```
In [104]:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
In [105]:
df = pd.read csv('/kaggle/input/mall-customers/Mall Customers.csv')
In [106]:
df.head()
Out[106]:
  CustomerID Gender Age Annual Income (k$) Spending Score (1-100)
0
                                   15
                                                     39
          1
              Male
                    19
1
          2
              Male
                    21
                                   15
                                                     81
                                                     6
2
          3 Female
                    20
                                   16
3
          4 Female
                    23
                                   16
                                                     77
          5 Female
                    31
                                   17
                                                     40
In [107]:
df.shape
Out[107]:
(200, 5)
In [108]:
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #
    Column
                              Non-Null Count Dtype
 0
   CustomerID
                              200 non-null int64
 1
   Gender
                              200 non-null object
 2
   Age
                              200 non-null int64
 3
                                            int64
   Annual Income (k$)
                              200 non-null
 4 Spending Score (1-100) 200 non-null
                                               int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
In [109]:
X = df.iloc[:, [3,4]].values
In [110]:
Χ
Out[110]:
array([[ 15,
              39],
       [ 15,
              81],
              6],
       [ 16,
       [ 16,
              77],
       [ 17,
              40],
              76],
       [ 17,
```

1 0

```
L ⊥ŏ,
         υJ,
[ 18,
        94],
        3],
[ 19,
[ 19,
        72],
[ 19,
        14],
[ 19,
        99],
[ 20,
        15],
 20,
        77],
[
  20,
[
        13],
  20,
        79],
[
  21,
        35],
[
[ 21,
        66],
        29],
[ 23,
        98],
        35],
[ 24,
        73],
[ 24,
[ 25,
        5],
[ 25,
        73],
[ 28,
        14],
[ 28,
        82],
[ 28,
        32],
[ 28,
        61],
[ 29,
        31],
[ 29,
        87],
[ 30,
        4],
        73],
[ 30,
[ 33,
        4],
[ 33,
        92],
[ 33,
        14],
[ 33,
        81],
[ 34,
        17],
        73],
[ 34,
[ 37,
        26],
[ 37,
        75],
[ 38,
        35],
[ 38,
        92],
[ 39,
        36],
[ 39,
        61],
[ 39,
        28],
[ 39,
        65],
        55],
[ 40,
[ 40,
        47],
[ 40,
        42],
[ 40,
        42],
[ 42,
        52],
[ 42,
        60],
  43,
[
        54],
        60],
[ 43,
 43,
[
        45],
[ 43,
        41],
[
 44,
        50],
[ 44,
        46],
[ 46,
        51],
[ 46,
        46],
[ 46,
        56],
[ 46,
        55],
[ 47,
        52],
[ 47,
        59],
[ 48,
        51],
[ 48,
        59],
        50],
[ 48,
[ 48,
        48],
[ 48,
        59],
[ 48,
        47],
[
  49,
        55],
[ 49,
        42],
  50,
[
        49],
[
  50,
        56],
 54,
[
        47],
```

[54,

[54,

[54,

54],

53], 48],

```
L 04,
        J∠J,
[ 54,
        42],
[ 54,
        51],
[ 54,
        55],
[ 54,
        41],
[ 54,
        44],
[ 54,
        57],
[ 54,
        46],
  57,
[
        58],
  57,
        55],
[
[ 58,
        60],
  58,
        46],
[
  59,
[
        55],
[ 59,
        41],
        49],
[ 60,
        40],
[ 60,
[ 60,
        42],
[ 60,
        52],
[ 60,
        47],
[ 60,
        50],
[ 61,
        42],
[ 61,
        49],
[ 62,
        41],
        48],
[ 62,
[ 62,
        59],
[ 62,
        55],
[ 62,
        56],
[ 62,
        42],
[ 63,
        50],
        46],
[ 63,
[ 63,
        43],
[ 63,
        48],
[ 63,
        52],
        54],
[ 63,
[ 64,
        42],
[ 64,
        46],
[ 65,
        48],
[ 65,
        50],
[ 65,
        43],
[ 65,
        59],
[ 67,
        43],
[ 67,
        57],
[ 67,
        56],
[ 67,
        40],
[ 69,
        58],
[ 69,
        91],
  70,
[
        29],
  70,
        77],
[
 71,
71,
71,
        35],
[
        95],
[
[
        11],
        75],
[
  71,
        9],
[ 71,
[ 71,
        75],
[ 72,
        34],
[ 72,
        71],
[ 73,
        5],
[ 73,
        88],
[ 73,
        7],
[ 73,
        73],
[ 74,
        10],
74,
        72],
 75,
        5],
[
  75,
        93],
[
  76,
        40],
[
 76,
77,
77,
77,
77,
        87],
[
[
        12],
        97],
        36],
[
[
        74],
```

[

[

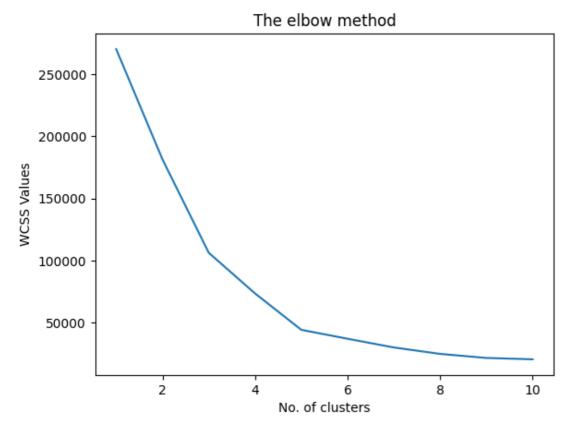
78,

22],

90],

```
[ / Ö,
              ⊥ / J ,
       [ 78,
               88],
       [ 78,
               20],
       [ 78,
               76],
       [ 78,
               16],
       [ 78,
               89],
       [ 78,
               1],
       [ 78,
               78],
       [ 78,
               1],
               73],
       [ 78,
         79,
       [
               35],
         79,
               83],
               5],
       [ 81,
       [ 81,
               93],
       [ 85,
               26],
       [ 85,
               75],
       [ 86,
               20],
       [ 86,
               95],
       [ 87,
               27],
       [ 87,
               63],
       [ 87,
               13],
       [ 87,
               75],
       [ 87,
               10],
               92],
       [ 87,
               13],
       [ 88,
       [ 88,
               86],
       [ 88,
               15],
       [ 88,
               69],
       [ 93,
               14],
       [ 93,
               90],
       [ 97,
               32],
               86],
       [ 97,
               15],
       [ 98,
       [ 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],
               28],
       [126,
       [126,
               74],
       [137,
               18],
       [137,
               83]])
In [111]:
from sklearn.cluster import KMeans
wcss = []
In [112]:
for i in range (1,11):
    kmeans = KMeans(n_clusters= i, init='k-means++', n_init=10, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
In [113]:
wcss = wcss[:10]
print(f"Length of wcss: {len(wcss)}")
Length of wcss: 10
In [114]:
```

```
plt.plot(range(1,11), wcss)
plt.title('The elbow method')
plt.xlabel('No. of clusters')
plt.ylabel('WCSS Values')
plt.show()
```



In [115]:

```
kmeansmodel = KMeans(n_clusters = 5, init='k-means++', n_init=10)
```

In [116]:

```
y_kmeans = kmeansmodel.fit_predict(X)
```

In [117]:

```
plt.scatter(X[y_kmeans == 0,0], X[y_kmeans == 0,1], s= 80, c = "red", label='Customer 1'
)
plt.scatter(X[y_kmeans == 1,0], X[y_kmeans == 1,1], s= 80, c = "blue", label='Customer 2
')
plt.scatter(X[y_kmeans == 2,0], X[y_kmeans == 2,1], s= 80, c = "yellow", label='Customer 3')
plt.scatter(X[y_kmeans == 3,0], X[y_kmeans == 3,1], s= 80, c = "cyan", label='Customer 4')
plt.scatter(X[y_kmeans == 4,0], X[y_kmeans == 4,1], s= 80, c = "black", label='Customer 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 100, c = 'magenta', label = 'Centroids')
plt.stabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

Clusters of customers

