Mall Customer Segmentation Analysis

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data=pd.read_csv("C:/Users/rockz/OneDrive/Documents/Mall Customers.csv")
data
```

Out[2]:

	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 [3]:
```
data.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 [4]:
```
data.nunique()
```

Out[4]:
```
CustomerID                200
Gender                      2
Age                        51
Annual Income (k$)         64
Spending Score (1-100)     84
dtype: int64
```

In [5]:
```
data.columns
```

Out[5]:
```
Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

In [6]:
```
data.describe()
```

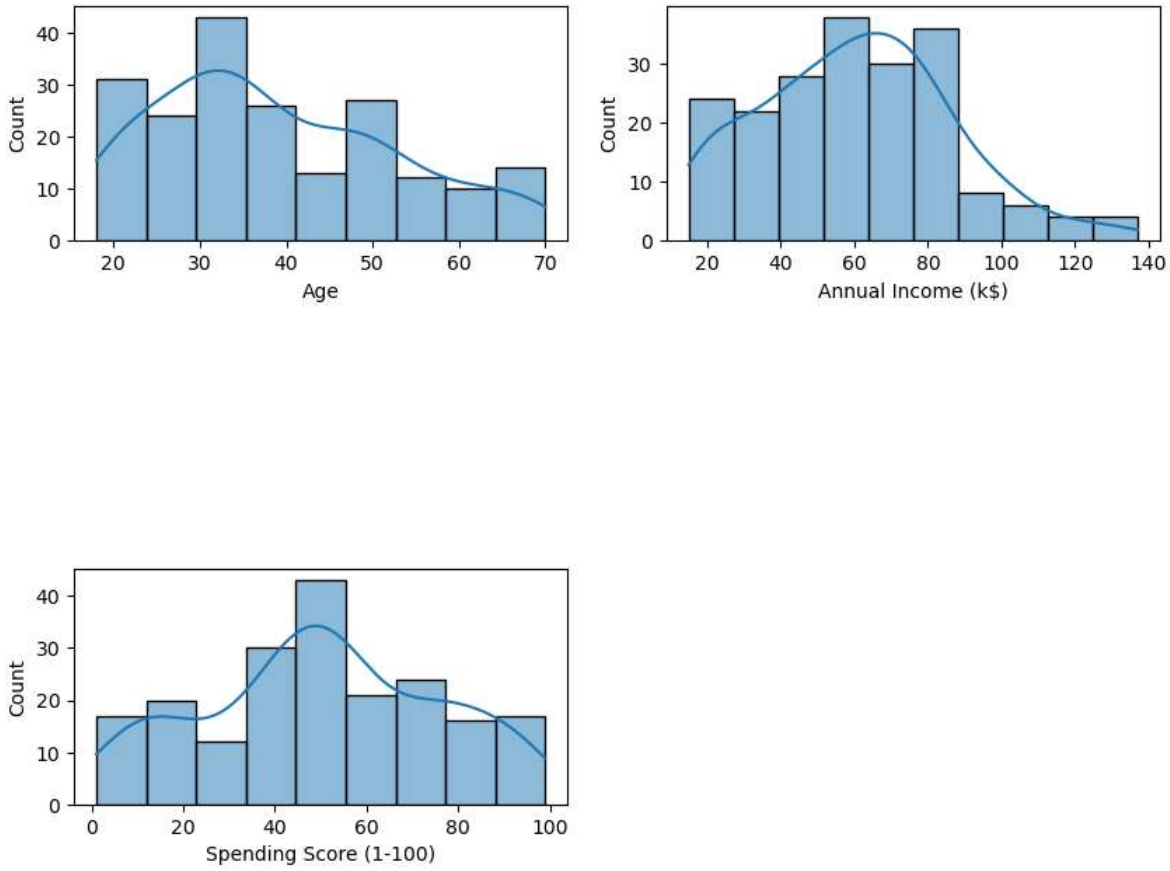
Out[6]:

	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

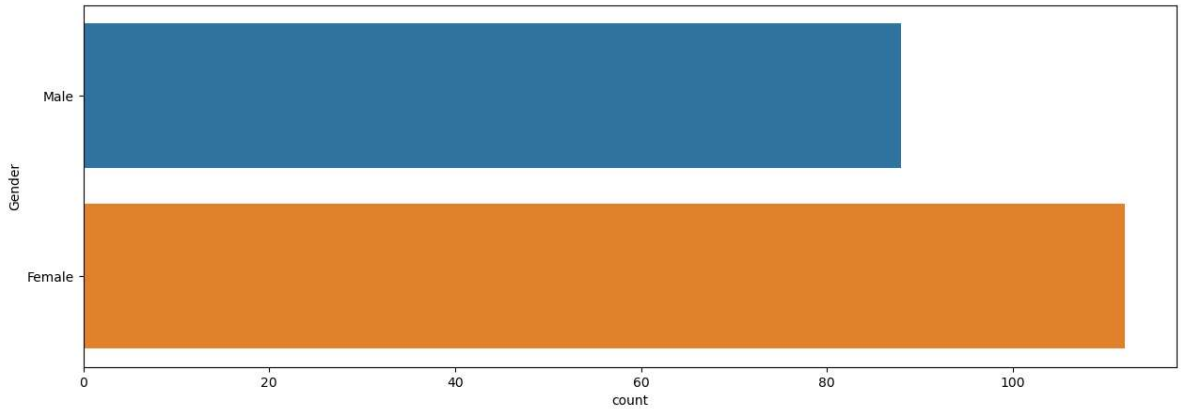
Visualization

In [7]:
```
plt.figure(figsize=(10,10))
plt.subplot(4,2,1)
sns.histplot(data=data['Age'], x=data['Age'],kde=True)
plt.subplot(4,2,2)
sns.histplot(data=data['Annual Income (k$)'],x=data['Annual Income (k$)'],kde=
plt.subplot(4,2,5)
sns.histplot(data=data['Spending Score (1-100)'],x=data['Spending Score (1-100
```

Out[7]: <AxesSubplot:xlabel='Spending Score (1-100)', ylabel='Count'>

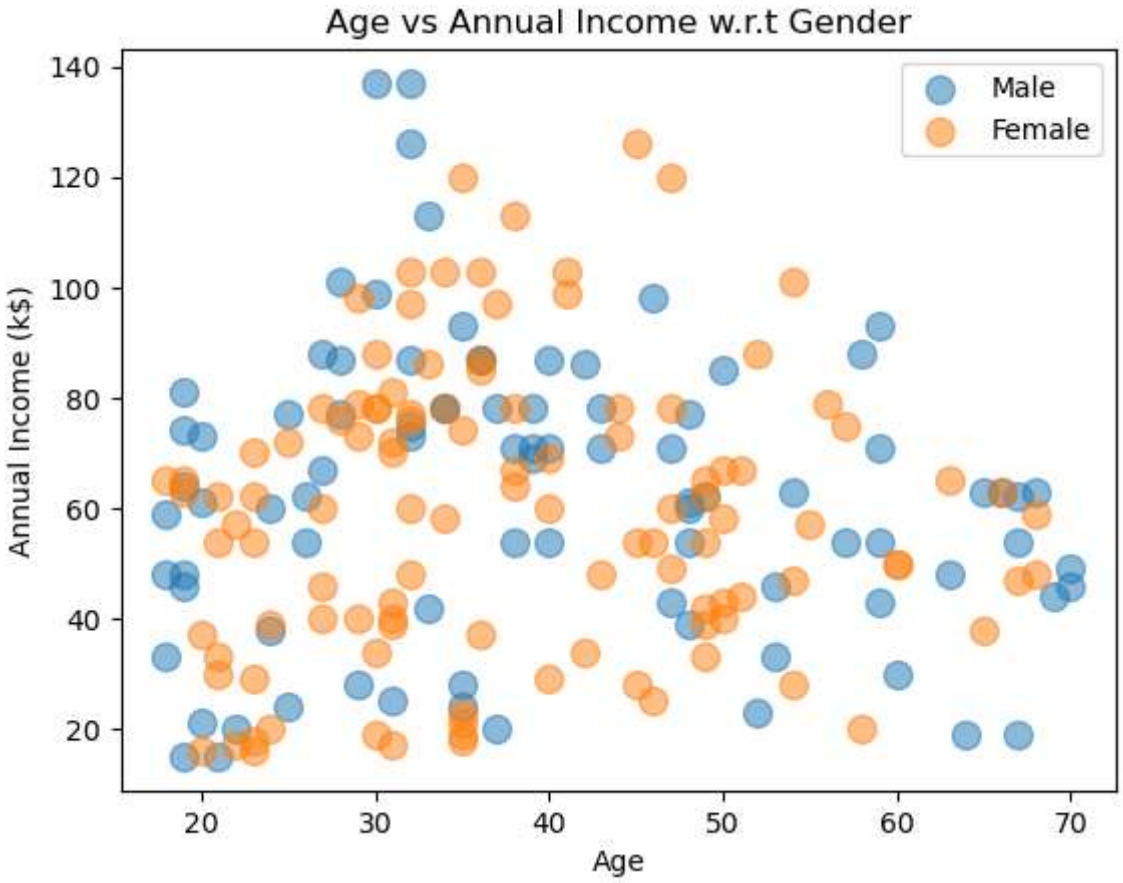


In [8]:
```
plt.figure(1, figsize = (15, 5))
sns.countplot(y = 'Gender',data = data)
plt.show()
```

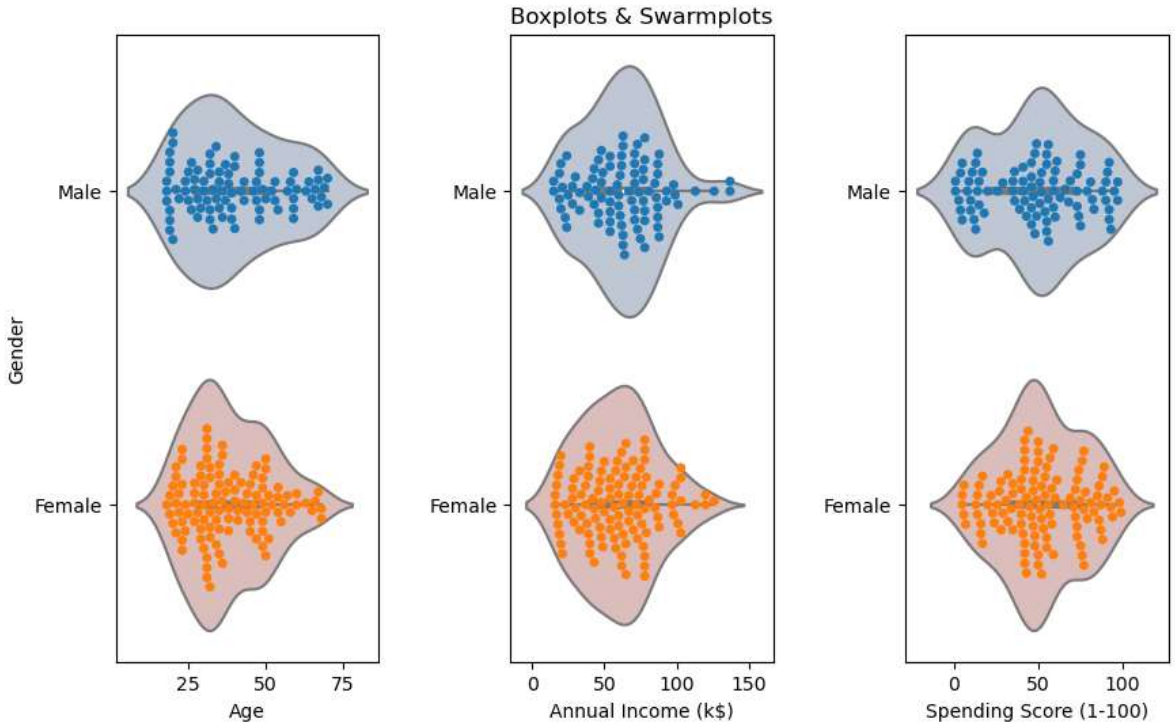


In [9]:
```
plt.figure(1)
for gender in ['Male','Female']:
    plt.scatter(x='Annual Income (k$)',y='Spending Score (1-100)',data=data[da
plt.xlabel('Annual Income (k$)'),plt.ylabel('Spending Score (1-100)')
plt.title('Annual Income vs Spending Score w.r.t Gender')
plt.legend()
plt.show()
```

In [10]:
```
plt.figure(1)
for gender in ['Male','Female']:
    plt.scatter(x='Age',y='Annual Income (k$)',data=data[data['Gender']==gende
plt.xlabel('Age'),plt.ylabel('Annual Income (k$)')
plt.title('Age vs Annual Income w.r.t Gender')
plt.legend()
plt.show()
```



In [11]:
```
plt.figure(1,figsize=(10,6))
n=0
for cols in ['Age','Annual Income (k$)','Spending Score (1-100)']:
    n += 1
    plt.subplot(1,3,n)
    plt.subplots_adjust(hspace=0.5,wspace=0.5)
    sns.violinplot(x=cols,y='Gender',data=data,palette='vlag')
    sns.swarmplot(x=cols,y='Gender',data=data)
    plt.ylabel('Gender' if n==1 else '')
    plt.title('Boxplots & Swarmplots'if n==2 else '')
plt.show()
```



In [12]:
```
plt.figure(1)
for gender in ['Male','Female']:
    plt.scatter(x='Age',y='Spending Score (1-100)',data=data[data['Gender']==g
plt.xlabel('Age'),plt.ylabel('Spending Score (1-100)')
plt.title('Age vs Spending Score w.r.t Gender')
plt.legend()
plt.show()
```

In [13]:
```
x=data[['Gender','Age','Annual Income (k$)','Spending Score (1-100)']].values
x[0:5]
```

Out[13]:
```
array([['Male', 19, 15, 39],
       ['Male', 21, 15, 81],
       ['Female', 20, 16, 6],
       ['Female', 23, 16, 77],
       ['Female', 31, 17, 40]], dtype=object)
```

In [14]:
```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
le.fit(['Male','Female'])
x[:,0] = le.transform(x[:,0])
```

In [15]:
```
x[0:5]
```

Out[15]:
```
array([[1, 19, 15, 39],
       [1, 21, 15, 81],
       [0, 20, 16, 6],
       [0, 23, 16, 77],
       [0, 31, 17, 40]], dtype=object)
```

Modelling

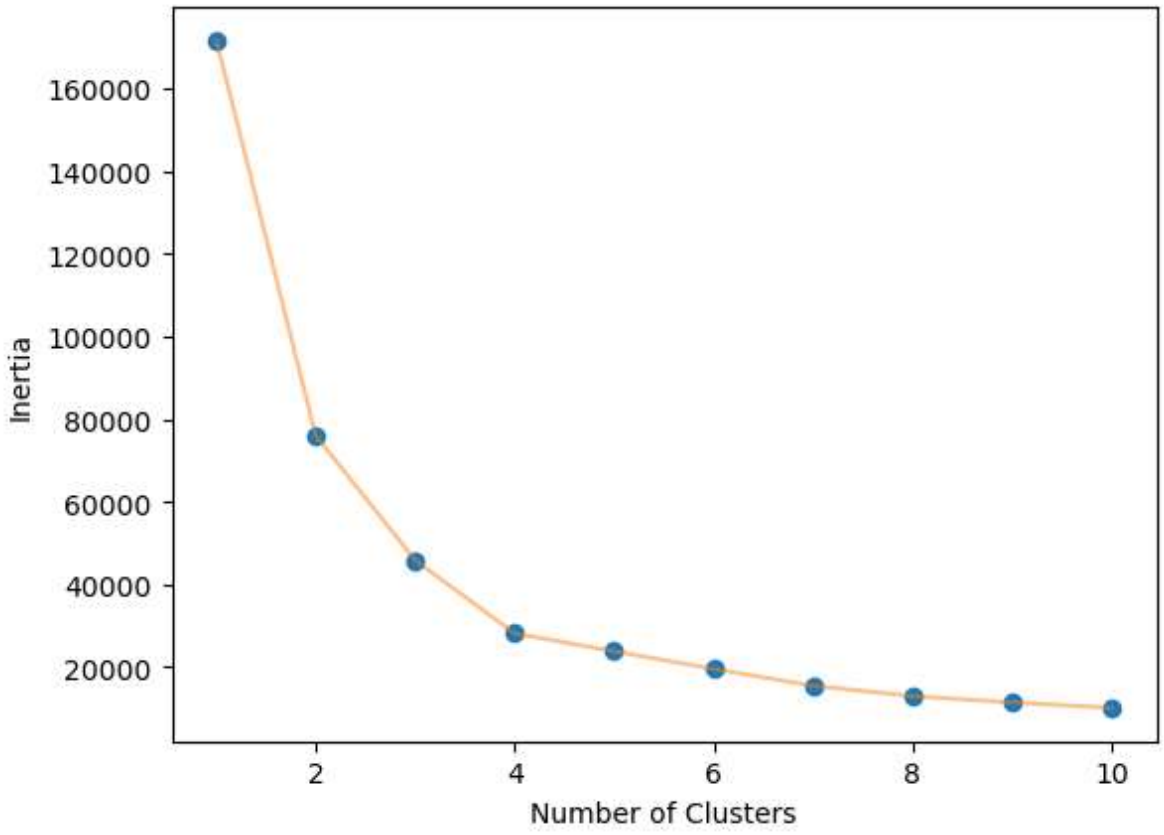
K-Means

In [16]:
```
from sklearn.cluster import KMeans
```

In [17]:
```
x1=data[['Age','Spending Score (1-100)']].iloc[:,:].values
inertia=[]
for n in range(1,11):
    algorithm=(KMeans(n_clusters=n,init='k-means++',n_init=10,max_iter=300,tol
    algorithm.fit(x1)
    inertia.append(algorithm.inertia_)
```

```
C:\Users\rockz\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1036: U
serWarning: KMeans is known to have a memory leak on Windows with MKL, when t
here are less chunks than available threads. You can avoid it by setting the
environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

In [18]:
```
plt.figure(1)
plt.plot(np.arange(1,11),inertia,'o')
plt.plot(np.arange(1,11),inertia,'-',alpha=0.5)
plt.xlabel('Number of Clusters'),plt.ylabel('Inertia')
plt.show()
```



In [20]:
```
algorithm=(KMeans(n_clusters=3,init='k-means++',n_init=10,max_iter=300,tol=0.0
```

In [24]:
```
pred=algorithm.fit_predict(x1)
pred
```

Out[24]:
```
array([2, 0, 1, 0, 2, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 2, 0, 1, 0, 2, 0,
       1, 0, 1, 0, 2, 2, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 2, 0, 2, 2,
       1, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0,
       2, 2, 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, 0, 1, 0, 2, 0, 1, 0, 1, 0,
       2, 0, 1, 0, 1, 0, 1, 0, 1, 0, 2, 0, 1, 0, 2, 0, 1, 0, 1, 0, 1, 0,
       1, 0, 1, 0, 1, 0, 2, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,
       1, 0, 1, 0, 2, 0, 1, 0, 2, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0,
       1, 0])
```

In [29]:
```
data['cluster']=pred
data
```

Out[29]:

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

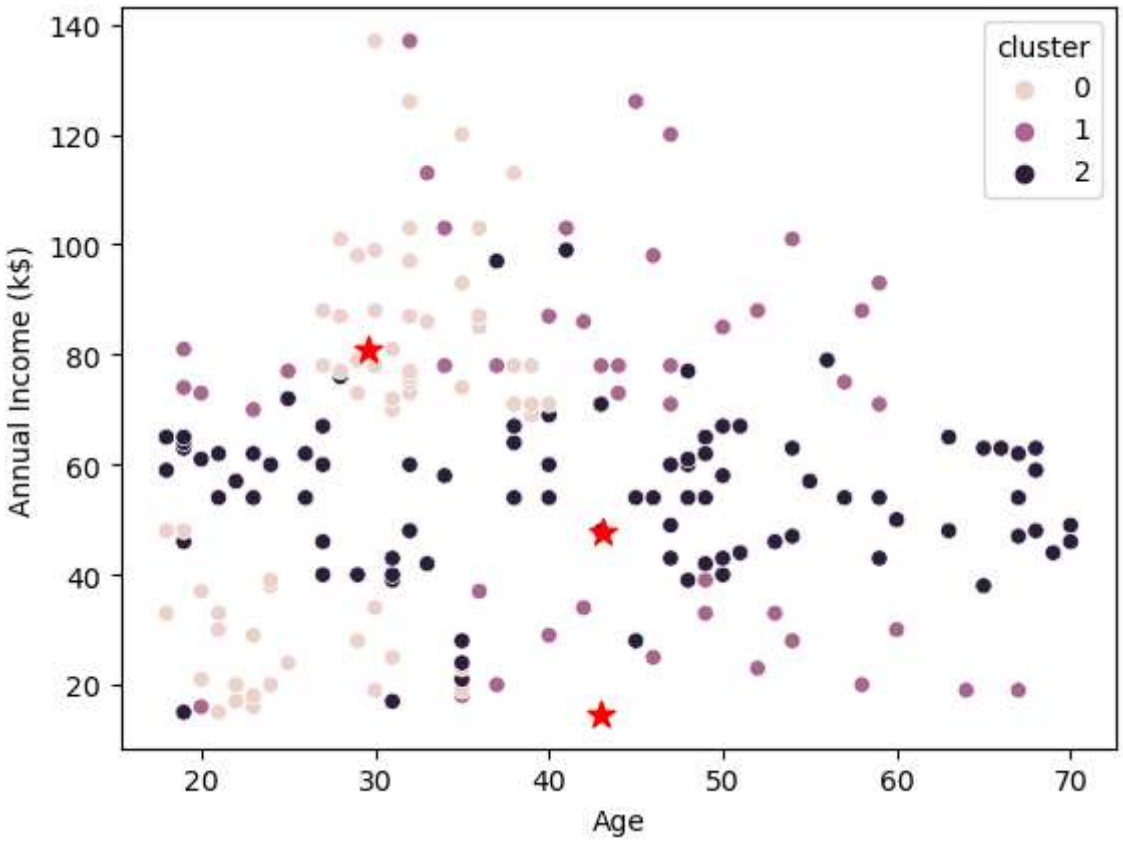
200 rows × 6 columns

In [30]:
```
m=algorithm.cluster_centers_
m
```

Out[30]:
```
array([[29.56451613, 80.74193548],
       [42.95744681, 14.59574468],
       [43.05494505, 47.78021978]])
```

In [33]:
```
sns.scatterplot(x=data['Age'],y=data['Annual Income (k$)'],hue=data['cluster'])
plt.scatter(x=m[:,0],y=m[:,1],marker="*",s=100,color="red")
```

Out[33]: <matplotlib.collections.PathCollection at 0x1ea157e9070>



In [ ]: