Assignment 3 - Matplotlib and Seaborn

Atharva Ramgirkar 19BCE0114

In [ ]:

Table of Content

- Importing Libraries
- Getting the data
- Matplotlib Visualizations
  - Line Plot
  - Scatter Plot
  - Bar Plot
  - Pie Chart
- Seaborn Visualizations
  - Scatter Plot
  - Line Plot
  - Count Plot
  - Box Plot
  - Pair Plot
  - Correlation Matrix
  - Heat Map

In [ ]:

Importing Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [ ]:

Getting the data

In [2]:
```
iris = sns.load_dataset('iris')
```

In [6]:
```
iris
```

Out[6]:

	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

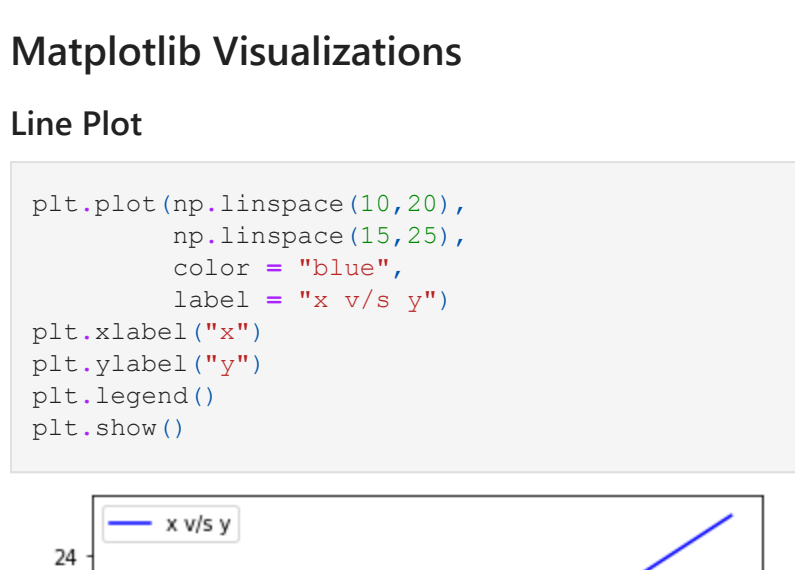
150 rows × 5 columns

In [ ]:

Matplotlib Visualizations

Line Plot

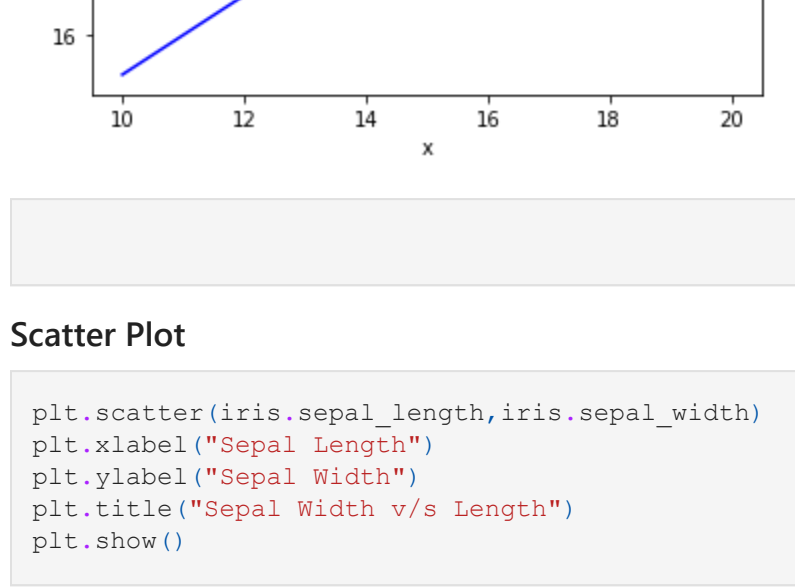
In [33]:
```
plt.plot(np.linspace(10,20),
         np.linspace(15,25),
         color = "blue",
         label = "x v/s y")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.show()
```



In [ ]:

Scatter Plot

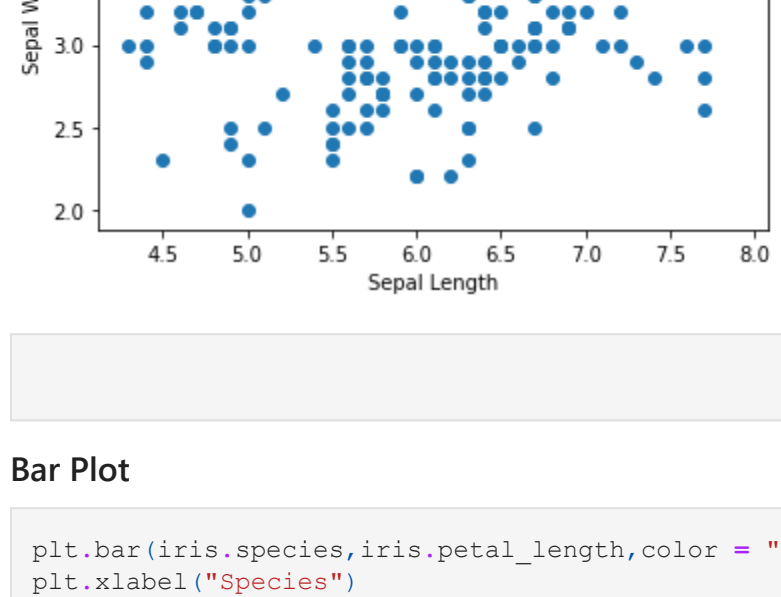
In [34]:
```
plt.scatter(iris.sepal_length,iris.sepal_width)
plt.xlabel("Sepal Length")
plt.ylabel("Sepal Width")
plt.title("Sepal Width v/s Length")
plt.show()
```



In [ ]:

Bar Plot

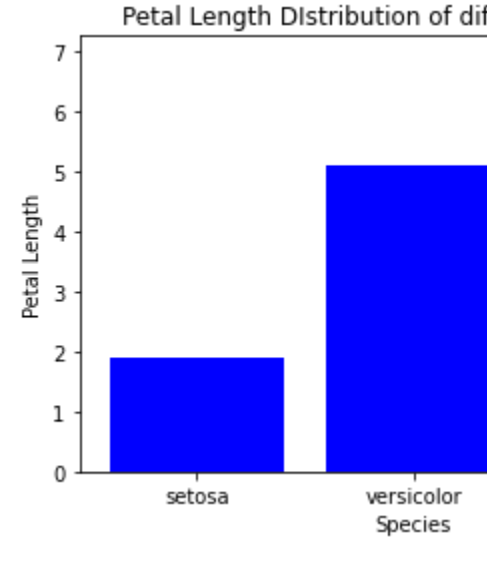
In [35]:
```
plt.bar(iris.species,iris.petal_length,color = "blue")
plt.xlabel("Species")
plt.ylabel("Petal Length")
plt.title("Petal Length DIstribution of different Iris Species")
plt.show()
```



In [ ]:

Pie Chart

In [38]:
```
plt.pie(list(iris.species.value_counts()),
        labels = iris.species.unique(),
        autopct= '%0.3f%%',
        colors = ["purple","pink","yellow"])
plt.show()
```



In [ ]:

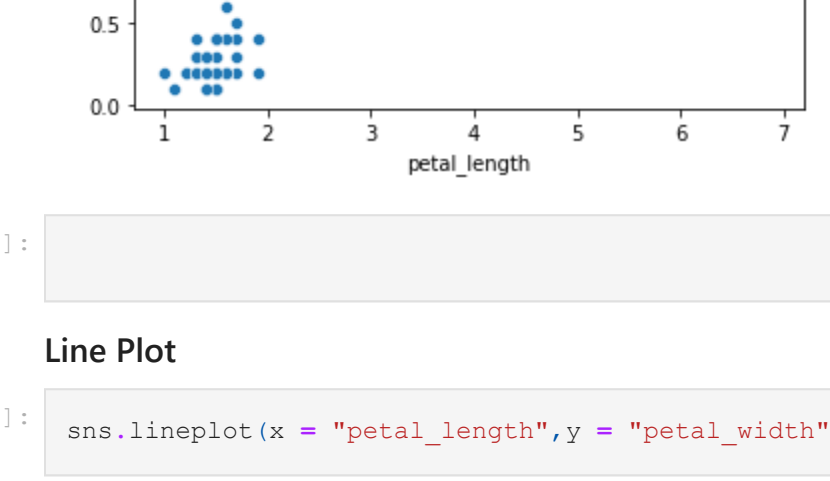
Seaborn Visualizations

In [ ]:

Scatter Plot

In [37]:
```
sns.scatterplot(x="petal_length",y = "petal_width",data = iris)
```

Out[37]: <AxesSubplot:xlabel='petal_length', ylabel='petal_width'>

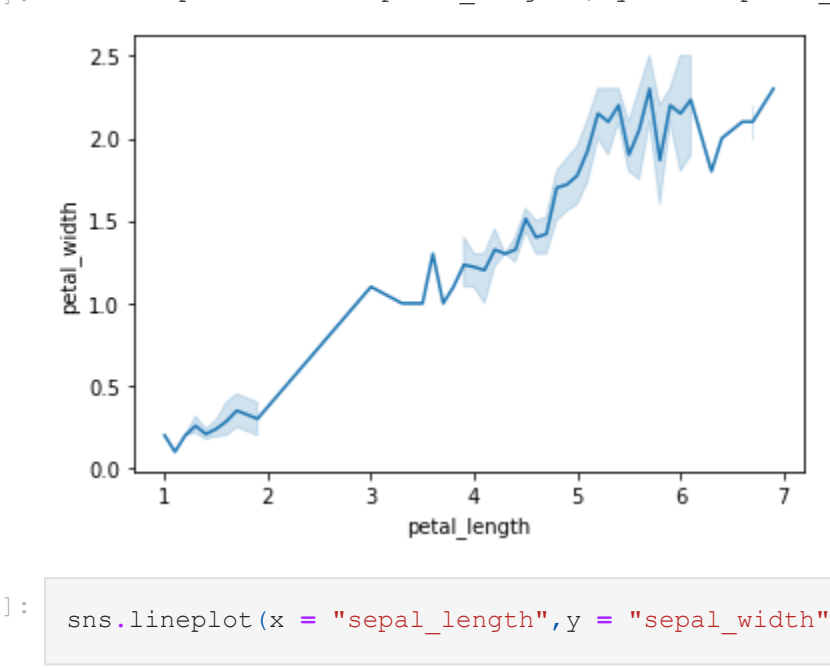


In [ ]:

Line Plot

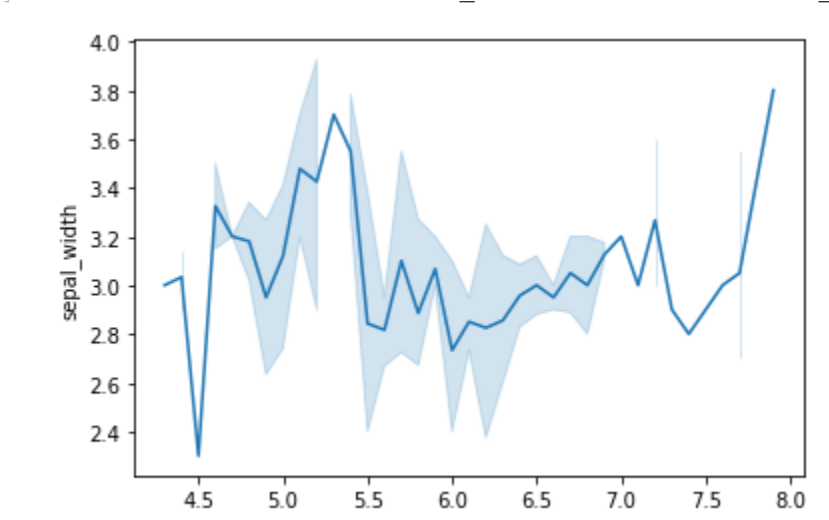
In [5]:
```
sns.lineplot(x = "petal_length",y = "petal_width",data = iris)
```

Out[5]: <AxesSubplot:xlabel='petal_length', ylabel='petal_width'>



In [8]:
```
sns.lineplot(x = "sepal_length",y = "sepal_width",data = iris)
```

Out[8]: <AxesSubplot:xlabel='sepal_length', ylabel='sepal_width'>

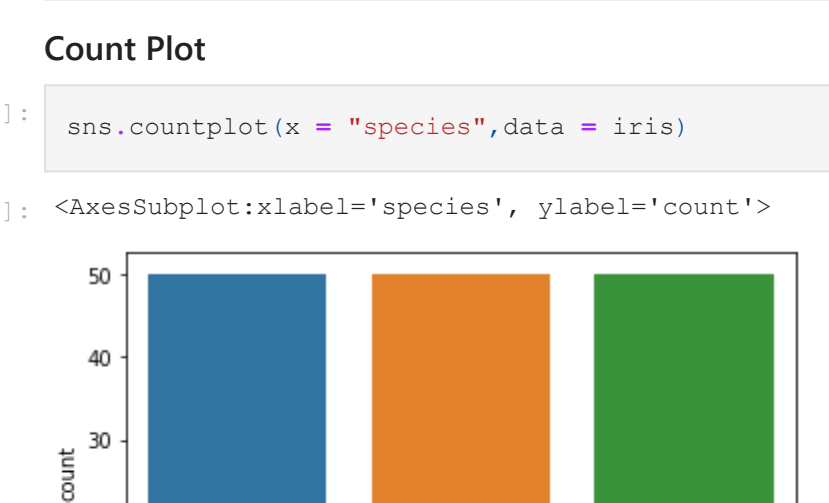


In [ ]:

Count Plot

In [9]:
```
sns.countplot(x = "species",data = iris)
```

Out[9]: <AxesSubplot:xlabel='species', ylabel='count'>

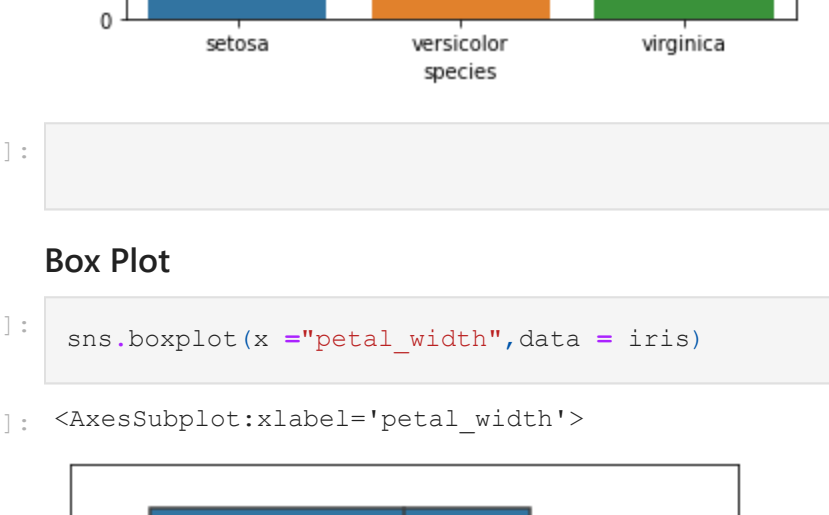


In [ ]:

Box Plot

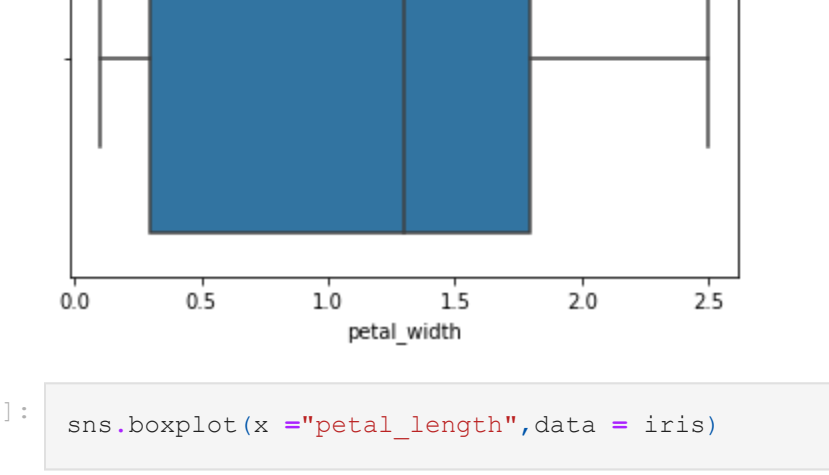
In [12]:
```
sns.boxplot(x ="petal_width",data = iris)
```

Out[12]: <AxesSubplot:xlabel='petal_width'>



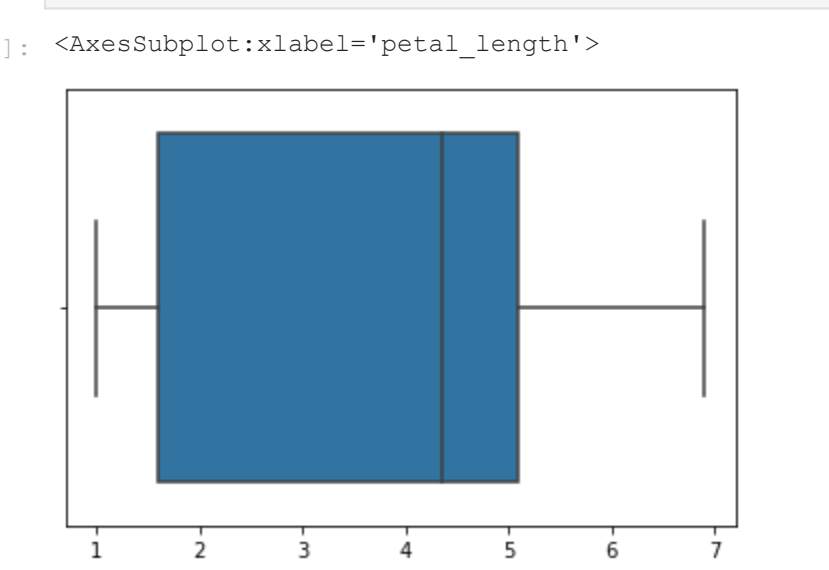
In [13]:
```
sns.boxplot(x ="petal_length",data = iris)
```

Out[13]: <AxesSubplot:xlabel='petal_length'>



In [14]:
```
sns.boxplot(x ="sepal_length",data = iris)
```

Out[14]: <AxesSubplot:xlabel='sepal_length'>



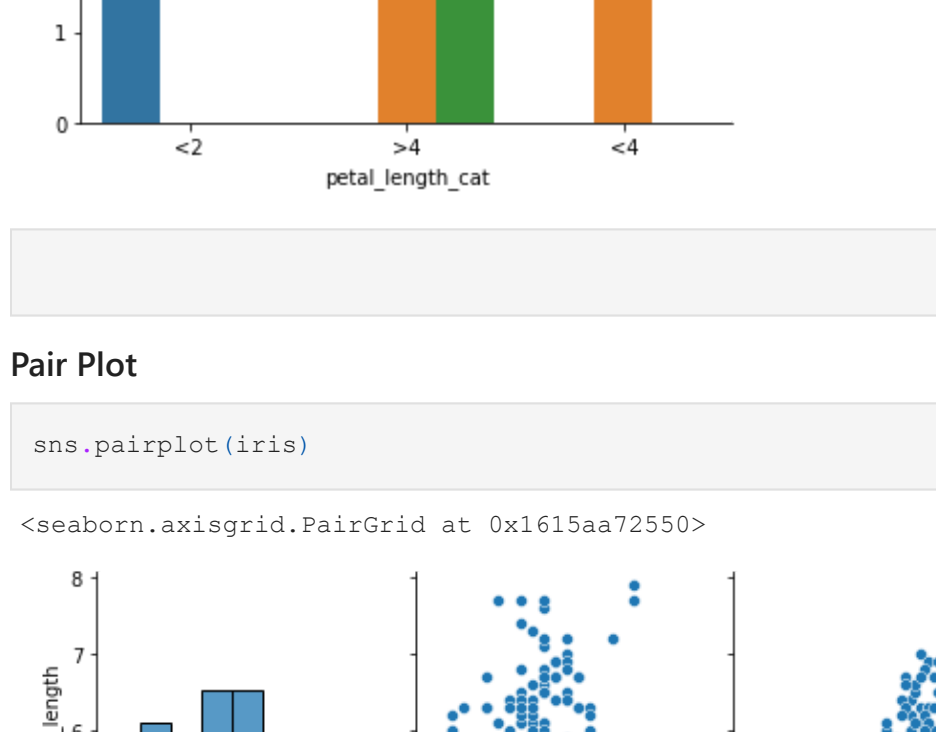
In [ ]:

Categorical Plot

In [27]:
```
iris["petal_length_cat"] = np.where(iris["petal_length"]<2,
                                    "<2",
                                    np.where(iris["petal_length"]<4,
                                             "<4",
                                             ">4"))
```

In [28]:
```
sns.catplot(x="petal_length_cat",
            y= "sepal_length",
            hue = "species",
            kind = "bar",
            data = iris)
```

Out[28]: <seaborn.axisgrid.FacetGrid at 0x1615bb2bd90>

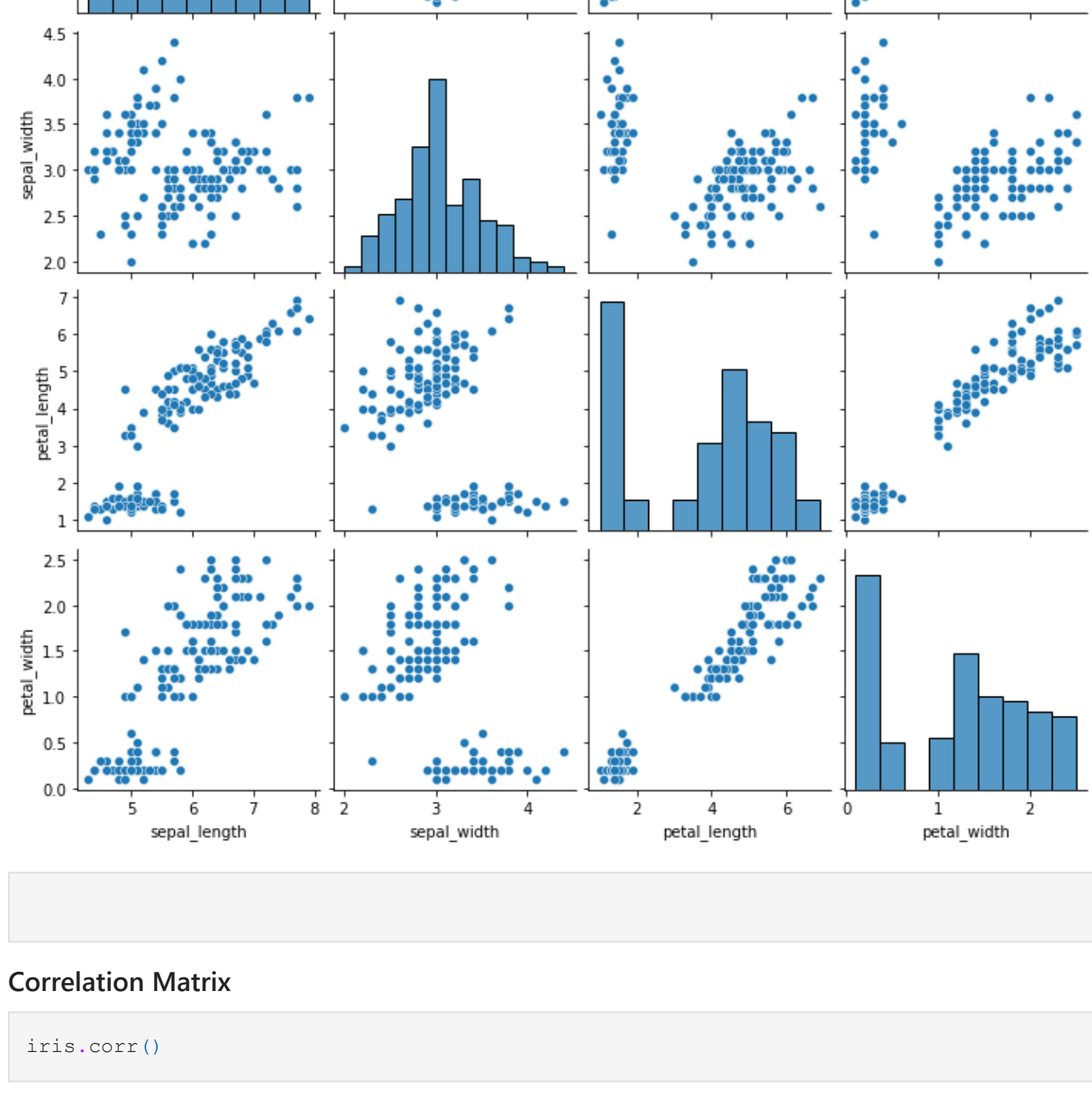


In [ ]:

Pair Plot

In [29]:
```
sns.pairplot(iris)
```

Out[29]: <seaborn.axisgrid.PairGrid at 0x1615aa72550>



In [ ]:

Correlation Matrix

In [30]:
```
iris.corr()
```

Out[30]:

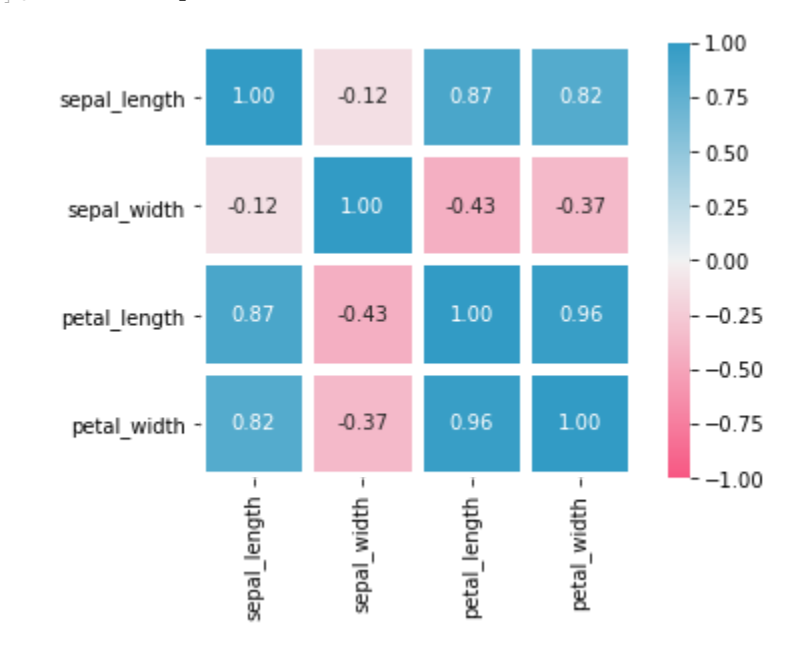
	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

In [ ]:

Heat Map

In [32]:
```
sns.heatmap(iris.corr(),
            annot=True,
            fmt=".2f",
            linewidths=5,
            cmap=sns.diverging_palette(0,
                                       230,
                                       90,
                                       60,
                                       as_cmap=True),
            vmin=-1,
            vmax=1,
            square=True)
```

Out[32]: <AxesSubplot:>



In [ ]: