Expectation:

- Become familiar with the common functions that are used for data manipulations
- Exploring new functions using help in R to solve the problems

This assignment is intended for practice. We shall discuss these problems (if you have any questions) in the morning session of CUTe. But please try to solve them. If you are stuck, you can post your questions in piazza or you can approach us during the weekdays. Please plan to complete this task by Wednesday. We expect you to post your solutions in Grader (the details of which will be posted to you in a couple of days) by 6:00 PM Wednesday 21st March 2018.

Review of topics covered

1. Given are the marks scored by the students in the 5 subjects in Marks.csv and when was the exam given in Exam.csv.
    a. Before reading the data into R, observe how the missing values are represented.
    b. Read the data and observe the structure of the data.
    c. Merge the two data files based on the common key.
    d. Impute the missing values appropriately

2. How many students have scored more than 60 in Subject A when the exam was given in March.

3. How many students have given exam in September

4. Create final marks and corresponding percentage for each student.

5. Create another column that has “class” of the student. If the average score greater than 75% and exam given in March, its ‘distinction”. If marks greater than 75% and exam given in September ‘dist_supplementary’. Greater than 60 in March is ‘I class’ and in September id ‘I class_supplementary’. And below 60 in March is ‘Fail’ and in September is ‘detained’.

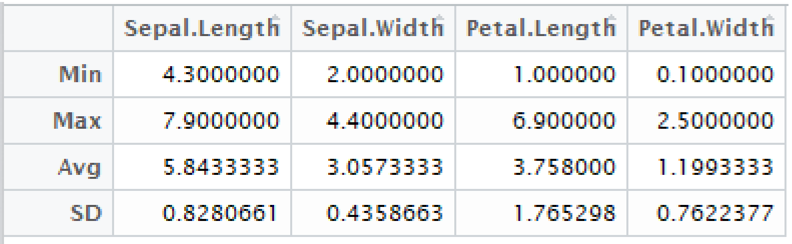
6. Give a plot to visualize average marks scored in March and September

7. Which two subjects have high correlation in March and in September separately.

8. Standardize the scores by each subject. You can use either ‘standardize’ function in vegan library or ‘scale’ function in base package.

These questions need a bit of exploration

1. Load “iris” dataset in to R. Now, write a custom function to do get the following output



	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Min	4.3000000	2.0000000	1.000000	0.1000000
Max	7.9000000	4.4000000	6.900000	2.5000000
Avg	5.8433333	3.0573333	3.758000	1.1993333
SD	0.8280661	0.4358663	1.765298	0.7622377

   a. Hint: We need to have the columns names of the iris data as columns and min, max etc as row names. Prepare a function in such a way that, all (Min, Max, Mean and SD) are computed within a function and then use this function in “apply”.
   Eg. If we want to create a function that can return square of a value.

```
F<- function(x){
            return(x^2)
}
```

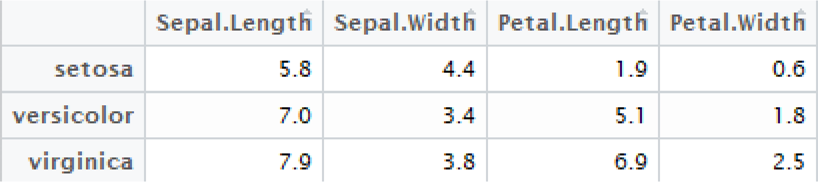
   F(2) outputs 4.
   For x (a numeric or a vector) is squared and returned.

   Lets say we want to find mean/average of a vector

```
F_average<-function(x){
                  return(sum(x)/length(x))
}
```

   F_average(1:5) outputs 3.

2. As we have seen in the previous lab day, tapply is a form of “Group_By” in SQL. Use this tapply function and extract the highest petal length for each of the species by using a user defined function. The output looks like this

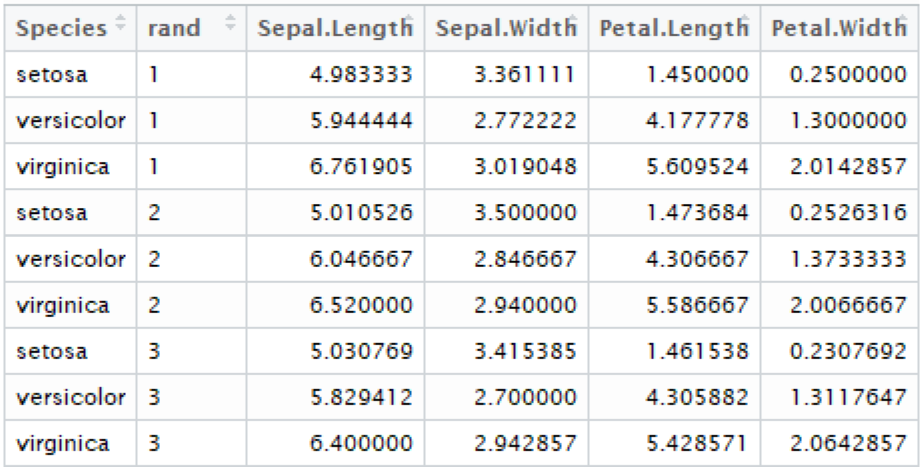


	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa	5.8	4.4	1.9	0.6
versicolor	7.0	3.4	5.1	1.8
virginica	7.9	3.8	6.9	2.5

3. Use aggregate function to extract the maximum value for each variable for each of the species. Observe that the values you got here are same as what you got in question 2.

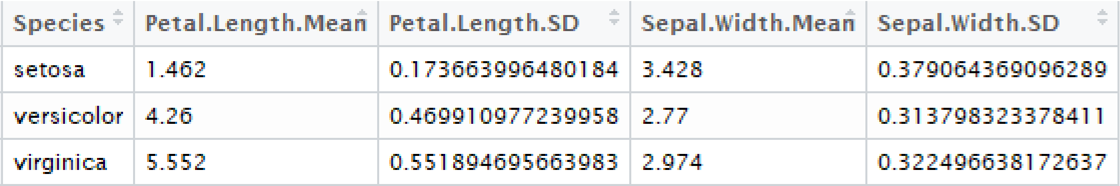
4. Create another column “Rand” to the iris data with values between 1 and 3 (integer values both 1 and 3 included) using a sample function. Convert this “rand” column into a factor. Extract the mean value for each of the variable with respect to both species and rand variables. The output you get might be similar to the one given below but need not be the same.

    a. Hint: In the help, observe the syntax for aggregate function on how to group by two different variables and two different columns by a single variable. How the “***formula”*** argument is changing



Species	rand	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa	1	4.983333	3.361111	1.450000	0.2500000
versicolor	1	5.944444	2.772222	4.177778	1.3000000
virginica	1	6.761905	3.019048	5.609524	2.0142857
setosa	2	5.010526	3.500000	1.473684	0.2526316
versicolor	2	6.046667	2.846667	4.306667	1.3733333
virginica	2	6.520000	2.940000	5.586667	2.0066667
setosa	3	5.030769	3.415385	1.461538	0.2307692
versicolor	3	5.829412	2.700000	4.305882	1.3117647
virginica	3	6.400000	2.942857	5.428571	2.0642857

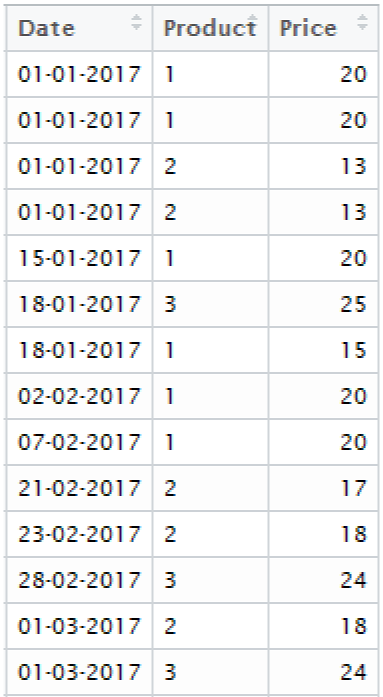
5. For the same data, get both mean and standard deviation for petal length and sepal width with respect to Species. The output would be as shown below



Species	Petal.Length.Mean	Petal.Length.SD	Sepal.Width.Mean	Sepal.Width.SD
setosa	1.462	0.173663996480184	3.428	0.379064369096289
versicolor	4.26	0.469910977239958	2.77	0.313798323378411
virginica	5.552	0.551894695663983	2.974	0.322496638172637

Hint: In aggregate, you need to use two functions “mean” and “sd”. You can create a user defined function for this. You may need to use “do.call”. Please read the documentation for “do.call”.

6. You have a data as shown below. The data in this form is known to be in “melt” form. Now to extract some information from this, we might want to manipulate the data.

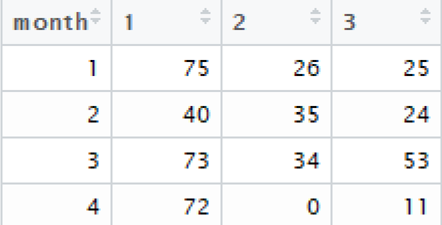


Date	Product	Price
01-01-2017	1	20
01-01-2017	1	20
01-01-2017	2	13
01-01-2017	2	13
15-01-2017	1	20
18-01-2017	3	25
18-01-2017	1	15
02-02-2017	1	20
07-02-2017	1	20
21-02-2017	2	17
23-02-2017	2	18
28-02-2017	3	24
01-03-2017	2	18
01-03-2017	3	24

For example, if we want to know in which month, the revenue was highest. Or How is business in the months given in the data, and if we want to look at the plots of monthly revenue, data in this form is not suitable. We need to convert it to “cast” form using a reshape function and the library for this is reshape2 (there are other methods to do this as well and we learn one of them, you can explore others as well)

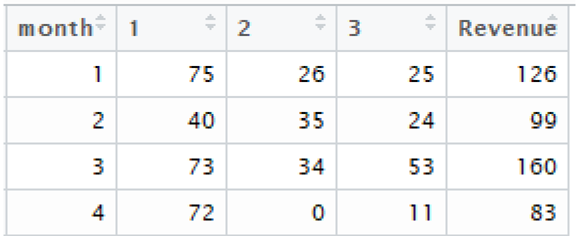
Syntax for reshape: dcast(data,formula,function,fill…)

a. Obtain the sum of sales for each month for each product using reshape function and the result is as shown below



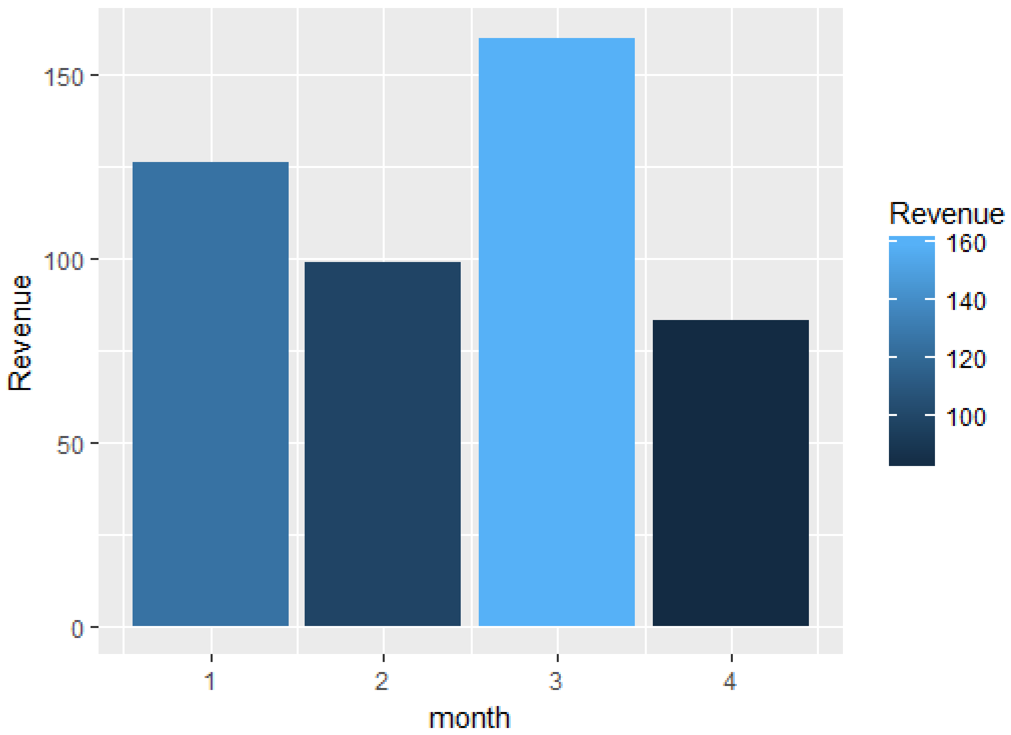
month	1	2	3
1	75	26	25
2	40	35	24
3	73	34	53
4	72	0	11

b. Now compute the total revenue for each month and the result is



month	1	2	3	Revenue
1	75	26	25	126
2	40	35	24	99
3	73	34	53	160
4	72	0	11	83

c. Now plot revenue vs month to visualize the revenue.



7. In the iris data set,
    a. subset the first 120 records and create a variable as V1. Create another variable V2 with the remaining 30 records.
    b. In both V1 and V2 remove the species column.
    c. In the last class we learnt about standardization. For standardization we need mean and standard deviation for each of the column.
    d. Now, create a function to save mean and standard deviation of each column of V1. Standardize V1.
    e. Use the mean and standard deviation obtained for each column of V1 to standardize each column of V2