```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt # Graphs & Visualization
import seaborn as sns
import os
import warnings
warnings.filterwarnings('ignore')
```

```
dataset = pd.read_csv('/content/drive/MyDrive/Sem 6/DBMI Lab/DMBI - Mini Project/Mall_Customers.csv')
```

```
dataset.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$)      196 non-null    float64
 4   Spending Score (1-100)  197 non-null    float64
dtypes: float64(2), int64(2), object(1)
memory usage: 7.9+ KB
```

```
dataset.head(22)
```

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **0** | 1 | Male | 19 | 15.0 | 39.0 |
| **1** | 2 | Male | 21 | 15.0 | 81.0 |
| **2** | 3 | Female | 20 | 16.0 | 6.0 |
| **3** | 4 | Female | 23 | 16.0 | 77.0 |
| **4** | 5 | Female | 31 | 17.0 | 40.0 |
| **5** | 6 | Female | 22 | 17.0 | 76.0 |
| **6** | 7 | Female | 35 | 18.0 | 6.0 |
| **7** | 8 | Female | 23 | 18.0 | 94.0 |
| **8** | 9 | Male | 64 | 19.0 | 3.0 |
| **9** | 10 | Female | 30 | 19.0 | 72.0 |
| **10** | 11 | Male | 67 | 19.0 | 14.0 |
| **11** | 12 | Female | 35 | 19.0 | 99.0 |
| **12** | 13 | Female | 58 | 20.0 | 15.0 |
| **13** | 14 | Female | 24 | 20.0 | 77.0 |
| **14** | 15 | Male | 37 | 20.0 | 13.0 |
| **15** | 16 | Male | 22 | 20.0 | 79.0 |
| **16** | 17 | Female | 35 | 21.0 | 35.0 |
| **17** | 18 | Male | 20 | 21.0 | 66.0 |
| **18** | 19 | Male | 52 | 23.0 | 29.0 |
| **19** | 20 | Female | 35 | 23.0 | 98.0 |
| **20** | 21 | Male | 35 | 24.0 | 35.0 |
| **21** | 22 | Male | 25 | NaN | 73.0 |

```
dataset.isnull().sum()
```

```
CustomerID              0
Gender                  0
Age                     0
Annual Income (k$)      4
Spending Score (1-100)  3
dtype: int64
```

```
dataset['Annual Income (k$)'].fillna(value = dataset['Annual Income (k$)'].mean(), inplace = True)


dataset['Spending Score (1-100)'].fillna(value = dataset['Spending Score (1-100)'].mean(), inplace = True)


dataset.isnull().sum()
```

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

```
d1= dataset['Annual Income (k$)']
q1, q3= np.percentile(d1,[25,75])
iqr = q3 - q1
lower_bound = q1 -(1.5 * iqr)
upper_bound = q3 +(1.5 * iqr)
print("Lower Bound Limit - ")
print(lower_bound)
print("\nUpper Bound Limit - ")
print(upper_bound)

med = dataset['Annual Income (k$)'].median()

dataset['Annual Income (k$)'] = np.where(dataset['Annual Income (k$)'] > upper_bound, med, dataset['Annual Income (k$)'])

# print('\nUpper Bound Outliers')
# for i in dataset['Annual Income (k$)']:
#   if i > upper_bound:
#     i = med

# print('\nLower Bound Outliers')
# for i in dataset['Annual Income (k$)']:
#   if i < lower_bound:
#     print(i)

# upper = np.where(dataset['Annual Income (k$)'] >= upper_bound)
# print(upper)
# lower = np.where(dataset['Annual Income (k$)'] <= lower_bound)
# print(lower)
```

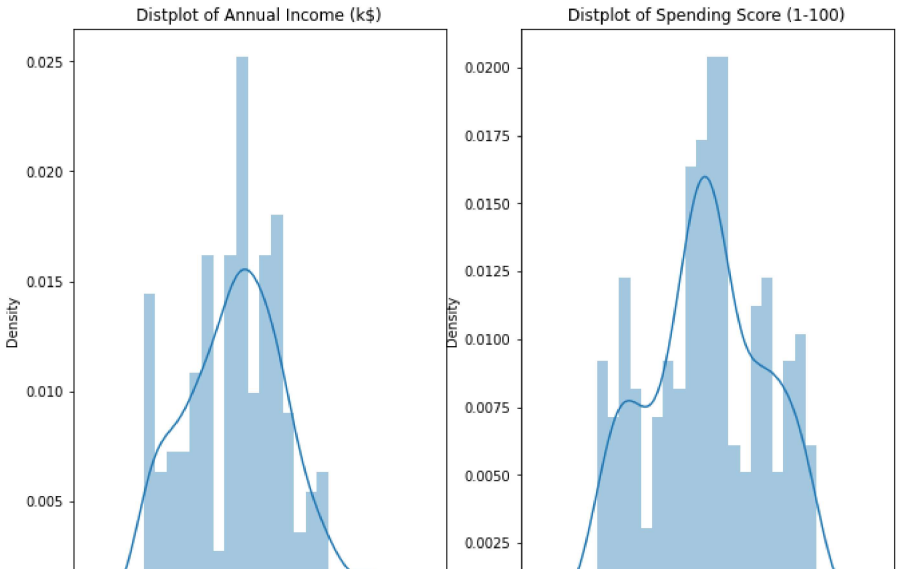
```
Lower Bound Limit -
-12.0

Upper Bound Limit -
132.0
```

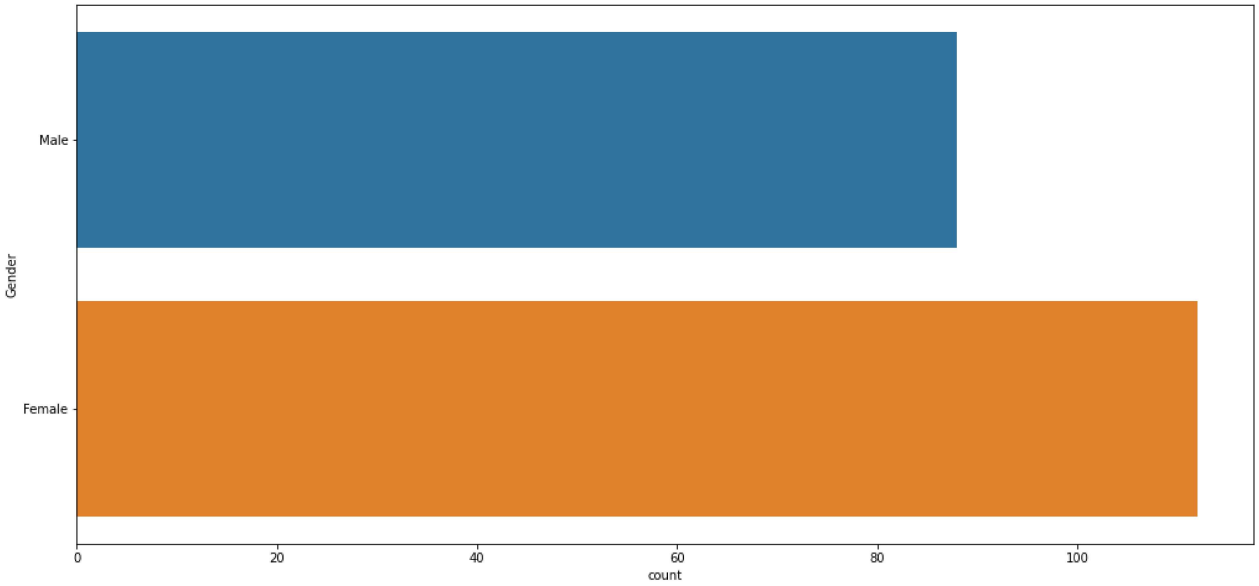
```
dataset.tail()
```

| | CustomerID | Gender | Age | Annual Income (k$) | Spending Score (1-100) |
|---|---|---|---|---|---|
| **195** | 196 | Female | 35 | 120.000000 | 79.0 |
| **196** | 197 | Female | 45 | 126.000000 | 28.0 |
| **197** | 198 | Male | 32 | 126.000000 | 74.0 |
| **198** | 199 | Male | 32 | 60.780612 | 18.0 |
| **199** | 200 | Male | 30 | 60.780612 | 83.0 |

```
plt.figure(1 , figsize = (17 , 8))
n = 0
for x in ['Annual Income (k$)' , 'Spending Score (1-100)']:
    n += 1
    plt.subplot(1 , 3 , n)
    sns.distplot(dataset[x] , bins = 20)
    plt.title('Distplot of {}'.format(x))
plt.show()
```

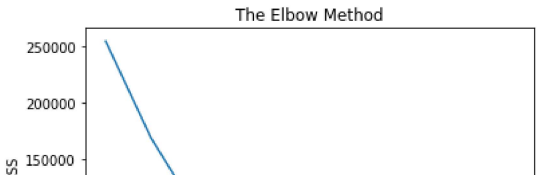



```
plt.figure(1 , figsize = (17 , 8))
sns.countplot(y = 'Gender' , data = dataset)
plt.show()
```




```
x = dataset.iloc[:, [3,4]].values
```

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1,11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



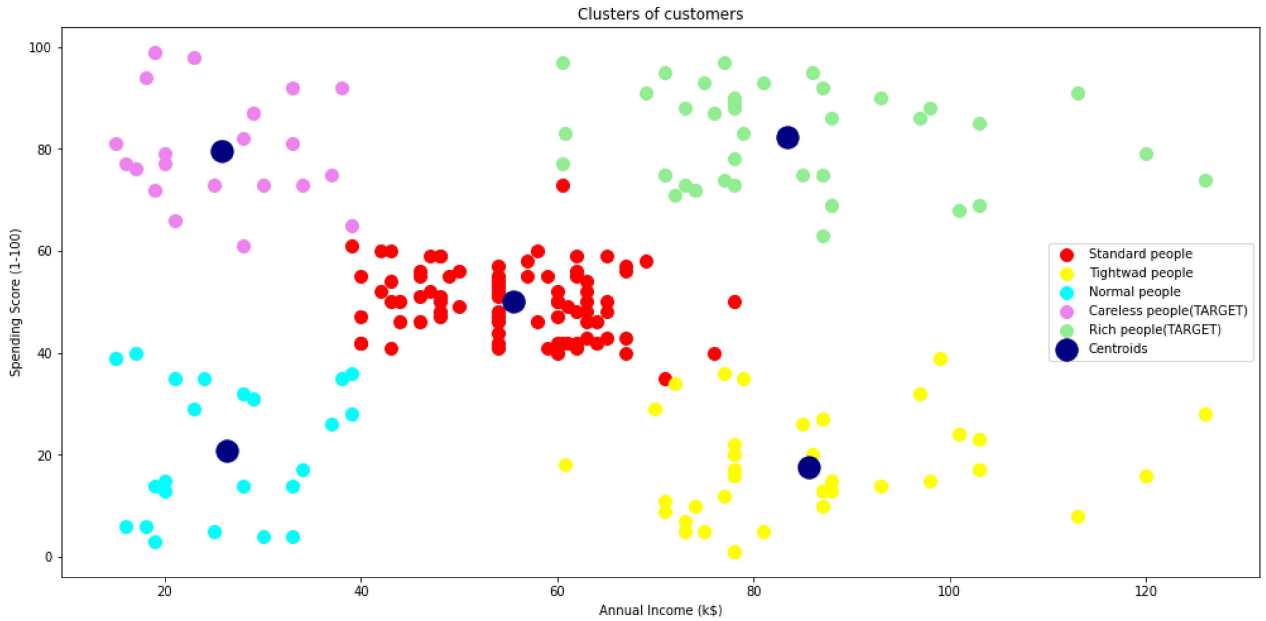

```
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
#Let's predict the x
y_kmeans = kmeans.fit_predict(x)
```

```
print(y_kmeans)
#We convert our prediction to dataframe so we can easily see this prediction in table form
df_pred = pd.DataFrame(y_kmeans)
df_pred.head()
```

```
[2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 0 2 3 2 3 2 3 2 3 2 3 2 3 2
 3 2 3 2 3 2 0 2 3 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 4 1 4 0 4 1 4 1 4 1 4 1 4 1 4 1 4 0 4 1 4 1 4
 1 4 1 4 1 0 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1
 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4]
```

| | 0 |
|---|---|
| **0** | 2 |
| **1** | 3 |
| **2** | 2 |
| **3** | 3 |
| **4** | 2 |

```
plt.figure(1 , figsize = (17 , 8))
plt.scatter(x[y_kmeans == 0, 0], x[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Standard people')
plt.scatter(x[y_kmeans == 1, 0], x[y_kmeans == 1, 1], s = 100, c = 'yellow', label = 'Tightwad people')
plt.scatter(x[y_kmeans == 2, 0], x[y_kmeans == 2, 1], s = 100, c = 'aqua', label = 'Normal people')
plt.scatter(x[y_kmeans == 3, 0], x[y_kmeans == 3, 1], s = 100, c = 'violet', label = 'Careless people(TARGET)')
plt.scatter(x[y_kmeans == 4, 0], x[y_kmeans == 4, 1], s = 100, c = 'lightgreen', label = 'Rich people(TARGET)')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'navy', label = 'Centroids')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```




```
#Cluster 0 (Red Color) -> Earning medium but spending medium
#cluster 1 (Yellow Colr) -> Earning High but spending very less
```

```
#cluster 2 (Aqua Color) -> Earning is low & spending is low
#cluster 3 (Violet Color) -> Earning is less but spending more -> Mall can target this type of people
#Cluster 4 (Lightgereen Color) -> Earning High & spending more -> Mall can target this type of people
#Navy color small circles is our Centroids


c0 = 0
c1 = 0
c2 = 0
c3 = 0
c4 = 0
for i in y_kmeans:
  if i == 0:
    c0 = c0 + 1
  elif i == 1:
    c1 = c1 + 1
  elif i == 2:
    c2 = c2 + 1
  elif i == 3:
    c3 = c3 + 1
  elif i == 4:
    c4 = c4 + 1

# total dataset value is 200. therefore percentage is count * 0.5

print("Percentage of Standard People (Earning medium but spending medium) : " + str(c0 * 0.5))
print("Percentage of Tightwad People (Earning High but spending very less) : " + str(c1 * 0.5))
print("Percentage of Normal People (Earning is low & spending is low) : " + str(c2 * 0.5))
print("Percentage of Careless People (Earning is less but spending more)[TARGET CUSTOMER SEGMENT] : " + str(c3 * 0.5))
print("Percentage of Rich People (Earning High & spending more)[TARGET CUSTOMER SEGMENT] : " + str(c4 * 0.5))
```

```
Percentage of Standard People (Earning medium but spending medium) : 41.0
Percentage of Tightwad People (Earning High but spending very less) : 18.0
Percentage of Normal People (Earning is low & spending is low) : 11.5
Percentage of Careless People (Earning is less but spending more)[TARGET CUSTOMER SEGMENT] : 10.5
Percentage of Rich People (Earning High & spending more)[TARGET CUSTOMER SEGMENT] : 19.0
```