**Question 0 : Predicting Job Salary**

Dataset is imported which is totally 10000 of rows and 12 columns.

**Cleaning Dataset**

print (salary.shape)

print (salary.columns)

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

column 'Title', 'ContractType', 'ContractTime' and 'Company' has null values.

print (salary["Title"].value\_counts())

print (salary["ContractType"].value\_counts())

print (salary["ContractTime"].value\_counts())

print (salary["Company"].value\_counts())

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

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

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

salary["ContractType"].fillna(value = 'full\_time', inplace=True)

import matplotlib.pyplot as plt

import numpy as np

fig = plt.figure()

ax = plt.axes()

x = salary['ContractTime']

y = salary['SalaryNormalized']

rng = np.random.RandomState(0)

colors = rng.rand(100)

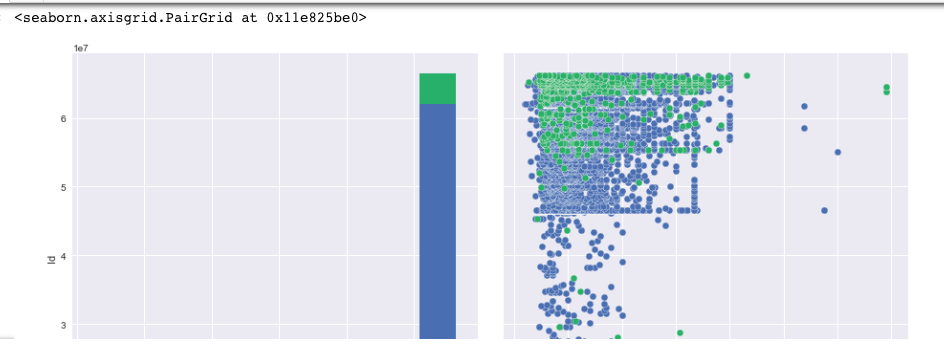
sizes = 1000 \* rng.rand(100)

plt.scatter(x, y)

%matplotlib inline

import seaborn as sns; sns.set()

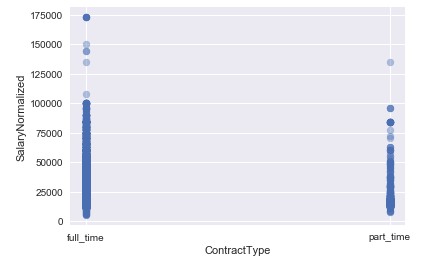
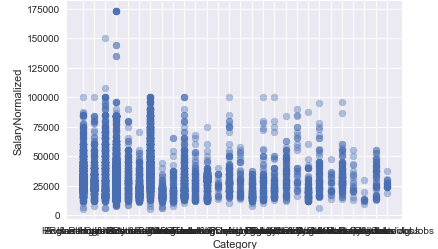
sns.pairplot(salary, hue = 'ContractTime', size = 6.5)



plt.scatter(salary['ContractType'], salary['SalaryNormalized'], alpha=0.4)

plt.xlabel('ContractType')

plt.ylabel('SalaryNormalized');

plt.scatter(salary['Category'], salary['SalaryNormalized'], alpha=0.4)

plt.xlabel("Category")

plt.ylabel('SalaryNormalized');

After splitting data into test and training datasets I have applied various techniques and analysis which are shown below:

**Multiple Linear Regression**

reg = linear\_model.LinearRegression()

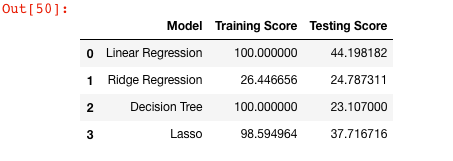
model = reg.fit(xtrain,ytrain)

predictions = reg.predict(xtest)

print(predictions)[0:10]



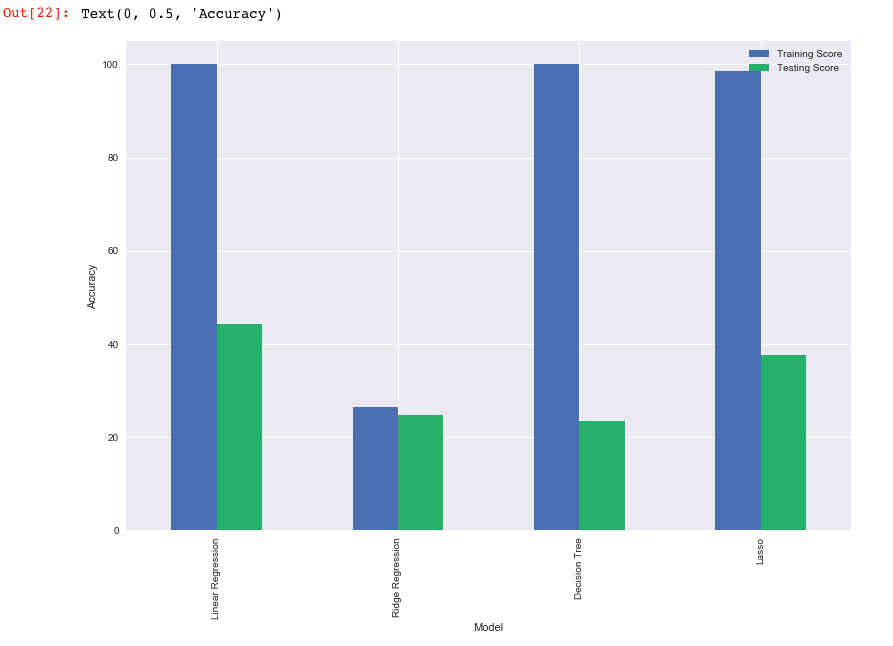
* **Accuracy** of Training and test-set score was found 100.00 and 44.20 respectively.
* Training and test-set score of **Ridge regression** was found 26.45 and 24.79 respectively.
* **Mean squared error and Variance-score** was found 146367769.05 and 0.44 respectively.
* Training and test-set score of **Decision Tree** was observed 100.0 and 23.107 respectively.
* Training and test-set score of **Lasso** was observed 98.59 and 37.72 respectively where number of features used are 3856.



results\_table.plot(kind = 'bar', x = 'Model', figsize = (9, 13))

plt.xticks(rotation = 'vertical')

plt.ylabel('Accuracy')



**Conclusion:**

we have developed 3 models which can be used for analysis of the given dataset.The best model based on their training and test data is the Regression and Multiple Linear.

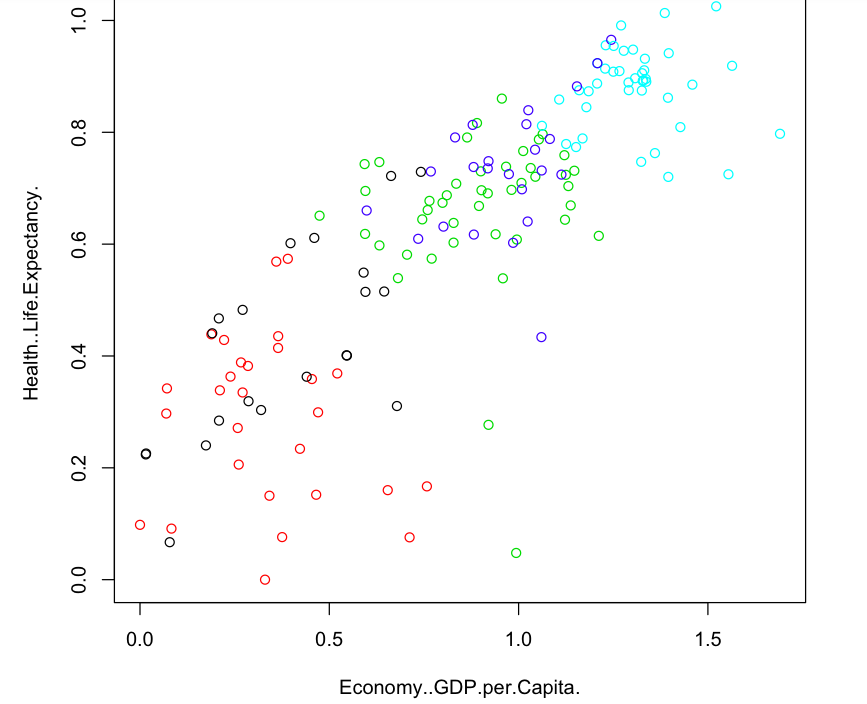
**Question 1: Exploring World Happiness**

Data is imported which have these following attributes.



* **K-Means Clustering on Happiness 2015 Dataset**

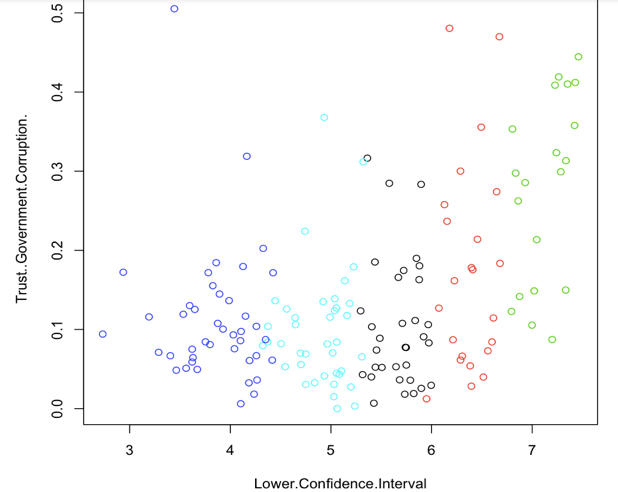
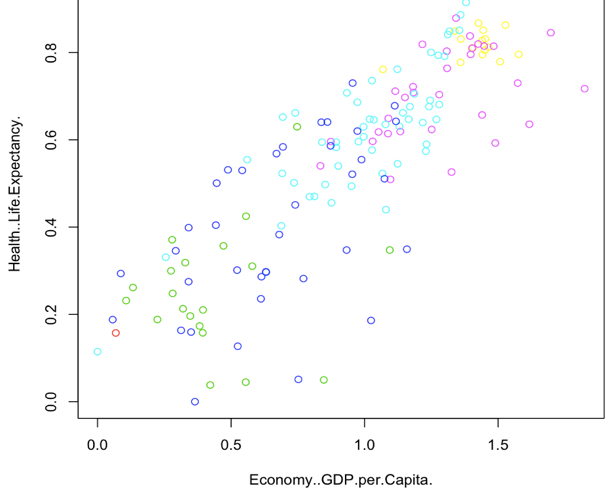
K-means clustering with 5 clusters of sizes is 21, 28, 44, 24, 41.

****

**Fig:** Plotting of the clusters based on k-means analysis with Health life Expectancy and Economy GDP Per capita attributes

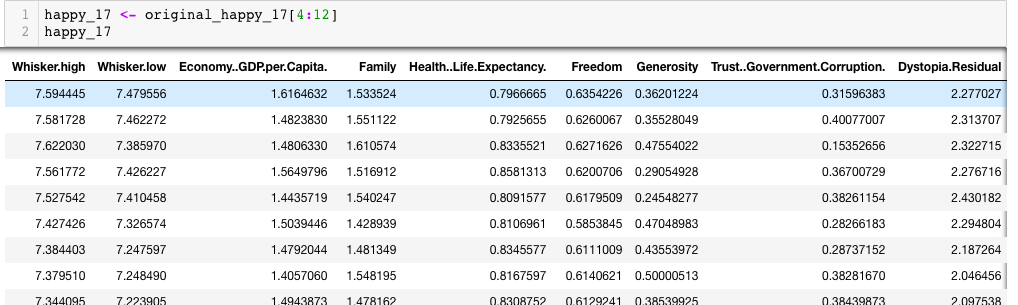
* **K-Means clustering on Happiness-2016 Dataset.**

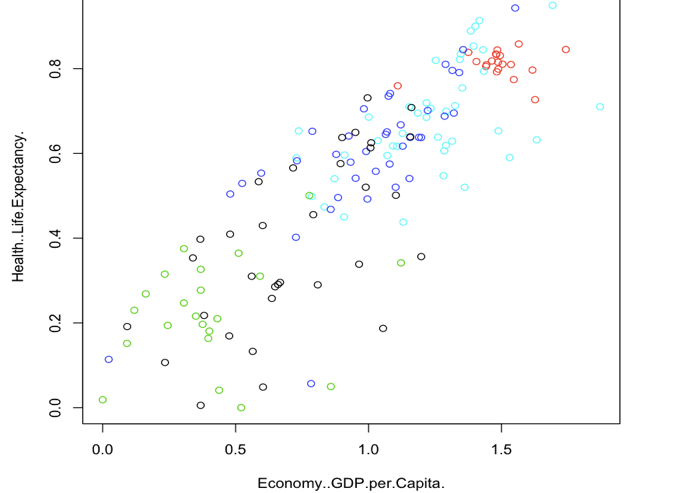
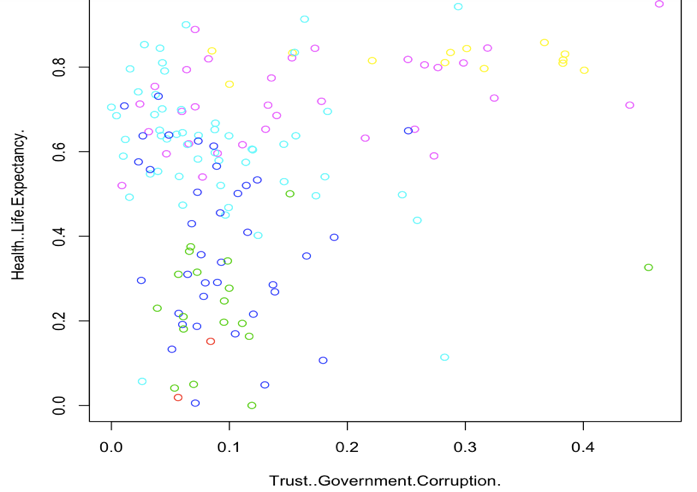
****

** **

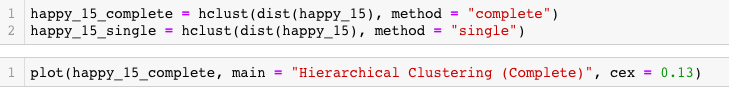
**Fig:** Above clustering analysis done based on different attributes between Trust.Government.corruption v/s Lower.confidence.Interval and Health.Life.Expectancy v/s Economy..GDP.per.Capita respectively.

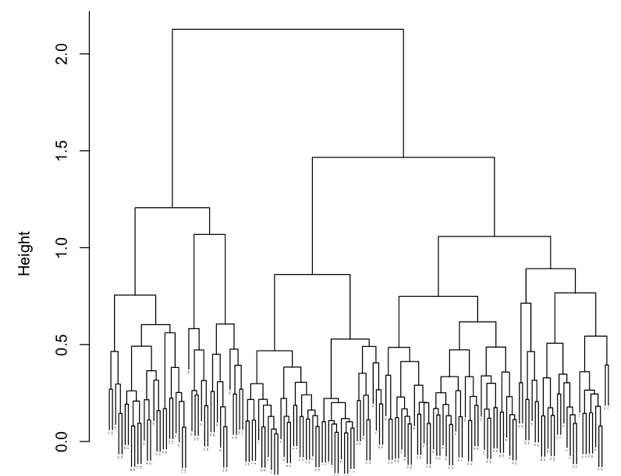
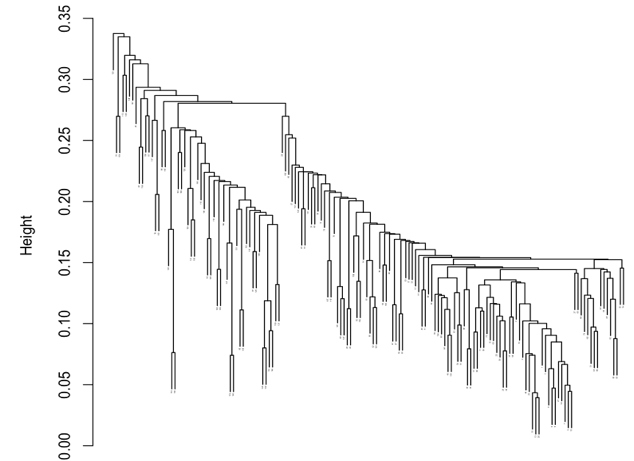
* **K-Means clustering on Happiness-2017 Dataset.**

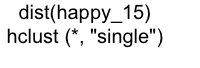
****

****

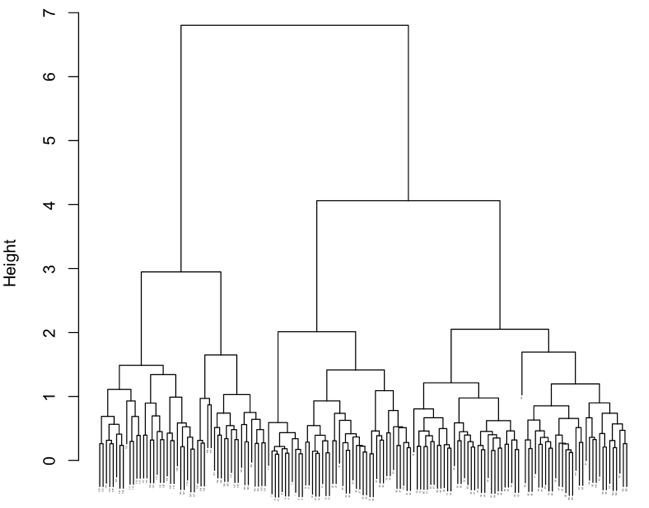
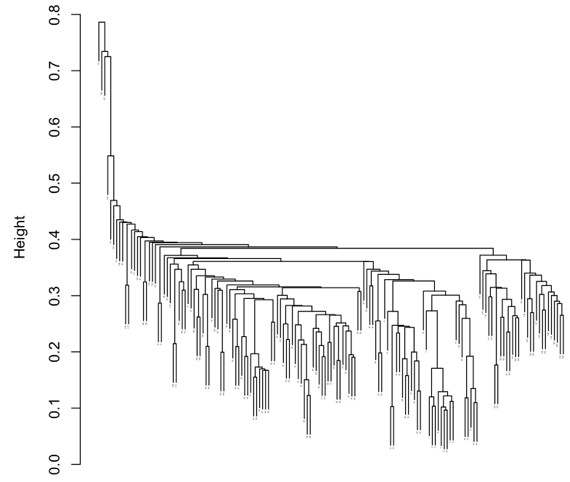
* **Hierarchical Clustering on Happiness-2015 Dataset.**

****

** **

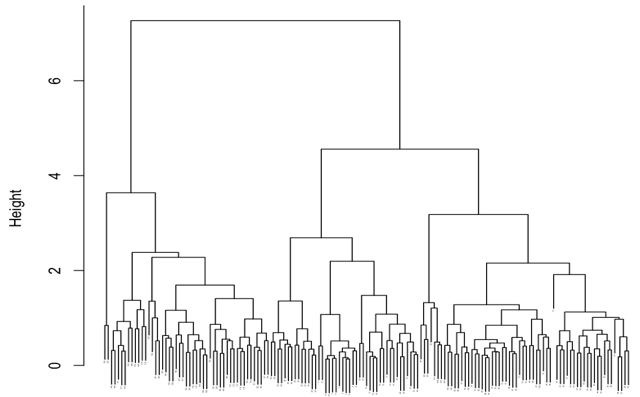
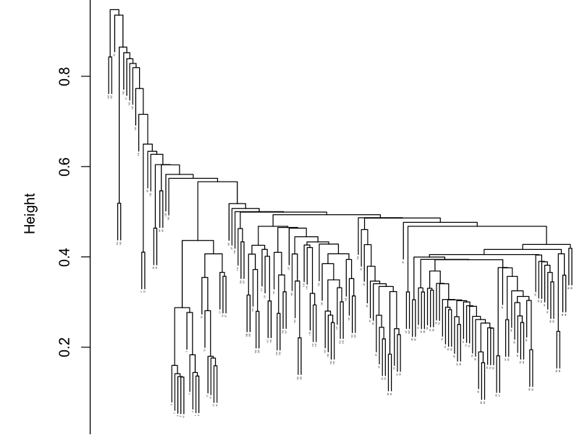
** **

* **Hierarchical Clustering on Happiness-2016 Dataset.**

 ****

** **

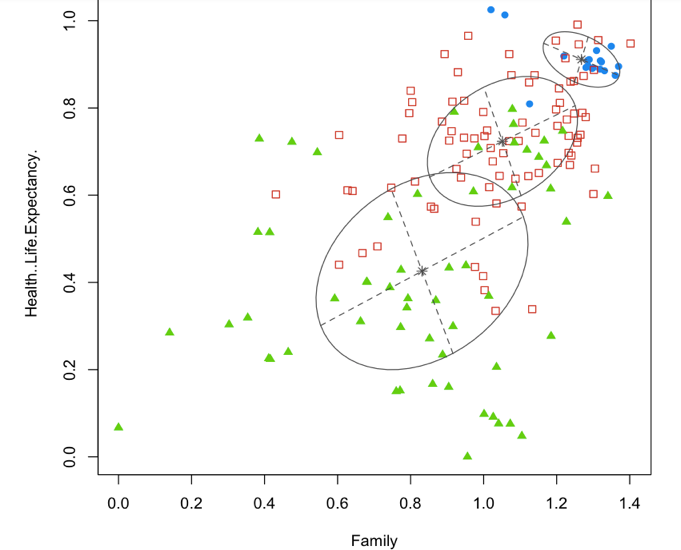
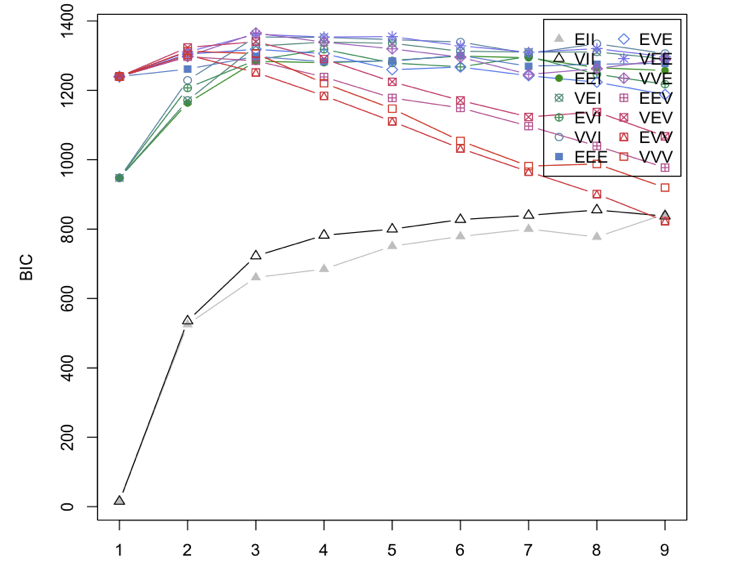
* **Hierarchical Clustering on Happiness-2017 Dataset.**

** **

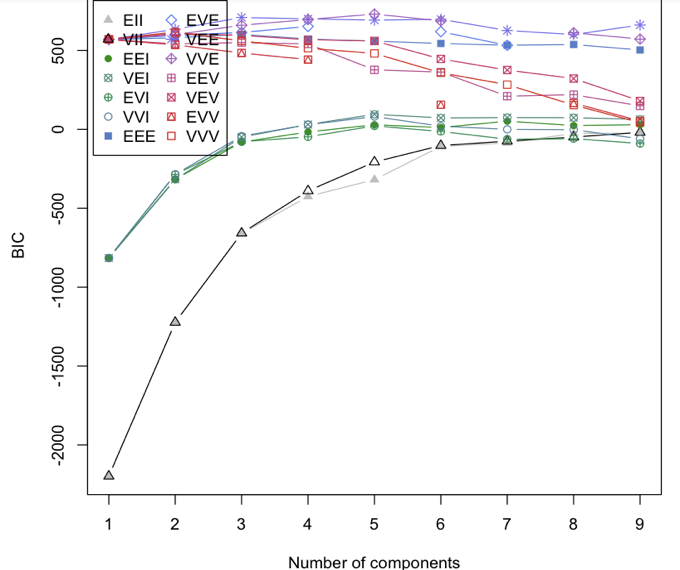
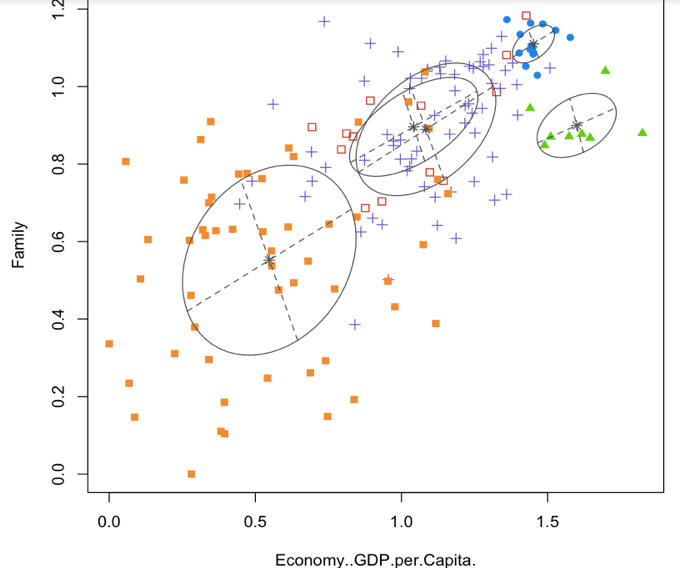
* **Gaussian Mixture on Happiness-2015 dataset.**

****

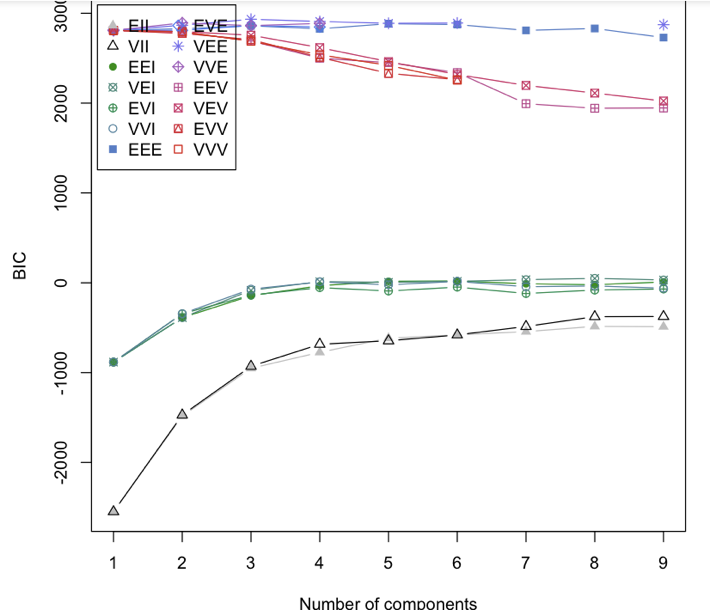
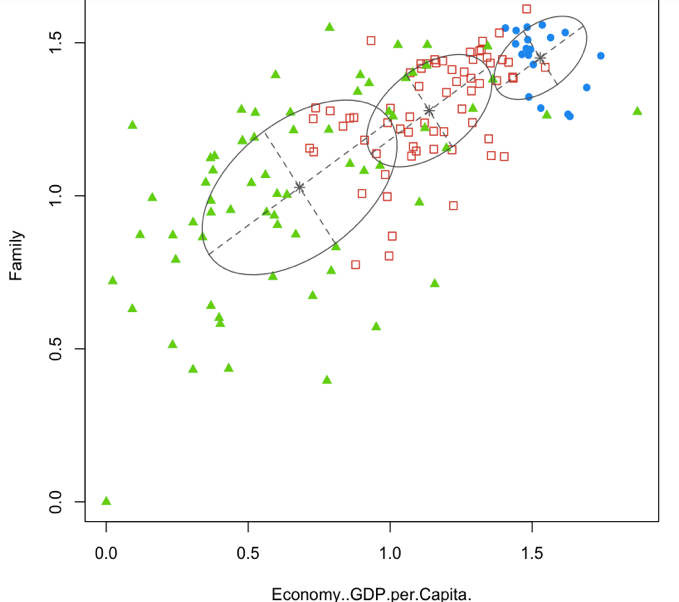
Fig: cluster\_1 countries are the top countries in happiness index whereas cluster\_4 countries are the last countries in happiness index.



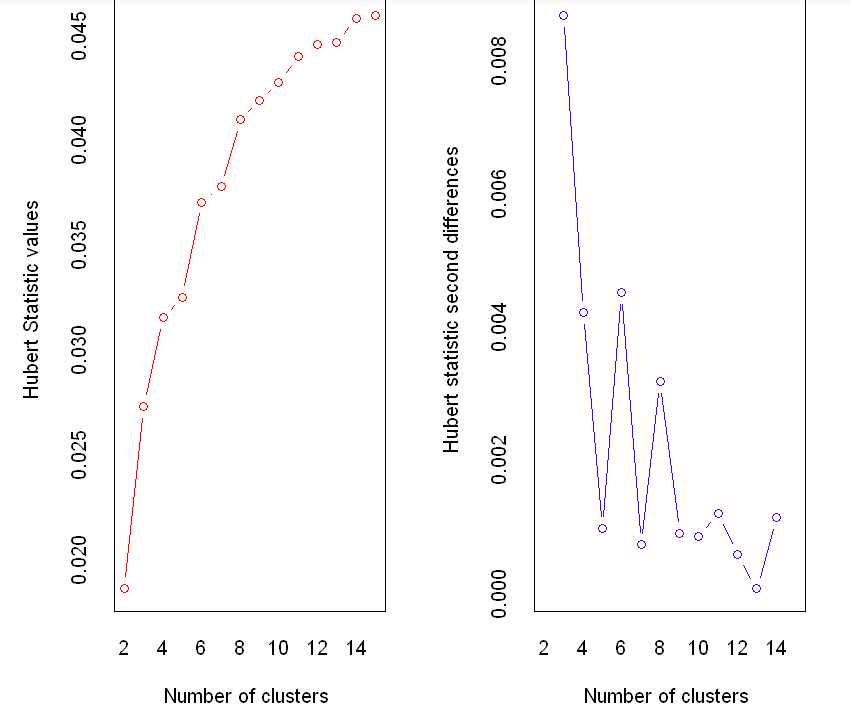
* **Gaussian Mixture on Happiness-2016 dataset.**

 ****

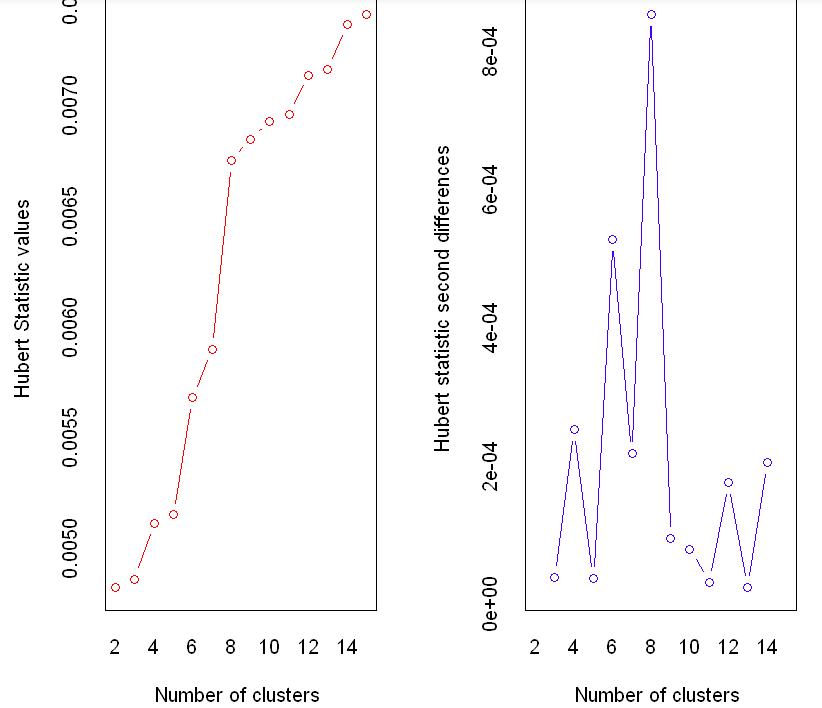
* **Gaussian Mixture on Happiness-2017 dataset.**

 ****

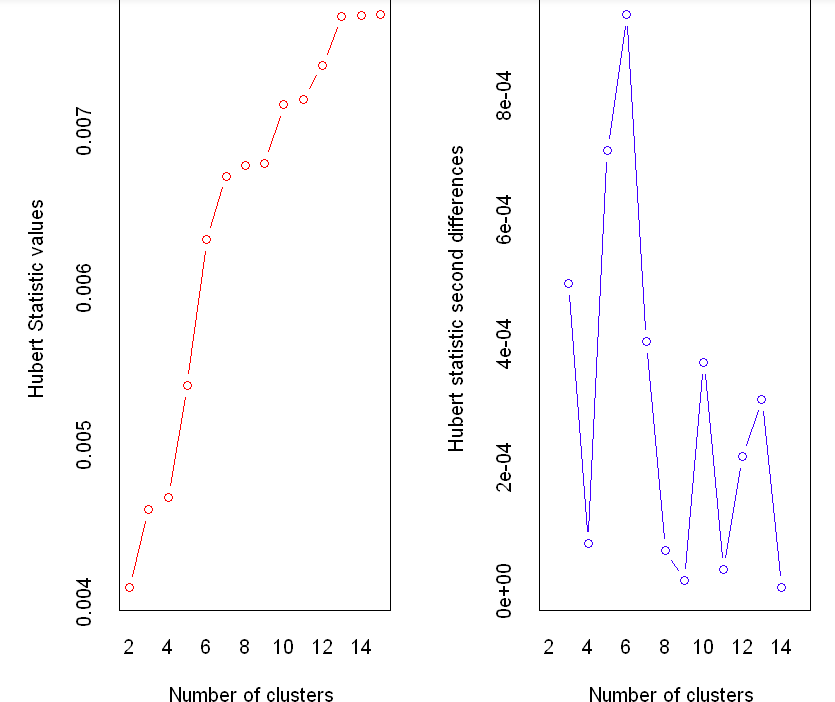
* **NbClust on Happiness-2015 dataset**



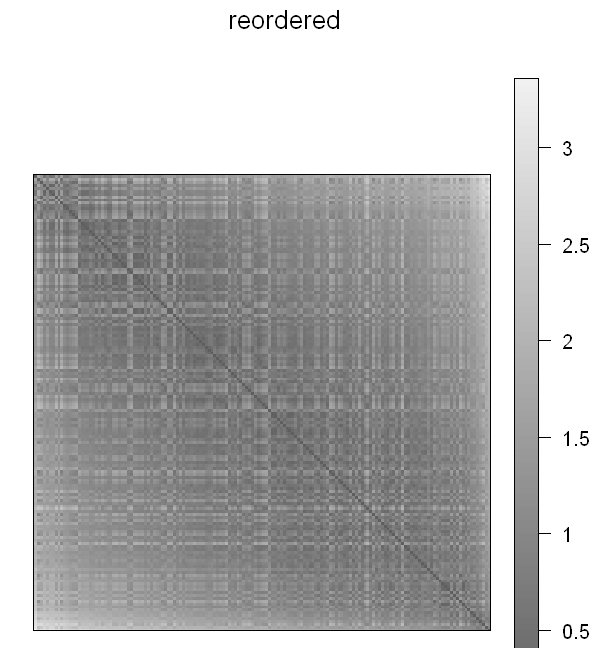
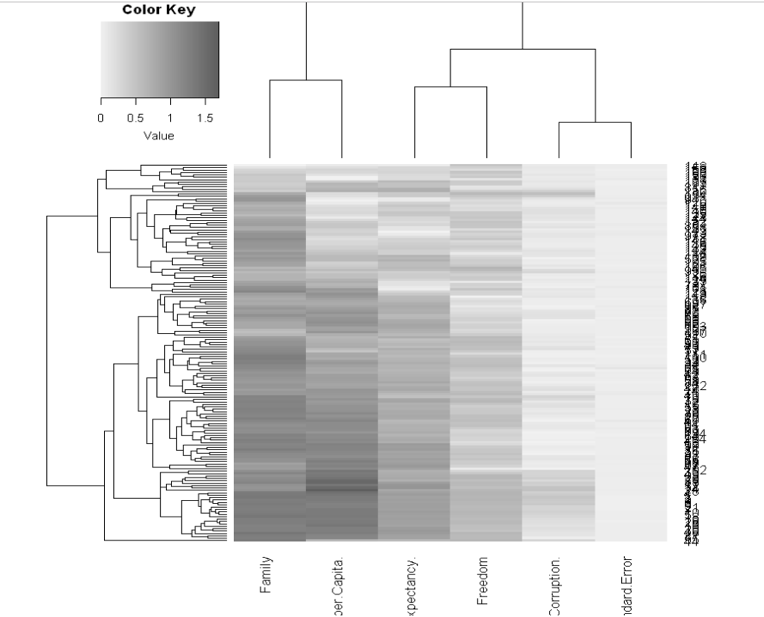
* **NbClust on Happiness 2016 dataset**

****

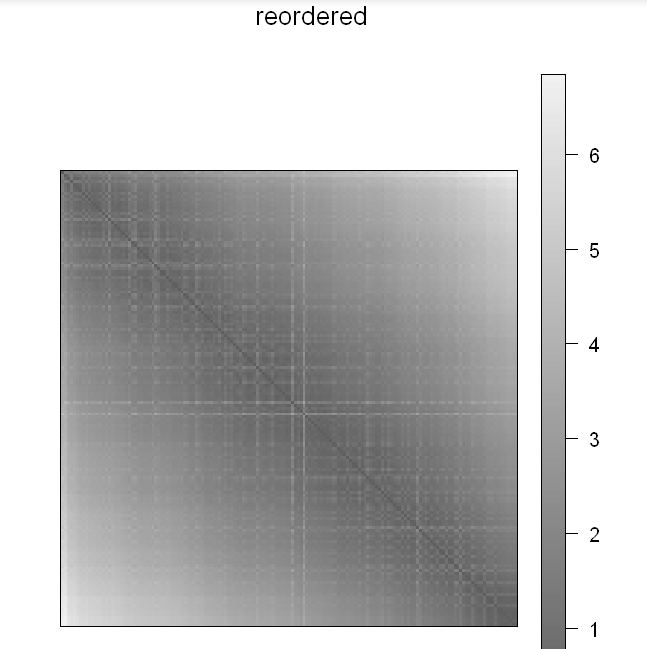
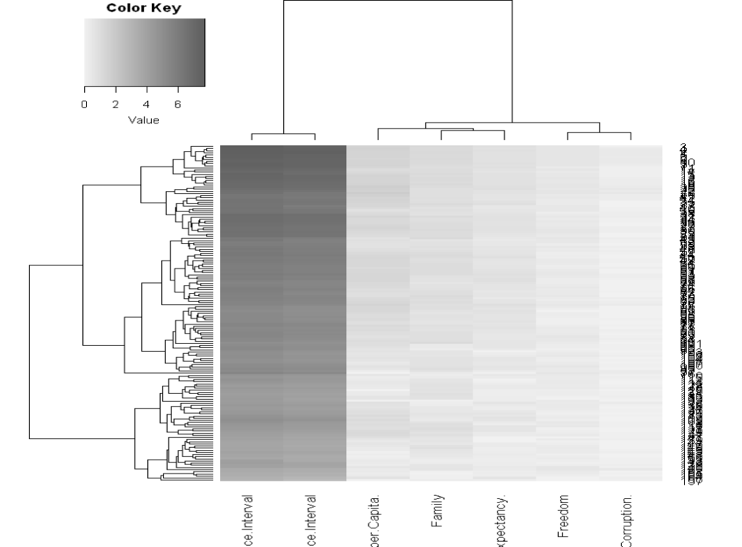
* **NbClust on Happiness 2017 dataset**

****

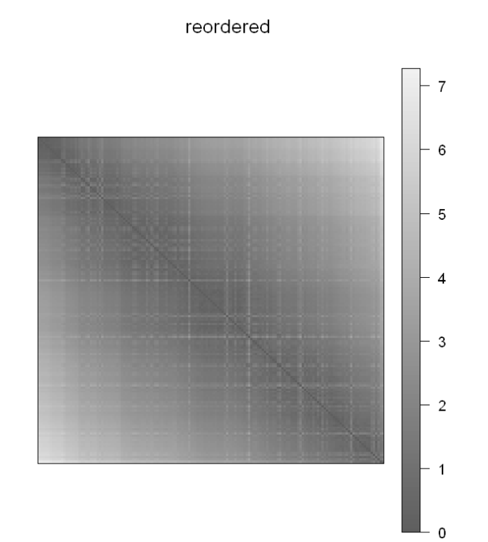
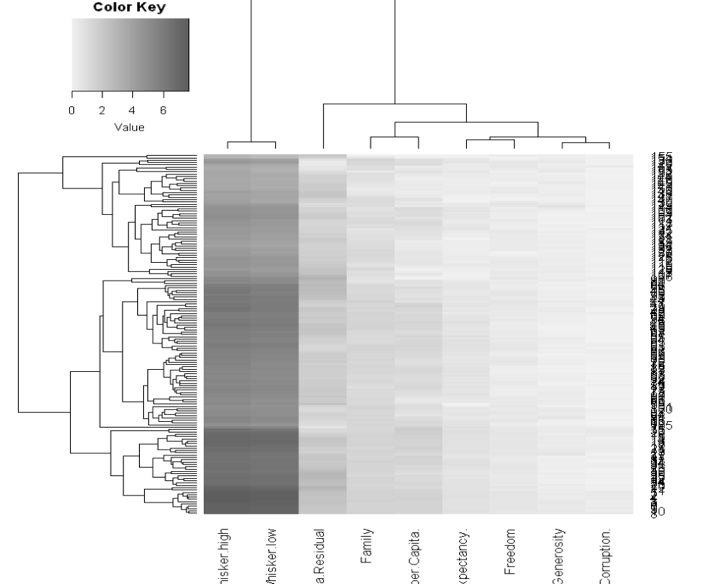
* **Seriation analysis on 2015 dataset**

 ****

* **Seriation analysis on 2016 dataset**

 ****

* **Seriation analysis on 2017 dataset**

** **

* **Conclusion**

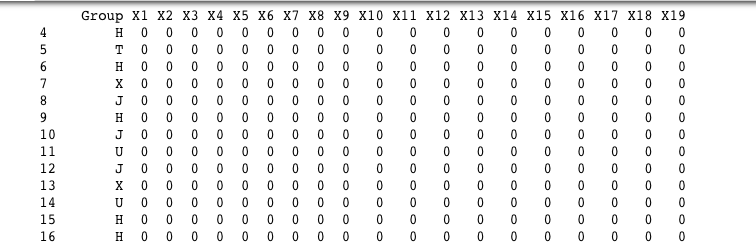
After conducting clustering analysis on the happiness data for 2015, 2016 and 2017 years. I listed down 5 countries from each clusters for all the three cases and I shown best method (VEE, VVE etc) for each dataset in order to understand various features in the dataset. We found that all top ten countries rank highly on all the main features found to support happiness after coducting seriation analysis. we can conclude that unemployment causes a major fall in happiness.

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

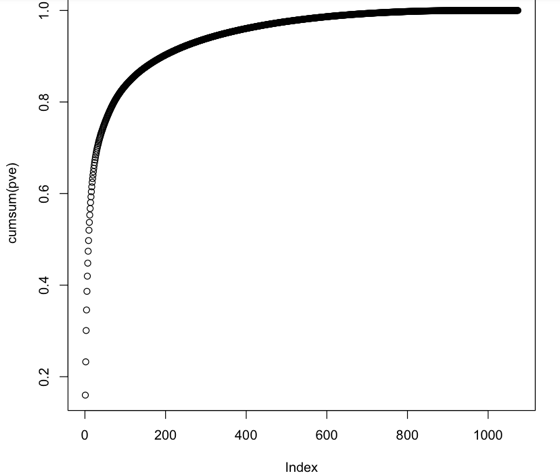
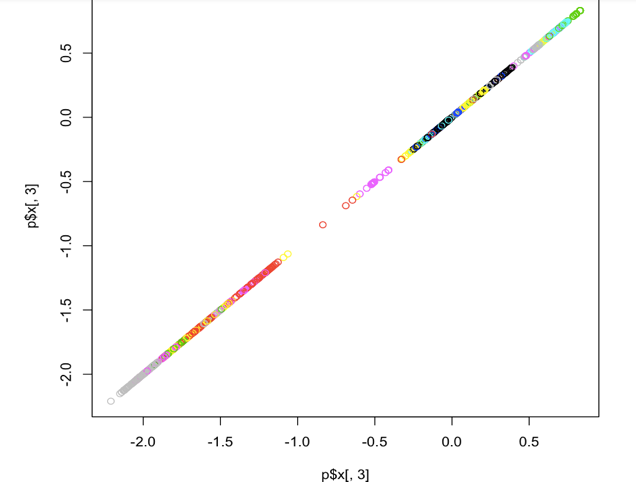
data = Mt

data$Group <- as.factor(data$Group)

print (data)



* **Principle component analysis**

 ****

* **M-Fold cross validation**

nfolds = 9

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, ]

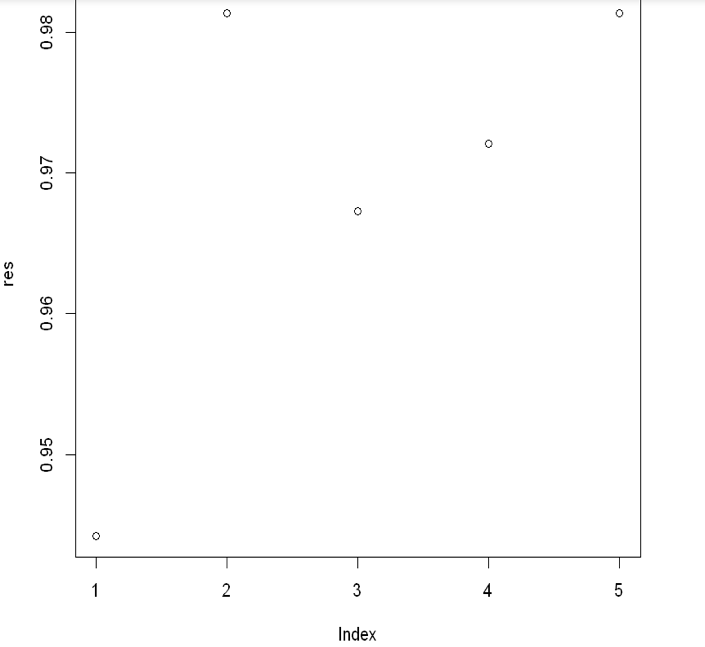
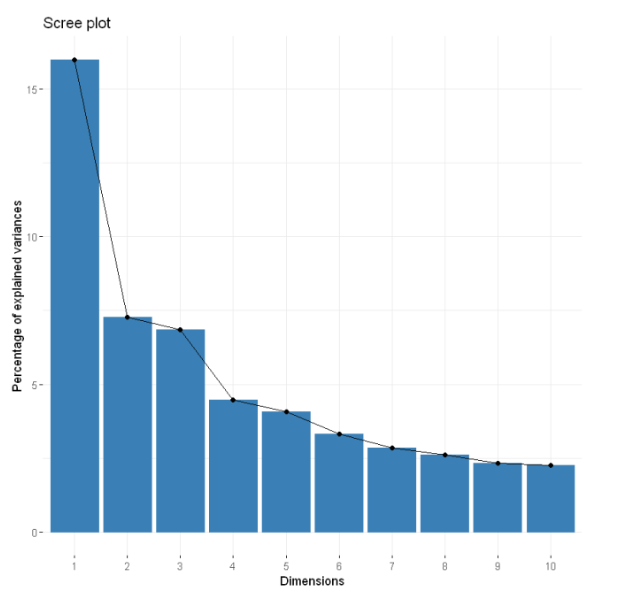
knn.pred = knn(trainData[, -1], testData[, -1], trainData$Group, k = 1)

print (mean(knn.pred == testData$Group))

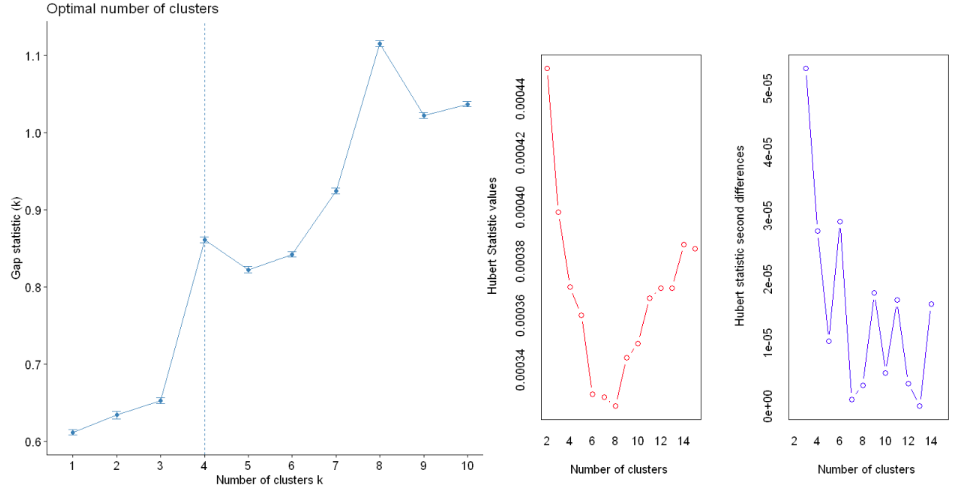
}

Tested with nfolds=9

* **Clustering Analysis**

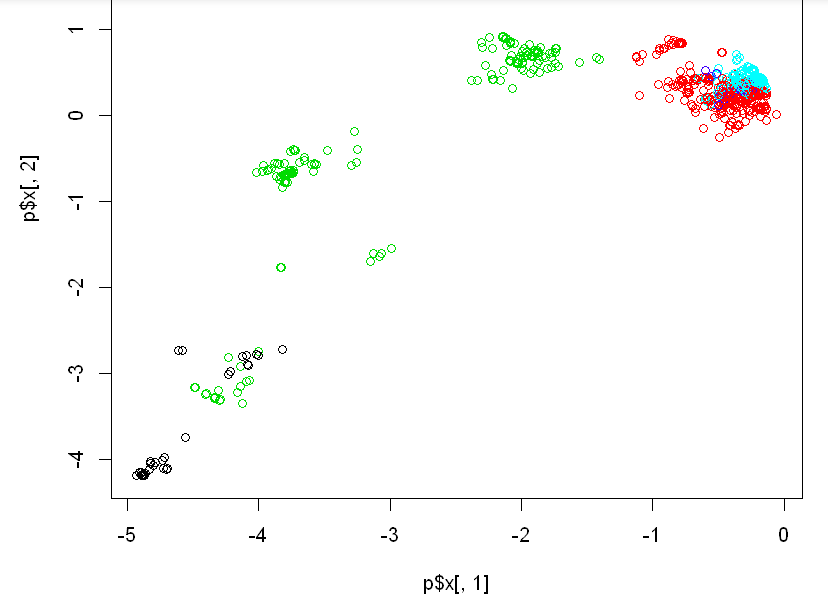
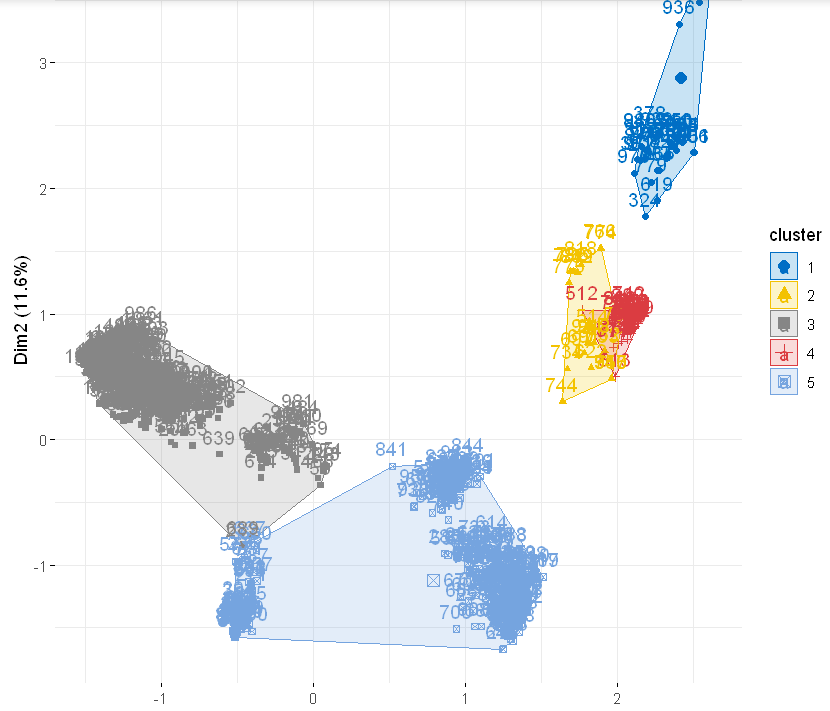


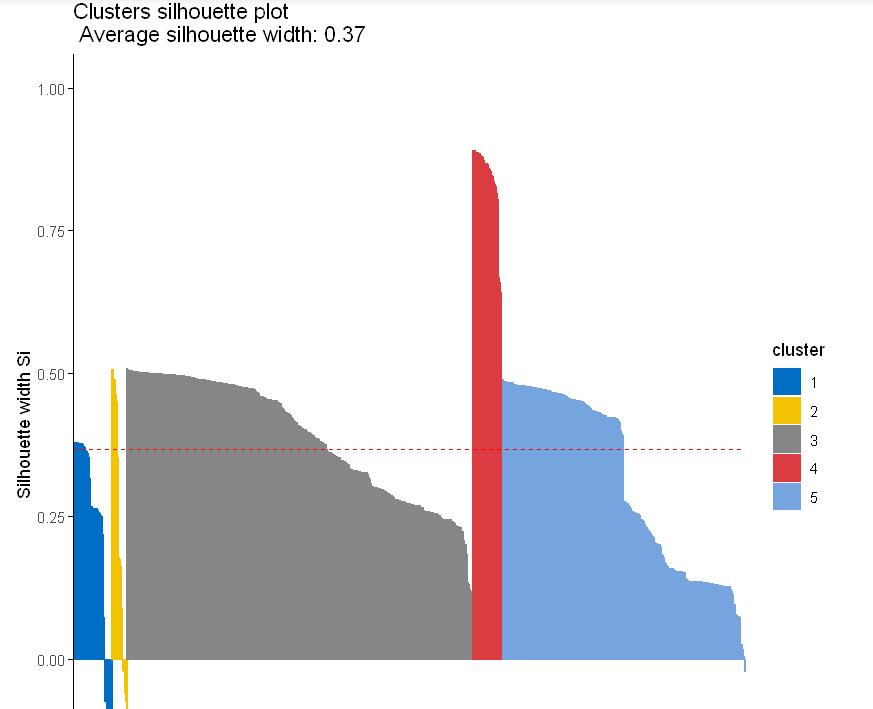
* **Using NbClust**



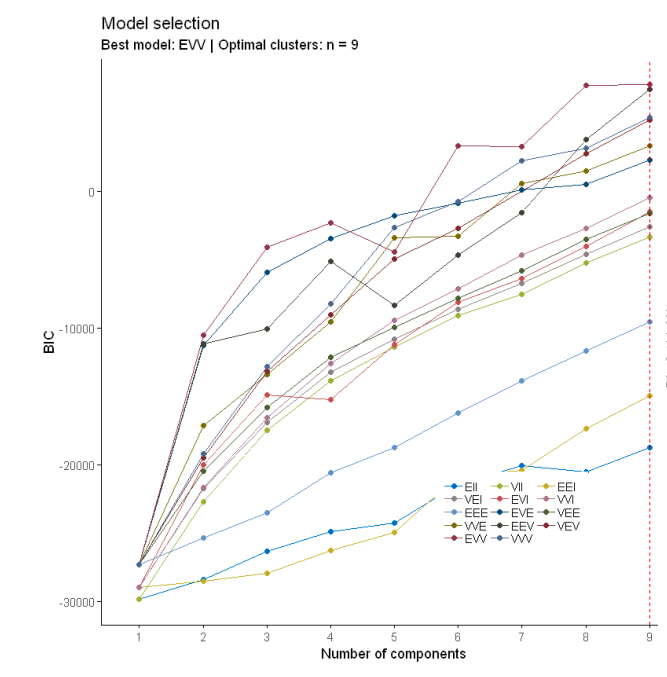
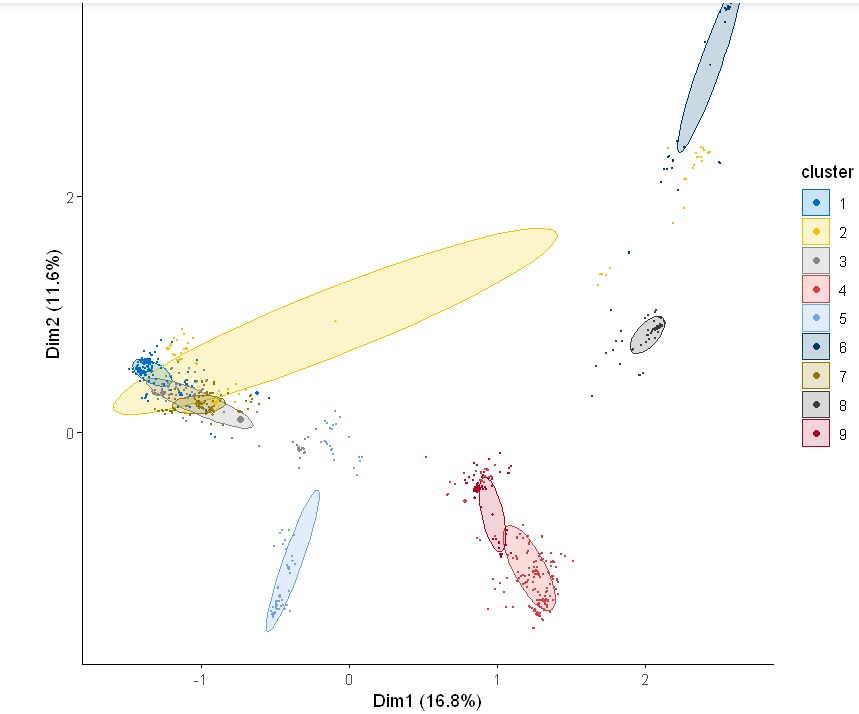
* **Using K-Means**

11 principle components had taken with 5 clusters.

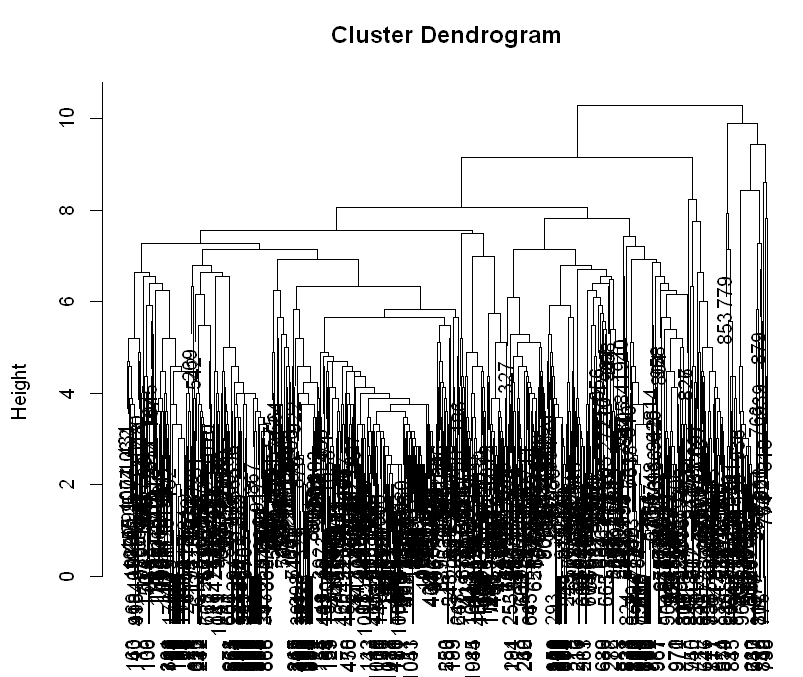
 



* **Using Gaussian mixture**

 ****

* **Hierarchical clustering**

****

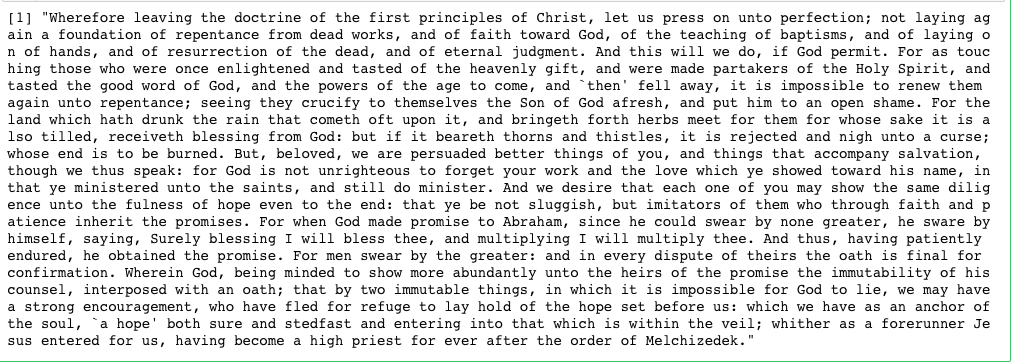
* **Conclusion**

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 wgere 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.

**Question 3: Text-Mining the Bible**

Printing the paragraph

print (text.Chapter[1139])



The file consists of 8 columns (as shown above). The 'text' column contains relevant text that we should take into consideration.

* **Frequency of Words**

myCorpus <- Corpus(VectorSource(ASV\_Books$text))

mystopwords <- c(stopwords('english'), "and", "unto", "shall")

myCorpus <- tm\_map(myCorpus, removeWords, mystopwords)

tdm <- TermDocumentMatrix(myCorpus, control = list(wordLengths = c(1, Inf)))

freq.terms <- findFreqTerms(tdm, lowfreq = 15)

term.freq <- rowSums(as.matrix(tdm))

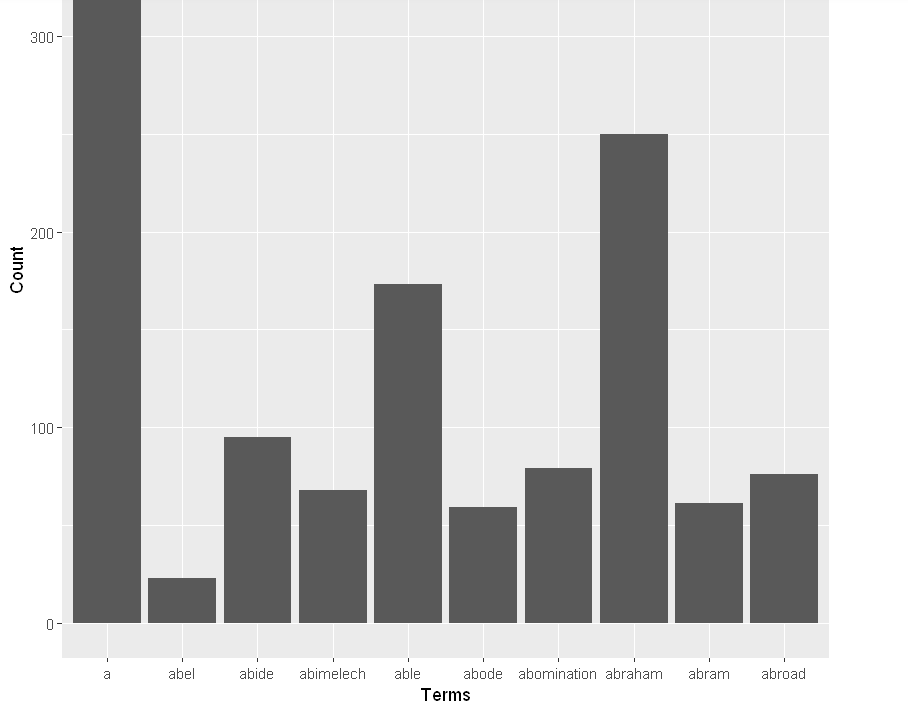
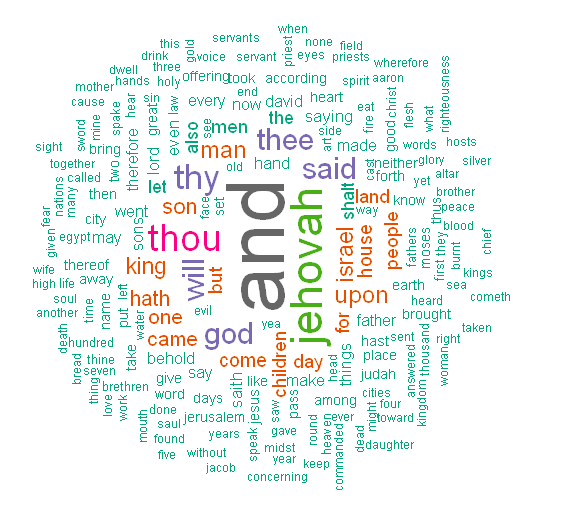
term.freq <- subset(term.freq, term.freq >= 15)

df <- data.frame(term = names(term.freq), freq = term.freq)

dim(df)

ggplot(df[1:10, 1:2], aes(x = term, y = freq)) + geom\_bar(stat = "identity") + xlab("Terms") + ylab("Count")

#dtm2 <- removeSparseTerms(v, sparse = 0.95)

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)

library(graph)

library(Rgraphviz)

library(wordcloud)

wordcloud(words = d$word, freq = d$freq, min.freq = 1,

max.words=200, random.order=FALSE, rot.per=0.35,

colors=brewer.pal(8, "Dark2"))

After that we find the most frequent used terms in the book and lowered down their frequency to 3

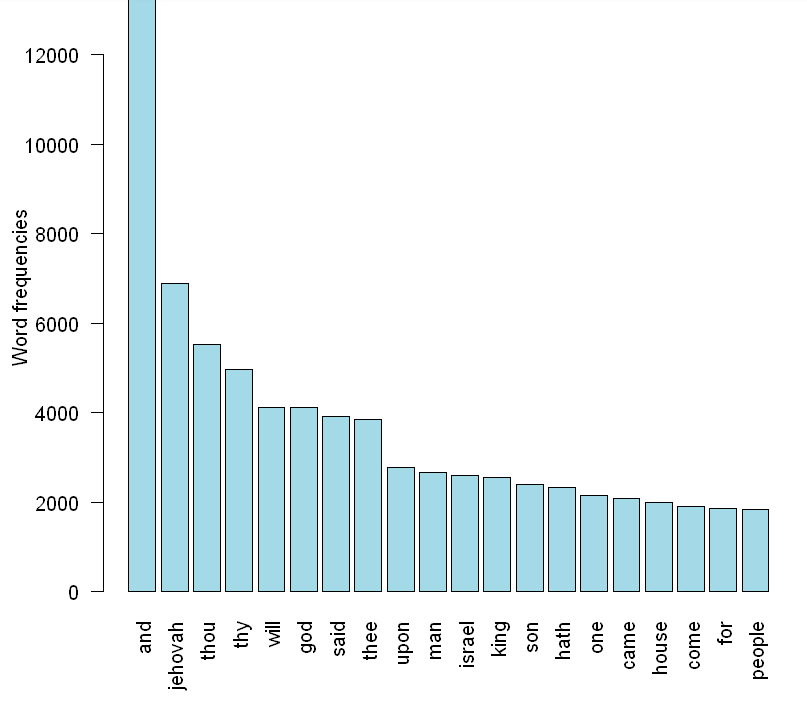


Fig: Most Frequent words graph

tdm2 <- removeSparseTerms(tdm, sparse = 0.95)

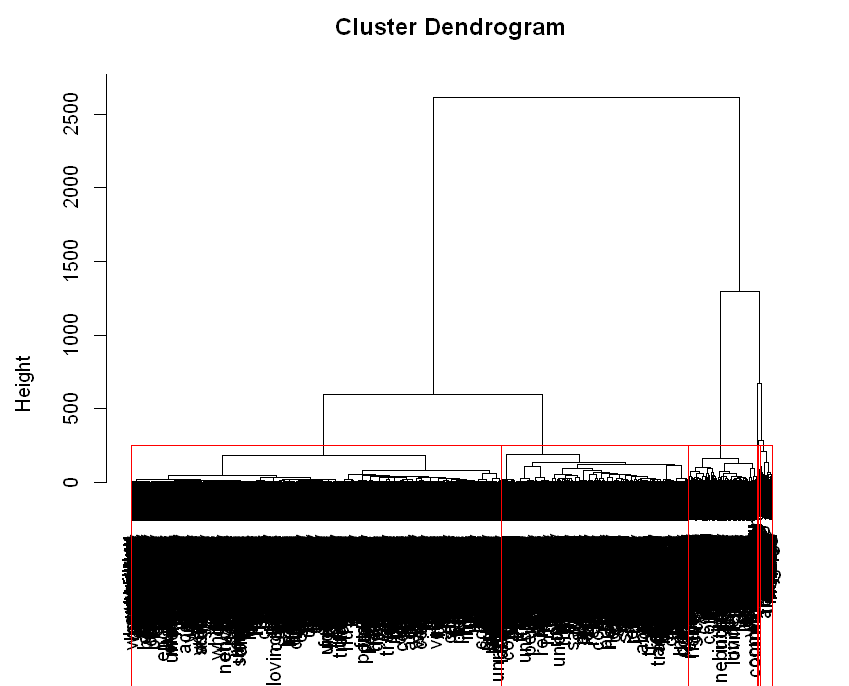
m2 <- as.matrix(tdm2)

distMatrix <- dist(scale(m2))

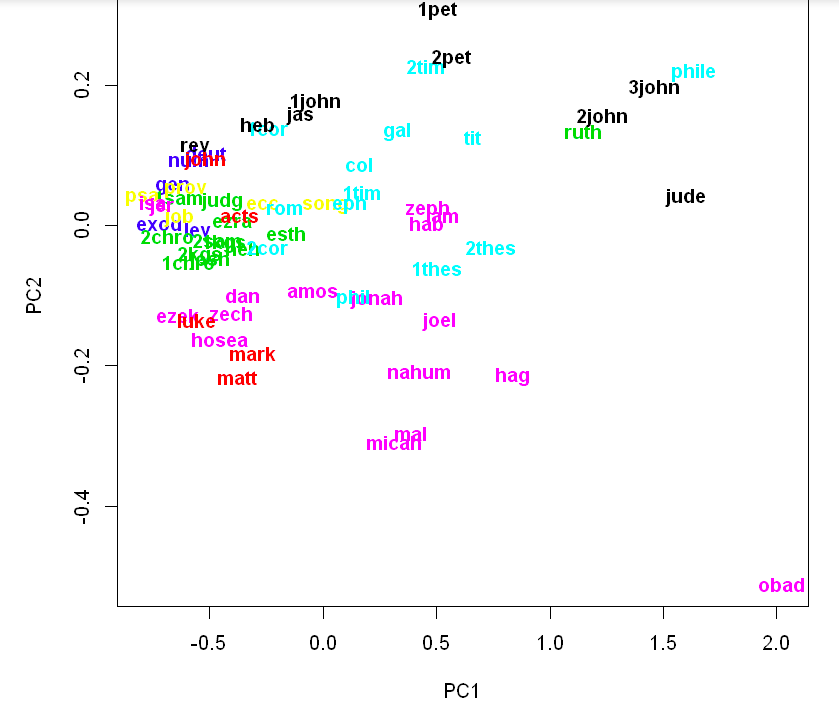
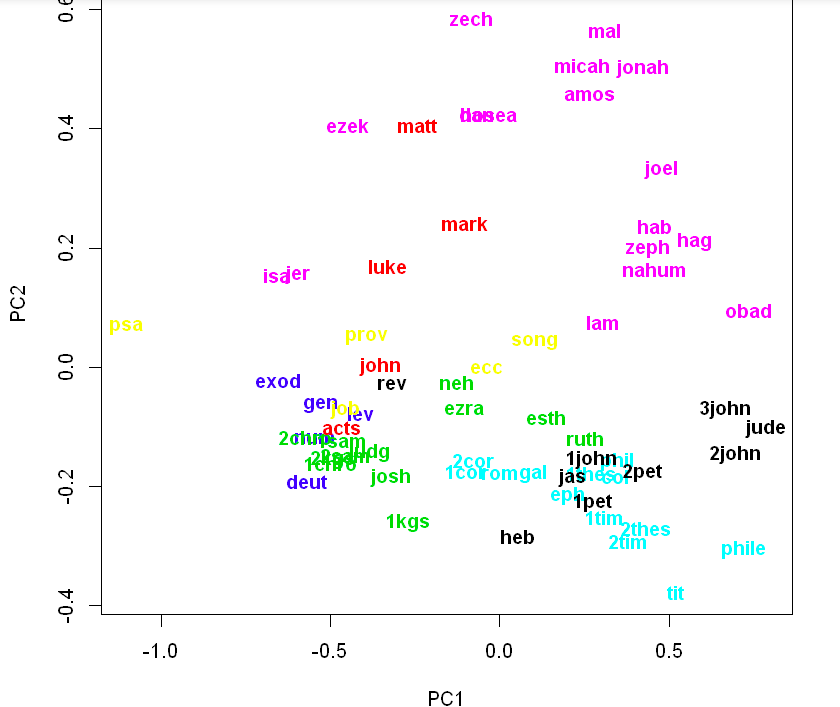
fit <- hclust(distMatrix, method = "ward")

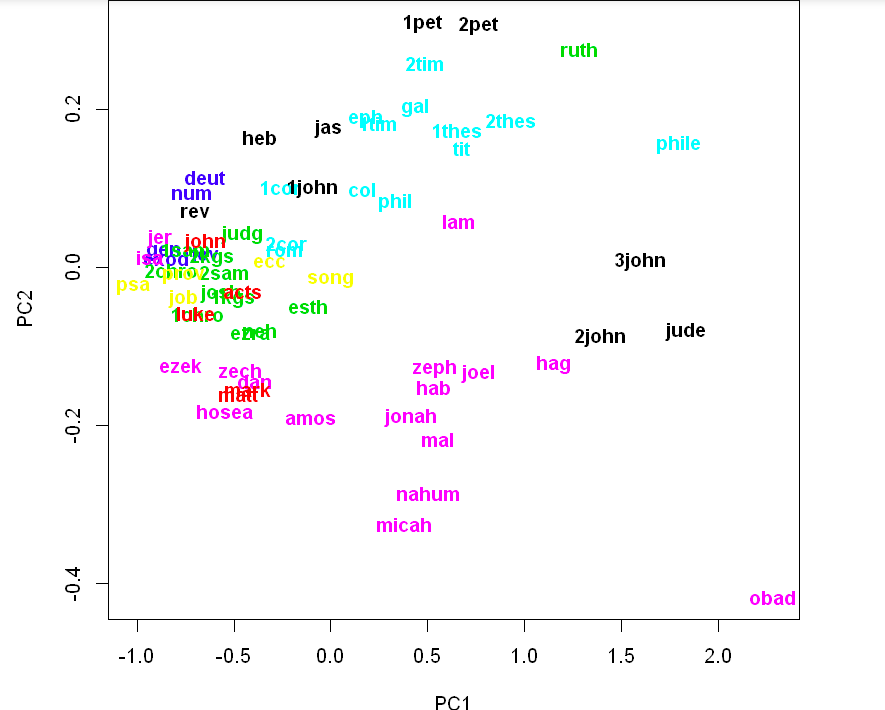
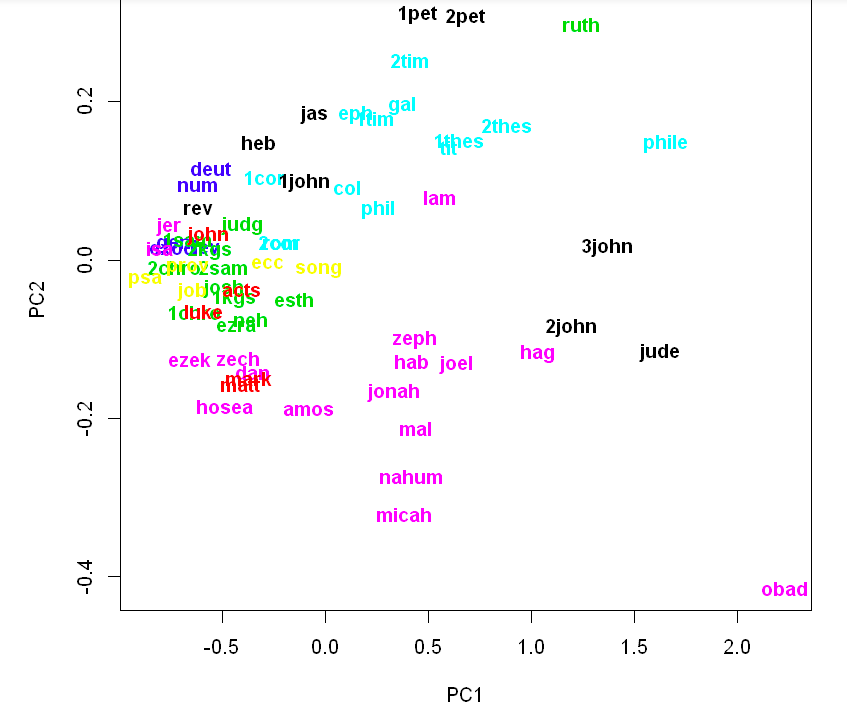
plot(fit)

rect.hclust(fit, k = 6)



* **Document Matrix**



* **PCA Analysis on the basis of Old and New testament**

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)

dend=as.dendrogram(hc.avg1)

labels\_colors(dend) <- as.numeric(as.factor(ASV\_Books$Sections[hc.avg1$order]))

#Change labels font size

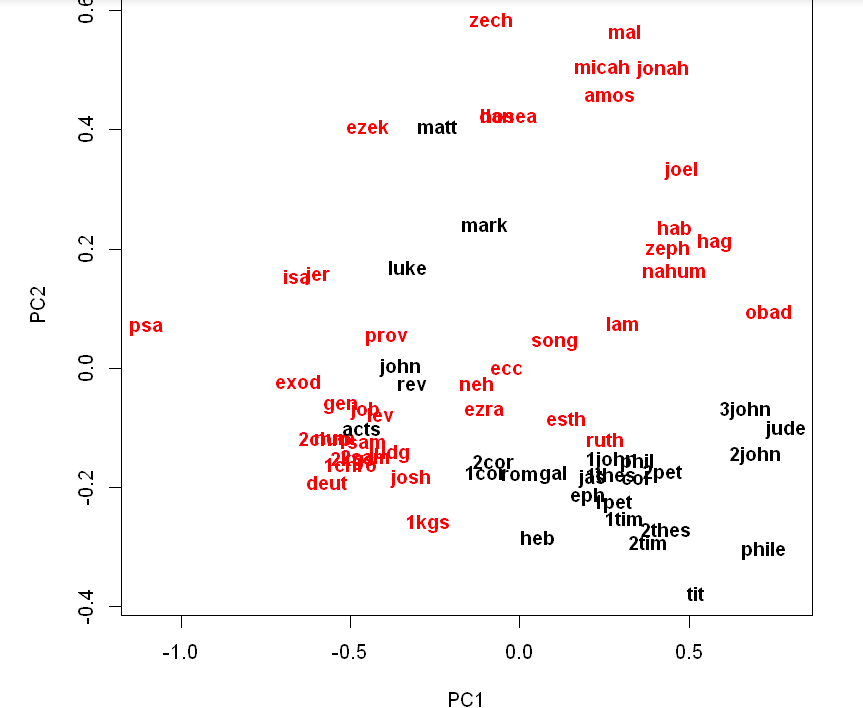
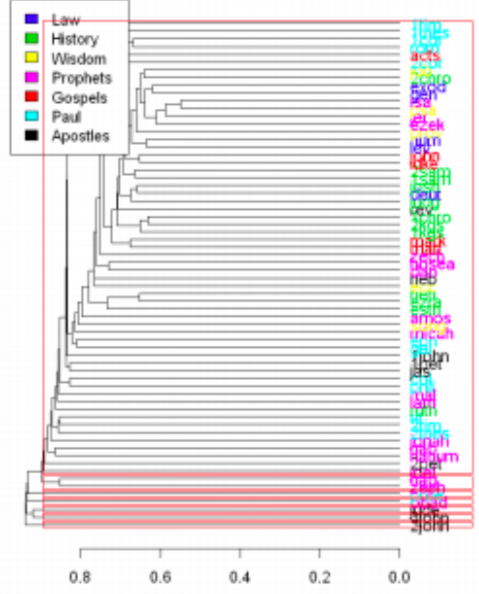
dend <- set(dend, "labels\_cex", 1.12)

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)

* **K-Means**

j = 1

for (idf\_weight in c(FALSE, TRUE)) {

for (stemfn\_name in c("None", "Porter")){

for (ngram\_length in c(1,3,7)) {

dtm = dtm.ngrams[[j]]

j = j+1

csim <- dtm / sqrt(rowSums(dtm\*dtm))

csim <- csim %\*% t(csim)

dist.mtx <- 1-csim

set.seed(1234)

km=kmeans(as.dist(dist.mtx), 7)

kmT=kmeans(as.dist(dist.mtx), 2)

# Calculate success rate

km.table = table(km$cluster, as.numeric(ASV\_Books$Sections))

km.match = solve\_LSAP(x = km.table, maximum = TRUE)

km.sxr = sum(km.table[cbind(seq\_along(km.match), km.match)]) / sum(km.table)

kmT.table = table(kmT$cluster, as.numeric(ASV\_Books$Testaments))

kmT.match = solve\_LSAP(x = kmT.table, maximum = TRUE)

kmT.sxr = sum(kmT.table[cbind(seq\_along(kmT.match), kmT.match)]) / sum(kmT.table)

cat(sprintf("n-grams: %d tf-idf: %i stemming: %s\n", ngram\_length, idf\_weight, stemfn\_name))

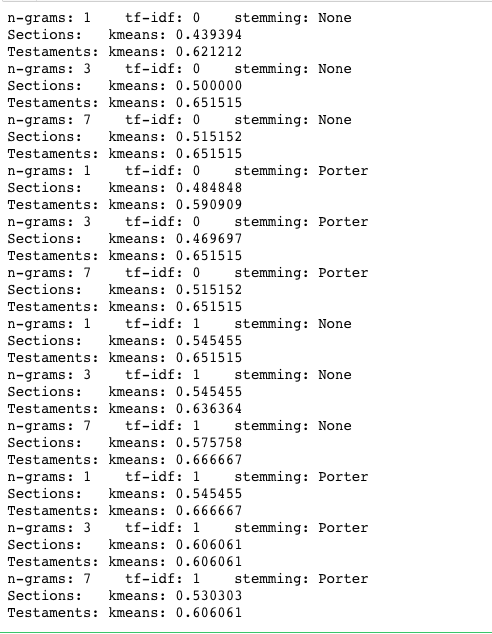
cat(sprintf("Sections: kmeans: %f\n", km.sxr))

cat(sprintf("Testaments: kmeans: %f\n", kmT.sxr))

}

}

}



dtm = dtm.ngrams[[7]]

csim <- dtm / sqrt(rowSums(dtm\*dtm))

csim <- csim %\*% t(csim)

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 - KMEANS")

text(x = fit.pca$x[,1], y = fit.pca$x[,2], labels = row.names(fit.pca$x), col=unclass(as.factor(km$cluster)), cex=.95, font=2)

mtext( cex = 1, text = "Seven Sections of the Bible based on Words Presence",

line=2,

outer=FALSE)

#PCA plot

fit.pca <- prcomp(as.dist(dist.mtx))

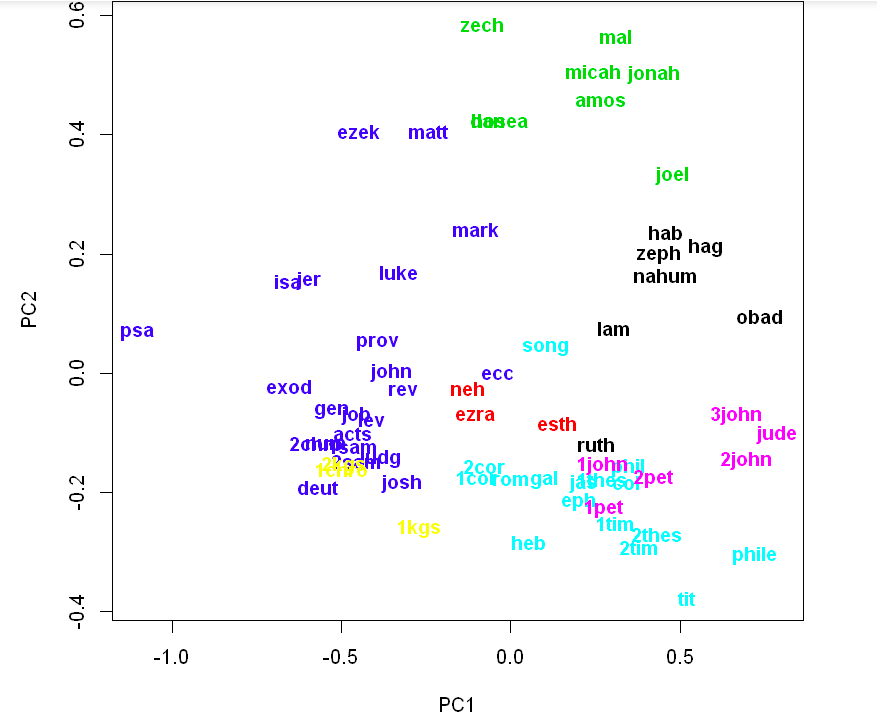
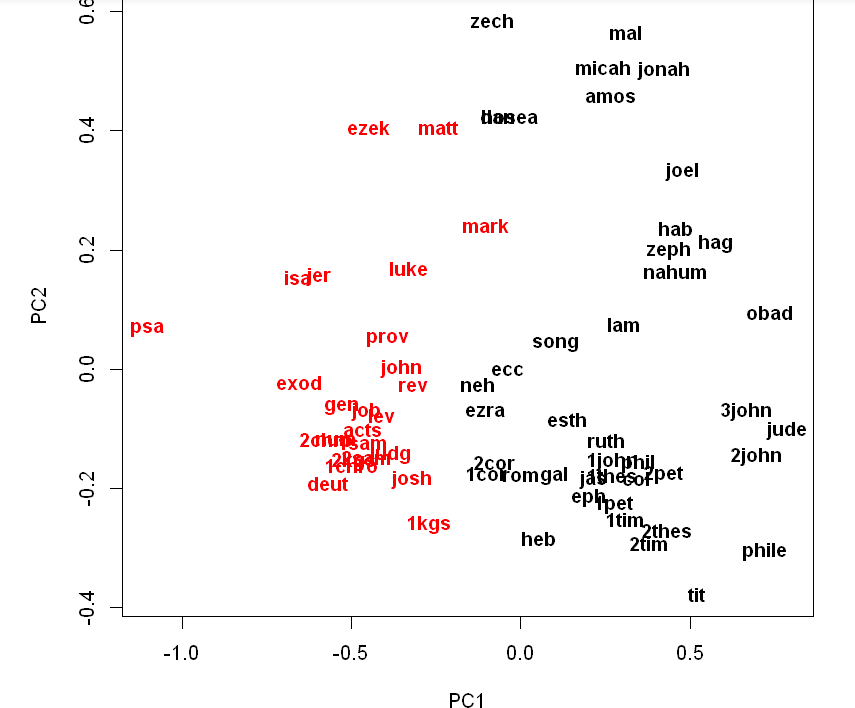
plot(fit.pca$x[,1:2], type='n',main="Two Testaments of the Bible based on Words Presence - KMEANS")

text(x = fit.pca$x[,1], y = fit.pca$x[,2], labels = row.names(fit.pca$x), col=unclass(as.factor(kmT$cluster)), cex=.95, font=2)

mtext( cex = 1, text = "Seven Sections of the Bible based on Words Presence",

line=2,

outer=FALSE)

* **Gaussian Mixture**

mc.sxr.best = 0

mcT.sxr.best = 0

j = 1

for (idf\_weight in c(FALSE, TRUE)) {

for (stemfn\_name in c("None", "Porter")){

for (ngram\_length in c(1,3,7)) {

dtm = dtm.ngrams[[j]]

j = j+1

csim <- dtm / sqrt(rowSums(dtm\*dtm))

csim <- csim %\*% t(csim)

dist.mtx <- 1-csim

set.seed(1234)

mc=Mclust(as.dist(dist.mtx), 7, verbose=F)

mcT=Mclust(as.dist(dist.mtx), 2, verbose=F)

# Calculate success rate

mc.table = table(mc$classification, as.numeric(ASV\_Books$Sections))

mc.match = solve\_LSAP(x = mc.table, maximum = TRUE)

mc.sxr = sum(mc.table[cbind(seq\_along(mc.match), mc.match)]) / sum(mc.table)

if (mc.sxr > mc.sxr.best) {

mc.sxr.best = mc.sxr

mc.best = mc

mc.j = j-1

}

mcT.table = table(mcT$classification, as.numeric(ASV\_Books$Testaments))

mcT.match = solve\_LSAP(x = mcT.table, maximum = TRUE)

mcT.sxr = sum(mcT.table[cbind(seq\_along(mcT.match), mcT.match)]) / sum(mcT.table)

if (mcT.sxr > mcT.sxr.best) {

mcT.sxr.best = mcT.sxr

mcT.best = mcT

mcT.j = j-1

}

cat(sprintf("n-grams: %d tf-idf: %i stemming: %s\n", ngram\_length, idf\_weight, stemfn\_name))

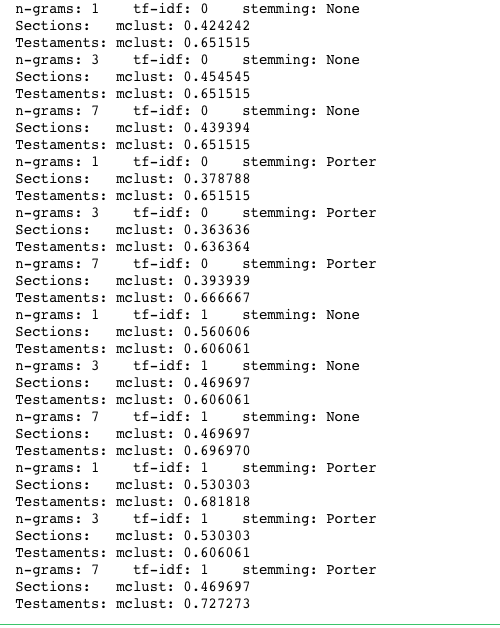
cat(sprintf("Sections: mclust: %f\n", mc.sxr))

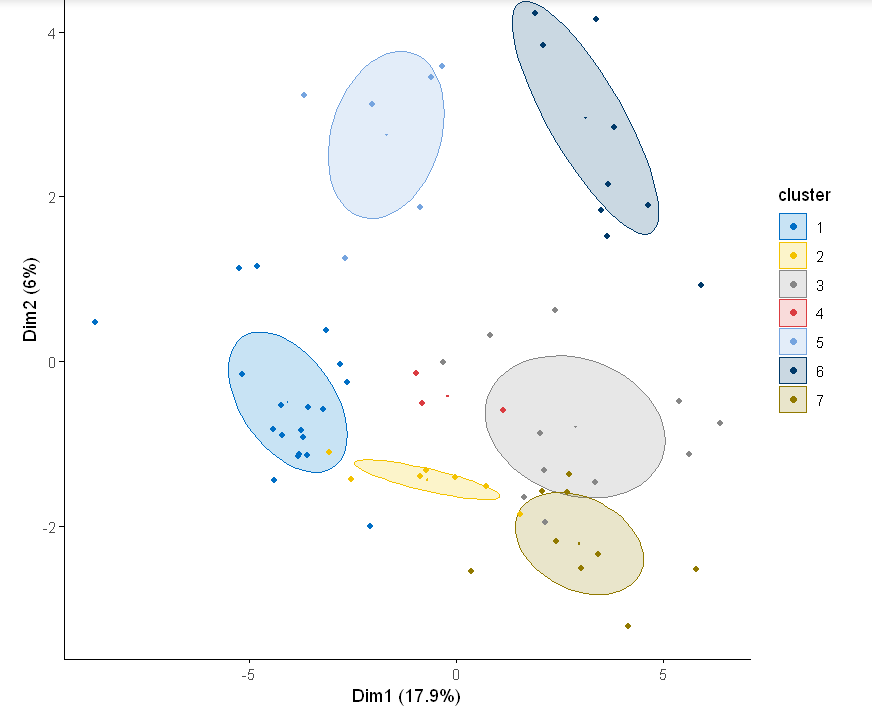
cat(sprintf("Testaments: mclust: %f\n", mcT.sxr))

}

}

}





**Conclusion:**

First bible is shrinked into 66 books of Old testament and New testament.

Then book analysis of bible has been done based on their sections. The most frequent found was according to the wordcloud graph which we got was "jehovah".

The results we got is dissimilar between old testament and new testament according to various analysis which has been done above.