(Intro to) Data Analysis in Python

John Serences, jserences@ucsd.edu

March 13th, 2019

Class 09

Reminder:

- All in-class code and lecture slides can be found on GitHub
  - https://github.com/JohnSerences/PSYC193_IntroPython_W2019

Course Schedule (approximate)

- Week00, January 9: What is Python?, Jupyter Environment (Google Colab), First Program, Intro to object types and methods
- Week01, January 16: More on object types, lists, for loops, list comprehensions, slicing lists
- Week02, January 23: If…elif…else statements, dictionaries
- Week03, January 30: User input, while statements, try/except statements
- Week04, February 6: NO CLASS
- Week05, February 13: Midterm, writing functions
- Week06, February 20: Classes, object-oriented programming
- Week07, February 27: File Input/Output, data formats for files (e.g. JSON)
- Week08, March 6: NumPy (numerical computing), Plotting (Matplotlib)
- **Week09, March 13: Pandas (data frames)**
- Final: Room/Time TBD

In class quiz on material from last week

- Questions from last week…

NumPy data arrays

```
[[ 0.69755819 -0.2270603   0.42633358 -0.52156878 -0.13138913]
 [-1.52254985  0.82578957  0.03824409  0.70180638  0.4895618 ]
 [-1.72148639  0.31922138 -0.50279569 -0.10679396 -0.0392484 ]
 [-0.80145525 -0.70854211  2.09932393  0.01783151 -2.09342844]
 [-0.13002857 -1.40458643 -0.1266029  -0.21018433 -0.40934157]]
```

Pandas

	stim1	stim2
Nrn0	36	59
Nrn1	17	60
Nrn2	6	11
Nrn3	8	76
Nrn4	9	86

Pandas – Series objects

- https://pandas.pydata.org/pandas-docs/stable/dsintro.html#series

- A **Series** is a 1D array that can hold any type of data (numeric types, non-numeric, Python objects and so forth).
  - Unlike a 1D numpy array, each entry is **labeled** with an index that is used to keep track of what each entry is, and can be used to lookup the value corresponding to each index during analysis (remember dictionaries?)
  - These labels are fixed - they will always index the same value unless you explicitly break that link.
  - The list of labels that forms the index can either be declared upon series creation or, by default, it will range from 0 to len(data)-1.
  - If you're going to use Pandas to organize your data, **specifying usable and informative labels** is a good idea because that's one of the main advantages of organizing your data in this manner - if you just want to fly blind then NumPy is usually fine on its own

Pandas – Series objects

- After creating a pandas series, you can do many common operations and access the functionality of other modules
- list of attributes and methods: https://pandas.pydata.org/pandas-docs/version/0.23.4/generated/pandas.Series.html
- A pd Series behaves similar to a NumPy ndarray, and can be passed to many NumPy functions
- Slicing also works like a ndarray - note that the index is also sliced
- Lots of built in methods as well that emulate NumPy functionality

Pandas – Series objects

- Although series can be treated much like NumPy arrays, there is one key difference (and often a big advantage)
- When you do an operation on a NumPy array, the operation is performed in an element-by-element manner
- However, when you do an operation on two pandas series, the operation will be applied to like-labeled values
- This can save a lot of trouble in terms of lining up corresponding entries in two data arrays when the data sets are initialized in different orders!

Pandas – Data frame objects

- https://pandas.pydata.org/pandas-docs/stable/getting_started/dsintro.html
- A DataFrame (DF) is a labeled data struture that can be thought of as a 2D extension of the Series object
- A DF can accept many types of input, from a 2D ndarray, multiple Series, a dict of 1D arrays, another DF, etc
- Like a Series, DFs contain data values and their labels. Because we're now dealing with a 2D structure, we call the **row labels the index argument** and the **column labels the column argument**.
  - Like a Series, if you don't explicitly assign row and column labels, then they will be auto-generated (but not as useful as specifying the labels yourself!)

Anaconda

- https://docs.anaconda.com/anaconda/install/

In class exercises

- Please finalize by Friday at midnight so that I can grade in time to give feedback.

- We'll reconvene today at 11:30 (or so) to go over answers.

- Does anyone want to go over the exam more? If so, please let me know.