Data science for the liberal arts

Kevin Lanning

2019-01-04

Contents

an invitation

This work-in-progress includes my notes for Introduction to Data Science at the Wilkes Honors College of Florida Atlantic University.

Data science is still a new field of study, and there are multiple approaches to teaching it and to its place in the college curriculum. This course is like those at the universities of North Carolina, British Columbia, Duke, Maryland, Wisconsin, Stanford, BYU, Harvard, Pennsylvania, and UC Berkeley (and will likely draw from all of these) in that it is closer to Statistics than to Computer Science.

But if our approach is closer to statistics than to programming, it is particularly close to statistics in its most applied and pragmatic form. The choice of statistical methods should follow from the data and problem at hand - or, as Loevinger (1957) once put it, statistics should be the handmaiden of real-world concerns rather than technology.

This pragmatic focus is not unique. Courses with similar goals (which we may again draw from) include those at Chicago, Georgia Tech, UC Santa Barbara, Princeton, UC Berkeley, at Berlin's Hertie School of Governance, and in Columbia's School of Journalism.

what will be in the class?

R

In my 2017 survey of introductory data science courses, I saw a pretty even split between those which begin with Python and those which begin with the statistical programming language R. This difference corresponds, loosely, to the split noted above: Computer science based approaches to data science are frequently grounded in Python, while stats based approaches are generally grounded in R. Our course, like those for most of the syllabi and courses linked above, will be based in R.

Reproducible science

The course will provide an introduction to some of the methods and tools of reproducible science. We will consider the replication crisis in the natural and social sciences, and then consider three distinct approaches which serve as partial solutions to the crisis. The first of these is training in a notebook-based approach to writing analyses, reports and projects (using R markdown). The second is using public repositories (such as the Open Science Framework and GitHub) to provide snapshots of projects over time. Finally, the third is to consider the place of significance testing in the age of Big Data, and to provide training in the use of descriptive, exploratory techniques of data analysis.

Good visualizations

Part of Type C data science is communication, and this includes not just writing up results, but also designing data displays that incisively convey the key ideas or features in a flood of data. We'll examine and develop data visualizations such as plots, networks and text clouds. More advanced topics may include maps, interactive displays, and animations.

~~All~~ *Some of* the data

It's been argued that in the last dozen years, humans have produced more than 60 times as much information as existed in the entire previous history of humankind. (It sounds like hyperbole, but even if it's off by an order of magnitude it's still amazing). There are plenty of data sources for us to examine, and we'll consider existing datasets from disciplines ranging from literature to economics to public health, with sizes ranging from a few dozen to millions of data points. We will also clean and create new datasets.

~~All~~ *Some of* the tools

In addition to R, we'll use a range of other tools: We'll communicate on the Slack platform. We'll write using markdown editors such as Typora. We'll certainly use spreadsheets such as Excel or Google Sheets. We *may* use additional tools for visualizing data such as Gephi and Tableau. In any event, **there will be computing throughout the course.** You will be expected to bring a laptop every day. (Please let Dr. Lanning know ASAP if you don't have access to this).

The place of data science in the WHC (and FAU) curriculum

At this writing, there is enthusiasm across units of FAU and its affiliated institutes, including Max Planck and FAU's Colleges of Science and Engineering as well as the WHC, for integrating data science into our curriculum. Within the WHC, a data science minor and a multi-track concentration are under development. Until these proposals have been formally approved, students interested in concentrating in Data Science are encouraged to pursue an individual concentration (see Dr. Lanning for details).

In addition, there are several integrated '4 + 1' pathways which will lead to a master's degree in the College of Engineering. These programs are also in progress; again, see Dr. Lanning for additional details.

should you enroll?

It's my intention that this course should serve every Wilkes Honors College student, regardless of concentration. The WHC was built and funded by the state of Florida to train tomorrow's leaders. That's you. The skills and insights that you will gain in this course will help you in graduate and professional schools, will help you in your careers, and will help you in your goal of making a better world. And it will help you train the next generation of data scientists, too.

How do I sign up?

If you are interested in taking the class, complete the Google form here. Note that enrollment in the class and lab will be limited to 30.

Part I

Introduction

Chapter 1

data science for the liberal arts

Hochster, in Hicks and Irizarry (2017), describes two broad types of data scientists: Type A (Analysis) data scientists, whose skills are like those of an applied **statistician**, and Type B (Building) data scientists, whose skills lie in problem solving or coding, using the skills of the **computer scientist**. This view arguably omits a critical component of the field, as data science is driven not just by statistics and computer science, but also by "domain expertise:"

1.1 type C data science = data science for the liberal arts

The iconic Venn diagram model of data science shown above suggests what we will call "Type C data science." It begins with "domain expertise" in your **concentration** in the arts, humanities, social and/or natural sciences, it both informs and can be informed by new methods and tools of data analysis, and it includes such things as **communication** (including writing and the design and display of quantitative data), **collaboration** (making use of the tools of team science), and **citizenship** (serving the public good, overcoming the digital divide, furthering social justice, increasing public health, diminishing human suffering, and making the world a more beautiful place). It's shaped, too, by an awareness of the fact that the world and workforce are undergoing massive **change**: This puts the classic liberal arts focus of "learning how to learn" (as opposed to memorization) at center stage. And Type C data science is shaped, not least, by the **creepiness** of living increasingly in a measured, observed world.

Type C data science does not merely integrate 'domain expertise' with statistics and computing, it places content squarely at the center. We can appreciate the compelling logic and power of statistics as well as the elegance of well-written code, but for the purposes of this book, these are means to an end. Programming and statistics are tools in the service of social and scientific problems and cultural concerns. Type C data science aims for work which is not merely cool, efficient, or elegant but responsible and meaningful.

1.2 the incompleteness of the data science Venn diagram

Data visualizations are starting points which can provide insights, typically highlighting big truths or effects by obscuring other, presumably smaller ones. The Venn diagram model of data science is no exception: As with other graphs, figures, and maps, it allows us to see by showing only part of the picture. What does it omit? That is, beyond **statistics**, **computing/hacking**, and **domain expertise**, what other skills contribute to the success of the data scientist?

The complexity of data science is such that individuals typically have expertise in some but not all facets of the area. Consequently, problem solving requires **collaboration**. Collaboration, even more than statistical and technical sophistication, is arguably the most distinctive feature of contemporary scholarship in the natural and social sciences as well as in the private sector (Isaacson, 2014).

Communication is central to data science because results are inconsequential unless they are recognized, understood, and built upon; facets of communication include oral presentations, written texts and, too, clear data visualizations.

Reproducibility is related to both communication and collaboration. There has been something of a crisis in recent years in the social and natural sciences as many results initially characterized as "statistically significant" have been found not to replicate. The reasons for this are multiple and presently contentious, but one path towards better science includes the public sharing of methods and data, ideally before experiments are undertaken. Reproducible methods are a key feature of contemporary data science.

Pragmatism refers to the relevance of work towards real-world goals.

Ideally, these pragmatic concerns take into account **ethical concerns** as well.

1.3 a dimension of depth

Cutting across these eight facets (statistics, computing, domain expertise, collaboration, communication, reproducibility, pragmatism, and ethics), a second dimension can be articulated. No one of us can excel in all eight domains, rather, we might aim towards goals ranging from **literacy** (can understand) through **proficiency** (can get by) to **fluency** (can practice) to **leadership** (can create new solutions or methods).

That is, we can think of a *continuum* of knowledge, skills, interests, and goals, ranging from that which characterizes the data *consumer* to the data *citizen* to the data science *contributor.* A Type C data science includes this dimension of 'depth' as well.

1.4 Google and the liberal arts

Data science is at its core empirical, and all of this rhetoric would be meaningless if not grounded in real world findings. Although it was recently reported that soft skills rather than STEM training were the most important predictors of success among Google employees, it's difficult to know whether these results would generalize to a less select group. Nonetheless, there is a clear need for individuals with well-rounded training in the liberal arts in data science positions and, conversely, learning data science is arguably a key part of a contemporary liberal arts education.

1.5 data sci and TMI

One difference between traditional statistics and data science is that the former is typically concerned with making inferences from datasets that are too *small*, while the latter is concerned with extracting a signal from data that is or are too *big* (Donoho (2017)).

The struggle to extract meaning from a sea of information - of finding needles in haystacks, of finding faint signals in a cacophony of overstimulation - is arguably the question of the age. It is a question we deal with as individuals on a moment-by-moment basis. It is a challenge I face as I wade through the many things that I could include in this class and these notes.

The *primacy of editing* or selection lies at the essence of human perception and the creation of art forms ranging from novels to film. And it is a key challenge that the data scientist faces as well.

1.6 discussion: what will you do with data science?

Imagine it is ten years from today. You are working in a cool job (yay). How, ideally, would ‘data science’ inform your professional contributions?

More proximally (closer to today) - what are your own goals for progress in data science, in terms of the model described above?

Chapter 2

getting started

We begin with a brief self-assessment and a description of some rudimentary tools.

Reflect on your own knowledge of data science, including the necessary-but-not-sufficient areas of computer programming and statistics.

2.1 are you already a programmer and statistician?

Regarding **programming**, you may know more than you think you do. Here's a simple program - a set of instructions - for producing a cup of coffee:

> add water to the kettle and turn it on
>
> if it's morning, put regular coffee in the French press, otherwise use decaf
>
> if the water has boiled, add it to the French press, else keep waiting
>
> if the coffee has steeped for four minutes, depress (smash) piston/plunger, else keep waiting
>
> pour coffee into cup
>
> enjoy

As a post-millennial student from a WEIRD culture (a Western, Educated, Industrialized, Rich Democracy; see Henrich et al. (2010)), you've 'programmed' computers, too, if only to enter a password, open an app, and upload a photo on your cell phone.

Statistics is, of course, relevant in a myriad of ways. Consider, for example, the prospects for Susie:

> **Exercise 2_1** *Susie is applying to two med schools. At School A, 25% of students are accepted, and at School B, 25% are accepted as well. You are Susie. Are you going to get in to at least one of these programs? What is the probability? Does your estimate depend upon any assumptions?*

Questions like these are important for us. If the combined probability is low, it *likely* (another probability concept) makes sense for Susie to spend time, money, and energy to apply to additional programs. If the probability is high, it may not. But problems like this are hard - our estimates of probability are frequently poorly calibrated, and combining probability estimates is challenging (see, e.g., Tversky and Kahneman (1974), and consider taking a course in *Behavioral Economics* or *Thinking and Decision Making* to learn more).

You may have worked with **data** in spreadsheets such as Excel or Google Sheets.

> **Exercise 2_2** Open the Google Sheet at http://bit.ly/dslaX2_1. Save a copy and edit it, entering the following in cell A6:
>
> *=SUM (A1:A5)*
>
> What is the result? **If you copy cell A6 to B6, what happens and why?**

In data science, spreadsheets are used largely to store data rather than to analyze it. Some *best practices* for using spreadsheets in data science are given in Broman and Woo (2017).

2.2 setting up your machine: some basic tools

Collaboration and communication are integral to data science. In the world beyond universities, the most important messaging and collaboration platform is **Slack.** Slack is a commercial app, but we will use the free tier. We'll use Slack for group work, class announcements, and help-seeking and help-providing.

Slack includes a simple *markdown* editor (for 'posts'). You can find an introduction to markdown syntax in Chapter 3 of Freeman and Ross (2017). I use **Typora** (currently free for both Windows and Mac), but there are many alternatives. Install a Markdown editor on your laptop and play with it.

Install **R** (https://cran.rstudio.com/) then **R studio** (https://www.rstudio.com/products/rstudio/#Desktop) on your own Windows or Mac laptop. If you get stuck, reach out to others on Slack; if you don't get stuck, help your classmates. We'll use R studio as a front end (an 'integrated development environment', or IDE) for R, and will write most of our code in R markdown which is, not surprisingly, a 'flavor' of markdown. We'll go into R in increasing depth beginning in the next chapter; if you want to get a head start, consider Carmichael (2017) Getting started and the first chapter of Wickham and Grolemund (2016). (Those documents, like this one, are all written in R markdown).

Finally, **Google Docs** is free and is convenient for collaborative work. One other important feature of Google Docs is that it provides a framework for *version control,* a critical skill in information management. You can learn more about how to see and revert to prior versions of a project in Google Docs here .

Version control can help you avoid the chaos and confusion of having a computer (or several computers) full of files that look like Cham's (2012) comic:

.

Never call anything 'final.doc'.

We'll be talking about the challenge of version control throughout this text - and I am hoping that my own habits in file management can improve as we move forward together.

2.3 discussion: who deserves a good grade?

In an introductory class in data science, students invariably come to class with different backgrounds. Should this be taken into account in assigning grades? That is, would it be possible (and desirable) to assign grades in a class based not just on what students know at the end of the term, but also on how much they have learned?

A formal, statistical approach to this could use regression analysis. That is, you could predict final exam scores from pretest scores, and use the residuals - the extent to which students did better or worse than expected - as a contributor to final exam grades. Interestingly, there would be an unusual incentive for students on this 'pretest' to do, seemingly perversely, as poorly as possible. How could this be addressed?

Another problem with this approach is that there may be 'ceiling effects' - students who are the strongest coming in to the class can't improve as much as those who have more room to grow. Again, how might this be addressed?

Chapter 3

an introduction to R

R is a system for *Reproducible* science, and reproducibility is essential (Baumer et al., 2014). R is a system for *Representing data* in cool, insight-facilitating ways. R is *Really popular*, and really growing. Learning R will make you a more attractive candidate for many graduate programs as well as jobs in the private sector.

3.1 um. in a sense, it's not possible to 'learn R'

Unlike, say, learning to ride a bicycle, fry an egg, or drive a car with a manual transmission, learning R is not a discrete accomplishment that one can be said to have mastered and from which one then moves on. Rather, R is an evolving, open system of applications and tools which is so vast that there is always more that one can achieve, new lessons that one can learn.

3.2 what R stands for

Historically, R grew out of S which could stand for Statistics. But what does R stand for?

R does not stand for 'argh,' although you may proclaim this in frustration ('arggh, why can't I get this to work?) or, perhaps, in satisfaction ('arggh, matey, that be a clever way of doing this').

R might stand for *Relatively high level.* Programming languages can be described along a continuum from high to low level, the former (like R) are more accessible to humans, the latter (like assembly language) more accessible to machines. Python, Java, and C++ are all more towards the middle of this continuum.

R stands, in part, for *Resources.* Because R is popular, there are many resources, including, for example -

- Online resources include the simple (and less simple) lessons of SwirlR, which offers the possibility of "learning R in R," as well as DataCamp, the Data Science Certificate Program at Johns Hopkins, and other MOOCs.

- Books include Peng (2015) - which includes not only videos of his lectures in the program at Hopkins, but also a brief list of still more resources - and Wickham and Grolemund (2016).
- You'll also learn (more directly) from people, including your classmates, as well as the broader community of people around the world. There are hundreds if not thousands of people, young and old, who are on the road with you. I am as well, just a step or two (hopefully) ahead.

Most importantly, I hope and expect that you'll find that R stands for ***Rewarding***. A language is a way of thinking about the world, and this is true for computer languages as well. You'll be challenged by its complexity, its idiosyncracy, its alien logic. But you will succeed, and you will find that you can do things that you did not believe possible.

R Studio, the environment we will use to write, test, and run R code, is a commercial enterprise whose business model, judged from afar, is an important one in the world of technology. Most of what R Studio offers is free (97% according to Garrett Grolemund in the video below). The commercial product they offer makes sense for a relative few, but it is sufficiently lucrative to fund the enterprise. The free product helps to drive the popularity of R studio; this widespread use, in turn, makes it increasingly essential for businesses to use. This mixed free/premium, or 'freemium,' model characterizes Slack as well, but while the ratio of free to paid users of Slack is on the order of 3:1, for R it is, I am guessing, an order of magnitude higher than this.

3.3 a few characteristics of R

R includes the base plus thousands of **packages**. These packages are customized add-ons which simplify certain tasks, such as text analysis. But there are, at this writing, over 50 different packages for text analysis - so where do you begin? One recent answer, and where we will start, is the curated list of packages which jointly comprise the tidyverse (Wickham and Grolemund, 2016).

> A few years ago, Peng (2015) speculated that "it would be straightforward to build an R package for ordering pizza." Does one exist now?

R is an **object-oriented** language - one conceptually organized around objects and data rather than actions and logic. In R, at the atomic level, objects include *characters, real numbers, integers, complex numbers, and logical.* These atoms are combined into vectors, which generally include objects of the same type (one kind of object, 'lists,' is an exception to this; Peng, 2015). Vectors can be further combined into **data frames**, which are two-dimensional tables or arrays. A **tibble** is a particular type of data frame which is, in some ways, handier to work with than other data frames. We'll be working extensively with data frames in general, and tibbles in particular, as we move forward.

Objects have attributes. Attributes of R include such things as name, dimensions (for vectors and arrays), class (that's the atomic thing), length, etc.

Real world data sets are messy, and frequently have **missing values.** In R, missing values may be characterized by NA (not available) or NaN (not a number, implying an undefined or impossible value).

3.4 finding help

To get a sense of some of the ways you can get help in R studio (and to see how a master uses the R Studio interface), consider the video at RStudio Essentials Webinar Series – Programming Part 1 – RStudio

For us, the key ideas in "looking for help" will include not just the tools on the R Studio IDE, but also (a) using google searches wisely, and (b) reaching out to your classmates on Slack.

There is an etiquette for help-seeking here as in the real world. Your search for help should begin by making sure that others will encounter the same result, then by stripping the problem down to its essence. Once you have reduced the problem to this *minimal, reproducible* essence, you will often be able to spot the problem yourself - and, if not, you will make it easier for others to help you. There is an R package (reprex) which will likely facilitate this, but I haven't tried it yet. Here is a good introduction.

3.5 discussion: is open-source software secure?

Perhaps the most important feature of R is that it is open-source software. This is important not just because it saves you money, but because contributing to the world of R is an act of digital democracy. In using and contributing to the world of R we open up knowledge to others who may lack our privileges. R, like Android or Wikipedia, is a tool for all of us, maintained and continually improved upon by the crowd.

But is open-source software safe? More generally, in a data-dependent world, who should be the guardians of the code that connects us?

Securing the Internet of Vehicles. To consider just one example, the computer systems in modern cars typically run millions of lines of code. As cars become increasingly autonomous, this complexity will only increase. (Incidentally, the Society of Automotive Engineers, or SAE, describes 6 levels of 'auto autonomy.' At this writing, the most sophisticated systems available to consumers, such as Tesla Autopilot, are at level 2. What lies ahead are cars which are self-driving on carefully selected, geo-fenced roads, and ultimately cars "which can operate on any road… a human driver could negotiate"). Our roads and highways will become an Internet of Vehicles (IOV), which will include not just connections between cars and an intelligent *cloud* 'above us' but also direct links between a distributed system of intelligent cars, stoplights, and road sensors in a *fog* 'around us' (Bonomi et al., 2012). **Fog computing** and the IOV will reduce travel times and increase both fuel efficiency and automotive safety.

Obviously, there are **cybersecurity** concerns. While the prospects for a chaotic, choreographed hack of hundreds of vehicles on the streets of Manhattan, such as that in the 2017 movie "The Fate of the Furious", are remote at best (or worst), there have been examples of "white-hat hackers" who have successfully infiltrated (and thereby helped secure) car information systems.

As the IOV develops, there will be vulnerabilities to privacy as well as safety, and the security of the system will be paramount. Different car manufacturers are taking different approaches to developing secure information systems, with many using a closed or proprietary approach. But the scope of the problem is so large that there is a movement towards pooling resources and encouraging collaboration among industry partners, academics, and citizen scientists in the development of an open-source autonomous driving platform, such as Apollo. Perhaps counterintuitively, there may be significant security advantages to using source code that is open to all (Clarke et al., 2009; FitzGerald et al., 2016).

Chapter 4

references

Bibliography

Baumer, B., Cetinkaya-Rundel, M., Bray, A., Loi, L., and Horton, N. J. (2014). R markdown: Integrating a reproducible analysis tool into introductory statistics. *arXiv preprint arXiv:1402.1894*.

Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM.

Broman, K. W. and Woo, K. H. (2017). Data organization in spreadsheets. *The American Statistician*, (just-accepted).

Clarke, R., Dorwin, D., and Nash, R. (2009). Is open source software more secure? *Homeland Security/Cyber Security*.

Donoho, D. (2017). 50 years of data science. In *Princeton NJ, Tukey Centennial Workshop*.

FitzGerald, B., Levin, P. L., and Parziale, J. (2016). *Open Source Software & the Department of Defense*. Center for a New American Security.

Freeman, M. and Ross, J. (2017). *Technical foundations of informatics, U Washington INFO 201*.

Henrich, J., Heine, S. J., and Norenzayan, A. (2010). The weirdest people in the world? *Behavioral and brain sciences*, 33(2-3):61–83.

Hicks, S. C. and Irizarry, R. A. (2017). A guide to teaching data science. *The American Statistician*, (just-accepted):00–00.

Isaacson, W. (2014). *The Innovators: How a Group of Inventors, Hackers, Geniuses and Geeks Created the Digital Revolution*. Simon and Schuster.

Loevinger, J. (1957). Objective tests as instruments of psychological theory. *Psychological reports*, 3(3):635–694.

Peng, R. D. (2015). *R programming for data science*. Lulu. com.

Tversky, A. and Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *science*, 185(4157):1124–1131.

Wickham, H. and Grolemund, G. (2016). R for data science.