BIG DATA ANALYTICS
COURSE CODE:
CREDITS: 3

UNIT-1

Dr.M.Amsaprabhaa

Assistant Professor

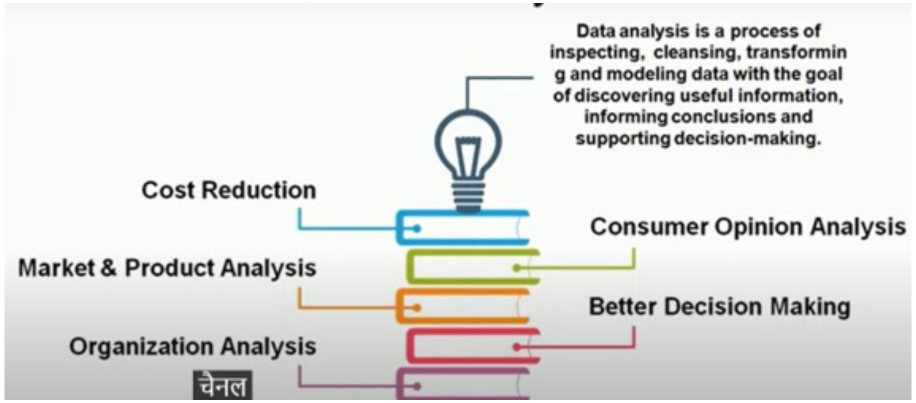
Department of CSE

Shiv Nadar University Chennai

Data Analytics using R Programming

- The R programming language is named after the first letters of the names of its inventors, Ross Ihaka and Robert Gentleman.

Data Analytics



Data Visualization

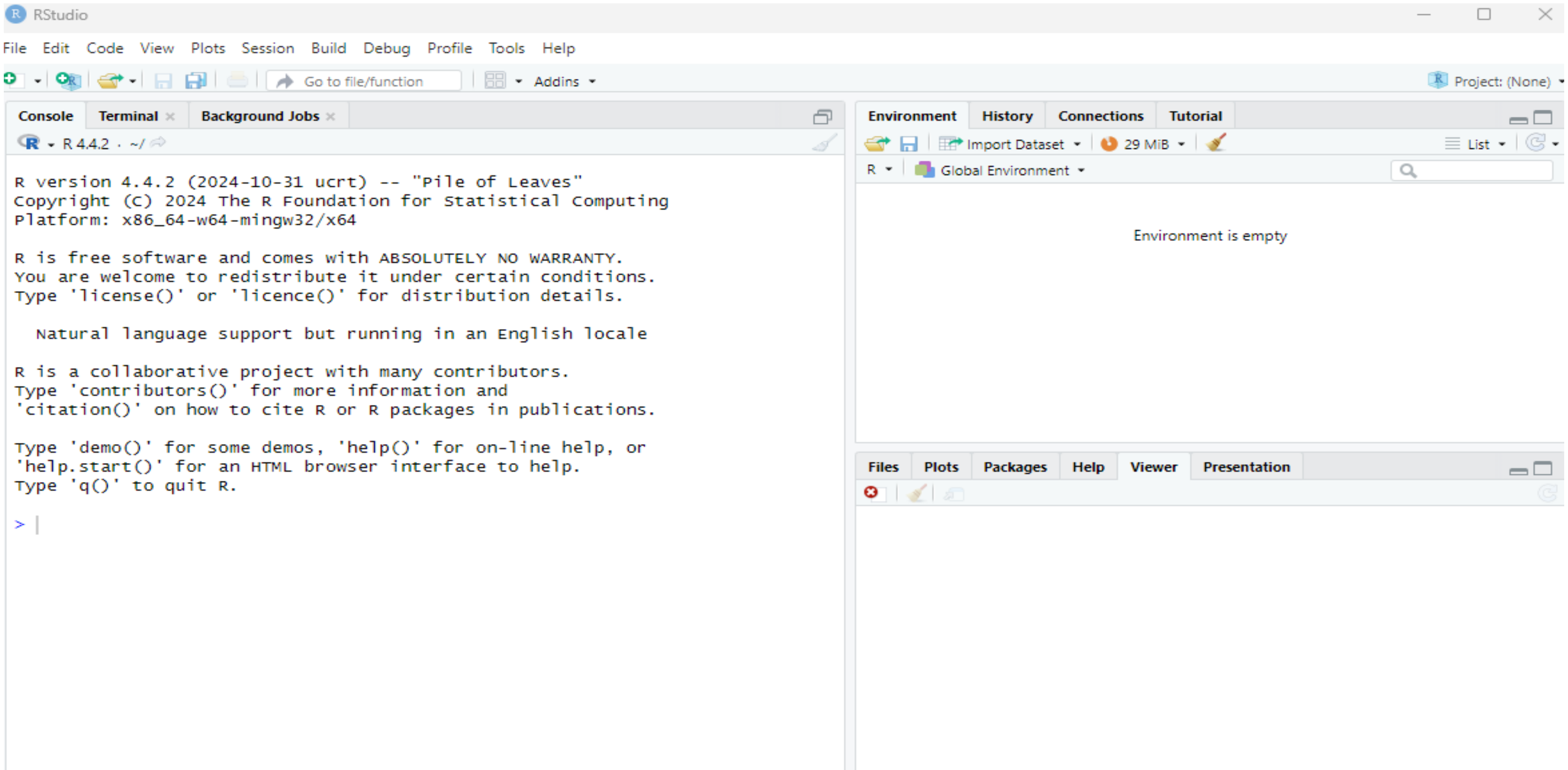
Data visualization is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from. The main goal of data visualization is to make it easier to identify patterns, trends and outliers in large data sets.

What is R Programming? & Why R Programming?

- R is a Programming language
- Environment for statistical computing and graphics.
- Well-designed publication-quality plots can be produced
- An effective data handling and storage facility
- A large, coherent, integrated collection of intermediate tools for data analysis
- Graphical facilities for data analysis and display either on-screen or on hardcopy

R Installation

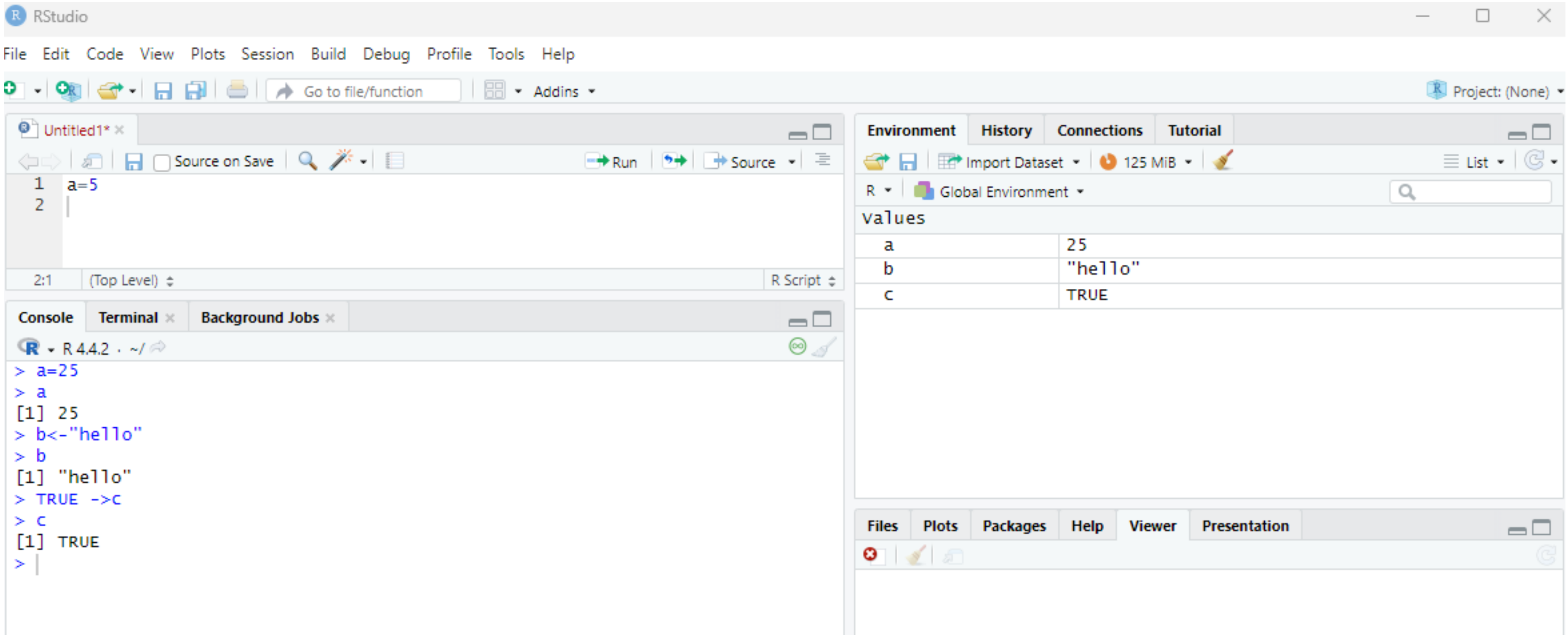
After R Installation

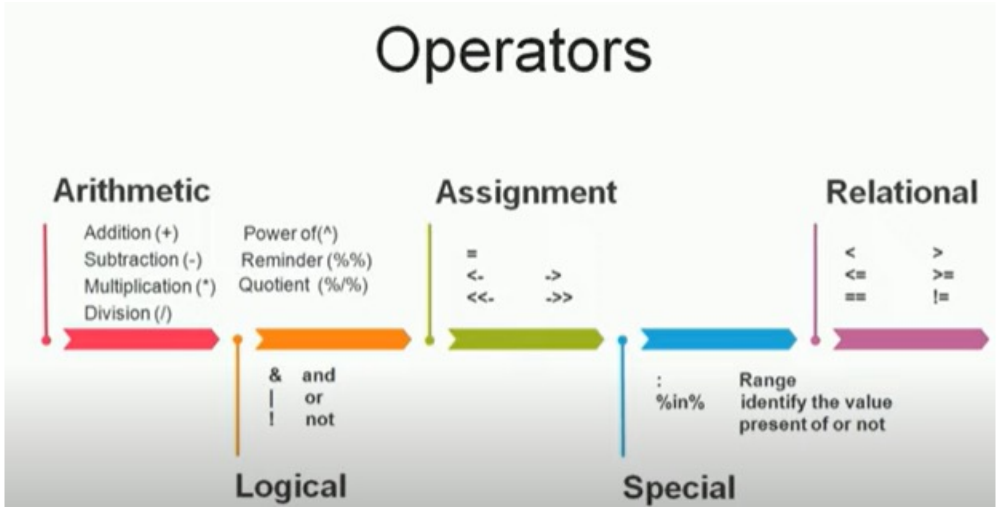


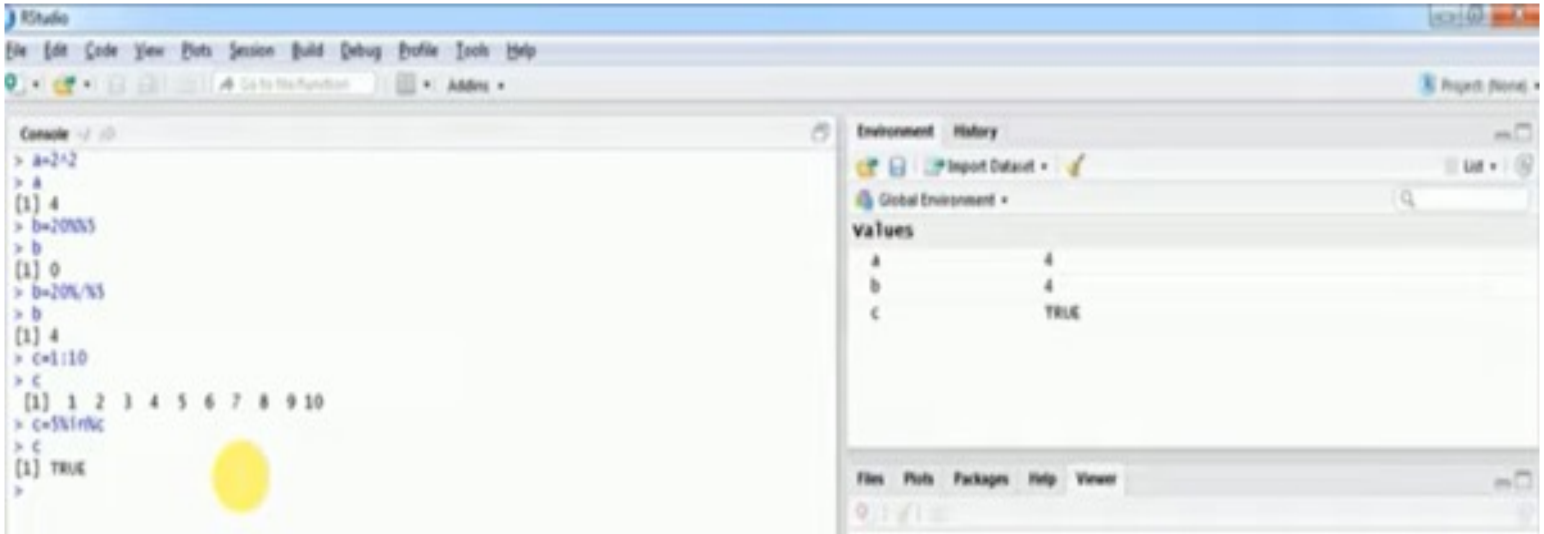
Variables in R

A **variable** is a name given to a memory location, which is used to store values in a computer **program**. **Variables** in **R programming** can be used to store numbers (real and complex), words, matrices, and even tables.

Variables in R




Operators
Arithmetic
Addition (+)
Subtraction (-)
Multiplication (*)
Division (/)
Power of(^)
Reminder (%%)
Quotient (%/%)
Assignment
=
<-
<<-
->
->>
Relational
<
<=
==
>
>=
!=
& and
| or
! not
Logical
:
%in%
Range
identify the value present of or not
Special


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Addins
Console
> a=2^2
> a
[1] 4
> b=20%%5
> b
[1] 0
> b=20%/%5
> b
[1] 4
> c=1:10
> c
[1] 1 2 3 4 5 6 7 8 9 10
> c=5%in%c
> c
[1] TRUE
>
Environment History
Import Dataset
Global Environment
Values
a 4
b 4
c TRUE
Files Plots Packages Help Viewer

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 process of data analysis would include all these steps for the given problem statement.
Example- Analyze the products that are being rapidly sold out and details of frequent customers of a retail shop.

- Defining the problem statement – Understand the goal, and what is needed to be done. In this case, our problem statement is – “The product is mostly sold out and list of customers who often visit the store.”
- Collection of data – Not all the company’s data is necessary, understand the relevant data according to the problem. Here the required columns are product ID, customer ID, and date visited.
- Preprocessing – Cleaning the data is mandatory to put it in a structured format before performing analysis.

DATA ANALYSIS USING R

- Removing outliers(noisy data).
- Removing null or irrelevant values in the columns. (Change null values to mean value of that column.)
- If there is any missing data, either ignore the tuple or fill it with a mean value of the column.

<u>Data Analysis using the Titanic dataset</u>

- You can download the titanic dataset (it contains data from real passengers of the titanic)from https://drive.google.com/file/d/15db6BpWU3NBi8LK0paPA7USQicF9URB2/view
- Save the dataset in the current working directory, now we will start analysis (getting to know our data).

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

Output:

```
  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
```

Our dataset contains all the columns like name, age, gender of the passenger and class they have traveled in, whether they have survived or not, etc. To understand the class(data type) of each column **sapply()** method can be used.

DATA ANALYSIS USING R

sapply(train, class)

Output:

```
PassengerId    Survived      Pclass        Name         Sex         Age 
  "integer"   "integer"   "integer" "character" "character"   "numeric" 
      SibSp       Parch      Ticket        Fare       Cabin    Embarked 
  "integer"   "integer" "character"   "numeric" "character" "character"
```

We can categorize the value **"survived"** into **"dead"** to 0 and **"alive"** to 1 using **factor()** function.

train$Survived=as.factor(train$Survived)
train$Sex=as.factor(train$Sex)
sapply(train, class)

Output:

```
PassengerId    Survived      Pclass        Name         Sex         Age 
  "integer"    "factor"   "integer" "character"    "factor"   "numeric" 
      SibSp       Parch      Ticket        Fare       Cabin    Embarked 
  "integer"   "integer" "character"   "numeric" "character" "character"
```

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

DATA ANALYSIS USING R

summary(train)

Output:

```
  PassengerId       Survived     Pclass          Name                Sex
 Min.   : 892.0   0:266    Min.   :1.000   Length:418         female:152
 1st Qu.: 996.2   1:152    1st Qu.:1.000   Class :character   male  :266
 Median :1100.5            Median :3.000   Mode  :character
 Mean   :1100.5            Mean   :2.266
 3rd Qu.:1204.8            3rd Qu.:3.000
 Max.   :1309.0            Max.   :3.000

      Age             SibSp            Parch            Ticket
 Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   Length:418
 1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
 Median :27.00   Median :0.0000   Median :0.0000   Mode  :character
 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
      Fare              Cabin             Embarked
 Min.   :  0.000   Length:418         Length:418
 1st Qu.:  7.896   Class :character   Class :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 we can 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

DATA ANALYSIS USING R

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

```
sum(is.na(train))
```

```
Output:
177
```

```
dropnull_train=train[rowSums(is.na(train))<=0,]
```

- dropnull_train contains only 631 rows because **(total rows in dataset (808) – null value rows (177) = remaining rows (631))**
- Now we will divide survived and dead people into a separate list from 631 rows.

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

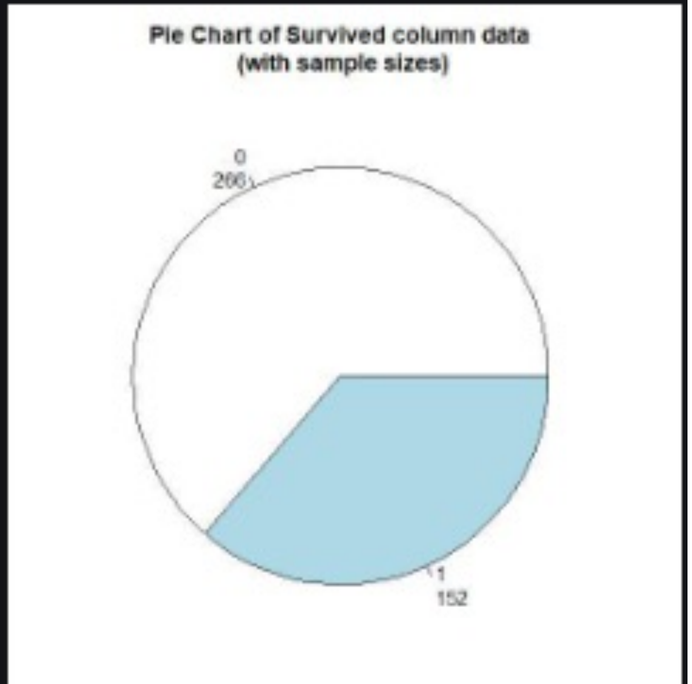
DATA ANALYSIS USING R

Now we can visualize the number of males and females dead and survived using

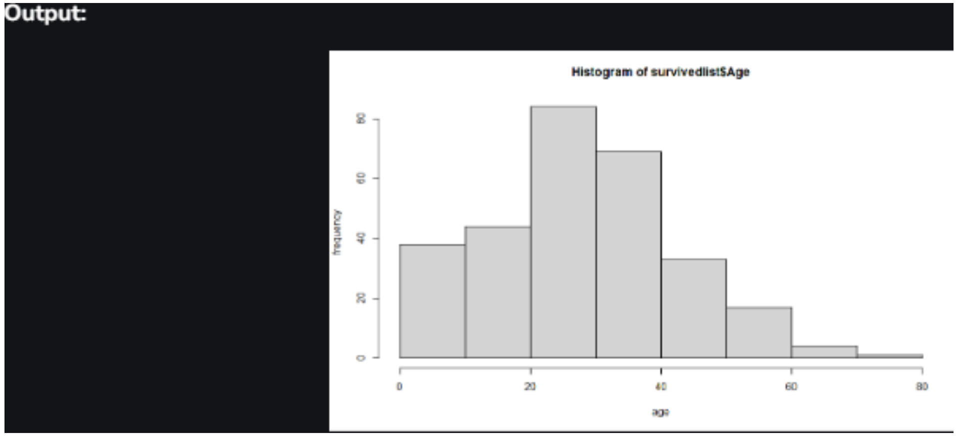
- bar plots (https://www.geeksforgeeks.org/r-bar-charts/)
- histograms (https://www.geeksforgeeks.org/histograms-in-r-language/)
- piecharts (https://www.geeksforgeeks.org/r-pie-charts/)

```
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)")
```

Output:


Pie Chart of Survived column data
(with sample sizes)
0
266
1
152

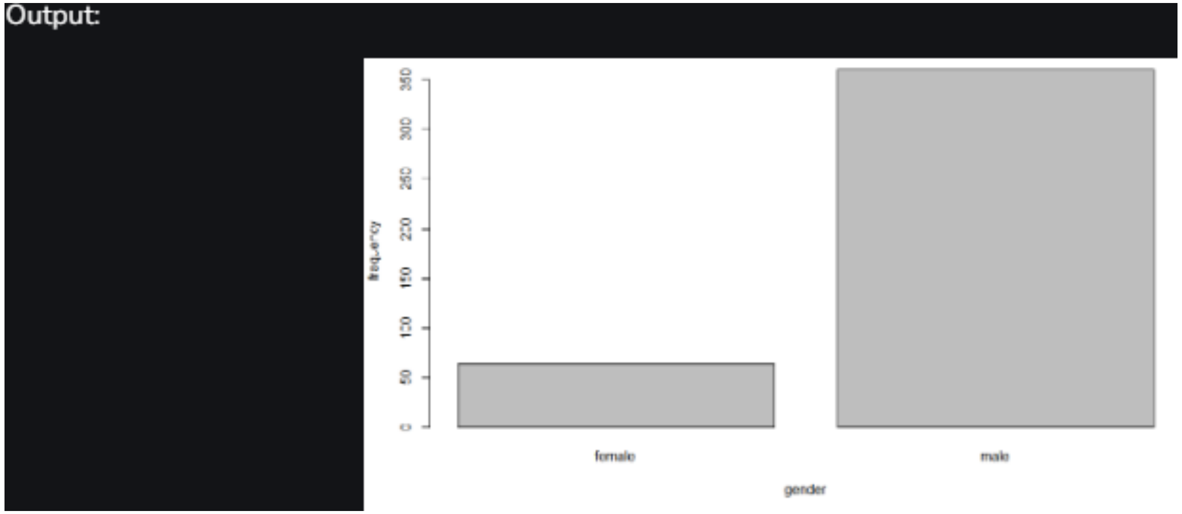
```
hist(survivedlist$Age,
    xlab="gender",
    ylab="frequency")
```



Now let’s draw a bar plot to visualize the number of males and females who were there on the titanic ship.

DATA ANALYSIS USING R

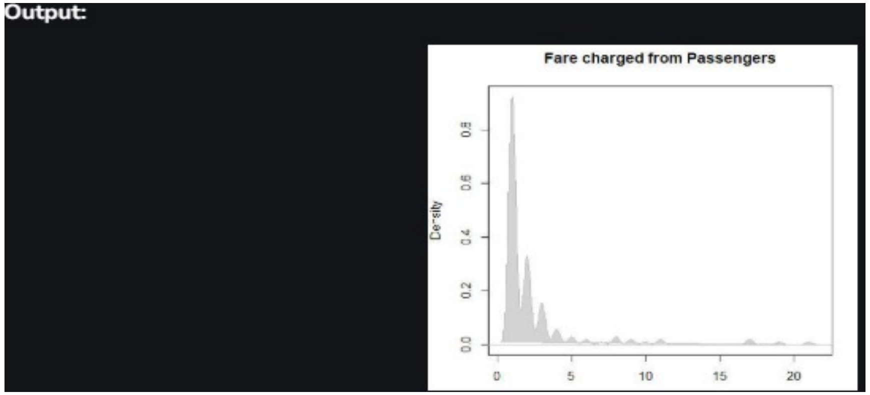
```
barplot(table(notsurvivedlist$Sex),
      xlab="gender",
      ylab="frequency")
```



From the barplot above we can analyze that there are nearly 350 males, and 50 females those are not survived in titanic.

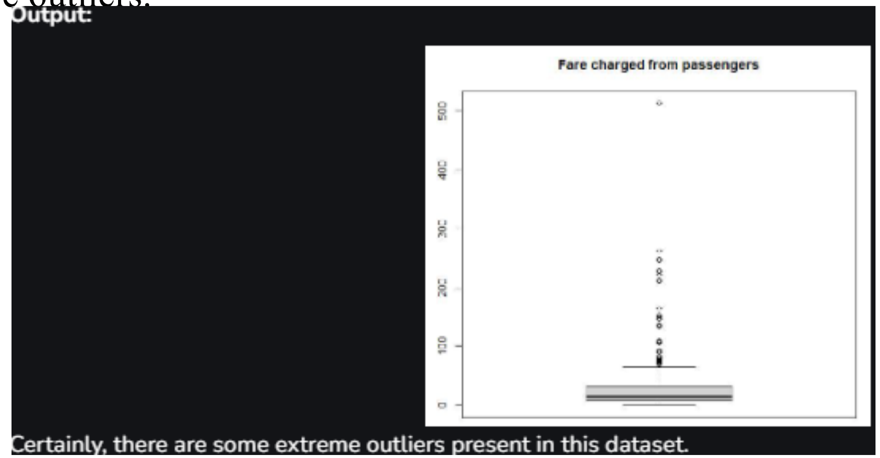
DATA ANALYSIS USING R

```
temp<-density(table(titanic$Fare))
plot(temp, type="n",
    main="Fare charged from Passengers")
polygon(temp, col="lightgray",
      border="gray")
```



- Here we can observe that there are some passengers who are charged extremely high.
- So, these values can affect our analysis as they are outliers.
- Let’s confirm their presence using a boxplot.

```
boxplot(titanic$Fare,
      main="Fare charged from passengers")
```



DATA ANALYSIS USING R

Performing Clustering

- Import required R libraries for data manipulation, clustering, and visualization.
- Read the Titanic dataset from the specified file path.
- Keep only the relevant columns for analysis.
- Transform categorical variables into numeric format using factor() function.
- Impute missing values for Age with the mean value and ensure there are no remaining missing values.
- Remove any rows with remaining missing values if necessary.
- Scale the data to ensure that all features contribute equally to clustering.
- Verify that the standardized data does not contain NaN or Inf values, which could affect clustering.
- Use the Elbow Method to find the optimal number of clusters; if this method fails, manually try different values.
- Run the K-means algorithm with the chosen number of clusters (e.g., k = 3) and a set seed for reproducibility.
- Ensure that the K-means clustering results have been successfully created.
- Append the cluster assignments to the original dataset for further analysis.
- Create a cluster plot to visualize the results of the K-means clustering.

DATA ANALYSIS USING R

```
# Load necessary libraries
library(tidyverse)
library(cluster)
library(factoextra)

# Load the dataset
titanic_data <- read.csv("C:/Users/Tonmoy/Downloads/titanic.csv")

# Data preprocessing
# Remove unnecessary columns
titanic_data <- titanic_data %>%
  select(PassengerId, Survived, Pclass, Sex, Age, SibSp, Parch, Fare)

# Convert categorical variables to numeric
titanic_data$Sex <- as.numeric(factor(titanic_data$Sex))

# Handle missing values
# Impute missing values for Age
titanic_data$Age[is.na(titanic_data$Age)] <- mean(titanic_data$Age, na.rm = TRUE)

# Ensure there are no remaining missing values
missing_values <- sum(is.na(titanic_data))
print(paste("Remaining missing values after imputation:", missing_values))
```

DATA ANALYSIS USING R

```
# If missing values still exist, handle them (e.g., impute with mean or remove rows/columns)
if (missing_values > 0) {
  # Remove rows with remaining missing values (if any)
  titanic_data <- na.omit(titanic_data)
}

# Standardize the data
titanic_scaled <- scale(titanic_data)

# Check for NaNs or Infs in the scaled data
if (any(is.nan(titanic_scaled)) || any(is.infinite(titanic_scaled))) {
  stop("Scaled data contains NaN or Inf values")
}

# Double-check the data for any anomalies
print(summary(titanic_scaled))

# Determine the optimal number of clusters
# If Elbow method still fails, manually try different k values
fviz_nbclust(titanic_scaled, kmeans, method = "wss") + labs(subtitle = "Elbow Method")
```

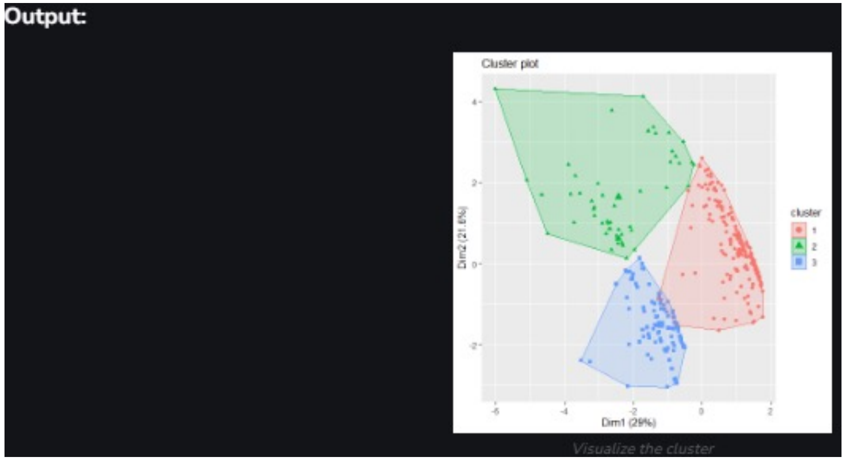
DATA ANALYSIS USING R

```
# Perform K-means clustering with the optimal number of clusters (e.g., k = 3)
set.seed(123)
kmeans_result <- kmeans(titanic_scaled, centers = 3, nstart = 25)

# Check if kmeans_result was created successfully
if (!exists("kmeans_result")) {
  stop("K-means clustering failed to create kmeans_result")
}

# Add cluster assignments to the original dataset
titanic_data$Cluster <- kmeans_result$cluster

# Visualize the clustering
fviz_cluster(kmeans_result, data = titanic_scaled, geom = "point", stand = FALSE)
```



DATA ANALYSIS USING R

Predictive Model

•Load a collection of R packages for data manipulation and visualization. It includes dplyr and ggplot2, among others.

•The caret package for training and evaluating machine learning models. It provides functions for data splitting (createDataPartition), model training (train), and performance evaluation (confusionMatrix).

•Loads the dataset from a specified file path into a data frame.

•Chooses relevant columns from the dataset to use for modeling. Here, it selects columns related to survival status and passenger features.

•Converts the Sex variable into a factor, which is then converted to numeric values. This is necessary because logistic regression models require numerical input.

•Computes the mean age (excluding missing values) to impute the missing values in the Age column.

•Counts remaining missing values.

•Removes rows with any remaining missing values.

•Converts the Survived variable to a factor. This is essential for classification tasks in logistic regression.

•Ensures reproducibility of the data split.

•Creates an 80-20 split of the data into training and testing sets.

•Trains a logistic regression model (method = “glm”) with a binomial family for binary classification.

•Generates predictions on the test set using the trained model.

•Ensures that the factor levels of titanic_test$Survived match those of predictions.

•Computes and prints the confusion matrix to evaluate model performance.

DATA ANALYSIS USING R

```
# Load necessary libraries
library(tidyverse)
library(caret)  # For createDataPartition, train, and confusionMatrix

# Load the dataset
titanic_data <- read.csv("C:/Users/Tonmoy/Downloads/titanic.csv")

# Data preprocessing
titanic_data <- titanic_data %>%
  select(Survived, Pclass, Sex, Age, SibSp, Parch, Fare)

# Convert categorical variables to numeric
titanic_data$Sex <- as.numeric(factor(titanic_data$Sex))

# Handle missing values by imputing with mean for Age
titanic_data$Age[is.na(titanic_data$Age)] <- mean(titanic_data$Age, na.rm = TRUE)

# Check for remaining missing values
missing_values <- sum(is.na(titanic_data))
print(paste("Remaining missing values after imputation:", missing_values))
```

DATA ANALYSIS USING R

```
# Remove rows with remaining missing values (if any)
if (missing_values > 0) {
  titanic_data <- na.omit(titanic_data)
}

# Convert Survived to factor for classification
titanic_data$Survived <- as.factor(titanic_data$Survived)

# Split the data into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(titanic_data$Survived, p = .8, list = FALSE)
titanic_train <- titanic_data[trainIndex, ]
titanic_test  <- titanic_data[-trainIndex, ]

# Train a logistic regression model
model <- train(Survived ~ ., data = titanic_train, method = "glm", family = binomial)
```

DATA ANALYSIS USING R

```
# Make predictions on the test set
predictions <- predict(model, titanic_test)

# Ensure both factors have the same levels
levels(titanic_test$Survived) <- levels(predictions)

# Evaluate the model
conf_matrix <- confusionMatrix(predictions, titanic_test$Survived)
print(conf_matrix)
```

Output:

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 53  0
         1  0 30

               Accuracy : 1
                 95% CI : (0.9565, 1)
    No Information Rate : 0.6386
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.6386
         Detection Rate : 0.6386
   Detection Prevalence : 0.6386
      Balanced Accuracy : 1.0000

       'Positive' Class : 0
```

DATA ANALYSIS USING R

Explanation of the output –

Accuracy: 100% – The model predicts all test cases correctly.

Sensitivity: 100% – The model identifies all positive cases correctly.

Specificity: 100% – The model identifies all negative cases correctly.

Kappa: 1 – A measure of agreement between the predicted and observed classifications.

The code trains a logistic regression model to predict survival based on various features of the Titanic dataset.

The model shows perfect accuracy, but this might be due to issues in the data or its split. Further validation is recommended.