Output -

```
[1] "Good Morning!"
```

Output -

```
[[1]]
[1] 2 4 8

[[2]]
[1] 1

[[3]]
[1] 2

[[4]]
[1] 3

[[5]]
function (x)  .primitive("sin")
```

Page no.: ____/
Date: __/__/____

practical - 1

1. Using R execute the basic commands, array, list and frames.

For Basic commands -

```
>mytext <- "Good Morning!"
>print(mytext)
```

For List -

```
>list1<- list(c(2,4,8), 1,2,3, sin)
>print(list1)
```

Output-

```
, , 1

     [,1]  [,2]  [,3]
[1,] "no"  "yes" "yes"
[2,] "no"  "yes" "no"
[3,] "yes" "no"  "yes"

, , 2

     [,1]  [,2]  [,3]
[1,] "no"  "yes" "no"
[2,] "yes" no    "yes"
[3,] "no"  yes   "no"
```

Output-

	gender	height	weight	Age
1.	female	152.0	35	14
2.	female	155.0	40	16
3.	Male	165.0	60	20
4.	female	170.0	55	17

Page no.: ____/
Date: ____/____/____/

For Array -

```
> A <- array(c('yes','no'), dim = c(3,3,2))
> print(A)
```

For Frame -

```
> BMI <- data.frame(
+ gender = c("female", "female", "Male", "female"),
+ height = c(152, 155, 165.2, 170.0),
+ weight = c (35, 40, 60, 55),
+ Age = c(14, 16, 20, 17)
+ )
> print (BMI)
```

Output -
[1] 11.0 5.2 10.0

Output -
[1] -5.0 -0.8 2.0

Output -
[1] 24.0 6.6 24.0

Output -
[1] 2 6 4

Output -
[1] 3.0 2.2 2.0

Date: __/__/____

practical - 2

2. Create a matrix using R perform the operations addition, inverse, transpose and multiplication operations.

⇒ For operations addition, inverse and multiplication -

a)
```
> p <- c(3, 2.2, 6)
> q <- c(8, 3, 4)
> print(p+q)
```

b)
```
> p <- c(3, 2.2, 6)
> q <- c(8, 3, 4)
> print(p-q)
```

c)
```
> p <- c(3, 2.2, 6)
> q <- c(8, 3, 4)
> print(p*q)
```

d)
```
> p <- c(16, 18, 16)
> q <- c(8, 3, 4)
> print(p/q)
```

e)
```
> p <- c(3, 5.2, 6)
> q <- c(8, 3, 4)
> print(p%*%q)
```

Output -
[1] 3 1 1

Output -
[1] 9 8 1

Output -
[1] 4 5 6 7 8 9 10

Output -
[1] FALSE
[2] TRUE

Output -

	[,1]	[,2]
[1,]	53	89
[2,]	89	161

Page no.: ____
Date: __/__/____

f) > p<-c(25, 3.2, 6)
> q<-c(8,3,4)
> print(p %% / %% q)

g) > p<- c(3,2,1)
> q<-c(2,3,4)
> print(p ^ q)

⇒ for transpose operation -

a) > p<- 4:10
print(p)

b) > p<- 12
> q<- 6
> t<-1:10
> print(p %% in %% t)
> print(q %% in %% t)

c) > m = matrix(c(1,6,4,1,12,4), nrow = 2, ncol = 3, byrow = TRUE
> t = m %% * %% t(m)
> print(t)

Output-

[1] 7.636364

Output-

[1] 4.8

Output-

[1] NA

Page no.: ____/

Date: ___/___/____/

practical-3

3. Using R execute the statistical functions: mean, median, mode, quartiles, range, inter quantile range. histogram.

⇒ For mean-

a) Find the mean of 12, 7, 3, 4, 2, 18, 2, 54, -21, 8, -5.

Code-

```
> x<- c(12,7,3,4,2,18,2,54,-21,8,-5)
> Result.mean <- mean(x)
> print(Result.mean)
```

b) Find the mean of 12, 7, 3, 4, 2, 18, 2, 54, -21, 8, -5 applying trim option.

Code-

```
x<-c(12,7,3,4,2,18,2,54,-21,8,-5)
Result.mean <- mean(x, trim =0.3)
print(Result.mean)
```

c) Find the mean of 12, 7, 3, 4, 2, 18, 2, 54, -21, 8, -5 applying NA option.

Code-

```
x<-c(12,7,3,4,2,18,2,54,-21,8,-5,NA)
Result.mean <-mean(x)
print(Result.mean)
```

Output- [1] 7.636364

Output-
[1] 4

Output-
[1] 2

Output-
[1] "it"

Page no.:
Date:

result.mean <-mean(x, na.rm = TRUE)
print(result.mean)

⇒ for median—

q) Find the median of 12, 7, 3, 4, 2, 18, 2, 54, -21, 8, -5.

Code-

```
x <-c(12,7,3,4,2,18,2,54,-21,8,-5)
median.result <- median(x)
print(median.result)
```

⇒ for mode—

q) Find the mode of 2, 1, 2, 3, 1, 2, 3, 4, 1, 5, 5, 3, 2, 2, 3.

Code-

```
getmode <- function(v){
uniqv <- unique(v)
uniqv[which.max(tabulate(match(v, uniqv)))]
}
v<-c(2,1,2,3,1,2,3,4,1,5,5,3,2,2,3)
result <-getmode(v)
print(result)

B<-c("o","it","the","it","it")
result <-getmode(B)
print(result)
```

Output-

[1] 3.5

Output-

[1] 1.2.2915

Page no.: ____/
Date: ____/____/____/

⇒ for Range -

Q. Find the range of the eruption duration in the data set faithful.

code-

```
duration = faithful $eruption
max(duration) - min(duration)
```

⇒ for Interquartile Range -

Q. Find the interquartile range of eruption duration in the data set faithful.

code-

```
duration = faithful $eruption
IQR(duration)
```