Matplotlib Basics

Matplotlib

- 2-D plotting library
- Can produce figures in a variety of formats.
- Can generate
  - Plots
  - Histogram
  - Bar charts
  - Scatterplots
  - and so on.

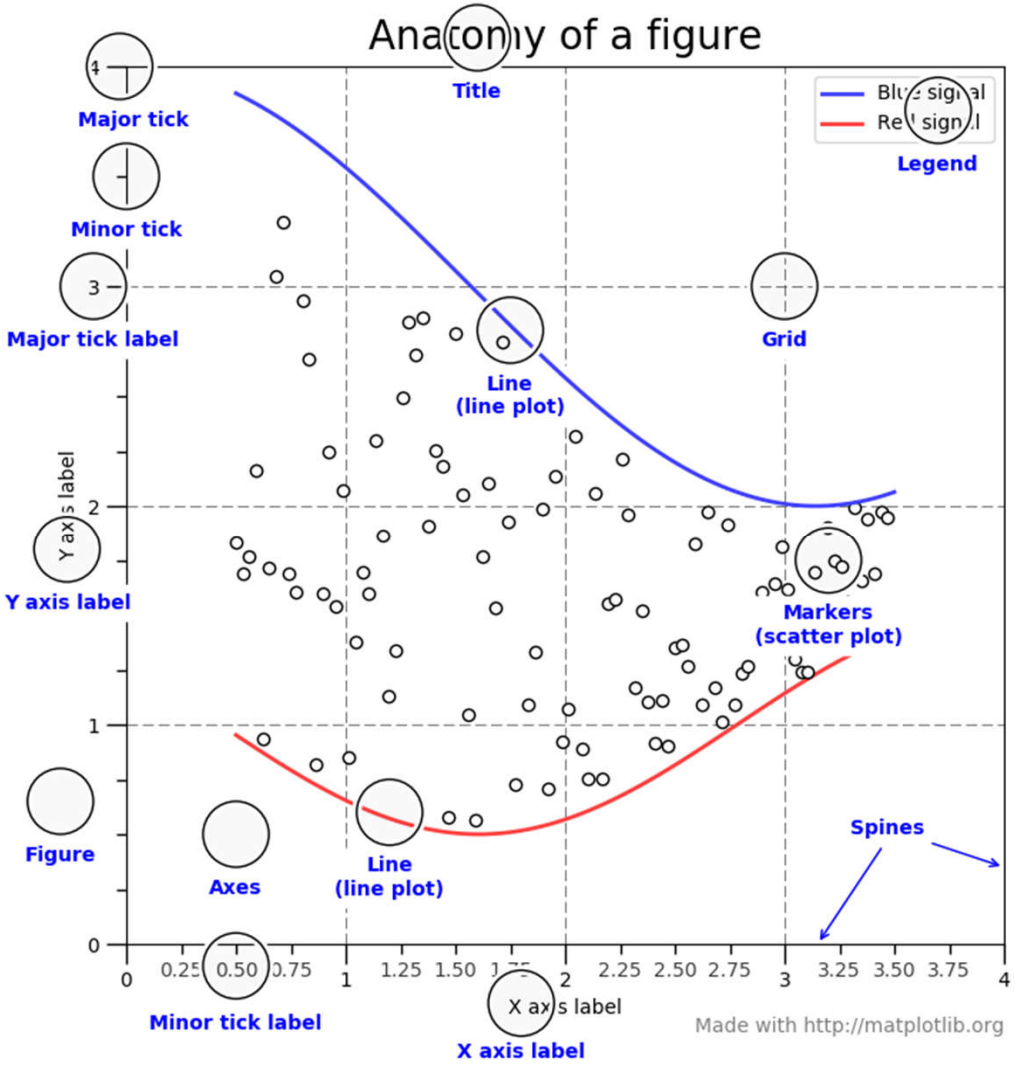
General Concepts

- Matplotlib is organized in a hierarchy.
- At the top of the hierarchy, is the matplotlib "state-machine environment"
  - Provided by pyplot
  - Use simple functions to add plot elements to current figure

General Concepts

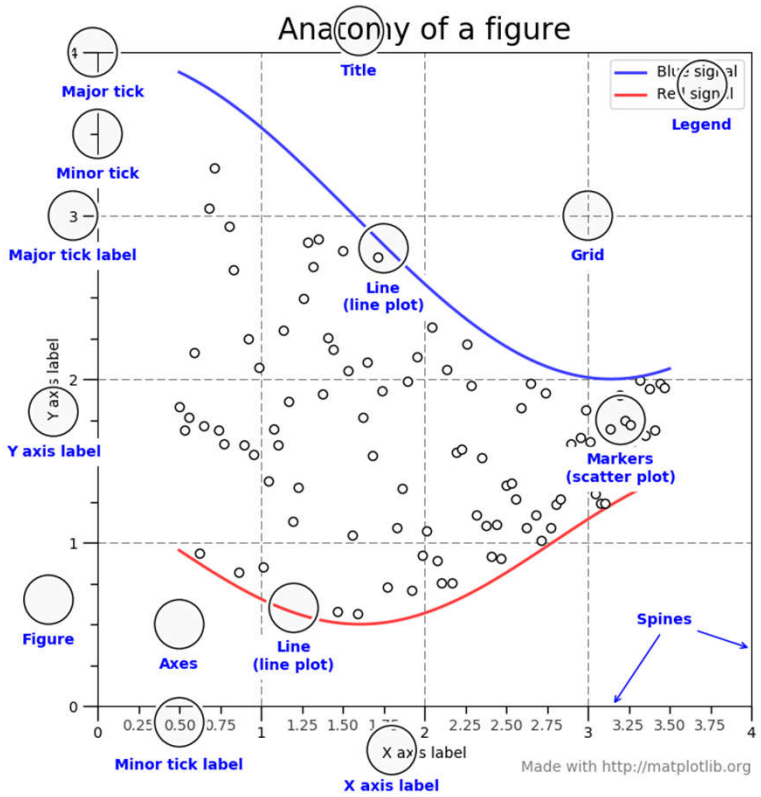
- At the next level of the hierarchy, is the matplotlib object-oriented interface
  - Minimal use of pyplot.
  - Only to create figure and axes objects
  - These objects are used to perform the plotting actions
- For even more control pyplot may be dropped and pure OOP approach can be followed.

Parts of a Figure



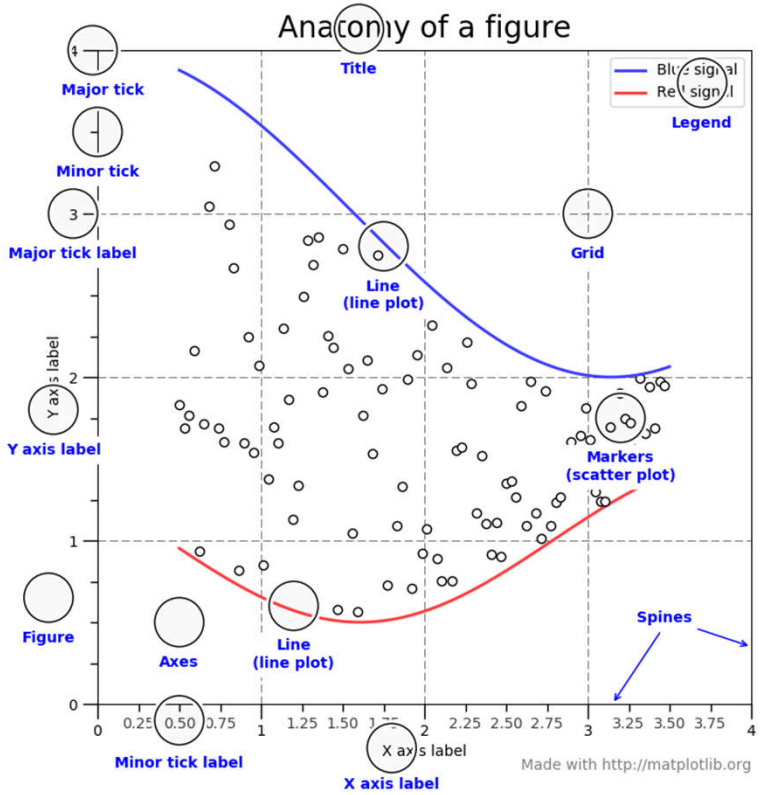
Parts of a Figure

- Figure
  - Keeps track of the child Axes, ‘special’ artists and the canvas
  - Can have any number of Axes



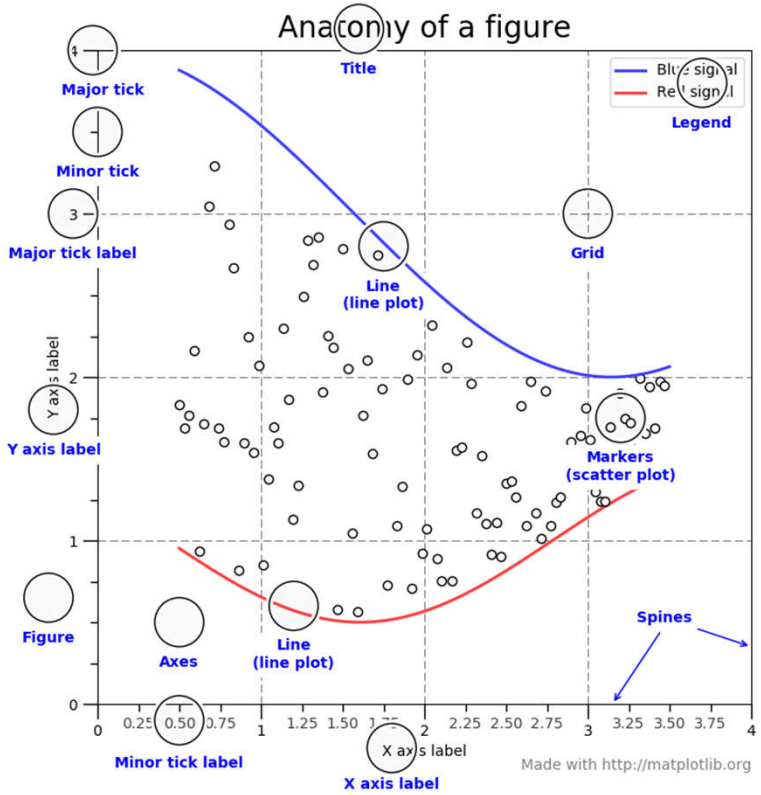
Parts of a Figure

- Axes
  - The region of the image with the data space.
  - One figure can have many axes
  - Each axes has a title, x-label, y-label



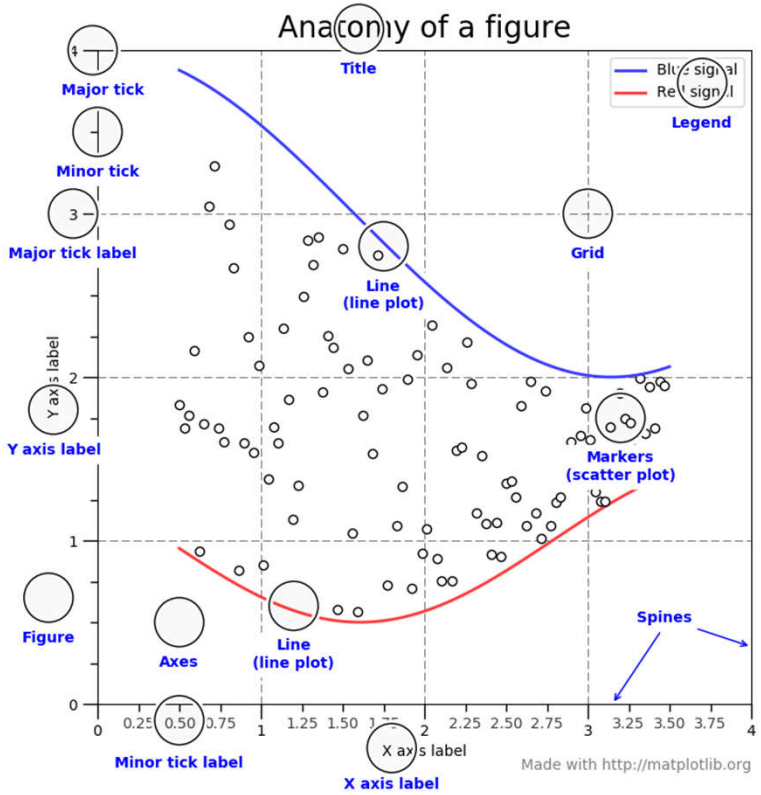
Parts of a Figure

- Axis
  - Number-line like objects
  - Sets the graph limit
  - Generates ticks
  - Location of ticklabels are determined by Locator
  - Format of ticklabels are determined by Formatter



Parts of a Figure

- Artist
  - Keeps track of the child Axes, ‘special’ artists and the canvas
  - Can have any number of Axes



Matplotlib

- One of three approaches are usually followed while using matplotlib.
  - Using pyplot API
  - Using object-oriented API
  - Using pylab API (Now deprecated)

Types of inputs

- All plotting functions expects
  - np.array
  - np.ma.masked_array
  - pandas and np.matrix may of may not work (better to convert np.array)

```
>>> b = np.matrix([[1,2],[3,4]])
>>> b
matrix([[1, 2],
        [3, 4]])
>>> b_asarray = np.array(b)
>>> b_asarray
array([[1, 2],
       [3, 4]])
>>>
```

Interactive vs. Non-interactive Mode

What happens if we do the following?

```
import matplotlib.pyplot as plt
plt.ion()
plt.plot([1.6, 2.7])
```

```
import matplotlib.pyplot as plt
plt.ioff()
plt.plot([1.6, 2.7])
```

pyplot API

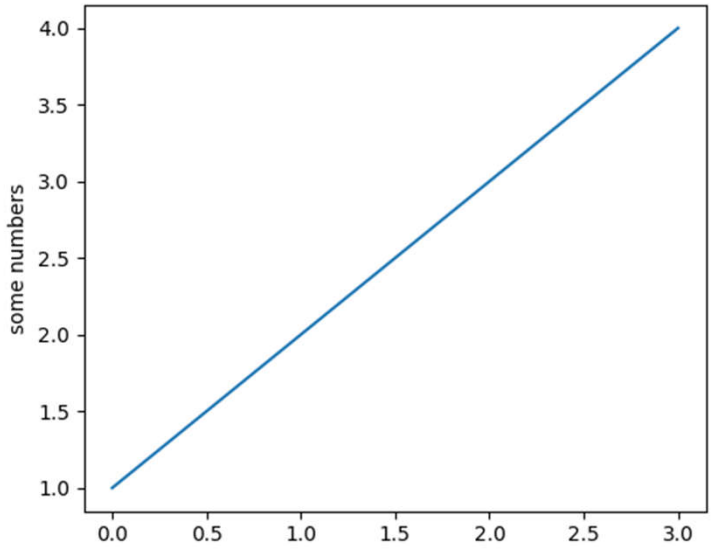
- Collection of commans style functions
- Each function makes some changes to a figure
  - Create a figure
  - Create a plotting area
  - Plot lines etc.
- Mainly intended for interactive plots.

Line plot

- plot()
  - Takes arbitrary number of arguments
  - Can be used to plot x vs. y
  - For every x, y pair of arguments, optional third argument is the format string that indicates the color and line type of the plot.
  - Format has two parts.
    - Color string,
    - Line style string.

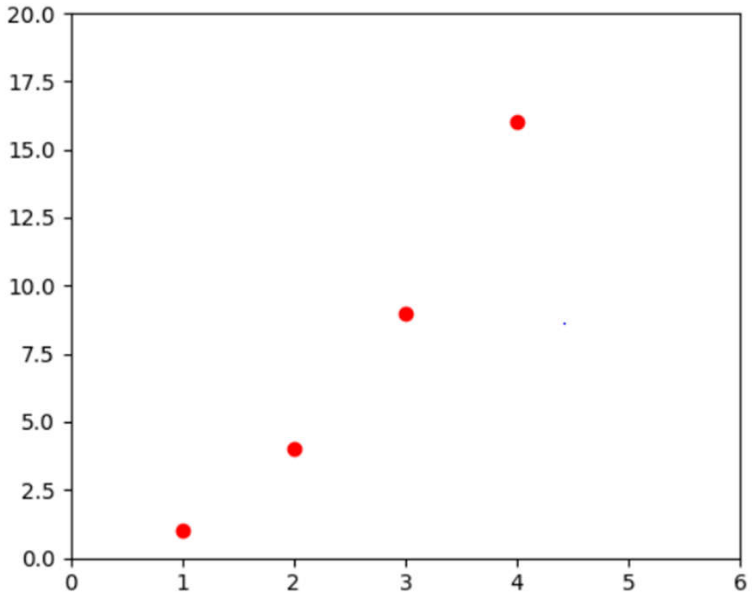
Line plot

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])
plt.ylabel('some numbers')
plt.show()
```



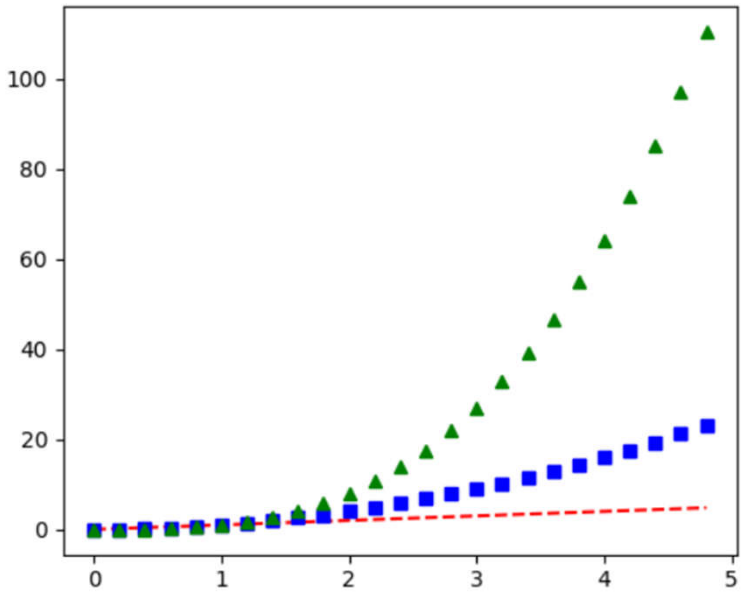
Line plot

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4], [1,4,9,16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```



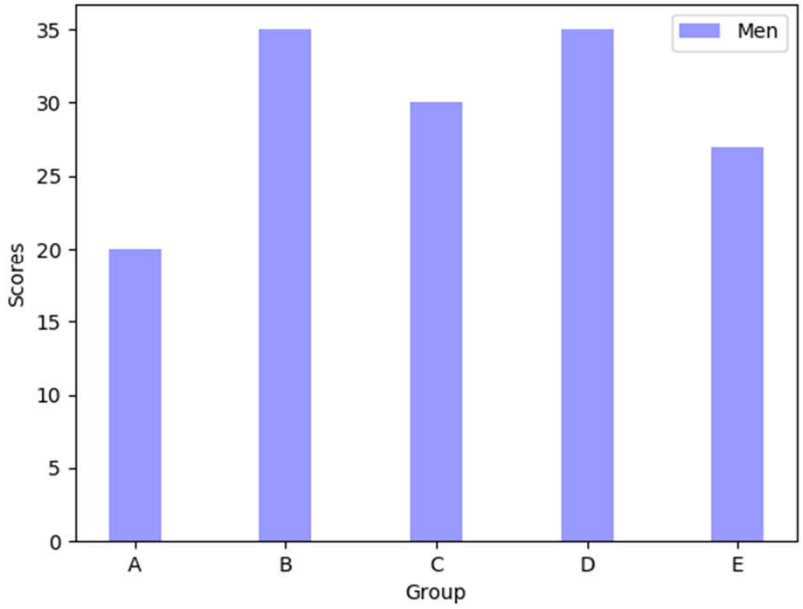
Line plot

```
import numpy as np
import matplotlib.pyplot as plt

# evenly sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)

# red dashes, blue squares and green triangles
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```



Bar Plot

```
import numpy as np
import matplotlib.pyplot as plt

n_groups = 5

means_men = (20, 35, 30, 35, 27)
fig, ax = plt.subplots()

index = np.arange(n_groups)
bar_width = 0.35

opacity = 0.4

rects1 = ax.bar(index, means_men, bar_width,
                alpha=opacity, color='b',
                label='Men')

ax.set_xlabel('Group')
ax.set_ylabel('Scores')
ax.set_xticks(index)
ax.set_xticklabels(('A', 'B', 'C', 'D', 'E'))
ax.legend()

plt.show()
```



Object-oriented API

- Greater control over the figure.
- First create some objects and then work on those objects.

Referece

- https://matplotlib.org/