```
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("C:/Users/Dell i5/OneDrive - Cape Peninsula University 
of Technology/Desktop/Portfolio projects/Customer 
segmentation/Mall_Customers.csv")
```

Univariate analysis

```
df.head()
```

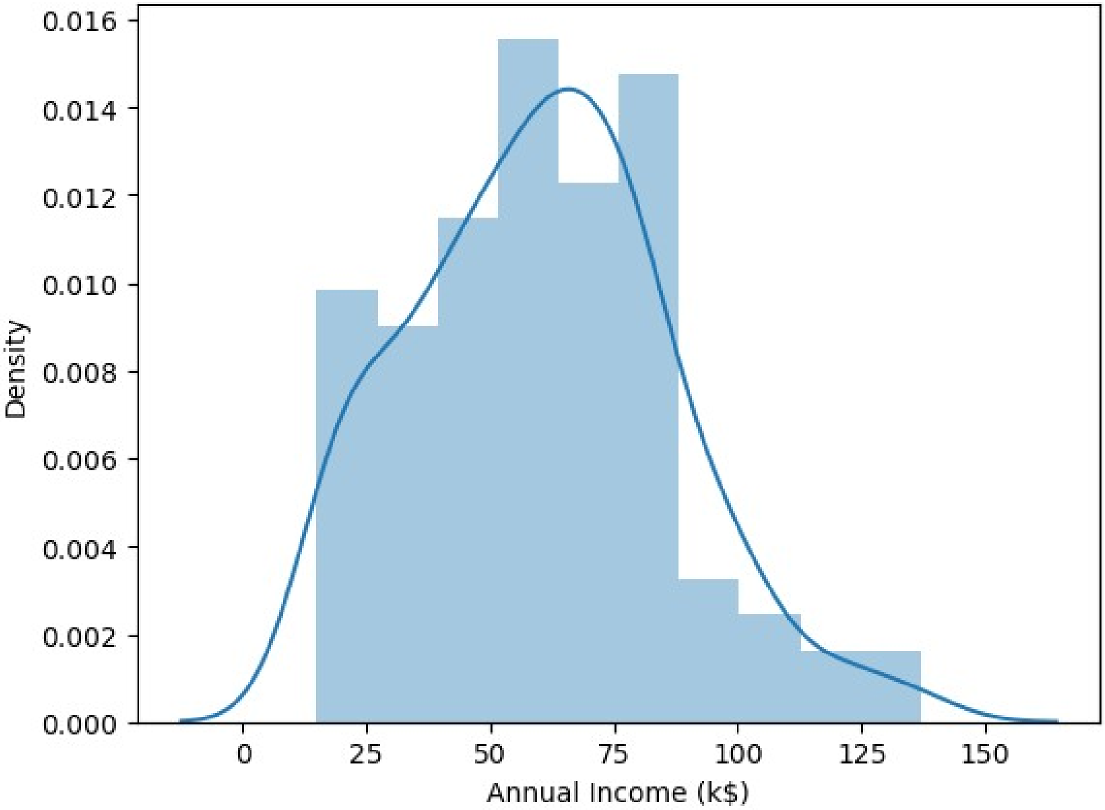
```
   CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4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

```
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$)"])
```

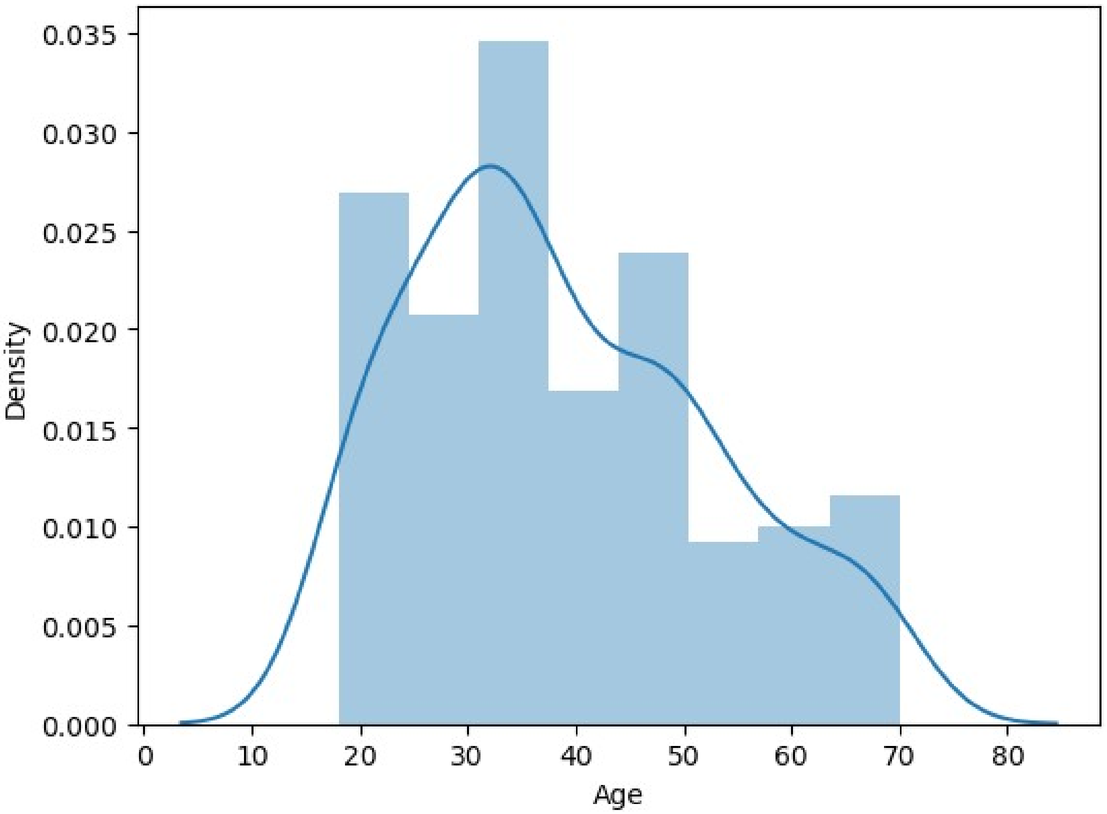
```
<AxesSubplot:xlabel='Annual Income (k$)', ylabel='Density'>
```

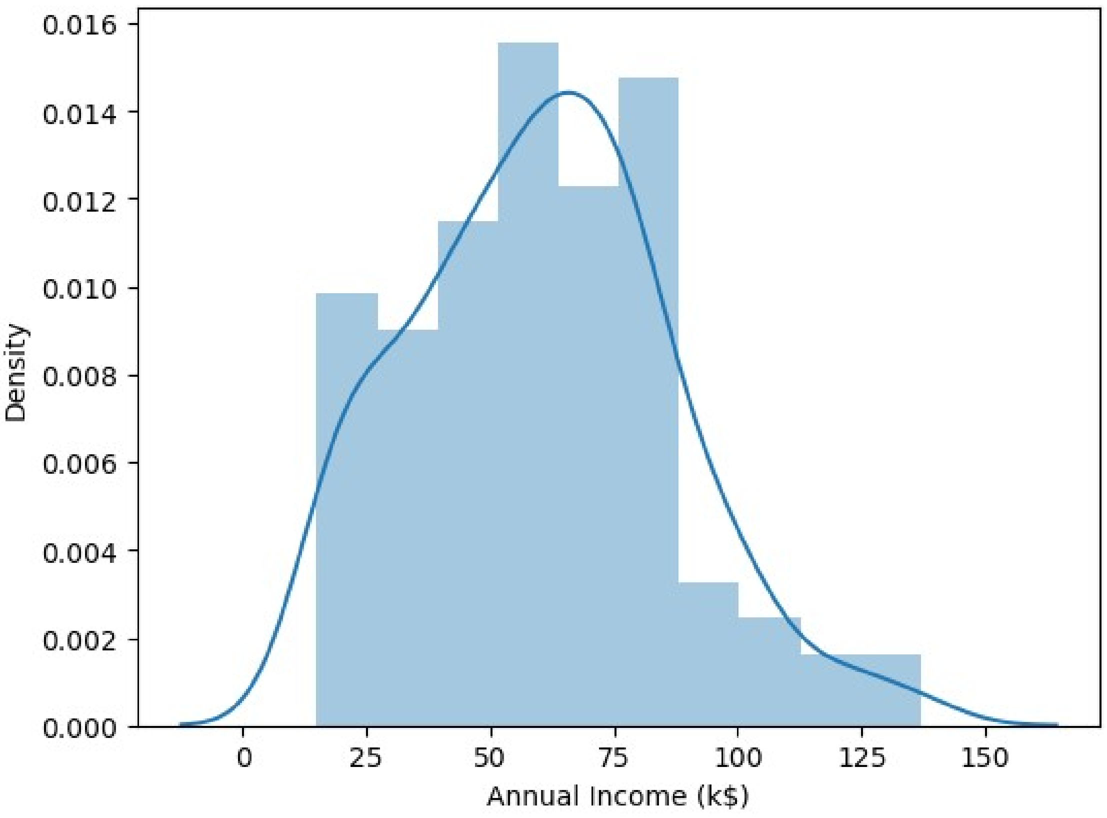


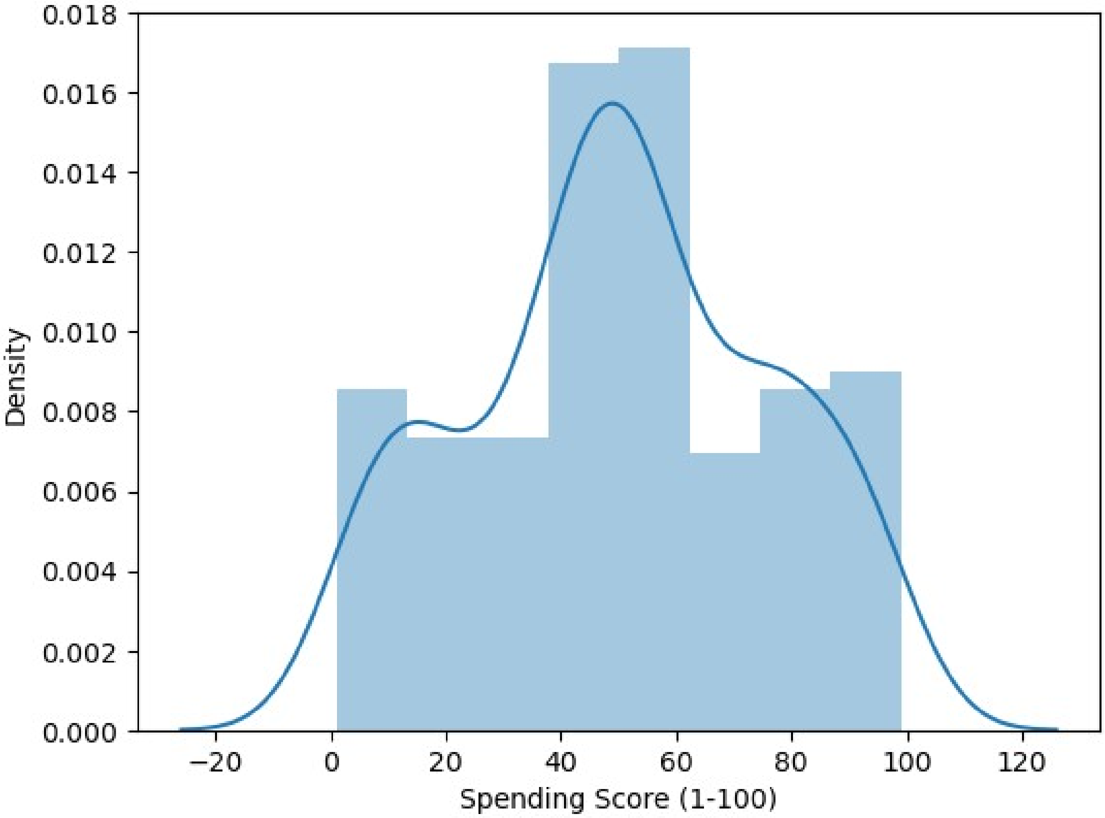
```
df.columns
```

```
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])
```

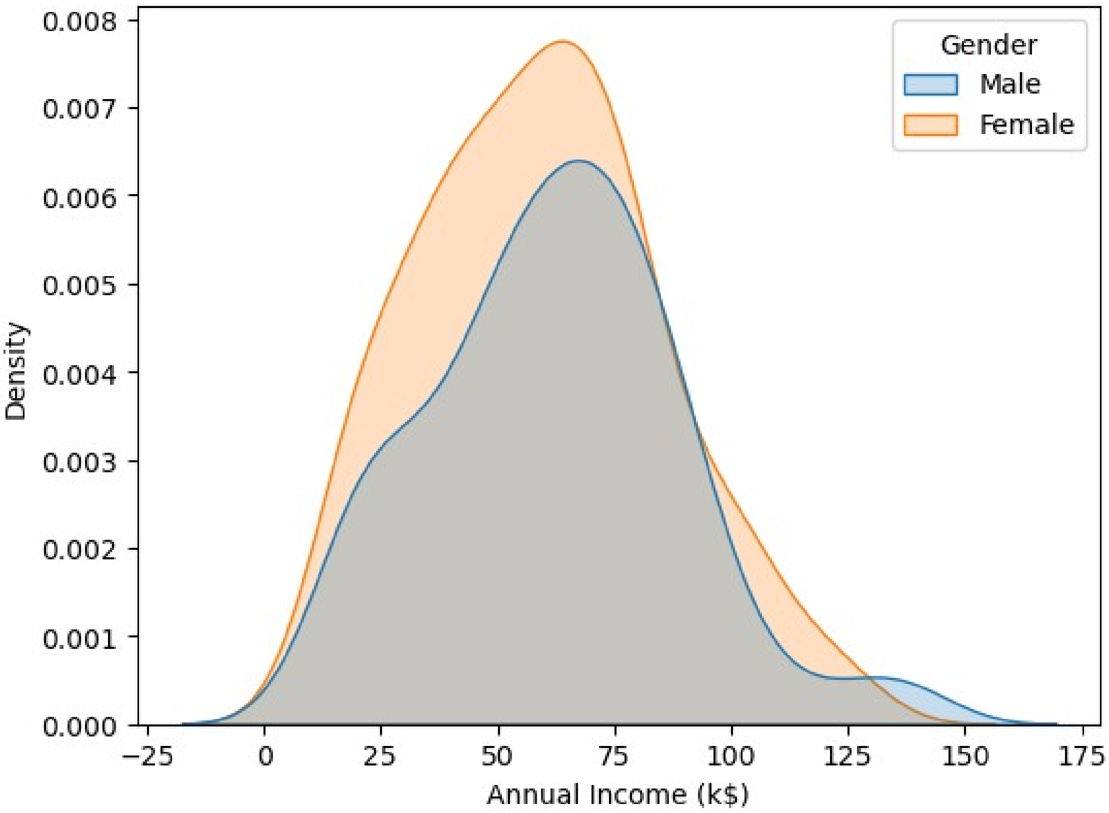

Density
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0
10
20
30
40
50
60
70
80
Age


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

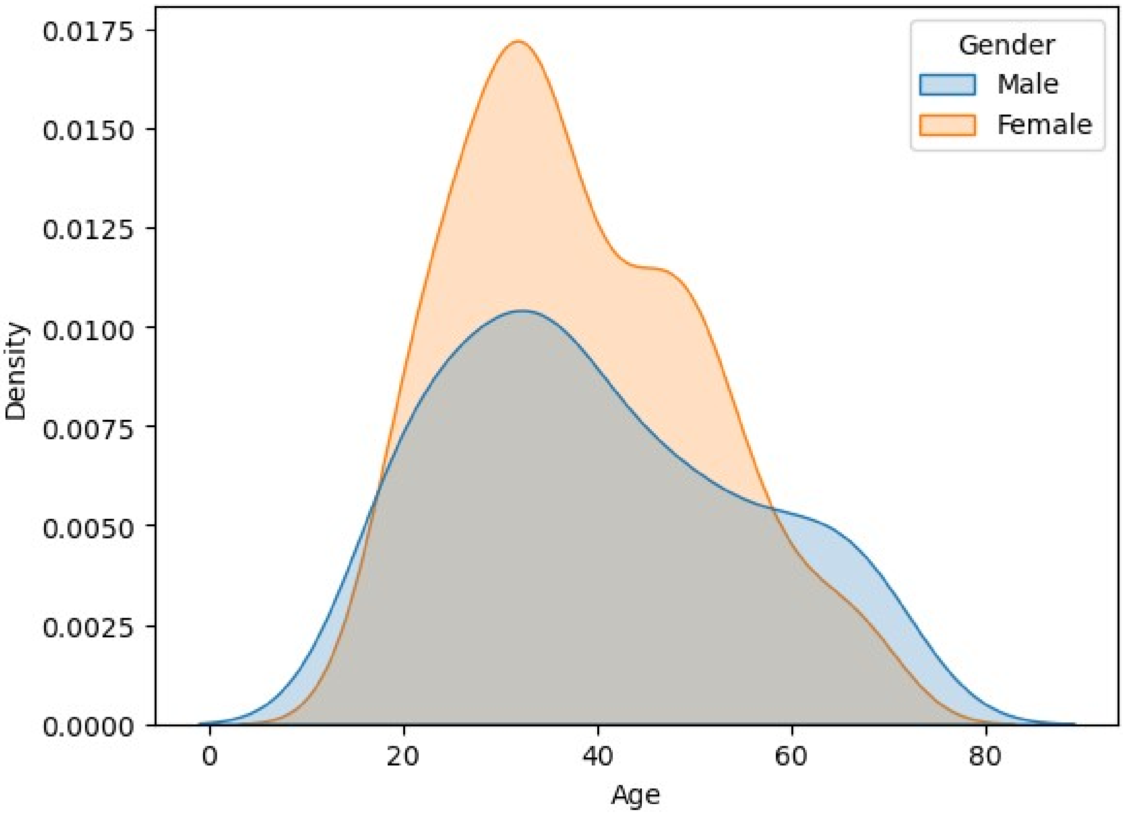


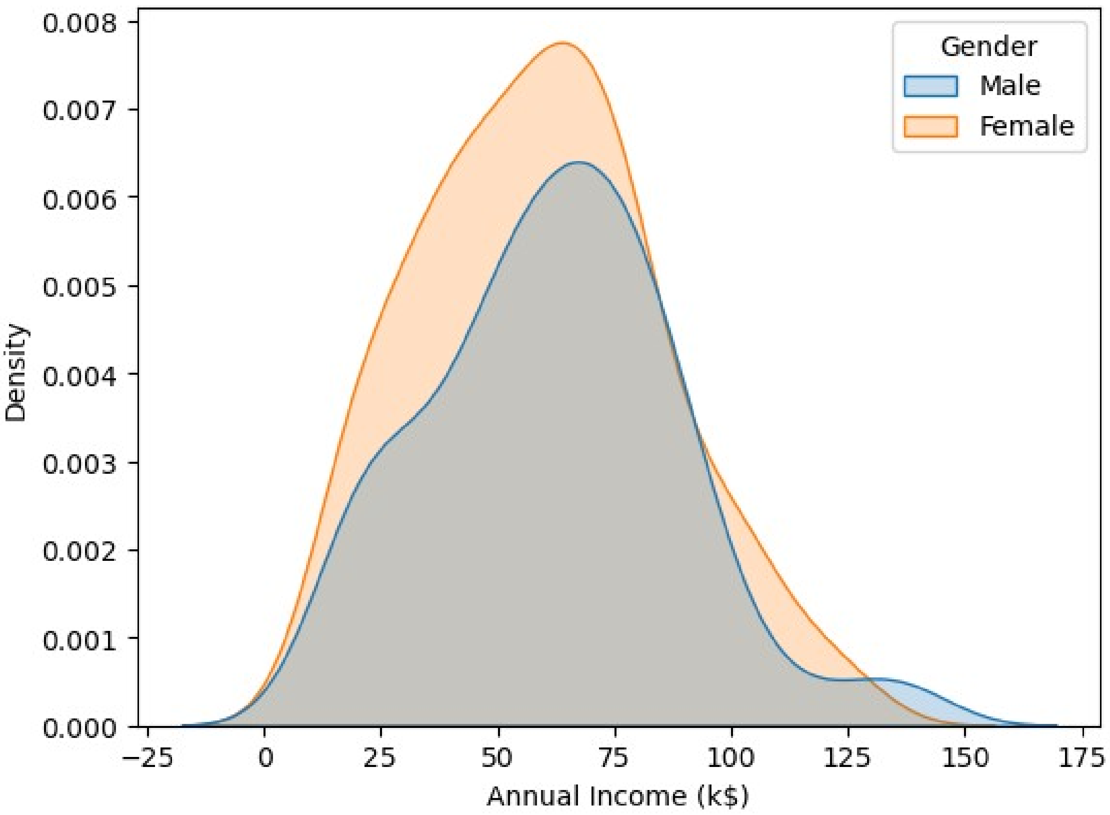
```
sns.kdeplot(df['Annual Income (k$)'],shade=True,hue=df['Gender'])
```

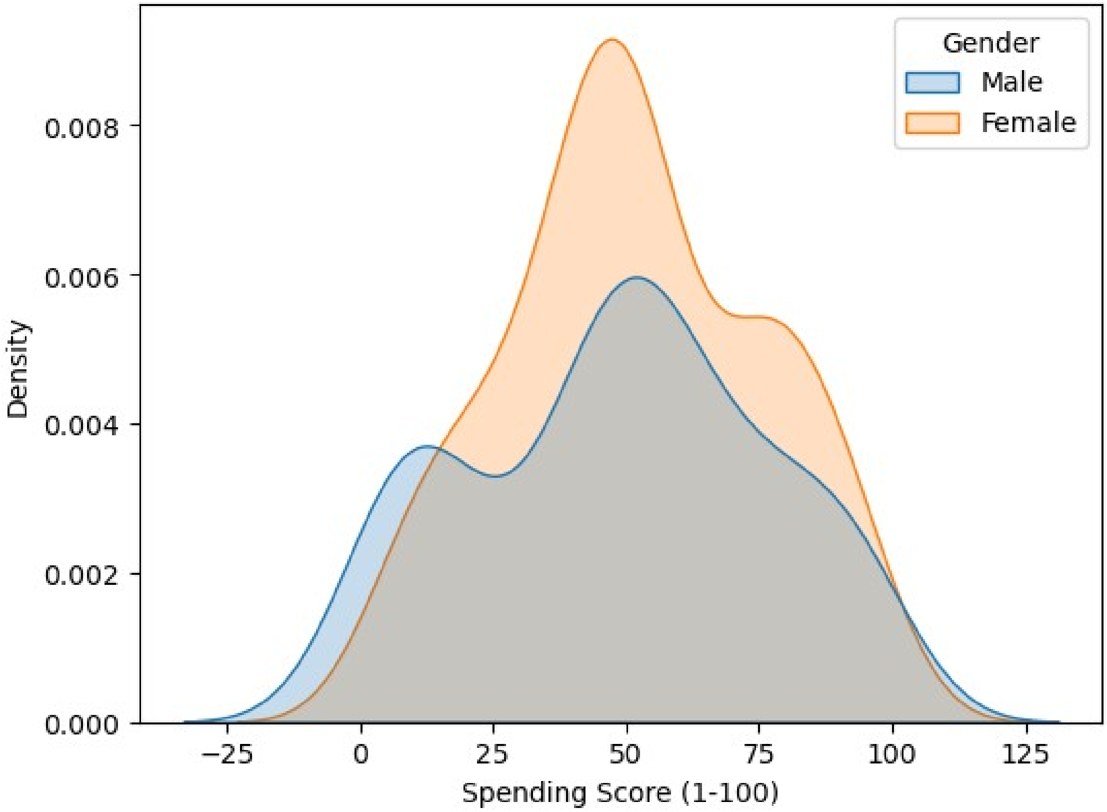
```
<AxesSubplot:xlabel='Annual Income (k$)', ylabel='Density'>
```



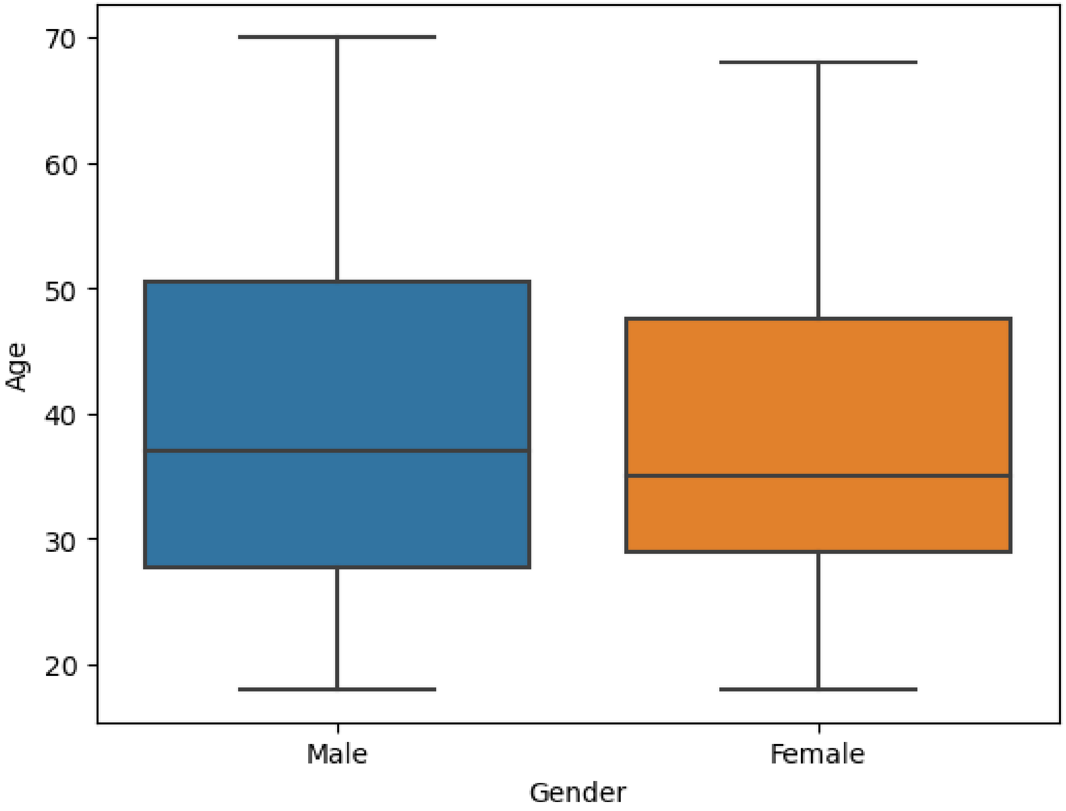
```
columns = ['Age', 'Annual Income (k$)','Spending Score (1-100)']
for i in columns:
    plt.figure()
    sns.kdeplot(df[i],shade=True,hue=df['Gender'])
```

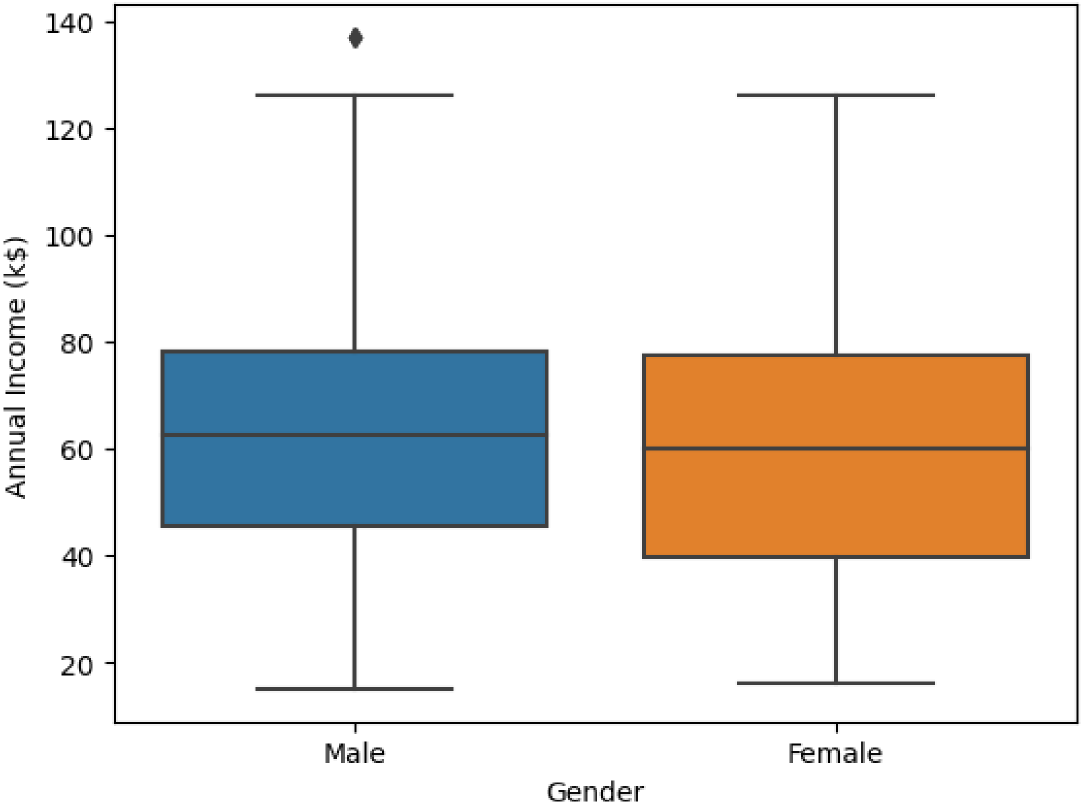

Gender
Male
Female
Density
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0
20
40
60
80
Age

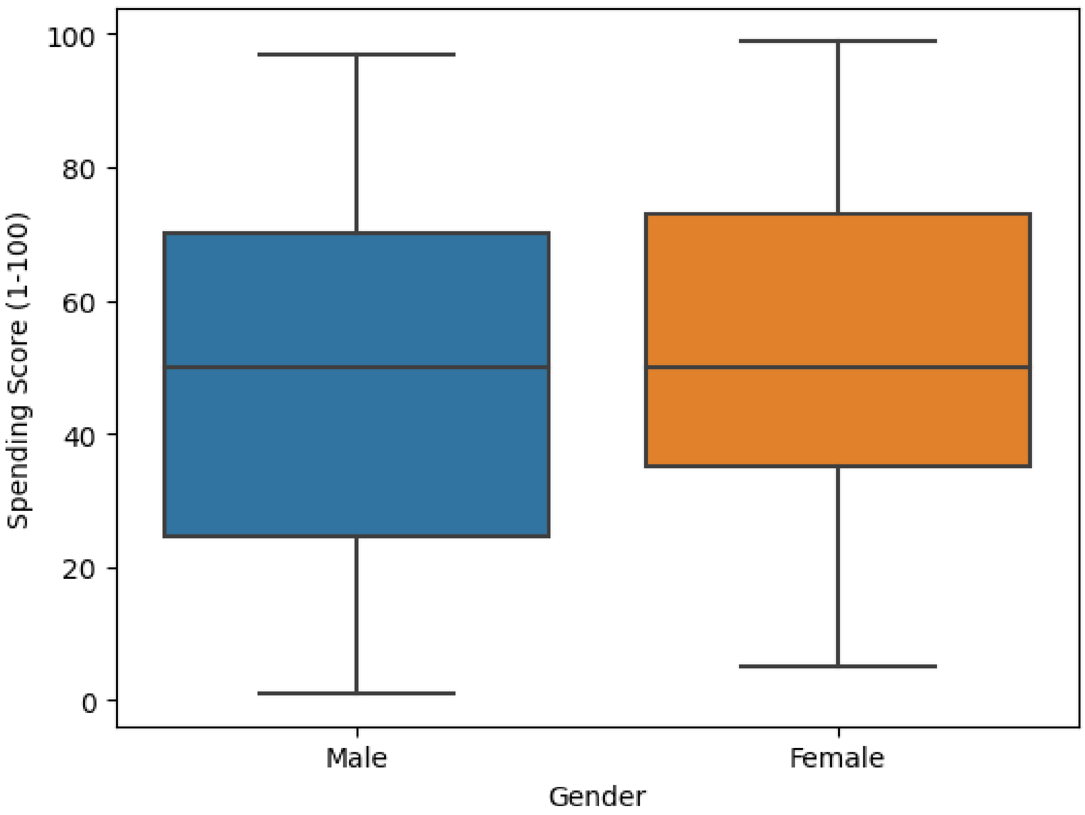

Gender
Male
Female
Density
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
−25
0
25
50
75
100
125
150
175
Annual Income (k$)



```
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



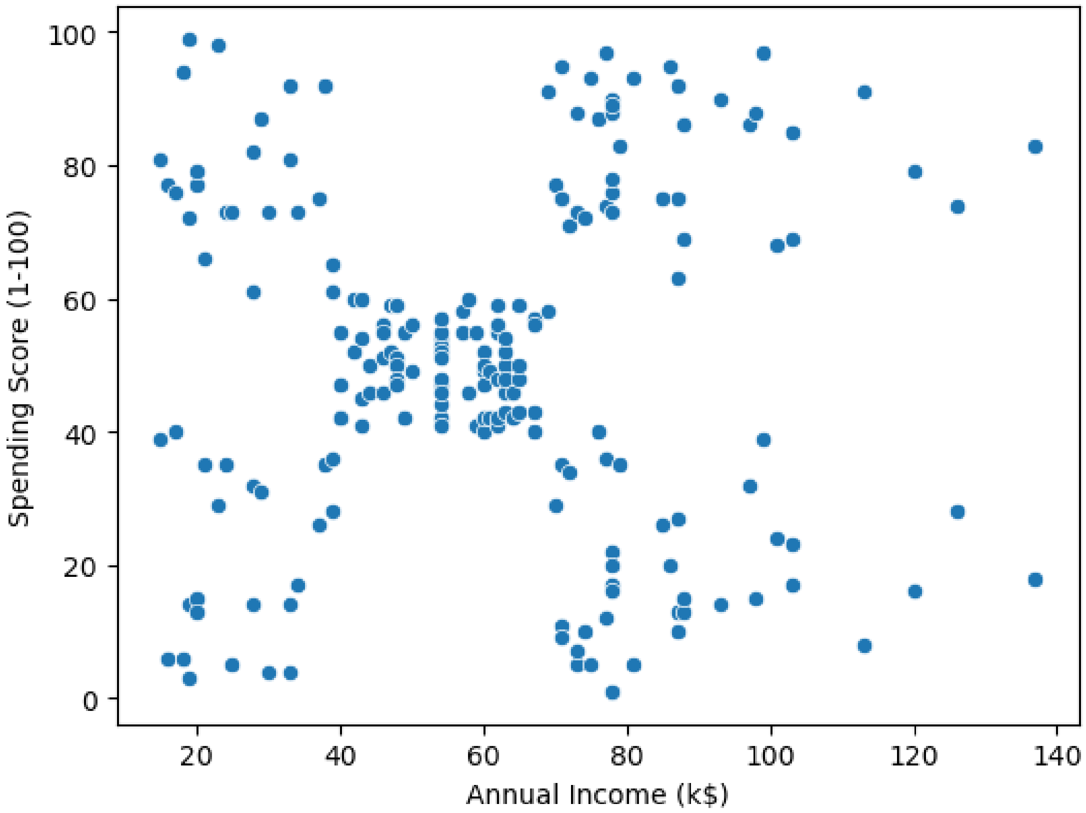
```
df['Gender'].value_counts(normalize=True)
```

```
Female    0.56
Male      0.44
Name: Gender, dtype: float64
```

Bivariate analysis

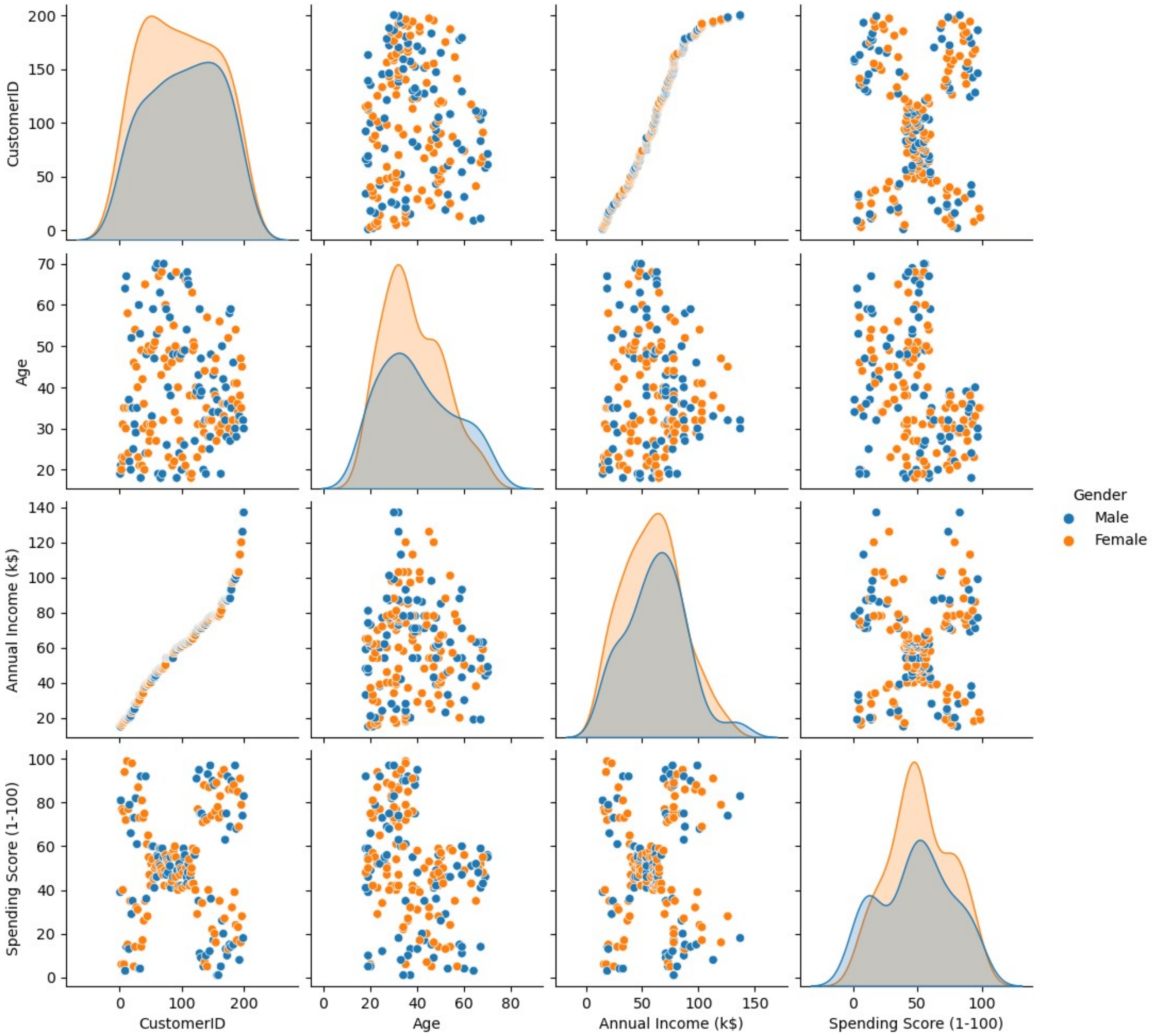
```
sns.scatterplot(data=df,x='Annual Income (k$)',y='Spending Score (1-
100)')
```

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



```
sns.pairplot(df,hue='Gender')
```

```
<seaborn.axisgrid.PairGrid at 0x156a7c34400>
```



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

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

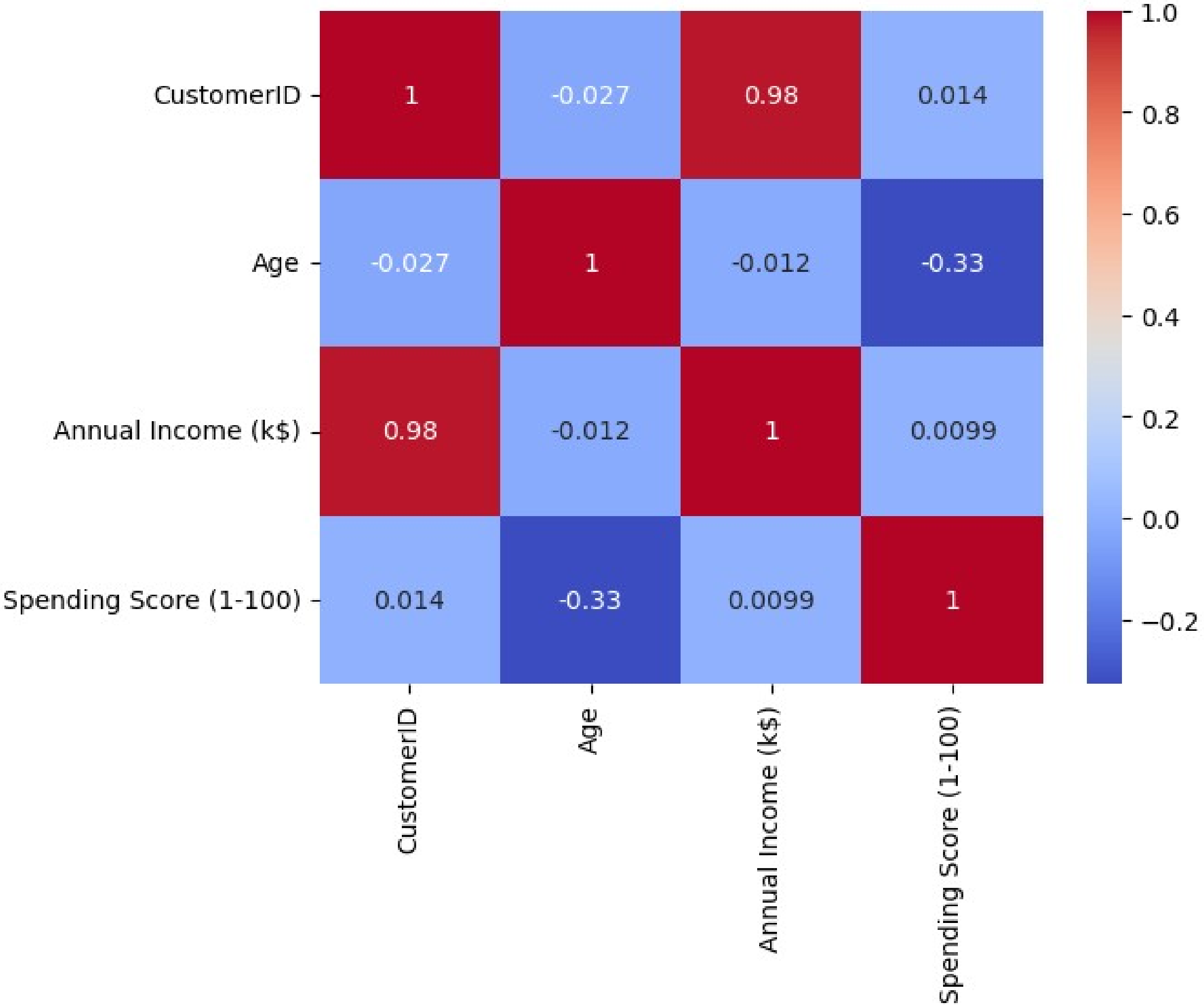
```
df.corr()
```

```
                        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
```

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

```
<AxesSubplot:>
```



Clustering

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

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

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

```
clustering1.labels_
```

```
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, 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,
       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])
```

```
df['Income Cluster']=clustering1.labels_
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


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

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

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

```
clustering1.inertia_
```

```
23517.330930930926
```

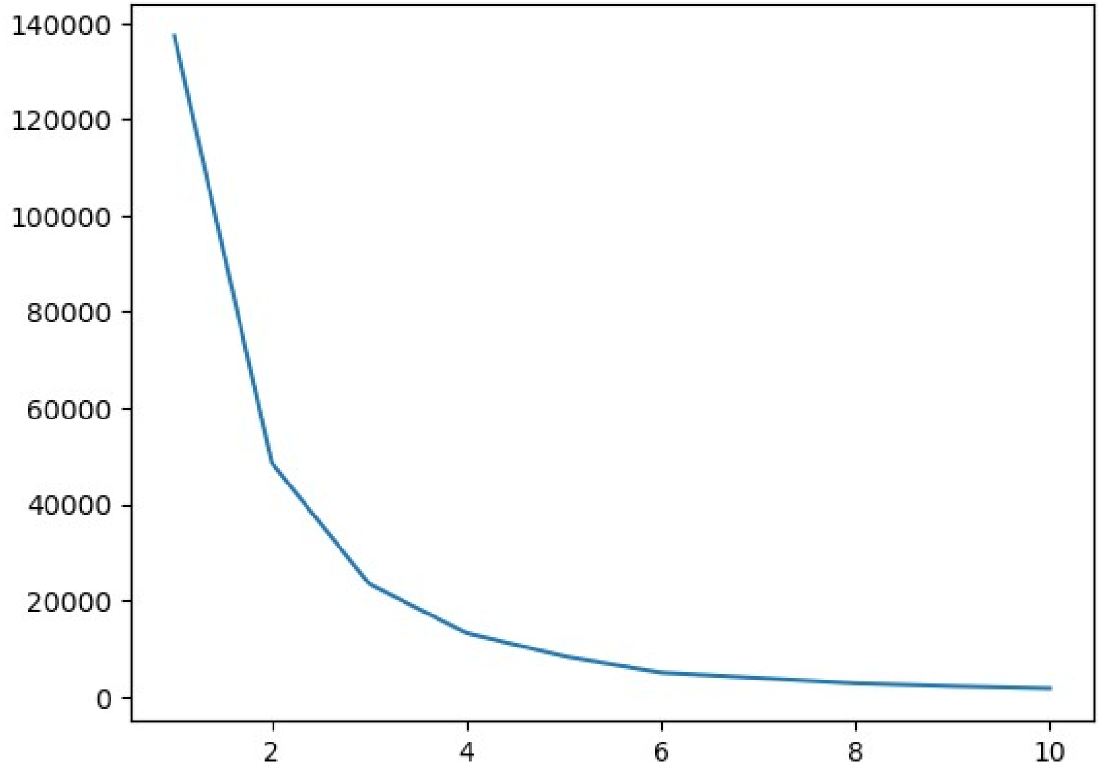
```
intertia_scores=[]
for i in range(1,11):
    kmeans=KMeans(n_clusters=i)
    kmeans.fit(df[['Annual Income (k$)']])
    intertia_scores.append(kmeans.inertia_)
```

```
intertia_scores
```

```
[137277.28000000003,
 48660.88888888889,
 23517.330930930926,
 13278.112713472487,
 8481.496190476191,
 5050.904761904763,
 3949.2756132756135,
 2822.4996947496943,
 2222.930303030303,
 1766.6142857142859]
```

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

```
[<matplotlib.lines.Line2D at 0x156a85a8be0>]
```



```
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()
```

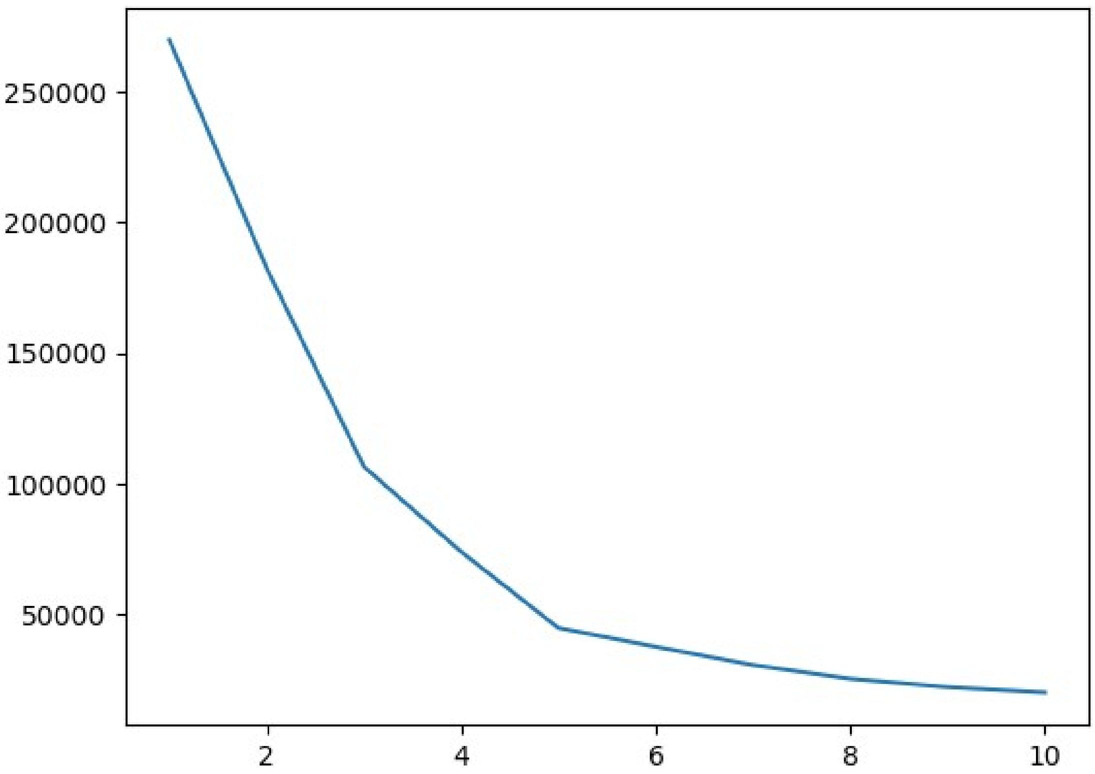
```
                      Age  Annual Income (k$)  Spending Score (1-100)
Income Cluster
0               39.500000           33.486486               50.229730
1               37.833333           99.888889               50.638889
2               38.722222           67.088889               50.000000
```

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

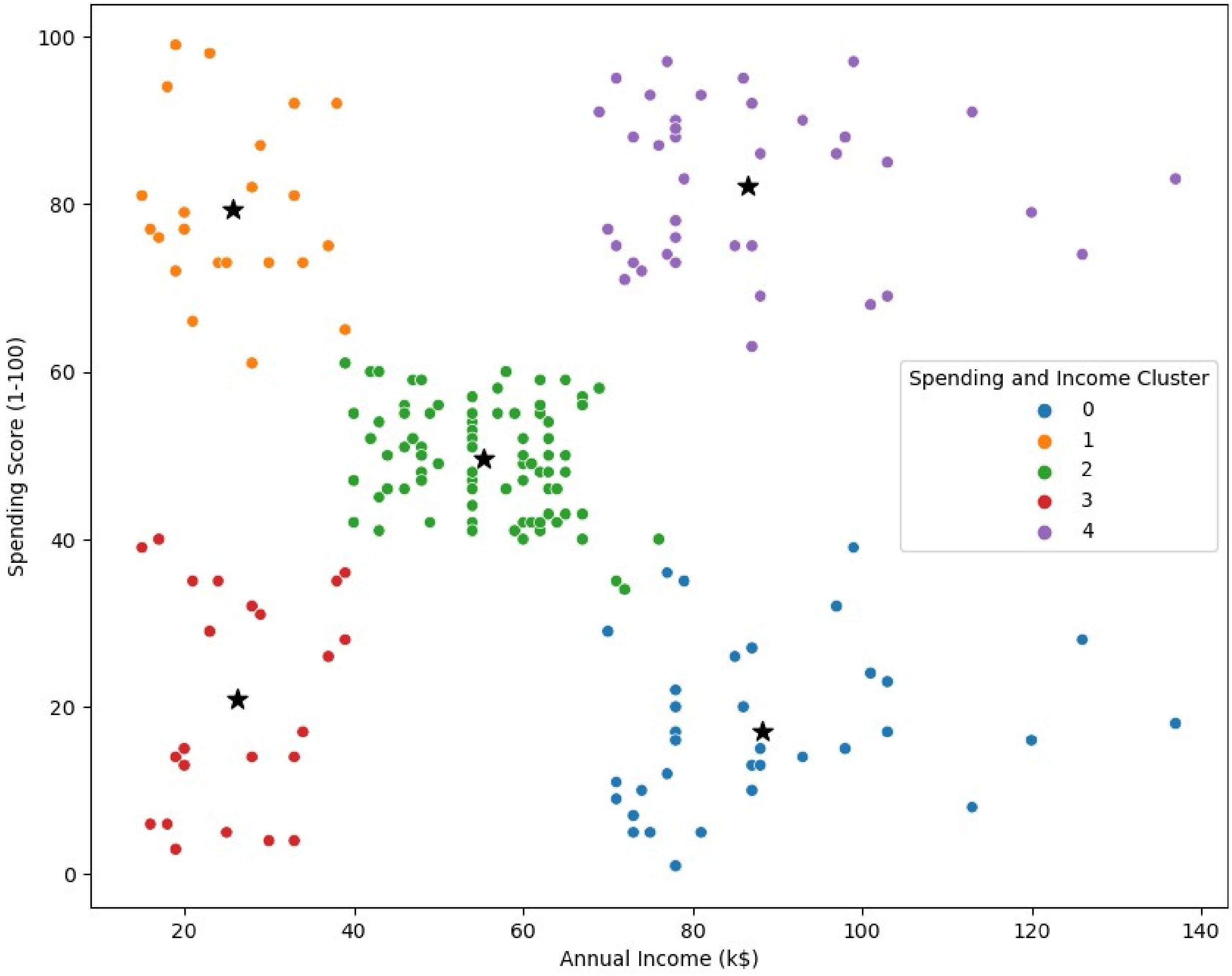
```
   CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)  \
0           1    Male   19                  15                      39

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                            3
1               0                            1
2               0                            3
3               0                            1
4               0                            3
```

```
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 0x156a8618670>]
```



```
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 Cluster',palette='tab10')
plt.savefig('clustering_bivaraiate.png')
```



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

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

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

```
                                   Age  Annual Income (k$)  \
Spending and Income Cluster
0                             41.114286           88.200000
1                             25.272727           25.727273
2                             42.716049           55.296296
```

```
3                                  45.217391          26.304348
4                                  32.692308          86.538462

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

```python
#mulivariate clustering
from sklearn.preprocessing import StandardScaler


scale = StandardScaler()


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


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

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

```
   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               0                            3            1
1               0                            1            1
2               0                            3            0
3               0                            1            0
4               0                            3            0
```

```
dff.columns
```

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

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

```
   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
```

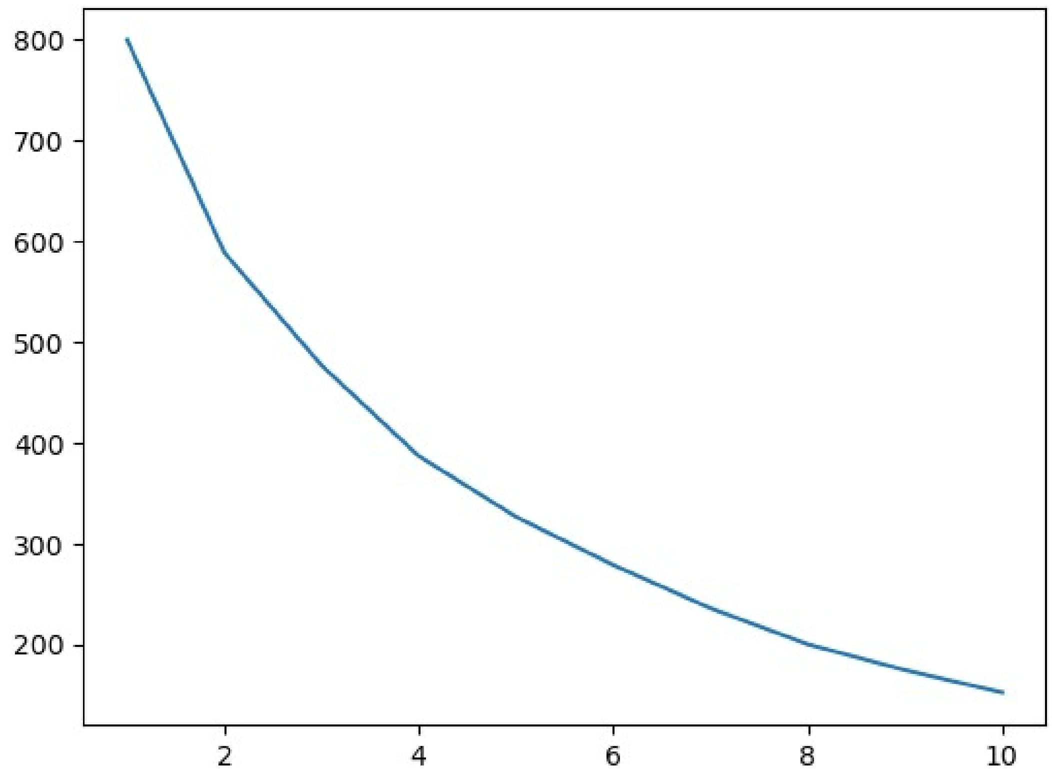
```
dff = scale.fit_transform(dff)
```

```
dff = pd.DataFrame(scale.fit_transform(dff))
dff.head()
```

```
          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
```

```
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)
```

```
[<matplotlib.lines.Line2D at 0x156a88fd580>]
```



```
df

     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                 0                            3
1                 0                            1
2                 0                            3
3                 0                            1
4                 0                            3
..              ...                          ...
195               1                            4
196               1                            0
197               1                            4
198               1                            0
199               1                            4

[200 rows x 7 columns]


df.to_csv('Clustering.csv')
```