INTRODUCTION TO DATA SCIENCE

MIDTERM PROJECT REPORT

SUBMITTED BY

SADIR AHMED ZIDAN

ID:20-43263-1

SECTION:C

SUBMITTED TO

DR. ABDUS SALAM

Assistant Professor, CS

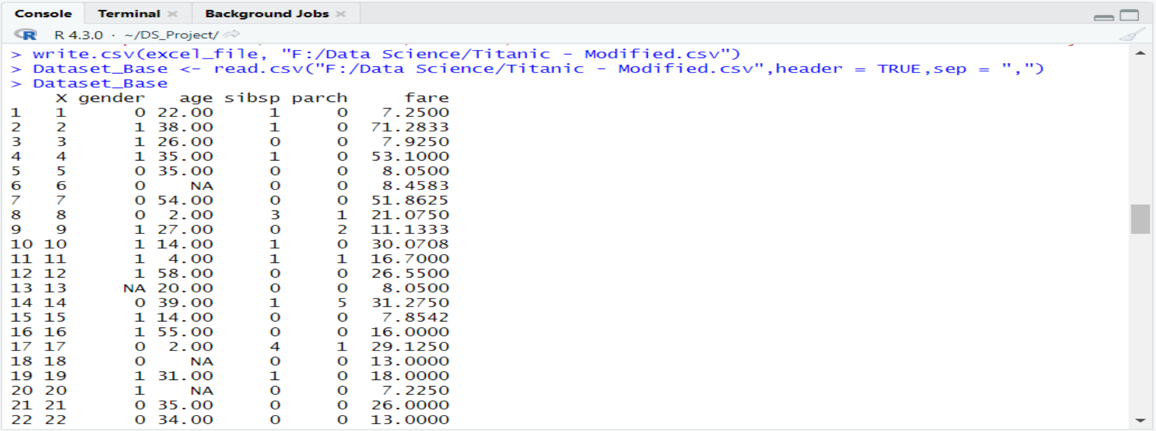
Dataset: In this project, a modified version of “TITANIC” dataset is used. This Titanic dataset is a well-known and frequently used dataset in the field of data science and machine learning. It contains information about the passengers who were aboard the Titanic during its ill-fated maiden voyage in 1912. The dataset provides a glimpse into the demographics and characteristics of the passengers, as well as their survival outcomes. The Titanic dataset provides valuable information about the passengers who were on board. It includes attributes such as gender, age, the number of siblings/spouses (sibsp) and parents/children (parch) accompanying the passenger, fare, port of embarkation, ticket class, passenger category, whether the passenger was alone or not, and the survival status.

- ❑ **Firstly, as I got the dataset in EXCEL format, I need to convert it to CSV format. The following code is about converting the dataset format and view it.**

CODE:

```
install.packages("readxl")

install.packages("writexl")

library(readxl)

library(writexl)

excel_file <- read_excel("F:/Data Science/Titanic - Modified.xlsx")

write.csv(excel_file, "F:/Data Science/Titanic - Modified.csv")

Dataset_Base <- read.csv("F:/Data Science/Titanic - Modified.csv",header = TRUE,sep = ",")

Dataset_Base

View(Dataset_Base)
```

OUTPUT:



```
> write.csv(excel_file, "F:/Data Science/Titanic - Modified.csv")
> Dataset_Base <- read.csv("F:/Data Science/Titanic - Modified.csv",header = TRUE,sep = ",")
> Dataset_Base
    X gender   age sibsp parch    fare
1   1      0 22.00     1     0  7.2500
2   2      1 38.00     1     0 71.2833
3   3      1 26.00     0     0  7.9250
4   4      1 35.00     1     0 53.1000
5   5      0 35.00     0     0  8.0500
6   6      0    NA     0     0  8.4583
7   7      0 54.00     0     0 51.8625
8   8      0  2.00     3     1 21.0750
9   9      1 27.00     0     2 11.1333
10 10      1 14.00     1     0 30.0708
11 11      1  4.00     1     1 16.7000
12 12      1 58.00     0     0 26.5500
13 13     NA 20.00     0     0  8.0500
14 14      0 39.00     1     5 31.2750
15 15      1 14.00     0     0  7.8542
16 16      1 55.00     0     0 16.0000
17 17      0  2.00     4     1 29.1250
18 18      0    NA     0     0 13.0000
19 19      1 31.00     1     0 18.0000
20 20      1    NA     0     0  7.2250
21 21      0 35.00     0     0 26.0000
22 22      0 34.00     0     0 13.0000
```

DATA EXPLORATION:

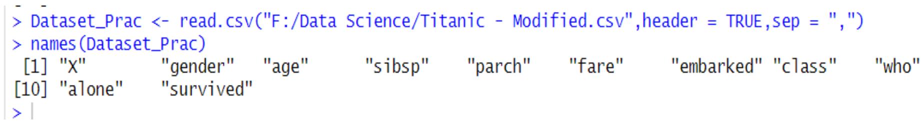
❑ **Checking the names of the variables and the attribute types**

CODE:

Dataset_Prac <- read.csv("F:/Data Science/Titanic - Modified.csv",header = TRUE,sep = ",")

names(Dataset_Prac)

OUTPUT:



```
> Dataset_Prac <- read.csv("F:/Data Science/Titanic - Modified.csv",header = TRUE,sep = ",")
> names(Dataset_Prac)
 [1] "X"        "gender"   "age"      "sibsp"    "parch"    "fare"     "embarked" "class"    "who"     
[10] "alone"    "survived"
> 
```

❑ **Checking the datatypes of the dataset**

CODE:

attributes <- names(Dataset_Base)

dataTypes <- c(typeof(Dataset_Base$X), typeof(Dataset_Base$gender), typeof(Dataset_Base$age),

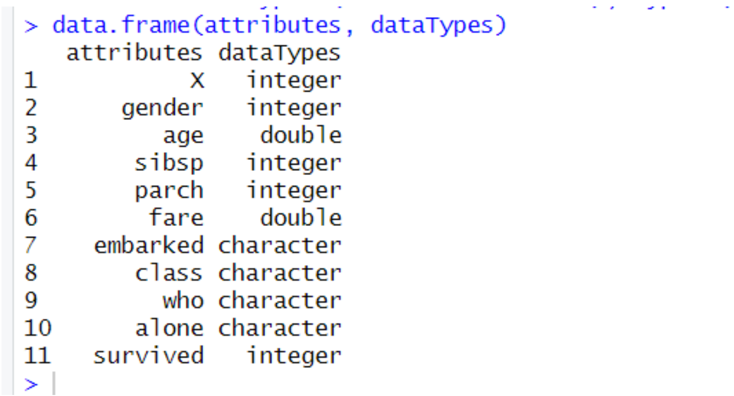
typeof(Dataset_Base$sibsp), typeof(Dataset_Base$parch), typeof(Dataset_Base$fare),

typeof(Dataset_Base$embarked), typeof(Dataset_Base$class), typeof(Dataset_Base$who),

typeof(Dataset_Base$alone), typeof(Dataset_Base$survived))

data.frame(attributes, dataTypes)

OUTPUT:



```
> data.frame(attributes, dataTypes)
   attributes dataTypes
1           X   integer
2      gender   integer
3         age    double
4       sibsp   integer
5       parch   integer
6        fare    double
7    embarked character
8       class character
9         who character
10      alone character
11   survived   integer
> 
```

- ❑ **Annotating column names according to the data to make it easily understandable and viewing it.**

<u>CODE:</u>

One column name at a time

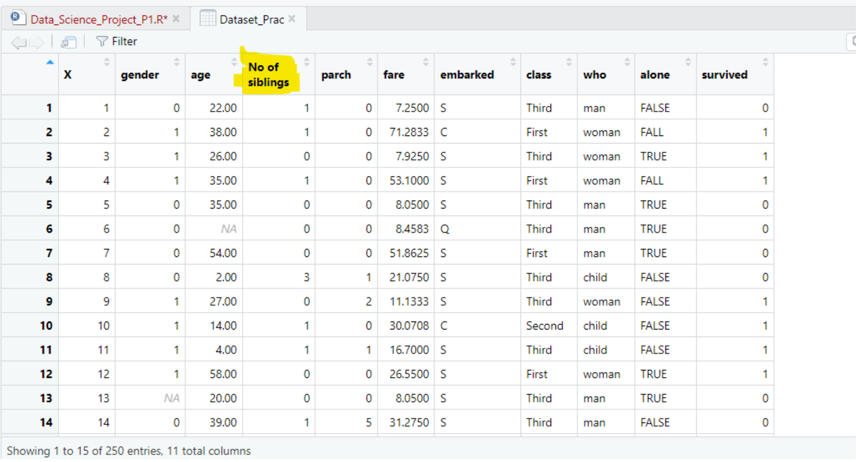
```
names(Dataset_Prac)[4]<-"No of siblings"

View(Dataset_Prac)
```

Multiple columns names at a time

```
colnames(Dataset_Prac) <- c("X","gender", "age", "No of siblings",
                    "no of parents/children aboard", "pass fare", "port embarkation",
                    "ticket class", "who(man/women/child)", "pass was alone or not?",
                    "survived or not")

View(Dataset_Prac)
```

<u>OUTPUT</u>:

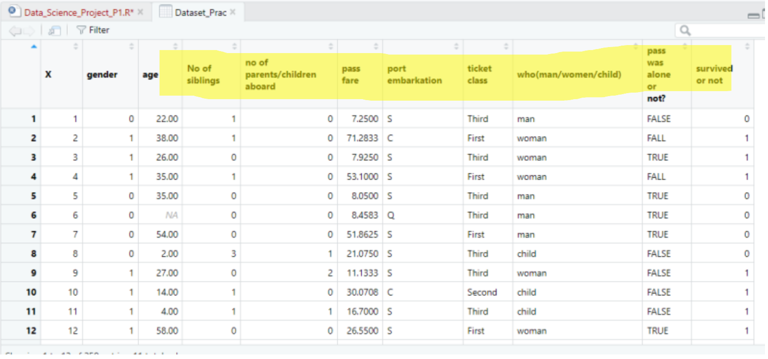
One column at a time



	X	gender	age	No of siblings	parch	fare	embarked	class	who	alone	survived
1	1	0	22.00	1	0	7.2500	S	Third	man	FALSE	0
2	2	1	38.00	1	0	71.2833	C	First	woman	FALL	1
3	3	1	26.00	0	0	7.9250	S	Third	woman	TRUE	1
4	4	1	35.00	1	0	53.1000	S	First	woman	FALL	1
5	5	0	35.00	0	0	8.0500	S	Third	man	TRUE	0
6	6	0	NA	0	0	8.4583	Q	Third	man	TRUE	0
7	7	0	54.00	0	0	51.8625	S	First	man	TRUE	0
8	8	0	2.00	3	1	21.0750	S	Third	child	FALSE	0
9	9	1	27.00	0	2	11.1333	S	Third	woman	FALSE	1
10	10	1	14.00	1	0	30.0708	C	Second	child	FALSE	1
11	11	1	4.00	1	1	16.7000	S	Third	child	FALSE	1
12	12	1	58.00	0	0	26.5500	S	First	woman	TRUE	1
13	13	NA	20.00	0	0	8.0500	S	Third	man	TRUE	0
14	14	0	39.00	1	5	31.2750	S	Third	man	FALSE	0

Showing 1 to 15 of 250 entries, 11 total columns

Multiple columns at a time



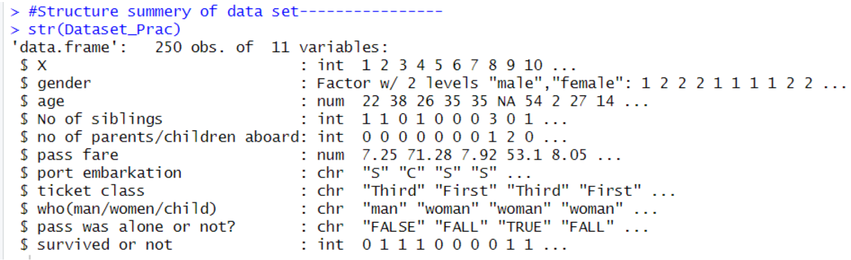
	X	gender	age	No of siblings	no of parents/children aboard	pass fare	port embarkation	ticket class	who(man/women/child)	pass was alone or not?	survived or not
1	1	0	22.00	1	0	7.2500	S	Third	man	FALSE	0
2	2	1	38.00	1	0	71.2833	C	First	woman	FALL	1
3	3	1	26.00	0	0	7.9250	S	Third	woman	TRUE	1
4	4	1	35.00	1	0	53.1000	S	First	woman	FALL	1
5	5	0	35.00	0	0	8.0500	S	Third	man	TRUE	0
6	6	0	NA	0	0	8.4583	Q	Third	man	TRUE	0
7	7	0	54.00	0	0	51.8625	S	First	man	TRUE	0
8	8	0	2.00	3	1	21.0750	S	Third	child	FALSE	0
9	9	1	27.00	0	2	11.1333	S	Third	woman	FALSE	1
10	10	1	14.00	1	0	30.0708	C	Second	child	FALSE	1
11	11	1	4.00	1	1	16.7000	S	Third	child	FALSE	1
12	12	1	58.00	0	0	26.5500	S	First	woman	TRUE	1

- ❑ **Getting structure summary of the dataset**

CODE:

str(Dataset_Prac)

OUTPUT:



```
> #Structure summery of data set----------------
> str(Dataset_Prac)
'data.frame':	250 obs. of  11 variables:
 $ X                          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ gender                     : Factor w/ 2 levels "male","female": 1 2 2 2 1 1 1 1 2 2 ...
 $ age                        : num  22 38 26 35 35 NA 54 2 27 14 ...
 $ No of siblings             : int  1 1 0 1 0 0 0 3 0 1 ...
 $ no of parents/children aboard: int  0 0 0 0 0 0 0 1 2 0 ...
 $ pass fare                  : num  7.25 71.28 7.92 53.1 8.05 ...
 $ port embarkation           : chr  "S" "C" "S" "S" ...
 $ ticket class               : chr  "Third" "First" "Third" "First" ...
 $ who(man/women/child)       : chr  "man" "woman" "woman" "woman" ...
 $ pass was alone or not?     : chr  "FALSE" "FALL" "TRUE" "FALL" ...
 $ survived or not            : int  0 1 1 1 0 0 0 0 1 1 ...
```

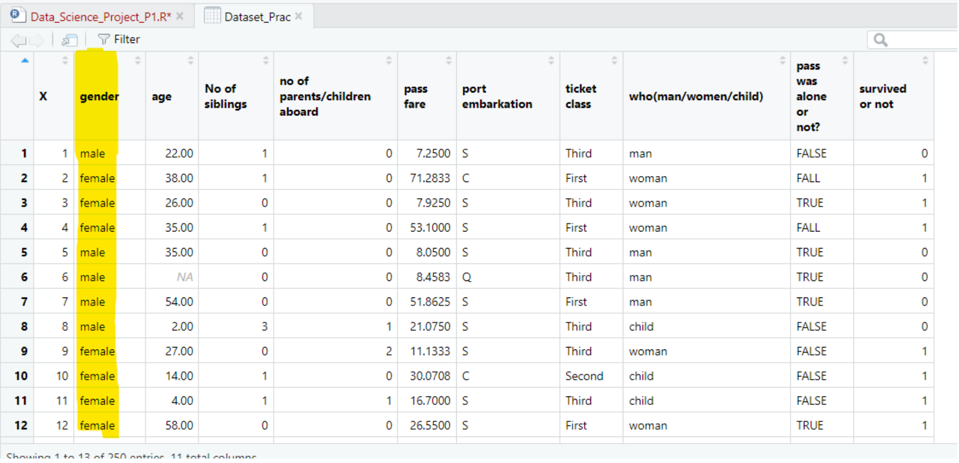
- ❑ **Annotating values of a variable. As the gender value is given as 0 and 1 instead of male and female. I have annotated those values as male and female to make it easier to understand.**

CODE:

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

View(Dataset_Prac)

OUTPUT:



	X	gender	age	No of siblings	no of parents/children aboard	pass fare	port embarkation	ticket class	who(man/women/child)	pass was alone or not?	survived or not
1	1	male	22.00	1	0	7.2500	S	Third	man	FALSE	0
2	2	female	38.00	1	0	71.2833	C	First	woman	FALL	1
3	3	female	26.00	0	0	7.9250	S	Third	woman	TRUE	1
4	4	female	35.00	1	0	53.1000	S	First	woman	FALL	1
5	5	male	35.00	0	0	8.0500	S	Third	man	TRUE	0
6	6	male	NA	0	0	8.4583	Q	Third	man	TRUE	0
7	7	male	54.00	0	0	51.8625	S	First	man	TRUE	0
8	8	male	2.00	3	1	21.0750	S	Third	child	FALSE	0
9	9	female	27.00	0	2	11.1333	S	Third	woman	FALSE	1
10	10	female	14.00	1	0	30.0708	C	Second	child	FALSE	1
11	11	female	4.00	1	1	16.7000	S	Third	child	FALSE	1
12	12	female	58.00	0	0	26.5500	S	First	woman	TRUE	1

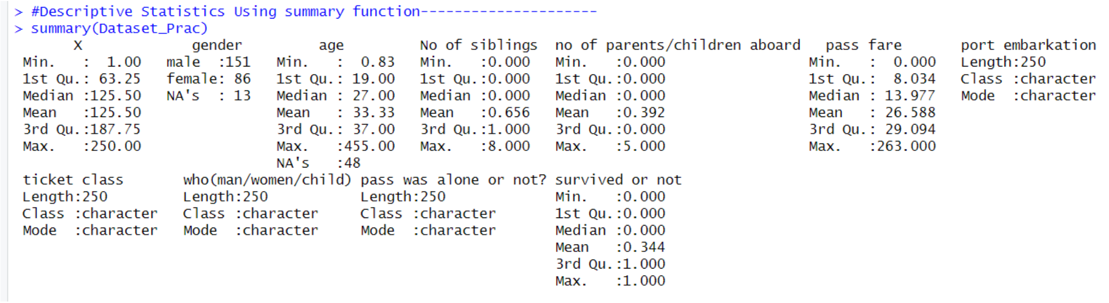
Showing 1 to 13 of 250 entries, 11 total columns

- ❑ **Getting Descriptive Statistics Using summary function**

CODE:

summary(Dataset_Prac)

OUTPUT:



```
> #Descriptive Statistics Using summary function--------------------
> summary(Dataset_Prac)
       X             gender          age          No of siblings  no of parents/children aboard   pass fare       port embarkation
 Min.   :  1.00   male  :151   Min.   :  0.83   Min.   :0.000   Min.   :0.000                 Min.   :  0.000   Length:250
 1st Qu.: 63.25   female: 86   1st Qu.: 19.00   1st Qu.:0.000   1st Qu.:0.000                 1st Qu.:  8.034   Class :character
 Median :125.50   NA's  : 13   Median : 27.00   Median :0.000   Median :0.000                 Median : 13.977   Mode  :character
 Mean   :125.50                Mean   : 33.33   Mean   :0.656   Mean   :0.392                 Mean   : 26.588
 3rd Qu.:187.75                3rd Qu.: 37.00   3rd Qu.:1.000   3rd Qu.:0.000                 3rd Qu.: 29.094
 Max.   :250.00                Max.   :455.00   Max.   :8.000   Max.   :5.000                 Max.   :263.000
                               NA's   :48
 ticket class        who(man/women/child) pass was alone or not? survived or not
 Length:250          Length:250           Length:250             Min.   :0.000
 Class :character    Class :character     Class :character       1st Qu.:0.000
 Mode  :character    Mode  :character     Mode  :character       Median :0.000
                                                                 Mean   :0.344
                                                                 3rd Qu.:1.000
                                                                 Max.   :1.000
```

- **Multiple column Standard deviation for numeric values. As we know, standard deviation can be measured only for numeric values, so I used summerise_if to find variables with numeric values and calculating their standard deviation using sd and keeping it in another variable to show separately. And used "dplyr" library to use summerise_if.**

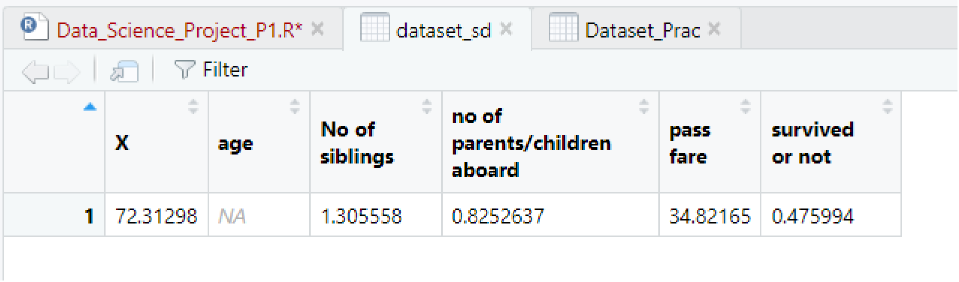
CODE:

library(dplyr)

dataset_sd <- Dataset_Prac %>% summarise_if(is.numeric,sd)

View(dataset_sd)

OUTPUT:



	X	age	No of siblings	no of parents/children aboard	pass fare	survived or not
1	72.31298	NA	1.305558	0.8252637	34.82165	0.475994

- **Row wise standard deviation is used to get standard deviation of specific column and row wise. For this purpose, I need to use "matrixStats" and "dplyr" library. Here I have calculated standard deviation for column 3,4 and 4,5. Also created new column to store these values.**

CODE:

library(matrixStats)

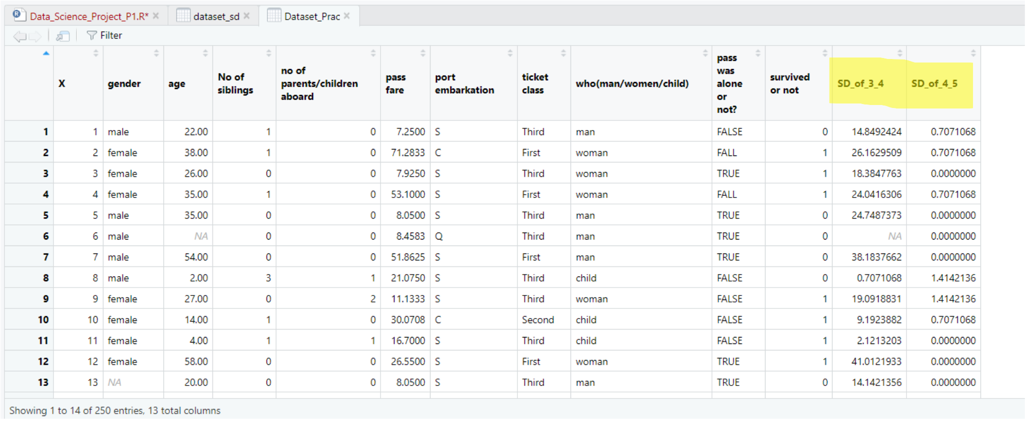
library(dplyr)

Dataset_Prac$SD_of_3_4=rowSds(as.matrix(Dataset_Prac[,c(3,4)]))

Dataset_Prac$SD_of_4_5=rowSds(as.matrix(Dataset_Prac[,c(4,5)]))

View(Dataset_Prac)

OUTPUT:



	X	gender	age	No of siblings	no of parents/children aboard	pass fare	port embarkation	ticket class	who(man/women/child)	pass was alone or not?	survived or not	SD_of_3_4	SD_of_4_5
1	1	male	22.00	1	0	7.2500	S	Third	man	FALSE	0	14.8492424	0.7071068
2	2	female	38.00	1	0	71.2833	C	First	woman	FALL	1	26.1629509	0.7071068
3	3	female	26.00	0	0	7.9250	S	Third	woman	TRUE	1	18.3847763	0.0000000
4	4	female	35.00	1	0	53.1000	S	First	woman	FALL	1	24.0416306	0.7071068
5	5	male	35.00	0	0	8.0500	S	Third	man	TRUE	0	24.7487373	0.0000000
6	6	male	NA	0	0	8.4583	Q	Third	man	TRUE	0	NA	0.0000000
7	7	male	54.00	0	0	51.8625	S	First	man	TRUE	0	38.1837662	0.0000000
8	8	male	2.00	3	1	21.0750	S	Third	child	FALSE	0	0.7071068	1.4142136
9	9	female	27.00	0	2	11.1333	S	Third	woman	FALSE	1	19.0918831	1.4142136
10	10	female	14.00	1	0	30.0708	C	Second	child	FALSE	1	9.1923882	0.7071068
11	11	female	4.00	1	1	16.7000	S	Third	child	FALSE	1	2.1213203	0.0000000
12	12	female	58.00	0	0	26.5500	S	First	woman	TRUE	1	41.0121933	0.0000000
13	13	NA	20.00	0	0	8.0500	S	Third	man	TRUE	0	14.1421356	0.0000000

Showing 1 to 14 of 250 entries, 13 total columns

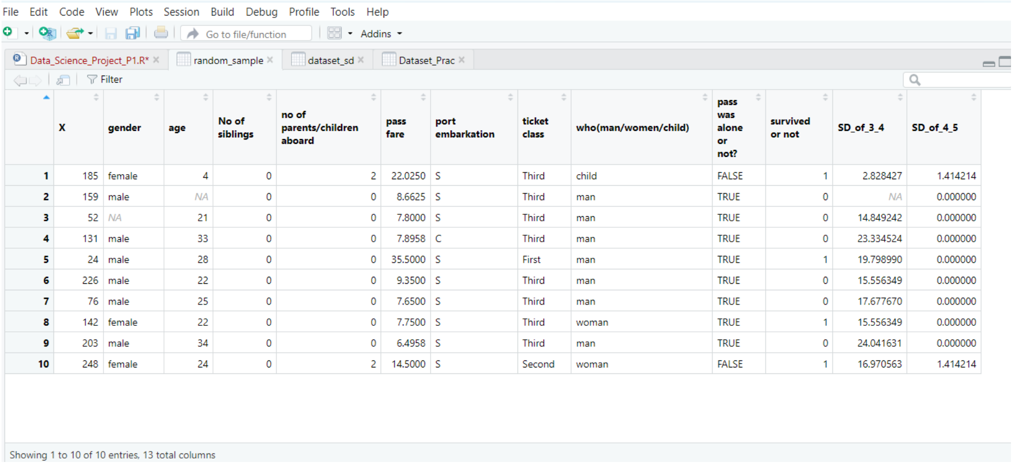
- ❑ **Taking random N rows from the dataset to know about data from a short set**
- ❑ **Viewing a single column and its SD in a different variable**

CODE:

```
random_sample <- sample_n(Dataset_Prac,10)

random_fare_sd <- random_sample$`pass fare`

sd(random_fare_sd)

View(random_sample)
```

OUTPUT:

Random sample dataset

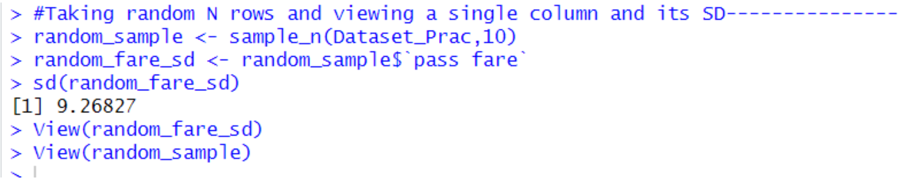


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Data_Science_Project_P1.R*
random_sample
dataset_sd
Dataset_Prac
Filter

	X	gender	age	No of siblings	no of parents/children aboard	pass fare	port embarkation	ticket class	who(man/women/child)	pass was alone or not?	survived or not	SD_of_3_4	SD_of_4_5
1	185	female	4	0	2	22.0250	S	Third	child	FALSE	1	2.828427	1.414214
2	159	male	NA	0	0	8.6625	S	Third	man	TRUE	0	NA	0.000000
3	52	NA	21	0	0	7.8000	S	Third	man	TRUE	0	14.849242	0.000000
4	131	male	33	0	0	7.8958	C	Third	man	TRUE	0	23.334524	0.000000
5	24	male	28	0	0	35.5000	S	First	man	TRUE	1	19.798990	0.000000
6	226	male	22	0	0	9.3500	S	Third	man	TRUE	0	15.556349	0.000000
7	76	male	25	0	0	7.6500	S	Third	man	TRUE	0	17.677670	0.000000
8	142	female	22	0	0	7.7500	S	Third	woman	TRUE	1	15.556349	0.000000
9	203	male	34	0	0	6.4958	S	Third	man	TRUE	0	24.041631	0.000000
10	248	female	24	0	2	14.5000	S	Second	woman	FALSE	1	16.970563	1.414214

Showing 1 to 10 of 10 entries, 13 total columns

Standard Deviation of "pass fare" variable from random sample dataset



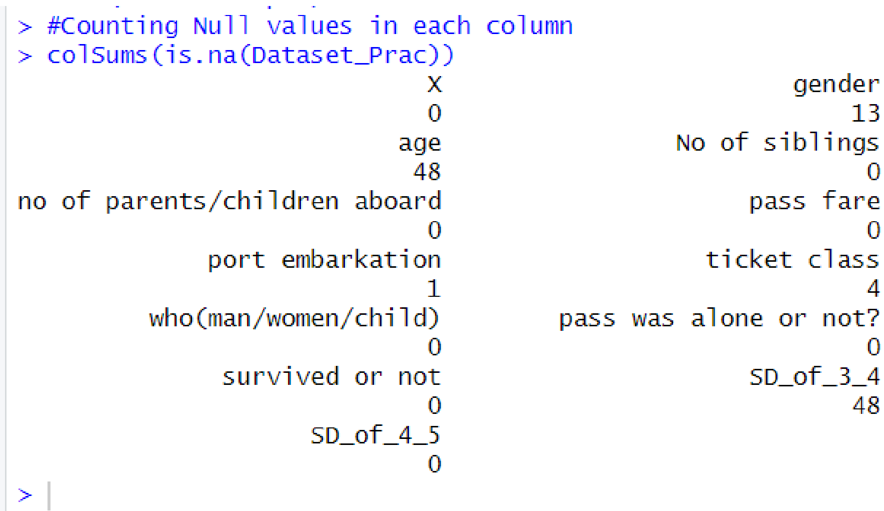
```
> #Taking random N rows and viewing a single column and its SD---------------
> random_sample <- sample_n(Dataset_Prac,10)
> random_fare_sd <- random_sample$`pass fare`
> sd(random_fare_sd)
[1] 9.26827
> View(random_fare_sd)
> View(random_sample)
>
```

- ❑ **Counting Null values in each column by using colSums, it will show the number of missing values for each column.**

CODE:

colSums(is.na(Dataset_Prac))

OUTPUT:



```
> #Counting Null values in each column
> colSums(is.na(Dataset_Prac))
                            X                        gender
                            0                            13
                          age                No of siblings
                           48                             0
no of parents/children aboard                     pass fare
                            0                             0
             port embarkation                  ticket class
                            1                             4
         who(man/women/child)        pass was alone or not?
                            0                             0
              survived or not                     SD_of_3_4
                            0                            48
                    SD_of_4_5
                            0
>
```

Data Preparation:

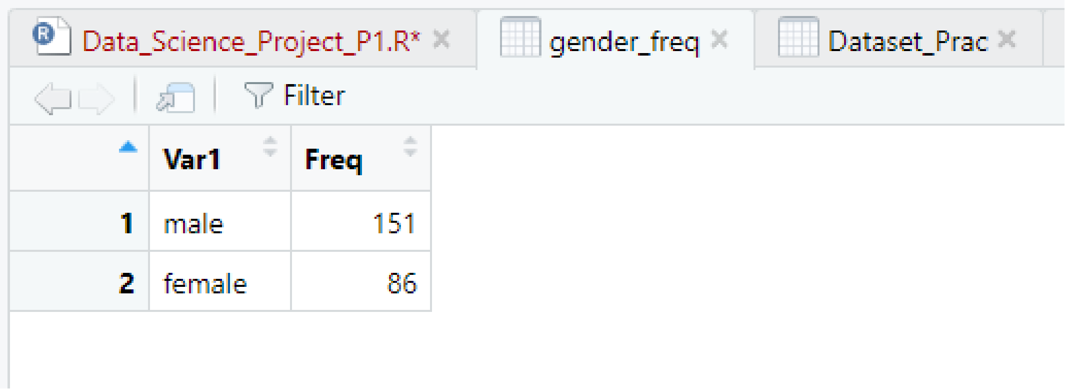
- ❑ **Outlier detection with missing values as I have not handled the missing values yet. To detect outliers, creating frequency table for categorical values.**
- ❑ **Created a frequency table for the "gender" variable first and keeping it in different dataset.**

CODE:

gender_freq <- table(Dataset_Prac$gender)

View(gender_freq)

OUTPUT:



	Var1	Freq
1	male	151
2	female	86

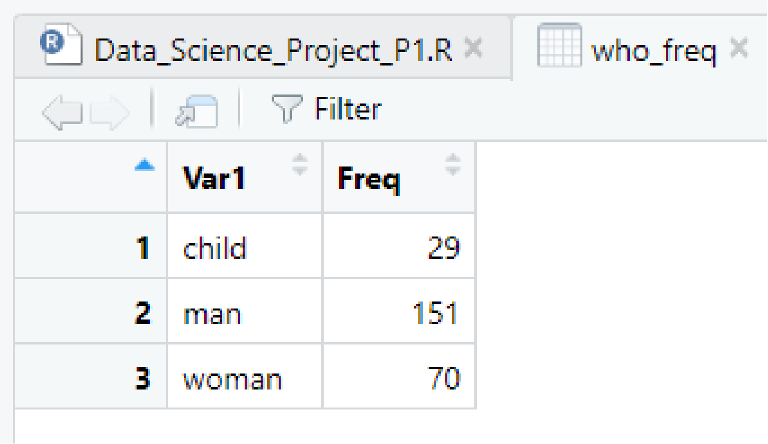
As there are just 2 types of values and understandable, so there are no outliers detected here.

- □ **Created a frequency table for the "who(man/women/child)" variable**

CODE:

```
who_freq <- table(Dataset_Prac$`who(man/women/child)`)

View(who_freq)
```

OUTPUT:



	Var1	Freq
1	child	29
2	man	151
3	woman	70

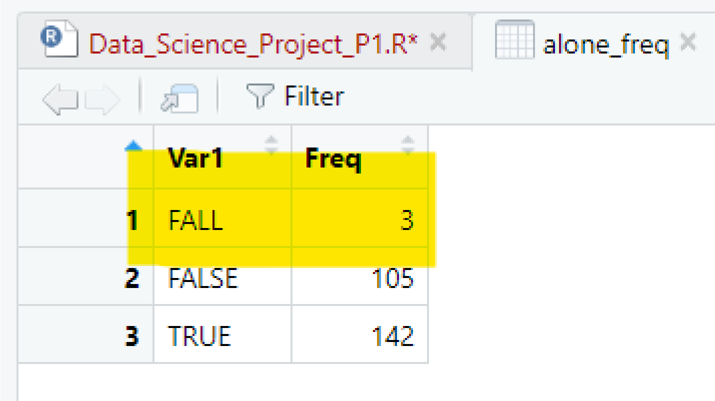
As the values are man, woman and child. Which is understandable for the variable and there is no outlier as invalid value.

- □ **Created a frequency table for the "pass was alone or not?" Variable**

CODE:

```
alone_freq <- table(Dataset_Prac$`pass was alone or not?`)

View(alone_freq)
```

OUTPUT:



	Var1	Freq
1	FALL	3
2	FALSE	105
3	TRUE	142

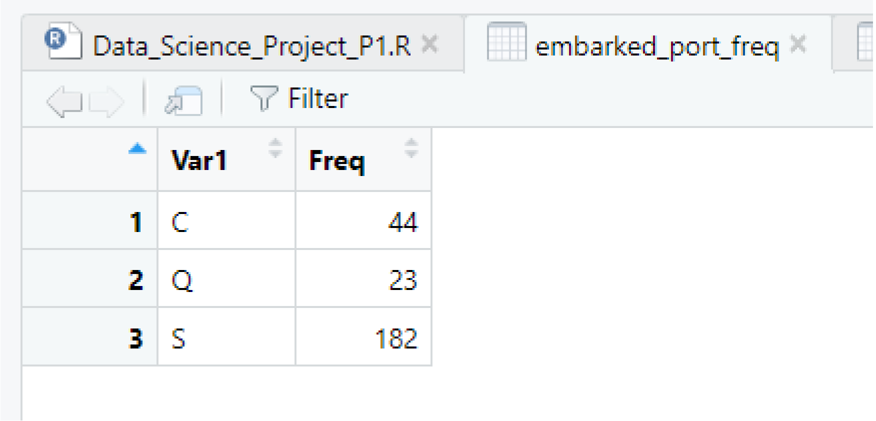
Here we can see, one type of value is invalid, which is not usual as false or true. So, outlier detected.

- ❑ **Created a frequency table for the "port embarkation" variable**

CODE:

```
embarked_port_freq <- table(Dataset_Prac$`port embarkation`)

View(embarked_port_freq)
```

OUTPUT:



	Var1	Freq
1	C	44
2	Q	23
3	S	182

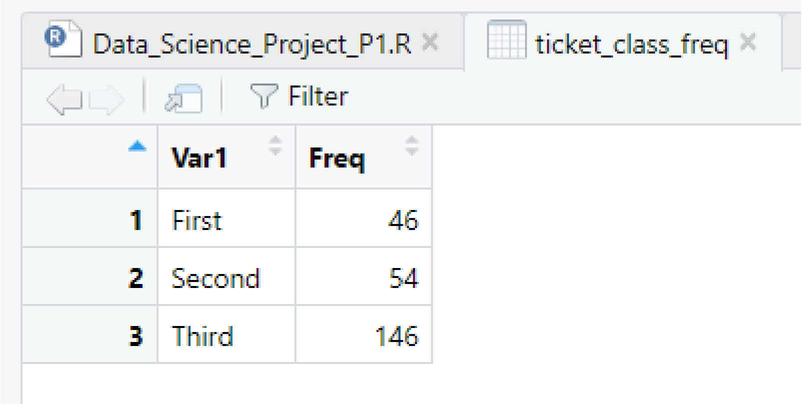
As we can see these values are usual and no outlier detected

- ❑ **Created a frequency table for the "ticket class" variable**

CODE:

```
ticket_class_freq <- table(Dataset_Prac$`ticket class`)

View(ticket_class_freq)
```

OUTPUT:



	Var1	Freq
1	First	46
2	Second	54
3	Third	146

Here, all the values seem usual, so, no outlier detected

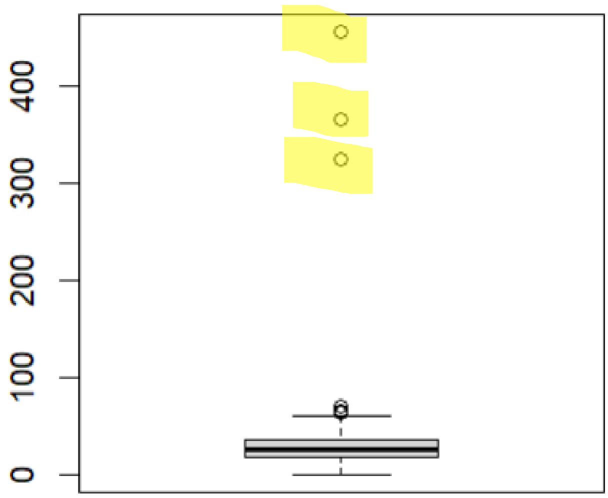
- ❑ **Detecting outliers for numeric values using visualization by box plot, scatter plot, histogram to check outliers. As outlier may appear as points far away from majority.**

Using Box plot to detect outlier for "age" variable

CODE:

```
age_plot <- boxplot(Dataset_Prac$age)

View(age_plot)
```

OUTPUT:

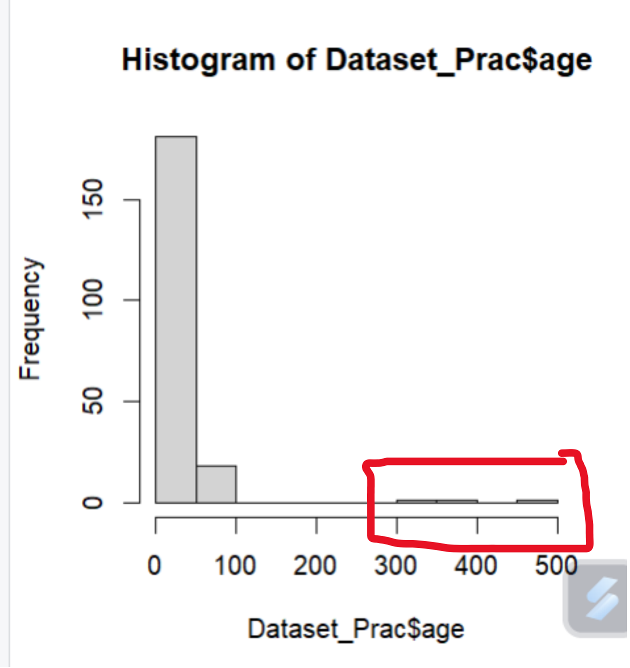


- **HISTOGRAM for "age" variable**

CODE:

hist(Dataset_Prac$age)

OUTPUT:

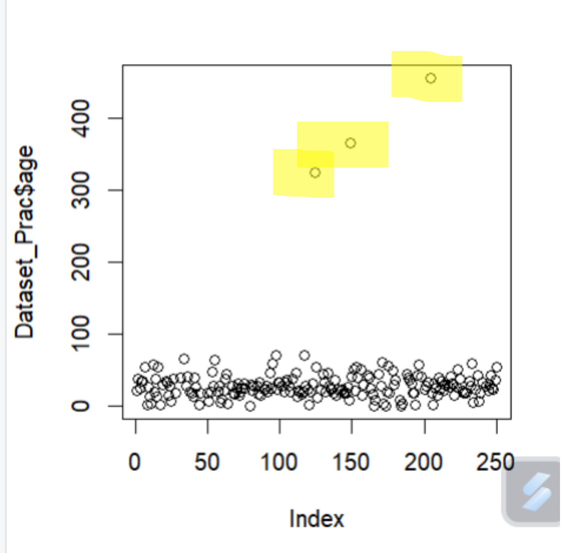


- **Scatter plot for "age"**

Code:

plot(Dataset_Prac$age)

OUTPUT:



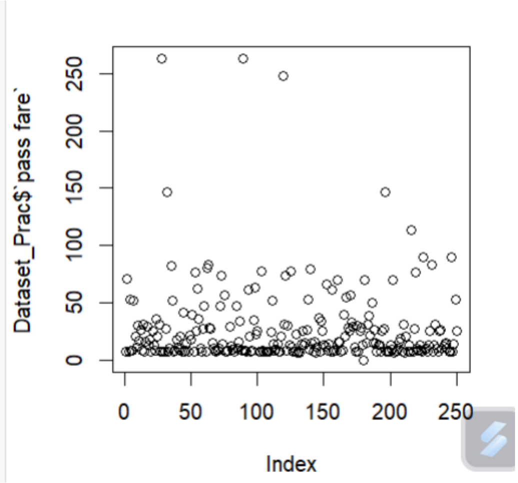
By seeing box plot, histogram and scatter plot for age variable, we can say there are some outliers, as the values are more than 100 in the age variable. So, outlier detected.

- **Scatter plot for "pass fare" variable**

CODE:

```
plot(Dataset_Prac$`pass fare`)
```

OUTPUT:

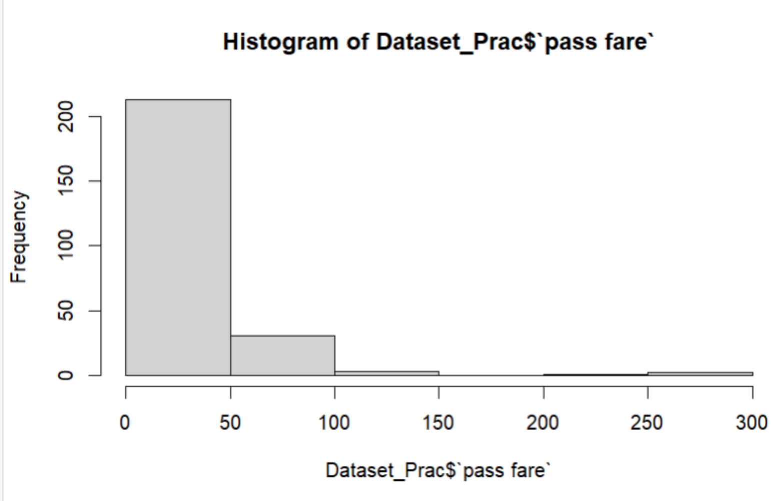


- **Histogram for "pass fare"**

CODE:

```
hist(Dataset_Prac$`pass fare`)
```

OUTPUT:



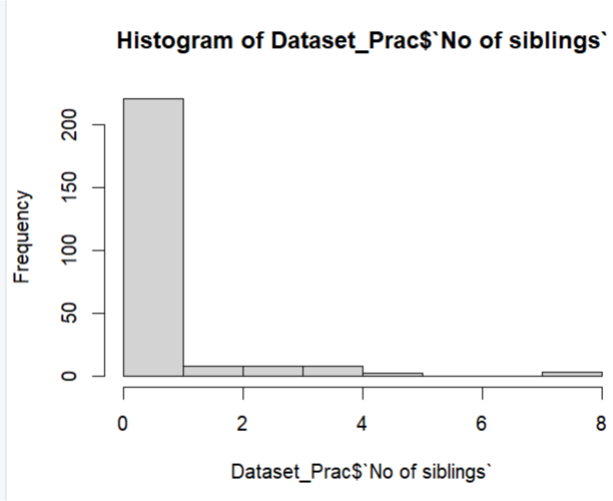
As fare can be varied for different classes and facilities, so the diversity can be ignored here and we can say, there is no outlier either.

- **Histogram for "No of Siblings" variable**

CODE:

```
hist(Dataset_Prac$`No of siblings`)
```

OUTPUT:



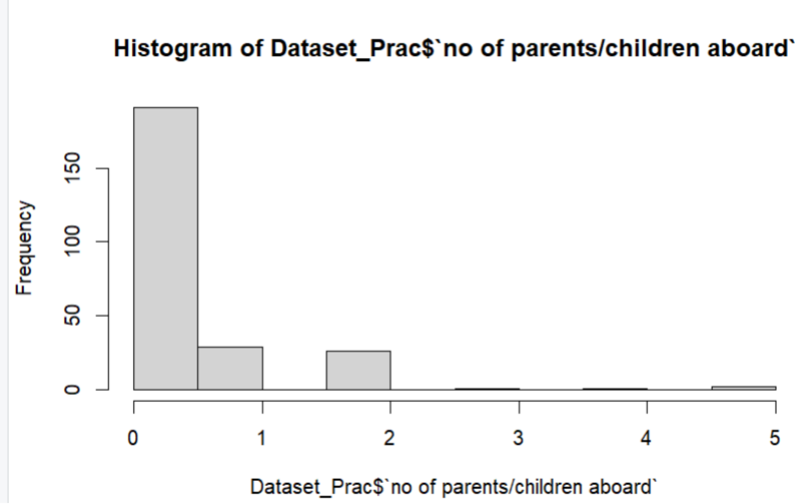
This number can also vary for person to person. So, not outlier detected.

- **Histogram for "no of parents/children aboard" variable**

CODE:

```
hist(Dataset_Prac$`no of parents/children aboard`)
```

OUTPUT:



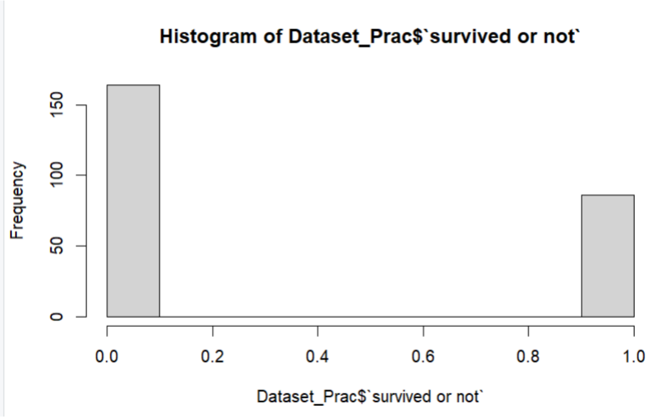
As this number can also vary for person to person, so, no outlier detected here.

- **HISTOGRAM for "survived or not" variable**

<u>CODE:</u>

```
hist(Dataset_Prac$`survived or not`)
```

<u>OUTPUT:</u>



This also satisfied, no outlier detection. Because survival was defined only by 0 and 1, and only these values are here.

- ❑ **HANDLING OUTLIERS. As I have detected outliers for 2 variables, 1) age, 2) pass was alone or not? Now I will handle outliers using different methods.**

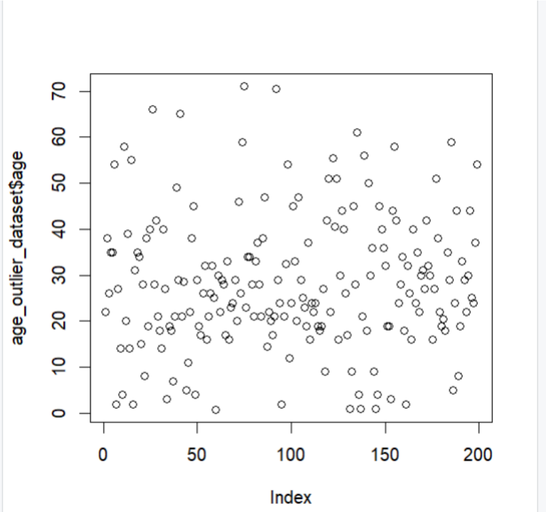
As I have discussed and detect outliers in "age" variable before. Now handling outlier in "age" variable.

- **Outlier handle by removing the rows with age more than 100 and keeping in separate dataset**

<u>CODE:</u>

```
age_outlier_dataset<-Dataset_Prac

age_outlier_dataset <- subset(age_outlier_dataset, age <= 100)

plot(age_outlier_dataset$age)

boxplot(age_outlier_dataset$age)

View(age_outlier_dataset)
```

<u>OUTPUT:</u>



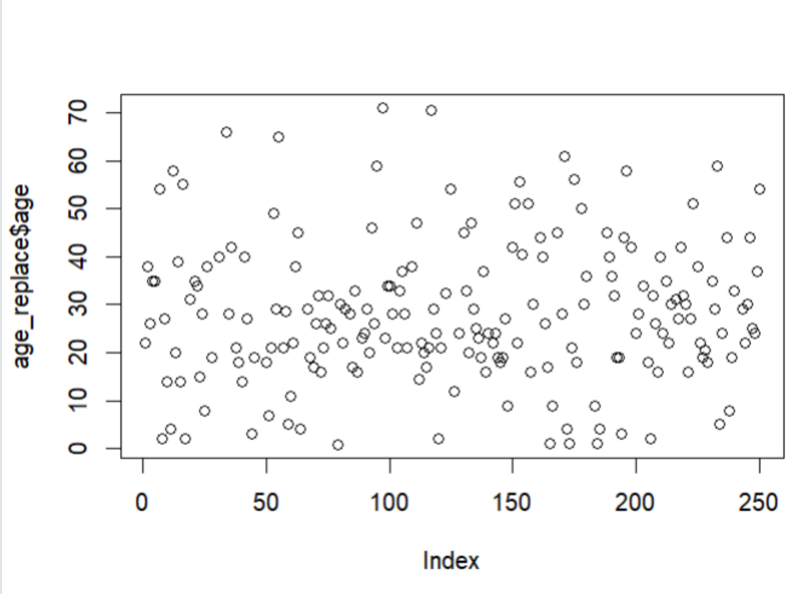
Here more than 100 years values are deleted. So, outlier handled properly.

Replacing more than 100 values with the mean value of age

<u>CODE:</u>

```
age_replace <- Dataset_Prac
age_replace$age[age_replace$age > 100] <- mean(age_replace$age)
View(age_replace)
plot(age_replace$age)
```

<u>OUTPUT:</u>



Taken the mean values to replace the outliers

- **Handling outlier in "pass was alone or not"**

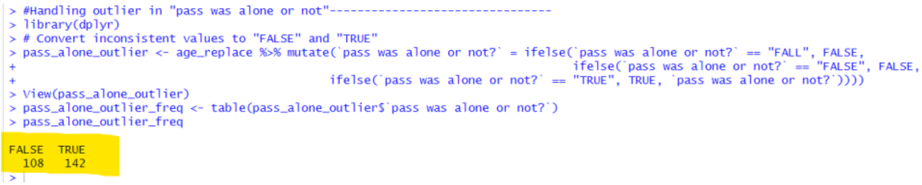
Converted inconsistent values to "FALSE" and "TRUE" to handle the outlier value "FAL"

CODE:

```
library(dplyr)

pass_alone_outlier <- age_replace %>% mutate(`pass was alone or not?` = ifelse(`pass was alone or not?` == "FALL", FALSE, ifelse(`pass was alone or not?` == "FALSE", FALSE,

                                        ifelse(`pass was alone or not?` == "TRUE", TRUE, `pass was alone or not?`))))

pass_alone_outlier_freq <- table(pass_alone_outlier$`pass was alone or not?`)

pass_alone_outlier_freq
```

And again, viewing the frequency table to check whether it worked or not!

OUTPUT:



```
> #Handling outlier in "pass was alone or not"--------------------------------
> library(dplyr)
> # Convert inconsistent values to "FALSE" and "TRUE"
> pass_alone_outlier <- age_replace %>% mutate(`pass was alone or not?` = ifelse(`pass was alone or not?` == "FALL", FALSE,
+                                                                    ifelse(`pass was alone or not?` == "FALSE", FALSE,
+                                             ifelse(`pass was alone or not?` == "TRUE", TRUE, `pass was alone or not?`))))
> View(pass_alone_outlier)
> pass_alone_outlier_freq <- table(pass_alone_outlier$`pass was alone or not?`)
> pass_alone_outlier_freq

FALSE  TRUE 
  108   142 
> 
```

Outlier handled, as there are no invalid values.

❑ **HANDLED MISSING VALUES. As I have found the missing values of each column before. Now Handling those missing values using different approaches.**

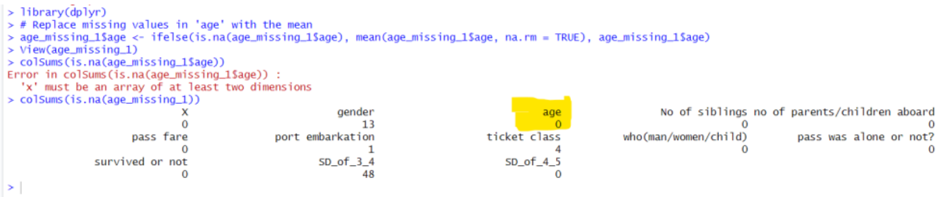
Handled missing values for age variable by replacing missing values in 'age' with the mean

CODE:

```
age_missing_1 <- Dataset_Prac

library(dplyr)

age_missing_1$age <- ifelse(is.na(age_missing_1$age), mean(age_missing_1$age, na.rm = TRUE), age_missing_1$age)

colSums(is.na(age_missing_1))
```

And viewing if there is any missing value remaining in this column

Output:



```
> library(dplyr)
> # Replace missing values in 'age' with the mean
> age_missing_1$age <- ifelse(is.na(age_missing_1$age), mean(age_missing_1$age, na.rm = TRUE), age_missing_1$age)
> View(age_missing_1)
> colSums(is.na(age_missing_1$age))
Error in colSums(is.na(age_missing_1$age)) : 
  'x' must be an array of at least two dimensions
> colSums(is.na(age_missing_1))
                             X                        gender                           age                No of siblings no of parents/children aboard
                             0                            13                             0                             0                             0
                     pass fare              port embarkation                  ticket class          who(man/women/child)        pass was alone or not?
                             0                             1                             4                             0                             0
               survived or not                     SD_of_3_4                     SD_of_4_5
                             0                            48                             0
> |
```

No missing value in "age" variable now. So, missing value handled.

- **Handled missing values in "gender" variable by replacing with the max frequent value**

<u>CODE:</u>

gender_missing_1 <- Dataset_Prac

library(dplyr)

which(is.na(Dataset_Prac$gender))

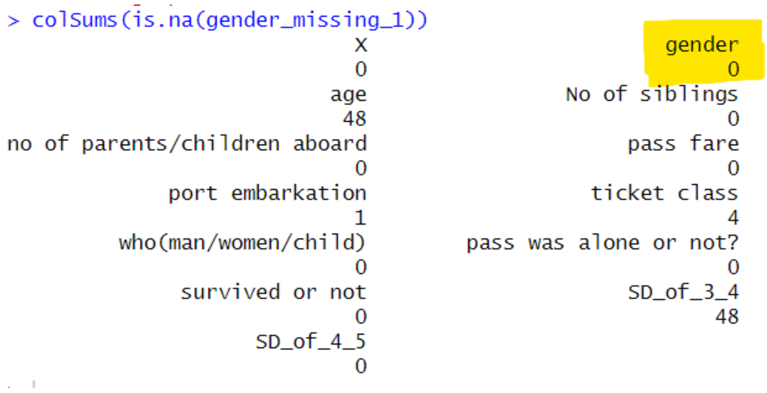
mode_gender <- names(which.max(table(gender_missing_1$gender)))

gender_missing_1$gender[is.na(gender_missing_1$gender)] <- mode_gender

colSums(is.na(gender_missing_1))

And viewing if there is any missing value remaining in this column

<u>OUTPUT:</u>



```
> colSums(is.na(gender_missing_1))
                            X                        gender
                            0                             0
                          age                No of siblings
                           48                             0
no of parents/children aboard                     pass fare
                            0                             0
             port embarkation                  ticket class
                            1                             4
         who(man/women/child)        pass was alone or not?
                            0                             0
              survived or not                     SD_of_3_4
                            0                            48
                    SD_of_4_5
                            0
```

No missing value remaining in gender column

- **Handled missing values by deleting the row with missing value**

<u>CODE:</u>

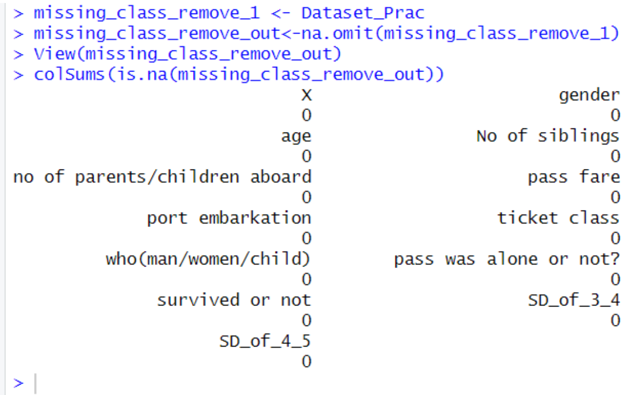
missing_class_remove_1 <- Dataset_Prac

missing_class_remove_out<-na.omit(missing_class_remove_1)

colSums(is.na(missing_class_remove_out))

And viewing if there is any missing value remaining in this column

Output:



```
> missing_class_remove_1 <- Dataset_Prac
> missing_class_remove_out<-na.omit(missing_class_remove_1)
> View(missing_class_remove_out)
> colSums(is.na(missing_class_remove_out))
                            X                        gender 
                            0                             0 
                          age                No of siblings 
                            0                             0 
no of parents/children aboard                     pass fare 
                            0                             0 
              port embarkation                  ticket class 
                            0                             0 
          who(man/women/child)           pass was alone or not? 
                            0                             0 
              survived or not                      SD_of_3_4 
                            0                             0 
                    SD_of_4_5 
                            0 
> 
```

- **Replace missing values in 'port embarkation' with the mode or most frequent value**

CODE:

missing_embark_1 <- Dataset_Prac

mode_port_embarkation <- names(which.max(table(missing_embark_1$`port embarkation`)))

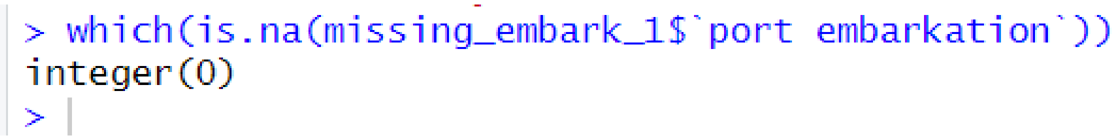
missing_embark_1$`port embarkation`[is.na(missing_embark_1$`port embarkation`)] <- mode_port_embarkation

View(missing_embark_1)

which(is.na(missing_embark_1$`port embarkation`))

And viewing if there is any missing value remaining in any row

OUTPUT:



```
> which(is.na(missing_embark_1$`port embarkation`))
integer(0)
> 
```

- ❑ **Till now I have handled outliers and missing values by exploring and creating different datasets and make them to view easily and for less complexity. Now I will change the main dataset and performed all the DATA PREPARATION part all together here.**

- **Handling Missing Values**

CODE:

For age variable

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

For gender variable

Dataset_Prac$gender[is.na(Dataset_Prac$gender)] <- mode_gender

For ticket class variable

tic_class <- names(which.max(table(Dataset_Prac$`ticket class`)))

Dataset_Prac$`ticket class`[is.na(Dataset_Prac$`ticket class`)] <- tic_class

For port embarkation variable

Dataset_Prac$`port embarkation`[is.na(Dataset_Prac$`port embarkation`)] <- mode_port_embarkation

Example of Deleting attribute with missing value (another way of handle missing value)

Dataset_Prac <- subset(Dataset_Prac, select = -SD_of_3_4)

Dataset_Prac <- subset(Dataset_Prac, select = -SD_of_4_5)

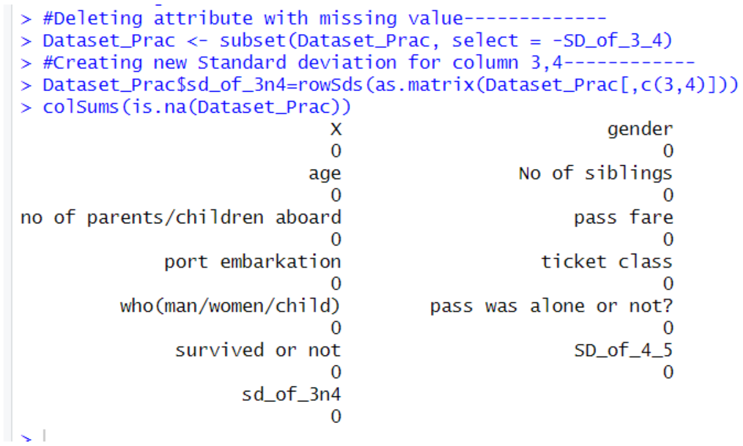
Creating new Standard deviation for column 3,4, as the missing values are handled now

Dataset_Prac$sd_of_3n4=rowSds(as.matrix(Dataset_Prac[,c(3,4)]))

Dataset_Prac$sd_of_4n5=rowSds(as.matrix(Dataset_Prac[,c(4,5)]))

colSums(is.na(Dataset_Prac))

<u>OUTPUT:</u>



```
> #Deleting attribute with missing value-------------
> Dataset_Prac <- subset(Dataset_Prac, select = -SD_of_3_4)
> #Creating new Standard deviation for column 3,4------------
> Dataset_Prac$sd_of_3n4=rowSds(as.matrix(Dataset_Prac[,c(3,4)]))
> colSums(is.na(Dataset_Prac))
                            X                        gender 
                            0                             0 
                          age                No of siblings 
                            0                             0 
no of parents/children aboard                     pass fare 
                            0                             0 
             port embarkation                  ticket class 
                            0                             0 
         who(man/women/child)        pass was alone or not? 
                            0                             0 
              survived or not                      SD_of_4_5 
                            0                             0 
                    sd_of_3n4 
                            0 
> 
```

No missing value remaining in main dataset

- ❑ **Handling previously found outlier in main dataset, without missing values now**

<u>CODE:</u>

For age variable

Replacing more than 100 years values with age mean value

```
Dataset_Prac$age[Dataset_Prac$age > 100] <- mean(Dataset_Prac$age)
```

```
plot(Dataset_Prac$age)
```

For pass was alone or not variable

Convert inconsistent values to "FALSE" and "TRUE"

```
Dataset_Prac <- Dataset_Prac %>% mutate(`pass was alone or not?` = ifelse(`pass was alone or not?` == "FALL", FALSE, ifelse(`pass was alone or not?` == "FALSE", FALSE,
                                                                    ifelse(`pass was alone or not?` == "TRUE", TRUE, `pass was alone or not?`))))
```

```
Dataset_Prac_freq <- table(Dataset_Prac$`pass was alone or not?`)
```

```
Dataset_Prac_freq
```

□ **Getting rid of noisy values as part of data preparation**

CODE:

Round the values in 'pass fare' to the nearest whole number

```
Dataset_Prac$`pass fare` <- round(Dataset_Prac$`pass fare`)
```

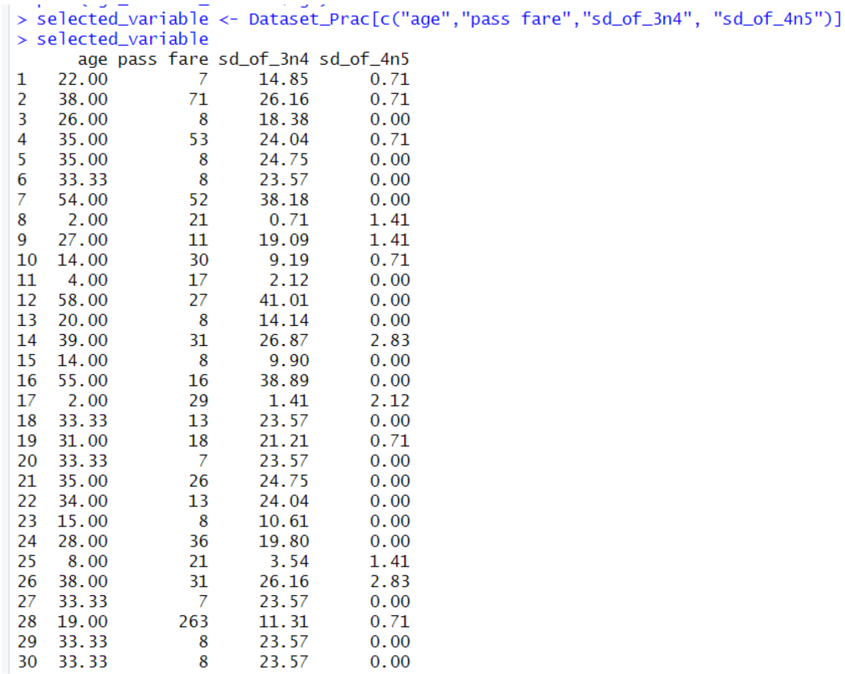
Round the values in "age", 'sd_of_3n4' and "sd_of_4_5" up to two decimal place number

```
Dataset_Prac$sd_of_3n4 <- round(Dataset_Prac$sd_of_3n4, 2)

Dataset_Prac$sd_of_4n5 <- round(Dataset_Prac$sd_of_4n5, 2)

Dataset_Prac$age <- round(Dataset_Prac$age, 2)

View(Dataset_Prac)
```

Viewing specific selected column for the previously done noisy value handle

```
selected_variable <- Dataset_Prac[c("age","pass fare","sd_of_3n4", "sd_of_4n5")]

selected_variable
```

OUTPUT:



```
> selected_variable <- Dataset_Prac[c("age","pass fare","sd_of_3n4", "sd_of_4n5")]
> selected_variable
      age pass fare sd_of_3n4 sd_of_4n5
1   22.00         7     14.85      0.71
2   38.00        71     26.16      0.71
3   26.00         8     18.38      0.00
4   35.00        53     24.04      0.71
5   35.00         8     24.75      0.00
6   33.33         8     23.57      0.00
7   54.00        52     38.18      0.00
8    2.00        21      0.71      1.41
9   27.00        11     19.09      1.41
10  14.00        30      9.19      0.71
11   4.00        17      2.12      0.00
12  58.00        27     41.01      0.00
13  20.00         8     14.14      0.00
14  39.00        31     26.87      2.83
15  14.00         8      9.90      0.00
16  55.00        16     38.89      0.00
17   2.00        29      1.41      2.12
18  33.33        13     23.57      0.00
19  31.00        18     21.21      0.71
20  33.33         7     23.57      0.00
21  35.00        26     24.75      0.00
22  34.00        13     24.04      0.00
23  15.00         8     10.61      0.00
24  28.00        36     19.80      0.00
25   8.00        21      3.54      1.41
26  38.00        31     26.16      2.83
27  33.33         7     23.57      0.00
28  19.00       263     11.31      0.71
29  33.33         8     23.57      0.00
30  33.33         8     23.57      0.00
```

- ❑ **Doing some exploration for the updated main dataset with no missing values and no outliers.**
- ❑ **Finding mean, mode and median of all columns for both numeric and categorical data and plotting them.**

CODE:

```
numericcols <- Dataset_Prac[, sapply(Dataset_Prac, is.numeric)]

categoricalcols <- Dataset_Prac[, !sapply(Dataset_Prac, is.numeric)]
```

Calculating mean

```
meanvalues_numeric <- colMeans(numericcols)

meanvalues_categorical <- sapply(categoricalcols, function(x) length(unique(x)) / length(x))
```

Calculating mode

```
modevalues_categorical <- sapply(categoricalcols, function(x) {
  unique_values <- unique(x)
  unique_values[which.max(tabulate(match(x, unique_values)))]})


modevalues_numeric <- sapply(numericcols, function(x) {
  unique_values <- unique(x)
  unique_values[which.max(tabulate(match(x, unique_values)))]})
```

Calculating median

```
medianvalues_numeric <- sapply(numericcols, median, na.rm = TRUE)
```

Combining numerical and categorical values for all columns

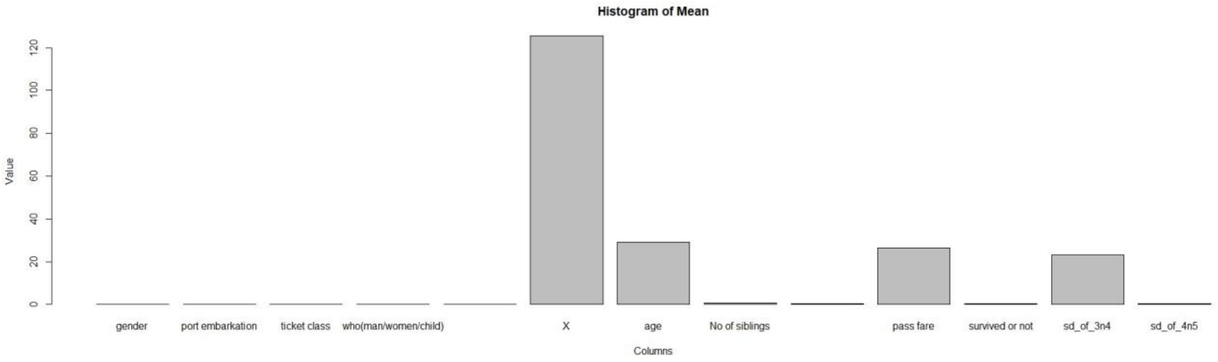
```
fmeanvalues <- c(meanvalues_categorical, meanvalues_numeric)
fmodevalues <- c(modevalues_categorical, modevalues_numeric)
```

Plotting the mean, mode and median values

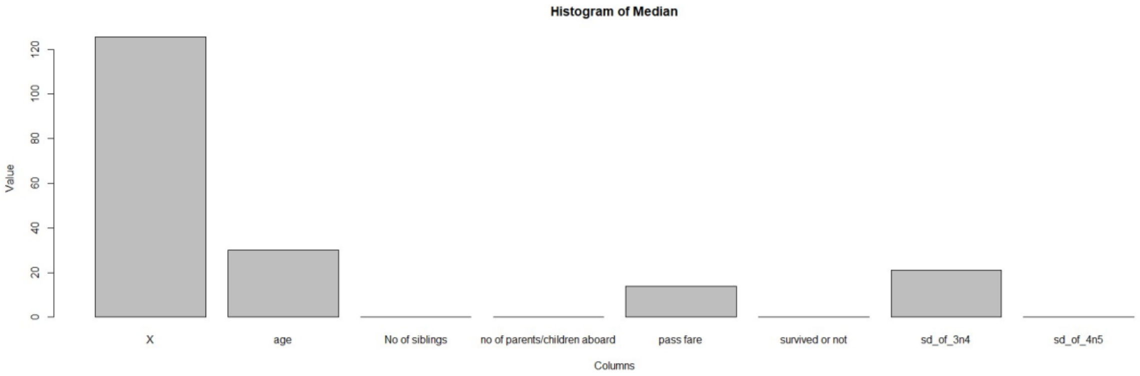
```
barplot(fmeanvalues, main = "Histogram of Mean", xlab = "Columns", ylab = "Value")
barplot(fmodevalues, main = "Histogram of Mode", xlab = "Columns", ylab = "Value")
barplot(fmedianvalues, main = "Histogram of Median", xlab = "Columns", ylab = "Value")
```

<u>OUTPUT:</u>

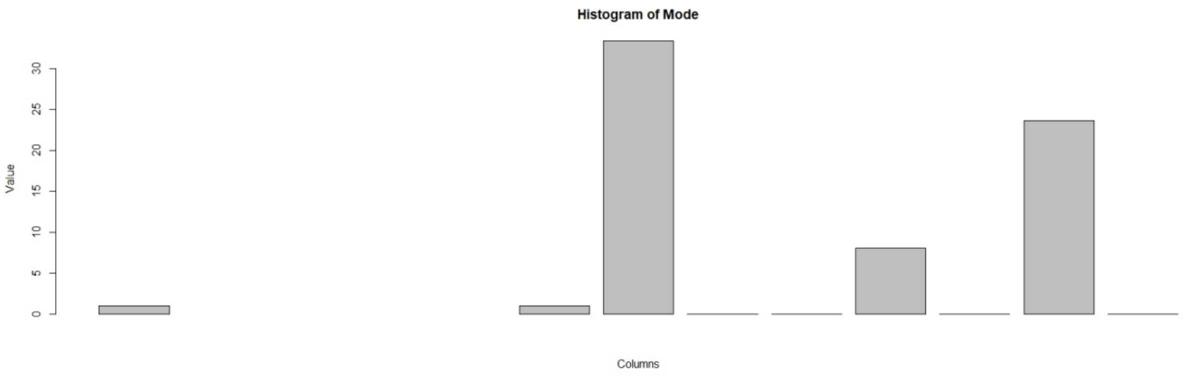
<u>MEAN</u>



<u>MEDIAN:</u>



<u>MODE:</u>

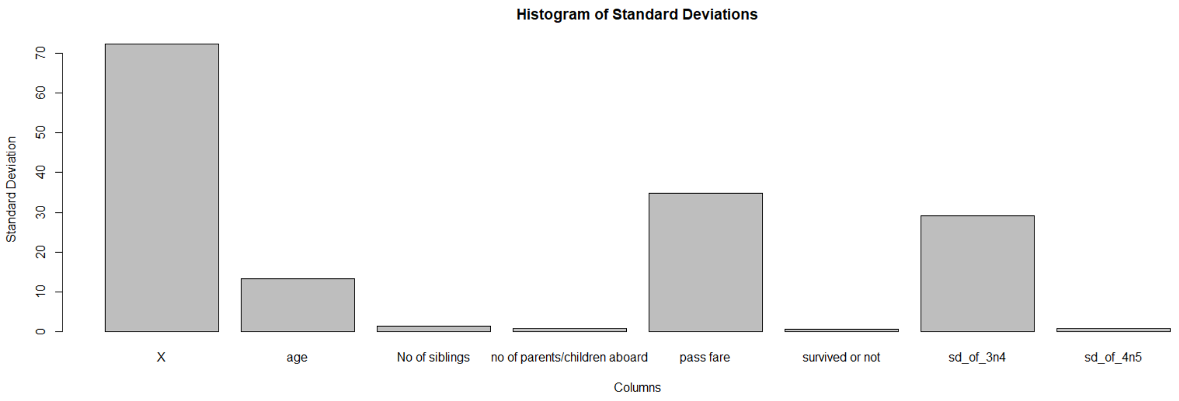


- □ **Calculate the standard deviation of numeric columns and plotting histogram of standard deviation values**

CODE:

```
library(dplyr)

main_sd <- Dataset_Prac %>% summarise_if(is.numeric, sd)

main_sd_values <- unlist(main_sd)

barplot(main_sd_values, main = "Histogram of Standard Deviations", xlab = "Columns", ylab = "Standard Deviation")
```

OUTPUT:


Histogram of Standard Deviations
Standard Deviation
X
age
No of siblings
no of parents/children aboard
pass fare
survived or not
sd_of_3n4
sd_of_4n5
Columns