

assn3

May 22, 2023

```
[14]: #Aishwarya kelgandre Roll no.73 batch T3
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
s1 =pd.Series(range(1,10,1))
s1
import pandas as pd
import numpy as np
student = pd.read_csv("E:\\TRINITY ACADEMY OF ENGINEERING PUNE\\TE_
↪2022-23\\assignment\\dsbda\\csv\\StudentsPerformance.csv")
```

```
[2]: 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                           1000 non-null   int64
 6   reading score                        1000 non-null   int64
 7   writing score                         1000 non-null   int64
dtypes: int64(3), object(5)
memory usage: 62.6+ KB
```

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

```
[15]:
```

	math_score	reading_score	writing_score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000

75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

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

```
[4]:   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           72           74
1        completed        69           90           88
2              none        90           95           93
3              none        47           57           44
4              none        76           78           75
```

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

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

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

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

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

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

```
gender
female    63.633205
male      68.728216
Name: math_score, dtype: float64
```

```
[22]: mean_math_test_preparation = student.
      ↪groupby(['gender', 'test_preparation_course']).math_score.mean()
print(mean_math_test_preparation)
```

```
gender  test_preparation_course
female  completed              67.195652
        none              61.670659
male    completed              72.339080
```

```
        none                66.688312
Name: math_score, dtype: float64
```

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

```
[23]: array([ 72,  69,  90,  47,  76,  71,  88,  40,  64,  38,  58,  65,  78,
          50,  18,  46,  54,  66,  44,  74,  73,  67,  70,  62,  63,  56,
          97,  81,  75,  57,  55,  53,  59,  82,  77,  33,  52,   0,  79,
          39,  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], dtype=int64)
```

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

	count	mean	std	min	25%	50%	75%	max
gender								
female	518.0	63.633205	15.491453	0.0	54.0	65.0	74.0	100.0
male	482.0	68.728216	14.356277	27.0	59.0	69.0	79.0	100.0

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

```
[26]: data = [1,2,3,4,5,6]
      st.mean(data)
```

```
[26]: 3.5
```

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

```
[27]: 3.5
```

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

```
[29]: 1
```

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

```
[30]: 7
```

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

```
[31]: 7.6
```

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

```
[32]: 7.6
```

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

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

```
[34]: 0    4.4
dtype: float64
```

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

```
[35]: 0
0    7
```

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

```
[36]: 0    4.5
dtype: float64
```

```
[38]: df1 = pd.read_csv("E:\\TRINITY ACADEMY OF ENGINEERING PUNE\\TE_
↪2022-23\\assignment\\dsbda\\csv\\housing.csv\\housing.csv")
df1
```

```
[38]:
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms
0	-122.23	37.88	41.0	880.0	129.0 \
1	-122.22	37.86	21.0	7099.0	1106.0
2	-122.24	37.85	52.0	1467.0	190.0
3	-122.25	37.85	52.0	1274.0	235.0
4	-122.25	37.85	52.0	1627.0	280.0
...
20635	-121.09	39.48	25.0	1665.0	374.0
20636	-121.21	39.49	18.0	697.0	150.0
20637	-121.22	39.43	17.0	2254.0	485.0
20638	-121.32	39.43	18.0	1860.0	409.0
20639	-121.24	39.37	16.0	2785.0	616.0

	population	households	median_income	median_house_value
0	322.0	126.0	8.3252	452600.0 \
1	2401.0	1138.0	8.3014	358500.0
2	496.0	177.0	7.2574	352100.0
3	558.0	219.0	5.6431	341300.0
4	565.0	259.0	3.8462	342200.0
...
20635	845.0	330.0	1.5603	78100.0
20636	356.0	114.0	2.5568	77100.0
20637	1007.0	433.0	1.7000	92300.0
20638	741.0	349.0	1.8672	84700.0
20639	1387.0	530.0	2.3886	89400.0

```

        ocean_proximity
0      NEAR BAY
1      NEAR BAY
2      NEAR BAY
3      NEAR BAY
4      NEAR BAY
...
20635    INLAND
20636    INLAND
20637    INLAND
20638    INLAND
20639    INLAND

```

[20640 rows x 10 columns]

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

```
[40]: 499.5396802325581
```

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

```
[41]: 409.0
```

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

```
[42]: 0      306.0
      Name: households, dtype: float64
```

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

```
[43]: 146176.03990028054
```

```
[50]: import pandas as pd
      data = pd.read_csv("E:\\TRINITY ACADEMY OF ENGINEERING PUNE\\TE_
      ↪2022-23\\assignment\\dsbda\\csv\\iris.csv")
      print('Iris-setosa')
```

Iris-setosa

```
[53]: setosa = data['Species'] == 'Iris-setosa'
      print(data[setosa].describe())
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.00000	50.000000	50.000000	50.00000
mean	25.50000	5.00600	3.418000	1.464000	0.24400
std	14.57738	0.35249	0.381024	0.173511	0.10721
min	1.00000	4.30000	2.300000	1.000000	0.10000
25%	13.25000	4.80000	3.125000	1.400000	0.20000
50%	25.50000	5.00000	3.400000	1.500000	0.20000

75%	37.75000	5.20000	3.675000	1.575000	0.30000
max	50.00000	5.80000	4.400000	1.900000	0.60000

```
[55]: print('\nIris-versicolor')
      setosa = data['Species'] == 'Iris-versicolor'
      print(data[setosa].describe())
```

```
Iris-versicolor
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count  50.00000      50.000000      50.000000      50.000000      50.000000
mean   75.50000       5.936000       2.770000       4.260000       1.326000
std    14.57738       0.516171       0.313798       0.469911       0.197753
min    51.00000       4.900000       2.000000       3.000000       1.000000
25%    63.25000       5.600000       2.525000       4.000000       1.200000
50%    75.50000       5.900000       2.800000       4.350000       1.300000
75%    87.75000       6.300000       3.000000       4.600000       1.500000
max    100.00000      7.000000       3.400000       5.100000       1.800000
```

```
[56]: print('\nIris-virginica')
      setosa = data['Species'] == 'Iris-virginica'
      print(data[setosa].describe())
```

```
Iris-virginica
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count  50.00000      50.00000      50.000000      50.000000      50.00000
mean   125.50000       6.58800       2.974000       5.552000       2.02600
std    14.57738       0.63588       0.322497       0.551895       0.27465
min    101.00000       4.90000       2.200000       4.500000       1.40000
25%    113.25000       6.22500       2.800000       5.100000       1.80000
50%    125.50000       6.50000       3.000000       5.550000       2.00000
75%    137.75000       6.90000       3.175000       5.875000       2.30000
max    150.00000       7.90000       3.800000       6.900000       2.50000
```

```
[ ]:
```