Assignment 2 (Deadline: Oct. 7)

Question One

Simulate a trail which rolls a “die” 100 times. How many times do you see $'5'$?

Answer

```
time <- seq(1:100)
number_time <- c(sample(1:6,100,replacement = T))
j <- 1
for (i in 1:100){
   if (data.frame(time,number_time)$number[i] == 5){
     j = j + 1
    }
}
print(paste("There are",j,"'5' in the trail."))
```

Question two

The data file “country.txt” contains numerous population indicators for a sample of 121 countries. Read the data file into R and answer the following questions. (Download the data from Ispace.)

Answer

```
country <- read.table("country.txt") # before enter this command, please
use getwd() to check the path

# part one
length(country)
print(paste("There are ",length(country),"veriables in this data set"))

length(country$V1)
print(paste("There are ",length(country$V1),"contries in this data set"))

# part two
developing_percentage <- length(country$V5[country$V5 == "developing"]) /
length(country$V5[-1])
paste(sprintf("%.1f%%", developing_percentage ))
# part three
V2 <- as.matrix(country$V2)
mean(as.numeric(V2[-1]))
```

```
print(paste("The mean of the variable GDP is ",mean(as.numeric(V2[-1]))))

# part Four
V4 <- as.matrix(country$V4)
print(paste("The standard deviation is ",sd(as.numeric(V4[-1]))))

# part Five
V2 <- as.matrix(country$V2)
print(paste("The range of GDP is
(",min(as.numeric(V2[-1])),",",max(as.numeric(V2[-1])),")"))

# part six
V2 <- as.matrix(country$V2)
V3 <- as.matrix(country$V3)
```

Question three

Write some lines of code that will figure out how many terms in the sum 1+2+3+... it requires for the sum to exceed one million.

```
i <- 0
sum <- 0

repeat{
  i = i + 1
  sum = sum + i

  if (sum >= 1000000 ){
    break
  }
}
print(paste("After",i,"times rolling, the first sum which is exceeded one
million is:",sum))
```

Question Four

Write a program to calculate the sum 1+2+3+...+300. Display the total after every 20 terms by using ‘if’ statement to check if the current number of terms is a multiple of 20.

```
for ( i in 0:299) {
  i = i + 1
  sum = sum + i
  if(i %% 20 == 0){ # every 20 terms we make a judgement
    # check if the number is the multiple of 20
    if(i %% 20 == 0) {
      print(paste("The",i,"the current number  is",i,"which is the multiple of 20."))
    } else if (i %% 20 != 0) {
      print(paste("The",i,"the current number is",i,"which is not the multiple of 20."))
    }
  }
}
```

Question Five

The game "3X+1" goes like this. If N is odd, multiply it by 3 and add 1. If N is even, divide it by 2. Repeat until N equals 1, if ever. Every value of N that anyone has ever checked eventually leads to 1, but it is an open mathematical problem (known as the Collatz conjecture) whether EVERY value of N eventually leads to 1.

Answer

```
collatz <- function(n, acc=c()) {
  if(n==1) return(c(acc, 1));
  collatz(ifelse(n%%2==0, n/2, 3*n +1), c(acc, n))}
```

Question Six

Write a program ThreeX that, given an input value of N, will play the game "3X+1" and print to the screen the values visited along the way. The program also needs to return the maximum value hit and the number of steps taken.

Answer

```
collatz <- function(n, acc=c()) {
  if(n==1) return(c(acc, 1));
  collatz(ifelse(n%%2==0, n/2, 3*n +1), c(acc, n))}
```

Question Seven

Write a function to compute running medians. Running medians is a simple smoothing method usually applied to time-series. For example, for the numbers $7, 5, 2, 8, 5, 5, 9, 4, 7, 8$ the running medians of length 3 are $5, 5, 5, 5, 5, 5, 7, 7$. The first running median is the median of the three numbers $7, 5$, and 2; the second running median is the median of $5, 2$, and 8; and so on. Your function should take two arguments: the data (say, x), and the number of observations for each median $(say, length)$. Notice that there are fewer running medians than observations. How many fewer?

Answer

```
install.packages("zoo")
library(zoo)
x <- 1:100
rollmedian(x,5)
```

Question Eight

Write a recursive R function to evaluate the value of y from the following equation where x is a vector of length n containing non-zero elements. The input of the function is vector of x and output function of is the value of y.

$$y = x(n) + \cfrac{1}{x(n-1) + \cfrac{1}{x(n-2) + \cfrac{1}{\cdots + \cfrac{1}{x(2) + \cfrac{1}{x(1)}}}}}$$

Answer

```
Question_Eight <- function(x){
  i <- 1
  n <- length(x)
  answer <- x[1]
    for(i in 1:(n-1)){
      y <- 1 / answer
      if(i == n-1){
        answer <- (x[i + 1] + y)
      }
      else {
        answer <- 1/(x[i] + y)
      }
    }
  return(answer)
}
```

Question Nine

The modification of the Broca’s formula is often used to calculate ideal body weight (IBW). The details are as follows: For men: ideal body weight (in kilograms)=(height (in centimeter)−100) × 0.90 For women: ideal body weight (in kilograms)=(height(in centimeter)-105) × 0.92 Note: the above formulas are fit for adults. Body weights within (1 ± 20%) × IBW are considered as normal. Outside this range, body weights are overweight or underweight. Write an R function, such that given the adult’s gender, height and weight, we can get the ideal body weight and the level of weight (normal, overweight, underweight).

Anwser

```
sex <- readline()
height <- readline()
height <- as.numeric(height)

if (height < 2 ){
  height = height * 100
}
if ( sex == 'female'){
  sex <- 'f'
}
if ( sex == 'male'){
  sex <- 'm'
}
if ( sex == 'f'){
  weight = (height - 105) * 0.92
  print(paste("You are a female, your ideal weight is between (",weight *
0.8, weight * 1.2,")"))
  }
if ( sex == 'm'){
  weight = (height - 100) * 0.90
  print(paste("You are a male, your ideal weight is between (",weight *
0.8, weight * 1.2,")"))
}
```

Question Ten

Consider a data frame df:

```
Id=c(1:10)
Age=c(14,12,15,10,23,21,41,56,78,12)
Sex=c('F','M','M','F','M','F','M','M','F','M')
Code=letters[1:10]
df=data.frame(Id,Age,Sex,Code)
```

Create a function that, given a data frame and two indexes, exchanges two values of the Code variable with each other.

Answer

```
interchange <- function(x,y){
  Id=c(1:10)
  Age=c(14,12,15,10,23,21,41,56,78,12)
  Sex=c('F','M','M','F','M','F','M','M','F','M')
  Code=letters[1:10]
  df=data.frame(Id,Age,Sex,Code)
  df[5] <- df[y]
  df[y] <- df[x]
  df[x] <- df[5]
  return(df[-5])
}
```

Question Eleven

Create a function that given a numeric vector X returns the digits 0 to 9 that are not in X. If $X = 0, 2, 4, 8$ the function return $1, 3, 5, 6, 7, 9$.

Answer

Answer

```
elimination <- function (x, y)
{
    x <- as.vector(x)
    y <- as.vector(y)
    unique(if (length(x) || length(y))
        x[match(x, y, 0L) == 0L]
    else x)
}
```

Challenge Problem

Problem: Write an R function to calculate the reduced row-echelon form of a matrix by Gaussian elimination. Critical code should be well-commented which clearly explains what the code is accomplishing.

```
A <- matrix(c(-2,4,6,0,0,0,-1,1,1,2,0,2),ncol = 4,nrow = 3)
reduced_row_echelon_form <- function(A)
{
    stopifnot(is.numeric(A))
    if (!is.matrix(A))
        stop("Input parameter 'A' must be a matrix.")
    nr <- nrow(A)
    nc <- ncol(A)
    tol <- eps() * max(nr, nc) * max(abs(A))
    r <- 1
    for (i in 1:nc) {
        pivot <- which.max(abs(A[r:nr, i]))
        pivot <- r + pivot - 1
        m <- abs(A[pivot, i])
        if (m <= tol) {
            A[r:nr, i] <- 0
        }
        else {
            A[c(pivot, r), i:nc] <- A[c(r, pivot), i:nc]
            A[r, i:nc] <- A[r, i:nc]/A[r, i]
            if (r == 1) {
                ridx <- c((r + 1):nr)
            }
            else if (r == nr) {
                ridx <- c(1:(r - 1))
            }
            else {
                ridx <- c(1:(r - 1), (r + 1):nr)
            }
            A[ridx, i:nc] <- A[ridx, i:nc] - A[ridx, i, drop = FALSE] %*%
                A[r, i:nc, drop = FALSE]
            if (r == nr)
                break
            r <- r + 1
        }
    }
    A[abs(A) < tol] <- 0
    return(A)
}
```