Session 4 Assignment:: (Basic Operation in R and Data Manipulation)

Problem Statement:

1. Create the vectors

(a) (2, 3, … , 29, 30)

Ans :c(2:30)

(b) (30, 29, … , 2)

Ans:c(30:2)

OR

sort(c(30:2),decreasing = T)

(c) (1, 2, 3, …. , 29, 30, 29, 28, , 2, 1)

Ans:

a<-seq(c(1:30))

a

b<-sort(c(29:1),decreasing = T)

b

d<-c(a,b)

d

(d) (4, 6, 3) and assign it to the name dev.

For parts (e), (f) and (g) .

Ans:

dev<-c(4,5,6)

dev

(e) (5, 6, 7, 5, 6, 7, , 5, 6, 7) where there are 10 occurrences of 5.

Ans:::

n1<-c(5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7)

n1

# Or- An efficient way of writing a code for the above vector as follows:

# To replicate the values in a given vector a specified number of times in a collated fashion then:-

n1<-rep(5:7,times=10)

n1

(f) (5, 6, 7, 5, 6, 7, , 5, 6, 7, 5) where there are 11 occurrences of 5, 10 occurrences of 6 and 10 occurrences of 7.

# To find the occurences/frequencies of various values in a given vector,

# use the function table to find them out as follows:

n2<-c(5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5)

occ2<-table(n2)

occ2

# To explicitly find the occurence/frequencies of specified values within a given

# vector,you need # to subset that vector

n2<-c(5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5)

occ2[names(occ2)==5]

occ2[names(occ2)==6]

occ2[names(occ2)==7]

# Or alternatively convert the given Vector into a data.frame as follows:

n2<-c(5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5,6,7,5)

as.data.frame(table(n2))

(g) (4, 4, , 4, 6, 6, , 6, 3, 3, , 3) where there are 10 occurrences of 4, 20 occurrences of 6 and 30 occurrences of 3.

n3<-c(4,4,4,4,4,4,4,4,4,4,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3)

n3

# OR -An efficient way of writing the code alternatively:-

# To replicate the values in the given vector in an uncollated fashion then:-

r1<-rep(4, each = 10)

r2<-rep(6, each = 20)

r3<-rep(3, each = 30)

n3<-c(r1,r2,r3)

n3

# To find the occurences/frequencies of various values in a given vector,

# use the function table to find them out as follows:

occ3<-table(n3)

occ3n3<-c(4,4,4,4,4,4,4,4,4,4,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3)

n3

2. Create a vector of the values of eX sin(x) at x = 3, 3.1, 3.2, , 6.

Ans ::

# The sine function sinx is one of the basic functions encountered in trigonometry.

# sine theta is the vertical coordinate of the arc endpoint

x <- seq(3, 6, by=0.1)

x

y <- exp(x)\*sin(x)

y

3. Execute the following lines which create two vectors of random integers which are chosen with

replacement from the integers 0, 1, : : : , 999. Both vectors have length 250.

set.seed(100)

x <- Sample (0:999, 250, replace=T)

y <- Sample (0:999, 250, replace=T)

1. Identify out the values in y which are > 500.

Ans:

y\_val <- y[c(y>500)==TRUE]

y\_val

1. Identify the index positions in y of the values which are > 700?

Ans :

in\_pos <- which(y>700)

in\_pos

1. What are the values in x which are in Same index position to the values in y which are > 400?

Ans:

eh<-x[y>400]

eh\_3

1. How many values in y are within 200 of the maximum value of the terms in y?

Ans:

sum(y>max(y)-200)

1. How many numbers in x are divisible by 2?

Ans:

divi <-length(x[x%%2])

divi

1. Sort the numbers in the vector x in the order of increasing values in y.

Ans :

egsort <- x[order(y)]

egsort

1. Create the vector (x1 + 2x2 - x3; x2 + 2x3 -x4 ,, xn−2 + 2xn−1 - xn).

Ans :

Vec\_Cre <-x[-c(249,250)] + 2\*x[-c(1,250)]-x[-c(1,2)]

vec\_Cre

1. Calculate:

4**.** Use the function paste to create the following character vectors of length 30:

(a) ("Label 1", "Label 2", ....., "Label 30"). Business Analytics with R

**Page 4**

\*Note that there is a single space between label and the number following.

Ans : 4.a)

# The argument ... means that it takes any number of objects. The argument sep is a

# character string that is used as a separator. The argument collapse is an optional string to

# indicate if we want all the terms to be collapsed into a single string.

char\_vec\_demo\_1 <- paste("Label", c(1:30), sep =" ", collapse = NULL)

char\_vec\_demo\_1 <- paste("Label", c(1:30), sep =" ", collapse = NULL)

char\_vec\_demo\_1

(b) ("FN1", "FN2", ..., "FN30").

\*\*In this case, there is no space between fn and the number following.

Ans 4.b)

char\_demo<-paste("FN", 1:30, sep = "", collapse = NULL)

char\_demo

5**.** Compound interest can be computed using the formula

A = P × (1 + R/100)n, where P is the original money lent, A is what it amounts to in n years at R percent per year interest.

Write R code to calculate the amount of money owed after n years, where n changes from 1 to 15 in yearly increments, if the money lent originally is 10000 Rupees and the interest rate remains constant throughout the period at 11.5%.

Ans 5 :

# The formula to find out Compund interest is: A = P (1 + R/100)

principal\_amt<-10000

int\_rate<-11.5

no\_of\_years<-15

Amount<-?

# By applying the above values in the given Compound Interest formula then:

for (int\_rate in no\_of\_years){

Amount <- principal\_amt \* (1+int\_rate/100)^(1:no\_of\_years)

print(Amount)

}

Amount

6) Generate the following matrices.

[,1] [,2] [,3] [,4]

[1,] 1 101 201 301

[2,] 2 102 202 302

[3,] 3 103 203 303

[4,] 4 104 204 304

[5,] 5 105 205 305

Ans 6.

#Matrix is a two dimensional data structure in R programming.

#Matrix is similar to vectors but additionally contains the dimension attribute.

Matrix\_d<-matrix(c(1:5,101:105,201:205,301:305),nrow=5,ncol=4)

matrix\_d

# To check whether the above variable is a matrix or not, we use Class function as follows:

class(matrix\_d)

# To understand the attributes/properties of the above matrix, we use attributes or dim function as follows:

attributes(matrix\_d)