

Sub : Foundations for Data Analytics

Name: Gudi Varaprasad

Reg.No: 19BCE7048

Slot No: L33+L34

Assignment 7 :

Date: 04/04/2021

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

```

> 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.

```
Console Terminal x Jobs x
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

```
> |
```