CS 636 Data Analytics with R Programming
Midterm Examination
Thursday, March 12, 2020
Form A

- This midterm examination consists of 10 pages (including this one), 7 questions, and 100 points. Please check to make sure you have all pages.

- This is a closed-book/notes exam, but an A4-sized cheating sheet is allowed.

- Write your answers on the given space for each question.

- Hand in this midterm examination with answers.

- You have 120 minutes to complete the exam.

On my honor, I pledge that I have not violated the provisions of the NJIT Academic Honor Code. I understand that if I do not include this pledge, as required by the instructor, with my signature, the instructor will not grade my work.

Last Name (printed): ____

First Name (printed): ____

UCID: ____

Signature: ____

Question	Score	Possible
1		10
2		20
3		10
4		10
5		10
6		20
7		20
Total		100

Problem 1 (10 pts): Please give the results of following commands (Note that Q1 and Q4 have two outputs):

1) rep(6:1, each=2);rep(5:12, times=2)

```
6 6 5 5 4 4 3 3 2 2 1 1  ;    5  6  7  8  9 10 11 12  5  6  7  8  9 10 11 12
```

2) seq(10, 6, by=3)

```
Error Message
```

3) order(10:20, decreasing=T)

```
[1] 11 10  9  8  7  6  5  4  3  2  1
```

4) print((1:4)<2|(1:4)%%2==0);
print((3:6)>2&&(3:6)%%2==0);

```
[1]  TRUE  TRUE FALSE  TRUE
[1] FALSE
```

5) x = c(1:10, "20");
print(max(x))

```
[1] "9"
```

Problem 2 (20 pts): Let gender = c(1,2,1,1,2,1,2,1,1,2); graduate = c(2,1,1,1,2,2,1,1,2,2); score=c(4:12, NA). A data frame is constructed as zz = data.frame(gender, graduate, score). Give the results of these R commands:

```
   gender graduate score
1      1        2     4
2      2        1     5
3      1        1     6
4      1        1     7
5      2        2     8
6      1        2     9
7      2        1    10
8      1        1    11
9      1        2    12
10     2        2    NA
```

1) table(zz[,"gender"])

```
1 2
6 4
```

2) apply(zz[-1,], 2, max)

```
gender graduate    score
     2        2       NA
```

3) zz[zz[,3]>9,]

```
   gender graduate score
7     2       1       10
8     1       1       11
9     1       2       12
NA   NA      NA       NA
```

4) which.min(zz$score)

```
[1] 1
```

5) zz[order(zz["graduate"],zz["score"]),]

```
   gender graduate score
2     2       1     5
3     1       1     6
4     1       1     7
7     2       1    10
8     1       1    11
1     1       2     4
5     2       2     8
6     1       2     9
9     1       2    12
10    2       2    NA
```

6) subset(zz, zz["gender"]==1)

```
  gender graduate score
1    1       2     4
3    1       1     6
4    1       1     7
6    1       2     9
8    1       1    11
9    1       2    12
```

7) tapply(zz$score, zz$graduate, mean, na.rm=T)

```
   1    2
7.80 8.25
```

8) apply(zz[-10,], 1, sum)

```
1  2  3  4  5   6   7  8  9
7  8  8  9 12  12 13 13 15
```

Problem 3 (10 pts): Let mylist=list(sex = c(2,1,2,1,2,2,2,1,1,2), smoking = c(0,1,0,1,0,1,1,1,0,0), age=c(41:50)). Give the results of these R commands:

1) length(mylist)

```
3
```

```
$sex
 [1] 2 1 2 1 2 2 2 1 1 2

$smoking
 [1] 0 1 0 1 0 1 1 1 0 0

$age
 [1] 41 42 43 44 45 46 47 48 49 50
```

2) lapply(mylist[1:2], function(addition){ addition+3})

```
$sex
 [1] 5 4 5 4 5 5 5 4 4 5

$smoking
 [1] 3 4 3 4 3 4 4 4 3 3
```

3) sapply(mylist, max)

```
sex smoking     age
  2       1      50
```

4) (mylist$sex-mylist$smoking)^3

```
[1] 8 0 8 0 8 1 1 0 1 8
```

Problem 4 (10 pts): Define zz=matrix(c(c(1,2,NA,4), seq(4,8), rep(5,6)), nrow = 3, ncol = 5). Use apply() function to compute the sum of each **row** and the average of each **column**(**ignore any missing value**).

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    4    6    5    5
[2,]    2    4    7    5    5
[3,]   NA    5    8    5    5
```

```
apply(zz, 1, sum)  ## Produces: 21 23 NA
```

```
apply(zz, 2, mean) ## Produces:  NA 4.333333 7.000000 5.000000 5.000000
```

Problem 5 (10 pts): Define geneExpr=data.frame(gene=LETTERS[1:10], expr1=c(6.1, 4.2, 2.8, 0.9, 0.1, 3.0, 2.6, 8.2, 3.4, 6.8), expr2=c(2.1, 4.5, 6.8, 7.9, 8.1, 5.0, 4.6, 3.2, 3.5, 7.8)). Give the results of the following R commands:

```
   gene expr1 expr2
1     A   6.1   2.1
2     B   4.2   4.5
3     C   2.8   6.8
4     D   0.9   7.9
5     E   0.1   8.1
6     F   3.0   5.0
7     G   2.6   4.6
8     H   8.2   3.2
9     I   3.4   3.5
10    J   6.8   7.8
```

a)
```
counter<-0
for (i in 1:length(geneExpr[,2])) {
        if (geneExpr[i,3]<=4){
                counter<-counter+1
        }
}
print(counter)
    3
```

b)
```
myMin<-function(x, k) { max(x[x<k]) }
apply(geneExpr[,-1],2, myMin, k=6)

expr1 expr2
  4.2   5.0
```

Problem 6 (20 pts): Please complete the following questions:

a) Please write a function, similar as dnorm(), named fun_nd, to compute the normal distribution. Given three parameters, variable x, mean, and standard deviation, it is returned based on the following formula:

$$y = \frac{1}{\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma}$$

$\mu =$ Mean
$\sigma =$ Standard Deviation

The results should print out the value of y. Testing code and expected outputs are shown below:

```
> fun_nd(1.5, 1, 4)
[1] 0.09895942
```

```
fun_nd<-function(x,mean,std){
  y<-0
  y<-exp(-(x-mean)^2/(2*std))/(std*sqrt(2*pi))
  return(y)
}
fun_nd(1.5,1,4)
```

b) Define a sequence as

$$a_n = 2 * a_{n-1} + a_{n-2}$$

starts with two 1s, $a_1 = 1, a_2 = 1$. Please write a function, Seq(n), which takes n as the input parameter and is supposed to return the n-th number in the defined sequence.

Problem 7 (20 pts): Please answer following questions：

a) Please write the results of the following R commands, library(stringr) already loaded:

1)
```
hw <- "Harry Potter";
str_sub(hw, end = 8);
```

```
[1] "Harry Po"
```

2)
```
str_pad("a", 5, pad = c("*", "_", "0"))
```

```
[1] "****a" "____a" "0000a"
```

3)
```
fruit <- c("pinapple", "pear", "banana", "apple");
str_detect(fruit, "^p", negate = TRUE)
```

```
[1] FALSE FALSE  TRUE  TRUE
```

4)
```
x <- c("<a> <b>", "<a> <>", "<a>", "", NA);
str_match(x, "<(.*?)> <(.*?)>")
```

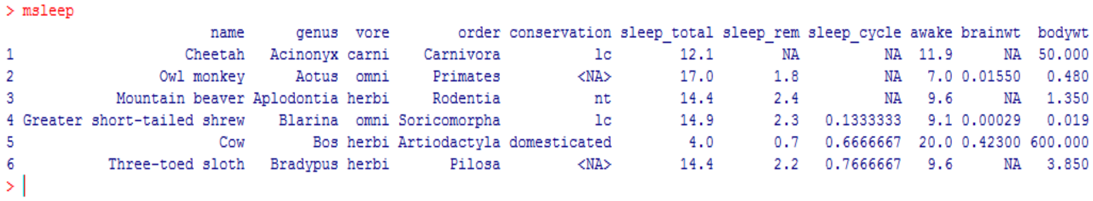
```
     [,1]      [,2] [,3]
[1,] "<a> <b>" "a"  "b"
[2,] "<a> <>"  "a"  ""
[3,] NA        NA   NA
[4,] NA        NA   NA
[5,] NA        NA   NA
```

5)
```
fruits <- c("one apple", "two pears", "three bananas");
str_replace(fruits, "[aeiou]", c("1", "2", "3"))
```

```
[1] "1ne apple"     "tw2 pears"     "thr3e bananas"
```

b) Suppose we have a data frame named msleep

```
> library("dplyr")
> msleep <- read.csv("msleep_ggplot2.csv")
> msleep = head(msleep)
>msleep
```



```
> msleep
                        name      genus  vore        order conservation sleep_total sleep_rem sleep_cycle awake brainwt  bodywt
1                    Cheetah   Acinonyx carni    Carnivora           lc        12.1        NA          NA  11.9      NA  50.000
2                 Owl monkey      Aotus  omni     Primates         <NA>        17.0       1.8          NA   7.0 0.01550   0.480
3            Mountain beaver Aplodontia herbi     Rodentia           nt        14.4       2.4          NA   9.6      NA   1.350
4 Greater short-tailed shrew    Blarina  omni Soricomorpha           lc        14.9       2.3   0.1333333   9.1 0.00029   0.019
5                        Cow        Bos herbi Artiodactyla domesticated         4.0       0.7   0.6666667  20.0 0.42300 600.000
6           Three-toed sloth   Bradypus herbi       Pilosa         <NA>        14.4       2.2   0.7666667   9.6      NA   3.850
> |
```

Please rewrite the following commands using equivalent functions from ‘dplyr’ package, which is supposed to return the same results.

1) msleep[, grep("^sl", colnames(msleep))]

select(msleep, starts_with("sl"))

```
  sleep_total sleep_rem sleep_cycle
1       12.1        NA          NA
2       17.0       1.8          NA
3       14.4       2.4          NA
4       14.9       2.3   0.1333333
5        4.0       0.7   0.6666667
6       14.4       2.2   0.7666667
```

2) subset(msleep, sleep_total>10& !is.na(sleep_cycle))

filter(msleep, sleep_total>10 &! is.na(sleep_cycle))

3) msleep[order(msleep$sleep_total, decreasing= T),]

arrange(msleep, desc(sleep_total))

4) colnames(msleep)[colnames(msleep)=="sleep_total"] = "st"

5) msleep$sleep_total_min = msleep$sleep_total * 60