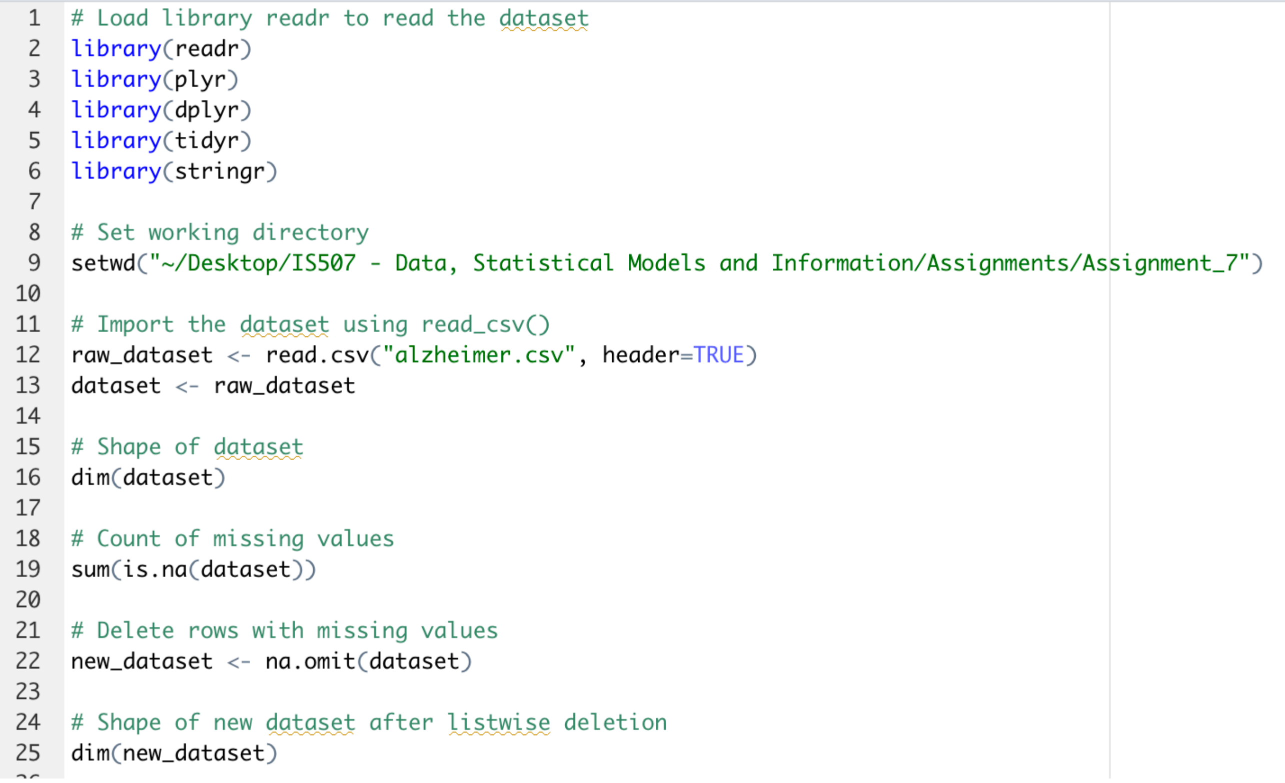
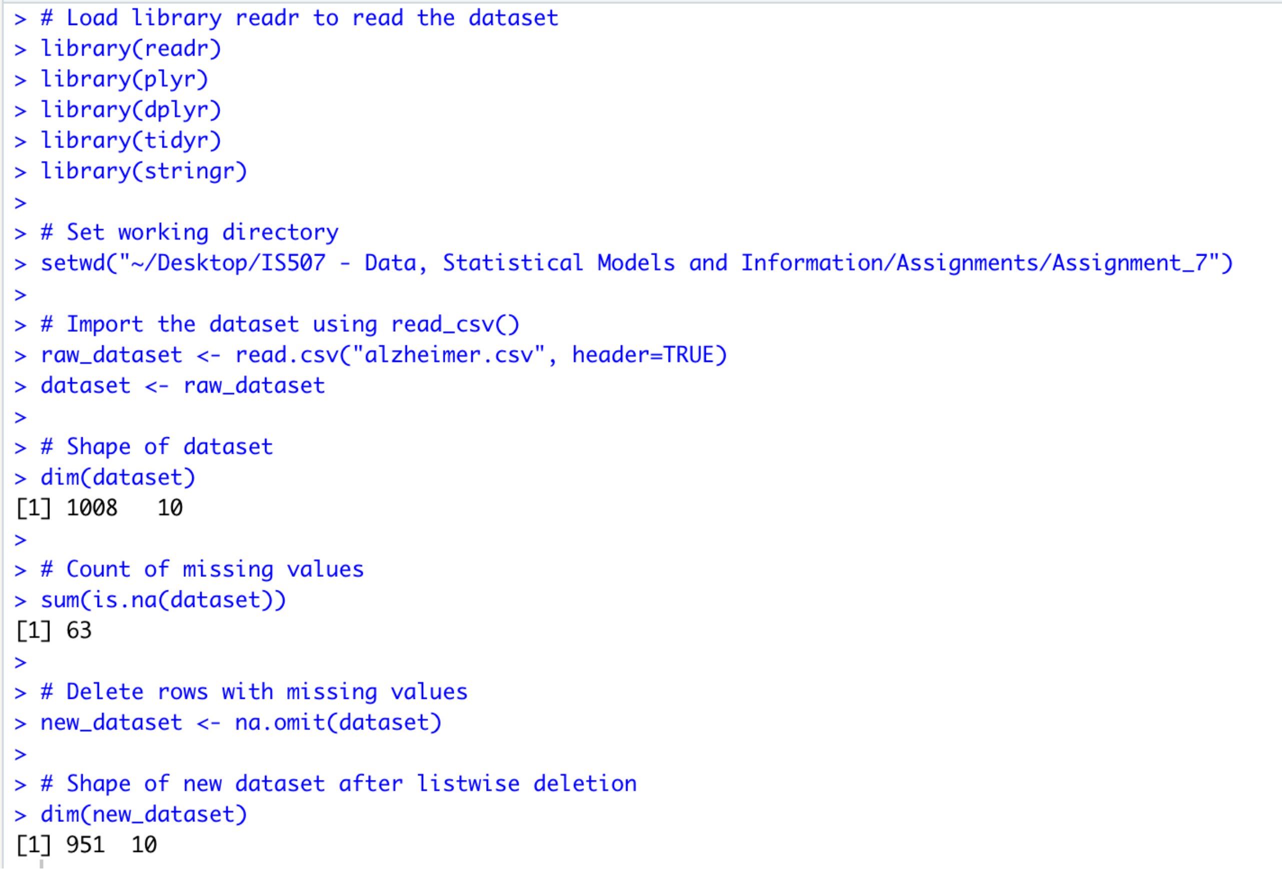
IS507 - Data, Statistical Models, and Information - Assignment 7

Dataset Cleaning:



```r
# Load library readr to read the dataset
library(readr)
library(plyr)
library(dplyr)
library(tidyr)
library(stringr)

# Set working directory
setwd("~/Desktop/IS507 - Data, Statistical Models and Information/Assignments/Assignment_7")

# Import the dataset using read_csv()
raw_dataset <- read.csv("alzheimer.csv", header=TRUE)
dataset <- raw_dataset

# Shape of dataset
dim(dataset)

# Count of missing values
sum(is.na(dataset))

# Delete rows with missing values
new_dataset <- na.omit(dataset)

# Shape of new dataset after listwise deletion
dim(new_dataset)
```



```
> # Load library readr to read the dataset
> library(readr)
> library(plyr)
> library(dplyr)
> library(tidyr)
> library(stringr)
>
> # Set working directory
> setwd("~/Desktop/IS507 - Data, Statistical Models and Information/Assignments/Assignment_7")
>
> # Import the dataset using read_csv()
> raw_dataset <- read.csv("alzheimer.csv", header=TRUE)
> dataset <- raw_dataset
>
> # Shape of dataset
> dim(dataset)
[1] 1008   10
>
> # Count of missing values
> sum(is.na(dataset))
[1] 63
>
> # Delete rows with missing values
> new_dataset <- na.omit(dataset)
>
> # Shape of new dataset after listwise deletion
> dim(new_dataset)
[1] 951  10
```

Problem 1: The Excel spreadsheet Alzheimer.csv contains one sheet named Alzheimer, which is data attempting to explain whether a patient has Alzheimer's Disease. These are data from a sample of 336 employees and consists of 9 variables for each patient. These are:

1) Dementia-Outcome variable-patient diagnosis
2) Gender-Female=0 and Male=1
3) Age-Age of patient (in years)
4) Education-Years of Education
5) SES-Socioeconomic Status 1=Low and 5=High
6) MMSE-Mini mental state examination score
7) CDR-Clinical Dementia Rating
8) eTIV-estimated total intracranial volume
9) nWBV-Normalize whole brain volume
10) ASF-Atlas Scaling Factor

Develop a **Linear Discriminant Analysis** model to classify the Dementia event from the other variables.

a) What is the performance of the classifier using cross-validation?

Solution:

In our dataset, we have the dependent variable as categorical which is Dementia.
Other variables are numeric and are the independent variables.

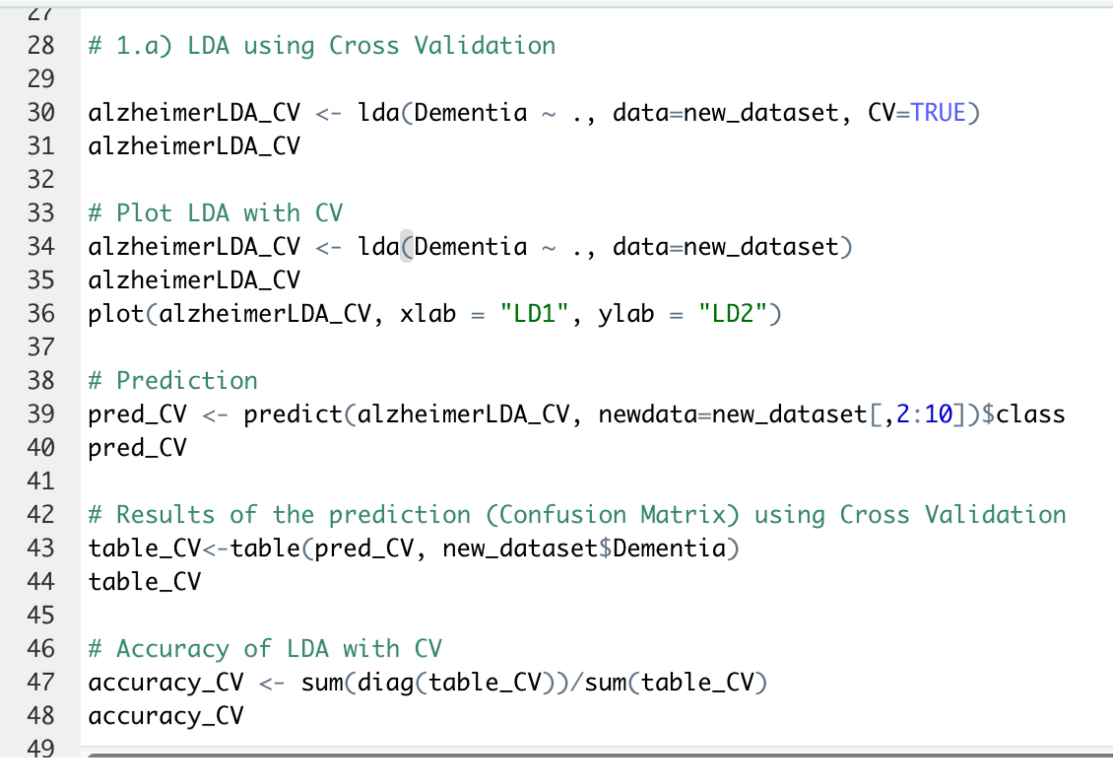
We firstly use the lda() function with CV=True to run a LDA model with cross validation. This will give us the LDA model with cross validation.

To plot it we run the model again without CV. The resulting model consists of the group means for the dependent variables, prior probabilities of the Dementia variable and the coefficients of the Linear Discriminant Dimension.

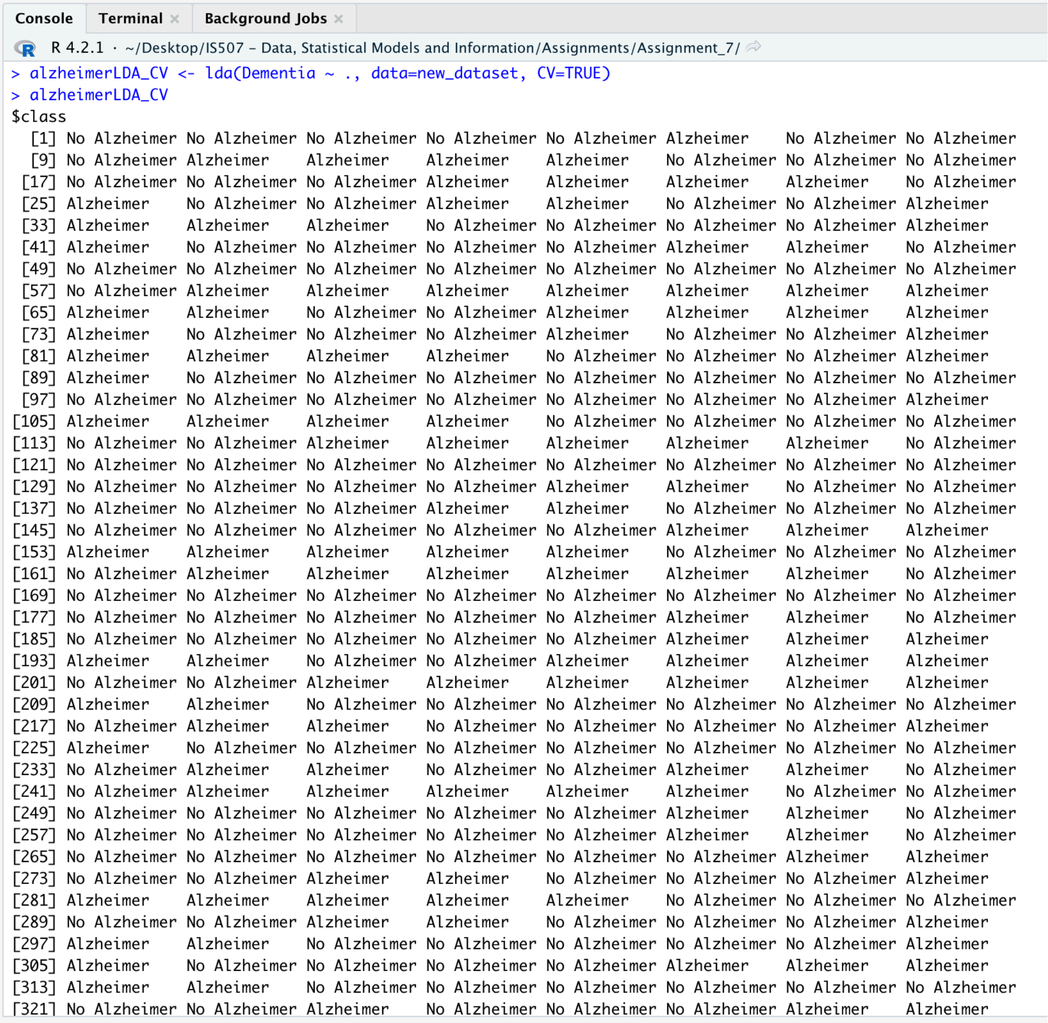
Using the generated model, we predict the class of Dementia for the dataset. The output of the prediction gives us the different class of the Dementia based on the input category. The prediction is then converted in a table (Confusion matrix) which gives us the value of true positives, true negatives, false positives, and fares negatives. We calculate the accuracy of the classifier as the ratio of sum of true positives and true negatives to the sum of all values.

In our LDA with CV, out of 951 rows we get 378 rows which are correctly classified as Alzheimer and 564 rows which are correctly classified as No Alzheimer. The remaining values are incorrectly classified. So accuracy = (378 + 564) / 951 = 99.05%.

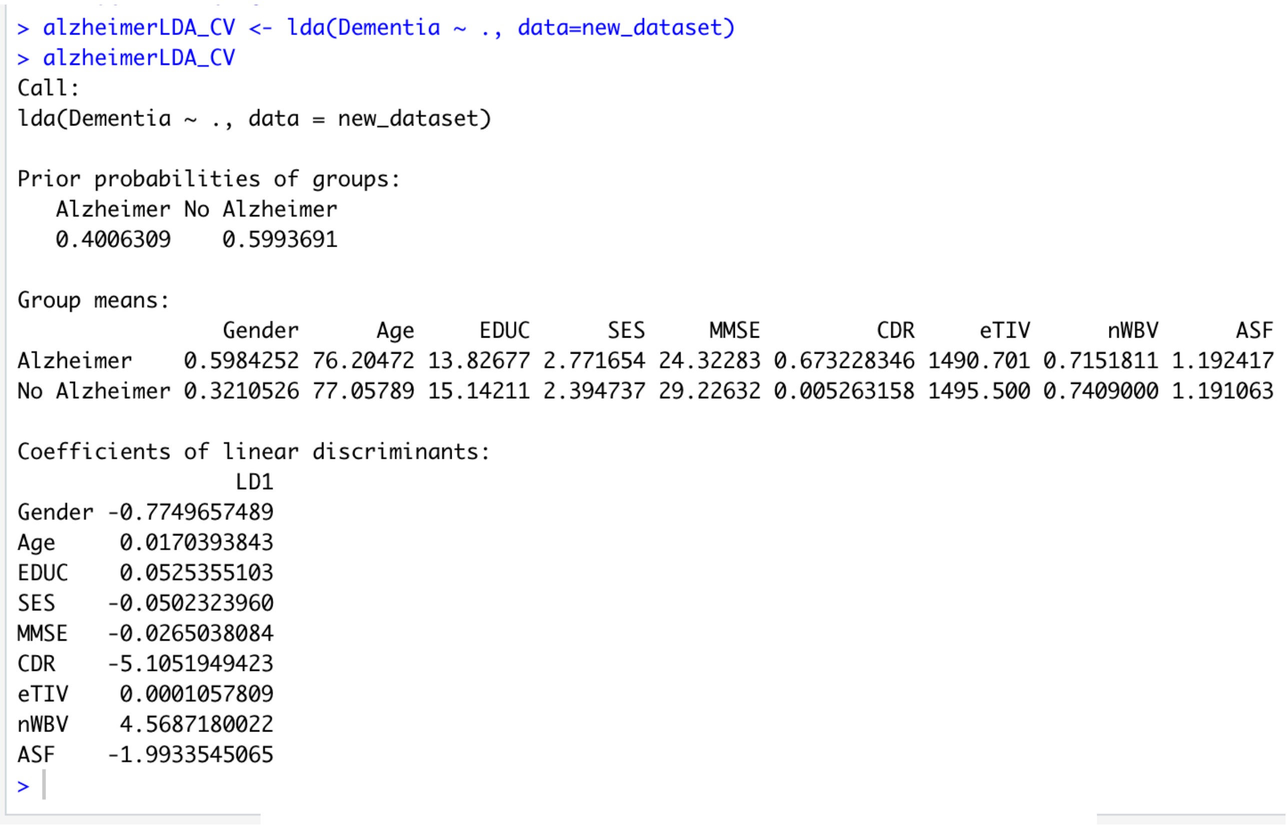
So, we can comment that the LDA model with CV has a very good accuracy of 99.05% and it can be very well used to classify whether the person has Alzheimer's or not.



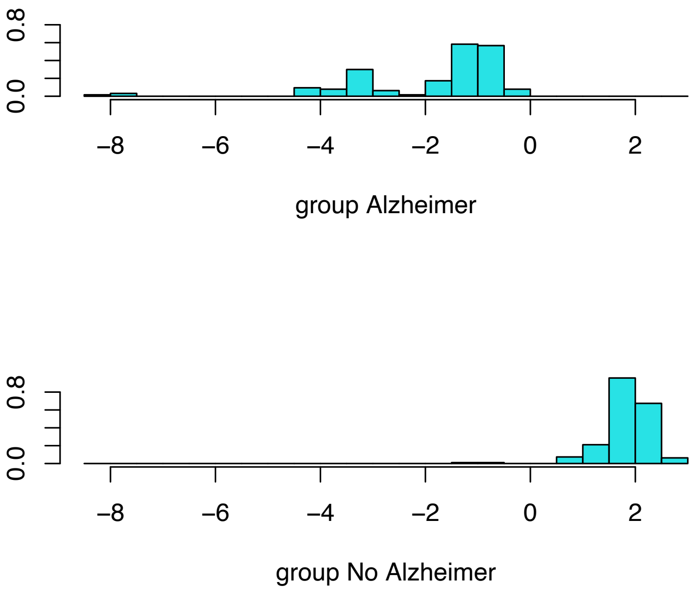
```r
# 1.a) LDA using Cross Validation

alzheimerLDA_CV <- lda(Dementia ~ ., data=new_dataset, CV=TRUE)
alzheimerLDA_CV

# Plot LDA with CV
alzheimerLDA_CV <- lda(Dementia ~ ., data=new_dataset)
alzheimerLDA_CV
plot(alzheimerLDA_CV, xlab = "LD1", ylab = "LD2")

# Prediction
pred_CV <- predict(alzheimerLDA_CV, newdata=new_dataset[,2:10])$class
pred_CV

# Results of the prediction (Confusion Matrix) using Cross Validation
table_CV<-table(pred_CV, new_dataset$Dementia)
table_CV

# Accuracy of LDA with CV
accuracy_CV <- sum(diag(table_CV))/sum(table_CV)
accuracy_CV

```

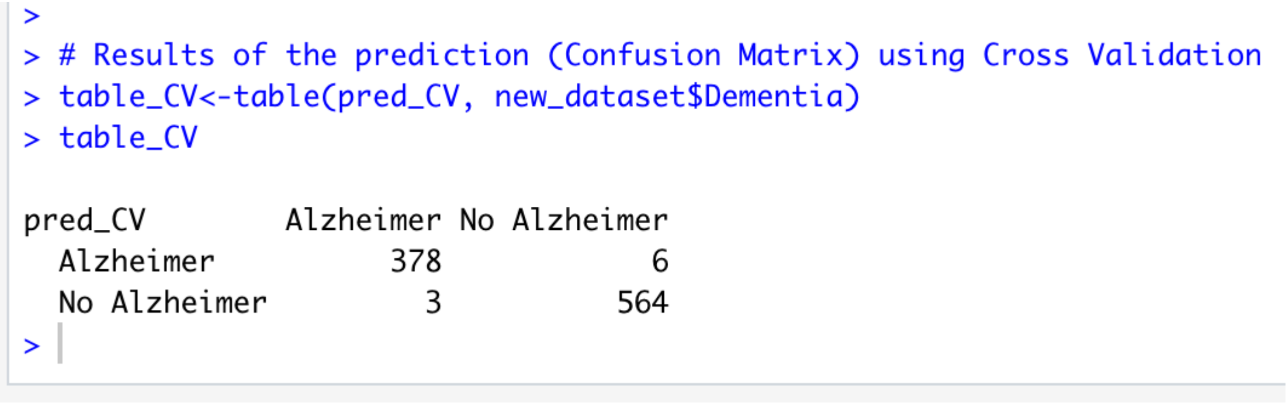

Console
Terminal
Background Jobs
R 4.2.1 · ~/Desktop/IS507 - Data, Statistical Models and Information/Assignments/Assignment_7/

```
> alzheimerLDA_CV <- lda(Dementia ~ ., data=new_dataset, CV=TRUE)
> alzheimerLDA_CV
$class
  [1] No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer    No Alzheimer No Alzheimer
  [9] No Alzheimer Alzheimer    Alzheimer    Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer
 [17] No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer    Alzheimer    Alzheimer    No Alzheimer
 [25] Alzheimer    No Alzheimer No Alzheimer Alzheimer    Alzheimer    No Alzheimer No Alzheimer Alzheimer
 [33] Alzheimer    Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer
 [41] Alzheimer    No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer    No Alzheimer
 [49] No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer
 [57] No Alzheimer Alzheimer    Alzheimer    Alzheimer    Alzheimer    Alzheimer    Alzheimer    Alzheimer
 [65] Alzheimer    Alzheimer    No Alzheimer No Alzheimer Alzheimer    Alzheimer    Alzheimer    Alzheimer
 [73] Alzheimer    No Alzheimer No Alzheimer No Alzheimer Alzheimer    No Alzheimer No Alzheimer Alzheimer
 [81] Alzheimer    Alzheimer    Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer Alzheimer
 [89] Alzheimer    No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer
 [97] No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer
[105] Alzheimer    Alzheimer    Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer No Alzheimer
[113] No Alzheimer No Alzheimer Alzheimer    Alzheimer    Alzheimer    Alzheimer    Alzheimer    No Alzheimer
[121] No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer
[129] No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer    No Alzheimer No Alzheimer
[137] No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer
[145] No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer    Alzheimer
[153] Alzheimer    Alzheimer    Alzheimer    Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer
[161] No Alzheimer Alzheimer    Alzheimer    Alzheimer    Alzheimer    Alzheimer    Alzheimer    No Alzheimer
[169] No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer
[177] No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer    No Alzheimer
[185] No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer    Alzheimer
[193] Alzheimer    Alzheimer    No Alzheimer No Alzheimer Alzheimer    Alzheimer    Alzheimer    Alzheimer
[201] No Alzheimer No Alzheimer Alzheimer    Alzheimer    Alzheimer    Alzheimer    Alzheimer    Alzheimer
[209] Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer
[217] No Alzheimer Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer
[225] Alzheimer    No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer
[233] No Alzheimer Alzheimer    Alzheimer    No Alzheimer No Alzheimer Alzheimer    Alzheimer    No Alzheimer
[241] No Alzheimer Alzheimer    Alzheimer    Alzheimer    Alzheimer    Alzheimer    No Alzheimer No Alzheimer
[249] No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer    No Alzheimer
[257] No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer    No Alzheimer
[265] No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer
[273] No Alzheimer No Alzheimer Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer Alzheimer
[281] Alzheimer    Alzheimer    Alzheimer    Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer
[289] No Alzheimer No Alzheimer Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer Alzheimer
[297] Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer
[305] Alzheimer    No Alzheimer No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer    Alzheimer
[313] Alzheimer    Alzheimer    No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer No Alzheimer
[321] No Alzheimer No Alzheimer Alzheimer    No Alzheimer No Alzheimer No Alzheimer Alzheimer    Alzheimer
```



```
> alzheimerLDA_CV <- lda(Dementia ~ ., data=new_dataset)
> alzheimerLDA_CV
Call:
lda(Dementia ~ ., data = new_dataset)

Prior probabilities of groups:
   Alzheimer No Alzheimer 
   0.4006309    0.5993691 

Group means:
                Gender      Age     EDUC      SES     MMSE         CDR     eTIV      nWBV      ASF
Alzheimer    0.5984252 76.20472 13.82677 2.771654 24.32283 0.673228346 1490.701 0.7151811 1.192417
No Alzheimer 0.3210526 77.05789 15.14211 2.394737 29.22632 0.005263158 1495.500 0.7409000 1.191063

Coefficients of linear discriminants:
                 LD1
Gender -0.7749657489
Age     0.0170393843
EDUC    0.0525355103
SES    -0.0502323960
MMSE   -0.0265038084
CDR    -5.1051949423
eTIV    0.0001057809
nWBV    4.5687180022
ASF    -1.9933545065
> 
```





```
> 
> # Results of the prediction (Confusion Matrix) using Cross Validation
> table_CV<-table(pred_CV, new_dataset$Dementia)
> table_CV

pred_CV        Alzheimer No Alzheimer
  Alzheimer          378            6
  No Alzheimer         3          564
> 
```

b) What is the performance of the classifier using training and testing?

Solution:

In our dataset, we have the dependent variable as categorical which is Dementia.
Other variables are numeric and are the independent variables.

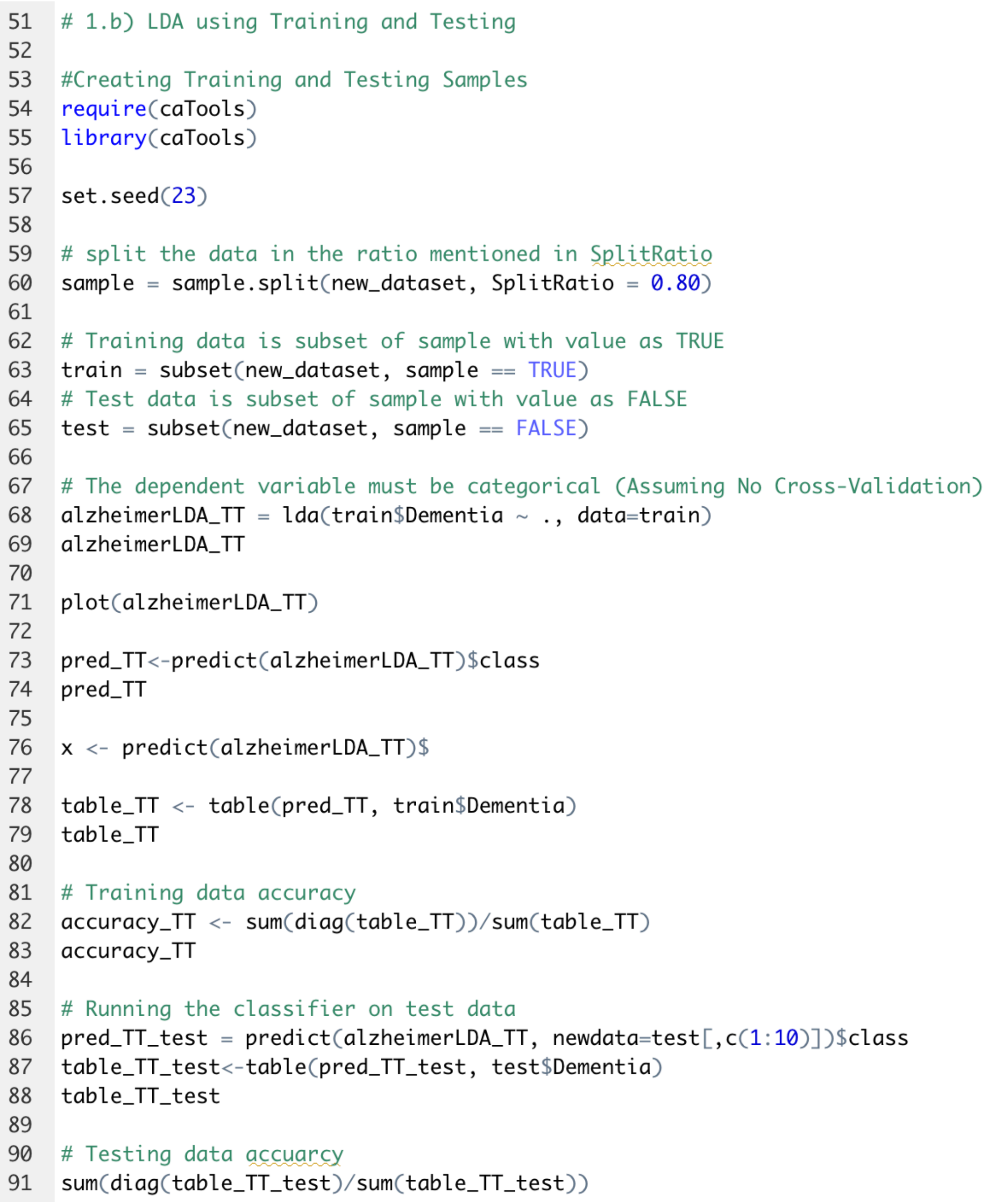
We firstly set a seed value so that we get the same set of rows in the train and test samples.
We then split the original dataset into 2 parts - train and test. We have used a ratio of 0.8 to split the data. It means 80% of the rows will be used in train data and the remaining 20% will be the test data.

We use the lda() function to generate a LDA model. Here, we will use the training data sample for creating the LDA classifier. We have 761 rows in the train dataset. Next, we predict the Dementia variable value for the training sample and generate a Confusion Matrix to calculate the accuracy of the train model. For the training dataset we get an accuracy = (303 + 453) / 761 = 99.34%

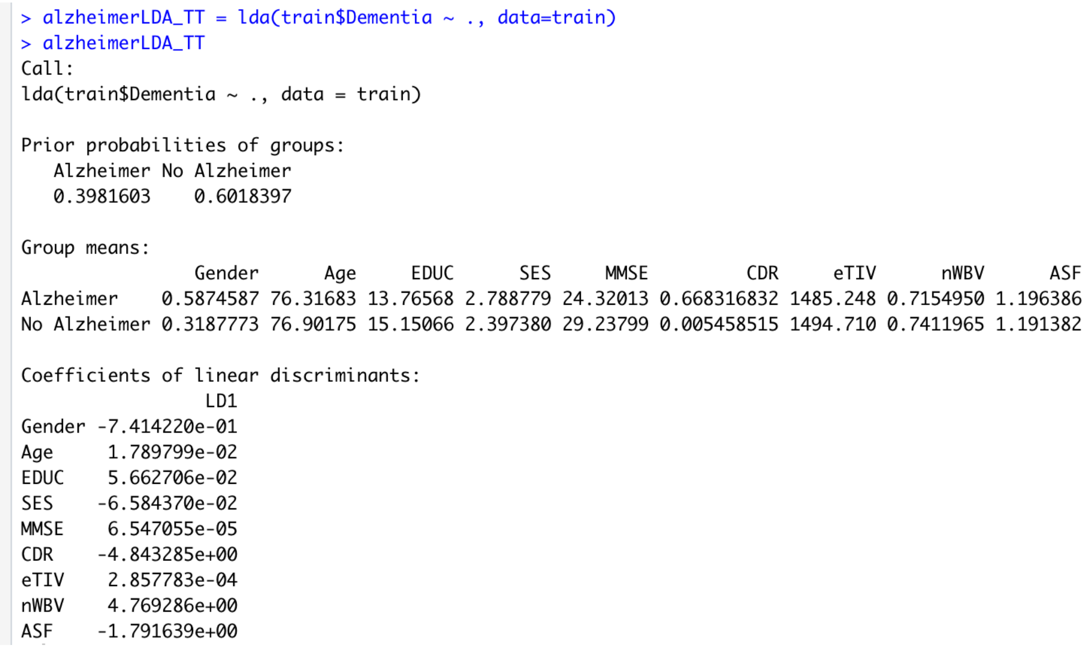
Using the generated model, we then run it on the test sample dataset to predict the class of Dementia for the dataset. We have 190 rows in the train dataset. The prediction is then converted in a table (Confusion Matrix) which gives us the value of true positives, true negatives, false positives, and fares negatives. We calculate the accuracy of the classifier as the ratio of sum of true positives and true negatives to the sum of all values.

In our LDA with Training and Testing, out of rows we get 378 rows which are correctly classified as Alzheimer and 564 rows which are correctly classified as No Alzheimer. The remaining values are incorrectly classified. So accuracy = (78 + 111) / 190 = 99.47%.

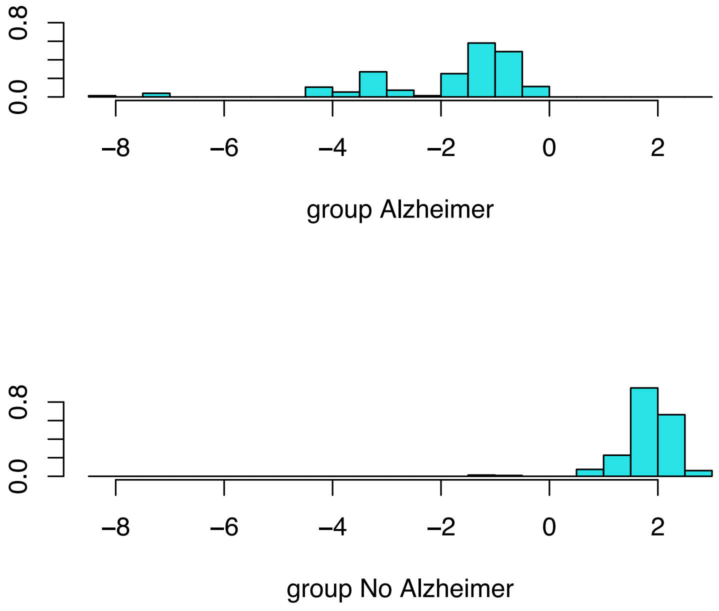
So, we can comment that the LDA model with Training and Testing has a very good accuracy of 99.47% and it can be very well used to classify whether the person has Alzheimer's or not.



```r
# 1.b) LDA using Training and Testing

#Creating Training and Testing Samples
require(caTools)
library(caTools)

set.seed(23)

# split the data in the ratio mentioned in SplitRatio
sample = sample.split(new_dataset, SplitRatio = 0.80)

# Training data is subset of sample with value as TRUE
train = subset(new_dataset, sample == TRUE)
# Test data is subset of sample with value as FALSE
test = subset(new_dataset, sample == FALSE)

# The dependent variable must be categorical (Assuming No Cross-Validation)
alzheimerLDA_TT = lda(train$Dementia ~ ., data=train)
alzheimerLDA_TT

plot(alzheimerLDA_TT)

pred_TT<-predict(alzheimerLDA_TT)$class
pred_TT

x <- predict(alzheimerLDA_TT)$

table_TT <- table(pred_TT, train$Dementia)
table_TT

# Training data accuracy
accuracy_TT <- sum(diag(table_TT))/sum(table_TT)
accuracy_TT

# Running the classifier on test data
pred_TT_test = predict(alzheimerLDA_TT, newdata=test[,c(1:10)])$class
table_TT_test<-table(pred_TT_test, test$Dementia)
table_TT_test

# Testing data accuarcy
sum(diag(table_TT_test)/sum(table_TT_test))
```



```
> alzheimerLDA_TT = lda(train$Dementia ~ ., data=train)
> alzheimerLDA_TT
Call:
lda(train$Dementia ~ ., data = train)

Prior probabilities of groups:
   Alzheimer No Alzheimer 
   0.3981603    0.6018397 

Group means:
               Gender      Age     EDUC      SES     MMSE         CDR     eTIV      nWBV      ASF
Alzheimer    0.5874587 76.31683 13.76568 2.788779 24.32013 0.668316832 1485.248 0.7154950 1.196386
No Alzheimer 0.3187773 76.90175 15.15066 2.397380 29.23799 0.005458515 1494.710 0.7411965 1.191382

Coefficients of linear discriminants:
                 LD1
Gender -7.414220e-01
Age     1.789799e-02
EDUC    5.662706e-02
SES    -6.584370e-02
MMSE    6.547055e-05
CDR    -4.843285e+00
eTIV    2.857783e-04
nWBV    4.769286e+00
ASF    -1.791639e+00
```



c) Would certain misclassification errors be worse than others? If so, how would you suggest measuring this?

<u>Solution:</u>

In case where the model predicts that a patient does not have Alzheimer but actually the patient has Alzheimer. In this case, we are predicting that the person is fine when in reality he/she is not. So, this can be a scenario where it is dangerous. This is an example of False Negative.

To measure this misclassification, we need a metric which can tell us how many samples are actually positive from the total positive results. This is **Sensitivity**.
To measure this misclassification, we need a metric which can tell us to how many samples are actually positive but have been classified as negative. This is **False Negative Rate.**

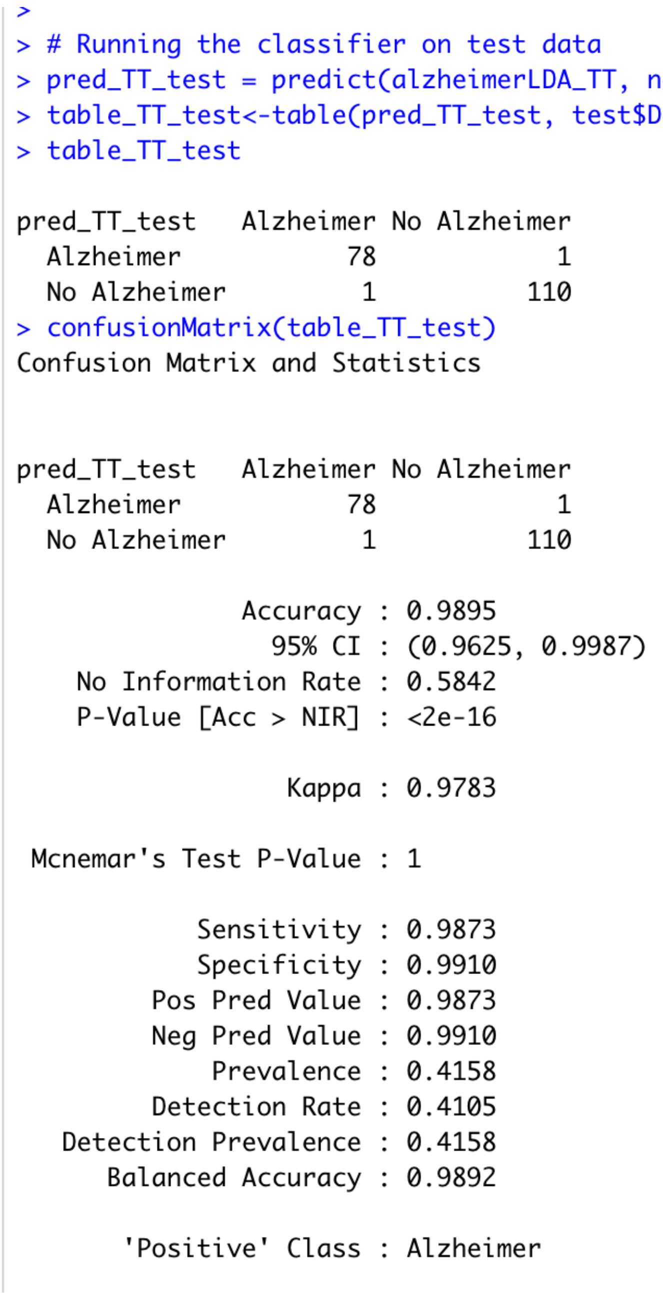
In our example, we can calculate the sensitivity as the ratio of true positive samples to the total positive samples.
When we use LDA with CV, we have sensitivity = 378 / (378 + 6) = 98.43%
When we use LDA with Training and Testing, we have sensitivity = 78 / (78 + 1) = 98.73%

In our example, we can calculate the false negative rate as the ratio of false negative samples to the total number of positive samples.
When we use LDA with CV, we have false negative rate = 6 / (378 + 6) = 1.57 %
When we use LDA with Training and Testing, we have false negative rate = 1 / (78 + 1) = 1.27%

We also can have cases when the patient does not suffer from Alzheimer but is classified having Alzheimer's. This means that the person is totally fine but is classified as sick and given treatment. This is an example for False Positive.

To measure this misclassification, we need a metric which can tell us how many samples are actually positive from the total negative results. This is **False Positive Rate**.
When we use LDA with CV, we have false positive rate = 3 / (3 + 564) = 0.53 %
When we use LDA with Training and Testing, we have false positive rate = 0 / (0 + 111) = 0%

Our model has a very **less False Negative Rate / Miss Rate as well as less False Positive Rate** which makes it a good classifier. Hence we can say that the model can do well when we given data.



```
>
> # Running the classifier on test data
> pred_TT_test = predict(alzheimerLDA_TT, newdata=test[,c(1:10)])$class
> table_TT_test<-table(pred_TT_test, test$Dementia)
> table_TT_test

pred_TT_test   Alzheimer No Alzheimer
  Alzheimer           78            1
  No Alzheimer         1          110
> confusionMatrix(table_TT_test)
Confusion Matrix and Statistics


pred_TT_test   Alzheimer No Alzheimer
  Alzheimer           78            1
  No Alzheimer         1          110

               Accuracy : 0.9895
                 95% CI : (0.9625, 0.9987)
    No Information Rate : 0.5842
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9783

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9873
            Specificity : 0.9910
         Pos Pred Value : 0.9873
         Neg Pred Value : 0.9910
             Prevalence : 0.4158
         Detection Rate : 0.4105
   Detection Prevalence : 0.4158
      Balanced Accuracy : 0.9892

       'Positive' Class : Alzheimer
```

Problem 2: Select one of the techniques (i.e. CA, Cluster Analysis, LDA) and apply it to some aspect of your final project dataset. Or research a new technique that we have not covered and apply it). Each team member should investigate a different aspect of the dataset.

Solution:

I have chosen to run the Cluster Analysis technique onto the OnlineNewsPopularity dataset of our project.

The research question I wish to answer is to divide the data into 3 clusters based on positive, negative or neutral sentiments. I have chosen a subset of the entire dataset to perform the clustering. The features used for the same are:

global_rate_postive_words - Rate of positive words in the content
global_rate_negative_words - Rate of negative words in the content
rate_postive_words - Rate of positive words among non-neutral tokens
rate_postive_words - Rate of negative words among non-neutral tokens
avg_positive_polarity - Average polarity of positive words
min_positive_polarity - Minimum polarity of positive words
max_positive_polarity - Maximum polarity of positive words
avg_negative_polarity - Average polarity of positive words
min_negative_polarity - Minimum polarity of positive words
max_negative_polarity - Maximum polarity of positive words

We have checked the normality of the subset using the Anderson-Darling test and then applied log and square root transformations to normalise them.

I want to cluster the data into 3 clusters - positive, negative and neutral. So, I have set a range for the optimal clusters from 1 to 5:
I have calculated the Within-groups Sum of Squares for the data and plotted a graph of the number of clusters against the Within-groups Sum of Squares.
The curve can give the optimal number of clusters where there is a bend or knee. In my case the graph knees at 3 indicating we can have 3 clusters.

I will be using the K-Means clustering algorithm with 3 centres to cluster the data.
When we plot the data we get 3 clusters where the inter-cluster distance is less and the clusters tend to overlap slightly.

Naming the clusters:

Cluster 1 —> Neutral sentiment articles

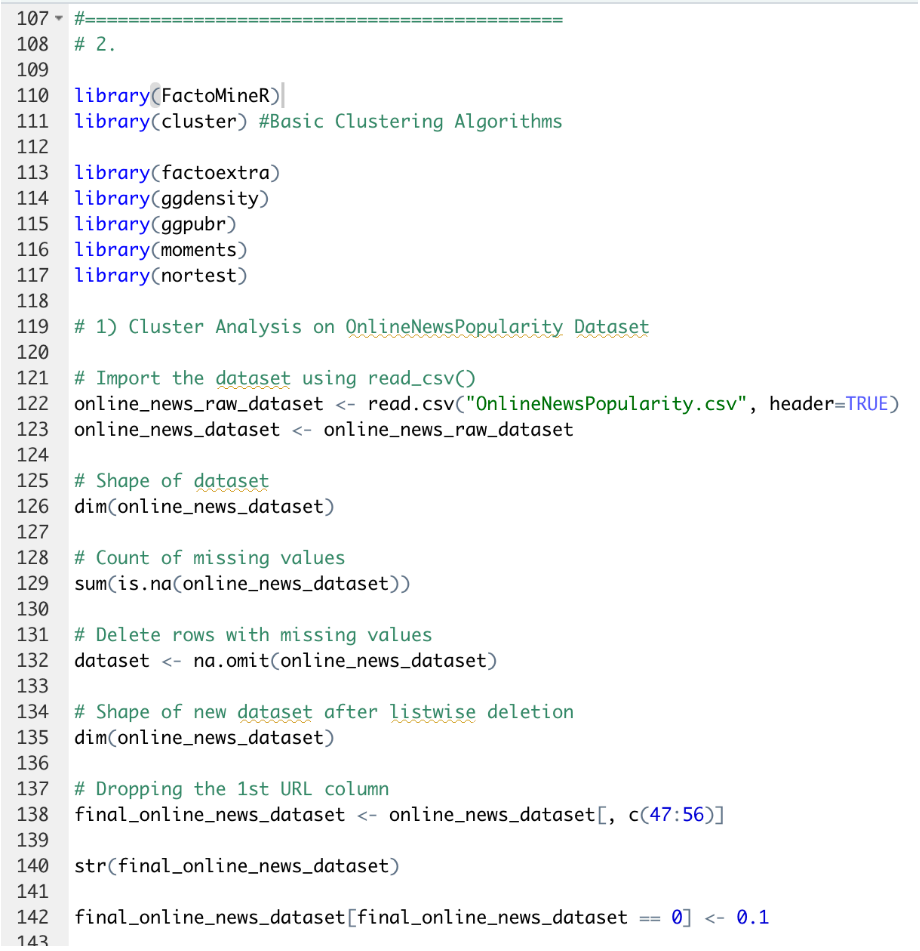
From the cluster centres we can observe that the articles in this group have a mix of positive and negative content. The global positive and negative rate is significant along with the positive and negative polarity values. So, we can conclude that such articles contain neutral sentiments.

Cluster 2 —> Negative sentiment articles

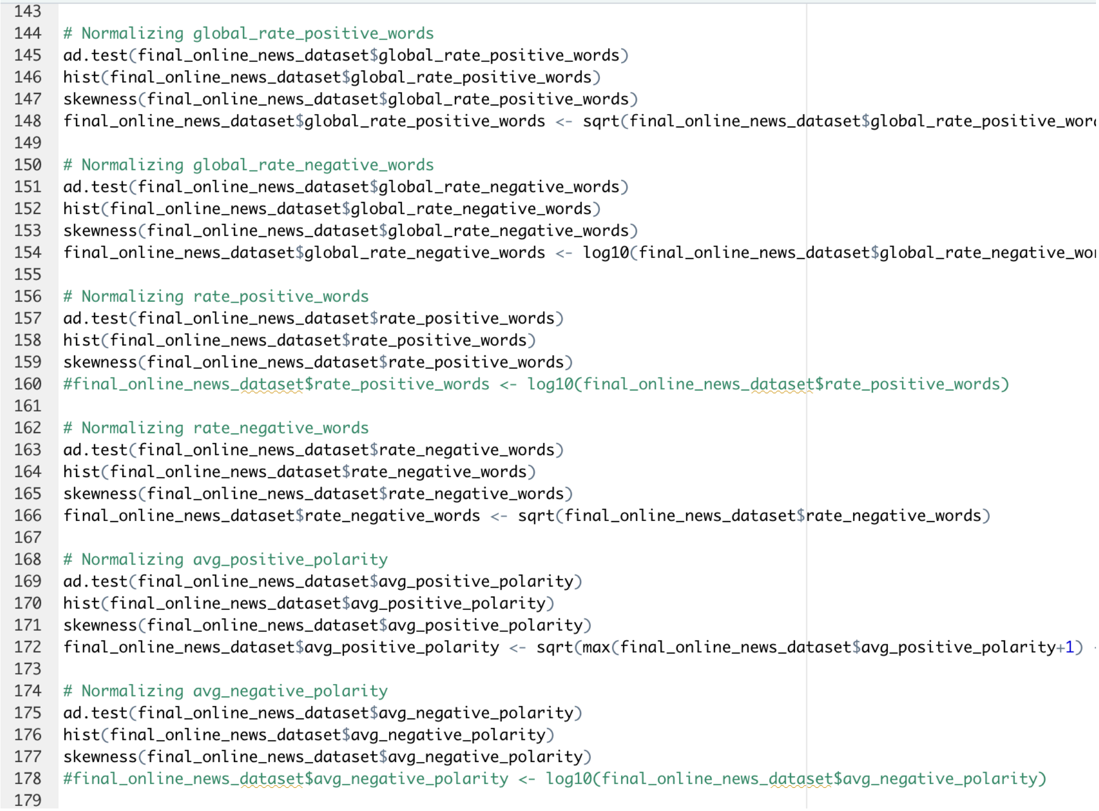
From the cluster centres we can observe that the articles in this group have a high negative tone. The global positive rate, positive rate and avg positive polarities have negative values. Whereas, the global negative rate, negative rate and max negative rates have positive values. So, we can conclude that such articles contain negative sentiments.

Cluster 3 —> Positive sentiment articles

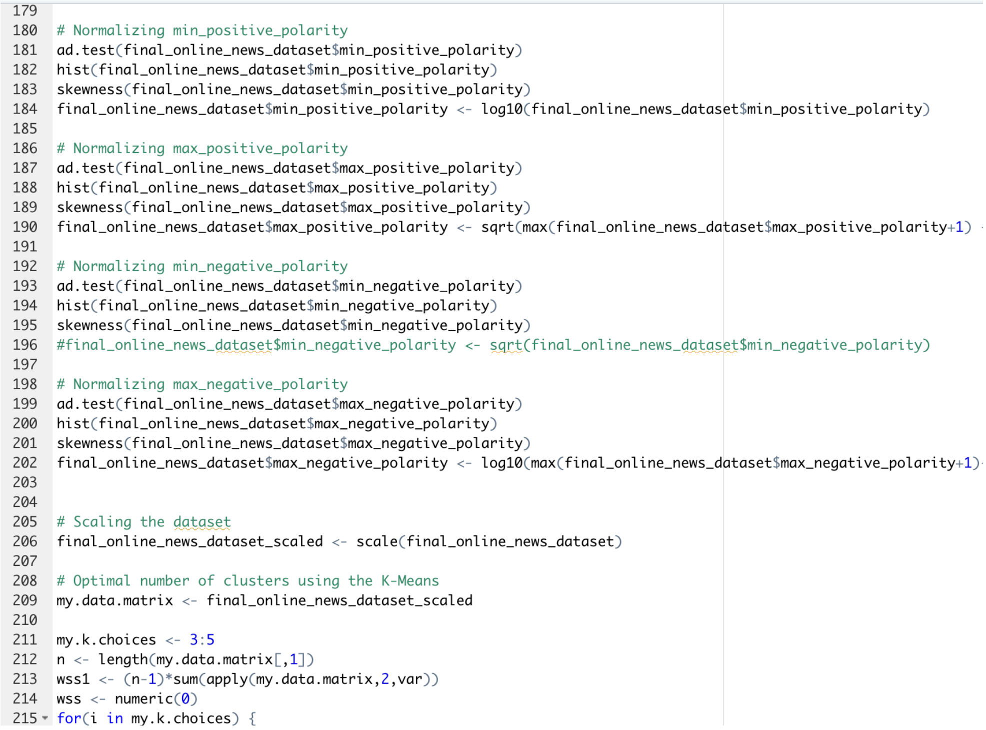
From the cluster centres we can observe that the articles in this group have a high positive tone. The global positive rate, positive rate have positive values. Whereas, the global negative rate, negative rate and avg negative rates have positive values. So, we can conclude that such articles contain positive sentiments.



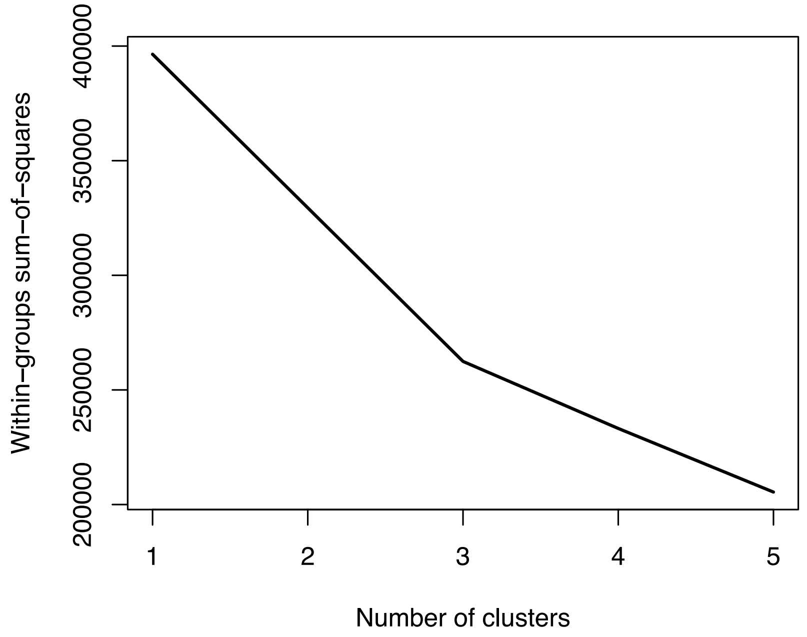
```r
#=============================================
# 2.

library(FactoMineR)
library(cluster) #Basic Clustering Algorithms

library(factoextra)
library(ggdensity)
library(ggpubr)
library(moments)
library(nortest)

# 1) Cluster Analysis on OnlineNewsPopularity Dataset

# Import the dataset using read_csv()
online_news_raw_dataset <- read.csv("OnlineNewsPopularity.csv", header=TRUE)
online_news_dataset <- online_news_raw_dataset

# Shape of dataset
dim(online_news_dataset)

# Count of missing values
sum(is.na(online_news_dataset))

# Delete rows with missing values
dataset <- na.omit(online_news_dataset)

# Shape of new dataset after listwise deletion
dim(online_news_dataset)

# Dropping the 1st URL column
final_online_news_dataset <- online_news_dataset[, c(47:56)]

str(final_online_news_dataset)

final_online_news_dataset[final_online_news_dataset == 0] <- 0.1
```

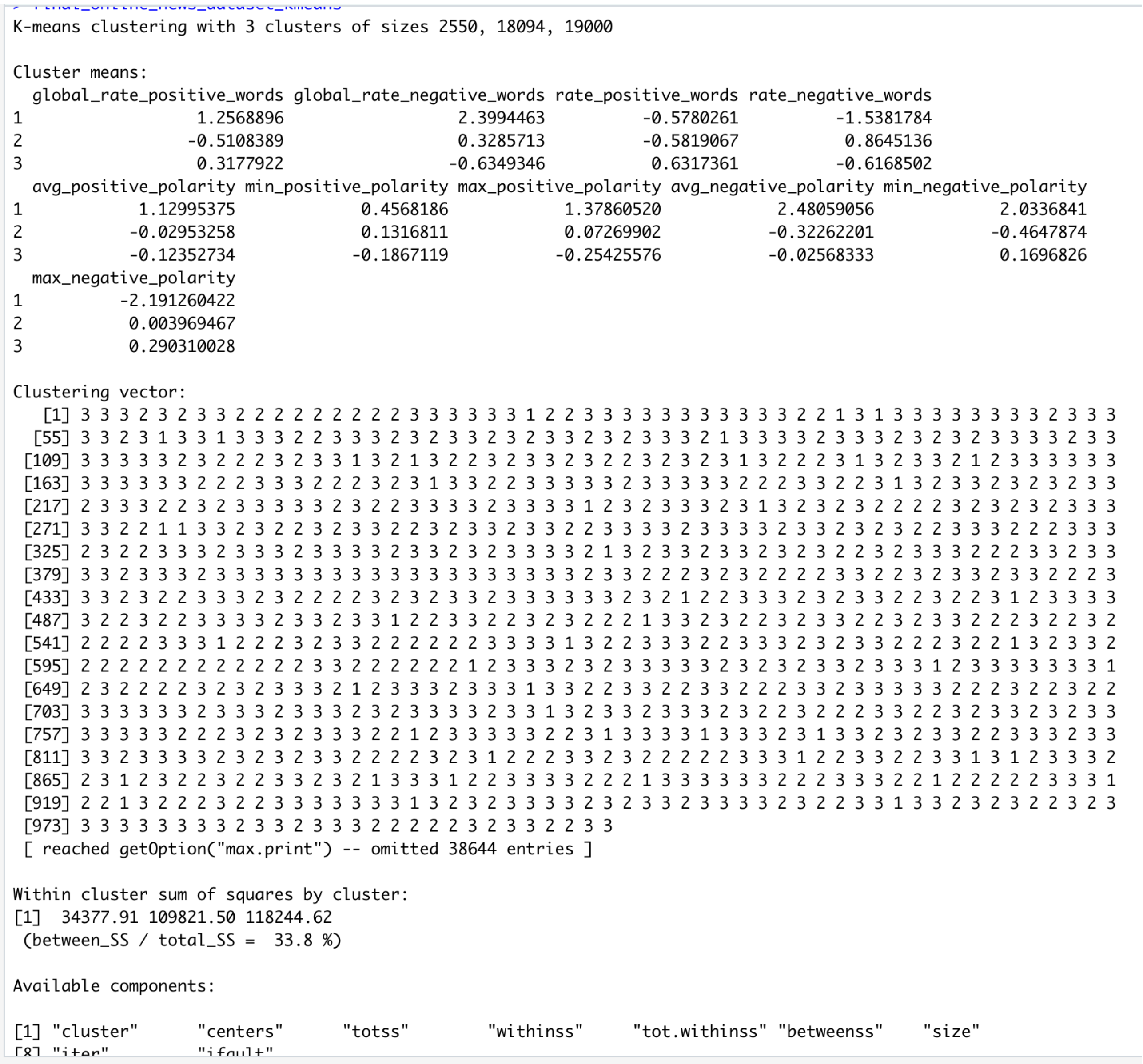


```
# Normalizing global_rate_positive_words
ad.test(final_online_news_dataset$global_rate_positive_words)
hist(final_online_news_dataset$global_rate_positive_words)
skewness(final_online_news_dataset$global_rate_positive_words)
final_online_news_dataset$global_rate_positive_words <- sqrt(final_online_news_dataset$global_rate_positive_wor

# Normalizing global_rate_negative_words
ad.test(final_online_news_dataset$global_rate_negative_words)
hist(final_online_news_dataset$global_rate_negative_words)
skewness(final_online_news_dataset$global_rate_negative_words)
final_online_news_dataset$global_rate_negative_words <- log10(final_online_news_dataset$global_rate_negative_wo

# Normalizing rate_positive_words
ad.test(final_online_news_dataset$rate_positive_words)
hist(final_online_news_dataset$rate_positive_words)
skewness(final_online_news_dataset$rate_positive_words)
#final_online_news_dataset$rate_positive_words <- log10(final_online_news_dataset$rate_positive_words)

# Normalizing rate_negative_words
ad.test(final_online_news_dataset$rate_negative_words)
hist(final_online_news_dataset$rate_negative_words)
skewness(final_online_news_dataset$rate_negative_words)
final_online_news_dataset$rate_negative_words <- sqrt(final_online_news_dataset$rate_negative_words)

# Normalizing avg_positive_polarity
ad.test(final_online_news_dataset$avg_positive_polarity)
hist(final_online_news_dataset$avg_positive_polarity)
skewness(final_online_news_dataset$avg_positive_polarity)
final_online_news_dataset$avg_positive_polarity <- sqrt(max(final_online_news_dataset$avg_positive_polarity+1)

# Normalizing avg_negative_polarity
ad.test(final_online_news_dataset$avg_negative_polarity)
hist(final_online_news_dataset$avg_negative_polarity)
skewness(final_online_news_dataset$avg_negative_polarity)
#final_online_news_dataset$avg_negative_polarity <- log10(final_online_news_dataset$avg_negative_polarity)

```

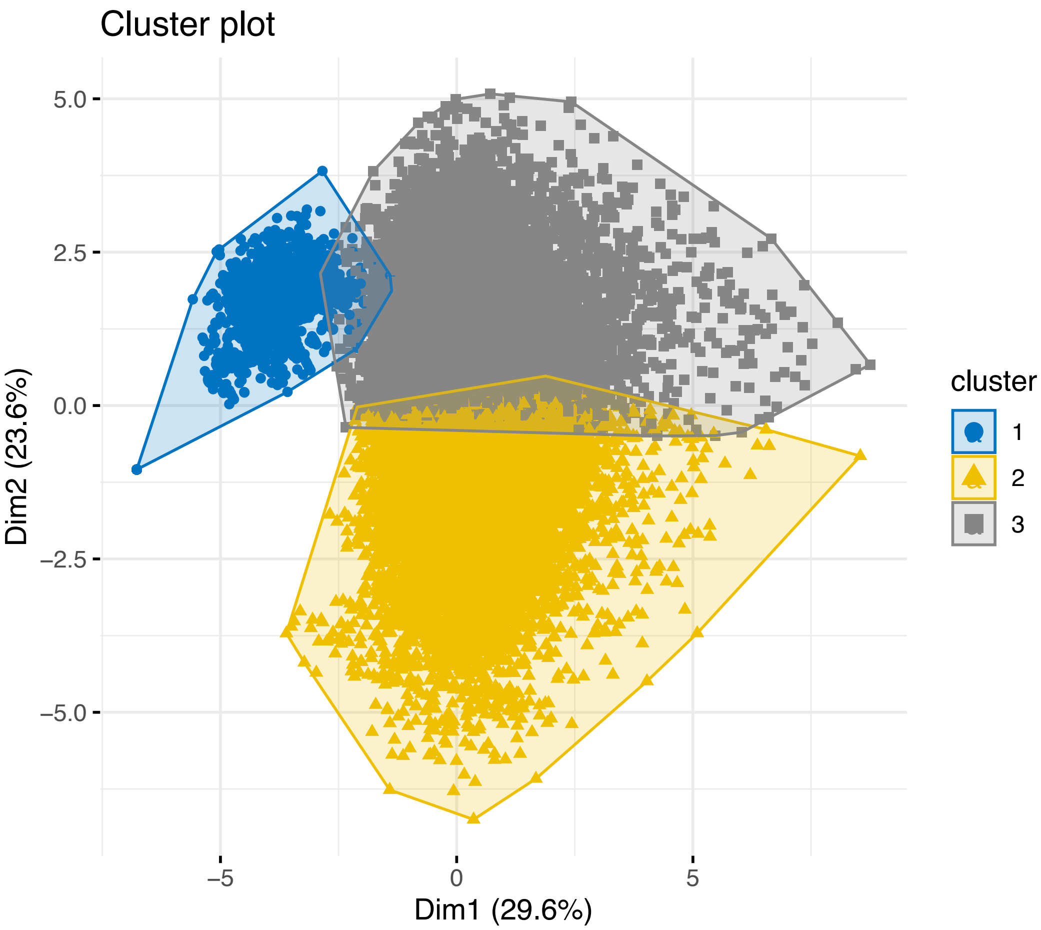


```

# Normalizing min_positive_polarity
ad.test(final_online_news_dataset$min_positive_polarity)
hist(final_online_news_dataset$min_positive_polarity)
skewness(final_online_news_dataset$min_positive_polarity)
final_online_news_dataset$min_positive_polarity <- log10(final_online_news_dataset$min_positive_polarity)

# Normalizing max_positive_polarity
ad.test(final_online_news_dataset$max_positive_polarity)
hist(final_online_news_dataset$max_positive_polarity)
skewness(final_online_news_dataset$max_positive_polarity)
final_online_news_dataset$max_positive_polarity <- sqrt(max(final_online_news_dataset$max_positive_polarity+1)

# Normalizing min_negative_polarity
ad.test(final_online_news_dataset$min_negative_polarity)
hist(final_online_news_dataset$min_negative_polarity)
skewness(final_online_news_dataset$min_negative_polarity)
#final_online_news_dataset$min_negative_polarity <- sqrt(final_online_news_dataset$min_negative_polarity)

# Normalizing max_negative_polarity
ad.test(final_online_news_dataset$max_negative_polarity)
hist(final_online_news_dataset$max_negative_polarity)
skewness(final_online_news_dataset$max_negative_polarity)
final_online_news_dataset$max_negative_polarity <- log10(max(final_online_news_dataset$max_negative_polarity+1)


# Scaling the dataset
final_online_news_dataset_scaled <- scale(final_online_news_dataset)

# Optimal number of clusters using the K-Means
my.data.matrix <- final_online_news_dataset_scaled

my.k.choices <- 3:5
n <- length(my.data.matrix[,1])
wss1 <- (n-1)*sum(apply(my.data.matrix,2,var))
wss <- numeric(0)
for(i in my.k.choices) {
```





```
> final_online_news_dataset_kmeans
K-means clustering with 3 clusters of sizes 2550, 18094, 19000

Cluster means:
  global_rate_positive_words global_rate_negative_words rate_positive_words rate_negative_words
1                  1.2568896                  2.3994463          -0.5780261          -1.5381784
2                 -0.5108389                  0.3285713          -0.5819067           0.8645136
3                  0.3177922                 -0.6349346           0.6317361          -0.6168502
  avg_positive_polarity min_positive_polarity max_positive_polarity avg_negative_polarity min_negative_polarity
1            1.12995375             0.4568186            1.37860520            2.48059056             2.0336841
2           -0.02953258             0.1316811            0.07269902           -0.32262201            -0.4647874
3           -0.12352734            -0.1867119           -0.25425576           -0.02568333             0.1696826
  max_negative_polarity
1          -2.191260422
2           0.003969467
3           0.290310028

Clustering vector:
   [1] 3 3 3 2 3 2 3 3 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 1 2 2 3 3 3 3 3 3 3 3 3 3 3 2 2 1 3 1 3 3 3 3 3 3 3 3 2 3 3 3
  [55] 3 3 2 3 1 3 3 1 3 3 3 2 2 3 3 3 2 3 2 3 3 2 3 2 3 3 2 3 2 3 3 3 2 1 3 3 3 3 2 3 3 3 2 3 2 3 2 3 3 3 3 2 3 3
 [109] 3 3 3 3 3 2 3 2 2 2 3 2 3 3 1 3 2 1 3 2 2 3 2 3 3 2 3 2 2 3 2 3 2 3 1 3 2 2 2 3 1 3 2 3 3 2 1 2 3 3 3 3 3 3
 [163] 3 3 3 3 3 3 2 2 2 3 3 3 2 2 2 3 2 3 1 3 3 2 2 3 3 3 3 3 2 3 3 3 3 3 2 2 2 3 3 2 2 3 1 3 2 3 3 2 3 2 3 2 3 3
 [217] 2 3 3 3 2 2 3 2 3 3 3 3 3 2 3 2 2 3 3 3 3 2 3 3 3 3 1 2 3 2 3 3 3 2 3 1 3 2 3 2 3 2 2 2 2 3 2 3 2 3 2 3 3 3
 [271] 3 3 2 2 1 1 3 3 2 3 2 2 3 2 3 3 2 2 3 2 3 3 2 3 3 2 2 3 3 3 3 2 3 3 3 3 2 3 3 2 3 2 3 2 2 3 3 3 2 2 2 3 3 3
 [325] 2 3 2 2 3 3 3 2 3 3 3 2 3 3 3 3 2 3 3 2 3 2 3 3 3 3 2 1 3 2 3 3 2 3 3 2 3 2 3 2 2 3 2 3 3 3 2 2 2 3 3 2 3 3
 [379] 3 3 2 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 2 2 2 3 2 3 2 2 2 2 3 3 2 2 3 2 3 3 2 3 3 2 2 2 3
 [433] 3 3 2 3 2 2 3 3 3 2 3 2 2 2 2 3 2 3 2 3 3 2 3 3 3 3 3 3 2 3 2 1 2 2 3 3 3 2 3 2 3 3 2 2 3 2 2 3 1 2 3 3 3 3
 [487] 3 2 2 3 2 2 3 3 3 3 2 3 3 2 3 3 1 2 2 3 3 2 2 3 2 3 2 2 2 1 3 3 2 3 2 2 3 2 3 3 2 2 3 2 3 3 2 2 2 3 2 2 3 2
 [541] 2 2 2 2 3 3 3 1 2 2 2 3 2 3 3 2 2 2 2 2 2 3 3 3 3 1 3 2 2 3 3 3 2 2 3 3 3 2 3 2 3 3 2 2 2 3 2 2 1 3 2 3 3 2
 [595] 2 2 2 2 2 2 2 2 2 2 2 2 3 3 2 2 2 2 2 2 1 2 3 3 3 2 3 2 3 3 3 3 3 2 3 2 3 2 3 3 2 3 3 3 1 2 3 3 3 3 3 3 3 1
 [649] 2 3 2 2 2 2 3 2 3 2 3 3 3 2 1 2 3 3 3 2 3 3 3 1 3 3 2 2 3 3 2 2 3 3 2 2 2 3 3 2 3 3 3 3 3 2 2 2 3 2 2 3 2 2
 [703] 3 3 3 3 3 3 2 3 3 3 2 3 3 3 2 3 2 3 3 3 3 2 3 3 1 3 2 3 3 2 3 3 3 2 3 2 2 3 2 2 2 3 3 2 2 3 2 3 3 2 3 2 3 3
 [757] 3 3 3 3 3 2 2 2 3 2 3 2 3 3 3 2 2 1 2 3 3 3 3 3 2 2 3 1 3 3 3 3 1 3 3 3 2 3 1 3 3 2 3 2 3 3 2 2 3 3 3 2 3 3
 [811] 3 3 2 3 3 3 3 2 3 2 3 2 3 3 2 2 2 2 3 2 3 1 2 2 2 3 3 2 3 2 2 2 2 2 3 3 3 1 2 2 3 3 2 2 3 3 1 3 1 2 3 3 3 2
 [865] 2 3 1 2 3 2 2 3 2 2 3 3 2 3 2 1 3 3 3 1 2 2 3 3 3 3 2 2 2 1 3 3 3 3 3 3 2 2 2 3 3 3 2 2 1 2 2 2 2 2 3 3 3 1
 [919] 2 2 1 3 2 2 2 3 2 2 3 3 3 3 3 3 3 1 3 2 3 2 3 3 3 3 2 3 2 3 3 2 3 3 3 3 2 3 2 2 3 3 1 3 3 2 3 2 3 2 2 3 2 3
 [973] 3 3 3 3 3 3 3 3 2 3 3 2 3 3 3 2 2 2 2 2 3 2 3 3 2 2 3 3
 [ reached getOption("max.print") -- omitted 38644 entries ]

Within cluster sum of squares by cluster:
[1]  34377.91 109821.50 118244.62
 (between_SS / total_SS =  33.8 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
```


Cluster plot
5.0
2.5
0.0
−2.5
−5.0
Dim2 (23.6%)
−5
0
5
Dim1 (29.6%)
cluster
1
2
3

Problem 3: Using Google Scholar, locate **a journal article**, which uses cluster analysis in your field of interest. **Write a summary** of the journal article and how it utilizes the cluster analysis in **two to three paragraphs**. **Cite the paper in APA format.**

Solution:

I have chosen the article 'Cluster analysis to identify elderly people's profiles: a healthcare strategy based on frailty characteristics' which discusses the use of clustering techniques to discuss the specific criteria of care needed for the population of old people in Brazil, taking into consideration their frailty conditions. The study was done using the medical records of 98 older people, and the major factors that affected the cluster formation included age, cognitive capacity, functional capacity, number of medications used, etc. As age increases, the incidence of non-communicable chronic diseases (NCCDs) increases, so the healthcare for the geriatric population must be planned, estimating the costs and risks.

The dataset consisted of different parameters that describe the overall condition of the older population, like age, cognitive status, depressive symptoms, functional capacity to perform activities of daily living (ADLs) and instrumental activities of daily living (IADLs), and the quantity of comorbidities. 98 records were selected out of the original 191 via a standard geriatric assessment. Once the dataset was compiled, Partition Cluster Analysis was used to identify the number of optimal customers. The idea was to minimize the distance between members of the same cluster and maximize the distance between different clusters. Since the sample size was small, the partition method was chosen to make 4 clusters.

The dataset was subjected to descriptive analysis for the general, male, and female populations. Cluster 1 is composed of a few less elderly people who have a high cognitive deficit and high functional loss on the ADL and IADL scales with a moderate number of diseases and medications. Cluster 2 included less elderly individuals of younger age who had better cognitive capacity and ADLs but a high number of comorbidities and a large number of medications. Cluster 3 was composed of old individuals with high cognitive defects, low functional loss on ADLs, more functional loss on IADLs, and a moderate number of medications. Cluster 4 included elderly individuals with very high cognitive defects, little functional loss on ADL and IADLs, and very few comorbidities and medications used. These data characterize the profile of this population and can be used to develop strategies to improve the impaired quality of life.

Citation:

Fattori, A., Oliveira, I. M., Alves, R. M., & Guariento, M. E. (2014). Cluster analysis to identify Elderly People's profiles: A healthcare strategy based on frailty characteristics. *Sao Paulo Medical Journal*, *132*(4), 224–230. https://doi.org/10.1590/1516-3180.2014.1324622

Extra Credit (5 points)

An academic paper from a conference or Journal will be posted to the Homework 4 content section of D2L. Review the paper and evaluate their usage of FA and LDA. In particular address the following: **(See article on Comparison of Latent Dirichlet Modeling and Factor Analysis for Topic Extraction A Lesson of History)**

a) What is the application of this paper?

Solution:

In the paper, a comparison has been drawn between the 2 known techniques for Topic Modeling - Factor Analysis (FA) and Latent Dirichlet Allocation. (LDA) It aims to see the similarity between the topics derived from the 3 different datasets using both the above mentioned techniques. It also proposes a new method to examine and compare which technique is better in terms of obtaining the coherence between the different topics extracted from the 3 datasets.

b) What is the research question the authors wish to answer in this paper?

Solution:

The research question the authors want to answer is to understand the perceived coherence of the derived topics from 3 datasets using the 2 different techniques of Factor Analysis and Latent Dirichlet Allocation. They also want to understand which is a better technique based on an evaluation method.

c) What is Latent Dirichlet Modeling and what can we learn from it?

Solution:

Probabilistic Latent Semantic Analysis (pLSA) makes use of probability for determining the hidden variables i.e. topics in large documents. Latent Dirichlet Modeling is a technique which is used in Topic Extraction / Modeling which aims to find the hidden topics in documents using Dirichlet probability distributions and allocates the distributions of those topics in the documents.
LDA is seen as an improvement to the pLSA and it can be used for Topic Modeling.

d) How does this paper utilize Factor Analysis?

Solution:

FA is used for Topic Extraction. A different type of FA called the Two-mode Factor Analysis is used in the paper. It is used to reduce the dimensionality of the data to find out the hidden topics in the texts. The approach was representing the words in the vocabulary as a linear function of the different topics (factors). The factor loadings in the linear combination formula shows how strongly each word is related to each of the topics.

e) What are the results and conclusions from this paper?

Solution:

A broad look at the results after the application of LDA and FA shows a significant difference in the topics that have been derived by the two techniques respectively.
In general it was found out that the words used by FA were more in number than by LDA to describe the same topics. For example using the hotel review dataset, LDA used 503 words to derive the topics whereas FA used 826 words. The same trend was observed on the remaining 2 datasets as well. In addition to this, the authors carry out comparable and non-comparable topic evaluation. In the comparable topic evaluation FA is rated more as it generates more coherent topics for 2 datasets. In the non-comparable topic evaluation, both LDA and FA offer similar topics and the difference is not very significant.

The authors conclude that FA has more capability to produce more coherent topics as compared to LDA. FA also is more suitable because it produces the same output every time. LDA being probabilistic in nature may produce different results unless a seed value is used. Another advantage is that FA gives a longer list of words which can describe the topic/text in a wholesome manner. On the other hand, LDA elects fewer words which have a high frequency of occurrence in the texts.

f) What other areas or fields do you think would benefit from LDA?

Solution:

- LDA can be used in online content generation systems which allows us to manage large number of different types of materials.
- It can be used in recommendation engines.
- Gene expression classification in oncology and cellular processes can also make use of LDA.

g) What other thoughts do you have on textual modeling?

Solution:

Textual Modeling / Topic Modeling is identification of words from topics present in large texts or corpus of data. It is important as it can be useful in extracting words from topics rather than from documents. Textual modeling can be used in a variety of applications like Sentiment Analysis, Recommender systems, Text summarisation, medical industry and more.

Dutta, B. (n.d.). *What is latent Dirichlet allocation (LDA) in NLP?* Analytics Steps. Retrieved December 6, 2022, from https://www.analyticssteps.com/blogs/what-latent-dirichlet-allocation-lda-nlp