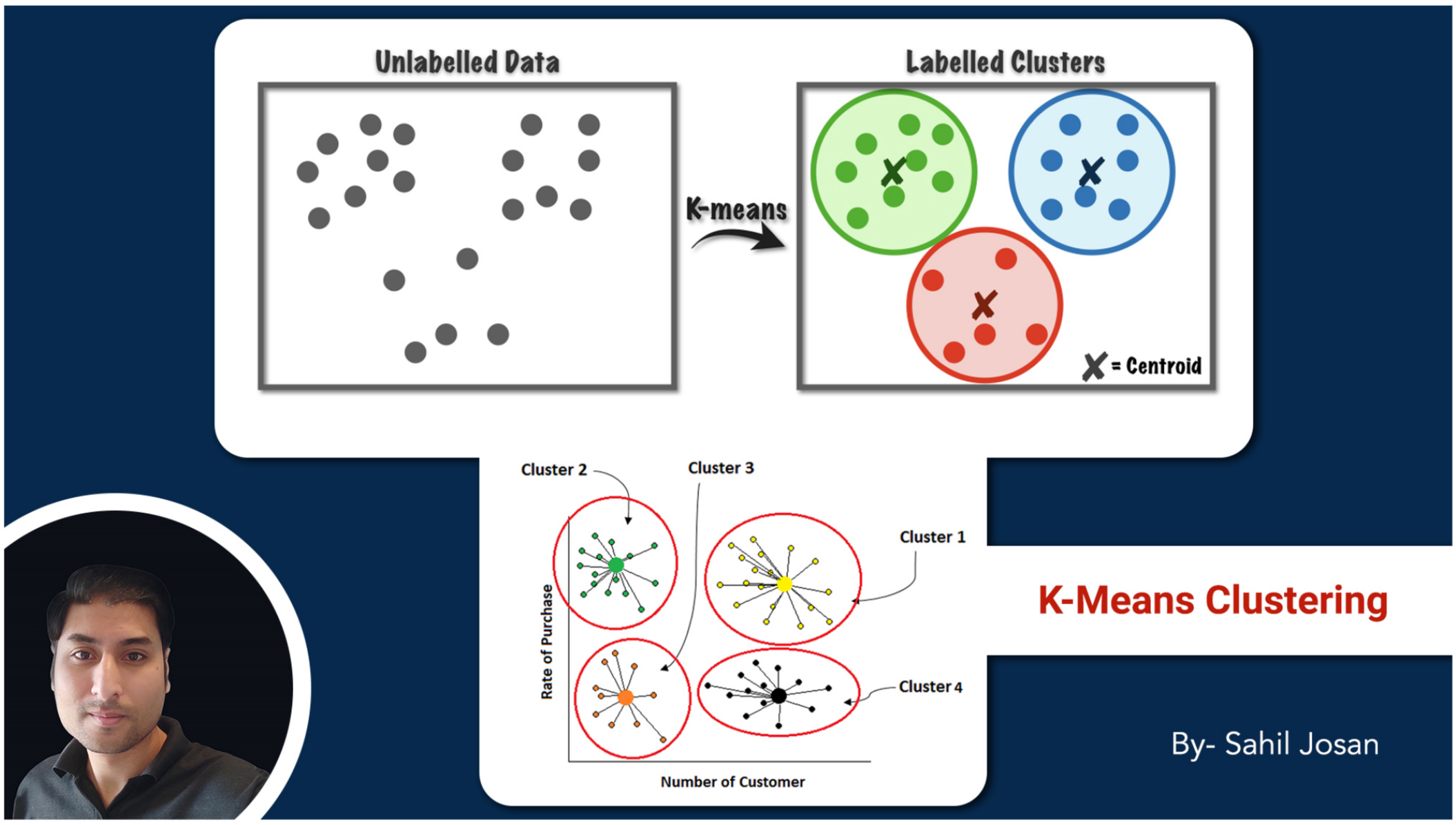
K-Means Clustering

Practical Implementation



In [ ]:
```
### Importing the required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')

from sklearn.cluster import KMeans
```

In [ ]:
```
data = pd.read_csv("https://raw.githubusercontent.com/Sahiljosan/Machine-Learning_Practical-I
data.head()
```

Out[ ]:

	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

In [ ]:
```
data.shape
```

Out[ ]:
```
(200, 5)
```

The dataset consists of following five features of 200 customers:

- CustomerID: Unique ID assigned to the customer

- Gender: Gender of the customer
- Age: Age of the customer
- Annual Income (k$): Annual Income of the customer
- Spending Score (1-100): Score assigned by the mall based on customer behavior and spending nature.

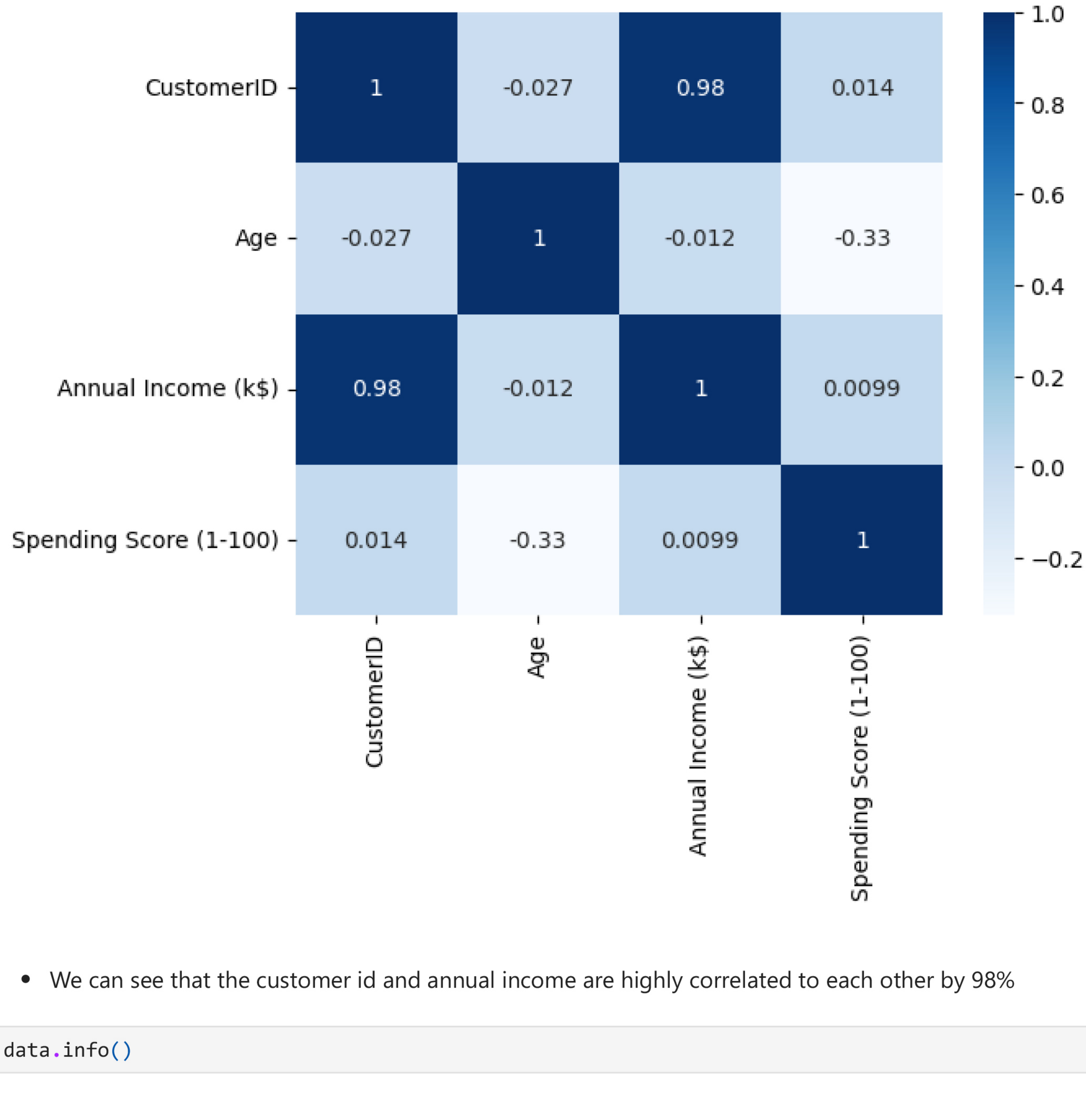
In [ ]:
```
data.corr()
```

Out[ ]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
CustomerID	1.000000	-0.026763	0.977548	0.013835
Age	-0.026763	1.000000	-0.012398	-0.327227
Annual Income (k$)	0.977548	-0.012398	1.000000	0.009903
Spending Score (1-100)	0.013835	-0.327227	0.009903	1.000000

In [ ]:
```
sns.heatmap(data.corr(),annot =True,cmap='Blues')
```

Out[ ]: <AxesSubplot:>



- We can see that the customer id and annual income are highly correlated to each other by 98%

In [ ]:
```
data.info()
```

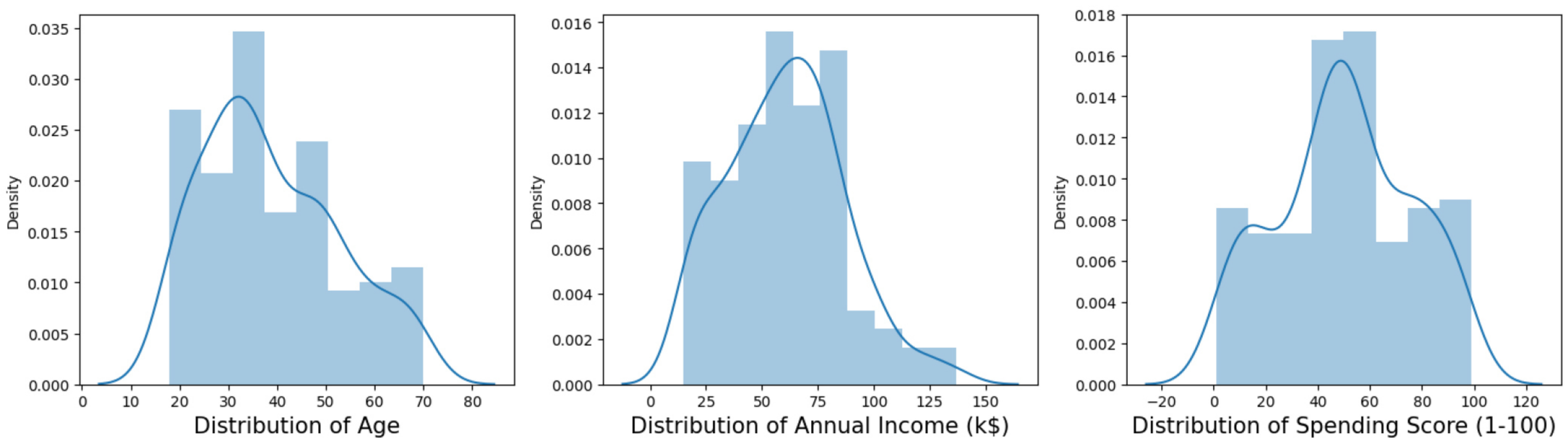
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    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
```

In [ ]:
```
numerical_col = [fea for fea in data if data[fea].dtype != "O"]
numerical_col
```

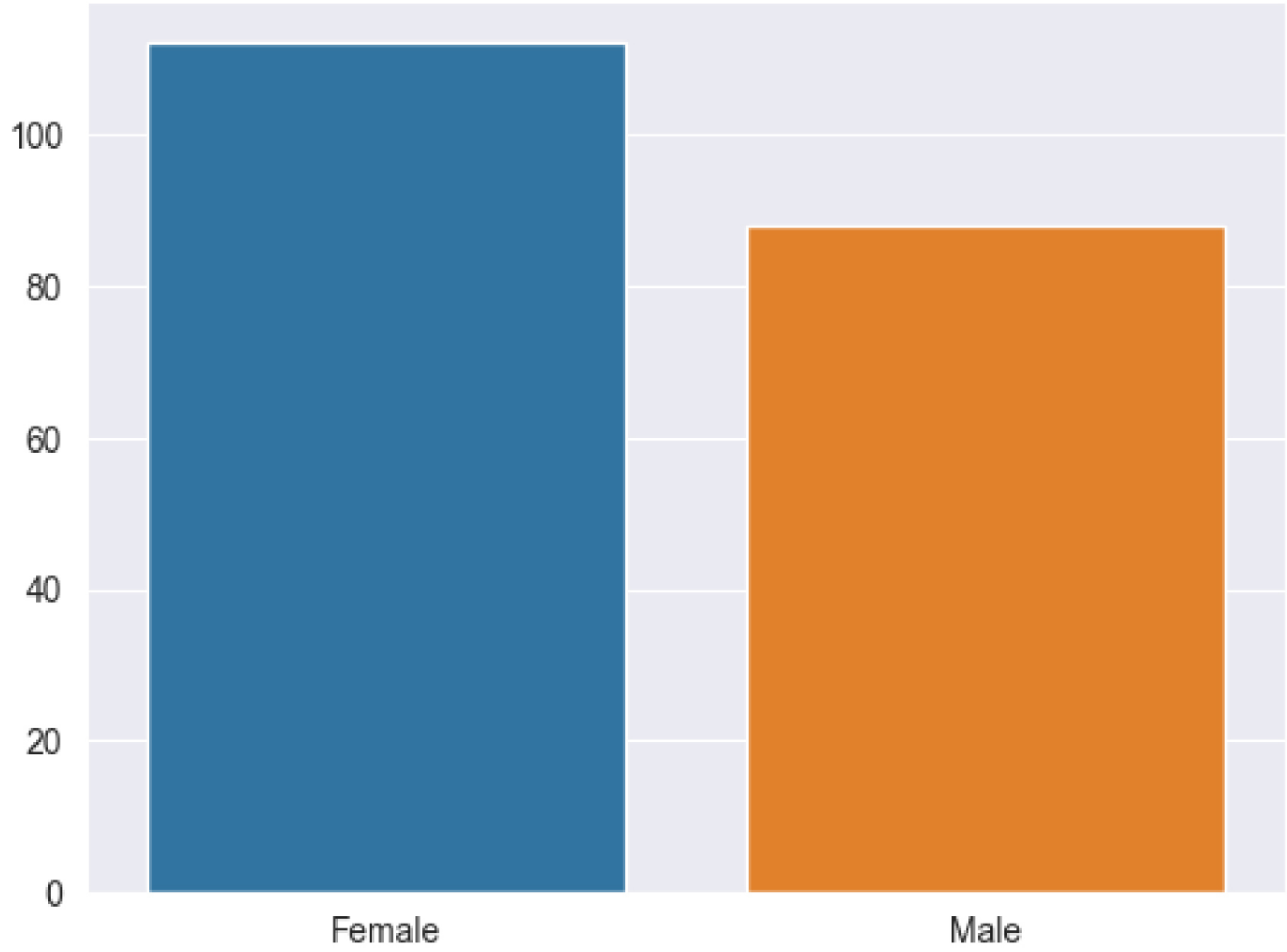
Out[ ]:
```
['CustomerID', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)']
```

Check Distribution

In [ ]:
```
feature_include = ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']
plt.figure(figsize=(15,8), facecolor='white')
# plt.suptitle('Distribution of Numerical Features',fontweight = "bold",fontsize=15,y = 1)
for i in range(0, len(feature_include)):
    plt.subplot(2, 3, i+1)
    sns.distplot(x=data[feature_include[i]],kde=True)
    plt.xlabel("Distribution of {}".format(feature_include[i]),fontsize = 15)
    plt.tight_layout()
```



In [ ]:
```
gen = data['Gender'].value_counts()
sns.set_style("darkgrid")
sns.barplot(x = gen.index,y = gen.values)
plt.show()
```



Check null values

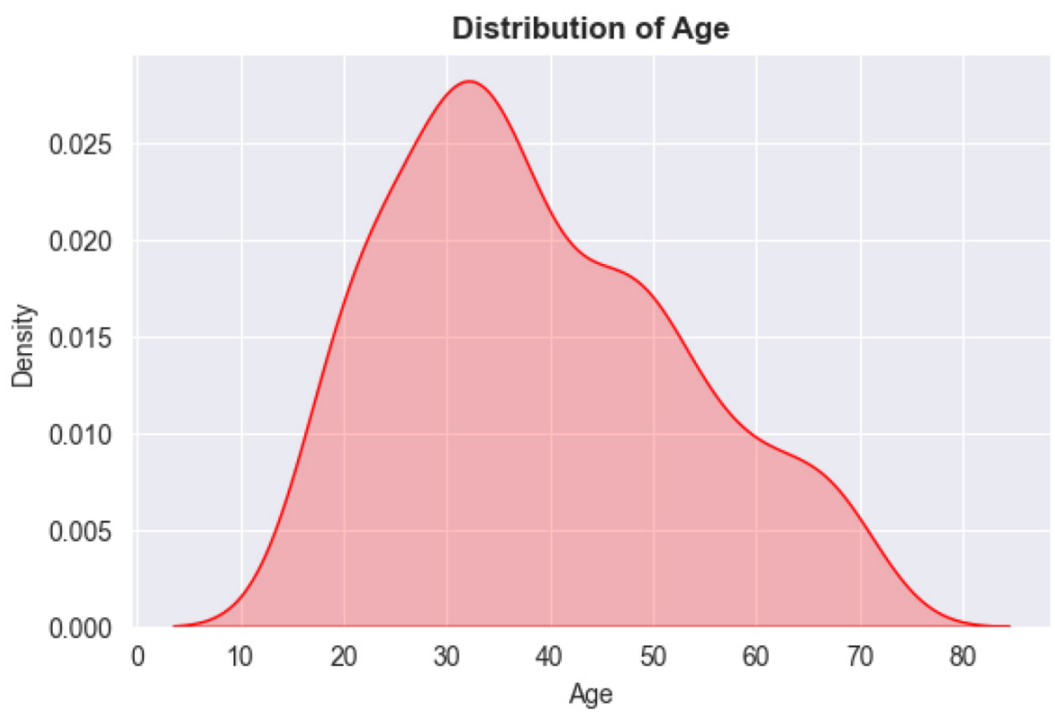
In [ ]:
```
data.isna().sum()
```

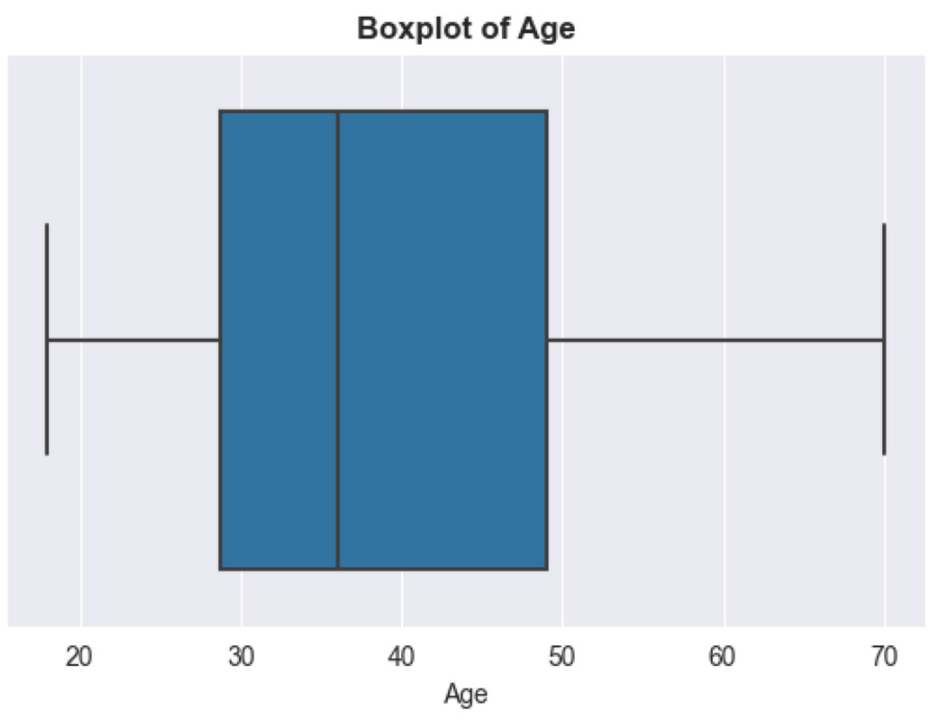
Out[ ]:
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

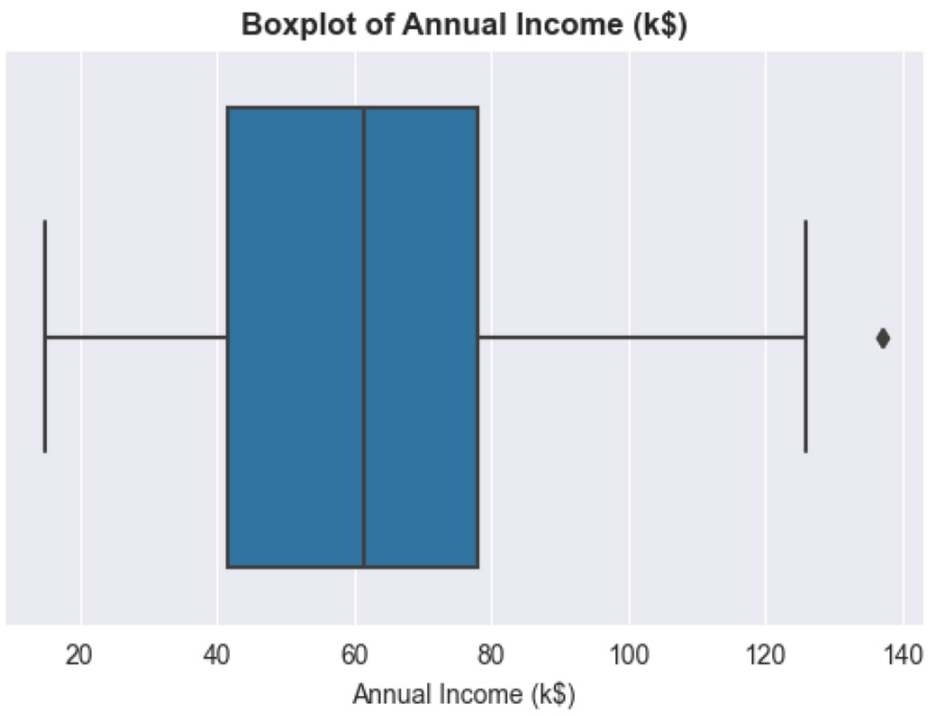
Check outlier and distribution

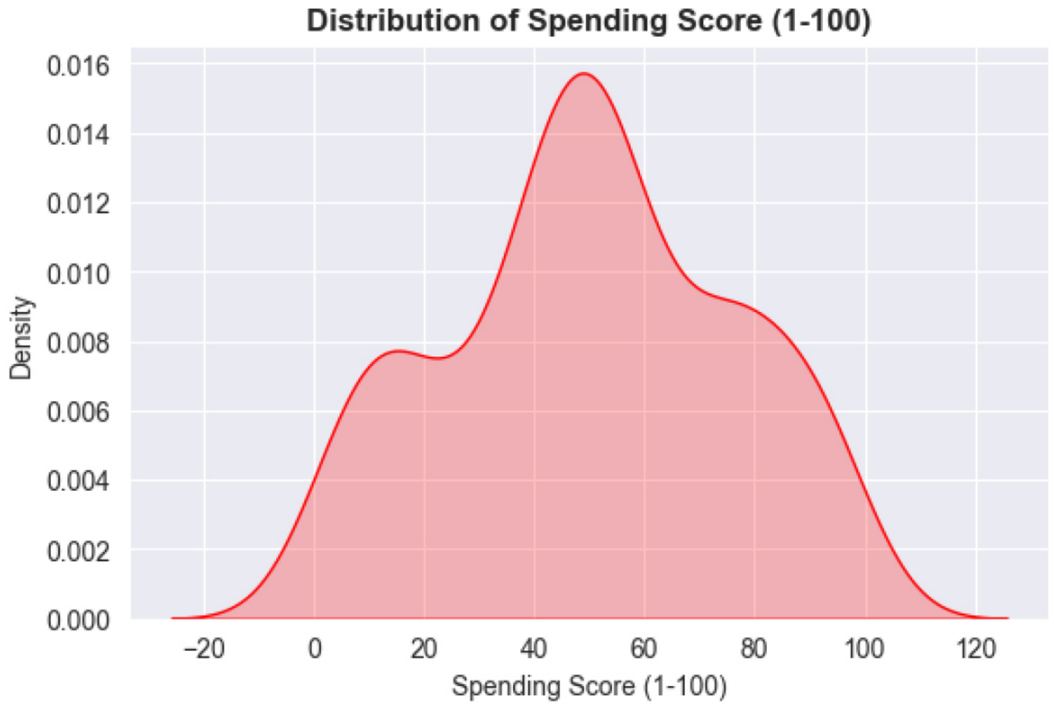
In [ ]:
```
for fea in feature_include:
    plt.figure(figsize = (14,4))
    plt.subplot(121)
    sns.kdeplot(x=data[fea],shade = True, color='r',data=data)
    plt.title("Distribution of {}".format(fea),fontweight = 'bold' )

    plt.subplot(122)
    sns.boxplot(x= fea,data = data[feature_include])
    plt.title("Boxplot of {}".format(fea),fontweight = 'bold' )
    plt.show()
```









- As we can see there are no outliers in the data
- so we can now build the model

In [ ]:
```
df = data.copy()
```

In [ ]:
```
X = df.iloc[:,3:]
```

- As we want to make clusters we do not require dependent variable
- We use 2 dimentional array

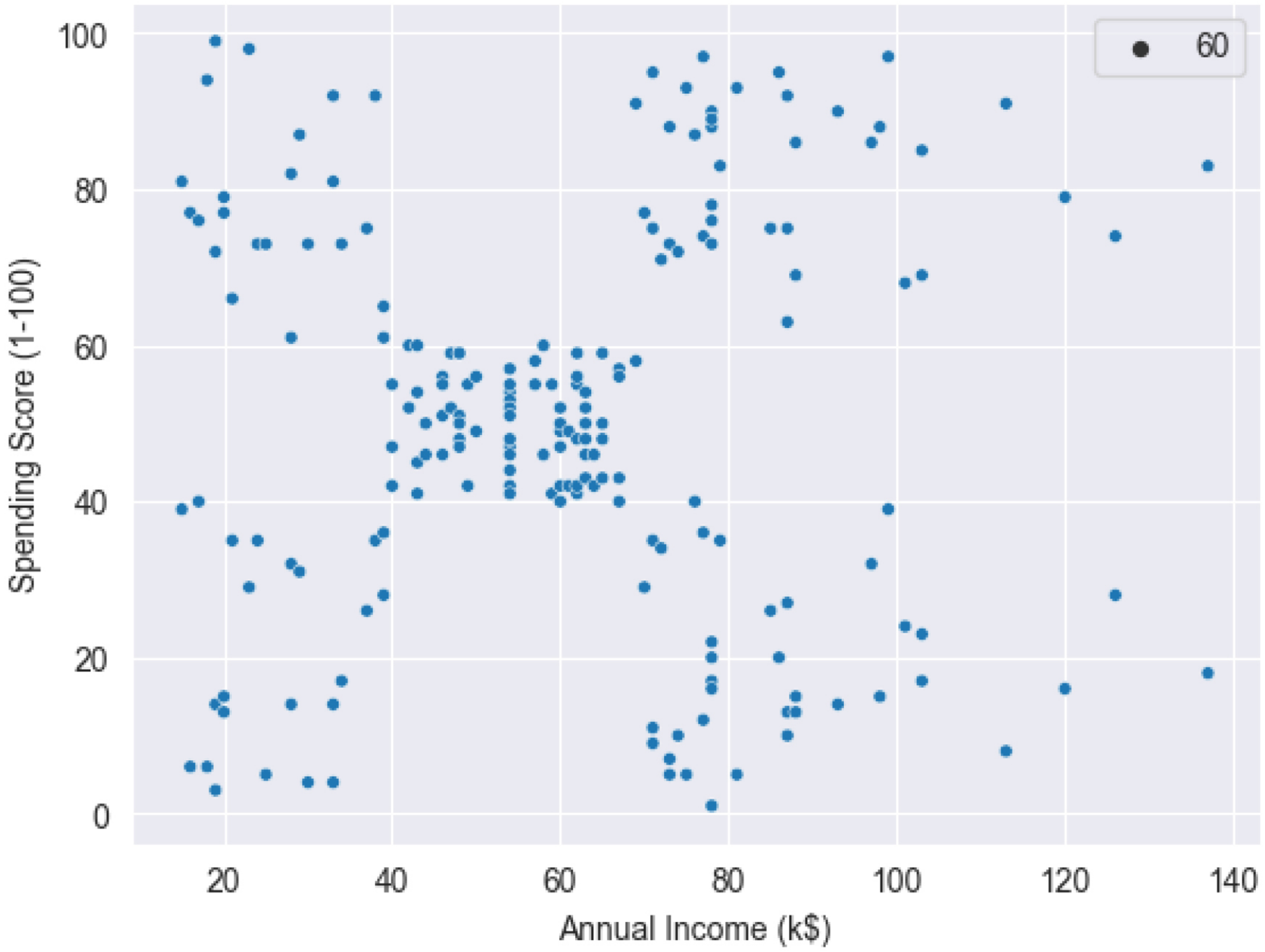
In [ ]:
```
X.head()
```

Out[ ]:

	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40

Scatterplot the input data

In [ ]:
```
sns.scatterplot(data = X,x = 'Annual Income (k$)', y = 'Spending Score (1-100)',size=60)
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Spending Score (1-100) vs Annual Income (k$)')
plt.show()
```



Importing K-means from sklearn

In [ ]:
```
from sklearn.cluster import KMeans
```

Reference

WCSS: Within Cluster sum of squares

In [ ]:
```
# calculate withing cluster sum of sqaure for different values of k
wcss = []
for i in range(1,11):
    km = KMeans(n_clusters=i)
    km.fit(X)
    wcss.append(km.inertia_)
```

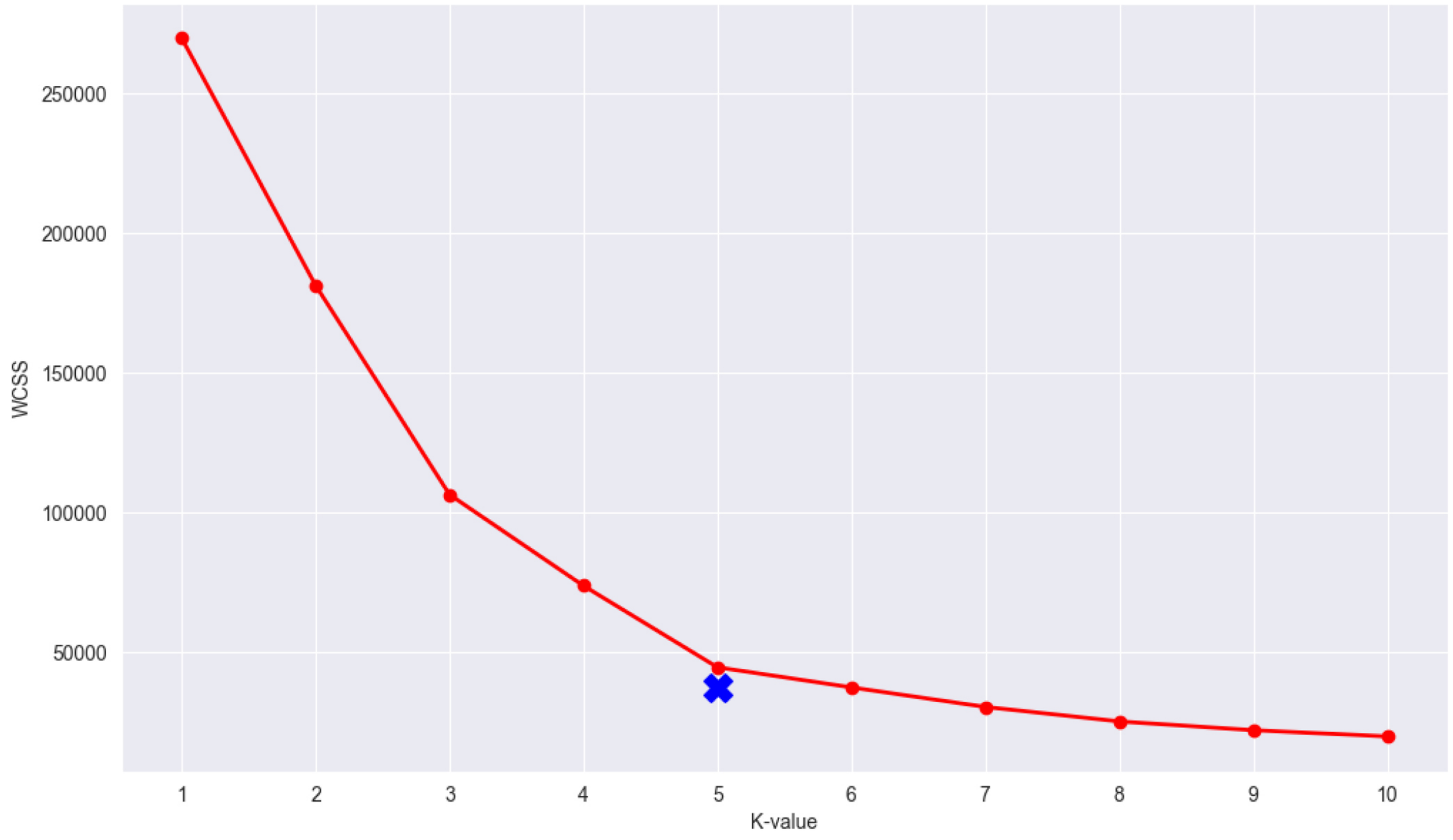
In [ ]:
```
wcss
```

Out[ ]:
```
[269981.28,
 181363.595959596,
 106348.3730621112,
 73679.78903948834,
 44448.45544793371,
 37239.83554245604,
 30227.606513152015,
 25043.89004329004,
 21898.331946797163,
 19657.783608703958]
```

Plot the elbow curve

In [ ]:
```
plt.figure(figsize=(12,7))
plt.plot(range(1,11),wcss,linewidth=2,color= 'red',marker = "o")
plt.scatter(5, wcss[5], marker="X", s=200, c="blue")
plt.xlabel("K-value")
```

```
plt.xticks(np.arange(1,11,1))
plt.ylabel("WCSS")
plt.show()
```



Observations:

- We can see that after 5 the line is smooth so that means k = 5 will be fine

In [ ]:
```
# Taking 5 clusters
km1= KMeans(n_clusters=5)
# Fitting the input data
km1.fit(X)
# Predicting the labels of the input data
y_pred = km1.predict(X)
```

In [ ]:
```
# These are our 5 clusters
y_pred
```

Out[ ]:
```
array([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, 1,
       2, 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, 0, 4, 0, 1, 0, 4, 0, 4, 0,
       1, 0, 4, 0, 4, 0, 4, 0, 4, 0, 1, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0,
       4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0,
       4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0,
       4, 0])
```

In [ ]:
```
# Add the cluster column to our main data
df['clusters'] = y_pred
# the new dataframe with the clustering is done
df
```

Out[ ]:

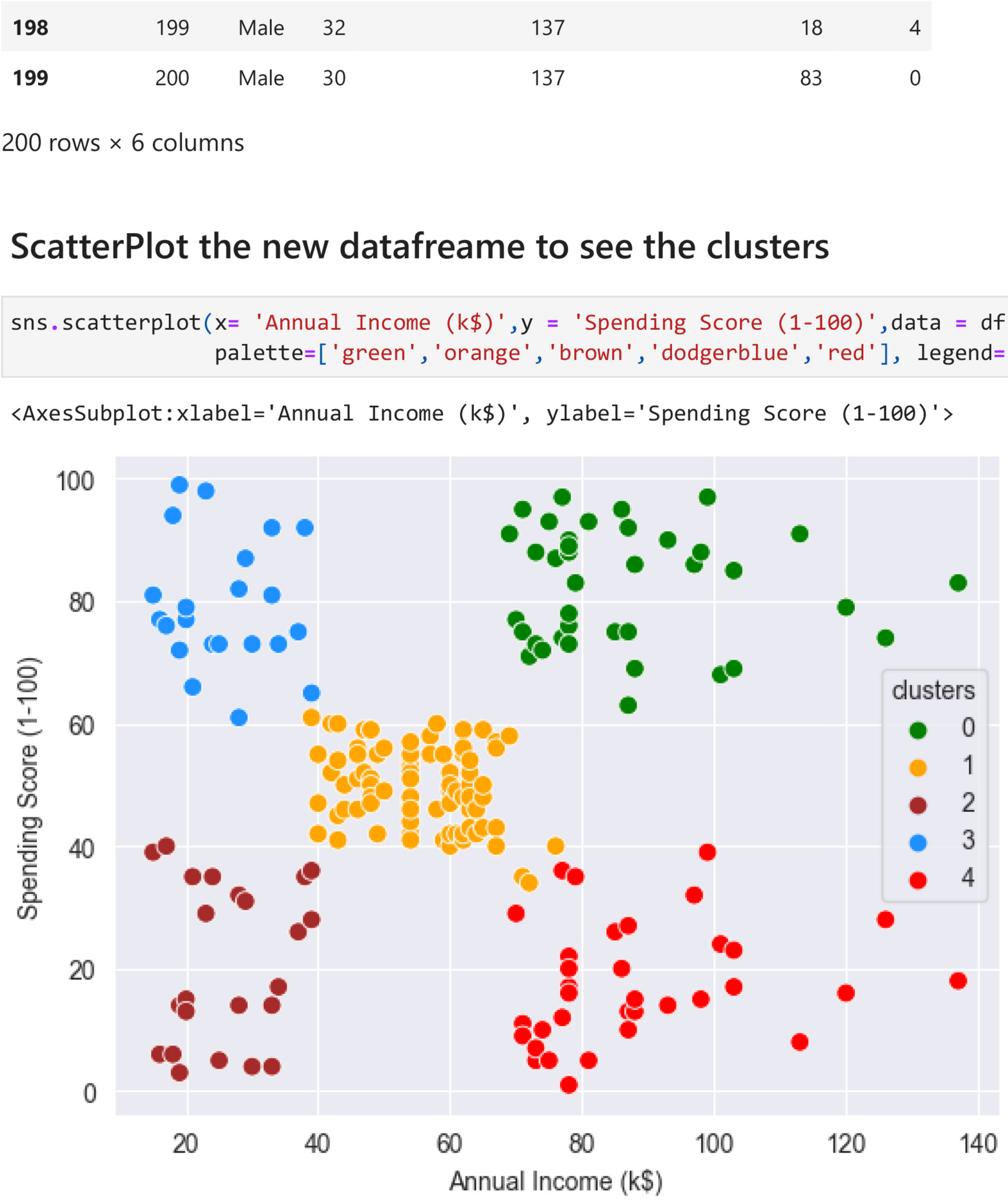
	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	clusters
0	1	Male	19	15	39	2
1	2	Male	21	15	81	3
2	3	Female	20	16	6	2
3	4	Female	23	16	77	3
4	5	Female	31	17	40	2
...	...	...	...	...	...	...
195	196	Female	35	120	79	0
196	197	Female	45	126	28	4
197	198	Male	32	126	74	0
198	199	Male	32	137	18	4
199	200	Male	30	137	83	0

200 rows × 6 columns

ScatterPlot the new datafreame to see the clusters

In [ ]:
```
sns.scatterplot(x= 'Annual Income (k$)',y = 'Spending Score (1-100)',data = df,hue = 'cluster
                palette=['green','orange','brown','dodgerblue','red'], legend='full',s = 60)
```

Out[ ]:
```
<AxesSubplot:xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'>
```



- This clusering in done by K-means clustering algoritm using 2D array (2 columns)

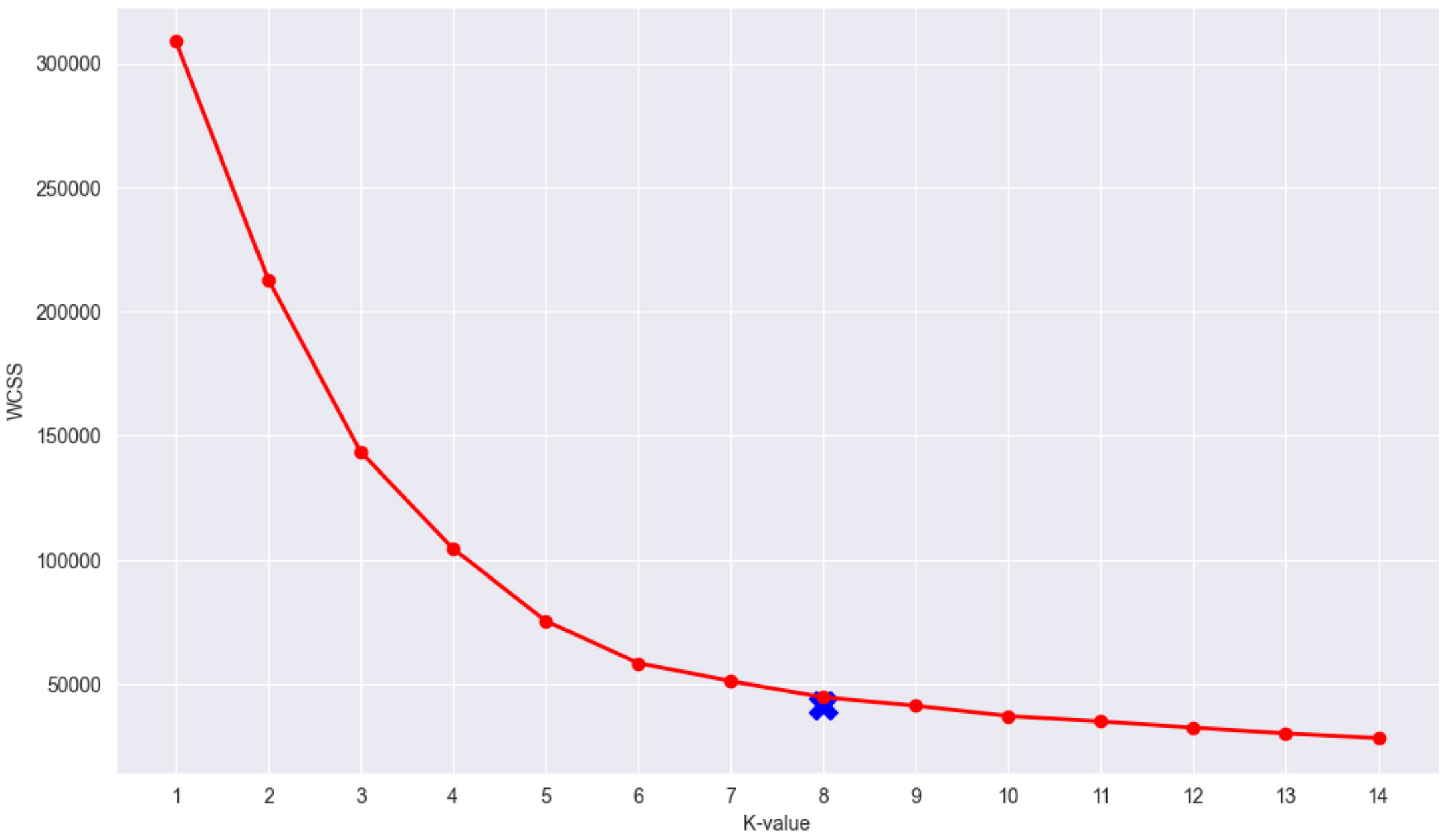
Do clustering using 3 Dimenstions (3 columns)

In [ ]:
```
#Taking the features
X2=df[["Age","Annual Income (k$)","Spending Score (1-100)"]]
```

```
# calculate withing cluster sum of sqaure for different values of k
wcss2 = []
for k in range(1,15):
    kmeans2 = KMeans(n_clusters=k,init="k-means++")
    kmeans2.fit(X2)
    wcss2.append(kmeans2.inertia_)
print(wcss2)

# plot the elbow curve
plt.figure(figsize=(12,7))
plt.plot(range(1,15),wcss2,linewidth=2,color= 'red',marker = "o")
plt.scatter(8, wcss2[8], marker="X", s=200, c="blue")
plt.xlabel("K-value")
plt.xticks(np.arange(1,15,1))
plt.ylabel("WCSS")
plt.show()
```

```
[308812.77999999997, 212840.16982097185, 143342.751571706, 104366.151455562, 75350.7791724877
6, 58300.4433215907, 51096.62442155546, 44640.0280485304, 41201.97470891177, 37071.8052855961
9, 34879.63441026809, 32343.037788600286, 30021.711420034215, 28123.651916262803]
```



- Here we take the value of k = 8
- Because after 8 wcss starts to diminish

In [ ]:
```
km2 = KMeans(n_clusters=8, init="k-means++")
y2 = km2.fit_predict(X2)
y2
```

Out[ ]:
```
array([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, 5, 3, 2, 6,
       2, 3, 5, 6, 6, 6, 5, 6, 6, 5, 5, 5, 5, 5, 6, 5, 5, 6, 5, 5, 5, 6,
       5, 5, 6, 6, 5, 5, 5, 5, 5, 6, 5, 6, 6, 5, 5, 6, 5, 5, 6, 5, 5, 6,
       6, 5, 5, 6, 5, 6, 6, 6, 5, 6, 5, 6, 6, 5, 5, 6, 5, 6, 5, 5, 5, 5,
       5, 6, 6, 6, 6, 6, 5, 5, 5, 5, 6, 6, 6, 1, 6, 1, 0, 1, 0, 1, 0, 1,
       6, 1, 0, 1, 0, 1, 0, 1, 0, 1, 6, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 4, 7, 4, 7, 4, 7, 4, 7, 4, 7, 4, 7, 4, 7, 4, 7, 4, 7,
       4, 7])
```

- From 0 to 7 these are our 8 clusters

In [ ]:
```
# Add these custers to our dataset
# remove old custers
df.drop('clusters',axis = 1,inplace=True)
# add new custers
df['clusters'] = y2
df
```

Out[ ]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	clusters
0	1	Male	19	15	39	2
1	2	Male	21	15	81	3
2	3	Female	20	16	6	2
3	4	Female	23	16	77	3
4	5	Female	31	17	40	2
...	...	...	...	...	...	...
195	196	Female	35	120	79	7
196	197	Female	45	126	28	4
197	198	Male	32	126	74	7
198	199	Male	32	137	18	4
199	200	Male	30	137	83	7

200 rows × 6 columns

In [ ]:
```
df['clusters'].unique()
```

Out[ ]:
```
array([2, 3, 5, 6, 1, 0, 4, 7])
```

3D Plot the clustering

- As we have input 3 Dimentional array (3 columns) now we have to 3D plot to see our clusters

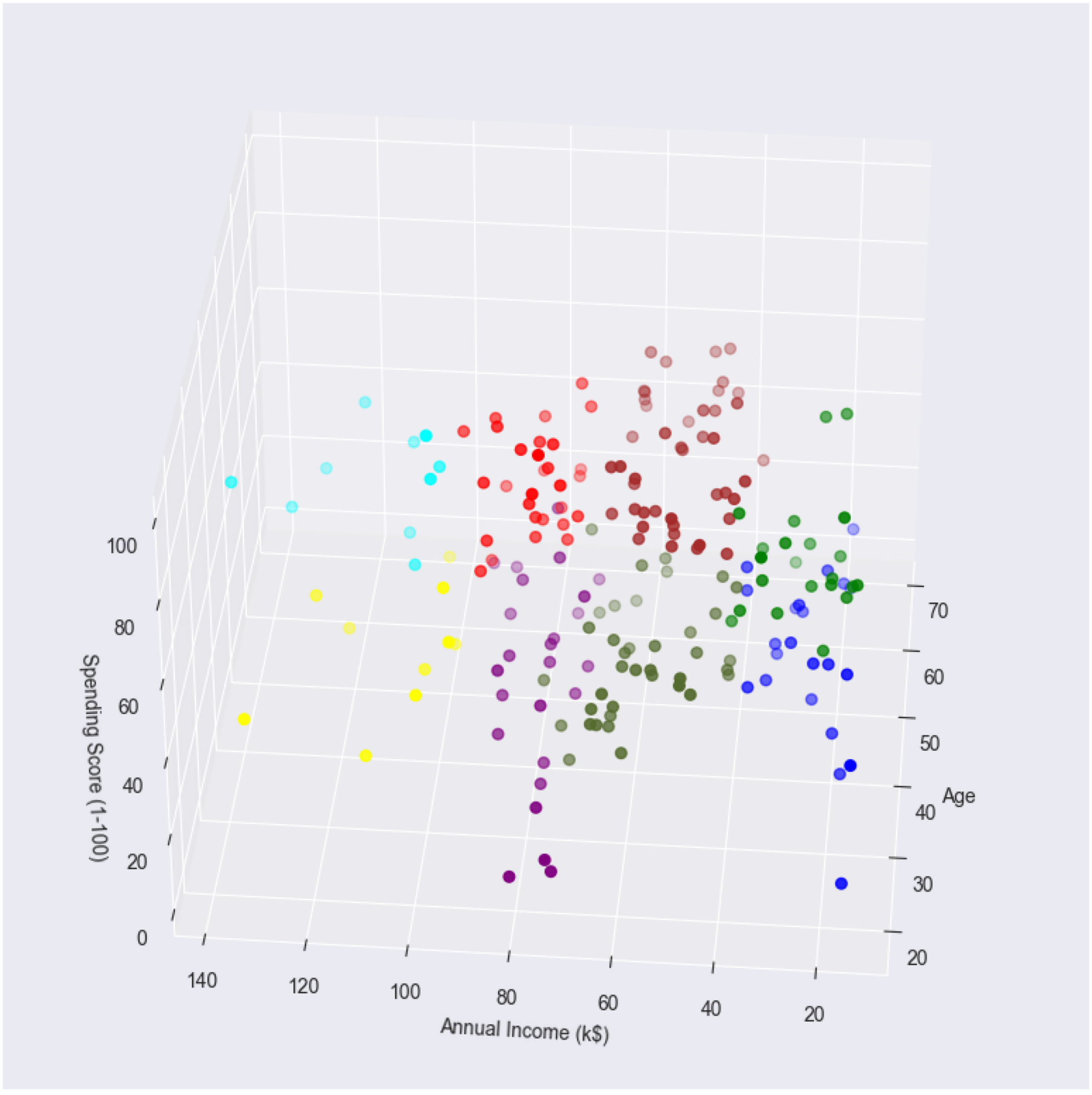
In [ ]:
```
fig = plt.figure(figsize=(10,10))

ax = fig.add_subplot(111,projection = '3d')

ax.scatter(df.Age[df.clusters == 0],df['Annual Income (k$)'][df['clusters'] == 0],df['Spendin
ax.scatter(df.Age[df.clusters == 1],df['Annual Income (k$)'][df['clusters'] == 1],df['Spendin
ax.scatter(df.Age[df.clusters == 2],df['Annual Income (k$)'][df['clusters'] == 2],df['Spendin
ax.scatter(df.Age[df.clusters == 3],df['Annual Income (k$)'][df['clusters'] == 3],df['Spendin
ax.scatter(df.Age[df.clusters == 4],df['Annual Income (k$)'][df['clusters'] == 4],df['Spendin
ax.scatter(df.Age[df.clusters == 5],df['Annual Income (k$)'][df['clusters'] == 5],df['Spendin
ax.scatter(df.Age[df.clusters == 6],df['Annual Income (k$)'][df['clusters'] == 6],df['Spendin
ax.scatter(df.Age[df.clusters == 7],df['Annual Income (k$)'][df['clusters'] == 7],df['Spendin
ax.view_init(35,185)
plt.xlabel("Age")
plt.ylabel("Annual Income (k$)")
ax.set_zlabel('Spending Score (1-100)')
plt.show()
```

3D Plot the clustering

- As we have input 3 Dimentional array (3 columns) now we have to 3D plot to see our clusters



In [ ]:
```
for i in range(0,len(df['clusters'].unique())):
    cust = df[df['clusters'] == i]
    print(f"Number of customer in group {i} = {len(cust)}")
    print(f"They are - {cust['CustomerID'].values}")
    print("-"*100)
```

In [ ]:
```
for i in range(0,len(df['clusters'].unique())):
    cust = df[df['clusters'] == i]
    print(f"Number of customer in group {i} = {len(cust)}")
    print(f"They are - {cust['CustomerID'].values}")
    print("-"*100)
```

```
Number of customer in group 0 = 10
They are - [181 183 185 187 189 191 193 195 197 199]
----------------------------------------------------------------------------------------------------

Number of customer in group 1 = 22
They are - [ 2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 46]
----------------------------------------------------------------------------------------------------

Number of customer in group 2 = 37
They are - [ 44  48  49  50  52  53  59  62  66  69  70  76  78  79  82  85  88  89
  92  94  95  96  98 100 101 104 106 112 113 114 115 116 121 122 123 133
 143]
----------------------------------------------------------------------------------------------------

Number of customer in group 3 = 29
They are - [124 126 128 130 132 134 136 138 140 142 144 146 148 150 152 154 156 158
 160 162 164 166 168 170 172 174 176 178 180]
----------------------------------------------------------------------------------------------------

Number of customer in group 4 = 22
They are - [ 1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 43 45]
----------------------------------------------------------------------------------------------------

Number of customer in group 5 = 10
They are - [182 184 186 188 190 192 194 196 198 200]
----------------------------------------------------------------------------------------------------

Number of customer in group 6 = 44
They are - [ 41  47  51  54  55  56  57  58  60  61  63  64  65  67  68  71  72  73
  74  75  77  80  81  83  84  86  87  90  91  93  97  99 102 103 105 107
 108 109 110 111 117 118 119 120]
----------------------------------------------------------------------------------------------------

Number of customer in group 7 = 26
They are - [125 127 129 131 135 137 139 141 145 147 149 151 153 155 157 159 161 163
 165 167 169 171 173 175 177 179]
----------------------------------------------------------------------------------------------------
```