In [2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
df = pd.read_csv("/Users/souvikchakraborty/Downloads/Mall_Customers.csv")
```

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

Univariate Analysis

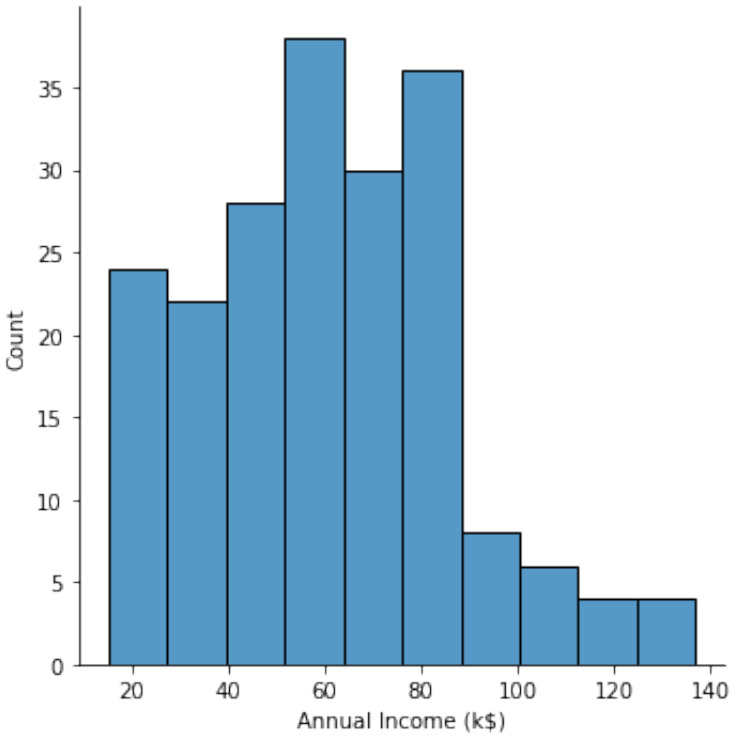
In [5]:
```
df.describe()
```

Out[5]:

	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 [6]:
```
sns.displot(df['Annual Income (k$)'])
```

Out[6]:
```
<seaborn.axisgrid.FacetGrid at 0x7fe1503b6ac0>
```

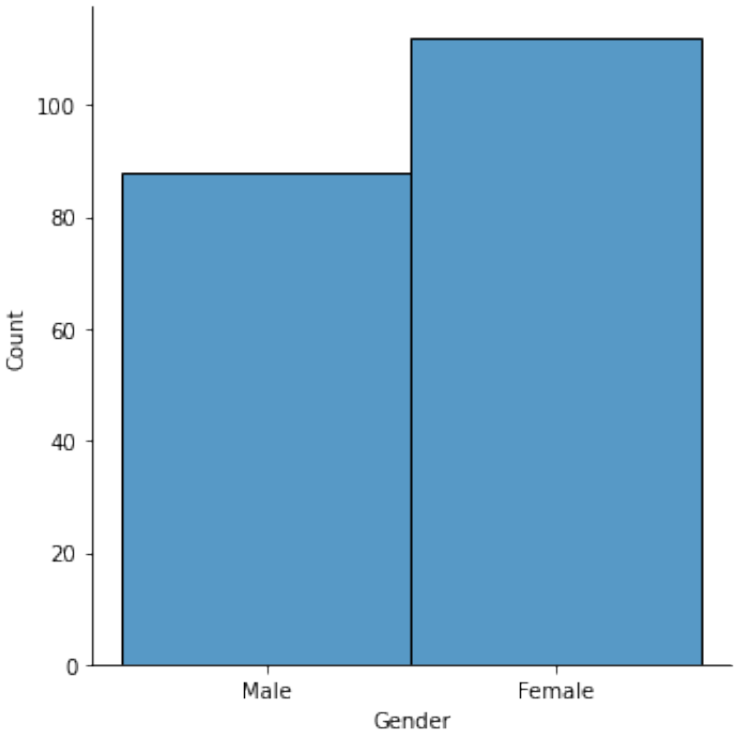


In [7]:
```
df.columns
```

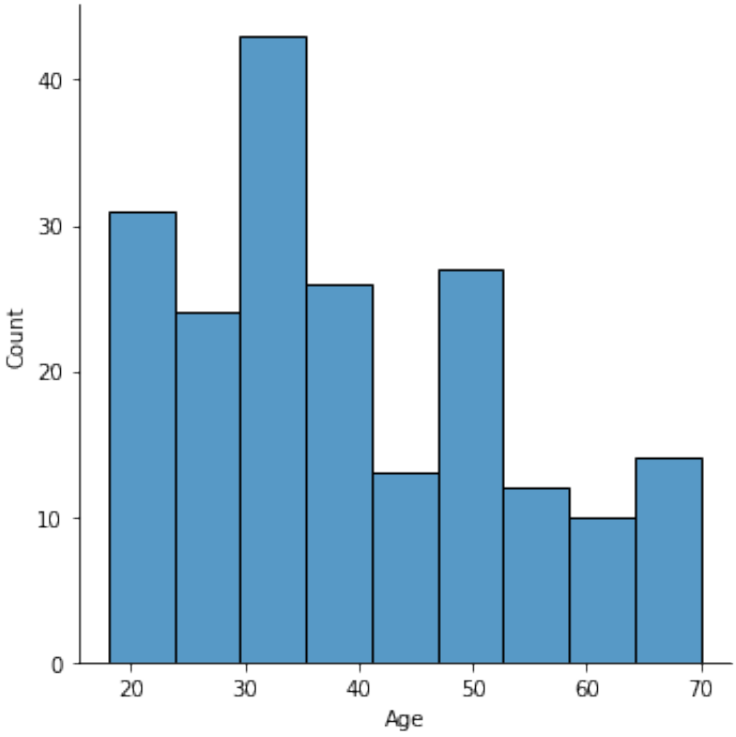
Out[7]:
```
Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

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

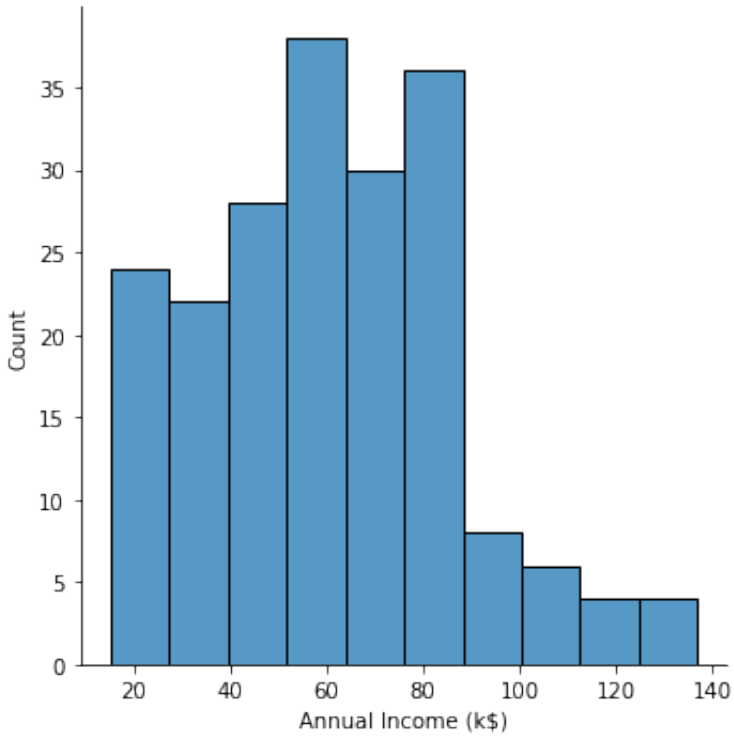
```
<Figure size 432x288 with 0 Axes>
```



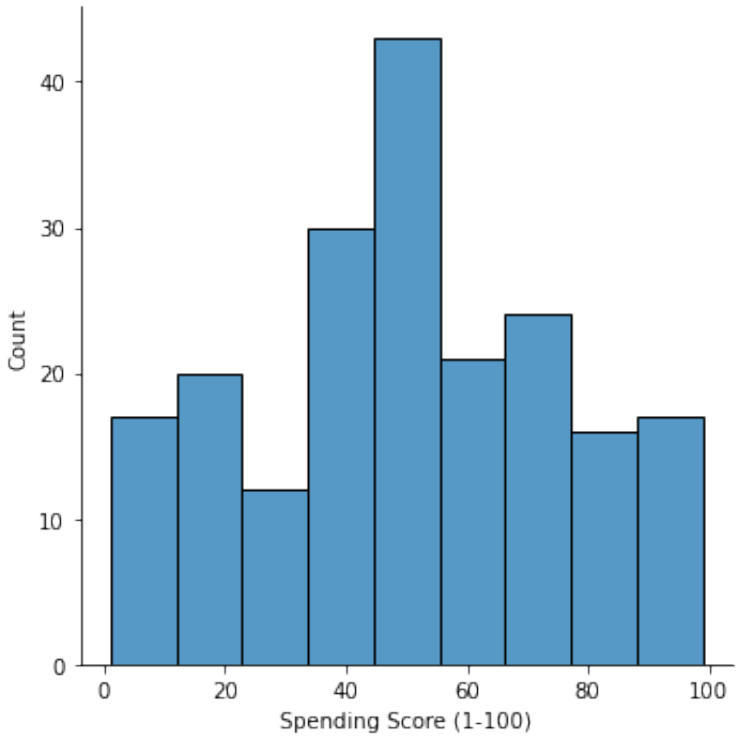

```
<Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

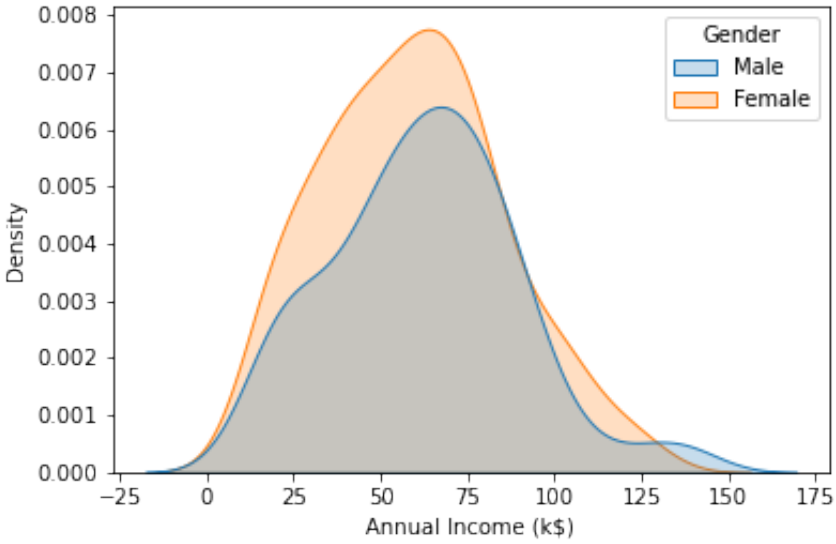



```
<Figure size 432x288 with 0 Axes>
```

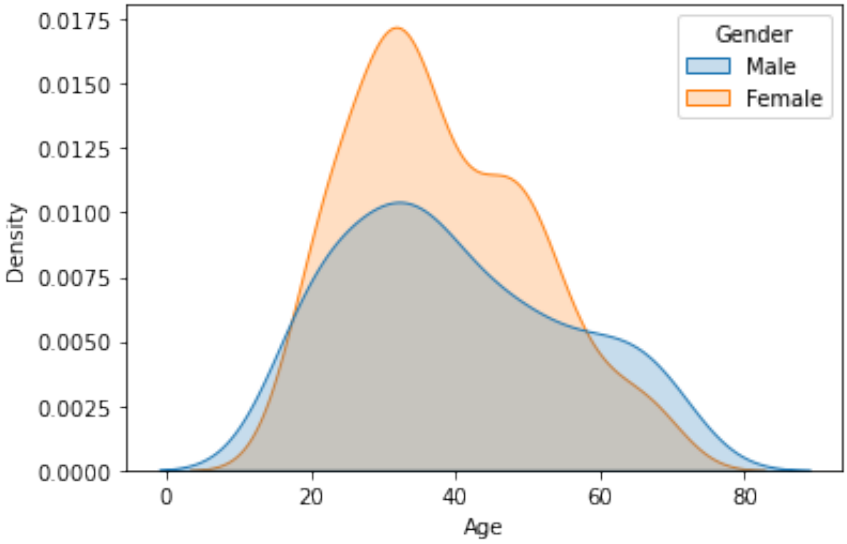


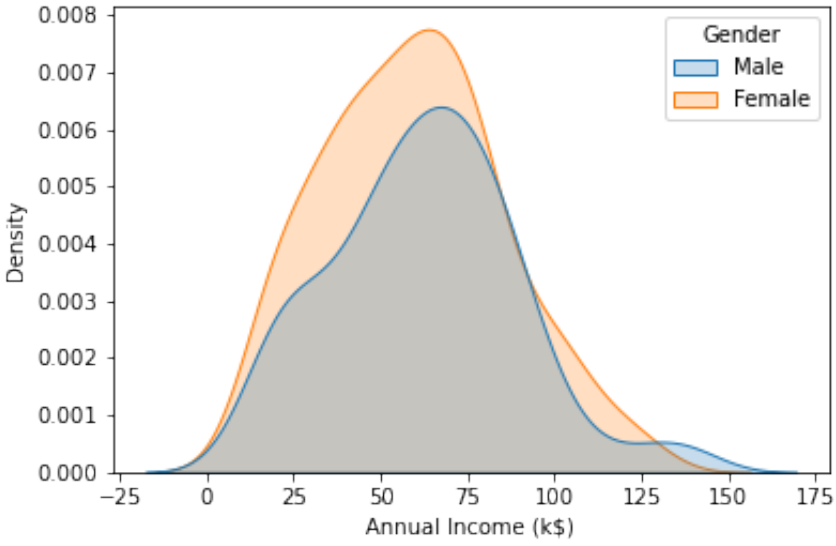
In [9]:
```
sns.kdeplot(df['Annual Income (k$)'], shade=True, hue=df['Gender'])
```

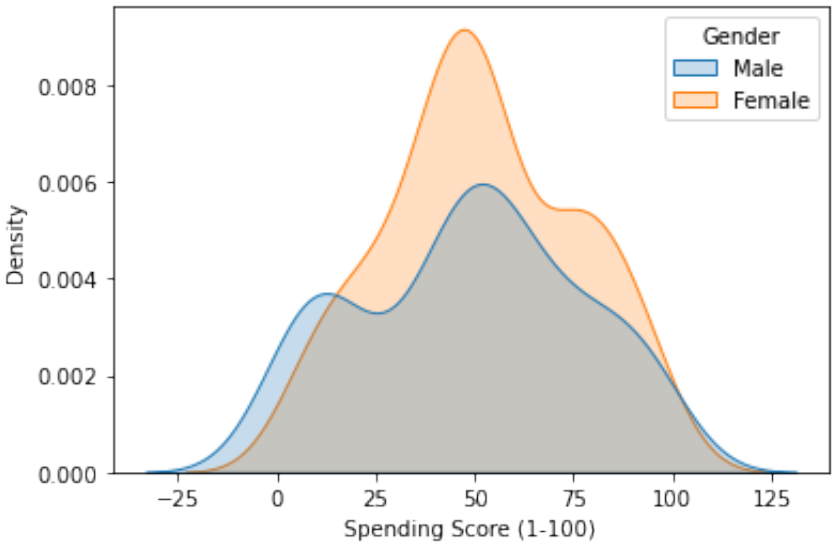
Out[9]:
```
<AxesSubplot:xlabel='Annual Income (k$)', ylabel='Density'>
```



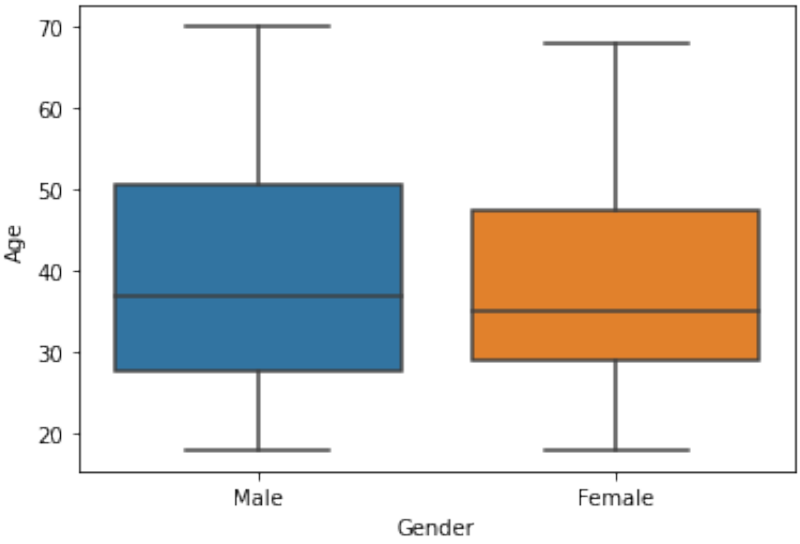
In [10]:
```
columns = ['Age', 'Annual Income (k$)',
       'Spending Score (1-100)']
for i in columns:
    plt.figure()
    sns.kdeplot(df[i], shade=True, hue=df['Gender'])
```

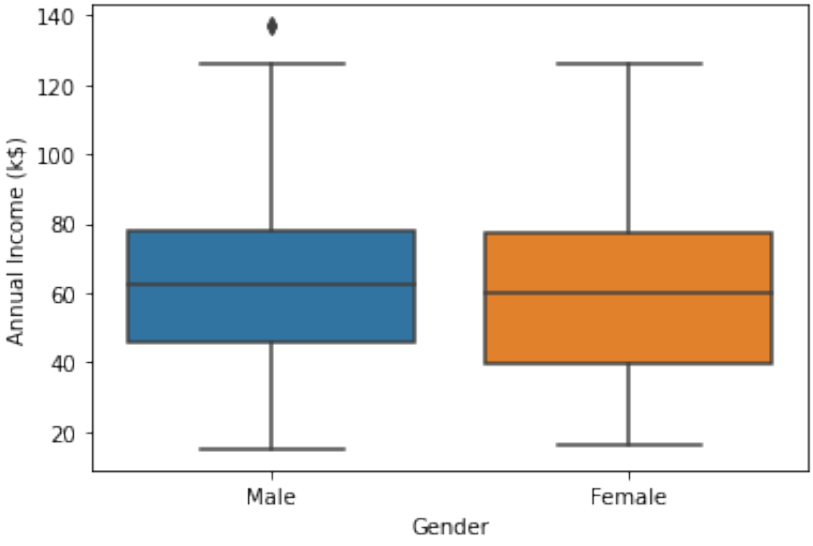


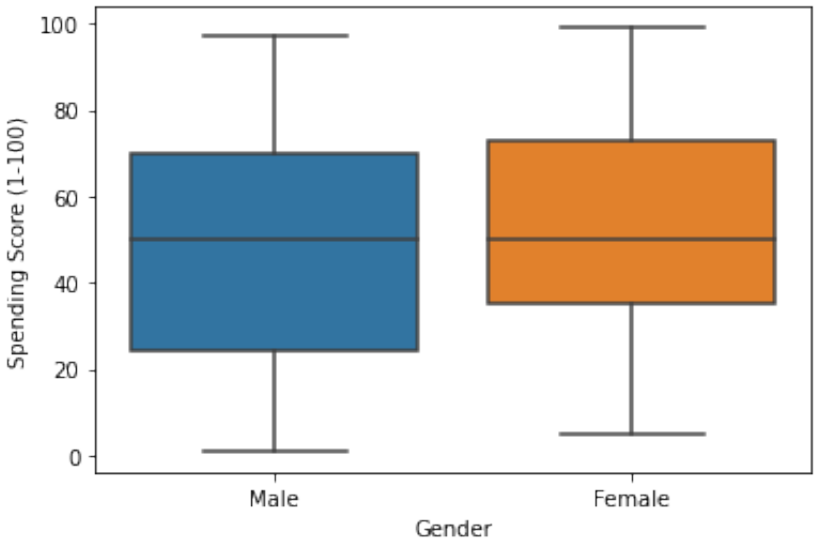




In [11]:
```
columns = ['Age', 'Annual Income (k$)',
       'Spending Score (1-100)']
for i in columns:
    plt.figure()
    sns.boxplot(data=df, x='Gender', y = df[i])
```







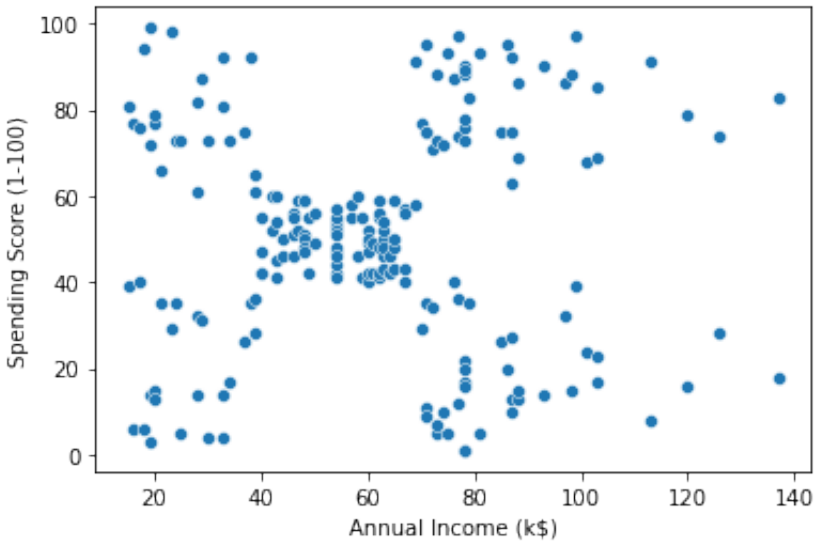
In [12]:
```
df['Gender'].value_counts(normalize=True)
```

Out[12]:
```
Female    0.56
Male      0.44
Name: Gender, dtype: float64
```

Bivariate Analysis

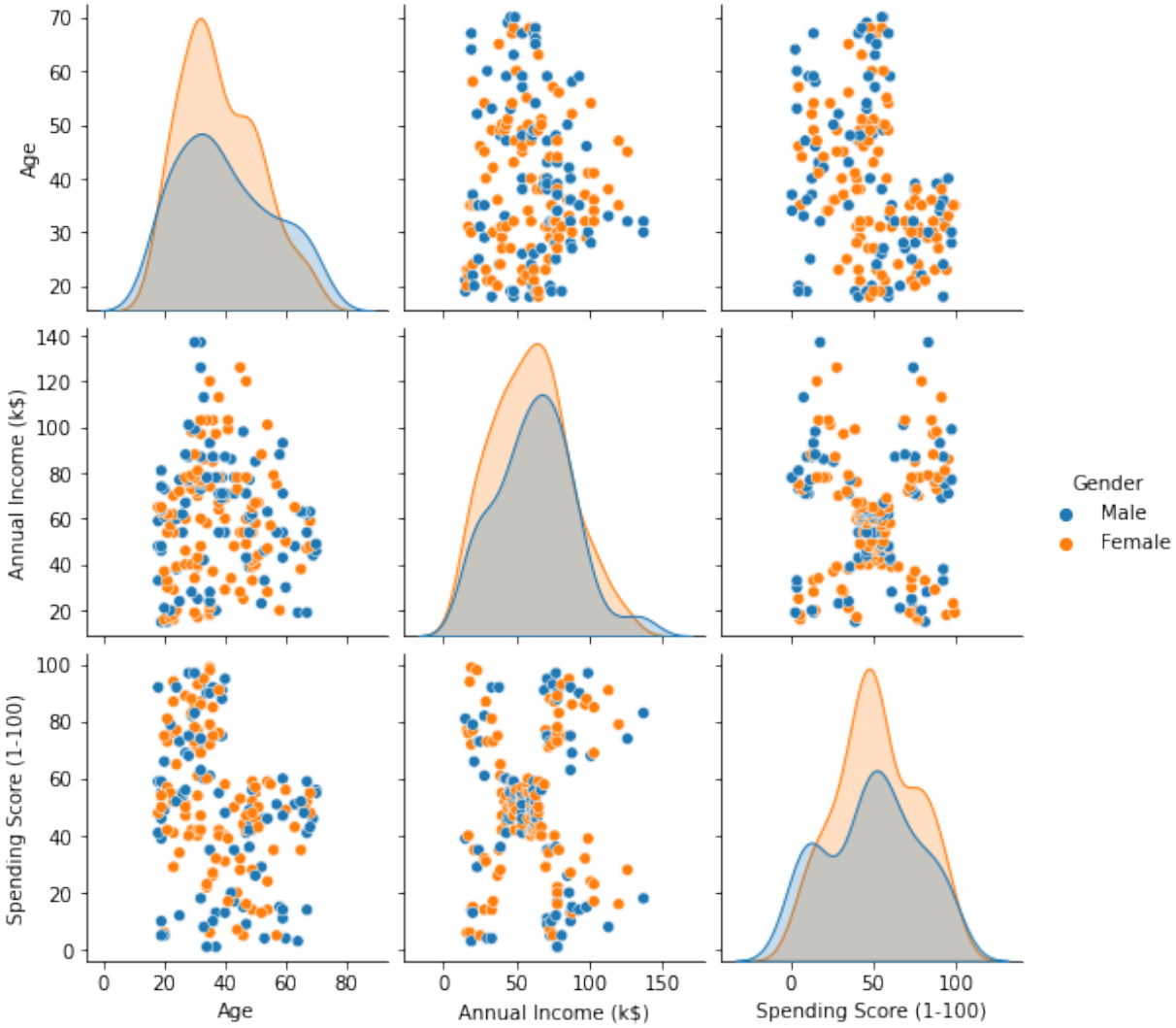
In [13]:
```
sns.scatterplot(data=df, x = 'Annual Income (k$)', y='Spending Score (1-1
```

Out[13]:
```
<AxesSubplot:xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'
>
```



In [14]:
```
df = df.drop('CustomerID', axis=1)
sns.pairplot(data=df, hue='Gender')
```

Out[14]:
```
<seaborn.axisgrid.PairGrid at 0x7fe1314c1eb0>
```



In [15]:
```
df.groupby(['Gender'])['Age', 'Annual Income (k$)',
       'Spending Score (1-100)'].mean()
```

Out[15]:

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

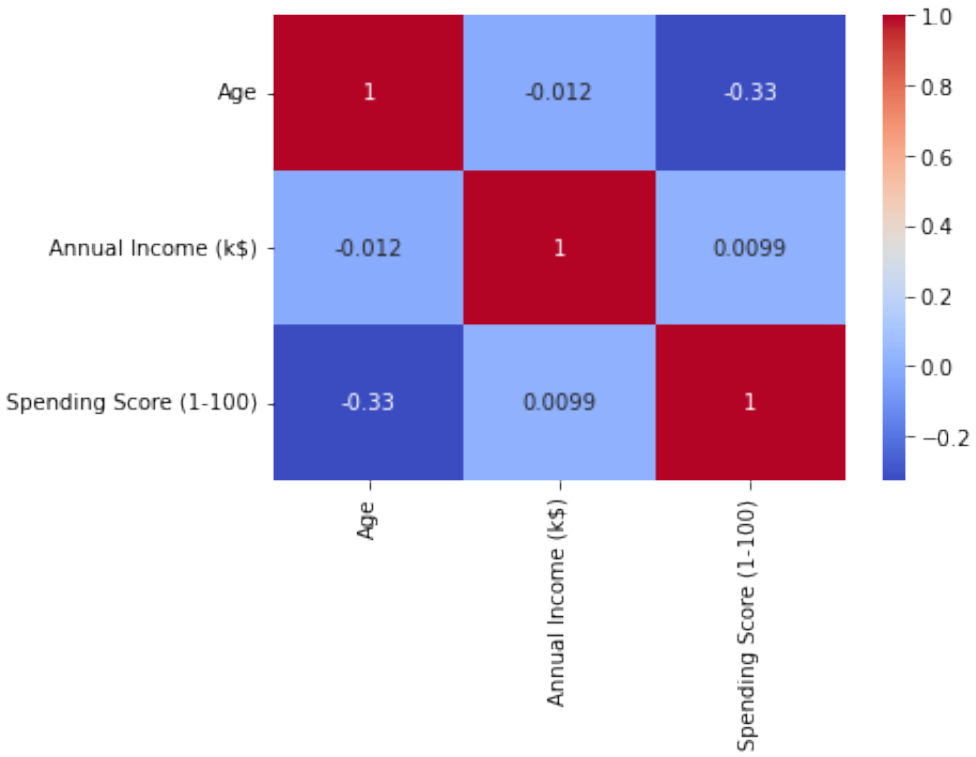
In [16]:
```
df.corr()
```

Out[16]:

	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 [17]:
```
sns.heatmap(df.corr(), annot=True, cmap = 'coolwarm')
```

Out[17]:
```
<AxesSubplot:>
```



Clustering - Univariate, Bivariate, Multivariate

In [37]:
```
clustering1 = KMeans(n_clusters=3)
```

In [38]:
```
clustering1.fit(df[['Annual Income (k$)']])
```

Out[38]:
```
KMeans(n_clusters=3)
```

In [39]:
```
clustering1.labels_
```

Out[39]:
```
array([2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 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, 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], dtype=int32)
```

In [40]:
```
df['Income Cluster'] = clustering1.labels_
df.head()
```

Out[40]:

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

In [41]:
```
df['Income Cluster'].value_counts()
```

Out[41]:
```
1    92
2    72
0    36
Name: Income Cluster, dtype: int64
```

In [42]:
```
clustering1.inertia_
```

Out[42]:
```
23528.152173913044
```

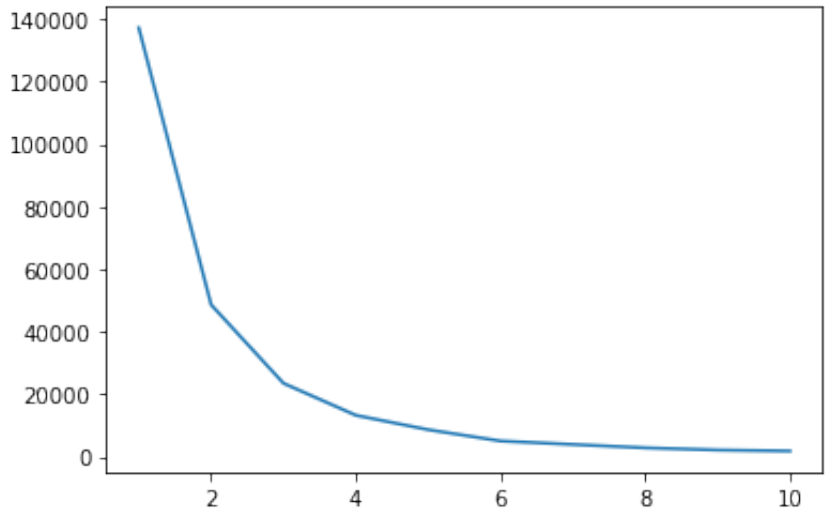
In [43]:
```
inertia_scores = []
for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(df[['Annual Income (k$)']])
    inertia_scores.append(kmeans.inertia_)
```

In [44]:
```
inertia_scores
```

Out[44]:
```
[137277.28000000003,
 48660.88888888889,
 23517.330930930933,
 13278.112713472485,
 8667.679614837509,
 5050.9047619047615,
 3941.4163614163613,
 2857.441697191697,
 2168.4787157287155,
 1844.9249999999997]
```

In [45]:
```
plt.plot(range(1,11), inertia_scores)
```

Out[45]:
```
[<matplotlib.lines.Line2D at 0x7fe145d6c460>]
```



In [46]:
```
df.columns
```

Out[46]:
```
Index(['Gender', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)',
       'Income Cluster'],
      dtype='object')
```

In [47]:
```
df.groupby('Income Cluster')['Age', 'Annual Income (k$)',
       'Spending Score (1-100)'].mean()
```

Out[47]:

Income Cluster	Age	Annual Income (k$)	Spending Score (1-100)
0	37.833333	99.888889	50.638889
1	39.184783	66.717391	50.054348
2	38.930556	33.027778	50.166667

Bivarite Clustering

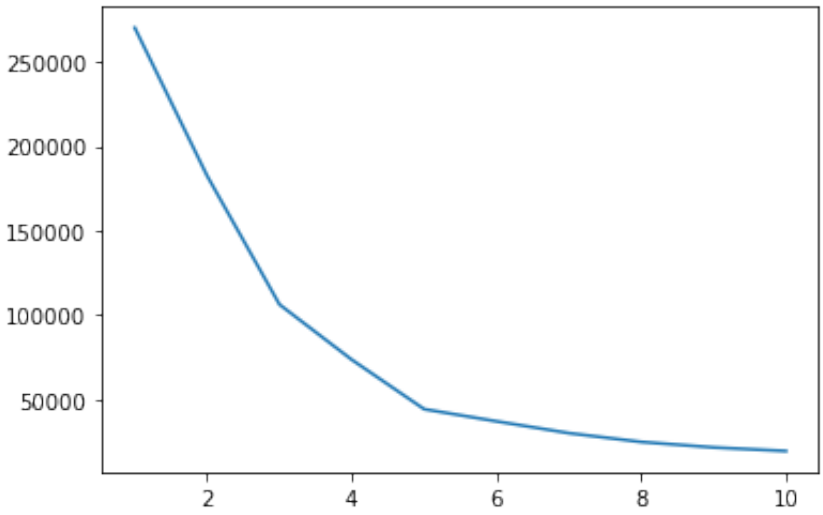
In [48]:
```
clustering2 = KMeans()
clustering2.fit(df[['Annual Income (k$)',
       'Spending Score (1-100)']])
clustering2.labels_
df['Spending and Income Cluster'] = clustering2.labels_
df.head()
```

Out[48]:

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

In [50]:
```
inertia_scores2 = []
for i in range(1,11):
    kmeans2 = KMeans(n_clusters=i)
    kmeans2.fit(df[['Annual Income (k$)','Spending Score (1-100)']])
    inertia_scores2.append(kmeans2.inertia_)
plt.plot(range(1,11),inertia_scores2)
```

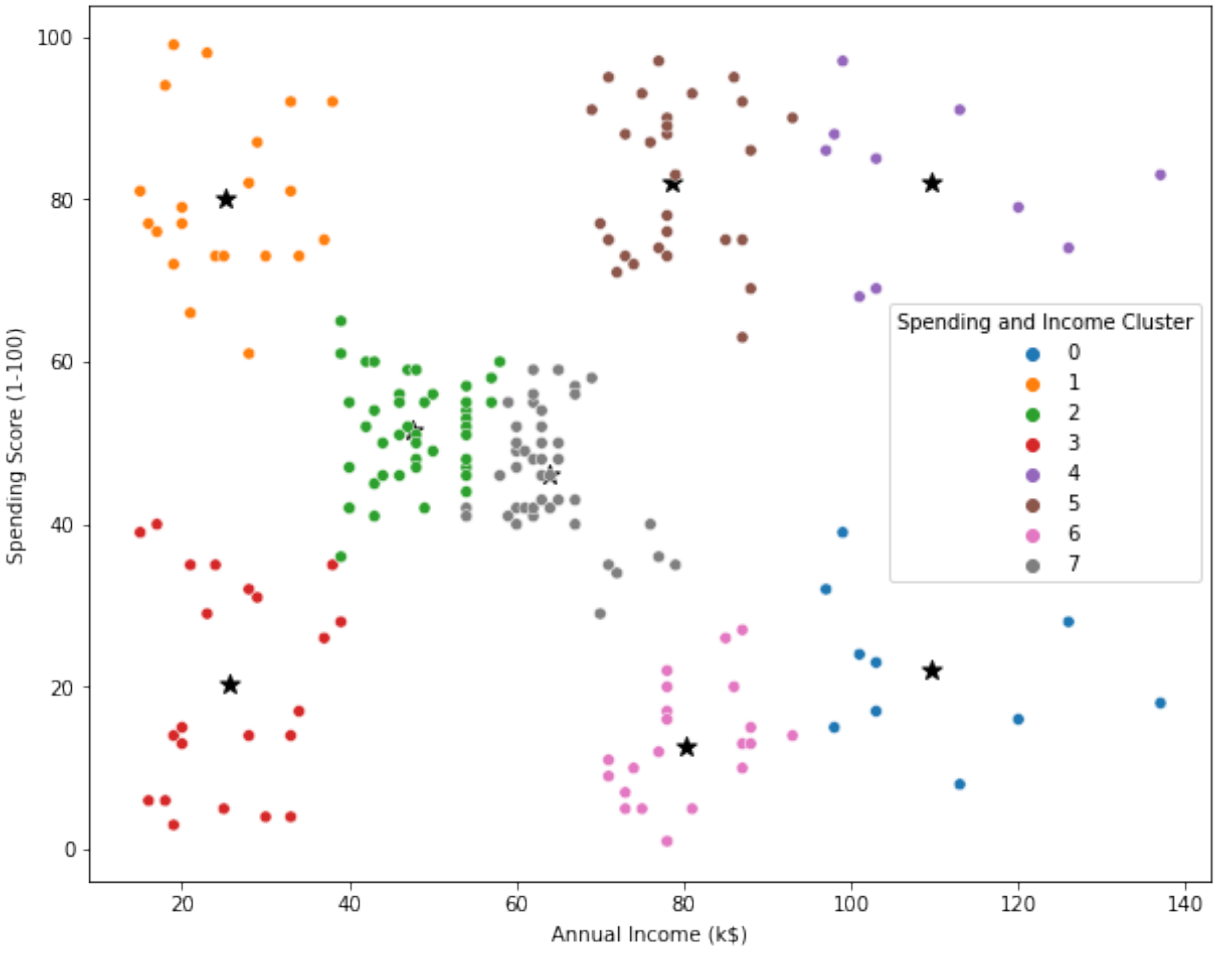
Out[50]:
```
[<matplotlib.lines.Line2D at 0x7fe13191f850>]
```



In [61]:
```
centers = pd.DataFrame(clustering2.cluster_centers_)
centers.columns = ['x','y']
```

In [64]:
```
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)
```

Out[64]:
```
<AxesSubplot:xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'
>
```



In [65]:
```
pd.crosstab(df['Spending and Income Cluster'], df['Gender'],normalize='in
```

Out[65]:

Spending and Income Cluster	Female	Male
0	0.700000	0.300000
1	0.571429	0.428571
2	0.590909	0.409091
3	0.636364	0.363636
4	0.600000	0.400000
5	0.517241	0.482759
6	0.318182	0.681818
7	0.595238	0.404762

In [66]:
```
df.groupby('Spending and Income Cluster')['Age', 'Annual Income (k$)',
       'Spending Score (1-100)'].mean()
```

Out[66]:

Spending and Income Cluster	Age	Annual Income (k$)	Spending Score (1-100)
0	41.000000	109.700000	22.000000
1	25.333333	25.095238	80.047619
2	43.477273	47.659091	51.613636
3	45.090909	25.727273	20.227273
4	32.200000	109.700000	82.000000
5	32.862069	78.551724	82.172414
6	41.000000	80.181818	12.681818
7	41.571429	63.952381	46.214286

In [68]:
```
#Multivariate Clustering
from sklearn.preprocessing import StandardScaler
```

In [69]:
```
scale = StandardScaler
```

In [70]:
```
df.head()
```

Out[70]:

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

In [72]:
```
dff = pd.get_dummies(df, drop_first=True)
dff.head()
```

Out[72]:

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

In [73]:
```
dff.columns
```

Out[73]:
```
Index(['Age', 'Annual Income (k$)', 'Spending Score (1-100)', 'Income Cluster',
       'Spending and Income Cluster', 'Gender_Male'],
      dtype='object')
```

In [ ]: