```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.pyplot import figure
import warnings
warnings.filterwarnings('ignore')

df= pd.read_csv('Mall_Customers.csv')

df
```

	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

```
df.shape
```

```
(200, 5)
```

```
df.describe()
```

	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

```
df.dtypes
```

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

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

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

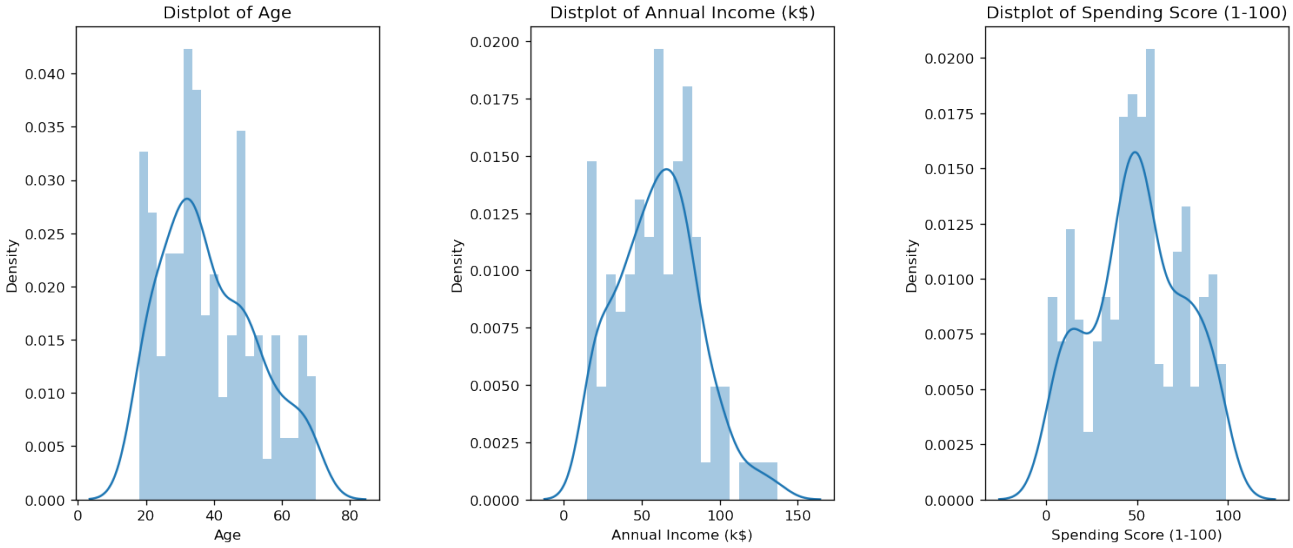
```
df.drop(['CustomerID'],axis=1, inplace=True)
```

```
df
```

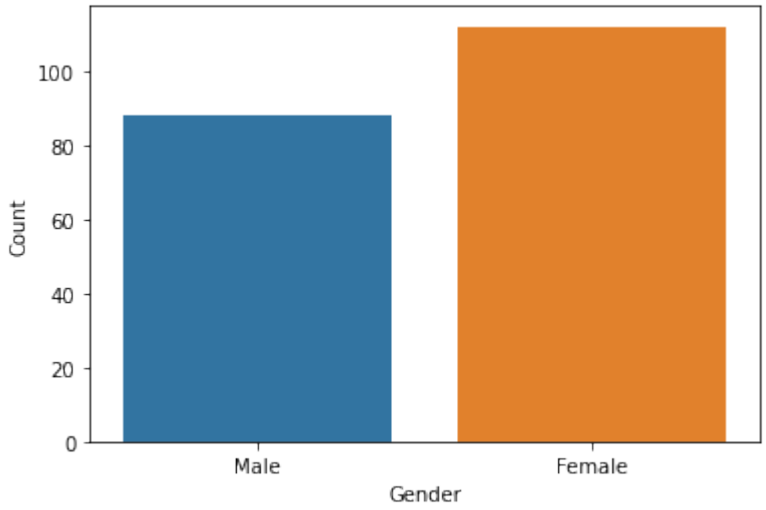
	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40
...	...	...	...	...
195	Female	35	120	79
196	Female	45	126	28
197	Male	32	126	74
198	Male	32	137	18
199	Male	30	137	83

200 rows × 4 columns

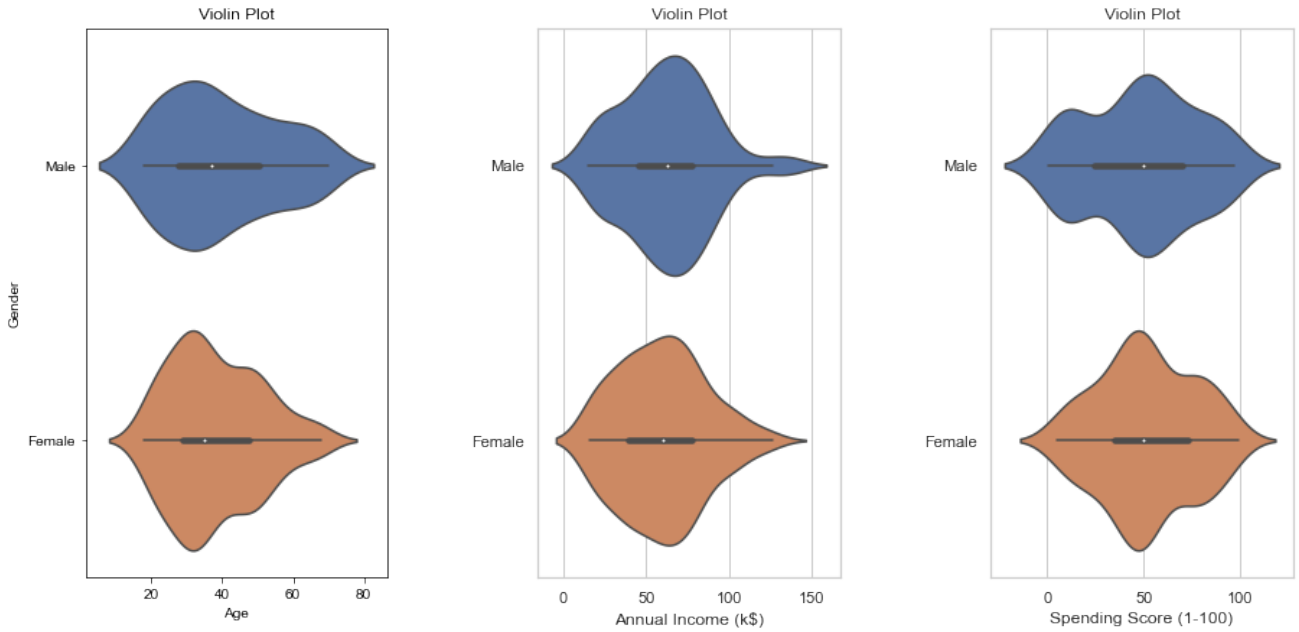
```
plt.figure(1, figsize=(15,6),dpi=120)
n=0
for x in ['Age','Annual Income (k$)','Spending Score (1-100)']:
    n+=1
    plt.subplot(1,3,n)
    plt.subplots_adjust(hspace=0.5,wspace=0.5)
    sns.distplot(df[x],bins=20)
    plt.title('Distplot of {}'.format(x))
plt.show()
```



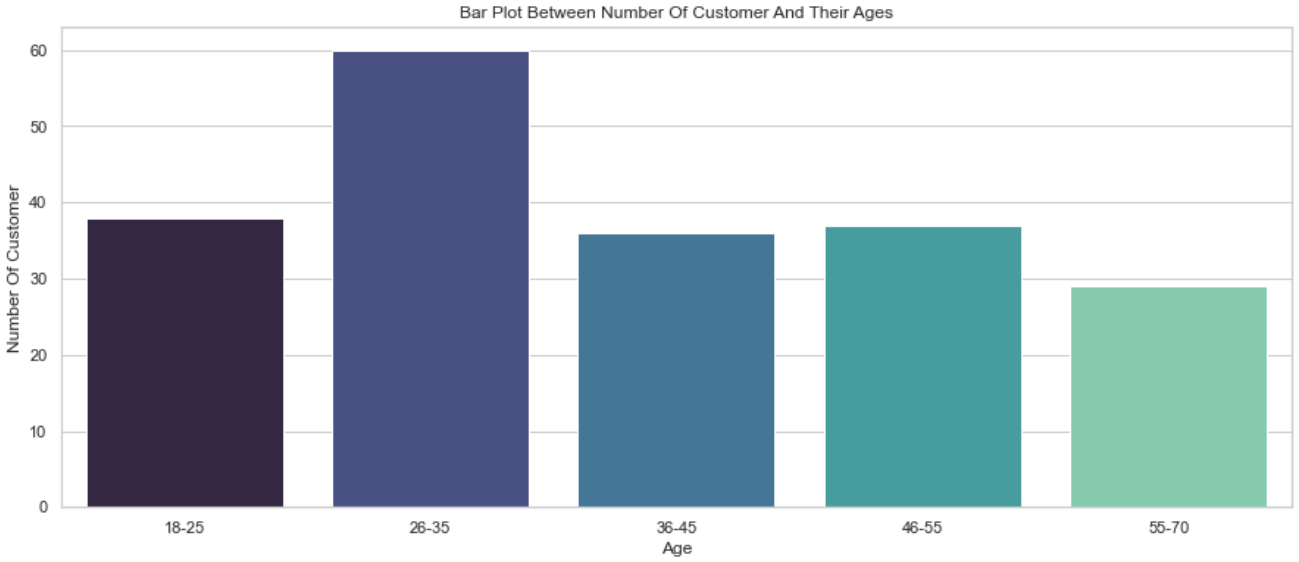

```
sns.countplot(df.Gender)
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```




```
plt.figure(1,figsize=(15,7))
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()
```

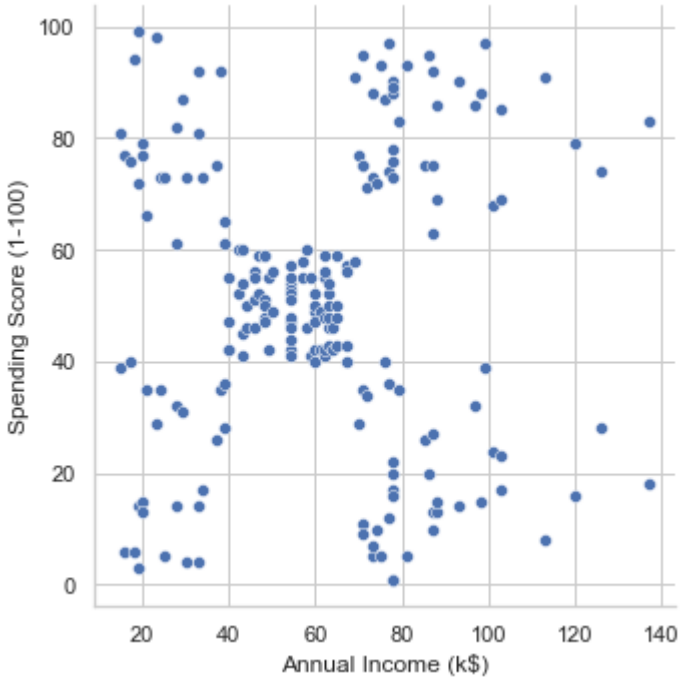



```
age1= df['Age'][(df['Age']>=18) & (df['Age']<=25)]
age2= df['Age'][(df['Age']>=26) & (df['Age']<=35)]
age3= df['Age'][(df['Age']>=36) & (df['Age']<=45)]
age4= df['Age'][(df['Age']>=46) & (df['Age']<=55)]
age5= df['Age'][(df['Age']>=56) & (df['Age']<=70)]
age_x= ['18-25','26-35','36-45','46-55','55-70']
age_y= [len(age1),len(age2),len(age3),len(age4),len(age5)]
plt.figure(figsize=(15,6))
sns.barplot(x=age_x,y=age_y,palette='mako')
plt.title('Bar Plot Between Number Of Customer And Their Ages')
plt.xlabel('Age')
plt.ylabel('Number Of Customer')
plt.show()
```

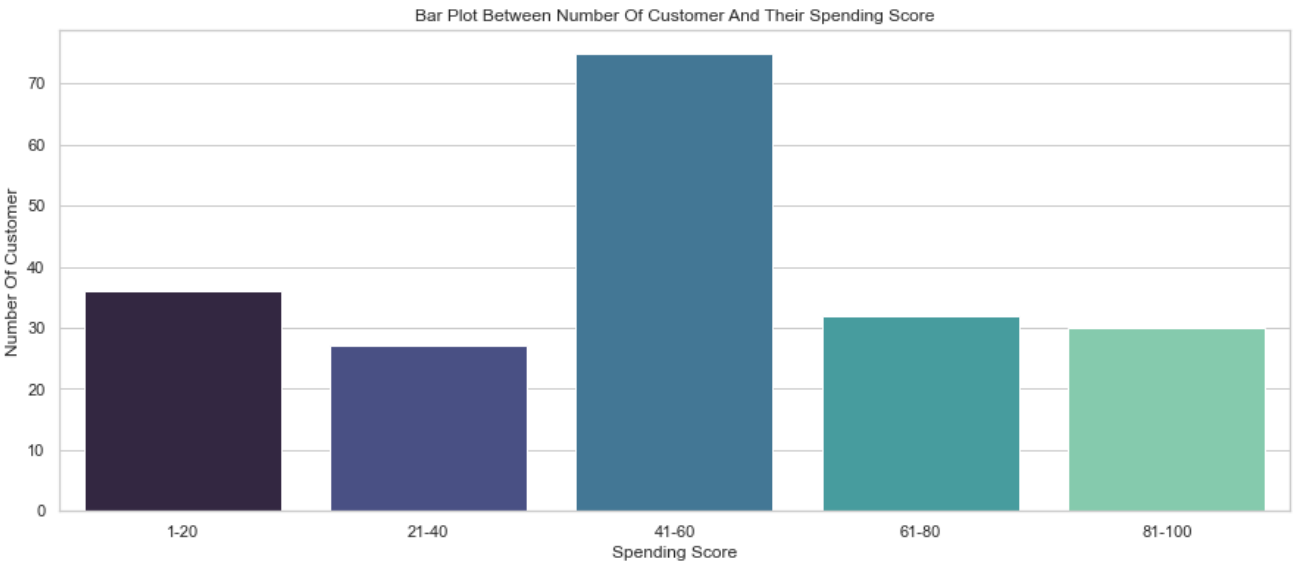



```
sns.relplot(x='Annual Income (k$)',y='Spending Score (1-100)', data=df)
```

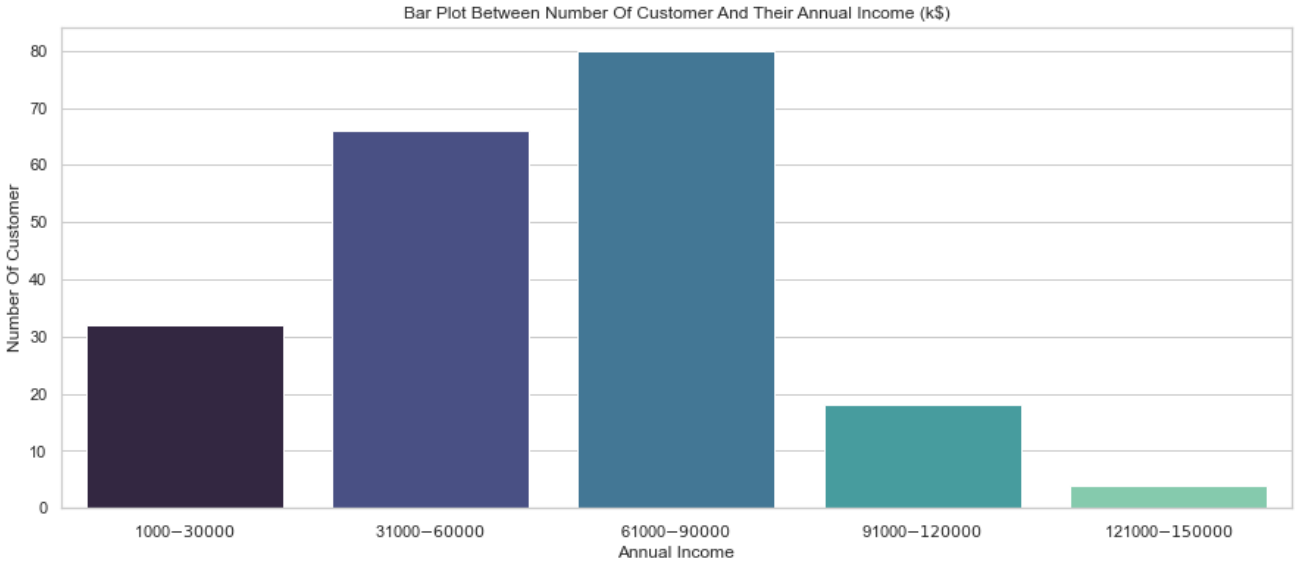
```
<seaborn.axisgrid.FacetGrid at 0x1c8c6386730>
```



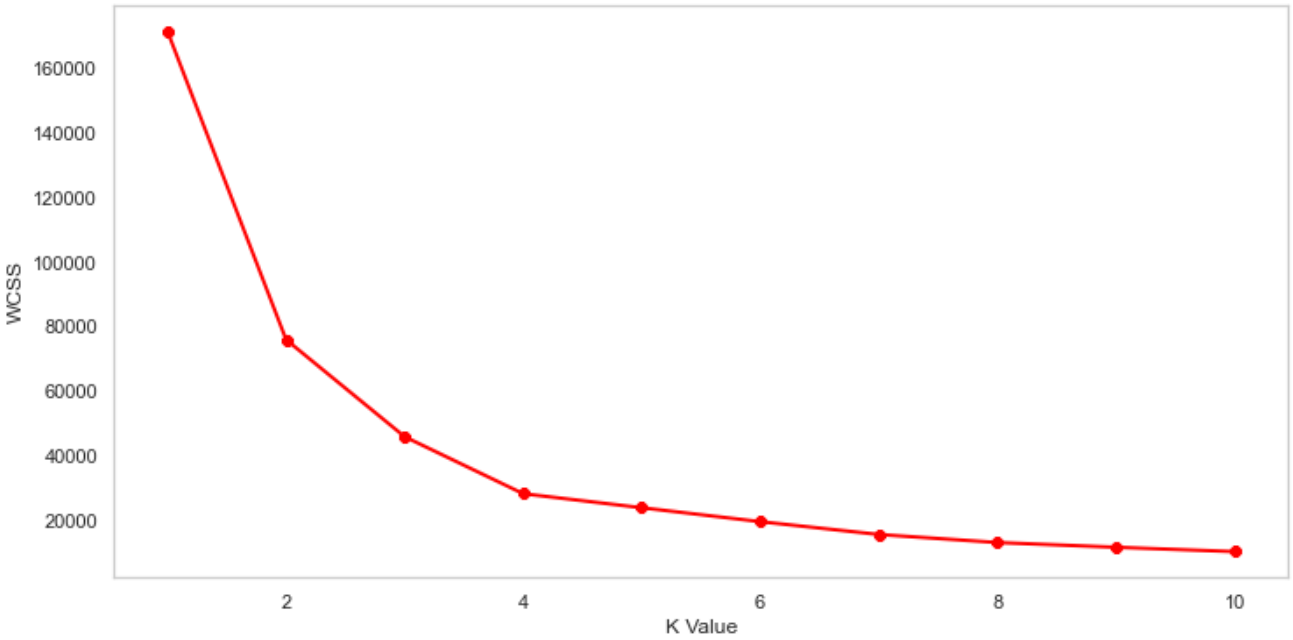

```
ss1= df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=1) & (df['Spending
ss2= df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=21) & (df['Spendin
ss3= df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=41) & (df['Spendin
ss4= df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=61) & (df['Spendin
ss5= df['Spending Score (1-100)'][(df['Spending Score (1-100)']>=81) & (df['Spendin
ss_x= ['1-20','21-40','41-60','61-80','81-100']
ss_y= [len(ss1),len(ss2),len(ss3),len(ss4),len(ss5)]
plt.figure(figsize=(15,6))
sns.barplot(x=ss_x,y=ss_y,palette='mako')
plt.title('Bar Plot Between Number Of Customer And Their Spending Score')
plt.xlabel('Spending Score')
plt.ylabel('Number Of Customer')
plt.show()
```



```
ai1= df['Annual Income (k$)'][(df['Annual Income (k$)']>=1) & (df['Annual Income (k
ai2= df['Annual Income (k$)'][(df['Annual Income (k$)']>=31) & (df['Annual Income (
ai3= df['Annual Income (k$)'][(df['Annual Income (k$)']>=61) & (df['Annual Income (
ai4= df['Annual Income (k$)'][(df['Annual Income (k$)']>=91) & (df['Annual Income (
ai5= df['Annual Income (k$)'][(df['Annual Income (k$)']>=121) & (df['Annual Income
ai_x= ['1000$-30000$','31000$-60000$','61000$-90000$','91000$-120000$','121000$-150
ai_y= [len(ai1),len(ai2),len(ai3),len(ai4),len(ai5)]
plt.figure(figsize=(15,6))
sns.barplot(x=ai_x,y=ai_y,palette='mako')
plt.title('Bar Plot Between Number Of Customer And Their Annual Income (k$)')
plt.xlabel('Annual Income')
plt.ylabel('Number Of Customer')
plt.show()
```




```
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()
```



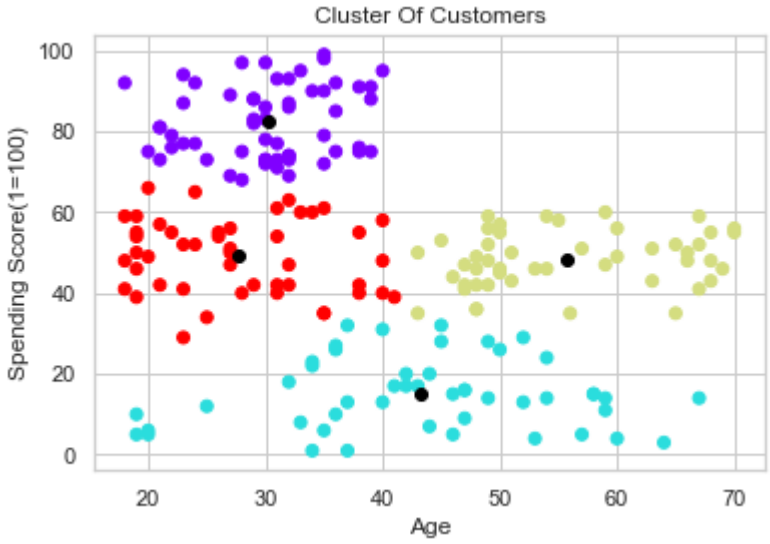

```
kmeans= KMeans(n_clusters=4)
label= kmeans.fit_predict(X1)
print(label)
```

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

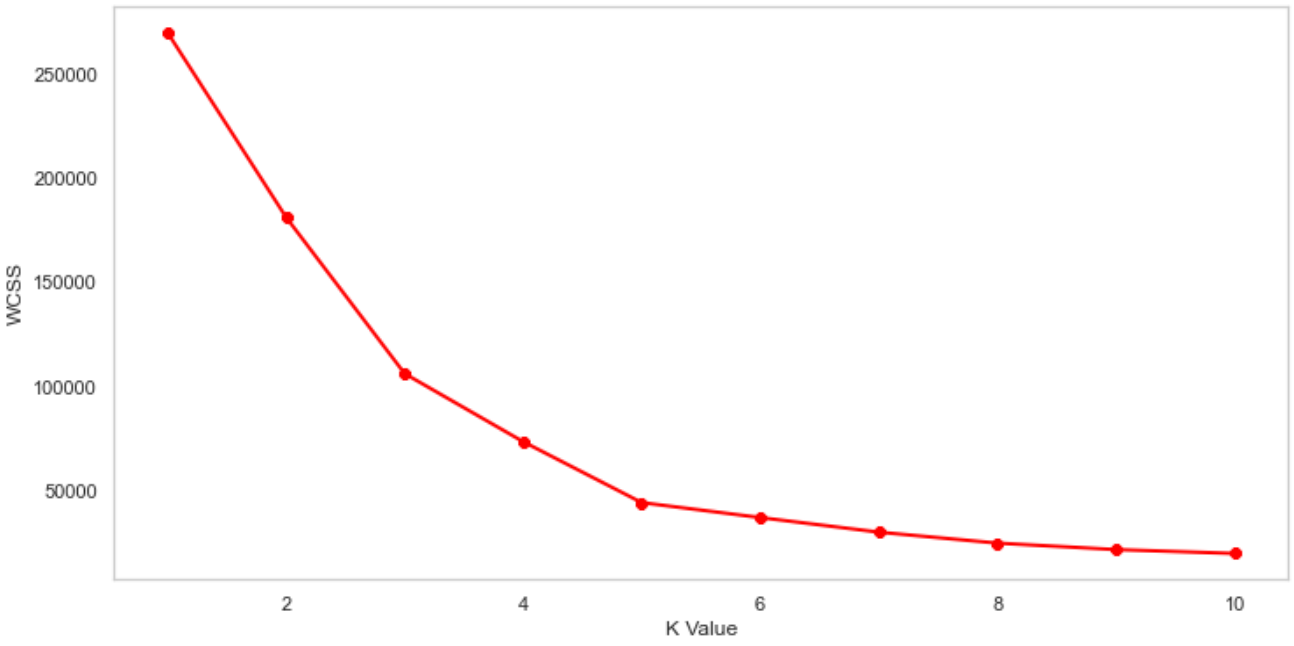
```
print(kmeans.cluster_centers_)
```

```
[[30.1754386  82.35087719]
 [43.29166667 15.02083333]
 [55.70833333 48.22916667]
 [27.61702128 49.14893617]]
```

```
plt.scatter(X1[:,0],X1[:,1],c=kmeans.labels_,cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black'
plt.title('Cluster Of Customers')
plt.xlabel('Age')
plt.ylabel('Spending Score(1=100)')
plt.show()
```




```
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()
```



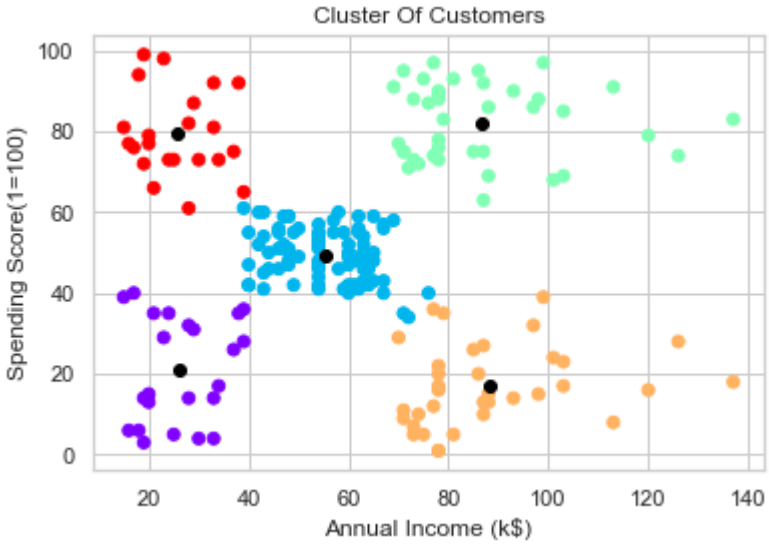

```
kmeans= KMeans(n_clusters=5)
label= kmeans.fit_predict(X2)
print(label)
```

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

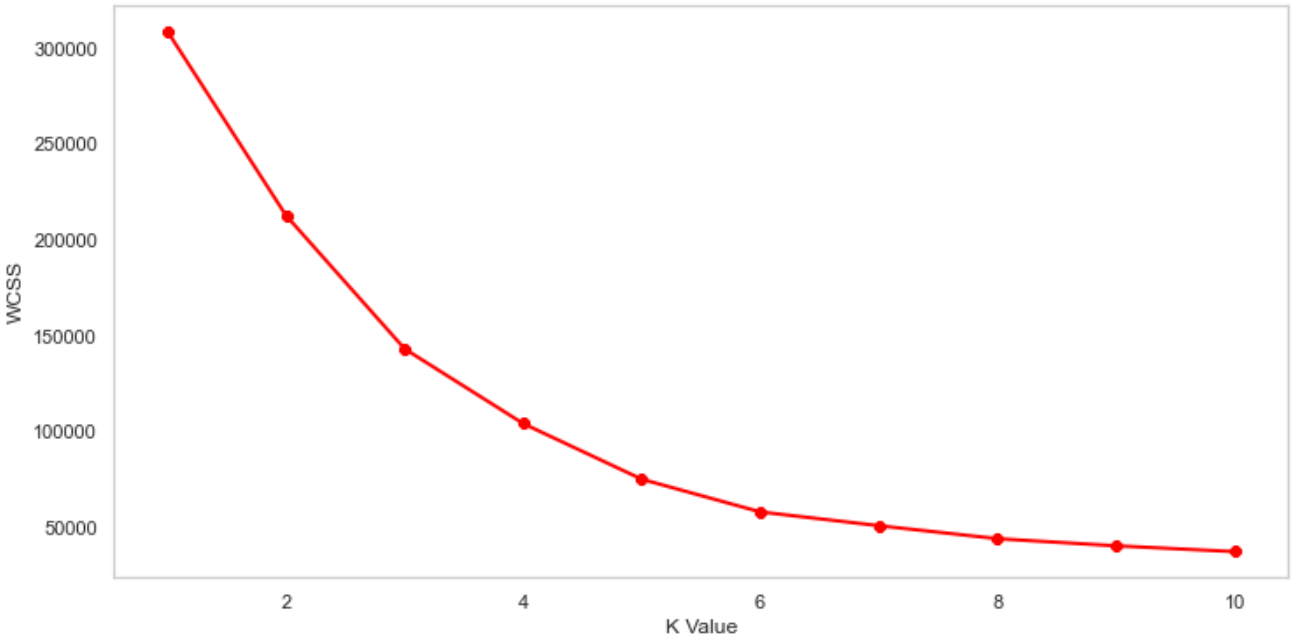
```
print(kmeans.cluster_centers_)
```

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

```
plt.scatter(X2[:,0],X1[:,1],c=kmeans.labels_,cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black'
plt.title('Cluster Of Customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score(1=100)')
plt.show()
```




```
X3= df.loc[:, ['Age','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(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()
```



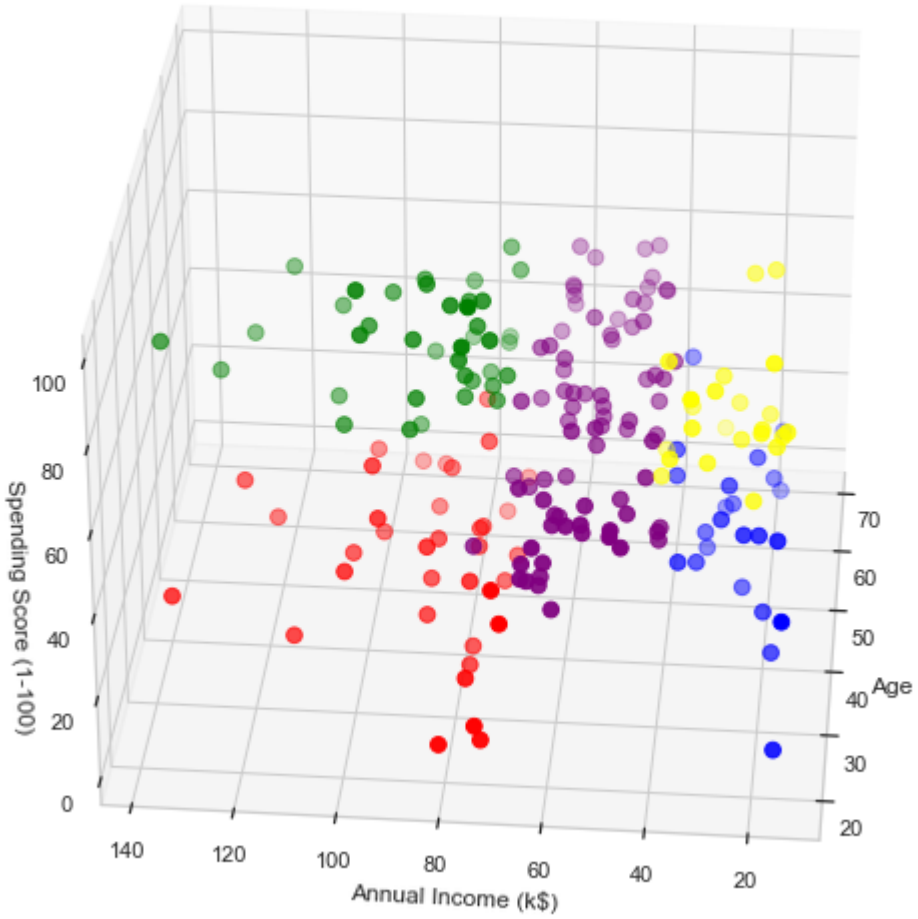

```
kmeans= KMeans(n_clusters=5)
label= kmeans.fit_predict(X3)
print(label)
```

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

```
print(kmeans.cluster_centers_)
```

```
[[24.96       28.04       77.        ]
 [43.72727273 55.48051948 49.32467532]
 [32.69230769 86.53846154 82.12820513]
 [40.66666667 87.75       17.58333333]
 [45.2173913  26.30434783 20.91304348]]
```

```
cluster= kmeans.fit_predict(X3)
df['label']=cluster
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$)"][df.label == 0], df["Spen
ax.scatter(df.Age[df.label == 1], df["Annual Income (k$)"][df.label == 1], df["Spen
ax.scatter(df.Age[df.label == 2], df["Annual Income (k$)"][df.label == 2], df["Spen
ax.scatter(df.Age[df.label == 3], df["Annual Income (k$)"][df.label == 3], df["Spen
ax.scatter(df.Age[df.label == 4], df["Annual Income (k$)"][df.label == 4], df["Spen
ax.view_init(30, 185)
plt.xlabel("Age")
plt.ylabel("Annual Income (k$)")
ax.set_zlabel('Spending Score (1-100)')
plt.show()
```



SQL Operations on Customer Data

We'll use SQLite to perform SQL operations on our customer dataset. This can help with data segmentation, filtering, and aggregation using SQL syntax.

```
import sqlite3

# Create an in-memory SQLite database
conn = sqlite3.connect(":memory:")
cursor = conn.cursor()

# Write the DataFrame to a SQL table
df.to_sql("customers", conn, index=False, if_exists="replace")
print("Data inserted into SQL table successfully.")
```

```
Data inserted into SQL table successfully.
```

```
# Sample SQL query: Customers aged between 20 and 40
query = "SELECT * FROM customers WHERE Age BETWEEN 20 AND 40"
result = pd.read_sql_query(query, conn)
result.head()
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)	label
0	Male	21	15	81	4
1	Female	20	16	6	2
2	Female	23	16	77	4
3	Female	31	17	40	2
4	Female	22	17	76	4

```
# Average Annual Income by Gender
query = "SELECT Gender, AVG(`Annual Income (k$)`) as Avg_Income FROM customers GROU
income_by_gender = pd.read_sql_query(query, conn)
income_by_gender
```

	Gender	Avg_Income
0	Female	59.250000
1	Male	62.227273

1. Top 10 Customers with Highest Spending Score

```
query = "SELECT * FROM customers ORDER BY `Spending Score (1-100)` DESC LIMIT 10"
top_spenders = pd.read_sql_query(query, conn)
top_spenders
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)	label
0	Female	35	19	99	4
1	Female	35	23	98	4
2	Male	28	77	97	3
3	Male	30	99	97	3
4	Male	40	71	95	3
5	Female	33	86	95	3
6	Female	23	18	94	4
7	Male	32	75	93	3
8	Female	31	81	93	3
9	Male	18	33	92	4

2. Count of Customers in Spending Score Ranges (Binned)

```
query = '''
SELECT
    CASE
        WHEN `Spending Score (1-100)` BETWEEN 1 AND 20 THEN '1-20'
        WHEN `Spending Score (1-100)` BETWEEN 21 AND 40 THEN '21-40'
        WHEN `Spending Score (1-100)` BETWEEN 41 AND 60 THEN '41-60'
        WHEN `Spending Score (1-100)` BETWEEN 61 AND 80 THEN '61-80'
        ELSE '81-100'
    END AS Spending_Range,
    COUNT(*) AS Customer_Count
FROM customers
GROUP BY Spending_Range
ORDER BY Spending_Range
'''
spending_bins = pd.read_sql_query(query, conn)
spending_bins
```

	Spending_Range	Customer_Count
0	1-20	36
1	21-40	27
2	41-60	75
3	61-80	32
4	81-100	30

3. Average Spending Score by Gender

```
query = "SELECT Gender, AVG(`Spending Score (1-100)`) as Avg_Score FROM customers G
avg_score_by_gender = pd.read_sql_query(query, conn)
avg_score_by_gender
```

	Gender	Avg_Score
0	Female	51.526786
1	Male	48.511364

4. Age Bucket Distribution

```
query = '''
SELECT
    CASE
        WHEN Age < 20 THEN '<20'
        WHEN Age BETWEEN 20 AND 29 THEN '20-29'
        WHEN Age BETWEEN 30 AND 39 THEN '30-39'
        WHEN Age BETWEEN 40 AND 49 THEN '40-49'
        ELSE '50+'
    END AS Age_Bucket,
    COUNT(*) AS Customer_Count
FROM customers
GROUP BY Age_Bucket
ORDER BY Age_Bucket
'''
age_distribution = pd.read_sql_query(query, conn)
age_distribution
```

	Age_Bucket	Customer_Count
0	20-29	43
1	30-39	61
2	40-49	39
3	50+	45
4	<20	12

5. Customers with Above-Average Income and Spending Score

```
query = '''
SELECT * FROM customers
WHERE
    `Annual Income (k$)` > (SELECT AVG(`Annual Income (k$)`) FROM customers) AND
    `Spending Score (1-100)` > (SELECT AVG(`Spending Score (1-100)`) FROM customers
'''
high_value_customers = pd.read_sql_query(query, conn)
high_value_customers
```

	Gender	Age	Annual Income (k$)	Spending Score (1-100)	label
0	Male	67	62	59	0
1	Male	26	62	55	0
2	Male	49	62	56	0
3	Male	65	63	52	0
4	Female	19	63	54	0
5	Female	49	65	59	0
6	Female	50	67	57	0
7	Male	27	67	56	0
8	Female	40	69	58	0
9	Male	39	69	91	3
10	Female	31	70	77	3
11	Male	40	71	95	3
12	Male	38	71	75	3
13	Male	39	71	75	3
14	Female	31	72	71	3
15	Female	29	73	88	3
16	Male	32	73	73	3
17	Female	35	74	72	3
18	Male	32	75	93	3
19	Female	32	76	87	3
20	Male	28	77	97	3
21	Female	32	77	74	3
22	Male	34	78	90	3
23	Male	39	78	88	3
24	Female	38	78	76	3
25	Female	27	78	89	3
26	Female	30	78	78	3
27	Female	30	78	73	3
28	Female	29	79	83	3
29	Female	31	81	93	3
30	Female	36	85	75	3
31	Female	33	86	95	3
32	Male	32	87	63	3
33	Male	28	87	75	3
34	Male	36	87	92	3
35	Female	30	88	86	3