ASSIGNMENT-4

Problem Statement: Customer Segmentation Analysis

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

from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report

from sklearn.svm import SVC
```

In [2]:
```
data = pd.read_csv("C:/Users/MANOHARI/Downloads/Mall_Customers.csv")
```

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
```

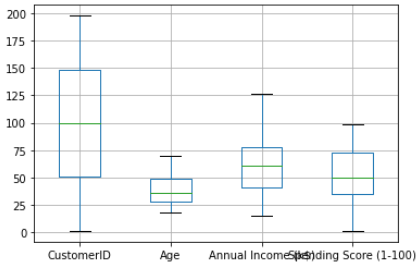
In [4]:
```
data.describe()
```

Out[4]:

	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

In [6]:
```
data.drop(data[data['Annual Income (k$)']>130].index,inplace=True)
data.boxplot()
```

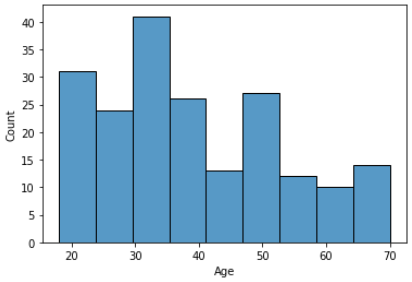
Out[6]:
```
<AxesSubplot:>
```



UNIVARIATE ANALYSIS

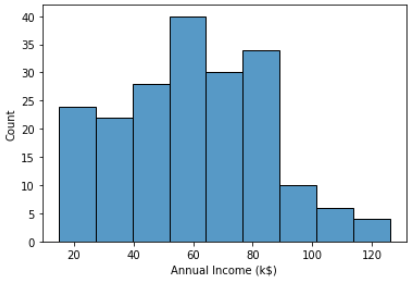
In [7]:
```
sns.histplot(data['Age'])
```

Out[7]:
```
<AxesSubplot:xlabel='Age', ylabel='Count'>
```



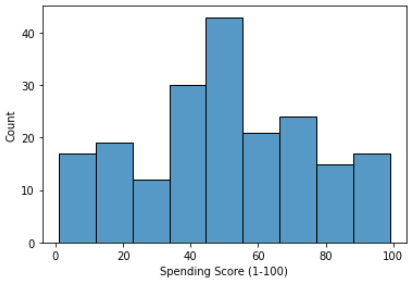
In [8]:
```
sns.histplot(data['Annual Income (k$)'])
```

Out[8]:
```
<AxesSubplot:xlabel='Annual Income (k$)', ylabel='Count'>
```



In [9]:
```
sns.histplot(data['Spending Score (1-100)'])
```

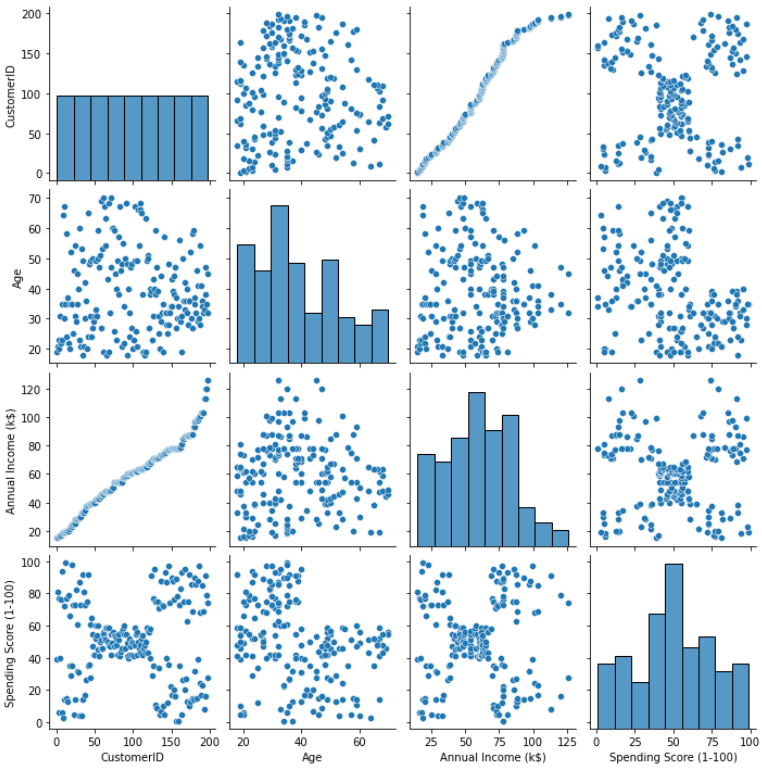
Out[9]:
```
<AxesSubplot:xlabel='Spending Score (1-100)', ylabel='Count'>
```



MULTIVARIATE ANALYSIS

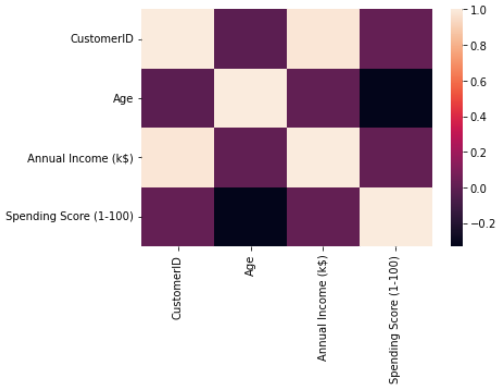
In [10]:
```
sns.pairplot(data)
```

Out[10]:
```
<seaborn.axisgrid.PairGrid at 0x19165d98a30>
```



In [11]:
```
sns.heatmap(data.corr())
```

Out[11]:
```
<AxesSubplot:>
```



ENCODING

In [12]:
```
encode = LabelEncoder()
data['Gender']=encode.fit_transform(data['Gender'])
data.head()
```

Out[12]:

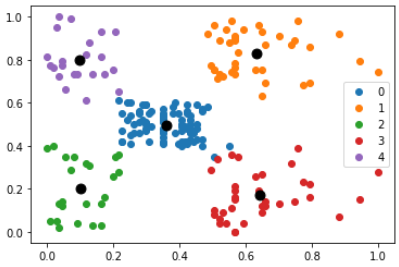
	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	1	19	15	39
1	2	1	21	15	81
2	3	0	20	16	6
3	4	0	23	16	77
4	5	0	31	17	40

In [13]:
```
scaling = MinMaxScaler()
data[['Annual Income (k$)']]=scaling.fit_transform(data[['Annual Income (k$)']])
data[['Spending Score (1-100)']]=scaling.fit_transform(data[['Spending Score (1-100)']])
data.head()
```

Out[13]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	1	19	0.000000	0.387755
1	2	1	21	0.000000	0.816327
2	3	0	20	0.009009	0.051020
3	4	0	23	0.009009	0.775510
4	5	0	31	0.018018	0.397959

In [14]:
```
clus=data.iloc[:, [3,4]].values
kmeans = KMeans(n_clusters=5,random_state=0)
label = data['Cluster'] = kmeans.fit_predict(clus)
centroids = kmeans.cluster_centers_
u_labels = np.unique(label)
for i in u_labels:
    plt.scatter(clus[label == i,0] , clus[label == i,1] , label = i)
plt.scatter(centroids[:,0] , centroids[:,1] , s = 80, color = 'black')
plt.legend()
plt.show()
```



TRAIN AND TEST SPLIT

In [15]:
```
data['Cluster'] = data['Cluster'].astype("category")
X=data.drop(['CustomerID','Cluster'],axis=1)
y=data['Cluster']
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=10)
model = SVC()
model.fit(X_train,y_train)
```

Out[15]:
```
SVC()
```

In [16]:
```
y_predict = model.predict(X_test)
```

In [17]:
```
plt.figure(figsize = (18,8))
sns.heatmap(confusion_matrix(y_test, y_predict), annot = True, xticklabels = y_test.unique(), yticklabels = y_test.unique())
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```