Customer Segmentation case study- Mall Data

Using K-Means clusting

In [40]:
```python
import pandas as pd
```

In [41]:
```python
mall= pd.read_csv(r'C:\Users\Admin\Desktop\Mall_Customers (1).csv')
```

In [42]:
```python
mall.head()
```

Out[42]:

	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

In [43]:
```python
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
```

In [44]:
```python
mall.isnull().sum()
```

Out[44]:
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

Explaratory Data Analysis

Find the top 3 customers based on their Average spending score.

In [47]:
```python
top3 = mall.groupby("CustomerID")[['Spending Score (1-100)']].mean().reset_index().sort_values(by='Spending Score (1-100)',ascending = False).head(3)
```

In [48]:
```python
top3
```

Out[48]:

	CustomerID	Spending Score (1-100)
11	12	99.0
19	20	98.0
145	146	97.0

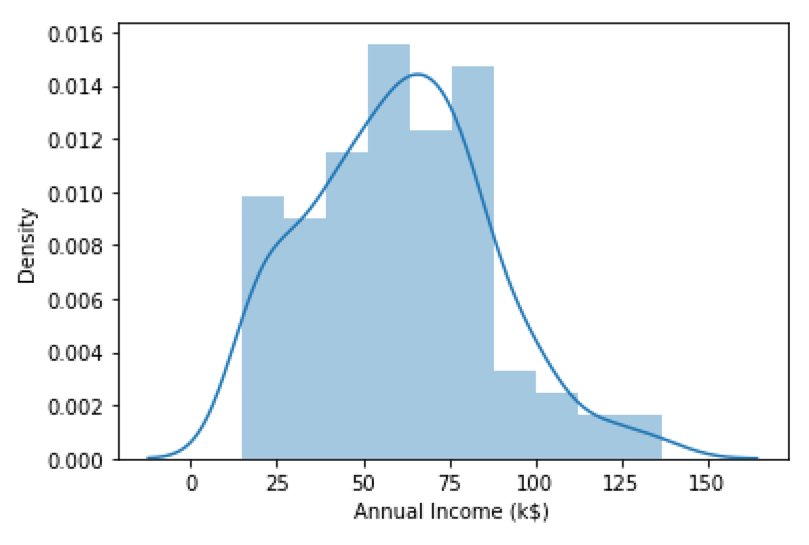
In [49]:
```python
import seaborn as sns
```

Find the distribution of annual income for the customers

In [52]:
```python
sns.distplot(mall['Annual Income (k$)'])
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[52]:
```
<AxesSubplot:xlabel='Annual Income (k$)', ylabel='Density'>
```

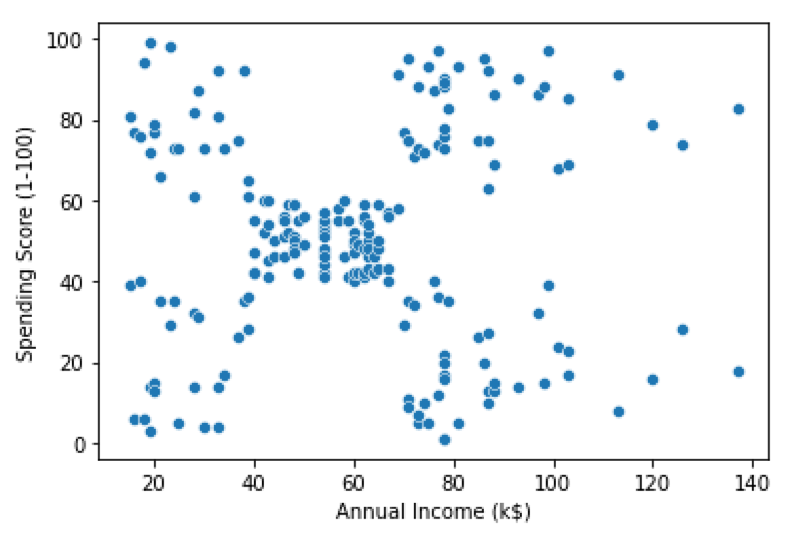


Draw a scatter plot between annual income and spending score.

In [53]:
```python
sns.scatterplot(mall['Annual Income (k$)'],mall['Spending Score (1-100)'])
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[53]:
```
<AxesSubplot:xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'>
```



Data Preprocessing

In [54]:
```python
# Dummy Variables

mall_dummy = pd.get_dummies(mall,columns=['Gender'],drop_first=True)
```

In [55]:
```python
mall_dummy.head()
```

Out[55]:

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

In [59]:
```python
mall_final = mall_dummy.drop(columns=['CustomerID'])
mall_final.head()
```

Out[59]:

	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

Building the clustering model

In [60]:
```python
from sklearn.cluster import KMeans
```

In [65]:
```python
# creating the clusters

for k in range(1,16):
    print (k)
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

In [66]:
```python
error = []
for k in range(1,16):
    km = KMeans(n_clusters=k)
    km.fit(mall_final)
    error.append(km.inertia_)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:881: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [67]:
```python
error
```

Out[67]:
```
[308862.06000000006,
 212889.44245524294,
 143391.59236035674,
 104414.67534220174,
 75399.61541401486,
 58348.64136331504,
 51130.69008126375,
 44355.31351771351,
 40884.85288045286,
 37481.34709919711,
 34626.17513808031,
 32175.494181903647,
 30439.972741289475,
 28528.727536416405,
 26795.873974509897]
```

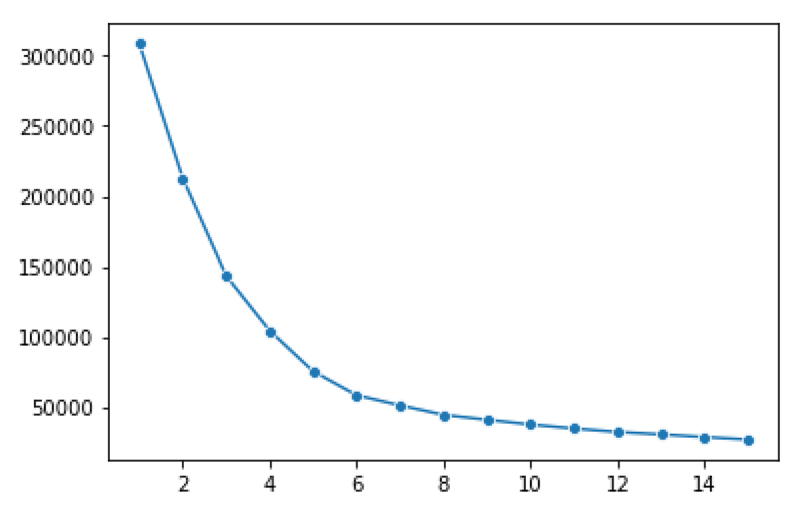
We will plot Number of Clusters in x-axis and Error in y-axis.

In [68]:
```python
# Lets plot the graph

sns.lineplot(range(1,16),error, marker ='o')
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[68]:
```
<AxesSubplot:>
```



In [69]:
```python
# We can choose the number of clusters as 5.

km = KMeans(n_clusters=5)
km.fit(mall_final)
```

Out[69]:
```
KMeans(n_clusters=5)
```

In [72]:
```python
mall_dummy['Cluster']= km.predict(mall_final)
mall_dummy
```

Out[72]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)	Gender_Male	Cluster
0	1	19	15	39	1	0
1	2	21	15	81	1	4
2	3	20	16	6	0	0
3	4	23	16	77	0	4
4	5	31	17	40	0	0
...	...	...	...	...	...	...
195	196	35	120	79	0	3
196	197	45	126	28	0	2
197	198	32	126	74	1	3
198	199	32	137	18	1	2
199	200	30	137	83	1	3

200 rows × 6 columns

In [74]:
```python
mall_dummy.groupby('Cluster').mean()
```

Out[74]:

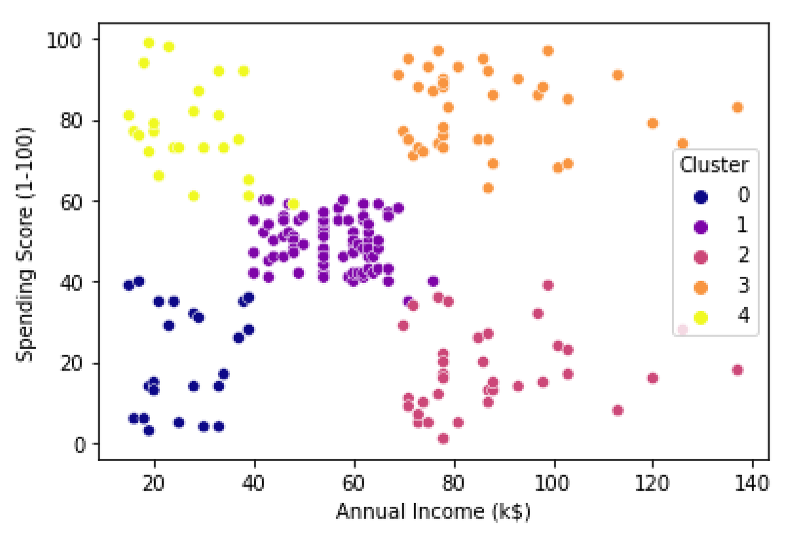
	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)	Gender_Male
Cluster					
0	23.000000	45.217391	26.304348	20.913043	0.391304
1	86.753247	43.727273	55.480519	49.324675	0.402597
2	163.500000	40.666667	87.750000	17.583333	0.527778
3	162.000000	32.692308	86.538462	82.128205	0.461538
4	27.480000	24.960000	28.040000	77.000000	0.440000

In [78]:
```python
#Lets plot the scatterplot


sns.scatterplot(mall_dummy['Annual Income (k$)'],mall_dummy['Spending Score (1-100)'],hue=mall_dummy['Cluster'],palette='plasma')
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[78]:
```
<AxesSubplot:xlabel='Annual Income (k$)', ylabel='Spending Score (1-100)'>
```



In [ ]: