

R PROGRAMMING

EXPERIMENT-1

ADDITION:

AIM:

To prove the program for addition using R-tool.

PROGRAM:

```
num1=as.integer(readline(prompt = "enter the first number:"))
num2=as.integer(readline(prompt = "enter the second number:"))
num3=num1+num2
print(num3)
```

OUTPUT:

```
Enter a number1 : 2
Enter a number2 : 2
[1] 4
```

RESULT:

Thus the basic program addition are executed successfully.

EXPERIMENT-2

SUBTRACTION:

AIM:

To prove the program for subtraction using R-tool.

PROGRAM:

```
num1=as.integer(readline(prompt = "enter the first number:"))
num2=as.integer(readline(prompt = "enter the second number:"))
num3=num1-num2
print(num3)
```

OUTPUT:

```
Enter a number1 : 4
Enter a number2 : 2
[1] 2
```

RESULT:

Thus the basic program subtraction are executed successfully.

EXPERIMENT-3

MULTIPLICATION:

AIM:

To prove the program for multiplication using R-tool.

PROGRAM:

```
num1=as.integer(readline(prompt = "enter the first number:"))
num2=as.integer(readline(prompt = "enter the second number:"))
num3=num1*num2
print(num3)
```

OUTPUT:

```
> source("~/active-rstudio-document")
enter the first number:3
enter the second number:2
[1] 6
> |
```

RESULT:

Thus the basic program multiplication are executed successfully.

EXPERIMENT-4

DIVISION:

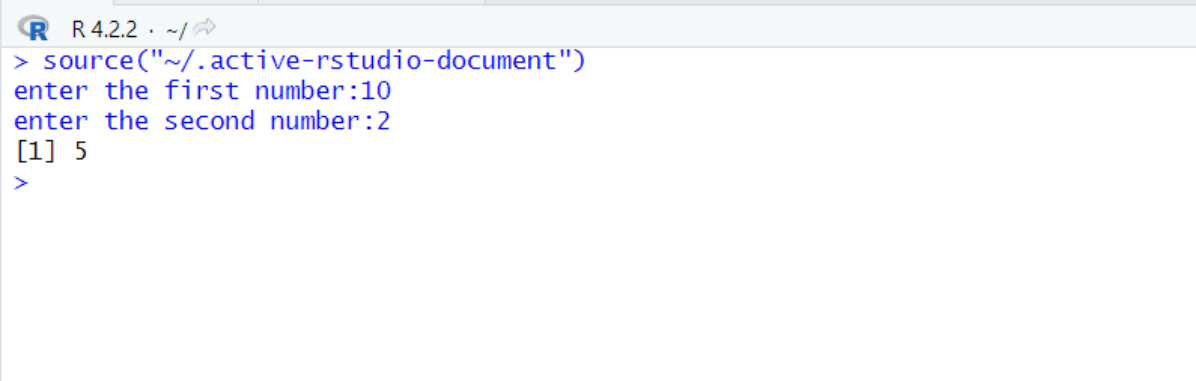
AIM:

To prove the program for division using R-tool.

PROGRAM:

```
num1=as.integer(readline(prompt = "enter the first number:"))
num2=as.integer(readline(prompt = "enter the second number:"))
num3=num1/num2
print(num3)
```

OUTPUT:



```
R 4.2.2 · ~/...
> source("~/active-rstudio-document")
enter the first number:10
enter the second number:2
[1] 5
>
```

RESULT:

Thus the basic program division was executed successfully.

EXPERIMENT-5

ODD OR EVEN:

AIM:

To write the program for odd or even using R-tool.

PROGRAM:

```
num=as.integer(readline(prompt="enter a number:"))
if((num%%2)==0)
{
  print("number is a even")
}else{
  print("number is odd")
}
```

OUTPUT:

```

> source("D:/folders/DWHDM/EXERCISE_1(BASIC_PROGRAMS)/1_odd_or_even.R")
Enter a number : 4
[1] "Number is even"
> source("D:/folders/DWHDM/EXERCISE_1(BASIC_PROGRAMS)/1_odd_or_even.R")
Enter a number : 5
[1] "Number is odd"
> source("D:/folders/DWHDM/EXERCISE_1(BASIC_PROGRAMS)/1_odd_or_even.R")
Enter a number : 1
[1] "Number is odd"
>

```

RESULT:

Thus the basic program odd or even was executed successfully.

EXPERIMENT-6

MEAN,MEDIAN,MODE:

AIM:

To write the program for mean,median,mode.

PROGRAM:

MEAN

```

names<-c("siri","mahir","chiru")
age<-c(23,24,25)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
mean(df $age)
write.csv(df,"datafr.csv")

```

MEDIAN

```

names<-c("siri","mahir","chiru")
age<-c(23,24,25)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
median(df $age)
write.csv(df,"datafr.csv")

```

MODE

```
names<-c("siri","mahi","chiru")
age<-c(23,24,25)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
mode(df $age)
write.csv(df,"datafr.csv")
```

OUTPUT:

```
> mode(df $age)
[1] "numeric"
```

```
> mean(df $age)
[1] 27.33333
```

```
> median(df $age)
[1] 24
```

```
> mode(df $age)
[1] "numeric"
```

RESULT:

Thus the central tendency and measure of dispersion is executed successfully.

EXPERIMENT-7

SUMMARY:

AIM:

To write the program for summary using R-tool.

PROGRAM:

```
names<-c("siri","mahi","chiru")
age<-c(23,24,25)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
summary(df $age)
write.csv(df,"datafr.csv")
```

OUTPUT:

```
> summary(df $age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 23.00  23.50   24.00   27.33  29.50   35.00
```

RESULT:

Thus the central tendency and measure of dispersion is executed successfully.

EXPERIMENT-8

GREATER AMONG THREE NUMBERS:

AIM:

To write the program for the greatest among three numbers.

PROGRAM:

```
x <- as.integer(readline(prompt = "Enter first number :"))
y <- as.integer(readline(prompt = "Enter second number :"))
z <- as.integer(readline(prompt = "Enter third number :"))

if (x > y && x > z) {
  print(paste("Greatest is :", x))
} else if (y > z) {
  print(paste("Greatest is :", y))
} else{
  print(paste("Greatest is :", z))
}
```

OUTPUT:

```
R 4.2.2 . ~/
> source("~/active-rstudio-document")
Enter first number :5
Enter second number :6
Enter third number :4
[1] "Greatest is : 6"
> |
```

RESULT:

Thus the greatest among the three numbers was executed successfully.

EXPERIMENT-9

IQR:

AIM:

To write the program for central tendency and data dispersion measures using R tool.

PROGRAM:

```
names<-c("siri","mahi","chiru")
age<-c(23,24,25)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
IQR(df $age)
write.csv(df,"datafr.csv")
```

OUTPUT:

```
> IQR(df $age)
[1] 6
```

RESULT:

Thus the program for central tendency and data dispersion measures was executed successfully.

EXPERIMENT-10

QUANTILE:

AIM:

To write the program for central tendency and data dispersion measures.

PROGRAM:

```
names<-c("siri","mahi","chiru")
age<-c(23,24,25)
```

```
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
quantile(df $age)
write.csv(df,"datafr.csv")
```

OUTPUT:

```
> quantile(df $age)
 0%   25%   50%   75%  100%
23.0 23.5 24.0 29.5 35.0
```

RESULT:

Thus the program for central tendency and data dispersion measures was executed successfully

EXPERIMENT-11

MID RANGE:

AIM:

To write the program for central tendency and data dispersion measures.

PROGRAM:

```
names <- c("siri", "mahi", "chiru")
age <- c(23, 24, 25)
marks <- c(88, 78, 25)
df <- data.frame(names, age, marks)
min_age <- min(df$age)
max_age <- max(df$age)
cat(min_age, max_age, "\n")
write.csv(df, "datafr.csv", row.names = FALSE)
cat("Data frame has been written to 'datafr.csv'\n")
```

OUTPUT:

```
> source("~/11.MID RANGE.R")
23 25
```

RESULT:

Thus the program for central tendency and data dispersion measures was executed successfully

EXPERIMENT-12

Z-SCOORE NORMALIZATION:

AIM:

To write the program for Z-scoore normalization using R-tool.

PROGRAM:

```
##Z score normalization
data <- matrix(c(10, 20, 30, 40, 50), nrow = 5, ncol = 1)
z_score_normalization <- function(x) {
  (x - mean(x)) / sd(x)
}
normalized_data <- apply(data, 2, z_score_normalization)
cat("Original data:\n")
print(data)
cat("\nNormalized data (Z-scores):\n")
print(normalized_data)
```

OUTPUT:

```
> source("~/12.Z-SCOORE NORMALIZATION.R")
Original data:
      [,1]
[1,]    10
[2,]    20
[3,]    30
[4,]    40
[5,]    50

Normalized data (Z-scores):
      [,1]
[1,] -1.2649111
[2,] -0.6324555
[3,]  0.0000000
[4,]  0.6324555
[5,]  1.2649111
>
```

RESULT:

Thus the Z-scoore normalization using R tool was executed successfully.

EXPERIMENT-13

MIN,MAX,MEAN,MINMAX:

AIM:

To write the program for the minimum,maximum,mean and minmax using r-TOOL

PROGRAM:

```
data <- c(10, 20, 30, 40, 50)

Mean<-mean(data)

Minimum<-min(data)

Maximum<-max(data)
```

```
MinMax<-(data-Minimum)/(Maximum-Minimum)

print(Mean)

print(Minimum)

print(Maximum)

print(MinMax)
```

OUTPUT:

```
> source("~/13.MIN,MAX,MEAN,MINMAX.R")
[1] 30
[1] 10
[1] 50
[1] 0.00 0.25 0.50 0.75 1.00
> |
```

RESULT:

Thus the program for min,max,minmax,mean was executed successfully.

EXPERIMENT-14

BAR PLOT AND HORIZONTAL BAR:

AIM:

To draw the bar plot and horizontal bar using R-tool.

PROGRAM:

```
a<-c(55,67,89,80,90)

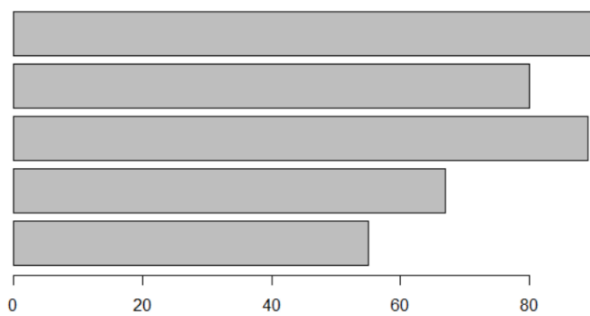
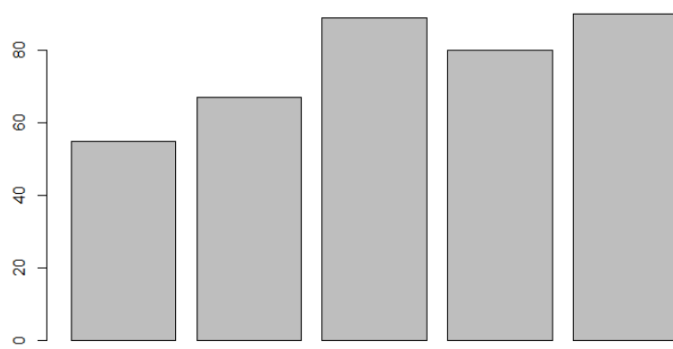
barplot(a)

a<-c(55,67,89,80,90)

barplot(a)

barplot(a,hORIZ=TRUE)
```

OUTPUT:



RESULT:

Thus the bar and horizontal bar plot was executed successfully.

EXPERIMENT-15

BOX PLOT:

AIM:

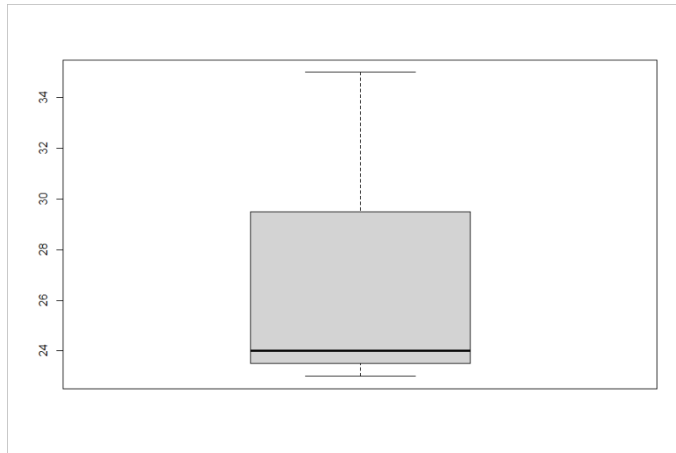
To draw the box plot using R-tool.

PROGRAM:

```
names<-c("siri","chru","loki")
```

```
age<-c(23,24,25)
marks<-c(88,78,25)
df<-data.frame(names,age,marks)
hist(df$age)
boxplot(df$age)
```

OUTPUT:



RESULT:

Thus the box plot was executed successfully.

EXPERIMENT-16

HISTOGRAM:

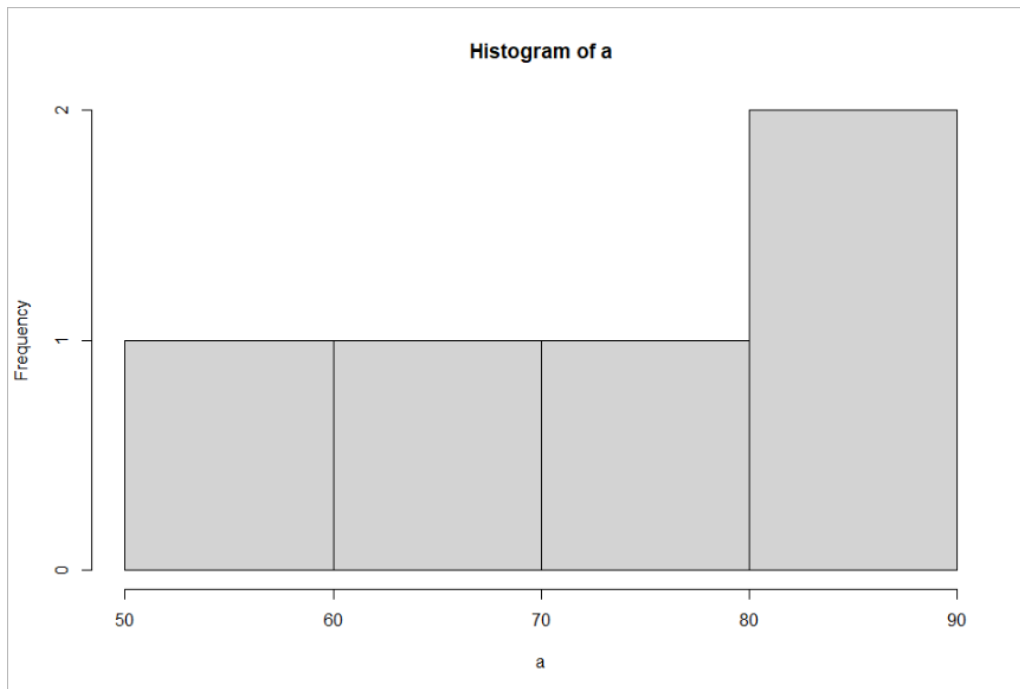
AIM:

To draw the histogram plot using R-tool.

PROGRAM:

```
a<-c(55,67,89,80,90)
hist(a)
```

OUTPUT:



RESULT:

Thus the histogram plot was executed successfully.

EXPERIMENT-17

CORRELATION ANALYSIS:

AIM:

To write the program for correlation analysis using R-tool.

PROGRAM:

```
diabetest1<-read_excel("C:/Users/M.Geetha/Downloads/NARA.xlsx")
diabetest1<-table(diabetest1 $Age,diabetest1 $Insulin)
diabetest1
chisq.test(diabetest1)
```

OUTPUT:

```
> diabetes1
```

	0	14	15	16	18	22	23	25	29	32	36	37	38	40	41	42	43	44	45	46	48	49	50	51
21	28	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1
22	29	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	1	0
23	10	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	2	0	0	0
24	15	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	0
25	18	1	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0

	52	53	54	55	56	57	58	59	60	61	63	64	65	66	67	68	70	71	72	73	74	75	76	77
21	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	0	2	0
22	0	1	1	0	0	0	1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	1	1	0

RESULT:

Thus the correlation analysis was executed successfully.

EXPERIMENT-18

SCATTER PLOT:

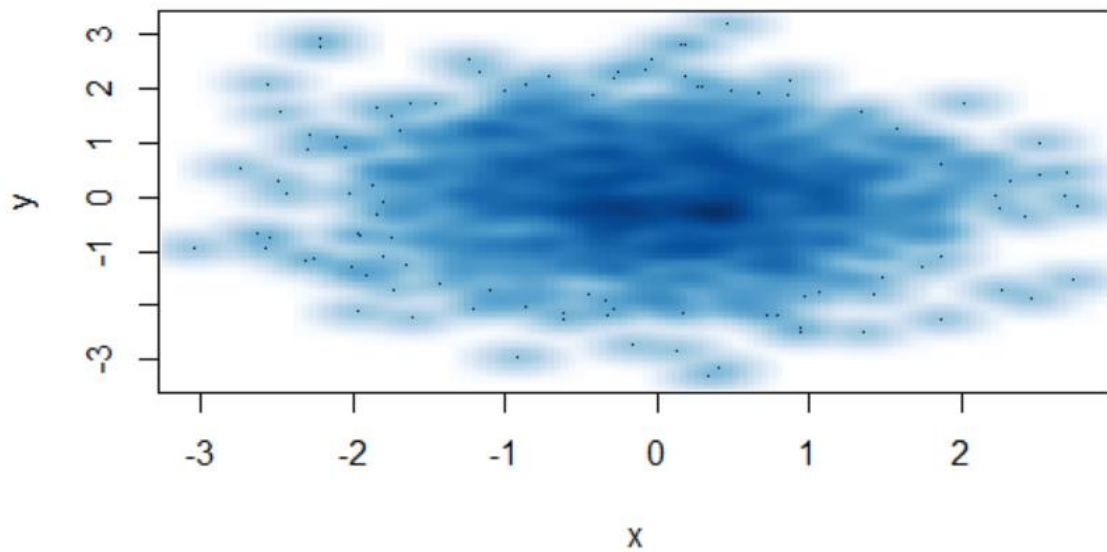
AIM:

To draw the scatter plot using R-tool

PROGRAM:

```
set.seed(9)
x <- rnorm(1000)
y <- rnorm(1000)
smoothScatter(y ~ x)
smoothScatter(x,y)
```

OUTPUT:



RESULT:

Thus the scatter plot was executed successfully.

EXPERIMENT-19

LINEAR REGRESSION:

AIM:

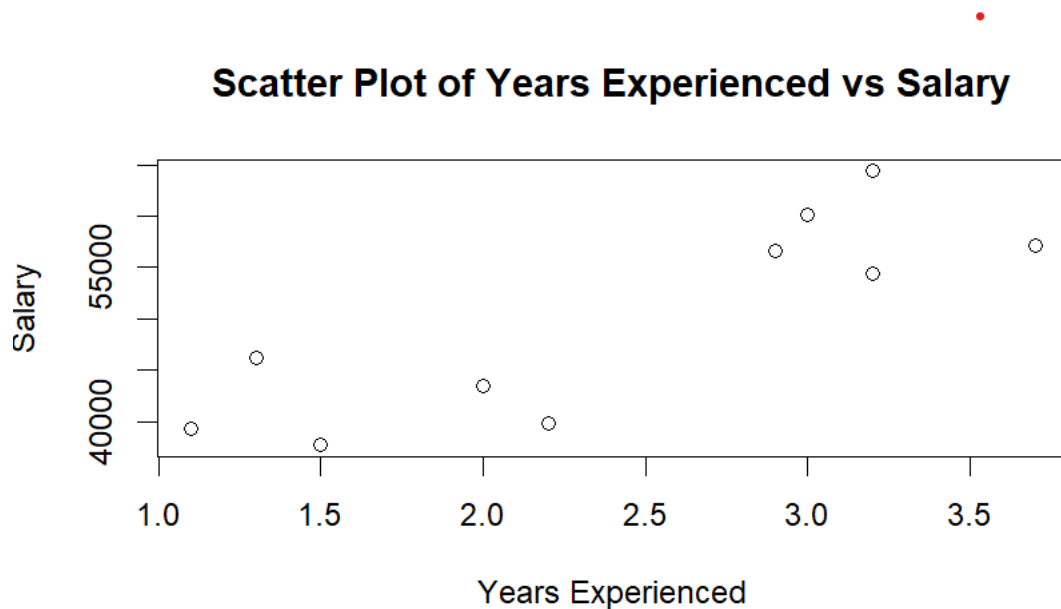
To write the program for the linear regression using R-tool.

PROGRAM:

```
# Create the data frame
data <- data.frame(
  Years_Exp = c(1.1, 1.3, 1.5, 2.0, 2.2, 2.9, 3.0, 3.2, 3.7),
  Salary = c(39343.00, 46205.00, 37731.00, 43525.00,
            39891.00, 56642.00, 60150.00, 54445.00, 64445.00, 57189.00)
)
# Create the scatter plot
plot(data$Years_Exp, data$Salary,
```

```
xlab = "Years Experienced",  
ylab = "Salary",  
main = "Scatter Plot of Years Experienced vs Salary")
```

OUTPUT:



RESULT:

Thus the linear regression program was executed successfully.

EXPERIMENT-20

MULTIPLE REGRESSION:

AIM:

To write the program for the multiple regression.

PROGRAM:

```
Input <- diabetes[,c("Age", "BloodPressure", "Glucose")]  
Model <- lm(Age~ BloodPressure+Glucose,data=input)  
Print(model)
```

OUTPUT:


```

> print(diabetes)

Call:
lm(formula = Age ~ BloodPressure + Glucose, data = input)

Coefficients:
(Intercept)  BloodPressure      Glucose
   14.33937      0.12399      0.08547

> |

```

```
A<- coef(model)[1]
```

```
Print(A)
```

OUTPUT:

```

> print(A)
(Intercept)
   14.33937

> |

```

```
xBloodPressure<- coef(model)[2]
```

```
yGlucose<- coef(model)[3]
```

```
print(xBloodPressure)
```

```
print(yGlucose)
```

OUTPUT:

```

> print(yGlucose)
Glucose
0.08547277

>

```

```
y = A+xBloodPressure + yGlucose
```

```
print(y)
```

OUTPUT:

```

>
> print(y)
(Intercept)
   14.54883

>

```

