Nicholas Eubank
Assistant Research Professor

MIDS NUMBER
Fall & 2019

Practical Data Science:
Wrestling with Data & Answering Questions

1 Course Description

Data Science is an intrinsically applied field, and yet all too often students are taught the advanced math and statistics behind data science tools, but are left to fend for themselves when it comes to learning the tools we use to do data science on a day-to-day basis or how to manage actual projects.

This course is designed to fill that gap by (a) providing students the opportunity to learn about and practice working with a variety of tools like the command line, git, github, python, anaconda, and jupyter lab, and (b) helping students develop practical skills for developing data science projects and managing data science workflows. By providing this training, our goal is to ensure all MIDS students (and non-MIDS students interested in data science) have a shared comfort with data science tools, data in all its forms, and project development.

This course will be divided into two parts:

- **Part 1: Data**
- **Part 2: Science**: This course adopts the view that Data Science is the study of how best to *answer questions* about the world *using quantitative data.*

This course is not be for everyone. If you were a computer science major as an undergraduate or worked in a job that made intense use of Python for Data Science applications, please speak to me after class. With that said, even students who have taken computer science courses may find that this class offers a very different perspective on familiar tools. CS programs tend to be oriented towards a style of programming best suited for software development which can differ substantially from the tools and style used in data science.

1.1 Pre-Requisites

This course is intended intended for incoming Masters in Data Science (MIDS) students. As such, the **only** pre-requisites are two things taught in the MIDS student boot-camp:

- a familiarity with basic python
- A familiarity with git and github

MIDS students: The only knowledge assumed is the portion of these topics covered in boot-camp, so unless you skipped (the mandatory) bootcamp, you should be good.

For non-MIDS students: By "basic python" I mean a familiarity with the core Python programming language, including concepts like variables, loops, lists, and dictionaries. Unfortunately, if you haven't worked with Python before, I'm afraid this course may not be for you. (This course will *not* assume familiarity with the Python data science stack (e.g. `numpy`, `pandas`), although incoming MIDS students will have had some basic exposure to these packages, and so some lessons may be more challenging for those seeing this material for the first time.)

Git and github are a lot easier to learn than Python, though, so if you know Python but not git and github, talk to me and we can figure something out.

But again, basic familiarity with Python and git are the **only** pre-requisite. The goal of this course is to ensure all students are comfortable with the tools being taught, regardless of educational or professional background. As such, we will assume *zero* knowledge beyond basic Python and git (and that only because we know it is something to which all MIDS students have already been exposed).

2 Learning Goals

In Data Science today, the only constant is change. With that in mind, in this course we will not only learn *how* popular tools work, but also:

- the logic that underlies their operation (so when new situations arise you will have a *generalized* understanding of the tool you can use to reason through your problem), and
- how to find help on your own.

In particular, by the end of this course, you will have developed the following abilities in each topic area:

The Command Line
Main Takeaway: The Command Line is just a way to interact with your operating system with text instead of with a mouse.

- Explain the value of the command line
- Manipulate files and work with command-line-only tools
- Anticipate the likely syntax of new tools you may come across

`numpy`
Main Takeaway: numpy is what makes Python useable for data science.

- Explain *why* numpy and pandas are so crucial to data science in Python
- Manipulate vectors and matrices with `numpy`

`pandas`
Main Takeaway: pandas is a hack, so if it drives you nuts, it's not your fault.

- Read in data of various formats with `pandas`
- Clean, organize, and reshape real-world data with `pandas`
- Move back and forth from `pandas` to `numpy`
- Pass data from `pandas` and `numpy` to `scikit-learn` functions

Git and Github
Main Takeaway: Bundling changes into discrete chunks is incredibly powerful

- Not sure yet...

Getting Help Online
Main Takeaway: Asking for help effectively takes effort

- Find appropriate forums for different types of questions
- Compose requests for help that are likely to get useful responses using Minimal Working Examples (MWEs) and proofs of effort.

Debugging
*Main Takeaway: Debugging as an **active** exercise*

- Isolate and analyze bugs quickly

Workflow Management
Main Takeaway: Projects change, so a good workflow must be adaptive

- Organize data science projects in a manner that is robust to future changes
- Organize, document, and comment projects to allow others (and your future self) to easily understand project organization

Defensive Programming
Main Takeaway: To err is human, so we must develop practices to protect ourselves from ourselves

- Understand the futility of "just trying to be careful"
- Compose code that is less likely to contains errors, and where errors that do occur are more likely to be caught.

3 Python

In this class we will primarily be working with Python.

Why Python? Because it's currently one of the two most-used programs in data science (the other being R, which you'll be working with in other classes), which means there is a good chance you'll be called upon to use it when working in teams.

It is worth emphasizing that we're not learning Python because it is necessarily the "the best" language. The reality is that there are *lots* of tools for statistical programming, and each has its own strengths and weaknesses (e.g. R, Stata, SPSS, Python, Julia, Matlab, etc., etc.). People often develop strong opinions about which language is *best*, and sometimes pass judgement on people who use other languages. Every programming language has its strengths and weaknesses, and what is "best" depends on your use-case (the types of things you are using the language to do). This is true not only because languages themselves have strengths and weaknesses, but also because the tools and packages that have been created for use in different languages differ (e.g. people just haven't made a good package for doing geo-spatial work in Julia yet, for example). And if you're working on teams, you'll also have to make decisions based on the backgrounds of

your tool sets. All of which is to say: there is no single *best* language for all purposes. But Python is a very popular, strong, general purpose language, so will serve as a great starting point.

As a result, over the course of your career you may find yourself gravitating to one tool or another as required by your research. But in providing you with a firm foundation in a very popular language like Python, you will not only be learning a tool that will allow you to do most everything you'll want to do in graduate school, but you will also be providing yourself with a solid foundation in *generalizable* skills that you will find useful if you later change platforms.

4 Class Organization

Because data science is an applied discipline, this will be an intensely applied class with *lots* of hands-on exercises.

Because of the importance of practice (and working through problems as they come up), in this class we will be "flipping the classroom": most weeks you will be expected to read instructional materials before class, and in class we will do hands-on programming exercises in an environment where help will be available.

5 Course Schedule

PART I. R

Week 1: Intro to R
- Class 1: Getting to Know R
  - Read Before Class: Welcome to R!
- Class 2: Vectors
  - Read Before Class: Vectors

Week 2: Intro to R (Continued); Cleaning and Manipulation
- Class 1: Datasets
  - Read Before Class: dataframes
- Class 2: Manipulating Data
  - Read Before Class: Manipulating Data

Week 3: Merging Data; Plotting
- Class 1: Merging Data
  - Read Before Class: Merging Data
- Class 2: Plotting
  - Read Before Class: Plotting Data

Week 4: Loops and Functions
- Class 1: For-Loops
  - Read Before Class: Loops
- Class 2: Functions

- Read Before Class: Functions

Week 5: Collapsing and Reshaping; Functions

Class 1: Collapsing and Reshaping
- Read Before Class: Collapsing
- Read Before Class: Reshaping

Class 2: Functions
- Read Before Class: Functions

Week 6: R Wrap-Up

Class 1: Lists
- Read Before Class: Lists

Class 2: To Be Decided...

PART II. The Tools of Data Science No One Taught You

Week 8: The Terminal / Command Line

Class 1: Command Line Basics
- Read: What is the terminal?
- Do: DataCamp Intro to Shell for Data Science, Parts 1 (Modifying Files and Directories) and 2 (Manipulating Data)

Class 2: R & Anaconda
- Find intro to anaconda or something??

Week 9: Git and Github

Class 1: Git Basics
- Read: Git and Github
- Do: Git-It Tutorial
- Maybe: Software carpentry?

Class 2: Collaborating on Github
- Find materials!

Week 10: Getting help online; Jupyter Labs

Week 11: Debugging and Troubleshooting

Week 12: Workflow management

PART III. Programming, a CS Perspective

- Week 13: Defensive Programming, Decomposition
  - Read: Defensive Programming
  - Do: Exercises??
- Week 14: More on Data Types (Floats, Ints, Strings, etc.)
  - Read: Data Types
  - Do: Exercises??

PART IV. Other Languages

- Week 15: Trade-offs of Different languages; Python
- Week 16: Python Libraries for Data Science (numpy, pandas)