Comp 3005: Data Science Bridge Course: Computer Science Programming Basics

Class 13: Data Visualization

Dalton Crutchfield
University of Denver

Introduction

Different Options

- Matplotlib
  - Primary; built on Numpy and Pandas
  - Static rendering
- Seaborn
  - Built on top of Matplotlib; simpler API
  - Robust use of color; built-in datasets; use of pandas
- Bokeh
  - Interactive rendering in web browser
  - Emits HTML with CSS, Javascript
- Plotnine
  - Supports a “grammar of graphics” akin to R’s ggplot2

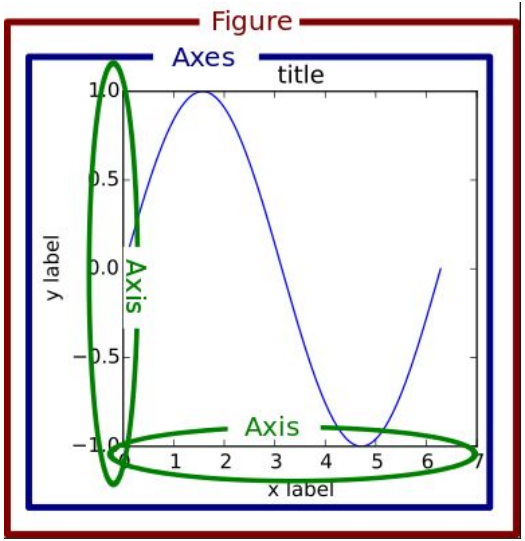
Imports

•Often we will see these imports, though only the last is needed to plot data:

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

Format of Visualization

•In Matplotlib (among others), this is how we will refer to the chart:



Matplotlib Basics

Plotting Data

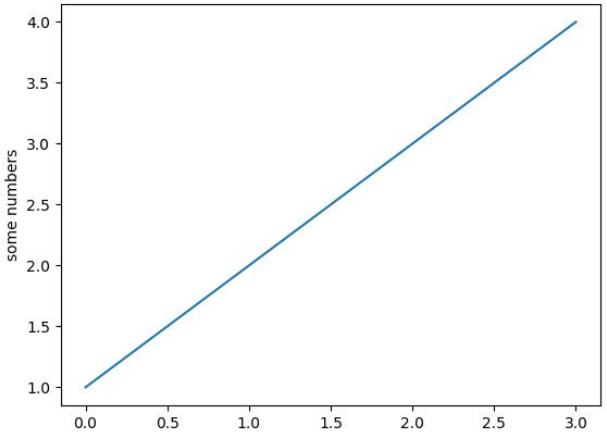
- To generate a basic plot you need to:
  - Have some data that you are plotting
  - Tell python to plot the data
  - Tell python to show the chart

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

Plotting Data

- To generate a basic plot you need to:
  - Have some data that you are plotting
  - Tell python to plot the data
  - Tell python to show the chart

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



Plot x vs. y

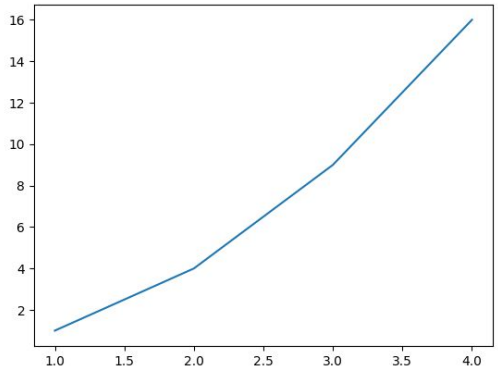
• plot is a versatile function, and will take an arbitrary number of arguments. For example, to plot x versus y, you can write:

```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
```

Plot x vs. y

• plot is a versatile function, and will take an arbitrary number of arguments. For example, to plot x versus y, you can write:

```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
```



Basic Formatting

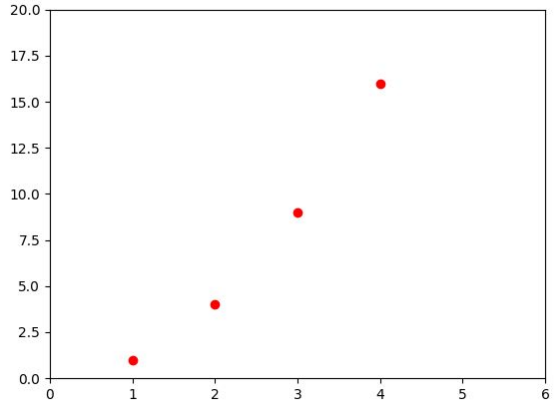
• For every x, y pair of arguments, there is an optional third argument which is the format string that indicates the color and line type of the plot.
• The letters and symbols of the format string are from MATLAB, and you concatenate a color string with a line style string.
• The default format string is 'b-', which is a solid blue line. For example, to plot the above with red circles, you would issue

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

Basic Formatting

• For every x, y pair of arguments, there is an optional third argument which is the format string that indicates the color and line type of the plot.
• The letters and symbols of the format string are from MATLAB, and you concatenate a color string with a line style string.
• The default format string is 'b-', which is a solid blue line. For example, to plot the above with red circles, you would issue

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



Using Lists

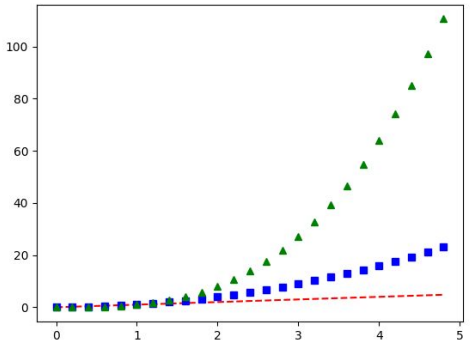
• If matplotlib were limited to working with lists, it would be fairly useless for numeric processing.
• Generally, you will use **numpy arrays** (In fact, all sequences are converted to numpy arrays internally).
• We can use these arrays to plot multiple lines at once:

```
import numpy as np

# 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()
```

Using Lists

• If matplotlib were limited to working with lists, it would be fairly useless for numeric processing.
• Generally, you will use **numpy arrays** (In fact, all sequences are converted to numpy arrays internally).
• We can use these arrays to plot multiple lines at once:

```
import numpy as np

# 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()
```



Subplots

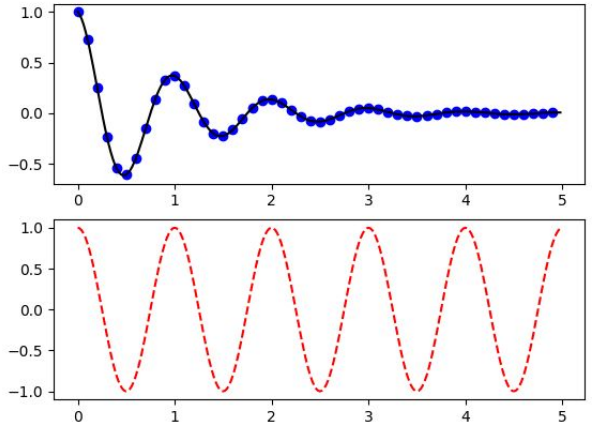
- MATLAB, and pyplot, have the concept of the current figure and the current axes.
- All plotting functions apply to the current axes.
- The function gca returns the current axes (a matplotlib.axes.Axes instance),
- gcf returns the current figure (a matplotlib.figure.Figure instance).
- Below is a script to create two subplots.

```
def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)

plt.figure()
plt.subplot(211)
plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')

plt.subplot(212)
plt.plot(t2, np.cos(2*np.pi*t2), 'r--')
plt.show()
```

Subplots

- MATLAB, and pyplot, have the concept of the current figure and the current axes.
- All plotting functions apply to the current axes.
- The function gca returns the current axes (a matplotlib.axes.Axes instance),
- gcf returns the current figure (a matplotlib.figure.Figure instance).
- Below is a script to create two subplots.

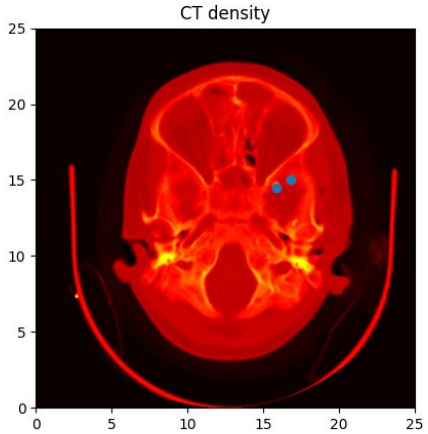
```
def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)

plt.figure()
plt.subplot(211)
plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')

plt.subplot(212)
plt.plot(t2, np.cos(2*np.pi*t2), 'r--')
plt.show()
```



Other possibilities

- **Images**

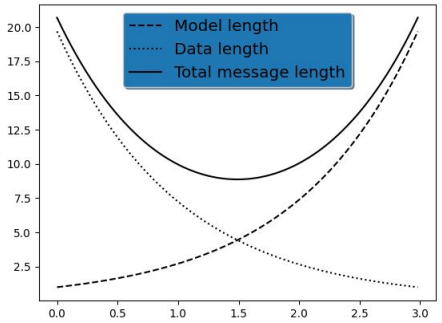
  •Matplotlib can display images (assuming equally spaced horizontal dimensions) using the imshow() function.



Other possibilities

- **Legends**

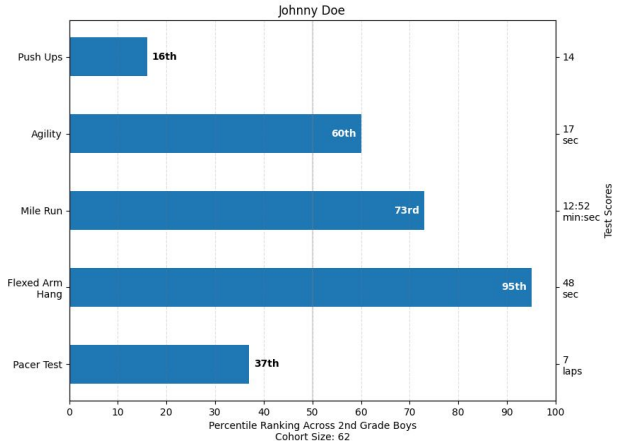
  - The legend() function automatically generates figure legends, with MATLAB-compatible legend-placement functions.



Other possibilities

- **Bar Charts**

  •Use the bar() function to make bar charts, which includes customizations such as error bars:

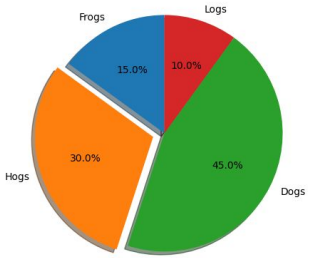


Other possibilities

- **Pie Charts**

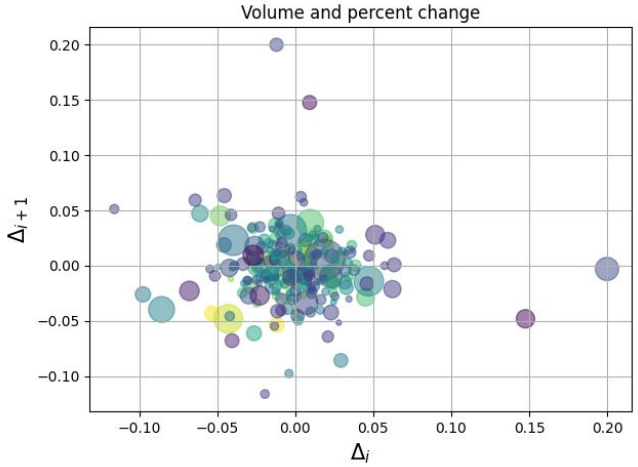
  - The pie() function allows you to create pie charts.

  - Optional features include auto-labeling the percentage of area, exploding one or more wedges from the center of the pie, and a shadow effect. Take a close look at the attached code, which generates this figure in just a few lines of code.



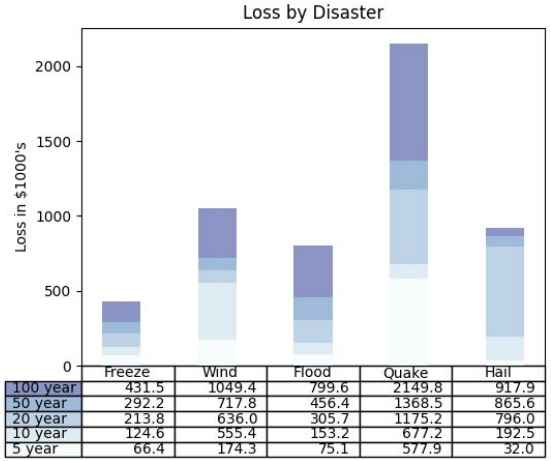
Other possibilities

- **Scatter Charts**
  - The scatter() function makes a scatter plot with (optional) size and color arguments.



Other possibilities

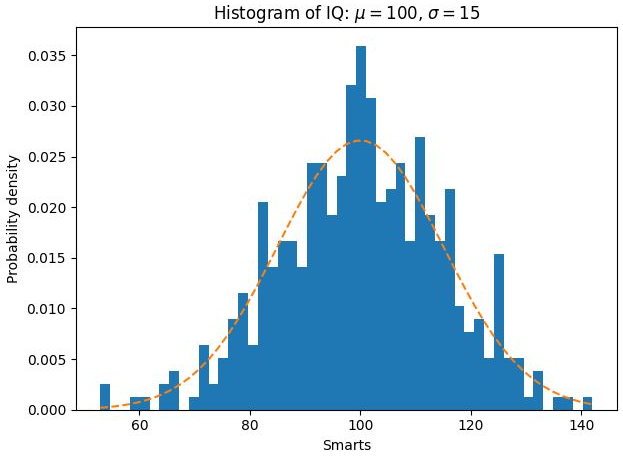
- **Tables**
  - The table() function adds a text table to an axes.



	Freeze	Wind	Flood	Quake	Hail
100 year	431.5	1049.4	799.6	2149.8	917.9
50 year	292.2	717.8	456.4	1368.5	865.6
20 year	213.8	636.0	305.7	1175.2	796.0
10 year	124.6	555.4	153.2	677.2	192.5
5 year	66.4	174.3	75.1	577.9	32.0

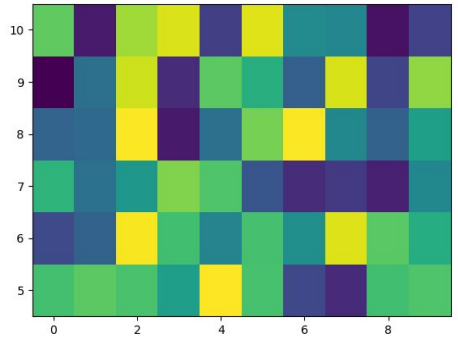
Other possibilities

- **Histograms**
  - The hist() function automatically generates histograms and returns the bin counts or probabilities:



Other possibilities

- **Contouring and pseudocolor**
  - The pcolormesh() function can make a colored representation of a two-dimensional array, even if the horizontal dimensions are unevenly spaced.
  - The contour() function is another way to represent the same data:



Other possibilities

- **And So Many More!**

Today in Tech History

October 26, 1861

Only two days after the Transcontinental Telegraph line opened, the Pony Express ceases operation. Prior to the opening of the cross-country telegraph line, the Pony Express was the fastest way to send communication between St. Jospeph, Missouri and San Franscisco, California.