In [8]:
```
#Practical No 3
#Ajit waman B-54
import pandas as pd
import numpy as np
student = pd.read_csv("/home/kj-comp/StudentsPerformance.csv")
```

In [9]:
```
student.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                   991 non-null    float64
 6   reading_score                995 non-null    float64
 7   writing_score                994 non-null    float64
dtypes: float64(3), object(5)
memory usage: 62.6+ KB
```

In [10]:
```
student.describe()
```

Out[10]:

	math_score	reading_score	writing_score
count	991.000000	995.000000	994.000000
mean	66.116044	69.223116	68.113682
std	15.217867	14.577775	15.182945
min	0.000000	17.000000	10.000000
25%	57.000000	59.000000	58.000000
50%	66.000000	70.000000	69.000000
75%	77.000000	79.000000	79.000000
max	100.000000	100.000000	100.000000

In [11]:
```
student.head()
```

Out[11]:

	gender	race/ethnicity	parental level of education	lunch	test_preparation_course	math_score	reading_s
0	female	group B	bachelor's degree	standard	none	72.0	
1	female	group C	some college	standard	completed	69.0	
2	female	group B	master's degree	standard	none	90.0	
3	male	group A	associate's degree	free/reduced	none	47.0	
4	male	group C	some college	standard	none	76.0	

In [12]:
```
male_female = student.groupby('gender')['gender'].count()
print(male_female)
```

```
gender
female    518
male      482
Name: gender, dtype: int64
```

In [13]:
```
student.test_preparation_course.unique()
```

Out[13]:
```
array(['none', 'completed'], dtype=object)
```

In [14]:
```
mean_math = student.groupby('gender').math_score.mean()
```

In [15]:
```
print(mean_math)
```

```
gender
female    63.654902
male      68.725572
Name: math_score, dtype: float64
```

In [18]:
```
mean_math_test_preparation = student.groupby(['gender','test_preparation_cours
e']).math_score.mean()
print(mean_math_test_preparation)
```

```
gender  test_preparation_course
female  completed                  67.331492
        none                       61.632219
male    completed                  72.339080
        none                       66.677524
Name: math_score, dtype: float64
```

In [19]:
```
student.math_score.unique()
```

Out[19]:
```
array([ 72.,  69.,  90.,  47.,  76.,  71.,  88.,  40.,  64.,  38.,  58.,
        nan,  78.,  50.,  18.,  46.,  54.,  66.,  65.,  44.,  74.,  73.,
        70.,  62.,  63.,  56.,  97.,  81.,  75.,  57.,  55.,  53.,  59.,
        82.,  77.,  33.,  52.,   0.,  79.,  39.,  67.,  45.,  60.,  61.,
        41.,  49.,  30.,  80.,  42.,  27.,  43.,  68.,  85.,  98.,  87.,
        51.,  99.,  84.,  91.,  83.,  89.,  22., 100.,  96.,  94.,  48.,
        35.,  34.,  86.,  92.,  37.,  28.,  24.,  26.,  95.,  36.,  29.,
        32.,  93.,  19.,  23.,   8.])
```

In [20]:
```
print(student.groupby('gender').math_score.describe())
```

```
        count       mean        std   min   25%   50%   75%    max
gender
female  510.0  63.654902  15.593640   0.0  54.0  65.0  74.0  100.0
male    481.0  68.725572  14.371106  27.0  59.0  69.0  79.0  100.0
```

In [21]:
```
groups = pd.cut(student['math_score'],bins=4)
groups
```

Out[21]:
```
0       (50.0, 75.0]
1       (50.0, 75.0]
2      (75.0, 100.0]
3       (25.0, 50.0]
4      (75.0, 100.0]
           ...
995    (75.0, 100.0]
996     (50.0, 75.0]
997     (50.0, 75.0]
998     (50.0, 75.0]
999    (75.0, 100.0]
Name: math_score, Length: 1000, dtype: category
Categories (4, interval[float64]): [(-0.1, 25.0] < (25.0, 50.0] < (50.0, 75.
0] < (75.0, 100.0]]
```

In [22]:
```
student.groupby(groups)['math_score'].count()
```

Out[22]:
```
math_score
(-0.1, 25.0]       7
(25.0, 50.0]     143
(50.0, 75.0]     567
(75.0, 100.0]    274
Name: math_score, dtype: int64
```

In [23]:
```
pd.crosstab(groups, student['gender'])
```

Out[23]:

gender math_score	female	male
(-0.1, 25.0]	7	0
(25.0, 50.0]	90	53
(50.0, 75.0]	301	266
(75.0, 100.0]	112	162

In [24]:
```
import statistics as st
```

In [25]:
```
data = [1,2,3,4,5,6]
```

In [26]:
```
st.mean(data)
```

Out[26]:
```
3.5
```

In [27]:
```
st.median(data)
```

Out[27]:
```
3.5
```

In [28]:
```
#Will show error as data is having no unique modal value
st.mode(data)
```

```
---------------------------------------------------------------------------
StatisticsError                           Traceback (most recent call last)
<ipython-input-28-7adf61ce2b58> in <module>
      1 #Will show error as data is having no unique modal value
----> 2 st.mode(data)

~/anaconda3/lib/python3.7/statistics.py in mode(data)
    504     elif table:
    505         raise StatisticsError(
--> 506                 'no unique mode; found %d equally common values' % le
n(table)
    507                 )
    508     else:

StatisticsError: no unique mode; found 6 equally common values
```

In [29]:
```
data1 = [1,2,7,5,4,7,8,2,1,7]
st.mode(data1)
```

Out[29]:
```
7
```

In [30]:
```
#Variance
st.variance(data1)
```

Out[30]:
```
7.6
```

In [31]:
```
#Variance
st.variance(data1)
```
Out[31]:
```
7.6
```

In [32]:
```
import pandas as pd
df = pd.DataFrame(data1)
```

In [33]:
```
df.mean()
```
Out[33]:
```
0    4.4
dtype: float64
```

In [34]:
```
df.mode()
```
Out[34]:

	0
0	7

In [35]:
```
df.median()
```
Out[35]:
```
0    4.5
dtype: float64
```

In [42]:
```
#using California housing train csv file
df1 = pd.read_csv("/home/kj-comp/california_housing_test(1).csv")
df1
```
Out[42]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	househo
0	-122.05	37.37	27.0	3885.0	661.0	1537.0	60
1	-118.30	34.26	43.0	1510.0	310.0	809.0	27
2	-117.81	33.78	27.0	3589.0	507.0	1484.0	49
3	-118.36	33.82	28.0	67.0	15.0	49.0	1
4	-119.67	36.33	19.0	1241.0	244.0	850.0	23
...	...	...	...	...	...	...	
2995	-119.86	34.42	23.0	1450.0	642.0	1258.0	60
2996	-118.14	34.06	27.0	5257.0	1082.0	3496.0	103
2997	-119.70	36.30	10.0	956.0	201.0	693.0	22
2998	-117.12	34.10	40.0	96.0	14.0	46.0	1
2999	-119.63	34.42	42.0	1765.0	263.0	753.0	26

3000 rows × 9 columns

In [43]:
```
df1.mean()
```

Out[43]:
```
longitude                -119.589200
latitude                   35.635390
housing_median_age         28.845333
total_rooms              2599.578667
total_bedrooms            529.950667
population               1402.798667
households                489.912000
median_income               3.807272
median_house_value     205846.275000
dtype: float64
```

In [44]:
```
df1["households"].mean()
```

Out[44]:
```
489.912
```

In [45]:
```
df1["households"].median()
```

Out[45]:
```
409.5
```

In [46]:
```
df1["households"].mode()
```

Out[46]:
```
0    273.0
1    375.0
2    614.0
dtype: float64
```

In [47]:
```
df1["households"].var()
```

Out[47]:
```
133533.75684161368
```

In [48]:
```
st.stdev(df1["households"])
```

Out[48]:
```
365.42270980552627
```

In [51]:
```
import pandas as pd
data = pd.read_csv("iris(1).csv")
print('Iris-setosa')
```

```
Iris-setosa
```

In [52]:
```
setosa = data['species'] == 'Iris-setosa'
print(data[setosa].describe())
```

```
       sepal_length  sepal_width  petal_length  petal_width
count           0.0          0.0           0.0          0.0
mean            NaN          NaN           NaN          NaN
std             NaN          NaN           NaN          NaN
min             NaN          NaN           NaN          NaN
25%             NaN          NaN           NaN          NaN
50%             NaN          NaN           NaN          NaN
75%             NaN          NaN           NaN          NaN
max             NaN          NaN           NaN          NaN
```

In [53]:
```
print('\nIris-versicolor')
setosa = data['species'] == 'Iris-versicolor'
print(data[setosa].describe())
```

```
Iris-versicolor
       sepal_length  sepal_width  petal_length  petal_width
count           0.0          0.0           0.0          0.0
mean            NaN          NaN           NaN          NaN
std             NaN          NaN           NaN          NaN
min             NaN          NaN           NaN          NaN
25%             NaN          NaN           NaN          NaN
50%             NaN          NaN           NaN          NaN
75%             NaN          NaN           NaN          NaN
max             NaN          NaN           NaN          NaN
```

In [54]:
```
print('\nIris-virginica')
setosa = data['species'] == 'Iris-virginica'
print(data[setosa].describe())
```

```
Iris-virginica
       sepal_length  sepal_width  petal_length  petal_width
count           0.0          0.0           0.0          0.0
mean            NaN          NaN           NaN          NaN
std             NaN          NaN           NaN          NaN
min             NaN          NaN           NaN          NaN
25%             NaN          NaN           NaN          NaN
50%             NaN          NaN           NaN          NaN
75%             NaN          NaN           NaN          NaN
max             NaN          NaN           NaN          NaN
```

In [ ]: