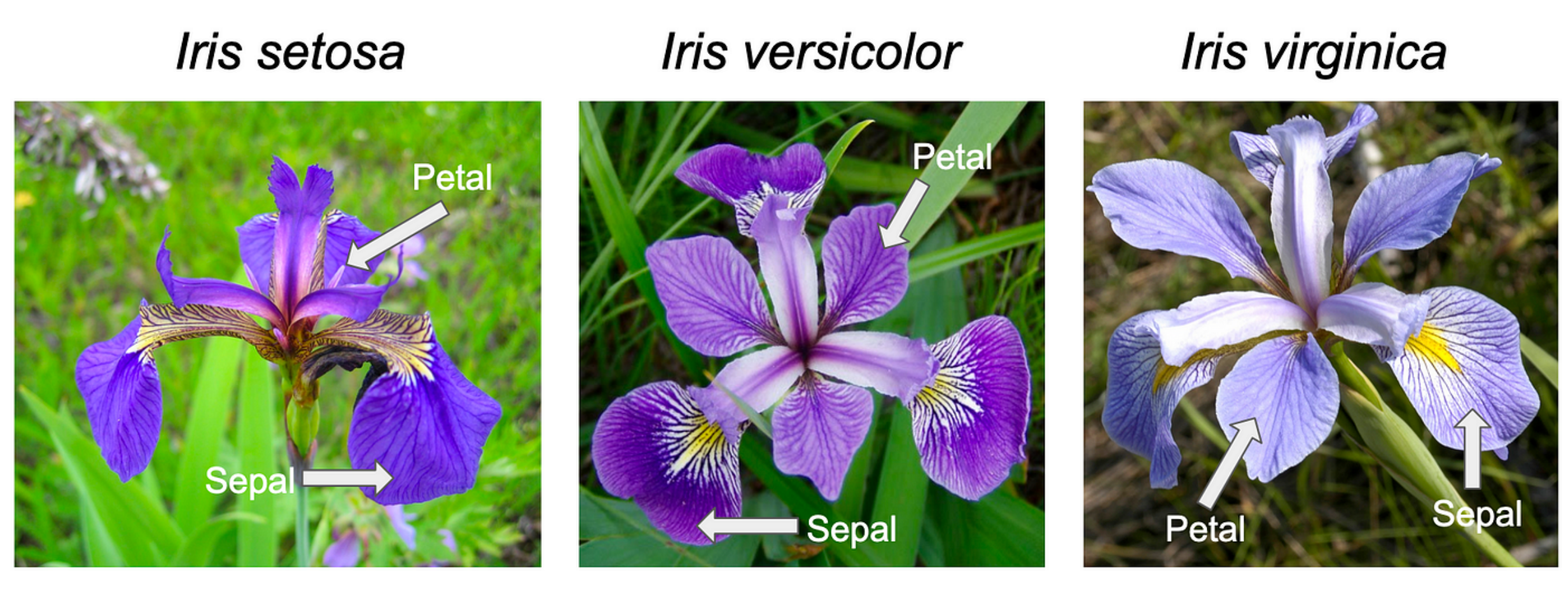
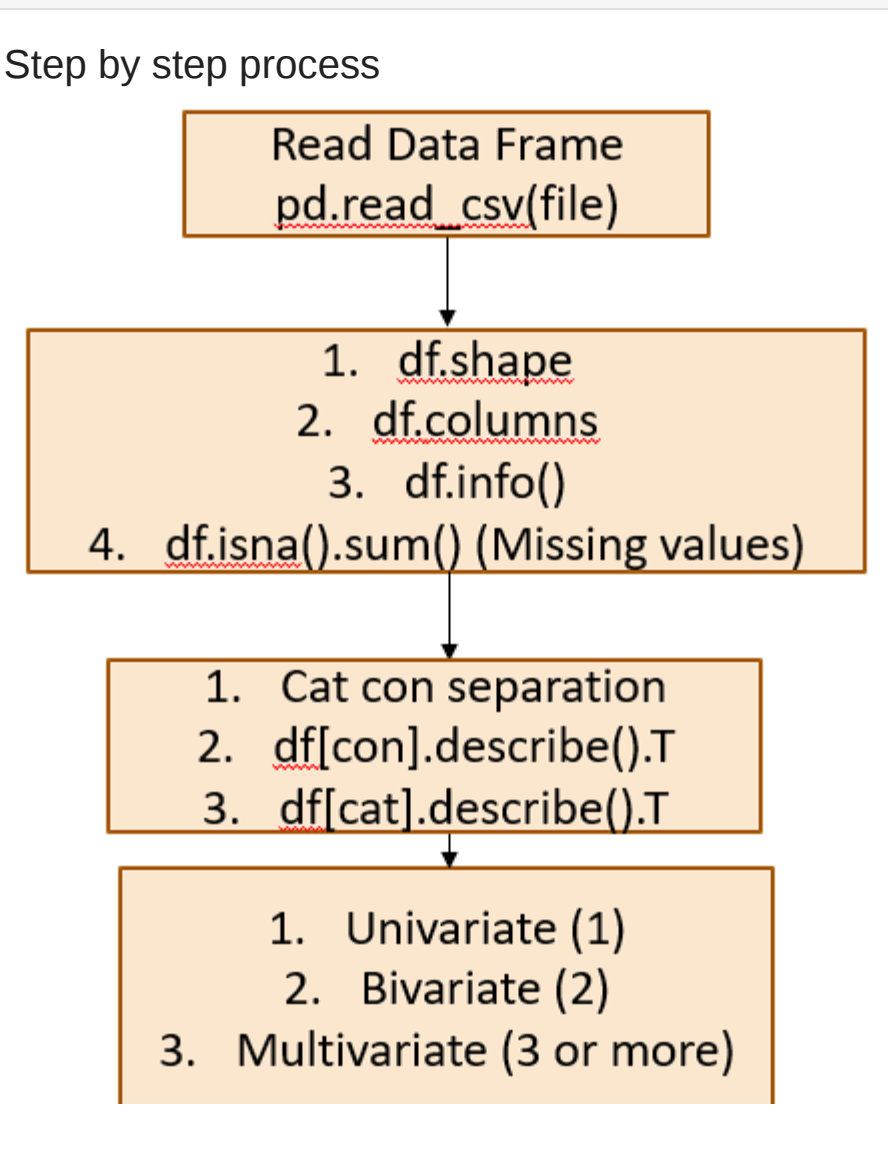
EDA Iris




```
In [1]:
from warnings import filterwarnings
filterwarnings('ignore')
```

EDA Step by step process



Step 1: Read CSV file

```
In [2]:
import os
os.chdir('E:/Datasets/')
```

```
In [3]:
import pandas as pd
df = pd.read_csv('iris.csv')
df.head()
```

Out[3]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Step 2 : Check the basic information about dataframe

```
In [4]:
df.shape
Out[4]:
(150, 5)
```

```
In [5]:
df.columns
Out[5]:
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

```
In [6]:
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
In [7]:
df.isna().sum()
Out[7]:
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

Step 3: Descriptive Statistics

```
In [9]:
# Cat con sep
cat = list(df.columns[df.dtypes=='object'])
cat
Out[9]:
['species']
```

```
In [11]:
con = list(df.columns[df.dtypes!='object'])
con
Out[11]:
['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
```

```
In [12]:
df[con].describe().T
```

Out[12]:

	count	mean	std	min	25%	50%	75%	max
sepal_length	150.0	5.843333	0.828066	4.3	5.1	5.80	6.4	7.9
sepal_width	150.0	3.057333	0.435866	2.0	2.8	3.00	3.3	4.4
petal_length	150.0	3.758000	1.765298	1.0	1.6	4.35	5.1	6.9
petal_width	150.0	1.199333	0.762238	0.1	0.3	1.30	1.8	2.5

```
In [13]:
df[cat].describe().T
```

Out[13]:

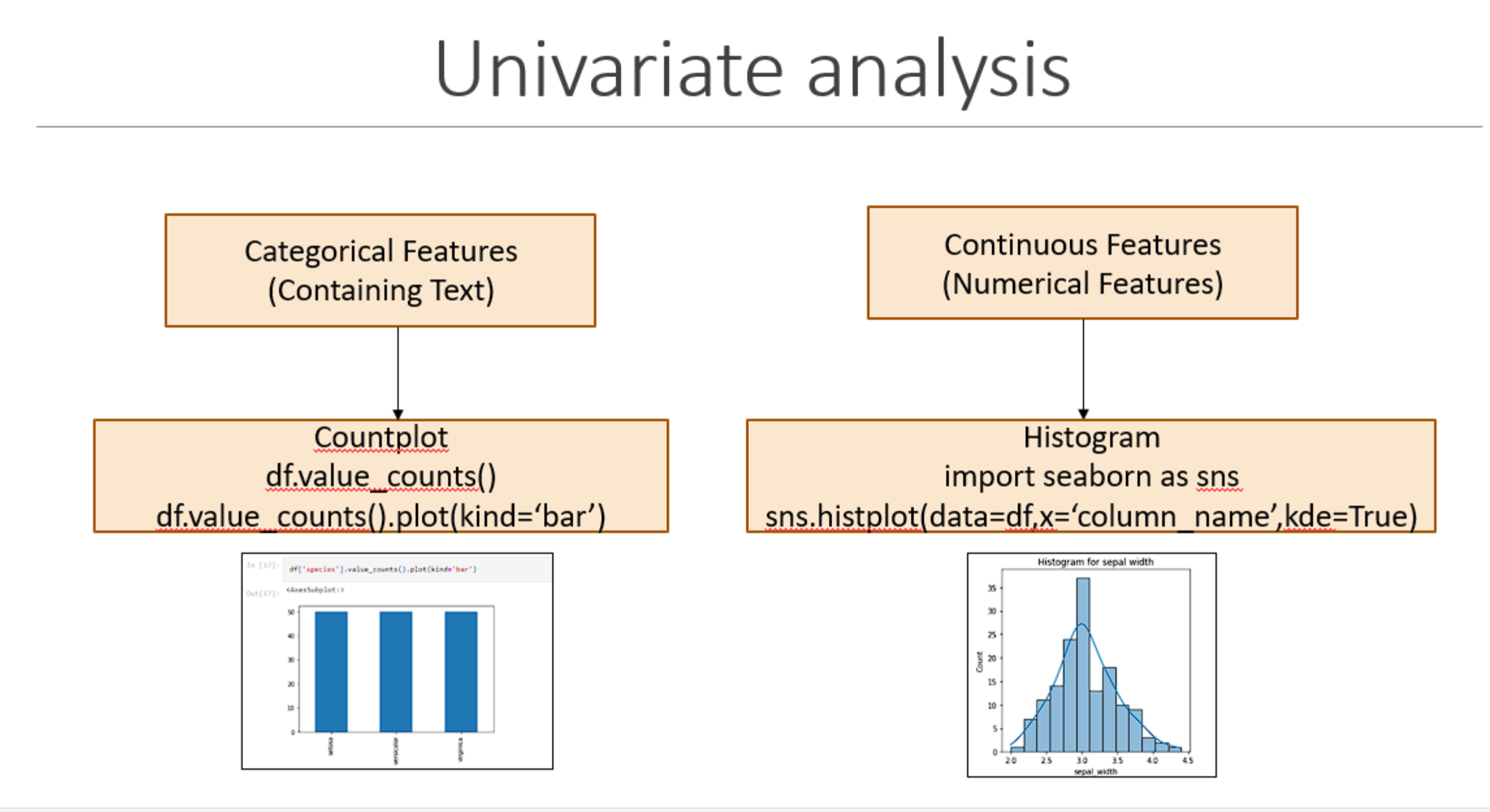
	count	unique	top	freq
species	150	3	setosa	50

```
In [14]:
df['species'].unique()
Out[14]:
array(['setosa', 'versicolor', 'virginica'], dtype=object)
```

Step 4: Visualization

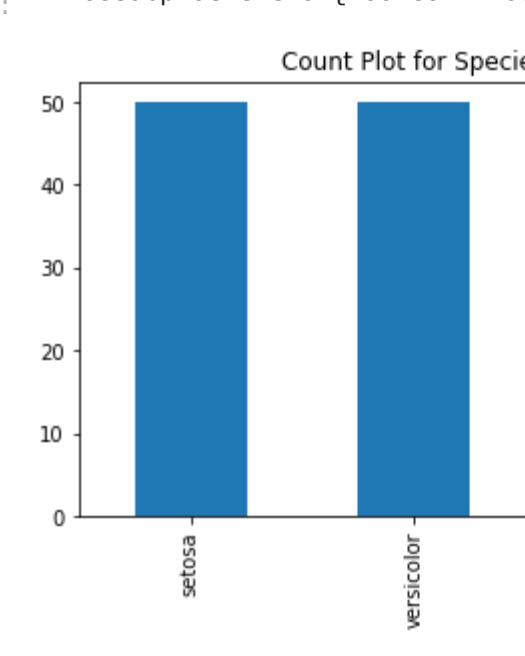
1. Univariate
2. Bivariate
3. Multivariate

```
In [17]:
import matplotlib.pyplot as plt
import seaborn as sns
```

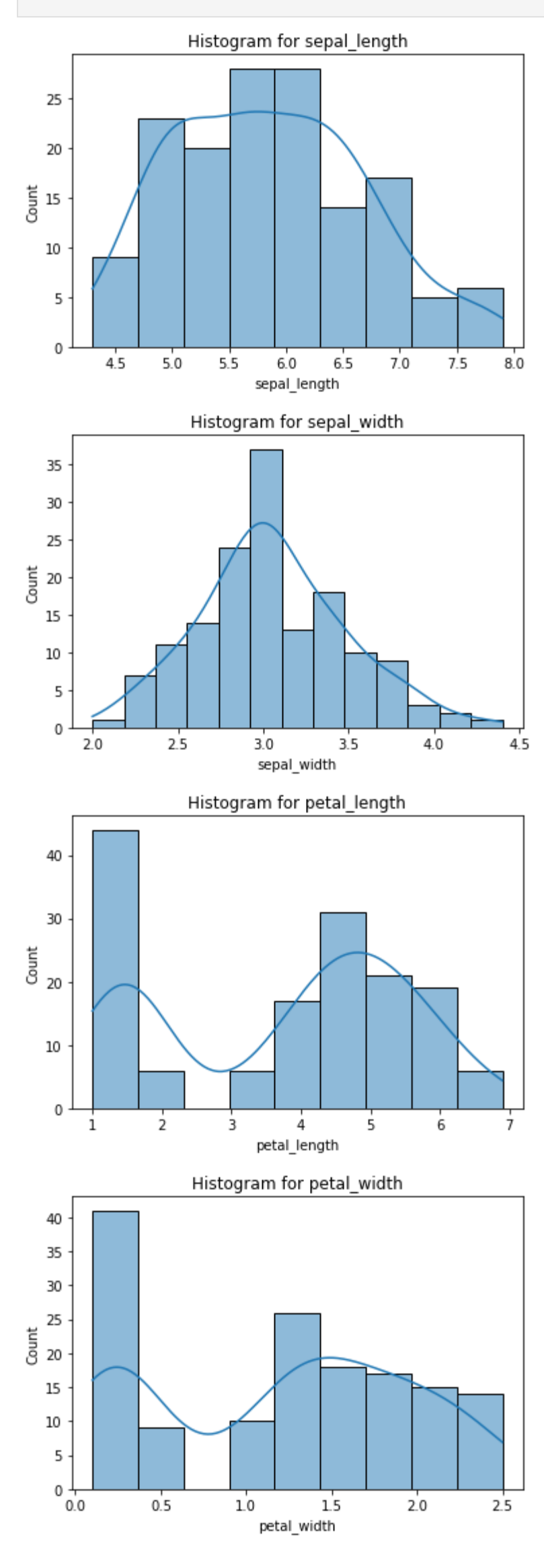



```
In [15]:
df['species'].value_counts()
Out[15]:
setosa        50
versicolor    50
virginica     50
Name: species, dtype: int64
```

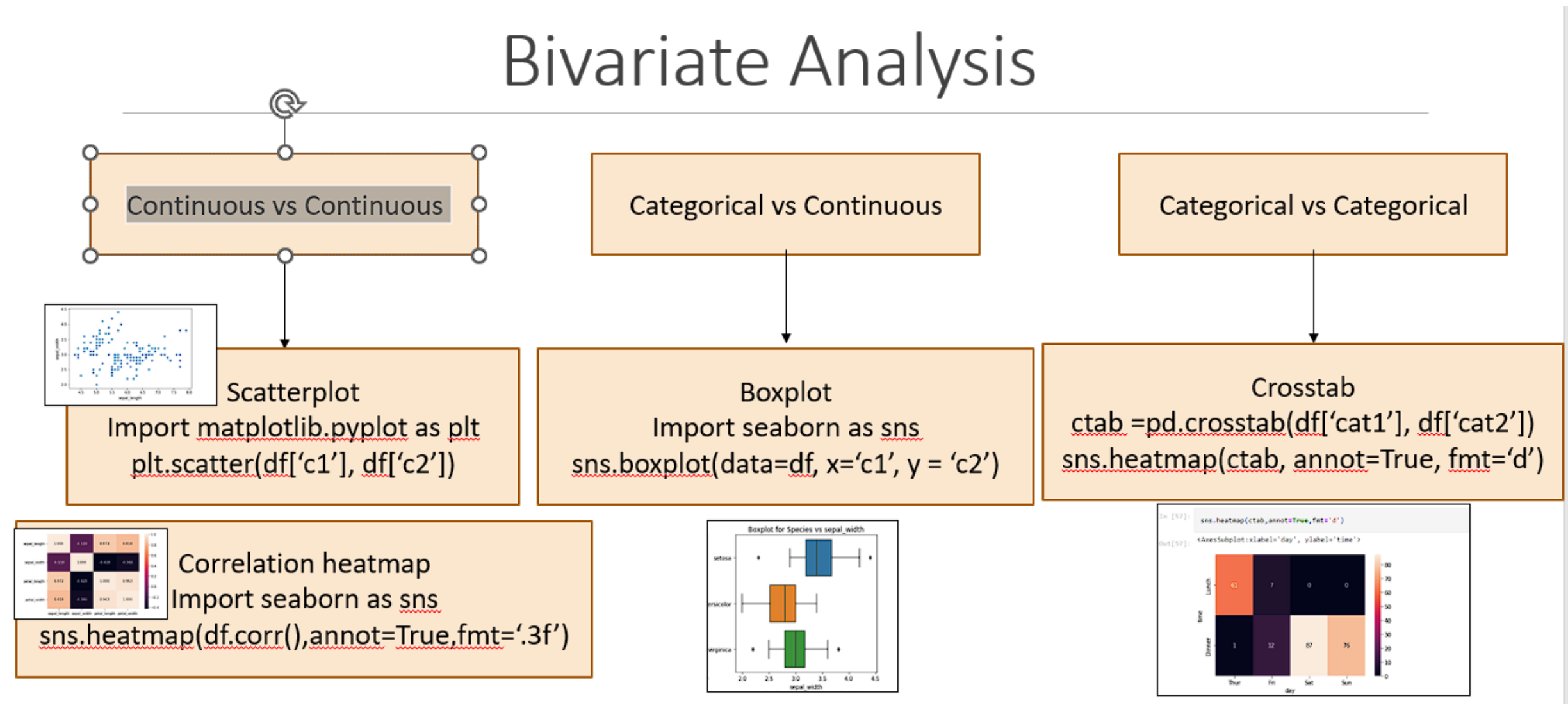
```
In [16]:
df['species'].value_counts().plot(kind='bar',title='Count Plot for Species')
Out[16]:
<AxesSubplot:title={'center':'Count Plot for Species'}>
```




```
In [18]:
for i in con:
    sns.histplot(data=df, x=i, kde=True)
    plt.title(f'Histogram for {i}')
    plt.show()
```

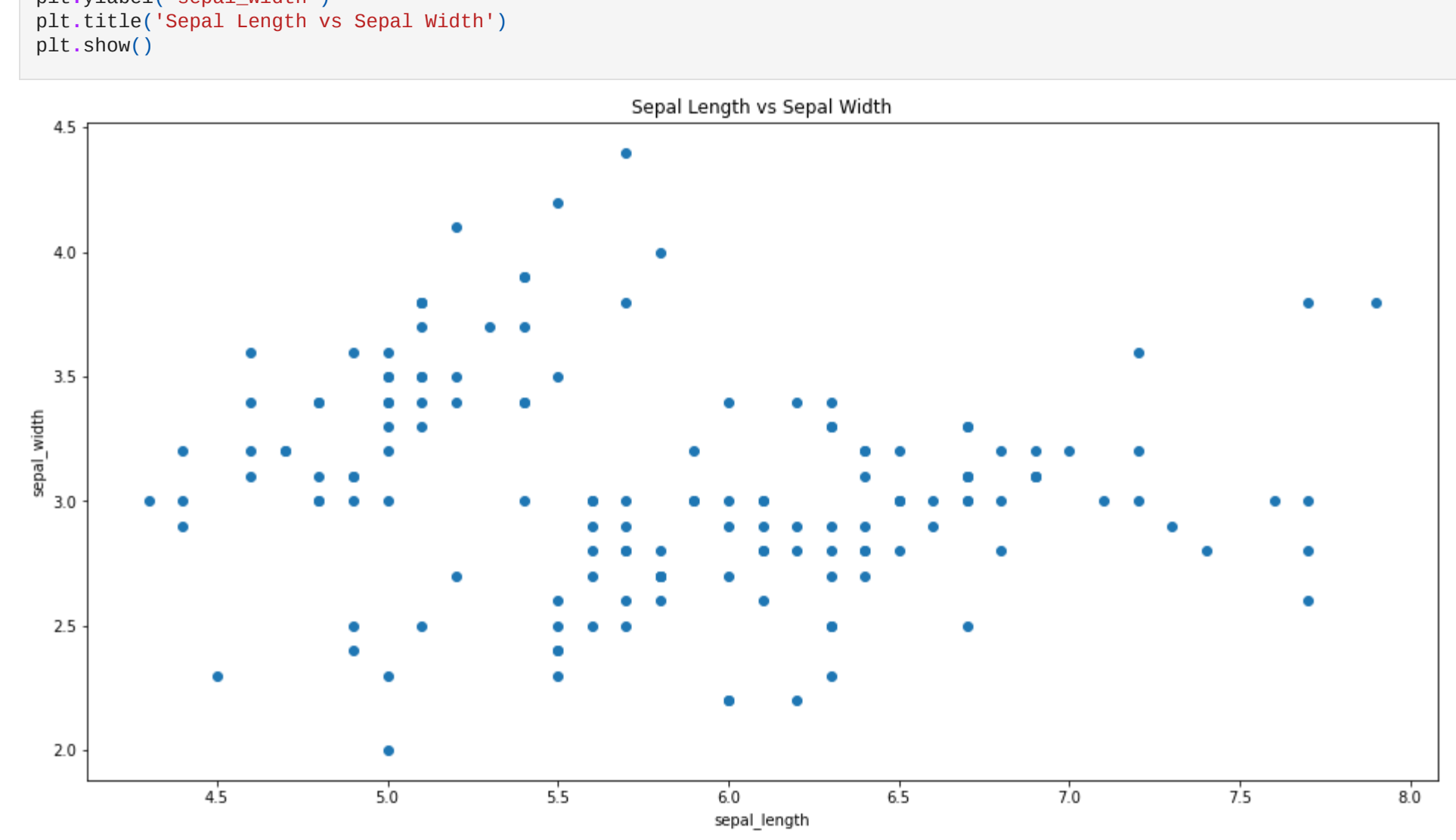


2. Bivariate Analysis

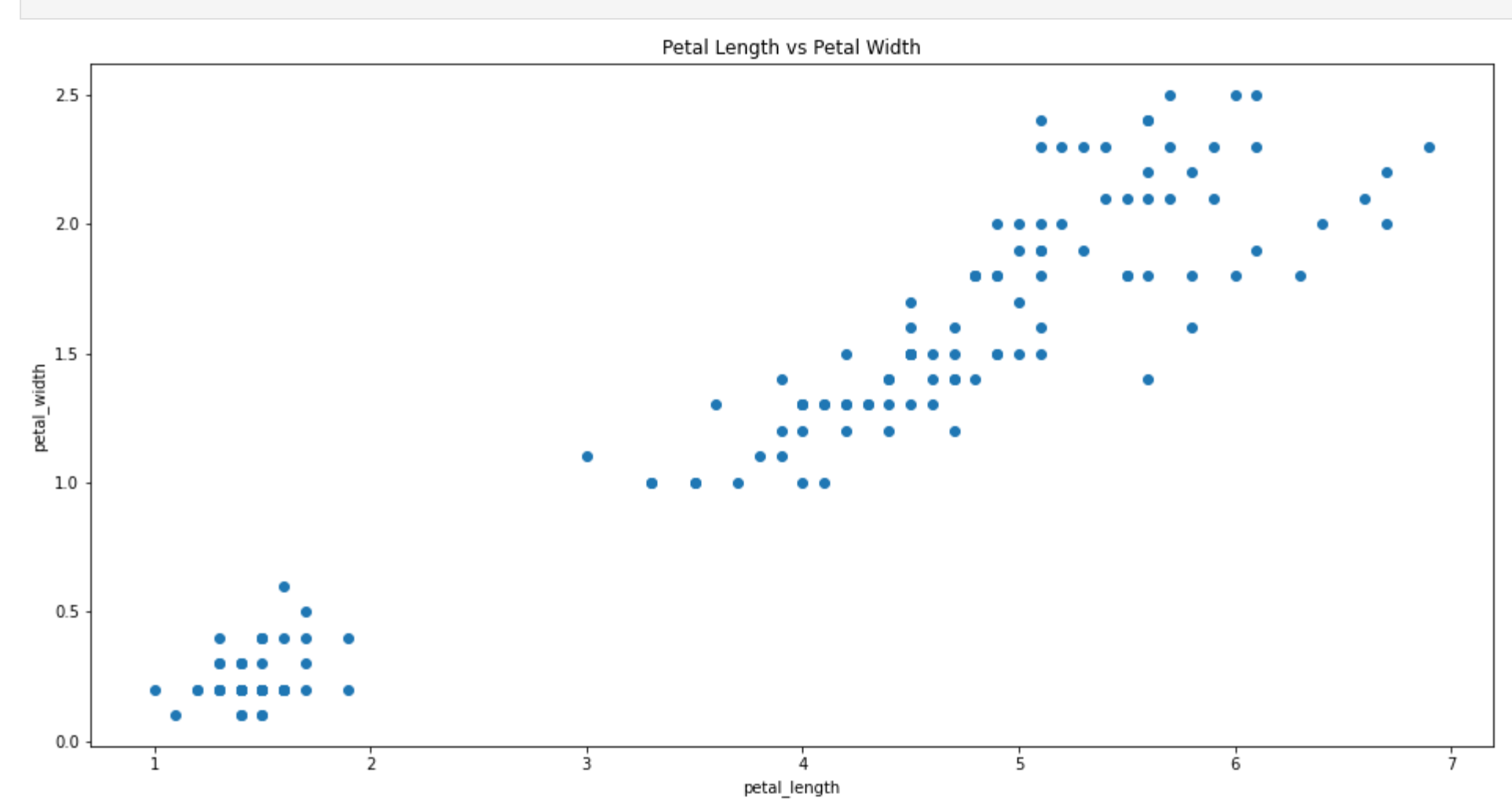



```
In [19]:
con
Out[19]:
['sepal_length', 'sepal_width', 'petal_length', 'petal_width']
```

```
In [23]:
plt.figure(figsize=(16,8))
plt.scatter(df['sepal_length'],df['sepal_width'])
plt.xlabel('sepal_length')
plt.ylabel('sepal_width')
plt.title('Sepal Length vs Sepal Width')
plt.show()
```




```
In [24]:
plt.figure(figsize=(16,8))
plt.scatter(df['petal_length'],df['petal_width'])
plt.xlabel('petal_length')
plt.ylabel('petal_width')
plt.title('Petal Length vs Petal Width')
plt.show()
```



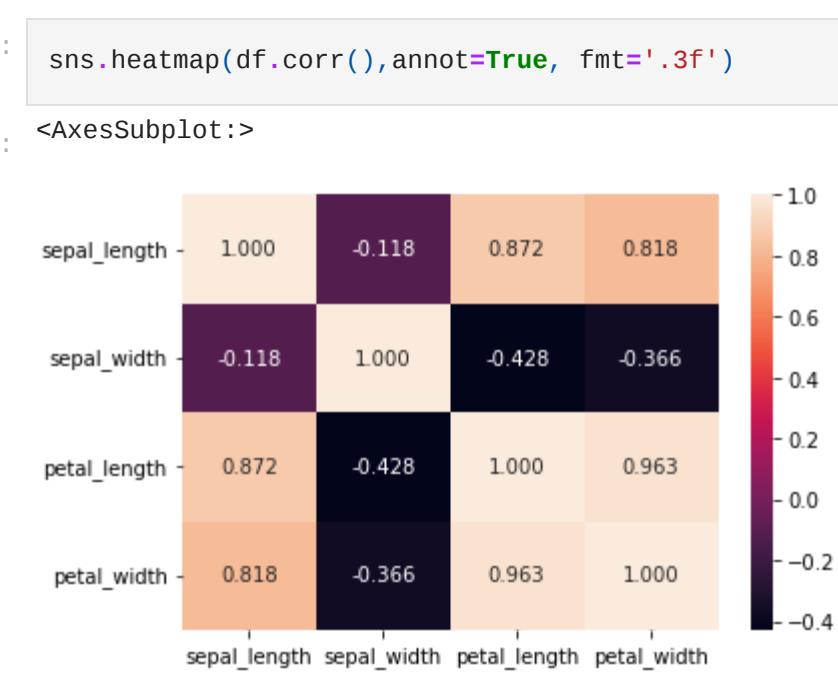
Correlation

```
In [25]:
df.corr()
```

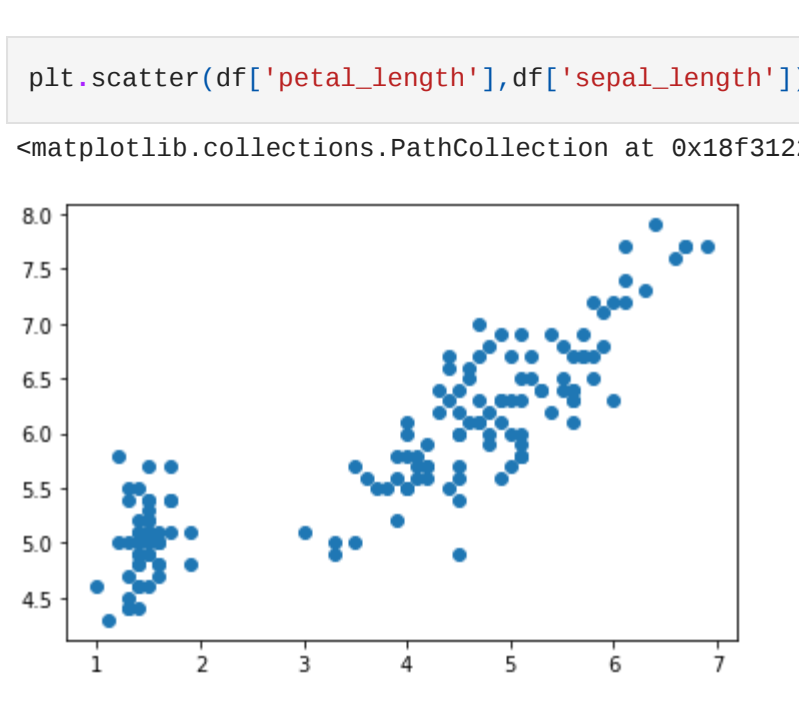
Out[25]:

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

```
In [32]:
sns.heatmap(df.corr(),annot=True, fmt='.3f')
Out[32]:
<AxesSubplot:>
```



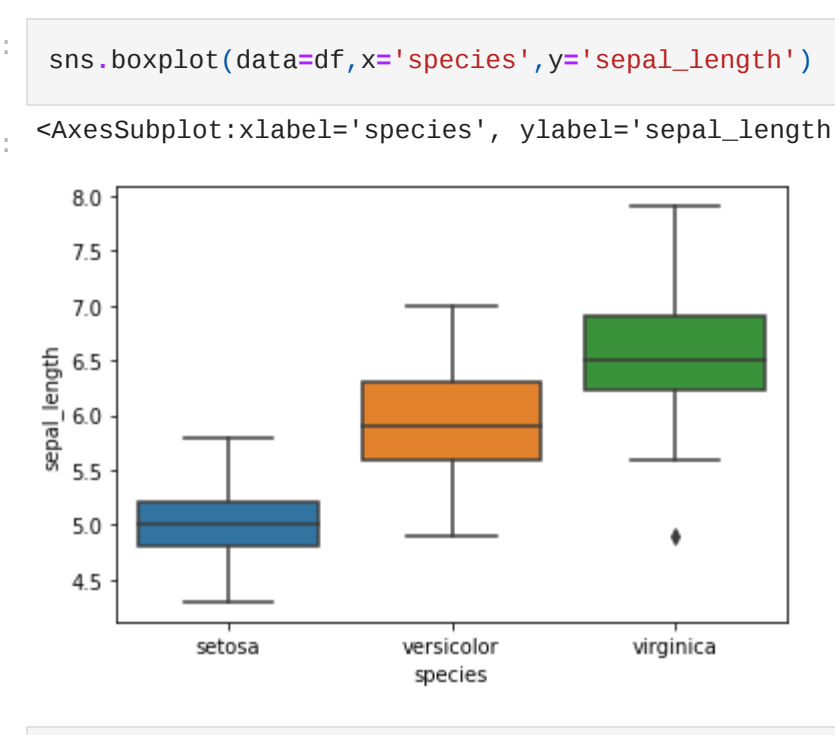

```
In [33]:
plt.scatter(df['petal_length'],df['sepal_length'])
Out[33]:
<matplotlib.collections.PathCollection at 0x18f31226b80>
```



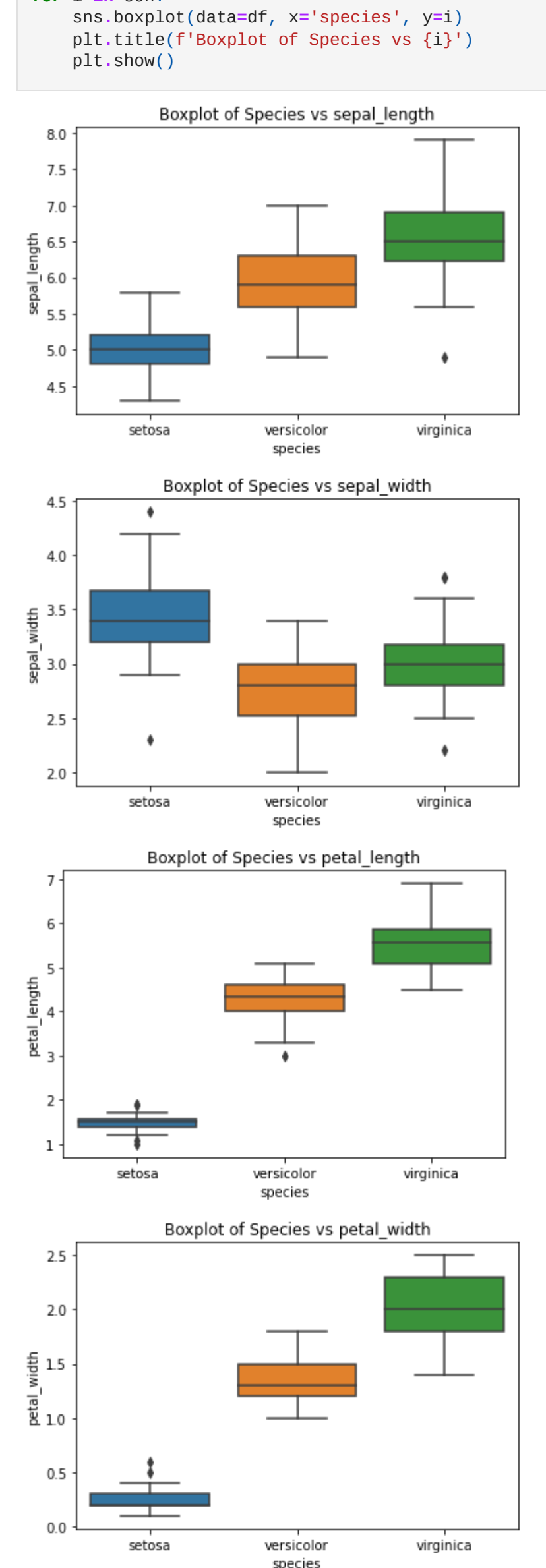
Cat vs Con

Boxplots

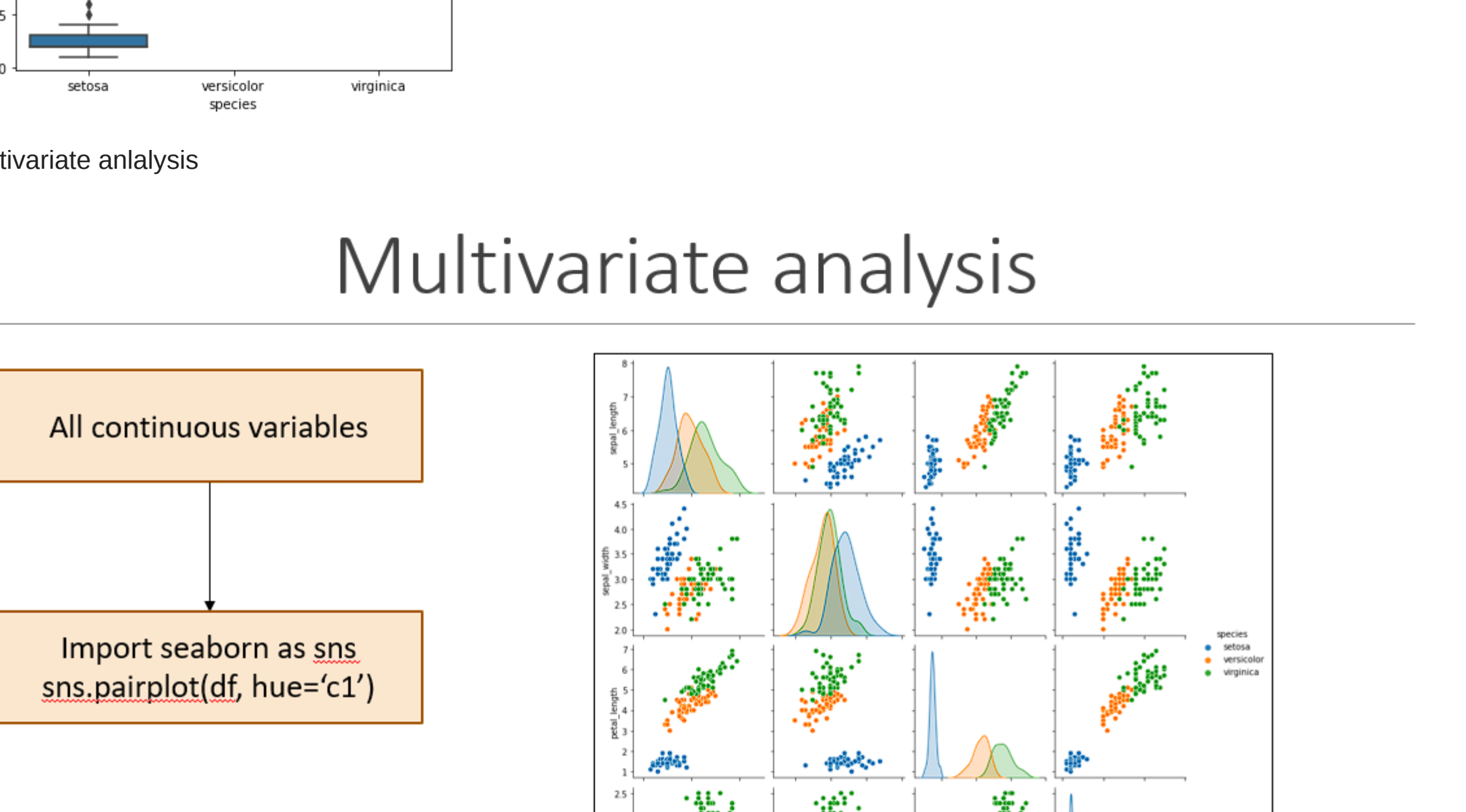
```
In [34]:
sns.boxplot(data=df,x='species',y='sepal_length')
Out[34]:
<AxesSubplot:xlabel='species', ylabel='sepal_length'>
```



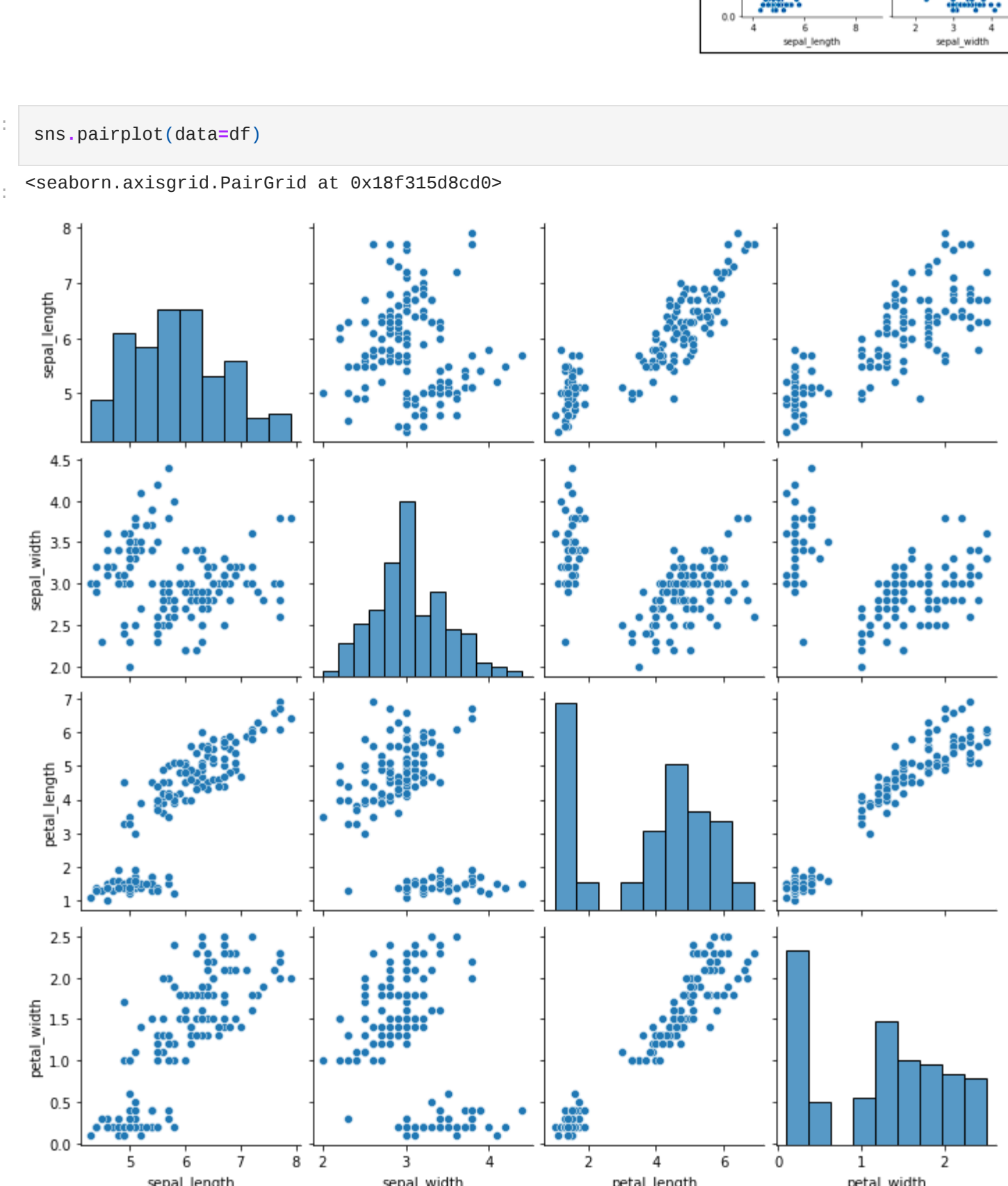

```
In [35]:
for i in con:
    sns.boxplot(data=df, x='species', y=i)
    plt.title(f'Boxplot of Species vs {i}')
    plt.show()
```



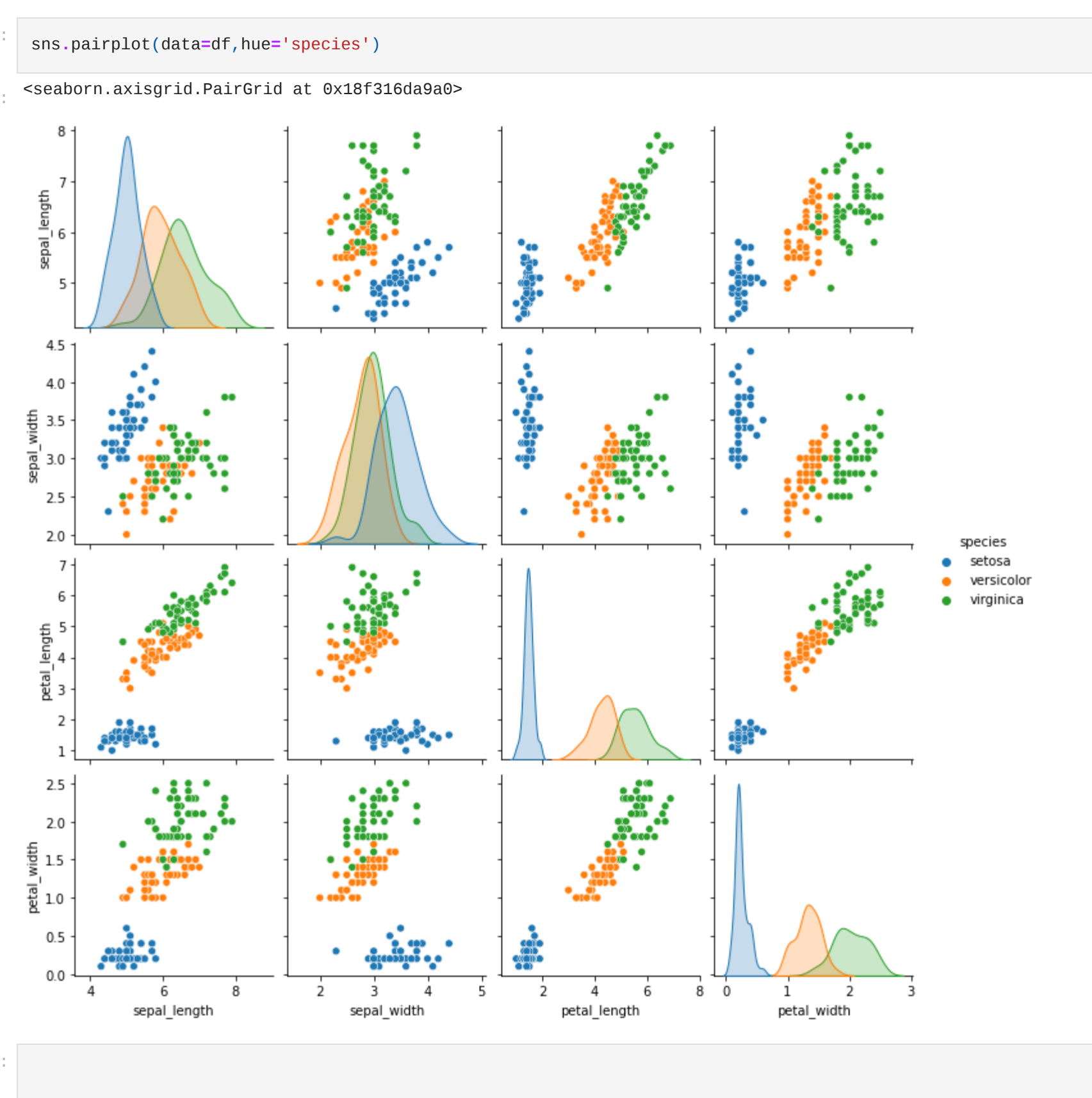
Multivariate anlalysis




```
In [36]:
sns.pairplot(data=df)
Out[36]:
<seaborn.axisgrid.PairGrid at 0x18f315d8cd0>
```




```
In [37]:
sns.pairplot(data=df,hue='species')
Out[37]:
<seaborn.axisgrid.PairGrid at 0x18f316da9a0>
```




```
In [ ]:
```