Data Visualization Techniques

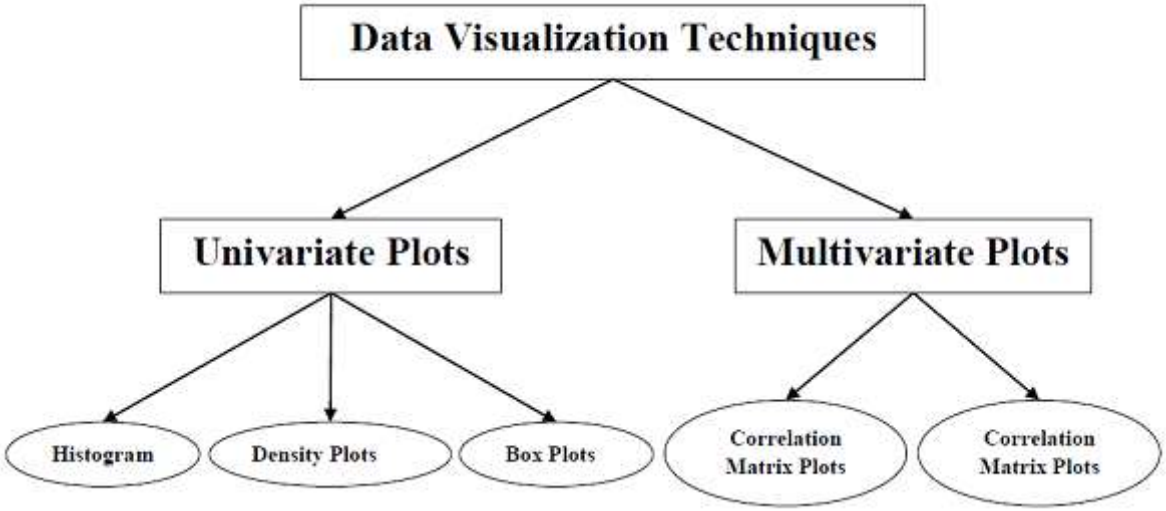
1. Different Types of Analysis for Data Visualization

- Mainly, there are three different types of analysis for Data Visualization:

  Univariate Analysis: In the univariate analysis, we will be using a single feature to analyze almost all of its properties.

  Bivariate Analysis: When we compare the data between exactly 2 features then it is known as bivariate analysis.

  Multivariate Analysis: In the multivariate analysis, we will be comparing more than 2 variables.



2. Import Libraries

First import basic libraries like **numpy** and **pandas** and Python data visualization libraries like **matplotlib** and **seaborn**.

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

import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

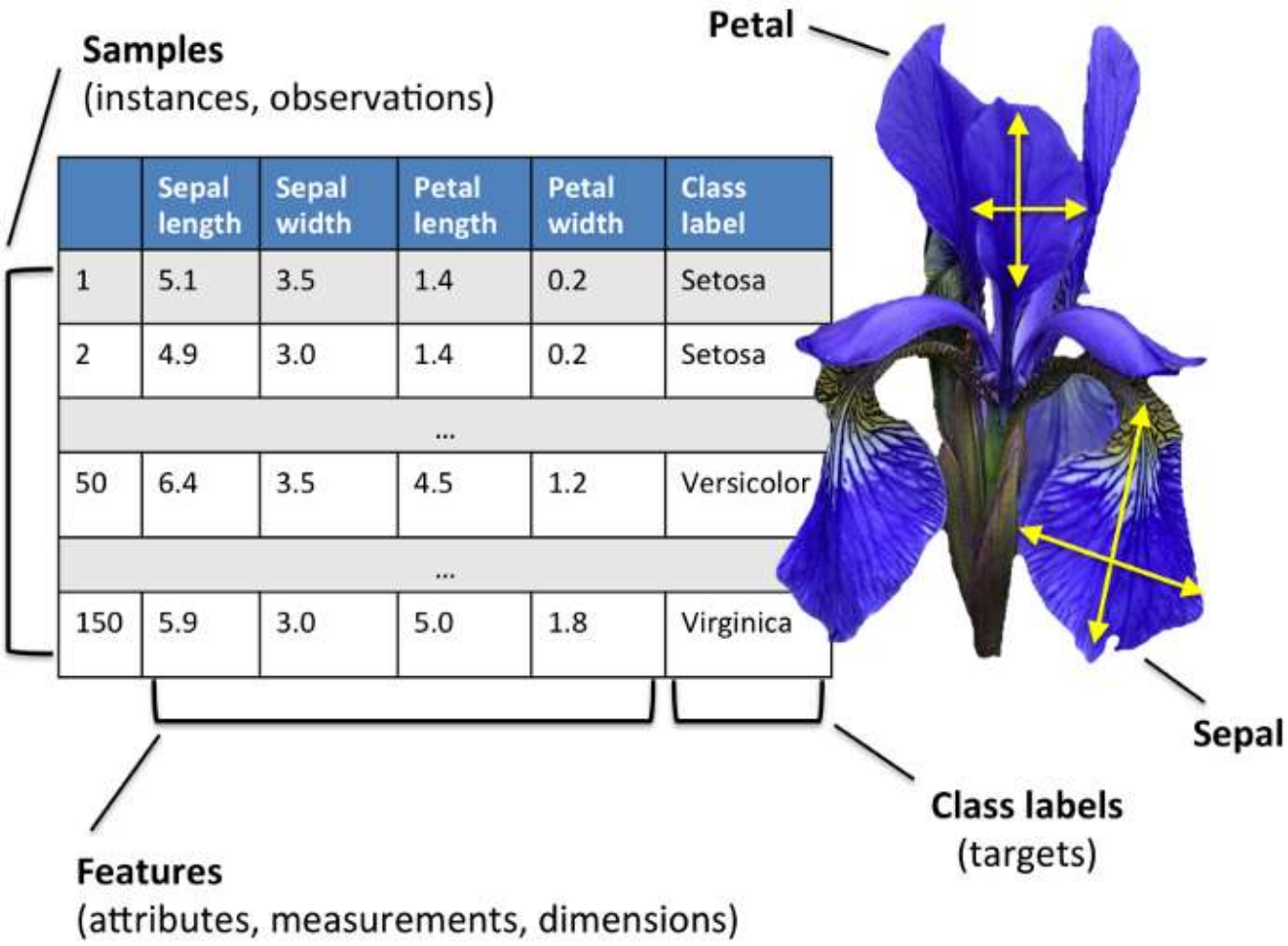
3. Understanding the Dataset

Next, load the data set from sklearn libraries:

In [8]:
```
data = pd.read_csv(r"C:\Users\Vivek 6666\Downloads\iris.csv")
data.head()
```

Out[8]:

	sepal.length	sepal.width	petal.length	petal.width	variety
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



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

3.1 Gaining information from data

In [4]:
```
data.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   variety       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Observations:

1. All columns are not having any Null Entries
2. Four columns are a numerical type
3. Only Single column categorical type

In [14]:
```
print(data.shape) #print number of rows and columns
```

```
(150, 5)
```

3.2 Statistical Insight

In [5]:
```
data.describe()
```

Out[5]:

	sepal.length	sepal.width	petal.length	petal.width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
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

Data Insights:

- Mean values
- Standard Deviation ,
- Minimum Values
- Maximum Values

Observations:

- So From this statistical information, we can conclude that there are a total of **150** data points and data is distributed among **3** species equally.
- So, we can say this is a balanced dataset.

3.3 Checking For Duplicate Entries

In [7]:
```
data[data.duplicated()]
```

Out[7]:

	sepal.length	sepal.width	petal.length	petal.width	variety
142	5.8	2.7	5.1	1.9	Virginica

In [16]:
```
print(data['variety'].value_counts()) # Counts of every unique Species value
```

```
Virginica     50
Setosa        50
Versicolor    50
Name: variety, dtype: int64
```

Therefore we shouldn't delete the entries as it might imbalance the data sets and hence will prove to be less useful for valuable insights

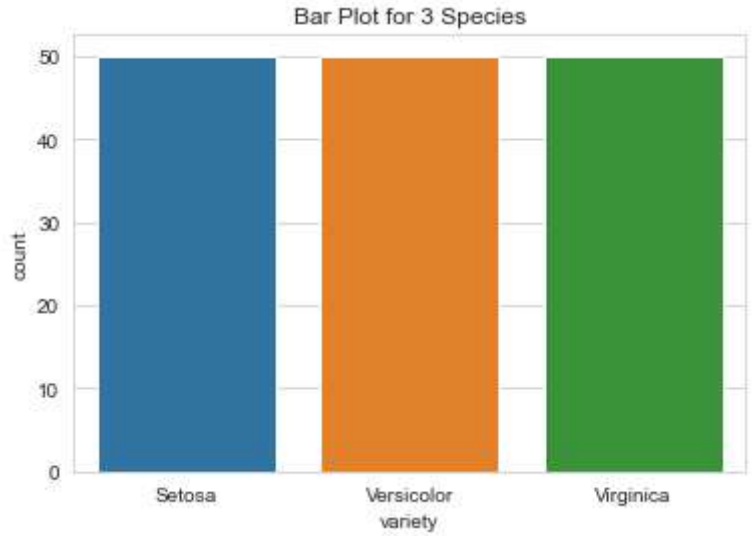
4. Data Visualization

Species count

4.1 Bar Plot

- A bar plot is a plot that presents categorical data with rectangular bars.
- We can count the values of various categories using bar plots.
- Here, the frequency of the observation is plotted.
- In this case, we are plotting the frequency of the three species in the Iris Dataset

In [83]:
```
sns.countplot('variety',data=data)
plt.title('Bar Plot for 3 Species')
plt.show()
```



Observations:

- All bars are of the same height as we know their frequencies are equal.
- Because the Iris Dataset is balanced.

Data Insight :

- This further visualizes that species are well balanced
- Each species (Iris virginica, setosa, versicolor) has 50 as it’s count

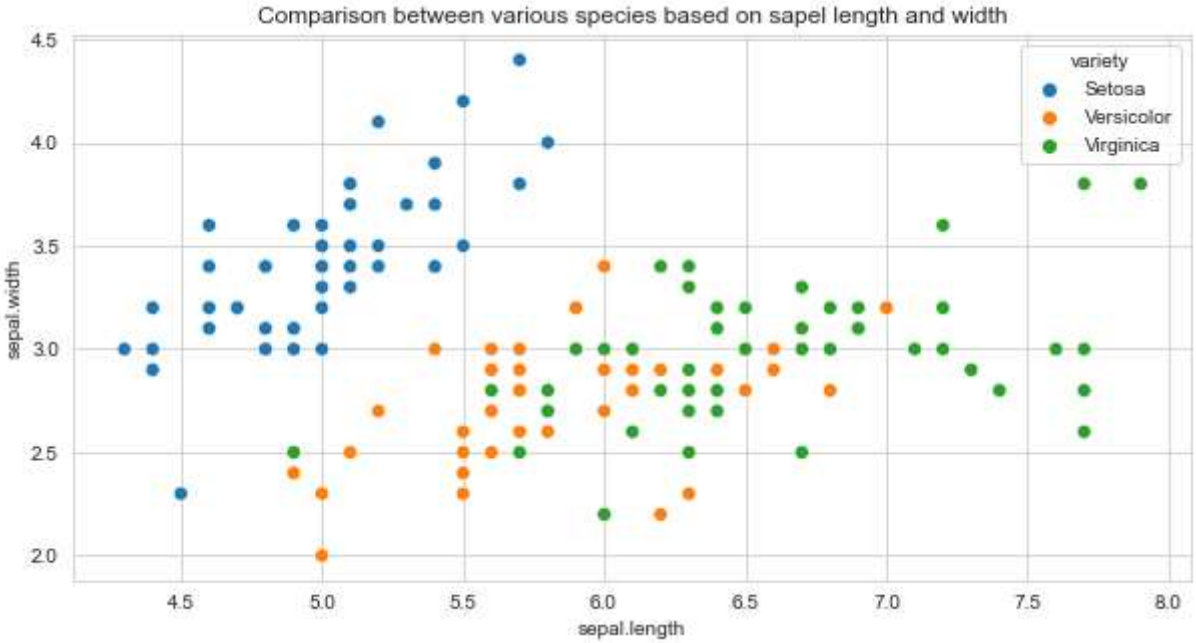
4.2 Scatter Plots

- Scatter plots can be leveraged to identify relationships between two variables.
- It can be effectively used in circumstances where the `dependent` variable can have multiple values for the `independent` variable.

4.2.1 Comparison between various species based on sepal length and width

In [84]:
```
plt.figure(figsize=(10,5))
plt.title('Comparison between various species based on sapel length and width')
sns.scatterplot(data['sepal.length'],data['sepal.width'],hue =data['variety'],s=50)
```

Out[84]:
```
<AxesSubplot:title={'center':'Comparison between various species based on sapel length and width'}, xlabel='se
pal.length', ylabel='sepal.width'>
```



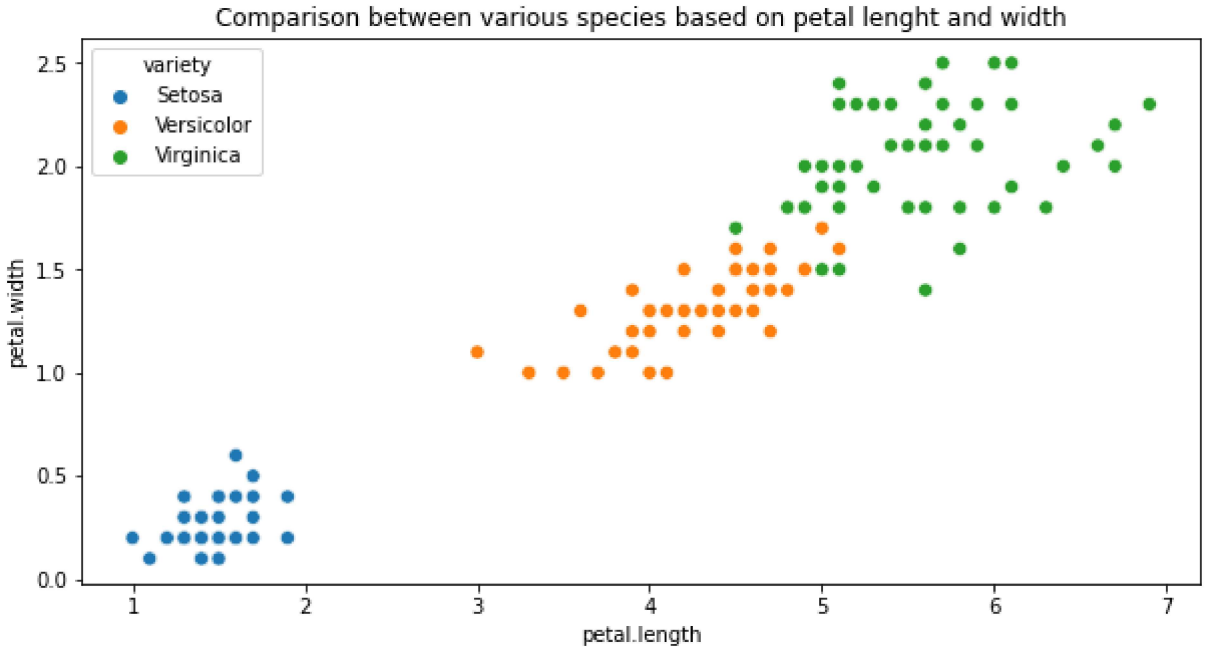
Observations:

- Setosa species have a smaller sepal length but higher width.
- Versicolor lies in almost middle for length as well as width
- Virginica has larger sepal lengths and smaller sepal widths

4.2.2 Comparison between various species based on petal length and width

In [10]:
```
plt.figure(figsize=(10,5))
plt.title('Comparison between various species based on petal lenght and width')
sns.scatterplot(data['petal.length'], data['petal.width'], hue = data['variety'], s= 50)
```

Out[10]:
```
<AxesSubplot:title={'center':'Comparison between various species based on petal lenght and width'}, xlabel='pe
tal.length', ylabel='petal.width'>
```



Observations:

1. Setosa species have the smallest petal length as well as petal width
2. Versicolor species have average petal length and petal width
3. Virginica species have the highest petal length as well as petal width

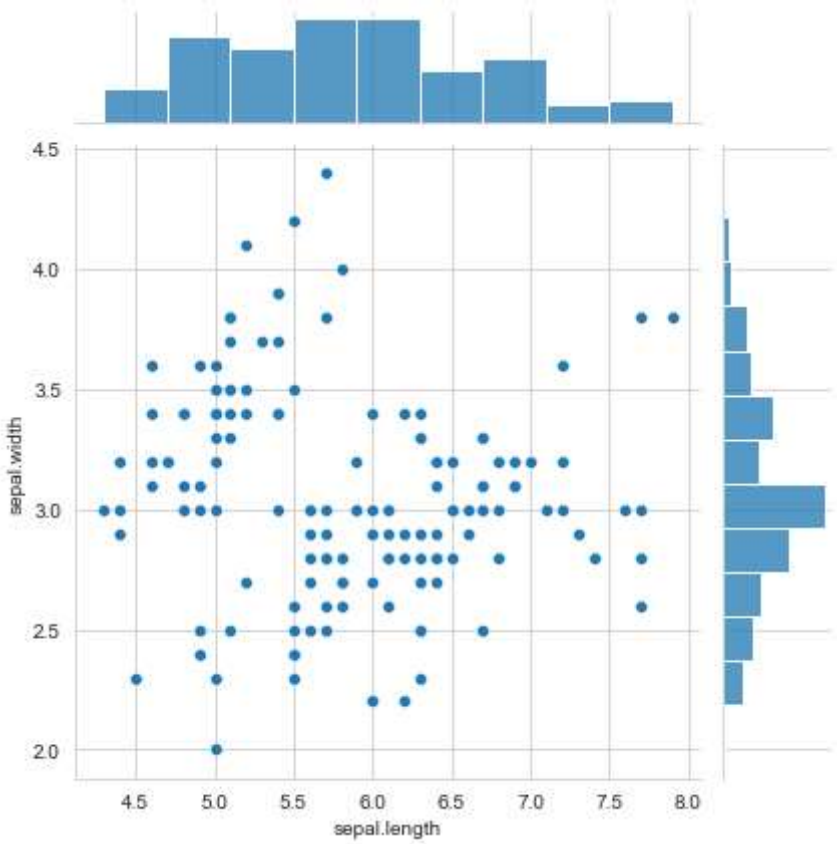
4.3 Joint plot:

- Jointplot is used to quickly visualize and analyze the relationship between two variables and describe their individual distributions on the same plot. And
- jointplot shows bivariate scatterplots and univariate histograms in the same figure.
- So as you can see this Joint plot represent SepalLength and Sepal Width attribute in the Iris dataset

In [85]:
```
sns.jointplot(x="sepal.length", y="sepal.width", data=data, size=6)
```

```
D:\Anaconda3-2020.11-Windows-x86_64\lib\site-packages\seaborn\axisgrid.py:2015: UserWarning: The `size` parame
ter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

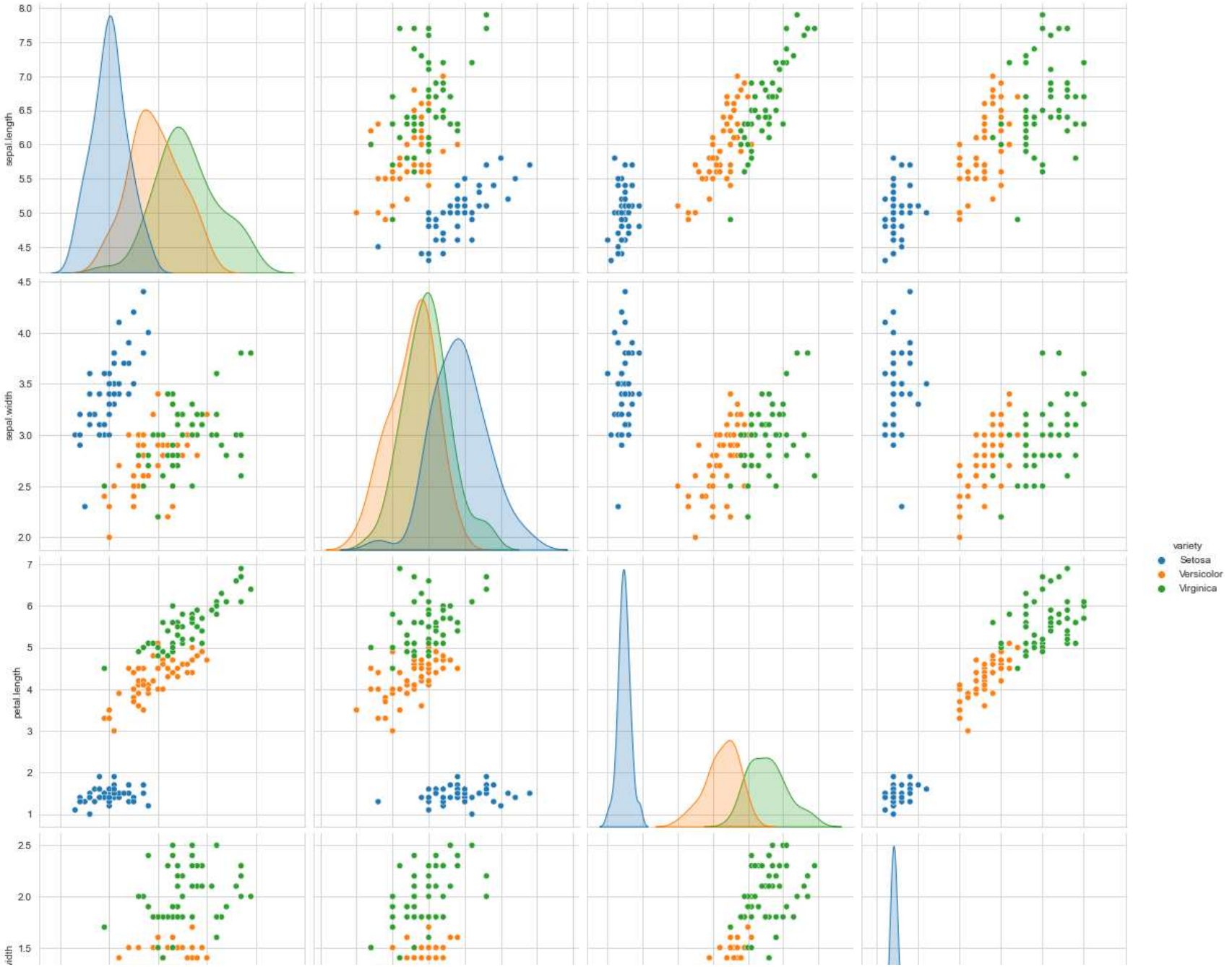
Out[85]:
```
<seaborn.axisgrid.JointGrid at 0x22f4aa90b20>
```

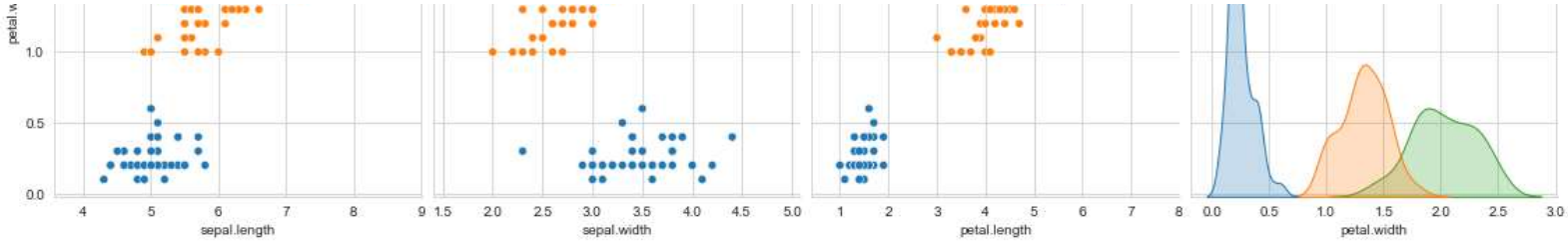


4.4 Pair plot

In [48]:
```
sns.pairplot(data,hue='variety',height=4)
```

Out[48]:
```
<seaborn.axisgrid.PairGrid at 0x22f4892fd90>
```





Observations

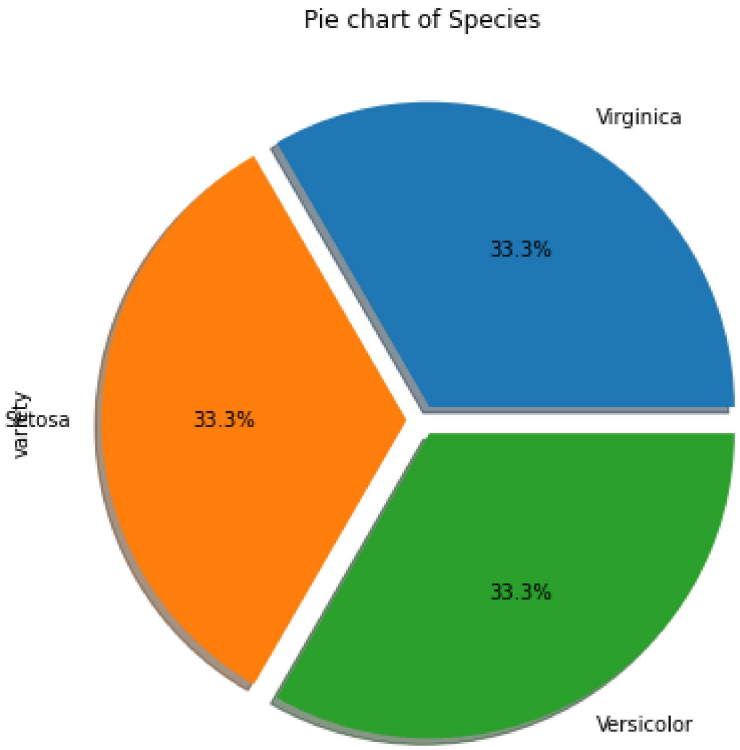
This pair plot indicates that:

- There is a High correlation between petal length and width columns.
- Setosa has both low petal length and width
- Versicolor has both average petal length and width
- Virginica has both high petal length and width.
- Sepal width for setosa is high and the length is low.
- Versicolor has average values for sepal dimensions.
- Virginica has small width but a large sepal length

4.5 Pie Chart

- Pie Chart is a circular chart that uses pie slices to show the relative size of data.
- The arc length of each pie slice is proportional to the quantity it represents.

In [19]:
```
data['variety'].value_counts().plot.pie(explode=[0.05,0.05,0.05],autopct='%1.1f%%',shadow=True,figsize=(7,7))
plt.title("Pie chart of Species")
plt.show()
```



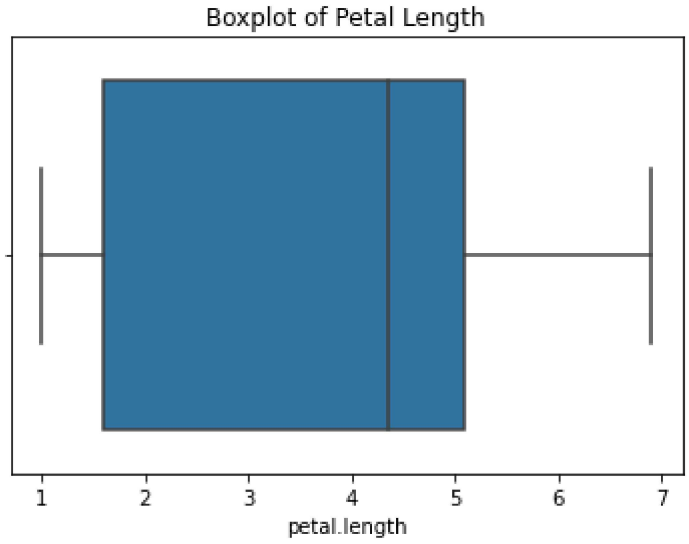
Observations:

- All three flowers are equal in proportion i.e. **33%** each.
- Balanced and imbalanced datasets can be easily classified using a pie chart.

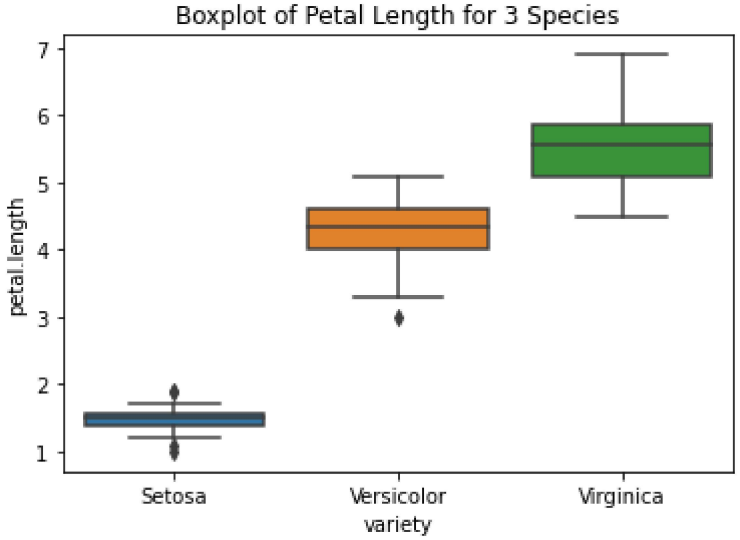
4.6 Box-plot

- Box-plot gives us a five-number summary of any variable: the five-number summary is minimum, maximum, the sample median, the first and third quartile.
- So as you can see this box plot represents 'Petal Length' for all three different species in a single plot.

In [21]:
```
sns.boxplot(x='petal.length', data=data)
plt.title('Boxplot of Petal Length')
plt.show()
```



In [22]:
```
sns.boxplot(x='variety',y='petal.length', data=data)
plt.title('Boxplot of Petal Length for 3 Species')
plt.show()
```



Observations:

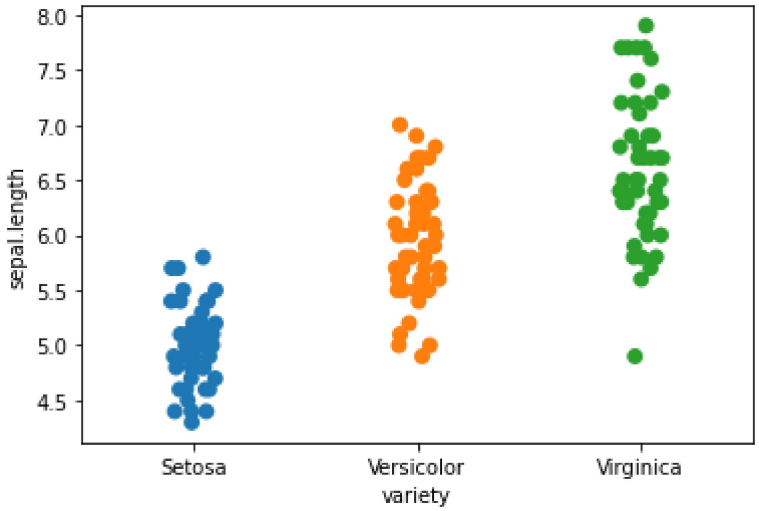
- The petal Length of Setosia is the smallest of all three.
- Virginica has the largest petal length.
- There is an outlier in Versicolor.

Similarly, we can draw box plots for other featuresas well.

4.7 Strip plot

- Strip Plot show all observations along with some representation of the underlying distribution.
- In simple words It is a graphical data analysis technique for summarizing a univariate data set.

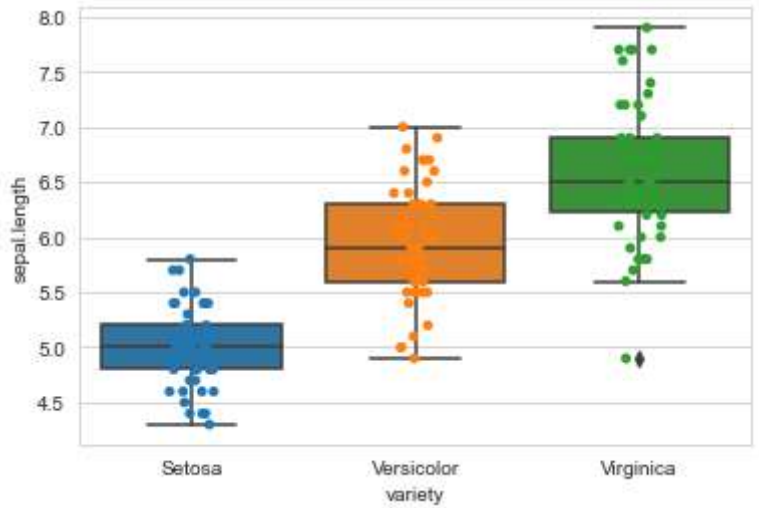
In [13]:
```
fig=plt.gcf()
fig=sns.stripplot(x='variety',y='sepal.length',data=data,jitter=True,size=8,orient='v')
```



4.7.1 Combining Box and Strip Plots

In [67]:
```
# One way we can extend this plot is adding a layer of individual points on top of it through Seaborn's striplot
# We'll use jitter=True so that all the points don't fall in single vertical lines above the species

ax = sns.boxplot(x="variety", y="sepal.length", data=data)
ax = sns.stripplot(x="variety", y="sepal.length", data=data, jitter=True, edgecolor="gray")
```

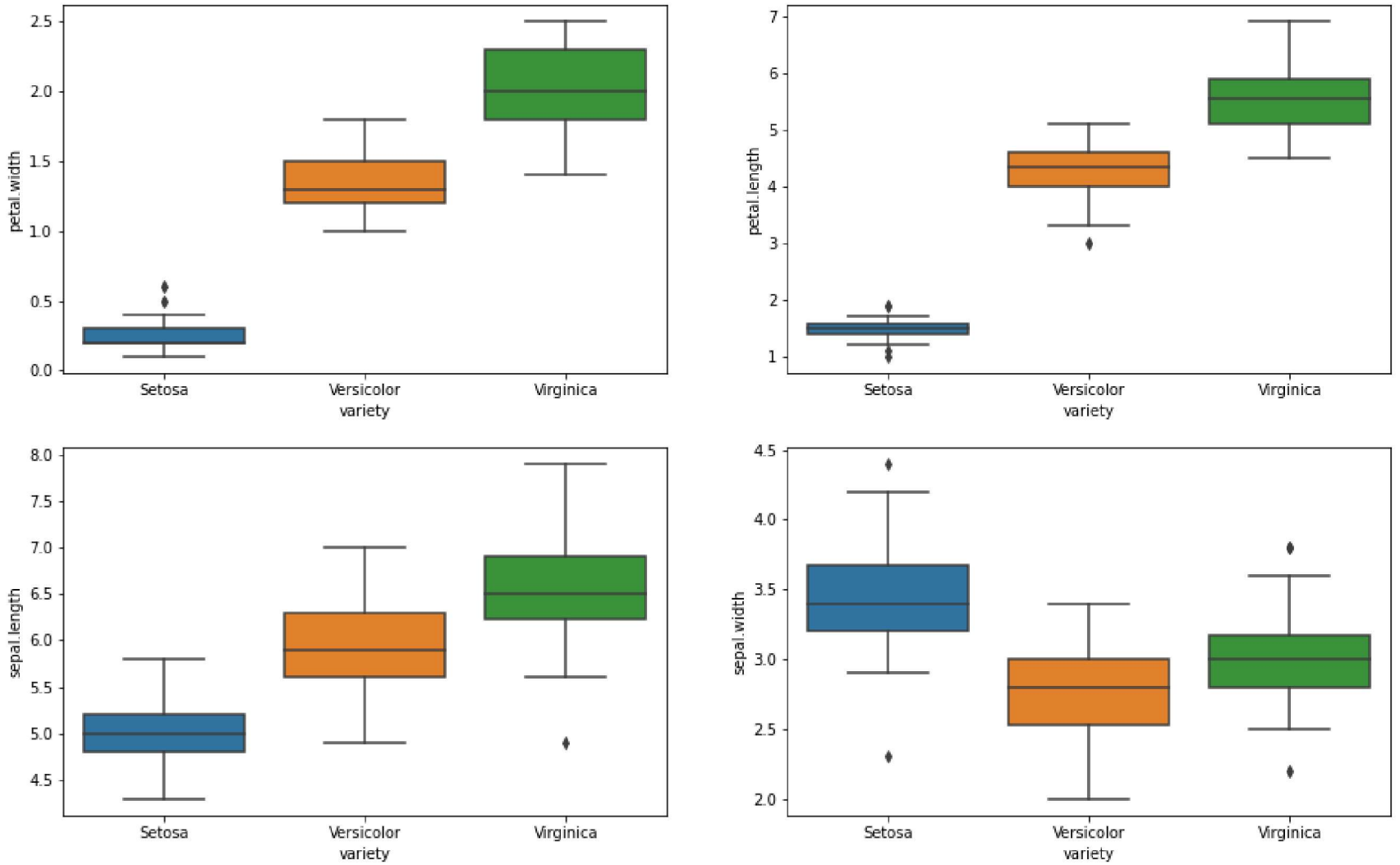


Observations:

- Setosa is having a smaller feature and less distributed
- Versicolor is distributed in an average manner and average features
- Virginica is highly distributed with a large no .of values and features
- Clearly the mean/ median values are being shown by each plots for various features(sepal length & width, petal length & width)

4.7.2 Box plots to know about distribution

In [11]:
```
fig, axes = plt.subplots(2, 2, figsize=(16,10))
sns.boxplot( y='petal.width', x= 'variety', data=data, orient='v' , ax=axes[0, 0])
sns.boxplot( y='petal.length', x= 'variety', data=data, orient='v' , ax=axes[0, 1])
sns.boxplot( y='sepal.length', x= 'variety', data=data, orient='v' , ax=axes[1, 0])
sns.boxplot( y='sepal.width', x= 'variety', data=data, orient='v' , ax=axes[1, 1])
plt.show()
```



These all 4 boxplots represent how the categorical feature “Species” is distributed with all other four input variables

Mean / Median Table for reference

In [52]:
```
data.groupby('variety').agg(['mean', 'median'])
```

Out[52]:

	sepal.length		sepal.width		petal.length		petal.width	
	mean	median	mean	median	mean	median	mean	median
variety								
Setosa	5.006	5.0	3.428	3.4	1.462	1.50	0.246	0.2
Versicolor	5.936	5.9	2.770	2.8	4.260	4.35	1.326	1.3
Virginica	6.588	6.5	2.974	3.0	5.552	5.55	2.026	2.0

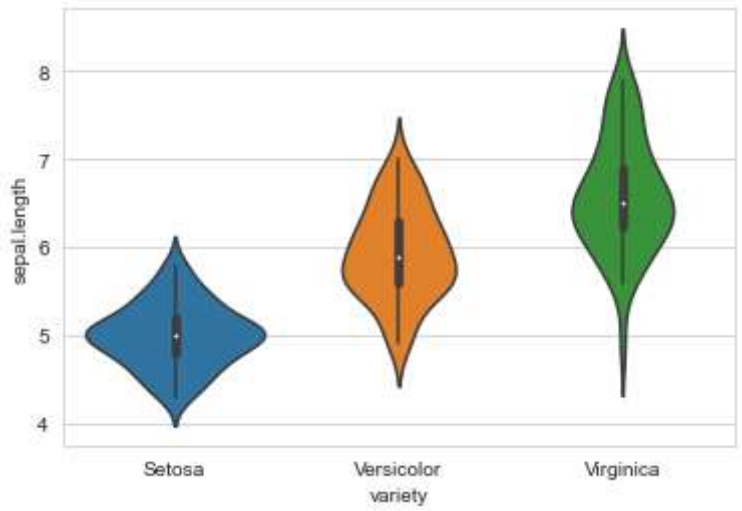
visualizing the distribution , mean and median using box plots & violin plots

4.8 Violin Plot

- The violin plot shows the density of the length and width of the species.
- The thinner part denotes that there is less density whereas the fatter part conveys higher density.

In [69]:
```
sns.violinplot(x="variety", y="sepal.length", data=data, size=6)
```

Out[69]:
```
<AxesSubplot:xlabel='variety', ylabel='sepal.length'>
```

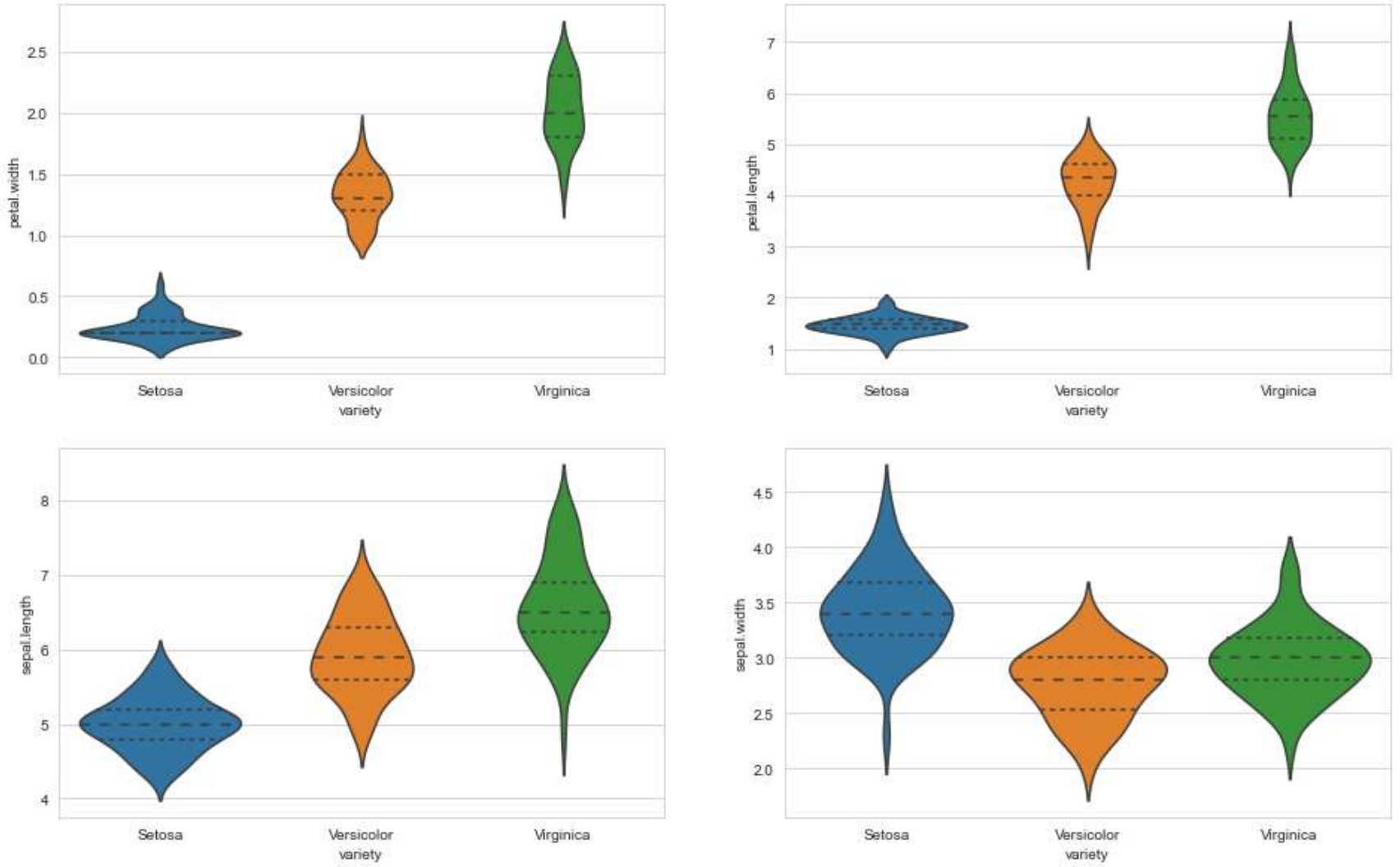


Observations:

- Setosa is having less distribution and density in the case of petal length & width
- Versicolor is distributed in an average manner and average features in case of petal length & width
- Virginica is highly distributed with a large no .of values and features in the case of sepal length & width

4.8.1 Violin Plot for checking distribution

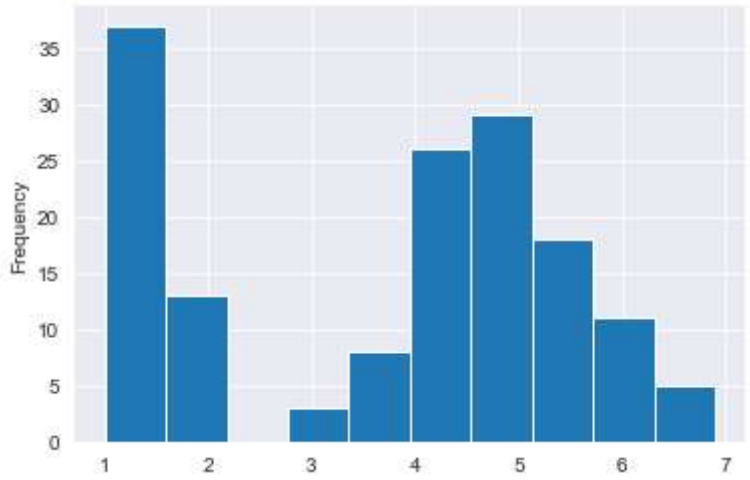
In [59]:
```
fig, axes = plt.subplots(2, 2, figsize=(16,10))
sns.violinplot( y='petal.width', x= 'variety', data=data, orient='v' , ax=axes[0, 0],inner='quartile')
sns.violinplot( y='petal.length', x= 'variety', data=data, orient='v' , ax=axes[0, 1],inner='quartile')
sns.violinplot( y='sepal.length', x= 'variety', data=data, orient='v' , ax=axes[1, 0],inner='quartile')
sns.violinplot( y='sepal.width', x= 'variety', data=data, orient='v' , ax=axes[1, 1],inner='quartile')
plt.show()
```



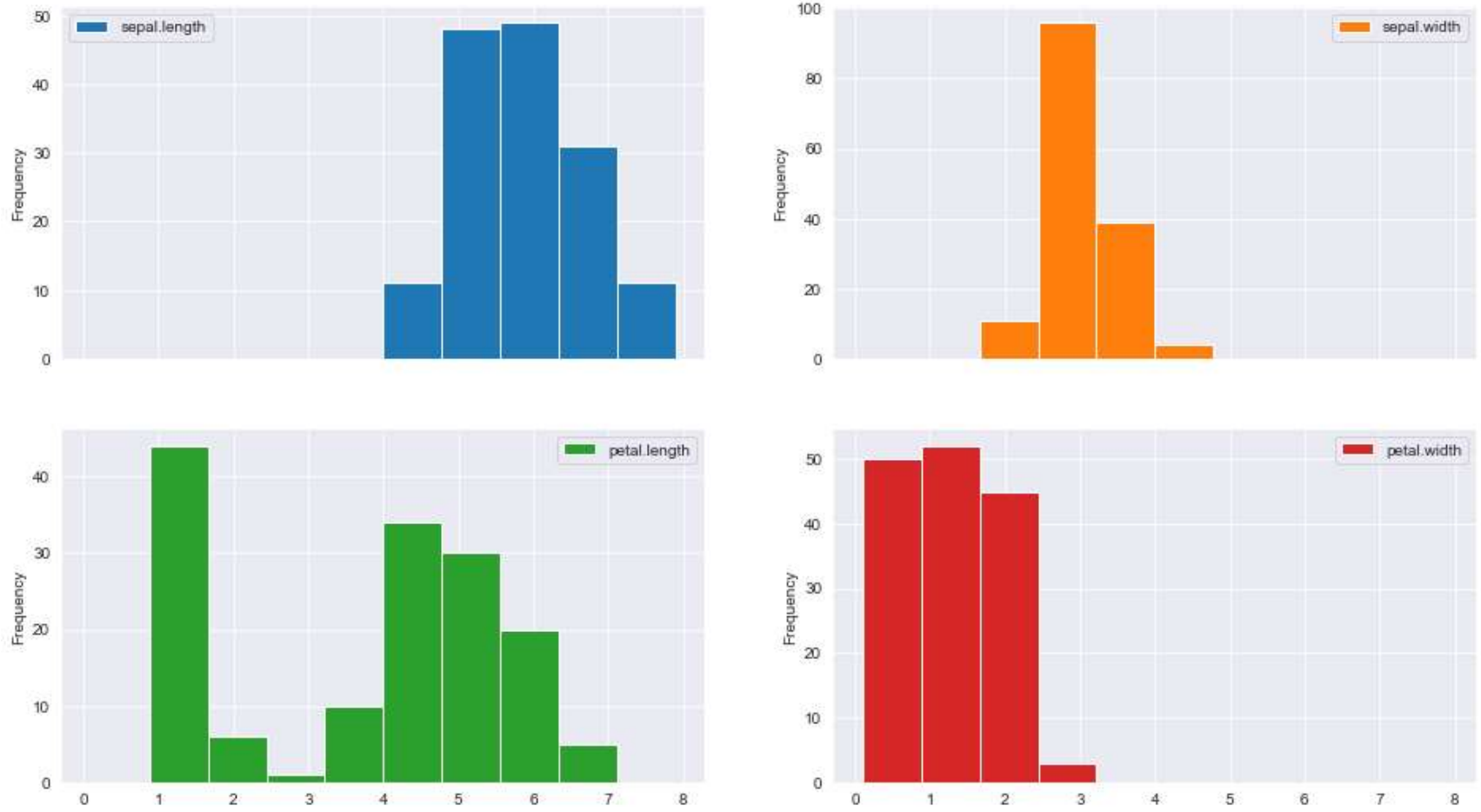
4.9 Histograms and PDF

- Histograms are used to represent the frequency distribution of continuous variables.
- The width of the histogram represents interval and the length represents frequency.
- PDF is a Probability Density Function that is basically smoothening of the histogram.

In [22]:
```
data['petal.length'].plot.hist()
plt.show()
```



In [23]:
```
data.plot.hist(subplots=True, layout=(2,2), figsize=(16,9))
plt.show()
```



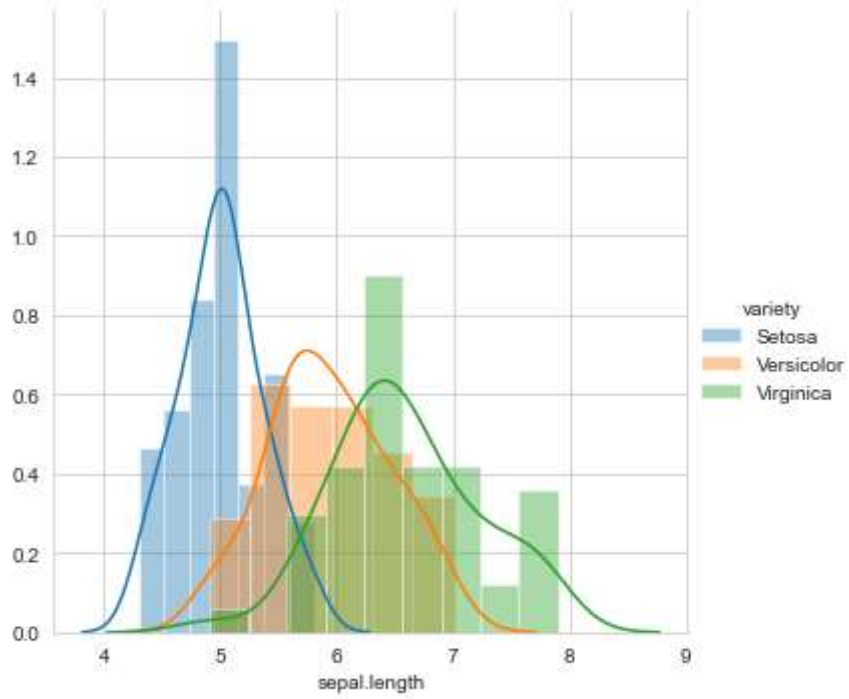
Observations:

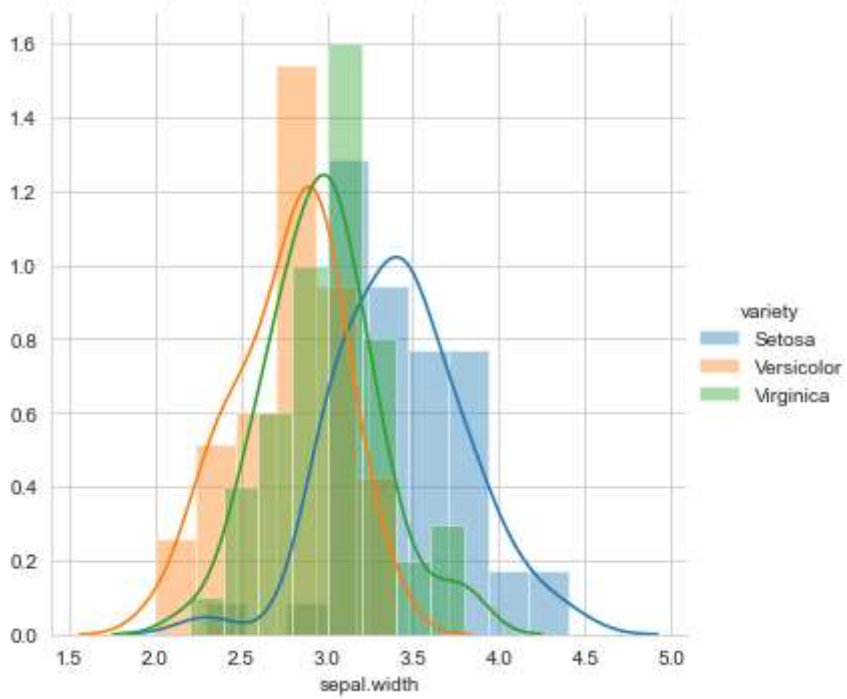
- The Highest frequency of sepal width is between 3.0 to 3.5 which is around 70.
- The Highest frequency of sepal length is between 5.5 and 6.0 which is around 35.
- The Highest frequency of petal width is between 0 to 0.5 which is around 50.
- The Highest frequency of petal length is between 0 to 0.5 which is around 50.

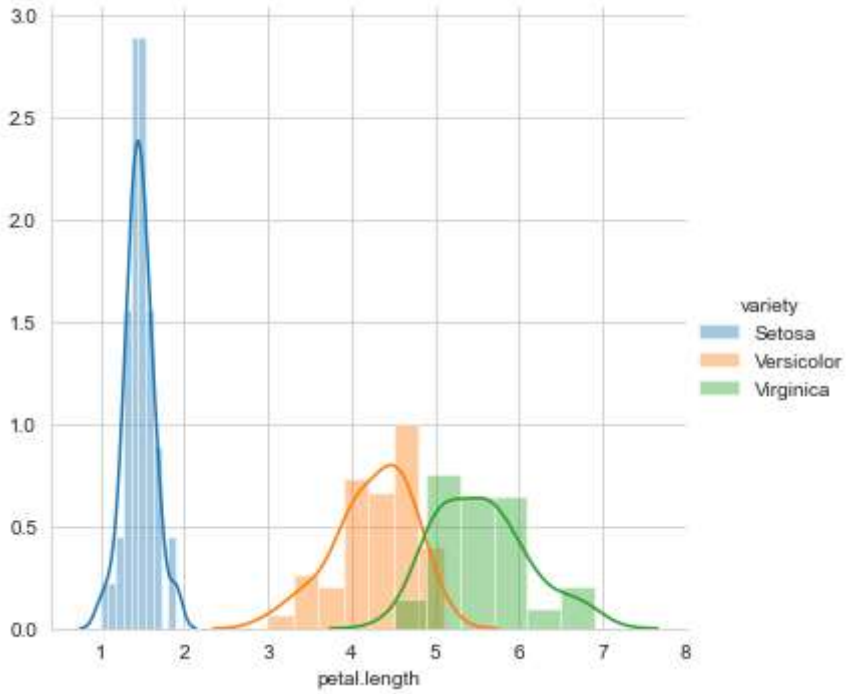
4.9.1 Plotting the Histogram & Probability Density Function (PDF)

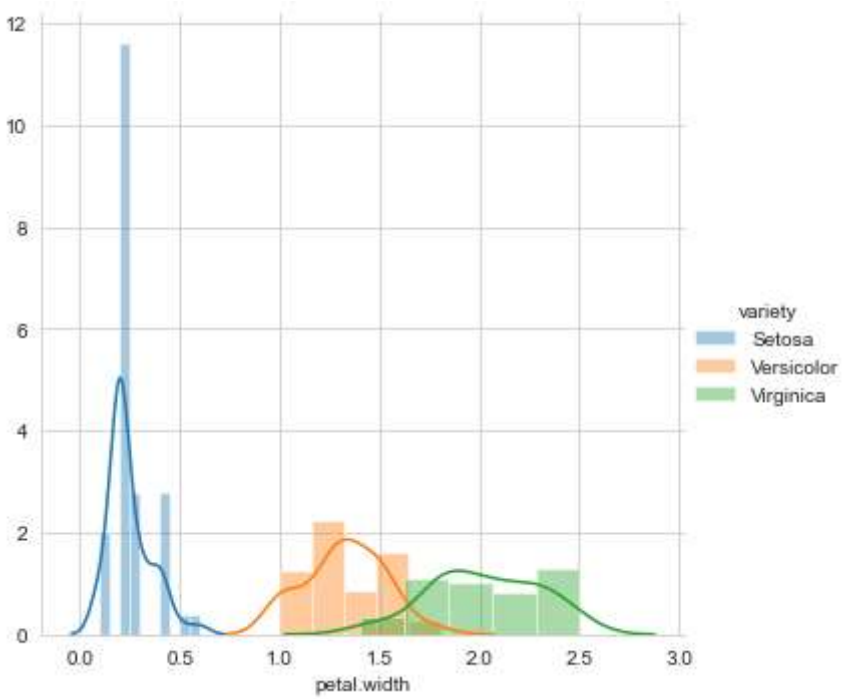
plotting the probability density function(PDF) with each feature as a variable on X-axis and it’s histogram and corresponding kernel density plot on Y-axis.

In [96]:
```
sns.FacetGrid(data, hue="variety", height=5) \
.map(sns.distplot, "sepal.length") \
.add_legend()
sns.FacetGrid(data, hue="variety", height=5) \
.map(sns.distplot, "sepal.width") \
.add_legend()
sns.FacetGrid(data, hue="variety", height=5) \
.map(sns.distplot, "petal.length") \
.add_legend()
sns.FacetGrid(data, hue="variety", height=5) \
.map(sns.distplot, "petal.width") \
.add_legend()
plt.show()
```




1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
sepal.width
variety
Setosa
Versicolor
Virginica


3.0
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7
8
petal.length
variety
Setosa
Versicolor
Virginica



Observations:

1. Plot 1 shows that there is a significant amount of overlap between the species on sepal length, so it is not an effective Classification feature
2. Plot 2 shows that there is an even higher overlap between the species on sepal width, so it is not an effective Classification feature
3. Plot 3 shows that petal length is a good Classification feature as it clearly separates the species. The overlap is extremely less (between Versicolor and Virginica) , Setosa is well separated from the rest two
4. Just like Plot 3, Plot 4 also shows that petal width is a good Classification feature. The overlap is significantly less (between Versicolor and Virginica) , Setosa is well separated from the rest two

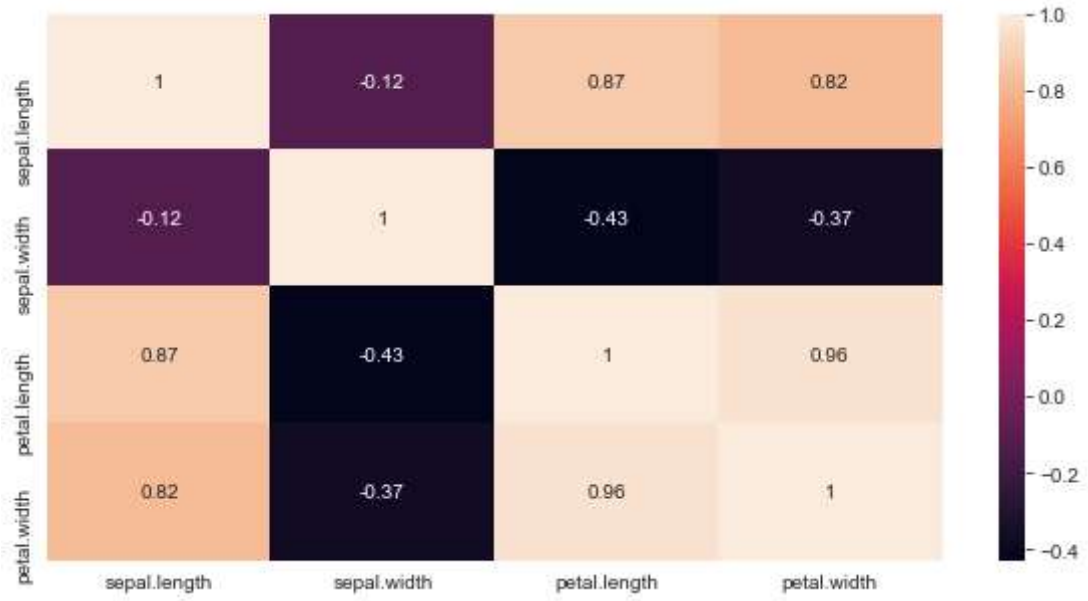
4.10 Heat Map

Checking Correlation

- It uses color in order to communicate the correlation between two variables. Values are between -1 to 1.
- 1 denotes a perfect positive correlation. 0 means no correlation and -1 means the highest negative correlation.

In [49]:
```
plt.figure(figsize=(10,5))
sns.heatmap(data.corr(),annot=True)
plt.plot()
```

Out[49]: []



Data Insights:

Observations:

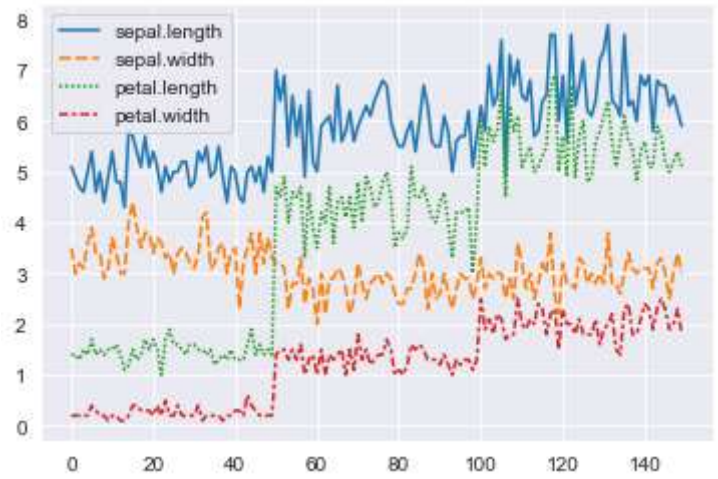
- Petal Length and Petal Width shows the highest positive correlation 0.96

- Petal Length shows a high positive correlation of 0.87 with Sepal Length as well.
- Petal Width shows a high positive correlation of 0.82 with the Sepal Length as well.
- Petal Length and Sepal Width shows a negative correlation of -0.43
- Sepal Width shows a negative correlation with the other 3 features

4.11 Line chart

- The line chart represents a series of data points connected by a straight line.
- It is generally used to visualize data that changes over time.

In [18]:
```
sns.set_style('darkgrid')
sns.lineplot(data=data.drop(['variety'], axis=1))
plt.show()
```

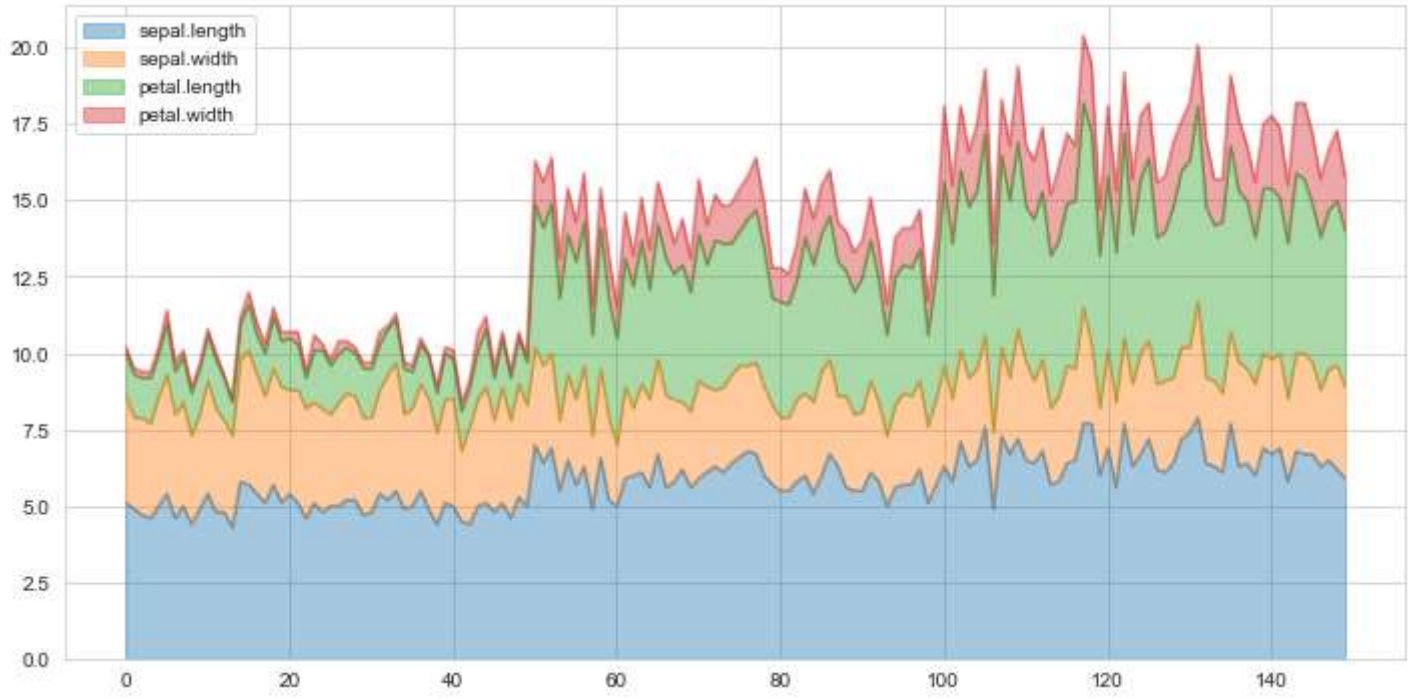


This line chart showing how Petal Width changes with change in Petal Length.

4.12 Area Plot

An Area Plot gives us a visual representation of Various dimensions of the Iris flower and their range in a dataset.

In [77]:
```
data.plot.area(y=['sepal.length','sepal.width','petal.length','petal.width'],alpha=0.4,figsize=(12, 6));
```



Conclusion

- The dataset is balanced i.e. equal records are present for all three species.
- We have four numerical columns while just one categorical column which in turn is our target column.
- A strong correlation is present between petal width and petal length.
- The setosa species is the most easily distinguishable because of its small feature size.
- The Versicolor and Virginica species are usually mixed and are sometimes hard to separate, while usually Versicolor has average feature sizes and virginica has larger feature sizes.