

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

1. Download the dataset:

1. Load the dataset into the tool.

```
data=pd.read_csv("/content/Mall_Customers.csv")
```

1. Perform Below Visualizations. · Univariate Analysis · Bi-Variate Analysis · Multi-Variate Analysis

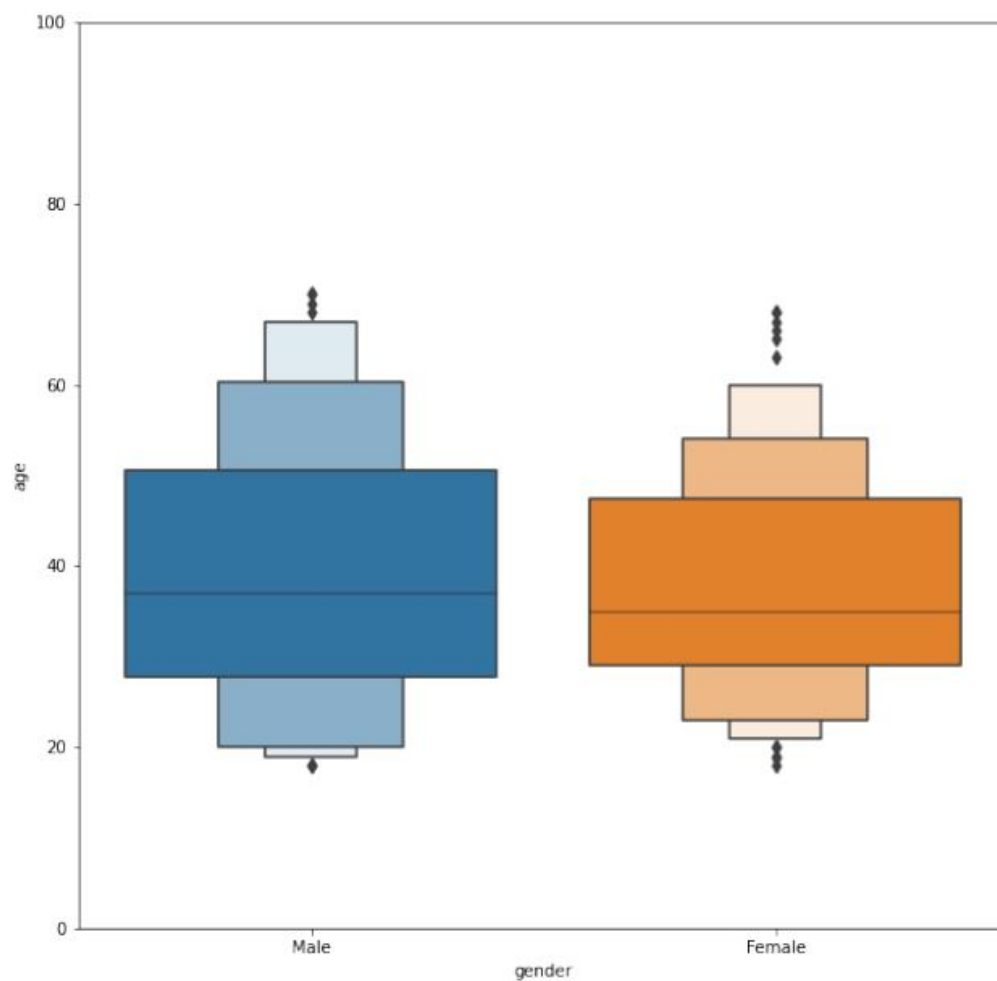
```
data.head()
```

	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

```
data.rename(columns={"CustomerID":"customer_id","Gender":"gender","Age":
"age","Annual Income (k$)":"annual_income",
"Spending Score (1-100)":"spending_scores"},inplace=True)
```

```
temp = pd.concat([data['age'], data['gender']], axis=1)
```

```
f, ax = plt.subplots(figsize=(10,10))
fig = sns.boxenplot(x='gender', y="age", data=data)
fig.axis(ymin=0, ymax=100);
```

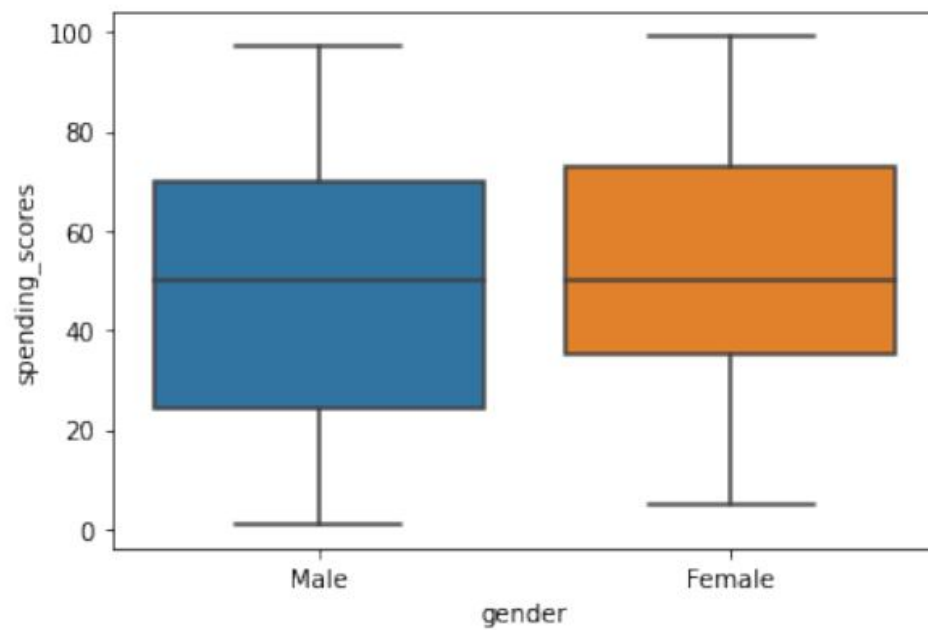


ANALYSIS

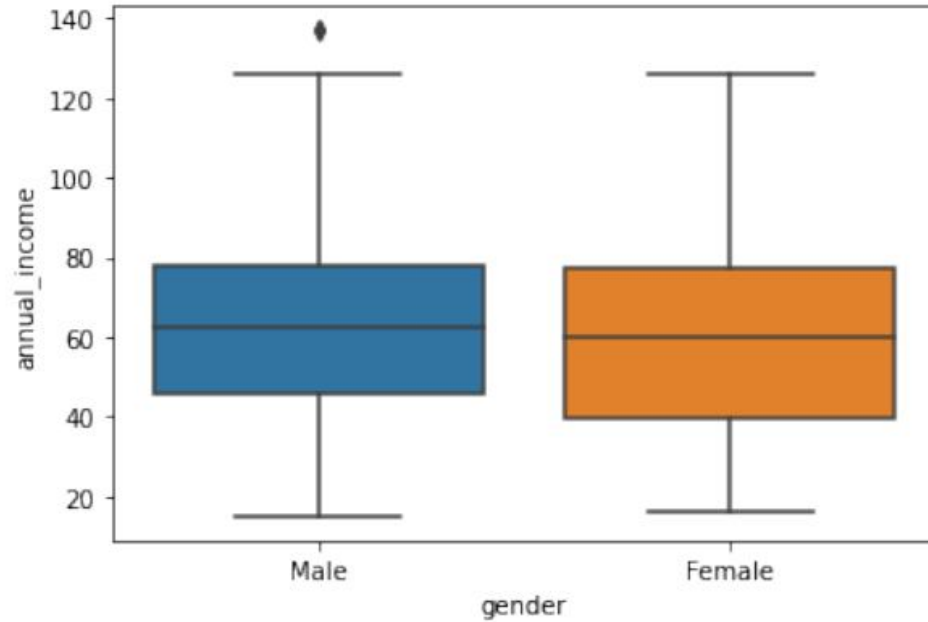
There is no difference in age of rings for male and female (18-70).

Count plot

```
sns.boxplot(x=data['gender'],y=data['spending_scores'])  
<matplotlib.axes._subplots.AxesSubplot at 0x7fe743f1b710>
```

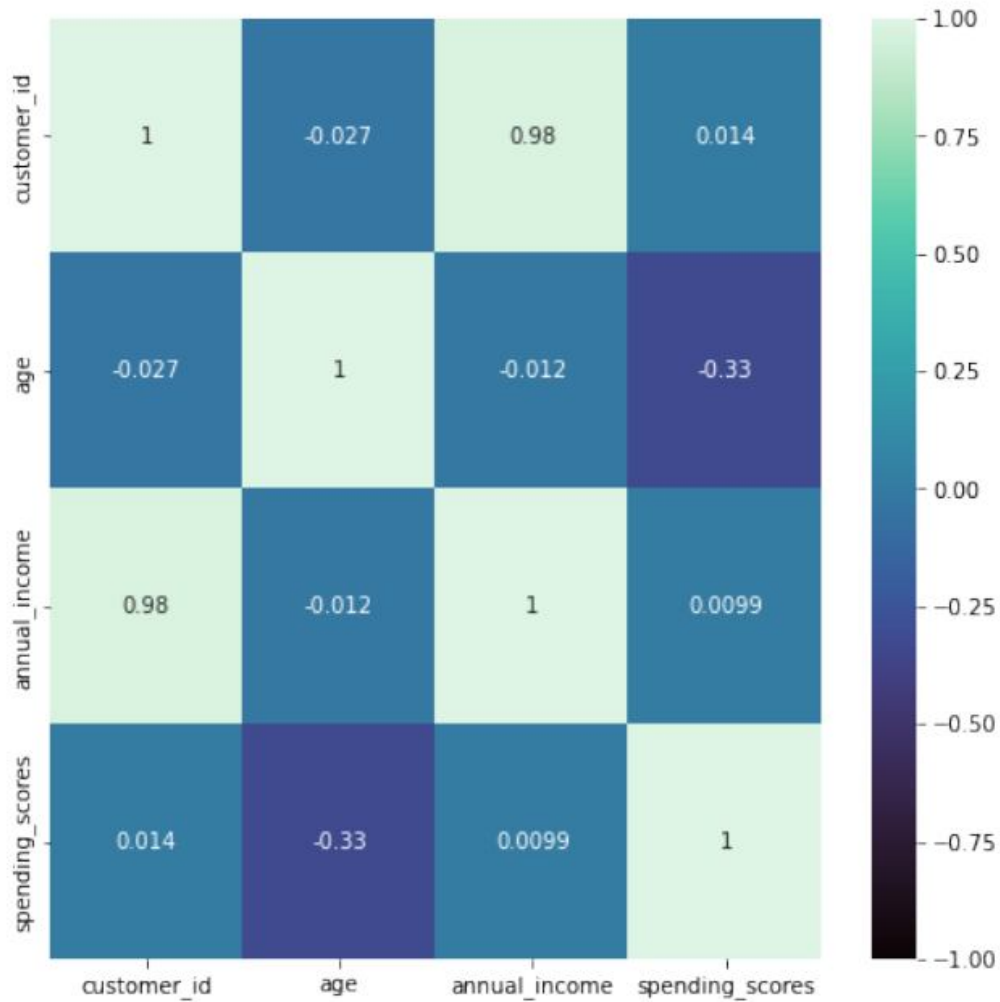


```
sns.boxplot(x=data['gender'],y=data['annual_income'])  
<matplotlib.axes._subplots.AxesSubplot at 0x7fe743f23250>
```



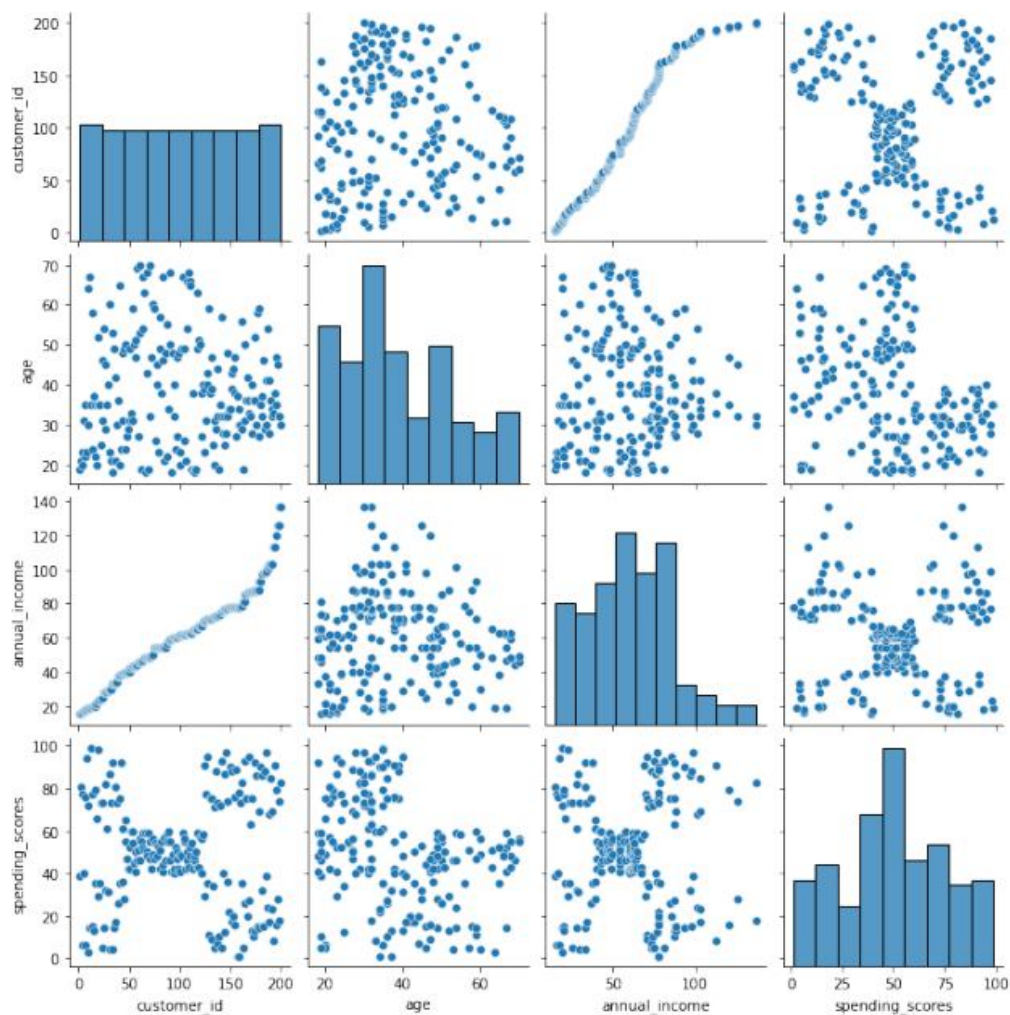
Coorelation Plot

```
corr=data.corr()
plt.figure(figsize=(8,8))
sn=sns.heatmap(corr,vmin=-1,center=0, annot = True, cmap = 'mako')
```



```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7fe7438d8310>
```



4. Perform descriptive statistics on the dataset.

```
data.head(10)
```

	customer_id	gender	age	annual_income	spending_scores
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
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72

```
data.shape
```

```
(200, 5)
```

```
data.describe()
```

	customer_id	age	annual_income	spending_scores
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

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  -
0   customer_id           200 non-null   int64
1   gender                200 non-null   object
2   age                   200 non-null   int64
3   annual_income         200 non-null   int64
4   spending_scores       200 non-null   int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

5. Check for Missing values and deal with them.

```
data[data.duplicated()]
```

```
Empty DataFrame
Columns: [customer_id, gender, age, annual_income, spending_scores]
Index: []
```

```
data.isna().sum()
```

```
customer_id      0
gender           0
age              0
annual_income    0
spending_scores  0
dtype: int64
```

there is no missing values and duplicates in dataframe

6. Find the outliers and replace them outliers

```
for i in data:
    if data[i].dtype=='int64':
        q1=data[i].quantile(0.25)
```



```

q3=data[i].quantile(0.75)
iqr=q3-q1
upper=q3+1.5*iqr
lower=q1-1.5*iqr
data[i]=np.where(data[i] >upper, upper, data[i])
data[i]=np.where(data[i] <lower, lower, data[i])

```

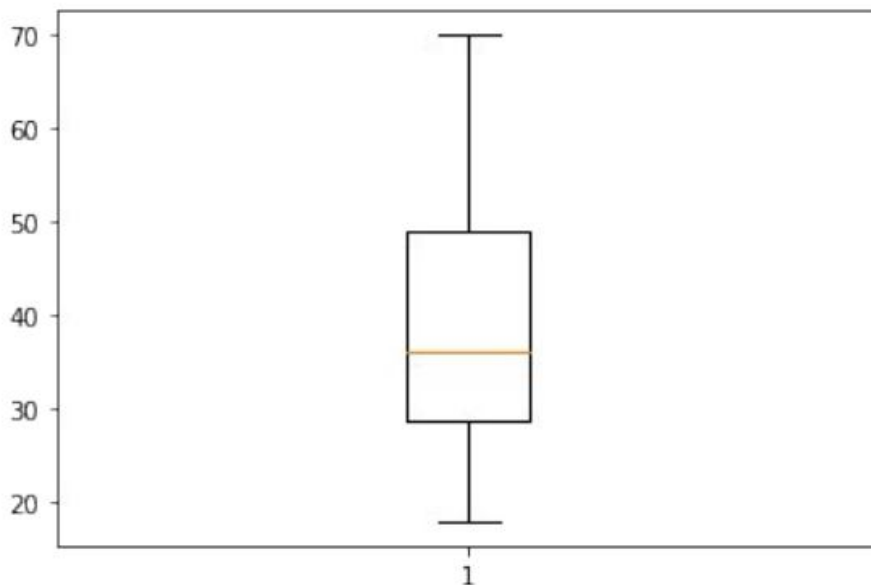
After removing outliers, boxplot will be like

```
plt.boxplot(data['age'])
```

```

{'whiskers': [<matplotlib.lines.Line2D at 0x7fe7418d49d0>,
<matplotlib.lines.Line2D at 0x7fe7418d4f10>],
'caps': [<matplotlib.lines.Line2D at 0x7fe7418d9490>,
<matplotlib.lines.Line2D at 0x7fe7418d99d0>],
'boxes': [<matplotlib.lines.Line2D at 0x7fe7418d4410>],
'medians': [<matplotlib.lines.Line2D at 0x7fe7418d9f50>],
'fliers': [<matplotlib.lines.Line2D at 0x7fe7418de4d0>],
'means': []}

```

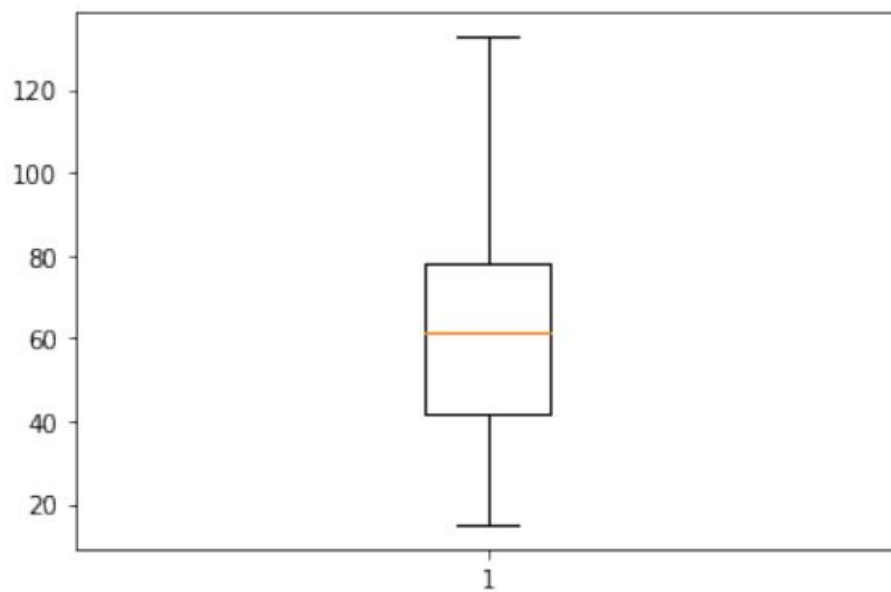


```
plt.boxplot(data['annual_income'])
```

```

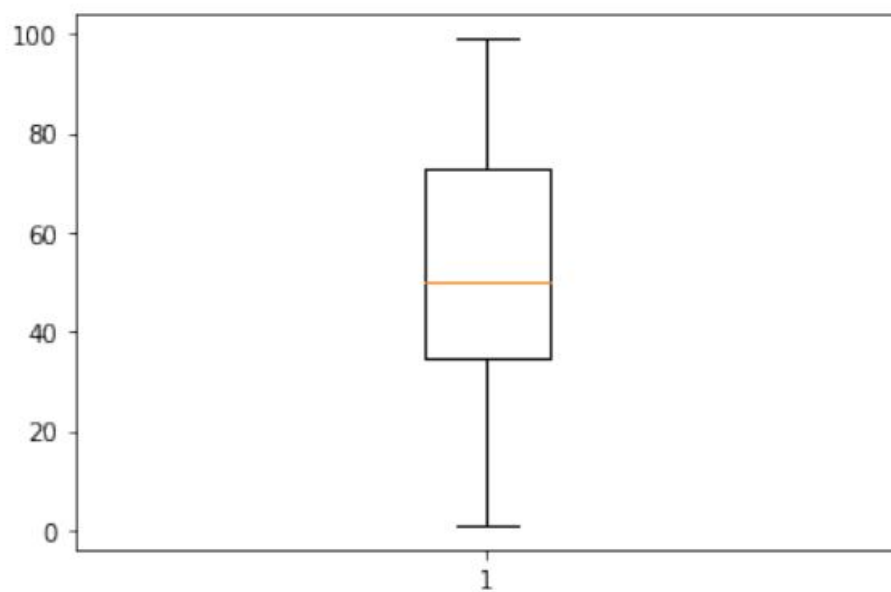
{'whiskers': [<matplotlib.lines.Line2D at 0x7fe741838510>,
<matplotlib.lines.Line2D at 0x7fe741838a50>],
'caps': [<matplotlib.lines.Line2D at 0x7fe741838f90>,
<matplotlib.lines.Line2D at 0x7fe74183e510>],
'boxes': [<matplotlib.lines.Line2D at 0x7fe741831f50>],
'medians': [<matplotlib.lines.Line2D at 0x7fe74183ea90>],
'fliers': [<matplotlib.lines.Line2D at 0x7fe74183efd0>],
'means': []}

```



```
plt.boxplot(data['spending_scores'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fe741823f50>,
<matplotlib.lines.Line2D at 0x7fe7417a74d0>],
'caps': [<matplotlib.lines.Line2D at 0x7fe7417a7a10>,
<matplotlib.lines.Line2D at 0x7fe7417a7f50>],
'boxes': [<matplotlib.lines.Line2D at 0x7fe7418239d0>],
'medians': [<matplotlib.lines.Line2D at 0x7fe7417af510>],
'fliers': [<matplotlib.lines.Line2D at 0x7fe7417afa50>],
'means': []}
```



7. Check for Categorical columns and perform encoding.

```
from sklearn.preprocessing import LabelEncoder
encoder=LabelEncoder()
data['gender']=encoder.fit_transform(data['gender'])
data.head()
```

	customer_id	gender	age	annual_income	spending_scores
0	1.0	1	19.0	15.0	39.0
1	2.0	1	21.0	15.0	81.0
2	3.0	0	20.0	16.0	6.0
3	4.0	0	23.0	16.0	77.0
4	5.0	0	31.0	17.0	40.0

8. Scaling the data

```
from sklearn.preprocessing import StandardScaler
df=StandardScaler()
data1=df.fit_transform(data)
```

data1

```
array([[ -1.7234121,  1.12815215, -1.42456879, -1.74542941, -
 0.43480148],
       [ -1.70609137,  1.12815215, -1.28103541, -1.74542941,
 1.19570407],
       [ -1.68877065, -0.88640526, -1.3528021 , -1.70708307, -
 1.71591298],
       [ -1.67144992, -0.88640526, -1.13750203, -1.70708307,
 1.04041783],
       [ -1.6541292 , -0.88640526, -0.56336851, -1.66873673, -
 0.39597992],
       [ -1.63680847, -0.88640526, -1.20926872, -1.66873673,
 1.00159627],
       [ -1.61948775, -0.88640526, -0.27630176, -1.6303904 , -
 1.71591298],
       [ -1.60216702, -0.88640526, -1.13750203, -1.6303904 ,
 1.70038436],
       [ -1.5848463 ,  1.12815215,  1.80493225, -1.59204406, -
 1.83237767],
       [ -1.56752558, -0.88640526, -0.6351352 , -1.59204406,
 0.84631002],
       [ -1.55020485,  1.12815215,  2.02023231, -1.59204406, -1.4053405
 ],
       [ -1.53288413, -0.88640526, -0.27630176, -1.59204406,
 1.89449216],
       [ -1.5155634 , -0.88640526,  1.37433211, -1.55369772, -
 1.36651894],
       [ -1.49824268, -0.88640526, -1.06573534, -1.55369772,
 1.04041783],
       [ -1.48092195,  1.12815215, -0.13276838, -1.55369772, -
```

1.44416206],
 [-1.46360123, 1.12815215, -1.20926872, -1.55369772,
 1.11806095],
 [-1.4462805 , -0.88640526, -0.27630176, -1.51535138, -
 0.59008772],
 [-1.42895978, 1.12815215, -1.3528021 , -1.51535138,
 0.61338066],
 [-1.41163905, 1.12815215, 0.94373197, -1.43865871, -
 0.82301709],
 [-1.39431833, -0.88640526, -0.27630176, -1.43865871, 1.8556706
],
 [-1.3769976 , 1.12815215, -0.27630176, -1.40031237, -
 0.59008772],
 [-1.35967688, 1.12815215, -0.99396865, -1.40031237,
 0.88513158],
 [-1.34235616, -0.88640526, 0.51313183, -1.36196603, -
 1.75473454],
 [-1.32503543, 1.12815215, -0.56336851, -1.36196603,
 0.88513158],
 [-1.30771471, -0.88640526, 1.08726535, -1.24692702, -1.4053405
],
 [-1.29039398, 1.12815215, -0.70690189, -1.24692702,
 1.23452563],
 [-1.27307326, -0.88640526, 0.44136514, -1.24692702, -0.7065524
],
 [-1.25575253, 1.12815215, -0.27630176, -1.24692702,
 0.41927286],
 [-1.23843181, -0.88640526, 0.08253169, -1.20858069, -
 0.74537397],
 [-1.22111108, -0.88640526, -1.13750203, -1.20858069,
 1.42863343],
 [-1.20379036, 1.12815215, 1.51786549, -1.17023435, -1.7935561
],
 [-1.18646963, -0.88640526, -1.28103541, -1.17023435,
 0.88513158],
 [-1.16914891, 1.12815215, 1.01549866, -1.05519534, -1.7935561
],
 [-1.15182818, 1.12815215, -1.49633548, -1.05519534,
 1.62274124],
 [-1.13450746, -0.88640526, 0.7284319 , -1.05519534, -1.4053405
],
 [-1.11718674, -0.88640526, -1.28103541, -1.05519534,
 1.19570407],
 [-1.09986601, -0.88640526, 0.22606507, -1.016849 , -
 1.28887582],
 [-1.08254529, -0.88640526, -0.6351352 , -1.016849 ,
 0.88513158],
 [-1.06522456, -0.88640526, -0.20453507, -0.90180999, -
 0.93948177],
 [-1.04790384, -0.88640526, -1.3528021 , -0.90180999,

0.96277471],
 [-1.03058311, -0.88640526, 1.87669894, -0.86346365, -
 0.59008772],
 [-1.01326239, 1.12815215, -1.06573534, -0.86346365,
 1.62274124],
 [-0.99594166, 1.12815215, 0.65666521, -0.82511731, -
 0.55126616],
 [-0.97862094, -0.88640526, -0.56336851, -0.82511731,
 0.41927286],
 [-0.96130021, -0.88640526, 0.7284319 , -0.82511731, -
 0.86183865],
 [-0.94397949, -0.88640526, -1.06573534, -0.82511731, 0.5745591
],
 [-0.92665877, -0.88640526, 0.80019859, -0.78677098,
 0.18634349],
 [-0.90933804, -0.88640526, -0.85043527, -0.78677098, -
 0.12422899],
 [-0.89201732, -0.88640526, -0.70690189, -0.78677098, -0.3183368
],
 [-0.87469659, -0.88640526, -0.56336851, -0.78677098, -0.3183368
],
 [-0.85737587, -0.88640526, 0.7284319 , -0.7100783 ,
 0.06987881],
 [-0.84005514, 1.12815215, -0.41983513, -0.7100783 ,
 0.38045129],
 [-0.82273442, -0.88640526, -0.56336851, -0.67173196,
 0.14752193],
 [-0.80541369, 1.12815215, 1.4460988 , -0.67173196,
 0.38045129],
 [-0.78809297, -0.88640526, 0.80019859, -0.67173196, -
 0.20187212],
 [-0.77077224, 1.12815215, 0.58489852, -0.67173196, -
 0.35715836],
 [-0.75345152, -0.88640526, 0.87196528, -0.63338563, -
 0.00776431],
 [-0.73613079, 1.12815215, 2.16376569, -0.63338563, -
 0.16305055],
 [-0.71881007, -0.88640526, -0.85043527, -0.55669295,
 0.03105725],
 [-0.70148935, 1.12815215, 1.01549866, -0.55669295, -
 0.16305055],
 [-0.68416862, 1.12815215, 2.23553238, -0.55669295,
 0.22516505],
 [-0.6668479 , 1.12815215, -1.42456879, -0.55669295,
 0.18634349],
 [-0.64952717, -0.88640526, 2.02023231, -0.51834661,
 0.06987881],
 [-0.63220645, -0.88640526, 1.08726535, -0.51834661,
 0.34162973],
 [-0.61488572, 1.12815215, 1.73316556, -0.48000028,

0.03105725],
 [-0.597565 , 1.12815215, -1.49633548, -0.48000028,
 0.34162973],
 [-0.58024427, -0.88640526, 0.29783176, -0.48000028, -
 0.00776431],
 [-0.56292355, -0.88640526, 2.091999 , -0.48000028, -
 0.08540743],
 [-0.54560282, 1.12815215, -1.42456879, -0.48000028,
 0.34162973],
 [-0.5282821 , -0.88640526, -0.49160182, -0.48000028, -
 0.12422899],
 [-0.51096138, 1.12815215, 2.23553238, -0.44165394,
 0.18634349],
 [-0.49364065, -0.88640526, 0.58489852, -0.44165394, -0.3183368
],
 [-0.47631993, -0.88640526, 1.51786549, -0.4033076 , -
 0.04658587],
 [-0.4589992 , -0.88640526, 1.51786549, -0.4033076 ,
 0.22516505],
 [-0.44167848, 1.12815215, 1.4460988 , -0.24992225, -
 0.12422899],
 [-0.42435775, 1.12815215, -0.92220196, -0.24992225,
 0.14752193],
 [-0.40703703, -0.88640526, 0.44136514, -0.24992225,
 0.10870037],
 [-0.3897163 , 1.12815215, 0.08253169, -0.24992225, -
 0.08540743],
 [-0.37239558, -0.88640526, -1.13750203, -0.24992225,
 0.06987881],
 [-0.35507485, -0.88640526, 0.7284319 , -0.24992225, -0.3183368
],
 [-0.33775413, 1.12815215, 1.30256542, -0.24992225,
 0.03105725],
 [-0.3204334 , 1.12815215, -0.06100169, -0.24992225,
 0.18634349],
 [-0.30311268, 1.12815215, 2.02023231, -0.24992225, -
 0.35715836],
 [-0.28579196, -0.88640526, 0.51313183, -0.24992225, -
 0.24069368],
 [-0.26847123, -0.88640526, -1.28103541, -0.24992225,
 0.26398661],
 [-0.25115051, 1.12815215, 0.65666521, -0.24992225, -
 0.16305055],
 [-0.23382978, -0.88640526, 1.15903204, -0.13488324,
 0.30280817],
 [-0.21650906, -0.88640526, -1.20926872, -0.13488324,
 0.18634349],
 [-0.19918833, -0.88640526, -0.34806844, -0.0965369 ,
 0.38045129],
 [-0.18186761, -0.88640526, 0.80019859, -0.0965369 , -

0.16305055],
 [-0.16454688, -0.88640526, 2.091999 , -0.05819057,
 0.18634349],
 [-0.14722616, 1.12815215, -1.49633548, -0.05819057, -
 0.35715836],
 [-0.12990543, 1.12815215, 0.65666521, -0.01984423, -
 0.04658587],
 [-0.11258471, -0.88640526, 0.08253169, -0.01984423, -
 0.39597992],
 [-0.09526399, -0.88640526, -0.49160182, -0.01984423, -0.3183368
],
 [-0.07794326, 1.12815215, -1.06573534, -0.01984423,
 0.06987881],
 [-0.06062254, -0.88640526, 0.58489852, -0.01984423, -
 0.12422899],
 [-0.04330181, -0.88640526, -0.85043527, -0.01984423, -
 0.00776431],
 [-0.02598109, 1.12815215, 0.65666521, 0.01850211, -0.3183368
],
 [-0.00866036, 1.12815215, -1.3528021 , 0.01850211, -
 0.04658587],
 [0.00866036, -0.88640526, -1.13750203, 0.05684845, -
 0.35715836],
 [0.02598109, -0.88640526, 0.7284319 , 0.05684845, -
 0.08540743],
 [0.04330181, 1.12815215, 2.02023231, 0.05684845,
 0.34162973],
 [0.06062254, 1.12815215, -0.92220196, 0.05684845,
 0.18634349],
 [0.07794326, 1.12815215, 0.7284319 , 0.05684845,
 0.22516505],
 [0.09526399, -0.88640526, -1.28103541, 0.05684845, -0.3183368
],
 [0.11258471, -0.88640526, 1.94846562, 0.09519478, -
 0.00776431],
 [0.12990543, 1.12815215, 1.08726535, 0.09519478, -
 0.16305055],
 [0.14722616, 1.12815215, 2.091999 , 0.09519478, -
 0.27951524],
 [0.16454688, 1.12815215, 1.94846562, 0.09519478, -
 0.08540743],
 [0.18186761, 1.12815215, 1.87669894, 0.09519478,
 0.06987881],
 [0.19918833, -0.88640526, -1.42456879, 0.09519478,
 0.14752193],
 [0.21650906, -0.88640526, -0.06100169, 0.13354112, -0.3183368
],
 [0.23382978, 1.12815215, -1.42456879, 0.13354112, -
 0.16305055],
 [0.25115051, -0.88640526, -1.49633548, 0.17188746, -

0.08540743],
 [0.26847123, -0.88640526, -1.42456879, 0.17188746, -
 0.00776431],
 [0.28579196, -0.88640526, 1.73316556, 0.17188746, -
 0.27951524],
 [0.30311268, -0.88640526, 0.7284319, 0.17188746,
 0.34162973],
 [0.3204334, -0.88640526, 0.87196528, 0.24858013, -
 0.27951524],
 [0.33775413, -0.88640526, 0.80019859, 0.24858013,
 0.26398661],
 [0.35507485, 1.12815215, -0.85043527, 0.24858013,
 0.22516505],
 [0.37239558, -0.88640526, -0.06100169, 0.24858013, -
 0.39597992],
 [0.3897163, -0.88640526, 0.08253169, 0.32527281,
 0.30280817],
 [0.40703703, 1.12815215, 0.010765, 0.32527281,
 1.58391968],
 [0.42435775, -0.88640526, -1.13750203, 0.36361914, -
 0.82301709],
 [0.44167848, -0.88640526, -0.56336851, 0.36361914,
 1.04041783],
 [0.4589992, 1.12815215, 0.29783176, 0.40196548, -
 0.59008772],
 [0.47631993, 1.12815215, 0.08253169, 0.40196548,
 1.73920592],
 [0.49364065, 1.12815215, 1.4460988, 0.40196548, -
 1.52180518],
 [0.51096138, 1.12815215, -0.06100169, 0.40196548,
 0.96277471],
 [0.5282821, 1.12815215, 0.58489852, 0.40196548, -1.5994483
],
 [0.54560282, 1.12815215, 0.010765, 0.40196548,
 0.96277471],
 [0.56292355, -0.88640526, -0.99396865, 0.44031182, -
 0.62890928],
 [0.58024427, -0.88640526, -0.56336851, 0.44031182,
 0.80748846],
 [0.597565, 1.12815215, -1.3528021, 0.47865816, -
 1.75473454],
 [0.61488572, -0.88640526, -0.70690189, 0.47865816,
 1.46745499],
 [0.63220645, -0.88640526, 0.36959845, 0.47865816, -
 1.67709142],
 [0.64952717, 1.12815215, -0.49160182, 0.47865816,
 0.88513158],
 [0.6668479, 1.12815215, -1.42456879, 0.51700449, -
 1.56062674],
 [0.68416862, -0.88640526, -0.27630176, 0.51700449,

0.84631002],
 [0.70148935, -0.88640526, 1.30256542, 0.55535083, -
 1.75473454],
 [0.71881007, 1.12815215, -0.49160182, 0.55535083, 1.6615628
],
 [0.73613079, -0.88640526, -0.77866858, 0.59369717, -
 0.39597992],
 [0.75345152, -0.88640526, -0.49160182, 0.59369717,
 1.42863343],
 [0.77077224, 1.12815215, -0.99396865, 0.6320435 , -
 1.48298362],
 [0.78809297, 1.12815215, -0.77866858, 0.6320435 ,
 1.81684904],
 [0.80541369, 1.12815215, 0.65666521, 0.6320435 , -
 0.55126616],
 [0.82273442, -0.88640526, -0.49160182, 0.6320435 ,
 0.92395314],
 [0.84005514, -0.88640526, -0.34806844, 0.67038984, -
 1.09476801],
 [0.85737587, 1.12815215, -0.34806844, 0.67038984,
 1.54509812],
 [0.87469659, 1.12815215, 0.29783176, 0.67038984, -
 1.28887582],
 [0.89201732, 1.12815215, 0.010765 , 0.67038984,
 1.46745499],
 [0.90933804, -0.88640526, 0.36959845, 0.67038984, -
 1.17241113],
 [0.92665877, -0.88640526, -0.06100169, 0.67038984,
 1.00159627],
 [0.94397949, -0.88640526, 0.58489852, 0.67038984, -
 1.32769738],
 [0.96130021, -0.88640526, -0.85043527, 0.67038984,
 1.50627656],
 [0.97862094, 1.12815215, -0.13276838, 0.67038984, -
 1.91002079],
 [0.99594166, -0.88640526, -0.6351352 , 0.67038984,
 1.07923939],
 [1.01326239, 1.12815215, -0.34806844, 0.67038984, -
 1.91002079],
 [1.03058311, -0.88640526, -0.6351352 , 0.67038984,
 0.88513158],
 [1.04790384, -0.88640526, 1.23079873, 0.70873618, -
 0.59008772],
 [1.06522456, -0.88640526, -0.70690189, 0.70873618,
 1.27334719],
 [1.08254529, 1.12815215, -1.42456879, 0.78542885, -
 1.75473454],
 [1.09986601, -0.88640526, -0.56336851, 0.78542885, 1.6615628
],
 [1.11718674, 1.12815215, 0.80019859, 0.9388142 , -

0.93948177],
 [1.13450746, -0.88640526, -0.20453507, 0.9388142 ,
 0.96277471],
 [1.15182818, 1.12815215, 0.22606507, 0.97716054, -
 1.17241113],
 [1.16914891, -0.88640526, -0.41983513, 0.97716054,
 1.73920592],
 [1.18646963, -0.88640526, -0.20453507, 1.01550688, -
 0.90066021],
 [1.20379036, 1.12815215, -0.49160182, 1.01550688,
 0.49691598],
 [1.22111108, 1.12815215, 0.08253169, 1.01550688, -
 1.44416206],
 [1.23843181, 1.12815215, -0.77866858, 1.01550688,
 0.96277471],
 [1.25575253, 1.12815215, -0.20453507, 1.01550688, -
 1.56062674],
 [1.27307326, 1.12815215, -0.20453507, 1.01550688,
 1.62274124],
 [1.29039398, -0.88640526, 0.94373197, 1.05385321, -
 1.44416206],
 [1.30771471, -0.88640526, -0.6351352 , 1.05385321,
 1.38981187],
 [1.32503543, 1.12815215, 1.37433211, 1.05385321, -
 1.36651894],
 [1.34235616, 1.12815215, -0.85043527, 1.05385321,
 0.72984534],
 [1.35967688, 1.12815215, 1.4460988 , 1.2455849 , -1.4053405
],
 [1.3769976 , 1.12815215, -0.27630176, 1.2455849 ,
 1.54509812],
 [1.39431833, -0.88640526, -0.13276838, 1.39897025, -0.7065524
],
 [1.41163905, -0.88640526, -0.49160182, 1.39897025,
 1.38981187],
 [1.42895978, 1.12815215, 0.51313183, 1.43731659, -
 1.36651894],
 [1.4462805 , -0.88640526, -0.70690189, 1.43731659,
 1.46745499],
 [1.46360123, -0.88640526, 0.15429838, 1.47566292, -
 0.43480148],
 [1.48092195, 1.12815215, -0.6351352 , 1.47566292,
 1.81684904],
 [1.49824268, -0.88640526, 1.08726535, 1.5523556 , -
 1.01712489],
 [1.5155634 , 1.12815215, -0.77866858, 1.5523556 ,
 0.69102378],
 [1.53288413, -0.88640526, 0.15429838, 1.62904827, -
 1.28887582],
 [1.55020485, -0.88640526, -0.20453507, 1.62904827,

```

1.35099031],
  [ 1.56752558, -0.88640526, -0.34806844, 1.62904827, -
1.05594645],
  [ 1.5848463 , -0.88640526, -0.49160182, 1.62904827,
0.72984534],
  [ 1.60216702, 1.12815215, -0.41983513, 2.01251165, -
1.63826986],
  [ 1.61948775, -0.88640526, -0.06100169, 2.01251165,
1.58391968],
  [ 1.63680847, -0.88640526, 0.58489852, 2.28093601, -
1.32769738],
  [ 1.6541292 , -0.88640526, -0.27630176, 2.28093601,
1.11806095],
  [ 1.67144992, -0.88640526, 0.44136514, 2.51101403, -
0.86183865],
  [ 1.68877065, 1.12815215, -0.49160182, 2.51101403,
0.92395314],
  [ 1.70609137, 1.12815215, -0.49160182, 2.76985181, -
1.25005425],
  [ 1.7234121 , 1.12815215, -0.6351352 , 2.76985181,
1.27334719]])

```

9. Perform any of the clustering algorithms

```

from sklearn.cluster import KMeans
data.drop('customer_id',axis=1,inplace=True)
km = KMeans(n_clusters=3, random_state=0)
data['Group or Cluster'] = km.fit_predict(data)
data.head()

```

	gender	age	annual_income	spending_scores	Group or Cluster
0	1	19.0	15.0	39.0	2
1	1	21.0	15.0	81.0	2
2	0	20.0	16.0	6.0	2
3	0	23.0	16.0	77.0	2
4	0	31.0	17.0	40.0	2

```

data['Group or Cluster'].value_counts()
2    123
1     39
0     38
Name: Group or Cluster, dtype: int64
import matplotlib.pyplot as plt
fig,ax = plt.subplots(figsize=(15,8))
sns.scatterplot(x=data['annual_income'],

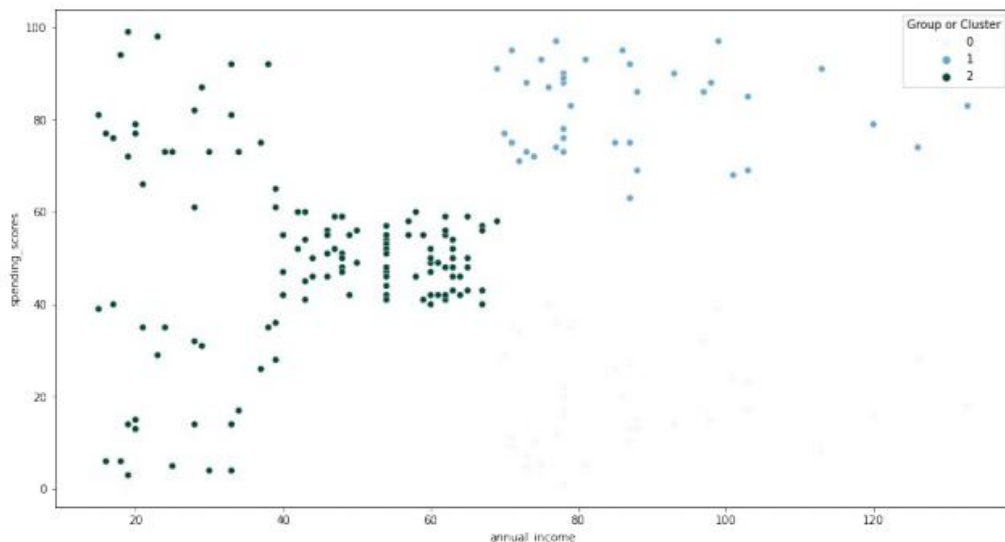
```

```

y=data['spending_scores'],
hue=data['Group or Cluster'],
palette='PuBuGn')

plt.show()

```



```

from sklearn.metrics import silhouette_score, silhouette_samples
score = silhouette_score(data,
                        km.labels_,
                        metric='euclidean')

```

score

0.3842057644019546

```

import matplotlib.pyplot as plt
from yellowbrick.cluster import SilhouetteVisualizer

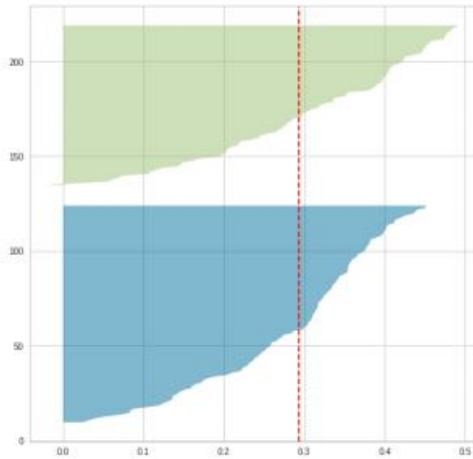
```

```

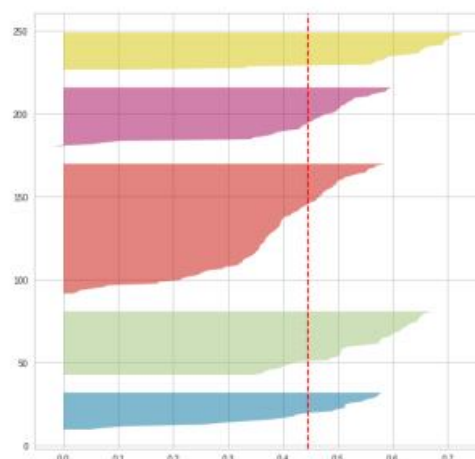
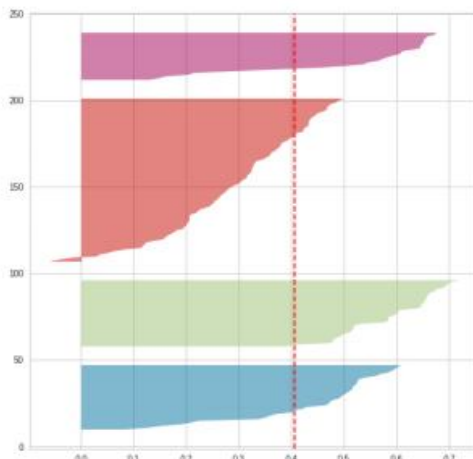
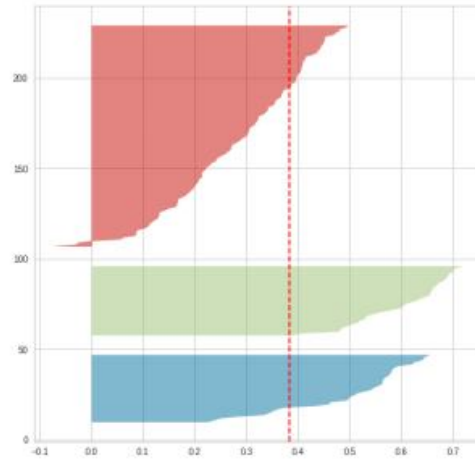
fig, ax = plt.subplots(2, 2, figsize=(20,20))
for i in [2, 3, 4, 5]:
    """
    Create KMeans instance for different number of clusters
    """
    km = KMeans(n_clusters=i,
                init='k-means++',
                n_init=10,
                max_iter=100,
                random_state=0)
    q, mod = divmod(i, 2)
    """
    Create SilhouetteVisualizer instance with KMeans instance
    Fit the visualizer
    """
    visualizer = SilhouetteVisualizer(km,

```

```
visualizer.fit(data)
```



```
colors='yellowbrick',  
ax=ax[q-1][mod])
```



11. Split the data into dependent and independent variables

```
x=data.iloc[:, :-1]  
x.head()
```

	gender	age	annual_income	spending_scores
0	1	19.0	15.0	39.0
1	1	21.0	15.0	81.0
2	0	20.0	16.0	6.0
3	0	23.0	16.0	77.0
4	0	31.0	17.0	40.0

```
y=data.iloc[:, -1]  
y.head()
```

```

0    2
1    2
2    2
3    2
4    2
Name: Group or Cluster, dtype: int32

```

12. Split the data into training and testing

```

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33)

x_train.shape
(134, 4)

x_test.shape
(66, 4)

y_train.shape
(134,)

y_test.shape
(66,)

```

13. Build the Model

```

from sklearn.linear_model import LogisticRegression
lgr = LogisticRegression()
lgr.fit(x_train,y_train)

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
    extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

LogisticRegression()

```

15. Test the Model

```

y_pred = lgr.predict(x_test)
print(y_pred)

[0 2 2 2 2 2 1 2 2 0 2 2 2 2 0 2 1 0 1 2 0 2 2 0 2 1 1 2 1 2 2 2 1 2 2
 2 2
 0 2 0 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 0 2 0 1 0 1 0 1 0]

```

16. Measure the performance using Evaluation Metrics

```

from sklearn.metrics import accuracy_score
score = accuracy_score(y_test,y_pred)
print(score)

1.0

```