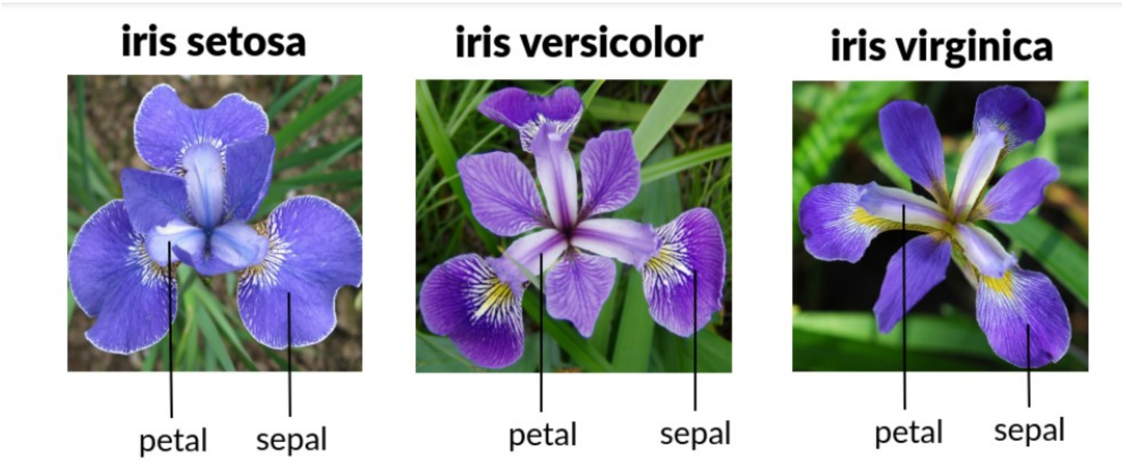
Sakshi Kharat

Oasis Infobyte (Data Science) - Task-1

Iiris Flower Classification



Importing the libraries

```
import pandas as pd

import matplotlib.pyplot as plt
```

Loading the Data

```
from google.colab import files


uploaded = files.upload()
```

```
<IPython.core.display.HTML object>

Saving Iris.csv to Iris (1).csv
```

```
import numpy as np

df = pd.read_csv('Iris.csv')
```

Viewing the Dataset

```
df
```

```
     Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
0     1            5.1           3.5            1.4           0.2
```

```
1      2            4.9           3.0            1.4           0.2
2      3            4.7           3.2            1.3           0.2
3      4            4.6           3.1            1.5           0.2
4      5            5.0           3.6            1.4           0.2
..   ...            ...           ...            ...           ...
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            Species
0       Iris-setosa
1       Iris-setosa
2       Iris-setosa
3       Iris-setosa
4       Iris-setosa
..              ...
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica

[150 rows x 6 columns]
```

Displaying the Dataset

```
df.info
```

```
<bound method DataFrame.info of       Id  SepalLengthCm  SepalWidthCm
PetalLengthCm  PetalWidthCm  \
0      1            5.1           3.5            1.4           0.2
1      2            4.9           3.0            1.4           0.2
2      3            4.7           3.2            1.3           0.2
3      4            4.6           3.1            1.5           0.2
4      5            5.0           3.6            1.4           0.2
..   ...            ...           ...            ...           ...
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            Species
0       Iris-setosa
1       Iris-setosa
2       Iris-setosa
3       Iris-setosa
4       Iris-setosa
```

```
..              ...
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica

[150 rows x 6 columns]>
```

Modifying the dataset by removing any Missing Values using fillna() method

```
df.isnull().sum()
```

```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

The Values are 0 meaning it has no Null Values all over the dataset

Viewing the Columns in the dataset

```
df.columns
```

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm',
'PetalWidthCm',
       'Species'],
      dtype='object')
```

```
df.describe
```

```
<bound method NDFrame.describe of       Id  SepalLengthCm
SepalWidthCm  PetalLengthCm  PetalWidthCm  \
0      1            5.1           3.5            1.4           0.2
1      2            4.9           3.0            1.4           0.2
2      3            4.7           3.2            1.3           0.2
3      4            4.6           3.1            1.5           0.2
4      5            5.0           3.6            1.4           0.2
..   ...            ...           ...            ...           ...
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            Species
0       Iris-setosa
1       Iris-setosa
2       Iris-setosa
```

```
3       Iris-setosa
4       Iris-setosa
..              ...
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica

[150 rows x 6 columns]>
```

```
df.head(10)
```

```
   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
5   6            5.4           3.9            1.7           0.4  Iris-
setosa
6   7            4.6           3.4            1.4           0.3  Iris-
setosa
7   8            5.0           3.4            1.5           0.2  Iris-
setosa
8   9            4.4           2.9            1.4           0.2  Iris-
setosa
9  10            4.9           3.1            1.5           0.1  Iris-
setosa
```

```
df.shape
```

```
(150, 6)
```

```
print(df)
```

```
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
0      1            5.1           3.5            1.4           0.2   
1      2            4.9           3.0            1.4           0.2   
2      3            4.7           3.2            1.3           0.2   
3      4            4.6           3.1            1.5           0.2   
4      5            5.0           3.6            1.4           0.2   
..   ...            ...           ...            ...           ...   
145  146            6.7           3.0            5.2           2.3   
146  147            6.3           2.5            5.0           1.9   
```

```
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            Species
0       Iris-setosa
1       Iris-setosa
2       Iris-setosa
3       Iris-setosa
4       Iris-setosa
..              ...
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica

[150 rows x 6 columns]
```

```
print(df[10:21])
```

```
    Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
10  11            5.4           3.7            1.5           0.2
Iris-setosa
11  12            4.8           3.4            1.6           0.2
Iris-setosa
12  13            4.8           3.0            1.4           0.1
Iris-setosa
13  14            4.3           3.0            1.1           0.1
Iris-setosa
14  15            5.8           4.0            1.2           0.2
Iris-setosa
15  16            5.7           4.4            1.5           0.4
Iris-setosa
16  17            5.4           3.9            1.3           0.4
Iris-setosa
17  18            5.1           3.5            1.4           0.3
Iris-setosa
18  19            5.7           3.8            1.7           0.3
Iris-setosa
19  20            5.1           3.8            1.5           0.3
Iris-setosa
20  21            5.4           3.4            1.7           0.2
Iris-setosa
```

```
sliced_data=df[10:21]
print(sliced_data)
```

```
    Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
```

```
10  11            5.4           3.7            1.5           0.2
Iris-setosa
11  12            4.8           3.4            1.6           0.2
Iris-setosa
12  13            4.8           3.0            1.4           0.1
Iris-setosa
13  14            4.3           3.0            1.1           0.1
Iris-setosa
14  15            5.8           4.0            1.2           0.2
Iris-setosa
15  16            5.7           4.4            1.5           0.4
Iris-setosa
16  17            5.4           3.9            1.3           0.4
Iris-setosa
17  18            5.1           3.5            1.4           0.3
Iris-setosa
18  19            5.7           3.8            1.7           0.3
Iris-setosa
19  20            5.1           3.8            1.5           0.3
Iris-setosa
20  21            5.4           3.4            1.7           0.2
Iris-setosa
```

```
df.iloc[5]
```

```
Id                         6
SepalLengthCm            5.4
SepalWidthCm             3.9
PetalLengthCm            1.7
PetalWidthCm             0.4
Species          Iris-setosa
Name: 5, dtype: object
```

```
df.loc[df["Species"] == "Iris-setosa"]
```

```
    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
5    6            5.4           3.9            1.7           0.4
Iris-setosa
6    7            4.6           3.4            1.4           0.3
Iris-setosa
```

```
7    8            5.0           3.4            1.5           0.2
Iris-setosa
8    9            4.4           2.9            1.4           0.2
Iris-setosa
9   10            4.9           3.1            1.5           0.1
Iris-setosa
10  11            5.4           3.7            1.5           0.2
Iris-setosa
11  12            4.8           3.4            1.6           0.2
Iris-setosa
12  13            4.8           3.0            1.4           0.1
Iris-setosa
13  14            4.3           3.0            1.1           0.1
Iris-setosa
14  15            5.8           4.0            1.2           0.2
Iris-setosa
15  16            5.7           4.4            1.5           0.4
Iris-setosa
16  17            5.4           3.9            1.3           0.4
Iris-setosa
17  18            5.1           3.5            1.4           0.3
Iris-setosa
18  19            5.7           3.8            1.7           0.3
Iris-setosa
19  20            5.1           3.8            1.5           0.3
Iris-setosa
20  21            5.4           3.4            1.7           0.2
Iris-setosa
21  22            5.1           3.7            1.5           0.4
Iris-setosa
22  23            4.6           3.6            1.0           0.2
Iris-setosa
23  24            5.1           3.3            1.7           0.5
Iris-setosa
24  25            4.8           3.4            1.9           0.2
Iris-setosa
25  26            5.0           3.0            1.6           0.2
Iris-setosa
26  27            5.0           3.4            1.6           0.4
Iris-setosa
27  28            5.2           3.5            1.5           0.2
Iris-setosa
28  29            5.2           3.4            1.4           0.2
Iris-setosa
29  30            4.7           3.2            1.6           0.2
Iris-setosa
30  31            4.8           3.1            1.6           0.2
Iris-setosa
31  32            5.4           3.4            1.5           0.4
Iris-setosa
```

```
32  33            5.2           4.1            1.5           0.1
Iris-setosa
33  34            5.5           4.2            1.4           0.2
Iris-setosa
34  35            4.9           3.1            1.5           0.1
Iris-setosa
35  36            5.0           3.2            1.2           0.2
Iris-setosa
36  37            5.5           3.5            1.3           0.2
Iris-setosa
37  38            4.9           3.1            1.5           0.1
Iris-setosa
38  39            4.4           3.0            1.3           0.2
Iris-setosa
39  40            5.1           3.4            1.5           0.2
Iris-setosa
40  41            5.0           3.5            1.3           0.3
Iris-setosa
41  42            4.5           2.3            1.3           0.3
Iris-setosa
42  43            4.4           3.2            1.3           0.2
Iris-setosa
43  44            5.0           3.5            1.6           0.6
Iris-setosa
44  45            5.1           3.8            1.9           0.4
Iris-setosa
45  46            4.8           3.0            1.4           0.3
Iris-setosa
46  47            5.1           3.8            1.6           0.2
Iris-setosa
47  48            4.6           3.2            1.4           0.2
Iris-setosa
48  49            5.3           3.7            1.5           0.2
Iris-setosa
49  50            5.0           3.3            1.4           0.2
Iris-setosa
```

```
df["Species"].value_counts()
```

```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

```
sum = df["SepalLengthCm"].sum()
```

```
print(sum)
```

```
876.5
```

```
mean = df["SepalLengthCm"].mean()
```

```
print(mean)
```

```
5.843333333333334
```

```
min = df["SepalLengthCm"].min()
```

```
print(min)
```

```
4.3
```

```
max = df["SepalLengthCm"].max()
```

Preprocessing The Data

```
df.isnull()
```

```
        Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
Species
0    False          False         False          False         False
False
1    False          False         False          False         False
False
2    False          False         False          False         False
False
3    False          False         False          False         False
False
4    False          False         False          False         False
False
..     ...            ...           ...            ...           ...
...
145  False          False         False          False         False
False
146  False          False         False          False         False
False
147  False          False         False          False         False
False
148  False          False         False          False         False
False
149  False          False         False          False         False
False

[150 rows x 6 columns]
```

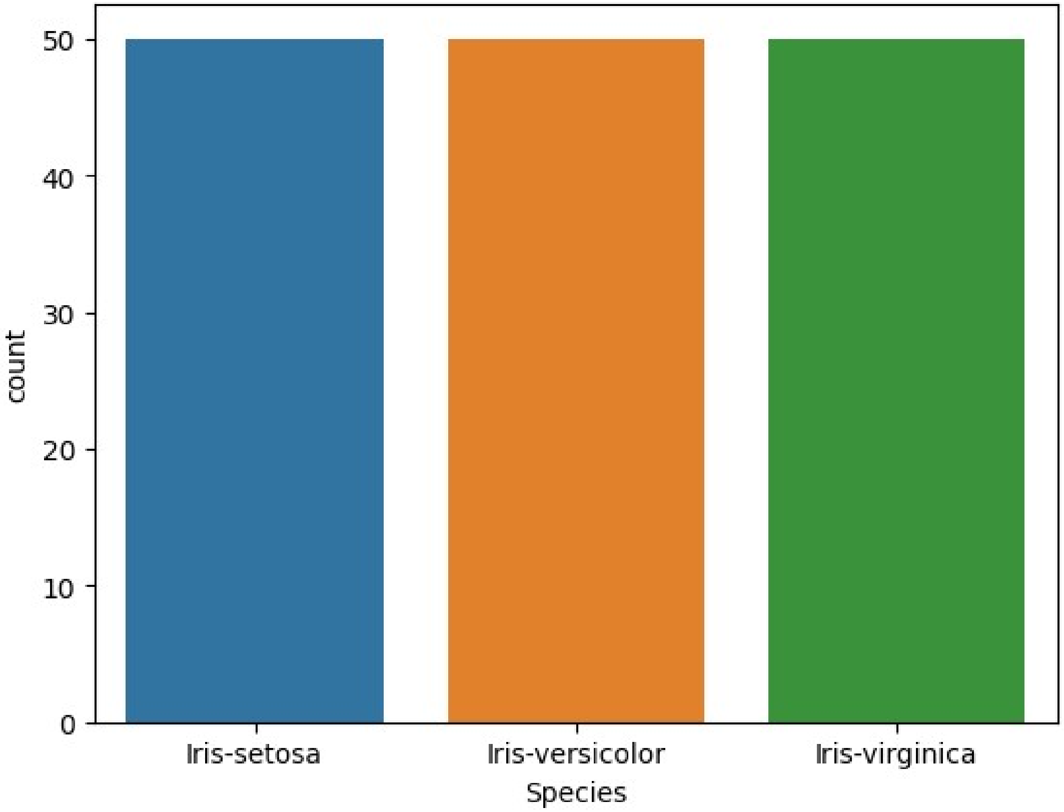
Data Visualization

```
import seaborn as sns
```

```
iris = sns.load_dataset("iris")
```

Count Plot

```
sns.countplot(x='Species', data=df, )
plt.show()
```

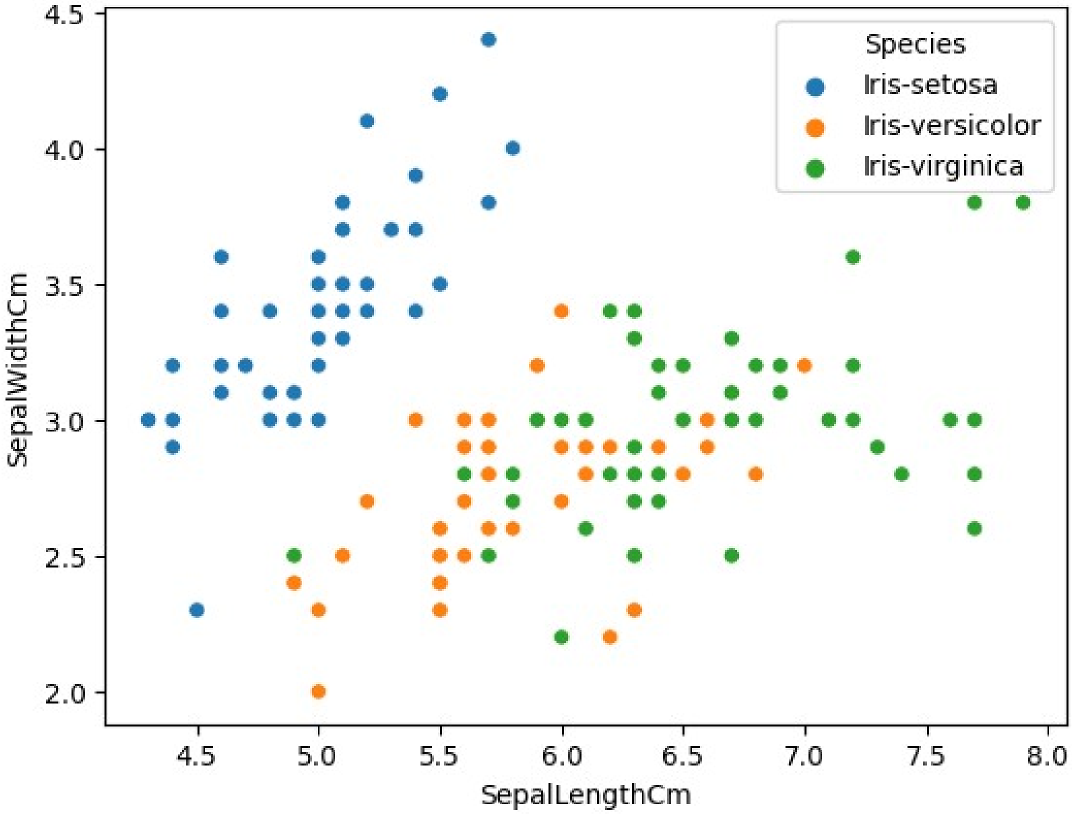


According to the Plot the total rows are 150 and on which 50 are Iris-Setosa, 50 are Iris-Versicolor and 50 are Iris-Virginica

Scatter Plot

```
sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm',
                hue='Species', data=df, )
```

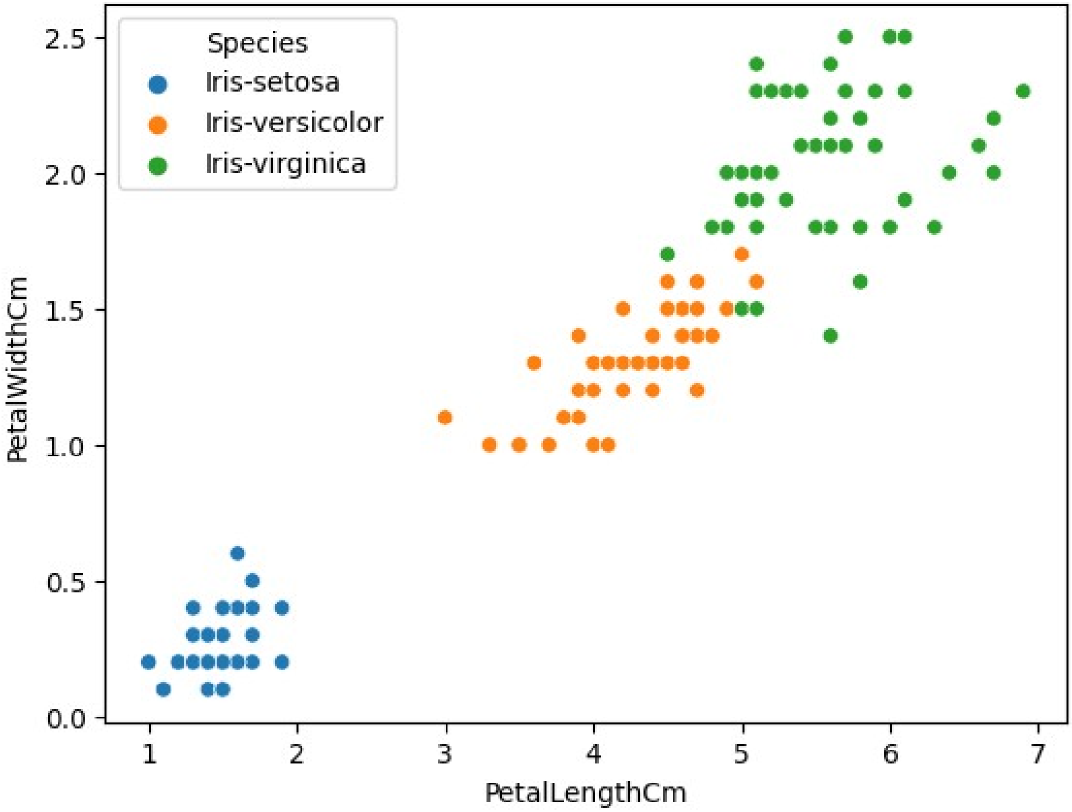
```
<Axes: xlabel='SepalLengthCm', ylabel='SepalWidthCm'>
```



According to the above plot 1) Iris-Setosa has smaller Sepal Length and larger Sepal Widths 2) Iris-Versicolor is the medium range of Sepal Length and Sepal Width 3) Iris-Virginica has larger sepal lengths and larger sepal width

```
sns.scatterplot(x='PetalLengthCm', y='PetalWidthCm',
                hue='Species', data=df, )
```

```
<Axes: xlabel='PetalLengthCm', ylabel='PetalWidthCm'>
```

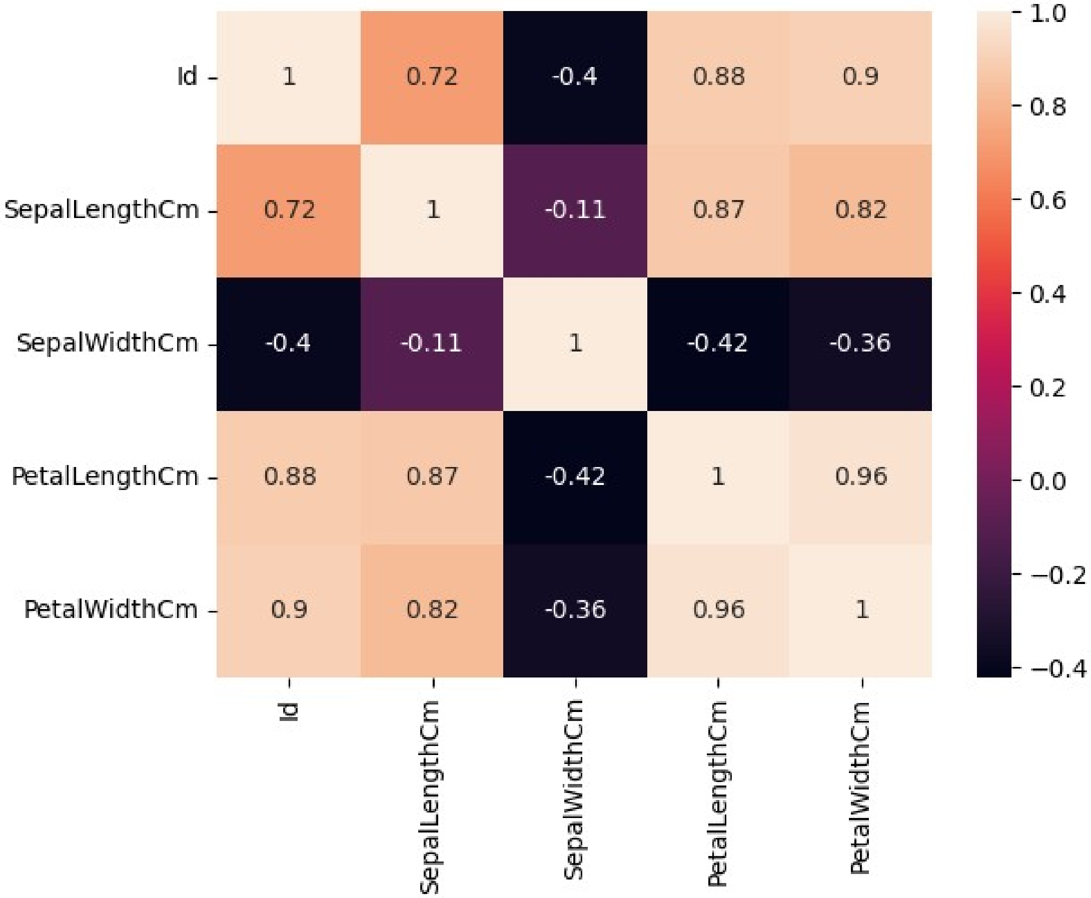


```
# This is formatted as code
```

This plot explains that Iris Setosa has smaller Petal Length and Smaller Petal width, while Iris- Versicolor lies in the middle and Iris- Virginica has Larger Petal Widtth and Smaller Petal Length

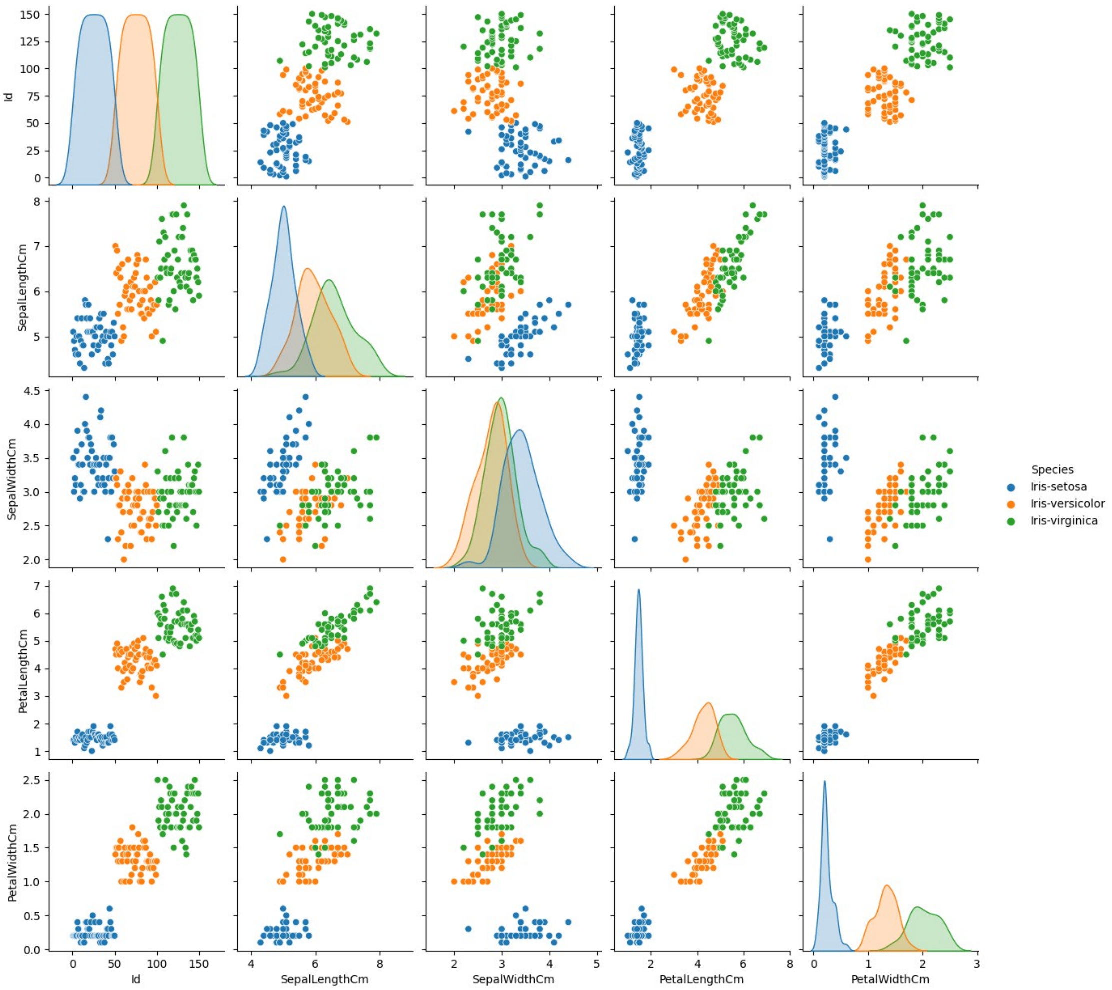
Heat Map

```
import seaborn as sns
import matplotlib.pyplot as plt


sns.heatmap(df.corr(method='pearson'),
            annot = True);

plt.show()
```



Pair Plot

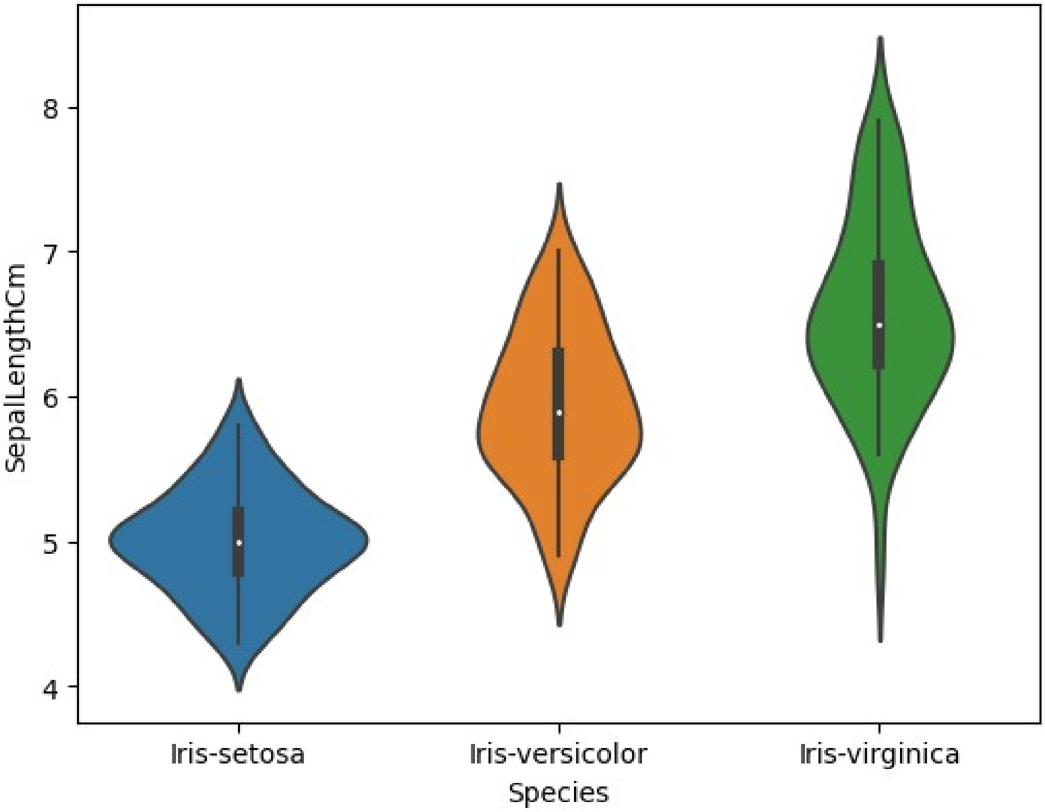
```
g = sns.pairplot(df,hue="Species")
```



Violin Plot

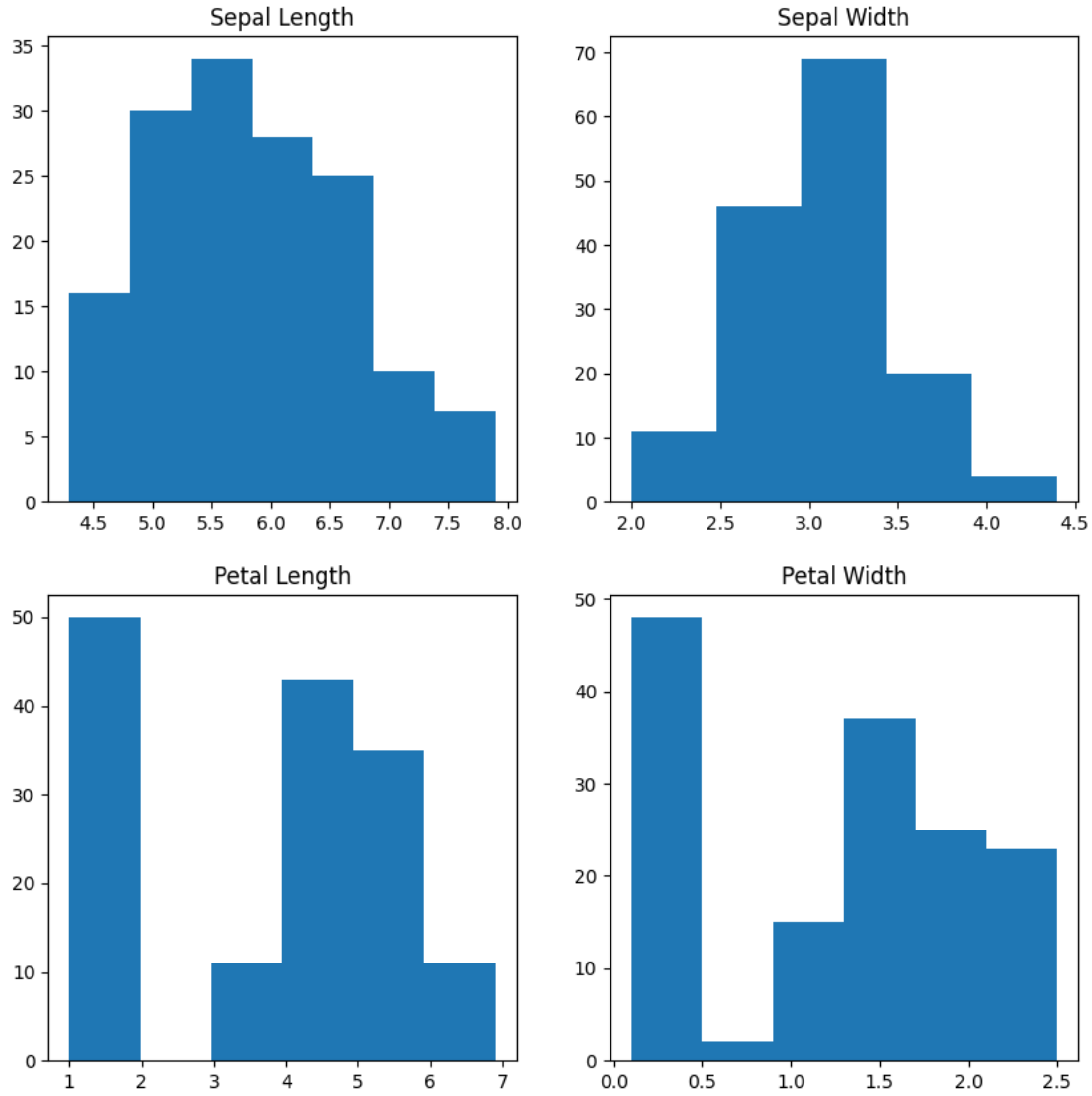
```
sns.violinplot(data=df, x='Species', y='SepalLengthCm')
```

```
<Axes: xlabel='Species', ylabel='SepalLengthCm'>
```



Histogram

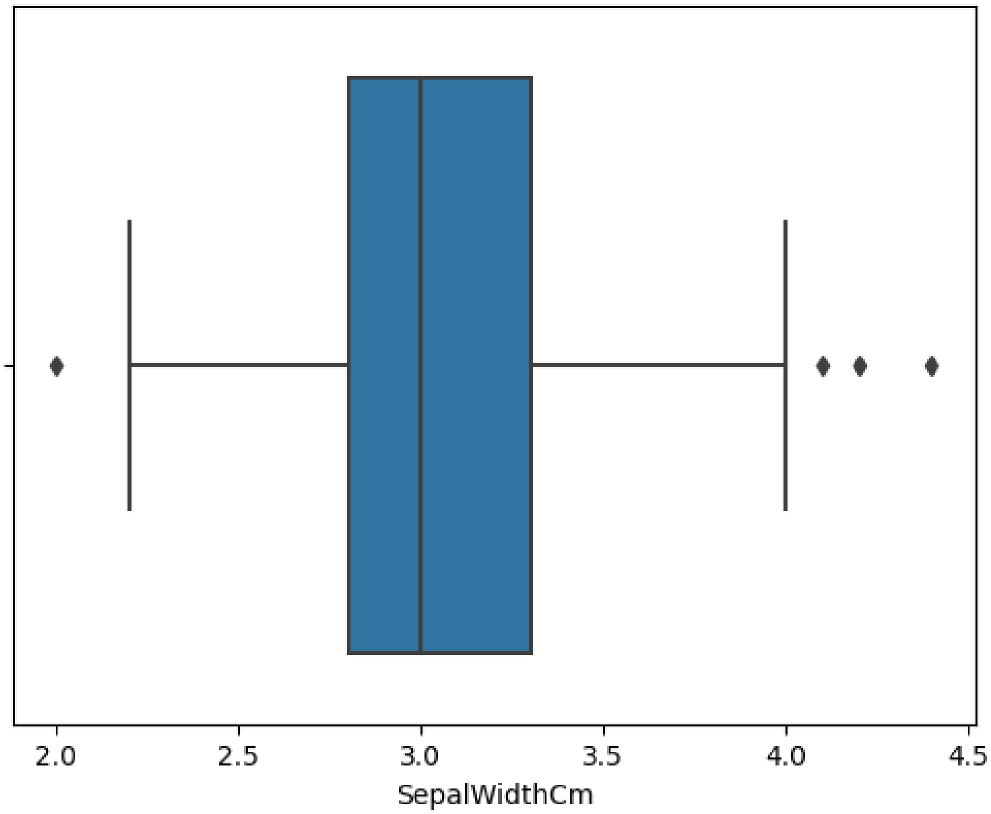
```
import seaborn as sns
import matplotlib.pyplot as plt


fig, axes = plt.subplots(2, 2, figsize=(10,10))

axes[0,0].set_title("Sepal Length")
axes[0,0].hist(df['SepalLengthCm'], bins=7)

axes[0,1].set_title("Sepal Width")
axes[0,1].hist(df['SepalWidthCm'], bins=5);

axes[1,0].set_title("Petal Length")
axes[1,0].hist(df['PetalLengthCm'], bins=6);

axes[1,1].set_title("Petal Width")
axes[1,1].hist(df['PetalWidthCm'], bins=6);
```



Box Plot

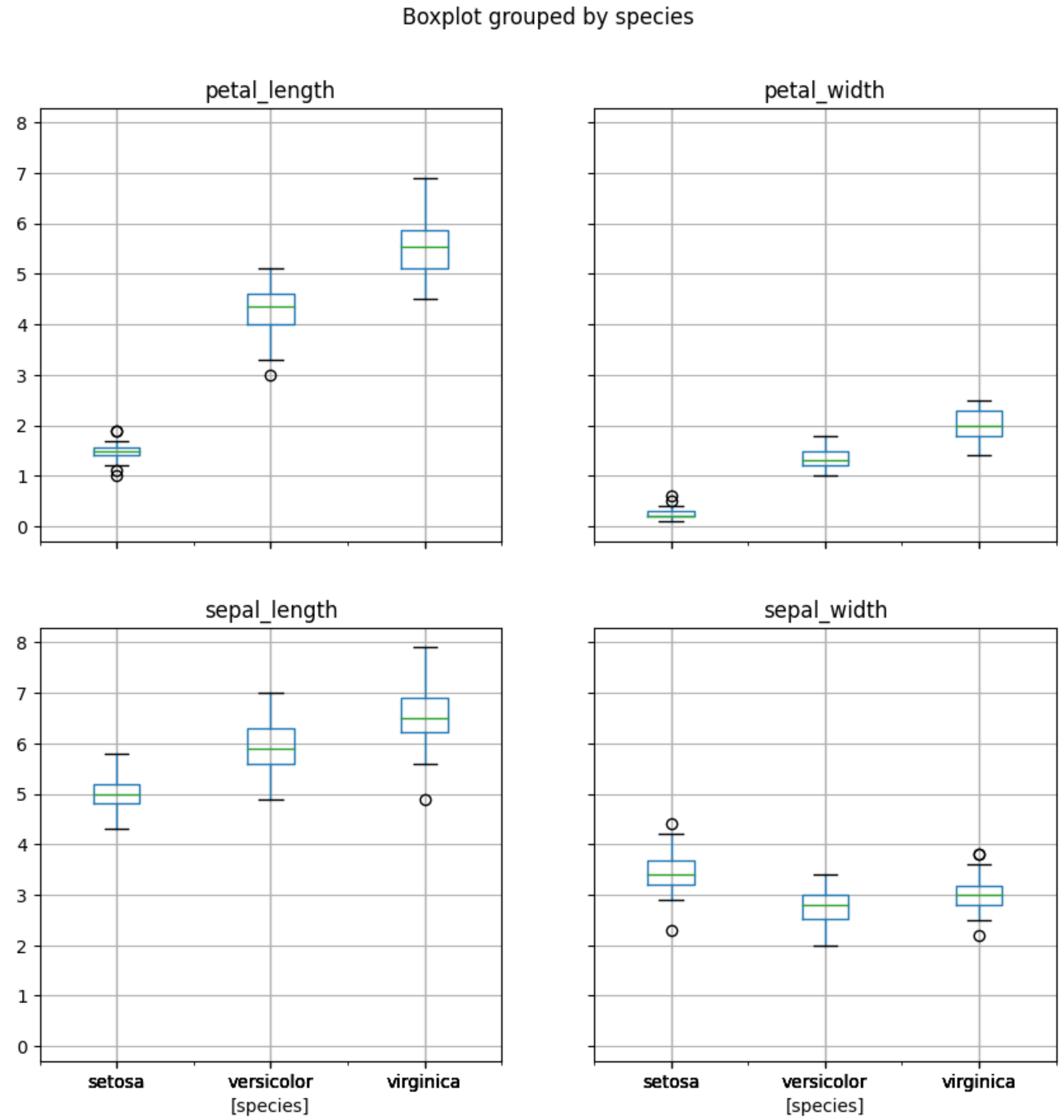
```
sns.boxplot(x='SepalWidthCm', data=df)
```

```
<Axes: xlabel='SepalWidthCm'>
```

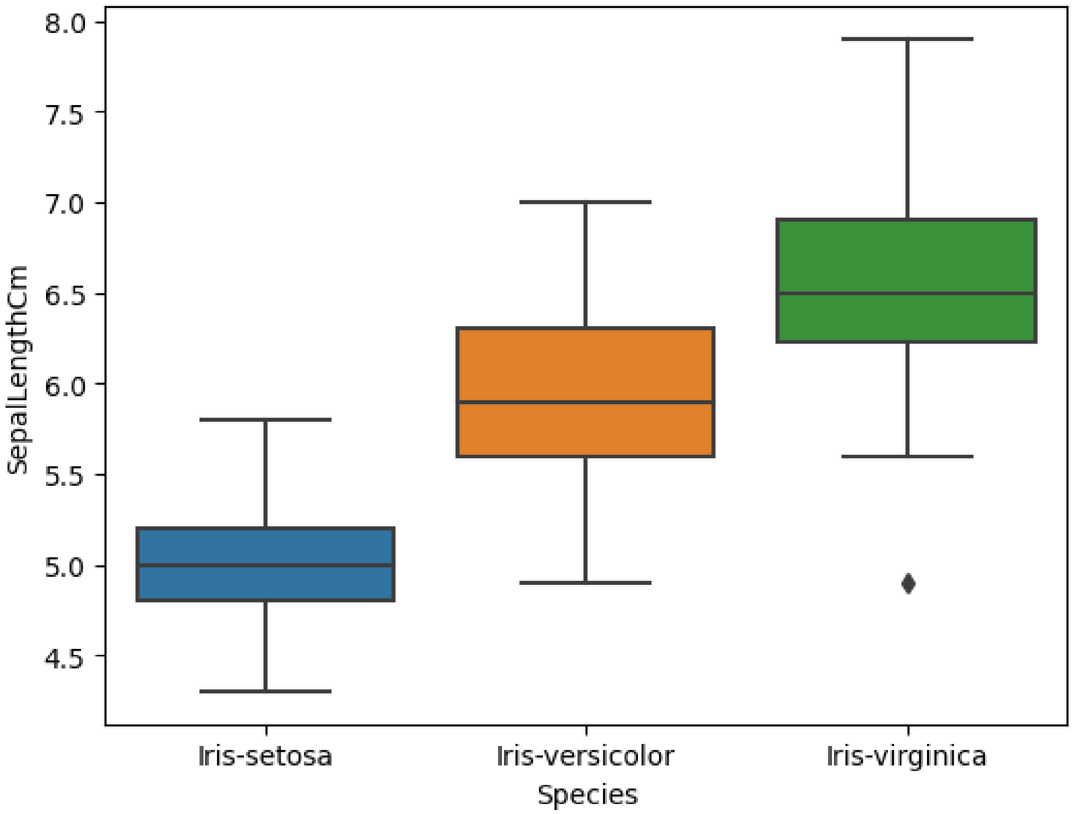


The values above 4 and below 2 are acting as outliers.

```
iris.boxplot(by = 'species', figsize=(10,10))
plt.show()
```

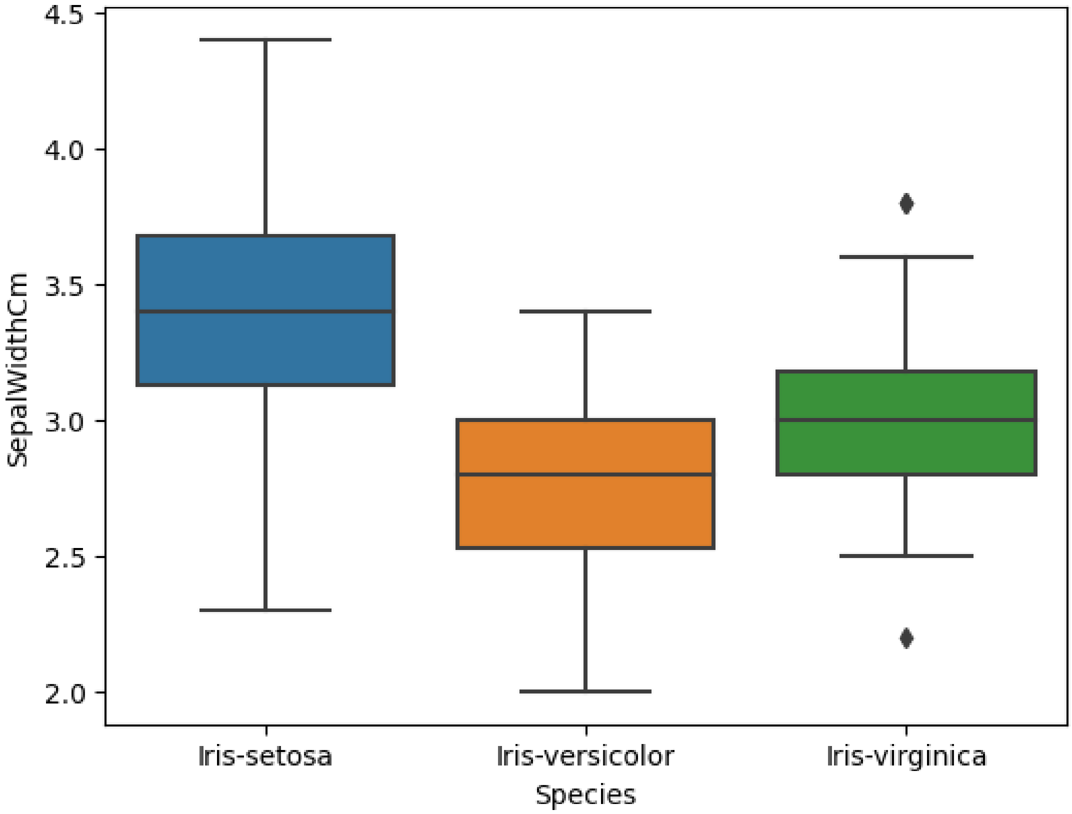


```
def graph(y):
    sns.boxplot(x="Species", y=y, data=df)
    plt.figure(figsize=(10,10))
graph('SepalLengthCm')

plt.show()
```



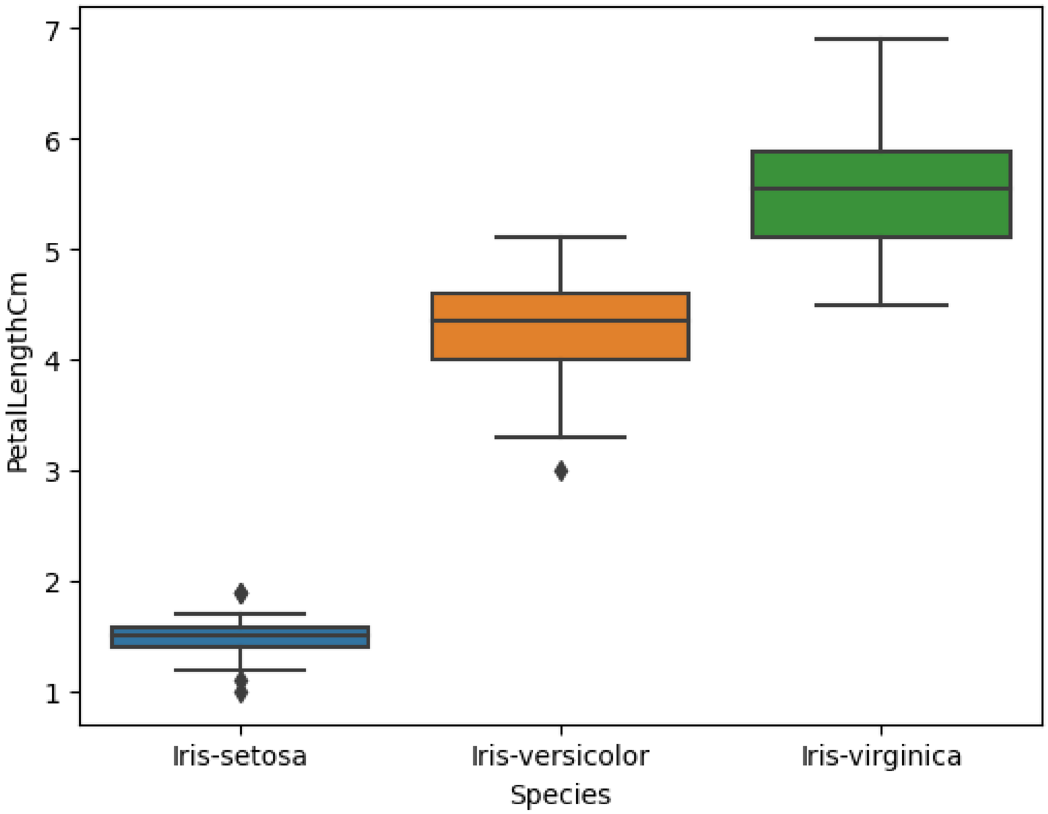
```
<Figure size 1000x1000 with 0 Axes>
```

```python
def graph(y):
    sns.boxplot(x="Species", y=y, data=df)
    plt.figure(figsize=(10,10))
graph('SepalWidthCm')


plt.show()
```



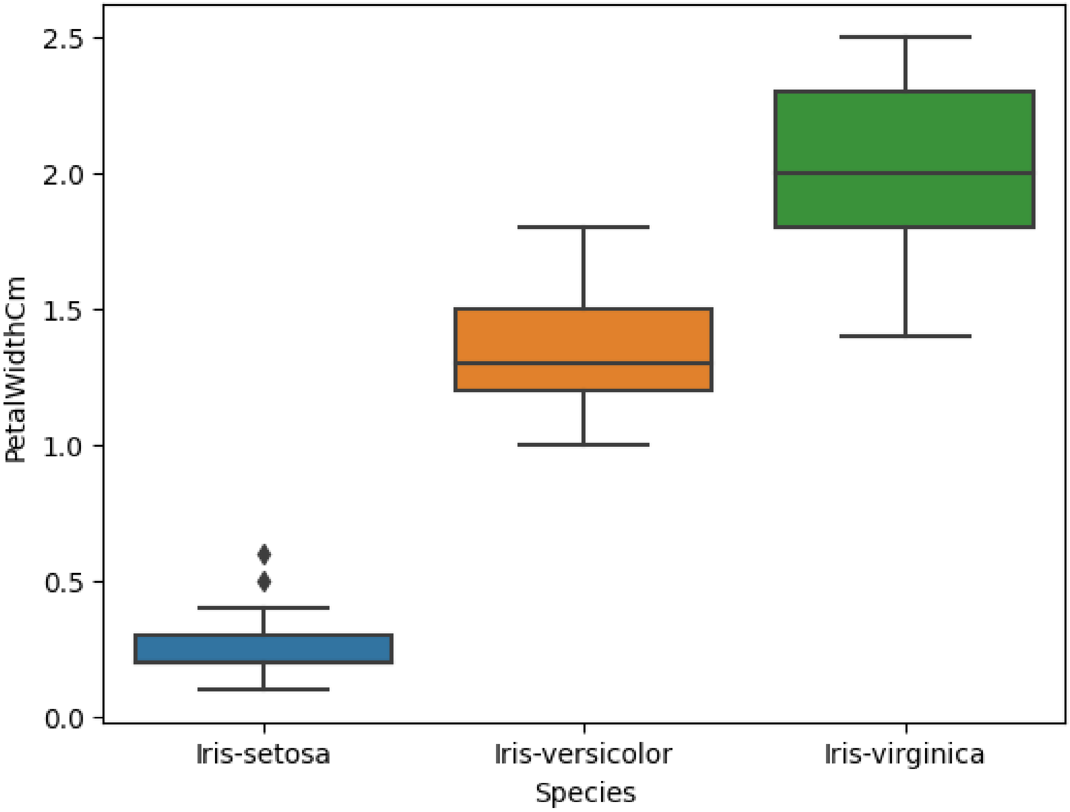
```
<Figure size 1000x1000 with 0 Axes>
```

```
def graph(y):
    sns.boxplot(x="Species", y=y, data=df)
    plt.figure(figsize=(10,10))

graph('PetalLengthCm')

plt.show()
```



```
<Figure size 1000x1000 with 0 Axes>
```

```
def graph(y):
    sns.boxplot(x="Species", y=y, data=df)
    plt.figure(figsize=(10,10))

graph('PetalWidthCm')

plt.show()
```



```
<Figure size 1000x1000 with 0 Axes>
```

Droping the Id Column

```
df = df.drop('Id', axis=1)
```

```
df
```

```
     SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  
Species
0              5.1           3.5            1.4           0.2     
Iris-setosa
1              4.9           3.0            1.4           0.2     
Iris-setosa
2              4.7           3.2            1.3           0.2     
Iris-setosa
3              4.6           3.1            1.5           0.2     
Iris-setosa
4              5.0           3.6            1.4           0.2     
Iris-setosa
..             ...           ...            ...           ...     
...
145            6.7           3.0            5.2           2.3  Iris-
virginica
146            6.3           2.5            5.0           1.9  Iris-
virginica
```

```
147           6.5           3.0            5.2           2.0  Iris-
virginica
148           6.2           3.4            5.4           2.3  Iris-
virginica
149           5.9           3.0            5.1           1.8  Iris-
virginica

[150 rows x 5 columns]
```

X contains Independent Variables and Y contains Dependent Varaibles

```
x = df.iloc[:, :4]
```

```
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]
```

This line selects all rows and the first four columns of the DataFrame df which are **SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm** using the iloc function. df.iloc[:, :4] selects all rows (:) and the columns indexed from 0 to 3 (:4). **The resulting x will contain the input features for classification.**

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

```
y
```

```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

This line selects all rows and the fifth column of the DataFrame df which is **(Species)**, which corresponds to the target labels for classification. Again, df.iloc[:, 4] selects all rows (:) and the column at index 4. **The resulting y will contain the target labels**

Import train_test_split to split the data into train and test datasets.

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=45)
```

```
x_train.shape
```

```
(120, 4)
```

x_train has a shape of (112, 4), it means that there are 112 training samples, and each sample has 4 features or input variables.

```
x_test.shape
```

```
(30, 4)
```

x_test has a shape of (38, 4), it means that there are 38 test samples, and each sample has 4 features or input variables.

```
y_train.shape
```

```
(120,)
```

```
y_test.shape
```

```
(30,)
```

Creating Model - Classification -- Classifiying the Iris Flower Dataset using the Logistic Regression

```
from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings('ignore')
```

```
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
```

```
from sklearn import preprocessing
from sklearn import utils
```

```
lab = preprocessing.LabelEncoder()
encoded = lab.fit_transform(y_train)
```

```
print(utils.multiclass.type_of_target(y_train))
```

```
multiclass
```

```
print(utils.multiclass.type_of_target(encoded))
```

```
multiclass
```

```
model = LogisticRegression()
```

Training the model using the fit method -- Passsing the x_train and y_train in the fit function

```
threshold = 0.5
```

```
model.fit(x_train,y_train)
```

```
LogisticRegression()
```

Predicting the results using Predict Method

```
y_pred=model.predict(x_test)
```

```
print(classification_report(y_test, y_pred))
```

```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        11
Iris-versicolor       0.88      1.00      0.93         7
 Iris-virginica       1.00      0.92      0.96        12

       accuracy                           0.97        30
      macro avg       0.96      0.97      0.96        30
   weighted avg       0.97      0.97      0.97        30
```

```
print("Accuracy Score:" , accuracy_score(y_test, y_pred))
```

```
Accuracy Score: 0.9666666666666667
```

```
print("Accuracy Score:" , accuracy_score(y_test, y_pred) *100)
```

```
Accuracy Score: 96.66666666666667
```

Accuracy is 96.66

Predicting the Model

```
y_pred = model.predict([[5.0, 3.6, 1.4, 0.2]])
print(*y_pred)
```

```
Iris-setosa
```

Predicted the Species of Iris Flower which is IRIS- SETOSA

KNN alogrithm

```
from sklearn.neighbors import KNeighborsClassifier
```

```
model = KNeighborsClassifier()
```

```
model.fit(x_test, y_test)
```

```
KNeighborsClassifier()
```

```
print("Accuracy:" ,model.score(x_test, y_test) )
```

```
Accuracy: 1.0
```

```
print("Accuracy:" ,model.score(x_test, y_test) * 100)
```

```
Accuracy: 100.0
```

```
y_pred = model.predict([[5.6, 2.5, 3.9, 1.1]])
print(*y_pred)
```

```
Iris-versicolor
```

The predicted flower is IRIS-VERSICOLOR

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
```

```
model = DecisionTreeClassifier()
```

```
model.fit(x_train, y_train)
```

```
DecisionTreeClassifier()
```

```
print("Accuracy:" ,model.score(x_test, y_test))
```

```
Accuracy: 1.0
```

```
print("Accuracy:", model.score(x_test, y_test) *100)
```

```
Accuracy: 100.0
```

Accuracy of the model is 100

Predict the Model

```
y_pred = model.predict([[6.2, 2.8, 4.8, 1.8]])
print(*y_pred)
```

```
Iris-virginica
```

Predicted flower is IRIS-VIRGINICA

Random Forest

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
```

```
model.fit(x_train, y_train)
```

```
RandomForestClassifier()
```

```
RF_predictions = model.predict(x_test)
```

```
print("Accuracy:", model.score(x_test, y_test) *100)
```

```
Accuracy: 96.66666666666667
```

Accuracy of the model is 93.33

```
y_pred = model.predict([[6.9, 1.8, 4.4, 1.6]])
print(*y_pred)
```

```
Iris-versicolor
```

Predicted flower is IRIS-Versicolor

Support Vector Machine Algorithm

```
from sklearn.svm import SVC
model = SVC()
```

```
model.fit(x_train, y_train)
```

```
SVC()
```

```
SVM_predictions = model.predict(x_test)
```

```
print("Accuracy:", model.score(x_test, y_test) *100)
```

```
Accuracy: 96.66666666666667
```

Accuracy of the model is 96.66

```
y_pred = model.predict([[5.1, 3.5, 1.4, 0.2]])
print(*y_pred)
```

```
Iris-setosa
```

Predicted flower is IRIS-SETOSA

End of the code