Applied Data Science Using R - INF 2167H - Assignment #1

Faria Khandaker

Use RStudio for this assignment. Edit the file `assignment01_Fall2020.Rmd` and insert your R code wherever you see the string "INSERT YOUR ANSWER HERE"

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. To be able to read the questions well, knit the document into word format first.

Sample Question and Solution

Use `seq()` to create the vector $(1, 2, 3, ..., 10)$.

```
seq(1,10)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

To properly read some questions (ex: question 2), knit the document to word or pdf before you start solving the assignment

Question 1

In the following exercises of question 1, use only seq, rep, or c

a) (1.5 points) Create the vector (5, 9, 13, …, 41). Note that each term in this sequence is of the form 1 + 4n where n = 1, …, 10.

```
n=1:10
a<-1+4*n
c(a)
```

```
##  [1]  5  9 13 17 21 25 29 33 37 41
```

b) (1.5 points) Create the vector (2, 3, 4, …, 10, 9, 8, …, 2).

```
m=c(seq(2,10),sort(seq(9,2), decreasing = T))
m
```

```
##  [1]  2  3  4  5  6  7  8  9 10  9  8  7  6  5  4  3  2
```

c) (1.5 points) Create the vector (1,2,3,…,1,2,3) in which the sequence (1,2,3) is repeated 5 times.

```r
rep(c(1,2,3),5)
```

```
##  [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

d) (1.5 points) Create the vector (1,1,…,1,2,2,…,2,3,3,…,3) where each number is repeated 7 times.

```r
a=1
b=2
c=3
num=c(rep(1,7),rep(2,7),rep(3,7))
num
```

```
##  [1] 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3
```

e) (1.5 points) Create the vector (10,20,20,30,30,30,…,100,…,100) where 10n is repeated n times.

```r
y=1:10
z<-c(rep(10*y,y))

z
```

```
##  [1]  10  20  20  30  30  30  40  40  40  40  50  50  50  50  50  60  60  60  60
## [20]  60  60  70  70  70  70  70  70  70  80  80  80  80  80  80  80  80  90  90
## [39]  90  90  90  90  90  90  90 100 100 100 100 100 100 100 100 100 100
```

Question 2

a) (1.5 points) Compute:

$$\sum_{n=1}^{100} n$$

```r
sum(1:100)
```

```
## [1] 5050
```

b) (1.5 points) Compute:

$$\sum_{n=1}^{100} n^2$$

```r
n=seq(100)
sum(n^2)
```

```
## [1] 338350
```

c) (1.5 points) Compute:

$$\sum_{n=10}^{20} \left(\frac{2^n}{n} + \frac{3^n}{n^3} \right)$$

```r
d=seq(10,20)
sum(((2^d)/d)+((3^d)/d^3))
```

```
## [1] 826751
```

d) (1.5 points) Compute:

$$\sum_{n=0}^{10} \frac{1}{n!}$$

Hint: Use `factorial(n)` to compute $n!$

```r
e=seq(10)
sum(1/factorial(e))
```

```
## [1] 1.718282
```

e) (1.5 points) Compute:

$$\sum_{n=1}^{20} \left(2n + \frac{1}{n^2}\right)$$

```r
f=seq(20)
sum((2*f)+(1/(f^2)))
```

```
## [1] 421.5962
```

Question 3

a) (1.5 point) Create an empty list `mylist`.

```r
mylist<-list()
mylist
```

```
## list()
```

b) (1.5 points) Add a component named `aa` whose value is 42.

```r
mylist<-list(mylist,"aa"=42)
```

c) (2.5 points) Add a component named `bb` whose value is the numeric vector $(1, 2, \ldots, 10)$.

```r
mylist<-list(mylist,"bb"=c(1:10))
```

d) (2 points) Add a component named `cc` whose value is the character vector ("Hello", "INF 2167").

```r
mylist<-list(mylist, "cc"= c("Hello", "INF 2167"))
```

e) (2 points) Add a component named `dd` whose value is a 4x3 matrix whose elements are $(1, 2, \ldots, 12)$ in column-major order.

```
mat <- matrix(1:12, 4, 3)
mylist<-list(mylist, "dd"= mat)
```

f) (0.5 point) Display `mylist` on the screen.

```
mylist
```

```
## [[1]]
## [[1]][[1]]
## [[1]][[1]][[1]]
## [[1]][[1]][[1]][[1]]
## list()
##
## [[1]][[1]][[1]]$aa
## [1] 42
##
##
## [[1]][[1]]$bb
##  [1]  1  2  3  4  5  6  7  8  9 10
##
##
## [[1]]$cc
## [1] "Hello"    "INF 2167"
##
##
## $dd
##      [,1] [,2] [,3]
## [1,]    1    5    9
## [2,]    2    6   10
## [3,]    3    7   11
## [4,]    4    8   12
```

Question 4

If you have not already done so, install the ISwR package on your computer using the command `install.packages("ISwR")`. Note that install.packages causes errors while knitting so make sure after you install the package to hash the command (i.e. #install.packages("ISWR"))

Loading the `ISwR` package into the current session.

```
#install.packages("ISwR")
library(ISwR)
```

a) (1 points) Display the head of the `thuesen` data frame.

```
dataset<-data("thuesen")
head(thuesen)
```

```
##   blood.glucose short.velocity
## 1          15.3           1.76
## 2          10.8           1.34
```

```
## 3           8.1           1.27
## 4          19.5           1.47
## 5           7.2           1.27
## 6           5.3           1.49
```

b) (4 points) Compute the mean of each variable using `sapply()`, removing the missing values.

```
nomissing <- na.omit(thuesen)
nomissing$blood.glucose<-as.numeric(nomissing$blood.glucose)
nomissing$short.velocity<-as.numeric(nomissing$short.velocity)
sapply(nomissing,function(x) mean(x))
```

```
##  blood.glucose short.velocity
##      10.373913       1.325652
```

c) (3 points) Create a numeric vectors `n1`, `n2`, and `n3` whose elements are the integers from 1 to 20, their squares, and their cubes.

```
n1=0
n2=0
n3=0
d1=seq(20)
n1=c(d1)
n2=c(d1^2)
n3=c(d1^3)
```

d) (2 points) Create a new data frame `nn` from the above three vectors.

```
nn=data.frame(n1,n2,n3)
```

e) (1 points) Display the tail of `nn`.

```
tail(nn)
```

```
##    n1  n2   n3
## 15 15 225 3375
## 16 16 256 4096
## 17 17 289 4913
## 18 18 324 5832
## 19 19 361 6859
## 20 20 400 8000
```

f) (4 points) Compute the sum of each variable in `nn` using `sapply`.

```
sapply(nn, function(x) sum(x))
```

```
##    n1    n2    n3
##   210  2870 44100
```

Question 5

a) (3 points) Create a 4x4 empty matrix, i.e. all elements equal to NA, display mat1.

```
mat1<- matrix(NA, 4,4)
mat1
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   NA   NA   NA   NA
## [2,]   NA   NA   NA   NA
## [3,]   NA   NA   NA   NA
## [4,]   NA   NA   NA   NA
```

b) (7 points) fill the middle 4 elements with the values 'This' 'is' 'the' 'middle' and display mat1.

```
mat1[2,2]<-"This"
mat1[3,2]<-"is"
mat1[2,3]<-"the"
mat1[3,3]<-"middle"
mat1
```

```
##      [,1] [,2]   [,3]     [,4]
## [1,] NA   NA     NA       NA
## [2,] NA   "This" "the"    NA
## [3,] NA   "is"   "middle" NA
## [4,] NA   NA     NA       NA
```

c) (10 points) This is the code to write the "This is the middle" row-wise

```
mat2<- matrix(NA, 4,4)
mat2[2,2]<-"This"
mat2[2,3]<-"is"
mat2[3,2]<-"the"
mat2[3,3]<-"middle"
mat2
```

```
##      [,1] [,2]   [,3]     [,4]
## [1,] NA   NA     NA       NA
## [2,] NA   "This" "is"     NA
## [3,] NA   "the"  "middle" NA
## [4,] NA   NA     NA       NA
```

Question 6

Use the tidyverse library a) (2 points) Import the dataset WineQuality (4898 rows x12 columns) available in the following website http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv, note that the type of separator in this dataset = ;

```
library(tidyverse)
winequality<-read.csv("http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality
```

b) (2 points) Add a column named total.acidity which is the sum of fixed.acidity and volatile.acidity Display the head of the new column

```
winequality<-mutate(winequality, total.acidity= fixed.acidity+volatile.acidity)
head(winequality$total.acidity)
```

```
## [1] 7.27 6.60 8.38 7.43 7.43 8.38
```

c) (4 points) Create a dataframe named bestWine to only include wine of best quality How many cases have wine of best quality? A wine of best quality is wine that has highest value in quality

```
#max(winequality$quality)
bestWine<- filter(winequality,winequality$quality >= 9)
nrow(bestWine)
```

```
## [1] 5
```

d) (4 points) Display the head of bestWine showing only the pH, alcohol, and quality

```
head(select(bestWine, pH,alcohol,quality))
```

```
##     pH alcohol quality
## 1 3.20    10.4       9
## 2 3.41    12.4       9
## 3 3.28    12.5       9
## 4 3.28    12.7       9
## 5 3.37    12.9       9
```

e) (5 points) Sort the WineQuality dataset in descending order of alcohol concentration Display only the tail of the alcohol, quality and density

```
sortedWine<-arrange(winequality, desc(winequality$alcohol))
tail(select(sortedWine, alcohol, quality, density))
```

```
##      alcohol quality density
## 4893     8.5       5 0.99398
## 4894     8.4       5 0.99429
## 4895     8.4       5 0.99429
## 4896     8.4       4 0.99536
## 4897     8.0       5 0.99332
## 4898     8.0       3 0.99688
```

f) (5 points) Add a variable to the data frame that takes value 1 if the food has higher citric acid than average, 0 otherwise.Call this variable HighAcid. Do the same for High chlorides, High sugar, and High sulphates. How many cases have both high acid and high chlorides?

```r
#calculating the means of sugar, citric acid and sulphates
avacid<-mean(winequality$citric.acid)
avchlor<-mean(winequality$chlorides)
avsug<-mean(winequality$residual.sugar)
avsul<-mean(winequality$sulphates)

#for high citric acid
winequality<-mutate(winequality, high.citric.acid= ifelse(citric.acid>avacid,1,0))

#for high chloride
winequality<-mutate(winequality, high.chloride= ifelse(chlorides>avchlor,1,0))

#for high sugar
winequality<-mutate(winequality, high.sugar= ifelse(residual.sugar>avsug,1,0))

#for high sulphates
winequality<-mutate(winequality, high.sulphate= ifelse(sulphates>avsul,1,0))

#high chlorides and high acid

sum(winequality$high.chloride & winequality$high.citric.acid == 1)
```

```
## [1] 852
```

g) (8 points) Create a function called Wine.check to detect bad quality wine. Use the flowchart attached to the assignment as a basis for this function. Hint: Use nested if statement inside the function

```r
Wine.check <- function(m){
  ifelse(winequality$high.citric.acid==0, "Pass", ifelse(winequality$high.chloride==0, "Pass",
                                                         ifelse(winequality$high.sugar==0, "Pass", "Fai
}
```

h) (7 points) Create a new variable called WineCheck using the output of the function. This variable will have values of either "pass" or "Fail". A "Fail" will occur when all acid, chlorides and sugar are high (=1), display the head of WineCheck

```r
Winecheck<-Wine.check(winequality)
head(Winecheck)
```

```
## [1] "Pass" "Pass" "Fail" "Pass" "Pass" "Fail"
```

i) (3 points) How many wines in the WineQuality data frame fail the WineCheck? (8 points)

```r
winequality<-mutate(winequality, winecheck=Wine.check(winequality1))
sum(winequality$winecheck=="Fail")
```

```
## [1] 478
```

END of Assignment #1. Good Luck!