

```
#Nihal Sayyad (CSE AIML 21)
```

```
#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 your name :")  
age<-readline("Enter your age :")  
cat(name,"is",age,"years old.")
```

```
#to print r version  
print(R.version.string)
```

```
#2. Write a R program to get the details of the objects in memory.
```

```
name = "Nihal"  
n1 = 21  
n2 = 2.1  
nums = c(7, 13, 21, 28, 37, 48, 57)  
print(ls())  
print("Details of the objects in memory :")  
print(ls.str())
```

```
#3 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
```

```
sequence<-c(20:50)  
print(sequence)  
avg<-mean(c(20:60))  
print(avg)  
addition<-sum(c(51:91))  
print(addition)
```

```
#4 Write a R program to create a vector which contains 10 random integer  
values
```

```
#between -50 and +50.
```

```
set.seed(100)  
vector<-sample(c(-50:50),10)  
print(vector)
```

```
#or
```

```
v = sample(-50:50, 10, replace=TRUE)  
print("Content of the vector:")  
print("10 random integer values between -50 and +50:")  
print(v)
```

```
#5 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
```

```

vector2<-c(1:100)
for(i in vector2){
  if(i%%3==0 & i%%5==0){
    vector2[i]="FizzBuzz"
  }
  else if(i%%3==0){
    vector2[i]="Fizz"
  }
  else if(i%%5==0){
    vector2[i]="Buzz"
  }
  else{
    vector2[i]=i
  }
}
print(vector2)

```

#6 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.

```

l<-letters[1:10]
print(l)
L<-LETTERS[17:26]
print(L)
print(LETTERS[22:24])

```

#or

```

t = head(letters, 10)
print(t)
t = tail(LETTERS, 10)
print(t)
e = tail(LETTERS[22:24])
print(e)

```

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

```

a<-c(10,20,30)
b<-c(40,50,60)
c<-c(70,80,90)
matrix1<-matrix(c(a,b,c),nrow = 3,ncol = 3)
matrix1

```

#8 Write a R program to read the .csv file and display the content

```
setwd("F:/4 SEM/R/POE")
data1<-read.csv("data.csv")
data1
```

#9 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 x 2 matrix with labels and fill  
#the matrix by columns.

```
#5 x 4 matrix
matrix2<-matrix(c(1:20),nrow = 5,ncol = 4)
matrix2
```

```
#3 x 3 matrix with labels
matrix3<-matrix(c(1:9),nrow = 3,ncol=3,byrow = TRUE)
rownames(matrix3)<-c("A", "B", "C")
colnames(matrix3)<-c("D", "E", "F")
matrix3
```

```
#2 x 2 matrix with labels
matrix4<-matrix(c(1:4),nrow = 2,ncol = 2)
rownames(matrix4)<-c("A", "B")
colnames(matrix4)<-c("C", "D")
matrix4
```

#or

```
m1 = matrix(1:20, nrow=5, ncol=4)
print("5 x 4 matrix:")
print(m1)
cells = c(1,3,5,7,8,9,11,12,14)
rnames = c("Row1", "Row2", "Row3")
cnames = c("Col1", "Col2", "Col3")
m2 = matrix(cells, nrow=3, ncol=3, byrow=TRUE, dimnames=list(rnames, cnames))
print("3 x 3 matrix with labels, filled by rows: ")
print(m2)
print("3 x 3 matrix with labels, filled by columns: ")
m3 = matrix(cells, nrow=3, ncol=3, byrow=FALSE, dimnames=list(rnames, cnames))
print(m3)
```

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

```
list1<-list(
  c("NIHAL", "ABHISHEK", "VIVEK", "ROHAN"),
  c(5,7,4,8),
  matrix(c(8,8,1,6),nrow = 2),
  atan(45)
)
list1
```

#11 Write a R program to create bell curve of a random normal distribution.

```
K<-floor(rnorm(20000,500,100))
t<-table(K)
barplot(t)
```

#12 Write a R program to create a Data frames which contain details of 5 employees and display the details

```
employees<-data.frame(Names=c("Nihal","Abhishek","Vivek","Rohan","Shreyash"),
                      Age=c(31,30,30,31,36),
                      Gender=c("M","M","M","M","M"),
                      Designation=c("Knowledge Engineer","Network Architect",
                                    "Junior Developer","Senior Developer",
                                    "Security Expert"))

print("Employee Details are :")
print(employees)
```

#13 Write a R program to create the system's idea of the current date with and without time

```
print("Today's Date is :")
print(Sys.Date())
```

```
print("Date with time is :")
print(Sys.time())
```

#14 Write a R program to rotate a given matrix 90 degree clockwise rotation.

```
matrix5<-matrix(c(1:16),nrow = 4,ncol = 4)
matrix5
t(apply(matrix5,2,rev))
```

#15 Write a R program to find row and column index of maximum and minimum value in a given matrix

```
set.seed(100)
matrix6<-matrix(sample(c(1:100),16),nrow = 4,ncol = 4)
matrix6
which(matrix6==max(matrix6),arr.ind = TRUE)
which(matrix6==min(matrix6),arr.ind = TRUE)
```

or

```
m = matrix(c(1:16), nrow = 4, byrow = TRUE)
print("Original Matrix:")
print(m)
result = which(m == max(m), arr.ind=TRUE)
print("Row and column of maximum value of the said matrix:")
print(result)
result = which(m == min(m), arr.ind=TRUE)
print("Row and column of minimum value of the said matrix:")
print(result)
```

#16 Write a R program to convert a matrix to a 1 dimensional array.

```
set.seed(50)
matrix7<-matrix(sample(c(1:100),16),nrow = 4,ncol = 4)
matrix7
array1<-array(matrix7)
array1
#or
m<-as.vector(matrix7)
print(m)
```

#17 Write a R program to access the element at 3rd column and 2nd row, only the

# 3rd row and only the 4th column of a given matrix.

```
set.seed(150)
matrix8<-matrix(sample(c(1:100),16),nrow = 4,ncol = 4)
matrix8
matrix8[2,3]
matrix8[3,]
matrix8[,4]
```

#or

```
row_names = c("row1", "row2", "row3", "row4")
col_names = c("col1", "col2", "col3", "col4")
M = matrix(c(1:16), nrow = 4, byrow = TRUE, dimnames = list(row_names,
                                                             col_names))

print("Original Matrix:")
print(M)
print("Access the element at 3rd column and 2nd row:")
print(M[2,3])
print("Access only the 3rd row:")
print(M[3,])
print("Access only the 4th column:")
print(M[,4])
```

#18 Write a R program to create an array using four given columns, three given #rows, and two given tables and display the content of the array.

```
set.seed(100)
array2<-array(sample(c(1:100),24),dim = c(3,4,2))
array2
```

#19 Write a R program to combine three arrays so that the first row of the #first array is followed by the first row of the second array and then first #row of the third array.

```
vec1<-rbind(rep("S",3),rep("T",3),rep("K",3))
vec1
vec2<-rbind(rep("I",3),rep("L",3),rep("U",3))
vec2
vec3<-rbind(rep("X",3),rep("Y",3),rep("Z",3))
vec3
```

```

new<-matrix(t(cbind(vec1,vec2,vec3)),ncol = 3,byrow = TRUE)
new

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

vec<-sample(1:5,24,replace = TRUE)
dim(vec)<-c(3,2,4)
print(vec)

#21 Create a small data frame representing a database of films.
#It should contain the fields title, director, year, country,
#and at least three films.Create a second data frame of the same format as
above
#,but containing just one new film.Merge the two data frames and sort()

filmdatal<-data.frame(FILM=c("Dr. Strange","TENET","Avengers:Infinity War"),
                      DIRECTOR=c("Sam Raimi","Christopher Nolan",
                                   "Russo Brothers"),
                      YEAR=c(2022,2020,2018),
                      COUNTRY=c("USA","UK","USA"))

filmdatal
filmdata2<-data.frame(FILM="Thor",
                      DIRECTOR="Taika Waititi",
                      YEAR=2022,
                      COUNTRY="USA")

filmdata2
newdata<-rbind(filmdatal,filmdata2)
newdata
#sort function doesn't work with the data frames
newdata[order(newdata$YEAR),]

#22 Write an R script to find subset of data set by using subset(),
aggregate()
#functions on mtcars data set

#Subsetting and Aggregating

cardata<-read.csv("mtcars.csv")
cardata

#subsetting using Brackets by Selecting Rows and Columns
cardata[c(1:3),]
cardata[c(1:3),c(3:5)]

#Subset Using Brackets by Excluding Rows and Columns
cardata[-c(4:nrow(cardata)),]
cardata[-c(4:nrow(cardata)),-c(5:ncol(cardata)) ]

#Subset Using Brackets with which() Function
cardata[which(cardata$hp>200),]
cardata[which(cardata$hp>200),1:5]

```

```

#Subset Data with subset() Function
subset(cardata, gear==5&hp>200)
subset(cardata, gear==5&hp>200,1:5)

#Subset Data in Combination of select() and filter() Functions
library(dplyr)
filter(cardata, gear==5 & hp>200)
select(filter(cardata, gear==5 & hp>200),1:5)

#Subset a Random Sample with sample() Function
set.seed(123) # For reproducibility of same result
cardata[sample(1:nrow(cardata), 3, replace = FALSE),]
cardata[sample(1:nrow(cardata), 3, replace = FALSE),1:5]

#Aggregating
agg_mean = aggregate(cardata[,5],by=list(cardata$gear),FUN=mean, na.rm=TRUE)
agg_mean

#Example for aggregate() function in R with sum:
agg_sum = aggregate(cardata[,5],by=list(cardata$gear),FUN=sum, na.rm=TRUE)
agg_sum

#Example for aggregate() function in R with count:
agg_count = aggregate(cardata[,5],by=list(cardata$gear),FUN=length)
agg_count

#Example for aggregate() function in R with maximum:
agg_max = aggregate(cardata[,5],by=list(cardata$gear),FUN=max, na.rm=TRUE)
agg_max

#Example for aggregate() function in R with minimum:
agg_max = aggregate(cardata[,5],by=list(cardata$gear),FUN=min, na.rm=TRUE)
agg_max

#23 Create dataframe empdetails (empid, name, salary, dept).
#Fill the data. Use dept as factor. Create another dataframe and merge both
#of them
set.seed(100)
empdetails<-data.frame(empid=sample(c(100:300),5),
                        name=c("Abhishek","Vivek","Rohan","Shreyash","Anirudha"),
                        salary=c(120,110,80,95,100),
                        dept=c("Developer","Developer","Developer","Expert",
                              "Expert"))

empdetails
factor(empdetails$dept)
empdetails1<-data.frame(empid=sample(c(100:300),1),
                        name="Sushant",
                        salary=130,
                        dept="HR")

empdetails1
empdetails2<-rbind(empdetails,empdetails1)

```

```
empdetails2
factor(empdetails2$dept)
```

```
#24 Load mtcars data. Perform summary, display column names ,
#analyze data using plot,scatter plot. Change size and shape
```

```
cardata<-read.csv("mtcars.csv")
cardata
summary(cardata)
print(colnames(cardata))
```

```
#wt vs mpg
wtMpg <- cardata[,c('wt','mpg')]
# Plot the chart for cars with weight between 2.5 to 5 and mileage between
# 15 and 30.
```

```
plot(x = wtMpg$wt,y = wtMpg$mpg,
      xlab = "Weight",
      ylab = "Milage",
      xlim = c(0,6),
      ylim = c(0,50),
      main = "Weight vs Milage",
      pch=16,
      col=wtMpg$mpg
```

```
)
```

```
#horsepower vs mieage
```

```
hpMpg <- cardata[,c('hp','mpg')]
plot(x = hpMpg$hp,y = hpMpg$mpg,
      xlab = "Horse Power",
      ylab = "Milage",
      xlim = c(50,350),
      ylim = c(10,35),
      main = "Horsepower vs Milage",
      pch=16,
      col=hpMpg$hp
```

```
)
```

```
#reshape data
```

```
transpose<-t(cardata)
transpose
```

```
cardata1<-cardata[1:10,]
cardata1
```

```
#25 Create a list .Perform following operation on list ,
#access first and last element in list, update list , add two list ,subtract
#two list
```

```
newlist1<-list(Name=c("Abhishek","Vivek"),Age=c(21,20),Score=c(120000,110000))
newlist1
```

```
#first element of list
```



```
newlist1[[1]]

#first element of first element in list
newlist1[[1]][[1]]
#or
newlist1[[c(1,1)]]

#last element of list
newlist1[[length(tail)]]

#another list
newlist2<-list(Grade=c("A","B+"),Attendance=c(90,84))
newlist2

#addition of two lists
newlist3<-c(newlist1,newlist2)
newlist3

#subtract two lists
setdiff(newlist3,newlist1)
```