In [3]:
```
import pandas as pd
import numpy as np
import random
import statistics
```

In [4]:
```
stud_data=pd.read_csv("StudentsPerformance (1).csv")
stud_data
```

Out[4]:

	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 [3]:
```
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 [4]:
```
stud_data.isnull().sum()
```

Out[4]:
```
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 [5]:
```
stud_data.shape
```

Out[5]:
```
(1000, 8)
```

In [6]:
```
stud_data.size
```

Out[6]:
```
8000
```

In [7]:
```
stud_data.columns
```

Out[7]:
```
Index(['gender', 'race/ethnicity', 'parental level of education', 'lunch',
       'test preparation course', 'math score', 'reading score',
       'writing score'],
      dtype='object')
```

In [8]:
```
stud_data.head(10)
```

Out[8]:

	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 [9]:
```
stud_data["math score"]=stud_data['math score'].replace(np.NaN,stud_data['math
stud_data['math score'].head()
```

Out[9]:
```
0    72.000000
1    69.000000
2    90.000000
3    66.127726
4    76.000000
Name: math score, dtype: float64
```

In [5]:
```
stud_data['reading score'].head(50)
```

Out[5]:
```
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 [6]:
```
stud_data['reading score']=stud_data['reading score'].replace(np.NaN,stud_data
stud_data['reading score'].head(50)
```

Out[6]:
```
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 [7]:
```
stud_data['writing score'].head(50)
```

Out[7]:
```
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 [8]:
```
stud_data['writing score']=stud_data['writing score'].replace(np.NaN,statistic
stud_data['writing score'].head(50)
```

Out[8]:
```
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 [9]:
```
df1=pd.read_csv("StudentsPerformance (1).csv")
df1
```

Out[9]:

	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 [10]:
```
df1.isnull().sum()
```

Out[10]:
```
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 [11]:
```
df1.dropna(inplace=True)
```

In [12]:
```
df1
```

Out[12]:

	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 [13]:
```
stud_per=pd.read_csv('StudentsPerformance (1).csv')
stud_per.head()
```

Out[13]:

	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 [14]:
```
stud_per['math score']=stud_per['math score'].fillna(0)
stud_per.isnull().sum()
```

Out[14]:
```
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 [15]:
```
stud_per['reading score']=stud_per['reading score'].fillna(method='ffill')
stud_per['reading score']
```

Out[15]:
```
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 [16]:
```
stud_per['reading score']=stud_per['reading score'].interpolate(method='linear
stud_per['reading score'].head()
```

Out[16]:
```
0    72.0
1    90.0
2    95.0
3    57.0
4    78.0
Name: reading score, dtype: float64
```

In [17]:
```
stud_data
```

Out[17]:

	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	74.0
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

1000 rows × 8 columns

In [18]:
```
stud_data
```

Out[18]:

	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	74.0
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

1000 rows × 8 columns

In [19]:
```
stud_data['gender']=np.where(stud_data['gender']=='female',0,1)
stud_data['gender']
```

Out[19]:
```
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: int32
```

In [20]:
```
stud_data
```

Out[20]:

	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	72.0	72.0	74.0
1	0	group C	some college	standard	completed	69.0	90.0	88.0
2	0	group B	master's degree	standard	none	90.0	95.0	93.0
3	1	group A	associate's degree	free/reduced	none	NaN	57.0	44.0
4	1	group C	some college	standard	none	76.0	78.0	75.0
...	...	...	...	...	...	...	...	...
995	0	group E	master's degree	standard	completed	88.0	99.0	95.0
996	1	group C	high school	free/reduced	none	62.0	55.0	55.0
997	0	group C	high school	free/reduced	completed	59.0	71.0	74.0
998	0	group D	some college	standard	completed	68.0	78.0	77.0
999	0	group D	some college	free/reduced	none	77.0	86.0	86.0

1000 rows × 8 columns

In [21]:
```
stud_data=stud_data.drop(['race/ethnicity', 'parental level of education',  'l
stud_data
```

Out[21]:

	gender	math score	reading score	writing score
0	0	72.0	72.0	74.0
1	0	69.0	90.0	88.0
2	0	90.0	95.0	93.0
3	1	NaN	57.0	44.0
4	1	76.0	78.0	75.0
...	...	...	...	...
995	0	88.0	99.0	95.0
996	1	62.0	55.0	55.0
997	0	59.0	71.0	74.0
998	0	68.0	78.0	77.0
999	0	77.0	86.0	86.0

1000 rows × 4 columns

In [22]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))

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

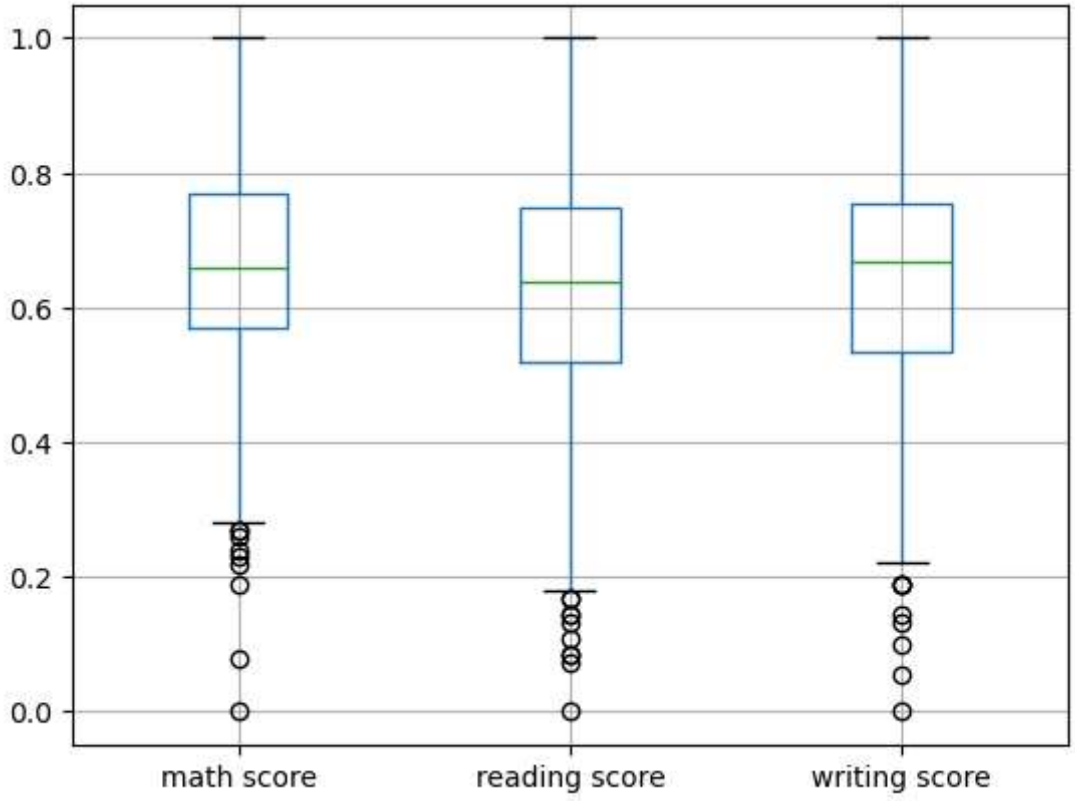
Out[22]:

	gender	math score	reading score	writing score
0	0.0	0.72	0.662651	0.711111
1	0.0	0.69	0.879518	0.866667
2	0.0	0.90	0.939759	0.922222
3	1.0	NaN	0.481928	0.377778
4	1.0	0.76	0.734940	0.722222
...	...	...	...	...
995	0.0	0.88	0.987952	0.944444
996	1.0	0.62	0.457831	0.500000
997	0.0	0.59	0.650602	0.711111
998	0.0	0.68	0.734940	0.744444
999	0.0	0.77	0.831325	0.844444

1000 rows × 4 columns

In [23]:
```
cols=['math score','reading score','writing score']
stud_data.boxplot(cols)
```

Out[23]: <Axes: >



In [24]:
```
stud_data[stud_data['math score']<20]
```

Out[24]:

	gender	math score	reading score	writing score
0	0.0	0.72	0.662651	0.711111
1	0.0	0.69	0.879518	0.866667
2	0.0	0.90	0.939759	0.922222
4	1.0	0.76	0.734940	0.722222
5	0.0	0.71	0.795181	0.755556
...	...	...	...	...
995	0.0	0.88	0.987952	0.944444
996	1.0	0.62	0.457831	0.500000
997	0.0	0.59	0.650602	0.711111
998	0.0	0.68	0.734940	0.744444
999	0.0	0.77	0.831325	0.844444

963 rows × 4 columns

In [25]:
```
stud_data[stud_data['reading score']<20]
```

Out[25]:

	gender	math score	reading score	writing score
0	0.0	0.72	0.662651	0.711111
1	0.0	0.69	0.879518	0.866667
2	0.0	0.90	0.939759	0.922222
3	1.0	NaN	0.481928	0.377778
4	1.0	0.76	0.734940	0.722222
...	...	...	...	...
995	0.0	0.88	0.987952	0.944444
996	1.0	0.62	0.457831	0.500000
997	0.0	0.59	0.650602	0.711111
998	0.0	0.68	0.734940	0.744444
999	0.0	0.77	0.831325	0.844444

1000 rows × 4 columns

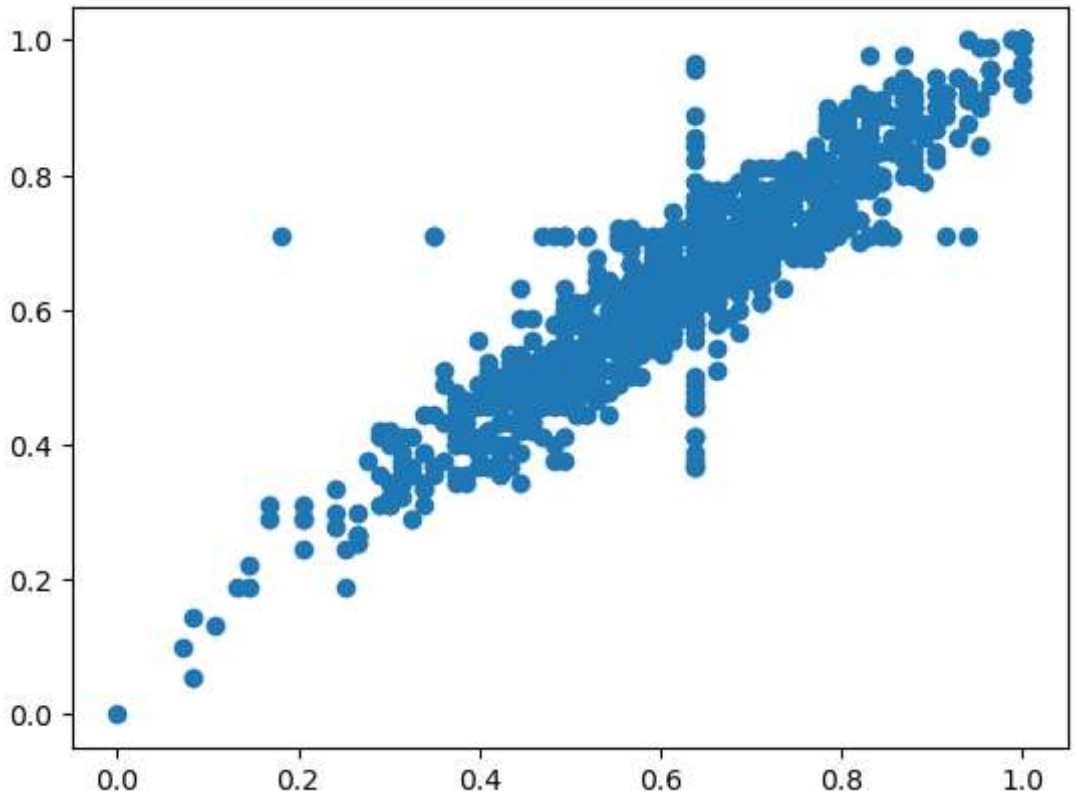
In [26]:
```
stud_data[stud_data['writing score']<20]
```

Out[26]:

	gender	math score	reading score	writing score
0	0.0	0.72	0.662651	0.711111
1	0.0	0.69	0.879518	0.866667
2	0.0	0.90	0.939759	0.922222
3	1.0	NaN	0.481928	0.377778
4	1.0	0.76	0.734940	0.722222
...	...	...	...	...
995	0.0	0.88	0.987952	0.944444
996	1.0	0.62	0.457831	0.500000
997	0.0	0.59	0.650602	0.711111
998	0.0	0.68	0.734940	0.744444
999	0.0	0.77	0.831325	0.844444

1000 rows × 4 columns

In [27]:
```
import matplotlib.pyplot as plt
#using scatterplot
scat=plt.subplot()
scat.scatter(stud_data['reading score'],stud_data['writing score'])
plt.show()
```



In [28]:
```
scat.set_xlabel('reading score')
scat.set_ylabel('Writing score')
plt.show()
```

In [29]:
```
np.where((stud_data['reading score']<20)&(stud_data['writing score']>1))
```

Out[29]:
```
(array([], dtype=int64),)
```

In [30]:
```
import scipy
from scipy import stats
```

In [31]:
```
z=np.abs(stats.zscore(stud_data['math score']))
z
```

Out[31]:
```
0     NaN
1     NaN
2     NaN
3     NaN
4     NaN
      ..
995   NaN
996   NaN
997   NaN
998   NaN
999   NaN
Name: math score, Length: 1000, dtype: float64
```

In [32]:
```
threshhold=0.00001
#display outliers

sample_outliers=np.where(z<threshhold)

sample_outliers
```

Out[32]:
```
(array([], dtype=int64),)
```

In [34]:
```
stud_data['math score'].plot(kind='hist')
```

Out[34]:
```
<Axes: ylabel='Frequency'>
```

In [35]:
```
stud_data['logmath']=np.log10(stud_data['math score'])
stud_data
```

```
C:\Users\user\anaconda3\lib\site-packages\pandas\core\arraylike.py:402: Runti
meWarning: divide by zero encountered in log10
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

Out[35]:

	gender	math score	reading score	writing score	logmath
0	0.0	0.72	0.662651	0.711111	-0.142668
1	0.0	0.69	0.879518	0.866667	-0.161151
2	0.0	0.90	0.939759	0.922222	-0.045757
3	1.0	NaN	0.481928	0.377778	NaN
4	1.0	0.76	0.734940	0.722222	-0.119186
...	...	...	...	...	...
995	0.0	0.88	0.987952	0.944444	-0.055517
996	1.0	0.62	0.457831	0.500000	-0.207608
997	0.0	0.59	0.650602	0.711111	-0.229148
998	0.0	0.68	0.734940	0.744444	-0.167491
999	0.0	0.77	0.831325	0.844444	-0.113509

1000 rows × 5 columns

In [ ]:
```

```