customer-dataset-eda

March 17, 2024

[5]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
#This dataset is downloaded from kaggle
#Source link: https://www.kaggle.com/datasets/ayushparwal2026/shopping-dataset/
 ↪code
```

[2]:
```
os.chdir("C:\\Users\\Nithilesh M\\Downloads")
```

[3]:
```
df=pd.read_csv("Shopping_data.csv")
df
```

[3]:

	CustomerID	Genre	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 x 5 columns]

[20]:
```
x=df["Age"]
x
```

[20]:
```
0      19
1      21
2      20
3      23
4      31
       ..
```

```
195    35
196    45
197    32
198    32
199    30
Name: Age, Length: 200, dtype: int64
```

[21]:
```
y=df["Annual Income (k$)"]
y
```

[21]:
```
0       15
1       15
2       16
3       16
4       17
      ...
195    120
196    126
197    126
198    137
199    137
Name: Annual Income (k$), Length: 200, dtype: int64
```

[19]:
```
pd.isnull(df).sum()
```

[19]:
```
CustomerID                0
Genre                     0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

[35]:
```
Spending_score=df["Spending Score (1-100)"]
```

[48]:
```
df.dtypes
```

[48]:
```
CustomerID                 int64
Genre                     object
Age                        int64
Annual Income (k$)         int64
Spending Score (1-100)     int64
dtype: object
```

[49]:
```
df["Age"]
```

[49]:
```
0      19
1      21
```

```
2      20
3      23
4      31
       ..
195    35
196    45
197    32
198    32
199    30
Name: Age, Length: 200, dtype: int64
```

[50]:
```
df["Annual Income (k$)"]
```

[50]:
```
0       15
1       15
2       16
3       16
4       17
      …
195    120
196    126
197    126
198    137
199    137
Name: Annual Income (k$), Length: 200, dtype: int64
```

[52]:
```
df["Spending Score (1-100)"]
```

[52]:
```
0      39
1      81
2       6
3      77
4      40
       ..
195    79
196    28
197    74
198    18
199    83
Name: Spending Score (1-100), Length: 200, dtype: int64
```

[53]:
```
df[['Age',"Spending Score (1-100)"]]#access 2 columns
```

[53]:
```
     Age  Spending Score (1-100)
0     19                      39
1     21                      81
2     20                       6
```

```
3     23                      77
4     31                      40
..   …                       …
195   35                      79
196   45                      28
197   32                      74
198   32                      18
199   30                      83

[200 rows x 2 columns]
```

[54]:
```
df.iloc[7:12]#access 5 rows and columns from 7 to 11
```

[54]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72
10	11	Male	67	19	14
11	12	Female	35	19	99

[55]:
```
df.describe()#gives 8 important statistical data
```

[55]:

	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

[56]:
```
df.head() #gives first 5 inputs of file
```

[56]:

	CustomerID	Genre	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

[57]:
```
df.tail() #gives last 5 inputs of file
```

[57]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
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
```

[58]:
```
df.columns #gives the names of columns
```

[58]:
```
Index(['CustomerID', 'Genre', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

[59]:
```
df["Age"].unique() #remove duplicates and print names of a variable
```

[59]:
```
array([19, 21, 20, 23, 31, 22, 35, 64, 30, 67, 58, 24, 37, 52, 25, 46, 54,
       29, 45, 40, 60, 53, 18, 49, 42, 36, 65, 48, 50, 27, 33, 59, 47, 51,
       69, 70, 63, 43, 68, 32, 26, 57, 38, 55, 34, 66, 39, 44, 28, 56, 41],
      dtype=int64)
```

[60]:
```
df['Annual Income (k$)'].mean()
```

[60]:
```
60.56
```

[62]:
```
df.shape#prints rows and columns
```

[62]:
```
(200, 5)
```

[63]:
```
df['Annual Income (k$)'].value_counts()
```

[63]:
```
Annual Income (k$)
54     12
78     12
48      6
71      6
63      6
       ..
58      2
59      2
16      2
64      2
137     2
Name: count, Length: 64, dtype: int64
```

[65]:
```
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   Genre                   200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

[66]:
```
df.drop(["CustomerID"],axis=1) #.drop([] ,axis=1) used to drop a column
```

[66]:
```
       Genre  Age  Annual Income (k$)  Spending Score (1-100)
0       Male   19                  15                      39
1       Male   21                  15                      81
2     Female   20                  16                       6
3     Female   23                  16                      77
4     Female   31                  17                      40
..       ...  ...                 ...                     ...
195   Female   35                 120                      79
196   Female   45                 126                      28
197     Male   32                 126                      74
198     Male   32                 137                      18
199     Male   30                 137                      83

[200 rows x 4 columns]
```

[67]:
```
pd.isnull(df)#checking blank data
```

[67]:
```
     CustomerID  Genre    Age  Annual Income (k$)  Spending Score (1-100)
0         False  False  False               False                   False
1         False  False  False               False                   False
2         False  False  False               False                   False
3         False  False  False               False                   False
4         False  False  False               False                   False
..          ...    ...    ...                 ...                     ...
195       False  False  False               False                   False
196       False  False  False               False                   False
197       False  False  False               False                   False
198       False  False  False               False                   False
199       False  False  False               False                   False

[200 rows x 5 columns]
```

[69]:
```
df.iloc[15:44, 2:4] #row column slicing
```

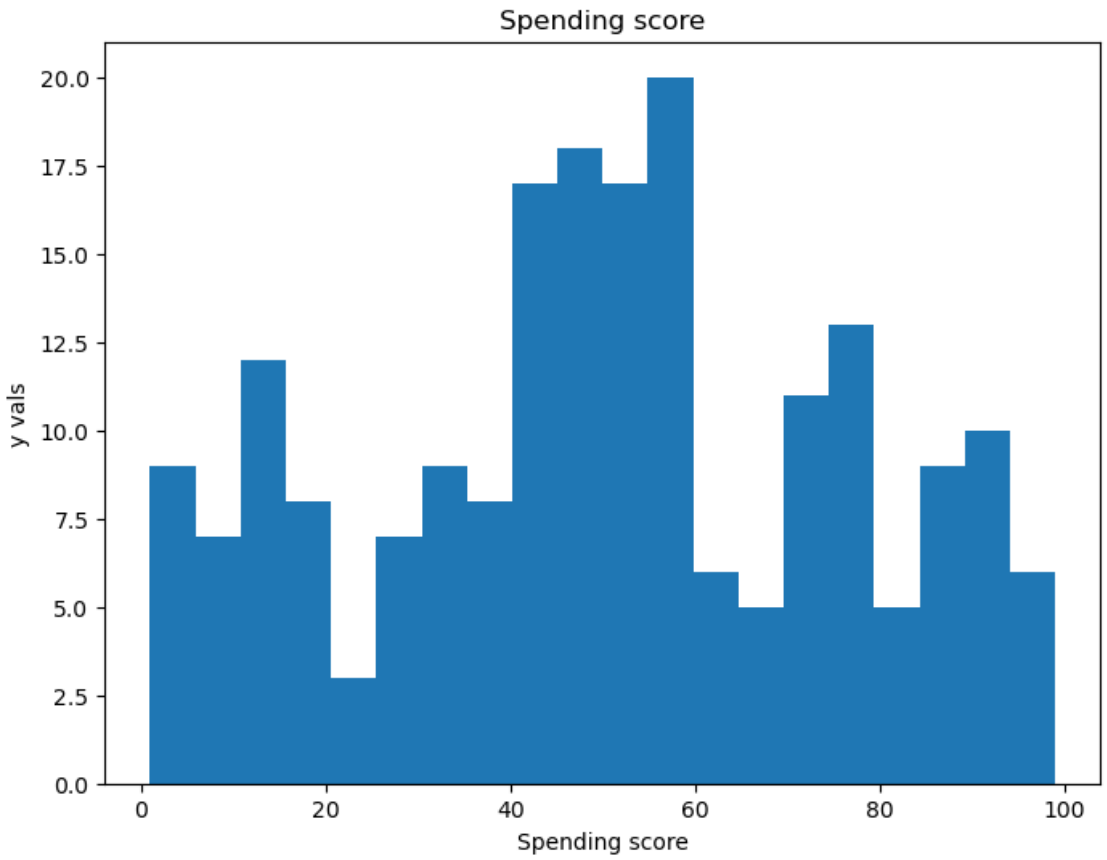
[69]:
```
    Age  Annual Income (k$)
15   22                  20
16   35                  21
17   20                  21
```

```
18   52                  23
19   35                  23
20   35                  24
21   25                  24
22   46                  25
23   31                  25
24   54                  28
25   29                  28
26   45                  28
27   35                  28
28   40                  29
29   23                  29
30   60                  30
31   21                  30
32   53                  33
33   18                  33
34   49                  33
35   21                  33
36   42                  34
37   30                  34
38   36                  37
39   20                  37
40   65                  38
41   24                  38
42   48                  39
43   31                  39
```

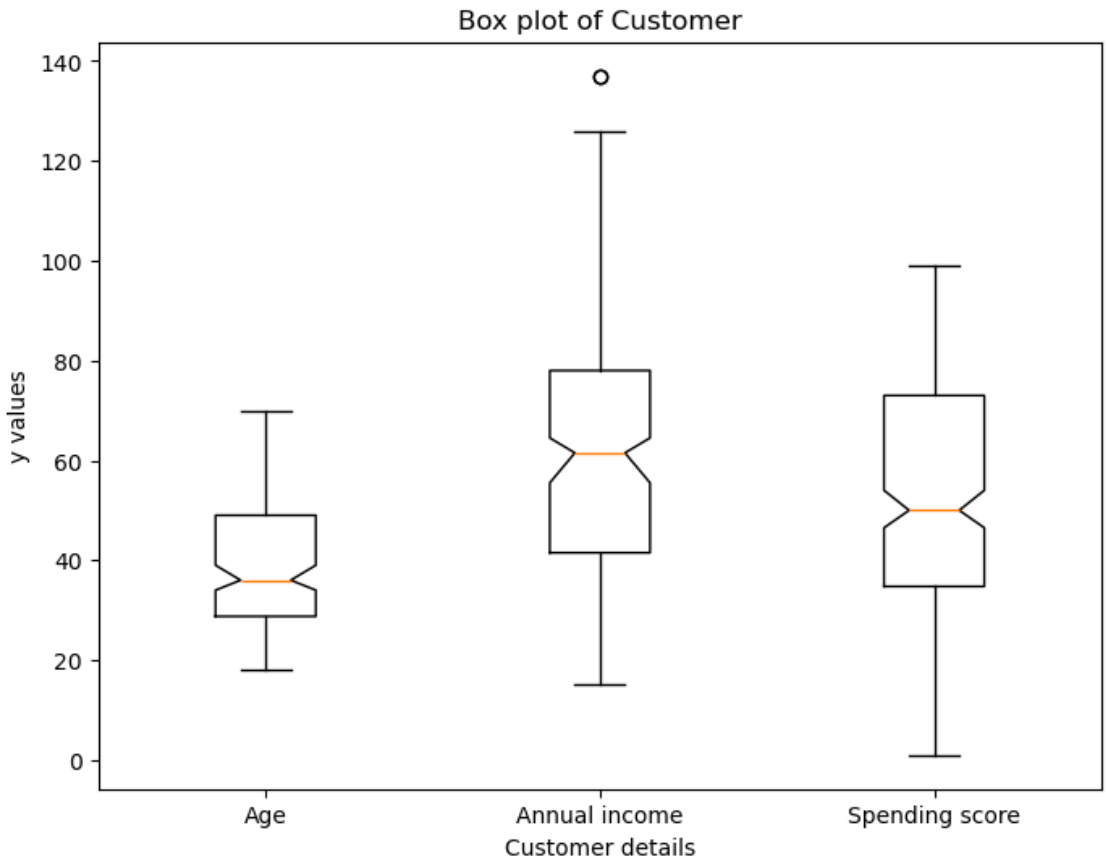
[70]:
```
pd.isnull(df).sum() #gives total number of blank places
```

[70]:
```
CustomerID                0
Genre                     0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

[71]:
```
fig=plt.figure(figsize=(8,6))
axes= fig.add_subplot(111)
axes.set_title("Spending score")
axes.set_xlabel('Spending score')
axes.set_ylabel("y vals")
axes.hist(Spending_score,bins=20,orientation='vertical')
plt.show()
```

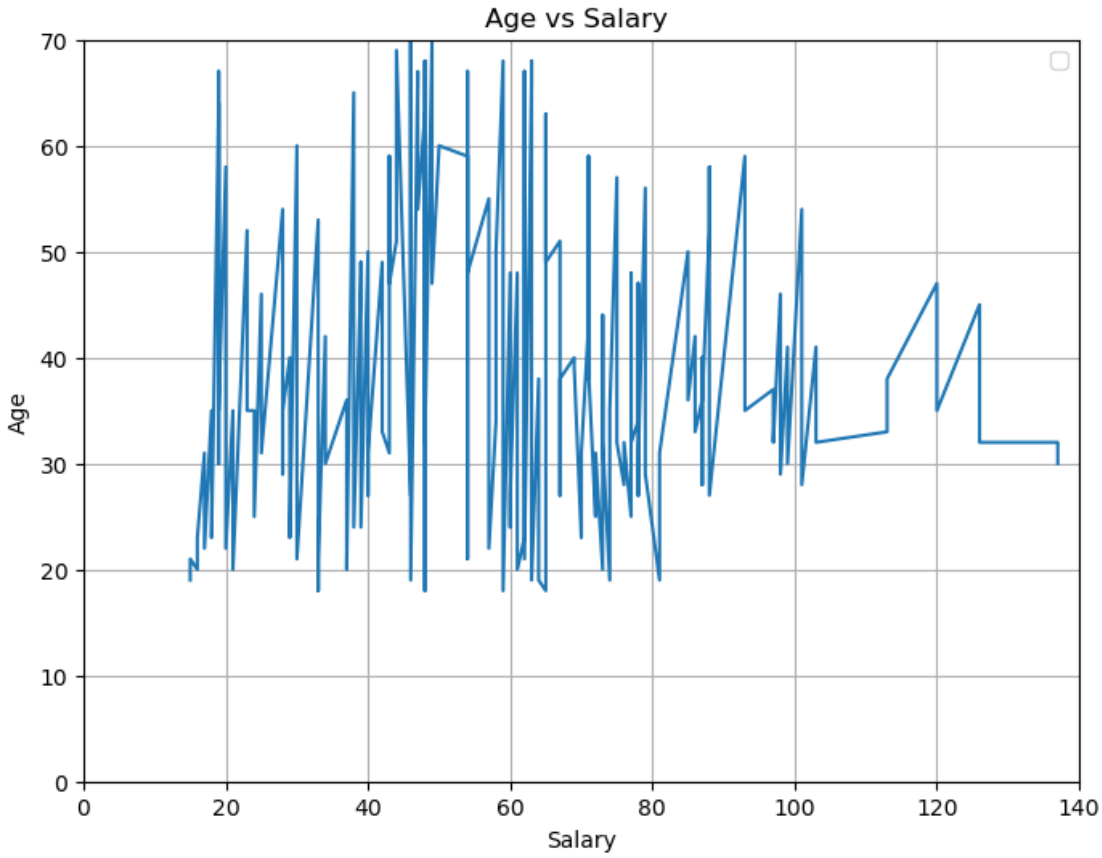


[72]:
```
age=df["Age"]
Annual_income=df["Annual Income (k$)"]
fig = plt.figure(figsize=(8,6))
ax = fig.add_subplot(111)
ax.set(title="Box plot of Customer",
      xlabel='Customer details', ylabel='y values')
ax.boxplot([age,Annual_income,Spending_score],
           labels=["Age", "Annual income", "Spending score"],
           notch=True, bootstrap=10000)
plt.show()
```

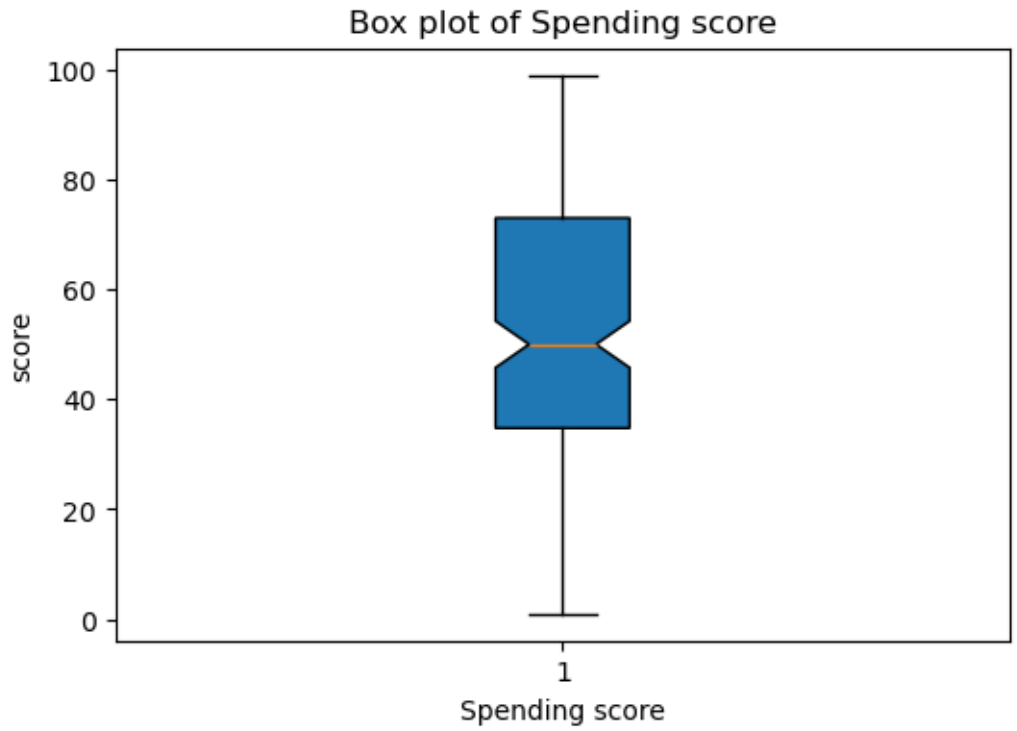


[73]:
```
fig = plt.figure(figsize=(8,6))
axes = fig.add_subplot(111)
axes.set_title("Age vs Salary ")
axes.set_xlabel("Salary")
axes.set_ylabel('Age')
axes.set_xlim([0,140])
axes.set_ylim([0,70])
axes.plot(y,x)
axes.legend()
axes.grid(True)
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
```



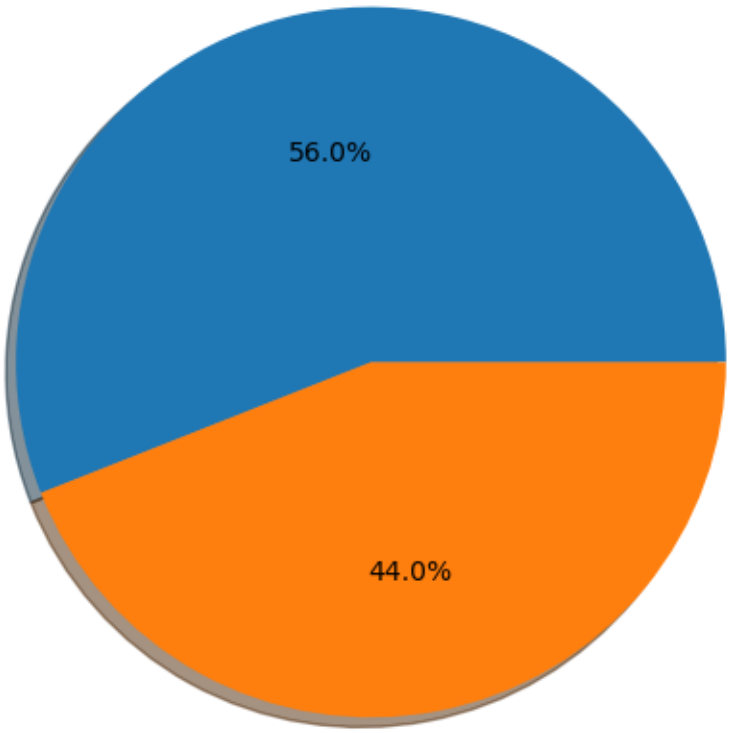
[74]:
```
fig = plt.figure(figsize=(6,4))
axes = fig.add_subplot(111)
axes.set(title="Box plot of Spending score",
      xlabel='Spending score', ylabel='score')
axes.boxplot(Spending_score, notch=True,patch_artist=True)
plt.show()
```



[75]:
```
fig=plt.figure(figsize=(8,6))
axes=fig.add_subplot(111)
explode= (0.2,0,0)
axes.pie(x=df["Genre"].value_counts(),shadow=True,autopct='%1.1f%%')
axes.set_xlabel("Gender division")
axes.legend()
plt.show
```

```
No artists with labels found to put in legend.  Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
```

[75]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Gender division

[ ]: