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

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

```
     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 x 5 columns]
```

Checking NULL values

```
dataset.isna().sum()
```

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

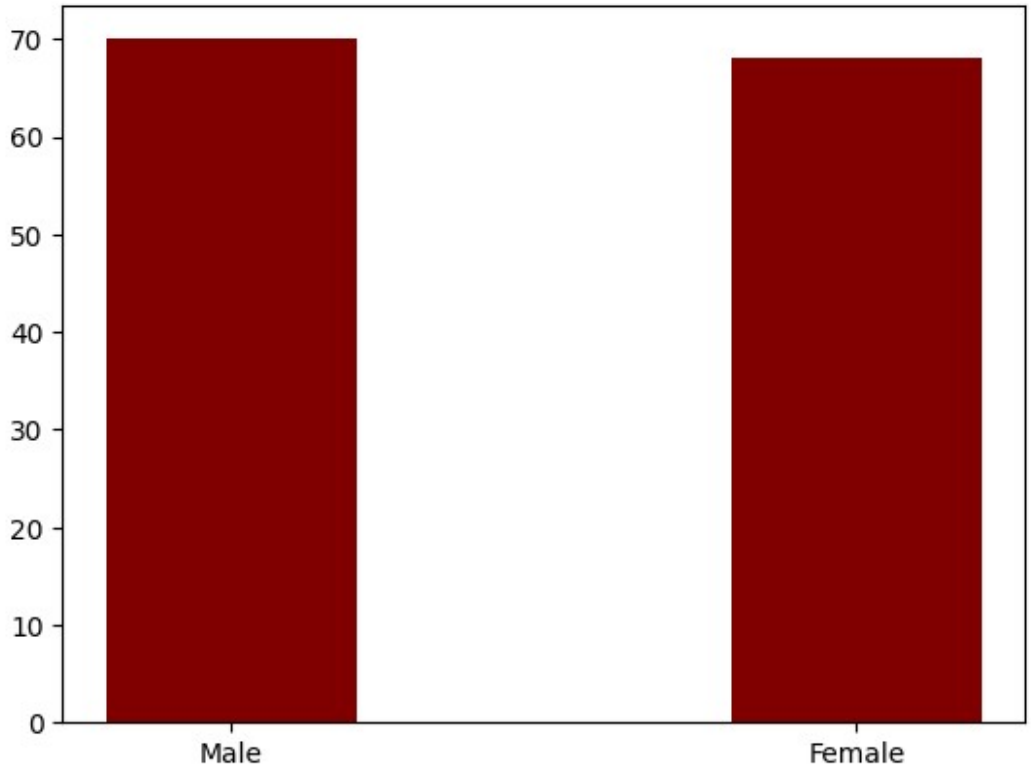
```
dataset.dtypes
```

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

Ploting some graphs

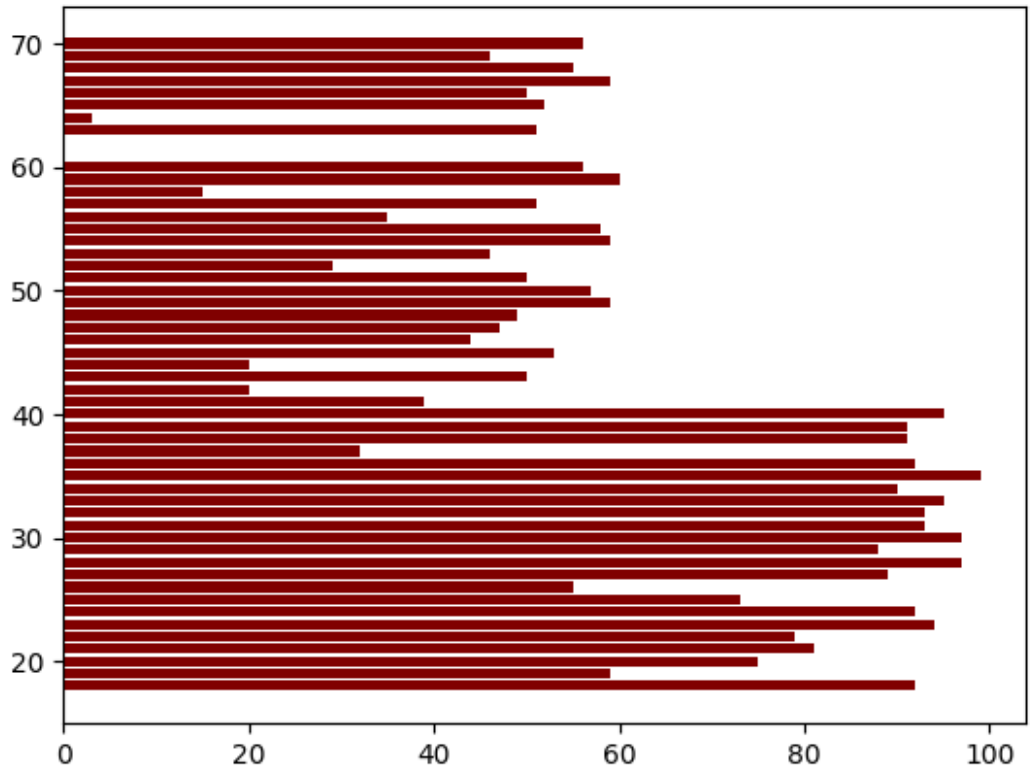
```
plt.bar(dataset['Genre'], dataset['Age'], color='maroon', width=0.4)
```

```
<BarContainer object of 200 artists>
```



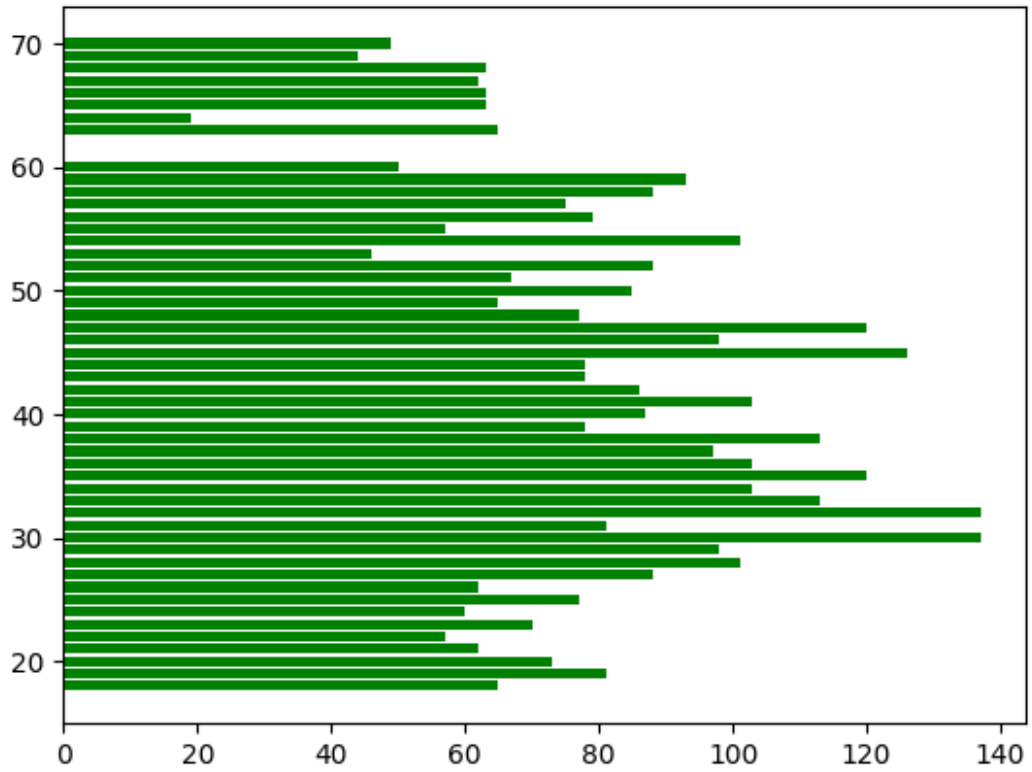

```
plt.barh(dataset['Age'], dataset['Spending Score (1-100)'],
color='maroon')
```

```
<BarContainer object of 200 artists>
```



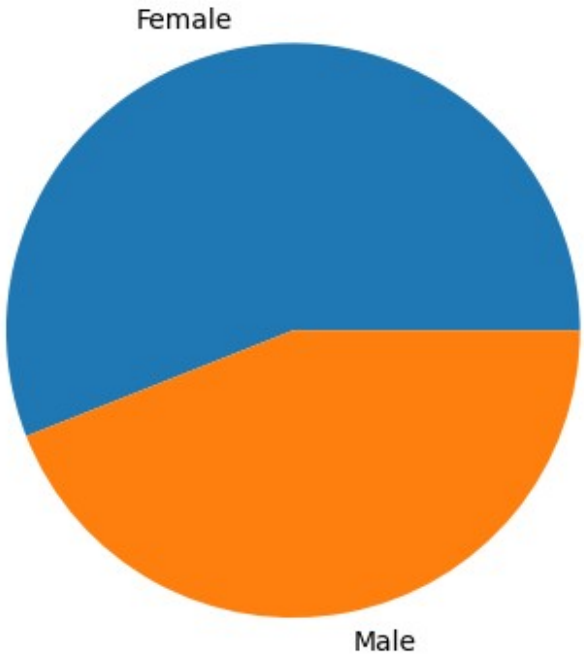

```
plt.barh(dataset['Age'], dataset['Annual Income (k$)'], color='green')
```

```
<BarContainer object of 200 artists>
```



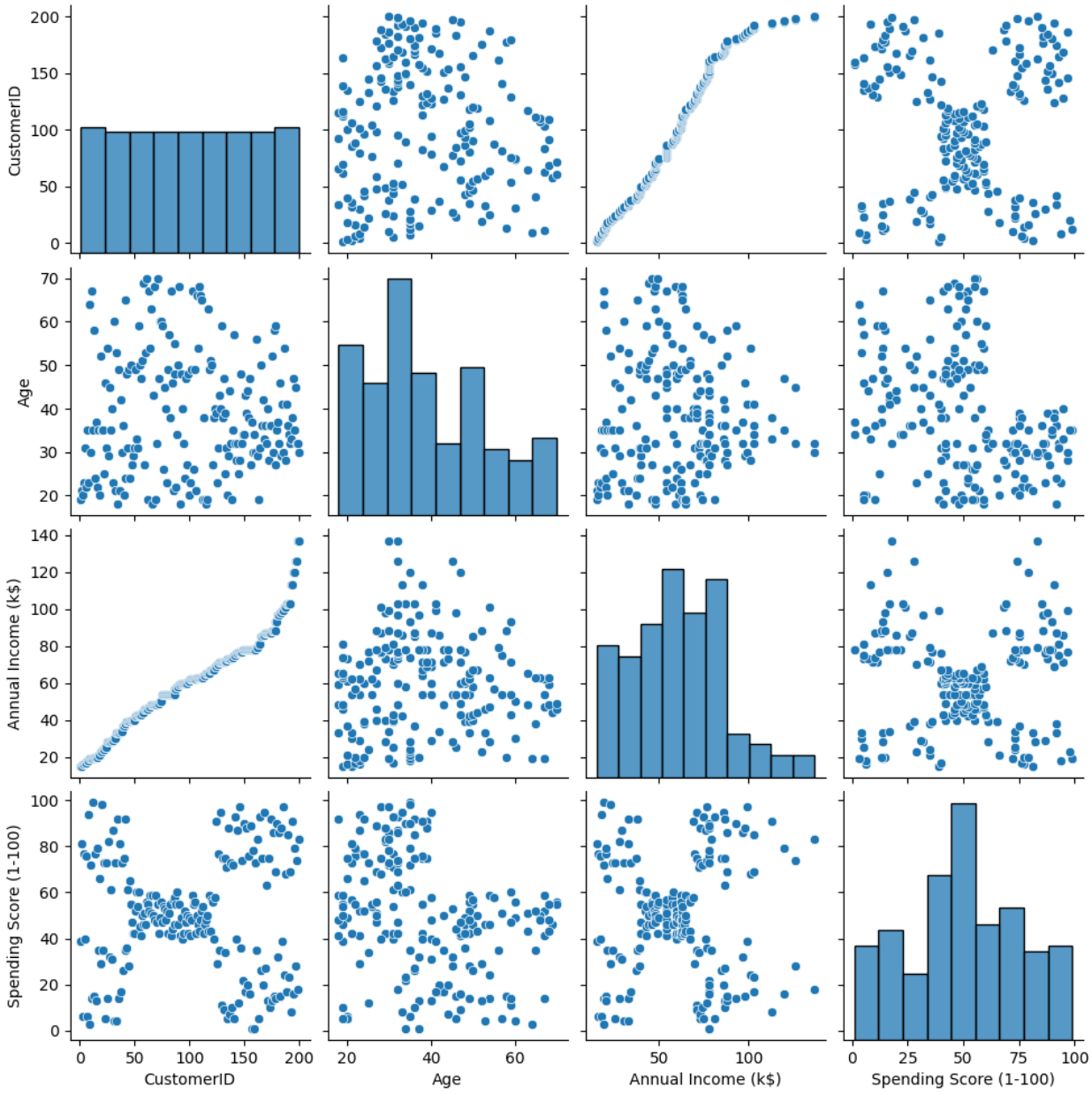

```
freq = dataset['Genre'].value_counts()
plt.pie(freq.values, labels=freq.index)
```

```
([<matplotlib.patches.Wedge at 0x1354b80e0>,
  <matplotlib.patches.Wedge at 0x135669430>],
 [Text(-0.20611945413751356, 1.080515974257694, 'Female'),
  Text(0.20611945413751367, -1.080515974257694, 'Male')])
```



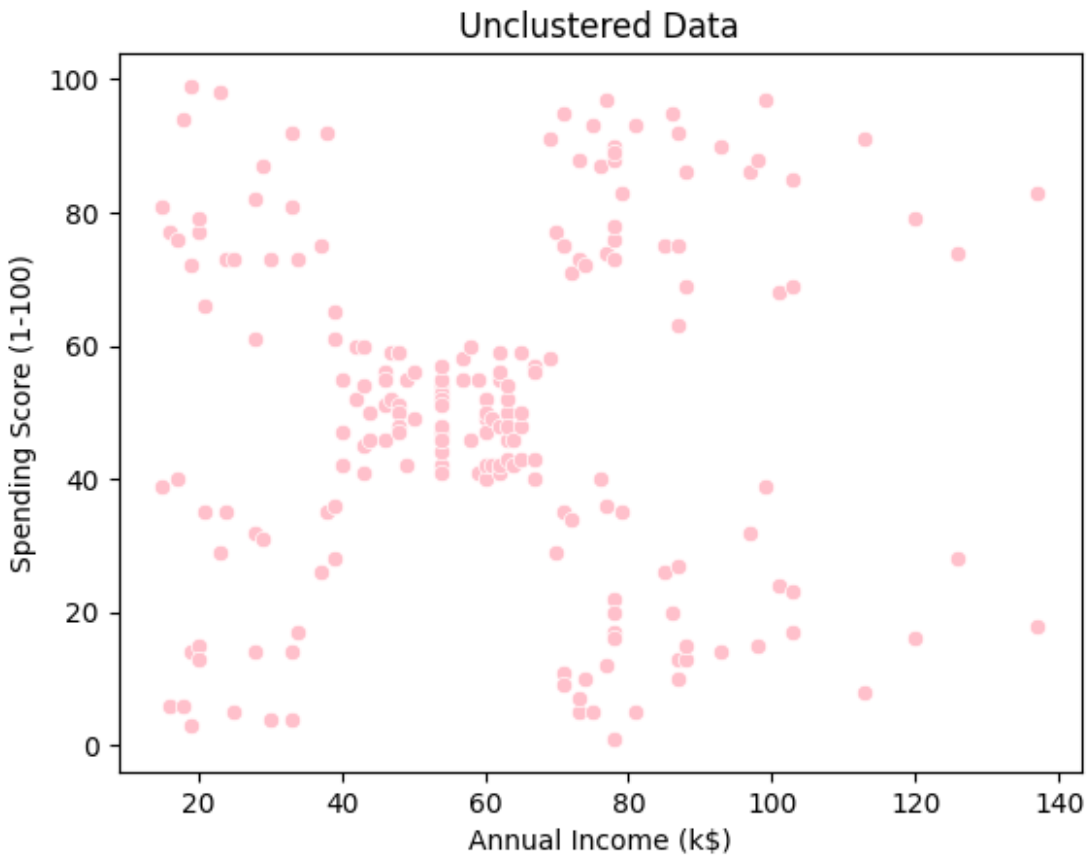

```
sns.pairplot(dataset)
```

```
<seaborn.axisgrid.PairGrid at 0x135232990>
```




```
plt.title("Unclustered Data")
sns.scatterplot(x=dataset['Annual Income (k$)'], y=dataset['Spending
Score (1-100)'], color='pink')
```

```
<Axes: title={'center': 'Unclustered Data'}, xlabel='Annual Income
(k$)', ylabel='Spending Score (1-100)'>
```



Making imp changes

1. Drop the customer_id column

2. Label Encode the Gender column

```
#dataset.drop('CustomerID', axis=1, inplace=True)
#dataset

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
dataset['Genre'] = le.fit_transform(dataset['Genre'])       # Male->1,
Female->0

dataset
```

```
     CustomerID  Genre  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
..           ...    ...  ...                  ...                       ..
.
195          196      0   35                  120
79
196          197      0   45                  126
28
197          198      1   32                  126
74
198          199      1   32                  137
18
199          200      1   30                  137
83

[200 rows x 5 columns]
```

K-Means clustering

```
from sklearn.cluster import KMeans

X = dataset.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 x 2 columns]
```

```
model =  KMeans(n_clusters=3)
model.fit(X)
```

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

```
preds = model.predict(X)
preds
```

```
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)
```

Displaying model's inertia

```
model.inertia_
```

```
106348.37306211119
```

Silhouette_score

```
from sklearn.metrics import silhouette_score

score = silhouette_score(X, preds)
score
```

```
0.46761358158775435
```

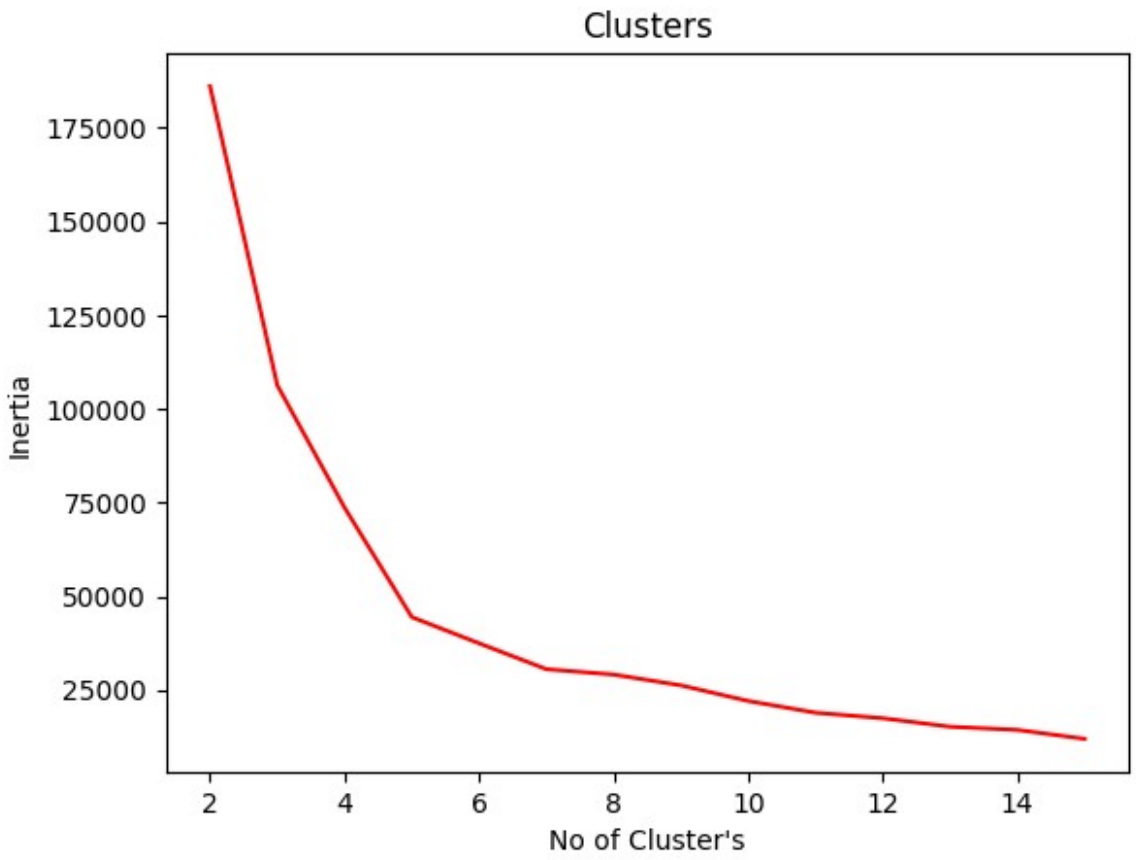
```
inertia = []
for i in range(2,16):
    model = KMeans(n_clusters=i)
    model.fit(X)

    inertia.append(model.inertia_)

inertia
```

```
[186186.60937282717,
 106348.37306211119,
 73679.78903948834,
 44448.45544793371,
 37455.984555160285,
 30552.71402546729,
 29114.982287331226,
 26230.947597654285,
 22066.344936947113,
 18929.8006127451,
 17456.658094098886,
 15195.506847012637,
 14350.331691271691,
 11962.18220015279]
```

Ploting graph of SSE scores

```
plt.title("Clusters")
sns.lineplot(x = range(2,16), y=inertia, color='red')
plt.xlabel("No of Cluster's")
plt.ylabel("Inertia")
```

```
Text(0, 0.5, 'Inertia')
```



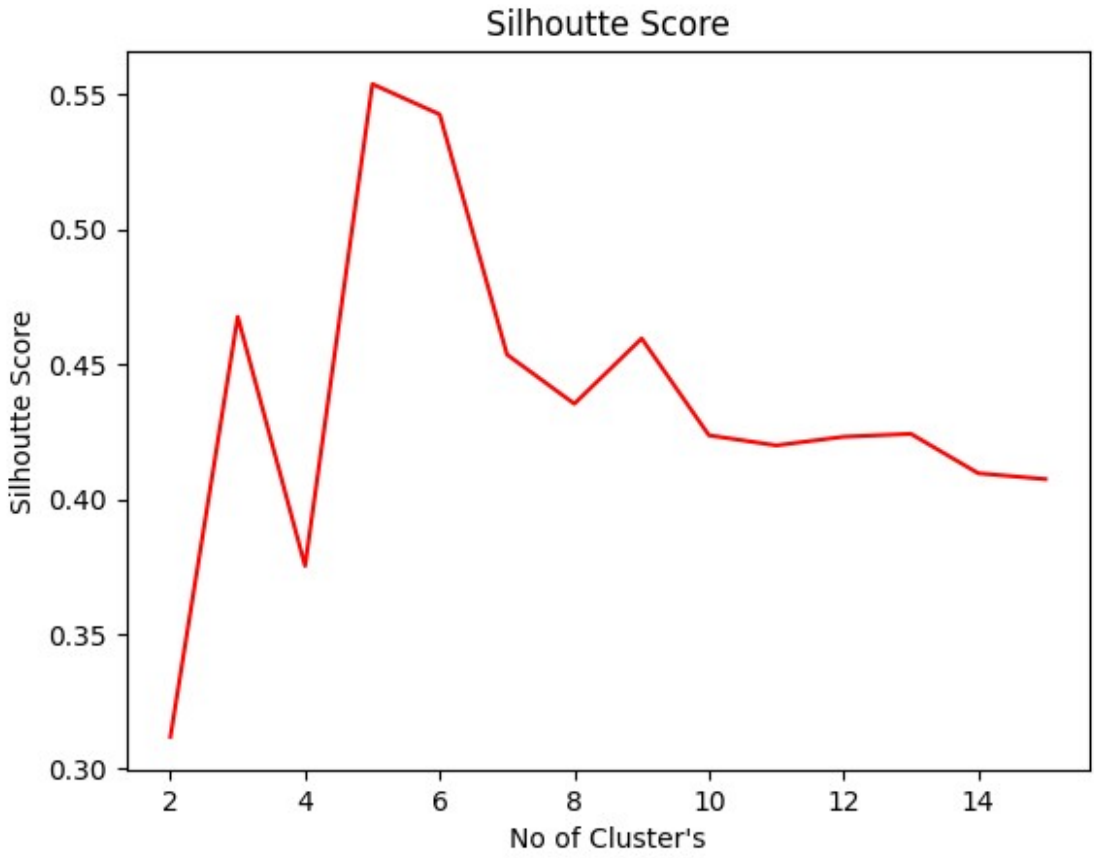

```
sil = []

for i in range(2,16):
    model = KMeans(n_clusters=i)
    model.fit(X)
    preds = model.predict(X)
    score = silhouette_score(X, preds)
    sil.append(score)

sil
```

```
[0.3117892685561811,
 0.46761358158775435,
 0.3751825606788526,
 0.553931997444648,
 0.5426749014789357,
 0.4535696333796981,
 0.435341079653073,
 0.4595491760122954,
 0.42362390476506256,
 0.4198914392028506,
 0.4230962892467343,
 0.42417774361579996,
```

```
 0.4095492277710123,
 0.40737920150094675]
```

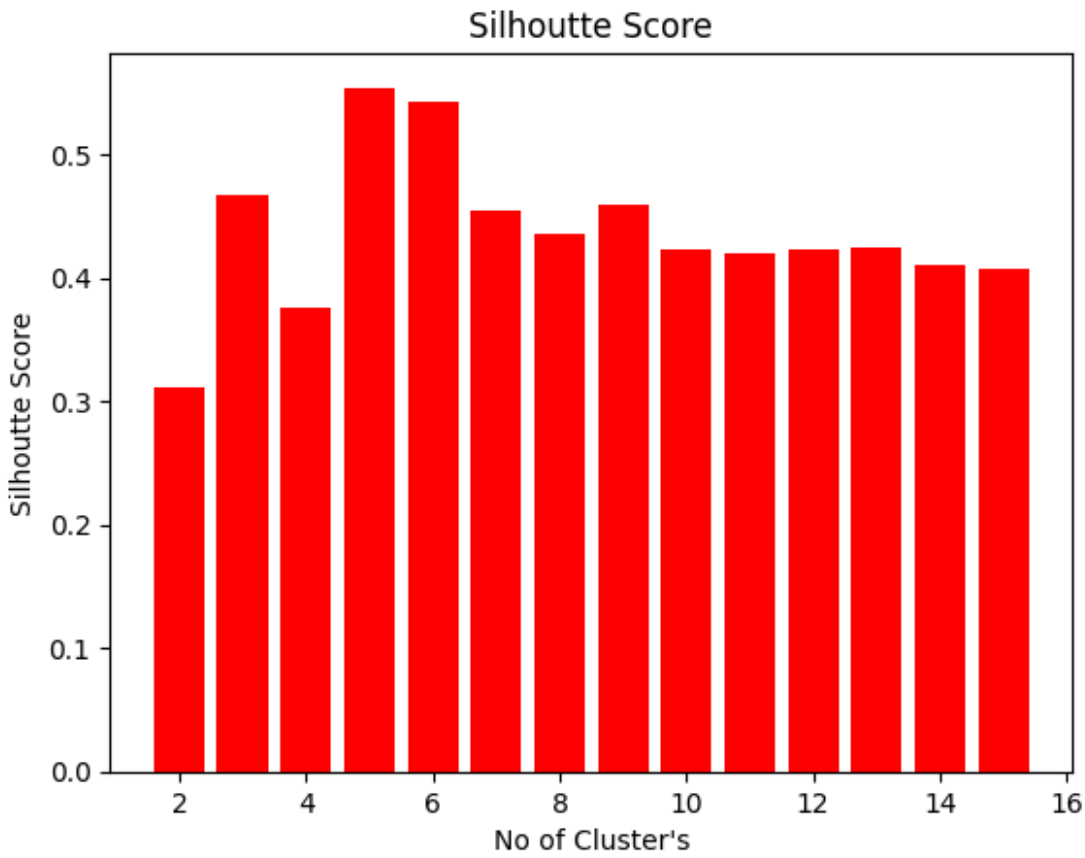
```
plt.title("Silhoutte Score")
sns.lineplot(x = range(2,16), y=sil, color='red')
plt.xlabel("No of Cluster's")
plt.ylabel("Silhoutte Score")
```

```
Text(0, 0.5, 'Silhoutte Score')
```




```
plt.title("Silhoutte Score")
plt.bar(range(2,16), sil, color='red')
plt.xlabel("No of Cluster's")
plt.ylabel("Silhoutte Score")
```

```
Text(0, 0.5, 'Silhoutte Score')
```



Building and training the model

```
from sklearn.model_selection import train_test_split

X_train, X_test = train_test_split(X, test_size=0.2)

len(X_train), len(X_test)
```

```
(160, 40)
```

```
model = KMeans(n_clusters=5)
model.fit(X_train)
```

```
KMeans(n_clusters=5)
```

```
train_preds = model.predict(X_train)

test_preds = model.predict(X_test)
test_preds
```

```
array([2, 3, 1, 4, 1, 3, 3, 3, 3, 2, 1, 3, 4, 4, 3, 2, 2, 4, 2, 2, 2,
4,
```

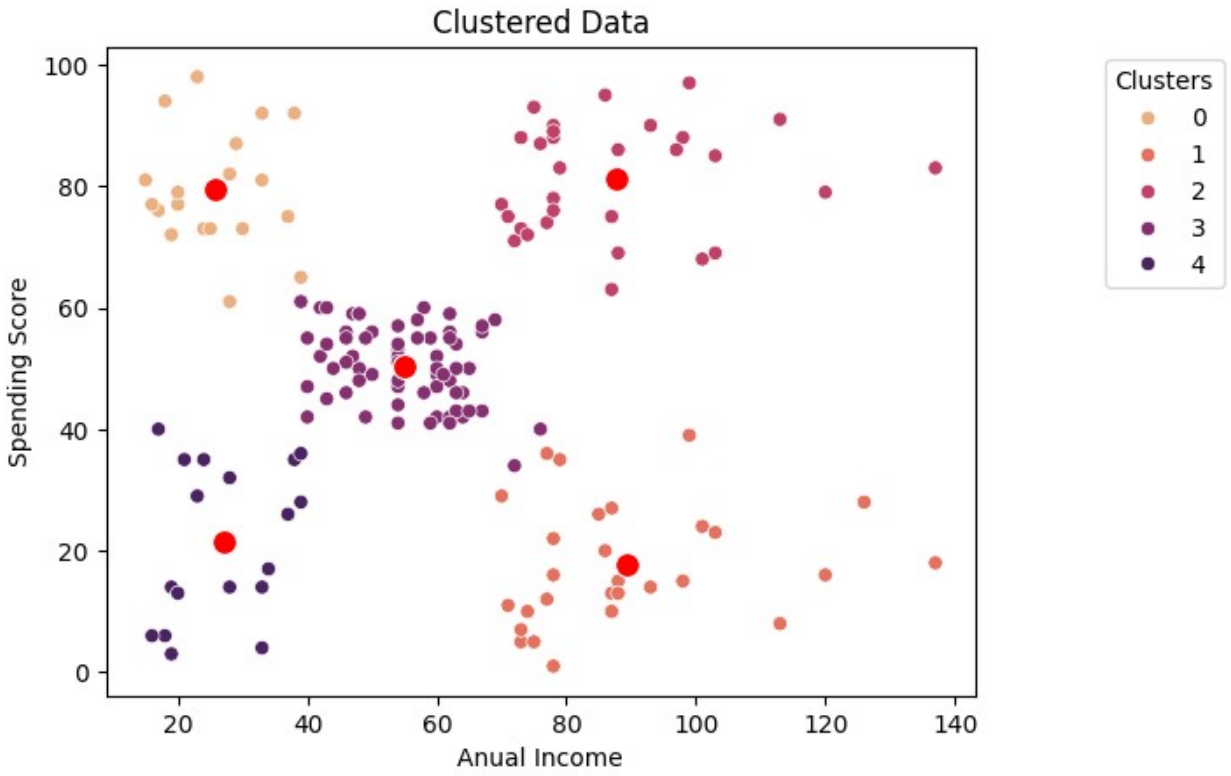
```
       2, 0, 3, 3, 2, 3, 3, 3, 0, 1, 1, 1, 0, 3, 3, 3, 3, 1],
dtype=int32)

model.inertia_

35031.603749759015

score = silhouette_score(X_test, test_preds)
score

0.5443655676402821
```

Finding out centers of the clusters

```
centers = model.cluster_centers_
centers

array([[25.89473684, 79.36842105],
       [89.32142857, 17.78571429],
       [87.66666667, 81.26666667],
       [55.06153846, 50.41538462],
       [27.        , 21.5       ]])
```

Ploting the Final Plot

```
X_train.columns

Index(['Annual Income (k$)', 'Spending Score (1-100)'],
dtype='object')

plt.title("Clustered Data")
sns.scatterplot(x=X_train['Annual Income (k$)'], y=X_train['Spending
Score (1-100)'], hue=train_preds, palette='flare')
sns.scatterplot(x=centers[:,0], y=centers[:,1], s=100, color='red')
plt.xlabel('Anual Income')
plt.ylabel('Spending Score')
plt.legend(title='Clusters', bbox_to_anchor=(1.3, 1), loc='upper
right')
plt.show()
```



Agglomerative Clustering

```
from sklearn.cluster import AgglomerativeClustering
model = AgglomerativeClustering(n_clusters=5)

cluster = model.fit(X_train)

train_preds = cluster.labels_
train_preds
```

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

```
sil = silhouette_score(X_train, train_preds)
sil
```

```
0.5546618760546765
```

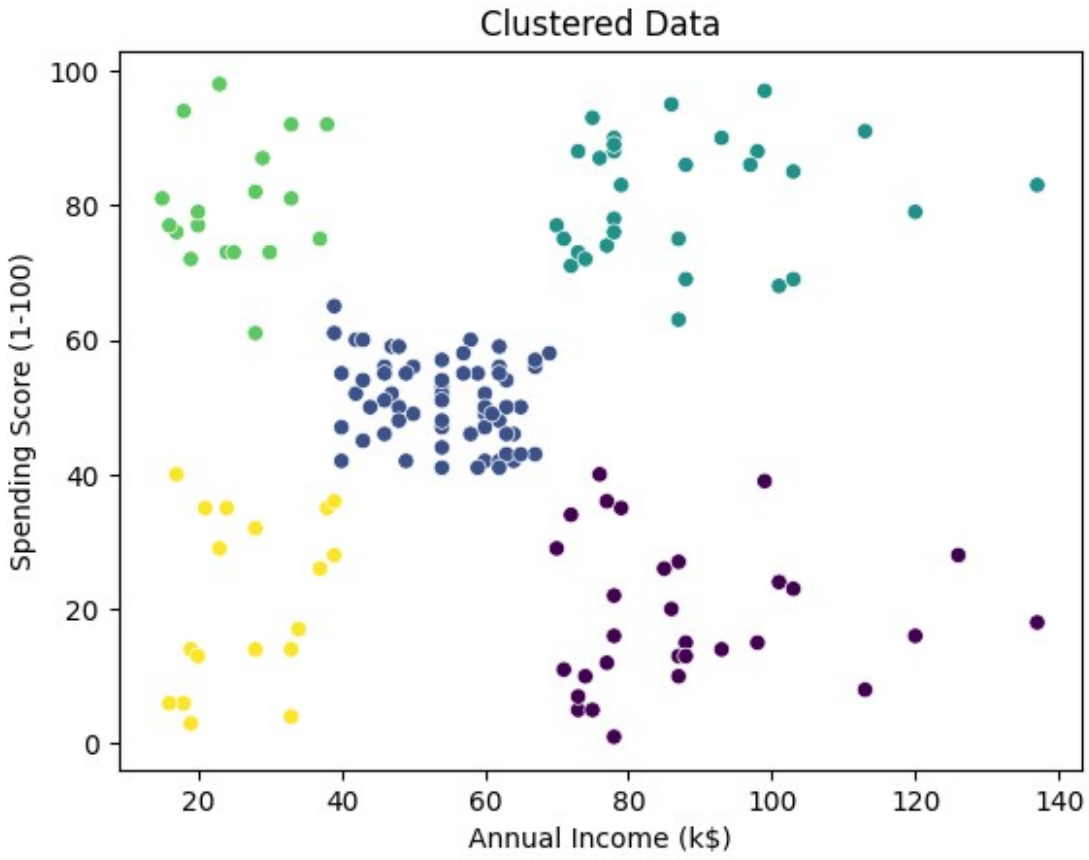
```
cluster = model.fit(X_test)
```

```
test_preds = cluster.labels_
test_preds
```

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

```
plt.title("Clustered Data")
sns.scatterplot(x = X_train['Annual Income (k$)'] , y =
X_train['Spending Score (1-100)'] , c = train_preds)
```

```
<Axes: title={'center': 'Clustered Data'}, xlabel='Annual Income
(k$)', ylabel='Spending Score (1-100)'>
```




```
from scipy.cluster.hierarchy import dendrogram, linkage
```

```
link = linkage(X_train, method='ward')

plt.figure(figsize=(10,7))
plt.title('Dendrogram')

dendrogram(link)
plt.xlabel('Distance')
plt.show()
```