



B.TECH(R-22) (II YEAR - I SEM) (2023-24)



R Programming Lab (R22A6781)

LECTURE LAB MANUAL

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY (Autonomous Institution – UGC, Govt. of India)

Recognized under 2(f) and 12(B) of UGC ACT 1956 (Affiliated to JNTUH, Hyderabad, Approved by AICTE-Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015 Certified) Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad–500100, Telangana State, India

Department of Computer Science and Engineering EMERGING TECHNOLOGIES

R Programming Lab (R22A6781) LAB MANUAL

Prepared by

P.Srinivas, Associate Professor & G.Gayatri, Assistant Professor

On 04.8.2022

Department of Computer Science and Engineering EMERGING TECHNOLOGIES

Vision

"To be at the forefront of Emerging Technologies and to evolve as a Centre of Excellence in Research, Learning and Consultancy to foster the students into globally competent professionals useful to the Society."

Mission

The department of CSE (Emerging Technologies) is committed to:

- To offer highest Professional and Academic Standards in terms of Personal growth and satisfaction.
- Make the society as the hub of emerging technologies and thereby capture opportunities in new age technologies.
- To create a benchmark in the areas of Research, Education and Public Outreach.
- To provide students a platform where independent learning and scientific study are encouraged with emphasis on latest engineering techniques.

QUALITY POLICY

- To pursue continual improvement of teaching learning process of Undergraduate and Post Graduate programs in Engineering & Management vigorously.
- To provide state of art infrastructure and expertise to impart the quality education and research environment to students for a complete learning experiences.
- Developing students with a disciplined and integrated personality.
- To offer quality relevant and cost effectiveprogrammes to produce engineers as per requirements of the industry need.

For more information: www.mrcet.ac.in

MRCET CAMPUS |AUTONOMOUS INSTITUTION-UGC, GOVT. OFINDIA

II Year B.Tech CSE(DS)-I Sem(R22)

L/T/P/C

-/-/2/1

(R22A6781)-RPROGRAMMINGLAB

Course Outcomes : At the end of the Course, the Student will be able to CO1: Show the installation of R Programming Environment.CO2:Utilize and R Data types for developing programs.CO3:Make use of different R Data Structures.CO4:Develop programming logic using R Packages.CO5:Analyze the data sets using R programming capabilities.

LIST OF PROGRAMS:

- Download and install R-Programming environment and install basic packages using install. Packages () command in R.
- 2. Learn al the basics of R-Programming (Data types ,Variables , Operators etc.)
- 3. Implement R-Loops with different examples.
- 4. Learn the basics of functions in R and implement with examples.
- 5. 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.
- 6. Implement different String Manipulation functions in R.
- 7. Implement different data structures in R(Vectors ,Lists ,Data Frames)
- 8. Write a program to read acsv file and analyze the data in the file in R
- 9. Create pie charts and bar charts using R.
- 10. Create a data set and do statistical analysis on the data using R.
- 11. Write R program to find Correlation and Covariance
- 12. Write R program for Regression Modeling
- 13. Write R program to build classification model using KNN algorithm
- 14. Write R program to build clustering model using K-mean algorithm

REFERENCES:

- 1. JaredP.Lander,*RforEveryone:AdvancedAnalyticsandGraphics*,2nd Edition,PearsonEducation,2018.
- 2. S.R.ManiSekharandT.V.SureshKumar,*ProgrammingwithR*,1st Edition, CENGAGE, 2017.

WEBREFERENCE:

1. https://www.r-project.org/

https://www.tutorialspoint.com/r/index.htm

S.No	List of Programs	Page No.
1	Download and install R-Programming environment and install basic packages using install .packages()command in R.	7
2	Learn all the basics of R-Programming (Data types ,Variables Operators etc.)	9
3	Implement R-Loops with different examples.	17
4	Learn the basic of functions in R and implement with examples.	20
5	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.	22
6	Implement different String Manipulation functions in R.	24
7	Implement different data structures in R (Vectors, Lists, Data Frames)	26
8	Write a program to read a csv file and analyze the data in the file in R	30
9	Create pie charts and bar charts using R.	37
10	Create a data set and do statistical analysis on the data using R.	39
11	Write R program to find Correlation and Covariance	40
12	Write R program for Regression Modeling	42
13	Write R program to build classification model using KNN algorithm	43
14	Write R program to build clustering model using K-mean algorithm	46

INDEX

Brief Introduction of R Programming Language :

R is an open-source programming language that is widely used as a statistical software and data analysis tool. R generally comes with the Command-line interface. R is available across widely used platforms like Windows, Linux, and macOS. Also, the R programming language is the latest cutting-edge tool.

It was designed by **Ross Ihaka and Robert Gentleman** at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team. R programming language is an implementation of the S programming language. It also combines with lexical scoping semantics inspired by Scheme. Moreover, the project conceives in 1992, with an initial version released in 1995 and a stable beta version in 2000.

Use of R Programming :

- It's a platform-independent language. This means it can be applied to all operating system.
- It's an open-source free language. That means anyone can install it in any organization without purchasing a license.
- R programming is used as a leading tool for machine learning, statistics, and data analysis. Objects, functions, and packages can easily be created by R.
- R programming language is not only a statistic package but also allows us to integrate with other languages (C, C++). Thus, can easily interact with many data sources and statistical packages.
- The R programming language has a vast community of users and it's growing day by day.
- R is currently one of the most requested programming languages in the Data Science job market that makes it the hottest trend nowadays

1. Installation of R-Studio on windows:

Step – 1: With R-base installed, let's move on to installing RStudio. To begin, goto <u>download RStudio</u> and click on the download button for RStudio desktop.

Step-2: Click on the link for the windows version of RStudio and save the exe file.

Step–3: Run the .exe and follow the installation instructions.

Click Next on the welcome window.

Enter/ browse the path to the installation folder and click Next to proceed.

Select the folder for the start menu shortcut or click on do not create shortcuts and then click Next. Wait for the installation process to complete.

Click Finish to end the installation.





Install the R Packages:-

- First, run RStudio.
- After clicking on the packages tab, click on install. The following dialog box will appear.
- In the Install Packages dialog, write the package name you want to install under the Packages field and then click install. This will install the packageyousearchedfororgiveyoualistofmatchingpackagesbasedonyour package text.

Installing Packages:-

Loading Packages:-

Once the package is downloaded to your computer you can access the functions and Resources provided by the package in two different ways: #load the package to use in the current R session library (package name)

Getting Help on Packages:-

"C:/Program Files/R/R-3.2.2/library"

install.packages("Package Name")
Install the package named "XML".
install.packages("XML")

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

Program Description :

Variables are nothing but reserved memory locations to store values. This means that, when create a variable you reserve some space in memory.

A variable provides us with named storage that our programs can manipulate. A variable in R can store an atomic vector, group of atomic vectors or a combination of many Robjects. A valid variable name consists of letters, numbers and the dot or underline characters. The variable name starts with a letter or the dot not followed by a number.

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations. R language is rich in built-in operators and provides following types of operators.

Data Types :

Numeric :

v <-23.5 print(class(v))

Logical

v <- TRUE print(class(v))

Integer

v <-2L print(class(v))

11 2004 1200 200 200 200 200 200 200 200 200 20		Project (None)
1 Child (Charles)	stery Connections Tutorial	-
*** (6, corr *** (7, corr	of Dataset + 3 122 568 + 1	= 14 × 16
2 (*Mit(Casto)) 4 (*Mit(Casto)) 5 (*Mit(Casto)) 5 (*Mit(Casto)) 5 (*Mit(Casto)) 5 (*Mit(Casto)) 5 (*Mit(Casto)) 5 (*Mit(Casto)) 5 (*Mit(Casto)) 5 (*Mit(Casto)) 7 (**Mit(Casto)) 7 (**Mit(niconnet -	9
11 Braindi I Hayadaha - Tana - T	51.66	
1) Balant 1 Haynol Bas	-1L	and a development of a statement of
11 Depart 1 Kings 1 12 Depart 1 Kings 1 13 Depart 1 Kings 1 14 Depart 1 Kings 1 15 Depart 1 15 Depart 1 15 Depart 1 15 Depart 1 16 Depart 1 17 Depart 1	NA Solecer	prive/besktop/skt skithoset-k.
11 Strategy ()	num [1:40] 0 0 0 1 0 0 0 0	1
11 British Languard Max. We have a second se	30	
1 Septembrie Manadater (1997) 1 Septembrie Manadater (1997)	5	
1 Second 1 Element 1 Element 1 Element 1 Provided 1 Pro	15	
1 Prince 1 Known 1 Known 2 Kno	NULL	
n Topiest : Eleganda - Eleganda - Eleganda - Eleganda - Participarte - Participar	151.	
1 (Protoci 1) (Protoci 2) (Pro		
1 (priorit) 1	ckapes Help Viewer Presentation	
1 Parload 1 Kingar (Kita	- Alfanot - O d	
E 422 - 4		
> + -03 = mage: = mage:		
fried(Cost())		
1 - miteger		
		0755 PM

R-objects.

- Vectors
- Lists
- Matrices
- Arrays
- Factors
- Data Frames

Vectors

When you want to create vector with more than one element, you should use c() function which means to combine the elements into a vector.

Create a vector.

apple <- c('red','green',"yellow")

print(apple)

Get the class of the vector.

print(class(apple))

8 Khadio				- a ×
File Edit Code View Plots Session Build Debug Profile Tools Help				
🔍 + 😋 🞯 + 🔛 😥 🝈 🔿 Go to fityfanction 🔡 + Addres +				Project (Norw)
Star @ ments @ ptar @ ptar @ metrs @ ptar @ stars	9)SLR × 9)United1*× 20	Environment History Conv	actions Tutorial	- 7
0.0 0 R 2 form on fam. 9 2 - 1	Tellan Ne Minute + 2	Carl In Present Detaust +	@ 122 MR - 💰	2 pa + 1 (2)
1 # Create a vector.		R + Gabai Environment +		0
<pre>2 apple <- c("top", "treed", "pellow")</pre>		apple	chr. [1:3] "ced" "green" "vellos"	
3 print(apple)		b	10.	
5 # Get the class of the vector.		factorial	1	
<pre>6 print(class(apple))</pre>		1	3.	
7		10	num [1:40] 26.3 27 27.2 27.5 27.7	
		isPrime	0	
		k	4.	
		nane	"sree"	
		nun	-11	
		P	"source(\"C:/isers/sreek/OneDrive/Desktep/S	AL SANTHOSHI-M.
		Files Plats Packages Hall	lp Viewer Presentation	
		🔅 🐠 🏓 Zoon 🛛 🚚 Espot	- 0 1	
71 (bp.(eve) :	R Senat a			
Console Terminal - Background Jobs -	-0			
R420/-P				
> Y <-2.				
> print(class(v))				
[1] "integer"				
> apple <- c('red', 'oreen', "vellow")				
> print(apple)				
 "red" "green" "yellow" 				
y # Get the class of the vector.				
> print(class(apple))				
 "character" 				
× 1				
🖬 🔎 👩 💽 🦮 🤰 🙆 👩 👰		0	🛃 28°C Mostly cloudy 🔷 🖗 🚳 🔊 /() 44	DNG 0257 PM

Lists

A list is an R-object which can contain many different types of elements inside it like vectors, functions and even another list inside it.

Create a list.

list1 <- list(c(2,5,3),21.3,sin)

Print the list.

print(list1)



Matrices

A matrix is a two-dimensional rectangular data set. It can be created using a vector input to the matrix function.

Create a matrix.

M =matrix(c('a','a','b','c','b','a'),nrow=2,ncol=3,byrow= TRUE)

print(M)

RStudio				- a ×
• • • • • • • • • • • • • • • • •				Project: (None) •
SLEX 0 HOURS 0 LOSE 0 LALEX 0 HOURS 0 LOSE 0 HAR A 0 LALEX (Environment History	Connections Tutorial	-0
Ci 10 H 2 Source on Save Q 2 - 1	-+ Run -+ - Source - =			E ta + 1 5 +
1 # Create a matrix.				9
2 M =matrix(c('a', 'a', 'b', 'c', 'b', 'a'), nrow=2, ncol=3, byrow= TRUE) 3 orint(M)		O list1	List of 3	Q. *
4		N	chr [1:6] "a" "c" "a" "b" "b" "a"	
5 6		values		
		a	5L	
		age	-10-	
		appre	tor [L1:5] red green yerrow	
		factorial	1	
		1	3	
		10	man [1:40] 26.3 27 27.2 27.5 27.7	*
		Files Plots Packag	es Help Viewer Presentation	- 0
			Tipot - 0	
61 (Optime) s	R Script 0			
Console Terminal × Background Jobs ×	-0			
(R 8420/ -)				
[1] 2 5 3	•			
[[2]] [1] 21.3				
[[3]] function (x) .Primitive("sin")				
<pre>> # Create a matrix. > M =matrix(c('a','a','b','c','b','a'),mrow=2,mcol=3,byrow= TRUE) > print(0)</pre>				
[,1] [,2] [,3] [1,] "a" "b" [2,] "c" "b" "a"				
■ Q © C 🐂 🦻 🛽 🖻	*		💊 🛃 28°C Mostly cloudy ^ වූ 🗔 🖬 දු අ	ENG 01.09.2021

Arrays

While matrices are confined to two dimensions, arrays can be of any number of dimensions. The array function takes a dim attribute which creates the required number of dimension. In the below example we create an array with two elements which are 3x3 matrices each.

Create an array.

```
a <- array(c('green','yellow'),dim= c(3,3,2))
print(a)</pre>
```

Studio			- o ×
ie bit Code View Plots Session Build Debug Profile Tools Help			Theorem Proved
THE DIMONTER DIPONTER DIPONTER DIPONTER DIPONTER	els =	Environment History Connections Tutorial	= -
CODE A Constant on Section 19 2 - 1	🔫 Run 🔭 🧈 Source 🍷 😤	🔐 🚽 📅 Import Dataset + 🕒 122 Mill + 🥑	= Liz + 1 (3
2 a <- array(c('brent', 'bellow'), dim- c(3,3,2))		R • Giobal Environment •	٩,
1 print(a)		Olist1 List of 3	٩
4		M chr [1:6] "a"	"c" "a" "b" "b" "a"
á 1		values	
		a chr [1:18] -91	reen yellow green yellow green yel.
		ape 10 apple chr. [1:3] *rei	f "oreen" "vellow"
		b 10	Jerree
		factorial 1	
		i 31.	
		10 num [1:40] 26.	.3 27 27.2 27.5 27.7
		Files Flots Packages Help Viewer Presentat	ion
		🐢 🕸 🔎 Zeom 🚢 Export - 0 🎸	
61 (Septern) :	R Sorjet a		
Console Terminal × Background Jobs ×			
R420//0			
> print(a) 1			
[,1] [,2] [,3] [1,] "green" "yellow" "green" [2,] "yellow" "green" "yellow" [3,] "green" "yellow" "green"			
, , 2			
[,1] (,2) (,3] [1,] "yellow" "green" "yellow" [2,] "green" "yellow" [3,] "yellow" "green" "yellow"			
>			
		🚯 🛃 28°C Mo	stly cloudy 🔿 🛱 🔯 😰 🔬 🔃 🕅 🔂 🖬 📰

Factors

Factors are the R-objects which are created using a vector. It stores the vector along with the distinct values of the elements in the vector as labels. The labels are always character irrespective of whether it is numeric or character or Boolean etc. in the input vector. They are useful in statistical modeling.

Factors are created using the **factor**() function. The **nlevels** functions gives the count of levels.

Create a vector.
apple_colors<- c('green','green','yellow','red','red','green')</pre>

Create a factor object.
factor_apple<- factor(apple_colors)</pre>

Print the factor.
print(factor_apple)
print(nlevels(factor_apple))
[1] green green yellow red redred green
Levels: green red yellow



Variables:

The variables can be assigned values using leftward, rightward and equal to operator. The values of the variables can be printed using **print**() or **cat**() function. The **cat**() function combines multiple items into a continuous print output.

Assignment using equal operator. var.1=c(0,1,2,3) # Assignment using leftward operator. var.2<- c("learn","R") # Assignment using rightward operator. c(TRUE,1)->var.3

print(var.1) cat ("var.1 is ",var.1,"\n") cat ("var.2 is ",var.2,"\n") cat ("var.3 is ",var.3,"\n")

Khuda	- 0
He lids Code View Picts Sesson Build Debug Pithle Tools Help	
2 + 🥦 🐲 - 🔛 🕼 🗇 / Al Gristerbeiter 👘 🗒 + Aldes +	B Project (N
ILE: Press . Pater . Pater . Press . Press . Press . Press . Press . Presser . It and	Environment History Connections Tutarial
GILD Hotomories Q.Z.II -the ** Hinne *	🖉 🔐 🔐 Immergent Delawet + 🍙 122 AMB + 🔏 🖉 👘 🖉
1 # Assignment using egual operator.	R + 🚯 Occur Environment + Q
3 # Assignment using leftward operator.	18561C 1866 [1.46] 0 0 0 1 0 0 0 0 1
4 var.2<- c("learn", "8")	51 5
5 # Assignment using rightward operator.	52 15
7	str NAL
8 print(var.1)	sam 15i
9 cat ("var.1 is ", var.1, "u")	v 21.
10 GRU ("WR-2 IS ", WR-2, "W") 11 GRU ("WR-2 IS ", WR-3, "W")	var NA, Integer_
12	var.1 num [1:4] 0 1 2 3
	var.2 chr [1:2] "Jearn" "R"
	Files Plats Packages Hulp Varver Presentation
	a a 2 Ateor - 9 2
121 Stephenil 5 Klorg	
Consule Terminal - Badigescel Jobs - mi	5
(a) 8420 - 477	
> # Assignment using equal operator.	
> Var.1=C(0,1,2,3) > # Assignment using leftward operator.	
> var, 24- c("Tearn", "R")	
> # Assignment using rightward operator.	
S CONCEPTION S	
> print(var.1)	
(1) 0 1 2 3 a fat (from 1 is 7 our 1 Tori)	
var.1 is 0123	
> cat ("var.2 is ",var.2,"vi")	
var.2 is learn R	
var.3 is 11	
×	
	the 🥂 20°C Mostly cloudy

R Operators :

Types of Operators

Arithmetic Operators v <- c(2,5.5,6) t <- c(8, 3, 4)

print(v+t)

Relational Operators

v <- c(2,5.5,6,9)

t <- c(8,2.5,14,9)

print(v>t)

Logical Operators

```
v <- c(3,1,TRUE,2+3i)

t <- c(4,1,FALSE,2+3i)

print(v&t)

Assignment Operators

v1 <- c(3,1,TRUE,2+3i)

v2 <<- c(3,1,TRUE,2+3i)

v3 = c(3,1,TRUE,2+3i)

print(v1)

print(v2)

print(v3)
```

Notes	- 8 x
• • • • • • • • • • • • • • • • • • •	T Pranti David
In the Case the the theory that the theory that the theory $h=0$. The theory $h=0$, the theory (theory here the theory here the theory (theory here theory here the theory (theory here theory here theory (theory here theory here theory here theory (theory here theory here theory here theory here theory (theory here theory h	Name Non-Construction Annual Construction Annual Construction Annual Construction 10 Mark Mark Construction Non-Construction Non-Construction Non-Construction 10 Mark Construction Non-Construction<
	a ren 3 an 100 111 11 1 1 This Rea Balays A War Anadama and an C 2 an 2019 An 100 An 100 An 100 An An 100 Anno 100 An
31 OptionEx Kings	
Canada Servinal - Background Jobs	
M. M. S. M. S. M. B. M. M	

3 Implement R-Loops with different examples.

Program Description :

A for loop is the most popular control flow statement. A for loop is used to iterate a vector. It is similar to the while loop. There is only one difference between for and while, i.e., in while loop, the condition is checked before the execution of the body, but in for loop condition is checked after the execution of the body.

Create fruit vector
fruit <- c('Apple', 'Orange', "Guava", 'Pinapple', 'Banana', 'Grapes')
Create the for statement
for (i in fruit){
 print(i)
}</pre>



Creating a matrix

mat <- matrix(data = seq(10, 21, by=1), nrow = 6, ncol =2)
Creating the loop with r and c to iterate over the matrix
for (r in 1:nrow(mat))
for (c in 1:ncol(mat))
 print(paste("mat[", r, ",",c, "]=", mat[r,c]))
print(mat)</pre>

Read (M-Mt)		-	0	×
File Edit View Misc Packages Windows Help				
R Console				
<pre>> - (Souther sentem = (Souther sentements); (), (), (), (), (), (), (), (), (), (),</pre>				
📰 🔎 Type here to search 🛛 🚱 📌 🛤	😰 😨 28°C Mostly doudy 🔨 ễ 切 40 8	NS 20.01	PM	Ę

R while loop :

A while loop is a type of control flow statements which is used to iterate a block of code several numbers of times. The while loop terminates when the value of the Boolean expression will be false.

In while loop, firstly the condition will be checked and then after the body of the statement will execute. In this statement, the condition will be checked n+1 time, rather than n times.

```
v <- c("Hello","while loop")
cnt <- 2
while (cnt < 7) {
    print(v)
    cnt = cnt + 1
}</pre>
```



4. Learn the basics of functions in R and implement with examples.

Program Description :

A function is a set of statements organized together to perform a specific task. R has a large number of in-built functions and the user can create their own functions.

In R, a function is an object so the R interpreter is able to pass control to the function, along with arguments that may be necessary for the function to accomplish the actions.

The function in turn performs its task and returns control to the interpreter as well as any result which may be stored in other objects.

Built-in Function

Create a sequence of numbers from 32 to 44. print(seq(32,44))

Find mean of numbers from 25 to 82.
print(mean(25:82))

Find sum of numbers frm 41 to 68.
print(sum(41:68))

• • • • • • • • • • • • • • • • • • •	le Edit Code View Plats Session Build Debug Profile Tools Help					
11 Dealer Punct	🔹 👒 🐨 - 🔛 🔐 🍈 🖈 Go to Refunction 🔤 - Addins -				- 8	Project: (No
(1) Control (1) Contro	th = 0 mont = 0 pth =	41°× >	Environment History Conne	ctions Tutorial		-
P (refine and second moders from 2 to 6.4. 0. 4.9. @ Goldmannee* 0. P (refine and second moders from 2 to 6.2. 13 13 13 P (refine and second moders from 2 to 6.2. 14 16 10 10 P (refine and second moders from 2 to 6.2. 16 16 16 16 16 P (refine and refine 2 to 6.2. 16	COLD H 2 Some in Same (A. 2000) Cold Hard State	+ Prove + 2	😭 🖬 🛛 Proport Dataset +	a 121 MB + 🥑		De - 10
1 Figure 1 Figure 1 1 Figure 2 Figure 2 1 Figure 3 Figure 3 1 Figure 4	1 F Create a sequence of numbers from 32 to 44.		R + 🚳 Global Environment +		9,	
A Print State of Andreas from 25 to 22. A Print State of Andreas from 25 to 22. A Print State of Andreas from 25 to 22. A Print State of Andreas from 25 to 23. A Print State	2 print(seq(32,44))		34	3		
b arr M. K. Kopper	4 # Find mean of numbers from 25 to #2.		52	15		
Print and " whether the dist to dist. Print and " whether the dist to dist.	5 print(mean(25:82))		var	NA integer		
# Proteiner (1.99) With Transmission (1.99)	7 # Find sum of numbers frm 41 to 68.		var1	"SREEKAR"		
9 9 10 10 10 10 10 10 10 10 10 10 10 10 10	8 print(sum(41:68))		var2	"SNIKITHA"		
I month in the second secon	9		var3	"MACHAWI"		
V Particular None 1 V0 Pa			x	int [1:100] 1 2 3 4 5 6 7 8 9 10		
V (Notional 1) (1) (Notional 1) (2) (1) (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2			У	int [1:100] 1 3 6 10 15 21 28 36 45 55		
1 None Nage Nage <t< td=""><td></td><td></td><td>2</td><td>1300.3</td><td></td><td></td></t<>			2	1300.3		
The Tax Mage Not Your Anustan Tore Tax Mage Not Your Anustan (a) (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2			ENICCIOIS			
10 Instant 1 1 Sampati - 0 2 10 Instant 1 Instant 1 1 Sampati - 0 11 Instant 1 Instant 1 Instant 1			Files Plots Packages Halp	Viewer Presentation		-
Name Operation Operation We Address max max We Address max max We Address max max Dir 2014 M S NN V N N H 40 44 44 max max F of time and P max(m Fr Rd 1 N N K 40 max max Dir 2014 M S NN V N H 40 44 44 max max F of time and P max(m Fr Rd 1 N N K 40 max max Dir 2014 M S NN V N H 40 44 max max F of time and P max(m Fr Rd 1 N N K 40 max max Dir 2014 M S N N N N N N N N N N N N N N N N N N			🖕 📫 🎤 Zoom 🛛 🚚 Espert	- 0 1		
Control Dependence Defendence # 04.11/* Defendence Defendence Defendence # 04.11/* Defendence Defendence Defendence # 04.11/* Defendence Defendence Defendence D 12 33 34 35 33 35 38 40 40 42 40 44 Defendence Defendence Defendence # 100 Best of Analyses from 25 88 82. Defendence Defendence Defendence [13 33.3] Defendence Defendence Defendence Defendence # 100 Best of Analyses from 35 88 82. Defendence Defendence Defendence [13 33.3] Defendence Defendence Defendence Defendence [13 33.3] Defendence Defendence Defendence Defendence [13 34 35 Defendence Defendence Defendence Defendence Defendence	1.1 (Top Level) 1	R Soriet a				
(B) 43.1	Censole Terminal × Rackground Jobs ×	- 7				
<pre>> forsts : support of matters from 10 to 4. (1) 33 35 35 35 35 35 39 46 44 24 04 4 > forst and i matters from 20 to 82. = bits : bits : > forst and i matters from 20 to 82. = bits : > forst :</pre>	@ 8420/-P					
(1) 213 24 25 28 27 38 39 40 42 0 44 5 4 164 mass of analysis for 25 18 42. (2) 5 4 16 mass of analysis for 25 18 42. (2) 5 4 16 mass of analysis for 44 to 68. (2) 5 4 16 mass of analysis for 44 to 68. (2) 5 4 16 mass of analysis for 44 to 68.	> # Create a sequence of numbers from 32 to 44.					
2 - A for is an of the S - A for the S - S - S - S - S - S - S - S - S - S	> print(seq(32,44))					
<pre>> # find end of makers free 20 18 82. [10 33.3 4] 20 33.5 4 2 = find end of makers free 41 to 64. > = find end (180) 20 5000</pre>						
DD 31.5 DD 31.5 C folds and folds (1.6) DD 315 D 315 D 315 C 415 D 315 C 415 D 315 C 415 C	> # Find mean of numbers from 25 to 82.					
- refer and makener fra 41 to 60. - protection (4.100) 1) 110	> print(mean(25:82)) [53:59:5					
* Find and of makers for 44 to 64. [1] 13430	>					
D 112	> # Find sum of numbers frm 41 to 68.					
	> print(sum(41:08)) [1] 1526					
	>					
					171	C 784

User-defined Function

We can create user-defined functions in R. They are specific to what a user wants and once created they can be used like the built-in functions. Below is an example of how a function is created and used.

Create a function to print squares of numbers in sequence.

```
new.function <- function(a) {
    for(i in 1:a) {
        b <- i^2
        print(b)
    }
}</pre>
```

Call the function new.function supplying 6 as an argument. new.function(6) **5.** Implement data frames in **R**. Write a program to join columns and rows in a data frame using cbind() and rbind() in **R**.

Program Description :

A data frame is a table or a two-dimensional array-like structure in which each column contains values of one variable and each row contains one set of values from each column.

#Creating vector objects

```
Name <- c("Shubham Rastogi","Nishka Jain","Gunjan Garg","Sumit Chaudhary")
```

```
Address <- c("Moradabad","Etah","Sambhal","Khurja")
```

Marks <- c(255,355,455,655)

#Combining vectors into one data frame

```
info <- cbind(Name,Address,Marks)</pre>
```

#Printing data frame

```
print(info)
```

Creating another data frame with similar columns

new.stuinfo <- data.frame(</pre>

```
Name = c("Deepmala","Arun"),
```

Address = c("Khurja", "Moradabad"),

Marks = c("755","855"),

stringsAsFactors=FALSE

```
)
```

#Printing a header.

cat("# # # The Second data framen")

#Printing the data frame.

print(new.stuinfo)

Combining rows form both the data frames.

all.info <- rbind(info,new.stuinfo)</pre>

Printing a header.

cat("# # # The combined data frame\n")

Printing the result.

print(all.info)



6. Implement different String Manipulation functions in R

Program Description :

String manipulation basically refers to the process of handling and analyzing strings. It involves various operations concerned with modification and parsing of strings to use and change its data. R offers a series of in-built functions to manipulate the contents of a string. In this article, we will study different functions concerned with the manipulation of strings in R.

Concatenation of Strings

String Concatenation is the technique of combining two strings. String Concatenation can be done using many ways:

• **paste**() **function** Any number of strings can be concatenated together using the **paste**() function to form a larger string. This function takes separator as argument which is used between the individual string elements and another argument 'collapse' which reflects if we wish to print the strings together as a single larger string. By default, the value of collapse is NULL

pr-1

R program for String concatenation

Concatenation using paste() function
str <- paste("Learn", "Code")
print (str)</pre>

				Project (None
1.R = 0 moo.R = 0 pr.R = 0 pr.R = 0 mor.R = 0 pr.R = 0 mR = 0 pr.R = 0 ST.R =	• Uetfiedt · · · · · · · ·	Environment History Cor	nections Tutorial	-5
🖓 🔄 🗗 🔛 🖉 Source on Save 🔍 🎽 📲	Run 🏎 🕂 Seurce - 2	🔐 🖂 🕮 Import Dataset + 🌙 121 MB + 🧹		≡ Let + 143
1 # R program for String concatenation		8 • 🦓 Global Environment		9
4 Concatenation using paste() function		34 (Thi	A/ "Lotes Code"	
<pre>4 str <- paste("Learn", "Code")</pre>		50	15	
S print (str)		Var	NA integer	
0		var1	"SREEKAR"	
		var2	"SNIKITHA"	
		var3	"MACHAY1"	
		x	int [1:100] 1 2 3 4 5 6 7 8 9 10	
		У	int [1:100] 1 3 6 10 15 21 28 36 45 55	
		z	1300.5	
		Functions		
		Files Plats Packages H	elp Viewer Presentation	-0
		The state of the s		
6.1 (Tap Level) 2	R Solpt 1			
Console Terminal : Background Jobs ::	-0			
R 8420-470				
(m) 2420-170 # R program for String concatenation ≥ Comparamentation of the system.				
(R 8420 - 470) ★ R Brogram for String concatenation > # Concatenation using paste() function > tr c= cast(Cast) ⁽¹⁾ (Colo ²)				
<pre>/gR R420 - +/* > # 8 program for String concatenation > # concatenation using paste() function > str <- paste("Learn", "code") > print (str)</pre>				
<pre>(@ R420 -+/0</pre>				
<pre>(@ Adds</pre>				
<pre>(@ ALG>-+^0 * Represent for String concitenation > Concitenation using parte() function > concitenation (Concer) > prior (Cite)) (team Code")</pre>				
<pre>(# ALD</pre>				
(# ALD of rolling containeration + Constantion using match) function - Constantion (Section Control (Control (Con				
8 Add				
<pre>% ALD</pre>				
(# ALD - view for Joring consistention * Constantion miles parts) Precision = Constantion (* Constantion) = Constantion (

pr-2

Concatenation using cat() function
str <- cat("learn", "code", "tech", sep = ":")
print (str)</pre>

Kitudio		- o ×
Re Edit Code View Plots Session Build Debug Postile Tools Help		
• • • • • • • • • • • • • • • • • • •		Project (None)
18 0 mon 8 0 pr58 0 p18 0 for 8 0 pc8 0 or 8	Environment History Connections Tutorial	
Contraction State 9 2 - 1	💣 🔂 🕐 inport Dataset + 🗳 121 MB + 🧹	🗏 Lat + 1 🗊
1 # Concatenation using cat() function	R • 🍈 Global Invironment •	9
<pre>2 Str <- Cat("learn", "code", "tech", sep = ":") 3 print (str)</pre>	38 83 100	
4	SUP NOLL	
	var Na İstanar	
	var1 "SREEKAR"	
	var2 "SNIKITHA"	
	var3 "MADHAVI"	
	x int [1:100] 1 2 3 4 5 6 7 8 9 10	
	y int [1:100] 1 3 6 10 15 21 28 36 45 55	53
	z 1300.5	
	Functions	
	Files Plots Packages Help Viewer Presentation	es
	1 1 1 2 2 Report - 1 0 1 1	
41 Clap Levels : R Script :		
Console Terminel : Background Jobe :		
Q 8420/ 0		
> # Constemation sping call) function pir <- call'tear," "code," recht", see = "(") MAL MAL		
🖬 🔎 😨 💘 🦻 🔘 💼 🛱	🚯 🌏 311C Partly cloudy 🔨 亞 國 🕫 成하다	ENG 07-28 PM

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

Program Description :

Vectors are the most basic R data objects and there are six types of atomic vectors. They are logical, integer, double, complex, character and raw.

Lists are the R objects which contain elements of different types like – numbers, strings, vectors and another list inside it. A list can also contain a matrix or a function as its elements. List is created using list() function.

```
Vectors
# Create a vector.
apple <- c('red','green',"yellow")
print(apple)
```

Get the class of the vector.
print(class(apple))



Lists

A list is an R-object which can contain many different types of elements inside it like vectors, functions and even another list inside it.

Create a list. list1 <- list (c (2, 5, 3), 21.3, sin)

Print the list.
print (list1)
[[1]]
[1] 2 5 3
[[2]]
[1] 21.3

[[3]] function (x) .Primitive("sin")

Ritudio				- a ×
File Edit Code View Picts Session Build Debug Profile Tools Help				
• • • • • • • • • • • • • • • • • • •				Project (None)
\$1.8 × 9. mcs.8 × 9.pt.8 × 9.pt.8 × 9.fm.8 × 9.pt.8 × 9.at.8 × 9.st.8 × 9.tm	🗆 👝 🔍 🗆 "Ibabb	Environment History	Connections Tutorial	
💿 🕼 🔛 🖉 Source on Saw 🔍 🎢 🕛 🔅 🗇 Run	Searce - ≥	🐨 📊 🐨 Import Datas	et + 🗳 122 Mill + 🔏	= Let - 1 🙄 -
1 # Create a list. 2 list1 co list(c(2.5.2) 21.3 sin)		R + 🚳 Global Environm	ert •	Q,
3		Data		
4 # Print the list.		O Data_Frame	3 obs. of 3 variables	
o 6 [[1]]		O df	40 obs. of 2 variables	
0 7 11 2 5 3		0 list1	List of 3	Q,
0 8 [[2]]		м	chr [1:6] "a" "c" "a" "b" "b" "a"	
9 [1] 21.3		values		
0 11 [[3]]		a	SL.	
12 function (x) .Primitive("sin")		age	10	
13		· factorial	1	
		Files Plots Packages	Help Viewer Presentation	
		🐠 📫 🏓 Zoon 🛛 🛺	Epot = 🔍 🧹	
86 (Top Level) 1	R Script #			
Console Terminal X Background Jobs X	-0			
🙀 R420/ 🕫				
> list1 <- list(c(2,5,3),21.3,sin)				
> # Print the list.				
> print(list1)				
[[1]]				
[1] 2 3 3				
[[2]]				
(1) 111				
[[3]] function (x) .Primitive("sin")				
> [[1]]				
>				
				0751 PM
·• 🖓 I 🥲 🐨 🖷 🥭 🐨 🖪 🛤			10 Humid ^ 🖓 🖾 🗰 🕼	Q10 ENU 01-09-2023

Matrices

A matrix is a two-dimensional rectangular data set. It can be created using a vector input to the matrix function.

```
# Create a matrix.
M =matrix ( c ('a', 'a', 'b', 'c', 'b', 'a'), nrow=2, ncol=3, byrow= TRUE)
print (M)
```

Kitadia Eka City Conte Verse Renore Build Datum Books Note Liain			- a ×
• • • • • • • • • • • • • • • • • • •			B Project (Nord)
Tax Barrel Barrel Barrel Parel Parel Barrel	Environment Hatary ar in Providence a - Canta freedom Data O cata, Franc O or w Values a age b factorial	$\label{eq:constraint} \begin{array}{l} \text{Constative} & Varial \\ n^{-1} \rightarrow 0 \left(2 1 0 \right) = \sqrt{n} \\ n^{-1} & n^{-1} \\ n^{-1} $	а странция 1 м - 9 0 1
	Non Park Parkager	Nega Viewer Presentation	-0
81 Obj. Geol. 2 Kingt	8		
Canasie Terretral - Background Jobs			
2 Constant Sectors			
🔳 🔎 😨 🐂 🎐 🔕 💼 🕫		🍪 📩 30°C Mostly cloudy ^ වු 🔯 🗉 ස්	41 ENG 8750 FM

Data Frames :

```
# Create a data frame
Data_Frame<- data.frame (
Training = c("Strength", "Stamina", "Other"),
Pulse = c(100, 150, 120),
Duration = c(60, 30, 45)
)
```

Print the data frame Data_Frame

RStatio			- 0 ×
File Edit Code View Plots Session Build Debug Profile Tools Help			
🔍 🔹 🦦 🛃 🔂 🍈 🏕 Go to Neyfunction 🛛 🖸 * Addins *			Project: (None)
S1R× 0 mccR× 0 p5R× 0 p1R× 0 torR*× 0 pcR*× 0 nLR*× 0 e1R× 0 S1R× 0 United*× >>	Environment History	Connections Tutorial	
Configure and the second secon	🛷 🖬 📅 Insert Data	art + 🕼 121 MB + 🖌	E tar + 1 (2) -
1 # Create a data frame	R - Gobal Environm	ert ·	0.
2 Data_Frame<- data_frame(Training = c("framethe" "frames" "Other")	Data		
4 Pulse = c(100, 150, 120).	O Data_Frame	3 obs. of 3 variables	
5 Duration = c(60, 30, 45)	O df	40 obs. of 2 variables	
6)	Values		
8 # Print the data frame	a	5L	
9 Data_Frame	age	"10"	
10	b	10.	
	factorial	1	
	1	3.	
	4 10	num [1:40] 26.3 27 27.2 27.5 27.7	
	Files Plots Packages	Help Viewer Presentation	
	da ab 1 P Zoon 1 - 2	fand - 0	
101 (lop Level) ± R.Script ±			
Consele Terminal × Background Jobs ×			
@ 8420/ @			
- FOLD and Add frame - FOLD and Add frame Training - (Chrometh, "Training - (Chrometh, " - Polar - (Chrometh, "Training - (Chrometh, " -) and this - (Chrometh, " -) and the - (Chrometh, "			
# D O C # 3 0 m @		😮 📩 30°C Mostly cloudy \land 🖗 🖾 🛋 d	© ENG 07:49 PM

8. Write a program to read a csv file and analyze the data in the file in R

Program Description :

In R, we can read data from files stored outside the R environment. We can also write data into files which will be stored and accessed by the operating system. R can read and write into various file formats like csv, excel, xml etc.

Getting and printing current working directory.

print(getwd())

Setting the current working directory.

setwd("C:\Users\sreek\OneDrive\Desktop\SAI SANTHOSHI-MRCET-2023")

Getting and printing the current working directory.

print(getwd())

Ne Lot format Vew Hop	Sant Al	- A 1		A	
0,1586,541479,51275_04140,0651 ,5hubham,613.3,2012-01-05,17 Arro114.525.2.2012.00.23.(merations	The state of the second st		(i=)		
umbaladi, cargo ang sa pang sa Sa pang sa br>pang sa pang sa pan	A real context in the second sec	Clast recaling As	Terr Terr Marcust Office - Marcust Office - R Search File R Search File	* *	

Reading a CSV file

data <- read.csv("record.csv")
print(data)</pre>

Output :



Analyzing the CSV File csv_data<- read.csv("record.csv") print(is.data.frame(csv_data)) print(ncol(csv_data)) print(nrow(csv_data))

fStudio					- 0 >
Edit Code View Plats Session Build Debug Profile Tools Help	p				
+ 🎕 💣 + 🔒 🔐 📥 🗚 Go to Ny/Lection 👘 🖽 + Adde	15 -				🐌 Projecti Olione
ante Olartes Olivertante Olartes Olartes Olar	X Blas (X Bas (X Bas (X Bas	- 7	Environment Matery	Connections Totacial	
10 D D C formers Save 9 2 1		Ne Prove + 2	CP III IP Inned Pat	and a 3 120 148 - 4	E 144 x 1 (5)
1			R + Castal Invites	next -	0
2 csv_data<- read.csv("record.csv") 3 print(is_data_frame(csv_data))			Data		
4 print(ncol(csv_data))		Ocsv_data 8 obs. of 5 variables	8 obs. of 5 variables		
<pre>5 print(nrow(csv_data))</pre>			O data	8 obs. of 5 variables	
i			Values		
			a	5L	
			age	"10"	
			b	100	
			factorial	1	
			(amalas	SL.	
			Files Plots Package	a Help Viewer Presentation	-
			6 10 P 2000 14	Essart - O d	
11 (Top Level) :		R Script a			
sole Terminal x Background Jobs x		- 0			
8430 - Other has Debin Deber Still McDobe MCD. 2011					
sonre et C. 2007 (1997) a resource de l'independent (1997) Balaktapi/sida : Santre ja me	den "Hele i "dolove". R")				
					1155 AM

Getting the maximum salary

Creating a data frame. csv_data<- read.csv("record.csv") # Getting the maximum salary from data frame. max_sal<- max(csv_data\$salary) print(max_sal)

Binds - B × Binds - B × C = B × K - B × K <th></th> <th></th> <th></th> <th></th> <th></th>					
The Disk Stars St	RStudio				- a ×
Control of the second sec	File Edit Code View Plots Session Build Debug Profile Tools Help				
Expert Part Part Part Part Part Part Part Pa	● + 🧠 🐨 + 🔒 🕼 🔅 A Colo Networking 💠 Adding +				Project (Nana) *
Image: Production of the sector frame of the sect	● prA × ● pr1x × ● Unitied3 × ● pr2A × ● pr3A × ● mrA × ● prA × ● pr4A × ● pr5A ×	-0	Environment Histor	ry Connections Tatorial	-0
Provide Address	(0.0) 🔊 🔛 🖉 fause on face 🔍 🎢 + 📋 🔅 🖼 👘 Rus 🤒	• Source + R	🖝 🖬 💷 Import 0	Durtaset = 😂 128 Milli = 🛛 🎻	🗏 UA - 1 🕥 -
• Backting (Status) (Sta	1 F Creating a data frame.		R 🔹 📫 Global Envir	orment -	٩,
A Porticit the section where the frame. A porticit the section where the sectio	3		O data	8 obs. of 5 variables	
Provide a final decision of the set of	4 # Getting the maximum salary from data frame.		Values		
To be set of the	6 print(max_sal)		a	SL.	
Price 1 P			age	"10"	
1 (Normal) Konstanting Statement and Stateme			b	10.	
In the second of the seco			1		
1 Restort 1 File Restort 1 Restort 1 1 Restort 1 Restort 1 Restort 1			1sPrime	0	
A Deciding of the second se			k	4L	
It Nature 1 N					
11 Notani 1 Tangan Anton Sangan Anton Sang			Files Plots Packs	ages Help Viewer Presentation	- 7
1 Start 2 Start 3 Start 4 Start 4 Start 5				- Deport + 0 1	
Term Term <td< th=""><th>1 Percent +</th><th></th><th></th><th></th><th></th></td<>	1 Percent +				
The fail conversion for the conversion for the conversion of the c		K JAPPA V			
<pre>start Control to a main frame. to Control to a main frame. to Control to a main control to a main frame. to the control to t</pre>	Console Terminal II Background Jobs II	= 0			
<pre>> societies readicativement card) + double card of a statistic readication card of a stat</pre>	<pre>VW R420 - C/Uses/week/OweDeskip/SA/SAVTHOSHEAMCET2023/ // > # Creating a data frame.</pre>				
<pre>- # difference states that from the first from the states of the st</pre>	> csv_data<- read.csv("record.csv")				
	<pre>> # disting the maximum salary from data frame. > max_salar_max(csv_dataScalary) > print(max_sal) [1] 932.8</pre>				
📲 💭 RT 🤨 😴 📻 🍠 🖪 🧐 🖏 🔯	🖬 🔎 H 🔕 😧 🐂 🎐 💼 🔕 🐗 🙉			😮 🏼 🛃 33°C Mostly cloudy 🔿	2 🖾 🗰 💠 di ENG 1200 PM 🖓

Getting the details of all the persons who are working in the IT department

Creating a data frame.

csv_data<- read.csv("record.csv")

#Getting the detais of all the pweson who are working in IT department
details <- subset(csv_data,dept=="IT")
print(details)</pre>

Studio		- 0 ×
te sir con new rich secon suit belog riche ton reg		The Design of Design
alter	Destances Harpy Connection Tutoria	
Reconstruction framework (and provide and any workship to 17 department provide and any other department) protection framework department()	8 - (B) distributions Data 8 dec. of 3 variables 0 cov_data 8 dec. of 3 variables 0 deca 15 variables 0 deca 15 variables 0 deca 15 variables 1 variables a 51 variables 200 "20"	
	factorial 1	
	Files Ports Packages Help Varuer Presentation	-5
17. Obsideed it		
Control - Reduced Maria - Th		
44. Life (Concentrational International Processing) 41. Life (Concentrational International		
🖬 🔎 🗏 😨 🍖 🧶 🔳 🔕 🙀	🍓 📑 33°C Mostly-doudy 🔷 🖗 🗐	■ + 61 Dis ^{1230 M}

Getting the details of the persons whose salary is greater than 600 and working in the IT department.

Creating a data frame.

csv_data<- read.csv("record.csv")

#Getting the detais of all the pweson who are working in IT department

details <- subset(csv_data,dept=="IT"&salary>600)

print(details)

Studie 1				- 0 ×
Ne list Cole Yee Pics Sesson Build Debug Pictie Tools Hep				There have
Parts Parts Presents Parts Parts Parts Parts Parts P	-14.	Records Manual	Constant Resid	
(i) C H discuster Q Z + U	Mary Mar Mary and D	The last of the same line		
1 # Creating a data frame.		E + A Cattal Incom	mail -	9
 csdata- read.csv("record.csv") Acting the detais of all the percent who are working in IT department details - subset(csv_data_dept-"lT"isalary=000) 		Data		
		O csv.data	# obs. of 5 variables	
		O data	8 obs. of 5 variables	
6 print(details)		O details	1 obs. of 5 variables	
		saloes		
			51.	
		age	-10-	
		B	10.	
		140.5507141		
		förer Plate Package	n Help Viewer Presentation	
		A 10 Page 14	Report - Q	
Arts Digital 1	Klogt 1			
Console Terminal - Bodiground Jobs -	=0			
*				
= A = 0 0 = s a a			🔥 📑 33°C. Ministly cloudy 🖉	• 0 03 • 0 01 ms 20-07-203 □

Getting details of those peoples who joined on or after 2014.

Creating a data frame.

csv_data<- read.csv("record.csv")

#Getting details of those peoples who joined on or after 2014

details <- subset(csv_data,as.Date(start_date)>as.Date("2014-01-01"))

print(details)

O KStudie				- 5 ×
He lidt Code Vew Picts Session Italic Debug Profile Tools Help				
🔍 + 🕲 🐲 + 🖓 🕼 🖉 de la traductor 👘 🗐 * Abbis *				Trajacit (Nora) *
O and t O prior O inemest - O point - O menter O paint - O point -	-0	Environment Matery Conner	Have Tutorial	- 7
O TO H 2 form of the Q F - 1	Contract of Street	at hi IP report Dataset +	0 101 MB - 🖌	10 to a 1 cl +
1 # Creating a data frame.		R +		Q.
2 CSV_MITA(= PEAR.CSV("PECIFE.CSV")		Data		
4 electring details of these peoples and juined on or after 2014		© csv_data	8 obs. of 5 variables	
5 details - subset(csv_data,as.Date(start_date))as.Date("2014-01-01"))		O data	8 obs. of 5 variables	
a and second the		O details	4 obs. of 5 variables	
		values		
			SL.	
		ape	*10*	
			101.	
		Factorial	1	*
		Files Flore Rockages Hulp	Vaser Presentation	-0
		to a A Monet		
to Shaked i	K Serger 1			
Corente Terretest > Background John :	-5			
# 143: Colomonical Academic Mathematical Mathmatextender Mathematical Mathematical Mathematical Mat				
🖬 🔎 🛤 😨 😨 🦮 🧶 💼 💿 🙉		6	ATC Mostly cloudy	D MARS an 10 4 10 10 4

Writing into a CSV file:

csv_data<- read.csv("record.csv")
#Getting details of those peoples who joined on or after 2014
details <- subset(csv_data,as.Date(start_date)>as.Date("2014-01-01"))
Writing filtered data into a new file.
write.csv(details,"output.csv")
new_details<- read.csv("output.csv")
print(new_details)</pre>

en Classes + 2	Bostemment History Connections Totalia Connections - Other - Connections Connections - Other - Connections Data	Street days I at + 1 S Q
en 🗂	Environment Highery Connections Tabulat C and the import Desait + C 13 546 + C t + C Gatal December + Data	=== ========= ========================
Seven + R	G ^{ar} (d.) (1 ^a input Datast - 🥥 131 MB - 😿 8 - Mg Gotal December - Data	0 a + 10
	Octata 0.05, 01 Odsta 0.05, 01 Odsta15 4.05, 01 Odsta15 4.05, 01 Values 3. App 101 D 100 Farmer 100 100 Farmer 101 100	vertables
Hop I m ^O		
	Player 6	Viter State of State

9. Create pie charts and bar charts using **R**

Program Description :

A pie-chart is a representation of values as slices of a circle with different colors. The slices are labeled and the numbers corresponding to each slice is also represented in the chart.

Create data for the graph.
geeks<- c(23, 56, 20, 63)
labels <- c("Mumbai", "Pune", "Chennai", "Bangalore")</pre>

Plot the chart.
pie(geeks, labels)



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")

Rhudio		– a ×
He tot Code Wew Hots Sesson Build Debug Hotle Iools Hep 0 • 0 @ @ • □ 0 @		Resident : Norwill +
• p18 = • p18 = • p28 = • p28 = •	Environment History Connections Tutorial	60
COLO B Source on Save Q. / + D - Ban 10 0 0 D Source + 2	Eler Bote Decision links Visuar Presentation	- 6
1 # Create the data for the chart 2 A <- C(17, 32, 8, 53, 1)	(a =) / Zoon → Buot - ♥	5 Papish + 3
a # # Plot the bar chart \$ barplot[4, xlab = "X-axis", ylab = "Y-axis", main ="6ar-chart")]	Bar-Chart	
	8	
	6 -	
	, <u>8</u> -	
564 (foplexe) s R Soriet s	Y-axi	
Console Terminal × Background Jobs ×	8 -	
V# R420/// > # Create the data for the chart		
> A < C(27, 32, 8, 33, 27) > > #Plot the bar chart - benchmark while the other than the second secon	ę -	
> barplot(A, xiab = "X-axis", yiab = "Y-axis", main ="Bar-Chart") >		
	X-axis	
🗄 🔘 Type here to search 🔒 🖽 🤮 👼 🚺	x ⁶ ∧ @ ₽4	ENG 31-08-2023

10. Create a data set and do statistical analysis on the data using R

Program Description :

The R Programming Language provides some easy and quick tools that let us convert our data into visually insightful elements like graphs.

? is used before a function
to get help on that function
?plot
?chickwts
data(chickwts) #loading data into workspace
plot(chickwts\$feed) # plot feed from chickwts
feeds=table(chickwts\$feed)
plots graph in decreasing order
barplot(feeds[order(feeds, decreasing=TRUE)])



11. Write R program to find Correlation and Covariance

Program Description :

Covariance shows the direction of the path of the linear relationship between the variables while a function is applied to them.

Correlation on the contrary measures both the power and direction of the linear relationship between two variables.

- # R program to illustrate
 # pearson Correlation Testing
 # Using cor()
 # Taking two numeric
 # Vectors with same length
 x = c(1, 2, 3, 4, 5, 6, 7)
 y = c(1, 3, 6, 2, 7, 4, 5)
- # Calculating

Correlation coefficient

Using cor() method

result = cor(x, y, method = "pearson")

Print the result

cat("Pearson correlation coefficient is:", result)

0 Khudie		- a ×
Pier balt Code View Picts bestaon Build Debug Picture Tocks Help • • • • • • • • • • • • • • • • • • •		Project (Norw) •
	Invironment History Connections Tuterial	
Cold Bictomentar 9 F + 1	C I Report Dataset 1 3 121 Mill x	- 100 x 100 x
1 (a) A and (a) (b) (b) (c) (b) (c) (b) (c) (c) (c) (c) (c) (c) (c) (c) (c) (c	1 Backbornser 0 V V	-
142 Dasterit i Eloist i		
County Territori V Redescent late v		
former of the result former of the result		
■ Q Q N S 0 E	🚱 🚺 Humid ∧ Ö 🖾 📼 🔅	4(DNG 01-09-2023

Covariance

Data vectors x <- c(1, 3, 5, 10)

y <- c(2, 4, 6, 20)

Print covariance using different methods
print(cov(x, y))
print(cov(x, y, method = "pearson"))
print(cov(x, y, method = "kendall"))
print(cov(x, y, method = "spearman"))

KStudio	- 0 X
File Edit Code View Plots Session Build Debug Profile Tools Help	
🔍 + 🌒 🔐 + 🔒 🔐 🖄 🔥 Colo Stylendon 👘 🗁 Addre +	Project (Nore)
\$18 · O monk · O prár · O prár · O prár · O prár · O rát · O rát · O stár · O stár · O stár · O monost · · · · · · · · · · · · · · · · · · ·	Environment History Connections Tutorial
Contraction of the second seco	2 14 Present Dataset + 2 121 148 + 2
$ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4$	1 0.00000000000000000000000000000000000
148 Balanta Blaine	
Faceda Tandad - Referenced Inter -	
$\label{eq:constraints} \left\{ \begin{array}{llllllllllllllllllllllllllllllllllll$	
# Q O C 🐂 > O 🛢 🖗	New record A D D to d and (200 M)

12. Write R program for Regression Modeling

Program Description :

Regression analysis is a very widely used statistical tool to establish a relationship model between two variables. One of these variable is called predictor variable whose value is gathered through experiments. The other variable is called response variable whose value is derived from the predictor variable.

Generate random IQ values with mean = 30 and sd =2 IQ <- rnorm(40, 30, 2)

Sorting IQ level in ascending order IQ <- sort(IQ)</pre>

1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1) # Data Frame df<- as.data.frame(cbind(IQ, result))

Print data frame

print(df)



13 .Write R program to build classification model using KNN algorithm

Program Description :

K-Nearest Neighbor or K-NN is a Supervised Non-linear classification algorithm. K-NN is a Nonparametric algorithm i.e it doesn't make any assumption about underlying data or its distribution. It is one of the simplest and widely used algorithm which depends on it's k value(Neighbors) and finds it's applications in many industries like finance industry, healthcare industry etc.

Loading data data(iris) # Structure str(iris) # Installing Packages install.packages("e1071") install.packages("caTools") install.packages("class") # Loading package library(e1071) library(caTools) library(class) # Loading data data(iris) head(iris) # Splitting data into train # and test data split <-sample.split(iris, SplitRatio= 0.7)</pre> train_cl<-subset(iris, split == "TRUE")</pre> test_cl<-subset(iris, split == "FALSE")</pre>

Feature Scaling
train_scale<-scale(train_cl[, 1:4])
test_scale<-scale(test_cl[, 1:4])</pre>

classifier_knn

Confusiin Matrix
cm <-table(test_cl\$Species, classifier_knn)
cm</pre>

Model Evaluation - Choosing K
Calculate out of Sample error
misClassError<-mean(classifier_knn !=test_cl\$Species)
print(paste('Accuracy =', 1-misClassError))</pre>

```
# K = 15
```

classifier_knn<-knn(train =train_scale,</pre>

test =test_scale,

cl =train_cl\$Species,

k = 15)

misClassError<-mean(classifier_knn !=test_cl\$Species)

```
print(paste('Accuracy =', 1-misClassError))
```

K = 19

classifier_knn<-knn(train =train_scale,</pre>

test =test_scale, cl =train_cl\$Species,

k = 19)

misClassError<-mean(classifier_knn !=test_cl\$Species)</pre>

print(paste('Accuracy =', 1-misClassError))

Ritude		- 0 ×
File Edit Code View Plots Session Build Debug Profile Tools Help		
🔍 - 👒 💇 - 🔒 🔂 🍈 🔺 Gete fiefunction 🛛 😳 - Addins -		Project: (None)
pla = 0 pla = 0 pla = 0 pla = 0 United!**	Environment History Connections Tatorial	80
Q D B @ Source on Sale Q P + D → Run → O O → Source + 3 3 € Installion markages	Files Plots Packages Help Viewer Presentation	-0
2 install.packages("e1071")	💽 New Folder 🗧 New Bank File + 🧧 Delete 🍙 Rename - 🎱 More +	
3 fnstall.packages("caTools")	Home > 22N31A6688	
4 Install.packages("class")	A Name Size	Modified
f = samte gettage illergrigtage illergrigtage illergrigtage illergrigtage intergrigtage int	t .	
Biti (top seve) z Ksorge		
Console Terminal × Background Jobs ×		
<pre># FAELULT:</pre>		
Error in As.matris(train) : object 'train_scaled' not found > prim(saster/accuracy =', 1-misclasserror)) [1] 'Accuracy = 0.95' >		1222.044

14 Write R program to build clustering model using K-mean algorithm

Program Description :

K Means Clustering in R Programming is an Unsupervised Non-linear algorithm that cluster data based on similarity or similar groups. It seeks to partition the observations into a pre-specified number of clusters. Segmentation of data takes place to assign each training example to a segment called a cluster.

Loading data
data(iris)

Structure
str(iris)
Installing Packages
install.packages("ClusterR")
install.packages("cluster")

Loading package
library(ClusterR)
library(cluster)

Removing initial label of
Species from original dataset
iris_1 <- iris[, -5]</pre>

Fitting K-Means clustering Model
to training dataset
set.seed(240) # Setting seed
kmeans.re <- kmeans(iris_1, centers = 3, nstart = 20)
kmeans.re</pre>

```
# Cluster identification for# each observationkmeans.re$cluster
```

```
# Confusion Matrix
cm <- table(iris$Species, kmeans.re$cluster)
cm</pre>
```

```
# Model Evaluation and visualization
plot(iris_1[c("Sepal.Length", "Sepal.Width")])
plot(iris_1[c("Sepal.Length", "Sepal.Width")],
    col = kmeans.re$cluster)
plot(iris_1[c("Sepal.Length", "Sepal.Width")],
    col = kmeans.re$cluster,
    main = "K-means with 3 clusters")
```

```
## Plotiing cluster centers
kmeans.re$centers
kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")]
```

cex is font size, pch is symbol
points(kmeans.re\$centers[, c("Sepal.Length", "Sepal.Width")],

col = 1:3, pch = 8, cex = 3)

Visualizing clusters

y_kmeans<- kmeans.re\$cluster

clusplot(iris_1[, c("Sepal.Length", "Sepal.Width")],

y_kmeans,

lines = 0, shade = TRUE,

color = TRUE,

labels = 2,

plotchar = FALSE,

span = TRUE,

main = paste("Cluster iris"),

xlab = 'Sepal.Length',

ylab = 'Sepal.Width')

Output :


