```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import scipy.cluster.hierarchy as sch
from sklearn.cluster import AgglomerativeClustering
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files Mall_Customers.csv

Mall_Customers.csv(text/csv) - 4286 bytes, last modified: 10/10/2025 - 100% done

```
Saving Mall_Customers.csv to Mall_Customers (1).csv
```

```
import pandas as pd
df = pd.read_csv("Mall_Customers.csv")  # after upload
```

```
df
```

	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 × 5 columns

```
x=df.iloc[:,3:]
```

```
x
```

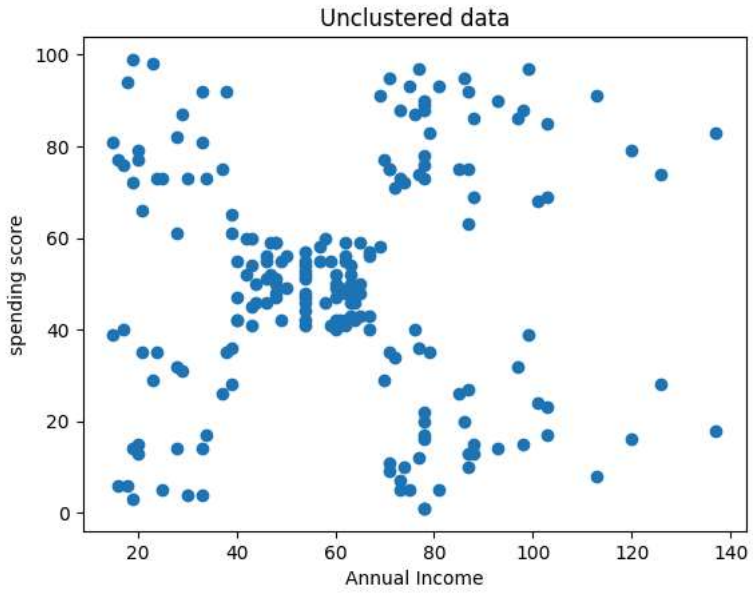
	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40
...	...	...
195	120	79
196	126	28
197	126	74
198	137	18
199	137	83

200 rows × 2 columns

```
plt.title('Unclustered data')
plt.xlabel("Annual Income")
```

```
plt.ylabel('spending score')
plt.scatter(x['Annual Income (k$)'],x['Spending Score (1-100)'])
```

```
<matplotlib.collections.PathCollection at 0x7b6347997bf0>
```




```
km=KMeans(n_clusters=6)
```

```
km.fit_predict(x)
```

```
array([2, 4, 5, 4, 2, 4, 5, 4, 5, 4, 5, 4, 5, 4, 5, 4, 2, 4, 2, 4, 2, 4,
       5, 4, 5, 4, 2, 4, 2, 4, 5, 4, 5, 4, 5, 4, 5, 4, 2, 4, 2, 4, 2, 3,
       2, 4, 3, 2, 2, 2, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 0, 1, 3, 1, 0, 1, 0, 1,
       3, 1, 0, 1, 0, 1, 0, 1, 0, 1, 3, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1], dtype=int32)
```

```
x.shape
```

```
(200, 2)
```

```
km.inertia_
```

```
40825.16946386947
```

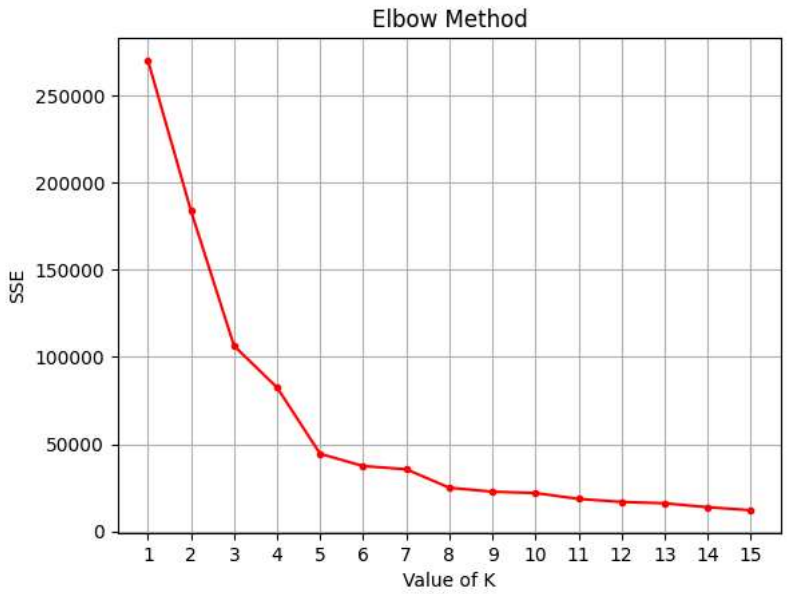
```
sse=[]
for k in range(1,16):
  km=KMeans(n_clusters=k)
  km.fit_predict(x)
  sse.append(km.inertia_)
```

```
sse
```

```
[269981.28000000014,
 183653.3289473683,
 106348.37306211119,
 82443.04256024676,
 44448.45544793369,
 37455.98455516028,
 35620.04757113041,
 25022.485004530332,
 22755.68634977502,
 21979.637120232714,
 18619.981243611463,
 16890.475555479014,
 16110.516548877475,
 13859.041296241297,
 12138.128862803862]
```

```
plt.title('Elbow Method')
plt.xlabel('Value of K')
plt.ylabel('SSE')
plt.grid()
plt.xticks (range(1,16))
plt.plot(range(1,16), sse, marker=".",color='red')
```

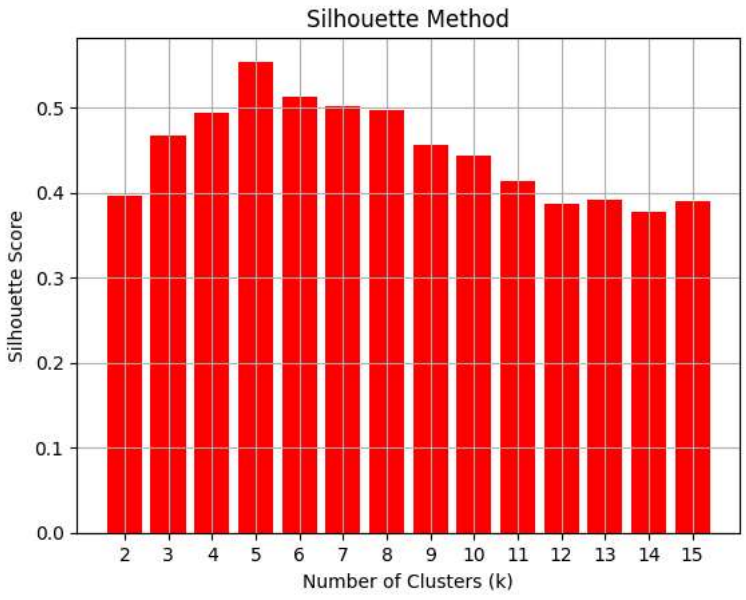
```
[<matplotlib.lines.Line2D at 0x7b633f251910>]
```




```
silh = []
for k in range(2, 16):
    km = KMeans(n_clusters=k, random_state=42)
    labels = km.fit_predict(x)
    score = silhouette_score(x, labels)
    silh.append(score)
```

```
plt.title("Silhouette Method")
plt.xlabel("Number of Clusters (k)")
plt.ylabel("Silhouette Score")
plt.grid()
plt.xticks(range(2,16))
plt.bar(range(2, 16), silh, color="red")
```

```
<BarContainer object of 14 artists>
```




```
km = KMeans(n_clusters=5, random_state=5)
```

```
labels = km.fit_predict(x)
```

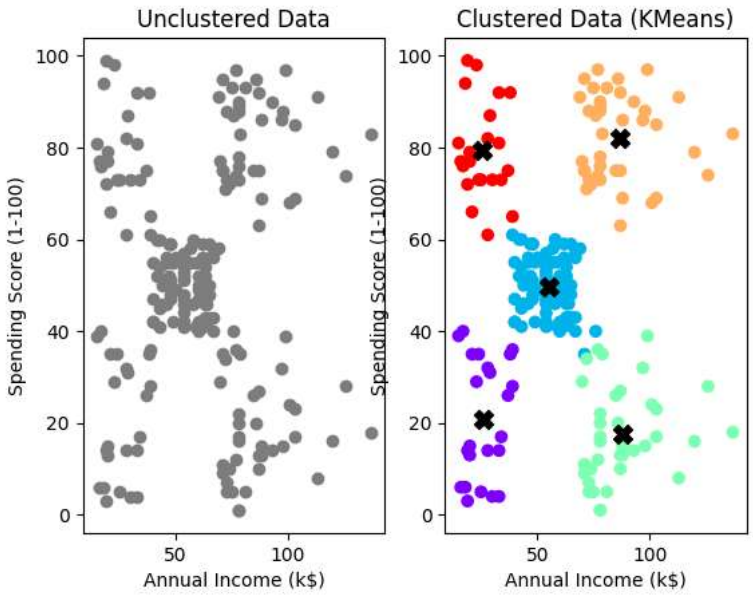
```
labels
```

```
array([0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4,
       0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 1,
       0, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 2, 3, 1, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 1, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3], dtype=int32)
```

```
cent = km.cluster_centers_
```

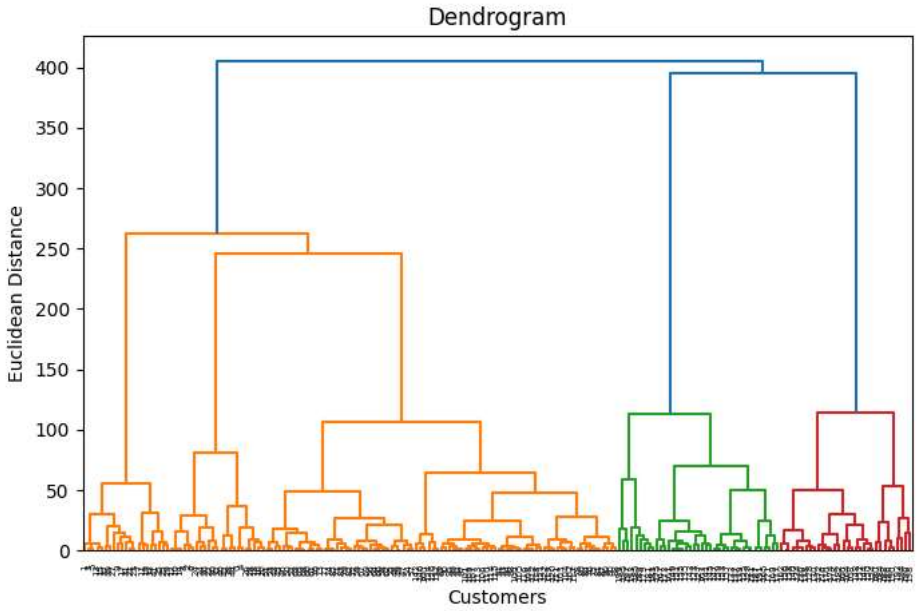
```
plt.subplot(1, 2, 1)
plt.title("Unclustered Data")
plt.xlabel("Annual Income (k$)")
plt.ylabel("Spending Score (1-100)")
plt.scatter(x["Annual Income (k$)"], x["Spending Score (1-100)"], color="gray")

plt.subplot(1, 2, 2)
plt.title("Clustered Data (KMeans)")
plt.xlabel("Annual Income (k$)")
plt.ylabel("Spending Score (1-100)")
plt.scatter(x["Annual Income (k$)"], x["Spending Score (1-100)"], c=labels, cmap="rainbow")
plt.scatter(cent[:, 0], cent[:, 1], s=100, color="black", marker="X")
```

```
<matplotlib.collections.PathCollection at 0x7b6347990740>
```




```
plt.figure(figsize=(8, 5))
dendrogram = sch.dendrogram(sch.linkage(x, method="ward"))
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Euclidean Distance")
```

```
Text(0, 0.5, 'Euclidean Distance')
```



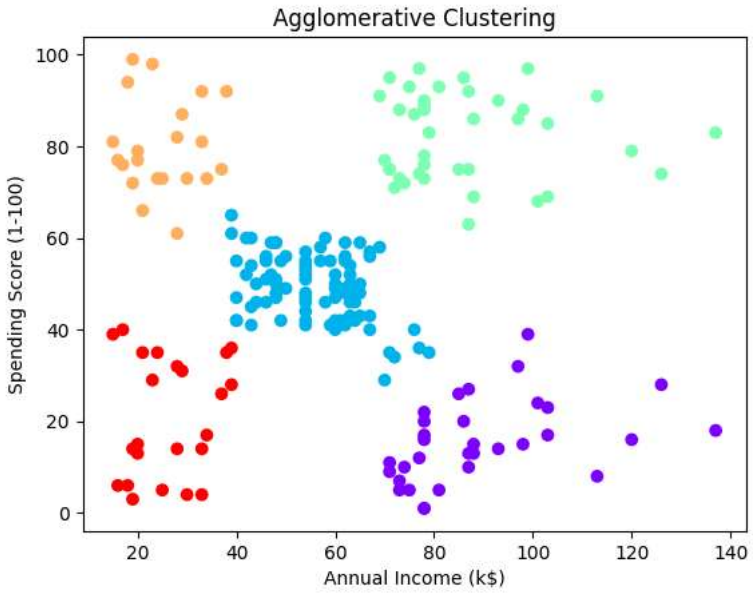

```
agl = AgglomerativeClustering(n_clusters=5)
alabel = agl.fit_predict(x)
```

```
alabel
```

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

```
plt.title("Agglomerative Clustering")
plt.xlabel("Annual Income (k$)")
plt.ylabel("Spending Score (1-100)")
plt.scatter(x["Annual Income (k$)"], x["Spending Score (1-100)"], c=alabel, cmap="rainbow")
```

```
<matplotlib.collections.PathCollection at 0x7b633e4115e0>
```




```
Start coding or generate with AI.
```