ASSIGNMENT- 4
CUSTOMER SEGMENTATION ANALYSIS

Assignment Date	28 October 2022
Student Name	V.MADHESH
Student Roll Number	820419104030
Maximum Marks	2 Marks

Importing the libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading the dataset:

Input:

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

Output:

	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

Encoding Categorical Columns

Input:

```
from sklearn.preprocessing import
LabelEncoderle = LabelEncoder()
df['Gender'] = le.fit_transform(df['Gender'])
df
```

Output:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	Cluster
0	1	1	19	15.00	39	2
1	2	1	21	15.00	81	2
2	3	0	20	16.00	6	2
3	4	0	23	16.00	77	2
4	5	0	31	17.00	40	2
...	...	...	...	...	...	...
195	196	0	35	120.00	79	3
196	197	0	45	126.00	28	1
197	198	1	32	126.00	74	3
198	199	1	32	60.55	18	1
199	200	1	30	60.55	83	3

200 rows × 6 columns

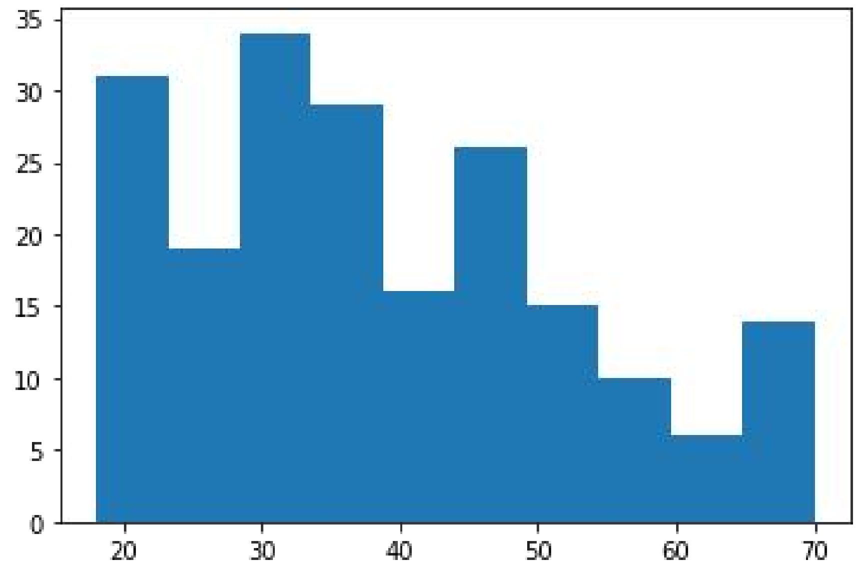
Visualizations
Univariate Analysis

Input:

```
plt.hist(df['Age'])
```

Output:

```
(array([31., 19., 34., 29., 16., 26., 15., 10.,   6., 14.]),
  array([18. , 23.2, 28.4, 33.6, 38.8, 44. , 49.2, 54.4, 59.6, 64.8, 70. ]),
  )
```

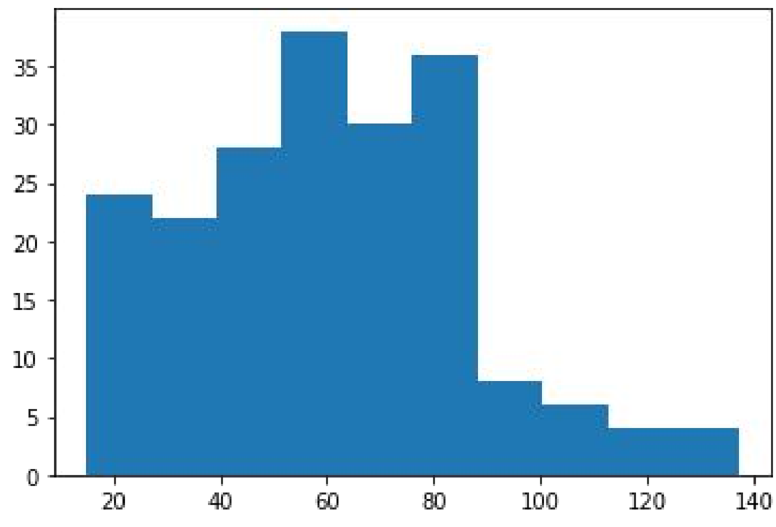


Input:

plt.hist(df['Annual Income (k$)'])

Output:

```
(array([24., 22., 28., 38., 30., 36.,   8.,   6.,   4.,   4.]),
  array([ 15. ,   27.2,   39.4,   51.6,   63.8,   76. ,   88.2, 100.4, 112.6,
              124.8, 137. ]),
  )
```

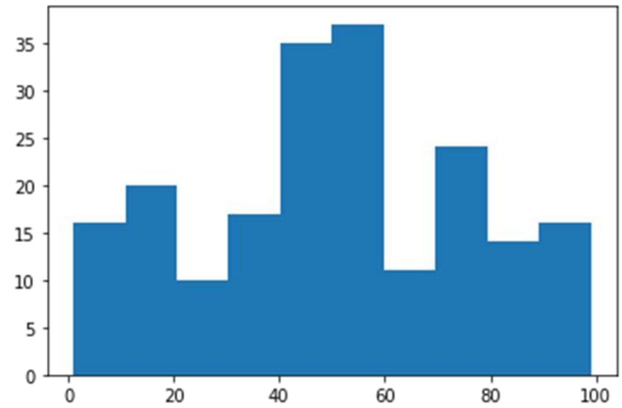


Input:

plt.hist(df['Spending Score (1-100)'])

Output:

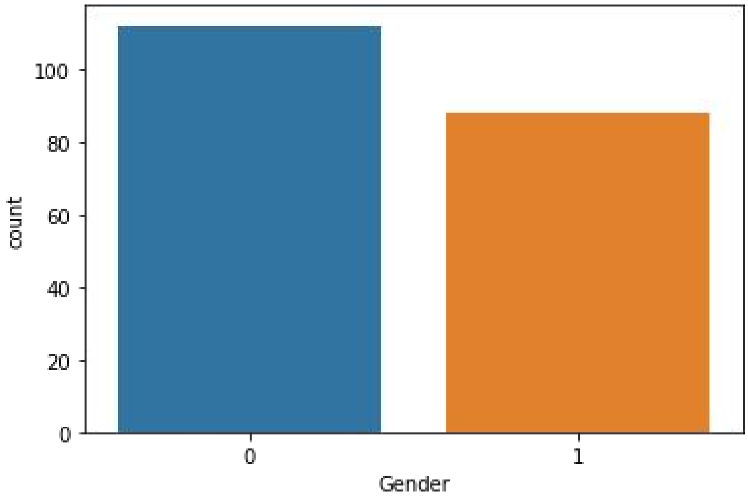
```
(array([16., 20., 10., 17., 35., 37., 11., 24., 14., 16.]),
  array([ 1. , 10.8, 20.6, 30.4, 40.2, 50. , 59.8, 69.6, 79.4, 89.2, 99. ]),
  )
```



Input:

sns.countplot(df['Gender'])

Output:

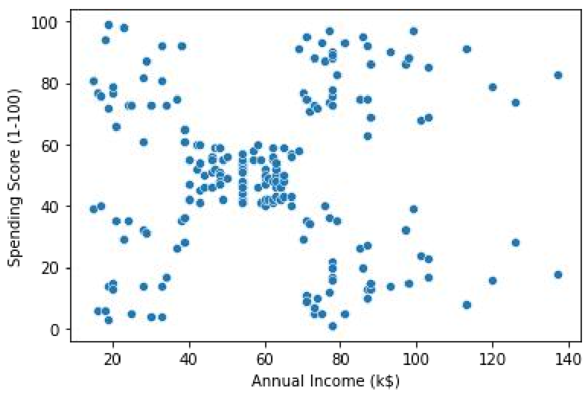


Bi-Variate Analysis

Input:

sns.scatterplot(df['Annual Income (k$)'], df['Spending Score (1-100)'])

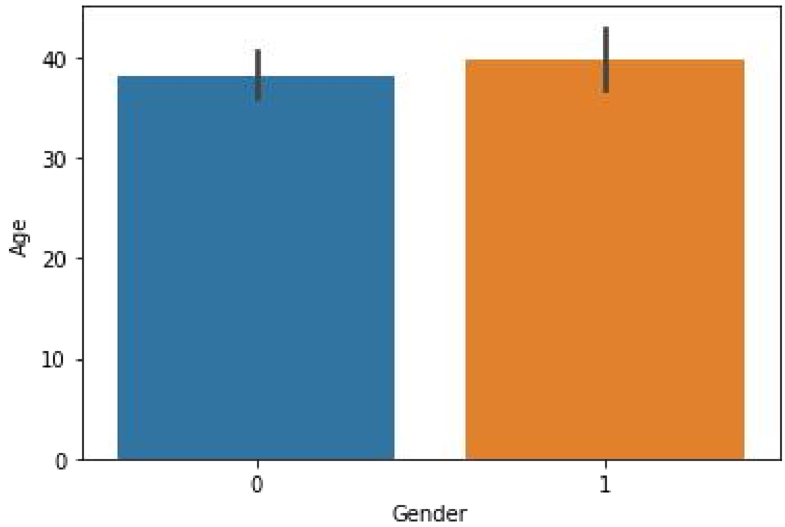
Output:



Input:

sns.barplot(df['Gender'], df['Age'])

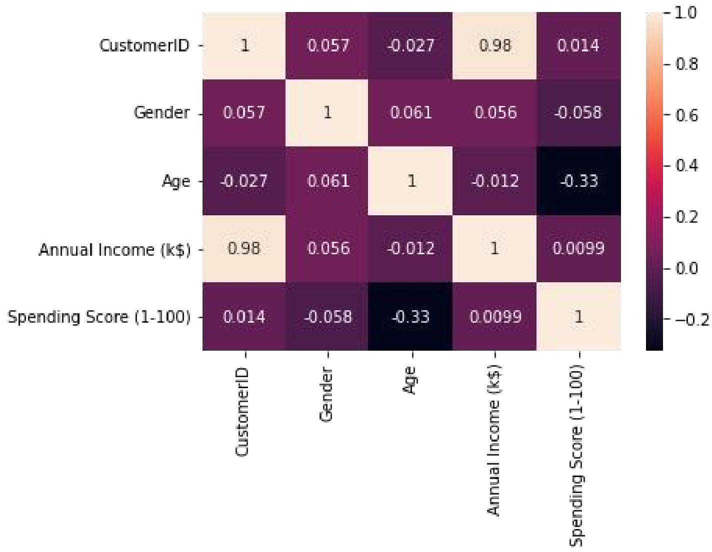
Output:



Input:

sns.heatmap(df.corr(), annot = **True**)

Output:

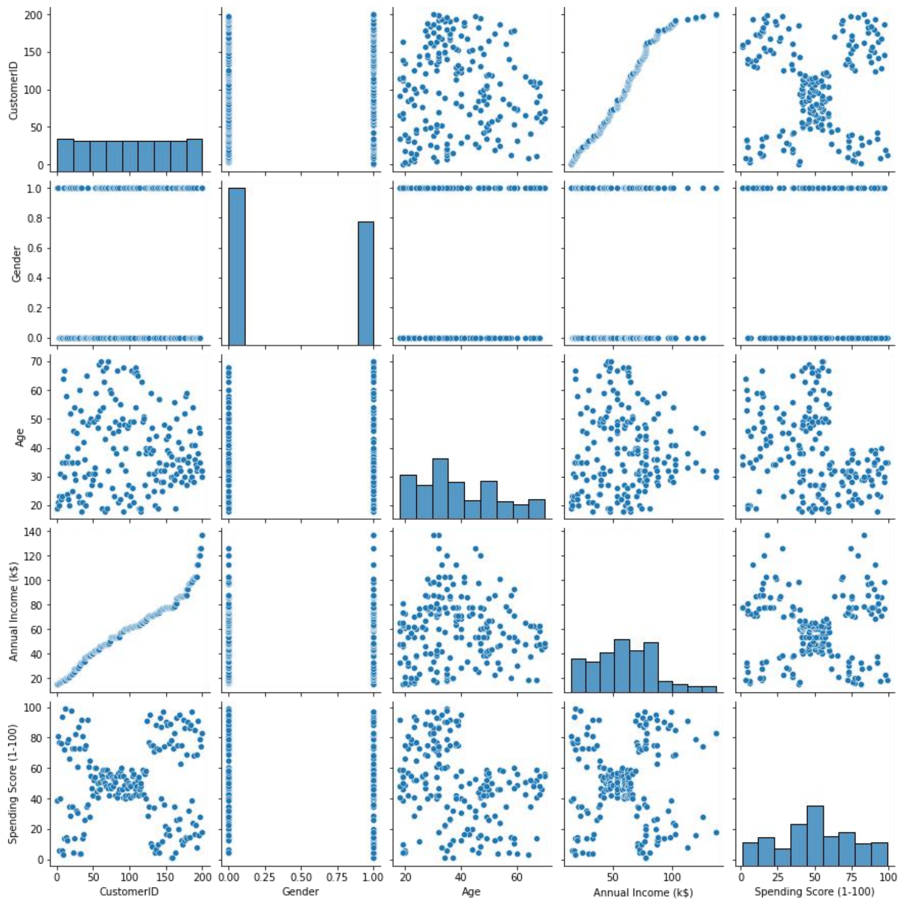


Multi-variate Analysis

Input:

sns.pairplot(df)

output:



Descriptive Statistics

Input:

df.info()

Output:

RangeIndex: 200 entries, 0 to 199Data columns
(total 5 columns):

#	Column	Non-Null Count	Dtype
---	---------------	---------------	------
0	CustomerID	200 non-null	int64
1	Gender	200 non-null	int64
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(5)
memory usage: 7.9 KB

Input:

df.describe()

Output:

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

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	0.440000	38.850000	60.560000	50.200000
std	57.879185	0.497633	13.969007	26.264721	25.823522
min	1.000000	0.000000	18.000000	15.000000	1.000000
25%	50.750000	0.000000	28.750000	41.500000	34.750000
50%	100.500000	0.000000	36.000000	61.500000	50.000000
75%	150.250000	1.000000	49.000000	78.000000	73.000000
max	200.000000	1.000000	70.000000	137.000000	99.000000

Input:

```
df.skew()
```

Output:

```
CustomerID                  0.000000
Gender                      0.243578
Age                         0.485569
Annual Income      (k$)     0.321843
Spending Score     (1-100) -0.047220
dtype: float64
```

Input:

```
df.kurt()
```

Output:

```
CustomerID                 -1.200000
Gender                     -1.960375
Age                        -0.671573
Annual Income      (k$)    -0.098487
Spending Score (1-100)    -0.826629
dtype: float64
```

Input:

df.corr()

Output:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
CustomerID	1.000000	0.057400	-0.026763	0.977548	0.013835
Gender	0.057400	1.000000	0.060867	0.056410	-0.058109
Age	-0.026763	0.060867	1.000000	-0.012398	-0.327227
Annual Income (k$)	0.977548	0.056410	-0.012398	1.000000	0.009903
Spending Score (1-100)	0.013835	-0.058109	-0.327227	0.009903	1.000000

Input:
df.var()

Output:
```
CustomerID                  3350.000000
Gender                         0.247638
Age                          195.133166
Annual Income     (k$)       689.835578
Spending Score     (1-100)   666.854271
dtype: float64
```

Input:
df.std()

Output:
```
CustomerID                  57.879185
Gender                       0.497633
Age                         13.969007
Annual Income     (k$)      26.264721
Spending Score (1-100)        25.823522
dtype: float64
```

Checking for missing values

Input:

df.isna().sum()

Output:

```
CustomerID                0
Gender                    0
Age                       0
Annual Income     (k$)    0
Spending Score (1-100)    0
dtype: int64
```

Input:

```
df.isna().sum().sum()
```

Output:

```
0
```

Input:

```
df.duplicated().sum()
```

Output:

```
0
```

Finding & Handling Ouliers

Input:

```
quantile = df.quantile(q = [0.25, 0.75])quantile
```

Output:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0.25	50.75	0.0	28.75	41.5	34.75
0.75	150.25	1.0	49.00	78.0	73.00

Input:

```
IQR = quantile.iloc[1] - quantile.iloc[0]IQR
```

Output:

```
CustomerID                99.50
Gender                    1.00
Age                       20.25
Annual Income (k$)        36.50
Spending Score (1-100)    38.25
dtype: float64
```

Input:

```
upper = quantile.iloc[1] + (1.5 *IQR)upper
```

Output:

```
CustomerID                     299.500
Gender                           2.500
Age                             79.375
Annual Income        (k$)      132.750
Spending Score (1-100)dtype:   130.375
float64
```

Input:

```
lower = quantile.iloc[0]lower - (1.5* IQR)
```

Output:

```
CustomerID                     -98.500
Gender                          -1.500
Age                             -1.625
Annual Income (k$)             -13.250
Spending Score (1-100)dtype:   -22.625
float64
```

Input:

```
df.mean()
```

Output:

```
CustomerID                100.50
Gender                      0.44
Age                        38.85
Annual Income (k$)         60.56
Spending Score (1-100)     50.20
dtype: float64
```

Input:

```
df['Annual Income (k$)'].max()
```

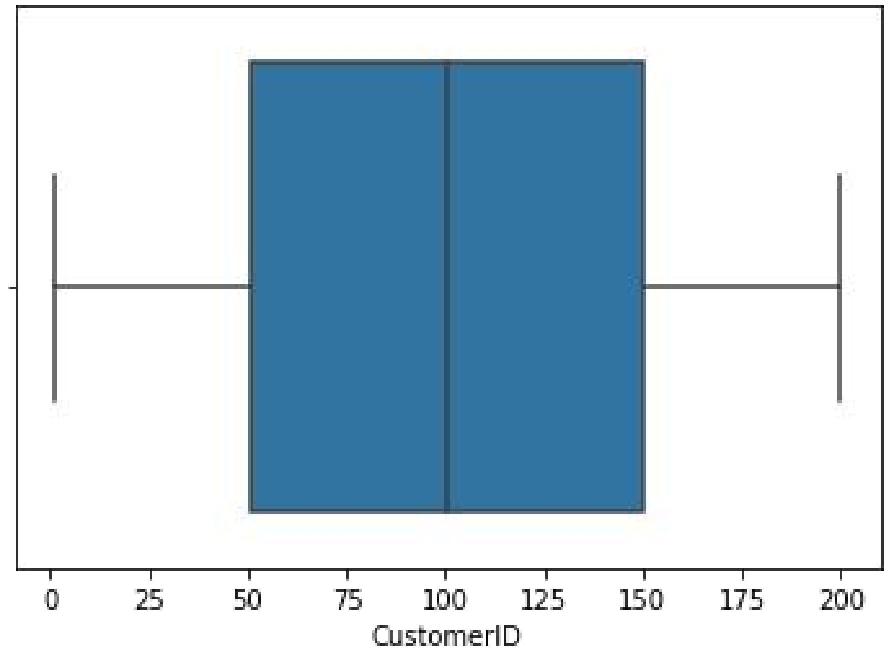
Output:

```
137
```

Input:

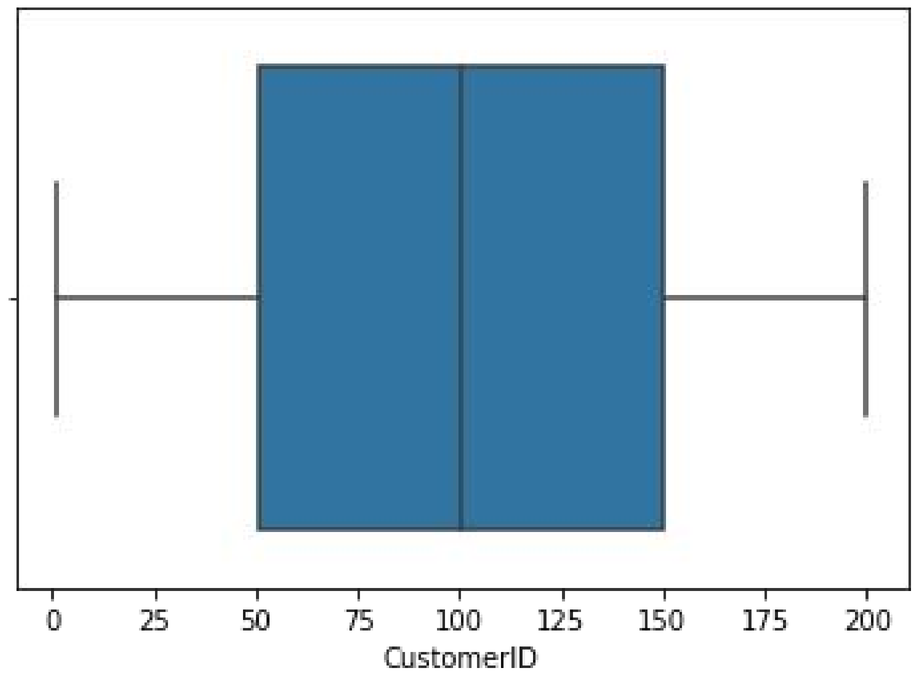
```
sns.boxplot(df['CustomerID'])
```

Output:



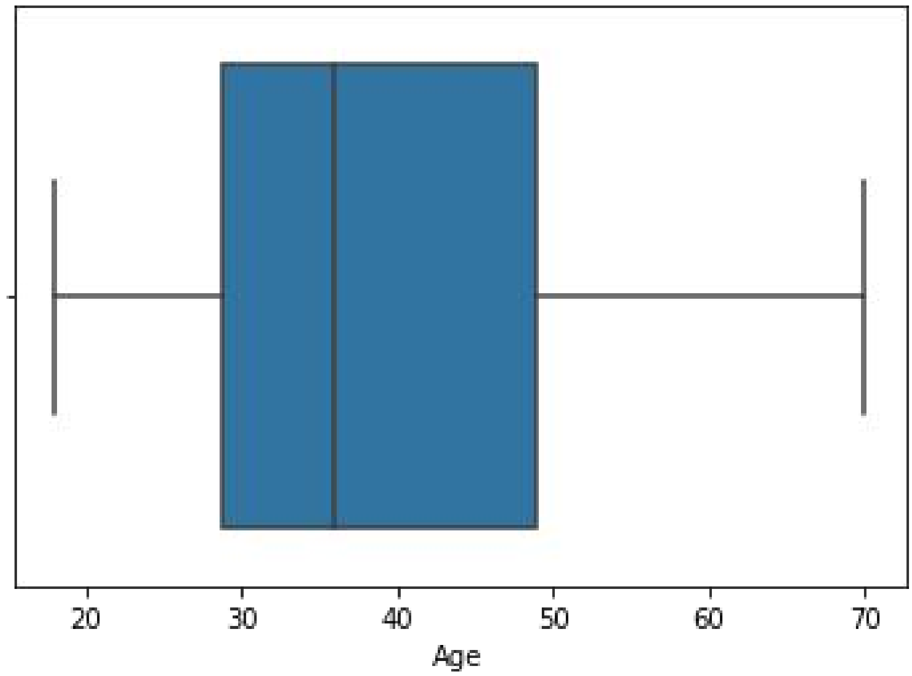
Input:
sns.boxplot(df['Gender'])

Output:



Input:
sns.boxplot(df['Age'])

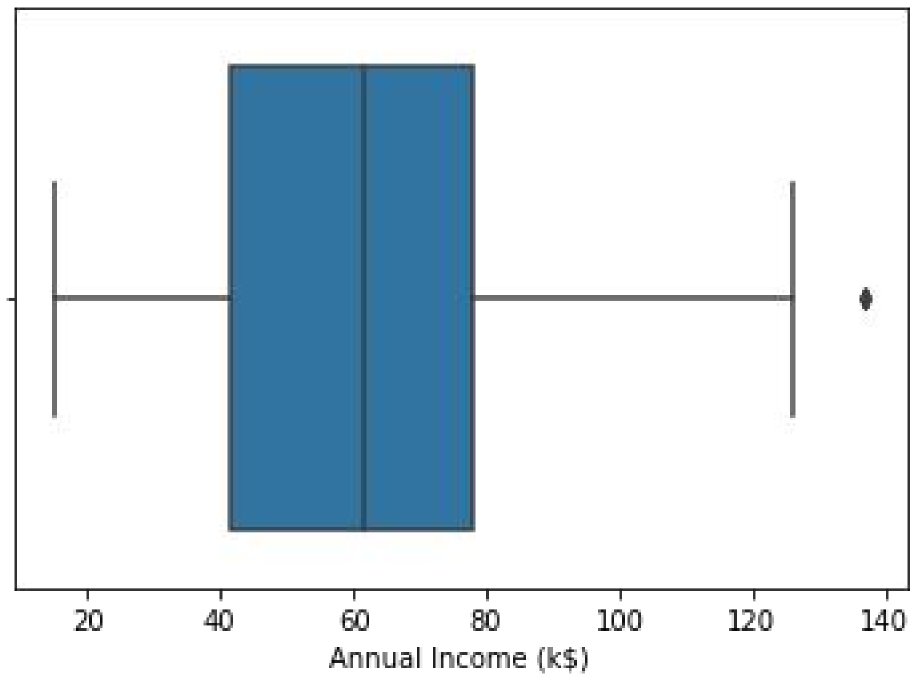
Output:



Input:

sns.boxplot(df['Annual Income (k$)'])

Output:

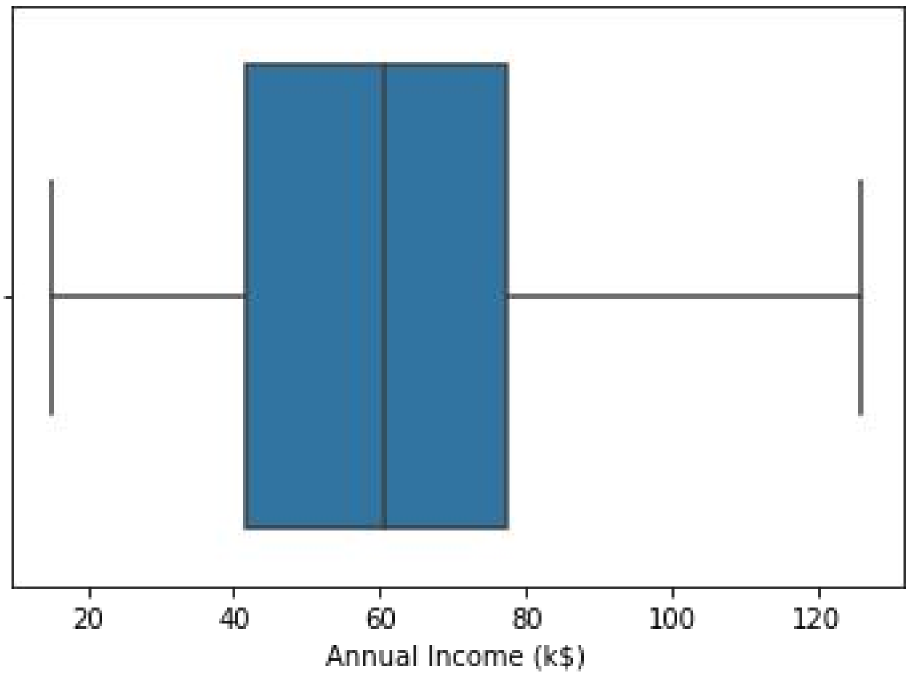


Input:

df['Annual Income (k$)'] = np.where(df['Annual Income (k$)'] > 132.750,60.55, df['Annual Income (k$)'])

sns.boxplot(df['Annual Income (k$)'])

Output:

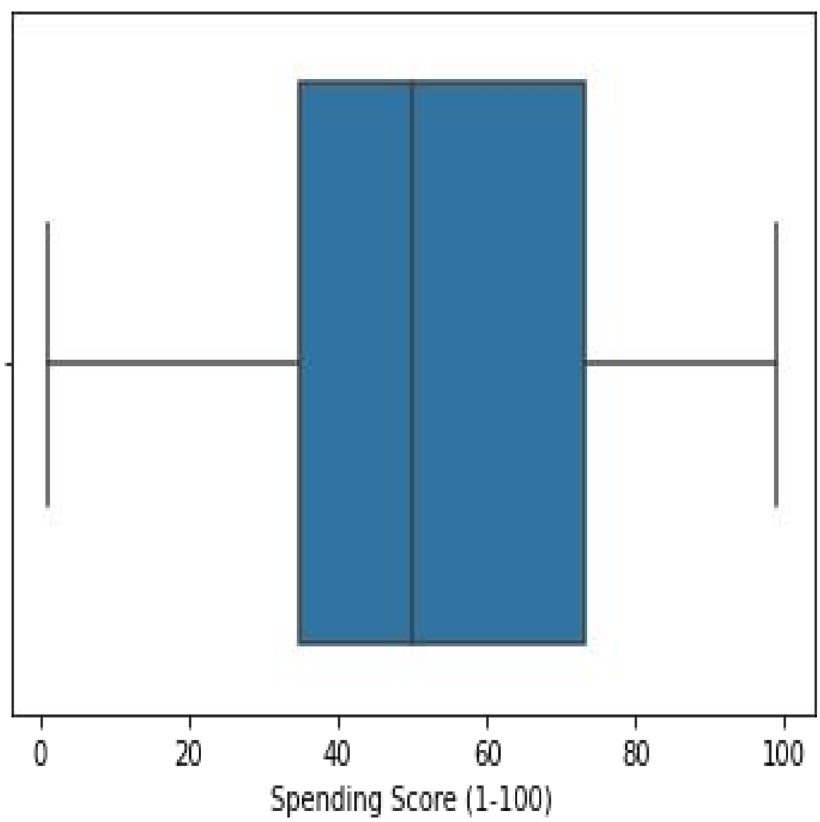


Input:
df['Annual Income (k$)'].max()

Output:
126.0

Input:
sns.boxplot(df['Spending Score (1-100)'])

Output:



Scaling the data

Input:

```
from sklearn.preprocessing import StandardScalerss =
StandardScaler().fit_transform(df)
ss
```

Output:

```
array([[-1.7234121 ,  1.12815215, -1.42456879, -1.78843062, -0.43480148],
       [-1.70609137,  1.12815215, -1.28103541, -1.78843062,  1.19570407],
       [-1.68877065, -0.88640526, -1.3528021 , -1.74850629, -1.71591298],
       [-1.67144992, -0.88640526, -1.13750203, -1.74850629,  1.04041783],
       [-1.6541292 , -0.88640526, -0.56336851, -1.70858195, -0.39597992],
       [-1.63680847, -0.88640526, -1.20926872, -1.70858195,  1.00159627],
       [-1.61948775, -0.88640526, -0.27630176, -1.66865761, -1.71591298],
       [-1.60216702, -0.88640526, -1.13750203, -1.66865761,  1.70038436],
       [-1.5848463 ,  1.12815215,  1.80493225, -1.62873328, -1.83237767],
       [-1.56752558, -0.88640526, -0.6351352 , -1.62873328,  0.84631002],
       [-1.55020485,  1.12815215,  2.02023231, -1.62873328, -1.4053405 ],
       [-1.53288413, -0.88640526, -0.27630176, -1.62873328,  1.89449216],
       [-1.5155634 , -0.88640526,  1.37433211, -1.58880894, -1.36651894],
       [-1.49824268, -0.88640526, -1.06573534, -1.58880894,  1.04041783],
       [-1.48092195,  1.12815215, -0.13276838, -1.58880894, -1.44416206],
       [-1.46360123,  1.12815215, -1.20926872, -1.58880894,  1.11806095],
       [-1.4462805 , -0.88640526, -0.27630176, -1.5488846 , -0.59008772],
       [-1.42895978,  1.12815215, -1.3528021 , -1.5488846 ,  0.61338066],
       [-1.41163905,  1.12815215,  0.94373197, -1.46903593, -0.82301709],
       [-1.39431833, -0.88640526, -0.27630176, -1.46903593,  1.8556706 ],
       [-1.3769976 ,  1.12815215, -0.27630176, -1.42911159, -0.59008772],
       [-1.35967688,  1.12815215, -0.99396865, -1.42911159,  0.88513158],
       [-1.34235616, -0.88640526,  0.51313183, -1.38918726, -1.75473454],
       [-1.32503543,  1.12815215, -0.56336851, -1.38918726,  0.88513158],
       [-1.30771471, -0.88640526,  1.08726535, -1.26941425, -1.4053405 ],
       [-1.29039398,  1.12815215, -0.70690189, -1.26941425,  1.23452563],
       [-1.27307326, -0.88640526,  0.44136514, -1.26941425, -0.7065524 ],
       [-1.25575253,  1.12815215, -0.27630176, -1.26941425,  0.41927286],
       [-1.23843181, -0.88640526,  0.08253169, -1.22948991, -0.74537397],
       [-1.22111108, -0.88640526, -1.13750203, -1.22948991,  1.42863343],
       [-1.20379036,  1.12815215,  1.51786549, -1.18956557, -1.7935561 ],
       [-1.18646963, -0.88640526, -1.28103541, -1.18956557,  0.88513158],
       [-1.16914891,  1.12815215,  1.01549866, -1.06979256, -1.7935561 ],
       [-1.15182818,  1.12815215, -1.49633548, -1.06979256,  1.62274124],
       [-1.13450746, -0.88640526,  0.7284319 , -1.06979256, -1.4053405 ],
       [-1.11718674, -0.88640526, -1.28103541, -1.06979256,  1.19570407],
       [-1.09986601, -0.88640526,  0.22606507, -1.02986823, -1.28887582],
       [-1.08254529, -0.88640526, -0.6351352 , -1.02986823,  0.88513158],
       [-1.06522456, -0.88640526, -0.20453507, -0.91009522, -0.93948177],
       [-1.04790384, -0.88640526, -1.3528021 , -0.91009522,  0.96277471],
       [-1.03058311, -0.88640526,  1.87669894, -0.87017088, -0.59008772],
       [-1.01326239,  1.12815215, -1.06573534, -0.87017088,  1.62274124],
       [-0.99594166,  1.12815215,  0.65666521, -0.83024654, -0.55126616],
       [-0.97862094, -0.88640526, -0.56336851, -0.83024654,  0.41927286],
       [-0.96130021, -0.88640526,  0.7284319 , -0.83024654, -0.86183865],
       [-0.94397949, -0.88640526, -1.06573534, -0.83024654,  0.5745591 ],
       [-0.92665877, -0.88640526,  0.80019859, -0.79032221,  0.18634349],
```

[-0.90933804, -0.88640526, -0.85043527, -0.79032221, -0.12422899],
[-0.89201732, -0.88640526, -0.70690189, -0.79032221, -0.3183368],
[-0.87469659, -0.88640526, -0.56336851, -0.79032221, -0.3183368],
[-0.85737587, -0.88640526, 0.7284319 , -0.71047353, 0.06987881],
[-0.84005514, 1.12815215, -0.41983513, -0.71047353, 0.38045129],
[-0.82273442, -0.88640526, -0.56336851, -0.6705492 , 0.14752193],
[-0.80541369, 1.12815215, 1.4460988 , -0.6705492 , 0.38045129],
[-0.78809297, -0.88640526, 0.80019859, -0.6705492 , -0.20187212],
[-0.77077224, 1.12815215, 0.58489852, -0.6705492 , -0.35715836],
[-0.75345152, -0.88640526, 0.87196528, -0.63062486, -0.00776431],
[-0.73613079, 1.12815215, 2.16376569, -0.63062486, -0.16305055],
[-0.71881007, -0.88640526, -0.85043527, -0.55077619, 0.03105725],
[-0.70148935, 1.12815215, 1.01549866, -0.55077619, -0.16305055],
[-0.68416862, 1.12815215, 2.23553238, -0.55077619, 0.22516505],
[-0.6668479 , 1.12815215, -1.42456879, -0.55077619, 0.18634349],
[-0.64952717, -0.88640526, 2.02023231, -0.51085185, 0.06987881],
[-0.63220645, -0.88640526, 1.08726535, -0.51085185, 0.34162973],
[-0.61488572, 1.12815215, 1.73316556, -0.47092751, 0.03105725],
[-0.597565 , 1.12815215, -1.49633548, -0.47092751, 0.34162973],
[-0.58024427, -0.88640526, 0.29783176, -0.47092751, -0.00776431],
[-0.56292355, -0.88640526, 2.091999 , -0.47092751, -0.08540743],
[-0.54560282, 1.12815215, -1.42456879, -0.47092751, 0.34162973],
[-0.5282821 , -0.88640526, -0.49160182, -0.47092751, -0.12422899],
[-0.51096138, 1.12815215, 2.23553238, -0.43100318, 0.18634349],
[-0.49364065, -0.88640526, 0.58489852, -0.43100318, -0.3183368],
[-0.47631993, -0.88640526, 1.51786549, -0.39107884, -0.04658587],
[-0.4589992 , -0.88640526, 1.51786549, -0.39107884, 0.22516505],
[-0.44167848, 1.12815215, 1.4460988 , -0.23138149, -0.12422899],
[-0.42435775, 1.12815215, -0.92220196, -0.23138149, 0.14752193],
[-0.40703703, -0.88640526, 0.44136514, -0.23138149, 0.10870037],
[-0.3897163 , 1.12815215, 0.08253169, -0.23138149, -0.08540743],
[-0.37239558, -0.88640526, -1.13750203, -0.23138149, 0.06987881],
[-0.35507485, -0.88640526, 0.7284319 , -0.23138149, -0.3183368],
[-0.33775413, 1.12815215, 1.30256542, -0.23138149, 0.03105725],
[-0.3204334 , 1.12815215, -0.06100169, -0.23138149, 0.18634349],
[-0.30311268, 1.12815215, 2.02023231, -0.23138149, -0.35715836],
[-0.28579196, -0.88640526, 0.51313183, -0.23138149, -0.24069368],
[-0.26847123, -0.88640526, -1.28103541, -0.23138149, 0.26398661],
[-0.25115051, 1.12815215, 0.65666521, -0.23138149, -0.16305055],
[-0.23382978, -0.88640526, 1.15903204, -0.11160848, 0.30280817],
[-0.21650906, -0.88640526, -1.20926872, -0.11160848, 0.18634349],
[-0.19918833, -0.88640526, -0.34806844, -0.07168415, 0.38045129],
[-0.18186761, -0.88640526, 0.80019859, -0.07168415, -0.16305055],
[-0.16454688, -0.88640526, 2.091999 , -0.03175981, 0.18634349],
[-0.14722616, 1.12815215, -1.49633548, -0.03175981, -0.35715836],
[-0.12990543, 1.12815215, 0.65666521, 0.00816453, -0.04658587],
[-0.11258471, -0.88640526, 0.08253169, 0.00816453, -0.39597992],
[-0.09526399, -0.88640526, -0.49160182, 0.00816453, -0.3183368],
[-0.07794326, 1.12815215, -1.06573534, 0.00816453, 0.06987881],
[-0.06062254, -0.88640526, 0.58489852, 0.00816453, -0.12422899],
[-0.04330181, -0.88640526, -0.85043527, 0.00816453, -0.00776431],
[-0.02598109, 1.12815215, 0.65666521, 0.04808886, -0.3183368],
[-0.00866036, 1.12815215, -1.3528021 , 0.04808886, -0.04658587],
[0.00866036, -0.88640526, -1.13750203, 0.0880132 , -0.35715836],
[0.02598109, -0.88640526, 0.7284319 , 0.0880132 , -0.08540743],
[0.04330181, 1.12815215, 2.02023231, 0.0880132 , 0.34162973],
[0.06062254, 1.12815215, -0.92220196, 0.0880132 , 0.18634349],

[0.07794326, 1.12815215, 0.7284319 , 0.0880132 , 0.22516505],
[0.09526399, -0.88640526, -1.28103541, 0.0880132 , -0.3183368],
[0.11258471, -0.88640526, 1.94846562, 0.12793754, -0.00776431],
[0.12990543, 1.12815215, 1.08726535, 0.12793754, -0.16305055],
[0.14722616, 1.12815215, 2.091999 , 0.12793754, -0.27951524],
[0.16454688, 1.12815215, 1.94846562, 0.12793754, -0.08540743],
[0.18186761, 1.12815215, 1.87669894, 0.12793754, 0.06987881],
[0.19918833, -0.88640526, -1.42456879, 0.12793754, 0.14752193],
[0.21650906, -0.88640526, -0.06100169, 0.16786187, -0.3183368],
[0.23382978, 1.12815215, -1.42456879, 0.16786187, -0.16305055],
[0.25115051, -0.88640526, -1.49633548, 0.20778621, -0.08540743],
[0.26847123, -0.88640526, -1.42456879, 0.20778621, -0.00776431],
[0.28579196, -0.88640526, 1.73316556, 0.20778621, -0.27951524],
[0.30311268, -0.88640526, 0.7284319 , 0.20778621, 0.34162973],
[0.3204334 , -0.88640526, 0.87196528, 0.28763488, -0.27951524],
[0.33775413, -0.88640526, 0.80019859, 0.28763488, 0.26398661],
[0.35507485, 1.12815215, -0.85043527, 0.28763488, 0.22516505],
[0.37239558, -0.88640526, -0.06100169, 0.28763488, -0.39597992],
[0.3897163 , -0.88640526, 0.08253169, 0.36748356, 0.30280817],
[0.40703703, 1.12815215, 0.010765 , 0.36748356, 1.58391968],
[0.42435775, -0.88640526, -1.13750203, 0.40740789, -0.82301709],
[0.44167848, -0.88640526, -0.56336851, 0.40740789, 1.04041783],
[0.4589992 , 1.12815215, 0.29783176, 0.44733223, -0.59008772],
[0.47631993, 1.12815215, 0.08253169, 0.44733223, 1.73920592],
[0.49364065, 1.12815215, 1.4460988 , 0.44733223, -1.52180518],
[0.51096138, 1.12815215, -0.06100169, 0.44733223, 0.96277471],
[0.5282821 , 1.12815215, 0.58489852, 0.44733223, -1.5994483],
[0.54560282, 1.12815215, 0.010765 , 0.44733223, 0.96277471],
[0.56292355, -0.88640526, -0.99396865, 0.48725657, -0.62890928],
[0.58024427, -0.88640526, -0.56336851, 0.48725657, 0.80748846],
[0.597565 , 1.12815215, -1.3528021 , 0.5271809 , -1.75473454],
[0.61488572, -0.88640526, -0.70690189, 0.5271809 , 1.46745499],
[0.63220645, -0.88640526, 0.36959845, 0.5271809 , -1.67709142],
[0.64952717, 1.12815215, -0.49160182, 0.5271809 , 0.88513158],
[0.6668479 , 1.12815215, -1.42456879, 0.56710524, -1.56062674],
[0.68416862, -0.88640526, -0.27630176, 0.56710524, 0.84631002],
[0.70148935, -0.88640526, 1.30256542, 0.60702958, -1.75473454],
[0.71881007, 1.12815215, -0.49160182, 0.60702958, 1.6615628],
[0.73613079, -0.88640526, -0.77866858, 0.64695391, -0.39597992],
[0.75345152, -0.88640526, -0.49160182, 0.64695391, 1.42863343],
[0.77077224, 1.12815215, -0.99396865, 0.68687825, -1.48298362],
[0.78809297, 1.12815215, -0.77866858, 0.68687825, 1.81684904],
[0.80541369, 1.12815215, 0.65666521, 0.68687825, -0.55126616],
[0.82273442, -0.88640526, -0.49160182, 0.68687825, 0.92395314],
[0.84005514, -0.88640526, -0.34806844, 0.72680259, -1.09476801],
[0.85737587, 1.12815215, -0.34806844, 0.72680259, 1.54509812],
[0.87469659, 1.12815215, 0.29783176, 0.72680259, -1.28887582],
[0.89201732, 1.12815215, 0.010765 , 0.72680259, 1.46745499],
[0.90933804, -0.88640526, 0.36959845, 0.72680259, -1.17241113],
[0.92665877, -0.88640526, -0.06100169, 0.72680259, 1.00159627],
[0.94397949, -0.88640526, 0.58489852, 0.72680259, -1.32769738],
[0.96130021, -0.88640526, -0.85043527, 0.72680259, 1.50627656],
[0.97862094, 1.12815215, -0.13276838, 0.72680259, -1.91002079],
[0.99594166, -0.88640526, -0.6351352 , 0.72680259, 1.07923939],
[1.01326239, 1.12815215, -0.34806844, 0.72680259, -1.91002079],
[1.03058311, -0.88640526, -0.6351352 , 0.72680259, 0.88513158],
[1.04790384, -0.88640526, 1.23079873, 0.76672692, -0.59008772],

```
[ 1.06522456, -0.88640526, -0.70690189, 0.76672692, 1.27334719],
[ 1.08254529, 1.12815215, -1.42456879, 0.8465756 , -1.75473454],
[ 1.09986601, -0.88640526, -0.56336851, 0.8465756 , 1.6615628 ],
[ 1.11718674, 1.12815215, 0.80019859, 1.00627294, -0.93948177],
[ 1.13450746, -0.88640526, -0.20453507, 1.00627294, 0.96277471],
[ 1.15182818, 1.12815215, 0.22606507, 1.04619728, -1.17241113],
[ 1.16914891, -0.88640526, -0.41983513, 1.04619728, 1.73920592],
[ 1.18646963, -0.88640526, -0.20453507, 1.08612162, -0.90066021],
[ 1.20379036, 1.12815215, -0.49160182, 1.08612162, 0.49691598],
[ 1.22111108, 1.12815215, 0.08253169, 1.08612162, -1.44416206],
[ 1.23843181, 1.12815215, -0.77866858, 1.08612162, 0.96277471],
[ 1.25575253, 1.12815215, -0.20453507, 1.08612162, -1.56062674],
[ 1.27307326, 1.12815215, -0.20453507, 1.08612162, 1.62274124],
[ 1.29039398, -0.88640526, 0.94373197, 1.12604595, -1.44416206],
[ 1.30771471, -0.88640526, -0.6351352 , 1.12604595, 1.38981187],
[ 1.32503543, 1.12815215, 1.37433211, 1.12604595, -1.36651894],
[ 1.34235616, 1.12815215, -0.85043527, 1.12604595, 0.72984534],
[ 1.35967688, 1.12815215, 1.4460988 , 1.32566764, -1.4053405 ],
[ 1.3769976 , 1.12815215, -0.27630176, 1.32566764, 1.54509812],
[ 1.39431833, -0.88640526, -0.13276838, 1.48536498, -0.7065524 ],
[ 1.41163905, -0.88640526, -0.49160182, 1.48536498, 1.38981187],
[ 1.42895978, 1.12815215, 0.51313183, 1.52528932, -1.36651894],
[ 1.4462805 , -0.88640526, -0.70690189, 1.52528932, 1.46745499],
[ 1.46360123, -0.88640526, 0.15429838, 1.56521366, -0.43480148],
[ 1.48092195, 1.12815215, -0.6351352 , 1.56521366, 1.81684904],
[ 1.49824268, -0.88640526, 1.08726535, 1.64506233, -1.01712489],
[ 1.5155634 , 1.12815215, -0.77866858, 1.64506233, 0.69102378],
[ 1.53288413, -0.88640526, 0.15429838, 1.724911 , -1.28887582],
[ 1.55020485, -0.88640526, -0.20453507, 1.724911 , 1.35099031],
[ 1.56752558, -0.88640526, -0.34806844, 1.724911 , -1.05594645],
[ 1.5848463 , -0.88640526, -0.49160182, 1.724911 , 0.72984534],
[ 1.60216702, 1.12815215, -0.41983513, 2.12415437, -1.63826986],
[ 1.61948775, -0.88640526, -0.06100169, 2.12415437, 1.58391968],
[ 1.63680847, -0.88640526, 0.58489852, 2.40362473, -1.32769738],
[ 1.6541292 , -0.88640526, -0.27630176, 2.40362473, 1.11806095],
[ 1.67144992, -0.88640526, 0.44136514, 2.64317075, -0.86183865],
[ 1.68877065, 1.12815215, -0.49160182, 2.64317075, 0.92395314],
[ 1.70609137, 1.12815215, -0.49160182, 0.03012291, -1.25005425],
[ 1.7234121 , 1.12815215, -0.6351352 , 0.03012291, 1.27334719]])
```

Clustering Algorithm

Input:

```
from sklearn.cluster import KMeansTWSS = []
k = list(range(2,9))
for i in k:
    kmeans = KMeans(n_clusters = i , init = 'k-means++')kmeans.fit(df)
    TWSS.append(kmeans.inertia_)TWSS
```

Output:

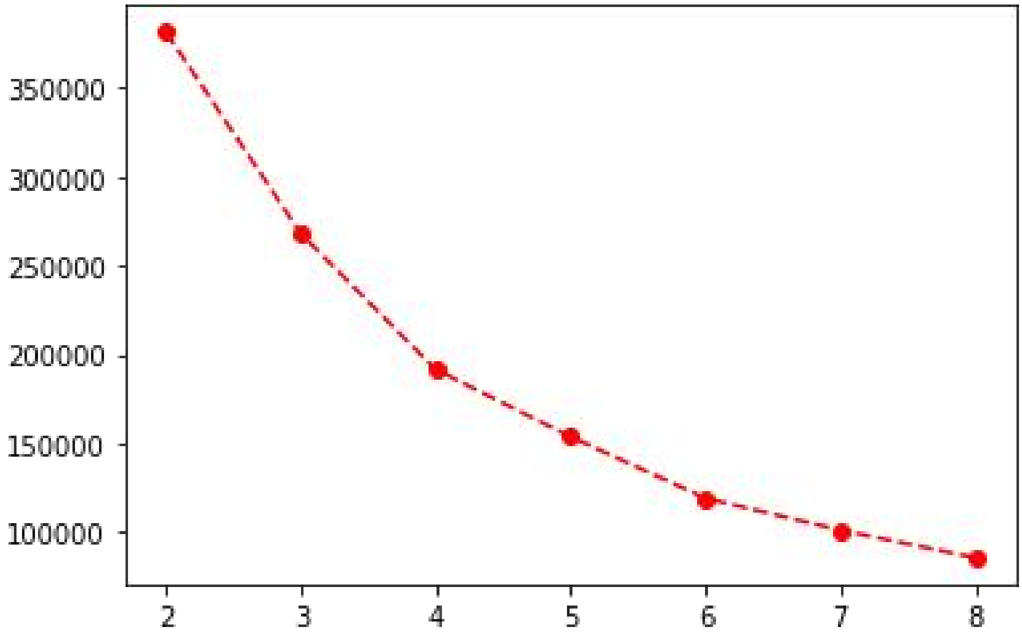
```
[381507.64738523855,
268062.55433747417,
191550.08627670942,
```

153777.55391034693,
119166.15727643928,
101239.32626154403,
85744.90139221892]

Input:

plt.plot(k,TWSS, 'ro--')

Output:



model = KMeans(n_clusters = 4)

Input:

model.fit(df)

Output:

KMeans(n_clusters=4)

Input:

mb = pd.Series(model.labels_)df['Cluster']

= mb

df

Output:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	Cluster
0	1	1	19	15.00	39	2
1	2	1	21	15.00	81	2

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	Cluster
2	3	0	20	16.00	6	2
3	4	0	23	16.00	77	2
4	5	0	31	17.00	40	2
...	...	...	...	...	...	...
195	196	0	35	120.00	79	3
196	197	0	45	126.00	28	1
197	198	1	32	126.00	74	3
198	199	1	32	60.55	18	1
199	200	1	30	60.55	83	3

200 rows × 6 columns