Python Project Mall

May 5, 2023

[1]:
```
import pandas as pd # Data manipulation
import seaborn as sns # Statistical visualization library
import matplotlib.pyplot as plt # Another visualization library
from sklearn.cluster import KMeans # For create clusters
import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
df = pd.read_csv("C:/Users/Drac_/OneDrive/Desktop/Python_Project_Data/
  ↪Mall_Customers.csv")
```

[3]:
```
df.head()
```

[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
```

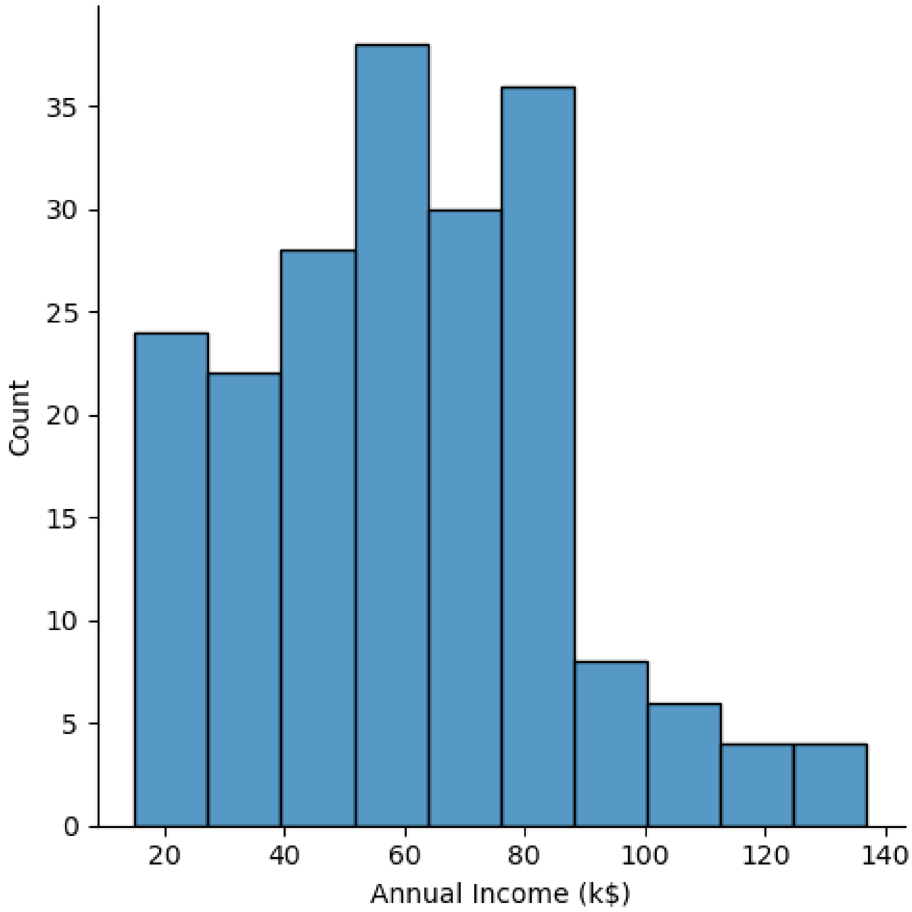
1 Univariate Analysis

[4]:
```
df.describe()
```

[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
```

[5]:
```
sns.displot(df["Annual Income (k$)"])
```

[5]:
```
<seaborn.axisgrid.FacetGrid at 0x2146169b0a0>
```

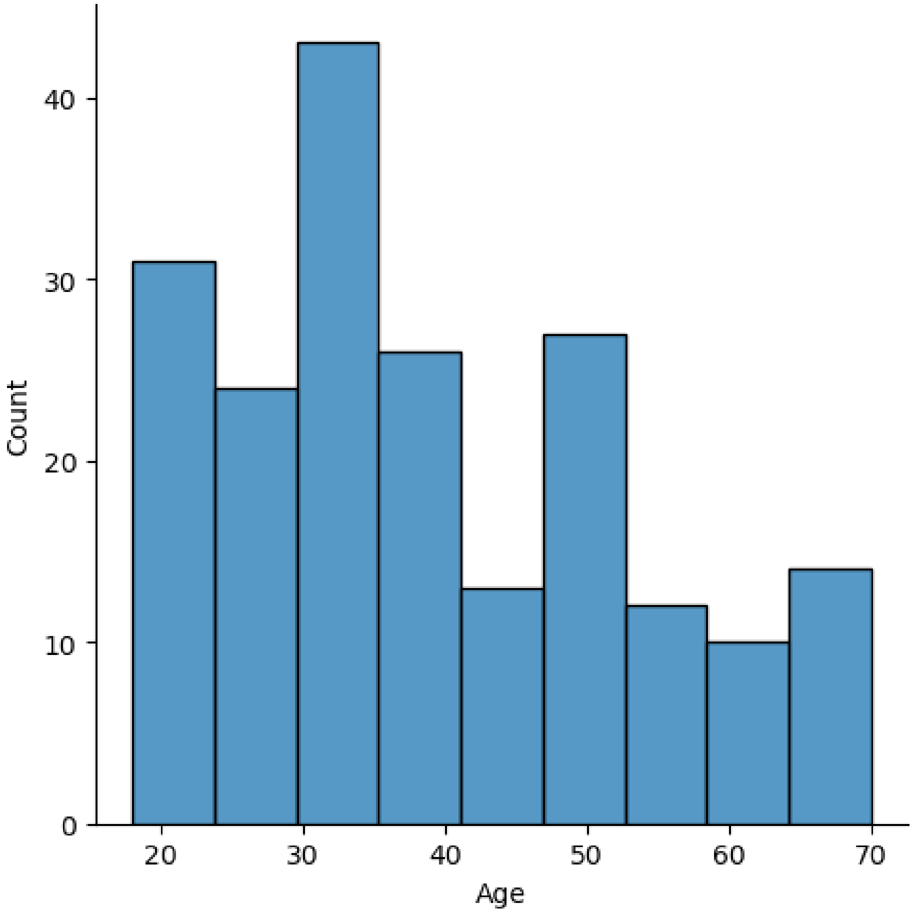


[6]:
```
df.columns
```

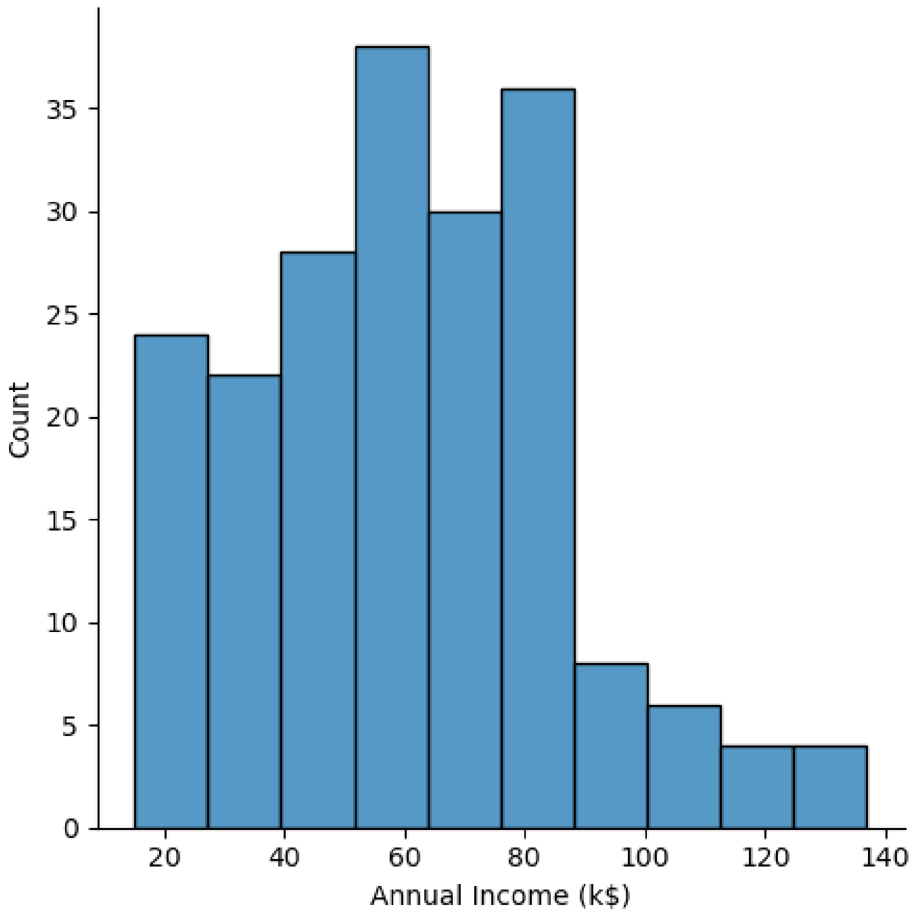
[6]:
```
Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

[7]:
```
columns = ['Age', 'Annual Income (k$)','Spending Score (1-100)']
for i in columns:
    plt.figure()
    sns.displot(df[i])
```

```
<Figure size 640x480 with 0 Axes>
```

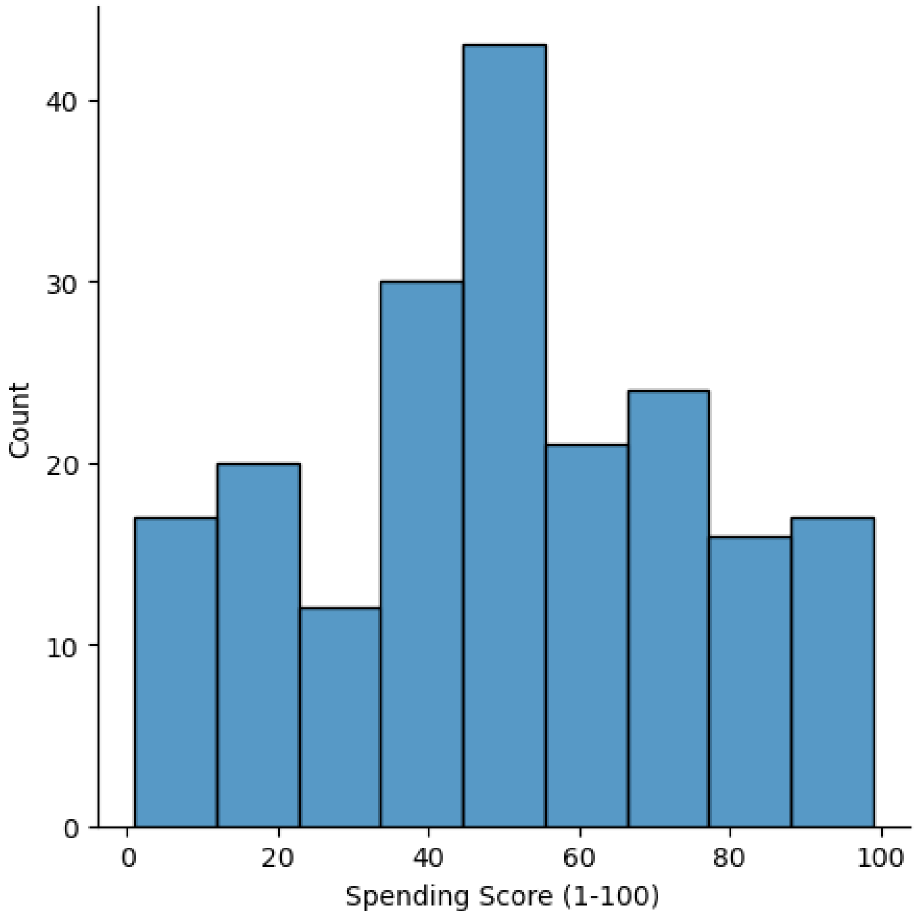

0
10
20
30
40
Count
20
30
40
50
60
70
Age

```
<Figure size 640x480 with 0 Axes>
```

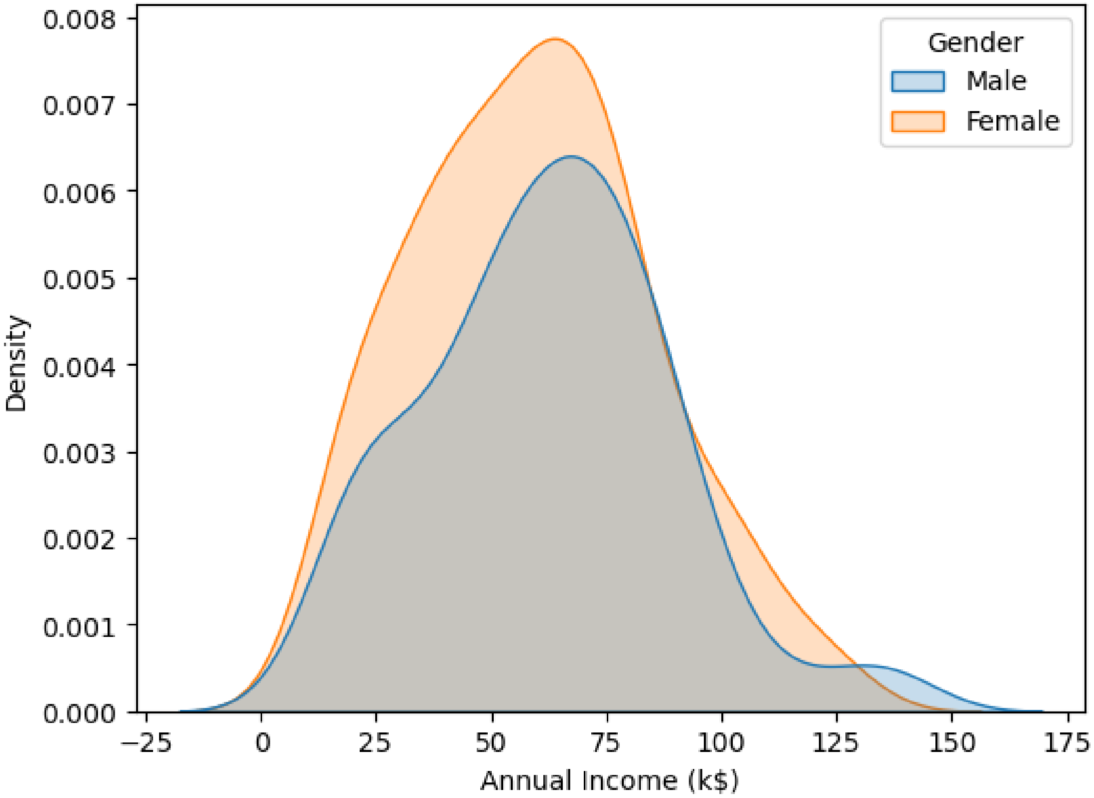


35
30
25
20
15
10
5
0
Count
20
40
60
80
100
120
140
Annual Income (k$)

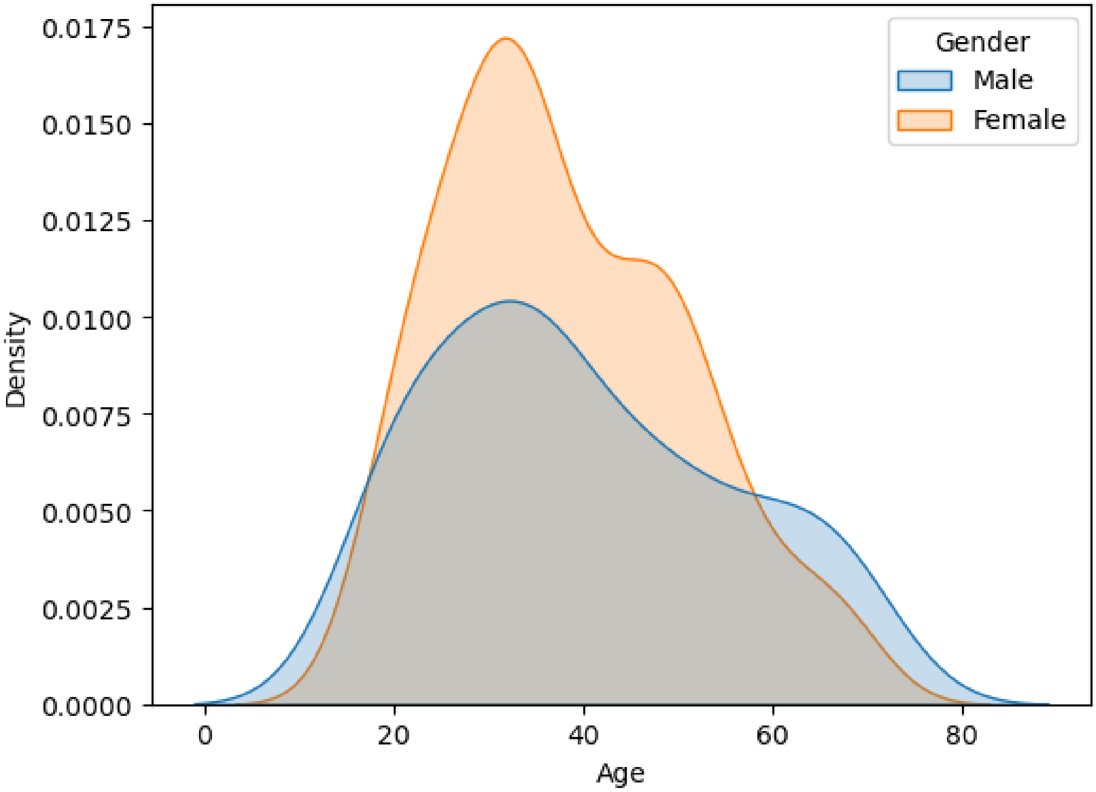
```
<Figure size 640x480 with 0 Axes>
```

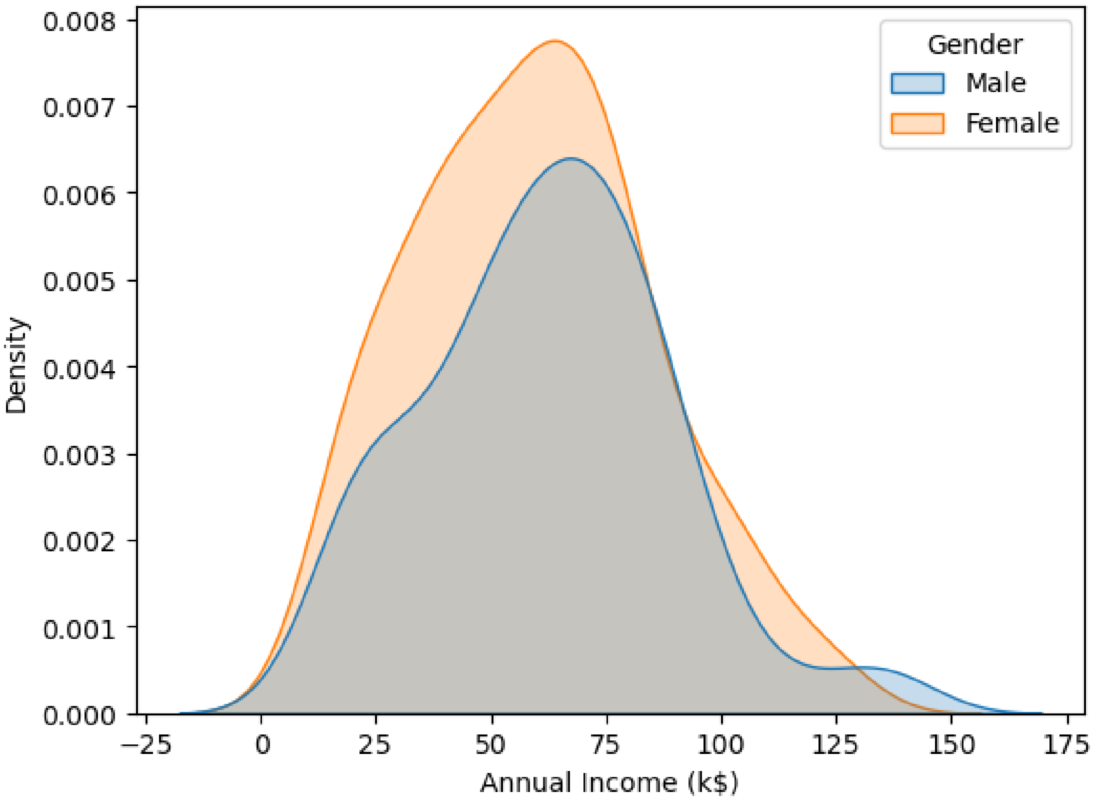


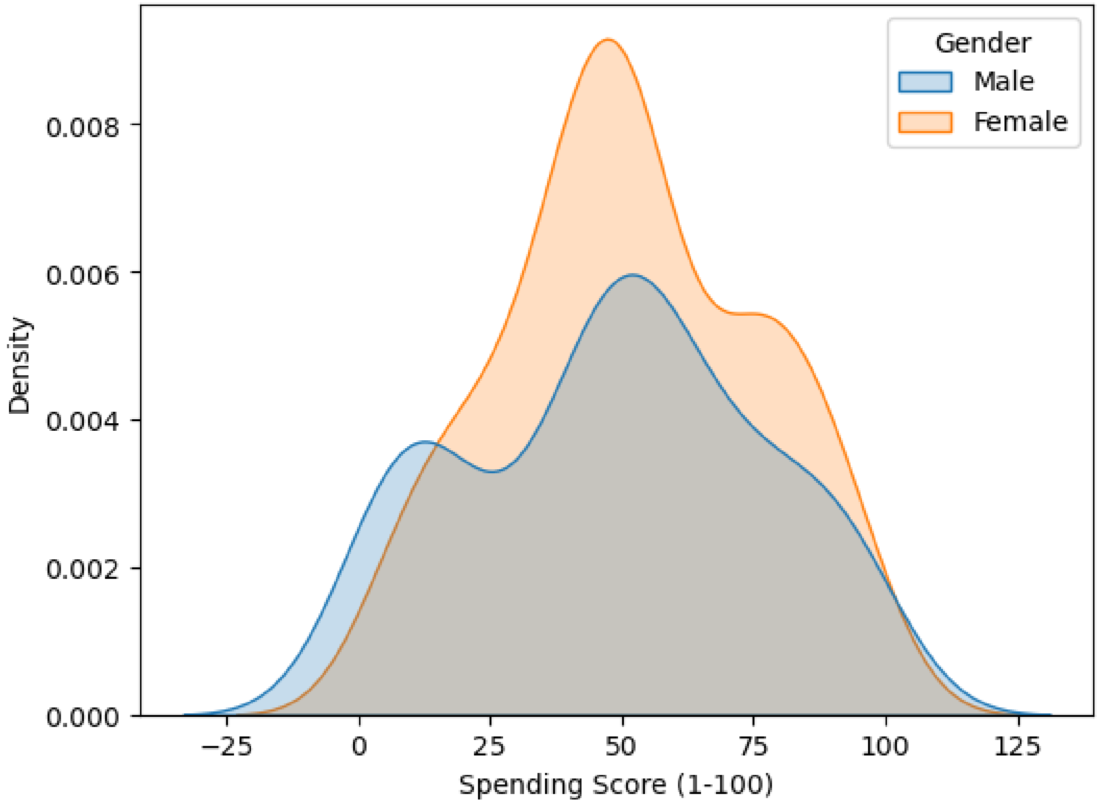
[8]:
```
sns.kdeplot(x=df["Annual Income (k$)"],shade=True,hue=df["Gender"]);
```



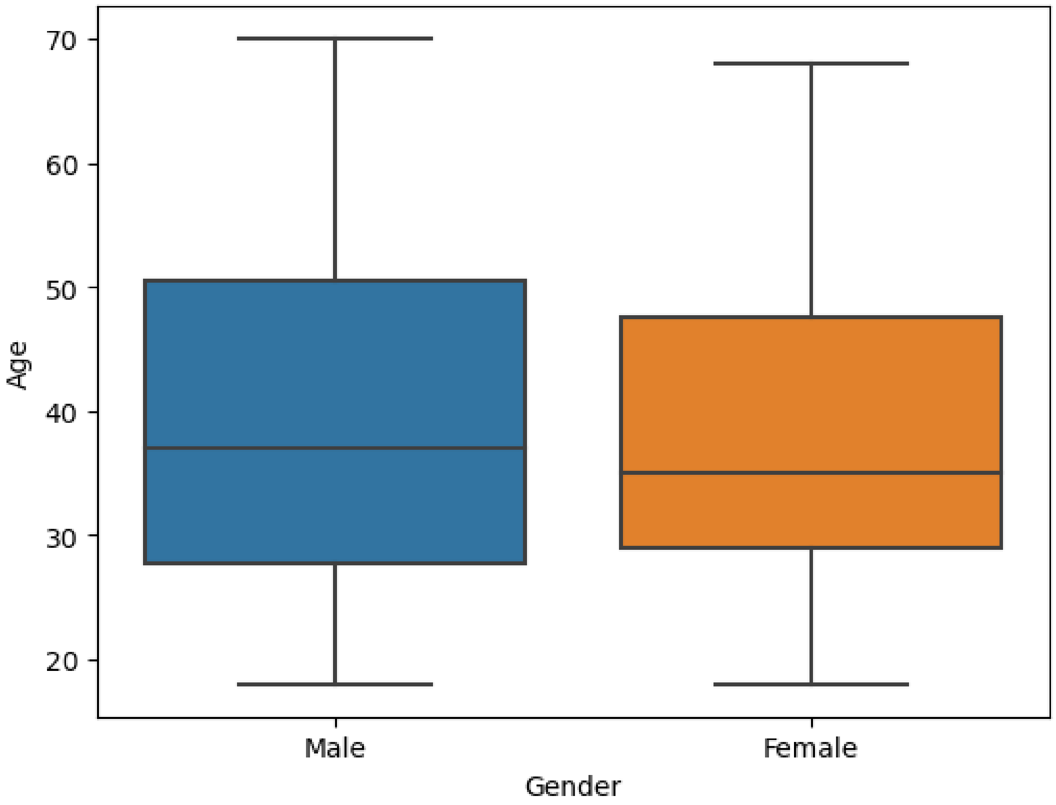
[9]:
```
columns = ['Age', 'Annual Income (k$)','Spending Score (1-100)']
for i in columns:
    plt.figure()
    sns.kdeplot(x=df[i],shade=True,hue=df["Gender"]);
```

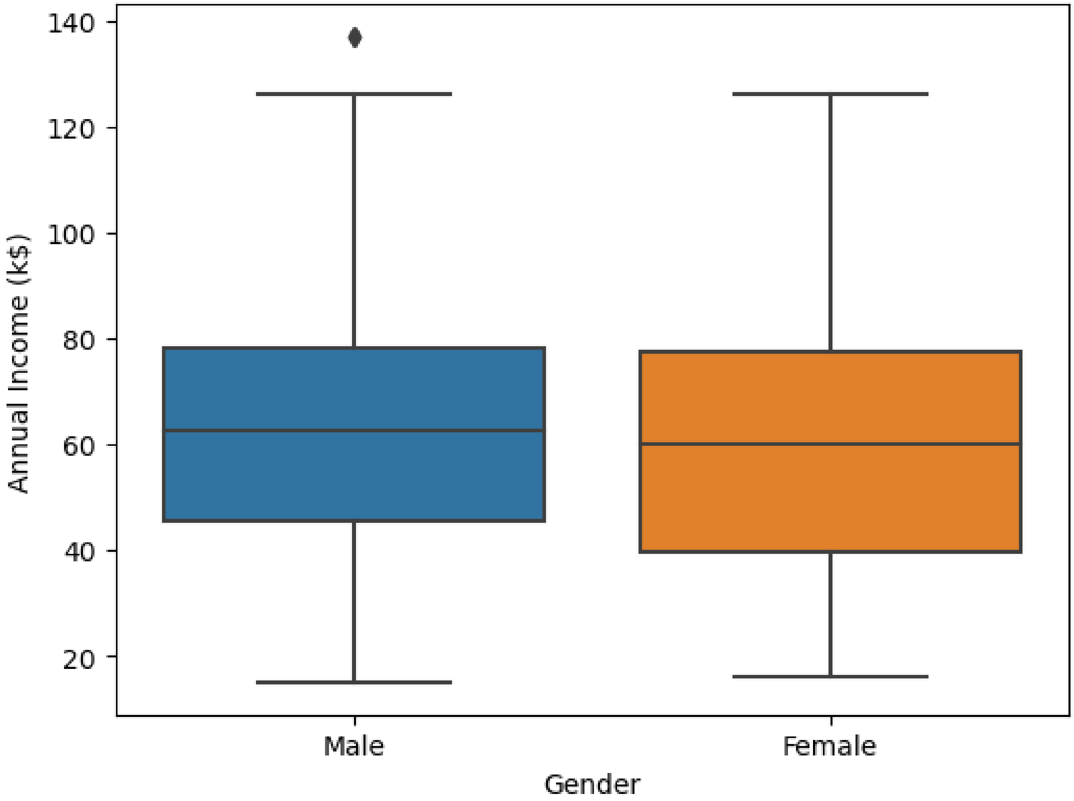

Gender
Male
Female
Density
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0
20
40
60
80
Age

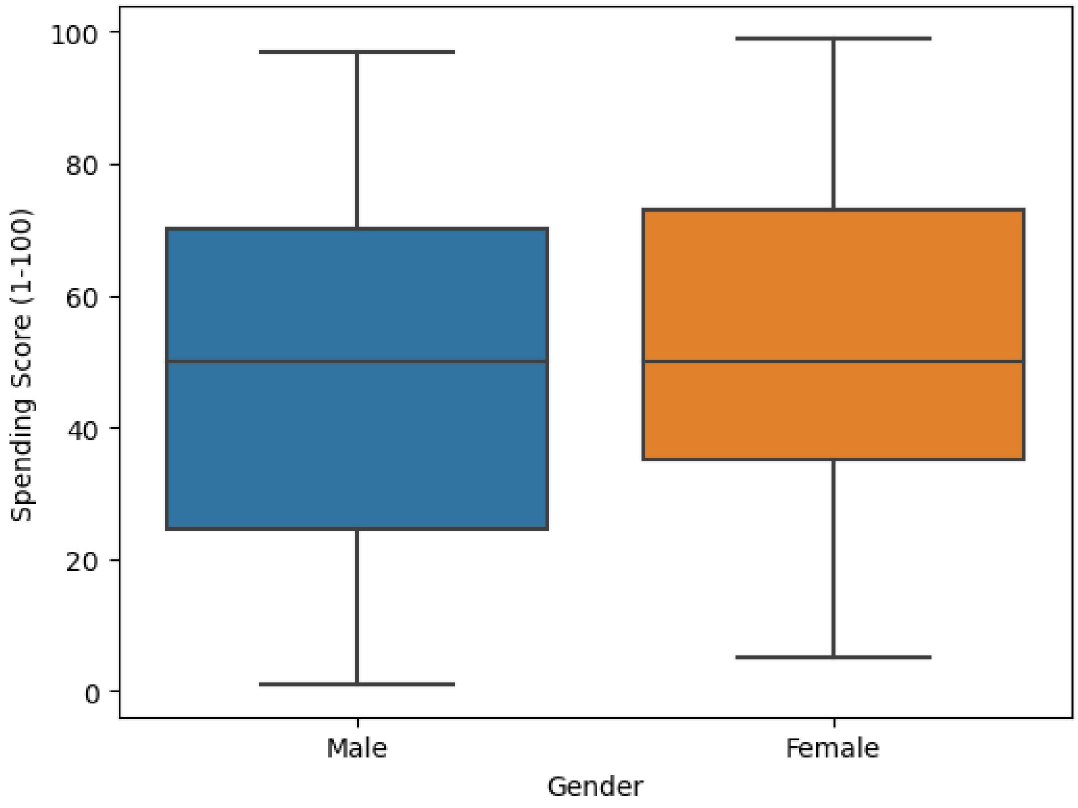

Gender
Male
Female
Density
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
−25
0
25
50
75
100
125
150
175
Annual Income (k$)



[10]:
```
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
Male
Female
Gender


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



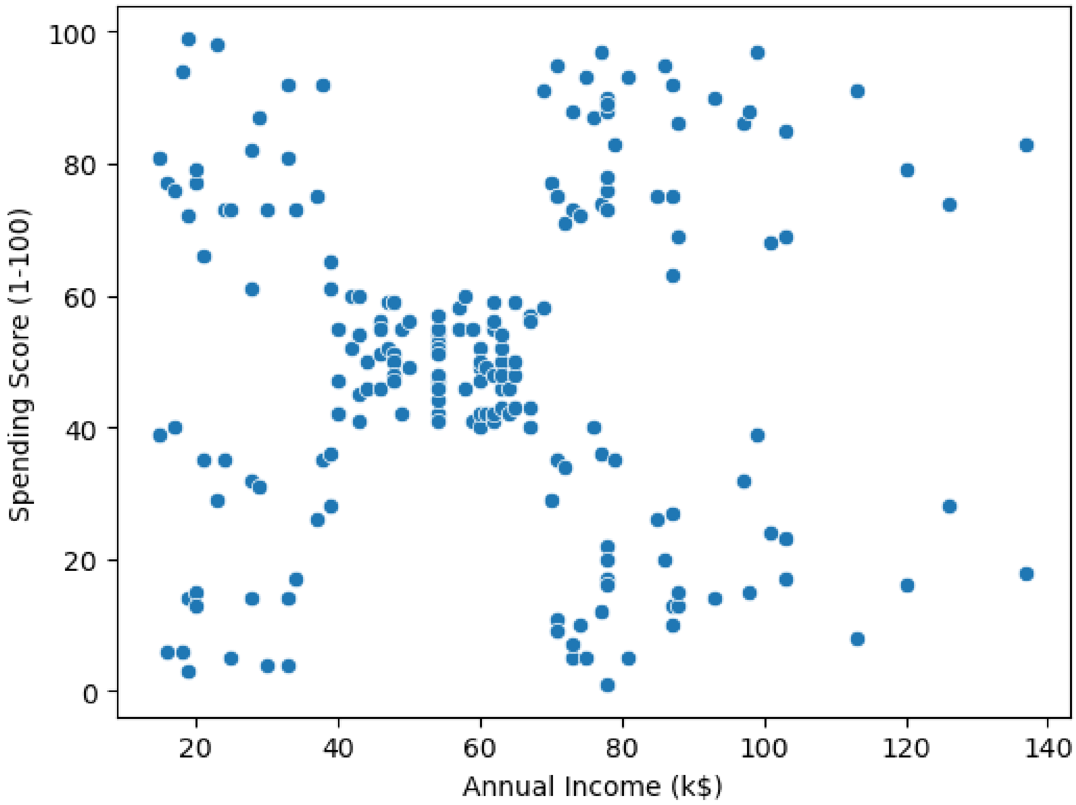
```
[11]: df["Gender"].value_counts(normalize=True)
```

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

2 Bivariate Analysis

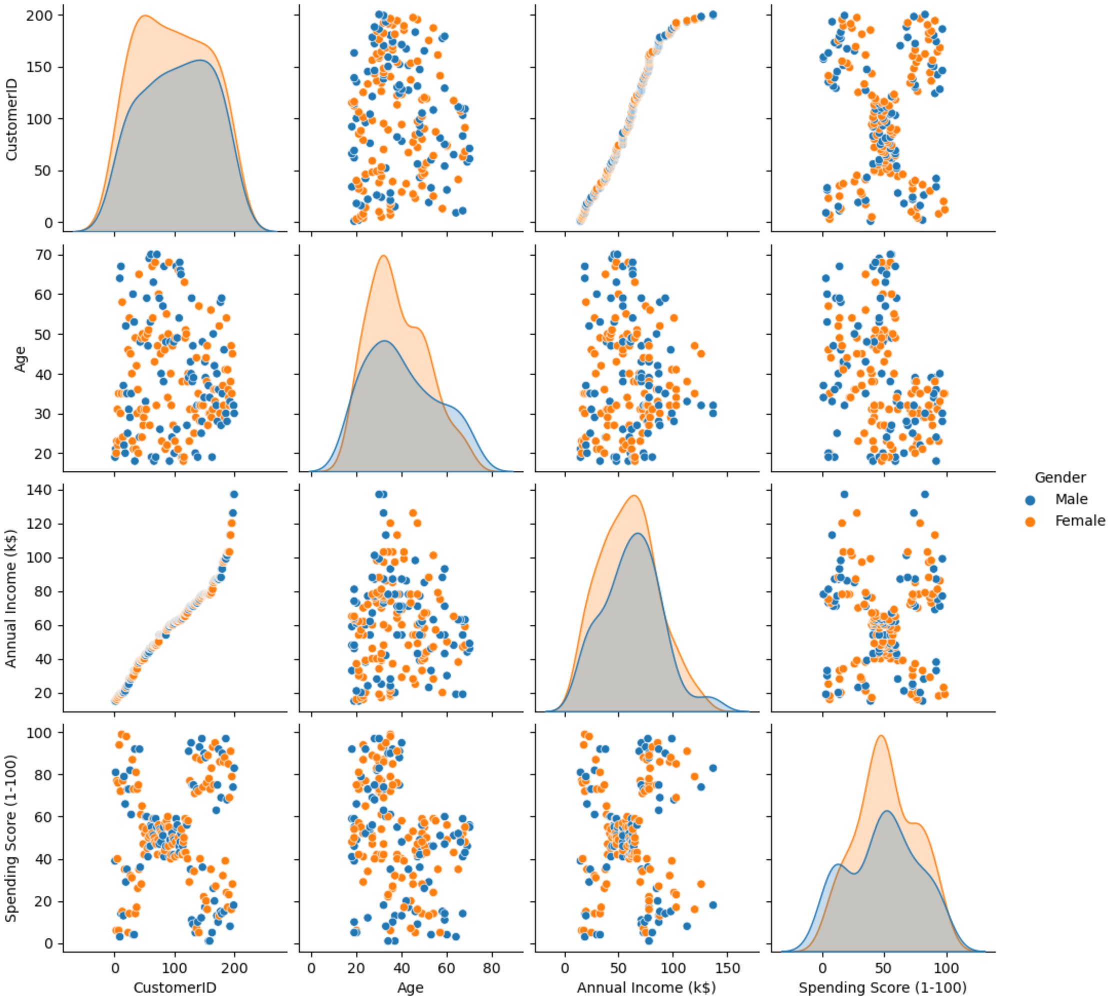
```
[12]: sns.scatterplot(data=df, x="Annual Income (k$)", y='Spending Score (1-100)')
```

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



```
[13]: #df=df.drop("CustomerID",axis=1) <- Don't need to run this again
      sns.pairplot(df, hue="Gender")
```

```
[13]: <seaborn.axisgrid.PairGrid at 0x21462f8e620>
```



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

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

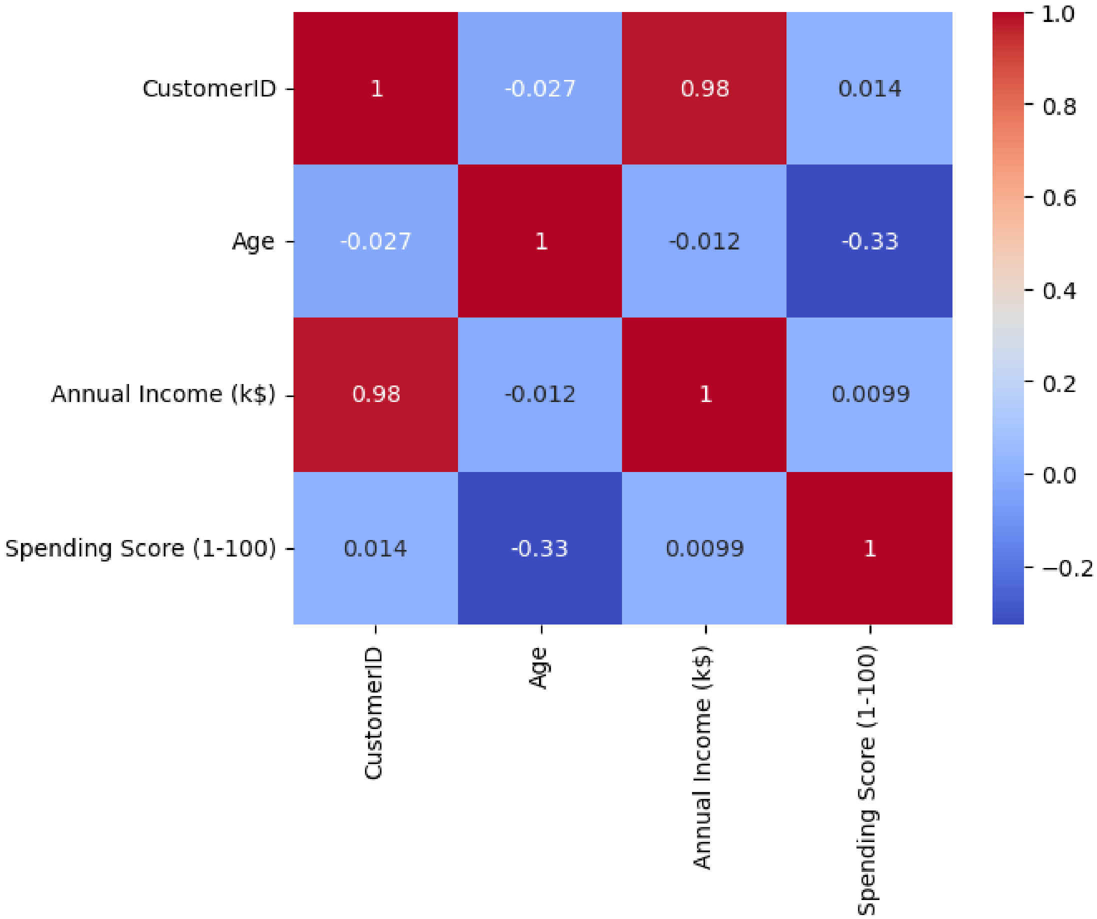
```
[15]: df.corr() # <- Correlation function
```

```
[15]:                         CustomerID       Age  Annual Income (k$)  \
      CustomerID                1.000000 -0.026763            0.977548
      Age                      -0.026763  1.000000           -0.012398
      Annual Income (k$)        0.977548 -0.012398            1.000000
      Spending Score (1-100)    0.013835 -0.327227            0.009903
```

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

[16]:
```
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
```

[16]:
```
<Axes: >
```



3 Clustering - Univariate, Bivariate, Multivariate

[68]:
```
clustering1 = KMeans(n_clusters=3)
```

[69]:
```
clustering1.fit(df[["Annual Income (k$)"]])
```

[69]:
```
KMeans(n_clusters=3)
```

[70]:

```
clustering1.labels_
```

[70]:

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 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, 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, 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])
```

[71]:

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

[71]:

```
   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

   Income Cluster
0               1
1               1
2               1
3               1
4               1
```

[72]:

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

[72]:

```
0    90
1    74
2    36
Name: Income Cluster, dtype: int64
```

[73]:

```
clustering1.inertia_ #Inertia represents is the distance between centroids
```

[73]:

```
23517.33093093093
```

[74]:

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

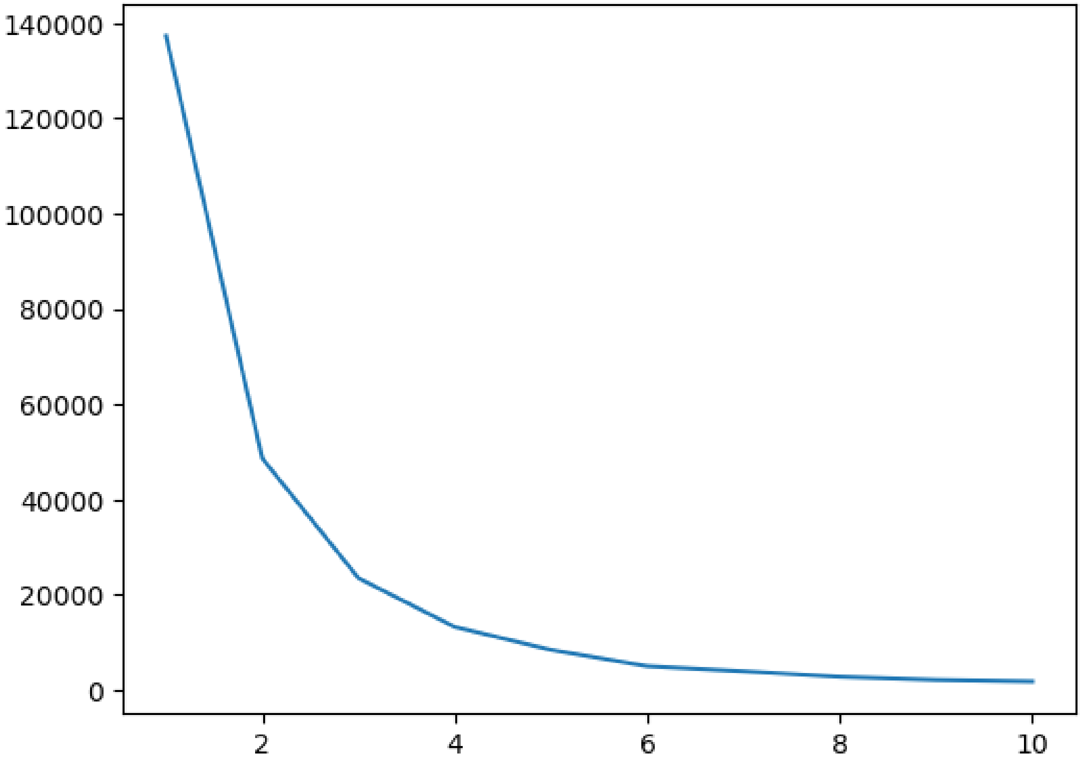
[75]:

```
intertia_scores
```

```
[75]: [137277.28000000003,
       48660.88888888888,
       23517.33093093093,
       13278.112713472487,
       8481.496190476191,
       5050.9047619047615,
       3976.358363858364,
       2822.499694749695,
       2173.287445887446,
       1859.0235042735042]
```

```
[76]: plt.plot(range(1,11),intertia_scores)
```

```
[76]: [<matplotlib.lines.Line2D at 0x2146930a800>]
```



```
[77]: df.columns
```

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

```
[78]: df.groupby("Income Cluster")['Age', 'Annual Income (k$)', 'Spending Score␣
       ↪(1-100)'].mean()
```

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

```
[79]: #Bivariate Clustering
```

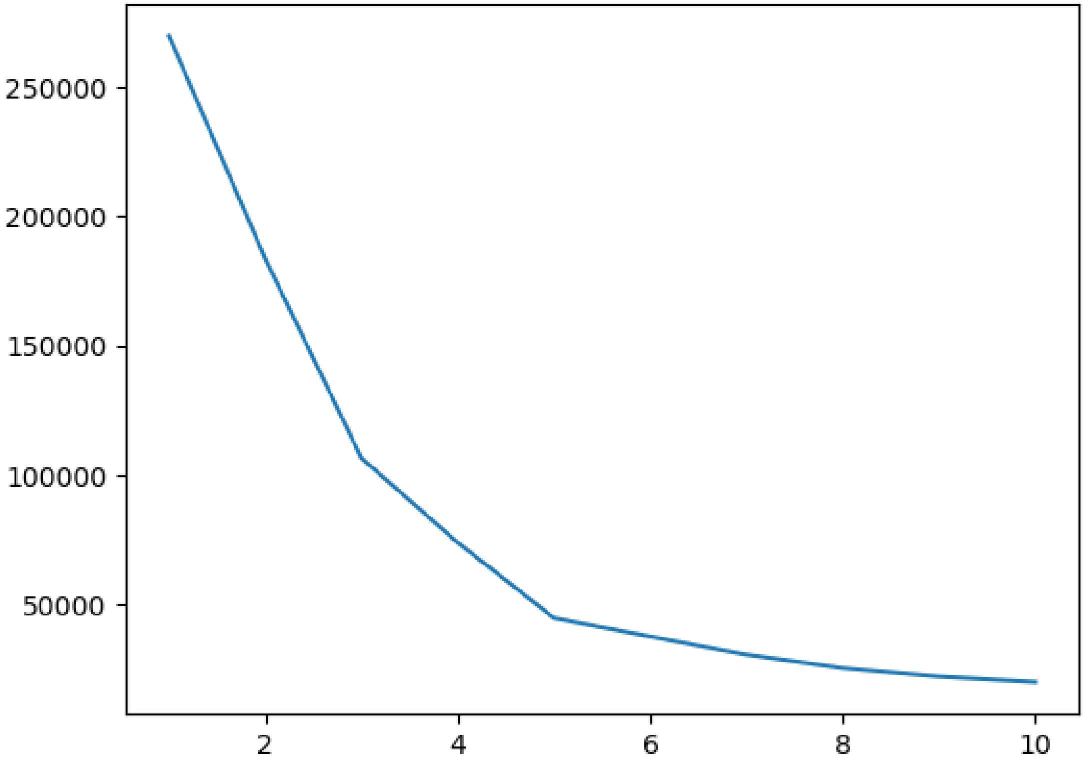
```
[84]: clustering2 = KMeans(n_clusters=5)
      clustering2.fit(df[['Annual Income (k$)','Spending Score (1-100)']])
      df["Spending and Income Cluster"] = clustering2.labels_
      df.head()
```

```
[84]:    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

         Income Cluster  Spending and Income Cluster
      0               1                            3
      1               1                            1
      2               1                            3
      3               1                            1
      4               1                            3
```

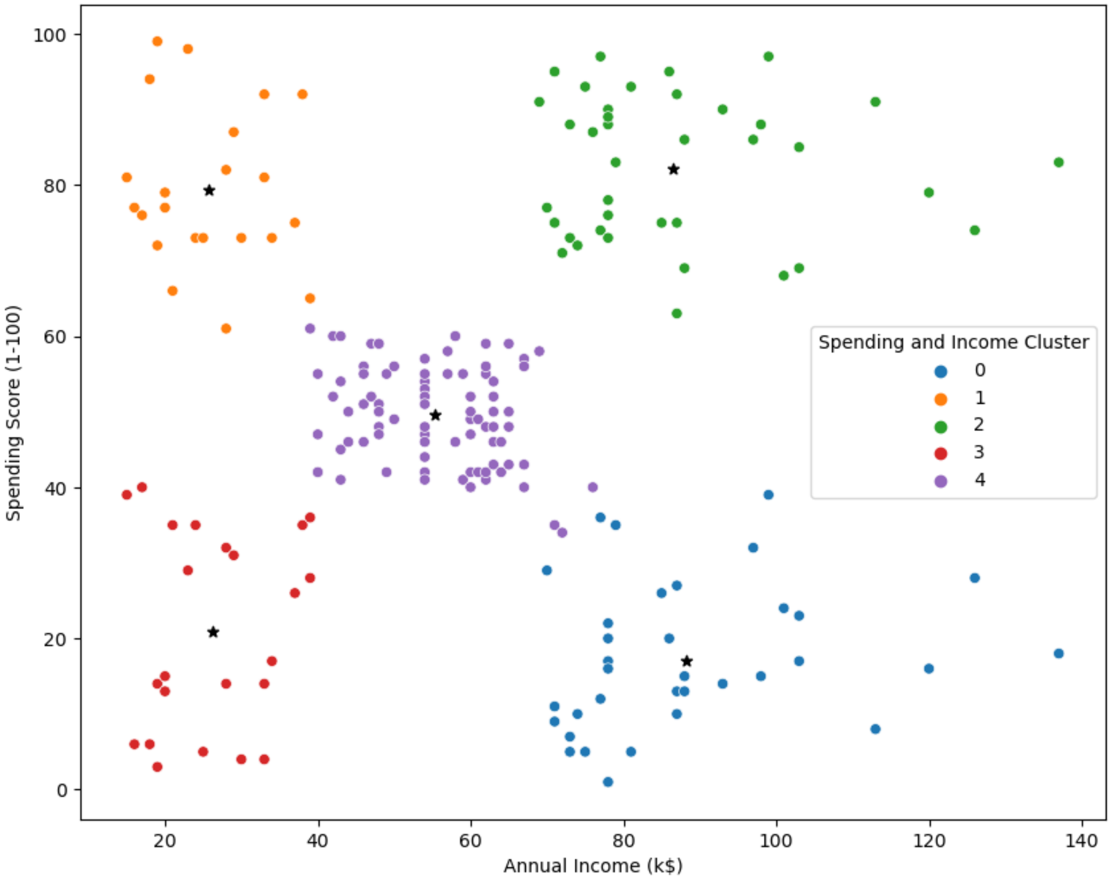
```
[85]: intertia_scores2=[]
      for i in range(1,11):
          kmeans2 = KMeans(n_clusters=i)
          kmeans2.fit(df[["Annual Income (k$)","Spending Score (1-100)"]])
          intertia_scores2.append(kmeans2.inertia_)
      plt.plot(range(1,11),intertia_scores2)
```

```
[85]: [<matplotlib.lines.Line2D at 0x214694793f0>]
```



[100]:
```
centers = pd.DataFrame(clustering2.cluster_centers_)
centers.columns = ["x","y"]
```

[116]:
```
plt.figure(figsize = (10,8))
plt.scatter(x=centers["x"], y=centers["y"],c="black",marker="*")
sns.scatterplot(data=df, x = "Annual Income (k$)", y = "Spending Score␣
 ↪(1-100)", hue = "Spending and Income Cluster", palette = "tab10")
plt.savefig("Clustering_bivariate.png")
```



[104]:
```
pd.crosstab(df["Spending and Income Cluster"],df["Gender"], normalize="index")
```

[104]:
```
Gender                          Female      Male
Spending and Income Cluster
0                             0.457143  0.542857
1                             0.590909  0.409091
2                             0.538462  0.461538
3                             0.608696  0.391304
4                             0.592593  0.407407
```

[105]:
```
df.groupby("Spending and Income Cluster")["Age","Annual Income (k$)",
       "Spending Score (1-100)"].mean()
```

[105]:
```
                                   Age  Annual Income (k$)  \
Spending and Income Cluster
0                            41.114286           88.200000
1                            25.272727           25.727273
2                            32.692308           86.538462
3                            45.217391           26.304348
```

```
4                            42.716049           55.296296

                             Spending Score (1-100)
Spending and Income Cluster
0                                         17.114286
1                                         79.363636
2                                         82.128205
3                                         20.913043
4                                         49.518519
```

```
[106]: #multivariate clustering
       from sklearn.preprocessing import StandardScaler
```

```
[107]: scale = StandardScaler()
```

```
[108]: df.head()
```

```
[108]:    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

          Income Cluster  Spending and Income Cluster
       0               1                            3
       1               1                            1
       2               1                            3
       3               1                            1
       4               1                            3
```

```
[109]: #dff = pd.get_dummies(df) <- return values from female and male, but just we␣
        ↪need one value, use drop for that
       dff = pd.get_dummies(df,drop_first=True)
       dff.head()
```

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

          Income Cluster  Spending and Income Cluster  Gender_Male
       0               1                            3            1
       1               1                            1            1
       2               1                            3            0
```

```
3               1                            1            0
4               1                            3            0
```

```
[110]: dff.columns
```

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

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

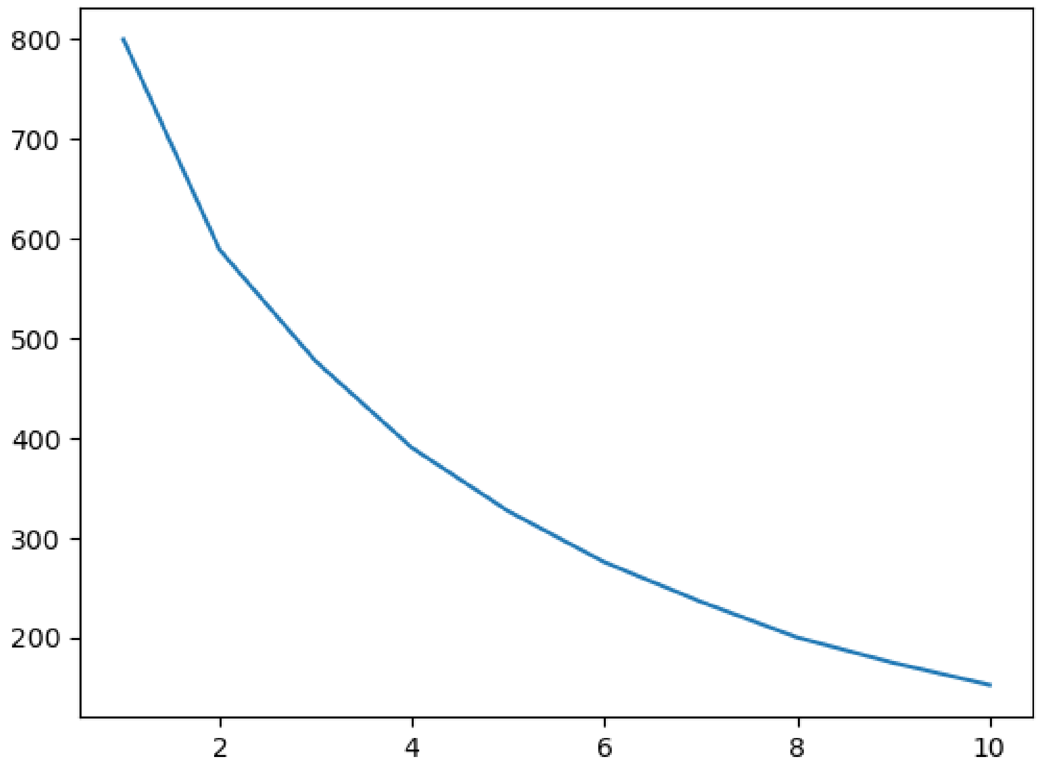
```
[111]:    Age  Annual Income (k$)  Spending Score (1-100)  Gender_Male
       0   19                  15                      39            1
       1   21                  15                      81            1
       2   20                  16                       6            0
       3   23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

```
[112]: dff = pd.DataFrame(scale.fit_transform(dff))
       dff.head()
```

```
[112]:           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
```

```
[113]: 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)
```

```
[113]: [<matplotlib.lines.Line2D at 0x2146bb31a50>]
```



```
[115]: df
```

```
[115]:      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

            Income Cluster  Spending and Income Cluster
       0                 1                            3
       1                 1                            1
       2                 1                            3
       3                 1                            1
       4                 1                            3
       ..              …                          …
```

```
195                2                             2
196                2                             0
197                2                             2
198                2                             0
199                2                             2

[200 rows x 7 columns]
```

```
[117]: df.to_csv("Clustering.csv")
```

```
[ ]:
```