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

```
df=pd.read_csv("/content/archive (4).zip")
```

```
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:]  #iloc-indexlocation
```

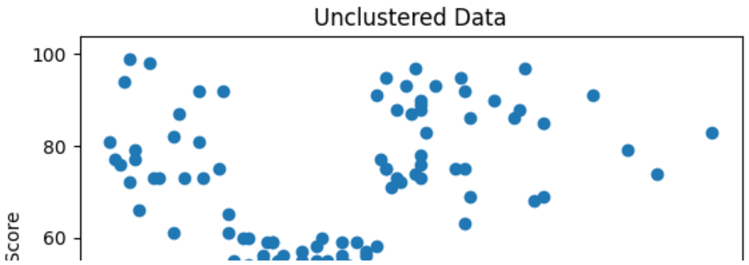
```
X
```

	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 0x7c2e9bea6980>
```



```
from sklearn.cluster import KMeans,AgglomerativeClustering
#K-Means is a partitioning method that aims to divide a dataset into K clusters
#Agglomerative clustering is a hierarchical clustering method (means ploting a graph)
```

```
km=KMeans(n_clusters=3)
```

```
X.shape
```

```
(200, 2)
```

```
km.fit_predict(X)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fro
  warnings.warn(
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 2, 1, 2, 1, 2, 1, 2,
       1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
       1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
       1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2,
       1, 2], dtype=int32)
```

```
ssee=km.inertia_
print("The sum Squared is:",ssee)
#sum of squared distances of data points to their nearest cluster center
```

```
The sum Squared is: 106348.37306211122
```

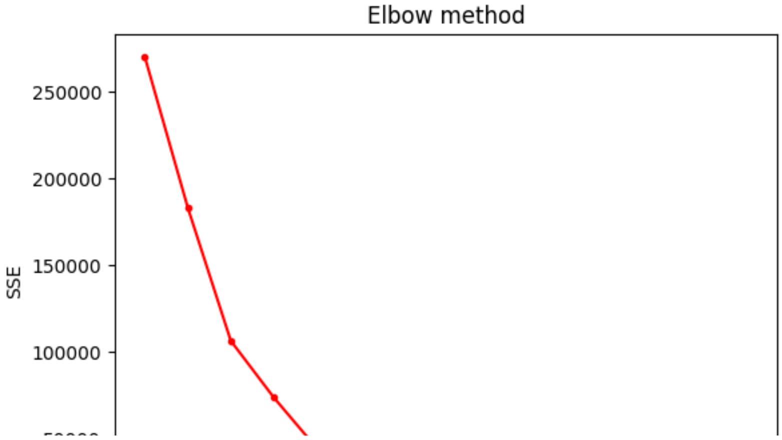
```
sse=[]
for k in range(1,16):
  km=KMeans(n_clusters=k)
  km.fit_predict(X)
  sse.append(km.inertia_)
```

```
sse
```

```
[269981.28,
 183069.17582751298,
 106348.37306211122,
 73679.78903948836,
 44448.4554479337,
 37265.86520484346,
 30273.394312070042,
 25018.781613414067,
 22847.052967563824,
 19664.68519600554,
 17636.649972700317,
 16506.039061460364,
 14525.757221847918,
 12744.657467532466,
 11728.187321012323]
```

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

```
[<matplotlib.lines.Line2D at 0x7c2e911e0ee0>]
```

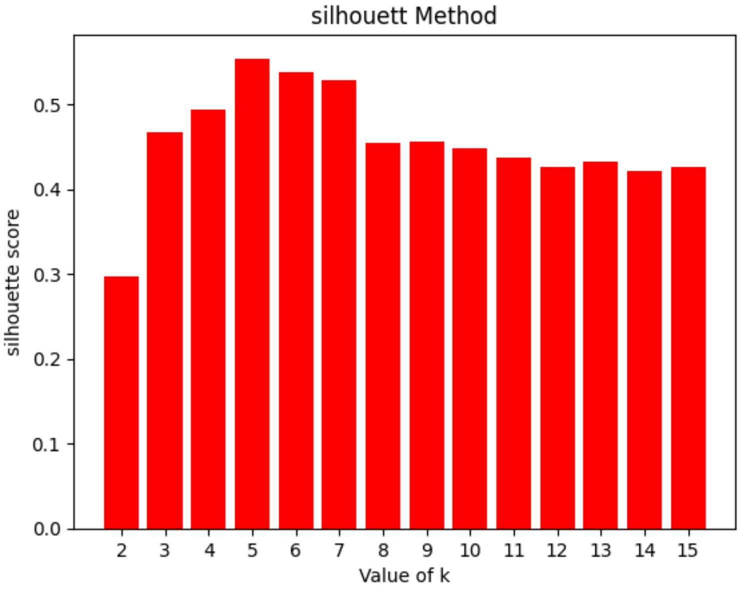


```
from sklearn.metrics import silhouette_score
```

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


plt.title("silhouett Method")
plt.xlabel("Value of k")
plt.ylabel("silhouette score")
plt.xticks(range(2,16))
plt.bar(range(2,16),silh,color="red")
```

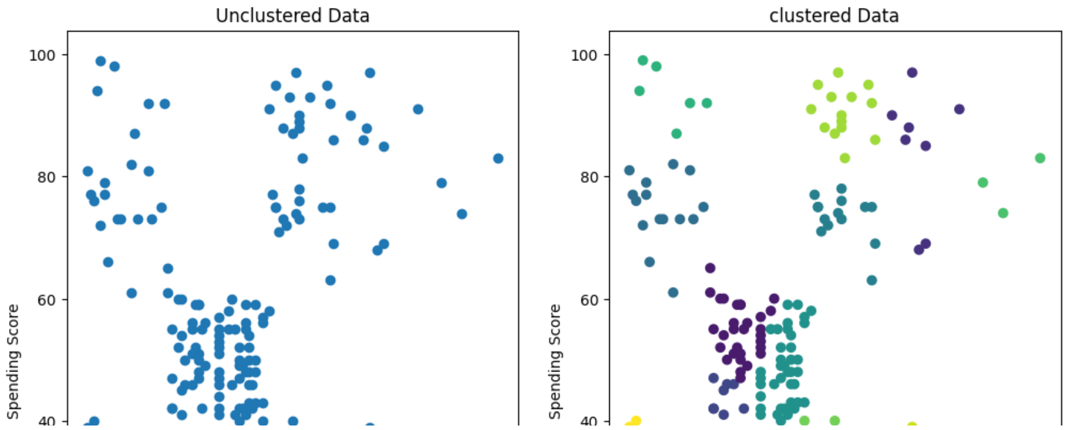
```
<BarContainer object of 14 artists>
```



```
plt.figure(figsize=(12,8))
plt.subplot(1,2,1)
plt.title('Unclustered Data')
plt.xlabel('Annual Income ')
plt.ylabel('Spending Score')
plt.scatter(X['Annual Income (k$)'], X['Spending Score (1-100)'])

plt.subplot(1,2,2)
plt.title('clustered Data')
plt.xlabel('Annual Income ')
plt.ylabel('Spending Score')
plt.scatter(X['Annual Income (k$)'], X['Spending Score (1-100)'],c=labels)
```

```
<matplotlib.collections.PathCollection at 0x7c2e8e0dead0>
```



```
km.inertia_
```

```
11932.10892857143
```

```
km.labels_
```

```
array([14,  5,  4,  5, 14,  5,  4,  9,  4,  5,  4,  9,  4,  5,  4,  5, 14,
        5, 14,  9, 14,  5,  4,  5,  4,  5, 14,  5, 14,  9,  4,  5,  4,  9,
        4,  5,  4,  5,  3,  5,  3,  9,  3,  1,  3,  1,  1,  3,  3,  3,  1,
        1,  1,  1,  3,  3,  1,  3,  1,  3,  1,  1,  1,  1,  1,  1,  1,  1,
        1,  1,  1,  3,  1,  1,  7,  1,  1,  7,  1,  7,  1,  1,  7,  7,  1,
        7,  1,  7,  1,  7,  7,  7,  7,  7,  7,  7,  7,  7,  7,  7,  7,  7,
        7,  7,  7,  7,  7,  7,  7,  7,  7,  7,  7,  7,  7,  7,  7,  7,  7,
        7,  7, 11,  7, 12, 11,  6, 11, 12,  0,  6,  0,  6, 11,  6,  0, 12,
        0,  6,  0,  6,  0, 12, 11, 12,  0, 12, 11,  6,  0, 12,  0, 12,  0,
        6,  0, 12,  0,  6,  0,  6, 11, 12,  0, 12, 13,  6, 13, 12, 13,  6,
        0,  6,  0, 12, 13, 12, 13,  6, 13,  2, 13,  2, 13,  2, 13,  2, 13,
        2, 13,  2, 13,  2,  8,  2,  8, 10,  8, 10,  8, 10], dtype=int32)
```

```
agl=AgglomerativeClustering(n_clusters=5)
```

```
alabels=km.fit_predict(X)
```

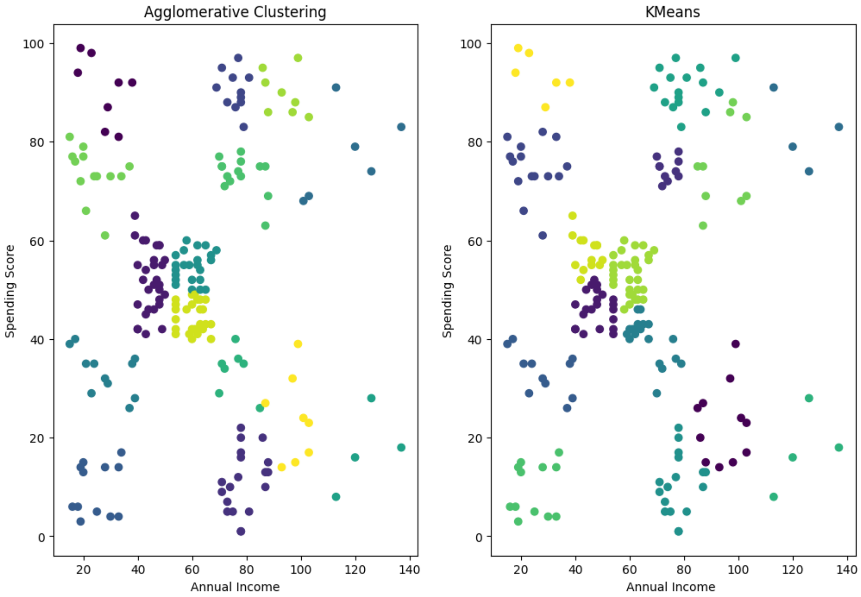
```
alabels
```

```
array([ 6, 11,  4, 11,  6, 11,  4,  0,  4, 11,  4,  0,  4, 11,  4, 11,  6,
       11,  6,  0,  6, 11,  4, 11,  4,  0,  6, 11,  6,  0,  4, 11,  4,  0,
        4,  0,  4, 11,  6, 11,  6,  0,  6,  1,  6,  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, 13,  7,  7, 13,  7, 13,  7,  7, 13, 13,  7,
       13,  7,  7,  7, 13,  7, 13, 13, 13, 13,  7, 13,  7, 13, 13, 13, 13,
        7,  7,  7, 13,  7, 13, 13, 13,  7,  7, 13, 13, 13,  7, 13,  7, 13,
        7,  7, 13,  7,  3,  9, 10,  9,  3,  2, 10,  2, 10,  9, 10,  2,  3,
        2, 10,  2, 10,  2,  3,  9,  3,  2,  3,  9, 10,  2,  3,  2,  3,  2,
       10,  2,  3,  2, 10,  2, 10,  9,  3,  2,  3,  9, 10,  2, 12, 14, 10,
        2, 10,  2, 12,  2, 12,  2, 10, 14, 12, 14, 12, 14, 12, 14, 12, 14,
        5, 14, 12, 14,  5,  8,  5,  8,  5,  8,  5,  8,  5], dtype=int32)
```

```
plt.figure(figsize=(12, 8))
plt.subplot(1, 2, 1)
plt.title('Agglomerative Clustering')
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.scatter(X['Annual Income (k$)'], X['Spending Score (1-100)'], c=alabels)

plt.subplot(1,2,2)
plt.title('KMeans')
plt.xlabel('Annual Income ')
plt.ylabel('Spending Score')
plt.scatter(X['Annual Income (k$)'], X['Spending Score (1-100)'],c=labels)
```

```
<matplotlib.collections.PathCollection at 0x7c2e8e6e1f60>
```


Agglomerative Clustering
KMeans
Spending Score
Annual Income
0
20
40
60
80
100
20
40
60
80
100
120
140