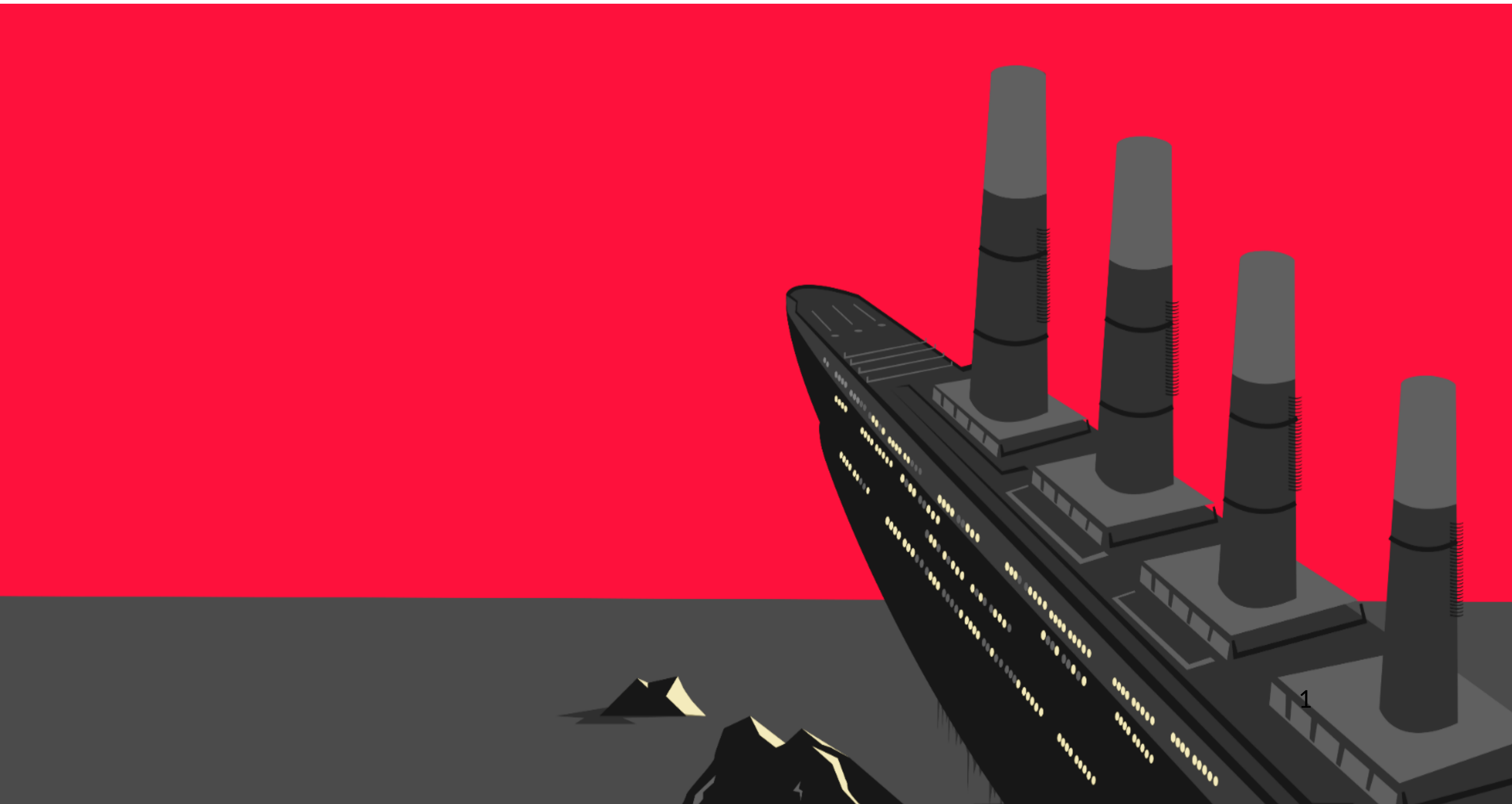
Case Study:

Applying Exploratory Data Analysis Techniques to the Titanic Dataset Using R

By:

Amr Eleraqi

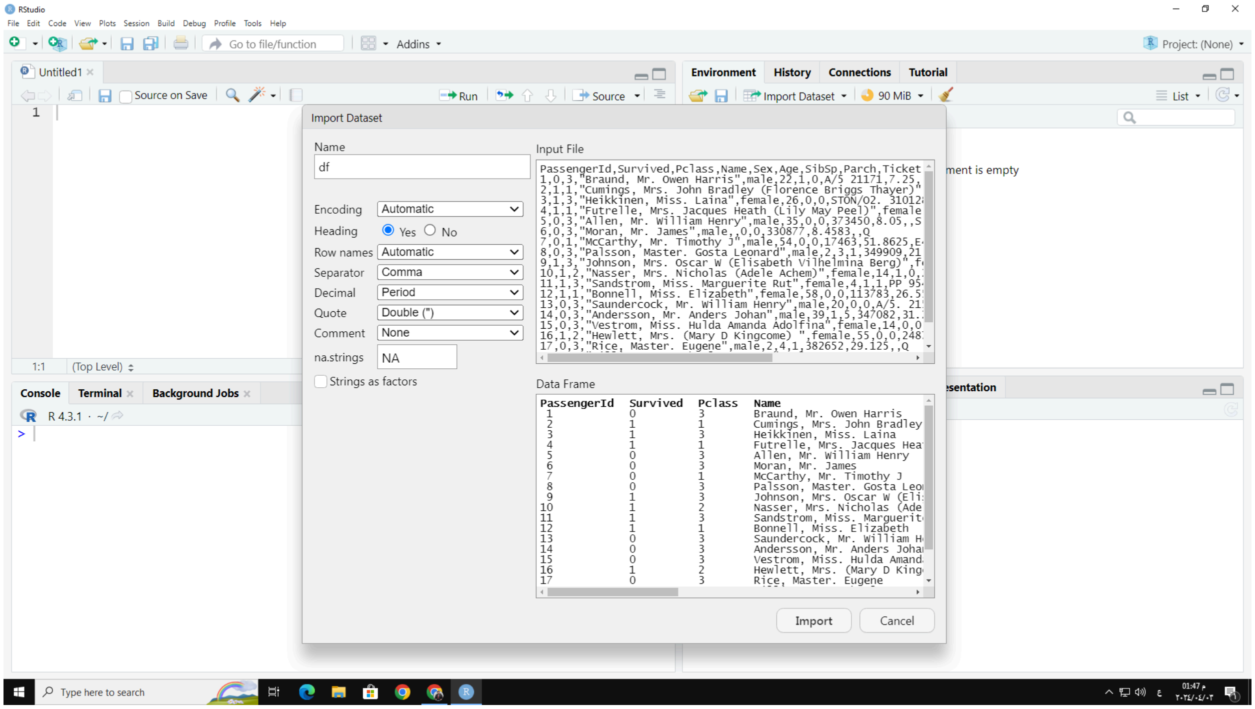
aeleraqi@aucegypt.edu



🌍 Data Source: https://github.com/datasciencedojo/datasets/blob/master/titanic.csv

➡️ Importing Data:

My preferred approach to import data into R Studio is through the use of the "Import Data" option found within the Environment tab. This feature offers an array of customizable settings tailored to the specific type of file being imported, whether it be a CSV or XLS document.



Import Dataset

Name: df

Encoding: Automatic
Heading: ● Yes ○ No
Row names: Automatic
Separator: Comma
Decimal: Period
Quote: Double (")
Comment: None
na.strings: NA
□ Strings as factors

Input File

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 31012
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",f
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,
11,1,3,"Sandstrom, Miss. Marguerite Rut",female,4,1,1,PP 95
12,1,1,"Bonnell, Miss. Elizabeth",female,58,0,0,113783,26.5
13,0,3,"Saundercock, Mr. William Henry",male,20,0,0,A/5. 21
14,0,3,"Andersson, Mr. Anders Johan",male,39,1,5,347082,31.
15,0,3,"Vestrom, Miss. Hulda Amanda Adolfina",female,14,0,0
16,1,2,"Hewlett, Mrs. (Mary D Kingcome) ",female,55,0,0,248
17,0,3,"Rice, Master. Eugene",male,2,4,1,382652,29.125,,Q
```

Data Frame

```
PassengerId  Survived  Pclass  Name
 1           0         3       Braund, Mr. Owen Harris
 2           1         1       Cumings, Mrs. John Bradley
 3           1         3       Heikkinen, Miss. Laina
 4           1         1       Futrelle, Mrs. Jacques Hea
 5           0         3       Allen, Mr. William Henry
 6           0         3       Moran, Mr. James
 7           0         1       McCarthy, Mr. Timothy J
 8           0         3       Palsson, Master. Gosta Leo
 9           1         3       Johnson, Mrs. Oscar W (Eli
10           1         2       Nasser, Mrs. Nicholas (Ade
11           1         3       Sandstrom, Miss. Marguerit
12           1         1       Bonnell, Miss. Elizabeth
13           0         3       Saundercock, Mr. William H
14           0         3       Andersson, Mr. Anders Joha
15           0         3       Vestrom, Miss. Hulda Amand
16           1         2       Hewlett, Mrs. (Mary D King
17           0         3       Rice, Master. Eugene
```

Import   Cancel

Exploring the Data:

After successfully importing the data set into R Studio, there are multiple functions available to help us explore the content effectively. Among those, `View()`, `head()` and `tail()` are particularly useful when navigating through large datasets. Let me briefly describe each of them below:

- `View()`: With view(), you can display the entire data frame interactively, enabling easy navigation and exploration of individual cells, columns, and rows directly within RStudio.

- `head()`: When dealing with extensive data sets, sometimes it is more practical to examine just a few initial lines rather than viewing the whole table. That's where head() comes in handy. It returns the first n observations (rows) of a given data frame, defaulting to six if no argument is provided.

  ```
  head(df, n = 10) # Display the first 10 rows instead
  ```

- `tail()`: Similar to head(), tail() displays the last n observations of the data frame, showing six rows by default.

```
tail(data_frame_name, n = 5) # Display only the last 5 rows
```

- `dim()`: returns the dimensions of the data frame, specifically indicating how many rows and columns it has.

Understanding the measurements:

After obtaining a general overview of the data frame using the above functions, delving deeper becomes crucial to thoroughly comprehend the nature of the variables and their respective measurements present in the dataset. Understanding the exact meaning of each column and knowing the measurement scales facilitate accurate downstream data analysis and interpretation.

Here are several key functions in R that allow us to acquire detailed insights into your data frame:

names()

Returns character vectors representing the column names of a data frame

```
> names(df)
 [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"
 [6] "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"
[11] "Cabin"       "Embarked"
```

str()

Displays the internal structure of objects, providing details on class, length, and modes of components

```
> str(df)
'data.frame':     891 obs. of  12 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley
(Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques
Heath (Lily May Peel)" ...
 $ Sex        : chr  "male" "female" "female" "female" ...
 $ 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 ...
 $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : chr  "" "C85" "" "C123" ...
 $ Embarked   : chr  "S" "C" "S" "S" ...
```

Brief descriptions for each variable:

Name	Title	Description
PassengerId	Passenger Identifier	Unique ID assigned to each passenger
Survived	Survivorship Status	Flag indicating whether the passenger lived (1) or died (0)
Pclass	Ticket Fare Class	Travel class category: First, Second, Third
Name	Passenger Full Name	Legal name of the passenger
Sex	Passenger Gender	Male or Female gender classification
Age	Age at Embarkation	Continuous age variable measured in years
SibSp	Family Members	Quantitative metric for sibling and spouse presence
Parch	Parents and Children	Counter measuring parental units and offspring
Ticket	Booking Reference	Alpha-numeric reference string tied to the ticket
Fare	Voyage Expense	Cost paid for the journey expressed in monetary terms
Cabin	Accommodation Space Assignment	Label designating allocated cabin compartment
Embarked	Boarding Location	Departure port of the passenger

Handling the Missing Values:

1. **Calculate the number of NA values in each column of df**

```
> colSums(is.na(df))
PassengerId    Survived      Pclass        Name         Sex         Age 
          0           0           0           0           0         177 
      SibSp       Parch      Ticket        Fare       Cabin    Embarked 
          0           0           0           0           0           0 
```

2. **Remove rows with missing values**

```
clean_df <- na.omit(df)
```

3. **Replacing the missing values with the mean**

The following command replaces any missing values in the Age column of the data frame df with the mean of the non-missing values in the same column

```
> df$Age[is.na(df$Age)] <- mean(df$Age,na.rm = T)
```

💡 Replacing missing values with the mean or median depends upon the underlying assumptions, level of tolerance towards extreme values, and sensitivity to influential data points. Both approaches have pros and cons, which necessitate careful consideration before applying them to fill in gaps caused by missingness.

This command counts the number of missing values in the Age column

```
> sum(is.na(df$Age))
[1] 0
```

calculates the mean of the non-missing values in the Age column

```
> mean(df$Age,na.rm = T)
[1] 29.69912
```

calculates the mean of the Age column

```
> mean(df$Age)
[1] 29.69912
```

Transforming Selected Variables into Factors

as.factor() conversions convert the Survived, Pclass, and Sex columns to factors. Recall that factors represent categorical variables encoded as integer values internally accompanied by labeled levels. This command converts the Survived, Pclass and Sex columns of the data frame df into factors

```
> df$Survived <- as.factor(df$Survived)
> df$Pclass <- as.factor(df$Pclass)
> df$Sex <- as.factor(df$Sex)
```

The following command checks if the Survived, Pclass and Sex columns of the data frame df is a factor

```
> is.factor(df$Survived)
[1] TRUE
```

```
> levels(df$Survived)
[1] "0"  "1"
```

One alternative method involves employing the `ifelse` function combined with assignment operators to create new factor variables with desired labels. Here's the modified example:

```
df$Survived <- ifelse(df$Survived==1,"Yes","Yes")
```

```
> levels(df$Survived)
[1] "Yes"  "Yes"
```

```
> str(df)
'data.frame':      891 obs. of  12 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 1 2 2 ...
 $ Pclass     : Factor w/ 3 levels "1","2","3": 3 1 3 1 3 3 1 3 3 2 ...
 $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley
(Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques
Heath (Lily May Peel)" ...
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 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 ...
 $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : chr  "" "C85" "" "C123" ...
 $ Embarked   : chr  "S" "C" "S" "S" ...
```

💡 For discrete variables transformed into factors, the str(df) command would reveal the shift from their original data types to factors, along with associated factor level information.

```
> summary(df)
  PassengerId     Survived Pclass      Name                 Sex     
 Min.   :  1.0   0:549    1:216   Length:891         female:314  
 1st Qu.:223.5   1:342    2:184   Class :character   male  :577  
 Median :446.0            3:491   Mode  :character               
 Mean   :446.0                                                   
 3rd Qu.:668.5                                                   
 Max.   :891.0                                                   
                                                                 
      Age            SibSp           Parch           Ticket         
 Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891        
 1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character  
 Median :28.00   Median :0.000   Median :0.0000   Mode  :character  
 Mean   :29.70   Mean   :0.523   Mean   :0.3816                     
 3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000                     
 Max.   :80.00   Max.   :8.000   Max.   :6.0000                     
 NA's   :177                                                        
      Fare            Cabin             Embarked        
 Min.   :  0.00   Length:891         Length:891        
 1st Qu.:  7.91   Class :character   Class :character  
 Median : 14.45   Mode  :character   Mode  :character  
 Mean   : 32.20                                        
 3rd Qu.: 31.00                                        
 Max.   :512.33                                        
```

💡 Regarding the output generated by summary(df), noticeable modifications arise in the presentation of categorical variables. Instead of reporting statistical measures typically used for continuous data, the resulting table shows frequency counts across factor categories.

Subsetting Data Frames

1. Selecting Specific Columns

The following command selects only the first six columns of the data frame df, creating a new data frame also named df.

```
> df <- df[ ,c(1:6)]
```

2. Selecting based on specific conditions

The following command creates a new data frame called survived containing only the rows where the value in the Survived column is equal to 1

```
> survived <- subset(df,Survived == 1)
```

The following command creates a new data frame called notsurvived containing only the rows where the value in the Survived column is equal to 0

```
> notsurvived <- subset(df,Survived == 0)
```

Pivoting Data Towards Specific Questions

Question 1: Determine the quantity of passengers classified under each combination of gender and ticket class.

```
> table(df$Pclass,df$Sex)

    female male
  1     94  122
  2     76  108
  3    144  347
```

Question 2: Quantify the amount of passengers separated according to their ticket class and survival outcome.

```
> table(df$Pclass,df$Survived)

      0   1
  1  80 136
  2  97  87
  3 372 119
```

Question 3: Estimate the number of passengers divided based on their gender and survival outcome.

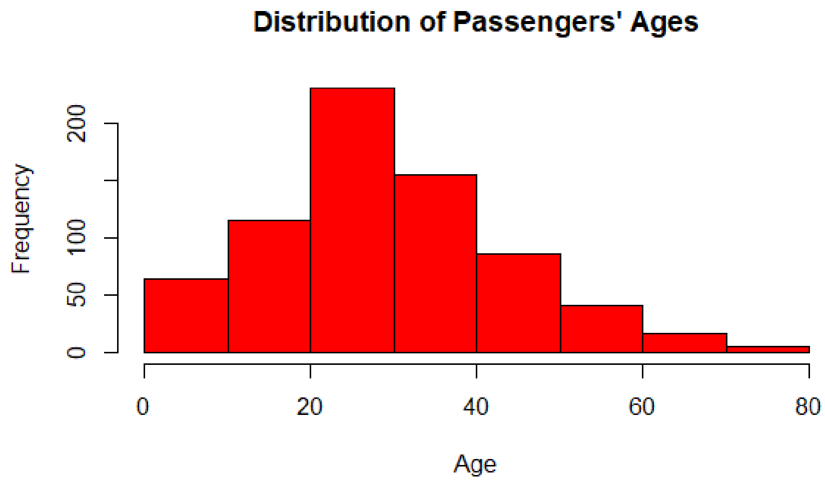
```
> table(df$Sex,df$Survived)

          0   1
  female  81 233
  male   468 109
```

Question 4: Evaluate the number of passengers allocated according to their gender, survival outcome, and ticket class in the Titanic dataset.

```
> table(df$Sex,df$Survived,df$Pclass)
, ,  = 1


          0   1
  female   3  91
  male    77  45

, ,  = 2


          0   1
  female   6  70
  male    91  17

, ,  = 3


          0   1
  female  72  72
  male   300  47
```

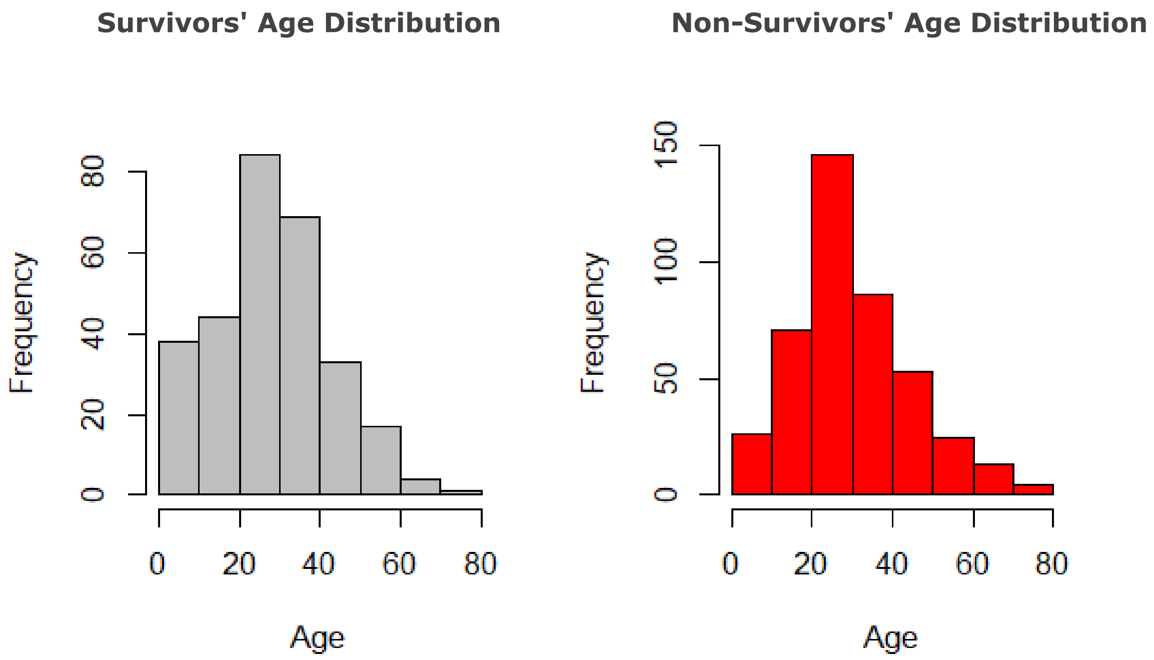
Data Visualization

```
> hist(df$Age, main="Distribution of Passengers' Ages", xlab="Age", col="red")
```



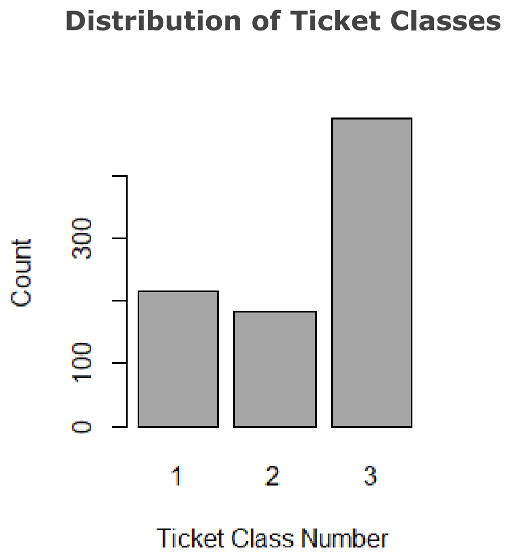
This histogram shows us the distribution of ages among the passengers on board the Titanic. We can see that the majority of passengers were between the ages of 20 and 40.

```
> par(mfrow=c(1,2))
> hist(survived$Age, main="Survivors' Age Distribution", xlab="Age",
col="gray")
> hist(notsurvived$Age, main="Non-Survivors' Age Distribution", xlab="Age",
col="red")
```

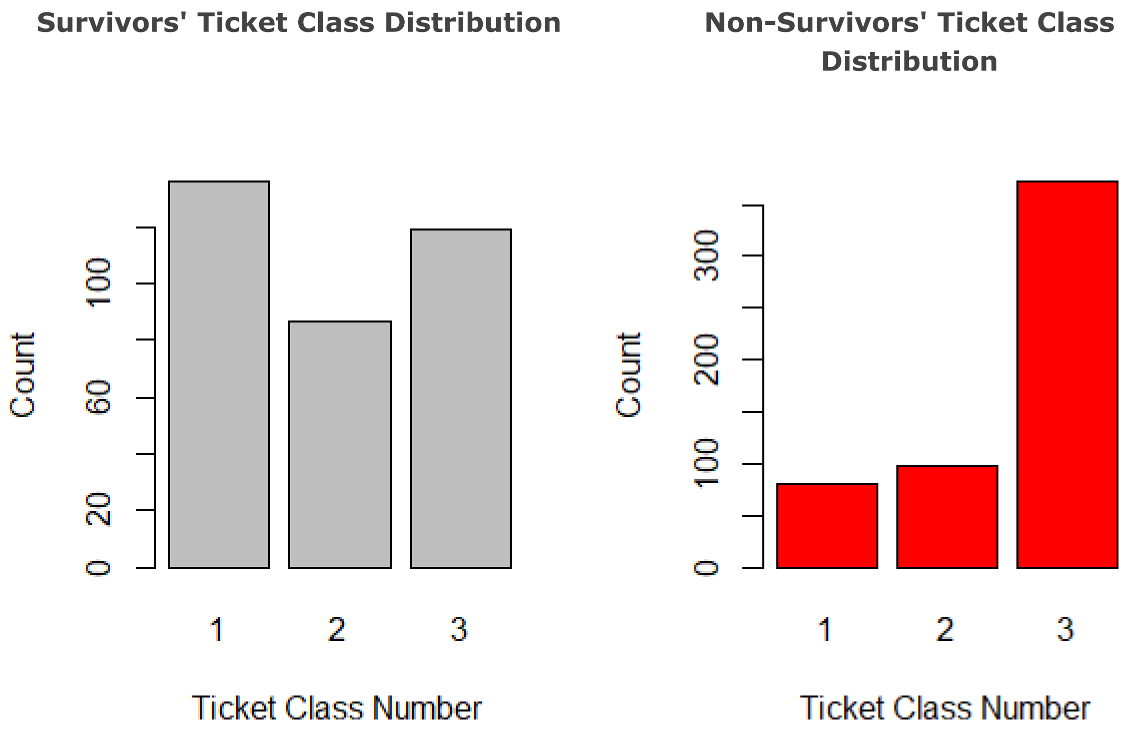


Younger Population Predominant: Majority of the passengers fall under the age range of approximately 20-40 years old, Clear Demarcation Between Survivors and Non-survivors: Survivors generally tend to be younger compared to non-survivors, revealed by the peak in the survivor age distribution occurring notably earlier than the trough in the non-survivor distribution.

```
plot(df$Pclass, main="Distribution of Ticket Classes", xlab="Ticket Class
Number", ylab="Count", col="darkgray")
```



```
> par(mfrow=c(1,2))
> plot(survived$Pclass, main="Survivors' Ticket Class Distribution",
xlab="Ticket Class Number", ylab="Count", col="gray")
> plot(notsurvived$Pclass, main="Non-Survivors' Ticket Class Distribution",
xlab="Ticket Class Number", ylab="Count", col="red")
```



There exists a visible gap between the ticket class distribution of survivors and non-survivors. More survivors belong to 1st and 2nd class compared to non-survivors, implying wealthier passengers had relatively better odds of surviving.