MA335 Final Project

Analysis of characteristics of Alzheimer

M.D.D.Karunaratne
Registration number: 2211744

Department of Mathematics
University of Essex
United Kingdom
August 10, 2023

Contents

Word count = 1794

Abstract

This study intends to investigate the relationship among disease detection and a set of variables associated to Alzheimer's disease. The dataset contains information on the diagnosis group and characteristics. Box plots and descriptive statistics show how the two groups differ from one another. To find patterns and groupings in the data, clustering techniques like K-means and hierarchical clustering are used. Additionally, logistic regression is used to forecast the "Group" variable. In closing, feature selection using the "Boruta" algorithm demonstrates the significance of every factor.

1 Introduction

The goal of this study is to analyse a data set that involve a number of Alzheimer's disease features in order to determine how they correlate with them being detected. Variables in the data set have diagnosis group (Nondemented, Demented), gender, age, years of education, socioeconomic status, mini mental state examination score, clinical dementia rating, estimated total intracranial volume, normalised whole brain volume, and atlas scaling factor. The objective is to explore how these attributes are linked to an Alzheimer's disease diagnosis which might help ensue early identification of the condition.

2 Preliminary Analysis (Question 1)

After following pre-processing steps such as removing missing values, the data set has 317 records of Alzheimer's disease features. It has 127 "Demented" and 190 "Nondemented" records of Alzheimer diagnosis. Below is the dataset summary,

	M/F	Age	EDUC	SES	MMSE	CDR	eTIV	nWBV	ASF
Minimum	1.0	60.0	6.0	1.0	4.0	0.0	1106	0.644	0.876
1st Quartile	1.0	71.0	12.0	2.0	27.0	0.0	1358	0.7	1.098
Median	1.0	76.0	15.0	2.0	29.0	0.0	1476	0.732	1.189
Mean	1.432	76.72	14.62	2.546	27.26	0.2729	1494	0.7306	1.192
3rd Quartile	2.0	82.0	16.0	3.0	30.0	0.5	1599	0.757	1.293
Maximum	2.0	98.0	23.0	5.0	30.0	2.0	2004	0.837	1.587

Table 1: Summary of the data set

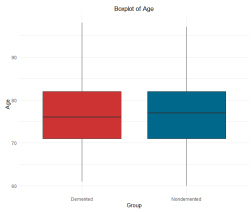
Table 1 presents the descriptive statistics of the variables of the data set. Apart from "M.F" and "SSE" rest are numerical variables and their minimum, 1st quartile, median, mean, 3rd quartile and maximum can be seen in the table.The two categorical characteristics of in the data set is analyzed in following frequency table.

	Low	Low Medium	Medium	Medium Medium	High
Male	28	66	48	31	7
Female	39	30	27	41	0

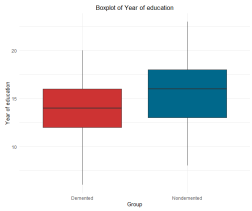
Table 2: Summary categorical variables

Next, the numerical variables are visualized in box plots to analyze their statistics. To begin with, the range and interquartile range (IQRs)of age's demented and non-demented groups are comparable, but the demented group has a slightly lower mean. The demented group's average and range for years of education variable are smaller than those of the non-demented group. Additionally, the non-demented group exhibits a wider IQR than the

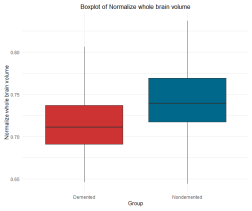
demented group, with the mean of non-demanted group are closer to the third quartile of the demented group for year of education parameter. Both groups have outliers on the mini-mental state evaluation while the median and 3Q are more closer to maximum for both groups. Each group contains one outlier in the boxplot of clinical dementia rating. This plot depicts that for the non-demanted group the values of clinical demantia rating is almost 0 for all the records while demented group has records ranging from 0.5 to 1 and a outlier in range 2. The estimated total intracranial volume shows that the demented group has outliers, whereas the non-demented group has a wider range. The average of both groups, though, are close. Next, it can be seen that the demented group differs from the non-demented group in terms of lower quartiles, averages, and upper quartiles for normalised whole brain volume attribute. For atlas scaling factor, a greater lower quartile and a lower upper quartile are seen in the demented group, while it still reveals identical averages. Overall, the contrasts between the demented and non-demented groups across a variety of factors, such as age, education, mini mental state examination, and brain sizes, are generally revealed by boxplots in figure 1.



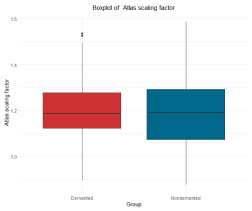
(a) Boxplot of Age



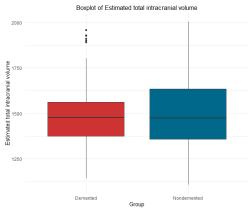
(b) Boxplot of Year of education



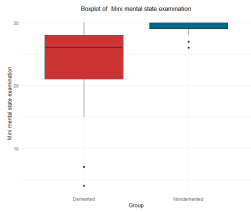
(c) Boxplot of Normalize whole brain volume



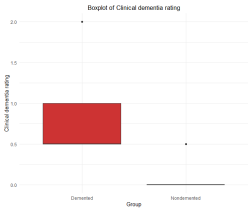
(d) Boxplot of Atlas scaling factor



(e) Boxplot of Estimated total intracranial volume



(f) Boxplot of Mini mental state examination



(g) Boxplot of Clinical dementia rating

Figure 1: Boxplots of numerical variables

3 Clustering (Question 2)

3.1 Similarity Measure

A method for calculating the similarity/dissimilarity between each pair of data is necessary for the clustering of observations. I plotted a distance matrix produced by calculating the separation between each pair of data. The plot is attached in the appendix at, 2.1. Clusters of low distances along a diagonal line can be seen, indicating comparable or high association of variables. Top-right and bottom-left corners have occasional clusters of high distances indicating poor correlations. Overall, the plot depicts clusters indicating distinct subsets of data with common patterns or traits, and unique groups or outliers.

3.2 K-Means Clustering

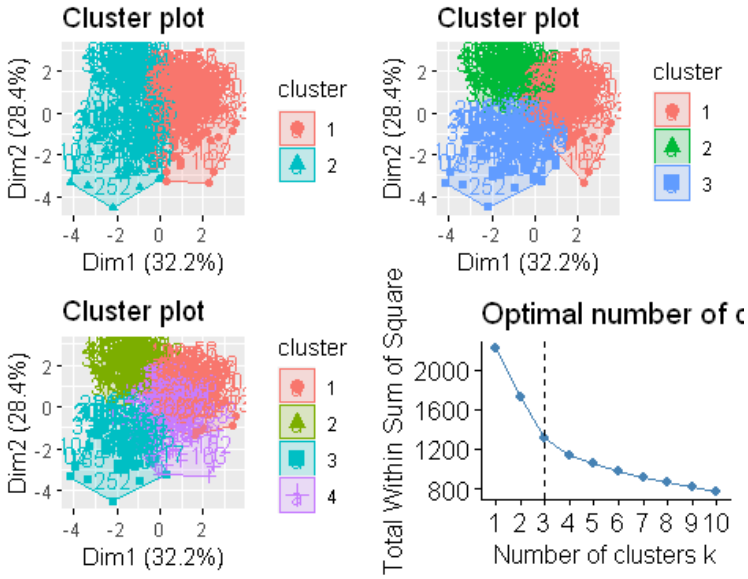


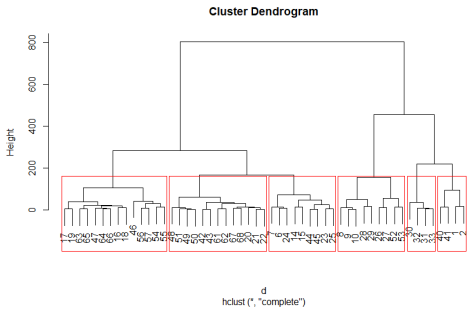
Figure 2: K-means clustering

Due to the simplicity and interpretability, K-means clustering is a commonly used technique. It employs a technique to minimise the within-cluster sum of squares. Clusters are defined by their centroids and each data point is assigned to a cluster that has the closest centroids. To begin the analysis, I tried multiple values(k=2,3,4). The figure 2 presents three k-means clusters I built. In the first plot, two precisely separated clusters can be seen. Next, I created 3 clusters where data points are clearly separated into 3 clusters. Furthermore, 4 clusters were built to determine if more clusters can be formed. However, the plot depicts overlapped clusters. In spite of that, I used "fviz_nbclust" function to determine

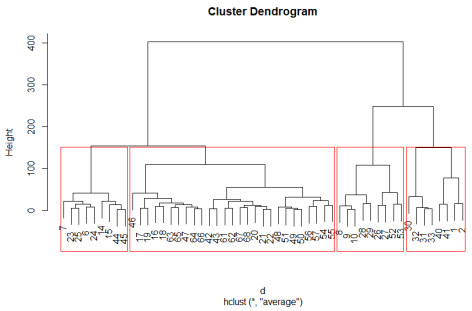
the optimal number of clusters. The elbow point gets spotted at three clusters in the plot. As a result, I can state that the optimum amount of clusters is three.

3.3 Hierarchical Clustering

Hierarchical clustering regard each observation as a separate cluster and it determines the pair of clusters that are most nearby and merges both of the most equivalent clusters. It produces a dendrogram depicting the cluster's hierarchical interaction. The distance between two clusters are calculated using the length of a single line from one to the other. I used the Euclidean distance.



(a) Dendogram of "complete" linkage method



(b) Dendogram of "average" linkage method

Figure 3: Dendograms

Next, linkage methods were applied. The linkage methods of "complete" and "average" provide clearer clusters which respectively consider maximum distance and average distance between any member of one cluster and any member of the other. The height of the dendrogram which represents the dissimilarity of two groups was used to decide the number of clusters. When the height of the line which separates a cluster started to drop, a horizontal line can be drawn to identify the number of clusters. In the 2 dendograms, you can see that 6 clusters are obtained for "complete" method while 4 clusters are identified for "average" method.

Moreover, I calculated the mean values of the variables to each cluster of "average" linkage method. The first cluster has small negative values for various variables, such as Age, EDUC, MMSE, CDR, eTIV, nWBV, and ASF. The second cluster combines positive and negative values, with a positive EDUC number indicating higher education levels. The third cluster has high positive values, with older individuals, higher education, and higher MMSE scores. The fourth cluster has lower positive values but a higher CDR score, indicating a higher clinical dementia grade.

cluster	Age	EDUC	MMSE	CDR	eTIV	nWBV	ASF
1	-0.03286855	-0.04210378	0.03947186	-0.04079384	-0.0100009	0.0252915	0.01167504
2	-1.052660243	0.47315221	-5.63598770	1.90276407	1.19031721	-1.35416116	-1.17143392
3	2.25715361	2.86478037	0.53648840	-0.71405246	-0.1293349	-0.9254870	0.01475689
4	1.83008085	1.15647454	-0.84476499	3.64730842	0.3658824	-0.7767633	-0.43632975

Table 3: Summary categorical variables

4 Logistic Regression (Question 3)

Logistic regression can be applied to solve binary classification problems. Following is the result of the model.

> summary(glm.fit)

Call: glm(formula = Group Age + EDUC + MMSE + CDR + eTIV + nWBV + M.F + SES + ASF, family = binomial, data = proj_data)

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.381e-04	-2.100e-08	-2.100e-08	2.100e-08	1.196e-04

Coefficients:

	Estimate Std.	Error	z value	Pr(>|z|)
(Intercept)	2.120e+03	5.225e+06	0.000	1.000
Age	-6.473e+00	7.727e+03	-0.001	0.999
EDUC	3.828e+00	1.205e+04	0.000	1.000
MMSE	-6.396e+00	1.979e+04	0.000	1.000
CDR	3.304e+02	1.615e+05	0.002	0.998
eTIV	-4.272e-01	1.774e+03	0.000	1.000
nWBV	-9.496e+02	2.294e+06	0.000	1.000
M.F	4.176e+01	6.482e+04	0.001	0.999
SES	-1.505e+01	4.034e+04	0.000	1.000
ASF	-2.403e+02	2.645e+06	0.000	1.000

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 4.2685e+02 on 316 degrees of freedom

Residual deviance: 5.4985e-08 on 307 degrees of freedom

AIC: 20

The logistic regression model has insignificant p-values, indicating lack of relationships with the dependent variable. The residual deviance is near zero, indicating inadequate fit, and the AIC value is high, indicating poor performance. The model is neither accurate nor

acceptable for predicting the "Group" variable. Further investigations reveal conflicting results to the model summary. The confusion matrix gives the summary of actual vs predicted as follows,

	Nondemanted	Demanted
Nondemanted	190	0
Demanted	0	127

Confusion matrix says all labels are correctly predicted, and the mean value of correct predictions is 1. This could be an overfitting scenario as the summary of the model implies that none of the variables are within the significance level. Therefore, implementing other classification models or improving the dataset is suitable to check the validity of results.

5 Feature Selection (Question 4)

First, I used "Boruta" wrapper algorithm which uses a random forest classification technique. A machine learning model is trained using the original variables and the shadow variables that correspond to them. On the basis of the model's performance metric, each feature's relevance is decided. However, this technique confirmed all the variables as important variables. To check how much important each variable is to predict the "Group", I built a LDA model. The ROC curve was used to determine variable importance.

- CDR - 0.9963
- MMSE - 0.8877
- nWBV - 0.6970
- M.F - 0.6387
- EDUC - 0.6238
- SES - 0.5917
- Age - 0.5344
- eTIV - 0.5048
- ASF - 0.5046

CDR and MMSE are the most significant features in the LDA model, with eTIV and ASF being the least important ones. All variables have over 50% importance in predicting and can be assumed that all the variables are important to consider as features.

6 Conclusion

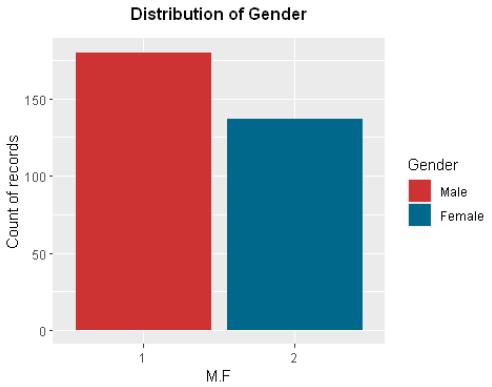
The study examined the relationship between Alzheimer's disease symptoms and diagnosis. Results showed three unique groupings of k-means clustering, and logistic regression revealed no statistically significant factors. The Boruta algorithm was used for feature selection. To conclude, it can be stated that further research using sophisticated techniques and a larger dataset is more appropriate to analyze the data set.

Appendix

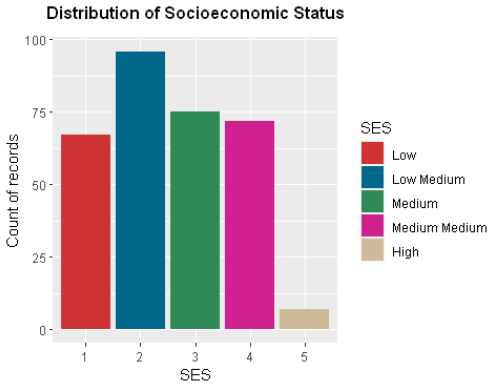
1 Exploratory Data Analysis

1.1 Categorical Variables

Barplots are a suitable way to visualize the distribution of categorical variables. Following are the bar plots of categorical variables,



(a) Bar plot of gender distribution



(b) Bar plot of socioeconomic Status distribution

Figure 4: Bar plots of categorical variables

The bar plot 4a, demonstrates the gender distribution. According to the plot, more males can be seen in the population. Bar plot 4b demonstrates the distribution of socioeconomic Status. "Low-medium" economic status have the greatest in population while "High" socioeconomic Status has the least participation in the population. "Low", "Medium" and "Medium medium" socioeconomic Status have a good participation in the population which are closer to similar proportions of participation.

2 Clustering

Clustering is a valuable data analysis technique for pattern recognition, detecting intrinsic patterns and groupings without prior knowledge or labeling. It improves data analysis by identifying parallels and dissimilarities among data points, providing insights into the dataset's structure. Clustering also helps in recognizing associations, dependencies, or groups, aiding in a deeper understanding of the material. It can also be used as a preprocessing step to minimize dataset dimensions by grouping related data points together.

2.1 Similarity Measure

Following is the plot of distance matrix as stated in section 3.1. To have a more clear picture of the distances, I divided the dataset into 4 data frames and calculated distances.

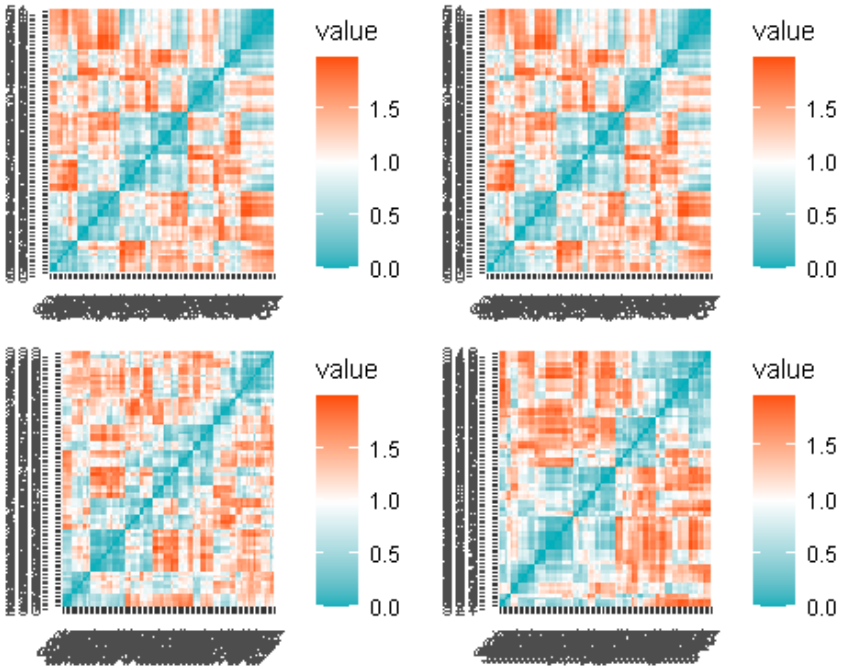


Figure 5: K-means clustering

2.2 Hierarchical Clustering

As stated in section 3.3, "complete" and "average" linkage methods demonstrates more clear clusters. Following is the clusters of "complete" linkage method.

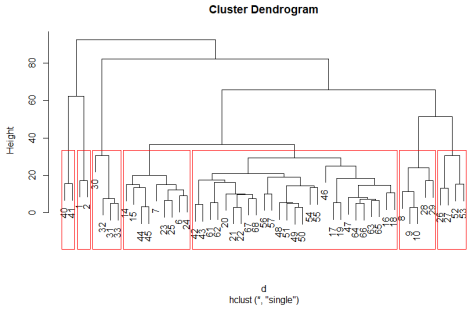
Cluster 1 consists of older individuals with higher education levels, higher clinical dementia

cluster	Age	EDUC	MMSE	CDR	eTIV	nWBV	ASF
1	2.086325	1.8397969	0.7091451	1.9027641	2.8401142	1.3228654	0.31786803
2	2.598812	2.8647804	0.7091451	0.5943558	1.0428649	1.1129026	2.35849810
3	1.061350	1.1564745	0.7091451	0.5943558	0.5865972	2.7926055	2.83106507
4	1.061350	1.1564745	-1.1037500	4.5195806	1.4546186	0.7454675	1.32744291
5	1.958203	-0.2101701	0.7091451	1.9027641	-0.1645751	0.3255418	2.45873958
6	2.726934	1.8397969	-0.3267950	4.5195806	1.2042278	0.7717129	0.04578402

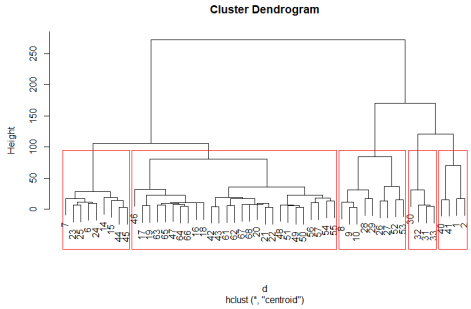
Table 4: Summary categorical variables

ratings, and larger estimated total intracranial capacity and normalised whole brain volume. Cluster 2 has mixed positive and negative values, with a larger atlas scaling factor. Cluster

3 includes younger individuals with lower academic achievements, lower Mini Mental State Examination scores, lower clinical dementia ratings, and smaller estimated total intracranial volume. Cluster 4 has a higher clinical dementia grade. Clusters 5 and 6 have mixed positive and negative values, with distinct characteristics such as older age and lower Mini Mental State Examination scores, and better education and higher clinical dementia ratings.
In section 3.3, "complete" and "average" linkage methods were discussed. Furthermore, "single" and "centroid" linkage methods can be applied in hierarchical clustering. Following figure 6a and 6b demonstrates the clusters. It is clearly visible that clusters "single" and "centroid" linkage methods can create 7 and 5 clusters respectively.



(a) Dendogram of "Single" linkage method



(b) Dendogram of "Centroid" linkage method

Figure 6: Dendograms of linkage methods

3 R Code

Pre-processing

```
#Set working directory
setwd("M:/MA335/Assignment")
#Read csv file into a data frame
proj_data <-  read.csv("project data.csv")
#Count number of rows in the data frame
nrow(proj_data)
#Convert M/F column to a factor variable
proj_data$M.F <- as.factor(proj_data$M.F)
#Convert the categorical variables into numerical values
proj_data$M.F <- as.numeric(proj_data$M.F)
#Check the structure of data frame
str(proj_data)
#Remove rows with "Group" value equal to Converted
proj_data <- subset(proj_data, !(Group == "Converted"))
#Count number of rows left in the data frame
nrow(proj_data)
#Remove rows with missing values
proj_data <- na.omit(proj_data)
#Count number of rows left in the data frame
nrow(proj_data)
```

Question 1

```
#Obtain a summery of the data frame
summary(proj_data)
#Count "Demented" observations
nrow(subset(proj_data, (Group == "Demented")))
#Count "Non-demented" observations
nrow(subset(proj_data, (Group == "Nondemented")))
#Draw a bar plot for the "Group" dependent variable
ggplot(proj_data, aes(x=as.factor(Group), fill=as.factor(Group) )) +
  geom_bar( ) +
  scale_fill_manual(values = c("brown3", "deepskyblue4") )+
  ylab("Count of records") +
  xlab("Group") +
  ggtitle("Disrtibution of  diagnosis of Alzheimer") +
  theme(plot.title = element_text(size = 12, hjust = 0.5, face = "bold", vjust = 1.5 , margin
= margin(b = 10))) +
  guides(fill= "none")
#Draw a frequency table for categorical variables of M.F and SES
table_proj <- table(proj_data$M.F, proj_data$SES); table_proj
#Draw a barplot for Gender variable
ggplot(proj_data, aes(x = as.factor(M.F), fill = as.factor(M.F))) + geom_bar() +
  scale_fill_manual(values = c("brown3", "deepskyblue4"), labels = c("Male", "Female")) +
  labs(fill = "Gender") + ylab("Count of records") +  xlab("M.F") +
  ggtitle("Distribution of Gender") +
  theme(plot.title = element_text(size = 12, hjust = 0.5, face = "bold", vjust = 1.5, margin
= margin(b = 10))) +
  guides(fill = guide_legend(title = "Gender"))  # Show legend with custom title and labels
```

```
#Draw a barplot for Socioeconomic Status variable
ggplot(proj_data, aes(x = as.factor(SES), fill = as.factor(SES))) +  geom_bar() +
  scale_fill_manual(values = c("brown3", "deepskyblue4", "seagreen", "violetred", "wheat3"),
                    labels = c("Low", "Low Medium", "Medium", "Medium Medium", "High")) +
  labs(fill = "SES") +  ylab("Count of records") +  xlab("SES") +
  ggtitle("Distribution of Socioeconomic Status") +
  theme(plot.title = element_text(size = 12, hjust = 0.5, face = "bold", vjust = 1.5, margin
= margin(b = 10))) +
  guides(fill = guide_legend(title = "SES"))  # Show legend with custom title and labels
# Create box plots for each numerical variable
#Numerical variable : Age
ggplot(proj_data, aes(x = as.factor(Group), y = Age, fill = as.factor(Group))) +  geom_boxplo
t() +
  xlab("Group") +  ylab("Age") +  scale_fill_manual(values = c("brown3", "deepskyblue4")) +
  ggtitle(paste("Boxplot of Age")) +  theme_minimal() +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5))
#Numerical variable : Year of education
ggplot(proj_data, aes(x = as.factor(Group), y = EDUC, fill = as.factor(Group))) +  geom_boxpl
ot() +
  xlab("Group") +  ylab("Year of education") +  scale_fill_manual(values = c("brown3", "deeps
kyblue4")) +
  ggtitle(paste("Mini mental state examination")) +  theme_minimal() +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5))
#Numerical variable : Year of education
ggplot(proj_data, aes(x = as.factor(Group), y = MMSE, fill = as.factor(Group))) +
  geom_boxplot() +  xlab("Group") +  ylab("Mini mental state examination") +
  scale_fill_manual(values = c("brown3", "deepskyblue4")) +  ggtitle(paste("Boxplot of  Mini
mental state examination")) +  theme_minimal() +  theme(legend.position = "none", plot.title
= element_text(hjust = 0.5))
#Numerical variable : Clinical dementia rating
ggplot(proj_data, aes(x = as.factor(Group), y = CDR, fill = as.factor(Group))) +
  geom_boxplot() +  xlab("Group") +  ylab("Clinical dementia rating") +
  scale_fill_manual(values = c("brown3", "deepskyblue4")) +  ggtitle(paste("Boxplot of Clinic
al dementia rating")) +
  theme_minimal() +  theme(legend.position = "none", plot.title = element_text(hjust = 0.5))
#Numerical variable : Estimated total intracranial volume
ggplot(proj_data, aes(x = as.factor(Group), y = eTIV, fill = as.factor(Group))) +  geom_boxpl
ot() +
  xlab("Group") +  ylab("Estimated total intracranial volume") +  scale_fill_manual(values =
c("brown3", "deepskyblue4")) +  ggtitle(paste("Boxplot of Estimated total intracranial volum
e")) +  theme_minimal() +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5))
#Numerical variable : Normalize whole brain volume
ggplot(proj_data, aes(x = as.factor(Group), y = nWBV, fill = as.factor(Group))) +  geom_boxpl
ot() +
  xlab("Group") +  ylab("Normalize whole brain volume") +  scale_fill_manual(values = c("brow
n3", "deepskyblue4")) +
  ggtitle(paste("Boxplot of Normalize whole brain volume")) +  theme_minimal() +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5))
#Numerical variable : Atlas scaling factor
ggplot(proj_data, aes(x = as.factor(Group), y = ASF, fill = as.factor(Group))) +  geom_boxplo
t() +
  xlab("Group") +  ylab("Atlas scaling factor") +  scale_fill_manual(values = c("brown3", "de
epskyblue4")) +
  ggtitle(paste("Boxplot of  Atlas scaling factor")) +  theme_minimal() +
```

```
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5))
#Store numerical variables into a vector
numerical_vars <- c("Age", "EDUC", "MMSE", "CDR", "eTIV", "nWBV", "ASF")
```

Question 2

```
#install.packages("factoextra")
library(factoextra)
#Compute distance matrix
#To obtain more clear plot, the data set was divided into 4 sets and compute the distance mat
rices
data1 <- proj_data[0:53, numerical_vars]
distance.corr1 <- get_dist(data1, stand = TRUE, method = "pearson")
d1 <- fviz_dist(distance.corr1, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E
07"))
data2 <- proj_data[81:160, numerical_vars]
distance.corr2 <- get_dist(data2, stand = TRUE, method = "pearson")
d2 <- fviz_dist(distance.corr1, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E
07"))
data3 <- proj_data[161:240, numerical_vars]
distance.corr3 <- get_dist(data3, stand = TRUE, method = "pearson")
d3 <- fviz_dist(distance.corr3, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E
07"))
data4 <- proj_data[241:317, numerical_vars]
distance.corr4 <- get_dist(data4, stand = TRUE, method = "pearson")
d4 <- fviz_dist(distance.corr4, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E
07"))

#Draw the distance plots
library(gridExtra)
grid.arrange(d1, d2, d3 , d4,  ncol = 2, nrow=2)

#K-means clustering
#Make a subset of the proj_data dataset that only contains the columns that relate to numeric
al variables.
numeric_data <- proj_data[, numerical_vars]
#Scaling to standardize the values of the variables
proj_data_scaled <- scale(numeric_data)
set.seed(123)
#Perform k-means clustering when k=2
kmeans2 <- kmeans(proj_data_scaled, centers = 2, nstart = 20)
#Perform k-means clustering when k=3
kmeans3 <- kmeans(proj_data_scaled, centers = 3, nstart = 20)
#Perform k-means clustering when k=4
kmeans4 <- kmeans(proj_data_scaled, centers = 4, nstart = 20)
#Check what's inside the first cluster created
kmeans2
#Check the structure of the cluster
str(kmeans2)
#Visualize kmeans2 cluster
f1<- fviz_cluster(kmeans2, data = proj_data_scaled)
#Visualize kmeans3 cluster
f2<- fviz_cluster(kmeans3, data = proj_data_scaled)
```

```
#Visualize kmeans4 cluster
f3<- fviz_cluster(kmeans4, data = proj_data_scaled)
#Visualize the optimal number of clusters of k-means clustering
f4<- fviz_nbclust(proj_data_scaled, kmeans, method = "wss")+
  geom_vline(xintercept = 3, linetype = 2)
#Draw all the plots in one canvas
grid.arrange(f1, f2, f3, f4 , nrow=2, ncol = 2)
#Hierarchical clustering
#Calculate the distance matrix
d <- dist(data1, method = "euclidean")
#Apply hierarchical clustering for different linkage methods
#Single linkage method
fit.single <- hclust(d, method="single")
#Complete linkage method
fit.complete <- hclust(d, method="complete")
#Average linkage method
fit.average <- hclust(d, method="average")
#Centroid linkage method
fit.centroid <-hclust(d, method="centroid")
#Print the dendogram of single linkage method
plot(fit.single)
#Cut tree into k=7 clusters
groups.fit.single <- cutree(fit.single, k=7)
#Draw dendogram with red borders around the 7 clusters
rect.hclust(fit.single, k=7, border="red")
#Checking how many observations are in each cluster of single linkage method
table(groups.fit.single)
#Print the dendogram of complete linkage method
plot(fit.complete)
#Cut tree into k=6 clusters
groups.fit.complete <- cutree(fit.complete, k=6)
#Draw dendogram with red borders around the 6 clusters
rect.hclust(fit.complete, k=6, border="red")
#Checking how many observations are in each cluster of complete linkage method
table(groups.fit.complete)
#Print the dendogram of average linkage method
plot(fit.average)
#Cut tree into k=4 clusters
groups.fit.average <- cutree(fit.average, k=4)
#Draw dendogram with red borders around the 4 clusters
rect.hclust(fit.average, k=4, border="red")
#Checking how many observations are in each cluster of average linkage method
table(groups.fit.average)
#Print the dendogram of centroid linkage method
plot(fit.centroid)
#Cut tree into k=3 clusters
groups.fit.centroid <- cutree(fit.centroid, k=3)
#Draw dendogram with red borders around the 3 clusters
rect.hclust(fit.centroid, k=3, border="red")
#Checking how many observations are in each cluster of centroid linkage method
table(groups.fit.centroid)
#Check the mean values of each attribute to the cluster in centroid linkage method
aggregate(data1, by=list(cluster=groups.fit.centroid), mean)
#Check the  max distance values of each attribute to the cluster in complete linkage method
aggregate(data1, by=list(cluster=groups.fit.complete), max)
```

```
#Check the mean values of each attribute to the cluster in average linkage method
aggregate(data1, by=list(cluster=groups.fit.average), mean)
```

Question 3

```
attach(proj_data)
#Categorical variable is encoded with binary values
proj_data$Group <- ifelse(proj_data$Group == "Demented", 1, 0)
#Build logistic regression model
glm.fit<-glm(Group~Age+EDUC+MMSE+CDR+eTIV+nWBV+M.F+SES+ASF,data=proj_data,family=binomial)
#check summary of the logistic regression model
summary(glm.fit)
#contrasts(as.factor(Group))
#Generate predicted probabilities of the logistic regression model
glm.probs <- predict(glm.fit,type="response") #Pr(Y=1|X)
#glm.predicted <- rep(0,nrow(proj_data))
#glm.predicted[glm.probs>0.5]=1
#Initializes a vector with zeros for each row
glm.predicted <- rep(0,nrow(proj_data))
#Assign values of the vector according to the predicted probabilities
glm.predicted[glm.probs>0.5]=1
#Generate a contingency table
table(glm.predicted, proj_data$Group)
#Calculate the accuracy of the predictions
mean(glm.predicted==proj_data$Group)
```

Question 4

```
#Random forest
#boruta
#install.packages("Boruta")
library(Boruta)
#Creates boruta object
boruta1 <- Boruta(proj_data$Group~., data=proj_data, doTrace=1)
#Derives the result from the Boruta object, highlighting the significance of each variable
decision<-boruta1$finalDecision
#Filters variables that have the "Confirmed" label when filtering the decision variable
signif <- decision[boruta1$finalDecision %in% c("Confirmed")]
#Prints the selected significant variables
print(signif)
#Generates a graphic of the variable importance based on the Boruta analysis.
plot(boruta1, xlab="", main="Variable Importance")
```

```
#Generates the attribute statistics needed for the Boruta analysis.
attStats(boruta1)
library(caret)
#install.packages("recipes")
library(MASS)
#Converts the "Group" variable to a factor
proj_data$Group <- as.factor(proj_data$Group)
#Use cross-validation
trControl <- trainControl(method = "cv", number = 10)
#Trains an LDA model
lda.fit <- train(Group~M.F+Age+EDUC+SES+MMSE+CDR+eTIV+nWBV+ASF,
                 method = "lda",
                 trControl = trControl,
                 metric = "Accuracy",
                 data = proj_data)
#computes the variable importance
importance <- varImp(lda.fit, scale=FALSE);importance
```