

```
In [ ]: import pandas as pd  
import numpy as np  
import random  
import statistics
```

```
In [ ]: stud_data=pd.read_csv("StudentsPerformance.csv")  
stud_data
```

```
Out[ ]:
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72.0	72.0	74.0
1	female	group C	some college	standard	completed	69.0	90.0	88.0
2	female	group B	master's degree	standard	none	90.0	95.0	93.0
3	male	group A	associate's degree	free/reduced	none	NaN	57.0	44.0
4	male	group C	some college	standard	none	76.0	78.0	75.0
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88.0	99.0	95.0
996	male	group C	high school	free/reduced	none	62.0	55.0	55.0
997	female	group C	high school	free/reduced	completed	59.0	71.0	NaN
998	female	group D	some college	standard	completed	68.0	78.0	77.0
999	female	group D	some college	free/reduced	none	77.0	86.0	86.0

1000 rows × 8 columns

```
In [ ]: stud_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1000 entries, 0 to 999  
Data columns (total 8 columns):  
 #   Column           Non-Null Count  Dtype     
---  --    
 0   gender          1000 non-null    object    
 1   race/ethnicity  1000 non-null    object    
 2   parental level of education 1000 non-null    object    
 3   lunch           1000 non-null    object    
 4   test preparation course 1000 non-null    object    
 5   math score      963 non-null     float64   
 6   reading score   956 non-null     float64   
 7   writing score   962 non-null     float64  
dtypes: float64(3), object(5)  
memory usage: 62.6+ KB
```

```
In [ ]: stud_data.isnull().sum()
```

```
Out[ ]: gender          0  
race/ethnicity      0  
parental level of education 0  
lunch              0  
test preparation course 0  
math score         37  
reading score       44  
writing score        38  
dtype: int64
```

```
In [ ]: stud_data.shape #rows and columns in dataset
```

```
Out[ ]: (1000, 8)
```

```
In [ ]: stud_data.size #total data cells in dataset
```

```
Out[ ]: 8000
```

```
In [ ]: stud_data.columns
```

```
Out[ ]: Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',  
             'test preparation course', 'math score', 'reading score',  
             'writing score'],  
            dtype='object')
```

```
In [ ]: stud_data.head(10)
```

```
Out[ ]:
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72.0	72.0	74.0
1	female	group C	some college	standard	completed	69.0	90.0	88.0
2	female	group B	master's degree	standard	none	90.0	95.0	93.0
3	male	group A	associate's degree	free/reduced	none	NaN	57.0	44.0
4	male	group C	some college	standard	none	76.0	78.0	75.0
5	female	group B	associate's degree	standard	none	71.0	83.0	78.0
6	female	group B	some college	standard	completed	88.0	95.0	92.0
7	male	group B	some college	free/reduced	none	40.0	43.0	39.0
8	male	group D	high school	free/reduced	completed	64.0	64.0	67.0
9	female	group B	high school	free/reduced	none	38.0	60.0	50.0

```
In [ ]: stud_data["math score"] = stud_data['math score'].replace(np.NaN,stud_data['math score'].mean()) #replacing nan value with mean  
stud_data['math score'].head()
```

```
Out[ ]:
```

0	72.000000
1	69.000000
2	90.000000
3	66.127726
4	76.000000

Name: math score, dtype: float64

```
In [ ]: stud_data['reading score'].head(50)
```

```
Out[ ]: 0    72.0
1    90.0
2    95.0
3    57.0
4    78.0
5    83.0
6    95.0
7    43.0
8    64.0
9    60.0
10   54.0
11   52.0
12   81.0
13   72.0
14   53.0
15   75.0
16   89.0
17   32.0
18   42.0
19   58.0
20   69.0
21   75.0
22   54.0
23   73.0
24   71.0
25   74.0
26   54.0
27   69.0
28   70.0
29   NaN
30   74.0
31   65.0
32   72.0
33   42.0
34   87.0
35   81.0
36   81.0
37   64.0
38   90.0
39   56.0
40   NaN
41   73.0
42   58.0
43   65.0
44   56.0
45   54.0
46   65.0
47   71.0
48   74.0
49   84.0
```

Name: reading score, dtype: float64

```
In [ ]: stud_data['reading score']=stud_data['reading score'].replace(np.NaN,stud_data['reading score'].median())
stud_data['reading score'].head(50)
```

```
Out[ ]: 0    72.0
1    90.0
2    95.0
3    57.0
4    78.0
5    83.0
6    95.0
7    43.0
8    64.0
9    60.0
10   54.0
11   52.0
12   81.0
13   72.0
14   53.0
15   75.0
16   89.0
17   32.0
18   42.0
19   58.0
20   69.0
21   75.0
22   54.0
23   73.0
24   71.0
25   74.0
26   54.0
27   69.0
28   70.0
29   70.0
30   74.0
31   65.0
32   72.0
33   42.0
34   87.0
35   81.0
36   81.0
37   64.0
38   90.0
39   56.0
40   70.0
41   73.0
42   58.0
43   65.0
44   56.0
45   54.0
46   65.0
47   71.0
48   74.0
49   84.0
Name: reading score, dtype: float64
```

```
In [ ]: stud_data['writing score'].head(50)
```

```
Out[ ]: 0    74.0
1    88.0
2    93.0
3    44.0
4    75.0
5    78.0
6    92.0
7    39.0
8    67.0
9    50.0
10   52.0
11   43.0
12   73.0
13   70.0
14   58.0
15   78.0
16   86.0
17   NaN
18   46.0
19   61.0
20   63.0
21   70.0
22   53.0
23   73.0
24   80.0
25   72.0
26   55.0
27   75.0
28   65.0
29   NaN
30   74.0
31   61.0
32   65.0
33   38.0
34   82.0
35   79.0
36   83.0
37   59.0
38   88.0
39   57.0
40   54.0
41   68.0
42   65.0
43   66.0
44   54.0
45   57.0
46   62.0
47   76.0
48   76.0
49   82.0
Name: writing score, dtype: float64
```

```
In [ ]: stud_data['writing score']=stud_data['writing score'].replace(np.NaN,statistics.mode(stud_data['writing score']))
stud_data['writing score'].head(50)
```

```
Out[ ]: 0    74.0
1    88.0
2    93.0
3    44.0
4    75.0
5    78.0
6    92.0
7    39.0
8    67.0
9    50.0
10   52.0
11   43.0
12   73.0
13   70.0
14   58.0
15   78.0
16   86.0
17   74.0
18   46.0
19   61.0
20   63.0
21   70.0
22   53.0
23   73.0
24   80.0
25   72.0
26   55.0
27   75.0
28   65.0
29   74.0
30   74.0
31   61.0
32   65.0
33   38.0
34   82.0
35   79.0
36   83.0
37   59.0
38   88.0
39   57.0
40   54.0
41   68.0
42   65.0
43   66.0
44   54.0
45   57.0
46   62.0
47   76.0
48   76.0
49   82.0
Name: writing score, dtype: float64
```

```
In [ ]: #using fill na drop na
df1=pd.read_csv("/content/drive/MyDrive/Colab Notebooks/StudentsPerformance.csv")
df1
```

```
Out[ ]:
```

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72.0	72.0	74.0
1	female	group C	some college	standard	completed	69.0	90.0	88.0
2	female	group B	master's degree	standard	none	90.0	95.0	93.0
3	male	group A	associate's degree	free/reduced	none	Nan	57.0	44.0
4	male	group C	some college	standard	none	76.0	78.0	75.0
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88.0	99.0	95.0
996	male	group C	high school	free/reduced	none	62.0	55.0	55.0
997	female	group C	high school	free/reduced	completed	59.0	71.0	Nan
998	female	group D	some college	standard	completed	68.0	78.0	77.0
999	female	group D	some college	free/reduced	none	77.0	86.0	86.0

1000 rows × 8 columns

```
In [ ]: df1.isnull().sum()
```

```
Out[ ]: gender          0  
race/ethnicity      0  
parental level of education 0  
lunch              0  
test preparation course 0  
math score         37  
reading score       44  
writing score        38  
dtype: int64
```

```
In [ ]: df1.dropna(inplace=True)
```

```
In [ ]: df1 #dropped the whole row having null value
```

Out[ ]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72.0	72.0	74.0
1	female	group C	some college	standard	completed	69.0	90.0	88.0
2	female	group B	master's degree	standard	none	90.0	95.0	93.0
4	male	group C	some college	standard	none	76.0	78.0	75.0
5	female	group B	associate's degree	standard	none	71.0	83.0	78.0
...	...	...	...	...	...	...	...	...
994	male	group A	high school	standard	none	63.0	63.0	62.0
995	female	group E	master's degree	standard	completed	88.0	99.0	95.0
996	male	group C	high school	free/reduced	none	62.0	55.0	55.0
998	female	group D	some college	standard	completed	68.0	78.0	77.0
999	female	group D	some college	free/reduced	none	77.0	86.0	86.0

885 rows × 8 columns

In [ ]:

```
#by using fillna

stud_per=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/StudentsPerformance.csv')
stud_per.head()
```

Out[ ]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72.0	72.0	74.0
1	female	group C	some college	standard	completed	69.0	90.0	88.0
2	female	group B	master's degree	standard	none	90.0	95.0	93.0
3	male	group A	associate's degree	free/reduced	none	NaN	57.0	44.0
4	male	group C	some college	standard	none	76.0	78.0	75.0

In [ ]:

```
stud_per['math score']=stud_per['math score'].fillna(0)
stud_per.isnull().sum()
```

Out[ ]:

gender	0
race/ethnicity	0
parental level of education	0
lunch	0
test preparation course	0
math score	0
reading score	44
writing score	38
dtype: int64	

In [ ]:

```
stud_per['reading score']=stud_per['reading score'].fillna(method='ffill')
stud_per['reading score']
```

```
Out[ ]: 0    72.0
1    90.0
2    95.0
3    57.0
4    78.0
...
995   99.0
996   55.0
997   71.0
998   78.0
999   86.0
Name: reading score, Length: 1000, dtype: float64
```

```
In [ ]: stud_per['reading score']=stud_per['reading score'].interpolate(method='linear',limit_direction='forward',axis=0)
stud_per['reading score'].head()
```

```
Out[ ]: 0    72.0
1    90.0
2    95.0
3    57.0
4    78.0
Name: reading score, dtype: float64
```

```
In [ ]: stud_data
```

```
Out[ ]:   gender race/ethnicity  parental level of education  lunch  test preparation course  math score  reading score  writing score
  0  female  group B      bachelor's degree  standard        none  72.000000    72.0     74.0
  1  female  group C      some college  standard  completed  69.000000    90.0     88.0
  2  female  group B      master's degree  standard        none  90.000000    95.0     93.0
  3   male  group A  associate's degree free/reduced        none  66.127726    57.0     44.0
  4   male  group C      some college  standard        none  76.000000    78.0     75.0
...
  ...
  ...
  ...
  995  female  group E      master's degree  standard  completed  88.000000    99.0     95.0
  996   male  group C      high school  free/reduced        none  62.000000    55.0     55.0
  997  female  group C      high school  free/reduced  completed  59.000000    71.0     74.0
  998  female  group D      some college  standard  completed  68.000000    78.0     77.0
  999  female  group D      some college  free/reduced        none  77.000000    86.0     86.0
```

1000 rows × 8 columns

```
In [ ]: stud_data
```

Out[ ]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72.000000	72.0	74.0
1	female	group C	some college	standard	completed	69.000000	90.0	88.0
2	female	group B	master's degree	standard	none	90.000000	95.0	93.0
3	male	group A	associate's degree	free/reduced	none	66.127726	57.0	44.0
4	male	group C	some college	standard	none	76.000000	78.0	75.0
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88.000000	99.0	95.0
996	male	group C	high school	free/reduced	none	62.000000	55.0	55.0
997	female	group C	high school	free/reduced	completed	59.000000	71.0	74.0
998	female	group D	some college	standard	completed	68.000000	78.0	77.0
999	female	group D	some college	free/reduced	none	77.000000	86.0	86.0

1000 rows × 8 columns

In [ ]:

```
stud_data['gender']=np.where(stud_data['gender']=='female',0,1)  
stud_data['gender']
```

Out[ ]:

```
0      0  
1      0  
2      0  
3      1  
4      1  
..  
995    0  
996    1  
997    0  
998    0  
999    0  
Name: gender, Length: 1000, dtype: int64
```

In [ ]:

```
stud_data
```

Out[ ]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	0	group B	bachelor's degree	standard	none	72.000000	72.0	74.0
1	0	group C	some college	standard	completed	69.000000	90.0	88.0
2	0	group B	master's degree	standard	none	90.000000	95.0	93.0
3	1	group A	associate's degree	free/reduced	none	66.127726	57.0	44.0
4	1	group C	some college	standard	none	76.000000	78.0	75.0
...	...	...	...	...	...	...	...	...
995	0	group E	master's degree	standard	completed	88.000000	99.0	95.0
996	1	group C	high school	free/reduced	none	62.000000	55.0	55.0
997	0	group C	high school	free/reduced	completed	59.000000	71.0	74.0
998	0	group D	some college	standard	completed	68.000000	78.0	77.0
999	0	group D	some college	free/reduced	none	77.000000	86.0	86.0

1000 rows × 8 columns

In [ ]:

#normalizing the data

```
stud_data=stud_data.drop(['race/ethnicity', 'parental level of education', 'lunch', 'test preparation course'],axis=1)
stud_data
```

Out[ ]:

	gender	math score	reading score	writing score
0	0	72.000000	72.0	74.0
1	0	69.000000	90.0	88.0
2	0	90.000000	95.0	93.0
3	1	66.127726	57.0	44.0
4	1	76.000000	78.0	75.0
...	...	...	...	...
995	0	88.000000	99.0	95.0
996	1	62.000000	55.0	55.0
997	0	59.000000	71.0	74.0
998	0	68.000000	78.0	77.0
999	0	77.000000	86.0	86.0

1000 rows × 4 columns

In [ ]:

#scaling

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
```

```
stud_data=pd.DataFrame(scaler.fit_transform(stud_data), columns=stud_data.columns)
stud_data
```

Out[ ]:

	gender	math score	reading score	writing score
0	0.0	0.720000	0.662651	0.711111
1	0.0	0.690000	0.879518	0.866667
2	0.0	0.900000	0.939759	0.922222
3	1.0	0.661277	0.481928	0.377778
4	1.0	0.760000	0.734940	0.722222
...	...	...	...	...
995	0.0	0.880000	0.987952	0.944444
996	1.0	0.620000	0.457831	0.500000
997	0.0	0.590000	0.650602	0.711111
998	0.0	0.680000	0.734940	0.744444
999	0.0	0.770000	0.831325	0.844444

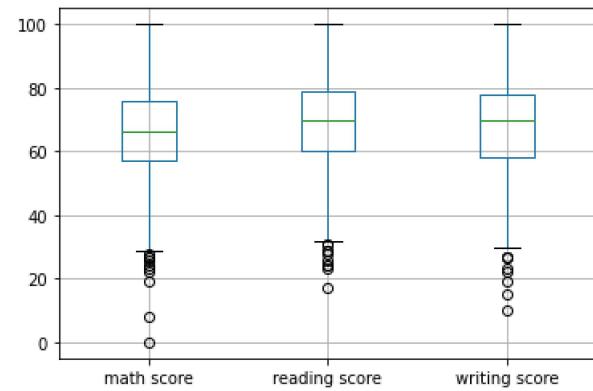
1000 rows × 4 columns

In [ ]:

```
#using boxplot
cols=['math score','reading score','writing score']
stud_data.boxplot(cols)
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f38f663ab10>
```



In [ ]:

```
stud_data[stud_data['math score']<20]
```

```
Out[ ]:    gender  math score  reading score  writing score
      59        0         0.0       17.0       10.0
     787        0        19.0       38.0       32.0
     980        0         8.0       24.0       23.0
```

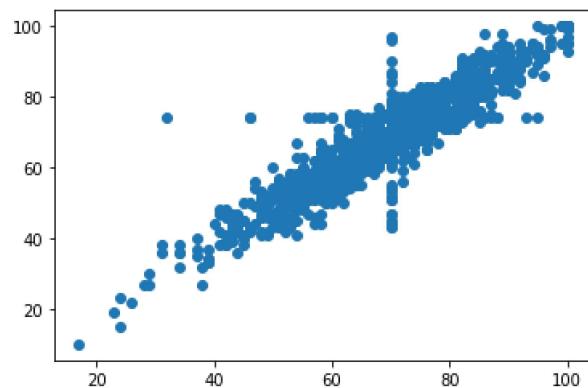
```
In [ ]: stud_data[stud_data['reading score']<20]
```

```
Out[ ]:    gender  math score  reading score  writing score
      59        0         0.0       17.0       10.0
```

```
In [ ]: stud_data[stud_data['writing score']<20]
```

```
Out[ ]:    gender  math score  reading score  writing score
      59        0         0.0       17.0       10.0
     327        1        28.0       23.0       19.0
     596        1        30.0       24.0       15.0
```

```
In [ ]: #importing matplotlib
import matplotlib.pyplot as plt
#using scatterplot
scat=plt.subplot()
scat.scatter(stud_data['reading score'],stud_data['writing score'])
plt.show()
```



```
In [ ]: scat.set_xlabel('reading score')
scat.set_ylabel('Writing score')
plt.show()
```

```
In [ ]: np.where((stud_data['reading score']<20)&(stud_data['writing score']>1))
```

```
Out[ ]: (array([59]),)
```

```
In [ ]: import scipy  
from scipy import stats
```

```
In [ ]: z=np.abs(stats.zscore(stud_data['math score']))  
z
```

```
Out[ ]: array([0.39518961, 0.19329699, 1.60654535, 0.          , 0.66437978,
   0.32789207, 1.47195027, 1.75833171, 0.14319072, 1.89292679,
   0.54697597, 1.75833171, 0.07589318, 0.79897486, 1.0853563 ,
   0.19329699, 1.47195027, 0.          , 1.35454646, 0.81616613,
   0.00859564, 0.07589318, 1.48914154, 0.19329699, 0.52978469,
   0.46248715, 0.19329699, 0.05870191, 0.26059453, 0.2777858 ,
   0.19329699, 0.21048826, 0.68157105, 1.75833171, 2.07762814,
   1.00086748, 0.52978469, 1.0853563 , 0.59708223, 0.61427351,
   0.05870191, 0.54697597, 0.88346367, 0.47967842, 1.0853563 ,
   0.07589318, 0.74886859, 0.00859564, 0.61427351, 1.06816502,
   0.88346367, 0.73167732, 0.          , 1.47195027, 0.32789207,
   2.2294145 , 1.06816502, 0.95076121, 0.54697597, 4.45023336,
   0.8662724 , 1.82562925, 0.2777858 , 0.19329699, 0.47967842,
   0.05870191, 1.421844 , 0.41238088, 0.34508334, 1.82562925,
   0.54697597, 0.21048826, 1.69103417, 0.34508334, 1.15265384,
   1.48914154, 2.43130712, 0.93356994, 0.34508334, 0.2777858 ,
   1.28724892, 1.15265384, 1.0853563 , 0.39518961, 1.62373663,
   0.46248715, 0.66437978, 0.32789207, 0.54697597, 0.46248715,
   0.07589318, 2.63319974, 0.32789207, 1.55643908, 0.8662724 ,
   0.          , 0.07589318, 0.21048826, 0.54697597, 0.07589318,
   0.8662724 , 0.12599945, 1.27005765, 0.41238088, 0.          ,
   0.54697597, 1.40465273, 0.00859564, 0.95076121, 0.26059453,
   0.73167732, 0.2777858 , 0.81616613, 1.01805875, 2.21222322,
   1.20276011, 0.59708223, 0.79897486, 1.01805875, 0.74886859,
   0.8662724 , 1.67384289, 1.47195027, 0.21048826, 1.13546256,
   1.40465273, 0.39518961, 0.07589318, 1.06816502, 1.01805875,
   1.53924781, 0.88346367, 1.40465273, 0.59708223, 0.52978469,
   0.54697597, 1.01805875, 0.26059453, 0.47967842, 0.32789207,
   0.66437978, 0.47967842, 1.62373663, 0.61427351, 1.47195027,
   2.96968745, 1.47195027, 0.          , 0.12599945, 2.27952077,
   0.2777858 , 0.73167732, 0.47967842, 0.81616613, 0.2777858 ,
   0.26059453, 0.00859564, 0.41238088, 0.34508334, 0.00859564,
   1.06816502, 0.59708223, 1.15265384, 0.95076121, 1.00086748,
   2.0103306 , 0.88346367, 0.54697597, 0.12599945, 0.05870191,
   0.39518961, 1.87573552, 0.8662724 , 0.21048826, 1.55643908,
   1.00086748, 1.35454646, 0.32789207, 0.95076121, 2.07762814,
   0.          , 1.35454646, 1.0853563 , 0.07589318, 1.421844 ,
   0.07589318, 0.93356994, 0.2777858 , 1.21995138, 0.73167732,
   0.00859564, 0.66437978, 0.2777858 , 0.73167732, 0.19329699,
   0.34508334, 0.47967842, 0.74886859, 1.421844 , 0.79897486,
   0.05870191, 0.07589318, 0.19329699, 0.61427351, 0.47967842,
   0.52978469, 1.06816502, 1.00086748, 0.52978469, 0.54697597,
   0.          , 2.09481941, 1.62373663, 0.41238088, 1.40465273,
   1.20276011, 1.13546256, 2.16211695, 0.00859564, 0.34508334,
   0.68157105, 1.40465273, 0.74886859, 1.33735519, 0.95076121,
   1.421844 , 0.39518961, 0.61427351, 0.12599945, 1.47195027,
   0.66437978, 1.35454646, 0.05870191, 1.74114044, 1.13546256,
   0.93356994, 0.21048826, 0.14319072, 0.81616613, 1.20276011,
   0.46248715, 0.93356994, 0.68157105, 0.47967842, 0.59708223,
   1.27005765, 1.53924781, 0.54697597, 0.          , 0.12599945,
   1.28724892, 0.32789207, 0.41238088, 0.93356994, 0.81616613,
   0.2777858 , 0.14319072, 0.79897486, 0.26059453, 0.07589318,
   0.14319072, 0.8662724 , 1.48914154, 2.21222322, 0.66437978,
   0.47967842, 0.21048826, 0.          , 1.47195027, 0.32789207,
   0.19329699, 0.54697597, 1.28724892, 0.07589318, 1.47195027,
   1.13546256, 1.27005765, 0.47967842, 0.07589318, 0.46248715,
   0.88346367, 1.421844 , 0.46248715, 0.26059453, 1.96022433,
   1.00086748, 2.07762814, 0.05870191, 1.47195027, 0.73167732,
```

0.66437978, 1.33735519, 0.21048826, 0.07589318, 0.79897486,  
0.05870191, 1.35454646, 0.32789207, 1.75833171, 1.60654535,  
1.00086748, 0. , 0.05870191, 0.93356994, 0.52978469,  
0.19329699, 2.21222322, 1.01805875, 0.88346367, 1.15265384,  
0.46248715, 0.00859564, 0.05870191, 0.12599945, 0. ,  
0.32789207, 0.73167732, 1.13546256, 0.21048826, 0.68157105,  
0.05870191, 0.59708223, 0.32789207, 1.55643908, 1.69103417,  
1.06816502, 0.34508334, 2.5659022 , 1.06816502, 1.69103417,  
0.32789207, 1.28724892, 0.2777858 , 1.60654535, 1.13546256,  
0.34508334, 0.66437978, 1.15265384, 2.83509237, 2.09481941,  
0.54697597, 0. , 0.19329699, 0.05870191, 0.8662724 ,  
0.39518961, 0.2777858 , 0.73167732, 0.59708223, 1.40465273,  
0.95076121, 0.00859564, 0.21048826, 0. , 0.47967842,  
0.34508334, 0.21048826, 1.62373663, 0.47967842, 0.93356994,  
0.54697597, 1.27005765, 0.95076121, 2.63319974, 0.47967842,  
1.15265384, 0.19329699, 0.34508334, 1.48914154, 0.46248715,  
1.20276011, 1.421844 , 0.52978469, 1.06816502, 0.47967842,  
1.35454646, 0.93356994, 0. , 0.32789207, 0.00859564,  
0.93356994, 1.40465273, 0.8662724 , 1.89292679, 1.89292679,  
0.05870191, 0.14319072, 0.61427351, 0.2777858 , 0. ,  
0.46248715, 0.73167732, 0.66437978, 0.61427351, 0.07589318,  
1.21995138, 1.0853563 , 1.27005765, 0.52978469, 0.41238088,  
0.47967842, 0.88346367, 1.15265384, 1.47195027, 0.81616613,  
0.21048826, 0.07589318, 1.06816502, 0.95076121, 1.40465273,  
0.26059453, 1.20276011, 0.32789207, 0.21048826, 1.01805875,  
0. , 0.32789207, 0.52978469, 0.12599945, 0.61427351,  
1.06816502, 0.61427351, 1.28724892, 0.47967842, 1.69103417,  
0.2777858 , 1.33735519, 0.19329699, 0.07589318, 0. ,  
0.14319072, 0.34508334, 0.34508334, 1.28724892, 0.46248715,  
1.0853563 , 0.59708223, 0.59708223, 0.26059453, 1.53924781,  
0.05870191, 0.79897486, 0.47967842, 0.46248715, 0.8662724 ,  
0.05870191, 0.19329699, 0. , 1.28724892, 1.00086748,  
0.14319072, 2.27952077, 0.07589318, 0.07589318, 0.88346367,  
1.96022433, 0.8662724 , 0.88346367, 2.27952077, 0.39518961,  
0.88346367, 0.81616613, 0.32789207, 0.73167732, 0.59708223,  
1.20276011, 2.70049728, 0.39518961, 0.73167732, 1.67384289,  
1.13546256, 0.21048826, 0.12599945, 0. , 1.60654535,  
0.32789207, 0.66437978, 0.93356994, 0.74886859, 0.66437978,  
0.46248715, 0.95076121, 0.12599945, 0.47967842, 1.15265384,  
0.26059453, 0.34508334, 0.41238088, 0.14319072, 0.8662724 ,  
0.07589318, 0.14319072, 1.13546256, 1.00086748, 0.81616613,  
0.12599945, 0.81616613, 0.47967842, 0. , 0.66437978,  
0.52978469, 1.87573552, 0.21048826, 1.94303306, 1.75833171,  
1.06816502, 0.12599945, 0.74886859, 0.8662724 , 1.33735519,  
0.66437978, 0.14319072, 0.2777858 , 0.81616613, 0.73167732,  
0.66437978, 0.52978469, 0.00859564, 0.00859564, 0.05870191,  
0.32789207, 1.67384289, 0.19329699, 0.81616613, 0.88346367,  
0.12599945, 0.68157105, 2.02752187, 2.49860466, 0.2777858 ,  
0.12599945, 1.28724892, 0.2777858 , 0. , 0.46248715,  
0.00859564, 1.01805875, 1.01805875, 1.27005765, 2.07762814,  
0.59708223, 0.8662724 , 1.00086748, 1.06816502, 0.14319072,  
0.79897486, 1.74114044, 0.39518961, 0.2777858 , 0.8662724 ,  
0.8662724 , 1.40465273, 1.75833171, 0.73167732, 0.88346367,  
2.29671204, 0.74886859, 0.34508334, 0.88346367, 0.46248715,  
0.52978469, 0.21048826, 2.0103306 , 0.21048826, 1.21995138,  
1.21995138, 1.74114044, 0.34508334, 0.21048826, 0.12599945,  
0.32789207, 1.67384289, 0.88346367, 1.0853563 , 0.52978469,  
1.75833171, 0.34508334, 1.00086748, 1.21995138, 0.88346367,

1.00086748, 0.73167732, 0.21048826, 0.46248715, 0.19329699,  
0.07589318, 0. , 1.48914154, 0.81616613, 1.21995138,  
0.54697597, 0.32789207, 0.12599945, 0. , 1.74114044,  
0.68157105, 2.43130712, 0.88346367, 0.19329699, 0.07589318,  
0.81616613, 2.49860466, 0. , 0.41238088, 1.20276011,  
0.59708223, 1.27005765, 1.75833171, 0.34508334, 0.54697597,  
0.19329699, 0.54697597, 1.87573552, 0.07589318, 1.06816502,  
0.41238088, 1.96022433, 1.47195027, 1.94303306, 0.07589318,  
2.09481941, 0.2777858 , 0.54697597, 2.27952077, 0.34508334,  
2.27952077, 0.19329699, 0. , 1.15265384, 1.48914154,  
0.05870191, 0.8662724 , 0.00859564, 0.59708223, 1.20276011,  
0.32789207, 0.05870191, 0.93356994, 1.33735519, 0.66437978,  
1.69103417, 0.52978469, 0.39518961, 0.52978469, 0.26059453,  
0.07589318, 0.47967842, 0.14319072, 1.0853563 , 0.19329699,  
0. , 0.12599945, 1.27005765, 0.07589318, 0.46248715,  
0.2777858 , 0.73167732, 0.19329699, 1.55643988, 1.60654535,  
0.52978469, 0.46248715, 0.74886859, 0.07589318, 0.93356994,  
1.0853563 , 0.21048826, 0.73167732, 0.46248715, 1.00086748,  
0.00859564, 0.95076121, 0.19329699, 0.07589318, 0.19329699,  
1.0853563 , 0.46248715, 0.26059453, 1.00086748, 0.21048826,  
0.05870191, 0.41238088, 0.2777858 , 2.49860466, 0.2777858 ,  
1.87573552, 1.27005765, 0.73167732, 0. , 1.80843798,  
1.15265384, 0.46248715, 0.00859564, 0.73167732, 1.15265384,  
0.8662724 , 0.59708223, 0.47967842, 0.61427351, 0.00859564,  
0.8662724 , 0.61427351, 1.40465273, 0.21048826, 0.47967842,  
0.2777858 , 1.35454646, 0.00859564, 1.53924781, 1.62373663,  
1.80843798, 0.93356994, 2.14492568, 1.00086748, 0.41238088,  
0.66437978, 0.46248715, 2.0103306 , 0.66437978, 1.67384289,  
0.2777858 , 0.74886859, 0.52978469, 1.0853563 , 1.28724892,  
1.00086748, 0.07589318, 0.12599945, 0.46248715, 0. ,  
0.12599945, 0.74886859, 1.40465273, 0.74886859, 0.88346367,  
0.05870191, 1.74114044, 0.88346367, 1.00086748, 0.34508334,  
0.93356994, 1.96022433, 1.00086748, 0.47967842, 0.74886859,  
0.39518961, 0.19329699, 0.19329699, 1.0853563 , 1.40465273,  
0.32789207, 0. , 0.8662724 , 0.73167732, 0.54697597,  
1.20276011, 0.74886859, 0.26059453, 0.95076121, 0.19329699,  
0.88346367, 1.21995138, 0.79897486, 0.2777858 , 0.41238088,  
0.52978469, 0.54697597, 0.66437978, 0.12599945, 0.54697597,  
0.95076121, 0.59708223, 0.95076121, 0.2777858 , 0.00859564,  
1.15265384, 0.00859564, 0. , 0.39518961, 1.87573552,  
1.35454646, 0.73167732, 0.66437978, 0.95076121, 1.67384289,  
2.29671204, 0.39518961, 3.17158007, 0.12599945, 0.95076121,  
1.21995138, 0.41238088, 0.00859564, 1.53924781, 1.62373663,  
0.61427351, 0.26059453, 0.26059453, 0.19329699, 0.95076121,  
0.05870191, 0.66437978, 1.40465273, 1.06816502, 0.46248715,  
0.59708223, 0.14319072, 1.69103417, 1.60654535, 0.47967842,  
1.01805875, 1.421844 , 0.81616613, 1.40465273, 0.39518961,  
1.87573552, 1.421844 , 0.34508334, 0.41238088, 0.73167732,  
1.27005765, 0.79897486, 1.15265384, 0.32789207, 1.21995138,  
0.2777858 , 0.68157105, 0.07589318, 0. , 0.12599945,  
0.34508334, 0.52978469, 0.14319072, 0.73167732, 0.54697597,  
0.41238088, 0.46248715, 0.59708223, 0.54697597, 0.00859564,  
1.82562925, 0.14319072, 2.90238991, 0.52978469, 1.75833171,  
0. , 1.67384289, 0.14319072, 0.47967842, 0.93356994,  
0.32789207, 0.34508334, 1.40465273, 1.06816502, 0.2777858 ,  
2.07762814, 0.59708223, 0. , 0.95076121, 1.40465273,  
0.88346367, 1.00086748, 1.82562925, 0.32789207, 2.07762814,  
1.06816502, 0.47967842, 0.34508334, 0.79897486, 1.15265384,

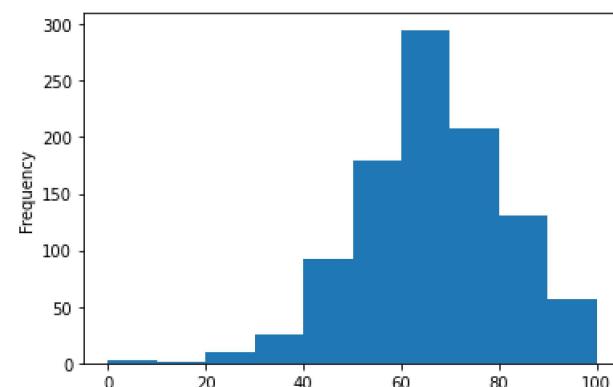
```
0.47967842, 0.26059453, 1.06816502, 1.60654535, 1.55643908,  
0.93356994, 1.00086748, 0.61427351, 0.47967842, 0.14319072,  
0.21048826, 0.32789207, 0.14319072, 0.74886859, 1.01805875,  
0.2777858 , 1.80843798, 0.81616613, 0.19329699, 1.48914154,  
1.33735519, 1.27005765, 1.0853563 , 1.47195027, 0.47967842,  
2.29671204, 2.02752187, 0.21048826, 0.05870191, 0.07589318,  
1.27005765, 0.46248715, 2.16211695, 1.80843798, 0.05870191,  
1.47195027, 0.61427351, 0.8662724 , 0.05870191, 0. ,  
1.0853563 , 0.19329699, 0.95076121, 1.28724892, 1.35454646,  
0.12599945, 2.27952077, 1.48914154, 0.61427351, 1.67384289,  
0.19329699, 2.09481941, 0.39518961, 0.81616613, 0.52978469,  
0.52978469, 0.14319072, 0.07589318, 1.35454646, 1.21995138,  
0.05870191, 0.2777858 , 0.34508334, 0.26059453, 2.14492568,  
0.26059453, 0.05870191, 0.61427351, 1.27005765, 0.73167732,  
0.39518961, 0.79897486, 1.00086748, 0.34508334, 0.54697597,  
0.81616613, 1.06816502, 1.15265384, 1.15265384, 0.61427351,  
1.87573552, 0. , 0.52978469, 0.54697597, 0.2777858 ,  
0.39518961, 1.20276011, 1.74114044, 1.421844 , 0.59708223,  
0.68157105, 1.21995138, 2.27952077, 0.07589318, 0.39518961,  
0.2777858 , 0.00859564, 0.21048826, 0.12599945, 0.59708223,  
1.53924781, 0.79897486, 0.88346367, 1.15265384, 0.81616613,  
0.14319072, 0. , 0.2777858 , 0.74886859, 1.67384289,  
3.91185303, 1.00086748, 0.8662724 , 0.79897486, 0.52978469,  
0.61427351, 1.75833171, 1.00086748, 1.48914154, 0.05870191,  
1.33735519, 0.07589318, 0.74886859, 0.2777858 , 0.21048826,  
1.47195027, 0.2777858 , 0.47967842, 0.12599945, 0.73167732])
```

```
In [ ]: threshhold=0.00001  
#display outliers  
  
sample_outliers=np.where(z  
sample_outliers
```

```
Out[ ]: (array([ 3, 17, 52, 95, 104, 147, 180, 210, 248, 267, 301, 314, 341,  
       353, 377, 389, 415, 429, 447, 473, 498, 533, 586, 593, 602, 627,  
       650, 688, 729, 751, 777, 828, 845, 857, 909, 951, 976]),)
```

```
In [ ]: stud_data['math score'].plot(kind='hist')
```

```
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f38f59e6290>
```



```
In [ ]: stud_data['logmath']=np.log10(stud_data['math score'])  
stud_data
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/arraylike.py:364: RuntimeWarning: divide by zero encountered in log10  
    result = getattr(ufunc, method)(*inputs, **kwargs)
```

```
Out[ ]:   gender  math score  reading score  writing score  logmath  
0         0      72.000000        72.0          74.0  1.857332  
1         0      69.000000        90.0          88.0  1.838849  
2         0      90.000000       95.0          93.0  1.954243  
3         1     66.127726        57.0          44.0  1.820384  
4         1     76.000000        78.0          75.0  1.880814  
...       ...       ...       ...       ...       ...  
995        0     88.000000       99.0          95.0  1.944483  
996        1     62.000000       55.0          55.0  1.792392  
997        0     59.000000       71.0          74.0  1.770852  
998        0     68.000000       78.0          77.0  1.832509  
999        0     77.000000       86.0          86.0  1.886491
```

1000 rows × 5 columns