PART-A

//**

Aim1) Write a program to create a 3 X 3 matrices A and B and perform the following operations

a. AT.B

b. BT.(A.AT)

c. (A.AT).BT

d. [(B.BT)+(A.AT)-100I3]-1

Name:

Reg. No:U05DP21S0

Date:

//**

```
A<-matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,ncol=3)

B<-matrix(c(9,8,7,6,5,4,3,2,1),nrow=3,ncol=3)

#a)AT.B

result_a<-t(A)%*%B

#b)BT.(A.AT)

result_b<-t(B)%*%(A%*%t(A))

#c)(A.AT).BT

result_c<-(A%*%t(A))%*%t(B)

#d)[(B.BT)+(A.AT)-100*diag(3)^(-1)]

result_d<-solve((B%*%t(B))+(A%*%t(A))-100*diag(3))

cat("matrix A:\n")

print(A)

cat("matrix B:\n")

print(B)

cat("\na)AT.B:\n")

print(result_a)

cat("\nb)BT.(A.AT):\n")
```

```
print(result_b)
cat("\nc)(A.AT).BT:\n")
print(result_c)
cat("\nd)[(B.BT)+(A.AT)-100*diag(3)]^(-1):\n")
print(result_d)
```

Output:

```
matrix A:
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
matrix B:
     [,1] [,2] [,3]
[1,]    9    6    3
[2,]    8    5    2
[3,]    7    4    1

a)AT.B:
     [,1] [,2] [,3]
[1,]   46   28   10
[2,]  118   73   28
[3,]  190  118   46

b)BT.(A.AT):
     [,1] [,2] [,3]
[1,] 1848 2202 2556
```

```
[2,] 1146 1365 1584
[3,]  444  528  612
```

c)(A.AT).BT:

```
     [,1] [,2] [,3]
[1,] 1332 1098  864
[2,] 1584 1305 1026
[3,] 1836 1512 1188
```

d)[(B.BT)+(A.AT)-100*diag(3)]^(-1):

```
             [,1]         [,2]         [,3]
[1,] -0.006620683  0.004061135  0.004742954
[2,]  0.004061135 -0.005938865  0.004061135
[3,]  0.004742954  0.004061135 -0.006620683
```

//**

Aim2: Write R program to find roots of quadratic equation using user defined function. Test the program user supplied values for all possible cases.

Name:
Reg.No: U05DP21S0
Date:

//**

```
quadratic_equation<-function(a,b,c){
 d<-b^2-4*a*c
 if(d>0){
  x1<-(-b+sqrt(d))/(2*a)
  x2<-(-b-sqrt(d))/(2*a)
  print(paste("the roots of the equation are:",x1,"and",x2))
 }
 else if(d==0){
  x<- -b/(2^a)
  print(paste("root of the equation is:",x))
  }else{
  print("the equation has no real roots")
  }
}
a<-as.numeric(readline("enter the value of a:"))
b<-as.numeric(readline("enter the value of b:"))
c<-as.numeric(readline("enter the value of c:"))
quadratic_equation(a,b,c)
```

Output1:

enter the value of a:1

enter the value of b:2

enter the value of c:3

[1] "the equation has no real roots"

Output2:

enter the value of a:2
enter the value of b:9
enter the value of c:4
[1] "the roots of the equation are: -0.5 and -4"

Output3:

enter the value of a:1
enter the value of b:2
enter the value of c:1
[1] "root of the equation is: -1"

/*/**

Aim3: Write R script to generate prime numbers between two numbers using loops.

Name:

Reg.No: U05DP21S0

Date:

/*/**

```
generate_primes<-function(a,b)
{
  primes<-c()
  for(n in a:b)
  {
    if(all(n%%2:(n-1)!=0))
    {
      primes<-append(primes,n)
    }
  }
  return(primes)
}
```

<u>Output:</u>

```
source("C:/kavana/PartA3new.r")
> generate_primes(2,50)
 [1]  3  5  7 11 13 17 19 23 29 31 37 41 43 47
> generate_primes(1,10)
[1] 3 5 7
```

/***/
Aim4: Write an R program to create a list containing strings, numbers, vectors and logical values and do the following manipulations over the list
a. Access the first element in the list
b. Give the names to the elements in the list
c. Add element at some positions in the list
d. Remove the element
e. print the first and third element
f. Update the third element

Name:
Reg.No: U05DP21S0
Date:
/***/

```
list_data<-list("Red",c(21,32,11),TRUE,51.23)

print(list_data)

list_data[1]

names(list_data)<-c("color","marks","flag","average")

list_data

p1=as.numeric(readline("Enter Position to insert"))

append(list_data,"canada",after=p1-1)

list_data

p2=as.numeric(readline("Enter Position to Remove"))

list_data[-p2]

list_data

list_data[1]

list_data[3]

list_data[3]<-88.97

list_data

list_data["average"]<-67.5
```

list_data

Output:

```
list_data<-list("Red",c(21,32,11),TRUE,51.23)
> print(list_data)
[[1]]
[1] "Red"
[[2]]
[1] 21 32 11
[[3]]
[1] TRUE
[[4]]
[1] 51.23

> list_data[1]
[[1]]
[1] "Red"

> names(list_data)<-c("color","marks","flag","average")
> list_data
$color
[1] "Red"
$marks
[1] 21 32 11
$flag
[1] TRUE
$average
```

```
[1] 51.23

> p1=as.numeric(readline("Enter Position to insert"))
Enter Position to insert1
> append(list_data,"canada",after=p1-1)
[[1]]
[1] "canada"
$color
[1] "Red"
$marks
[1] 21 32 11
$flag
[1] TRUE
$average
[1] 51.23

> list_data
$color
[1] "Red"
$marks
[1] 21 32 11
$flag
[1] TRUE
$average
[1] 51.23

> p2=as.numeric(readline("Enter Position to Remove"))
```

```
Enter Position to Remove2
> list_data[-p2]
$color
[1] "Red"
$flag
[1] TRUE
$average
[1] 51.23

> list_data
$color
[1] "Red"
$marks
[1] 21 32 11
$flag
[1] TRUE
$average
[1] 51.23

> list_data[1]
$color
[1] "Red"

> list_data[3]
$flag
[1] TRUE
```

```
> list_data[3]<-88.97
> list_data
$color
[1] "Red"
$marks
[1] 21 32 11
$flag
[1] 88.97
$average
[1] 51.23

> list_data["average"]<-67.5
> list_data
$color
[1] "Red"
$marks
[1] 21 32 11
$flag
[1] 88.97
$average
[1] 67.5
```

/***/
Aim5: The following table shows the time taken (in minutes) by 100 students to travel to school on a particular day.

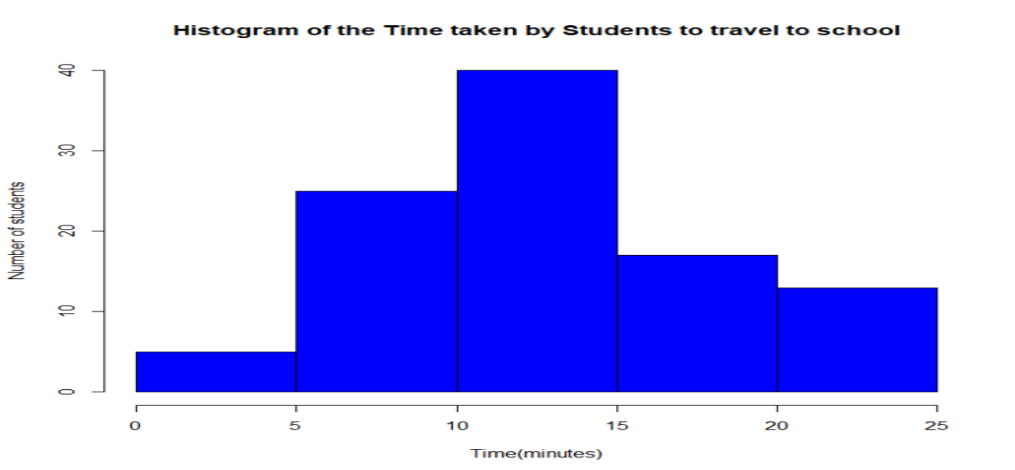
Time	0-5	5-10	10-15	15-20	20-25
No. of students	5	25	40	17	13

a. Draw the histogram
b. Draw frequency polygon

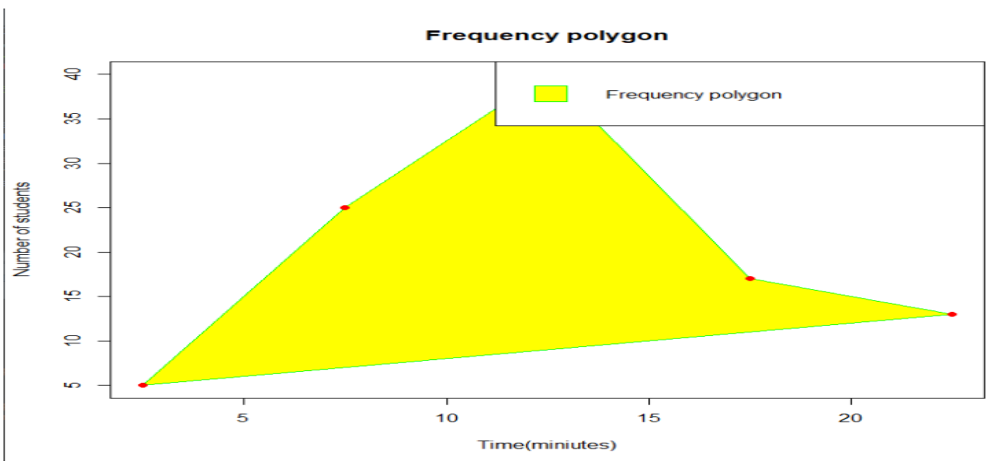
Name:
Reg.No: U05DP21S0
Date:
/***/

```
time_intervals<-c("0-5","5-10","10-15","15-20","20-25")

no_of_students<-c(5,25,40,17,13)

midpoints<-c(2.5,7.5,12.5,17.5,22.5)

time_taken<-c(rep(2.5,5),rep(7.5,25),rep(12.5,40),rep(17.5,17),rep(22.5,13))

hist(time_taken,

    breaks=c(0,5,10,15,20,25),

    col="blue",

    xlab="Time(minutes)",

    ylab="Number of students",

    main="Histogram of the Time taken by Students to travel to school")

plot(midpoints,no_of_students,type="n",xlab="Time(miniutes)",ylab="Number of students",main="Frequency polygon")

polygon(midpoints,no_of_students,col="yellow",border="green")

points(midpoints,no_of_students,pch=16,col="red")

legend("topright",legend="Frequency polygon",fill="yellow",border="green")
```

Output:



Histogram of the Time taken by Students to travel to school
Number of students
0
10
20
30
40
0
5
10
15
20
25
Time(minutes)



Frequency polygon
Frequency polygon
Number of students
5
10
15
20
25
30
35
40
5
10
15
20
Time(miniutes)

/***/

Aim6: Write an R program to create a Data Frame with following details and do the following operations.

ItemCode	itemCategory	ItemPrice
1001	Electronics	700
1002	Desktop Supplies	300
1003	Office Supplies	350
1004	USB	400
1005	CD Drive	800

a. **Subset the Data frame and display the details of only those items whose price is greater than or equal to 350.**
b. **Subset the Data frame and display only the items where the category is either "Office Supplies" or "Desktop Supplies"**
c. **Subset the Data frame and display the items where the Itemprice between 300 and 700**
d. **Compute the sum of all ItemPrice**
e. **Create another Data Frame called "item-details" with three different fields itemCode, ItemQtyonHand and ItemReorderLvl and merge the two frames.**

Name:

Reg.No: U05DP21S0

Date:

/***/

```
Data<-data.frame(itemCode=c(1001,1002,1003,1004,1005),
        itemCategory=c("Electronics","DesktopSuppliers","OfficeSuppliers","USB","CD Drives"),
        itemPrice=c(700,300,350,400,800))


subset_a<-data[data$itemPrice>=350,]
```

```
subset_b<-data[data$itemCategory%in%c("OfficeSuppliers","DesktopSuppliers"),]
subset_c<-data[data$itemPrice>=300&data$itemPrice<=700,]
total_Price<-sum(data$itemPrice)
item_details<-data.frame(
  itemCode=c(1001,1002,1003,1004,1005),
  itemQtyonhand=c(10,15,20,5,12),
  itemReorderLvl=c(2,5,3,4,6)
        )
merge_data<- merge(data,item_details,by="itemCode")
print("a.subset greater than=350")
print(subset_a)
print("b.subset item is office or desktop")
print(subset_b)
print("c.between 300 and 700")
print(subset_c)
print("d.sum of the item")
print(total_Price)
print("e.Merge data")
print(merge_data)
```

Output:

```
[1] "a.subset greater than=350"
 itemCode    itemCategory    itemPrice
1   1001      Electronics        700
3   1003    OfficeSuppliers      350
4   1004          USB            400
5   1005      CD Drives          800
```

[1] "b.subset item is office or desktop"

	itemCode	itemCategory	itemPrice
2	1002	DesktopSuppliers	300
3	1003	OfficeSuppliers	350

[1] "c.between 300 and 700"

	itemCode	itemCategory	itemPrice
1	1001	Electronics	700
2	1002	DesktopSuppliers	300
3	1003	OfficeSuppliers	350
4	1004	USB	400

[1] "d.sum of the item"

[1] 2550

[1] "e.Merge data"

	itemCode	itemCategory	itemPrice	itemQtyonhand	itemReorderLvl
1	1001	Electronics	700	10	2
2	1002	DesktopSuppliers	300	15	5
3	1003	OfficeSuppliers	350	20	3
4	1004	USB	400	5	4
5	1005	CD Drives	800	12	6

/***/

Aim7: Create a factor marital_status with levels Married, single, divorced. Perform the following operations on this factor a. Check the variable is a factor

- **b. Access the 2nd and 4th element in the factor**
- **c. Remove third element from the factor**
- **d. Modify the second element of the factor**
- **e. Add new level widowed to the factor and add the same level to the factor marital_status**

Name:
Reg.No: U05DP21S0
Date:

/***/

```
marital_status<-
factor(c("Married","Single","Divorced","Widow","Married","Single","Widow","Divorced"))

is.factor(marital_status)

marital_status[c(2,4)]

marital_status<-marital_status[-3]

marital_status

marital_status[2]<-"Widow"

print(marital_status)

marital_status

levels(marital_status)<-c(levels(marital_status),"Widowed")

marital_status
```

Output:

```
> is.factor(marital_status)
[1] TRUE
> marital_status[c(2,4)]
[1] Single Widow
Levels: Divorced Married Single Widow
> marital_status<-marital_status[-3]
> marital_status
[1] Married  Single   Widow   Married  Single   Widow   Divorced
Levels: Divorced Married Single Widow
> marital_status[2]<-"Widow"
> print(marital_status)
[1] Married  Widow   Widow   Married  Single   Widow   Divorced
Levels: Divorced Married Single Widow
> marital_status
[1] Married  Widow   Widow   Married  Single   Widow   Divorced
Levels: Divorced Married Single Widow
> levels(marital_status)<-c(levels(marital_status),"Widowed")
> marital_status
[1] Married  Widow   Widow   Married  Single   Widow   Divorced
Levels: Divorced Married Single Widow Widowed
```

//**

Aim 8: Write a R language Script for following operation on Iris Data Set

1. Load the Iris Dataset

2. View first six rows of iris dataset

3. Summarize iris dataset

4. Display number of rows and columns

5. Display column names of dataset.

6. Create histogram of values for sepal length

7. Create scatterplot of sepal width vs. sepal length

8. Create boxplot of sepal width vs. sepal length

9. Find Pearson correlation between Sepal.Length and Petal.Length

10. Create correlation matrix for dataset

Name:

Reg.No: U05DP21S0

Date:

/*/**

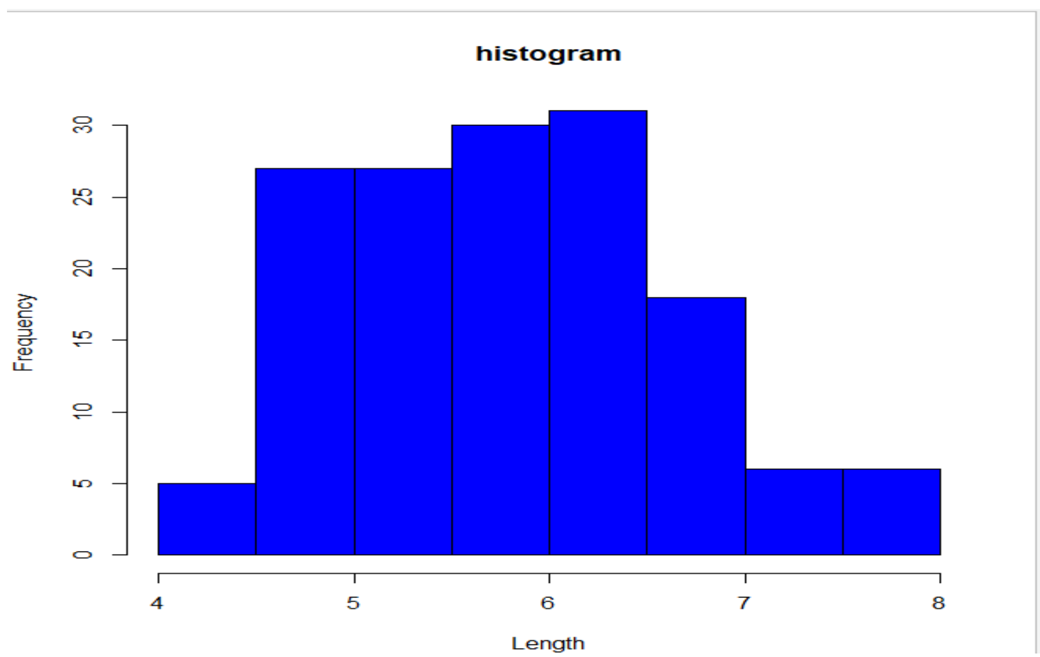
```
data("iris")

head(iris)

summary(iris)

dim(iris)

names(iris)

hist(iris$Sepal.Length,col="blue",main="histogram",xlab="Length",ylab="Frequency")

plot(iris$Sepal.Width,iris$Sepal.Length,col="blue",main="scatter plot",xlab="Sepal
Width",ylab="Sepal Length")

plot(iris$Sepal.Length~Species,data=iris,col="blue",border="green",main="Sepal Length by
species",xlab="Species",ylab="Sepal Length")

cor(iris$Sepal.Length,iris$Sepal.Width,method=c("pearson"))

str(iris)
```

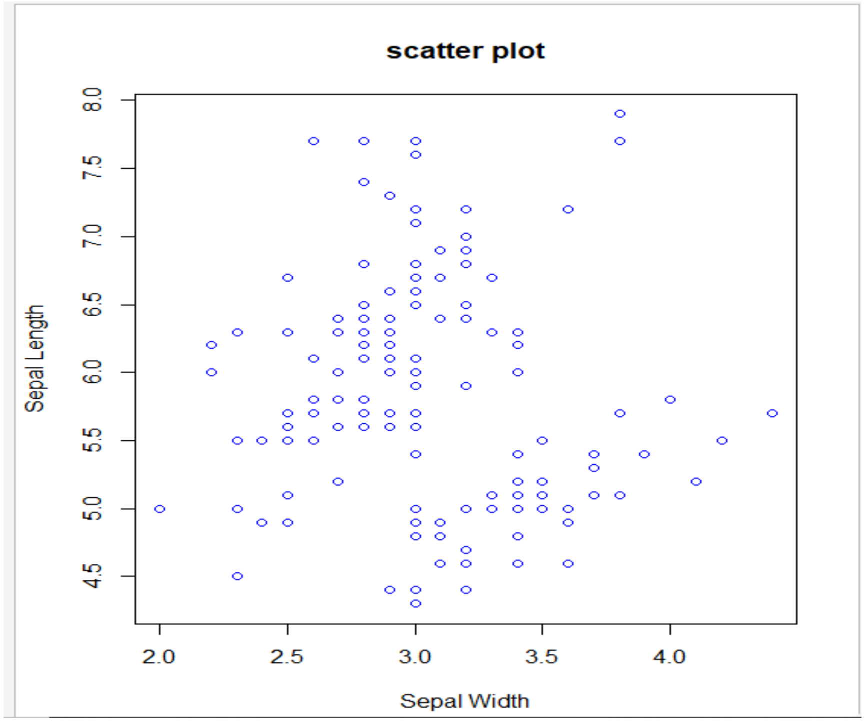
Output:

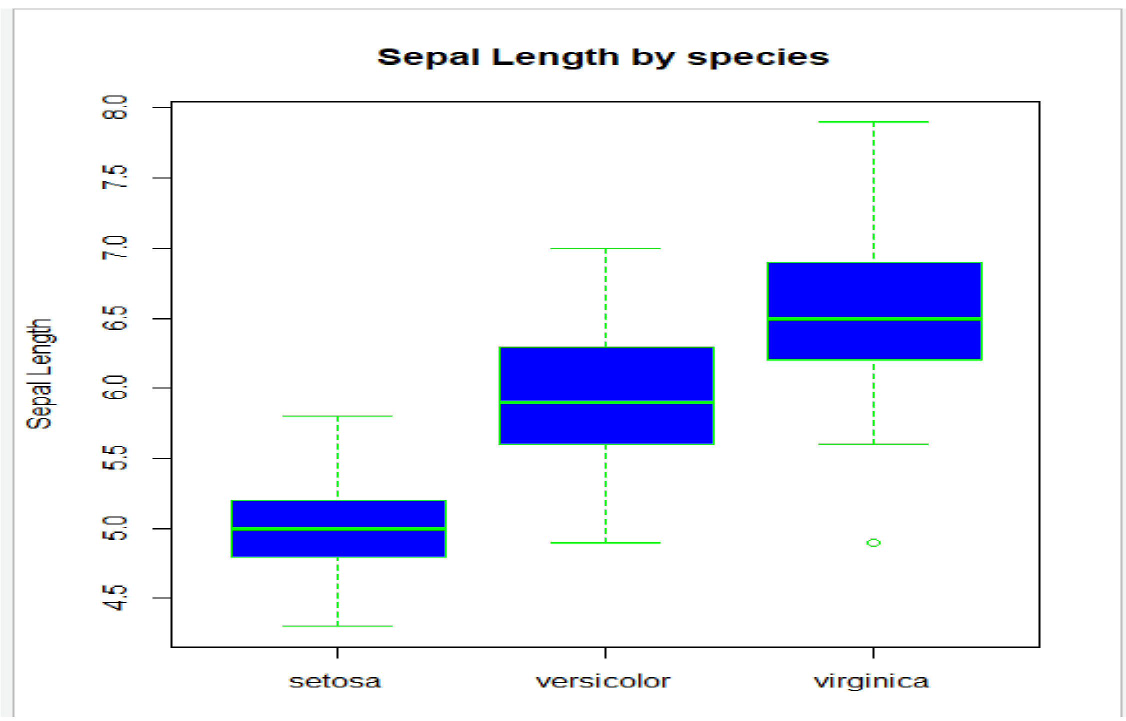
```
data("iris")

> head(iris)

  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.2  setosa

> summary(iris)

  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
Species
 setosa    :50
 versicolor:50
 virginica :50

 > dim(iris)
[1] 150   5
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
```

[5] "Species"

```
> hist(iris$Sepal.Length,col="blue",main="histogram",xlab= ="Length",ylab="Frequency")

> plot(iris$Sepal.Width,iris$Sepal.Length,col="blue",main="scatter plot",xlab="Sepal Width",ylab="Sepal Length")

> plot(iris$Sepal.Length~Species,data=iris,col="blue",border="green",main="Sepal Length by species",xlab="Species",ylab="Sepal Length")

> cor(iris$Sepal.Length,iris$Sepal.Width,method=c("pearson"))

[1] -0.1175698

> str(iris)

'data.frame':   150 obs. of  5 variables:

 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...

 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...

 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...

 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...

 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1
```




scatter plot
Sepal Length
Sepal Width
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
2.0
2.5
3.0
3.5
4.0


Sepal Length by species
Sepal Length
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
setosa
versicolor
virginica

PART-B

/**/

Aim 1: Write a R program to create a Vector containing following 8 values and perform the
following operations.
4 3 0 5 2 9 4 5
a. Find mean, median, mode.
b. Find the range.
c. Find the 35th and 78th percentile.
d. Find the variance and standard deviation
e. Find the interquartile range.
f. Find the z-score for each value.
Name:
Reg.No: U05DP21S0
Date:
/***/

```
vec<-c(4,3,0,5,2,9,4,5)
paste("Mean=",mean(vec))
paste("Median=",median(vec))
getmode<-function(v){
uniqv<-unique(v)
uniqv[which.max(tabulate(match(v,uniqv)))]}
mode<-getmode(vec)
paste("Mode=",mode)
paste("Range=",diff(range(vec)))
quantile(vec,prob=c(0.35,0.78))
paste("variance=",var(vec))
paste("Standard deviation=",sd(vec))
paste("Interquantile range=",IQR(vec))
vec_zscore<-((vec-mean(vec))/sd(vec))
vec_zscore
```

Output:

```
vec<-c(4,3,0,5,2,9,4,5)
> paste("Mean=",mean(vec))
[1] "Mean= 4"
> paste("Median=",median(vec))
[1] "Median= 4"
> getmode<-function(v){
+ uniqv<-unique(v)
+ uniqv[which.max(tabulate(match(v,uniqv)))]}
> mode<-getmode(vec)
> paste("Mode=",mode)
[1] "Mode= 4"
> paste("Range=",diff(range(vec)))
[1] "Range= 9"
> quantile(vec,prob=c(0.35,0.78))
 35%  78%
3.45 5.00
> paste("variance=",var(vec))
[1] "variance= 6.85714285714286"
> paste("Standard deviation=",sd(vec))
[1] "Standard deviation= 2.61861468283191"
> paste("Interquantile range=",IQR(vec))
[1] "Interquantile range= 2.25"
> vec_zscore<-((vec-mean(vec))/sd(vec))
> vec_zscore
[1]  0.0000000 -0.3818813 -1.5275252  0.3818813 -0.7637626  1.9094065  0.0000000
0.3818813
```

/***/

Aim 2: Write R script to find the correlation coefficient and type of correlation between advertisement expenses and sales volume using Karl Pearson's coefficient of correlation method (Direct Method).

Firm	1	2	3	4	5	6	7	8	9	10
Advertisement Exp. (Rs. In Lakhs)	11	13	14	16	16	15	15	14	13	13
Sales Volume (Rs. In Lakhs)	50	50	55	60	65	65	65	60	60	50

Name:
Reg.No: U05DP21S0
Date:

/**/

```
advertisement_exp<-c(11,13,14,16,16,15,15,14,13,13)

sales_volume<-c(50,50,55,60,65,65,65,60,60,50)

mean_ad_exp<-mean(advertisement_exp)

mean_sales_volume<-mean(sales_volume)

sum_deviation_product<-sum((advertisement_exp-mean_ad_exp)*(sales_volume-
mean_sales_volume))

sum_squared_dev_ad_exp<-sum((advertisement_exp-mean_add_exp)^2)

sum_squared_dev_sales_volume<-sum((sales_volume-mean_sales_volume)^2)

correlation_coefficient<-
sum_deviation_product/sqrt(sum_squared_dev_ad_exp*sum_squared_dev_sales_volume)

if(correlation_coefficient>0){

  correlation_type<-"positive correlation"

} else if

(correlation_coefficient<0){

  correlation_type<-"Negative correlation"
```

```
} else{
  correlation_type<-"No correlation"
}
cat("Correlation Coeffiecient:",correlation_coefficient,"\n")
cat("Type of correlation:",correlation_type,"\n")> cat("Correlation Coeffiecient:",correlation_coefficient,"\n")
```

Output:

```
Correlation Coeffiecient: 0.7865665
Type of correlation: positive correlation
```

/***/

Aim3: Write R script to compute the regression equation of y on x from the following data. Predict the value of y when x=7

X	2	4	5	6	8	11
Y	18	12	10	8	7	5

Name:
Reg.No: U05DP21S0
Date:

/***/

```
x<-c(2,4,5,6,8,11)

y<-c(18,12,10,8,7,5)

model<-lm(y~x)

summary(model)

new_data<-data.frame(x=7)

predicted_y<-predict(model,newdata=new_data)

cat("Regression
equation:y=",round(coefficients(model)[1],2),"+",round(coefficients(model)[2],2),"x\n")

cat("Predicted y when x=7:",round(predicted_y,2),"\n")
```

Output:

```
cat("Regression equation:y=",round(coefficients(model)[1],2),"+",round(coefficients

(model)[2],2),"x\n")

Regression equation:y= 18.04 + -1.34 x

> cat("Predicted y when x=7:",round(predicted_y,2),"\n")

Predicted y when x=7: 8.66
```

/***/

Aim4: The times taken by a large group of students to complete a piece of homework, T minutes, are Normally distributed with a mean of 57 minutes and standard deviation of 6.5. Find the probability that the time taken by a random student from the group to complete this homework will be less than 60 minutes. Write R script to Find the probability that the time taken by a random student from the group to complete this homework

a) Will be less than 60 minutes

b) Between 50 and 80 minutes

Name:

Reg.No: U05DP21S0

Date:

/**/

```
mean_time<-57

std_deviation<-6.5

prob_less_than_60<-pnorm(60,mean=mean_time,sd=std_deviation)

cat("probability that time is less than 60 minutes:",prob_less_than_60,"\n")

prob_between_50_and_80<-pnorm(80,mean=mean_time,sd=std_deviation)-
pnorm(50,mean=mean_time,sd=std_deviation)

cat("Probability that time isbetween 50 and 80 minutes:",prob_between_50_and_80,"\n")
```

Output:

probability that time is less than 60 minutes: 0.6777938

Probability that time is between 50 and 80 minutes: 0.8590415

//**

Aim5: Write R script to perform the following using binomial distribution

i. If n=4 and p=0.10, find P(x=3)

ii. If n=12 and p=0.45, find P(5<=x<=7)

Name:
Reg.No: U05DP21S0
Date:

//**

```
n1<-4

p1<-0.10

prob_x_3<-dbinom(3,size=n1,prob=p1)

n2<-12

p2<-0.45

prob_x_between_5_7<-sum(dbinom(5:7,size=n2,prob=p2))

cat("i.p(x=3)=",prob_x_3,"\n")

cat("ii.p(5<=x<=7)=",prob_x_between_5_7,"\n")

        cat("i.p(x=3)=",prob_x_3,"\n")
```

<u>Output:</u>

```
i.p(x=3)= 0.0036

> cat("ii.p(5<=x<=7)=",prob_x_between_5_7,"\n")

ii.p(5<=x<=7)= 0.583828
```

/***/

Aim6: Perform the following using uniform distribution between 200 and 240

i. P(x>230)

ii. P(205≤x≤220)

Name:
Reg.No: U05DP21S0
Date:
/*/**

```
n<-10000

random_numbers<-runif(n,min=200,max=240)

probability_x_gt_230<-mean(random_numbers>230)

cat("i.P(x>230):",probability_x_gt_230,"\n")

probability_x_between_205_and_220<-mean(random_numbers>=205&
random_numbers<=220)

cat("ii.P(205<=x<=220):",probability_x_between_205_and_220,"\n")
```

<u>Output:</u>

```
> probability_x_between_205_and_220<-mean(random_numbers>=250 &
random_numbers<=220)

i.p(x>230): 0.2479

> cat("ii.p(205<=x<=220):",probability_x_between_205_and_220,"\n")

ii.P(205<=x<=220): 0.3765
```

/***/

Aim7: Following are the scores of max vertical jumps before and after the training program. Test whether the training program is helpful to the students (Use Paired t-test).

Player	Max Vertical Jump Before Training Program	Max Vertical Jump After Training Program
Player 1	22	24
Player 2	20	22
Player 3	19	19
Player 4	24	22
Player 5	25	28
Player 6	25	26
Player 7	28	28
Player 8	22	24
Player 9	30	30
Player 10	27	29
Player 11	24	25
Player 12	18	20
Player 13	16	17
Player 14	19	18
Player 15	19	18
Player 16	28	28
Player 17	24	26
Player 18	25	27
Player 19	25	27
Player 20	23	24

Name:
Reg.No: U05DP21S0
Date:

/*/**

```
data<-data.frame(
  player=1:20,
  before=c(22,20,19,24,25,25,28,22,30,27,24,18,16,19,19,28,24,25,25,23),
  after=c(24,22,19,22,28,26,28,24,30,29,25,20,17,18,18,28,26,27,27,24))
result<-t.test(data$before,data$after,paired=TRUE)
cat("paired t_test result:\n")
cat("t-value:",result$statistic,"\n")
cat("p-value:",result$p.value,"\n")
cat("degrees of freedom:",result$parameter,"\n")
cat("confidence interval of the difference:",result$conf.int,"\n")
cat("effect size(cohen'sd):",(mean(data$before)-mean(data$after))/sd(data$before-data$after),"\n")
alpha<-0.05
if(result$p.value<alpha){
  cat("the training program is statstically significant in improving max vertical jumps.\n")
}else{
  cat("there is no significant improvemnet in max vertical  jumps after the training program.\n")
}
```

Output:

```
> source("~/rp/partb7.r")
paired t_test result:
t-value: -3.226173
p-value: 0.004445371
degrees of freedom: 19
confidence interval of the difference: -1.566325 -0.3336745
effect size(cohen'sd): -0.7213943
the training program is statstically significant in improving max vertical jumps.
```

/***/

Aim8: A company has three manufacturing plants, and company officials want to determine
whether there is difference in the average age of workers at the three locations. The following data are the ages of five randomly selected workers at each plant. Perform a one-way ANOVA to determine whether there is a significant difference in the mean ages of the workers at three plants. Use α=0.01. Write R script for the above problem.

Plant (Employee Ages)

1	2	3
29	32	25
27	33	24
30	31	24
27	34	25
28	30	25

Name:
Reg.No: U05DP21S0
Date:

/***/

```
plant1 <- c(29,27,30,27,28)

plant2 <- c(32,33,31,34,30)

plant3 <- c(25,24,24,25,25)


data1 <- data.frame(

  Plant = factor(rep(1:3,each = 5)),

  Age = c(plant1,plant2,plant3)

)

data1
```

```
result <- aov(Age~Plant,data = data1)
summary(result)

pvalue <- summary(result)[[1]][["Pr(>F)"]][1]
alpha <- 0.01
pvalue
if(pvalue<alpha){
  cat("There is a sgnificant in the mean ages of workers at three plants (p-value = ",pvalue,")")
}else{
  cat("There is no sgnificant in the mean ages of workers at three plants (p-value = ",pvalue,")")
}
```

Output:

data1

	plant	age
1	1	29
2	1	27
3	1	30
4	1	27
5	1	28
6	2	32
7	2	33
8	2	31
9	2	34
10	2	30
11	3	25

```
12       3          24

13       3          24

14       3          25

15       3          25

> result<-aov(age~plant,data=data1)

> summary(result)

              Df     Sum Sq      Mean Sq     F value     Pr(>F)

plant          2      136.9           68.47       45.64      2.46e-06

Residuals     12       18.0            1.50

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> pvalue<-summary(result)[[1]][["Pr(>F)"]][1]

> alpha<-0.01

> pvalue

[1] 2.459041e-06

>   if(pvalue < alpha){

+   cat("There is a significant difference in the mean ages of workers atthree plants(p_value=",pvalue,")")

+ }else{

+   cat("There is no significant difference in the mean ages of workers atthree plants(p_value=",pvalue,")")

+ }

There is a significant difference in the mean ages of workers atthree plants(p_value= 2.459041e-06 )
```