

CME 193: Introduction to Scientific Python

Lecture 5: Data Visualization & Web Scraping

Dan Frank

Institute for Computational and Mathematical Engineering
(ICME)

January 23, 2014

Administrivia

- ▶ HW 4 due next class
- ▶ No HW 5
- ▶ Austin Benson will teach guest lecture

IDEs, Debugging, Version Control, etc.

Matplotlib

Web Scraping

Debugging: General Principles

How do I make my code bug free?

1. First, don't write bugs! (e.g. informative variable/function names, encapsulation, etc.)
2. Isolate the problem: module \Rightarrow function \Rightarrow variable
3. Be able to consistently reproduce the problem
4. Read the error messages
5. print statements
6. Explore the code interactively with a debugger

Debugging

I've written a function or script and it's giving me a weird error...
now what?

```
import numpy as np

def some_buggy_function():
    A = np.arange(1, 10)
    # import ipdb; ipdb.set_trace() # BREAKPOINT
    A /= 2.
    return np.sum(5 / A)
```

Debugging: ipdb

What can I do within the debugger?

1. n (next)
2. ENTER (repeat previous)
3. q (quit)
4. p `variable` (print value)
5. c (continue)
6. l (list where you are)
7. s (step into subroutine)
8. r (continue till the end of the subroutine)
9. plus anything you can normally do at a python terminal

Debugging: ipdb

```
danfrank@stiletto:~$ vim src/python-course/part-5/lecture-5/code/debug.py
danfrank@stiletto:~$ python src/python-course/part-5/lecture-5/code/debug.py
src/python-course/part-5/lecture-5/code/debug.py:7: RuntimeWarning: divide by zero encountered in divide
  return np.sum(5 / A)
danfrank@stiletto:~$ vim src/python-course/part-5/lecture-5/code/debug.py
danfrank@stiletto:~$ python src/python-course/part-5/lecture-5/code/debug.py
> /home/danfrank/src/python-course/part-5/lecture-5/code/debug.py(6)some_buggy_function()
5      import ipdb; ipdb.set_trace() # BREAKPOINT
----> 6      A /= 2.
7      return np.sum(5 / A)

ipdb> n
> /home/danfrank/src/python-course/part-5/lecture-5/code/debug.py(7)some_buggy_function()
6      A /= 2.
----> 7      return np.sum(5 / A)
8

ipdb> A
array([0, 1, 1, 2, 2, 3, 3, 4, 4])
ipdb> A.dtype
dtype('int64')
ipdb> l
2
3 def some_buggy_function():
4     A = np.arange(1, 10)
5     import ipdb; ipdb.set_trace() # BREAKPOINT
6     A /= 2.
----> 7     return np.sum(5 / A)
8
9 some_buggy_function()

ipdb> 
```

Not all software engineers use them but Integrated Development Environments can be very helpful in developing code. Functionality may include

- ▶ source code editor with syntax highlighting
- ▶ autocompletion and goto definitions
- ▶ debugging
- ▶ integration with terminal

No standard IDE for Python, but eclipse, emacs, vim, etc. can be configured. Each coder has their own setup. iPython provides many of these functions within a terminal.

Version Control

Version control enables different people working on the same code base to coordinate their efforts.

- ▶ maintain history of code development
- ▶ tracking differences between current versions and older versions
- ▶ create branches of code so different features can be worked on simultaneously and merged together later

There are many tools for version control but the standards are **subversion** and more recently **git**. This course is currently coordinated through github.com

IDEs, Debugging, Version Control, etc.

Matplotlib

Web Scraping

What is Matplotlib?

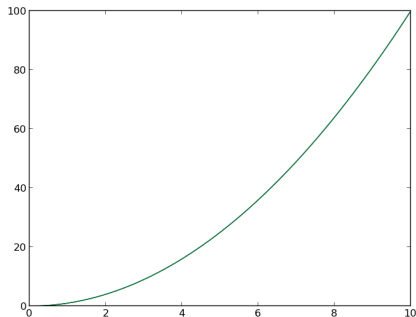
matplotlib.org: matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in python scripts, the python and ipython shell (ala MATLAB or Mathematica), web application servers, and six graphical user interface toolkits.

- ▶ matplotlib is the standard Python plotting library
- ▶ We will primarily be using `matplotlib.pyplot` for data analysis
- ▶ Can create histograms, power spectra, bar charts, errorcharts, scatterplots, etc with a few lines of code

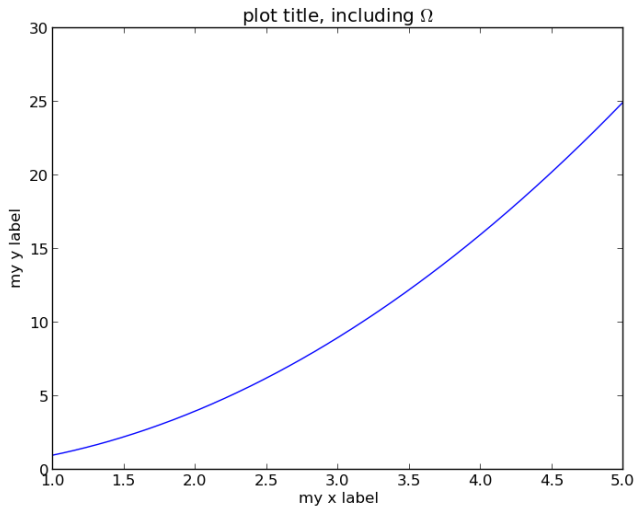
Scatter Plot

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

x = np.linspace(0, 10, 1000)
y = np.power(x, 2)
plt.plot(x, y)
plt.savefig('line_plot.png')
```



Scatter Plot+



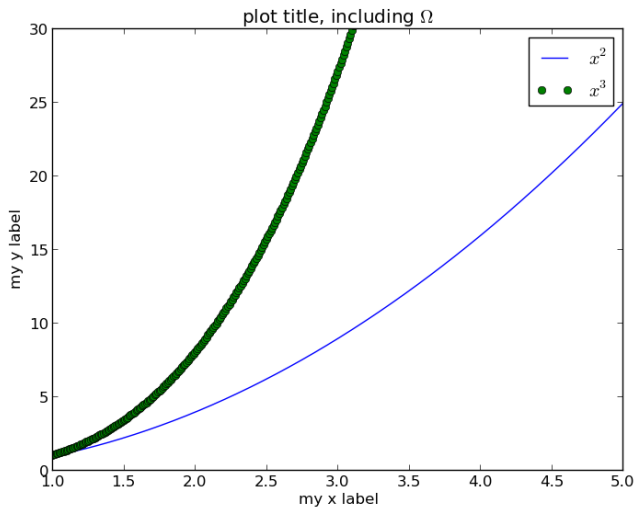
Scatter Plot+

Adding titles and labels

```
x = np.linspace(0, 10, 1000)
y = np.power(x, 2)
plt.plot(x, y)
plt.xlim((1, 5))
plt.ylim((0, 30))
plt.xlabel('my x label')
plt.ylabel('my y label')
plt.title('plot title, including  $\Omega$ ')

plt.savefig('line_plot_plus.png')
```

Scatter Plot++



Scatter Plot++

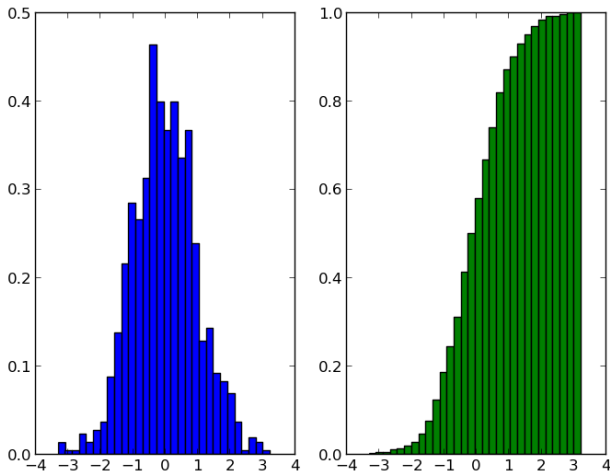
Adding multiple lines and a legend

```
x = np.linspace(0, 10, 1000)
y1 = np.power(x, 2)
y2 = np.power(x, 3)

plt.plot(x, y1, 'b-', x, y2, 'go')
plt.xlim((1, 5))
plt.ylim((0, 30))
plt.xlabel('my x label')
plt.ylabel('my y label')
plt.title('plot title, including  $\Omega$ ')
plt.legend(('x^2', 'x^3'))

plt.savefig('line_plot_plus2.png')
```


Histogram



Histogram

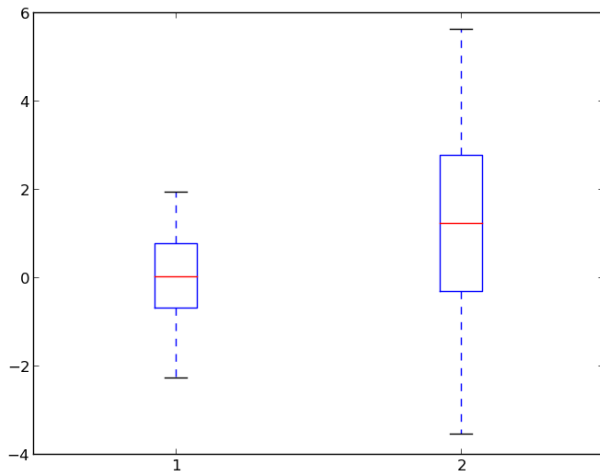
```
data = np.random.randn(1000)

# histogram (pdf)
plt.subplot(1, 2, 1)
plt.hist(data, bins=30, normed=True, facecolor='b')

# empirical cdf
plt.subplot(1, 2, 2)
plt.hist(data, bins=30, normed=True, color='g',
          cumulative=True)

plt.savefig('histogram.png')
```

Box Plot

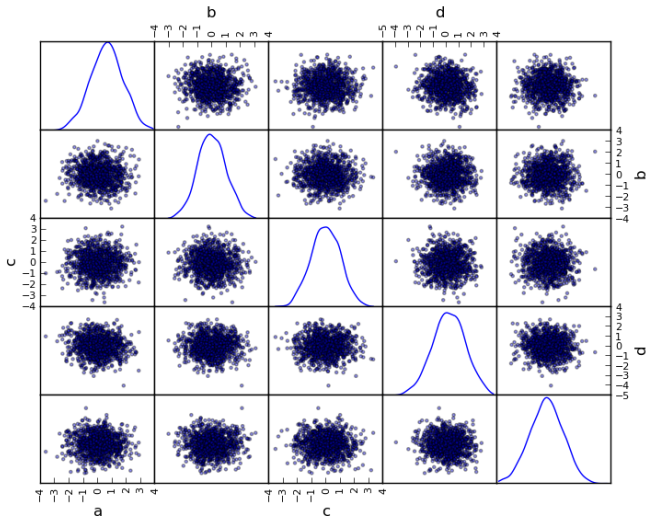


Box Plot

```
samp1 = np.random.normal(loc=0., scale=1., size=100)
samp2 = np.random.normal(loc=1., scale=2., size=100)

plt.boxplot((samp1, samp2))
plt.savefig('boxplot.png')
```

Scatter Plot Matrix



Scatter Plot Matrix

matplotlib doesn't have everything, especially functions that are designed to act on more than one axis at once.

```
from pandas.tools.plotting import scatter_matrix
from pandas import DataFrame

df = DataFrame(np.random.normal(loc=0.,
                                scale=1.,
                                size=(1000, 5)),
               columns=['a', 'b', 'c', 'd', 'e'])
scatter_matrix(df, alpha=0.4, diagonal='kde')

plt.savefig('scattermatrix.png')
```

Image Plot

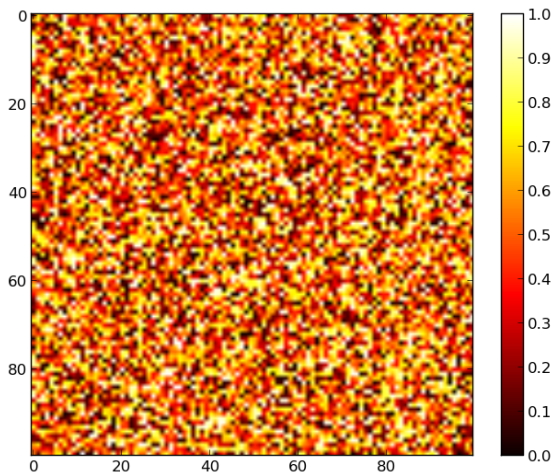
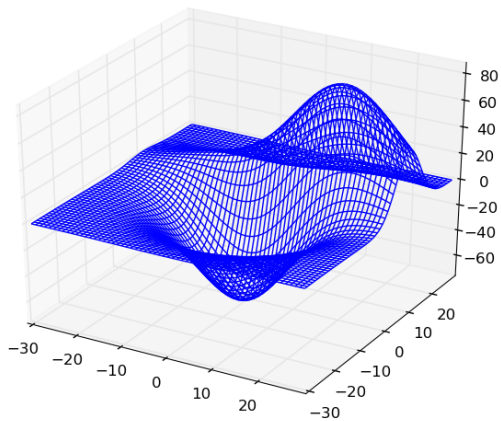


Image Plot

```
A = np.random.random((100, 100))  
plt.imshow(A)  
plt.hot()  
plt.colorbar()  
  
plt.savefig('imageplot.png')
```


Wire Plot



Wire Plot

matplotlib toolkits extend functionality for other kinds of visualization

```
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt

ax = plt.subplot(111, projection='3d')
X, Y, Z = axes3d.get_test_data(0.1)
ax.plot_wireframe(X, Y, Z)

plt.savefig('wire.png')
```

IDEs, Debugging, Version Control, etc.

Matplotlib

Web Scraping

Web Scraping Ingredients

Webscraping HTML involves...

- ▶ visual inspection - Chrome Developer Mode & Firebug
- ▶ browser sessions and interacting with HTML - mechanize
- ▶ HTML parsing/searching - BeautifulSoup

warning: javascript makes things tricky... check out *selenium* if you need to interact with javascript

Webscraping Example

Want to verify that wikipedia's plot

Anscombe's quartet - Wikipedia, the free encyclopedia - Google Chrome

en.wikipedia.org/wiki/Anscombe's_quartet

Article Talk

Anscombe's quartet

From Wikipedia, the free encyclopedia

Anscombe's quartet comprises four datasets that have nearly identical simple statistical properties, yet appear very different when graphed. Each dataset consists of eleven (x,y) points. They were constructed in 1971 by the statistician **Francis Anscombe** to demonstrate both the importance of graphing data before analyzing it and the effect of **outliers** on statistical properties.^[1]

For all four datasets

Property	Value
Mean of x in each case	9 (exact)
Variance of x in each case	11 (exact)
Mean of y in each case	7.68 (to 3 decimal places)
Variance of y in each case	4.122 or 4.127 (to 3 decimal places)
Correlation between x and y in each case	0.818 or 0.8 (to 3 decimal places)
Linear regression line in each case	$y = 3.00 + 0.500x$ in 2 and 3 decimal places, respectively

The first scatter plot (top left) appears to be a simple linear relationship, corresponding to two variables correlated and following the assumption of normality. The second graph (top right) is not distributed normally, while an obvious relationship between the two variables can be observed. It is not linear, and the Pearson correlation coefficient is not relevant. In the first graph (bottom left), the distribution is linear, but with a different regression line, which is affected by the one outlier which exerts enough influence to alter the regression line and lower the correlation coefficient from 1 to 0.818. Finally, the fourth graph (bottom right) shows another example where one outlier is enough to produce a high correlation coefficient, even though the relationship between the two variables is not linear.

The quartet is still often used to illustrate the importance of looking at a set of data graphically before starting to analyze according to a particular type of relationship, and the inadequacy of basic statistic properties for describing realistic datasets.^{[1][2][3][4]}

The datasets are as follows. The x values are the same for the first three datasets.^[1]

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	6.58	10.0	7.46	8.0	6.58
8.0	6.95	8.0	12.74	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	5.26	11.0	7.81	8.0	6.47
14.0	9.96	14.0	9.15	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.98	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	10.0	10.50
12.0	10.84	12.0	9.13	12.0	6.95	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.82	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

A procedure to generate similar data sets with identical statistics and dissimilar graphics has since been developed.^[5]

References

Anscombes_qua...svg

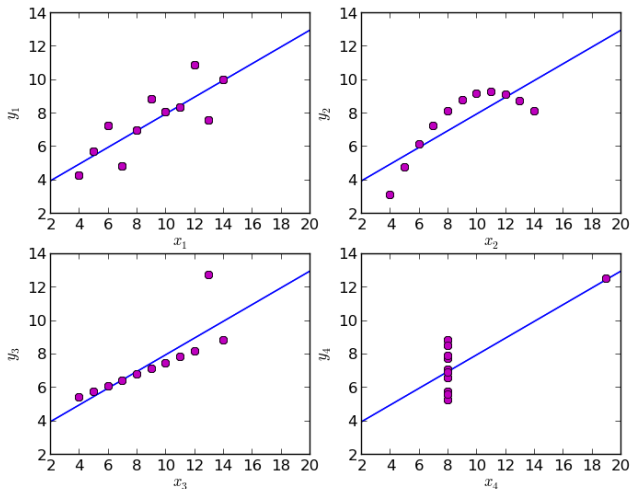
Show all downloads...

All four sets are identical when examined using simple summary statistics, but very considerably when graphed.

Webscraping Example

Recreate the plot ourselves by scraping the data and plotting

CME 193: Anscombe's quartet by Daniel Frank



Webscraping Example: Visual Inspection

In Chrome Developer Mode we can use 'inspect element' to look at the HTML associated with the table we're interested in.

The screenshot shows a Google Chrome browser window displaying the Wikipedia page for "Anscombe's quartet". The page content includes text explaining that the quartet is used to illustrate the importance of looking at data graphically and that the relationship between variables is not linear. Below the text is a table titled "Anscombe's quartet" with four columns labeled I, II, III, and IV, each containing two sub-columns for x and y values.

Anscombe's quartet							
I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
10.1	8.77	10.1	8.17	9.96	7.81	8.1	5.96
10.2	7.26	10.2	8.76	10.01	8.84	8.1	5.76
10.4	6.13	10.4	8.10	10.04	8.81	8.1	5.56
10.5	5.39	10.5	9.26	10.05	8.84	8.1	5.26
10.6	8.10	10.6	8.44	10.07	8.69	8.1	5.56
10.7	9.13	10.7	7.06	10.08	8.17	8.1	5.06
10.8	8.74	10.8	8.81	10.03	8.96	8.1	5.84
10.9	5.25	10.9	8.84	10.08	8.10	8.1	5.25
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71

The Chrome Developer Tools are open, showing the HTML structure of the table. The table is a single row with four cells, each containing a table with two columns (x and y). The HTML structure is as follows:

```
<table class="wikitable" style="text-align: center; margin-left: auto; margin-right: auto; border="1">
  <caption>Anscombe's quartet</caption>
  <tbody>
    <tr>
      <td><table><tr><th>x</th><th>y</th></tr><tr><td>10.0</td><td>8.04</td></tr><tr><td>10.1</td><td>8.77</td></tr><tr><td>10.2</td><td>7.26</td></tr><tr><td>10.4</td><td>6.13</td></tr><tr><td>10.5</td><td>5.39</td></tr><tr><td>10.6</td><td>8.10</td></tr><tr><td>10.7</td><td>9.13</td></tr><tr><td>10.8</td><td>8.74</td></tr><tr><td>10.9</td><td>5.25</td></tr><tr><td>13.0</td><td>7.58</td></tr></tbody>
    </tr>
  </tbody>
</table>
```

The right-hand pane shows the Computed Style for the table, including the border and padding properties.

Webscraping Example: Visual Inspection

```
# missing imports and bad multilines!
br = mechanize.Browser()
br.addheaders = [( 'User-agent', 'Mozilla/5.0 (Macintosh; U; ' +
    ' Intel Mac OS X 10_6; en-us) ' +
    ' AppleWebKit/531.9 (KHTML like Gecko) Version/4.0.3 Safari/531.9' )]
soup = BeautifulSoup(br.open("http://en.wikipedia.org/wiki/Anscombe's_quartet").read())

tbl = soup.find(lambda tag: (tag.name == 'caption') and \
                           (tag.text == "Anscombe's quartet")).parent
arr_list = []
for row in tbl.findAll('tr'):
    elements = row.findAll('td')
    if len(elements) != 0:
        try:
            np.float(elements[0].string)
        except:
            continue
        arr_list.append(np.array([np.float(e.string) for e in elements]))
data = np.vstack(arr_list)

grid = np.linspace(2, 20, 100)
for i in xrange(4):
    x = data[:, 2 * i]; y = data[:, 2 * i + 1]
    a, b = scipy.polyfit(x, y, 1)
    plt.subplot(2, 2, i + 1)
    plt.plot(grid, a * grid + b, 'b-', x, y, 'mo')
    plt.xlabel("$x_" + str(i + 1) + "$"); plt.ylabel("$y_" + str(i + 1) + "$")
    plt.xlim((2, 20)); plt.ylim((2, 14))

plt.suptitle("CME 193: Anscombe's quartet by Daniel Frank")
```