1-7 -> CSV

8-13 -> EXCEL

14-17 ->others

18-23 -> vectors

24-27 -> lists

28-32 -> matrix

33-35 -> factors

36-41 -> functions

42-44 -> arrays

-------------------------------------------------------------------------------------------------------

CSV

1. Write a R program to read the content of the CSV file and show as data frame.

mydata <- read.csv("C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\spam.csv")

class(mydata)

print(mydata)

1. Write a R Program to [export CSV File without Row Names](https://www.geeksforgeeks.org/export-csv-file-without-row-names-in-r/)

my\_data <- data.frame(

Name = c("John", "Jane", "Bob"),

Age = c(25, 30, 22),

Salary = c(50000, 60000, 45000)

)

print(my\_data)

outputcsvfile <-"C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\two.csv"

write.csv(my\_data, file = outputcsvfile, row.names = FALSE)

1. [Write a R Program to write to CSV without index ?](https://www.geeksforgeeks.org/how-to-write-to-csv-in-r-without-index/)

my\_data <- data.frame(

Name = c("John", "Jane", "Bob"),

Age = c(25, 30, 22),

Salary = c(50000, 60000, 45000)

)

print(my\_data)

outputcsvfile <-"C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\three.csv"

write.csv(my\_data, file = outputcsvfile, row.names = FALSE)

cat("Data frame exported to", outputcsvfile, "\n")

1. Write a R Program to [Append row to CSV](https://www.geeksforgeeks.org/append-row-to-csv-using-r/) file.

existing\_csv\_file <- "C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\four.csv"

new\_row <- data.frame(

Name = "Alice",

Age = 28,

Salary = 55000

)

write.table(new\_row, file = existing\_csv\_file, append = TRUE, sep = ",", col.names = FALSE, row.names = FALSE)

cat("Row appended to", existing\_csv\_file, "\n")

1. [Write a R Program to calculate mean, median uand variance of a CSV file ?](https://www.geeksforgeeks.org/how-to-calculate-mean-of-a-csv-file-in-r/)

csv\_file\_path <- "C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\three.csv"

my\_data <- read.csv(csv\_file\_path)

selected\_columns <- c("Age", "Salary")

mean\_values <- sapply(my\_data[selected\_columns], mean)

median\_values <- sapply(my\_data[selected\_columns], median)

variance\_values <- sapply(my\_data[selected\_columns], var)

cat("Mean:\n")

print(mean\_values)

cat("\nMedian:\n")

print(median\_values)

cat("\nVariance:\n")

print(variance\_values)

1. **Write a R program to display the structure of the CSV file.**

csv\_file\_path <- "C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\spam.csv"

my\_data <- read.csv(csv\_file\_path)

str(my\_data)

1. Write a R program to convert the text file into CSV file.

text\_file\_path <- "C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\seven.txt"

my\_data <- read.table(text\_file\_path, header = TRUE, sep = ",")

output\_csv\_file <- "C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\seven\_output.csv"

write.csv(my\_data, file = output\_csv\_file, row.names = FALSE)

cat("Data frame exported to", output\_csv\_file, "\n")

**Excel**

1. [**Write a R Program to import an Excel File**](https://www.geeksforgeeks.org/how-to-import-an-excel-file-into-r/)

#install.packages("readxl")

library(readxl)

excelfilepath <- "C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\List of Colleges and Courses Report.xlsx"

data <- read\_excel(excelfilepath)

head(data)

1. [**Write a R Program to export a DataFrame to Excel File**](https://www.geeksforgeeks.org/how-to-export-a-dataframe-to-excel-file-in-r/)

library(writexl)

install.packages("writexl")

df <- data.frame(

Name = c("raju", "ravi", "shiva"),

Age = c(25, 30, 22),

Score = c(95, 89, 75)

)

exepath <- "C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\nine.xlsx"

write\_xlsx(df, exepath)

df=read\_excel("C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\nine.xlsx")

print(df)

1. **Write a R Program to** [**convert an Excel column into a list of vectors.**](https://www.geeksforgeeks.org/convert-an-excel-column-into-a-list-of-vectors-in-r/)

library(readxl)

excel\_data <- read\_excel("C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\nine.xlsx", sheet = "Sheet1")

column\_data <- excel\_data$Age

list\_of\_vectors <- split(column\_data, seq\_along(column\_data))

print(list\_of\_vectors)

1. [**Write a R Program to convert Excel column to vector in R ?**](https://www.geeksforgeeks.org/how-to-convert-excel-column-to-vector-in-r/)

library(readxl)

excel\_data <- read\_excel("C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\nine.xlsx", sheet = "Sheet1")

cv <- excel\_data$Age

print(cv)

1. [**Write a R Program to convert excel content into DataFrame.**](https://www.geeksforgeeks.org/how-to-convert-excel-content-into-dataframe-in-r/)

library(readxl)

ed <- read\_excel("C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\nine.xlsx", sheet = "Sheet1")

df <- as.data.frame(ed)

print(df)

1. **Write a R Program to** [**Delete rows with empty cells from Excel.**](https://www.geeksforgeeks.org/delete-rows-with-empty-cells-from-excel-using-r/)

library(tidyxl)

library(dplyr)

ed <- read\_excel("C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\nine.xlsx", sheet = "Sheet1")%>%

filter\_all(all\_vars(!is.na(.)))

print(ed)

**Others**

1. **Write a R program to find the linear regression for a dataset.**

x <- 1:10

y <- 2 \* x + rnorm(10)

linear\_model <- lm(y ~ x)

plot(x, y,

main="Scatter Plot with Linear Regression",

xlab="X",

ylab="Y")

abline(linear\_model, col="red")

1. **Write a R program to fill the null values in a data set by 3**

df <- read.csv("C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\15\_input.csv")

df[is.na(df)]=3

print(df)

1. **Write a R program to fine the no of rows and columns in a data set**

df <- read.csv("C:\\Users\\repan\\OneDrive\\Desktop\\r prgms\\15\_input.csv")

num\_rows <- nrow(df)

num\_cols <- ncol(df)

print(paste("Number of Rows:", num\_rows))

print(paste("Number of Columns:", num\_cols))

1. **Write a R program to display the decision tree for a data set.**

library(rpart.plot)

data(iris)

tree\_model <- rpart(Species ~ ., data = iris, method = "class")

prp(tree\_model)

**Vector**

1. [Find the elements of a vector that are not in another vector in R](https://www.geeksforgeeks.org/find-the-elements-of-a-vector-that-are-not-in-another-vector-in-r/)

vector1 <- c(1, 2, 3, 4, 5)

vector2 <- c(3, 4, 5, 6, 7)

elements\_not\_in\_vector2 <- vector1[!(vector1 %in% vector2)]

print(elements\_not\_in\_vector2)

1. Write a R program to take input from the user (name and age) and display the values. Also print the version of R installation.

name <- readline("Enter name:")

age <- as.integer(readline("Enter age: "))

print(paste(name))

print(paste(age))

print(paste(R.version.string))

1. Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.

seq= 20:50

meannum=mean(20:60)

sumofnum=sum(51:91)

cat(seq)

cat(meannum)

cat(sumofnum)

1. Write a R program to create a vector which contains 10 random integer values between -50 and +50.

randvec=sample(-50:50,10,replace=TRUE)

cat(randvec)

1. Write a R program to create three vectors numeric data, character data and logical data. Display the content of the vectors and their type.

vec1=c(1,2,3)

vec2=c("r","Raju")

vec3=c(TRUE,FALSE,TRUE)

cat(typeof(vec1),vec1)

cat(class(vec2),vec2)

cat(class(vec3),vec3)

1. Write a R program to find the maximum and the minimum value of a given vector.

vec1=c(1,2,3,4,10)

cat("Maximum is :",max(vec1))

cat("Minimum is :",min(vec1))

**List**

1. Write a R program to create a list of elements using vectors, matrices and a functions. Print the content of the list.

vec=c(1,2,3)

mat=matrix(1:6,nrow=2)

fun=function(x) x \* 2

list1=list(vec,mat,fun)

print(list1)

1. Write a R program to create a list of random numbers in normal distribution and count occurrences of each value.

randomnum=rnorm(30,mean = 0,sd=1)

occurance=table(randomnum)

print(occurance)

1. [How to add Key Value Pair to List in R ?](https://www.geeksforgeeks.org/how-to-add-key-value-pair-to-list-in-r/)

list1=list(name="raju",age=20)

list1$city="kuppam"

print(list1)

1. [Access Index Names of List Using lapply Function in R](https://www.geeksforgeeks.org/access-index-names-of-list-using-lapply-function-in-r/)

list1 <- list(name = "Raju", city = "kuppam", age = 20)

newlist <- lapply(names(list1), function (x) x)

print(newlist)

**Matrix**

1. Write a R program to create three vectors a,b,c with 3 integers. Combine the three vectors to become a 3×3 matrix where each column represents a vector. Print the content of the matrix.

a=c(1,2,3)

b=c(4,5,6)

c=c(7,8,9)

mat=matrix(c(a,b,c),ncol=3,byrow=FALSE)

print(mat)

1. Write a R program to create a 5 x 4 matrix , 3 x 3 matrix with labels and fill the matrix by rows and 2 × 2 matrix with labels and fill the matrix by columns.

mat54=matrix(1:20,nrow=5,ncol=4,byrow=TRUE)

rownames(mat54)=c("row1","row2","row3","row4","row5")

colnames(mat54)=c("col1","col2","col3","col4")

print(mat54)

mat33=matrix(1:9,nrow=3,ncol=3)

rownames(mat33)=c("r1","r2","r3")

colnames(mat33)=c("c1","c2","c3")

print(mat33)

mat22=matrix(1:4,nrow=2,ncol=2)

rownames(mat22)=c("r1","r2")

colnames(mat22)=c("c1","c2")

print(mat22)

1. Multiply a matrix by its transpose while ignoring missing values in R

A <- matrix(c(1,NA,3,4), nrow = 2, ncol = 2)

print(A)

print(t(A))

C <- A %\*% t(A)

print(C)

1. Write a R program to [find the power of a matrix](https://www.geeksforgeeks.org/find-the-power-of-a-matrix-in-r/)

mat =matrix(1:4, nrow = 2)

power = mat ^ 2

print(power)

1. Write a R program to [fill an empty matrix in R](https://www.geeksforgeeks.org/fill-an-empty-matrix-in-r/)

empmat <- matrix(NA, nrow = 4, ncol = 4)

print(empmat)

fill <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16)

empmat[] <- fill

print(empmat)

**Factors**

1. Write a R program to find the factors of a given number.

find\_factors <- function(num) {

factors <- c()

for (i in 1:num) {

if (num %% i == 0) {

factors <- c(factors, i)

}

}

return(factors)

}

result <- find\_factors(12)

cat("Factors of", 12, "are:", result, "\n")

1. [Write a R program to change the order of levels of a factor](https://www.geeksforgeeks.org/how-to-change-the-order-of-levels-of-a-factor-in-r/).

myfactor=factor(c("low","high","medium","low"))

ordfact=factor(myfactor,levels = c("low","high","medium"))

print(ordfact)

1. Write a R program to convert two given factors into a single factor.

factor1 <- factor(c("A", "B", "C", "A"))

factor2 <- factor(c("X", "Y", "X", "Z"))

combined\_factor <- factor(paste(factor1, factor2))

print(combined\_factor)

**Functions**

1. Write a R program to get the first 10 Fibonacci numbers

fibonacci <- numeric(10)

fibonacci[1] <- 0

fibonacci[2] <- 1

for (i in 3:10) {

fibonacci[i] <- fibonacci[i - 1] + fibonacci[i - 2]

}

print(fibonacci)

1. Write a R program to get all prime numbers up to a given number (based on the sieve of Eratosthenes).

get\_prime\_numbers <- function(n) {

primes <- rep(TRUE, n)

primes[1:2] <- FALSE

for (p in 2:sqrt(n)) {

if (primes[p]) {

for (i in seq(p^2, n, p)) {

primes[i] <- FALSE

}

}

}

return(which(primes))

}

n <- 100

prime\_numbers <- get\_prime\_numbers(n)

print(paste(prime\_numbers,collapse = " ,"))

1. Write a R program to print the numbers from 1 to 100 and print "Fizz" for multiples of 3, print "Buzz" for multiples of 5, and print "FizzBuzz" for multiples of both.

for (i in 1:100) {

if (i %% 3 == 0 && i %% 5 == 0) {

print("FizzBuzz")

} else if (i %% 3 == 0) {

print("Fizz")

} else if (i %% 5 == 0) {

print("Buzz")

} else {

print(i)

}

}

1. Write a R program to find the number of occurrence of a character in an string.

text <- "This is a sample text with some sample words."

tar= "s"

count = sum(charToRaw(text) == charToRaw(tar))

print(count)

1. Write a R program to extract first 10 english letter in lower case and last 10 letters in upper case and extract letters between 22nd to 24th letters in upper case.

letters\_lower <- letters[1:10]

letters\_upper <- LETTERS[17:26]

letters\_between <- toupper(letters[22:24])

print(letters\_lower)

print(letters\_upper)

print(letters\_between)

1. Write a R program to get the unique elements of a given string and unique numbers of vector.

string <- "abracadabra"

strsp=unique(unlist(strsplit(string,"")))

vector <- c(1, 2, 2, 3, 4, 4, 5)

unique\_numbers <- unique(vector)

print(unique\_numbers)

print(strsp)

**Array**

1. Write a R program to create an array, passing in a vector of values and a vector of dimensions. Also provide names for each dimension

values=c(1,2,3,4,5,6)

dims=c(2,3)

dimnames=list(c("r1","r2"),c("c1","c2","c3"))

ary=array(values,dims,dimnames = dimnames)

print(ary)

1. Write a R program to create an array with three columns, three rows, and two "tables", taking two  vectors as input to the array.  Print the array.

data1 <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)

data2 <- c(10, 11, 12, 13, 14, 15, 16, 17, 18)

array\_dims <- c(3, 3, 2)

my\_array <- array(c(data1, data2), dim = array\_dims)

print(my\_array)

1. Write a R program to create an 3 dimensional array of 24 elements using the dim() function.

my\_array <- array(1:24, dim = c(3, 4, 2))

print(my\_array)