

## exp-3

April 26, 2024

```
[ ]: #exp_3
     #Name:Tushar Holkar
     #Roll No: A-36
```

```
[2]: import pandas as pd
     import numpy as np
     student = pd.read_csv("/home/kj-comp/Tushar Holkar/GCR/DB/StudentsPerformance.
     ↪csv")
```

```
[3]: 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
```

```
[4]: student.describe()
```

```
[4]:
```

	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

```
[5]: student.head()
```

```
[5]:   gender race/ethnicity parental level of education      lunch \
0  female      group B      bachelor's degree    standard
1  female      group C      some college    standard
2  female      group B      master's degree    standard
3   male      group A      associate's degree  free/reduced
4   male      group C      some college    standard

   test_preparation_course  math_score  reading_score  writing_score
0                none        72.0         72.0         74.0
1          completed        69.0         90.0         88.0
2                none        90.0         95.0         93.0
3                none        47.0         57.0         44.0
4                none        76.0         78.0         75.0
```

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

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

```
[7]: student.test_preparation_course.unique()
```

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

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

```
[9]: print(mean_math)
```

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

```
[11]: mean_math_test_preparation = student.
      ↪groupby(['gender', 'test_preparation_course']).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
```

```
[12]: student.math_score.unique()
```

```
[12]: 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.])
```

```
[13]: 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

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

```
[14]: 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, right]): [(-0.1, 25.0] < (25.0, 50.0] < (50.0, 75.0] < (75.0, 100.0]]
```

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

```
[15]: 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
```

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

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

```
[17]: import statistics as st
```

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

```
[19]: st.mean(data)
```

```
[19]: 3.5
```

```
[20]: st.median(data)
```

```
[20]: 3.5
```

```
[21]: st.mode(data)
```

```
[21]: 1
```

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

```
[22]: 7
```

```
[23]: st.variance(data1)
```

```
[23]: 7.6
```

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

```
[25]: df.mean()
```

```
[25]: 0    4.4
      dtype: float64
```

```
[26]: df.mode()
```

```
[26]: 0
      0  7
```

```
[27]: df.median()
```

```
[27]: 0    4.5
      dtype: float64
```

```
[28]: df1 = pd.read_csv("/home/kj-comp/california_housing_test.csv")
      df1
```

```
[28]:
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	\
0	-122.05	37.37	27.0	3885.0	661.0	
1	-118.30	34.26	43.0	1510.0	310.0	
2	-117.81	33.78	27.0	3589.0	507.0	
3	-118.36	33.82	28.0	67.0	15.0	
4	-119.67	36.33	19.0	1241.0	244.0	
...	...	...	...	...	...	
2995	-119.86	34.42	23.0	1450.0	642.0	
2996	-118.14	34.06	27.0	5257.0	1082.0	
2997	-119.70	36.30	10.0	956.0	201.0	
2998	-117.12	34.10	40.0	96.0	14.0	
2999	-119.63	34.42	42.0	1765.0	263.0	

  

	population	households	median_income	median_house_value
0	1537.0	606.0	6.6085	344700.0
1	809.0	277.0	3.5990	176500.0
2	1484.0	495.0	5.7934	270500.0
3	49.0	11.0	6.1359	330000.0
4	850.0	237.0	2.9375	81700.0
...	...	...	...	...
2995	1258.0	607.0	1.1790	225000.0
2996	3496.0	1036.0	3.3906	237200.0
2997	693.0	220.0	2.2895	62000.0
2998	46.0	14.0	3.2708	162500.0
2999	753.0	260.0	8.5608	500001.0

[3000 rows x 9 columns]

```
[29]: df1.mean()
```

```
[29]: 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
```

```
[30]: df1["households"].mean()
```

```
[30]: 489.912
```

```
[31]: df1["households"].median()
```

```
[31]: 409.5
```

```
[32]: df1["households"].mode()
```

```
[32]: 0    273.0  
      1    375.0  
      2    614.0  
      Name: households, dtype: float64
```

```
[33]: df1["households"].var()
```

```
[33]: 133533.75684161368
```

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

```
[34]: 365.42270980552627
```

```
[36]: import pandas as pd  
      data = pd.read_csv("/home/kj-comp/iris.csv")  
      print('Iris-setosa')
```

Iris-setosa

```
[37]: 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

```
[38]: 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

```
[39]: 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

```
[ ]:
```