K-Mean Clustering (Customer Segmentation)

March 2, 2024

1 Customer Segmentation using K-Mean Clustering

2 Import Libraries

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import re
from sklearn.cluster import KMeans
```

3 Importing the dataset

[3]:
```
df = pd.read_csv("C:/Users/Dharini/Downloads/Mall_Customers.csv")
```

4 Data Anlaysis

[4]:
```
df.head()
```

[4]:

	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

[5]:
```
df.tail()
```

[5]:

	CustomerID	Gender	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

[8]:
```
df.shape
```

[8]:
```
(200, 5)
```

[9]:
```
df.isnull().sum()
```

[9]:
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

[10]:
```
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   Gender                  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
```

- Conclusions

1. No Null Value
2. Gender has the dtype of object, which needs to be converted

[11]:
```
df.describe()
```

[11]:

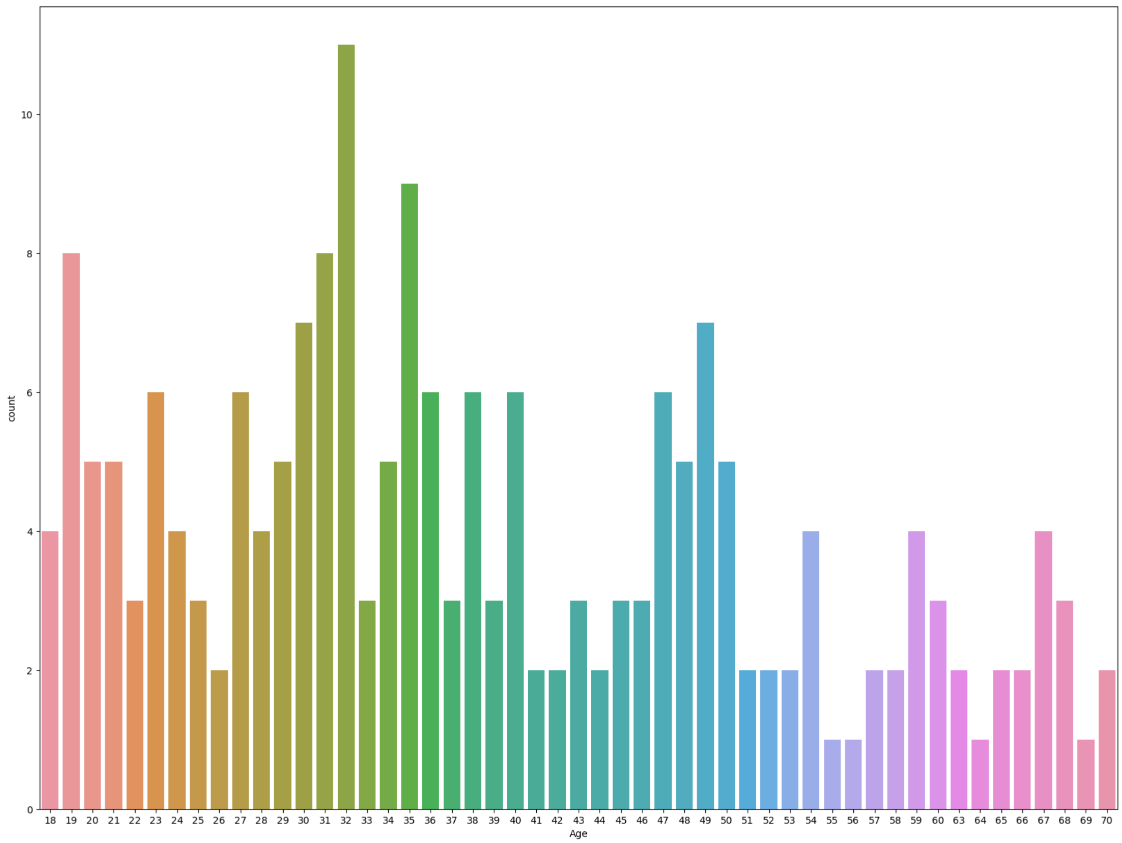
	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000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

- Conclusions

1. Avg. age of Customer is 38
2. Customer that visits the mall is having the age b/w 18 to 70
3. Having the Avg. annual income as $60k

[12]:
```
plt.figure(figsize=(20,15))
sns.countplot(data=df, x='Age')
```

[12]: <AxesSubplot: xlabel='Age', ylabel='count'>



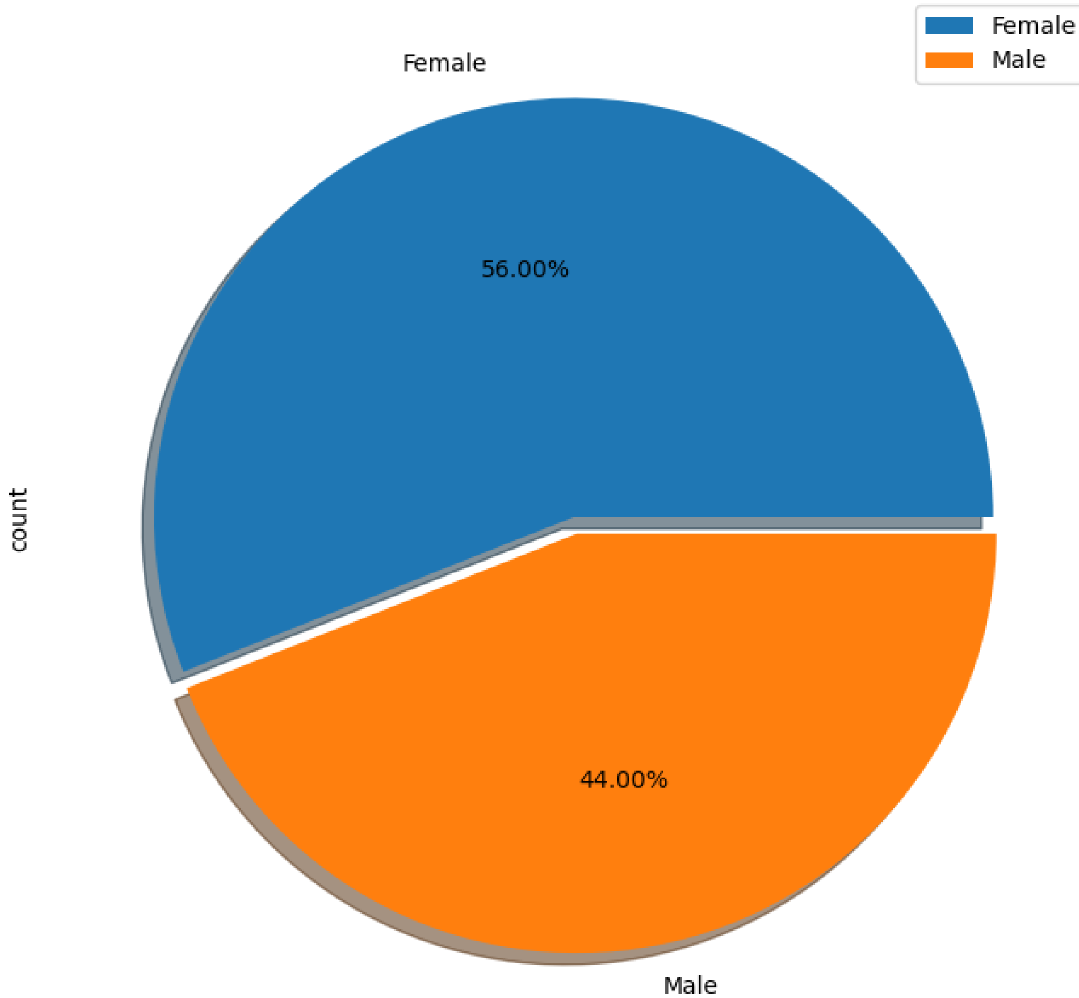
[13]:
```
df.head()
```

[13]:

	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

[14]:
```
#showing how many % of male and female visits the mall with the help of a plot
plt.figure(figsize=(8,8))
df.Gender.value_counts().plot(kind='pie', autopct='%.2f%%', shadow=True,
                              explode=(0,0.04))
plt.legend()
```

[14]: <matplotlib.legend.Legend at 0x281426ff370>



- Conclusion

1. No. of females > No. of males
2. Females are 12% more than Male
3. Female visits the mall most of the time.

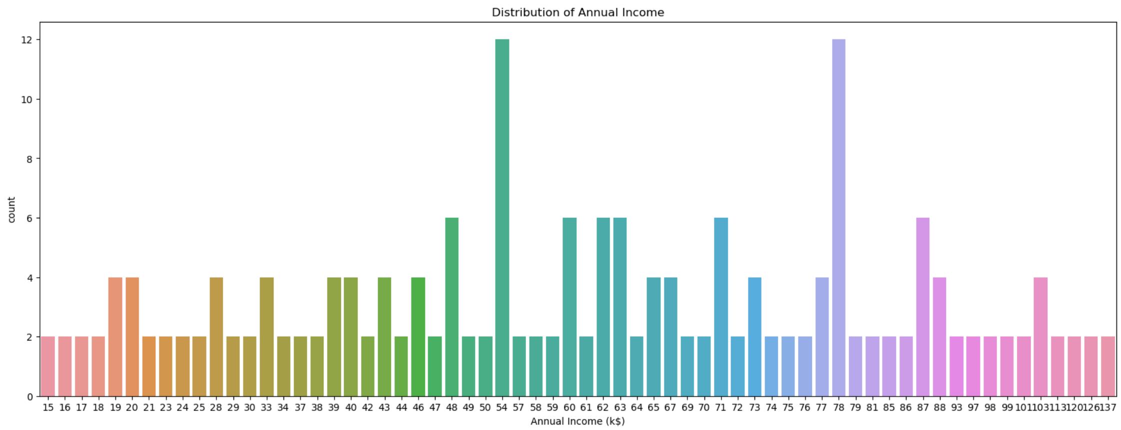
[15]:
```
df.head(2)
```

[15]:
```
   CustomerID Gender  Age  Annual Income (k$)  Spending Score (1-100)
0           1   Male   19                  15                      39
1           2   Male   21                  15                      81
```

[23]:
```
# Distribution of Income
plt.figure(figsize=(20, 7))
sns.countplot(x='Annual Income (k$)', data=df)
```

```
plt.title('Distribution of Annual Income')
```

[23]: Text(0.5, 1.0, 'Distribution of Annual Income')

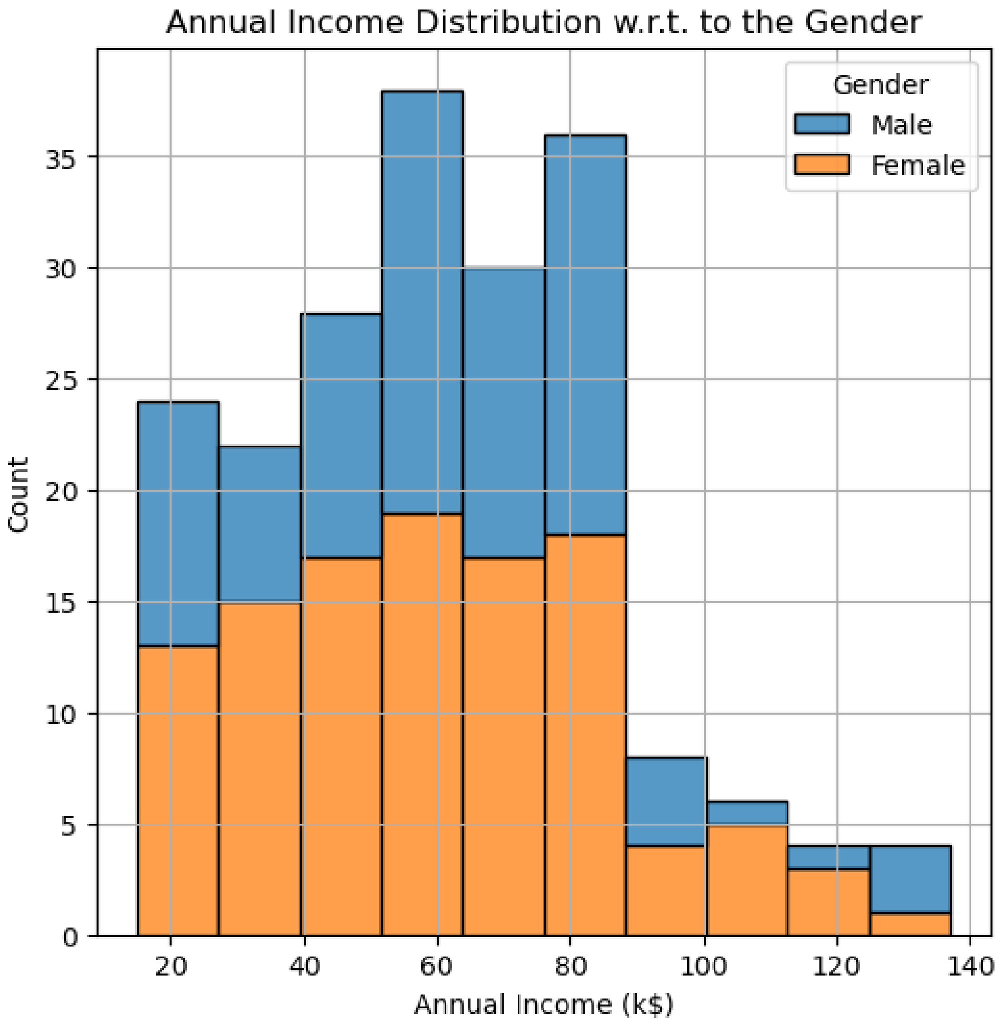


- Conclusions

1. The customers that are having the salary of $54(k) and $78(K), visits the mall most of time

[24]:
```
# Show the Annual Income Distribution with respect to the Gender

plt.figure(figsize=(6,6))
sns.histplot(df,x='Annual Income (k$)',hue='Gender', multiple='stack')
plt.title('Annual Income Distribution w.r.t. to the Gender')
plt.grid()
```



- Conclusion

The histogram provides a visual representation of the annual income distribution with respect to gender, allowing for insights into potential income disparities and informing various decision-making processes.

5 Seaborn lmplot

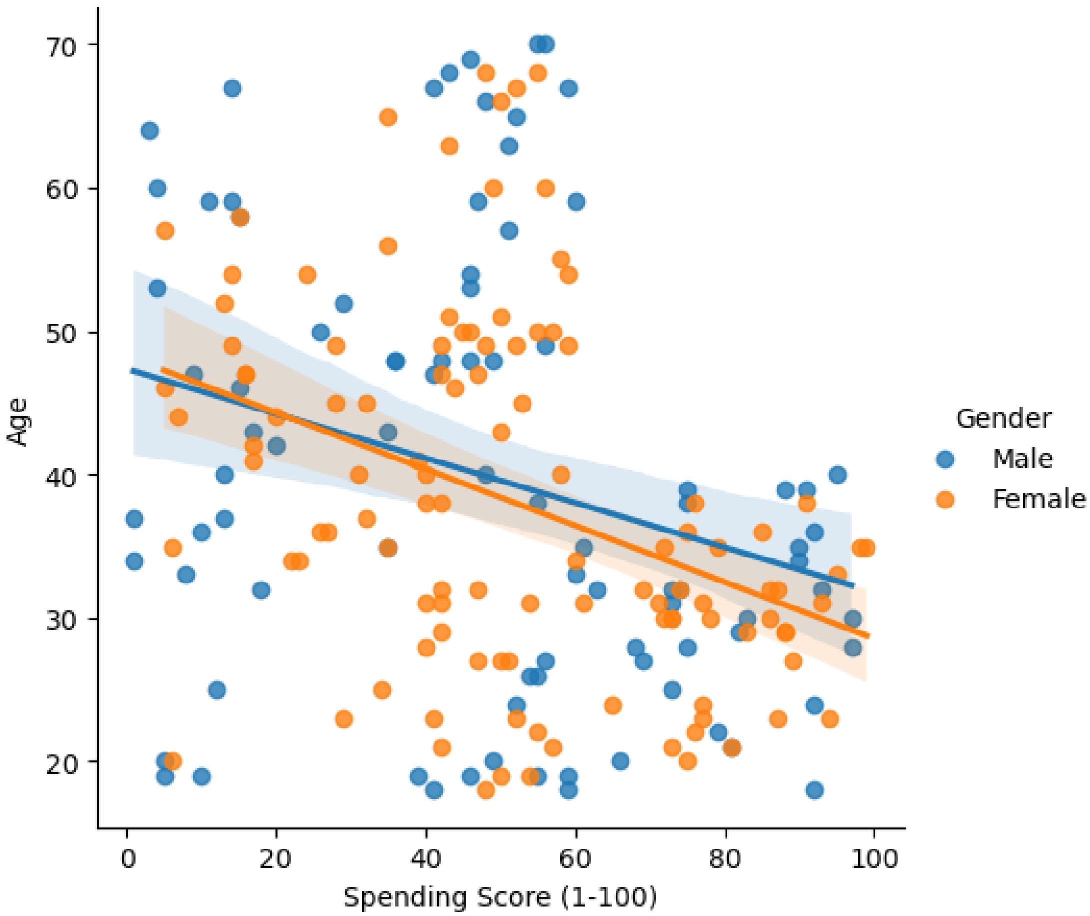
[25]:
```
df.head()
```

[25]:
```
   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
```

[26]:
```python
sns.lmplot(data=df,x='Spending Score (1-100)', y='Age', hue='Gender')
```

[26]: <seaborn.axisgrid.FacetGrid at 0x28141fce3a0>



6 Machine Learning

[27]:
```python
df.head()
```

[27]:

	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

[28]:
```python
# Label Encoding
from sklearn.preprocessing import LabelEncoder
```

[29]:
```python
enc = LabelEncoder()
```

[30]:
```python
df.Gender = enc.fit_transform(df.Gender)
```

[31]:
```python
df.head()
```

[31]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	1	19	15	39
1	2	1	21	15	81
2	3	0	20	16	6
3	4	0	23	16	77
4	5	0	31	17	40

[32]:
```python
df.drop('CustomerID', axis=1, inplace=True)
```

- K-Mean Clustering

[33]:
```python
ssd = []
# clusters will be 1-10
for i in range(1,11):
  Kmodel = KMeans(n_clusters=i, n_init=15,max_iter=500)
  Kmodel.fit(df)
  ssd.append(Kmodel.inertia_)
```

```
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1440: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1440: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1440: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1440: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1440: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less
```

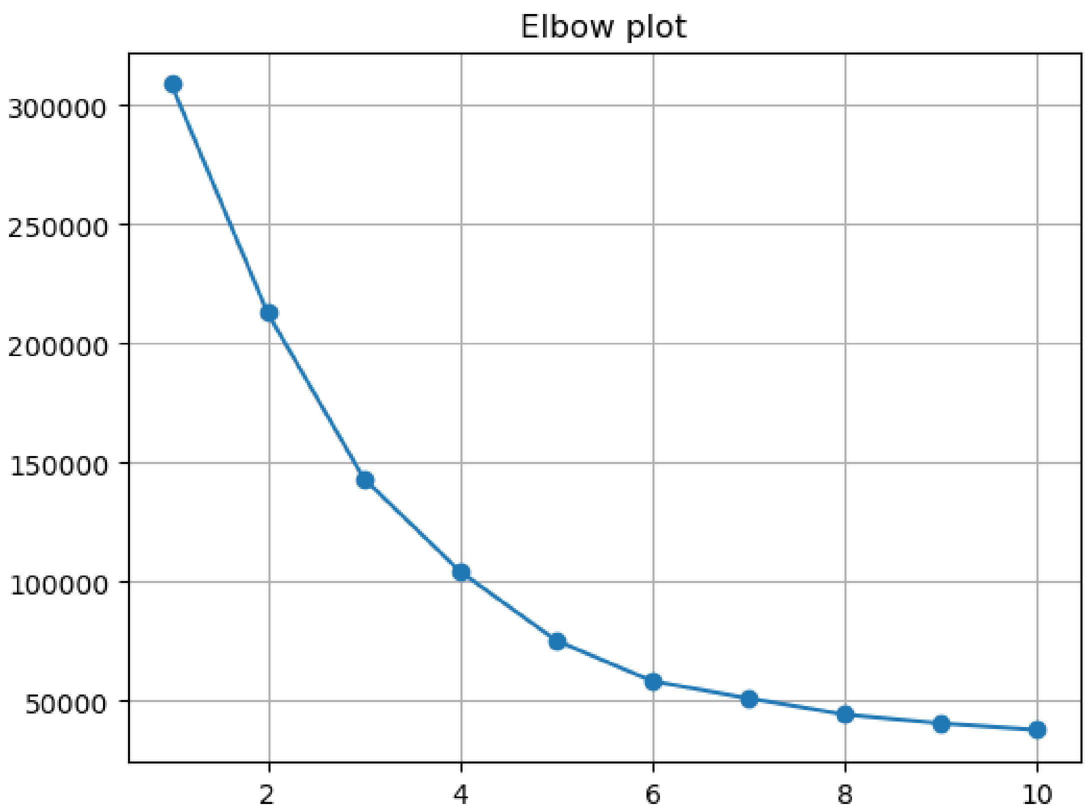
```
chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1440: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1440: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1440: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1440: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1440: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
```

[34]:
```
ssd
```

[34]:
```
[308862.06,
 212889.442455243,
 143391.59236035674,
 104414.67534220166,
 75399.61541401484,
 58348.64136331505,
 51167.19736842105,
 44355.31351771352,
 40639.660394660765,
 37895.206575973054]
```

[35]:
```
plt.plot(range(1,11), ssd, marker='o')
plt.grid()
plt.title('Elbow plot')
```

[35]:
```
Text(0.5, 1.0, 'Elbow plot')
```



- k=6 because after 6 the curve is becoming linear
- i.e we can make 6 categories of customers

[37]:
```
Kmodel = KMeans(n_clusters=6)
```

[38]:
```
Kmodel.fit(df)
# Kmodel.fit_predict(df)
# Kmodel.predict(df)
```

```
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning:
The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the
value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
D:\anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1440: UserWarning:
KMeans is known to have a memory leak on Windows with MKL, when there are less
chunks than available threads. You can avoid it by setting the environment
variable OMP_NUM_THREADS=1.
  warnings.warn(
```

[38]:
```
KMeans(n_clusters=6)
```

[39]:
```
prediction = Kmodel.predict(df)
```

```
[40]: prediction
```

```
[40]: array([5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0,
       5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 5, 0, 4, 0, 5, 2,
       5, 0, 4, 2, 2, 2, 4, 2, 2, 4, 4, 4, 4, 4, 2, 4, 4, 2, 4, 4, 4, 2,
       4, 4, 2, 2, 4, 4, 4, 4, 4, 2, 4, 2, 2, 4, 4, 2, 4, 4, 2, 4, 4, 2,
       2, 4, 4, 2, 4, 2, 2, 2, 4, 2, 4, 2, 2, 4, 4, 2, 4, 2, 4, 4, 4, 4,
       4, 2, 2, 2, 2, 2, 4, 4, 4, 4, 2, 2, 2, 3, 2, 3, 1, 3, 1, 3, 1, 3,
       2, 3, 1, 3, 1, 3, 1, 3, 1, 3, 2, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
       1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
       1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3,
       1, 3])
```

```
[41]: len(prediction)
```

```
[41]: 200
```

```
[42]: Kmodel.cluster_centers_
```

```
[42]: array([[ 0.40909091, 25.27272727, 25.72727273, 79.36363636],
       [ 0.57142857, 41.68571429, 88.22857143, 17.28571429],
       [ 0.34210526, 27.        , 56.65789474, 49.13157895],
       [ 0.46153846, 32.69230769, 86.53846154, 82.12820513],
       [ 0.43181818, 56.34090909, 53.70454545, 49.38636364],
       [ 0.40909091, 44.31818182, 25.77272727, 20.27272727]])
```

```
[43]: len(Kmodel.cluster_centers_)
```

```
[43]: 6
```

```
[44]: df.head()
```

[44]:

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	19	15	39
1	1	21	15	81
2	0	20	16	6
3	0	23	16	77
4	0	31	17	40

```
[45]: # to see which customer belongs to which cluster
df['Cluster'] = prediction
```

```
[46]: df.head()
```

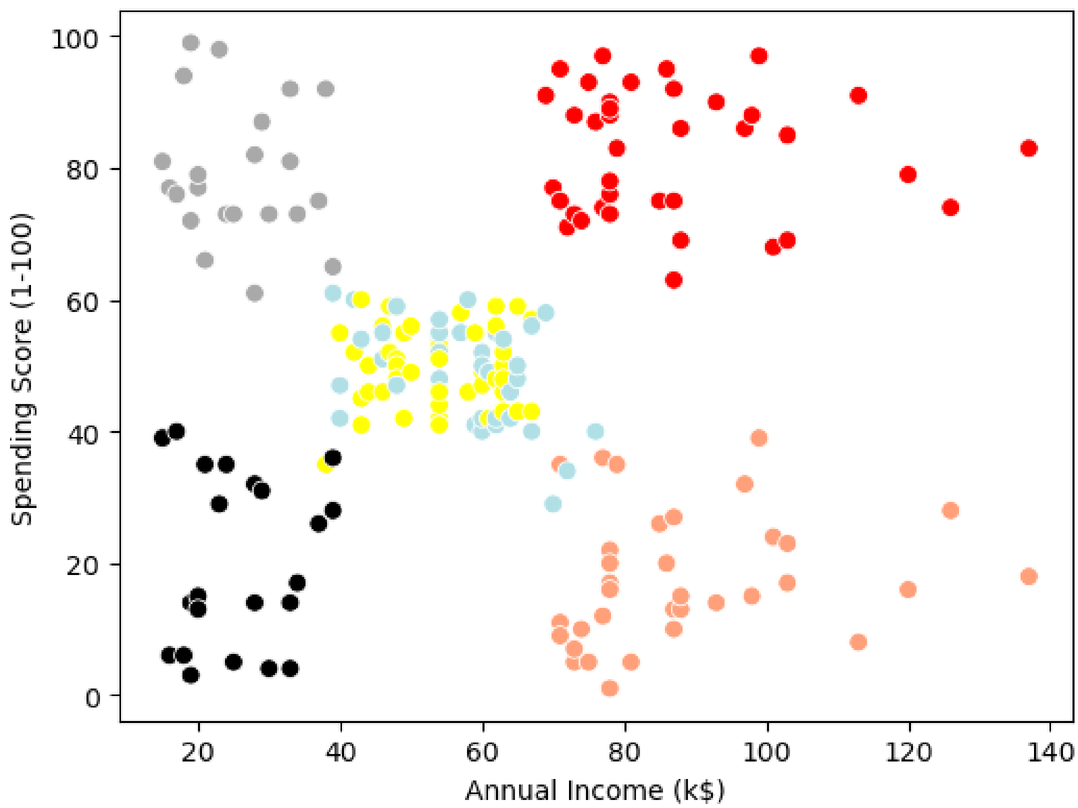
[46]:

	Gender	Age	Annual Income (k$)	Spending Score (1-100)	Cluster
0	1	19	15	39	5
1	1	21	15	81	0
2	0	20	16	6	5

```
3         0    23                    16                        77         0
4         0    31                    17                        40         5
```

[47]:
```
# visualization
color = np.array(['darkgray', 'lightsalmon', 'powderblue', 'red', 'yellow',␣
 ↪'black'])
sns.scatterplot(x = df['Annual Income (k$)'],
                y = df['Spending Score (1-100)'],
                c=color[Kmodel.labels_],
                s=50)
```

[47]:
```
<AxesSubplot: xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'>
```



[48]:
```
from scipy.cluster.hierarchy import linkage, dendrogram
```

[49]:
```
var = linkage(df, method='ward')
```

[50]:
```
plt.figure(figsize=(20,15))
dendrogram(var, leaf_rotation=90)
```

```
[50]: {'icoord': [[15.0, 15.0, 25.0, 25.0],
        [5.0, 5.0, 20.0, 20.0],
        [45.0, 45.0, 55.0, 55.0],
        [65.0, 65.0, 75.0, 75.0],
        [50.0, 50.0, 70.0, 70.0],
        [35.0, 35.0, 60.0, 60.0],
        [105.0, 105.0, 115.0, 115.0],
        [95.0, 95.0, 110.0, 110.0],
        [85.0, 85.0, 102.5, 102.5],
        [47.5, 47.5, 93.75, 93.75],
        [145.0, 145.0, 155.0, 155.0],
        [135.0, 135.0, 150.0, 150.0],
        [125.0, 125.0, 142.5, 142.5],
        [165.0, 165.0, 175.0, 175.0],
        [185.0, 185.0, 195.0, 195.0],
        [170.0, 170.0, 190.0, 190.0],
        [133.75, 133.75, 180.0, 180.0],
        [70.625, 70.625, 156.875, 156.875],
        [12.5, 12.5, 113.75, 113.75],
        [215.0, 215.0, 225.0, 225.0],
        [205.0, 205.0, 220.0, 220.0],
        [235.0, 235.0, 245.0, 245.0],
        [265.0, 265.0, 275.0, 275.0],
        [255.0, 255.0, 270.0, 270.0],
        [240.0, 240.0, 262.5, 262.5],
        [212.5, 212.5, 251.25, 251.25],
        [295.0, 295.0, 305.0, 305.0],
        [285.0, 285.0, 300.0, 300.0],
        [335.0, 335.0, 345.0, 345.0],
        [325.0, 325.0, 340.0, 340.0],
        [315.0, 315.0, 332.5, 332.5],
        [355.0, 355.0, 365.0, 365.0],
        [385.0, 385.0, 395.0, 395.0],
        [375.0, 375.0, 390.0, 390.0],
        [360.0, 360.0, 382.5, 382.5],
        [323.75, 323.75, 371.25, 371.25],
        [292.5, 292.5, 347.5, 347.5],
        [231.875, 231.875, 320.0, 320.0],
        [425.0, 425.0, 435.0, 435.0],
        [415.0, 415.0, 430.0, 430.0],
        [405.0, 405.0, 422.5, 422.5],
        [445.0, 445.0, 455.0, 455.0],
        [413.75, 413.75, 450.0, 450.0],
        [465.0, 465.0, 475.0, 475.0],
        [485.0, 485.0, 495.0, 495.0],
        [470.0, 470.0, 490.0, 490.0],
        [515.0, 515.0, 525.0, 525.0],
```

```
[505.0, 505.0, 520.0, 520.0],
[535.0, 535.0, 545.0, 545.0],
[585.0, 585.0, 595.0, 595.0],
[575.0, 575.0, 590.0, 590.0],
[565.0, 565.0, 582.5, 582.5],
[555.0, 555.0, 573.75, 573.75],
[540.0, 540.0, 564.375, 564.375],
[512.5, 512.5, 552.1875, 552.1875],
[480.0, 480.0, 532.34375, 532.34375],
[431.875, 431.875, 506.171875, 506.171875],
[605.0, 605.0, 615.0, 615.0],
[625.0, 625.0, 635.0, 635.0],
[610.0, 610.0, 630.0, 630.0],
[655.0, 655.0, 665.0, 665.0],
[645.0, 645.0, 660.0, 660.0],
[620.0, 620.0, 652.5, 652.5],
[675.0, 675.0, 685.0, 685.0],
[715.0, 715.0, 725.0, 725.0],
[705.0, 705.0, 720.0, 720.0],
[735.0, 735.0, 745.0, 745.0],
[712.5, 712.5, 740.0, 740.0],
[695.0, 695.0, 726.25, 726.25],
[680.0, 680.0, 710.625, 710.625],
[765.0, 765.0, 775.0, 775.0],
[755.0, 755.0, 770.0, 770.0],
[785.0, 785.0, 795.0, 795.0],
[762.5, 762.5, 790.0, 790.0],
[805.0, 805.0, 815.0, 815.0],
[835.0, 835.0, 845.0, 845.0],
[825.0, 825.0, 840.0, 840.0],
[810.0, 810.0, 832.5, 832.5],
[855.0, 855.0, 865.0, 865.0],
[885.0, 885.0, 895.0, 895.0],
[875.0, 875.0, 890.0, 890.0],
[860.0, 860.0, 882.5, 882.5],
[821.25, 821.25, 871.25, 871.25],
[776.25, 776.25, 846.25, 846.25],
[695.3125, 695.3125, 811.25, 811.25],
[636.25, 636.25, 753.28125, 753.28125],
[469.0234375, 469.0234375, 694.765625, 694.765625],
[915.0, 915.0, 925.0, 925.0],
[905.0, 905.0, 920.0, 920.0],
[945.0, 945.0, 955.0, 955.0],
[935.0, 935.0, 950.0, 950.0],
[912.5, 912.5, 942.5, 942.5],
[985.0, 985.0, 995.0, 995.0],
[975.0, 975.0, 990.0, 990.0],
```

```
[965.0, 965.0, 982.5, 982.5],
[927.5, 927.5, 973.75, 973.75],
[1025.0, 1025.0, 1035.0, 1035.0],
[1015.0, 1015.0, 1030.0, 1030.0],
[1005.0, 1005.0, 1022.5, 1022.5],
[1055.0, 1055.0, 1065.0, 1065.0],
[1045.0, 1045.0, 1060.0, 1060.0],
[1013.75, 1013.75, 1052.5, 1052.5],
[1085.0, 1085.0, 1095.0, 1095.0],
[1075.0, 1075.0, 1090.0, 1090.0],
[1135.0, 1135.0, 1145.0, 1145.0],
[1125.0, 1125.0, 1140.0, 1140.0],
[1115.0, 1115.0, 1132.5, 1132.5],
[1105.0, 1105.0, 1123.75, 1123.75],
[1082.5, 1082.5, 1114.375, 1114.375],
[1165.0, 1165.0, 1175.0, 1175.0],
[1155.0, 1155.0, 1170.0, 1170.0],
[1185.0, 1185.0, 1195.0, 1195.0],
[1205.0, 1205.0, 1215.0, 1215.0],
[1190.0, 1190.0, 1210.0, 1210.0],
[1225.0, 1225.0, 1235.0, 1235.0],
[1245.0, 1245.0, 1255.0, 1255.0],
[1230.0, 1230.0, 1250.0, 1250.0],
[1200.0, 1200.0, 1240.0, 1240.0],
[1162.5, 1162.5, 1220.0, 1220.0],
[1098.4375, 1098.4375, 1191.25, 1191.25],
[1033.125, 1033.125, 1144.84375, 1144.84375],
[950.625, 950.625, 1088.984375, 1088.984375],
[581.89453125, 581.89453125, 1019.8046875, 1019.8046875],
[275.9375, 275.9375, 800.849609375, 800.849609375],
[63.125, 63.125, 538.3935546875, 538.3935546875],
[1275.0, 1275.0, 1285.0, 1285.0],
[1265.0, 1265.0, 1280.0, 1280.0],
[1295.0, 1295.0, 1305.0, 1305.0],
[1315.0, 1315.0, 1325.0, 1325.0],
[1355.0, 1355.0, 1365.0, 1365.0],
[1345.0, 1345.0, 1360.0, 1360.0],
[1335.0, 1335.0, 1352.5, 1352.5],
[1320.0, 1320.0, 1343.75, 1343.75],
[1300.0, 1300.0, 1331.875, 1331.875],
[1375.0, 1375.0, 1385.0, 1385.0],
[1395.0, 1395.0, 1405.0, 1405.0],
[1380.0, 1380.0, 1400.0, 1400.0],
[1315.9375, 1315.9375, 1390.0, 1390.0],
[1415.0, 1415.0, 1425.0, 1425.0],
[1445.0, 1445.0, 1455.0, 1455.0],
[1435.0, 1435.0, 1450.0, 1450.0],
```

```
[1420.0, 1420.0, 1442.5, 1442.5],
[1352.96875, 1352.96875, 1431.25, 1431.25],
[1475.0, 1475.0, 1485.0, 1485.0],
[1465.0, 1465.0, 1480.0, 1480.0],
[1495.0, 1495.0, 1505.0, 1505.0],
[1515.0, 1515.0, 1525.0, 1525.0],
[1545.0, 1545.0, 1555.0, 1555.0],
[1535.0, 1535.0, 1550.0, 1550.0],
[1520.0, 1520.0, 1542.5, 1542.5],
[1500.0, 1500.0, 1531.25, 1531.25],
[1472.5, 1472.5, 1515.625, 1515.625],
[1565.0, 1565.0, 1575.0, 1575.0],
[1585.0, 1585.0, 1595.0, 1595.0],
[1570.0, 1570.0, 1590.0, 1590.0],
[1635.0, 1635.0, 1645.0, 1645.0],
[1625.0, 1625.0, 1640.0, 1640.0],
[1615.0, 1615.0, 1632.5, 1632.5],
[1605.0, 1605.0, 1623.75, 1623.75],
[1580.0, 1580.0, 1614.375, 1614.375],
[1494.0625, 1494.0625, 1597.1875, 1597.1875],
[1392.109375, 1392.109375, 1545.625, 1545.625],
[1272.5, 1272.5, 1468.8671875, 1468.8671875],
[1665.0, 1665.0, 1675.0, 1675.0],
[1655.0, 1655.0, 1670.0, 1670.0],
[1685.0, 1685.0, 1695.0, 1695.0],
[1735.0, 1735.0, 1745.0, 1745.0],
[1725.0, 1725.0, 1740.0, 1740.0],
[1715.0, 1715.0, 1732.5, 1732.5],
[1705.0, 1705.0, 1723.75, 1723.75],
[1690.0, 1690.0, 1714.375, 1714.375],
[1662.5, 1662.5, 1702.1875, 1702.1875],
[1755.0, 1755.0, 1765.0, 1765.0],
[1795.0, 1795.0, 1805.0, 1805.0],
[1785.0, 1785.0, 1800.0, 1800.0],
[1775.0, 1775.0, 1792.5, 1792.5],
[1760.0, 1760.0, 1783.75, 1783.75],
[1815.0, 1815.0, 1825.0, 1825.0],
[1835.0, 1835.0, 1845.0, 1845.0],
[1820.0, 1820.0, 1840.0, 1840.0],
[1855.0, 1855.0, 1865.0, 1865.0],
[1885.0, 1885.0, 1895.0, 1895.0],
[1875.0, 1875.0, 1890.0, 1890.0],
[1860.0, 1860.0, 1882.5, 1882.5],
[1915.0, 1915.0, 1925.0, 1925.0],
[1905.0, 1905.0, 1920.0, 1920.0],
[1935.0, 1935.0, 1945.0, 1945.0],
[1955.0, 1955.0, 1965.0, 1965.0],
```

```
 [1985.0, 1985.0, 1995.0, 1995.0],
 [1975.0, 1975.0, 1990.0, 1990.0],
 [1960.0, 1960.0, 1982.5, 1982.5],
 [1940.0, 1940.0, 1971.25, 1971.25],
 [1912.5, 1912.5, 1955.625, 1955.625],
 [1871.25, 1871.25, 1934.0625, 1934.0625],
 [1830.0, 1830.0, 1902.65625, 1902.65625],
 [1771.875, 1771.875, 1866.328125, 1866.328125],
 [1682.34375, 1682.34375, 1819.1015625, 1819.1015625],
 [1370.68359375, 1370.68359375, 1750.72265625, 1750.72265625],
 [300.75927734375, 300.75927734375, 1560.703125, 1560.703125]],
'dcoord': [[0.0, 4.123105625617661, 4.123105625617661, 0.0],
 [0.0, 15.198684153570664, 15.198684153570664, 4.123105625617661],
 [0.0, 1.7320508075688772, 1.7320508075688772, 0.0],
 [0.0, 3.0, 3.0, 0.0],
 [1.7320508075688772, 5.477225575051661, 5.477225575051661, 3.0],
 [0.0, 6.723094525588644, 6.723094525588644, 5.477225575051661],
 [0.0, 6.082762530298219, 6.082762530298219, 0.0],
 [0.0, 6.952217871538069, 6.952217871538069, 6.082762530298219],
 [0.0, 13.552367566837413, 13.552367566837413, 6.952217871538069],
 [6.723094525588644,
  19.726462767899708,
  19.726462767899708,
  13.552367566837413],
 [0.0, 7.280109889280518, 7.280109889280518, 0.0],
 [0.0, 8.18535277187245, 8.18535277187245, 7.280109889280518],
 [0.0, 12.16552506059644, 12.16552506059644, 8.18535277187245],
 [0.0, 7.810249675906654, 7.810249675906654, 0.0],
 [0.0, 7.937253933193772, 7.937253933193772, 0.0],
 [7.810249675906654, 16.15549442140351, 16.15549442140351, 7.937253933193772],
 [12.16552506059644,
  25.739075352467495,
  25.739075352467495,
  16.15549442140351],
 [19.726462767899708,
  43.262423127952054,
  43.262423127952054,
  25.739075352467495],
 [15.198684153570664,
  47.414443071333295,
  47.414443071333295,
  43.262423127952054],
 [0.0, 9.16515138991168, 9.16515138991168, 0.0],
 [0.0, 13.416407864998737, 13.416407864998737, 9.16515138991168],
 [0.0, 7.615773105863909, 7.615773105863909, 0.0],
 [0.0, 7.0710678118654755, 7.0710678118654755, 0.0],
 [0.0, 13.73559851869101, 13.73559851869101, 7.0710678118654755],
```

```
[7.615773105863909,
 15.853495934125489,
 15.853495934125489,
 13.73559851869101],
[13.416407864998737,
 29.244657631779518,
 29.244657631779518,
 15.853495934125489],
[0.0, 7.615773105863909, 7.615773105863909, 0.0],
[0.0, 19.235384061671347, 19.235384061671347, 7.615773105863909],
[0.0, 3.1622776601683795, 3.1622776601683795, 0.0],
[0.0, 9.763879010584539, 9.763879010584539, 3.1622776601683795],
[0.0, 19.485036994233976, 19.485036994233976, 9.763879010584539],
[0.0, 11.224972160321824, 11.224972160321824, 0.0],
[0.0, 5.196152422706632, 5.196152422706632, 0.0],
[0.0, 13.051181300301263, 13.051181300301263, 5.196152422706632],
[11.224972160321824,
 22.268063828421784,
 22.268063828421784,
 13.051181300301263],
[19.485036994233976,
 41.849200178205976,
 41.849200178205976,
 22.268063828421784],
[19.235384061671347,
 53.49870195725417,
 53.49870195725417,
 41.849200178205976],
[29.244657631779518, 82.8117342731894, 82.8117342731894, 53.49870195725417],
[0.0, 3.0, 3.0, 0.0],
[0.0, 5.322906474223771, 5.322906474223771, 3.0],
[0.0, 11.711817393840576, 11.711817393840576, 5.322906474223771],
[0.0, 12.68857754044952, 12.68857754044952, 0.0],
[11.711817393840576,
 22.85825890132492,
 22.85825890132492,
 12.68857754044952],
[0.0, 7.0, 7.0, 0.0],
[0.0, 7.280109889280518, 7.280109889280518, 0.0],
[7.0, 8.831760866327846, 8.831760866327846, 7.280109889280518],
[0.0, 3.0, 3.0, 0.0],
[0.0, 4.203173404306163, 4.203173404306163, 3.0],
[0.0, 5.916079783099616, 5.916079783099616, 0.0],
[0.0, 2.449489742783178, 2.449489742783178, 0.0],
[0.0, 2.9439202887759492, 2.9439202887759492, 2.449489742783178],
[0.0, 4.7784237289438165, 4.7784237289438165, 2.9439202887759492],
[0.0, 7.162401831787994, 7.162401831787994, 4.7784237289438165],
```

```
[5.916079783099616,
 11.144505372604023,
 11.144505372604023,
 7.162401831787994],
[4.203173404306163,
 16.447897535348805,
 16.447897535348805,
 11.144505372604023],
[8.831760866327846,
 28.47956660986459,
 28.47956660986459,
 16.447897535348805],
[22.85825890132492,
 35.728340091785476,
 35.728340091785476,
 28.47956660986459],
[0.0, 4.47213595499958, 4.47213595499958, 0.0],
[0.0, 6.164414002968976, 6.164414002968976, 0.0],
[4.47213595499958, 8.77496438739212, 8.77496438739212, 6.164414002968976],
[0.0, 8.12403840463596, 8.12403840463596, 0.0],
[0.0, 19.44222209522358, 19.44222209522358, 8.12403840463596],
[8.77496438739212, 24.6083609484953, 24.6083609484953, 19.44222209522358],
[0.0, 5.196152422706632, 5.196152422706632, 0.0],
[0.0, 2.23606797749979, 2.23606797749979, 0.0],
[0.0, 3.696845502136472, 3.696845502136472, 2.23606797749979],
[0.0, 5.477225575051661, 5.477225575051661, 0.0],
[3.696845502136472,
 10.960535266734619,
 10.960535266734619,
 5.477225575051661],
[0.0, 14.658330964563008, 14.658330964563008, 10.960535266734619],
[5.196152422706632,
 19.12677007059303,
 19.12677007059303,
 14.658330964563008],
[0.0, 3.4641016151377544, 3.4641016151377544, 0.0],
[0.0, 5.773502691896257, 5.773502691896257, 3.4641016151377544],
[0.0, 6.557438524302, 6.557438524302, 0.0],
[5.773502691896257, 15.711991174471384, 15.711991174471384, 6.557438524302],
[0.0, 4.47213595499958, 4.47213595499958, 0.0],
[0.0, 4.242640687119285, 4.242640687119285, 0.0],
[0.0, 7.2571803523590805, 7.2571803523590805, 4.242640687119285],
[4.47213595499958, 9.825137827701623, 9.825137827701623, 7.2571803523590805],
[0.0, 3.1622776601683795, 3.1622776601683795, 0.0],
[0.0, 4.242640687119285, 4.242640687119285, 0.0],
[0.0, 6.48074069840786, 6.48074069840786, 4.242640687119285],
[3.1622776601683795,
```

```
 11.224972160321824,
 11.224972160321824,
 6.48074069840786],
[9.825137827701623,
 22.441033844277317,
 22.441033844277317,
 11.224972160321824],
[15.711991174471384,
 35.35533905932738,
 35.35533905932738,
 22.441033844277317],
[19.12677007059303,
 51.631006861819166,
 51.631006861819166,
 35.35533905932738],
[24.6083609484953,
 62.757477922528786,
 62.757477922528786,
 51.631006861819166],
[35.728340091785476,
 79.43617983429633,
 79.43617983429633,
 62.757477922528786],
[0.0, 2.0, 2.0, 0.0],
[0.0, 6.733003292241384, 6.733003292241384, 2.0],
[0.0, 5.830951894845301, 5.830951894845301, 0.0],
[0.0, 8.286535263104037, 8.286535263104037, 5.830951894845301],
[6.733003292241384,
 15.695009822658069,
 15.695009822658069,
 8.286535263104037],
[0.0, 3.872983346207417, 3.872983346207417, 0.0],
[0.0, 11.0, 11.0, 3.872983346207417],
[0.0, 16.401219466856727, 16.401219466856727, 11.0],
[15.695009822658069,
 34.60732099811637,
 34.60732099811637,
 16.401219466856727],
[0.0, 3.605551275463989, 3.605551275463989, 0.0],
[0.0, 6.855654600401044, 6.855654600401044, 3.605551275463989],
[0.0, 9.460443964212251, 9.460443964212251, 6.855654600401044],
[0.0, 5.744562646538029, 5.744562646538029, 0.0],
[0.0, 12.288205727444506, 12.288205727444506, 5.744562646538029],
[9.460443964212251,
 30.922714720966624,
 30.922714720966624,
 12.288205727444506],
```

```
[0.0, 2.23606797749979, 2.23606797749979, 0.0],
[0.0, 5.916079783099617, 5.916079783099617, 2.23606797749979],
[0.0, 2.23606797749979, 2.23606797749979, 0.0],
[0.0, 3.8729833462074166, 3.8729833462074166, 2.23606797749979],
[0.0, 5.0, 5.0, 3.8729833462074166],
[0.0, 7.362064927722386, 7.362064927722386, 5.0],
[5.916079783099617,
 16.727223320085134,
 16.727223320085134,
 7.362064927722386],
[0.0, 1.0, 1.0, 0.0],
[0.0, 5.196152422706632, 5.196152422706632, 1.0],
[0.0, 3.7416573867739413, 3.7416573867739413, 0.0],
[0.0, 3.7416573867739413, 3.7416573867739413, 0.0],
[3.7416573867739413,
 6.4031242374328485,
 6.4031242374328485,
 3.7416573867739413],
[0.0, 3.7416573867739413, 3.7416573867739413, 0.0],
[0.0, 5.196152422706632, 5.196152422706632, 0.0],
[3.7416573867739413,
 9.137833441248532,
 9.137833441248532,
 5.196152422706632],
[6.4031242374328485, 16.3783393541592, 16.3783393541592, 9.137833441248532],
[5.196152422706632,
 27.503305586453028,
 27.503305586453028,
 16.3783393541592],
[16.727223320085134,
 35.37583583133478,
 35.37583583133478,
 27.503305586453028],
[30.922714720966624,
 60.367148324869596,
 60.367148324869596,
 35.37583583133478],
[34.60732099811637, 82.4265018372623, 82.4265018372623, 60.367148324869596],
[79.43617983429633,
 186.06538069042523,
 186.06538069042523,
 82.4265018372623],
[82.8117342731894,
 241.40194206041446,
 241.40194206041446,
 186.06538069042523],
[47.414443071333295,
```

```
 278.12797950929695,
 278.12797950929695,
 241.40194206041446],
[0.0, 8.426149773176359, 8.426149773176359, 0.0],
[0.0, 18.284784202536635, 18.284784202536635, 8.426149773176359],
[0.0, 1.0, 1.0, 0.0],
[0.0, 2.449489742783178, 2.449489742783178, 0.0],
[0.0, 2.6457513110645907, 2.6457513110645907, 0.0],
[0.0, 4.43471156521669, 4.43471156521669, 2.6457513110645907],
[0.0, 7.438637868140466, 7.438637868140466, 4.43471156521669],
[2.449489742783178, 8.831760866327846, 8.831760866327846, 7.438637868140466],
[1.0, 13.06713434537198, 13.06713434537198, 8.831760866327846],
[0.0, 5.196152422706632, 5.196152422706632, 0.0],
[0.0, 7.3484692283495345, 7.3484692283495345, 0.0],
[5.196152422706632,
 13.435028842544401,
 13.435028842544401,
 7.3484692283495345],
[13.06713434537198,
 18.45038391651151,
 18.45038391651151,
 13.435028842544401],
[0.0, 4.69041575982343, 4.69041575982343, 0.0],
[0.0, 6.164414002968976, 6.164414002968976, 0.0],
[0.0, 11.633285577743433, 11.633285577743433, 6.164414002968976],
[4.69041575982343, 22.80058478782215, 22.80058478782215, 11.633285577743433],
[18.45038391651151, 51.12341852388812, 51.12341852388812, 22.80058478782215],
[0.0, 4.58257569495584, 4.58257569495584, 0.0],
[0.0, 10.908712114635714, 10.908712114635714, 4.58257569495584],
[0.0, 6.48074069840786, 6.48074069840786, 0.0],
[0.0, 5.477225575051661, 5.477225575051661, 0.0],
[0.0, 4.242640687119285, 4.242640687119285, 0.0],
[0.0, 5.8309518948453, 5.8309518948453, 4.242640687119285],
[5.477225575051661, 7.77174369109018, 7.77174369109018, 5.8309518948453],
[6.48074069840786, 10.96227296555913, 10.96227296555913, 7.77174369109018],
[10.908712114635714,
 20.214563349935894,
 20.214563349935894,
 10.96227296555913],
[0.0, 4.47213595499958, 4.47213595499958, 0.0],
[0.0, 8.18535277187245, 8.18535277187245, 0.0],
[4.47213595499958, 10.024968827881711, 10.024968827881711, 8.18535277187245],
[0.0, 3.7416573867739413, 3.7416573867739413, 0.0],
[0.0, 9.273618495495702, 9.273618495495702, 3.7416573867739413],
[0.0, 13.435028842544401, 13.435028842544401, 9.273618495495702],
[0.0, 19.27433526739638, 19.27433526739638, 13.435028842544401],
[10.024968827881711,
```

```
 28.62885102673719,
 28.62885102673719,
 19.27433526739638],
[20.214563349935894, 62.8875725690357, 62.8875725690357, 28.62885102673719],
[51.12341852388812, 75.82351119778876, 75.82351119778876, 62.8875725690357],
[18.284784202536635,
 105.23277007394557,
 105.23277007394557,
 75.82351119778876],
[0.0, 13.564659966250536, 13.564659966250536, 0.0],
[0.0, 23.35237318418266, 23.35237318418266, 13.564659966250536],
[0.0, 8.306623862918075, 8.306623862918075, 0.0],
[0.0, 7.416198487095663, 7.416198487095663, 0.0],
[0.0, 13.940349110884322, 13.940349110884322, 7.416198487095663],
[0.0, 18.12916618784953, 18.12916618784953, 13.940349110884322],
[0.0, 25.05194603219478, 25.05194603219478, 18.12916618784953],
[8.306623862918075, 32.33264604080525, 32.33264604080525, 25.05194603219478],
[23.35237318418266, 52.75477861451668, 52.75477861451668, 32.33264604080525],
[0.0, 3.0, 3.0, 0.0],
[0.0, 5.196152422706632, 5.196152422706632, 0.0],
[0.0, 9.146948489341495, 9.146948489341495, 5.196152422706632],
[0.0, 9.60902353693305, 9.60902353693305, 9.146948489341495],
[3.0, 26.814175355583846, 26.814175355583846, 9.60902353693305],
[0.0, 7.874007874011811, 7.874007874011811, 0.0],
[0.0, 12.409673645990857, 12.409673645990857, 0.0],
[7.874007874011811,
 17.05872210923198,
 17.05872210923198,
 12.409673645990857],
[0.0, 5.0, 5.0, 0.0],
[0.0, 9.327379053088816, 9.327379053088816, 0.0],
[0.0, 11.532562594670797, 11.532562594670797, 9.327379053088816],
[5.0, 18.606450494385005, 18.606450494385005, 11.532562594670797],
[0.0, 5.196152422706632, 5.196152422706632, 0.0],
[0.0, 8.306623862918075, 8.306623862918075, 5.196152422706632],
[0.0, 7.54983443527075, 7.54983443527075, 0.0],
[0.0, 4.242640687119285, 4.242640687119285, 0.0],
[0.0, 3.3166247903554, 3.3166247903554, 0.0],
[0.0, 5.0, 5.0, 3.3166247903554],
[4.242640687119285, 17.67484087622856, 17.67484087622856, 5.0],
[7.54983443527075,
 24.460463726476767,
 24.460463726476767,
 17.67484087622856],
[8.306623862918075,
 34.34655316455661,
 34.34655316455661,
```

```
  24.460463726476767],
 [18.606450494385005, 40.3773864103824, 40.3773864103824, 34.34655316455661],
 [17.05872210923198, 47.4326484644372, 47.4326484644372, 40.3773864103824],
 [26.814175355583846, 76.24651156817464, 76.24651156817464, 47.4326484644372],
 [52.75477861451668, 116.3878736933412, 116.3878736933412, 76.24651156817464],
 [105.23277007394557,
  397.93342818367654,
  397.93342818367654,
  116.3878736933412],
 [278.12797950929695,
  411.9934269529356,
  411.9934269529356,
  397.93342818367654]],
'ivl': ['7',
 '11',
 '19',
 '1',
 '3',
 '5',
 '13',
 '15',
 '17',
 '9',
 '21',
 '23',
 '37',
 '31',
 '35',
 '39',
 '33',
 '41',
 '25',
 '29',
 '8',
 '10',
 '12',
 '30',
 '32',
 '22',
 '24',
 '34',
 '2',
 '6',
 '14',
 '0',
 '4',
 '16',
```

```
'20',
'36',
'38',
'18',
'26',
'28',
'86',
'104',
'117',
'119',
'88',
'122',
'66',
'76',
'77',
'81',
'79',
'83',
'85',
'107',
'118',
'98',
'89',
'101',
'92',
'96',
'54',
'59',
'55',
'71',
'40',
'42',
'44',
'90',
'102',
'82',
'110',
'106',
'109',
'108',
'116',
'46',
'50',
'56',
'53',
'63',
'74',
```

```
'80',
'73',
'64',
'72',
'60',
'70',
'57',
'62',
'67',
'47',
'48',
'49',
'69',
'52',
'58',
'27',
'45',
'43',
'51',
'94',
'93',
'112',
'121',
'142',
'124',
'132',
'91',
'100',
'105',
'111',
'99',
'113',
'114',
'115',
'61',
'65',
'68',
'75',
'78',
'84',
'87',
'95',
'97',
'103',
'120',
'199',
'195',
```

```
'197',
'129',
'131',
'147',
'159',
'125',
'139',
'133',
'137',
'157',
'161',
'153',
'165',
'187',
'191',
'169',
'171',
'177',
'151',
'123',
'127',
'145',
'163',
'135',
'155',
'141',
'143',
'149',
'167',
'173',
'175',
'179',
'193',
'185',
'189',
'181',
'183',
'198',
'194',
'196',
'180',
'184',
'192',
'186',
'190',
'182',
'188',
```

```
 '156',
 '158',
 '162',
 '144',
 '134',
 '138',
 '126',
 '146',
 '160',
 '164',
 '170',
 '172',
 '148',
 '166',
 '168',
 '174',
 '176',
 '178',
 '128',
 '140',
 '130',
 '136',
 '154',
 '150',
 '152'],
'leaves': [7,
 11,
 19,
 1,
 3,
 5,
 13,
 15,
 17,
 9,
 21,
 23,
 37,
 31,
 35,
 39,
 33,
 41,
 25,
 29,
 8,
 10,
```

```
12,
30,
32,
22,
24,
34,
2,
6,
14,
0,
4,
16,
20,
36,
38,
18,
26,
28,
86,
104,
117,
119,
88,
122,
66,
76,
77,
81,
79,
83,
85,
107,
118,
98,
89,
101,
92,
96,
54,
59,
55,
71,
40,
42,
44,
90,
102,
```

```
82,
110,
106,
109,
108,
116,
46,
50,
56,
53,
63,
74,
80,
73,
64,
72,
60,
70,
57,
62,
67,
47,
48,
49,
69,
52,
58,
27,
45,
43,
51,
94,
93,
112,
121,
142,
124,
132,
91,
100,
105,
111,
99,
113,
114,
115,
61,
```

```
65,
68,
75,
78,
84,
87,
95,
97,
103,
120,
199,
195,
197,
129,
131,
147,
159,
125,
139,
133,
137,
157,
161,
153,
165,
187,
191,
169,
171,
177,
151,
123,
127,
145,
163,
135,
155,
141,
143,
149,
167,
173,
175,
179,
193,
185,
189,
```

```
 181,
 183,
 198,
 194,
 196,
 180,
 184,
 192,
 186,
 190,
 182,
 188,
 156,
 158,
 162,
 144,
 134,
 138,
 126,
 146,
 160,
 164,
 170,
 172,
 148,
 166,
 168,
 174,
 176,
 178,
 128,
 140,
 130,
 136,
 154,
 150,
 152],
'color_list': ['C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
```

```
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
```

```
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
```

```
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
```

```
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C0',
```

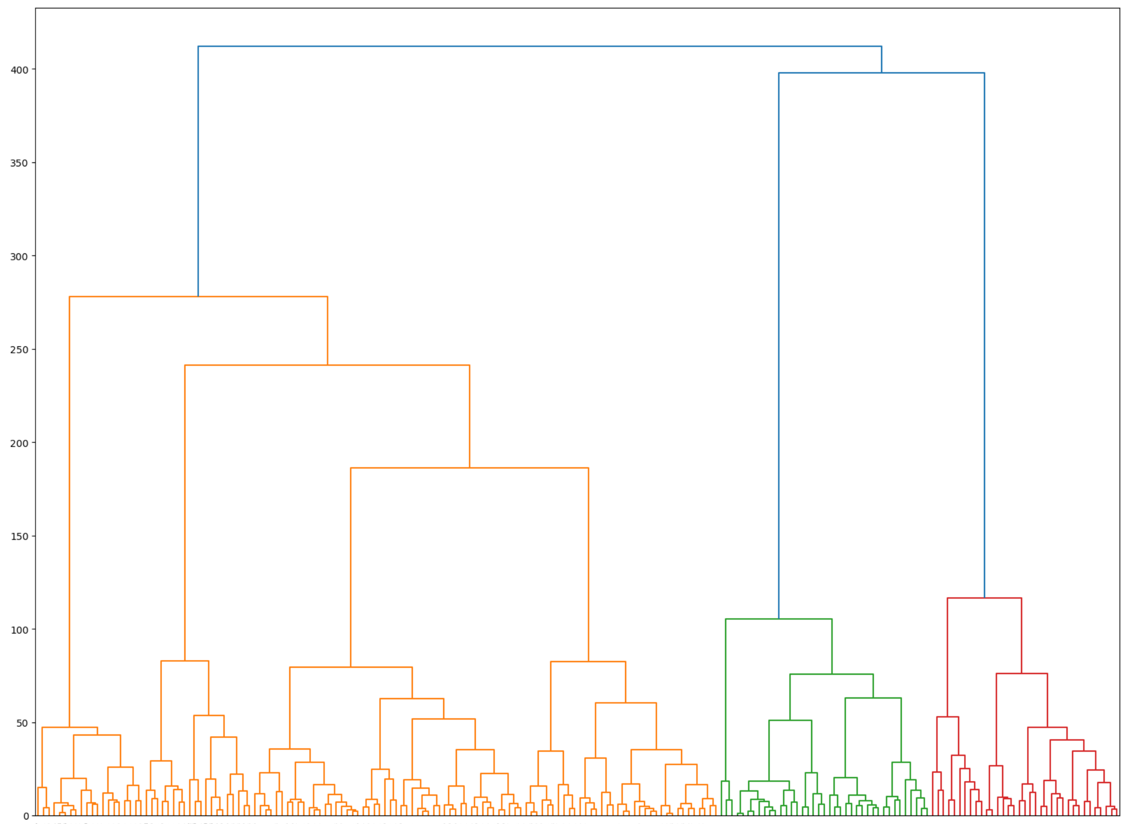
```
 'C0'],
'leaves_color_list': ['C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
 'C1',
```

```
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
```

```
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C1',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
```

```
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C2',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
```

```
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3',
'C3']}
```



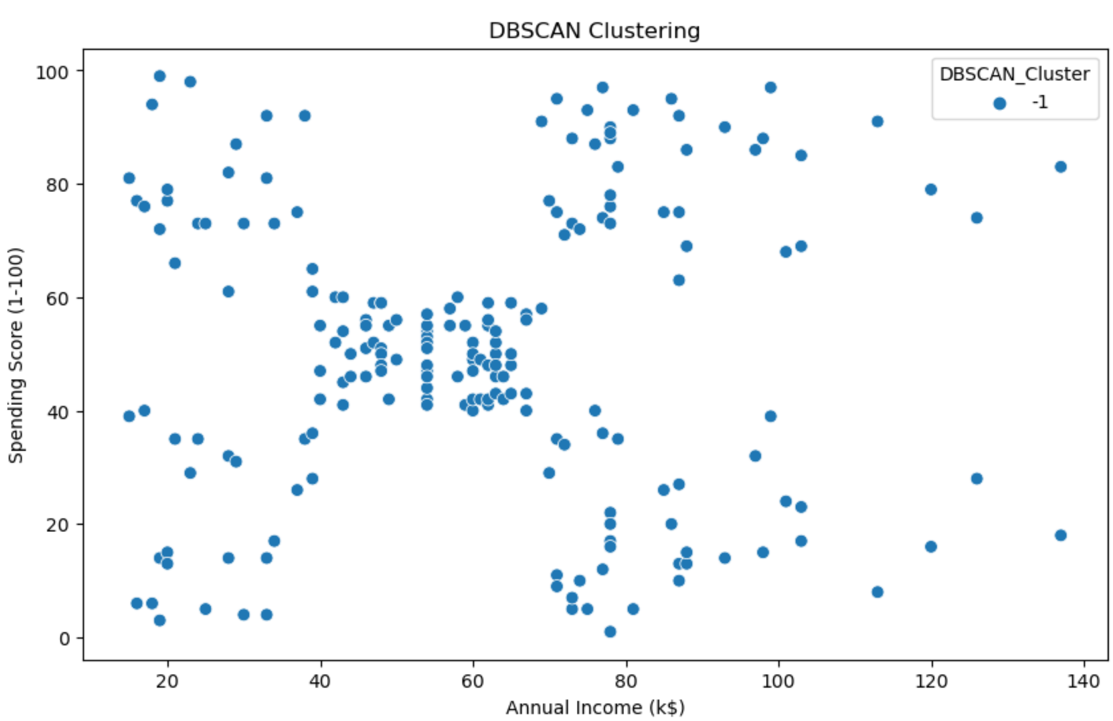
[54]:
```python
# DBSCAN
from sklearn.cluster import DBSCAN
```

[66]:
```python
# DBSCAN clustering
dbscan_model = DBSCAN(eps=0.3, min_samples=5)
dbscan_model.fit(df)
df['DBSCAN_Cluster'] = dbscan_model.labels_
```

[67]:
```
dbscan_labels = df['DBSCAN_Cluster']
dbscan_labels
```

[67]:
```
0      -1
1      -1
2      -1
3      -1
4      -1
       ..
195    -1
196    -1
197    -1
198    -1
199    -1
Name: DBSCAN_Cluster, Length: 200, dtype: int64
```

[68]:
```
# Visualize DBSCAN clustering
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Annual Income (k$)',
                y='Spending Score (1-100)',
                hue='DBSCAN_Cluster',
                data=df,
                palette='tab10',
                legend='full',
                s=50)
plt.title('DBSCAN Clustering')
plt.show()
```



```
[ ]:
```