

```
In [2]: #Implementing required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

In [3]: #reading CSV into dataframe
df = pd.read_csv("/home/anaconda/Downloads/Dataset/iris.csv")
```

```
In [5]: df
```

Out[5]:

	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
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa
10	5.4	3.7	1.5	0.2	setosa
11	4.8	3.4	1.6	0.2	setosa
12	4.8	3.0	1.4	0.1	setosa
13	4.3	3.0	1.1	0.1	setosa
14	5.8	4.0	1.2	0.2	setosa
15	5.7	4.4	1.5	0.4	setosa
16	5.4	3.9	1.3	0.4	setosa
17	5.1	3.5	1.4	0.3	setosa
18	5.7	3.8	1.7	0.3	setosa
19	5.1	3.8	1.5	0.3	setosa
20	5.4	3.4	1.7	0.2	setosa
21	5.1	3.7	1.5	0.4	setosa
22	4.6	3.6	1.0	0.2	setosa
23	5.1	3.3	1.7	0.5	setosa
24	4.8	3.4	1.9	0.2	setosa
25	5.0	3.0	1.6	0.2	setosa
26	5.0	3.4	1.6	0.4	setosa
27	5.2	3.5	1.5	0.2	setosa
28	5.2	3.4	1.4	0.2	setosa
29	4.7	3.2	1.6	0.2	setosa
...
120	6.9	3.2	5.7	2.3	virginica
121	5.6	2.8	4.9	2.0	virginica
122	7.7	2.8	6.7	2.0	virginica
123	6.3	2.7	4.9	1.8	virginica
124	6.7	3.3	5.7	2.1	virginica
125	7.2	3.2	6.0	1.8	virginica
126	6.2	2.8	4.8	1.8	virginica
127	6.1	3.0	4.9	1.8	virginica
128	6.4	2.8	5.6	2.1	virginica
129	7.2	3.0	5.8	1.6	virginica
130	7.4	2.8	6.1	1.9	virginica
131	7.9	3.8	6.4	2.0	virginica
132	6.4	2.8	5.6	2.2	virginica
133	6.3	2.8	5.1	1.5	virginica
134	6.1	2.6	5.6	1.4	virginica
135	7.7	3.0	6.1	2.3	virginica
136	6.3	3.4	5.6	2.4	virginica
137	6.4	3.1	5.5	1.8	virginica
138	6.0	3.0	4.8	1.8	virginica
139	6.9	3.1	5.4	2.1	virginica
140	6.7	3.1	5.6	2.4	virginica
141	6.9	3.1	5.1	2.3	virginica
142	5.8	2.7	5.1	1.9	virginica
143	6.8	3.2	5.9	2.3	virginica
144	6.7	3.3	5.7	2.5	virginica
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 x 5 columns

```
In [6]: df.shape
```

Out[6]: (150, 5)

```
In [7]: df.columns
```

Out[7]: Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'], dtype='object')

```
In [9]: df.info()
```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
sepal_length 150 non-null float64
sepal_width 150 non-null float64
petal_length 150 non-null float64
petal_width 150 non-null float64
species 150 non-null object
dtypes: float64(4), object(1)
memory usage: 5.9+ KB

```
In [11]: df.dtypes
```

Out[11]: sepal_length float64
sepal_width float64
petal_length float64
petal_width float64
species object
dtype: object

```
In [8]: df.describe()
```

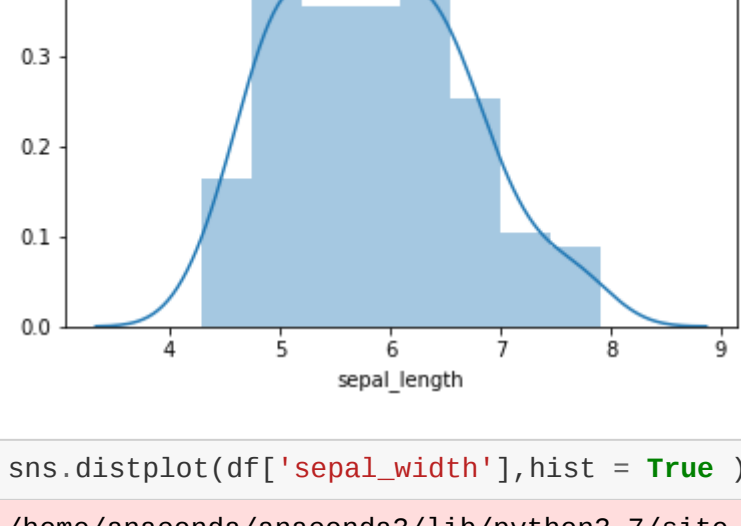
Out[8]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
In [12]: #Data Visualisation
sns.distplot(df['sepal_length'],hist = True )
```

/home/anaconda/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use 'arr[tuple(seq)]' instead of 'arr[seq]'. In the future this will be interpreted as an array index, 'arr[np.array(seq)]', which will result either in an error or a different result.
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

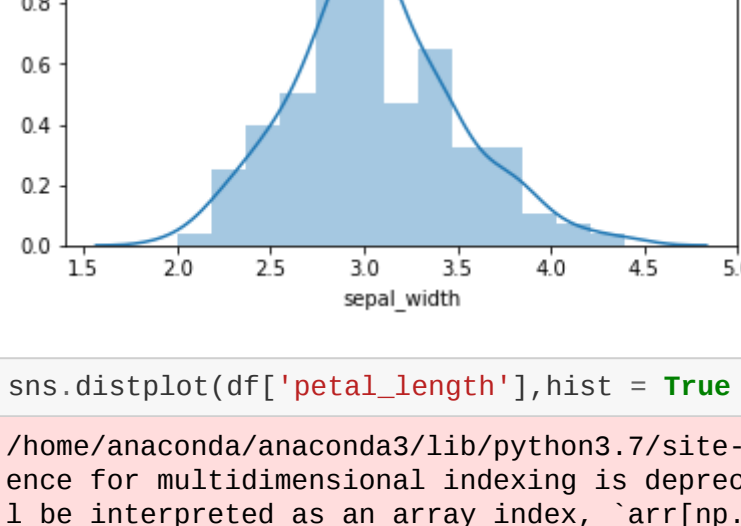
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc3803fc048>



```
In [13]: sns.distplot(df['sepal_width'],hist = True )
```

/home/anaconda/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use 'arr[tuple(seq)]' instead of 'arr[seq]'. In the future this will be interpreted as an array index, 'arr[np.array(seq)]', which will result either in an error or a different result.
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

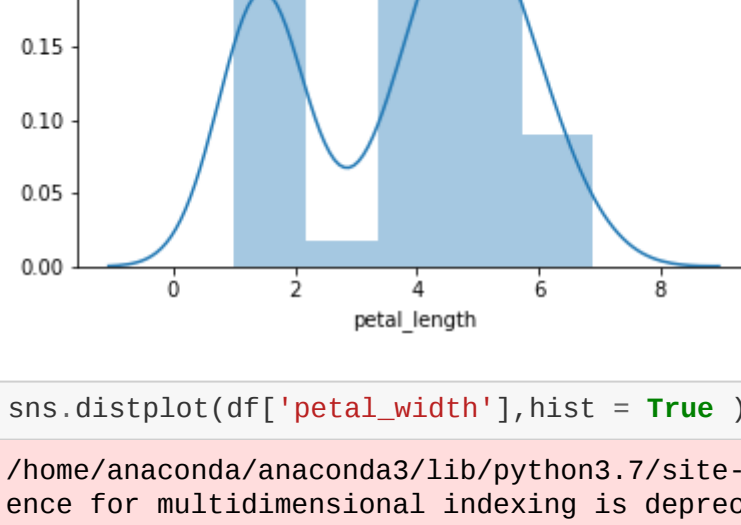
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc380053358>



```
In [14]: sns.distplot(df['petal_length'],hist = True )
```

/home/anaconda/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use 'arr[tuple(seq)]' instead of 'arr[seq]'. In the future this will be interpreted as an array index, 'arr[np.array(seq)]', which will result either in an error or a different result.
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

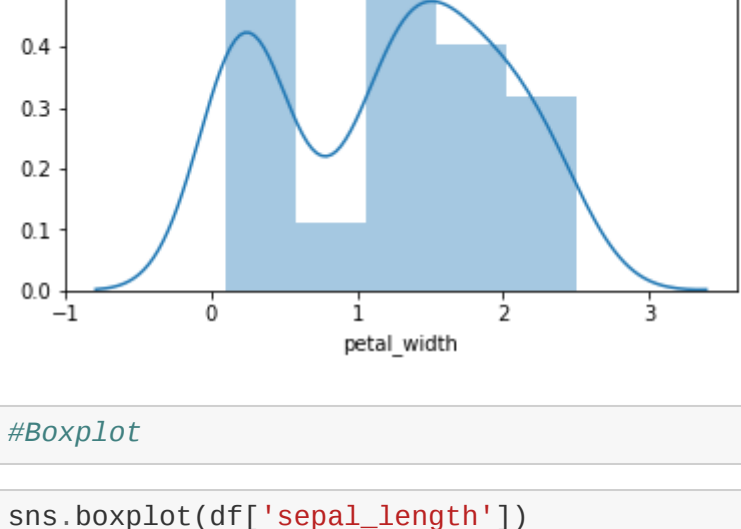
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc37ffe9cf8>



```
In [15]: sns.distplot(df['petal_width'],hist = True )
```

/home/anaconda/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use 'arr[tuple(seq)]' instead of 'arr[seq]'. In the future this will be interpreted as an array index, 'arr[np.array(seq)]', which will result either in an error or a different result.
return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

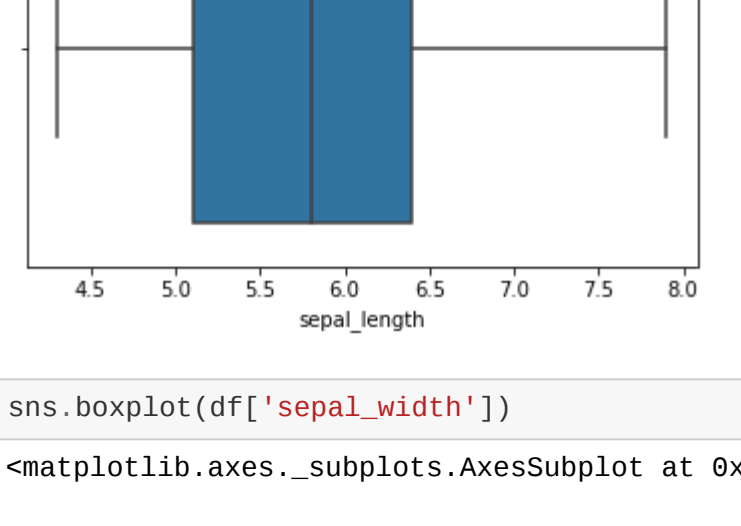
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc37ff43e10>



```
In [ ]: #BoxPlot
```

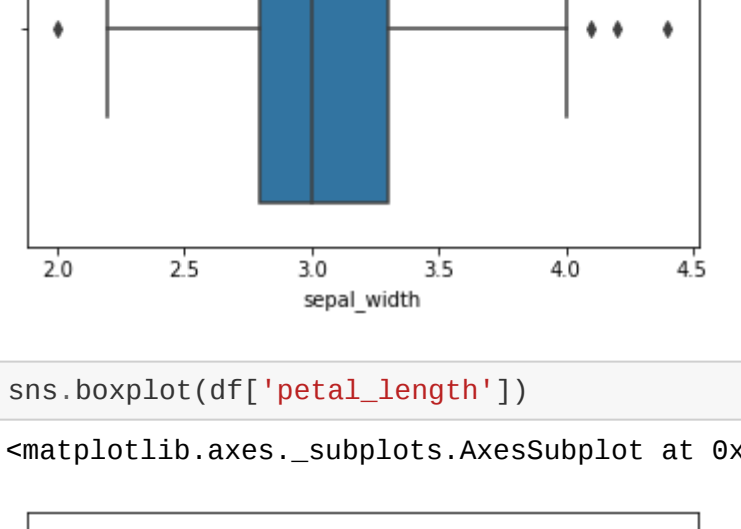
```
In [17]: sns.boxplot(df['sepal_length'])
```

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc37fe8ce80>



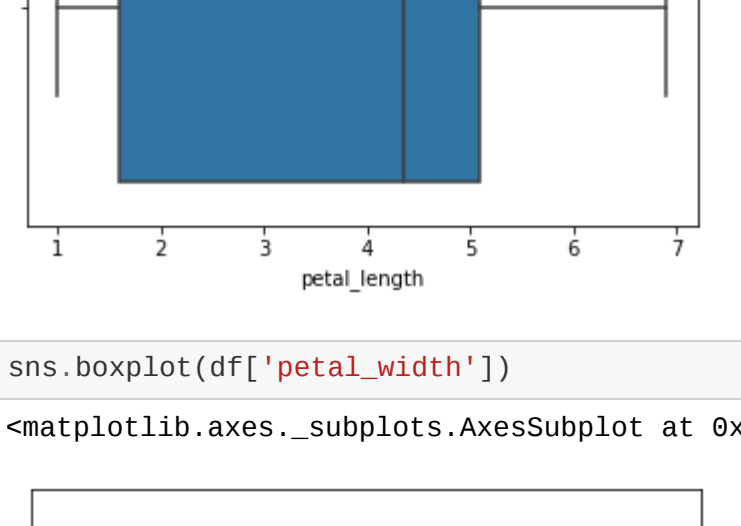
```
In [18]: sns.boxplot(df['sepal_width'])
```

Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc37fe5eac8>



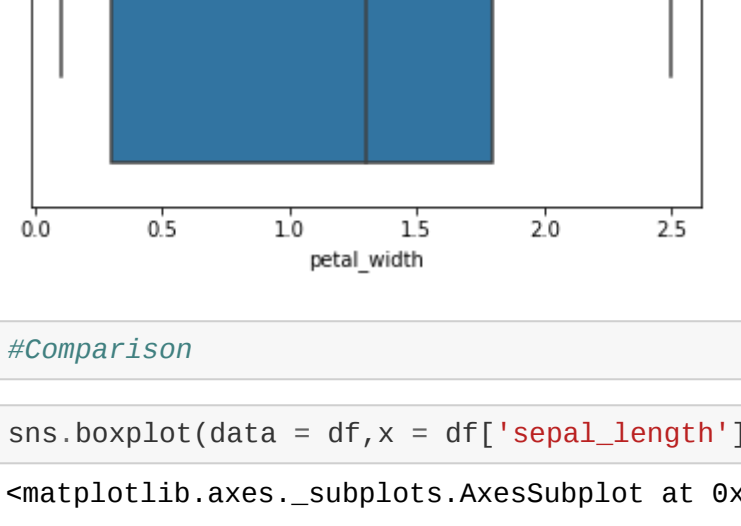
```
In [19]: sns.boxplot(df['petal_length'])
```

Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc37fd40240>



```
In [20]: sns.boxplot(df['petal_width'])
```

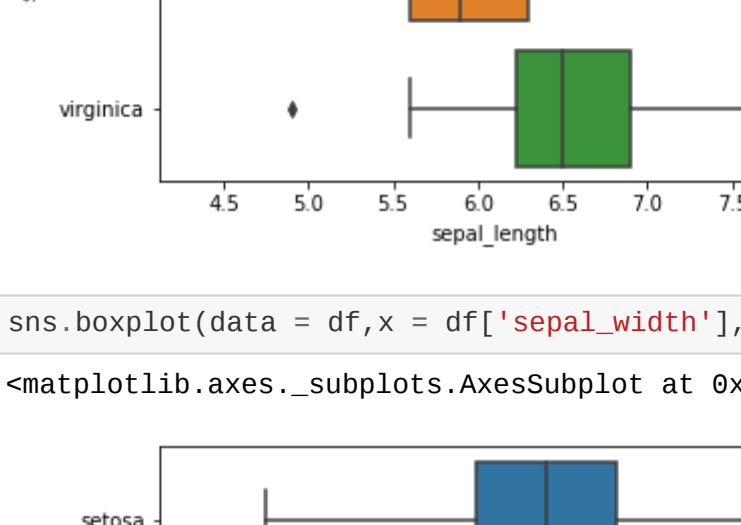
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc37fd8ac88>



```
In [21]: #Comparison
```

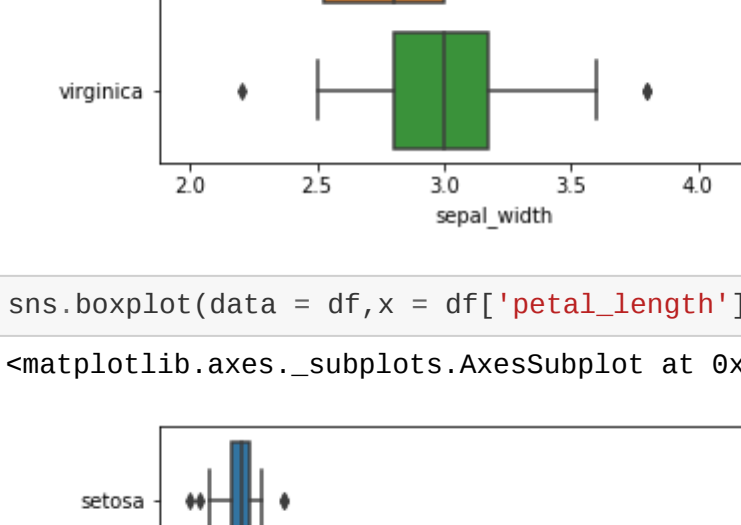
```
In [22]: sns.boxplot(data = df,x = df['sepal_length'],y = df['species'])
```

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc37fd78e10>



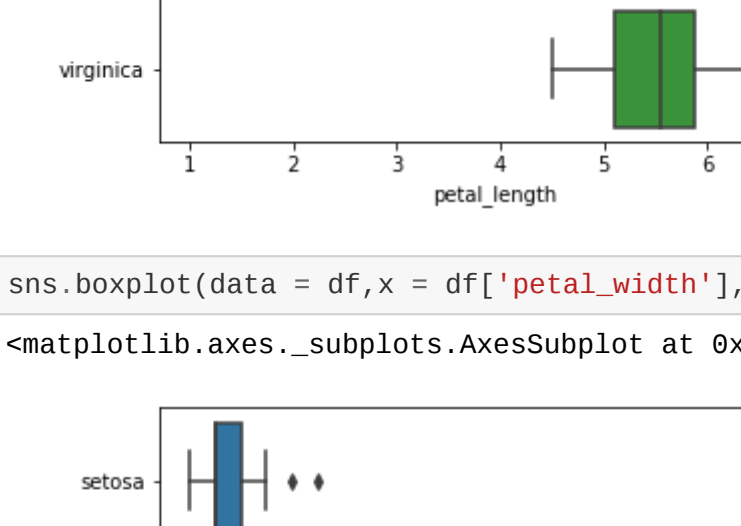
```
In [24]: sns.boxplot(data = df,x = df['sepal_width'],y = df['species'])
```

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc37fd0da08>



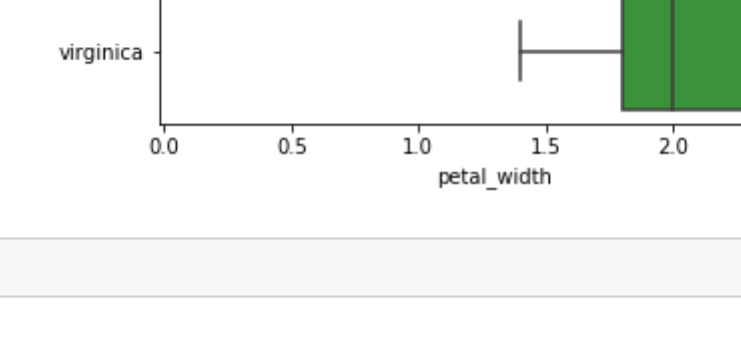
```
In [25]: sns.boxplot(data = df,x = df['petal_length'],y = df['species'])
```

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc37fd18630>



```
In [26]: sns.boxplot(data = df,x = df['petal_width'],y = df['species'])
```

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc37fd90f28>



```
In [ ]: 
```