NAME:Aditya Somani CLASS:TE-A ROLL NO:T1851061

ASSIGNMENT-4

Part B : Assignments based on R & Python

Aim:

To Visualize the data using R/Python by plotting the graphs for assignment no. 2 and 3

Introduction

R studio Supports following types of Charts & Graphs:

- Pie chart • Bar chart • Box plots • Histograms • Line graphs • Scatter plots

Pie Chart:

- R Programming language has numerous libraries to create charts and graphs.
- A pie-chart is a representation of values as slices of a circle with different colors.
- The slices are labeled and the numbers corresponding to each slice is also represented in the chart.
- In R the pie chart is created using the pie() function which takes positive numbers as a vector input.
- The additional parameters are used to control labels, color, title etc.

Pie Chart-Syntax:

- The basic syntax for creating a pie-chart using the R is − pie(x, labels, radius, main, col, clockwise)
- Following is the description of the parameters used −– x is a vector containing the numeric values used in the pie chart.
- labels is used to give description to the slices.
- radius indicates the radius of the circle of the pie chart.(value between −1 and +1). – main indicates the title of the chart.
- col indicates the color palette.
- clockwise is a logical value indicating if the slices are drawn clockwise or anti clockwise.

Pie Chart Example:

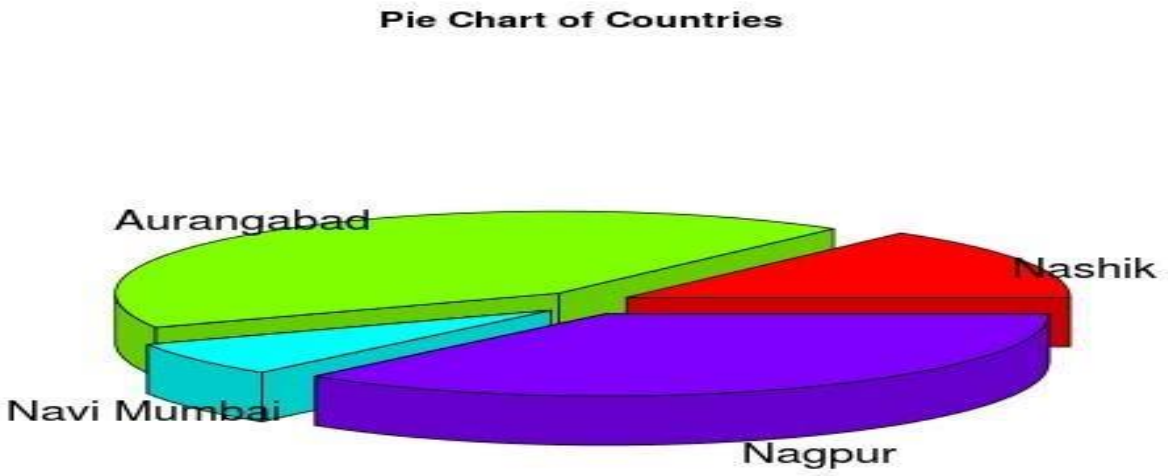
```
# Create data for the graph.
x <- c(21, 62, 10, 53)
labels <- c("London", "New York", "Singapore",
"Mumbai")

# Give the chart file a name.
png(file = "city.png")

# Plot the chart.
pie(x,labels)

# Save the file.
dev.off()
```

Pie Chart With Colors:

# Create data for the graph.
x <- c(21, 62, 10, 53) labels <- c("Pune", "Nashik", "Aurangabad", "Mumbai") # Give the chart file a name. png(file = "city_title_colours.png")
# Plot the chart with title and rainbow color pallet.
pie(x, labels, main = "City pie chart", col = rainbow(length(x)))
# Save the file.
dev.off()

3D Pie Chart:

# Get the library.
library(plotrix)
# Create data for the graph.
x <- c(21, 62, 10,53)
lbl <- c("Nashik","Aurangabad","Navi Mumbai","Nagpur")
png(file = "3d_pie_chart.png") # Plot the chart.
pie3D(x,labels = lbl,explode = 0.1, main = "Pie Chart of
Countries ")
dev.off()



Bar Charts:

- A bar chart represents data in rectangular bars with length of the bar proportional to the value of the variable.
- R uses the function barplot() to create bar charts.
- R can draw both vertical and horizontal bars in the bar chart.
- In bar chart each of the bars can be given different colors.

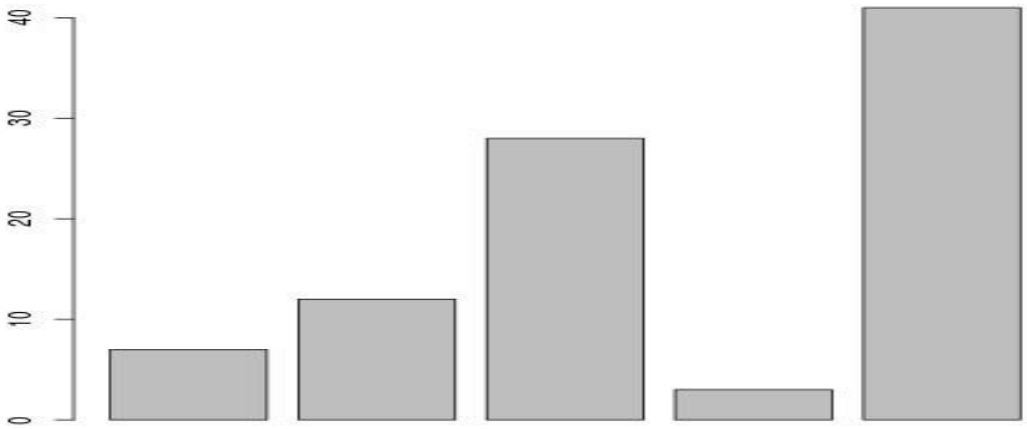
Bar Charts- Syntax:

- The basic syntax to create a bar-chart in R is − barplot(H, xlab, ylab, main, names.arg, col)
- Following is the description of the parameters used −
- H is a vector or matrix containing numeric values used in bar chart.
- xlab is the label for x axis.
- ylab is the label for y axis.
- main is the title of the bar chart.
- names.arg is a vector of names appearing under each bar. – col is used to give colors to the bars in the graph.

Barchart Example:

```
# Create the data for the
chart. H <- c(7,12,28,3,41)
# Give the chart file a name.
```

```
png(file = "barchart.png") #
Plot the bar chart.
barplot(H) #
Save the
file.
dev.off()
```

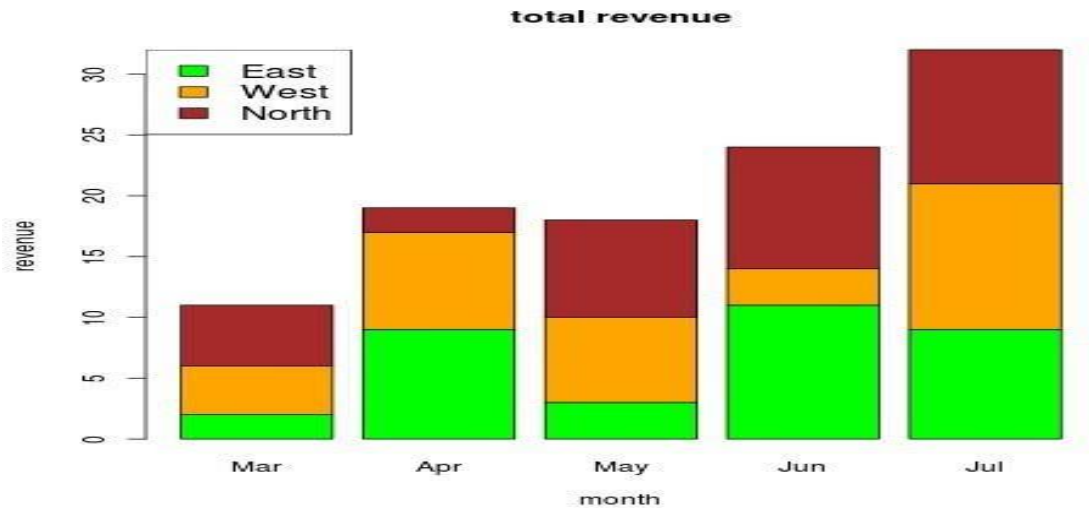


Barchart with Attribute:

```
# Create the data for the chart.
H <- c(7,12,28,3,41)
M <- c("Mar","Apr","May","Jun","Jul")
# Give the chart file a name. png(file =
"barchart_months_revenue.png") # Plot
the bar chart.
barplot(H,names.arg = M,xlab = "Month",ylab =
"Revenue",col = "blue", main = "Revenue chart",border
= "red")
dev.off()
```

Barchart Stacked:

```
colors <- c("green","orange","brown")
months <-
c("Mar","Apr","May","Jun","Jul") regions
<- c("East","West","North")
Values <- matrix(c(2,9,3,11,9,4,8,7,3,12,5,2,8,10,11),nrow =
3,ncol = 5,byrow = TRUE)
png(file =
"barchart_stacked.png")
barplot(Values,main = "total revenue",names.arg = months,xlab =
"month",ylab = "revenue", col = colors)
legend("topleft", regions, cex = 1.3, fill = colors)
dev.off()
```



Boxplot:

- Boxplots are a measure of how well distributed is the data in a data set.
- It divides the data set into three quartiles. This graph represents the minimum, maximum, median, first quartile and third quartile in the data set.
- It is also useful in comparing the distribution of data across data sets by drawing boxplots for each of them.
- Boxplots are created in R by using the boxplot() function.

Boxplot Syntax:

- The basic syntax to create a boxplot in R is − boxplot(x, data, notch, varwidth, names, main)
- Following is the description of the parameters used − – x is a vector or a formula – data is the data frame.
- notch is a logical value. Set as TRUE to draw a notch. – varwidth is a logical value. Set as true to draw width of the box proportionate to the sample size.
- names are the group labels which will be printed under each boxplot. – main is used to give a title to the graph.

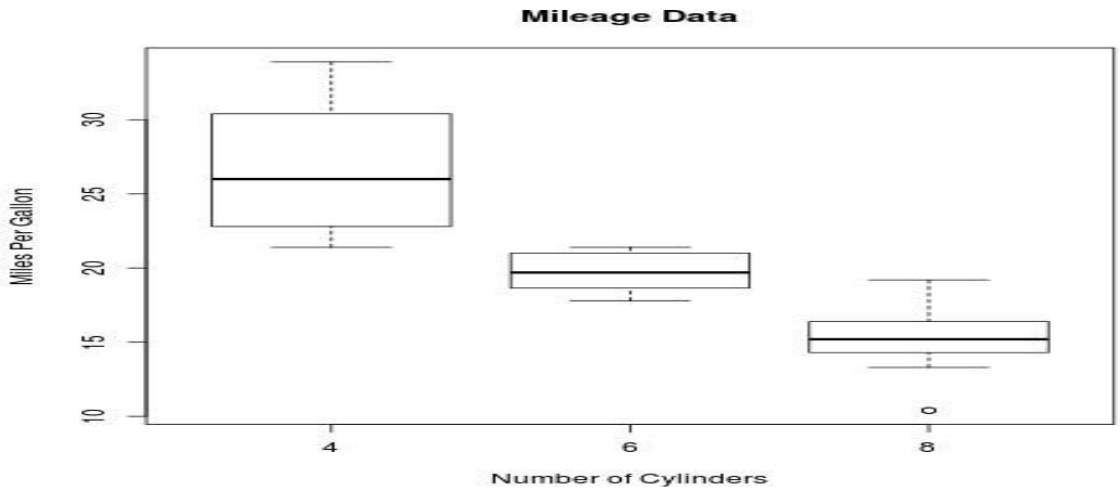
Boxplot Example:

• We use the data set "mtcars" available in the R environment to create a basic boxplot. Let's look at the columns "mpg" and "cyl" in mtcars.

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

```
> 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
```

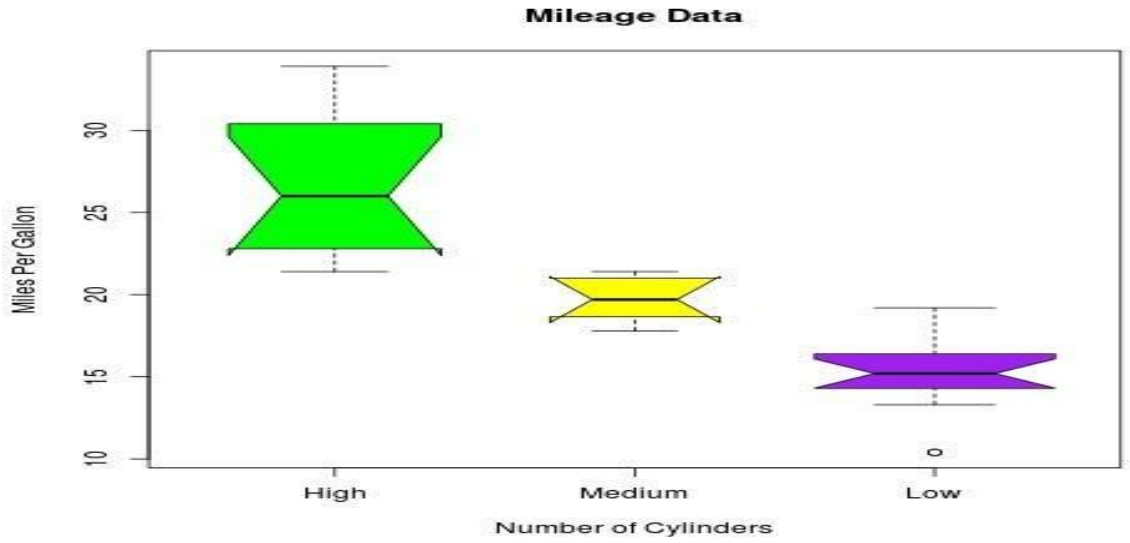
Give the chart file a name. png(file = "boxplot.png") # Plot the chart.
boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders", ylab = "Miles Per Gallon", main = "Mileage Data") # Save the file.
dev.off()



Boxplot with Notch:

png(file = "boxplot_with_notch.png")
# Plot the chart.
boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders", ylab = "Miles Per Gallon", main = "Mileage

Data", notch = TRUE, varwidth
= TRUE,
col = c("green","yellow","purple"),
names = c("High","Medium","Low"))
dev.off()



Histogram:

- A histogram represents the frequencies of values of a variable bucketed into ranges.
- Histogram is similar to bar chart but the difference is it groups the values into continuous ranges.
- Each bar in histogram represents the height of the number of values present in that range.
- R creates histogram using hist() function. This function takes a vector as an input and uses some more parameters to plot histograms.

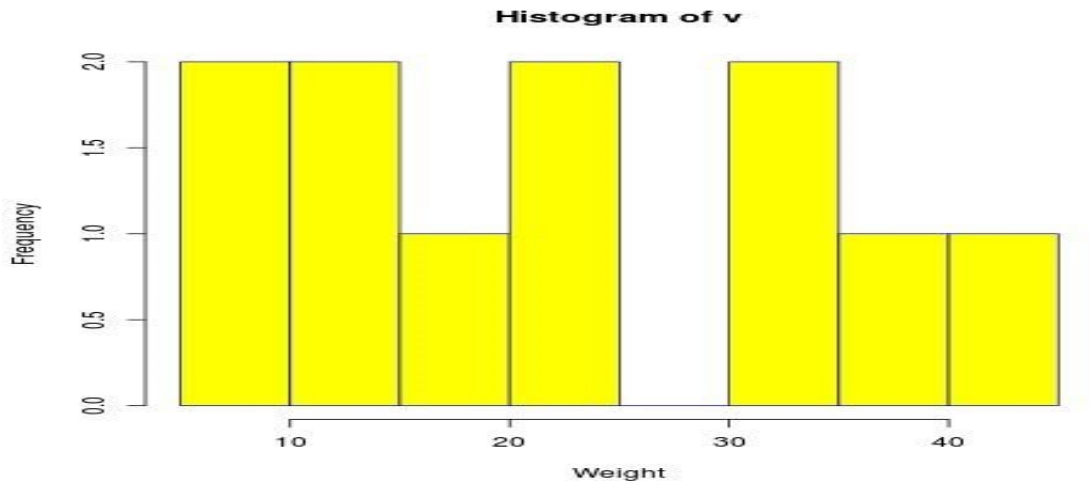
Histogram Syntax:

- The basic syntax for creating a histogram using R is −hist(v,main,xlab,xlim,ylim,breaks,col,border)
- Following is the description of the parameters used − v is a vector containing numeric values used in histogram. main indicates title of the chart. col is used to set color of the bars. border is used to set border color of each bar.

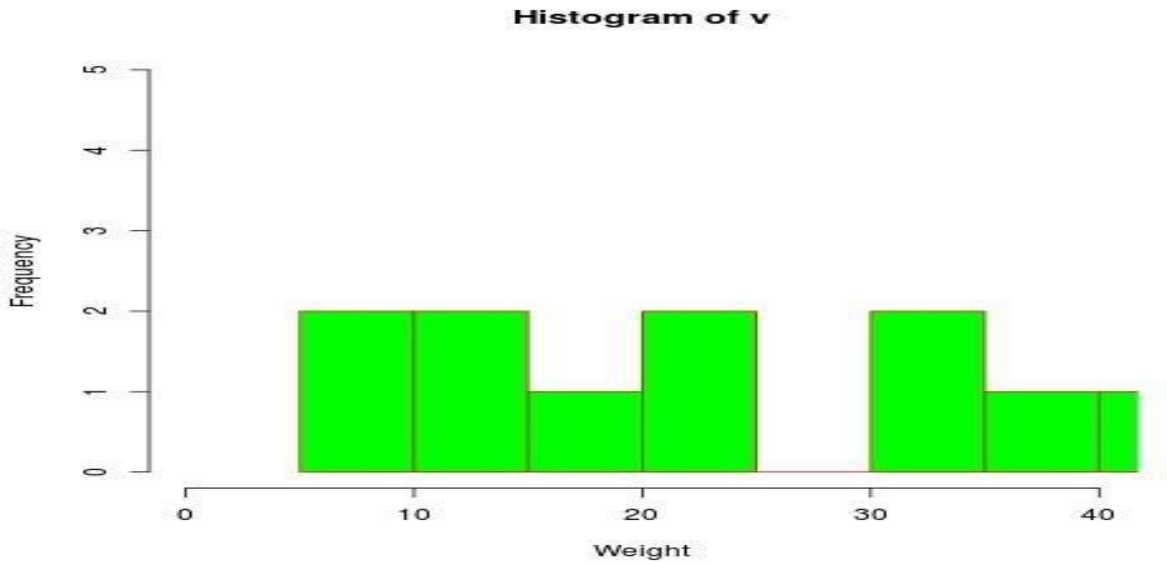
xlab is used to give description of x-axis.
xlim is used to specify the range of values on the x-axis.
ylim is used to specify the range of values on the y-axis.
breaks is used to mention the width of each bar.

Histogram Example:

```
# Create data for the graph.
v <- c(9,13,21,8,36,22,12,41,31,33,19) # Give the
chart file a name. png(file = "histogram.png") #
Create the histogram. hist(v,xlab = "Weight",col =
"yellow",border = "blue") # Save the file.
dev.off()
```




```
# Create data for the graph.
v <- c(9,13,21,8,36,22,12,41,31,33,19) #
Give the chart file a name. png(file =
"histogram_lim_breaks.png") # Create the
histogram. hist(v,xlab = "Weight",col =
"green",border = "red", xlim = c(0,40), ylim
= c(0,5), breaks = 5) dev.off()
```



Line Graph:

- A line chart is a graph that connects a series of points by drawing line segments between them.
- These points are ordered in one of their coordinate (usually the x-coordinate) value.
- Line charts are usually used in identifying the trends in data.
- The plot() function in R is used to create the line graph.

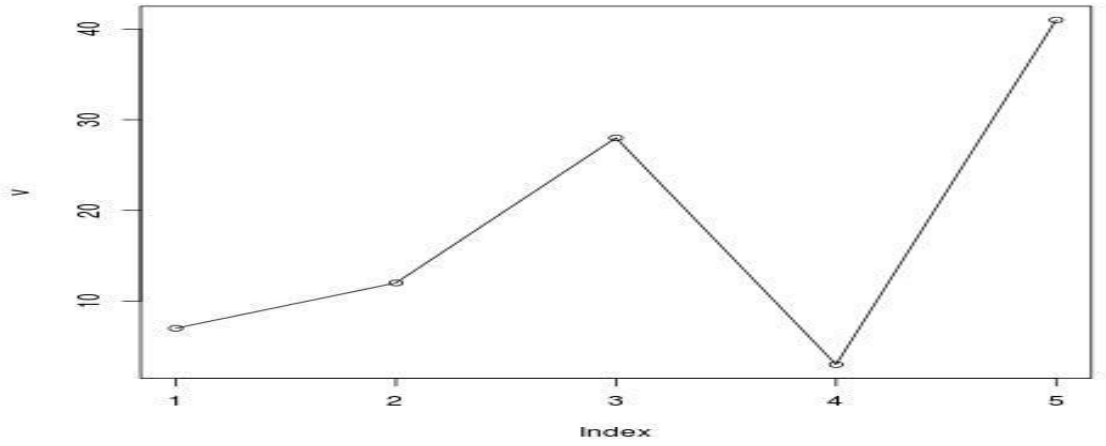
Line Graph – Syntax:

- The basic syntax to create a line chart in R is − plot(v,type,col,xlab,ylab) • Following is the description of the parameters used − – v is a vector containing the numeric values.
- type takes the value "p" to draw only the points, "i" to draw only the lines and "o" to draw both points and lines.
- xlab is the label for x axis.
- ylab is the label for y axis.
- main is the Title of the chart.
- col is used to give colors to both the points and lines

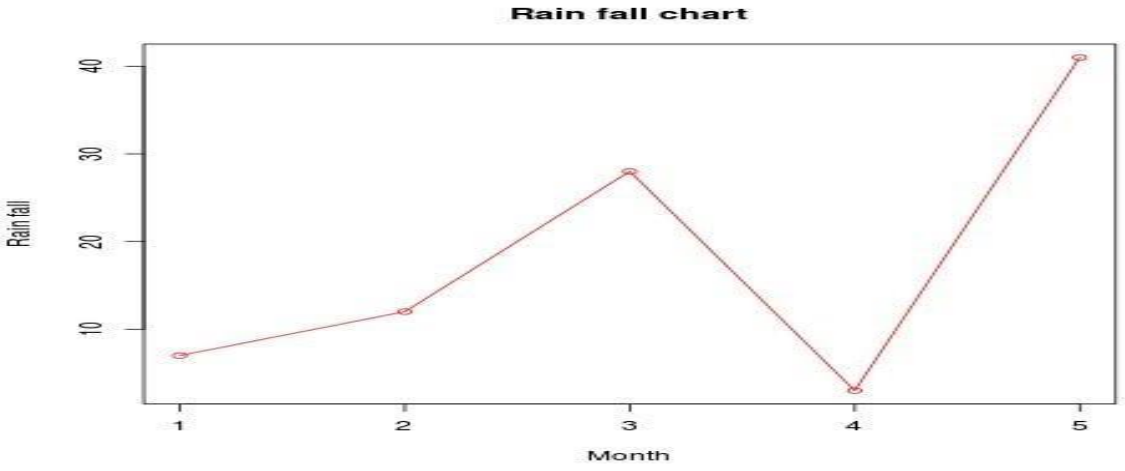
Line Graph-Example:

```
# Create the data for the chart.
v <- c(7,12,28,3,41) #
Give the chart file a
name. png(file =
"line_chart.png") # Plot
```

the line graph. plot(v,type
= "o")
# Save the file.
dev.off()

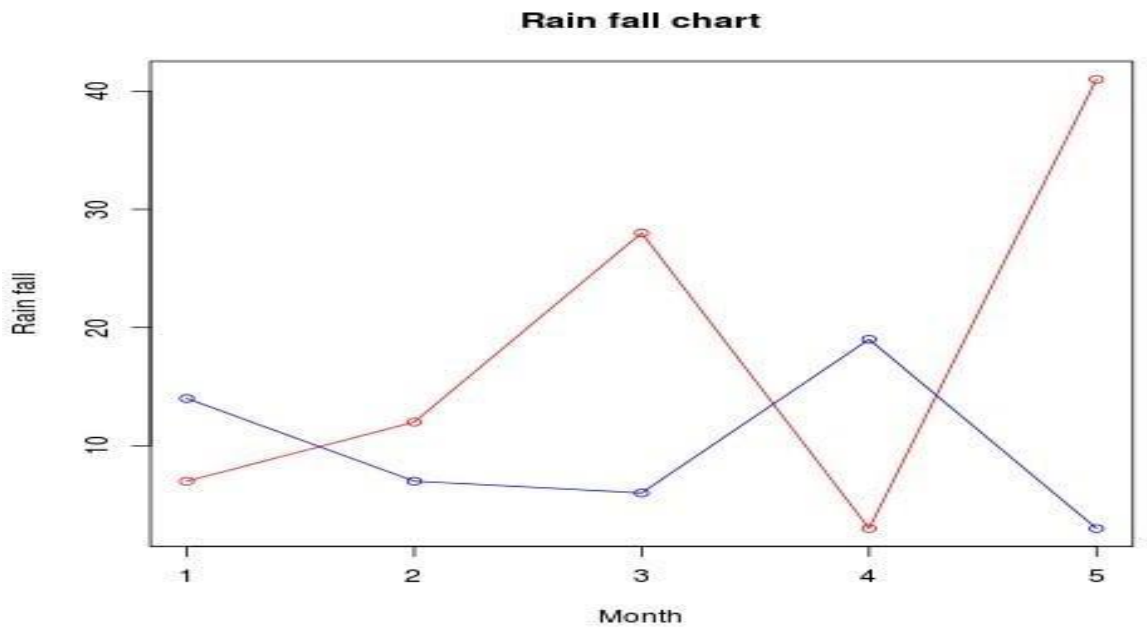


# Create the data for the chart.
v <- c(7,12,28,3,41) # Give the chart file a
name. png(file =
"line_chart_label_colored.png") # Plot the
bar chart. plot(v,type = "o", col = "red",
xlab = "Month", ylab = "Rain fall", main =
"Rain fall chart") # Save the file.
dev.off()



Multiple Lines in Chart-Example:

```
# Create the data for the chart.
v <-
c(7,12,28,3,41) t <-
c(14,7,6,19,3)
# Give the chart file a name. png(file =
"line_chart_2_lines.png") # Plot the bar chart.
plot(v,type = "o",col = "red", xlab = "Month", ylab =
"Rain fall", main = "Rain fall chart") lines(t, type = "o",
col = "blue") dev.off()
```



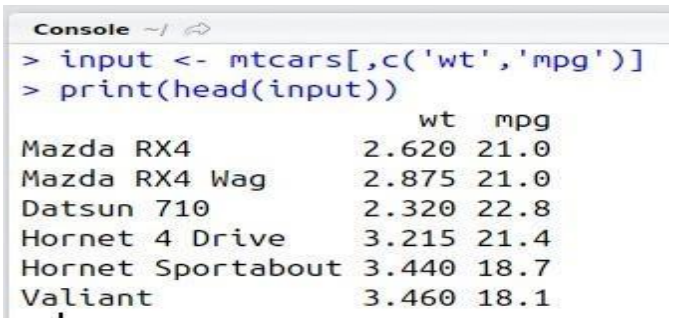
Scatter plot:

- Scatterplots show many points plotted in the Cartesian plane.
- Each point represents the values of two variables.
- One variable is chosen in the horizontal axis and another in the vertical axis.
- The simple scatterplot is created using the plot() function.

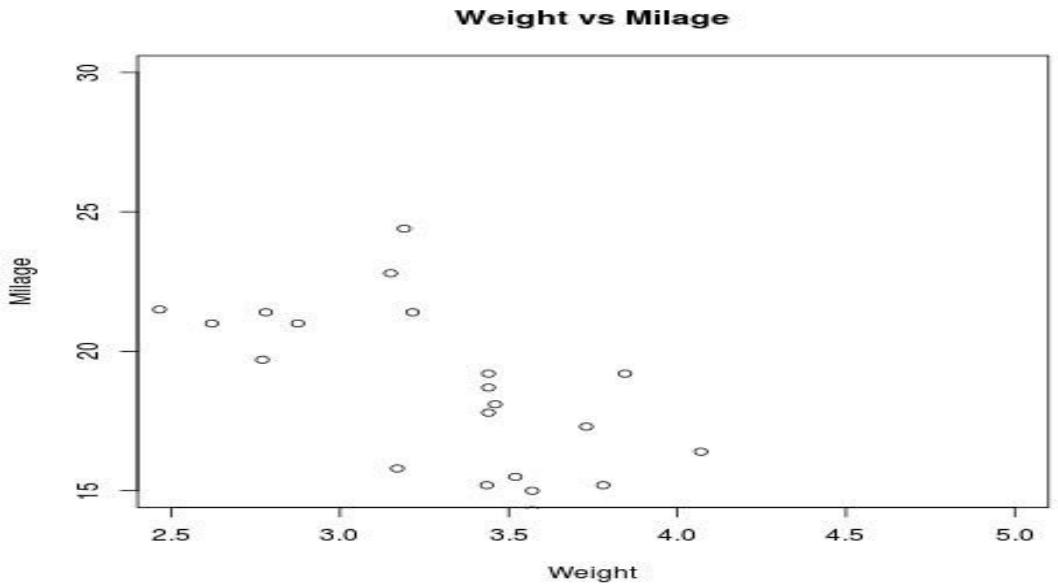
Scatterplot Example:

- We use the data set "mtcars" available in the R environment to create a basic scatterplot.
- Let's use the columns "wt" and "mpg" in mtcars.

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




```
# Get the input values.
input <- mtcars[,c('wt','mpg')]
png(file = "scatterplot.png")
# Plot the chart for cars with weight between 2.5 to 5 and
mileage between 15 and 30.
plot(x = input$wt,y =
input$mpg, xlab = "Weight",
ylab = "Milage", xlim =
c(2.5,5), ylim = c(15,30), main
= "Weight vs Milage"
)
dev.off()
```



Scatter plot Matrices:

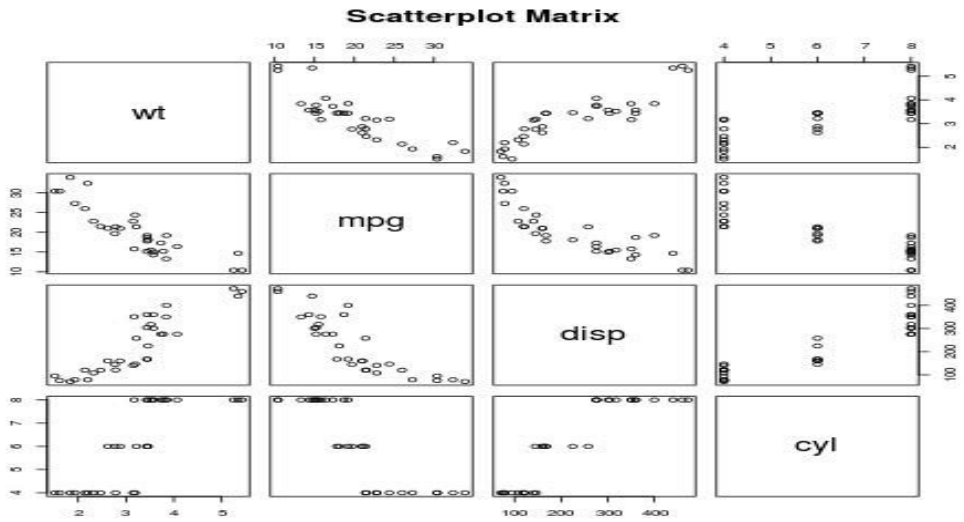
- When we have more than two variables and we want to find the correlation between one variable versus the remaining ones we use scatterplot matrix.
- We use pairs() function to create matrices of scatterplots.
- Syntax:

pairs(formula, data)

- Following is the description of the parameters used −
  formula represents the series of variables used in pairs. *data* represents the data set from which the variables will be taken.

Scatterplot Matrices- Example:

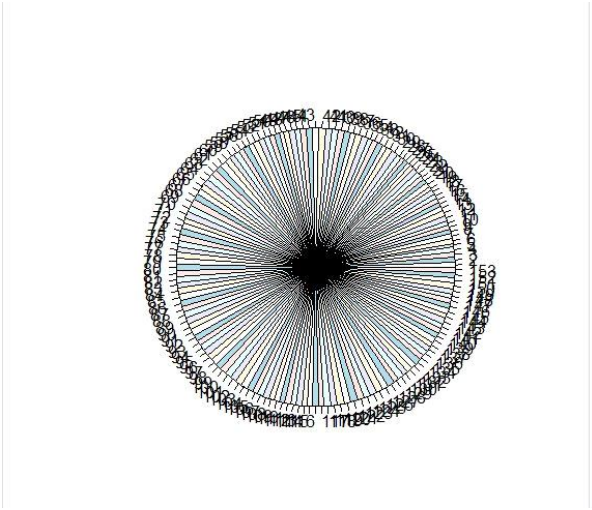
```
# Give the chart file a name.
png(file =
"scatterplot_matrices.png")
# Plot the matrices between 4 variables giving 12
plots.
# One variable with 3 others and total 4 variables.
pairs(~wt+mpg+disp+cyl,data = mtcars,main =
"Scatterplot Matrix")
dev.off()
```



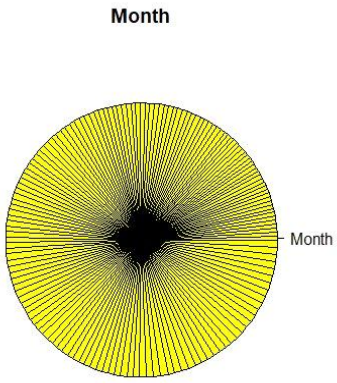
Conclusion: Thus we have learnt how to Visualize the data with different types of Graphs & Charts in R Studio.

EXECUTION:

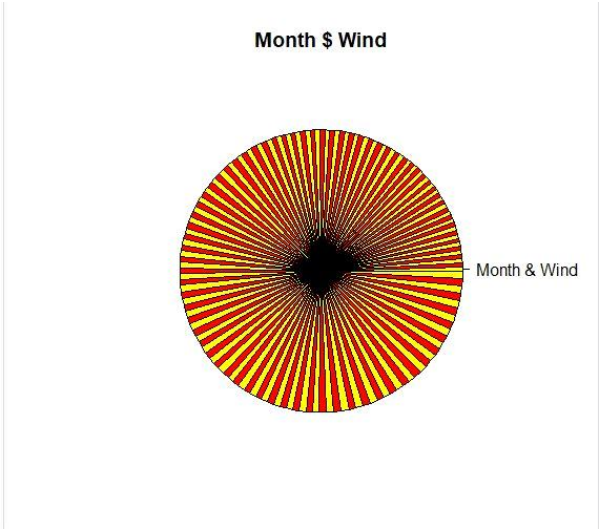
```
> pie(airquality$Temp)
```



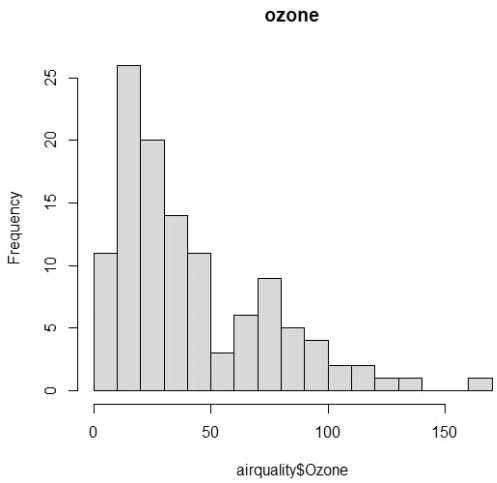
```
> pie(airquality$Month,labels="Month",col = "Yellow",main = "Month")
```



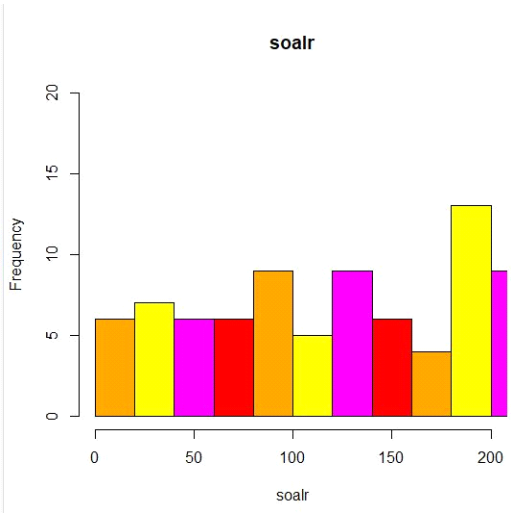
> pie(airquality$Month,airquality$Wind,labels="Month & Wind",col = c("Yellow","red"),main = "Month $ Wind")



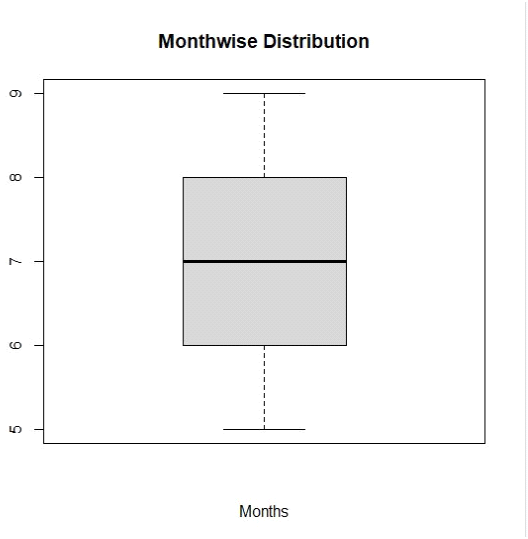
> hist(airquality$Ozone,breaks = 20, main = "ozone")



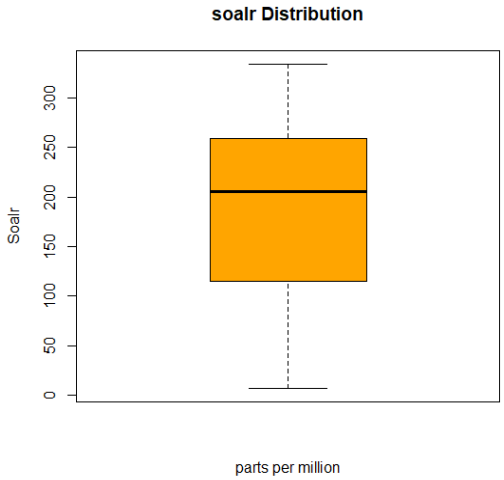
> hist(airquality$Solar.R,breaks = 20, main = "soalr",xlab = "soalr",xlim = c(0,200),ylim = c(0,20),col = c("orange","yellow","magenta","red"))



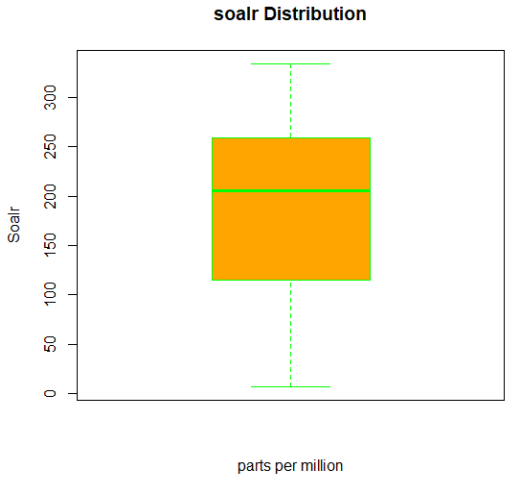
> boxplot(airquality$Month,main = "Monthwise Distribution",xlab = "Months")



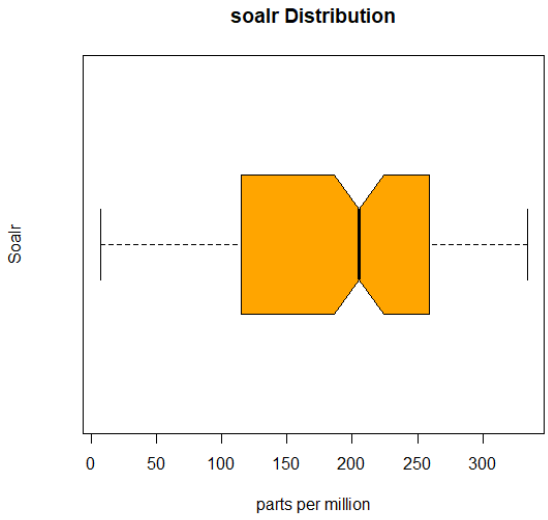
> boxplot(airquality$Solar.R, main = "soalr Distribution",xlab = "parts per million",ylab ="Soalr",col = "orange")



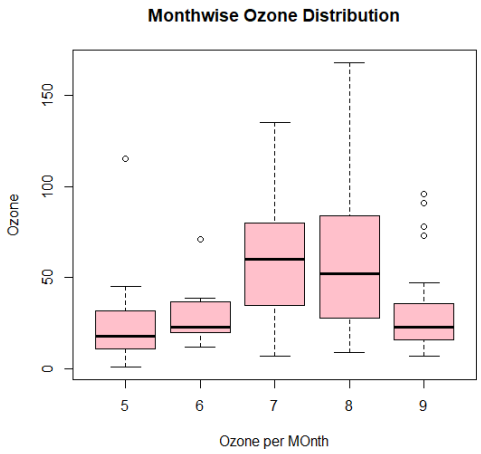
> boxplot(airquality$Solar.R, main = "soalr Distribution",xlab = "parts per million",ylab ="Soalr",col = "orange",border = "green")



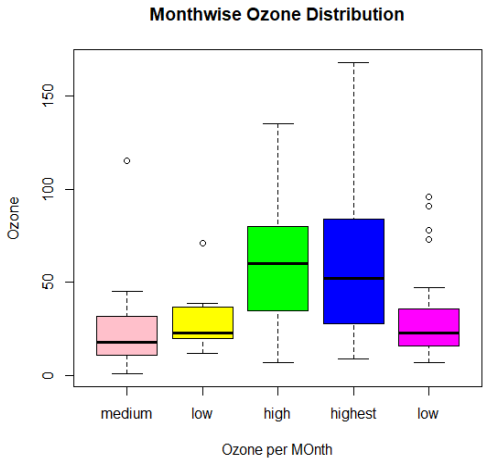
> boxplot(airquality$Solar.R, main = "soalr Distribution",xlab = "parts per million",ylab ="Soalr",col = "orange",notch = TRUE,horizontal = TRUE)



> boxplot(airquality$Ozone~airquality$Month, main = "Monthwise Ozone Distribution",xlab = "Ozone per MOnth",ylab ="Ozone",col = "pink")

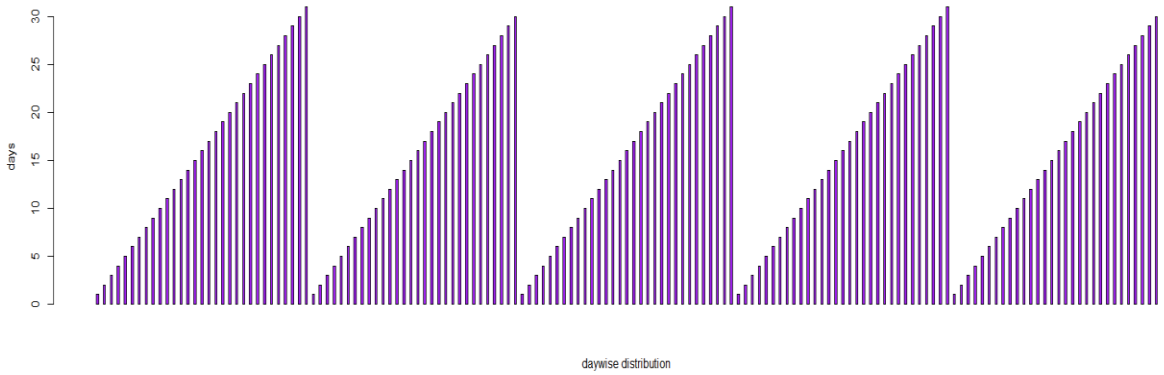


> boxplot(airquality$Ozone~airquality$Month, main = "Monthwise Ozone Distribution",xlab = "Ozone per MOnth",ylab ="Ozone",col = c("pink","yellow","green","blue","magenta"),names=c("medium","low","high","highest","low"))



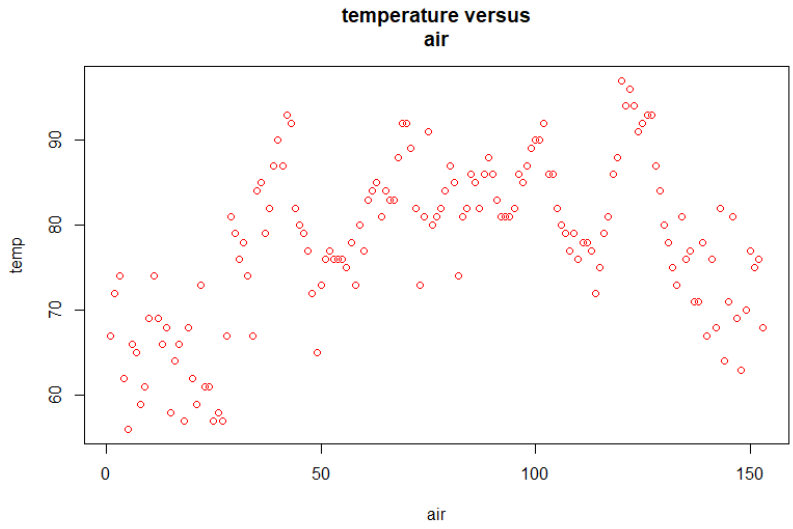
> barplot(airquality$Day)

> barplot(airquality$Day,xlab = "daywise distribution", ylab = "days",space = 2,col = "purple")



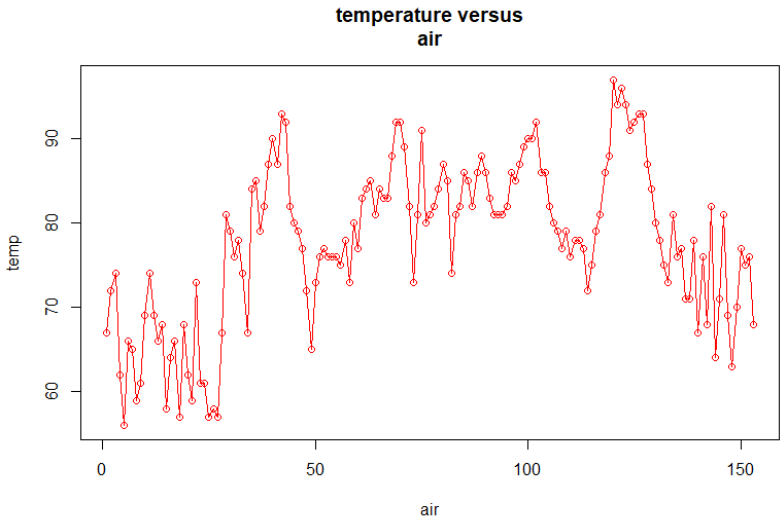
> plot(airquality$Temp,type="p",main="temperature versus

+ air",col="red",xlab="air",ylab="temp")

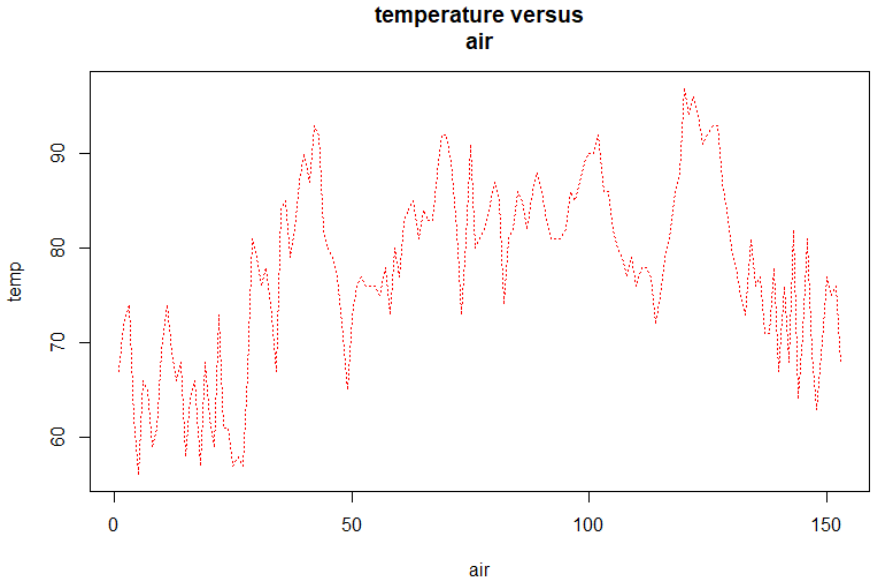


> plot(airquality$Temp,type="o",main="temperature versus

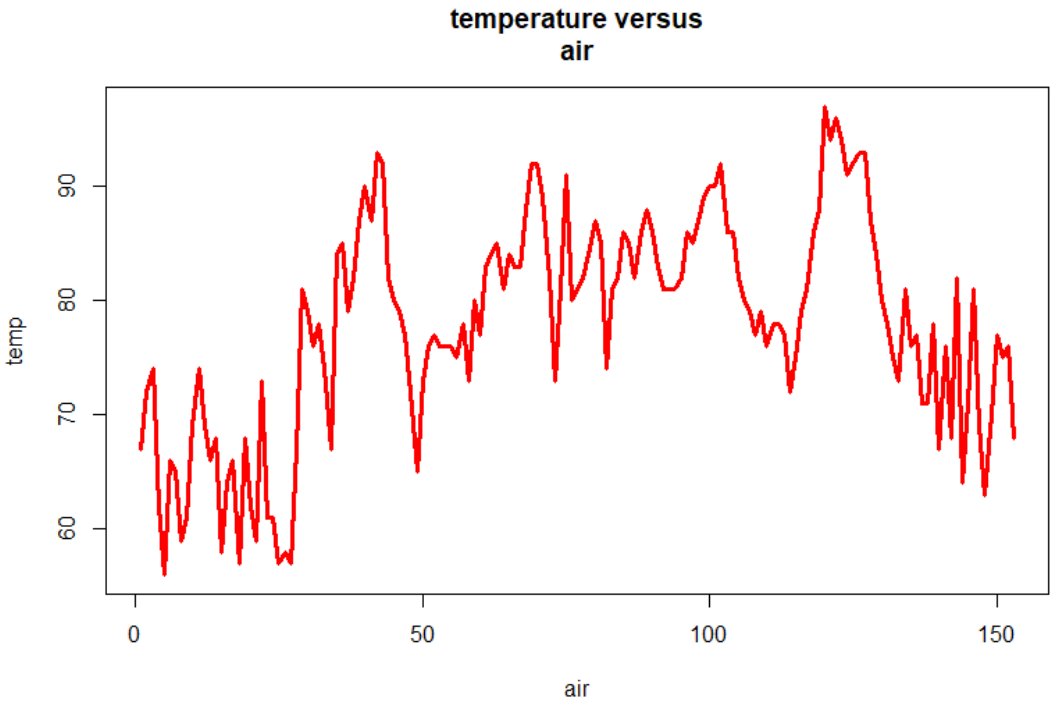
+ air",col="red",xlab="air",ylab="temp")




```
> plot(airquality$Temp,type="l",main="temperature versus
+ air",col="red",xlab="air",ylab="temp",lty=3)
```




```
> plot(airquality$Temp,type="l",main="temperature versus
+ air",col="red",xlab="air",ylab="temp",lty=1,lwd=3)
```




```
> plot(airquality$Temp,type="o",main="temperature versus
+ air",col="red",xlab="air",ylab="temp",lty=1,lwd=2,pch=8)
```