IMPORTING THE DEPENDENCIES

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
```

DATA COLLECTION AND ANALYSIS

```
# loading the data from csv file to Pandas Dataframe
customer_data = pd.read_csv('/content/Mall_Customers.csv')
```

```
# first 5 rows in dataframe
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

```
# finding the number of rows and columns
customer_data.shape
```

```
(200, 5)
```

```
# getting some information about the dataset
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.describe()
```

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

```
# checkinhg for missing values
customer_data.isnull().sum()
```

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

CHOOSING THE ANNUAL INCOME COLUMN AND SPENDING SCORE COLUMN - ONLY RELEVANT ATTRIBUTES

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

```
[[ 15  39]
 [ 15  81]
 [ 16   6]
 [ 16  77]
 [ 17  40]
 [ 17  76]
 [ 18   6]
 [ 18  94]
 [ 19   3]
 [ 19  72]
```

```
 [ 19  14]
 [ 19  99]
 [ 20  15]
 [ 20  77]
 [ 20  13]
 [ 20  79]
 [ 21  35]
 [ 21  66]
 [ 23  29]
 [ 23  98]
 [ 24  35]
 [ 24  73]
 [ 25   5]
 [ 25  73]
 [ 28  14]
 [ 28  82]
 [ 28  32]
 [ 28  61]
 [ 29  31]
 [ 29  87]
 [ 30   4]
 [ 30  73]
 [ 33   4]
 [ 33  92]
 [ 33  14]
 [ 33  81]
 [ 34  17]
 [ 34  73]
 [ 37  26]
 [ 37  75]
 [ 38  35]
 [ 38  92]
 [ 39  36]
 [ 39  61]
 [ 39  28]
 [ 39  65]
 [ 40  55]
 [ 40  47]
 [ 40  42]
 [ 40  42]
 [ 42  52]
 [ 42  60]
 [ 43  54]
 [ 43  60]
 [ 43  45]
 [ 43  41]
 [ 44  50]
 [ 44  46]
```

CHOOSING THE NUMBER OF CLUSTERS

WCSS - WITHIN CLUSTER SUM OF SQUARES

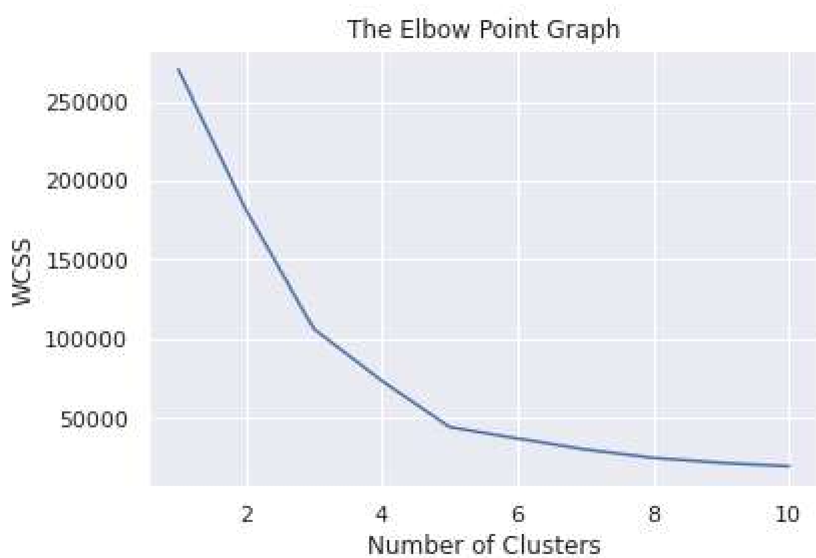
```
# FINDING WCSS VALUE FOR DIFFERENT NUMBER OF CLUSTERS

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_)
```

```
# Plot an Elbow graph
```

```
sns.set()
plt.plot(range(1,11),wcss)
plt.title('The Elbow Point Graph')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show()
```



OPTIMUM NUMBER OF CLUSTERS = 5 as significant drop occurs at x=5

Training the Clustering Model

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

# return a label for each data points based on their cluster
Y =kmeans.fit_predict(X)

print(Y)
```

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

VISUALIZING ALL THE CLUSTERS

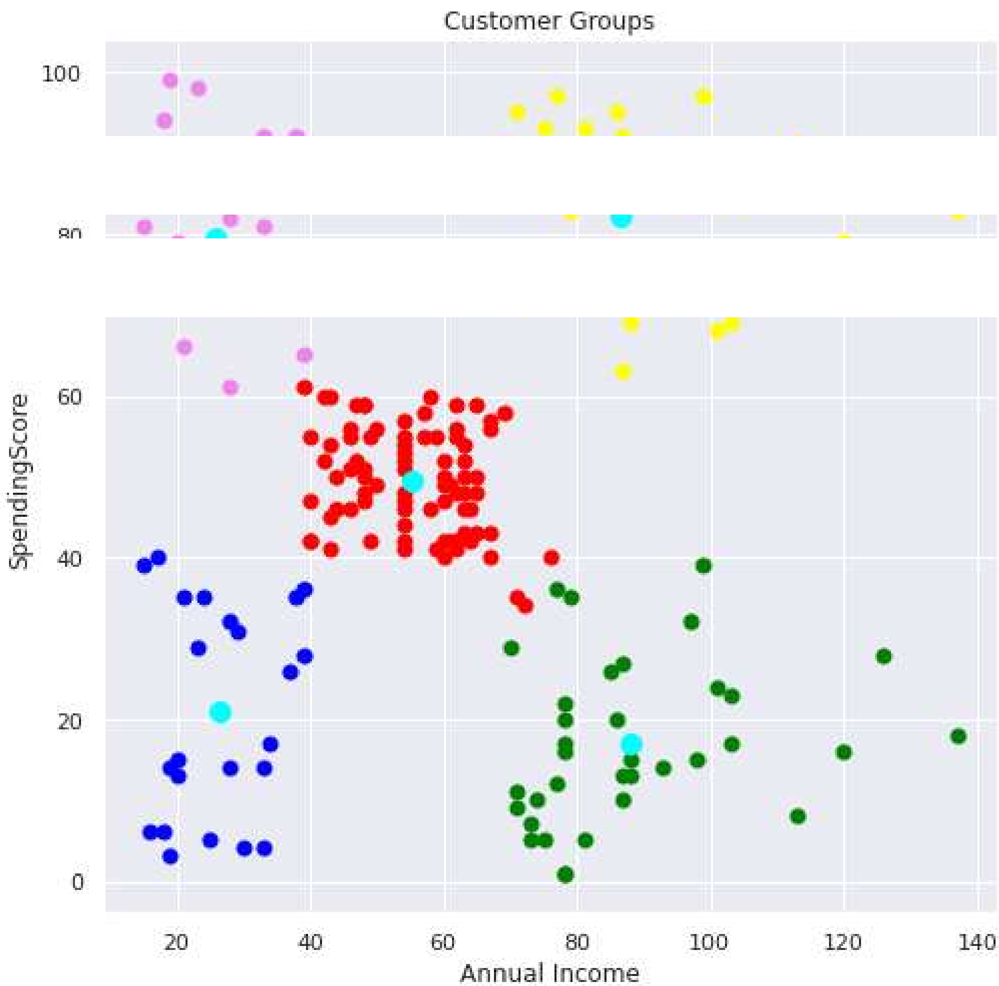
5 Clusters - 0,1,2,3,4

```
# PLOTTING ALL THE CLUSTERS AND THEIR CENTROIDS

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='violet',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='cyan',label=

plt.title('Customer Groups')
plt.xlabel('Annual Income')
plt.ylabel('SpendingScore')
plt.show()
```



✓ 0s completed at 4:58 PM