Lab Session 6

Importing and Exporting Data and Data visualization

Working with CSV files in R Programming

sample.csv

```
id, name, department, salary, projects
1,   A,     IT,         60754,   4
2,   B,     Tech,       59640,   2
3,   C,     Marketing, 69040,   8
4,   D,     Marketing, 65043,   5
5,   E,     Tech,       59943,   2
6,   F,     IT,         65000,   5
7,   G,     HR,         69000,   7
```

Reading a CSV file

```
df <- read.csv(file = 'sample.csv')
print(df)

# print number of columns
print (ncol(df))

# print number of rows
print(nrow(df))
```

Querying with CSV files

```
min_pro <- min(df$projects)
print(min_pro)


newdf <- subset(df, department == "HR" & projects <10)
print (newdf)
```

Writing into a CSV file

```
write.csv(newdf, "new_sample.csv")
```

```
new_data <-read.csv(file ='new_sample.csv')
print(new_data)
```

The column X contains the row numbers of the original CSV file. In order to remove it, we can specify an additional argument in the write.csv() function that set row names to FALSE.

```
csv_data <- read.csv(file ='sample.csv')

new_csv <- subset(csv_data, department == "HR" & projects <10)

write.csv(new_csv, "new_sample.csv", row.names = FALSE)

new_data <-read.csv(file ='new_sample.csv')

print(new_data)
```

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

Consider the following *airquality* data set for visualization 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

Types of Data Visualizations
Some of the various types of visualizations offered by R are:

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.

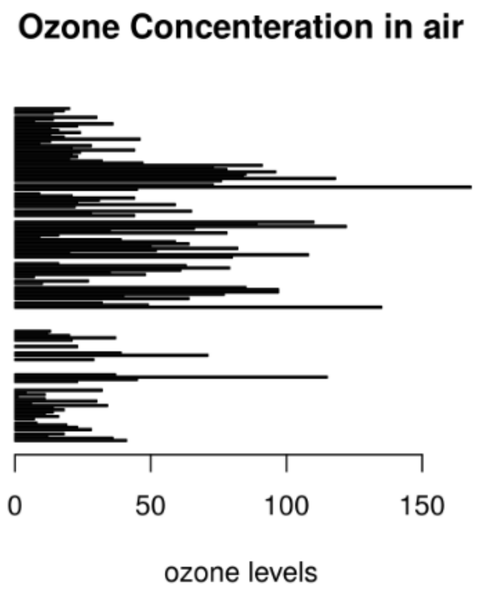
They are generally used for continuous and categorical variable plotting.

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)
```

Output:

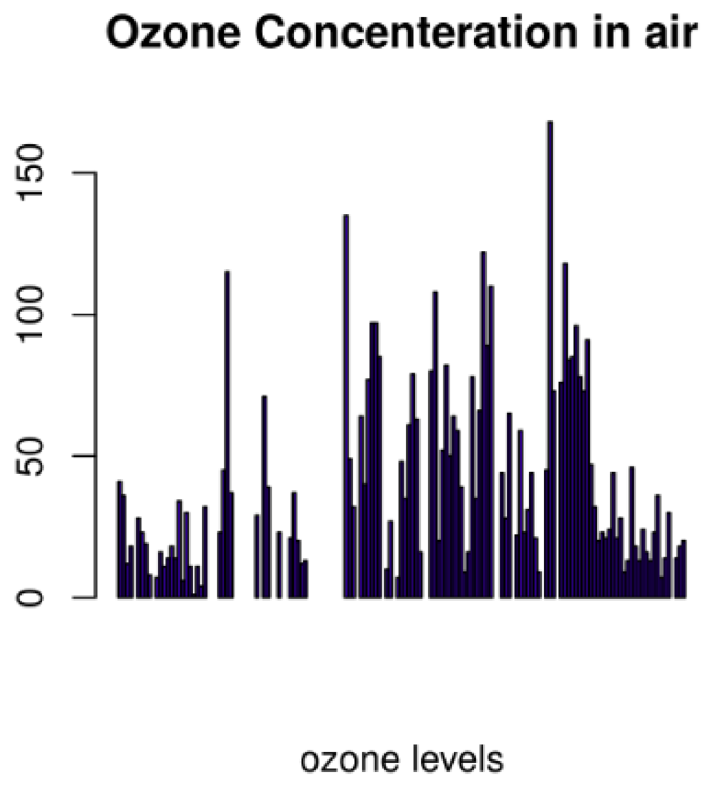


Example 2:

# Vertical Bar Plot for Ozone concentration in air

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

Output:



Bar plots are used for the following scenarios:

- 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.

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.

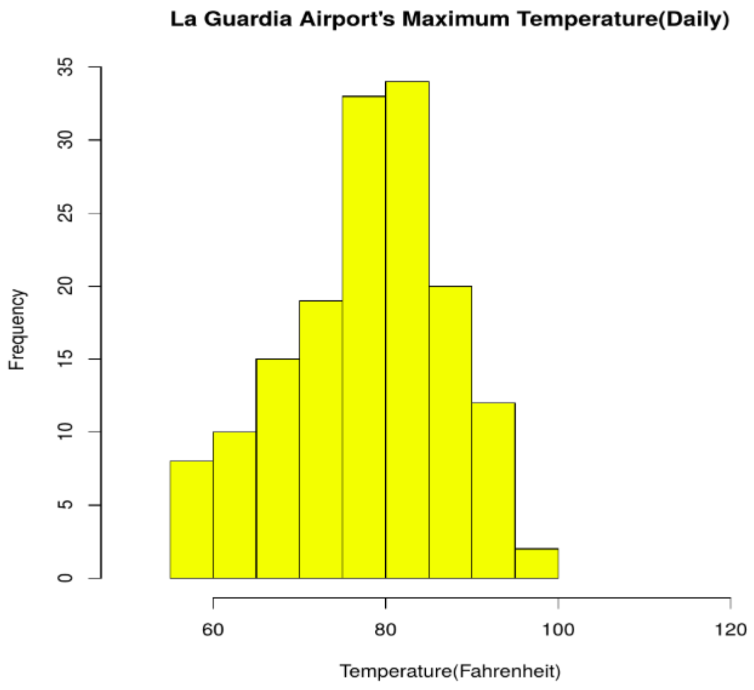
In a Histogram, continuous values are grouped and displayed in these bins whose size can be varied.

Example:

```
# 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)
```

Output:



For a histogram, the parameter **xlim** can be used to specify the interval within which all values are to be displayed.
Another parameter **freq** when set to *TRUE* denotes the frequency of the various values in the histogram.

Histograms are used in the following scenarios:

- To verify an equal and symmetric distribution of the data.
- To identify deviations from expected values.

Box Plot

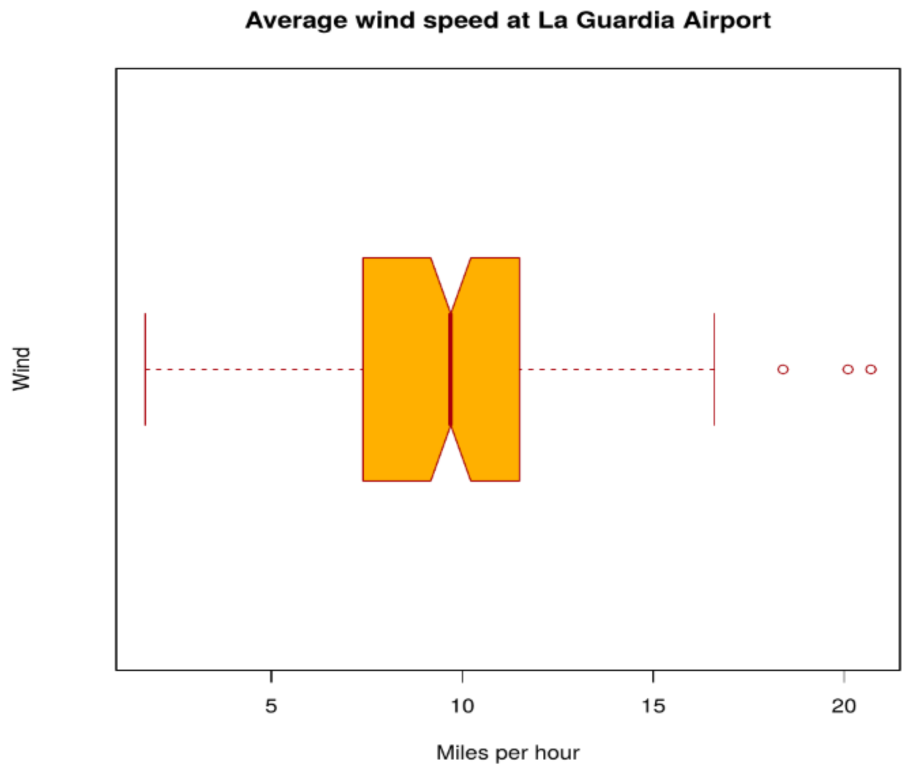
The statistical summary of the given data is presented graphically using a boxplot. A boxplot depicts information like the minimum and maximum data point, the median value, first and third quartile, and interquartile range.

Example:

# 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)
```

Output:



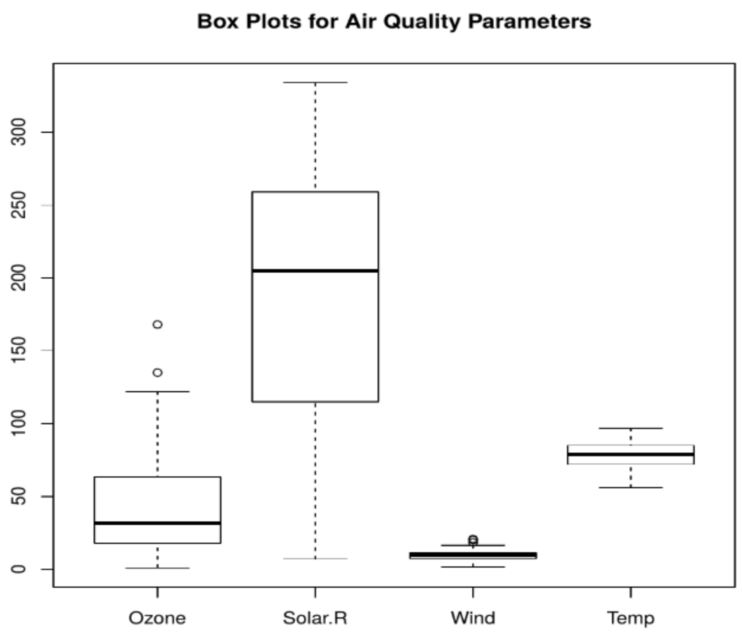
Multiple box plots can also be generated at once through the following code:

Example:

```
# Multiple Box plots, each representing an Air Quality Parameter

boxplot(airquality[, 1:4],  main ='Box Plots for Air Quality Parameters')
```

Output:



Box Plots are used for:

- **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.**

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**.**

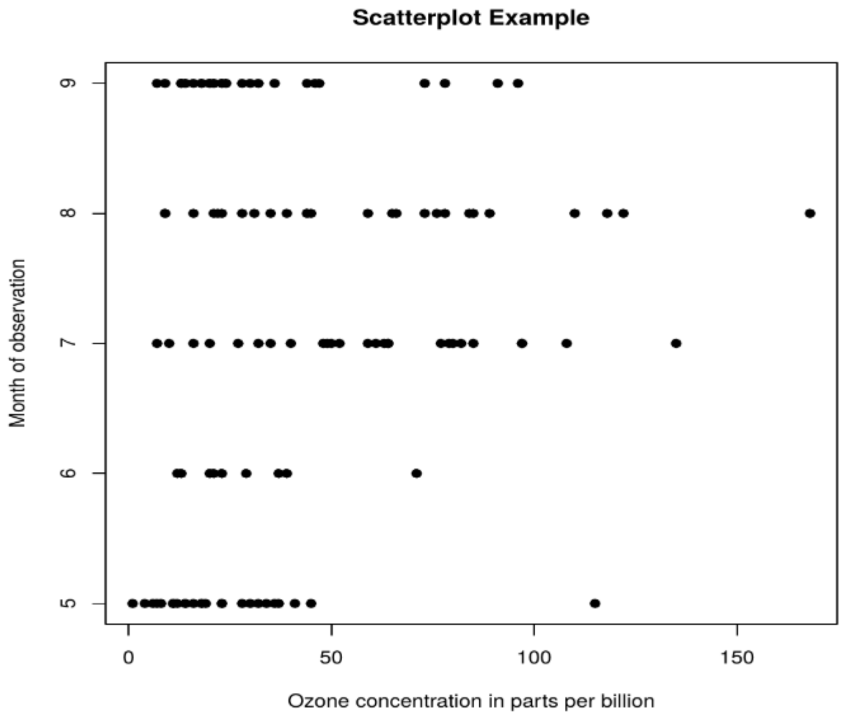
Example:

```
# Scatter plot for Ozone Concentration per month

plot(airquality$Ozone, airquality$Month,      main ="Scatterplot Example",

    xlab ="Ozone Concentration in parts per billion",     ylab =" Month of
observation ")
```

Output:



Scatter Plots are used in the following scenarios:

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

Heat Map

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

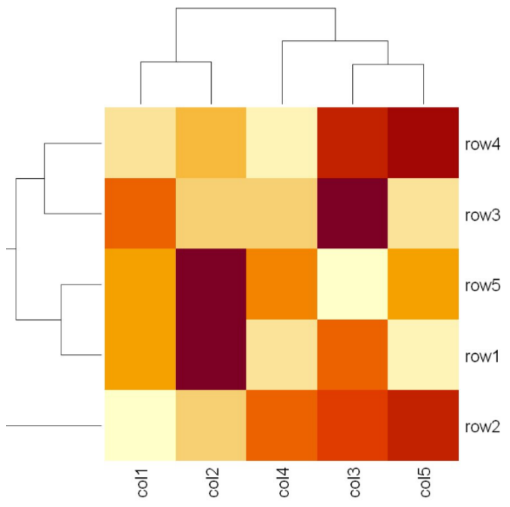
Syntax: heatmap(data)

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(25, 0, 5), nrow = 5, ncol = 5)
# Column names
colnames(data) <- paste("col", 1:5)
rownames(data) <- paste("row", 1:5)
  # Draw a heatmap
heatmap(data)
```

Output:



Map visualization in R

Here we are using maps package to visualize 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")
```

Output: