Homework #3

Nathaniel Haulk

10/14/2021

Question 1

```
set.seed(12) # to be reproducible

## Creates a 50 x 10 matrix with the values 1-500
A = matrix(data = runif(n = 1:500), nrow = 50, ncol = 10)

## Adds column names with lake and a number separated with a "_"
colnames(A) = paste("lake", 1:10, sep = "_")


#########################################
## Calculating Average using for loop
#########################################


## Creates a matrix to record means in
AloopMeans = matrix(data = 0, nrow = 1, ncol = 10)

## Adds column names with lake and a number separated with a "_"
colnames(AloopMeans) = paste("lake", 1:10, sep = "_")

## for loop to run through each column and calculate the mean.
for(i in 1:ncol(A)){

  ##
  AloopMeans[i] = sum(A[,i])/NROW(A[,i])
}

## Print means generated using the for loop
print(AloopMeans)
```

```
##         lake_1    lake_2    lake_3    lake_4    lake_5    lake_6    lake_7
## [1,] 0.4601492 0.4992815 0.5987037 0.4580486 0.4719578 0.4965216 0.5110536
##         lake_8    lake_9   lake_10
## [1,] 0.4577936 0.5193423 0.4856413
```

```
####################################
## Using colMeans
####################################

## Records Average from colMeans
AcolMeans = colMeans(A)

## Prints Averages from colMeans
print(AcolMeans)
```

```
##    lake_1    lake_2    lake_3    lake_4    lake_5    lake_6    lake_7    lake_8
## 0.4601492 0.4992815 0.5987037 0.4580486 0.4719578 0.4965216 0.5110536 0.4577936
##    lake_9   lake_10
## 0.5193423 0.4856413
```

Question 2

```
x = array(1:27, dim = c(3, 3, 3))

## Used to record each number as character


x.ap = NULL

for(i in 1:dim(x)[1]){
  for(q in 1:dim(x)[2]){
    x.char = NULL
    for(u in 1:dim(x)[3]){
      x.char = cbind(x.char,as.character(x[i,q,u]))
    }
   # x.ap[i,q] =
  }
}
```

Question 3

```
## How many numbers of the Fibonacci Sequence you want to generate.
fib.length = 30

## Makes a list of 30 elements all set to 0.
Fib.seq = matrix(0, ncol = 1, nrow=fib.length)

## Sets the second term of the matrix to 1.
Fib.seq[2] = 1

## For loop to calcuate sum of previous two values.
for (i in 3:fib.length){
  Fib.seq[i] = Fib.seq[i-1] + Fib.seq[i-2]
```

```r
} # Closes for loop

## Prints out the first 30 terms of the Fibonacci Sequence.
print(Fib.seq)
```

```
##         [,1]
##  [1,]      0
##  [2,]      1
##  [3,]      1
##  [4,]      2
##  [5,]      3
##  [6,]      5
##  [7,]      8
##  [8,]     13
##  [9,]     21
## [10,]     34
## [11,]     55
## [12,]     89
## [13,]    144
## [14,]    233
## [15,]    377
## [16,]    610
## [17,]    987
## [18,]   1597
## [19,]   2584
## [20,]   4181
## [21,]   6765
## [22,]  10946
## [23,]  17711
## [24,]  28657
## [25,]  46368
## [26,]  75025
## [27,] 121393
## [28,] 196418
## [29,] 317811
## [30,] 514229
```

Question 4

```r
## Lists the top 105 songs from the radio station KITS San Francisco on Jan 1, 1992.
top105 = readLines("http://www.textfiles.com/music/ktop100.txt")

## Removes missing no. 54 and 55
top105 = top105[-c(64, 65)]

y = grep("^[0-9]+([.][0-9])?", top105, value = T)

z = gsub(" .*$", "",  y)

print(z)
```

```
##   [1] "1."    "2."    "3."    "4."    "5."    "6."    "7."    "8."    "9."
##  [10] "10."   "11."   "12."   "13."   "14."   "15."   "16."   "17."   "18."
##  [19] "19."   "20."   "21."   "22."   "23."   "24."   "25."   "26."   "27."
##  [28] "28."   "29."   "30."   "31."   "32."   "33."   "34."   "35."   "36."
##  [37] "37."   "38."   "39."   "40."   "41."   "42."   "43."   "44."   "45."
##  [46] "46."   "47."   "48."   "49."   "50."   "51."   "52."   "53."   "56."
##  [55] "57."   "58."   "59."   "60."   "61."   "62."   "63."   "64."   "65."
##  [64] "66."   "67."   "68."   "69."   "70."   "71."   "72."   "73."   "74."
##  [73] "75."   "76."   "77."   "78."   "79."   "80."   "81."   "82."   "83."
##  [82] "83."   "84."   "85."   "86."   "87."   "88."   "89."   "90."   "91."
##  [91] "91."   "92."   "93."   "94."   "95."   "96."   "97."   "97."   "98."
## [100] "99."   "100."  "101."  "102."  "103."  "104."  "105."  "105.3"
```

Question 5

```
## Removes the period from each number
A = sub("\\D", "",  z)

## Converts the strings to numbers
A = as.numeric(A)

## creates a list that defines any repeated value as True
dup.data = duplicated(A)

## How many ties are present:
print(sum(dup.data))
```

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

```
## for loop to test for repeats and print out what position the tie occurred at.
for(i in 1:length(dup.data)){


  if(dup.data[i] == TRUE)
    print(A[i])
}
```

```
## [1] 83
## [1] 91
## [1] 97
```

‘