**Introduction to Data Science**

**Midterm Project**

**Data Set Description:** The size of the dataset is 151 samples, which have seven fields, where six fields are for input fields and one field for an output field. Age, gender**(0 for Female, 1 for Male)** ,heart rate (impulse), systolic BP (pressurehight), diastolic BP (pressurelow), blood sugar(glucose) are representing the input fields, while the output field pertains to the presence of heart attack (class), which is divided into two categories (negative and positive); negative refers to the absence of a heart attack, while positive refers to the presence of a heart attack.

A table of numbers on a white background

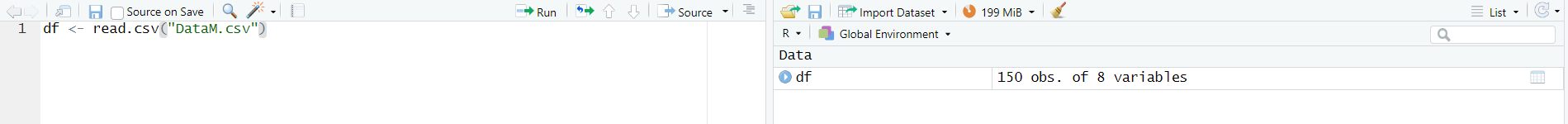
Description automatically generated

**Preparation Steps:** firstly, we go through the data and find out that the missing values were replaced with blank space, so I must delete the blank space manually from excel.

**Read Data:** Initially we started to read the data which is in the CSV file. Read the files we used read.CSV by default functions to read the file. Here we put the data file and project file in the same directory. We can also use path variable to do this task.

df <- read.csv("Data.csv")

**Code And Output**



**Exploring Data:** Here we use head (), tail () and summary () names(df) str(df) to get some basic knowledge about this data where the data types are and if there any missing attributes available in this data. We can get such information from this execution. We can see here the maximum and the minimum value of each attribute and the null values.

head(df)

tail(df)

summary(df)

names(df)

str(df)

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

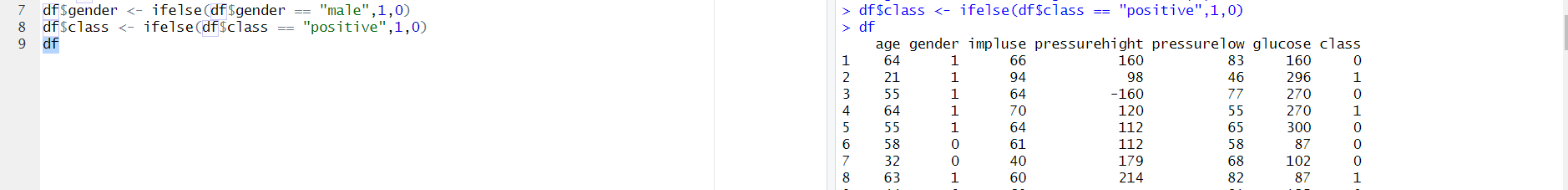
Description automatically generated**

**Replace the character Value into Binary value:** in here Gender and class attribute has the characterized value so we must make it in numerical order such as there are two different kinds of attributes so we can easily make it in binary order so here I take male as 1 and positive as 1. And female as 0 and negative as 0. Here we make just a simple equals condition to replace that,

df$gender <- ifelse(df$gender == "male",1,0)

df$class <- ifelse(df$gender == "positive",1,0)

df

****

**Fixing the Null value:** Here firstly we see the summary that is there any null value out there after that we've seen that we have null value in age and pressure high. So, we go with the mean replacement approach where we calculate the mean of this attribute and replace the missing values with the mean value. And for pressure high we use the maximum valued attribute replaced by the null value. Here is the result,

mean (df$age, na.rm =TRUE)

df$age <- as.integer(replace(df$age, is.na(df$age), mean(df$age, na.rm = TRUE)))

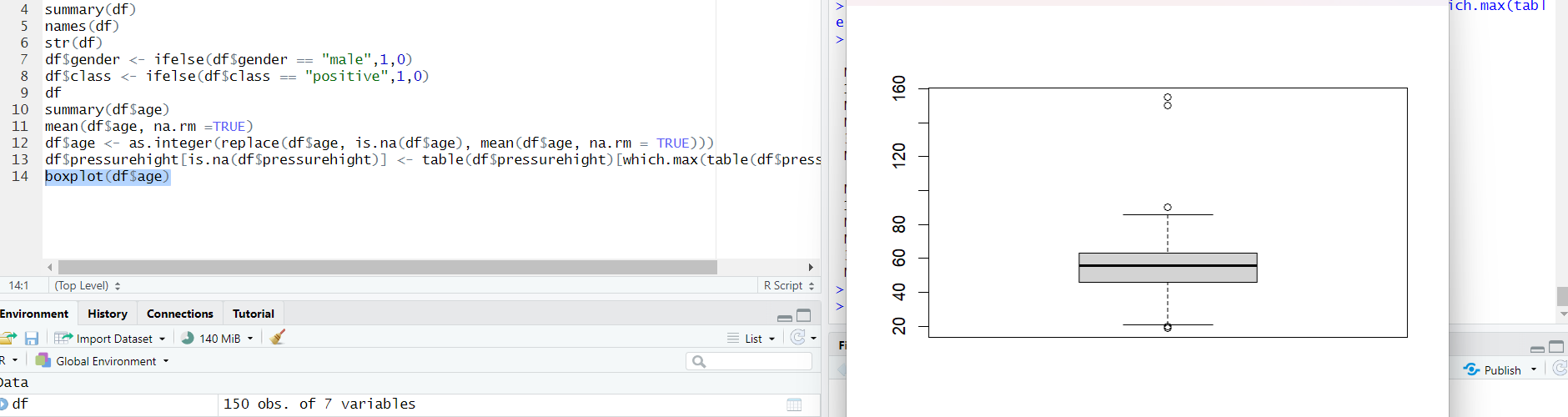
df$pressurehight[is.na(df$pressurehight)] <- table(df$pressurehight)[which.max(table(df$pressurehight))]

A screenshot of a computer

Description automatically generated

**Checking Outlier for Age:** For this we must make a box plot because, box plot describes the outliers better than anything here we can see there are 3 outliers into age, so we must remove it or must scale it.

**boxplot(df$age)**

****

**Removing Outlier for Age:** Firstly, we see that is there any value greater than one 100. After pointing that out we directly go for removing them because they are the outlier which will be effective to the main result

A screenshot of a computer

Description automatically generated

The expression scale(df$age) will standardize the values in the age column of the data frame df. The expression which(scale(df$age)<100) will then return the indices of the rows where the standardized values are less than 100.

A screenshot of a computer

Description automatically generated

which(df$age>100)

which(scale(df$age)<100)

which(colnames(df)=="age")

df <- df[-138, ]

df <- df[-147, ]

**Fixing The noise value for Impulse:** in here there is some impulse value which is around 60 to 110 but there is one single value which has 1111 so I think that is a mistaken when the data was taken. So, I replaced the value with 111.

**df$impluse[df$impluse == 1111] <- 111**

**A screenshot of a computer

Description automatically generated**

**Removing Outlier for Impulse:** After plotting a box plot for impulse we can see there were 2 outliers, so we had to remove it and uh we did it right away.

boxplot(df$impluse,df$pressurehight,df$pressurelow,df$glucose)

which(df$impluse>120)

df <- df[-22, ]

which(colnames(df)=="impluse")

df <- df[-c(31, 126), ]

summary(df$impluse)

df$impluse[df$impluse == 1111] <- 111

**A screenshot of a computer

Description automatically generated**

**Fixing The noise value for pressurehight:** when we are performing the summary query, we had seen that there were some negative values in the data set I thought uh that was a noisy value, so we had to replace that negative value to positive value.

**summary(df$pressurehight)**

**any (df$pressurehight < 0)**

**which(df$pressurehight< 0)**

**df$pressurehight[df$pressurehight == -160] <- 160**

**Removing Outlier for pressurehight:** waiting there are some values which is over 160 so I think that was an outlier and that should be replaced by something else otherwise the data set will be affected very badly. So that's why I replace those values with 160 which is considered high.

**which(df$pressurehight< 0)**

**df$pressurehight[df$pressurehight == -160] <- 160**

**which(df$pressurehight>160)**

**df$pressurehight[df$pressurehight >= 160] <- 160**

A screenshot of a computer

Description automatically generated

**Removing Outlier for pressurehight:** There's some value which is very low to other values in the dataset, so I replaced with that value to 40. And that solves the outlier issue.

summary(df$pressurelow)

which(df$pressurelow< 40)

df$pressurelow[df$pressurelow < 40] <- 40

**A screenshot of a computer

Description automatically generated**

**Removing Outlier for glucose:** There were too many outliers in attribute of glucose. So, for a lot of them we can't remove them that will affect our data set as well So what we did we did replace the value which is greater than 200 with the mean value of glucose which solved the outlier issue after that we converted in integer and now there is no outlier.

A screenshot of a computer

Description automatically generated

**Plotting A complete box plot:**

**boxplot(df$age,df$impluse,df$pressurehight,df$pressurelow,df$glucose,main = "Boxplot of Patient Data",xlab = "Variable",ylab = "Value" )A screenshot of a computer

Description automatically generated**

**Plotting A Histogram for Age:**

**ggplot(df, aes(x = age)) +**

**geom\_histogram(binwidth = 5, fill = "blue", color = "black") +**

**labs(title = "Age Histogram", x = "Age", y = "Frequency")**

**A screenshot of a computer

Description automatically generated**

**Plotting A Histogram for impulse:**

**ggplot(df, aes(x =impluse )) +**

**geom\_histogram(binwidth = 5, fill = "red", color = "black") +**

**labs(title = "impluse Histogram", x = "impluse", y = "Frequency")A screenshot of a computer

Description automatically generated**

**Plotting A Histogram for Glucose:**

**ggplot(df, aes(x = glucose)) +**

**geom\_histogram(binwidth = 20, fill = "orange", color = "black") +**

**labs(title = "Glucose Histogram", x = "Glucose", y = "Frequency")**

**A screenshot of a computer

Description automatically generated**

**Plotting A Histogram for pressurehight:**

**ggplot(df, aes(x = pressurehight)) +**

**geom\_histogram(binwidth = 10, fill = "green", color = "black") +**

**labs(title = "Pressure High Histogram", x = "Pressure High", y = "Frequency")**

**A screenshot of a computer

Description automatically generated**

**Plotting A Histogram for pressurelow:**

**ggplot(df, aes(x = pressurelow)) +**

**geom\_histogram(binwidth = 10, fill = "purple", color = "black") +**

**labs(title = "Pressure Low Histogram", x = "Pressure Low", y = "Frequency")**

**A screenshot of a computer

Description automatically generated**

**Plotting A Bar Graph for Class :**

**A screenshot of a computer

Description automatically generated**