**NAME: Vedashree Bhalerao**
**CLASS: AIA-1**
**BATCH: B**
**ROLL NO: 2213191**

# Big Data Analytics Lab

**AIM: Perform text analysis using R.**

**Theory:**

R Programming Language is used for statistical computing and is used by many data miners and statisticians for developing statistical software and data analysis. It includes machine learning algorithms, linear regression, time series, and statistical inference to name a few. R and its libraries implement a wide variety of statistical and graphical techniques, including linear and non-linear modeling, classical, statistical tests, time-series analysis, classification, clustering, and others.

Any value written inside the double quote is treated as a string in R. String is an array of characters and these collections of characters are stored inside a variable. Internally R stores every string within double quotes, even when you create them with a single quote.

**Text Processing in R**
Using Built-in Type in R
Using Tidyverse module
Using regex and external module
Using grep()

Text analysis (TA) is a machine learning technique used to automatically extract valuable insights from unstructured text data. Companies use text analysis tools to quickly digest online data and documents, and transform them into actionable insights.

You can use text analysis to extract specific information, like keywords, names, or company information from thousands of emails, or categorize survey responses by sentiment and topic.
When you put machines to work on organizing and analyzing your text data, the insights and benefits are huge.

***Steps involved:***

Step 1: Import dataset with setting delimiter
Step 2: Text Cleaning or Preprocessing
- Remove Punctuations, Numbers
- Stemming
- Convert each word into its lower case

Step 3: Tokenization, involves splitting sentences and words from the body of the text.
Step 4: Making the bag of words and analyse the final result.

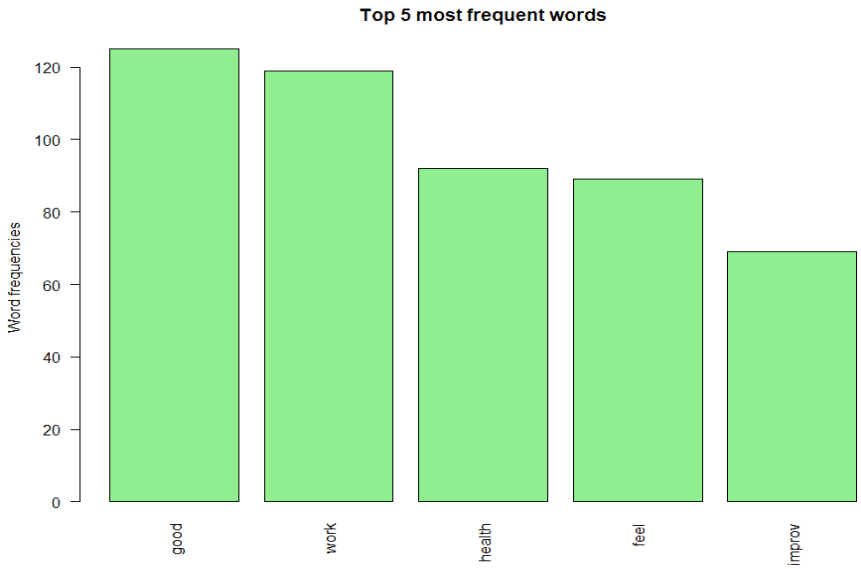
CODE:
To be executed in Colab R notebook with the data set named:
“TeamHealthRawDataForDemo”.Lessen the number of lines for quick execution.

Link:
https://drive.google.com/file/d/1fdydH9UoaOAP6JVCvmmjLrV8atC141e9/view?usp=sharing

```
install.packages("tm")  # for text mining
install.packages("SnowballC") # for text stemming
install.packages("wordcloud") # word-cloud generator
install.packages("RColorBrewer") # color palettes
install.packages("syuzhet") # for sentiment analysis
install.packages("ggplot2") # for plotting graphs
# Load
library("tm")
library("SnowballC")
library("wordcloud")
library("RColorBrewer")
library("syuzhet")
library("ggplot2")


text <- readLines(file.choose())
# Load the data as a corpus
TextDoc <- Corpus(VectorSource(text))


#Replacing "/", "@" and "|" with space
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
TextDoc <- tm_map(TextDoc, toSpace, "/")
TextDoc <- tm_map(TextDoc, toSpace, "@")
TextDoc <- tm_map(TextDoc, toSpace, "\\|")
# Convert the text to lower case
TextDoc <- tm_map(TextDoc, content_transformer(tolower))
# Remove numbers
TextDoc <- tm_map(TextDoc, removeNumbers)
# Remove english common stopwords
TextDoc <- tm_map(TextDoc, removeWords, stopwords("english"))
# Remove your own stop word
# specify your custom stopwords as a character vector
TextDoc <- tm_map(TextDoc, removeWords, c("s", "company", "team"))
# Remove punctuations
TextDoc <- tm_map(TextDoc, removePunctuation)
# Eliminate extra white spaces
TextDoc <- tm_map(TextDoc, stripWhitespace)
```

```
# Text stemming - which reduces words to their root form
TextDoc <- tm_map(TextDoc, stemDocument)


# Build a term-document matrix
TextDoc_dtm <- TermDocumentMatrix(TextDoc)
dtm_m <- as.matrix(TextDoc_dtm)
# Sort by descearing value of frequency
dtm_v <- sort(rowSums(dtm_m),decreasing=TRUE)
dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)
# Display the top 5 most frequent words
head(dtm_d, 5)


# Plot the most frequent words
barplot(dtm_d[1:5,]$freq, las = 2, names.arg = dtm_d[1:5,]$word,
        col ="lightgreen", main ="Top 5 most frequent words",
        ylab = "Word frequencies")

#generate word cloud
set.seed(1234)
wordcloud(words = dtm_d$word, freq = dtm_d$freq, min.freq = 5,
          max.words=100, random.order=FALSE, rot.per=0.40,
          colors=brewer.pal(8, "Dark2"))
```

Output:




**Conclusion:**

We have successfully studied and performed text analysis using R programming.

**CODE:**

```
install.packages("readtext")

require(readtext) # For files import

require(dplyr)

require(tidytext)



x=readtext("*.txt")

x


names(x)

xx=as_tibble(x)

xx$doc_id=c("crow","hare","lion")
```

```
xx

y=unnest_tokens(xx,word,text) %>%
  anti_join(stop_words)
y



# table(stop_words$lexicon)

word= y %>% count(word, sort = TRUE) %>%
  print(n = 10)

#=====================================
#Visualization
#=====================================

library(wordcloud)
par(mar = c(0.1,0.1,0.1,0.1))
word %>% with(wordcloud(word, n,
                        max.words = 100,
                        min.freq = 3,
                        rot.per = .35,
                        random.order = T,
                        random.color = T,
```

```
                colors = rainbow(8)))

#---------------------------------

library(wordcloud2)

wcd <- as.data.frame(word)

par(mar = c(0.1,0.1,0.1,0.1))

wordcloud2(wcd[1:30, ])


#Install and load necessary packages

install.packages(c("tm", "tidytext", "stringr", "ggplot2", "dplyr"))

library(tm)

library(tidytext)

library(stringr)

library(ggplot2)

library(dplyr)


# Sample text data

text_data <- c("R is a programming language for statistical computing and
graphics.",

        "It is widely used & among @ statisticians and data miners.",

        "R is an implementation of the S programming language combined
with lexical scoping semantics inspired by Scheme.",

        "R is highly extensible,hs,asgh, and has many packages available
123456.")
```

```
# Create a corpus
corpus <- Corpus(VectorSource(text_data))

# Preprocess the text

corpus <- tm_map(corpus, content_transformer(toupper))
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords, stopwords("en"))
corpus <- tm_map(corpus, stripWhitespace)

inspect(corpus)
```

**OUTPUT:**

## Practical No: 2-2

## AIM: *Perform data analysis using R programming*

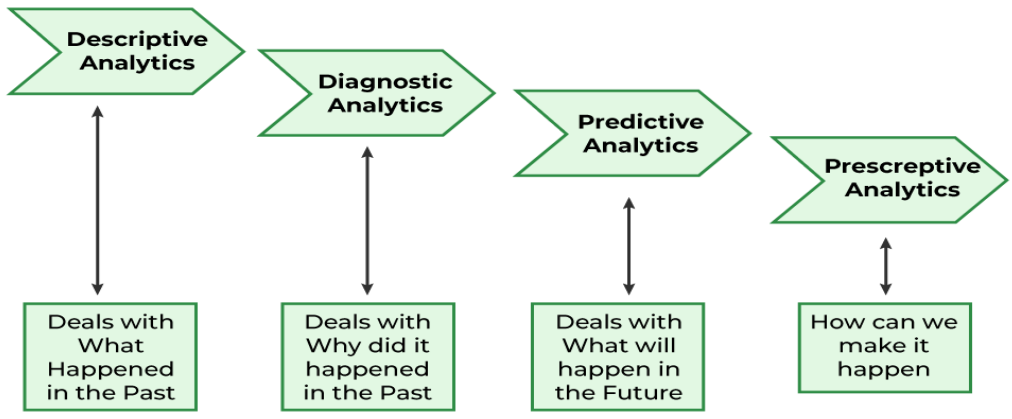
## THEORY:

***Data analysis using R:*** **Data Analysis** is a subset of data analytics, it is a process where the objective has to be made clear, collect the relevant data, preprocess the data, perform analysis(understand the data, explore insights), and then visualize it. The last step visualization is important to make people understand what's happening in the firm.

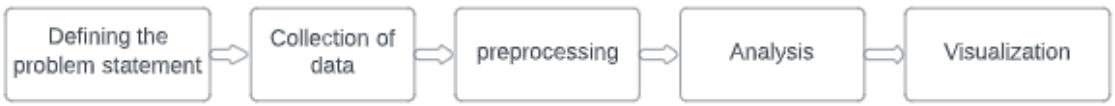
### *Types of Data Analytics*

There are four major types of data analytics:

1. Predictive (forecasting)
2. Descriptive (business intelligence and data mining)
3. Prescriptive (optimization and simulation)
4. Diagnostic analytics



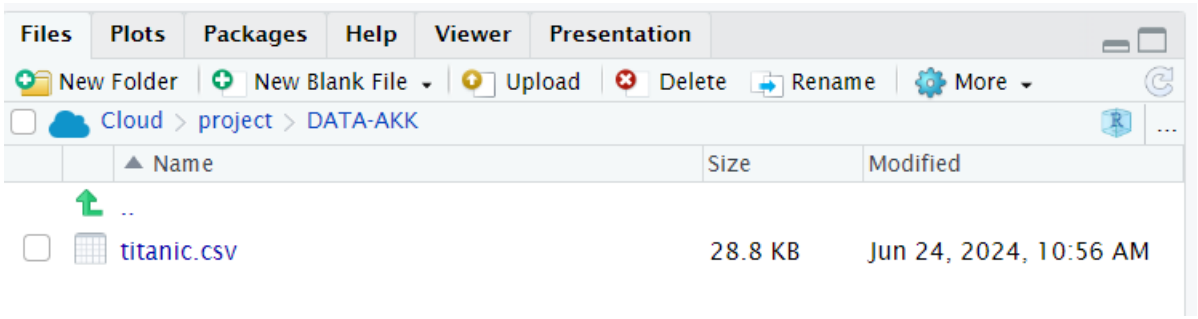

### *Steps in Data Analysis*



***Data Analysis using the Titanic dataset:***

Save the dataset in the current working directory, now we will start analysis (getting to know our data).

*Students can prefer the Free Posit Cloud. Posit Cloud (formerly RStudio Cloud) lets you access Posit's powerful set of data science tools right in your browser – no installation or complex configuration required. And can choose to sign in for free. (read site instructions carefully)*




**titanic=read.csv("train.csv")**

**head(titanic)**

```
> titanic=read.csv("titanic.csv")
> head(titanic)
  PassengerId Survived Pclass                                                Name    Sex
1         892        0      3                                    Kelly, Mr. James   male
2         893        1      3                    Wilkes, Mrs. James (Ellen Needs) female
3         894        0      2                           Myles, Mr. Thomas Francis   male
4         895        0      3                                    Wirz, Mr. Albert   male
5         896        1      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female
6         897        0      3                          Svensson, Mr. Johan Cervin   male
   Age SibSp Parch  Ticket    Fare Cabin Embarked
1 34.5     0     0  330911  7.8292              Q
2 47.0     1     0  363272  7.0000              S
3 62.0     0     0  240276  9.6875              Q
4 27.0     0     0  315154  8.6625              S
5 22.0     1     1 3101298 12.2875              S
6 14.0     0     0    7538  9.2250              S
```

To understand the class(data type) of each column **sapply()** method can be used.

**sapply(titanic,class)**

```
> sapply(titanic, class)
PassengerId    Survived      Pclass        Name         Sex         Age       SibSp
  "integer"   "integer"   "integer" "character" "character"   "numeric"   "integer"
       Parch      Ticket        Fare       Cabin    Embarked
  "integer" "character"   "numeric" "character" "character"
> |
```

To analyze data using a summary of all the columns, their values, and data types. summary() can be used for this purpose.

**summary(titanic)**

```
> summary(titanic)
  PassengerId       Survived          Pclass          Name
 Min.   : 892.0   Min.   :0.0000   Min.   :1.000   Length:418
 1st Qu.: 996.2   1st Qu.:0.0000   1st Qu.:1.000   Class :character
 Median :1100.5   Median :0.0000   Median :3.000   Mode  :character
 Mean   :1100.5   Mean   :0.3636   Mean   :2.266
 3rd Qu.:1204.8   3rd Qu.:1.0000   3rd Qu.:3.000
 Max.   :1309.0   Max.   :1.0000   Max.   :3.000

     Sex                 Age            SibSp            Parch
 Length:418         Min.   : 0.17   Min.   :0.0000   Min.   :0.0000
 Class :character   1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000
 Mode  :character   Median :27.00   Median :0.0000   Median :0.0000
                    Mean   :30.27   Mean   :0.4474   Mean   :0.3923
                    3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000
                    Max.   :76.00   Max.   :8.0000   Max.   :9.0000
                    NA's   :86
    Ticket               Fare            Cabin             Embarked
 Length:418         Min.   :  0.000   Length:418         Length:418
 Class :character   1st Qu.:  7.896   Class :character   Class :character
 Mode  :character   Median : 14.454   Mode  :character   Mode  :character
                    Mean   : 35.627
                    3rd Qu.: 31.500
                    Max.   :512.329
                    NA's   :1
```

From the above summary Students to extract below observations:

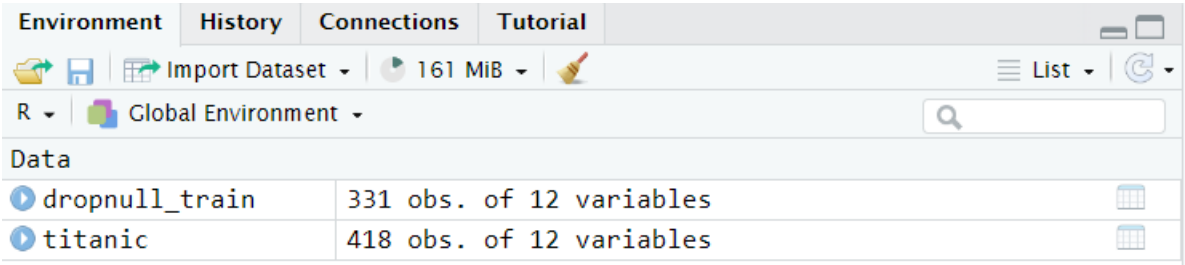
- Total passengers: 891
- The number of total people who survived: 342
- Number of total people dead: 549
- Number of males in the titanic: 577
- Number of females in the titanic: 314
- Maximum age among all people in titanic: 80
- Median age: 28

Preprocessing of the data is important before analysis, so null values have to be checked and removed.

**sum(is.na(train))**

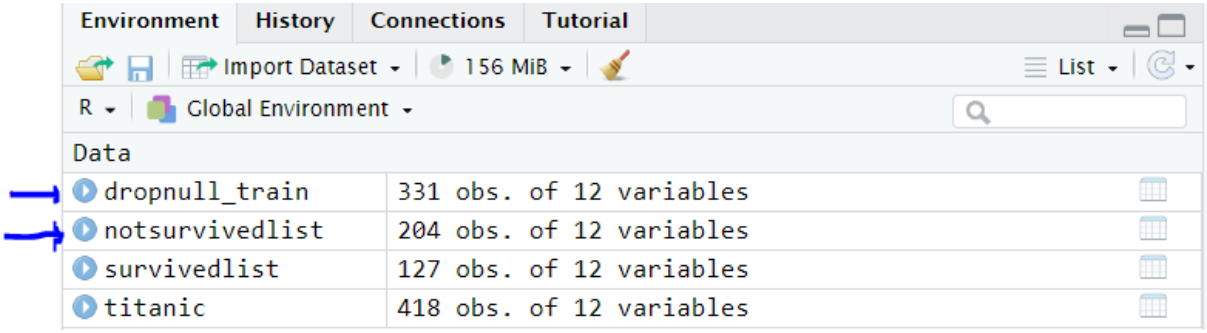
**dropnull_train=titanic[rowSums(is.na(titanic))<=0,]**

- dropnull_train contains only 331 rows because (total rows in dataset (418) – null value rows (87) = remaining rows (331) )
- Now lets will divide survived and dead people into a separate list from 331 rows.




**survivedlist=dropnull_train[dropnull_train$Survived == 1,]**
**notsurvivedlist=dropnull_train[dropnull_train$Survived == 0,]**

```
> survivedlist=dropnull_train[dropnull_train$Survived == 1,]
> notsurvivedlist=dropnull_train[dropnull_train$Survived == 0,]
> 
```




**Visualization:**

Now to visualize the number of males and females dead and survived using *bar plots, histograms, and piecharts.*

**Bar charts** are a popular and effective way to visually represent categorical data in a structured manner. R stands out as a powerful programming language for data analysis and visualization.

A bar chart also known as bar graph is a pictorial representation of data that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. In other words, it is the pictorial representation of the dataset. These data sets contain the numerical values of variables that represent the length or height.

R uses the barplot() function to create bar charts. Here, both vertical and Horizontal bars can be drawn.

**Syntax: barplot(H, xlab, ylab, main, names.arg, col, horiz = TRUE)**

Parameters:

H: This parameter is a vector or matrix containing numeric values which are used in bar chart.

xlab: This parameter is the label for x axis in bar chart.

ylab: This parameter is the label for y axis in bar chart.

main: This parameter is the title of the bar chart.

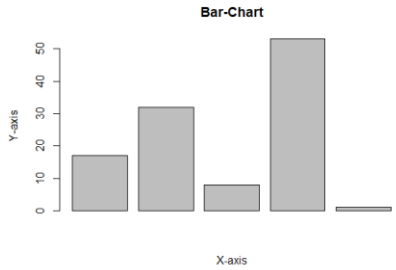
names.arg: This parameter is a vector of names appearing under each bar in bar chart.

col: This parameter is used to give colors to the bars in the graph.

horizontal = TRUE

Ex:

```
# Create the data for the chart
A <- c(17, 32, 8, 53, 1)

# Plot the bar chart
barplot(A, xlab = "X-axis", ylab = "Y-axis", main ="Bar-Chart")
```



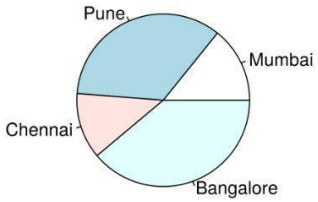

A **pie chart** is a circular statistical graphic, which is divided into slices to illustrate numerical proportions. It depicts a special chart that uses "pie slices", where each sector shows the relative sizes of data. A circular chart cuts in the form of radii into segments describing relative frequencies or magnitude also known as a circle graph. R Programming Language uses the function pie() to create pie charts. It takes positive numbers as a vector input.

**Syntax: pie(x, labels, radius, main, col, clockwise)**

Parameters:
x: This parameter is a vector that contains the numeric values which are used in the pie chart.
labels: This parameter gives the description to the slices in pie chart.
radius: This parameter is used to indicate the radius of the circle of the pie chart.(value between -1 and +1).
main: This parameter is represents title of the pie chart.
clockwise: This parameter contains the logical value which indicates whether the slices are drawn clockwise or in anti clockwise direction.
col: This parameter give colors to the pie in the graph.

Ex:

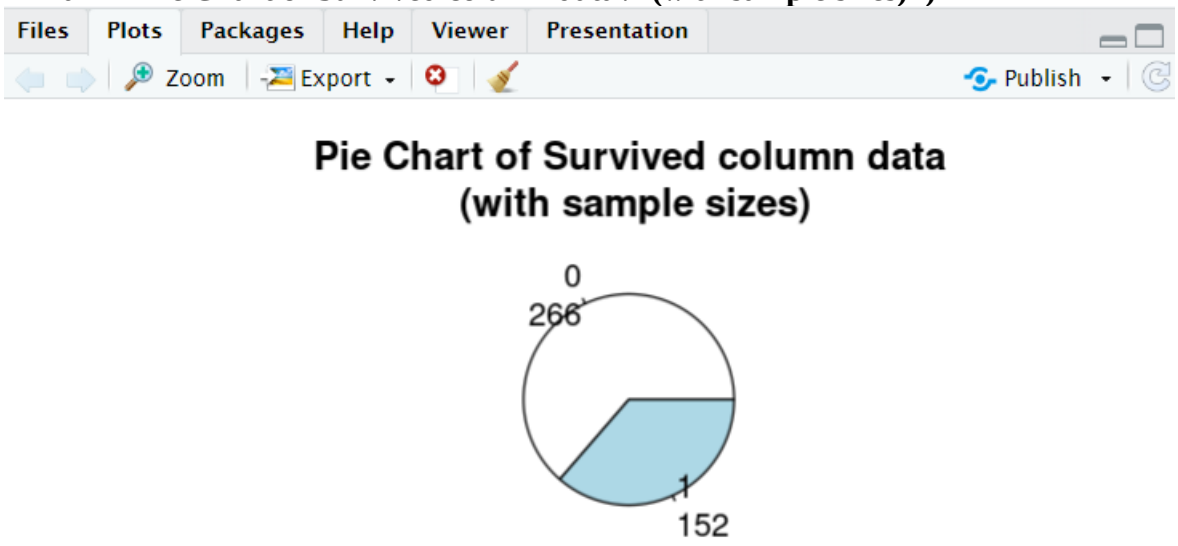
```
# Create data for the graph.
Count<- c(23, 56, 20, 63)
labels <- c("Mumbai", "Pune", "Chennai", "Bangalore")

# Plot the chart.
pie(count, labels)
```




For the Titanic data set, creating a pie chart to visualize the number of males and females dead and survived.

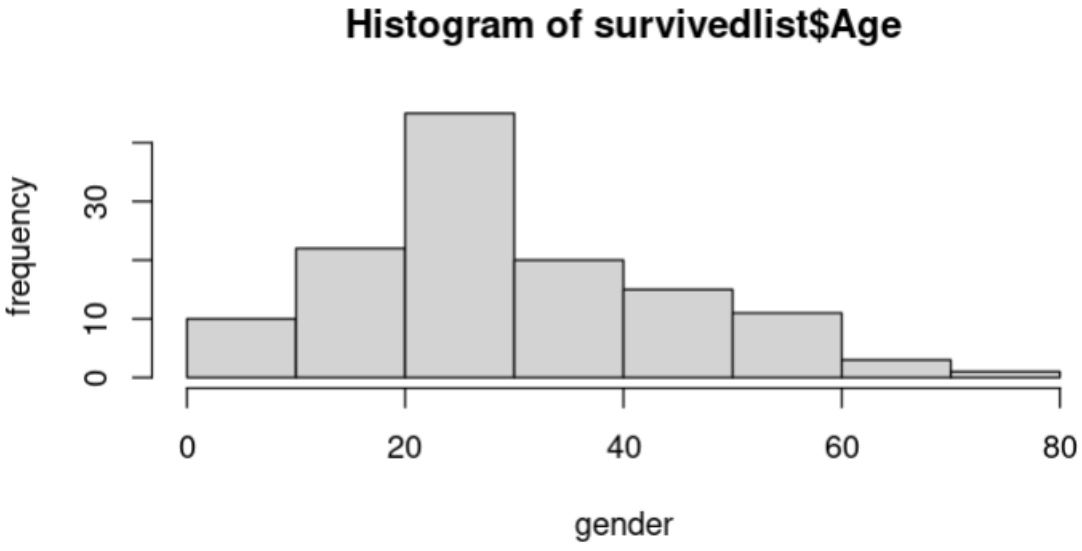
```
mytable <- table(titanic$Survived)
lbls <- paste(names(mytable), "\n", mytable, sep="")
pie(mytable,
   labels = lbls,
```

**main="Pie Chart of Survived column data\n (with sample sizes)")**




**hist(survivedlist$Age,**
**xlab="gender",**
**ylab="frequency")**

```
> hist(survivedlist$Age,
+      xlab="gender",
+      ylab="frequency")
> 
```




**barplot(table(notsurvivedlist$Sex),**
**xlab="gender",**
**ylab="frequency")**

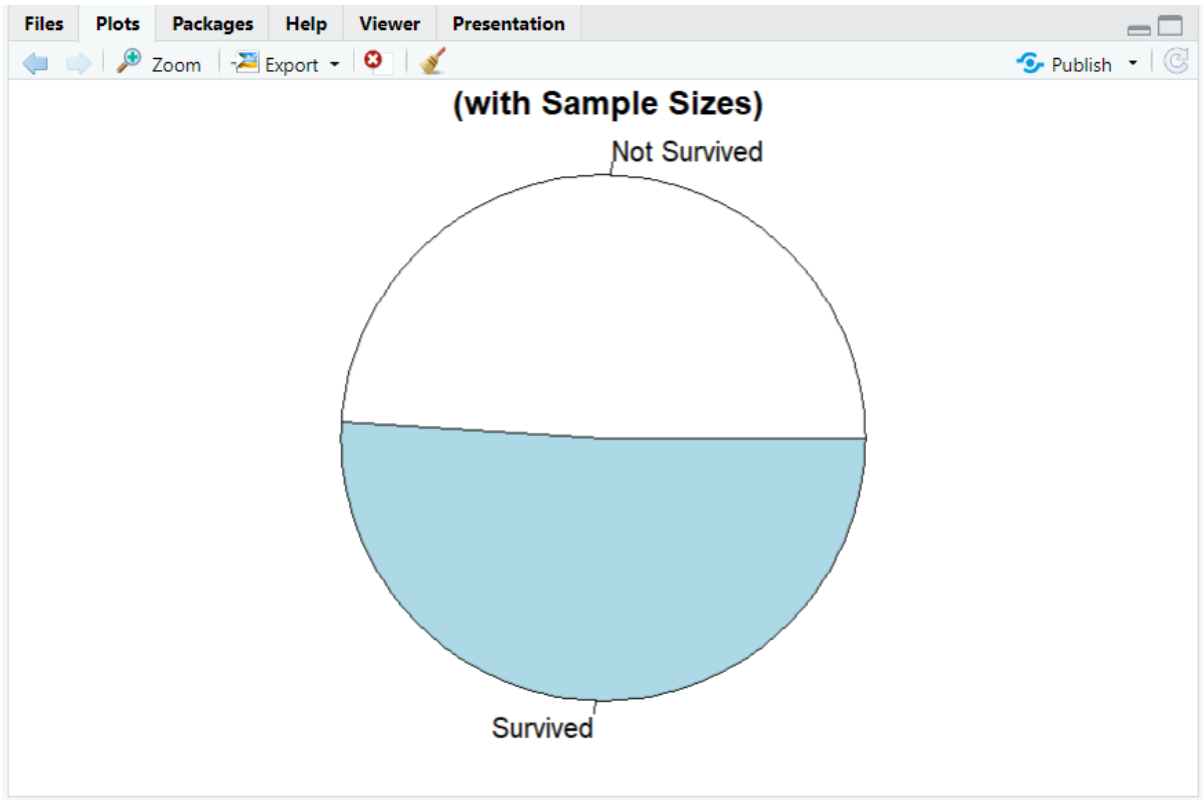
**Conclusion:**

We have successfully studied and performed Data Analysis and visualise it using R Programming.

## CODE:

```
titanic=read.csv("tested.csv")
head(titanic)

sapply(titanic,class)

summary(titanic)


sum(is.na(titanic))
dropnull_train=titanic[rowSums(is.na(titanic))<=0,]


survivedlist=dropnull_train[dropnull_train$Survived == 1,]
notsurvivedlist=dropnull_train[dropnull_train$Survived == 0,]

# Sample Data
mytable <- table(sample(c("Survived", "Not Survived"), 100, replace = TRUE))
lbls <- names(mytable)

# Check the data
print(mytable)
print(lbls)

# Ensure there are no NA values
mytable <- na.omit(mytable)

# Create the Pie Chart
pie(mytable, labels = lbls, main = "Pie Chart of Survived Column Data\n (with
Sample Sizes)")
```

**OUTPUT:**


Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
(with Sample Sizes)
Not Survived
Survived