

# Graphics in Python

Libraries:

**pandas**

**matplotlib**

**seaborn**

# Graphics Packages

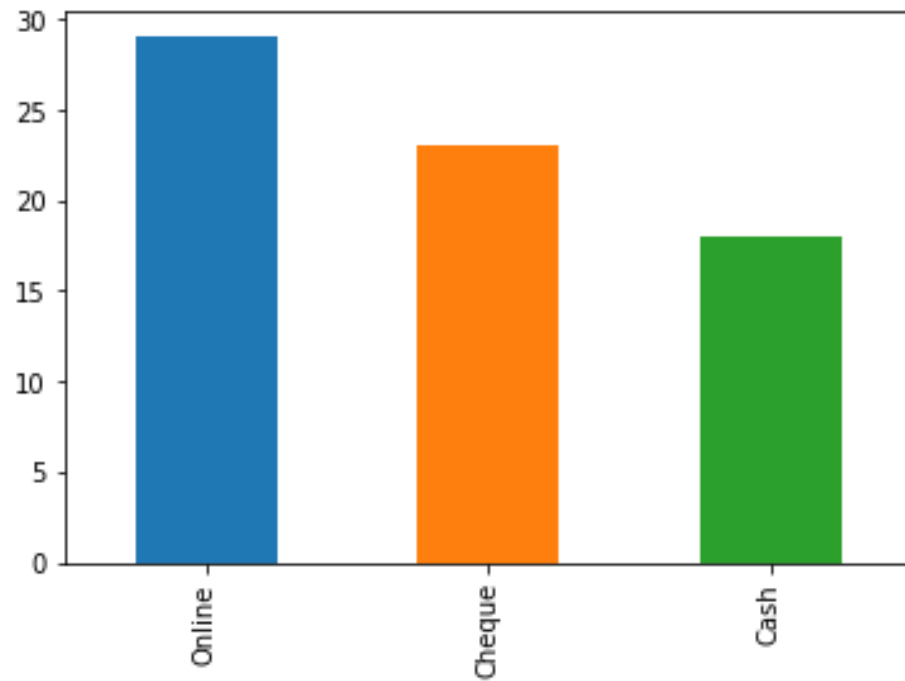
- The graphs can be created with the help of packages:
  - Pandas
  - Matplotlib
  - Seaborn

# Graphs using pandas

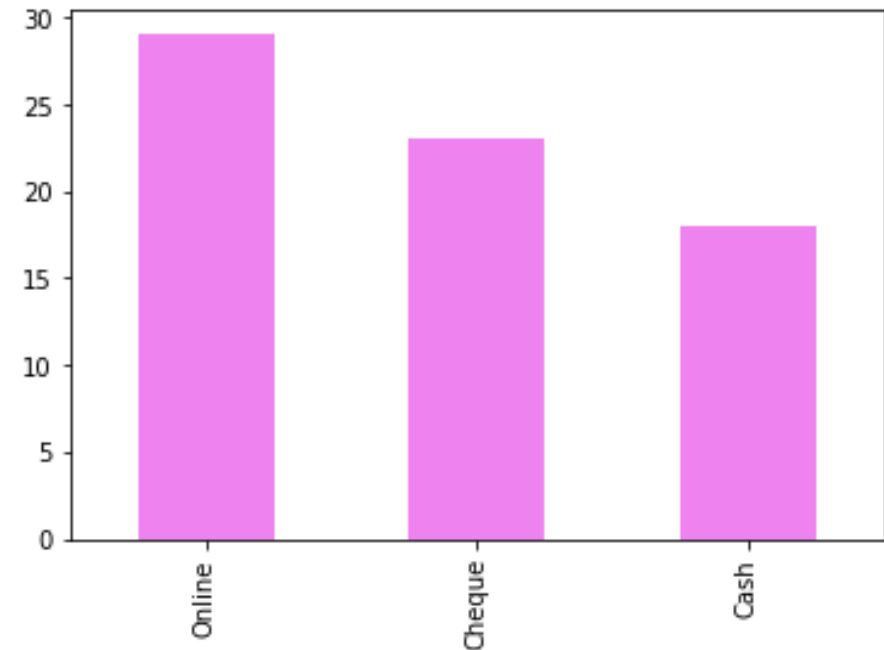
- pandas has a plot method to which we can pass the argument kind=
- kind : str
  - 'line' : line plot (default)
  - 'bar' : vertical bar plot
  - 'barh' : horizontal bar plot
  - 'hist' : histogram
  - 'box' : boxplot
  - 'kde' : Kernel Density Estimation plot
  - 'density' : same as 'kde'
  - 'area' : area plot
  - 'pie' : pie plot

# Bar Chart

```
In [21]: cts = Orders['Payment Terms'].value_counts()  
...: cts.plot(kind='bar')  
...: plt.show()
```

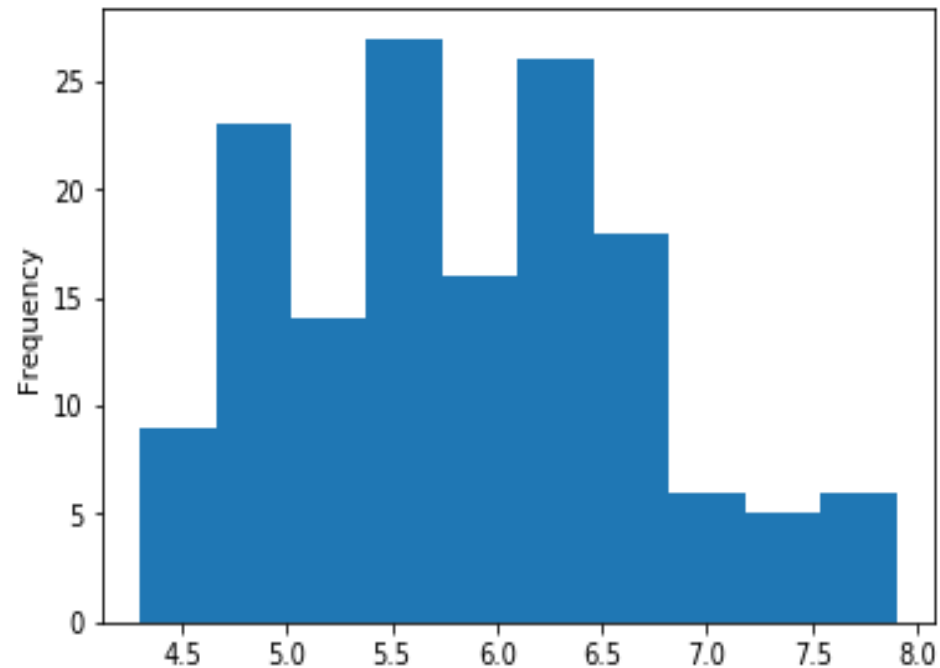


```
In [24]: cts = Orders['Payment Terms'].value_counts()  
...: cts.plot(kind='bar', color="violet")  
...: plt.show()
```

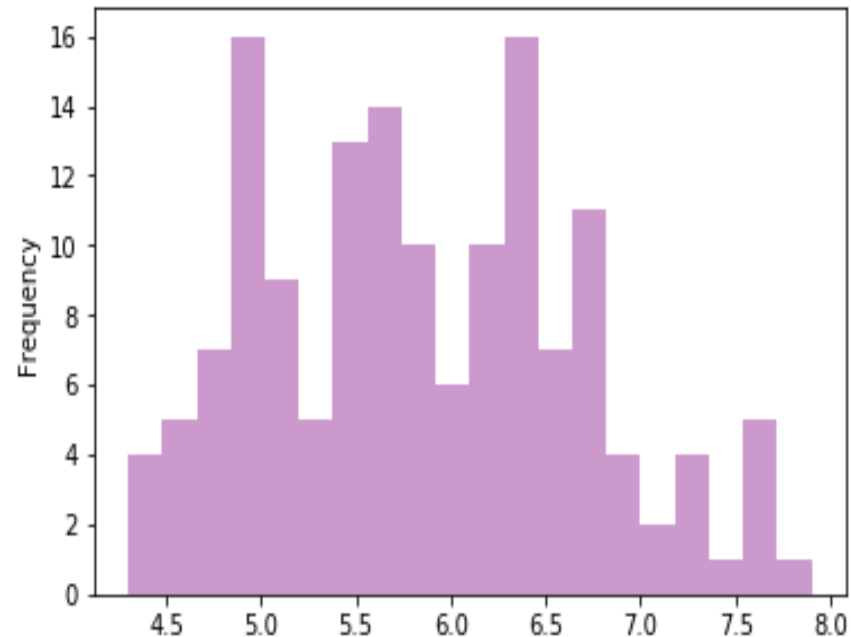


# Histogram

```
In [8]: iris['Sepal.Length'].plot(kind='hist')  
...: plt.show()
```

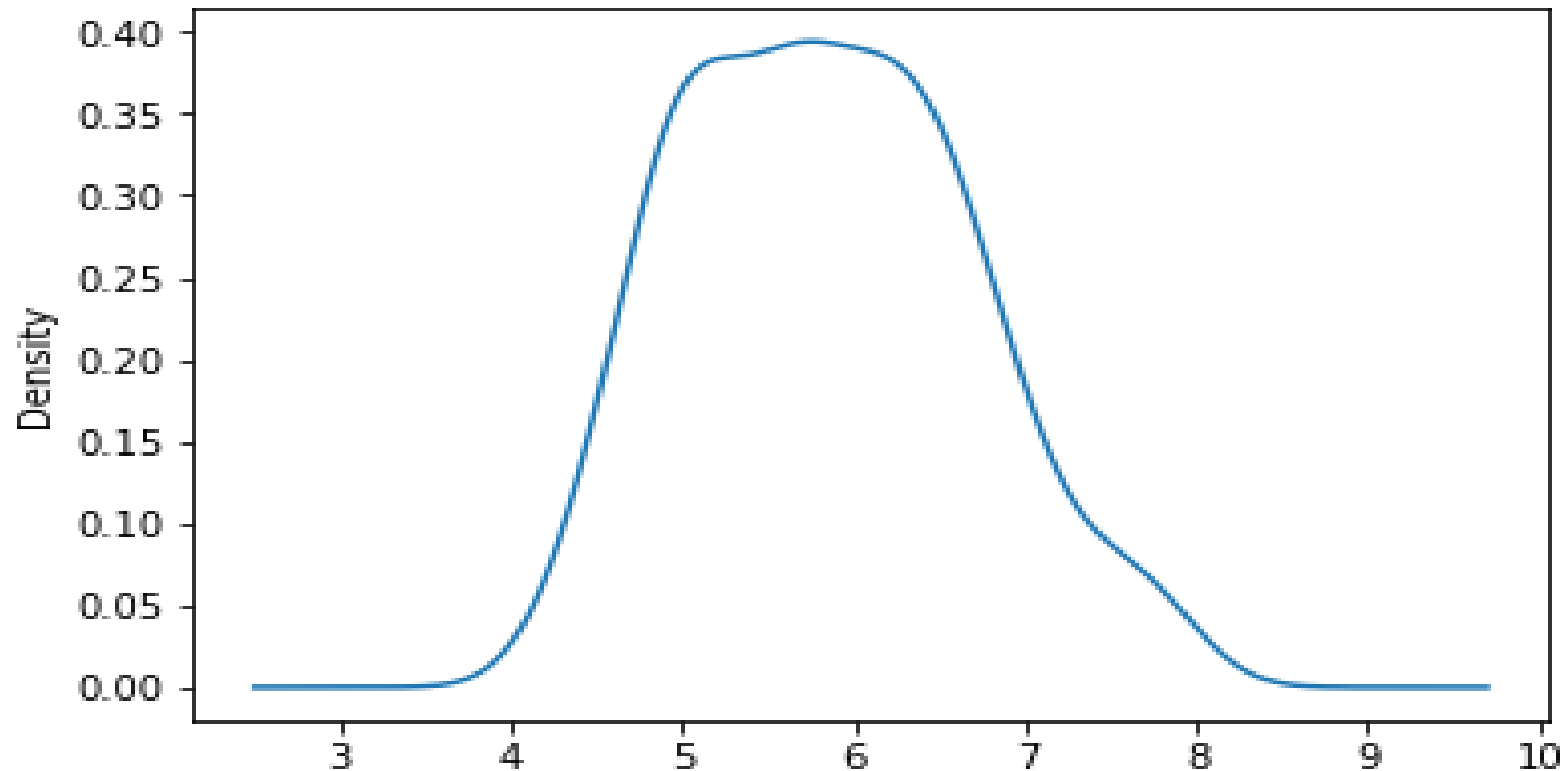


```
In [19]:  
iris['Sepal.Length'].plot(kind='hist',color="purple",alpha=0.4,bins=20)  
...: plt.show()
```



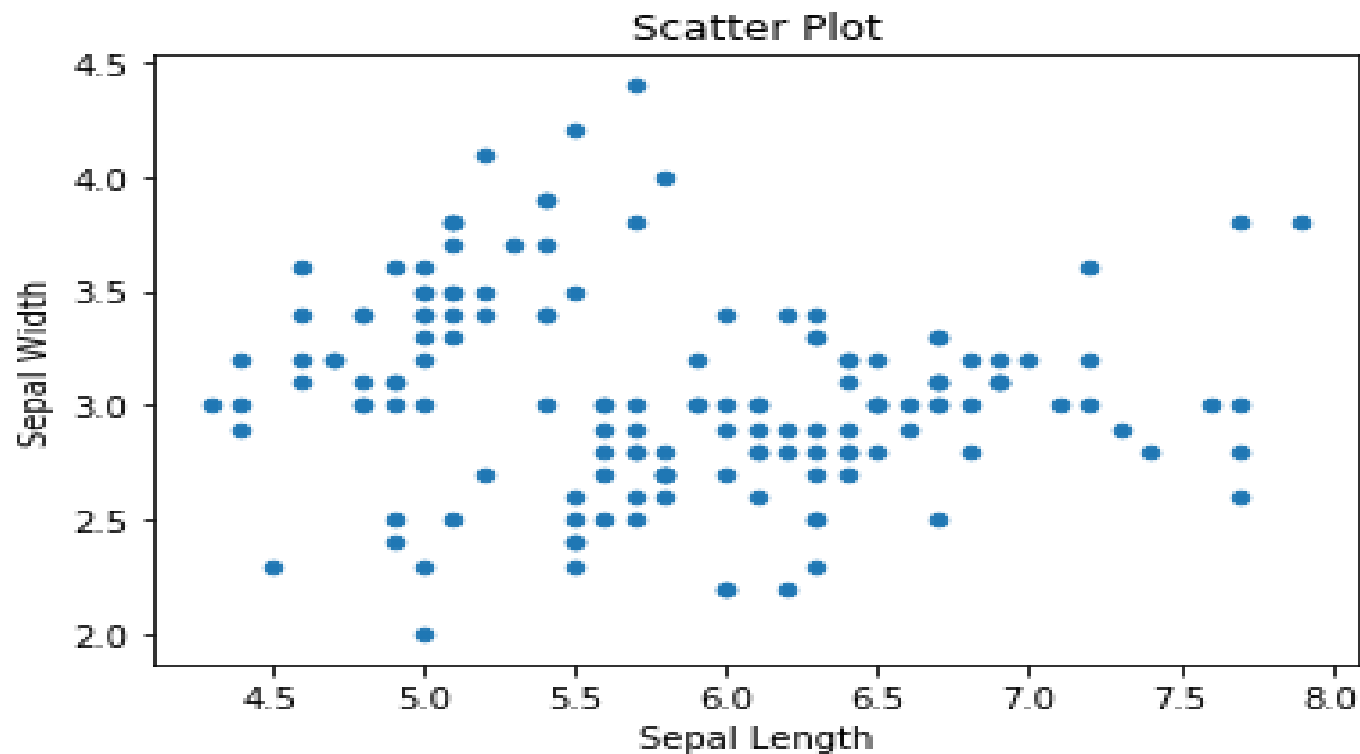
# Density Plot

```
In [28]: iris['Sepal.Length'].plot(kind='kde')  
.....: plt.show()
```



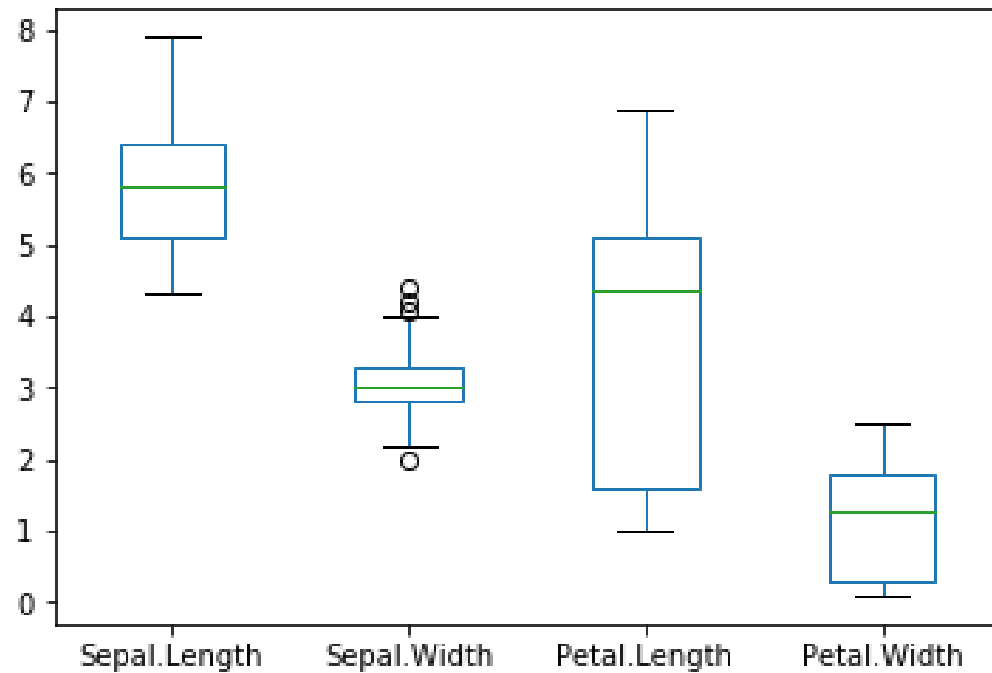
# Scatter Plot

```
In [33]: iris.plot(kind='scatter', x='Sepal.Length', y='Sepal.Width')
....: plt.xlabel("Sepal Length")
....: plt.ylabel("Sepal Width")
....: plt.title("Scatter Plot")
....: plt.show()
```

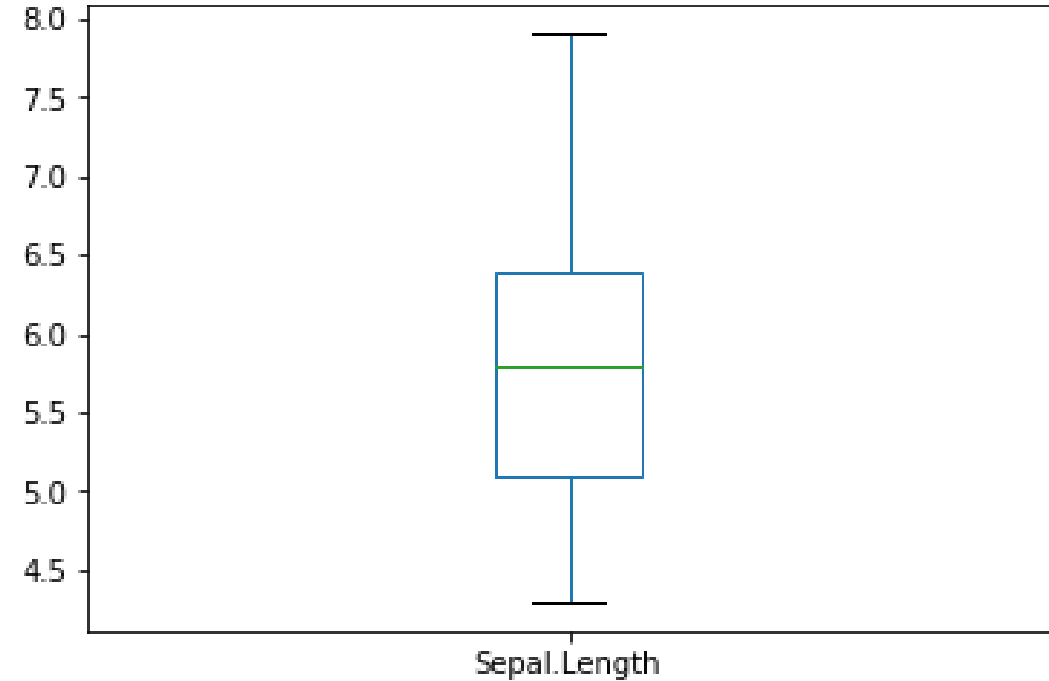


# Boxplot

```
In [34]: iris.plot(kind='box')  
...: plt.show()
```



```
In [39]: iris['Sepal.Length'].plot(kind='box')  
...: plt.show()
```



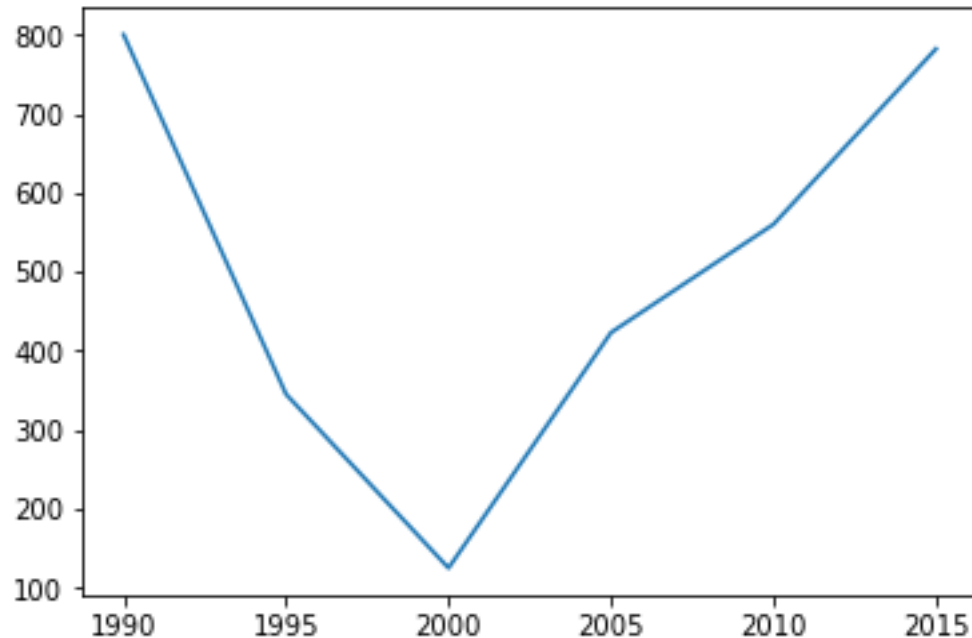


# Library matplotlib

- matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms
- matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code.

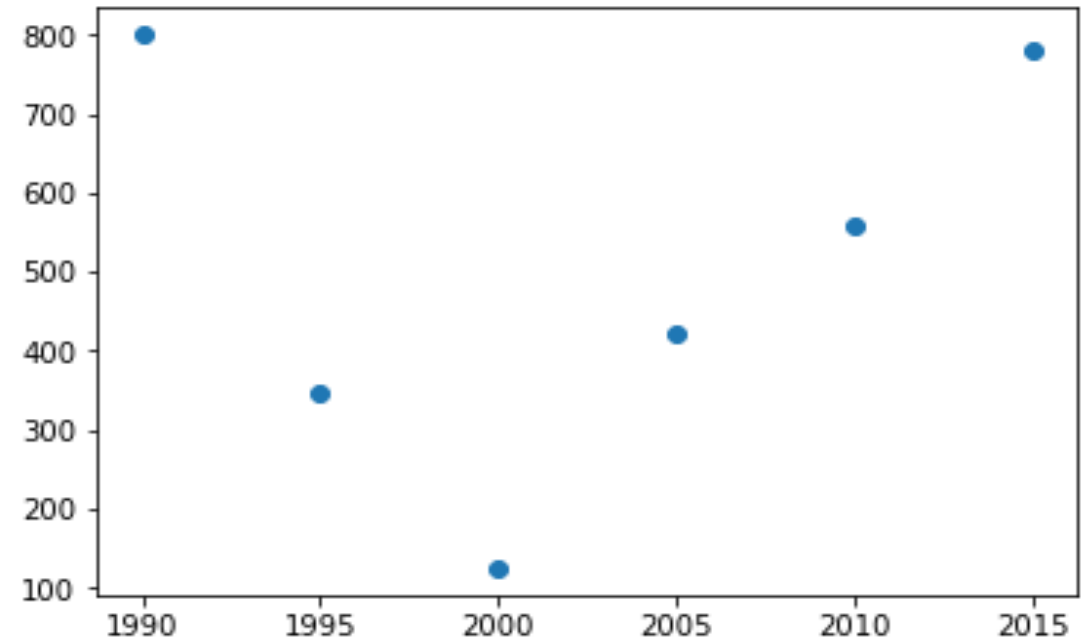
# Line Plot

```
import matplotlib.pyplot as plt
x = [1990,1995,2000,2005,2010,2015]
y = [800,345,125,423,560,782]
plt.plot(x,y)
plt.show()
```



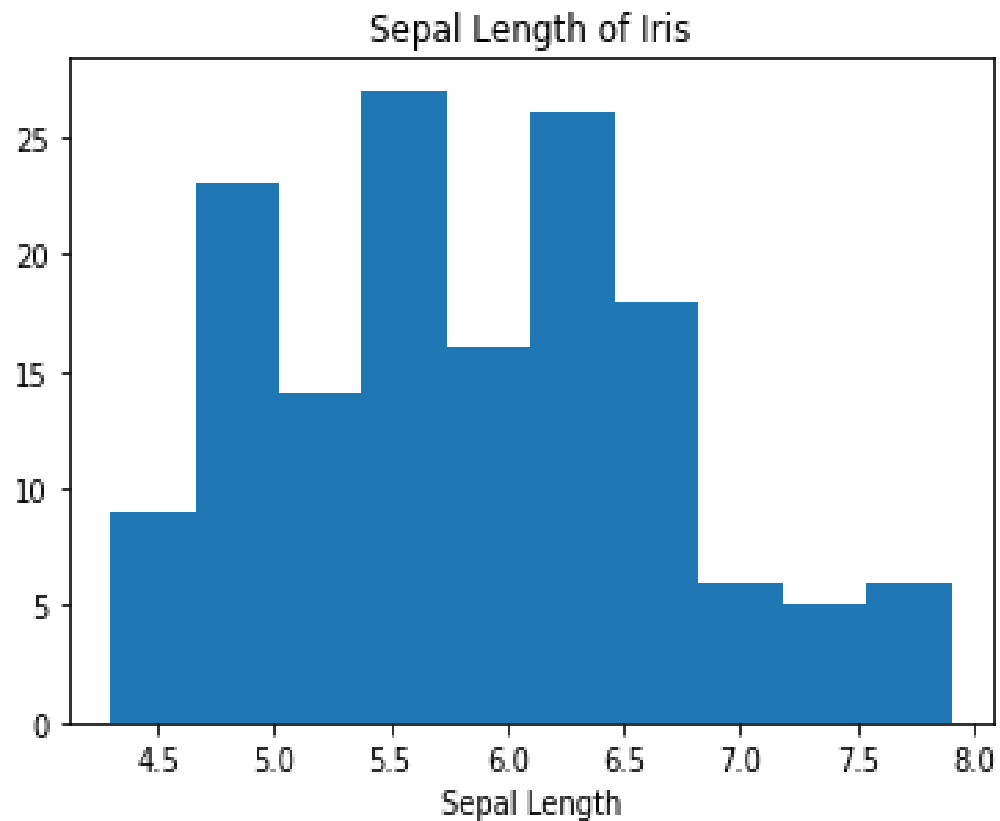
# Scatter Plot

```
import matplotlib.pyplot as plt
x = [1990,1995,2000,2005,2010,2015]
y = [800,345,125,423,560,782]
plt.scatter(x,y)
plt.show()
```

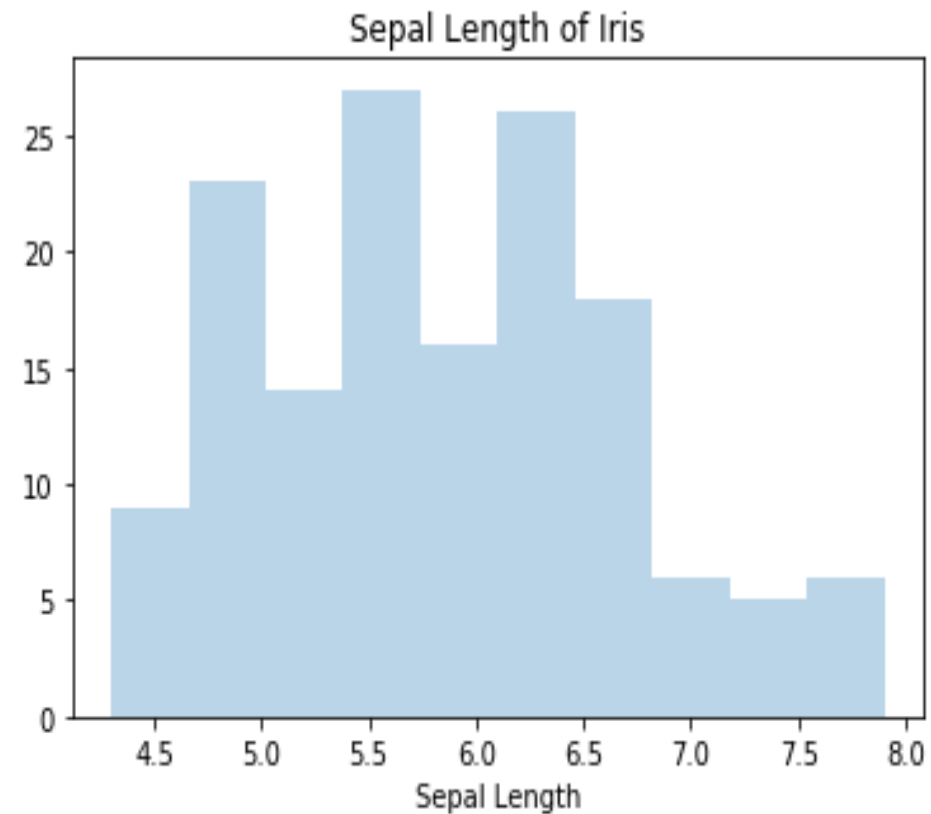


# Histogram

```
In [40]: plt.hist(iris['Sepal.Length'] , bins=10)
....: plt.xlabel("Sepal Length")
....: plt.title("Sepal Length of Iris")
....: plt.show()
```

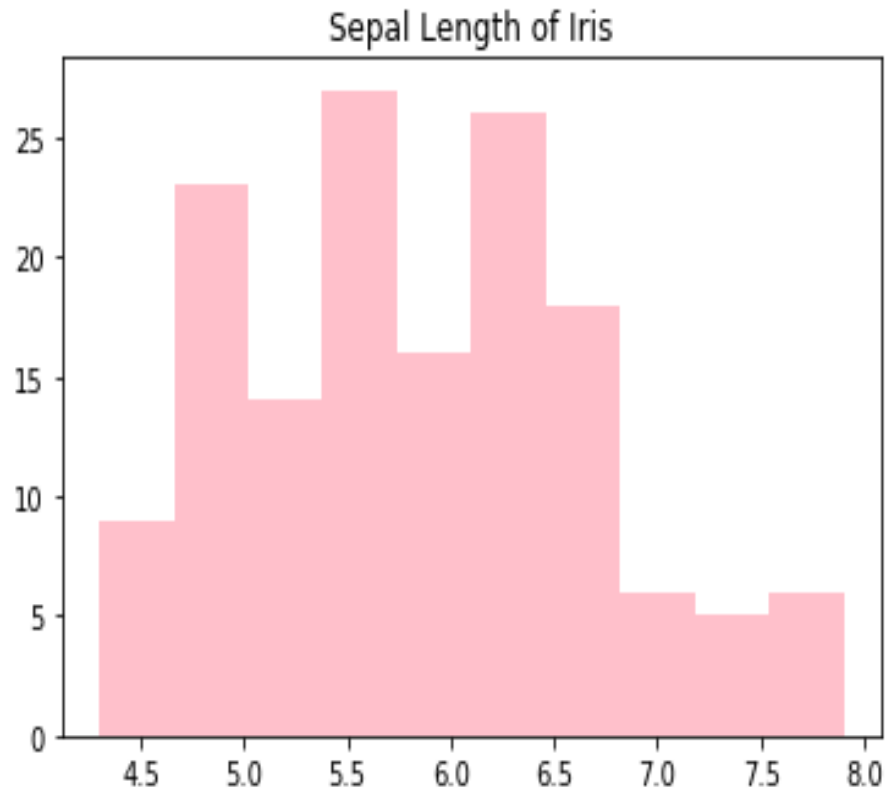


```
In [42]: plt.hist(iris['Sepal.Length'] , bins=10, alpha=0.3)
....: plt.xlabel("Sepal Length")
....: plt.title("Sepal Length of Iris")
....: plt.show()
```

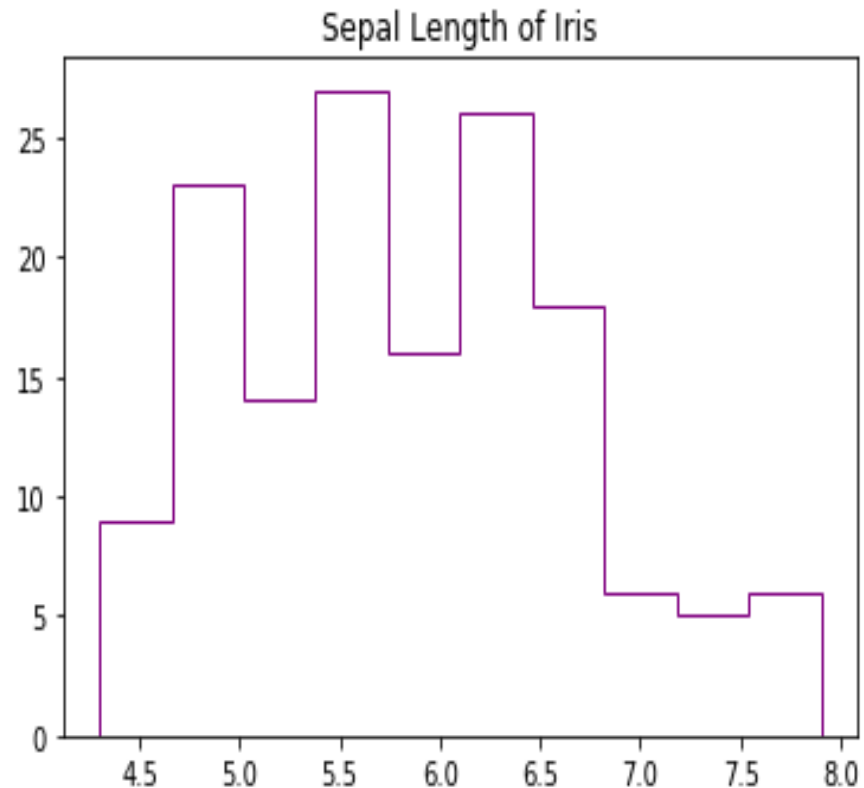


# Histogram

```
In [19]: plt.hist(iris['Sepal.Length'],color='pink')  
...: plt.title("Sepal Length of Iris")  
Out[19]: Text(0.5,1,'Sepal Length of Iris')
```

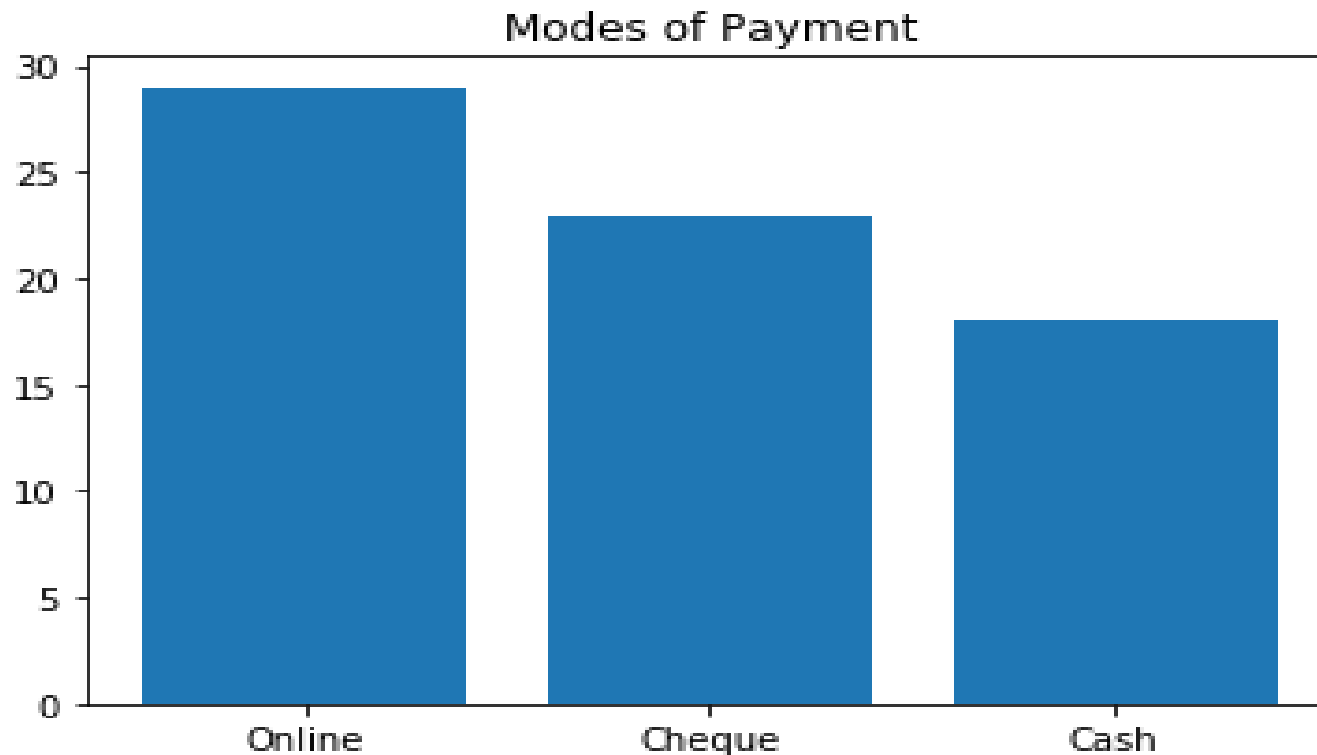


```
In [18]: plt.hist(iris['Sepal.Length'],color='purple',histtype='step')  
...: plt.title("Sepal Length of Iris")  
Out[18]: Text(0.5,1,'Sepal Length of Iris')
```



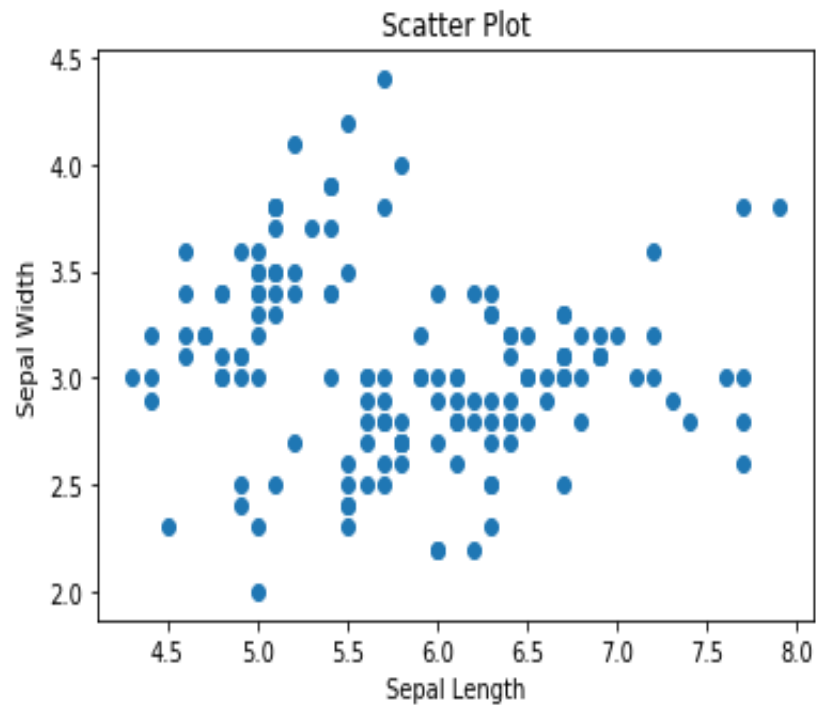
# Bar Chart

```
In [4]: cts = Orders['Payment Terms'].value_counts()  
....: plt.bar(cts.index, cts)  
....: plt.title("Modes of Payment")  
....: plt.show()
```

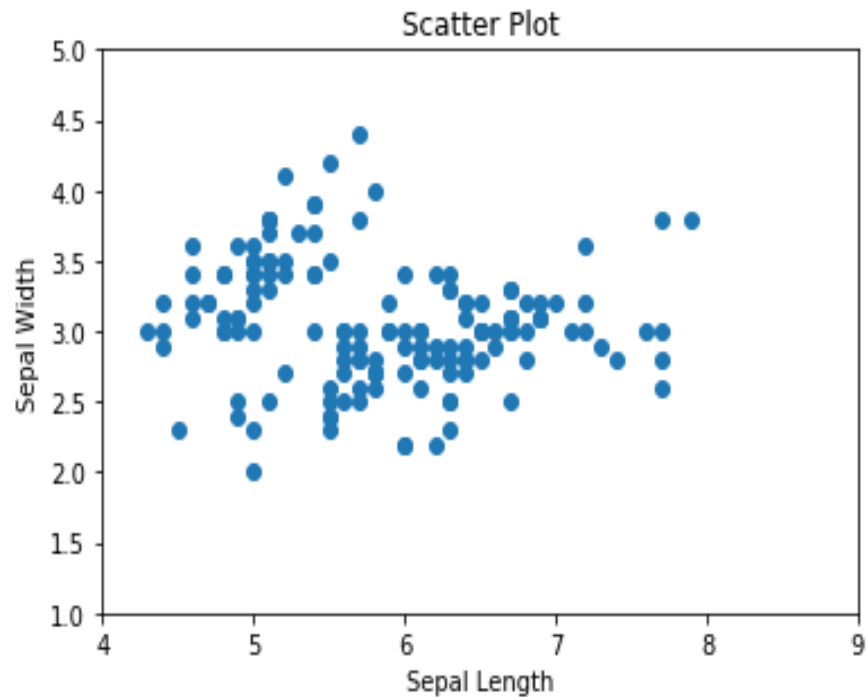


# Customizing the axes

```
In [20]: plt.scatter(iris['Sepal.Length'], iris['Sepal.Width'])
...: plt.xlabel("Sepal Length")
...: plt.ylabel("Sepal Width")
...: plt.title('Scatter Plot')
...: plt.show()
...:
```

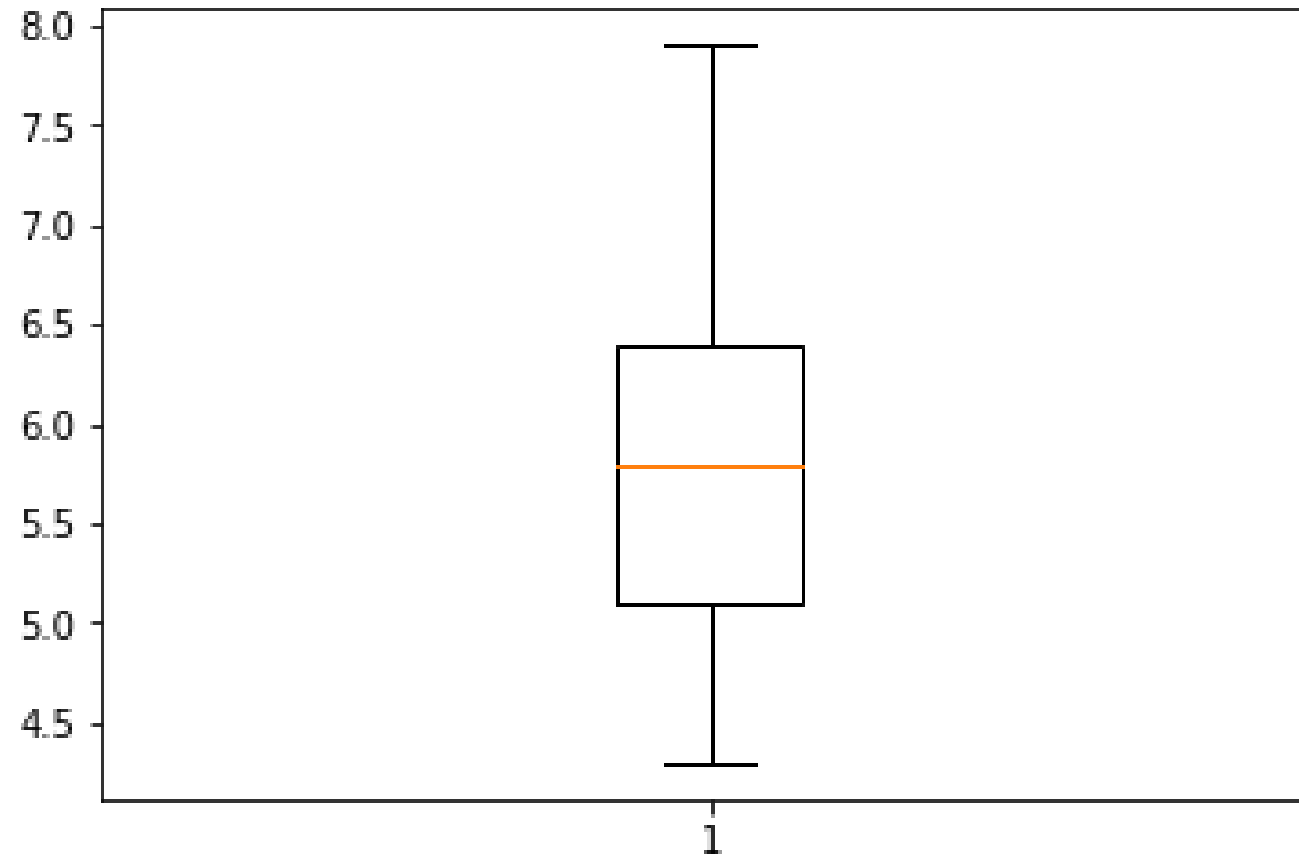


```
In [21]: plt.scatter(iris['Sepal.Length'], iris['Sepal.Width'])
...: plt.xlabel("Sepal Length")
...: plt.ylabel("Sepal Width")
...: plt.title('Scatter Plot')
...: plt.axis((4,9,1,5))
...: plt.show()
```



# Boxplot

```
In [61]: plt.boxplot(iris['Sepal.Length'])  
...: plt.show()
```



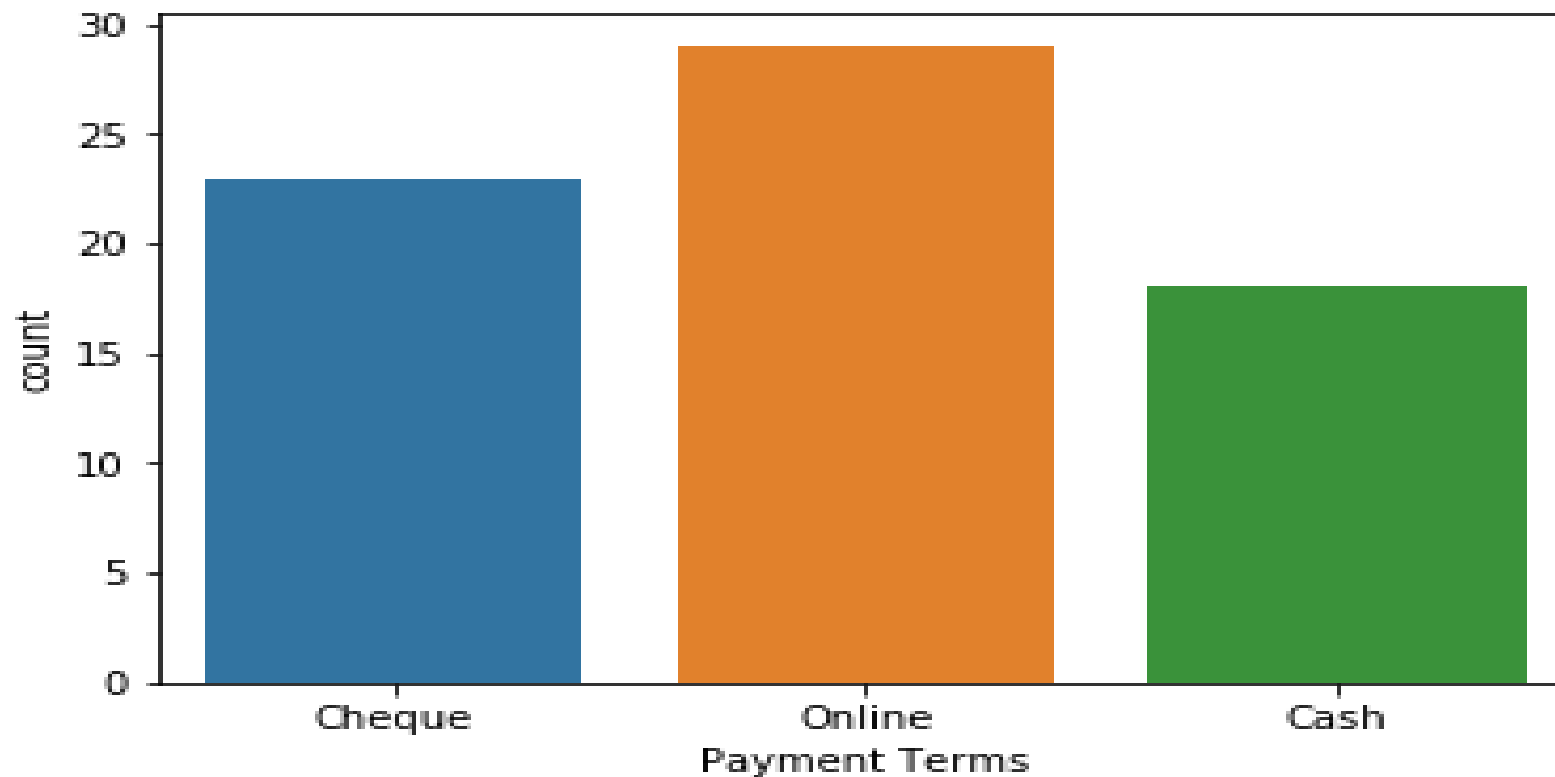
# Library seaborn

- Seaborn is a Python data visualization library based on matplotlib.
- It provides a high-level interface for drawing attractive and informative statistical graphics.



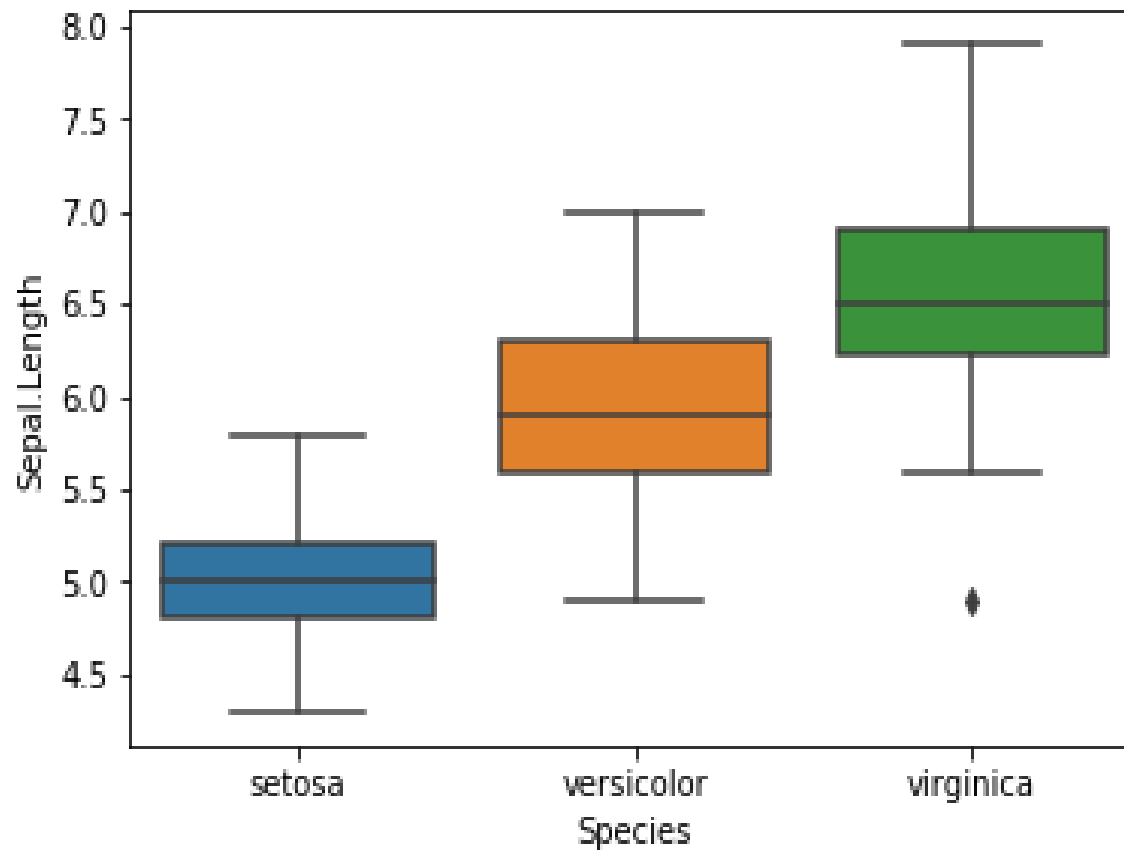
# Bar Chart

```
In [22]: import seaborn as sns  
....: sns.countplot('Payment Terms', data=Orders)  
....: plt.show()
```



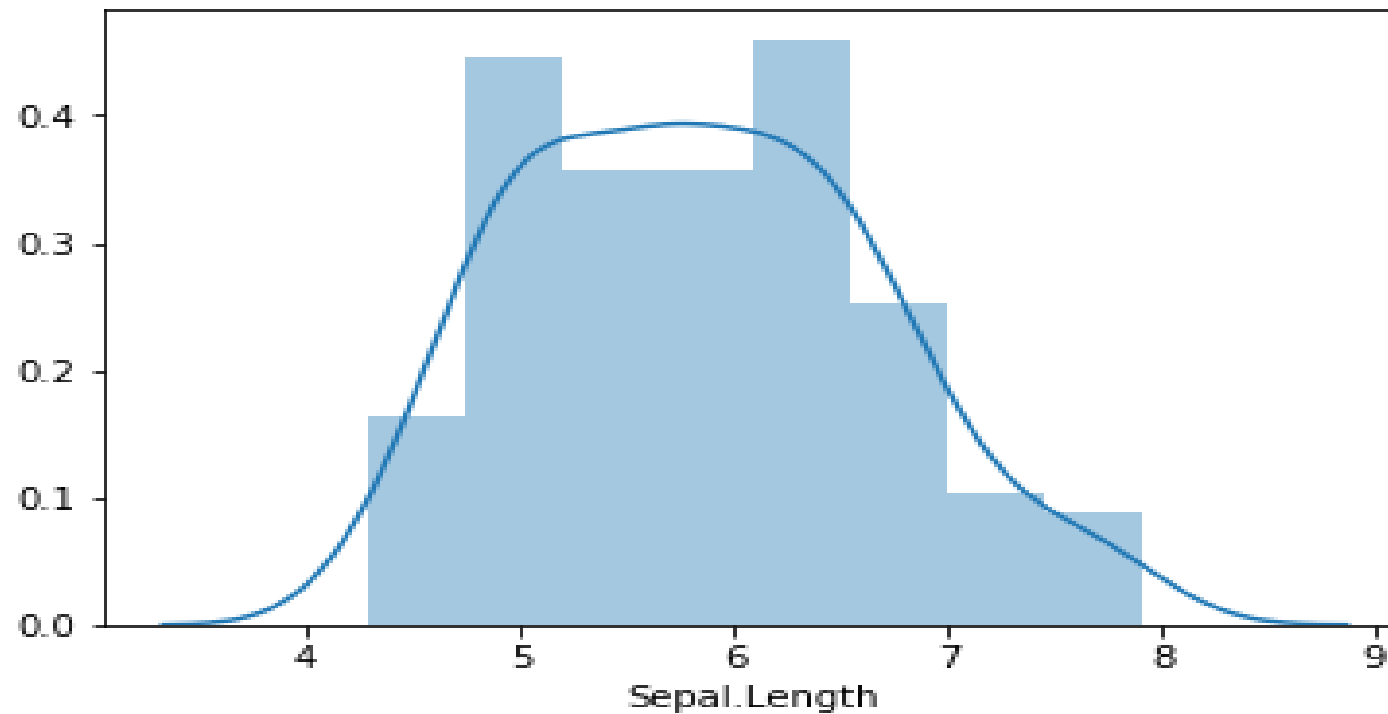
# Boxplot

```
In [33]: sns.boxplot(x='Species', y='Sepal.Length', data=iris)
...: plt.show()
```



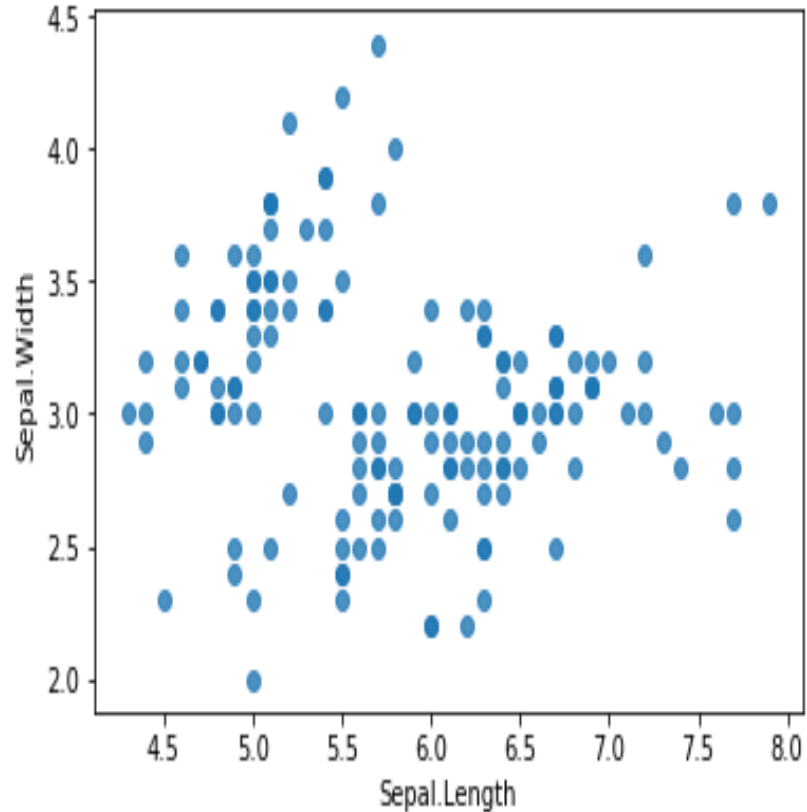
# Histogram

```
sns.distplot(iris['Sepal.Length'])  
plt.show()
```

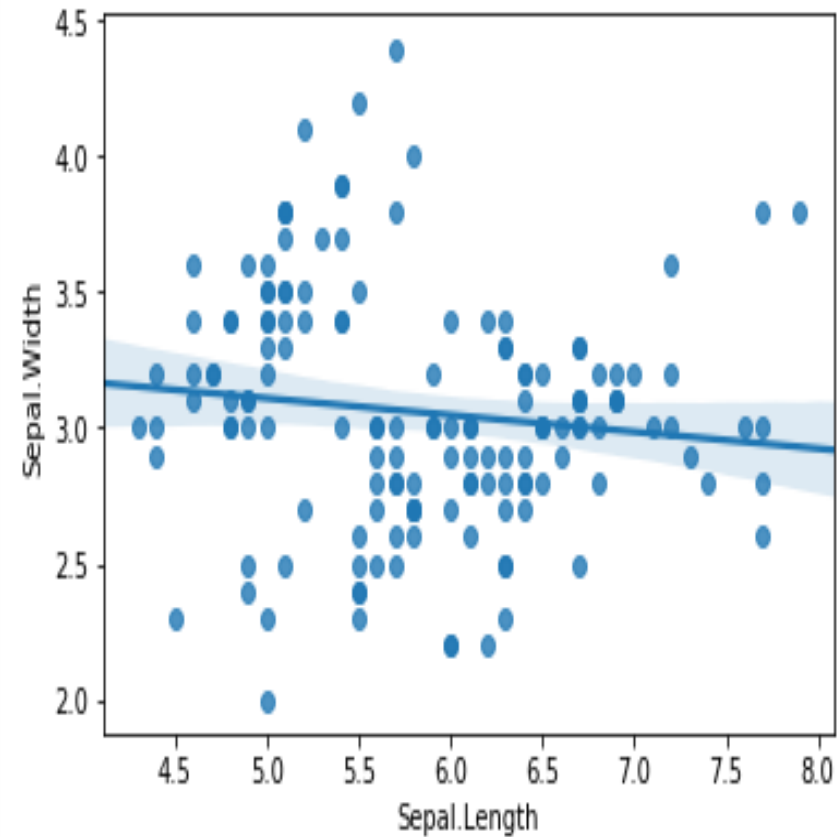


# Scatter Plot

```
In [35]: sns.regplot(x='Sepal.Length', y='Sepal.Width', data=iris,  
fit_reg=False)  
...: plt.show()
```

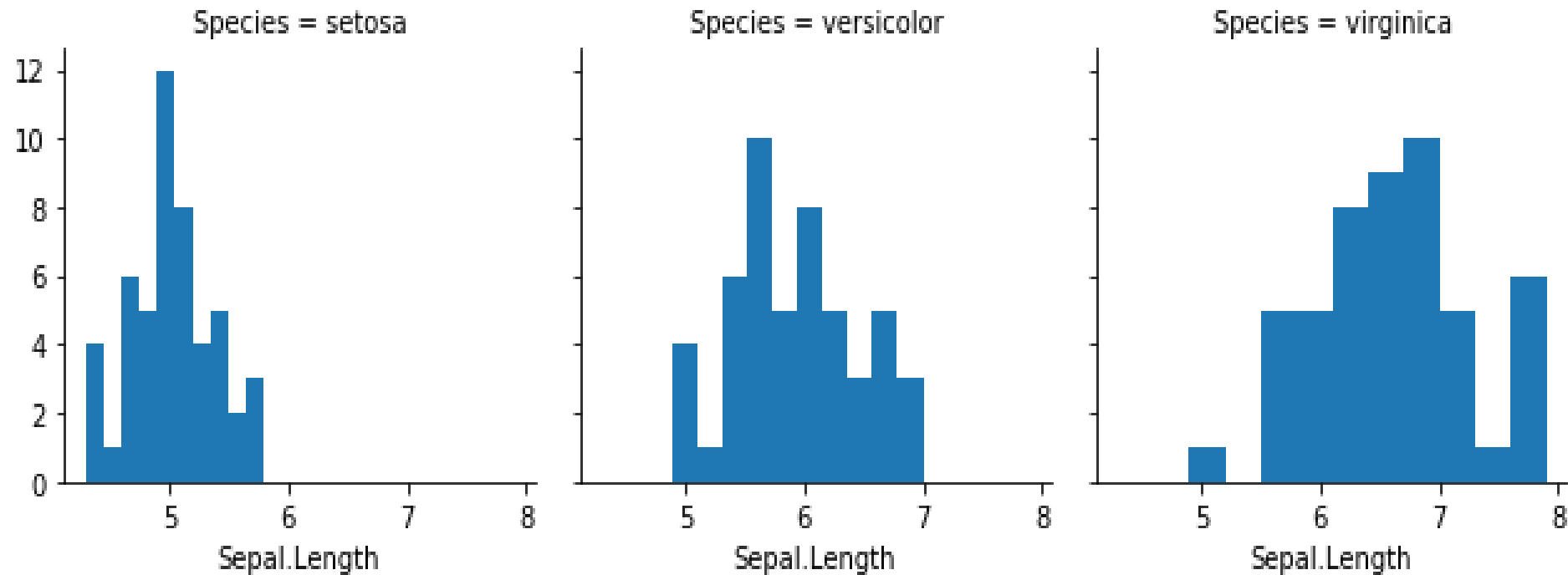


```
In [36]: sns.regplot(x='Sepal.Length', y='Sepal.Width', data=iris)  
...: plt.show()
```



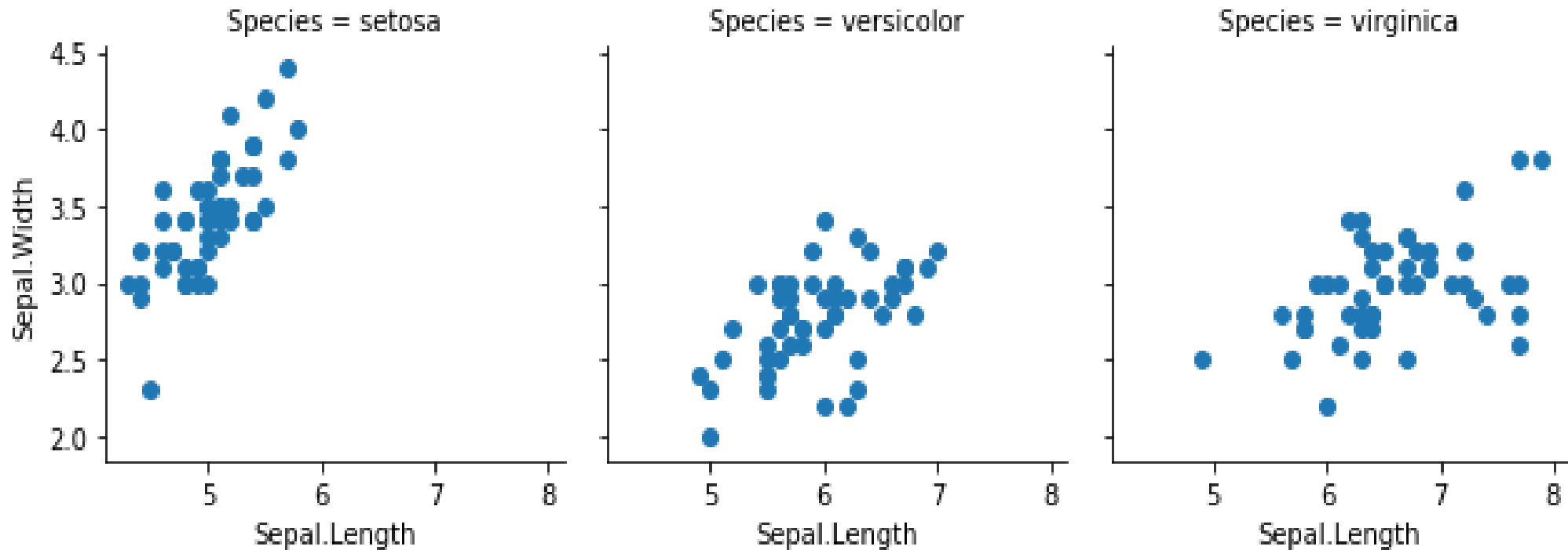
# Facet Grid

```
In [37]: g = sns.FacetGrid(iris, col="Species")  
....: g = g.map(plt.hist, "Sepal.Length")  
....: plt.show()
```



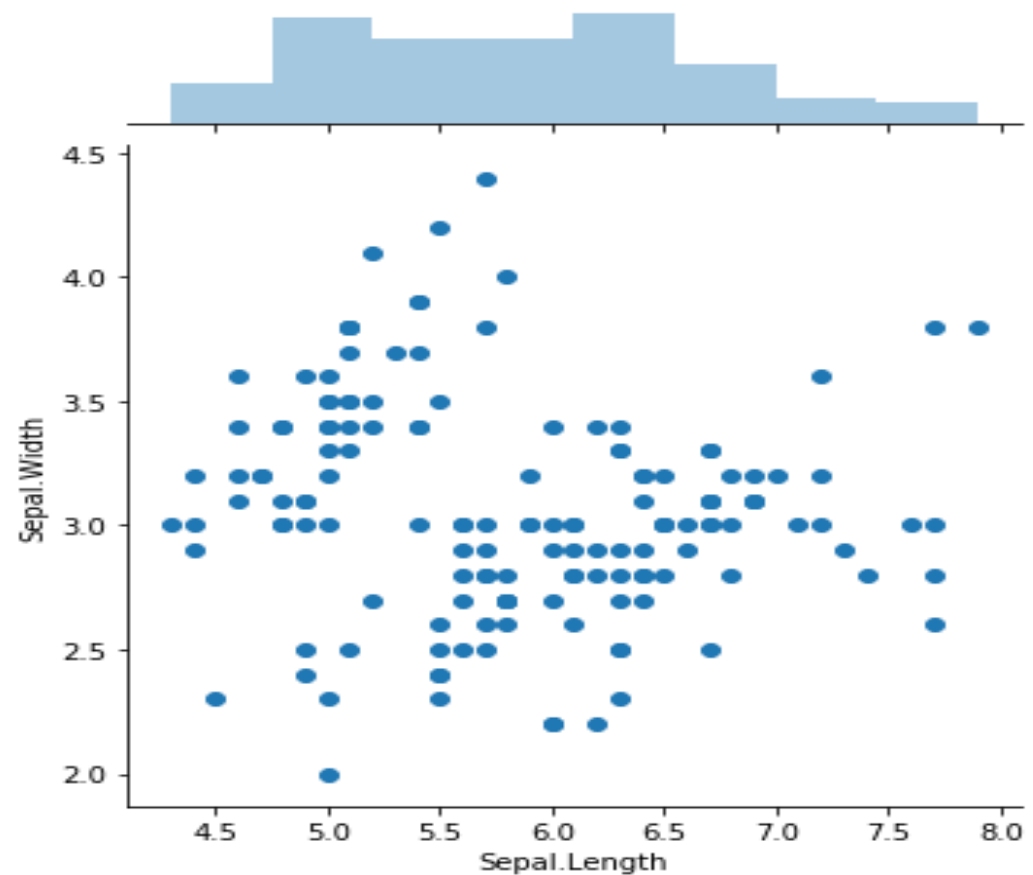
# Facet Grid

```
In [39]: g = sns.FacetGrid(iris, col="Species")  
....: g = g.map(plt.scatter, "Sepal.Length", "Sepal.Width")  
....: plt.show()
```



# Joint Plot

```
In [37]: sns.jointplot(x='Sepal.Length', y='Sepal.Width', data=iris)
...: plt.show()
```



# We covered...

- Using all the three options:
  - Bar Chart
  - Histogram
  - Density Plot
  - Scatter Plot
  - Boxplot
- Advanced Graphs:
  - Facet Grid
  - Sub Plot
  - Joint Plot



# Questions?