Galvanize: Intro to Python for Data Science

* Metis, Galvanize, General Assembly

6:00 pm - Networking & Announcements

* Anthony, Evangelist, Galvanize
* Galvanize: Dynamic learning community for technology (events, educational programs, workspace)
* Data Science Immersive (3 months long, full-time)
* Master of Science in Data Science (partnered with University of Connecticut)
* Web Development Immersive (6 months long)
* Python Part-Time Evening Classes

6:30 pm - Why Use Python for Data Science?

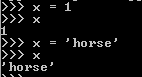
6:50 pm - Working with Python’s NumPy, SciPy

7:50 pm - Working with Pandas (DataFrames)

8:30 pm - Statistical Analysis in Python

9:00 pm - Wrap-Up and Additional Resources

Introduction to Python for Data Analysis, Jared Thompson

* Agenda
  + Why Python
  + Jupyter Notebook
  + Getting Started with Python
  + Common Data Science/Analysis Libs
  + Tricks
  + Where to Go Next
* What to Expect?
  + Quick overview (Python / DS)
  + Introduction
  + Advanced stuff
  + Get outside of your comfort zone
  + Be patient
  + Practice at home
* Why Python?
  + Scalability
  + Extensive [Data] Analytics Lis &Community
    - SciPy.org (Mathematics, Science, Engineering)
    - StatsModels (Statistics)
    - Pandas (Frameworks)
    - SciKit-Learn (ML)
  + Graphics
    - Libraries (ggplot, matplotlib)
    - APIs – Plot.ly
  + Easy to learn and code
* What is Python?
  + An Open source, high-level, dynamic scripting language
    - OS: Free! (both binaries and source files)
    - High-level: interpreted (add(a, b)); not compiled
    - Dynamic: Things that would happen at compile time happen at runtime instead
  + Dynamic Typing
    - 
* Strengths and Weaknesses
  + Strengths
    - Python's popularity comes for the strength of design
    - BDFL (benevolent dictator for life): Guido Van Rossum
    - Has unified design philosophy
    - Emphasizes reusability and readability of code
  + Weaknesses
    - Slower than lower-level languages
* Python, how?
  + Anaconda Project (<http://continuum.io/anaconda>)
  + iPython Notebook (<http://ipython.org>)
* Other Tools
  + Julia
  + R
  + Matlab
  + Strata (SPSS, SAS)
* Books
  + Python
  + Think Python
  + Python for Data Analysis
* Python in the Data Science Workflow
* What is Data Science/Data Exploration?
  + Extract useful information and knowledge from data
    - Clarify info from large data
    - Imrpove decision-make
    - Tell compelling stories
* Python as a tool for Data Science
  + Diagram: Doing Data Science // O’Reilly 2014
* Data Preparation
  + The goal of “Pre-Processing” is to convert data into a standard format
  + Standard format allows for input to alogirthms
* Analysis/Model
  + Different algorithms for optimal choice
    - SciPy
    - Scikit-learn
    - StasModels
  + Built as “black boxes”
* Vizualization
  + Since seeing is believing…
  + Prettyplotlib
* Data Science Library Introduction
  + Numpy: Vectors, arrays, matrices, tensors (matrices of matrices)
  + Matplotlib
  + Scipy
  + Pandas
  + Scikit
  + StatsModels
* Basic Data Structures
  + Array, Lists => NumPy Array
  + Tuples => immutable arrays of arbitrary elements
  + String
  + Associative Arrays (hash tables)
  + Sets
  + Control Flow
* NumPy
  + Matrix Multiplication
  + Matrix object
  + Ndarray element-wise operations
* Pandas
  + “OS library providing high-performance data structures and data analysis tools”
    - For bigger data, not gigantic (that’s for Spark)
    - Examples
      * All the marketing data for the last 2 years
      * A year of chemical engineering data
  + Series – 1D labeled array
    - Series indexed by a series of labels
    - Can be addressed or sliced like an array
    - Somewhat akin to dictionaries and easy to convert a dict to a series
    - “Trick”/Help(!)
    - Provides vector operation support and index alignment
  + DataFrame => 2D table
  + Some notable features
    - Data loading
    - Data selection using indexes
    - Group
    - Indexing
    - Conditional Indexing
    - Column Renaming
    - Remove Missing Values => .dropna()
    - Replace Missing Values => .fillna()
    - Map & Apply Functions
      * Map => map(lambda x : ‘map\_’ + x)
      * Apply => apple(np.sqrt)
    - Applymap => .applymap()
    - Vectorized String Operations
    - Groupby
    - Statistics
      * Covariance Tables
      * Correlation Tables
    - Merge and Join
      * Merge => total join
      * Join => left join
      * Mount one table to another so that you have 2 tables linked together based on their labels
    - Basic Plotting: Lines => df.plot()
    - Histograms => df.hist()
* Resources & Next Steps
  + After Jupyter => Pycharm
  + <http://cli.learncodethehardway.org>
  + <http://learnpythonthehardway.org>
  + <http://khanacademy.org/math/probability/regression>
  + Anaconda
  + <http://wakari.io> => Online IDE
  + Google for Education => <https://developers.google.com/edu/python/set-up>
* Question
  + Want to find out how much you weigh in stone. A concise program can make short work of this task. Since stone is 14 pounds, and there are 2.2 pounds in a kilogram, the formula:
    - m­­stone = (mkg x 2.2) / 14
* Question
  + Find all numbers divisible by 7 but not a multiple of 5, between 2000 and 3200