2022MCS120009_Assigment_01

Santhosh Kumar N

09/03/2022

1. Calculate the square root of 12345 and perform a log2 transformation on the result

Ans:

```
sqrt(12345)
```

```
## [1] 111.1081
```

```
log2(sqrt(12345))
```

```
## [1] 6.79582
```

2.Create the vector (20,21,22,23,. . . 37,38,39,40,39,38,37,. . . ,23,22,21,20) in R.

Ans:

```
v <- c(20:37,36:20)

print(v)
```

```
##  [1] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 36 35 34 33 32 31 30
## [26] 29 28 27 26 25 24 23 22 21 20
```

3.Create the vector (5, 9, 2, 5, 9, 2, . . . , 5, 9,2, 5) where there are 13 occurrences of 5, 12 occurrences of 9 and 12 occurrences of 2 in R.

Ans:

```
v <- rep(c(5,9,2),times=c(13,12,12))

print(v)
```

```
##  [1] 5 5 5 5 5 5 5 5 5 5 5 5 5 9 9 9 9 9 9 9 9 9 9 9 2 2 2 2 2 2 2 2 2 2 2 2 2
```

4. TopCoder is one of the original platforms for competitive programming online,Write the R code for the numbers 33 to 99, print "Top" if the number is a multiple of 3, "Coder" if the number is a multiple of 9, "TopCoder" if the number is a multiple of both 3 and 9, and simply print the number otherwise

Ans:

```
nums = c(33:99)

for(x in nums){
  if (x%%3 == 0 & x%%9 == 0){
    cat(x,":",'TopCoder',"\n")
    next
  }

  else if(x%%3 == 0){
    cat(x,":",'Top',"\n")
    next
  }

  else if(x%%9 == 0){
    cat(x,":",'Coder',"\n")
    next
  }

  else{
    cat(x,"\n")
  }
}
```

```
## 33 : Top
## 34
## 35
## 36 : TopCoder
## 37
## 38
## 39 : Top
## 40
## 41
## 42 : Top
## 43
## 44
## 45 : TopCoder
## 46
## 47
## 48 : Top
## 49
## 50
## 51 : Top
## 52
## 53
## 54 : TopCoder
## 55
## 56
## 57 : Top
## 58
## 59
## 60 : Top
## 61
## 62
## 63 : TopCoder
```

```
## 64
## 65
## 66 : Top
## 67
## 68
## 69 : Top
## 70
## 71
## 72 : TopCoder
## 73
## 74
## 75 : Top
## 76
## 77
## 78 : Top
## 79
## 80
## 81 : TopCoder
## 82
## 83
## 84 : Top
## 85
## 86
## 87 : Top
## 88
## 89
## 90 : TopCoder
## 91
## 92
## 93 : Top
## 94
## 95
## 96 : Top
## 97
## 98
## 99 : TopCoder
```

5. Create a R script that will print 'This is a Matrix' if the variable matrix_sample is a matrix, otherwise print "This is not a Matrix". Hint: check out help(is.matrix)

Ans:

```
checkTypeOfMatrix <- function(matrix_sample){
  return (is.matrix(matrix_sample))
}

A <- matrix(c(1:5))

ifelse(checkTypeOfMatrix(A) == TRUE,print("This is a Matrix"),print("This is not a Matrix"))
```

```
## [1] "This is a Matrix"
```

```
## [1] "This is a Matrix"
```

```
B <-c(1,2,3,4,5,6)

ifelse(checkTypeOfMatrix(B) == TRUE,print("This is a Matrix"),print("This is not a Matrix"))
```

```
## [1] "This is not a Matrix"
```

```
## [1] "This is not a Matrix"
```

6. We have a collection of balls in a ball pit.Now our ball pit is filled with balls of different colours such as blue,red,yellow,green,violet,black and white

a Enter the list of colours into a vector called ball_colour
Ans:

```
ball_colour = c('blue','red','yellow','green','violet','black','white')
```

b.Display the fourth element in the vector and Enter some numerical weight data into a vector called ball_weight. *Ans:*

```
print(ball_colour[4])
```

```
## [1] "green"
```

```
ball_weight = c(55,42,83,24,75,96,17)
```

c. Join the two vectors into a data frame called ball_desc containing 2 columns and 4 rows.Describe the first column as colour and the second one as weight
Ans:

```
ball_desc = data.frame(ball_colour[1:4], ball_weight[1:4])

names(ball_desc) <- c('colour','weight')

ball_desc
```

```
##   colour weight
## 1   blue     55
## 2    red     42
## 3 yellow     83
## 4  green     24
```

7. Consider the two vectors xvec and yvec given below:

```
set.seed(99)
sampvec1 <- sample(0:100, 25)
sampvec2 <- sample(0:99, 25)
sampvec1
```

```
##  [1] 47 32 43 21 61 31 12 19 30 67  8 81 87 29 85 83 95 13  3 77  6 70 37 84 57
```

```
sampvec2
```

```
##  [1] 53 45 43 57 64 13 59 30 99 62 37 78 31 84 70 16 28 44 11 72 51 89 58 39 34
```

a.Determines the index of the minimum of the sampvec1 and maximum of sampvec2 vector

Ans:

```
index = 1;

for(x in sampvec1){
  if(x == min(sampvec1)){
    print(x)
    print(index)
    break
  }
}
```

```
## [1] 3
## [1] 1
```

b. Find out the values in sampvec1 which are greater than 44

Ans:

```
for(x in sampvec1){
  if(x > 44){
    print(x)
  }}
```

```
## [1] 47
## [1] 61
## [1] 67
## [1] 81
## [1] 87
## [1] 85
## [1] 83
## [1] 95
## [1] 77
## [1] 70
## [1] 84
## [1] 57
```

c. How many numbers in sampvec2 are divisible by 7?

Ans:

```
counter = 0

for(x in sampvec2){
  if(x %% 7 == 0){
    counter = counter + 1
  }
}

print(counter)
```

```
## [1] 3
```

d. Sort the numbers in the vector sampvec1 in the order of decreasing values in the sampvec2

Ans:

```
df = data.frame(c(sampvec1),c(sampvec2))
print(df[order(df$c.sampvec2., decreasing = TRUE), ])
```

```
##    c.sampvec1. c.sampvec2.
## 9           30          99
## 22          70          89
## 14          29          84
## 12          81          78
## 20          77          72
## 15          85          70
## 5           61          64
## 10          67          62
## 7           12          59
## 23          37          58
## 4           21          57
## 1           47          53
## 21           6          51
## 2           32          45
## 18          13          44
## 3           43          43
## 24          84          39
## 11           8          37
## 25          57          34
## 13          87          31
## 8           19          30
## 17          95          28
## 16          83          16
## 6           31          13
## 19           3          11
```

8. Assume that you are interested in Rectangular Prisms, and have measured the height ,weight and breadth of 4 rectangular prisms.Using these value we make three vectors Length,Width and Height as follows: Length <- c(8.2,16,15,9) Width <- c(5,7,10,6) Height <- c(15.8,3,5,4)

a. The volume of a rectangular prism is length x width x height. Make a vector with the volumes of the 4 rectangular prism

Ans:

```
Length <- c(8.2,16,15,9)
Width <- c( 5,7,10,6)
Height <- c(15.8,3,5,4)

v = as.vector(Length * Width * Height)

print(v)
```

```
## [1] 647.8 336.0 750.0 216.0
```

b. Compute the mean, median and standard deviation of the rectangular prism volumes

Ans:

```
cat("mean:",mean(v),"\n")
```

```
## mean: 487.45
```

```
cat("median:",median(v),"\n")
```

```
## median: 491.9
```

```
cat("standard deviation:",sd(v))
```

```
## standard deviation: 252.4987
```

c. Compute the mean of volume for the rectangular prism if Length less than 10 ,elase print "Length greater than 10"

Ans:

```
count=1
for(x in Length){
  if(x<10){
    print(mean(v[count]))
  }else{
    print("Length greater than 10");
  }
  count = count+1;
}
```

```
## [1] 647.8
## [1] "Length greater than 10"
## [1] "Length greater than 10"
## [1] 216
```

9. Write a function which takes a single argument which is a matrix.The function should eturn a matrix which is the same as the function argument but every odd number is tripled nd even number doubled on the given matrix

Ans:

```
my_function <- function(matrix){

  # get the row number of matrix
  row = nrow(matrix)

  # get the column number of matrix
  cols = ncol(matrix)

  # create empty list
  rv1 <- c()

  for(x in c(matrix)){
    if((x%%2)==0){
      rv1 <- c(rv1,x*x)
    }else{
      rv1 <- c(rv1,x*x*x)
    }
```

```
  }

  return (matrix(c(rv1),nrow = row,ncol = cols))
}

A = matrix(
  # Taking sequence of elements
  c(-5,0,5,10,-4,1,6,11,-3,2,7,12,-2,3,8,13,-1,4,9,14),
  # No of rows
  nrow = 5,
  # No of columns
  ncol = 4,
  byrow = TRUE
)


print("Given Matrix")
```

```
## [1] "Given Matrix"
```

```
print(A)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   -5    0    5   10
## [2,]   -4    1    6   11
## [3,]   -3    2    7   12
## [4,]   -2    3    8   13
## [5,]   -1    4    9   14
```

```
print("Output")
```

```
## [1] "Output"
```

```
print(my_function(A))
```

```
##      [,1] [,2] [,3] [,4]
## [1,] -125    0  125  100
## [2,]   16    1   36 1331
## [3,]  -27    4  343  144
## [4,]    4   27   64 2197
## [5,]   -1   16  729  196
```

10. For this exercise we'll use the built-in dataset state.x77. df <- as.data.frame(state.x77) head(df)

Ans:

```
df <- as.data.frame(state.x77)

head(df)
```

```
##            Population Income Illiteracy Life Exp Murder HS Grad Frost   Area
## Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
## Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
## Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
```

```
## Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
## California      21198   5114        1.1    71.71   10.3    62.6    20 156361
## Colorado         2541   4884        0.7    72.06    6.8    63.9   166 103766
```

a. Find out how many states have an income of less than 5000 and Find out which is the state with the lowest income

Ans:

```
dim(df[df$Income < 5000,])[1]
```

```
## [1] 42
```

```
colnames(t(df[df$Income == min(df$Income),]))
```

```
## [1] "Mississippi"
```

b. Create a data frame with the datasets state.area, state.division, state.name, state.region and add the data frame column-wise to state.x77

Ans:

```
df1 = data.frame(state.x77)

df2 = data.frame(state.area, state.division, state.name, state.name)

df1 <- cbind(df1,state.area,df1,state.division,df1,state.name,df1,state.name)

head(df1,5)
```

```
##            Population Income Illiteracy Life.Exp Murder HS.Grad Frost   Area
## Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
## Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
## Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
## Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
## California      21198   5114        1.1    71.71   10.3    62.6    20 156361
##            state.area Population Income Illiteracy Life.Exp Murder HS.Grad
## Alabama         51609       3615   3624        2.1    69.05   15.1    41.3
## Alaska         589757        365   6315        1.5    69.31   11.3    66.7
## Arizona        113909       2212   4530        1.8    70.55    7.8    58.1
## Arkansas        53104       2110   3378        1.9    70.66   10.1    39.9
## California     158693      21198   5114        1.1    71.71   10.3    62.6
##            Frost   Area     state.division Population Income Illiteracy
## Alabama       20  50708 East South Central       3615   3624        2.1
## Alaska       152 566432            Pacific        365   6315        1.5
## Arizona       15 113417           Mountain       2212   4530        1.8
## Arkansas      65  51945 West South Central       2110   3378        1.9
## California    20 156361            Pacific      21198   5114        1.1
##            Life.Exp Murder HS.Grad Frost   Area state.name Population Income
## Alabama       69.05   15.1    41.3    20  50708    Alabama       3615   3624
## Alaska        69.31   11.3    66.7   152 566432     Alaska        365   6315
## Arizona       70.55    7.8    58.1    15 113417    Arizona       2212   4530
## Arkansas      70.66   10.1    39.9    65  51945   Arkansas       2110   3378
## California    71.71   10.3    62.6    20 156361 California      21198   5114
##            Illiteracy Life.Exp Murder HS.Grad Frost   Area state.name
## Alabama           2.1    69.05   15.1    41.3    20  50708    Alabama
## Alaska            1.5    69.31   11.3    66.7   152 566432     Alaska
```

```
## Arizona           1.8    70.55    7.8    58.1    15 113417    Arizona
## Arkansas          1.9    70.66   10.1    39.9    65  51945   Arkansas
## California        1.1    71.71   10.3    62.6    20 156361 California
```

c. Add a variable to the data frame which should categorize the level of illiteracy: 0-1 : low, 1-2: average, 2-10: high

Ans:

```
df <- as.data.frame(state.x77)

df$categ_illiteracy <- "NA"
df$categ_illiteracy[df$Illiteracy <1] <- 'low'
df$categ_illiteracy[df$Illiteracy >1 & df$Illiteracy <2] <- 'average'
df$categ_illiteracy[df$Illiteracy >2 & df$Illiteracy <10] <- 'high'

head(df,10)
```

```
##             Population Income Illiteracy Life Exp Murder HS Grad Frost   Area
## Alabama           3615   3624        2.1    69.05   15.1    41.3    20  50708
## Alaska             365   6315        1.5    69.31   11.3    66.7   152 566432
## Arizona           2212   4530        1.8    70.55    7.8    58.1    15 113417
## Arkansas          2110   3378        1.9    70.66   10.1    39.9    65  51945
## California       21198   5114        1.1    71.71   10.3    62.6    20 156361
## Colorado          2541   4884        0.7    72.06    6.8    63.9   166 103766
## Connecticut       3100   5348        1.1    72.48    3.1    56.0   139   4862
## Delaware           579   4809        0.9    70.06    6.2    54.6   103   1982
## Florida           8277   4815        1.3    70.66   10.7    52.6    11  54090
## Georgia           4931   4091        2.0    68.54   13.9    40.6    60  58073
##             categ_illiteracy
## Alabama                 high
## Alaska               average
## Arizona              average
## Arkansas             average
## California           average
## Colorado                 low
## Connecticut          average
## Delaware                 low
## Florida              average
## Georgia                   NA
```

d. Find out which state has area greater than 21,000, with low literacy rate and income

Ans:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df <- as.data.frame(state.x77)

df1 = df %>% filter(df$Area > 21000 & df$Illiteracy == min(df$Illiteracy))
df2 = df1 %>% filter(df1$Income == min(df1$Income))
print(colnames(t(df2)))
```

```
## [1] "South Dakota"
```

11. We have a vector vec containing the number 23,78,42,63,90,15. Create a for loop that, given a numeric vector, prints out one number per line, with its square and cube alongside

Ans:

```
v = c(23,78,42,63,90,15)

for(x in v){
  cat(x,x*x,x*x*x,"\n")
}
```

```
## 23 529 12167
## 78 6084 474552
## 42 1764 74088
## 63 3969 250047
## 90 8100 729000
## 15 225 3375
```

a. Show how to use a while loop to achieve the same result

Ans:

```
v = c(23,78,42,63,90,15)

i <- 1

while (i<(length(v) + 1)){
  cat(v[i],v[i]*v[i],v[i]*v[i]*v[i],"\n")

  i = i +1
}
```

```
## 23 529 12167
## 78 6084 474552
## 42 1764 74088
## 63 3969 250047
## 90 8100 729000
## 15 225 3375
```

b. Show how to achieve the same result without the use of an explicit loop

Ans:

```
v = c(23,78,42,63,90,15)

myfuction<-function(x){
  c(x,x*2,x^3)
```

```
}

movies_lower <-lapply(v, myfuction)
str(movies_lower)
```

```
## List of 6
##  $ : num [1:3] 23 46 12167
##  $ : num [1:3] 78 156 474552
##  $ : num [1:3] 42 84 74088
##  $ : num [1:3] 63 126 250047
##  $ : num [1:3] 90 180 729000
##  $ : num [1:3] 15 30 3375
```

12. Calculate the following by writing code snippets:

- $$\sum_{i=1}^{90}(i^3 + 4i^2 - 8i)$$
- $$\sum_{i=1}^{35}(\frac{2^i}{i^2} + \frac{3^i}{i^3})$$

Ans:

```
myfuction <-function(){

  output = c()

  for(i in 1:90){
    output[i] = i^3 + (4 * i^2) - (8*i)
    }
  print(sum(output))
}

myfuction()
```

```
## [1] 17724525
```

Ans:

```
myfuction <-function(){

  output = c()

  for(i in 1:35){
    output[i] =  ((2^i / i^2) + (3^i / i^3))
    }
  print(sum(output))
}

myfuction()
```

```
## [1] 1.835362e+12
```