In [1]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df = pd.read_csv(r"C:\Users\patha\Downloads\Mall-Customer-Segmentation-main\Mall_Customers.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	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

Univariate Analysis

In [4]:

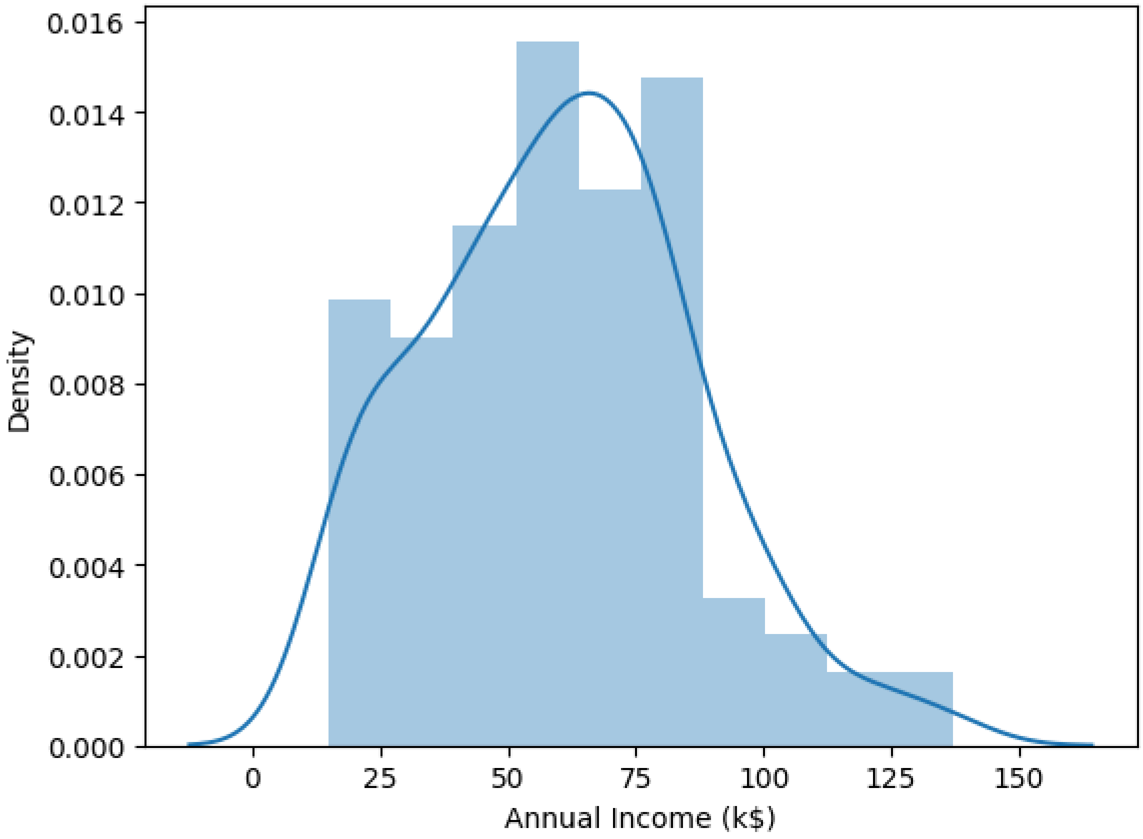
```
df.describe()
```

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]:

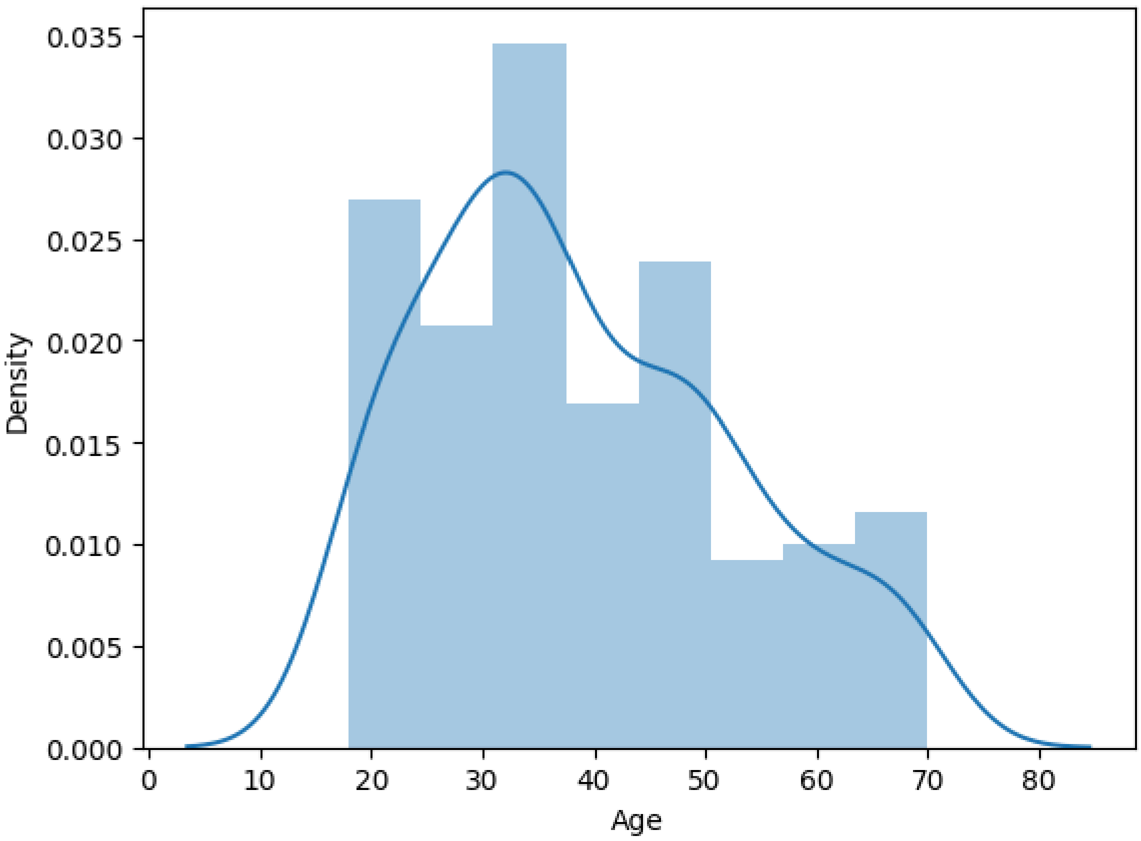
```
sns.distplot(df['Annual Income (k$)']);
```

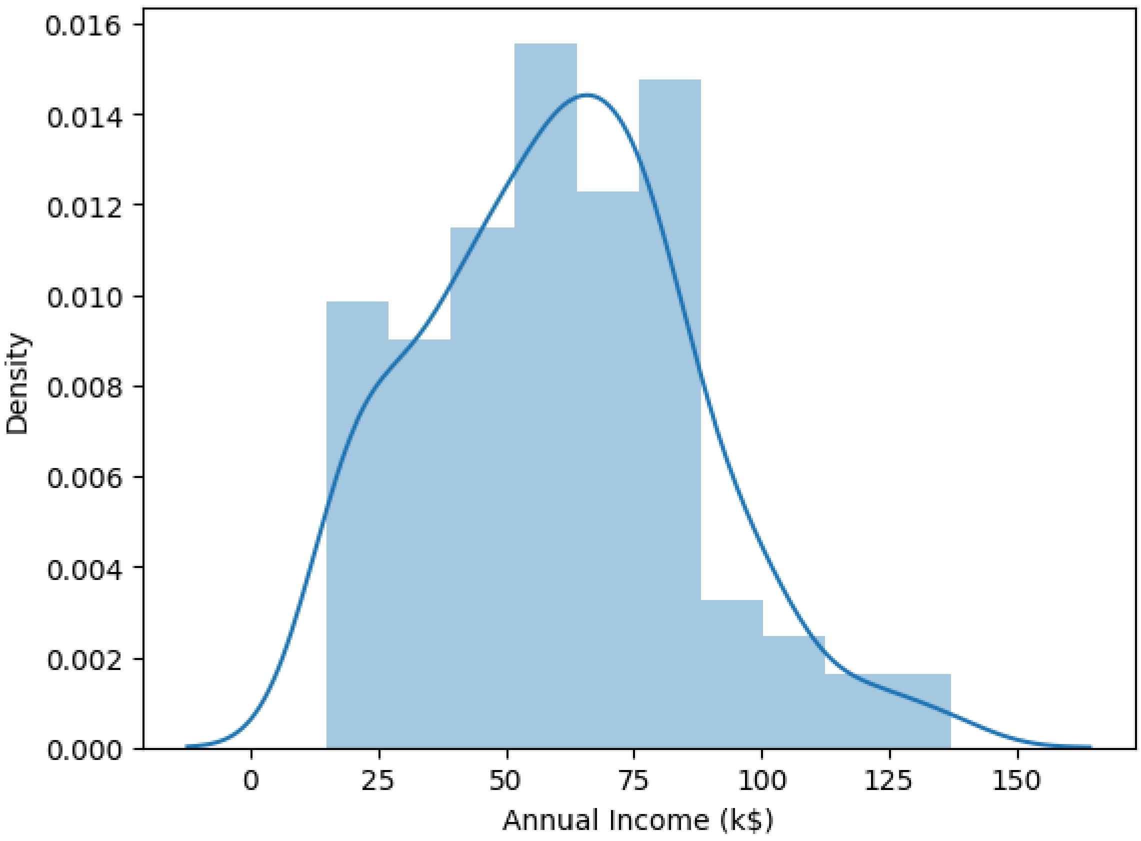


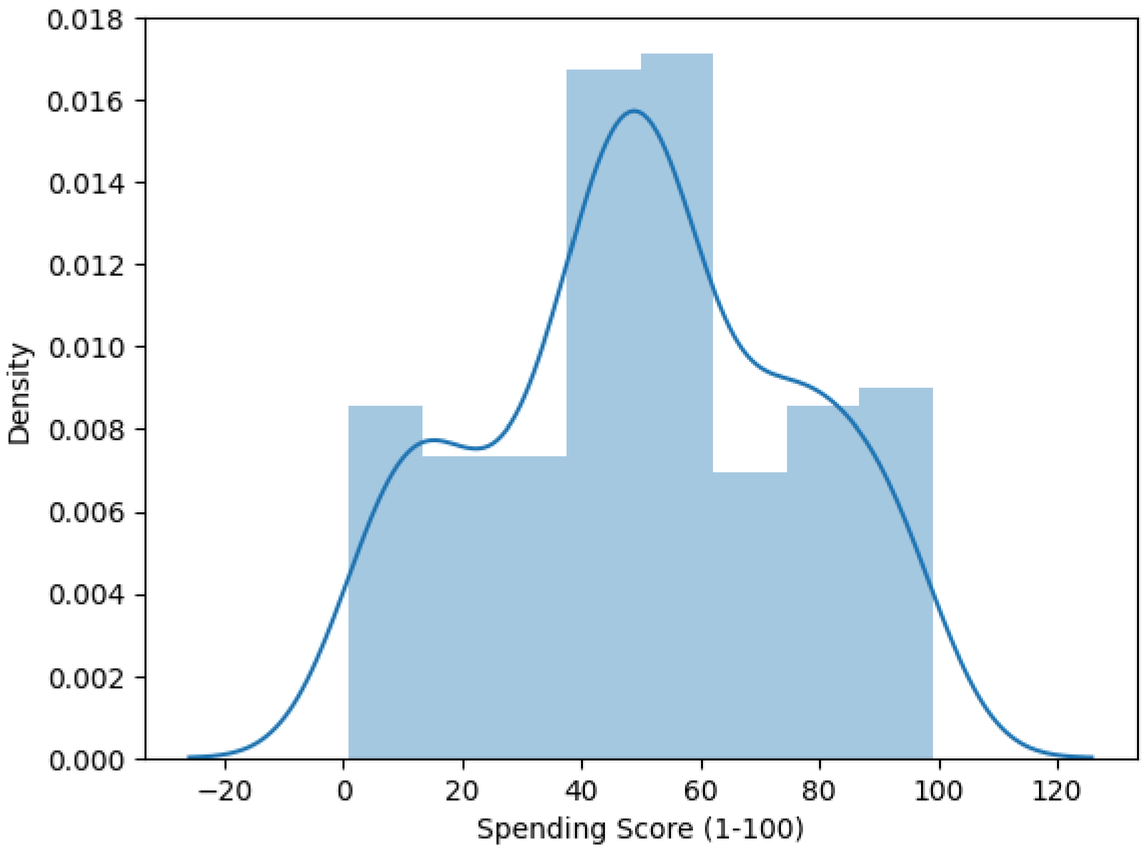
```
In [6]: df.columns
```

```
Out[6]: Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
               'Spending Score (1-100)'],
              dtype='object')
```

```
In [8]: columns = ['Age', 'Annual Income (k$)','Spending Score (1-100)']
        for i in columns:
            plt.figure()
            sns.distplot(df[i])
```


0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
0
10
20
30
40
50
60
70
80
Age


0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
Density
0
25
50
75
100
125
150
Annual Income (k$)



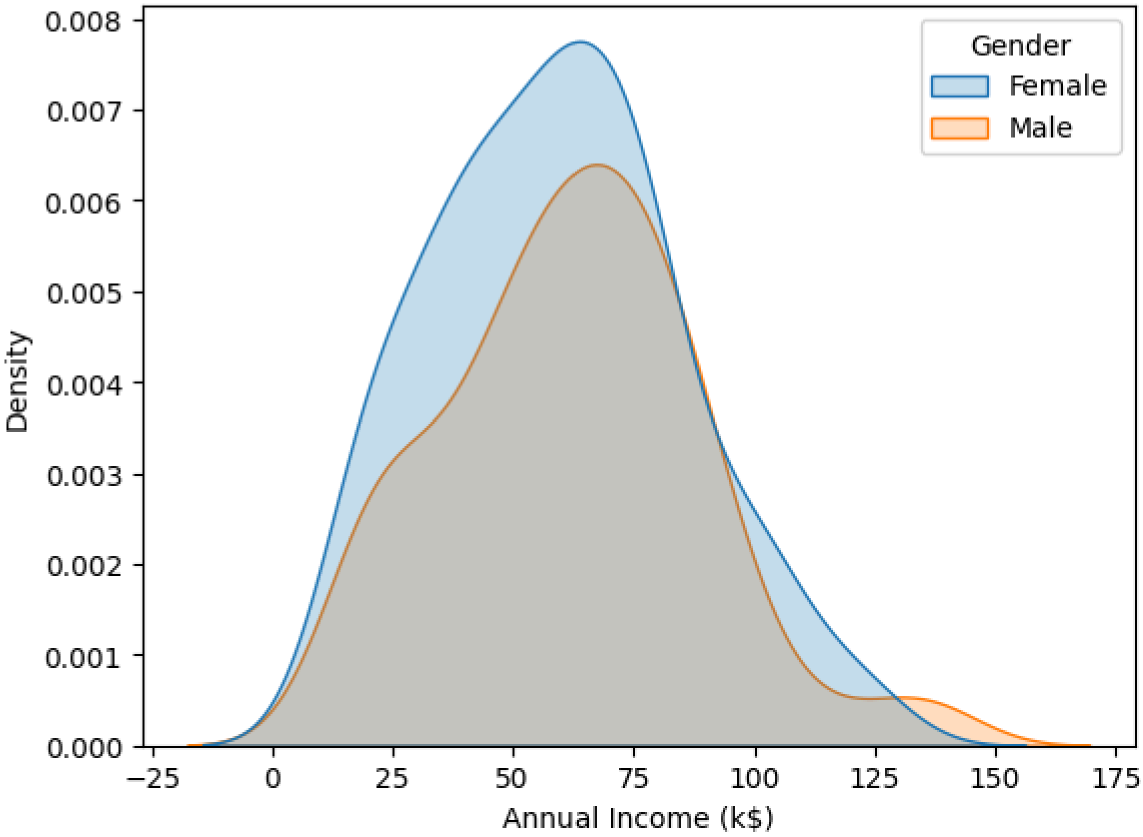
In [8]:

```python
# Convert 'Gender' column to categorical
df['Gender'] = pd.Categorical(df['Gender'])

# Plot KDE with 'Gender' as hue
sns.kdeplot(data=df, x='Annual Income (k$)', shade=True, hue='Gender')
```

Out[8]:

```
<Axes: xlabel='Annual Income (k$)', ylabel='Density'>
```

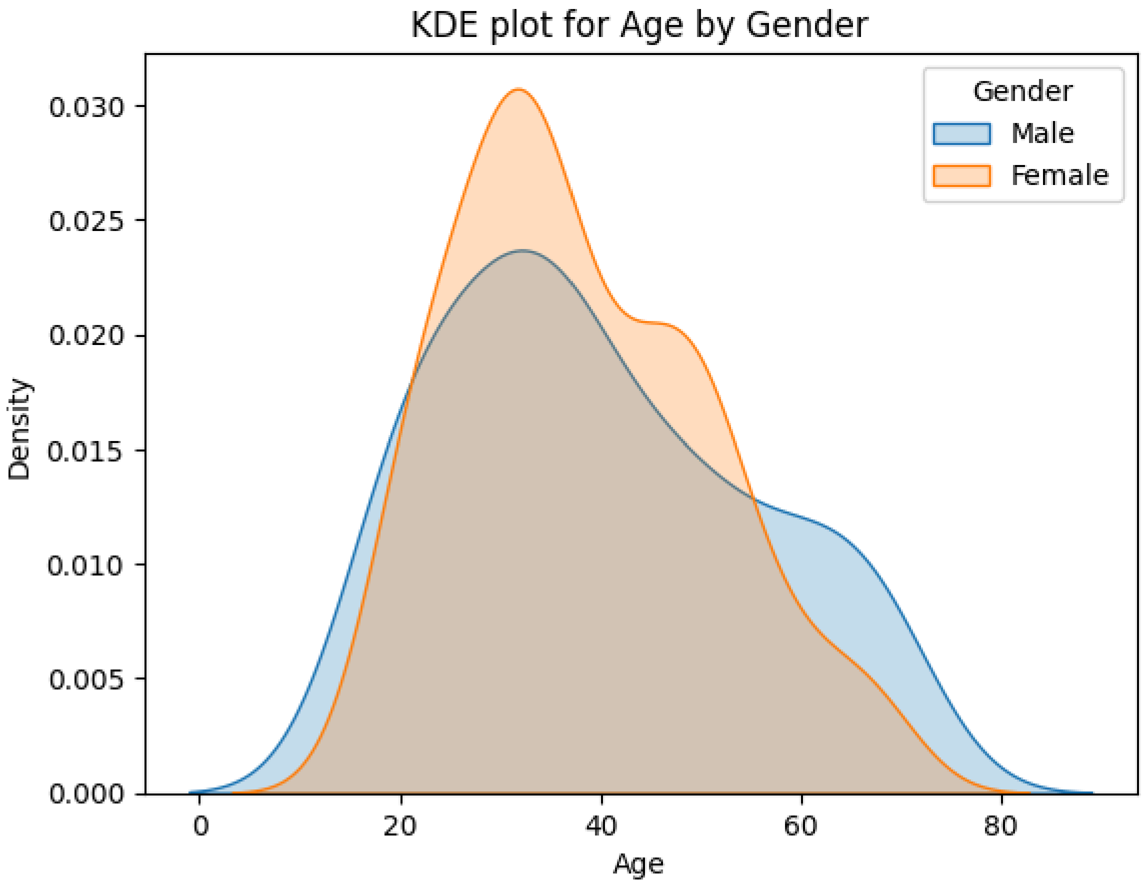


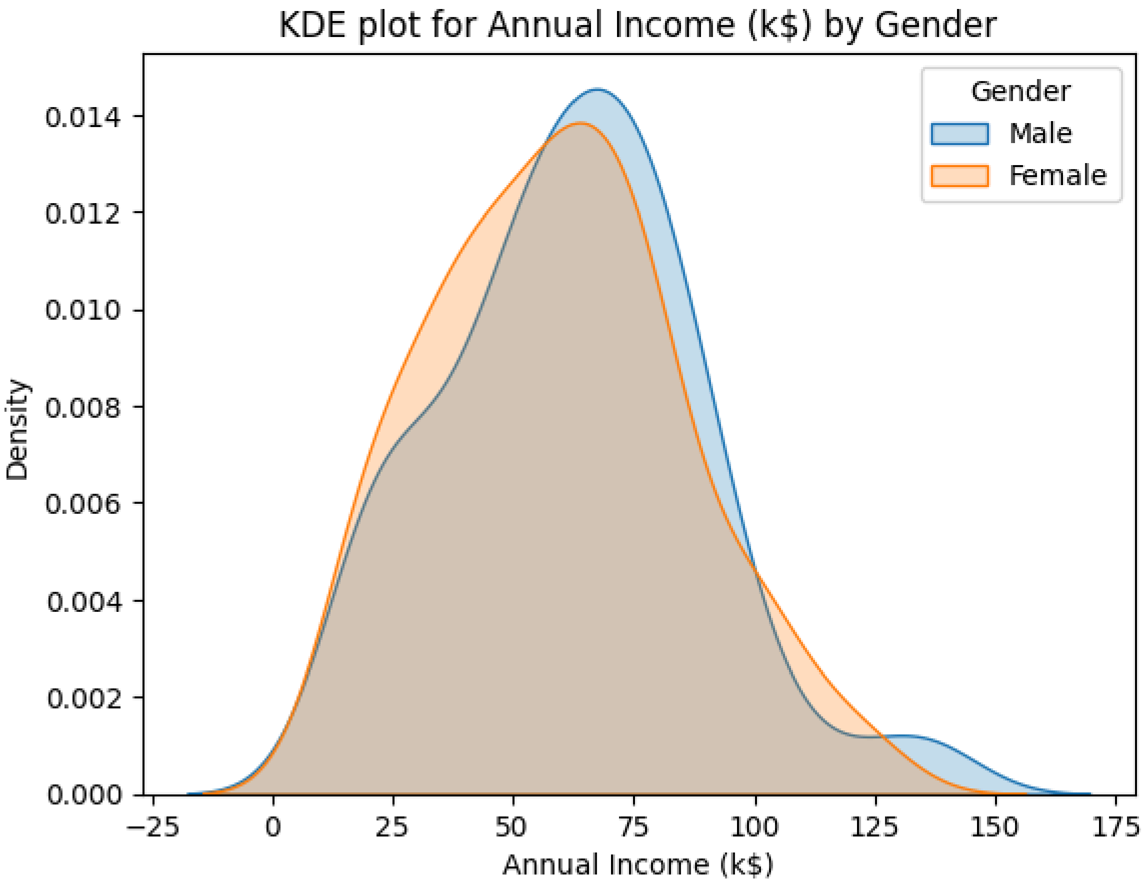
In [12]:

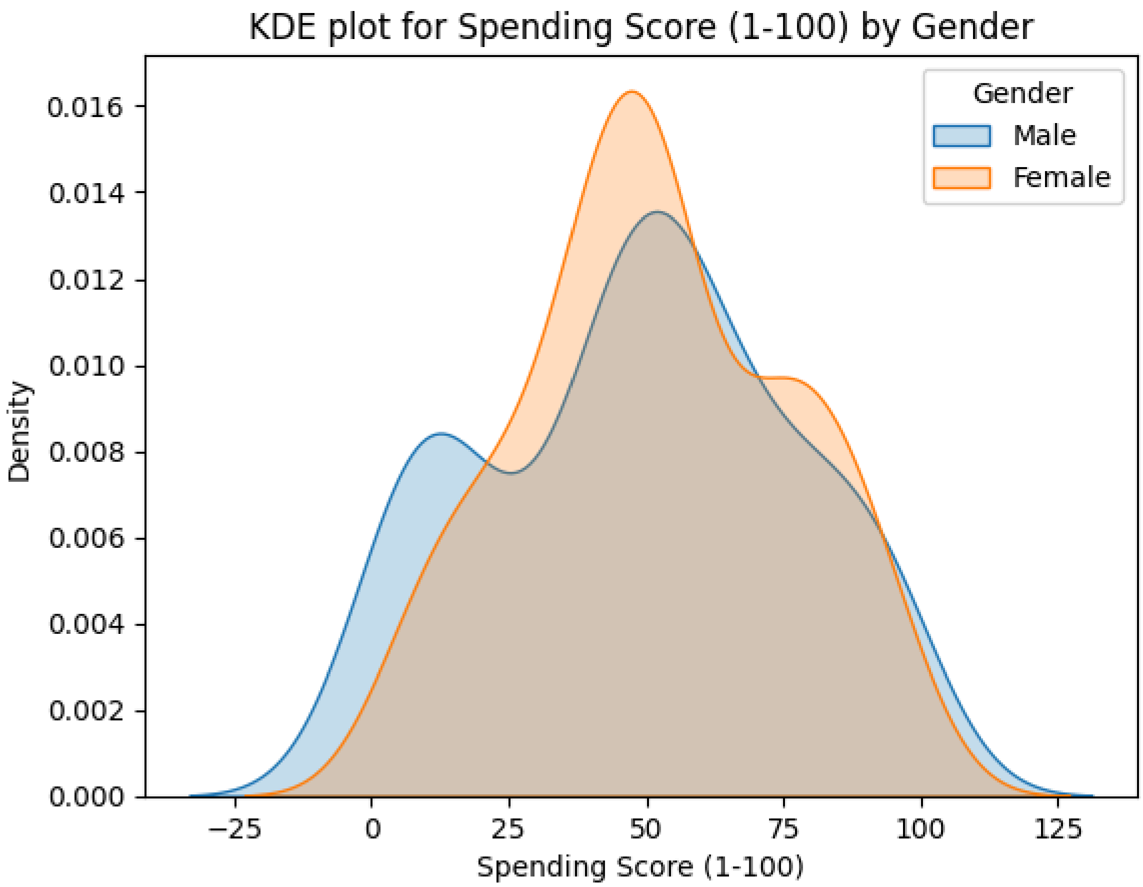
```
import seaborn as sns
import matplotlib.pyplot as plt

columns = ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']

for i in columns:
    plt.figure()
    for gender_category in df['Gender'].unique():
        sns.kdeplot(data=df[df['Gender'] == gender_category][i], shade=True, label=gender_category)
    plt.title(f'KDE plot for {i} by Gender')
    plt.legend(title='Gender')
    plt.xlabel(i)
    plt.ylabel('Density')
```

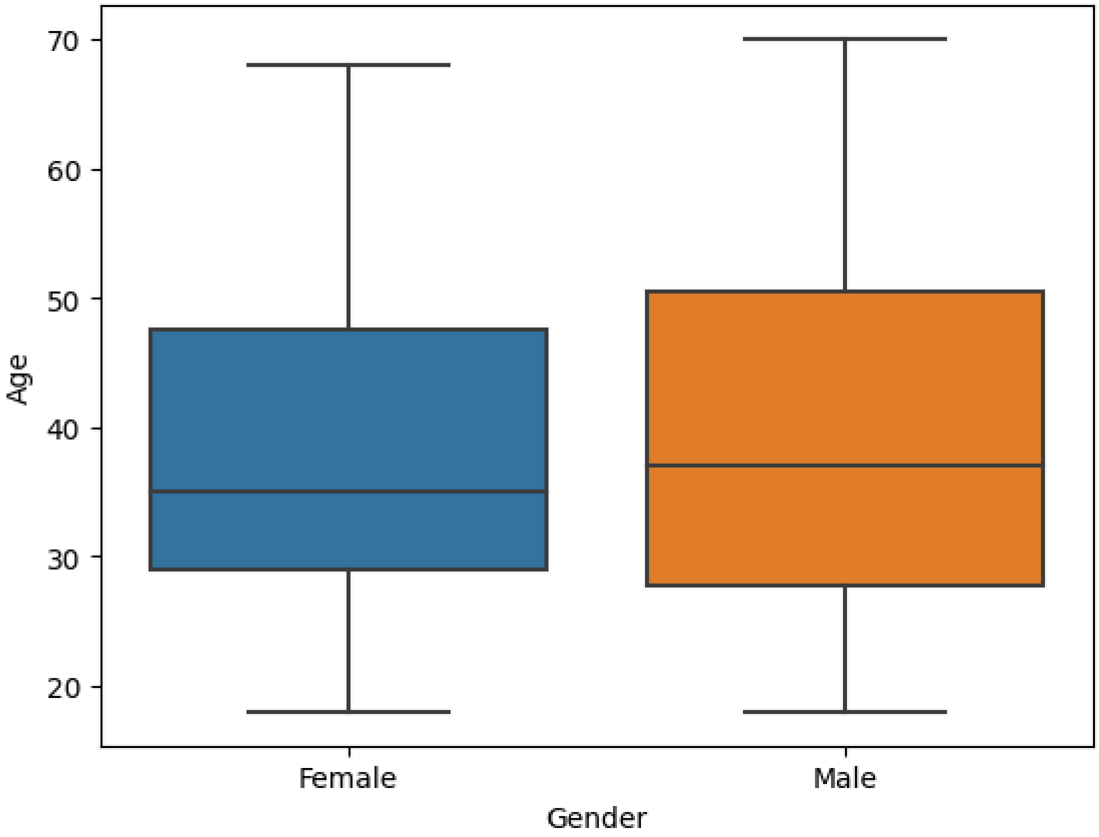

KDE plot for Age by Gender
Gender
Male
Female
Density
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
20
40
60
80
Age

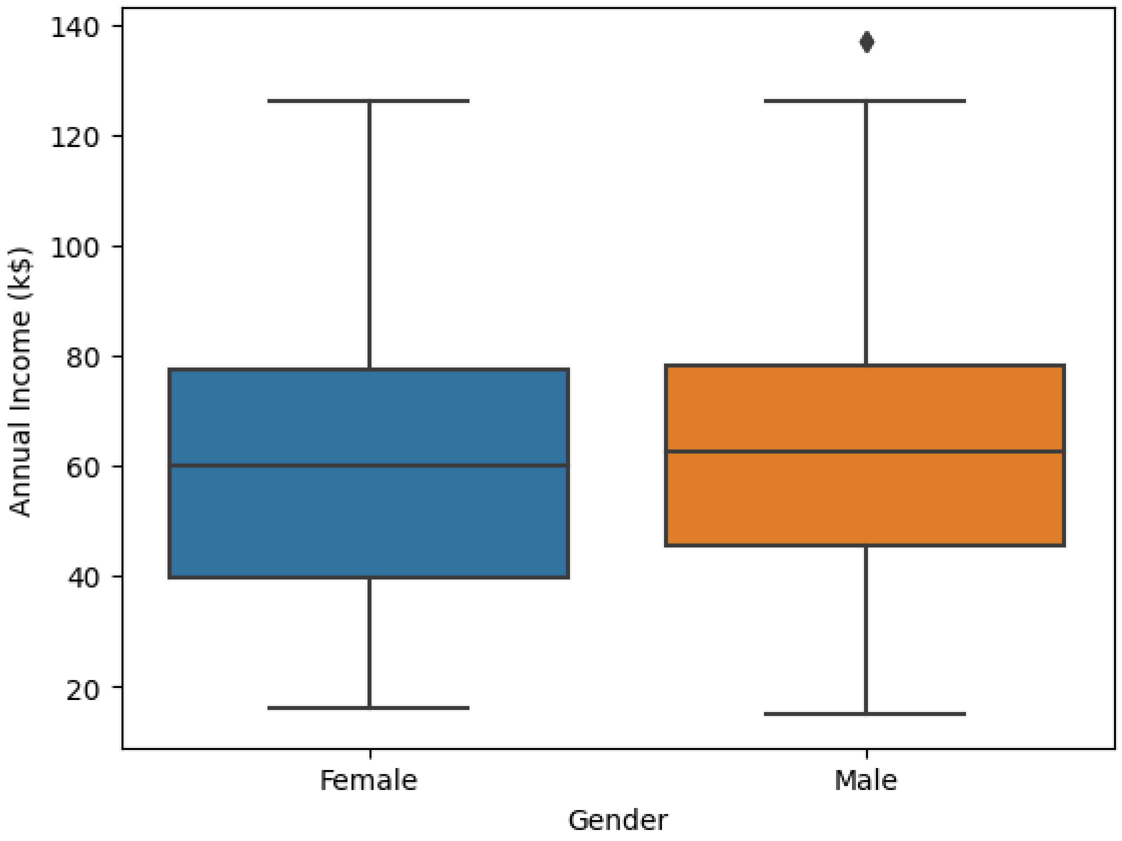

KDE plot for Annual Income (k$) by Gender
Density
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
−25
0
25
50
75
100
125
150
175
Annual Income (k$)
Gender
Male
Female

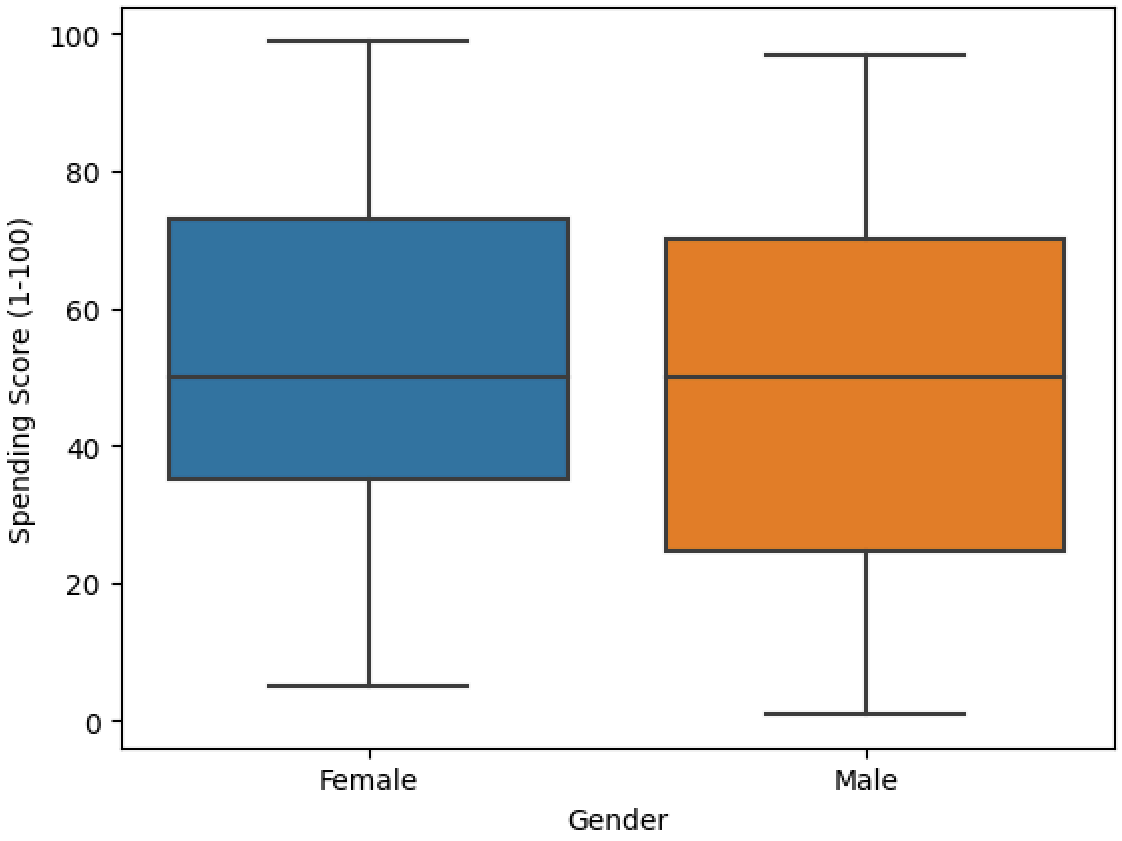


In [13]:

```
columns = ['Age', 'Annual Income (k$)','Spending Score (1-100)']
for i in columns:
    plt.figure()
    sns.boxplot(data=df,x='Gender',y=df[i])
```


70
60
50
40
30
20
Age
Female
Male
Gender


140
120
100
80
60
40
20
Annual Income (k$)
Female
Male
Gender



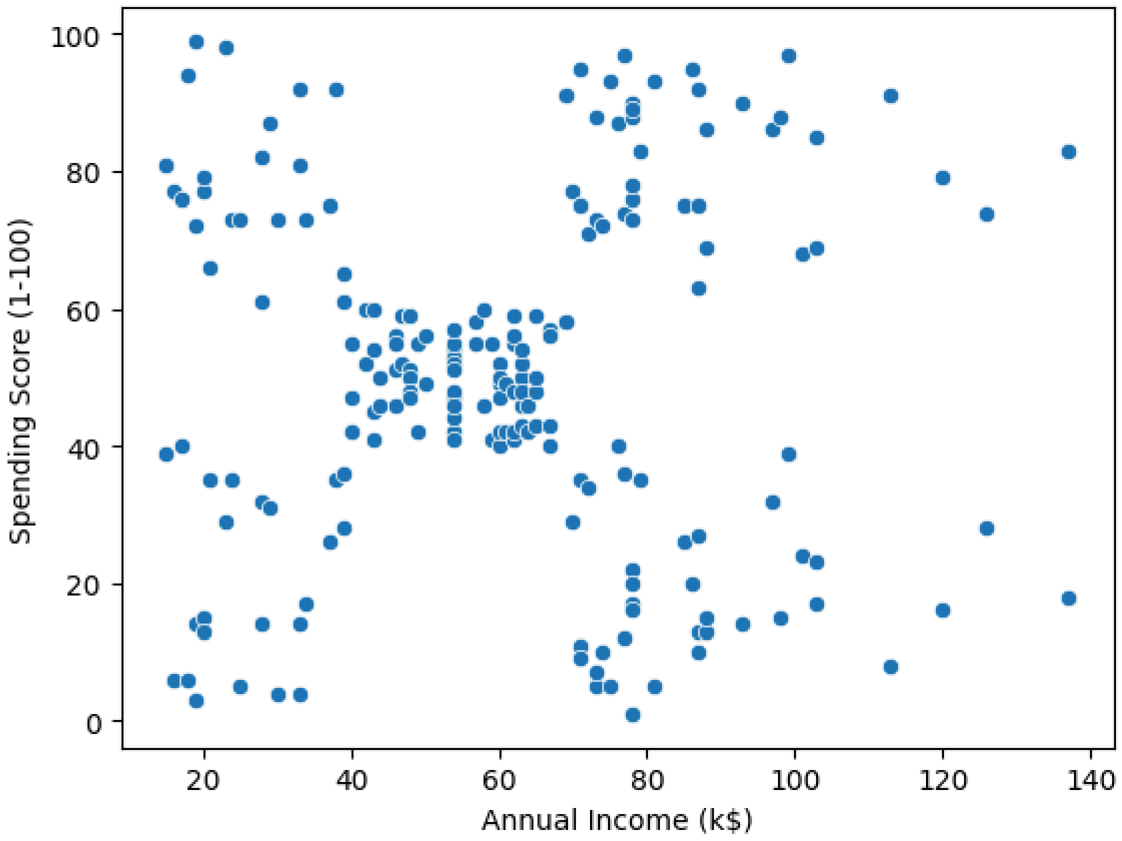
```
In [14]: df['Gender'].value_counts(normalize=True)
```

```
Out[14]: Female    0.56
         Male      0.44
         Name: Gender, dtype: float64
```

Bivariate Analysis

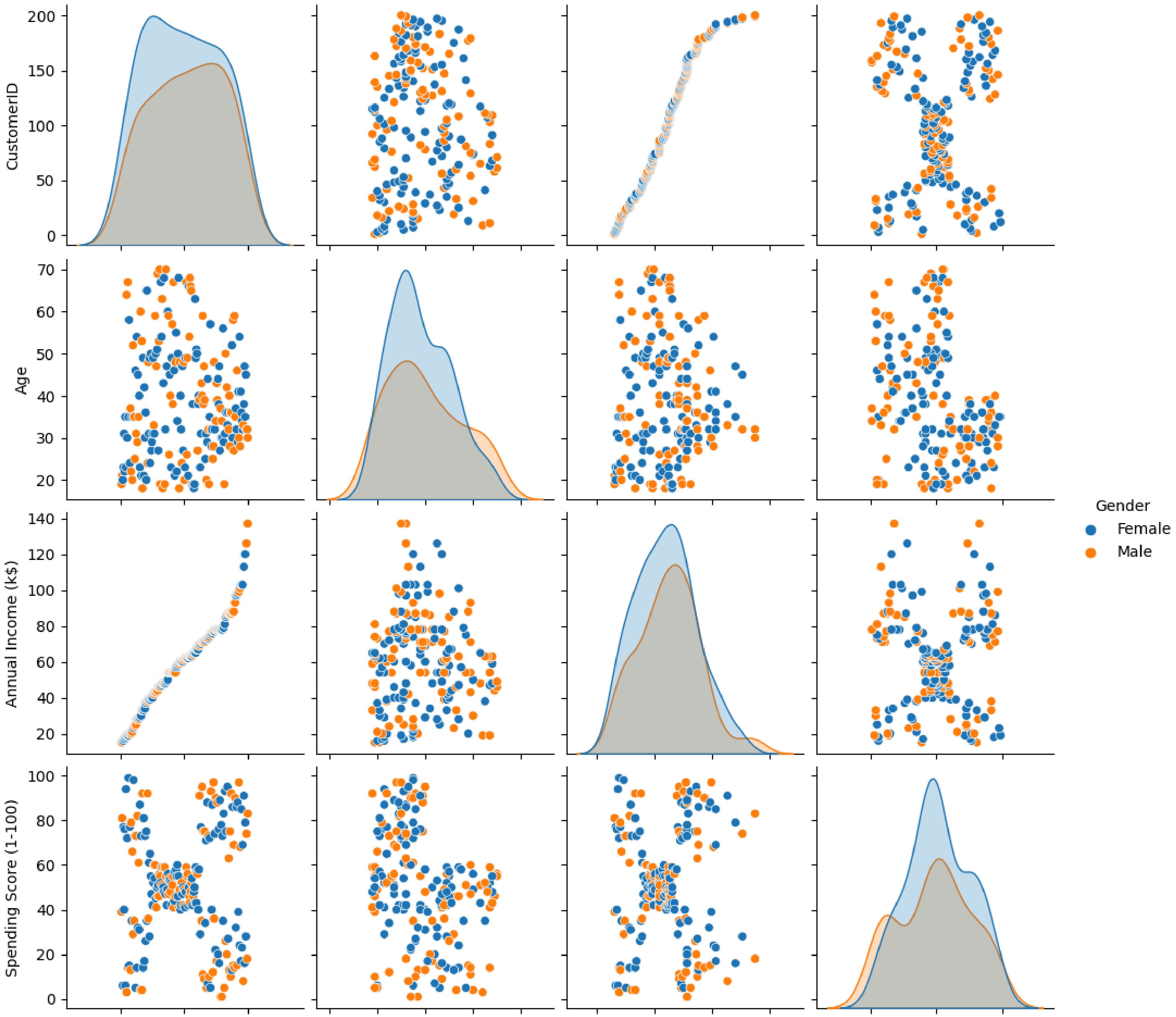
```
In [15]: sns.scatterplot(data=df, x='Annual Income (k$)',y='Spending Score (1-100)' )
```

```
Out[15]: <Axes: xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'>
```



In [16]:
```
#df=df.drop('CustomerID',axis=1)
sns.pairplot(df,hue='Gender')
```

Out[16]:
```
<seaborn.axisgrid.PairGrid at 0x1fc3ecb9fd0>
```


CustomerID
200
150
100
50
0
Age
70
60
50
40
30
20
Annual Income (k$)
140
120
100
80
60
40
20
Spending Score (1-100)
100
80
60
40
20
0
Gender
● Female
● Male

In [17]:

```
df.groupby(['Gender'])['Age', 'Annual Income (k$)',
       'Spending Score (1-100)'].mean()
```

Out[17]:

	Age	Annual Income (k$)	Spending Score (1-100)
Gender			
Female	38.098214	59.250000	51.526786
Male	39.806818	62.227273	48.511364

In [18]:

```
df.corr()
```

Out[18]:

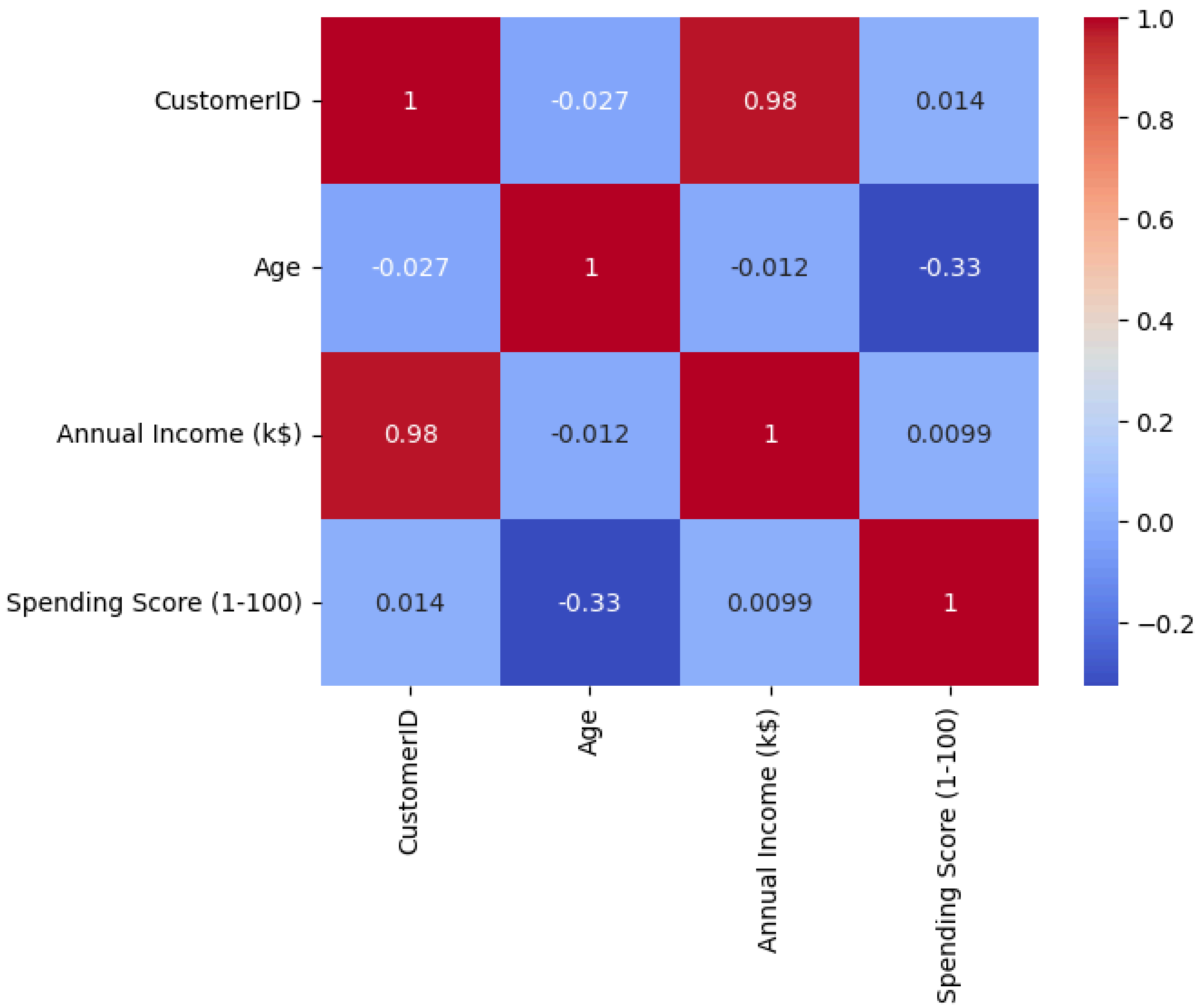
	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
CustomerID	1.000000	-0.026763	0.977548	0.013835
Age	-0.026763	1.000000	-0.012398	-0.327227
Annual Income (k$)	0.977548	-0.012398	1.000000	0.009903
Spending Score (1-100)	0.013835	-0.327227	0.009903	1.000000

In [19]:

```
sns.heatmap(df.corr(),annot=True,cmap='coolwarm')
```

Out[19]:

```
<Axes: >
```



Clustering - Univariate, Bivariate, Multivariate

```
In [20]: clustering1 = KMeans(n_clusters=3)
```

```
In [21]: clustering1.fit(df[['Annual Income (k$)']])
```

Out[21]:

```
▾ KMeans
KMeans(n_clusters=3)
```

In [22]:

```
clustering1.labels_
```

Out[22]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 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, 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, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2])
```

In [23]:

```
df['Income Cluster'] = clustering1.labels_
df.head()
```

Out[23]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	Income Cluster
0	1	Male	19	15	39	0
1	2	Male	21	15	81	0
2	3	Female	20	16	6	0
3	4	Female	23	16	77	0
4	5	Female	31	17	40	0

In [24]:

```
df['Income Cluster'].value_counts()
```

Out[24]:

```
1    104
0     74
2     22
Name: Income Cluster, dtype: int64
```

In [25]:

```
clustering1.inertia_
```

Out[25]:

```
24361.25921375922
```

In [26]:

```
intertia_scores=[]
for i in range(1,11):
    kmeans=KMeans(n_clusters=i)
    kmeans.fit(df[['Annual Income (k$)']])
    intertia_scores.append(kmeans.inertia_)
```

In [27]:

```
intertia_scores
```

Out[27]:

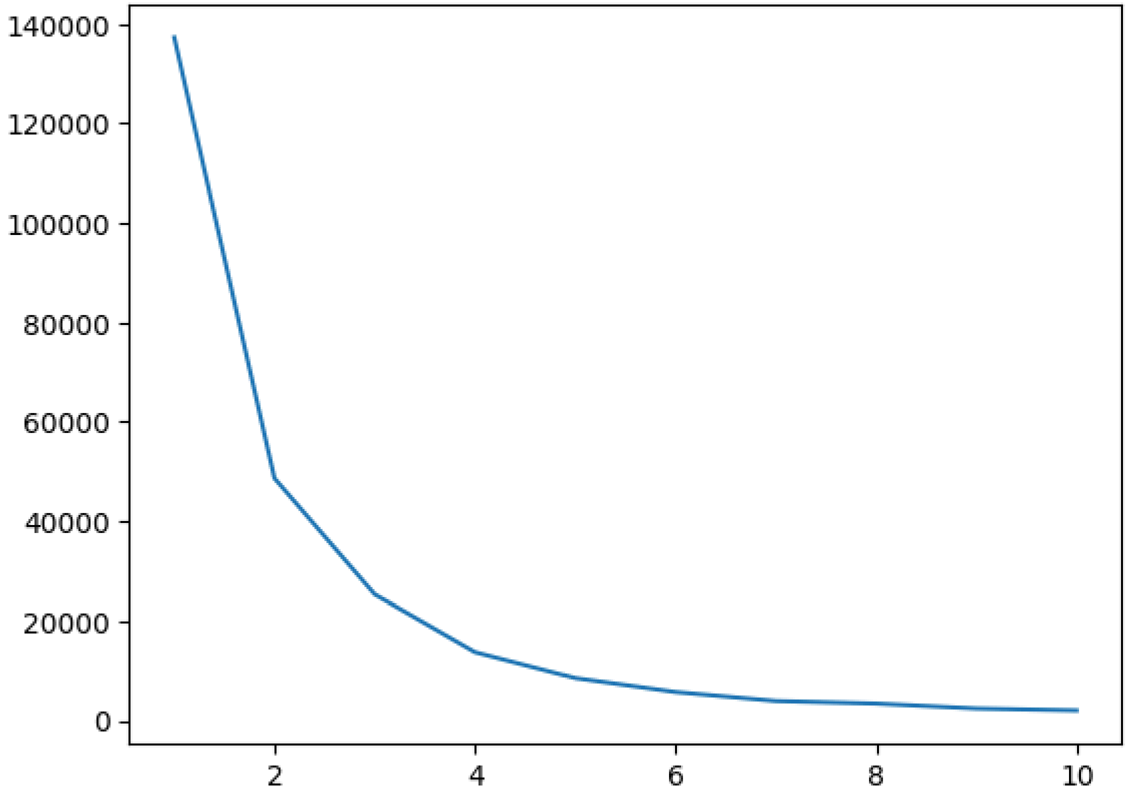
```
[137277.28000000003,
 48660.88888888889,
 25341.285871863227,
 13757.071717171717,
 8534.41515455305,
 5728.855832763727,
 3931.9880952380945,
 3413.6828834907787,
 2420.9949328449325,
 2035.475968475968]
```

In [28]:

```
plt.plot(range(1,11),intertia_scores)
```

Out[28]:

```
[<matplotlib.lines.Line2D at 0x1fc3e560b50>]
```



```
In [29]: df.columns
```

```
Out[29]: Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
                'Spending Score (1-100)', 'Income Cluster'],
               dtype='object')
```

```
In [30]: df.groupby('Income Cluster')['Age', 'Annual Income (k$)',
                'Spending Score (1-100)'].mean()
```

Out[30]:

	Age	Annual Income (k$)	Spending Score (1-100)
Income Cluster			
0	39.500000	33.486486	50.229730
1	38.663462	69.750000	49.798077
2	37.545455	108.181818	52.000000

In [31]:

```
#Bivariate Clustering
```

In [32]:

```
clustering2 = KMeans(n_clusters=5)
clustering2.fit(df[['Annual Income (k$)','Spending Score (1-100)']])
df['Spending and Income Cluster'] =clustering2.labels_
df.head()
```

Out[32]:

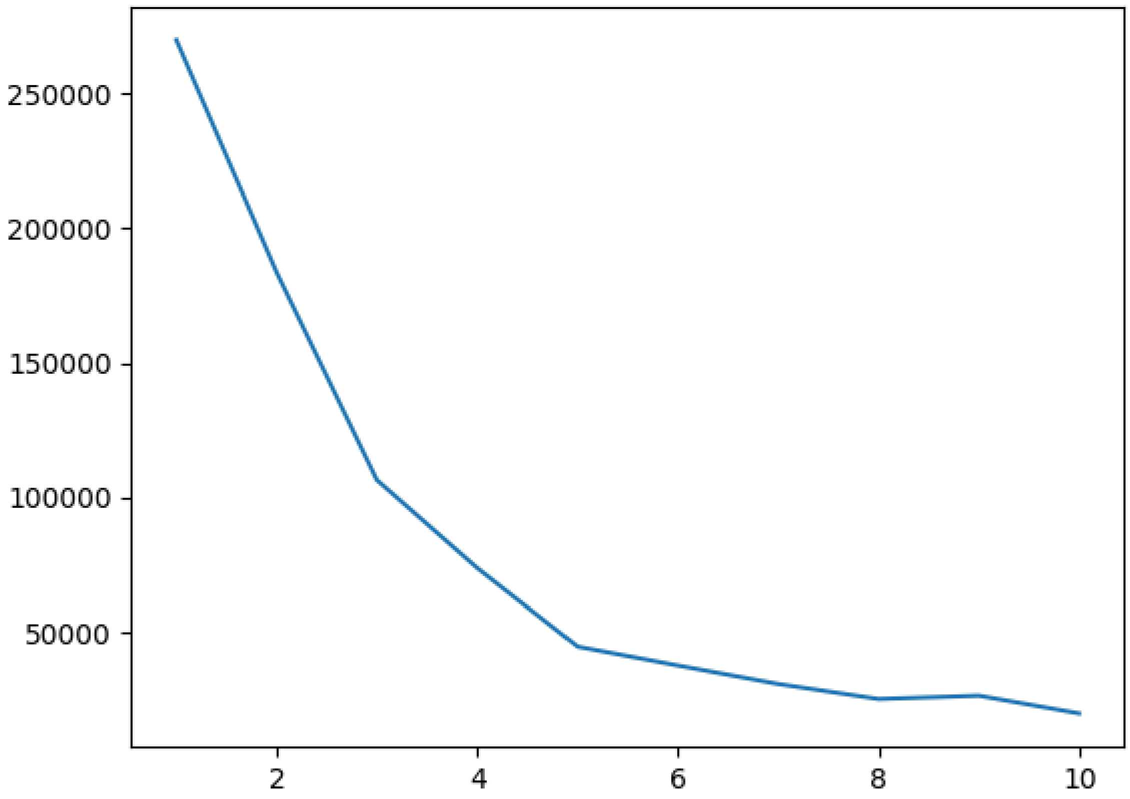
	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	Income Cluster	Spending and Income Cluster
0	1	Male	19	15	39	0	4
1	2	Male	21	15	81	0	0
2	3	Female	20	16	6	0	4
3	4	Female	23	16	77	0	0
4	5	Female	31	17	40	0	4

In [33]:

```
intertia_scores2=[]
for i in range(1,11):
    kmeans2=KMeans(n_clusters=i)
    kmeans2.fit(df[['Annual Income (k$)','Spending Score (1-100)']])
    intertia_scores2.append(kmeans2.inertia_)
plt.plot(range(1,11),intertia_scores2)
```

Out[33]:

```
[<matplotlib.lines.Line2D at 0x1fc3e451190>]
```

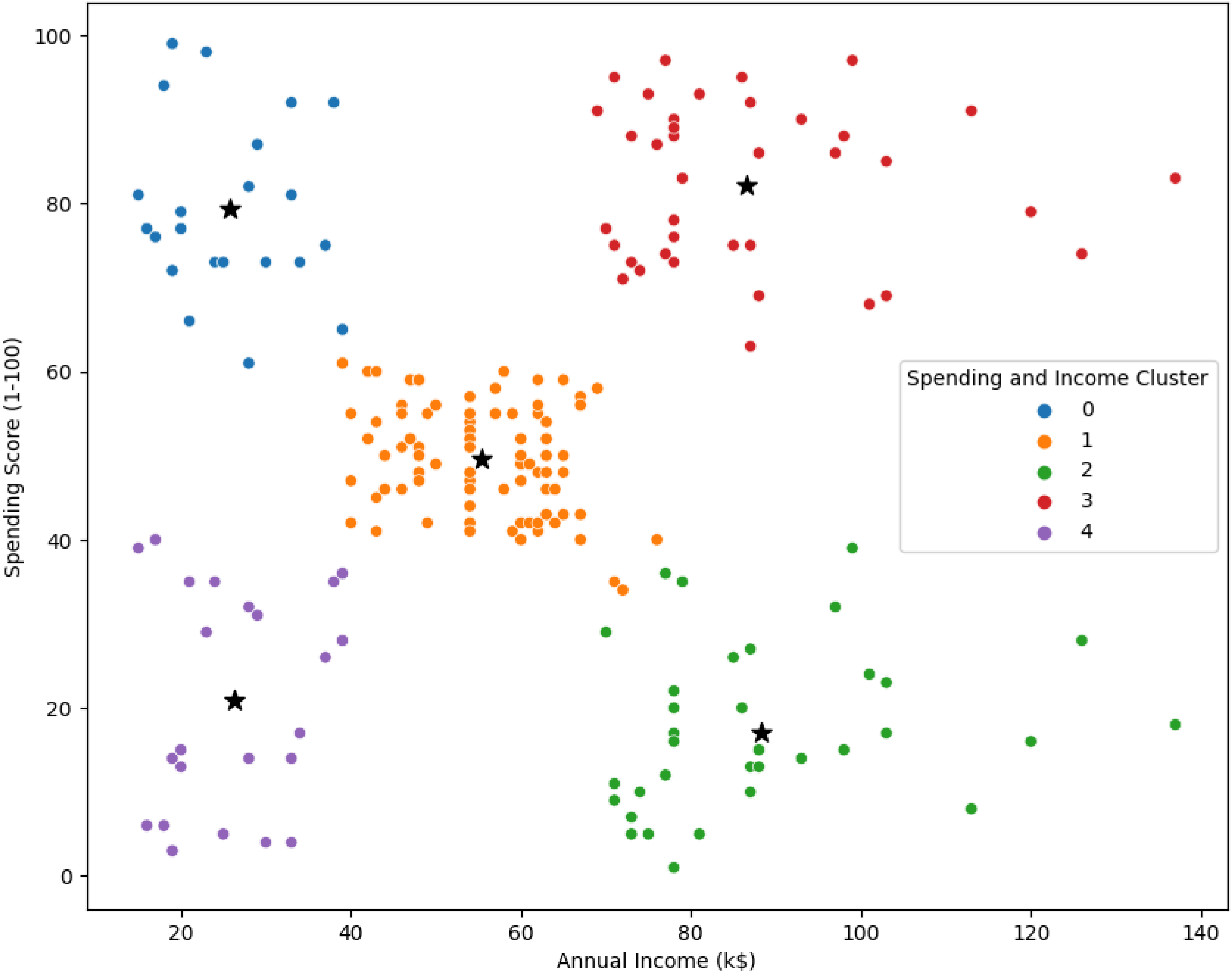


In [34]:

```
centers =pd.DataFrame(clustering2.cluster_centers_)
centers.columns = ['x','y']
```

In [35]:

```
plt.figure(figsize=(10,8))
plt.scatter(x=centers['x'],y=centers['y'],s=100,c='black',marker='*')
sns.scatterplot(data=df, x ='Annual Income (k$)',y='Spending Score (1-100)',hue='Spending and Income Cluster',palette
plt.savefig('clustering_bivaraiate.png')
```



```
In [36]: pd.crosstab(df['Spending and Income Cluster'],df['Gender'],normalize='index')
```

Out[36]:

Gender	Female	Male
Spending and Income Cluster		
0	0.590909	0.409091
1	0.592593	0.407407
2	0.457143	0.542857
3	0.538462	0.461538
4	0.608696	0.391304

In [37]:

```python
df.groupby('Spending and Income Cluster')['Age', 'Annual Income (k$)',
       'Spending Score (1-100)'].mean()
```

Out[37]:

	Age	Annual Income (k$)	Spending Score (1-100)
Spending and Income Cluster			
0	25.272727	25.727273	79.363636
1	42.716049	55.296296	49.518519
2	41.114286	88.200000	17.114286
3	32.692308	86.538462	82.128205
4	45.217391	26.304348	20.913043

In [38]:

```python
#mulivariate clustering
from sklearn.preprocessing import StandardScaler
```

In [39]:

```python
scale = StandardScaler()
```

In [40]:

```python
df.head()
```

Out[40]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	Income Cluster	Spending and Income Cluster
0	1	Male	19	15	39	0	4
1	2	Male	21	15	81	0	0
2	3	Female	20	16	6	0	4
3	4	Female	23	16	77	0	0
4	5	Female	31	17	40	0	4

In [41]:

```
dff = pd.get_dummies(df,drop_first=True)
dff.head()
```

Out[41]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)	Income Cluster	Spending and Income Cluster	Gender_Male
0	1	19	15	39	0	4	1
1	2	21	15	81	0	0	1
2	3	20	16	6	0	4	0
3	4	23	16	77	0	0	0
4	5	31	17	40	0	4	0

In [42]:

```
dff.columns
```

Out[42]:

```
Index(['CustomerID', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)',
       'Income Cluster', 'Spending and Income Cluster', 'Gender_Male'],
      dtype='object')
```

In [43]:

```
dff = dff[['Age', 'Annual Income (k$)', 'Spending Score (1-100)','Gender_Male']]
dff.head()
```

Out[43]:

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

In [44]:

```
dff = scale.fit_transform(dff)
```

In [45]:

```
dff = pd.DataFrame(scale.fit_transform(dff))
dff.head()
```

Out[45]:

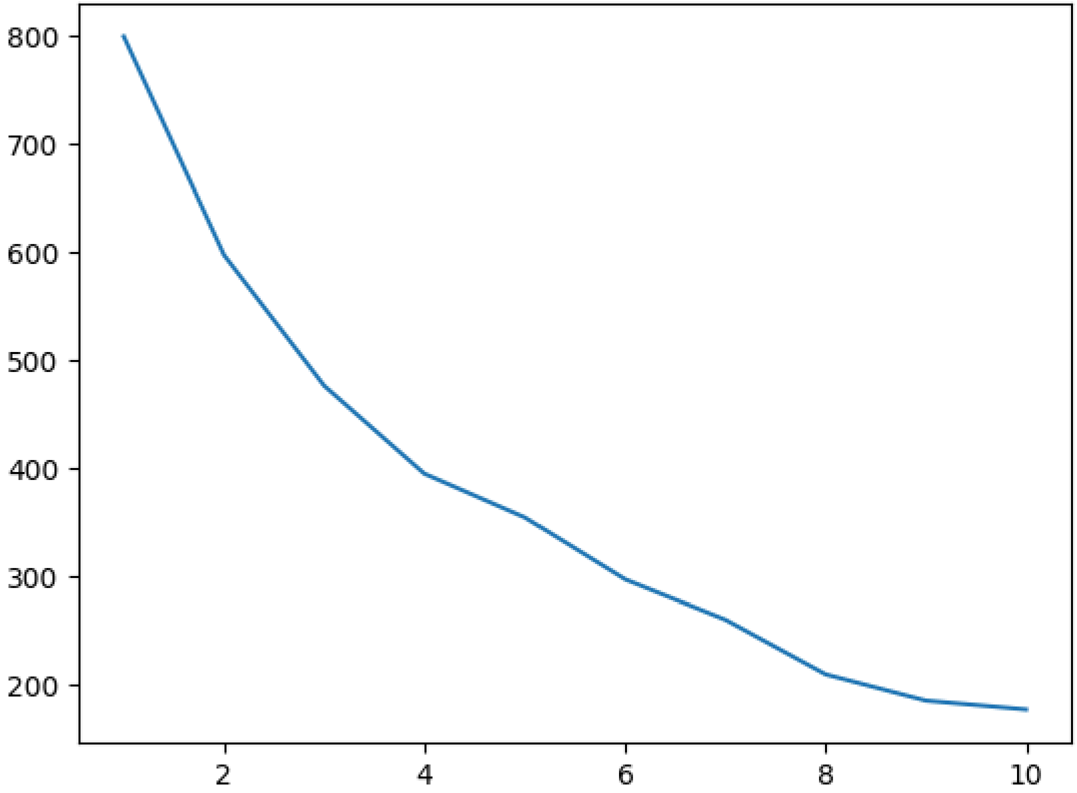
	0	1	2	3
0	-1.424569	-1.738999	-0.434801	1.128152
1	-1.281035	-1.738999	1.195704	1.128152
2	-1.352802	-1.700830	-1.715913	-0.886405
3	-1.137502	-1.700830	1.040418	-0.886405
4	-0.563369	-1.662660	-0.395980	-0.886405

In [46]:

```
intertia_scores3=[]
for i in range(1,11):
    kmeans3=KMeans(n_clusters=i)
    kmeans3.fit(dff)
    intertia_scores3.append(kmeans3.inertia_)
plt.plot(range(1,11),intertia_scores3)
```

Out[46]:

```
[<matplotlib.lines.Line2D at 0x1fc3e4e3d10>]
```



```
In [47]: df
```

Out[47]:

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

200 rows × 7 columns

In [48]:

```
df.to_csv('Clustering.csv')
```