Lab Assignment no 3 ¶

Aim:Perform the following operations on any open source dataset (e.g., data.csv)

1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.
2. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of ‘Iris-setosa’, ‘Iris-versicolor’ and ‘Iris-versicolor’ of iris.csv dataset.

Provide the codes with outputs and explain everything that you do in this step.

In [27]:

```
import pandas as pd
file_path=r"C:\Users\shrey\OneDrive\Desktop\MALL_CUSTOMER.csv"
df=pd.read_csv(file_path)
df.head()
```

Out[27]:

	CustomerID	Age	Annual Income($)	Spending Score	Gender
0	1	33.0	186.0	56.0	male
1	2	18.0	127.0	26.0	male
2	3	25.0	132.0	37.0	male
3	4	25.0	100.0	63.0	male
4	5	29.0	104.0	42.0	male

In [28]:

```
df
```

Out[28]:

	CustomerID	Age	Annual Income($)	Spending Score	Gender
0	1	33.0	186.0	56.0	male
1	2	18.0	127.0	26.0	male
2	3	25.0	132.0	37.0	male
3	4	25.0	100.0	63.0	male
4	5	29.0	104.0	42.0	male
...	...	...	...	...	...
195	196	25.0	161.0	93.0	male
196	197	25.0	189.0	40.0	male
197	198	33.0	125.0	5.0	male
198	199	19.0	108.0	14.0	male
199	200	34.0	112.0	36.0	male

200 rows × 5 columns

In [29]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   CustomerID         200 non-null    int64  
 1   Age                184 non-null    float64
 2   Annual Income($)   184 non-null    float64
 3   Spending Score     185 non-null    float64
 4   Gender             200 non-null    object 
dtypes: float64(3), int64(1), object(1)
memory usage: 7.9+ KB
```

In [10]:

```
df.shape
```

Out[10]:

```
(200, 5)
```

In [12]:

```
df.head
```

Out[12]:

```
<bound method NDFrame.head of      CustomerID   Age  Annual Income($)  Spe
nding Score Gender
0             1  33.0             186.0            56.0   male
1             2  18.0             127.0            26.0   male
2             3  25.0             132.0            37.0   male
3             4  25.0             100.0            63.0   male
4             5  29.0             104.0            42.0   male
..          ...   ...               ...             ...    ...
195         196  25.0             161.0            93.0   male
196         197  25.0             189.0            40.0   male
197         198  33.0             125.0             5.0   male
198         199  19.0             108.0            14.0   male
199         200  34.0             112.0            36.0   male

[200 rows x 5 columns]>
```

In [13]:

```
df.tail
```

Out[13]:

```
<bound method NDFrame.tail of      CustomerID   Age  Annual Income($)  Spe
nding Score Gender
0             1  33.0             186.0            56.0   male
1             2  18.0             127.0            26.0   male
2             3  25.0             132.0            37.0   male
3             4  25.0             100.0            63.0   male
4             5  29.0             104.0            42.0   male
..          ...   ...               ...             ...    ...
195         196  25.0             161.0            93.0   male
196         197  25.0             189.0            40.0   male
197         198  33.0             125.0             5.0   male
198         199  19.0             108.0            14.0   male
199         200  34.0             112.0            36.0   male

[200 rows x 5 columns]>
```

In [14]:
```
df.describe()
```

Out[14]:

	CustomerID	Age	Annual Income($)	Spending Score
count	200.000000	184.000000	184.000000	185.000000
mean	100.500000	26.342391	148.244565	49.470270
std	57.879185	5.133959	29.339728	28.099985
min	1.000000	18.000000	100.000000	1.000000
25%	50.750000	22.000000	122.000000	26.000000
50%	100.500000	26.000000	150.000000	47.000000
75%	150.250000	30.000000	170.250000	72.000000
max	200.000000	35.000000	200.000000	100.000000

In [15]:
```
df.Age.mean()
```

Out[15]:
```
26.342391304347824
```

In [16]:
```
df.Age.mode()
```

Out[16]:
```
0    30.0
Name: Age, dtype: float64
```

In [17]:
```
df.Age.median()
```

Out[17]:
```
26.0
```

In [18]:
```
df.groupby(['Age']).count()
```

Out[18]:

Age	CustomerID	Annual Income($)	Spending Score	Gender
18.0	15	14	13	15
19.0	12	11	11	12
20.0	3	3	3	3
21.0	8	8	7	8
22.0	13	12	12	13
23.0	9	7	9	9
24.0	5	5	5	5
25.0	16	15	16	16
26.0	14	14	12	14
27.0	12	9	12	12
28.0	6	5	6	6
29.0	10	10	9	10
30.0	18	17	16	18
31.0	10	7	10	10
32.0	8	7	7	8
33.0	5	5	4	5
34.0	9	9	7	9
35.0	11	10	10	11

In [20]:
```
df.groupby(['Gender']).count()
```

Out[20]:

Gender	CustomerID	Age	Annual Income($)	Spending Score
female	20	20	8	20
male	180	164	176	165

In [21]:
```
df.Age.std()
```

Out[21]:
```
5.133959234335101
```

In [24]:
```
df[['Age' , 'Annual Income($)', 'Spending Score']].mean()
```

Out[24]:
```
Age                  26.342391
Annual Income($)    148.244565
Spending Score       49.470270
dtype: float64
```

In [30]:

```
df[['Age' , 'Annual Income($)', 'Spending Score']].mode()
```

Out[30]:

	Age	Annual Income($)	Spending Score
0	30.0	170.0	26.0

In [32]:

```
df[['Age' , 'Annual Income($)', 'Spending Score']].median()
```

Out[32]:

```
Age                  26.0
Annual Income($)    150.0
Spending Score       47.0
dtype: float64
```

In [33]:

```
df[['Age' , 'Annual Income($)', 'Spending Score']].max()
```

Out[33]:

```
Age                  35.0
Annual Income($)    200.0
Spending Score      100.0
dtype: float64
```

In [34]:

```
df[['Age' , 'Annual Income($)', 'Spending Score']].std()
```

Out[34]:

```
Age                  5.133959
Annual Income($)    29.339728
Spending Score      28.099985
dtype: float64
```

In [35]:

```
df2 = df.groupby('Gender')
```

In [36]:

```
df
```

Out[36]:

	CustomerID	Age	Annual Income($)	Spending Score	Gender
0	1	33.0	186.0	56.0	male
1	2	18.0	127.0	26.0	male
2	3	25.0	132.0	37.0	male
3	4	25.0	100.0	63.0	male
4	5	29.0	104.0	42.0	male
...	...	...	...	...	...
195	196	25.0	161.0	93.0	male
196	197	25.0	189.0	40.0	male
197	198	33.0	125.0	5.0	male
198	199	19.0	108.0	14.0	male
199	200	34.0	112.0	36.0	male

200 rows × 5 columns

In [37]:

```python
for Gender, Gender_f in df2:
    print(Gender)
    print(Gender_f)

```

```
female
     CustomerID   Age  Annual Income($)  Spending Score  Gender
5             6  35.0             174.0            68.0  female
6             7  32.0             114.0            71.0  female
7             8  32.0             127.0            49.0  female
8             9  28.0               NaN            19.0  female
9            10  30.0               NaN            58.0  female
10           11  35.0               NaN            34.0  female
11           12  32.0               NaN            17.0  female
12           13  27.0               NaN            18.0  female
13           14  27.0               NaN            26.0  female
14           15  31.0               NaN            65.0  female
15           16  22.0               NaN            39.0  female
16           17  25.0               NaN            65.0  female
17           18  19.0               NaN            89.0  female
18           19  31.0               NaN            76.0  female
22           23  23.0               NaN            93.0  female
28           29  29.0             198.0             4.0  female
33           34  31.0             176.0            30.0  female
56           57  24.0             107.0            74.0  female
94           95  28.0             106.0             9.0  female
172         173  25.0             152.0            93.0  female
male
     CustomerID   Age  Annual Income($)  Spending Score Gender
0             1  33.0             186.0            56.0   male
1             2  18.0             127.0            26.0   male
2             3  25.0             132.0            37.0   male
3             4  25.0             100.0            63.0   male
4             5  29.0             104.0            42.0   male
..          ...   ...               ...             ...    ...
195         196  25.0             161.0            93.0   male
196         197  25.0             189.0            40.0   male
197         198  33.0             125.0             5.0   male
198         199  19.0             108.0            14.0   male
199         200  34.0             112.0            36.0   male

[180 rows x 5 columns]
```

In [39]:

```
df2.get_group('male')
```

Out[39]:

	CustomerID	Age	Annual Income($)	Spending Score	Gender
0	1	33.0	186.0	56.0	male
1	2	18.0	127.0	26.0	male
2	3	25.0	132.0	37.0	male
3	4	25.0	100.0	63.0	male
4	5	29.0	104.0	42.0	male
...	...	...	...	...	...
195	196	25.0	161.0	93.0	male
196	197	25.0	189.0	40.0	male
197	198	33.0	125.0	5.0	male
198	199	19.0	108.0	14.0	male
199	200	34.0	112.0	36.0	male

180 rows × 5 columns

In [41]:

```
df2.get_group('female')
```

Out[41]:

	CustomerID	Age	Annual Income($)	Spending Score	Gender
5	6	35.0	174.0	68.0	female
6	7	32.0	114.0	71.0	female
7	8	32.0	127.0	49.0	female
8	9	28.0	NaN	19.0	female
9	10	30.0	NaN	58.0	female
10	11	35.0	NaN	34.0	female
11	12	32.0	NaN	17.0	female
12	13	27.0	NaN	18.0	female
13	14	27.0	NaN	26.0	female
14	15	31.0	NaN	65.0	female
15	16	22.0	NaN	39.0	female
16	17	25.0	NaN	65.0	female
17	18	19.0	NaN	89.0	female
18	19	31.0	NaN	76.0	female
22	23	23.0	NaN	93.0	female
28	29	29.0	198.0	4.0	female
33	34	31.0	176.0	30.0	female
56	57	24.0	107.0	74.0	female
94	95	28.0	106.0	9.0	female
172	173	25.0	152.0	93.0	female

In [43]:
```
df2[['Age' , 'Annual Income($)', 'Spending Score']].median()
```

Out[43]:

	Age	Annual Income($)	Spending Score
Gender			
female	28.5	139.5	53.5
male	26.0	150.0	47.0

In [44]:
```
df2[['Age' , 'Annual Income($)', 'Spending Score']].mean()
```

Out[44]:

	Age	Annual Income($)	Spending Score
Gender			
female	28.300000	144.250000	49.850000
male	26.103659	148.426136	49.424242

In [45]:
```
df2[['Age' , 'Annual Income($)', 'Spending Score']].max()
```

Out[45]:

	Age	Annual Income($)	Spending Score
Gender			
female	35.0	198.0	93.0
male	35.0	200.0	100.0

In [46]:
```
df2[['Age' , 'Annual Income($)', 'Spending Score']].min()
```

Out[46]:

	Age	Annual Income($)	Spending Score
Gender			
female	19.0	106.0	4.0
male	18.0	100.0	1.0

In [47]:
```
df2[['Age' , 'Annual Income($)', 'Spending Score']].std()
```

Out[47]:

	Age	Annual Income($)	Spending Score
Gender			
female	4.317650	35.668113	28.995962
male	5.185656	29.129371	28.079841

In [51]:
```
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/i
```

In [49]:
```
df3 = pd.read_csv(url)
```

In [50]:
```
df3
```

Out[50]:

	5.1	3.5	1.4	0.2	Iris-setosa
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa
...	...	...	...	...	...
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

149 rows × 5 columns

In [52]:
```
df3.columns=("A" , "B" , " C " , "D" , "E")
```

In [53]:
```
df3
```

Out[53]:

	A	B	C	D	E
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa
...	...	...	...	...	...
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

149 rows × 5 columns

In [54]:
```
df4 =df3.groupby("E")
```

In [55]:
```
df4
```

Out[55]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001A686244F10
>
```

In [56]:

```
df4.get_group("Iris-setosa")
```

Out[56]:

	A	B	C	D	E
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa
5	4.6	3.4	1.4	0.3	Iris-setosa
6	5.0	3.4	1.5	0.2	Iris-setosa
7	4.4	2.9	1.4	0.2	Iris-setosa
8	4.9	3.1	1.5	0.1	Iris-setosa
9	5.4	3.7	1.5	0.2	Iris-setosa
10	4.8	3.4	1.6	0.2	Iris-setosa
11	4.8	3.0	1.4	0.1	Iris-setosa
12	4.3	3.0	1.1	0.1	Iris-setosa
13	5.8	4.0	1.2	0.2	Iris-setosa
14	5.7	4.4	1.5	0.4	Iris-setosa
15	5.4	3.9	1.3	0.4	Iris-setosa
16	5.1	3.5	1.4	0.3	Iris-setosa
17	5.7	3.8	1.7	0.3	Iris-setosa
18	5.1	3.8	1.5	0.3	Iris-setosa
19	5.4	3.4	1.7	0.2	Iris-setosa
20	5.1	3.7	1.5	0.4	Iris-setosa
21	4.6	3.6	1.0	0.2	Iris-setosa
22	5.1	3.3	1.7	0.5	Iris-setosa
23	4.8	3.4	1.9	0.2	Iris-setosa
24	5.0	3.0	1.6	0.2	Iris-setosa
25	5.0	3.4	1.6	0.4	Iris-setosa
26	5.2	3.5	1.5	0.2	Iris-setosa
27	5.2	3.4	1.4	0.2	Iris-setosa
28	4.7	3.2	1.6	0.2	Iris-setosa
29	4.8	3.1	1.6	0.2	Iris-setosa
30	5.4	3.4	1.5	0.4	Iris-setosa
31	5.2	4.1	1.5	0.1	Iris-setosa
32	5.5	4.2	1.4	0.2	Iris-setosa
33	4.9	3.1	1.5	0.1	Iris-setosa
34	5.0	3.2	1.2	0.2	Iris-setosa
35	5.5	3.5	1.3	0.2	Iris-setosa
36	4.9	3.1	1.5	0.1	Iris-setosa
37	4.4	3.0	1.3	0.2	Iris-setosa
38	5.1	3.4	1.5	0.2	Iris-setosa

	A	B	C	D	E
39	5.0	3.5	1.3	0.3	Iris-setosa
40	4.5	2.3	1.3	0.3	Iris-setosa
41	4.4	3.2	1.3	0.2	Iris-setosa
42	5.0	3.5	1.6	0.6	Iris-setosa
43	5.1	3.8	1.9	0.4	Iris-setosa
44	4.8	3.0	1.4	0.3	Iris-setosa
45	5.1	3.8	1.6	0.2	Iris-setosa
46	4.6	3.2	1.4	0.2	Iris-setosa
47	5.3	3.7	1.5	0.2	Iris-setosa
48	5.0	3.3	1.4	0.2	Iris-setosa

In [57]:

```
df4.get_group("Iris-virginica")
```

Out[57]:

	A	B	C	D	E
99	6.3	3.3	6.0	2.5	Iris-virginica
100	5.8	2.7	5.1	1.9	Iris-virginica
101	7.1	3.0	5.9	2.1	Iris-virginica
102	6.3	2.9	5.6	1.8	Iris-virginica
103	6.5	3.0	5.8	2.2	Iris-virginica
104	7.6	3.0	6.6	2.1	Iris-virginica
105	4.9	2.5	4.5	1.7	Iris-virginica
106	7.3	2.9	6.3	1.8	Iris-virginica
107	6.7	2.5	5.8	1.8	Iris-virginica
108	7.2	3.6	6.1	2.5	Iris-virginica
109	6.5	3.2	5.1	2.0	Iris-virginica
110	6.4	2.7	5.3	1.9	Iris-virginica
111	6.8	3.0	5.5	2.1	Iris-virginica
112	5.7	2.5	5.0	2.0	Iris-virginica
113	5.8	2.8	5.1	2.4	Iris-virginica
114	6.4	3.2	5.3	2.3	Iris-virginica
115	6.5	3.0	5.5	1.8	Iris-virginica
116	7.7	3.8	6.7	2.2	Iris-virginica
117	7.7	2.6	6.9	2.3	Iris-virginica
118	6.0	2.2	5.0	1.5	Iris-virginica
119	6.9	3.2	5.7	2.3	Iris-virginica
120	5.6	2.8	4.9	2.0	Iris-virginica
121	7.7	2.8	6.7	2.0	Iris-virginica
122	6.3	2.7	4.9	1.8	Iris-virginica
123	6.7	3.3	5.7	2.1	Iris-virginica
124	7.2	3.2	6.0	1.8	Iris-virginica
125	6.2	2.8	4.8	1.8	Iris-virginica
126	6.1	3.0	4.9	1.8	Iris-virginica
127	6.4	2.8	5.6	2.1	Iris-virginica
128	7.2	3.0	5.8	1.6	Iris-virginica
129	7.4	2.8	6.1	1.9	Iris-virginica
130	7.9	3.8	6.4	2.0	Iris-virginica
131	6.4	2.8	5.6	2.2	Iris-virginica
132	6.3	2.8	5.1	1.5	Iris-virginica
133	6.1	2.6	5.6	1.4	Iris-virginica
134	7.7	3.0	6.1	2.3	Iris-virginica
135	6.3	3.4	5.6	2.4	Iris-virginica
136	6.4	3.1	5.5	1.8	Iris-virginica
137	6.0	3.0	4.8	1.8	Iris-virginica

	A	B	C	D	E
138	6.9	3.1	5.4	2.1	Iris-virginica
139	6.7	3.1	5.6	2.4	Iris-virginica
140	6.9	3.1	5.1	2.3	Iris-virginica
141	5.8	2.7	5.1	1.9	Iris-virginica
142	6.8	3.2	5.9	2.3	Iris-virginica
143	6.7	3.3	5.7	2.5	Iris-virginica
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

In [58]:

```
df4.mean()
```

Out[58]:

	A	B	C	D
E				
Iris-setosa	5.004082	3.416327	1.465306	0.244898
Iris-versicolor	5.936000	2.770000	4.260000	1.326000
Iris-virginica	6.588000	2.974000	5.552000	2.026000

In [59]:

```
df4.std()
```

Out[59]:

	A	B	C	D
E				
Iris-setosa	0.355879	0.384787	0.175061	0.108130
Iris-versicolor	0.516171	0.313798	0.469911	0.197753
Iris-virginica	0.635880	0.322497	0.551895	0.274650

Name:Devesh Kashikar
Roll.no:13217

In [ ]:

```

```