In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [11]:

```
df = pd.read_csv(r'C:\Users\Shree\Desktop\customer-segmentation-dataset\Mall_Customers.csv')
df
```

Out[11]:

	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
...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

In [3]:

```
df.shape
```

Out[3]:

```
(200, 5)
```

In [4]:

```
df.describe()
```

Out[4]:

	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

In [5]:

```
df.dtypes
```

Out[5]:

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

In [6]:

```
df.isnull().sum()
```

Out[6]:

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

In [12]:

```
df.drop(['CustomerID'], axis = 1 , inplace = True)
df
```

Out[12]:

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

200 rows × 4 columns

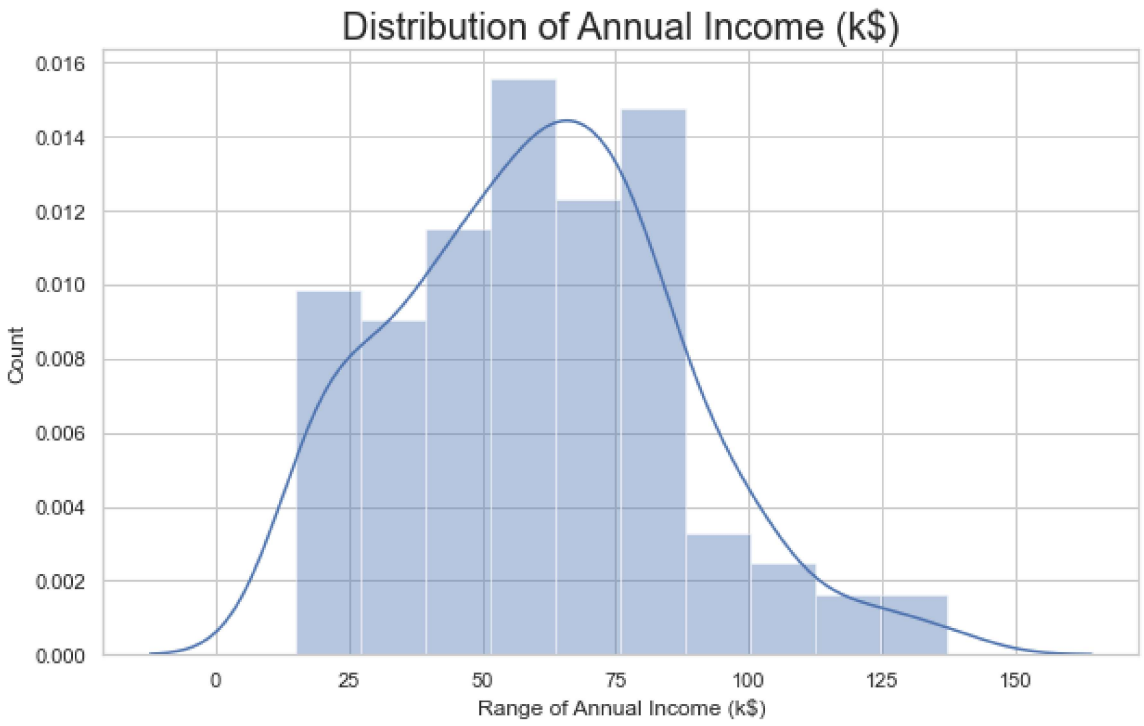
In [13]:

```
plt.figure(figsize=(10, 6))
sns.set(style = 'whitegrid')
sns.distplot(df['Annual Income (k$)'])
plt.title('Distribution of Annual Income (k$)', fontsize = 20)
plt.xlabel('Range of Annual Income (k$)')
plt.ylabel('Count')
```

```
C:\Users\Shree\anaconda3\lib\site-packages\seaborn\distributions.py:2551:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[13]:

```
Text(0, 0.5, 'Count')
```



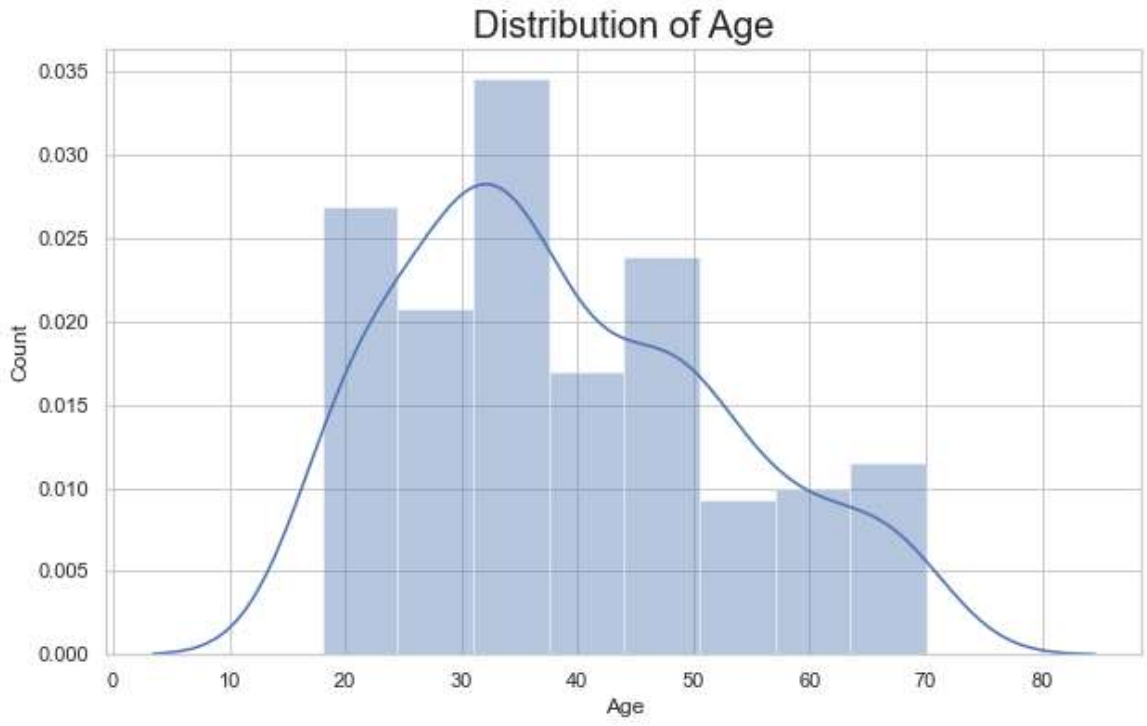
In [14]:

```
plt.figure(figsize=(10, 6))
sns.set(style = 'whitegrid')
sns.distplot(df['Age'])
plt.title('Distribution of Age', fontsize = 20)
plt.xlabel('Age')
plt.ylabel('Count')
```

```
C:\Users\Shree\anaconda3\lib\site-packages\seaborn\distributions.py:2551:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[14]:

```
Text(0, 0.5, 'Count')
```



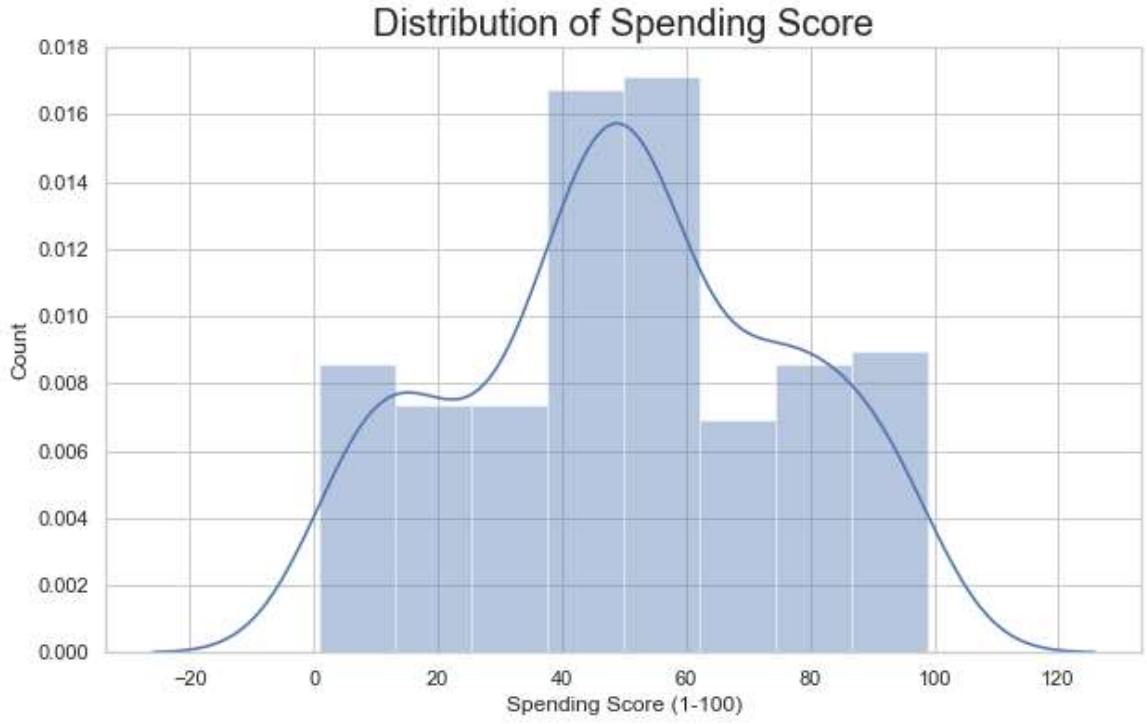
In [15]:

```
plt.figure(figsize=(10, 6))
sns.set(style = 'whitegrid')
sns.distplot(df['Spending Score (1-100)'])
plt.title('Distribution of Spending Score', fontsize = 20)
plt.xlabel('Spending Score (1-100)')
plt.ylabel('Count')
```

```
C:\Users\Shree\anaconda3\lib\site-packages\seaborn\distributions.py:2551:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

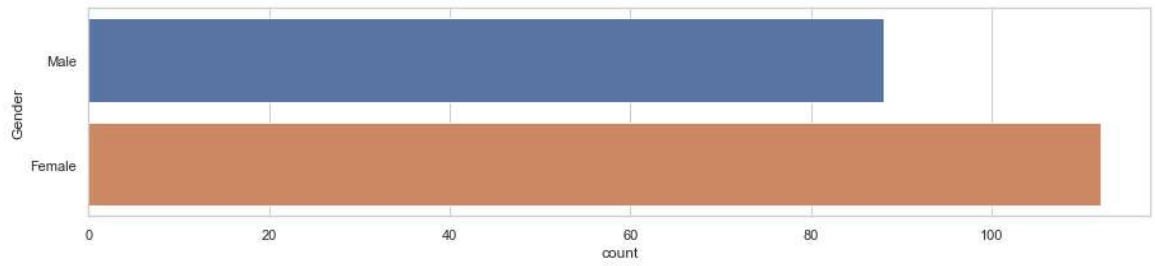
Out[15]:

```
Text(0, 0.5, 'Count')
```



In [16]:

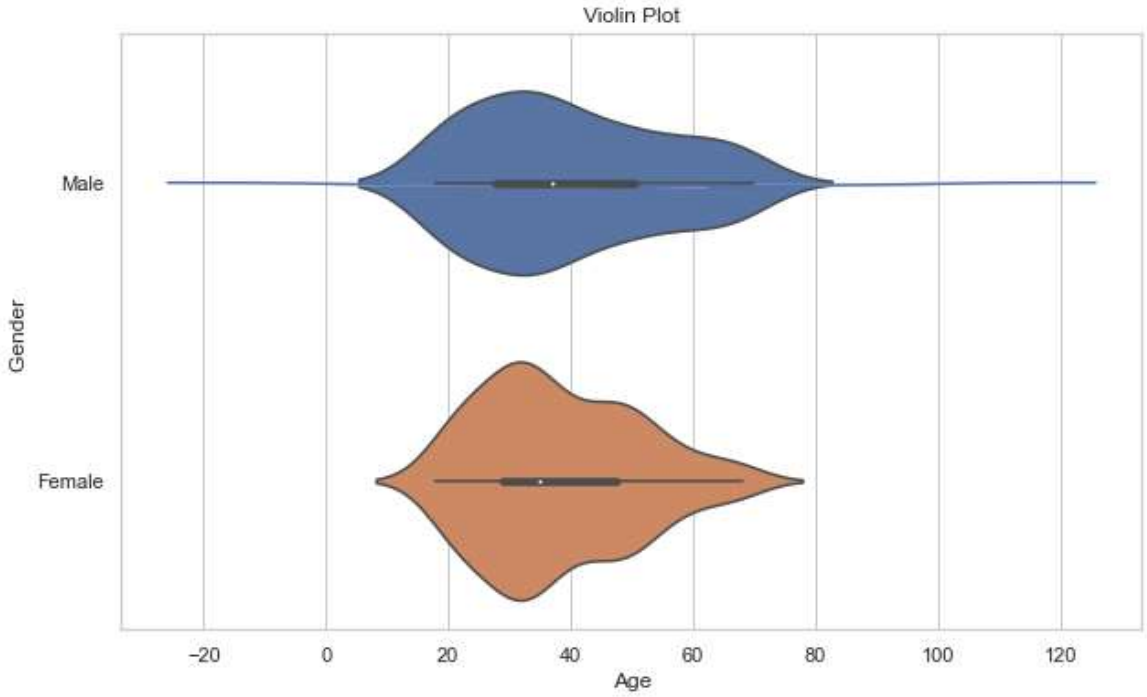
```
plt.figure(figsize = (15,3))
sns.countplot(y = 'Gender' , data = df)
plt.show()
```



In [17]:

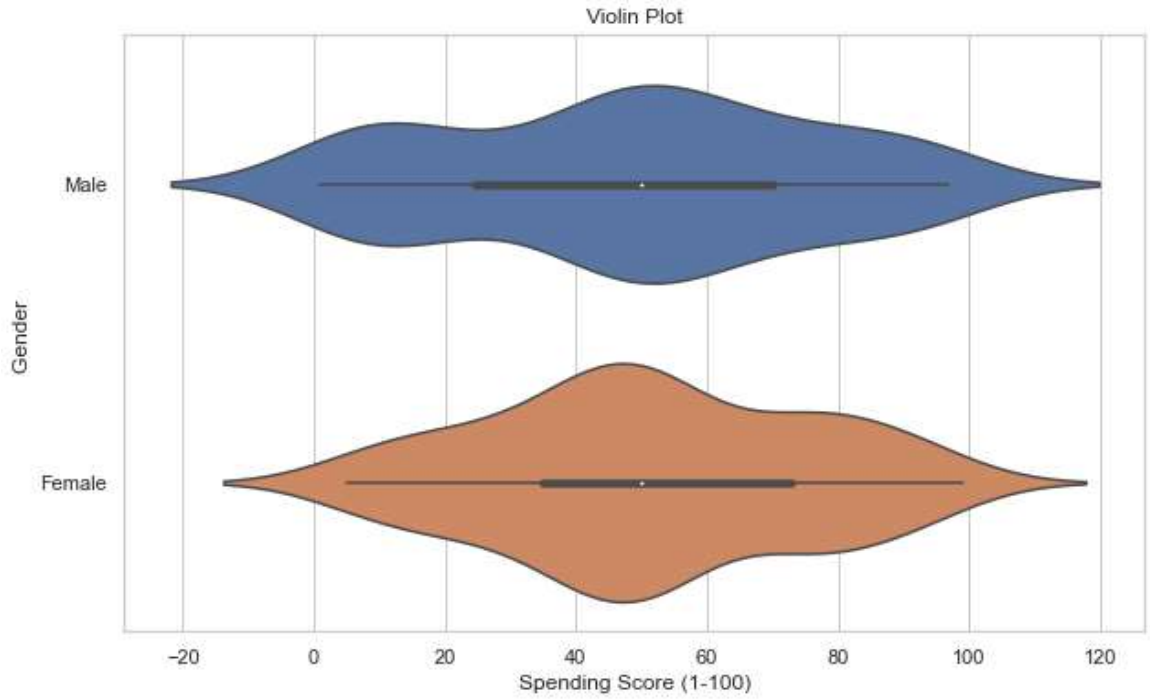
```
plt.figure(figsize=(10, 6))
sns.set(style = 'whitegrid')
sns.distplot(df['Spending Score (1-100)'])
sns.violinplot(x = 'Age'  , y = 'Gender' , data = df)
plt.title('Violin Plot')
# plt.xlabel('Spending Score (1-100)')
plt.ylabel('Gender ')
plt.show()
```

```
C:\Users\Shree\anaconda3\lib\site-packages\seaborn\distributions.py:2551:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```



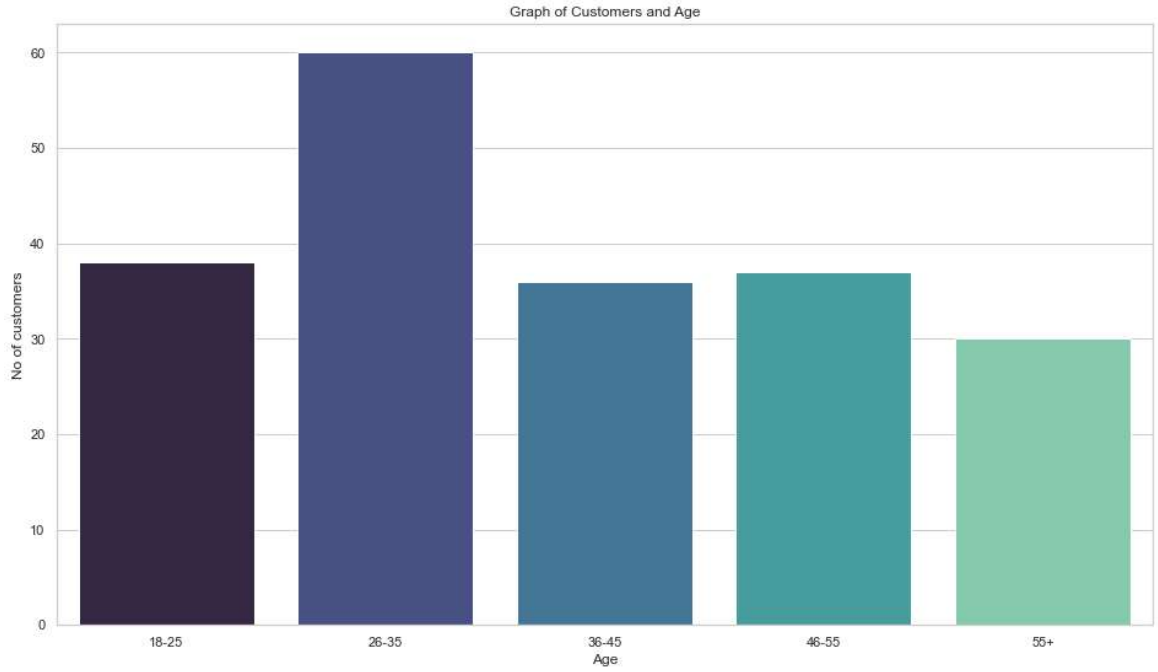
In [18]:

```
plt.figure(figsize=(10, 6))
sns.set(style = 'whitegrid')
# sns.distplot(df['Spending Score (1-100)'])
sns.violinplot(x = 'Spending Score (1-100)'  , y = 'Gender' , data = df)
plt.title('Violin Plot')
# plt.xlabel('Spending Score (1-100)')
plt.ylabel('Gender ')
plt.show()
```



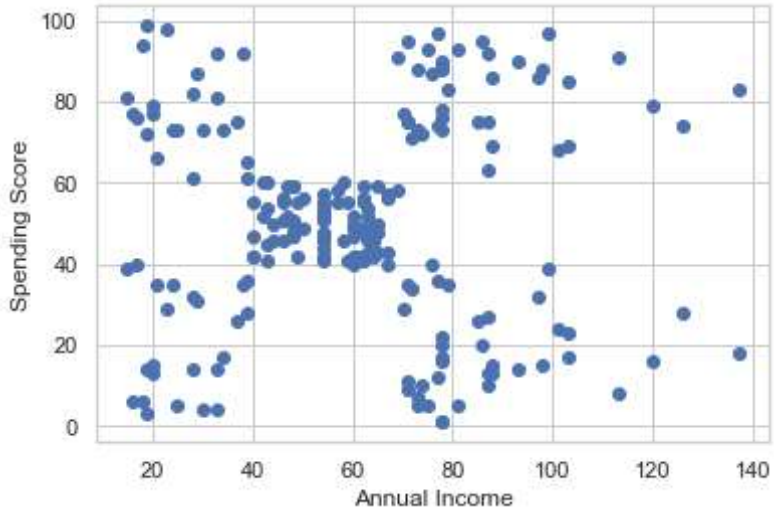
In [19]:

```
age_18to25 = df.Age[(df.Age>= 18 ) & (df.Age<= 25)]
age_26to35 = df.Age[(df.Age>= 26) & (df.Age <= 35)]
age_36to45 = df.Age[(df.Age>= 36) & (df.Age <= 45)]
age_46to55 = df.Age[(df.Age >= 46) & (df.Age<= 55)]
age_Above55 = df.Age[(df.Age>=55)]

age_x = ['18-25' , '26-35','36-45','46-55','55+']
age_y = [len(age_18to25) , len(age_26to35),len(age_36to45),len(age_46to55),len(age_Abov
e55)]

plt.figure(figsize = (16,9))
sns.barplot(x = age_x , y = age_y , palette = 'mako')
plt.title("Graph of Customers and Age")
plt.xlabel('Age')
plt.ylabel('No of customers')
plt.show()
```



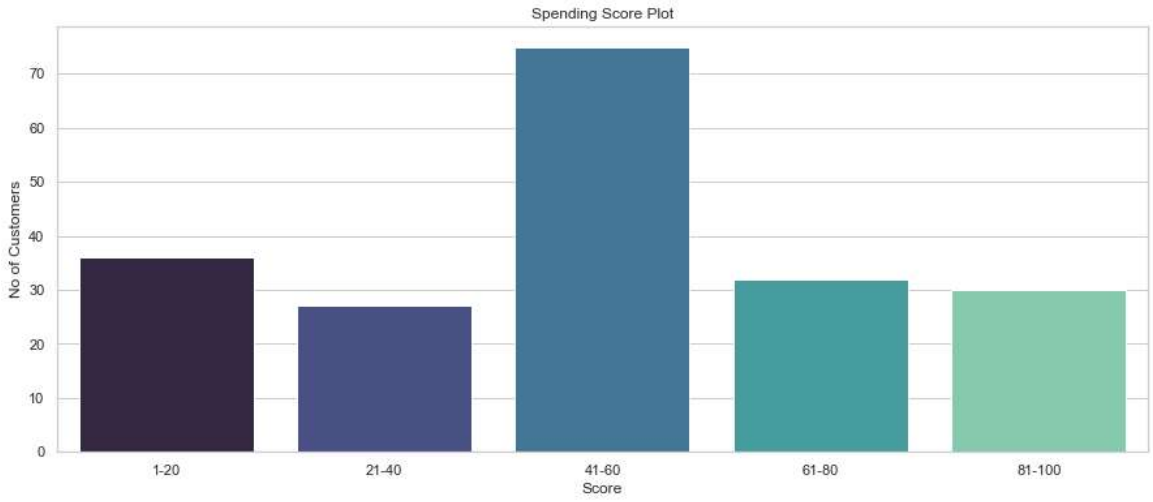
In [20]:

```
plt.scatter( df['Annual Income (k$)'] ,  df['Spending Score (1-100)'])
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.show()
```



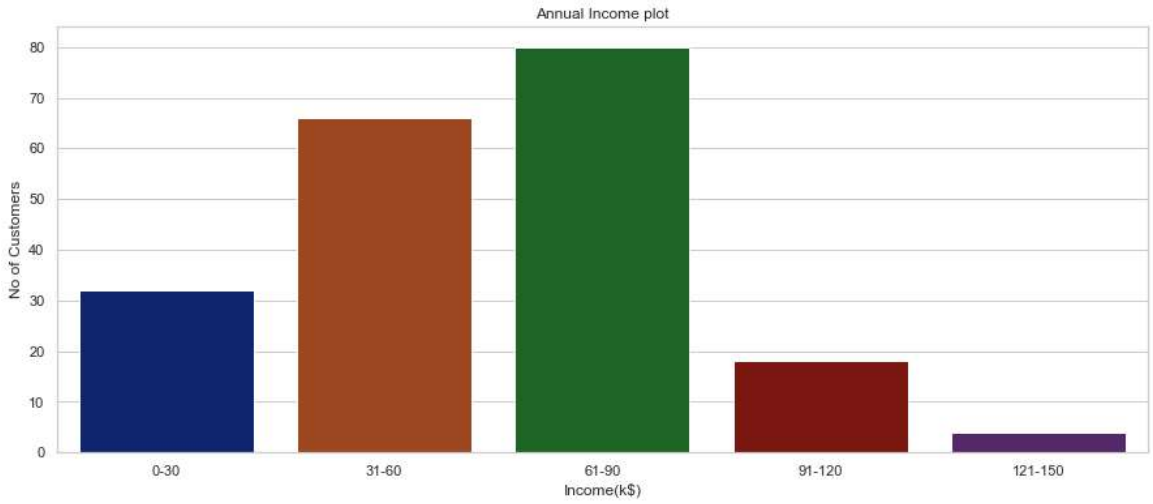
In [21]:

```
ss_1to20 = df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=1) & (df['Spendi
ng Score (1-100)']<= 20)]
ss_21to40 = df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=21) & (df['Spen
ding Score (1-100)']<= 40)]
ss_41to60 = df['Spending Score (1-100)'][(df['Spending Score (1-100)']>= 41) & (df['Spe
nding Score (1-100)']<= 60)]
ss_61to80 = df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=61) & (df['Spen
ding Score (1-100)']<= 80)]
ss_81to100 = df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=81) & (df['Spe
nding Score (1-100)']<= 99)]
# ss_above99 = df['Spending Score (1-100)'][df['Spending Score (1-100)']>=99]

ss_x = ['1-20','21-40','41-60','61-80','81-100']
ss_y = [len(ss_1to20.values),len(ss_21to40.values),len(ss_41to60.values),len(ss_61to80.
values),len(ss_81to100.values)]
plt.figure(figsize = (15,6))
sns.barplot(x = ss_x , y = ss_y , palette = 'mako')
plt.title("Spending Score Plot")
plt.xlabel('Score')
plt.ylabel('No of Customers')
plt.show()
```



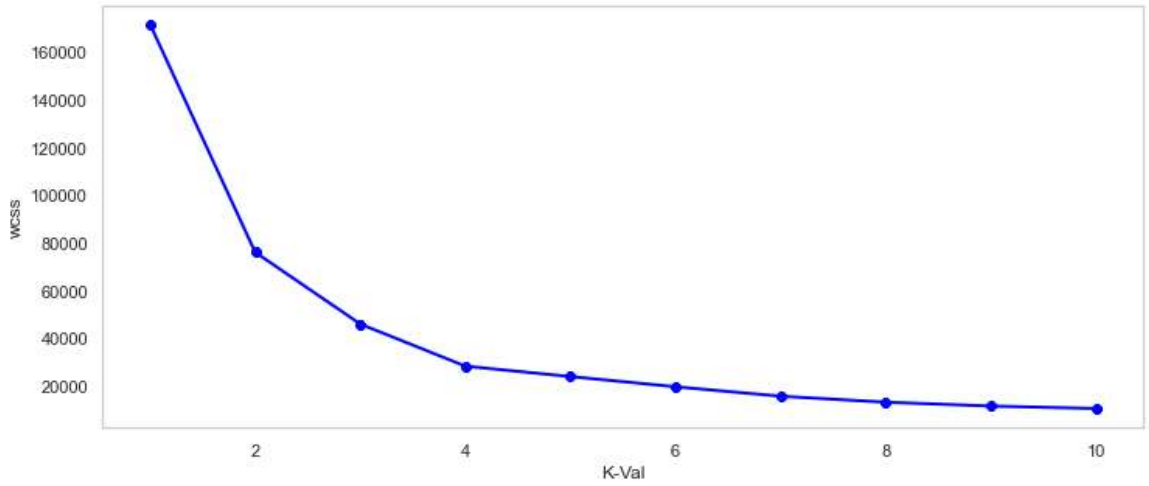
In [22]:

```
ai_0to30 = df['Annual Income (k$)'][(df['Annual Income (k$)']>=0) & (df['Annual Income
 (k$)']<= 30)]
ai_31to60 = df['Annual Income (k$)'][(df['Annual Income (k$)']>=31) & (df['Annual Incom
e (k$)']<= 60)]
ai_61to90 = df['Annual Income (k$)'][(df['Annual Income (k$)']>=61) & (df['Annual Incom
e (k$)']<= 90)]
ai_91to120 = df['Annual Income (k$)'][(df['Annual Income (k$)']>=91) & (df['Annual Inco
me (k$)']<= 120)]
ai_121to150 = df['Annual Income (k$)'][(df['Annual Income (k$)']>=121) & (df['Annual In
come (k$)']<= 150)]

ai_x = ['0-30','31-60','61-90','91-120','121-150']
ai_y = [len(ai_0to30.values),len(ai_31to60.values),len(ai_61to90.values),len(ai_91to120
.values),len(ai_121to150.values)]

plt.figure(figsize = (15,6))
sns.barplot(x = ai_x , y = ai_y , palette = 'dark')
plt.title("Annual Income plot")
plt.xlabel('Income(k$)')
plt.ylabel('No of Customers')
plt.show()
```



In [26]:

```
X1 = df.loc[:,["Age","Spending Score (1-100)"]].values
from sklearn.cluster import KMeans
wcss = []
for k in range(1,11):
    kmeans = KMeans(n_clusters = k , init = "k-means++")
    kmeans.fit(X1)
    wcss.append(kmeans.inertia_)
plt.figure(figsize = (12,5))
plt.grid()
plt.plot(range(1,11),wcss,linewidth=2,color="blue",marker="8")
plt.xlabel("K-Val")
plt.ylabel("wcss")
plt.show()
```



In [27]:

```
km = KMeans(n_clusters = 5)
label = km.fit_predict(X1)
label
```

Out[27]:

```
array([3, 1, 0, 1, 3, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 4, 3, 4, 1, 4, 1,
       0, 1, 0, 1, 4, 3, 4, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 2, 1, 4, 3,
       4, 3, 2, 3, 3, 3, 2, 3, 3, 2, 4, 4, 2, 2, 3, 2, 2, 3, 2, 2, 2, 3,
       4, 2, 3, 3, 2, 4, 2, 2, 2, 3, 4, 4, 3, 4, 2, 3, 2, 4, 3, 4, 2, 3,
       3, 4, 2, 3, 4, 4, 3, 3, 4, 3, 4, 3, 3, 4, 2, 3, 2, 3, 2, 2, 2, 2,
       2, 3, 4, 3, 3, 3, 2, 2, 4, 2, 3, 4, 3, 1, 3, 1, 4, 1, 0, 1, 0, 1,
       3, 1, 0, 1, 0, 1, 0, 1, 0, 1, 3, 1, 0, 1, 4, 1, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 0, 1, 4, 1, 0, 1, 4, 1, 0, 1, 4, 3, 0, 1, 0, 1, 0, 1,
       0, 1, 0, 1, 4, 1, 0, 1, 4, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 4, 1,
       0, 1])
```

In [28]:

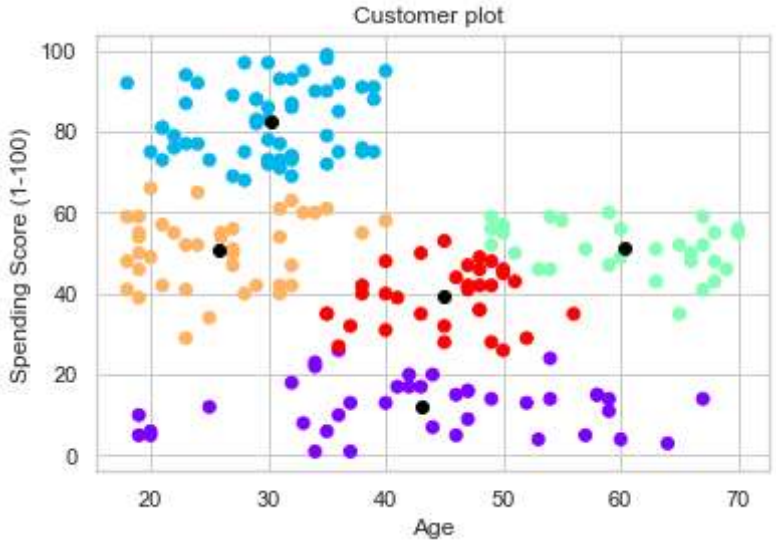
```
#centroids:
centroids = km.cluster_centers_
centroids
```

Out[28]:

```
array([[43.1       , 12.2       ],
       [30.1754386 , 82.35087719],
       [60.36666667, 51.16666667],
       [25.775     , 50.775     ],
       [44.96969697, 39.15151515]])
```

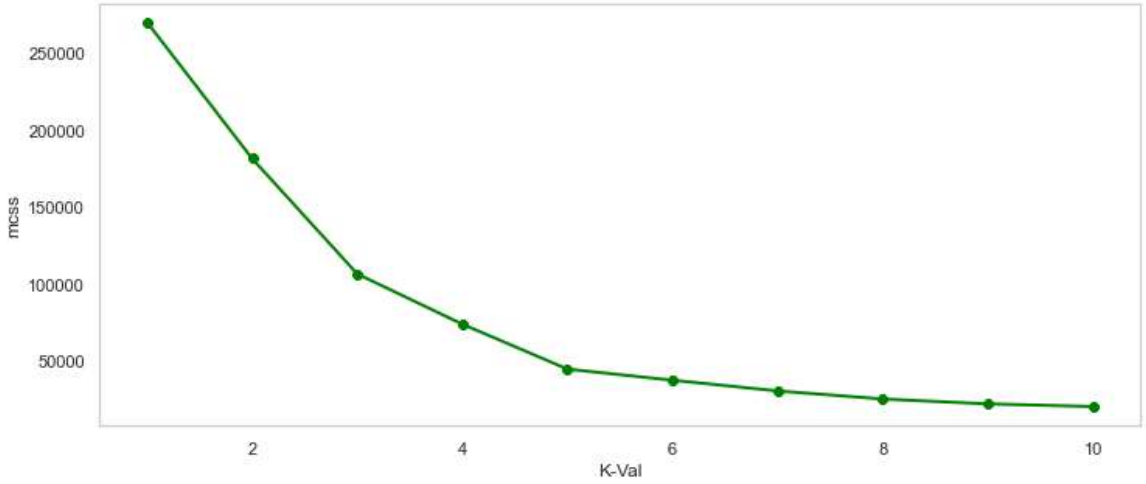
In [31]:

```
plt.scatter(X1[:,0],X1[:,1] , c = km.labels_ , cmap = 'rainbow')
plt.scatter(centroids[:,0] , centroids[:,1] , color = 'black')
plt.title('Customer plot ')
plt.xlabel('Age')
plt.ylabel('Spending Score (1-100)')
plt.show()
```



In [35]:

```
X2 = df.loc[:,["Annual Income (k$)","Spending Score (1-100)"]].values
from sklearn.cluster import KMeans
mcss = []
for k in range(1,11):
    kmeans = KMeans(n_clusters = k , init = "k-means++")
    kmeans.fit(X2)
    mcss.append(kmeans.inertia_)
plt.figure(figsize = (12,5))
plt.grid()
plt.plot(range(1,11),mcss,linewidth=2,color="green",marker="8")
plt.xlabel("K-Val")
plt.ylabel("mcss")
plt.show()
```



In [38]:

```
km_1 = KMeans(n_clusters = 5)
label = km_1.fit_predict(X2)
label
```

Out[38]:

```
array([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, 3,
       1, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 0, 2, 3, 2, 0, 2, 0, 2,
       3, 2, 0, 2, 0, 2, 0, 2, 0, 2, 3, 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])
```

In [41]:

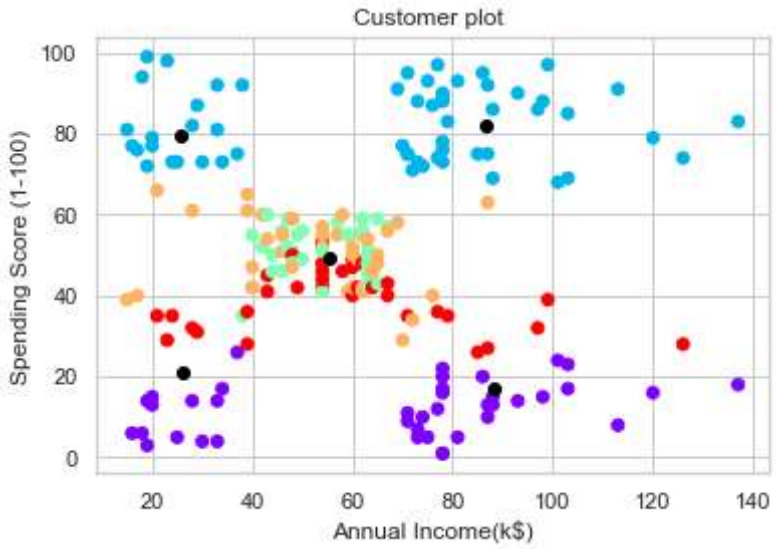
```
centroids_1 = km_1.cluster_centers_
centroids_1
```

Out[41]:

```
array([[88.2       , 17.11428571],
       [26.30434783, 20.91304348],
       [86.53846154, 82.12820513],
       [55.2962963 , 49.51851852],
       [25.72727273, 79.36363636]])
```

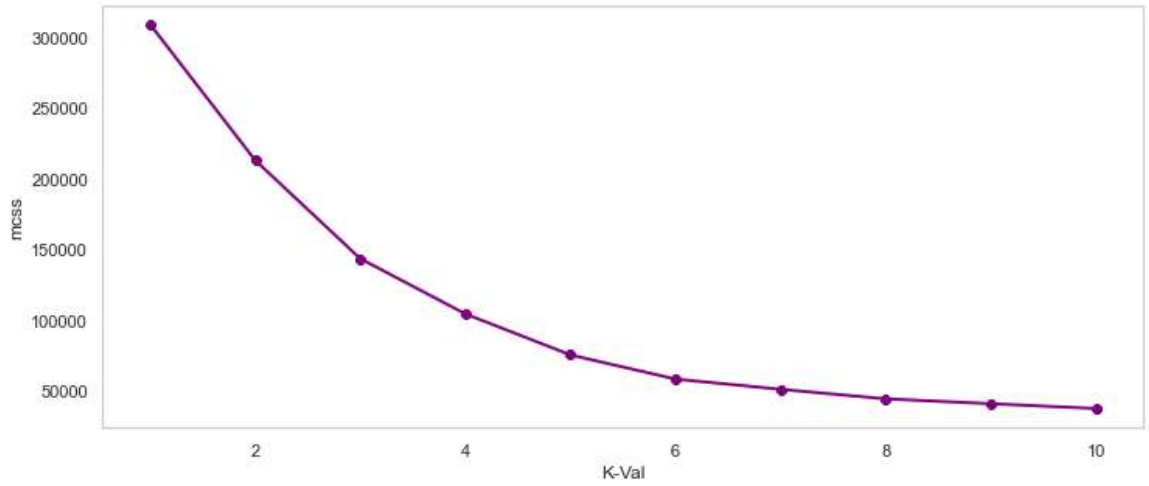
In [44]:

```
plt.scatter(X2[:,0],X2[:,1] , c = km.labels_ , cmap = 'rainbow')
plt.scatter(centroids_1[:,0] , centroids_1[:,1] , color = 'black')
plt.title('Customer plot ')
plt.xlabel('Annual Income(k$)')
plt.ylabel('Spending Score (1-100)')
plt.show()
```



In [46]:

```
X3 = df.iloc[:,1:]
mcss = []
for k in range(1,11):
    kmeans = KMeans(n_clusters = k , init = "k-means++")
    kmeans.fit(X3)
    mcss.append(kmeans.inertia_)
plt.figure(figsize = (12,5))
plt.grid()
plt.plot(range(1,11),mcss,linewidth=2,color="purple",marker="8")
plt.xlabel("K-Val")
plt.ylabel("mcss")
plt.show()
```



In [47]:

```
km_2 = KMeans(n_clusters = 6)
label = km_2.fit_predict(X3)
label
```

Out[47]:

```
array([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, 1, 4, 1, 3,
       0, 4, 1, 3, 3, 3, 1, 3, 3, 1, 1, 1, 1, 1, 3, 1, 1, 3, 1, 1, 1, 3,
       1, 1, 3, 3, 1, 1, 1, 1, 1, 3, 1, 3, 3, 1, 1, 3, 1, 1, 3, 1, 1, 3,
       3, 1, 1, 3, 1, 3, 3, 3, 1, 3, 1, 3, 3, 1, 1, 3, 1, 3, 1, 1, 1, 1,
       1, 3, 3, 3, 3, 3, 1, 1, 1, 1, 3, 3, 3, 2, 3, 2, 5, 2, 5, 2, 5, 2,
       3, 2, 5, 2, 5, 2, 5, 2, 5, 2, 3, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
       5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
       5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2, 5, 2,
       5, 2])
```

In [48]:

```
centroids_3 = km_2.cluster_centers_
centroids_3
```

Out[48]:

```
array([[44.14285714, 25.14285714, 19.52380952],
       [56.15555556, 53.37777778, 49.08888889],
       [32.69230769, 86.53846154, 82.12820513],
       [27.        , 56.65789474, 49.13157895],
       [25.27272727, 25.72727273, 79.36363636],
       [41.68571429, 88.22857143, 17.28571429]])
```

In [49]:

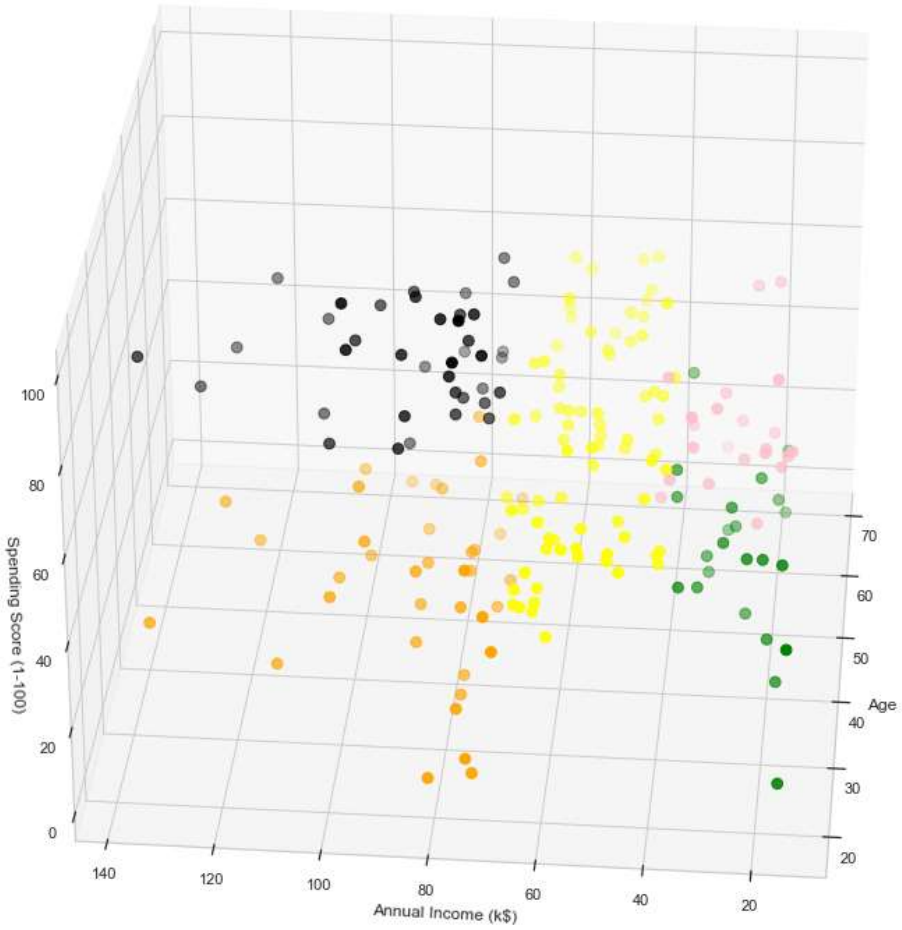
```
df['Labels'] = label
df['Labels']
```

Out[49]:

```
0      0
1      4
2      0
3      4
4      0
      ..
195    2
196    5
197    2
198    5
199    2
Name: Labels, Length: 200, dtype: int32
```

In [56]:

```
km = KMeans(n_clusters=5)
clusters = km.fit_predict(df.iloc[:,1:])
df["label"] = clusters

from mpl_toolkits.mplot3d import Axes3D


fig = plt.figure(figsize=(25,15))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(df.Age[df.label == 0], df["Annual Income (k$)"][df.label == 0], df["Spending
Score (1-100)"][df.label == 0], c='pink', s=60)
ax.scatter(df.Age[df.label == 1], df["Annual Income (k$)"][df.label == 1], df["Spending
Score (1-100)"][df.label == 1], c='green', s=60)
ax.scatter(df.Age[df.label == 2], df["Annual Income (k$)"][df.label == 2], df["Spending
Score (1-100)"][df.label == 2], c='yellow', s=60)
ax.scatter(df.Age[df.label == 3], df["Annual Income (k$)"][df.label == 3], df["Spending
Score (1-100)"][df.label == 3], c='orange', s=60)
ax.scatter(df.Age[df.label == 4], df["Annual Income (k$)"][df.label == 4], df["Spending
Score (1-100)"][df.label == 4], c='black', s=60)
ax.view_init(30, 185)
plt.xlabel("Age")
plt.ylabel("Annual Income (k$)")
ax.set_zlabel('Spending Score (1-100)')
plt.show()
```


Spending Score (1-100)
100
80
60
40
20
0
140
120
100
80
60
40
20
Annual Income (k$)
70
60
50
40
30
20
Age