Sub : Foundations for Data Analytics

Name: Gudi Varaprasad

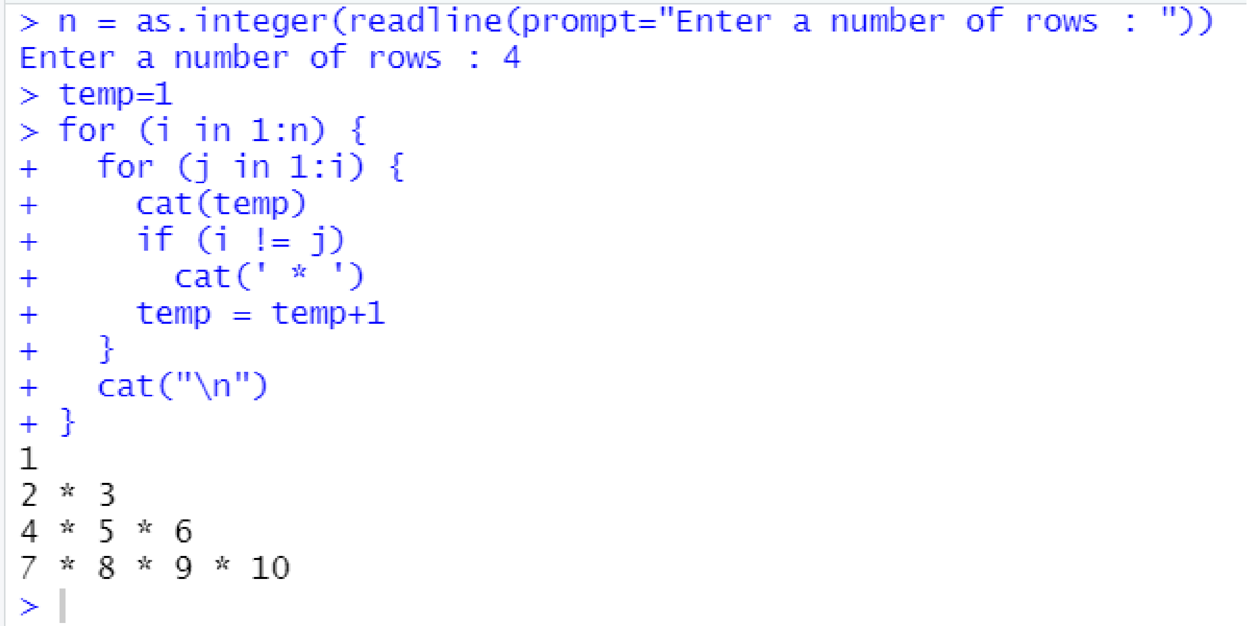
Reg.No: 19BCE7048

Slot No: L33+L34

Assignment 6 : **Date: 09/03/2021**

1. WAP in R to print the given pattern and take n from the user

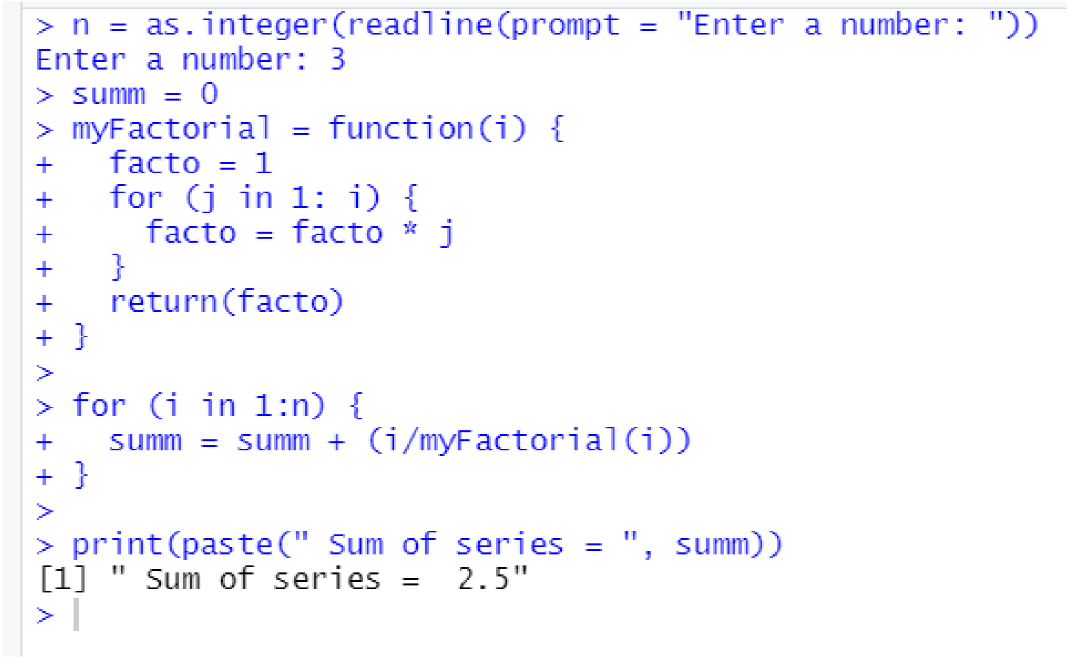
```
1
2    *    3
4    *    5    *    6
7    *    8    *    9    *    10
```



```
> n = as.integer(readline(prompt="Enter a number of rows : "))
Enter a number of rows : 4
> temp=1
> for (i in 1:n) {
+   for (j in 1:i) {
+     cat(temp)
+     if (i != j)
+       cat(' * ')
+     temp = temp+1
+   }
+   cat("\n")
+ }
1
2 * 3
4 * 5 * 6
7 * 8 * 9 * 10
> 
```

2. WAP in R to find the sum ( ) of the series :

1/1! + 2/2! + 3/3! +...+N/N!. Use method user defined method for factorial ( ). (function calling with in function)



```
> n = as.integer(readline(prompt = "Enter a number: "))
Enter a number: 3
> summ = 0
> myFactorial = function(i) {
+   facto = 1
+   for (j in 1: i) {
+     facto = facto * j
+   }
+   return(facto)
+ }
>
> for (i in 1:n) {
+   summ = summ + (i/myFactorial(i))
+ }
>
> print(paste(" Sum of series = ", summ))
[1] " Sum of series =  2.5"
> 
```

3. Convert the data frame1 to data frame2 as given format. Create groupings or categories for infant, children, young, adults and elderly as given below

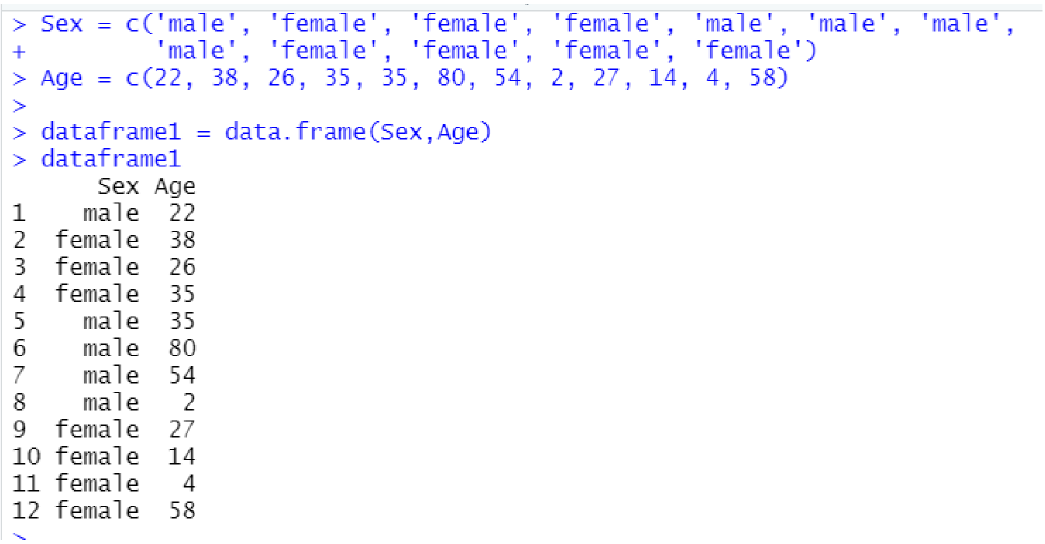
0 to 2 = ‘Toddler/Baby’

3 to 17 = ‘Child’

19 to 40 = ‘Young’

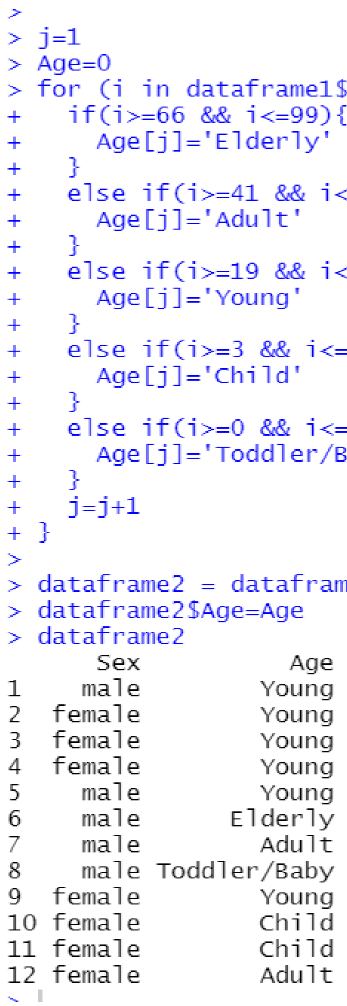
41 to 65 = ‘Adult’

66 to 99=’Elderly’



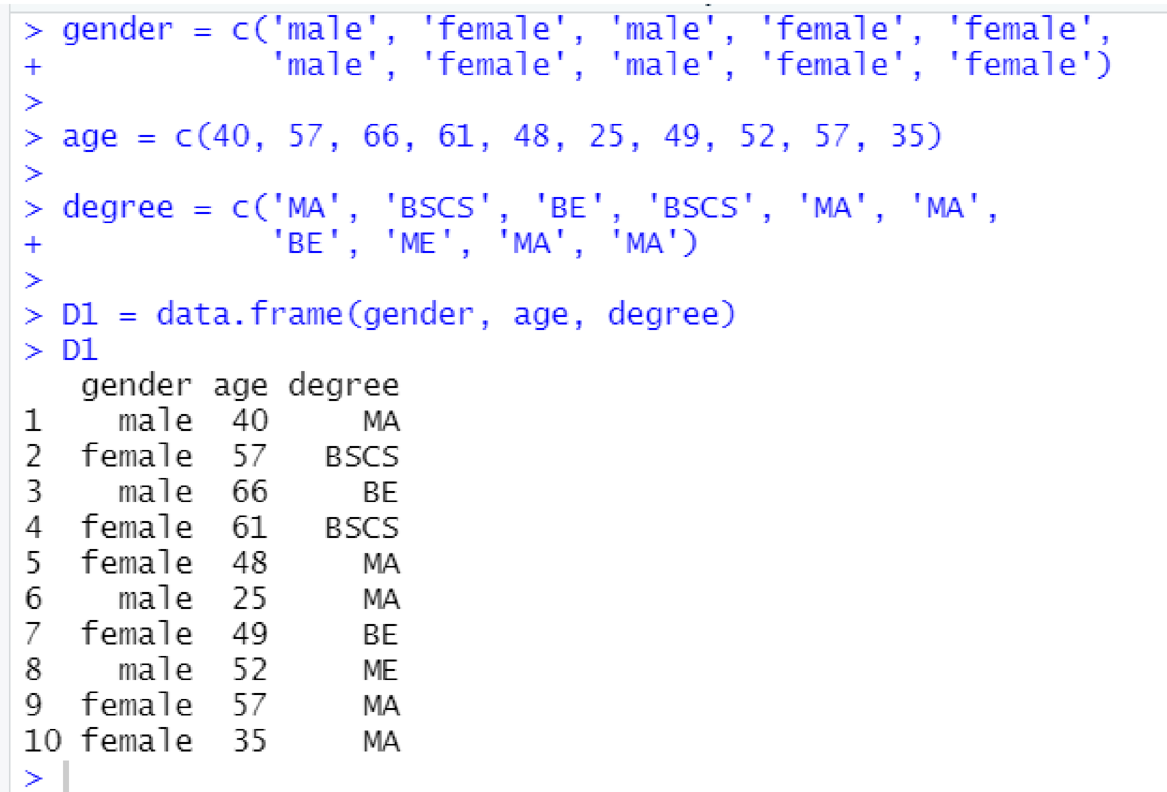
```
> Sex = c('male', 'female', 'female', 'female', 'male', 'male', 'male',
+         'male', 'female', 'female', 'female', 'female')
> Age = c(22, 38, 26, 35, 35, 80, 54, 2, 27, 14, 4, 58)
>
> dataframe1 = data.frame(Sex,Age)
> dataframe1
      Sex Age
1    male  22
2  female  38
3  female  26
4  female  35
5    male  35
6    male  80
7    male  54
8    male   2
9  female  27
10 female  14
11 female   4
12 female  58
>
```

After grouping the data frame :



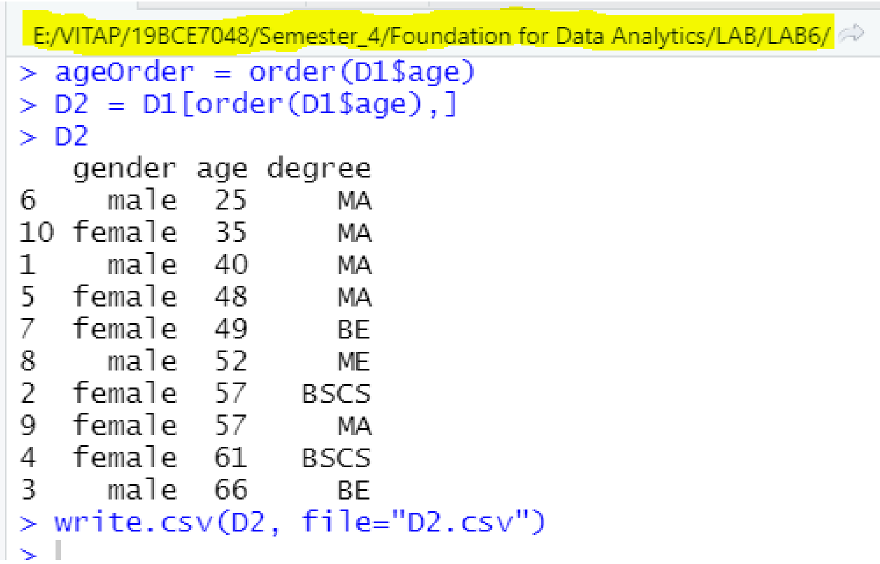
```
>
> j=1
> Age=0
> for (i in dataframe1$Age){
+   if(i>=66 && i<=99){
+     Age[j]='Elderly'
+   }
+   else if(i>=41 && i<=65){
+     Age[j]='Adult'
+   }
+   else if(i>=19 && i<=40){
+     Age[j]='Young'
+   }
+   else if(i>=3 && i<=17){
+     Age[j]='Child'
+   }
+   else if(i>=0 && i<=2){
+     Age[j]='Toddler/Baby'
+   }
+   j=j+1
+ }
>
> dataframe2 = dataframe1
> dataframe2$Age=Age
> dataframe2
      Sex          Age
1    male        Young
2  female        Young
3  female        Young
4  female        Young
5    male        Young
6    male      Elderly
7    male        Adult
8    male Toddler/Baby
9  female        Young
10 female        Child
11 female        Child
12 female        Adult
>
```

4. Create a Data frame as given below as D1



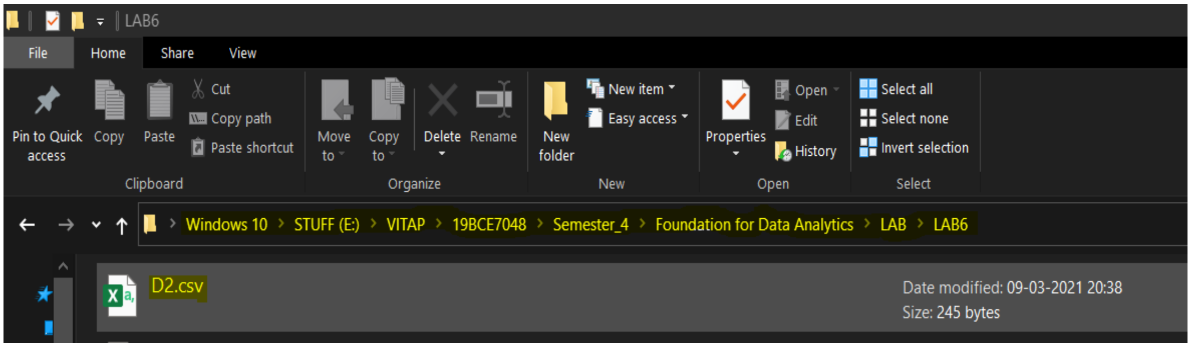
```
> gender = c('male', 'female', 'male', 'female', 'female',
+            'male', 'female', 'male', 'female', 'female')
>
> age = c(40, 57, 66, 61, 48, 25, 49, 52, 57, 35)
>
> degree = c('MA', 'BSCS', 'BE', 'BSCS', 'MA', 'MA',
+            'BE', 'ME', 'MA', 'MA')
>
> D1 = data.frame(gender, age, degree)
> D1
   gender age degree
1    male  40     MA
2  female  57   BSCS
3    male  66     BE
4  female  61   BSCS
5  female  48     MA
6    male  25     MA
7  female  49     BE
8    male  52     ME
9  female  57     MA
10 female  35     MA
> 
```

i) Sort the data frame D1 in the ascending order by using order () based on the variable age and save as D2.



```
E:/VITAP/19BCE7048/Semester_4/Foundation for Data Analytics/LAB/LAB6/
> ageOrder = order(D1$age)
> D2 = D1[order(D1$age),]
> D2
   gender age degree
6    male  25     MA
10 female  35     MA
1    male  40     MA
5  female  48     MA
7  female  49     BE
8    male  52     ME
2  female  57   BSCS
9  female  57     MA
4  female  61   BSCS
3    male  66     BE
> write.csv(D2, file="D2.csv")
> 
```

The saved Excel File (D2.csv) :



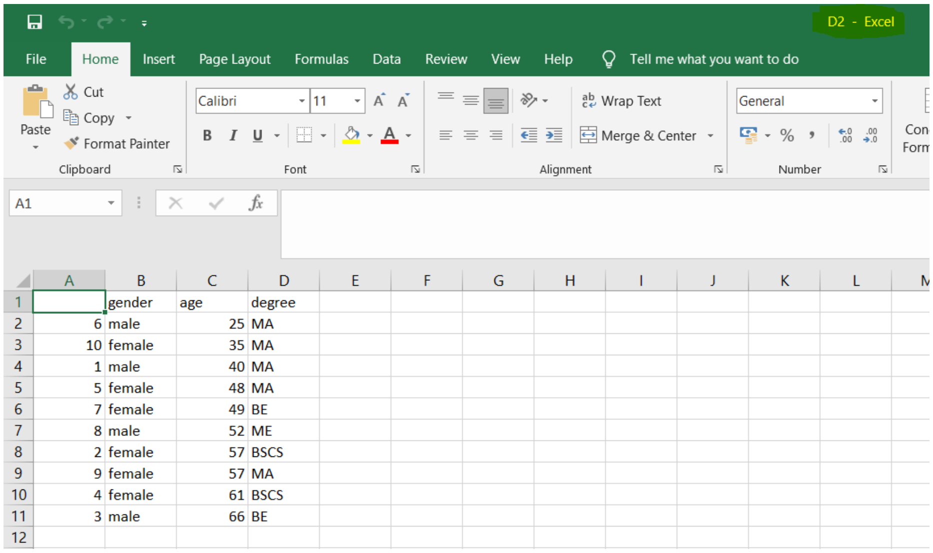
Windows 10 › STUFF (E:) › VITAP › 19BCE7048 › Semester_4 › Foundation for Data Analytics › LAB › LAB6

D2.csv

Date modified: 09-03-2021 20:38

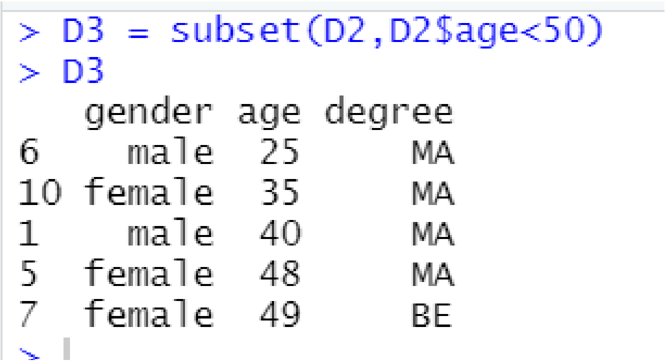
Size: 245 bytes

D2.csv :



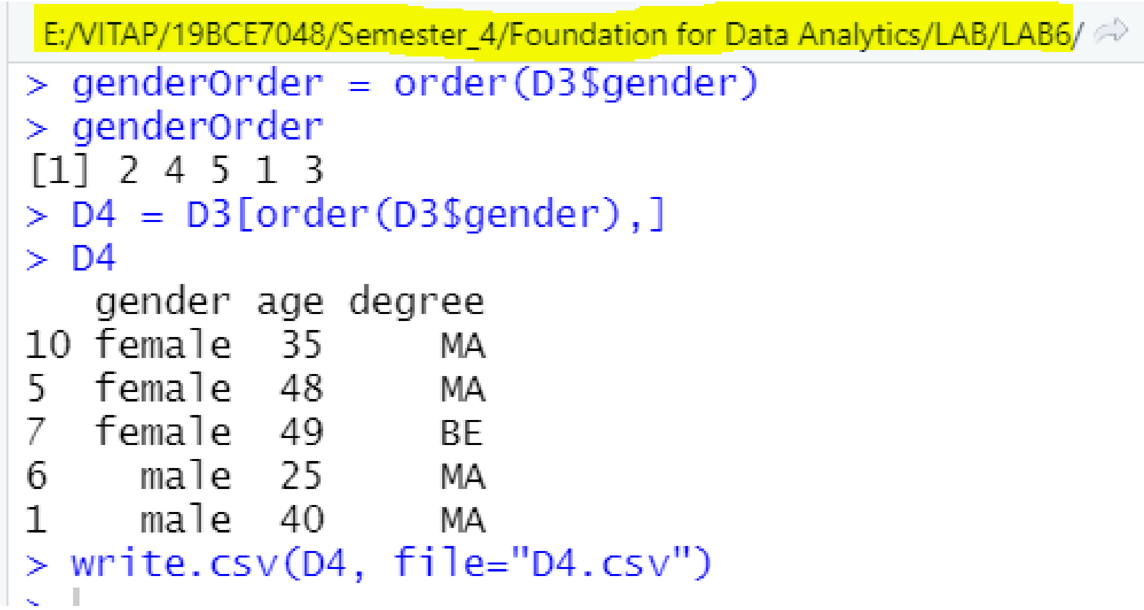
	A	B	C	D
1		gender	age	degree
2	6	male	25	MA
3	10	female	35	MA
4	1	male	40	MA
5	5	female	48	MA
6	7	female	49	BE
7	8	male	52	ME
8	2	female	57	BSCS
9	9	female	57	MA
10	4	female	61	BSCS
11	3	male	66	BE
12				

ii) Create data frame D3 from D2 where age is below 50.



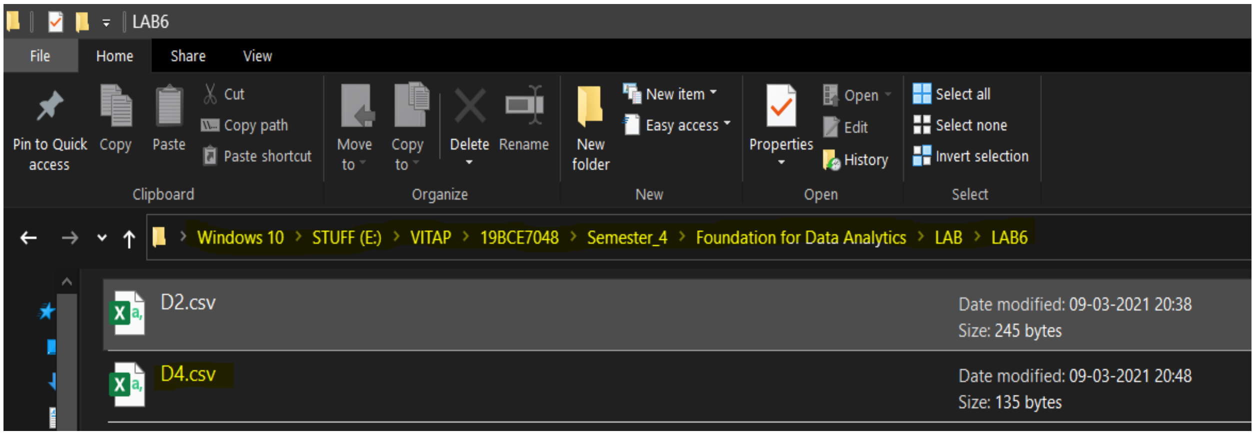
```
> D3 = subset(D2,D2$age<50)
> D3
   gender age degree
6    male  25     MA
10 female  35     MA
1    male  40     MA
5  female  48     MA
7  female  49     BE
>
```

iii) Again sort D3 ascending order by using order () based on the variable Gender and save as D4.



```
E:/VITAP/19BCE7048/Semester_4/Foundation for Data Analytics/LAB/LAB6/
> genderOrder = order(D3$gender)
> genderOrder
[1] 2 4 5 1 3
> D4 = D3[order(D3$gender),]
> D4
   gender age degree
10 female  35     MA
5  female  48     MA
7  female  49     BE
6    male  25     MA
1    male  40     MA
> write.csv(D4, file="D4.csv")
```

The saved Excel File (D4.csv) :

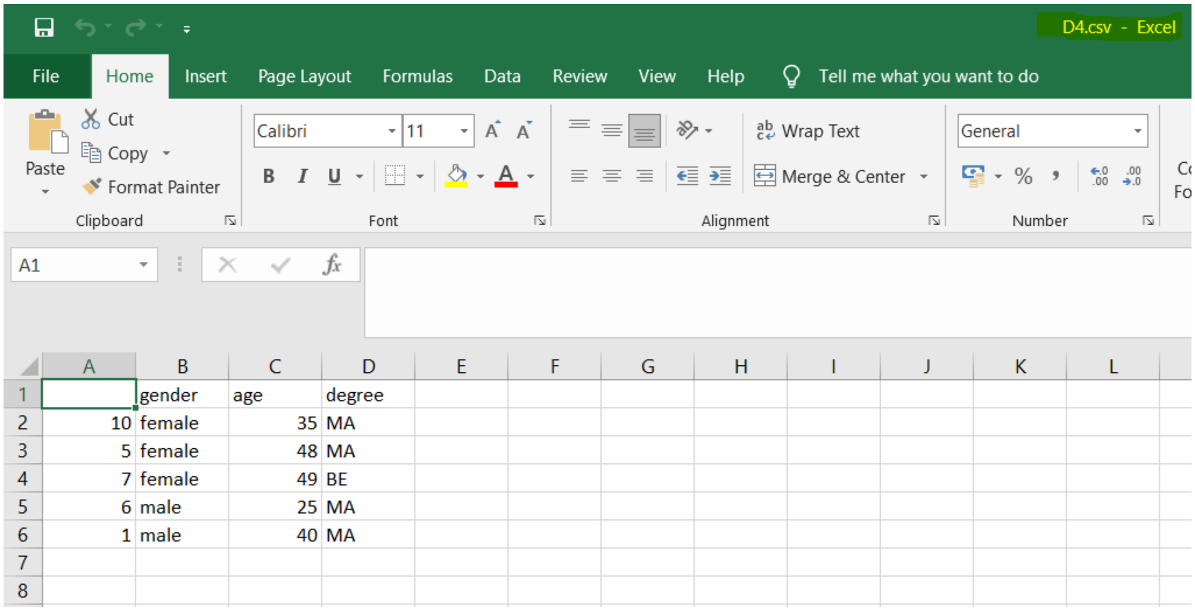


Windows 10 › STUFF (E:) › VITAP › 19BCE7048 › Semester_4 › Foundation for Data Analytics › LAB › LAB6

D2.csv Date modified: 09-03-2021 20:38 Size: 245 bytes

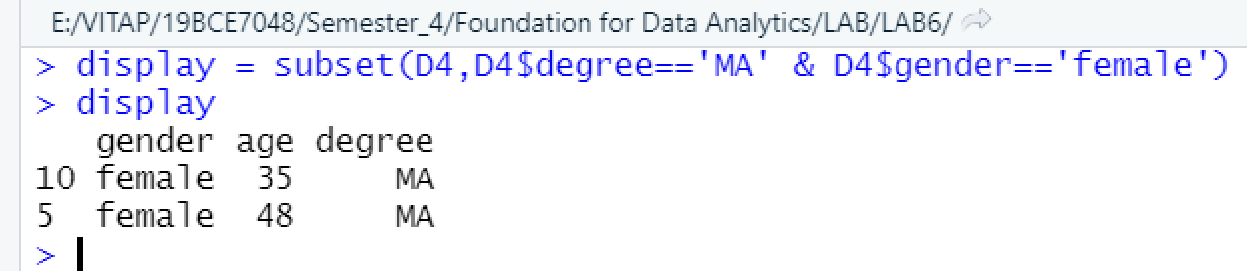
D4.csv Date modified: 09-03-2021 20:48 Size: 135 bytes

D4.csv :



	A	B	C	D	E	F	G	H	I	J	K	L
1		gender	age	degree								
2	10	female	35	MA								
3	5	female	48	MA								
4	7	female	49	BE								
5	6	male	25	MA								
6	1	male	40	MA								
7												
8												

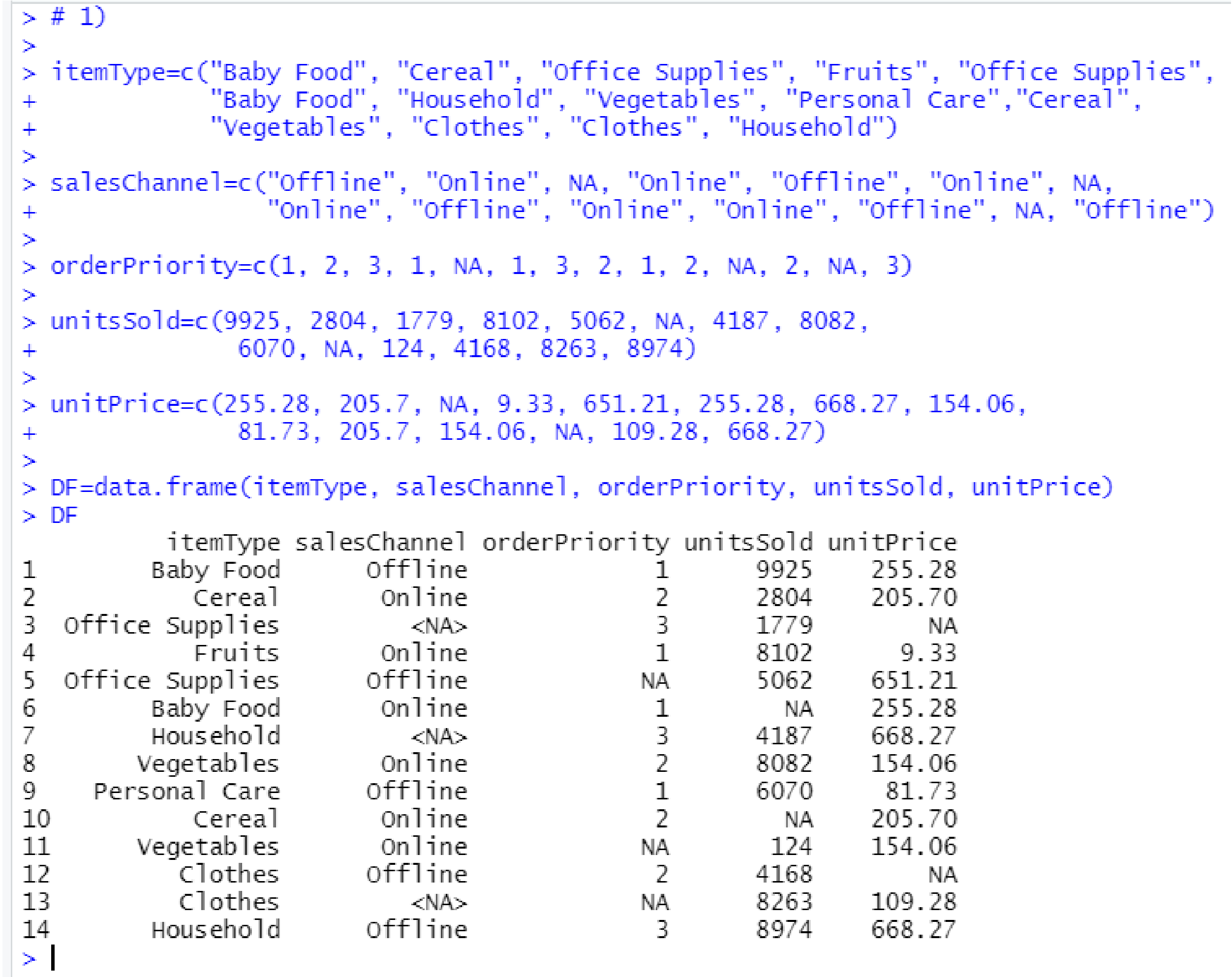
iv) Display only the female having MA degree from D4.



```
E:/VITAP/19BCE7048/Semester_4/Foundation for Data Analytics/LAB/LAB6/
> display = subset(D4,D4$degree=='MA' & D4$gender=='female')
> display
   gender age degree
10 female  35     MA
5  female  48     MA
> 
```

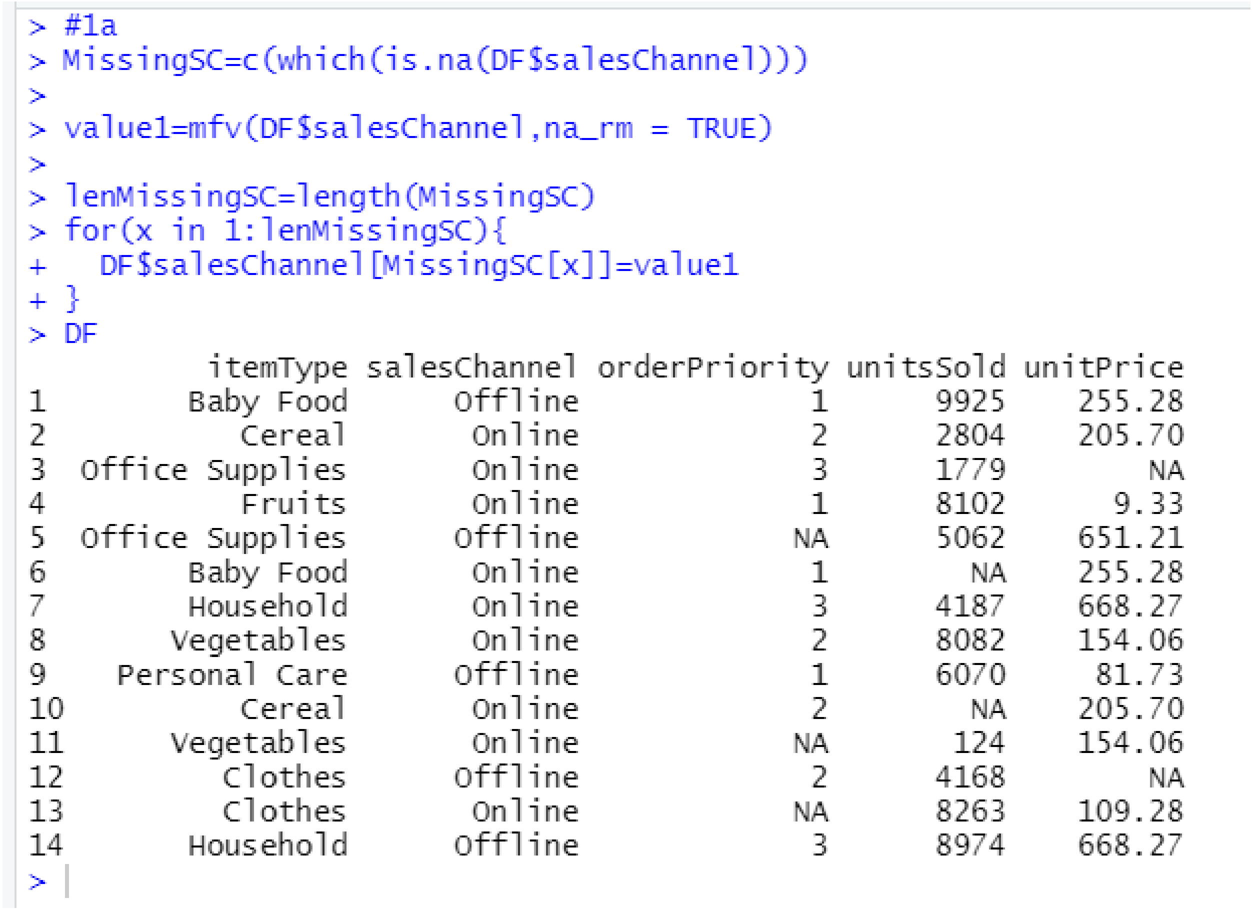
Assignment 7 : Date: 29/03/2021

1. Fill the missing value of the given table :



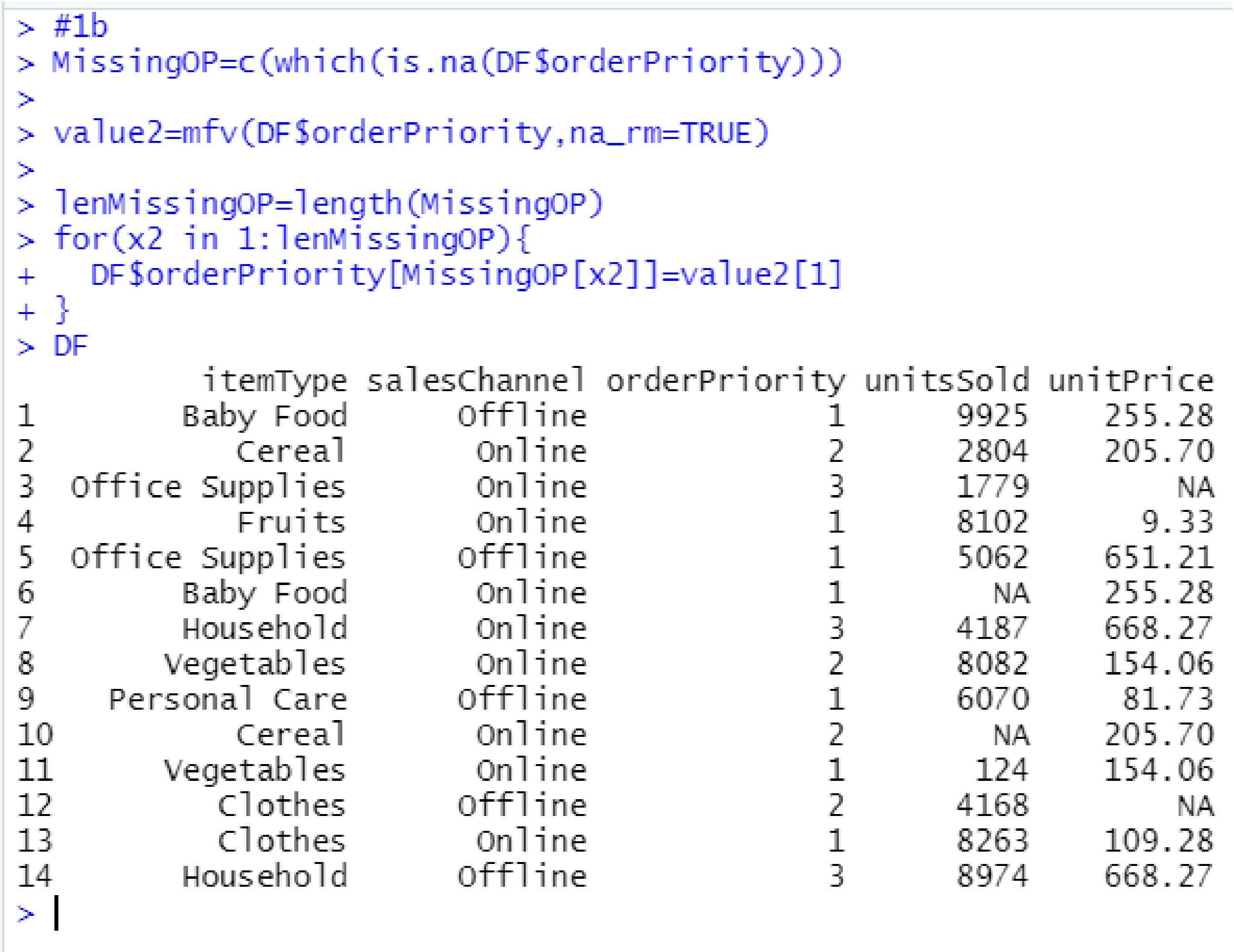
```
> # 1)
> 
> itemType=c("Baby Food", "Cereal", "Office Supplies", "Fruits", "Office Supplies",
+            "Baby Food", "Household", "Vegetables", "Personal Care","Cereal",
+            "Vegetables", "Clothes", "Clothes", "Household")
> 
> salesChannel=c("Offline", "Online", NA, "Online", "Offline", "Online", NA,
+                "Online", "Offline", "Online", "Online", "Offline", NA, "Offline")
> 
> orderPriority=c(1, 2, 3, 1, NA, 1, 3, 2, 1, 2, NA, 2, NA, 3)
> 
> unitsSold=c(9925, 2804, 1779, 8102, 5062, NA, 4187, 8082,
+              6070, NA, 124, 4168, 8263, 8974)
> 
> unitPrice=c(255.28, 205.7, NA, 9.33, 651.21, 255.28, 668.27, 154.06,
+              81.73, 205.7, 154.06, NA, 109.28, 668.27)
> 
> DF=data.frame(itemType, salesChannel, orderPriority, unitsSold, unitPrice)
> DF
          itemType salesChannel orderPriority unitsSold unitPrice
1        Baby Food      Offline             1      9925    255.28
2           Cereal       Online             2      2804    205.70
3  Office Supplies         <NA>             3      1779        NA
4           Fruits       Online             1      8102      9.33
5  Office Supplies      Offline            NA      5062    651.21
6        Baby Food       Online             1        NA    255.28
7        Household         <NA>             3      4187    668.27
8       Vegetables       Online             2      8082    154.06
9    Personal Care      Offline             1      6070     81.73
10          Cereal       Online             2        NA    205.70
11      Vegetables       Online            NA       124    154.06
12         Clothes      Offline             2      4168        NA
13         Clothes         <NA>            NA      8263    109.28
14       Household      Offline             3      8974    668.27
> 
```

a) Fill Sales Channel by mode by finding the highest frequency of the entry.



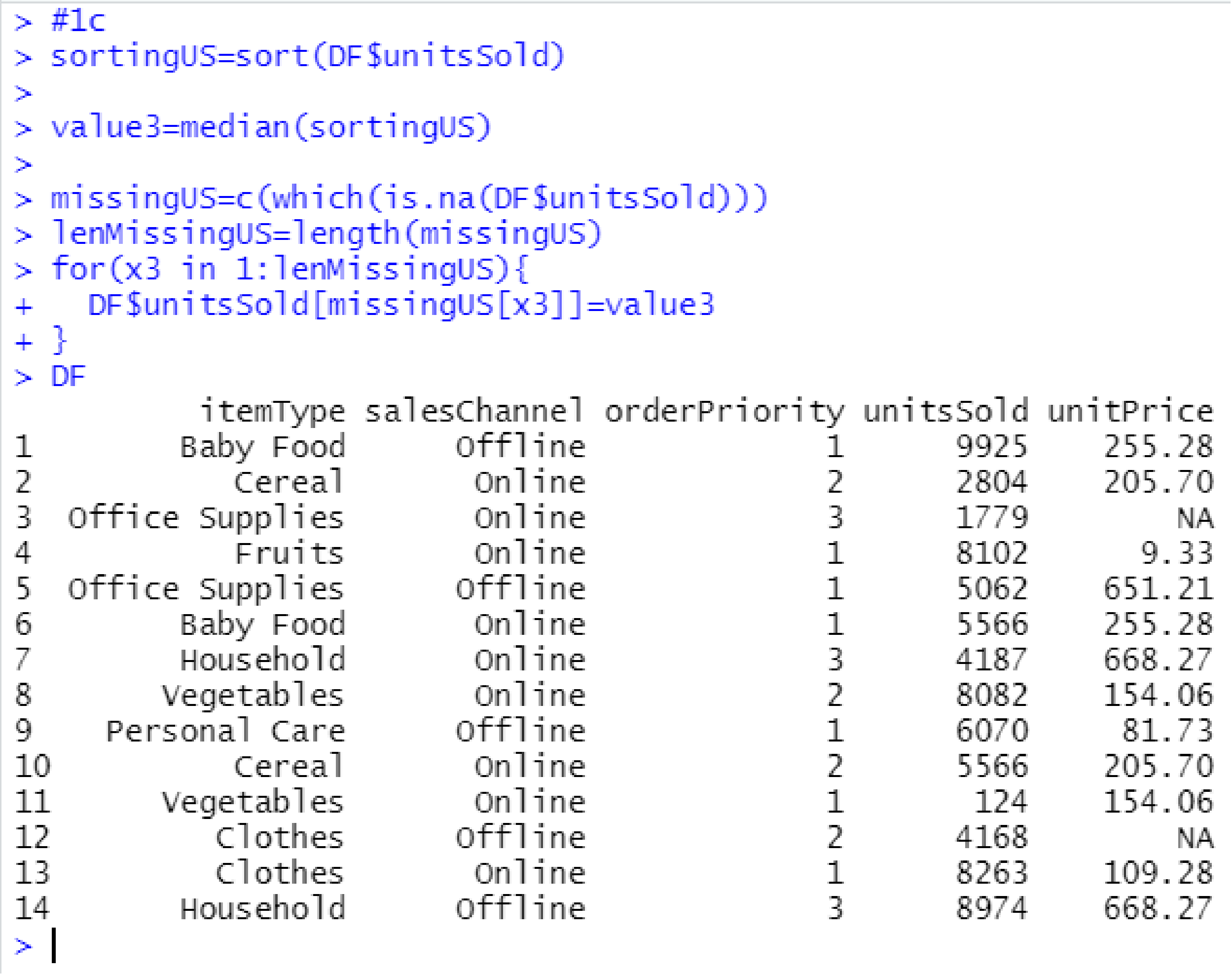
```
> #1a
> MissingSC=c(which(is.na(DF$salesChannel)))
> 
> value1=mfv(DF$salesChannel,na_rm = TRUE)
> 
> lenMissingSC=length(MissingSC)
> for(x in 1:lenMissingSC){
+   DF$salesChannel[MissingSC[x]]=value1
+ }
> DF
          itemType salesChannel orderPriority unitsSold unitPrice
1        Baby Food      Offline             1      9925    255.28
2           Cereal       Online             2      2804    205.70
3  Office Supplies       Online             3      1779        NA
4           Fruits       Online             1      8102      9.33
5  Office Supplies      Offline            NA      5062    651.21
6        Baby Food       Online             1        NA    255.28
7        Household       Online             3      4187    668.27
8       Vegetables       Online             2      8082    154.06
9    Personal Care      Offline             1      6070     81.73
10          Cereal       Online             2        NA    205.70
11      Vegetables       Online            NA       124    154.06
12         Clothes      Offline             2      4168        NA
13         Clothes       Online            NA      8263    109.28
14       Household      Offline             3      8974    668.27
> 
```

b) Fill Order Priority by mode by finding the highest frequency of the entry.



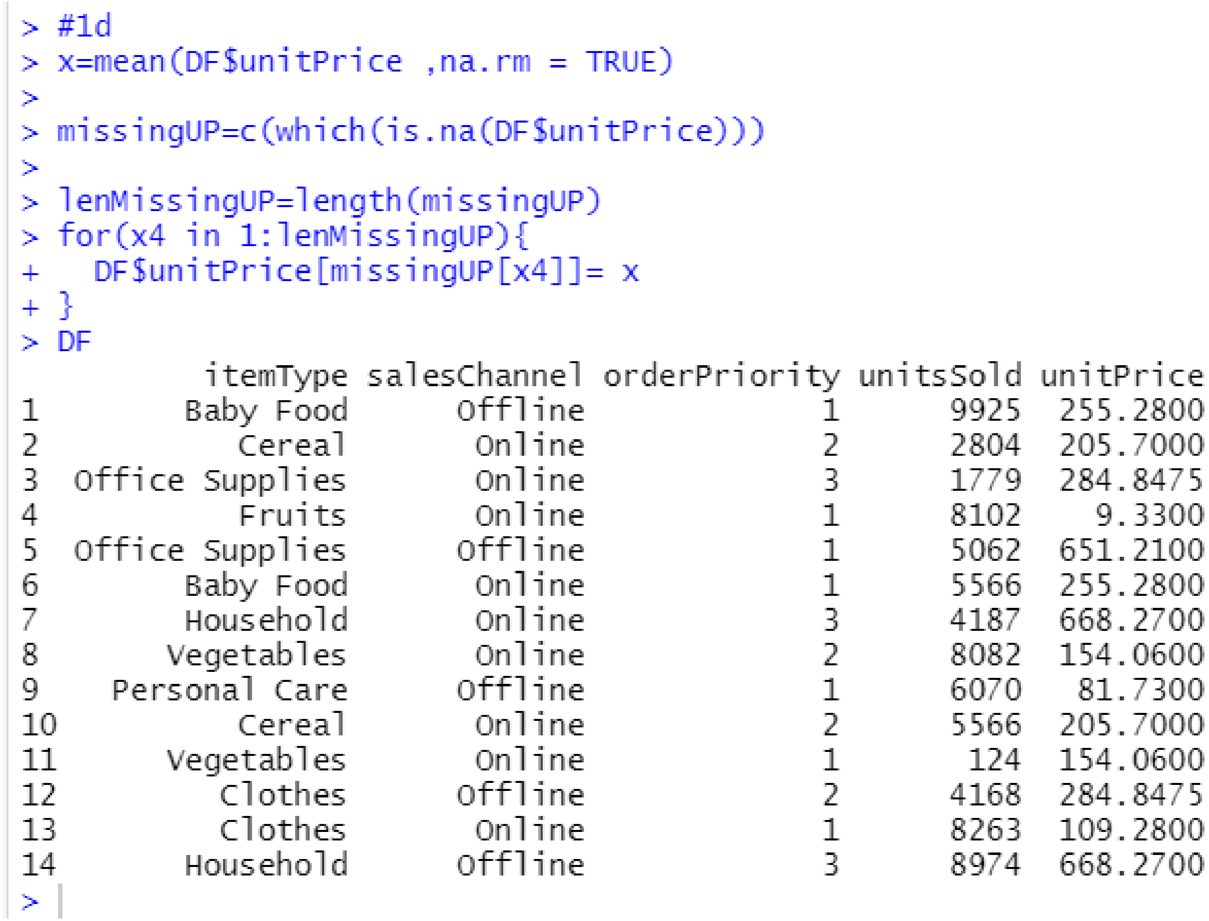
```
> #1b
> MissingOP=c(which(is.na(DF$orderPriority)))
> 
> value2=mfv(DF$orderPriority,na_rm=TRUE)
> 
> lenMissingOP=length(MissingOP)
> for(x2 in 1:lenMissingOP){
+   DF$orderPriority[MissingOP[x2]]=value2[1]
+ }
> DF
          itemType salesChannel orderPriority unitsSold unitPrice
1       Baby Food      Offline             1      9925    255.28
2          Cereal       Online             2      2804    205.70
3  Office Supplies      Online             3      1779        NA
4          Fruits       Online             1      8102      9.33
5  Office Supplies     Offline             1      5062    651.21
6       Baby Food       Online             1        NA    255.28
7       Household       Online             3      4187    668.27
8      Vegetables       Online             2      8082    154.06
9    Personal Care     Offline             1      6070     81.73
10         Cereal       Online             2        NA    205.70
11     Vegetables       Online             1       124    154.06
12        Clothes      Offline             2      4168        NA
13        Clothes       Online             1      8263    109.28
14      Household      Offline             3      8974    668.27
> 
```

c) Fill Units Sold by median. First sort it in ascending order and then find the median.



```
> #1c
> sortingUS=sort(DF$unitsSold)
>
> value3=median(sortingUS)
>
> missingUS=c(which(is.na(DF$unitsSold)))
> lenMissingUS=length(missingUS)
> for(x3 in 1:lenMissingUS){
+   DF$unitsSold[missingUS[x3]]=value3
+ }
> DF
          itemType salesChannel orderPriority unitsSold unitPrice
1        Baby Food      Offline             1      9925    255.28
2           Cereal       Online             2      2804    205.70
3  Office Supplies       Online             3      1779        NA
4           Fruits       Online             1      8102      9.33
5  Office Supplies      Offline             1      5062    651.21
6        Baby Food       Online             1      5566    255.28
7        Household       Online             3      4187    668.27
8       Vegetables       Online             2      8082    154.06
9    Personal Care      Offline             1      6070     81.73
10          Cereal       Online             2      5566    205.70
11      Vegetables       Online             1       124    154.06
12         Clothes      Offline             2      4168        NA
13         Clothes       Online             1      8263    109.28
14       Household      Offline             3      8974    668.27
> |
```

d) Fill Unit Price by mean.



```
> #1d
> x=mean(DF$unitPrice ,na.rm = TRUE)
> 
> missingUP=c(which(is.na(DF$unitPrice)))
> 
> lenMissingUP=length(missingUP)
> for(x4 in 1:lenMissingUP){
+   DF$unitPrice[missingUP[x4]]= x
+ }
> DF
          itemType salesChannel orderPriority unitsSold unitPrice
1        Baby Food      Offline             1      9925  255.2800
2           Cereal       Online             2      2804  205.7000
3  Office Supplies       Online             3      1779  284.8475
4           Fruits       Online             1      8102    9.3300
5  Office Supplies      Offline             1      5062  651.2100
6        Baby Food       Online             1      5566  255.2800
7        Household       Online             3      4187  668.2700
8       Vegetables       Online             2      8082  154.0600
9    Personal Care      Offline             1      6070   81.7300
10          Cereal       Online             2      5566  205.7000
11      Vegetables       Online             1       124  154.0600
12         Clothes      Offline             2      4168  284.8475
13         Clothes       Online             1      8263  109.2800
14       Household      Offline             3      8974  668.2700
> 
```

2. (a) Write a R program to print the pattern using the user defined function Patt ( ) given below and take a number of rows as input from the user.

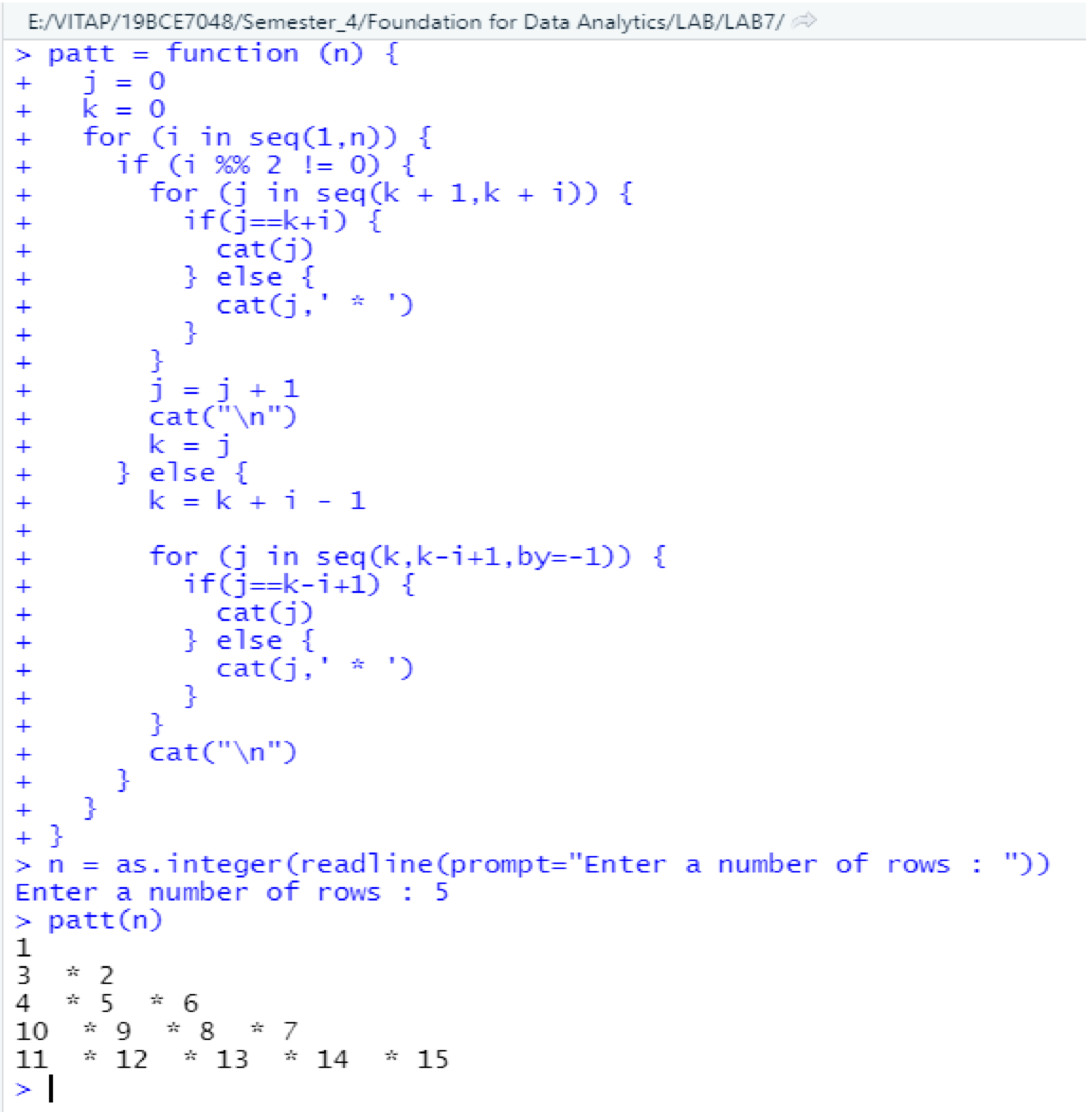
1

3*2

4*5*6

10*9*8*7

11*12*13*14*15

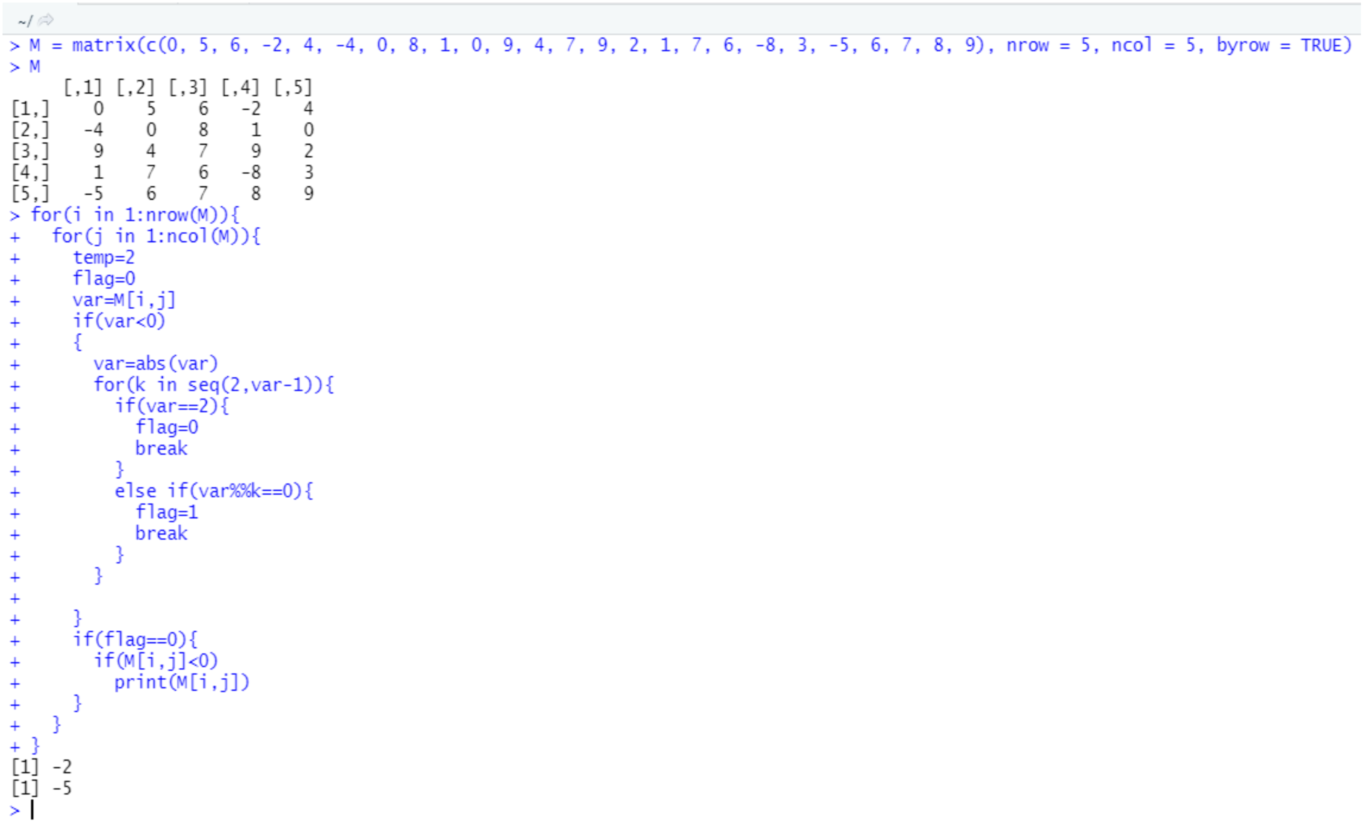


E:/VITAP/19BCE7048/Semester_4/Foundation for Data Analytics/LAB/LAB7/

```
> patt = function (n) {
+   j = 0
+   k = 0
+   for (i in seq(1,n)) {
+     if (i %% 2 != 0) {
+       for (j in seq(k + 1,k + i)) {
+         if(j==k+i) {
+           cat(j)
+         } else {
+           cat(j,' * ')
+         }
+       }
+       j = j + 1
+       cat("\n")
+       k = j
+     } else {
+       k = k + i - 1
+ 
+       for (j in seq(k,k-i+1,by=-1)) {
+         if(j==k-i+1) {
+           cat(j)
+         } else {
+           cat(j,' * ')
+         }
+       }
+       cat("\n")
+     }
+   }
+ }
> n = as.integer(readline(prompt="Enter a number of rows : "))
Enter a number of rows : 5
> patt(n)
1
3   * 2
4   * 5   * 6
10   * 9   * 8   * 7
11   * 12   * 13   * 14   * 15
> 
```

(b) Write the R Program to create a 5X5 matrix and display only the negative number which is the prime number present in the above matrix.

0	5	6	-2	4
-4	0	8	1	0
9	4	7	9	2
1	7	6	-8	3
-5	6	7	8	9

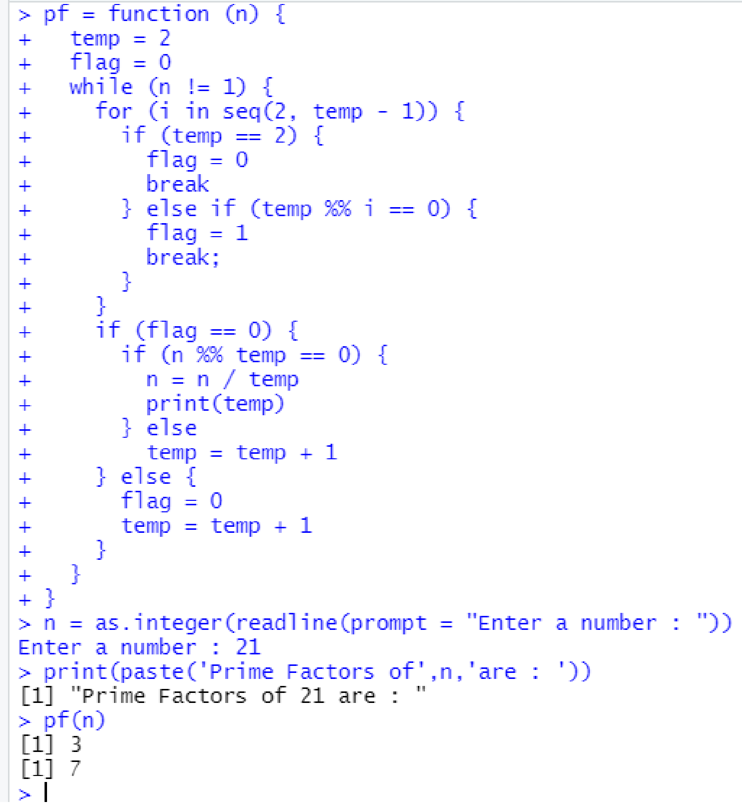


```
> M = matrix(c(0, 5, 6, -2, 4, -4, 0, 8, 1, 0, 9, 4, 7, 9, 2, 1, 7, 6, -8, 3, -5, 6, 7, 8, 9), nrow = 5, ncol = 5, byrow = TRUE)
> M
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    5    6   -2    4
[2,]   -4    0    8    1    0
[3,]    9    4    7    9    2
[4,]    1    7    6   -8    3
[5,]   -5    6    7    8    9
> for(i in 1:nrow(M)){
+   for(j in 1:ncol(M)){
+     temp=2
+     flag=0
+     var=M[i,j]
+     if(var<0)
+     {
+       var=abs(var)
+       for(k in seq(2,var-1)){
+         if(var==2){
+           flag=0
+           break
+         }
+         else if(var%%k==0){
+           flag=1
+           break
+         }
+       }
+
+     }
+     if(flag==0){
+       if(M[i,j]<0)
+         print(M[i,j])
+     }
+   }
+ }
[1] -2
[1] -5
> |
```

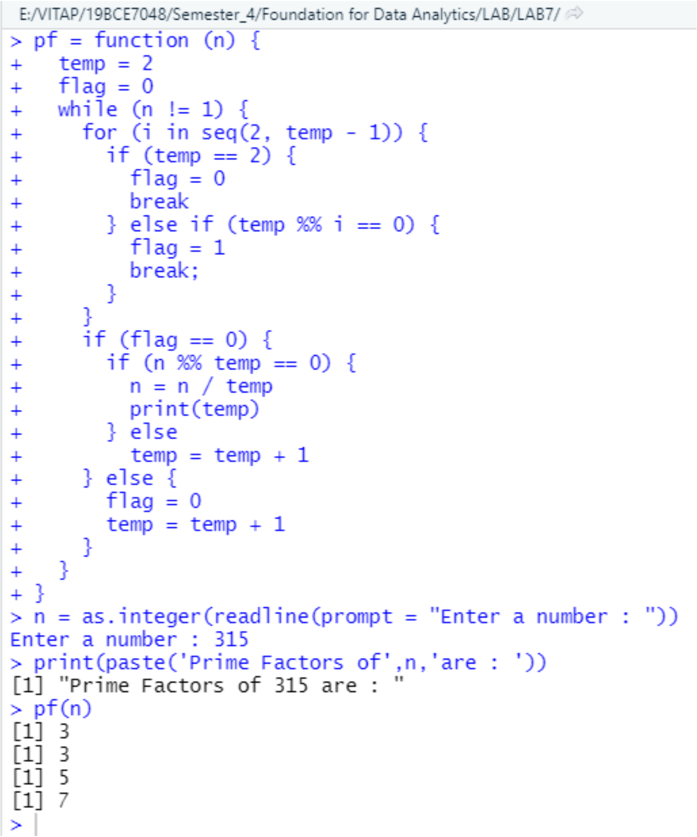
(c) Write an R program using function PF( ) to print all prime factors of n. Take n as input from the user.

input = 21 output =3,7

input=315 output=3, 3, 5, 7



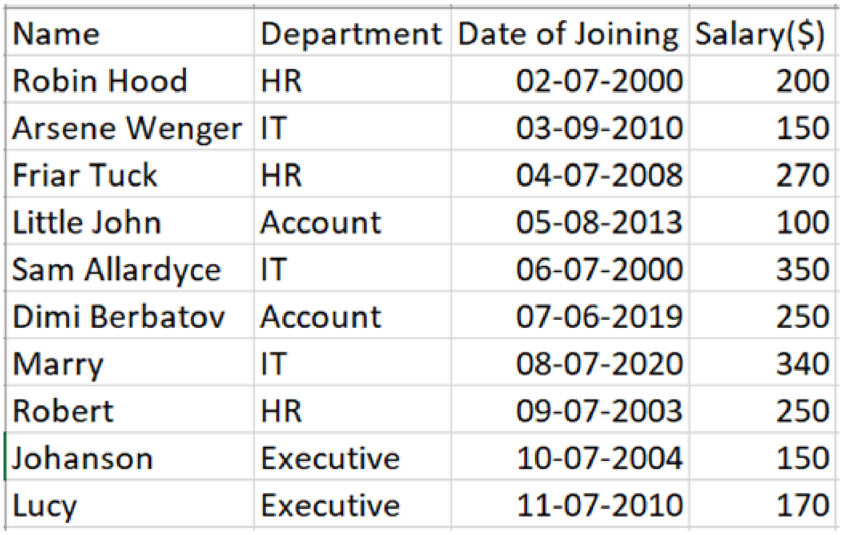
```
> pf = function (n) {
+   temp = 2
+   flag = 0
+   while (n != 1) {
+     for (i in seq(2, temp - 1)) {
+       if (temp == 2) {
+         flag = 0
+         break
+       } else if (temp %% i == 0) {
+         flag = 1
+         break;
+       }
+     }
+     if (flag == 0) {
+       if (n %% temp == 0) {
+         n = n / temp
+         print(temp)
+       } else
+         temp = temp + 1
+     } else {
+       flag = 0
+       temp = temp + 1
+     }
+   }
+ }
> n = as.integer(readline(prompt = "Enter a number : "))
Enter a number : 21
> print(paste('Prime Factors of',n,'are : '))
[1] "Prime Factors of 21 are : "
> pf(n)
[1] 3
[1] 7
> 
```



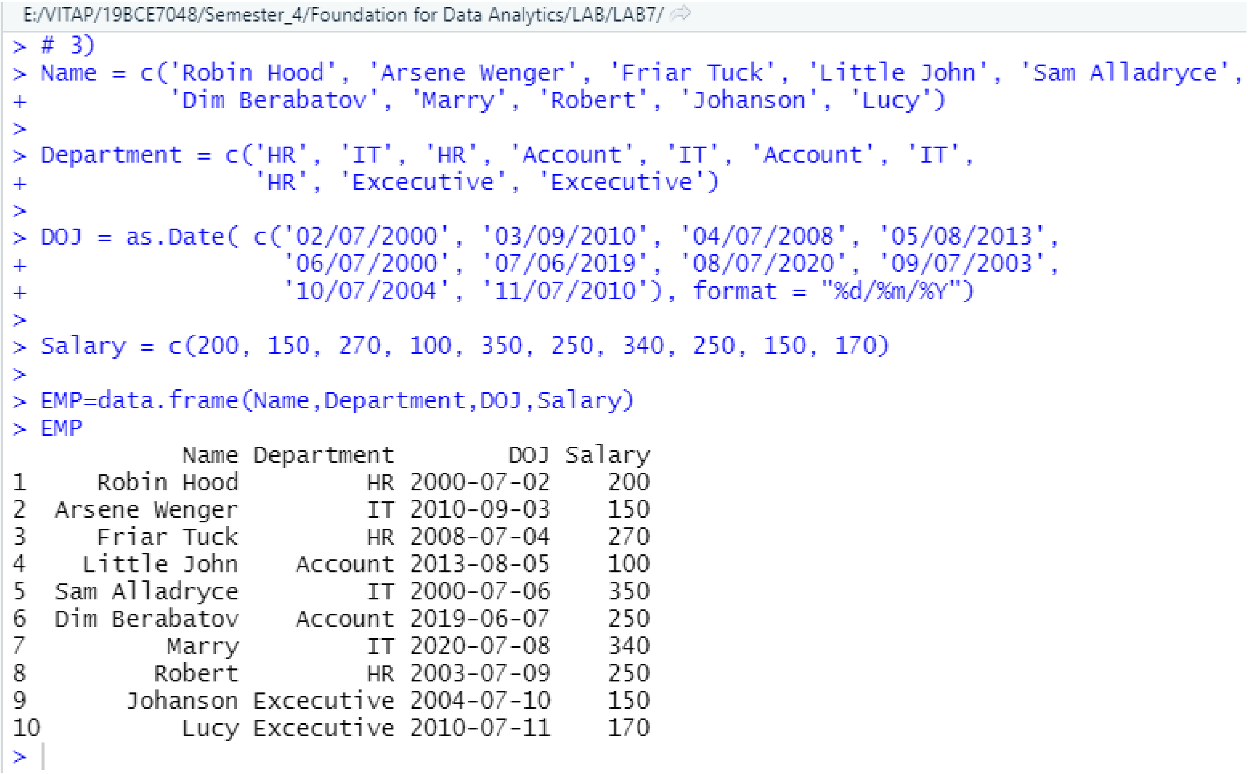
E:/VITAP/19BCE7048/Semester_4/Foundation for Data Analytics/LAB/LAB7/

```
> pf = function (n) {
+   temp = 2
+   flag = 0
+   while (n != 1) {
+     for (i in seq(2, temp - 1)) {
+       if (temp == 2) {
+         flag = 0
+         break
+       } else if (temp %% i == 0) {
+         flag = 1
+         break;
+       }
+     }
+     if (flag == 0) {
+       if (n %% temp == 0) {
+         n = n / temp
+         print(temp)
+       } else
+         temp = temp + 1
+     } else {
+       flag = 0
+       temp = temp + 1
+     }
+   }
+ }
> n = as.integer(readline(prompt = "Enter a number : "))
Enter a number : 315
> print(paste('Prime Factors of',n,'are : '))
[1] "Prime Factors of 315 are : "
> pf(n)
[1] 3
[1] 3
[1] 5
[1] 7
> 
```

3. Create a Data frame EMP as given below



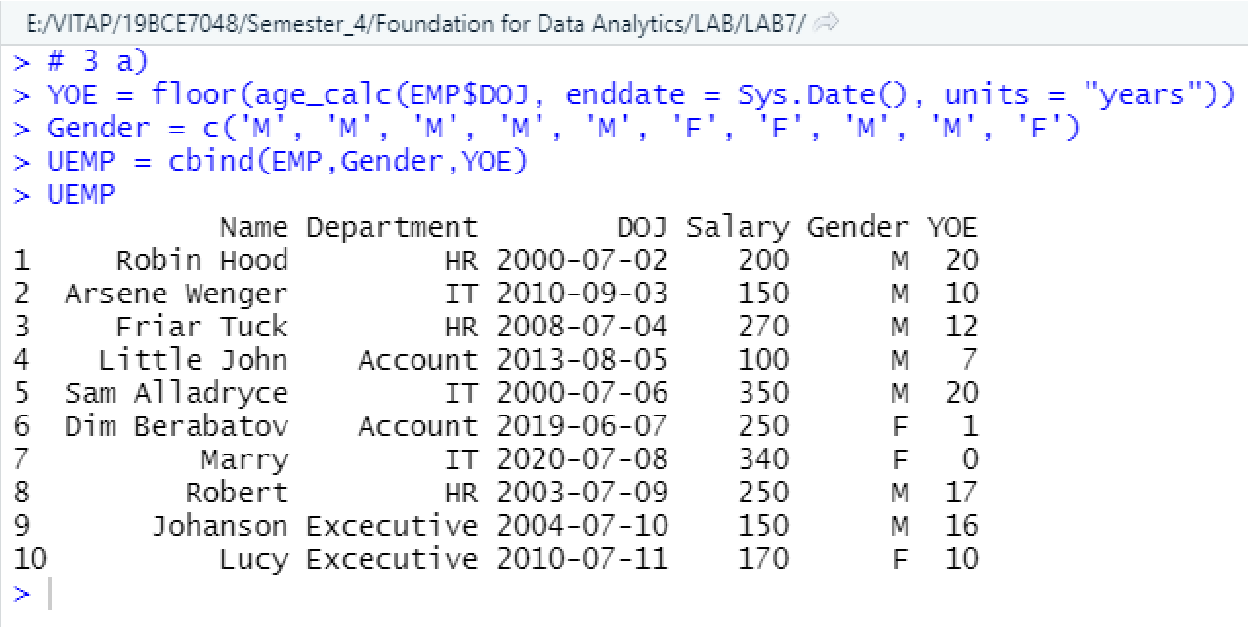
Name	Department	Date of Joining	Salary($)
Robin Hood	HR	02-07-2000	200
Arsene Wenger	IT	03-09-2010	150
Friar Tuck	HR	04-07-2008	270
Little John	Account	05-08-2013	100
Sam Allardyce	IT	06-07-2000	350
Dimi Berbatov	Account	07-06-2019	250
Marry	IT	08-07-2020	340
Robert	HR	09-07-2003	250
Johanson	Executive	10-07-2004	150
Lucy	Executive	11-07-2010	170



E:/VITAP/19BCE7048/Semester_4/Foundation for Data Analytics/LAB/LAB7/

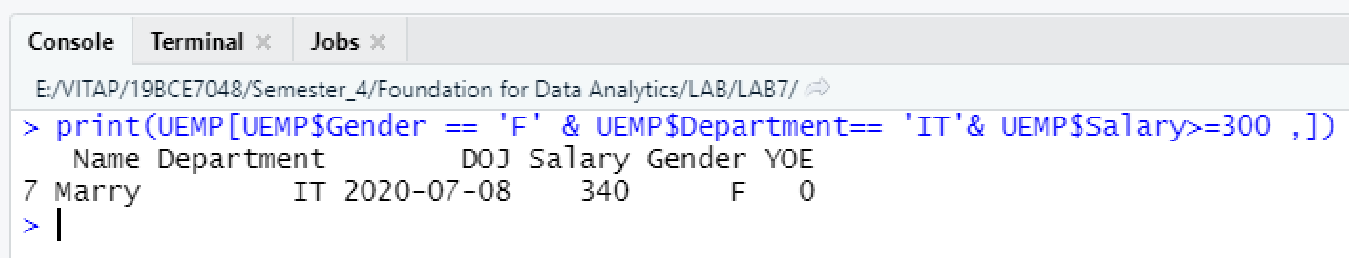
```
> # 3)
> Name = c('Robin Hood', 'Arsene Wenger', 'Friar Tuck', 'Little John', 'Sam Alladryce',
+          'Dim Berabatov', 'Marry', 'Robert', 'Johanson', 'Lucy')
>
> Department = c('HR', 'IT', 'HR', 'Account', 'IT', 'Account', 'IT',
+                'HR', 'Excecutive', 'Excecutive')
>
> DOJ = as.Date( c('02/07/2000', '03/09/2010', '04/07/2008', '05/08/2013',
+                  '06/07/2000', '07/06/2019', '08/07/2020', '09/07/2003',
+                  '10/07/2004', '11/07/2010'), format = "%d/%m/%Y")
>
> Salary = c(200, 150, 270, 100, 350, 250, 340, 250, 150, 170)
>
> EMP=data.frame(Name,Department,DOJ,Salary)
> EMP
            Name Department        DOJ Salary
1     Robin Hood         HR 2000-07-02    200
2  Arsene Wenger         IT 2010-09-03    150
3     Friar Tuck         HR 2008-07-04    270
4    Little John    Account 2013-08-05    100
5  Sam Alladryce         IT 2000-07-06    350
6  Dim Berabatov    Account 2019-06-07    250
7          Marry         IT 2020-07-08    340
8         Robert         HR 2003-07-09    250
9       Johanson Excecutive 2004-07-10    150
10          Lucy Excecutive 2010-07-11    170
> 
```

(a) Calculate the Year of experience with respect to current date and append to the data frame as Experience column, add Gender column, and name the data frame as UEMP.



```
E:/VITAP/19BCE7048/Semester_4/Foundation for Data Analytics/LAB/LAB7/
> # 3 a)
> YOE = floor(age_calc(EMP$DOJ, enddate = Sys.Date(), units = "years"))
> Gender = c('M', 'M', 'M', 'M', 'M', 'F', 'F', 'M', 'M', 'F')
> UEMP = cbind(EMP,Gender,YOE)
> UEMP
             Name Department        DOJ Salary Gender YOE
1      Robin Hood         HR 2000-07-02    200      M  20
2  Arsene Wenger          IT 2010-09-03    150      M  10
3      Friar Tuck         HR 2008-07-04    270      M  12
4     Little John    Account 2013-08-05    100      M   7
5  Sam Alladryce          IT 2000-07-06    350      M  20
6  Dim Berabatov     Account 2019-06-07    250      F   1
7           Marry         IT 2020-07-08    340      F   0
8          Robert         HR 2003-07-09    250      M  17
9        Johanson Excecutive 2004-07-10    150      M  16
10           Lucy Excecutive 2010-07-11    170      F  10
> 
```

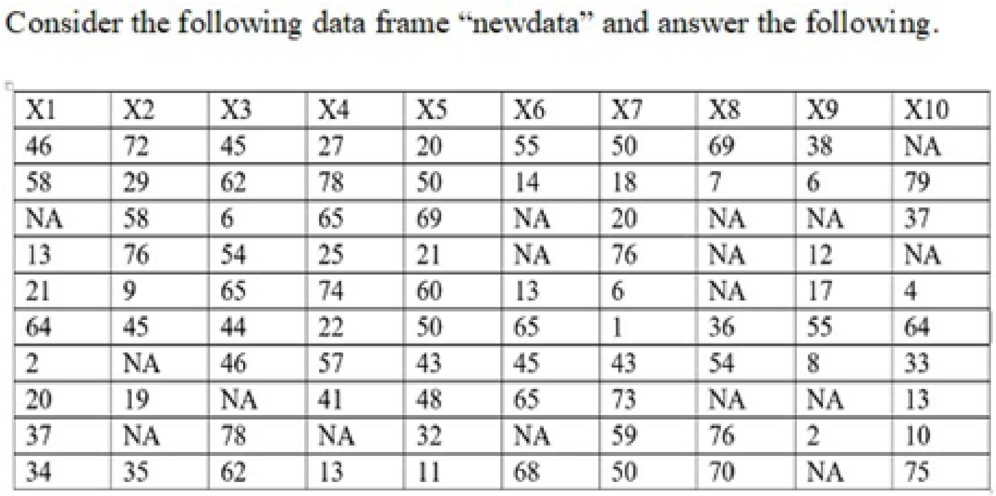
(b) Display the data where the female is from the IT department who got more than equal to 300$ salary from the UEMP.



```
E:/VITAP/19BCE7048/Semester_4/Foundation for Data Analytics/LAB/LAB7/
> print(UEMP[UEMP$Gender == 'F' & UEMP$Department== 'IT'& UEMP$Salary>=300 ,])
   Name Department        DOJ Salary Gender YOE
7 Marry         IT 2020-07-08    340      F   0
> |
```

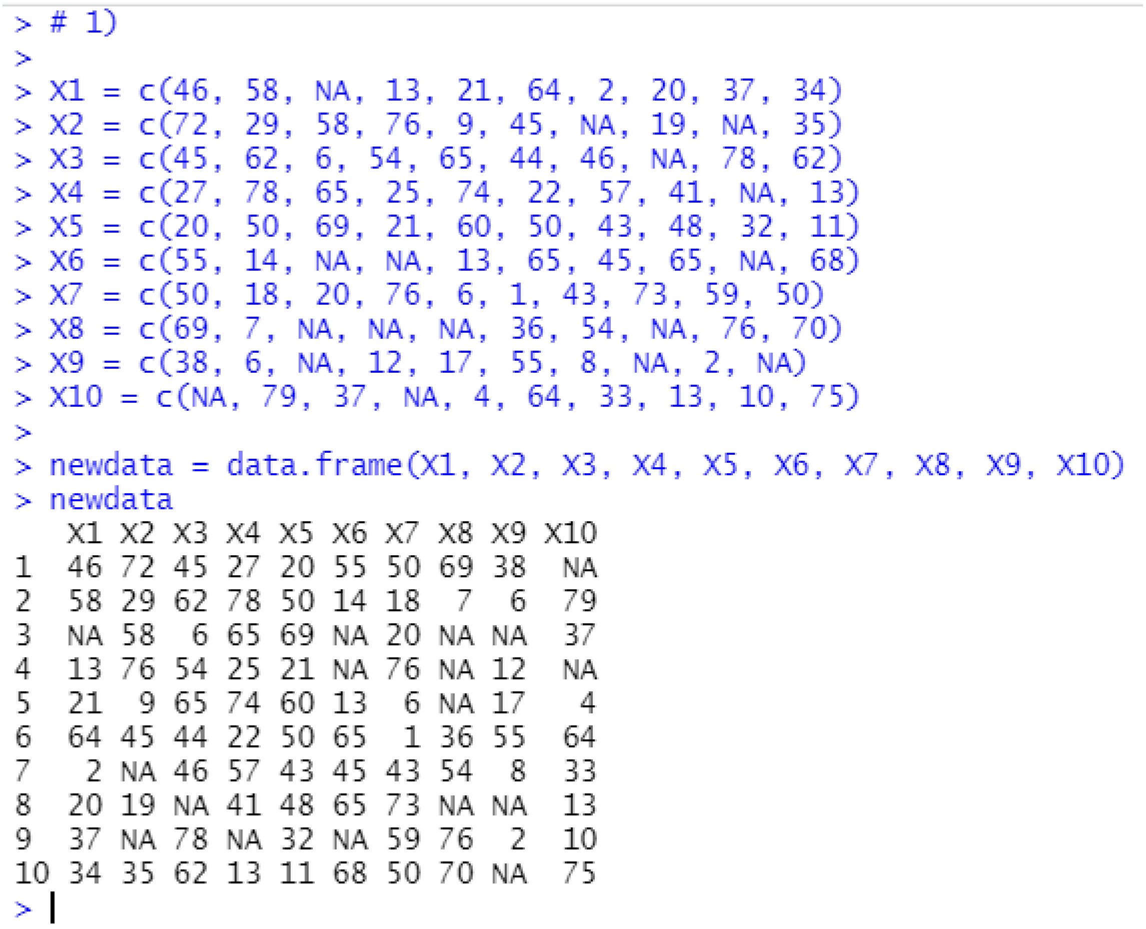
Assignment 8 : **Date: 06/04/2021**

1.



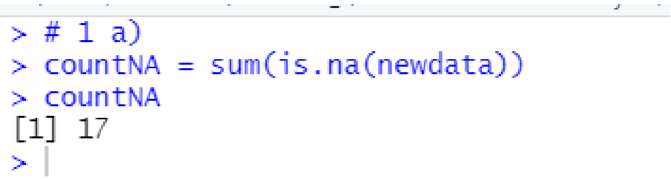
Consider the following data frame “newdata” and answer the following.

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
46	72	45	27	20	55	50	69	38	NA
58	29	62	78	50	14	18	7	6	79
NA	58	6	65	69	NA	20	NA	NA	37
13	76	54	25	21	NA	76	NA	12	NA
21	9	65	74	60	13	6	NA	17	4
64	45	44	22	50	65	1	36	55	64
2	NA	46	57	43	45	43	54	8	33
20	19	NA	41	48	65	73	NA	NA	13
37	NA	78	NA	32	NA	59	76	2	10
34	35	62	13	11	68	50	70	NA	75



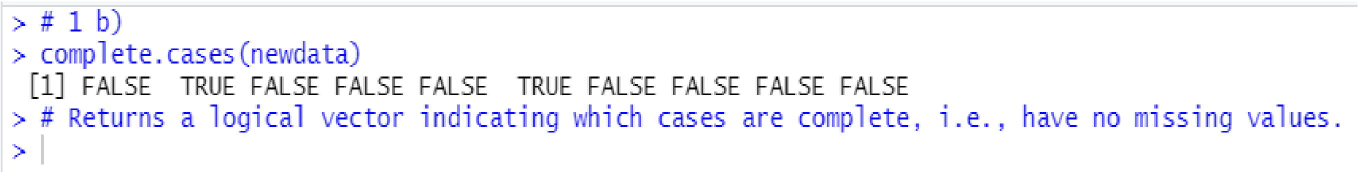
```
> # 1)
>
> X1 = c(46, 58, NA, 13, 21, 64, 2, 20, 37, 34)
> X2 = c(72, 29, 58, 76, 9, 45, NA, 19, NA, 35)
> X3 = c(45, 62, 6, 54, 65, 44, 46, NA, 78, 62)
> X4 = c(27, 78, 65, 25, 74, 22, 57, 41, NA, 13)
> X5 = c(20, 50, 69, 21, 60, 50, 43, 48, 32, 11)
> X6 = c(55, 14, NA, NA, 13, 65, 45, 65, NA, 68)
> X7 = c(50, 18, 20, 76, 6, 1, 43, 73, 59, 50)
> X8 = c(69, 7, NA, NA, NA, 36, 54, NA, 76, 70)
> X9 = c(38, 6, NA, 12, 17, 55, 8, NA, 2, NA)
> X10 = c(NA, 79, 37, NA, 4, 64, 33, 13, 10, 75)
>
> newdata = data.frame(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10)
> newdata
   X1 X2 X3 X4 X5 X6 X7 X8 X9 X10
1  46 72 45 27 20 55 50 69 38  NA
2  58 29 62 78 50 14 18  7  6  79
3  NA 58  6 65 69 NA 20 NA NA  37
4  13 76 54 25 21 NA 76 NA 12  NA
5  21  9 65 74 60 13  6 NA 17   4
6  64 45 44 22 50 65  1 36 55  64
7   2 NA 46 57 43 45 43 54  8  33
8  20 19 NA 41 48 65 73 NA NA  13
9  37 NA 78 NA 32 NA 59 76  2  10
10 34 35 62 13 11 68 50 70 NA  75
> |
```

a) Write a command to print total no. of missing values.



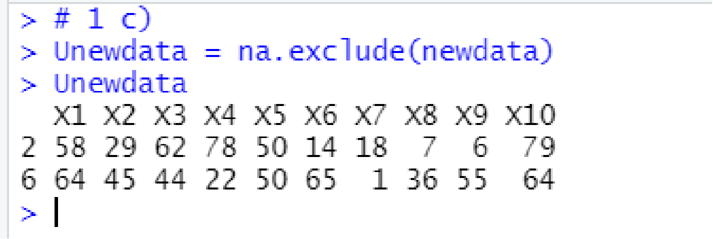
```
> # 1 a)
> countNA = sum(is.na(newdata))
> countNA
[1] 17
> 
```

b) What is the output of the function complete.cases(newdata)



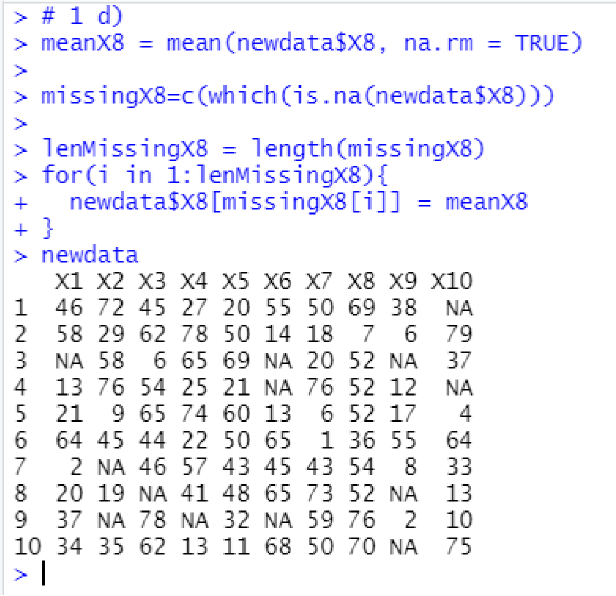
```
> # 1 b)
> complete.cases(newdata)
 [1] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
> # Returns a logical vector indicating which cases are complete, i.e., have no missing values.
> 
```

c) How to drop out any rows with missing values.



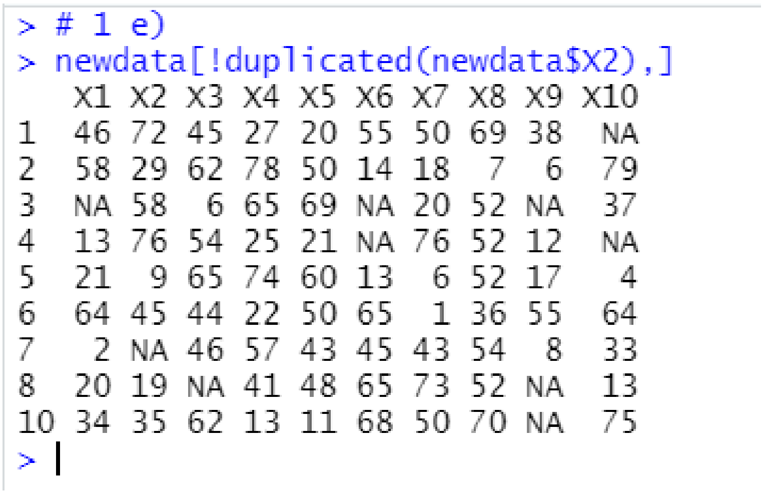
```
> # 1 c)
> Unewdata = na.exclude(newdata)
> Unewdata
  X1 X2 X3 X4 X5 X6 X7 X8 X9 X10
2 58 29 62 78 50 14 18  7  6  79
6 64 45 44 22 50 65  1 36 55  64
> 
```

d) Write a command to replace missing values in the column X8 with the mean of remaining X8 values.



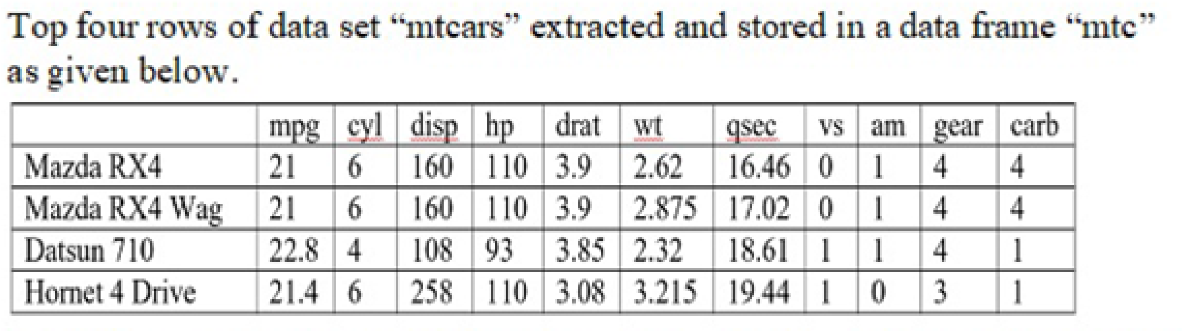
```
> # 1 d)
> meanX8 = mean(newdata$X8, na.rm = TRUE)
> 
> missingX8=c(which(is.na(newdata$X8)))
> 
> lenMissingX8 = length(missingX8)
> for(i in 1:lenMissingX8){
+   newdata$X8[missingX8[i]] = meanX8
+ }
> newdata
   X1 X2 X3 X4 X5 X6 X7 X8 X9 X10
1  46 72 45 27 20 55 50 69 38  NA
2  58 29 62 78 50 14 18  7  6  79
3  NA 58  6 65 69 NA 20 52 NA  37
4  13 76 54 25 21 NA 76 52 12  NA
5  21  9 65 74 60 13  6 52 17   4
6  64 45 44 22 50 65  1 36 55  64
7   2 NA 46 57 43 45 43 54  8  33
8  20 19 NA 41 48 65 73 52 NA  13
9  37 NA 78 NA 32 NA 59 76  2  10
10 34 35 62 13 11 68 50 70 NA  75
> 
```

e) How to remove duplicates records based on X2 values.



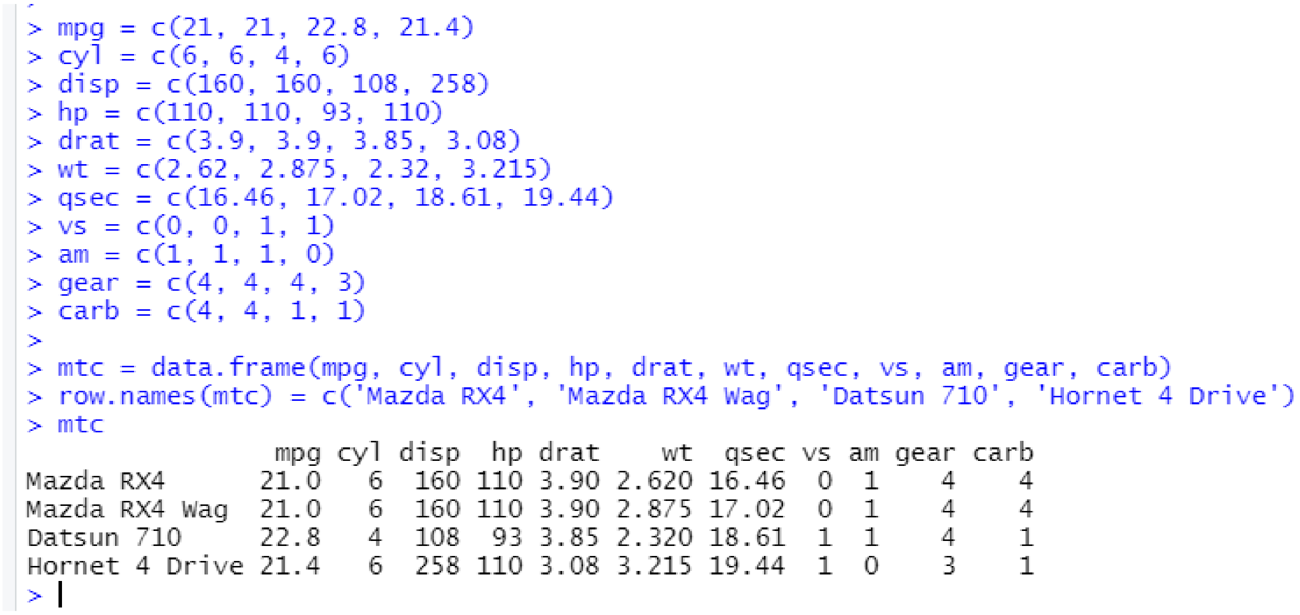
```
> # 1 e)
> newdata[!duplicated(newdata$X2),]
   X1 X2 X3 X4 X5 X6 X7 X8 X9 X10
1  46 72 45 27 20 55 50 69 38  NA
2  58 29 62 78 50 14 18  7  6  79
3  NA 58  6 65 69 NA 20 52 NA  37
4  13 76 54 25 21 NA 76 52 12  NA
5  21  9 65 74 60 13  6 52 17   4
6  64 45 44 22 50 65  1 36 55  64
7   2 NA 46 57 43 45 43 54  8  33
8  20 19 NA 41 48 65 73 52 NA  13
10 34 35 62 13 11 68 50 70 NA  75
> 
```

2.



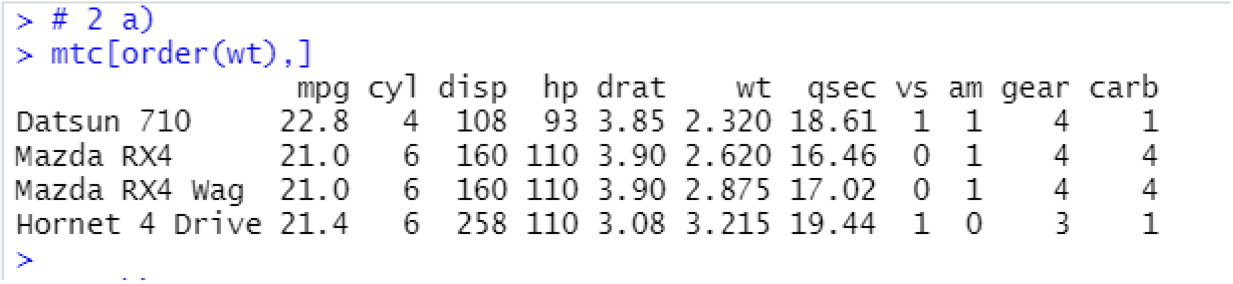
Top four rows of data set “mtcars” extracted and stored in a data frame “mtc” as given below.

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1



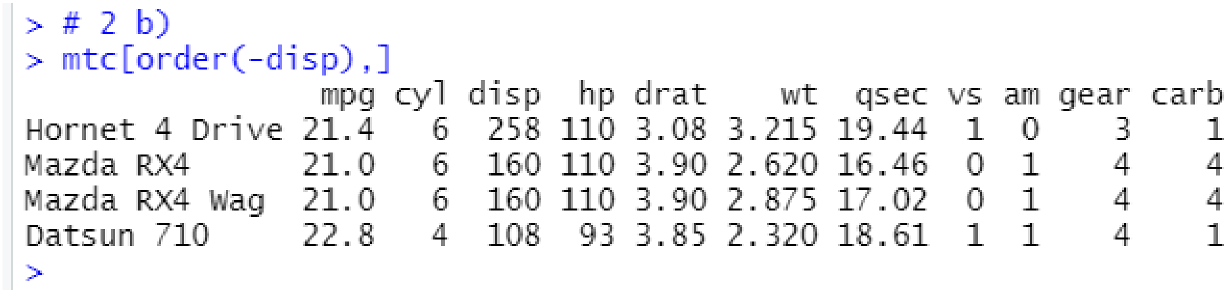
```
> mpg = c(21, 21, 22.8, 21.4)
> cyl = c(6, 6, 4, 6)
> disp = c(160, 160, 108, 258)
> hp = c(110, 110, 93, 110)
> drat = c(3.9, 3.9, 3.85, 3.08)
> wt = c(2.62, 2.875, 2.32, 3.215)
> qsec = c(16.46, 17.02, 18.61, 19.44)
> vs = c(0, 0, 1, 1)
> am = c(1, 1, 1, 0)
> gear = c(4, 4, 4, 3)
> carb = c(4, 4, 1, 1)
> 
> mtc = data.frame(mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb)
> row.names(mtc) = c('Mazda RX4', 'Mazda RX4 Wag', 'Datsun 710', 'Hornet 4 Drive')
> mtc
                mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710     22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
> 
```

a) Write the command to sort the above observations in the increasing order of the attribute "'wt" and write down the output.



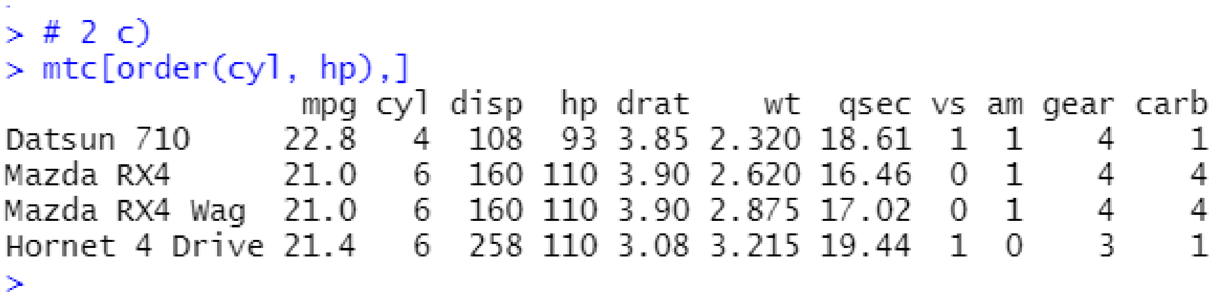
```
> # 2 a)
> mtc[order(wt),]
                mpg cyl disp  hp drat    wt  qsec vs am gear carb
Datsun 710     22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
>
```

b) Write the command to sort the above observations in the decreasing order of the attribute "disp" and write down the output.



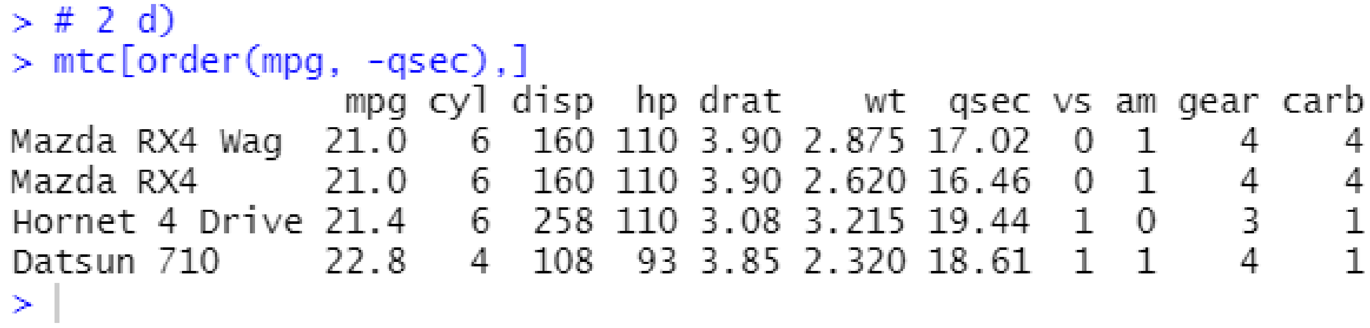
```
> # 2 b)
> mtc[order(-disp),]
                mpg cyl disp  hp drat    wt  qsec vs am gear carb
Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710     22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
>
```

c) Write the command to sort the above observations in the increasing order of the attributes both "cyl" and "hp" and write down the output.



```
> # 2 c)
> mtc[order(cyl, hp),]
                mpg cyl disp  hp drat    wt  qsec vs am gear carb
Datsun 710     22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
>
```

d) Write the command to sort the above observations in the increasing order of the attribute “mpg” and decreasing order of the attribute “qsec” (both the conditions at the same time) and write down the output.



```
> # 2 d)
> mtc[order(mpg, -qsec),]
                mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Datsun 710     22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
> 
```