Department of Statistics
2020/21 – Semester II

STA272 – STATISTICAL COMPUTING

Assignment IV

Due: 26-APR-21 **Time: 12h00**

Instructions:

- All of your work must be typeset using Rmarkdown and submitted online through the course's blackboard shell.
- Any work submitted late would be penalized as follows:
  - any work submitted before midnight of the due date would attract a penalty of up to 10%
  - any work submitted a day late would attract a penalty of up to 25%
  - any work submitted two days late would attract a penalty of up to 50%
  - otherwise you'll be awarded a zero mark.
- You are encouraged to discuss the assignment with others but at the end you must submit your individual work.
- Any form cheating is not allowed and plagiarized work will be awarded a zero mark.

Q1. Examine the following code snippets and correct them where needed according to what they are intended for.

Please note that you can not a new code that does a similar task.

(a) Compute proportion estimate of obtaining a face of 1 or 2 in a fair die as follows $\hat{p} = \sum_{i=1}^{n} \mathbb{I}_{x_i \leq 2}/n$, where

$$\mathbb{I}_{x_i \leq 2} = \begin{cases} 1 & x_i \leq 2 \\ 0 & \text{otherwise.} \end{cases}$$

```
set.seed(1)
die = sample(c(1:6), 1000, replace = TRUE)
success = rep(0, 1000)

for(i in 1:100)
{
 if(die > 2){success[i] == 1}
}

hat.p = MEAN(success)
```

(b) Calculate $\text{cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})$

```
X = rnorm(50,3,1)
Y = 2*x + rnorm(50)

covFUN = function(x,y){
 xbar = mean(x)
 ybar = mean(y)
 cova = sum(x-xbar)*sum(y-ybar)/(n-1)
 return(cova)
}

covFun(y=Y, X)
```

Q2. The Gini index is defined by the following equation.

$$g(x) = \frac{2\sum_{i=1}^{n} i x_i}{n \sum_{i=1}^{n} x_i} - \frac{n+1}{n}$$

(a) Write down a user-defined function (call it `gini.fn`) that will calculate a value of this index for any given vector.

(b) Evaluate the Gini index for the number of hours spent watching television per week for a sample of 34 households.

```
23.1 15.9 21.0 26.0 25.1 14.7 24.2 16.6 18.2 16.5 20.7 15.3 17.7
19.1 22.7 21.9 14.6 26.3 25.8 9.4 17.0 21.2 17.9 24.7 21.1 17.2
19.1 22.7 24.0 24.7 22.5 8.3 2.5 30.4
```

Q3. A Newton method is one of the most popular numerical techniques used to finding roots of an algebraic function. That is, solving for $f(x) = 0$. According to the Newton method, if $f(x)$ has a first derivative $f'(x)$ then the following algorithm will converge to a root of the above equation if the starting point is close enough.

```
Algorithm
a. Picking a starting value x0
b. For each estimate xn, calculate a new estimate
```

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

```
c. Repeat step (b) until the estimate are very close
   together or until the method fails.
```

Use R and the Newton method to approximate the root of

$$f(x) = \sin^2(2x - 1) - \cos(3x^2); \quad 0 \leq x \leq \pi/2$$

`HINT: A while loop is the most appropriate for this problem. But first use curve to visualize this graph and see where the root are.`

Q4. The Taylor series for $\sin^{-1}(x)$ for $|x| \leq 1$ is given by

$$\sin^{-1}(x) = x + \frac{x^3}{2 \cdot 3} + \frac{3x^5}{2 \cdot 4 \cdot 5} + \frac{3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \frac{3 \cdot 5 \cdot 7 \cdot x^9}{2 \cdot 4 \cdot 6 \cdot 8 \cdot 9} + \ldots$$

(a) Write a function in `R` that will approximate $\sin^{-1}(x)$ to some allowable tolerance error. The arguments of your function should be `x` and `tol` (tolerance error), and return the approximate value of $\sin^{-1}(x)$ and the total number of terms (`N`) summed together to give an approximate within the pre-stated tolerance error.

(b) Use your function to approximate $\sin^{-1}(\pi/4)$ within a tolerance error of 0.001. Compare your solution with the `R` computed value.

EACH QUESTION IS WORTH 5 MARKS