

American International University-Bangladesh (AIUB)  
Department of Computer Science  
Faculty of Science &Technology (FST)  
FALL 23-24

Section: C  
INTRODUCTION TO DATA SCIENCE

**REPORT SUBMITTED BY**

|  |  |  |
| --- | --- | --- |
| SN | Student Name | Student ID |
| 1 | RIAZ SARKER | 20-42362-1 |
| 2 | MD. RAFID HASAN | 20-42421-1 |

**SUBMITTED TO**

**TOHEDUL ISLAM**

Assistant Professor, Faculty, Department of Computer Science

# Description:

The heart\_attack\_data dataset is widely recognized for its information on patient. According to the provided information, the medical dataset classifies either heart attack or none. It is frequently employed as a benchmark in data analysis and machine learning tasks. This dataset offers valuable insights into the demographics and characteristics of the heart attack patient, as well as their heart condition. By studying this dataset, researchers and data scientists can explore various factors that potentially influenced heart Disease rates on patient, including Age, Sex, ChestPain Type, RestingBp, Cholesterol, FastingBS, RestingECG, MaxHr etc.

# Attributes:

Age: Age of the patient

Sex: Sex of the patient

ChestPain Type: There is four type of ChestPain Type of patient (Typical Angina, Atypical Angina, Non-Anginal Pain, Asymptomatic)

RestingBP: Resting blood pressure (in mm Hg) of Patient.

Cholesterol: Cholesterol in mg/dl fetched via BMI sensor

FastingBS: Fasting blood sugar > (120 mg/dl)

RestingECG: Resting Electrocardiographic results of Patient

MaxHR: Maximum heart rate achieved of Patient.

ExerciseAngina: Exercise induced angina on heart attack Patient.

Oldpeak: ST depression induced by exercise relative to rest.

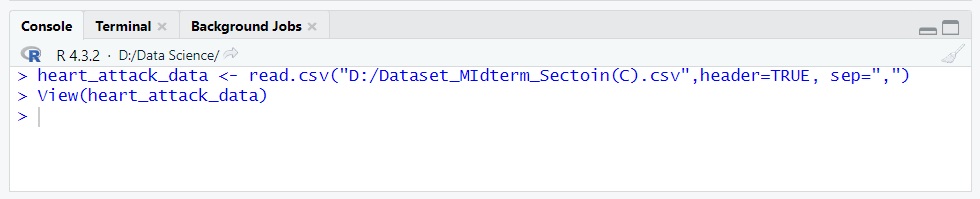
ST\_Slope: The slope of the peak exercise ST segment

HeartDisease: Diagnosis of heart disease (angiographic disease status).

# Import Dataset:

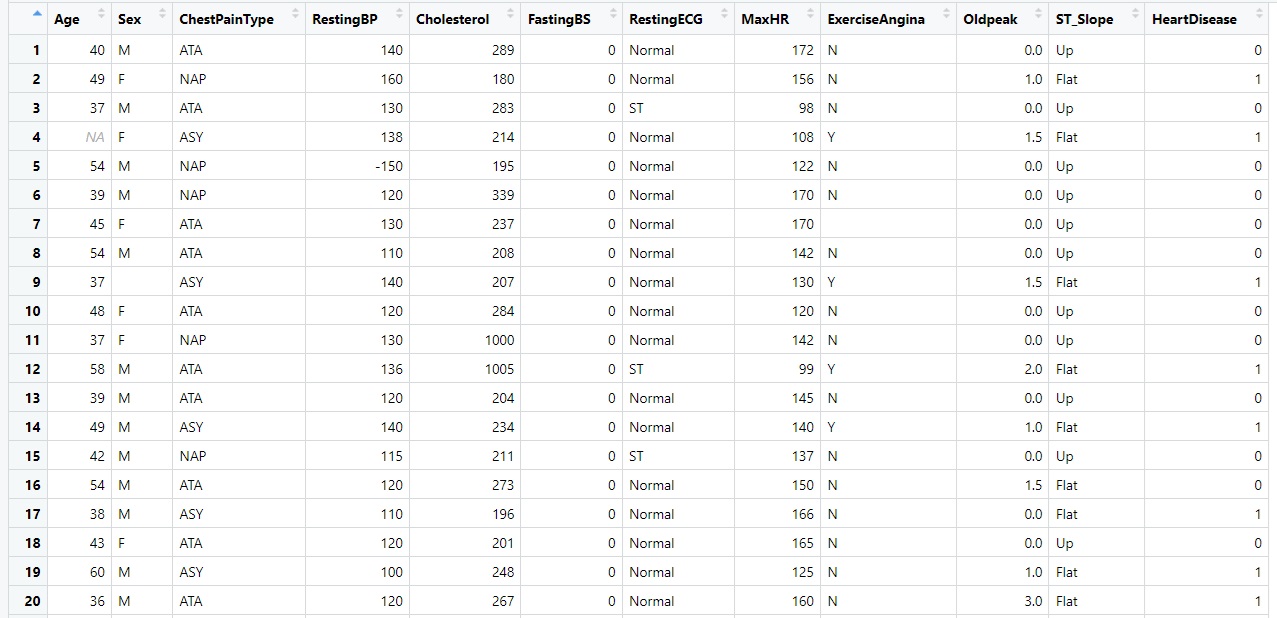
heart\_attack\_data <- read.csv("D:/Dataset\_MIdterm\_Sectoin(C).csv",header=TRUE, sep=",")

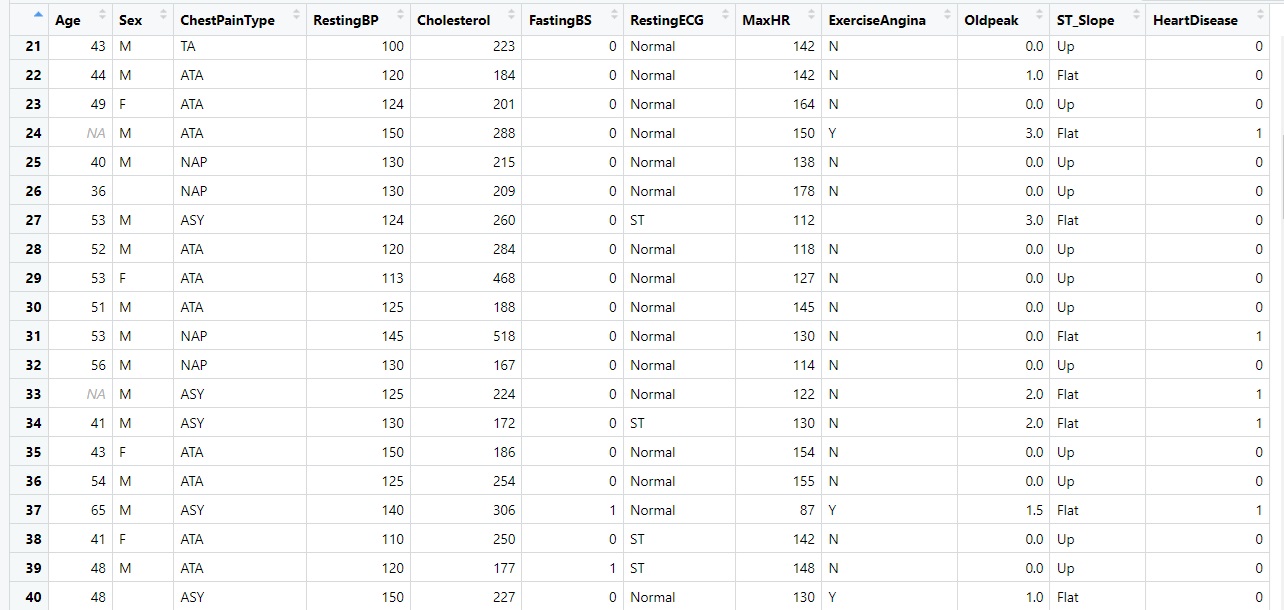
View(heart\_attack\_data)

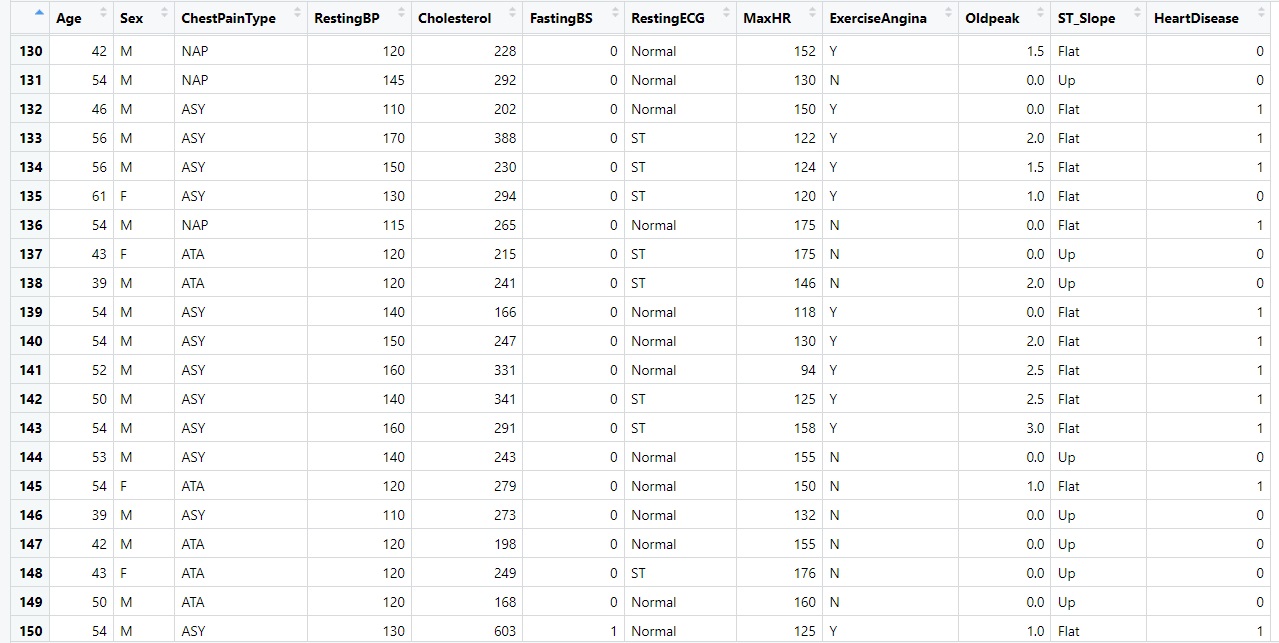


**Explanation:** Imports the heart\_attack\_data dataset from a designated file path, storing it in the variable "heart\_attack\_data". The dataset is read as a CSV file, with headers included and values separated by patient informetion. To facilitate exploration and analysis, executing "View(heart\_attack\_data)" opens a separate viewer window, displaying the imported dataset.

**DISPLAYING THE DATA\_SET (heart\_attack\_data):**







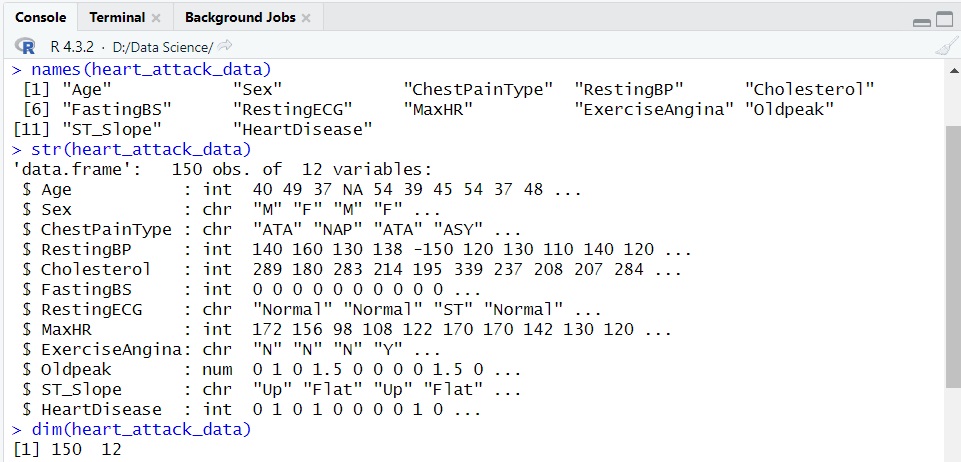
# Visualizing the Dataset:

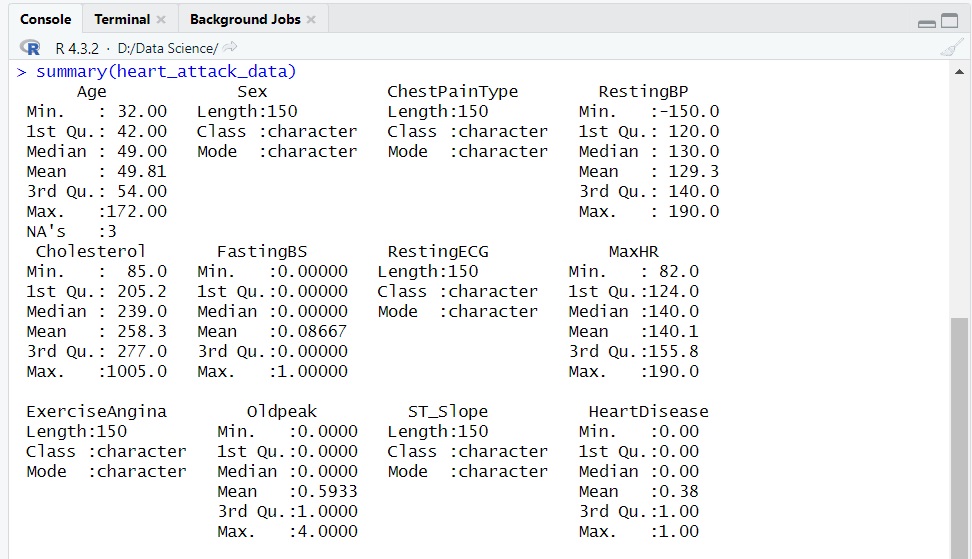
names(heart\_attack\_data)

str(heart\_attack\_data)

dim(heart\_attack\_data)

summary(heart\_attack\_data)

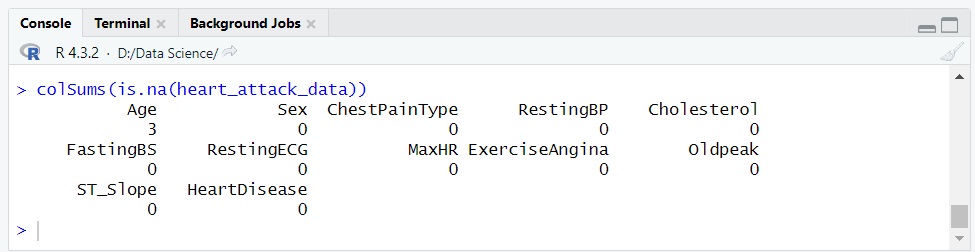




**Explanation:** Visualizing the dataset involves exploring its variables and structure. The command "names(heart\_attack\_data)" retrieves the column names, providing an overview of the available data. "str(heart\_attack\_data)" displays the structure, data types, and summaries of the variables. Additionally, "dim(heart\_attack\_data)" reveals the dataset dimensions, giving insights into its size. Lastly, "summary(heart\_attack\_data)" provides statistical summaries, aiding in understanding the numerical aspects of the dataset. These visualization techniques contribute to comprehending the dataset and supporting data analysis and decision-making processes.

**Find Missing Values:**

colSums(is.na(heart\_attack\_data))

****

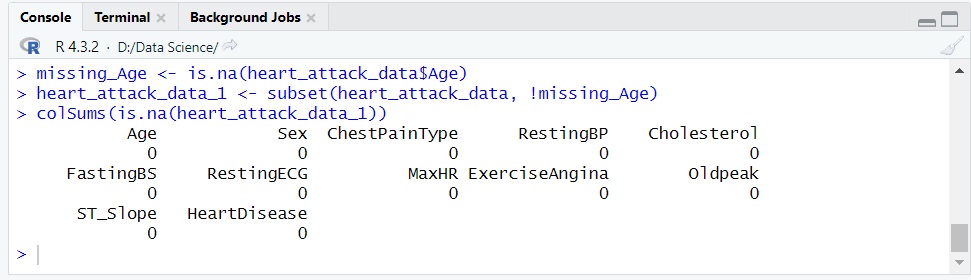
**Explanation:** Identifying missing values in the heart\_attack\_data dataset involves assessing the presence of null or missing data points. This process allows researchers to determine the extent of missingness across different variables. By analyzing the column-wise sums of missing values, insights can be gained into the distribution and quantity of missing data, aiding in subsequent data cleaning and imputation strategies.

**Remove Missing Values (Age):**

missing\_Age <- is.na(heart\_attack\_data$Age)

heart\_attack\_data\_1 <- subset(heart\_attack\_data, !missing\_Age)

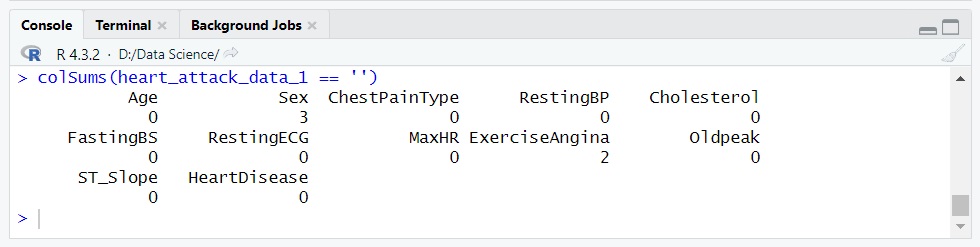
colSums(is.na(heart\_attack\_data\_1))



**Explanation:** To address missing values (Age) in the heart\_attack\_data dataset, a process of removal can be performed. By creating a logical variable, "missing\_Age," to identify the missing values, the dataset can be subsetted to exclude these cases. The resulting dataset, "heart\_attack\_data\_1," will no longer contain missing Age values. By subsequently checking for missing values using "colSums(is.na(heart\_attack\_data\_1))," researchers can confirm that the Age variable is now free of missing data.

**Find Empty Data:**

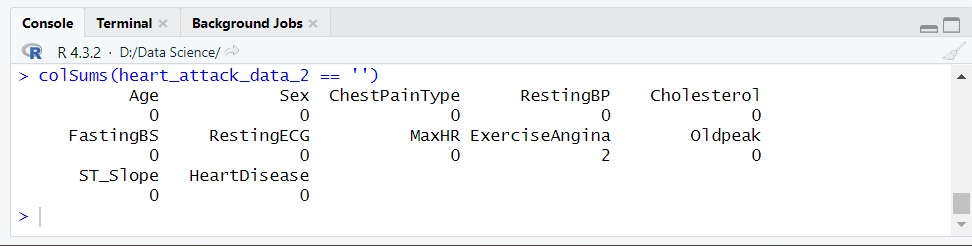
colSums(heart\_attack\_data\_1 == '')



**Remove Empty Data (Sex):**

heart\_attack\_data\_2 <- heart\_attack\_data\_1[!heart\_attack\_data\_1$Sex == "", ]

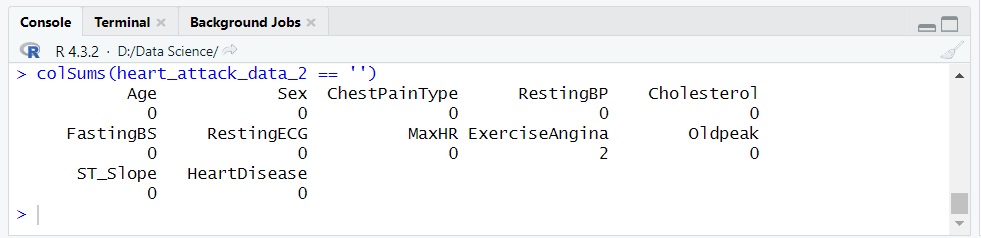
colSums(heart\_attack\_data\_2 == '')



**Explanation:** To remove empty or blank values from the heart\_attack\_data dataset, an assessment is conducted to determine the presence of empty data using column-wise calculations. After identifying the empty values, the dataset is filtered to exclude rows where specific variables have empty entries. By reevaluating the dataset, it can be confirmed that the empty data has been successfully removed.

**Again Find Empty Data:**

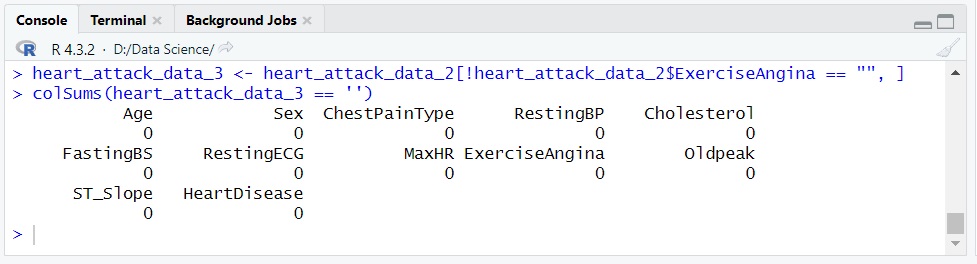
colSums(heart\_attack\_data\_2 == '')



**Remove Empty Data (ExerciseAngina):**

heart\_attack\_data\_3 <- heart\_attack\_data\_2[!heart\_attack\_data\_2$ExerciseAngina == "", ]

colSums(heart\_attack\_data\_3 == '')



**Explanation:** To remove empty or blank values from the heart\_attack\_data dataset, an assessment is conducted to determine the presence of empty data using column-wise calculations. After identifying the empty values, the dataset is filtered to exclude rows where specific variables have empty entries. By reevaluating the dataset, it can be confirmed that the empty data has been successfully removed.

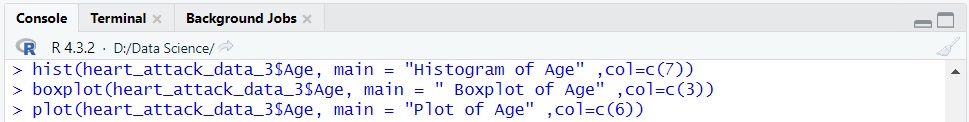
**Find Outlier in Age attribute:**

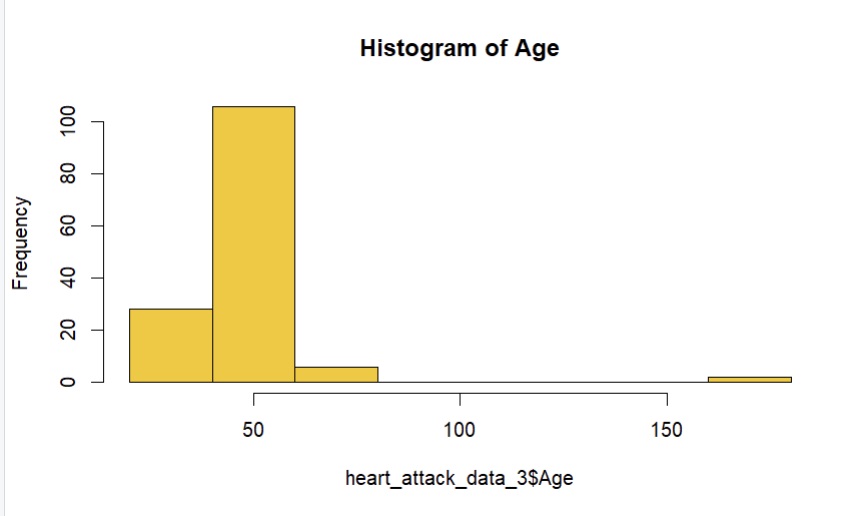
hist(heart\_attack\_data\_3$Age, main = "Histogram of Age" ,col=c(7))

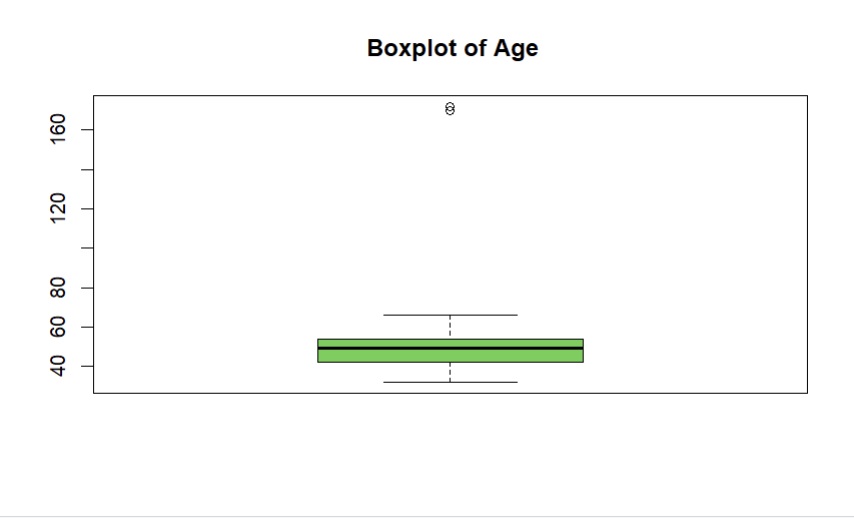
boxplot(heart\_attack\_data\_3$Age, main = " Boxplot of Age" ,col=c(3))

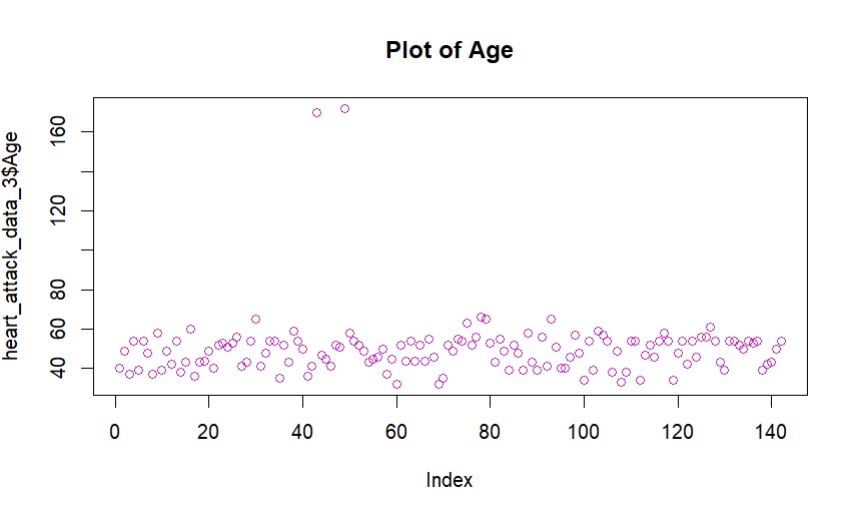
plot(heart\_attack\_data\_3$Age, main = "Plot of Age" ,col=c(6))

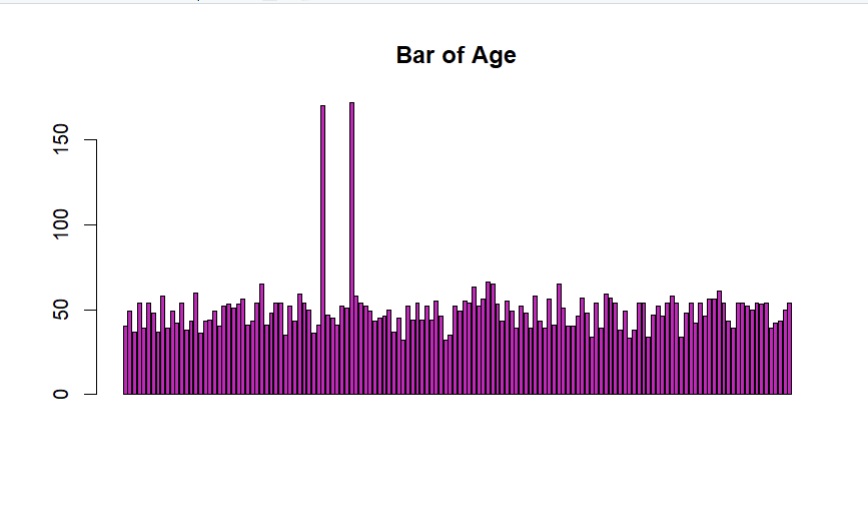
barplot(heart\_attack\_data\_3$Age, main = "Bar of Age" ,col=c(6))



****

****

****

****

**Explanation:** To identify outliers in the Age attribute of the heart\_attack\_data dataset, various visualization techniques can be employed. A histogram provides an overview of the Age distribution, highlighting any unusual values. A box plot visually represents the range and distribution of Ages, making it easier to identify outliers. Scatter plots can also be utilized to examine individual data points and detect any extreme values that deviate significantly from the overall pattern. These visualization approaches help in identifying and understanding outliers within the Age attribute.

**Remove Outliers in Age attribute:**

Age\_Q1 <- quantile(heart\_attack\_data\_3$Age, 0.25, na.rm = TRUE)

Age\_Q3 <- quantile(heart\_attack\_data\_3$Age, 0.75, na.rm = TRUE)

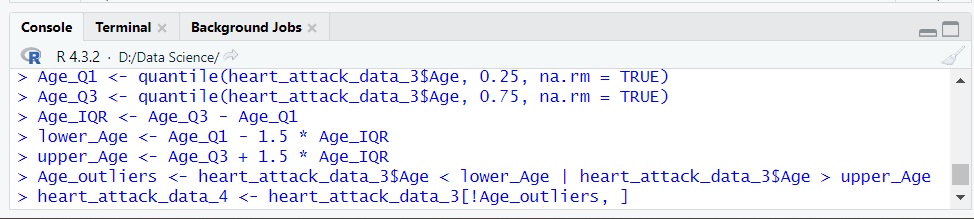
Age\_IQR <- Age\_Q3 - Age\_Q1

lower\_Age <- Age\_Q1 - 1.5 \* Age\_IQR

upper\_Age <- Age\_Q3 + 1.5 \* Age\_IQR

Age\_outliers <- heart\_attack\_data\_3$Age < lower\_Age | heart\_attack\_data\_3$Age > upper\_Age

heart\_attack\_data\_4 <- heart\_attack\_data\_3[!Age\_outliers, ]



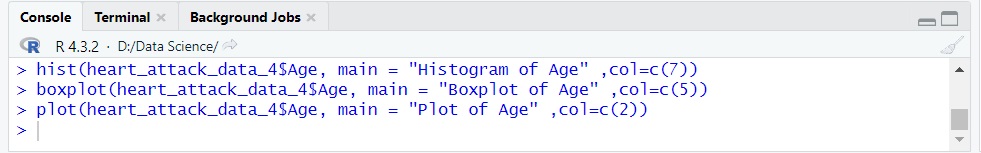
**For Graph (Age):**

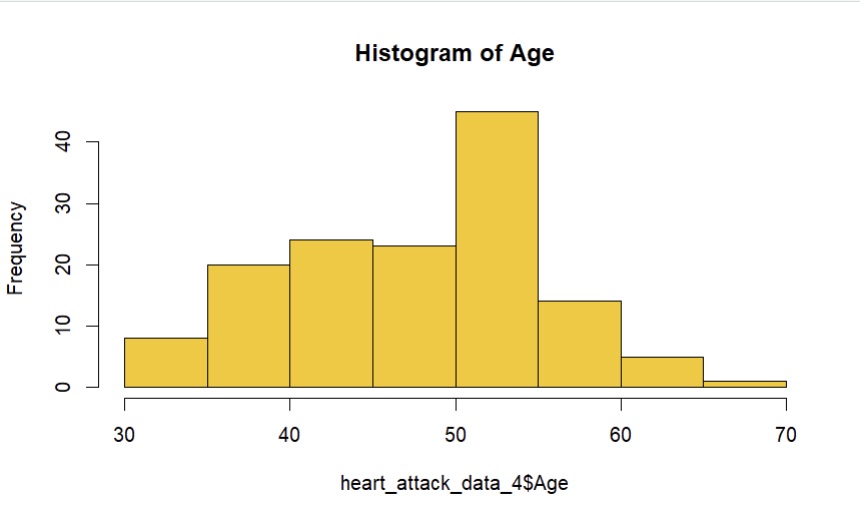
hist(heart\_attack\_data\_4$Age, main = "Histogram of Age" ,col=c(7))

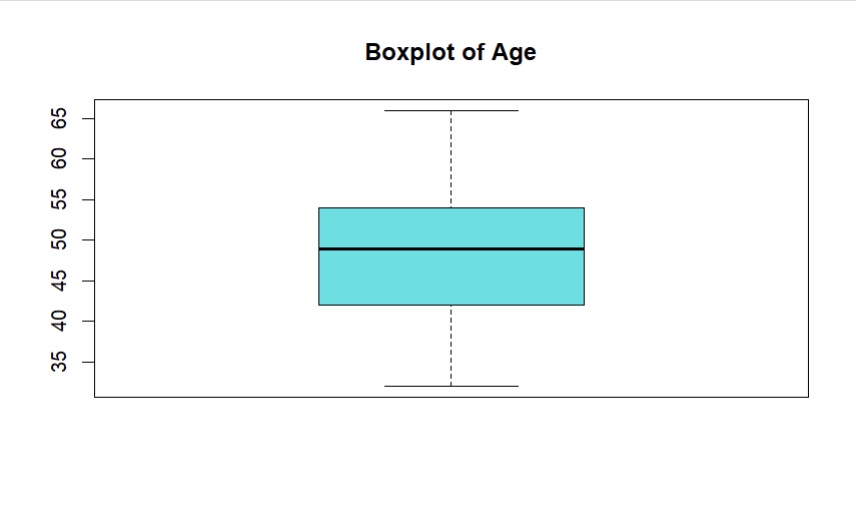
boxplot(heart\_attack\_data\_4$Age, main = "Boxplot of Age" ,col=c(5))

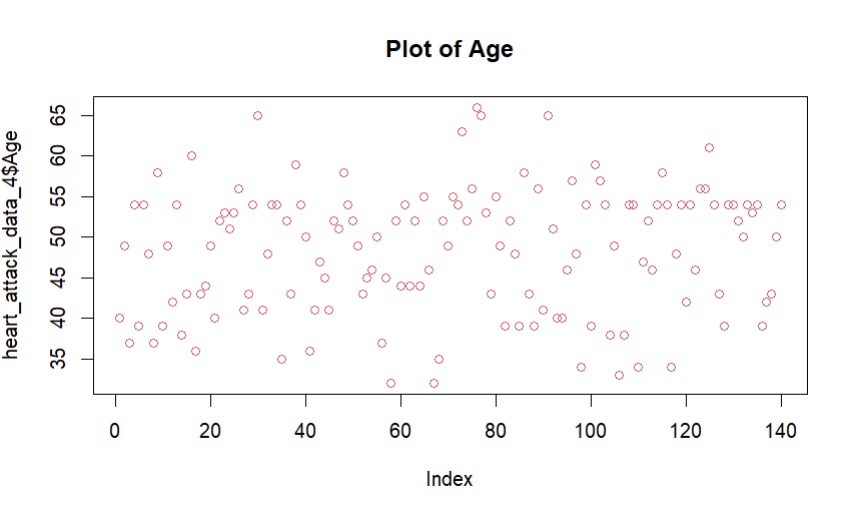
plot(heart\_attack\_data\_4$Age, main = "Plot of Age" ,col=c(3))

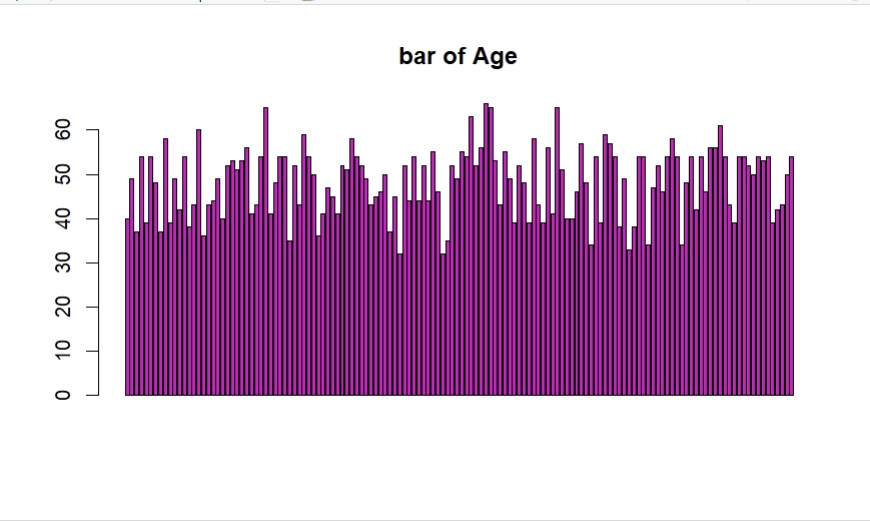
barplot(heart\_attack\_data\_4$Age, main = "bar of Age" ,col=c(6))











**Explanation:** To remove outliers in Age attribute from the heart\_attack\_data dataset using Tukey's fences method. The lower quartile (Q1) and upper quartile (Q3) are calculated, and the interquartile range (IQR) is determined. The lower and upper fences are established by subtracting and adding 1.5 times the IQR, respectively. Age values falling outside these fences are identified as outliers and filtered out, resulting in a new dataset, heart\_attack\_data\_4, without outliers in Age. The resulting Age distribution can be visualized using a Histogram, Boxplot, Plot and Barplot.

**Checking of Error Values:**

attribute\_names <- names(heart\_attack\_data\_4)

for (attribute in attribute\_names)

{

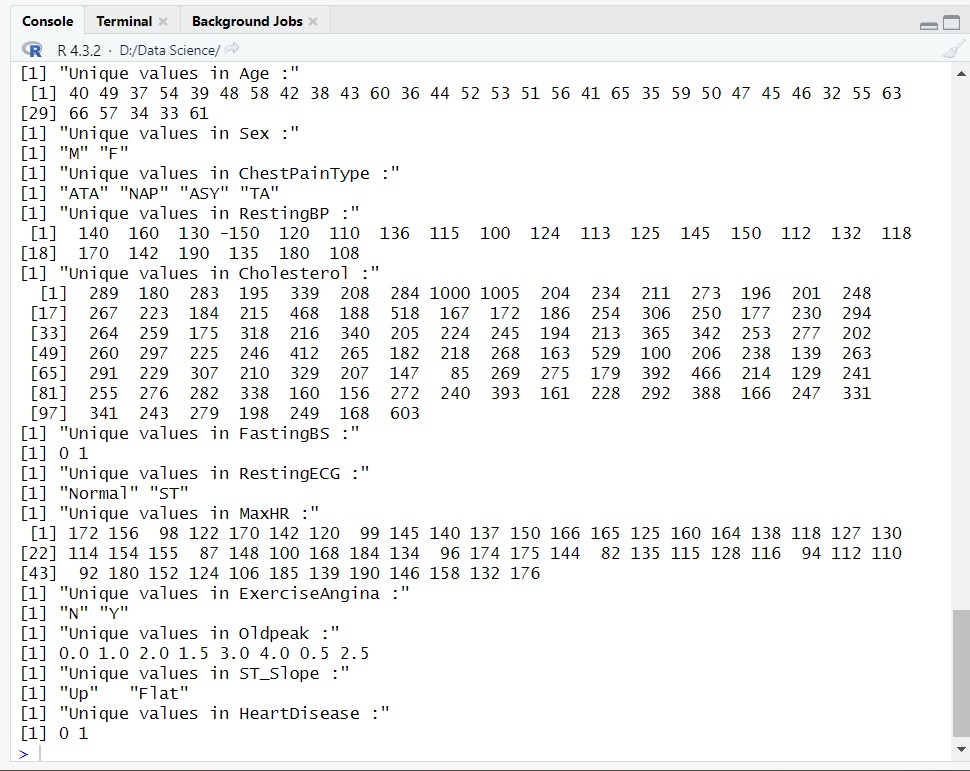
unique\_values <- unique(heart\_attack\_data\_4[[attribute]])

print(paste("Unique values in", attribute, ":"))

print(unique\_values)

}





**Explanation:** During the analysis of the heart\_attack\_data dataset, a rigorous assessment was performed to identify and rectify error values. This involved iterating through each attribute and extracting unique values to detect anomalies. Among the attributes examined, special attention was given to the "Sex" attribute, where valid values were sought after verifying data accuracy. By meticulously addressing the short from of Male and Female and confirming the validity of attribute values, the dataset's integrity was ensured, laying the foundation for reliable analysis and interpretations.

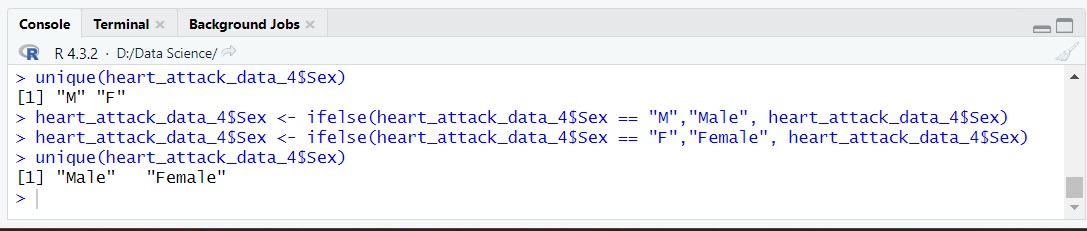
**Fixing the Value of Sex:**

unique(heart\_attack\_data\_4$Sex)

heart\_attack\_data\_4$Sex <- ifelse(heart\_attack\_data\_4$Sex == "M","Male", heart\_attack\_data\_4$Sex)

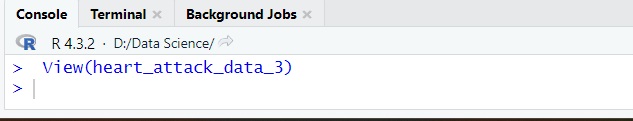
heart\_attack\_data\_4$Sex <- ifelse(heart\_attack\_data\_4$Sex == "F","Female", heart\_attack\_data\_4$Sex)

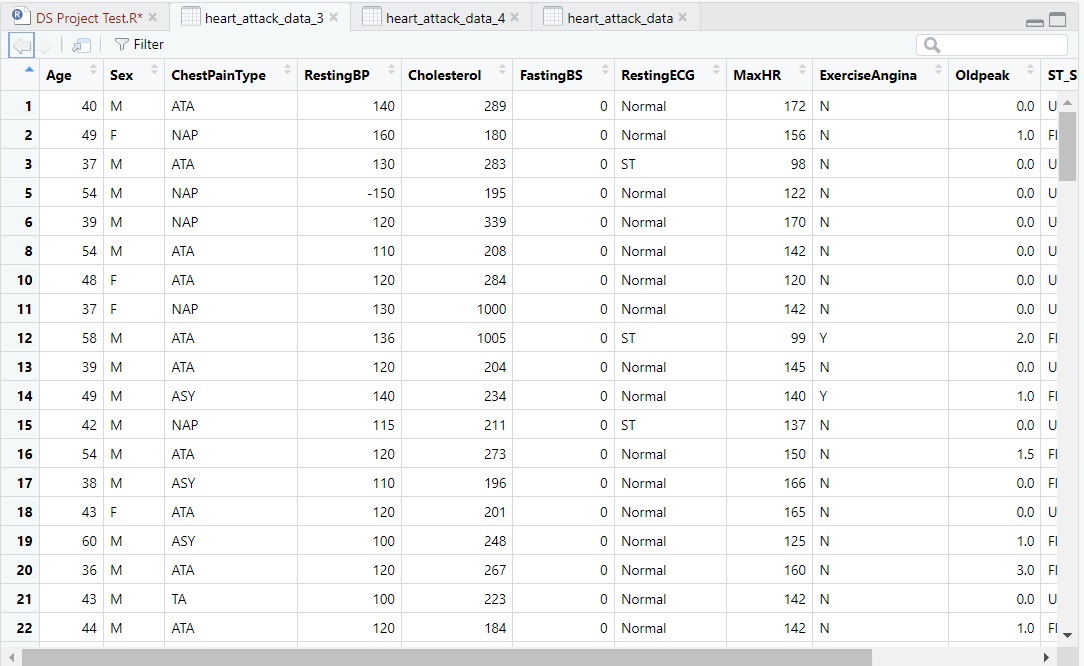
unique(heart\_attack\_data\_4$Sex)



**Previous Data (Sex):**

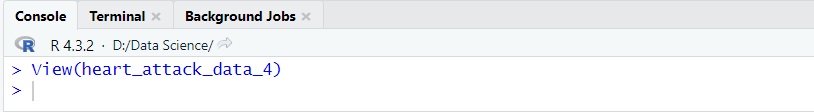
View(heart\_attack\_data\_3)

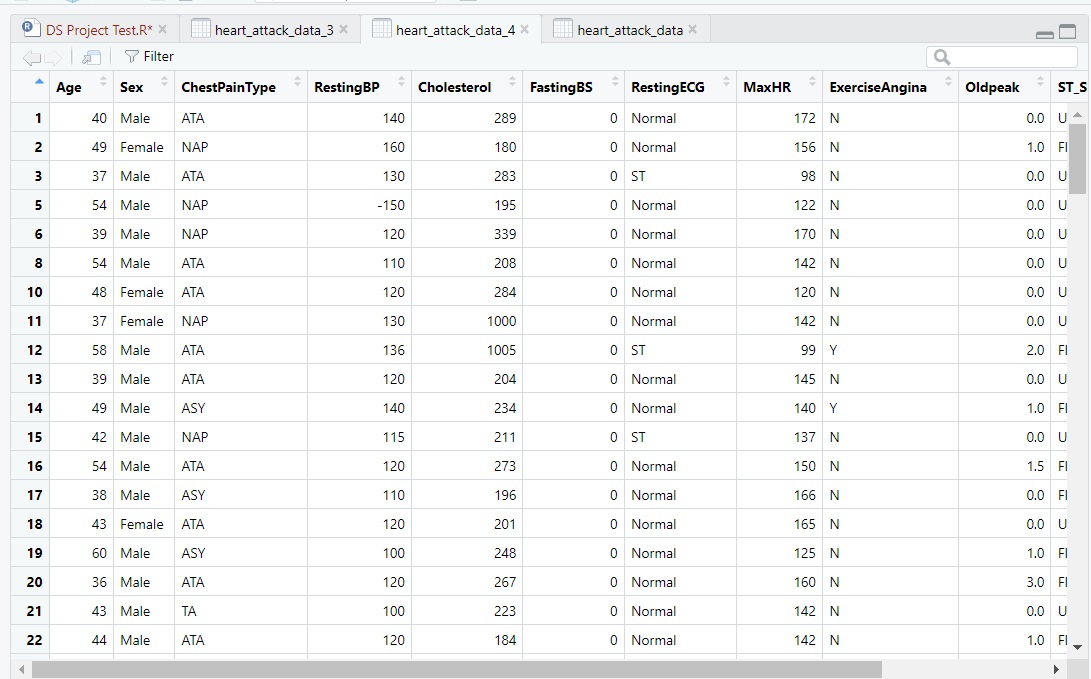




**Updated Data (Sex):**

View(heart\_attack\_data\_4)



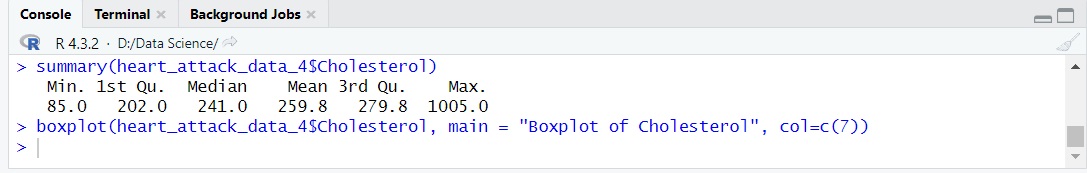
****

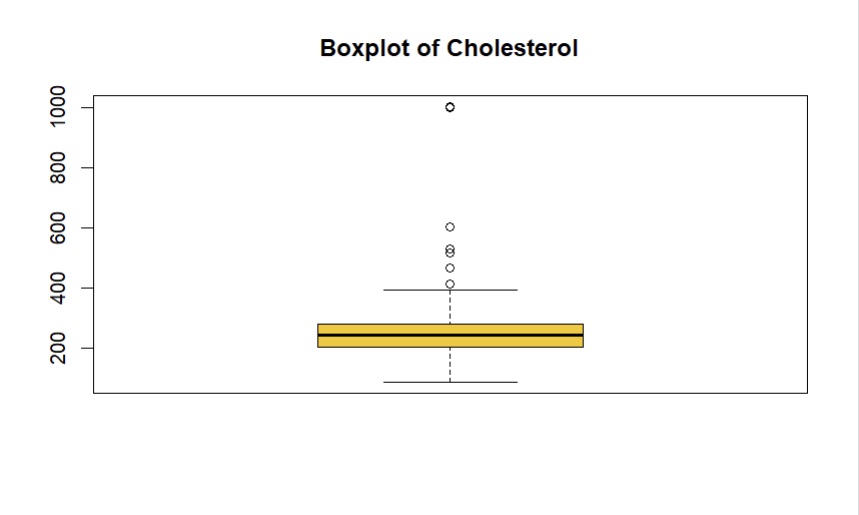
**Explanation:** To correct or write the full from of values in the "Sex" attribute of the heart\_attack\_data dataset, the unique values in this attribute were examined. Subsequently, erroneous entries such as "M" were replaced with the correct label "Male" using conditional statements. Similarly, "F" were corrected to "Female". By fixing these incorrect values, the dataset's integrity was restored. A final examination of the unique values confirmed that the corrections were successfully implemented.

**Detect Noisy Value (Cholesterol):**

summary(heart\_attack\_data\_4$Cholesterol)

boxplot(heart\_attack\_data\_4$Cholesterol, main = "Boxplot of Cholesterol", col=c(7))

****

**After resolve the noisy value (Cholesterol):**

Cholesterol\_Q1 <- quantile(heart\_attack\_data\_4$Cholesterol, 0.25, na.rm = TRUE)

Cholesterol\_Q3 <- quantile(heart\_attack\_data\_4$Cholesterol, 0.75, na.rm = TRUE)

Cholesterol\_IQR <- Cholesterol\_Q3 - Cholesterol\_Q1

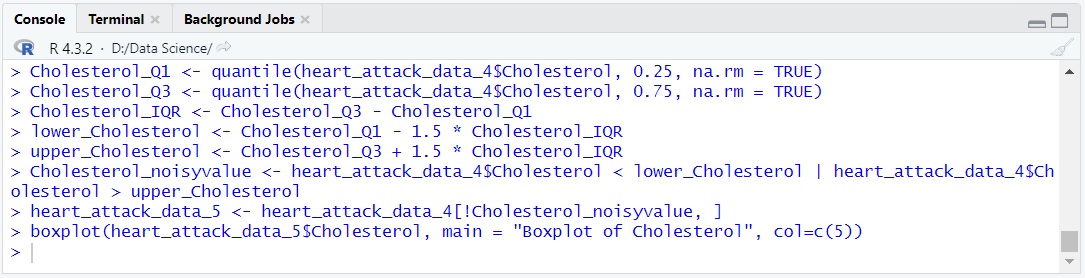
lower\_Cholesterol <- Cholesterol\_Q1 - 1.5 \* Cholesterol\_IQR

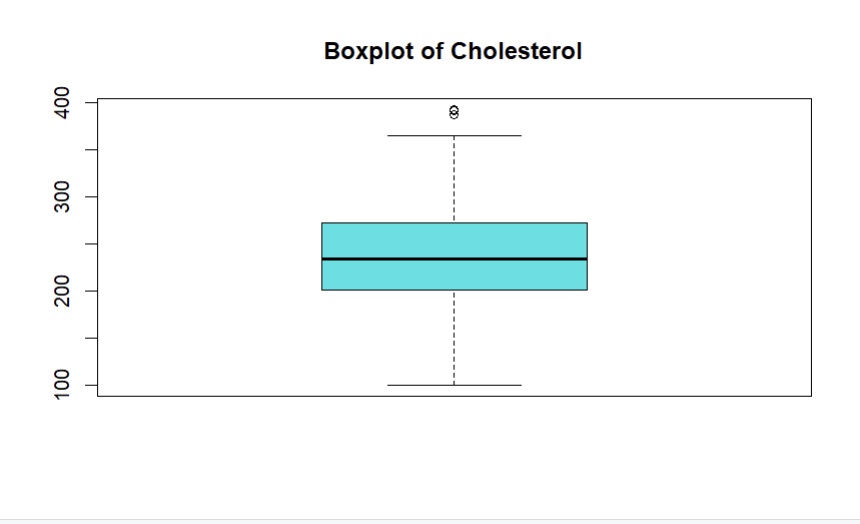
upper\_Cholesterol <- Cholesterol\_Q3 + 1.5 \* Cholesterol\_IQR

Cholesterol\_noisyvalue <- heart\_attack\_data\_4$Cholesterol < lower\_Cholesterol | heart\_attack\_data\_4$Cholesterol > upper\_Cholesterol

heart\_attack\_data\_5 <- heart\_attack\_data\_4[!Cholesterol\_noisyvalue, ]

boxplot(heart\_attack\_data\_5$Cholesterol, main = "Boxplot of Cholesterol", col=c(5))





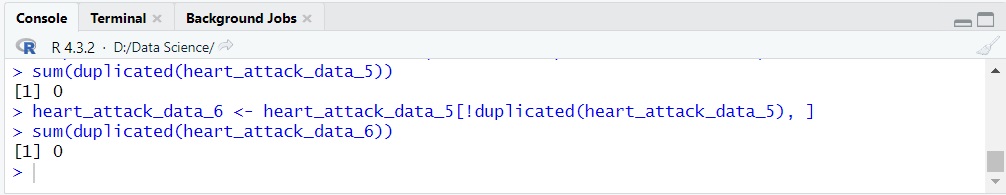
**Explanation:** To identify noisy values in the "Cholesterol" attribute of the heart\_attack\_data dataset, the summary statistics of Cholesterol were examined. This provided insights into the distribution and range of values. Additionally, a box plot was created to visualize the distribution, enabling the detection of potential outliers or extreme values. By analyzing these statistical measures and visual representations, noisy values within the "Cholesterol" attribute could be detected for further investigation and data quality assurance.

**Remove Duplicate Data:**

sum(duplicated(heart\_attack\_data\_5))

heart\_attack\_data\_6 <- heart\_attack\_data\_5[!duplicated(heart\_attack\_data\_5), ]

sum(duplicated(heart\_attack\_data\_6))



**Explanation:** To eliminate duplicate data from the heart\_attack\_data dataset, a process of removing identical observations based on all variables can be performed. This helps ensure that each row in the dataset is unique and prevents any redundant information from skewing the analysis or results. By eliminating duplicate data, researchers can work with a more accurate and reliable dataset for further exploration and analysis. But fortunately there is no Duplicate data on the heart\_attack\_data dataset.

**Conversion Data Type of Attributes:**

heart\_attack\_data\_6$RestingECG <- factor(heart\_attack\_data\_6$RestingECG,levels=c('Normal','ST'),labels=c("NORMAL","ABNORMALITY"))

heart\_attack\_data\_6$Sex <- factor(heart\_attack\_data\_6$Sex,levels=c('Male','Female'),labels=c("MALE","FEMALE"))

heart\_attack\_data\_6$ExerciseAngina <- factor(heart\_attack\_data\_6$ExerciseAngina,levels=c('Y','N'),labels=c("YES","NO"))

unique(heart\_attack\_data\_5$ChestPainType)

heart\_attack\_data\_6$ChestPainType <- factor(heart\_attack\_data\_6$ChestPainType,levels=c('ATA','NAP','ASY','TA'),labels=c("ATYPICAL ANGINA","NON-ANGINAL PAIN","ASYMPTONIC

","TYPICAL ANGINA"))

heart\_attack\_data\_6$HeartDisease <- factor(heart\_attack\_data\_6$HeartDisease,levels=c(1,0),labels=c("MORE CHANCE","LESS CHANCE"))

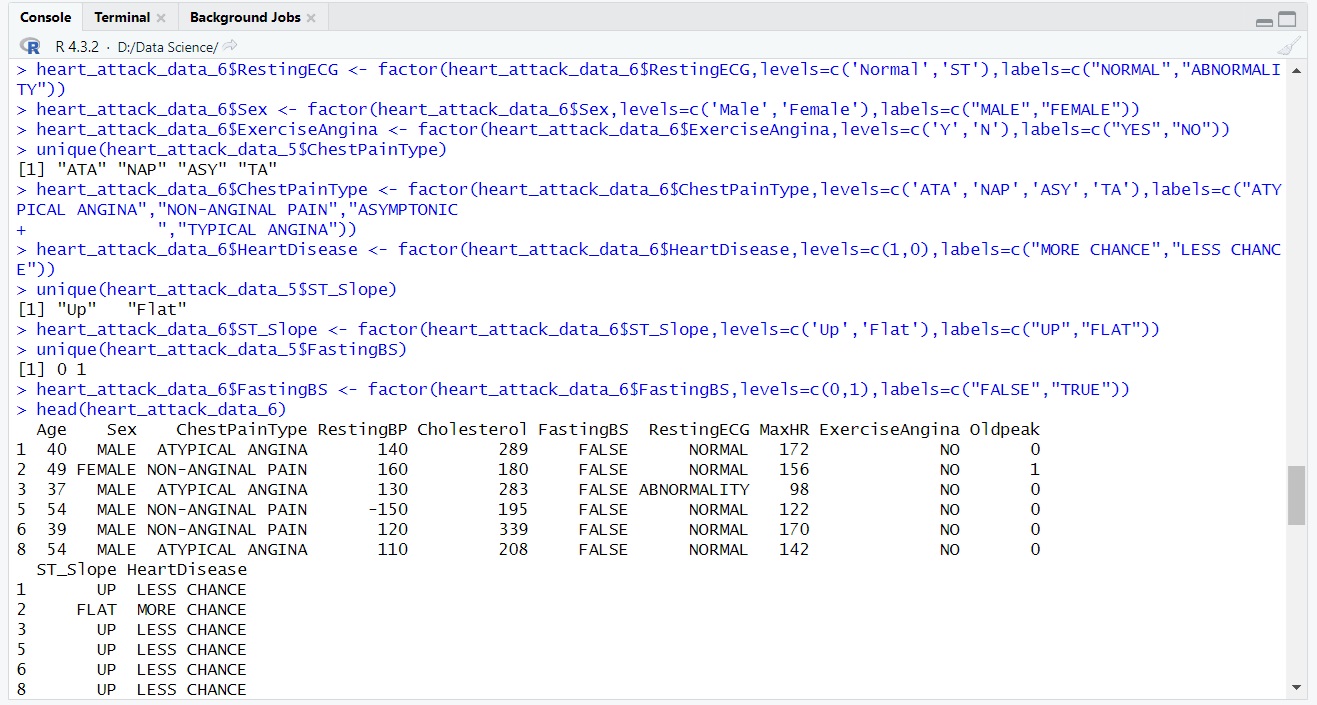
unique(heart\_attack\_data\_5$ST\_Slope)

heart\_attack\_data\_6$ST\_Slope <- factor(heart\_attack\_data\_6$ST\_Slope,levels=c('Up','Flat'),labels=c("UP","FLAT"))

unique(heart\_attack\_data\_5$FastingBS)

heart\_attack\_data\_6$FastingBS <- factor(heart\_attack\_data\_6$FastingBS,levels=c(0,1),labels=c("FALSE","TRUE"))

head(heart\_attack\_data\_6)



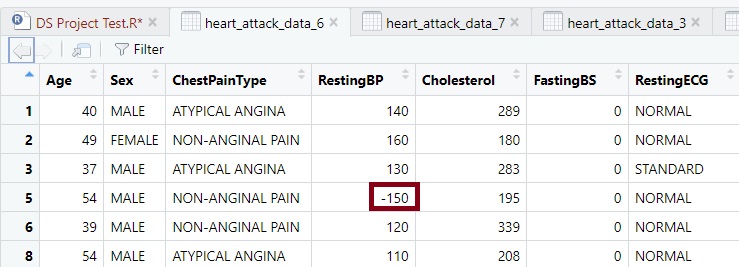
**Explanation:** To ensure appropriate data representation, several attribute data types were converted in the heart\_attack\_data dataset. The "Sex" attribute was transformed into a factor variable with labels "MALE" and "FEMALE". The "ChestPainType" attribute was converted there full form after rounding. Similarly, the "RestingECG" attribute was changed to a factor variable with labels "NORMAL", "ABNORMALITY". The "ExerciseAngina" attribute was converted to a factor variable with labels "YES" and "NO". Lastly, the "HeartDisease" attribute was transformed into a factor variable with labels "HIGH CHANCE" and "LOW CHANCE". A preview of the updated dataset can be observed through the head of "heart\_attack\_data\_6".

**Remove Negative Values from Dataset:**

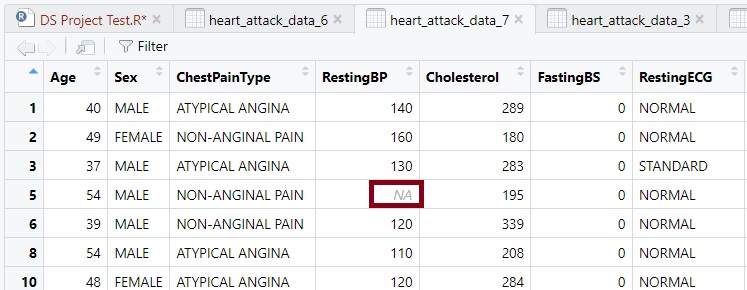
heart\_attack\_data\_7 <- heart\_attack\_data\_6

heart\_attack\_data\_7[heart\_attack\_data\_6 < 0] <- NA

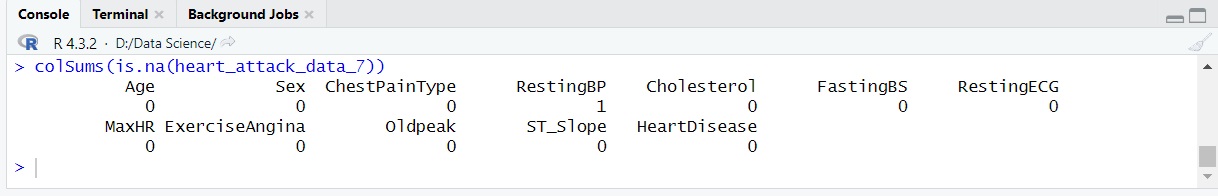


**Before:**

**After:**



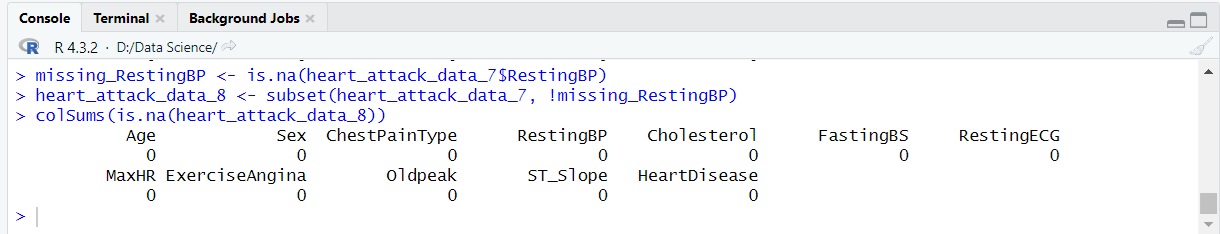
colSums(is.na(heart\_attack\_data\_7))



missing\_RestingBP <- is.na(heart\_attack\_data\_7$RestingBP)

heart\_attack\_data\_8 <- subset(heart\_attack\_data\_7, !missing\_RestingBP)

colSums(is.na(heart\_attack\_data\_8))

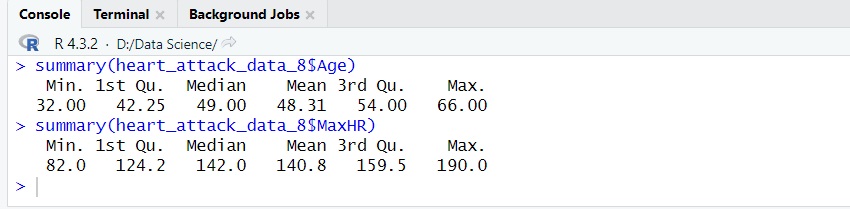


**Explanation:** To eliminate Negative data from the heart\_attack\_data dataset, a process of removing negative data based on all variables can be performed. This helps ensure that each row in the dataset is Positive avlues and prevents any redundant information from skewing the analysis or results. By eliminating Negative data, researchers can work with a more accurate and reliable dataset for further exploration and analysis.

**Univariate Data Exploration:**

summary(heart\_attack\_data\_8$Age)

summary(heart\_attack\_data\_8$MaxHR)



**For Age attribute:**

mean(heart\_attack\_data\_8$Age)

median(heart\_attack\_data\_8$Age)

var(heart\_attack\_data\_8$Age)

sd(heart\_attack\_data\_8$Age)

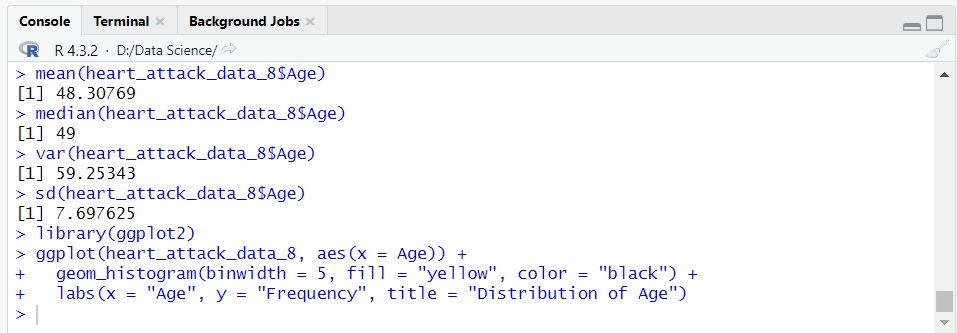
install.packages("ggplot2")

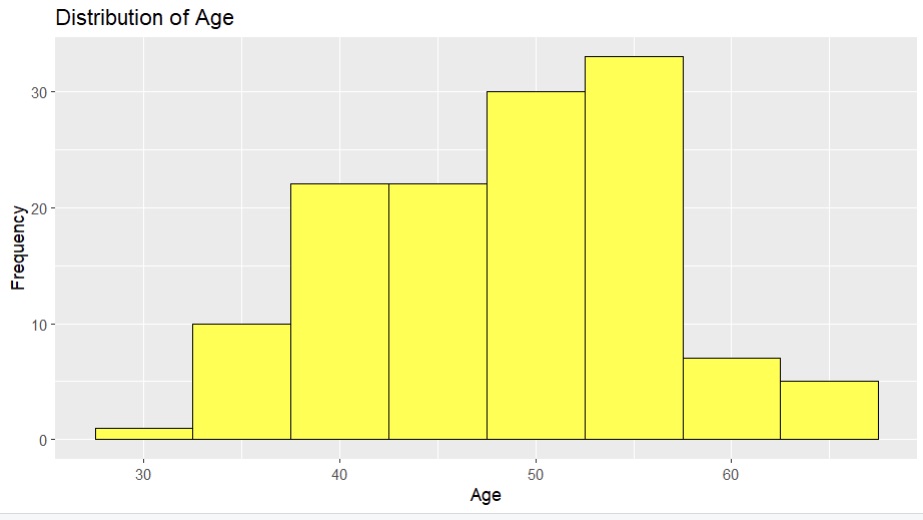
library(ggplot2)

ggplot(heart\_attack\_data\_8, aes(x = Age)) +

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

labs(x = "Age", y = "Frequency", title = "Distribution of Age")





**For MaxHR attribute:**

mean(heart\_attack\_data\_8$MaxHR)

median(heart\_attack\_data\_8$MaxHR)

var(heart\_attack\_data\_8$MaxHR)

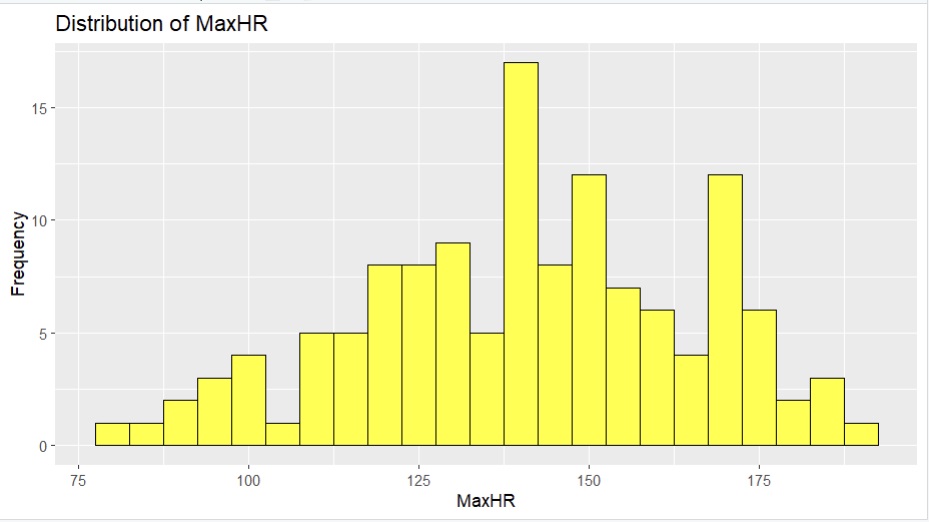
sd(heart\_attack\_data\_8$MaxHR)

library(ggplot2)

ggplot(heart\_attack\_data\_8, aes(x = MaxHR)) +

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

labs(x = "MaxHR", y = "Frequency", title = "Distribution of MaxHR")

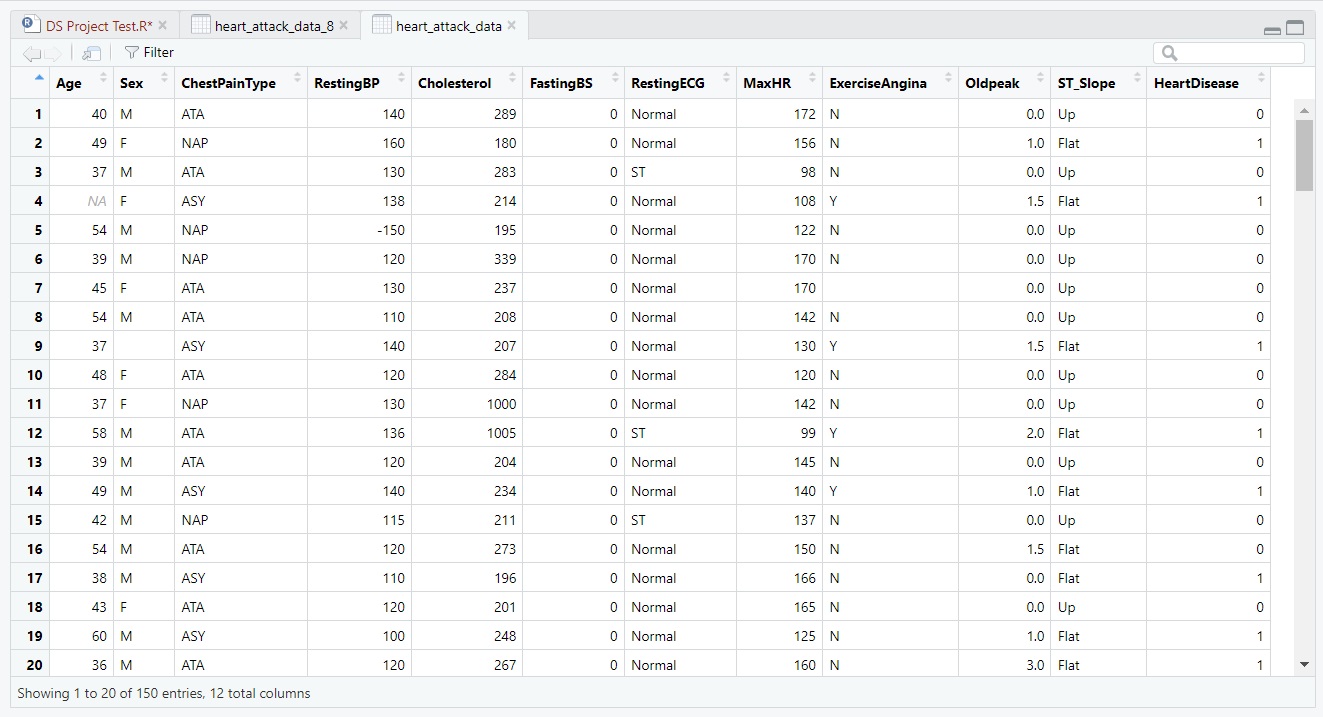


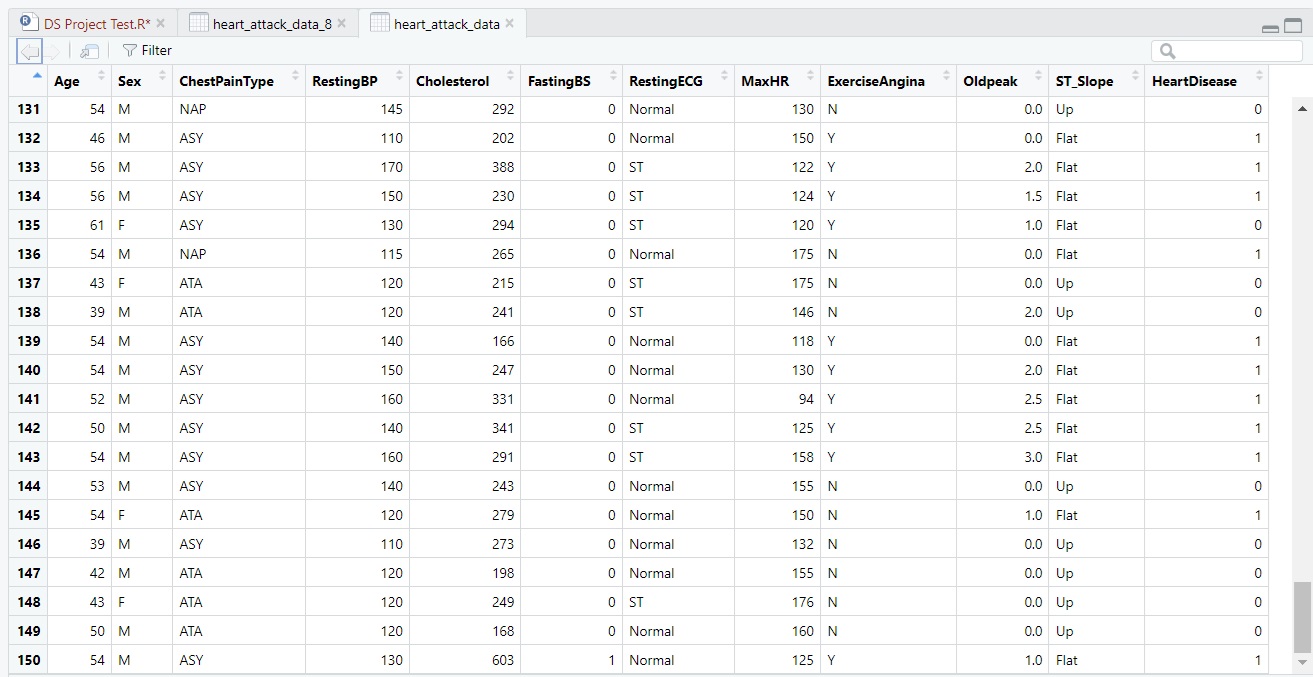
**Explanation:** Univariate data exploration involves examining individual variables in the dataset. Measures such as the mean, median, variance, and standard deviation provide insights into the central tendency, spread, and variability of the data. The "ggplot2" library is commonly used for creating visualizations, such as histograms, to visualize the distribution of variables. Additionally, labeling the axes and adding titles using "labs()" further enhances the interpretability of the visualizations, enabling a comprehensive exploration of the dataset.

# Discussion and Conclusion:

The dataset that was given to us at the beginning of the study showed a significant level of disarray. It had a large number of outliers, missing values, and null values. A number of data preparation procedures were used to solve these problems and get the data ready for analysis. To learn more about each variable in the dataset, a univariate data exploration was also carried out on it.

**Unprepared Dataset-**





**Prepared Dataset-**

