CustSeg

January 22, 2024

1 Mall Customer Segment Cluster Analysis

1.1 Import Libraries

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

[2]:
```
# Read data in mall_customers.csv into a dataframe
mall = pd.read_csv('../sample-notebooks/Mall_Customers.csv')
```

[3]:
```
# Display first 5 rows of mall dataframe
mall.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

[7]:
```
# Display information about dataframe
mall.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
```

```
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

1.2 Univariate Analysis

Analysis looking at one variable

[9]:
```
# Basic statistics on dataframe
mall.describe().transpose()
```

[9]:

	count	mean	std	min	25%	50%	75%	\
CustomerID	200.0	100.50	57.879185	1.0	50.75	100.5	150.25	
Age	200.0	38.85	13.969007	18.0	28.75	36.0	49.00	
Annual Income (k$)	200.0	60.56	26.264721	15.0	41.50	61.5	78.00	
Spending Score (1-100)	200.0	50.20	25.823522	1.0	34.75	50.0	73.00	

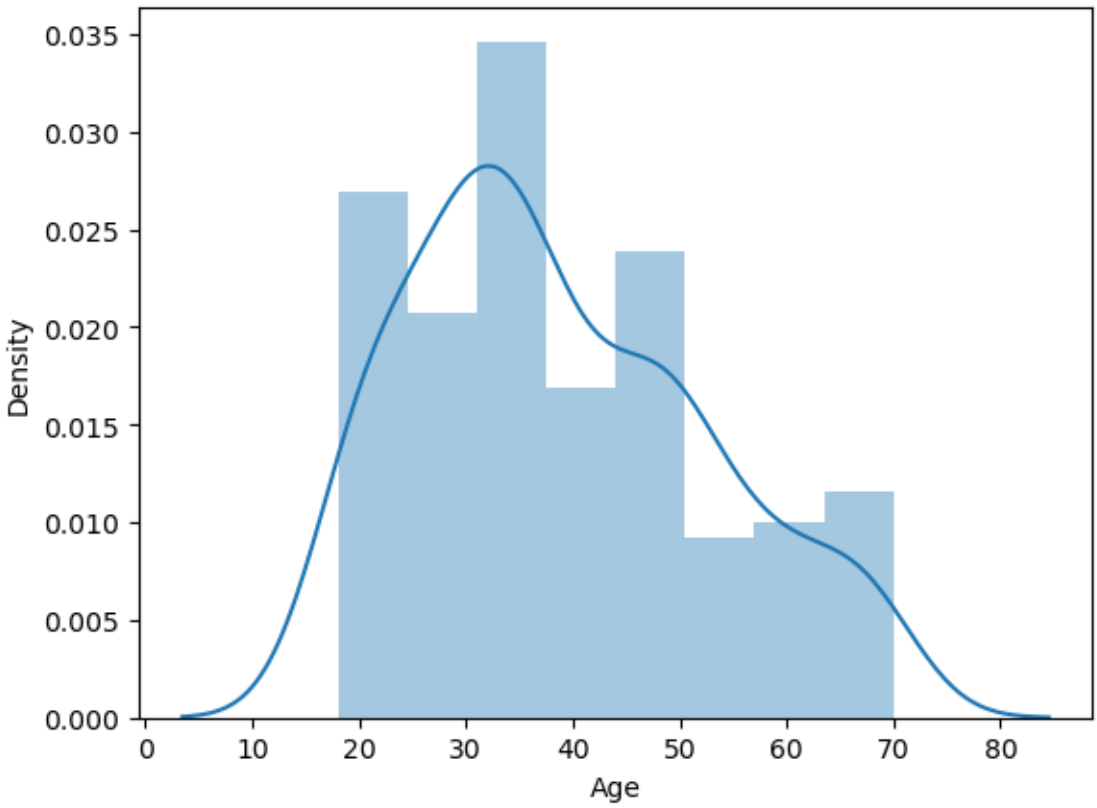
	max
CustomerID	200.0
Age	70.0
Annual Income (k$)	137.0
Spending Score (1-100)	99.0

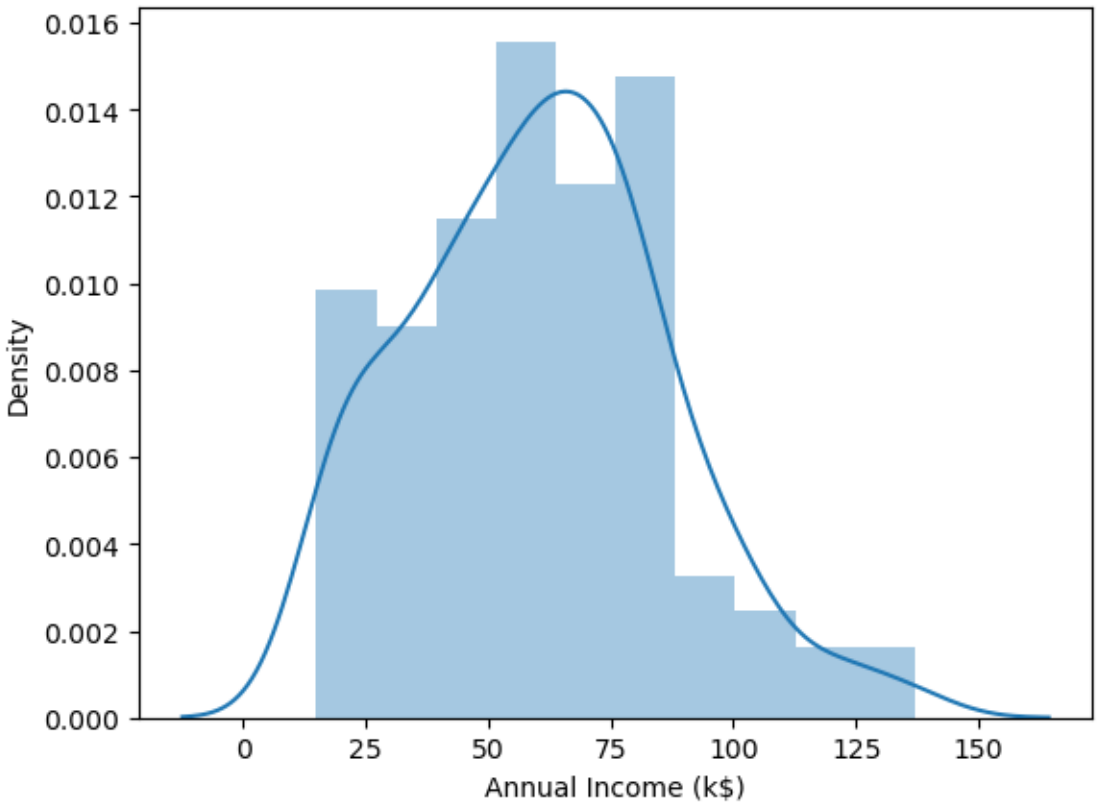
[11]:
```
# Get a list of the columns in the mall dataframe
mall.columns
```

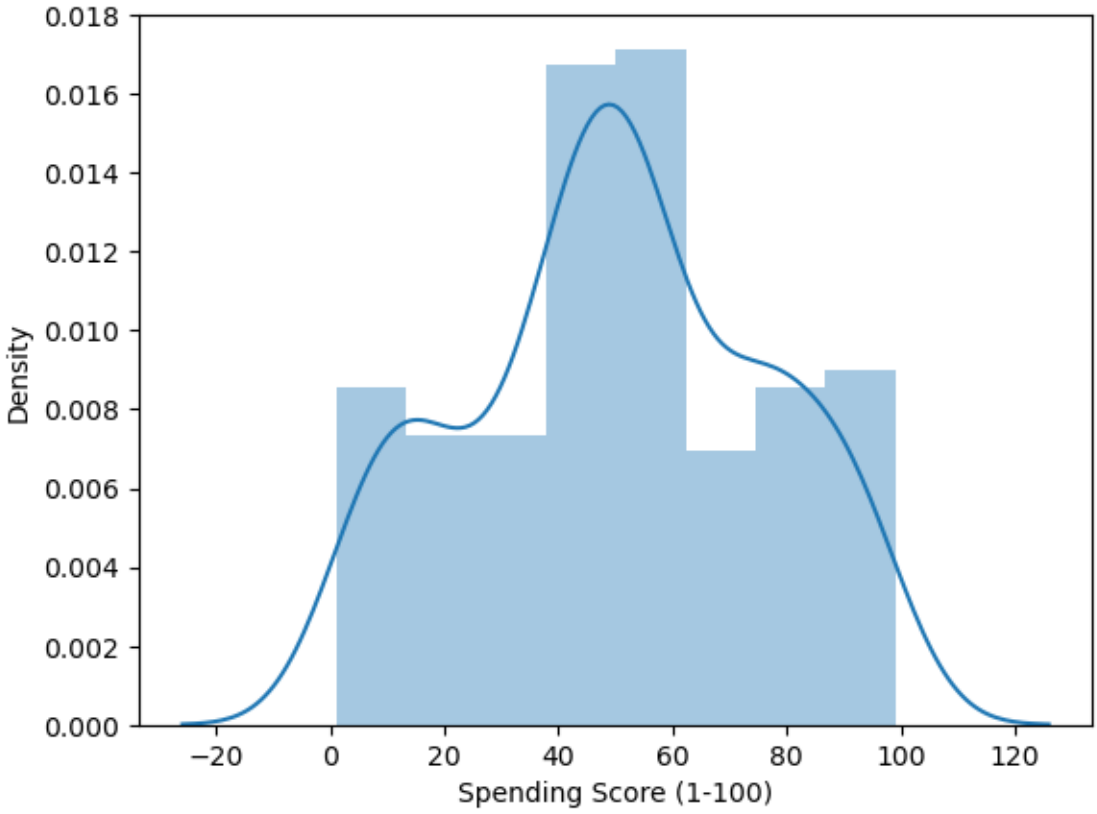
[11]:
```
Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

[13]:
```
# create a list variable with the numeric column names
columns = ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']
```

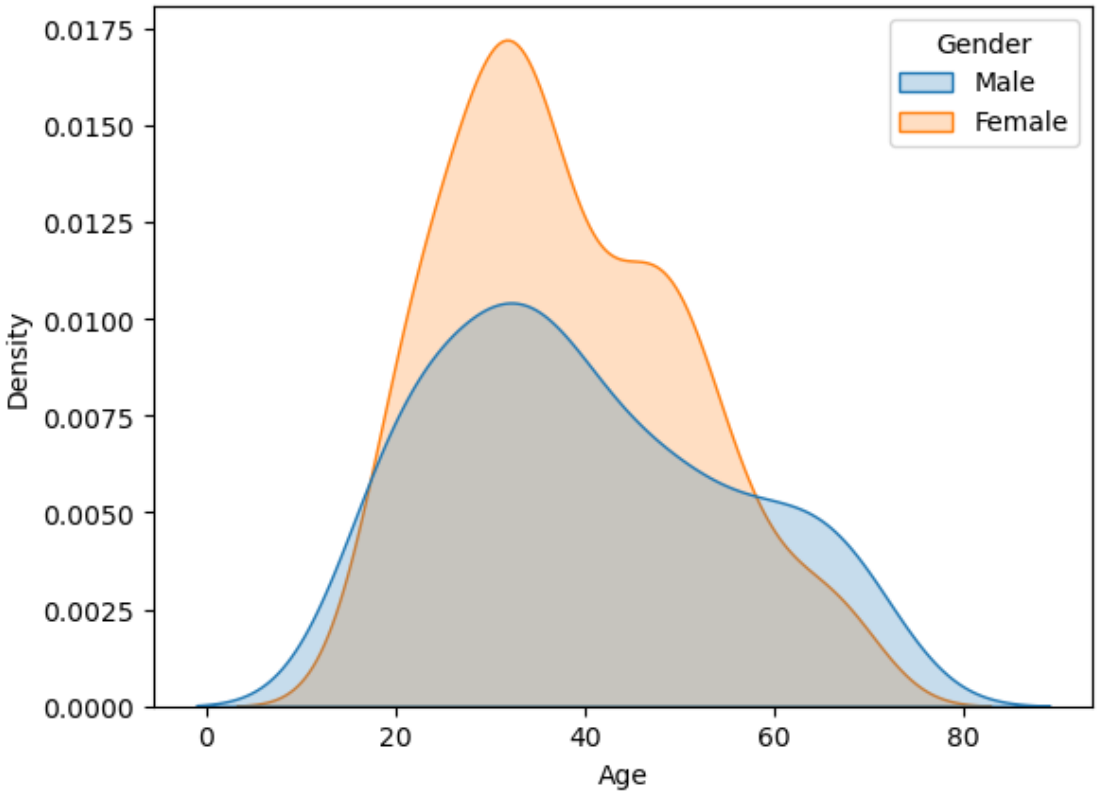
[15]:
```
# Create histograms for Age, Annual Income and Spending Score
for i in columns:
    plt.figure()
    sns.distplot(mall[i])
```

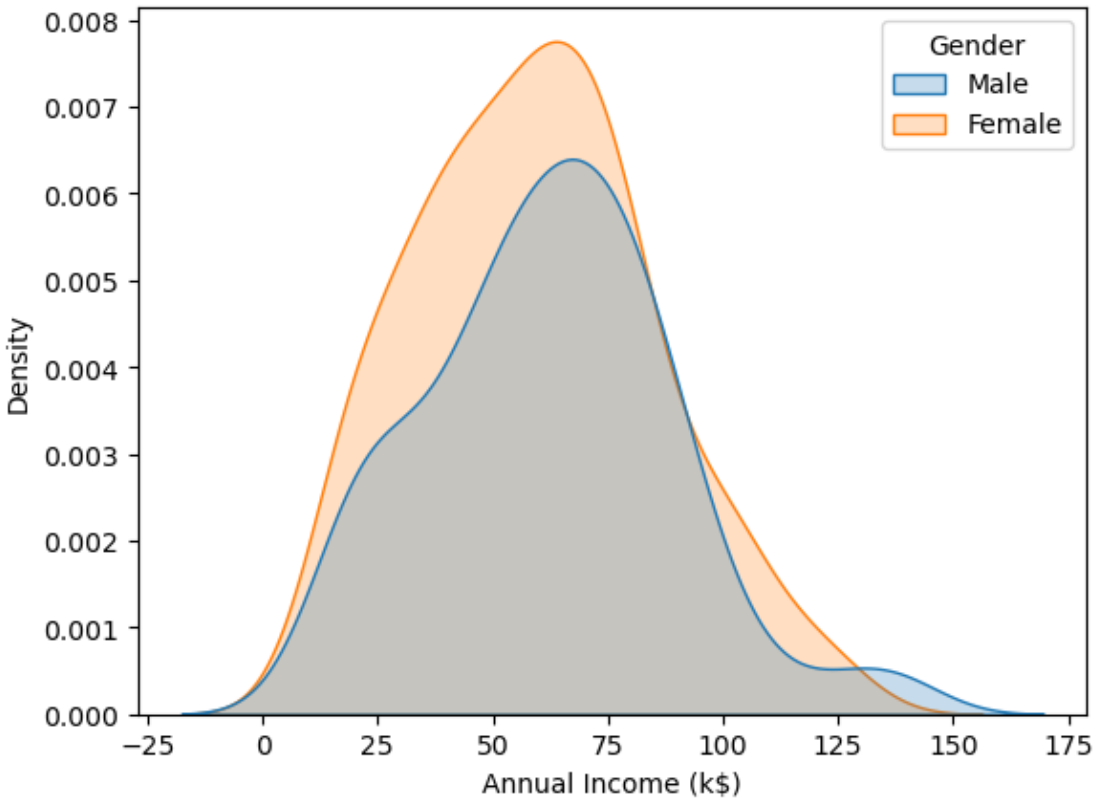

0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
0
10
20
30
40
50
60
70
80
Age

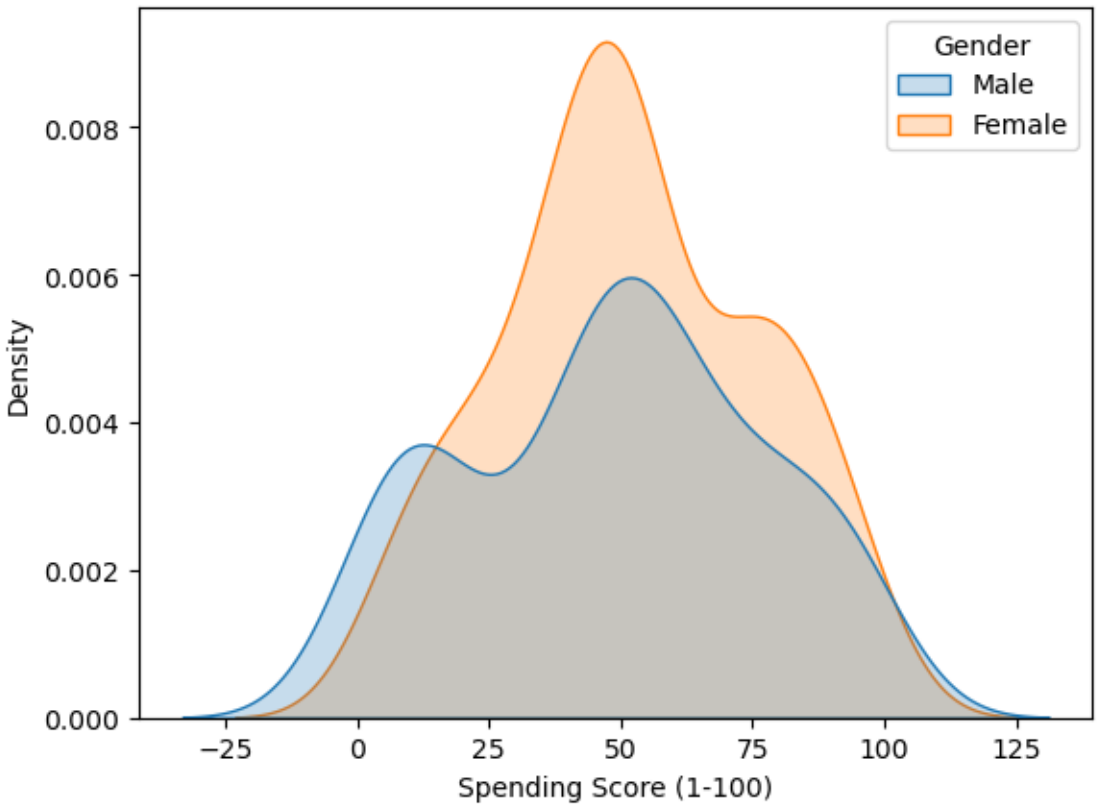

0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
Density
0
25
50
75
100
125
150
Annual Income (k$)



[17]:
```
# Create kdeplots for Age, Annual Income and Spending Score
for i in columns:
    plt.figure()
    sns.kdeplot(data=mall, x=mall[i],shade=True, hue=mall['Gender'])
```

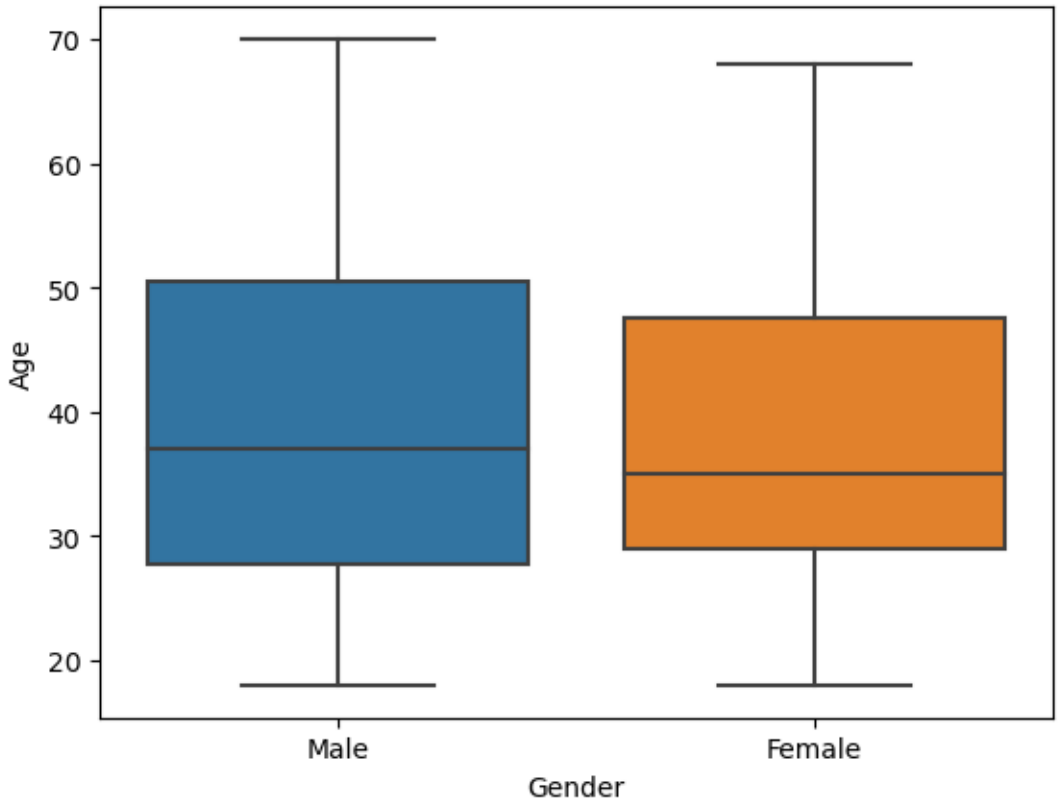

Gender
Male
Female
Density
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
20
40
60
80
Age

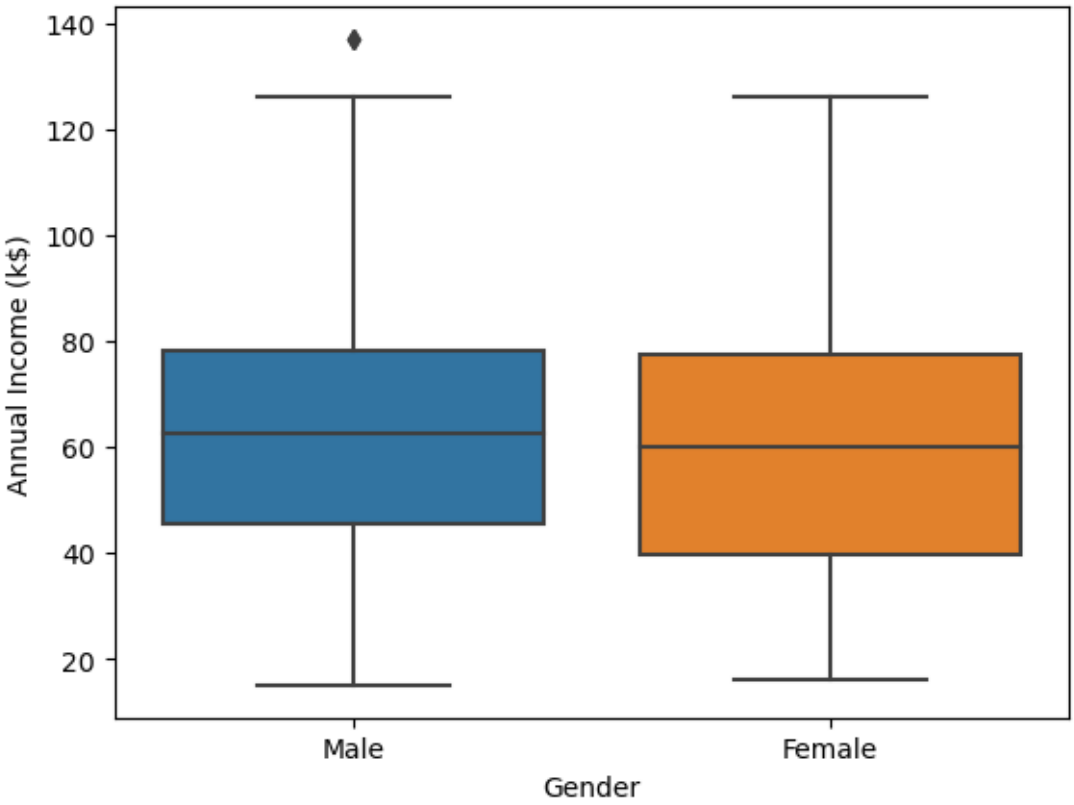

Gender
Male
Female
Density
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
−25
0
25
50
75
100
125
150
175
Annual Income (k$)

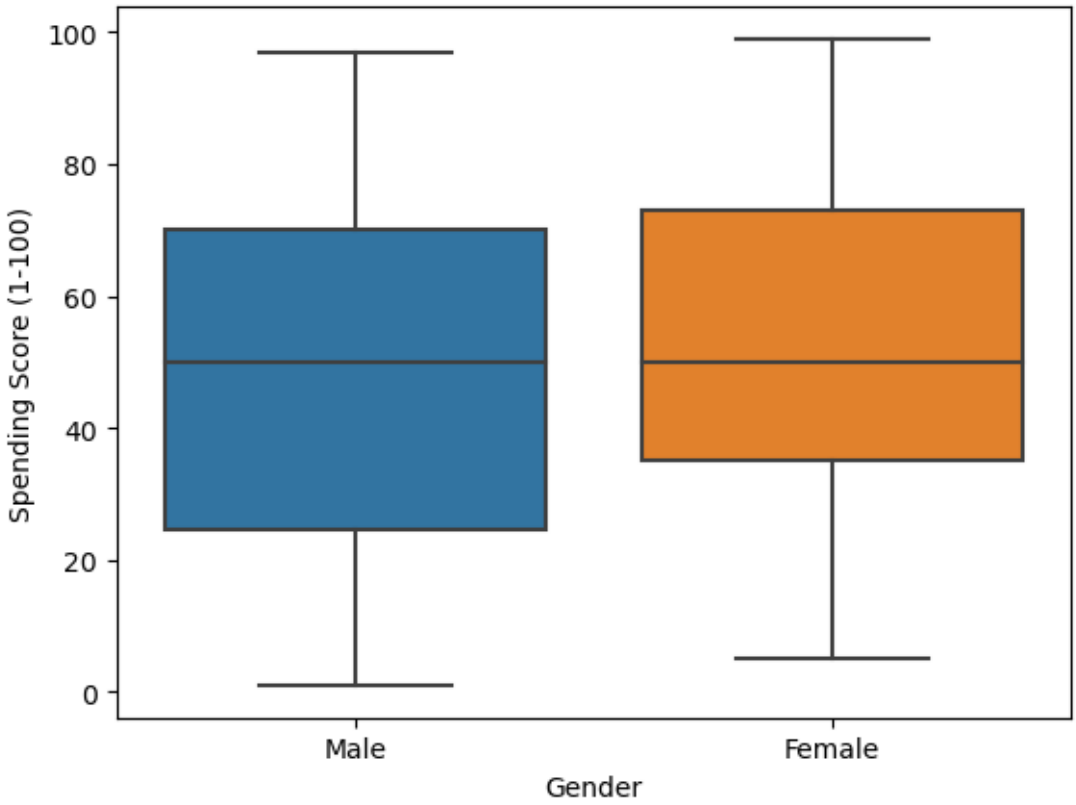


[19]:
```
# Create boxplots for Age, Annual Income and Spending Score
for i in columns:
    plt.figure()
    sns.boxplot(data=mall, x='Gender', y=mall[i])

# outlier value for male Annual Income (k$)
```


70
60
50
40
30
20
Age
Male
Female
Gender


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



[21]:
```
# Count of Gender values
mall['Gender'].value_counts()
```

[21]:
```
Gender
Female    112
Male       88
Name: count, dtype: int64
```

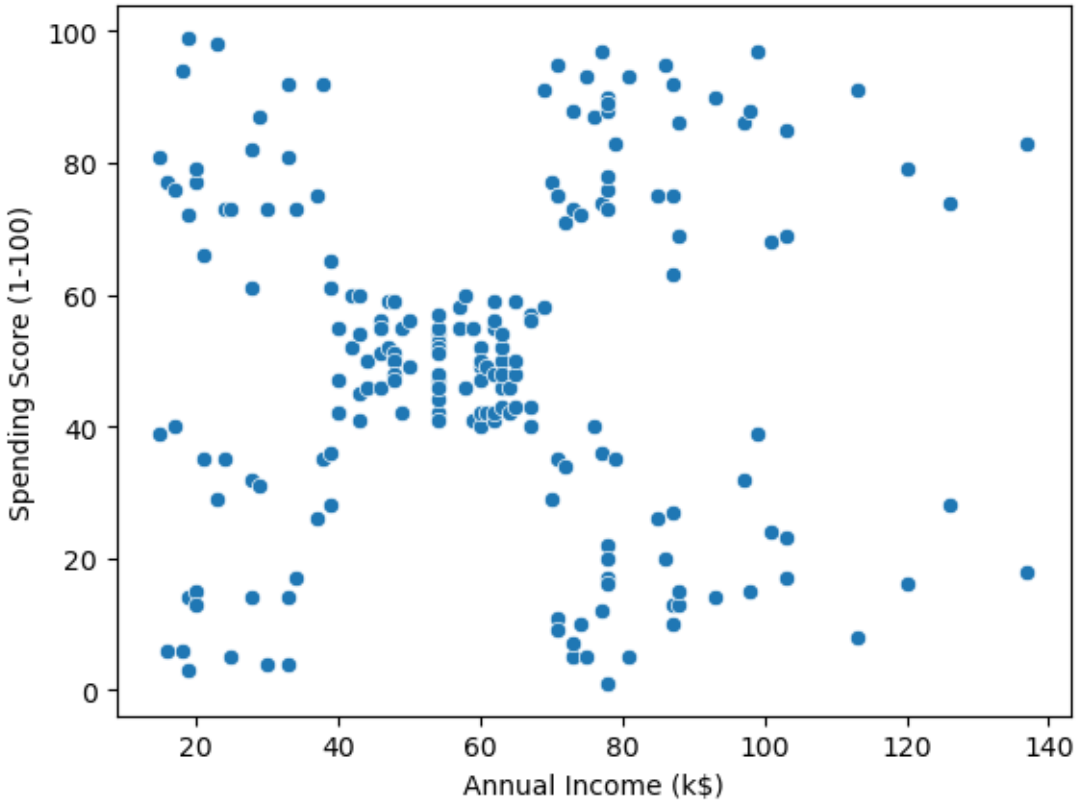
[23]:
```
# Percentage for Gender values
mall['Gender'].value_counts(normalize=True)
```

[23]:
```
Gender
Female    0.56
Male      0.44
Name: proportion, dtype: float64
```

1.3 Bivariate Analysis

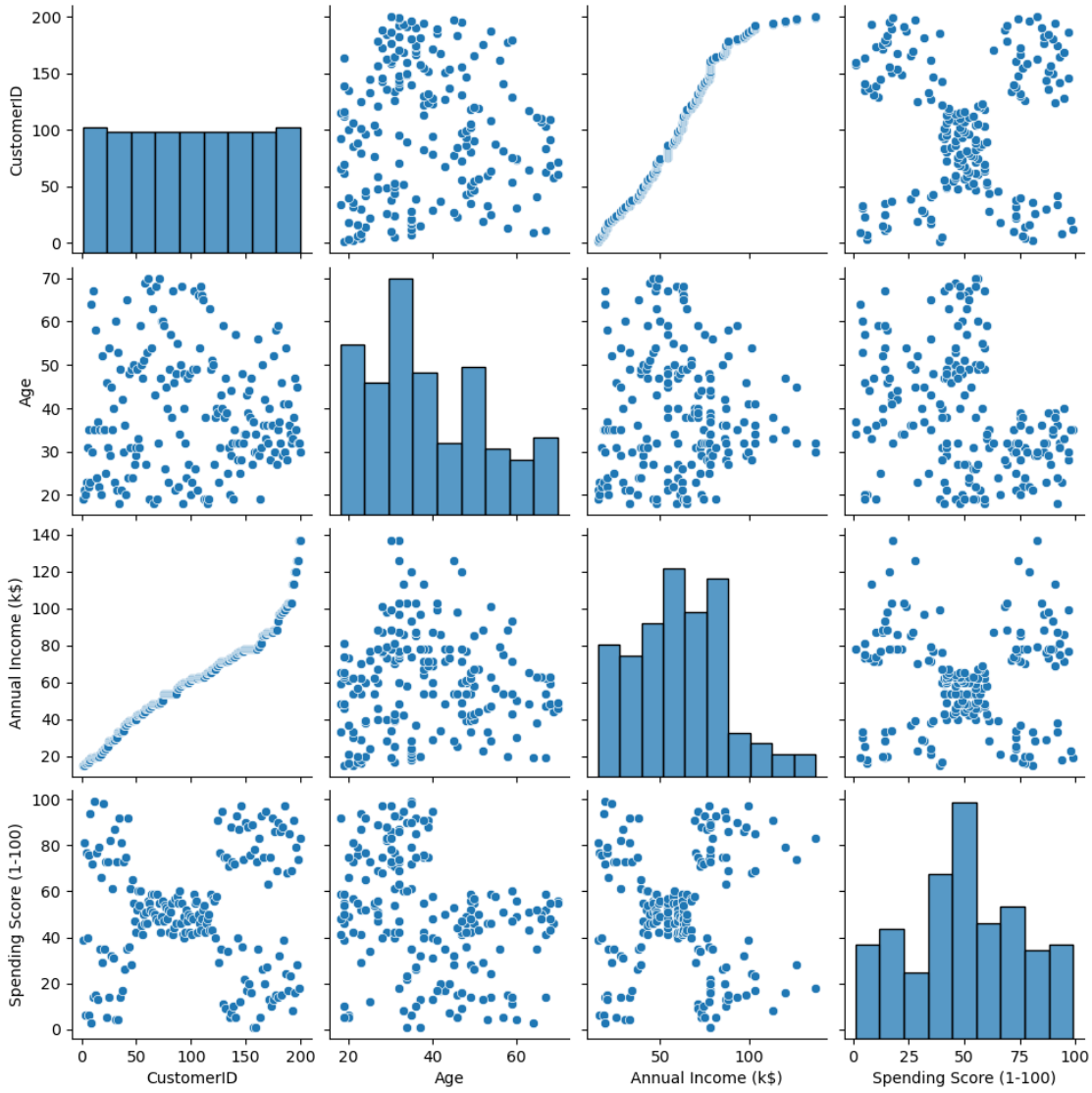
Analysis comparing two variables

[25]:
```
# Scatterplot of Annual Income (k$) and Spending Score (1-100)
sns.scatterplot(data=mall, x = 'Annual Income (k$)', y = 'Spending Score␣
↪(1-100)')
plt.show()
```

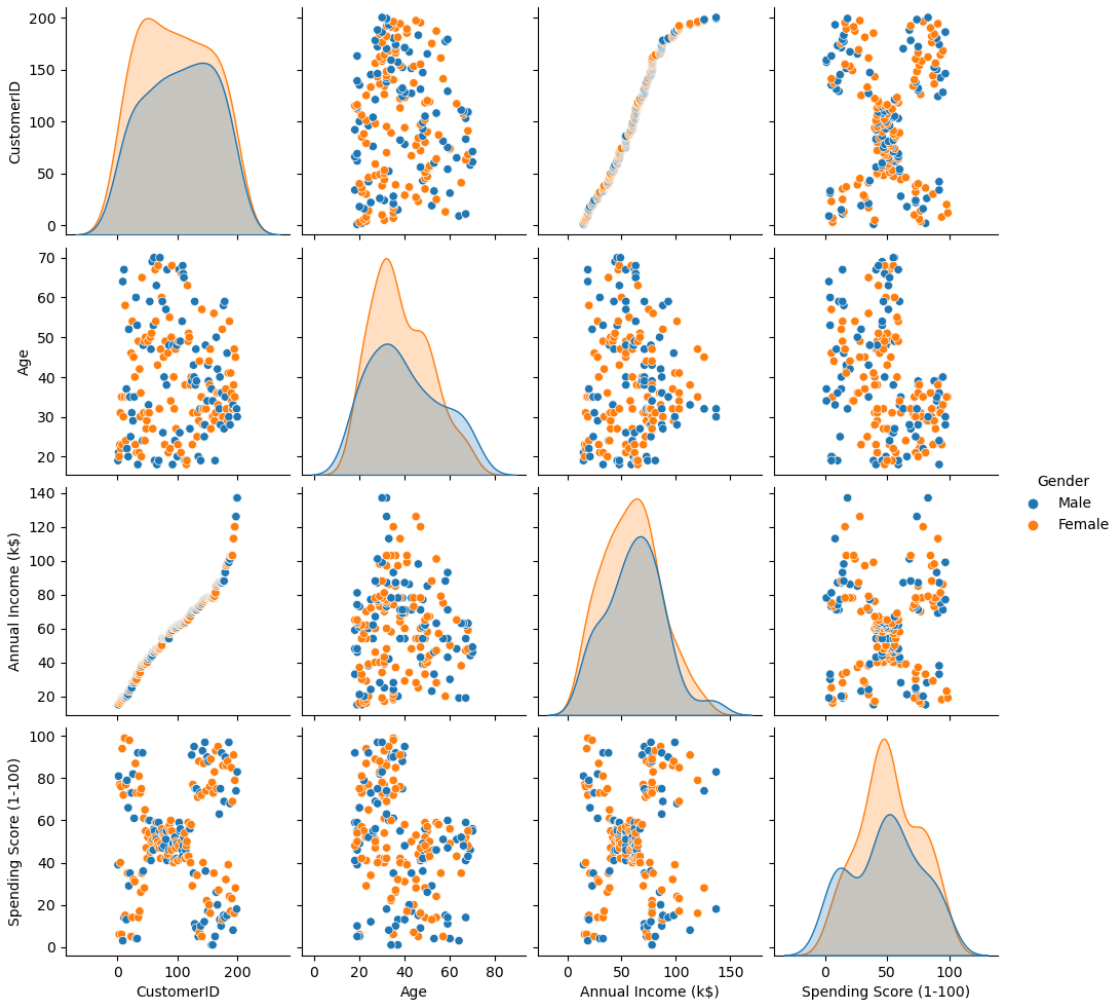


[27]:
```
# Drop CustomerID.  CustomerID does not add any value
#mall = mall.drop('CustomerID', axis=1)

# Pairplot.  all combinations of features
sns.pairplot(mall)
plt.show()
```



[29]:
```
# Pairplot.  all combinations of features.  Separate by Gender
sns.pairplot(mall, hue='Gender')
plt.show()
```



[31]:
```
# Mean values by Gender
mall.groupby('Gender')[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']].
↪mean()
```

[31]:

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

[33]:
```
# Correlation
mall.corr(numeric_only = True)
```

[33]:

	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
```

[35]:
```
# Graph correlation using a heatmap
sns.heatmap(mall.corr(numeric_only=True),cmap="YlGnBu", annot=True)
```

[35]: <Axes: >

1.4 Clustering

Clustering is a type of unsupervised learning methood that groups unlabeled examples into groups based on a similiarities

1.5 Clustering - Univariate

Clustering based on one feature

[49]:
```
# k-means algorithm - Clustering1 - One feature
clustering1 = KMeans(n_clusters=3, random_state=42)
```

[51]:
```
# k-means algorithm - Annual Income (k$)
clustering1.fit(mall[['Annual Income (k$)']])
```

[51]:
```
KMeans(n_clusters=3, random_state=42)
```

[53]:
```
clustering1.labels_
```

[53]:
```
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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 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, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2], dtype=int32)
```

[55]:
```
mall['Income Cluster'] = clustering1.labels_
mall.head()
```

[55]:

	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

[57]:
```
mall['Income Cluster'].value_counts()
```

[57]:
```
Income Cluster
1    90
0    74
2    36
```

```
Name: count, dtype: int64
```

[59]:
```
# Score.  distance between the centroids
clustering1.inertia_
```

[59]:
```
23517.330930930926
```

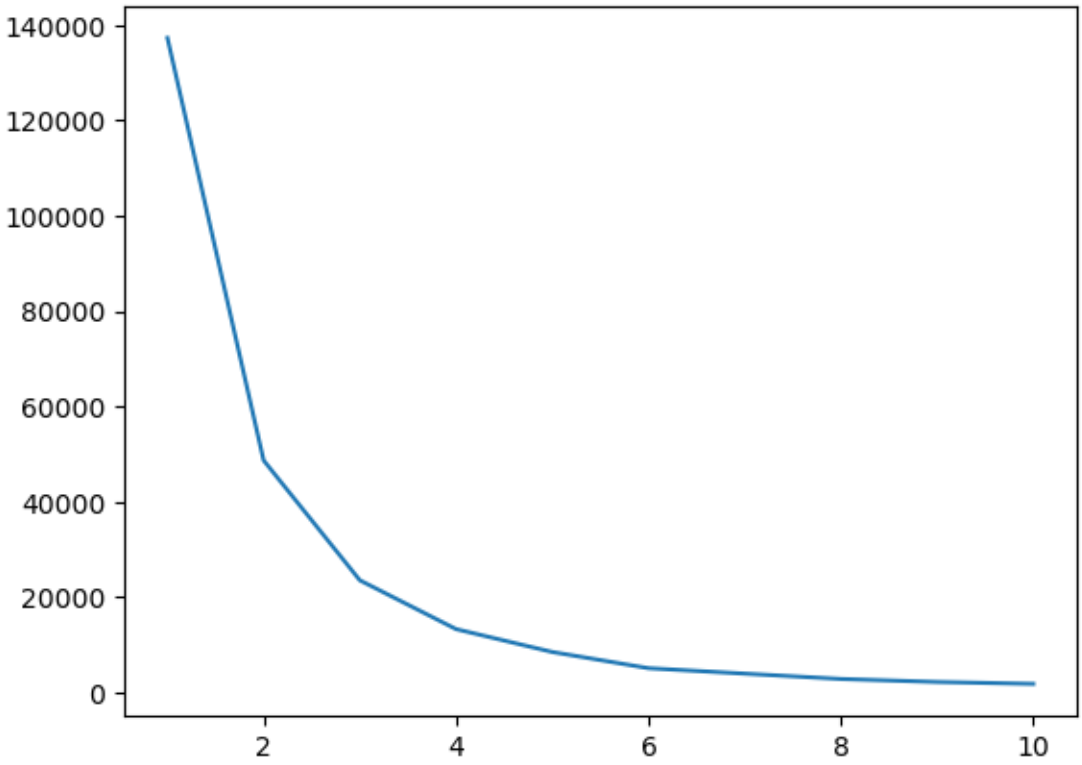
[61]:
```
# Determine how many clusters to use for KMeans
# Calculate the intertia scores for a range
intertia_scores = []
for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(mall[['Annual Income (k$)']])
    intertia_scores.append(kmeans.inertia_)
```

[63]:
```
# Display calculated scores
intertia_scores
```

[63]:
```
[137277.28,
 48660.88888888889,
 23517.330930930926,
 13278.112713472488,
 8481.49619047619,
 5081.484660267269,
 3941.4163614163617,
 2822.4996947496943,
 2193.0907275730815,
 1796.5129870129872]
```

[65]:
```
# Plot calculated scores
# looks like elbow starts at 3, recalculate KMeans using 3 clusters
plt.plot(range(1,11), intertia_scores)
```

[65]:
```
[<matplotlib.lines.Line2D at 0x7fa902fea890>]
```



[67]:
```
# Mean for Age, Annual Income, & Spending Score grouped by Income Cluster
mall.groupby('Income Cluster')[['Age', 'Annual Income (k$)', 'Spending Score␣
↪(1-100)']].mean()
```

[67]:

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

[155]:
```
# Swarmplot for Annual Income (k$)
sns.set(rc={'figure.figsize':(5,5)})
sns.swarmplot(x=mall['Annual Income (k$)'], y= mall['Gender'], hue=mall['Income␣
↪Cluster']).set(title='Income Cluster By Gender')
```

[155]:
```
[Text(0.5, 1.0, 'Income Cluster By Gender')]
```

The above graph shows the distribution of the three income clusters separated by gender. Income is concentrated below $80K.

1.6 Clustering - Bivariate

Clustering looking at two features

[122]:
```
# kmeans algorithm - Clustering2 - Bivariate - Two features
clustering2 = KMeans(n_clusters=5, random_state = 42)
clustering2.fit(mall[['Annual Income (k$)', 'Spending Score (1-100)']])
clustering2.labels_
mall['Spending and Income Cluster'] = clustering2.labels_
mall.head()
```

[122]:

	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	

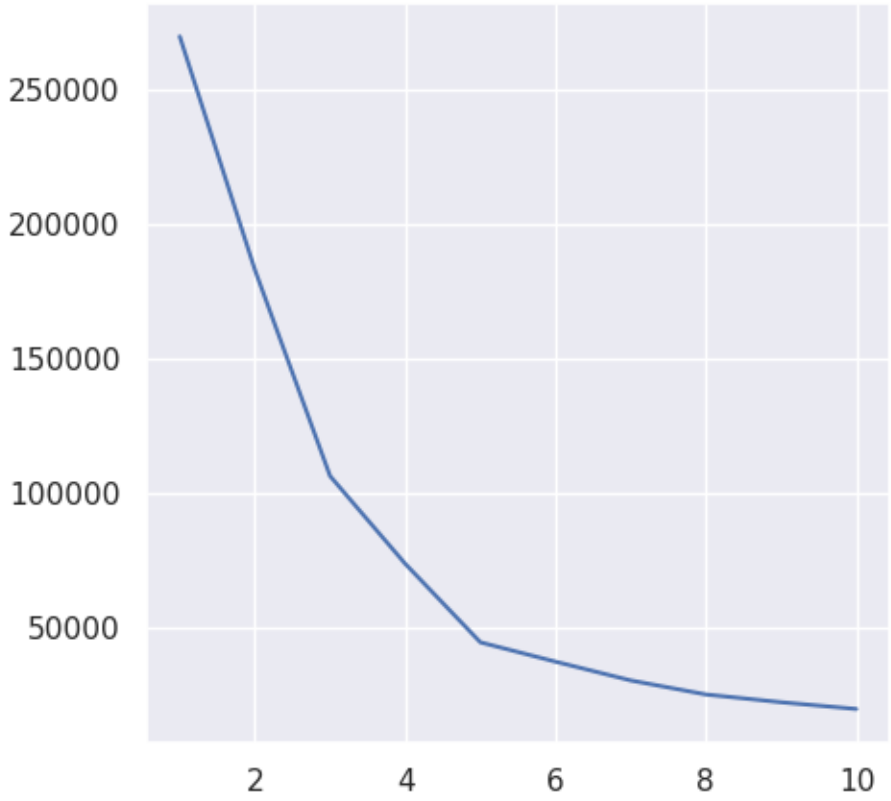
```
4           5  Female   31                  17                         40

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

[124]:
```
 # Determine how many clusters to use for KMeans
# Calculate the intertia scores for a range
intertia_scores2 = []
for i in range(1,11):
    kmeans2 = KMeans(n_clusters=i)
    kmeans2.fit(mall[['Annual Income (k$)', 'Spending Score (1-100)']])
    intertia_scores2.append(kmeans2.inertia_)

# Plot calculated scores
# looks like elbow starts at 5, recalculate KMeans using 5 clusters
plt.plot(range(1,11), intertia_scores2)
```

[124]:
```
[<matplotlib.lines.Line2D at 0x7fa8f9fe4dd0>]
```



[128]:
```
# Coordinates for cluster centers
centers = pd.DataFrame(clustering2.cluster_centers_)
centers.columns = ['x','y']
centers.head()
```

[128]:

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

[151]:
```
plt.figure(figsize=(10,8))
plt.scatter(x=centers['x'], y = centers['y'], s=100, c='black', marker='*')
sns.scatterplot(data=mall, x = 'Annual Income (k$)' , y = 'Spending Score␣
 ↪(1-100)', hue=
               'Spending and Income Cluster', palette = 'tab10').
 ↪set(title='Spending and Income Clusters')
# to save plot:
plt.savefig("../sample-notebooks/clustering_bivariate.png")
```

[132]:
```
# Spending and Income Cluster value counts by Gender
pd.crosstab(mall['Spending and Income Cluster'],mall['Gender'])
```

[132]:

Gender	Female	Male
Spending and Income Cluster		
0	48	33
1	21	18
2	13	9
3	16	19
4	14	9

[134]:
```
# Spending and Income Cluster percentage by Gender
pd.crosstab(mall['Spending and Income Cluster'],mall['Gender'], normalize=True)
```

[134]:

Gender	Female	Male
Spending and Income Cluster		
0	0.240	0.165
1	0.105	0.090

```
2                              0.065  0.045
3                              0.080  0.095
4                              0.070  0.045
```

[136]:
```
# Mean for Age, Annual Income, & Spending Score grouped by Income Cluster
mall.groupby('Spending and Income Cluster')[['Age', 'Annual Income (k$)',␣
 ↪'Spending Score (1-100)']].mean()
```

[136]:
```
                                    Age  Annual Income (k$)  \
Spending and Income Cluster
0                             42.716049           55.296296
1                             32.692308           86.538462
2                             25.272727           25.727273
3                             41.114286           88.200000
4                             45.217391           26.304348

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

[138]:
```
mall.head()
```

[138]:
```
   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               0                            4
1               0                            2
2               0                            4
3               0                            2
4               0                            4
```

[140]:
```
# Save file as csv
mall.to_csv("../sample-notebooks/clustering.csv")
```

[161]:
```
mall.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
```

```
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   CustomerID                    200 non-null    int64
 1   Gender                        200 non-null    object
 2   Age                           200 non-null    int64
 3   Annual Income (k$)            200 non-null    int64
 4   Spending Score (1-100)        200 non-null    int64
 5   Income Cluster                200 non-null    int32
 6   Spending and Income Cluster   200 non-null    int32
dtypes: int32(2), int64(4), object(1)
memory usage: 9.5+ KB
```

1.7 Recommendation: Target clusters 1 and 2.

Both clusters have spending scores above 60.

1.7.1 Group 1 (Cluster 1 - Orange)

[220]:
```
group1=mall[mall['Spending and Income Cluster']==1]
group1.head()
```

[220]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	Income Cluster	Spending and Income Cluster
123	124	Male	39	69	91	1	1
125	126	Female	31	70	77	1	1
127	128	Male	40	71	95	1	1
129	130	Male	38	71	75	1	1
131	132	Male	39	71	75	1	1

1.7.2 Group 1 (Cluster 1 - Orange): Average Age, Income, and Spending Score

[218]:
```
group1.groupby('Gender')[['Age', 'Annual Income (k$)', 'Spending Score␣
↪(1-100)']].mean().round(2)
```

[218]:

Gender	Age	Annual Income (k$)	Spending Score (1-100)
Female	32.19	86.05	81.67
Male	33.28	87.11	82.67

1.7.3 Group 2 (Cluster 2 - Green)

[227]:
```
group2=mall[mall['Spending and Income Cluster']==2]
group2.head()
```

[227]:
```
    CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)  \
1            2    Male   21                  15                      81
3            4  Female   23                  16                      77
5            6  Female   22                  17                      76
7            8  Female   23                  18                      94
9           10  Female   30                  19                      72

   Income Cluster  Spending and Income Cluster
1               0                            2
3               0                            2
5               0                            2
7               0                            2
9               0                            2
```

1.7.4 Group 2 (Cluster 2 - Green): Average Age, Income, Spending Score

[229]:
```
group2.groupby('Gender')[['Age', 'Annual Income (k$)', 'Spending Score␣
↪(1-100)']].mean().round(2)
```

[229]:
```
              Age  Annual Income (k$)  Spending Score (1-100)
Gender
Female  25.46               25.69                   80.54
Male    25.00               25.78                   77.67
```

[ ]:
```

```