**Practical Journal**

**QUANTITATIVE TECHNIQUES**

**Submitted By**

Name: Abdul Khalique Mistry  
SAP ID: 53003230005

Class: S.Y. B.Sc. IT.

Division: B

**Department of Information Technology**

**SVKM’s Usha Pravin Gandhi College of Arts,**

**Science & Commerce**

**(AUTONOMOUS)**

**NAAC Accredited ‘A+’ Grade with CGPA 3.27   
(Affiliated to University of Mumbai)**

**Vile Parle (West), Mumbai 400056**

2024-25

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR.No** | **Topic** | **Date** | **Page.no** | **Signature** |
| **1** | **Introduction to R** | **10/12/24** | **2** |  |
| **2** | **R Programming Structure** | **17/12/24** | **10** |  |
| **3** | **R-Charts and Graphs** | **07/01/25** | **16** |  |
| **4** | **Going Statistics in R** | **04/02/25** | **22** |  |
| **5** | **Discrete Mathematics and R** | **11/02/25** | **31** |  |

|  |  |  |
| --- | --- | --- |
| image1.jpeg | **Shri Vile Parle Kelavani Mandal’s**  **Usha Pravin Gandhi College of Arts, Science & Commerce**  **Vile Parle (West), Mumbai 400 056**  **(AUTONOMOUS)**  **NAAC Accredited ‘A+’ Grade with CGPA 3.27** | image2.jpeg |

**Class: S.Y.B. Sc.IT**

Roll No\_\_\_\_\_\_\_\_\_

**SAP ID: \_\_\_53003220126\_\_\_\_**

**CERTIFICATE**

This certificate affirms that the experimental work detailed in this Journal conforms with the Bachelor of Science in Information Technology program for the course **"QUANTITATIVE TECHNIQUES "** as prescribed by syllabus. The experiments were conducted in the Computer Laboratory of Usha Pravin Gandhi College of Arts, Science & Commerce.

Student Miss/Master: **ABDUL KHALIQUE MISTRY** of Class S.Y.B.S.C.-IT, Division 'B', has successfully completed \_\_\_ out of \_\_\_ experiments during the Academic Year 2024-2025.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Subject In-Charge:

Date:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Co-ordinator Head of Department

Dept. of Information Technology Dept. of Information Technology

Chap-1

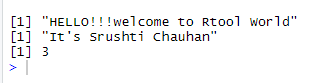
Code:

print("HELLO!!!welcome to Rtool World")

print("It's Srushti Chauhan")

print(2+1)

Output



Datatype

Code:

v <- TRUE

print(class(v))

s <- -23.5

print(class(s))

d <- -2L

print(class(d))

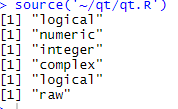
z <- 2+3i

print(class(z))

r <- TRUE

print(class(r))

i<- charToRaw("HELOOO")

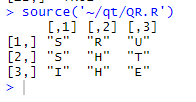
Output:

Code:

M=matrix(c('S','R','U','S','H','T','I','H','E'),nrow=3,ncol=3,byrow=TRUE)

print(M)

Output:



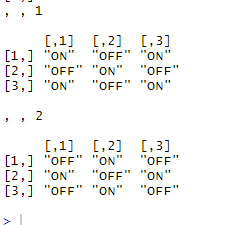
#CREATE AN ARRAY

Code:

A <- array(c('ON','OFF'),dim=c(3,3,2))

print(A)

Output:



#create a vector

Code:

permission<-c('Read','Write','Execute');

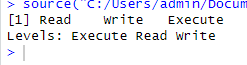
#create a factor

f\_permission<-factor(permission)

#print the factor

print(f\_permission)

Output:



#create data frame

Code

stud<-data.frame(

sno=c(101,102,103,104),

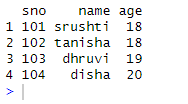
name=c("srushti","tanisha","dhruvi","disha"),

age=c(18,18,19,20)

);

print(stud)

Output:



#variable assignment

Code

#assignment using equal operator

v1=c(2,3,6,8)

#assignment using rightward opertor

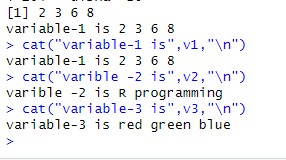
c('R','programming')->v2

#assignment using leftward operator

v3<-c("red","green","blue")

print(v1)

Output:

image9.png

#Data types of a variable

Code:

v1<-c(2,4,6,7,10)

cat("class of variable-1 is",class(v1),"\n")

v3<-c("red","green","blue")

cat("class of variable-3 is",class(v3),"\n")

Output:

image11.png

#finding variable

Code:

print(ls())

Output:

image12.png

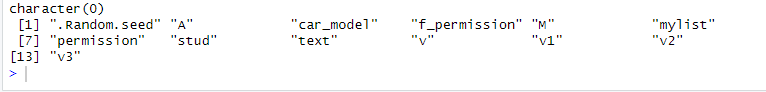
#list all variable with given pattern

Code:

print(ls(pattern="color"))

print(ls(all.name=TRUE))

Ouput:



#DELETING VARIABLE

Code:

rm(v1)

print(v1)

image14.png

Operators

a<-c(1,2,3)

> b<-c(4,5,6)

> print(a+b)

[1] 5 7 9

> s<-c(1,2)

> d<-c(2,3)

> print(s\*d)

[1] 2 6

>

|  |
| --- |
| w<-c(1,4,8)  > h<-c(15,44,57)  > print(w/h)  [1] 0.06666667 0.09090909 0.14035088 |
|  |
|  |

a<-c(1,4,6)

> b<-c(13,44,57)

> print(b-a)

[1] 12 40 51

>

a<-c(1,4,5)

> b<-c(13,44,57)

> print(a%%b)

[1] 1 4 5

> a<-c(1,2,3)

> b<-c(4,5,6)

> print(a%/%b)

[1] 0 0 0

> a<-c(1,2,3)

> b<-c(4,5,6)

> print(a^b)

[1] 1 32 729

> a<-c(1,2,3)

> b<-c(4,5,6)

> print(a<b)

[1] TRUE TRUETRUE

a<-c(1,2,3)

> b<-c(4,5,6)

> print(a>b)

[1] FALSE FALSEFALSE

> a<-c(1,2,3)

> b<-c(4,5,6)

> print(a==b)

[1] FALSE FALSEFALSE

> a<-c(1,2,3)

> b<-c(4,5,6)

> print(a<=b)

[1] TRUE TRUETRUE

> a<-c(1,2,3)

> b<-c(4,5,6)

> print(a>=b)

[1] FALSE FALSEFALSE

a<-c(1,2,3)

> b<-c(4,5,6)

> print(a!=b)

[1] TRUE TRUETRUE

> a<-c(12,34,TRUE,2+3i)

> b<-c(65,4,FALSE,2+3i)

> print(a&b)

[1] TRUETRUE FALSE TRUE

> a<-c(12,34,TRUE,2+3i)

> b<-c(65,4,FALSE,2+3i)

> print(a|b)

[1] TRUE TRUETRUETRUE

> a<-c(12,34,TRUE,2+3i)

> b<-c(65,4,FALSE,2+3i)

> print(!b)

[1] FALSE FALSE TRUE FALSE

> a<-3:6

> print(a)

[1] 3 4 5 6

> a<-5

> b<-10

> t<-1:5

>print(a %in% t)

[1] TRUE

>print(b %in% t)

[1] FALSE

|  |
| --- |
| > #create a sequence of number from 1 to 10  > print(seq(1,10))  [1] 1 2 3 4 5 6 7 8 9 10  >  > #mean of number from 24 to 145  > print(mean(24:145))  [1] 84.5  >  > #sum of numbers from 25 to 50  > print(sum(25:50))  [1] 975 |
|  |
|  |

Chap-2

STRING

Code:

a<-'start and end with single quote'

print(a)

b<-"start and end with double quotes"

print(b)

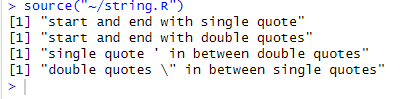
c<-"single quote ' in between double quotes"

print(c)

d<- 'double quotes " in between single quotes'

print(d)

Ouput:



Code:

X<-"WELCOME"

Y<-"R PROGRAMMIMG";

Z<-"CHAPTER 1"

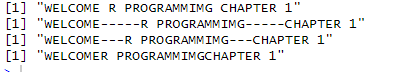
print(paste(X,Y,Z))

print(paste(X,Y,Z,sep = "-----"))

print(paste(X,Y,Z,sep="---",collapse = ""))

print(paste(X,Y,Z,sep = "",collapse = ""))

Ouput:



Code:

#number padding with blank space at beginning

a<-format(23.5,width=5)

print(a)

#justify string with center

a<-format("R-book",width=6,justify="c")

print(a)

#justify string at left

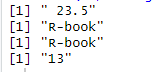
a<-format("R-book",width=6,justify="l")

print(a)

#Format as string.

a<-format(13)

print(a)

Ouput:

Code:

result<-nchar("count the number of characters")

image18.pngprint(result)

#convert into upper case

str<-toupper("R programming")

print(str)

#convert into lower case

str<-tolower("R progarmmming")

image19.pngprint(str)

#demo for substring function

str<-substring("R progarmming",1,9)

print(str)

image20.pngOuput:

SINGLE element vector

#vector of type

Code:

print(23.5)

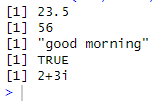
print(56L)

print("good morning")

print(TRUE)

print(2+3i)

Ouput:



Multiple element vector

#create sequence from 5 to 15

Code:

v<-5:15

print(v)

#create sequence from 2.2 to 10.2

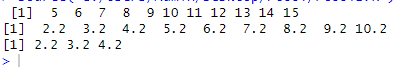
s<- 2.2:10.2

print(s)

l<-2.2:5.10

print(l)

Ouput:



vector manipulation

Code:

x<-c(3,4,6,8)

y<-c(1,3,5,7)

add<-x+y

print(add)

sub<-x-y

print(sub)

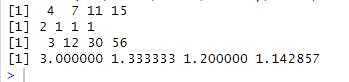
mul<-x\*y

print(mul)

div<-x/y

print(div)

Output:



#vector element recycling

Code:

x<-c(4,6,8,2)

y<-c(1,5)

add.result<-x+y

print(add.result)

sub.result<-x-y

print(sub.result)

Ouput:

image24.png

image25.png

Code:

x<-c(12,45,2,89,567,4,33,23,11)

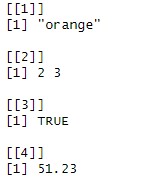
s<-sort(x)

print(s)

revsort<-sort(x,decreasing = TRUE)

print(revsort)

Output:



#List

Code:

#create a list

l<-list("orange",c(2:3),TRUE,51.23)

print(l)

#naming list element

l<-list(c("orange","apple"),matrix(c(3,5,2,-2,-9,-2,8), nrow=3),

list("monday",24))

names(l)<-c("friits","matrix","list element")

print(l)

#manipulating list

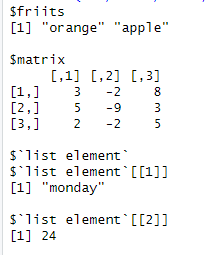
l<-list(c("orange","apple"),matrix(c(3,5,2,-2,-9,-2,8), nrow=3),

list("monday",24))

names(l)<-c("friits","matrix","list element")

print(l)

Output:



Code:

l[1]<-"new element"

print(l[1])

#remove the last element

l[4]<-NULL

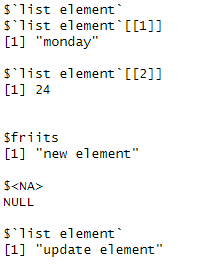
print(l[4])

#update the 3rd element

l[3]<-"update element"

print(l[3])

Output:



#merging list

Code:

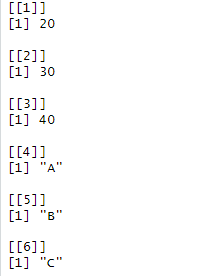
x1<-list(20,30,40)

x2<-list("A","B","C")

m\_list<-c(x1,x2)

print(m\_list)

Ouput:



#converting list to vector

Code:

x1<-list(10:15)

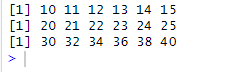
x2<-list(20:25)

v1<-unlist(x1)

v2<-unlist(x2)

print(v1)

Output:



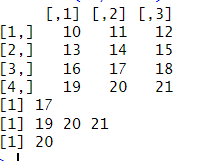
Code:

print(v2)

res<-v1+v2

print(res)

Output:



Matrix

Code:

#accessing element of a matrix

p<- matrix(c(10:21),nrow=4,byrow=TRUE)

print(p)

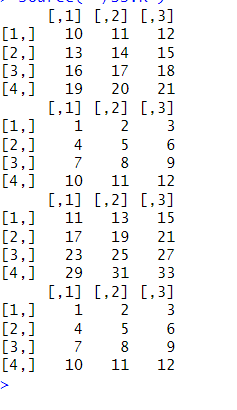
#access the element at 3rd row and 2nd column

print(p[3,2])

#access the element zt 4th row

print(p[4,])

Output:



#access the element at 4th row and 2nd column

Code:

print(p[4,2])

#matrix addition and subtraction

P<-matrix(c(10:21),nrow=4,byrow=TRUE)

Q<-matrix(c(1:12),nrow=4,byrow=TRUE)

print(P)

print(Q)

#addition

add=P+Q

print(add)

#subtraction

sub=P-Q

print(Q)

#matrix MULTIPLICATION and DIVISION

P<-matrix(c(10:21),nrow=4,byrow=TRUE)

Q<-matrix(c(1:12),nrow=4,byrow=TRUE)

print(P)

print(Q)

#MULTI

add=P\*Q

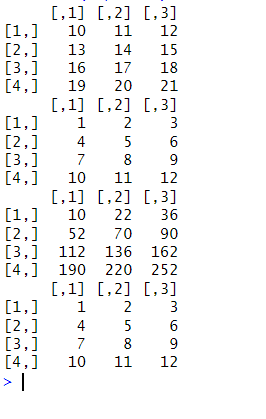
print(add)

#DIV

sub=P/Q

print(Q)

Output:



Array

#CREATE ARRAY

Code:

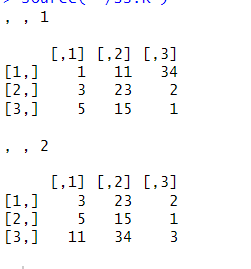
v1<-c(1,3,5)

v2<-c(11,23,15,34,2)

v<-array(c(v1,v2),dim=c(3,3,2))

print(v)

Output:



#ACCESSING ARRAY ELEMENT

Code:

v1<-c(1,3,5)

v2<-c(11,27,15,24,2)

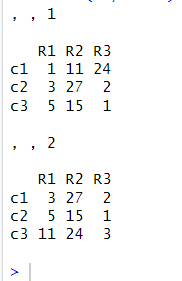
column\_name<-c("c1","c2","c3")

row\_name<-c("R1","R2","R3")

v<-array(c(v1,v2),dim=c(3,3,2),dimnames = list(column\_name,row\_name))

print(v)

Output:



#MANIPULATING ARRAY ELEMENT

Code:

v1<-c(1,3,5)

v2<-c(11,23,15,34,2)

v<-array(c(v1,v2),dim=c(3,3,2))

v3<-c(3,5,7)

v4<-c(34,5,2,7,4)

u<-array(c(v3,v4),dim=c(3,3,2))

#create matrix of above arrays u and v

m1<-u[,,2]

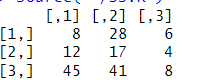
m2<-v[,,2]

#addition of m1 and m2

result<-m1+m2

print(result)

Output:



Code:

#R-Factors

color<-c("red","blue","pink")

print(color)

print(is.factor(color))

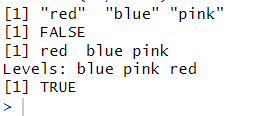
#apply the factor function

f\_data<-factor(color)

print(f\_data)

print(is.factor(f\_data))

Output:



#factor in data frame

Code:

color\_code<-c(101,102,103,104)

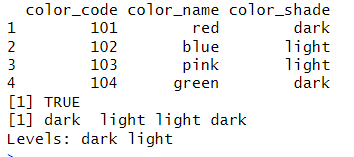
color\_name<-c("red","blue","pink","green")

color\_shade<-c("dark","light","light","dark")

d1<-data.frame(color\_code,color\_name,color\_shade)

print(d1)

Output:



#test if color shade is factor

Code:

print(is.factor(d1$color\_shade))

print(d1$color\_shade)

#R-DATA FRAMES

stud<-data.frame(

sno=c(101,102,103,104,105),

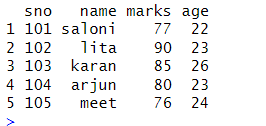
name=c("saloni","lita","karan","arjun","meet"),

marks=c(77,90,85,80,76),

age=c(22,23,26,23,24));

print(stud)

Output:



Code:

#summary od data frames

stud<-data.frame(

sno=c(101,102,103,104,105),

name=c("saloni","lita","karan","arjun","meet"),

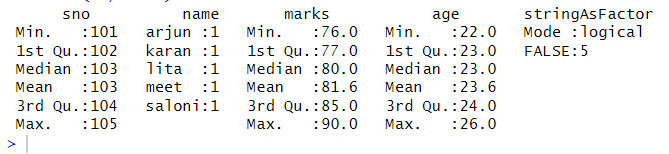
marks=c(77,90,85,80,76),

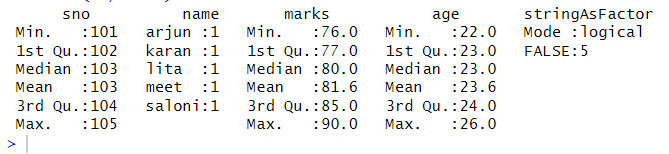
age=c(22,23,26,23,24),

stringAsFactor=FALSE);

print(summary(stud))

Output:



#extract data from data frame

Code:

stud<-data.frame(

sno=c(101,102,103,104,105),

name=c("saloni","lita","karan","arjun","meet"),

marks=c(77,90,85,80,76),

age=c(22,23,26,23,24),

stringAsFactor=FALSE);

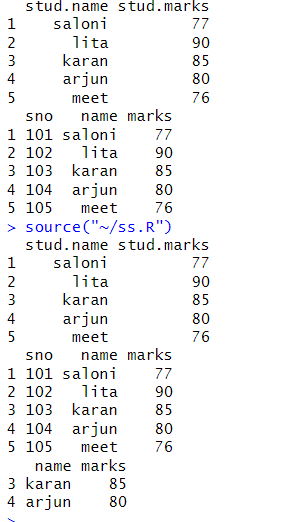
a<-data.frame(stud$name,stud$marks)

print(a)

s<-stud[1:3]

print(s)

Output:



#extract 3rd and 4th rows with 2nd and 3rs column

Code:

l<-stud[c(3,4),c(2,3)]

print(l)

#expand data frame

stud<-data.frame(

sno=c(101,102,103,104,105),

name=c("saloni","lita","karan","arjun","meet"),

marks=c(77,90,85,80,76),

age=c(22,23,26,23,24));

stud\_new<-data.frame(

sno=c(106:107),

name=c("saloni","lita"),

marks=c(77,90),

age=c(22,23),

dept=c("finance","HR"));

#add new column

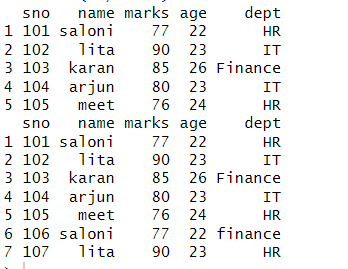
stud$dept<-c("HR","IT","Finance","IT","HR")

a<-stud

print(a)

#add row

stud\_final<-rbind(stud,stud\_new)

print(stud\_final)

Output:

CHP-5 CHARTS AND GRAPH

R-Pie Charts

Code:

x<-c(32,56,78,66)

y<-c("oracle","data science","r-tool","DBA")

#give name to file

png(file="Course.jpg")

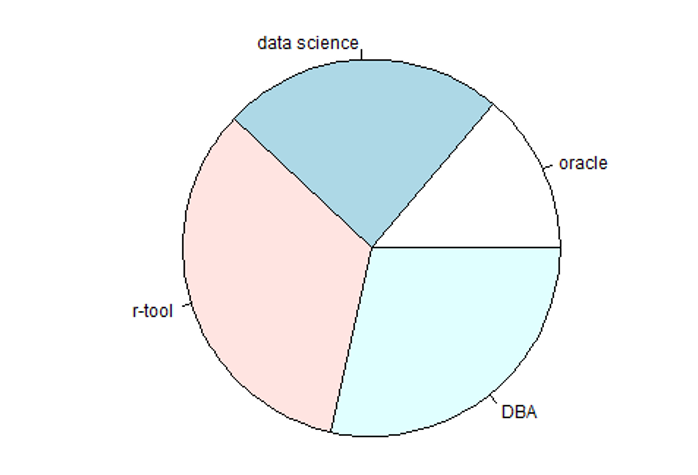
#plot the pie chart

pie(x,y)

dev.off()

library(plotrix)

Output:



3D pie chart

Code:

#create data

slices<-c(10,12,4,16,9)

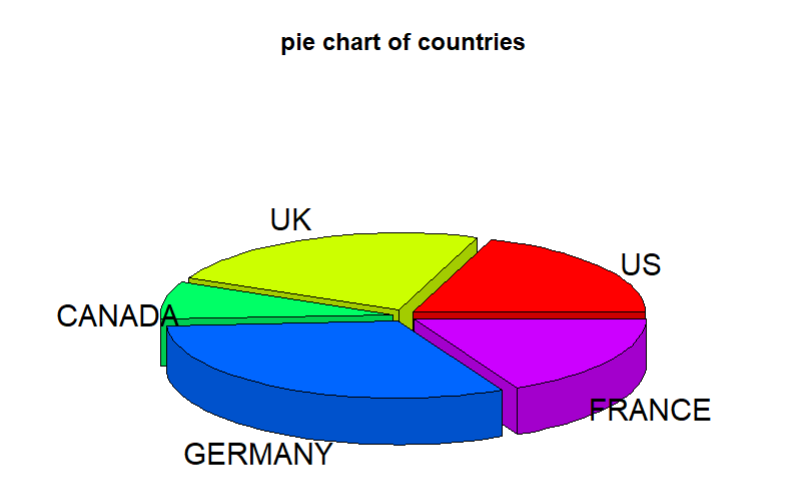
lbls<-c("US","UK","CANADA","GERMANY","FRANCE")

#plot chart

pie3D(slices,labels=lbls,explode=0.05,

main="pie chart of countries")

Output:



R-Bar charts

Code:

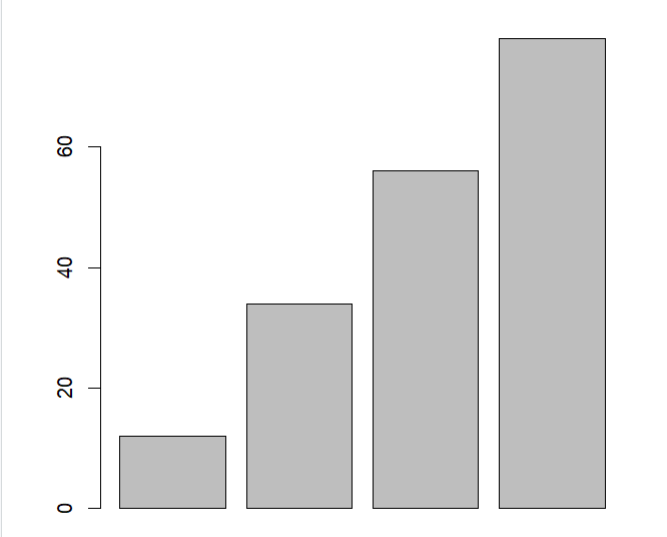
h<-c(12,34,56,78)

png(file="barchart.jpg")

barplot(h)

dev.off()

Output:



#group bar chart and stacked bar chart

Code:

f <- c("orange", "green") # Colors

m <- c("Dec", "Jan") # Months (2 months, matching the columns in `values`)

# Creating the values matrix (3 rows, 2 columns)

values <- matrix(c(2, 4, 5, 6, 2, 6), nrow = 3, ncol = 2, byrow = TRUE)

# Creating the stacked bar chart and saving it as a .jpg file

png(file = "barchart\_stack.jpg")

barplot(values,

main = "Total Revenue by Month",

names.arg = m, # Ensure the number of months matches the number of columns

beside = FALSE, # This ensures the bars are stacked

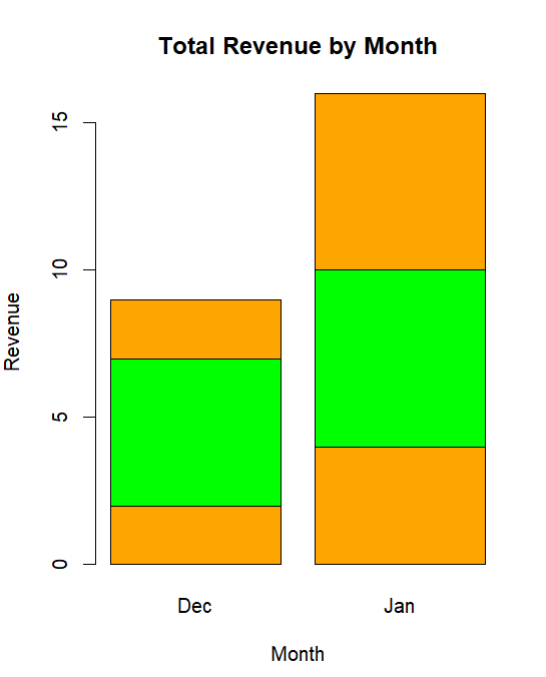
xlab = "Month",

ylab = "Revenue",

col = f) # Colors for the bars

dev.off() # Save the plot and close the graphics device

Output:



R-Boxplots

# Selecting 'mpg' and 'cyl' columns from the 'mtcars' dataset

Code:

lp <- mtcars[, c('mpg', 'cyl')]

# Saving the boxplot as a .jpg file

png(file = "boxplot.jpg")

# Creating the boxplot

boxplot(mpg ~ cyl, data = lp,

xlab = "Number of Cylinders",

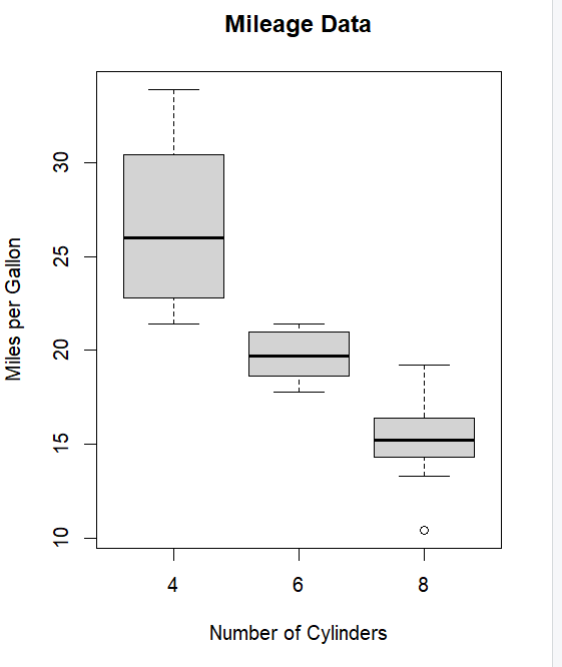
ylab = "Miles per Gallon",

main = "Mileage Data")

# Closing the device to save the plot

dev.off()

Output:



Code:

#R-Boxplots with notch

# Selecting 'mpg' and 'cyl' columns from the 'mtcars' dataset

lp <- mtcars[, c('mpg', 'cyl')]

# Saving the boxplot as a .jpg file

png(file = "boxplot.jpg")

# Creating the boxplot with notches, variable width, and custom colors

boxplot(mpg ~ cyl, data = mtcars,

xlab = "Number of Cylinders",

ylab = "Miles per Gallon",

main = "Mileage Data",

notch = TRUE, # Corrected typo from 'nothch' to 'notch'

varwidth = TRUE, # Corrected case to lowercase 'varwidth'

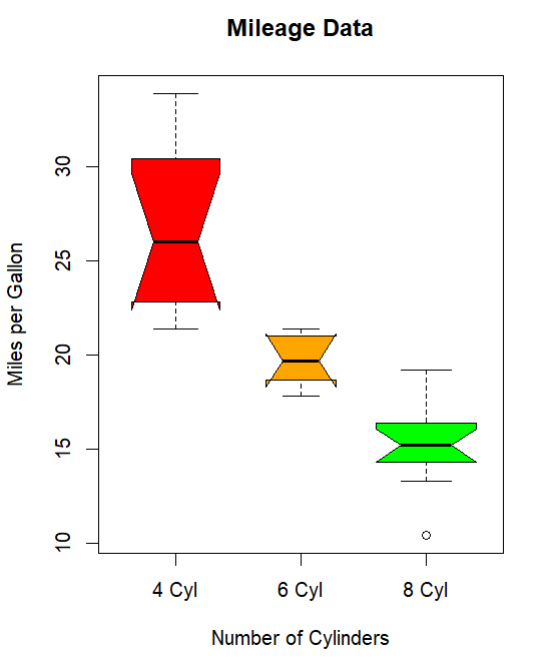
col = c("red", "orange", "green"), # Colors for each box

names = c("4 Cyl", "6 Cyl", "8 Cyl")) # Custom names for the x-axis

# Closing the device to save the plot

dev.off()

Output:



R-Histogram

Code:

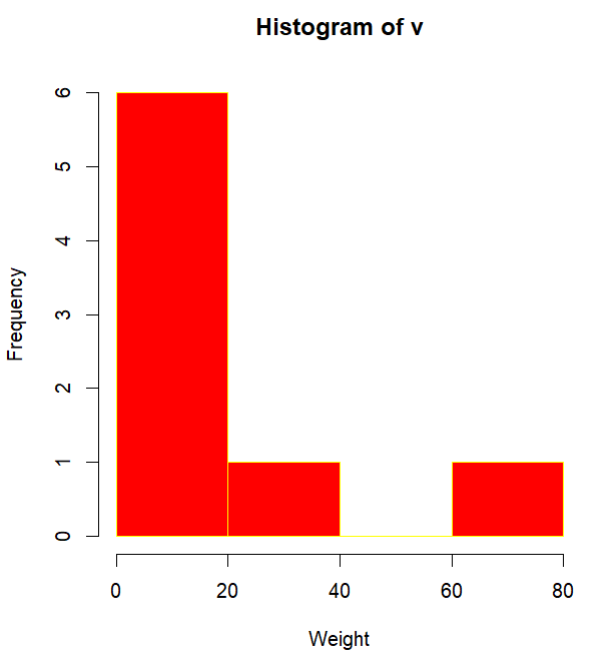
v<-c(3,4,5,23,4,5,5,66)

png(file="histogram.jpg")

hist(v,xlab="Weight",col="red",border="yellow")

dev.off()

Output:



#R-Line Graphs

Code:

# Defining the data

v <- c(10, 3, 25, 6, 22)

# Saving the line chart as a .jpg file

png(file = "Line\_chart.jpg")

# Creating the line graph

plot(v, type = "o", # 'o' means both lines and points

xlab = "Index", # Label for the x-axis

ylab = "Values", # Label for the y-axis

main = "Line Graph of Values", # Title of the graph

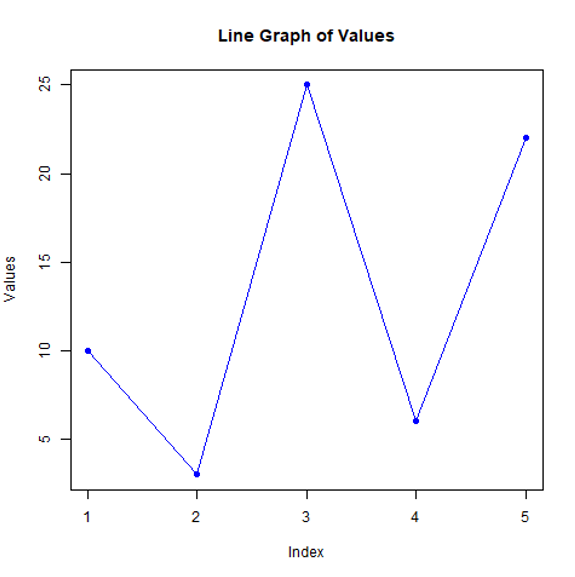
col = "blue", # Line color

pch = 16) # Point style

# Closing the device to save the plot

dev.off()

Output:



# multiple lines in line chart

Code:

v <- c(10, 3, 25, 6, 22)

t<-c(14,7,6,19,4)

# Saving the line chart as a .jpg file

png(file = "Line\_chart\_2.jpg")

# Creating the line graph

plot(v, type = "o", # 'o' means both lines and points

xlab = "Month", # Label for the x-axis

ylab = "Rain fail", # Label for the y-axis

main = "Rain fall chart", # Title of the graph

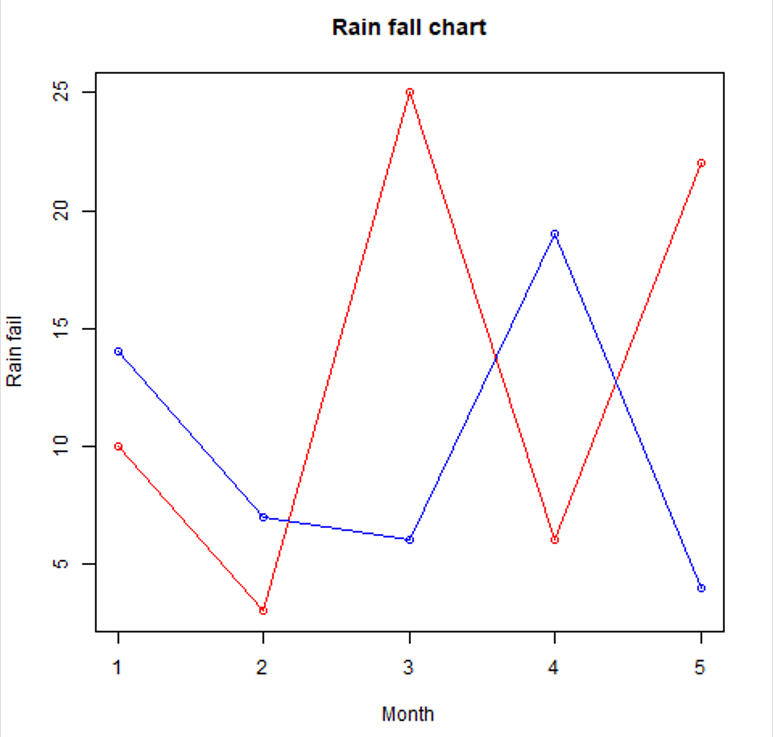
col = "red", ) # Line color

lines(t,type="o",col="blue")

# Closing the device to save the plot

dev.off()

Output:



#R-Scatterplots

# Selecting 'wt' (weight) and 'hp' (horsepower) columns from the 'mtcars' dataset

Code:

ip <- mtcars[, c("wt", "hp")]

# Saving the scatterplot as a .jpg file

png(file = "scatter.jpg")

# Creating the scatterplot

plot(x = ip$wt, y = ip$hp,

xlab = "Weight", # Label for the x-axis

ylab = "HP", # Label for the y-axis

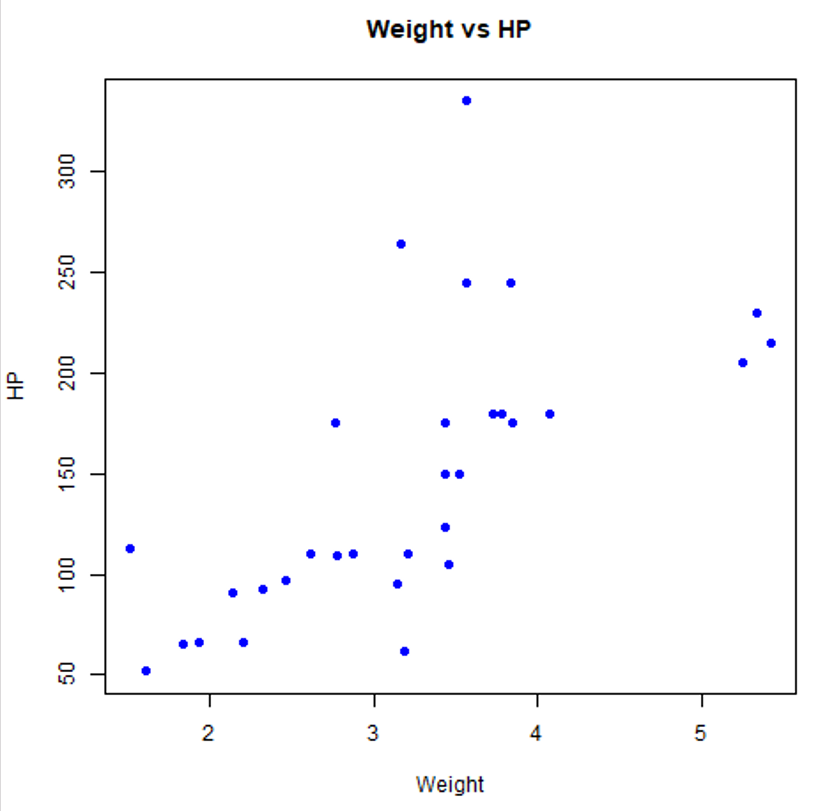
main = "Weight vs HP", # Title of the graph

col = "blue") # Color of the points

# Closing the device to save the plot

dev.off()

Output:



#R-scatter matrix

Code:

ip <- mtcars[, c("wt", "mpg", "cyl")]

# Saving the scatter plot matrix as a .jpg file

png(file = "scattermatrix.jpg")

# Creating the scatter plot matrix

pairs(ip,

main = "Scatter Matrix of Weight, MPG, and Cylinders") # Title of the graph

# Closing the device to save the plot

dev.off()

Output:

