**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)**

**Faculty of Science and Technology (FST)**

**Course Title: INTRODUCTION TO DATA SCIENCE**

**Fall 2023-2024**

**Section: (A), Group: 06**

# Project Title: Apply data preparation steps (which can be applied) and do the univariate data exploration for the given data set.

**Supervised By**

**TOHEDUL ISLAM**

**Faculty of Science and Technology**

**American International University-Bangladesh**

**Submitted By:**



| NAME | ID |
|---|---|
| TONMOY DEY | 20-44206-3 |
| SHOWMITRA ROY | 20-44208-3 |

## Dataset Description:

This is a heart disease classification dataset consist of 151 samples. There are seven variable consisting of age, gender (male, female), impulse, and pressure high, pressure low, glucose, kcm, troponin and the last one is class variable, which is known as the outcome. In the dataset only gender and class variable are consisting of categorical except those all are numerical. Moreover, class variable is also divided in to two categories (positive and negative); if the outcome is positive then there is existence of heart attack. On the other hand, if it is negative then no heart attack.

## Attributes:

**Age:** The age of the individuals.

**Gender:** It gives us idea whether it is male or female.

**Impulse:** An impulse is sudden force or desire.

**Pressure Height:** pressure height and altitude have an impact on heart attack risk.

**Pressure Low:** When the body does not have enough water, the amount of blood in the body decreases.

**Glucose:** It is the main type of sugar in the blood and is the major source of energy for the body's cells.

**Class (Target Variable):** It give us idea about the existence of heart attack, classified as NEGATIVE OR POSITIVE.

**PURPOSE:** The heart disease classification dataset is used to predict whether an individual might suffer from heart disease or not, based on age, gender (male or female), impulse, pressureheight, pressurelow and glucose.

## Project Overview:

A critical step in data analysis is data pre-processing, which is transforming unprocessed data into a format that computers and machine learning systems can easily understand and analyse. In actuality, raw data is often jumbled with plenty of errors, require cleaning before it may be used to a particular task. Moreover, univariate analysis is required, which involves evaluating each variable in a dataset independently without taking the relationships between variables into account.

It is noticeable that the data set is not well formatted. The dataset has to be cleaned and pre-processed before using it.

# Data pre-processing:

### 1. Importing the Dataset:

The dataset is located in a file called Dataset_MId.csv in the current working directory. To begin data pre-processing using R, the first step is to import the dataset. Once imported, the Dataset_MId.csv file is transformed into an R data frame and stored in a variable named "Dataset". After printing the dataset, it looks like this-

**R code:**

**Dataset<-read.csv("F:/Dataset_MId.csv")**

**print(Dataset)**

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> Dataset<-read.csv("F:/Dataset_MId.csv")
> print(Dataset)
    age gender impluse pressurehight pressurelow glucose    class
1    64   male      66           160          83     160 negative
2    21   male      94            98          46     296 positive
3    55   male      64          -160          77     270 negative
4    64   male      70           120          55     270 positive
5    55   male      64           112          65     300 negative
6    58 female      61           112          58      87 negative
7    32 female      40           179          68     102 negative
8    63   male      60           214          82      87 positive
9    44 female      60            NA          81     135 negative
10   67             61           160          95     100 negative
11   NA female      60           166          90     102 negative
12   63 female      60           150          10     198 negative
13   64   male      60           199           5      92 positive
14   54 female      94           122          67      97 negative
15   47   male      76           120          70     319 negative
16   61   male      81            NA          66     134 positive
17   86 female      73           114          68      87 positive
18   45 female      70           100          68      96 negative
19   37 female      72           107          86     274 negative
20   45   male      60           109          65      89 positive
```

## 2. Dealing with Missing Values:

For checking the missing value (NA) present in column name: age[5] gender[0], impulse[0], pressureheight[2], pressurelow[0], glucose[0] and class[0]. We need to use the give code to find the missing value.

**R code:**

**colSums(is.na(Dataset))**

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> colSums(is.na(Dataset))
          age        gender       impluse pressurehight   pressurelow       glucose
            5             0             0             2             0             0
        class
            0
>
```

Before, the dataset look like this -

Mid_Project.R* × Dataset ×

Filter

| | age | gender | impluse | pressurehight | pressurelow | glucose | class |
|---|---|---|---|---|---|---|---|
| 8 | 63 | male | 60 | 214 | 82 | 87 | positive |
| 9 | 44 | female | 60 | NA | 81 | 135 | negative |
| 10 | 67 | | 61 | 160 | 95 | 100 | negative |
| 11 | NA | female | 60 | 166 | 90 | 102 | negative |
| 12 | 63 | female | 60 | 150 | 10 | 198 | negative |
| 13 | 64 | male | 60 | 199 | 5 | 92 | positive |
| 14 | 54 | female | 94 | 122 | 67 | 97 | negative |
| 15 | 47 | male | 76 | 120 | 70 | 319 | negative |
| 16 | 61 | male | 81 | NA | 66 | 134 | positive |
| 17 | 86 | female | 73 | 114 | 68 | 87 | positive |
| 18 | 45 | female | 70 | 100 | 68 | 96 | negative |
| 19 | 37 | female | 72 | 107 | 86 | 274 | negative |
| 20 | 45 | male | 60 | 109 | 65 | 89 | positive |
| 21 | 60 | male | 92 | 151 | 78 | 301 | negative |
| 22 | 48 | male | 135 | 98 | 60 | 100 | positive |
| 23 | 52 | male | 76 | 109 | 85 | 227 | positive |
| 24 | 30 | male | 63 | 110 | 68 | 107 | positive |
| 25 | NA | male | 63 | 320 | 63 | 269 | positive |
| 26 | 72 | male | 64 | 106 | 68 | 111 | positive |

Showing 7 to 26 of 150 entries, 7 total columns

**2.2** Now, as these columns are in the numerical format, we can replace the missing value with the mean value of those columns. R code for replacing missing value by the mean.

**R Code:**

```
Dataset$age <- ifelse(is.na(Dataset$age),mean(Dataset$age, na.rm = TRUE),Dataset$age)

Dataset$pressurehight <- ifelse(is.na(Dataset$pressurehight),mean(Dataset$pressurehight, na.rm = TRUE),Dataset$pressurehight)

print(Dataset)
```

After the conversion:

Mid_Project.R* × Dataset ×

Filter

| | age | gender | impluse | pressurehight | pressurelow | glucose | class |
|---|---|---|---|---|---|---|---|
| 8 | 63.00000 | male | 60 | 214.0000 | 82 | 87 | positive |
| 9 | 44.00000 | female | 60 | 127.0878 | 81 | 135 | negative |
| 10 | 67.00000 | | 61 | 160.0000 | 95 | 100 | negative |
| 11 | 56.13793 | female | 60 | 166.0000 | 90 | 102 | negative |
| 12 | 63.00000 | female | 60 | 150.0000 | 10 | 198 | negative |
| 13 | 64.00000 | male | 60 | 199.0000 | 5 | 92 | positive |
| 14 | 54.00000 | female | 94 | 122.0000 | 67 | 97 | negative |
| 15 | 47.00000 | male | 76 | 120.0000 | 70 | 319 | negative |
| 16 | 61.00000 | male | 81 | 127.0878 | 66 | 134 | positive |
| 17 | 86.00000 | female | 73 | 114.0000 | 68 | 87 | positive |
| 18 | 45.00000 | female | 70 | 100.0000 | 68 | 96 | negative |
| 19 | 37.00000 | female | 72 | 107.0000 | 86 | 274 | negative |
| 20 | 45.00000 | male | 60 | 109.0000 | 65 | 89 | positive |
| 21 | 60.00000 | male | 92 | 151.0000 | 78 | 301 | negative |
| 22 | 48.00000 | male | 135 | 98.0000 | 60 | 100 | positive |
| 23 | 52.00000 | male | 76 | 109.0000 | 85 | 227 | positive |
| 24 | 30.00000 | male | 63 | 110.0000 | 68 | 107 | positive |
| 25 | 56.13793 | male | 63 | 320.0000 | 63 | 269 | positive |
| 26 | 72.00000 | male | 64 | 106.0000 | 68 | 111 | positive |

Showing 7 to 27 of 150 entries, 7 total columns

**2.3** Here we can see that in the "gender" column, some values are missing. We can find it out in this way-

**R Code:**

**Dataset[,2]**

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> Dataset[,2]
  [1] "male"   "male"   "male"   "male"   "male"   "female" "female" "male"   "female"
 [10] ""       "female" "female" "male"   "female" "male"   "male"   "female" "female"
 [19] "female" "male"   "male"   "male"   "male"   "male"   "male"   "male"   "male"
 [28] "female" "female" "male"   "male"   "male"   "male"   "female" "male"   "male"
 [37] "male"   "male"   ""       "male"   "male"   "male"   "male"   "male"   "male"
 [46] ""       "male"   "male"   "male"   "female" "male"   "female" "female" "female"
 [55] "female" "male"   "male"   "male"   "female" "male"   "male"   "female" "male"
 [64] "male"   "male"   "male"   "male"   "male"   "male"   "male"   "male"   "female"
 [73] "male"   "female" "male"   "male"   "female" "male"   "male"   "female" "male"
 [82] "female" "male"   "female" "male"   "female" "male"   "male"   "female" "male"
 [91] "male"   "male"   "female" "female" "male"   "male"   "male"   "female" "male"
[100] "male"   "male"   "male"   "female" "female" "male"   "female" "male"   "male"
[109] "male"   "male"   "male"   "male"   "female" "male"   "male"   "female" "female"
[118] "male"   "male"   "male"   "male"   "female" "male"   "male"   "male"   "male"
[127] "male"   "female" "female" "male"   "male"   "male"   "female" "male"   "male"
[136] "male"   "male"   "female" "male"   "female" "female" "male"   "male"   "female"
[145] "male"   "female" "female" "female" "male"   "male"
>
```

As the gender column is categorical so we can overcome this problem using the most frequent value in the place of missing value.

**R Code:**

**print(max(Dataset$gender))**

**Dataset<-edit(Dataset)**

**Dataset[,2]**

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> print(max(Dataset$gender))
[1] "male"
```

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> Dataset[,2]
  [1] "male"   "male"   "male"   "male"   "male"   "female" "female" "male"   "female"
 [10] "male"   "female" "female" "male"   "female" "male"   "male"   "female" "female"
 [19] "female" "male"   "male"   "male"   "male"   "male"   "male"   "male"   "male"
 [28] "male"   "female" "male"   "male"   "male"   "male"   "female" "male"   "male"
 [37] "male"   "male"   "male"   "male"   "male"   "male"   "male"   "male"   "male"
 [46] "male"   "male"   "male"   "male"   "male"   "male"   "female" "female" "female"
 [55] "female" "male"   "male"   "male"   "female" "male"   "male"   "female" "male"
 [64] "male"   "male"   "male"   "male"   "male"   "male"   "male"   "male"   "female"
 [73] "male"   "female" "male"   "male"   "female" "male"   "male"   "female" "male"
 [82] "female" "male"   "female" "male"   "female" "male"   "male"   "female" "male"
 [91] "male"   "male"   "female" "female" "male"   "male"   "male"   "female" "male"
[100] "male"   "male"   "male"   "female" "female" "male"   "female" "male"   "male"
[109] "male"   "male"   "male"   "male"   "female" "male"   "male"   "female" "female"
[118] "male"   "male"   "male"   "male"   "female" "male"   "male"   "male"   "male"
[127] "male"   "female" "female" "male"   "male"   "male"   "female" "male"   "male"
[136] "male"   "male"   "female" "male"   "female" "female" "male"   "male"   "female"
[145] "male"   "female" "female" "female" "male"   "male"
>
```

## 3. Dealing with Data types and Conversion:

As we can see that couple of columns contain decimal place data. So, to overcome it. We will use the below code to round it up.

**R Code:**

**Dataset$age <- as.numeric(format(round(Dataset$age,0)))**

**Dataset$pressurehight<- as.numeric(format(round(Dataset$pressurehight, 0)))**

**print(Dataset)**

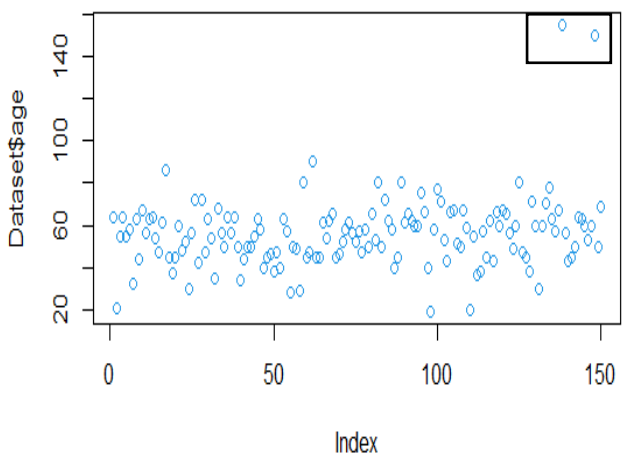
```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> Dataset$age <- as.numeric(format(round(Dataset$age,0)))
> Dataset$pressurehight<- as.numeric(format(round(Dataset$pressurehight,0)))
> print(Dataset)
    age gender impluse pressurehight pressurelow glucose    class
1    64   male      66           160          83     160 negative
2    21   male      94            98          46     296 positive
3    55   male      64          -160          77     270 negative
4    64   male      70           120          55     270 positive
5    55   male      64           112          65     300 negative
6    58 female      61           112          58      87 negative
7    32 female      40           179          68     102 negative
8    63   male      60           214          82      87 positive
9    44 female      60           127          81     135 negative
10   67   male      61           160          95     100 negative
11   56 female      60           166          90     102 negative
12   63 female      60           150          10     198 negative
13   64   male      60           199           5      92 positive
14   54 female      94           122          67      97 negative
15   47   male      76           120          70     319 negative
16   61   male      81           127          66     134 positive
17   86 female      73           114          68      87 positive
18   45 female      70           100          68      96 negative
19   37 female      72           107          86     274 negative
20   45   male      60           109          65      89 positive
21   60   male      92           151          78     301 negative
22   48   male     135            98          60     100 positive
23   52   male      76           109          85     227 positive
24   30   male      63           110          68     107 positive
25   56   male      63           320          63     269 positive
```

## 4. Dealing with Outliers:

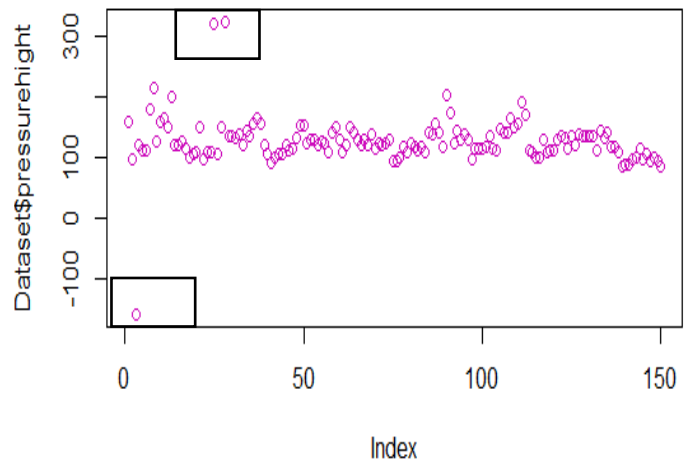
Data, which are different from the rest of the dataset, known as OUTLIERS. To check the outliers, we have applied the below code:

**R Code:**

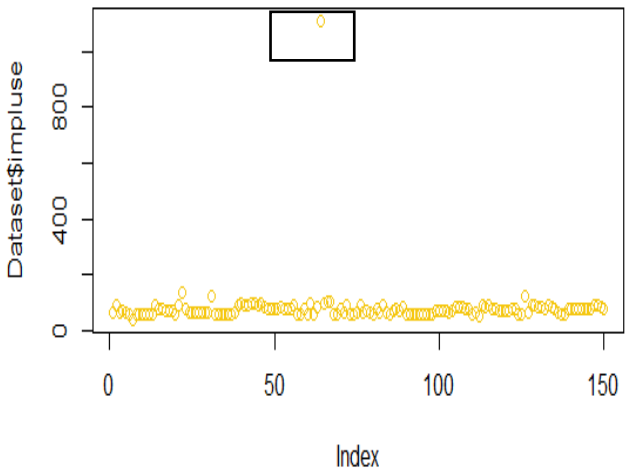
```
plot(Dataset$age,col=4)
plot(Dataset$pressurehight, col=6)
plot(Dataset$impluse, col=7)
plot(Dataset$pressurelow, col=5)
```



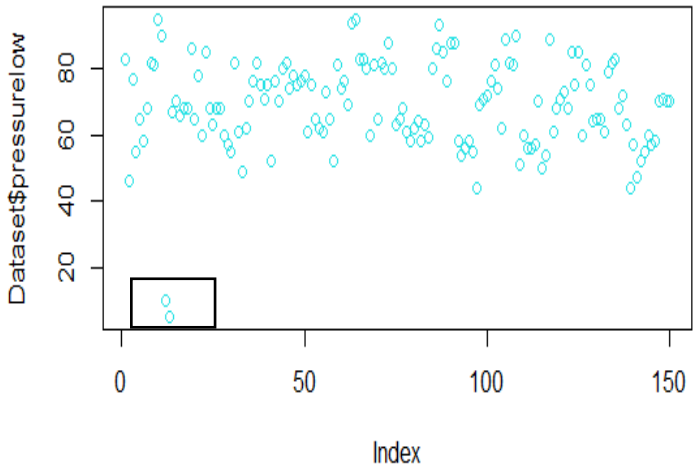

**For Age**




**For Pressureheight**




**For Impluse**




**For Pressurelow**

In conclusion, we can see that in maximum column in our dataset is consist of outliers, except glucose and class.

# Univariate Data Exploration:

In data science, univariate exploration refers to the process of examining each variable in a dataset independently, without taking into account the correlation between them. Gaining a fundamental grasp of a variable's variability, trend, and distribution can be accomplished with this kind of analysis.

## 5. Finding Mean, Median, Variance and Standard Deviation.

To find out the exploration of the age attribute, we have to use the below code written in R.

**R Code:**

**mean(Dataset$age)**

**median(Dataset$age)**

**var(Dataset$age)**

**sd(Dataset$age)**

```
Console   Terminal   Background Jobs
R 4.3.1 · ~/
> mean(Dataset$age)
[1] 56.13333
> median(Dataset$age)
[1] 56
> var(Dataset$age)
[1] 288.9083
> sd(Dataset$age)
[1] 16.9973
```

To find out the exploration of the impulse attribute, we have to use the below code written in R.

**R Code:**

**mean(Dataset$impluse)**

**median(Dataset$impluse)**

**var(Dataset$impluse)**

**sd(Dataset$impluse)**

```
Console   Terminal   Background Jobs
R 4.3.1 · ~/
> mean(Dataset$impluse)
[1] 81.98
> median(Dataset$impluse)
[1] 74
> var(Dataset$impluse)
[1] 7374.812
> sd(Dataset$impluse)
[1] 85.87672
```

To find out the exploration of the pressureheight attribute, we have to use the below code written in R.

**R Code:**

**mean(Dataset$pressurehight)**

**median(Dataset$pressurehight)**

**var(Dataset$pressurehight)**

**sd(Dataset$pressurehight)**

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> mean(Dataset$pressurehight)
[1] 127.0867
> median(Dataset$pressurehight)
[1] 122.5
> var(Dataset$pressurehight)
[1] 1599.489
> sd(Dataset$pressurehight)
[1] 39.99361
```

To find out the exploration of the pressurelow attribute, we have to use the below code written in R.

**R Code:**

**mean(Dataset$pressurelow)**

**median(Dataset$pressurelow)**

**var(Dataset$pressurelow)**

**sd(Dataset$pressurelow)**

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> mean(Dataset$pressurelow)
[1] 68.95333
> median(Dataset$pressurelow)
[1] 69
> var(Dataset$pressurelow)
[1] 186.3535
> sd(Dataset$pressurelow)
[1] 13.65114
```

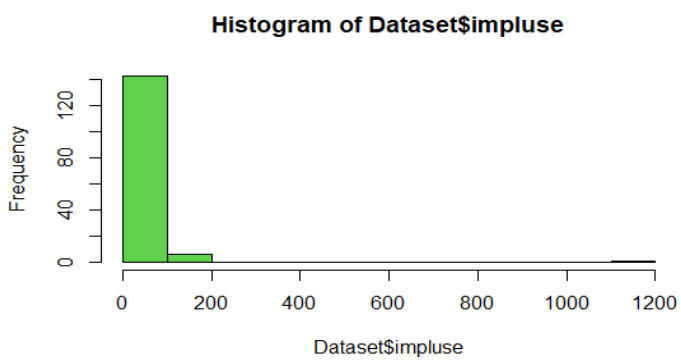
Here, we found out the mean, median, variance and standard deviation of age, impulse, pressureheight, pressurelow.

6. **Now, we draw a histogram for age, impulse, pressureheight, pressurelow and gender attributes for analysis.**

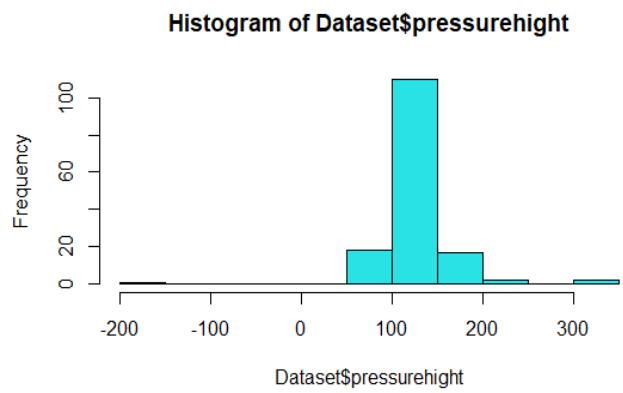
**R Code:**

```
hist(Dataset$age)

hist(Dataset$impluse,col=3)

hist(Dataset$pressurehight,col=5)

hist(Dataset$pressurelow,col=7)

hist(Dataset$glucose,col=4)

barplot(table(Dataset$gender), ylab = "Frequency", xlab = "Gender")
```

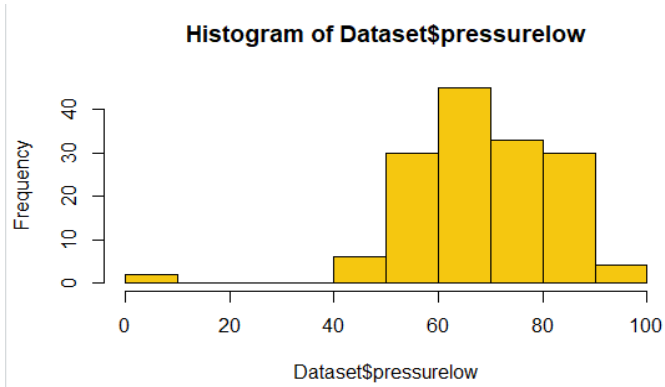

From the above histogram, we can see that the maximum number of people is between 40 to 60. Secondly, there are nearly 45 people between 70 to 80. Moreover, up to 18 people are adult and the rest are the left-over people. From the rest there are also outliers.



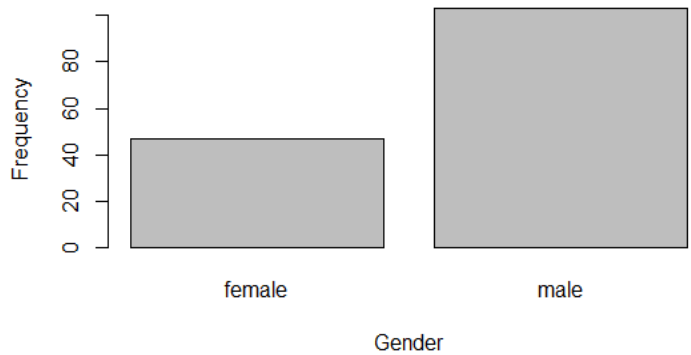

To begin with, we can see that there are nearly 140 people having their impulse within 100 and others within 200. Moreover, there is a tiny presence of outliers too.




The above histogram gives us an idea that nearly two thirds of people having pressure height within 100 to 150. Secondly, few having pressure height above 200. Lastly, there is a presence of outliers too within the above histogram.




From the above diagram, we can see that nearly one-fourth of people having pressure low within 60 to 70. On the other hand, the remaining two thirds are facing low pressure from 50 to 60 and 70 to 80. There is a presence of outliers in the histogram too.



The above histogram gives us a simple ideal about the number of male and female in the dataset. We can simply say that the number male is more than female.

### 7. Standard deviation of each attribute

Here, we also downloaded “dplyr” and “matrixStats” package. To find out the standard deviation of each attribute.

**R Code:**

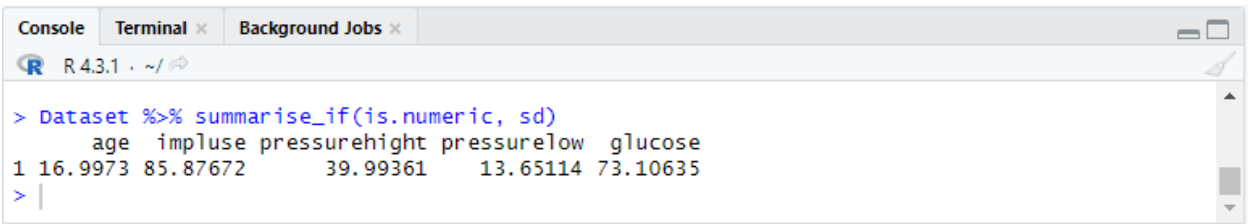
**install.packages("dplyr")**

**install.packages("matrixStats")**

**library(matrixStats)**

**library(dplyr)**

**Dataset %>% summarise_if(is.numeric, sd)**




Here, we calculate the standard deviation of each numerical attribute.

## 8. Removing Outliers:

From the univariate data exploration, we find that the presence of outliers the value of mean mode, variance and standard deviation is bigger. So that we must remove these outliers.

**Remove outliers from age attribute:**

**R Code:**

**age_bounds <- quantile(Dataset$age, c(0.25, 0.75))**

**IQR_age <- IQR(Dataset$age)**

**lower_age <- age_bounds[1] - 1.5 * IQR_age**

**upper_age <- age_bounds[2] + 1.5 * IQR_age**

**Dataset <- Dataset[Dataset$age >= lower_age & Dataset$age <= upper_age,]**

```
plot(Dataset$age,col=4)
```

## Remove outliers from pressurehight attribute:

**R Code:**

```
pressurehight_bounds <- quantile(Dataset$pressurehight, c(0.25, 0.75))

IQR_pressurehight <- IQR(Dataset$pressurehight)

lower_pressurehight <- pressurehight_bounds[1] - 1.5 * IQR_pressurehight

upper_pressurehight <- pressurehight_bounds[2] + 1.5 * IQR_pressurehight

Dataset <- Dataset[Dataset$pressurehight >= lower_pressurehight & Dataset$pressurehight
<= upper_pressurehight,]

plot(Dataset$pressurehight, col=6)
```

## Remove outliers from impluse attribute:

**R Code:**

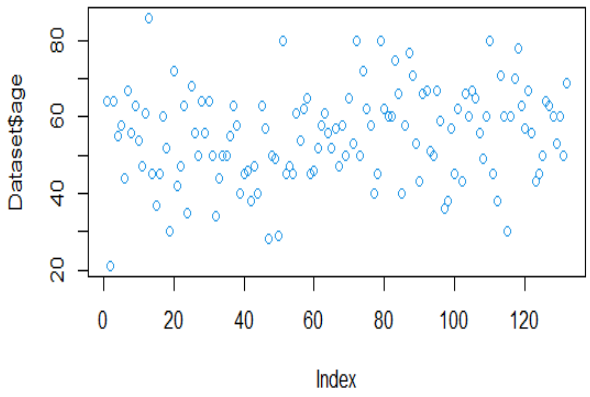
```
impluse_bounds <- quantile(Dataset$impluse, c(0.25, 0.75))

IQR_impluse <- IQR(Dataset$impluse)

lower_impluse <- impluse_bounds[1] - 1.5 * IQR_impluse

upper_impluse <- impluse_bounds[2] + 1.5 * IQR_impluse

Dataset <- Dataset[Dataset$impluse >= lower_impluse & Dataset$impluse <=
upper_impluse, ]

plot(Dataset$impluse, col=7)
```

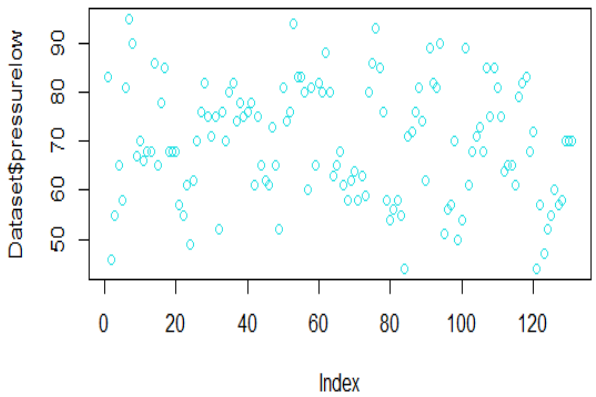
## Remove outliers from pressurelow attribute:

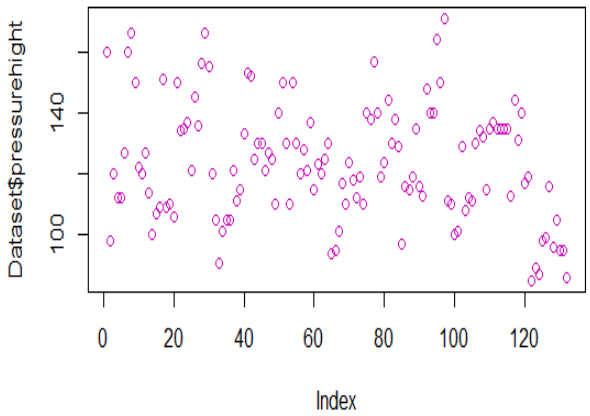
**R Code:**

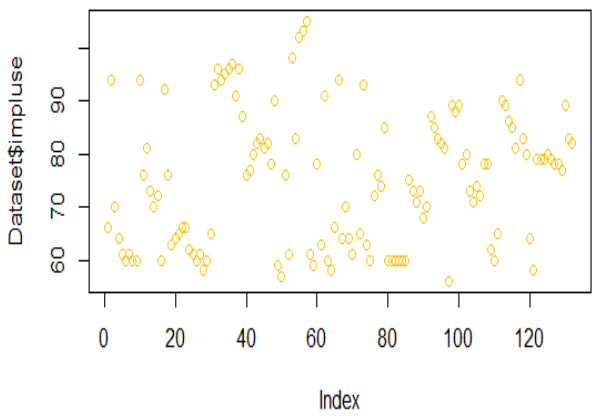
```
pressurelow_bounds <- quantile(Dataset$pressurelow, c(0.25, 0.75))

IQR_pressurelow <- IQR(Dataset$pressurelow)

lower_pressurelow <- pressurelow_bounds[1] - 1.5 * IQR_pressurelow

upper_pressurelow <- pressurelow_bounds[2] + 1.5 * IQR_pressurelow

Dataset <- Dataset[Dataset$pressurelow >= lower_pressurelow & Dataset$pressurelow <=
upper_pressurelow, ]

plot(Dataset$pressurelow, col=5)
```

# Outputs:


Dataset$age
20
40
60
80
0
20
40
60
80
100
120
Index


Dataset$pressurelow
50
60
70
80
90
0
20
40
60
80
100
120
Index


Dataset$pressurehight
100
140
0
20
40
60
80
100
120
Index


Dataset$impluse
60
70
80
90
0
20
40
60
80
100
120
Index

# Discussion & Conclusion:

The given dataset was very messy. Moreover, there was a combination of categorical and numerical value. Moreover, there is present of outliers in the dataset. The dataset was like this-

Mid_Project.R* × Dataset ×

Filter

| | age | gender | impluse | pressurehight | pressurelow | glucose | class |
|---|---|---|---|---|---|---|---|
| 8 | 63 | male | 60 | 214 | 82 | 87 | positive |
| 9 | 44 | female | 60 | NA | 81 | 135 | negative |
| 10 | 67 | | 61 | 160 | 95 | 100 | negative |
| 11 | NA | female | 60 | 166 | 90 | 102 | negative |
| 12 | 63 | female | 60 | 150 | 10 | 198 | negative |
| 13 | 64 | male | 60 | 199 | 5 | 92 | positive |
| 14 | 54 | female | 94 | 122 | 67 | 97 | negative |
| 15 | 47 | male | 76 | 120 | 70 | 319 | negative |
| 16 | 61 | male | 81 | NA | 66 | 134 | positive |
| 17 | 86 | female | 73 | 114 | 68 | 87 | positive |
| 18 | 45 | female | 70 | 100 | 68 | 96 | negative |
| 19 | 37 | female | 72 | 107 | 86 | 274 | negative |
| 20 | 45 | male | 60 | 109 | 65 | 89 | positive |
| 21 | 60 | male | 92 | 151 | 78 | 301 | negative |
| 22 | 48 | male | 135 | 98 | 60 | 100 | positive |
| 23 | 52 | male | 76 | 109 | 85 | 227 | positive |
| 24 | 30 | male | 63 | 110 | 68 | 107 | positive |
| 25 | NA | male | 63 | 320 | 63 | 269 | positive |
| 26 | 72 | male | 64 | 106 | 68 | 111 | positive |

Showing 7 to 26 of 150 entries, 7 total columns

After Applying data preparation steps and the univariate data exploration for the given data set., we got the dataset looks like this-

Console Terminal Background Jobs

R 4.3.1 · ~/

```
> print(Dataset)
    age gender impluse pressurehight pressurelow glucose    class
1    64   male      66           160          83     160 negative
2    21   male      94            98          46     296 positive
4    64   male      70           120          55     270 positive
5    55   male      64           112          65     300 negative
6    58 female      61           112          58      87 negative
9    44 female      60           127          81     135 negative
10   67   male      61           160          95     100 negative
11   56 female      60           166          90     102 negative
14   54 female      94           122          67      97 negative
15   47   male      76           120          70     319 negative
16   61   male      81           127          66     134 positive
17   86 female      73           114          68      87 positive
18   45 female      70           100          68      96 negative
19   37 female      72           107          86     274 negative
20   45   male      60           109          65      89 positive
21   60   male      92           151          78     301 negative
23   52   male      76           109          85     227 positive
24   30   male      63           110          68     107 positive
26   72   male      64           106          68     111 positive
27   42   male      65           150          68     101 negative
29   47 female      66           134          57     279 positive
30   63   male      66           135          55     166 negative
32   35   male      62           137          61     321 negative
33   68   male      61           121          49      98 positive
34   56 female      60           145          62     105 negative
35   50   male      61           136          70     136 positive
36   64   male      58           156          76      82 positive
37   56   male      60           166          82     117 negative
38   64   male      65           155          75     107 negative
39   50   male      93           120          71     120 negative
40   34   male      96           105          75     136 positive
41   44   male      94            91          52     208 negative
```

Now, we can use this clean, pre-processed dataset for further use.