```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import AdaBoostClassifier
```

```
data = pd.read_csv('Mall_Customers.csv')
```

```
data
```

```
     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 x 5 columns]
```

```
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
```

```
data.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
```

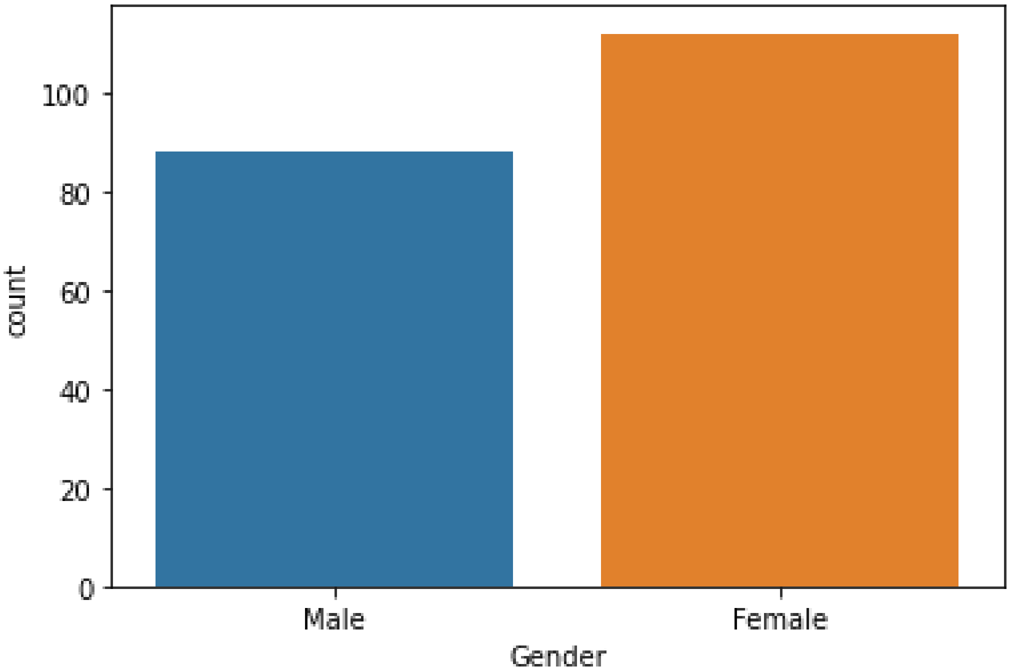
```
data.isnull().sum()
```

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

```
sns.countplot(data.Gender)
```

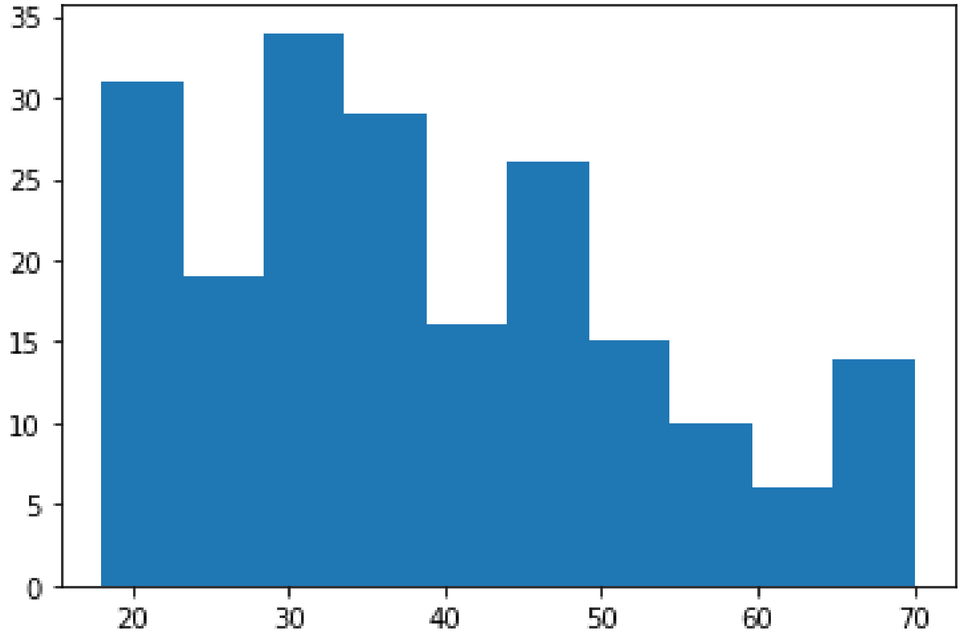
```
D:\Anaconda\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From
version 0.12, the only valid positional argument will be `data`, and
passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```



```
plt.hist(data.Age)
```

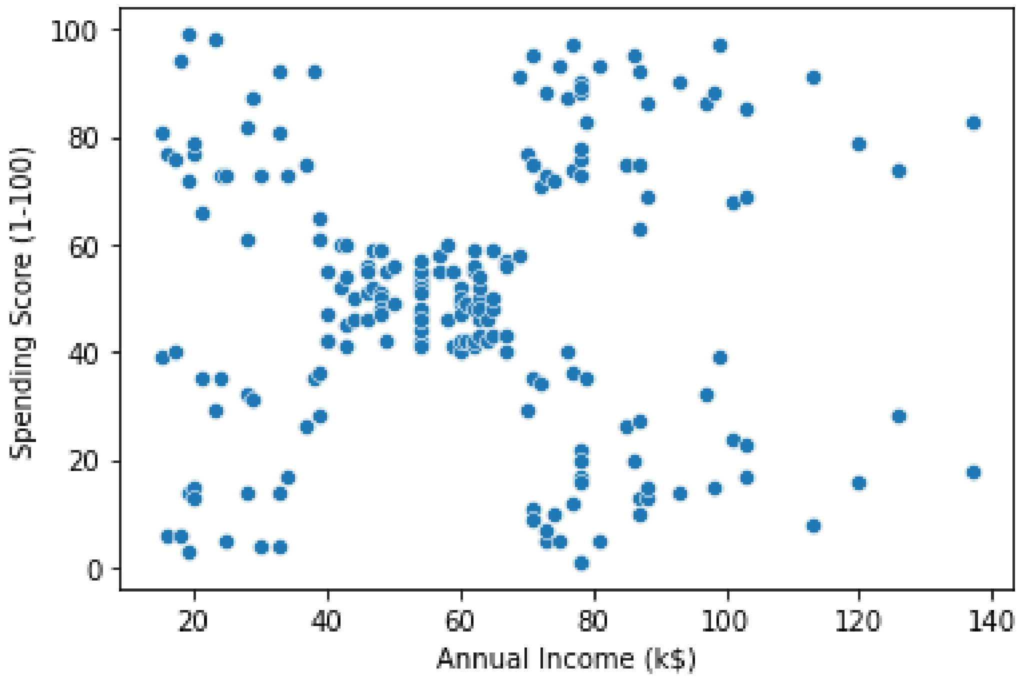
```
(array([31., 19., 34., 29., 16., 26., 15., 10.,  6., 14.]),
 array([18. , 23.2, 28.4, 33.6, 38.8, 44. , 49.2, 54.4, 59.6, 64.8,
70. ]),
 <BarContainer object of 10 artists>)
```



```
sns.scatterplot(data['Annual Income (k$)'],data['Spending Score (1-
100)'])
```

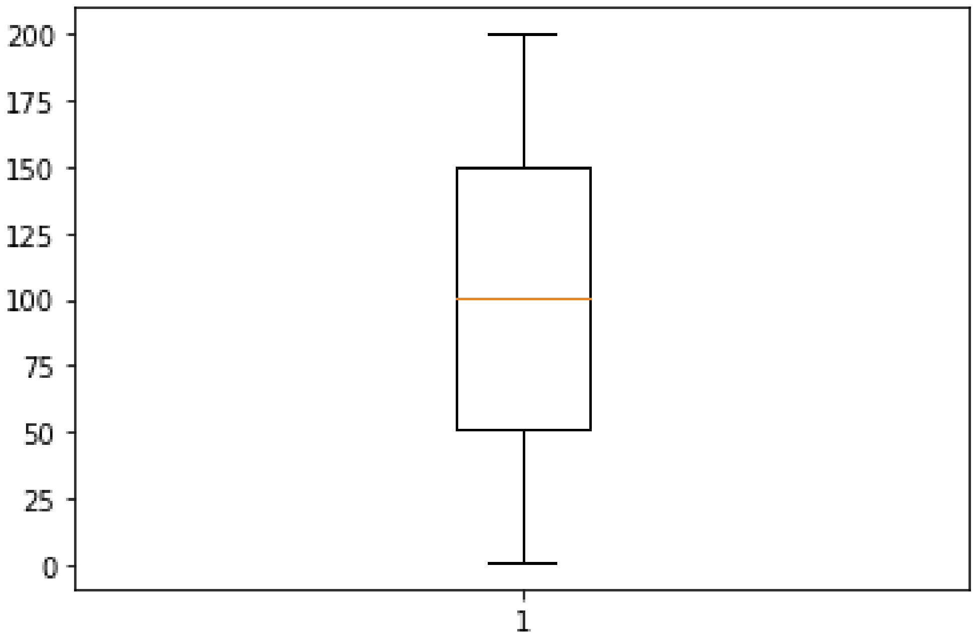
```
D:\Anaconda\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
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

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



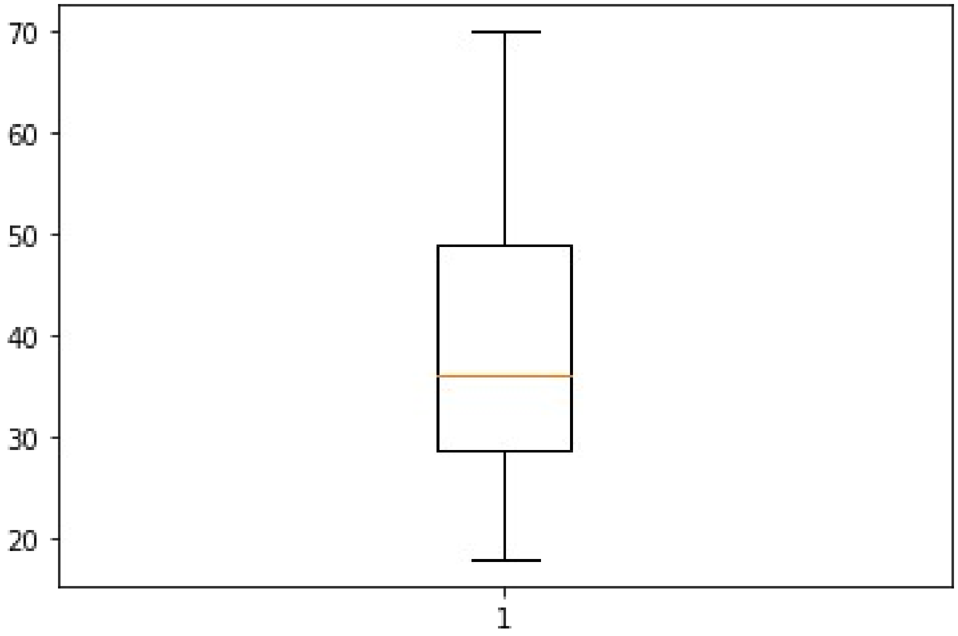
```
plt.boxplot(data.iloc[0:,0])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1a310b28940>,
  <matplotlib.lines.Line2D at 0x1a310b28c10>],
 'caps': [<matplotlib.lines.Line2D at 0x1a310b28fa0>,
  <matplotlib.lines.Line2D at 0x1a310b38370>],
 'boxes': [<matplotlib.lines.Line2D at 0x1a310b284f0>],
 'medians': [<matplotlib.lines.Line2D at 0x1a310b38700>],
 'fliers': [<matplotlib.lines.Line2D at 0x1a310b38a90>],
 'means': []}
```



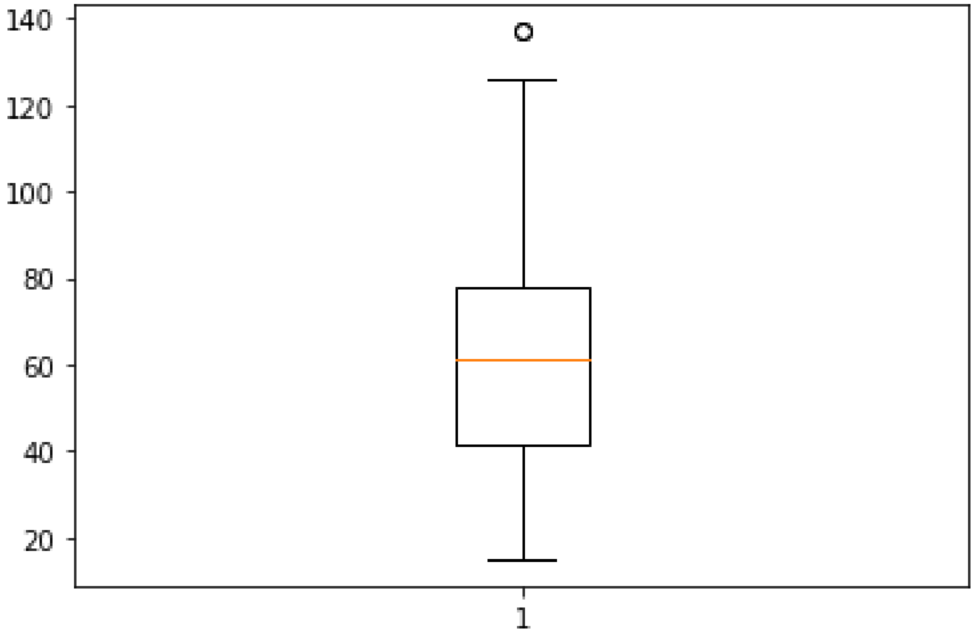
```
plt.boxplot(data.iloc[0:,2])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1a310e8b2e0>,
  <matplotlib.lines.Line2D at 0x1a310e8b5e0>],
 'caps': [<matplotlib.lines.Line2D at 0x1a310e8b940>,
  <matplotlib.lines.Line2D at 0x1a310e8bcd0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1a310e7df70>],
 'medians': [<matplotlib.lines.Line2D at 0x1a310e940a0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1a310e94430>],
 'means': []}
```



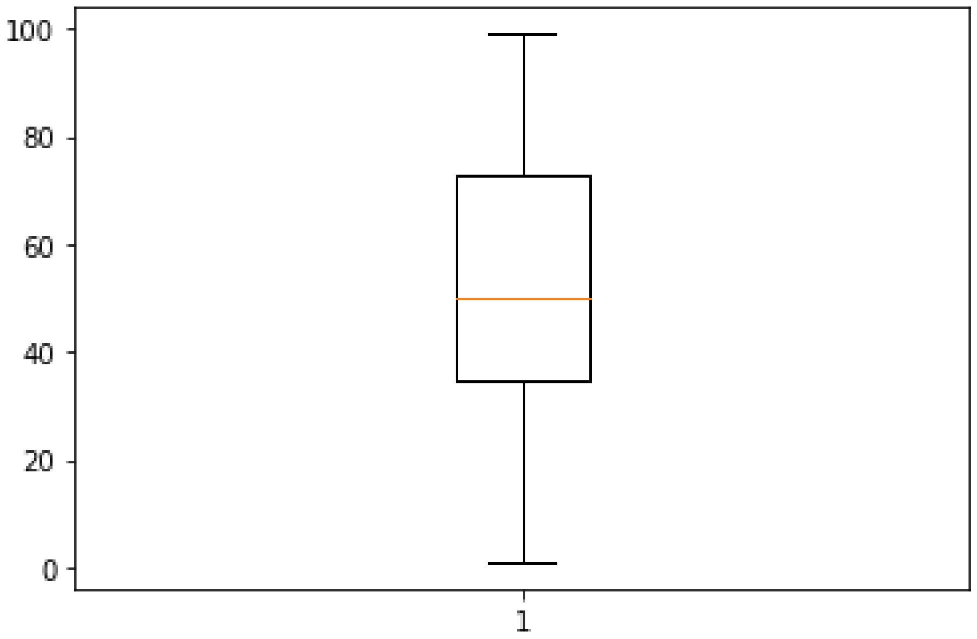
```
plt.boxplot(data.iloc[0:,3])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1a310ee9be0>,
  <matplotlib.lines.Line2D at 0x1a310ee9f70>],
 'caps': [<matplotlib.lines.Line2D at 0x1a310ef6340>,
  <matplotlib.lines.Line2D at 0x1a310ef66d0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1a310ee9910>],
 'medians': [<matplotlib.lines.Line2D at 0x1a310ef6a60>],
 'fliers': [<matplotlib.lines.Line2D at 0x1a310ef6df0>],
 'means': []}
```



```
plt.boxplot(data.iloc[0:,4])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1a310f54a60>,
  <matplotlib.lines.Line2D at 0x1a310f54df0>],
 'caps': [<matplotlib.lines.Line2D at 0x1a310f631c0>,
  <matplotlib.lines.Line2D at 0x1a310f63550>],
 'boxes': [<matplotlib.lines.Line2D at 0x1a310f546d0>],
 'medians': [<matplotlib.lines.Line2D at 0x1a310f638e0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1a310f63c70>],
 'means': []}
```



```
data = pd.get_dummies(data,columns=["Gender"])
```

```
data
```

```
     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   
..          ...  ...                 ...                     ...   
195         196   35                 120                      79   
196         197   45                 126                      28   
197         198   32                 126                      74   
198         199   32                 137                      18   
199         200   30                 137                      83   

     Gender_Female  Gender_Male  
0                0            1  
1                0            1  
2                1            0  
3                1            0  
4                1            0  
..             ...          ...  
195              1            0  
196              1            0  
197              0            1  
198              0            1  
199              0            1  
```

```
[200 rows x 6 columns]
```

```
data.drop('CustomerID',axis=1,inplace=True)
```

```
x = data.drop('Spending Score (1-100)',axis=1)
y = data['Spending Score (1-100)']
```

```
trainx,testx,trainy,testy =
x.iloc[0:150,0:],x.iloc[150:,0:],y.iloc[0:150],y.iloc[150:]
```

```
model1 = RandomForestClassifier().fit(trainx,trainy)
model1.score(testx,testy)
```

```
0.04
```

```
model2 = GradientBoostingClassifier().fit(trainx,trainy)
model2.score(testx,testy)
```

```
0.02
```

```
model3 = AdaBoostClassifier().fit(trainx,trainy)
model3.score(testx,testy)
```

```
0.02
```

```
model3.predict(testx)
```

```
array([90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90,
90,
       90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90,
90,
       90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90, 90,
90],
      dtype=int64)
```

```
pred_y  = model1.predict(testx)
```

```
mean_squared_error(testy,pred_y)
```

```
1749.56
```