Investigation of multi-variate datasets

Glass Identification

Description

This dataset was developed initially to determine whether a glass type was float or non-float. Glass identification can help in crime investigation. With the identification of the type of glass which is left as evidence in the crime scene, the investigation can be boosted. In this dataset, different instances of such glasses are observed or considered as data samples. There are 214 observation samples in this data. For each sample, nine features of the sample are recorded. Those features are:

- Refractive Index (RI)
- Sodium (Na)
- Magnesium (Mg)
- Alumunium (Al)
- Silicon (Si)
- Potassium (K)
- Calcium (Ca)
- Barium (Ba)
- Iron (Fe)

Along with such features, types of the glass referring to an individual sample are provided in a class column. The classes are maintained as follows:

1. Class 1 for building windows float processed
2. Class 2 for building windows non float processed
3. Class 3 for vehicle windows float processed
4. Class 4 for vehicle windows non float processed
5. Class 5 for containers
6. Class 6 for tableware
7. Class 7 for headlamps

This is a classification task in which new samples containing such features will be tested. The type of glass of such new samples should be identified by the algorithm developed.

Data Preparation

Initially the data was loaded using the read.table method in R. The data was downloaded from the UCI machine learning repo website. The columns of the data loaded are provided with their respective name. On observing the first 10 samples of the data, the following result was obtained.

After the loading of the dataset, the column names were changed according to the description of the dataset given in UCI ML repo website. After changing the column names following first 10 observations of the data was obtained.

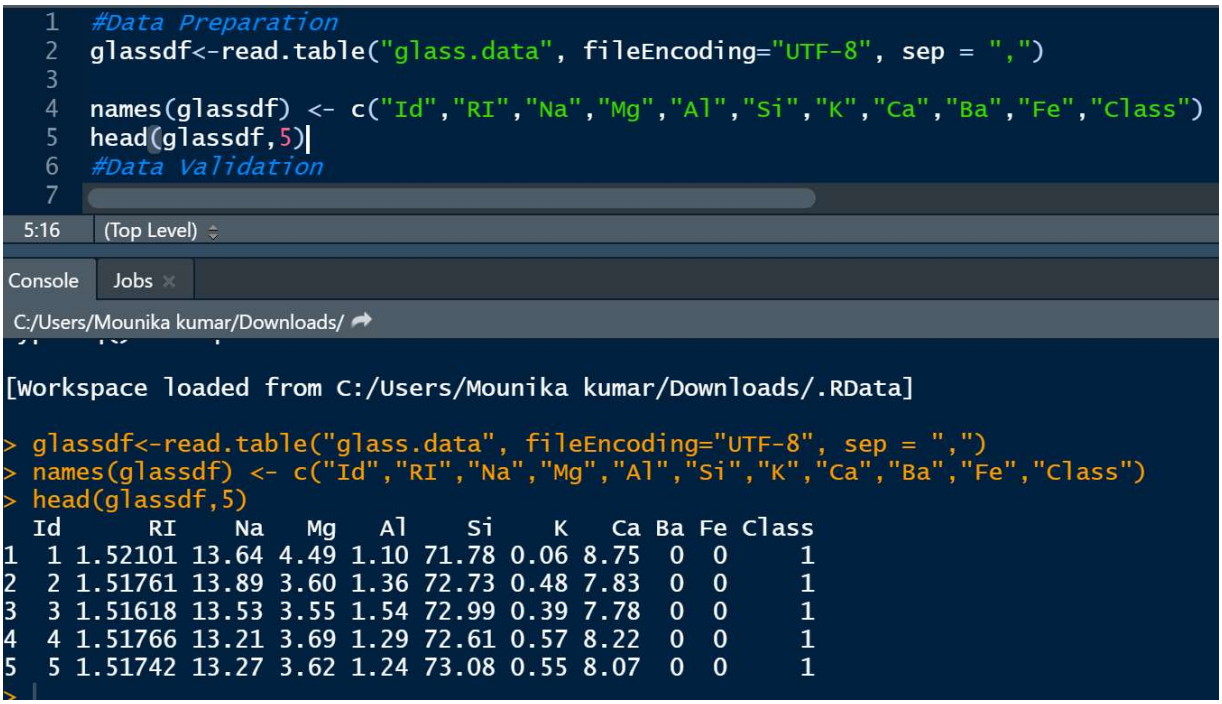


Fig: Screenshot showing the first 10 observations in glass identification dataset and dimension of the dataset

The dimension of the dataset suggests that there are 214 samples of such glass data and 10 features including the glass Id and a class representing the type of glass.

Data Validation

In this step, we check whether the number of columns and rows is equal to the number mentioned in the metadata or not and get the summary to compare with the one given in the glass.tag file.

Initially, we check whether there are null values in the dataset or not.

```
 9  sum(is.na(glassdf)) # checking if there is na value
10  stopifnot(ncol(glassdf)==11)
11  stopifnot(nrow(glassdf)==214)
12  summary(glassdf[1:9])
13
```

11:30 (Top Level)

Console Jobs

C:/Users/Mounika kumar/Downloads/

```
> summary(glassdf[1:9])
       Id              RI              Na              Mg              Al
 Min.   :  1.00   Min.   :1.511   Min.   :10.73   Min.   :0.000   Min.   :0.290
 1st Qu.: 54.25   1st Qu.:1.517   1st Qu.:12.91   1st Qu.:2.115   1st Qu.:1.190
 Median :107.50   Median :1.518   Median :13.30   Median :3.480   Median :1.360
 Mean   :107.50   Mean   :1.518   Mean   :13.41   Mean   :2.685   Mean   :1.445
 3rd Qu.:160.75   3rd Qu.:1.519   3rd Qu.:13.82   3rd Qu.:3.600   3rd Qu.:1.630
 Max.   :214.00   Max.   :1.534   Max.   :17.38   Max.   :4.490   Max.   :3.500
       Si              K               Ca              Ba
 Min.   :69.81   Min.   :0.0000   Min.   : 5.430   Min.   :0.000
 1st Qu.:72.28   1st Qu.:0.1225   1st Qu.: 8.240   1st Qu.:0.000
 Median :72.79   Median :0.5550   Median : 8.600   Median :0.000
 Mean   :72.65   Mean   :0.4971   Mean   : 8.957   Mean   :0.175
 3rd Qu.:73.09   3rd Qu.:0.6100   3rd Qu.: 9.172   3rd Qu.:0.000
 Max.   :75.41   Max.   :6.2100   Max.   :16.190   Max.   :3.150
```

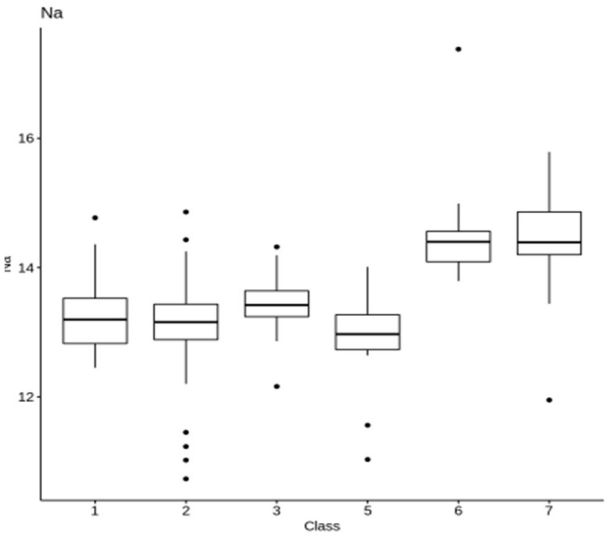
Fig: Screenshot representing data validation of the glass identification dataset.

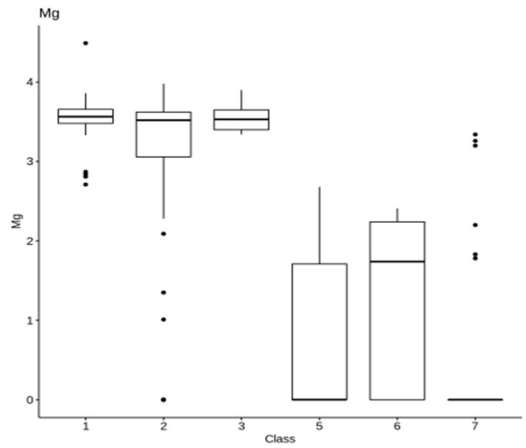
Data Analysis

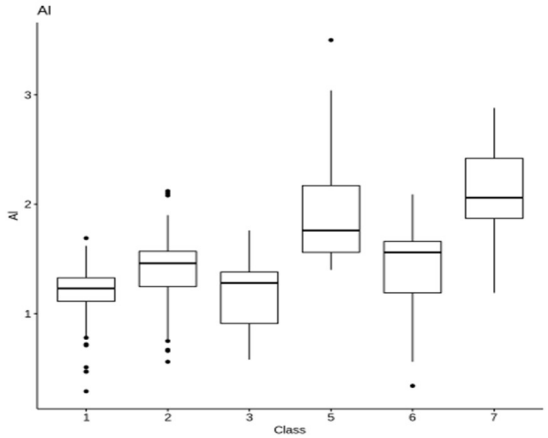
Box plot

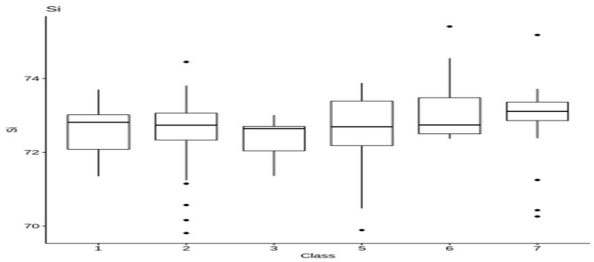
Box Plot shows some outliers present in the data that needs further cleaning. They are useful because they show the average score of the dataset. They are specifically used to get a visual representation of the dispersion of a data set, signs of skewness, mean values and so on. “ggpubr” and “ggplot2” package of R was utilized to generate box plots for the dataset. Following box plot was generated for the feature columns like “Na”, “Mg”, “Al”, “Si”, “K”, “Ca” and “Ba”.

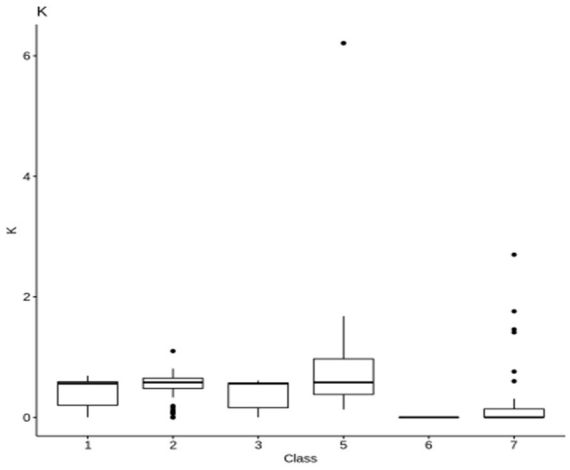
```
19  ggboxplot(glassdf, x = "Class", y = c("Na", "Mg","Al","Si","K","Ca","Ba"),
20            merge = FALSE, palette = "jco")
```

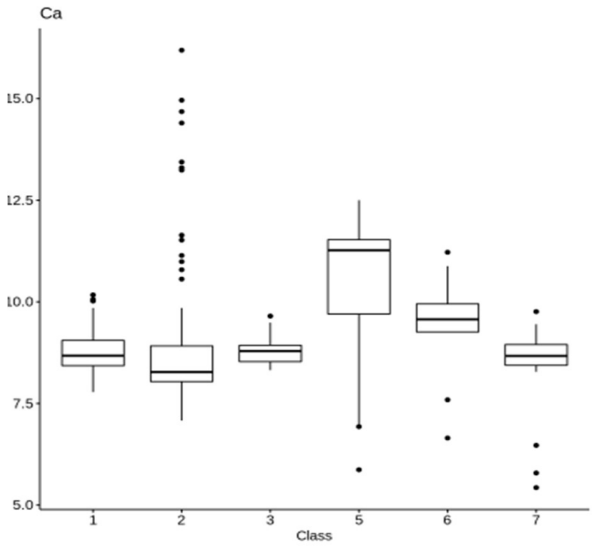

Na
16
14
12
1
2
3
5
6
7
Class


Mg
4
3
2
1
0
Mg
1
2
3
5
6
7
Class


Al
3
2
1
Al
1
2
3
5
6
7
Class


Si
74
72
70
Si
1
2
3
5
6
7
Class


K
6
4
2
0
K
1
2
3
5
6
7
Class


Ca
15.0
12.5
10.0
7.5
5.0
1
2
3
5
6
7
Class

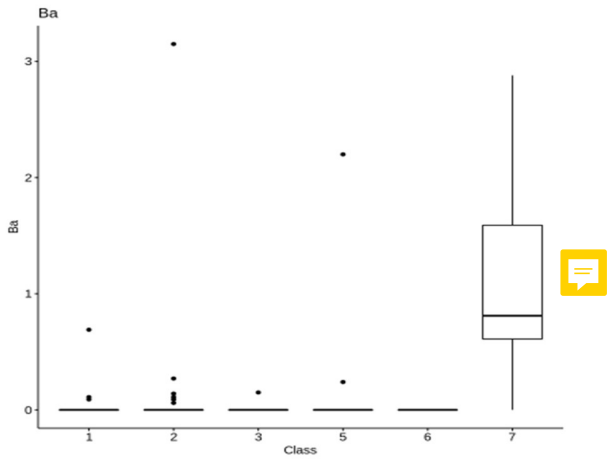


Fig: Box plot of the dataset attributes with respect to class

Correlation Plot

Correlation plot was performed on the dataset to evaluate the linear relationship measure of the features and class of the dataset. Following result was obtained:

```
21  cor(glassdf)
22  install.packages("psych")
23  library(psych)
24  pairs.panels(glassdf[2:10], stars = TRUE)
```

```
Id      0.49011329  0.06123208  0.003148751  0.0907999431
RI     -0.40732603 -0.54205220 -0.289832711  0.8104026963
Na      0.15679367 -0.06980881 -0.266086504 -0.2754424856
Mg     -0.48179851 -0.16592672  0.005395667 -0.4437500264
Al      1.00000000 -0.00552372  0.325958446 -0.2595920102
Si     -0.00552372  1.00000000 -0.193330854 -0.2087321537
K       0.32595845 -0.19333085  1.000000000 -0.3178361547
Ca     -0.25959201 -0.20873215 -0.317836155  1.0000000000
Ba      0.47940390 -0.10215131 -0.042618059 -0.1128409671
Fe     -0.07440215 -0.09420073 -0.007719049  0.1249682190
Class   0.59882921  0.15156526 -0.010054464  0.0009522246
                Ba           Fe         Class
Id      0.4510013746 -0.072794273  0.8773565792
RI     -0.0003860189  0.143009609 -0.1642372146
Na      0.3266028795 -0.241346411  0.5028980423
Mg     -0.4922621178  0.083059529 -0.7449928875
Al      0.4794039017 -0.074402151  0.5988292084
Si     -0.1021513105 -0.094200731  0.1515652579
K      -0.0426180594 -0.007719049 -0.0100544638
Ca     -0.1128409671  0.124968219  0.0009522246
Ba      1.0000000000 -0.058691755  0.5751614590
Fe     -0.0586917554  1.000000000 -0.1882775640
Class   0.5751614590 -0.188277564  1.0000000000
```

Fig: Correlation plot of the features and class of the dataset

The correlation plot suggests that two best oxides that best predicts the refractive index of the glass are:

1. Calcium (Ca)
2. Iron (Fe)

Similarly, we can draw conclusion from the correlation plot that the two best oxides that best predicts the class of the glass are:

1. Aluminium (Al)
2. Barium (Ba)

Scatter plot

The function pair.panels is the “psych” R package is used to generate a scatter plot of the matrices in which we have bivariate scatter plot below the diagonal, histograms of the data of the column in the diagonal and Pearson correlation above the diagonal. Following scatter plot was generated for the feature columns of the dataset.

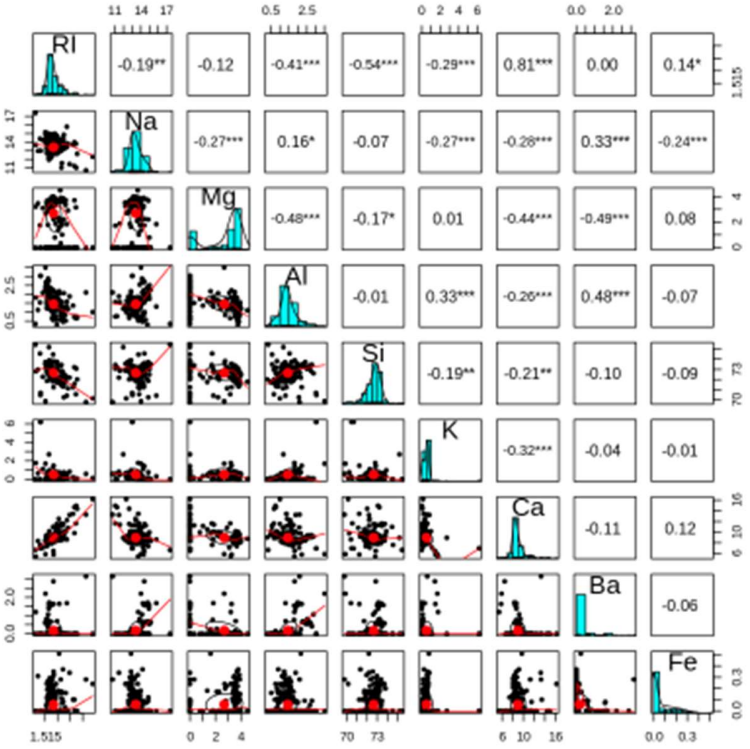
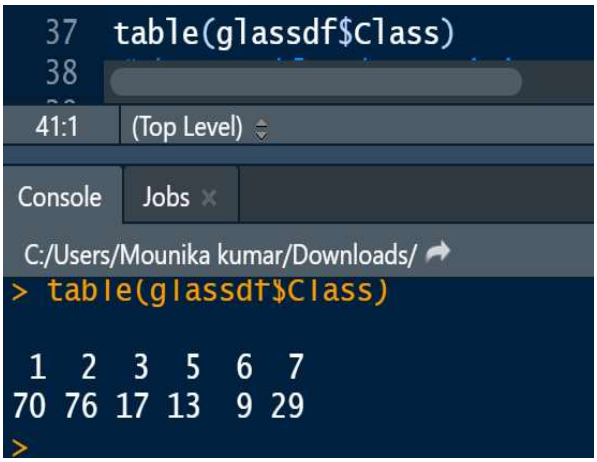


Fig: Scatter plot of the features of glass dataset

MANOVA test

Multivariate variance analysis (MANOVA) of many dependent variables is essentially an ANOVA. In other words, ANOVA measures the mean difference between two or more groups, while MANOVA tests the mean difference between two or more vectors. It can be evaluated

concurrently using a multivariate variance analysis where there are several response variables (MANOVA).
MANOVA assumptions
- The answer vector should usually be assigned to
- Homogeneity of variances across the predictor spectrum.
- Linearity between all pairs of variables that are dependent
- Adequate sample size
- Lack of an outlier, univariate or multivariate

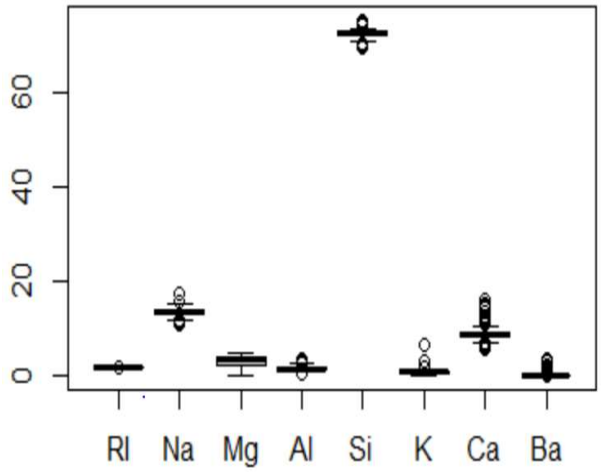


Above table shows a minimum of 9 data which is greater than the number of classes. Thus, the first assumption is verified. We make the use of libraries such as “tidyverse”, “rstatix”, “car”, “broom” for this task. Initially, we should remove any outliers present in the data. To do this we perform,

```
> glassdf %>%
+   group_by(Class) %>%
+   identify_outliers(Fe)
# A tibble: 12 x 13
   Class    Id    RI    Na    Mg    Al    Si     K    Ca    Ba    Fe is.outlier is.extreme
   <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <lgl>      <lgl>
 1     1    45  1.52  12.7  3.43  1.19  73.0  0.62  8.76  0     0.3  TRUE       FALSE
 2     1    57  1.51  13.0  3.47  1.12  73.0  0.62  8.35  0     0.31 TRUE       FALSE
 3     3   162  1.52  13.6  3.54  0.75  72.6  0.16  8.89  0.15  0.24 TRUE       FALSE
 4     3   163  1.52  14.2  3.78  0.91  71.4  0.23  9.14  0     0.37 TRUE       TRUE
 5     5   175  1.52  12.8  1.61  2.17  72.2  0.76  9.7   0.24  0.51 TRUE       TRUE
 6     5   176  1.52  13.0  0.33  1.51  73.4  0.13 11.3   0     0.28 TRUE       TRUE
 7     7   192  1.52  14.8  0     2.38  73.3  0     8.76  0.64  0.09 TRUE       TRUE
 8     7   193  1.52  14.2  0     2.79  73.5  0.04  9.04  0.4   0.09 TRUE       TRUE
 9     7   194  1.52  14.8  0     2     73.0  0     8.53  1.59  0.08 TRUE       TRUE
10     7   195  1.52  14.6  0     1.98  73.3  0     8.52  1.57  0.07 TRUE       TRUE
11     7   196  1.52  14.1  0     2.68  73.4  0.08  9.07  0.61  0.05 TRUE       TRUE
12     7   197  1.52  13.9  0     2.54  73.2  0.14  9.41  0.81  0.01 TRUE       TRUE
```

Fig: Removing outliers in the dataset

```
> boxplot(glassdf[2:9])$out
  [1]  1.52667  1.52320  1.51215  1.52725  1.52410  1.52475  1.53125
  [8]  1.53393  1.52664  1.52739  1.52777  1.52614  1.52369  1.51115
 [15]  1.51131  1.52315  1.52365 11.45000 10.73000 11.23000 11.02000
 [22] 11.03000 17.38000 15.79000  0.29000  0.47000  0.47000  0.51000
 [29]  3.50000  3.04000  3.02000  0.34000  2.38000  2.79000  2.68000
 [36]  2.54000  2.34000  2.66000  2.51000  2.42000  2.74000  2.88000
 [43] 70.57000 69.81000 70.16000 74.45000 69.89000 70.48000 70.70000
 [50] 74.55000 75.41000 70.26000 70.43000 75.18000  1.68000  6.21000
 [57]  6.21000  1.76000  1.46000  2.70000  1.41000 11.64000 10.79000
 [64] 13.24000 13.30000 16.19000 11.52000 10.99000 14.68000 14.96000
 [71] 14.40000 11.14000 13.44000  5.87000 11.41000 11.62000 11.53000
 [78] 11.32000 12.24000 12.50000 11.27000 10.88000 11.22000  6.65000
 [85]  5.43000  5.79000  6.47000  0.09000  0.11000  0.69000  0.14000
 [92]  0.11000  3.15000  0.27000  0.09000  0.06000  0.15000  2.20000
 [99]  0.24000  1.19000  1.63000  1.68000  0.76000  0.64000  0.40000
[106]  1.59000  1.57000  0.61000  0.81000  0.66000  0.64000  0.53000
[113]  0.63000  0.56000  1.71000  0.67000  1.55000  1.38000  2.88000
[120]  0.54000  1.06000  1.59000  1.64000  1.57000  1.67000
>
```



Finally, we obtain the clean dataset as shown below:

```
> cleanedDf
   Id      RI    Na   Mg   Al    Si    K    Ca   Ba   Fe Class
1   1 1.52101 13.64 4.49 1.10 71.78 0.06  8.75 0.00 0.00     1
2   2 1.51761 13.89 3.60 1.36 72.73 0.48  7.83 0.00 0.00     1
3   3 1.51618 13.53 3.55 1.54 72.99 0.39  7.78 0.00 0.00     1
4   4 1.51766 13.21 3.69 1.29 72.61 0.57  8.22 0.00 0.00     1
5   5 1.51742 13.27 3.62 1.24 73.08 0.55  8.07 0.00 0.00     1
6   6 1.51596 12.79 3.61 1.62 72.97 0.64  8.07 0.00 0.26     1
7   7 1.51743 13.30 3.60 1.14 73.09 0.58  8.17 0.00 0.00     1
8   8 1.51756 13.15 3.61 1.05 73.24 0.57  8.24 0.00 0.00     1
9   9 1.51918 14.04 3.58 1.37 72.08 0.56  8.30 0.00 0.00     1
10 10 1.51755 13.00 3.60 1.36 72.99 0.57  8.40 0.00 0.11     1
11 11 1.51571 12.72 3.46 1.56 73.20 0.67  8.09 0.00 0.24     1
12 12 1.51763 12.80 3.66 1.27 73.01 0.60  8.56 0.00 0.00     1
13 13 1.51589 12.88 3.43 1.40 73.28 0.69  8.05 0.00 0.24     1
14 14 1.51748 12.86 3.56 1.27 73.21 0.54  8.38 0.00 0.17     1
15 15 1.51763 12.61 3.59 1.31 73.29 0.58  8.50 0.00 0.00     1
16 16 1.51761 12.81 3.54 1.23 73.24 0.58  8.39 0.00 0.00     1
17 17 1.51784 12.68 3.67 1.16 73.11 0.61  8.70 0.00 0.00     1
18 18 1.52196 14.36 3.85 0.89 71.36 0.15  9.15 0.00 0.00     1
19 19 1.51911 13.90 3.73 1.18 72.12 0.06  8.89 0.00 0.00     1
```

Finally, we perform MANOVA test result by executing following code:

```
> # MANOVA test
> e_manova <- manova(cbind(Na, Mg) ~ as.factor(Class), data = glassdf)
> summary(e_manova)
                 Df  Pillai approx F num Df den Df    Pr(>F)
as.factor(Class)  5 0.90752   34.557     10    416 < 2.2e-16 ***
Residuals       208
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> summary.aov(e_manova)
 Response Na :
                 Df Sum Sq Mean Sq F value    Pr(>F)
as.factor(Class)  5 57.805  11.561  28.548 < 2.2e-16 ***
Residuals       208 84.233   0.405
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response Mg :
                 Df Sum Sq Mean Sq F value    Pr(>F)
as.factor(Class)  5 271.10  54.219  65.544 < 2.2e-16 ***
Residuals       208 172.06   0.827
```

Fig” Results of MANOVA test

R Code:

#Data Preparation
glassdf<-read.table("glass.data", fileEncoding="UTF-8", sep = ",")

names(glassdf) <- c("Id","RI","Na","Mg","Al","Si","K","Ca","Ba","Fe","Class")

```
head(glassdf,5)
#Data Validation
sum(is.na(glassdf)) # checking if there is na value
stopifnot(ncol(glassdf)==11)
stopifnot(nrow(glassdf)==214)
summary(glassdf[1:9])
#above attributes matches the information given in the meta data so it is a valid dataset
install.packages("GGally")
install.packages("ggpubr")
#box plot to visualize the data and see if outliers are present.
library(GGally)
library(ggpubr)
ggboxplot(glassdf, x = "Class", y = c("Na", "Mg","Al","Si","K","Ca","Ba"),
          merge = FALSE, palette = "jco")
cor(glassdf)
install.packages("psych")
library(psych)
pairs.panels(glassdf[2:10], stars = TRUE)
#box plot shows some outliers present in the data that needs further cleaning and pairs pannel
#performing MNOVA test
# Absense of univariate or multivariate outliers.
# Multivariate normality.

table(glassdf$Class)
#Above table shows minimum of 9 data number of class
install.packages("tidyverse")
install.packages("rstatix")
install.packages("car")
install.packages("broom")
library(tidyverse)
library(rstatix)
library(car)
library(broom)
#removing outliers if any
glassdf %>%
  group_by(Class) %>%
  identify_outliers(Fe)
boxplot(glassdf[2:9])$out

cleanedDf<-glassdf
cleanedDf<- glassdf[-which(glassdf$Na %in% outliersNa),]
cleanedDf
# MANOVA test
e_manova <- manova(cbind(Na, Mg) ~ as.factor(Class), data = glassdf)
```

summary(e_manova)
summary.aov(e_manova)

EL-Nino

Description

The data collection comprises oceanographic and meteorological observations of the surface taken from a series of buoys distributed in the equatorial Pacific. In order to grasp and forecast El Nino/Southern Oscillation (ENSO) cycles, the data is supposed to help.The dataset has following feature columns:

- Date
- Latitude
- Longitude
- Zonal winds
- Meridional winds
- Relative Humidity
- Air Temperature
- Sea Surface Temperature
- Subsurface Temperature

Initially, the data was downloaded from the uci ml repo website and loaded using R. Apart from data analysis of other dataset using Google Colab Notebooks, data analysis for this dataset was done using R studio.The .dat extension file downloaded was read using read.table function in R.

```
9  dataset<-read.table("tao-all2.dat")
```

The first 10 instances of the data are shown below:

```
> head(dataset,10)
   obs year month day   date latitude longitude zon.winds mer.winds humidity air_temp s.s.temp
1    1   80     3   7 800307    -0.02   -109.46      -6.8       0.7        .    26.14    26.24
2    2   80     3   8 800308    -0.02   -109.46      -4.9       1.1        .    25.66    25.97
3    3   80     3   9 800309    -0.02   -109.46      -4.5       2.2        .    25.69    25.28
4    4   80     3  10 800310    -0.02   -109.46      -3.8       1.9        .    25.57    24.31
5    5   80     3  11 800311    -0.02   -109.46      -4.2       1.5        .     25.3    23.19
6    6   80     3  12 800312    -0.02   -109.46      -4.4       0.3        .    24.72    23.64
7    7   80     3  13 800313    -0.02   -109.46      -3.2       0.1        .    24.66    24.34
8    8   80     3  14 800314    -0.02   -109.46      -3.1       0.6        .    25.17    24.14
9    9   80     3  15 800315    -0.02   -109.46        -3         1        .    25.59    24.24
10  10   80     3  16 800316    -0.02   -109.46      -1.2         1        .    26.71    25.94
```

Fig: Screenshot showing first 10 instances of El Nino dataset

The columns of the dataset were renamed to appropriate names since column names were not appropriate in the dataset.

```
names(dataset) <- c("obs","year","month","day","date","latitude",
                    "longitude","zon.winds","mer.winds","humidity","air_temp",
                    "s.s.temp")
```

After renaming the columns of the dataset, the head of the dataset is given below with 10 as parameter.

```
> head(dataset,10)
   obs year month day   date latitude longitude zon.winds mer.winds humidity air_temp s.s.temp
1    1   80     3   7 800307    -0.02   -109.46      -6.8       0.7        .    26.14    26.24
2    2   80     3   8 800308    -0.02   -109.46      -4.9       1.1        .    25.66    25.97
3    3   80     3   9 800309    -0.02   -109.46      -4.5       2.2        .    25.69    25.28
4    4   80     3  10 800310    -0.02   -109.46      -3.8       1.9        .    25.57    24.31
5    5   80     3  11 800311    -0.02   -109.46      -4.2       1.5        .     25.3    23.19
6    6   80     3  12 800312    -0.02   -109.46      -4.4       0.3        .    24.72    23.64
7    7   80     3  13 800313    -0.02   -109.46      -3.2       0.1        .    24.66    24.34
8    8   80     3  14 800314    -0.02   -109.46      -3.1       0.6        .    25.17    24.14
9    9   80     3  15 800315    -0.02   -109.46        -3         1        .    25.59    24.24
10  10   80     3  16 800316    -0.02   -109.46      -1.2         1        .    26.71    25.94
```

Fig: Screenshot showing first 10 instances of El Nino dataset

Data Validation

Data Validation was done by checking the number of rows and columns present in the dataset and comparing with the information of the dataset present in the UCI ml repo website. Data validation was done using stopifnot function in R as shown below:

```
stopifnot(ncol(dataset)==12)
stopifnot(nrow(dataset)==178080)
```

Data Cleaning

Initially, we have to check whether the data contains any null values or not to lead to whether data cleaning is required or not.

```
#checking if data contains na values to check if cleaning is required
row.has.na <- apply(dataset, 1, function(x){any(is.na(x))})
sum(row.has.na)
```

The result obtained was 0 indicating there are no null values in the dataset. However, there are several missing values in the dataset. Thus, we apply further data cleaning steps in the dataset as shown below:

```
str(dataset)
dataset<-dataset [ dataset$humidity != ".", ]
dataset<-dataset [ dataset$mer.winds != ".", ]
dataset<-dataset [ dataset$zon.winds != ".", ]
dataset<-dataset [ dataset$air_temp != ".", ]
dataset<-dataset [ dataset$s.s.temp != ".", ]
head(dataset)
```

After the data cleaning steps for the missing values, we convert non numeric columns into numeric using following apply function.

```
#converting non numeric columns to numeric now
dataset[, c(8:12)] <- sapply(dataset[, c(8:12)], as.numeric)
```

Data Analysis

After the data cleaning process, we perform analysis on the data. Initially, the summary of the dataset is generated using the summary function in R. We obtain the following results as a summary of the dataset.

```
> summary(dataset)
      obs              year          month             day              date          latitude
 Min.   :     1   Min.   :80.0   Min.   : 1.000   Min.   : 1.00   Min.   :800307   Min.   :-8.8100
 1st Qu.: 44521   1st Qu.:92.0   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.:920116   1st Qu.:-2.0100
 Median : 89041   Median :94.0   Median : 6.000   Median :16.00   Median :940601   Median : 0.0100
 Mean   : 89041   Mean   :93.3   Mean   : 6.505   Mean   :15.72   Mean   :933690   Mean   : 0.4736
 3rd Qu.:133560   3rd Qu.:96.0   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:960617   3rd Qu.: 4.9800
 Max.   :178080   Max.   :98.0   Max.   :12.000   Max.   :31.00   Max.   :980623   Max.   : 9.0500
   longitude       zon.winds          mer.winds          humidity          air_temp
 Min.   :-180.00   Length:178080      Length:178080      Length:178080      Length:178080
 1st Qu.:  -4.95   Class :character   Class :character   Class :character   Class :character
 Median :   1.26   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Mean   : -54.03
 3rd Qu.: 147.01
 Max.   : 171.08
   s.s.temp
 Length:178080
 Class :character
 Mode  :character
```

Fig: Screenshot showing the summary generated of the dataset

After generating such summary of the dataset, we calculate the column wise variance of the dataset.

```
38  #calculating the column wise variance
39  sapply(dataset, function(x) c(sum=sum(x), var=var(x), sd=sd(x)))
40
41  cor(dataset, method = c("pearson"))
```

```
> sapply(dataset, function(x) c(sum=sum(x), var=var(x), sd=sd(x)))
             obs         year        month          day         date     latitude    longitude     zon.winds
sum 8.950456e+09 8.907525e+06 6.106670e+05 1.478797e+06 8.913780e+10 28632.120000 -6654057.820 -314952.60000
var 2.436275e+09 3.704085e+00 1.213311e+01 7.738409e+01 3.680275e+08    22.760447    16571.924      11.71836
sd  4.935864e+04 1.924600e+00 3.483262e+00 8.796823e+00 1.918404e+04     4.770791      128.732       3.42321
       mer.winds     humidity     air_temp     s.s.temp
sum -4364.000000 7.639323e+06 2.542110e+06 2.619106e+06
var     9.127821 2.782842e+01 2.803886e+00 3.504357e+00
sd      3.021228 5.275265e+00 1.674481e+00 1.871993e+00
```

We also calculate the pearson correlation coefficient of different features of the dataset using cor() function. The correlation result obtained are shown below:

```
> cor(dataset, method = c("pearson"))
                  obs        year       month          day         date     latitude    longitude
obs         1.000000e+00  0.10375420 -0.014416822  8.310442e-05  0.10382756 -0.066061948 -0.077338409
year        1.037542e-01  1.00000000 -0.186230314 -1.650334e-02  0.99984067  0.022338535 -0.034951130
month      -1.441682e-02 -0.18623031  1.000000000  1.397444e-02 -0.16866828 -0.001172944 -0.008975828
day         8.310442e-05 -0.01650334  0.013974444  1.000000e+00 -0.01584435 -0.001630223 -0.002007378
date        1.038276e-01  0.99984067 -0.168668284 -1.584435e-02  1.00000000  0.022388635 -0.035227904
latitude   -6.606195e-02  0.02233853 -0.001172944 -1.630223e-03  0.02238864  1.000000000  0.096650818
longitude  -7.733841e-02 -0.03495113 -0.008975828 -2.007378e-03 -0.03522790  0.096650818  1.000000000
zon.winds  -5.325591e-02  0.02628708  0.063149029  5.034420e-03  0.02752089  0.117910964  0.364256266
mer.winds  -1.149686e-01 -0.08510761  0.265412871  5.668425e-03 -0.08056076 -0.092177641 -0.024334832
humidity   -2.127996e-02 -0.01178534 -0.132518308 -2.519505e-04 -0.01422967  0.158110703 -0.042777213
air_temp    9.700228e-02  0.05575506 -0.134154398 -7.447595e-03  0.05349586  0.076123327  0.249049776
s.s.temp    1.131862e-01  0.05368943 -0.098812652 -4.826383e-03  0.05206646  0.125118887  0.304026760
```

	zon.winds	mer.winds	humidity	air_temp	s.s.temp
obs	-0.05325591	-0.114968561	-0.0212799637	0.097002284	0.113186177
year	0.02628708	-0.085107611	-0.0117853442	0.055755057	0.053689428
month	0.06314903	0.265412871	-0.1325183084	-0.134154398	-0.098812652
day	0.00503442	0.005668425	-0.0002519505	-0.007447595	-0.004826383
date	0.02752089	-0.080560756	-0.0142296673	0.053495859	0.052066463
latitude	0.11791096	-0.092177641	0.1581107031	0.076123327	0.125118887
longitude	0.36425627	-0.024334832	-0.0427772131	0.249049776	0.304026760
zon.winds	1.00000000	0.079762856	0.0635534633	0.233155956	0.376015071
mer.winds	0.07976286	1.000000000	0.0776474601	-0.339253593	-0.284897302
humidity	0.06355346	0.077647460	1.0000000000	-0.388058598	-0.324347810
air_temp	0.23315596	-0.339253593	-0.3880585979	1.000000000	0.940233008
s.s.temp	0.37601507	-0.284897302	-0.3243478105	0.940233008	1.000000000

Fig: Screenshot of the correlation result obtained from the dataset

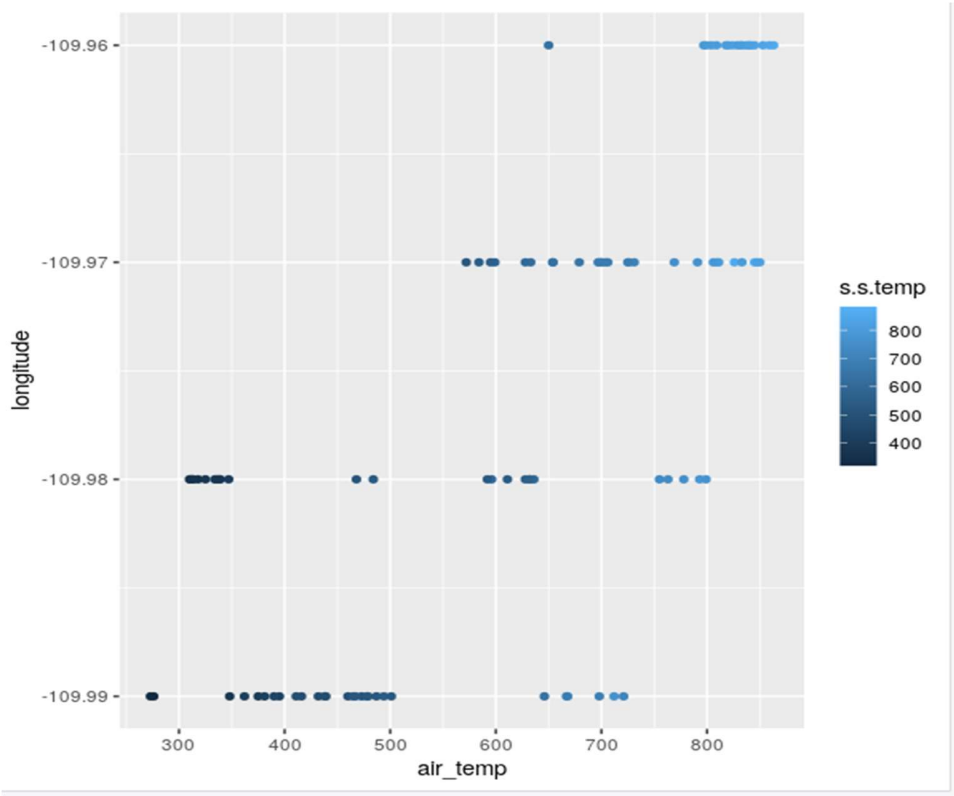
The correlation coefficient indicates the level of linear relationship between the two variables. The correlation coefficient close to -1 indicates strong negative linear relationship whereas close to +1 indicates strong positive linear relationship. From the above correlation plot, we can deduce that, “air_temp” and “s_s. Temp” are highly correlated with a strong positive linear relationship. In the wind data, both zonal and southern winds fluctuated from -10 m/s to 10 m/s. No linear relation was seen in the plot of the two wind variables. Also, against the other three meteorological results, the plots of and wind variable showed no linear relationships. In the tropical Pacific, relative humidity levels usually vary from 70 percent to 90 percent.

Both the temperature of the air and the temperature of the water surface ranged from 20 to 30 degrees Celsius. A positive linear relationship is seen in the plot of the two temperature variables. There are also equivalent plot designs for the two temperatures as one is plotted against time. There was no linear relation between the graphs of the other meteorological variables and the temperature variables.

After correlation plot, we performed data visualization using scatter plot considering only the first 100 instances of the data of the air temperature against longitude and lattitude.

```
45  ggplot(dataset[1:100,], aes(x=air_temp,y=longitude, colour = s.s.temp))
46  + geom_point(position = position_dodge(width = .3))
47  ggplot(dataset[1:100,], aes(x=air_temp,y=latitude, colour = s.s.temp))
48  + geom_point(position = position_dodge(width = .3))
```

Following scatter plot was obtained



Scatter plot of first 100 instances of data of longitude and air temperature

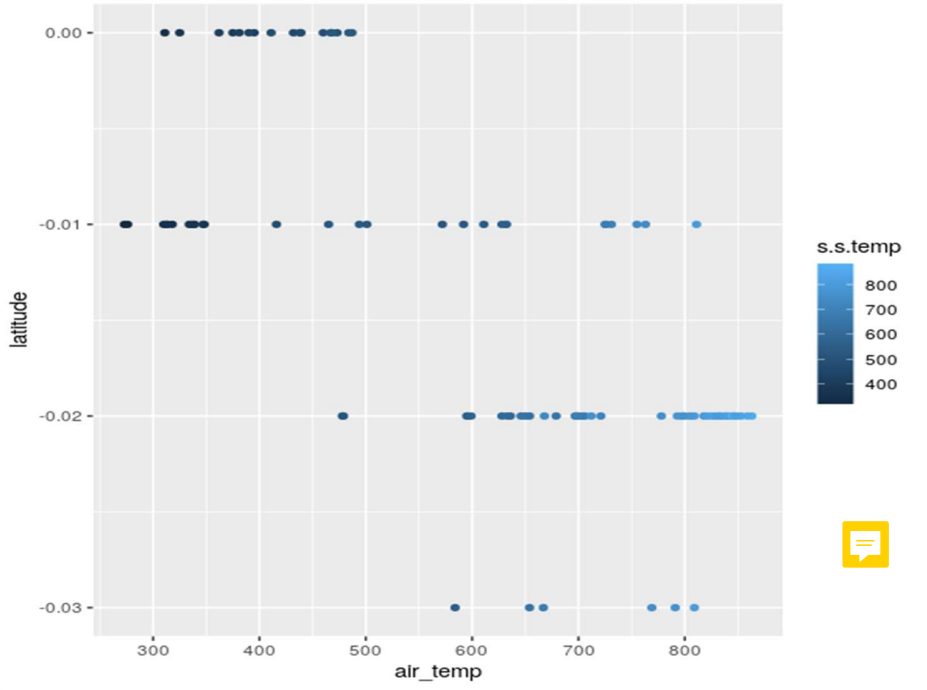


Fig: Screenshot showing the scatter plot of first 100 instances of data of latitude and air temperature

R-Code:

```
library(dplyr)
library(tidyverse)
library(ggpubr)
library(Hmisc)
library(psych)
library(ggplot2)

#reading dataset
dataset<-read.table("tao-all2.dat")
names(dataset) <- c("obs","year","month","day","date","latitude",
                    "longitude","zon.winds","mer.winds","humidity","air_temp",
                    "s.s.temp")
head(dataset,10)

#data validation
stopifnot(ncol(dataset)==12)
stopifnot(nrow(dataset)==178080)

#checking if data contains na values to check if cleaning is required
row.has.na <- apply(dataset, 1, function(x){any(is.na(x))})
sum(row.has.na)

#sum(row.has.na) is 0 so there are no nan values. performing further cleaning of missing values
str(dataset)
dataset<-dataset [ dataset$humidity != ".", ]
dataset<-dataset [ dataset$mer.winds != ".", ]
dataset<-dataset [ dataset$zon.winds != ".", ]
dataset<-dataset [ dataset$air_temp != ".", ]
dataset<-dataset [ dataset$s.s.temp != ".", ]
head(dataset)

#converting non numeric columns to numeric now
dataset[, c(8:12)] <- sapply(dataset[, c(8:12)], as.numeric)


str(dataset)
summary(dataset)
#calculating the column wise variance
sapply(dataset, function(x) c(sum=sum(x), var=var(x), sd=sd(x)))
```

```
cor(dataset, method = c("pearson"))
#corelation matrix here describe

#datavisualization with scatterplot of 100 rows
ggplot(dataset[1:100,], aes(x=air_temp,y=longitude, colour = s.s.temp))
+ geom_point(position = position_dodge(width = .3))
ggplot(dataset[1:100,], aes(x=air_temp,y=latitude, colour = s.s.temp))
+ geom_point(position = position_dodge(width = .3))
```

HCV Data Analysis

Description

HCV dataset is the dataset containing laboratory values and information such as demography, sex of several blood donors and Hepatitis C patients. The main objective of this dataset is to identify whether an observation represents blood donor, suspect blood donor, hepatitis, fibrosis or cirrhosis. There are 10 laboratory values affecting the result and demographic information like age, sex and id or no of the patient in the dataset.

The features in the dataset are discussed below:

- Patient ID/ No
- Age
- Sex
- ALB
- ALP
- AST
- BIL
- CHE
- CHOL
- CREA
- GGT
- PROT

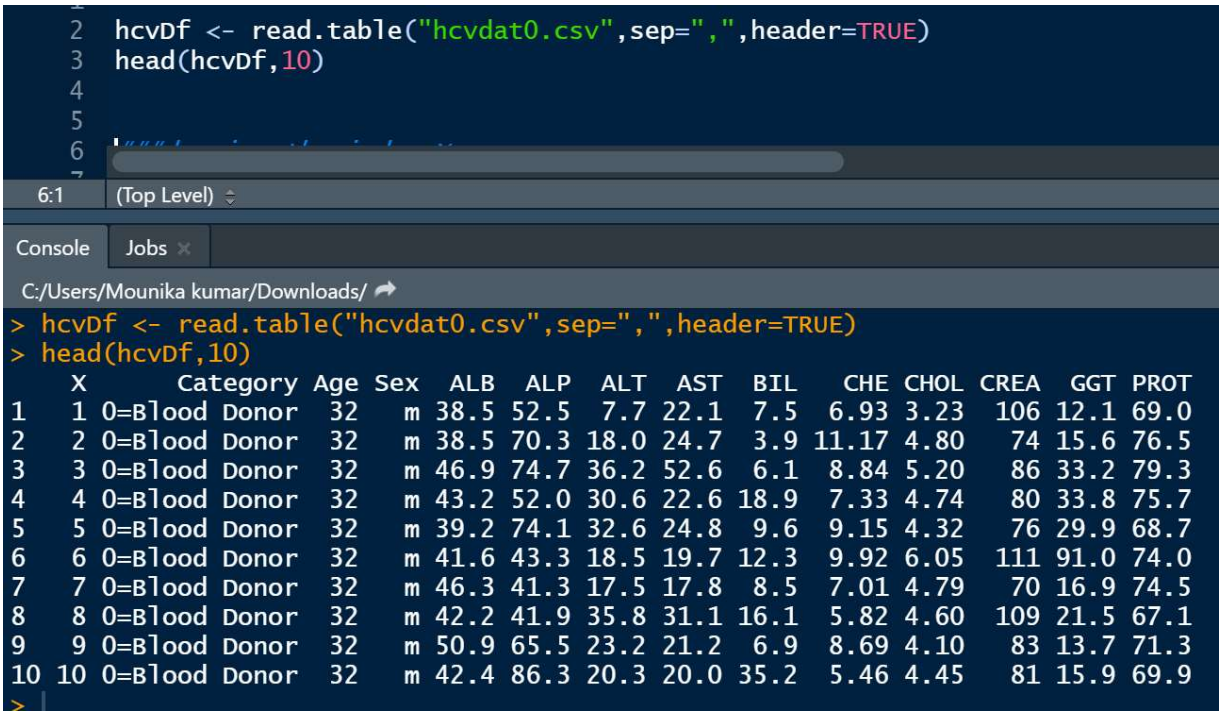
According to such features, the class category of the patient which should be classifies are listed out below:

1. Class 0 representing Blood Donor
2. Class 0's representing suspect Blood Donor
3. Class 1 representing hepatitis
4. Class 2 representing Fibrosis
5. Class 3 representing Cirrhosis

The main objective in this dataset is to classify patient to above classes according to thier demographic data like age, sex and laboratry data such as ALB, ALP and os on.

Data Preparation

The dataset was downloaded from the UCI ML repository website and loaded using the read.table method in R. First ten observations of the data are shown below.



```
2  hcvDf <- read.table("hcvdat0.csv",sep=",",header=TRUE)
3  head(hcvDf,10)
4
5
6
```

6:1 (Top Level)

Console Jobs

C:/Users/Mounika kumar/Downloads/

```
> hcvDf <- read.table("hcvdat0.csv",sep=",",header=TRUE)
> head(hcvDf,10)
    X      Category Age Sex  ALB  ALP  ALT  AST  BIL   CHE CHOL CREA  GGT PROT
1   1 0=Blood Donor  32   m 38.5 52.5  7.7 22.1  7.5  6.93 3.23  106 12.1 69.0
2   2 0=Blood Donor  32   m 38.5 70.3 18.0 24.7  3.9 11.17 4.80   74 15.6 76.5
3   3 0=Blood Donor  32   m 46.9 74.7 36.2 52.6  6.1  8.84 5.20   86 33.2 79.3
4   4 0=Blood Donor  32   m 43.2 52.0 30.6 22.6 18.9  7.33 4.74   80 33.8 75.7
5   5 0=Blood Donor  32   m 39.2 74.1 32.6 24.8  9.6  9.15 4.32   76 29.9 68.7
6   6 0=Blood Donor  32   m 41.6 43.3 18.5 19.7 12.3  9.92 6.05  111 91.0 74.0
7   7 0=Blood Donor  32   m 46.3 41.3 17.5 17.8  8.5  7.01 4.79   70 16.9 74.5
8   8 0=Blood Donor  32   m 42.2 41.9 35.8 31.1 16.1  5.82 4.60  109 21.5 67.1
9   9 0=Blood Donor  32   m 50.9 65.5 23.2 21.2  6.9  8.69 4.10   83 13.7 71.3
10 10 0=Blood Donor  32   m 42.4 86.3 20.3 20.0 35.2  5.46 4.45   81 15.9 69.9
>
```

Fig: Screenshot showing first 10 observations of HCV dataset

Data Preprocessing

Following processes were involved in the data preprocessing step:

- Convert non numeric data to numeric
  The category column values are represented as “0=Blood Donor”. However, for classification, we should convert it into numeric form. Thus, initially, such non numeric columns are converted in numeric form. For this a new column, class is added which has the numeric representation of the category class and category class is removed. The column “Sex” is also converted to numeric form by replacing the “m” value with 0 and “f” value with 1.
- Balancing the dataset
  Number of data representing each category is not equal and an imbalanced dataset leads to poor classification. Thus, to balance the dataset, we remove certain data from category 0 as a process of undersampling.
- Removing null values

While checking the null values in the dataset, there were 19 observations containing null values. Such null values were dropped.

After performing above steps in the dataset in R, the first ten observations of the dataset are shown below:

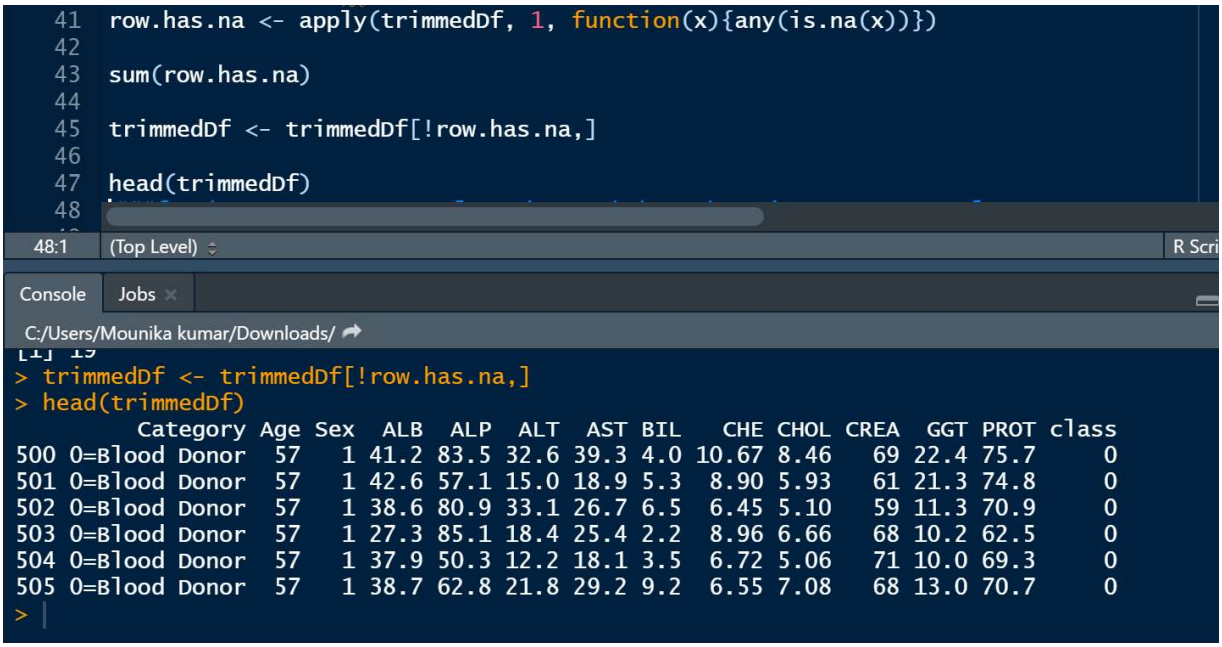
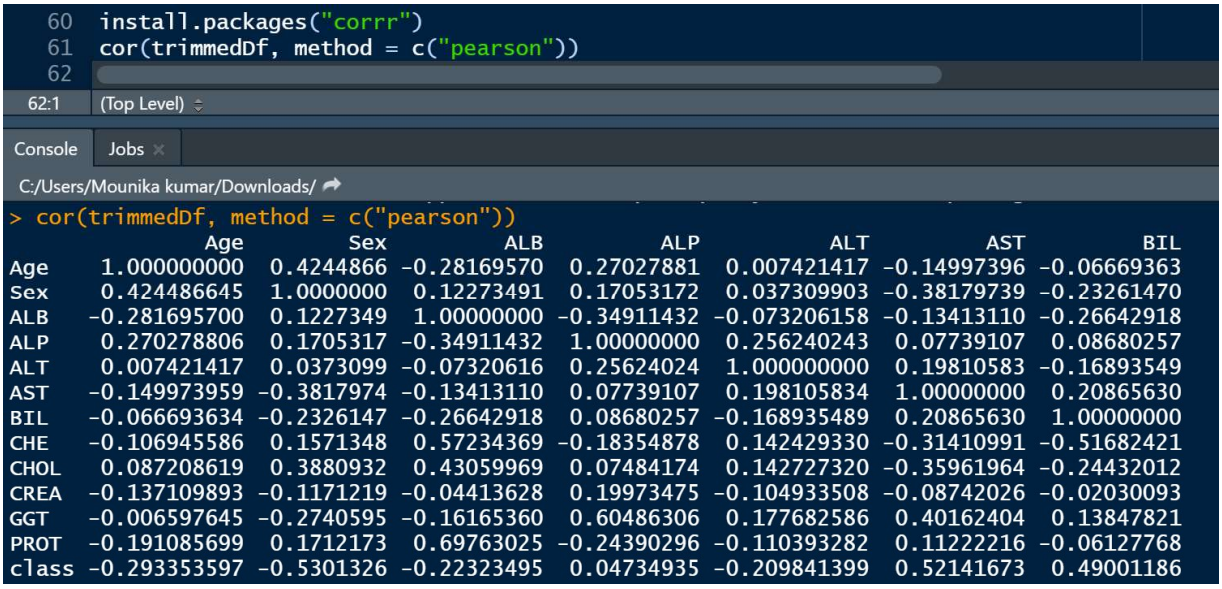


Fig: Screenshot representing first 6 instances of data after preprocessing

Data Analysis

- **Correlation Plot**

Initially, we evaluate the correlation between the columns in the dataset. Following correlation plot is obtained.



	CHE	CHOL	CREA	GGT	PROT	class
Age	-0.1069456	0.08720862	-0.13710989	-0.006597645	-0.19108570	-0.29335360
Sex	0.1571348	0.38809316	-0.11712191	-0.274059536	0.17121734	-0.53013257
ALB	0.5723437	0.43059969	-0.04413628	-0.161653598	0.69763025	-0.22323495
ALP	-0.1835488	0.07484174	0.19973475	0.604863061	-0.24390296	0.04734935
ALT	0.1424293	0.14272732	-0.10493351	0.177682586	-0.11039328	-0.20984140
AST	-0.3141099	-0.35961964	-0.08742026	0.401624043	0.11222216	0.52141673
BIL	-0.5168242	-0.24432012	-0.02030093	0.138478214	-0.06127768	0.49001186
CHE	1.0000000	0.51411856	-0.08251360	-0.226743533	0.25154733	-0.53494946
CHOL	0.5141186	1.00000000	-0.10310494	-0.109729771	0.29843494	-0.54406212
CREA	-0.0825136	-0.10310494	1.00000000	0.099302994	-0.07352305	0.25359435
GGT	-0.2267435	-0.10972977	0.09930299	1.000000000	-0.01720549	0.35085312
PROT	0.2515473	0.29843494	-0.07352305	-0.017205493	1.00000000	0.09081886
class	-0.5349495	-0.54406212	0.25359435	0.350853123	0.09081886	1.00000000

Fig: Correlation plot of the different categorical variables with the response variable

The correlation plot is obtained using the “corrr” package of R. This plot is the Pearson correlation plot of each and every categorical variables in the dataset. The correlation coefficient indicates the level of linear relationship between the two variables. The correlation coefficient close to -1 indicates strong negative linear relationship whereas close to +1 indicates strong positive linear relationship. From above correlation plot, we can deduce that the features “AST” and “BIL” have moderate positive linear relationship with the class column. Similarly, features like “Sex” and “CHE” have moderate negative relationship with the class column. We can also deduce the similar features from the correlation plot. We can deduce that features like “PROT” and “ALB” are highly correlated and columns like “ALP” and “GGT” are also highly correlated. These similar features can be discarded in the feature selection step. Thus from correlation plot evaluation, we discard features like “ALB” and “GGT”.

- **Scatter Plot**

Scatter plot is generated for the dataset using the pairs.panels function from the “psych” package in R. This feature is used to produce a matrix scatter plot with bivariate scatter plots below the diagonal, histograms on the diagonal, and the correlation of Pearson above the diagonal. Following scatter plot is obtained using this package.

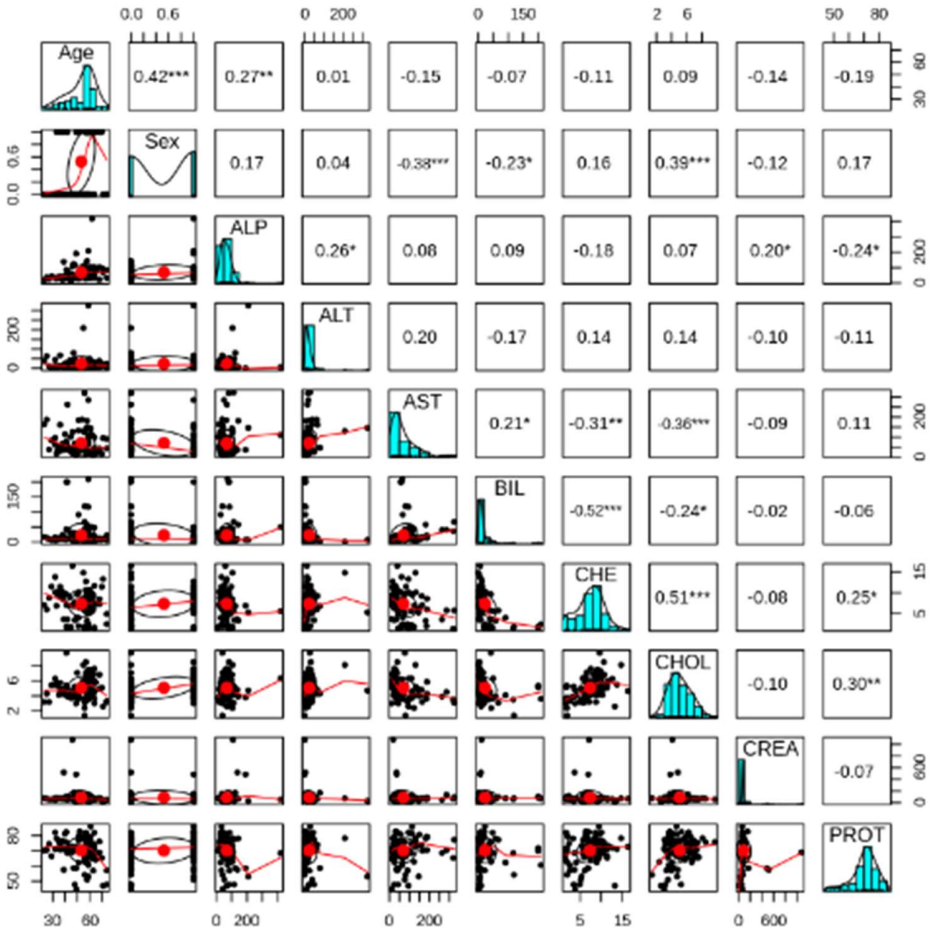


Fig: Scatter plot of the feature selected HVC dataset

Summary of the trimmed dataframe is also generated using summary() method in R. Following is the summary generated.

From the above scatter plot, we can evaluate that features such as “Sex” follows binomial distribution whereas features like “PROT”, “CHE”, “CHOL”, “Age” follows normal distribution.

```
trimmedDf<-subset(trimmedDf, select = -c(ALB,GGT))
install.packages("psych")
library(psych)
pairs.panels(trimmedDf[0:10], stars = TRUE)
summary(trimmedDf)
```

```
> summary(trimmedDf)
      Age              Sex              ALP              ALT              AST        
 Min.   :23.00   Min.   :0.0000   Min.   : 11.30   Min.   :  0.90   Min.   : 10.60  
 1st Qu.:46.00   1st Qu.:0.0000   1st Qu.: 39.30   1st Qu.:  8.30   1st Qu.: 26.40  
 Median :57.00   Median :1.0000   Median : 59.90   Median : 15.00   Median : 41.30  
 Mean   :53.27   Mean   :0.5258   Mean   : 69.03   Mean   : 23.98   Mean   : 70.39  
 3rd Qu.:60.00   3rd Qu.:1.0000   3rd Qu.: 85.10   3rd Qu.: 26.50   3rd Qu.: 95.40  
 Max.   :74.00   Max.   :1.0000   Max.   :416.60   Max.   :325.30   Max.   :324.00  
      BIL              CHE              CHOL             CREA             PROT       
 Min.   :  0.80   Min.   : 1.420   Min.   :1.430   Min.   :   9.00   Min.   :44.80  
 1st Qu.:  6.00   1st Qu.: 5.750   1st Qu.:4.010   1st Qu.:  60.50   1st Qu.:66.80  
 Median : 10.00   Median : 7.510   Median :4.890   Median :  66.90   Median :71.30  
 Mean   : 22.81   Mean   : 7.374   Mean   :5.071   Mean   :  89.64   Mean   :70.17  
 3rd Qu.: 19.00   3rd Qu.: 9.450   3rd Qu.:6.080   3rd Qu.:  76.70   3rd Qu.:75.70  
 Max.   :209.00   Max.   :16.410   Max.   :9.670   Max.   :1079.10   Max.   :86.00  
     class      
 Min.   :0.000  
 1st Qu.:0.000  
 Median :2.000  
 Mean   :1.845  
 3rd Qu.:3.000  
 Max.   :4.000
```

Fig: Summary of the trimmed dataset

Classification

The major task in this dataset is to classify the patient whether they belong to “Blood Donor” or “Suspect Blood Donor” or “Hepatitis” or “Fibrosis” or “Cirrhosis”. To develop a classification model, extreme gradient boosting is used. The extreme gradient boosting is done through the use of “xgboost” package in R. Initially the cleaned and preprocessed to a train test split of ratio 0.75: train and 0.25: test. After the train test split, the xgb.DMatrix is prepared for the entire dataset separately for training and testing data. The xgb model developed using this package is subjected to fitting the training dataset and also a k fold cross validation with k =5. After the training of the model and validating model with 5 fold cross validation, out-of-fold `prediction errors were assessed. After assessing such errors, the out-of-fold prediction were obtained as follows:

	X1	X2	X3	X4	X5	max_prob	label
1	0.008672563	0.007500616	0.962205410	0.013373098	0.008248290	3	3
2	0.971750736	0.010804682	0.005258259	0.005274942	0.006911426	1	1
3	0.130634144	0.090780802	0.197728708	0.457172155	0.123684257	4	4
4	0.009848676	0.009375906	0.957201004	0.013822731	0.009751710	3	3
5	0.008947621	0.007084195	0.960911453	0.013860202	0.009196499	3	3
6	0.010673221	0.027824568	0.936309218	0.015313293	0.009879630	3	3

Fig: Out-of-fold prediction errors obtained from xgboost 5 fold cross validation

For the evaluation of the multi-class classification model, confusion matrix and other evaluation metrics were generated from the model subjecting the model with actual test data and the predictions generated from the model from xgboost classification model. Following classification results were obtained from the model:

```
Confusion Matrix and Statistics

          Reference
Prediction  1  2  3  4  5
         1 24  0  0  0  0
         2  0  3  0  0  0
         3  0  0 15  0  0
         4  0  0  0 10  0
         5  0  0  0  0 20

Overall Statistics

               Accuracy : 1
                 95% CI : (0.9501, 1)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA
```

```
Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity            1.0000  1.00000   1.0000   1.0000   1.0000
Specificity            1.0000  1.00000   1.0000   1.0000   1.0000
Pos Pred Value         1.0000  1.00000   1.0000   1.0000   1.0000
Neg Pred Value         1.0000  1.00000   1.0000   1.0000   1.0000
Precision              1.0000  1.00000   1.0000   1.0000   1.0000
Recall                 1.0000  1.00000   1.0000   1.0000   1.0000
F1                     1.0000  1.00000   1.0000   1.0000   1.0000
Prevalence             0.3333  0.04167   0.2083   0.1389   0.2778
Detection Rate         0.3333  0.04167   0.2083   0.1389   0.2778
Detection Prevalence   0.3333  0.04167   0.2083   0.1389   0.2778
Balanced Accuracy      1.0000  1.00000   1.0000   1.0000   1.0000
```

Fig: Classification report for the test data in HCV dataset

Thus, using the xgboost classification technique for this dataset, we achieved an overall accuracy of 95.8% in this dataset. The F1 score of the classification model is very high for three classes whereas around 0.88 F_score was obtained for the other two classes.

R Code:

```
hcvDf <- read.table("hcvdat0.csv",sep=",",header=TRUE)
head(hcvDf,10)

###droping the index X
newdf<-subset(hcvDf, select = -c(X))
head(newdf,5)

###converting non numeric columns to numeric

newdf<-transform(newdf,Sex=ifelse((Sex=="m"),0,1))
```

```
myFunction <- function(x){
  my_category <- x[1][1]
  value = 0
  if (my_category=='0=BloodDonor'){
    value = 0
  }else if(my_category=='0s=suspect Blood Donor'){
    value = 1
  }else if(my_category=="1=Hepatitis"){
    value = 2
  }else if(my_category=="2=Fibrosis"){
    value = 3
  }else if(my_category=="3=Cirrhosis"){
    value = 4
  }
  return (value)
  #further values ignored (if there are more than 2 columns)
  value <- if(a==b) a + b else b - a
  #or more complicated stuff
  return(value)
}

newdf$class <- apply(newdf, 1, myFunction)

### the number of data of each category is not equal from the above analysis. so removing
certain data from category 0 to balance the dataset
trimmedDf<-newdf[500:615,]

### removing the na values
row.has.na <- apply(trimmedDf, 1, function(x){any(is.na(x))})

sum(row.has.na)

trimmedDf <- trimmedDf[!row.has.na,]

head(trimmedDf)
###further no category column is needed so dropping category column
trimmedDf<-subset(trimmedDf, select = -c(Category))

head(trimmedDf)
###reindexing the dataframe

row.names(trimmedDf)<-NULL
```

```
head(trimmedDf,1)

###checking the correlation between different categorical variables with the response variable

install.packages("corrr")
cor(trimmedDf, method = c("pearson"))
####here PROT and ALB are highly correlated,ALP and GGT are also highly corellated so we
can keep one and discard the other.
#discarding ALB and GGT
trimmedDf<-subset(trimmedDf, select = -c(ALB,GGT))
install.packages("psych")
library(psych)
pairs.panels(trimmedDf[0:10], stars = TRUE)
summary(trimmedDf)

### performing multiclass classification on the data
install.packages("xgboost")
library("xgboost")

train_index <- sample(1:nrow(trimmedDf), nrow(trimmedDf)*0.75)
# Full data set
data_variables <- as.matrix(trimmedDf[,-1])
data_label <- trimmedDf[,"class"]
data_matrix <- xgb.DMatrix(data = as.matrix(trimmedDf), label = data_label)
# split train data and make xgb.DMatrix
train_data   <- data_variables[train_index,]
train_label  <- data_label[train_index]
train_matrix <- xgb.DMatrix(data = train_data, label = train_label)
# split test data and make xgb.DMatrix
test_data  <- data_variables[-train_index,]
test_label <- data_label[-train_index]
test_matrix <- xgb.DMatrix(data = test_data, label = test_label)

###fitting the model and using k-fold for the error estimation..

numberOfClasses <- length(unique(trimmedDf$class))
xgb_params <- list("objective" = "multi:softprob",
                   "eval_metric" = "mlogloss",
                   "num_class" = numberOfClasses)
nround    <- 50 # number of XGBoost rounds
cv.nfold  <- 5

# Fit cv.nfold * cv.nround XGB models and save OOF predictions
cv_model <- xgb.cv(params = xgb_params,
```

```
                data = train_matrix,
                nrounds = nround,
                nfold = cv.nfold,
                verbose = FALSE,
                prediction = TRUE)
###Assess Out-of-Fold Prediction Error

OOF_prediction <- data.frame(cv_model$pred) %>%
  mutate(max_prob = max.col(., ties.method = "last"),
         label = train_label + 1)
head(OOF_prediction)

###confusion matrix
install.packages("caret")
library("caret")
install.packages('e1071', dependencies=TRUE)

# confusion matrix
confusionMatrix(factor(OOF_prediction$max_prob),
                factor(OOF_prediction$label),
                mode = "everything")
```

Early Stage Diabetics

Description

This dataset is the data for the prediction of early stage diabetes. This dataset was prepared from a direct questionnaire from the patients of Sylhet Disease Hospital which is in Bangladesh. This dataset is prepared for a classification task for determining whether a patient is suffering from early stage diabetes or not.

The dataset consists of the following columns representing features of the dataset.

- Age
  The values of the age column ranges from 20 to 60.
- Sex
  The sex column values is indicated as Male and Female
- Polyuria
  The polyuria column indicates whether the patient has polyuria or not. The indication is done as “‘Yes” for having Polyuria and “No” for not having Polyuria.
- Sudden Weight Loss
  This column indicates whether the patient has sudden weight loss or not. The values are “Yes” and “No”.
- Weakness
  This column indicates whether the patient has weakness or not. The values are “Yes” and “No”.

- Polyphagia
  This column indicates whether the patient has Polyphagia or not. The values are “Yes” and “No”.
- Genital Thrush
  This column indicates whether the patient has Genital Thrush or not. The values are “Yes” and “No”.
- Visual Blurring
  This column indicates whether the patient has Visual Blurring or not. The values are “Yes” and “No”.
- Itching
  This column indicates whether the patient has Itching or not. The values are “Yes” and “No”.
- Irritability
  This column indicates whether the patient has Irritability or not. The values are “Yes” and “No”.
- Delayed Healing
  This column indicates whether the patient has Delayed Healing or not. The values are “Yes” and “No”.
- Partial Paresis
  This column indicates whether the patient has Partial Paresis or not. The values are “Yes” and “No”.
- Muscle stickiness
  This column indicates whether the patient has Muscle stickiness or not. The values are “Yes” and “No”.
- Alopecia
  This column indicates whether the patient has Alopecia or not. The values are “Yes” and “No”.
- Obesity
  This column indicates whether the patient has Obesity or not. The values are “Yes” and “No”.

According to these features the class variable is differentiated into two classes:

1. Positive
   This class indicates that the patient has risk of early stage diabetes.
2. Negative
   This class indicates that the patent has no risk of early stage diabetes.

Data Preparation

A csv format data of early stage diabetes dataset was downloaded from the UCI machine learning repository website using read.csv method in R.
Following data were obtained from the dataset.

```
diabetesDf<- read.csv("diabetes_data_upload.csv")
###Printing top 10 data
head(diabetesDf,10)
```

```
   Age Gender Polyuria Polydipsia sudden.weight.loss weakness Polyphagia Genital.thrush visual.blurring
1   40   Male       No        Yes                No      Yes         No             No              No
2   58   Male       No         No                No      Yes         No             No             Yes
3   41   Male      Yes         No                No      Yes        Yes             No              No
4   45   Male       No         No               Yes      Yes        Yes            Yes              No
5   60   Male      Yes        Yes               Yes      Yes        Yes             No             Yes
6   55   Male      Yes        Yes                No      Yes        Yes             No             Yes
7   57   Male      Yes        Yes                No      Yes        Yes            Yes              No
8   66   Male      Yes        Yes               Yes      Yes         No             No             Yes
9   67   Male      Yes        Yes                No      Yes        Yes            Yes              No
10  70   Male       No        Yes               Yes      Yes        Yes             No             Yes
   Itching Irritability delayed.healing partial.paresis muscle.stiffness Alopecia Obesity    class
1      Yes           No             Yes              No              Yes      Yes      Yes Positive
2       No           No              No             Yes              No       Yes       No Positive
3      Yes           No             Yes              No              Yes      Yes       No Positive
4      Yes           No             Yes              No              No       No        No Positive
5      Yes          Yes             Yes             Yes              Yes      Yes      Yes Positive
6      Yes           No             Yes              No              Yes      Yes      Yes Positive
7       No           No             Yes             Yes              No       No        No Positive
8      Yes          Yes              No             Yes              Yes      No        No Positive
9      Yes          Yes              No             Yes              Yes      No       Yes Positive
10     Yes          Yes              No              No              No       Yes       No Positive
```

Fig: Screenshot showing first 10 observations in diabetes dataset

Data Preprocessing

The features of the dataset such as Gender, Polyuria and others have character datatype in their data observations. As a first step, we convert such character type data into numeric form for further processing and easy data wrangling.

For example, the values in the Gender column like “Male” are replaced with 1 and “Female” is replaced with 0.

```
diabetesDf<-transform(diabetesDf, Gender=ifelse((Gender == "Female"), 0, 1))
diabetesDf<- transform(diabetesDf,Polyuria=ifelse((Polyuria=="No"),0,1))
diabetesDf<-transform(diabetesDf,Polydipsia=ifelse((Polydipsia=="No"),0,1))
diabetesDf<-transform(diabetesDf,sudden.weight.loss=ifelse((sudden.weight.loss=="No"),0,1))
diabetesDf<-transform(diabetesDf, weakness=ifelse((weakness == "Female"), 0, 1))
diabetesDf<-transform(diabetesDf,Polyphagia=ifelse((Polyphagia=="No"),0,1))
diabetesDf<-transform(diabetesDf,Genital.thrush=ifelse((Genital.thrush=="No"),0,1))
diabetesDf<-transform(diabetesDf,visual.blurring=ifelse((visual.blurring=="No"),0,1))
diabetesDf<-transform(diabetesDf,Itching=ifelse((Itching=="No"),0,1))
diabetesDf<-transform(diabetesDf,Irritability=ifelse((Irritability=="No"),0,1))
diabetesDf<-transform(diabetesDf,delayed.healing=ifelse((delayed.healing=="No"),0,1))
diabetesDf<-transform(diabetesDf,partial.paresis=ifelse((partial.paresis=="No"),0,1))
diabetesDf<-transform(diabetesDf,muscle.stiffness=ifelse((muscle.stiffness=="No"),0,1))
diabetesDf<-transform(diabetesDf,Alopecia=ifelse((Alopecia=="No"),0,1))
diabetesDf<-transform(diabetesDf,Obesity=ifelse((Obesity=="No"),0,1))

diabetesDf<-transform(diabetesDf,class=ifelse((class=="Negative"),0,1))
```

Fig: Code snippet for data preprocessing

After performing such conversion of values in the feature columns of the dataset, the first 9 instances of the dataset are shown below:

```
> head(diabetesDf,9)
  Age Gender Polyuria Polydipsia sudden.weight.loss weakness Polyphagia Genital.thrush visual.blurring
1  40      1        0          1                 0        1          0              0               0
2  58      1        0          0                 0        1          0              0               1
3  41      1        1          0                 0        1          1              0               0
4  45      1        0          0                 1        1          1              1               0
5  60      1        1          1                 1        1          1              0               1
6  55      1        1          1                 0        1          1              0               1
7  57      1        1          1                 0        1          1              1               0
8  66      1        1          1                 1        1          0              0               1
9  67      1        1          1                 0        1          1              1               0
  Itching Irritability delayed.healing partial.paresis muscle.stiffness Alopecia Obesity class
1       1           0               1               0                 1        1      1     1
2       0           0               0               1                 0        1      0     1
3       1           0               1               0                 1        1      0     1
4       1           0               1               0                 0        0      0     1
5       1           1               1               1                 1        1      1     1
6       1           0               1               0                 1        1      1     1
7       0           0               1               1                 0        0      0     1
8       1           1               0               1                 1        0      0     1
9       1           1               0               1                 1        0      1     1
```

Fig: Screenshot showing first 9 observations of diabetes dataset after conversion to numeric data

Data Analysis

Correlation plot

Correlation plot is generated for the dataset using the “corrr” package in R. Pearson coefficient of correlation is developed. The correlation plot was obtained using the following block of code.

```
27  install.packages("corrr")
28  stats::cor(diabetesDf[0:17], method = "pearson")
```

Following correlation plot was obtained.

	Age	Gender	Polyuria	Polydipsia	sudden.weight.loss	weakness
Age	1.00000000	0.062872072	0.19978075	0.13738160	0.064808352	NA
Gender	0.06287207	1.000000000	-0.26889367	-0.31226171	-0.281840104	NA
Polyuria	0.19978075	-0.268893673	1.00000000	0.59860910	0.447206974	NA
Polydipsia	0.13738160	-0.312261709	0.59860910	1.00000000	0.405965224	NA
sudden.weight.loss	0.06480835	-0.281840104	0.44720697	0.40596522	1.000000000	NA
weakness	NA	NA	NA	NA	NA	1
Polyphagia	0.31557686	-0.219968054	0.37387320	0.31683898	0.243510844	NA
Genital.thrush	0.09651862	0.208960967	0.08727265	0.02808109	0.089857760	NA
visual.blurring	0.40272935	-0.208092287	0.23509455	0.33124972	0.068754243	NA
Itching	0.29655889	-0.052496497	0.08828905	0.12871570	-0.004516473	NA
Irritability	0.20162459	-0.013735368	0.23774048	0.20344613	0.140340005	NA
delayed.healing	0.25750101	-0.101977620	0.14987278	0.11569078	0.088139774	NA
partial.paresis	0.23274235	-0.332288337	0.44166358	0.44224917	0.264013522	NA
muscle.stiffness	0.30770277	-0.090541880	0.15293771	0.18072325	0.109756358	NA
Alopecia	0.32169133	0.327871306	-0.14419180	-0.31096369	-0.202727001	NA
Obesity	0.14045834	-0.005395518	0.12656705	0.09869116	0.169293882	NA
class	NA	NA	NA	NA	NA	NA

	Polyphagia	Genital.thrush	visual.blurring	Itching	Irritability
Age	0.31557686	0.09651862	0.40272935	0.296558890	0.20162459
Gender	-0.21996805	0.20896097	-0.20809229	-0.052496497	-0.01373537
Polyuria	0.37387320	0.08727265	0.23509455	0.088289053	0.23774048
Polydipsia	0.31683898	0.02808109	0.33124972	0.128715697	0.20344613
sudden.weight.loss	0.24351084	0.08985776	0.06875424	-0.004516473	0.14034000
weakness	NA	NA	NA	NA	NA
Polyphagia	1.00000000	-0.06371246	0.29354529	0.144390394	0.23946631
Genital.thrush	-0.06371246	1.00000000	-0.14840820	0.125336261	0.16055073
visual.blurring	0.29354529	-0.14840820	1.00000000	0.291191180	0.07709501
Itching	0.14439039	0.12533626	0.29119118	1.000000000	0.11400562
Irritability	0.23946631	0.16055073	0.07709501	0.114005616	1.00000000
delayed.healing	0.26397979	0.13611128	0.17776658	0.453316447	0.12687657
partial.paresis	0.37356944	-0.19561236	0.36415573	0.116668617	0.15157106
muscle.stiffness	0.32003097	-0.10018760	0.41236853	0.215574910	0.20163700
Alopecia	-0.05349779	0.20484654	0.01460357	0.266505732	0.04370776
Obesity	0.02978497	0.05382765	0.10900454	0.001894402	0.12780059
class	NA	NA	NA	NA	NA

	delayed.healing	partial.paresis	muscle.stiffness	Alopecia	Obesity	class
Age	0.25750101	0.232742347	0.30770277	0.32169133	0.140458336	NA
Gender	-0.10197762	-0.332288337	-0.09054188	0.32787131	-0.005395518	NA
Polyuria	0.14987278	0.441663578	0.15293771	-0.14419180	0.126567047	NA
Polydipsia	0.11569078	0.442249174	0.18072325	-0.31096369	0.098691160	NA
sudden.weight.loss	0.08813977	0.264013522	0.10975636	-0.20272700	0.169293882	NA
weakness	NA	NA	NA	NA	NA	NA
Polyphagia	0.26397979	0.373569439	0.32003097	-0.05349779	0.029784975	NA
Genital.thrush	0.13611128	-0.195612364	-0.10018760	0.20484654	0.053827653	NA
visual.blurring	0.17776658	0.364155732	0.41236853	0.01460357	0.109004545	NA
Itching	0.45331645	0.116668617	0.21557491	0.26650573	0.001894402	NA
Irritability	0.12687657	0.151571058	0.20163700	0.04370776	0.127800592	NA
delayed.healing	1.00000000	0.187381620	0.25007828	0.29017936	-0.066338965	NA
partial.paresis	0.18738162	1.000000000	0.23263363	-0.22157580	-0.009401322	NA
muscle.stiffness	0.25007828	0.232633626	1.00000000	0.04075823	0.158910432	NA
Alopecia	0.29017936	-0.221575798	0.04075823	1.00000000	0.029229120	NA
Obesity	-0.06633896	-0.009401322	0.15891043	0.02922912	1.000000000	NA
class	NA	NA	NA	NA	NA	1

Fig: Correlation plot of the early stage diabetes data

The correlation coefficient close to -1 indicates strong negative linear relationship whereas close to +1 indicates strong positive linear relationship. From the correlation plot obtained for the data, we can evaluate that the attributes like “Polyuria” and “Polydipsia” have correlation values very close to +1 with the class variable thus indicating that these attributes have strong positive relationship and they affect the prediction of the class variable highly. Similarly, attributes such as “Gender” and “Alopecia” have correlation values very close to -1 indicating that these attributes have a strong negative relationship with the class column.

Similarly, from the correlation plot, we can observe that the correlation coefficient between the features are less , that means each feature is different from each other and has an effect on the output. Thus, we should keep all features. Similarly, we should also remove the “Weakness”

attribute. We remove the “Weakness” attribute and observe the first 5 observation using following code:

```
> newdf<-subset(diabetesDf, select = -c(weakness) )
> head(newdf,5)
  Age Gender Polyuria Polydipsia sudden.weight.loss Polyphagia Genital.thrush visual.blurring
1  40      1        0          1                  0          0              0               0
2  58      1        0          0                  0          0              0               1
3  41      1        1          0                  0          1              0               0
4  45      1        0          0                  1          1              1               0
5  60      1        1          1                  1          1              0               1
  Itching Irritability delayed.healing partial.paresis muscle.stiffness Alopecia Obesity class
1       1            0               1                0                1       1       1     1
2       0            0               0                1                0       1       0     1
3       1            0               1                0                1       1       0     1
4       1            0               1                0                0       0       0     1
5       1            1               1                1                1       1       1     1
```

Fig: Code for removing “Weakness” attribute

Variance

Variance of each and every features or columns of the dataset was calculated and evaluated using the “dplyr” package in R.

```
33  library(dplyr)
34  newdf %>% summarise_if(is.numeric, var)
```

Following results were obtained.

```
       Age    Gender  Polyuria Polydipsia sudden.weight.loss Polyphagia Genital.thrush
1 147.6581 0.2333482 0.2504669  0.2477805          0.2436305  0.2485216      0.1736475
  visual.blurring   Itching Irritability delayed.healing partial.paresis muscle.stiffness  Alopecia
1       0.2477805 0.2503001    0.1839484       0.2488476       0.2456796        0.2348266 0.2261709
    Obesity class
1 0.1408626     0
```

Fig: Variance obtained for each features

Data Visualization

Scatter Plot

For the visualization of data, scatter plots of each and every features of the dataset was plotted using the “psych” package in R. The pair_plot() function of the package was used for plotting the scatter plot. The number of features being high, visually effective scatter plot was not obtained. Thus, scatter plots of only 4 columns were plotted and the relation between data was analyzed. After plotting the scatter plot for four columns, the following result was obtained.

```
38  pairs.panels(newdf[0:4], stars = TRUE)
```

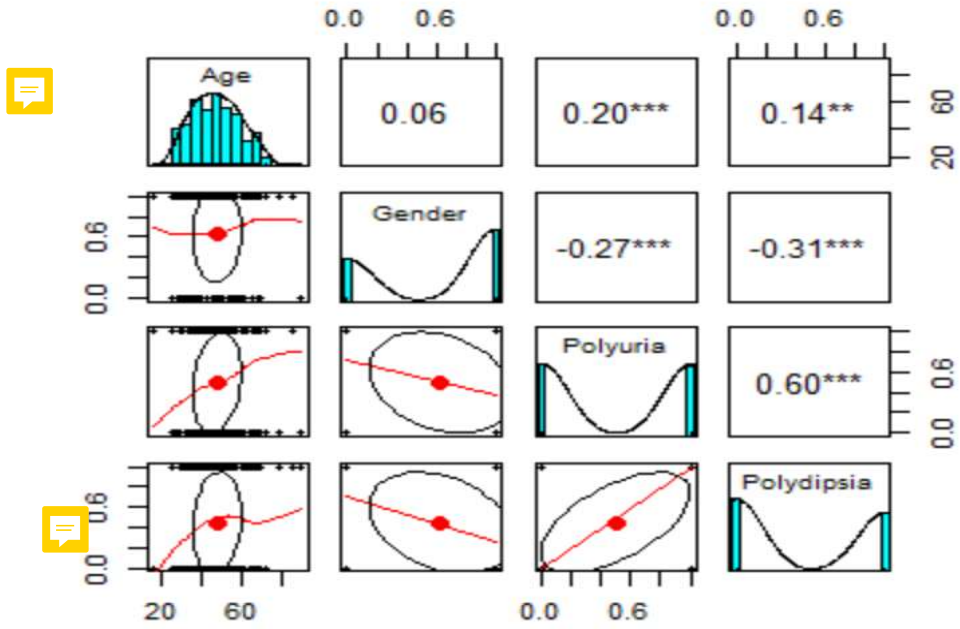


Fig: Scatter plot of the four columns of the dataset.

Scatter plot is done by using the pairs.panels() feature of the “psych” package. This feature is used to produce a matrix scatter plot with bivariate scatter plots below the diagonal, histograms on the diagonal, and the correlation of Pearson above the diagonal. From the above scatter plot, we can evaluate that the “Age” attribute follows normal distribution whereas other attributes are mainly categorical in the dataset.

R Code:

```
#Reading dataframe
diabetesDf<- read.csv("diabetes_data_upload.csv")
#Printing top 10 data
head(diabetesDf,10)
#Printing bottom 10 data for analysis
tail(diabetesDf,10)
#coverting the character to numeric form for easy data wrangling
diabetesDf<-transform(diabetesDf, Gender=ifelse((Gender == "Female"), 0, 1))
diabetesDf<- transform(diabetesDf,Polyuria=ifelse((Polyuria=="No"),0,1))
diabetesDf<-transform(diabetesDf,Polydipsia=ifelse((Polydipsia=="No"),0,1))
diabetesDf<-transform(diabetesDf,sudden.weight.loss=ifelse((sudden.weight.loss=="No"),0,1))
diabetesDf<-transform(diabetesDf, weakness=ifelse((weakness == "Female"), 0, 1))
diabetesDf<-transform(diabetesDf,Polyphagia=ifelse((Polyphagia=="No"),0,1))
diabetesDf<-transform(diabetesDf,Genital.thrush=ifelse((Genital.thrush=="No"),0,1))
```

```
diabetesDf<-transform(diabetesDf,visual.blurring=ifelse((visual.blurring=="No"),0,1))
diabetesDf<-transform(diabetesDf,Itching=ifelse((Itching=="No"),0,1))
diabetesDf<-transform(diabetesDf,Irritability=ifelse((Irritability=="No"),0,1))
diabetesDf<-transform(diabetesDf,delayed.healing=ifelse((delayed.healing=="No"),0,1))
diabetesDf<-transform(diabetesDf,partial.paresis=ifelse((partial.paresis=="No"),0,1))
diabetesDf<-transform(diabetesDf,muscle.stiffness=ifelse((muscle.stiffness=="No"),0,1))
diabetesDf<-transform(diabetesDf,Alopecia=ifelse((Alopecia=="No"),0,1))
diabetesDf<-transform(diabetesDf,Obesity=ifelse((Obesity=="No"),0,1))

diabetesDf<-transform(diabetesDf,class=ifelse((class=="Negative"),0,1))
head(diabetesDf,9)
install.packages("corrr")
stats::cor(diabetesDf[0:17], method = "pearson")
#seems like correlation between the features are less , that means each features are different from each other and has effect on the output. so keeping all features except weakness
newdf<-subset(diabetesDf, select = -c(weakness) )
head(newdf,5)
#seeing the variance
library(dplyr)
newdf %>% summarise_if(is.numeric, var)
install.packages("psych")
library(psych)
#for visualization pair plots have been plotted below. For better figure I would plot only 4 coloums and see relation between the data. Age seems to follow normal distribution while other features are mainly categorical
pairs.panels(newdf[0:4], stars = TRUE)
```