Class 11 - NumPy

[W200] MIDS Python Summer 2018

Agenda

Course Schedule

Project 2 Discussion
Last Project 1 Presentations
The PyData Ecosystem
Functional Programming
NumPy
NumPy Breakout

Schedule

Class 10 - Project 1 Showcase

Class 11 - NumPy

Class 12 - Data Analysis with Pandas

Class 13 - More Data Analysis with Pandas

Class 14 - Code Testing and Final Project Showcase

Course Schedule:

https://docs.google.com/spreadsheets/d/11DxadnNwyFaJIPYLUJSPUlNGCtTenBCR4yaR1CbFBKg

Schedule | Where we're going - projects/exams

Live Session 11 - Discuss Final Project
Live Session 12 - Proposal Finalized
Live Session 13 - Final Exam Distributed
Live Session 14 - Final Exam Due Projects Due, Final Project Showcase

Course Schedule:
https://docs.google.com/spreadsheets/d/11DxadnNwyFaJIPYLUJSPUlNGCtTenBCR4yaR1CbFBKg

Grading | Reminder of Breakdown

1. Homework (30%)
2. Midterm (10%)
3. Project 1 (20%)
4. Final (10%)
5. Project 2 (20%)
6. Participation (10%)

Schedule | second to last HW, 3 Asyncs left

1) You have a homework and async assignment due next class
2) Your proposal for Project 2 is also due next class.

We are grading your Midterm and Project 1

Assignment Review | Week 11

How did the homework go this week?

What was the hardest? What was easiest?

Agenda

Course Schedule

Project 2 Discussion

Last Project 1 Presentations

The PyData Ecosystem

Functional Programming

NumPy

NumPy Breakout

Project 2 | Proposal

With your group (2-3 people) come up with a 1 - 2 page proposal about the questions that you intend to ask of the data. This should include:

- Initial plots, figures or tables.
- References to column names and the analysis that they may provide.
- Additional datasets that you plan on including in your analysis like the weather data. This means links, columns that you'll join on, etc.
- What you plan to cover in the final report and how you plan on organizing it.

Project 2 | The Report

The report will be 8+ pages (including appropriately sized figures) and will be a report on what you found out from the data.

This should focus on telling stories and explaining the narrative of the exploration and challenges associated with that.

The report should not include any code - all code should be included in a subfolder in either plain python files or in jupyter notebooks.

Project 2 | Style Guidelines 1

Your analysis is a written argument.

- Good writing style is key.
- The key is to aim for clarity and exposition.
- Organize your argument clearly
- Guide the reader through the evidence in the data.
- Proofread

If you don’t have something nice to say (about your plot), don’t display it at all.

- No output dumps
- Every graph should be mentioned in your writing, explain what it *means*.

Project 2 | Style Guidelines 2

Document decisions

- If you decide that observations should be removed, state which ones.
- If values are suspicious, but you leave them in, state that too.
- If you transform a variable, for example, by taking the logarithm, state that.
- Your justification can often be very brief (just a sentence), but make sure the reader can follow your logic.

Characterize relationships between variables

- Keep in mind the purpose of the analysis.
- If you're interested in explaining the price of a house, look to see what kind of relationship that variable has with the other variables.

Project 2 | Style Guidelines 3

Use only descriptive statistics

- Descriptive statistics summarize a particular sample of data.
- In the field of inferential statistics (which you'll learn in w203) you'll see how to create a model that represents the population (or process) from which the sample came from, and to make assertions about that population.
- Since we haven't taught you any inference, please don't use it.
- Beware of the word 'significant'
    - This has a technical meaning, implying that you've performed a statistical test.

Project 2 | Groups

Discuss Project Groups

One option is to plan based on your availability in this tracker:

https://docs.google.com/spreadsheets/d/1MjMaUMcLagaP4XAfAE4qpnzB7OSLjGK8uULpJF2Lhsw

- Please add your project repo location

Project 2 | Forming Hypotheses

Which comes first? The question(s), or the data?

Project 2 | Forming Hypotheses

Which comes first? The question(s), or the data?

1. Data First - e.g., if your company collects data in course of business
2. Question First - e.g., if you have a strategic or research question to answer.

For this project, we suggest an iterative approach.

Project 2 | Forming Hypotheses

For this project, we suggest an iterative approach.

1. Look through potential data sources
2. Discuss questions that could be interesting
3. Confirm that the data source can answer the question
    a. If “yes”, develop your proposal
    b. If “no”, ask if there are similar questions you can answer with your data. OR find a new dataset.

Note: You may decide to combine multiple data sets for your project. For example, merging weather data onto traffic data to get additional insights. You can brainstorm this kind of combination as you look at data.

Project Hints | GitHub Review

Note: **files over 100MB** cannot be stored on GitHub. Use DropBox or alternative methods to share.

https://www.sendthisfile.com/

Note: merging notebooks is **not practical**. They get REALLY mangled.

*Keep local copies
*Communicate about who is working on the book
*Pull before you push
*Or use a collaborative notebook (jupyterlab with the google drive extension)

Project 2 | Grading

1-2 page proposal: 10%

8 page paper: 70%

Final class presentation: 20%

https://docs.google.com/spreadsheets/d/1MjMaUMcLagaP4XAfAE4qpnzB7OSLjGK8uULpJF2Lhsw

Project 2 | Grading

Paper Grading (70%):

- 10% Questions
- 20% Data Cleaning / Sanity Checks
- 20% Compelling Text and Data Stories
- 20% Compelling Figures

Agenda

Course Schedule

Project 2 discussion

Last Project 1 Presentations

The PyData Ecosystem

Functional Programming

NumPy

NumPy Breakout

Agenda

Course Schedule

Project 2 discussion

Last Project 1 Presentations

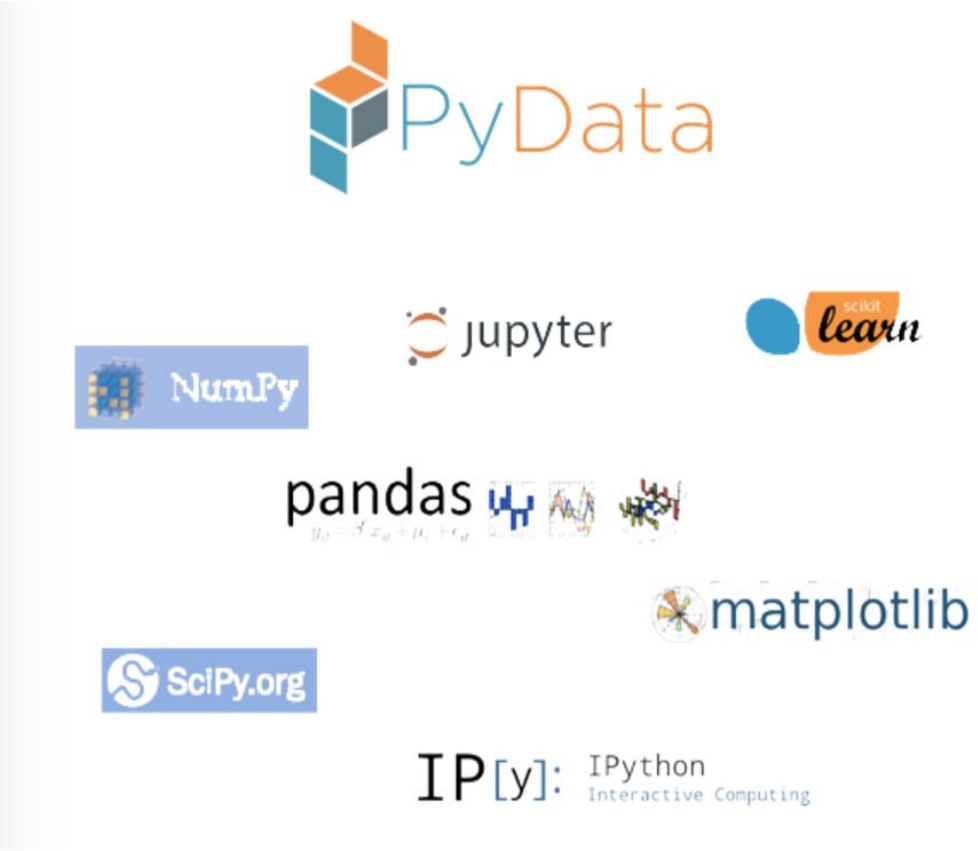
The PyData Ecosystem

Functional Programming

NumPy

NumPy Breakout

PyData Ecosystem | Packages



PyData Ecosystem | Packages

This Course:

- **Jupyter -** notebooks
- **Ipython -** interactive programming
- **NumPy** - n-dimensional arrays
- **Pandas** - more user friendly analytics
- **MatPlotLib** - plotting and graphics library

Not Covered:

- **SciPy** - Scientific computing framework on top of NumPy
- **Scikit Learn** - Machine Learning
- **Statsmodels** - Models from classical statistics
- **Seaborn, ggplot** - More powerful visualization

Agenda

Course Schedule

Project 2 discussion

Last Project 1 Presentations

The PyData Ecosystem

Functional Programming

NumPy

NumPy Breakout

Pause

A brief reflection on Object Oriented Programming...

1. If you were going to create “Pokemon Go”, what would your objects be?
2. What objects would be subclasses of others?

Functional Programming | Two Paradigms

Object Oriented Programming

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which are data structures that contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods.

Functional Programming

Functional programming is a programming paradigm, a style of building the structure and elements of computer programs, that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data

Functional Programming | Two Paradigms

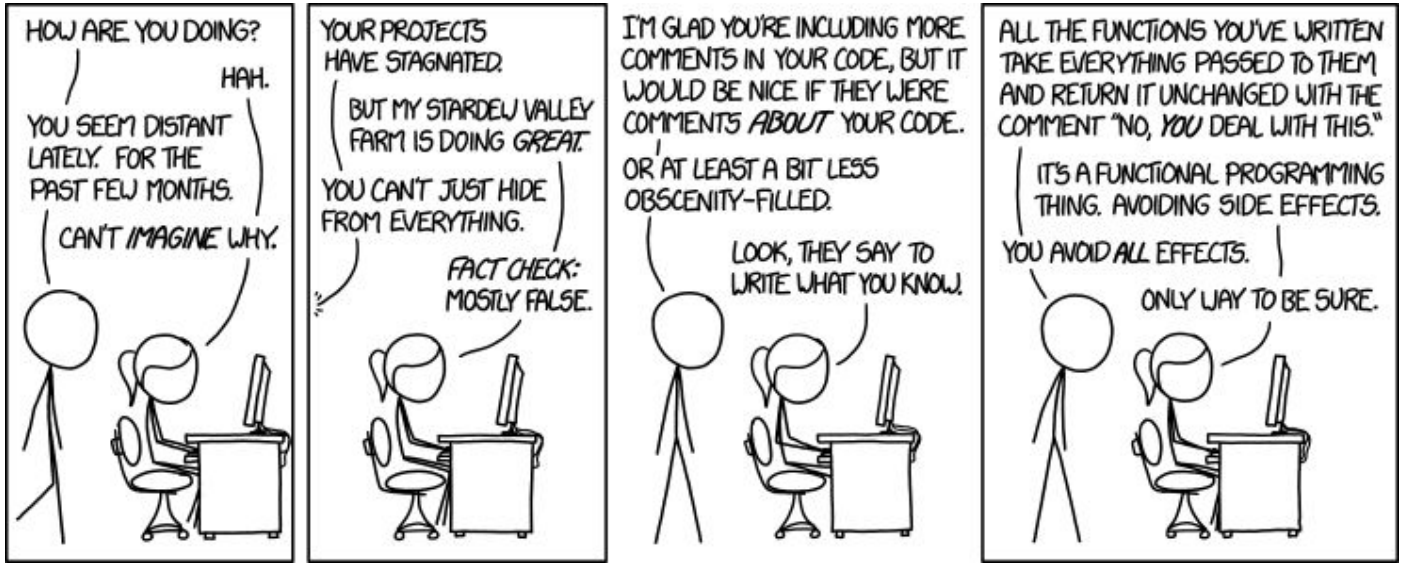
All programs have “data” and “behaviors”

- **Data** - the “stuff” a program knows
- **Behavior** - the “stuff” a program can do

In object-oriented programming, we store our data and the behavior together within a class. E.g., attributes (data) and methods (behavior)

In functional programming, we keep the data and the behavior separate. E.g., a dataset (data) and functions that operate on the data (behavior)

Functional Programming | Relevant xkcd



https://xkcd.com/1790/

Functional Programming | Python Functions

We have learned several functions useful in a functional programming approach:

1. List comprehensions
2. map()
3. filter()
4. “lambda”

Agenda

Course Schedule

Project 2 discussion

Last Project 1 Presentations

The PyData Ecosystem

Functional Programming

NumPy

NumPy Breakout

NumPy | The Basics

NumPy gives you the ability to work with **n-dimensional** arrays of numeric data of many types.

Pandas is built on top of NumPy and provides a more user friendly experience. There, we work with a “dataset” and include non-numeric variables.

Understanding NumPy is critical to understanding more advanced packages.

A basic understanding of NumPy will deepen your understanding of Pandas.

NumPy offers vectorized operations

** But see this: https://timothyhelton.github.io/pandas_best_practices.html

NumPy | Python Functions

1. np.array()
2. np.arange(), np.linspace()
3. np.min(), np.max(), np.std(), np.var()
4. np.argmax(), np.argmin()
5. np.shape(), np.reshape()
6. np.zeros()
7. np.random.seed(), np.random.random_integers()
8. np.vstack(), np.hstack()

Dealing with n-dimensions: “Axis = “

Agenda

Course Schedule
Project 2 discussion
Last Project 1 Presentations
The PyData Ecosystem
Functional Programming
NumPy
NumPy Breakout