DESCRIPTIVE STATISTICS

~Mousumi Swain

CS-22-045

**PRACTICAL NO. 1**

> #Data Input & Arithmetic Operation

>

> x <- 10

> y <- 5

>

> #Addition

> x+y

[1] 15

> #Subtraction

> x-y

[1] 5

> #Multiplication

> x\*y

[1] 50

> #Division

> x/y

[1] 2

> #Exponent

> x^2

[1] 100

> #Modulus

> x%%y

[1] 0

> #Integer Division

> x%/%y

[1] 2

> #Logical Operators

> x<y

[1] FALSE

> x>y

[1] TRUE

> x==y

[1] FALSE

>

> #Vector of Strings

> fruit <- c("Mango","Apple","Orange","Grapes")

> print(fruit)

[1] "Mango" "Apple" "Orange" "Grapes"

> #Vector of Numbers

> num <- c(9,6,5,3,4,4,3,5,9,8)

> print(num)

[1] 9 6 5 3 4 4 3 5 9 8

> #Vector of Sequence Number

> num <- 2:9

> print(num)

[1] 2 3 4 5 6 7 8 9

> #Vector of decimal sequential Number

> num1 <- 1.2:9.2

> print(num1)

[1] 1.2 2.2 3.2 4.2 5.2 6.2 7.2 8.2 9.2

> #Vector of decimal sequential number where last element is not used

> num2 <- 1.2:9.5

> print(num2)

[1] 1.2 2.2 3.2 4.2 5.2 6.2 7.2 8.2 9.2 > #Vector of Logical Values

> a <- c("True","False","True","False")

> print(a)

[1] "True" "False" "True" "False"

> #Length of Vector

> length(num)

[1] 8

> #Sorting of Vector

> sort(num)

[1] 2 3 4 5 6 7 8 9

> #Access the items

> num[4]

[1] 5

> #Acess the first and fifth items

> num[c(1,5)]

[1] 2 6

> #Acessitem in sequences from first to fifth

> num[1:5]

[1] 2 3 4 5 6

> #Acess the all element rather than forth

> num[c(-4)]

[1] 2 3 4 6 7 8 9

> #Change a item in a vector

> num[5] <- 99

> print(num)

[1] 2 3 4 5 99 7 8

9 > #Repetition of a vector

> rep(c(1,5,3,7,6),times=4)

[1] 1 5 3 7 6 1 5 3 7 6 1 5 3 7 6 1 5 3 7 6

> #Sequence vector

> seq(from=1,to=15,by=3)

[1] 1 4 7 10 13

> seq(1,15,3)

[1] 1 4 7 10 13

> seq(length=10,from=4,by=4)

[1] 4 8 12 16 20 24 28 32 36 40

>

>

> #Vector Operation

>

> a <- 1:10

> b <- 2:11

>

> a+b

[1] 3 5 7 9 11 13 15 17 19 21 > a-b

[1] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

> a\*b

[1] 2 6 12 20 30 42 56 72 90 110

> a/b

[1] 0.5000000 0.6666667 0.7500000 0.8000000 0.8333333

0.8571429

0.8750000

[8] 0.8888889 0.9000000 0.9090909

> a^4

[1] 1 16 81 256 625 1296 2401 4096 6561 10000 > b^5

[1] 32 243 1024 3125 7776 16807 32768 59049 100000

161051

> a%%b

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

> a%/%b

[1] 0 0 0 0 0 0 0 0 0 0

> #To convert negative to positive

> abs(a-b)

[1] 1 1 1 1 1 1 1 1 1 1

> #Square

> sqrt(a+b)

[1] 1.732051 2.236068 2.645751 3.000000 3.316625 3.605551

3.872983

4.123106

[9] 4.358899 4.582576

> #Round to upward number of decimal number

> ceiling(a/b)

[1] 1 1 1 1 1 1 1 1 1 1

> #Round to downward number of decimal number

> floor(a/b)

[1] 0 0 0 0 0 0 0 0 0 0

>

> #Matrix

>

> #Matrix of numeric

> a <-matrix(c(9,0,0,4,7,8,8,1,9,0),,nrow=5,ncol=4)

> print(a)

[,1] [,2] [,3] [,4]

[1,] 9 8 9 8

# [2,] 0 8 0 8

# [3,] 0 1 0 1

# [4,] 4 9 4 9

[5,] 7 0 7 0

> b <- matrix(c("Mousumi'."Manoj","Swain"),nrow=3,ncol=2)

Error: unexpected symbol in "b <- matrix(c("Mousumi'."Manoj"

> b <- matrix(c("Mousumi","Manoj","Swain"),nrow=3,ncol=2)

> print(b)

[,1] [,2]

[1,] "Mousumi" "Mousumi"

[2,] "Manoj" "Manoj"

[3,] "Swain" "Swain"

>

> #Access matrix item

>

> #Item at place

> a[1,3]

[1] 9

> #Wholw row access

> a[1,]

[1] 9 8 9 8

> #whole column access

> a[,3]

[1] 9 0 0 4 7

> #More than one row access

> a[c(1,4),]

[,1] [,2] [,3] [,4]

[1,] 9 8 9 8

[2,] 4 9 4 9

> #More than one column access

> a[,c(1,4)]

[,1] [,2]

[1,] 9 8

[2,] 0 8

[3,] 0 1

[4,] 4 9

[5,] 7 0

> #Add column after the matrix

> b <- cbind

(a,c(7,5,6,8,3)) > print(b)

[,1] [,2] [,3] [,4] [,5]

[1,] 9 8 9 8 7

[[2,] 0 8 0 8 5](#_Toc17489)

[[3,] 0 1 0 1 6](#_Toc17490)

[[4,] 4 9 4 9 8](#_Toc17491)

[5,] 7 0 7 0 3

> #Add column before the matrix

> b <- cbind(c(7,5,6,8,3),a)

> print(b)

[,1] [,2] [,3] [,4] [,5]

[1,] 7 9 8 9 8

[2,] 5 0 8 0 8

[3,] 6 0 1 0 1

[4,] 8 4 9 4 9

[5,] 3 7 0 7 0

> #Add row after the matrix

> b <- rbind(a,c(7,5,6,8))

> print(b)

[,1] [,2] [,3] [,4]

[1,] 9 8 9 8

[2,] 0 8 0 8

[3,] 0 1 0 1

[4,] 4 9 4 9

[5,] 7 0 7 0

[6,] 7 5 6 8

> #Add row before the matrix

> b <- rbind(c(7,5,6,8),a)

> print(b)

[,1] [,2] [,3] [,4]

[1,] 7 5 6 8

[2,] 9 8 9 8

[3,] 0 8 0 8

[4,] 0 1 0 1

[5,] 4 9 4 9

[6,] 7 0 7 0 > #Check an item present in a matrix or not

> "9"%in%a [1] TRUE

> "1"%in%a

[1] TRUE

> #To check dimensionof a matrix

> dim(a)

[1] 5 4

> #To check length of a matrix

> length(a)

[1] 20

>

>

> #Matrix Operation

>

> x <- c(9,6,5,3,4,4,3,5)

> a <- matrix(data=x,nrow=4,ncol=4)

> b <- matrix(data=x,nrow=4,ncol=4)

> print(a)

[,1] [,2] [,3] [,4]

[1,] 9 4 9 4

[2,] 6 4 6 4

[3,] 5 3 5 3

[4,] 3 5 3

5

> print(b)

[,1] [,2] [,3] [,4]

[1,] 9 4 9 4

[2,] 6 4 6 4

[3,] 5 3 5 6

[4,] 3 5 3 6

> a+b [,1] [,2] [,3] [,4] [1,] 18 8 18 8 [2,]

12 8 12 8 [3,] 10 6 10 **Error! Bookmark not defined.**

[4,] 6 10 6 10 > a-b

[,1] [,2] [,3] [,4]

[1,] 0 0 0 0

[2,] 0 0 0 0

[3,] 0 0 0 0

[4,] 0 0 0 0 > a\*b

[,1] [,2] [,3] [,4]

[1,] 81 16 81 16

[2,] 36 16 36 16

[3,] 25 9 25 9

[4,] 9 25 9 25 > a/b

[,1] [,2] [,3] [,4]

[1,] 1 1 1 1

[2,] 1 1 1 1

[3,] 1 1 1 1

[4,] 1 1 1 1 > a^4

[,1] [,2] [,3] [,4]

[1,] 6561 256 6561 256

[2,] 1296 256 1296 256

[3,] 625 81 625 81

[4,] 81 625 81 625

> b^5

[,1] [,2] [,3] [,4]

[1,] 59049 1024 59049 1024

[2,] 7776 1024 7776 1024

[3,] 3125 243 3125 243

[4,] 243 3125 243 3125

> a%%b

[,1] [,2] [,3] [,4]

[1,] 0 0 0 0

[2,] 0 0 0 0

[3,] 0 0 0 0

[4,] 0 0 0 0 > a%/%b

[,1] [,2] [,3] [,4]

[1,] 1 1 1 1

[2,] 1 1 1 1

[3,] 1 1 1 1

[4,] 1 1 1 1

>

>

> #Data Frame

>

> data.frame(

+ Name =c("Mousumi","Kartik","Dastagir"),

+ age =c(17,20,18),

+ height =c(166,165,170))

Name age height

1. Mousumi 17 166
2. Kartik 20 165
3. Dastagir 18 170 >

> #In-build functions

> x <- 1:10

> print(x)

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

> length(x)

[1] 10

> max(x) [1] 10

> min(x) [1] 1

> sum(x)

[1] 55

> range(x)

[1] 1 10

> cumsum(x)

[1] 1 3 6 10 15 21 28 36 45 55

> mean(x)

[1] 5.5

> median(x)

[1] 5.5

> sort(x)

[1] 1 2 3 4 5 6 7 8 9

10 > sort(x,decreasing = TRUE)

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

> var(x)

[1] 9.166667

>

>

**PRACTICAL NO. 2**

>

> #Frequency Distribution

> > colour <- c("red","red","red","yellow","orange","yellow","violet","violet ","red")

> data.frame

(table(colour)) colour

Freq 1 orange 1

1. red 4
2. violet 2
3. yellow 2

>

>

> #Grouped Frequency Distribution

>

> a=c(83,51,66,61,82,65,54,56,92,60,65,87,68,64,51,

+ 70,75,66,74,68,44,55,78,69,98,67,82,77,79,62,38,88,76,99,

+ 84,47,60,42,66,74,91,71,83,80,68,65,51,56,73,55)

> data.frame(table(cut(a,c(0:7\*9+37))))

Var1 Freq

1. (37,46] 3
2. (46,55] 7
3. (55,64] 7
4. (64,73] 14
5. (73,82] 10
6. (82,91] 6
7. (91,100] 3

>

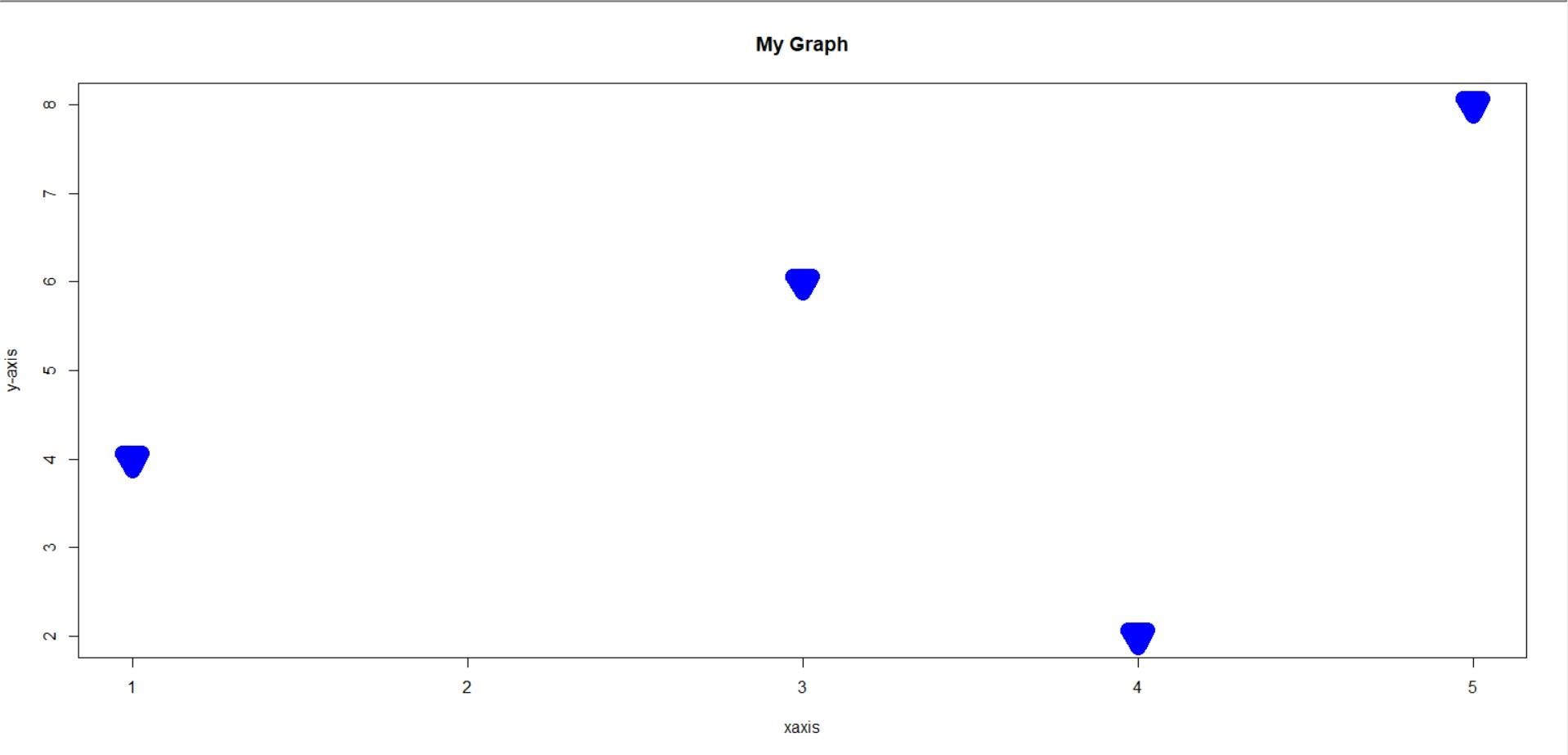
>

> #Diagrams and plot

>

> #R plot

> plot(c(1,3,4,5),c(4,6,2,8),pch=25,cex=2,col="blue",lwd="15",main="My Graph",xlab="xaxis",ylab="y-axis")



> #R line

> plot(1:10,type="l",col="blue",lwd="15",lty=6,main="My

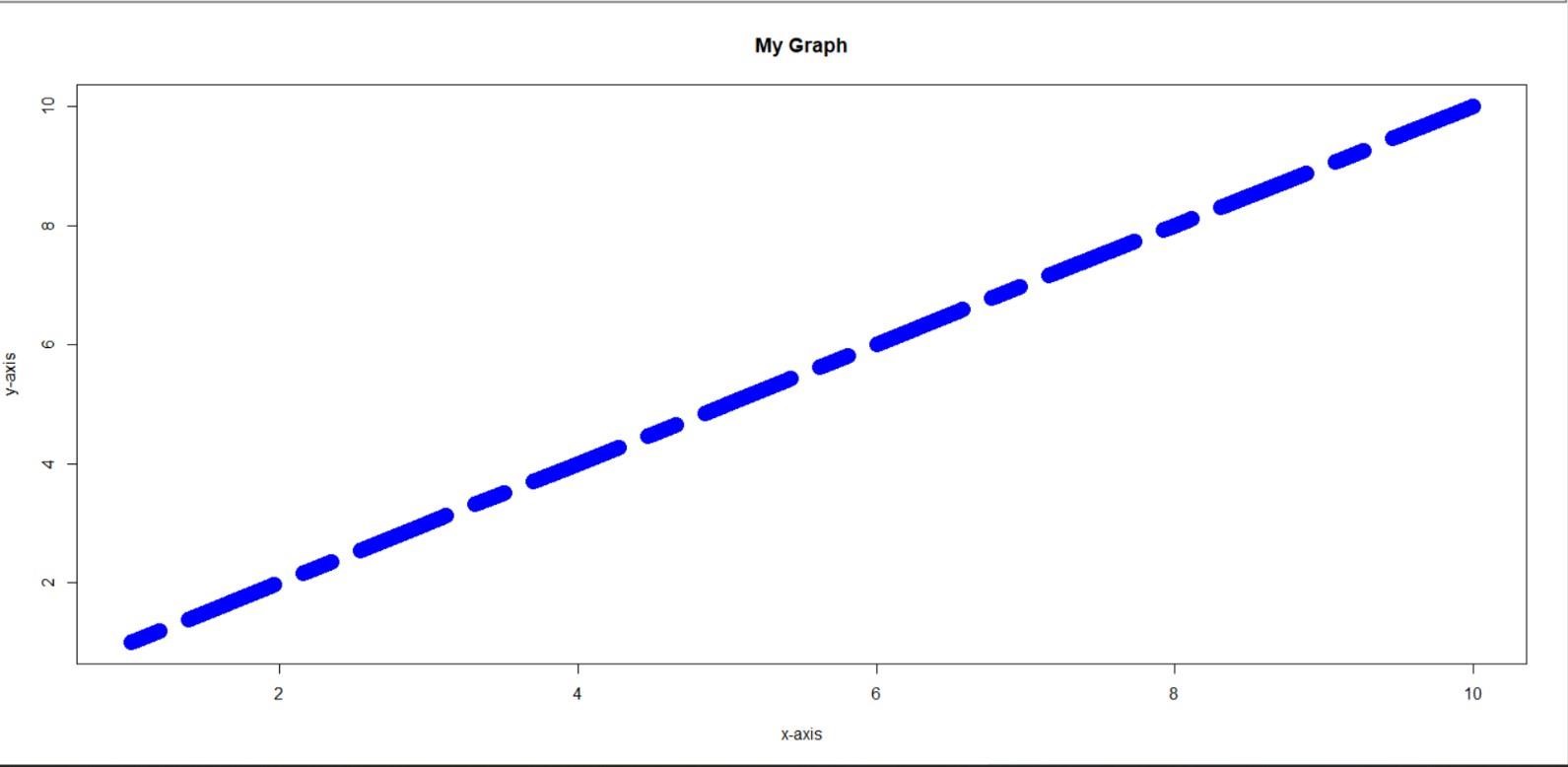
Graph",xlab="xaxis",ylab="y-axis")

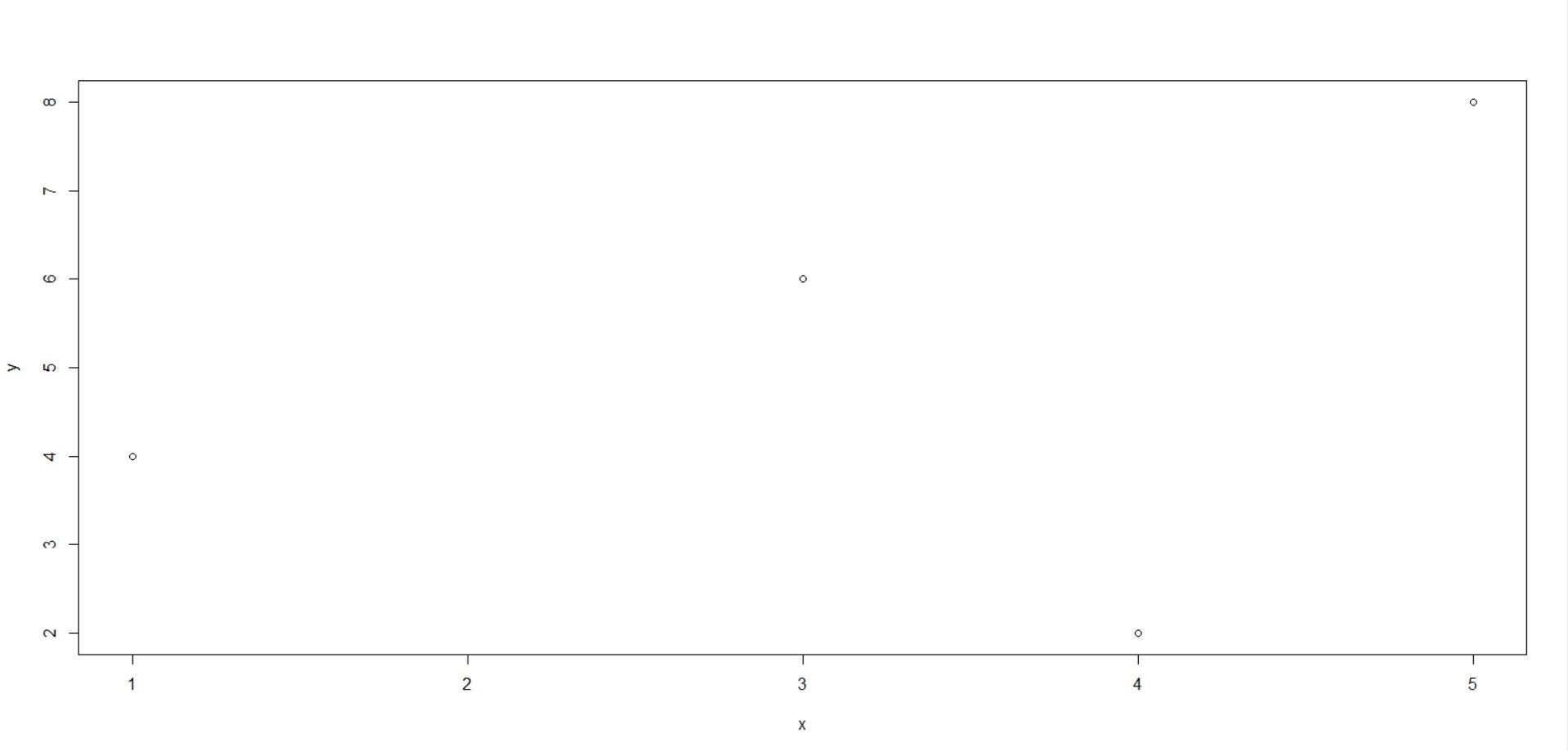
> #R 2 line

> x <- c(1,3,4,5)

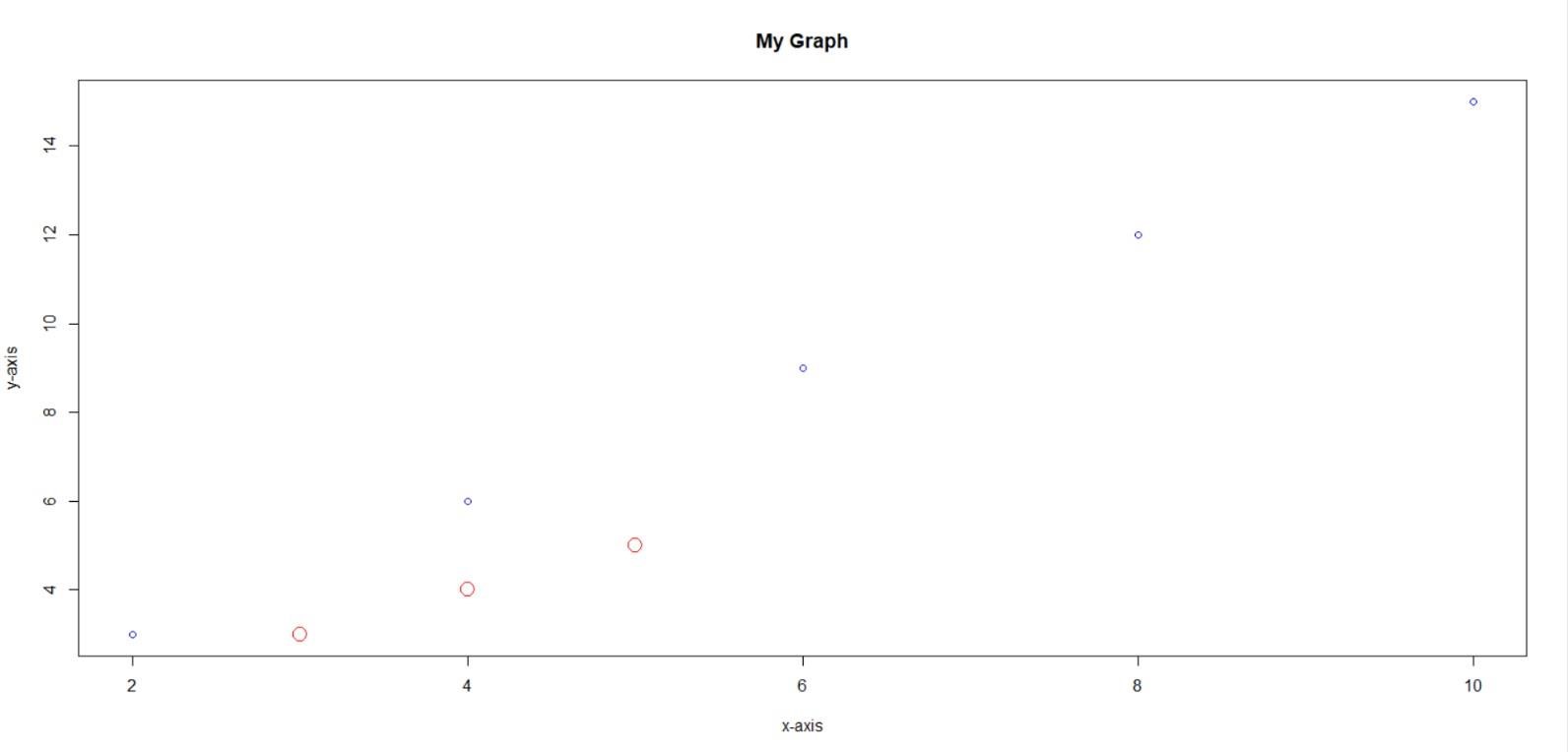
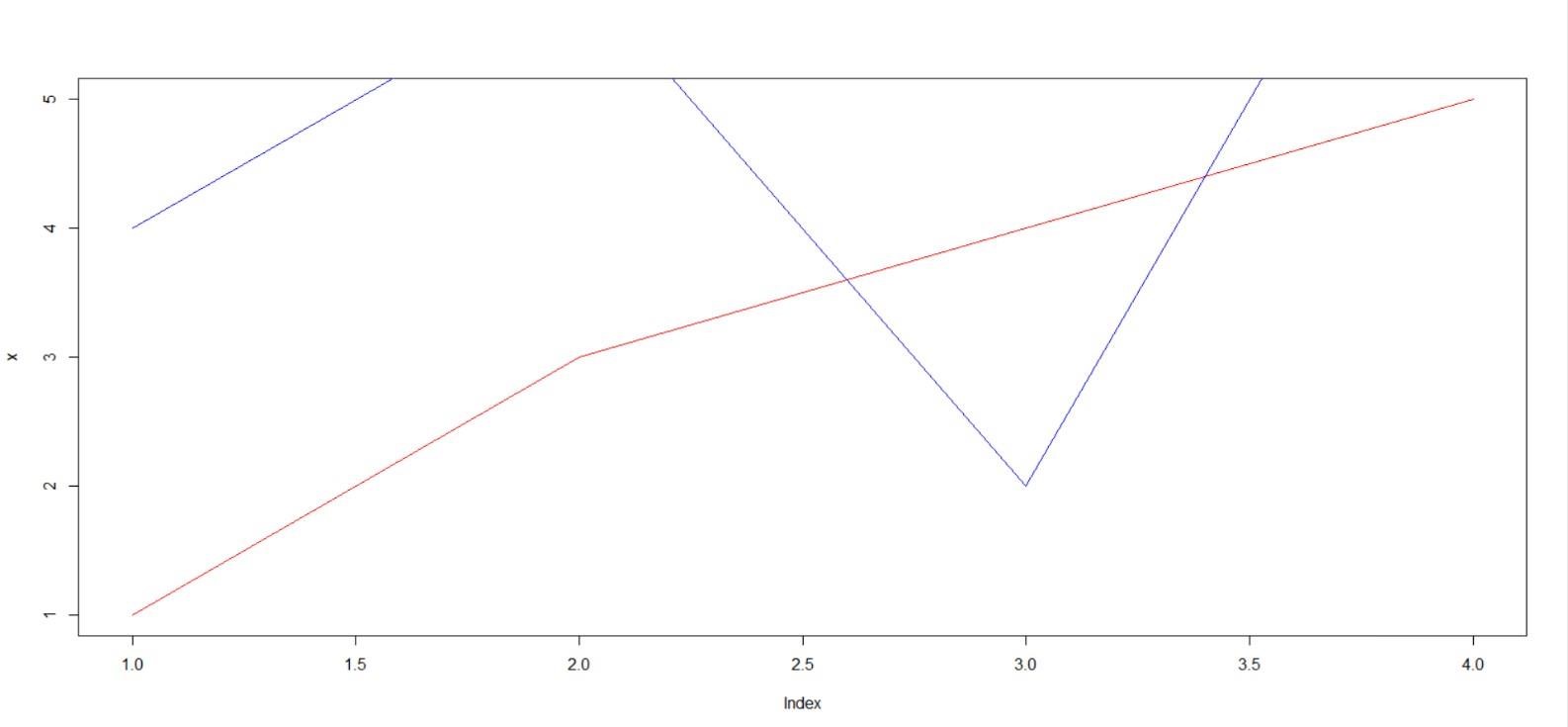
> y <- c(4,6,2,8)

> plot(x,y)



 > plot(x,type="l",col="red")

> lines(y,type="l",col="blue")



> #R Scatter plot

> a <- seq(from=2,to=10,by=2)

> b <- seq(3,15,3)

> c <- 1:5

> d <- 1:5

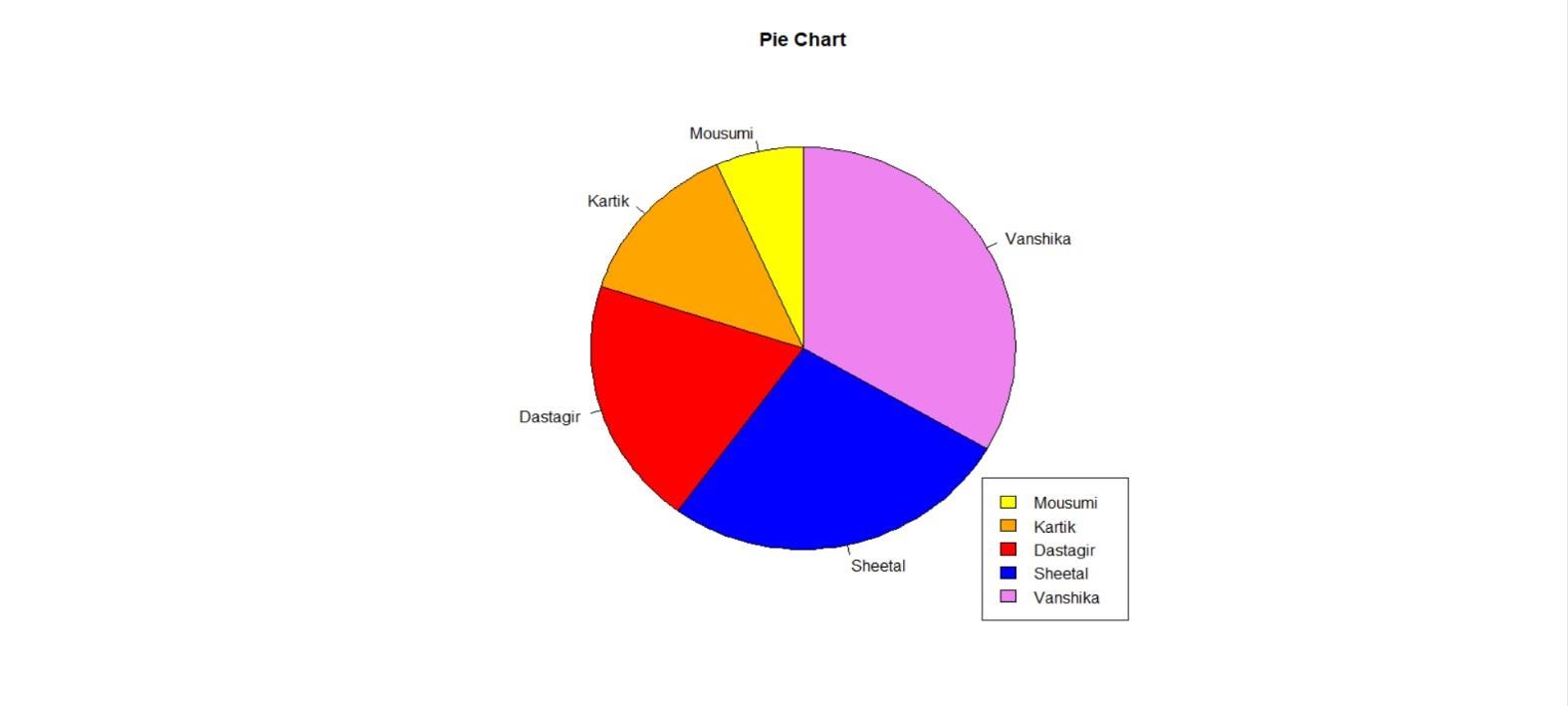
> plot(a,b,col="blue",main="My Graph",xlab="xaxis",ylab="y-axis") > points(c,d,col="red",cex=2)\ > #R pie

> x <- c("Mousumi","Kartik","Dastagir","Sheetal","Vanshika")

> y <- c("yellow","orange","red","blue","violet")

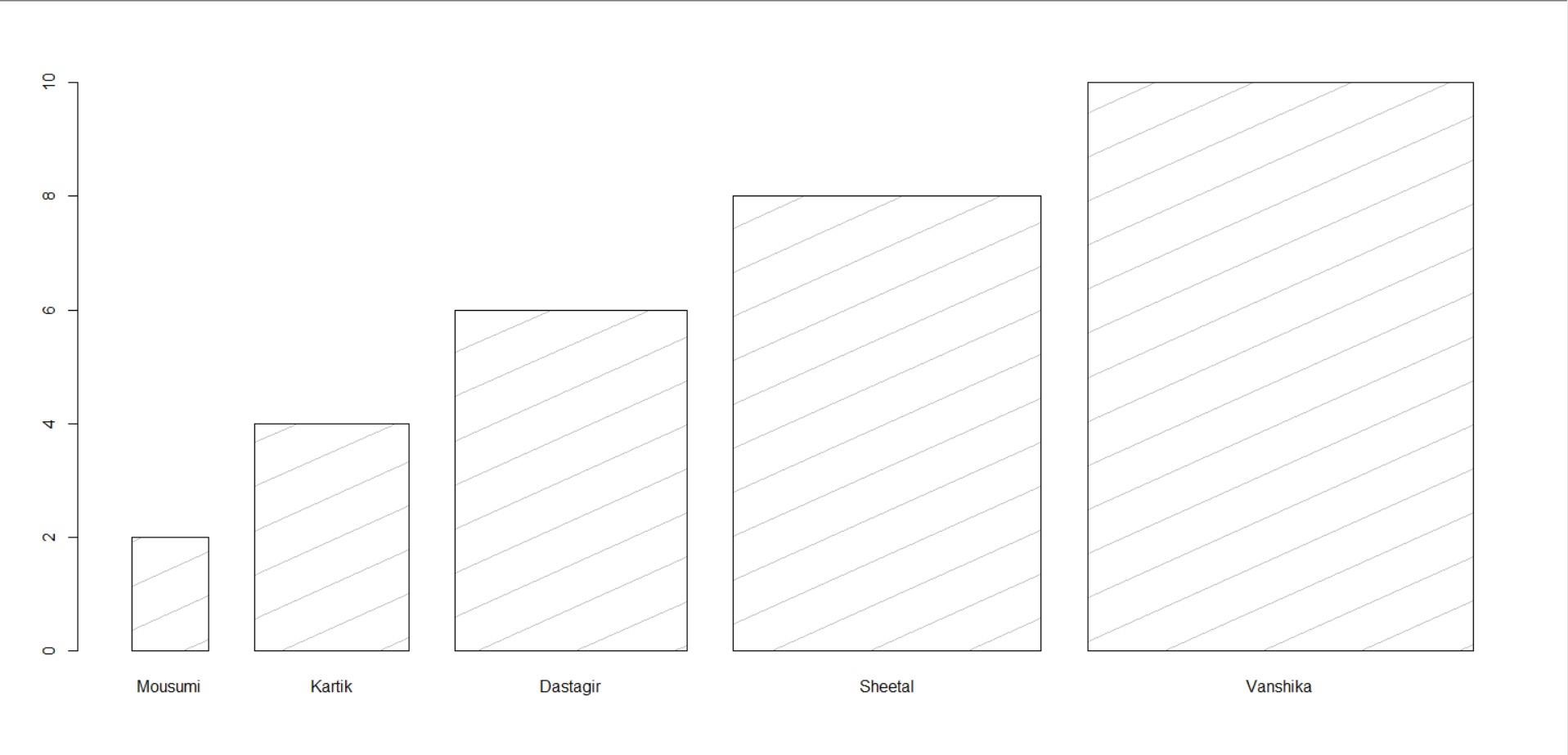
> pie(a,init.angle=90,labels=x,main = "Pie Chart",col=y)

> legend("bottomright",x,fill=y)

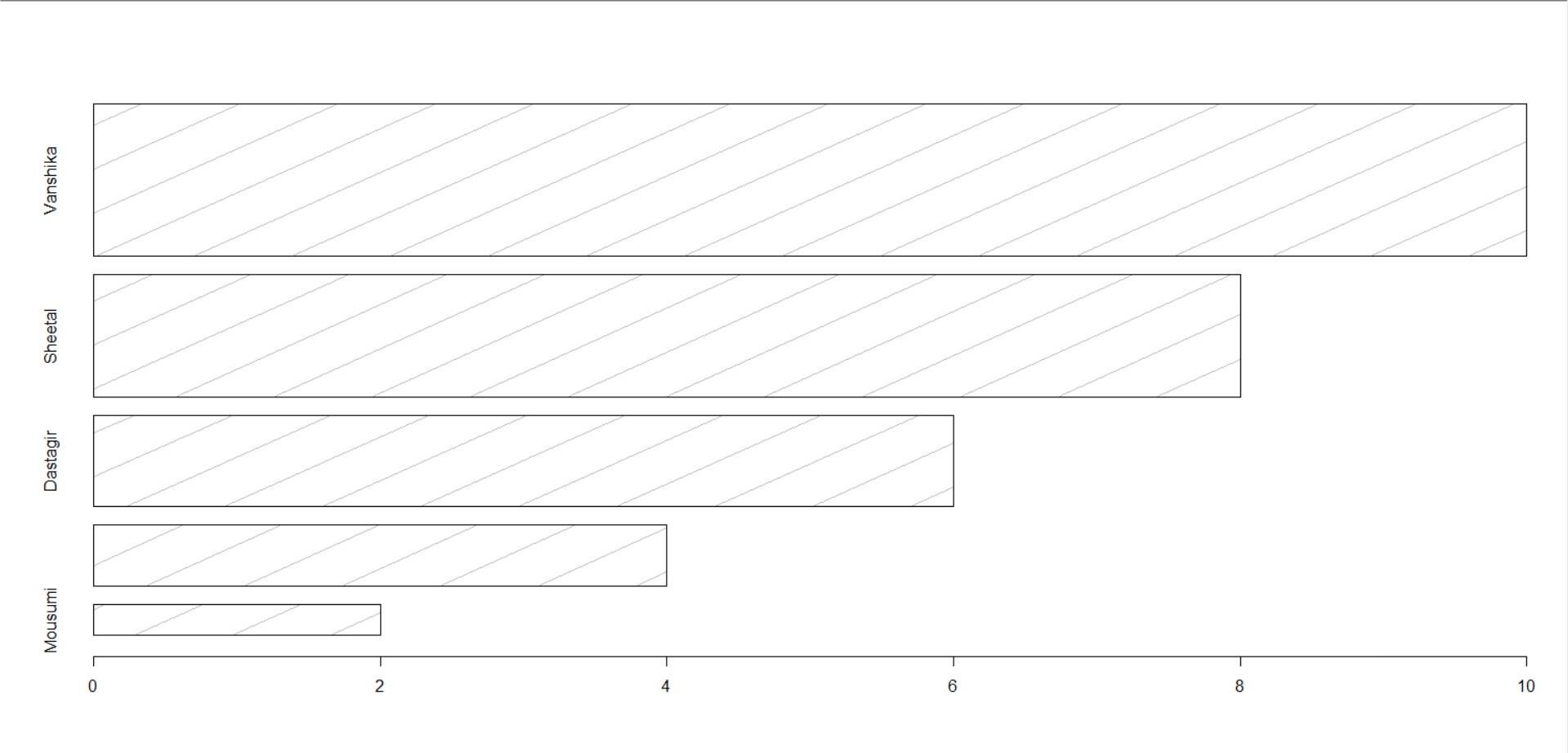


> #Vertical bar graph

> barplot(a,names.arg = x,density = 4,width = c(1,2,3,4,5))



> #Horizontal bar graph > barplot(a,names.arg = x,density = 4,width = c(1,2,3,4,5),horiz = TRUE)



**PRACTICAL NO.3**

> #data

> x <- seq(1,15,2)

> print(x)

[1] 1 3 5 7 9 11 13 15

>

> #Arithemetic mean

> mean(x) [1]

8

> #Median

> median(x)

[1] 8

> #Mode

> mod <- 10%%2

> print(mod)

1. 0

> #Partition Values

> quantile(x)

0% 25% 50% 75% 100%

1.0 4.5 8.0 11.5 15.0

**PRACTICAL NO. 4**

> a <- c(23,23,78,45,65,31,24,56,2)

>

> #1

> #Range

> range(a)

[1] 2 78

> #Highest value

> b <- max(a)

> #Lowest value

> c <- min(a)

> d <- (b-c)/(b+c)

> #Co-efficient of range

> print(d)

[1] 0.95

>

> #2

> b <- quantile(a,.75)

> c <- quantile(a,.25)

> #Quartile Deviation

> b-c/2

75%

44.5

> #Co-efficient of quartile deviation

> (b-c)/(b+c)

75%

0.4177215

>

> #3

> #Standard deviation

> sd(a)

[1] 24.21317

> #Variance

> var(a)

[1] 586.2778

> #Co-efficient of variance

> sd(a)/mean(a)\*100

[1] 62.80074

>

**PRACTICAL NO. 5**

library(moments)

>

> #Sequences

> a<-seq(2,20,2)

>

> #Central moments

> all.moments(a,order.max = 4,central = TRUE)

[1] 1.0 0.0 33.0 0.0 1933.8

>

> #Raw moments

> all.moments(a,order.max = 4,central = FALSE)

[1] 1.0 11.0 154.0 2420.0 40532.8

>

**PRACTICAL NO. 6**

>

> a<- c(65, 76, 64, 73, 74, 80, 71, 68, 66,

+ 81, 79, 75, 70, 62, 83, 63, 77, 78)

>

> print(paste("Karl Pearson's coefficient of skewness = ",3\*(mean(a)- median(a))/sd(a)))

[1] "Karl Pearson's coefficient of skewness = -0.450912695691985" >

> print(paste("Bowley's coefficient of skewness =

",(quantile(a, .75)+quantile(a, .25)-2\*median(a))/(quantile(a, .75)- quantile(a, .25))))

[1] "Bowley's coefficient of skewness = -0.244444444444444" >

> print(paste("Moment coefficient of skewness = ",skewness(a)))

[1] "Moment coefficient of skewness = -0.133183830666877" >

**PRACTICAL NO.7**

>

> a<- c(65, 76, 64, 73, 74, 80, 71, 68, 66,

+ 81, 79, 75, 70, 62, 83, 63, 77, 78)

>

> print(paste('Moment coefficient of Kurtorsis = ', kurtosis(a)))

[1] "Moment coefficient of Kurtorsis = 1.7409225063741"

>

**PRACTICAL NO.8**

>

> x <- c(65,66,67,67,68,69,70,72)

> y <- c(67,68,65,68,72,72,69,71)

> > print(paste("Karl Pearson's correlation coefficient = ",cor(x, y, method ="pearson")))

[1] "Karl Pearson's correlation coefficient = 0.603022689155527" > print(paste("Spearman's Rank correlation coefficient = ",cor(x, y, method ="spearman")))

[1] "Spearman's Rank correlation coefficient = 0.684861062761281" >