Data Analysis using R

By

Dr.Parkavi.A

Unit 2

Lec 1

Types of Data Visualizations in R

- Data visualization in R can be accomplished using various functions and libraries such as ggplot2 for more advanced and customizable plots. Below are examples of different types of plots, including scatter plots, bar plots, histograms, box plots, scatter plot matrices, 3D scatter plots, and heat maps.
- graphs, charts, maps
- decisions

Airquality data set

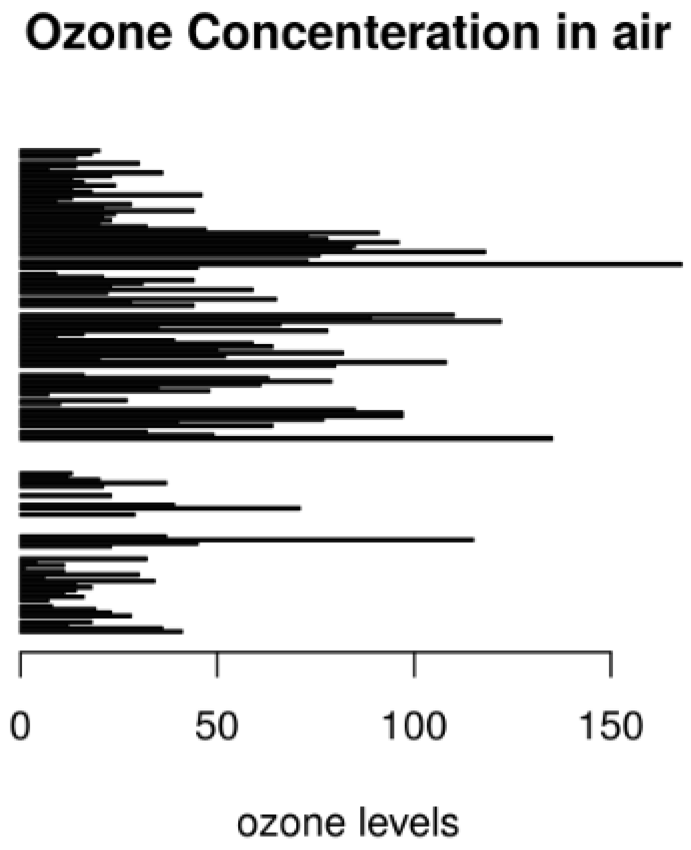
Ozone	Solar R.	Wind	Temp	Month	Day
41	190	7.4	67	5	1
36	118	8.0	72	5	2
12	149	12.6	74	5	3
18	313	11.5	62	5	4
NA	NA	14.3	56	5	5
28	NA	14.9	66	5	6

Bar Plot

- horizontal and vertical

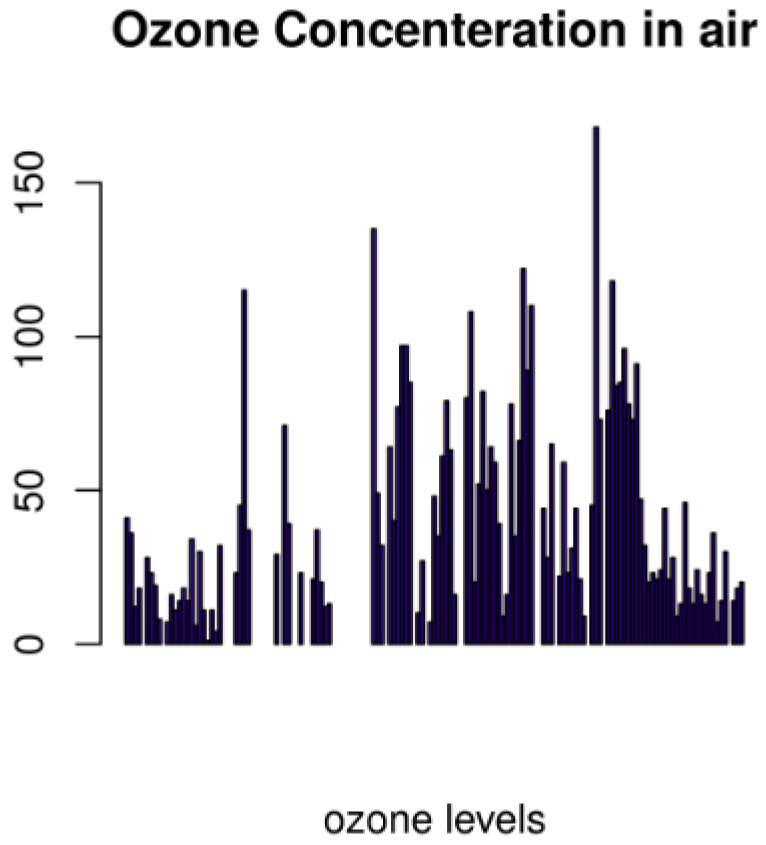
- To perform a comparative study between the various data categories in the data set.
- To analyze the change of a variable over time in months or years.

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



Eg

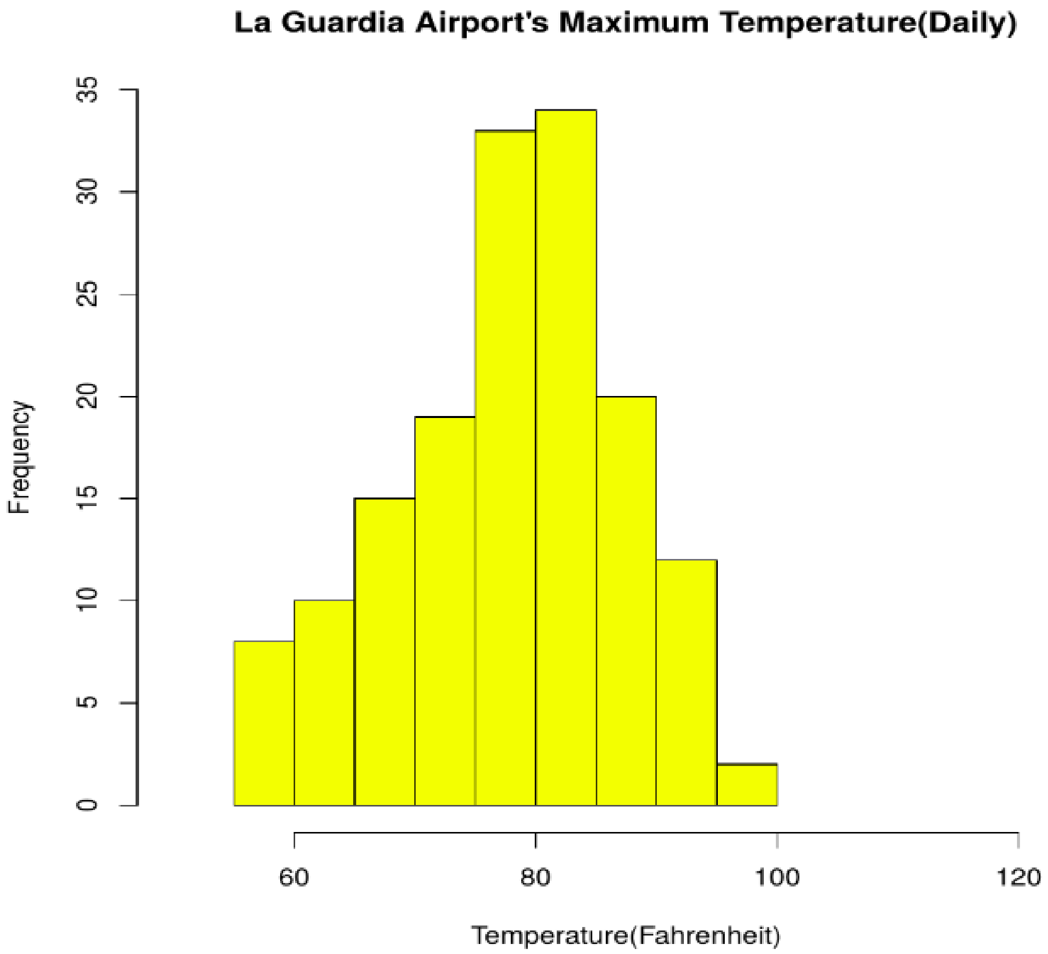
```
# Vertical Bar Plot for
# Ozone concentration in air
barplot(airquality$Ozone, main = 'Ozone Concenteration in air', xlab = 'ozone levels', col ='blue', horiz = FALSE)
```



Histogram

- like a bar chart as it uses bars of varying height to represent data distribution.
- values are grouped into consecutive intervals called bins
- To verify an equal and symmetric distribution of the data.
- To identify deviations from expected values.

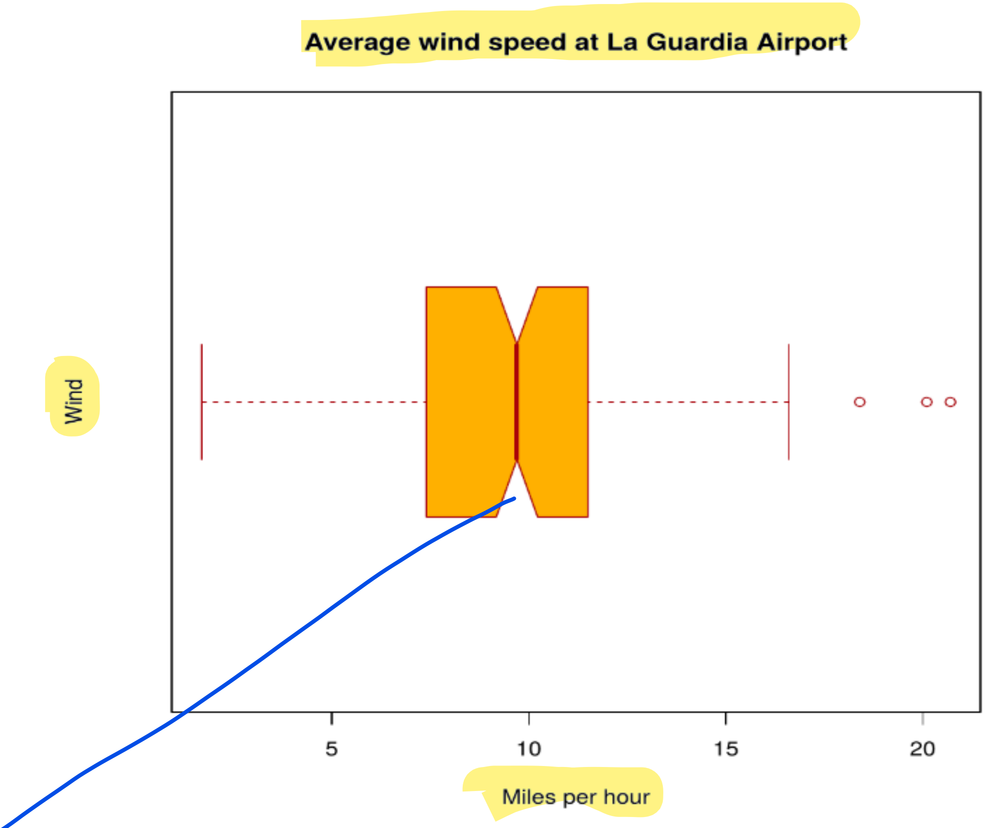
```
# Histogram for Maximum Daily Temperature
data(airquality)
hist(airquality$Temp, main ="La Guardia Airport's\
Maximum Temperature(Daily)",
            xlab ="Temperature(Fahrenheit)",
            xlim = c(50, 125), col ="yellow",
            freq = TRUE)
```



Box Plot

- data is presented graphically using a boxplot
- minimum and maximum data point, the median value, first and third quartile, and interquartile range.

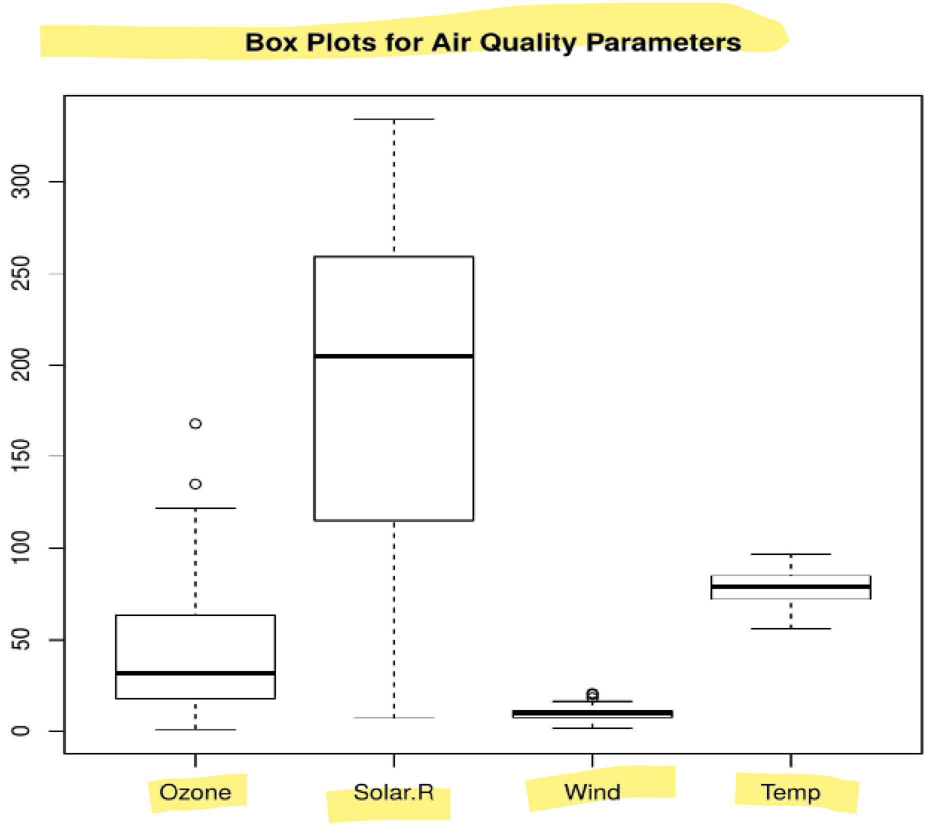
```
# Box plot for average wind speed
data(airquality)
boxplot(airquality$Wind, main = "Average
wind speed\
at La Guardia Airport",
                    xlab = "Miles per hour", ylab =
"Wind",
                    col = "orange", border =
"brown",
                    horizontal = TRUE, notch = TRUE)
```



eg

- To give a comprehensive statistical description of the data through a visual cue.
- To identify the outlier points that do not lie in the inter-quartile range of data.

```
# Multiple Box plots, each representing
# an Air Quality Parameter
boxplot(airquality[, 0:4],
                    main ='Box Plots for Air
Quality Parameters')
```



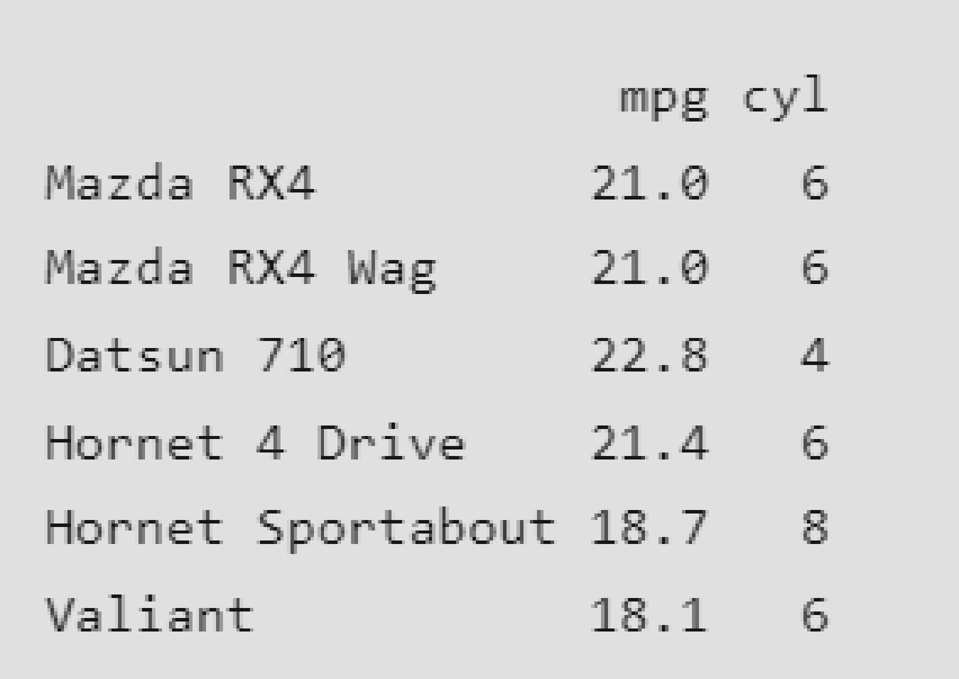
Outlier's: extremely high or extremely low data point relative to the nearest data point and
the rest of the neighboring co-existing values in a data graph or dataset

Boxplots

- used to display information in the form of distribution by drawing boxplots
- This distribution of data is based on five sets (minimum, first quartile, median, third quartile, and maximum).
- ***x:*** *This parameter sets as a vector or a formula.*
- ***data:*** *This parameter sets the data frame.*
- ***notch:*** *This parameter is the label for horizontal axis.*
- ***varwidth**: This parameter is a logical value. Set as true to draw width of the box proportionate to the sample size.*
- ***main:*** *This parameter is the title of the chart.*
- ***names:*** *This parameter are the group labels that will be showed under each boxplot.*

eg

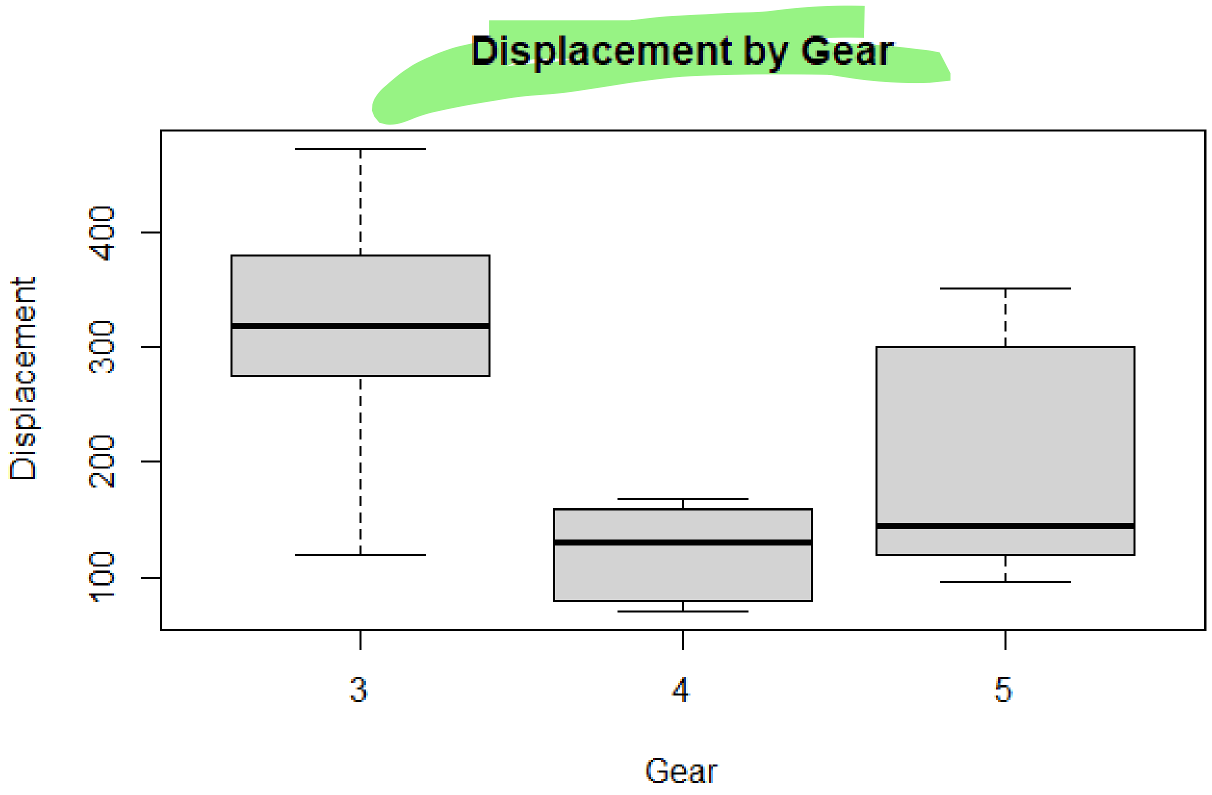
```
input <- mtcars[, c('mpg', 'cyl')]
print(head(input))
```



```
                   mpg cyl
Mazda RX4         21.0   6
Mazda RX4 Wag     21.0   6
Datsun 710        22.8   4
Hornet 4 Drive    21.4   6
Hornet Sportabout 18.7   8
Valiant           18.1   6
```

eg

```
# Load the dataset
data(mtcars)

# Create the box plot
boxplot(disp ~ gear, data = mtcars,
                   main = 
"Displacement by Gear",
                   xlab = "Gear",
                   ylab = 
"Displacement")
```

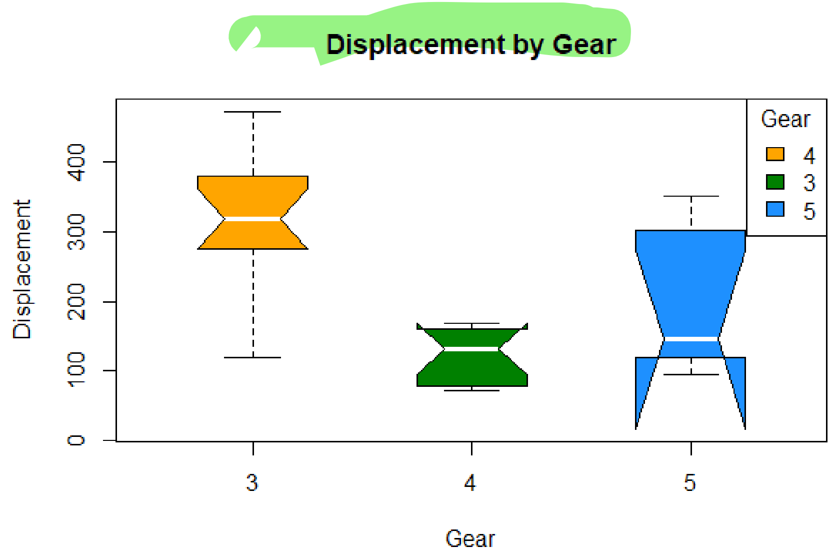


Boxplot using notch

- Find out how the medians of different data groups match with each other.

eg

```
# Load the dataset
data(mtcars)

# Set up plot colors
my_colors <- c("#FFA500", "#008000", "#1E90FF", "#FF1493")

# Create the box plot with customized aesthetics
boxplot(disp ~ gear, data = mtcars,
                    main = "Displacement by Gear", xlab = "Gear", ylab = "Displacement",
                    col = my_colors, border = "black", notch = TRUE, notchwidth = 0.5,
                    medcol = "white", whiskcol = "black", boxwex = 0.5, outpch = 19,
                    outcol = "black")

# Add a legend
legend("topright", legend = unique(mtcars$gear),
          fill = my_colors, border = "black", title = "Gear")
```

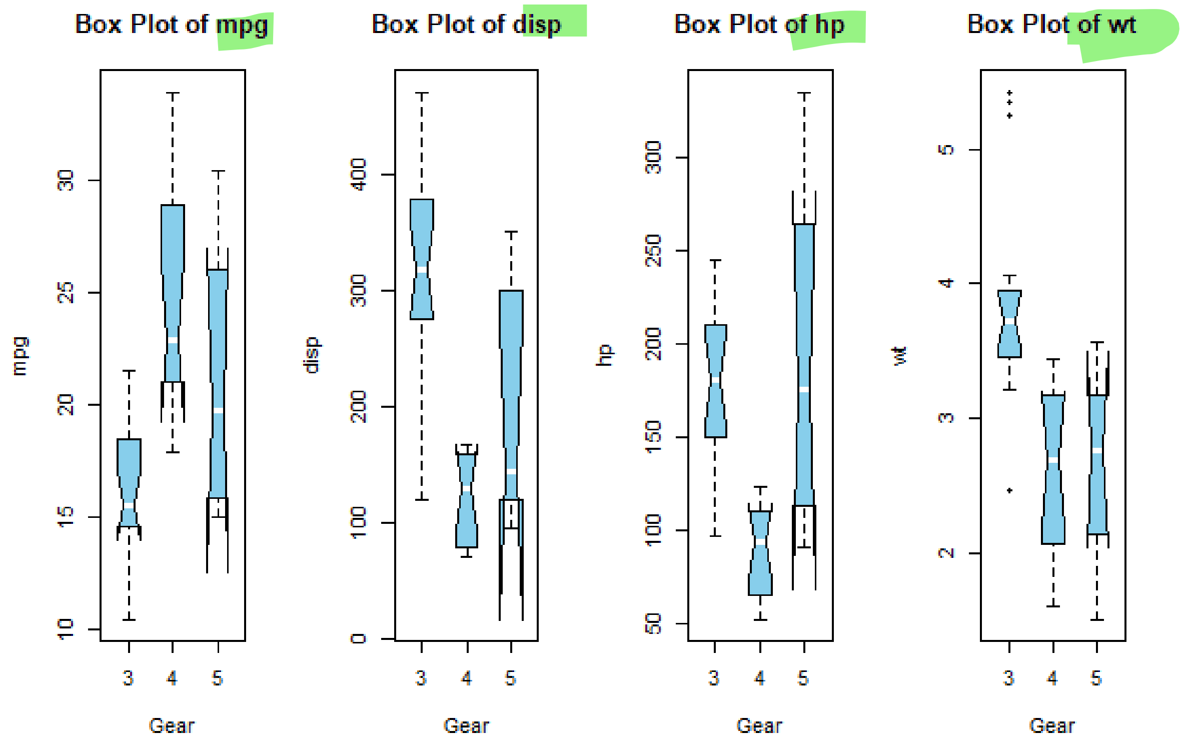


Contd..

- **col:** Uses a vector of colours (my_colors) to change the fill colour of the boxes.
  borders: Sets the box borders’ colour to black.
  notch: To illustrate confidence intervals, a notch is added to the boxes.
  notchwidth: Manages the notches’ width.
  medcol: Makes the median line’s colour white.
  whiskcol: Sets the whiskers’ colour to black with the whiskcol command.
  boxwex: Modifies the boxes’ width.
  outpch: Sets the outliers’ shapes to solid circles.
  outcol: Changes the outliers’ colour to black.

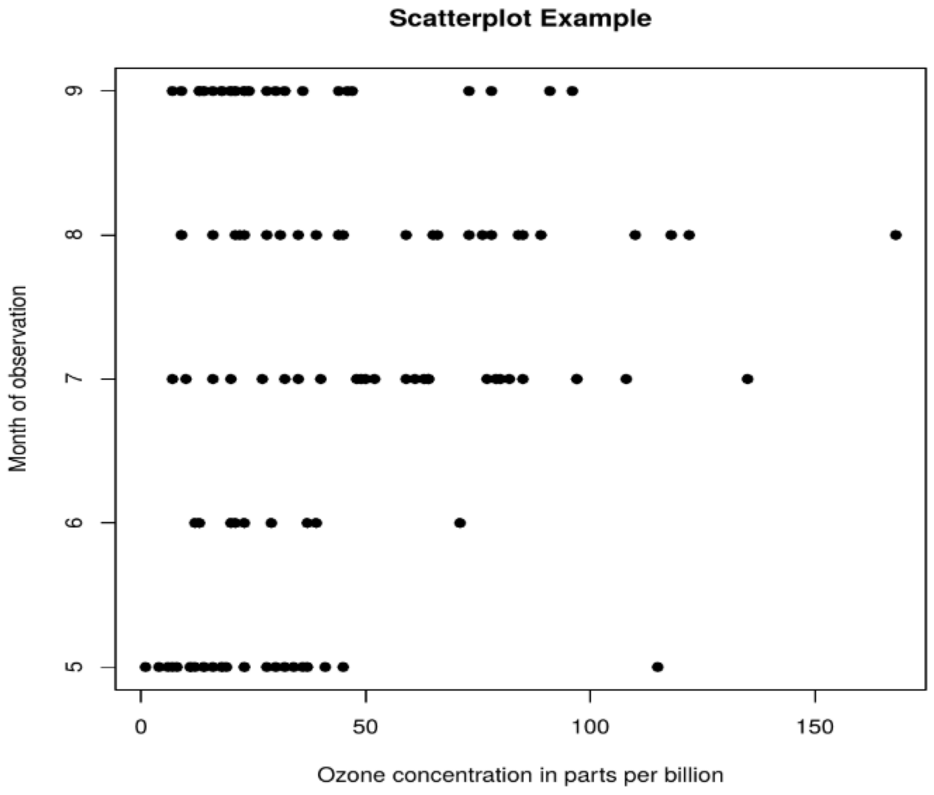
Multiple Boxplot

```
# Load the dataset
data(mtcars)
# Define the variables for the box plots
variables <- c("mpg", "disp", "hp", "wt")
# Set up the plotting layout
par(mfrow = c(1, length(variables)))
# Create the box plots
for (var in variables) {
boxplot(get(var) ~ gear, data = mtcars,main = paste("Box
Plot of", var),xlab = "Gear",ylab = var,    col = "skyblue",
border = "black",     notch = TRUE,         notchwidth =
0.5,medcol = "white",
whiskcol = "black",boxwex = 0.5,          outpch = 19,
                outcol = "black")
}
# Reset the plotting layout
par(mfrow = c(1, 1))
```



Scatter Plot

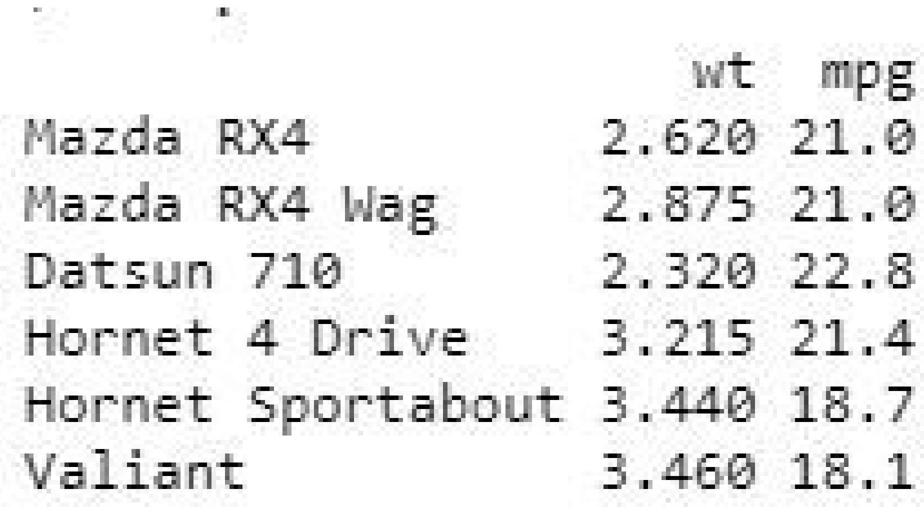
```
# Scatter plot for Ozone
Concentration per month
data(airquality)


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



Creating a Scatterplot Graph

- We are using the required parameters to plot the graph.
- In this ‘xlab’ describes the X-axis and ‘ylab’ describes the Y-axis.

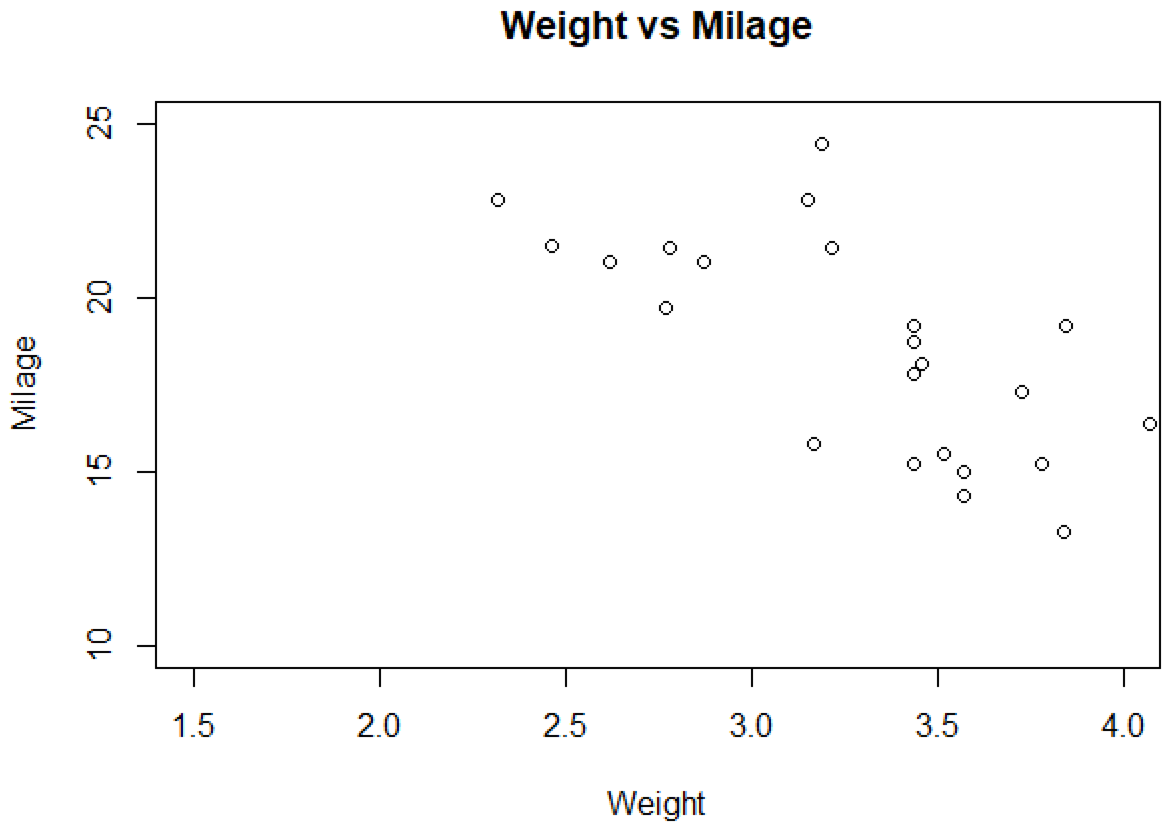
```
input <- mtcars[, c('wt', 'mpg')]
print(head(input))
```



```
                     wt  mpg
Mazda RX4         2.620 21.0
Mazda RX4 Wag     2.875 21.0
Datsun 710        2.320 22.8
Hornet 4 Drive    3.215 21.4
Hornet Sportabout 3.440 18.7
Valiant           3.460 18.1
```

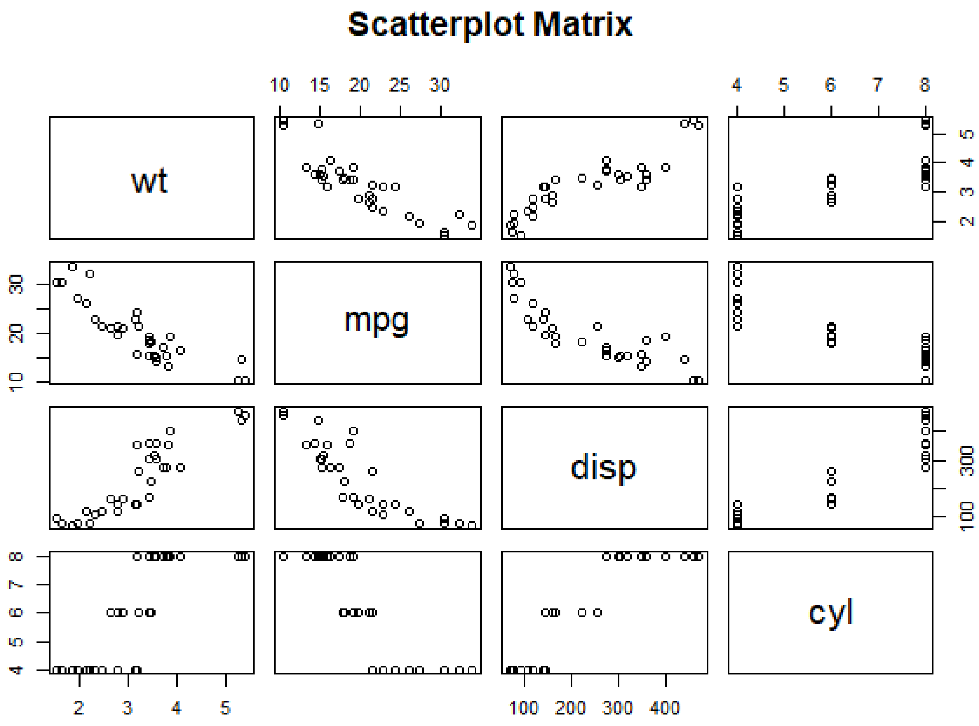
eg

```
# Get the input values.
input <- mtcars[, c('wt', 'mpg')]

# Plot the chart for cars with
# weight between 1.5 to 4 and
# mileage between 10 and 25.
plot(x = input$wt, y = input$mpg,
        xlab = "Weight",
        ylab = "Milage",
        xlim = c(1.5, 4),
        ylim = c(10, 25),
        main = "Weight vs Milage"
)
```



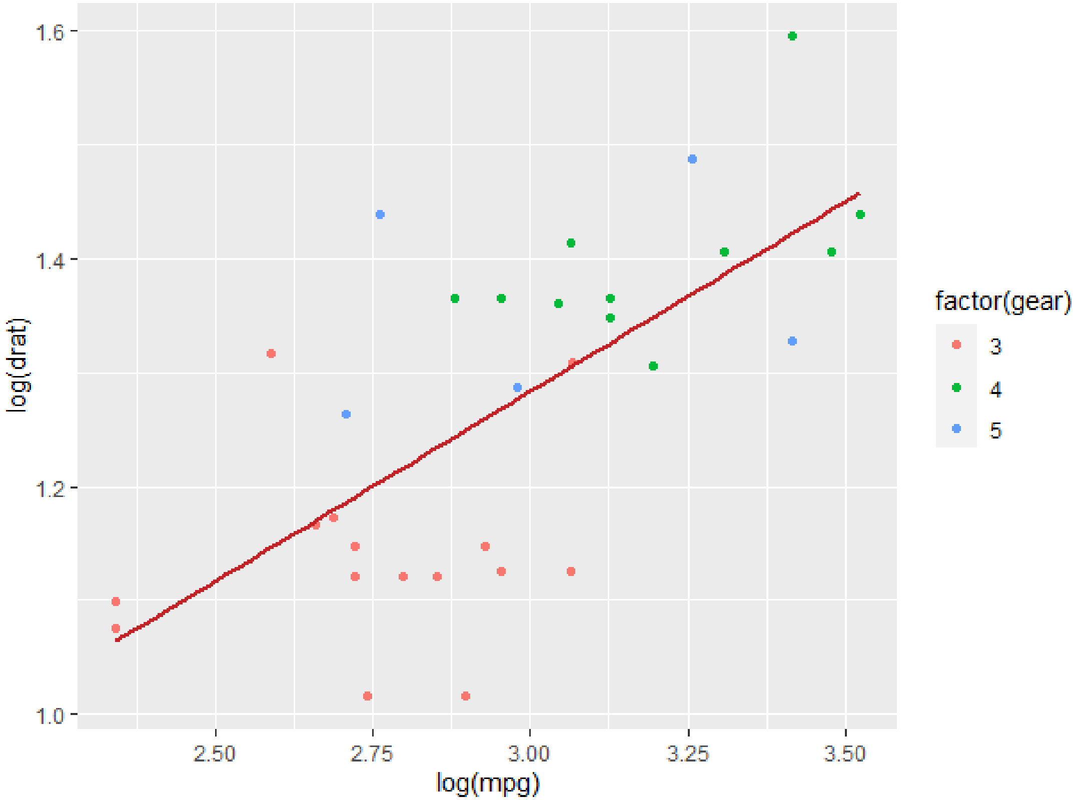
Scatterplot Matrices

```
# Plot the matrices between
# 4 variables giving 12 plots.

# One variable with 3 others
# and total 4 variables.
pairs(~wt + mpg + disp + cyl, data =
mtcars,
        main = "Scatterplot Matrix")
```



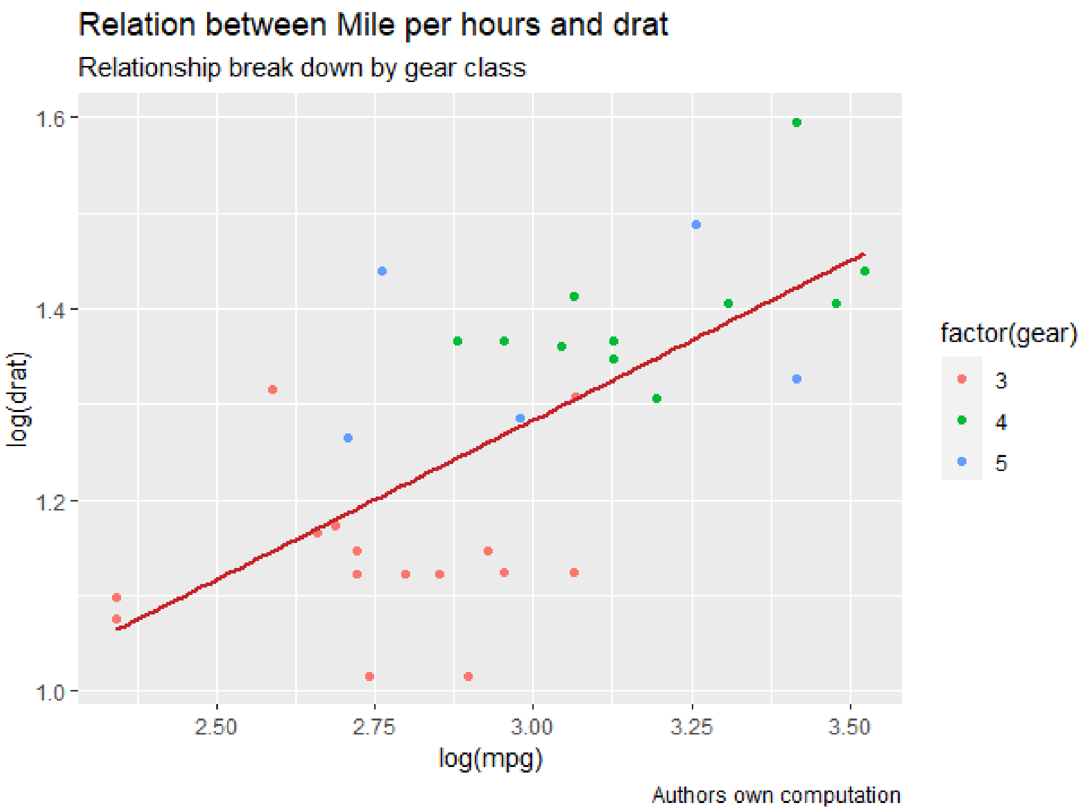
Scatterplot with fitted values

```
# Loading ggplot2 package
library(ggplot2)

# Creating scatterplot with fitted values.
# An additional function stat_smooth
# is used for linear regression.
ggplot(mtcars, aes(x = log(mpg), y = log(drat))) +
                  geom_point(aes(color = factor(gear))) +
                  stat_smooth(method = "lm",
                  col = "#C42126", se = FALSE, size = 1
)
```



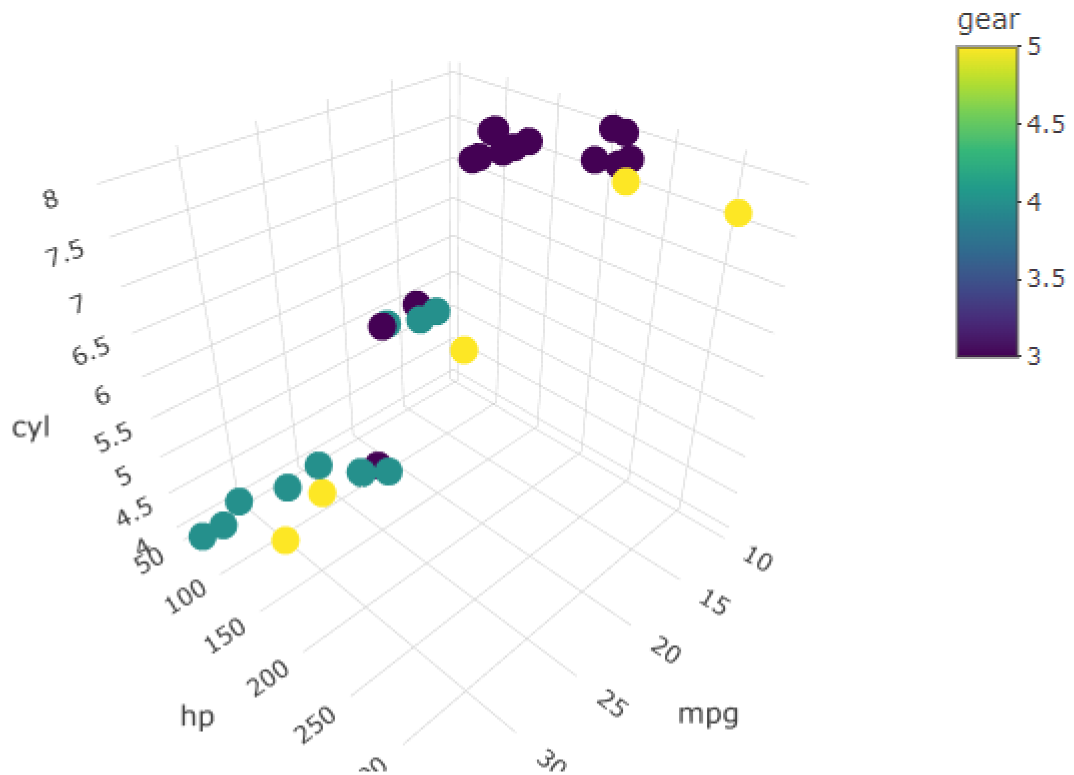
Adding title with dynamic name

```
# Loading ggplot2 package

library(ggplot2)

# Creating scatterplot with fitted values.

# An additional function stst_smooth

# is used for linear regression.

new_graph<-ggplot(mtcars, aes(x = log(mpg),y = log(drat)))
+geom_point(aes(color = factor(gear))) + stat_smooth(method = "lm",col =
"#C42126",se = FALSE, size = 1)

# in above example lm is used for linear regression

# and se stands for standard error.

# Adding title with dynamic name

new_graph + labs(title = "Relation between Mile per hours and drat",

                        subtitle = "Relationship break down by gear class",

                        caption = "Authors own computation"

)
```



3D Scatterplots

```
# 3D Scatterplot
library(plotly)
attach(mtcars)
plot_ly(data=mtcars,x=~mpg,y=
~hp,z=~cyl,color=~gear)
```



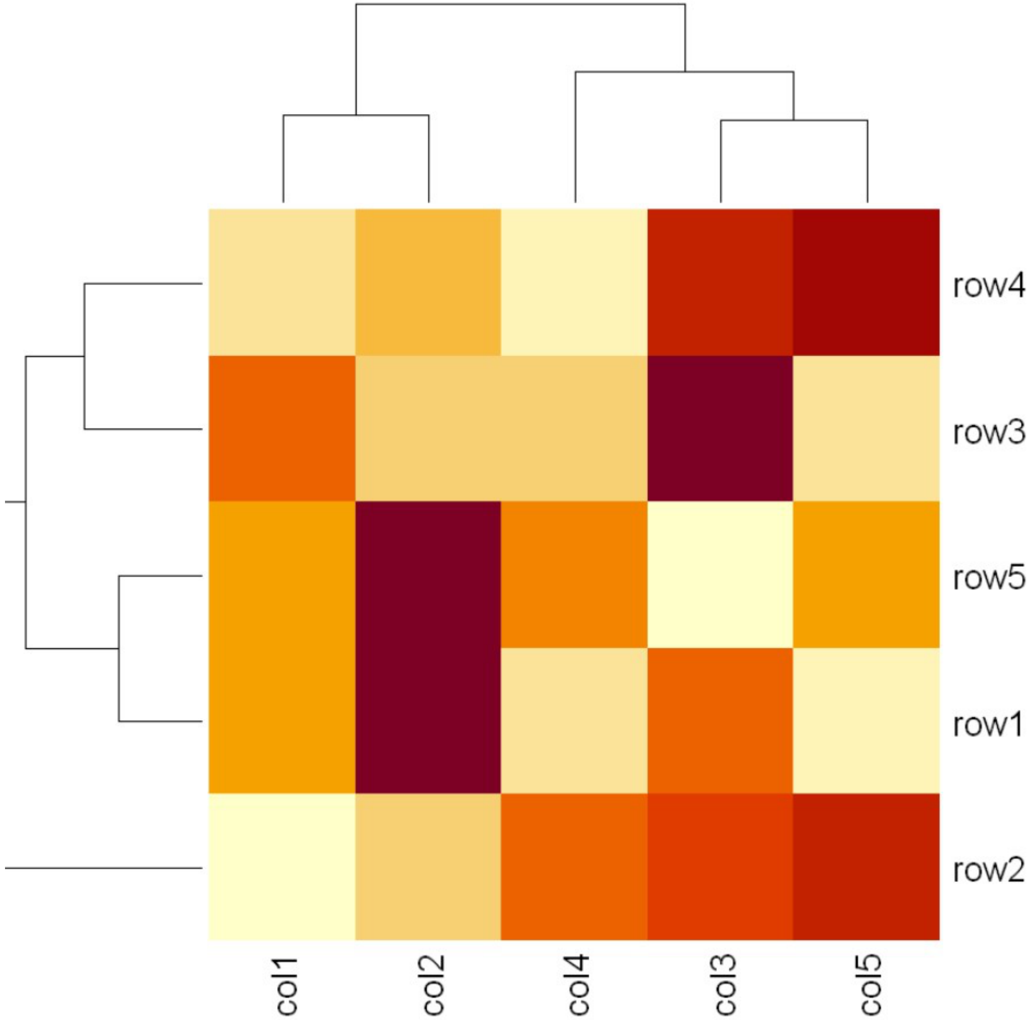
Contd..

- To show whether an association exists between bivariate data.
- To measure the strength and direction of such a relationship.

Heat Map

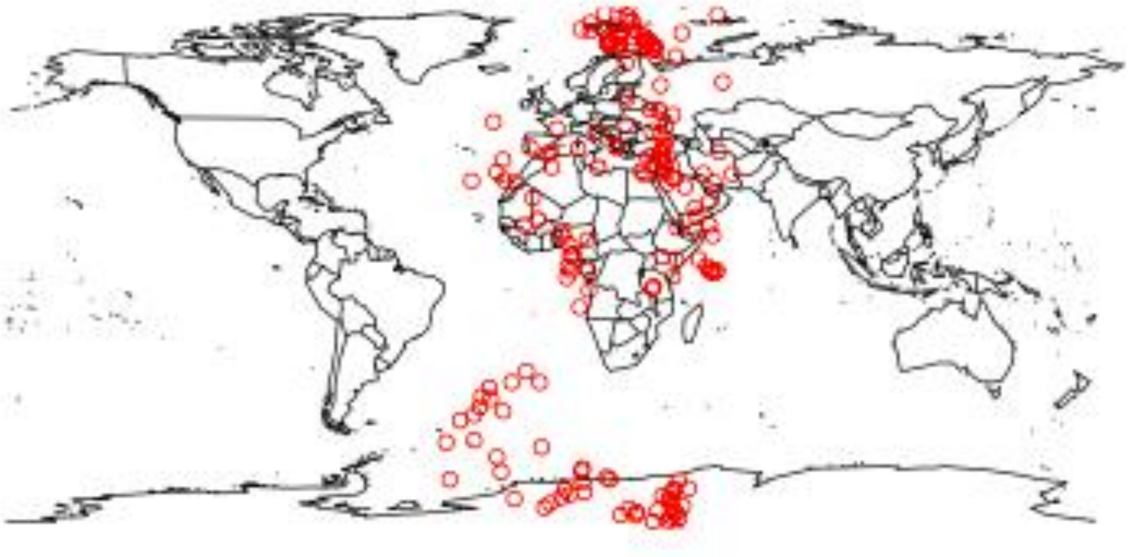
- graphical representation of data using colors to visualize the value of the matrix.

```
# 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)
```



Map visualization in R

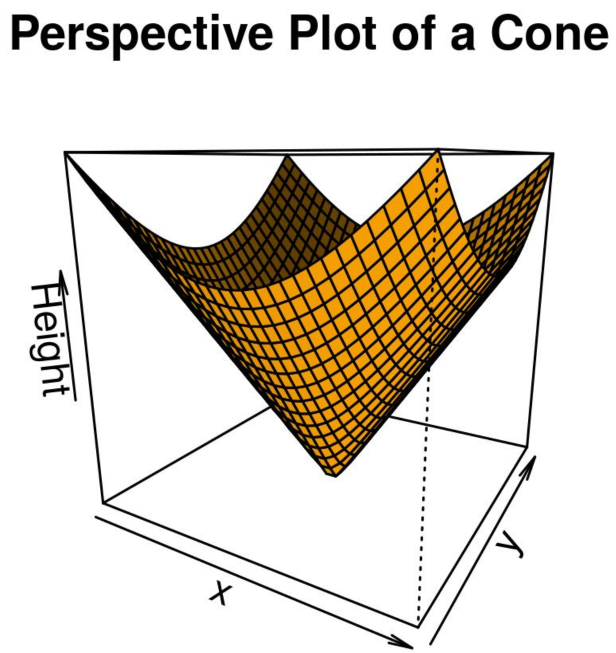
```
# Install and load the required packages
install.packages("maps")
library(maps)

# Read dataset
data <- read.csv("worldcities.csv")

# Convert dataset into a dataframe
df <- data.frame(data)

# Plot world map
map(database = "world")

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



3D Graphs in R

- preps() function
- create 3D surfaces

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


Perspective Plot of a Cone
Height
x
y