In [1]:
```
# importing neccessary and required libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score
```

In [2]:
```
# loading data set
df = pd.read_csv('Iris.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

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

Out[4]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

In [5]:
```
df.shape
```

Out[5]:
```
(150, 6)
```

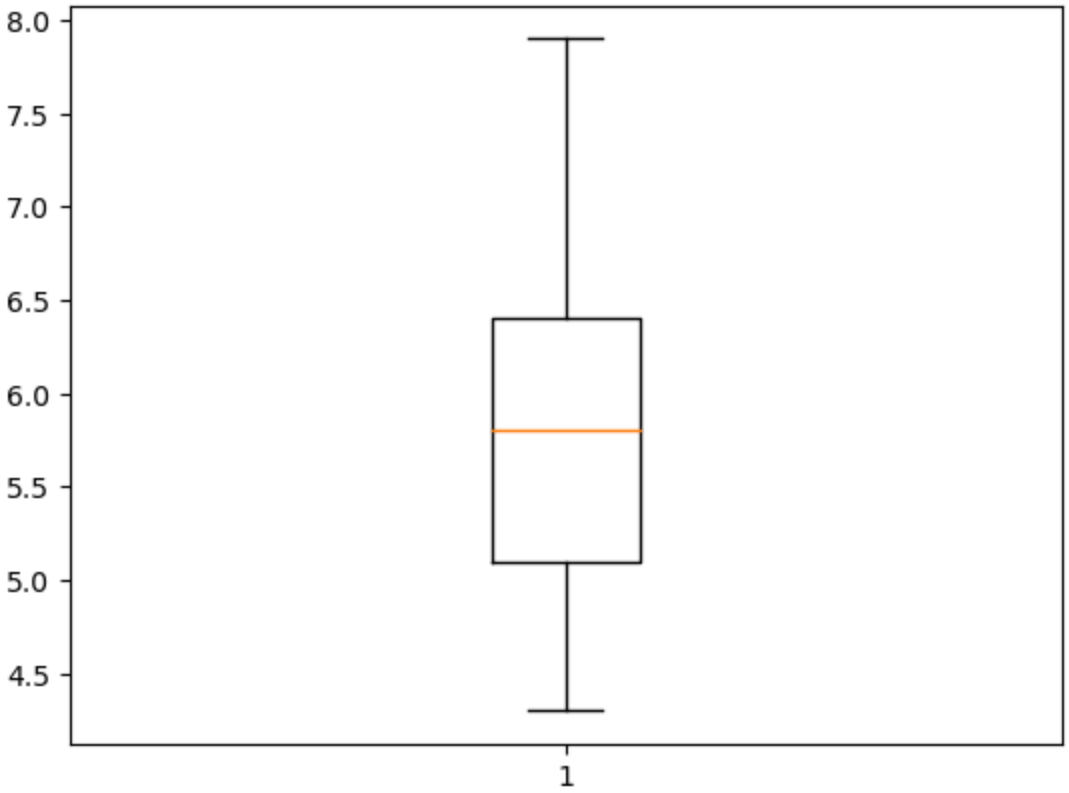
In [6]:
```
data = df.groupby('Species')
```

In [7]:
```
df['Species'].unique()
```

Out[7]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

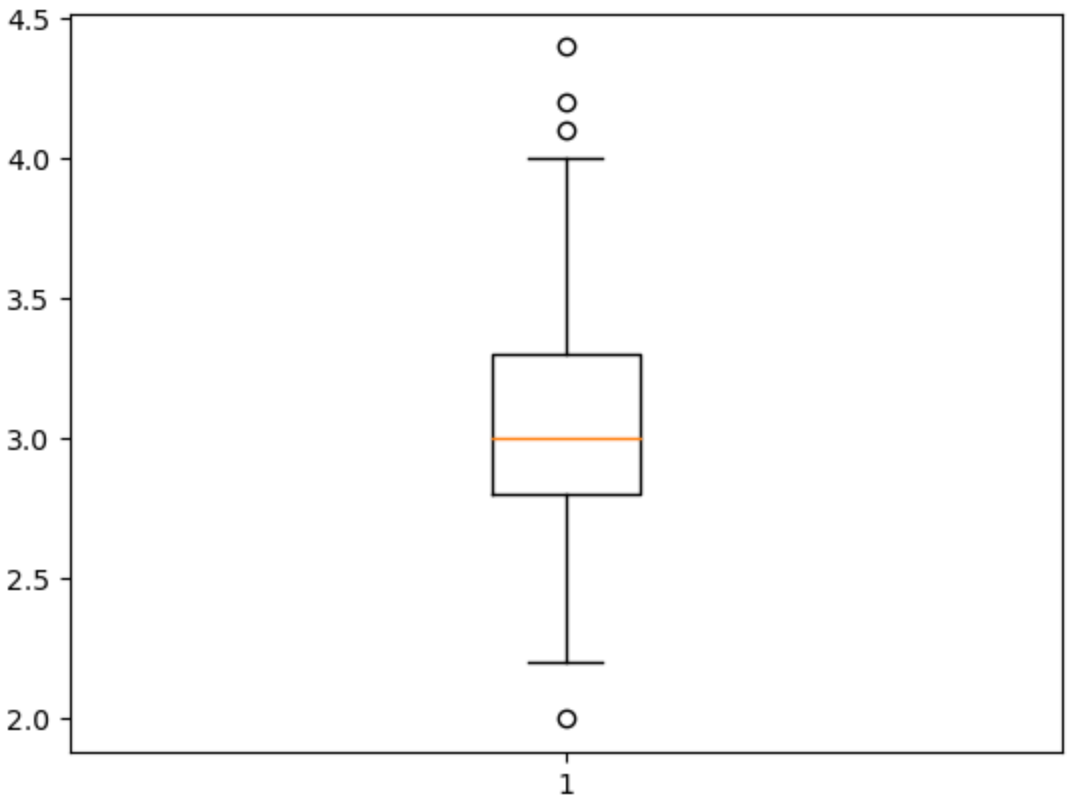
In [8]:
```
plt.boxplot(df['SepalLengthCm'])
```

Out[8]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1fb0c4e64d0>,
  <matplotlib.lines.Line2D at 0x1fb0c4e7210>],
 'caps': [<matplotlib.lines.Line2D at 0x1fb0c4e7ed0>,
  <matplotlib.lines.Line2D at 0x1fb0c4ecbd0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1fb0c475b10>],
 'medians': [<matplotlib.lines.Line2D at 0x1fb0c4ed810>],
 'fliers': [<matplotlib.lines.Line2D at 0x1fb0c4ee290>],
 'means': []}
```



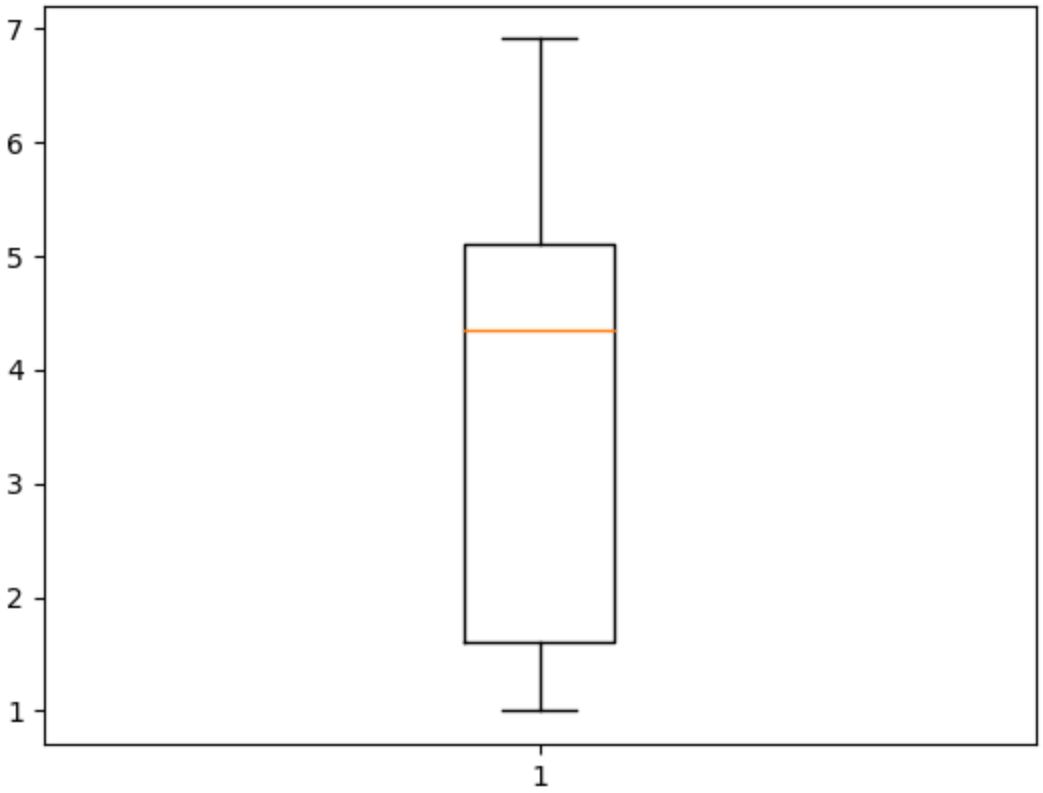
In [10]:
```
plt.boxplot(df['SepalWidthCm'])
```

Out[10]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1fb11b86210>,
  <matplotlib.lines.Line2D at 0x1fb11b85910>],
 'caps': [<matplotlib.lines.Line2D at 0x1fb11b84650>,
  <matplotlib.lines.Line2D at 0x1fb11b87310>],
 'boxes': [<matplotlib.lines.Line2D at 0x1fb11bbc350>],
 'medians': [<matplotlib.lines.Line2D at 0x1fb11b86d90>],
 'fliers': [<matplotlib.lines.Line2D at 0x1fb11c66110>],
 'means': []}
```



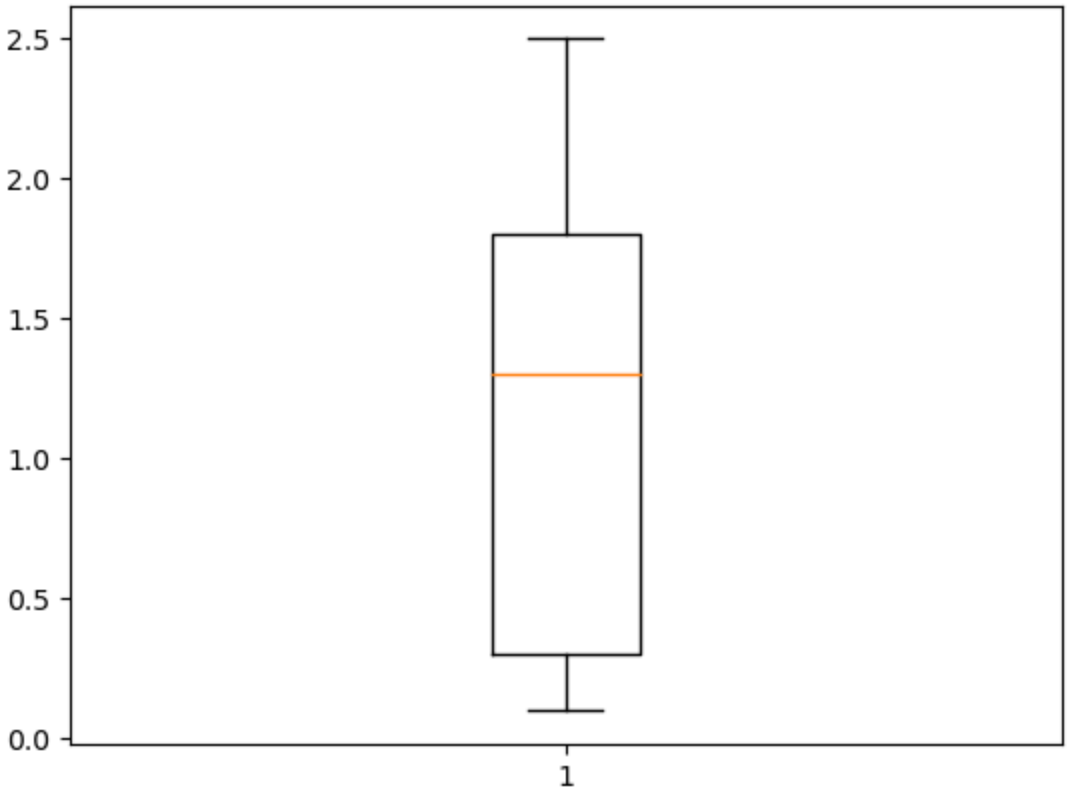
In [11]:
```
plt.boxplot(df['PetalLengthCm'])
```

Out[11]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1fb0c475850>,
  <matplotlib.lines.Line2D at 0x1fb11ccccd0>],
 'caps': [<matplotlib.lines.Line2D at 0x1fb11ccd410>,
  <matplotlib.lines.Line2D at 0x1fb11ccdc90>],
 'boxes': [<matplotlib.lines.Line2D at 0x1fb11cb7990>],
 'medians': [<matplotlib.lines.Line2D at 0x1fb11cce610>],
 'fliers': [<matplotlib.lines.Line2D at 0x1fb0cd06010>],
 'means': []}
```



In [12]:
```
plt.boxplot(df['PetalWidthCm'])
```

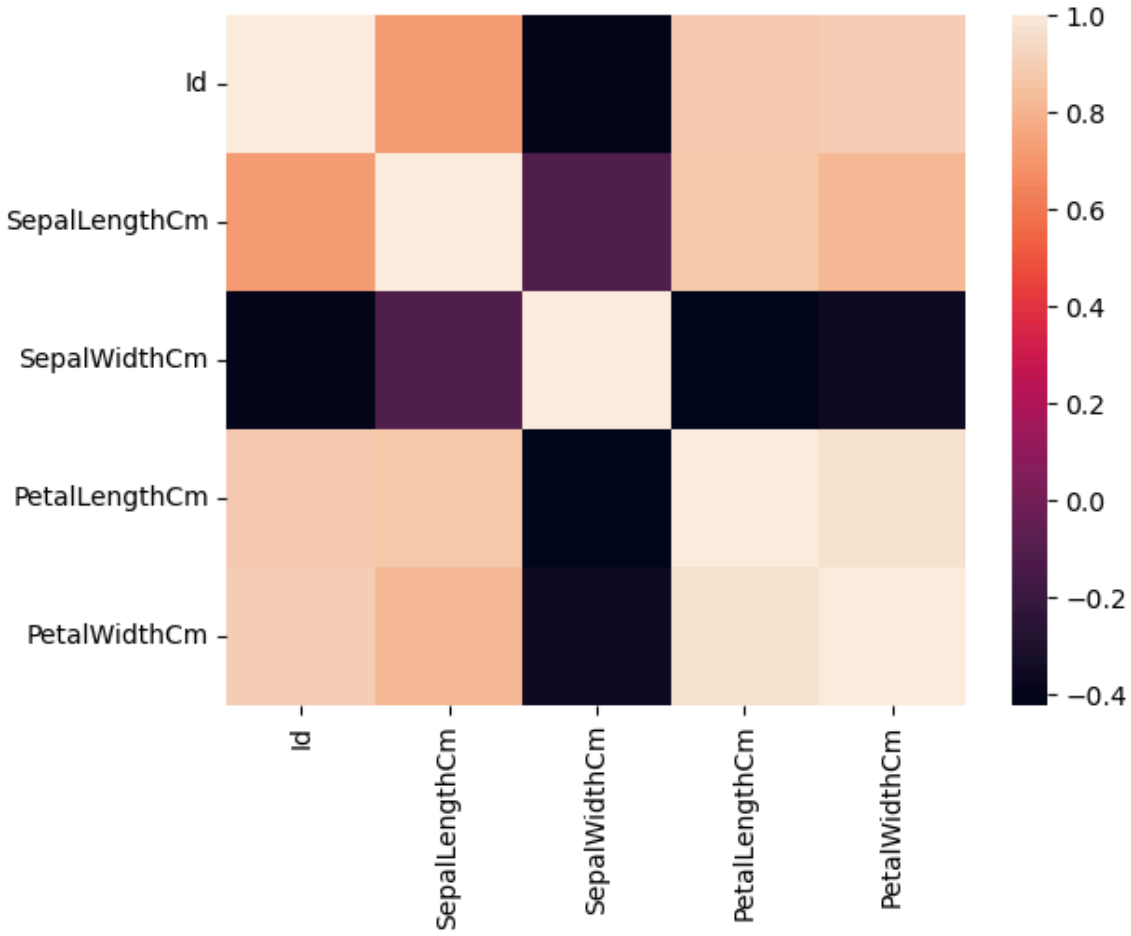
Out[12]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1fb11d36b90>,
  <matplotlib.lines.Line2D at 0x1fb11d37dd0>],
 'caps': [<matplotlib.lines.Line2D at 0x1fb11d44490>,
  <matplotlib.lines.Line2D at 0x1fb11d44cd0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1fb11d36cd0>],
 'medians': [<matplotlib.lines.Line2D at 0x1fb11d45510>],
 'fliers': [<matplotlib.lines.Line2D at 0x1fb11d23410>],
 'means': []}
```



In [13]:
```
sns.heatmap(df.corr())
```

```
C:\Users\Lenovo\AppData\Local\Temp\ipykernel_5096\58359773.py:1: FutureWarn
ing: The default value of numeric_only in DataFrame.corr is deprecated. In
a future version, it will default to False. Select only valid columns or sp
ecify the value of numeric_only to silence this warning.
  sns.heatmap(df.corr())
```

Out[13]: <Axes: >



In [14]:
```
df.drop('Id',axis=1,inplace=True)
```

In [17]:
```
sp={'Iris-setosa':1,'Iris-versicolor':2,'Iris-virginica':3}
```

In [18]:
```
df.Species=[sp[i] for i in df.Species]
```

In [19]:
```
df
```

Out[19]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	1
1	4.9	3.0	1.4	0.2	1
2	4.7	3.2	1.3	0.2	1
3	4.6	3.1	1.5	0.2	1
4	5.0	3.6	1.4	0.2	1
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	3
146	6.3	2.5	5.0	1.9	3
147	6.5	3.0	5.2	2.0	3
148	6.2	3.4	5.4	2.3	3
149	5.9	3.0	5.1	1.8	3

150 rows × 5 columns

In [20]:
```
X = df.iloc[:,0:4]
print(X)
```

```
     SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
0              5.1           3.5            1.4           0.2
1              4.9           3.0            1.4           0.2
2              4.7           3.2            1.3           0.2
3              4.6           3.1            1.5           0.2
4              5.0           3.6            1.4           0.2
..             ...           ...            ...           ...
145            6.7           3.0            5.2           2.3
146            6.3           2.5            5.0           1.9
147            6.5           3.0            5.2           2.0
148            6.2           3.4            5.4           2.3
149            5.9           3.0            5.1           1.8

[150 rows x 4 columns]
```

In [21]:
```
Y =df.iloc[:,4]
print(Y)
```

```
0      1
1      1
2      1
3      1
4      1
      ..
145    3
146    3
147    3
148    3
149    3
Name: Species, Length: 150, dtype: int64
```

In [22]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.20, r
```

In [23]:
```
model=LinearRegression()
```

In [24]:
```
model.fit(X,Y)
```

Out[24]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [25]:
```
model.score(X,Y)
```

Out[25]:
```
0.9304223675331595
```

In [26]:
```
model.coef_
```

Out[26]:
```
array([-0.10974146, -0.04424045,  0.22700138,  0.60989412])
```

In [27]:
```
model.intercept_
```

Out[27]:
```
1.1920839948281436
```

In [28]:
```
Y_pred = model.predict(X_test)
```

In [29]:
```
print( 'MSE : %.2f' % np.mean((Y_pred - Y_test)**2))
```

```
MSE : 0.03
```

In [ ]:
```

```