American International University- Bangladesh

Data Science

Project Report

Summer 22-23

Name : Nayeem Abdul Qaiyum
ID : 20-43581-1
Section : C

Date Of Submit: 18.07.23

Submitted By

Abdus Salam
Assistant Professor, CS

Description:
The Titanic dataset is a widely-known data science collection of information about passengers aboard the Titanic, including their age, gender, siblings, parents/children, passenger fare, Port of Embarkation, ticket class, categories, and survival status. The dataset contains numerous rows and 10 columns, with some data points missing. It includes integer, numeric, and character attributes, and the goal is to obtain a clean preprocessed dataset. The dataset includes various types of passengers, including man, women, children, and those who were alone or not.

Table of Contents

Project Solution

Import data:

Explanation: Insert all of the data from the excel file first, and then save the document as a dataset file. then convert the dataset file's format to a CSV file. After importing my CSV file into RStudio, I add the following code.

Code Segment:

```
dataset <- read.csv("D:/11th Semester/Data Science/Mid Project/titanic.csv")
dataset
```

Output:

```
> dataset <- read.csv("D:/11th Semester/Data Science/Mid Project/titanic.csv")
>   View(dataset)
> dataset
    gender   age sibsp parch     fare embarked  class   who alone survived
1        0 22.00     1     0   7.2500        S  Third   man FALSE        0
2        1 38.00     1     0  71.2833        C  First woman  FALL        1
3        1 26.00     0     0   7.9250        S  Third woman  TRUE        1
4        1 35.00     1     0  53.1000        S  First woman  FALL        1
5        0 35.00     0     0   8.0500        S  Third   man  TRUE        0
6        0    NA     0     0   8.4583        Q  Third   man  TRUE        0
7        0 54.00     0     0  51.8625        S  First   man  TRUE        0
8        0  2.00     3     1  21.0750        S  Third child FALSE        0
9        1 27.00     0     2  11.1333        S  Third woman FALSE        1
10       1 14.00     1     0  30.0708        C Second child FALSE        1
11       1  4.00     1     1  16.7000        S  Third child FALSE        1
12       1 58.00     0     0  26.5500        S  First woman  TRUE        1
13      NA 20.00     0     0   8.0500        S  Third   man  TRUE        0
14       0 39.00     1     5  31.2750        S  Third   man FALSE        0
15       1 14.00     0     0   7.8542        S  Third child  TRUE        0
16       1 55.00     0     0  16.0000        S Second woman  TRUE        1
17       0  2.00     4     1  29.1250        Q  Third child FALSE        0
18       0    NA     0     0  13.0000        S Second   man  TRUE        1
19       1 31.00     1     0  18.0000        S  Third woman FALSE        0
20       1    NA     0     0   7.2250        C  Third woman  TRUE        1
21       0 35.00     0     0  26.0000        S Second   man  TRUE        0
22       0 34.00     0     0  13.0000        S Second   man  TRUE        1
23       1 15.00     0     0   8.0292        Q  Third child  TRUE        1
24       0 28.00     0     0  35.5000        S  First   man  TRUE        1
25       1  8.00     3     1  21.0750        S        child FALSE        0
26       1 38.00     1     5  31.3875        S  Third woman FALSE        1
27       0    NA     0     0   7.2250        C  Third   man  TRUE        0
```

Find the shape of the dataset:

Code Segment:

```
nrow(dataset)
```

Output:

```
> #Find the shape of the data set
> nrow(dataset)
[1] 250
```

Code Segment:

```
ncol(dataset)
```

Output:

```
> ncol(dataset)
[1] 10
>
```

Code Segment:

length(dataset)

Output:

```
> length(dataset)
[1] 10
>
```

Show the attributes names of the dataset:

Code Segment:

names(dataset)

Output:

```
> names(dataset)
 [1] "gender"   "age"      "sibsp"    "parch"    "fare"     "embarked" "class"    "who"
 [9] "alone"    "survived"
>
```

Find the structure of the dataset:

Explanation: The dataset's structure, including the variables, their data types, and the initial values, is shown using the str() function. This will provide us with a summary of the dataset.

Code Segment:

str(dataset)

Output:

```
> #Find the types of data for all attributes
> str(dataset)
'data.frame':   250 obs. of  10 variables:
 $ gender  : int  0 1 1 1 0 0 0 0 1 1 ...
 $ age     : num  22 38 26 35 35 NA 54 2 27 14 ...
 $ sibsp   : int  1 1 0 1 0 0 0 3 0 1 ...
 $ parch   : int  0 0 0 0 0 0 0 1 2 0 ...
 $ fare    : num  7.25 71.28 7.92 53.1 8.05 ...
 $ embarked: chr  "S" "C" "S" "S" ...
 $ class   : chr  "Third" "First" "Third" "First" ...
 $ who     : chr  "man" "woman" "woman" "woman" ...
 $ alone   : chr  "FALSE" "FALL" "TRUE" "FALL" ...
 $ survived: int  0 1 1 1 0 0 0 0 1 1 ...
>
```

The first few rows of dataset:

Explanation: The first few rows of the dataset are shown by the head() function. This will allow us to understand the data and ensure that it was imported properly.

Code Segment:

```
head(dataset)
```

Output:

```
> #The first few rows of data set
> head(dataset)
  gender age sibsp parch    fare embarked class   who alone survived
1      0  22     1     0  7.2500        S Third   man FALSE        0
2      1  38     1     0 71.2833        C First woman  FALL        1
3      1  26     0     0  7.9250        S Third woman  TRUE        1
4      1  35     1     0 53.1000        S First woman  FALL        1
5      0  35     0     0  8.0500        S Third   man  TRUE        0
6      0  NA     0     0  8.4583        Q Third   man  TRUE        0
>
```

Summary of the dataset:

Explanation: For numerical variables in the dataset, summary statistics (count, mean, median, etc.) are provided using the summary() function. This will give us insights into the distribution and central tendencies of the variables.

Code Segment:

```
summary(dataset)
```

Output:

```
> #Summary of the data set
> summary(dataset)
     gender            age            sibsp           parch            fare           embarked
 Min.   :0.0000   Min.   :  0.83   Min.   :0.000   Min.   :0.000   Min.   :  0.000   Length:250
 1st Qu.:0.0000   1st Qu.: 19.00   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:  8.034   Class :character
 Median :0.0000   Median : 27.00   Median :0.000   Median :0.000   Median : 13.977   Mode  :character
 Mean   :0.3629   Mean   : 33.33   Mean   :0.656   Mean   :0.392   Mean   : 26.588
 3rd Qu.:1.0000   3rd Qu.: 37.00   3rd Qu.:1.000   3rd Qu.:0.000   3rd Qu.: 29.094
 Max.   :1.0000   Max.   :455.00   Max.   :8.000   Max.   :5.000   Max.   :263.000
 NA's   :13       NA's   :48
    class               who              alone             survived
 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
```

Find the type of attribute:

Explanation: We can identify the type of a column by using sapply.

Code Segment:

```
sapply(dataset, class)
```

Output:

```
> #Find the type of attribute
> sapply(dataset, class)
    gender        age      sibsp      parch       fare   embarked      class        who      alone   survived
 "integer"   "numeric"  "integer"  "integer"  "numeric" "character" "character" "character" "character"  "integer"
>
```

Measure of spread range and standard deviation:

Explanation: The mean gives the average value, the median shows the middle value, and the mode shows the value which occurs most frequently for each feature. These measurements help me understand the dataset's distribution and usual values, which I can use to draw inferences and make comparisons as part of our research.

Code Segment:

For Gender:

```
gender_range <- range(dataset$gender, na.rm = TRUE)

print(gender_range)

gender_sd <- sd(dataset$gender, na.rm = TRUE)

print(gender_sd)
```

Output:

```
> gender_range <- range(dataset$gender, na.rm = TRUE)
> print(gender_range)
[1] 0 1
> gender_sd <- sd(dataset$gender, na.rm = TRUE)
> print(gender_sd)
[1] 0.4818452
```

For Age:

```
age_range <- range(dataset$age, na.rm = TRUE)

print(age_range)

age_sd <- sd(dataset$age, na.rm = TRUE)

print(age_sd)
```

Output:

```
> age_range <- range(dataset$age, na.rm = TRUE)
> print(age_range)
[1]   0.83 455.00
> age_sd <- sd(dataset$age, na.rm = TRUE)
> print(age_sd)
[1] 45.7735
>
```

For sibsp:

```
sibsp_range <- range(dataset$sibsp, na.rm = TRUE)
print(sibsp_range)
sibsp_sd <- sd(dataset$sibsp, na.rm = TRUE)
print(sibsp_sd)
```

Output:

```
> sibsp_range <- range(dataset$sibsp, na.rm = TRUE)
> print(sibsp_range)
[1] 0 8
> sibsp_sd <- sd(dataset$sibsp, na.rm = TRUE)
> print(sibsp_sd)
[1] 1.305558
>
```

For parch:

```
parch_range <- range(dataset$parch, na.rm = TRUE)
print(parch_range)
parch_sd <- sd(dataset$parch, na.rm = TRUE)
print(parch_sd)
```

Output:

```
> parch_range <- range(dataset$parch, na.rm = TRUE)
> print(parch_range)
[1] 0 5
> parch_sd <- sd(dataset$parch, na.rm = TRUE)
> print(parch_sd)
[1] 0.8252637
>
```

For fare:

```
fare_range <- range(dataset$fare, na.rm = TRUE)
print(fare_range)
fare_sd <- sd(dataset$fare, na.rm = TRUE)
print(fare_sd)
```

Output:

```
> fare_range <- range(dataset$fare, na.rm = TRUE)
> print(fare_range)
[1]   0 263
> fare_sd <- sd(dataset$fare, na.rm = TRUE)
> print(fare_sd)
[1] 34.82165
>
```

For Survived:

```
survived_range <- range(dataset$survived, na.rm = TRUE)
print(survived_range)
survived_sd <- sd(dataset$fsurvived, na.rm = TRUE)
print(survived_sd)
```

Output:

```
> survived_range <- range(dataset$survived, na.rm = TRUE)
> print(survived_range)
[1] 0 1
> survived_sd <- sd(dataset$fsurvived, na.rm = TRUE)
> print(survived_sd)
[1] NA
```

Visualization:

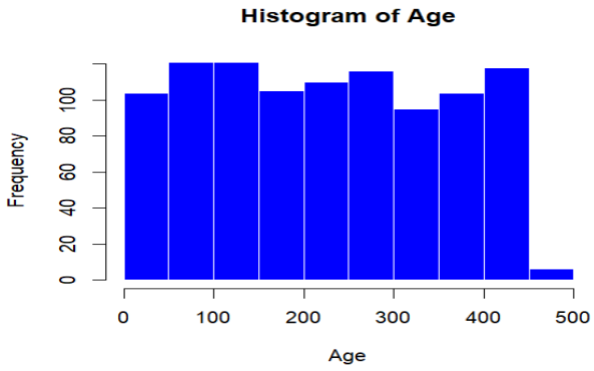
Explanation: Standard deviation measures the difference in a data set from the mean, with high deviation indicating wide data points and low deviation indicating closer points. Here I also create histogram for

For age:

Code Segment:

```
mean_val <- 33.33

sd_val <- 45.7735

age_range <- c(0.83, 455)

age_data <- runif(1000, min = age_range[1], max = age_range[2])

hist(age_data,

    main = "Histogram of Age",

    xlab = "Age", ylab = "Frequency",

    col = "blue", border = "white")
```

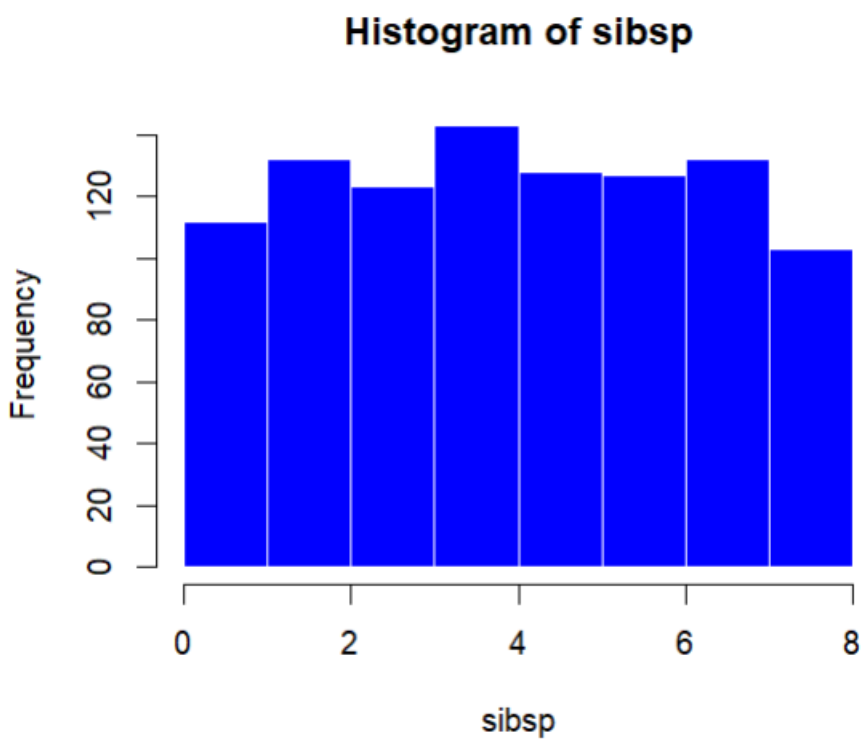
Output:



For sibsp:

```
mean_val <- 0.656

sd_val <- 1.305558

sibsp_range <- c(0, 8)

sibsp_data <- runif(1000, min = sibsp_range[1], max = sibsp_range[2])


hist(sibsp_data,
    main = "Histogram of sibsp",
    xlab = "sibsp", ylab = "Frequency",
    col = "blue", border = "white")
```

Output:

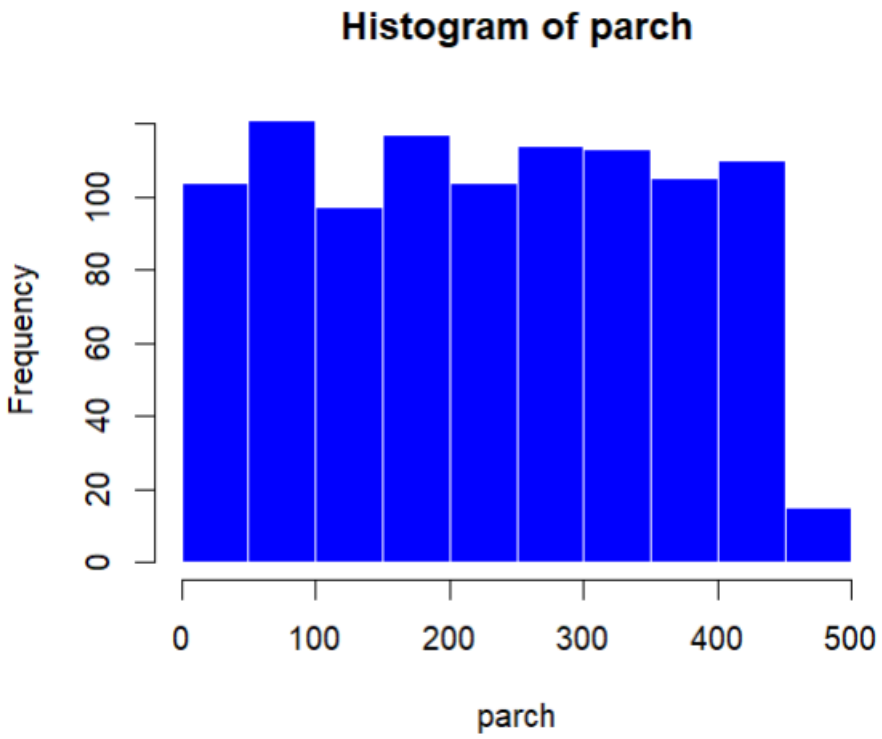


For parch:

```
mean_val <- 0.392

sd_val <- 0.8252637
```

```
parch_range <- c(0, 5)
parch_data <- runif(1000, min = parch_range[1], max = parch_range[2])

hist(age_data,
    main = "Histogram of parch",
    xlab = "parch", ylab = "Frequency",
    col = "blue", border = "white")
```

Output:

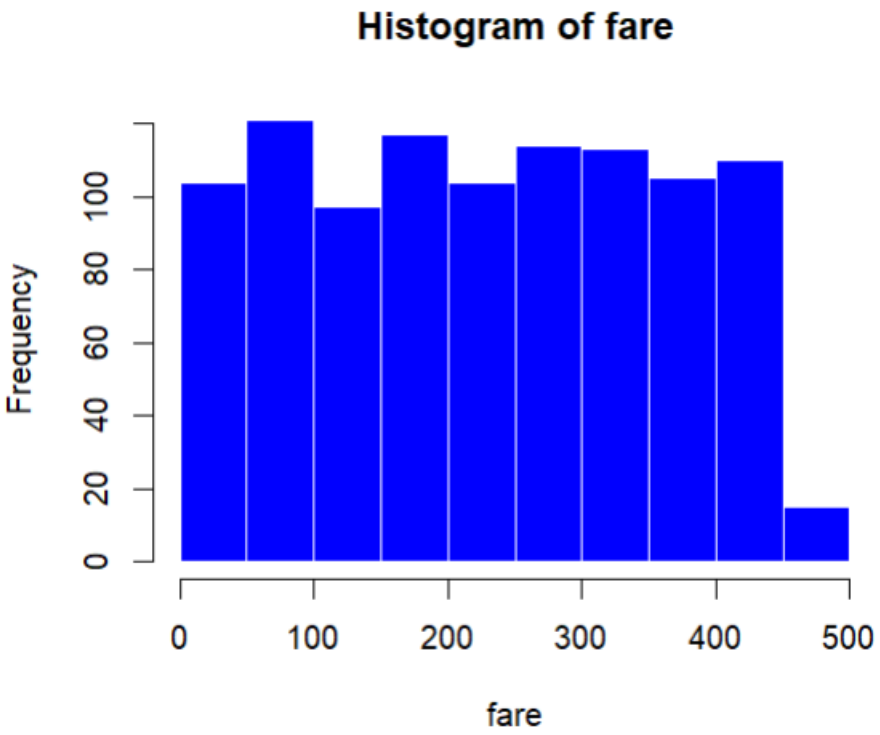


For fare:

```
mean_val <- 26.588
sd_val <- 34.82165
fare_range <- c(0, 263)
fare_data <- runif(1000, min =fare_range[1], max = fare_range[2])
```

```
hist(age_data,
    main = "Histogram of fare",
    xlab = "fare", ylab = "Frequency",
    col = "blue", border = "white")
```

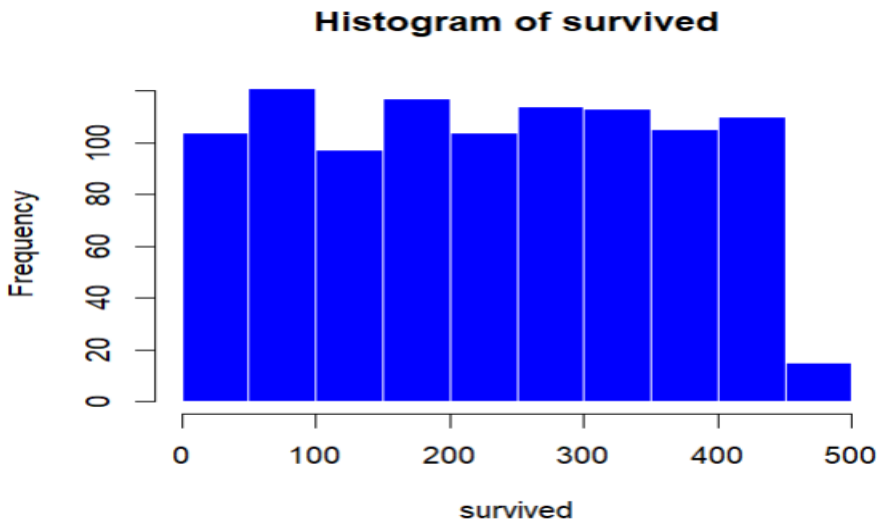
Output:



For survived:

```
mean_val <- 0.344
sd_val <- NA
survived_range <- c(0, 1)
survived_data <- runif(1000, min =survived_range[1], max = survived_range[2])

hist(age_data,
    main = "Histogram of survived",
    xlab = "survived", ylab = "Frequency",
    col = "blue", border = "white")
```

Output:



Find the missing value for all column:

Explanation: It's crucial to identify and handle missing values in a dataset since they can introduce biased and affect the accuracy of our research and findings. I can identify the missing values in every column using the following code.

To extract the rows with missing values from the "dataset" dataset, use the code "dataset[!complete.cases(dataset),]".

Code Segment:

```
number_of_missing_value=colSums(is.na(dataset))
number_of_missing_value
```

Output:

```
> number_of_missing_value=colSums(is.na(dataset))
> number_of_missing_value
  gender      age    sibsp    parch     fare embarked    class      who    alone survived
      13       48        0        0        0        0        0        0        0        0
>
> dataset[!complete.cases(dataset),]
    gender age sibsp parch     fare embarked  class   who alone survived
6        0  NA     0     0   8.4583        Q  Third   man  TRUE        0
13      NA  20     0     0   8.0500        S  Third   man  TRUE        0
18       0  NA     0     0  13.0000        S Second   man  TRUE        1
20       1  NA     0     0   7.2250        C  Third woman  TRUE        1
27       0  NA     0     0   7.2250        C  Third   man  TRUE        0
29       1  NA     0     0   7.8792        Q  Third woman  TRUE        1
30       0  NA     0     0   7.8958        S  Third   man  TRUE        0
32       1  NA     1     0 146.5208        C  First woman FALSE        1
33       1  NA     0     0   7.7500        Q  Third woman  TRUE        1
34      NA  66     0     0  10.5000        S Second   man  TRUE        0
37       0  NA     0     0   7.2292        C  Third   man  TRUE        1
43       0  NA     0     0   7.8958        C  Third   man  TRUE        0
46       0  NA     0     0   8.0500        S  Third   man  TRUE        0
47       0  NA     1     0  15.5000        Q  Third   man FALSE        0
48       1  NA     0     0   7.7500        Q  Third woman  TRUE        1
49       0  NA     2     0  21.6792        C  Third   man FALSE        0
52      NA  21     0     0   7.8000        S  Third   man  TRUE        0
56      NA  NA     0     0  35.5000        S  First   man  TRUE        1
65       0  NA     0     0  27.7208        C  First   man  TRUE        0
66       0  NA     1     1  15.2458        C  Third   man FALSE        1
77      NA  NA     0     0   7.8958        S  Third   man  TRUE        0
78       0  NA     0     0   8.0500        S  Third   man  TRUE        0
83       1  NA     0     0   7.7875        Q  Third woman  TRUE        1
88       0  NA     0     0   8.0500        S  Third   man  TRUE        0
96       0  NA     0     0   8.0500        S  Third   man  TRUE        0
98      NA  23     0     1  63.3583        C  First   man FALSE        1
102      0  NA     0     0   7.8958        S  Third   man  TRUE        0
108      0  NA     0     0   7.7750        S  Third   man  TRUE        1
109     NA  38     0     0   7.8958        S  Third   man  TRUE        0
110      1  NA     1     0  24.1500        Q  Third woman FALSE        1
```

Missing value of Gender & Age:

Explanation: I can identify which rows are missing values by using the code below.

Code Segment:

```
missing_gender=which(is.na(dataset$gender))
missing_gender
missing_age=which(is.na(dataset$age))
missing_age
```

Output:

```
> missing_gender=which(is.na(dataset$gender))
> missing_gender
 [1]  13  34  52  56  77  98 109 135 177 194 210 214 246
> missing_age=which(is.na(dataset$age))
> missing_age
 [1]   6  18  20  27  29  30  32  33  37  43  46  47  48  49  56  65  66  77  78  83  88  96 102 108 110 122 127 129 141
[30] 155 159 160 167 169 177 181 182 186 187 197 199 202 215 224 230 236 241 242
>
```

Most frequent value:

Explanation: The gender column in our data set has an invalid value. We may extract the gender attribute's most frequent value using function and code.

```
find_mode <- function(x) {
  u <- unique(x)
  tab <- tabulate(match(x, u))
  u[tab == max(tab)]
}
most_frequent_gender=find_mode(dataset$gender)
most_frequent_gender
```

Output:

```
> find_mode <- function(x) {
+   u <- unique(x)
+   tab <- tabulate(match(x, u))
+   u[tab == max(tab)]
+ }
> most_frequent_gender=find_mode(dataset$gender)
> most_frequent_gender
[1] 0
>
```

Replacing missing value by most frequent value for gender attributes:

```
dataset$gender[is.na(dataset$gender)]<-most_frequent_gender
print(dataset)
```

Output:

```
> dataset$gender[is.na(dataset$gender)]<-most_frequent_gender
> print(dataset)
    gender   age sibsp parch     fare embarked  class   who alone survived
1        0 22.00     1     0   7.2500        S  Third   man FALSE        0
2        1 38.00     1     0  71.2833        C  First woman  FALL        1
3        1 26.00     0     0   7.9250        S  Third woman  TRUE        1
4        1 35.00     1     0  53.1000        S  First woman  FALL        1
5        0 35.00     0     0   8.0500        S  Third   man  TRUE        0
6        0    NA     0     0   8.4583        Q  Third   man  TRUE        0
7        0 54.00     0     0  51.8625        S  First   man  TRUE        0
8        0  2.00     3     1  21.0750        S  Third child FALSE        0
9        1 27.00     0     2  11.1333        S  Third woman FALSE        1
10       1 14.00     1     0  30.0708        C Second child FALSE        1
11       1  4.00     1     1  16.7000        S  Third child FALSE        1
12       1 58.00     0     0  26.5500        S  First woman  TRUE        1
13       0 20.00     0     0   8.0500        S  Third   man  TRUE        0
14       0 39.00     1     5  31.2750        S  Third   man FALSE        0
15       1 14.00     0     0   7.8542        S  Third child  TRUE        0
16       1 55.00     0     0  16.0000        S Second woman  TRUE        1
17       0  2.00     4     1  29.1250        Q  Third child FALSE        0
18       0    NA     0     0  13.0000        S Second   man  TRUE        1
19       1 31.00     1     0  18.0000        S  Third woman FALSE        0
20       1    NA     0     0   7.2250        C  Third woman  TRUE        1
21       0 35.00     0     0  26.0000        S Second   man  TRUE        0
22       0 34.00     0     0  13.0000        S Second   man  TRUE        1
23       1 15.00     0     0   8.0292        Q  Third child  TRUE        1
24       0 28.00     0     0  35.5000        S  First   man  TRUE        1
25       1  8.00     3     1  21.0750        S        child FALSE        0
26       1 38.00     1     5  31.3875        S  Third woman FALSE        1
27       0    NA     0     0   7.2250        C  Third   man  TRUE        0
28       0 19.00     3     2 263.0000        S  First   man FALSE        0
29       1    NA     0     0   7.8792        Q  Third woman  TRUE        1
30       0    NA     0     0   7.8958        S  Third   man  TRUE        0
31       0 40.00     0     0  27.7208        C  First   man  TRUE        0
32       1    NA     1     0 146.5208        C  First woman FALSE        1
33       1    NA     0     0   7.7500        Q  Third woman  TRUE        1
```

Data Cleaning:

Explanation: We may create a clean dataset by removing missing values, which will enable us to analyze our dataset more efficiently.

1. Rows with missing values should be deleted. We can remove the row of missing values by using the na.omit() function.It is a particular type of cleaning missing value.
2. Using the mean value, recover missing values.
3. Using the mode value, recover missing values.

Deleting row for clean data:

Code Segment:

```
remove_missing<-na.omit(dataset)
print(remove_missing)
```

Output:

```
> remove_missing<-na.omit(dataset)
> print(remove_missing)
    gender     age sibsp parch      fare embarked  class    who alone survived
1        0   22.00     1     0    7.2500        S  Third    man FALSE        0
2        1   38.00     1     0   71.2833        C  First woman  FALL        1
3        1   26.00     0     0    7.9250        S  Third woman  TRUE        1
4        1   35.00     1     0   53.1000        S  First woman  FALL        1
5        0   35.00     0     0    8.0500        S  Third    man  TRUE        0
7        0   54.00     0     0   51.8625        S  First    man  TRUE        0
8        0    2.00     3     1   21.0750        S  Third child FALSE        0
9        1   27.00     0     2   11.1333        S  Third woman FALSE        1
10       1   14.00     1     0   30.0708        C Second child FALSE        1
11       1    4.00     1     1   16.7000        S  Third child FALSE        1
12       1   58.00     0     0   26.5500        S  First woman  TRUE        1
13       0   20.00     0     0    8.0500        S  Third    man  TRUE        0
14       0   39.00     1     5   31.2750        S  Third    man FALSE        0
15       1   14.00     0     0    7.8542        S  Third child  TRUE        0
16       1   55.00     0     0   16.0000        S Second woman  TRUE        1
17       0    2.00     4     1   29.1250        Q  Third child FALSE        0
19       1   31.00     1     0   18.0000        S  Third woman FALSE        0
21       0   35.00     0     0   26.0000        S Second    man  TRUE        0
22       0   34.00     0     0   13.0000        S Second    man  TRUE        1
23       1   15.00     0     0    8.0292        Q  Third child  TRUE        1
24       0   28.00     0     0   35.5000        S  First    man  TRUE        1
25       1    8.00     3     1   21.0750        S         child FALSE        0
26       1   38.00     1     5   31.3875        S  Third woman FALSE        1
28       0   19.00     3     2  263.0000        S  First    man FALSE        0
31       0   40.00     0     0   27.7208        C  First    man  TRUE        0
34       0   66.00     0     0   10.5000        S Second    man  TRUE        0
35       0   28.00     1     0   82.1708        C  First    man FALSE        0
36       0   42.00     1     0   52.0000        S  First    man FALSE        0
38       0   21.00     0     0    8.0500        S  Third    man  TRUE        0
39       1   18.00     2     0   18.0000        S  Third woman FALSE        0
```

Using Mean:

age_mean=mean(dataset$age,na.rm=T)
recover_missing_age_mean = dataset$age[is.na(dataset$age)]<-age_mean
recover_missing_age_mean
print(dataset)

Output:

```
> age_mean=mean(dataset$age,na.rm=T)
> recover_missing_age_mean = dataset$age[is.na(dataset$age)]<-age_mean
> recover_missing_age_mean
[1] 33.32837
> print(dataset)
    gender      age sibsp parch      fare embarked  class    who alone survived
1        0 22.00000     1     0    7.2500        S  Third    man FALSE        0
2        1 38.00000     1     0   71.2833        C  First woman  FALL        1
3        1 26.00000     0     0    7.9250        S  Third woman  TRUE        1
4        1 35.00000     1     0   53.1000        S  First woman  FALL        1
5        0 35.00000     0     0    8.0500        S  Third    man  TRUE        0
6        0 33.32837     0     0    8.4583        Q  Third    man  TRUE        0
7        0 54.00000     0     0   51.8625        S  First    man  TRUE        0
8        0  2.00000     3     1   21.0750        S  Third child FALSE        0
9        1 27.00000     0     2   11.1333        S  Third woman FALSE        1
10       1 14.00000     1     0   30.0708        C Second child FALSE        1
11       1  4.00000     1     1   16.7000        S  Third child FALSE        1
12       1 58.00000     0     0   26.5500        S  First woman  TRUE        1
13       0 20.00000     0     0    8.0500        S  Third    man  TRUE        0
14       0 39.00000     1     5   31.2750        S  Third    man FALSE        0
15       1 14.00000     0     0    7.8542        S  Third child  TRUE        0
16       1 55.00000     0     0   16.0000        S Second woman  TRUE        1
17       0  2.00000     4     1   29.1250        Q  Third child FALSE        0
18       0 33.32837     0     0   13.0000        S Second    man  TRUE        1
19       1 31.00000     1     0   18.0000        S  Third woman FALSE        0
20       1 33.32837     0     0    7.2250        C  Third woman  TRUE        1
21       0 35.00000     0     0   26.0000        S Second    man  TRUE        0
22       0 34.00000     0     0   13.0000        S Second    man  TRUE        1
23       1 15.00000     0     0    8.0292        Q  Third child  TRUE        1
24       0 28.00000     0     0   35.5000        S  First    man  TRUE        1
25       1  8.00000     3     1   21.0750        S         child FALSE        0
26       1 38.00000     1     5   31.3875        S  Third woman FALSE        1
27       0 33.32837     0     0    7.2250        C  Third    man  TRUE        0
28       0 19.00000     3     2  263.0000        S  First    man FALSE        0
29       1 33.32837     0     0    7.8792        Q  Third woman  TRUE        1
30       0 33.32837     0     0    7.8958        S  Third    man  TRUE        0
31       0 40.00000     0     0   27.7208        C  First    man  TRUE        0
32       1 33.32837     1     0  146.5208        C  First woman FALSE        1
33       1 33.32837     0     0    7.7500        Q  Third woman  TRUE        1
```

Using Mode:

```
age_mode=find_mode(dataset$age)
recover_missing_age_mode = dataset$age[is.na(dataset$age)]<-age_mode
recover_missing_age_mode
print(dataset)
```

Output:

```
> age_mode=find_mode(dataset$age)
> recover_missing_age_mode = dataset$age[is.na(dataset$age)]<-age_mode
> recover_missing_age_mode
[1] 33.32837
> print(dataset)
    gender       age sibsp parch      fare embarked  class    who alone survived
1        0 22.00000     1     0    7.2500        S  Third    man FALSE        0
2        1 38.00000     1     0   71.2833        C  First woman  FALL        1
3        1 26.00000     0     0    7.9250        S  Third woman  TRUE        1
4        1 35.00000     1     0   53.1000        S  First woman  FALL        1
5        0 35.00000     0     0    8.0500        S  Third    man  TRUE        0
6        0 33.32837     0     0    8.4583        Q  Third    man  TRUE        0
7        0 54.00000     0     0   51.8625        S  First    man  TRUE        0
8        0  2.00000     3     1   21.0750        S  Third child FALSE        0
9        1 27.00000     0     2   11.1333        S  Third woman FALSE        1
10       1 14.00000     1     0   30.0708        C Second child FALSE        1
11       1  4.00000     1     1   16.7000        S  Third child FALSE        1
12       1 58.00000     0     0   26.5500        S  First woman  TRUE        1
13       0 20.00000     0     0    8.0500        S  Third    man  TRUE        0
14       0 39.00000     1     5   31.2750        S  Third    man FALSE        0
15       1 14.00000     0     0    7.8542        S  Third child  TRUE        0
16       1 55.00000     0     0   16.0000        S Second woman  TRUE        1
17       0  2.00000     4     1   29.1250        Q  Third child FALSE        0
18       0 33.32837     0     0   13.0000        S Second    man  TRUE        1
19       1 31.00000     1     0   18.0000        S  Third woman FALSE        0
20       1 33.32837     0     0    7.2250        C  Third woman  TRUE        1
21       0 35.00000     0     0   26.0000        S Second    man  TRUE        0
22       0 34.00000     0     0   13.0000        S Second    man  TRUE        1
23       1 15.00000     0     0    8.0292        Q  Third child  TRUE        1
24       0 28.00000     0     0   35.5000        S  First    man  TRUE        1
25       1  8.00000     3     1   21.0750        S        child FALSE        0
26       1 38.00000     1     5   31.3875        S  Third woman FALSE        1
27       0 33.32837     0     0    7.2250        C  Third    man  TRUE        0
28       0 19.00000     3     2  263.0000        S  First    man FALSE        0
29       1 33.32837     0     0    7.8792        Q  Third woman  TRUE        1
30       0 33.32837     0     0    7.8958        S  Third    man  TRUE        0
```

Annotate:

Explanation: To improve data accuracy, interpretability, and analysis for better decision-making, I use annotations in this case. Here, I annotate embarked, class, who and alone .

Code Segment:

For embarked:

```
dataset$embarked<-factor(dataset$embarked,levels=c("S","C","Q"),labels=c(1,2,3))

print(dataset$embarked)

print(dataset)
```

Output:

```
> dataset$embarked<-factor(dataset$embarked,levels=c("S","C","Q"),labels=c(1,2,3))
> print(dataset$embarked)
  [1] 1    2    1    1    1    3    1    1    1    2    1    1    1    1    1    1    3    1    1    2    1    1    3
 [24] 1    1    1    2    1    3    1    2    2    3    1    2    1    2    1    1    2    1    1    2    2    3    1
 [47] 3    3    2    1    1    1    2    1    2    1    1    2    1    1    2    <NA> 1    1    2    2    1    1    1
 [70] 1    1    1    1    2    1    1    1    1    1    1    1    1    3    1    1    1    1    1    1    1    1    1
 [93] 1    1    1    1    2    2    1    1    1    1    1    1    1    1    1    1    1    3    1    2    1    1    2
[116] 1    3    1    2    1    1    1    2    1    1    2    3    1    2    1    2    1    1    1    1    2    1    1
[139] 1    2    2    1    1    3    1    1    1    1    1    1    1    1    1    1    1    2    3    1    1    1    1
[162] 1    1    1    1    1    1    1    1    1    1    3    1    1    2    1    1    2    1    1    2    1    1    1
[185] 1    1    3    1    3    1    1    1    1    1    2    2    3    1    3    1    1    1    1    2    1    1    1
[208] 2    3    2    1    1    1    1    3    2    1    1    2    1    1    1    1    1    1    1    1    1    1    1
[231] 1    1    1    1    1    1    1    1    1    1    2    3    1    1    2    3    1    1    1    1
Levels: 1 2 3
> print(dataset)
    gender      age sibsp parch     fare embarked  class   who alone survived
1        0 22.00000     1     0   7.2500        1  Third   man FALSE        0
2        1 38.00000     1     0  71.2833        2  First woman  FALL        1
3        1 26.00000     0     0   7.9250        1  Third woman  TRUE        1
4        1 35.00000     1     0  53.1000        1  First woman  FALL        1
5        0 35.00000     0     0   8.0500        1  Third   man  TRUE        0
6        0 33.32837     0     0   8.4583        3  Third   man  TRUE        0
7        0 54.00000     0     0  51.8625        1  First   man  TRUE        0
8        0  2.00000     3     1  21.0750        1  Third child FALSE        0
9        1 27.00000     0     2  11.1333        1  Third woman FALSE        1
10       1 14.00000     1     0  30.0708        2 Second child FALSE        1
11       1  4.00000     1     1  16.7000        1  Third child FALSE        1
12       1 58.00000     0     0  26.5500        1  First woman  TRUE        1
13       0 20.00000     0     0   8.0500        1  Third   man  TRUE        0
14       0 39.00000     1     5  31.2750        1  Third   man FALSE        0
15       1 14.00000     0     0   7.8542        1  Third child  TRUE        0
16       1 55.00000     0     0  16.0000        1 Second woman  TRUE        1
17       0  2.00000     4     1  29.1250        3  Third child FALSE        0
18       0 33.32837     0     0  13.0000        1 Second   man  TRUE        1
19       1 31.00000     1     0  18.0000        1  Third woman FALSE        0
20       1 33.32837     0     0   7.2250        2  Third woman  TRUE        1
```

For class:

dataset$class<-factor(dataset$class,levels=c("First","Second","Third"),labels=c(11,22,33))

print(dataset$class)

print(dataset)

Output:

```
> dataset$class<-factor(dataset$class,levels=c("First","Second","Third"),labels=c(11,22,33))
> print(dataset$class)
  [1] 33   11   33   11   33   33   11   33   33   22   33   11   33   33   33   22   33   22   33   33   22   22   33
 [24] 11   <NA> 33   33   11   33   33   11   11   33   22   11   11   33   33   33   33   33   22   33   22   33   33
 [47] 33   33   33   33   33   33   11   22   11   11   22   33   22   33   33   11   11   33   11   33   22   33   33
 [70] 33   22   33   22   33   33   33   33   33   22   33   33   33   33   11   22   33   33   33   11   33   33   33
 [93] 11   33   33   33   11   11   22   22   33   33   11   33   33   33   33   33   33   33   33   11   33   33   33   33
[116] 33   <NA> 22   11   33   22   33   22   22   11   33   33   33   33   33   33   33   33   22   22   22   11   11
[139] 33   11   33   33   33   33   22   22   33   33   22   22   22   11   33   33   33   11   33   33   33   33   33
[162] 22   33   33   33   33   11   33   11   33   11   33   33   33   11   33   33   11   22   33   33   22   33   22
[185] 33   11   33   11   33   33   22   22   <NA> 22   11   11   33   33   33   22   33   33   33   33   33   33   33
[208] 33   33   11   33   22   33   22   33   11   33   22   11   22   33   22   33   33   11   33   22   33   22   <NA>
[231] 11   33   22   33   22   33   22   22   22   22   33   33   22   33   33   11   33   22   11   22
Levels: 11 22 33
> print(dataset)
    gender      age sibsp parch     fare embarked class   who alone survived
1        0 22.00000     1     0   7.2500        1    33   man FALSE        0
2        1 38.00000     1     0  71.2833        2    11 woman  FALL        1
3        1 26.00000     0     0   7.9250        1    33 woman  TRUE        1
4        1 35.00000     1     0  53.1000        1    11 woman  FALL        1
5        0 35.00000     0     0   8.0500        1    33   man  TRUE        0
6        0 33.32837     0     0   8.4583        3    33   man  TRUE        0
7        0 54.00000     0     0  51.8625        1    11   man  TRUE        0
8        0  2.00000     3     1  21.0750        1    33 child FALSE        0
9        1 27.00000     0     2  11.1333        1    33 woman FALSE        1
10       1 14.00000     1     0  30.0708        2    22 child FALSE        1
11       1  4.00000     1     1  16.7000        1    33 child FALSE        1
12       1 58.00000     0     0  26.5500        1    11 woman  TRUE        1
13       0 20.00000     0     0   8.0500        1    33   man  TRUE        0
14       0 39.00000     1     5  31.2750        1    33   man FALSE        0
15       1 14.00000     0     0   7.8542        1    33 child  TRUE        0
16       1 55.00000     0     0  16.0000        1    22 woman  TRUE        1
17       0  2.00000     4     1  29.1250        3    33 child FALSE        0
18       0 33.32837     0     0  13.0000        1    22   man  TRUE        1
19       1 31.00000     1     0  18.0000        1    33 woman FALSE        0
20       1 33.32837     0     0   7.2250        2    33 woman  TRUE        1
21       0 35.00000     0     0  26.0000        1    22   man  TRUE        0
```

For who:

dataset$who<-factor(dataset$who,levels=c("man","woman","child"),labels=c(44,55,66))

print(dataset$who)

print(dataset)

Output:

```
> dataset$who<-factor(dataset$who,levels=c("man","woman","child"),labels=c(44,55,66))
> print(dataset$who)
  [1] 44 55 55 55 44 44 44 66 55 66 66 55 44 44 66 55 66 44 55 55 44 44 66 44 66 55 44 44 55 44 44 55 55 44 44 44 44 44
 [39] 55 66 55 55 44 66 55 44 44 55 44 55 66 44 55 55 44 44 55 44 66 66 44 55 44 66 44 44 55 44 55 44 44 55 44 44 44 44
 [77] 44 44 66 55 44 44 55 44 55 55 44 44 55 44 44 44 44 44 44 44 44 44 55 44 55 44 44 44 44 44 55 44 44 55 44 66 44 55
[115] 55 44 44 44 44 66 44 44 44 55 44 66 44 44 55 44 44 44 55 55 44 44 55 44 44 44 55 55 55 44 44 44 44 66 44 44 44 55
[153] 44 44 44 44 55 44 44 44 44 55 44 44 66 66 55 55 44 44 44 66 66 44 44 44 44 55 44 44 55 44 66 66 66 44 55 44 44 44
[191] 55 44 55 66 55 55 44 44 55 55 44 44 44 44 44 66 44 44 55 44 44 55 44 44 44 44 55 55 44 55 44 44 44 44 44 44 44 44
[229] 44 55 55 44 44 66 44 55 44 66 44 44 55 55 44 44 44 44 55 55 44 44
Levels: 44 55 66
> print(dataset)
    gender      age sibsp parch      fare embarked class  who alone survived
1        0 22.00000     1     0    7.2500        1    33   44 FALSE        0
2        1 38.00000     1     0   71.2833        2    11   55  FALL        1
3        1 26.00000     0     0    7.9250        1    33   55  TRUE        1
4        1 35.00000     1     0   53.1000        1    11   55  FALL        1
5        0 35.00000     0     0    8.0500        1    33   44  TRUE        0
6        0 33.32837     0     0    8.4583        3    33   44  TRUE        0
7        0 54.00000     0     0   51.8625        1    11   44  TRUE        0
8        0  2.00000     3     1   21.0750        1    33   66 FALSE        0
9        1 27.00000     0     2   11.1333        1    33   55 FALSE        1
10       1 14.00000     1     0   30.0708        2    22   66 FALSE        1
11       1  4.00000     1     1   16.7000        1    33   66 FALSE        1
12       1 58.00000     0     0   26.5500        1    11   55  TRUE        1
13       0 20.00000     0     0    8.0500        1    33   44  TRUE        0
14       0 39.00000     1     5   31.2750        1    33   44 FALSE        0
15       1 14.00000     0     0    7.8542        1    33   66  TRUE        0
16       1 55.00000     0     0   16.0000        1    22   55  TRUE        1
17       0  2.00000     4     1   29.1250        3    33   66 FALSE        0
18       0 33.32837     0     0   13.0000        1    22   44  TRUE        1
19       1 31.00000     1     0   18.0000        1    33   55 FALSE        0
20       1 33.32837     0     0    7.2250        2    33   55  TRUE        1
21       0 35.00000     0     0   26.0000        1    22   44  TRUE        0
22       0 34.00000     0     0   13.0000        1    22   44  TRUE        1
23       1 15.00000     0     0    8.0292        3    33   66  TRUE        1
24       0 28.00000     0     0   35.5000        1    11   44  TRUE        1
25       1  8.00000     3     1   21.0750        1  <NA>   66 FALSE        0
```

For alone:

dataset$alone<-factor(dataset$alone,levels=c("FALSE","FALL","TRUE"),labels=c(0,5,1))

print(dataset$alone)

print(dataset)

Output:

```
> dataset$alone<-factor(dataset$alone,levels=c("FALSE","FALL","TRUE"),labels=c(0,5,1))
> print(dataset$alone)
  [1] 0 5 1 5 1 1 1 0 0 0 0 1 1 0 1 1 0 1 0 1 1 1 1 1 1 0 0 1 0 1 1 1 0 1 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0 0 0 1 1 1
 [59] 0 0 1 1 0 0 1 0 1 1 0 0 1 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 0 1 0 1 1 1 0 0 1 1 1 0 0 0 1 1 0 1 0 1 0 1 1 1 1 0 1 0 1 5 1 1
[117] 1 0 0 0 0 1 0 1 0 0 1 1 0 1 1 1 0 0 1 1 0 0 1 1 0 1 0 1 1 0 1 0 0 1 1 0 1 0 1 0 1 1 1 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 1
[175] 1 0 0 1 1 1 0 1 0 0 0 1 0 1 0 1 1 1 0 0 1 1 1 0 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1
[233] 1 0 1 1 0 0 1 1 0 0 1 1 1 0 1 0 0 0
Levels: 0 5 1
> print(dataset)
    gender      age sibsp parch      fare embarked class  who alone survived
1        0 22.00000     1     0    7.2500        1    33   44     0        0
2        1 38.00000     1     0   71.2833        2    11   55     5        1
3        1 26.00000     0     0    7.9250        1    33   55     1        1
4        1 35.00000     1     0   53.1000        1    11   55     5        1
5        0 35.00000     0     0    8.0500        1    33   44     1        0
6        0 33.32837     0     0    8.4583        3    33   44     1        0
7        0 54.00000     0     0   51.8625        1    11   44     1        0
8        0  2.00000     3     1   21.0750        1    33   66     0        0
9        1 27.00000     0     2   11.1333        1    33   55     0        1
10       1 14.00000     1     0   30.0708        2    22   66     0        1
11       1  4.00000     1     1   16.7000        1    33   66     0        1
12       1 58.00000     0     0   26.5500        1    11   55     1        1
13       0 20.00000     0     0    8.0500        1    33   44     1        0
14       0 39.00000     1     5   31.2750        1    33   44     0        0
15       1 14.00000     0     0    7.8542        1    33   66     1        0
16       1 55.00000     0     0   16.0000        1    22   55     1        1
17       0  2.00000     4     1   29.1250        3    33   66     0        0
18       0 33.32837     0     0   13.0000        1    22   44     1        1
19       1 31.00000     1     0   18.0000        1    33   55     0        0
20       1 33.32837     0     0    7.2250        2    33   55     1        1
21       0 35.00000     0     0   26.0000        1    22   44     1        0
22       0 34.00000     0     0   13.0000        1    22   44     1        1
23       1 15.00000     0     0    8.0292        3    33   66     1        1
24       0 28.00000     0     0   35.5000        1    11   44     1        1
25       1  8.00000     3     1   21.0750        1  <NA>   66     0        0
26       1 38.00000     1     5   31.3875        1    33   55     0        1
27       0 33.32837     0     0    7.2250        2    33   44     1        0
28       0 19.00000     3     2  263.0000        1    11   44     0        0
29       1 33.32837     0     0    7.8792        3    33   55     1        1
```

Outlier:

Explanation: A dataset's outliers can be utilized to spot problems with data quality, understand data distribution, identify deviations, and improve model performance.

Code Segment:

For age:

```
sort(dataset$age)
```

Output:

```
> sort(dataset$age)
  [1]   0.83000   1.00000   1.00000   1.00000   2.00000   2.00000   2.00000   2.00000   3.00000   3.00000   4.00000
 [12]   4.00000   4.00000   4.00000   5.00000   5.00000   7.00000   8.00000   8.00000   9.00000   9.00000   9.00000
 [23]  11.00000  12.00000  14.00000  14.00000  14.00000  14.50000  15.00000  16.00000  16.00000  16.00000  16.00000
 [34]  16.00000  16.00000  17.00000  17.00000  17.00000  17.00000  18.00000  18.00000  18.00000  18.00000  18.00000
 [45]  18.00000  19.00000  19.00000  19.00000  19.00000  19.00000  19.00000  19.00000  19.00000  19.00000  19.00000
 [56]  20.00000  20.00000  20.00000  20.00000  20.50000  21.00000  21.00000  21.00000  21.00000  21.00000  21.00000
 [67]  21.00000  21.00000  21.00000  22.00000  22.00000  22.00000  22.00000  22.00000  22.00000  22.00000  22.00000
 [78]  22.00000  23.00000  23.00000  23.00000  24.00000  24.00000  24.00000  24.00000  24.00000  24.00000  24.00000
 [89]  24.00000  24.00000  25.00000  25.00000  25.00000  26.00000  26.00000  26.00000  26.00000  26.00000  26.00000
[100]  27.00000  27.00000  27.00000  27.00000  27.00000  28.00000  28.00000  28.00000  28.00000  28.00000  28.00000
[111]  28.00000  28.50000  29.00000  29.00000  29.00000  29.00000  29.00000  29.00000  29.00000  29.00000  30.00000
[122]  30.00000  30.00000  30.00000  30.00000  30.00000  31.00000  31.00000  32.00000  32.00000  32.00000  32.00000
[133]  32.00000  32.50000  33.00000  33.00000  33.00000  33.00000  33.32837  33.32837  33.32837  33.32837  33.32837
[144]  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837
[155]  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837
[166]  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837
[177]  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  33.32837  34.00000
[188]  34.00000  34.00000  34.00000  35.00000  35.00000  35.00000  35.00000  35.00000  36.00000  36.00000  37.00000
[199]  37.00000  37.00000  38.00000  38.00000  38.00000  38.00000  38.00000  39.00000  40.00000  40.00000  40.00000
[210]  40.00000  40.00000  40.50000  42.00000  42.00000  42.00000  42.00000  44.00000  44.00000  44.00000  44.00000
[221]  45.00000  45.00000  45.00000  45.00000  46.00000  47.00000  47.00000  49.00000  50.00000  51.00000  51.00000
[232]  51.00000  54.00000  54.00000  54.00000  55.00000  55.50000  56.00000  58.00000  58.00000  59.00000  59.00000
[243]  61.00000  65.00000  66.00000  70.50000  71.00000 325.00000 365.00000 455.00000
```

Code Segment:

```
dataset_outlier=subset(dataset,age<=19)

dataset_outlier
```

Output:

```
> dataset_outlier=subset(dataset,age<=19)
> dataset_outlier
    gender   age sibsp parch     fare embarked class  who alone survived
8        0  2.00     3     1  21.0750        1    33   66     0        0
10       1 14.00     1     0  30.0708        2    22   66     0        1
11       1  4.00     1     1  16.7000        1    33   66     0        1
15       1 14.00     0     0   7.8542        1    33   66     1        0
17       0  2.00     4     1  29.1250        3    33   66     0        0
23       1 15.00     0     0   8.0292        3    33   66     1        1
25       1  8.00     3     1  21.0750        1  <NA>   66     0        0
28       0 19.00     3     2 263.0000        1    11   44     0        0
39       1 18.00     2     0  18.0000        1    33   55     0        0
40       1 14.00     1     0  11.2417        2    33   66     0        1
44       1  3.00     1     2  41.5792        2    22   66     0        1
45       1 19.00     0     0   7.8792        3    33   55     1        1
50       1 18.00     1     0  17.8000        1    33   55     0        0
51       0  7.00     4     1  39.6875        1    33   66     0        0
59       1  5.00     1     2  27.7500        1    22   66     0        1
60       0 11.00     5     2  46.9000        1    33   66     0        0
64       0  4.00     3     2  27.9000        1    33   66     0        0
68       0 19.00     0     0   8.1583        1    33   44     1        0
69       1 17.00     4     2   7.9250        1    33   55     0        1
72       1 16.00     5     2  46.9000        1    33   55     0        0
79       0  0.83     0     2  29.0000        1    22   66     0        1
85       1 17.00     0     0  10.5000        1    22   55     1        1
87       0 16.00     1     3  34.3750        1    33   44     0        0
112      1 14.50     1     0  14.4542        2    33   66     0        0
115      1 17.00     0     0  14.4583        2    33   55     1        0
120      1  2.00     4     2  31.2750        1    33   66     0        0
126      0 12.00     1     0  11.2417        2    33   66     0        1
137      1 19.00     0     2  26.2833        1    11   55     0        1
139      0 16.00     0     0   9.2167        1    33   44     1        0
144      0 19.00     0     0   6.7500        3    33   44     1        0
145      0 18.00     0     0  11.5000        1    22   44     1        0
146      0 19.00     1     1  36.7500        1    22   44     0        0
148      1  9.00     2     2  34.3750        1    33   66     0        0
157      1 16.00     0     0   7.7333        3    33   55     1        1
```

Code Segment:

```
dataset_outlier_location=which(dataset$age<19)

dataset_outlier_location
```

Output:

```
> dataset_outlier_location=which(dataset$age<19)
> dataset_outlier_location
 [1]   8  10  11  15  17  23  25  39  40  44  50  51  59  60  64  69  72  79  85  87 112 115 120 126 139 145 148 157 164
[30] 165 166 172 173 176 183 184 185 194 205 206 209 221 229 234 238
>
```

Code Segment:

```
dataset$age[dataset_outlier_location]<=NA

print(dataset)
```

Output:

```
> dataset$age[dataset_outlier_location]<-NA
> print(dataset)
    gender       age sibsp parch      fare embarked class  who alone survived
1        0 22.00000     1     0    7.2500        1    33   44     0        0
2        1 38.00000     1     0   71.2833        2    11   55     5        1
3        1 26.00000     0     0    7.9250        1    33   55     1        1
4        1 35.00000     1     0   53.1000        1    11   55     5        1
5        0 35.00000     0     0    8.0500        1    33   44     1        0
6        0 33.32837     0     0    8.4583        3    33   44     1        0
7        0 54.00000     0     0   51.8625        1    11   44     1        0
8        0       NA     3     1   21.0750        1    33   66     0        0
9        1 27.00000     0     2   11.1333        1    33   55     0        1
10       1       NA     1     0   30.0708        2    22   66     0        1
11       1       NA     1     1   16.7000        1    33   66     0        1
12       1 58.00000     0     0   26.5500        1    11   55     1        1
13       0 20.00000     0     0    8.0500        1    33   44     1        0
14       0 39.00000     1     5   31.2750        1    33   44     0        0
15       1       NA     0     0    7.8542        1    33   66     1        0
16       1 55.00000     0     0   16.0000        1    22   55     1        1
17       0       NA     4     1   29.1250        3    33   66     0        0
18       0 33.32837     0     0   13.0000        1    22   44     1        1
19       1 31.00000     1     0   18.0000        1    33   55     0        0
20       1 33.32837     0     0    7.2250        2    33   55     1        1
21       0 35.00000     0     0   26.0000        1    22   44     1        0
22       0 34.00000     0     0   13.0000        1    22   44     1        1
23       1       NA     0     0    8.0292        3    33   66     1        1
24       0 28.00000     0     0   35.5000        1    11   44     1        1
25       1       NA     3     1   21.0750        1  <NA>   66     0        0
26       1 38.00000     1     5   31.3875        1    33   55     0        1
27       0 33.32837     0     0    7.2250        2    33   44     1        0
28       0 19.00000     3     2  263.0000        1    11   44     0        0
29       1 33.32837     0     0    7.8792        3    33   55     1        1
30       0 33.32837     0     0    7.8958        1    33   44     1        0
31       0 40.00000     0     0   27.7208        2    11   44     1        0
32       1 33.32837     1     0  146.5208        2    11   55     0        1
33       1 33.32837     0     0    7.7500        3    33   55     1        1
```

For fare:

sort(dataset$fare)

Output:

```
> sort(dataset$fare)
  [1]   0.0000   6.4958   6.7500   6.9750   7.0500   7.0500   7.1250   7.1417   7.2250   7.2250   7.2250   7.2250
 [13]   7.2292   7.2292   7.2292   7.2500   7.2500   7.2500   7.2500   7.3125   7.5500   7.6500   7.6500   7.7333
 [25]   7.7500   7.7500   7.7500   7.7500   7.7500   7.7500   7.7500   7.7500   7.7500   7.7750   7.7750   7.7750
 [37]   7.7750   7.7875   7.7958   7.8000   7.8542   7.8542   7.8542   7.8542   7.8792   7.8792   7.8958   7.8958
 [49]   7.8958   7.8958   7.8958   7.8958   7.8958   7.8958   7.8958   7.8958   7.9250   7.9250   7.9250   7.9250
 [61]   7.9250   7.9250   8.0292   8.0500   8.0500   8.0500   8.0500   8.0500   8.0500   8.0500   8.0500   8.0500
 [73]   8.0500   8.0500   8.0500   8.0500   8.0500   8.0500   8.0500   8.1583   8.4042   8.4583   8.6542   8.6625
 [85]   8.6625   8.6625   9.0000   9.2167   9.3500   9.4750   9.5000   9.5000   9.8250  10.4625  10.5000  10.5000
 [97]  10.5000  10.5000  10.5000  10.5000  10.5000  10.5000  10.5000  10.5000  11.1333  11.1333  11.2417  11.2417
[109]  11.5000  12.2750  12.4750  12.5250  13.0000  13.0000  13.0000  13.0000  13.0000  13.0000  13.0000  13.0000
[121]  13.0000  13.0000  13.0000  13.0000  13.5000  14.4542  14.4542  14.4542  14.4583  14.5000  14.5000  14.5000
[133]  15.0458  15.0500  15.2458  15.2458  15.5000  15.5000  15.5000  15.5000  15.7500  15.8500  15.8500  15.8500
[145]  16.0000  16.1000  16.7000  17.8000  18.0000  18.0000  18.7875  20.5250  20.5750  21.0000  21.0000  21.0000
[157]  21.0750  21.0750  21.6792  22.0250  22.3583  23.0000  24.1500  25.4667  25.4667  25.9250  26.0000  26.0000
[169]  26.0000  26.0000  26.0000  26.0000  26.0000  26.0000  26.2500  26.2833  26.5500  26.5500  27.0000  27.7208
[181]  27.7208  27.7208  27.7500  27.9000  27.9000  28.7125  29.0000  29.1250  29.1250  30.0708  30.0708  30.6958
[193]  31.0000  31.2750  31.2750  31.3875  31.3875  31.3875  33.5000  34.3750  34.3750  34.6542  35.5000  35.5000
[205]  36.7500  39.0000  39.6875  39.6875  41.5792  46.9000  46.9000  47.1000  50.0000  51.8625  52.0000  52.0000
[217]  52.5542  53.1000  53.1000  55.0000  56.4958  56.4958  61.1750  61.3792  61.9792  63.3583  66.6000  69.5500
[229]  69.5500  69.5500  71.2833  73.5000  73.5000  76.2917  76.7292  77.2875  77.2875  79.2000  80.0000  82.1708
[241]  83.4750  83.4750  90.0000  90.0000 113.2750 146.5208 146.5208 247.5208 263.0000 263.0000
>
```

Code Segment:

dataset_outlier=subset(dataset,fare<=8.034)

dataset_outlier

Output:

```
> dataset_outlier=subset(dataset,fare<=8.034)
> dataset_outlier
    gender       age sibsp parch    fare embarked class  who alone survived
1        0  22.00000     1     0  7.2500        1    33   44     0        0
3        1  26.00000     0     0  7.9250        1    33   55     1        1
15       1        NA     0     0  7.8542        1    33   66     1        0
20       1  33.32837     0     0  7.2250        2    33   55     1        1
23       1        NA     0     0  8.0292        3    33   66     1        1
27       0  33.32837     0     0  7.2250        2    33   44     1        0
29       1  33.32837     0     0  7.8792        3    33   55     1        1
30       0  33.32837     0     0  7.8958        1    33   44     1        0
33       1  33.32837     0     0  7.7500        3    33   55     1        1
37       0  33.32837     0     0  7.2292        2    33   44     1        1
43       0  33.32837     0     0  7.8958        2    33   44     1        0
45       1  19.00000     0     0  7.8792        3    33   55     1        1
48       1  33.32837     0     0  7.7500        3    33   55     1        1
52       0  21.00000     0     0  7.8000        1    33   44     1        0
58       0  28.50000     0     0  7.2292        2    33   44     1        0
61       0  22.00000     0     0  7.2292        2    33   44     1        0
69       1        NA     4     2  7.9250        1    33   55     0        1
76       0  25.00000     0     0  7.6500        1    33   44     1        0
77       0  33.32837     0     0  7.8958        1    33   44     1        0
83       1  33.32837     0     0  7.7875        3    33   55     1        1
92       0  20.00000     0     0  7.8542        1    33   44     1        0
95       0  59.00000     0     0  7.2500        1    33   44     1        0
101      1  28.00000     0     0  7.8958        1    33   55     1        0
102      0  33.32837     0     0  7.8958        1    33   44     1        0
105      0  37.00000     2     0  7.9250        1    33   44     0        0
106      0  28.00000     0     0  7.8958        1    33   44     1        0
107      1  21.00000     0     0  7.6500        1    33   55     1        1
108      0  33.32837     0     0  7.7750        1    33   44     1        1
109      0  38.00000     0     0  7.8958        1    33   44     1        0
116      0  21.00000     0     0  7.9250        1    33   44     1        0
117      0  70.50000     0     0  7.7500        3  <NA>   44     1        0
127      0  33.32837     0     0  7.7500        3    33   44     1        0
```

Code Segment:

```
dataset_outlier_location=which(dataset$fare<8.034)

dataset_outlier_location
```

Output:

```
> dataset_outlier_location=which(dataset$fare<8.034)
> dataset_outlier_location
 [1]   1   3  15  20  23  27  29  30  33  37  43  45  48  52  58  61  69  76  77  83  92  95 101 102 105 106 107 108 109
[30] 116 117 127 128 130 131 132 142 144 147 155 157 163 174 176 180 190 193 197 199 203 204 209 211 213 215 217 224 228
[59] 232 236 244 245 247
>
```

Code Segment:

```
dataset$fare[dataset_outlier_location]<-NA

print(dataset)
```

Output:

```
> dataset$fare[dataset_outlier_location]<-NA
> print(dataset)
    gender      age sibsp parch     fare embarked class  who alone survived
1        0 22.00000     1     0       NA        1    33   44     0        0
2        1 38.00000     1     0  71.2833        2    11   55     5        1
3        1 26.00000     0     0       NA        1    33   55     1        1
4        1 35.00000     1     0  53.1000        1    11   55     5        1
5        0 35.00000     0     0   8.0500        1    33   44     1        0
6        0 33.32837     0     0   8.4583        3    33   44     1        0
7        0 54.00000     0     0  51.8625        1    11   44     1        0
8        0       NA     3     1  21.0750        1    33   66     0        0
9        1 27.00000     0     2  11.1333        1    33   55     0        1
10       1       NA     1     0  30.0708        2    22   66     0        1
11       1       NA     1     1  16.7000        1    33   66     0        1
12       1 58.00000     0     0  26.5500        1    11   55     1        1
13       0 20.00000     0     0   8.0500        1    33   44     1        0
14       0 39.00000     1     5  31.2750        1    33   44     0        0
15       1       NA     0     0       NA        1    33   66     1        0
16       1 55.00000     0     0  16.0000        1    22   55     1        1
17       0       NA     4     1  29.1250        3    33   66     0        0
18       0 33.32837     0     0  13.0000        1    22   44     1        1
19       1 31.00000     1     0  18.0000        1    33   55     0        0
20       1 33.32837     0     0       NA        2    33   55     1        1
21       0 35.00000     0     0  26.0000        1    22   44     1        0
22       0 34.00000     0     0  13.0000        1    22   44     1        1
23       1       NA     0     0       NA        3    33   66     1        1
24       0 28.00000     0     0  35.5000        1    11   44     1        1
25       1       NA     3     1  21.0750        1  <NA>   66     0        0
26       1 38.00000     1     5  31.3875        1    33   55     0        1
27       0 33.32837     0     0       NA        2    33   44     1        0
28       0 19.00000     3     2 263.0000        1    11   44     0        0
29       1 33.32837     0     0       NA        3    33   55     1        1
30       0 33.32837     0     0       NA        1    33   44     1        0
31       0 40.00000     0     0  27.7208        2    11   44     1        0
32       1 33.32837     1     0 146.5208        2    11   55     0        1
```

Data Transformation:

Explanation: As we already know, normalization, summarization, noise removal, smoothing, and data summarization are all processes in the data transformation process. I applied normalization to the data set we utilized.

Normalization:

Explanation: Normalization techniques have a favorable effect on the statistical distribution of the data since they enable us to reduce the size of the variables. I've standardized the columns in this data set to range from 1 to 5.

Code Segment:

```
min_max_normalization<-function(x){(x-min(x))/(max(x)-min(x))}

dataset<-as.data.frame(lapply(dataset[1:5],min_max_normalization))

dataset
```

Output:

```
> min_max_normalization<-function(x){(x-min(x))/(max(x)-min(x))}
> dataset<-as.data.frame(lapply(dataset[1:5],min_max_normalization))
> dataset
   gender age sibsp parch fare
1       0  NA 0.125   0.0   NA
2       1  NA 0.125   0.0   NA
3       1  NA 0.000   0.0   NA
4       1  NA 0.125   0.0   NA
5       0  NA 0.000   0.0   NA
6       0  NA 0.000   0.0   NA
7       0  NA 0.000   0.0   NA
8       0  NA 0.375   0.2   NA
9       1  NA 0.000   0.4   NA
10      1  NA 0.125   0.0   NA
11      1  NA 0.125   0.2   NA
12      1  NA 0.000   0.0   NA
13      0  NA 0.000   0.0   NA
14      0  NA 0.125   1.0   NA
15      1  NA 0.000   0.0   NA
16      1  NA 0.000   0.0   NA
17      0  NA 0.500   0.2   NA
18      0  NA 0.000   0.0   NA
19      1  NA 0.125   0.0   NA
20      1  NA 0.000   0.0   NA
21      0  NA 0.000   0.0   NA
22      0  NA 0.000   0.0   NA
23      1  NA 0.000   0.0   NA
24      0  NA 0.000   0.0   NA
25      1  NA 0.375   0.2   NA
26      1  NA 0.125   1.0   NA
27      0  NA 0.000   0.0   NA
28      0  NA 0.375   0.4   NA
29      1  NA 0.000   0.0   NA
30      0  NA 0.000   0.0   NA
```

Invalid Value:

Explanation: Invalid values in a dataset are used to represent missing or unknown data, ensuring data completeness and providing a standardized representation for missing information.

Code Segment:

For who(Most Frequent Value):

```
find_mode <- function(x) {
  u <- unique(x)
  tab <- tabulate(match(x, u))
  u[tab == max(tab)]
}
most_frequent_who=find_mode(dataset$who)
most_frequent_who
```

Output:

```
> find_mode <- function(x) {
+   u <- unique(x)
+   tab <- tabulate(match(x, u))
+   u[tab == max(tab)]
+ }
> most_frequent_who=find_mode(dataset$who)
> most_frequent_who
[1] "man"
```

Code Segment:

```
dataset$who[16]<-most_frequent_who
print(dataset)
```

Output:

11	1	4.00	1	1	16.7000	S	Third	child	FALSE	1
12	1	58.00	0	0	26.5500	S	First	woman	TRUE	1
13	NA	20.00	0	0	8.0500	S	Third	man	TRUE	0
14	0	39.00	1	5	31.2750	S	Third	man	FALSE	0
15	1	14.00	0	0	7.8542	S	Third	child	TRUE	0
16	1	55.00	0	0	16.0000	S	Second	man	TRUE	1
17	0	2.00	4	1	29.1250	Q	Third	child	FALSE	0
18	0	NA	0	0	13.0000	S	Second	man	TRUE	1
19	1	31.00	1	0	18.0000	S	Third	woman	FALSE	0
20	1	NA	0	0	7.2250	C	Third	woman	TRUE	1
21	0	35.00	0	0	26.0000	S	Second	man	TRUE	0
22	0	34.00	0	0	13.0000	S	Second	man	TRUE	1
23	1	15.00	0	0	8.0292	Q	Third	child	TRUE	1
24	0	28.00	0	0	35.5000	S	First	man	TRUE	1