Experiment No.8
Data Visualization using Hive/PIG/R/Tableau/.
Date of Performance: 25/09/2023
Date of Submission: 11/10/2023

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

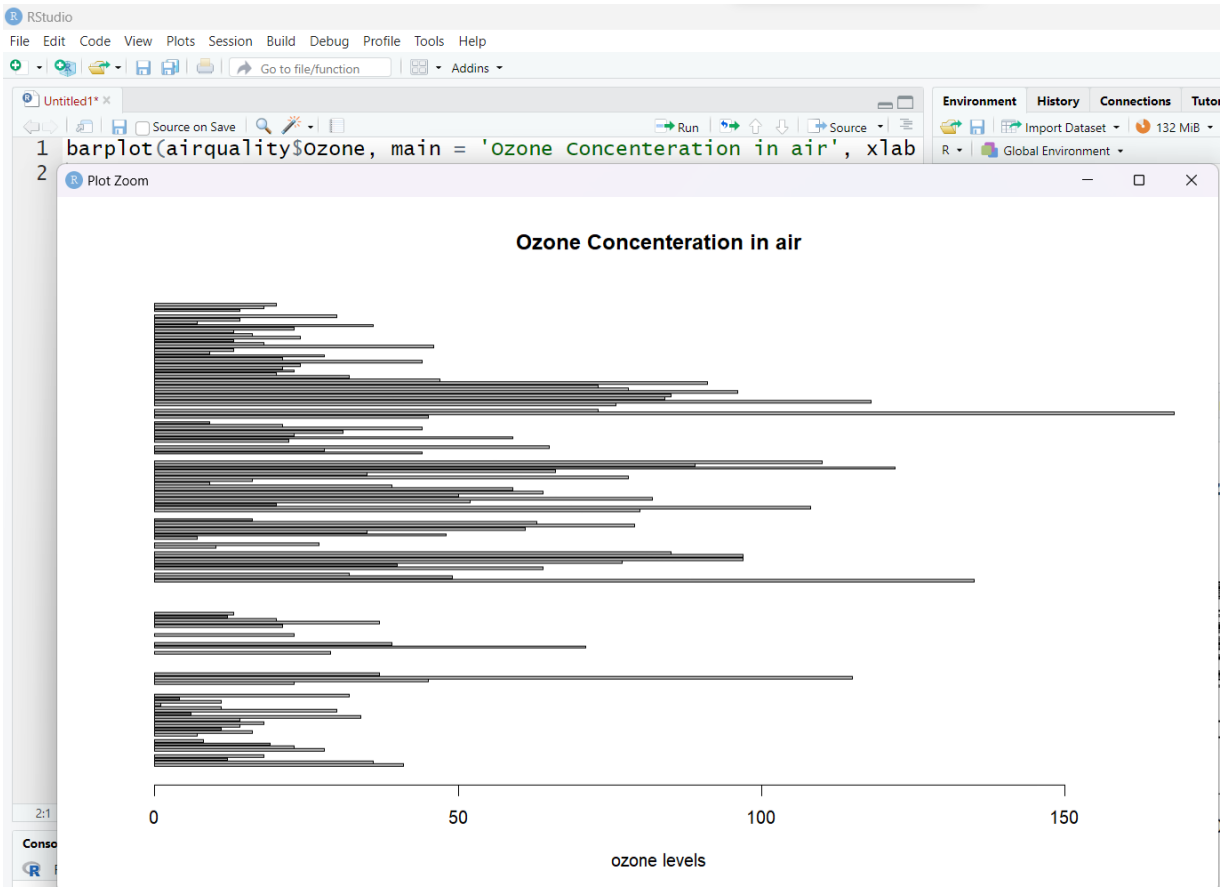
1.Bar Plot

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.

Code:

```
barplot(airquality$Ozone, main = 'Ozone
    Concenteration in air', xlab = 'ozone
    levels', horiz = TRUE)
```

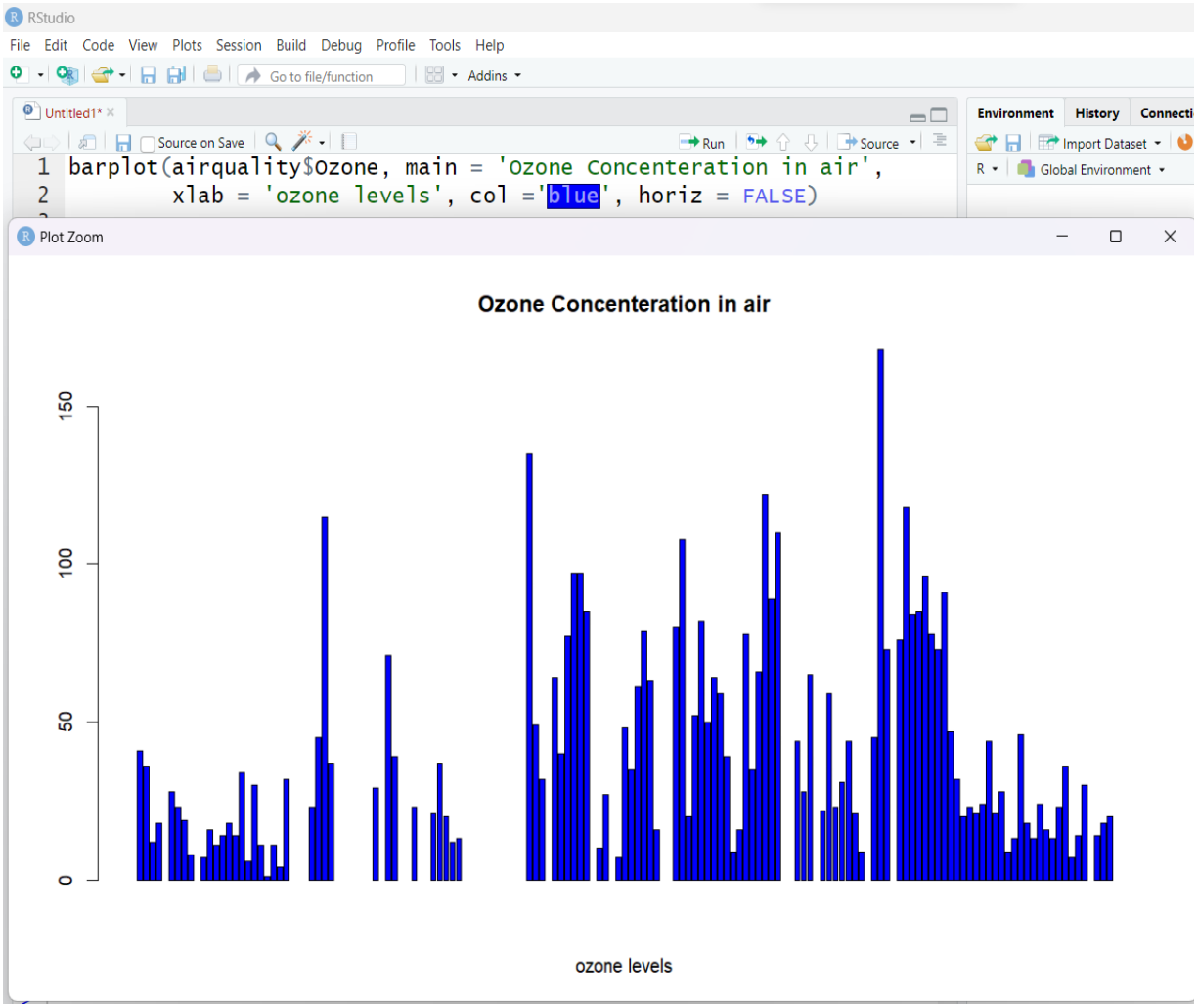
Output:



Code:

```
barplot(airquality$Ozone, main = 'Ozone Concenteration in air', xlab =
'ozone levels', col ='blue', horiz = FALSE)
```

Output:



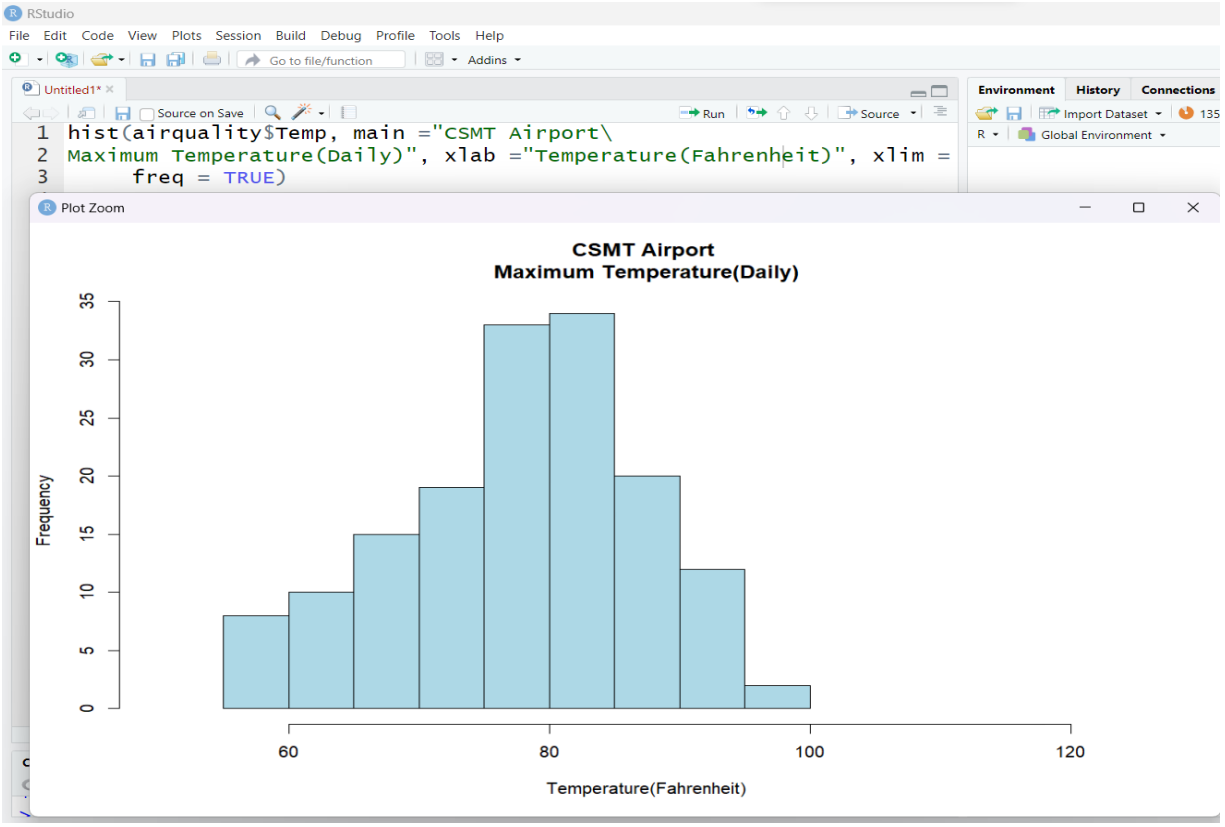
2. Histogram

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.

Code:

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

Output:



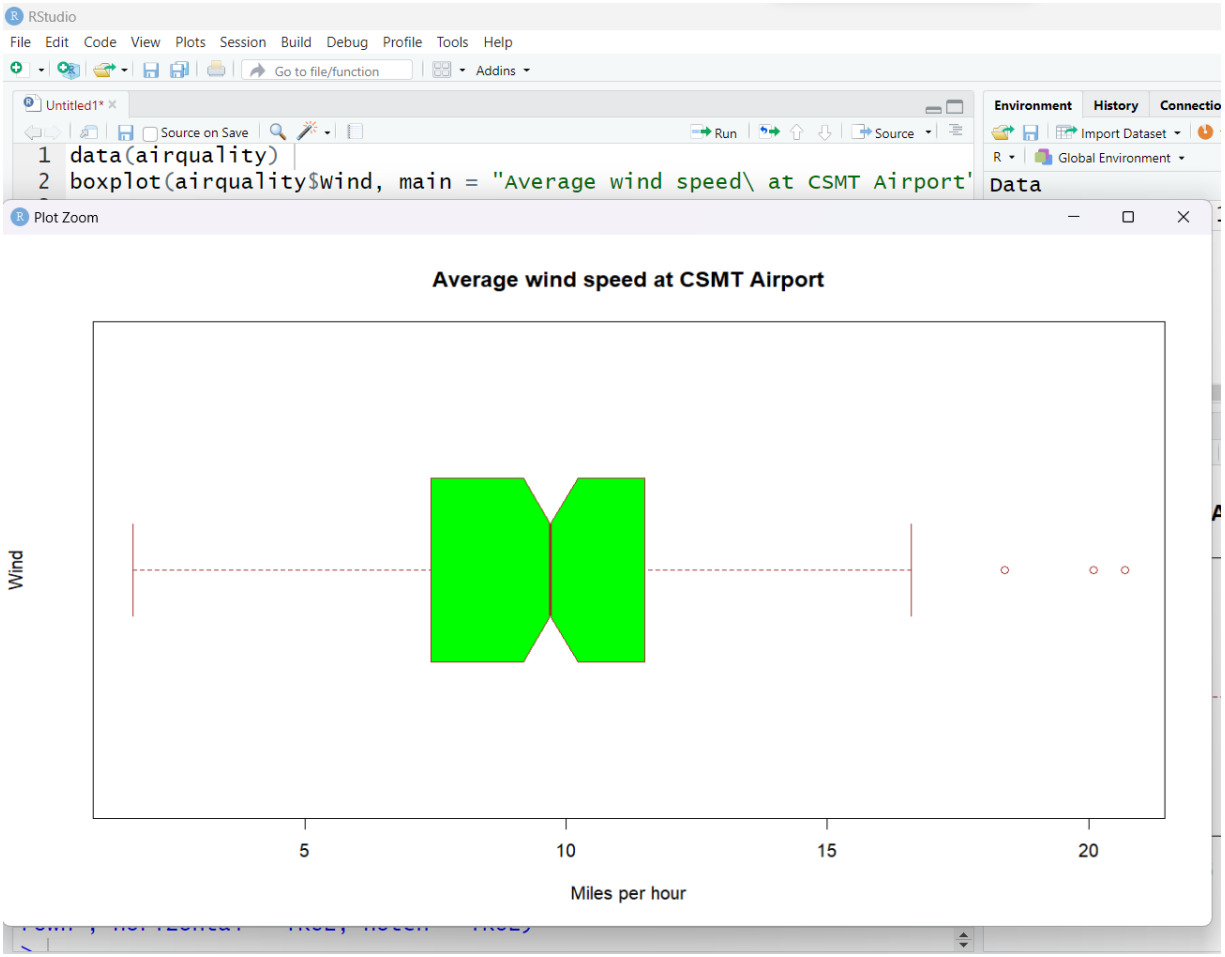
3. Box Plot

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.

Code:

```
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)
```

Output:



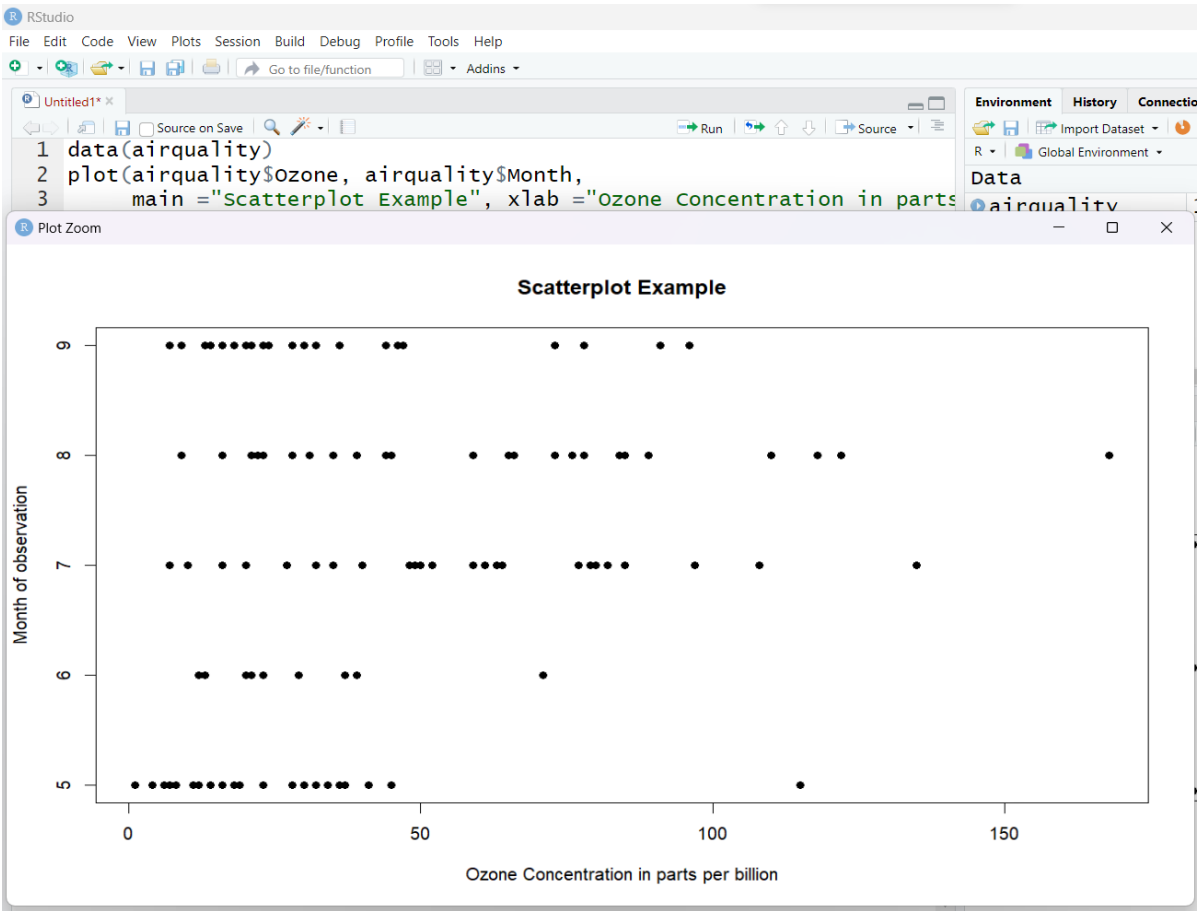
4. Scatter Plot

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.

Code:

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

Output:



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.

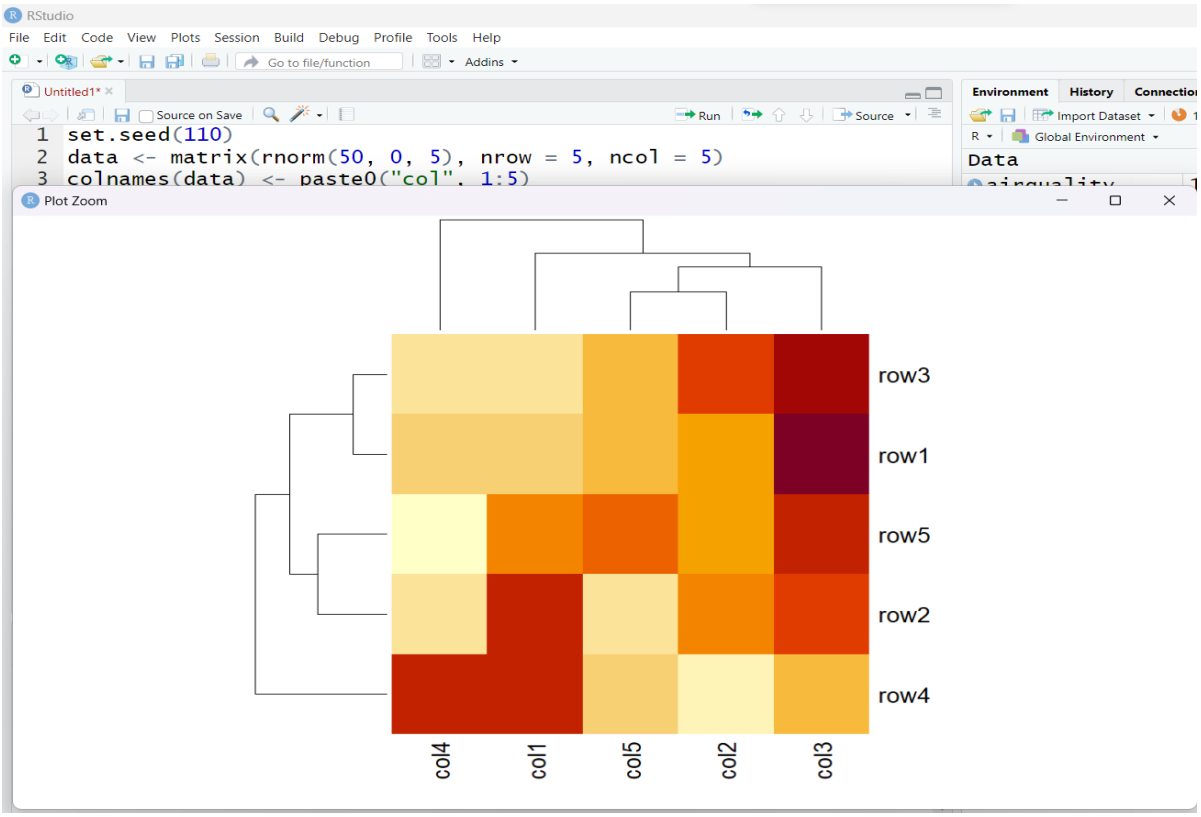
Syntax: heatmap(data)

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

Code:

```
set.seed(110)
data <- matrix(rnorm(50, 0, 5), nrow = 5, ncol = 5)
colnames(data) <- paste0("col", 1:5)
rownames(data) <- paste0("row", 1:5)
heatmap(data)
```

Output:



6. Map visualization in R

Code:

install.packages("maps")

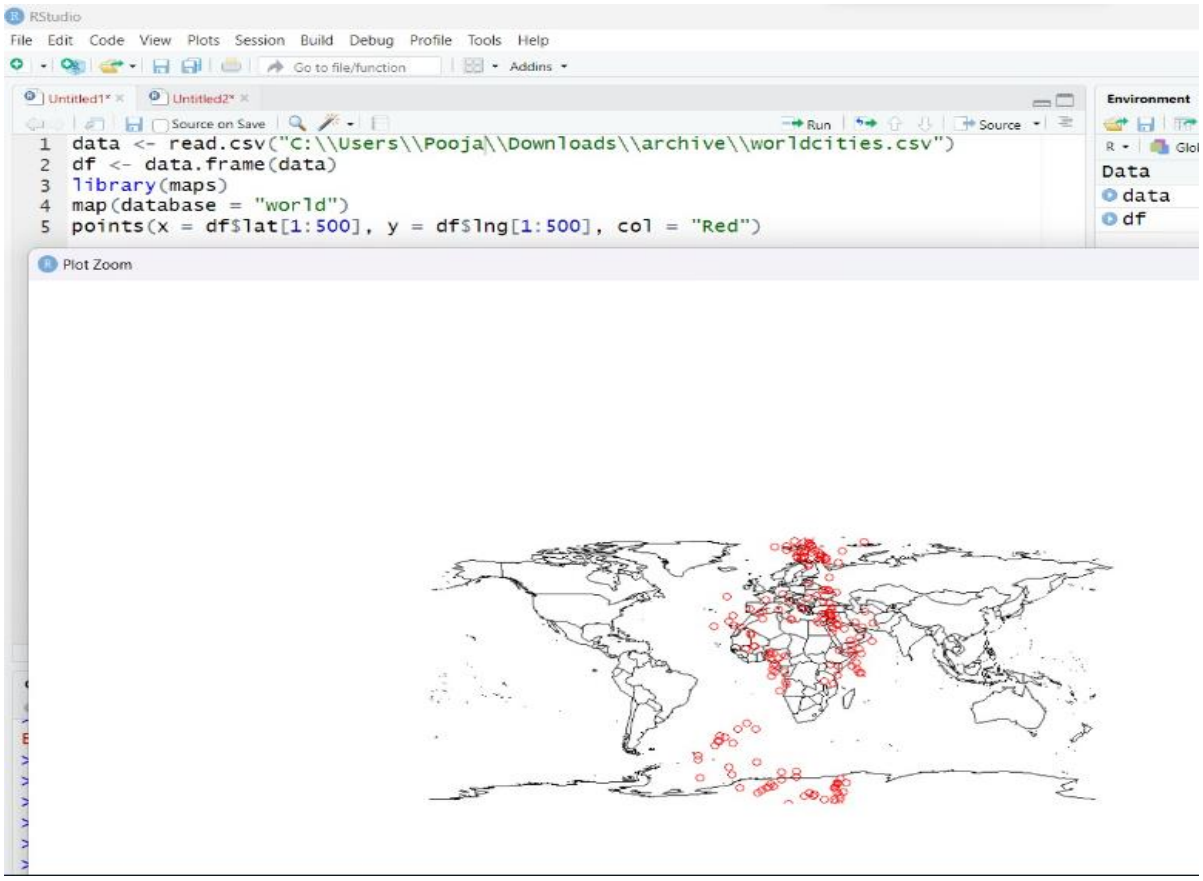
Link of the dataset: worldcities.csv

data <- read.csv("worldcities.csv") df <- data.frame(data)

map(database = "world")

points(x = df$lat[1:500], y = df$lng[1:500], col = "Red")

Output:



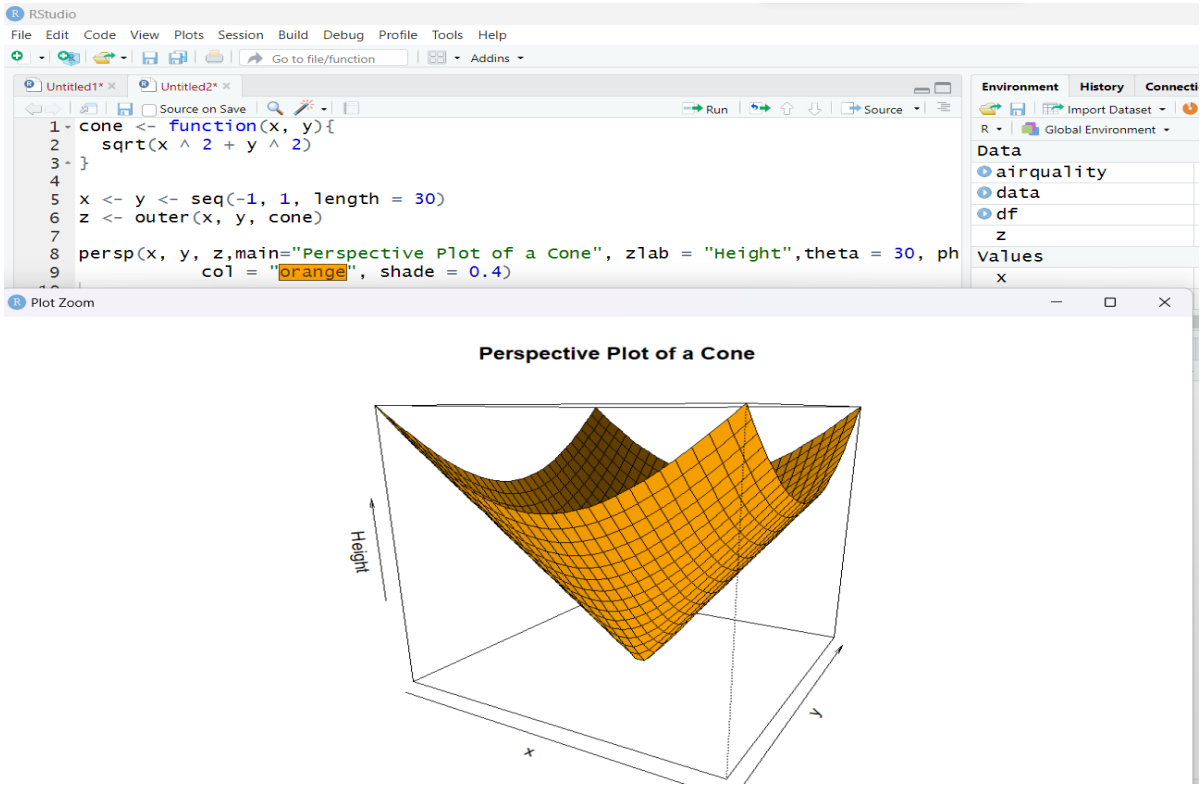
7. 3D Graphs in R

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)

Code:

```
cone <- function(x, y){ sqrt(x ^ 2 + y ^ 2) }
x <- y <- seq(-1, 1, length = 30) z <- outer(x, y, cone)
persp(x, y, z,
main="Perspective Plot of a Cone",
zlab = "Height", theta = 30, phi = 15,
col = "orange", shade = 0.4)
```

Output:



Conclusion:

One of the most important parts of data analysis is data visualization, and R is an excellent programming language and environment for making many different kinds of data visualizations. R has a number of tools and packages made especially for data visualization. Users may develop distinctive and captivating visual representations thanks to R's rich toolset and packages, which enhance data comprehension and help with well-informed decision-making. In an environment where data complexity will only rise, data visualization in R is a crucial tool for transforming data into insights that can be implemented.

In this project, we successfully use R programming to visualize data..