

# IRIS FLOWER CLASSIFICATION USING ML

DONE BY

S.IRFAAN MUSTAFA

## DATA ANALYSIS USING PANDAS

Importing pandas

```
In [3]: import pandas as pd
```

Reading the csv file

```
In [4]: df = pd.read_csv("Downloads/iris.csv.csv")
```

Printing the first five rows

```
In [5]: df.head()
```

	sepalength	sepalwidth	petallength	petalwidth	class
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

To check the number of rows and columns

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

```
Out[7]: (150, 5)
```

To check the information about the dataset

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

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

To check the brief description of the dataset

```
In [39]: df.describe()
```

	sepalength	sepalwidth	petallength	petalwidth	
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	

Randomly select a row from the Dataframe

```
In [43]: df.sample(n=2,axis=0)
```

	sepalength	sepalwidth	petallength	petalwidth	class
9	4.9	3.1	1.5	0.1	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa

To check the data for any missing values

```
In [10]: df.isnull()
```

	sepalength	sepalwidth	petallength	petalwidth	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
...	...	...	...	...	...
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	False	False	False	False	False

150 rows x 5 columns

To check the missing datavalues and return then in form of numbers

```
In [12]: df.isnull().sum()
```

```
Out[12]: sepalength    0
sepalwidth    0
petallength    0
petalwidth    0
class          0
dtype: int64
```

To check if duplicate data values are present

```
In [26]: ndf = df.drop_duplicates()
ndf
```

	sepalength	sepalwidth	petallength	petalwidth	class
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

147 rows x 5 columns

To count the no of values present in any column

```
In [27]: ndf.value_counts("class")
```

```
Out[27]: class
Iris-versicolor    59
Iris-virginica     49
Iris-setosa        48
dtype: int64
```

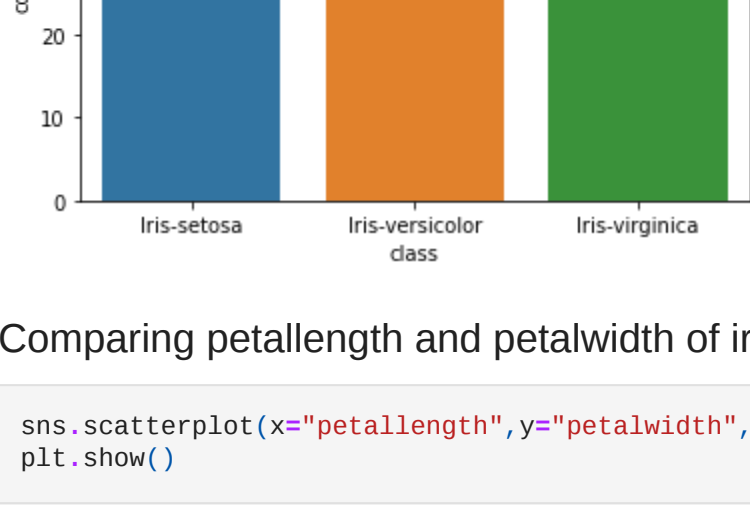
## DATA VISUALIZATION

Importing seaborn and matplotlib

```
In [29]: import seaborn as sns
import matplotlib.pyplot as plt
```

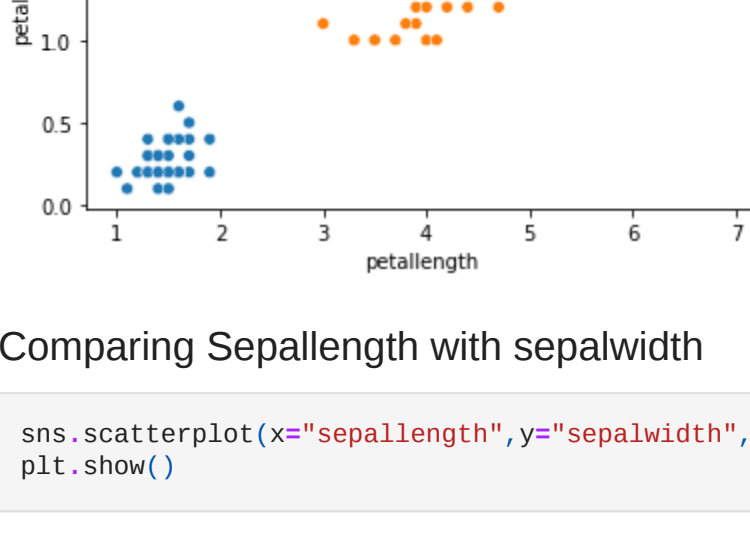
Checking the count of any column present in the dataset

```
In [32]: sns.countplot(x="class", data=ndf)
plt.show()
```



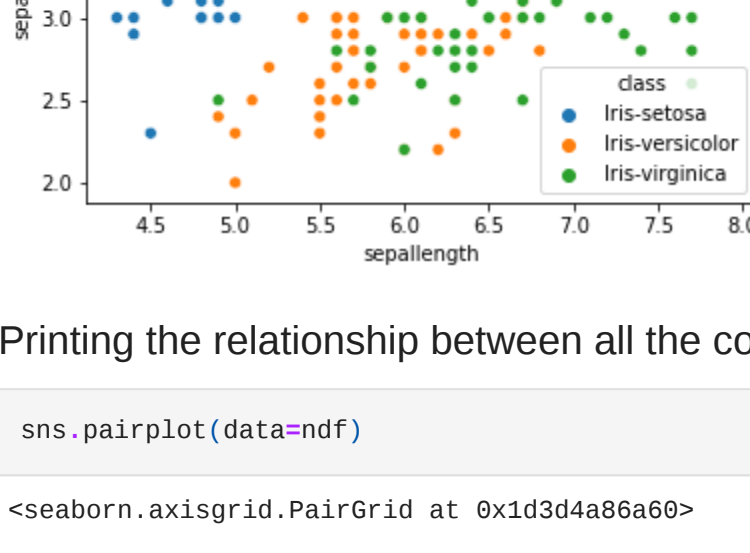
Comparing petallength and petalwidth of iris flowers

```
In [32]: sns.scatterplot(x="petallength",y="petalwidth",hue="class",data=ndf)
plt.show()
```



Comparing Sepalength with sepalwidth

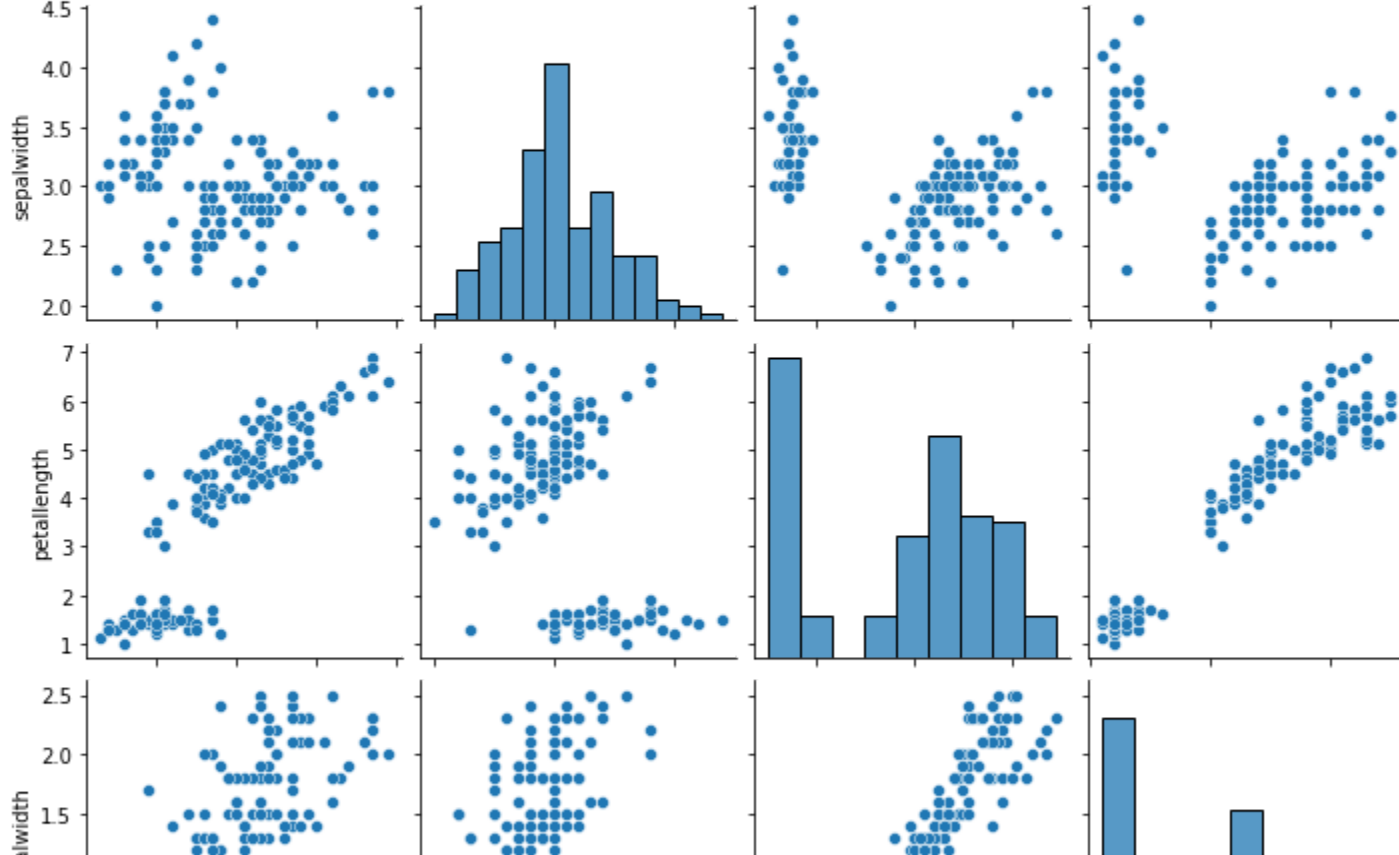
```
In [43]: sns.scatterplot(x="sepalength",y="sepalwidth",hue="class",data=ndf)
plt.show()
```



Printing the relationship between all the columns

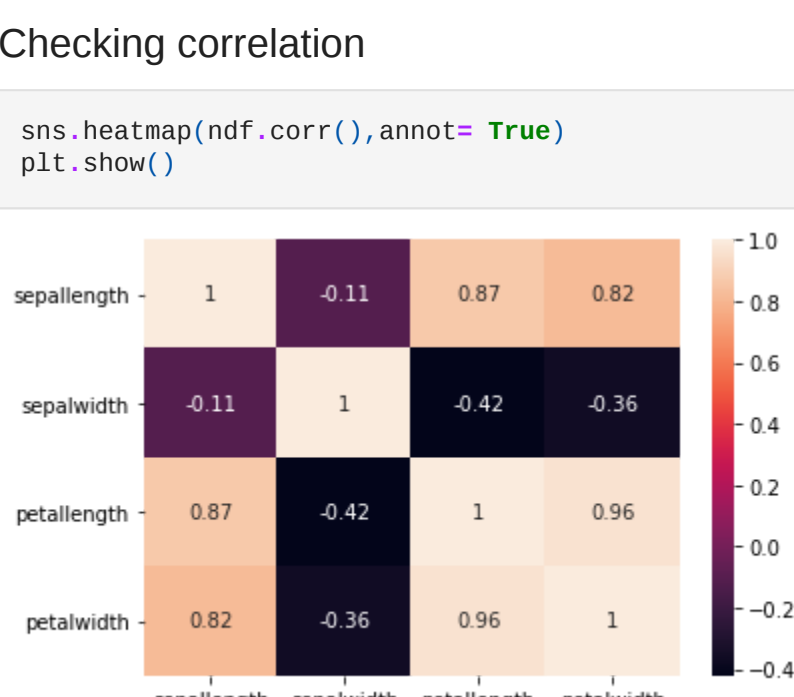
```
In [44]: sns.pairplot(data=ndf)
```

```
Out[44]: <seaborn.axisgrid.PairGrid at 0x1d3d4a86a60>
```



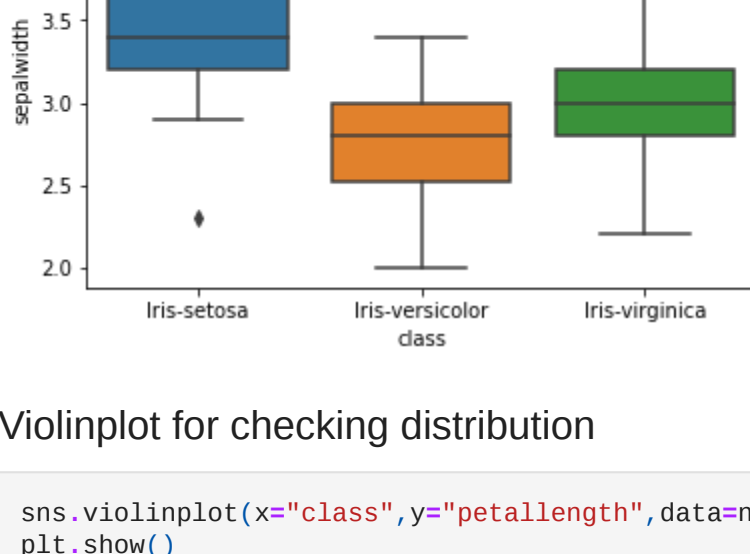
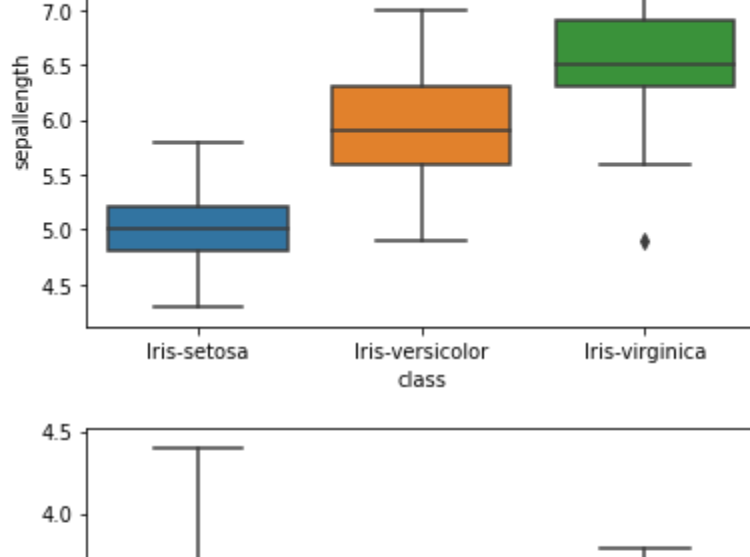
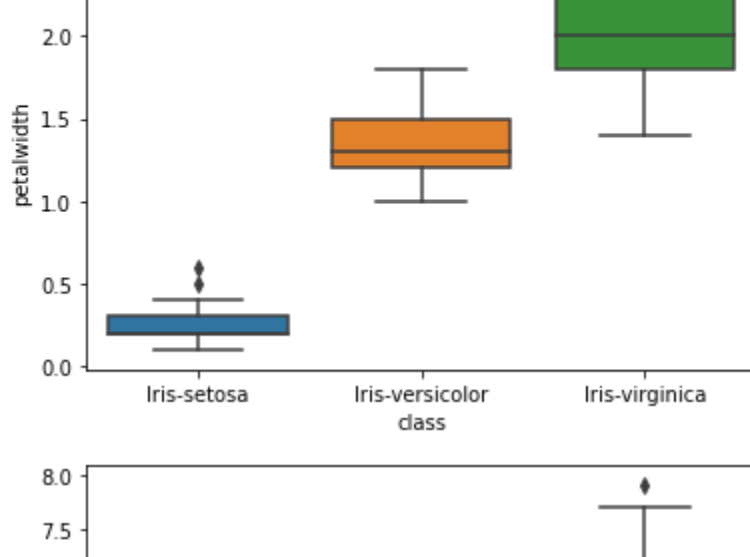
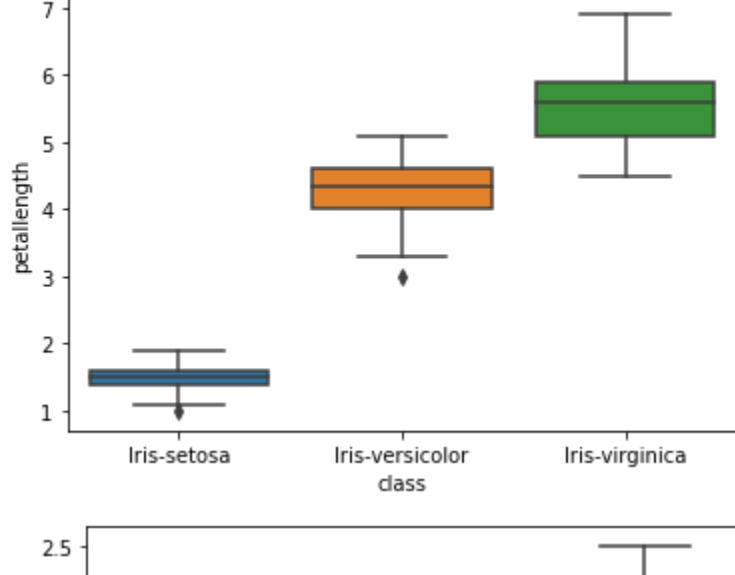
Checking correlation

```
In [50]: sns.heatmap(ndf.corr(),annot=True)
plt.show()
```



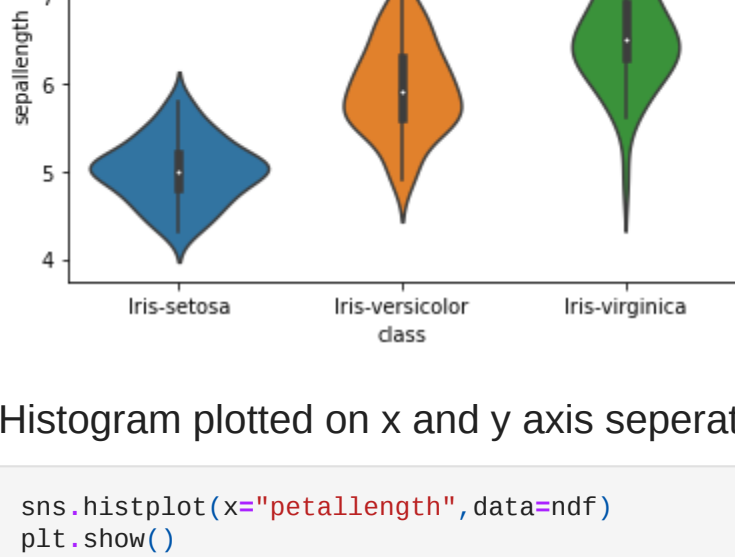
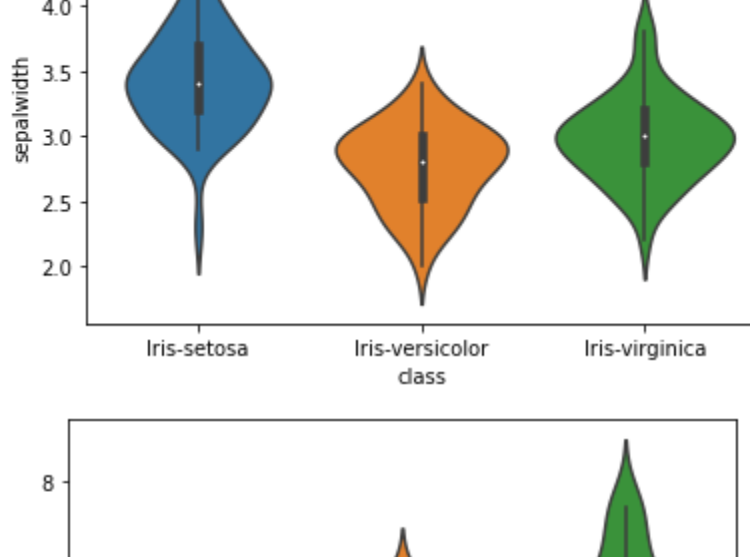
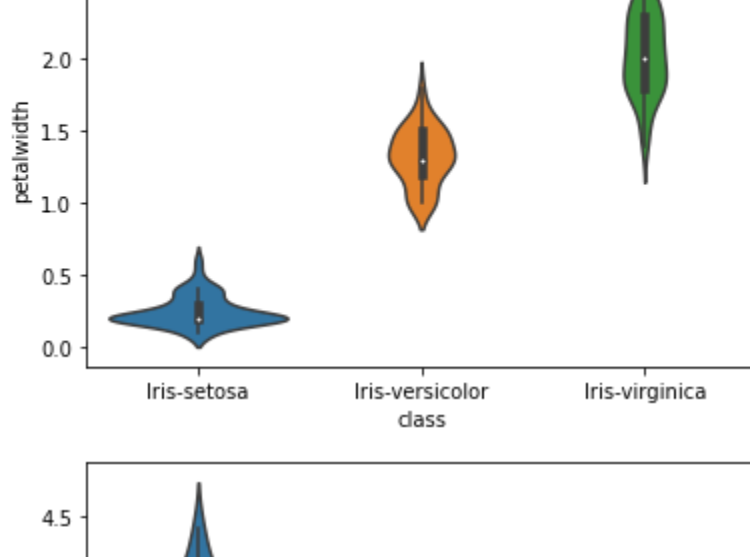
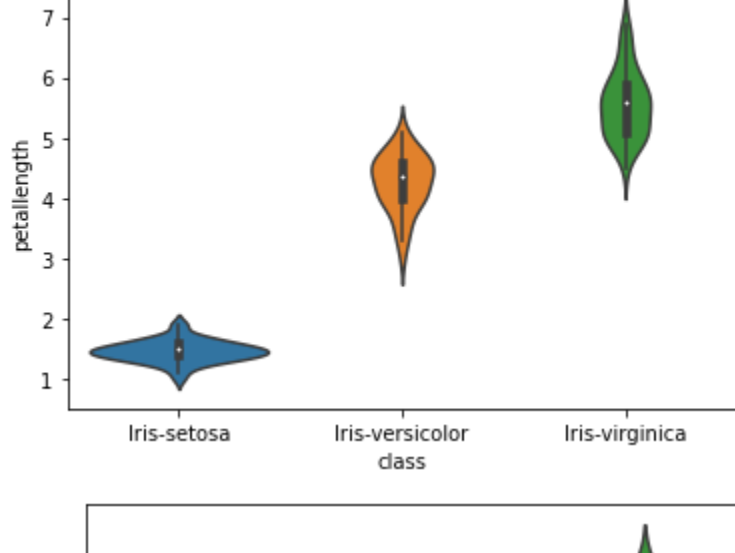
Boxplot to know about the distribution

```
In [59]: sns.boxplot(x="class",y="petallength",data=ndf)
plt.show()
sns.boxplot(x="class",y="petalwidth",data=ndf)
plt.show()
sns.boxplot(x="class",y="sepalength",data=ndf)
plt.show()
sns.boxplot(x="class",y="sepalwidth",data=ndf)
plt.show()
```



Violinplot for checking distribution

```
In [50]: sns.violinplot(x="class",y="petallength",data=ndf)
plt.show()
sns.violinplot(x="class",y="petalwidth",data=ndf)
plt.show()
sns.violinplot(x="class",y="sepalength",data=ndf)
plt.show()
sns.violinplot(x="class",y="sepalwidth",data=ndf)
plt.show()
```



Histogram plotted on x and y axis seperately on different data

```
In [63]: sns.histplot(x="petallength",data=ndf)
plt.show()
sns.histplot(y="petalwidth",data=ndf)
plt.show()
sns.histplot(x="sepalength",data=ndf)
plt.show()
sns.histplot(y="sepalwidth",data=ndf)
plt.show()
```

