Collecting Data

There are several ways to collect data:
• Using existing datasets
• Create/Simulate your own dataset
• Using Web scraping
• Using API
Web Scraping

We can collect data using web scraping using one of the following methods:
• Using simple tools like wget
• Using Selenium for dynamic loaded pages
• Using web scraping frameworks like Scrapy
• Writing your own code
Using Application Programming Interfaces

We can use various websites’ Application Programming Interfaces (APIs) to collect data from various platforms, such as:

- Twitter
- Reddit
- Google Maps
- Kaggle
- Github
Recommended Read

- Python Data Science Handbook, Chapter 1 IPython: Beyond Normal Python by Jake VanderPlas
- The Unix Shell by Software Carpentry Foundation
- Practical Introduction to Web Scraping in Python by Colin OKeefe
MANIPULATING DATA
NUMERICAL PYTHON (NUMPY)
Source: Python Data Science Handbook, Chapter 1 IPython: Beyond Normal Python by Jake VanderPlas
NumPy - The Basics

- Supports large multi-dimensional arrays and matrices
- Contains large collection of high-level mathematical functions to operate on these arrays
- Tools for reading / writing array data to disk

Useful Reading:
- Chapter 4. NumPy Basics: Arrays and Vectorized Computation, Python for Data Analysis, by Wes McKinney
- Chapter 2. Introduction to Numpy, Python Data Science Handbook, by Jake VanderPlas
WORKING WITH PANDAS & DATAFRAMES

\[ y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it} \]
In Data Science, 80% of time spent prepare data, 20% of time spent complain about need for prepare data.
Pandas

Pros:
• Provides flexible and expressive data structures
• Easy to handle missing data
• Columns can easily be added and deleted

Cons:
• Good for several gigabytes of data
• Mostly single threaded
• Complex Group By operations
“My rule of thumb for pandas is that you should have 5 to 10 times as much RAM as the size of your dataset”

Wes McKinney, 2017
Pandas Objects

- DataFrame
  - Column
  - Series
    - Values
  - NumPy
Let’s move to the Notebook