# Prediction of Alzheimer’s Disease

**Prepared By :**

**Ria Tilak Bhattacharya**

**2214122**

# Prediction of Alzheimer’s Disease

Ria Tilak Bhattacharya

June 20, 2023

**Abstract**

The most common cause of dementia is Alzheimer’s disease. This study aims to determine if the variables in the dataset can accurately predict whether a patient has dementia or not. Various different analyses have been performed on the data and a model has been fitted to the data given to us to provide predictions with a 99 % accuracy rate.

## Contents

Word Count 1810

# 1 Introduction

Dementia is a form of memory loss and other cognitive impairments severe enough to interfere with daily life, most frequently caused by Alzheimer's disease. Aging is the biggest risk factor, and those 65 and older comprise the majority of Alzheimer's patients. There is no cure for dementia, but a global effort is currently underway to discover new ways to treat the illness, postpone its onset, and stop it from developing.[1] In this study, we are going to examine a dataset that has various Alzheimer's characteristics. The goal is to determine whether there is a connection between such traits and the diagnosis, which is whether the patient is suffering from Alzheimer's (demented) or not (non-demented).

# 2 Data-set Explanation

The dataset consists of 373 records in total and 10 variables relating to Alzheimer's characteristics. Here Group is the target variable whose value we aim to predict and the rest nine are our predictor variables.146 patients were classified as Demented, 190 as Non-Demented, and 37 as Converted.The variables in the dataset are **Group**(Nondemented, Demented, Converted), **M/F**(Gender), **Age**, **EDUC** (Years of education),**SES**(Socioeconomic Status:1-5, 1-low, 5-high), **MMSE**(Mini-mental state examination), **CDR** (Clinical dementia rating), **eTIV**(Estimated total intracranial volume), **nWBV**(Normalize whole brain volume), **ASF**(Atlas scaling factor).

# 3 Preliminary Analysis

The rows with Group="Converted" were removed from the data which left us with 336 records. Next, we checked for missing values in the data using the aggr function from the VIM library. SES has the maximum missing data followed by MMSE. (Appendix Figure 3) The other variables do not have any missing data. There are two methods for resolving this problem. We can remove the rows with missing data but we will be left with only 317 records. The second is the imputation method which we have used in our analysis to replace the missing values with substitute values. To impute the missing data, we used the random forest method from the mice library.[2] The variable M/F has been converted into numeric values where 1 is for females and 2 is for males. In the next section, we are going to present the methods and analysis of the four tasks given in the project assignment.

# 4 Analysis

## 4.1 Exploratory Data Analysis

In this section, we will analyze the data using descriptive statistics through graphical and numerical representations.

- **Graphical Representation**: In Figure 1, Figure 1a demonstrates that there are more men than women among the demented and more women than men among the non-demented. In Figure 1b we observe that the age group from 73-80 has the highest number of dementia cases. Next, we observe a boxplot in Figure 1c where we see the demented group has a higher median than the non-demented group for SES levels 1, 2, 4, and 5. The data is more dispersed for the non-demented in SES status 1,2 and 4 however there is very less data for the non-demented group in SES 5. The median is higher for the non-demented group only at SES level 3. We also observe some outliers in the data in SES levels 1,2,4 for the demented and level 3 for both the demented and non-demented groups. In the scatter plot in Figure 1d for SES vs nWBV the values are present for both demented and non-demented groups from SES levels 1-4 but the values for level 5 are primarily for the demented group. The non-demented group has values for higher nWBV, while the majority of the values for the demented group are for lower nWBV values.

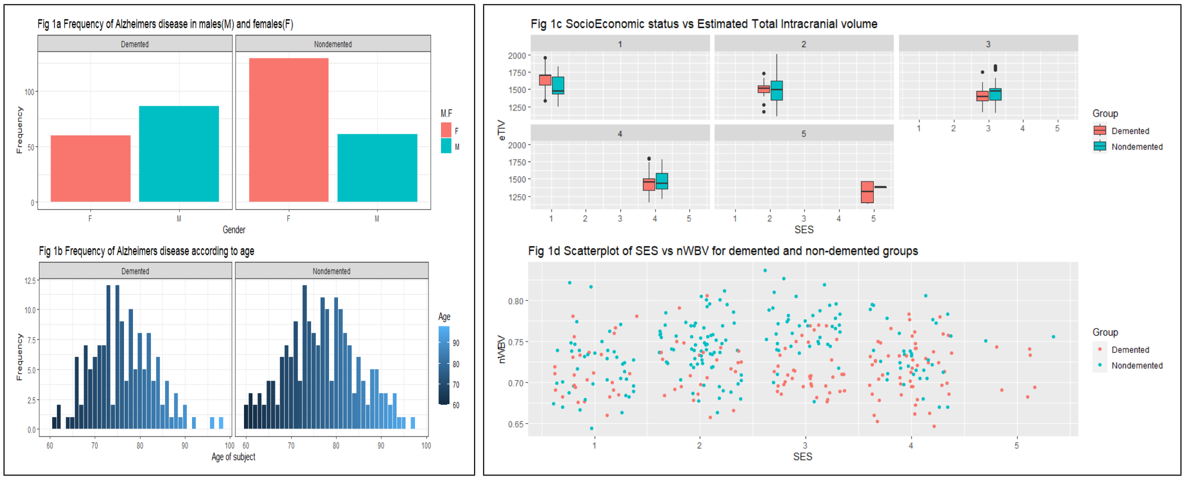

Figure 1: Frequency of Alzheimer's disease in male/female and distribution according to age(1a and 1b) and Boxplot for SES vs eTIV and Violin scatter plot for SES vs nWBV(1c and 1d)

- **Numerical Analysis**: We will explore the numerical characteristics and correlations between the different variables. We have presented a summary table of the numerical characteristics of the data in Table 1. We can observe the minimum and maximum values for each variable, and the mean and median. The 1st Quartile is the value under which 25% of the data-points are found and the 3rd Quartile is the value under which 75% of the data-points are found. The interquartile range, which is the difference between the first and third quartiles, reveals how the data is distributed around the median. A small interquartile range which is the case for most of our variables indicates that the data is clustered around the median. The data is more skewed when the interquartile range is bigger.[3]

- **Correlation matrix of the nine predictor variables**: We will be using the correlation matrix displayed in Figure 2a(Figure 2) to identify how closely the variables are related to each other. Most of the variables have a low positive or negative correlation indicating a weak relationship between them. ASF has a negative correlation of -0.99 with eTIV and -0.55 with M.F. SES has a negative correlation of -0.72 with EDUC. M.F. has a positive correlation of 0.56 with eTIV and MMSE has a positive correlation of 0.37 with nWBV.

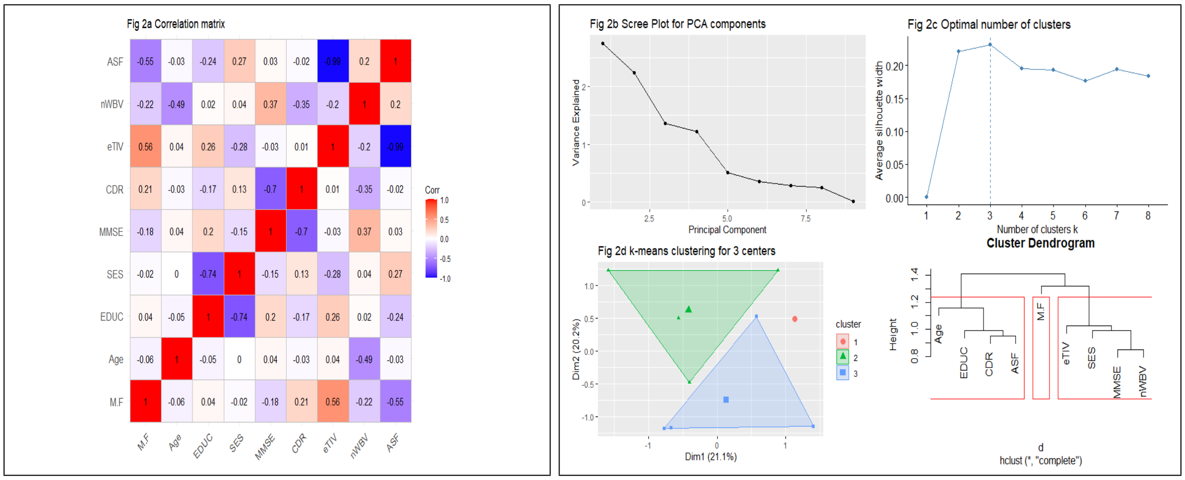
Fig 2a Correlation matrix

| | M.F | Age | EDUC | SES | MMSE | CDR | eTIV | nWBV | ASF |
|---|---|---|---|---|---|---|---|---|---|
| ASF | -0.55 | -0.03 | -0.24 | 0.27 | 0.03 | -0.02 | -0.99 | 0.2 | 1 |
| nWBV | -0.22 | -0.49 | 0.02 | 0.04 | 0.37 | -0.35 | -0.2 | 1 | 0.2 |
| eTIV | 0.56 | 0.04 | 0.26 | -0.28 | -0.03 | 0.01 | 1 | -0.2 | -0.99 |
| CDR | 0.21 | -0.03 | -0.17 | 0.13 | -0.7 | 1 | 0.01 | -0.35 | -0.02 |
| MMSE | -0.18 | 0.04 | 0.2 | -0.15 | 1 | -0.7 | -0.03 | 0.37 | 0.03 |
| SES | -0.02 | 0 | -0.74 | 1 | -0.15 | 0.13 | -0.28 | 0.04 | 0.27 |
| EDUC | 0.04 | -0.05 | 1 | -0.74 | 0.2 | -0.17 | 0.26 | 0.02 | -0.24 |
| Age | -0.06 | 1 | -0.05 | 0 | 0.04 | -0.03 | 0.04 | -0.49 | -0.03 |
| M.F | 1 | -0.06 | 0.04 | -0.02 | -0.18 | 0.21 | 0.56 | -0.22 | -0.55 |

Figure 2: Correlation matrix of the nine predictor variables(Fig 2a) and Scree Plot for PCA components(Fig 2b), Optimal number of clusters(Fig 2c), k-means cluster for 3 centers(Fig 2d), and Cluster dendrogram

## 4.2 Clustering Algorithms

**Principal component analysis(PCA)** is a dimensionality reduction method that breaks down the original features into a new collection of uncorrelated variables. To lower the dimensionality of the data and eliminate redundant information, we used PCA before k-means and hierarchical clustering. This can help improve the performance of the clustering algorithm. We can determine the number of principal components to retain based on a scree plot or variance

|  | Age | EDUC | SES | MMSE | CDR | eTIV | nWBV | ASF |
|---|---|---|---|---|---|---|---|---|
| Min | 60.00 | 6.0 | 1.000 | 4.00 | 0.0000 | 1106 | 0.6440 | 0.876 |
| 1st Quartile | 71.00 | 12.0 | 2.000 | 26.00 | 0.0000 | 1357 | 0.7007 | 1.097 |
| Median | 76.00 | 14.0 | 3.000 | 29.00 | 0.0000 | 1475 | 0.7310 | 1.190 |
| Mean | 76.71 | 14.5 | 2.583 | 27.17 | 0.2946 | 1491 | 0.7302 | 1.194 |
| 3rd Quartile | 82.00 | 16.0 | 4.000 | 30.00 | 0.5000 | 1600 | 0.7560 | 1.293 |
| Max. | 98.00 | 23.0 | 5.000 | 30.00 | 2.0000 | 2004 | 0.8370 | 1.587 |

Table 1: Summary Table

plot(in Appendix Figure 3). From the scree plot(Figure 2b) in Figure 2, we can see that most of the data can be selected by using the first five components. The variance plot(Appendix Figure 3) shows that more than 80% of the data is contained in the first five components. The goal of k-means clustering is to divide n observations into k clusters, where each observation belongs to the cluster that has the closest mean.[4] Next, we use the average Silhouette width to find the optimal number of clusters. A high value indicates good clustering and we get this optimal value at 3 clusters(Fig 2c). As we can see in the **K-means clustering**(Fig 2d) in Figure 2 the three clusters are well separated with clear boundaries indicating distinct groups within the data. **Hierarchical clustering** groups similar objects into a dendrogram. We used the ”complete” linkage method where the distance between two clusters is the distance between any single data point in the two clusters. The dendrogram was divided into 3 clusters(Figure 2) which consist of:- [Age, EDUC, CDR, ASF], [M.F], and [eTIV, SES, MMSE, nWBV]

## 4.3 Feature Selection

Feature selection before logistic regression can help in building more accurate, comprehensible, and efficient models while reducing the probability of over-fitting and instability. We will implement some feature selection methods and try to find the most important features before fitting a logistic regression model to predict the **Group** variable. The group variable is converted into numerical values where 0 represents demented and 1 represents non-demented. We use the randomForest function to generate the classification model and use the varImp method to calculate feature importance.(Appendix figure 4) CDR has the highest importance score followed by MMSE and eTIV. Then we used three feature selection methods and compared the results.

1. **Wrapper Method(Forward Selection)**: Forward selection starts off with no variables in the model, tests each one as it is added, then keeps the ones that are regarded to be the most statistically significant—repeating the procedure until the outcomes are

optimal. For our model after five iterations, it achieves the lowest AIC value of -945.38 and the model with the selected features is:- $Group \leftarrow CDR + EDUC + M.F + eTIV$ (Appendix Table 2)

2. **Wrapper Method(Backward Elimination)**: In this method, we start with the entire model and then remove the unimportant features with the highest p-values till we have the optimal set of significant features. In backward elimination, it removed the features ASF, MMSE, SES, and Age with p-value higher than the significance level. (Appendix Table 3) After four iterations it reaches the lowest AIC value of -945.38 and the model with the selected features is:- $Group \leftarrow M.F + EDUC + CDR + eTIV$ So we see that both the forward and backward selection methods return the same set of optimal features.

3. **Boruta Method**: Boruta works as a wrapper algorithm around random forest. It produces random shadow copies of the features(noise), then compares the feature to the noise to see if it is superior and, therefore, worth preserving. The features with Z-score(standard deviation from the mean) higher than the maximum Z-scores of its shadow copies are selected. (Appendix figure 4) In our model, the Boruta method returns all the features as Confirmed.

## 4.4 Logistic Regression model

The data was split into 70:30 for training and testing purposes respectively. We use the glm method to model the relationship between the response variable Group and the predictor variables M.F, EDUC, CDR, and eTIV selected from forward and backward selection methods. $Group \leftarrow -1.701e-01 + 1.665e+01M.F - 4.129e-01EDUC - 9.595e+01CDR + 1.005e-02eTIV$.The p-values are high for all the predictor variables (Appendix Table 4) which suggests that there is weak evidence against the null hypothesis, indicating that the coefficient is not significantly different from zero. Then we applied 10-fold cross-validation on the training data and got an average mean squared error of 0.0054. A low value indicates a good model fit. However, p-values are not the only metrics and we will use accuracy, precision, and recall to measure the model's predictive performance. For our model, the accuracy was 0.99 which means that the model correctly predicted the outcome for approximately 99% of the instances in the dataset. This is a high accuracy rate and suggests that the model is performing very well in classifying both positive and negative instances. We get precision value 1 which means that all instances predicted as positive by the model are truly positive. There are no false positive predictions, indicating that the model is highly reliable and accurate when it classifies an instance as positive. This is an ideal precision score, indicating that the model does not make any type I errors (false positives). A recall of 0.9818 implies that the model correctly identified approximately 98.18% of the positive instances in the dataset. It suggests that the model has high sensitivity and can capture a large proportion of the actual positive instances. The F1 score is 0.99 which indicates a good balance between precision and recall.

We also tried an alternative model called the LASSO model.[5] It performs both regularization and feature selection by shrinking the coefficients of less relevant predictors toward zero. The data was again split into 70:30 for training and testing purposes respectively and the data was scaled. Then we performed cross-validation and selected the lambda value with minimum cross-validated error. Then we used this optimal lambda value to fit the LASSO model. M.F, SES, MMSE, and ASF have zero coefficient values which mean they have been completely removed from the model by the LASSO regularization. $Group \leftarrow -0.06854653 + 0.7611497Age - 0.2022408EDUC - 8.4511257CDR + 1.0273835eTIV + 1.0210872nWBV$ . A positive coefficient indicates that an increase in the corresponding predictor variable is associated with an increase in the predicted outcome variable, holding all other variables constant. A negative coefficient indicates that an increase in the corresponding predictor variable is associated with a decrease in the predicted outcome variable, holding all other variables constant. For this model, we get an accuracy of 0.99, a precision of 1, and a recall of 0.9821.

## 5 Discussion and Conclusion

We observed that dementia cases are more prevalent in males than females. Most cases are in the age group of 73-80. The majority of the variables do not have a high correlation with each other. Applying PCA helped to reduce the dimensionality of the data and three well-separated clusters were found showing distinct groups within the data. Next, we performed feature selection to find optimal features to fit the model instead of using all nine variables. From both forward and backward selection methods we find a list of optimal variables and fit two types of logistic regression models on this data. Both models show a high accuracy of 0.99. However, the LASSO model can be given preference because it can perform cross-validation and feature selection by itself and present a model fitted to the optimal predictor variables. In conclusion, we observe that using the given characteristics we can fit a strong model to predict with high accuracy the diagnosis of demented/non-demented patients.

## References

[1] What is alzheimer's?

[2] Analytics Vidhya [Preprint]. Tutorial on 5 powerful r packages used for imputing missing values.(2020).

[3] C. (2019) Taylor. What are the first and third quartiles?

[4] Wikipedia contributors (2023). K-means clustering.

[5] Zach (2020). Lasso regression in r (step-by-step).

# A Appendix A

### Variable description

1. **Group**:- Group of the diagnosis (Nondemented, Demented, Converted)
2. **M/F**:- Gender of the patients, Male /Female
3. **Age**:- The age of the patients ranges from 60-98
4. **EDUC**:- Years of education range from 6-23
5. **SES**:- Socioeconomic Status (1-5, 1-low, 5-high)
6. **MMSE**:- Mini-mental state examination. mild Alzheimer's disease: MMSE 21–26, moderate Alzheimer's disease: MMSE 10–20, moderately severe Alzheimer's disease: MMSE 10–14, severe Alzheimer's disease: MMSE less than 10
7. **CDR**:- Clinical dementia rating. No dementia (CDR = 0), questionable dementia (CDR = 0.5), mild cognitive impairment (CDR = 1), moderate cognitive impairment (CDR = 2), and severe cognitive impairment (CDR = 3).
8. **eTIV**:- Estimated total intracranial volume
9. **nWBV**:- Normalize whole brain volume
10. **ASF**:- Atlas scaling factor

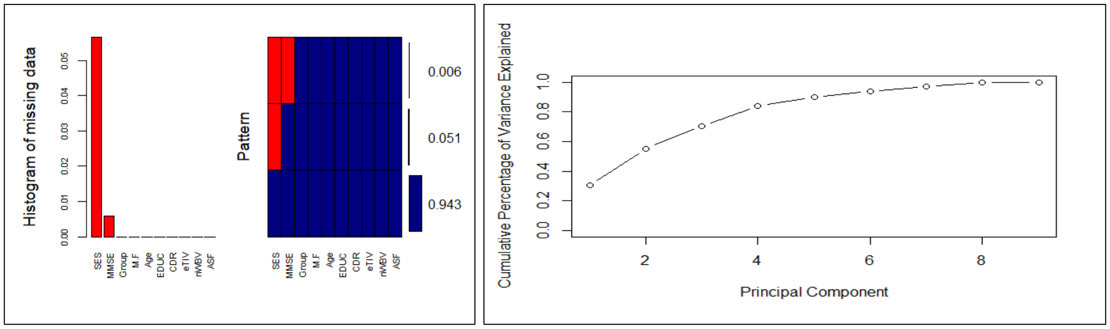

Figure 3: Histogram of missing data and PCA variance plot

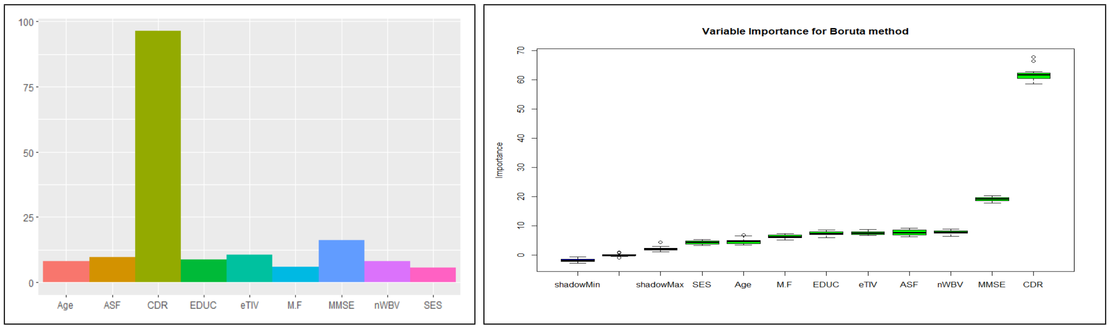

Figure 4: Feature importance and variable importance for Boruta method

| | Estimate | Std. Error | t value | Pr(>—t—) |
|---|---|---|---|---|
| (Intercept) | 4.570e-01 | 1.220e-01 | 3.746 | 0.000212 |
| CDR | -1.041e+00 | 3.593e-02 | -28.970 | <2e-16 |
| M.F | -1.620e-01 | 3.337e-02 | -4.854 | 1.87e-06 |
| eTIV | 2.778e-04 | 9.315e-05 | 2.982 | 0.003077 |
| EDUC | 1.612e-02 | 4.836e-03 | 3.333 | 0.000958 |

Table 2: Coefficients for the final model in wrapper method forward feature selection

| | Estimate | Std. Error | t value | Pr(>—t—) |
|---|---|---|---|---|
| (Intercept) | 0.1223296 | 1.5200723 | 0.080 | 0.9359 |
| M.F | -0.1481023 | 0.0341850 | -4.332 | 1.97e-05 |
| Age | 0.0030611 | 0.0021514 | 1.423 | 0.1557 |
| EDUC | 0.0121252 | 0.0069303 | 1.750 | 0.0811 |
| SES | -0.0178762 | 0.0177088 | -1.009 | 0.3135 |
| MMSE | -0.0024313 | 0.0050934 | -0.477 | 0.6334 |
| CDR | -1.0304856 | 0.0496873 | -20.739 | <2e-16 |
| eTIV | 0.0001542 | 0.0005028 | 0.307 | 0.7592 |
| nWBV | 0.8344815 | 0.4824431 | 1.730 | 0.0846 |
| ASF | -0.1489599 | 0.6331623 | -0.235 | 0.8142 |

Table 3: Coefficients of the full model in wrapper method backward feature selection

| | Estimate | Std. Error | t value | Pr(>—t—) |
|---|---|---|---|---|
| (Intercept) | -1.701e-01 | 9.724e+03 | 0.000 | 1.000 |
| M.F | 1.665e+01 | 7.025e+03 | 0.002 | 0.998 |
| EDUC | -4.129e-01 | 3.558e-01 | -1.160 | 0.246 |
| CDR | -9.595e+01 | 1.945e+04 | -0.005 | 0.996 |
| eTIV | 1.005e-02 | 9.159e-03 | 1.097 | 0.273 |

Table 4: Coefficients for the logistic regression model

**R code:**

```
#install and load required packages
if (!require(dplyr))
  {install.packages("dplyr")}
library(dplyr)

if (!require(ggplot2))
{install.packages("ggplot2")}
library(ggplot2)

if (!require(mice))
{install.packages("mice")}
library(mice)

if (!require(VIM))
{install.packages("VIM")}
library(VIM)

if (!require(corrplot))
{install.packages("corrplot")}
library(corrplot)

if (!require(factoextra))
{install.packages("factoextra")}
library(factoextra)

if (!require(caret))
{install.packages("caret")}
library(caret)
```

```
if (!require(gridExtra))
{install.packages("gridExtra")}
library(gridExtra)

if (!require(ggcorrplot))
{install.packages("ggcorrplot")}
library(ggcorrplot)

#set working directory
setwd("E:/DataScandApp/MA335_DataModeling/project")

#load csv file project data into R
Alzheimer_data = read.csv('project data.csv',header=T)

#Preliminary Analysis
#Data pre-processing

#checking the dimensions of the data set
dim(Alzheimer_data) #373 rows and 10 variables

#remove rows with group=Converted
Alzheimer_data = Alzheimer_data %>% filter(Group!="Converted")
#336 rows and 10 variables

#handle missing data
#check which variables have missing data
aggr_plot <- aggr(Alzheimer_data,col=c('navyblue','red'),numbers=TRUE,
                  sortVars=TRUE,labels=names(Alzheimer_data),cex.axis=.7,gap=3,
                  ylab=c("Histogram of missing data","Pattern"))
##SES has the maximum missing data followed by MMSE.the other variables do not have missing data.
```

```
##put graph and count values in console o/p

#from the mice library missing data is imputed using random forests
imputed_data <- mice(Alzheimer_data,m=5,method="rf")
summary(imputed_data)

imputed_data$imp$SES
imputed_data$imp$MMSE

#update the data frame with the imputed values for SES and MMSE
Alzheimer_data <- complete(imputed_data,1)

#check again if any missing values left
sapply(Alzheimer_data, function(x) sum(is.na(x)))
#so now no more missing values present in the data set

str(Alzheimer_data)

######################################################################################
###########

# Analysis using descriptive statistics

 #Frequency of Alzheimers disease in males(M) and females(F)
  Alzheimer_genderbasis = Alzheimer_data %>% group_by(M.F,Group) %>%
  summarize(count = n())

  img1 = Alzheimer_genderbasis %>%
  ggplot(aes(M.F,count,fill = M.F)) +
  geom_col(show.legend=TRUE,) +
  ggtitle("Fig 1a Frequency of Alzheimers disease in males(M) and females(F)")+
  xlab("Gender") + ylab("Frequency") +
  theme_bw() +
  facet_wrap(~Group)
  img1

  #Frequency of Alzheimers disease according to age
  Alzheimer_agebasis = Alzheimer_data %>%  group_by(Age,Group) %>%
  summarize(count = n())

  img2 = Alzheimer_agebasis %>%
  ggplot(aes(Age,count,fill = Age)) +
  geom_col(show.legend=TRUE) +
  ggtitle("Fig 1b Frequency of Alzheimers disease according to age")+
  xlab("Age of subject") + ylab("Frequency") +
  theme_bw() +
  facet_wrap(~Group)
  img2

  #box-plot for demented and non demented groups in SES vs eTIV
  Alzheimer_eTIV = Alzheimer_data %>%  group_by(SES,Group,eTIV) %>%
  summarize(count = n())

  img3 = ggplot(Alzheimer_eTIV, aes(x = SES, y = eTIV, fill = Group)) +
  geom_boxplot() +
  ggtitle("Fig 1c SocioEconomic status vs Estimated Total Intracranial volume")+
  facet_wrap(~SES)
  img3

  #scatter plot of SES vs nWBV for demented and non-demented groups
```

```r
  img4=ggplot(Alzheimer_data,aes(x=SES,y=nWBV,color=Group)) +
  geom_jitter() +
  ggtitle("Fig 1d Scatterplot of SES vs nWBV for demented and non-demented groups")
  img4

  #combine the plots using grid.arrange function
  grid.arrange(img1, img2, nrow = 2)
  grid.arrange(img3, img4, nrow = 2)

#convert M/F into numeric values
#female is 1 and male is 2 . 180 females and 137 males
Alzheimer_data$M.F = as.numeric(as.factor(Alzheimer_data$M.F))



#numerical analysis of data
summary(Alzheimer_data[-c(1,2)])
#correlation matrix
cormat <- round(x = cor(Alzheimer_data[-1],use="pairwise.complete.obs"), digits = 2)
ggcorrplot(cormat,lab=TRUE) + ggtitle("Fig 2a Correlation matrix")

##*****************************************************************************************
###########################################################################################


#PCA
#By applying PCA before k-means clustering, we can reduce the dimensionality of the data
#and eliminate redundant information, which can help improve the performance of the clustering algorithm.

#dropped the GROUP variable from the dataset
```

```
pca1<- prcomp(Alzheimer_data[-1] , scale =TRUE)
summary(pca1)
# the loading's of the pca components
print(pca1,digit=2)
pca1.loadings<- pca1$rotation
pca1.scores <- pca1$x

#The percentage of variance explained and the cumulative variance explained #pca5
per.var <- pca1$sdev^2
prop.var.expl <- per.var/sum(per.var); prop.var.expl
cumsum(prop.var.expl)
plot(pca1,type="l")
plot(prop.var.expl, xlab=" Principal Components ", ylab=" Percentage of Variance Explained ",
     ylim=c(0,1) ,type='b')

plot(cumsum(prop.var.expl), xlab=" Principal Component ",
     ylab =" Cumulative Percentage of Variance Explained ",
     ylim=c(0,1), type='b')
##more than 80 % of variance explained by 5 PCA components



pca1$rotation=-pca1$rotation
pca1$x=-pca1$x
biplot(pca1,scale=0)

 screeplot = qplot(c(1:9), per.var) +
  geom_line()+
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("Fig 2b Scree Plot for PCA components")
 screeplot


axes <- predict(pca1, newdata = Alzheimer_data)
head(axes, 4)



##******************************************************************************************************


##***********************************************************************************************
## now we try cluster after doing pca to check the results
##we take the first 5 pca components and perform kmean cluster on them
pca_data = pca1.loadings[,-c(6:10)]

#trying kmeans clusters for various values of k
kmeans2 <- kmeans(pca_data, centers = 2, nstart = 20)
kmeans3 <- kmeans(pca_data, centers = 3, nstart = 20)
kmeans4 <- kmeans(pca_data, centers = 4, nstart = 20)
kmeans5 <- kmeans(pca_data, centers = 5, nstart = 20)

f1 <- fviz_cluster(kmeans2, geom = "point", data = pca_data) + ggtitle("k = 2")
f2 <- fviz_cluster(kmeans3, geom = "point", data = pca_data) +
ggtitle("Fig 2d k-means clustering for 3 centers")
f3 <- fviz_cluster(kmeans4, geom = "point", data = pca_data) + ggtitle("k = 4")
f4 <- fviz_cluster(kmeans5, geom = "point", data = pca_data) + ggtitle("k = 5")
grid.arrange(f1, f2, f3,f4, nrow = 2)

#select optimal number of clusters
f5 = fviz_nbclust(pca1$x, FUNcluster=kmeans, k.max = 8) +
  ggtitle("Fig 2c Optimal number of clusters")
```

```
#we select cluster kmeans3 with 3 centers


#arrange images in grid
grid.arrange(screeplot,f5,f2,nrow=2)


##hierarchical clustering


#Start by calculating the distance matrix
d <- dist(pca_data, method = "euclidean")


#Apply hierarchical clustering for linkage method complete
fit.complete <- hclust(d, method="complete")


# print the dendrogram
plot(fit.complete)
groups.fit.complete <- cutree(fit.complete, k=3)
# draw dendrogram with red borders around the 3 clusters
rect.hclust(fit.complete, k=3, border="red")


#Checking how many observations are in each cluster
table(groups.fit.complete)


##****************************************************************************************
**


#Variable selection/feature selection


#Convert Group variable to numeric values where 0 represents Demented
#and 1 represents non-Demented groups.
df1 = Alzheimer_data
```

```
df1$Group = ifelse(df1$Group == "Demented", 0, 1)


## check feature importance
library(randomForest)
model <- randomForest(Group ~ M.F + Age + EDUC + SES + MMSE + CDR + eTIV + nWBV + ASF,
                      data = df1, importance=TRUE)
#Conditional=True, adjusts for correlations between predictors.
i_scores <- varImp(model, conditional=TRUE)


#Gathering row names in 'var'  and converting it to the factor
#to provide 'fill' parameter for the bar chart.
i_scores <- i_scores %>% tibble::rownames_to_column("var")
i_scores$var<- i_scores$var %>% as.factor()


i_scores = i_scores %>% arrange(Overall)
i_bar <- ggplot(data = i_scores) +
  geom_bar(
    stat = "identity",#it leaves the data without count and bin
    mapping = aes(x = var, y=Overall,fill=var),
    show.legend = FALSE,
    width = 1
  ) +
  labs(x = NULL, y = NULL)
i_bar


##WRAPPER METHOD(FORWARD),BACKWARD)


model1<-lm(Group~1,data=df1)
step1<-step(model1,scope=~M.F + Age + EDUC + SES + MMSE + CDR + eTIV + nWBV + ASF,
            method='forward')
```

```
summary(step1)

##WRAPPER METHOD(BACKWARD)
y<-df1[,1]
X<-df1[,2:10]
model2<-lm(y~.,data=X)
summary(model2)
step2<-step(model2,method="backward")
summary(step2)



##BORUTA
library(Boruta)
boruta1 <- Boruta(y ~., data=X, doTrace=1)
decision<-boruta1$finalDecision
signif <- decision[boruta1$finalDecision %in% c("Confirmed")]
print(signif)
plot(boruta1, xlab="", main="Variable Importance for Boruta method")
attStats(boruta1)

############################################################################
##Logistic Regression model.

##glm

# Create Training and Test data
#split into 70:30
df1$Group = as.factor(df1$Group)

# Split the data into training and test sets
```

```
trainingRowIndex <- sample(1:nrow(df1), 0.7*nrow(df1))  # row indices for 70% training data
trainingData <- df1[trainingRowIndex, ]  # model training data
testData  <- df1[-trainingRowIndex, ] #test data
Group.test <- df1$Group[-trainingRowIndex]

#M.F + EDUC + CDR + eTIV from forward and backward selection
library(boot)
glm.fit1 <- glm(Group~ M.F + EDUC + CDR + eTIV,data=trainingData ,family =binomial(link="logit"))
summary(glm.fit1)
#k fold cross validation
cv1<-cv.glm(trainingData,glm.fit1,K=10)
cv1$delta
#preidction
predict <- predict(glm.fit1, testData, type = 'response')

# confusion matrix
table_mat <- table(testData$Group, predict > 0.5)
table_mat

#calculate accuracy
accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
accuracy_Test

#calculate precision
precision <- function(matrix) {
  # True positive
  tp <- matrix[2, 2]
  # false positive
  fp <- matrix[1, 2]
  return (tp / (tp + fp))
}

#calculate recall
recall <- function(matrix) {
  # true positive
  tp <- matrix[2, 2]# false positive
  fn <- matrix[2, 1]
  return (tp / (tp + fn))
}

prec <- precision(table_mat)
prec
rec <- recall(table_mat)
rec
#calculate f1 score
f1 <- 2 * ((prec * rec) / (prec + rec))
f1

###*******************************************************************

#LASSO model

df1$Group = as.factor(df1$Group)
y <- df1[,1]
X <- df1[,2:10] #all variables

# Split the data into training and test sets
set.seed(42)  # Set a seed for reproducibility
train_indices <- sample(1:nrow(X), nrow(X) * 0.7)  # 70% for training, 30% for testing
X_train <- X[train_indices, ]
```

```
y_train <- y[train_indices]
X_test <- X[-train_indices, ]
y_test <- y[-train_indices]

# Standardize the predictor variables
X_train_scaled <- scale(X_train)
X_test_scaled <- scale(X_test)

# Fit a LASSO logistic regression model
library(glmnet)
#By performing cross-validation, we can estimate the model's performance on unseen data
#and select the lambda value that provides the best trade-off between model complexity
#and fit to the data.
lasso_model <- cv.glmnet(x = X_train_scaled, y = y_train, family = "binomial", alpha = 1)

# Find the optimal lambda (penalty parameter) based on cross-validation
lasso_model$lambda       # Lambda values
lasso_model$cvm          # Cross-validated error rates
best_lambda <- lasso_model$lambda.min #the lambda value with the minimum cross-validated error

# Fit the LASSO model with the optimal lambda
lasso_model_final <- glmnet(x = X_train_scaled, y = y_train, family = "binomial", alpha = 1, lambda = best_lambda)

# Predict on the test set
X_test_scaled <- as.matrix(X_test_scaled)  # Convert to matrix if needed
y_pred <- predict(lasso_model_final, newx = X_test_scaled, type = "response")
y_pred <- ifelse(y_pred > 0.5, 1, 0)  # Convert probabilities to binary predictions

# Create the confusion matrix
```

```
confusion_mat <- confusionMatrix(as.factor(y_pred), y_test)

# Print the confusion matrix
print(confusion_mat)


# Evaluate the model
accuracy <- sum(y_pred == y_test) / length(y_test)
precision <- sum(y_pred[y_test == 1] == y_test[y_test == 1]) / sum(y_pred == 1)
recall <- sum(y_pred[y_test == 1] == y_test[y_test == 1]) / sum(y_test == 1)
accuracy;precision;recall
# Access the coefficients and intercept
coefficients <- coef(lasso_model_final)
intercept <- coefficients[1]
coefficients <- coefficients[-1]
intercept
coefficients


##########################################################################################
```