Importing Libraries

```
In [8]:
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler
```

Loading the data

```
In [9]:
df= pd.read_csv(r'C:\Users\bhuva\Desktop\Customers.csv', index_col = 0) #loads the csv file into a pandas dataframe
df.head() #returns the first 5 rows
```

Out[9]:

	Genre	Age	Annual Income (k$)	Spending Score (1-100)
CustomerID				
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40

Renaming a column in the dataset

```
In [10]:
df.rename(columns= {'Genre': 'Gender'}, inplace = True) #To rename column 2 from Genre to Gender
df.head() #Checking if the correction has been effected
```

Out[10]:

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
CustomerID				
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40

Checking data types

```
In [12]:
df.dtypes #returns the data types of the variables
```

```
Out[12]:
Gender                    object
Age                        int64
Annual Income (k$)         int64
Spending Score (1-100)     int64
dtype: object
```

Checking shape

```
In [13]:
df.shape #retuns the number of rows and columns in the dataset.
```

```
Out[13]:
(200, 4)
```

Descriptive statistics

```
In [14]:
df.describe() #returns the descriptive statistics of the dataset.
```

Out[14]:

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

Looking for null or missing values

```
In [15]:
df.isnull().sum() #returns the number of missing values
```

```
Out[15]:
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

Looking for duplicated values

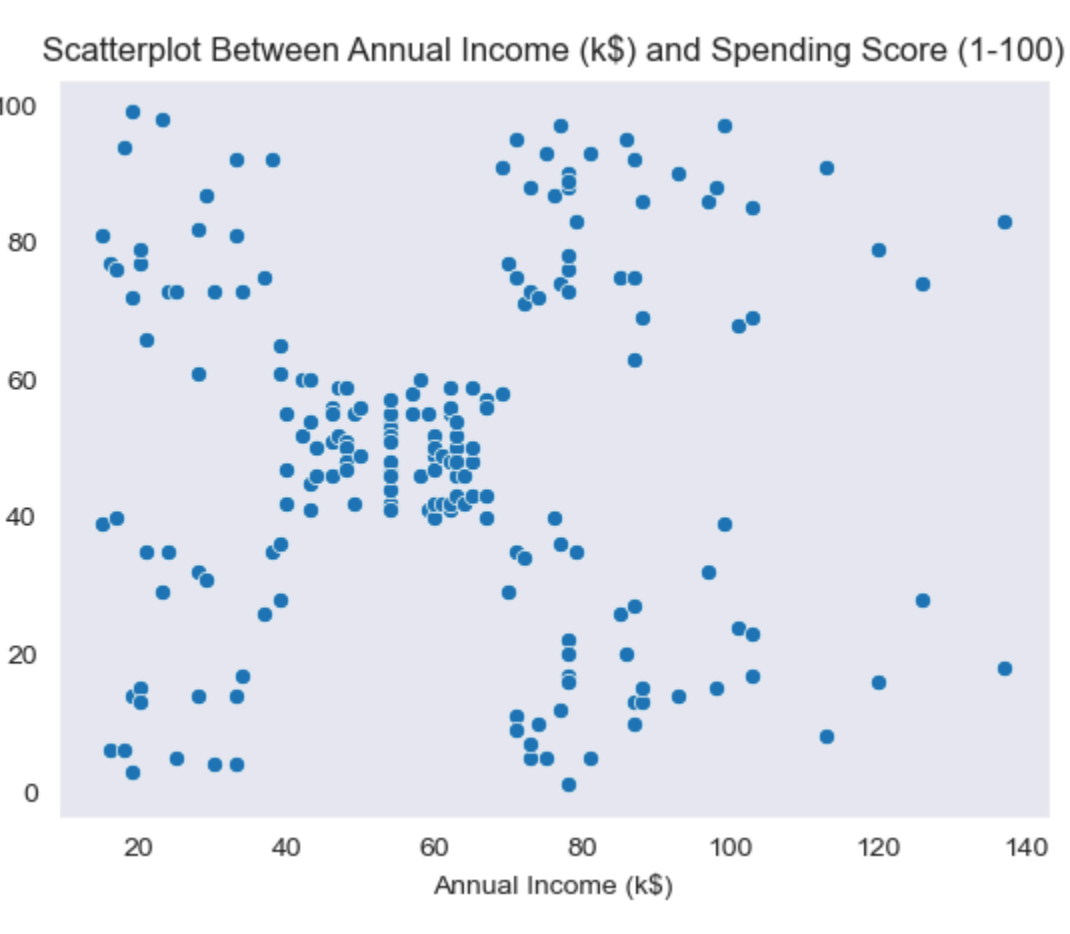
```
In [16]:
df.duplicated() #Checking for duplicate values.
```

```
Out[16]:
CustomerID
1      False
2      False
3      False
4      False
5      False
       ...
196    False
197    False
198    False
199    False
200    False
Length: 200, dtype: bool
```

Bivariate Analysis — Scatterplot

```
In [17]:
sns.set_style('dark')
sns.scatterplot(x = 'Annual Income (k$)', y = 'Spending Score (1-100)', data = df)
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Scatterplot Between Annual Income (k$) and Spending Score (1-100)')
```

```
Out[17]:
Text(0.5, 1.0, 'Scatterplot Between Annual Income (k$) and Spending Score (1-100)')
```



Feature Selection(Choosing the columns of interest for clustering)

```
In [18]:
X = df.loc[:,['Annual Income (k$)','Spending Score (1-100)']].values
```

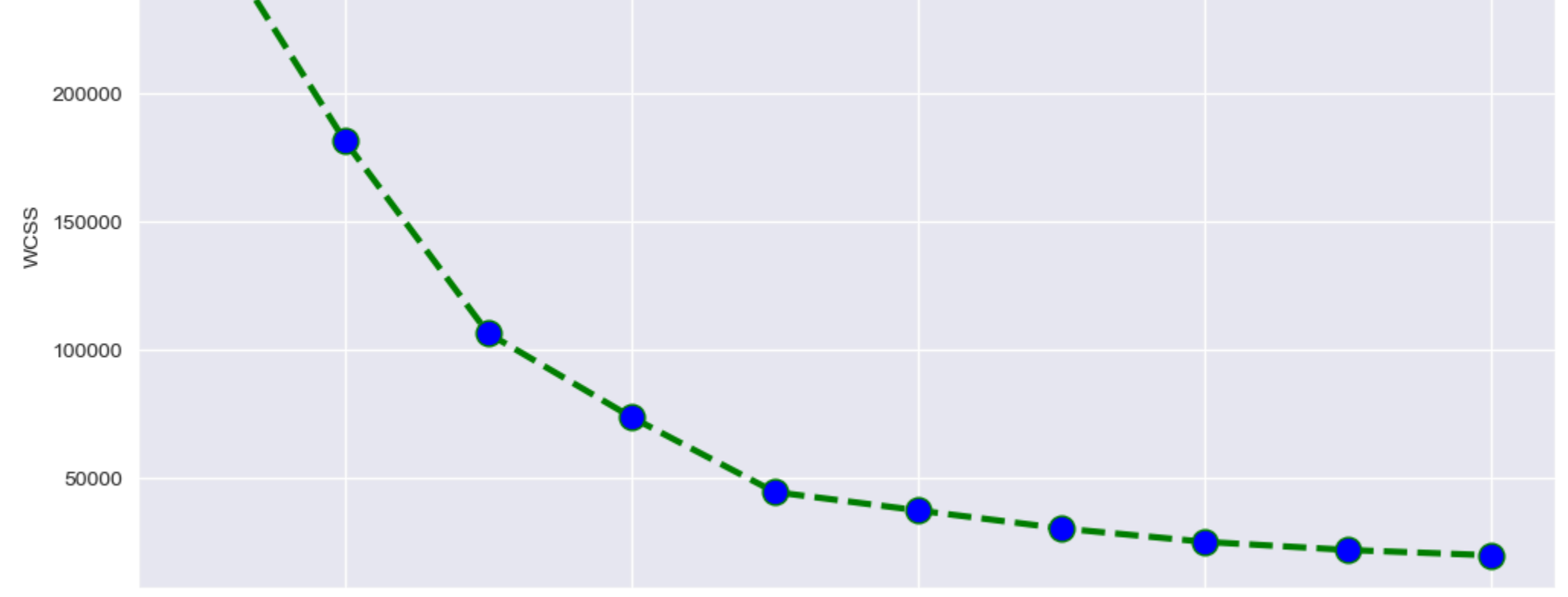
```
In [24]:
scaler = MinMaxScaler().fit(X) #It makes an object of the MinMaxScaler and then we fit it on our variable X.
print(scaler)
scaler.transform(X)
```

```
MinMaxScaler()
Out[24]:
array([[0.        , 0.3877551 ],
       [0.        , 0.81632653],
       [0.00819672, 0.05102041],
       [0.00819672, 0.7755102 ],
       [0.01639344, 0.39795918],
       [0.01639344, 0.76530612],
       [0.02459016, 0.05102041],
       [0.02459016, 0.94897959],
       [0.03278689, 0.02040816],
       [0.03278689, 0.7244898 ],
       [0.03278689, 0.13265306],
       [0.03278689, 1.        ],
       [0.04098361, 0.14285714],
       [0.04098361, 0.7755102 ],
       [0.04098361, 0.12244898],
       [0.04098361, 0.79591837],
       [0.04918033, 0.34693878],
       [0.04918033, 0.66326531],
       [0.06557377, 0.28571429],
       [0.06557377, 0.98979592],
       [0.07377049, 0.34693878],
       [0.07377049, 0.73469388],
       [0.08196721, 0.04081633],
       [0.08196721, 0.73469388],
       [0.10655738, 0.13265306],
       [0.10655738, 0.82653061],
       [0.10655738, 0.31632653],
       [0.10655738, 0.6122449 ],
       [0.1147541 , 0.30612245],
       [0.1147541 , 0.87755102],
       [0.12295082, 0.03061224],
       [0.12295082, 0.73469388],
       [0.14754098, 0.03061224],
       [0.14754098, 0.92857143],
       [0.14754098, 0.13265306],
       [0.14754098, 0.81632653],
       [0.1557377 , 0.16326531],
       [0.1557377 , 0.73469388],
       [0.18032787, 0.25510204],
       [0.18032787, 0.75510204],
       [0.18852459, 0.34693878],
       [0.18852459, 0.92857143],
       [0.19672131, 0.35714286],
       [0.19672131, 0.6122449 ],
       [0.19672131, 0.2755102 ],
       [0.19672131, 0.65306122],
       [0.20491803, 0.55102041],
       [0.20491803, 0.46938776],
       [0.20491803, 0.41836735],
       [0.20491803, 0.41836735],
       [0.22131148, 0.52040816],
       [0.22131148, 0.60204082],
       [0.2295082 , 0.54081633],
       [0.2295082 , 0.60204082],
       [0.2295082 , 0.44897959],
       [0.2295082 , 0.40816327],
       [0.23770492, 0.5       ],
       [0.23770492, 0.45918367],
       [0.25409836, 0.51020408],
       [0.25409836, 0.45918367],
       [0.25409836, 0.56122449],
       [0.25409836, 0.55102041],
       [0.26229508, 0.52040816],
       [0.26229508, 0.59183673],
       [0.2704918 , 0.51020408],
       [0.2704918 , 0.59183673],
       [0.2704918 , 0.5       ],
       [0.2704918 , 0.47959184],
       [0.2704918 , 0.59183673],
       [0.2704918 , 0.46938776],
       [0.27868852, 0.55102041],
       [0.27868852, 0.41836735],
       [0.28688525, 0.48979592],
       [0.28688525, 0.56122449],
       [0.31967213, 0.46938776],
       [0.31967213, 0.54081633],
       [0.31967213, 0.53061224],
       [0.31967213, 0.47959184],
       [0.31967213, 0.52040816],
       [0.31967213, 0.41836735],
       [0.31967213, 0.51020408],
       [0.31967213, 0.55102041],
       [0.31967213, 0.40816327],
       [0.31967213, 0.43877551],
       [0.31967213, 0.57142857],
       [0.31967213, 0.45918367],
       [0.3442623 , 0.58163265],
       [0.3442623 , 0.55102041],
       [0.35245902, 0.60204082],
       [0.35245902, 0.45918367],
       [0.36065574, 0.55102041],
       [0.36065574, 0.40816327],
       [0.36885246, 0.48979592],
       [0.36885246, 0.39795918],
       [0.36885246, 0.41836735],
       [0.36885246, 0.52040816],
       [0.36885246, 0.46938776],
       [0.36885246, 0.5       ],
       [0.37704918, 0.41836735],
       [0.37704918, 0.48979592],
       [0.3852459 , 0.40816327],
       [0.3852459 , 0.47959184],
       [0.3852459 , 0.59183673],
       [0.3852459 , 0.55102041],
       [0.3852459 , 0.56122449],
       [0.3852459 , 0.41836735],
       [0.39344262, 0.5       ],
       [0.39344262, 0.45918367],
       [0.39344262, 0.42857143],
       [0.39344262, 0.47959184],
       [0.39344262, 0.52040816],
       [0.39344262, 0.54081633],
       [0.40163934, 0.41836735],
       [0.40163934, 0.45918367],
       [0.40983607, 0.47959184],
       [0.40983607, 0.5       ],
       [0.40983607, 0.42857143],
       [0.40983607, 0.59183673],
       [0.42622951, 0.42857143],
       [0.42622951, 0.57142857],
       [0.42622951, 0.56122449],
       [0.42622951, 0.39795918],
       [0.44262295, 0.58163265],
       [0.44262295, 0.91836735],
       [0.45081967, 0.28571429],
       [0.45081967, 0.7755102 ],
       [0.45901639, 0.34693878],
       [0.45901639, 0.95918367],
       [0.45901639, 0.10204082],
       [0.45901639, 0.75510204],
       [0.45901639, 0.08163265],
       [0.45901639, 0.75510204],
       [0.46721311, 0.33673469],
       [0.46721311, 0.71428571],
       [0.47540984, 0.04081633],
       [0.47540984, 0.8877551 ],
       [0.47540984, 0.06122449],
       [0.47540984, 0.73469388],
       [0.48360656, 0.09183673],
       [0.48360656, 0.7244898 ],
       [0.49180328, 0.04081633],
       [0.49180328, 0.93877551],
       [0.5       , 0.39795918],
       [0.5       , 0.87755102],
       [0.50819672, 0.1122449 ],
       [0.50819672, 0.97959184],
       [0.50819672, 0.35714286],
       [0.50819672, 0.74489796],
       [0.51639344, 0.21428571],
       [0.51639344, 0.90816327],
       [0.51639344, 0.16326531],
       [0.51639344, 0.8877551 ],
       [0.51639344, 0.19387755],
       [0.51639344, 0.76530612],
       [0.51639344, 0.15306122],
       [0.51639344, 0.89795918],
       [0.51639344, 0.        ],
       [0.51639344, 0.78571429],
       [0.51639344, 0.        ],
       [0.51639344, 0.73469388],
       [0.52459016, 0.34693878],
       [0.52459016, 0.83673469],
       [0.54098361, 0.04081633],
       [0.54098361, 0.93877551],
       [0.57377049, 0.25510204],
       [0.57377049, 0.75510204],
       [0.58196721, 0.19387755],
       [0.58196721, 0.95918367],
       [0.59016393, 0.26530612],
       [0.59016393, 0.63265306],
       [0.59016393, 0.12244898],
       [0.59016393, 0.75510204],
       [0.59016393, 0.09183673],
       [0.59016393, 0.92857143],
       [0.59836066, 0.12244898],
       [0.59836066, 0.86734694],
       [0.59836066, 0.14285714],
       [0.59836066, 0.69387755],
       [0.63934426, 0.13265306],
       [0.63934426, 0.90816327],
       [0.67213115, 0.31632653],
       [0.67213115, 0.86734694],
       [0.68032787, 0.14285714],
       [0.68032787, 0.8877551 ],
       [0.68852459, 0.3877551 ],
       [0.68852459, 0.97959184],
       [0.70491803, 0.23469388],
       [0.70491803, 0.68367347],
       [0.72131148, 0.16326531],
       [0.72131148, 0.85714286],
       [0.72131148, 0.2244898 ],
       [0.72131148, 0.69387755],
       [0.80327869, 0.07142857],
       [0.80327869, 0.91836735],
       [0.86065574, 0.15306122],
       [0.86065574, 0.79591837],
       [0.90983607, 0.2755102 ],
       [0.90983607, 0.74489796],
       [1.        , 0.17346939],
       [1.        , 0.83673469]])
```

Choosing the Optimum Number of Clusters

```
In [27]:
plt.figure(figsize = (12,6))
plt.grid()
plt.plot(range(1,11),wcss, color='green', linestyle='dashed', linewidth = 3,
         marker='o', markerfacecolor='blue', markersize=12)
plt.title('The Elbow Point Graph')
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.show
```

```
Out[27]:
<function matplotlib.pyplot.show(close=None, block=None)>
```



Training the K-Means Clustering Model

```
In [30]:
kmeans= KMeans(n_clusters = 5, init = 'k-means++') #initialize the class object
label= kmeans.fit_predict(X) #returns a cluster number for each of the data points
print(label)
```

```
C:\Users\bhuva\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_i
nit` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\bhuva\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1436: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than
available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
[3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3
 2 3 2 3 2 3 0 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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 4 1 0 1 4 1 4 1 0 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 1 4 1 4 1 4 1 4 1 4 1 4 1]
```

Checking the centers of out clusters (Also known as Centroids)

```
In [31]:
print(kmeans.cluster_centers_)
```

```
[[55.2962963  49.51851852]
 [86.53846154 82.12820513]
 [25.72727273 79.36363636]
 [26.30434783 20.91304348]
 [88.2        17.11428571]]
```

Visualizing all the clusters

```
In [32]:
plt.figure(figsize=(8,8))
plt.scatter(X[label == 0,0], X[label== 0,1], s=50, c='green', label='Cluster 1')
plt.scatter(X[label == 1,0], X[label== 1,1], s=50, c='yellow', label='Cluster 2')
plt.scatter(X[label == 2,0], X[label== 2,1], s=50, c='red', label='Cluster 3')
plt.scatter(X[label == 3,0], X[label== 3,1], s=50, c='purple', label='Cluster 4')
plt.scatter(X[label == 4,0], X[label== 4,1], s=50, c='blue', label='Cluster 5')
plt.scatter(kmeans.cluster_centers_ [:,0], kmeans.cluster_centers_ [:,1], s= 100, c='black', marker= '*', label='Centriods') #Plotting the centriods
plt.title('Customer groups')
plt.xlabel('Annual Income')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```