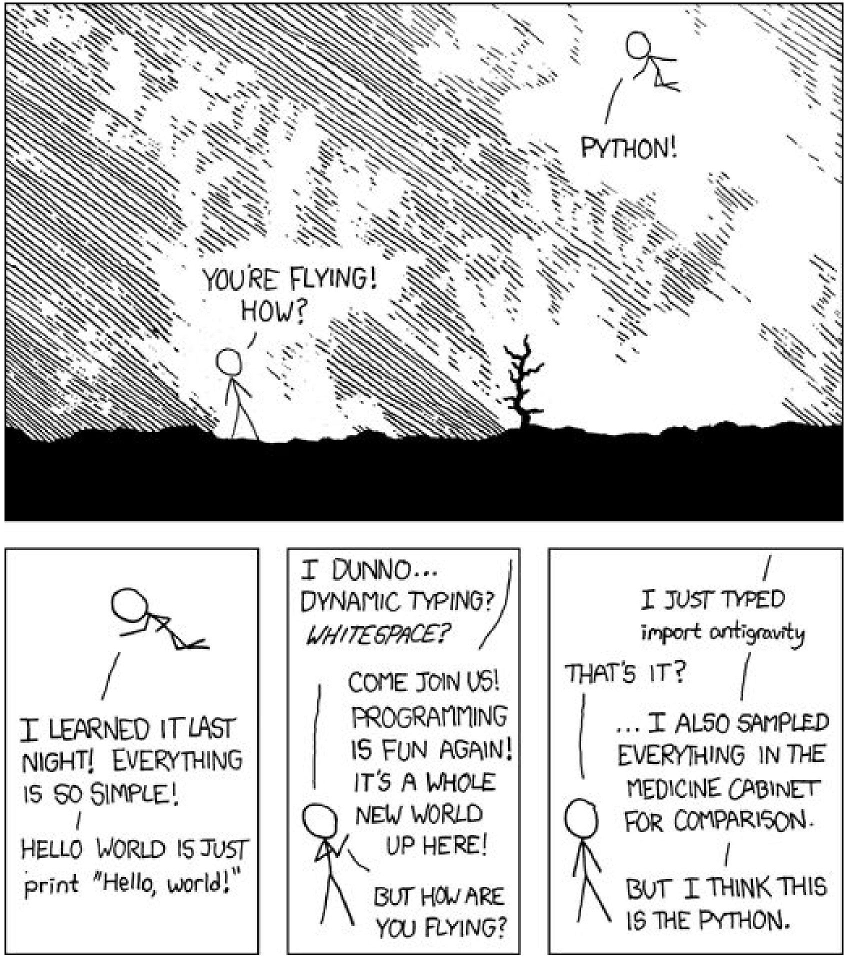
CSE 519: Data Science
Steven Skiena
Stony Brook University

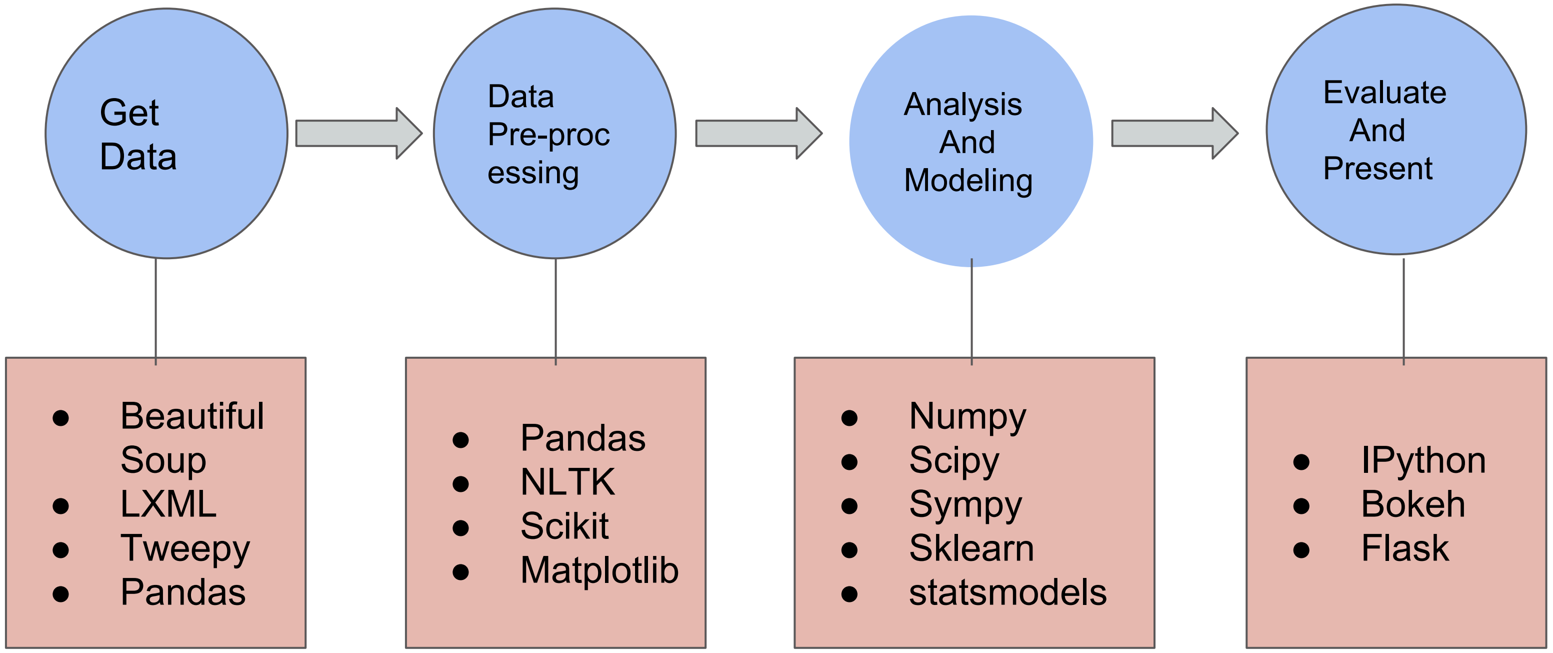
Lecture 3: Python for Data Science I


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Lecture Goals

- Overview of how to use Python for Data Science.
- Not a Python 101
  - Assume you already know Python or are willing to learn.
  - See http://www.learnpython.org/
- Learn by example
  - Demonstrate by solving actual problems.

Data Science with Python

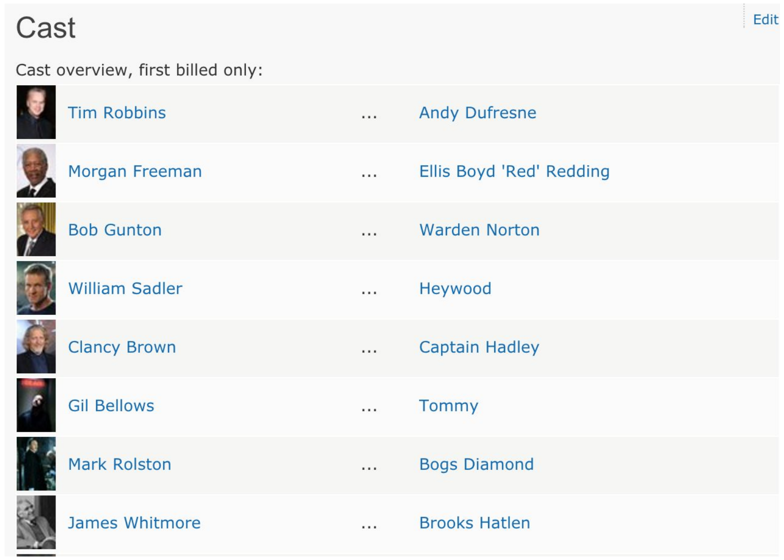


Step 1: Get Data

- Several Python packages to easily scrape and download data
  - HTML and XML: Beautiful Soup
  - Twitter: Tweepy
  - Reddit: PRAW
  - Wikipedia Processing: wikipedia
  - Stackoverflow - PyStackExchange

Example

- Scrape IMDB and get actor names and characters in Shawshank Redemption



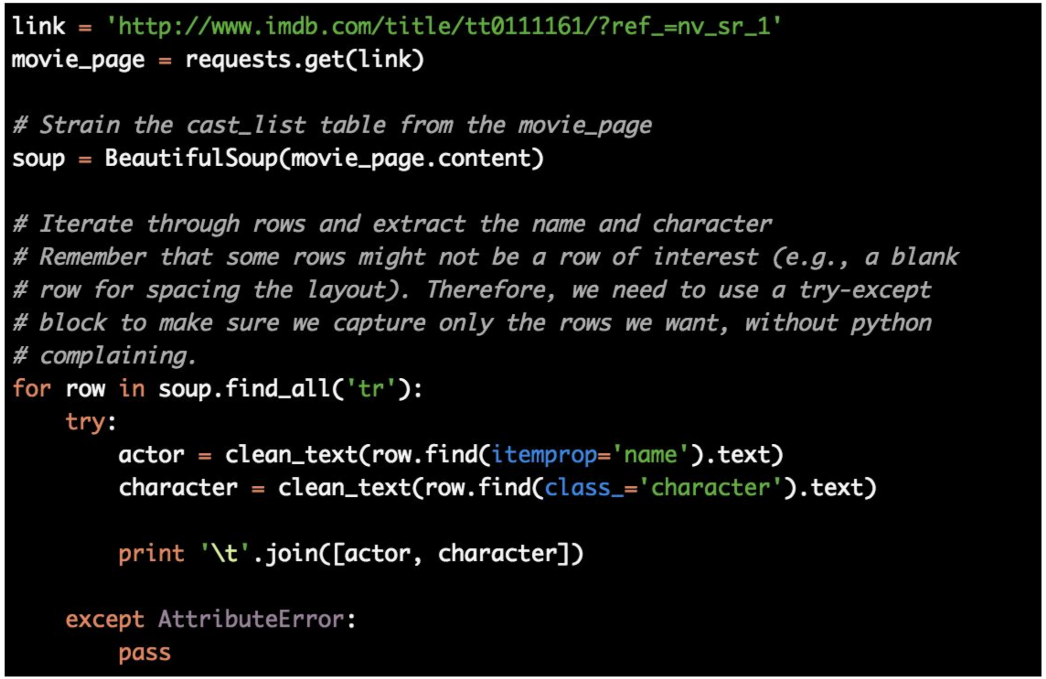
Cast

Edit

Cast overview, first billed only:

Tim Robbins	...	Andy Dufresne
Morgan Freeman	...	Ellis Boyd 'Red' Redding
Bob Gunton	...	Warden Norton
William Sadler	...	Heywood
Clancy Brown	...	Captain Hadley
Gil Bellows	...	Tommy
Mark Rolston	...	Bogs Diamond
James Whitmore	...	Brooks Hatlen

Sample code using Beautiful Soup



```
link = 'http://www.imdb.com/title/tt0111161/?ref_=nv_sr_1'
movie_page = requests.get(link)

# Strain the cast_list table from the movie_page
soup = BeautifulSoup(movie_page.content)

# Iterate through rows and extract the name and character
# Remember that some rows might not be a row of interest (e.g., a blank
# row for spacing the layout). Therefore, we need to use a try-except
# block to make sure we capture only the rows we want, without python
# complaining.
for row in soup.find_all('tr'):
    try:
        actor = clean_text(row.find(itemprop='name').text)
        character = clean_text(row.find(class_='character').text)

        print '\t'.join([actor, character])

    except AttributeError:
        pass
```

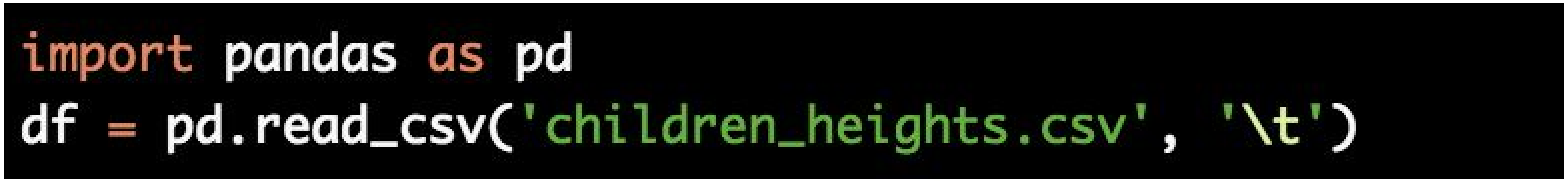
See https://raw.githubusercontent.com/5harad/datascience/master/webscraping/01-bs/get_cast_from_movie.py for full code

Using Pandas to load CSV or Tables

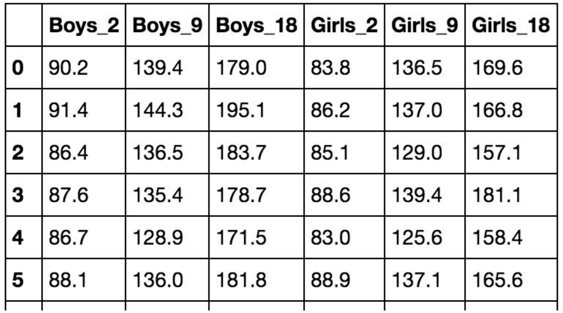
- Pandas is spreadsheet software for Python
  - A table is called a DataFrame
  - A 1-D array of numbers is called a Series
- Important Features
  - Easily load CSV, TSV files
  - Can easily load data in chunks if needed.
  - Support group-by, indexing, selection, merge operations
  - Data Analysis Functions like mean, median

Example

- Load data containing height in centimeters of boys and girls through ages 2, 9,18 years.



```
import pandas as pd
df = pd.read_csv('children_heights.csv', '\t')
```



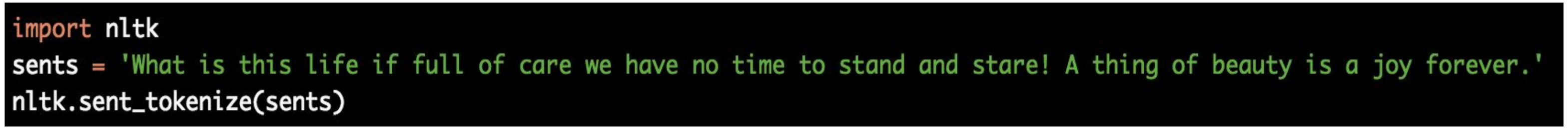
	Boys_2	Boys_9	Boys_18	Girls_2	Girls_9	Girls_18
0	90.2	139.4	179.0	83.8	136.5	169.6
1	91.4	144.3	195.1	86.2	137.0	166.8
2	86.4	136.5	183.7	85.1	129.0	157.1
3	87.6	135.4	178.7	88.6	139.4	181.1
4	86.7	128.9	171.5	83.0	125.6	158.4
5	88.1	136.0	181.8	88.9	137.1	165.6

Step 2: Preprocessing

- Raw data might need to be pre-processed
- Specialized packages might need to be used based on type of data
    - Numeric Data: numpy, pandas
    - Text Data: NLTK
    - Image Data: scikit-image
- Preprocess the data only once! Don’t waste CPU cycles doing it each time!

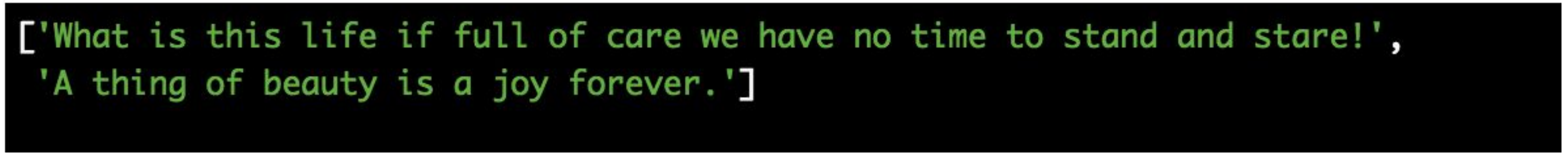
Example: Textual Data

- Split text into sentences



```
import nltk
sents = 'What is this life if full of care we have no time to stand and stare! A thing of beauty is a joy forever.'
nltk.sent_tokenize(sents)
```

- Output

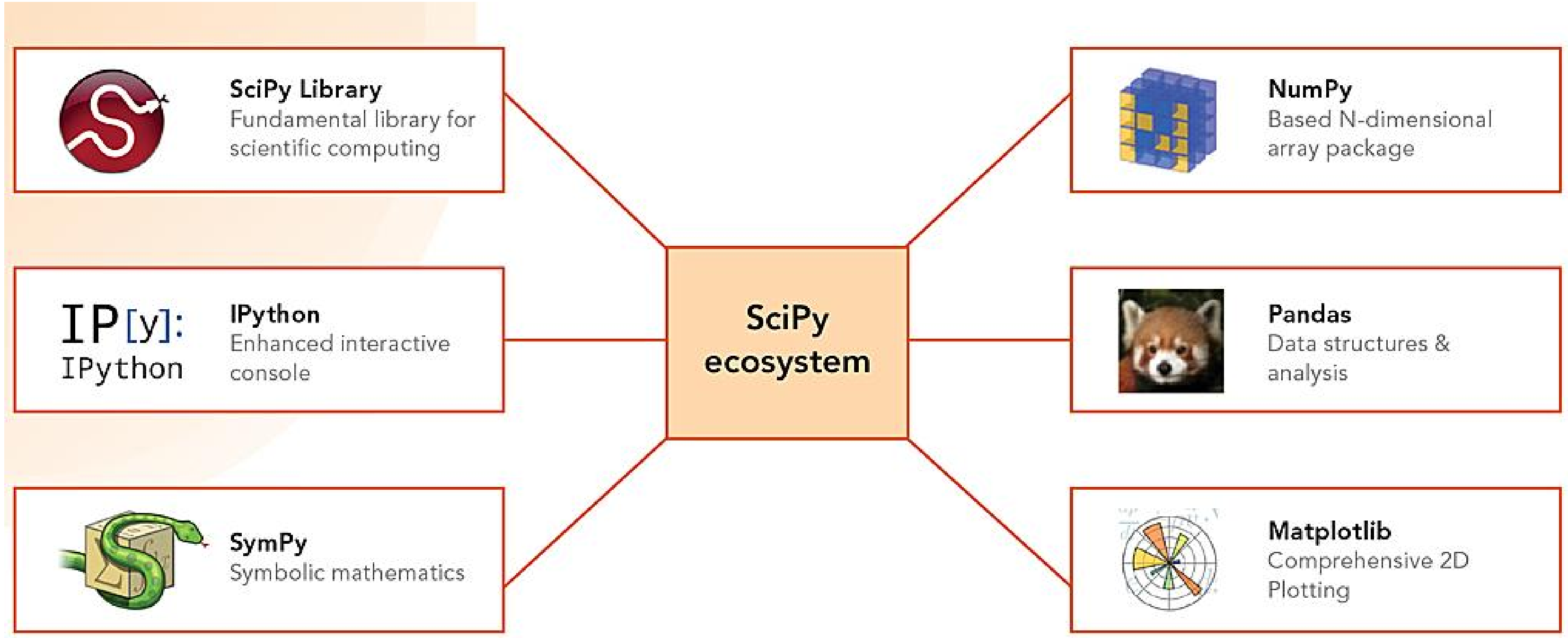


```
['What is this life if full of care we have no time to stand and stare!',
 'A thing of beauty is a joy forever.']
```

Step 3: Modeling and Analysis

- Build or infer a mathematical model for the problem
- The Scientific Python (Scipy) stack is most useful in this step
- Several Distributions (pre-packaged) available:
  - Enthought
  - Anaconda

Scientific Python Eco-system


SciPy Library
Fundamental library for scientific computing
IP[y]: IPython
IPython
Enhanced interactive console
SymPy
Symbolic mathematics
SciPy ecosystem
NumPy
Based N-dimensional array package
Pandas
Data structures & analysis
Matplotlib
Comprehensive 2D Plotting

http://www.esri.com/~/media/Images/Content/news/arcuser/0115/scipy_2-lg.jpg

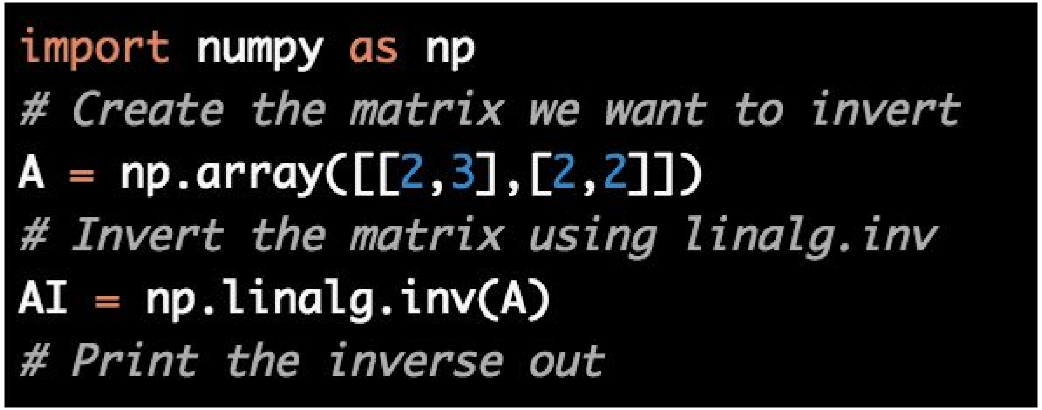
Numpy overview

- Provides a fast, efficient implementation of N-d array (ndarray)
- Several statistical operations supported:np.mean, np.std, np.median
- Supports linear algebra operations: dot product, cross product
- Fast Fourier Transforms, Signal Processing operations also supported

Using Numpy Example

- Invert the matrix $\begin{pmatrix} 2 & 3 \\ 2 & 2 \end{pmatrix}$

- Sample code using Numpy



```
import numpy as np
# Create the matrix we want to invert
A = np.array([[2,3],[2,2]])
# Invert the matrix using linalg.inv
AI = np.linalg.inv(A)
# Print the inverse out
```

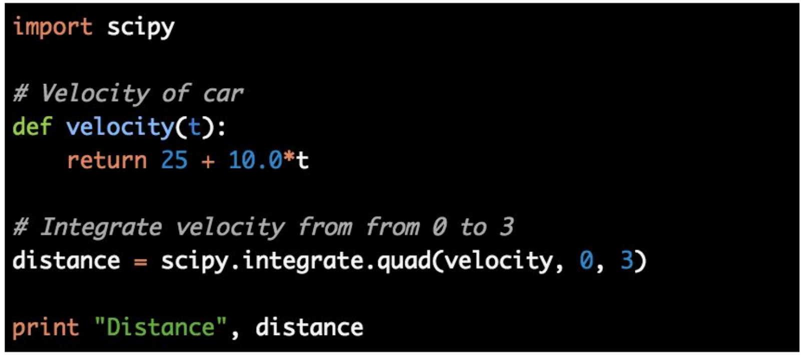
$$\Rightarrow \begin{pmatrix} -1 & \frac{3}{2} \\ 1 & -1 \end{pmatrix}$$

Scipy overview

- Package containing extensive functionality for use by scientists
  - Linear Algebra (scipy.linalg)
  - Optimization (scipy.optimize)
  - Statistics (scipy.stats)
  - Signal Processing: (scipy.signal)
  - Special functions (like Gamma): (scipy.special)

Using SciPy example

- A car’s velocity in (mph) at time t is given by: 25 + 10t. Find the distance in miles covered by the car in 3 hours.
- Solution: 120 miles



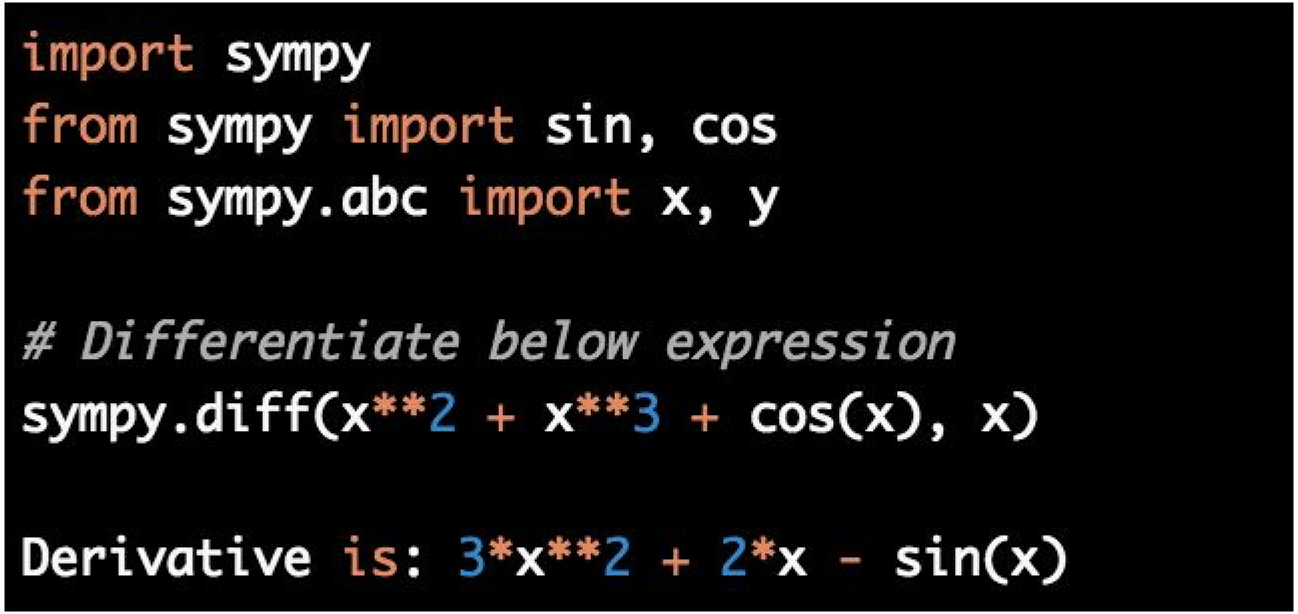
```
import scipy

# Velocity of car
def velocity(t):
    return 25 + 10.0*t

# Integrate velocity from from 0 to 3
distance = scipy.integrate.quad(velocity, 0, 3)

print "Distance", distance
```

SymPy overview

- Symbolic Manipulation in Python
- Supports differentiation, integration, simplifying equations etc
- Useful in modeling especially machine learning
- Most used for computing exact solutions

Using SymPy

- Differentiate an expression analytically



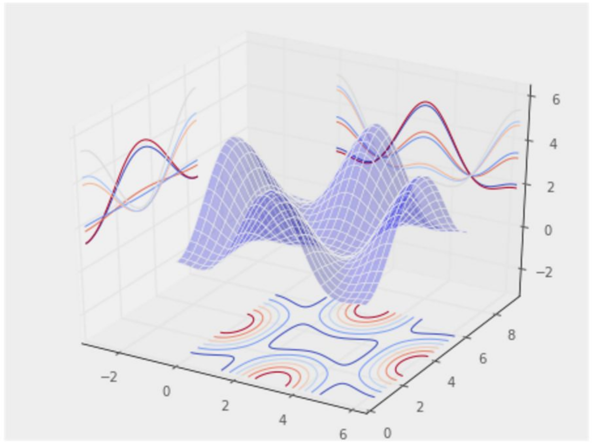
```
import sympy
from sympy import sin, cos
from sympy.abc import x, y

# Differentiate below expression
sympy.diff(x**2 + x**3 + cos(x), x)

Derivative is: 3*x**2 + 2*x - sin(x)
```

Visualization in Python: Matplotlib

- Matplotlib is basic plotting library in Python
- Can easily create figures and manipulate them
- Support for
  - Scatter plots
  - Charts
  - Bar Charts, Pie Charts
  - Box and Whisker Plots
  - Lines

Example Visualiztion

```
from mpl_toolkits.mplot3d.axes3d import Axes3D

alpha = 0.7
phi_ext = 2 * np.pi * 0.5

def flux_qubit_potential(phi_m, phi_p):
    return 2 + alpha - 2 * np.cos(phi_p)*np.cos(phi_m) - alpha * np.cos(phi_ext - 2*phi_p)

phi_m = np.linspace(0, 2*np.pi, 100)
phi_p = np.linspace(0, 2*np.pi, 100)
X,Y = np.meshgrid(phi_p, phi_m)
Z = flux_qubit_potential(X, Y).T

fig = plt.figure(figsize=(8,6))

ax = fig.add_subplot(1,1,1, projection='3d')

ax.plot_surface(X, Y, Z, rstride=4, cstride=4, alpha=0.25)
cset = ax.contour(X, Y, Z, zdir='z', offset=-np.pi, cmap=plt.cm.coolwarm)
cset = ax.contour(X, Y, Z, zdir='x', offset=-np.pi, cmap=plt.cm.coolwarm)
cset = ax.contour(X, Y, Z, zdir='y', offset=3*np.pi, cmap=plt.cm.coolwarm)

ax.set_xlim3d(-np.pi, 2*np.pi);
ax.set_ylim3d(0, 3*np.pi);
ax.set_zlim3d(-np.pi, 2*np.pi);
```



Problem

- Are boys taller than girls on an average?
  - Get data
  - Form hypothesis
  - Analyze data
  - Interpret results