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

```
df=pd.read_csv('/content/Mall_Customers.csv')
```

```
df.head()
```

	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

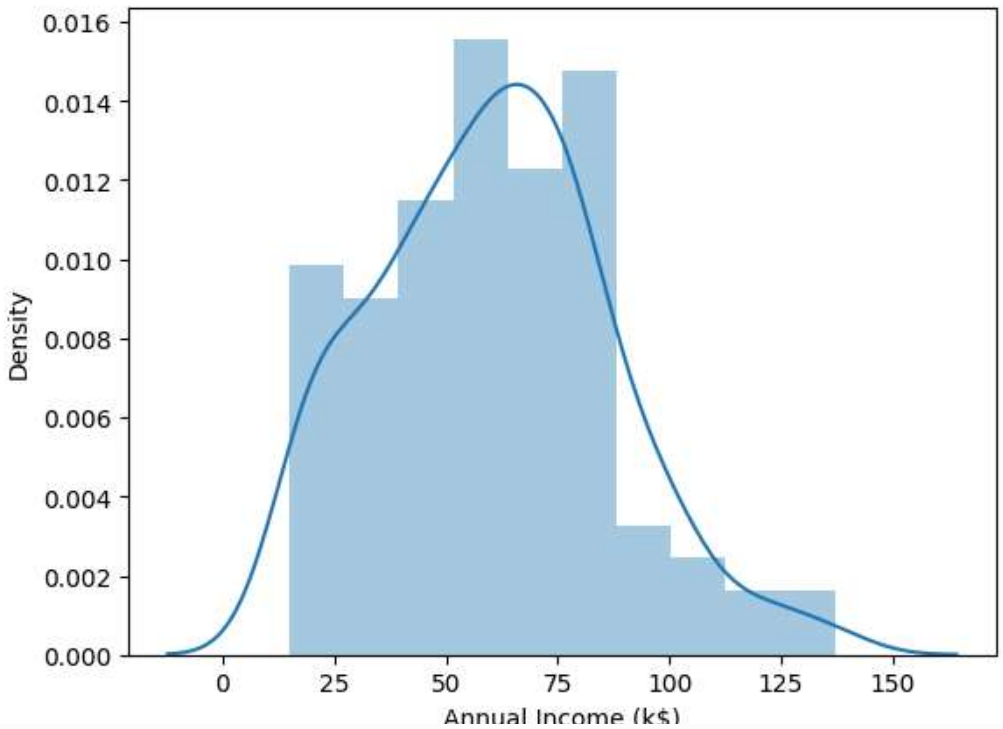
Next steps: Generate code with df | View recommended plots | New interactive sheet

Univariate Analysis

```
df.describe()
```

	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

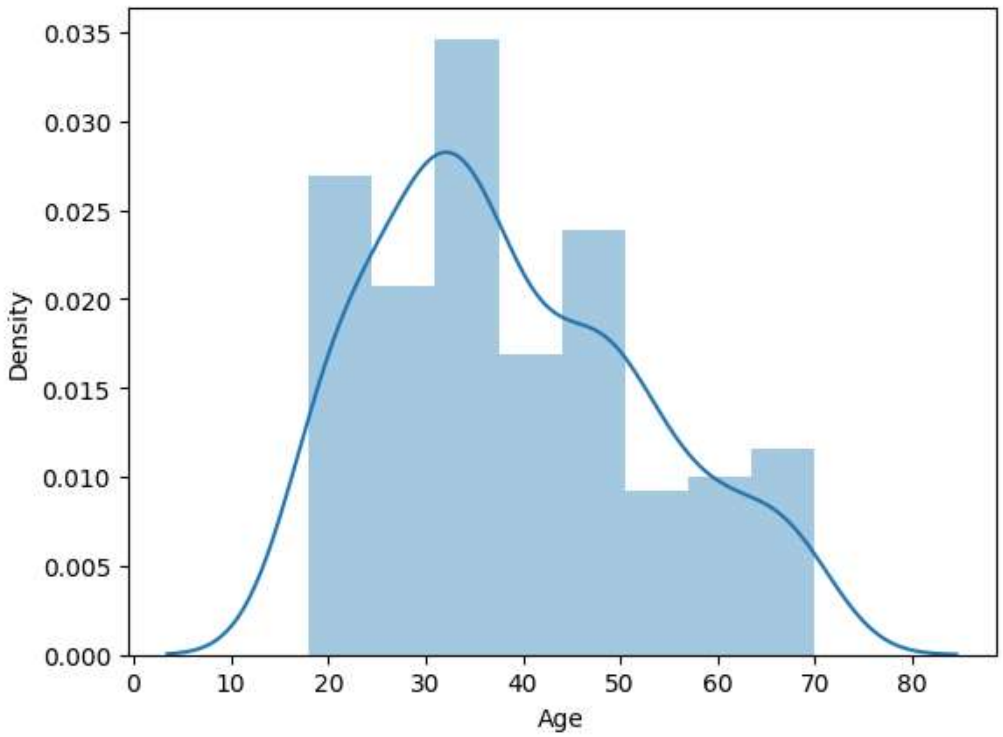
```
sns.distplot(df['Annual Income (k$)']);
```

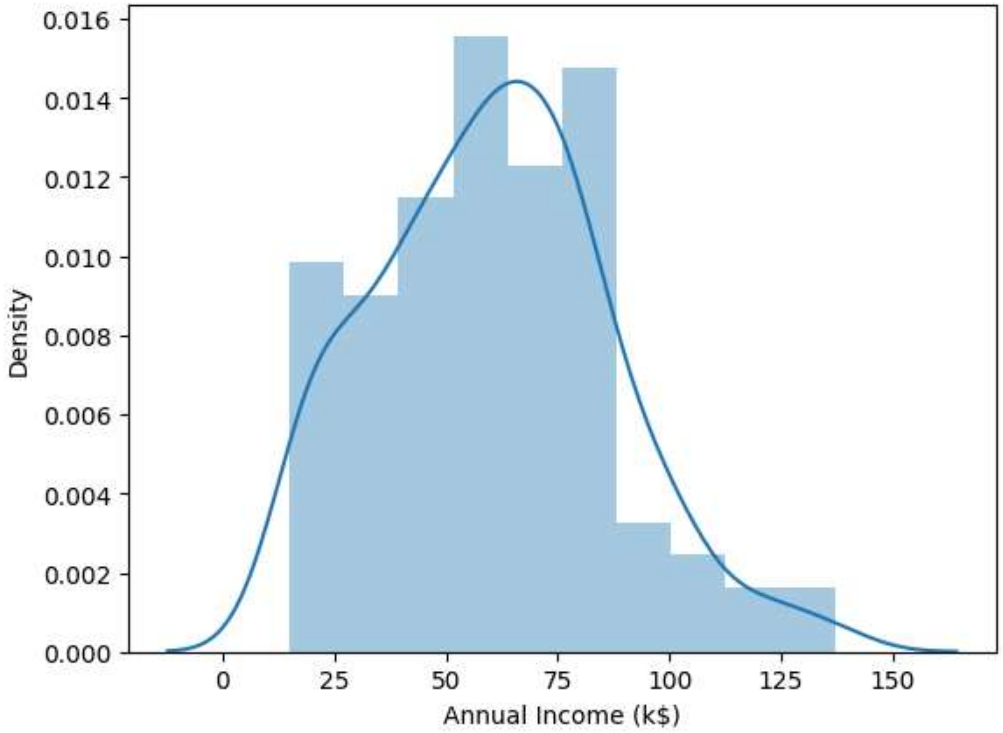


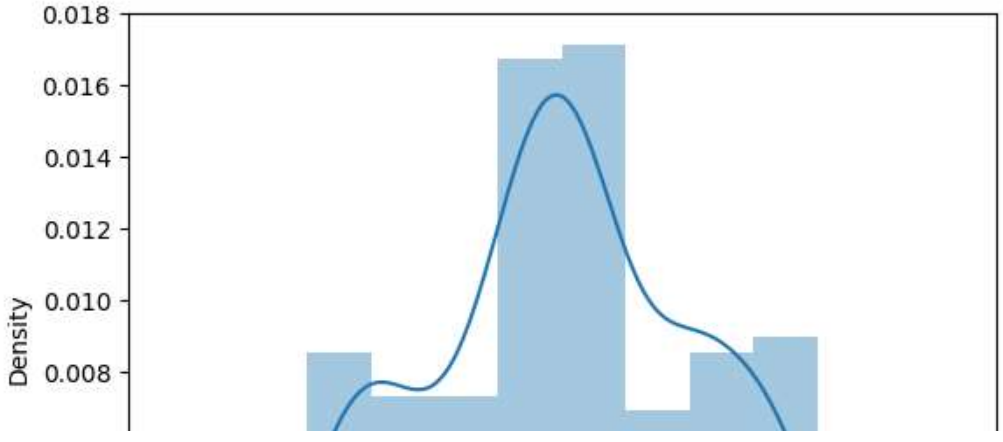

```
df.columns
```

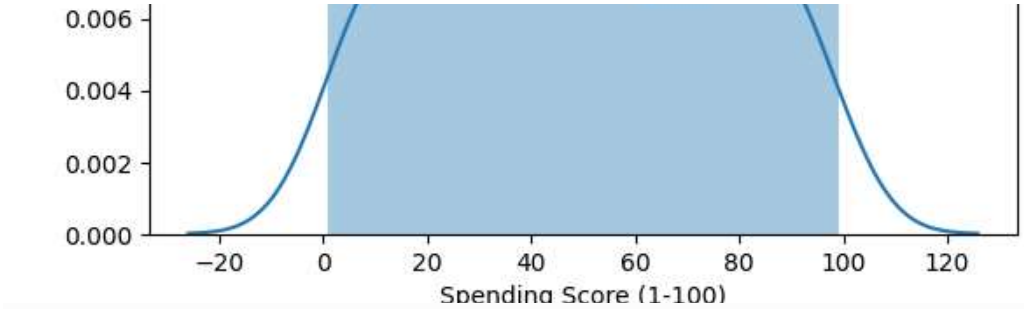
```
Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

```
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

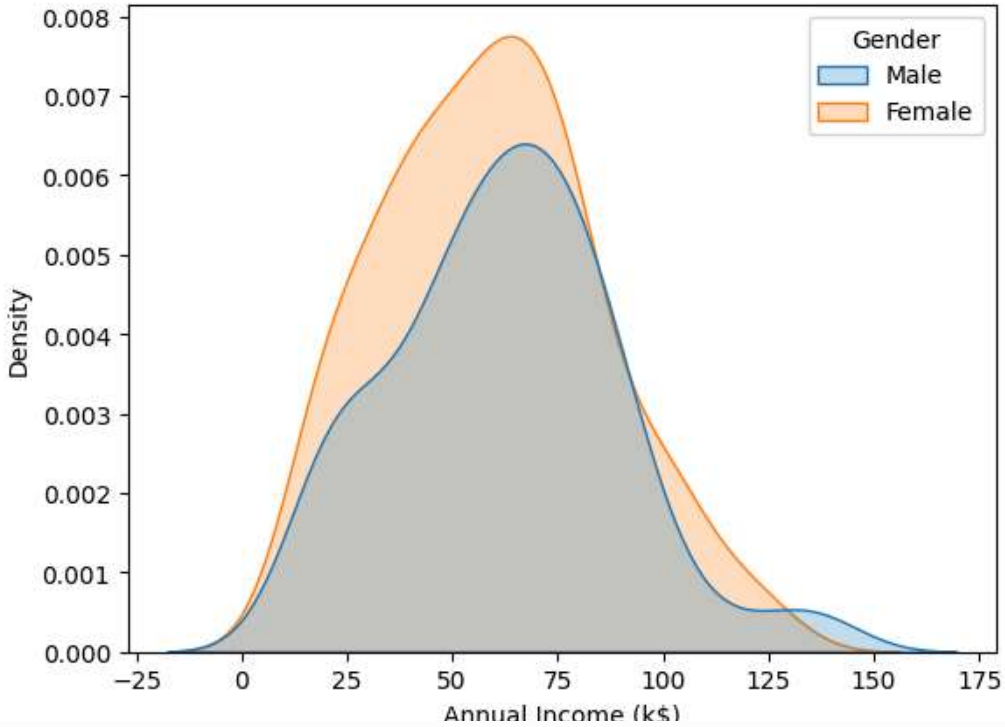

0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
Density
0
25
50
75
100
125
150
Annual Income (k$)


0.018
0.016
0.014
0.012
0.010
0.008
Density

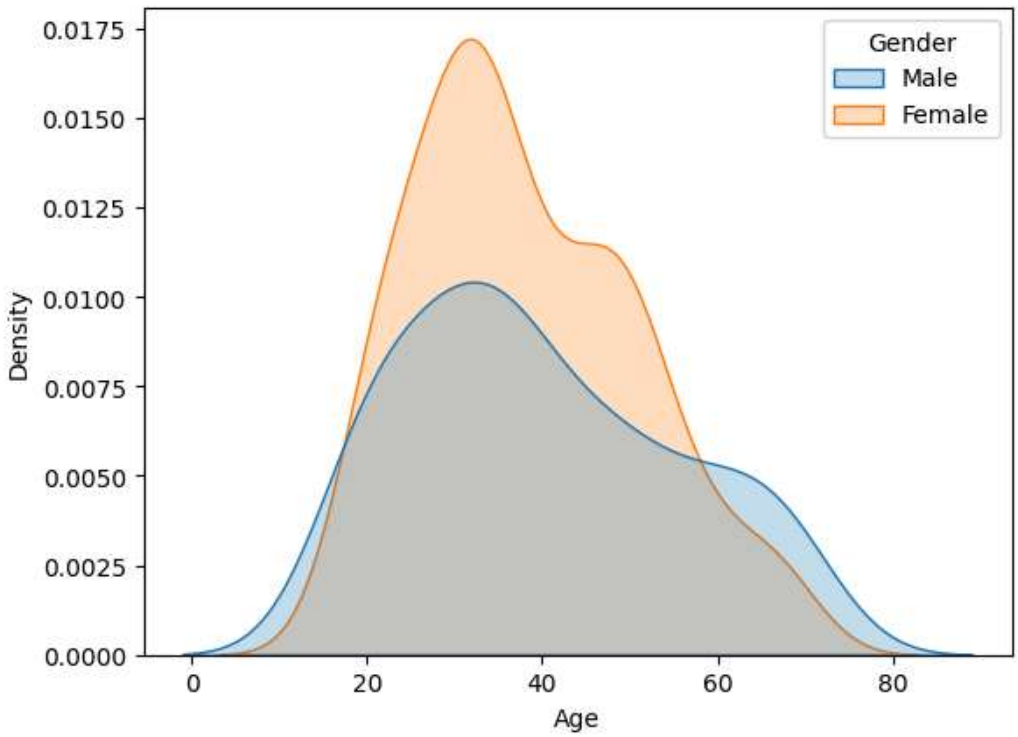


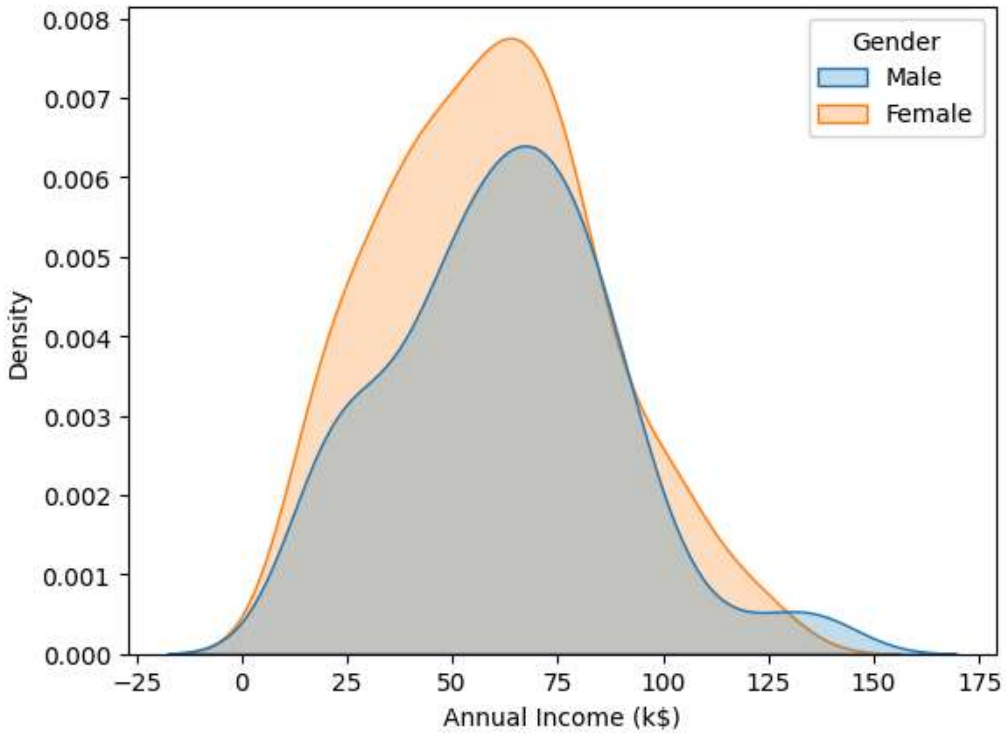

```
sns.kdeplot(data=df, x='Annual Income (k$)', hue='Gender', shade=True)
```

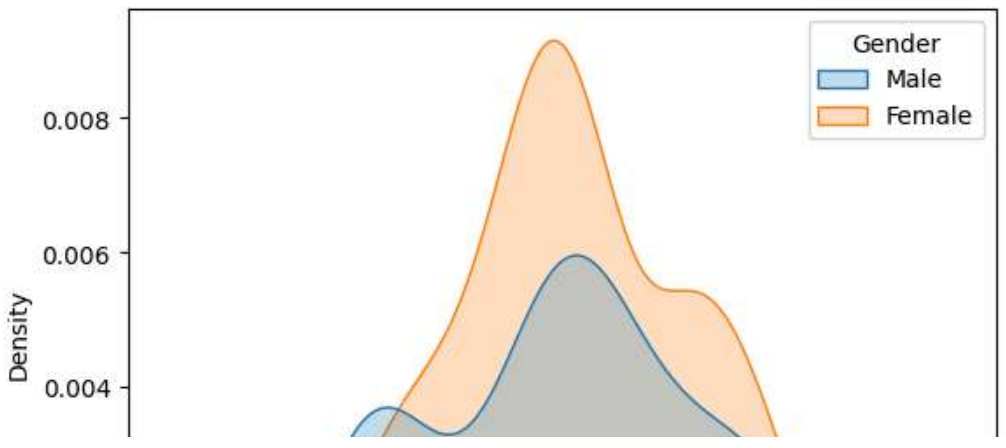
```
<Axes: xlabel='Annual Income (k$)', ylabel='Density'>
```

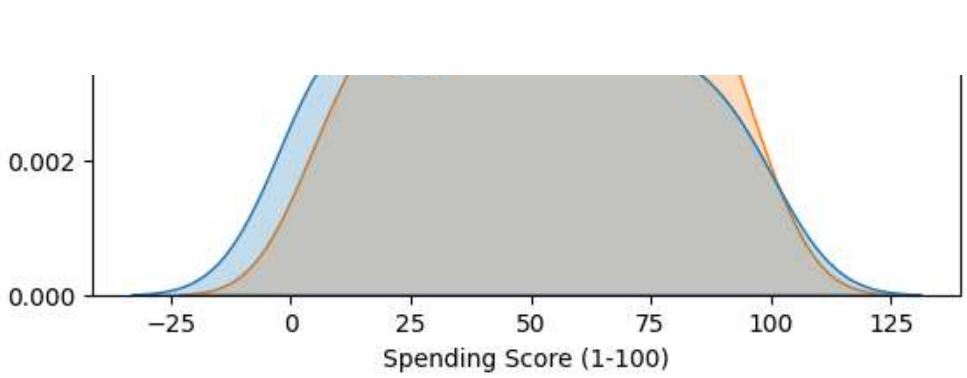



```
columns = ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']
for i in columns:
  plt.figure()
  sns.kdeplot(data=df, x=i, hue='Gender', shade=True)
```

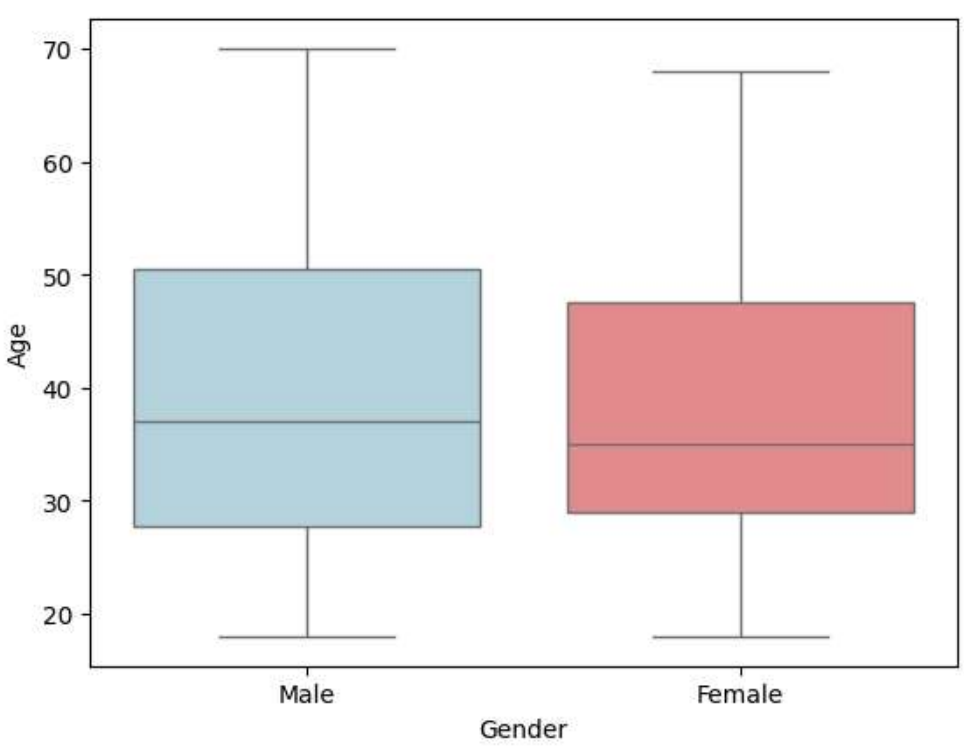

Gender
Male
Female
Density
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
20
40
60
80
Age

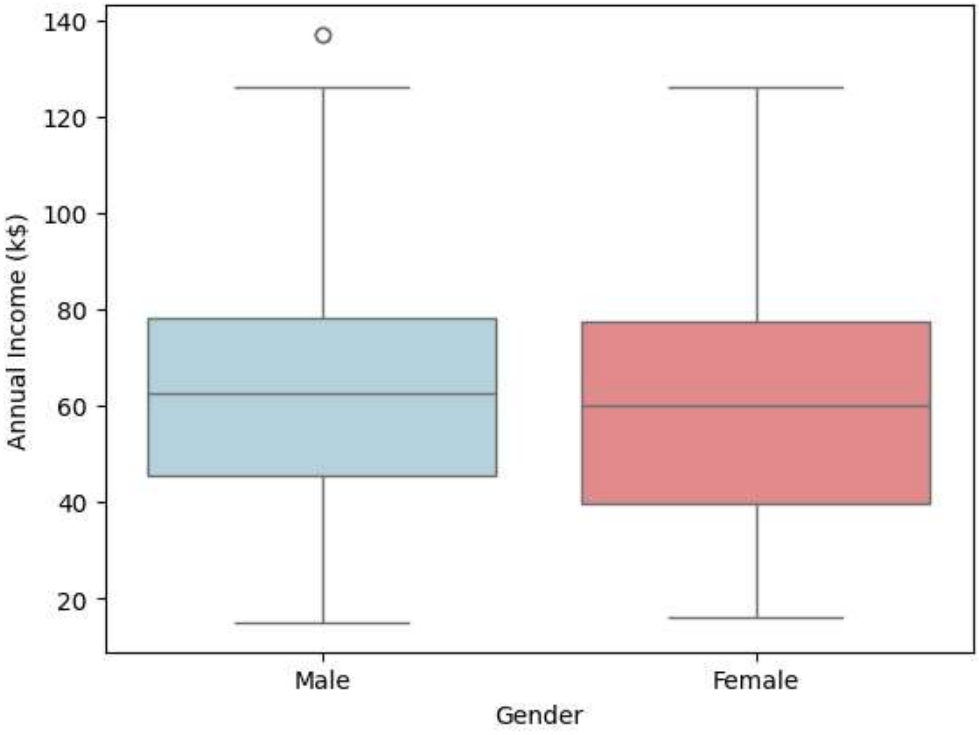

Gender
Male
Female
Density
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
−25
0
25
50
75
100
125
150
175
Annual Income (k$)

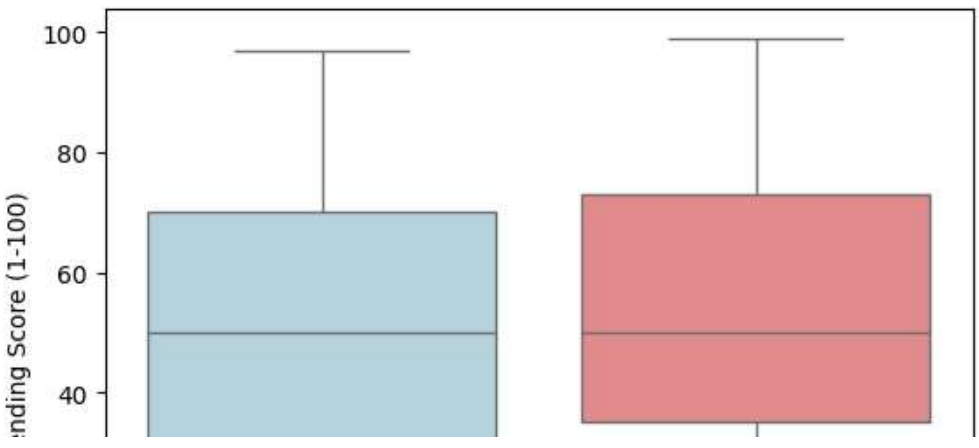

Gender
Male
Female
Density
0.008
0.006
0.004

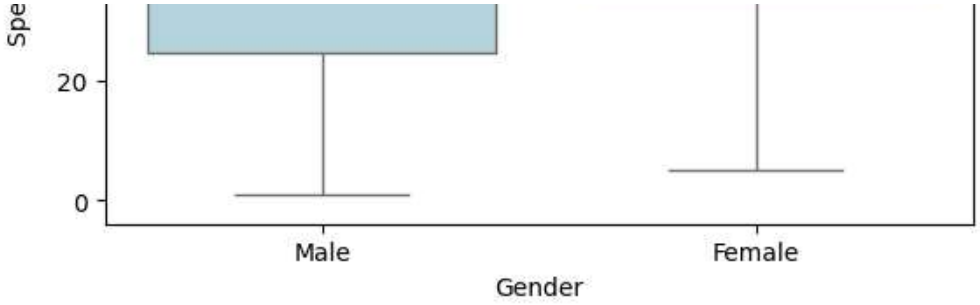



```
columns = ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']
for i in columns:
    plt.figure()
    sns.boxplot(data=df, x='Gender', y=df[i], palette=['lightblue', 'lightcoral'])
```


70
60
50
40
30
20
Age
Male
Female
Gender


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


100
80
60
40
nding Score (1-100)




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

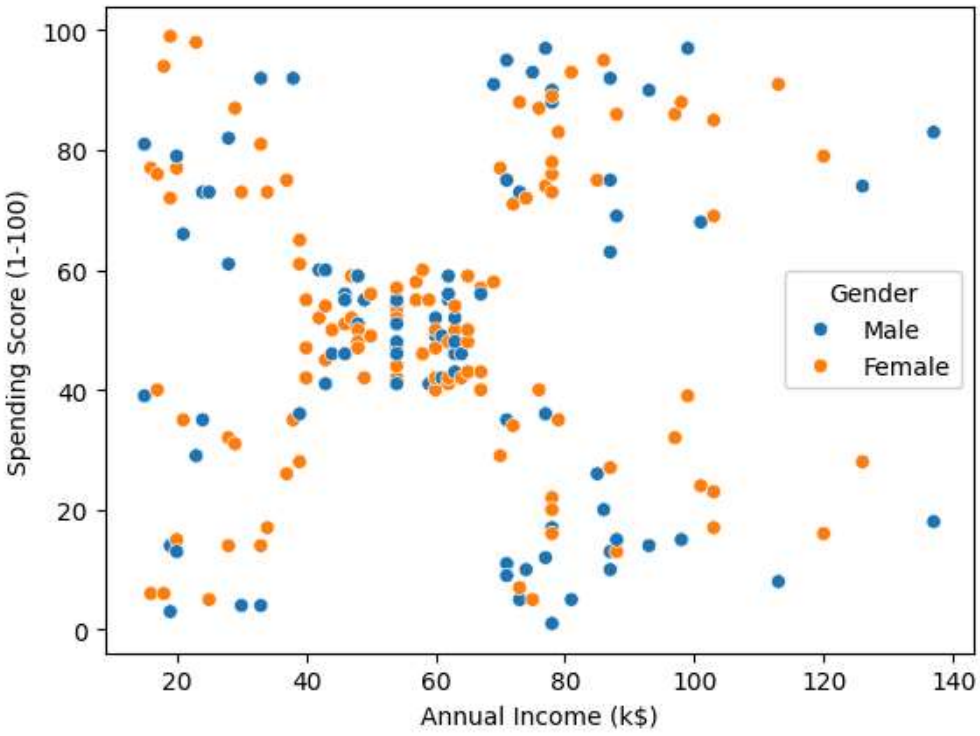
Gender	proportion
Female	0.56
Male	0.44

dtype: float64

Bivariate Analysis

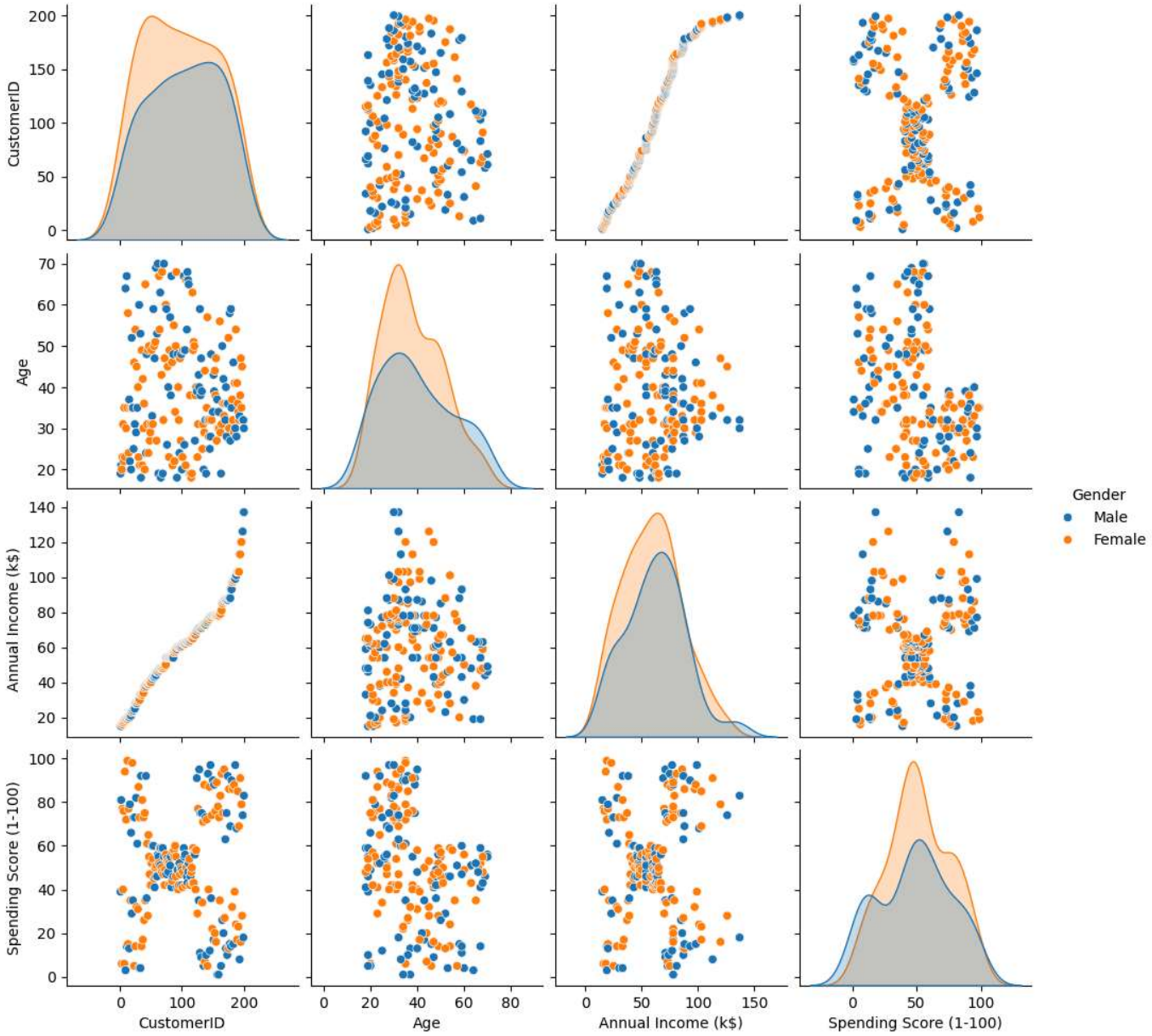
```
sns.scatterplot(data=df, x='Annual Income (k$)', y='Spending Score (1-100)', hue='Gender')
```

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




```
#df=df.drop('CustomerID',axis=1)
sns.pairplot(df,hue='Gender')
```

<seaborn.axisgrid.PairGrid at 0x788f40ebe830>

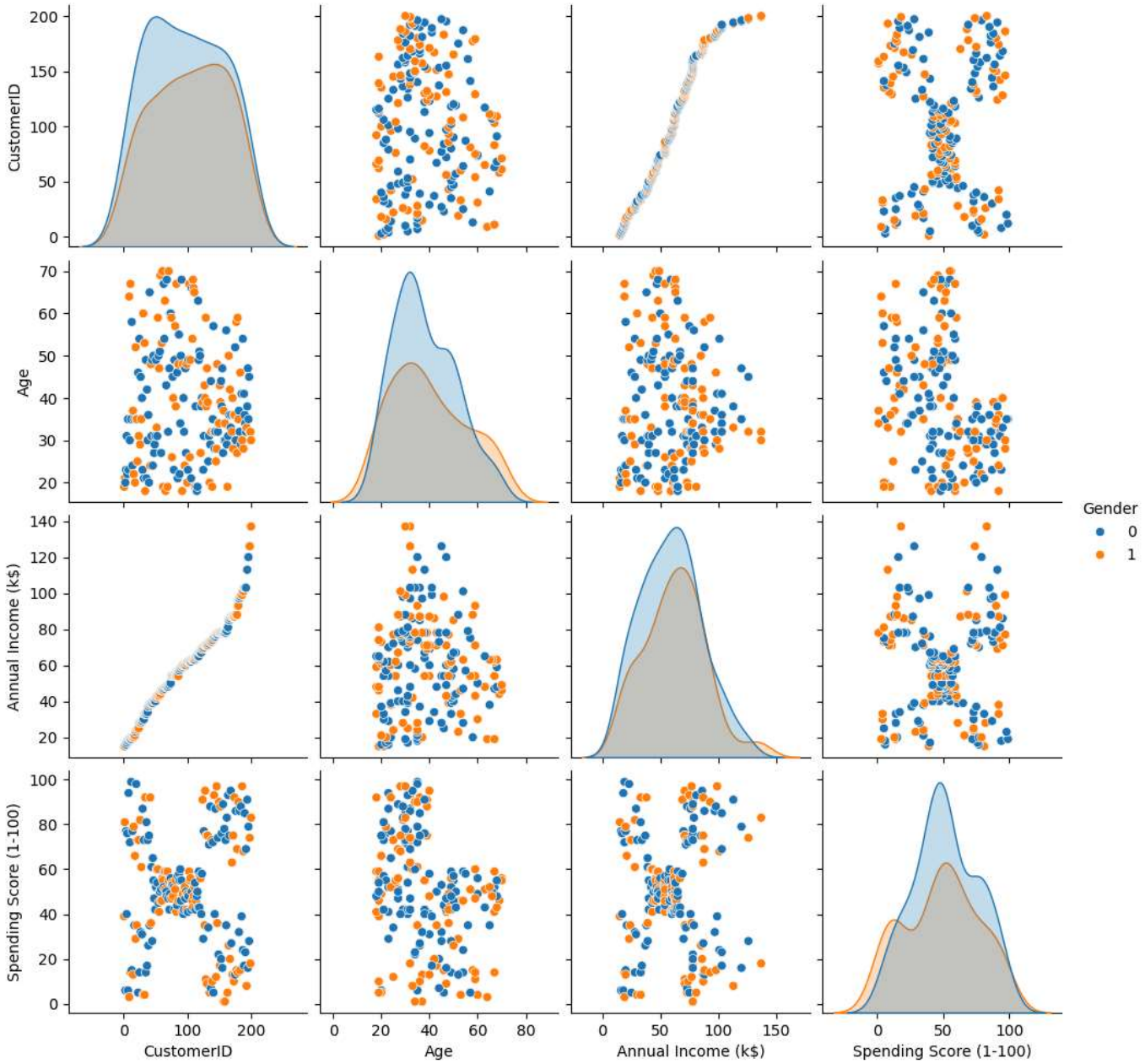



```
df.groupby(['Gender'])[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']].mean()
```

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

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Gender'] = le.fit_transform(df['Gender'])
sns.pairplot(df, hue='Gender')
```

<seaborn.axisgrid.PairGrid at 0x788f3e3658d0>

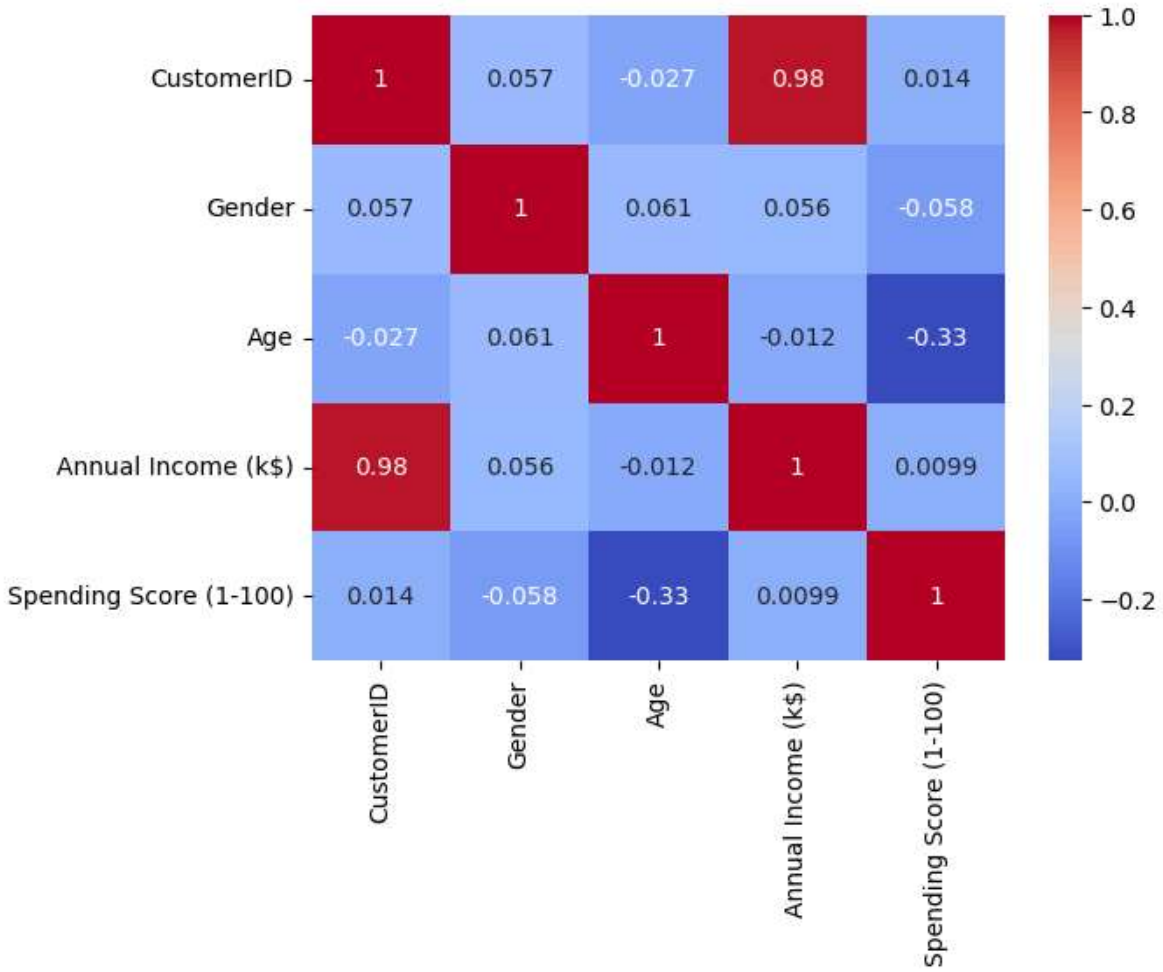



```
df.corr()
```

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
CustomerID	1.000000	0.057400	-0.026763	0.977548	0.013835
Gender	0.057400	1.000000	0.060867	0.056410	-0.058109
Age	-0.026763	0.060867	1.000000	-0.012398	-0.327227
Annual Income (k$)	0.977548	0.056410	-0.012398	1.000000	0.009903
Spending Score (1-	0.013835	0.058109	0.327227	0.009903	1.000000

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

```
<Axes: >
```



Start coding or generate with AI.

Clustering - Univariate, Bivariate, Multivariate

```
clustering1 = KMeans(n_clusters=3)
```

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

```
KMeans
KMeans(n_clusters=3)
```

```
clustering1.labels_
```

```
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, 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, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

```
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1], dtype=int32)
```

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

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

Next steps: Generate code with df | View recommended plots | New interactive sheet

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

Income Cluster	count
0	86
2	58
1	56

dtype: int64

```
clustering1.inertia_
```

```
25640.457784396807
```

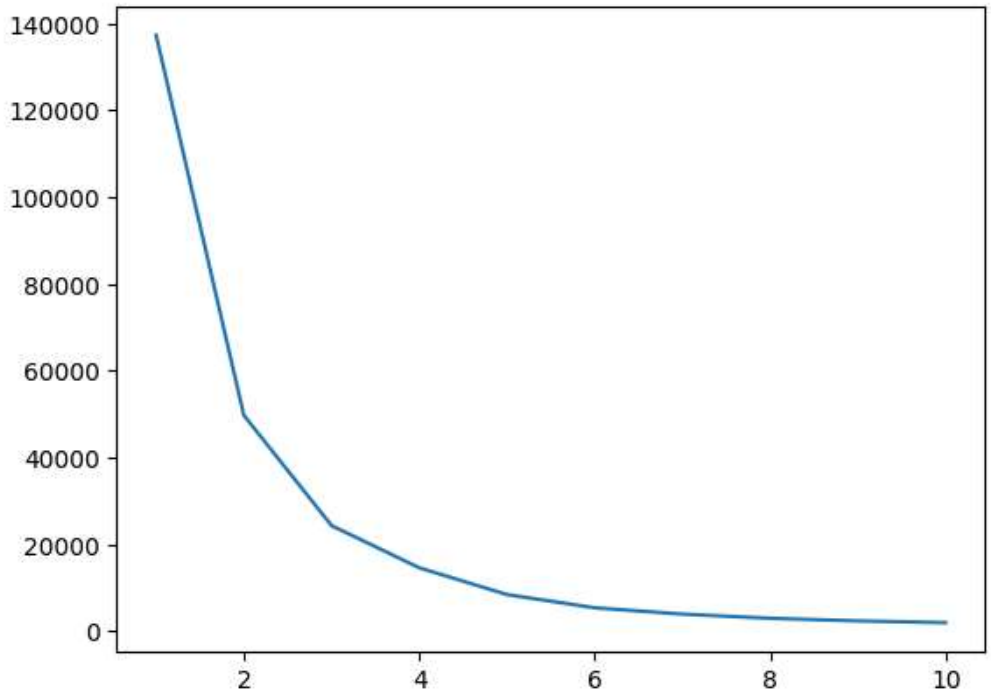
```
intertia_scores=[]
for i in range(1,11):
    kmeans=KMeans(n_clusters=i)
    kmeans.fit(df[['Annual Income (k$)']])
    intertia_scores.append(kmeans.inertia_)
```

```
intertia_scores
```

```
[137277.2800000002,
 49761.737012987025,
 24361.259213759215,
 14647.235170393054,
 8481.49619047619,
 5430.245925925928,
 3989.608946608945,
 3021.5955988456008,
 2426.67332112332,
 2013.230710466005]
```

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

```
[<matplotlib.lines.Line2D at 0x788f3d916c80>]
```




```
df.columns
```

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

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

Income Cluster	Age	Annual Income (k$)	Spending Score (1-100)
0	41.279070	60.906977	50.337209
1	36.910714	92.142857	50.517857
2	37.120690	29.551724	49.689655

Bivariate Clustering

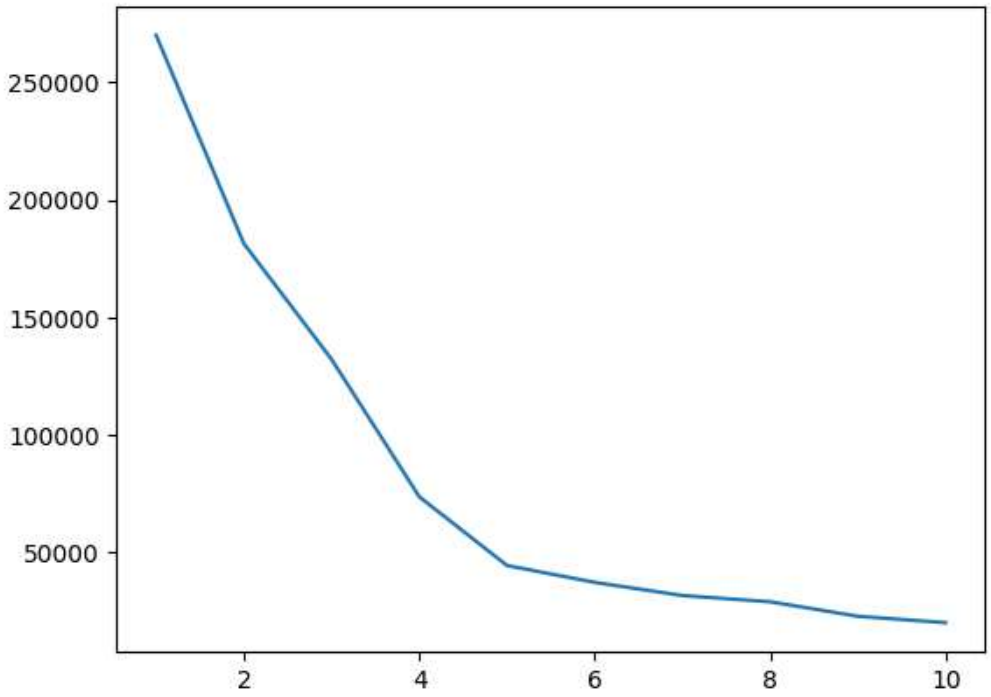
```
clustering2 = KMeans(n_clusters=5)
clustering2.fit(df[['Annual Income (k$)','Spending Score (1-100)']])
df['Spending and Income Cluster'] =clustering2.labels_
df.head()
```

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

Next steps: Generate code with df | View recommended plots | New interactive sheet

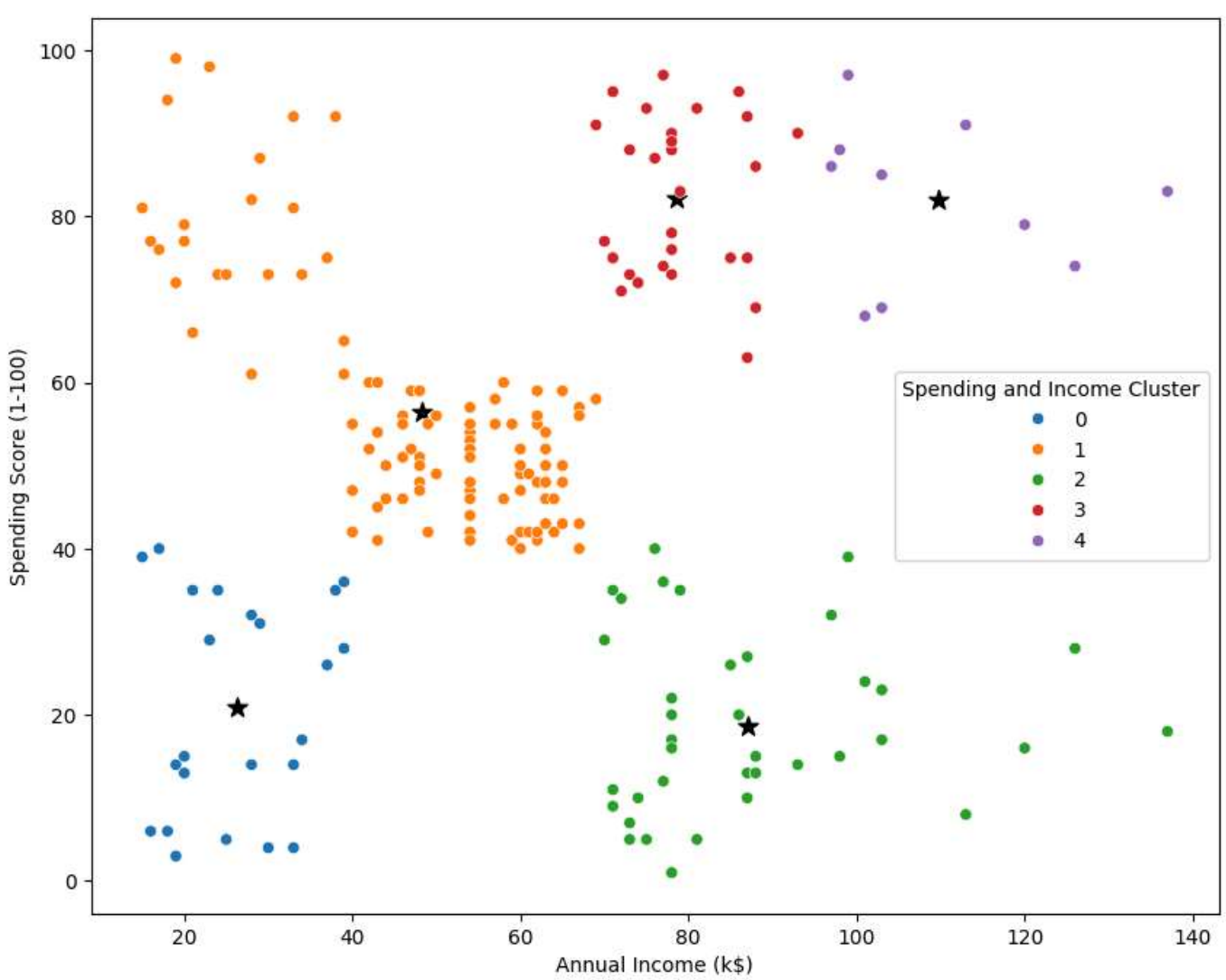
```
intertia_scores2=[]
for i in range(1,11):
    kmeans2=KMeans(n_clusters=i)
    kmeans2.fit(df[['Annual Income (k$)','Spending Score (1-100)']])
    intertia_scores2.append(kmeans2.inertia_)
plt.plot(range(1,11),intertia_scores2)
```

```
[<matplotlib.lines.Line2D at 0x788f3d7fd810>]
```




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


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 Clust
plt.savefig('clustering_bivaraiate.png')
```




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

Gender	0	1
Spending and Income Cluster		
0	0.608696	0.391304
1	0.590000	0.410000
2	0.473684	0.526316
3	0.517241	0.482759
4	0.600000	0.400000

Generate

ValueError: Cannot subset columns with a tuple with more than one element. Use a list instead.
df.groupby('Spending and Income Cluster')['Age', 'Annual Income (k$)',

Close

1 of 1

Undo Changes

Use code with caution

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

Spending and Income Cluster	Age	Annual Income (k$)	Spending Score (1-100)
0	45.217391	26.304348	20.913043
1	39.200000	48.260000	56.480000
2	40.394737	87.000000	18.631579
3	32.862069	78.551724	82.172414
4	32.200000	109.700000	82.000000

Mulivariate clustering

```
from sklearn.preprocessing import StandardScaler
scale = StandardScaler()
df.head()
```

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

Next steps: Generate code with df | View recommended plots | New interactive sheet

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

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

Next steps: Generate code with dff | View recommended plots | New interactive sheet

```
dff.columns
```

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

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

```
dff.head()
```

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

Next steps: Generate code with dff | View recommended plots | New interactive sheet

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