Practical 2

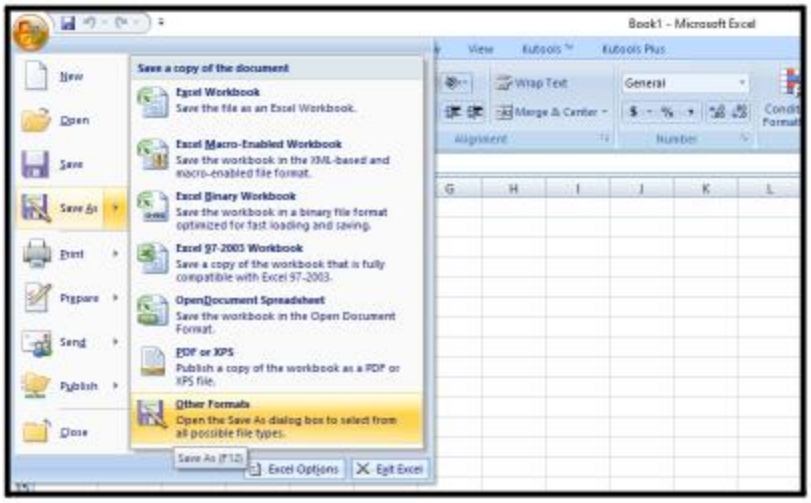
Tutorial

Steps:

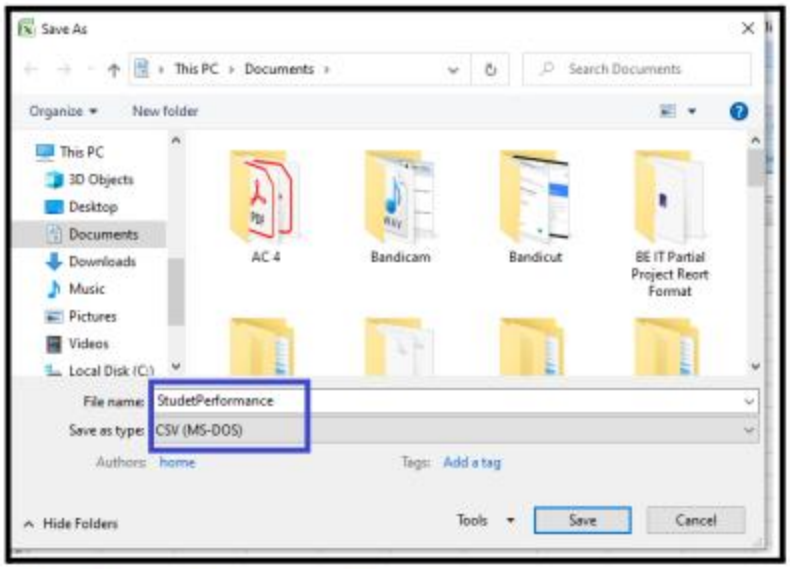
1. **For the 2nd pract we will need datasets that have missing values in them.**
2. **You can either download a dataset from kaggle then edit it to have missing values or follow the below procedure to create a data set of our own in excel**

The step to create the dataset are as follows:

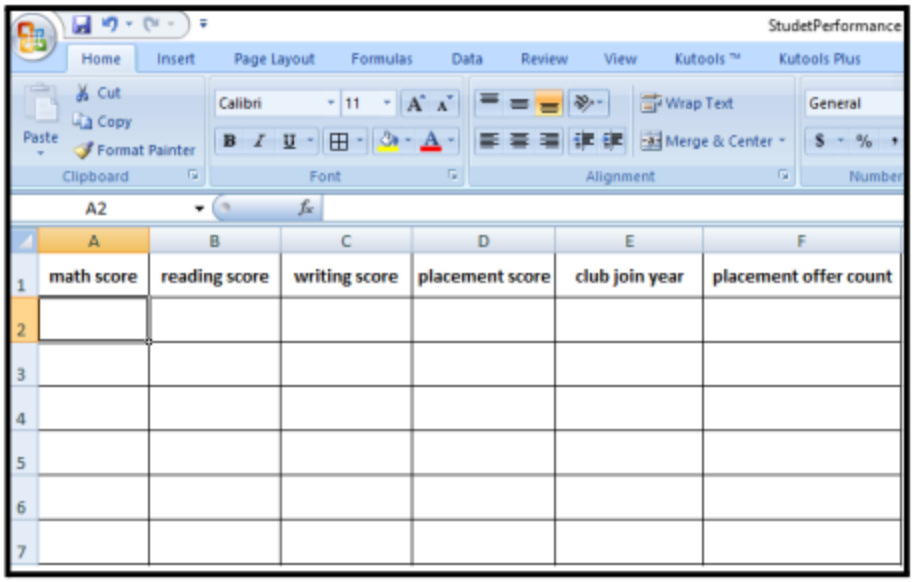
Step 1: Open Microsoft Excel and click on Save As. Select Other .Formats



Step 2: Enter the name of the dataset and Save the dataset astye CSV(MS-DOS).

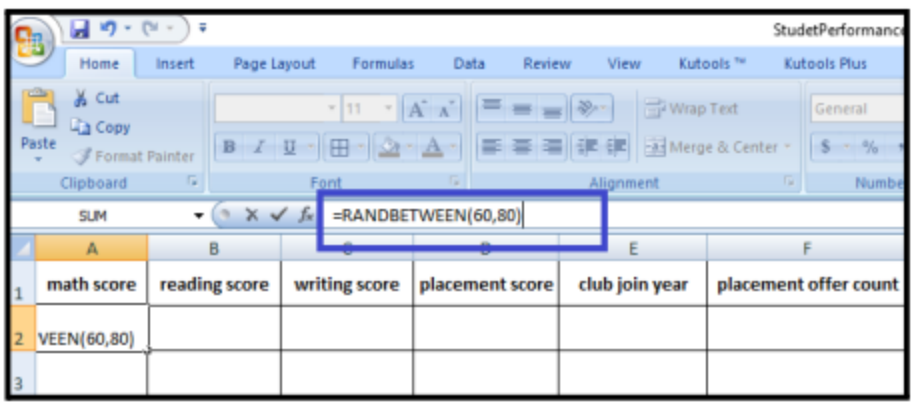


Step 3: Enter the name of features as column header.



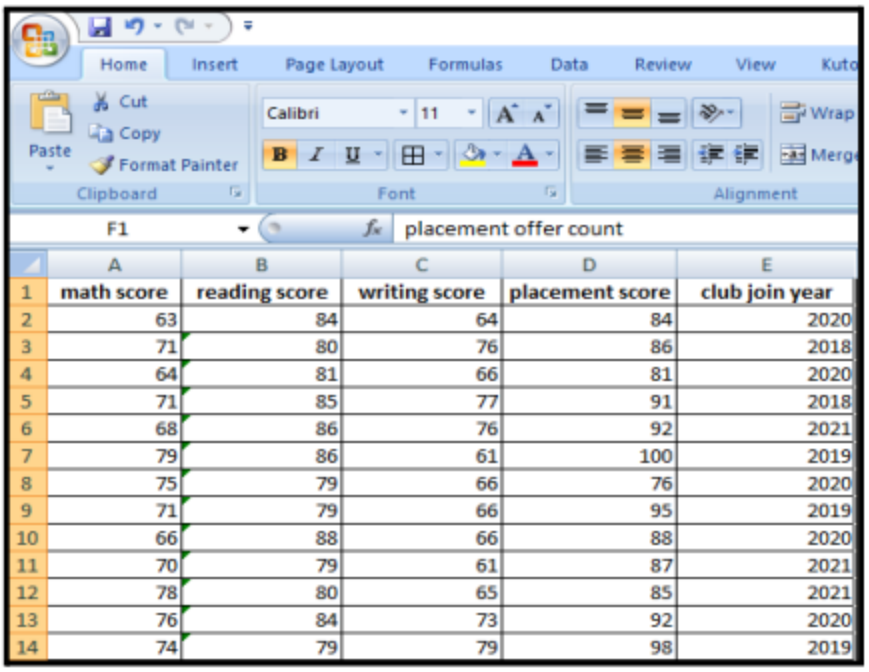
Step 3: Fill the dara by using **RANDOMBETWEEN** function. For every feature , fill the data by considering above spectified range.

one example is given:



Scroll down the cursor for 30 rows to create 30 instances.

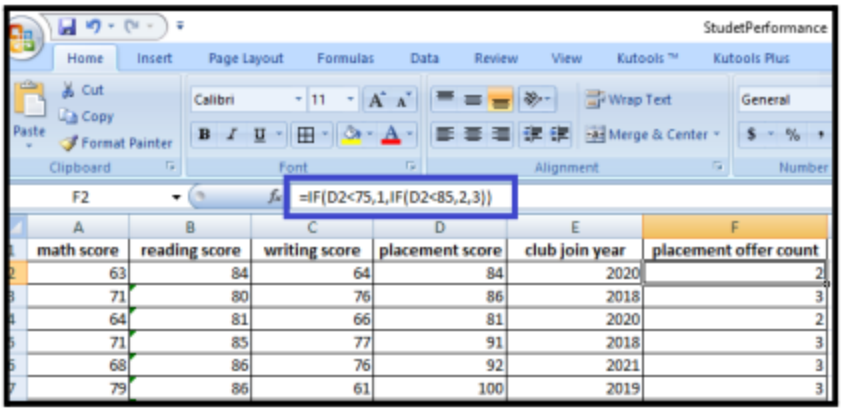
Repeat this for the features, Reading_Score, Writing_Score, Placement_Score, Club_Join_Date.



F1 placement offer count

	A	B	C	D	E
1	math score	reading score	writing score	placement score	club join year
2	63	84	64	84	2020
3	71	80	76	86	2018
4	64	81	66	81	2020
5	71	85	77	91	2018
6	68	86	76	92	2021
7	79	86	61	100	2019
8	75	79	66	76	2020
9	71	79	66	95	2019
10	66	88	66	88	2020
11	70	79	61	87	2021
12	78	80	65	85	2021
13	76	84	73	92	2020
14	74	79	79	98	2019

The placement count largely depends on the placement score. It is considered that if placement score <75, 1 offer is facilitated; for placement score >75 , 2 offer is facilitated and for else (>85) 3 offer is facilitated. Nested If formula is used for ease of data filling.



F2 =IF(D2<75,1,IF(D2<85,2,3))

	A	B	C	D	E	F
1	math score	reading score	writing score	placement score	club join year	placement offer count
2	63	84	64	84	2020	2
3	71	80	76	86	2018	3
4	64	81	66	81	2020	2
5	71	85	77	91	2018	3
6	68	86	76	92	2021	3
7	79	86	61	100	2019	3

Step 4: In 20% data, fill the impurities. The range of math score is [60,80], updating a few instances values below 60 or above 80. Repeat this for Writing_Score [60,80], Placement_Score[75-100], Club_Join_Date [2018-2021].

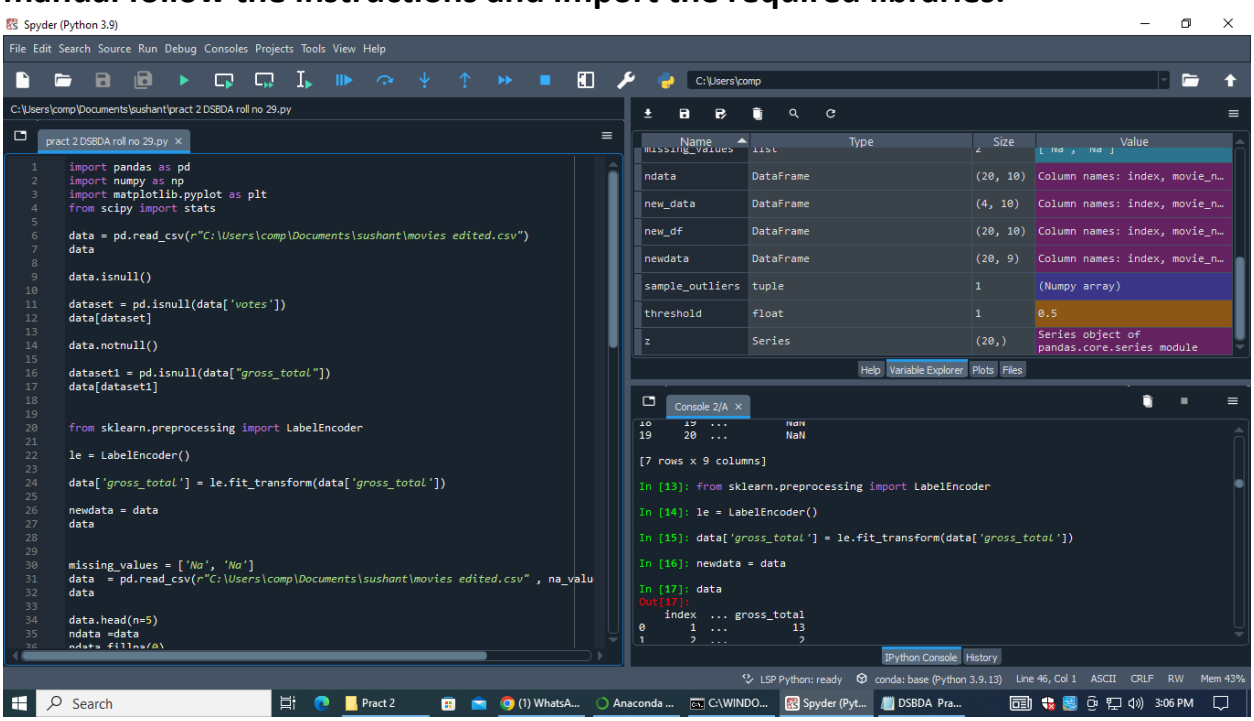
	A	B	C	D	E
1	math score	reading score	writing score	placement score	club join year
2	68	94	64	90	2018
3	72	85	70	86	2018
4	94	90	64	91	2020

Step 5: To violate the ruleof response variable, update few valus . If placement scoreis greater then 85, facilated only 1 offer.

	A	B	C	D	E	F
1	math score	reading score	writing score	placement score	club join year	placement offer count
2	70	91	64	87	2019	3
3	77	75	67	81	2020	2
4	94	84	73	99	2019	3
5	78	84	77	96	2020	1

The dataset is created with the given description.

3. **Now after the dataset is done open spyder in anaconda. As given in the manual follow the instructions and import the required libraries.**



4. **Follow the manual to execute various operations. the syntax is given on each section of the manual.**

1. Checking for missing values using isnull() and notnull()

● Checking for missing values using isnull() In order to check null values in Pandas DataFrame, isnull() function is used. This function return dataframe of Boolean values which are True for NaN values.

Algorithm:

Step 1 : Import pandas and numpy in order to check missing values in Pandas DataFrame import pandas as pd import numpy as np

Step 2: Load the dataset in dataframe object df
df=pd.read_csv("/content/StudentsPerformanceTest1.csv")

Step 3: Display the data frame df

Step 4: Use isnull() function to check null values in the dataset. df.isnull()

Step 5: To create a series true for NaN values for specific columns. for example math score in dataset and display data with only math score as NaN series = pd.isnull(df["math score"]) df[series]

2. Checking for missing values using notnull()

In order to check null values in Pandas Dataframe, notnull() function is used. This function return dataframe of Boolean values which are False for NaN values.
Algorithm:

Step 1 : Import pandas and numpy in order to check missing values in Pandas DataFrame import pandas as pd import numpy as np

Step 2: Load the dataset in dataframe object df
df=pd.read_csv("/content/StudentsPerformanceTest1.csv")

Step 3: Display the data frame df

Step 4: Use notnull() function to check null values in the dataset. df.notnull()

Step 5: To create a series true for NaN values for specific columns. for example math score in dataset and display data with only math score as NaN series1 = pd.notnull(df["math score"]) df[series1]

See that there are also categorical values in the dataset, for this, you need to use Label Encoding or One Hot Encoding.

Code:

```
 from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
 df['gender'] = le.fit_transform(df['gender'])
 newdf=df
df
```

2. Filling missing values using dropna(), fillna(), replace()

In order to fill null values in a datasets, fillna(), replace() functions are used. These functions replace NaN values with some value of their own. All these functions help in filling null values in datasets of a DataFrame.

● **For replacing null values with NaN**

```
missing_values = ["Na", "na"]
df = pd.read_csv("StudentsPerformanceTest1.csv", na_values =
missing_values)
df
```

● **Filling null values with a single value**

Step 1 : Import pandas and numpy in order to check missing values in Pandas DataFrame

```
import pandas as pd
import numpy as np
```

Step 2: Load the dataset in dataframe object df

```
df=pd.read_csv("/content/StudentsPerformanceTest1.csv")
```

Step 3: Display the data frame

```
df
```

Step 4: filling missing value using fillna()

```
ndf=df
ndf.fillna(0)
```

Step 5: filling missing values using mean, median and standard deviation of that column.

```
data['math score'] = data['math score'].fillna(data['math score'].mean())
data[''math score''] = data[''math score''].fillna(data[''math score''].median())
data['math score''] = data[''math score''].fillna(data[''math score''].std())
```

replacing missing values in forenoon column with minimum/maximum number of that column

```
data[''math score''] = data[''math score''].fillna(data[''math score''].min())
data[''math score''] = data[''math score''].fillna(data[''math score''].max())
```

Filling null values in dataset

To fill null values in dataset use inplace=true

```
m_v=df['math score'].mean()
df['math score'].fillna(value=m_v, inplace=True)
df
```

● **Filling a null values using replace() method**

Following line will replace Nan value in dataframe with value -99

```
ndf.replace(to_replace = np.nan, value = -99)
```

● **Deleting null values using dropna() method**

In order to drop null values from a dataframe, dropna() function is used. This function drops Rows/Columns of datasets with Null values in different ways.

1. Dropping rows with at least 1 null value
2. Dropping rows if all values in that row are missing
3. Dropping columns with at least 1 null value.
4. Dropping Rows with at least 1 null value in CSV file

Algorithm:

Step 1 : Import pandas and numpy in order to check missing values in Pandas DataFrame

```
import pandas as pd
import numpy as np
```

Step 2: Load the dataset in dataframe object df

```
df=pd.read_csv("/content/StudentsPerformanceTest1.csv")
```

Step 3: Display the data frame

df

Step 4:To drop rows with at least 1 null value

ndf.dropna()

Step 5: To Drop rows if all values in that row are missing

ndf.dropna(how = 'all')

Step 6: To Drop columns with at least 1 null value.

ndf.dropna(axis = 1)

Step 7 : To drop rows with at least 1 null value in CSV file.

making new data frame with dropped NA values

new_data = ndf.dropna(axis = 0, how ='any')

new_data

Detecting outliers using Boxplot:

It captures the summary of the data effectively and efficiently with only a simple box and whiskers. Boxplot summarizes sample data using 25th, 50th, and 75th percentiles. One can just get insights(quartiles, median, and outliers) into the dataset by just looking at its boxplot.

Algorithm:

Step 1 : Import pandas and numpy libraries

import pandas as pd

import numpy as np

Step 2: Load the dataset in dataframe object df

df=pd.read_csv("/content/demo.csv")

Step 3: Display the data frame

df

Step 4:Select the columns for boxplot and draw the boxplot.

col = ['math score', 'reading score' , 'writing

score','placement score']

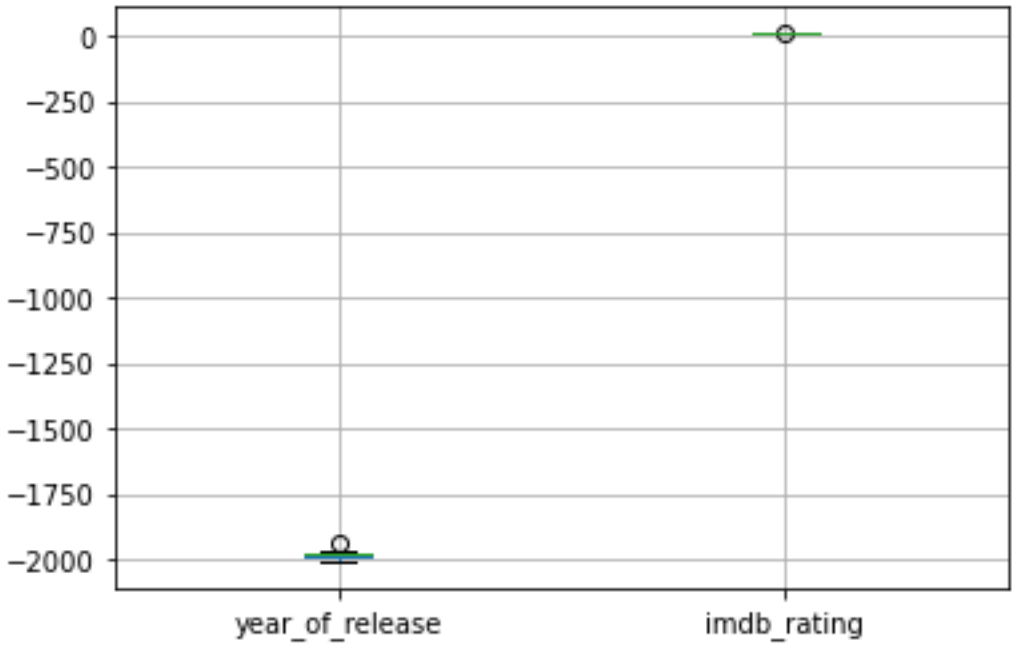
df.boxplot(col)

Step 5: We can now print the outliers for each column with reference to the box plot.

print(np.where(df['math score']>90))

print(np.where(df['reading score']<25))

print(np.where(df['writing score']<30))



Detecting outliers using Scatterplot

Algorithm:

Step 1 : Import pandas , numpy and matplotlib libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Step 2: Load the dataset in dataframe object df

```
df=pd.read_csv("/content/demo.csv")
```

Step 3: Display the data frame

```
df
```

Step 4: Draw the scatter plot with placement score and placement offer count

```
fig, ax = plt.subplots(figsize = (18,10))
ax.scatter(df['placement score'], df['placement offer
count'])
plt.show()
```

Detecting outliers using Z-Score:

Z-Score is also called a standard score. This value/score helps to understand how far is the data point from the mean. And after setting up a threshold value one can utilize z score values of data points to define the outliers.

Zscore = (data_point -mean) / std. deviation

Algorithm:

Step 1 : Import numpy and stats from scipy libraries

```
import numpy as np
```

```
from scipy import stats
```

Step 2: Calculate Z-Score for mathscore column

```
z = np.abs(stats.zscore(df['math score']))
```

Step 3: Print Z-Score Value. It prints the z-score values of each data item of the column

```
print(z)
```

Step 4: Now to define an outlier threshold value is chosen.

```
threshold = 0.18
```

Step 5: Display the sample outliers

```
sample_outliers = np.where(z <threshold)
sample_outliers
```

Histogram :

Algorithm:

Step 1 : Detecting outliers using Z-Score for the Math_score variable and remove the outliers.

Step 2: Observe the histogram for math_score variable.

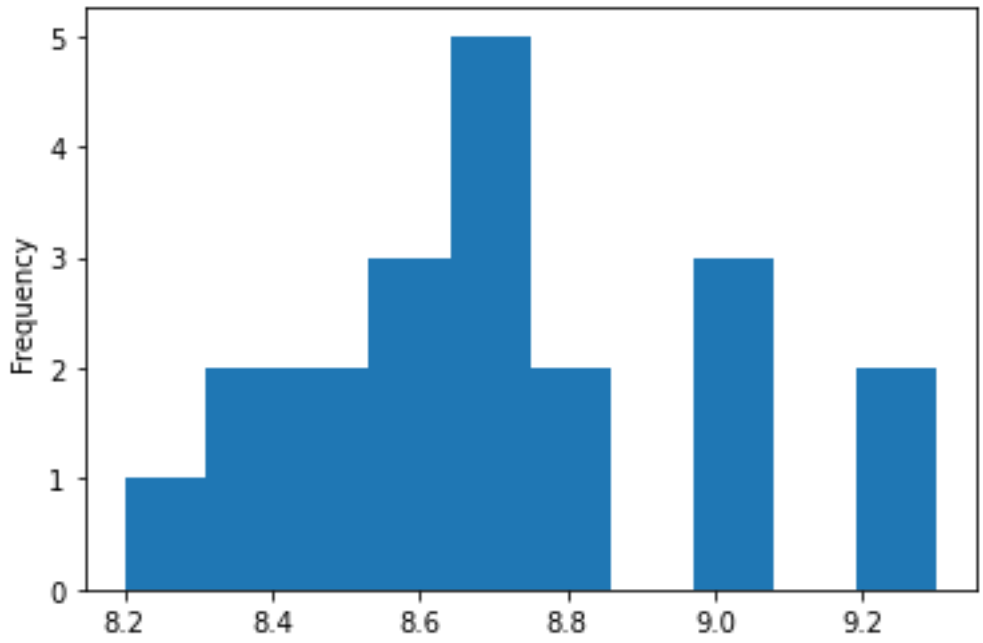
```
import matplotlib.pyplot as plt
new_df['math score'].plot(kind = 'hist')
```

Step 3: Convert the variables to logarithm at the scale 10.

```
df['log_math'] = np.log10(df['math score'])
```

Step 4: Observe the histogram for math_score variable.

```
df['log_math'].plot(kind = 'hist')
```



Finally save the file

Copy the code and output into a single text file.