In [1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
for dirname, _, filenames in os.walk(r"C:\Users\susha\ML Jupyter Notebook\Mall_Custome
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

In [2]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import silhouette_score
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings("ignore")
```

In [3]:
```
df = pd.read_csv(r"C:\Users\susha\ML Jupyter Notebook\Mall_Customers.csv")
df
```

Out[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
...	...	...	...	...	...
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

200 rows × 5 columns

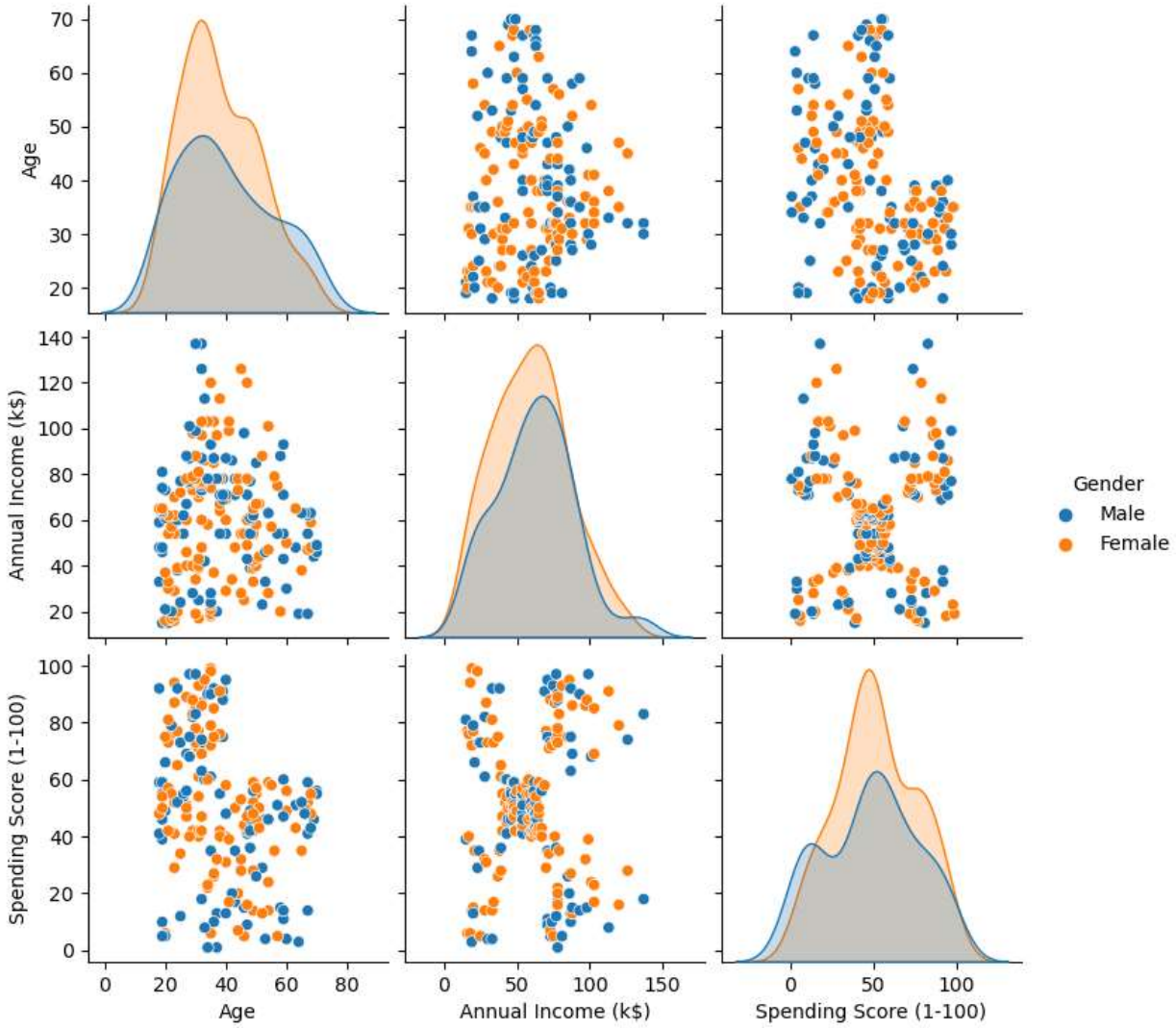
In [4]:
```
df.info()
```

```
<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
```

In [5]:
```
fig= px.scatter(df, x="Annual Income (k$)", y="Spending Score (1-100)", size="Annual I
fig.show()
```

In [6]:
```
df2 = df.drop(columns=["CustomerID"])
sns.pairplot(df2, hue='Gender', kind='scatter', diag_kind='kde')
```

Out[6]:
```
<seaborn.axisgrid.PairGrid at 0x22e57acf550>
```



In [7]:
```
Features = df[["Age", "Annual Income (k$)", "Spending Score (1-100)"]]
```

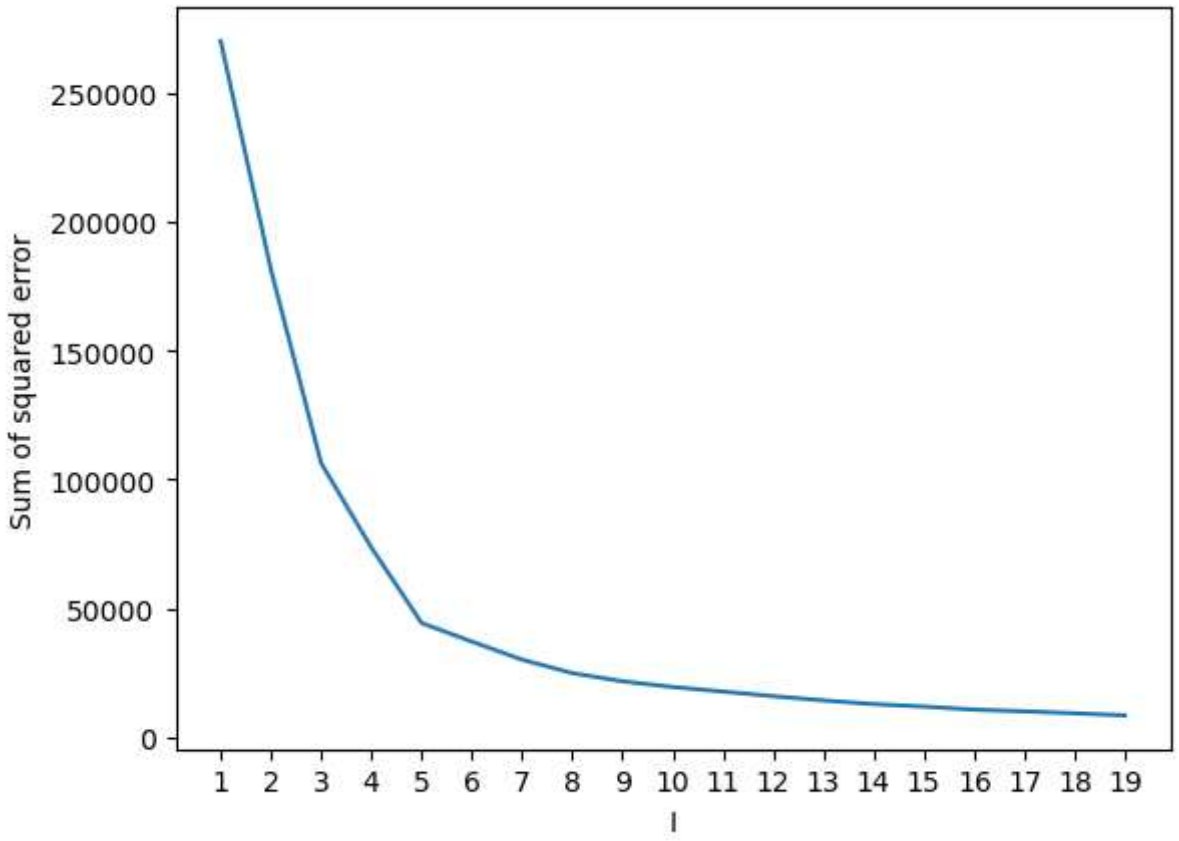
In [8]:
```
sse = []
for i in range (1, 20):
    km = KMeans(n_clusters=i)
    km.fit(df[["Annual Income (k$)", "Spending Score (1-100)"]])
    sse.append(km.inertia_)
```

In [9]:
```
sse
```

Out[9]:
```
[269981.28,
 181363.59595959593,
 106348.37306211118,
 73679.78903948836,
 44448.45544793371,
 37233.81451071001,
 30241.343617936593,
 25029.253424935887,
 21862.092672182887,
 19636.75396489815,
 17879.739741460093,
 16099.925688363925,
 14437.201795784862,
 13015.595876742305,
 12060.324164054335,
 10873.307467532466,
 10220.256699098003,
 9452.271636409567,
 8656.515417262475]
```

In [10]:
```
plt.plot(range(1, 20), sse)
plt.xlabel("I")
plt.ylabel("Sum of squared error")
plt.xticks(range(1, 20))
```

Out[10]:
```
([<matplotlib.axis.XTick at 0x22e57d77250>,
  <matplotlib.axis.XTick at 0x22e58c80a50>,
  <matplotlib.axis.XTick at 0x22e576ab410>,
  <matplotlib.axis.XTick at 0x22e58d2d610>,
  <matplotlib.axis.XTick at 0x22e58d2ee90>,
  <matplotlib.axis.XTick at 0x22e58d38810>,
  <matplotlib.axis.XTick at 0x22e58d3af10>,
  <matplotlib.axis.XTick at 0x22e58d41010>,
  <matplotlib.axis.XTick at 0x22e58d42950>,
  <matplotlib.axis.XTick at 0x22e58d38550>,
  <matplotlib.axis.XTick at 0x22e58d49a50>,
  <matplotlib.axis.XTick at 0x22e58d4bb90>,
  <matplotlib.axis.XTick at 0x22e58d4dd50>,
  <matplotlib.axis.XTick at 0x22e58d4fdd0>,
  <matplotlib.axis.XTick at 0x22e58d4d390>,
  <matplotlib.axis.XTick at 0x22e58d52390>,
  <matplotlib.axis.XTick at 0x22e58d544d0>,
  <matplotlib.axis.XTick at 0x22e58d56550>,
  <matplotlib.axis.XTick at 0x22e58d5c490>],
 [Text(1, 0, '1'),
  Text(2, 0, '2'),
  Text(3, 0, '3'),
  Text(4, 0, '4'),
  Text(5, 0, '5'),
  Text(6, 0, '6'),
  Text(7, 0, '7'),
  Text(8, 0, '8'),
  Text(9, 0, '9'),
  Text(10, 0, '10'),
  Text(11, 0, '11'),
  Text(12, 0, '12'),
  Text(13, 0, '13'),
  Text(14, 0, '14'),
  Text(15, 0, '15'),
  Text(16, 0, '16'),
  Text(17, 0, '17'),
  Text(18, 0, '18'),
  Text(19, 0, '19')])
```



In [11]:
```
model = KMeans(n_clusters= 5 )
y_predict = model.fit_predict(Features)
y_predict
```

Out[11]:
```
array([0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4,
       0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4,
       0, 4, 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, 3, 2, 3, 1, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3])
```

In [12]:
```
df["Cluster"] = y_predict
df.head(10)
```

Out[12]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)	Cluster
0	1	Male	19	15	39	0
1	2	Male	21	15	81	4
2	3	Female	20	16	6	0
3	4	Female	23	16	77	4
4	5	Female	31	17	40	0
5	6	Female	22	17	76	4
6	7	Female	35	18	6	0
7	8	Female	23	18	94	4
8	9	Male	64	19	3	0
9	10	Female	30	19	72	4

In [13]:
```
fig = px.scatter(df, x='Annual Income (k$)', y='Spending Score (1-100)', color='Cluste

# Show the plot
fig.show()
```

In [14]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)', hue='Cluster', dat
plt.scatter(model.cluster_centers_[:, 1], model.cluster_centers_[:, 2], s=200, c='red'
plt.title('Mall Customer Segmentation ')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```