**American International University-Bangladesh**

A picture containing text

Description automatically generated

**INTRODUCTION TO DATA SCIENCE**

**Mid Project**

**Submitted by:**

**Name:** Fabiha Tasnim  
**Id:** 20-43426-1  
**Section:** B

**Submitted to:**

**TOHEDUL ISLAM**

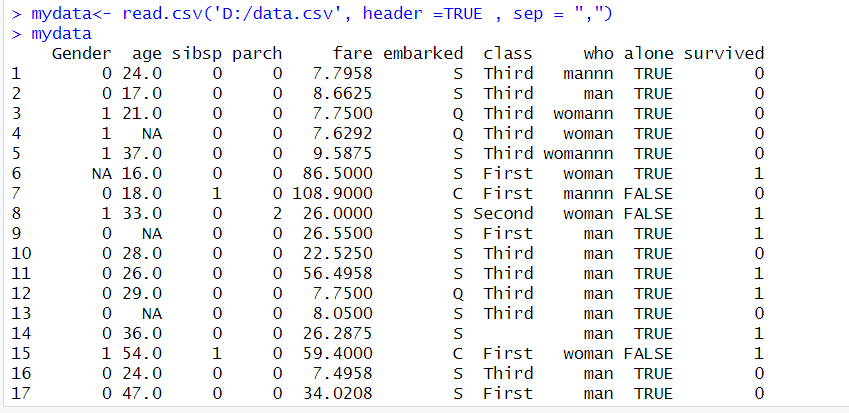
**Project Description:**

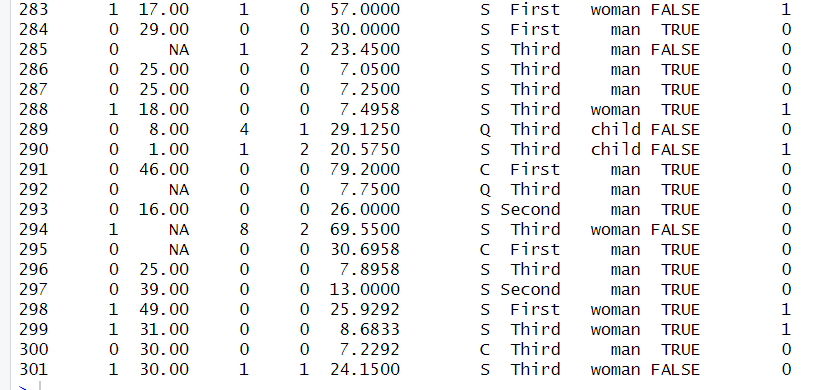
The purpose of this project is to do data preparation and univariate exploration, which entails cleaning up and converting raw data into information that can be used for analysis. This is a titanic dataset. Different columns are used in this dataset such as gender, age, sibsp, parch, fare, embarked, class, who, alone, survived.  Here, 65% of people are male and 35% of people are female. 0 is meant for male and 1 is for females. In the case of survived column 1 is yes and 0 is not to be assumed. There are noisy values and missing values that need to be preprocessed before the data set can be considered complete. Missing data is a common issue in databases that can arise from a column not having a value or from data not being collected properly. The goal is to clean the data by data preparation. After that, explore the dataset.

**Import Dataset:**

mydata<- read.csv('D:/data.csv', header =TRUE , sep = ",")

mydata

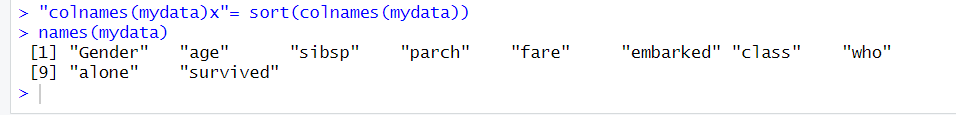




After importing, there were 301 data in this dataset.

**Names of attributes:**

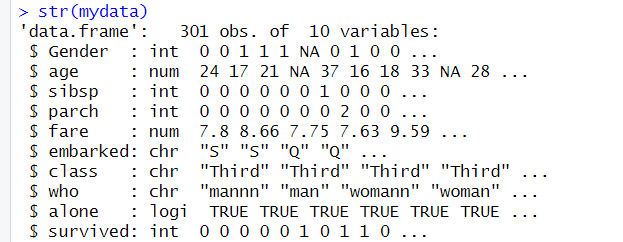
names(mydata)



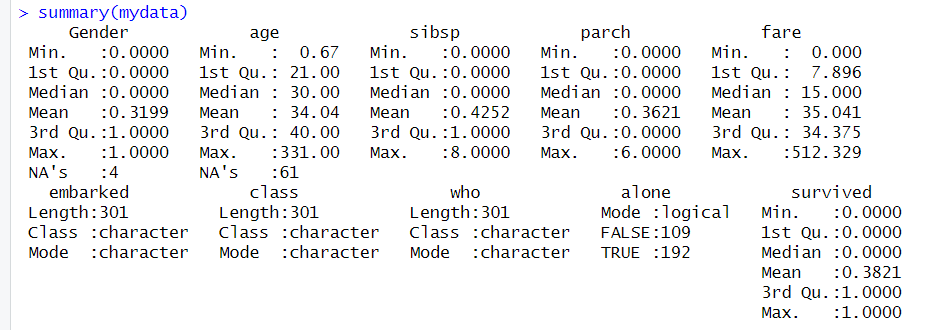
By this, attribute’s name has been shown.

**Structure and Summary of dataset:**

str(mydata)



summary(mydata)



By using summary () function Descriptive Statistics has been shown.

**Data type for dataset:**

typeof(mydata$Gender)

typeof(mydata$age)

typeof(mydata$parch)

typeof(mydata$fare)

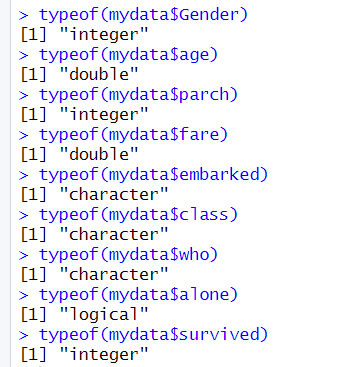
typeof(mydata$embarked)

typeof(mydata$class)

typeof(mydata$who)

typeof(mydata$alone)

typeof(mydata$survived)

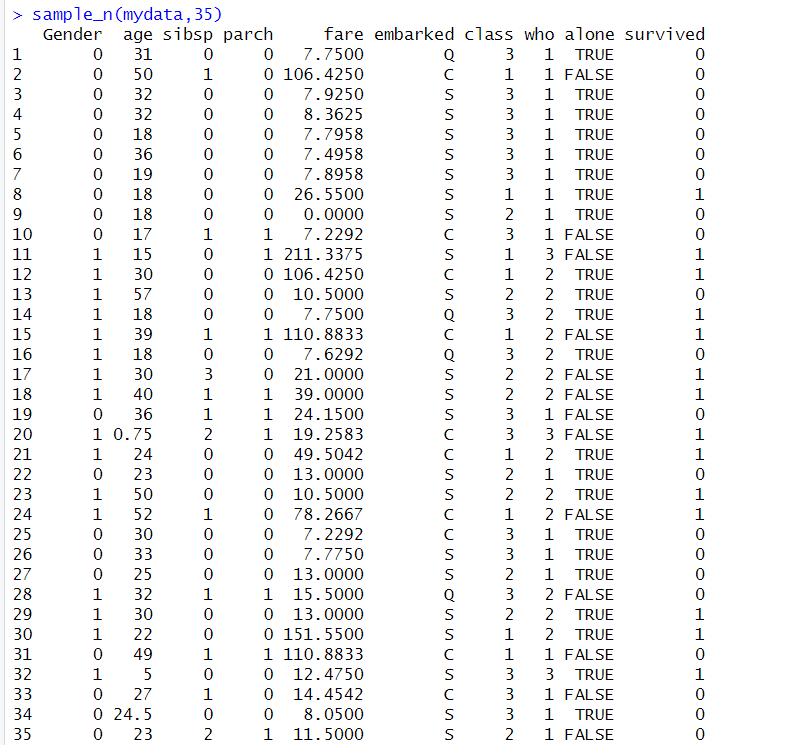


**Sampling Dataset:**

install.packages("dplyr")

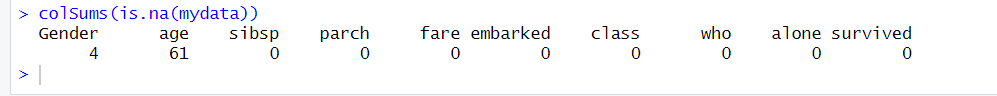
library("dplyr")

sample\_n(mydata,35)



**Find the missing values of rows:**

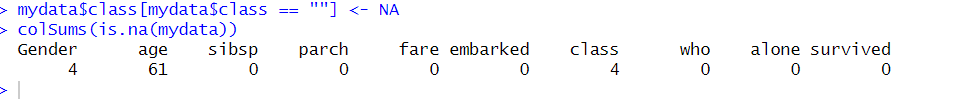
colSums(is.na(mydata))



To discover the missing values of blank row,

mydata$class[mydata$class == ""] <- NA

colSums(is.na(mydata))

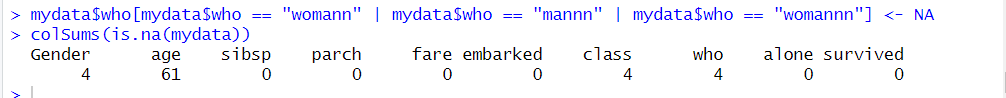


The blank row of class attribute converted to NA.

For invalid values,

mydata$who[mydata$who == "womann" | mydata$who == "mannn" | mydata$who == "womannn"] <- NA

colSums(is.na(mydata))

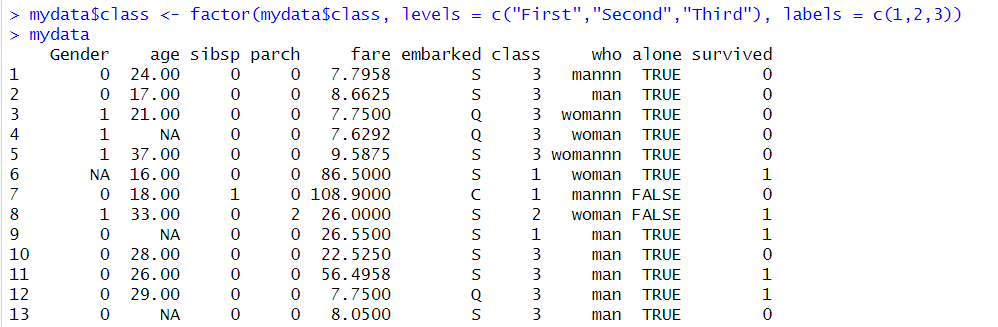


The invalid values of rows converted to NA.

**Conversion of Who, Class and Gender attribute:**

mydata$class <- factor(mydata$class, levels = c("First","Second","Third"), labels = c(1,2,3))

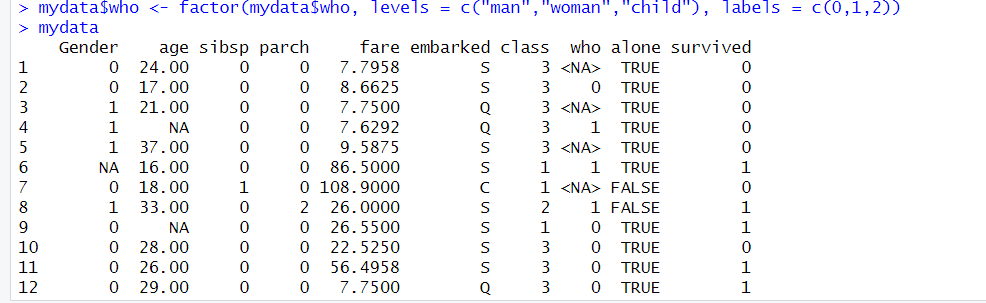
mydata



Converting ‘First’, ‘Second’ and ‘Third’ as 1, 2 &3.

mydata$who <- factor(mydata$who, levels = c("man","woman","child"), labels = c(0,1,2))

mydata

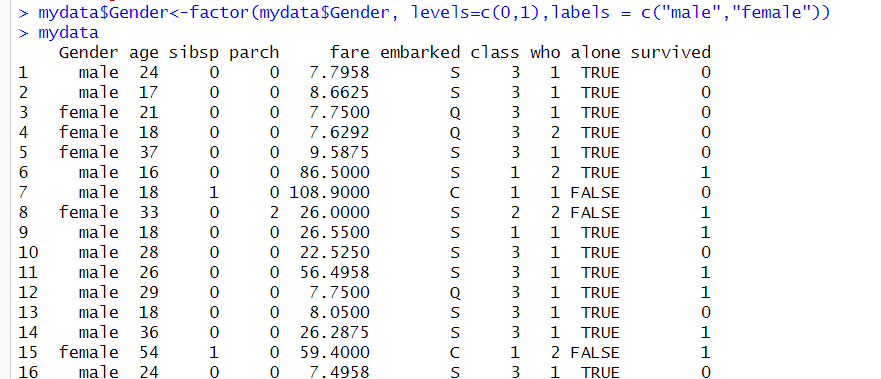


Annotating ‘man’, ‘woman’ & ‘child’ as 0,1 & 2.

mydata$Gender<-factor(mydata$Gender, levels=c(0,1),labels = c("male","female"))

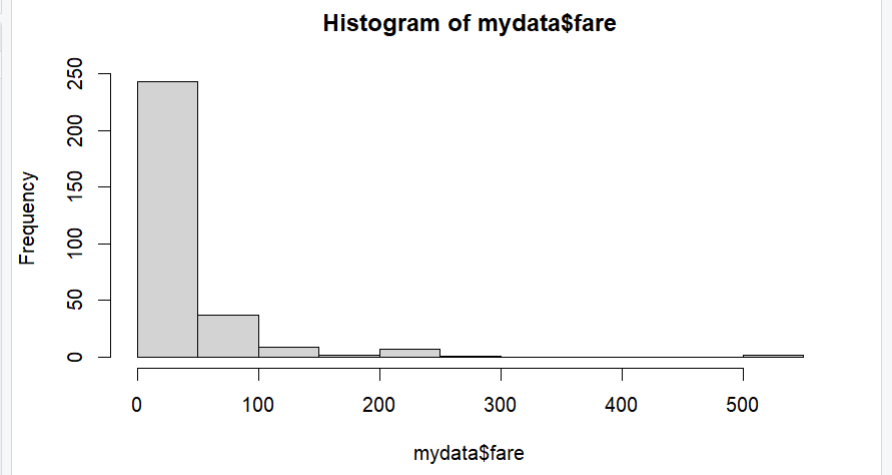
mydata

Converting 0 & 1 as ‘male’ and ‘female’.

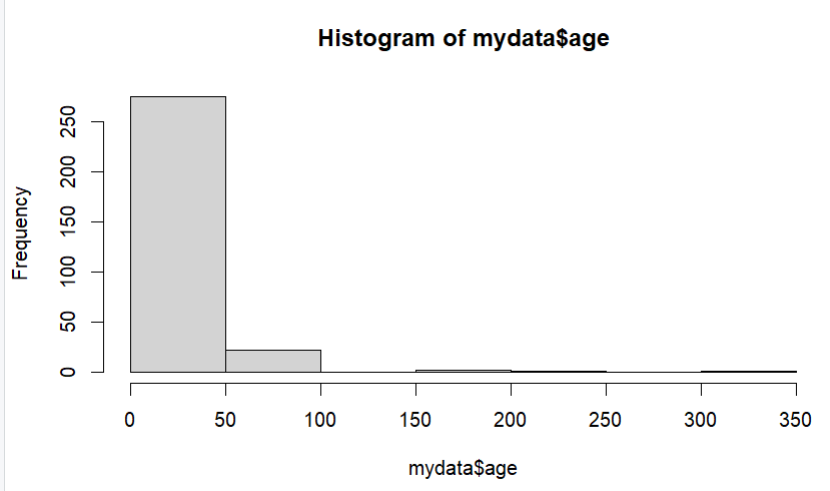


**Detecting outliers for finding noisy values:**

hist(mydata$fare)



hist(mydata$age)



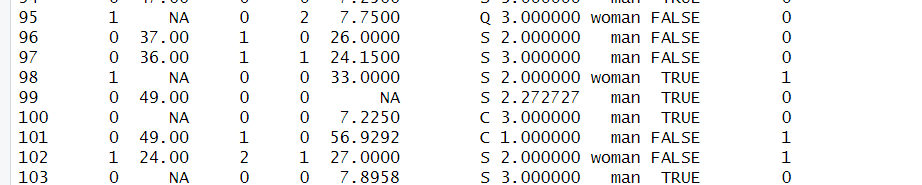
**Converting outliers as missing values:**

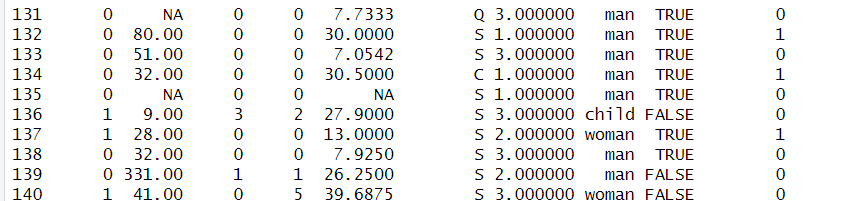
mydata$fare[ mydata$fare > 250] = NA

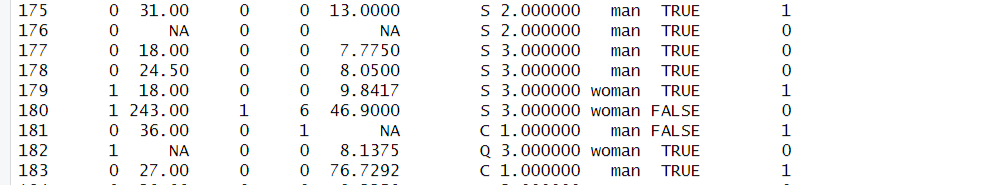
mydata

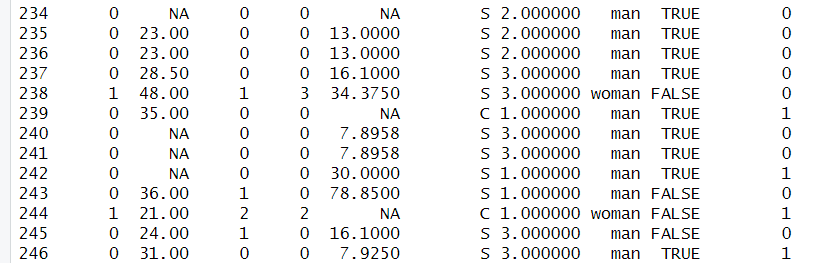
mydata$fare[ mydata$fare < 2 ] = NA

mydata





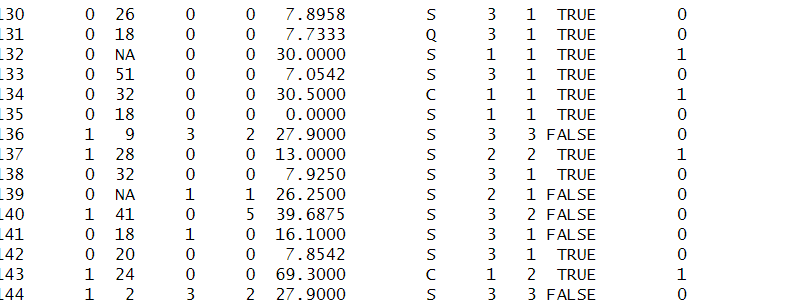




For fare column, the outliers has been detected and converted to missing values.

mydata$age[mydata$age > 70] = NA

mydata

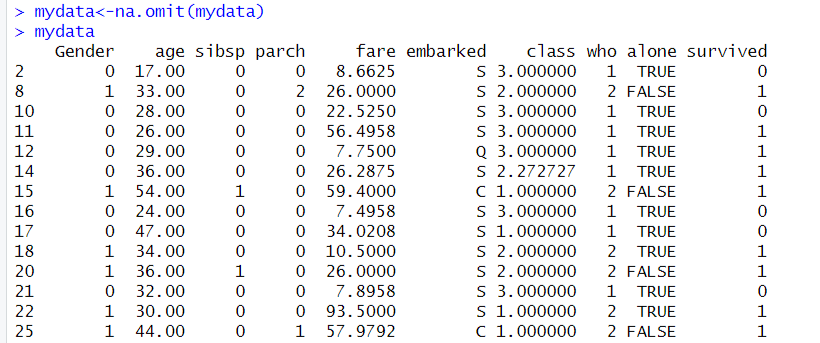


**Recover missing values:**

1. **Deleting the rows of missing value**

mydata<-na.omit(mydata)

mydata



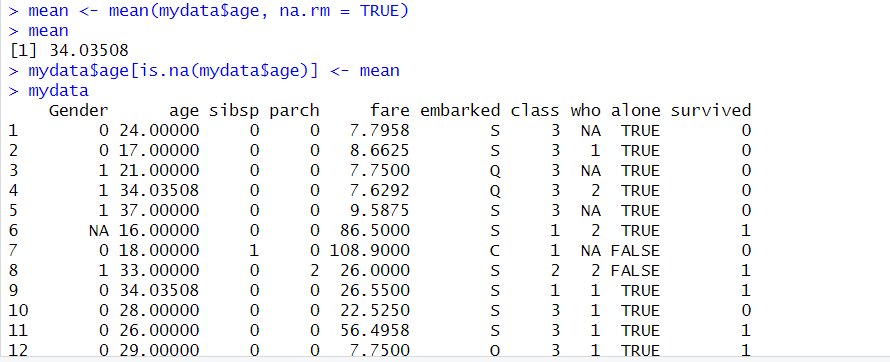
1. **Recovering missing values by mean value.**

mean <- mean(mydata$age, na.rm = TRUE)

mean

mydata$age[is.na(mydata$age)] <- mean

mydata

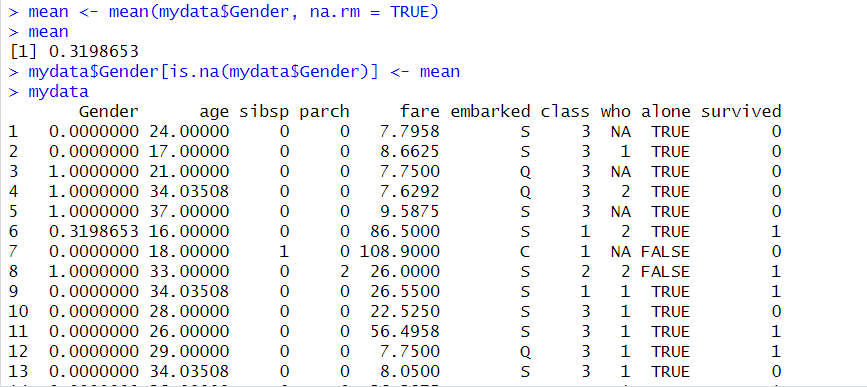


mean <- mean(mydata$Gender, na.rm = TRUE)

mean

mydata$Gender[is.na(mydata$Gender)] <- mean

mydata

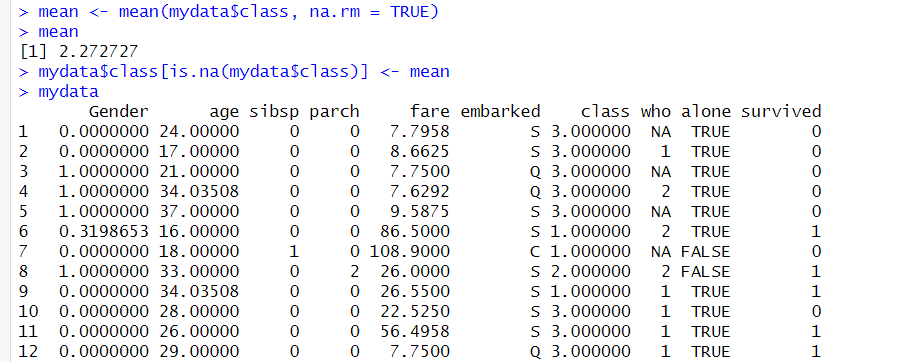


mean <- mean(mydata$class, na.rm = TRUE)

mean

mydata$class[is.na(mydata$class)] <- mean

mydata

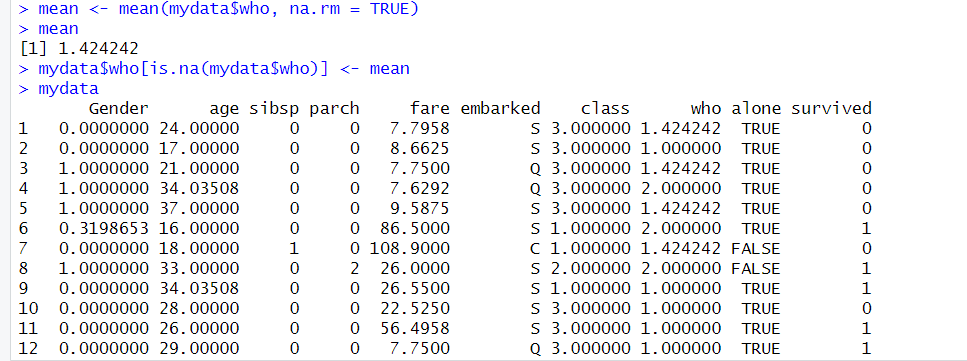


mean <- mean(mydata$who, na.rm = TRUE)

mean

mydata$who[is.na(mydata$who)] <- mean

mydata

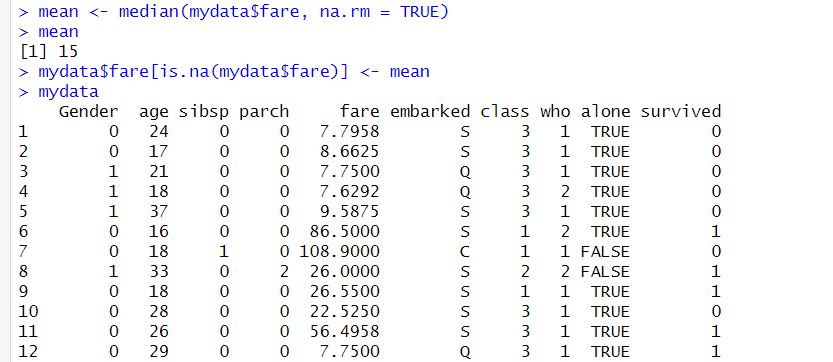


mean <- median(mydata$fare, na.rm = TRUE)

mean

mydata$fare[is.na(mydata$fare)] <- mean

mydata



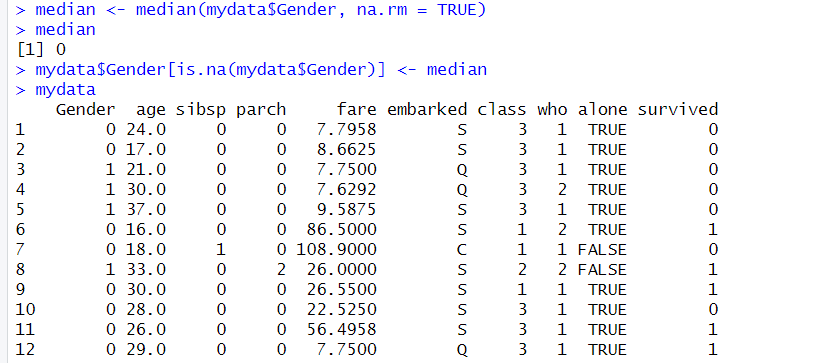
1. **Recovering missing values by median value.**

median <- median(mydata$Gender, na.rm = TRUE)

median

mydata$Gender[is.na(mydata$Gender)] <- median

mydata

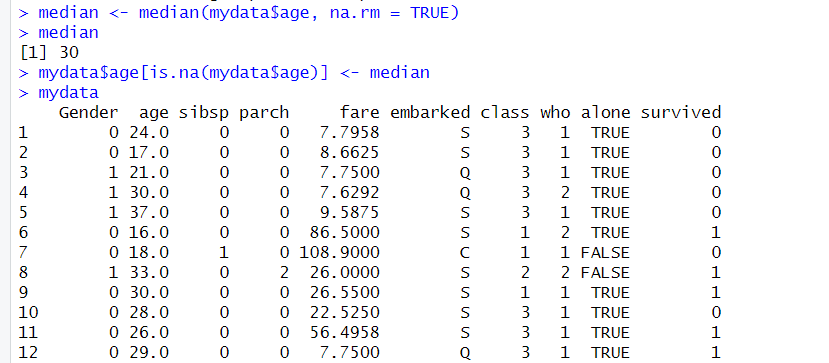


median <- median(mydata$age, na.rm = TRUE)

median

mydata$age[is.na(mydata$age)] <- median

mydata

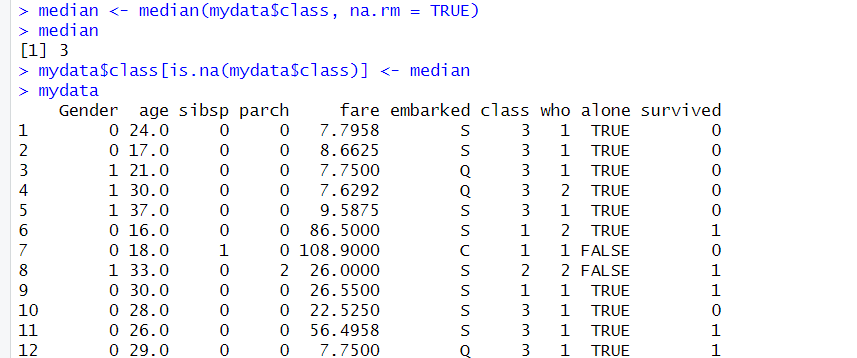


median <- median(mydata$class, na.rm = TRUE)

median

mydata$class[is.na(mydata$class)] <- median

mydata

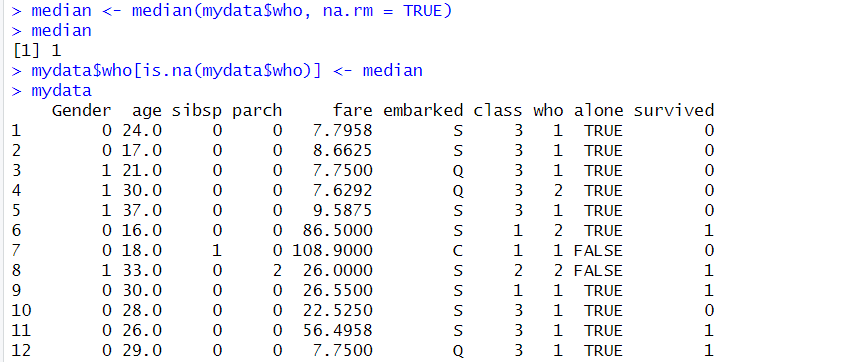


median <- median(mydata$who, na.rm = TRUE)

median

mydata$who[is.na(mydata$who)] <- median

mydata

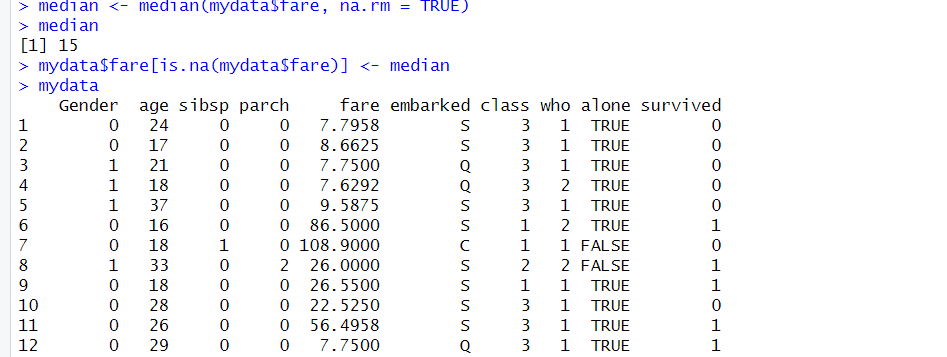


median <- median(mydata$fare, na.rm = TRUE)

median

mydata$fare[is.na(mydata$fare)] <- median

mydata



1. **Recovering missing values by mode value.**

mode <- names(sort(-table(mydata$Gender))) [1]

mode

mydata$Gender[is.na(mydata$Gender)] <- mode

mydata

mode <- names(sort(-table(mydata$age))) [1]

mode

mydata$age[is.na(mydata$age)] <- mode

mydata

mode <- names(sort(-table(mydata$class))) [1]

mode

mydata$class[is.na(mydata$class)] <- mode

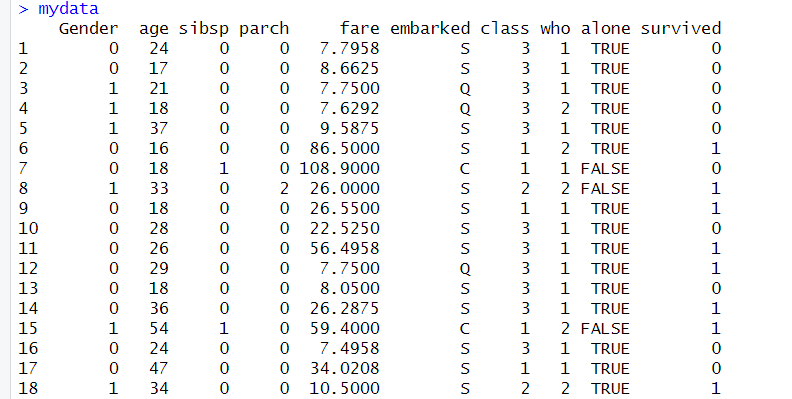
mydata

mode <- names(sort(-table(mydata$who))) [1]

mode

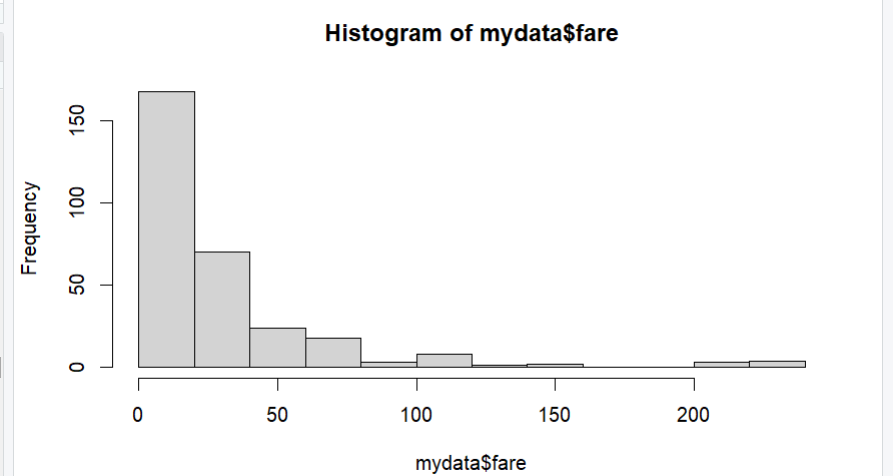
mydata$who[is.na(mydata$who)] <- mode

mydata

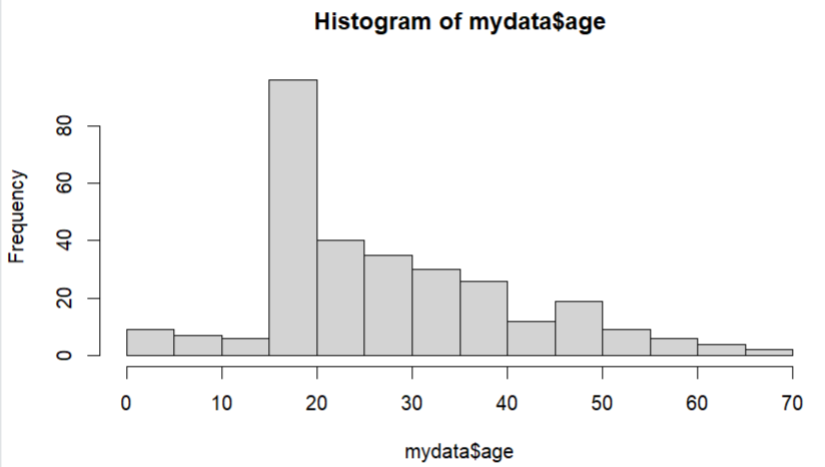


**After Recovering the outliers:**

hist(mydata$fare)

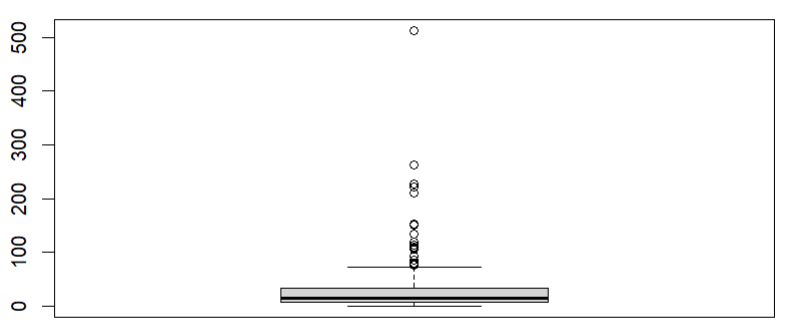


hist(mydata$age)



boxplot(mydata$fare)

Before outlier the fare attribute,



mydata$fare[ mydata$fare > 200] = NA

mydata

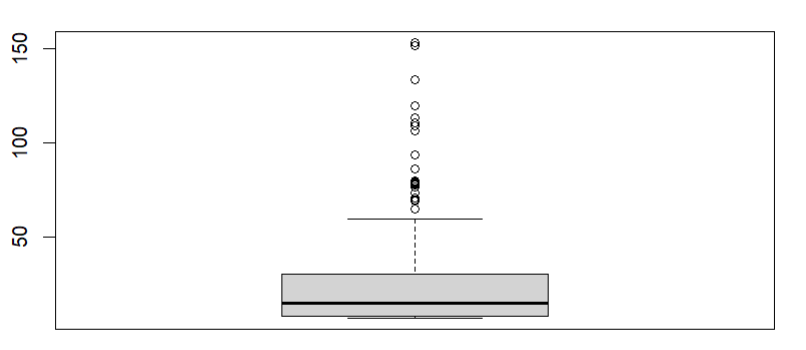
median <- median(mydata$fare, na.rm = TRUE)

median

mydata$fare[is.na(mydata$fare)] <- median

mydata

After recovering the outlier values,



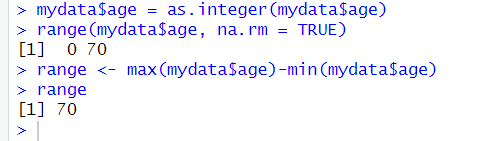
**Standard Deviation and Range:**

**For age,**

range(mydata$age, na.rm = TRUE)

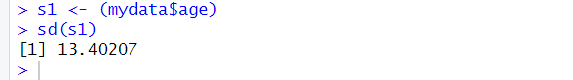
range <- max(mydata$age)-min(mydata$age)

range



s1 <- (mydata$age)

sd(s1)

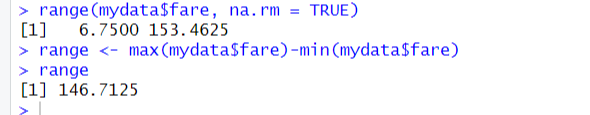


For Fare,

range(mydata$fare, na.rm = TRUE)

range <- max(mydata$fare)-min(mydata$fare)

range



s2 <- (mydata$fare)

sd(s2)



**Univariate Exploration:**

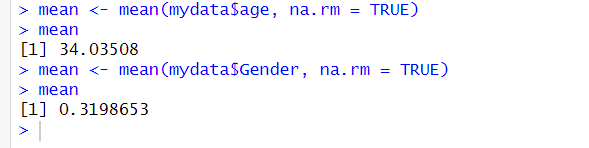
**Mean:**

mean <- mean(mydata$age, na.rm = TRUE)

mean

mean <- mean(mydata$Gender, na.rm = TRUE)

mean



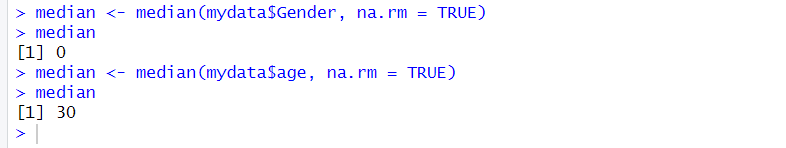
**Median:**

median <- median(mydata$Gender, na.rm = TRUE)

median

median <- median(mydata$age, na.rm = TRUE)

median



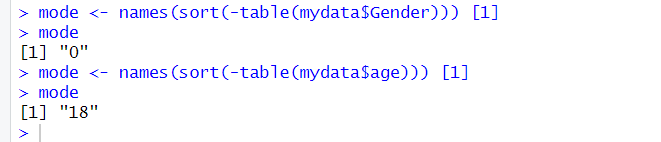
**Mode:**

mode <- names(sort(-table(mydata$Gender))) [1]

mode

mode <- names(sort(-table(mydata$age))) [1]

mode



**The Final Dataset:**

