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.

This course will be divided into two parts:

- **Part 1: Data Wrangling**: In Part 1 of this course, students will develop hands-on experience manipulating real world data using a range of data science tools (including the command line, python, jupyter, git, and github).
- **Part 2: Answering Questions**: This course adopts the view that Data Science is the study of how best to *answer questions* about the world *using quantitative data*. In Part 2 of this course, students will learn to develop data science projects to answer meaningful questions via backwards design, and to manage projects from inception to presentation of results.

The first portion of the course will provide students with extensive hands-on experience manipulating real (often messy, error ridden, and poorly documented) data using the a range of bread-and-butter data science tools (like the command line, git, python (especially numpy and pandas), jupyter notebooks, and more). The goal of these exercises is to ensure students are comfortable working with data in most any form. In addition to being of intrinsic value, developing these skills will also ensure that in advanced statistics or machine learning courses, students can focus on understanding the concepts being taught rather than having to split their attention between concepts and the nuts and bolts of data manipulation required to complete assignments.

In the second portion of the class, we will take a step back from the nuts and bolts of data manipulation and talk about how to approach the central task of data science: answering questions about the world. In particular, we'll discuss how to use backwards design to plan data science projects, how to refine questions to ensure they are answerable, how to evaluate whether you've actually answered the question you set out to answer, and how to pick the *most appropriate* data science tool based on the question you seek to answer.

2 For Whom Is This Course Meant?

2.1 Pre-Requisites

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

- A familiarity with basic python
- A familiarity with basic statistics (i.e. what you'd get from an intro stats course)
- 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, dictionaries, and defining functions. Unfortunately, if you haven't worked with Python before, I'm afraid you will likely find this course hard to follow.

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.

Much of this course will focus on learn about and getting experience working with the Python packages `numpy` and `pandas`. **While familiarity with these packages is not an explicit pre-requisite for this course, you should be aware that incoming MIDS students have been exposed to these packages through *DataCamp* tutorials they completed over the summer. As a result, if you come into this course without ever having seen those packages, you may have to do some extra work since you'll be seeing these packages for the first time. This should not prevent you from being able to succeed in this course, but if you are in this position please talk to me after class to make a plan.**

2.2 For Whom Would This Course Be Inappropriate?

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, as the first portion of this course might be somewhat boring for you. 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. Moreover, project design should be new to anyone who hasn't worked in the data science field.

3 What Do You Mean By Data Science?

There are, broadly speaking, two branches of what is often referred to as Data Science, which I will term *Software Development Data Science* and *Data Analysis Data Science*.

In *Software Development Data Science*, programmers write programs that gets bundled up in software and distributed widely, or gets run on the cloud for millions of people. For example, software development data scientists wrote the recommendation engine that lets Netflix tell you what movies you might enjoy, or what people might be your friends on Facebook. As a result, they generally write *generalizable* code that is designed to run on data with a known structure.

In *Data Analysis Data Science*, the data scientist is generally employed to answer a single, specific question. For example, a Data Analysis Data Scientist may be hired to figure out how to reduce anti-biotic resistant infections in a hospital, or to identify what campaign promises are most likely to convince voters to support a politician. As a result, Data Analysis Data Scientists are generally writing code that is only meant to be used for their specific project. Moreover, Data Analysis Data Scientists don't generally have the luxury of working with data with a known structure – where a Netflix Data Scientist may get data from a company database that's clean and well organized, a Data Analysis Data Scientist may have to work with data that has come from lots of different sources and which no one has cleaned and organized (e.g. notes from nurses, or voting data from different states compiled by hand by minimum wage government employees).

To be clear, these branches are not completely distinct. Most data scientists do things that fall into both categories (for example, even a Software Developer will likely do some *ad hoc* analyses before developing a fully deployable tool). But these two types of data science do emphasize different skills. Software Development Data Scientists, for example, are well served by traditional computer science curricula, and need a much deeper understanding of concepts like object-oriented programming, and software deployment. By contrast, Data Analysis Data Scientists need to be comfortable working with data in different formats, and to understand how to clean and fit together datasets that were never actually built to be integrated.

The focus of this course will be on the skills of Data Analysis Data Science: cleaning and merging data, data exploration, and designing projects to answer very specific questions. If you're interested in policy analysis, or health-sector analysis, or applied empirical research, this course is for you; if you're interested in developing programs you can deploy in an iPhone app to improve recommendations, then while there will be material that will be of use to you (the Python data science stack, working at the command line, git and github), the emphasis of the material won't quite be what you're looking for.

4 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.

5 Class Organization

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

To make it possible for us to work through problems together as they arise, we will dedicate most of our class time to completing these exercises in small groups. That means that students will be required to read instructional material *before every class* so they will be ready to do these exercises. This is what is referred to as "flipping the classroom."

In order to make this class organization work, it will be ***critically*** important that students do their assigned readings before *every* class, and as discussed below, this will be reflected in how grades are assigned in this class. Students who do not complete their assigned readings and tutorials before each class should not expect to receive good grades, regardless of performance on project assignments.

6 Assignments & Grading

6.1 Participation (25% of Grade)

Note that a major component of good participation is good *preparation*. Because we will mostly reserve class time for hands-on exercises, it is absolutely critical that students do their assigned readings before *every* class. Students who do not work through the instructional materials they have been assigned before class will not only get very little out of the hands-on exercises designed to reinforce the assigned materials, but they will also undermine the learning of the students they are asked to work with. With that in mind, students who do not complete their assigned readings before every class should expected to see this reflected in their participation grades.

Participation will be graded as follows:[1]

[1]This rubric is adapted from that of Duke Political Science Professor Adriane Fresh.

A range. You are fully *and consistently* engaged in class discussion and exercises. You both listen and contribute actively. You are well prepared for class. Having done more than merely read the material, you have spent time thinking *carefully and deeply* about the material's relationship to other materials and ideas presented in previous classes. You are not only able to answer questions about the material, but also come to class with thoughtful questions. When working in teams, you work *with* your partner. If your partner is struggling with an exercise, you help them understand the material rather than just completing the material on your own. If you are struggling with material, you ask for help (both from the instructor and your fellow students) and do not simply lean on your partner to complete the exercise.

B range. You are engaged in class discussion and exercises. You listen and contribute regularly. You come well-prepared to class having read the material and your contributions show your familiarity, but your level of engagement lacks the depth accumulated through extra time spent thinking about the material. When working in teams, you work *with* your partner when they have a similar level of understanding, but do not always invest in helping a struggling partner to understand the material. You often ask for help when you are struggling, but other times you let your partner just complete the exercise.

C range. You have met the minimum requirements of participation. You are usually, but not always prepared. You participate sometimes, but not regularly. The comments that you offer show a basic familiarity with the materials, but do not help to build a coherent or productive discussion. When working in teams, you only sometimes work *with* your partner. When your partner is struggling, you often just do the exercise yourself. If you are struggling, you often do not ask for help and allow your partner to take over the exercise.

D range. You have not met the minimum requirements of participation. You are unprepared for class. You have not read with the material with sufficient engagement to know even the most basic elements. When working in teams, you do not attempt to work *with* your partner. When your partner is struggling, you just do the exercise yourself. If you are struggling, you do not ask for help and allow your partner to take over the exercise.

As should be clear from this rubric, above all it is important to emphasize that participation is evaluated on the basis of *quality* and *consistently*, *not* quantity. Moreover, when completing in-class exercises, good participation is not about finishing first or without ever asking for help; good participation in in-class exercises is about helping your partner understand the material, and asking for help when you need it.

If students consistently fail to come to class prepared, the instructor reserves the right to introduce quizzes at that start of each class to directly evaluate class preparation.

6.2 Interim Assignments (25% of Grade)

Over the course of the semester, students will be asked to complete a number of small assignments as homework. These assignments will, in total, be worth 25% of student grades.

6.3 Mid-Semester Data Science Project. 25%

At the end of Part 1 of this course, students will be assigned a mid-term Data Science Project. The goal and general framework for this team project will be provided to students, but the project will require students to complete the analysis component of a full data science project, including gathering data, cleaning and merging that data, analyzing the data, and presenting results.

6.4 Final Data Science Project Proposal. 25%

At the end of Part 2 of this course, students will be required to submit a Data Science Project Proposal. Using backwards-design principles, this proposal will include not only a tractable, answerable question, but also specification of what the answer to that question will look like, what data will be required to generate that answer, and a strategy for managing project workflow.

6.5 Late Assignments, Make Up Exams and Extra Credit

Grading All assignments will be given a numerical score on a 0-100 scale. These scores will be multiplied by the value of the assignment (see above) and the following scale will be used to assign a final letter grade.

98-100 A+	88-80.9 B+	78-79.9 C+	60-70 D
93-97.9 A	83-87.9 B	73-77.9 C	below 60 D
90-92.9 A-	80-82.9 B-	70-72.9 C-	

7 Texts

We will rely on two primary texts for this course (both of which, thankfully, are reasonably priced):

- *Python Data Science Handbook: Essential Tools for Working with Data* by Jake VanderPlas. Referred to in the syllabus as JVP.
- *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, Second Edition* by Wes McKinney. Referred to in the syllabus as WM.
  Make sure to buy the Second Edition!.

We will also do some readings from Code: The Hidden Language of Computer Hardware and Software by Petzold, Charles. It's a fun book and not very expensive, but we won't use it a lot so copies of relevant chapters will be provided if you don't want to buy it.

Week Start Date	Week	Class	Topic
26-Aug	Week 1	Class 1	Intro
26-Aug	Week 1	Class 2	Command Line Basics
2-Sep	Week 2	Class 1	Advanced Command Line
2-Sep	Week 2	Class 2	Jupyter Lab / Notebooks
9-Sep	Week 3	Class 1	Python v. R / variables as pointers
9-Sep	Week 3	Class 2	Numpy Basics
16-Sep	Week 4	Class 1	Numpy Numeric Data Types
16-Sep	Week 4	Class 2	Pandas: Series & DataFrames
23-Sep	Week 5	Class 1	Pandas: Indices & Missing
23-Sep	Week 5	Class 2	Pandas: Loading and saving data
30-Sep	Week 6	Class 1	Pandas: Cleaning
30-Sep	Week 6	Class 2	Pandas: Merging
7-Oct	Week 7	Class 1	FALL BREAK
7-Oct	Week 7	Class 2	Pandas: Groupby / Split Apply Combine
14-Oct	Week 8	Class 1	Pandas: Reshaping
14-Oct	Week 8	Class 2	Pandas: Categorical Data; Eval and Query
21-Oct	Week 9	Class 1	Collaborating using Github
21-Oct	Week 9	Class 2	Getting Help Online
28-Oct	Week 10	Class 1	Defensive Programming
28-Oct	Week 10	Class 2	Plotting with plotnine
4-Nov	Week 11	Class 1	Statistics with statsmodels
4-Nov	Week 11	Class 2	Machine Learning with sckikit-learn
11-Nov	Week 12	Class 1	Strings
11-Nov	Week 12	Class 2	UNSCHEDULED FOR FLEXIBILITY
18-Nov	Week 13	Class 1	UNSCHEDULED FOR FLEXIBILITY
18-Nov	Week 13	Class 2	Data Science: Questions
25-Nov	Week 14	Class 1	Data Science: Backwards Design
25-Nov	Week 14	Class 2	THANKSGIVING BREAK
2-Dec	Week 15	Class 1	Data Science: Backwards Design II
2-Dec	Week 15	Class 2	Project Proposal Workshopping
9-Dec	Week 16	Class 1	Project Proposal Workshopping
			FINALS BEGIN DEC 11TH

8 Course Schedule

Because one aim of this course is to ensure that all MIDS students have a solid foundation for their time at Duke, the exact organization of this course is likely to change regularly as the course proceeds. Students will therefore be expected to regularly (i.e. before every class) check on the updated course schedule (which will include assignments for the next class) at `www.praticaldatascience.edu`.

However, what follows is an approximate outline of the topics we will cover: