SHOPPING CUSTOMER SEGMENTATION

Mall Custormers where segmented using unsupervised machine learning in the form of clustering.

Problem Statement: Understand the target customers for the marketing team to plan a strategy.

Approach

-Perform quick Exploration Data Analysis
-Use KMeans clustering algorithm to create our segments
-Use summary statistics on the cluster
-Visualize

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
df= pd.read_csv("C:\\Users\\Joseph\\Desktop\\Mall_Customers.csv")
df.head(10)
```

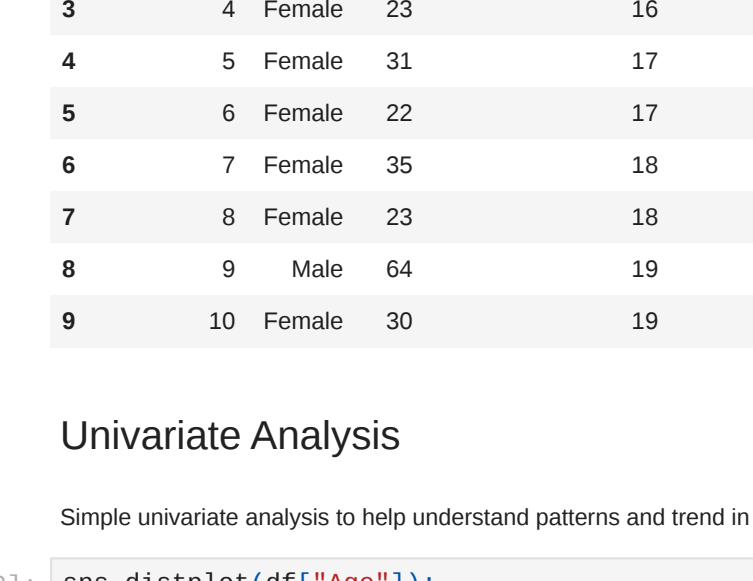
Out[2]:

	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
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72

Univariate Analysis

Simple univariate analysis to help understand patterns and trend in the data

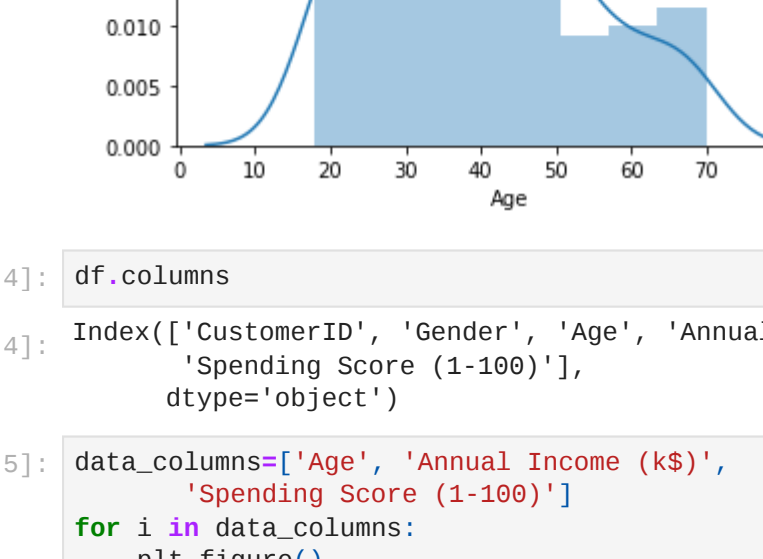
In [3]:
```
sns.distplot(df["Age"]);
```

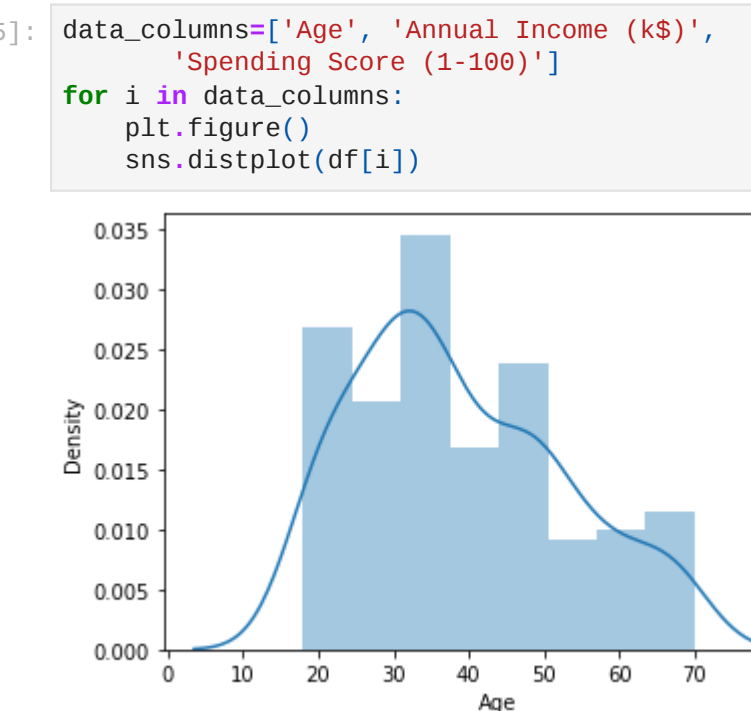


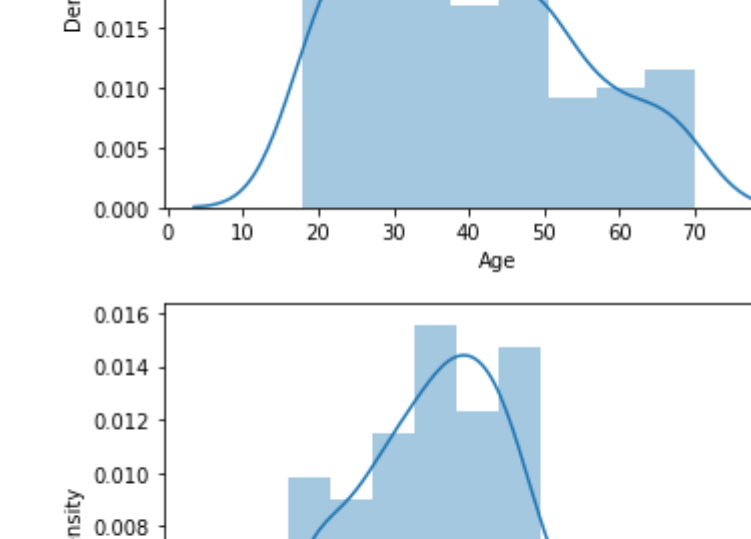
In [4]:
```
df.columns
```

Out[4]:
```
Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

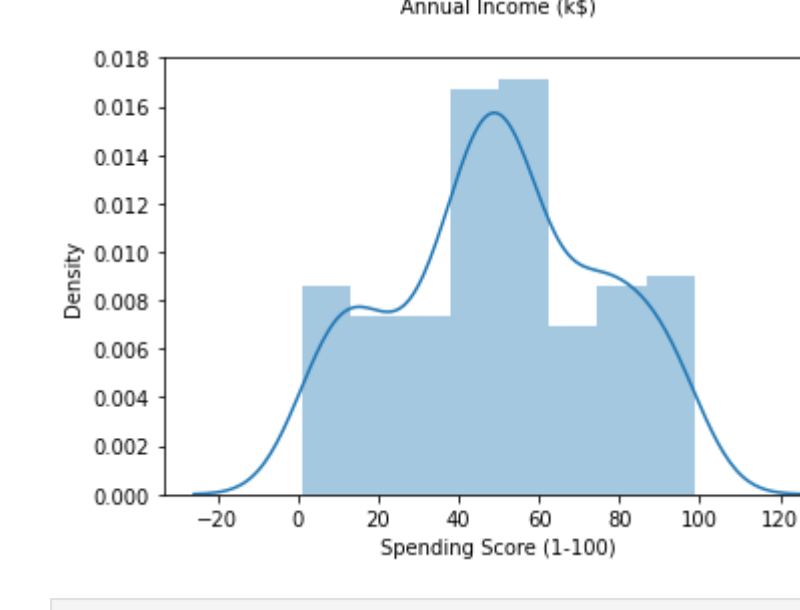
In [5]:
```
data_columns=['Age', 'Annual Income (k$)',
      'Spending Score (1-100)']
for i in data_columns:
    plt.figure()
    sns.distplot(df[i])
```



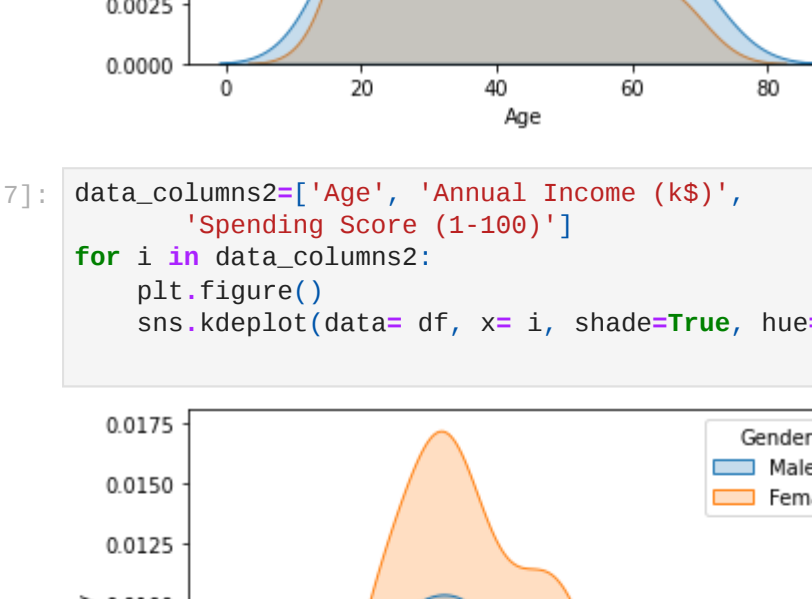


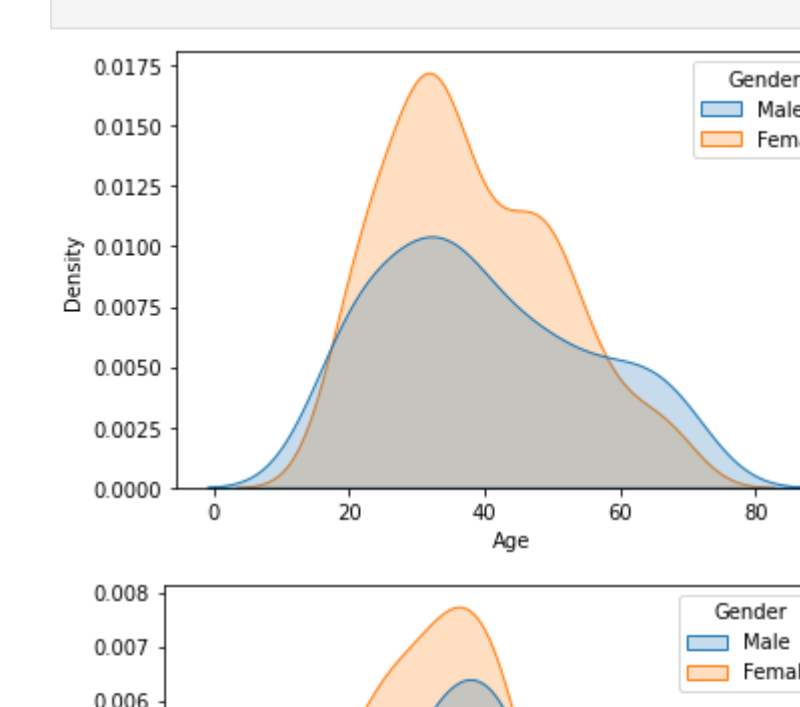


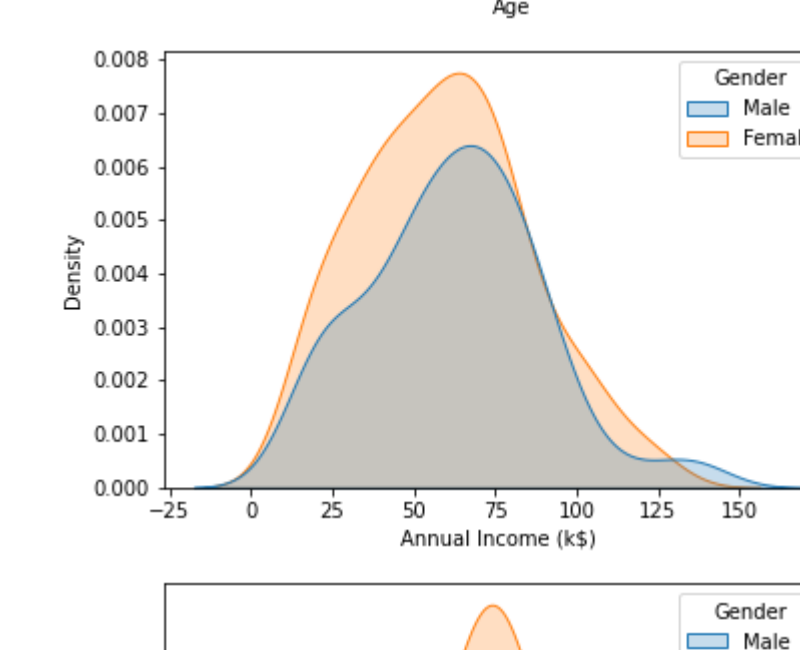
In [6]:
```
sns.kdeplot(data= df, x= "Age", shade= True, hue= "Gender");
```



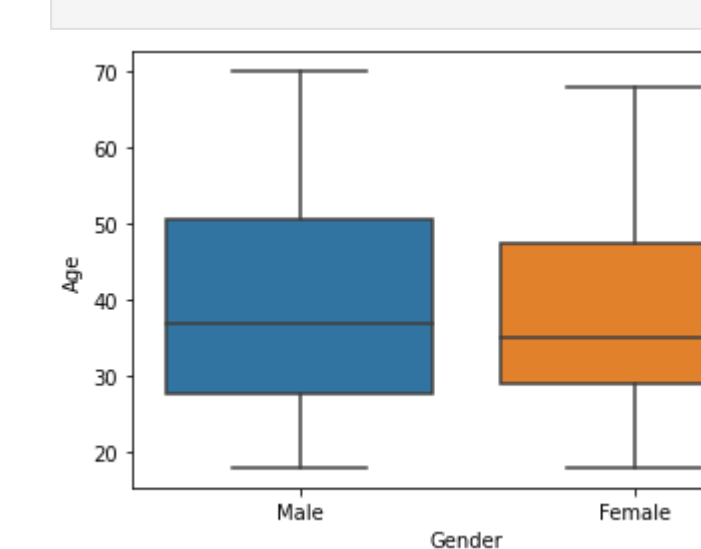
In [7]:
```
data_columns2=['Age', 'Annual Income (k$)',
      'Spending Score (1-100)']
for i in data_columns2:
    plt.figure()
    sns.kdeplot(data= df, x= i, shade=True, hue= "Gender")
```

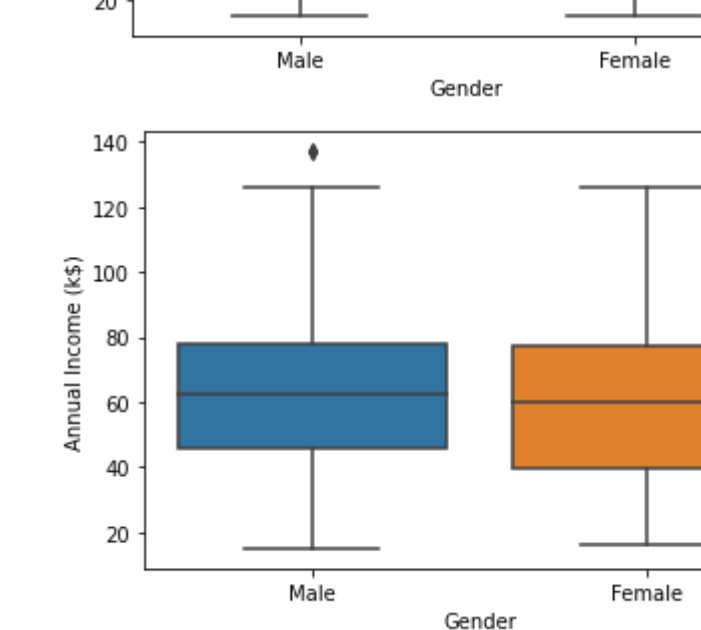


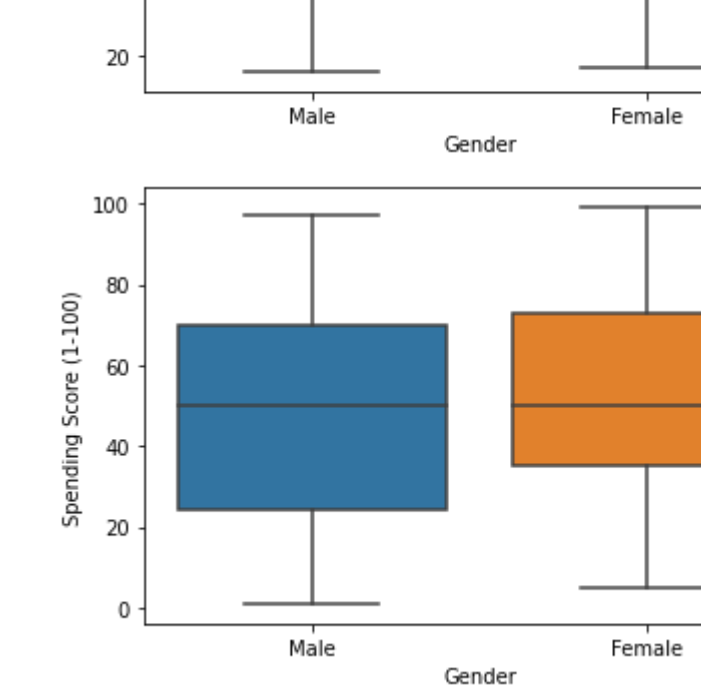




In [8]:
```
data_columns2=['Age', 'Annual Income (k$)',
      'Spending Score (1-100)']
for i in data_columns2:
    plt.figure()
    sns.boxplot(data= df, x= "Gender", y= i)
```

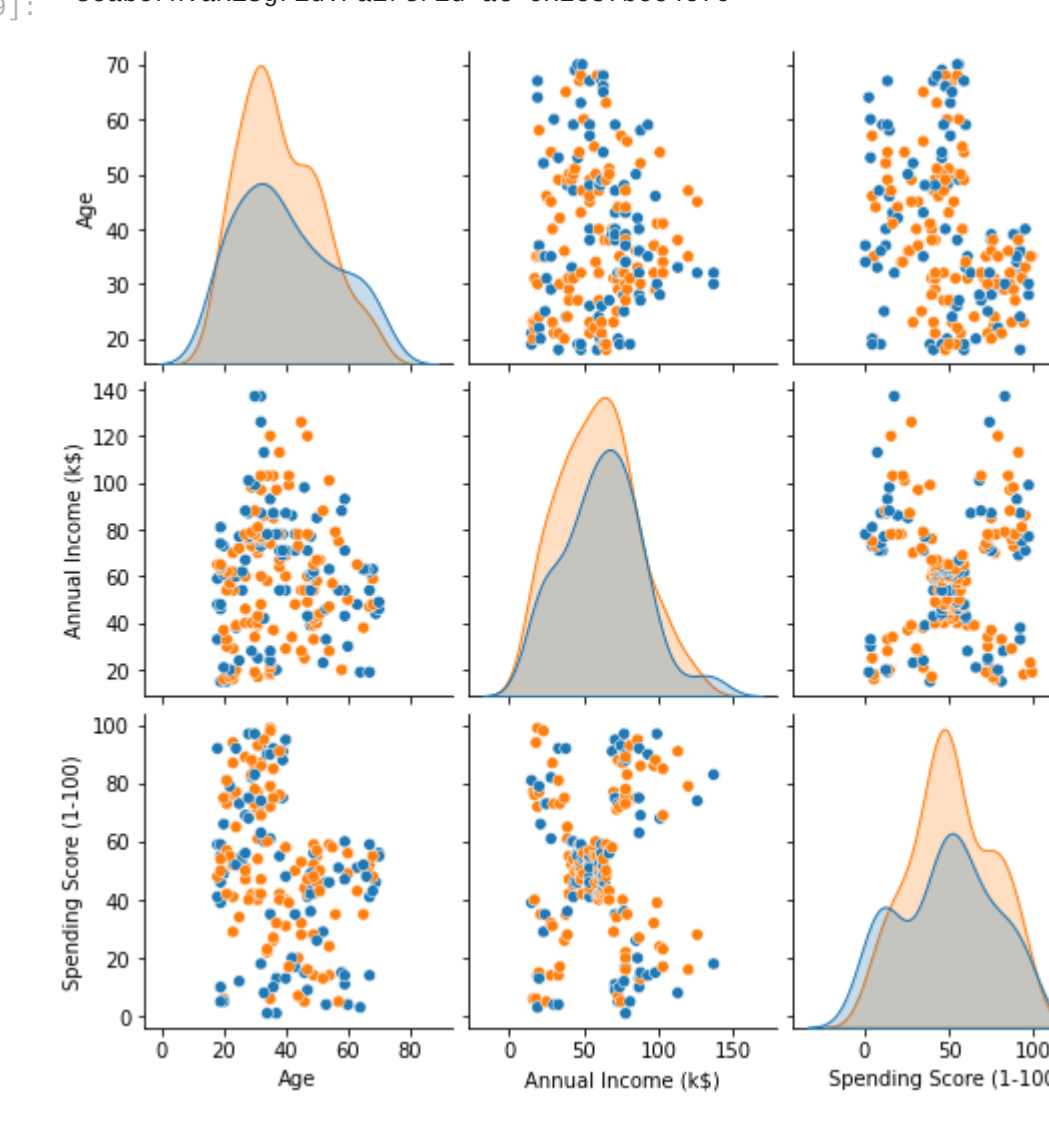






In [9]:
```
df2= df.drop('CustomerID', axis=1)
sns.pairplot(df2, hue= "Gender")
#df.columns
```

Out[9]: <seaborn.axisgrid.PairGrid at 0x2c87bcc4c70>



Group dataset by gender and find the mean values

In [10]:
```
df.groupby("Gender")['Age', 'Annual Income (k$)','Spending Score (1-100)'].mean()
```

Out[10]:

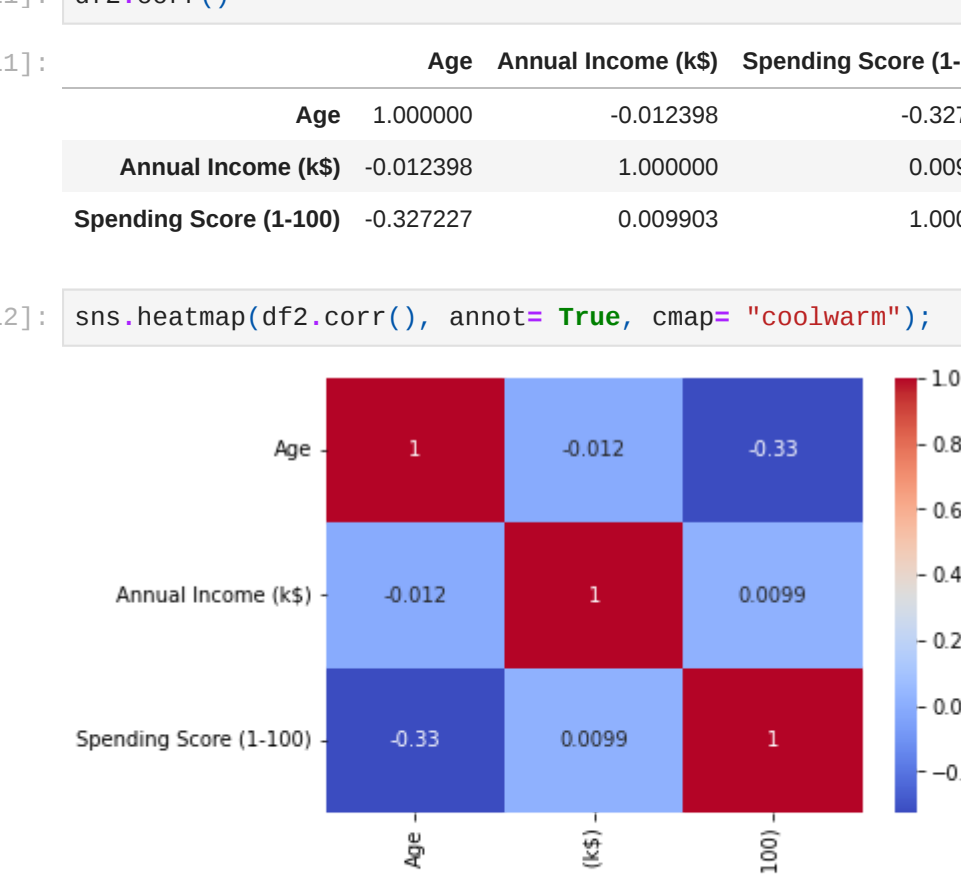
Gender	Age	Annual Income (k$)	Spending Score (1-100)
Female	38.098214	59.250000	51.526786
Male	39.806818	62.227273	48.511364

In [11]:
```
df2.corr()
```

Out[11]:

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

In [12]:
```
sns.heatmap(df2.corr(), annot= True, cmap= "coolwarm");
```



Clustering: Univariate, Bivariate, Multivariate

Univariate

In [13]:
```
cluster1= KMeans(n_clusters= 3)
```

In [14]:
```
cluster1.fit(df[["Annual Income (k$)"]])
```

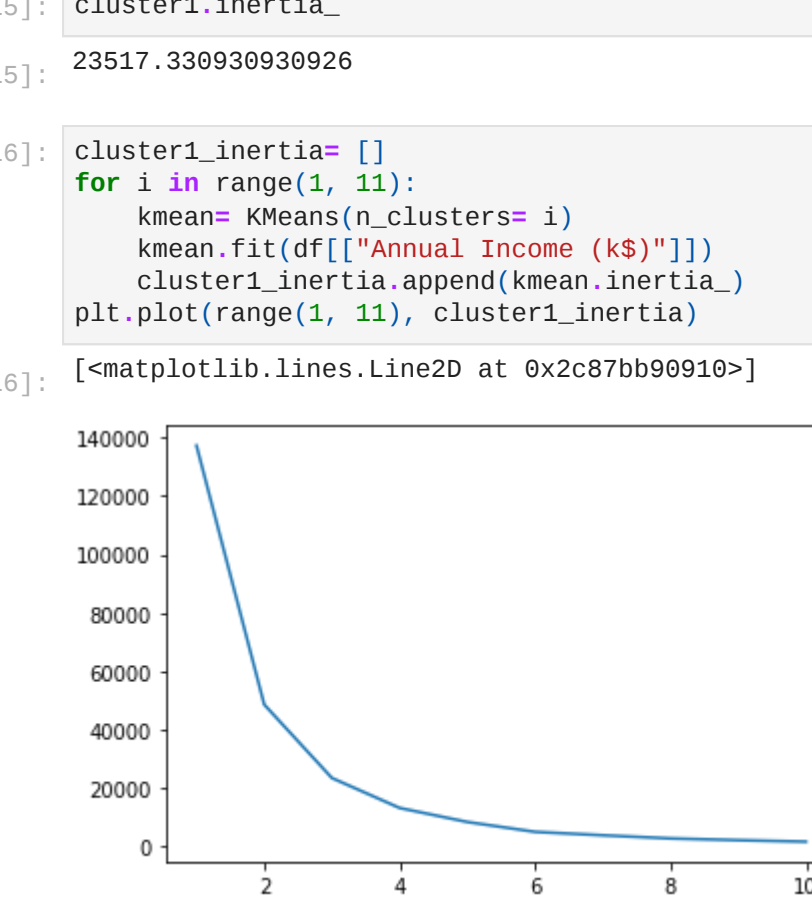
Out[14]:
```
KMeans
KMeans(n_clusters=3)
```

In [15]:
```
cluster1.inertia_
```

Out[15]: 23517.330930930926

In [16]:
```
cluster1_inertia= []
for i in range(1, 11):
    kmean= KMeans(n_clusters= i)
    kmean.fit(df[["Annual Income (k$)"]])
    cluster1_inertia.append(kmean.inertia_)
plt.plot(range(1, 11), cluster1_inertia)
```

Out[16]: [<matplotlib.lines.Line2D at 0x2c87bb90910>]



In [17]:
```
df2["Income Cluster"]= cluster1.labels_
```

In [18]:
```
df2.head()
```

Out[18]:

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

In [19]:
```
df2.groupby("Income Cluster")["Age", "Annual Income (k$)", "Spending Score (1-100)"].mean()
```

Out[19]:

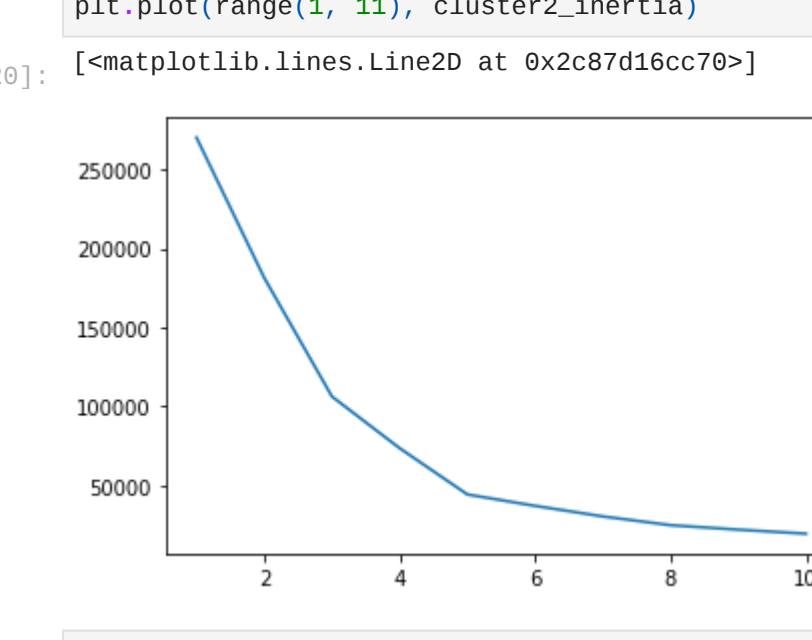
Income Cluster	Age	Annual Income (k$)	Spending Score (1-100)
0	39.500000	33.486486	50.229730
1	38.722222	67.088889	50.000000
2	37.833333	99.888889	50.638889

Bivariate

Plot a graph to check the number of cluster to use, using the elbow method

In [20]:
```
cluster2_inertia= []
for i in range(1,11):
    kmean= KMeans(n_clusters= i)
    kmean.fit(df[["Annual Income (k$)", "Spending Score (1-100)"]])
    cluster2_inertia.append(kmean.inertia_)
plt.plot(range(1, 11), cluster2_inertia)
```

Out[20]: [<matplotlib.lines.Line2D at 0x2c87d16cc70>]



In [21]:
```
cluster2= KMeans(n_clusters= 5)
```

In [22]:
```
cluster2.fit(df[["Annual Income (k$)", "Spending Score (1-100)"]])
```

Out[22]:
```
KMeans
KMeans(n_clusters=5)
```

In [23]:
```
cluster2.labels_
```

Out[23]:
```
array([3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0,
       3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 1,
       3, 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, 2, 4, 2, 1, 2, 4, 2, 4, 2,
       1, 2, 4, 2, 4, 2, 4, 2, 4, 2, 1, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2,
       4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2,
       4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2, 4, 2,
       4, 2])
```

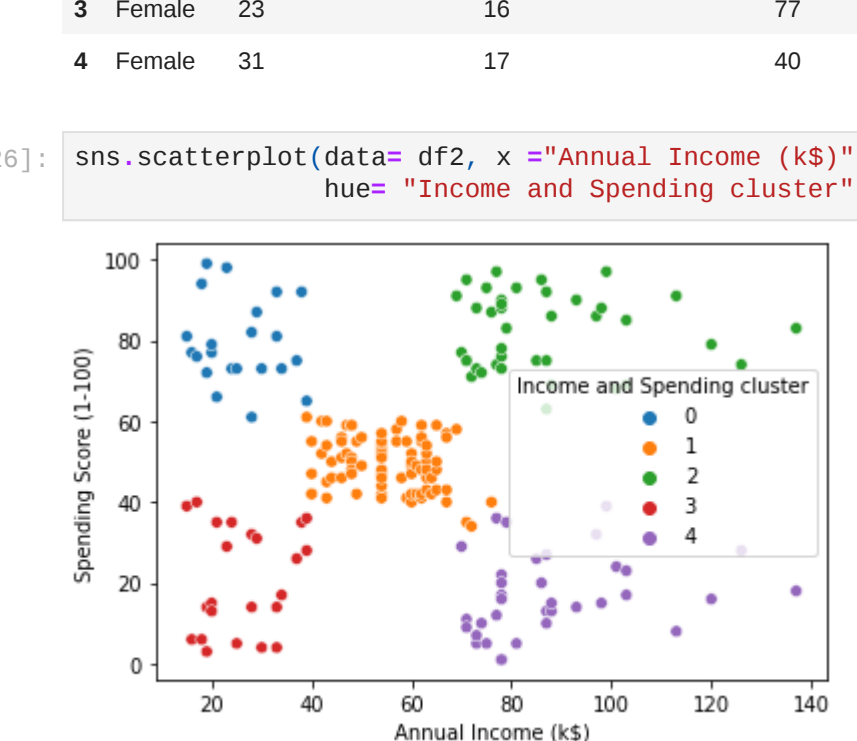
In [24]:
```
df2["Income and Spending cluster"]= cluster2.labels_
```

In [25]:
```
df2.head()
```

Out[25]:

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

In [26]:
```
sns.scatterplot(data= df2, x ="Annual Income (k$)", y= "Spending Score (1-100)",
                hue= "Income and Spending cluster", palette= "tab10");
```



In [27]:
```
cluster2.cluster_centers_
```

Out[27]:
```
array([[25.72727273, 79.36363636],
       [55.2962963 , 49.51851852],
       [86.53846154, 82.12820513],
       [26.30434783, 20.91304348],
       [88.2       , 17.11428571]])
```

In [28]:
```
centers= pd.DataFrame(cluster2.cluster_centers_)
centers
```

Out[28]:

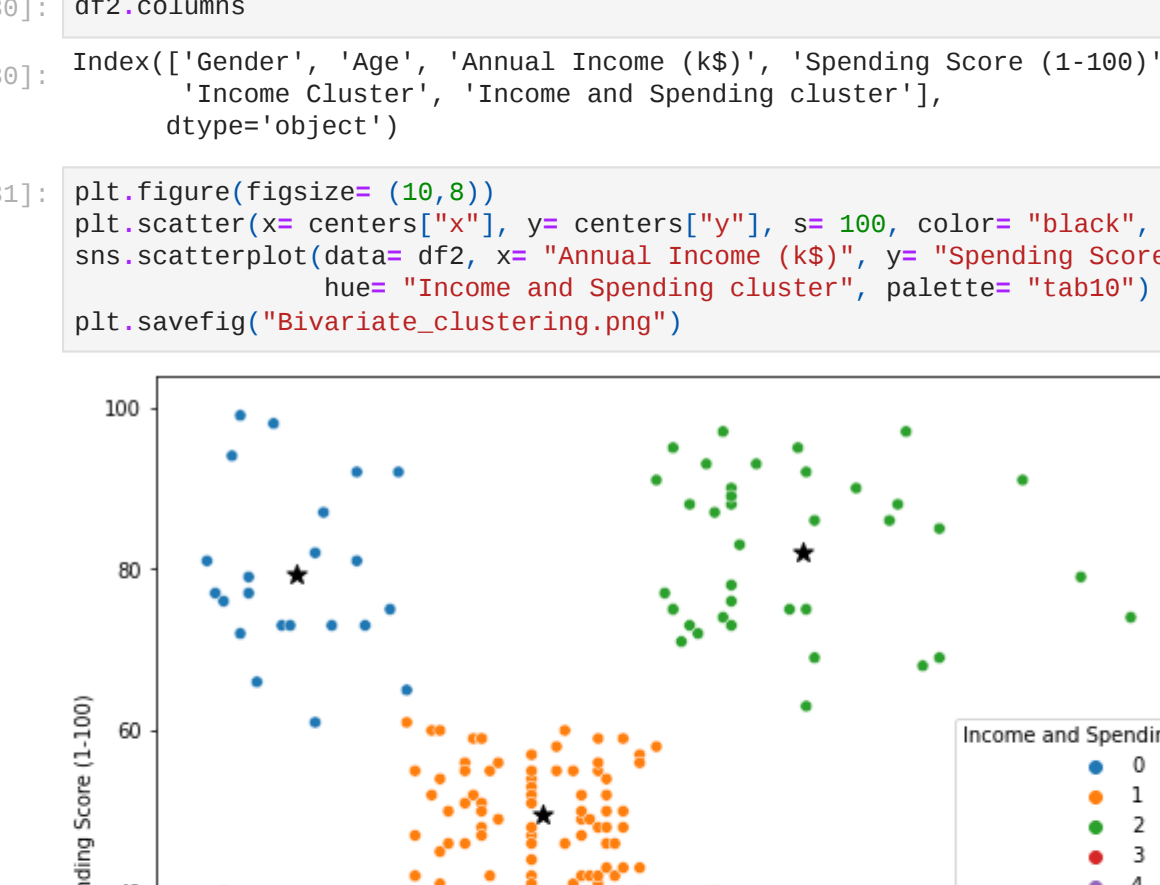
	0	1
0	25.727273	79.363636
1	55.296296	49.518519
2	86.538462	82.128205
3	26.304348	20.913043
4	88.200000	17.114286

In [29]:
```
centers.columns= ["x", "y"]
```

In [30]:
```
df2.columns
```

Out[30]:
```
Index(['Gender', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)',
       'Income Cluster', 'Income and Spending cluster'],
      dtype='object')
```

In [31]:
```
plt.figure(figsize= (10,8))
plt.scatter(x= centers["x"], y= centers["y"], s= 100, color= "black", marker= "*")
sns.scatterplot(data= df2, x= "Annual Income (k$)", y= "Spending Score (1-100)",
                hue= "Income and Spending cluster", palette= "tab10")
plt.savefig("Bivariate_clustering.png")
```



In [32]:
```
pd.crosstab(df2["Income and Spending cluster"], df2["Gender"],normalize= "index")
```

Out[32]:

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

In [33]:
```
df2.groupby("Income and Spending cluster")["Age", "Annual Income (k$)",
                                            "Spending Score (1-100)"].mean()
```

Out[33]:

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

Multivariate Clustering

For multivariate clustering we scale the data given

In [34]:
```
from sklearn.preprocessing import StandardScaler
scale= StandardScaler()
```

In [35]:
```
df.head(2)
```

Out[35]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81

In [36]:
```
df= pd.get_dummies(df, drop_first= True)
dff.head()
```

Out[36]:

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

In [37]:
```
dff= dff[["Age", "Annual Income (k$)", "Spending Score (1-100)", "Gender_Male"]]
dff.head()
```

Out[37]:

	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 [38]:
```
dff1= dff[["Age", "Annual Income (k$)", "Spending Score (1-100)", "Gender_Male"]]
dff1= scale.fit_transform(dff)
```

In [39]:
```
dff1=pd.DataFrame(dff1)
dff1
```

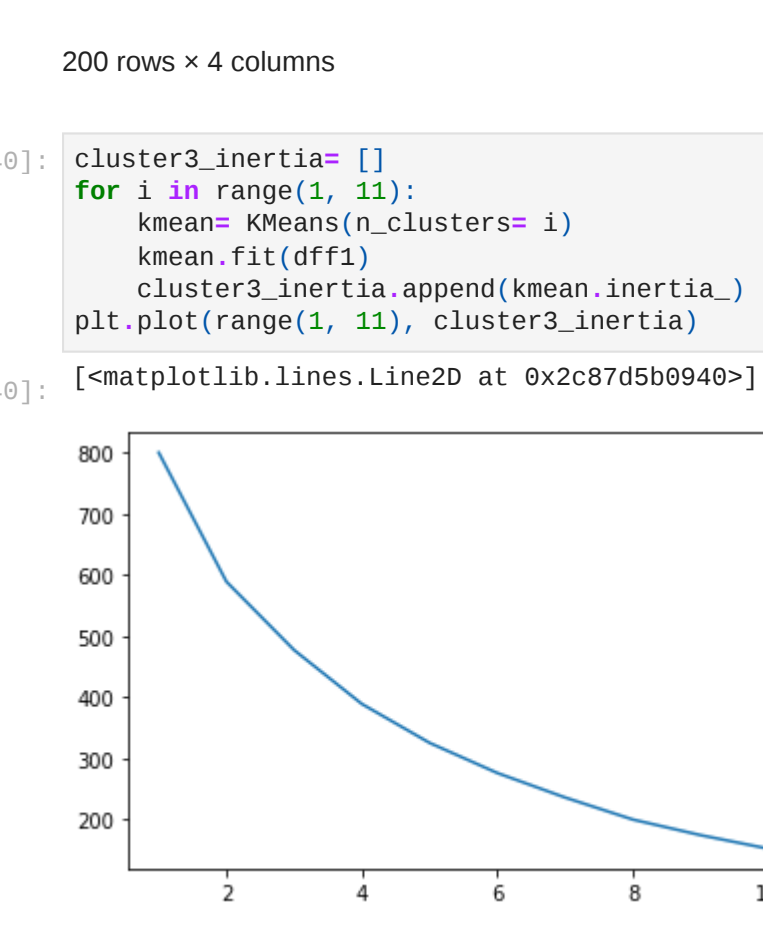
Out[39]:

	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
...	...	...	...	...
195	-0.276302	2.268791	1.118061	-0.886405
196	0.441365	2.497807	-0.861839	-0.886405
197	-0.491602	2.497807	0.923953	1.128152
198	-0.491602	2.917671	-1.250054	1.128152
199	-0.635135	2.917671	1.273347	1.128152

200 rows × 4 columns

In [40]:
```
cluster3_inertia= []
for i in range(1, 11):
    kmean= KMeans(n_clusters= i)
    kmean.fit(dff1)
    cluster3_inertia.append(kmean.inertia_)
plt.plot(range(1, 11), cluster3_inertia)
```

Out[40]: [<matplotlib.lines.Line2D at 0x2c87d5b0940>]



In [41]:
```
cluster3= KMeans(n_clusters= 3)
cluster3.fit(dff1)
```

Out[41]:
```
KMeans
KMeans(n_clusters=3)
```

In [42]:
```
dff1["Cluster"]= cluster3.labels_
```

In [43]:
```
dff1.head()
```

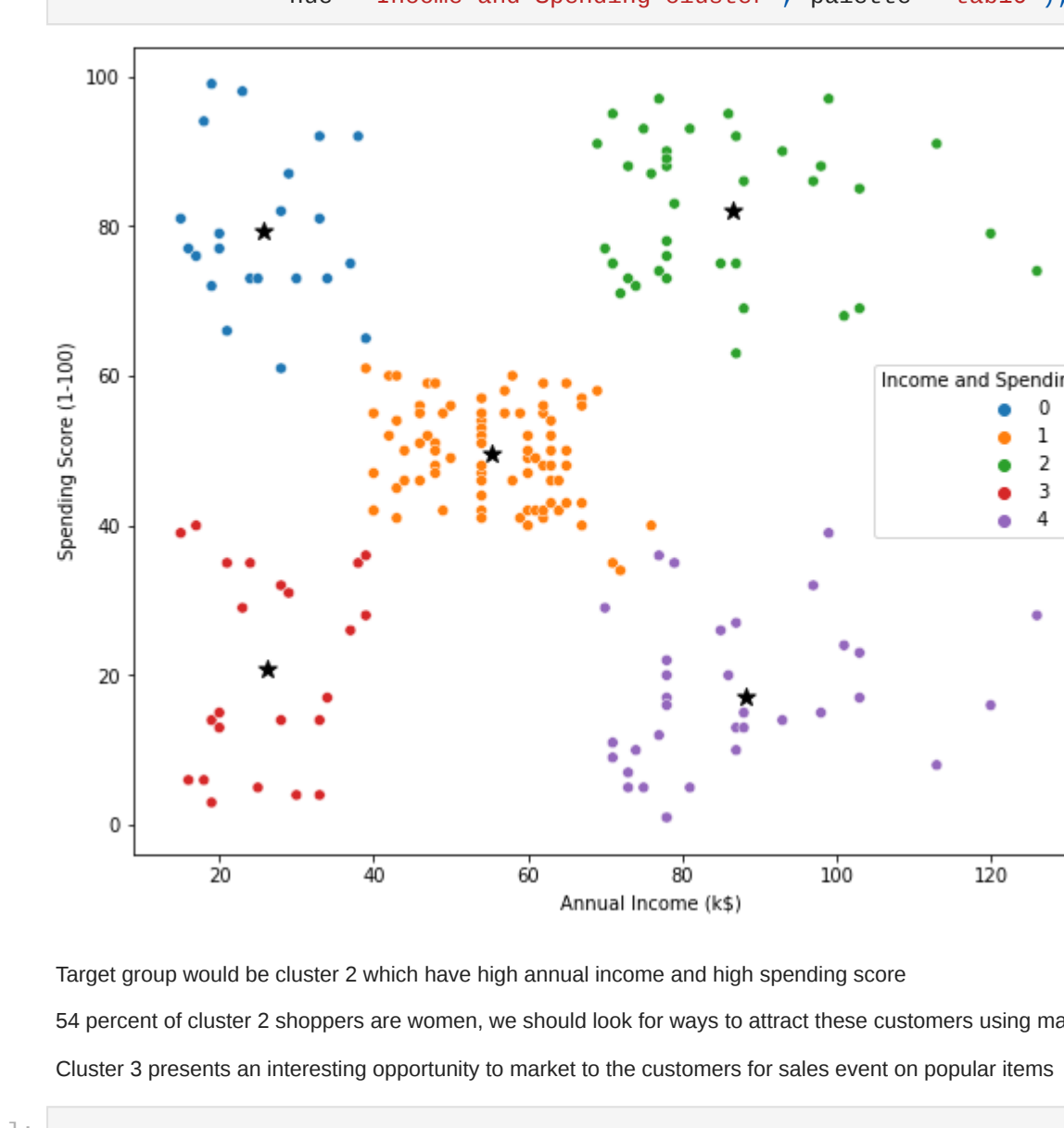
Out[43]:

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

In [ ]:
```
df2.head()
df2.to_csv("clustered_mall_customers.csv")
```

Recommendations

In [46]:
```
plt.figure(figsize= (10,8))
plt.scatter(x= centers["x"], y= centers["y"], s= 100, color= "black", marker= "*")
sns.scatterplot(data= df2, x= "Annual Income (k$)", y= "Spending Score (1-100)",
                hue= "Income and Spending cluster", palette= "tab10");
```



Target group would be cluster 2 which have high annual income and high spending score

54 percent of cluster 2 shoppers are women, we should look for ways to attract these customers using marketing campaign targeting popular items in this cluster

Cluster 3 presents an interesting opportunity to market to the customers for sales event on popular items

In [ ]: