

practice-assignment-3

May 4, 2025

```
[5]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[6]: df=pd.read_csv("CustData.csv")
df
```

```
[6]:
```

	CustomerID	Gender	Age	Income
0	1	Male	43	41
1	2	Male	32	43
2	3	Female	26	31
3	4	Female	21	43
4	5	Female	22	37
5	6	Male	42	24
6	7	Male	48	14
7	8	Male	34	37
8	9	Male	21	41
9	10	Female	46	37
10	11	Female	36	40
11	12	Female	23	48
12	13	Male	21	25
13	14	Male	18	45
14	15	Male	41	19
15	16	Male	19	20
16	17	Female	40	43
17	18	Female	33	19
18	19	Female	47	15
19	20	Female	35	30
20	21	Male	27	38
21	22	Male	20	30
22	23	Male	38	40
23	24	Male	50	46
24	25	Female	35	46
25	26	Female	18	32
26	27	Female	18	31
27	28	Male	46	13

28	29	Male	23	26
29	30	Male	34	26

```
[7]: print(df.head())
      print(df.tail())
```

	CustomerID	Gender	Age	Income
0	1	Male	43	41
1	2	Male	32	43
2	3	Female	26	31
3	4	Female	21	43
4	5	Female	22	37

	CustomerID	Gender	Age	Income
25	26	Female	18	32
26	27	Female	18	31
27	28	Male	46	13
28	29	Male	23	26
29	30	Male	34	26

```
[8]: print(df.describe())
      print(df.isna())
```

	CustomerID	Age	Income
count	30.000000	30.000000	30.000000
mean	15.500000	31.900000	32.666667
std	8.803408	10.551516	10.535108
min	1.000000	18.000000	13.000000
25%	8.250000	21.250000	25.250000
50%	15.500000	33.500000	34.500000
75%	22.750000	40.750000	41.000000
max	30.000000	50.000000	48.000000

	CustomerID	Gender	Age	Income
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
5	False	False	False	False
6	False	False	False	False
7	False	False	False	False
8	False	False	False	False
9	False	False	False	False
10	False	False	False	False
11	False	False	False	False
12	False	False	False	False
13	False	False	False	False
14	False	False	False	False
15	False	False	False	False

```

16      False  False  False  False
17      False  False  False  False
18      False  False  False  False
19      False  False  False  False
20      False  False  False  False
21      False  False  False  False
22      False  False  False  False
23      False  False  False  False
24      False  False  False  False
25      False  False  False  False
26      False  False  False  False
27      False  False  False  False
28      False  False  False  False
29      False  False  False  False

```

```
[9]: df.dtypes
```

```

[9]: CustomerID      int64
     Gender          object
     Age            int64
     Income          int64
     dtype: object

```

```

[10]: print(df.loc[:, "CustomerID"].mean())
      print(df.loc[:, "CustomerID"].median())
      print(df.loc[:, "CustomerID"].var())
      print(df.loc[:, "CustomerID"].std())
      print(df.mode())

```

```

15.5
15.5
77.5
8.803408430829505

```

	CustomerID	Gender	Age	Income
0	1	Male	18.0	37.0
1	2	NaN	21.0	43.0
2	3	NaN	NaN	NaN
3	4	NaN	NaN	NaN
4	5	NaN	NaN	NaN
5	6	NaN	NaN	NaN
6	7	NaN	NaN	NaN
7	8	NaN	NaN	NaN
8	9	NaN	NaN	NaN
9	10	NaN	NaN	NaN
10	11	NaN	NaN	NaN
11	12	NaN	NaN	NaN
12	13	NaN	NaN	NaN
13	14	NaN	NaN	NaN

14	15	NaN	NaN	NaN
15	16	NaN	NaN	NaN
16	17	NaN	NaN	NaN
17	18	NaN	NaN	NaN
18	19	NaN	NaN	NaN
19	20	NaN	NaN	NaN
20	21	NaN	NaN	NaN
21	22	NaN	NaN	NaN
22	23	NaN	NaN	NaN
23	24	NaN	NaN	NaN
24	25	NaN	NaN	NaN
25	26	NaN	NaN	NaN
26	27	NaN	NaN	NaN
27	28	NaN	NaN	NaN
28	29	NaN	NaN	NaN
29	30	NaN	NaN	NaN

```
[11]: print(df.loc[:, "Income"].mean())
print(df.loc[:, "Income"].median())
print(df.loc[:, "Income"].var())
print(df.loc[:, "Income"].std())
print(df.mode())
```

```
32.666666666666664
34.5
110.98850574712645
10.5351082456293
```

	CustomerID	Gender	Age	Income
0	1	Male	18.0	37.0
1	2	NaN	21.0	43.0
2	3	NaN	NaN	NaN
3	4	NaN	NaN	NaN
4	5	NaN	NaN	NaN
5	6	NaN	NaN	NaN
6	7	NaN	NaN	NaN
7	8	NaN	NaN	NaN
8	9	NaN	NaN	NaN
9	10	NaN	NaN	NaN
10	11	NaN	NaN	NaN
11	12	NaN	NaN	NaN
12	13	NaN	NaN	NaN
13	14	NaN	NaN	NaN
14	15	NaN	NaN	NaN
15	16	NaN	NaN	NaN
16	17	NaN	NaN	NaN
17	18	NaN	NaN	NaN
18	19	NaN	NaN	NaN
19	20	NaN	NaN	NaN

20	21	NaN	NaN	NaN
21	22	NaN	NaN	NaN
22	23	NaN	NaN	NaN
23	24	NaN	NaN	NaN
24	25	NaN	NaN	NaN
25	26	NaN	NaN	NaN
26	27	NaN	NaN	NaN
27	28	NaN	NaN	NaN
28	29	NaN	NaN	NaN
29	30	NaN	NaN	NaN

```
[12]: print(df.min())
      print(df.max())
```

```
CustomerID      1
Gender          Female
Age             18
Income          13
dtype: object
CustomerID      30
Gender          Male
Age             50
Income          48
dtype: object
```

```
[13]: df
```

```
[13]:
```

	CustomerID	Gender	Age	Income
0	1	Male	43	41
1	2	Male	32	43
2	3	Female	26	31
3	4	Female	21	43
4	5	Female	22	37
5	6	Male	42	24
6	7	Male	48	14
7	8	Male	34	37
8	9	Male	21	41
9	10	Female	46	37
10	11	Female	36	40
11	12	Female	23	48
12	13	Male	21	25
13	14	Male	18	45
14	15	Male	41	19
15	16	Male	19	20
16	17	Female	40	43
17	18	Female	33	19
18	19	Female	47	15

19	20	Female	35	30
20	21	Male	27	38
21	22	Male	20	30
22	23	Male	38	40
23	24	Male	50	46
24	25	Female	35	46
25	26	Female	18	32
26	27	Female	18	31
27	28	Male	46	13
28	29	Male	23	26
29	30	Male	34	26

```
[14]: print(df.groupby(['Gender']).mean())
print(df.groupby(['Gender']).median())
print(df.groupby(['Gender'])['Age'].mean())
print(df.groupby(['Gender'])['Age'].median())
print(df.groupby(['Gender'])['Age'].std())
print(df.groupby(['Gender'])['Age'].var())
```

	CustomerID	Age	Income
Gender			
Female	15.153846	30.769231	34.769231
Male	15.764706	32.764706	31.058824

	CustomerID	Age	Income
Gender			
Female	17.0	33.0	37.0
Male	15.0	34.0	30.0

Gender	
Female	30.769231
Male	32.764706

Name: Age, dtype: float64

Gender	
Female	33.0
Male	34.0

Name: Age, dtype: float64

Gender	
Female	10.125495
Male	11.093520

Name: Age, dtype: float64

Gender	
Female	102.525641
Male	123.066176

Name: Age, dtype: float64

```
[15]: print(df.groupby(['Gender'])['Age'].max())
print(df.groupby(['Gender'])['Age'].min())
```

Gender

```
Female      47
Male        50
Name: Age, dtype: int64
Gender
Female      18
Male        18
Name: Age, dtype: int64
```

```
[16]: print(df.groupby(['Gender'])['Age'].quantile(0.25))
      print(df.groupby(['Gender'])['Age'].quantile(0.5))
      print(df.groupby(['Gender'])['Age'].quantile(0.75))
```

```
Gender
Female      22.0
Male        21.0
Name: Age, dtype: float64
Gender
Female      33.0
Male        34.0
Name: Age, dtype: float64
Gender
Female      36.0
Male        42.0
Name: Age, dtype: float64
```

```
[17]: print(df.groupby(['Gender'])['Income'].mean())
      print(df.groupby(['Gender'])['Income'].median())
      print(df.groupby(['Gender'])['Income'].std())
      print(df.groupby(['Gender'])['Income'].var())
```

```
Gender
Female      34.769231
Male        31.058824
Name: Income, dtype: float64
Gender
Female      37.0
Male        30.0
Name: Income, dtype: float64
Gender
Female      9.883942
Male        11.025372
Name: Income, dtype: float64
Gender
Female      97.692308
Male        121.558824
Name: Income, dtype: float64
```

```
[32]: print(df.groupby(['Gender'])['Income'].max())
print(df.groupby(['Gender'])['Income'].quantile(0.25))
print(df.groupby(['Gender'])['Income'].quantile(0.5))
print(df.groupby(['Gender'])['Income'].quantile(0.75))
```

```
Gender
Female      48
Male        46
Name: Income, dtype: int64
Gender
Female      31.0
Male        24.0
Name: Income, dtype: float64
Gender
Female      37.0
Male        30.0
Name: Income, dtype: float64
Gender
Female      43.0
Male        41.0
Name: Income, dtype: float64
```

```
[34]: print(df.groupby(['Age'])['Income'].mean())
print(df.groupby(['Age'])['Income'].median())
print(df.groupby(['Age'])['Income'].var())
print(df.groupby(['Age'])['Income'].std())
```

```
Age
18      36.000000
19      20.000000
20      30.000000
21      36.333333
22      37.000000
23      37.000000
26      31.000000
27      38.000000
32      43.000000
33      19.000000
34      31.500000
35      38.000000
36      40.000000
38      40.000000
40      43.000000
41      19.000000
42      24.000000
43      41.000000
46      25.000000
47      15.000000
```



```

48    14.000000
50    46.000000
Name: Income, dtype: float64
Age
18    32.0
19    20.0
20    30.0
21    41.0
22    37.0
23    37.0
26    31.0
27    38.0
32    43.0
33    19.0
34    31.5
35    38.0
36    40.0
38    40.0
40    43.0
41    19.0
42    24.0
43    41.0
46    25.0
47    15.0
48    14.0
50    46.0

```

```

Name: Income, dtype: float64
Age
18    61.000000
19         NaN
20         NaN
21    97.333333
22         NaN
23   242.000000
26         NaN
27         NaN
32         NaN
33         NaN
34    60.500000
35   128.000000
36         NaN
38         NaN
40         NaN
41         NaN
42         NaN
43         NaN
46   288.000000
47         NaN

```

```

48      NaN
50      NaN
Name: Income, dtype: float64
Age
18      7.810250
19      NaN
20      NaN
21      9.865766
22      NaN
23     15.556349
26      NaN
27      NaN
32      NaN
33      NaN
34      7.778175
35     11.313708
36      NaN
38      NaN
40      NaN
41      NaN
42      NaN
43      NaN
46     16.970563
47      NaN
48      NaN
50      NaN
Name: Income, dtype: float64

```

```

[38]: print(df.groupby(['Age'])['Income'].quantile(0.25))
      print(df.groupby(['Age'])['Income'].quantile(0.5))
      print(df.groupby(['Age'])['Income'].quantile(0.75))

```

```

Age
18     31.50
19     20.00
20     30.00
21     33.00
22     37.00
23     31.50
26     31.00
27     38.00
32     43.00
33     19.00
34     28.75
35     34.00
36     40.00
38     40.00
40     43.00

```

```
41    19.00
42    24.00
43    41.00
46    19.00
47    15.00
48    14.00
50    46.00
```

Name: Income, dtype: float64

Age

```
18    32.0
19    20.0
20    30.0
21    41.0
22    37.0
23    37.0
26    31.0
27    38.0
32    43.0
33    19.0
34    31.5
35    38.0
36    40.0
38    40.0
40    43.0
41    19.0
42    24.0
43    41.0
46    25.0
47    15.0
48    14.0
50    46.0
```

Name: Income, dtype: float64

Age

```
18    38.50
19    20.00
20    30.00
21    42.00
22    37.00
23    42.50
26    31.00
27    38.00
32    43.00
33    19.00
34    34.25
35    42.00
36    40.00
38    40.00
40    43.00
```

```

41    19.00
42    24.00
43    41.00
46    31.00
47    15.00
48    14.00
50    46.00
Name: Income, dtype: float64

```

```
[40]: vm=pd.read_csv("Iris.csv")
      vm
```

```
[40]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	\
0	1	5.1	3.5	1.4	0.2	
1	2	4.9	3.0	1.4	0.2	
2	3	4.7	3.2	1.3	0.2	
3	4	4.6	3.1	1.5	0.2	
4	5	5.0	3.6	1.4	0.2	
..	
145	146	6.7	3.0	5.2	2.3	
146	147	6.3	2.5	5.0	1.9	
147	148	6.5	3.0	5.2	2.0	
148	149	6.2	3.4	5.4	2.3	
149	150	5.9	3.0	5.1	1.8	

```

      Species
0      Iris-setosa
1      Iris-setosa
2      Iris-setosa
3      Iris-setosa
4      Iris-setosa
..      ...
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica

```

```
[150 rows x 6 columns]
```

```
[42]: vm.columns
```

```
[42]: Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
          'Species'],
          dtype='object')
```

```
[44]: vm.dtypes
```

```
[44]: Id                int64
      SepalLengthCm    float64
      SepalWidthCm     float64
      PetalLengthCm    float64
      PetalWidthCm     float64
      Species          object
      dtype: object
```

```
[46]: vm.Species.describe()
```

```
[46]: count                150
      unique                3
      top      Iris-setosa
      freq                50
      Name: Species, dtype: object
```

```
[48]: vm.Species.mode()
```

```
[48]: 0      Iris-setosa
      1      Iris-versicolor
      2      Iris-virginica
      Name: Species, dtype: object
```

```
[50]: iSetosa = (vm['Species']== 'Iris-setosa')
      iVersicolor = (vm['Species']== 'Iris-versicolor')
      iVirginica = (vm['Species']== 'Iris-virginica')
```

```
[54]: print(iSetosa.mean())
      print(iSetosa.median())
      print(iSetosa.var())
      print(iSetosa.std())
```

```
0.3333333333333333
0.0
0.22371364653243855
0.4729837698404022
```

```
[56]: print(iVirginica.mean())
      print(iVirginica.median())
      print(iVirginica.var())
      print(iVirginica.std())
```

```
0.3333333333333333
0.0
0.22371364653243844
0.4729837698404021
```

```
[58]: print(iVersicolor.mean())
      print(iVersicolor.median())
      print(iVersicolor.var())
      print(iVersicolor.std())
```

```
0.3333333333333333
0.0
0.2237136465324385
0.47298376984040214
```

```
[62]: print(vm[iVirginica].describe())
      print(vm[iSetosa].describe())
      print(vm[iVersicolor].describe())
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.00000	50.00000	50.00000	50.00000
mean	125.50000	6.58800	2.97400	5.55200	2.02600
std	14.57738	0.63588	0.322497	0.551895	0.27465
min	101.00000	4.90000	2.20000	4.50000	1.40000
25%	113.25000	6.22500	2.80000	5.10000	1.80000
50%	125.50000	6.50000	3.00000	5.55000	2.00000
75%	137.75000	6.90000	3.17500	5.87500	2.30000
max	150.00000	7.90000	3.80000	6.90000	2.50000

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.00000	50.00000	50.00000	50.00000
mean	25.50000	5.00600	3.41800	1.46400	0.24400
std	14.57738	0.35249	0.381024	0.173511	0.10721
min	1.00000	4.30000	2.30000	1.00000	0.10000
25%	13.25000	4.80000	3.12500	1.40000	0.20000
50%	25.50000	5.00000	3.40000	1.50000	0.20000
75%	37.75000	5.20000	3.67500	1.57500	0.30000
max	50.00000	5.80000	4.40000	1.90000	0.60000

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	50.00000	50.00000	50.00000	50.00000	50.00000
mean	75.50000	5.93600	2.77000	4.26000	1.32600
std	14.57738	0.516171	0.313798	0.469911	0.197753
min	51.00000	4.90000	2.00000	3.00000	1.00000
25%	63.25000	5.60000	2.52500	4.00000	1.20000
50%	75.50000	5.90000	2.80000	4.35000	1.30000
75%	87.75000	6.30000	3.00000	4.60000	1.50000
max	100.00000	7.00000	3.40000	5.10000	1.80000

```
[ ]:
```