Madhur Jaripatke

Roll No. 48

TE A Computer

RMDSSOE, Warje, Pune

3. Descriptive Statistics - Measures of Central Tendency and variability

Perform the following operations on any open source dataset (e.g., data.csv) the age groups. Create a list that contains a numeric value for each response to the

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

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 [1]:

```
import pandas as pd
from sklearn import preprocessing
```

Importing the dataset

In [2]:

```
df = pd.read_csv('Datasets/Mall_Customers.csv')
df
```

Out[2]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

Exploratory Data Analysis

In [3]:

```
df.describe()
```

Out[3]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

In [4]:

```
df.min()
```

Out[4]:

```
CustomerID                     1
Gender                    Female
Age                           18
Annual Income (k$)            15
Spending Score (1-100)         1
dtype: object
```

In [5]:

```
df.groupby(['Gender'])['Age'].mean()
```

Out[5]:

```
Gender
Female    38.098214
Male      39.806818
Name: Age, dtype: float64
```

In [6]:

```
df.groupby(['Gender'])['Age'].median()
```

Out[6]:

```
Gender
Female    35.0
Male      37.0
Name: Age, dtype: float64
```

In [7]:

```
df.groupby(['Gender'])['Age'].std()
```

Out[7]:

```
Gender
Female    12.644095
Male      15.514812
Name: Age, dtype: float64
```

In [8]:

```
df.groupby(['Gender'])['Annual Income (k$)'].mean()
```

Out[8]:

```
Gender
Female    59.250000
Male      62.227273
Name: Annual Income (k$), dtype: float64
```

In [9]:

```
df.groupby(['Gender'])['Annual Income (k$)'].median()
```

Out[9]:

```
Gender
Female    60.0
Male      62.5
Name: Annual Income (k$), dtype: float64
```

In [10]:

```
df.groupby(['Gender'])['Annual Income (k$)'].std()
```

Out[10]:

```
Gender
Female    26.011952
Male      26.638373
Name: Annual Income (k$), dtype: float64
```

In [11]:

```
df.groupby(['Gender'])['Age'].median()
```

Out[11]:

```
Gender
Female    35.0
Male      37.0
Name: Age, dtype: float64
```

In [12]:

```
df.groupby(['Gender']).mean()
```

Out[12]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
Gender				
Female	97.562500	38.098214	59.250000	51.526786
Male	104.238636	39.806818	62.227273	48.511364

In [13]:

```
df.groupby(['Gender']).median()
```

Out[13]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
Gender				
Female	94.5	35.0	60.0	50.0
Male	106.5	37.0	62.5	50.0

In [14]:

```
df.groupby(['Gender']).min()
```

Out[14]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
Gender				
Female	3	18	16	5
Male	1	18	15	1

In [15]:

```
df.groupby(['Gender']).max()
```

Out[15]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
Gender				
Female	197	68	126	99
Male	200	70	137	97

In [16]:

```
x = df.drop(axis=1, columns=['Gender'])
x
```

Out[16]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
0	1	19	15	39
1	2	21	15	81
2	3	20	16	6
3	4	23	16	77
4	5	31	17	40
...	...	...	...	...
195	196	35	120	79
196	197	45	126	28
197	198	32	126	74
198	199	32	137	18
199	200	30	137	83

200 rows × 4 columns

Encoding

In [17]:

```python
enc = preprocessing.OneHotEncoder()
enc_df = pd.DataFrame(enc.fit_transform(df[['Gender']]).toarray())
enc_df
```

Out[17]:

	0	1
0	0.0	1.0
1	0.0	1.0
2	1.0	0.0
3	1.0	0.0
4	1.0	0.0
...	...	...
195	1.0	0.0
196	1.0	0.0
197	0.0	1.0
198	0.0	1.0
199	0.0	1.0

200 rows × 2 columns

In [18]:

```python
df_encode = x.join(enc_df)
df_encode
```

Out[18]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)	0	1
0	1	19	15	39	0.0	1.0
1	2	21	15	81	0.0	1.0
2	3	20	16	6	1.0	0.0
3	4	23	16	77	1.0	0.0
4	5	31	17	40	1.0	0.0
...	...	...	...	...	...	...
195	196	35	120	79	1.0	0.0
196	197	45	126	28	1.0	0.0
197	198	32	126	74	0.0	1.0
198	199	32	137	18	0.0	1.0
199	200	30	137	83	0.0	1.0

200 rows × 6 columns

In [19]:

```python
df1 = pd.read_csv('Datasets/Iris.csv')
df1
```

Out[19]:

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

150 rows × 6 columns

In [20]:

```
df1.describe()
```

Out[20]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [21]:

```
set1 = (df1['Species'] == 'Iris-virginica')
print(df1[set1].describe())
```

```
              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
```

In [22]:

```
set2 = (df1['Species'] == 'Iris-versicolor')
print(df1[set2].describe())
```

```
              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
```

In [23]:

```
set3 = (df1['Species'] == 'Iris-setosa')
print(df1[set3].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
```

In [24]:

```
df1['Species'].unique()
```

Out[24]:

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

Grouping

In [25]:

```
df1.groupby(['Species']).mean()
```

Out[25]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Species					
Iris-setosa	25.5	5.006	3.418	1.464	0.244
Iris-versicolor	75.5	5.936	2.770	4.260	1.326
Iris-virginica	125.5	6.588	2.974	5.552	2.026

In [26]:

```
df1.groupby(['Species']).median()
```

Out[26]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Species					
Iris-setosa	25.5	5.0	3.4	1.50	0.2
Iris-versicolor	75.5	5.9	2.8	4.35	1.3
Iris-virginica	125.5	6.5	3.0	5.55	2.0

In [27]:

```
df1.groupby(['Species']).std()
```

Out[27]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Species					
Iris-setosa	14.57738	0.352490	0.381024	0.173511	0.107210
Iris-versicolor	14.57738	0.516171	0.313798	0.469911	0.197753
Iris-virginica	14.57738	0.635880	0.322497	0.551895	0.274650