Hindi Vidya Prachar Samiti's

RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ARTS, SCIENCE & COMMERCE (EMPOWERED AUTONOMUS)

Advanced Data Analytics

Name: Chetan Raut

Roll No.: 10414

Class: MSc Data Science and Artificial Intelligence Part-II

Ramniranjan Jhunjhunwala College of Arts, Science and Commerce

Department of Data Science and Artificial Intelligence

CERTIFICATE

This is to certify **Chetan Raut** of MSc. Data Science and Artificial Intelligence, **Roll No. 10414** has successfully completed the practical of **ADVANCED DATA ANALYTICS** during the AcademicYear 2024-2025.

Date:

Prof. Rahul Yadav
(Prof-in-charge)

External Examiner

INDEX

Prac. No	Practical Name	Date	Signature
1.	Learn all the basics of R-Programming (Data types, Variables Operators etc.)	03/07/2024	
2.	Implement R-Loops with different examples.	08/07/2024	
3.	Learn the basics of functions in R and implement them with examples.	10/07/2024	
4.	Implement data frames in R. Write a program to join columns and rows in a data frame using c bind()and r bind() in R.	15/07/2024	
5.	Implement different String Manipulation functions in R	16/07/2024	
6.	Implement different data structures in R (Vectors, Lists, Data Frames)	16/07/2024	
7.	Write a program to read a csv file and analyze the data in the file in R	17/07/2024	
8.	Create pie charts and bar charts using R.	22/07/2024	
9.	Calculate Summary Statistics in Excel	06/08/2024	
10.	Create Graphs in Excel	07/08/2024	
11.	Generate Comparative Statistics in Excel	12/08/2024	
12.	Tabulation, bar diagram, Multiple Bar diagram, Pie diagram, **Measure of central tendency:** Mean, median, mode, **Measure of dispersion:** variance, standard deviation, Coefficient of variation. Correlation, regression lines.	13/08/2024	
13.	t-test, F-test, ANOVA one way classification, chi square test, independence of attributes.	13/08/2024	
14.	**Time Series Forecasting:** Method of least squares, moving average method. Inference and discussion of results.	14/08/2024	
15.	Advanced Data Analysis using PivotTables and Pivot Charts	03/09/2024	

PRACTICAL NO.: 01
Date: 03/07/2024

AIM: Learn all the basics of R-Programming (Data types, Variables Operators etc.)

DATA TYPES
> a=TRUE
> class(a)
[1] "logical"

> a=FALSE
> class(a)
[1] "logical"

> b=c(12,25.5,3)
> class(b)
[1] "numeric"
> b=c(12.99999,25.5,3)
> class(b)
[1] "numeric"
> b=c(-12.99999,25.5,-3)
> class(b)
[1] "numeric"

> d=c(24L,55L)
> class(d)
[1] "integer"
> d=c(24L,55L,-45L)
> class(d)
[1] "integer"

> d=2+3i
> class(d)
[1] "complex"
> d=i
> class(d)
[1] "character"
> d = 1i
> class(d)
[1] "complex"

> c="e"
> class(c)
[1] "character"
> c=c("a","b","c")
> class(c)
[1] "character"

OPERATORS
> x=c(2,4,5,6)
> y=c(2,3,9,8)
> x+y
[1] 4 7 14 14
> x-y

```
[1] 0 1 -4 -2
> x*y
[1] 4 12 45 48
> x/y
[1] 1.0000000 1.3333333 0.5555556 0.7500000
> x%/%y
[1] 1 1 0 0
> x^y
[1] 4 64 1953125 1679616
> x>y
[1] FALSE TRUE FALSE FALSE
> x<y
[1] FALSE FALSE TRUE TRUE
> x==y
[1] TRUE FALSE FALSE FALSE
> x=c(10,20,30,40)
> y=c(20,20,30,50)
> x<=y
[1] TRUE TRUE TRUE TRUE
> x>=y
[1] FALSE TRUE TRUE FALSE
> x!=y
[1] TRUE FALSE FALSE TRUE

> thismatrix <- matrix(c("apple", "banana", "cherry", "orange", "mango", "pineapple"), nrow = 3,
ncol = 2)
> thismatrix
[,1] [,2]
[1,] "apple" "orange"
[2,] "banana" "mango"
[3,] "cherry" "pineapple"
> thismatrix <- thismatrix[-c(1), -c(1)] # exclude first row and first column
> print(thismatrix)
[1] "mango" "pineapple"

> x= c(1,2+3i, TRUE,-10)
> y= c(2,4+2i, FALSE,-20)
> x&y
[1] TRUE TRUE FALSE TRUE
> x=c(1,2+3i,TRUE,-10L)
> y=c(2,4+2i,FALSE,-20)
> x & y
[1] TRUE TRUE FALSE TRUE
> x=-10L
> class(x)
[1] "integer"
> x=-20
> class(x)
[1] "numeric"

> x=c(3,0,TRUE,-10L)
> y=c(0,0,FALSE,-20+2i)
> x | y
[1] TRUE FALSE TRUE TRUE
> x=c(3,0,TRUE,-10L)
> !x
```

```
[1] FALSE TRUE FALSE FALSE
> x=c(TRUE,FALSE,10,2+3i)
> y=c(,FALSE,10,2+3i)
> y=c(1,2,TRUE,10)
> x[1] && y[1]
[1] TRUE
> x[2] && y[2]
[1] FALSE
> x[0] && y[0]
[1] NA
> x[1] && y[2]
[1] TRUE
> x=c(TRUE,FALSE,10,2+3i)
> y=c(1,2,TRUE,10)
> x[1] & y[1]
[1] TRUE
> x[2] || y[2]
[1] TRUE

> x<-c(2,3,4)
> x
[1] 2 3 4
> x<<-c(5,3,4)
> x
[1] 5 3 4
> x=c(5,8,4)
> x
[1] 5 8 4
> c(5,8,4) -> x
> x
[1] 5 8 4
> c(5,8,4) ->> x
> x
[1] 5 8 4
```

PRACTICAL NO.: 02

Date: 08/07/2024

AIM: Implement R-Loops with different examples.

```
> # R programming for loop
> for (val 1:5)
Error: unexpected numeric constant in "for (val 1"
> for (val in 1:5)
+ {
+ print(val)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
> week <- c('sunday','monday','tuesday','wednesday','thursday','friday','saturday')
> for (day in week)
+ { print(day)
+ }
[1] "sunday"
[1] "monday"
[1] "tuesday"
[1] "wednesday"
[1] "thursday"
[1] "friday"
[1] "saturday"
>
>
> my_matrix<-matrix(1:9,nrow=3)
> my_matrix
[,1] [,2] [,3]
[1,] 1 4 7
[2,] 2 5 8
[3,] 3 6 9
> for ( i in seq_len(nrow(my_matrix)))
+ {print(i)}
[1] 1
[1] 2
[1] 3
> for ( j in seq_len(ncol(my_matrix)))
+ {print(j)}
[1] 1
[1] 2
[1] 3
> for ( i in seq_len(nrow(my_matrix)))
+ {
+ for ( j in seq_len(ncol(my_matrix)))
+ {
+ current_element <- my_matrix[i,j]
+ print(paste('the element is:',current_element))
+ }
+ }
```

[1] "the element is: 1"
[1] "the element is: 4"
[1] "the element is: 7"
[1] "the element is: 2"
[1] "the element is: 5"
[1] "the element is: 8"
[1] "the element is: 3"
[1] "the element is: 6"
[1] "the element is: 9"

```
> # fibonaaci series
> print_fibonacci <- function(n) {
+ a <- 0
+ b <- 1
+ cat("Fibonacci Sequence:")
+ for ( i in 1:n){
+ cat(a," ")
+ next_num <- a+b
+ a <- b
+ b<-next_num
+ }
+ }
> number <- 10
> print(number)
```

[1] 10

```
> print_fibonacci(number)
```

Fibonacci Sequence:0 1 1 2 3 5 8 13 21 34

```
> my_dataframe <- data.frame(
+ Name = c("Joy", "Juliya", "Boby", "Marry"),
+ Age = c(40, 25, 19, 55),
+ Gender = c("M", "F", "M", "F")
+ )
> print(my_dataframe)
```

Name Age Gender
1 Joy 40 M
2 Juliya 25 F
3 Boby 19 M
4 Marry 55 F

```
> for (i in seq_len(nrow(my_dataframe))) {
+ current_row <- my_dataframe[i, ]
+ print(paste("The current row is:", toString(current_row)))
+ }
```

[1] "The current row is: Joy, 40, M"
[1] "The current row is: Juliya, 25, F"
[1] "The current row is: Boby, 19, M"
[1] "The current row is: Marry, 55, F"

```
> #while loop factorial
> n <- 5
>
> # assigning the factorial variable
> # and iteration variable to 1
> factorial <- 1
> i <- 1
```

```
>
> # using while loop
> while (i <= n)
+ {
+
+ # multiplying the factorial variable
+ # with the iteration variable
+ factorial = factorial * i
+
+ # incrementing the iteration variable
+ i = i + 1
+ }
>
> # displaying the factorial
> print(factorial)
[1] 120

> print_fibonacci_series <- function(limit) {
+ a <- 0
+ b <- 1
+
+
+ cat(a, b, sep = " ")
+
+ # Iterate while the next Fibonacci number is less than or equal to the limit
+ while (b <= limit) {
+ # Calculate the next Fibonacci number
+ next_num <- a + b
+
+ # Print the next Fibonacci number
+ cat(" ", next_num)
+
+ # Update variables for the next iteration
+ a <- b
+ b <- next_num
+ }
+ }
>
> # Test the function with a limit of 100
> limit <- 100
> cat("Fibonacci series up to", limit, "is: ")
Fibonacci series up to 100 is: > print_fibonacci_series(limit)
0 1 1 2 3 5 8 13 21 34 55 89 144>
> print_fibonacci_series <- function(limit) {
+ a <- 0
+ b <- 1
+
+
+ cat(a, b, sep = " ")
+
+ # Iterate while the next Fibonacci number is less than or equal to the limit
+ while (b <= limit) {
+ # Calculate the next Fibonacci number
+ next_num <- a + b
+
+ # Print the next Fibonacci number
```

```
+ cat(" ", next_num)
+
+ # Update variables for the next iteration
+ a <- b
+ b <- next_num
+ }
+ }
>
> # Test the function with a limit of 100
> limit <- 100
> cat("Fibonacci series up to", limit, "is: ")
Fibonacci series up to 100 is: >
> print_fibonacci_series <- function(bound) {
+ a <- 0
+ b <- 1
+
+ cat(a, b, sep = " ")
+
+ while (b <= bound) {
+ next_num <- a + b
+
+ cat(" ", next_num)
+
+ # Update variables for the next iteration
+ a <- b
+ b <- next_num
+ }
+ }
>
> bound <- 200
> cat("Fibonacci series up to", bound, "is: ")
Fibonacci series up to 200 is: > print_fibonacci_series(bound)
0 1 1 2 3 5 8 13 21 34 55 89 144 233>
>
> #repeat loop
> val = 1
> repeat
+ {
+ print(val)
+ val = val + 1
+
+ if(val > 5)
+ {
+ break
+ }
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
>
> #repeat 5 times using repeat loop
> i<-0
> repeat
+ {
```

```
+ print("print")
+ i=i+1
+ if (i==5)
+ {
+ break
+ }
+ }
[1] "print"
[1] "print"
[1] "print"
[1] "print"
[1] "print"
>
> i<-0
> repeat
+ {
+ print("print")
+ i=i+1
+ if (i<=5)
+ {
+ break
+ }
+ }
[1] "print"
```

PRACTICAL NO.: 03
Date: 10/07/2024

AIM: Learn the basics of functions in R and implement them with examples.

```
> #sort function
> x=c(10,20,35,47,57,66)
> x
[1] 10 20 35 47 57 66
> sort(x)
[1] 10 20 35 47 57 66
> sort(x,decreasing=T)
[1] 66 57 47 35 20 10
> sort(x,decreasing=F)
[1] 10 20 35 47 57 66
> x=scan()
1: 10 20 30 40 10 50 20 70 30 50
11: x
> x
[1] 10 20 35 47 57 66
> x=scan()
1: 10 20 30 40 50 20 30 10
9:
Read 8 items
> x
[1] 10 20 30 40 50 20 30 10
> unique(x)
[1] 10 20 30 40 50

> x=c(10.234567,25.37580,35.67843)
> round(x,2)
[1] 10.23 25.38 35.68
> x=-10:5
> abs(x)
[1] 10 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5
> x=-10:15
> abs(x)
[1] 10 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
>
> x=1:20
> fivenum(x)
[1] 1.0 5.5 10.5 15.5 20.0
> summary(x)
Min. 1st Qu. Median Mean 3rd Qu. Max.
1.00 5.75 10.50 10.50 15.25 20.00
> length(x)
[1] 20
> max(x)
[1] 20
> min(x)
[1] 1
> range(x)
[1] 1 20
> sum(x)
[1] 210
```

```
> cumsum(x)
[1] 1 3 6 10 15 21 28 36 45 55 66 78 91 105 120 136 153 171 190 210
> mean(x)
[1] 10.5
> median(x)
[1] 10.5
> var(x)
[1] 35
> diff(x)
[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

PRACTICAL NO.: 04
Date: 15/07/2024

AIM: Implement data frames in R. Write a program to join columns and rows in a data frame using c bind()and r bind() in R.

>data=data.frame(Subjects=c("java","python","IT"),id=c(1,2,3),names=c("Rohini","Ram","Raj"),marks=c(75,80,89)
+)
> data
Subjects id names marks
1 java 1 Rohini 75
2 python 2 Ram 80
3 IT 3 Raj 89

> aggregate(data$marks,list(data$Subjects),FUN=sum)
Group.1 x
1 IT 89
2 java 75
3 python 80
> data=data.frame(Subjects=c("java","python","IT","python"),id=c(1,2,3,4),names=c("Rohini","Ram","Raj","Soham"),marks=c(75,80,89,75)
+)
> aggregate(data$marks,list(data$Subjects),FUN=sum)
Group.1 x
1 IT 89
2 java 75
3 python 155
> aggregate(data$marks,list(data$Subjects),FUN=min)
Group.1 x
1 IT 89
2 java 75
3 python 75
> aggregate(data$marks,list(data$Subjects),FUN=max)
Group.1 x
1 IT 89
2 java 75

3 python 80
> matrix=matrix(c(1:12,nrow=4,byrow=TRUE)
+)
> matrix=matrix(c(1:12),nrow=4,byrow=TRUE)
> matrix
[,1] [,2] [,3]
[1,] 1 2 3
[2,] 4 5 6
[3,] 7 8 9
[4,] 10 11 12
> t(matrix)
[,1] [,2] [,3] [,4]
[1,] 1 4 7 10
[2,] 2 5 8 11
[3,] 3 6 9 12
> dim(matrix)
[1] 4 3

> data
Subjects id names marks
1 java 1 Rohini 75
2 python 2 Ram 80
3 IT 3 Raj 89
4 python 4 Soham 75
> info=cbind(data)
> info
Subjects id names marks
1 java 1 Rohini 75
2 python 2 Ram 80
3 IT 3 Raj 89
4 python 4 Soham 75
> info=cbind(data$id,data)
> info
data$id Subjects id names marks
1 1 java 1 Rohini 75
2 2 python 2 Ram 80
3 3 IT 3 Raj 89
4 4 python 4 Soham 75
> info=cbind(data$id,data$names,data$Subjects,data$marks)
> info
[,1] [,2] [,3] [,4]
[1,] "1" "Rohini" "java" "75"
[2,] "2" "Ram" "python" "80"
[3,] "3" "Raj" "IT" "89"
[4,] "4" "Soham" "python" "75"
> matrix2=matrix(c(13:18),nrow=2,byrow=TRUE)
> matrix2
[,1] [,2] [,3]
[1,] 13 14 15
[2,] 16 17 18
> info=cbind(matrix,matrix2)
Error in cbind(matrix, matrix2) :
number of rows of matrices must match (see arg 2)
> matrix1=matrix(c(1:6),nrow=2,byrow=TRUE)
> info=cbind(matrix1,matrix2)
> info
[,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1 2 3 13 14 15
[2,] 4 5 6 16 17 18
> info=rbind(matrix1,matrix2)
> info
[,1] [,2] [,3]
[1,] 1 2 3
[2,] 4 5 6
[3,] 13 14 15
[4,] 16 17 18
> data
Subjects id names marks
1 java 1 Rohini 75
2 python 2 Ram 80
3 IT 3 Raj 89
4 python 4 Soham 75
> data2=data.frame(Subjects=c("data science","AI"),id=c(5,6),names=c("Ashiwni","Rohan"),marks=c(7
0,50))

> merge(data,data2)
[1] Subjects id names marks
<0 rows> (or 0-length row.names)
> merge(data,data2,all=TRUE)
Subjects id names marks
1 AI 6 Rohan 50
2 data science 5 Ashiwni 70
3 IT 3 Raj 89
4 java 1 Rohini 75
5 python 2 Ram 80
6 python 4 Soham 75
> merge(data,data2,all=FALSE)
[1] Subjects id names marks
<0 rows> (or 0-length row.names)
> merge(data,data2,all=FALSE)
[1] Subjects id names marks
<0 rows> (or 0-length row.names)
> merge(data,data2,all=TRUE)
Subjects id names marks
1 AI 6 Rohan 50
2 data science 5 Ashiwni 70
3 IT 3 Raj 89
4 java 1 Rohini 75
5 python 2 Ram 80
6 python 4 Soham 75
> merge(data2,data,all=TRUE)
Subjects id names marks
1 AI 6 Rohan 50
2 data science 5 Ashiwni 70
3 IT 3 Raj 89
4 java 1 Rohini 75
5 python 2 Ram 80
6 python 4 Soham 75
> data
Subjects id names marks
1 java 1 Rohini 75
2 python 2 Ram 80
3 IT 3 Raj 89
4 python 4 Soham 75
> data2
Subjects id names marks
1 data science 5 Ashiwni 70
2 AI 6 Rohan 50
> rbind(data,data2)
Subjects id names marks
1 java 1 Rohini 75
2 python 2 Ram 80
3 IT 3 Raj 89
4 python 4 Soham 75
5 data science 5 Ashiwni 70
6 AI 6 Rohan 50
> merge(matrix1,matrix2)
[1] V1 V2 V3
<0 rows> (or 0-length row.names)
> merge(matrix1,matrix2,all=TRUE)
V1 V2 V3

1 1 2 3
2 4 5 6
3 13 14 15
4 16 17 18
> rose=c("red","yellow","blue","white")
> factor(rose)
[1] red yellow blue white
Levels: blue red white yellow
> rose=c("red","yellow","blue","white","white")
> factor(rose)
[1] red yellow blue white white

Levels: blue red white yellow

PRACTICAL NO.: 05
Date: 16/07/2024

AIM: Implement different String Manipulation functions in R

> str="AI is growing day by day"
> nchar(str)
[1] 24

> tolower(str)
[1] "ai is growing day by day"

> toupper(str)
[1] "AI IS GROWING DAY BY DAY"

> a= "Congratulation"
> b= "on your joining"
> paste(a,b)
[1] "Congratulation on your joining"

> paste(a,b,sep='-')
[1] "Congratulation-on your joining"

> chartr("DS","AI",str)
[1] "AI is growing day by day"

> chartr("AI","DS",str)
[1] "DS is growing day by day"

> substring("Pretty",2,3)
[1] "re"

> substring("Pretty",1,5)
[1] "Prett"

PRACTICAL NO.: 06
Date: 16/07/2024

AIM: Implement different data structures in R (Vectors, Lists, Data Frames)

```
> x = c(1,3,5,7,9)
> y = c(2,4,6,8,0)
>
> x+y
[1]  3  7 11 15  9
> x-y
[1] -1 -1 -1 -1  9
>
> class(x)
[1] "numeric"
> type of (y)
Error: unexpected symbol in "type of"
>
> class(x)
[1] "numeric"
> class(y)
[1] "numeric"
>
> a = c(1L,3L,5L)
>
> typeof(x)
[1] "double"
> typeof(y)
[1] "double"
>
> typeof(a)
[1] "integer"
>
> class(a)
[1] "integer"
>
> which(x == 5)
[1] 3
> which(y == 4)
[1] 2
>
> # list
> a = list(student_name = c('X','Y','Z'), subject = c('Maths', 'Stats', 'Physics'), marks = c(60, 70, 80))
> a
$student_name
[1] "X" "Y" "Z"

$subject
[1] "Maths"  "Stats"   "Physics"

$marks
[1] 60 70 80

> a[student_name]
```

```
Error: object 'student_name' not found
> a['student_name']
$student_name
[1] "X" "Y" "Z"

> a['student_name'][1]
$student_name
[1] "X" "Y" "Z"

> a['student_name', 1]
Error in a["student_name", 1] : incorrect number of dimensions
> print(a$student_name[1])
[1] "X"
>
> a['student_name']
$student_name
[1] "X" "Y" "Z"

> print(a$student_name[1])
[1] "X"
>
> a[[1]][4] = 'A'
> a[[2]][4] = 'Chem'
> a[[3]][4] = '50'
> a
$student_name
[1] "X" "Y" "Z" "A"

$subject
[1] "Maths"  "Stats"   "Physics" "Chem"

$marks
[1] "60" "70" "80" "50"

> # converting list to vector
> unlist(a)
student_name1 student_name2 student_name3 student_name4      subject1      subject2      subject3
subject4       marks1       marks2
          "X"           "Y"           "Z"           "A"       "Maths"       "Stats"     "Physics"       "Chem"          "60"
"70"
       marks3       marks4
         "80"          "50"
> as.vector(unlist(a))
 [1] "X"       "Y"       "Z"       "A"       "Maths"   "Stats"   "Physics" "Chem"    "60"      "70"      "80"      "50"
>
> df = data.frame(student_name = c('X','Y','Z'), subject = c('Maths', 'Stats', 'Physics'), marks = c(60, 70,
80))
> df
  student_name subject marks
1            X   Maths    60
2            Y   Stats    70
3            Z Physics    80
>
> print(data.frame(df$subject)
+ )
  df.subject
```

```
1      Maths
2      Stats
3    Physics

> print(data.frame(df$subject))
  df.subject

1      Maths
2      Stats
3    Physics

> summary(df)
 student_name         subject              marks
 Length:3           Length:3           Min.   :60
 Class :character   Class :character   1st Qu.:65
 Mode  :character   Mode  :character   Median :70
                                       Mean   :70
                                       3rd Qu.:75
                                       Max.   :80
> dim(df)
```
[1] 3 3

PRACTICAL NO.: 07

Date: 17/07/2024

AIM: Write a program to read a csv file and analyze the data in the file in R

> is.null(data)
[1] FALSE
> summary(data)
PassengerId Survived Pclass Name Sex Age
SibSp Parch Ticket Fare Cabin
Min. : 1.0 Min. :0.0000 Min. :1.000 Length:891 Length:891 Min. :
0.42 Min. :0.000 Min. :0.0000 Length:891 Min. : 0.00 Length:891
1st Qu.:223.5 1st Qu.:0.0000 1st Qu.:2.000 Class :character Class :character 1st Qu.:2
0.12 1st Qu.:0.000 1st Qu.:0.0000 Class : character 1st Qu.: 7.91 Class :character
Median :446.0 Median :0.0000 Median :3.000 Mode :character Mode :character Median :2
8.00 Median :0.000 Median :0.0000 Mode :character Median : 14.45 Mode :character
Mean :446.0 Mean :0.3838 Mean :2.309 Mean :2
9.70 Mean :0.523 Mean :0.3816 Mean : 32.20
3rd Qu.:668.5 3rd Qu.:1.0000 3rd Qu.:3.000 3rd Qu.:3
8.00 3rd Qu.:1.000 3rd Qu.:0.0000 3rd Qu.: 31.00
Max. :891.0 Max. :1.0000 Max. :3.000 Max. :8
0.00 Max. :8.000 Max. :6.0000 Max. :512.33

> head(data)
PassengerId Survived Pclass Name Sex Age SibS
p Parch Ticket Fare Cabin Embarked
1 1 0 3 Braund, Mr. Owen Harris male 22
1 0 A/5 21171 7.2500 S
2 2 1 1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female 38
1 0 PC 17599 71.2833 C85 C
3 3 1 3 Heikkinen, Miss. Laina female 26
0 0 STON/O2. 3101282 7.9250 S
4 4 1 1 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35
1 0 113803 53.1000 C123 S
5 5 0 3 Allen, Mr. William Henry male 35
0 0 373450 8.0500 S
6 6 0 3 Moran, Mr. James male NA
0 0 330877 8.4583 Q

> tail(data)
PassengerId Survived Pclass Name Sex Age SibSp Parch
Ticket Fare Cabin Embarked
886 886 0 3 Rice, Mrs. William (Margaret Norton) female 39 0 5
382652 29.125 Q
887 887 0 2 Montvila, Rev. Juozas male 27 0 0
211536 13.000 S
888 888 1 1 Graham, Miss. Margaret Edith female 19 0 0
112053 30.000 B42 S
889 889 0 3 Johnston, Miss. Catherine Helen "Carrie" female NA 1 2 W
./C. 6607 23.450 S
890 890 1 1 Behr, Mr. Karl Howell male 26 0 0
111369 30.000 C148 C
891 891 0 3 Dooley, Mr. Patrick male 32 0 0

> typeof(data$"Survived")
[1] "integer"

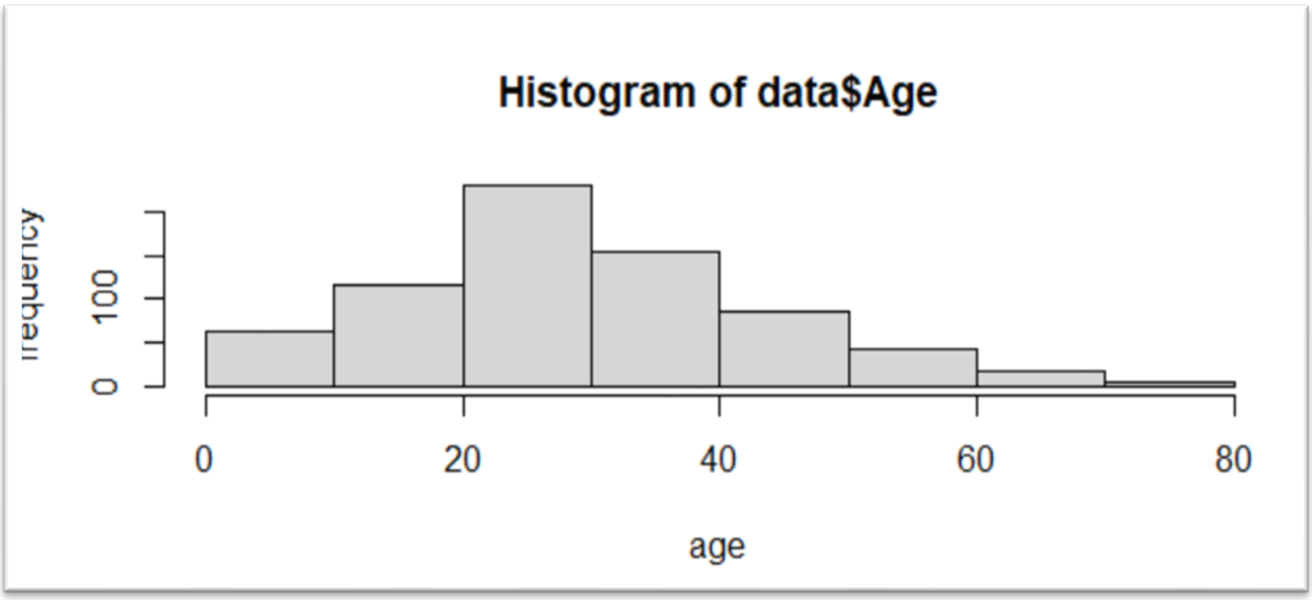
> sapply(data,class)
PassengerId Survived Pclass Name Sex Age SibSp Parch
Ticket Fare Cabin Embarked
"integer" "integer" "integer" "character" "character" "numeric" "integer" "integer" "
character" "numeric" "character" "character"

> sum(is.na(data))
[1] 177

> dim(data)
[1] 891 12
> dropnull_data = data[rowSums(is.na(data))<=0,]
> dim(data)
[1] 891 12
> dim(dropnull_data)
[1] 714 12
> df = na.omit(data)
> dim(df)
[1] 714 12

> ttable <- table(data$Survived)
> lbls <- paste(names(ttable), "\n", ttable, sep="")
> pie(ttable, labels = lbls, main="Pie Chart")
>

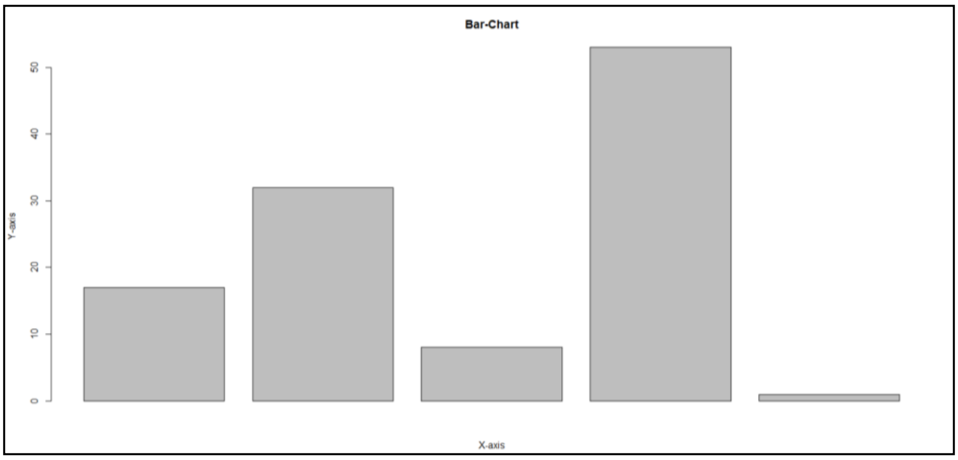
> hist(data$Age,xlab="gender",ylab="frequency")



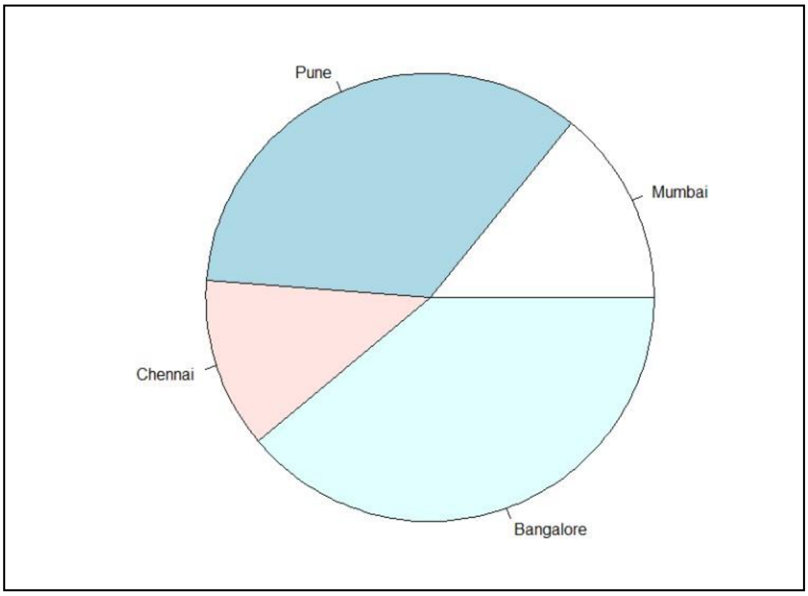
PRACTICAL NO.: 08
Date: 22/07/2024

AIM: Create pie charts and bar charts using R.

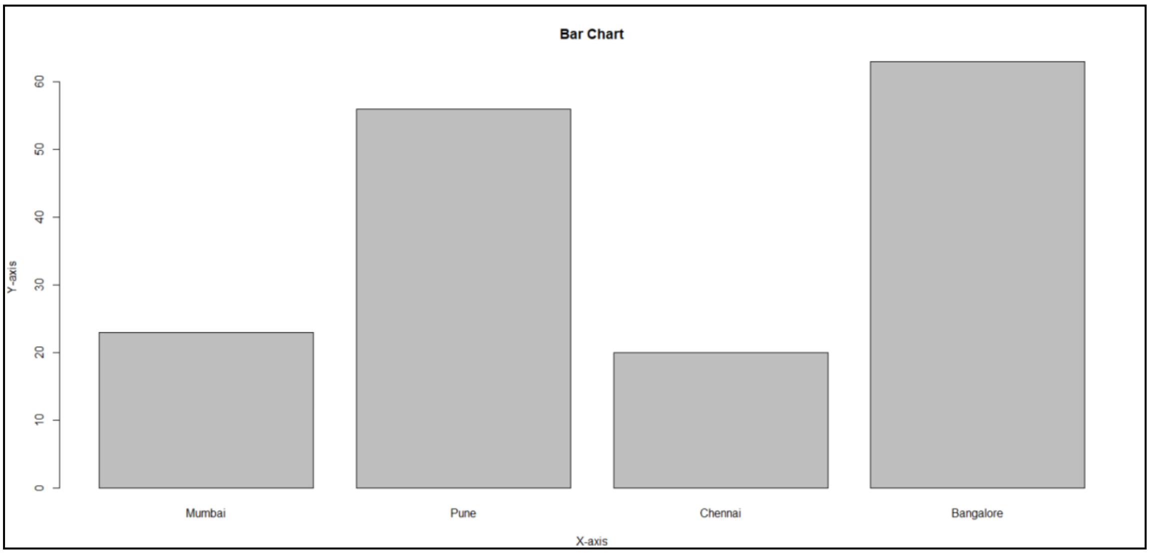
 # Create the data for the chart
> A <- c(17, 32, 8, 53, 1)
 # Plot the bar chart
> barplot(A, xlab = "X-axis", ylab = "Y-axis", main ="Bar-Chart")



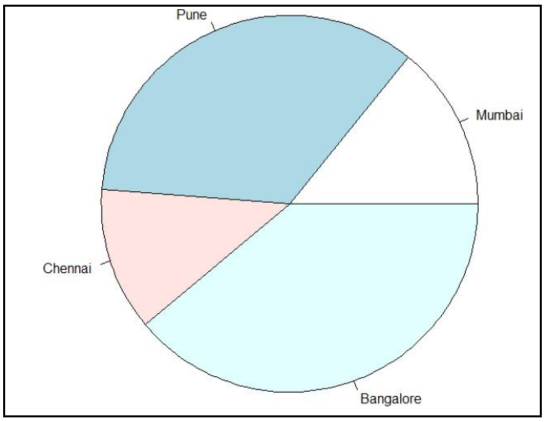
geeks<- c(23, 56, 20, 63)
> labels <- c("Mumbai", "Pune", "Chennai", "Bangalore")
 # Plot the chart.
> pie(geeks, labels)



> barplot(top$geeks, names.arg=top$labels, xlab = "X-axis" , ylab = "Y-axis", main="Bar Chart")



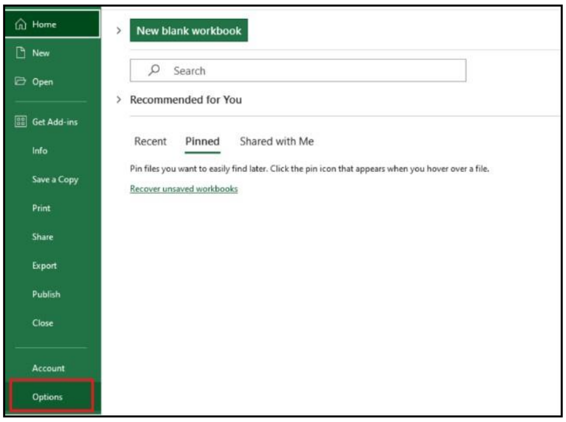
> pie(top$geeks, labels=labels)

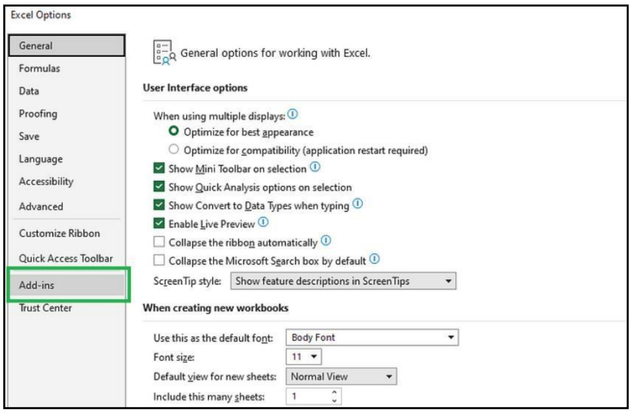


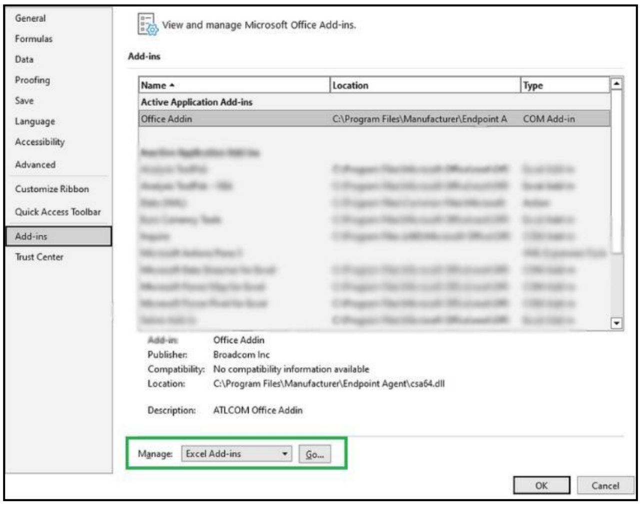
PRACTICAL NO.: 09
Date: 06/08/2024

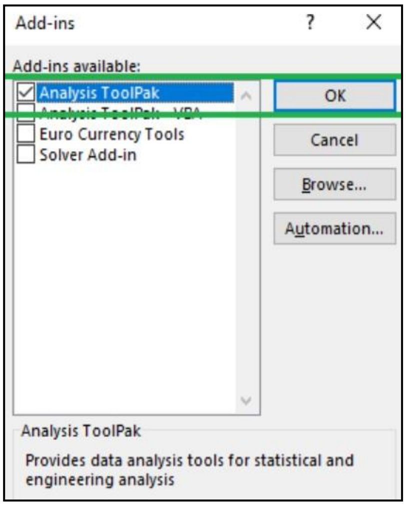
AIM: Calculate Summary Statistics in Excel

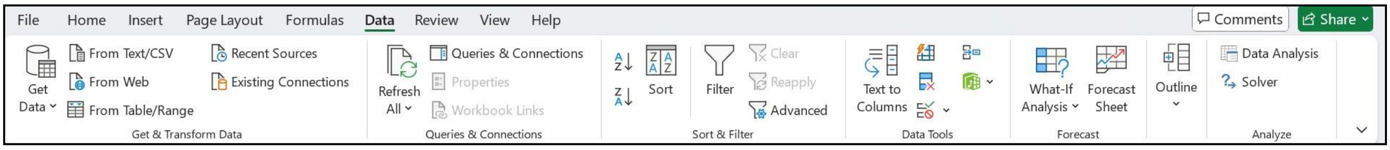
Steps to be followed to get “DATA ANALYSIS” add ons in the EXCEL









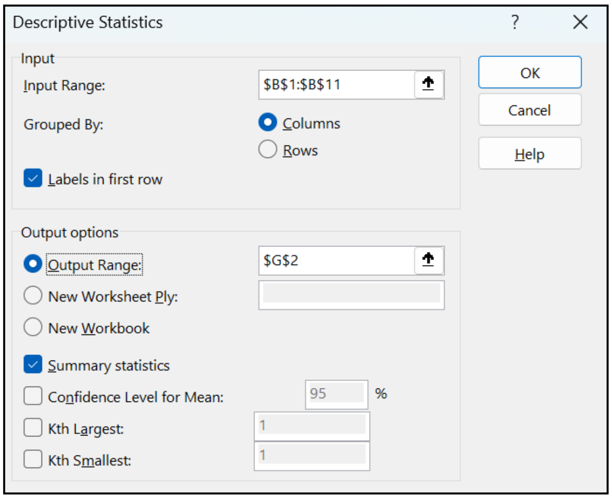


DATA

Product	Sale	Price	Revenue	Region
Electronics	150	100	15,000	North
Electronics	120	150	18,000	South
Furniture	80	200	16,000	East
Furniture	50	250	12,500	West
Clothing	200	50	10,000	North
Clothing	180	55	9,900	South
Electronics	110	120	13,200	East
Electronics	60	225	13,500	West
Furniture	220	45	9,900	North
Clothing	90	140	12,600	South

Summary Statistics
Steps to be followed:
Ribbon (Top bar) >> Data >> Data Analysis (Present in right top corner) >> Descriptive Statics

Select Input Range (as given below) > Select Output range >> Check the Label in first row box >> Check the Summary Statistics box >> Click on OK

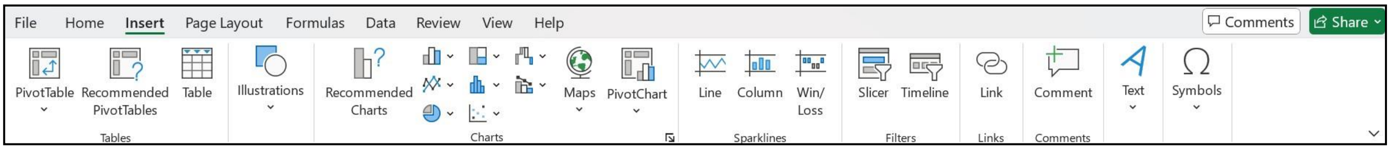


Sale	
Mean	126
Standard Error	18.7498148
Median	115
Mode	#N/A
Standard Deviation	59.2921205
Sample Variance	3515.55556
Kurtosis	-1.24083056
Skewness	0.35469032
Range	170
Minimum	50
Maximum	220
Sum	1260
Count	10

PRACTICAL NO.: 10

Date: 07/08/2024

AIM: Create graphs in Excel



Original data

Month	Profit
November	590
November	590
March	1366
March	1188
May	238
May	297
May	199
May	100
February	1096
February	1046
July	398
July	398
July	349
July	349
August	369
August	517
September	148
September	74
January	74
January	74
May	443

>> Copy and Paste Month column in a different cell from the original Data set

>>With the copied data selected, go to the Data tab and click Remove Duplicates.

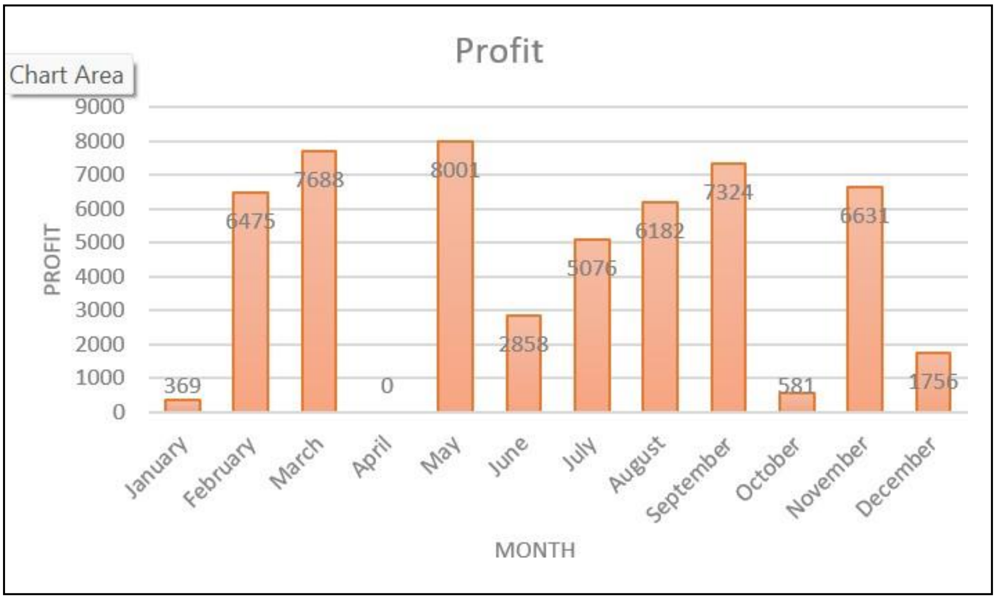
>>Next to each unique month, use the SUMIF function to calculate the total profit:

>>Original Month data is in column A, and Profit data is in column B, and unique months are in column D, enter the following formula in the cell next to the first unique month

Formula used : =SUMIF()

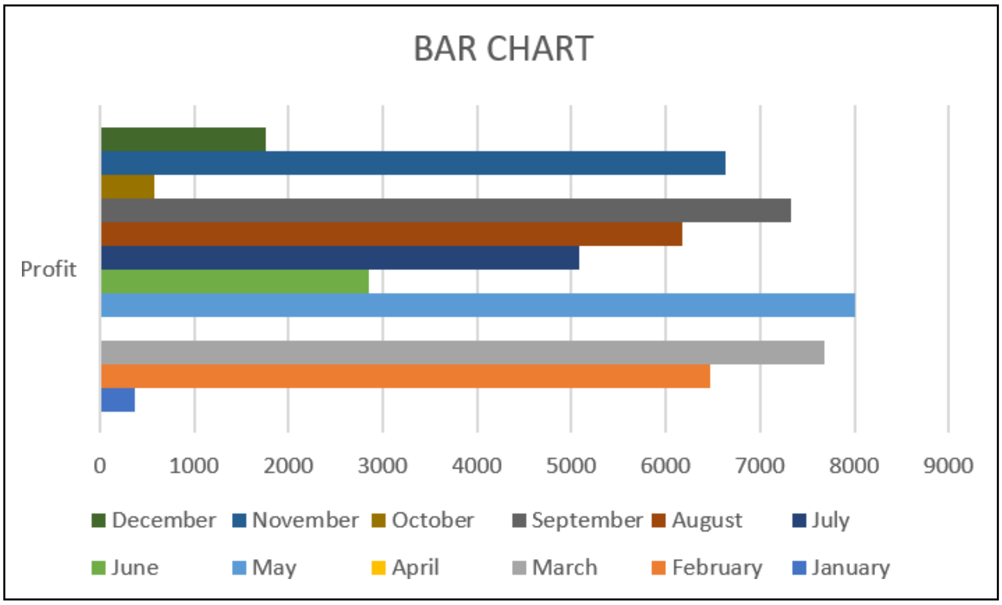
COLUMN CHART

Month	Profit
January	369
February	6475
March	7688
April	0
May	8001
June	2858
July	5076
August	6182
September	7324
October	581
November	6631
December	1756



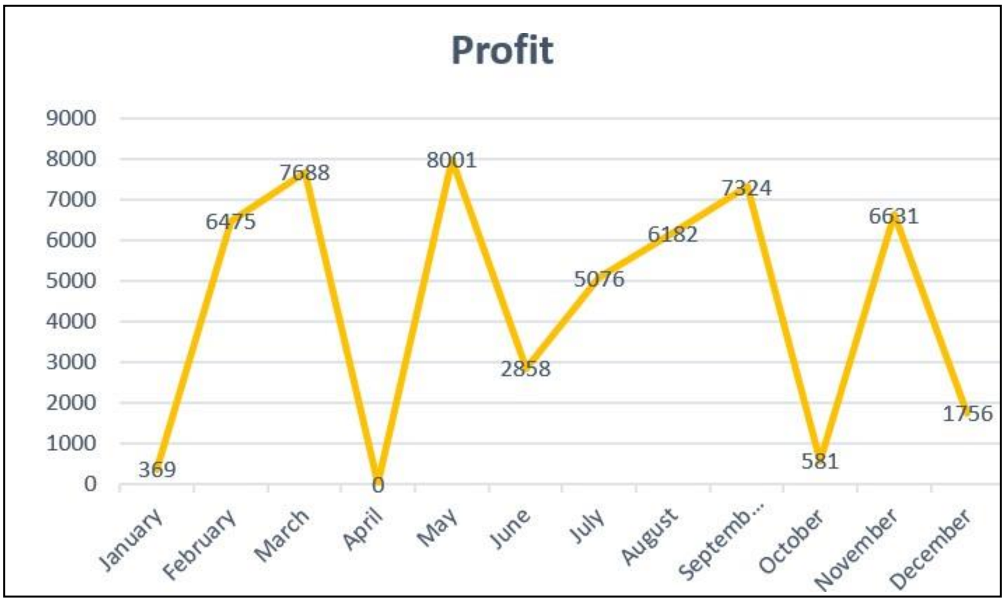
BAR CHART

Month	Profit
January	369
February	6475
March	7688
April	0
May	8001
June	2858
July	5076
August	6182
September	7324
October	581
November	6631
December	1756



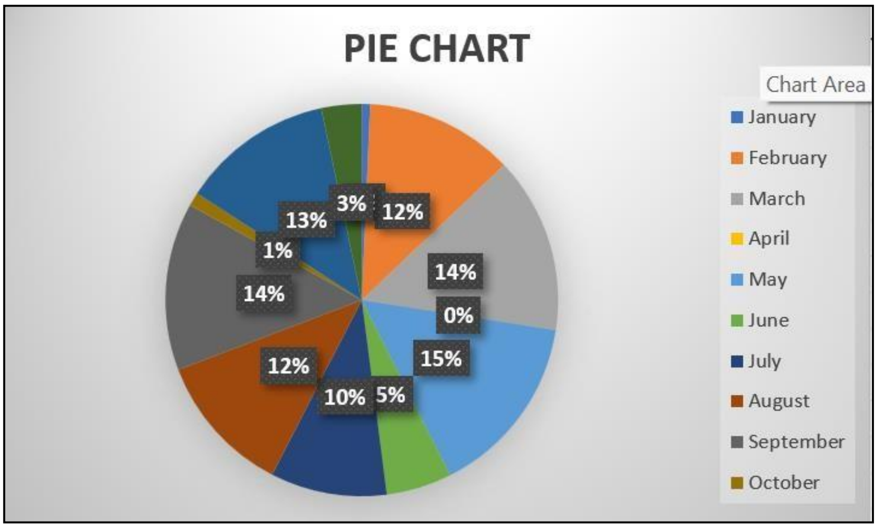
LINE CHART

Month	Profit
January	369
February	6475
March	7688
April	0
May	8001
June	2858
July	5076
August	6182
September	7324
October	581
November	6631
December	1756



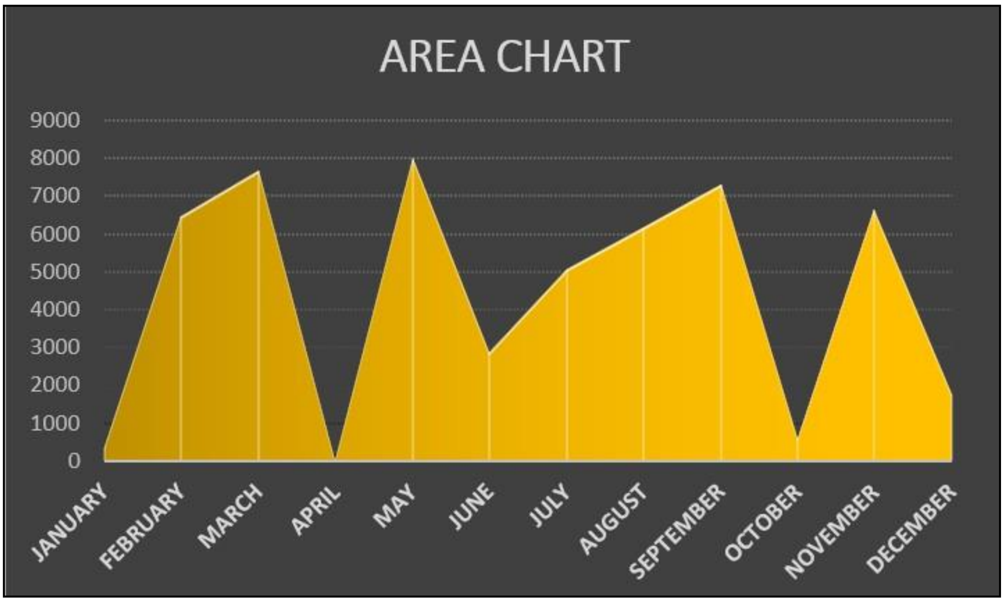
PIE CHART

Month	Profit
January	369
February	6475
March	7688
April	0
May	8001
June	2858
July	5076
August	6182
September	7324
October	581
November	6631
December	1756



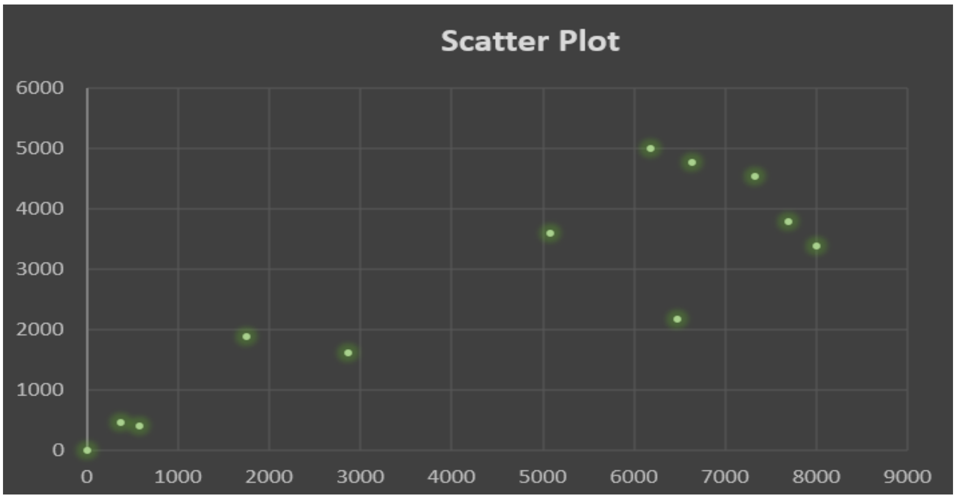
AREA CHART

Month	Profit
January	369
February	6475
March	7688
April	0
May	8001
June	2858
July	5076
August	6182
September	7324
October	581
November	6631
December	1756



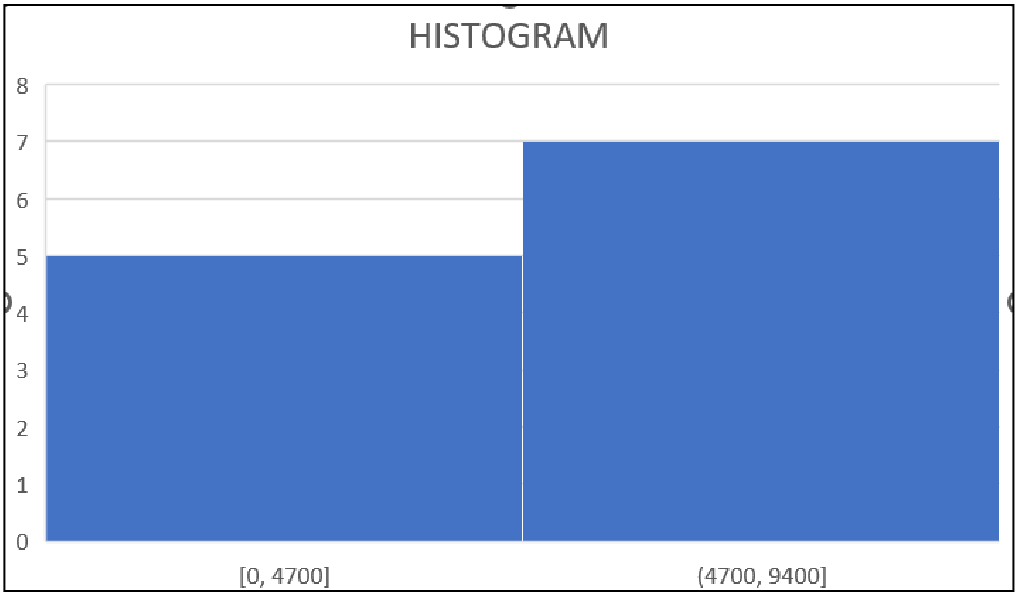
SCATTER PLOT

Month	Profit	Cost
January	369	450
February	6475	2160
March	7688	3780
April	0	0
May	8001	3375
June	2858	1620
July	5076	3600
August	6182	4995
September	7324	4545
October	581	405
November	6631	4770
December	1756	1890



HISTOGRAM

Month	Profit
January	369
February	6475
March	7688
April	0
May	8001
June	2858
July	5076
August	6182
September	7324
October	581
November	6631
December	1756



PRACTICAL NO.: 11

Date: 12/08/2024

AIM: GENERATE COMPARATIVE STATISTICS IN EXCEL

Steps to be followed for SUMMARY STATISTICS
Ribbon (Top bar) >> Data >> Data Analysis (Present in right top corner) >> Descriptive Statics

Select Input Range (as given below) > Select Output range >> Check the Label in first row box >> Check the Summary Statistics box >> Click on OK

Profit	Cost	Revenue
590	360	950
590	360	950
1366	1035	2401
1188	900	2088
238	180	418
297	225	522
199	180	379
100	90	190
1096	990	2086
1046	945	1991
398	360	758
398	360	758
349	315	664
349	315	664
369	225	594
517	315	832
148	90	238
74	45	119
74	45	119
74	45	119
443	270	713
590	360	950

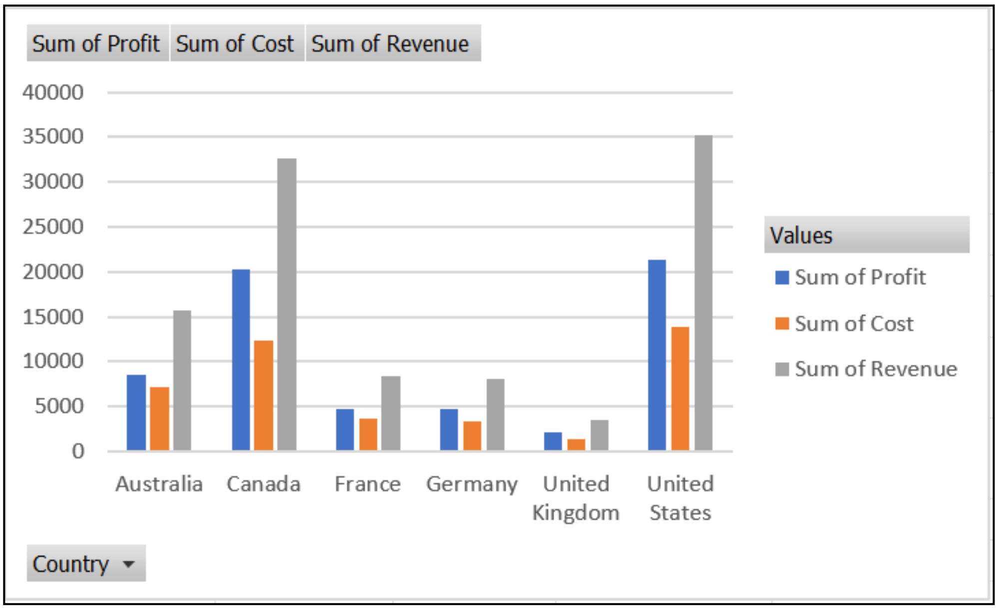
Profit		*Cost*		*Revenue*	
Mean	641.1538	Mean	440.7692	Mean	1081.923
Standard Error	86.82059	Standard Error	56.62206	Standard Error	142.5336
Median	443	Median	315	Median	758
Mode	74	Mode	360	Mode	119
Standard Deviation	542.1944	Standard Deviation	353.6046	Standard Deviation	890.1221
Sample Variance	293974.8	Sample Variance	125036.2	Sample Variance	792317.3
Kurtosis	-0.11165	Kurtosis	-0.76663	Kurtosis	-0.43829
Skewness	1.090334	Skewness	0.865558	Skewness	0.980914
Range	1771	Range	1080	Range	2851
Minimum	74	Minimum	45	Minimum	119
Maximum	1845	Maximum	1125	Maximum	2970
Sum	25005	Sum	17190	Sum	42195
Count	39	Count	39	Count	39

Using Pivot Table

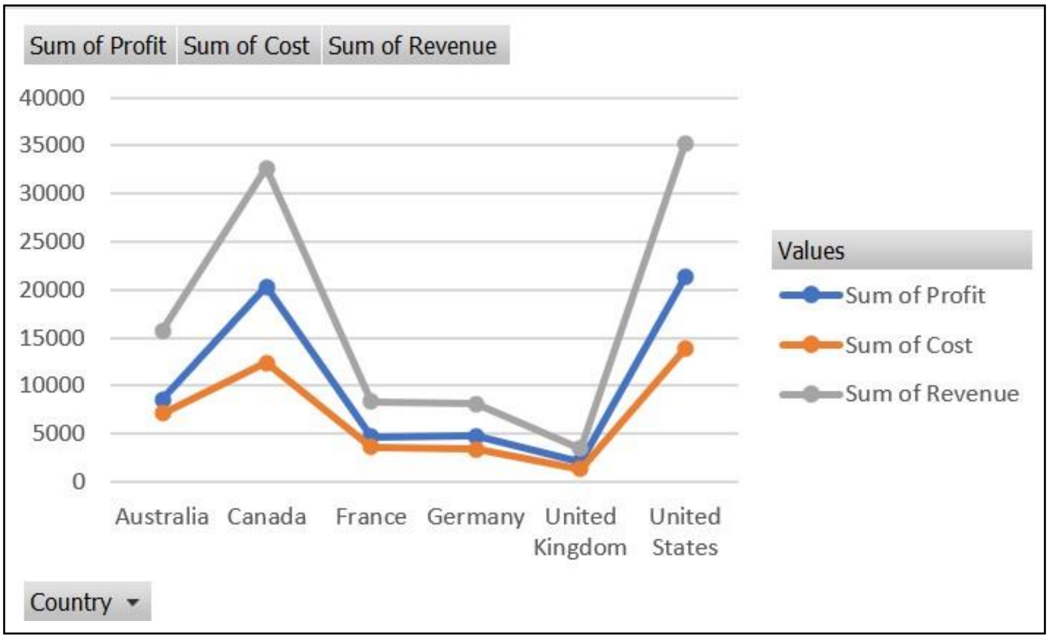
Country	Profit	Cost	Revenue
Canada	590	360	950
Canada	590	360	950
Australia	1366	1035	2401
Australia	1188	900	2088
Australia	238	180	418
Australia	297	225	522
Australia	199	180	379
Australia	100	90	190
Australia	1096	990	2086
Australia	1046	945	1991
Australia	398	360	758
Australia	398	360	758
Australia	349	315	664
Australia	349	315	664
Canada	369	225	594
Canada	517	315	832
Canada	148	90	238
Canada	74	45	119
Canada	74	45	119
Canada	74	45	119
Canada	443	270	713
Canada	590	360	950

Row Labels	Sum of Profit	Sum of Cost	Sum of Revenue
Australia	8587	7155	15742
Canada	20296	12375	32671
France	4721	3600	8321
Germany	4756	3375	8131
United Kingdom	2106	1350	3456
United States	21296	13860	35156
Grand Total	**61762**	**41715**	**103477**

COLUMN CHART



LINE CHART



PRACTICAL NO.: 12
Date: 13/08/2024

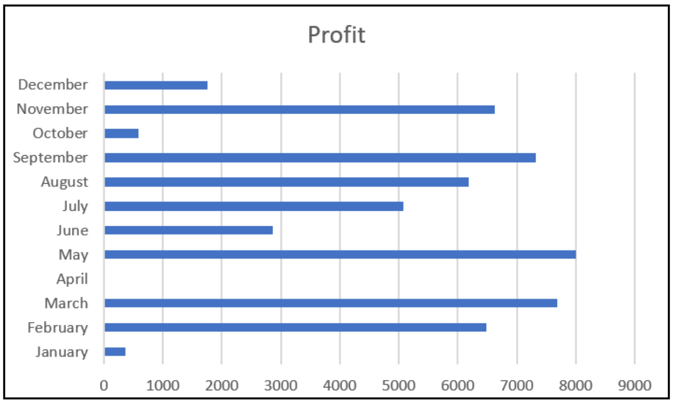
AIM: Tabulation, bar diagram, Multiple Bar diagram, Pie diagram, *Measure of central tendency:* Mean, median, mode, *Measure of dispersion:* variance, standard deviation, Coefficient of variation. Correlation, regression lines.

TABULATION

Date	Day	Month	Year	Customer_	Age_Grou	Customer_	Country	State	Product_C	Sub_Cate	Product	Order_Qu	Unit_Cost	Unit_Price	Profit
26/11/201:	26	November	2013	19	Youth (<2!	M	Canada	British Col	Accessori	Bike Rack	Hitch Rack - 4-Bike	8	45	120	590
26/11/201!	26	November	2015	19	Youth (<2!	M	Canada	British Col	Accessori	Bike Rack	Hitch Rack - 4-Bike	8	45	120	590
23/3/2014	23	March	2014	49	Adults (35	M	Australia	New Soutl	Accessori	Bike Rack	Hitch Rack - 4-Bike	23	45	120	1366
23/3/2016	23	March	2016	49	Adults (35	M	Australia	New Soutl	Accessori	Bike Rack	Hitch Rack - 4-Bike	20	45	120	1188
15/5/2014	15	May	2014	47	Adults (35	F	Australia	New Soutl	Accessori	Bike Rack	Hitch Rack - 4-Bike	4	45	120	238
15/5/2016	15	May	2016	47	Adults (35	F	Australia	New Soutl	Accessori	Bike Rack	Hitch Rack - 4-Bike	5	45	120	297
22/5/2014	22	May	2014	47	Adults (35	F	Australia	Victoria	Accessori	Bike Rack	Hitch Rack - 4-Bike	4	45	120	199
22/5/2016	22	May	2016	47	Adults (35	F	Australia	Victoria	Accessori	Bike Rack	Hitch Rack - 4-Bike	2	45	120	100
22/2/2014	22	February	2014	35	Adults (35	M	Australia	Victoria	Accessori	Bike Rack	Hitch Rack - 4-Bike	22	45	120	1096
22/2/2016	22	February	2016	35	Adults (35	M	Australia	Victoria	Accessori	Bike Rack	Hitch Rack - 4-Bike	21	45	120	1046
30/7/2013	30	July	2013	32	Young Adı	F	Australia	Victoria	Accessori	Bike Rack	Hitch Rack - 4-Bike	8	45	120	398
30/7/2015	30	July	2015	32	Young Adı	F	Australia	Victoria	Accessori	Bike Rack	Hitch Rack - 4-Bike	8	45	120	398
15/7/2013	15	July	2013	34	Young Adı	M	Australia	Victoria	Accessori	Bike Rack	Hitch Rack - 4-Bike	7	45	120	349
15/7/2015	15	July	2015	34	Young Adı	M	Australia	Victoria	Accessori	Bike Rack	Hitch Rack - 4-Bike	7	45	120	349
2/8/2013	2	August	2013	29	Young Adı	M	Canada	British Col	Accessori	Bike Rack	Hitch Rack - 4-Bike	5	45	120	369
2/8/2015	2	August	2015	29	Young Adı	M	Canada	British Col	Accessori	Bike Rack	Hitch Rack - 4-Bike	7	45	120	517
2/9/2013	2	Septembe	2013	29	Young Adı	M	Canada	British Col	Accessori	Bike Rack	Hitch Rack - 4-Bike	2	45	120	148
2/9/2015	2	Septembe	2015	29	Young Adı	M	Canada	British Col	Accessori	Bike Rack	Hitch Rack - 4-Bike	1	45	120	74
22/1/2014	22	January	2014	29	Young Adı	M	Canada	British Col	Accessori	Bike Rack	Hitch Rack - 4-Bike	1	45	120	74
22/1/2016	22	January	2016	29	Young Adı	M	Canada	British Col	Accessori	Bike Rack	Hitch Rack - 4-Bike	1	45	120	74
17/5/2014	17	May	2014	29	Young Adı	M	Canada	British Col	Accessori	Bike Rack	Hitch Rack - 4-Bike	6	45	120	443

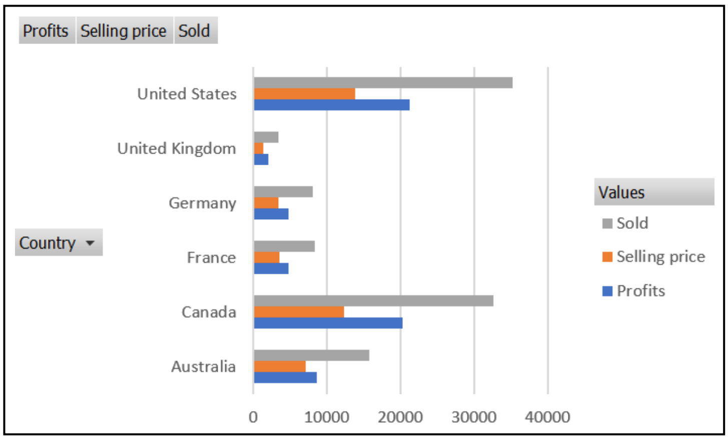
BAR DIAGRAM

Month	Profit
January	369
February	6475
March	7688
April	0
May	8001
June	2858
July	5076
August	6182
September	7324
October	581
November	6631
December	1756



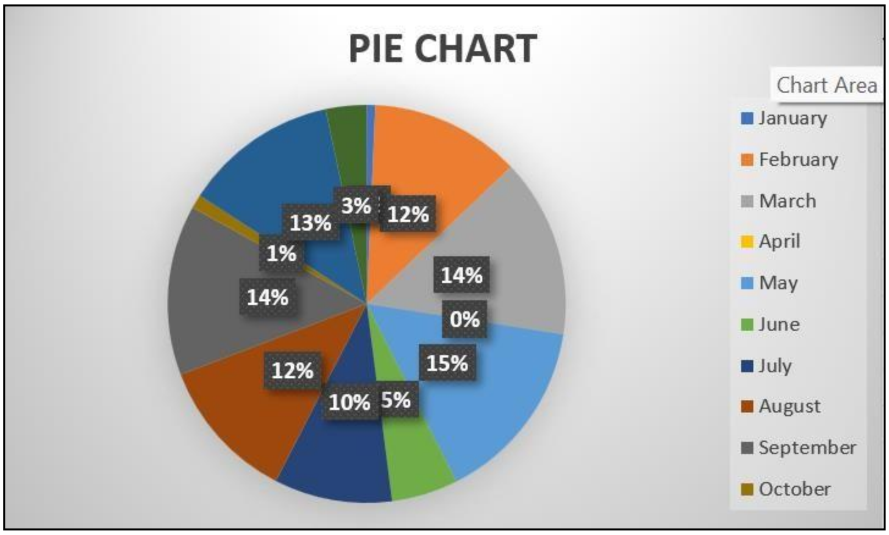
USING PIVOT TABLE

Country	Profits	Selling price	Sold
Australia	8587	7155	15742
Canada	20296	12375	32671
France	4721	3600	8321
Germany	4756	3375	8131
United Kingdom	2106	1350	3456
United States	21296	13860	35156
Grand Total	**61762**	**41715**	**103477**



PIE DIAGRAM

Month	Profit
January	369
February	6475
March	7688
April	0
May	8001
June	2858
July	5076
August	6182
September	7324
October	581
November	6631
December	1756



MEASURE OF CENTRAL TENDENCY

Profit	Cost	Revenue
590	360	950
590	360	950
1366	1035	2401
1188	900	2088
238	180	418
297	225	522
199	180	379
100	90	190
1096	990	2086
1046	945	1991
398	360	758
398	360	758
349	315	664
349	315	664
369	225	594
517	315	832
148	90	238
74	45	119
74	45	119
74	45	119
443	270	713
590	360	950

Central Tendency

Column1	Profit	Cost	Revenue
Mean	623.8586	421.3636	1045.2222
Median	436	315	787
Mode	74	360	119

Mean: Mean gives you an overall average, which might be skewed by extreme values.
Formula used "=AVERAGE()"

Median: Median is your middle-ground, which gives a better sense of centrality when dealing with skewed data.
Formula used "=MEDIAN()"

Mode: Mode tells you the most common value, which is useful in understanding the frequency of specific values.
Formula used "=MODE()"

MEASURE OF DISPERSION

Profit	Cost	Revenue
590	360	950
590	360	950
1366	1035	2401
1188	900	2088
238	180	418
297	225	522
199	180	379
100	90	190
1096	990	2086
1046	945	1991
398	360	758
398	360	758
349	315	664
349	315	664
369	225	594
517	315	832
148	90	238
74	45	119
74	45	119
74	45	119
443	270	713
590	360	950

Measure of Dispersion			
Column1	Profit	Cost	Revenue
Range	2129	1305	3434
Variance	277497.2	118245.9	751925.83
Standard Deviation	526.78	343.869	867.13657
I.Q.R	545	337.5	882.5
C.O.V	84.43901	81.6086	82.961934

Range: Range provides a quick sense of the overall spread.
Formula used "=MAX() - MIN()"

Variance and Standard Deviation: Variance and Standard Deviation give detailed insights into how data points are dispersed around the mean.
Formula used "=VAR.S()"
"=STDEV.S()"

Interquartile Range (IQR): Interquartile Range (IQR) is particularly useful when dealing with skewed data or outliers.
Formula used "=QUARTILE.INC(input_range, 3) - QUARTILE.INC(input_range, 1)"

Coefficient of Variation (CV): Coefficient of Variation (CV) allows for comparison across datasets of different units or scales.
Formula used "=(STDEV.S() / AVERAGE()) * 100"

Correlation and Covariance

Ribbon (Top bar) >> Data >> Data Analysis (Present in right top corner) >> Descriptive Statics

Select Input Range (as given below) > Select Output range >> Check the Label in first row box >> Correlation/Covariance >> Click on OK

Regression Lines
Select Input range () Ribbon → Insert → Recommended charts → All charts → Scatter

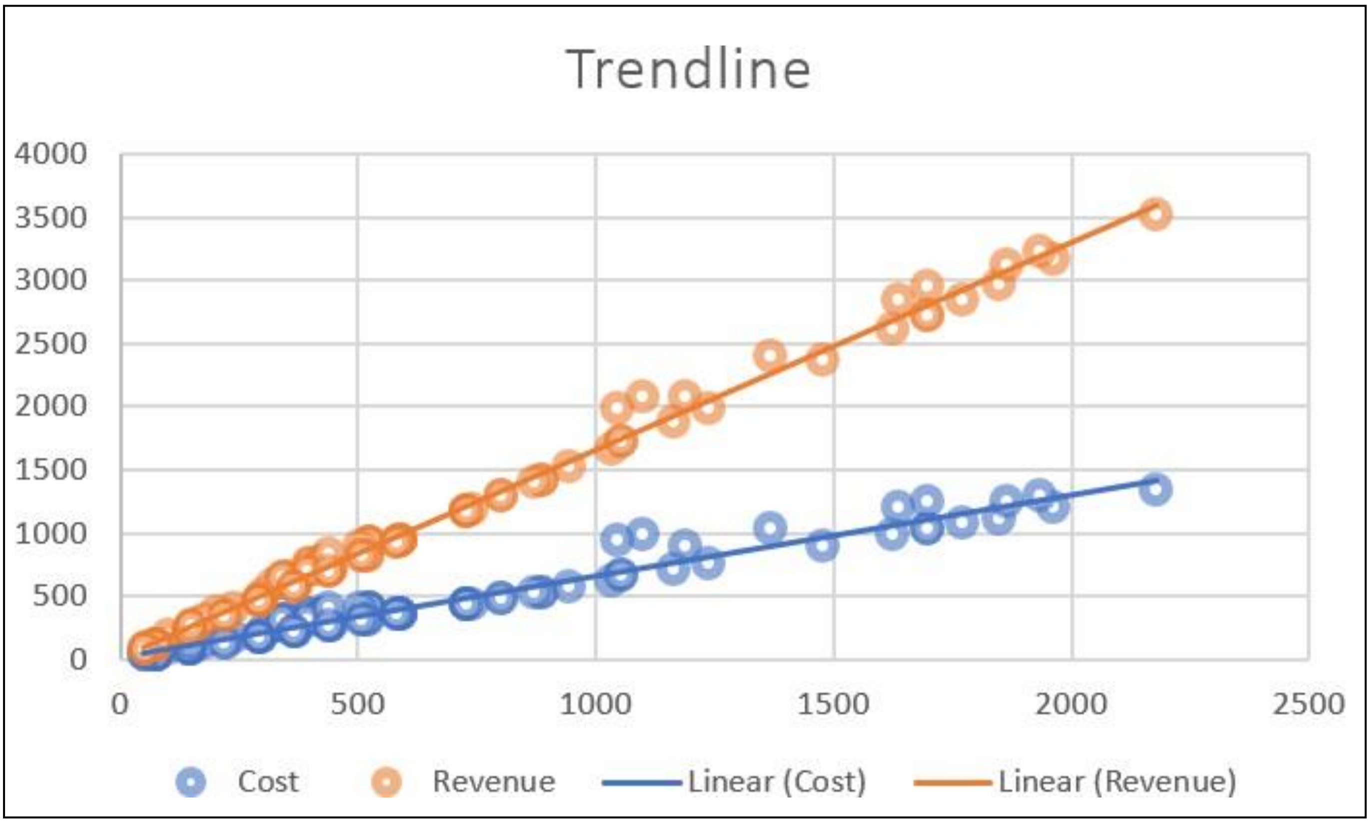
Profit	Cost	Revenue
590	360	950
590	360	950
1366	1035	2401
1188	900	2088
238	180	418
297	225	522
199	180	379
100	90	190
1096	990	2086
1046	945	1991
398	360	758
398	360	758
349	315	664
349	315	664
369	225	594
517	315	832
148	90	238
74	45	119
74	45	119
74	45	119
443	270	713
590	360	950

CORRELATION

	Profit	*Cost*	*Revenue*
Profit	1		
Cost	0.983152	1	
Revenue	0.997369	0.9938156	1

COVARIANCE

	Profit	*Cost*	*Revenue*
Profit	277497.2		
Cost	178091.4	118245.87	
Revenue	455588.6	296337.27	751925.8



PRACTICAL NO.: 13
Date: 13/08/2024

AIM: Aim: t-test, F-test, ANOVA One way Classification, Chi Square test (Independence of Attributes).

Profit	Cost
590	360
590	360
1366	1035
1188	900
238	180
297	225
199	180
100	90
1096	990
1046	945
398	360
398	360
349	315
349	315
369	225
517	315
148	90
74	45
74	45
74	45
443	270
590	360

t-Test: Two-Sample Assuming Unequal Variances

	Profit	*Cost*
Mean	623.8585859	421.3636364
Variance	280328.7553	119452.4583
Observations	99	99
Hypothesized Mean Difference	0	
df	169	
t Stat	3.186549007	
P(T<=t) one-tail	0.000857624	
t Critical one-tail	1.653919942	
P(T<=t) two-tail	0.001715248	
t Critical two-tail	1.974100447	

Conclusion :

Here,

t Stat > t Critical one tail

i.e. 3.1865 > 1.6539

P(T<=t) one tail < Alpha

i.e. 0.00008 < 0.05

Hence,

We Reject the Null Hypothesis

t-test using Data Analysis tool in Excel
Ribbon >> Data >> Data Analysis >> t-Test: Two-Sample Assuming Unequal Variances

Select Variable 1 range >> Select Variable 2 range >> Select Output range >> check Labels in 1st row >> Ok

A clinic provides a program to help their clients lose weight and asks a consumer agency to investigate the effectiveness of the program. The agency takes a sample of 15 people, weighing each person in the sample before the program begins and 3 months later produces the table.
Determine whether the program is effective.

Hypothesis:
H0: The program is not effective.
H1: The program is effective.

Person	Before	After
1	210	197
2	205	195
3	193	191
4	182	174
5	259	236
6	239	226
7	164	157
8	197	196
9	222	201
10	211	196
11	187	181
12	175	164
13	186	181
14	243	229
15	246	231

Paired t-test using Data Analysis tool in Excel
Ribbon >> Data >> Data Analysis >>
t-Test: Paired Two Sample for Means

Select Variable 1 range >> Select Variable 2 range >> Select Output range >> check Labels in 1st row >> Ok

Conclusion:
Here,
t Stat > t Critical one-tail
i.e 6.6897 > 2.1448
P(T<=t) one-tail < Alpha
i.e 5.14E-06 < 0.05
Hence,
We Reject the Null Hypothesis

t-Test: Paired Two Sample for Means		
	Before	*After*
Mean	207.9333333	197
Variance	815.7809524	595
Observations	15	15
Pearson Correlation	0.983720406	
Hypothesized Mean Difference	0	
df	14	
t Stat	6.689699535	
P(T<=t) one-tail	5.13783E-06	
t Critical one-tail	1.761310136	
P(T<=t) two-tail	1.02757E-05	
t Critical two-tail	2.144786688	

Therefore,
The program is effective.

F-test using Data Analysis tool in Excel
Ribbon >> Data >> Data Analysis >> **F-Test: Two-Sample for Variances**

Select Variable 1 range >> Select Variable 2 range >> Select Output range >> check Labels in 1st row >> Ok

Profit	Cost
590	360
590	360
1366	1035
1188	900
238	180
297	225
199	180
100	90
1096	990
1046	945
398	360
398	360
349	315
349	315
369	225
517	315
148	90
74	45
74	45
74	45
443	270
590	360

F-Test Two-Sample for Variances		
	Profit	*Cost*
Mean	623.8585859	421.3636364
Variance	280328.7553	119452.4583
Observations	99	99
df	98	98
F	2.346780965	
P(F<=f) one-tail	1.65951E-05	
F Critical one-tail	1.396443091	

Conclusion:
Here,
F > F Critical one-tail
i.e 2.3655 > 1.672
P(F<=f) one-tail < Alpha
i.e 0.0015 < 0.05
Hence,
We Reject the Null Hypothesis

F-Test: Used to compare variances. We compare the spread or variability of two groups.

ANOVA (One way Classification)

A trial was run to check the effects of different diets. Positive numbers indicate weight loss and negative numbers indicate weight gain. Check if there is an average difference in the weight of people following different diets using an ANOVA Table.

Hypothesis:
H0: There is no average difference in the weight of people following different diets
H1: There is an average difference in the weight of people following different diets.

Low Fat	Low Calorie	Low Protein	Low Carb
8	2	3	2
9	4	5	2
6	3	4	-1
7	5	2	0
3	1	3	3

Anova: Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
Low Fat	5	33	6.6	5.3
Low Calorie	5	15	3	2.5
Low Protein	5	17	3.4	1.3
Low Carb	5	6	1.2	2.7

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	75.75	3	25.25	8.559322	0.001278	3.238872
Within Groups	47.2	16	2.95			
Total	122.95	19				

Conclusion:
Here,
F > F Critical
i.e 8.559322 > 3.238872

P-value < Alpha
i.e 0.001278 < 0.05

Hence,
We Reject the Null Hypothesis

Anova: Single Factor using Data Analysis tool in Excel
Ribbon >> Data >> Data Analysis >> **Anova: Single Factor**

Select Input range >> Select Output range >> check Labels in 1st row >> Ok

Therefore,
There is an average difference in the weight of people following different diets.

Chi Square test (Independence of Attributes)

Suppose you wish to classify defects in the furniture produced by a manufacturing plant based on the type of defects and the production shift. A total of 309 furniture defects were recorded, and the defects were classified as one of four types A, B, C, and D.
At the same time, each piece of defective furniture was identified according to the production shift.

Shift	Type of defect			
	A	B	C	D
1	15	21	45	13
2	26	31	34	5
3	33	17	49	20

Hypothesis:
H0: The Furniture is not defective.
H1: The Furniture is defective.

We calculate Expected value using
Eij = (Row Total × Column Total) / Grand Total

OBSERVED					
	TYPE OF DEFECT				
SHIFT	A	B	C	D	R TOTAL
1	15	21	45	13	94
2	26	31	34	5	96
3	33	17	49	20	119
C TOTAL	74	69	128	38	309

EXPECTED				
	TYPE OF DEFECT			
SHIFT	A	B	C	D
1	22.51133	20.99029	38.93851	11.55987
2	22.99029	21.43689	39.76699	11.80583
3	28.49838	26.57282	49.2945	14.6343

Conclusion:
P-Value < Alpha
i.e 0.003873 < 0.05
Hence,
We Reject the Null Hypothesis

To calculate the **p-value** we use :
=CHISQ.TEST(actual_range, expected_range)
Here,
actual_range: The range containing the observed data.
expected_range: The range containing the expected frequencies.

PRACTICAL NO.: 14

Date: 14/08/2024

AIM: Time series: Forecasting Method of Least Squares, Moving Average Method. Inference and Discussion of Results.

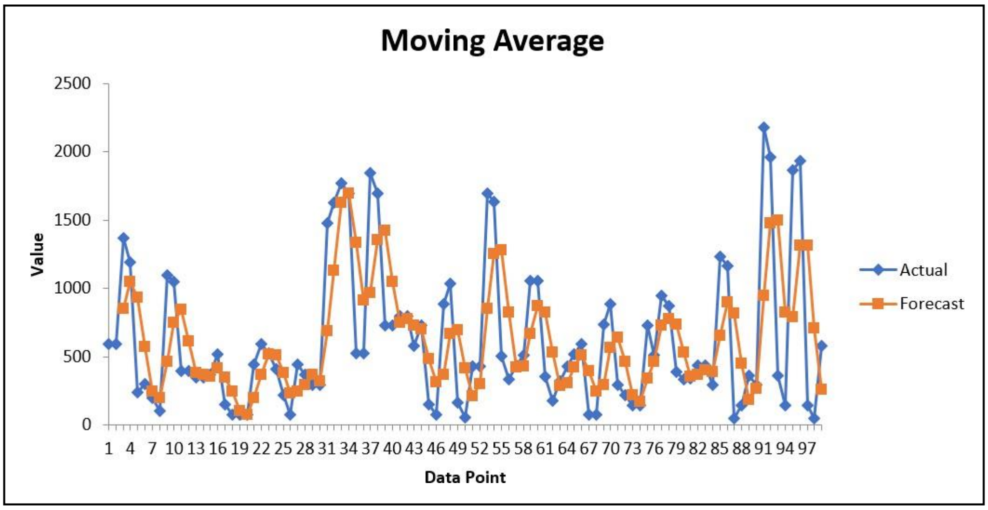
Moving Average Method using Data Analysis tool in Excel:

Ribbon >> Data >> Data Analysis >> **Moving Average**

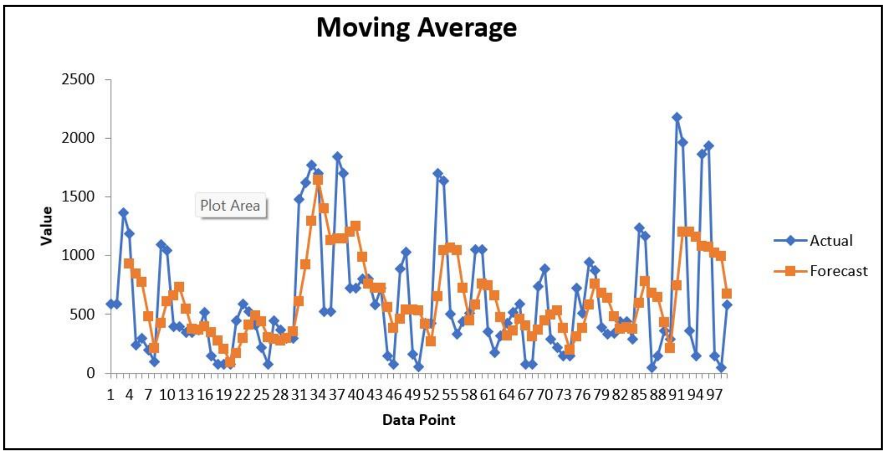
Select Input range >> Interval (= 3, 4, 5) >> Select Output range >> check Labels in 1st row

Year	Profit
2013	590
2015	590
2014	1366
2016	1188
2014	238
2016	297
2014	199
2016	100
2014	1096
2016	1046
2013	398
2015	398
2013	349
2015	349
2013	369
2015	517
2013	148
2015	74
2014	74
2016	74
2014	443
2016	590

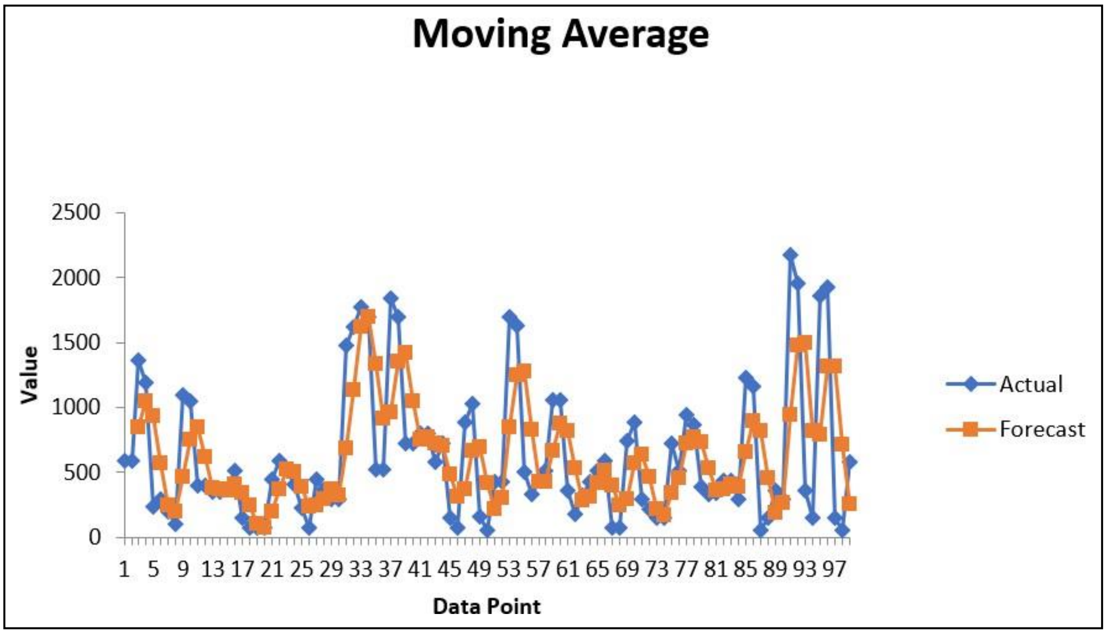
3 yrs MA
#N/A
#N/A
848.66667
1048
930.66667
574.33333
244.66667
198.66667
465
747.33333
846.66667
614
381.66667
365.33333
355.66667
411.66667
344.66667
246.33333
98.666667
74
197
369



4 yrs MA
#N/A
#N/A
#N/A
933.5
845.5
772.25
480.5
208.5
423
610.25
660
734.5
547.75
373.5
366.25
396
345.75
277
203.25
92.5
166.25
295.25



#N/A
848.6667
1048
930.6667
574.3333
244.6667
198.6667
465
747.3333
846.6667
614
381.6667
365.3333
355.6667
411.6667
344.6667
246.3333
98.66667
74
197
369
519



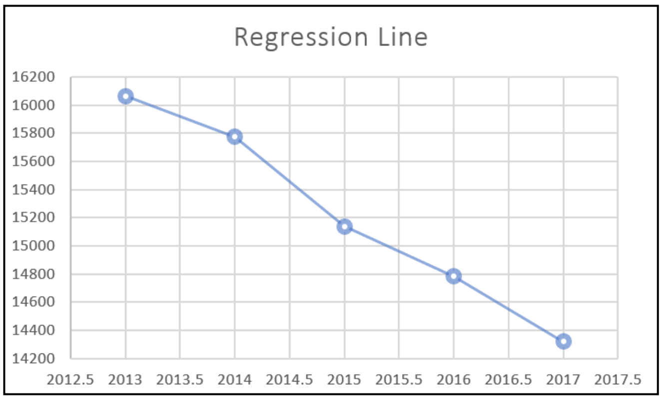
Method of Least Squares

2013	16065
2014	15775
2015	15139
2016	14783

SLOPE	-448.2
INTERCEPT	918339.4

=D15*2017+D16

Years	Profit
2013	16065
2014	15775
2015	15139
2016	14783
2017	14320



PRACTICAL NO.: 15

Date: 03/09/2024

AIM: Advanced Data Analysis using PivotTables and Pivot Charts

Dataset

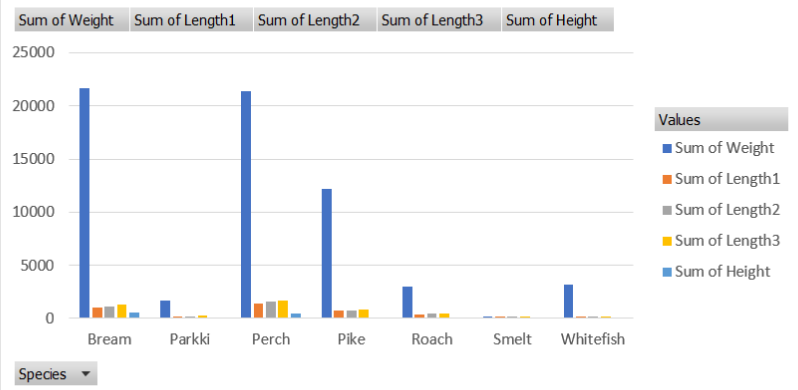
Species	Weight	Length1	Length2	Length3	Height	Width
Bream	242	23.2	25.4	30	11.52	4.02
Bream	290	24	26.3	31.2	12.48	4.3056
Bream	340	23.9	26.5	31.1	12.3778	4.6961
Bream	363	26.3	29	33.5	12.73	4.4555
Bream	430	26.5	29	34	12.444	5.134
Bream	450	26.8	29.7	34.7	13.6024	4.9274
Bream	500	26.8	29.7	34.5	14.1795	5.2785
Bream	390	27.6	30	35	12.67	4.69
Bream	450	27.6	30	35.1	14.0049	4.8438
Bream	500	28.5	30.7	36.2	14.2266	4.9594
Bream	475	28.4	31	36.2	14.2628	5.1042
Bream	500	28.7	31	36.2	14.3714	4.8146
Bream	500	29.1	31.5	36.4	13.7592	4.368
Bream	340	29.5	32	37.3	13.9129	5.0728
Bream	600	29.4	32	37.2	14.9544	5.1708
Bream	600	29.4	32	37.2	15.438	5.58
Bream	700	30.4	33	38.3	14.8604	5.2854
Bream	700	30.4	33	38.5	14.938	5.1975
Bream	610	30.9	33.5	38.6	15.633	5.1338
Bream	650	31	33.5	38.7	14.4738	5.7276
Bream	575	31.3	34	39.5	15.1285	5.5695
Bream	685	31.4	34	39.2	15.9936	5.3704
Bream	620	31.5	34.5	39.7	15.5227	5.2801

Using Pivot to find the total sum of the individual columns grouped by species

Row Labels	Sum of Weight	Sum of Length1	Sum of Length2	Sum of Length3	Sum of Height
Bream	21624	1060.7	1158.8	1342.4	531.4124
Parkki	1703	206	223.8	250.7	98.5867
Perch	21405.4	1441.2	1562	1656	440.2647
Pike	12218	722.1	773.2	828.2	131.1341
Roach	3041	412.9	445.5	499.4	133.8959
Smelt	156.5	157.6	166.9	182.5	30.9312
Whitefish	3186	172.8	187.9	205.9	60.163
Grand Total	**63333.9**	**4173.3**	**4518.1**	**4965.1**	**1426.388**

Using Pivot to plot graphs

Row Labels	Sum of Weight	Sum of Length1	Sum of Length2	Sum of Length3	Sum of Height
Bream	21624	1060.7	1158.8	1342.4	531.4124
Parkki	1703	206	223.8	250.7	98.5867
Perch	21405.4	1441.2	1562	1656	440.2647
Pike	12218	722.1	773.2	828.2	131.1341
Roach	3041	412.9	445.5	499.4	133.8959
Smelt	156.5	157.6	166.9	182.5	30.9312
Whitefish	3186	172.8	187.9	205.9	60.163
Grand Total	**63333.9**	**4173.3**	**4518.1**	**4965.1**	**1426.388**



Implementing Slicer with the help of Pivot

Row Labels	Sum of Weight	Sum of Length1	Sum of Length2	Sum of Length3	Sum of Height
Bream	21624	1060.7	1158.8	1342.4	531.4124
Whitefish	3186	172.8	187.9	205.9	60.163
Grand Total	**24810**	**1233.5**	**1346.7**	**1548.3**	**591.5754**