Experiment No.8
Data Visualization using Hive/PIG/**R**/Tableau/.
Date of Performance:
Date of Submission:

AIM : Data Visualization using Hive/PIG/**R**/Tableau/.

THEORY :

Data visualisation is the technique used to deliver insights in data using visual cues such as graphs, charts, maps, and many others. This is useful as it helps in intuitive and easy understanding of the large quantities of data and thereby make better decisions regarding it.

The popular data visualisation tools that are available are Tableau, Plotly, R, Google Charts, Infogram, and Kibana. The various data visualisation platforms have different capabilities, functionality, and use cases. They also require a different skill set. This article discusses the use of R for data visualisation.

R is a language that is designed for statistical computing, graphical data analysis, and scientific research. It is usually preferred for data visualisation as it offers flexibility and minimum required coding through its packages.

Consider the following air quality data set for visualisation in R:

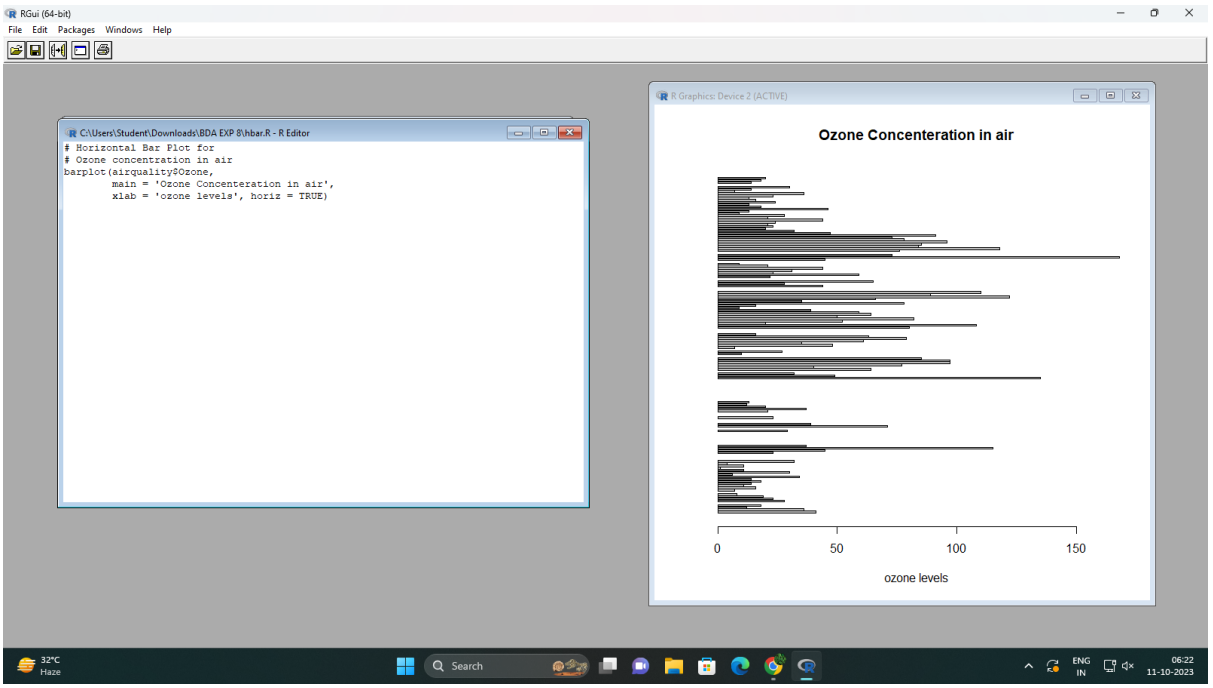
Ozone	Solar R.	Wind	Temp	Month	Day
41	190	7.4	67	5	1
36	118	8.0	72	5	2
12	149	12.6	74	5	3
18	313	11.5	62	5	4
NA	NA	14.3	56	5	5
28	NA	14.9	66	5	6

1.Bar Plot

There are two types of bar plots- horizontal and vertical which represent data points as horizontal or vertical bars of certain lengths proportional to the value of the data item. They are generally used for continuous and categorical variable plotting. By setting the horiz parameter to true and false, we can get horizontal and vertical bar plots respectively.

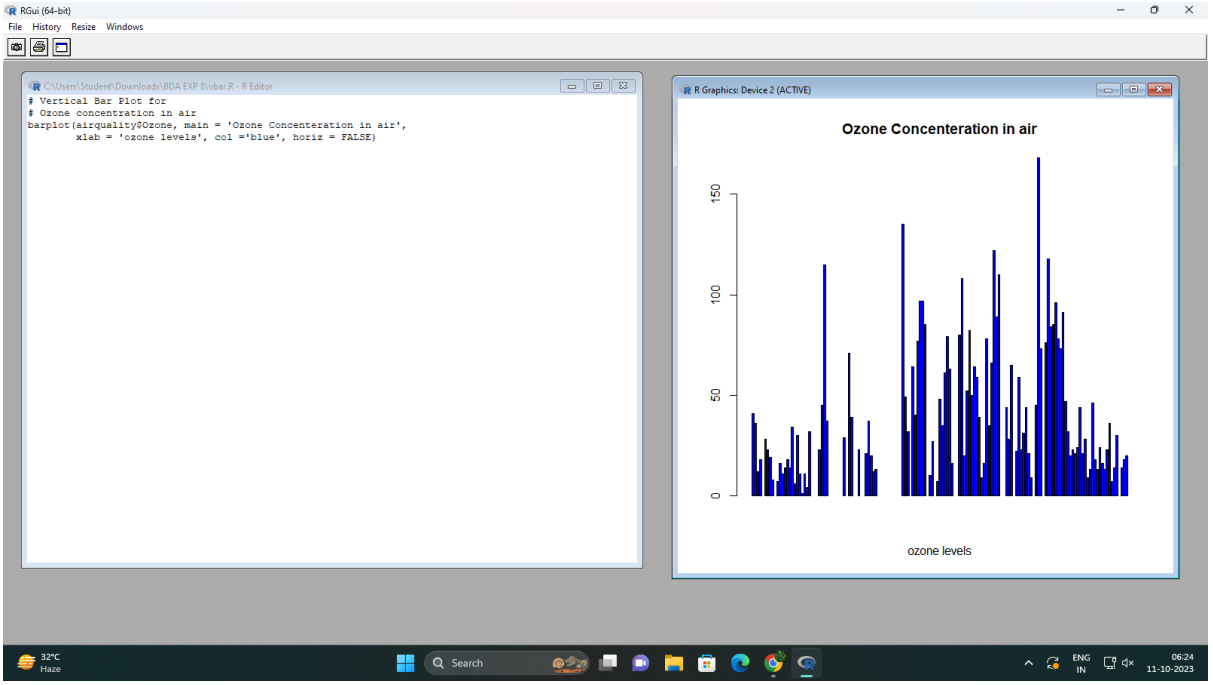
Example 1:

```
# Horizontal Bar Plot for
# Ozone concentration in air
barplot(airquality$Ozone,
        main = 'Ozone Concenteration in air',
        xlab = 'ozone levels', horiz = TRUE)
```



Example 2:

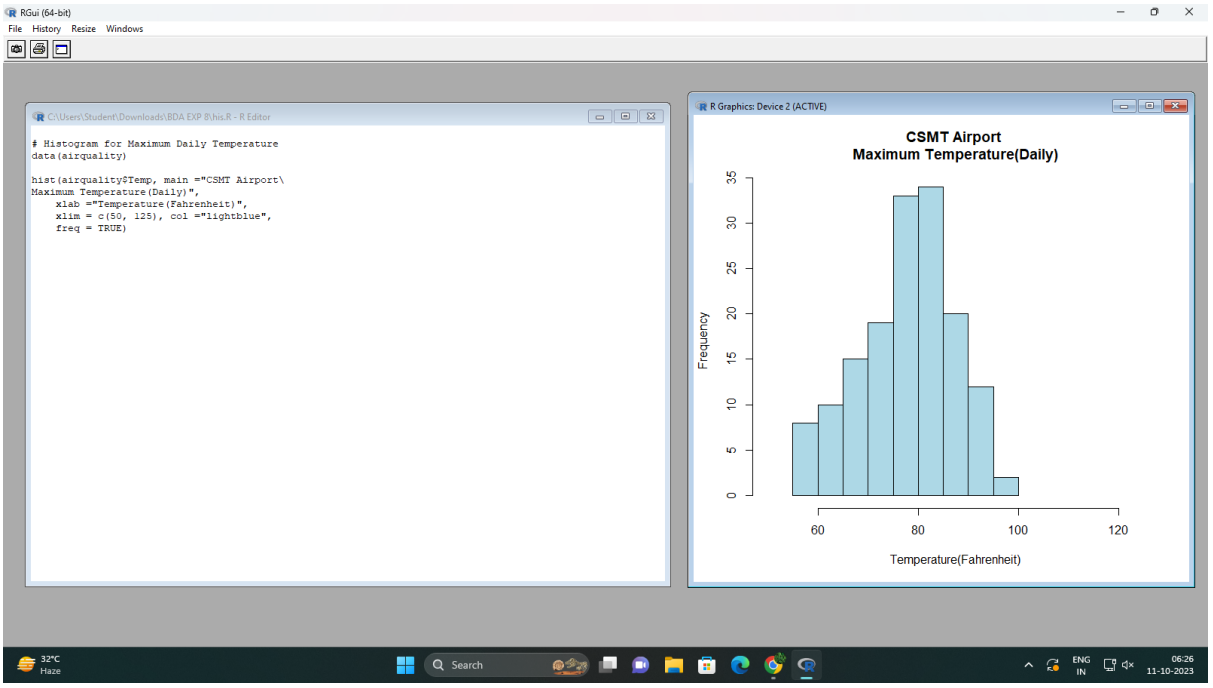
```
# Vertical Bar Plot for
# Ozone concentration in air
barplot(airquality$Ozone, main = 'Ozone Concenteration in air',
        xlab = 'ozone levels', col ='blue', horiz = FALSE)
```



2. Histogram

A histogram is like a bar chart as it uses bars of varying height to represent data distribution.
However, in a histogram values are grouped into consecutive intervals called bins. In a Histogram, continuous values are grouped and displayed in these bins whose size can be varied.

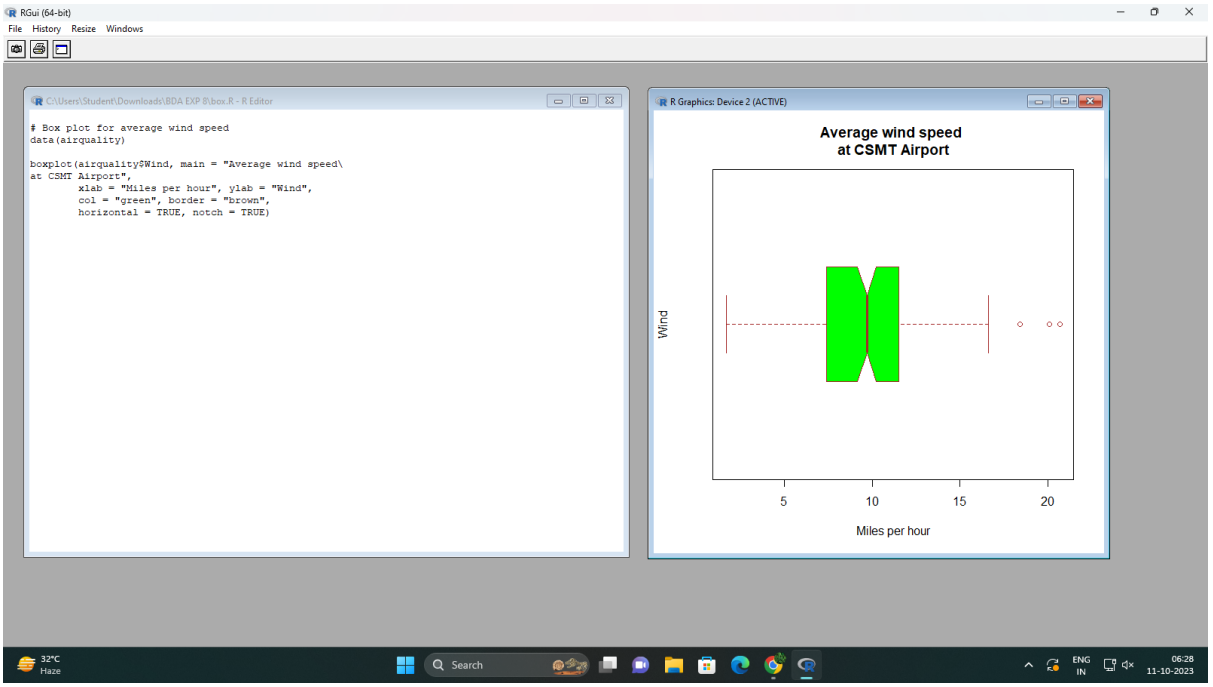
```
# Histogram for Maximum Daily Temperature
data(airquality)

hist(airquality$Temp, main ="CSMT Airport\
Maximum Temperature(Daily)",
    xlab ="Temperature(Fahrenheit)",
    xlim = c(50, 125), col ="lightblue",
    freq = TRUE)
```



3. Box Plot

The statistical summary of the given data is presented graphically using a boxplot. A box plot depicts information like the minimum and maximum data point, the median value, first and third quartile, and interquartile range.

```
# Box plot for average wind speed
data(airquality)
boxplot(airquality$Wind, main = "Average wind speed\
at CSMT Airport",
        xlab = "Miles per hour", ylab = "Wind",
        col = "green", border = "brown",
        horizontal = TRUE, notch = TRUE)
```

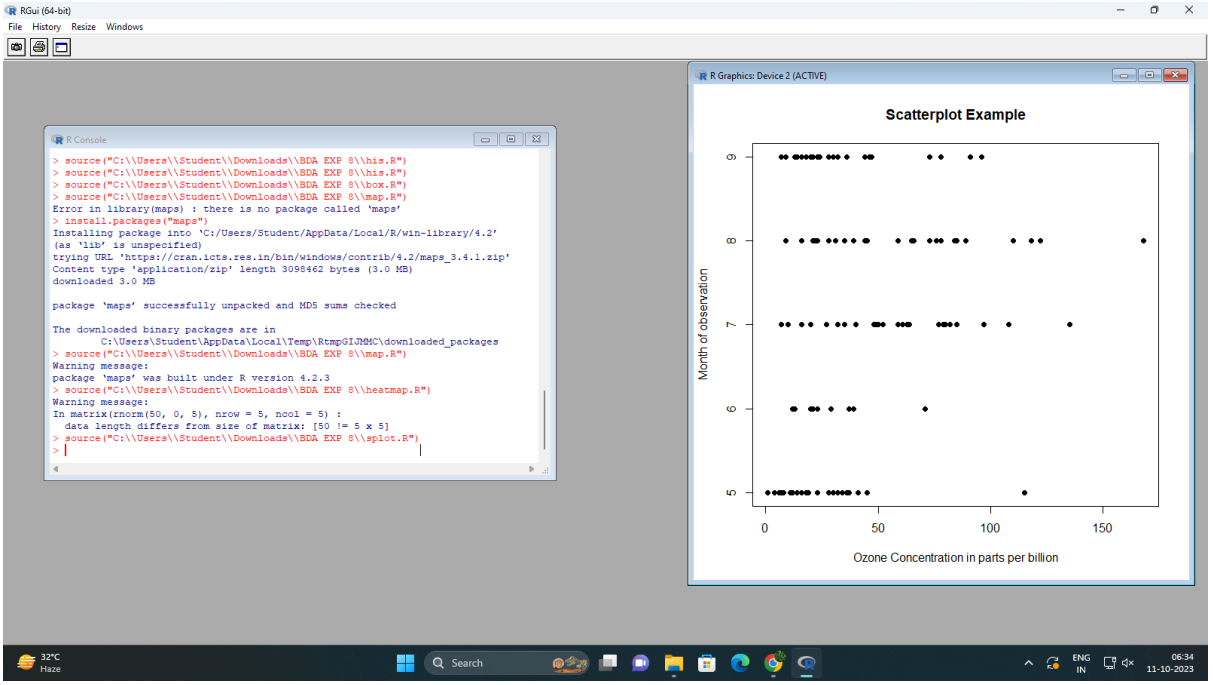


4.Scatter Plot

A scatter plot is composed of many points on a Cartesian plane. Each point denotes the value
taken by two parameters and helps us easily identify the relationship between them.

Example:.

```
# Scatter plot for Ozone Concentration per month
data(airquality)
plot(airquality$Ozone, airquality$Month,
 main ="Scatterplot Example",
 xlab ="Ozone Concentration in parts per billion",
 ylab =" Month of observation ", pch = 19)
```

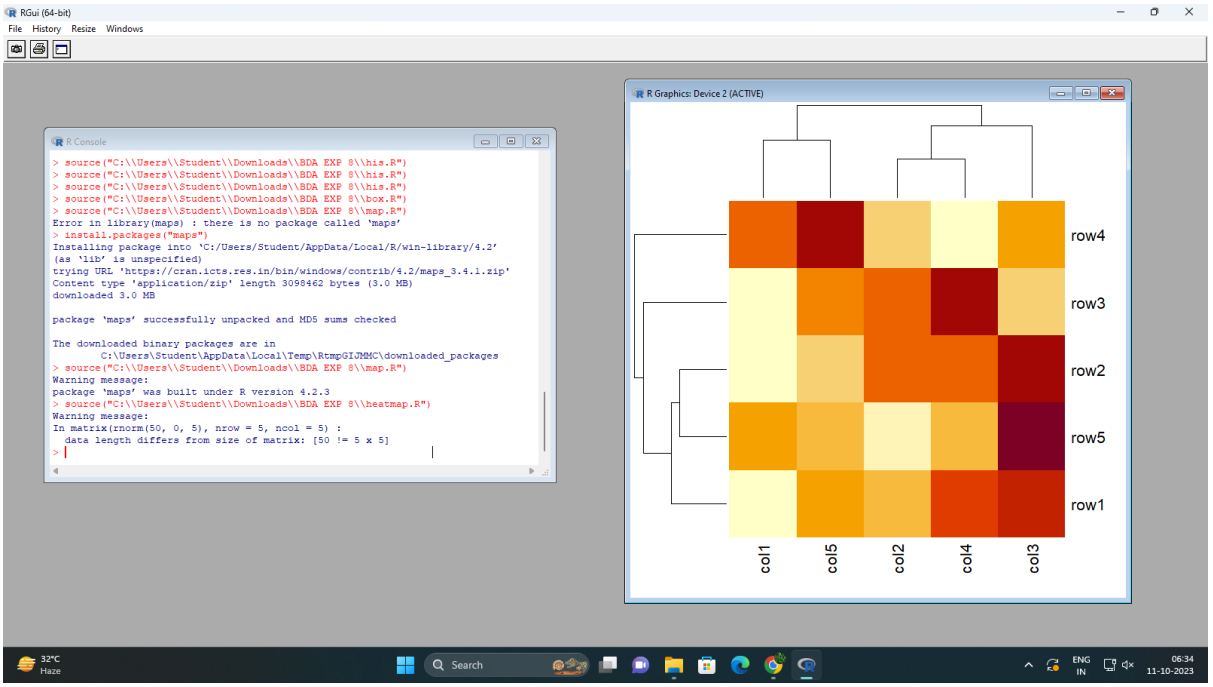


5. Heat Map

Heatmap is defined as a graphical representation of data using colours to visualise the value of the matrix. heatmap() function is used to plot heatmap.

Syntax: heatmap(data)
Parameters: data: It represent matrix data, such as values of rows and columns
Return: This function draws a heatmap.

```
# Set seed for reproducibility
# set.seed(110)

# Create example data
data <- matrix(rnorm(50, 0, 5), nrow = 5, ncol = 5)

# Column names
colnames(data) <- paste0("col", 1:5)
rownames(data) <- paste0("row", 1:5)

# Draw a heatmap
heatmap(data)
```

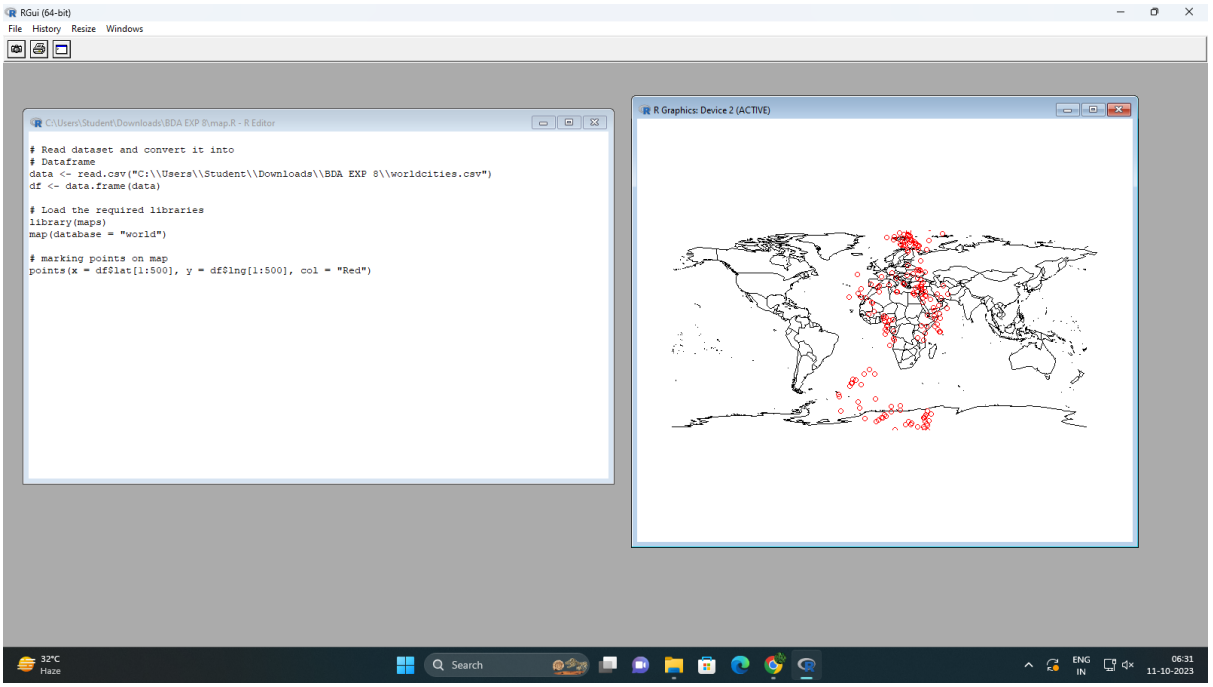


6. Map visualisation in R

Here we are using maps package to visualise and display geographical maps using an R programming language.

install.packages("maps")
Link of the dataset: worldcities.csv

```
# Read dataset and convert it into
# Dataframe
data <- read.csv("worldcities.csv")
df <- data.frame(data)

# Load the required libraries
library(maps)
map(database = "world")

# marking points on map
points(x = df$lat[1:500], y = df$lng[1:500], col = "Red")
```



7. 3D Graphs in R

Here we will use the preps() function, This function is used to create 3D surfaces in perspective view. This function will draw perspective plots of a surface over the x–y plane.

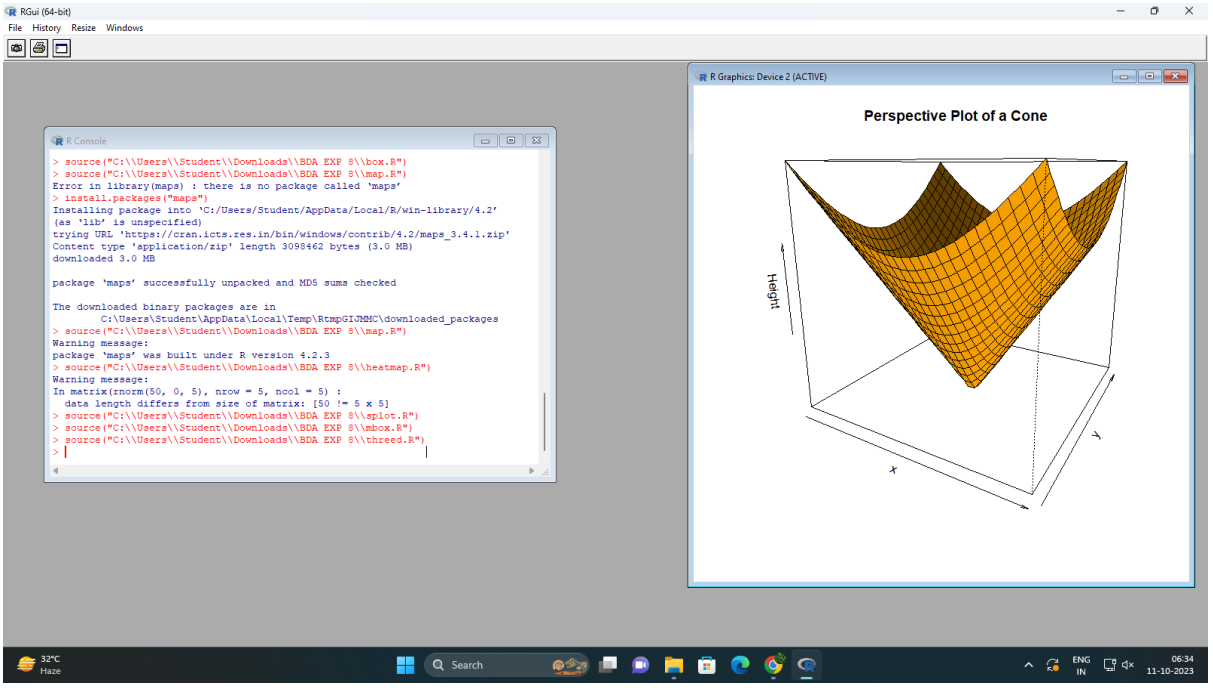
Syntax: persp(x, y, z)

Parameter: This function accepts different parameters i.e. x, y and z where x and y are vectors defining the location along x- and y-axis. z-axis will be the height of the surface in the matrix z.

Return Value: persp() returns the viewing transformation matrix for projecting 3D coordinates (x, y, z) into the 2D plane using homogeneous 4D coordinates (x, y, z, t).

```
# Adding Titles and Labeling Axes to Plot
cone <- function(x, y){
sqrt(x ^ 2 + y ^ 2)
}

# prepare variables.
x <- y <- seq(-1, 1, length = 30)
z <- outer(x, y, cone)

# plot the 3D surface
# Adding Titles and Labeling Axes to Plot
```

```
persp(x, y, z,
main="Perspective Plot of a Cone",
zlab = "Height",
theta = 30, phi = 15,
col = "orange", shade = 0.4)
```



CONCLUSION :

The experiment of Data Visualization in R is a crucial aspect of data analysis and communication. R's versatile tools and packages enable the creation of powerful and tailored visualisations, enhancing data understanding and decision-making. As data complexity grows, data visualisation in R remains a vital tool for transforming data into actionable insights.The flexibility and customization options within R's data visualisation capabilities make it the preferred choice for data scientists, researchers, and analysts. It allows them to tailor visualisations to specific project requirements and data characteristics, ensuring that the right information is conveyed to the right people.