

Working with Data**Professor Benjamin Schmidt**benschmidt.org/WWD20Email: bs145@nyu.edu

Office Hours: Tuesday, 10-12, 20 Cooper Sq Room 520

Overview

Data analysis in the humanities presents challenges of scale, interpretation, and communication distinct from the social sciences or sciences.

This seminar will explore the emerging practices of data analysis in the digital humanities from two sides: a critical perspective aiming to be more responsible readers of cultural analytics, and a creative perspective to equip you to perform new forms of data analysis yourself.

Our goal is to make it possible to merging forms of data analysis taking place in humanities scholarship, both in terms of applying algorithms and in terms of better investigating the presuppositions and biases of the digital object. We'll aim to come out much more sophisticated in the use of computational techniques and much more informed about how others might use them.

Some of the key questions we'll aim to answer are:

1. What light can algorithmic approaches shed on live questions in humanistic scholarship?
2. How can you come to understand a new algorithm?
3. What new forms of research are enabled by the use of data?
4. What sort of data do practicing humanists want museums and libraries to make available?

A wide variety of types of data will be used but we will focus particularly on methods for analyzing *texts* in the context of other methods. If your interests lie elsewhere, don't worry too much—as you'll learn, most of the textual approaches we'll consider are easily adaptable for (and in many cases, originally developed for) other sources of data.

Over the course of the semester, you should work to develop your own collection of data. Working with these texts will allow us to ask more sophisticated questions on large documents of scholarly importance.

% Course Goals

1. Be able to contribute to debates about the place of data analysis in the humanities from both a technical and theoretical perspective, in a way that lets you responsibly elicit “data” (or as Johanna Drucker would have it, “capta”) out of more humanistic stores of knowledge.
2. Acquire proficiency in the manipulation, transformation, and graphic presentation of data in the R programming language for use in the context of exploratory data analysis and presentation.
3. Know the appropriate conditions for using, and be able to use, some of the major machine learning algorithm for data classification, clustering, and dimensionality reduction.
4. Execute projects creatively deploying and combining these methods in ways that contribute to humanistic understanding.

Coding and Scripting in R

This course will have you writing some code in the R language. There is an extensive debate about whether digital humanists need to learn to code which we're not going to engage in; the fact of the matter is simply that if you want to either *do* data analysis in the humanities, coding will often be the only way to realize your personal vision; and if you want to *build resources in the humanities* that *others* might want analyze, you'll need to know what sophisticated users want to do with your tools to make them work for them.

I have no expectation that anyone will come out of this a full-fledged developer. In fact, I hope that by doing some actual scripting, you'll come to see that these debates over learning to code brush over a lot of intermediate stages. We'll be focusing in particular in developing skills less in full-fledged "programming," but in "scripting." That means instructing a computer in every stage of your work flow; using a language rather than a Graphical User Interface (GUI), which may be almost all the program you've used before. This takes more time at first, but has some extraordinary advantages over working in a GUI:

1. Your work is saved and open for inspection.
2. If you want to discover an error, you can correct it without losing the work done after.
3. If you want to amend your process (analyze a hundred books instead of ten, for instance) but do the same analysis, you can alter the code only slightly.

Why R?

First off: you don't *have* to do R. If you already know python and want to build on that, it is possible to do almost everything in this class using pandas for dataframe analysis and altair for visualization. At some point, I may try to rewrite the whole class to support *either*.

In this class, you will have to do some coding as well about *data*. Exploratory data analysis which operates on as just thinking about data analysis in the humanities. If you've never coded before, this will be frustrating from time to time. (In fact, if you've done a lot of coding before, it will *still* be frustrating!)

We'll be working entirely in the "R" language, developed specifically for statistical computing. This has three main advantages for the sort of work that historians do:

1. It is easy to download and install, though the program RStudio. This makes it easy to do "scripting" rather than true programming, where you can test your results step by step. It also means that R takes the least time to get from raw data to pretty plots of anything this side of Excel. RStudio also offers a number of features that make it easier to explore data interactively.
2. It has a set of packages we'll be using for data analysis. These packages, whose names you will scattered through this text, are `ggplot2`, `tidyr`, `dplyr`, and the like. These are not core R libraries, but they are widely used and offer the most intellectually coherent approach to data analysis and presentation of any computing framework in existence. That means that even if you don't use these particular tools in the future, working with them should help you develop a more coherent way of thinking about what data is from the computational side, and what you as a humanist might be able to do with it. These tools are rooted in a long line of software based on making it easy for individuals to manipulate data: read the optional source on the history of database populism to see more. The ways of thinking you get from this will serve you well in thinking about relational databases, structured data for archives, and a welter of other sources.

3. It is free: both “free as in beer,” and “free as in speech,” in the mantra of the Free Software Foundation. That means that it—like the rest of the peripheral tools we’ll talk about—won’t suddenly become inaccessible if you lose a university affiliation.

R vs. Python vs. Javascript: which is the best language for humanities computing? Different computer languages serve different purposes. If you have taken ever taken an introductory computer science course, you might have learned a different language, like python, Java, C, or Lisp.

Although computing languages are equivalent in a certain, abstract sense, they each channel you towards thinking in particular ways. So when we

Which of these languages is best? It depends, obviously, on what you want to do. If you only learn a single language, there’s a strong argument that it should Python, which is a widespread, swiss-army-knife type language that can frequently run quite quickly. But python generally promotes a kind of thinking about how you can get a problem done.

What R—especially tidyverse R—does best is let you abstract back from thinking about *programming* to thinking about *data*. Exploratory data analysis which operates on a particular base class, the ‘data frame.’ We’ll talk about this more in Chapter 3; but the point is that it provides a coherent, basic language for describing any data set in terms of groupings, summary statistics, and visualization.

The closest analogues to these in other languages are less elegant and less well thought out. Python has widely used tool called pandas for analyzing data that is fast, powerful, and effective. But it is also more challenging for beginners than it need be. If you Google problems with pandas you’ll be confronted with a variety of problems;¹ R for data science has been a little less confusing.

The place of pre-packaged software.

One thing you can’t do in this course, though, is rely on the out-the-box approaches prevalent in many DH programs. ArcGIS or QGIS may be the best way to make maps, and Gephi the best way to do network analysis. But as this is a course in *data analysis*, I want you to think about the fundamental operations of cartography and network analysis as simply subsets of a broader field, which is hard to see from the confines. All of these things are possible in R. And unlike graphical tools, working in a language *saves your workflow*. If you make a map with laboriously positioned points in ArcGIS, your operations aren’t open for inspection. In R, though, every step you take and every move you make can be preserved. This is called *reproducible research*, and it is among the most important contributions you can make when working collaboratively.

Textbook

This course works alongside of an [online textbook](#) that I will try to keep up to date with what we’re working on. This will change over the course of the semester, but the first several chapters are readable right now.

The exercises are linked out of this textbook; they should be discoverable by the time you that need them.

¹Ten years ago, experts tended to pontificate that python was better than R because it had a small standard library, cleaner syntax, and promoted a single way to do things effectively. One of the great ironies of modern data science is that, for programming with data, the situation has almost completely reversed; the pandas library presents a

R Package

Since we're using R for this course, I'm putting the materials for it online as an R "package." That means a bundle of code and data that you can install to your local computer and use.

To use this, you'll need to install R and the RStudio environment that it uses. Some instructions I think should work for this are [here](#)

That package is online at <https://github.com/HumanitiesDataAnalysis/HumanitiesDataAnalysis>. But in general, there should be no reason to access it there; instead, you will install it within Rstudio on your machine or on any machine you work on. (The computers in Data Services at Bobst, for example, should have RStudio on them; you can just pull up this website, run the code below, and have the basic things you need to work with in this class.

The purpose of this is twofold.

1. We're going to learn how to do some basic things that it will be simpler to just have code that does again and again.
2. I'm going to distribute a lot of *data* of various sorts, and this provides a simple way for getting it on to your computer.

```
if(!require(remotes)) {install.packages("remotes")}  
remotes::install_github("HumanitiesDataAnalysis/HumanitiesDataAnalysis", update = FALSE)
```

There are a few other R packages associated with this course that we'll use as we go along, by me and others. They will be installed in similar ways.

In general, you'll then be able to start working by adding the following text to the beginning of your code (make sure it's inside an R Markdown block.)

```
library(HumanitiesDataAnalysis)  
library(tidyverse)
```

Remember to keep this on hand; you will probably need to rerun some of these command quite often.

Requirements

This is an unconventional methods course, because we'll be looking at two very different kinds of methods in tandem. literature and code. Your work should be balanced between these, so you can think about *how* to use the methods we learn.

I assume that you have a computer that runs Windows, Mac OS X, or Ubuntu. If you use some more esoteric operating system, I will not help you with installation.

If you do not own a computer or if your computer is unsuitable for the work in this class (as a rule of thumb: if it cost less than \$1,000 more than 4 years ago), we can set up an environment in the cloud for you to run RStudio in. I generally don't advise this, though, because it makes it harder for you to continue your work later.

Readings and Attendance

You should attend class each week having completed the assigned readings and ready to discuss them. Let me know in advance if you are going to be missing class or if you must attend remotely.

Problem Sets

To help consolidate your programming abilities, there will be weekly problem sets. The first can be completed in any form you like; later ones should be submitted as PDF, or docx files 'knitted' inside R markdown.

Sometimes it will be hard to get them to knit. Post for help on the Slack if you can't get it to work.

They should be completed weekly, submitted over slack. Problem sets are required but ungraded;

Free projects

The course text will generally end with a 'free project' that presupposes that you are working with your own dataset. We'll talk about these datasets starting in the second week; in the third week on, you can also use one that I'll supply of historic New York City population.

Final Projects

These free projects are first stabs towards creating a work of analysis of your own. As we move into the later weeks of the semester, you should figure out which of the the various data sets we've used may be particularly interesting and find a way to build out on the techniques and strategies to create something novel. Most likely, this will be an *experiment* along the lines of the "Quantitative Formalism" pamphlet we read later. In it, you will take the fundamental advantages of a programming environment **combine** some of the various methods and strategies we've learned in a programming environment, or build out some new ones. Appropriate products might include a large-format print map, a set of blog posts exploring generic distances, or a conference-poster style exploration.

Keeping up to date

This is an advanced graduate seminar; I hope that the syllabus will change in response to your own interests and readings.

This flexibility may cause problems of "versioning": what version of the syllabus should you believe? So for the record, the priority for what to do consists of:

1. Any e-mails or announcements in class or on Slack.
2. The current version of the syllabus on the course web site.
3. The most recent paper copy of the syllabus handed out.

Absences and Coronavirus.

In order to protect people, you must not attend class while potentially carrying covid-19. It will generally be an option to Zoom into class if you alert me to a medical reason in advance.

Students attending remotely should generally have cameras on and engage with work in the class as though they were present.

Grading and effort.

As graduate students, you should be starting to get a sense of what is important to *you*; I'm not going to quiz you on individual areas that you read.

But it's also important that you work consistently in this class for it to be of much use.

I have provided below a default rubric for success in this class. It adds up to 110% because I want you to have some choice for how to distribute your energies.

If you have goals that you feel will not be met here, please meet with me in the first two weeks of class and we can come up with an alternate grade contract.

1. Submit the exercise portions of the problem sets each week demonstrating an engaged attempt to complete the problems. These should be done individually, but it's fine to take from other people's answers *if* you credit them. (20%)
2. Post at least 6 of the free exercise attempts to the course slack for various weeks. These may be done collaboratively; they can be handed in up to a week after the rest of the associated problem set without repercussion. (20%)
3. Bonus points for especially interesting free exercise attempts or deep engagement with readings in class. (10%)
4. Complete various, short non-programming assignments described on syllabus, on schedule and usefully. (10%)
5. Attend each class and participate actively and supportively in classroom discussions. (15%)
6. Hand in a final project that demonstrates and develops your skills in the quantitative and descriptive parts of this class. (25%)

Schedule

Mostly we'll be reading articles in this course available online. One are required for purchase. If you have difficulty obtaining any texts, please let me know as soon as possible.

In week 1, you'll read my spiel about what humanists need to understand when they read CS. My answer is, in general—you need to know *what they did*, but not *how they did it*. I've put some CS papers in this syllabus to expand your thinking about what's possible. You should absolutely, positively, not aim to understand the process in a CS paper, or worry about all the notation in the economics or political science papers. As a rule, if you see a fancy equation in an article *not* written by a humanist, you can probably skip the whole section for the time being.

There are two required texts, both online. One is my text for this course at [<http://hdf.benschmidt.org/R>], which includes technical materials and programming code; the other is [*Data Feminism*], by Catherine D'Ignazio and Lauren Klein, MIT Press 2019. It's available open-access online, but you can also purchase a copy if you like. <https://data-feminism.mitpress.mit.edu/>.

The textbook sections include programming exercises, which should be submitted as Word, PDF, or HTML the Thursday after class.

Following Nick Montfort's practice in [Exploratory Programming for the Arts and Humanities](#), each of the problems ends with a "free exercise." I encourage you to post at least 6 of these to the course Slack over the semester; it would also make sense to do these cumulatively as a blog series, Twitter thread, etc.

Defining and transforming

Mon, Jan 24 Introductions

Due Fri, Jan 28: Install R on your computer, or—if you want to use a different programming—meet with me to discuss.

The programs [R](#) and [Rstudio](#) Rstudio is a wrapper program around the R language that we'll be using for almost every assignment.

Mon, Jan 31 What is (could be) Humanities Data Analysis?

Readings

- [@michel_quantitative_2011](#)
- [@drucker_humanities_2011](#)
- Gergely Baics, Wright Kennedy, Rebecca Kobrin, Laura Kurgan, Leah Meisterlin, Dan Miller, Mae Ngai. *Mapping Historical New York: A Digital Atlas*. New York, NY: Columbia University. 2021. <https://mappinghny.com>

Online text

- Introduction/The Gift of Data](<https://hdf.benschmidt.org/R/introduction#the-gift-of-data>)
- [Working in a Programming Language](#)
- [Introduction to Data](#) Merely skim the parts about regular expressions—this is something to come back to when you need it.

Due Mon, Jan 31: Choose two datasets to discuss in class that are relevant to your research interests, so far as you're able to find them.

One should be something that you can actually download, almost certainly in the form of a CSV or Excel File.

The other should be something that you know exists, but that you might not be fully able to work with yet.

For both of them, fill out the [online spreadsheet](#). (That should be editable from your .nyu address.) The goal here is to reduce this to a *tabular* dataset. Describe what each of the fields in this dataset would be. (No more than 10 fields per dataset).

Do not describe the dataset as a whole aside from the columns—see if you can capture it in the individual elements.

Due Mon, Jan 31: Install the course package inside your own copy of R Studio, as described at the end of “Working in a Programming Language chapter.” We'll talk through any problems in class.

Due Thu, Feb 03: Finish the exercises for “Introduction to Data” installing R and Rstudio. You will be complete when you can ‘knit’ your file to a Microsoft Word document, HTML page, or PDF. (Note that PDFs can be a bit harder.)

Mon, Feb 07 Information %>% Data

guest: Nicholas Wolf, NYU Libraries

Readings

- Klein and D'Ignazio, chapters 1 and 2.
- (Much more practical) Hadley Wickham, "Tidy Data," Journal of Statistical Software, 2015
- Browse online: [New York City Directories](#).

Online text

- [The Data Table](#)
- [Counting Things](#)
- Please report issues, and use the "[Troubleshooting Guide](#)"

agenda: [Class agenda](#)

Mon, Feb 14 Data Visualization

Readings

- Jacques Bertin, *Semiology of Graphics: Diagrams, Networks, Maps*, selections. (Shared over Slack)
- D'Ignazio and Klein, *Data Feminism*, [Chapter 3: On Rational, Scientific, Objective Viewpoints from Mythical, Imaginary, Impossible Standpoints](#).
- Read <https://ggplot2.tidyverse.org/reference/>, paying particular attention to the first section; the different "geoms".
- Browse the [Vega Gallery](#); this is the equivalent to ggplot2 in Javascript (and also widely used in Python)
- "Streetscapes: Mozart, Marx, and a Dictator." Die Zeit online, 2018-02-13. [link](#)

Online text

- [Visualizing Data](#)

Related texts not to read

- @daston_objectivity_2007, Chapter 7 (On the Sciences, today.)

Due Mon, Feb 14: ON PAPER, draw a speculative visualization of the New York City directories data and try to describe it in terms of the geoms and marks in Bertin and the ggplot docs. Bring it to class.

Due Wed, Feb 16: [Counting Things](#), exercises

Mon, Feb 21 No class: President's Day

Due Thu, Feb 24: Visualizing data, exercises.

Mon, Feb 28 Counting, grouping, and accounting for how only things that get counted count.

description: A huge amount of work is just about finding interesting things to count. Often, sophisticated work can just be figuring how to count something new. Here we look a little bit at how you can, simply count something.

Readings

- @logan_national_2017 As with all econ in this class, read for the general findings, and data, *not* the methodology.
- @underwood_transformation_2018 (This is basically chapter 4 of the other Underwood, but I like the order in the article version better for this week.)
- D'Ignazio and Klein, *Data Feminism*, Chapter 4. '[What Gets Counted Counts](#)'

Online text

- [Cleaning Data](#)

Related texts not to read

- Too long, too old, and too out of print to assign: but be aware of the granddaddy of them all: @tukey_exploratory_1977

Practicum for next class

- Circle back to the analysis set. Do something more with the collection of book titles.
- If you successful finished much of the last problem set:
- Take a stab at the problems for [Cleaning Data and tidying data](#)

Mon, Mar 07 Data modeling and data merging.

Readings

- Melanie Walsh and Maria Antoniak, The Goodreads "Classics": A Computational Study of Readers, Amazon, and Crowdsourced Amateur Criticism, *[Cultural Analytics 2021](#)
- Tim Berners-Lee, "The Semantic Web", [Scientific American 2001](#).

Online text

- Combining datasets: Merges, joins, and standards.

Due Thu, Mar 10: Problems for Combining datasets: Merges, joins, and standards.

Due Fri, Mar 11: From practicum: place on the course Slack two ggplot visualizations results from a **join** between two different datasets. Try to be goofy on one and serious with the others. You may use text fields if you want.

Mon, Mar 14 No class: Spring Break

Texts, maps, and data

Mon, Mar 21 Text as Data, 1

Readings

- Gentzkow et al, Journal of Economic Literature, <https://doi.org/10.1257/jel.20181020> (Skim only.)
- Klein and D’Ignazio, [Unicorns, Janitors, Ninjas, Wizards, and Rock Stars](#).
- Julia Silge and David Robinson, *Text Mining with R: A Tidy Approach*. Read online at <https://www.tidytextmining.com>. Preface and Chapters 1-4. If you have no interest in analyzing text, you can skip this.

practicum for next class: -“[Texts as Data, exercises](#).”

Due Fri, Mar 25: Place on the course Slack two ggplot visualizations results from a **join** between two different datasets. Try to be goofy on one and serious with the others. You may use text fields for the join.

Mon, Mar 28 Text as Data, 2

description: Once text is data, you can explore and reconfigure it. There are a variety of ways to do this.

Readings

- Stephen Ramsay, “The Hermeneutics of Screwing Around,” 2010. <https://libraries.uh.edu/wp-content/uploads/Ramsay-The-Hermeneutics-of-Screwing-Around.pdf>.
- @witmore_text_2010

Online text for this class session

- Chapter 9.1, [The Variable Document Model](#)
- Chapter 9.2, Getting Data.
- Chapter 10.1, [Three Metrics](#)

agenda: [Class agenda](#)

Due Thu, Mar 31: Chapters 9, Exercises. (Not chapter 10, which is due in three weeks.)

Mon, Apr 04 Space as Data

Readings

- Klein and D’Ignazio, [The Numbers Don’t Speak for Themselves](#)
- @blevins_space_2014
- Anbinder et al, Networks and Opportunities: A Digital History of Ireland’s Great Famine Refugees in New York. *American Historical Review*, 2019. Be sure to spend a good amount of time in the online map as well as the printed article.
- Review the New York City Atlas.

- Pleiades project site. Browse in general: and read <https://pleiades.stoa.org/help/data-structure>, and think about types of uncertainty/ambiguity in geovisualization.

Online text for this class

- Space as Data (complete).
- If you are interested in mapping, browse the TOC for [Lovell et al, Geocomputation in R](#). Note that Lovell uses the tmap package for mapping while we stick to ggplot2 with the spatial geometries function geom_sf. If you really want to make–say–a zoomable map, you may want to explore tmap on your own.

The algorithmic toolkit for exploring humanities datasets.

Mon, Apr 11 Thinking statistically - Klein and D’Ignazio, Chapter 6. “<https://data-feminism.mitpress.mit.edu/pub/czq9dfs5/release/3>”

Due Mon, Apr 11: Identify data/datasets you’ll be working with for the rest of the class

Due Thu, Apr 14: Problems for thinking statistically and Dunning Log-Likelihood comparisons.

Due Fri, Apr 15: Go back to a problem you have already done, and use bootstrap sampling as an estimate of uncertainty on a problem set that you have already done.

Mon, Apr 18 Supervised Learning and Predictive Models

note: From this point on, the weekly readings and topics are about specific applications of algorithms to different types of problems. To this point, everything we’ve done has been foundational—from here on out, it’s more about specific applications that you *can* do if you want, but don’t necessarily need to.

Readings

- Klein and D’Ignazio, Chapter 7. “<https://data-feminism.mitpress.mit.edu/pub/czq9dfs5/release/3?readingCollection=ocd867ef>”
- Underwood, Predicting Genre.

online text: Classification

Mon, Apr 25 Clustering, topic modeling, and unsupervised approaches

Readings

- @allison_quantitative_2011 [Link](#)

agenda: [Class agenda](#)

Due Mon, Apr 25: due

Due Mon, Apr 25: text

Mon, May 02 The Embedding Strategy and representation learning.

description: Modern machine learning requires data, but it doesn't just look like an XML or TEI representation. Instead, a particular trick for turning items into strings of numbers—the **embedding strategy**—has emerged as the dominant ways for computers to represent information to themselves.

Readings

- Tukey, Friedman, and Fisherkeller: Introduction to Prim-9. [This is on YouTube, or I have a local copy.](#)
- Ryan Heuser, [Abstraction: A Literary History](#)
- Sandeep Soni, Lauren F. Klein, and Jacob Eisenstein. 'Abolitionist Networks: Modeling Language Change in Nineteenth-Century Activist Newspapers'. *Cultural Analytics* January 18, 2021.
- If you want to exist in this modern world, you should know something about deep learning. @lecun_deep_2015 is the single central text. It's hard to understand, but we'll take a stab. Come with questions, not understanding?
- PixPlot, Yale DH Lab: <http://dhlab.yale.edu/projects/pixplot/>

Assignment for this class

- Submit a draft piece about New York City or your own dataset for peer review.

Online text

- Vector Space Models, Principal Components Analysis, and similarity.

Mon, May 09 Going deep

Readings

- Lauren Tilton and Taylor Arnold, "Distant Viewing," *Digital Scholarship in the Humanities*, 2019. <https://www.distantviewing.org/pdf/distant-viewing.pdf>
- Melvin Wevers, Tomas Smits. "The visual digital turn: Using neural networks to study historical images". <https://doi.org/10.1093/lc/fqy085>. 18 January 2019
- Bengio, Lecun, and Hinton, "Deep Learning"
- Online Text, "Basic Neural Networks with Keras"

agenda: [Class agenda](#)

Agenda Notes

Notes for Mon, Feb 07

1. Rstudio installation and debug issues. What are packages, etc.
2. Any python holdouts?
3. It's to use Jupyter instead of RStudio if you prefer; but you will to install locally, because there are too many dependencies to re- download to Google Colab each time.

4. Drucker and Michel.
5. Polar opposites, so I find it helpful to find out which one you all find more amenable.
6. The question of where data comes from. Google Ngrams.
 - [02138](#)
 - [If](#)
 - [Four-letter words](#)
7. Issues of representation and the gift of data.
 - Wild ways of thinking about datasets.
8. Your datasets
9. A new section: Ontologies are formal languages for particular domains.
10. Categorical fields.
11. Introduction to Counting.

Notes for Mon, Mar 28

- “Collaboration?”
- Finding Texts—pushing mostly to Wednesday
 - Tokenization alternatives.
- “Discuss Shore and Ramsay—can we have fun?”
- “Discuss Witmore and free discussion of problems that can be approached as different sets of documents.”

Notes for Mon, Apr 25

- Talk about Allison and about Underwood. Distinguishing clustering and classification.
- Walk through of the vector space model concept in R.
- Pointing towards how to do classification in R.
- Walk through of basic clustering strategies.

Notes for Mon, May 09

- General check-in
- What can Deep Learning do?
- What can Deep Learning do for you?
- What about Word Embeddings?
- What about different forms of storytelling?

Those whose syllabi I have taken readings, ideas, and (in one case) a unit title from include Andrew Goldstone, Johanna Drucker, Lev Manovich, Jason Heppler, Lauren Klein, Maria Antoniak, and Ted Underwood.

I’ve leaned especially heavily in some places on Ryan Cordell’s 2017 offering of a version of this course at North-eastern University. Thanks also to the graduate students who took it in 2015 and 2019 at that institution.