Customer Segmentation Using KMeans Clustering Method

```
# install required library and packages

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings('ignore')
```

```
# load the csv

customer_data = pd.read_csv('/content/Mall_Customers.csv')
customer_data.head()
```

	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

Next steps: Generate code with customer_data View recommended plots New interactive sheet

```
customer_data.shape
```

```
(200, 5)
```

```
customer_data.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
```

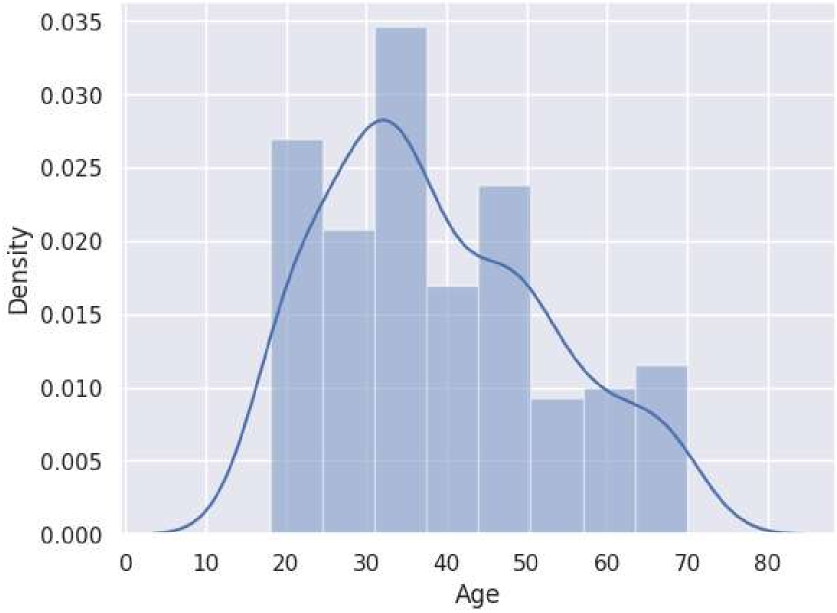
```
customer_data.isnull().sum()

# no null data value present in data
```

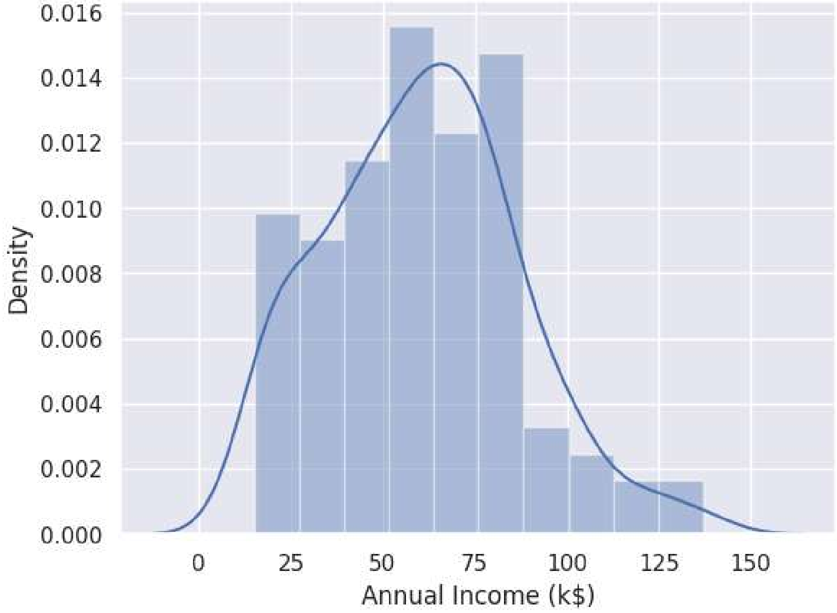
	0
CustomerID	0
Gender	0
Age	0
Annual Income (k$)	0
Spending Score (1-100)	0

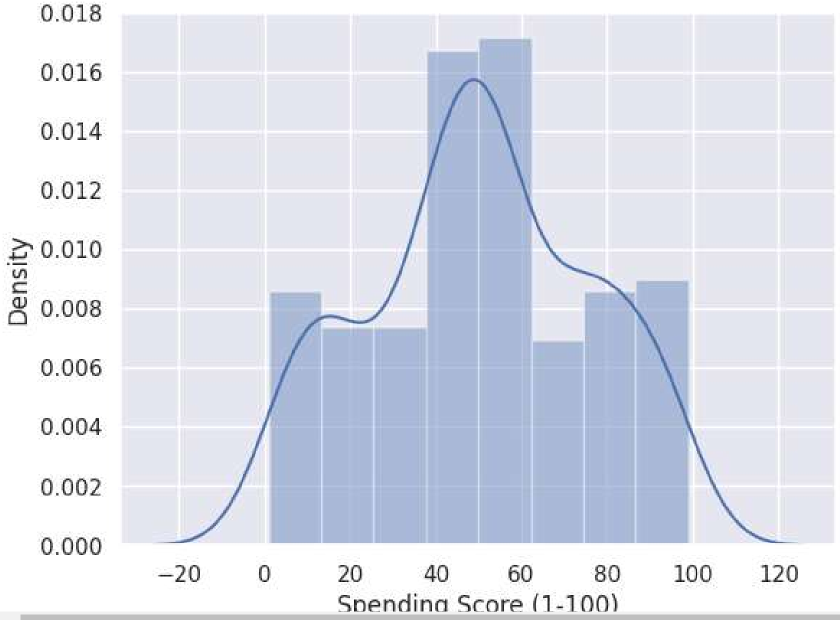
```
# Check Column Names

customer_data.columns
```

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

```
# generate density visuals (distplot) for columns

columns = ['Age', 'Annual Income (k$)','Spending Score (1-100)']

for i in columns:
  plt.figure()
  sns.distplot(customer_data[i])
```



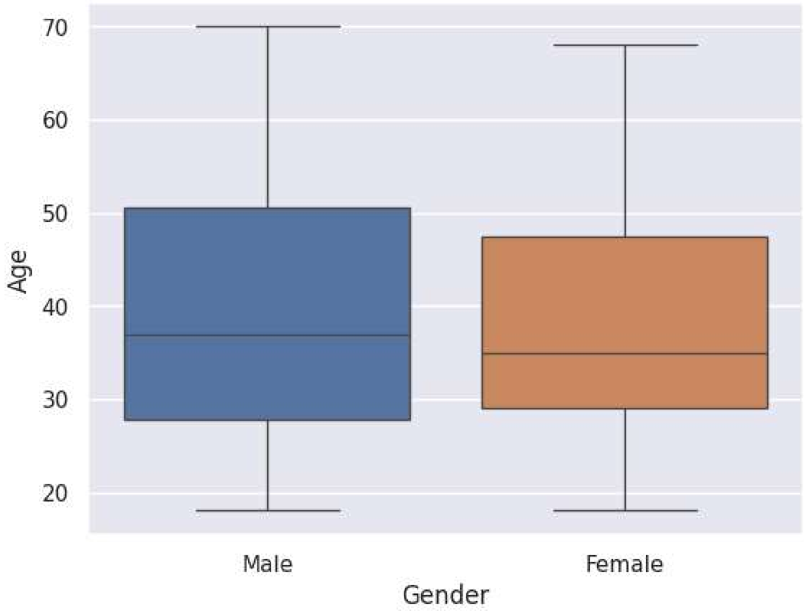


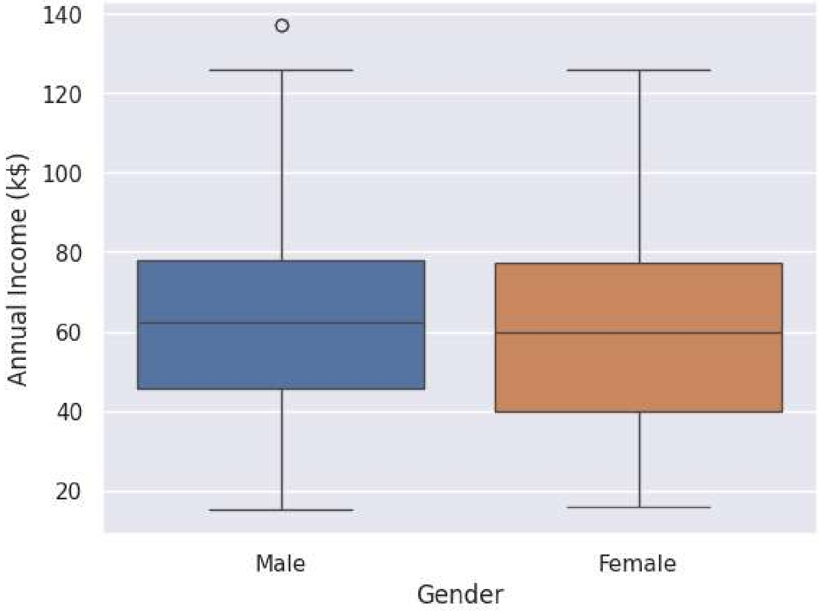


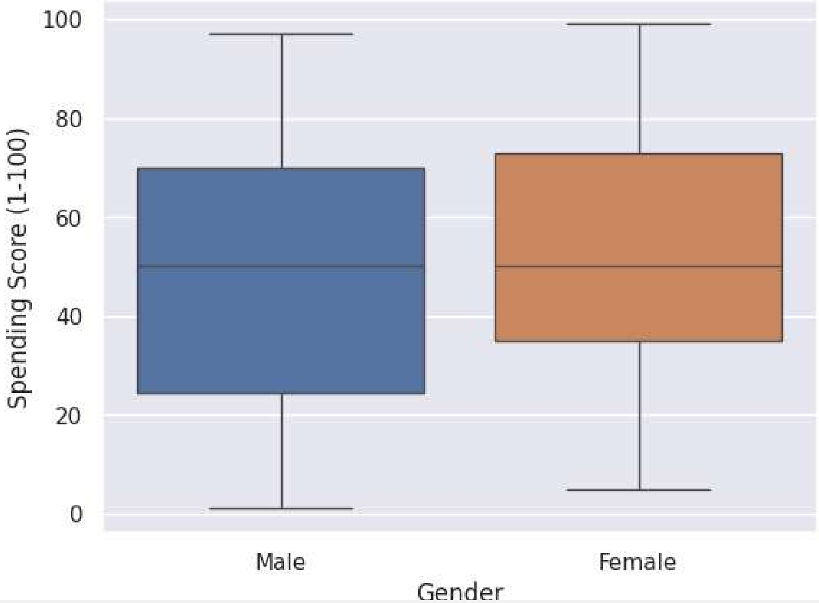
Lets Visualize all the columns with box plot

```
# lets visualise on basis of gender
```

```
for i in columns:
  plt.figure()
  sns.boxplot(data=customer_data, x = 'Gender', y = customer_data[i], hue=customer_data['Gender'])
```







Insights gained from gender box plot visuals

1. Females are lesser in age than men
2. Annual Income of Women is less than Men
3. Women spend more than the men

Using KMeans Clustering Method to segment the customers into clusters

```
X = customer_data.iloc[:,[3,4]].values
```

```
X
```

```
       [ 76,  40],
       [ 76,  87],
       [ 77,  12],
       [ 77,  97],
       [ 77,  36],
       [ 77,  74],
       [ 78,  22],
       [ 78,  90],
       [ 78,  17],
       [ 78,  88],
       [ 78,  20],
       [ 78,  76],
       [ 78,  16],
       [ 78,  89],
       [ 78,   1],
       [ 78,  78],
       [ 78,   1],
       [ 78,  73],
       [ 79,  35],
       [ 79,  83],
       [ 81,   5],
       [ 81,  93],
       [ 85,  26],
       [ 85,  75],
       [ 86,  20],
       [ 86,  95],
       [ 87,  27],
       [ 87,  63],
       [ 87,  13],
       [ 87,  75],
       [ 87,  10],
       [ 87,  92],
       [ 88,  13],
       [ 88,  86],
       [ 88,  15],
       [ 88,  69],
       [ 93,  14],
       [ 93,  90],
       [ 97,  32],
       [ 97,  86],
       [ 98,  15],
       [ 98,  88],
       [ 99,  39],
       [ 99,  97],
       [101,  24],
       [101,  68],
       [103,  17],
       [103,  85],
       [103,  23],
       [103,  69],
       [113,   8],
       [113,  91],
       [120,  16],
       [120,  79],
       [126,  28],
       [126,  74],
       [137,  18],
       [137,  83]])
```

Elbow Method (No of Joints)

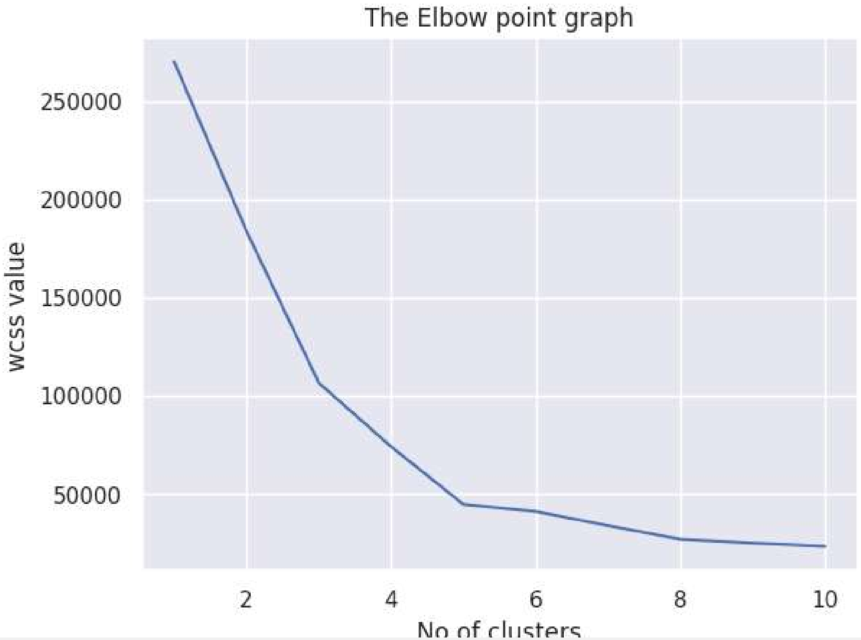
Choosing no of clusters using wcss (within cluster sum of squares methods) method

```
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_)
```

```
print(wcss)
```

```
7198, 44448.45544793369, 40825.16946386947, 33642.57922077922, 26686.837785187785, 24766.471609793436, 23103.122085983905]
```

```
# plot the graph

sns.set()
plt.plot(range(1,11),wcss)
plt.title("The Elbow point graph")
plt.xlabel("No of clusters")
plt.ylabel("wcss value")
plt.show()

# the last elbow point is at 5
# means cluster-size = 5
```



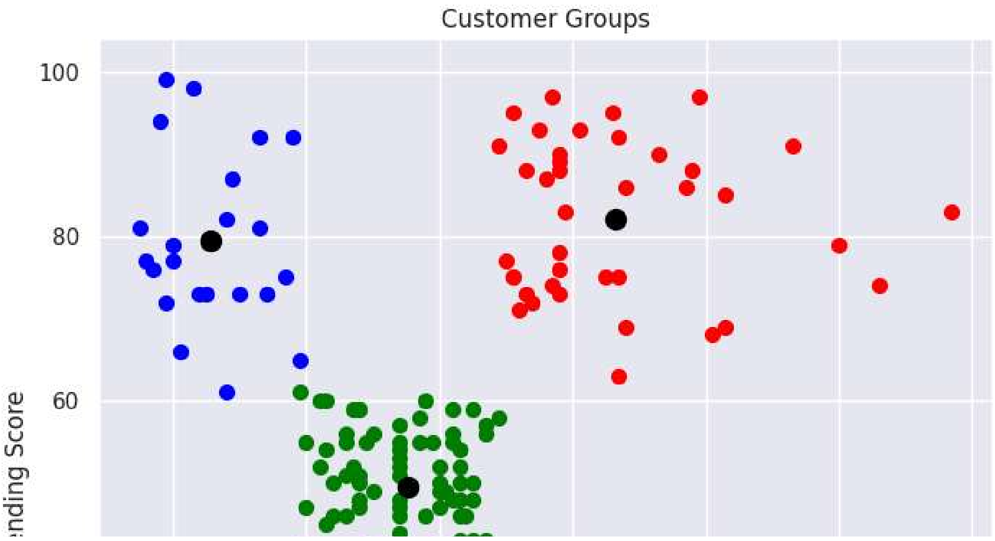
optimum number of clusters = 5

Training the kmeans cluster model

```
kmeans = KMeans(n_clusters=5, init='k-means++', random_state = 0)

#return a label to data based on their cluster

Y = kmeans.fit_predict(X)
print(Y)
```

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

```
# plotting the graph

plt.figure(figsize=(8,8))

plt.scatter(X[Y==0,0], X[Y==0,1],s=50, c='green', label = 'cluster 1' )
plt.scatter(X[Y==1,0], X[Y==1,1],s=50, c='red', label = 'cluster 2' )
plt.scatter(X[Y==2,0], X[Y==2,1],s=50, c='yellow', label = 'cluster 3' )
plt.scatter(X[Y==3,0], X[Y==3,1],s=50, c='purple', label = 'cluster 4' )
plt.scatter(X[Y==4,0], X[Y==4,1],s=50, c='blue', label = 'cluster 5' )

# plot the centroids

plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], s=100, c='black', label = 'centroids')

plt.title("Customer Groups")
plt.xlabel("Annual Income")
plt.ylabel("Spending Score")
plt.show()
```


Customer Groups
ending Score