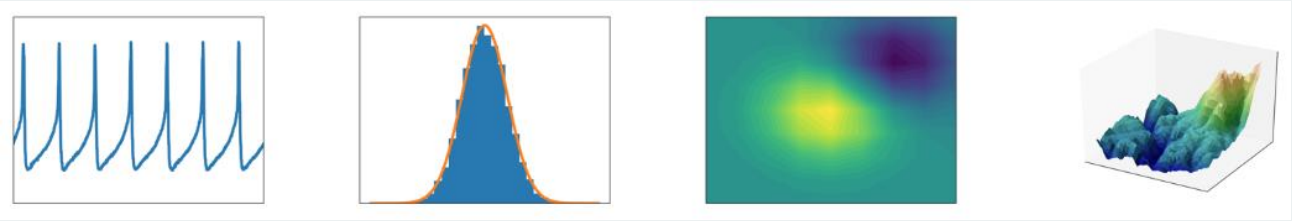
Data Visualization with Matplotlib

Foundations of AI Academy

matplotlib

Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms

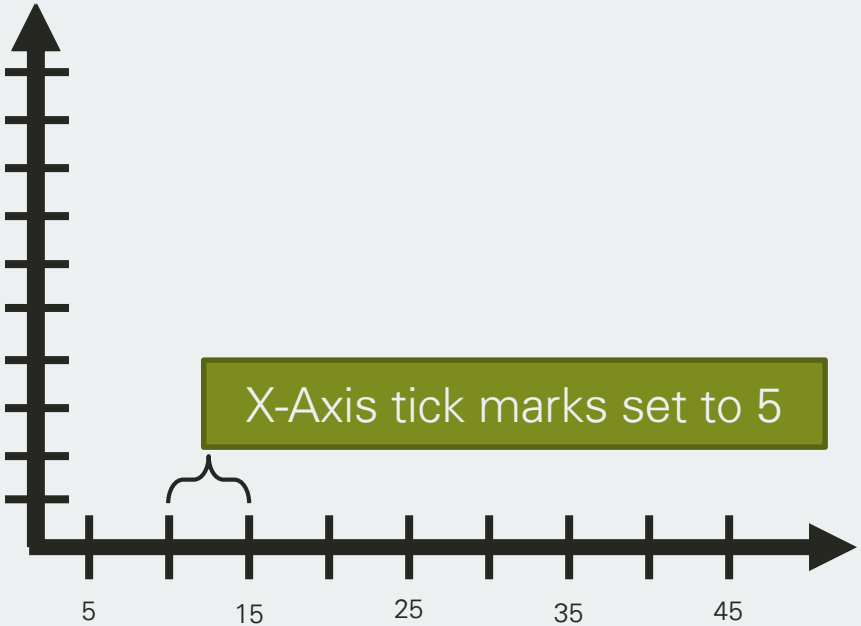
https://matplotlib.org/index.html



The Axes

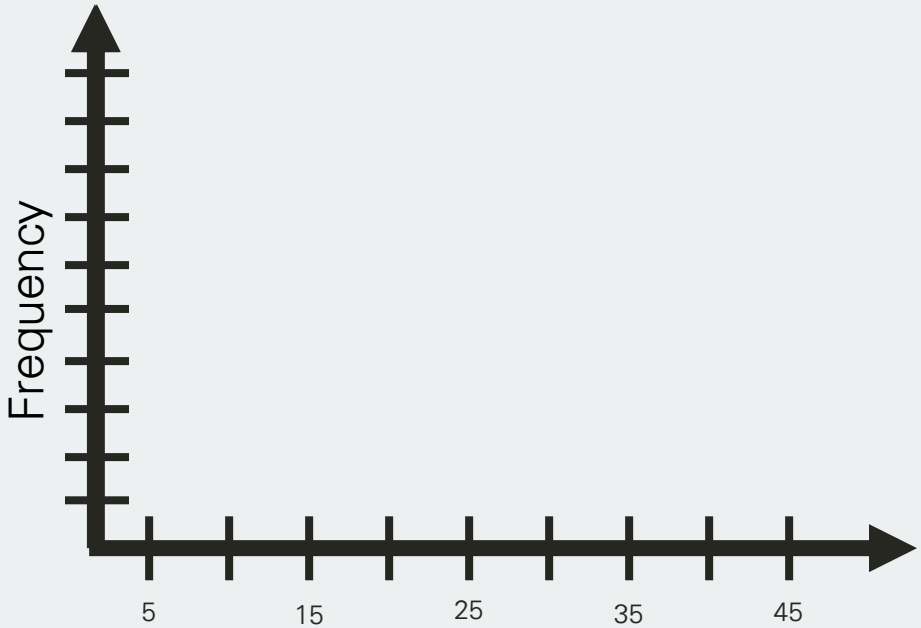
When you plot your data, values on the x-axis are measured across the y-axis

The Ticks



Tick marks denote the **rate** at which each axis grows

The Labels



Labels help define **what** you are visualizing

Pyplot

Designed to mirror functions from MATLAB

The most common usage of matplotlib

```
import matplotlib.pyplot as plt
```

Similar to normal imports, this says to import the pyplot API as plt

Plotting with Pyplot

```
plt.plot([1, 2, 3, 4]
```

Denotes the values plotted across the X-axis

Plotting with Pyplot

```
plt.plot([1, 2, 3, 4],[5, 6, 7, 8])
```

Denotes the values plotted across the Y-axis

Plotting with Pyplot

```
plt.plot([1, 2, 3, 4],[5, 6, 7, 8])
plt.ylabel('Frequency')
```

Denotes the label for the Y-axis

Plotting with Pyplot

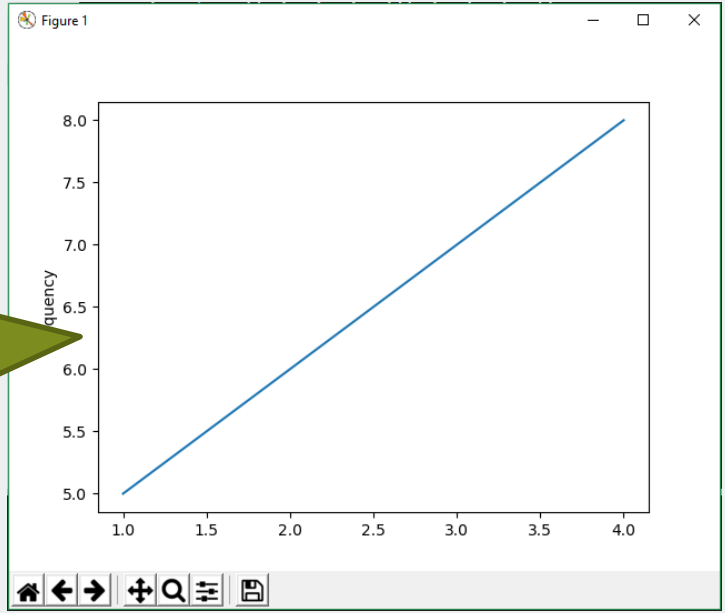
```
plt.plot([1, 2, 3, 4],[5, 6, 7, 8])
plt.ylabel('Frequency')
plt.show()
```

Creates the chart

Plotting with Pyplot

```
plt.plot([1, 2, 3, 4],[5, 6, 7, 8])
plt.ylabel('Frequency')
plt.show()
```

Displays a "chart window" that allows you to zoom, pan, or save the chart

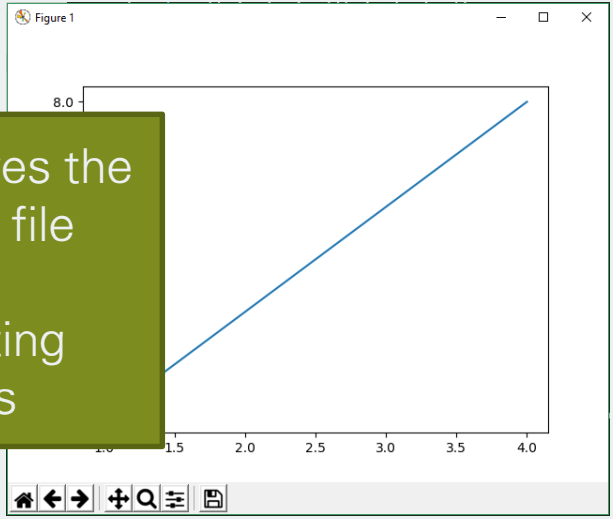


Plotting with Pyplot

```
plt.plot([1, 2, 3, 4],[5, 6, 7, 8])
plt.ylabel('Frequency')
plt.savefig('chart.png')
```

Automatically saves the chart to a PNG file

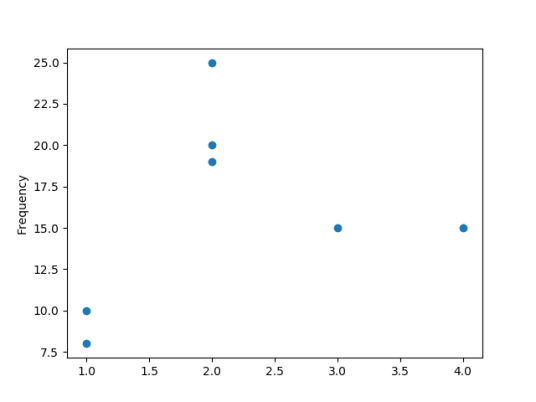
Good for creating multiple files



Plotting with Pyplot

```
x = [ 1,  2,  2,  2, 1,  3,  4]
y = [10, 20, 25, 19, 8, 15, 15]
plt.scatter(x, y)
plt.ylabel('Frequency')
plt.show()
```

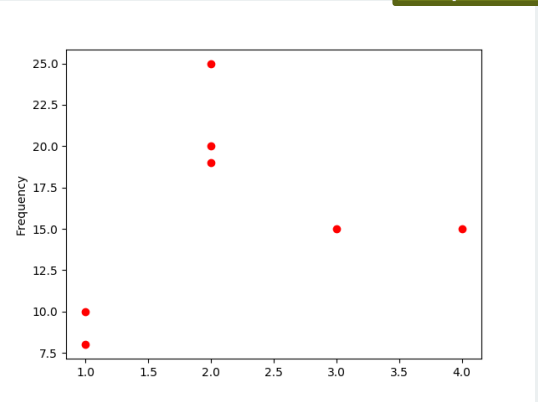
Changing from .plot to .scatter will convert your chart into a scatter plot



Plotting with Pyplot

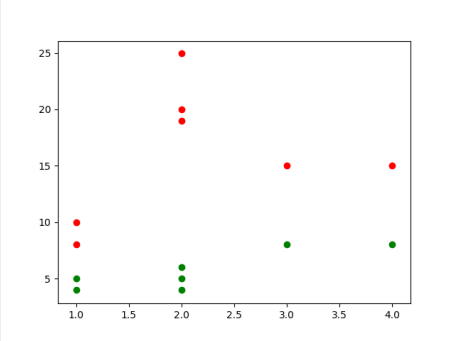
```
x = [ 1,  2,  2,  2, 1,  3,  4]
y = [10, 20, 25, 19, 8, 15, 15]
plt.scatter(x, y, color='r')
plt.ylabel('Frequency')
plt.show()
```

We can also add the color parameter to specify color schemes



Plotting with Pyplot

```
x = [ 1,  2,  2,  2, 1,  3,  4]
y = [10, 20, 25, 19, 8, 15, 15]
plt.scatter(x, y, color='r')
x = [ 1,  2,  2,  2, 1,  3,  4]
y = [ 5,  5,  5,  5, 5,  8,  8]
plt.scatter(x, y, color='green')
plt.show()
```

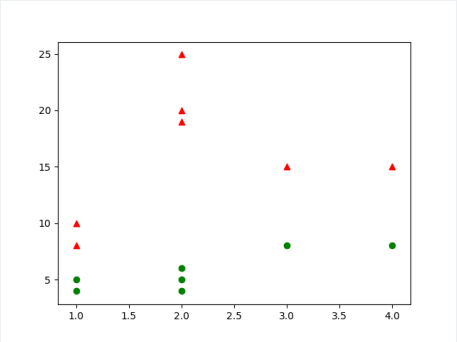


Multiple scatters can be used to visualize different data points

Plotting with Pyplot

```
x = [ 1,  2,  2,  2, 1,  3,  4]
y = [10, 20, 25, 19, 8, 15, 15]
plt.scatter(x, y, color='r', marker='^')
x = [ 1,  2,  2,  2, 1,  3,  4]
y = [ 5,  5,  5,  5, 5,  8,  8]
plt.scatter(x, y, color='green')
plt.show()
```

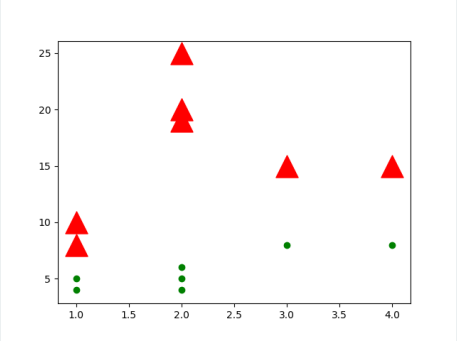
Can also specify different types of **markers** for data points



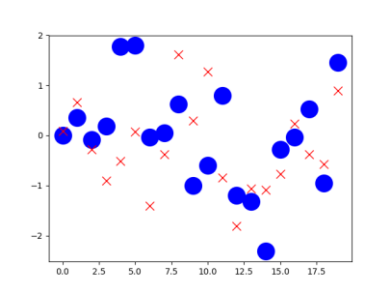
Plotting with Pyplot

```
x = [ 1,  2,  2,  2, 1,  3,  4]
y = [10, 20, 25, 19, 8, 15, 15]
plt.scatter(x, y, color='r', marker='^', s=500)
x = [ 1,  2,  2,  2, 1,  3,  4]
y = [ 5,  5,  5,  5, 5,  8,  8]
plt.scatter(x, y, color='green')
plt.show()
```

Also size



Plotting with Pyplot

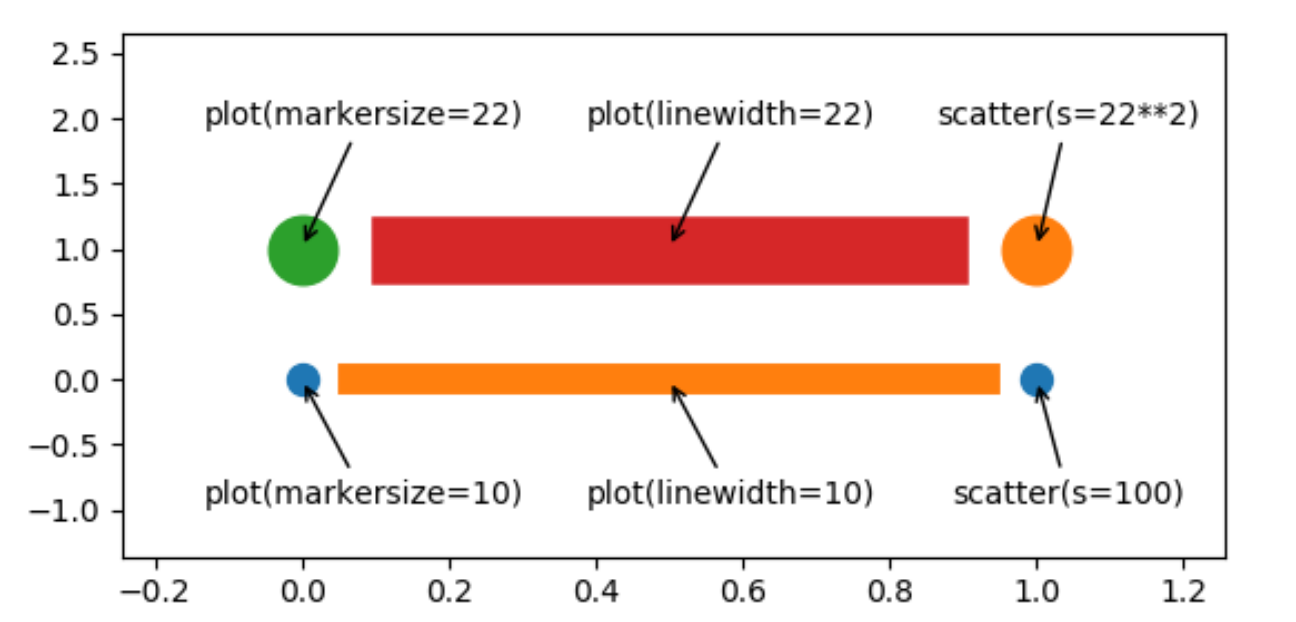



```
import numpy as np
import matplotlib.pyplot as plt
x1 = np.random.randn(20)
x2 = np.random.randn(20)
# you can specify the marker size two ways:
# blue circle with size 10
plt.plot(x1, 'bo', markersize=20)
# ms is an alias for markersize
plt.plot(x2, 'rx', ms=10)
plt.show()
```

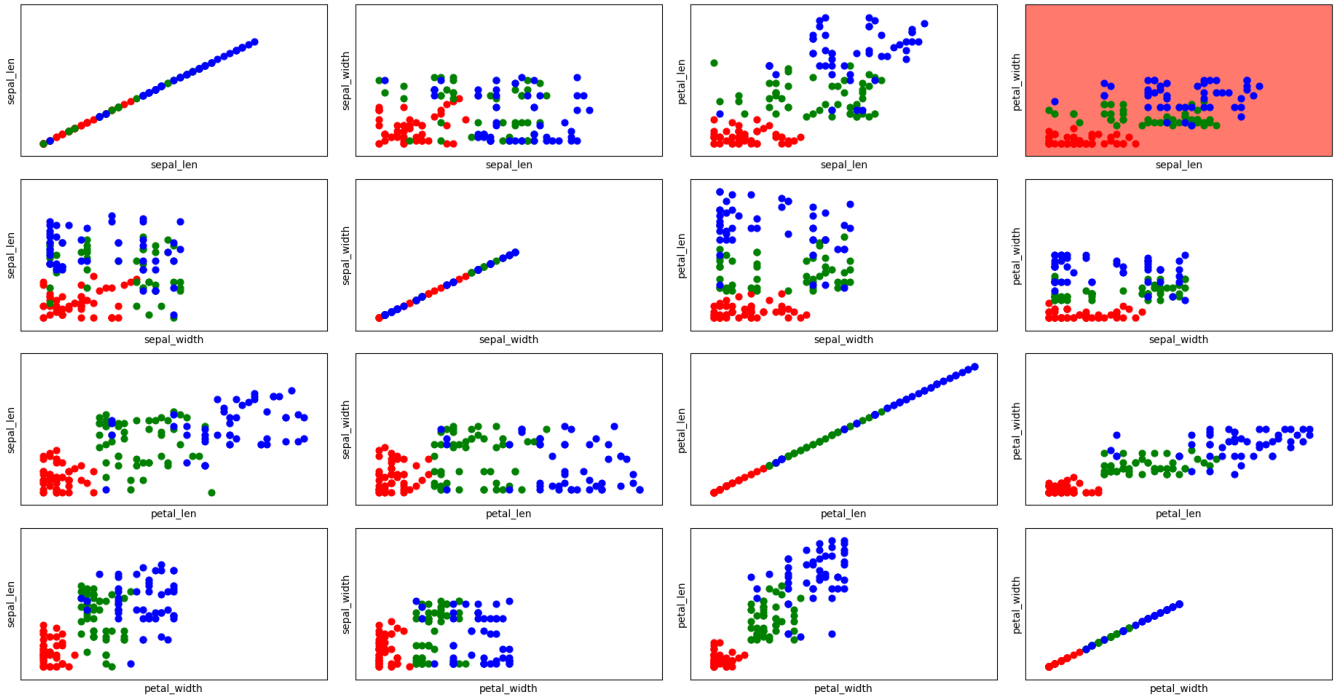
You can also scatter with .plot and a string with marker

Example

https://stackoverflow.com/a/47403507/1558159



Subplots



Subplots

Allow you to generate multiple charts with a single

```
fig, axes = plt.subplots(4,4, sharex=True, sharey=True)
```

4 Rows

Subplots

Allow you to generate multiple charts with a single

```
fig, axes = plt.subplots(4,4, sharex=True, sharey=True)
```

4 Columns

Subplots

Allow you to generate multiple charts with a single

```
fig, axes = plt.subplots(4,4, sharex=True, sharey=True)
```

That all share the same XY Axes

Subplots

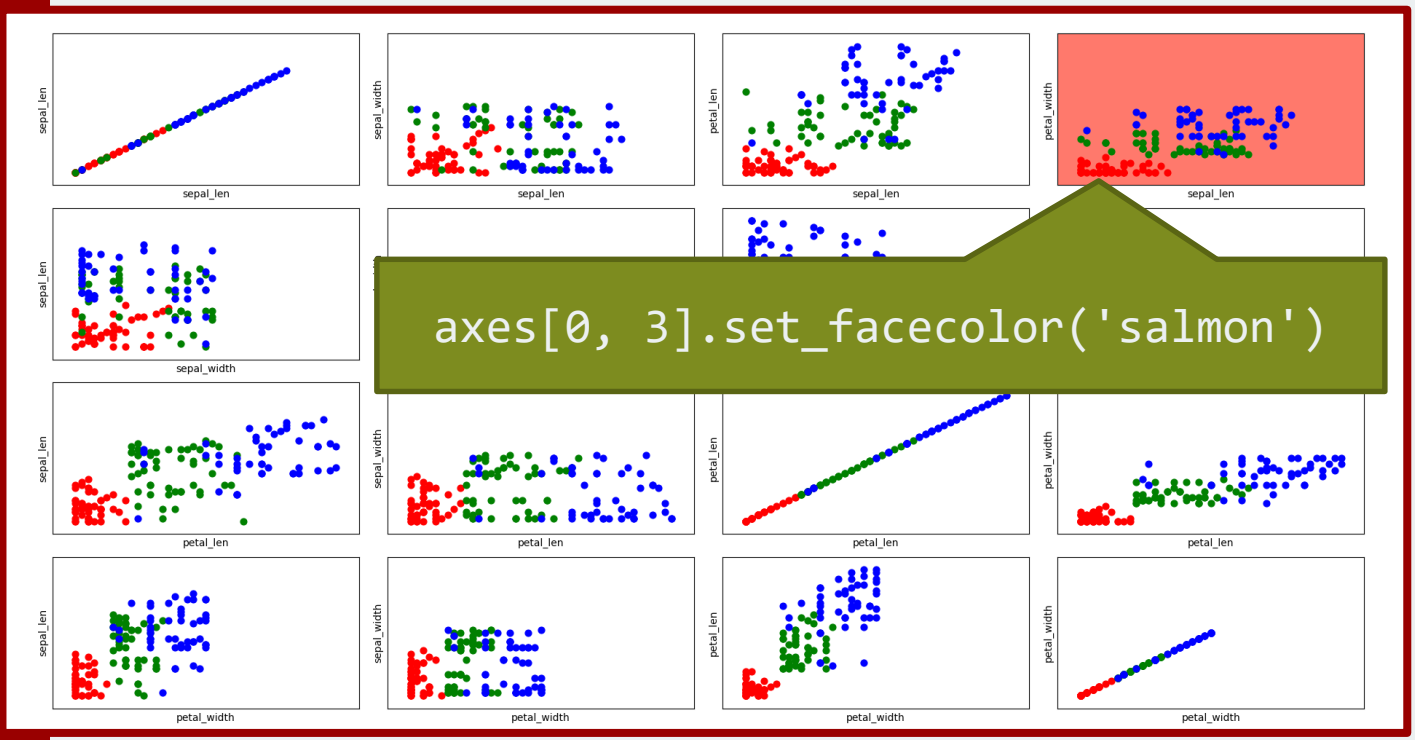
Allow you to generate multiple charts with a single

```
fig, axes = plt.subplots(4,4, sharex=True, sharey=True)
```

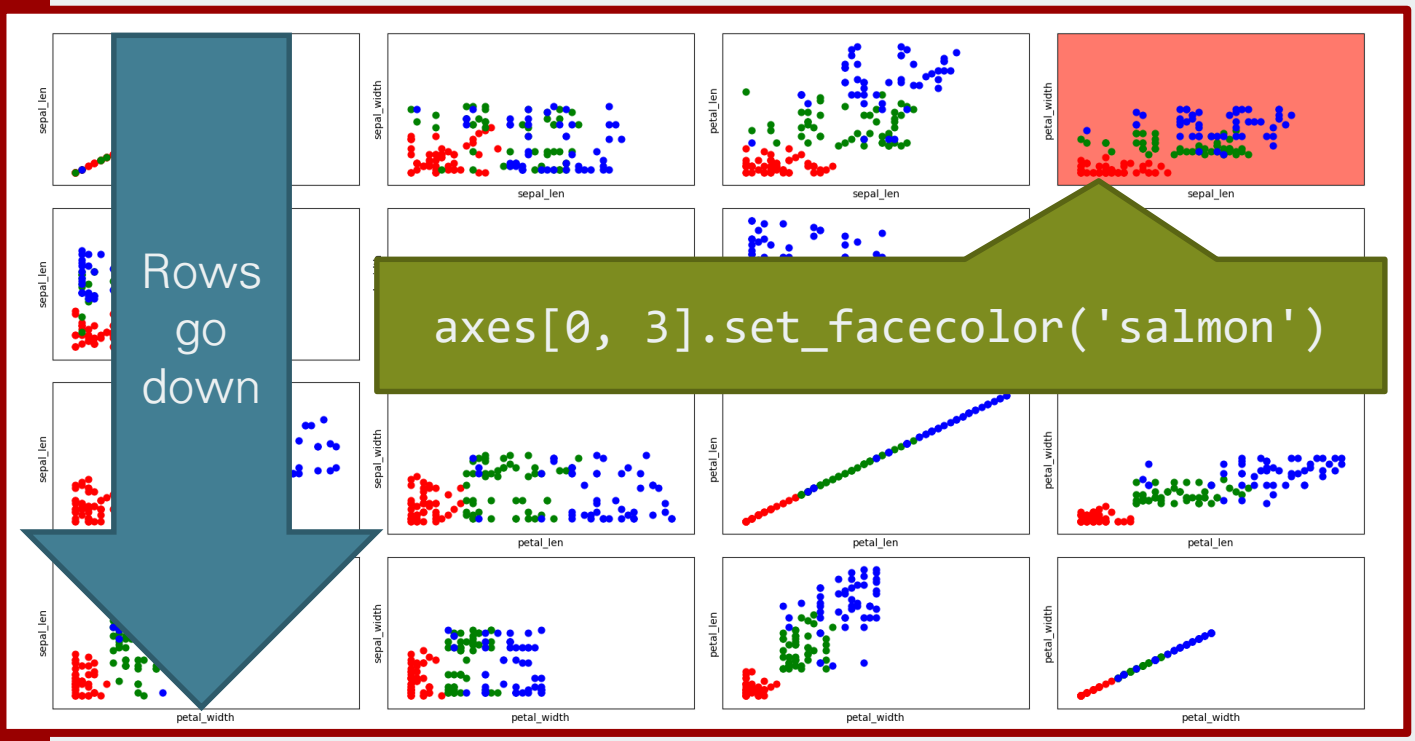
Gets unpacked into two variables

`fig` – the overall figure
`axes` – a **dictionary** where the individual subplots are referenced by a tuple `[row, col]`

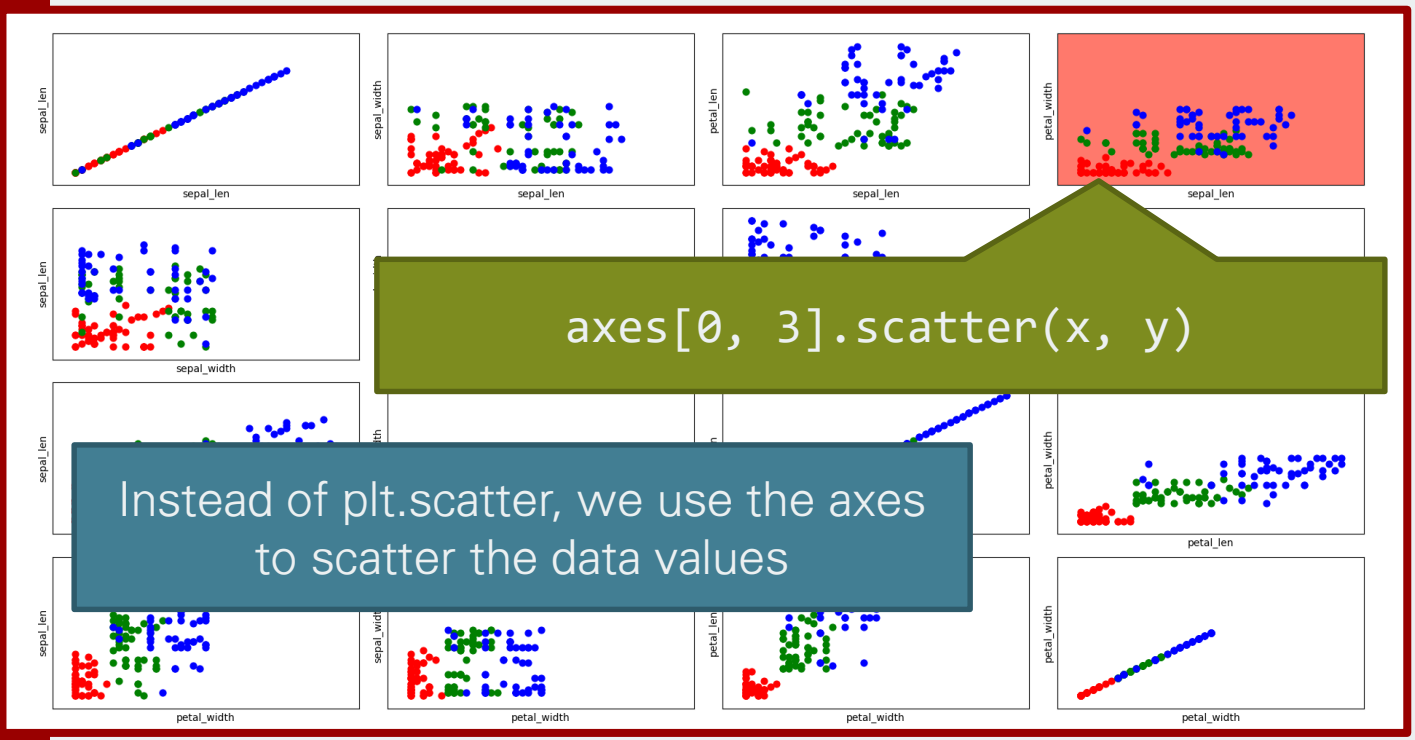
Subplots


sepal_len
sepal_width
petal_len
petal_width
axes[0, 3].set_facecolor('salmon')

Subplots


Rows go down
axes[0, 3].set_facecolor('salmon')
sepal_len
sepal_width
petal_len
petal_width

Subplots


sepal_len
sepal_width
petal_len
petal_width
axes[0, 3].scatter(x, y)
Instead of plt.scatter, we use the axes to scatter the data values