In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore', category=UserWarning)
warnings.filterwarnings('ignore', category=FutureWarning)
```

In [2]:
```
df = pd.read_csv('Mall_Customers.csv')
df.head()
```

Out[2]:

	CustomerID	Genre	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 [3]:
```
df.rename(columns={'Genre':'Gender'},inplace = True)
```

In [4]:
```
df.head()
```

Out[4]:

	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 [5]:
```
df.shape
```

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

In [6]:
```
df.describe()
```

Out[6]:

	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 [7]:
```
df.dtypes
```

Out[7]:
```
CustomerID                int64
Gender                   object
Age                       int64
Annual Income (k$)        int64
Spending Score (1-100)    int64
dtype: object
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

In [9]:
```
df.drop(["CustomerID"],axis=1,inplace = True)
```

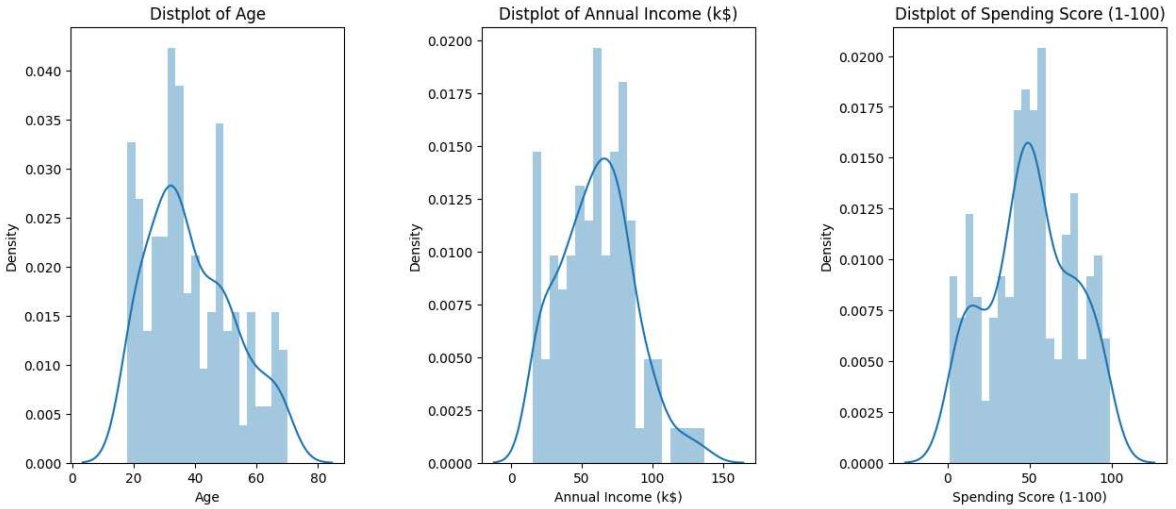
In [10]:
```
df.head()
```

Out[10]:

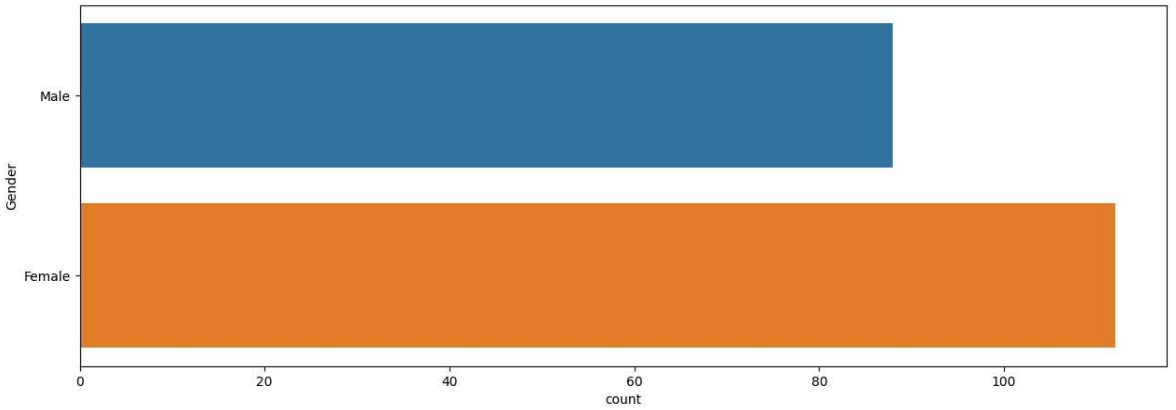
	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40

In [11]:
```
plt.figure(1, figsize = (15,6))
n = 0
for x in ['Age','Annual Income (k$)','Spending Score (1-100)']:
    n += 1
    plt.subplot(1,3,n)
    plt.subplots_adjust(hspace = 0.5, wspace = 0.5)
    sns.distplot(df[x], bins = 20)
```

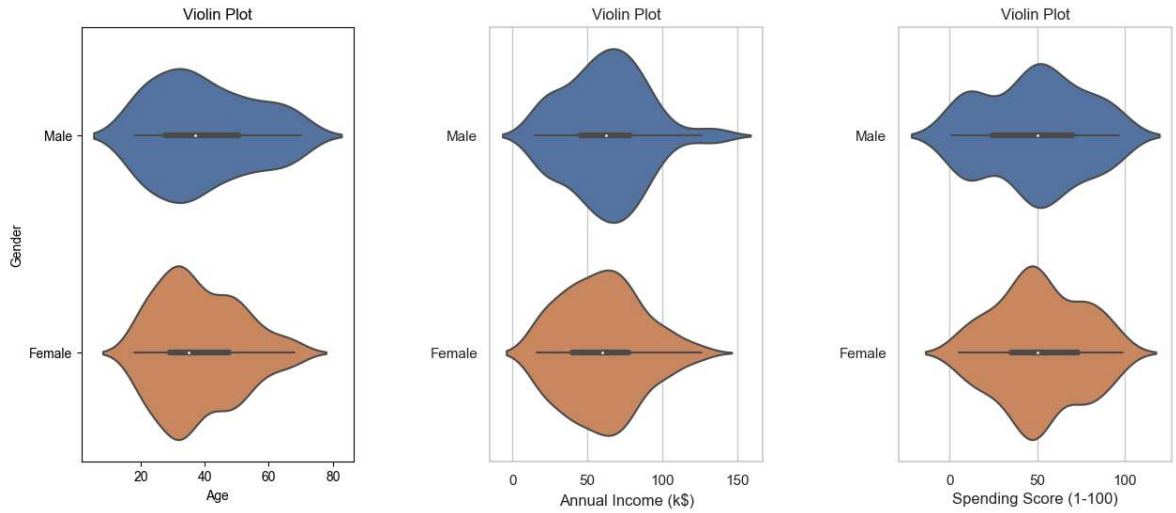
```
    plt.title("Distplot of {}".format(x))
plt.show()
```



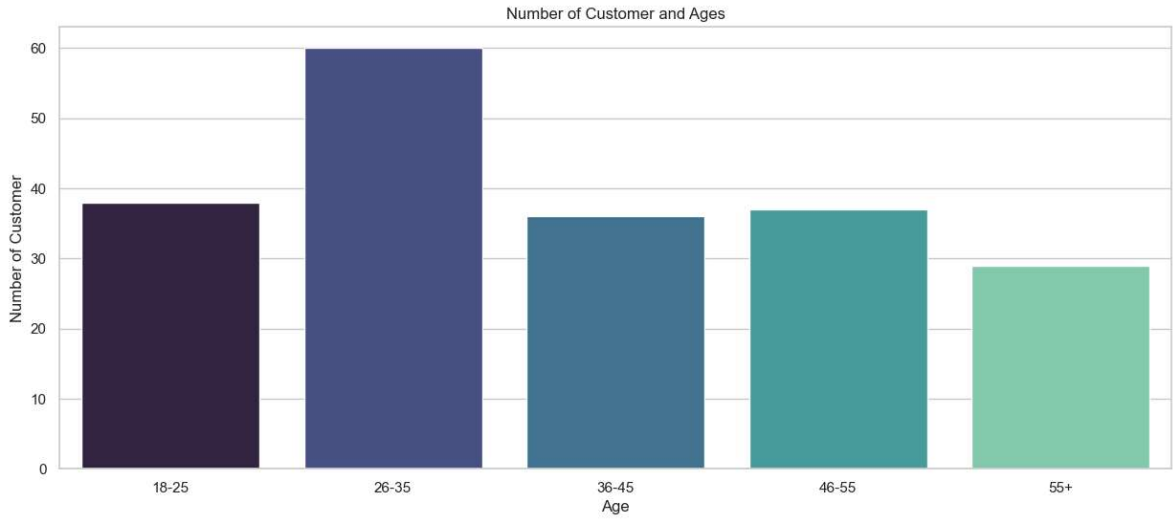
In [12]:
```
plt.figure(figsize=(15,5))
sns.countplot(y='Gender',data = df)
plt.show()
```



In [13]:
```
plt.figure(1, figsize = (15,6))
n = 0
for cols in ['Age','Annual Income (k$)','Spending Score (1-100)']:
    n += 1
    plt.subplot(1,3,n)
    sns.set(style = "whitegrid")
    plt.subplots_adjust(hspace = 0.5, wspace = 0.5)
    sns.violinplot(x = cols, y = 'Gender', data = df)
    plt.ylabel('Gender' if n == 1 else '')
    plt.title('Violin Plot')
plt.show()
```

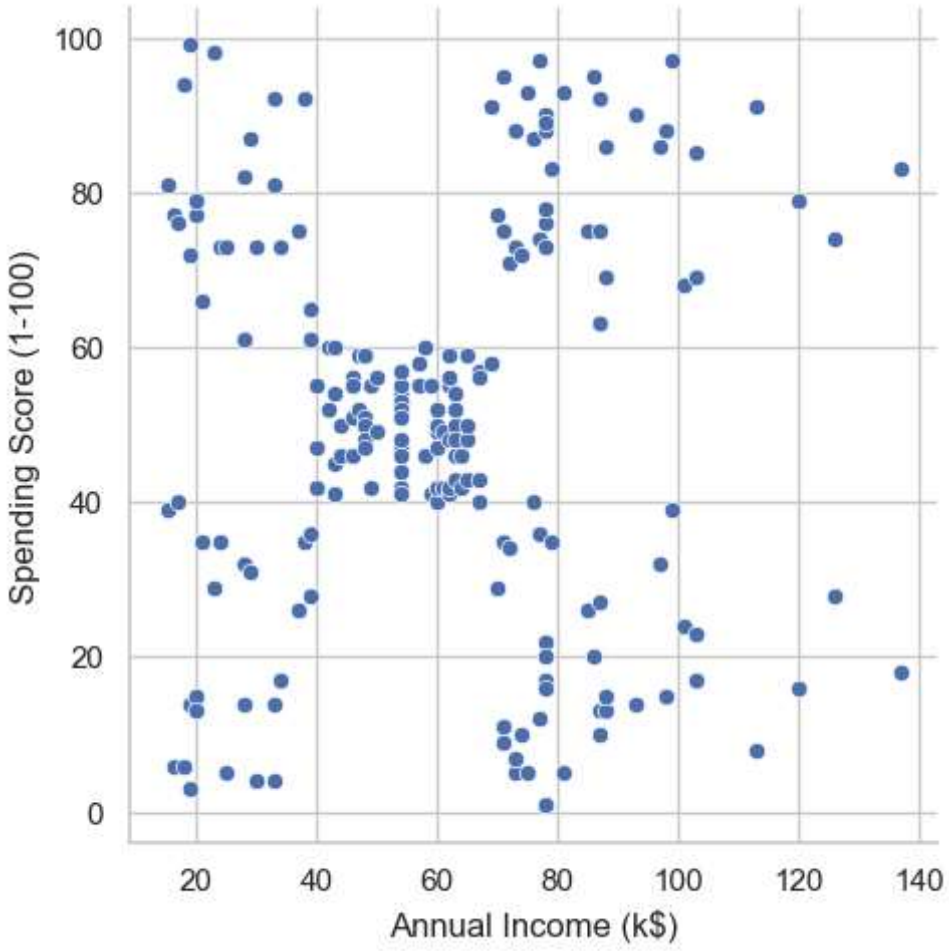


In [14]:
```
age_18_25 = df.Age[(df.Age>=18) & (df.Age <= 25)]
age_26_35 = df.Age[(df.Age>=26) & (df.Age <= 35)]
age_36_45 = df.Age[(df.Age>=36) & (df.Age <= 45)]
age_46_55 = df.Age[(df.Age>=46) & (df.Age <= 55)]
age_55above = df.Age[(df.Age>=56)]

agex = ["18-25","26-35","36-45","46-55","55+"]
agey = [len(age_18_25.values),len(age_26_35.values),len(age_36_45.values),len(ag

plt.figure(figsize = (15,6))
sns.barplot(x = agex, y = agey, palette = 'mako')
plt.title('Number of Customer and Ages')
plt.xlabel('Age')
plt.ylabel('Number of Customer')
plt.show()
```



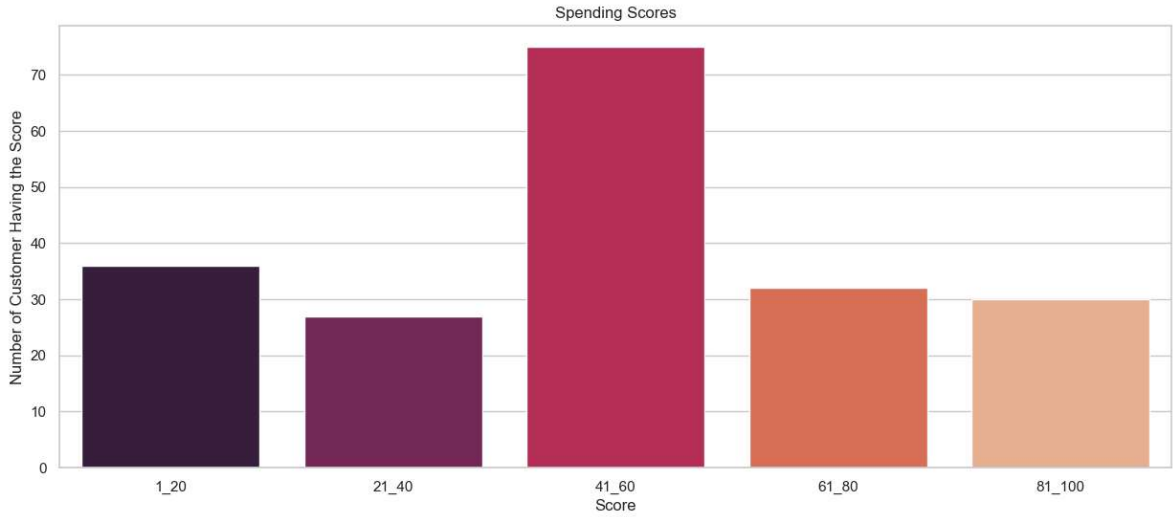
In [15]:
```
sns.relplot(x = "Annual Income (k$)", y = "Spending Score (1-100)", data = df)
```

Out[15]:
```
<seaborn.axisgrid.FacetGrid at 0x154a218b310>
```

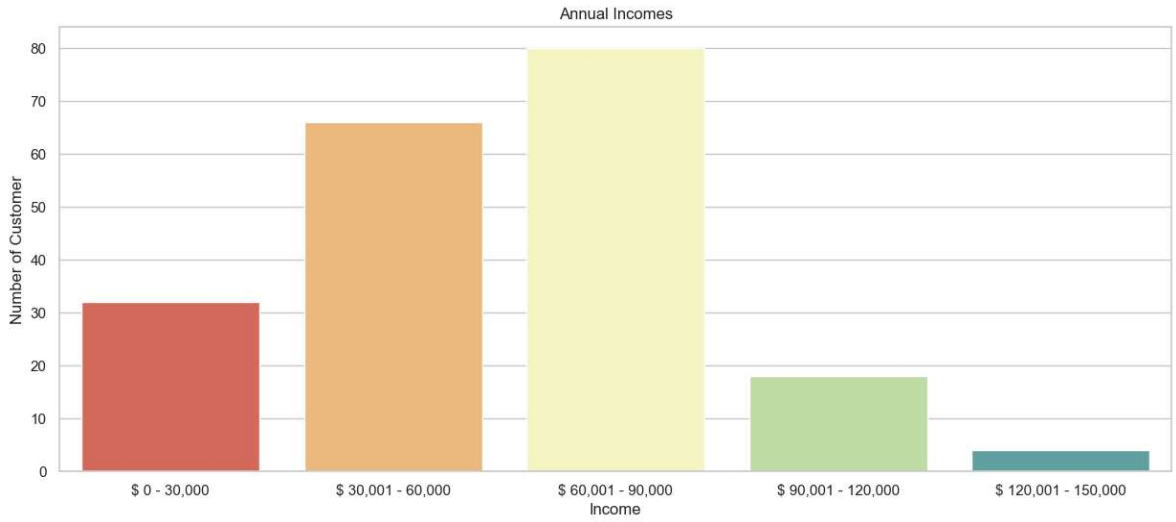


In [16]:

```
ss_1_20  = df["Spending Score (1-100)"][(df["Spending Score (1-100)"]>= 1) & (df
ss_21_40  = df["Spending Score (1-100)"][(df["Spending Score (1-100)"]>= 21) & (
ss_41_60  = df["Spending Score (1-100)"][(df["Spending Score (1-100)"]>= 41) & (
ss_61_80  = df["Spending Score (1-100)"][(df["Spending Score (1-100)"]>= 61) & (
ss_81_100  = df["Spending Score (1-100)"][(df["Spending Score (1-100)"]>= 81) &

ssx = ["1_20","21_40","41_60","61_80","81_100"]
ssy = [len(ss_1_20.values),len(ss_21_40.values),len(ss_41_60.values),len(ss_61_8

plt.figure(figsize = (15,6))
sns.barplot(x = ssx, y = ssy, palette = 'rocket')
plt.title('Spending Scores')
plt.xlabel('Score')
plt.ylabel('Number of Customer Having the Score')
plt.show()
```

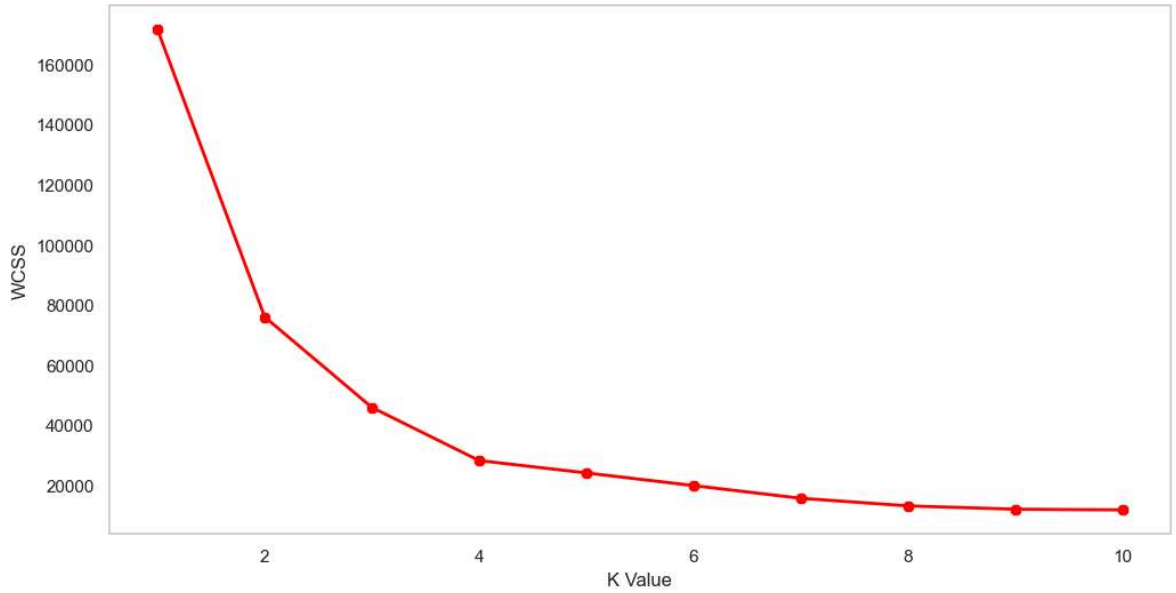


In [17]:
```
ai0_30 = df['Annual Income (k$)'][(df['Annual Income (k$)']>=0) & (df['Annual In
ai31_60 = df['Annual Income (k$)'][(df['Annual Income (k$)']>=31) & (df['Annual
ai61_90 = df['Annual Income (k$)'][(df['Annual Income (k$)']>=61) & (df['Annual
ai91_120 = df['Annual Income (k$)'][(df['Annual Income (k$)']>=91) & (df['Annual
ai121_150 = df['Annual Income (k$)'][(df['Annual Income (k$)']>=121) & (df['Annu

aix = ["$ 0 - 30,000", "$ 30,001 - 60,000", "$ 60,001 - 90,000", "$ 90,001 - 120
aiy = [len(ai0_30.values),len(ai31_60.values),len(ai61_90.values),len(ai91_120.v

plt.figure(figsize = (15,6))
sns.barplot(x = aix, y = aiy, palette = 'Spectral')
plt.title('Annual Incomes')
plt.xlabel('Income')
plt.ylabel('Number of Customer')
plt.show()
```



In [18]:
```
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=(12,6))
plt.grid()
plt.plot(range(1,11),wcss, linewidth=2 , color = 'red', marker = '8')
```

```
plt.xlabel('K Value')
plt.ylabel('WCSS')
plt.show()
```



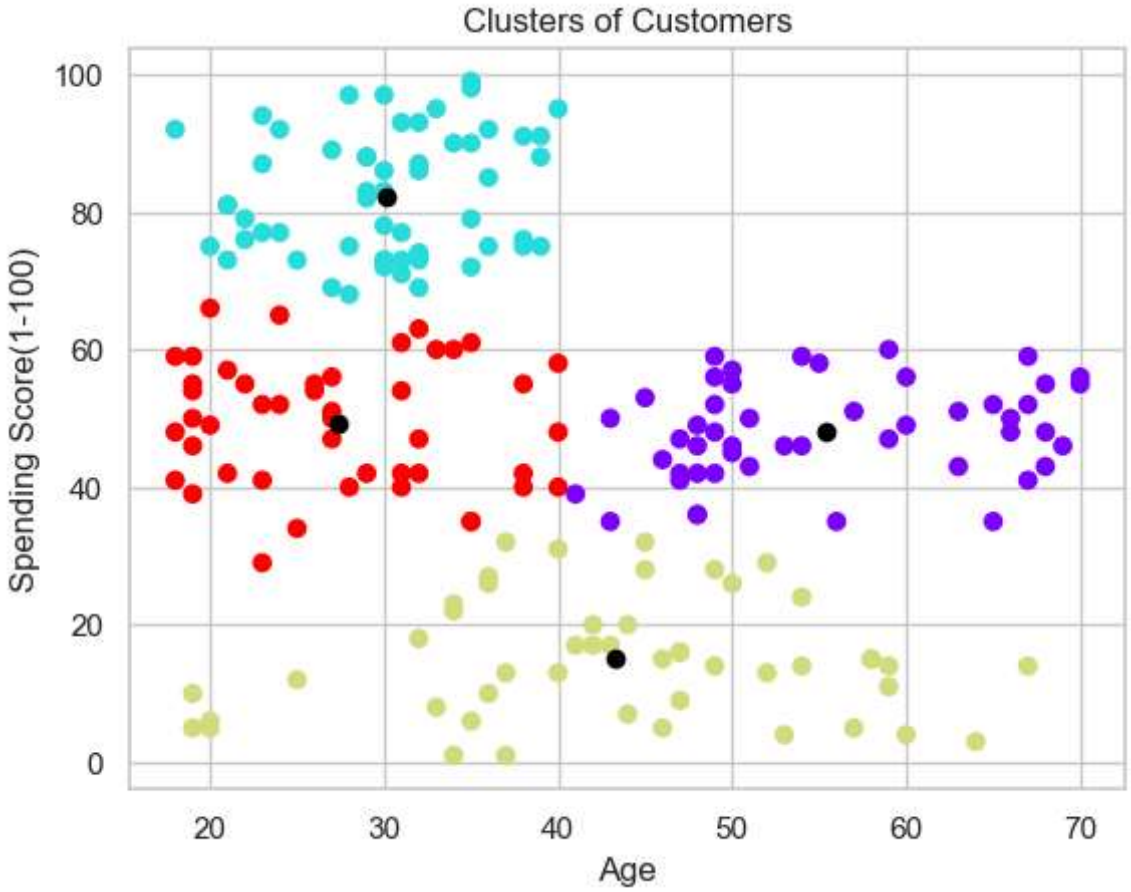
In [19]:
```
kmeans = KMeans(n_clusters = 4)
label = kmeans.fit_predict(x1)
print(label)
```

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

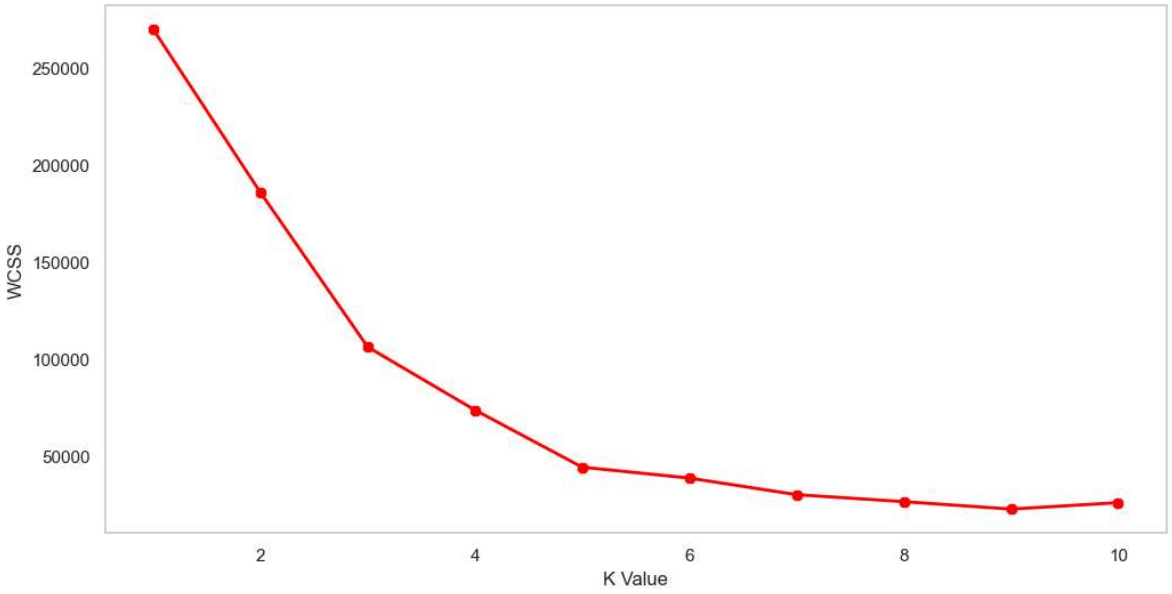
In [20]:
```
print(kmeans.cluster_centers_)
```

```
[[55.40816327 48.04081633]
 [30.1754386  82.35087719]
 [43.29166667 15.02083333]
 [27.32608696 49.36956522]]
```

In [21]:
```
plt.scatter(x1[:,0],x1[:,1],c = kmeans.labels_, cmap = 'rainbow')
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1],color = '
plt.title('Clusters of Customers')
plt.xlabel('Age')
plt.ylabel('Spending Score(1-100)')
plt.show()
```



In [22]:
```
x2 =df.loc[:, ['Annual Income (k$)','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(x2)
    wcss.append(kmeans.inertia_)
plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11),wcss, linewidth=2 , color = 'red', marker = '8')
plt.xlabel('K Value')
plt.ylabel('WCSS')
plt.show()
```



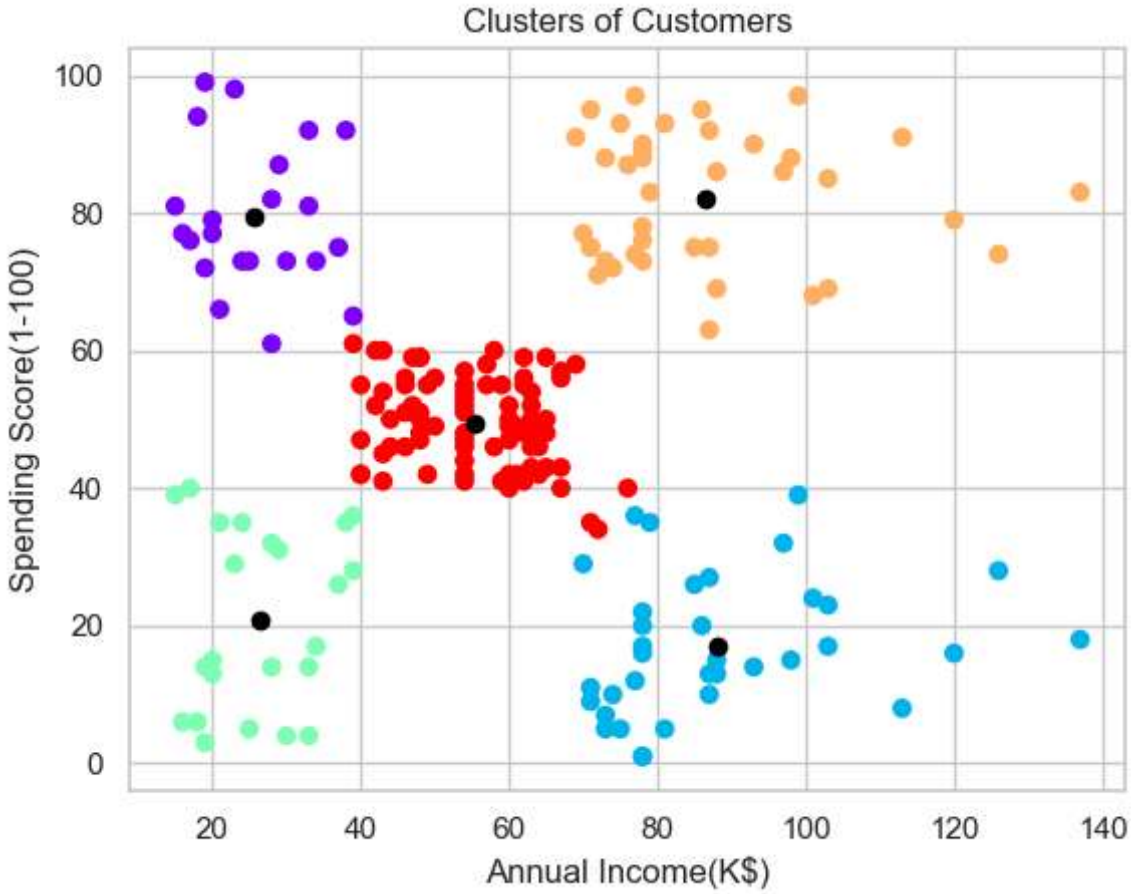
In [23]:
```
kmeans = KMeans(n_clusters = 5)
label = kmeans.fit_predict(x2)
print(label)
```

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

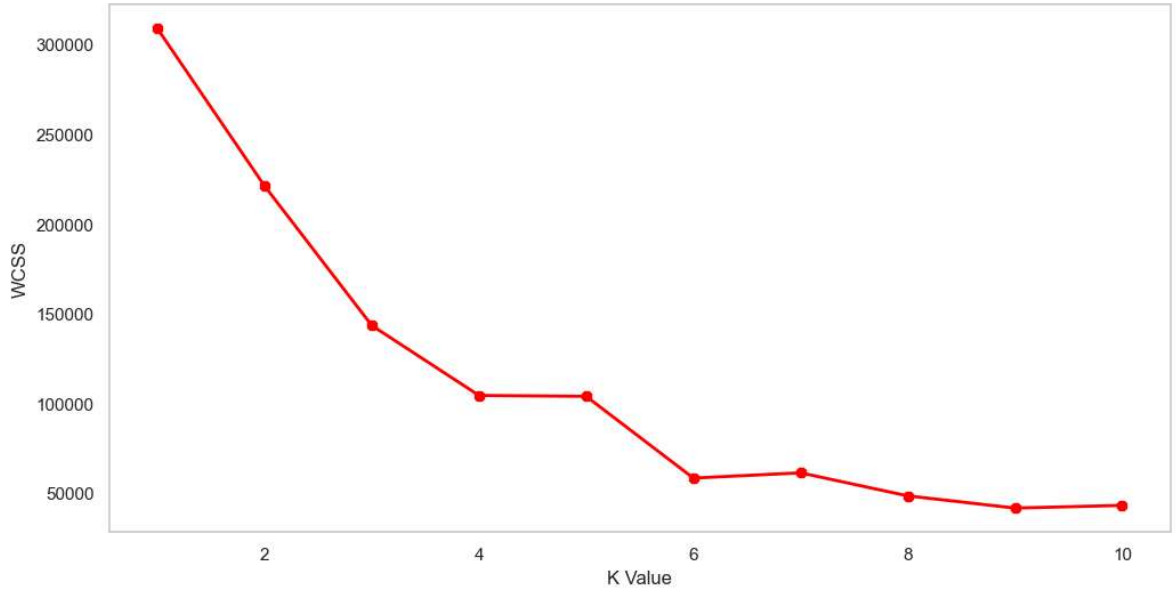
In [24]:
```
print(kmeans.cluster_centers_)
```

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

In [25]:
```
plt.scatter(x2[:,0],x2[:,1],c = kmeans.labels_, cmap = 'rainbow')
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1],color = '
plt.title('Clusters of Customers')
plt.xlabel('Annual Income(K$)')
plt.ylabel('Spending Score(1-100)')
plt.show()
```



In [26]:
```
x3 = df.iloc[:,1:]
wcss=[]
for k in range(1,11):
    kmeans = KMeans(n_clusters = k, init = 'k-means++')
    kmeans.fit(x3)
    wcss.append(kmeans.inertia_)
plt.figure(figsize=(12,6))
plt.grid()
plt.plot(range(1,11),wcss, linewidth=2 , color = 'red', marker = '8')
plt.xlabel('K Value')
plt.ylabel('WCSS')
plt.show()
```



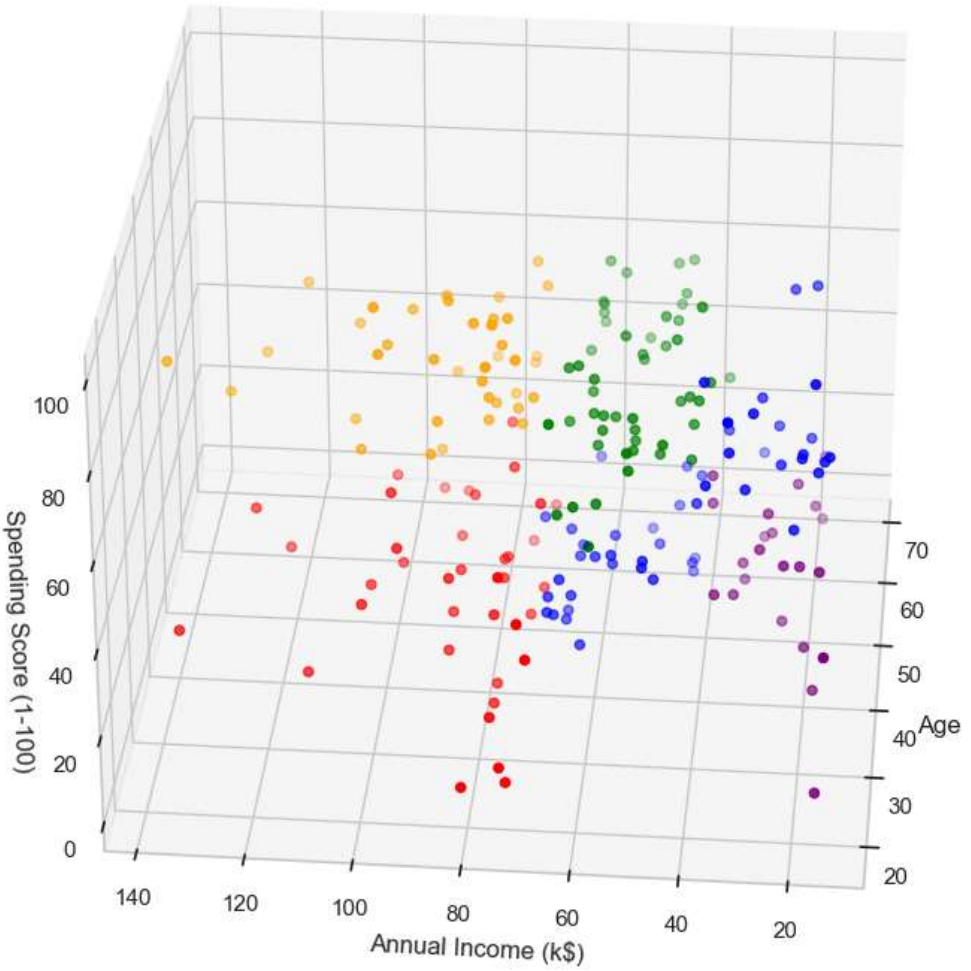
In [27]:
```
kmeans = KMeans(n_clusters = 5)
label = kmeans.fit_predict(x3)
print(label)
```

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

In [28]:
```
print(kmeans.cluster_centers_)
```

```
[[40.32432432 87.43243243 18.18918919]
 [24.96       28.04       77.        ]
 [32.69230769 86.53846154 82.12820513]
 [43.93421053 55.21052632 49.44736842]
 [45.2173913  26.30434783 20.91304348]]
```

In [29]:
```
clusters = kmeans.fit_predict(x3)
df['label'] = clusters

from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize = (20,10))
ax = fig.add_subplot(111,projection='3d')
ax.scatter(df.Age[df.label == 0],df['Annual Income (k$)'][df.label == 0], df['Sp
ax.scatter(df.Age[df.label == 1],df['Annual Income (k$)'][df.label == 1], df['Sp
ax.scatter(df.Age[df.label == 2],df['Annual Income (k$)'][df.label == 2], df['Sp
ax.scatter(df.Age[df.label == 3],df['Annual Income (k$)'][df.label == 3], df['Sp
ax.scatter(df.Age[df.label == 4],df['Annual Income (k$)'][df.label == 4], df['Sp
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

In [ ]: