Nicholas Eubank
Assistant Research Professor

MIDS NUMBER
Fall & 2019

Programming for Data Science

1 Course Description

The aim of this course is to provide an introduction to the principles and concepts of programming. While there will be many similarities between this course and an introductory computer science course, this course is designed specifically for data science, and as such will emphasize methods for analyzing real world data rather than the "software development" skills (learning to write applications and programs) often taught in computer science introductory courses.

The course will be focused in large part on teaching students R, an extremely popular statistical programming language. However, this is not a course *about* R; rather, while we will use R in this course, we will be doing so as a means of teaching generalizable principles that will apply in any programming language.

In addition to working with R, this course will also provide training in a number of ancillary tools that are often overlooked in the training of data scientists, but which are absolutely critical to the day-to-day life of a data scientist, including:

- Git and Github (for collaboration and project management)
- The Command Line / Terminal
- Getting Help Online (no seriously – there's more to it than you may think!)

2 Learning Goals

By the end of Programming for Data Science, our goal is for you to be able to:

- Load data in R
- Manipulate the data in basic ways (like tabulating results)
- Clean and merge *real world dirty data* for analysis
- Organize your workflow for a project
- Program in a manner that minimizes the likelihood of mistakes, and maximizes the likelihood that when mistakes occur, you will catch them
- Find help online when you get stuck

- Collaborate with others using git and github

3 Required Programming Background

None.

We have *absolutely no expectations that students will have any experience with programming!* This is an introductory course, and save MIDS bootcamp programs, we fully recognize that many students have never worked with statistical software. That's FINE! If you know how to use google and email, you have plenty of experience – everything else we'll take care of.

If do have experience with a programming language, however, worry not: in my experience, most people who learned to use tools like R, Stata, or Matlab did so in a somewhat haphazard manner. They were given some code, they learned to emulate it, and they can now stitch together code that does what they want. But most students were never taught any of the organizing principles of programming. So if you are one of those students, you may find parts of this course easier than other students, but you will still come away with a new, deeper understanding of these tools that should make you more comfortable and productive in your life as a data scientist.

Not that at times during this course, I may make statements about how the tools we're learning (like R) compare to different tools (like Stata or Python) if there are students with experience in these other tools in the class. However, if we make those comparisons, it is only to help those students the traps one can fall in if one's background is in another language. It is in no way because we *expect* all students to have experience with other tools.

4 R

In this class we will be focused on teaching you to use a program for statistical analysis called *R* (yup, just the letter).

Why R? Because it's currently one of the two most-used programs in data science (the other being Python, which we'll work with in Advanced Programming for Data Science), which means there is a good chance you'll be called upon to use it when working in teams. Moreover, it's a much easier tool to get started with than a language like Python.

It is worth emphasizing that we're not teaching you R because we think it is the best. 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.). People develop really strong opinions about what language is *best*, and sometimes pass judgement on people who use other languages. We would like to discourage this type of thinking. Personally, I (Nick) regularly work in at least four different programming languages depending on which is best suited to the task at hand, so I think I have reasonable authority to say: there is no single *best* language for all purposes.

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 R, we feel confident that we will not only be providing you with tools that will allow you to do most everything you'll want to do in graduate school, but we will also be providing you with *generalizable* skills around data manipulation that you will find useful if you later change platforms.

5 Class Organization

In this class we will be "flipping the classroom" – that means that most weeks, you will be **required** to review tutorials between classes so we can spend our class time doing hands-on programming exercises in an environment where help will be available. These tutorials will not generally be very long, and I **strongly** recommend that while you read through them you do so with an open programming session so you can just play around a little, trying out the things you learn. The research on learning to program is exceedingly clear on this point: **the only way to learn to program is to actually program**, so the more time you spend playing with the tools we are using, making mistakes, and troubleshooting, the more you will learn.

6 Course Schedule

PART I. R

- Week 1: Getting to Know R; Variables
- Week 2: Vectors and DataFrames
- Week 3: Data Cleaning and Manipulation with dplyr
- Week 4: Merging Data; For-Loops
- Week 5: Plotting
- Week 6: Functions
- Week 7: Lists, [other?]

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

- Week 8: The Terminal / Command Line
- Week 9: Git and Github
- Week 10: Getting help online; Jupyter Labs
- Week 11: Workflow management

PART III. Programming, a CS Perspective

- Week 12: Defensive Programming, Decomposition
- Week 13: Data Types (Floats, Ints, Strings, etc.)

PART IV. Other Languages

- Week 14: Trade-offs of Different languages; Python
- Week 15: Python Libraries for Data Science (numpy, pandas)