Name = Maalika Maini

Internship at Let's grow more

In [15]:

```python
# Loading Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
```

Loadind datasets

In [16]:

```python
data_link="http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
iris_data=pd.read_csv("http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data")
```

Reading the dataset

In [17]:

```python
iris_data.head() # starting 5 values
```

Out[17]:

	5.1	3.5	1.4	0.2	Iris-setosa
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [18]:

```python
iris_data.tail() #last 5 values
```

Out[18]:

	5.1	3.5	1.4	0.2	Iris-setosa
144	6.7	3.0	5.2	2.3	Iris-virginica
145	6.3	2.5	5.0	1.9	Iris-virginica
146	6.5	3.0	5.2	2.0	Iris-virginica
147	6.2	3.4	5.4	2.3	Iris-virginica
148	5.9	3.0	5.1	1.8	Iris-virginica

In [19]:

```python
iris_data.columns
```

Out[19]: Index(['5.1', '3.5', '1.4', '0.2', 'Iris-setosa'], dtype='object')

In [20]:

```python
columns=['sepal_length','sepal_width','petal_length','petal_width','class']
iris_data.columns=columns
iris_data.head()
```

Out[20]:

	sepal_length	sepal_width	petal_length	petal_width	class
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

Explorartory Data Analysis on Dataset

In [21]:

```python
iris_data.shape # To get no.of columns
```

Out[21]: (149, 5)

In [22]:

```python
iris_data.describe()
```

Out[22]:

	sepal_length	sepal_width	petal_length	petal_width
count	149.000000	149.000000	149.000000	149.000000
mean	5.848322	3.051007	3.774497	1.205369
std	0.828594	0.433499	1.759651	0.761292
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.400000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [24]:

```python
iris_data.isnull()
```

Out[24]:

	sepal_length	sepal_width	petal_length	petal_width	class
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
144	False	False	False	False	False
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False

149 rows × 5 columns

In [10]:

```python
# Checking missing values
iris_data.isnull().sum()
```

Out[10]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
class           0
dtype: int64
```

In [12]:

```python
iris_data.value_counts() # To get the count of each features in the columns
```

Out[12]:

```
sepal_length  sepal_width  petal_length  petal_width  class
4.9           3.1          1.5           0.1          Iris-setosa        3
5.8           2.7          5.1           1.9          Iris-virginica     2
6.2           3.4          5.4           2.3          Iris-virginica     1
6.3           2.3          4.4           1.3          Iris-versicolor    1
              2.5          4.9           1.5          Iris-versicolor    1
                                                                        ..
5.5           2.4          3.7           1.0          Iris-versicolor    1
                           3.8           1.1          Iris-versicolor    1
              2.5          4.0           1.3          Iris-versicolor    1
              2.6          4.4           1.2          Iris-versicolor    1
7.9           3.8          6.4           2.0          Iris-virginica     1
Length: 146, dtype: int64
```

In [13]:

```python
iris_data.max()
```

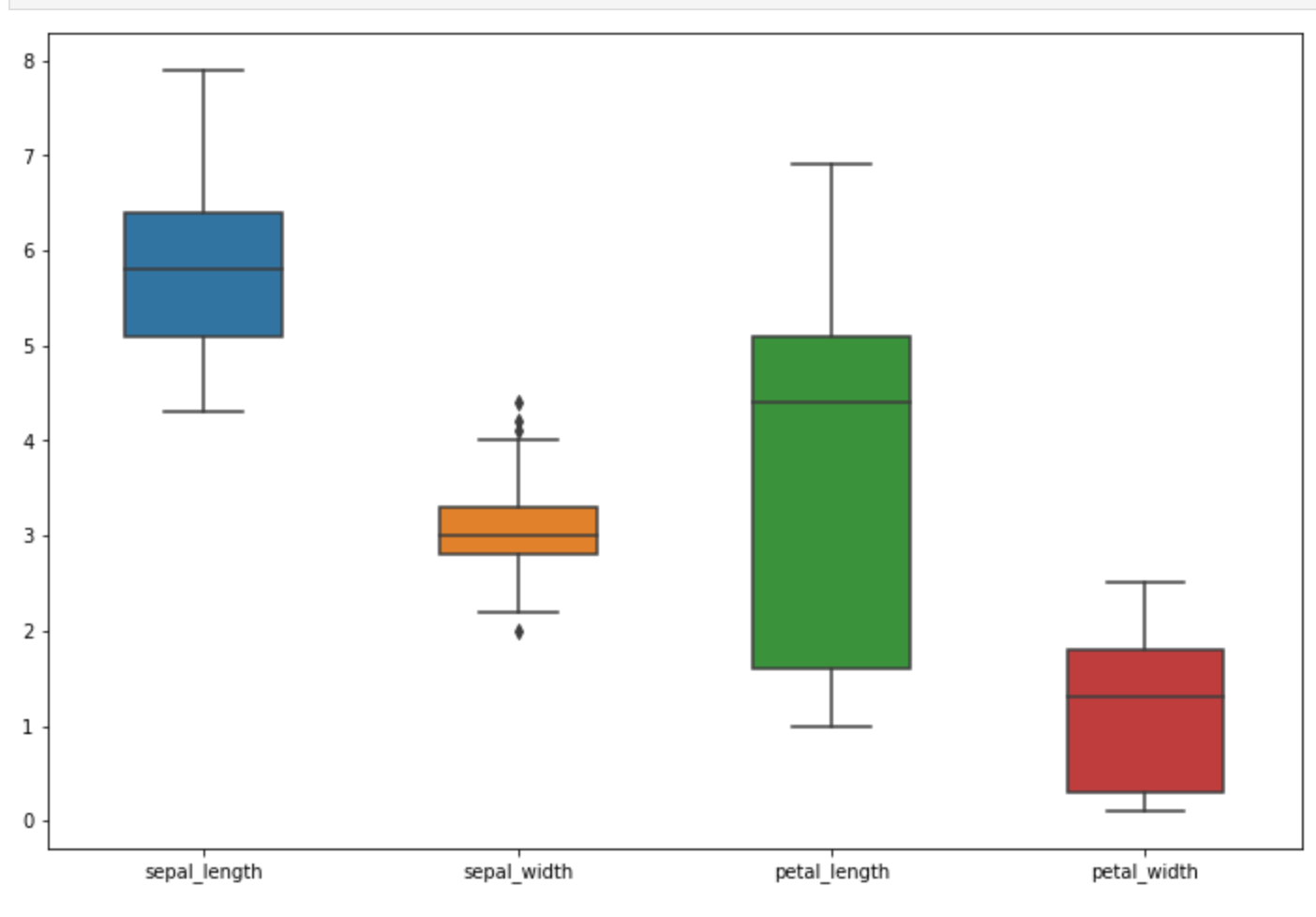
Out[13]:

```
sepal_length                7.9
sepal_width                 4.4
petal_length                6.9
petal_width                 2.5
class           Iris-virginica
dtype: object
```

Visualization

In [22]:

```python
plt.figure(figsize=(12,8))
sns.boxplot(data=iris_data,width=0.5,fliersize=5)
plt.show()
```



In [23]:

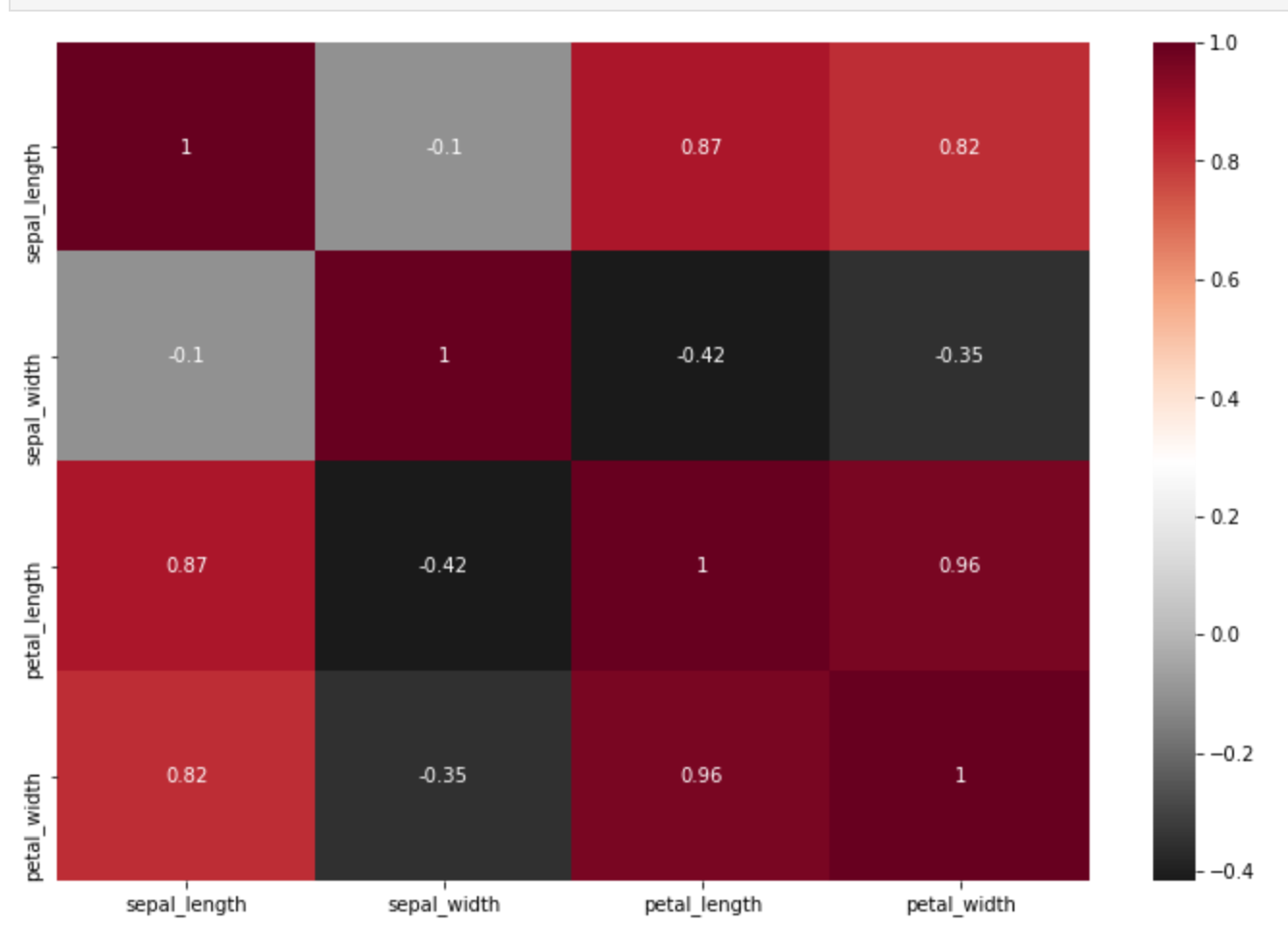
```python
corr=iris_data.corr()
corr
```

Out[23]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.103784	0.871283	0.816971
sepal_width	-0.103784	1.000000	-0.415218	-0.350733
petal_length	0.871283	-0.415218	1.000000	0.962314
petal_width	0.816971	-0.350733	0.962314	1.000000

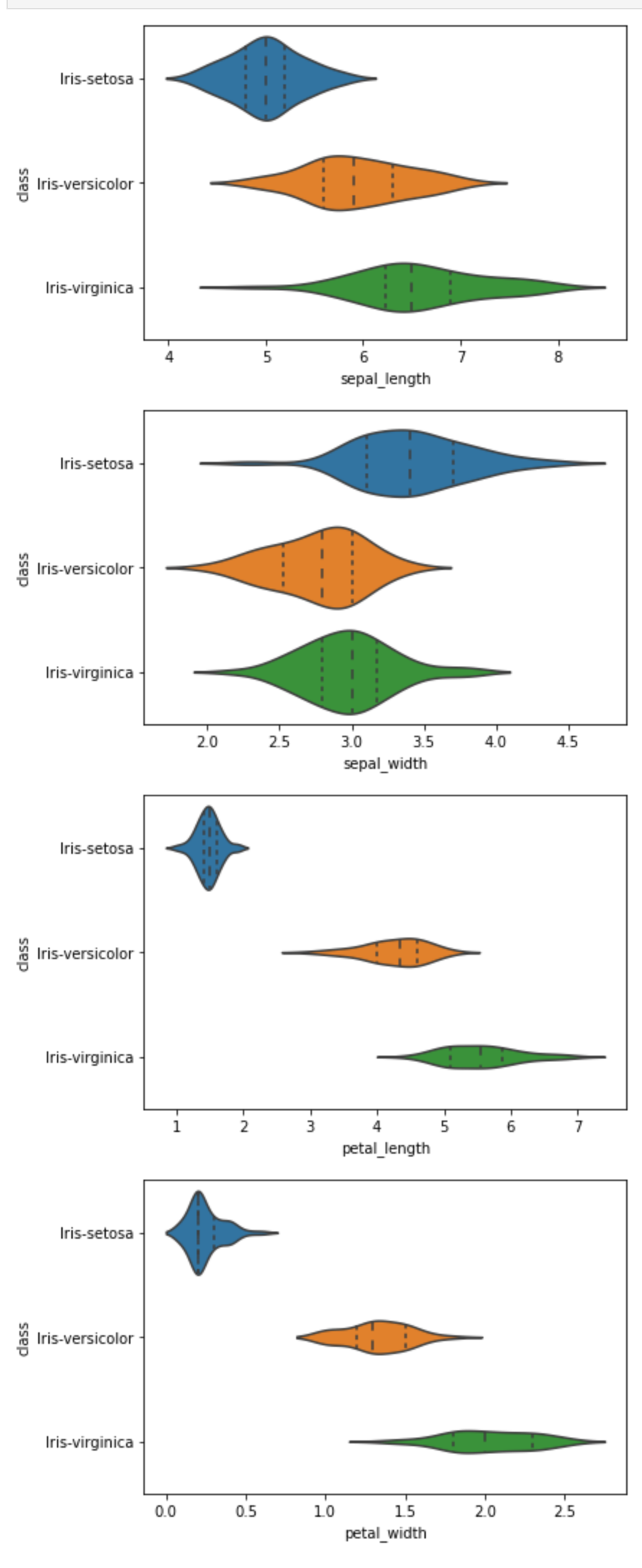
In [24]:

```python
plt.figure(figsize=(12,8))
sns.heatmap(corr,annot=True,cmap='RdGy_r')
plt.show()
```



In [25]:

```python
# Viol in Plot
sns.violinplot(y='class',x='sepal_length',data=iris_data,inner='quartile')
plt.show()
sns.violinplot(y='class',x='sepal_width',data=iris_data,inner='quartile')
plt.show()
sns.violinplot(y='class',x='petal_length',data=iris_data,inner='quartile')
plt.show()
sns.violinplot(y='class',x='petal_width',data=iris_data,inner='quartile')
plt.show()
```

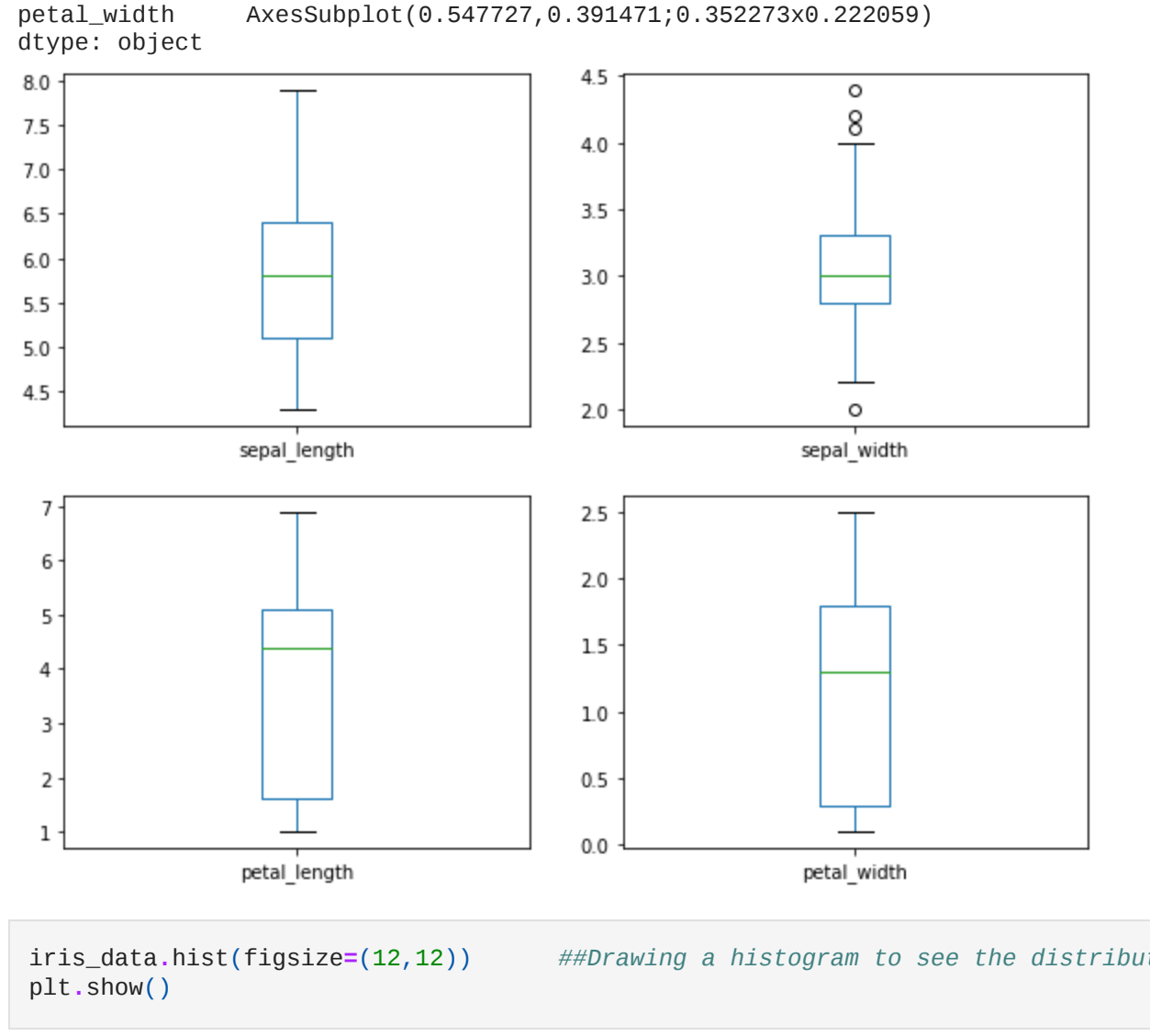


In [26]:

```python
# Box and Whisker plots
iris_data.plot(kind='box',subplots=True, layout=(3,2),figsize=(10,12))
```

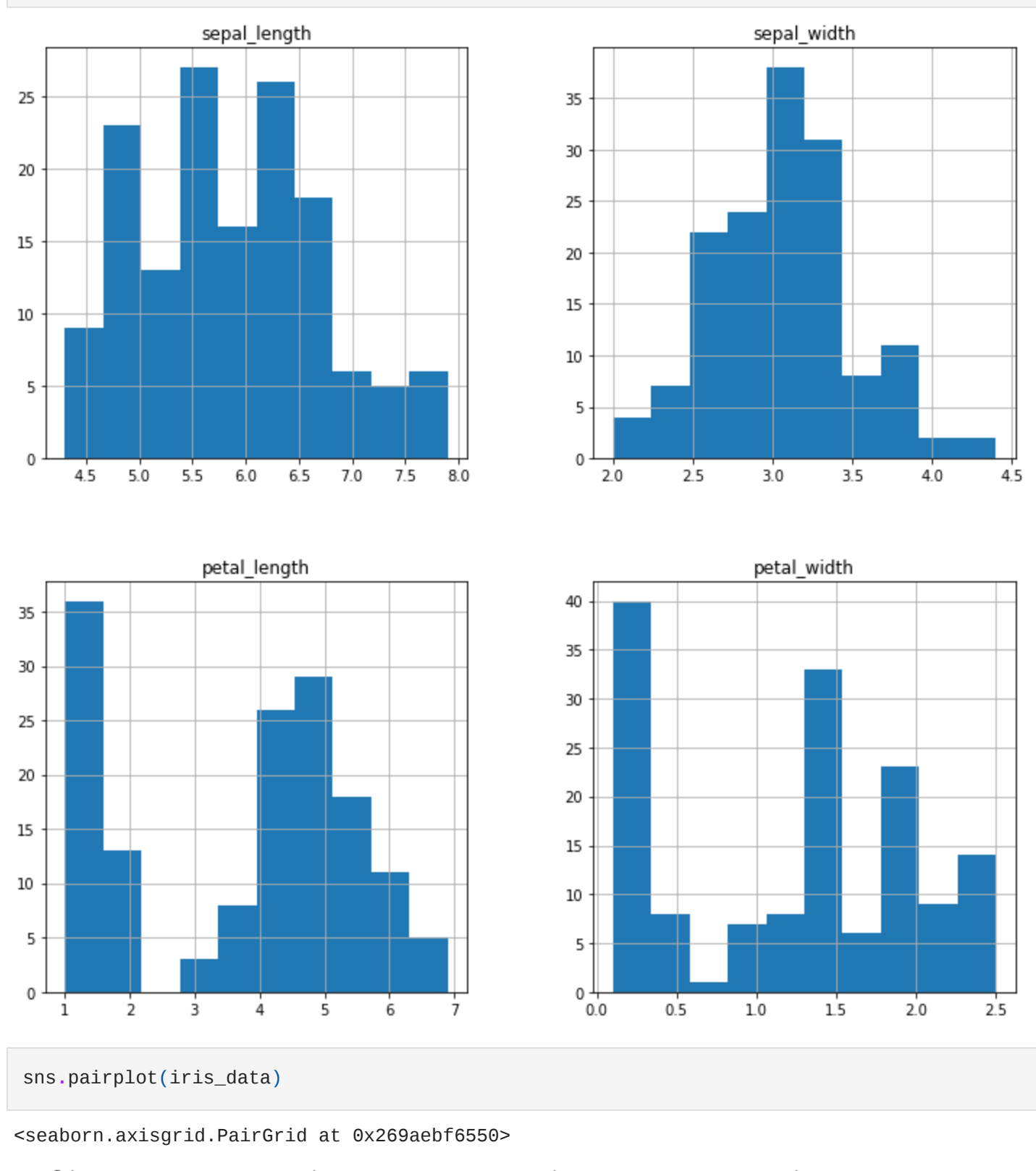
Out[26]:

```
sepal_length       AxesSubplot(0.125,0.657941;0.352273x0.222059)
sepal_width     AxesSubplot(0.547727,0.657941;0.352273x0.222059)
petal_length       AxesSubplot(0.125,0.391471;0.352273x0.222059)
petal_width     AxesSubplot(0.547727,0.391471;0.352273x0.222059)
dtype: object
```



In [27]:

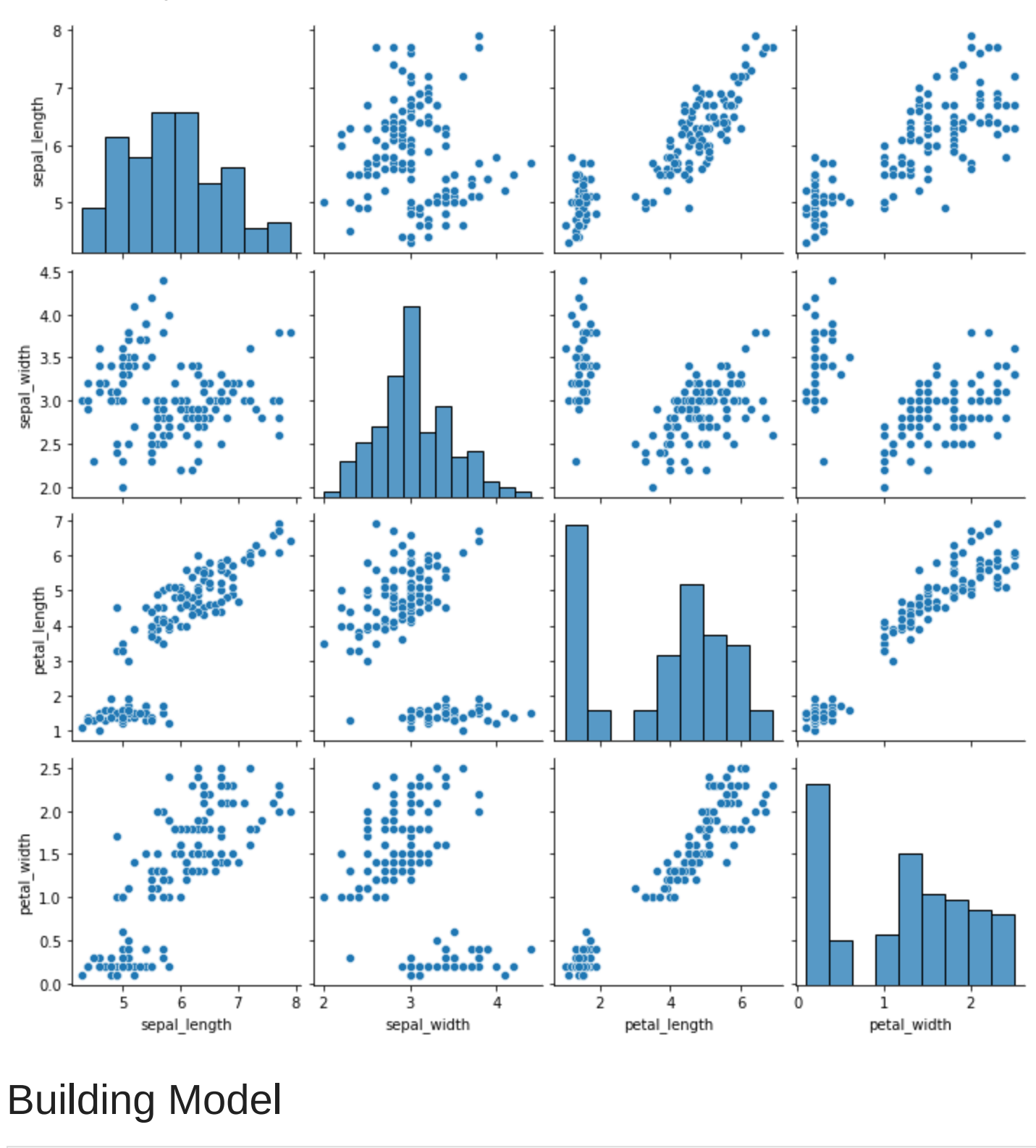
```python
iris_data.hist(figsize=(12,12))      ##Drawing a histogram to see the distribution of data
plt.show()
```



In [28]:

```python
sns.pairplot(iris_data)
```

Out[28]: <seaborn.axisgrid.PairGrid at 0x269aebf6550>



Building Model

In [33]:

```python
x=iris_data.drop(['class'],axis=1)
y=iris_data['class']
print(f'x shape:{x.shape}| y shape:{y.shape}')
```

```
x shape:(149, 4)| y shape:(149,)
```

In [34]:

```python
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.10,random_state=1)
```

In [37]:

```python
model=[]
model.append(('SVC',SVC(gamma='auto')))
```

In [45]:

```python
model=SVC(gamma='auto')
model.fit(x_train,y_train)
predict=model.predict(x_test)
```

In [47]:

```python
print(f'Test Accuracy:{accuracy_score(y_test,predict)}')
print(f'Classification_report:{classification_report(y_test,predict)}')
```

```
Test Accuracy:0.9333333333333333
Classification_report:                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00         4
Iris-versicolor       0.90      1.00      0.95         9
 Iris-virginica       1.00      0.50      0.67         2

       accuracy                           0.93        15
      macro avg       0.97      0.83      0.87        15
   weighted avg       0.94      0.93      0.92        15
```