PLOTTING AND PLINTING IN R

KERAMA EZEKIEL JAMES

2023-03-29

WELCOME TO R PROGRAMMING AND ENJOY

data()  
data("Orange")  
data("mtcars")  
View(Orange)  
attach(Orange)  
ncol(Orange)

## [1] 3

nrow(Orange)

## [1] 35

names(Orange)

## [1] "Tree" "age" "circumference"

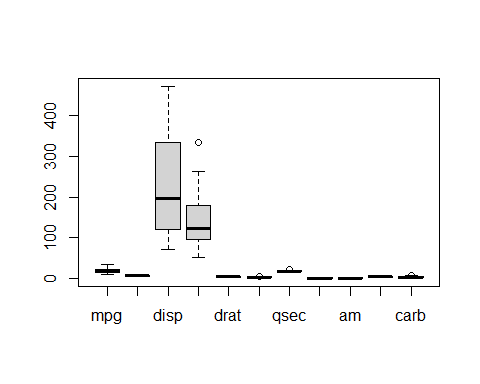
data()  
data("mtcars")  
data("pressure")  
View(pressure)  
df <- data.frame(name = c("John", "Lisa", "Tom", "Sarah"),  
 age = c(25, 30, 20, 35))  
  
# Sort the data frame based on the age column  
sorted\_df <- df[order(df$age),]  
  
# Print the sorted data frame  
print(sorted\_df)

## name age  
## 3 Tom 20  
## 1 John 25  
## 2 Lisa 30  
## 4 Sarah 35

View(df <- data.frame(name = c("John", "Lisa", "Tom", "Sarah"),  
 age = c(25, 30, 20, 35)))  
  
  
#How to do sorting in R programming  
# Create a data frame  
df <- data.frame(name = c("John", "Lisa", "Tom", "Sarah"),  
 age = c(25, 30, 20, 35))  
  
# Sort the data frame based on the age column  
sorted\_df <- df[order(df$age),]  
  
# Print the sorted data frame  
print(sorted\_df)

## name age  
## 3 Tom 20  
## 1 John 25  
## 2 Lisa 30  
## 4 Sarah 35

#Sorting of the mtcars  
data("mtcars")  
  
boxplot(mtcars)



data("mtcars", package = "datasets")  
  
#Sorting of the data in an ascending order  
data("mtcars")  
View(mtcars)  
  
sorted\_data <- mtcars[order(mtcars$gear),]  
  
print(sorted\_data$gear)

## [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5

sorted\_data <- mtcars[order(mtcars$mpg),]  
print(sorted\_data$mpg)

## [1] 10.4 10.4 13.3 14.3 14.7 15.0 15.2 15.2 15.5 15.8 16.4 17.3 17.8 18.1 18.7  
## [16] 19.2 19.2 19.7 21.0 21.0 21.4 21.4 21.5 22.8 22.8 24.4 26.0 27.3 30.4 30.4  
## [31] 32.4 33.9

data("swiss")  
View(swiss)  
  
# Load a sample data set  
data("swiss", package = "datasets")  
  
# Sort the data by fertility rate (per capital) in ascending order  
sorted\_data <- swiss[order(swiss$Fertility),]  
  
sorted\_data <- swiss[order(swiss$Agriculture),]  
print(sorted\_data$Agriculture)

## [1] 1.2 7.7 15.2 16.7 17.0 17.6 18.7 19.4 26.8 27.7 34.0 35.3 36.5 37.6 38.4  
## [16] 39.7 43.5 45.1 45.2 46.6 49.5 50.9 53.3 54.1 55.1 58.1 59.8 60.7 60.8 62.0  
## [31] 63.1 63.5 64.5 64.9 67.5 67.8 69.3 70.2 71.2 72.6 73.0 75.9 78.2 84.6 84.9  
## [46] 85.9 89.7

# Print the sorted data by fertility rate  
print(sorted\_data$Fertility)

## [1] 35.0 65.7 54.3 72.7 80.2 64.4 67.6 55.7 58.3 42.8 71.7 76.1 85.8 77.6 70.4  
## [16] 92.5 76.9 83.1 82.9 44.7 65.4 56.6 82.4 57.4 65.0 74.2 65.5 68.9 60.5 64.1  
## [31] 79.3 72.0 87.1 79.4 66.9 92.4 61.7 83.8 72.5 68.3 65.1 65.0 70.5 92.2 69.3  
## [46] 75.5 77.3

# Sort the data by infant mortality rate in ascending order for females  
sorted\_data <- swiss[order(swiss$Infant.Mortality[swiss$Catholic < median(swiss$Catholic)]),]  
  
# Print the sorted data by infant mortality rate for females  
print(sorted\_data$Infant.Mortality[sorted\_data$Catholic < median(sorted\_data$Catholic)])

## [1] 19.1 20.0 18.0 22.4 10.8 20.0 22.7 22.2 16.5 20.6 18.7

#How to attach data  
attach(swiss)  
names(swiss)

## [1] "Fertility" "Agriculture" "Examination" "Education"   
## [5] "Catholic" "Infant.Mortality"

#The head gives the first six rows and six columns of the data set  
head(swiss)

## Fertility Agriculture Examination Education Catholic  
## Courtelary 80.2 17.0 15 12 9.96  
## Delemont 83.1 45.1 6 9 84.84  
## Franches-Mnt 92.5 39.7 5 5 93.40  
## Moutier 85.8 36.5 12 7 33.77  
## Neuveville 76.9 43.5 17 15 5.16  
## Porrentruy 76.1 35.3 9 7 90.57  
## Infant.Mortality  
## Courtelary 22.2  
## Delemont 22.2  
## Franches-Mnt 20.2  
## Moutier 20.3  
## Neuveville 20.6  
## Porrentruy 26.6

#the tail gives the last six rows and columns of the data set  
tail(swiss)

## Fertility Agriculture Examination Education Catholic  
## Neuchatel 64.4 17.6 35 32 16.92  
## Val de Ruz 77.6 37.6 15 7 4.97  
## ValdeTravers 67.6 18.7 25 7 8.65  
## V. De Geneve 35.0 1.2 37 53 42.34  
## Rive Droite 44.7 46.6 16 29 50.43  
## Rive Gauche 42.8 27.7 22 29 58.33  
## Infant.Mortality  
## Neuchatel 23.0  
## Val de Ruz 20.0  
## ValdeTravers 19.5  
## V. De Geneve 18.0  
## Rive Droite 18.2  
## Rive Gauche 19.3

#How to check for a missing value in a data set  
colSums(is.na(swiss))

## Fertility Agriculture Examination Education   
## 0 0 0 0   
## Catholic Infant.Mortality   
## 0 0

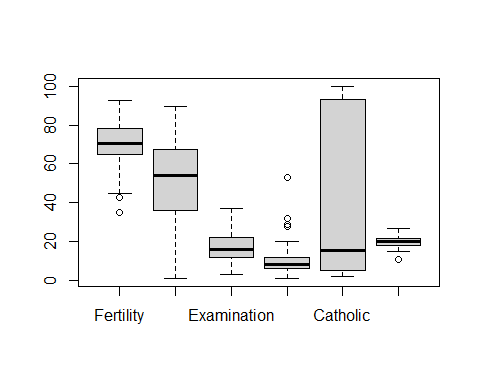
#How to display all the data  
#How to show the number of rows and columns in a data set  
nrow(swiss)

## [1] 47

ncol(swiss)

## [1] 6

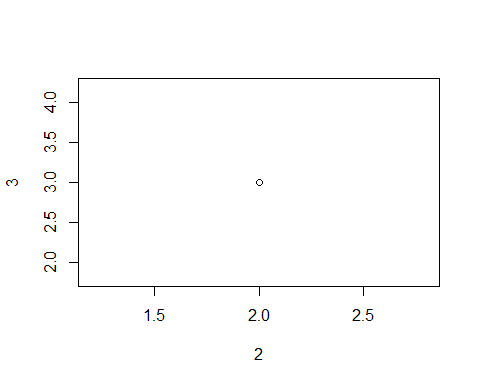
boxplot(swiss)



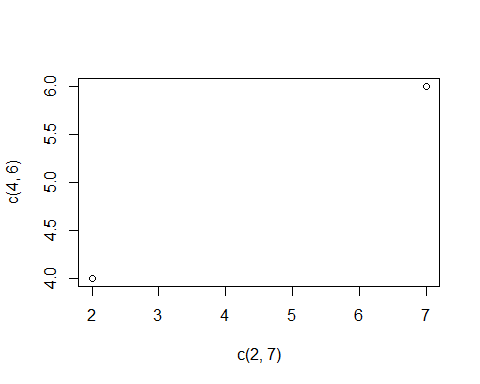
summary(swiss)

## Fertility Agriculture Examination Education   
## Min. :35.00 Min. : 1.20 Min. : 3.00 Min. : 1.00   
## 1st Qu.:64.70 1st Qu.:35.90 1st Qu.:12.00 1st Qu.: 6.00   
## Median :70.40 Median :54.10 Median :16.00 Median : 8.00   
## Mean :70.14 Mean :50.66 Mean :16.49 Mean :10.98   
## 3rd Qu.:78.45 3rd Qu.:67.65 3rd Qu.:22.00 3rd Qu.:12.00   
## Max. :92.50 Max. :89.70 Max. :37.00 Max. :53.00   
## Catholic Infant.Mortality  
## Min. : 2.150 Min. :10.80   
## 1st Qu.: 5.195 1st Qu.:18.15   
## Median : 15.140 Median :20.00   
## Mean : 41.144 Mean :19.94   
## 3rd Qu.: 93.125 3rd Qu.:21.70   
## Max. :100.000 Max. :26.60

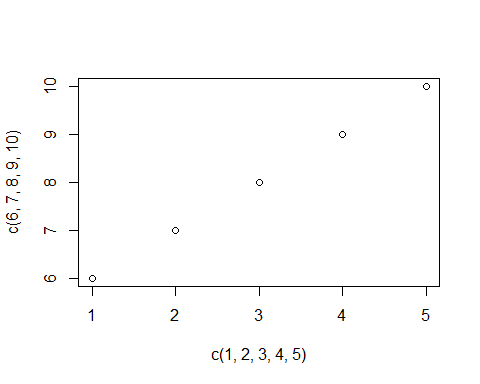
#This function describes the type of data in the data sets as they can be numeric or string in nature  
# From the summary we can comment that there is know big difference between the mean and the median.  
# From the summary we can comment that there is a great difference between the standard deviation and the maximum value.  
# From the summary we can comment that the standard deviation is almost twice the minimum value.  
  
#DATA VISUALIZATION IN R  
#plot() is used to plot points in R  
plot(2,3)



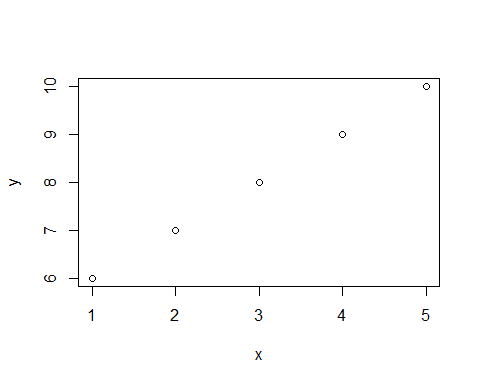
plot(c(2,7),c(4,6))



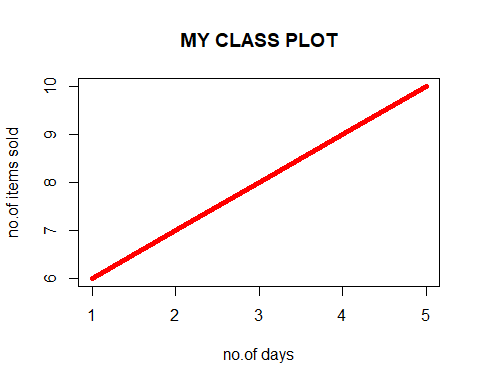
plot(c(1,2,3,4,5),c(6,7,8,9,10))



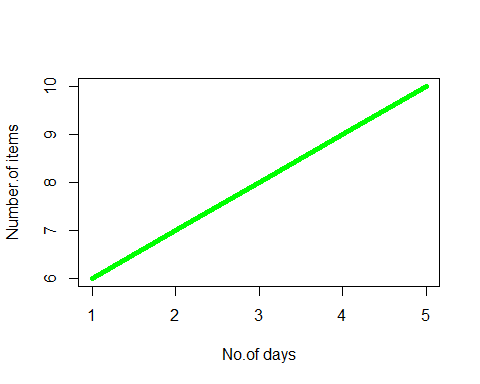
x <- c(1,2,3,4,5)  
y <- c(6,7,8,9,10)  
plot(x,y)



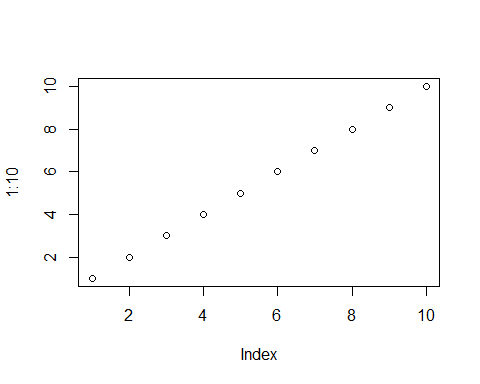
plot(x,y, col="Red",xlab = "no.of days",ylab = "no.of items sold",main="MY CLASS PLOT",type ="l",lwd=5)



plot(x,y,type = "l",col="Green",lwd=5, ylab="Number.of items", xlab="No.of days")



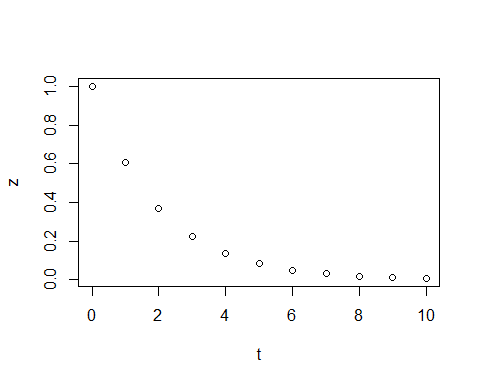
#Plotting of the sequential points  
plot(1:10)



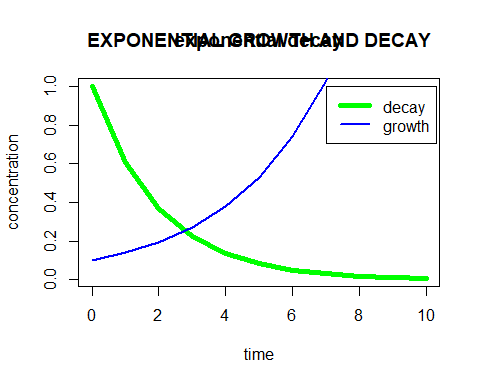
#Let t be a variable for time and z be a variable for the exponential  
t<- 0:10  
t

## [1] 0 1 2 3 4 5 6 7 8 9 10

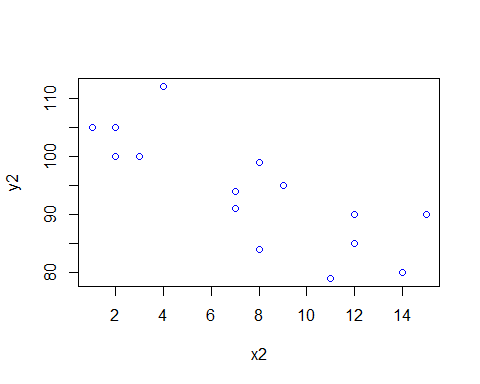
z<-exp(-t/2)  
plot(t,z)



plot(t,z,type="l",col="green", lwd=5,xlab="time",ylab="concentration",main="exponential decay")  
#line plot with multiple series  
#w is increasing with time  
w=0.1\*exp(t/3)  
lines(t,w,col="blue",lwd=2)  
title("EXPONENTIAL GROWTH AND DECAY")  
legend(7,1,c("decay","growth"),lwd=c(5,2),col=c("green","blue"),)



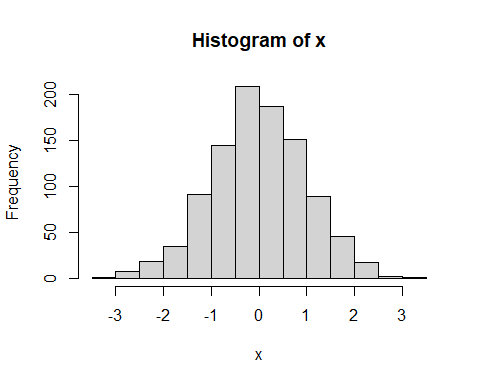
#Scatter plots   
x2<-c(2,2,8,1,15,8,12,9,7,3,11,4,7,14,12)  
y2<-c(100,105,84,105,90,99,90,95,94,100,79,112,91,80,85)  
plot(x2,y2,col="Blue")



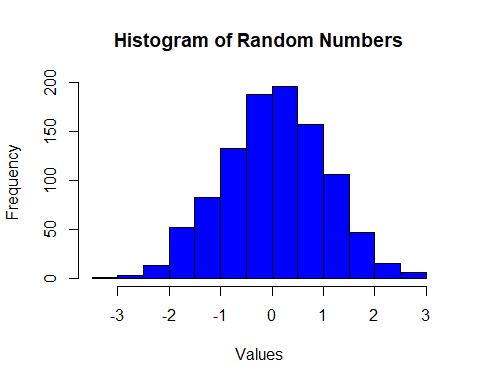
#Carrying out the summation using R  
# Create a data frame  
df <- data.frame(name = c("Alice", "Bob", "Charlie", "David"),  
 age = c(23, 31, 27, 29))  
  
# Find the summation of the age column in the data frame  
summation <- sum(df$age)  
  
# Print the summation  
print(summation)

## [1] 110

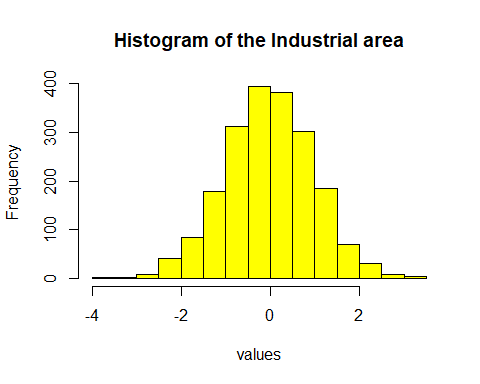
df <- data.frame(name ="Janes","Waviti")  
  
#Plotting of the histogram in R  
# Create a vector of random numbers  
x <- rnorm(1000)  
  
# Plot a histogram of the vector  
hist(x)



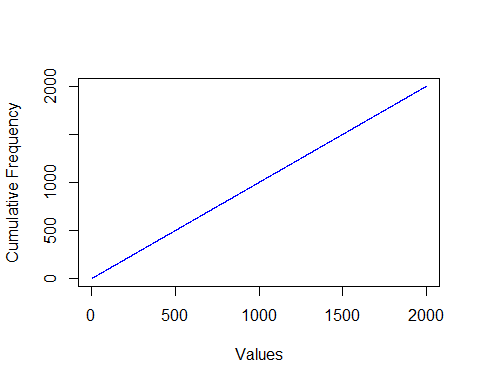
#Histogram with customizations  
# Create a vector of random numbers  
x <- rnorm(1000)  
  
# Plot a histogram of the vector with customizations  
hist(x, breaks = 20, col = "blue", main = "Histogram of Random Numbers",  
 xlab = "Values", ylab = "Frequency")



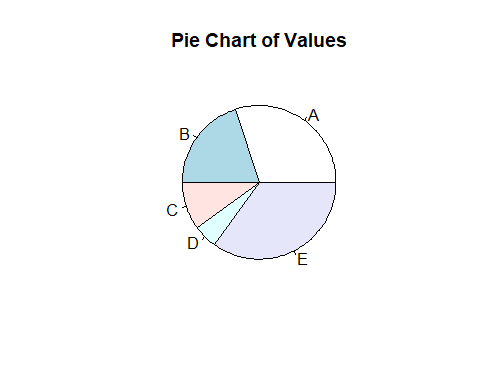
#Plotting of the friquency polygon in R  
# Create a vector of random numbers  
x <- rnorm(1000)  
  
#Creation of the second histogram  
x <- rnorm(2000)  
#Plotting of the selected points  
hist(x,breaks = 20, col = "yellow", main = "Histogram of the Industrial area",xlab = "values",ylab = "Frequency")



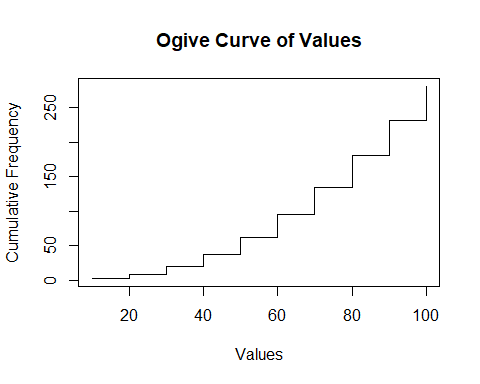
# Create a frequency table of the vector  
freq\_table <- table(x)  
  
# Calculate the cumulative frequency of the frequency table  
cum\_freq <- cumsum(freq\_table)  
  
# Plot the frequency polygon  
plot(cum\_freq, col = "blue", type = "l", xlab = "Values", ylab = "Cumulative Frequency")



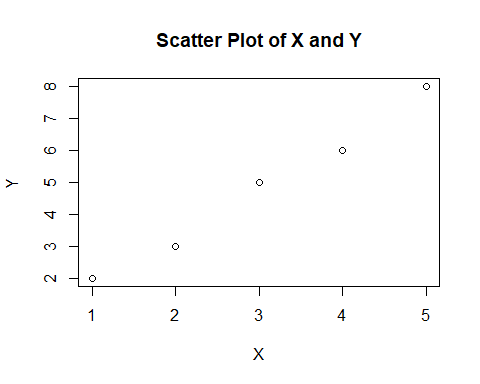
#Plotting of the piechart  
# Create a vector of values  
values <- c(30, 20, 10, 5, 35)  
  
# Create a vector of labels  
labels <- c("A", "B", "C", "D", "E")  
  
# Plot a pie chart of the values with labels  
pie(values, labels = labels, main = "Pie Chart of Values")



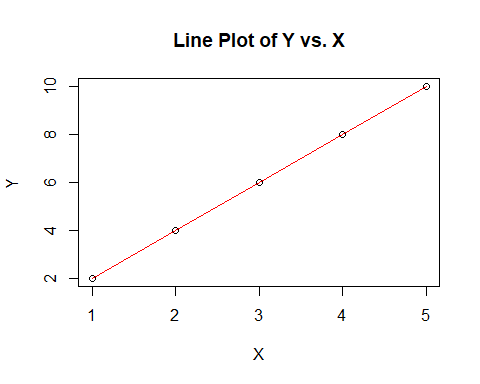
#Plotting of the ogive curve in R  
# Create a vector of values  
values <- c(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)  
  
# Create a vector of frequencies  
freq <- c(3, 6, 11, 17, 25, 33, 40, 46, 50, 50)  
  
# Calculate the cumulative frequencies  
cum\_freq <- cumsum(freq)  
  
# Plot the ogive curve  
plot(values, cum\_freq, type = "s", xlab = "Values", ylab = "Cumulative Frequency",  
 main = "Ogive Curve of Values")



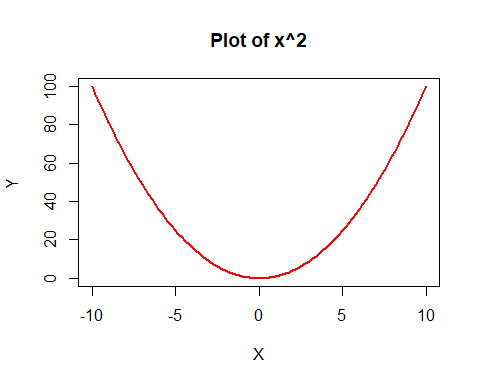
#Plotting of the simple scatter plot in R  
# Create two vectors of data  
x <- c(1, 2, 3, 4, 5)  
y <- c(2, 3, 5, 6, 8)  
  
# Plot a scatter plot of the data  
plot(x, y, main = "Scatter Plot of X and Y", xlab = "X", ylab = "Y")



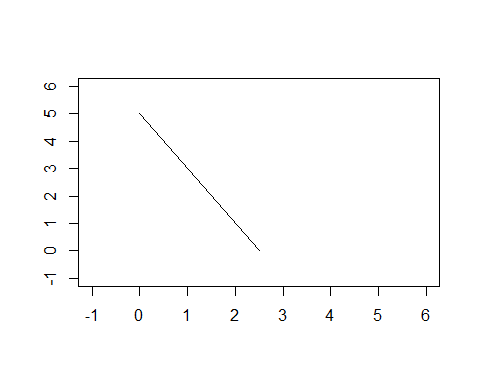
#Plotting of the line in R  
# Create a vector of x values  
x <- c(1, 2, 3, 4, 5)  
  
# Create a vector of y values  
y <- c(2, 4, 6, 8, 10)  
  
# Plot the points  
plot(x, y, type = "p", main = "Line Plot of Y vs. X", xlab = "X", ylab = "Y")  
  
# Add a line  
lines(x, y, type = "l", col = "red")



#Plotting of the curve in R  
# Define the function to plot  
myfunc <- function(x) {  
 return(x^2)  
}  
  
# Plot the curve  
plot(myfunc, xlim = c(-10, 10), ylim = c(0, 100), main = "Plot of x^2", xlab = "X", ylab = "Y")  
curve(myfunc, add = TRUE, col = "red", lwd = 2)



#Plotting of the triangle in R  
# Create a matrix with the coordinates of the vertices of the triangle  
coords <- matrix(c(0, 5, 2.5, 0, 0, 5), ncol = 2, byrow = TRUE)  
  
# Plot the triangle  
  
# Create a matrix with the coordinates of the vertices of the triangle  
coords <- matrix(c(0, 5, 2.5, 0, 0, 5), ncol = 2, byrow = TRUE)  
  
# Plot the triangle  
plot(coords, type = "n", xlim = c(-1, 6), ylim = c(-1, 6), xlab = "", ylab = "")  
polygon(coords, col = "blue")



#Plotting of the rectangle in R  
# Plot a rectangle with the lower left corner at (1, 1) and the upper right corner at (4, 3)  
plot(c(0, 5), c(0, 5), type = "n", xlab = "", ylab = "")  
rect(1, 1, 4, 3, col = "blue")  
  
#Finding the summation of the data  
library(epicalc)

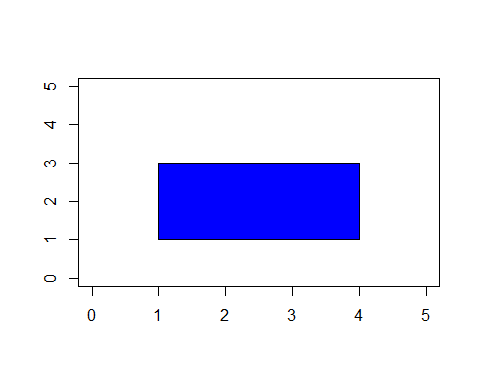
## Warning: package 'epicalc' was built under R version 4.2.2

## Loading required package: foreign

## Loading required package: survival

## Loading required package: MASS

## Loading required package: nnet



data("mtcars")  
use(mtcars)  
summ(mtcars)

##   
## No. of observations = 32  
##   
##   
## Var. name obs. mean median s.d. min. max.   
## 1 mpg 32 20.09 19.2 6.03 10.4 33.9   
## 2 cyl 32 6.19 6 1.79 4 8   
## 3 disp 32 230.72 196.3 123.94 71.1 472   
## 4 hp 32 146.69 123 68.56 52 335   
## 5 drat 32 3.6 3.7 0.53 2.76 4.93   
## 6 wt 32 3.22 3.33 0.98 1.51 5.42   
## 7 qsec 32 17.85 17.71 1.79 14.5 22.9   
## 8 vs 32 0.44 0 0.5 0 1   
## 9 am 32 0.41 0 0.5 0 1   
## 10 gear 32 3.69 4 0.74 3 5   
## 11 carb 32 2.81 2 1.62 1 8