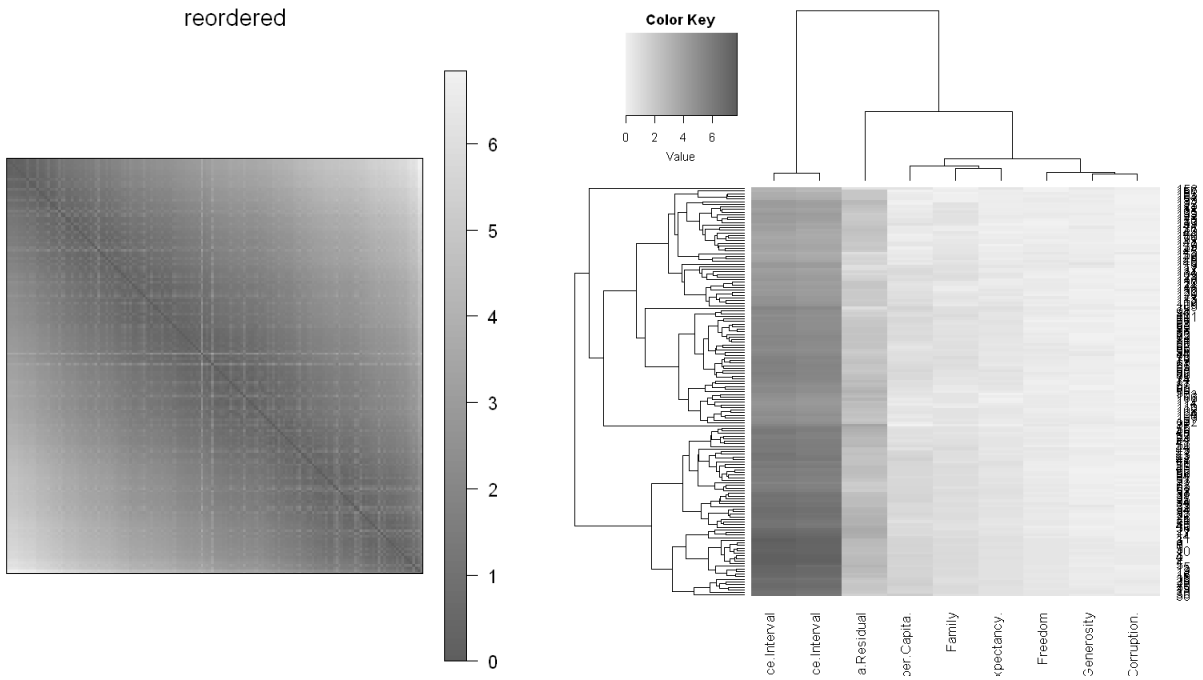
Seriation analysis on 2015 dataset

```
library(seriation)
x <- as.matrix(happy_15)
d <- dist(x)
o <- seriate(d)
pimage(d, main='original')
pimage(d, o, main='reordered')
```



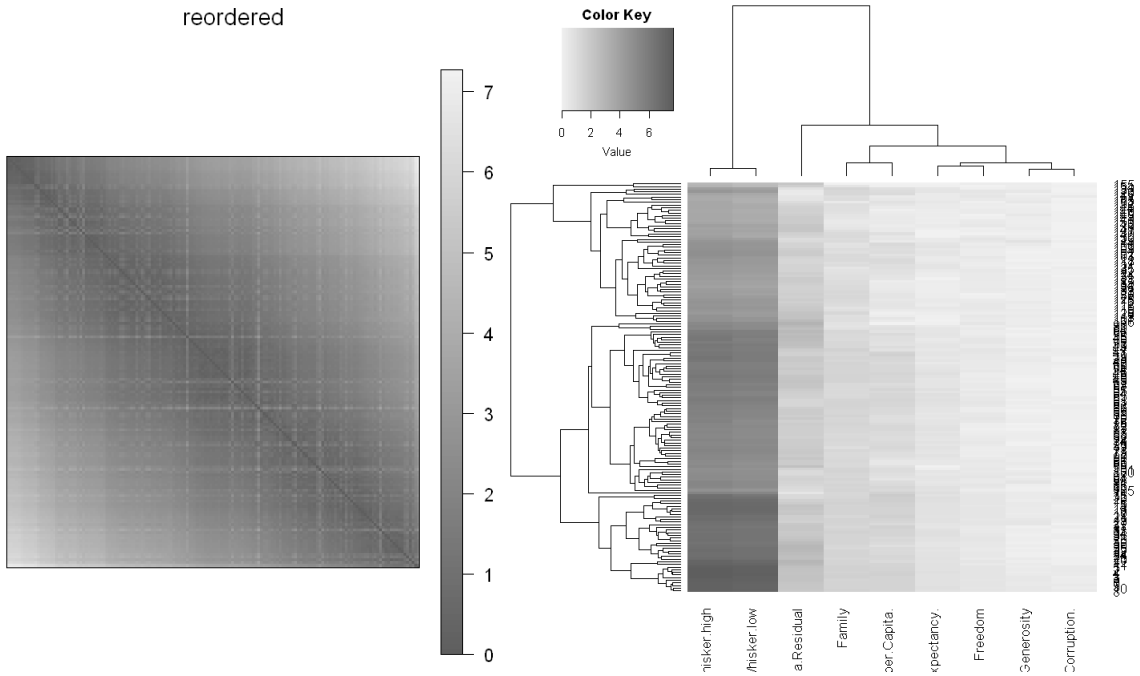
Seriation analysis on 2016 dataset

```
library(seriation)
x <- as.matrix(happy_16)
d <- dist(x)
o <- seriate(d)
pimage(d, main='original')
pimage(d, o, main='reordered')
```



Seriation analysis on 2017 dataset

```
library(seriation)
x <- as.matrix(happy_17)
d <- dist(x)
o <- seriate(d)
pimage(d, main='original')
pimage(d, o, main='reordered')
```



Discussion

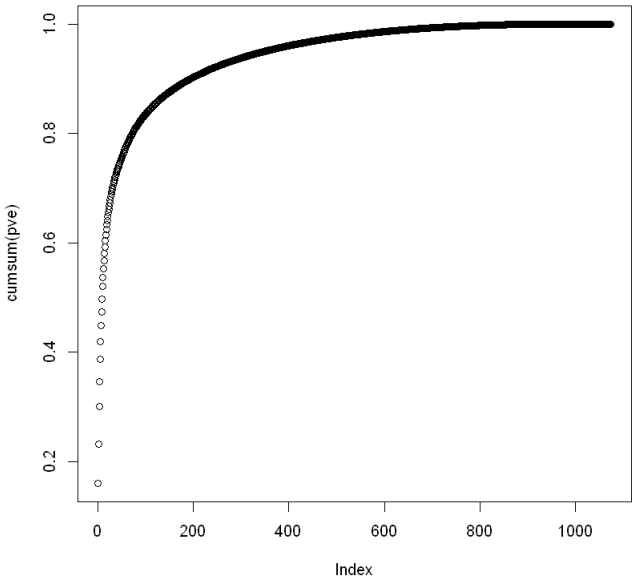
Thus we have conducted a thorough clustering analysis on the happiness data for 2015, 2016 and 2017 years. We have shown best method (VEE, VVE etc) for each dataset in order to understand various features in the dataset. 2015, 2016 and 2017 happiness dataset reflects 4, 5 and 3 clusters of countries respectively. We have also also listed down 5 countries from each clusters for all the three cases. Further we have conducted seriation analysis on the three datasets and represented the generated order of the countries. We found that all top ten countries rank highly on all the main features found to support happiness. We also conclude that unemployment causes a major fall in happiness, and even for those in work the quality of work can cause major variations in happiness. Moreover heat maps are also developed to better under the database.

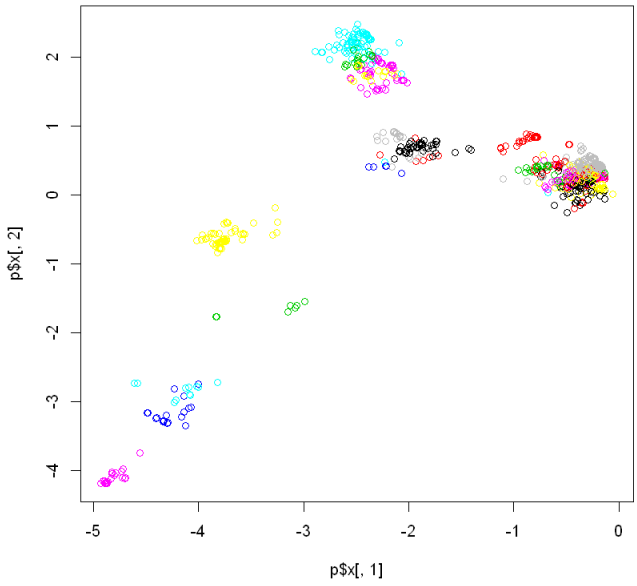
Question 2: Finding Groupings in Human's Mitochondrial SNP/Mutations Patterns

Principle component analysis

```
p = prcomp(data[, -1], center = FALSE, scale = FALSE)
p.var = p$sdev^2
pve = p.var / sum(p.var)
head(cumsum(pve))
```

```
0.159825680506812 0.232456519089664 0.300967930649964 0.34577995466343 0.386510550413912 0.419683381497639
```





M-Fold cross validation

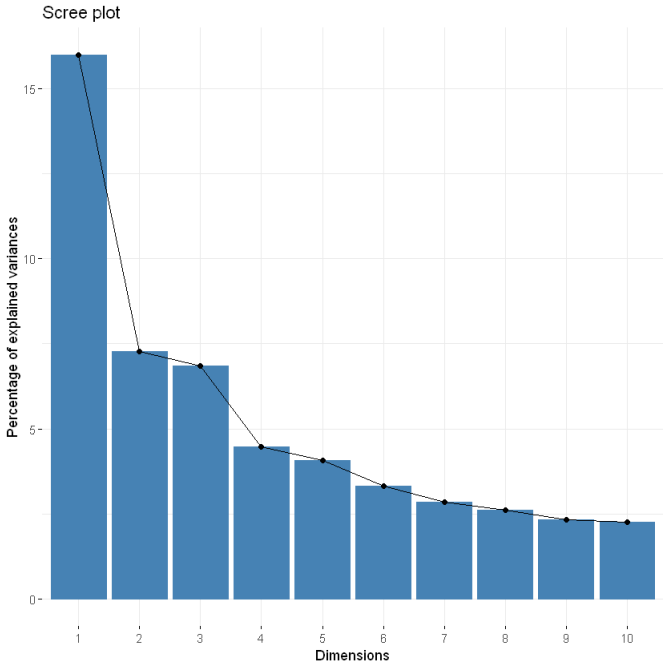
```
#We have taken m = 5 in this problem. We can also take any positive number for performing
this exercise.
nfolds = 5
folds <- cut(seq(1, nrow(kdata)), breaks = nfolds, labels = FALSE)
for (i in 1:nfolds)
    {
    testIndexes <- which(folds == i, arr.ind = TRUE)
    testData <- kdata[testIndexes, ]
    trainData <- kdata[-testIndexes, ]
```

```
    knn.pred = knn(trainData[, -1], testData[, -1], trainData$Group, k = 1)
    print (mean(knn.pred == testData$Group))
}
```

```
[1] 0.9860465 [1] 0.9860465 [1] 0.9813084 [1] 0.9581395 [1] 0.9767442
```

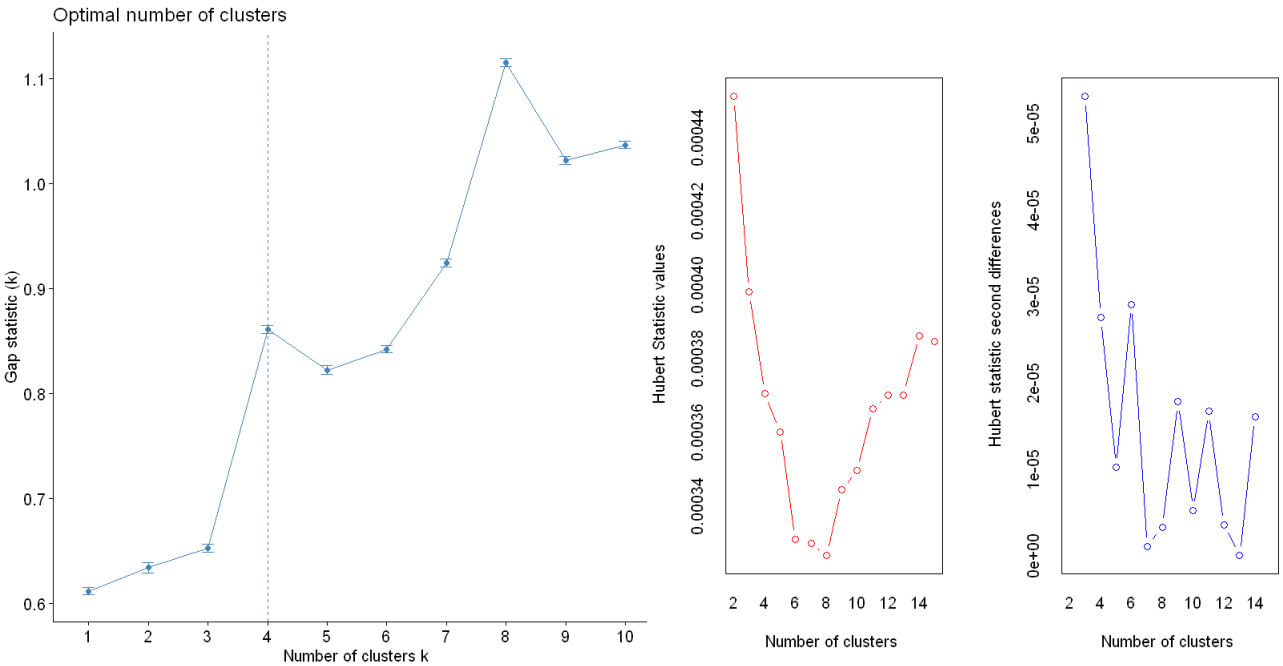
Clustering Analysis

fviz_eig(p)



Using NbClust

pcs = 18
fviz_nbclust(p$x[,1:pcs], kmeans, method = "gap_stat")

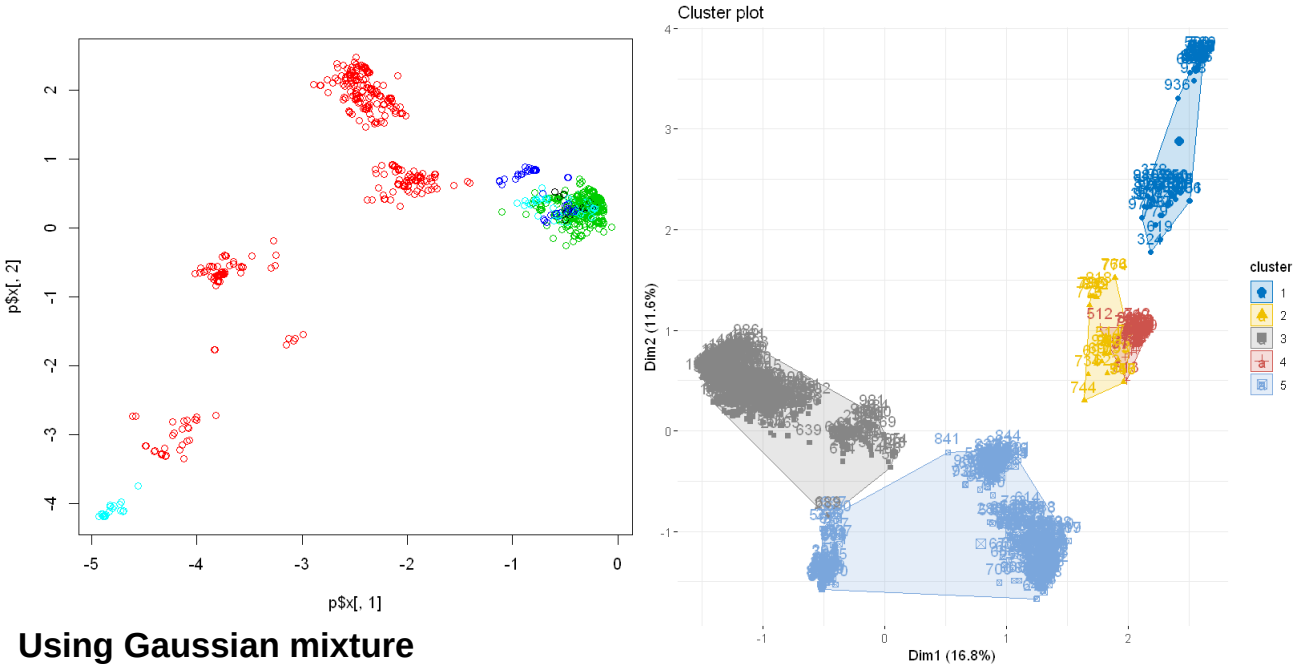


```
library(NbClust)
nb <- NbClust(p$x[,1:pcs], distance='euclidean', method='complete')
```

```
*******************************************************************
* Among all indices:
* 1 proposed 2 as the best number of clusters
* 3 proposed 3 as the best number of clusters
* 2 proposed 5 as the best number of clusters
* 6 proposed 6 as the best number of clusters
* 1 proposed 9 as the best number of clusters
* 1 proposed 11 as the best number of clusters
* 1 proposed 14 as the best number of clusters
* 6 proposed 15 as the best number of clusters
***** Conclusion *****
* According to the majority rule, the best number of clusters is 6
*******************************************************************
```

Using K-Means

```
#We have taken 10 Principle components and 5 clusters.
pcs = 10
clusters = 5
km <- kmeans(p$x[, 1:pcs], clusters)
plot(p$x[, 1], p$x[, 2], col = km$cluster)
#Cluster Plot. This will plot the first 2 variables only.
km.res = eclust(p$x[, 1:pcs], "kmeans", k = clusters, graph = F)
fviz_cluster(km.res, ggtheme = theme_minimal(), palette = 'jco')
```



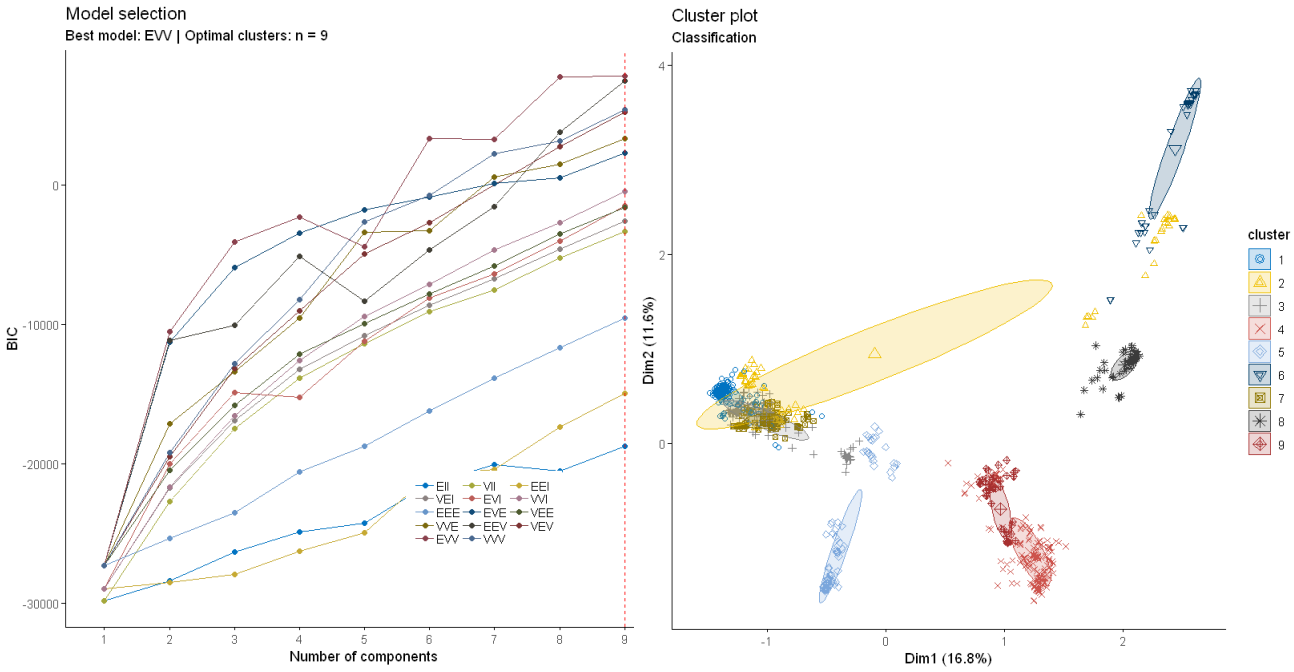
Using Gaussian mixture

```
citation("mclust")
library(mclust)
pcs = 10
mc <-Mclust(p$x[, 1:pcs])
summary(mc)
```

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------
Mclust EVV (ellipsoidal, equal volume) model with 9 components: log.likelihood n df BIC ICL 5929.687 1074 585
```

```
7776.575 7776.033
Clustering table: 1 2 3 4 5 6 7 8 9 203 100 136 224 114 42 113 63 79
```

#For model-based plot.
fviz_mclust(mc, "BIC", palette = "jco")
#For cluster based plot on classification
fviz_mclust(mc, "classification", geom = "point", palette = "jco")



Discussion

The given mitochondria dataset is studied using clustering methods. First we have used Principle component analysis and found that 347 and 545 principle components accounts for 95% and 98% variance respectively. Then we have produced some useful graphs to understand the dataset in a better way. Further we have used K-Means, Hierarchical, Gaussian mixture and DB Scan clustering methods on the given dataset and knn method to access the accuracy of the model on the dataset.

Question 3: Text-Mining the Bible

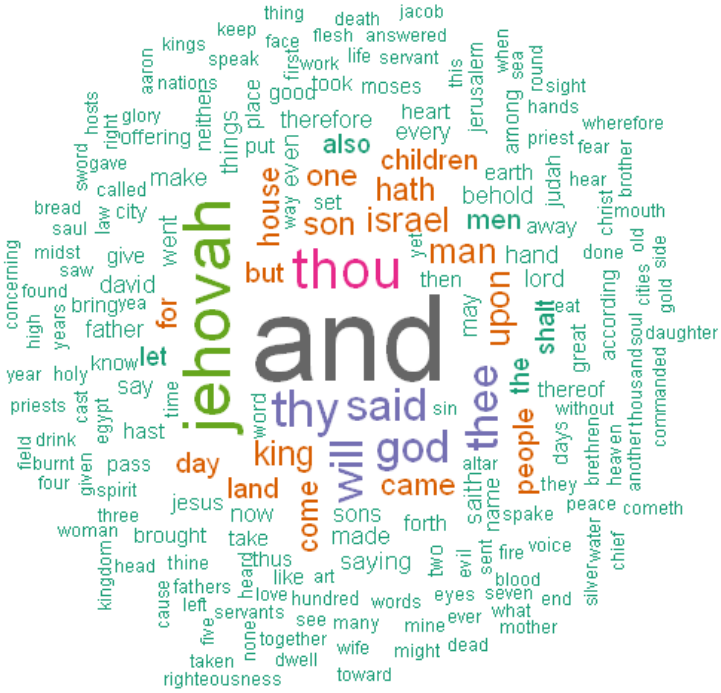
#Just to see a particular chapter
print (text.Chapter[1180])

```
[1] "and he stood upon the sand of the sea. And I saw a beast coming up out of the sea, having ten horns, and
seven heads, and on his horns ten diadems, and upon his heads names of blasphemy. And the beast which I saw was
like unto a leopard, and his feet were as `the feet' of a bear, and his mouth as the mouth of a lion: and the
dragon gave him his power, and his throne, and great authority. And `I saw' one of his heads as though it had
been smitten unto death; and his death-stroke was healed: and the whole earth wondered after the beast; and
they worshipped the dragon, because he gave his authority unto the beast; and they worshipped the beast,
saying, Who is like unto the beast? And who is able to war with him? and there was given to him a mouth
speaking great things and blasphemies; and there was given to him authority to continue forty and two months.
And he opened his mouth for blasphemies against God, to blaspheme his name, and his tabernacle, `even' them
that dwell in the heaven. And it was given unto him to make war with the saints, and to overcome them: and
there was given to him authority over every tribe and people and tongue and nation. And all that dwell on the
earth shall worship him, `every one' whose name hath not been written from the foundation of the world in the
book of life of the Lamb that hath been slain. If any man hath an ear, let him hear. If any man `is' for
captivity, into captivity he goeth: if any man shall kill with the sword, with the sword must he be killed.
Here is the patience and the faith of the saints. And I saw another beast coming up out of the earth; and he
had two horns like unto lamb, and he spake as a dragon. And he exerciseth all the authority of the first beast
in his sight. And he maketh the earth and them dwell therein to worship the first beast, whose death-stroke was
healed. And he doeth great signs, that he should even make fire to come down out of heaven upon the earth in
the sight of men. And he deceiveth them that dwell on the earth by reason of the signs which it was given him
to do in the sight of the beast; saying to them that dwell on the earth, that they should make an image to the
beast who hath the stroke of the sword and lived. And it was given `unto him' to give breath to it, `even' to
the image to the breast, that the image of the beast should both speak, and cause that as many as should not
worship the image of the beast should be killed. And he causeth all, the small and the great, and the rich and
the poor, and the free and the bond, that there be given them a mark on their right hand, or upon their
forehead; and that no man should be able to buy or to sell, save he that hath the mark, `even' the name of the
beast or the number of his name. Here is wisdom. He that hath understanding, let him count the number of the
beast; for it is the number of a man: and his number is Six hundred and sixty and six."
```

Frequency of Words

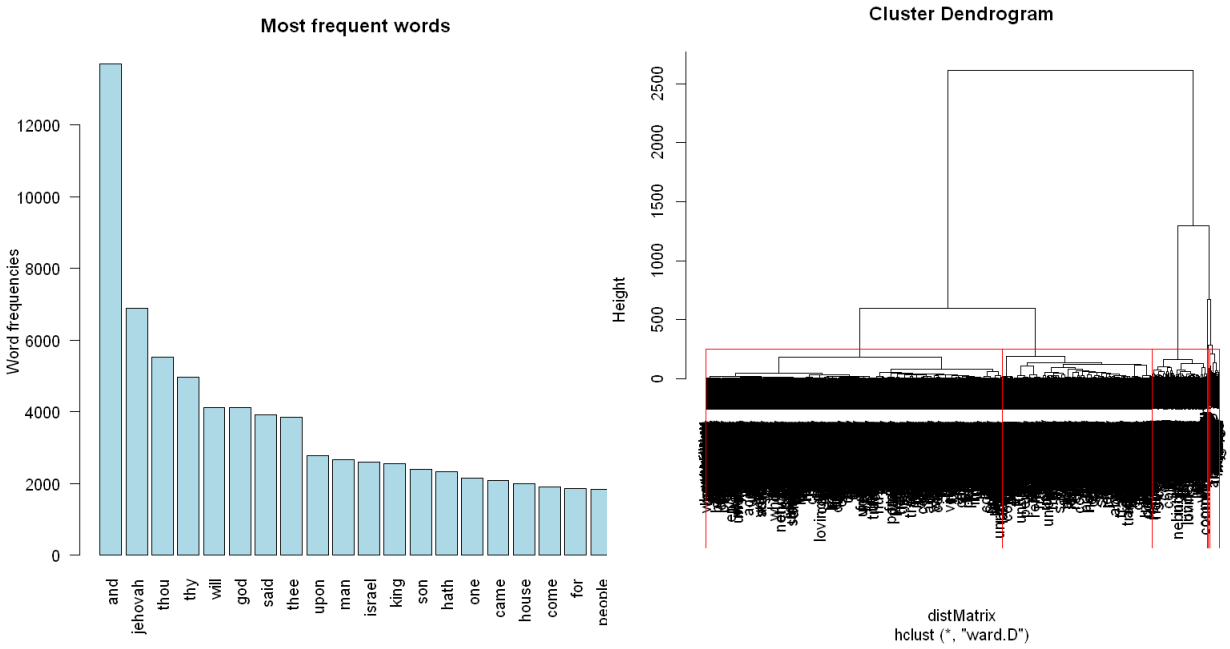
docs <- tm_map(docs, removeWords, mystopwords)
dtm <- TermDocumentMatrix(docs)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
head(d, 10)

word	freq	
and	and	13707
jehovah	jehovah	6888
thou	thou	5513
thy	thy	4956
will	will	4117
god	god	4112
said	said	3910
thee	thee	3844
upon	upon	2777
man	man	2670



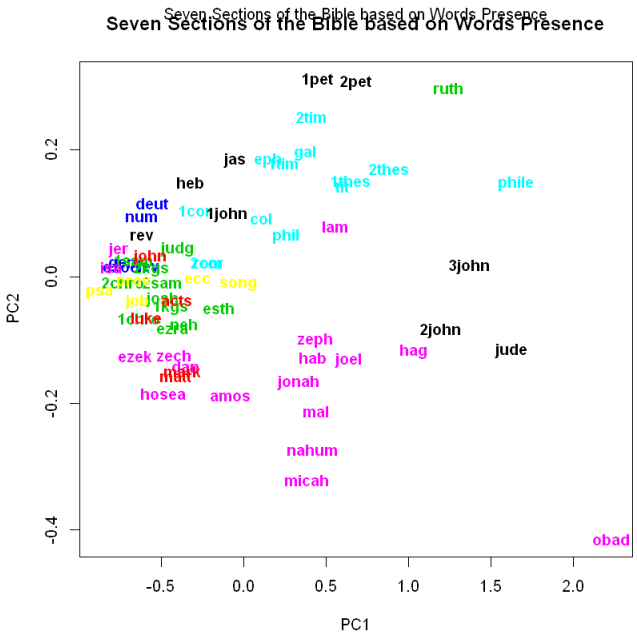
Cluster Dendogram

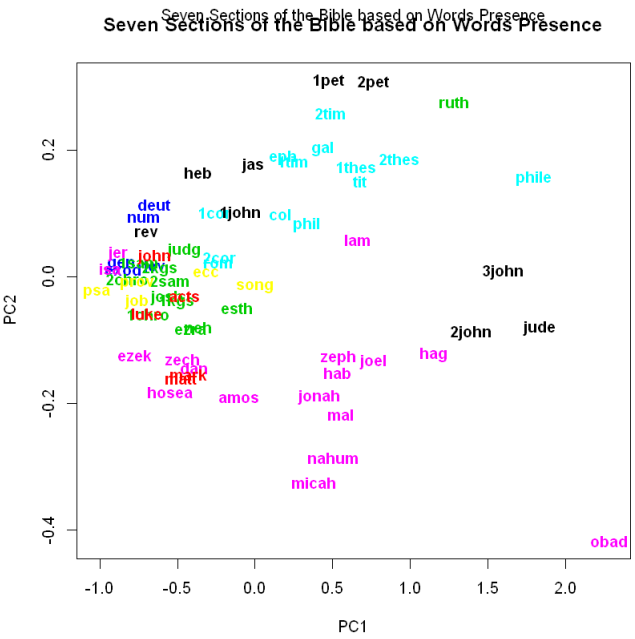
```
tdm2 <- removeSparseTerms(tdm, sparse = 0.95)
m2 <- as.matrix(tdm2)
distMatrix <- dist(scale(m2))
fit <- hclust(distMatrix, method = "ward")
plot(fit)
rect.hclust(fit, k = 6)
```



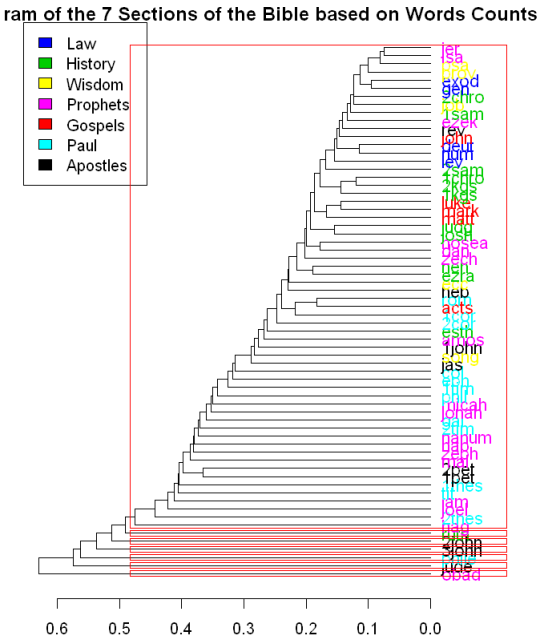
Using Distance Matrix

```
for (i in 1:7){
dtm = dtm.ngrams[[i]]
###################
# ## Use term raw Frequency counts
## Calculate document-to-document cosine similarity (scalar product)
csim <- dtm / sqrt(rowSums(dtm*dtm))
csim <- csim %*% t(csim)
# Turn that cosine similarity matrix into a distance matrix
dist.mtx <- 1-csim
#PCA plot
fit.pca <- prcomp(as.dist(dist.mtx))
plot(fit.pca$x[,1:2], type='n',main="Seven Sections of the Bible based on Words Presence")
text(x = fit.pca$x[,1], y = fit.pca$x[,2], labels = row.names(fit.pca$x),
col=unclass(as.factor(ASV_Books$Sections)), cex=.95, font=2)
mtext( cex = 1, text = "Seven Sections of the Bible based on Words Presence",
      line=2,
      outer=FALSE)
}
```



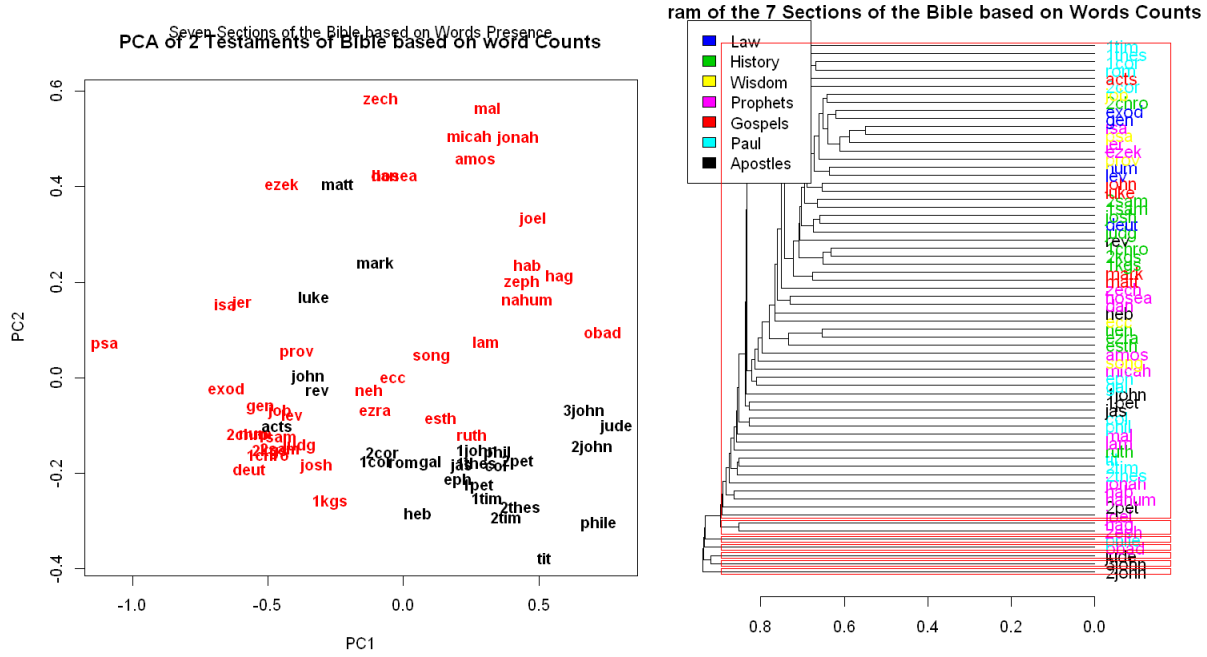



```
dend=as.dendrogram(hc.avg)
# Coloring the leaves according to 'Sections'
labels_colors(dend) <- as.numeric(as.factor(ASV_Books$Sections[hc.avg$order]))
#Change labels font size
dend <- set(dend, "labels_cex", 1.12)
par(mar = c(4,1,1,12))
plot(dend, horiz = TRUE, main='Dendrogram of the 7 Sections of the Bible based on Words
Counts ')
legend("topleft", legend = unique(ASV_Books$Sections), fill
as.numeric(as.factor(unique(ASV_Books$Sections))))
rect.dendrogram(dend, k=7, border="red", horiz=T)
```



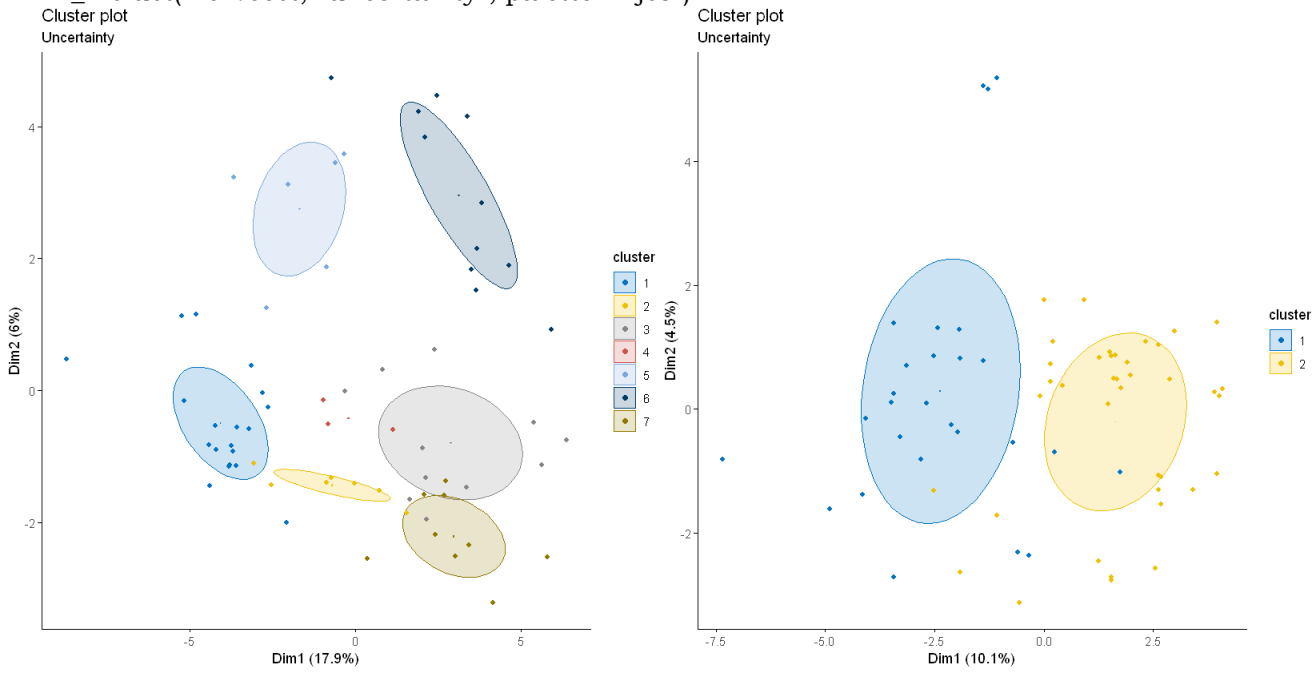
PCA Analysis on the basis of Old and New testament

```
#PCA plot
plot(fit.pca$x[,1:2], type='n', main="PCA of 2 Testaments of Bible based on word Counts")
text(x = fit.pca$x[,1], y = fit.pca$x[,2], labels = row.names(fit.pca$x),
col=unclass(as.factor(ASV_Books$Testaments)), cex=.95, font=2)
mtext( cex = 1, text = "Seven Sections of the Bible based on Words Presence",
       line=2,
       outer=FALSE)
```



Gaussian Mixture

```
fviz_mclust(mc.best, "uncertainty", palette = "jco")
fviz_mclust(mcT.best, "uncertainty", palette = "jco")
```



Discussion

We collapse the bible book into 66 books of Old testament and New testament and further into corresponding chapters. Then we have done a thorough book analysis of bible based on 7 different sections. The wordcloud suggests that "and" is the most frequently used word followed by "jehovah". We have used a few classification methods for text mining. The results indicates that there is a strong dissimilarity between old testament and new testament and between sections of the two testaments of the bible.

Thanks