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

In [2]:
```
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)
```

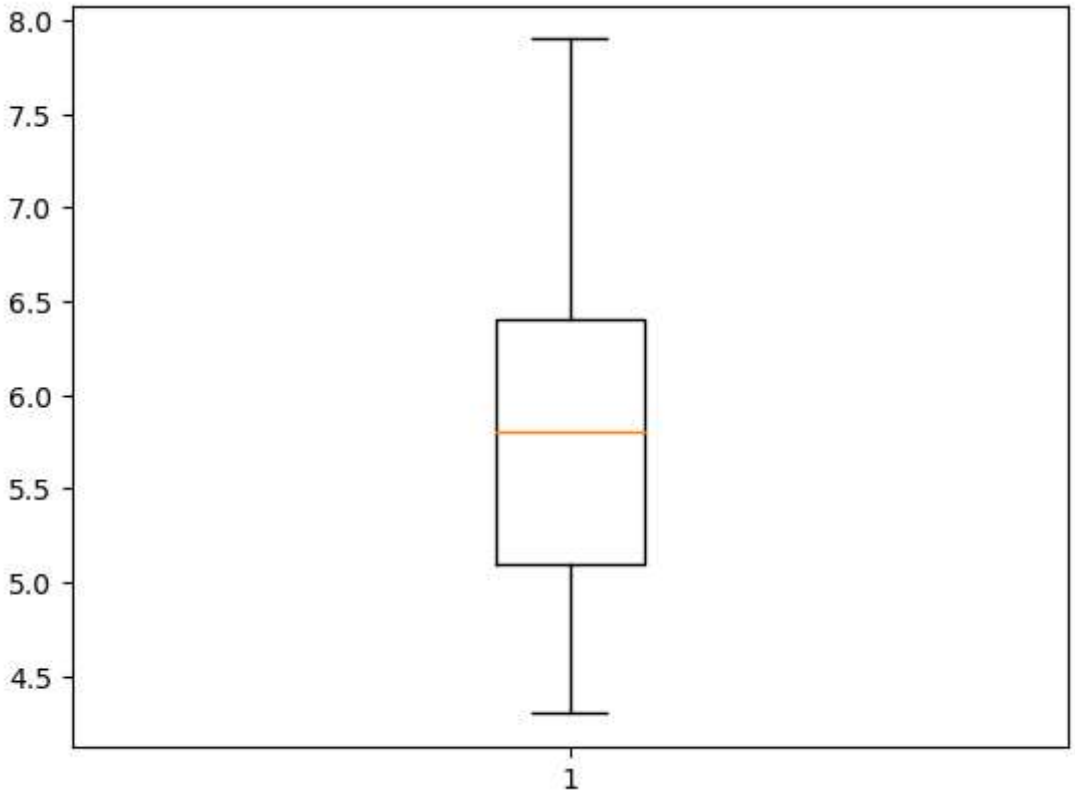
In [6]:
```
data=df.groupby('Species')
```

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

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

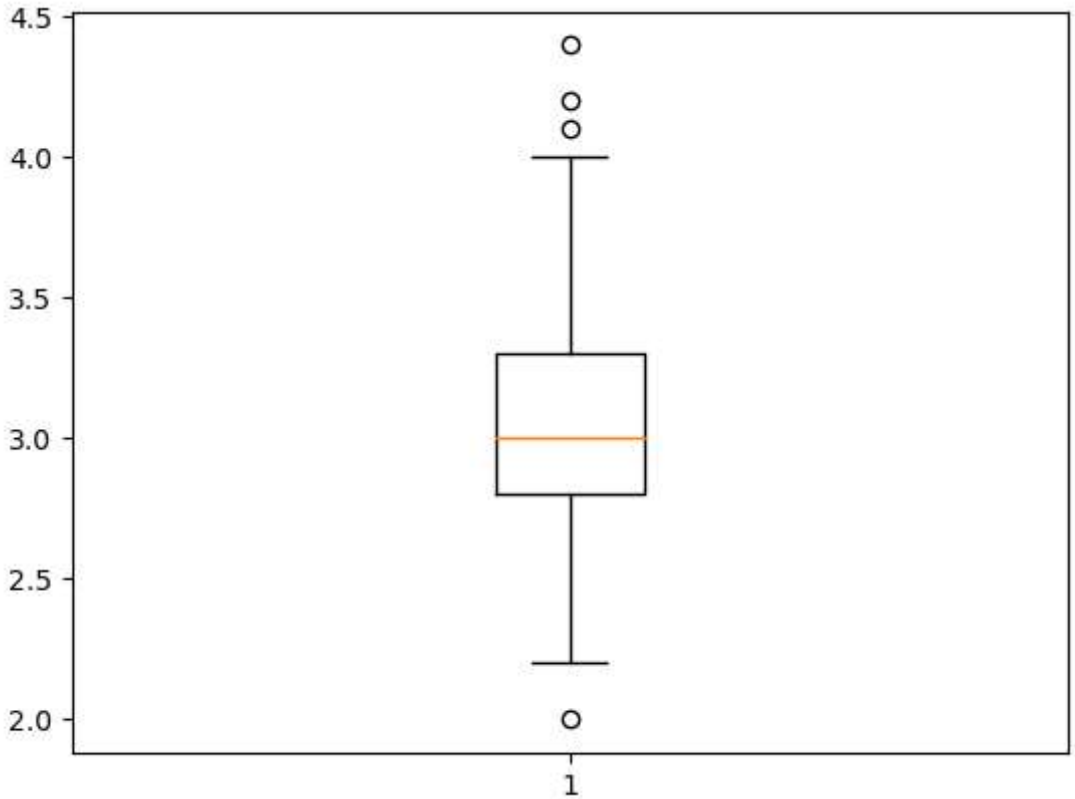
In [8]:
```
#2. visualizing the dataset
plt.boxplot(df['SepalLengthCm'])
```

Out[8]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x2389ad93210>,
  <matplotlib.lines.Line2D at 0x2389ba11d10>],
 'caps': [<matplotlib.lines.Line2D at 0x2389b414d50>,
  <matplotlib.lines.Line2D at 0x2389ba2d790>],
 'boxes': [<matplotlib.lines.Line2D at 0x2389ba118d0>],
 'medians': [<matplotlib.lines.Line2D at 0x2389ba2e390>],
 'fliers': [<matplotlib.lines.Line2D at 0x2389b2714d0>],
 'means': []}
```



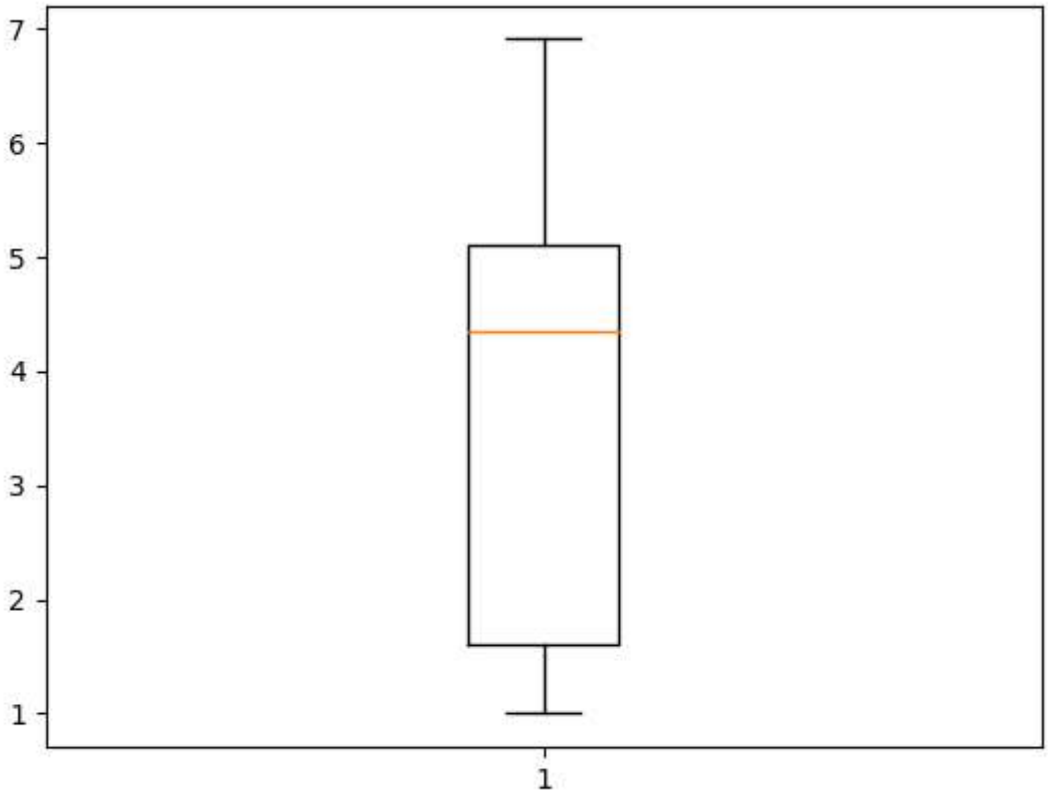
In [9]:
```
plt.boxplot(df['SepalWidthCm'])
```

Out[9]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x2389baf8e50>,
  <matplotlib.lines.Line2D at 0x2389baf9a90>],
 'caps': [<matplotlib.lines.Line2D at 0x2389bafa650>,
  <matplotlib.lines.Line2D at 0x2389bafb0d0>],
 'boxes': [<matplotlib.lines.Line2D at 0x2389baf8290>],
 'medians': [<matplotlib.lines.Line2D at 0x2389bafbc90>],
 'fliers': [<matplotlib.lines.Line2D at 0x2389bac20d0>],
 'means': []}
```



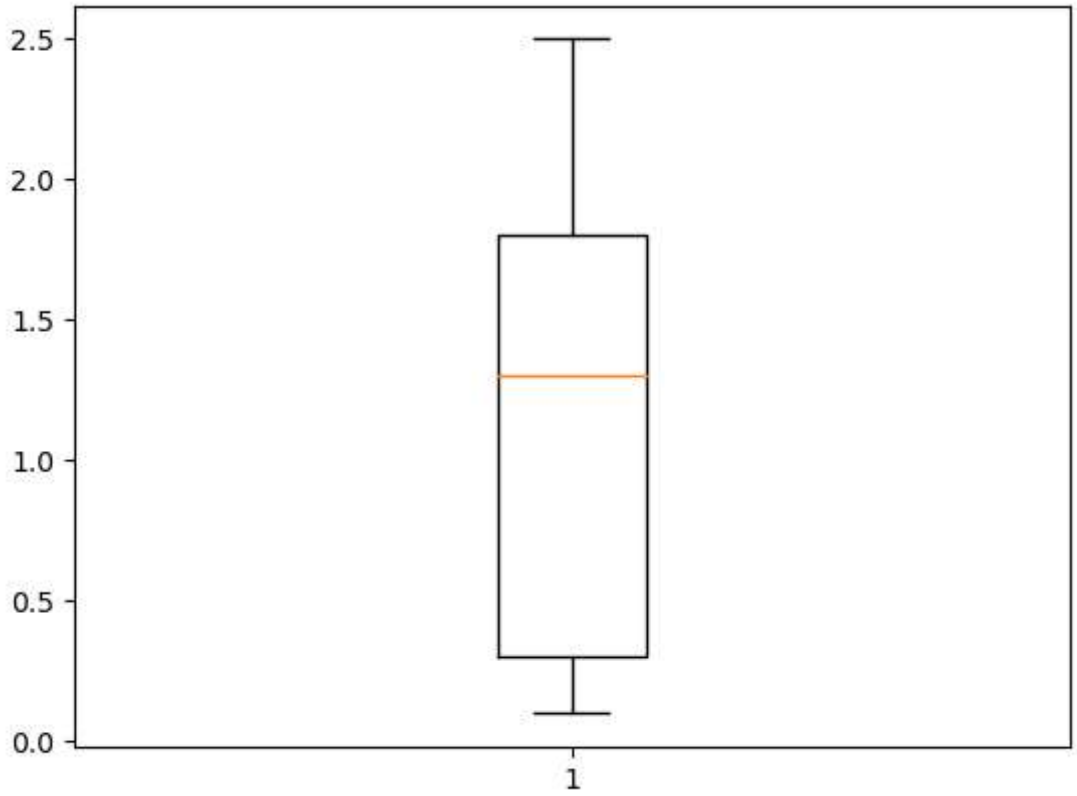
In [10]:
```
plt.boxplot(df['PetalLengthCm'])
```

Out[10]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x2389bb66290>,
  <matplotlib.lines.Line2D at 0x2389bb67010>],
 'caps': [<matplotlib.lines.Line2D at 0x2389bb67b50>,
  <matplotlib.lines.Line2D at 0x2389bb6c710>],
 'boxes': [<matplotlib.lines.Line2D at 0x2389bb65610>],
 'medians': [<matplotlib.lines.Line2D at 0x2389bb6d190>],
 'fliers': [<matplotlib.lines.Line2D at 0x2389bb1dbd0>],
 'means': []}
```



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

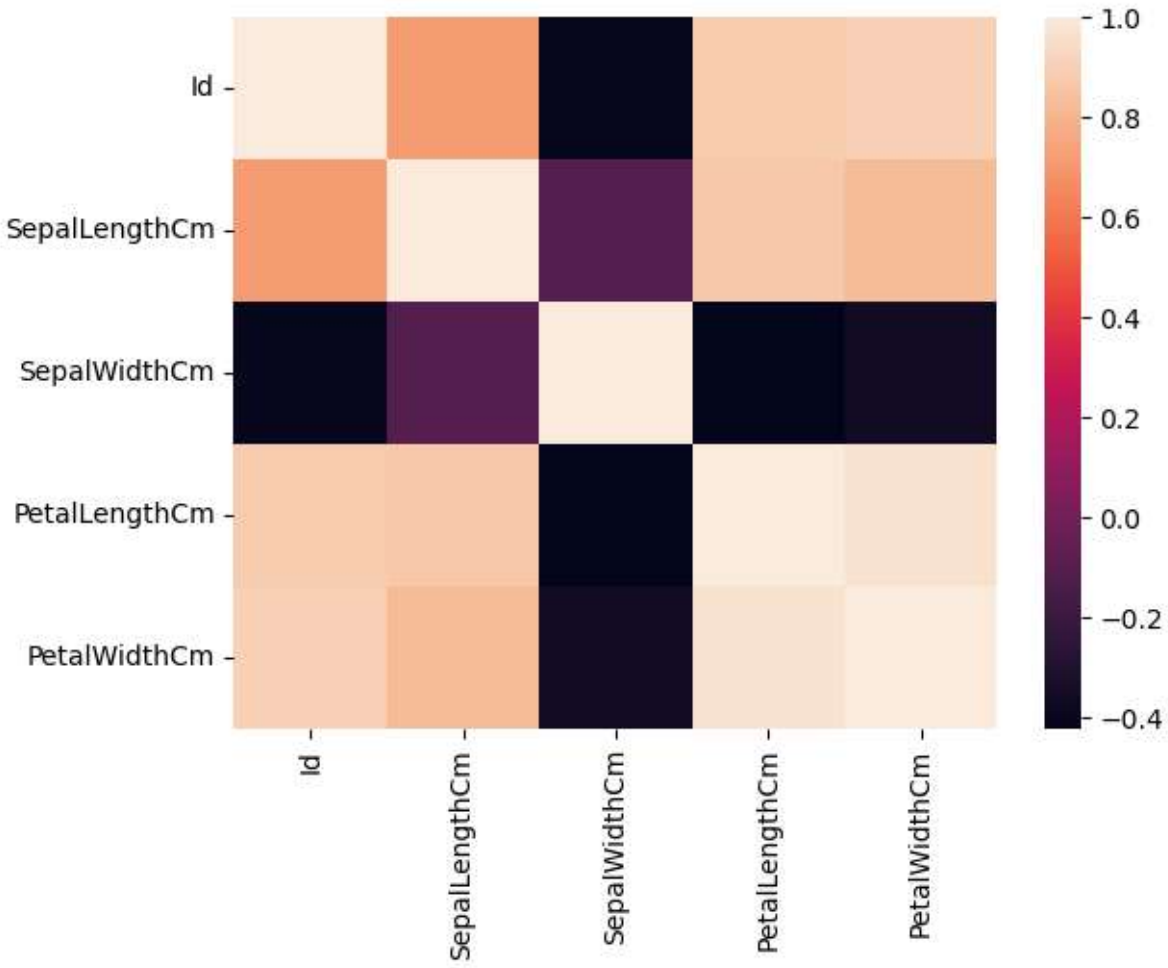
Out[11]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x2389bbcc5d0>,
  <matplotlib.lines.Line2D at 0x2389bbcd0d0>],
 'caps': [<matplotlib.lines.Line2D at 0x2389bbcdc10>,
  <matplotlib.lines.Line2D at 0x2389bbcd710>],
 'boxes': [<matplotlib.lines.Line2D at 0x2389bbc38d0>],
 'medians': [<matplotlib.lines.Line2D at 0x2389bbcf110>],
 'fliers': [<matplotlib.lines.Line2D at 0x2389bbc3710>],
 'means': []}
```



In [12]:
```
sns.heatmap(df.corr())
```

```
C:\Users\91772\AppData\Local\Temp\ipykernel_22852\58359773.py:1: FutureWarning: Th
e default value of numeric_only in DataFrame.corr is deprecated. In a future versi
on, it will default to False. Select only valid columns or specify the value of nu
meric_only to silence this warning.
  sns.heatmap(df.corr())
```

Out[12]:
```
<Axes: >
```



In [13]:
```
#3. Data Preparation
df.drop('Id',axis=1,inplace=True)
```

In [14]:
```
sp={'Iris-setosa':1,'Iris-versicolor':2,'Iris-virginica':3}
```

In [15]:
```
df.Species=[sp[i] for i in df.Species]
```

In [16]:
```
df
```

Out[16]:

	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 [17]:
```
X=df.iloc[:,0:4]
```

In [19]:
```
X
```

Out[19]:

	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 × 4 columns

In [20]:
```
y=df.iloc[:,4]
```

In [21]:
```
y
```

Out[21]:
```
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=0.33,random_state=42)
```

In [23]:
```
#Traning Model
model=LinearRegression()
```

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

Out[24]:
```
LinearRegression
LinearRegression()
```

In [25]:
```
model.score(X,y) #coef of prediction
```

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]:
```
1.192083994828144
```

In [28]:
```
#Making Prediction
y_pred=model.predict(X_test)
```

In [29]:
```
#Model Evolution
print("Mean squared error: %.2f" % np.mean((y_pred - y_test) ** 2))
```

```
Mean squared error: 0.04
```

In [ ]: