VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Accredited by NBA

S. S. Education Trust's

S.G Balekundri Institute of Technology

Shivabasava Nagar Belagavi, Karnataka, India – 590010

LAB MANUAL

COURSE NAME: R Programming

COURSE CODE: BCS358B

SEMESTER III

2023-24

R Programming		Semester	3
Course Code	BCS358B	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	0:0:2:0	SEE Marks	50
Credits	01	Exam Hours	02
Examination type (SEE)	Practical		

Course objectives:

- To explore and understand how R and R Studio interactive environment.
- To understand the different data Structures, data types in R.
- To learn and practice programming techniques using R programming.
- To import data into R from various data sources and generate visualizations.
- To draw insights from datasets using data analytics techniques.

Sl.NO	Experiments
1	Demonstrate the steps for installation of R and R Studio. Perform the following: a) Assign different type of values to variables and display the type of variable. Assign different types such as Double, Integer, Logical, Complex and Character and understand the difference between each data type. b) Demonstrate Arithmetic and Logical Operations with simple examples. c) Demonstrate generation of sequences and creation of vectors. d) Demonstrate Creation of Matrices e) Demonstrate the Creation of Matrices from Vectors using Binding Function. f) Demonstrate element extraction from vectors, matrices and arrays **Suggested Reading** – Text Book 1 – Chapter 1 (What is R, Installing R, Choosing an IDE – RStudio, How to Get Help in R, Installing Extra Related Software), Chapter 2 (Mathematical Operations and Vectors, Assigning Variables, Special Numbers, Logical Vectors), Chapter 3 (Classes, Different Types of Numbers, Other Common Classes, Checking and Changing Classes, Examining Variables)
2	Assess the Financial Statement of an Organization being supplied with 2 vectors of data: Monthly Revenue and Monthly Expenses for the Financial Year. You can create your own sample data vector for this experiment) Calculate the following financial metrics: a. Profit for each month. b. Profit after tax for each month (Tax Rate is 30%). c. Profit margin for each month equals to profit after tax divided by revenue. d. Good Months – where the profit after tax was greater than the mean for the year. e. Bad Months – where the profit after tax was less than the mean for the year. f. The best month – where the profit after tax was max for the year. g. The worst month – where the profit after tax was min for the year. **Note:** a. All Results need to be presented as vectors b. Results for Dollar values need to be calculated with $0.01 precision, but need to be presented in Units of $1000 (i.e 1k) with no decimal points c. Results for the profit margin ratio need to be presented in units of % with no decimal point. d. It is okay for tax to be negative for any given month (deferred tax asset) e. Generate CSV file for the data. **Suggested Reading** – Text Book 1 – Chapter 4 (Vectors, Combining Matrices)
3	Develop a program to create two 3 X 3 matrices A and B and perform the following operations a) Transpose of the matrix b) addition c) subtraction d) multiplication **Suggested Reading** – Text Book 1 – Chapter 4 (Matrices and Arrays – Array Arithmetic)
4	Develop a program to find the factorial of given number using recursive function calls. **Suggested Reading** – Reference Book 1 – Chapter 5 (5.5 – Recursive Programming) Text Book 1 – Chapter 8 (Flow Control and Loops – If and Else, Vectorized If, while loops, for loops), Chapter 6 (Creating and Calling Functions, Passing Functions to and from other functions)

5 Develop an R Program using functions to find all the prime numbers up to a specified number by the method of Sieve of Eratosthenes.

Suggested Reading – Reference Book

1 - Chapter 5 (5.5 – Recursive Programming)

Text Book 1 – Chapter 8 (Flow Control and Loops – If and Else, Vectorized If, while loops, for loops), Chapter 6 (Creating and Calling Functions, Passing Functions to and from other functions)

6 The built-in data set mammals contain data on body weight versus brain weight. Develop R commands to:

a) Find the Pearson and Spearman correlation coefficients. Are they similar?

b) Plot the data using the plot command.

c) Plot the logarithm (log) of each variable and see if that makes a difference.

Suggested Reading – Text Book 1 –Chapter 12 – (Built-in Datasets) Chapter 14 – (Scatterplots)

Reference Book 2 – 13.2.5 (Covariance and Correlation)

7 Develop 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) Create another Data Frame called "item-details" with three different fields itemCode, ItemQtyonHand and ItemReorderLvl and merge the two frames

Suggested Reading –Textbook 1: Chapter 5 (Lists and Data Frames)

8 Let us use the built-in dataset air quality which has Daily air quality measurements in New York, May to September 1973. Develop R program to generate histogram by using appropriate arguments for the following statements.

a) Assigning names, using the air quality data set.

b) Change colors of the Histogram

c) Remove Axis and Add labels to Histogram

d) Change Axis limits of a Histogram

e) Add Density curve to the histogram

Suggested Reading –Reference Book 2 – Chapter 7 (7.4 – The ggplot2 Package), Chapter 24 (Smoothing and Shading)

9 Design a data frame in R for storing about 20 employee details. Create a CSV file named "input.csv" that defines all the required information about the employee such as id, name, salary, start_date, dept. Import into R and do the following analysis.

a) Find the total number rows & columns

b) Find the maximum salary

c) Retrieve the details of the employee with maximum salary

d) Retrieve all the employees working in the IT Department.

e) Retrieve the employees in the IT Department whose salary is greater than 20000 and write these

10	Using the built in dataset mtcars which is a popular dataset consisting of the design and fuel consumption patterns of 32 different automobiles. The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973-74 models). Format A data frame with 32 observations on 11 variables : [1] mpg Miles/(US) gallon, [2] cyl Number of cylinders [3] disp Displacement (cu.in.), [4] hp Gross horsepower [5] drat Rear axle ratio,[6] wt Weight (lb/1000) [7] qsec 1/4 mile time, [8] vs V/S, [9] am Transmission (0 = automatic, 1 = manual), [10] gear Number of forward gears, [11] carb Number of carburetors Develop R program, to solve the following: a) What is the total number of observations and variables in the dataset? b) Find the car with the largest hp and the least hp using suitable functions c) Plot histogram / density for each variable and determine whether continuous variables are normally distributed or not. If not, what is their skewness? d) What is the average difference of gross horse power(hp) between automobiles with 3 and 4 number of cylinders(cyl)? Also determine the difference in their standard deviations. e) Which pair of variables has the highest Pearson correlation? **References (Web links):** 1. https://cran.r-project.org/web/packages/explore/vignettes/explore_mtcars.html 2. https://www.w3schools.com/r/r_stat_data_set.asp 3. https://rpubs.com/BillB/217355
11	Demonstrate the progression of salary with years of experience using a suitable data set (You can create your own dataset). Plot the graph visualizing the best fit line on the plot of the given data points. Plot a curve of Actual Values vs. Predicted values to show their correlation and performance of the model. Interpret the meaning of the slope and y-intercept of the line with respect to the given data. Implement using lm function. Save the graphs and coefficients in files. Attach the predicted values of salaries as a new column to the original data set and save the data as a new CSV file. **Suggested Reading** – Reference Book 2 – Chapter 20 (General Concepts, Statistical Inference, Prediction)

Course outcomes (Course Skill Set):
At the end of the course the student will be able to:

- Explain the fundamental syntax of R data types, expressions and the usage of the R-Studio IDE
- Develop a program in R with programming constructs: conditionals, looping and functions.
- Apply the list and data frame structure of the R programming language.
- Use visualization packages and file handlers for data analysis..

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

Continuous Internal Evaluation (CIE):

CIE marks for the practical course are **50 Marks.**

The split-up of CIE marks for record/ journal and test are in the ratio **60:40**.

- Each experiment is to be evaluated for conduction with an observation sheet and record write-up. Rubrics for the evaluation of the journal/write-up for hardware/software experiments are designed by the faculty who is handling the laboratory session and are made known to students at the beginning of the practical session.
- Record should contain all the specified experiments in the syllabus and each experiment write-up will be evaluated for 10 marks.
- Total marks scored by the students are scaled down to **30 marks** (60% of maximum marks).
- Weightage to be given for neatness and submission of record/write-up on time.
- Department shall conduct a test of 100 marks after the completion of all the experiments listed in the syllabus.
- In a test, test write-up, conduction of experiment, acceptable result, and procedural knowledge will carry a weightage of 60% and the rest 40% for viva-voce.
- The suitable rubrics can be designed to evaluate each student's performance and learning ability.
- The marks scored shall be scaled down to **20 marks** (40% of the maximum marks).

The Sum of scaled-down marks scored in the report write-up/journal and marks of a test is the total CIE marks scored by the student.

Semester End Evaluation (SEE):

- SEE marks for the practical course are 50 Marks.
- SEE shall be conducted jointly by the two examiners of the same institute, examiners are appointed by the Head of the Institute.
- The examination schedule and names of examiners are informed to the university before the conduction of the examination. These practical examinations are to be conducted between the schedule mentioned in the academic calendar of the University.
- All laboratory experiments are to be included for practical examination.
- (Rubrics) Breakup of marks and the instructions printed on the cover page of the answer script to be strictly adhered to by the examiners. OR based on the course requirement evaluation

PROGRAM OUTCOME’S (PO's)

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems

2. **Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

INSTITUTION VISION STATEMENT and MISSION STATEMENT

VISION STATEMENT

To impart Quality Education with Human values and emerge as one of the Nation's leading Institutions in the field of Technical Education and Research.

MISSION STATEMENTS

- **S**trive to encourage ideas, talents and value systems.
- **G**uide students to be successful in their endeavor with moral and ethical values.
- **B**uild relation with Industries and National Laboratories to support in the field of Engineering and Technology.
- **I**nculcate a thirst for knowledge in students and help them to achieve Academic Excellence and Placement.
- **T**rain and develop the faculty to achieve Professional,Organizational objectives, and excel in Research and Development

VISION STATEMENT and MISSION STATEMENTS

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION STATEMENT

"To strive for excellence in imparting knowledge of Computer Science and Engineering to produce IT professionals committed to human values".

MISSION STATEMENTS

Mission 1: Impart quality education in cutting edge technologies to achieve excellence in computer science and engineering to solve real-world problems.

Mission 2: Imbibe human values and ethical responsibilities in professional endeavors.

PROGRAM SPECIFIC OUTCOMES:

PSO1: To analyze and resolve the engineering problems related to Artificial Intelligence and Big Data analytics for efficient design of a computer-based system of varying complexity.

PSO2: To design, develop, and arrive at the optimal solution for complex computer science engineering problems with synthesized optimal hardware and software.

PSO3: Apply reasoning informed by the contextual knowledge of computer science and engineering to resolve societal, health, safety and environmental problems.

PROGRAM EDUCATIONAL OBJECTIVEs (PEOs)

PEO1: To prepare graduates to succeed in IT-enabled professional careers, higher studies and research by providing a contextually appropriate academic environment.

PEO2: To prepare graduates to be independent and adapt to the changing technologies by inculcating life-long learning ability, leadership qualities and entrepreneurial skills.

PEO3: To prepare graduates to be committed citizens with social, ethical, and professional concerns.

Introduction toRprogramming:

R is a programming language and free software developed by Ross Ihaka and Robert Gentleman in1993. R possesses an extensive catalog of statistical and graphical methods. It includes machinelearning algorithms, linear regression, time series, statistical inference to name a few. Most of the Rlibraries arewritteninR,butforheavycomputational tasks,C,C++andFortran codesarepreferred. R is not only entrusted by academic, but many large companies also use R programminglanguage,includingUber,Google,Airbnb,Facebookandsoon.

DataanalysiswithRisdoneinaseriesofsteps;programming,transforming,discovering,modelingandcommunicatetheresults.

Program:Risaclearandaccessibleprogrammingtool

Transform: Ris madeupofacollectionoflibrariesdesignedspecificallyfordatascience

Discover: Investigatethedata, refineyour hypothesisandanalyzethem

Model:Rprovidesawidearrayoftoolsto capturetherightmodelforyourdata

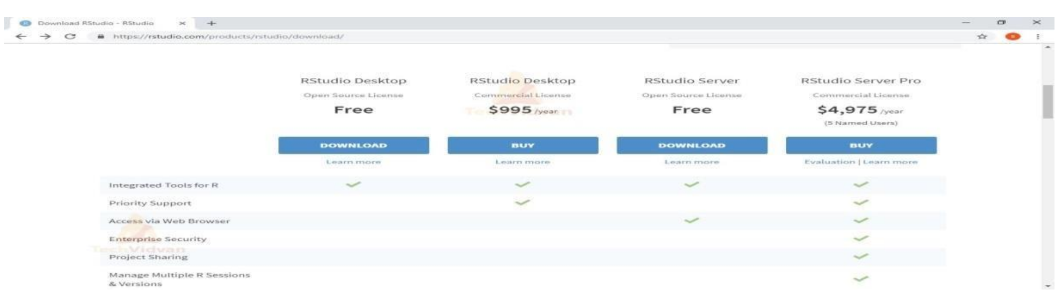
Communicate: Integrate codes, graphs, and outputs to a report with R Markdown or build Shinyappstosharewiththeworld

What isRusedfor?

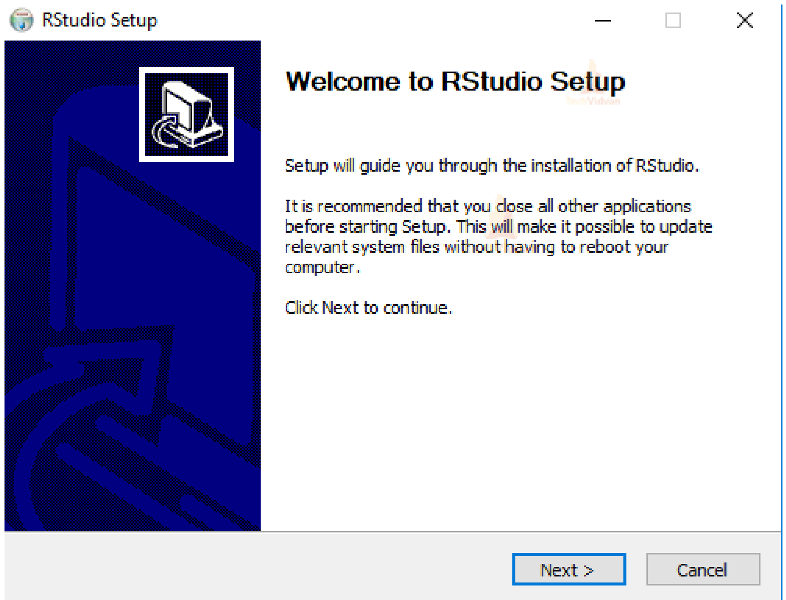
- Statisticalinferen
- ceDataanalysis
- Machinelearningalgorithm

InstallationofR-Studioonwindows:

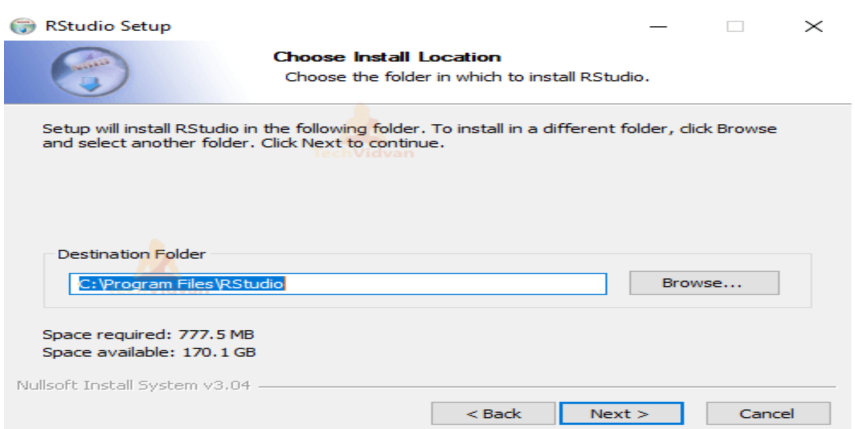
Step – 1: With R-base installed, let’s move on to installing R Studio. To begin, gotodownload RStudioand clickonthe download buttonforRStudiodesktop.



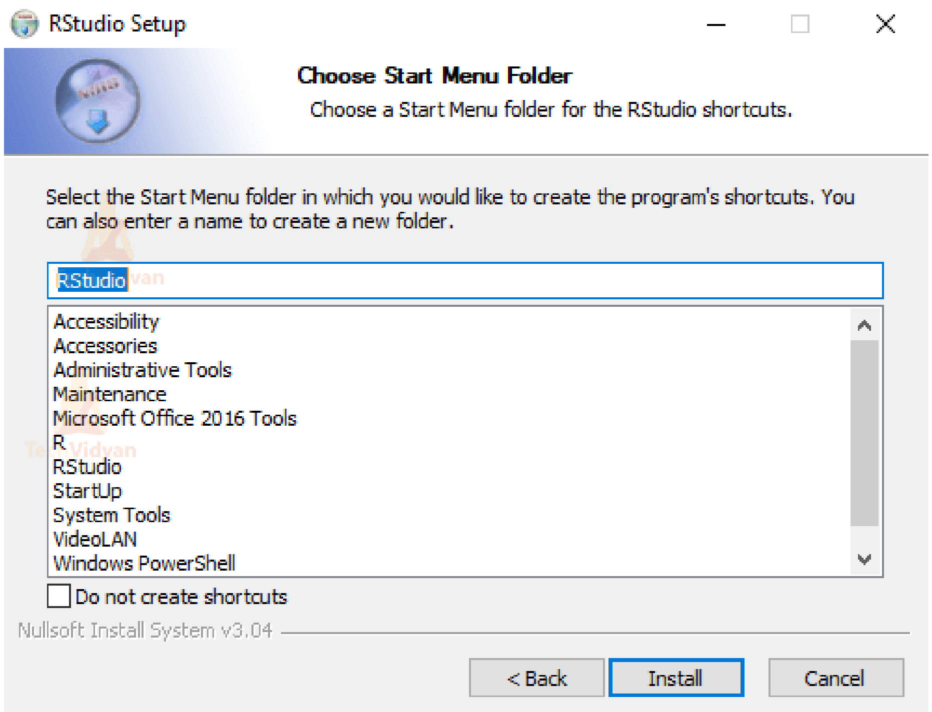
- Step–2:Clickonthelink forthewindowsversionofRStudioand
- savethe.exefile.Step–3:Runthe .exe andfollowthe installationinstructions.
- 3. ClickNext onthewelcome window.



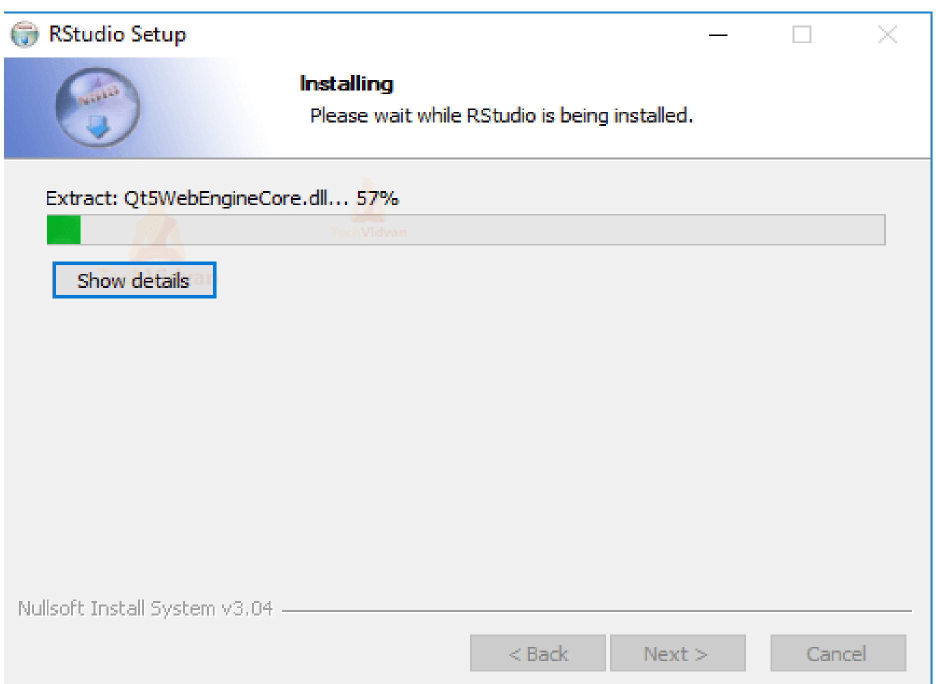
- Enter/browsethepathtotheinstallation folderandclickNextto proceed.



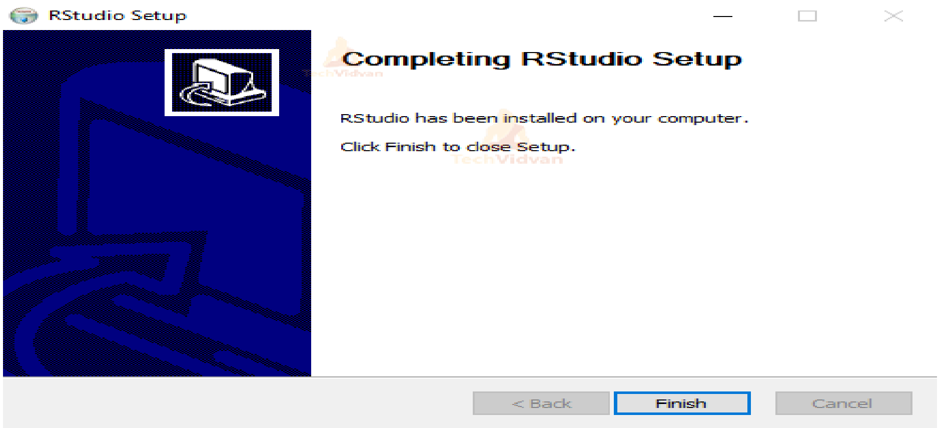
Select the folder for the start menu shortcut or click on do not create shortcuts and thenclickNext.



Waitfortheinstallationprocesstocomplete.



ClickFinishtoendtheinstallation.



InstalltheRPackages: -

InRStudio,ifyourequireaparticularlibrary,thenyoucangothroughthefollowinginstructions:

- First,runRStudio.
- After clicking on the packages tab, click on install. The following dialog box willappear.
- In the Install Packages dialog, write the package name you want to install underthe Packagesfield andthen clickinstall.Thiswillinstall the packageyousearchedfororgiveyoualistofmatchingpackagesbasedonyour packagetext.

InstallingPackages: -
ThemostcommonplacetogetpackagesfromisCRAN.ToinstallpackagesfromCRANyouuseinstall. Packages ("packagename").Forinstance,ifyouwanttoinstalltheggplot2package, which is a very popular visualization package, you would type the following in theconsole: -
Syntax: -
#installpackagefromCRAN
install.
Packages("ggplot2")**Loadin**
gPackages: -

Oncethepackageisdownloadedtoyourcomputer youcanaccessthefunctionsand resourcesprovidedbythepackageintwodifferentways:
#loadthepackagetouseinthecurrentRsession
library (package name)

AssignmentOperators: -

Thefirstoperatoryou'llrunintoistheassignmentoperator.Theassignmentoperatorisusedto assign a value. For instance, we can assign the value 3 to the variable x using the <- assignmentoperator.

assignmentx<-3

Interestingly,Ractuallyallows forfiveassignmentoperators:

#**leftwardassignment**

x<-valuex=value

x<<-value

#

rightward assignment

value -> xvalue->>x

TheoriginalassignmentoperatorinRwas<-andhascontinuedtobethepreferredamongRusers. The = assignment operator was added in 2001 primarily because it is the acceptedassignment operator in many other languages and beginners to R coming from otherlanguages weresopronetouseit.

Theoperators<<- isnormallyonlyusedinfunctionswhichwewillnotgetintothedetails.

Evaluation

Wecanthenevaluatethevariablebysimplytypingxat thecommandline whichwillreturnthevalueofx.Notethat priortothevaluereturnedyou'llsee## [1]inthecommandline. This simply implies that the output returned is the first output. Note that you can type anycommentsinyour codebyprecedingthecommentwiththehashtag(#)symbol.Anyvalues,symbols,andtexts following#willnotbe evaluated.

evaluationx

##[1] 3

CaseSensitivity

Lastly,notethatRisacasesensitiveprogramminglanguage.Meaningallvariables,functions,andobjec tsmustbe calledbytheirexactspelling:

x<-1

y<-3

z <-4

x * y * z##[1]12

x*Y*z

##Errorineval(expr,envir,enclos): object'Y'notfound

BasicArithmetic

AtitsmostbasicfunctionRcanbeusedasacalculator.Whenapplyingbasicarithmetic, thePEMDAS order of operations applies: parentheses first followed by exponentiation,multiplicationanddivision,andfinaladditionandsubtraction.

8+9/5^ 2

##[1]8.36

8+9/(5^2)
##[1]8.36
8+(9/5)^ 2
##[1]11.24
(8+9)/5^ 2
##[1]0.68
BydefaultRwilldisplaysevendigitsbut thiscanbechangedusing options ()aspreviouslyoutlined.
1/7
##[1]0.1428571
options (digits=3)
1/7
##[1]0.143
pi
##[1]3.141592654
options (digits = 22)pi
##[1]3.141592653589793115998
We can also perform integer divide (%/%) and modulo (%%) functions. The integer dividefunctionwillgivetheintegerpartofafractionwhilethemodulowillprovidetheremainder.42/4 #regulardivision
##[1]10.5
42%/%4 #integerdivision
##[1] 10
42%%4 #modulo(remainder)
##[1] 2

MiscellaneousMathematicalFunctions

Therearemanybuilt-infunctionstobeawareof.Theseincludebutarenotlimitedtothefollowing.Goaheadandrunthiscodeinyourconsole.
x<-10
abs(x) #absolutevaluesqrt(x) #squareroot
exp(x) #exponentialtransformationlog(x) #logarithmictransformation
cos(x) #cosineandothertrigonometricfunctions

Head: -
Tobegin,wearegoing torunthe headfunction,whichallowsustoseethefirst6rowsbydefault.Wearegoingtooverridethe defaultandasktopreviewthefirst10rows.
>head (df,10)

Tail: -Tailfunctionallowsustoseethelastnobservations fromagivendataframe. Thedefaultvalueforniss6.Usercanspecifyvalueofnasperas requirements.

>tail(mtcars,n=5)

DimandGlimpse

Next,wewillrunthedimfunctionwhichdisplaysthedimensionsofthetable.Theoutputtakes the form of row, column. And then we run the glimpse function from the plypackage. This will display a vertical preview of the dataset. It allows us to easily previewdatatypeandsampledata.

1.Demonstrate the steps for installation of R and R Studio. Perform the following:
a) Assign different type of values to variables and display the type of variable. Assign different types
such as Double, Integer, Logical, Complex and Character and understand the difference between each data type.
b) Demonstrate Arithmetic and Logical Operations with simple examples.
c) Demonstrate generation of sequences and creation of vectors.
d) Demonstrate Creation of Matrices
e) Demonstrate the Creation of Matrices from Vectors using Binding Function.
f) Demonstrate element extraction from vectors, matrices and arrays

Data Types in R:

```
# Double
double_var<- 3.14
print(class(double_var))
# Integer
int_var<- 42L
print(class(int_var))
# Logical
logical_var<- TRUE
print(class(logical_var))
# Complex
complex_var<- 2 + 3i
print(class(complex_var))
# Character
char_var<- "Hello, World!"
print(class(char_var))
```

Arithmetic Operators in R:

Operator	Description
+	Addition
-	Subtraction
*	Multiplication

/	Division
^ or **	Exponentiation
%%	Modulo Division (remainder of division)
%/%	Integer Division

Logical Operators in R:

<table>
<tr><th>Operator</th><th>Description</th></tr>
<tr><td>&</td><td>Element-wise Logical AND operator. It returns TRUE if both elements are TRUE</td></tr>
<tr><td>&&</td><td>Logical AND operator - Returns TRUE if both statements are TRUE</td></tr>
<tr><td>|</td><td>Elementwise- Logical OR operator. It returns TRUE if one of the statement is TRUE</td></tr>
<tr><td>||</td><td>Logical OR operator. It returns TRUE if one of the statement is TRUE.</td></tr>
<tr><td>!</td><td>Logical NOT - returns FALSE if statement is TRUE</td></tr>
</table>

Vectors in R:

- In R programming, a vector is the most common data structure. It is an array of data elements, each the same type (integer, double, character, logical, or complex).
- Vectors can be atomic, also called scalar variable.
- The function c(), which stands for concatenate, is useful for creating vectors.
- Another useful function for creating vectors is the seq() function, which generates sequences.
  - seq(start, end)
  - start:stop
- We can name the elements of a numeric vector using the names() function.

Matrices in R:

- A matrix is a two dimensional data set with columns and rows.
- A column is a vertical representation of data, while a row is a horizontal representation of data.
  - matrix(data, nrow, ncol, byrow)
  - cbind(v1, v2, v3, …)
    - Combines vectors by columns
  - rbind(v1, v2, v3, …)
    - Combines vectors by rows

Element extraction:

- We use square brackets for subsetting to access specific elements of a vector or matrix.
- If the entries of a vector are named, they may be accessed by referring to their name.
- We get the number of elements using length() function.

EX 2:Assess the Financial Statement of an Organization being supplied with 2 vectors of data: Monthly Revenue

and Monthly Expenses for the Financial Year. You can create your own sample data vector for this
experiment) Calculate the following financial metrics:
a. Profit for each month.
b. Profit after tax for each month (Tax Rate is 30%).
c. Profit margin for each month equals to profit after tax divided by revenue.
d. Good Months – where the profit after tax was greater than the mean for the year.
e. Bad Months – where the profit after tax was less than the mean for the year.
f. The best month – where the profit after tax was max for the year.
g. The worst month – where the profit after tax was min for the year.
Note:
a. All Results need to be presented as vectors
b. Results for Dollar values need to be calculated with $0.01 precision, but need to be presented in
Units of $1000 (i.e 1k) with no decimal points
c. Results for the profit margin ratio need to be presented in units of % with no decimal point.
d. It is okay for tax to be negative for any given month (deferred tax asset)
e. Generate CSV file for the data.
Suggested Reading –

Source Code:

```
data <- read.csv("data.csv")
revenue <- c(data$revenue)
expenses <- c(data$expenses)
profits <- revenue - expenses
print("Profits for each month")
for(i in 1:12) {
  cat("Profit for month", i, "is", profits[i], "\n")
}
profit_after_tax<- profits - round(profits * 0.3, 2)
for(i in 1:12) {
  cat("Profit after tax for month", i, "is", profit_after_tax[i], "\n")
}
profit_margin<- round(profit_after_tax/revenue, 2) * 100
for(i in 1:12) {
  cat("Profit margin for month", i, "is", paste(profit_margin[i], "%"), "\n")
}
mean_for_year<- mean(profit_after_tax)
good_months = c()
bad_months = c()
for(i in 1: length(profit_after_tax)) {
  if(profit_after_tax[i] >mean_for_year)
```

```
good_months = c(good_months, i)
  if(profit_after_tax[i] <mean_for_year)
bad_months = c(bad_months, i)
}
cat("Following months had profit after tax higher than mean:", good_months)
cat("Following months had profit after tax lower than mean:", bad_months)
max_profit = max(profit_after_tax)
min_profit = min(profit_after_tax)
cat("Month where profit after tax is highest is", which(TRUE == (profit_after_tax == max_profit)))
cat("Month where profit after tax is lowest is", which(TRUE == (profit_after_tax == min_profit))
```

EX 3: Develop a program to create two 3 X 3 matrices A and B and perform the following operations

a) Transpose of the matrix b) addition c) subtraction d) multiplication

```
# create matrix with 3 rows and 3 columns
Matrix = matrix(1:9, nrow = 3)

# print the matrix
print(Matrix)

# create another matrix
M2 = Matrix

# Loops for Matrix Transpose
for (i in 1:nrow(M2))
{
        # iterate over each row
        for (j in 1:ncol(M2))
        {
                # iterate over each column
                # assign the correspondent elements
                # from row to column and column to row.
                M2[i, j] <- Matrix[j, i]
        }
}

# print the transposed matrix
print(M2)
```

- **Output:**

```
     [, 1] [, 2] [, 3]
[1,] 1    3    5
[2,] 2    4    6
     [, 1] [, 2]
[1, ]    1    2
[2, ]    3    4
[3, ]    5    6
```

EX 4: Develop a program to find the factorial of given number using recursive function calls.

Full Code

```
# Recursive function to find the factorial of a number
factorial <- function(n) {
  if (n == 0) {
    return(1)
  } else {
    return(n * factorial(n - 1))
  }
}

# Input the number for which you want to find the factorial
number <- as.integer(readline(prompt = "Enter a number: "))

# Check if the input is a non-negative integer
if (number < 0) {
  cat("Factorial is not defined for negative numbers.\n")
} else {
  result <- factorial(number)
  cat(paste("The factorial of", number, "is", result, "\n"))
}
```

Output

```
## Enter a number:
```

1

```
## The factorial of 5 is 120
## Enter a number:
## Factorial is not defined for negative numbers.
```

EX 5: Develop an R Program using functions to find all the prime numbers up to a specified number by the method of Sieve of Eratosthenes

Develop an R Program using functions to find all the prime numbers up to a specified number by the method of Sieve of Eratosthenes.

Solution

Full Code

```
# Function to find all prime numbers up to a specified limit using the Sieve of Eratosthenes
sieve_of_eratosthenes <- function(limit) {
  # Create a logical vector "is_prime" initialized to TRUE for all numbers from 2 to the limit
  is_prime <- rep(TRUE, limit)

  # Set up variables
  p <- 2
  while (p^2 <= limit) {
    # If p is marked as prime, then mark all multiples of p as non-prime
    if (is_prime[p]) {
      k <- p
      while(k <= limit){
        if(k+p <= limit){
          is_prime[k+p] <- FALSE
        }
        k <- k + p
      }
    }
    p <- p + 1
  }
```

```
  # Return a vector of prime numbers up to the limit (excluding 0 and 1)
  primes <- which(is_prime)
  return(primes)
}

# Input the limit (up to which you want to find prime numbers)
limit <- as.integer(readline(prompt = "Enter a limit to find prime numbers: "))

# Check if the input limit is greater than or equal to 2
if (limit < 2) {
  cat("Prime numbers start from 2. Enter a valid limit.\n")
} else {
  prime_numbers <- sieve_of_eratosthenes(limit)
  cat("Prime numbers up to", limit, "are: ", paste(prime_numbers, collapse = ", "), "\n")
}
```

Output

```
## Enter a limit to find prime numbers:

## Prime numbers up to 25 are:  1, 2, 3, 5, 7, 11, 13, 17, 19, 23
```

EX 6: The built-in data set mammals contain data on body weight versus brain weight. Develop R commands to:
a) Find the Pearson and Spearman correlation coefficients. Are they similar?
b) Plot the data using the plot command.
c) Plot the logarithm (log) of each variable and see if that makes a difference

```
# Load the "MASS" package:
library(MASS)

# Load the mammals dataset
data(mammals)

# Calculate Pearson correlation coefficient
pearson_correlation <- cor(mammals$brain, mammals$body, method = "pearson")

# Calculate Spearman correlation coefficient
spearman_correlation <- cor(mammals$brain, mammals$body, method = "spearman")

# Print the correlation coefficients
cat("Pearson Correlation Coefficient:", pearson_correlation, "\n")
cat("Spearman Correlation Coefficient:", spearman_correlation, "\n")

# Plot the data
plot(mammals$body, mammals$brain, xlab = "Body Weight", ylab = "Brain Weight",
main = "Body Weight vs. Brain Weight")

# Calculate the log of body weight and brain weight
log_body <- log(mammals$body)
log_brain <- log(mammals$brain)

# Plot the log-transformed data
plot(log_body, log_brain, xlab = "Log Body Weight", ylab = "Log Brain Weight",
main = "Log Body Weight vs. Log Brain Weight")
```

Output

```
## Pearson Correlation Coefficient: 0.9341638

## Spearman Correlation Coefficient: 0.9534986
```

EX 7:

Develop 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) Create another Data Frame called "item-details" with three different fields itemCode, ItemQtyonHand

EX 8: Develop R program to create a Data Frame with following details and do the following operations.

item Code Item Category Item Price

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) Create another Data Frame called "item-details" with three different fields item Code, Item QtyonHand and ItemReorderLvl and merge the two frames

```
# Create the initial data frame
data <- data.frame(
  itemCode = c(1001, 1002, 1003, 1004, 1005),
  itemCategory = c("Electronics", "Desktop Supplies", "Office Supplies", "USB", "CD Drive"),
  itemPrice = c(700, 300, 350, 400, 800)
)

# a) Subset the Data frame and display details of items with price >= 350
subset_a <- data[data$itemPrice >= 350, ]

# b) Subset the Data frame and display items with category "Office Supplies" or "Desktop Supplies"
subset_b <- data[data$itemCategory %in% c("Office Supplies", "Desktop Supplies"), ]

# c) Create another data frame "item-details"
item_details <- data.frame(
  itemCode = c(1001, 1002, 1003, 1004, 1005),
  ItemQtyonHand = c(10, 20, 15, 30, 25),
  ItemReorderLvl = c(5, 10, 8, 15, 12)
)

# Merge the two data frames using itemCode
merged_data <- merge(data, item_details, by = "itemCode")

# Display the results
cat("Original Data Frame:\n")
print(data)
```



```
cat("\nSubset of items with price >= 350:\n")
print(subset_a)

cat("\nSubset of items with category 'Office Supplies' or 'Desktop Supplies':\n")
```

```
print(subset_b)

cat("\n'item-details' Data Frame:\n")
print(item_details)

cat("\nMerged Data Frame:\n")
print(merged_data)
```

Output

```
## Original Data Frame:

##   itemCode     itemCategory itemPrice
## 1     1001      Electronics       700
## 2     1002 Desktop Supplies       300
## 3     1003  Office Supplies       350
## 4     1004              USB       400
## 5     1005         CD Drive       800

##
## Subset of items with price >= 350:

##   itemCode    itemCategory itemPrice
## 1     1001     Electronics       700
## 3     1003 Office Supplies       350
## 4     1004             USB       400
## 5     1005        CD Drive       800
```

```
##
## Subset of items with category 'Office Supplies' or 'Desktop Supplies':
##   itemCode     itemCategory itemPrice
## 2     1002 Desktop Supplies       300
## 3     1003  Office Supplies       350
##
## 'item-details' Data Frame:
##   itemCode ItemQtyonHand ItemReorderLvl
## 1     1001            10              5
## 2     1002            20             10
## 3     1003            15              8
## 4     1004            30             15
## 5     1005            25             12
##
## Merged Data Frame:
##   itemCode     itemCategory itemPrice ItemQtyonHand ItemReorderLvl
## 1     1001      Electronics       700            10              5
## 2     1002 Desktop Supplies       300            20             10
## 3     1003  Office Supplies       350            15              8
## 4     1004              USB       400            30             15
## 5     1005         CD Drive       800            25             12
```

EX 9: Design a data frame in R for storing about 20 employee details. Create a CSV file named "input.csv" that defines all the required information about the employee such as id, name, salary, start date, dept. Import into R and do the following analysis.

a) Find the total number rows & columns

b) Find the maximum salary

c) Retrieve the details of the employee with maximum salary

d) Retrieve all the employees working in the IT Department.

e) Retrieve the employees in the IT Department whose salary is greater than 20000 and write these details into another file "output.csv"

Step 1: Create a CSV file named "input.csv" with employee details.

Here's an example of how you can create a data frame in R with 20 employee details and then save it to a CSV file:

```
# Create a data frame with employee details
employee_data <- data.frame(
  id = 1:20,
  name = c("John", "Alice", "Bob", "Mary", "David", "Sara", "Michael", "Olivia", "Lucas", "Emma", "Jame
  salary = c(45000, 55000, 60000, 70000, 75000, 62000, 80000, 52000, 58000, 67000, 71000, 59000, 68000,
  start_date = as.Date(c("2023-01-15", "2022-11-05", "2023-02-20", "2021-08-10", "2022-05-15", "2022-03
  dept = c("IT", "HR", "IT", "Finance", "IT", "Sales", "IT", "HR", "Finance", "IT", "Sales", "HR", "IT"
)

# Save the data frame to a CSV file
write.csv(employee_data, "input.csv", row.names = FALSE)
```

Step 2: Import the data from "input.csv" and perform the analysis.

```
# Import data from "input.csv"
employee_data <- read.csv("input.csv")

# a) Find the total number of rows and columns
n_rows <- nrow(employee_data)
n_cols <- ncol(employee_data)
cat("Total number of rows:", n_rows, "\n")
cat("Total number of columns:", n_cols, "\n")

# b) Find the maximum salary
max_salary <- max(employee_data$salary)
cat("Maximum salary:", max_salary, "\n")

# c) Retrieve the details of the employee with the maximum salary
employee_with_max_salary <- employee_data[employee_data$salary == max_salary, ]
cat("Details of employee with maximum salary:\n")
print(employee_with_max_salary)

# d) Retrieve all the employees working in the IT Department
it_department_employees <- employee_data[employee_data$dept == "IT", ]
cat("Employees working in the IT Department:\n")
print(it_department_employees)


# e) Retrieve the employees in the IT Department whose salary is greater than 20000
it_department_high_salary <- it_department_employees[it_department_employees$salary > 20000, ]

# Write these details into another file "output.csv"
write.csv(it_department_high_salary, "output.csv", row.names = FALSE)
```

Output

```
## Total number of rows: 20

## Total number of columns: 5

## Maximum salary: 80000

## Details of employee with maximum salary:

##   id    name salary start_date dept
## 7  7 Michael  80000 2022-12-12   IT

## Employees working in the IT Department:

##    id      name salary start_date dept
## 1   1      John  45000 2023-01-15   IT
## 3   3       Bob  60000 2023-02-20   IT
## 5   5     David  75000 2022-05-15   IT
## 7   7   Michael  80000 2022-12-12   IT
## 10 10      Emma  67000 2023-03-10   IT
## 13 13     Ethan  68000 2022-04-25   IT
## 16 16       Mia  61000 2022-05-11   IT
## 18 18 Charlotte  72000 2023-01-05   IT
```

EX 11: Using the built in dataset mtcars which is a popular dataset consisting of the design and fuel consumption patterns of 32 different automobiles. The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973-74 models). Format A data frame with 32 observations on 11 variables : [1] mpg Miles/(US) gallon, [2] cyl Number of cylinders [3] disp Displacement (cu.in.), [4] hp Gross horsepower [5] drat Rear axle ratio,[6] wt Weight (lb/1000) [7] qsec 1/4 mile time, [8] vs V/S, [9] am Transmission (0 = automatic, 1 = manual), [10] gear Number of forward gears, [11] carb Number of carburetors
Develop R program, to solve the following:
a) What is the total number of observations and variables in the dataset?
b) Find the car with the largest hp and the least hp using suitable functions

c) Plot histogram / density for each variable and determine whether continuous variables are normally distributed or not. If not, what is their skewness?

d) What is the average difference of gross horse power(hp) between automobiles with 3 and 4 number of cylinders(cyl)? Also determine the difference in their standard deviations.

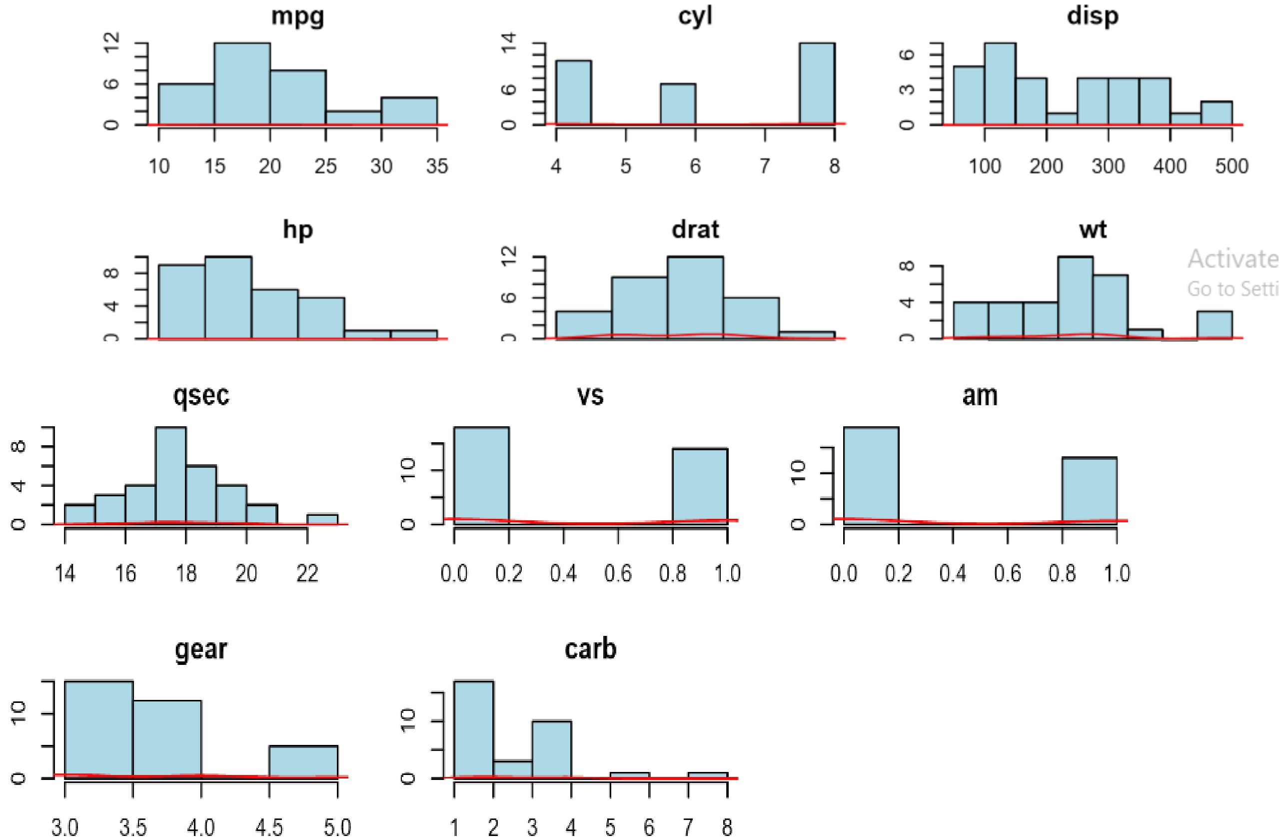
e) Which pair of variables has the highest Pearson correlation?

```
# Load the mtcars dataset
data(mtcars)

# a) Total number of observations and variables
n_observations <- nrow(mtcars)
n_variables <- ncol(mtcars)
cat("Total number of observations:", n_observations, "\n")
cat("Total number of variables:", n_variables, "\n")

# b) Car with the largest and least hp
car_with_largest_hp <- mtcars[which.max(mtcars$hp), ]
car_with_least_hp <- mtcars[which.min(mtcars$hp), ]
cat("Car with the largest hp:\n")
print(car_with_largest_hp)
cat("Car with the least hp:\n")
print(car_with_least_hp)

# Set custom graphical parameters for smaller margins
par(mfrow = c(4, 3))
par(mar = c(3, 3, 2, 1))  # Adjust the margins as needed (bottom margin is set to 1)
```

```
# c) Plot histogram/density for each variable and check for normality
for (col in names(mtcars)) {
  hist(mtcars[, col], main = col, xlab = col, col = "lightblue")
  lines(density(mtcars[, col]), col = "red")
}

# Restore default graphical parameters
par(mfrow = c(1, 1))
par(mar = c(5, 4, 4, 2) + 0.1)
```

Output

```
## Total number of observations: 32

## Total number of variables: 11

## Car with the largest hp:

##                mpg cyl disp  hp drat   wt qsec vs am gear carb
## Maserati Bora  15    8  301 335 3.54 3.57 14.6  0  1    5    8

## Car with the least hp:

##               mpg cyl disp hp drat    wt  qsec vs am gear carb
## Honda Civic 30.4   4 75.7 52 4.93 1.615 18.52  1  1    4    2
```



EX 11: Demonstrate the progression of salary with years of experience using a suitable data set (You can create your own dataset). Plot the graph visualizing

the best fit line on the plot of the given data points. Plot a curve of Actual Values vs. Predicted values to show their correlation and performance of the model.

Interpret the meaning of the slope and y-intercept of the line with respect to the given data. Implement using lm function. Save the graphs and coefficients in files. Attach the predicted values of salaries as a new column to the original data set and save the data as a new CSV file.

```
# Create a sample dataset
set.seed(123)  # Set a random seed for reproducibility
years_of_experience <- 0:10  # Years of experience
salaries <- 30000 + 2000 * years_of_experience + rnorm(11, mean = 0, sd = 5000)  # Generate salaries wi

# Create a data frame
data <- data.frame(Experience = years_of_experience, Salary = salaries)

# Fit a linear regression model
model <- lm(Salary ~ Experience, data = data)

# Plot the data points and best fit line
plot(data$Experience, data$Salary, main = "Salary vs. Years of Experience", xlab = "Years of Experience
abline(model, col = "red")

# Save the plot
png("salary_vs_experience_plot.png")
plot(data$Experience, data$Salary, main = "Salary vs. Years of Experience", xlab = "Years of Experience
abline(model, col = "red")
dev.off()

# Predict values using the model
predicted_values <- predict(model, data)

# Plot the curve of actual values vs. predicted values
plot(data$Salary, predicted_values, main = "Actual Values vs. Predicted Values", xlab = "Actual Salary"
abline(h = 0, v = 0, col = "red")
```

```
# Save the correlation plot
png("actual_vs_predicted_plot.png")
plot(data$Salary, predicted_values, main = "Actual Values vs. Predicted Values", xlab = "Actual Salary"
abline(h = 0, v = 0, col = "red")
dev.off()

# Interpret the slope and y-intercept of the line


coefficients <- coef(model)
slope <- coefficients["Experience"]
intercept <- coefficients["(Intercept)"]
cat("The slope (coefficient for Experience) is", slope, "\n")
cat("The y-intercept (Intercept) is", intercept, "\n")

# Attach the predicted values as a new column to the original data frame
data$Predicted_Salary <- predicted_values

# Save the data with predicted values as a new CSV file
write.csv(data, "salary_vs_experience_data_with_predictions.csv", row.names = FALSE)
```

Output

REFERENCES

1. . Cotton, R. (2013). Learning R: A Step by Step Function Guide to Data Analysis. 1st ed. O'Reilly Media Inc.
2. Jones, O., Maillardet. R. and Robinson, A. (2014). Introduction to Scientific Programming and Simulation Using R. Chapman & Hall/CRC, The R Series.
3. 2Davies, T.M. (2016) The Book of R: A First Course in Programming and Statistics. No Starch Press.