Course Details

Title: Humanities Data Analysis

Location: Digital Humanities Summer Institute, University of Victoria, British Columbia

Dates: 10-14 June 2019

Instructors:

- Ryan Cordell (r.cordell@northeastern.edu)
- Greg Palermo (palermo.g@husky.neu.edu)

Course Description

The basic outlines of the course are sketched on the DHSI website:

This course introduces humanities researchers to the R programming language, with a focus on the analysis and visualization of tabular data (e.g. census records, bibliographic catalogs, etc.) using the Tidyverse suite of R packages. Before the course week, students will be asked to read a few touchstone essays wrestling with the peculiar qualities of humanistic data and the transformations of computational analysis. These essays will undergird our work during the week, helping us tie our practical work with R to broader questions about the nature of evidence in the humanities. The bulk of HDA will be devoted to demystifying the basic syntax of R (along with the operations of RStudio) and learning to import data (primarily as data frames); explore data through common but essential transformations; and visualize data using scatterplots, histograms, and related graphs. The course is a condensed version of this graduate seminar which Ryan Cordell teaches at Northeastern University in Boston.

Code of Conduct

Our code of conduct for this course borrows directly from the stellar model outlined by Northeastern's Feminist Coding Collective and can be considered as a complement to DHSI's Statement on Ethics and Inclusion. The Feminist Coding Collective's Code of Conduct and Community Guidelines are well worth consulting in full, but I have copied and lightly adapted those items most pertinent to the work we will do together during this week.

- **It's okay not to know**: Assume that no one inherently knows what we're learning. We all come to this class with different backgrounds and abilities; none of us (including the instructor) will know everything and that is okay! Encourage a space where it's okay to ask questions.
- **Be respectful**: Do not use harmful language or stereotypes that target people of all different gender, abilities, races, ages, ethnicities, languages, socioeconomic classes, body types, sexualities, and other aspects of identity.
- **Online spaces**: Respect each other in both physical and digital spaces.
- **Collaborative and inclusive interactions**: Avoid speaking over each other. Instead, we want to practice listening to each other and speaking with each other, not at each other.
- **Use "I" statements**: focusing on your own interpretation of a situation, rather than placing blame or critiquing someone else.
- **Harassment clause**: The following behaviors are considered harassment and unacceptable in this community (these are borrowed from the Django Code of Conduct):
  - Violent threats or language directed against another person.
  - Discriminatory jokes and language.
  - Posting sexually explicit or violent material.
  - Posting (or threatening to post) other people's personally identifying information ("doxing").
  - Personal insults, especially those using racist or sexist terms.
  - Unwelcome sexual attention.
  - Advocating for, or encouraging, any of the above behavior.
  - Repeated harassment of others. In general, if someone asks you to stop, then stop.

Prerequisites

This course presumes no prior knowledge of R or any other programming language. As much as possible, we have tried to build the lessons to presume no technical expertise beyond the installation of applications on your computer. You will be asked to install some software both prior to and during the week of classes. Once it's installed, we will expect you to be willing to experiment and develop new technical skills. Some of the tools we test you may find useful for your research program; some you will not. But we do expect you to try them with enthusiasm and an open mind.

On "Coding" in the Digital Humanities

In this course, you will think about coding and you will have to do some coding. 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 from time to time!

For at least the past decade, the question of whether humanists should code has been a vexed one in the digital humanities. In this course won't dwell on these debates, except to say that the answer to "should I learn to code?" is almost always, "what is your research question?" or "what kinds of questions do you want to teach students to answer?" This course will presume that your research or teaching questions involve either the analysis of data—in which case coding may be the only way to realize your specific vision—or building resources other scholars might want to analyze—in which case you should know the kinds of things sophisticated users will want to do with your tools, so you can make them work better. In other words, this course will not argue every humanist needs to learn to code, but it presumes *you*, specifically, might.

We certainly do not expect anyone to come out of this class a full-fledged developer, nor could we teach you how to become one in one week, however intensive the workshop. We'll be focusing on building skills less in full-fledged "programming" than in "scripting." That means instructing a computer in every stage of your work flow, and often involves tweaking code written by others rather than starting from scratch. We hope that by doing some scripting, you'll come to see that debates over learning to code brush over a lot of intermediate stages and flatten a range of skills into a simple binary (pun intended) achievement.

Even scripting will require you to use a programming language rather than a Graphical User Interface (GUI), which may be almost all the programs you've used before. Using a language takes more time at first, but has some distinct advantages over working in a GUI:

1. Your work is saved and more visible for inspection.
2. If you discover an error, you can correct it without losing the work done after the error was made.

3. If you want to amend your process (to analyze a hundred books instead of ten, for instance) but perform the same analysis, you can alter the code only slightly.
4. Perhaps most importantly, working in a programming language will help you better understand the step-by-step processes involved in computational analysis, including the computational analyses that underlie GUIs. Doing this work should help you be more aware of how computers think—or, better, how people think with computers. Even if you never touch a line of code after leaving this class, I hope the experience of it will make you a more thoughtful and critical user of all sorts of programs hereafter.

Course Software & Data

- Teaching datasets will be provided for students on the first day of class. Students will also have the opportunity to explore their own data, if they are so inclined, particularly in the final few days of class.
- We will use RStudio rather than RStudio Cloud in this course, as we had the chance to test all of our scripts yet with the latter, which is relatively new. You should install RStudio on your computer before the first day of class, as we'll start using it almost immediately on the first day.
- One of our exercises will require us to use the online dictionary Wordnik's API (application programming interface). Don't worry if that's not a meaningful term yet. The important thing is that you will need to sign up to get a Wordnik API key at https://developer.wordnik.com/. If you donate $5 (to a very worthy cause!) you can get the key within a day, but the free requests can take up to 7 days. Please do this ahead of our class so that you will have a key in time!

Why R?

This week we will work in the R programming language, developed for statistical computing. This has three main advantages for the sort of work that historians, literary scholars, and other humanists do:

1. R is easy to download and install through the program RStudio or, more recently, the cloud-based application RStudio Cloud. RStudio makes it easy to do scripting and test your results step by step. RStudio also offers a number of features that make it easy to explore data interactively.
2. R has lots of packages we can use for data analysis, such as dplyr, tidyr, and ggplot2. These are not core R libraries, but they are widely used and offer an intellectually coherent approach to data analysis and presentation. That means that even if you don't use these particular tools in the future, working with them should help you develop a 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. The ways of thinking you get from this work will serve you will in thinking about relational databases, structured data for archives, and a welter of other sources.
3. R is free: both "free as in beer," and "free as in speech," in the mantra of the Free Software Foundation. That means that R–like the rest of the peripheral tools we'll talk about—won't suddenly become inaccessible if you lose a university affiliation.
4. It's a pirate's favorite programming language (give it a second). Pirates are important historical and literary figures.

HDA Course Schedule:

Day 1: Monday, June 10

- Morning 1: DHSI Orientation
- Morning 2: Introduction to the class; getting started with RStudio
- Afternoon 1: The grammar of R
- Afternoon 2: Building a poetry bot

Day 2: Tuesday, June 11

- Morning 1: Data frames and tibbles
- Morning 2: Exercises
- Afternoon 1: Transforming tabular data 1
- Afternoon 2: Exercises

Day 3: Wednesday, June 12

- Morning 1: Transforming tabular data 2/Data visualization 1
- Morning 2: Exercises and/or work with student data
- Afternoon 1: Data visualization 2
- Afternoon 2: Exercises and/or work with student data

Day 4; Thursday, June 13

- Morning 1: Student choice (from: text analysis, classifying/clustering, mapping, topic modeling, vector space analysis)
- Morning 2: Exercises and/or work with student data
- Afternoon 1: Student choice (from: text analysis, classifying/clustering, mapping, topic modeling, vector space analysis)
- Afternoon 2: Exercises and/or work with student data

Day 5: Friday, June 14

- Morning 1: Student choice
- Morning 2: Course wrap up; final questions and certificates

Readings

Below are two lists of readings: a core set of articles we would like you to read, if at all possible, in preparation for our week together. We will devote some class time to discussing these specific texts and some of the larger topics and issues they point us toward. These articles are all provided in PDF form in this coursepack and are organized, roughly, chronologically.

The second list of penumbral readings is more suggestive: a bibliography for further research after the course, should your interests continue to unfold in these directions.

Core Readings:

- Hadley Wickham, "The Split-Apply-Combine Strategy for Data Analysis" (2011), https://www.jstatsoft.org/article/view/v040i01
- Cecily Carver, "Things I Wish Someone Had Told Me When I Was Learning How to Code" (22 November 2013), https://medium.freecodecamp.org/things-i-wish-someone-had-told-me-when-i-was-learning-how-to-code-565fc9dcb329
- Catherine D'Ignazio and Lauren F. Klein, "Feminist Data Visualization" (2015) http://www.kanarinka.com/wp-content/uploads/2015/07/IEEE_Feminist_Data_Visualization.pdf
- Ted Underwood, "Seven Ways Humanists Are Using Computers to Understand Text" (4 June 2015), https://tedunderwood.com/2015/06/04/seven-ways-humanists-are-using-computers-to-understand-text/
- Katie Rawson and Trevor Muñoz, "Against Cleaning" (7 July 2016), http://curatingmenus.org/articles/against-cleaning/
- Benjamin M. Schmidt, "Do Humanists Need to Understand Algorithms?" *Debates in Digital Humanities 2016*, http://dhdebates.gc.cuny.edu/debates/text/99
- Lincoln Mullen, "Isn't It Obvious?" (10 January 2018), https://lincolnmullen.com/blog/isnt-it-obvious/
- Moacir P. de Sá Pereira, "Representation Matters" (2018), http://xpmethod.plaintext.in/torn-apart/reflections/moacir_p_de_sa_pereira_2.html
- Jo Guldi, "Critical Search: A Procedure for Guided Reading in Large-Scale Textual Corpora," *Journal of Cultural Analytics (2018), https://doi.org/10.22148/16.030

Penumbral Readings:

The following are organized—not ideally, we acknowledge—in a rough chronological order. We have separated critical articles from textbooks and tutorials,

so you can find practical or theoretical help, as you require.

Critical Writing

- Jeannette M. Wing, "Computational Thinking," Communications of the ACM, 49.3 (Mar. 2006): pg. 33–35
- Danah Boyd and Kate Crawford, "Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon," *Information, Communication & Society* 15.5 (2012): pg. 662-679
- Ted Underwood, "Topic Modeling Made Just Simple Enough" (7 April 2012), https://tedunderwood.com/2012/04/07/topic-modeling-made-just-simple-enough/
- The topic modeling issue of the *Journal of Digital Humanities* 2.1 (Winter 2012), http://journalofdigitalhumanities.org/2-1/
- Lisa Gitelman, *"Raw Data" is an Oxymoron*, MIT Press (2013)
- Lauren F. Klein, "The Image of Absence: Archival Silence, Data Visualization, and James Hemings," *American Literature* 85.4 (2013)
- Tanya Clement, "Distant Listening or Playing Visualisations Pleasantly with the Eyes and Ears," *Digital Studies / Le champ numérique* 3.2 (26 July 2013), https://www.digitalstudies.org/ojs/index.php/digital_studies/article/view/228
- Brandon T. Locke, "Critical Data Literacy in the Humanities Classroom" (13 August 2013), http://brandontlocke.com/2018/08/13/critical-data-literacy-in-the-humanities-classroom.html
- Bethany Nowviskie, "Ludic Algorithms," *Pastplay: Teaching and Learning History with Technology*, University of Michigan Press (2014), http://quod.lib.umich.edu/d/dh/12544152.0001.001/1:5/--pastplay-teaching-and-learning-history-with-technology?g=dculture;rgn=div1;view=fulltext;xc=1#5.3
- Stephen Ramsay, "The Hermeneutics of Screwing Around; or What You Do with a Million Books," *Pastplay: Teaching and Learning History with Technology*, University of Michigan Press (2014), http://quod.lib.umich.edu/d/dh/12544152.0001.001/1:5/--pastplay-teaching-and-learning-history-with-technology?g=dculture;rgn=div1;view=fulltext;xc=1#5.1
- David Mimno, "Data Carpentry" (2015), http://www.mimno.org/articles/carpentry/
- Michael A. Gavin, "The Arithmetic of Concepts: a response to Peter de Bolla" (18 September 2015), http://modelingliteraryhistory.org/2015/09/18/the-arithmetic-of-concepts-a-response-to-peter-de-bolla/
- All the essays from *Debates in Digital Humanities 2016*'s Forum: Text Analysis At Scale section, perhaps especially:
  - Stephen Ramsay, "Humane Computation," http://dhdebates.gc.cuny.edu/debates/text/94
  - Tanya E. Clement, "The Ground Truth of DH Text Mining," http://dhdebates.gc.cuny.edu/debates/text/96

  - Lisa Marie Rhody, "Why I Dig: Feminist Approaches to Text Analysis," http://dhdebates.gc.cuny.edu/debates/text/97
  - Joanna Swafford, "Messy Data and Faulty Tools," http://dhdebates.gc.cuny.edu/debates/text/100
- Hoyt Long and Richard Jean So, "Literary Pattern Recognition: Modernism between Close Reading and Machine Learning," *Critical Inquiry* 42.2 (2016)
- Frederick W. Gibbs, "New Forms of History: Critiquing Data and Its Representations," *The American Historian* (February 2016), https://tah.oah.org/february-2016/new-forms-of-history-critiquing-data-and-its-representations/
- Ryan Heuser, "Word Vectors in the Eighteenth Century, Episode 1: Concepts" (14 April 2016), http://ryanheuser.org/word-vectors-1/, and "Episode 2: Methods" (1 June 2016), http://ryanheuser.org/word-vectors-2/
- Sarah Allison, "Other People's Data: Humanities Edition," *CA: the Journal of Cultural Analytics* (8 December 2016), http://culturalanalytics.org/2016/12/other-peoples-data-humanities-edition/
- Andrew Piper, "Fictionality," *CA: the Journal of Cultural Analytics* (20 December 2016), http://culturalanalytics.org/2016/12/fictionality/
- Annette Vee, *Coding Literacy: How Computer Programming Is Changing Writing*, The MIT Press (2017).
- Andrew Goldstone, "The Doxa of Reading," *PMLA* 132.3 (2017)
- Richard Jean So, "All Models Are Wrong," *PMLA* 132.3 (2017)
- Johanna Drucker, "Non-representational approaches to modeling interpretation in a graphical environment," *Digital Scholarship in the Humanities* 33.2 (2017): pg. 248-263.
- Candice Lanius and Gaines S. Hubbell. "The New Data: Argumentation amidst, on, with, and in Data," *Theorizing Digital Rhetoric*, ed. Aaron Hess and Amber Davisson, 1st edition, Routledge (2017): pg. 126–39.
- L. Aull, *First-Year University Writing: A Corpus-Based Study with Implications for Pedagogy*, 1st ed., Palgrave Macmillan (2015)
- Elyse Graham, "Introduction: Data Visualisation and the Humanities," *English Studies* 98.5 (4 July 2017), https://doi.org/10.1080/0013838X.2017.1332021
- Laura K. Nelson, "Computational Grounded Theory: A Methodological Framework," *Sociological Methods & Research* (21 November 2017), https://doi.org/10.1177/0049124117729703
- Clifford Lynch, "Stewardship in the 'Age of Algorithms,' " *First Monday* 22.12 (4 December 2017), http://firstmonday.org/ojs/index.php/fm/article/view/8097
- Katherine Bode, *A World of Fiction: Digital Collections and the Future of Literary History*, Univ. of Michigan Press (2018)
- Andrew Piper, *Enumerations: Data and Literary Study*, Univ. of Chicago Press (2018)
- Journal of Writing Analytics, vol 2 (2018) https://journals.colostate.edu/index.php/analytics/issue/view/13/showToc

- Lauren Klein, "Distant Reading After Moretti" (10 January 2018), http://lklein.com/2018/01/distant-reading-after-moretti/
- Ted Underwood, David Bamman, and Sabrina Lee, "The Transformation of Gender in English-Language Fiction," *CA: the Journal of Cultural Analytics* (13 February 2018), http://culturalanalytics.org/2018/02/the-transformation-of-gender-in-english-language-fiction/
- Richard Jean So, Hoyt Long, and Yuancheng Zhu, "Race, Writing, and Computation: Racial Difference and the US Novel, 1880-2000," *CA: Journal of Cultural Analytics* (11 January 2019), http://culturalanalytics.org/2019/01/race-writing-and-computation-racial-difference-and-the-us-novel-1880-2000/
- Ted Underwood, *Distant Horizons: Digital Evidence and Literary Change*, Univ. of Chicago Press (2019)
- Catherine D'Ignazio and Lauren Klein, *Data Feminism*, MIT Open Press (In Open Review, 2019), https://bookbook.pubpub.org/data-feminism

Textbooks and Tutorials

- Garrett Grolemund and Hadley Wickham, *R for Data Science* (2017), https://r4ds.had.co.nz/index.html
- Scott Weingart, "Teaching Yourself to Code in DH" (26 February 2017), http://scottbot.net/teaching-yourself-to-code-in-dh/
- Taryn Dewar, "R Basics with Tabular Data," Programming Historian (5 September 2016), https://programminghistorian.org/en/lessons/r-basics-with-tabular-data
- Taylor Arnold and Lauren Tilton, "Basic Text Processing in R," Programming Historian (27 March 2017), https://programminghistorian.org/en/lessons/basic-text-processing-in-r
- Nabeel Siddiqui, "Data Wrangling and Management in R," Programming Historian (31 July 2017), https://programminghistorian.org/en/lessons/data_wrangling_and_management_in_R
- Ryan Deschamps, "Correspondence Analysis for Historical Research with R," Programming Historian (13 September 2017), https://programminghistorian.org/en/lessons/correspondence-analysis-in-R
- Jeff Blackadar, "Introduction to MySQL with R," Programming Historian (3 May 2018), https://programminghistorian.org/en/lessons/getting-started-with-mysql-using-r
- Alex Brey, "Temporal Network Analysis with R," Programming Historian, (4 November 2018), https://programminghistorian.org/en/lessons/temporal-network-analysis-with-r

Journal of Statistical Software

April 2011, Volume 40, Issue 1. *http://www.jstatsoft.org/*

The Split-Apply-Combine Strategy for Data Analysis

Hadley Wickham
Rice University

Abstract

Many data analysis problems involve the application of a split-apply-combine strategy, where you break up a big problem into manageable pieces, operate on each piece independently and then put all the pieces back together. This insight gives rise to a new R package that allows you to smoothly apply this strategy, without having to worry about the type of structure in which your data is stored.

The paper includes two case studies showing how these insights make it easier to work with batting records for veteran baseball players and a large 3d array of spatio-temporal ozone measurements.

Keywords: R, apply, split, data analysis.

1. Introduction

What do we do when we analyze data? What are common actions and what are common mistakes? Given the importance of this activity in statistics, there is remarkably little research on how data analysis happens. This paper attempts to remedy a very small part of that lack by describing one common data analysis pattern: Split-apply-combine. You see the split-apply-combine strategy whenever you break up a big problem into manageable pieces, operate on each piece independently and then put all the pieces back together. This crops up in all stages of an analysis:

- During data preparation, when performing group-wise ranking, standardization, or normalization, or in general when creating new variables that are most easily calculated on a per-group basis.
- When creating summaries for display or analysis, for example, when calculating marginal means, or conditioning a table of counts by dividing out group sums.

- During modeling, when fitting separate models to each panel of panel data. These models may be interesting in their own right, or used to inform the construction of a more sophisticated hierarchical model.

The split-apply-combine strategy is similar to the map-reduce strategy for processing large data, recently popularized by Google. In map-reduce, the map step corresponds to split and apply, and reduce corresponds to combine, although the types of reductions are much richer than those performed for data analysis. Map-reduce is designed for a highly parallel environment, where work is done by hundreds or thousands of independent computers, and for a wider range of data processing needs than just data analysis.

Just recognizing the split-apply-combine strategy when it occurs is useful, because it allows you to see the similarly between problems that previously might have appeared unconnected. This helps suggest appropriate tools and frees up mental effort for the aspects of the problem that are truly unique. This strategy can be used with many existing tools: APL's array operators (Friendly and Fox 1994), **Excel**'s pivot tables, the SQL `group by` operator, and the `by` argument to many SAS procedures. However, the strategy is even more useful when used with software specifically developed to support it; matching the conceptual and computational tools reduces cognitive impedance. This paper describes one implementation of the strategy in R (R Development Core Team 2010), the **plyr** package.

In general, **plyr** provides a replacement for loops for a large set of practical problems, and abstracts away from the details of the underlying data structure. An alternative to loops is not required because loops are slow (in most cases the loop overhead is small compared to the time required to perform the operation), but because they do not clearly express intent, as important details are mixed in with unimportant book-keeping code. The tools of **plyr** aim to eliminate this extra code and illuminate the key components of the computation.

Note that **plyr** makes the strong assumption that each piece of data will be processed only once and independently of all other pieces. This means that you can not use these tools when each iteration requires overlapping data (like a running mean), or it depends on the previous iteration (like in a dynamic simulation). Loops are still most appropriate for these tasks. If more speed is required, you can either recode the loops in a lower-level language (like C or Fortran) or solve the recurrence relation to find a closed form solution.

To motivate the development and use of **plyr**, Section 2 compares code that uses **plyr** functions with code that uses tools available in base R. Section 3 introduces the **plyr** family of tools, describes the three types of input and four types of output, and details the way in which input is split up and output is combined back together. The **plyr** package also provides a number of helper functions for error recovery, splatting, column-wise processing, and reporting progress, described in Section 4. Section 5 discusses the general strategy that these functions support, including two case studies that explore the performance of veteran baseball players, and the spatial-temporal variation of ozone. Finally, Section 6 maps existing R functions to their **plyr** counterparts and lists related packages. Section 7 describes future plans for the package.

This paper describes version 1.0 of **plyr**, which requires R 2.10.0 or later and has no run-time dependencies. The **plyr** package is available on the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=plyr`. Information about the latest version of the package can be found online at `http://had.co.nz/plyr`. To install it from within R, run `install.packages("plyr")`. The code used in this paper is available online in the supplemental materials.

Notation. *Array* includes the special cases of vectors (1d arrays) and matrices (2d arrays). Arrays can be made out of any *atomic* vector: Logical, character, integer, or numeric. A *list-array* is a non-atomic array (a list with dimensions), which can contain any type of data structure, such as a linear model or 2d kernel density estimate. *Dimension labels* refer to `dimnames()` for arrays; `rownames()` and `colnames()` for matrices and data frames; and `names()` for atomic vectors and lists.

2. Motivation

How does the explicit specification of this strategy help? What are the advantages of **plyr** over `for` loops or the built-in apply functions? This section compares **plyr** code to base R code with a teaser from Section 5.2, where we remove seasonal effects from 6 years of monthly satellite measurements, taken on a 24×24 grid. The 41 472 measurements are stored in a $24 \times 24 \times 72$ array. A single location (`ozone[x, y, ]`) is a vector of 72 values (6 years $\times 12$ months) .

We can crudely deseasonalize a location by looking at the residuals from a robust linear model:

```
R> one <- ozone[1, 1, ]
R> month <- ordered(rep(1:12, length = 72))
R> model <- rlm(one ~ month - 1)
R> deseas <- resid(model)
R> deseasf <- function(value) rlm(value ~ month - 1)
```

The challenge is to apply this function to each location, reassembling the output into the same form as the input, a 3d array. It would also be nice to keep the models in a 2d list-array, so we can reference a local model (`model[[1, 1]]`) in a similar way to referencing a local time series (`ozone[1, 1, ]`); keeping data-structures consistent reduces cognitive effort. In base R, we can tackle this problem with nested loops, or with the apply family of functions, as shown by Table 1.

The main disadvantage of the loops is that there is lot of book-keeping code: The size of the array is hard coded in multiple places and we need to create the output structures before filling them with data. The apply functions, `apply()` and `lapply()`, simplify the task, but there is not a straightforward way to go from the 2d array of models to the 3d array of residuals. In **plyr**, the code is much shorter because these details are taken care of:

```
R> models <- aaply(ozone, 1:2, deseasf)
R> deseas <- aaply(models, 1:2, resid)
```

You may be wondering what these function names mean. All **plyr** functions have a concise but informative naming scheme: The first and second characters describe the input and output data types. The input determines how the data should be split, and the output how it should be combined. Both of the functions used above input and output an *a*rray. Other data types are *l*ists and *d*ata frames. Because **plyr** caters for every combination of input and output data types in a consistent way, it is easy to use the data structure that feels most natural for a given problem.

For example, instead of storing the ozone data in a 3d array, we could also store it in a data frame. This type of format is more common if the data is ragged, irregular, or incomplete;

For loops

```
models <- as.list(rep(NA, 24 * 24))
dim(models) <- c(24, 24)

deseas <- array(NA, c(24, 24, 72))
dimnames(deseas) <- dimnames(ozone)

for (i in seq_len(24)) {
  for(j in seq_len(24)) {
    mod <- deseasf(ozone[i, j, ])

    models[[i, j]] <- mod
    deseas[i, j, ] <- resid(mod)
  }
}
```

Apply functions

```
models <- apply(ozone, 1:2, deseasf)
resids_list <- lapply(models, resid)

resids <- unlist(resids_list)
dim(resids) <- c(72, 24, 24)
deseas <- aperm(resids, c(2, 3, 1))
dimnames(deseas) <- dimnames(ozone)
```

Table 1: Compare of for loops and apply functions

if we did not have measurements at every possible location for every possible time point. Imagine the data frame is called `ozonedf` and has columns `lat`, `long`, `time`, `month`, and `value`. To repeat the deseasonalization task with this new data format, we first need to tweak our workhorse method to take a data frame as input:

```
R> deseasf_df <- function(df) {
+    rlm(value ~ month - 1, data = df)
+  }
```

Because the data could be ragged, it is difficult to use a `for` loop and we will use the base R functions `split()`, `lapply()` and `mapply()` to complete the task. Here the split-apply-combine strategy maps closely to built-in R functions: We split with `split()`, apply with `lapply()` and then combine the pieces into a single data frame with `rbind()`.

```
R> pieces <- split(ozonedf, list(ozonedf$lat, ozonedf$long))
R> models <- lapply(pieces, deseasf_df)
R> results <- mapply(function(model, df) {
+    cbind(df[rep(1, 72), c("lat", "long")], resid(model))
+  }, models, pieces)
R> deseasdf <- do.call("rbind", results)
```

Most of the complication here is in attaching appropriate labels to the data. The type of labels needed depends on the output data structure, e.g., for arrays, dimnames are labels, while for data frames, values in additional columns are the labels. Here, we needed to use `mapply()` to match the models to their source data in order to extract informative labels. **plyr** takes care of adding the appropriate labels, so it only takes two lines:

```
R> models <- dlply(ozonedf, .(lat, long), deseasf_df)
R> deseas <- ldply(models, resid)
```

`dlply` takes a data frame and returns a list, and `ldply` does the opposite: It takes a list and returns a data frame. Compare this code to the code needed when the data was stored in an array.

The following section describes the **plyr** functions in more detail. If your interest has been whetted by this example, you might want to skip ahead to Section 5.2 to learn more about this example and see some plots of the data before and after removing the seasonal effects.

3. Usage

Table 2 lists the basic set of **plyr** functions. Each function is named according to the type of input it accepts and the type of output it produces: `a` = array, `d` = data frame, `l` = list, and `_` means the output is discarded. The input type determines how the big data structure is broken apart into small pieces, described in Section 3.1; and the output type determines how the pieces are joined back together again, described in Section 3.2.

The effects of the input and outputs types are orthogonal, so instead of having to learn all 12 functions individually, it is sufficient to learn the three types of input and the four types of output. For this reason, we use the notation `d*ply` for functions with common input, a complete row of Table 2, and `*dply` for functions with common output, a column of Table 2.

The functions have either two or three main arguments, depending on the type of input:

- `a*ply(.data, .margins, .fun, ..., .progress = "none")`
- `d*ply(.data, .variables, .fun, ..., .progress = "none")`
- `l*ply(.data, .fun, ..., .progress = "none")`

The first argument is the `.data` which will be split up, processed and recombined. The second argument, `.variables` or `.margins`, describes how to split up the input into pieces. The third argument, `.fun`, is the processing function, and is applied to each piece in turn. All further arguments are passed on to the processing function. If you omit `.fun` the individual pieces will not be modified, but the entire data structure will be converted from one type to another. The `.progress` argument controls display of a progress bar, and is described at the end of Section 4.

Note that all arguments start with ".". This prevents name clashes with the arguments of the processing function, and helps to visually delineate arguments that control the repetition

Input \ *Output*	Array	Data frame	List	Discarded
Array	`aaply`	`adply`	`alply`	`a_ply`
Data frame	`daply`	`ddply`	`dlply`	`d_ply`
List	`laply`	`ldply`	`llply`	`l_ply`

Table 2: The 12 key functions of **plyr**. Arrays include matrices and vectors as special cases.

from arguments that control the individual steps. Some functions in base R use all uppercase argument names for this purpose, but I think this method is easier to type and read.

3.1. Input

Each type of input has different rules for how to split it up, and these rules are described in detail in the following sections. In short:

- Arrays are sliced by dimension in to lower-d pieces: `a*ply()`.
- Data frames are subsetted by combinations of variables: `d*ply()`.
- Each element in a list is a piece: `l*ply()`.

Technical note. The way the input can be split up is determined not by the type of the data structure, but the methods that it responds to. An object split up by `a*ply()` must respond to `dim()` and accept multidimensional indexing; by `d*ply()`, must work with `split()` and be coercible to a list; by list, must work with `length()` and `[[`. This means that data frames can be passed to `a*ply()`, where they are treated like 2d matrices, and to `l*ply()` where they are treated as a list of vectors (the variables).

Input: Array (`a*ply`)

The `.margins` argument of `a*ply` describes which dimensions to slice along. If you are familiar with `apply`, `a*ply` works the same way. There are four possible ways to do this for the 2d case. Figure 1 illustrates three of them:

- `.margins = 1`: Slice up into rows.
- `.margins = 2`: Slice up into columns.
- `.margins = c(1,2)`: Slice up into individual cells.

The fourth way is to not split up the matrix at all, and corresponds to `.margins = c()`. However, there is not much point in using **plyr** to do this!

The 3d case is a little more complicated. We have three possible 2d slices, three 1d, and one 0d. These are shown in Figure 2. Note how the pieces of the 1d slices correspond to the intersection of the 2d slices. The margins argument works correspondingly for higher dimensions, with an combinatorial explosion in the number of possible ways to slice up the array.

Special case: `m*ply` A special case of operating on arrays corresponds to the `mapply` function of base R. `mapply` seems rather different at first glance: It accepts multiple inputs as separate arguments, compared to `a*ply` which takes a single array argument. However, the separate arguments to `mapply()` must have the same length, so conceptually it is the same underlying data structure. The **plyr** equivalents are named `maply`, `mdply`, `mlply` and `m_ply`.

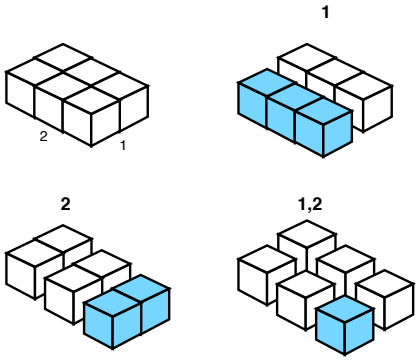


Figure 1: The three ways to split up a 2d matrix, labelled above by the dimensions that they slice. Original matrix shown at top left, with dimensions labelled. A single piece under each splitting scheme is colored blue.

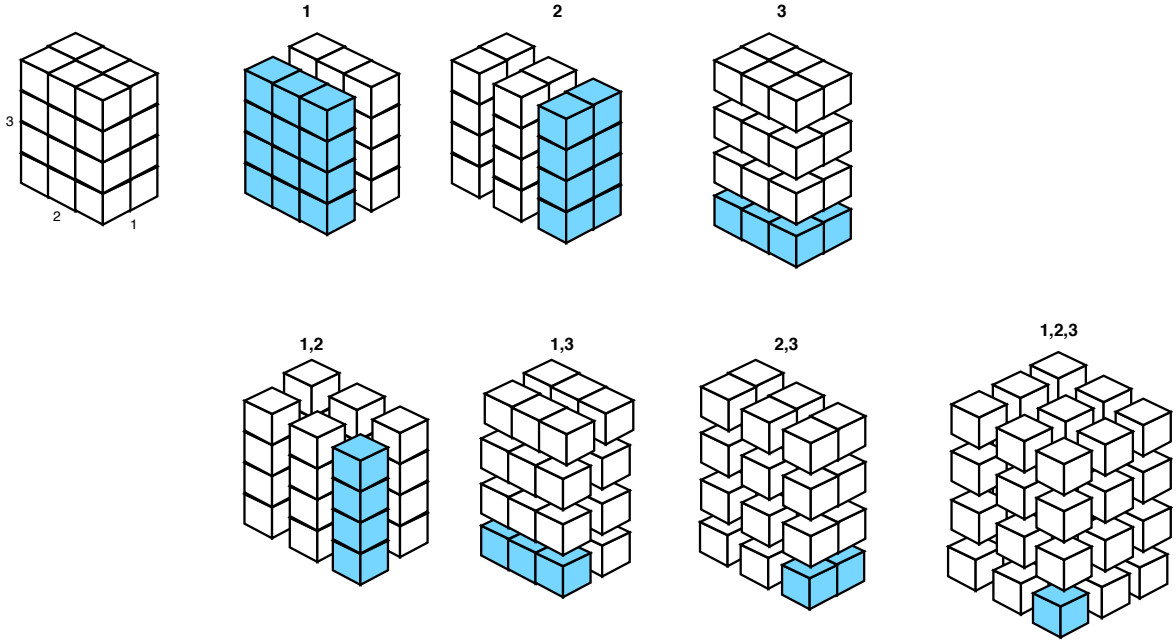


Figure 2: The seven ways to split up a 3d array, labelled above by the dimensions that they slice up. Original array shown at top left, with dimensions labelled. Blue indicates a single piece of the output.

`m*ply()` takes a matrix, list-array, or data frame, splits it up by rows and calls the processing function supplying each piece as its parameters. Figure 3 shows how you might use this to draw random numbers from normal distributions with varying parameters.

Input: Data frame (`d*ply`)

When operating on a data frame, you usually want to split it up into groups based on combinations of variables in the data set. For `d*ply` you specify which variables (or functions of variables) to use. These variables are specified in a special way to highlight that they are

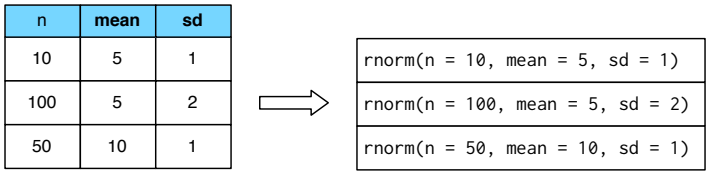


Figure 3: Using `m*ply` with `rnorm()`, `m*ply(data, rnorm)`. The function is called once for each row, with arguments given by the columns. Arguments are matched by position, or name, if present.

computed first from the data frame, then the global environment (in which case it is your responsibility to ensure that their length is equal to the number of rows in the data frame).

- `.(var1)` will split the data frame into groups defined by the value of the `var1` variable. If you use multiple variables, `.(a, b, c)`, the groups will be formed by the interaction of the variables, and output will be labelled with all three variables. For array output, there will be three dimensions whose dimension names will be the values of a, b, and c in the input data frame; for data frame output there will be three extra columns with the values of a, b, and c; and for list output, the element names will be the values of a, b, and c appended together separated by periods, along with a `split_labels` attribute which contains the splits as a data frame.
- You can also use functions of variables: `.(round(a))`, `.(a * b)`. When outputting to a data frame, ugly names (produced by `make.names()`) may result, but you can override them by specifying names in the call: `.(product = a * b)`.

Alternatively, you can use two more familiar ways of describing the splits:

- As a character vector of column names: `c("var1", "var2")`.
- With a (one-sided) formula `~ var1 + var2`.

Figure 4 shows two examples of splitting up a simple data frame. Splitting up data frames is easier to understand (and to draw!) than splitting up arrays, because they are only 2 dimensional.

Input: List (`l*ply`)

Lists are the simplest type of input to deal with because they are already naturally divided into pieces: The elements of the list. For this reason, the `l*ply` functions do not need an argument that describes how to break up the data structure. Using `l*ply` is equivalent to using `a*ply` on a 1d array. `l*ply` can also be used with atomic vectors.

Special case: `r*ply` A special case of operating on lists corresponds to `replicate()` in base R, and is useful for drawing distributions of random numbers. This is a little bit different to the other **plyr** methods. Instead of the `.data` argument, it has `.n`, the number of replications to run, and instead of a function it accepts a expression, which is evaluated afresh for each replication.

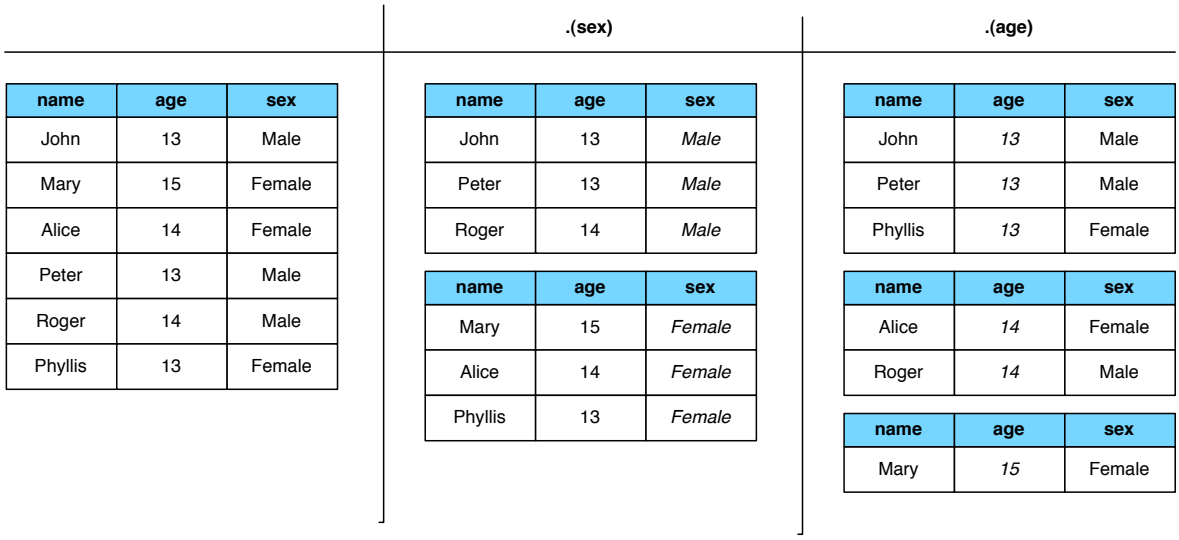


Figure 4: Two examples of splitting up a data frame by variables. If the data frame was split up by both sex and age, there would only be one subset with more than one row: 13-year-old males.

Output	Processing function restrictions	Null output
`*aply`	atomic array, or list	`vector()`
`*dply` frame	data frame, or atomic vector	`data.frame()`
`*lply`	none	`list()`
`*_ply`	none	—

Table 3: Summary of processing function restrictions and null output values for all output types. Explained in more detail in each output section.

3.2. Output

The output type defines how the pieces will be joined back together and how they will be labelled. The labels are particularly important as they allow matching up of input and output.

The input and output types are the same, except there is an additional output data type, `_`, which discards the output. This is useful for functions like `plot()` and `write.table()` that are called only for their side effects, not their return value.

The output type also places some restrictions on what type of results the processing function should return. Generally, the processing function should return the same type of data as the eventual output, (i.e., vectors, matrices and arrays for `*aply` and data frames for `*dply`) but some other formats are accepted for convenience and are described in Table 3. These are explained in more detail in the individual output type sections.

Output: Array (`*aply`)

With array output the shape of the output array is determined by the input splits and the dimensionality of each individual result. Figures 5 and 6 illustrate this pictorially for simple

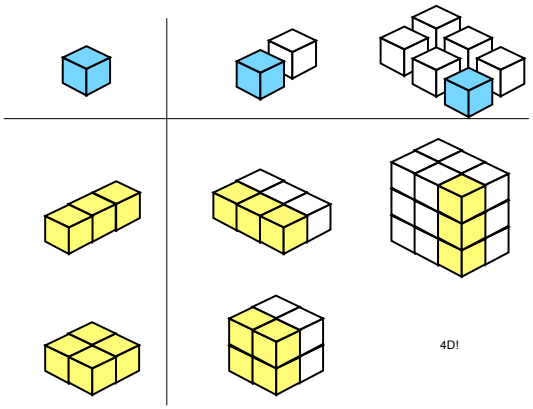


Figure 5: Results from outputs of various dimensionality from a *single* value, shown top left. Columns indicate input: (left) a vector of length two, and (right) a 3×2 matrix. Rows indicate the shape of a single processed piece: (top) a vector of length 3, (bottom) a 2×2 matrix. Extra dimensions are added perpendicular to existing ones. The array in the bottom-right cell is 4d and so is not shown.

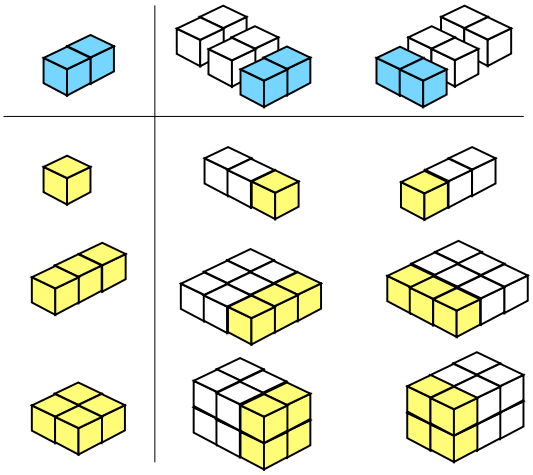


Figure 6: Results from outputs of various dimensionality from a *1d vector*, shown top left. Columns indicate input: (left) a 2×3 matrix split by rows and (right) and 3×2 matrix split by columns. Rows indicate the shape of a single processed piece: (top) a single value, (middle) a vector of length 3, and (bottom) a 2×2 matrix.

1d and 2d cases, and the following code shows another way to think about it.

```
R> x <- array(1:24, 2:4)
R> shape <- function(x) if (is.vector(x)) length(x) else dim(x)
R> shape(x)

[1] 2 3 4

R> shape(aaply(x, 2, function(y) 0))
```

```
[1] 3

R> shape(aaply(x, 2, function(y) rep(1, 5)))

[1] 3 5

R> shape(aaply(x, 2, function(y) matrix(0, nrow = 5, ncol = 6)))

[1] 3 5 6

R> shape(aaply(x, 1, function(y) matrix(0, nrow = 5, ncol = 6)))

[1] 2 5 6
```

For array input, the pieces contribute to the output array in the expected way. The dimension labels of the output array will be the same as the dimension labels of the splits (i.e., the dimensions indexed by .margin in the input array.) List input is treated like a 1d array. For data frame input, the output array gets a dimension for each variable in the split, labelled by values of those variables.

The processing function should return an object of the same type and dimensionality each time it is called. This can be an atomic array (e.g., numeric, character, logical), or a list. If the object returned by the processing function is an array, its dimensions are included in the output array after the split dimensions. If it is a list, the output will be a list-array (i.e., a list with dimensions based on the split, and elements of the list are the objects returned by the processing function). If there are no results, `*aply` will return a logical vector of length 0.

All `*aply` functions have a `drop.` argument. When this is true, the default, any dimensions of length one will be dropped. This is useful because in R, a vector of length three is not equivalent to a 3×1 matrix or a $3 \times 1 \times 1$ array.

Output: Data frame (`*dply`*)*

When the output is a data frame, it will the results as well as additional label columns. These columns make it possible to merge the old and new data if required. If the input was a data frame, there will be a column for each splitting variable; if a list, a column for list names (if present); if an array, a column for each splitting dimension. Figure 7 illustrates this for data frame input.

The processing functions should either return a data.frame, or an atomic vector of fixed length, which will be interpreted as a row of a data frame. In contrast to `*aply`, the shape of the results can vary: The piece-wise results are combined together with `rbind.fill()`, so that any piece missing columns used in another piece will have those columns filled in with missing values. If there are no results, `*dply` will return an empty data frame. **plyr** provides an `as.data.frame` method for functions which can be handy: `as.data.frame(mean)` will create a new function which outputs a data frame.

Output: List (`*lply`*)*

This is the simplest output format, where each processed piece is joined together in a list. The list also stores the labels associated with each piece, so that if you use `ldply` or `laply`

.(sex)

sex	value
Male	3
Female	3

.(age)

age	value
13	3
14	2
15	1

.(sex, age)

sex	age	value
Male	13	2
Male	14	1
Female	13	1
Female	14	1
Female	15	1

Figure 7: Illustrating the output from using `ddply()` on the example from Figure 4 with `nrow()`. Splitting variables shown above each example. Note how the extra labeling columns are added so that you can identify to which subset the results apply.

to further process the list the labels will appear as if you had used `aaply`, `adply`, `daply` or `ddply` directly. `llply` is convenient for calculating complex objects once (e.g., models), from which you later extract pieces of interest into arrays and data frames.

There are no restrictions on the output of the processing function. If there are no results, `*lply` will return a list of length 0.

Output: Discarded (`*_ply`)

Sometimes it is convenient to operate on a list purely for the side effects, e.g., plots, caching, and output to screen/file. In this case `*_ply` is a little more efficient than abandoning the output of `*lply` because it does not store the intermediate results.

The `*_ply` functions have one additional argument, `.print`, which controls whether or not each result should be printed. This is useful when working with **lattice** (Sarkar 2008) or **ggplot2** (Wickham 2010) graphics.

4. Helpers

The **plyr** package also provides a number of helper function which take a function (or functions) as input and return a new function as output.

- `splat()` converts a function that takes multiple arguments to one that takes a list as its single argument. This is useful when you want a function to operate on a data frame, without manually pulling it apart. In this case, the column names of the data frame will match the argument names of the function. For example, compare the following two `ddply` calls, one with, and one without `spat`:

```
R> hp_per_cyl <- function(hp, cyl, ...) hp / cyl
R> splat(hp_per_cyl)(mtcars[1,])
R> splat(hp_per_cyl)(mtcars)
R> ddply(mtcars, .(round(wt)),
+    function(df) mean_hp_per_cyl(df$hp, df$cyl))
R> ddply(mtcars, .(round(wt)), splat(mean_hp_per_cyl))
```

Generally, splatted functions should have `...` as an argument, as this will consume unused variables without raising a error. For more information on how splat works, see `do.call`.

`m*ply` uses `splat()` to call the processing function given to it: `m*ply(a_matrix, FUN)` is equivalent to `a*ply(a_matrix, 1, splat(FUN))`.

- `each()` takes a list of functions and produces a function that runs each function on the inputs and returns a named vector of outputs. For example, `each(min, max)` is short hand for `function(x) c(min = min(x), max = max(x))`. Using `each()` with a single function is useful if you want a named vector as output.

- `colwise()` converts a function that works on vectors, to one that operates column-wise on a data frame, returning a data frame. For example, `colwise(median)` is a function that computes the median of each column of a data.frame.

  The optional `.if` argument specializes the function to only run on certain types of vector, e.g., `.if = is.factor` or `.if = is.numeric`. These two restrictions are provided in the pre-made `calcolwise` and `numcolwise`. Alternatively, you can provide a vector of column names, and `colwise()` only operate on those columns.

- `failwith()` sets a default value to return if the function throws an error. For example, `failwith(NA, f)` will return an `NA` whenever `f` throws an error.

  The optional `quiet` argument suppresses any notification of the error when `TRUE`.

- Given a function, `as.data.frame.function()` creates a new function which coerces the output of the input function to a data frame. This is useful when you are using `*dply()` and the default column-wise output is not what you want.

There is one additional helper function analogous to the base function 'transform', but instead of returning the original data frame with modified or new columns added to it, it returns just the modified or new columns in a new data frame. Because this is very useful for creating group-wise summaries, it is called `summarize()` (or `summarize()`).

Each **plyr** function also has a `.progress` argument which allows you to monitor the progress of long running operations. There are four different progress bars:

- `"none"`, the default. No progress bar is displayed.

- `"text"` provides a textual progress bar.

- `"win"` and `"tk"` provide graphical progress bars for Windows and systems with the **tcltk** package (Dalgaard 2001), including the base distribution of R for Mac and Linux.

This is useful because it allows you to gauge how long a task will take and so you know if you need to leave your computer on overnight, or try a different approach entirely. Psychologically, adding a progress bar also makes it feel like it takes much less time. The progress bars assume that processing each piece takes the same amount of time, so may not be 100% accurate.

5. Strategy

Having learned the basic structure and operation of the **plyr** family of functions, you will now see some examples of using them in practice. The following two case studies explore two data sets: A data frame of batting records from long-term baseball players, and a 3d array recording ozone measurements that vary over space and time. Neither of these data studies do more than scratch the surface of possible analyses, but do show of a number of different ways to use **plyr**.

Both cases follow a similar process:

1. Extract a subset of the data for which it is easy to solve the problem.
2. Solve the problem by hand, checking results as you go.
3. Write a function that encapsulates the solution.
4. Use the appropriate **plyr** function to split up the original data, apply the function to each piece and join the pieces back together.

The code shown in this paper is necessarily abbreviated. The data sets are large, and often only small subsets of the data are shown. The code focuses on data manipulation, and much of the graphics code is omitted. You are encouraged to experiment with the full code yourself, available as a supplement to this paper.

All plots are created using **ggplot2** (Wickham 2010), but the process is very similar regardless of the graphics package used.

5.1. Case study: Baseball

The `baseball` data set contains the batting records for all professional US players with 15 or more years of data. In this example we will focus on four of the variables in the data: `id`, which identifies the player, `year` the year of the record; `rbi`, runs batted in, the number of runs that the player made in the season; and `ab`, at bat or the number of times the player faced a pitcher. The other variables are described in `?baseball`.

What we will explore is the performance of a batter over his career. To get started, we need to calculate the "career year", i.e. the number of years since the player started playing. This is easy to do if we have a single player:

```
R> baberuth <- subset(baseball, id == "ruthba01")
R> baberuth <- transform(baberuth, cyear = year - min(year) + 1)
```

To do this for all players, we do not need to write our own function, because we can apply `transform()` to each piece:

```
R> baseball <- ddply(baseball, .(id), transform,
+    cyear = year - min(year) + 1)
```

To summarize the pattern across all players, we first need to figure out what the common patterns are. A time series plot of rbi/ab, runs per bat, is a good place to start. We do this

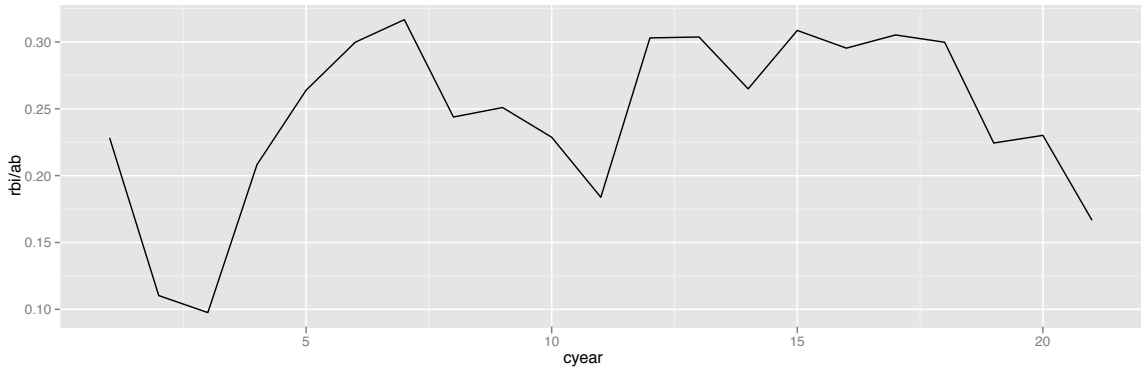


Figure 8: Runs per bat for Babe Ruth.

for Babe Ruth, as shown in Figure 8, then write a function to do it for any player (taking care to ensure common scale limits) and then use `d_ply` to save a plot for every player to a pdf. We use two tricks here: `reorder` to sort the players in order of average rbi / ab, and `failwith` to ensure that even if a single plot does not work we will still get output for the others. We also restrict the data to focus only on records where `ab` is greater than 25: This prevents problems with a small number on the denominator.

```
R> baseball <- subset(baseball, ab >= 25)
R> xlim <- range(baseball$cyear, na.rm=TRUE)
R> ylim <- range(baseball$rbi / baseball$ab, na.rm=TRUE)
R> plotpattern <- function(df) {
+    qplot(cyear, rbi / ab, data = df, geom = "line",
+      xlim = xlim, ylim = ylim)
+  }
R> pdf("paths.pdf", width = 8, height = 4)
R> d_ply(baseball, .(reorder(id, rbi / ab)), failwith(NA, plotpattern),
+    .print = TRUE)
R> dev.off()
```

Flicking through the 1145 plots reveals few common patterns, although many players do seem to have a roughly linear trend with quite a bit of noise. We will start by fitting a linear model to each player and then exploring the results. This time we will skip doing it by hand and go directly to the function. (Not recommended in practice!)

```
R> model <- function(df) {
+   lm(rbi / ab ~ cyear, data = df)
+  }
R> model(baberuth)

Call:
lm(formula = rbi/ab ~ cyear, data = df)

Coefficients:
```

id	intercept	slope	rsquare
aaronha01	0.18	0.00	0.00
abernte02	0.00		0.00
adairje01	0.09	−0.00	0.01
adamsba01	0.06	0.00	0.03
adamsbo03	0.09	−0.00	0.11
adcocjo01	0.15	0.00	0.23

Table 4: The first few rows of the `bcoefs` data frame. Note that the player ids from the original data have been preserved.

```
(Intercept)        cyear
    0.20797      0.00332
```

```
R> bmodels <- dlply(baseball, .(id), model)
```

Now we have a list of 1145 models, one for each player. To do something interesting with these, we need to extract some summary statistics. We will extract the coefficients of the model (the slope and intercept), and a measure of model fit (R^2) so we can ensure we are not drawing conclusions based on models that fit the data very poorly. The first few rows of `coef` are shown in Table 4.

```
R> rsq <- function(x) summary(x)$r.squared
R> bcoefs <- ldply(bmodels, function(x) c(coef(x), rsquare = rsq(x)))
R> names(bcoefs)[2:3] <- c("intercept", "slope")
```

Figure 9 displays the distribution of R^2 across the models. The models generally do a very bad job of fitting the data, although there are few with an R^2 very close to 1. We can see the data that generated these perfect fits by merging the coefficients with the original data, and then selecting records with an R^2 of 1:

```
R> baseballcoef <- merge(baseball, bcoefs, by = "id")
R> subset(baseballcoef, rsquare > 0.999)$id
```

```
 [1] "bannifl01" "bannifl01" "bedrost01" "bedrost01" "burbada01" "burbada01"
 [7] "carrocl02" "carrocl02" "cookde01"  "cookde01"  "davisma01" "davisma01"
[13] "jacksgr01" "jacksgr01" "lindbpa01" "lindbpa01" "oliveda02" "oliveda02"
[19] "penaal01"  "penaal01"  "powerte01" "powerte01" "splitpa01" "splitpa01"
[25] "violafr01" "violafr01" "wakefti01" "wakefti01" "weathda01" "weathda01"
[31] "woodwi01"  "woodwi01"
```

All the models with a perfect fit only have two data points. Figure 10 is another attempt to summarize the models. These plots show a negative correlation between slope and intercept, and the particularly bad models have estimates for both values close to 0. This concludes the baseball player case study, which used used `ddply`, `d_ply`, `dlply` and `ldply`. Our statistical analysis was not very sophisticated, but the tools of **plyr** made it very easy to work at the player level. This is an sensible first step when creating a hierarchical model.

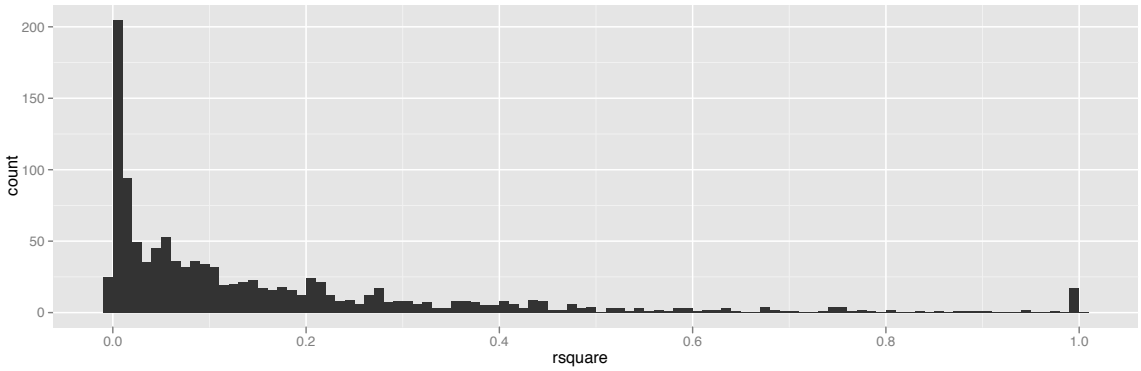


Figure 9: Histogram of model R^2 with bin width of 0.05. Most models fit very poorly!

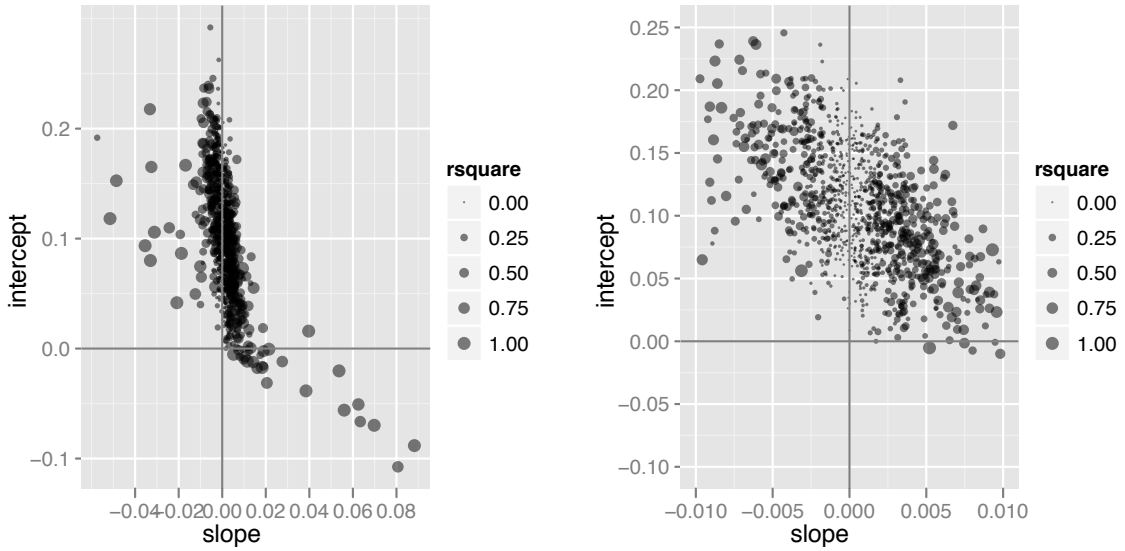


Figure 10: A scatterplot of model intercept and slope, with one point for each model (player). The size of the points is proportional to the R^2 of the model. Vertical and horizontal lines emphasize the x and y origins.

5.2. Case study: Ozone

In this case study we will analyze a 3d array that records ozone levels over a 24×24 spatial grid at 72 time points (Hobbs, Wickham, Hofmann, and Cook 2010). This produces a $24 \times 24 \times 72$ 3d array, containing a total of 41 472 data points. Figure 11 shows one way of displaying this data. Conditional on spatial location, each star glyph shows the evolution of ozone levels for each of the 72 months (6 years, 1995–2000). The construction of the glyph is described in Figure 12; it is basically a time series in polar coordinates. The striking seasonal patterns make it difficult to see if there are any long-term changes. In this case study, we will explore how to separate out and visualize the seasonal effects. Again we will start with the simplest case: A single time point, from location (1, 1). Figure 13 displays this in two ways: As a single line over time, or a line for each year over the months. This plot illustrates the striking

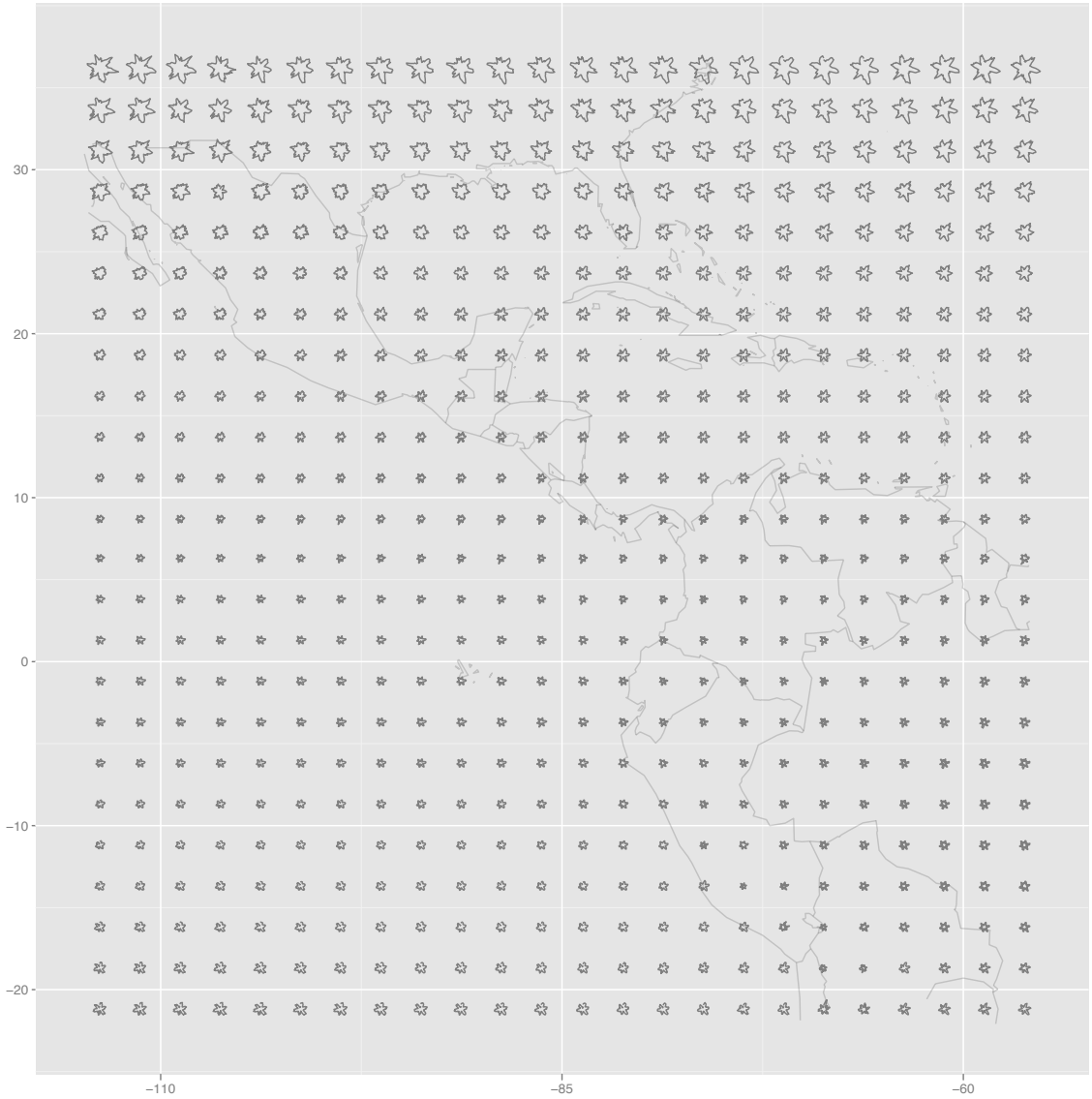


Figure 11: Star glyphs showing variation in ozone over time at each spatial location.

seasonal variation at this time point. The following code sets up some useful variables.

```
R> value <- ozone[1, 1, ]
R> time <- 1:72 / 12
R> month.abbr <- c("Jan", "Feb", "Mar", "Apr", "May",
+   "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")
R> month <- factor(rep(month.abbr, length = 72), levels = month.abbr)
R> year <- rep(1:6, each = 12)
```

We are going to use a quick and dirty method to remove the seasonal variation: Residuals from a robust linear model (Venables and Ripley 2002) that predicts the amount of ozone

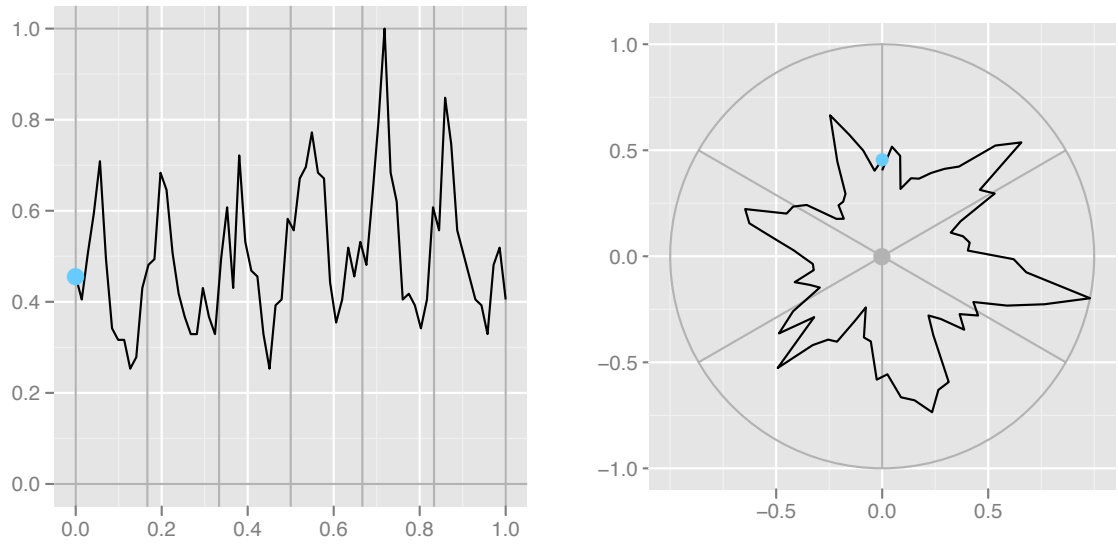


Figure 12: Star glyphs are time-series (left) plotted in polar coordinates (right). Both time and ozone value have been scaled to lie between 0 and 1: The smallest value in the entire dataset will be 0 and the largest will be 1. Grey lines indicate these boundaries, as well as the boundaries between the six years. A red point shows the position of the first value: it is close to the last value in the glyph. This glyph is the glyph on the top-left of Figure 11.

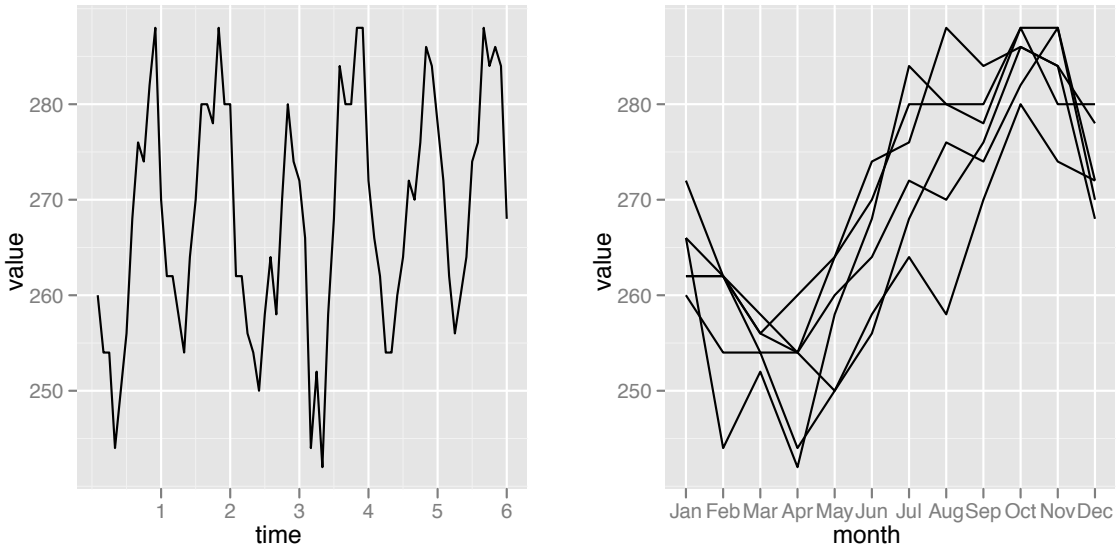


Figure 13: Two ways of displaying the seasonal changes. (Left) A single time series over all six years and (right) a line for each year.

for each month. We could use a regular linear model, but then our seasonal estimates might be thrown off by an unusual month. Figure 14 shows the deseasonalized trend from location (1, 1).

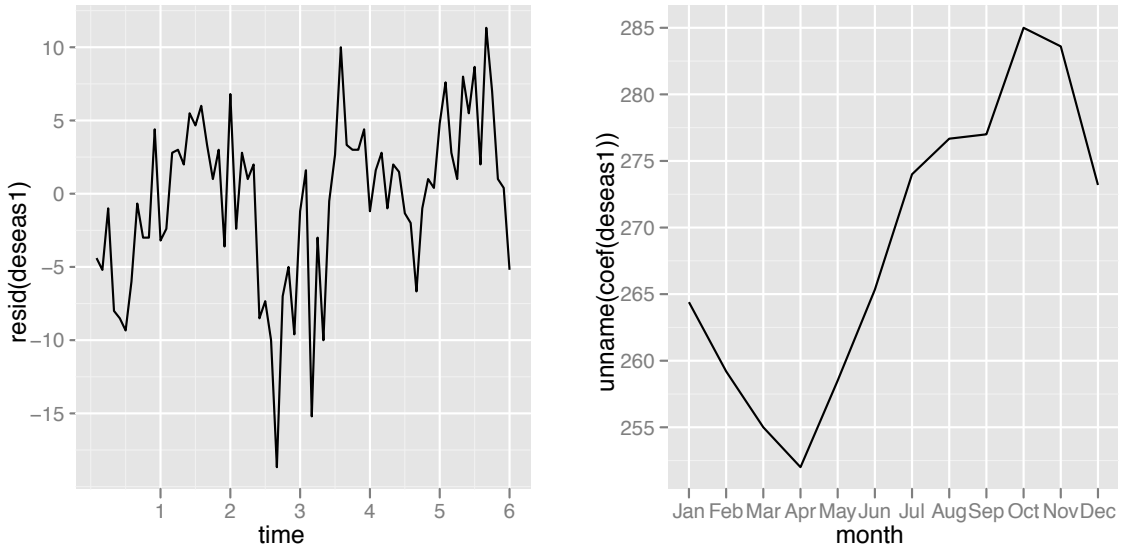


Figure 14: Deseasonalized ozone trends. (Left) Deseasonalized trend over six years. (Right) Estimates of seasonal effects. Compare to Figure 13.

```
R> library("MASS")
R> deseas1 <- rlm(value ~ month - 1)
R> summary(deseas1)

Call: rlm(formula = value ~ month - 1)
Residuals:
   Min     1Q Median     3Q    Max
 -18.7   -3.3    1.0    3.0   11.3

Coefficients:
         Value  Std. Error t value
monthJan 264.40   2.75      96.19
monthFeb 259.20   2.75      94.30
monthMar 255.00   2.75      92.77
monthApr 252.00   2.75      91.68
monthMay 258.51   2.75      94.05
monthJun 265.34   2.75      96.53
monthJul 274.00   2.75      99.68
monthAug 276.67   2.75     100.66
monthSep 277.00   2.75     100.78
monthOct 285.00   2.75     103.69
monthNov 283.60   2.75     103.18
monthDec 273.20   2.75      99.39

Residual standard error: 4.45 on 60 degrees of freedom

R> coef(deseas1)
```

```
monthJan monthFeb monthMar monthApr monthMay monthJun monthJul monthAug
     264      259      255      252      259      265      274      277
monthSep monthOct monthNov monthDec
     277      285      284      273
```

We next turn this into a function and fit the model to each spatial location. This does take a little while, but we are fitting 576 models! As is common when fitting large numbers of models, some of the models does not fit very well, and `rlm()` does not converge. We figure out where these lie by looking at the `converged` attribute for each model. In a real analysis it would be important to figure why these locations are troublesome and deal with them appropriately, but here we will just ignore them.

```
R> deseasf <- function(value) rlm(value ~ month - 1, maxit = 50)
R> models <- alply(ozone, 1:2, deseasf)

Warning message: rlm failed to converge in 50 steps
Warning message: rlm failed to converge in 50 steps
Warning message: rlm failed to converge in 50 steps
Warning message: rlm failed to converge in 50 steps
Warning message: rlm failed to converge in 50 steps
Warning message: rlm failed to converge in 50 steps
Warning message: rlm failed to converge in 50 steps

R> failed <- laply(models, function(x) !x$converged)
```

From those models we extract the deseasonalized values (the residuals) and the seasonal coefficients. Looking at the dimensionality we see that they are in the same format as the original data. We also carefully label the new dimensions. This is important: Just as data frames should have descriptive variable names, arrays should always have descriptive dimension labels.

```
R> coefs <- laply(models, coef)
R> dimnames(coefs)[[3]] <- month.abbr
R> names(dimnames(coefs))[3] <- "month"
R> deseas <- laply(models, resid)
R> dimnames(deseas)[[3]] <- 1:72
R> names(dimnames(deseas))[3] <- "time"
R> dim(coefs)

[1] 24 24 12

R> dim(deseas)

[1] 24 24 72
```

We now have a lot of data to try and understand: For each of the 576 locations we have 12 estimates of monthly effects, and 72 residuals. There are many different ways we could

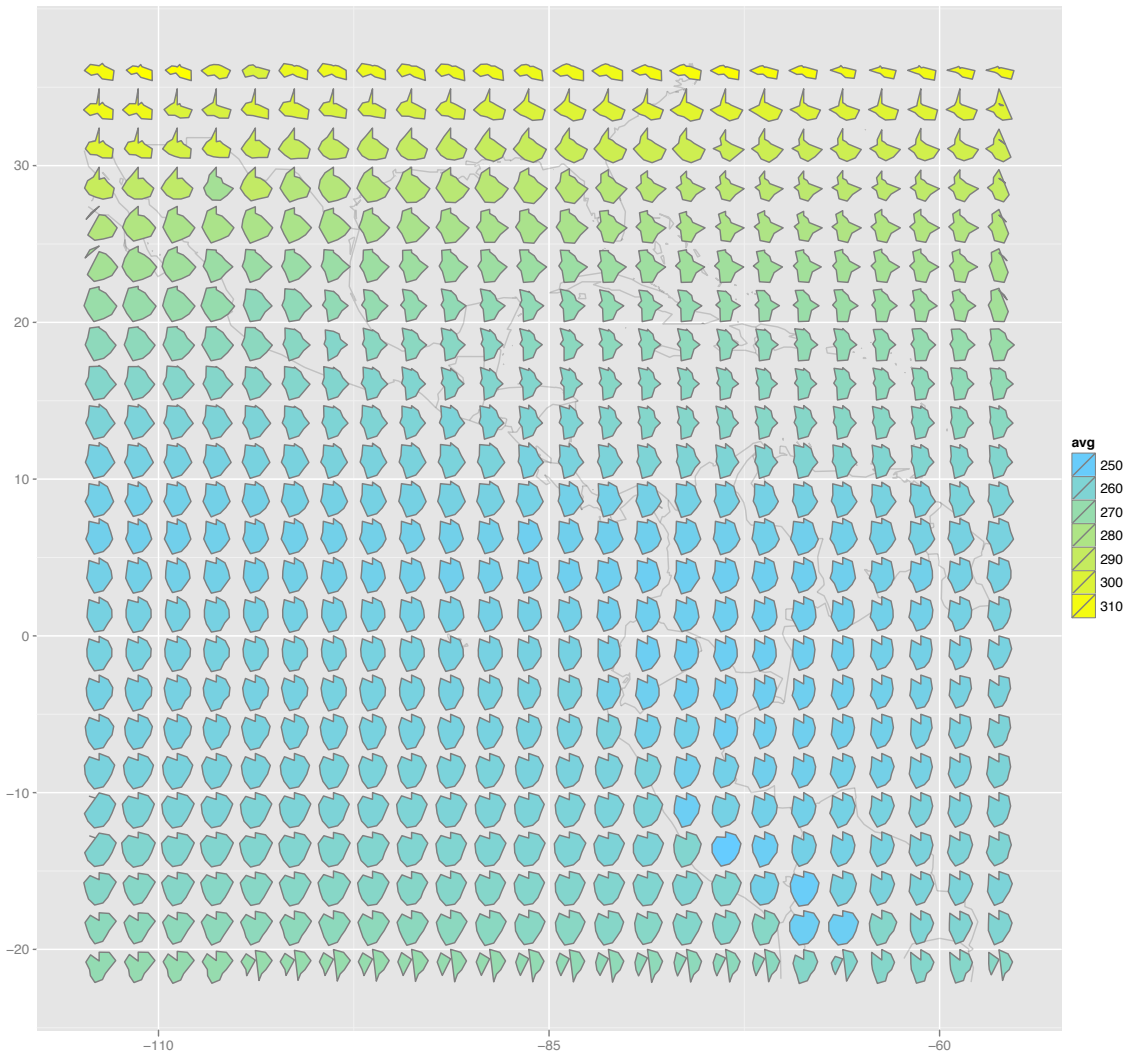


Figure 15: Star glyphs showing seasonal variation. Each star shows the twelve estimates of monthly seasonal effects, standardized to have maximum one. This focuses on the overall pattern of changes, rather than the absolute values, given by the glyph colour. Note the strong spatial correlation: Nearby glyphs have similar shapes.

visualize this data. Figures 15 and 16 visualize these results with star glyph plots. For plotting, it is more convenient to have the data in data frames. There are a few different ways to do this: We can convert from the 3d array to a data frame with `melt()` from the **reshape** package, or use `ldply()` instead of `laply()`. For this example, we will use a combination of these techniques. We will convert the original array to a data frame, add on some useful columns, and then perform the same steps as above with this new format. Notice how our effort labeling the array dimensions pays off with informative column names in `coefs_df`: `lat`, `long` and `month`.

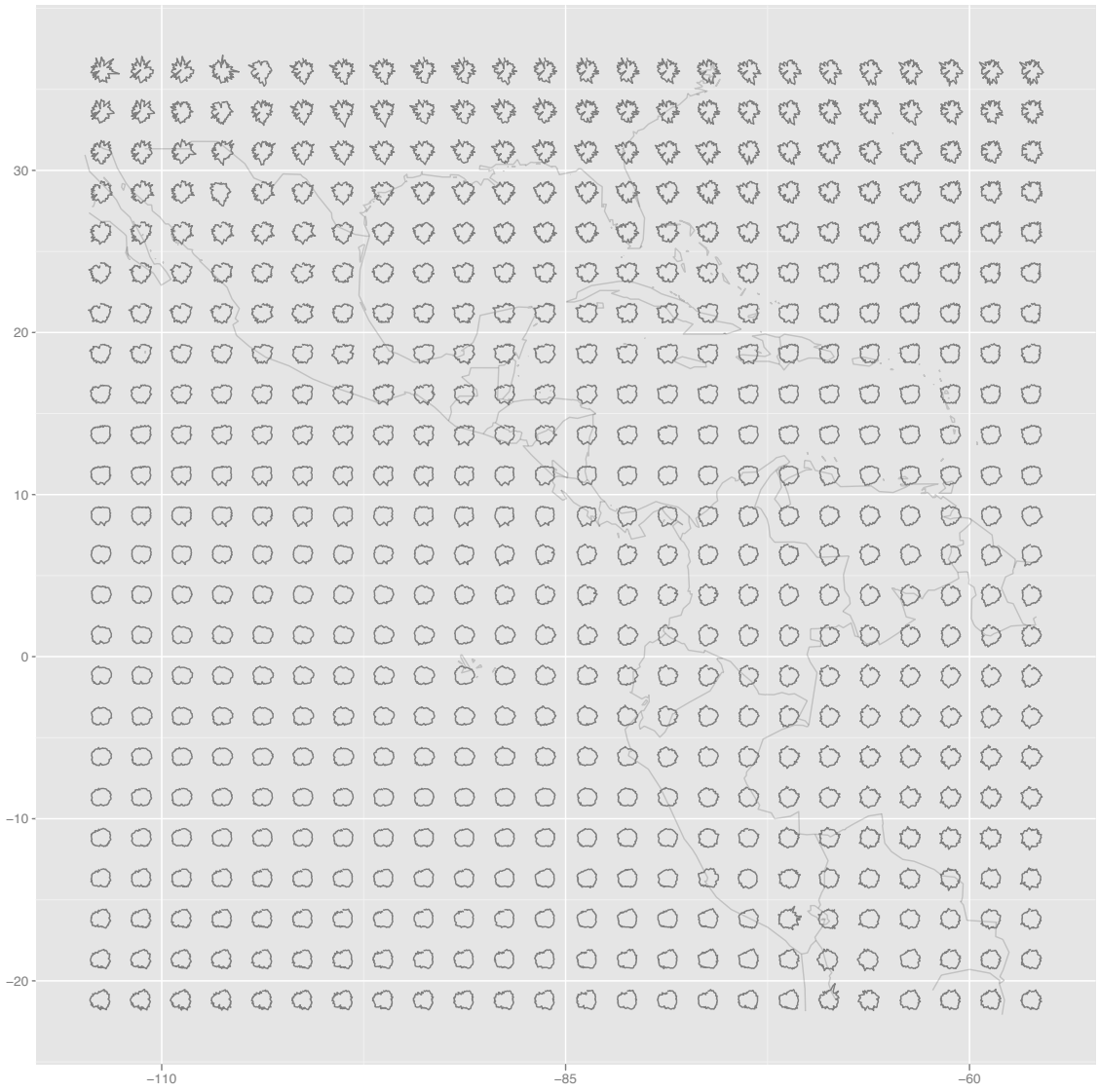


Figure 16: Star glyphs showing deseasonalized trends. Each star shows six years of data, with seasonal trend removed. This plot contains a lot of data—over 40,000 observations—and rewards detailed study. Looking at a printed version also helps as the resolution of a printer (600 dpi) is much higher than that of the screen (∼100 dpi). Interesting features include the higher variability in the North, locations in the mountains of South America with a large difference between starting and ending temperatures, and an unusual month common to many of the locations in the Pacific.

```
R> coefs_df <- melt(coefs)
R> head(coefs_df)

    lat long month value
1 -21.2 -114   Jan   264
```

```
2 -18.7 -114   Jan   261
3 -16.2 -114   Jan   261
4 -13.7 -114   Jan   259
5 -11.2 -114   Jan   256
6  -8.7 -114   Jan   255
```

```
R> coefs_df <- ddply(coefs_df, .(lat, long), transform,
+    avg = mean(value),
+    std = value / max(value)
+  )
R> head(coefs_df)
```

```
    lat long month value avg   std
1 -21.2 -114   Jan   264 269 0.928
2 -21.2 -114   Feb   259 269 0.909
3 -21.2 -114   Mar   255 269 0.895
4 -21.2 -114   Apr   252 269 0.884
5 -21.2 -114   May   259 269 0.907
6 -21.2 -114   Jun   265 269 0.931
```

```
R> deseas_df <- melt(deseas)
R> head(deseas_df)
```

```
    lat long time value
1 -21.2 -114    1 -4.40
2 -18.7 -114    1 -3.33
3 -16.2 -114    1 -2.96
4 -13.7 -114    1 -5.00
5 -11.2 -114    1 -4.00
6  -8.7 -114    1 -3.00
```

The star glyphs show temporal patterns conditioned on location. We can also look at spatial pattern conditional on time. One way to do this is to draw tile plots where each cell of the 24×24 grid is colored according to its value. The following code sets up a function with constant scales to do that. Figure 17 shows the spatial variation of seasonal coefficients for January and July.

```
R> coef_limits <- range(coefs_df$value)
R> coef_mid <- mean(coefs_df$value)
R> monthsurface <- function(mon) {
+    df <- subset(coefs_df, month == mon)
+    qplot(long, lat, data = df, fill = value, geom="tile") +
+    scale_fill_gradient(limits = coef_limits,
+      low = brightblue, high = "yellow") +
+      map + opts(aspect.ratio = 1)
+  }
```

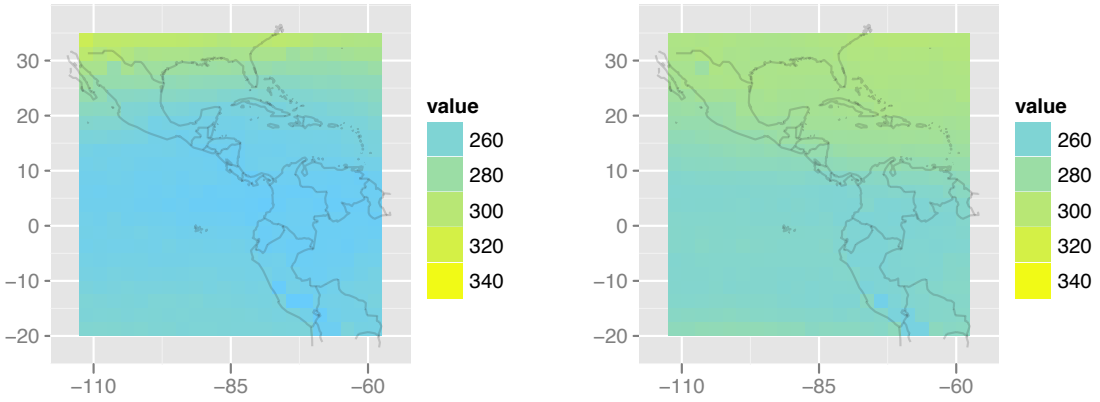


Figure 17: Tile plots of coefficients for January (left) and July (right).

We could do the same thing for the values themselves, but we would probably want to make an animation rather than looking at all 72 plots individually. The `*_ply` functions are useful for making animations because we are only calling the plotting function for its side effects, not because we are interested in its value.

```
R> pdf("ozone-animation.pdf", width = 8, height = 8)
R> l_ply(month.abbr, monthsurface, .print = TRUE)
R> dev.off()
```

5.3. Other uses

The `transform()`, `summarise()` and `subset()` functions work well in combination with **plyr**. Transform makes it very easy to perform randomization within groups. For example the following expression returns a data frame like `coefs_df`, but with the values in the `time` column randomized within each latitude/longitude grouping:

```
ddply(coefs_df, .(lat, long), transform, time = sample(time))
```

This technique is useful for performing block bootstrapping and other related permutation tests, and is related to the `ave` function in base R. Scaling variables within a group is also trivial:

```
ddply(coefs_df, .(lat, long), transform, value = scale(value))
```

We can create group-wise summaries with `summarise()` (or `summarize()`). For example, it is easy to summarize the range of ozone at each location:

```
ddply(coefs_df, .(lat, long), summarise,
  ozone_min = min(value), ozone_max = max(ozone))
```

Base function	Input	Output	**plyr** function
`aggregate`	d	d	`ddply` + `colwise`
`apply`	a	a/l	`aaply` / `alply`
`by`	d	l	`dlply`
`lapply`	l	l	`llply`
`mapply`	a	a/l	`maply` / `mlply`
`replicate`	r	a/l	`raply` / `rlply`
`sapply`	l	a	`laply`

Table 5: Mapping between apply functions and **plyr** functions.

Group-wise subsetting is easy with `subset()`. For example, if we wanted to extract the observation in each group with the lowest value ozone of ozone, it is just as easy:

```
ddply(coefs_df, .(lat, long), subset, value == min(value))
```

For simulations, `mdply()` can be very useful, because it is easy to generate a grid of parameter values and then evaluate them. This can also be useful when testing many possible combinations to input to a function.

```
mdply(expand.grid(mean = 1:5, sd = 1:5), as.data.frame(rnorm), n = 10)
```

6. Related work

There are a number of other approaches to solving the problems that **plyr** solves. You can always use loops, but loops create a lot of book-keeping code that obscures the intent of your algorithm. This section describes other high-level approaches similar to **plyr**.

Table 5 describes the functions in base R that work similarly to functions in **plyr**. The built-in R functions focus mainly on arrays and lists, not data frames, and most attempt to return an atomic data structure if possible, and if not, a list. This ambiguity of the output type is fine for interactive use, but does make programming with these functions tricky. Compared to `aaply`, `apply` returns the new dimensions first, rather than last, which means it is not idempotent when used with the `identity` function. In contrast, `aaply(x, a, identity) == aperm(x, unique(c(a, seq_along(dim(x)))))` for all `a`.

Related functions `tapply` and `sweep` have no corresponding function in **plyr**, and remain useful. `merge` is useful for combining summaries with the original data.

Contributed packages also tackle this problem:

- The **doBy** (Højsgaard 2006, 2011) package provides versions of `order`, `sample`, `split`, `subset`, `summary` and `transform` that make it easy to perform each of these operations on subsets of data frames, joining the results back into a data frame. These functions are rather like a specialized version of `ddply` with a formula based interface, which, particularly for `summary`, makes it easy to only operate on selected columns.

- The **abind** (Plate and Heiberger 2011) package provides the `abind()` function which can be used to construct multidimensional arrays in a similar way to the `*aply` functions.
- The **gdata** (Warnes 2010) package contains a bundle of helpful data manipulation functions, including `frameApply` which works like `ddply` or `dlply` depending on its arguments.
- The **scope** (Bergsma 2007) package provides `scope`, `scoop`, `skim`, `score` and `probe` which provide a composable set of functions for operating symbolically on subsets of data frames.
- The **reshape** (Wickham 2007) package is similar to **Excel** pivot tables and provides tools for rearranging matrices and data frames. The `cast` function in the **reshape** package is closely related to `aaply`.
- The **sqldf** (Grothendieck 2010) package allows you to use SQL commands with R data frames. This gives the user access to a powerful set-based data access language.

7. Conclusion

Speed-wise **plyr** is competitive with R for small to moderate sized datasets, and generally a little faster for large datasets split by many different values. It is more memory-efficient than the naive split-apply-combine approach because **plyr** is careful not to make an extra copy of the data in the split step. Further efficiency gains are possible, particularly by implementing key parts C for maximum speed and memory efficiency. The basic algorithm of **plyr** is trivially parallelizable, and future versions will integrate with the **foreach** package (REvolution Computing 2009) to make use of multiple cores and multiple machines.

More generally, what are other common strategies used in data analysis? How can we identify these strategies and then develop software to support them? It is difficult to step back and identify these patterns as trivial details may obscure the common components; it took four years of thinking about related problems before I recognized this split-apply-combine strategy. However, the task is important because the patterns are so useful. Personally, identifying the split-apply-combine strategy has made it much easier for me to solve common data analysis problems, and I have also found useful when teaching others how to do data analysis.

8. Acknowledgments

Thanks go to Norman Josephy, Austin F. Frank, Antony Unwin, Joseph Voelkel, Erik Iverson, and Jean-Olivier Irisson for their comments on early versions of this paper.

References

Bergsma T (2007). ***scope**: Data Manipulation Using Arbitrary Row and Column Criteria*. R package version 2.2, URL http://CRAN.R-project.org/src/contrib/Archive/scope/.

Dalgaard P (2001). "A Primer on the R-Tcl/Tk Package." *R News*, **1**(3), 27–31. URL http://CRAN.R-project.org/doc/Rnews/.

Friendly M, Fox J (1994). "Using APL2 to Create an Object-Oriented Environment for Statistical Computation." *Journal of Computational and Graphical Statistics*, **3**, 387–407.

Grothendieck G (2010). ***sqldf**: Perform SQL Selects on R Data Frames*. R package version 0.3-5, URL http://CRAN.R-project.org/package=sqldf.

Hobbs J, Wickham H, Hofmann H, Cook D (2010). "Glaciers Melt as Mountains Warm: A Graphical Case Study." *Computational Statistics*, **25**(4), 569–586. Special issue for ASA Statistical Computing and Graphics Data Expo 2007.

Højsgaard S (2006). "The **doBy** Package." *R News*, **6**(2), 47–49. URL http://CRAN.R-project.org/doc/Rnews/.

Højsgaard S (2011). ***doBy**: Groupwise Summary Statistics, General Linear Contrasts, LSMEANS (Least-Squares-Means), and Other Utilities*. R package version 4.2.3. With contributions from Ulrich Halekoh, Jim Robison-Cox, Kevin Wright, Alessandro A. Leidi, URL http://CRAN.R-project.org/package=doBy.

Plate T, Heiberger R (2011). ***abind**: Combine Multi-dimensional Arrays*. R package version 1.3-0, URL http://CRAN.R-project.org/package=abind.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

REvolution Computing (2009). ***foreach**: Foreach looping construct for R*. R package version 1.3.0, URL http://CRAN.R-project.org/package=foreach.

Sarkar D (2008). ***lattice**: Multivariate Data Visualization with R*. Springer-Verlag, New York.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York.

Warnes GR (2010). ***gdata**: Various R Programming Tools for Data Manipulation*. R package version 2.8.0. With contributions from Ben Bolker, Gregor Gorjanc, Gabor Grothendieck, Ales Korosec, Thomas Lumley, Don MacQueen, Arni Magnusson, Jim Rogers, and others, URL http://CRAN.R-project.org/package=gdata.

Wickham H (2007). "Reshaping data with the **reshape** package." *Journal of Statistical Software*, **21**(12), 1–20. URL http://www.jstatsoft.org/v21/i12/paper.

Wickham H (2010). ***ggplot2**: An Implementation of the Grammar of Graphics*. R package version 0.8.7, URL http://CRAN.R-project.org/package=ggplot2.

Affiliation:

Hadley Wickham
Rice University
Houston, TX 77251-1892, United States of America
E-mail: hadley@rice.edu
URL: http://had.co.nz/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 40, Issue 1 *Submitted:* 2009-09-24
April 2011 *Accepted:* 2010-12-27

Things I Wish Someone Had Told Me When I Was Learning How to Code

freeCodeCamp (https://medium.freecodecamp.org/things-i-wish-someone-had-told-me-when-i-was-learning-how-to-code-565fc9dcb329?gi=462ce53a4e6a) · by Cecily Carver · November 22, 2013

And what I've learned from teaching others

(https://medium.freecodecamp.org/@cecilycarver?source=post_header_lockup)
Cecily Carver (https://medium.freecodecamp.org/@cecilycarver)
Nov 22, 2013

Before you learn to code, think about *what* you want to code

Knowing how to code is mostly about building things, and the path is a lot clearer when you have a sense of the end goal. If your goal is "learn to code," without a clear idea of the kinds of programs you will write and how they will make your life better, you will probably find it a frustrating exercise.

I'm a little ashamed to admit that part of my motivation for studying computer science was that I wanted to prove I was smart, and I wanted to be able to get Smart Person jobs. I also liked thinking about math and theory (this book (http://www.amazon.ca/Godel-Escher-Bach-Eternal-Golden/dp/0465026567) blew my mind at an impressionable age) and the

program was a good fit. It wasn't enough to sustain me for long, though, until I found ways to connect technology to the things I really loved, like music and literature.

So, what do you want to code? Websites? Games? iPhone apps? A startup that makes you rich? Interactive art? Do you want to be able to impress your boss or automate a tedious task so you can spend more time looking at otter pictures? Perhaps you simply want to be more employable, add a buzzword to your resume, or fulfill the requirements of your educational program. All of these are worthy goals. Make sure you know which one is yours, and study accordingly.

There's nothing mystical about it

Coding is a skill like any other. Like language learning, there's grammar and vocabulary to acquire. Like math, there are processes to work through specific types of problems. Like all kinds of craftsmanship and art-making, there are techniques and tools and best practices that people have developed over time, specialized to different tasks, that you're free to use or modify or discard.

This guy (http://www.joelonsoftware.com/articles/ThePerilsofJavaSchools.html) (a very smart guy! Whose other writings I enjoy and frequently agree with!) posits that there is a bright line between people with the True Mind of a Programmer and everyone else, who are lacking the intellectual capacity needed to succeed in the field. That bright line consists, according to him, of pointers and recursion (there are primers here (http://alumni.cs.ucr.edu/~pdiloren/C++_Pointers/) and here (http://inventwithpython.com/blog/2011/08/11/recursion-explained-with-the-flood-fill-algorithm-and-zombies-and-cats/) for the curious).

I learned about pointers and recursion in school, and when I understood them, it was a delightful jolt to my brain—the kind of intellectual pleasure that made me want to study computer science in the first place. But, outside of classroom

exercises, the number of times I've had to be familiar with either concept to get things done has been relatively small. And when helping others learn, over and over again, I've watched people complete interesting and rewarding projects without knowing anything about either one.

There's no point in being intimidated or wondering if you're Smart Enough. Sure, the more complex and esoteric your task, the higher the level of mastery you will need to complete it. But this is true in absolutely every other field. Unless you're planning to make your living entirely by your code, chances are you don't have to be a recursion-understanding genius to make the thing you want to make.

It never works the first time

And probably won't the second or third time

When you first start learning to code, you'll very quickly run up against this particular experience: you think you've set up everything the way you're supposed to, you've checked and re-checked it, and it still. doesn't. work. You don't have a clue where to begin trying to fix it, and the error message (if you're lucky enough to have one at all) might as well say "fuck you." You might be tempted to give up at this point, thinking that you'll never figure it out, that you're not cut out for this. I had that feeling the first time I tried to write a program in C++, ran it, and got only the words "segmentation fault" for my trouble.

But this experience is so common for programmers of all skill levels that it says absolutely nothing about your intelligence, tech-savviness, or suitability for the coding life. It will happen to you as a beginner, but it will also happen to you as an experienced programmer. The main difference will be in how you respond to it.

I've found that a big difference between new coders and experienced coders is faith: faith that things are going wrong for a logical and discoverable reason, faith that problems are fixable, faith that there is a way to accomplish the goal. The path from "not working" to "working" might not be obvious, but with patience you can usually find it.

Someone will always tell you you're doing it wrong

There are almost always many different approaches to a particular problem, with no single "right way." A lot of programmers get very good at advocating for their preferred way, but that doesn't mean it's the One True Path. Going head-to-head with people telling me I was Wrong, and trying to figure out if they were right, was one of the more stressful aspects of my early career.

If you're coding in a team with other people, someone will almost certainly take issue with something that you're doing. Sometimes they'll be absolutely correct, and it's always worth investigating to see whether you are, in fact, Doing It Wrong. But sometimes they will be full of shit, or re-enacting an ancient and meaningless dispute where it would be best to just follow a style guide and forget about it.

On the other hand, if you're the kind of person who enjoys ancient but meaningless disputes (grammar nerds, I'm looking at you), you've come to the right place.

Someone will always tell you you're not a real coder

"Coding" means a lot of different things to a lot of different people, and it looks different now from how it used to. And, funnily enough, the tools and packages and frameworks that make it faster and easier for newcomers or even trained developers to build things are most likely to be tarred with the "not for REAL

coders" brush. (See: "Return of the Real Programmer (http://blog.enfranchisedmind.com/2009/04/return-of-the-real-programmer/)")

Behind all this is the fear that if "anyone" can call themselves a programmer, the title will become meaningless (http://fullcomment.nationalpost.com/2013/08/21/chase-felker-youre-not-a-computer-programmer-and-thats-ok/). But I think that this gatekeeping is destructive.

Use the tools that make it easiest to build the things you want to build. If that means your game was made in Stencyl or GameMaker rather than written from scratch, that's fine. If your first foray into coding is HTML or Excel macros, that's fine. Work with something you feel you can stick with.

As you get more comfortable, you'll naturally start to find those tools limiting rather than helpful and look for more powerful ones. But most of the time, few people will ever even look at your code or even ask what you used—It's what you make with it that counts.

Worrying about "geek cred" will slowly kill you

See above. I used to worry a lot, especially in school, about whether I was identifying myself as "not a real geek" (and therefore less worthy of inclusion in tech communities) through my clothing, my presentation, my choice of reading material and even my software customization choices. It was a terrible waste of energy and I became a lot more functional after I made the decision to let it all go.

You need to internalize this: your ability to get good at coding has *nothing* to do with how well you fit into the various geek subcultures. This goes double if you know deep down that you'll never quite fit. The energy you spend proving

yourself should be going into making things instead. And, if you're an indisputable geek with cred leaking from your eye sockets, keep this in mind for when you're evaluating someone else's cred level. It may not mean what you think it does.

Sticking with it is more important than the method

There's no shortage of articles about the "right" or "best" way to learn how to code, and there are lots of potential approaches. You can learn the concepts from a book (http://pine.fm/LearnToProgram/) or by completing interactive exercises (http://www.codecademy.com/) or by debugging things that others have written (http://learnpythonthehardway.org/book/intro.html). And, of course, there are lots of languages you might choose as your first to learn, with advocates for each.

A common complaint with "teach yourself to code" programs and workshops is that you'll breeze happily through the beginner material and then hit a steep curve where things get more difficult very quickly. You know how to print some lines of text on a page but have no idea where to start working on a "real," useful project. You might feel like you were just following directions without really understanding, and blame the learning materials.

When you get to this stage, most of the tutorials and online resources available to you are much less useful because they assume you're already an experienced and comfortable programmer. The difficulty is further compounded by the fact that "you don't know what you don't know." Even trying to figure out what to learn next is a puzzle in itself.

You'll hit this wall no matter what "learn to code" program you follow, and the only way to get past it is to persevere. This means you keep trying new things, learning more information, and figuring out, piece by piece, how to build your

project. You're a lot more likely to find success in the end if you have a clear idea of why you're learning to code in the first place.

If you keep putting bricks on top of each other, it might take a long time but eventually you'll have a wall. This is where that faith I mentioned earlier comes in handy. If you believe that with time and patience you can figure the whole coding thing out, in time you almost certainly will.

freeCodeCamp (https://medium.freecodecamp.org/things-i-wish-someone-had-told-me-when-i-was-learning-how-to-code-565fc9dcb329?gi=462ce53a4e6a) · by Cecily Carver · November 22, 2013

Feminist Data Visualization

Catherine D'Ignazio and Lauren F. Klein

Abstract—In this paper, we begin to outline how feminist theory may be productively applied to information visualization research and practice. Other technology- and design-oriented fields such as Science and Technology Studies, Human-Computer Interaction, Digital Humanities, and Geography/GIS have begun to incorporate feminist principles into their research. Feminism is not (just) about women, but rather draws our attention to questions of epistemology – who is included in dominant ways of producing and communicating knowledge and whose perspectives are marginalized. We describe potential applications of feminist theory to influence the information design process as well as to shape the outputs from that process.

Index Terms—Visualization, inclusion, digital humanities, critical perspectives, feminism.

1 Introduction

When exploring the intersection of data visualization and the digital humanities, one must consider not only how the domain of digital humanities – and of the humanities more generally – can provide new opportunities for the design and application of visualization tools and techniques, but also how theories from the humanities can themselves inform visualization design. Research in the field of data visualization is often framed in terms of how it helps to "reveal" knowledge [15], support narrative storytelling [70], or otherwise facilitate pathways to "insight" [12]. These same keywords are often employed – and challenged – in humanistic theories that explore how knowledge is produced, transmitted, and perceived. Among the earliest and the most enduring of these theoretical schools is what is known as *feminist theory*. A body of work that owes its emergence to the women's suffrage movements of the nineteenth century, feminist theory evolved through several "waves" over the course of the twentieth century, and now encompasses a range of ideas about how identity is constructed, how power is assigned, and how knowledge is generated, as well as how a range of *intersectional* forces [19] such as race, class, and ability, combine to influence the experience of being in the world.

In this paper, we outline a feminist approach to visualization, drawing upon a set of canonical and contemporary theories from the humanities in order to show how visualization research can be adapted to emphasize the situated nature of knowledge and its perception. Our goal is to encourage the development of a range of alternative visualization practices that better emphasize the design decisions associated with data and its visual display. We are particularly interested in exposing the assumptions involved in choices about data type, categorization schema, visual typology, interaction mode, and intended audience; as well as those associated with the qualitative aspects of visualization design and its reception, such as the composition and structure of the design team, the identification and involvement of user communities, the contextual and affective factors that influence both the design and reception of visualizations, and the many forms of labor that contribute to a successful visualization design. By identifying these assumptions and associating them with the core principals of what we term *feminist data visualization*, we hope to expand the conversation about what visualization for – and with – the humanities could become.

- *Catherine D'Ignazio is with Emerson College and Massachusetts Institute of Technology. E-mail: catherine_dignazio@emerson.edu.*
- *Lauren F. Klein is with the Georgia Institute of Technology. E-mail: lauren.klein@lmc.gatech.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

2 Related Work

Feminism is "not (just) a women's issue," as Johanna Drucker reminds us, nor does feminist theory help to inform issues of gender alone [25]. As the binary distinction between male and female, as well as the hierarchical relation that posits male above female, have been abstracted to serve as models for a range of structures and systems, feminist theory has been marshaled in order to challenge the validity of a variety of binaristic and hierarchical configurations. By the same token, expansions of feminist theory – crucially, intersectional feminism – have been employed to overturn systems of oppression that cannot be reduced to a single structure or source. We lead with this theoretical lens so as to frame the four related fields of inquiry that have contributed to our formulation of feminist data visualization: feminist science and technology studies, feminist human-computer interaction, feminist digital humanities, and critical cartography & GIS. In the following sections, we summarize the main contributions of each field in more detail.

2.1 Feminist Science and Technology Studies

Science and Technology Studies (STS) is an interdisciplinary field that emerged in the 1960s and 70s. STS examines the social, cultural, and historical aspects of science and technology. Feminist theory and analysis has played a key role in this field, leading to the development of original theoretical frameworks [4, 6, 34, 39] as well as the sustained challenge to dominant epistemological perspectives [37, 47, 56, 80]. One of the key contributions of STS has been to challenge the idea that science and/or technology is objective and neutral by demonstrating how scientific thought is situated in particular cultural, historical, economic, and social systems [77]. Feminist STS, both implicitly and explicitly, looks to the perspectives of those marginalized by current power configurations (including and especially those marginalized because of gender, sexuality, race, and/or ethnicity) as a way of exposing how their perspectives are not included in what is considered "objective" truth [74]. Challenging neutrality, objectivity, and universality does not mean that feminist STS retreats to a position of relativism or solipsism, however. The field rejects neither the scientific process nor quantitative ways of knowing the world. Rather, feminist STS allows us to see how all knowledge is situated, how certain perspectives are excluded from the current knowledge regime, and how multiple true objectivities are possible.

2.2 Feminist Human Computer Interaction

In the field of human-computer interaction (HCI), there is an emerging conversation about how to draw from feminist theory and other critical perspectives for the design of interactive and computational systems. Lucy Suchman's work has long explored the implications of feminist theory for technology production and use [75, 76]. More recently, Shaowen Bardzell has asserted that feminism can be deployed throughout the design process to produce a "generative contribution" [5]. Feminist HCI design work has included foci on female

makers and hackerspaces [32], motherhood as a life phase [3], postpartum technologies [22], and talking back to street harassment [23]. While historically, HCI aspired to "universal" usability, the early 2000s saw a proliferation of work that challenged that idea through design practice [24, 60]. Feminist HCI builds on that work and draws on feminist standpoint theory [39] to explicitly valorize marginal perspectives so as to "expose the unexamined assumptions of dominant epistemological paradigms" [5]. In relation to visualization in particular, Peter Hall coined the term *critical visualization* in 2008 to describe practices that counteract "the technological view" of visualization, a view which emphasizes technique and efficiency while eliding historical, social and rhetorical concerns [35]. Marian Dörk et al. built on this concept to elucidate design principles for working with data [26]. In both feminist HCI and critical information visualization, researchers have introduced design principles that attempt to draw attention to how knowledge resides in specific bodies (*disclosure/self-disclosure*, *embodiment*), how power is distributed throughout the design process (*empowerment*, *advocacy*, *ecology*) and how to include more voices and alternative perspectives in the design process, as well as the experience of the resulting artifact (*participation*, *pluralism*, *plurality*).

2.3 Feminist Digital Humanities

Since the field's inception, digital humanities (DH) has entailed a sustained attention to certain feminist concerns. The Orlando Project [10] and the Women Writers Project [31] are early and enduring examples of how DH has emphasized the recovery of female literary and cultural contributions. In recent years, the fields of DH and STS have begun to converge, resulting in a range of projects that incorporate feminist theoretical perspectives into their digital work. Key work in this area has included wearable representations of Twitter activity related to reproductive health [50] and embodied enactments of historical health data [67], as well as content management platforms [58], social media collectives [9], scholarly networks [30], and educational opportunities [78]. In terms of visualization work, gender – especially as it relates to issues of authorship and style – has long served as a subject of DH research, e.g. [46]. However, the visualizations that accompany such analyses almost always employ standard representational techniques [45]. Recently, Miriam Posner [65] identified the development of new visual strategies for the representation of non-binary gender as one of the most pressing challenges of DH today. Other work that seeks to incorporate embodied and affective modes of perception into new visualization forms, e.g. [49, 79], promises to extend feminist digital humanities visualization work in exciting ways. To date, however, this work has been conducted in isolation from the visualization community. Additional partnerships between DH scholars and visualization researchers, along the lines of interdisciplinary projects to visualize the sonic aspects of poetry [57] or the ambiguities of temporal data [59], constitute a rich site for future inquiry.

2.4 Critical Cartography & GIS

In the late 1980s and early 1990s, geographers challenged conventional academic cartography by linking maps and other visual representations of geographic knowledge explicitly to power using the critical theories of Michel Foucault [17, 18]. Cartographers such as JB Harley challenged the perceived neutrality of the map and introduced notions of ideology and bias [40, 41]. While he did not explicitly draw on feminist theory, Harley argued for the situatedness of maps as historically and culturally contingent documents. During the same period, Denis Wood connected maps explicitly to the rise of the nation-state and showed how maps serve political interests [82]. Other scholars linked Geographic Information Systems (GIS) to an impoverished techno-positivism [64] and militarism [71]. Subsequent scholarship theorized the map more as rhetorical proposition than depiction of "fact" [81]. The declassification of GIS technology and the introduction of locative functions into everyday devices like mobile phones has led to a flourishing of artistic and critical mapping practices [20] like Laura Kurgan's work that intentionally introduces social and political questions through visualization [51]. Relatedly, there has also been an expansion of participatory design strategies for democratizing geographic information collection and visualization [27] for those who have been excluded from dominant mapping practices, such as indigenous populations [16]. Since 2000, scholars have articulated explicitly feminist approaches to mapping and GIS [28, 42, 62] including nuanced considerations of gender and mapping with new technologies [73]. Mei-Po Kwan used the term *feminist visualization* to describe how GIS could be used in ways that are compatible with feminist epistemologies and politics [52]. Her design principles include grounding mapping practices in women's everyday lives and political struggles, as well as incorporating qualitative and narrative components into spatial representations.

3 Principles of Feminist Data Visualization

In what follows, we introduce six core principles of feminist data visualization. As our intention is to directly impact the design of future visualizations, we follow our explanation of each principle with a set of questions relating to design process and design output. We should note, also, that while our primary focus is on visualization design and the related issues of interaction and display, our feminist approach requires that we expand the design frame so as to account for the range of social forces and material conditions that influence the design process. In other words, a feminist approach to data visualization, while centered on design, insists that data, design, and community of use, are inextricably intertwined.

3.1 Rethink Binaries

Central to feminist theory is the disavowal of binary distinctions– not only between the categories of male and female, but also between nature and culture [37], subject and object [43], reason and emotion [54], and body and world [4], among many others. A feminist approach to data visualization should therefore emphasize representational strategies premised on multiplicity rather than binaries, and acknowledge the limits of any binaristic view [53]. This approach is exemplified by (if not limited to) the representation of gender; typically recorded as binary and discrete variables – e.g. either male or female – gender might be better represented as continuous and multidimensional [29]. Not only a challenge for the visualization phase of research, rethinking the representation of gender, among other binaristic categories, challenges us to inquire how the processes associated with data collection and classification, as well as their visual display, might be made to better account for a range of multiple and fluid categories.

Design Process Questions: Is our data the right type? What categories have we taken for granted? How can we register responses that do not fit into the categories we have provided, even and especially if they are "edge cases" and "outliers"?

Design Output Questions: How do we communicate the limits of our categories in the final representation? How can we allow the user to refactor the categories we have presented for view?

3.2 Embrace Pluralism

Feminist theory seeks to challenge claims of objectivity, neutrality and universalism, emphasizing instead how knowledge is always constructed within the context of a specific subject position [8, 38, 39, 54]. In the context of data visualization, a focus on the designer's own subject position can help to expose the decisions, both implicit and explicit, that contribute to the creation of any particular visual display. Both Bardzell and Dörk et al. have framed this quality around "self-disclosure" [5, 26]. We believe that self-disclosure, and an embrace of pluralism more generally, can do more; it can help to encourage alternatives to the single "view from nowhere" so often favored in visualization design [21]. Ideally, a focus on pluralism would help visualization research move away from its current emphasis on "objective" presentation in favor of designs that facilitate pathways to multiple truths.

Design Process Questions: Whose voices are not represented on the design team but might be important for the conceptualization of the project? Who is being envisioned as the ideal user? How could additional perspectives be accommodated, even those considered marginal? Whose perspectives have been excluded from the

categorization schema? For example, collecting gender in female/male buckets excludes transgender, gender-fluid and two-spirit people.

Design Output Questions: Can the artifact communicate the subject positions of the researcher(s) and designer(s) in a transparent way? Whose view of the world does the visualization represent? Can the visualization communicate whose voices are missing? Could perspective-taking be a useful strategy to consider for multiple views on the data?

3.3 Examine Power and Aspire to Empowerment

Historically, women and other marginalized groups have experienced the negative effects of hierarchical structures of power. Feminist approaches seek to overturn these hierarchies by promoting horizontal systems of knowledge transmission. Such systems insist on a two-way relation between subject and object of knowledge [36, 39]. A feminist approach to data visualization therefore acknowledges the user as a source of knowledge in the design as well as the reception of any visual interface. The creation of knowledge is, after all, always a shared endeavor.

Following from this point is a related principle: that users are bound to the communities that shape them. Aspiring to empowerment, then, may involve designing for and evaluating the success of a visualization at the scale of the community rather than the individual user. This reorientation can help us to acknowledge the communities who provide us with our design challenges, while also ensuring that the outcomes of our design research connect back to the communities that first made them possible. It can also help us to listen to community concerns and co-design visualizations to advance their goals, while building capacity to achieve them within the community.

Design Process Questions: How is power distributed across the design team? Whose voice matters more and why? How can end users' voices be more fully integrated into the design process? Can we build capacity in user communities, or enlarge our internal perspectives, by employing a more participatory design process?

Design Output Questions: Can the visualization empower the end user and/or her community, group, or organization? When do values often assumed to be a social good, such as "choice," "openness," or "access," result in disempowerment instead?

3.4 Consider Context

A central premise of feminist theory is that all knowledge is *situated* [36], where "situated" refers to the particular social, cultural, and material context in which that knowledge is produced [33]. A feminist approach to data visualization must therefore consider how diverse contexts can influence the production of a visualization, and think through the various ways in which any particular visualization output might be received. In the context of an Enlightenment model of knowledge production, in which additional information leads to increased understanding, a model that allows for the user to "drill down" to more information might be the logical solution for the display of an information system; but this is not the most appropriate choice for more horizontal knowledge frameworks, or those premised on exchange. As another example, consider standard practices of data cleaning. As designers, we often require "clean" data to construct our visualizations. Loukissas argues that local context is lost when we homogenize data [55]. An awareness of what we can learn from local context may yield richer and more informative visualization designs.

Design Process Questions: How can we leverage human-centered design [14] and participatory design [72] methods to learn about and with our end users, including learning more about their culture, history, circumstances, and worldviews? How can we let these insights shape our design practice and change our notions about what constitutes "good" information design?

Design Output Questions: What kinds of terminology, symbols, and cultural artifacts have meaning to end users, and how can we incorporate those into our designs? What might we learn if we were to visualize "messy" data [68]? How do we take context into account in the assessment of visualizations?

3.5 Legitimize Embodiment and Affect

Feminist theory recognizes embodied and affective experiences – that is, experiences that derive from sensation and emotion – as ways of knowing on par with more quantitative methods of knowing and experiencing the world [13]. By definition, visualization rests on the production and assimilation of visual knowledge. But even the most efficiency-oriented and task-driven visualizations have embodied and affective impact, if only to communicate their utility, economy, and purposefulness by way of the visual domain. With the rise of popular forms of visualization such as *data journalism*, designers have begun to intentionally leverage affect in order to create an emotional bond with a story or issue [11], or to engage and impress readers with beauty and complexity [61]. These affective dimensions of visualization have been under-explored in traditional visualization research. Acknowledging the importance of embodiment and affect also has implications for how we evaluate visualizations. Not simply about accomplishing a particular task, could we include measures of embodied and affective responses to visualizations as indicators of their effectiveness?

Design Process Questions: How can we leverage embodied and affective experience to enhance visualization design and engage users? What kinds of expertise might we need on our design team in order to do that? (e.g. fine art, graphic design, animation, or communication specialists)

Design Output Questions: What kinds of embodied and affective experience has meaning to end users? Should we consider tactile, experiential, or social ways of accessing the data visualization? Can we consider visualization outputs in an expanded field, such as data murals [7], data sculptures [1], public walks [2], quilts [48] and installations [63]?

3.6 Make Labor Visible

Information design processes often start with data, but a feminist approach would insist that they begin by working backwards to surface the actors (both individual and institutional) that have labored to generate a particular dataset. Starting with questions of data provenance helps to credit the bodies that make visualization possible – the bodies that collect the data, that digitize them, that clean them, and that maintain them. However, most data provenance research focuses on technical rather than human points of origination and integration [66]. With its emphasis on under-valued forms of labor, a feminist approach to visualization can help to render visible the bodies that shape and care for data at every stage of the process. This relates to the concept of *provenance rhetoric* [44] in which authors of narrative visualizations cite data sources and methods which may help build credibility with the audience. Making labor visible also has implications for fair attribution and credit for the resulting artifact, especially in light of the fact that women and other underrepresented groups have been notoriously excluded from sharing in credit for scientific work [69].

Design Process Questions: Can the team work backwards from the given data to document their provenance and talk to their caregivers? Has the team discussed roles, responsibilities, and credit in advance of publication?

Design Output Questions: Is it feasible to provide a metadata visualization that shows the provenance of the data and their stakeholders (caregivers) at each step? Have we properly attributed work on the project?

4 Conclusion and Next Steps

In this paper, we have outlined six principles for feminist data visualization: *Rethink Binaries*, *Embrace Pluralism*, *Examine Power and Aspire to Empowerment*, *Consider Context*, *Legitimize Embodiment and Affect*

and *Make Labor Visible*. These are preliminary and offered for the purposes of beginning a dialogue about how the digital humanities and information visualization communities can productively exchange theories, concepts, and methods. Applying humanistic theories to design processes and artifacts may be new territory for many humanists, just as grappling with questions of subjectivity, power, and oppression may be new territory for many visualization researchers. As data visualization becomes a mainstream technique for making meaning and creating stories about the world [70], questions of inclusion, authorship, framing [44], reception, and social impact will become increasingly important. In this regard, the humanities and specifically feminist theory have much to offer.

Acknowledgments

The authors wish to thank Patsy Baudoin as an early supporter and discussion partner in this work.

References

[1] Data Physicalization - Data Physicalization Wiki. http://dataphys.org/. Accessed: 2016-08-15.

[2] Boston Coastline: Future Past. https://elab.emerson.edu/projects/data-and-art/boston-coastline-future-past, June 2015. Accessed: 2016-08-15.

[3] M. Balaam, J. Robertson, G. Fitzpatrick, R. Say, G. Hayes, M. Mazmanian, and B. Parmar. Motherhood and HCI. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pp. 3215–3218. ACM, New York, NY, USA, 2013. doi: 10.1145/2468356.2479650

[4] K. Barad. *Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning*. Duke University Press, 2007.

[5] S. Bardzell and J. Bardzell. Towards a Feminist HCI Methodology: Social Science, Feminism, and HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 675–684. ACM, New York, NY, USA, 2011. doi: 10.1145/1978942.1979041

[6] J. Bennett. *Vibrant Matter: A Political Ecology of Things*. Duke University Press, Durham, 2009.

[7] R. Bhargava, R. Kadouaki, G. Castro, E. Bhargava, and C. D'Ignazio. Data Murals: Using the Arts to Build Data Literacy. *Journal of Community Informatics*, 12, 2016.

[8] S. Bordo. *The Flight to Objectivity: Essays on Cartesianism and Culture*. SUNY Press, 1987.

[9] R. Boylorn, B. Cooper, S. Morris, E. Pandit, S. Davis-Faulkner, Crunkista, Chanel, and R. Raimist. Crunk Feminist Collective, 2016.

[10] S. Brown, P. Clements, and I. Grundy. Orlando: Women's Writing in the British Isles from the Beginnings to the Present, 2016.

[11] A. Cairo. Emotional Data Visualization: Periscopic's "U.S. Gun Deaths" and the Challenge of Uncertainty. http://www.peachpit.com/articles/article.aspx?p=2036558, Apr. 2013. Accessed: 2016-08-15.

[12] S. K. Card, J. Mackinlay, and B. Shneiderman, eds. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, San Francisco, 1999.

[13] P. T. Clough and J. Halley, eds. *The Affective Turn: Theorizing the Social*. Duke University Press, Durham, 2007.

[14] M. Cooley. Human-centered design. *Information design*, pp. 59–81, 2000.

[15] C. Coopmans, J. Vertesi, M. Lynch, and S. Woolgar, eds. *Representation in Scientific Practice Revisited*. MIT Press, Cambridge, 2014.

[16] J. M. Corbett, L. Chaplin, and G. R. Gibson. Indigenous Mapping. *International Encyclopaedia of Human Geography*, 1:377–382, 2009.

[17] J. W. Crampton. *Mapping: A Critical Introduction to Cartography and GIS*. Wiley-Blackwell, Malden, Mass, 2010.

[18] J. W. Crampton and J. Krygier. An Introduction to Critical Cartography. *ACME: An International Journal for Critical Geographies*, 4(1):11–33, 2005.

[19] K. Crenshaw. Demarginalizing the Intersection of Race and Sex: A Black Feminist Critique of Antidiscrimination Doctrine, Feminist Theory and Antiracist Politics. *University of Chicago Legal Forum*, pp. 139–167, 1989.

[20] C. D'Ignazio. Art and cartography. *International Encyclopedia of Human Geography*, 1:190–206, 2009.

[21] C. D'Ignazio. What Would Feminist Data Visualization Look Like? https://civic.mit.edu/feminist-data-visualization, Dec. 2015. Accessed: 2016-08-15.

[22] C. D'Ignazio, A. Hope, B. Michelson, R. Churchill, and E. Zuckerman. A Feminist HCI Approach to Designing Postpartum Technologies: "When I First Saw a Breast Pump I Was Wondering if It Was a Joke". In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 2612–2622. ACM, New York, NY, USA, 2016. doi: 10.1145/2858036.2858460

[23] J. P. Dimond, M. Dye, D. Larose, and A. S. Bruckman. Hollaback!: The Role of Storytelling Online in a Social Movement Organization. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pp. 477–490. ACM, New York, NY, USA, 2013. doi: 10.1145/2441776.2441831

[24] J. P. Djajadiningrat, W. W. Gaver, and J. W. Fres. Interaction Relabelling and Extreme Characters: Methods for Exploring Aesthetic Interactions. In *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '00, pp. 66–71. ACM, New York, NY, USA, 2000. doi: 10.1145/347642.347664

[25] J. Drucker. -empyre- :: View topic - Post #1: Background frameworks. http://empyre.library.cornell.edu/phpBB2/viewtopic.php?t=1255, July 2016. Accessed: 2016-08-15.

[26] M. Drk, P. Feng, C. Collins, and S. Carpendale. Critical InfoVis: Exploring the Politics of Visualization. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pp. 2189–2198. ACM, New York, NY, USA, 2013. doi: 10.1145/2468356.2468739

[27] S. Elwood. Critical Issues in Participatory GIS: Deconstructions, Reconstructions, and New Research Directions. *Transactions in GIS*, 10(5):693–708, Sept. 2006. doi: 10.1111/j.1467-9671.2006.01023.x

[28] S. Elwood. Volunteered geographic information: future research directions motivated by critical, participatory, and feminist GIS. *GeoJournal*, (3/4):173, 2008.

[29] A. Fausto-Sterling. *Sexing the Body: Gender Politics and the Construction of Sexuality*. Basic Books, New York, revised ed., 2000.

[30] FemTechNet. FemTechNet. http://femtechnet.org/, 2016. Accessed: 2016-08-15.

[31] J. Flanders, S. Bauman, and A. Clark. Women Writers Project. http://www.wwp.northeastern.edu/, 2016. Accessed: 2016-08-15.

[32] S. Fox, R. R. Ulgado, and D. Rosner. Hacking Culture, Not Devices: Access and Recognition in Feminist Hackerspaces. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, pp. 56–68. ACM, New York, NY, USA, 2015. doi: 10.1145/2675133.2675223

[33] C. Geertz. *Local Knowledge: Further Essays in Interpretive Anthropology*. Basic Books, New York, 2000.

[34] E. A. Grosz. *Volatile Bodies: Toward a Corporeal Feminism*. Indiana University Press, Bloomington, 1994.

[35] P. A. Hall. Critical Visualization. In *Design and the Elastic Mind*. Museum of Modern Art, New York, 2008.

[36] D. Haraway. Situated Knowledges: The Science Question in Feminism and the Privilege of Partial Perspective. *Feminist Studies*, 14(3):575–599, 1988. doi: 10.2307/3178066

[37] D. Haraway. *Simians, Cyborgs, and Women: The Reinvention of Nature*. Routledge, New York, 1991.

[38] S. G. Harding. *The Science Question in Feminism*. Cornell University Press, Ithaca, 1986.

[39] S. G. Harding. *Whose Science? Whose Knowledge?: Thinking from Women's Lives*. Cornell University Press, New York, 1991.

[40] J. Harley. Maps, knowledge, and power. In *The iconography of landscape: Essays on the symbolic representation, design and use of past environments*, pp. 275–312. Cambridge Univ. Press, 1988.

[41] J. B. Harley. Deconstructing the Map. *Cartographica*, 26(2):1, 1989.

[42] L. M. Harris. Deconstructing the Map after 25 Years: Furthering Engagements with Social Theory. *Cartographica*, 50(1):50–53, 2015. doi: 10.3138/carto.50.1.10

[43] S. J. Hekman. *Gender and Knowledge: Elements of a Postmodern Feminism*. Northeastern University Press, Boston, 1990.

[44] J. Hullman and N. Diakopoulos. Visualization Rhetoric: Framing Effects in Narrative Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2231–2240, Dec. 2011. doi: 10.1109/TVCG.2011.255

[45] S. Jánicke, G. Franzini, M. F. Cheema, and G. Scheuermann. On Close and Distant Reading in Digital Humanities: A Survey and Future Challenges. In R. Borgo, F. Ganovelli, and I. Viola, eds., *Eurographics Conference on Visualization (EuroVis) - STARs*. The Eurographics Association, 2015. doi: 10.2312/eurovisstar.20151113

[46] M. Jockers. *Macroanalysis: Digital Methods and Literary History*. University of Illinois Press, Urbana-Champaign, 2013.

[47] E. F. Keller. *Reflections on Gender and Science: Tenth Anniversary Paperback Edition*. Yale University Press, New Haven, anniversary ed., 1996.

[48] L. Klein. Floor Charts on the Floor Screen. `https://medium.com/genres-of-scholarly-knowledge-production/visualization-as-argument-and-on-the-floor-736bb8859cf`, Jan. 2015. Accessed: 2016-08-15.

[49] L. Klein, A. Vujic, and S. Kucheryavykh. Georgia Tech Digital Humanities Lab: Elizabeth Peabody Visualization Project, 2016.

[50] K. B. Knight. Danger, Jane Roe! Embodied Data Visualization as Feminist Critique.. In *Modern Language Association - MLA*, Jan. 2016.

[51] L. Kurgan. *Close up at a distance: Mapping, technology, and politics*. MIT Press, Cambridge, 2013.

[52] M.-P. Kwan. Feminist Visualization: Re-envisioning GIS as a Method in Feminist Geographic Research. *Annals of the Association of American Geographers*, 92(4):645–661, Dec. 2002. doi: 10.1111/1467-8306.00309

[53] C. Landström. Queering feminist technology studies. *Feminist Theory*, 8(1):7–26, 2007. doi: 10.1177/1464700107074193

[54] G. Lloyd. *The Man of Reason: "Male" and "Female" in Western Philosophy*. Routledge, New York, 2002.

[55] Y. A. Loukissas. Taking big data apart: local readings of composite media collections. *Information, Communication & Society*, 0(0):1–14, 2016. doi: 10.1080/1369118X.2016.1211722

[56] C. Malabou and M. Jeannerod. *What Should We Do with Our Brain?* Fordham University Press, New York, 2008.

[57] N. McCurdy, J. Lein, K. Coles, and M. Meyer. Poemage: Visualizing the Sonic Topology of a Poem. In *Proceedings of InfoVis 2015*, pp. 439–448, Jan. 2016.

[58] T. McPherson, E. Loyer, C. Dietrich, S. Anderson, and P. Ethington. *Scalar*. Alliance for Networking Visual Culture, 2016.

[59] E. Meeks and K. Grossner. *Topotime*. Stanford University Libraries, 2013.

[60] C. Neustaedter and P. Sengers. Autobiographical Design in HCI Research: Designing and Learning Through Use-it-yourself. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, pp. 514–523. ACM, New York, NY, USA, 2012. doi: 10.1145/2317956.2318034

[61] A. Parlapiano and J. Ashkenas. How the Recession Reshaped the Economy, in 255 Charts. *The New York Times*, June 2014.

[62] M. Pavlovskaya. Feminism, Maps and GIS A2 - Kitchin, Rob. In N. Thrift, ed., *International Encyclopedia of Human Geography*, pp. 37–43. Elsevier, Oxford, 2009.

[63] L. Perovich. Big bar chart. `http://www.lauraperovich.com/projects/bigbarchart.html`, 2014. Accessed: 2016-08-15.

[64] J. Pickles. *Ground truth : the social implications of geographic information systems*. Mappings. Guilford Press, New York, 1995.

[65] M. Posner. Whats Next: The Radical, Unrealized Potential of Digital Humanities. In M. K. Gold and L. F. Klein, eds., *Debates in the Digital Humanities 2016*. University of Minnesota Press, Minneapolis, 2016.

[66] E. D. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):31–40, Jan. 2016. doi: 10.1109/TVCG.2015.2467551

[67] J. Rajko, E. Standley, J. Wernimont, M. Krzyzaniak, T. ODonnell, D. Feinberg, J. Akerly, S. Rajko, J. Sayers, J. Gervais, N. Belojevic, and J. Winterton. Vibrant Lives. `https://vibrantlives.live/`, 2016. Accessed: 2016-08-15.

[68] K. Rawson and T. Munoz. Against Cleaning, July 2016.

[69] A. Sayre. *Rosalind Franklin and DNA*. Norton, New York, 1975.

[70] E. Segel and J. Heer. Narrative Visualization: Telling Stories with Data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, Nov. 2010. doi: 10.1109/TVCG.2010.179

[71] N. Smith. History and philosophy of geography: real wars, theory wars. *Progress in human geography*, 16(2):257–271, 1992.

[72] C. Spinuzzzi. The Methodology of Participatory Design. *Technical Communication*, 52(2):163–174, May 2005.

[73] M. Stephens. Gender and the GeoWeb: divisions in the production of user-generated cartographic information. *GeoJournal*, 78(6):981–996, Aug. 2013. doi: 10.1007/s10708-013-9492-z

[74] B. Subramaniam. Moored metamorphoses: A retrospective essay on feminist science studies. *Signs*, 34(4):951–980, 2009. doi: 10.1086/597147

[75] L. Suchman. Supporting articulation work: aspects of a feminist practice of technology production. In *IFIP Transactions A (Computer Science and Technology)*, vol. A-57, pp. 7–21, Jan. 1994. A-57.

[76] L. Suchman. Located accountabilities in technology production. *Scandinavian journal of information systems*, 14(2):7, 2002.

[77] J. Wajcman. Feminist theories of technology. *Cambridge Journal of Economics*, 34(1):143–152, 2010. doi: 10.1093/cje/ben057

[78] J. Wernimont and E. Losh. Feminist digital humanities: Theoretical, social, and material engagements around making and breaking computational media, June 2016.

[79] J. Wernimont, J. Rajko, E. Standley, S. Rajko, and M. Krzyzaniak. Vibrant Lives presents: The Living Net, June 2016.

[80] E. A. Wilson. *Gut Feminism*. Duke University Press, Durham, 2015.

[81] D. Wood. *Rethinking the Power of Maps*. Guilford Press, 2010.

[82] D. Wood and J. Fels. *The power of maps*. Guilford Press, 1992.

Seven ways humanists are using computers to understand text.

tedunderwood.com (https://tedunderwood.com/2015/06/04/seven-ways-humanists-are-using-computers-to-understand-text/) · by tedunderwood · June 4, 2015

[This is an updated version of a blog post I wrote three years ago, (https://tedunderwood.com/2012/08/14/where-to-start-with-text-mining/) which organized introductory resources for a workshop. Getting ready for another workshop this summer, I glanced back at the old post and realized it's out of date, because we've collectively covered a lot of ground in three years. Here's an overhaul.]

Why are humanists using computers to understand text at all?
Part of the point of the phrase "digital humanities" is to claim information technology as something that belongs *in* the humanities — not an invader from some other field. And it's true, humanistic interpretation has always had a technological dimension: we organized writing with commonplace books and concordances before we took up keyword search [Nowviskie, 2004; Stallybrass, 2007].

But framing new research opportunities as a specifically humanistic movement called "DH" has the downside of obscuring a bigger picture. Computational methods are transforming the social and natural sciences as much as the humanities, and they're doing so partly by creating new conversations between disciplines. One of the main ways computers are changing the textual humanities is by mediating new connections to social science. The statistical models that help sociologists understand social stratification and social change haven't in the past contributed much to the humanities, because it's been difficult to connect quantitative models to the richer, looser sort of evidence

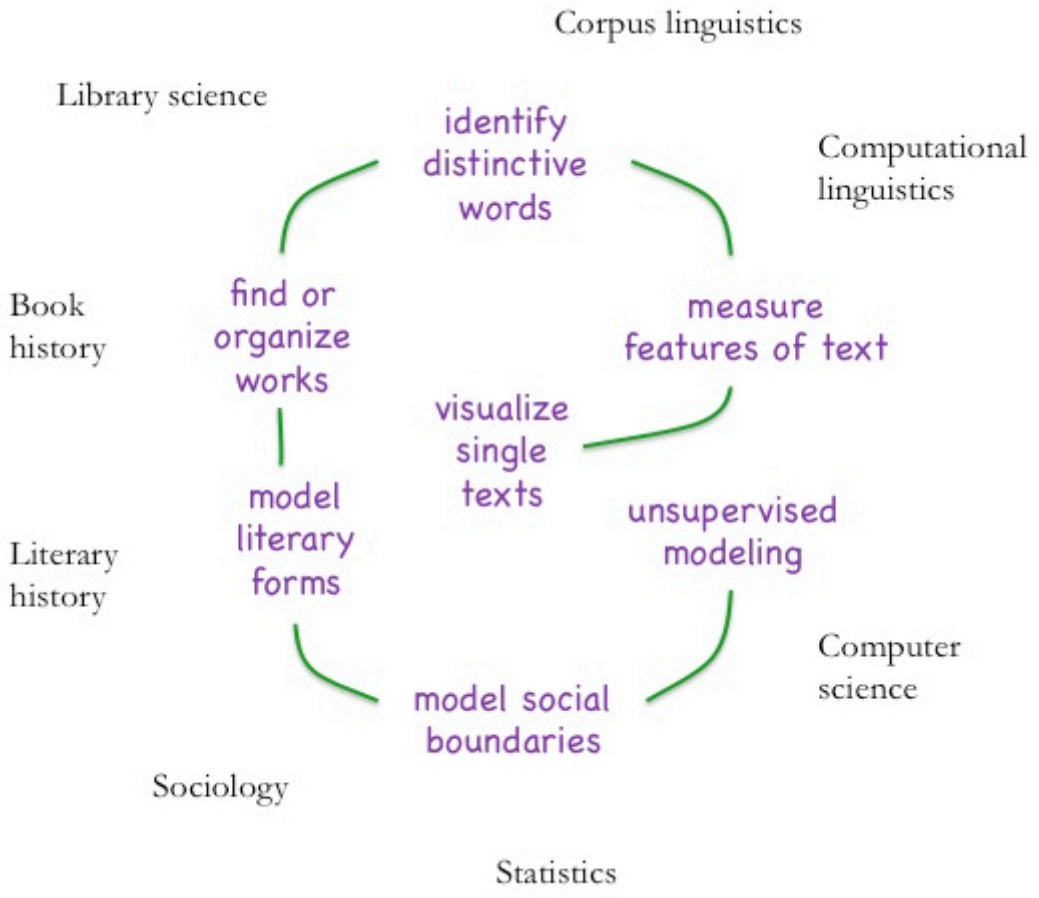
provided by written documents. But that barrier is dissolving. As new methods make it easier to represent unstructured text in a statistical model, a lot of fascinating questions are opening up for social scientists and humanists alike [O'Connor et. al. 2011].
(https://people.cs.umass.edu/~wallach/workshops/nips2011css/papers/OConnor.pdf)

In short, computational analysis of text is not a specific new technology or a subfield of digital humanities; it's a wide-open conversation in the space between several different disciplines. Humanists often approach this conversation hoping to find digital tools that will automate familiar tasks. That's a good place to start: I'll mention tools you could use to create a concordance or a word cloud. And it's fair to stop there. More involved forms of text analysis do start to resemble social science, and humanists are under no obligation to dabble in social science.

But I should also warn you that digital tools are gateway drugs. This thing called "text analysis" or "distant reading" is really an interdisciplinary conversation about methods, and if you get drawn into the conversation, you may find that you want to try a lot of things that aren't packaged yet as tools.

What can we actually do?

The image below is a map of a few things you might do with text (inspired by, though different from, Alan Liu's map of "digital humanities") (https://prezi.com/hjkj8ztj-clv/map-of-digital-humanities/). The idea is to give you a loose sense of how different activities are related to different disciplinary traditions. We'll start in the center, and spiral out; this is just a way to organize discussion, and isn't necessarily meant to suggest a sequential work flow.

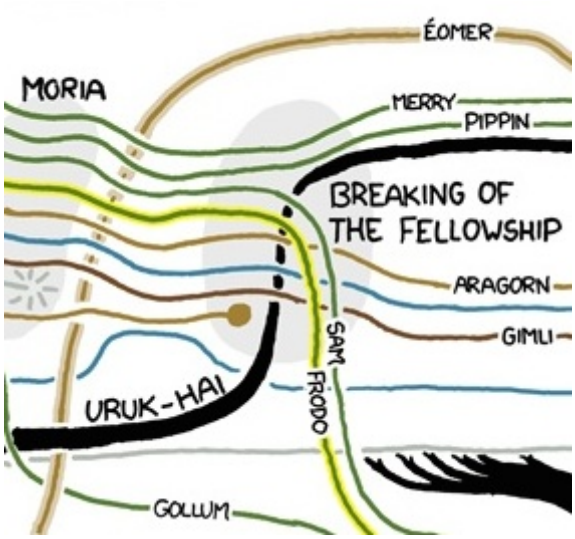


(https://tedunderwood.files.wordpress.com/2015/05/casualmap1.jpg)

1) Visualize single texts.

Text analysis is sometimes represented as part of a "new modesty" in the humanities [Williams] (http://www.chroniclecareers.com/article/The-New-Modesty-in-Literary/150993/). Generally, that's a bizarre notion. Most of the methods described in this post aim to reveal patterns hidden from individual readers — not a particularly modest project. But there are a few forms of analysis that might count as surface readings, because they visualize textual patterns that are open to direct inspection.

For instance, people love cartoons by Randall Munroe that visualize the plots of familiar movies (https://xkcd.com/657/) by showing which characters are together at different points in the narrative.



(https://xkcd.com/657/)

Detail from an xkcd cartoon.

These cartoons reveal little we didn't know. They're fun to explore in part because the narratives being represented *are* familiar: we get to rediscover familiar material in a graphical medium that makes it easy to zoom back and forth between macroscopic patterns and details. Network graphs that connect characters (http://moviegalaxies.com) are fun to explore for a similar reason. It's still a matter of debate what (if anything) they reveal; it's important to keep in mind that fictional networks can behave very differently from real-world social networks [Elson, et al., 2010]. (http://www1.cs.columbia.edu/~delson/pubs/ACL2010-ElsonDamesMcKeown.pdf) But people tend to find them interesting.

A concordance also, in a sense, tells us nothing we couldn't learn by reading on our own. But critics nevertheless find them useful. If you want to make a concordance for a single work (or for that matter a whole library), AntConc is a good tool. (http://www.laurenceanthony.net/software.html)

Visualization strategies themselves (http://textvis.lnu.se) are a topic that could deserve a whole separate discussion.

2) Choose features to represent texts.

A scholar undertaking computational analysis of text needs to answer two questions. First, how are you going to represent texts? Second, what are you going to do with that representation once you've got it? Most what follows will focus on the second question, because there are a lot of equally good answers to the first one — and your answer to the first question doesn't necessarily constrain what you do next.

In practice, texts are often represented simply by counting the various words they contain (they are treated as so-called "bags of words"). Because this representation of text is radically different from readers' sequential experience of language, people tend to be surprised that it works. But the goal of computational analysis is not, after all, to reproduce the modes of understanding readers have already achieved. If we're trying to reveal large-scale patterns that *wouldn't* be evident in ordinary reading, it may not actually be necessary to retrace the syntactic patterns that organize readers' understanding of specific passages. And it turns out that a lot of large-scale questions are registered at the level of word choice: authorship, theme, genre, intended audience, and so on. The popularity of Google's Ngram Viewer (https://books.google.com/ngrams) shows that people often find word frequencies interesting in their own right.

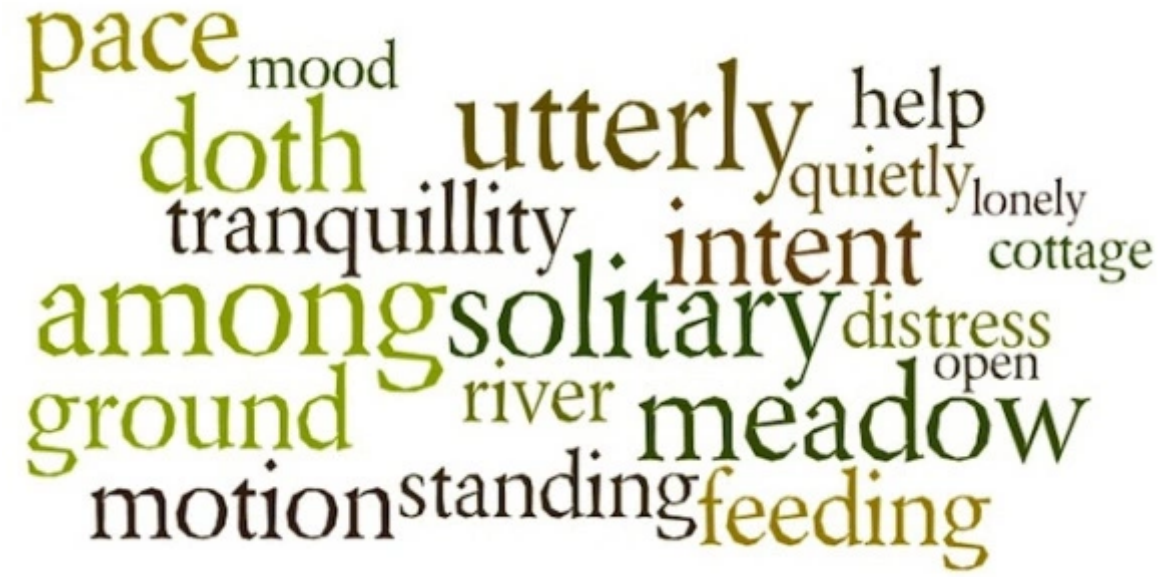
But there are lots of other ways to represent text. You can count two-word phrases, or measure white space if you like. Qualitative information that can't be counted can be represented as a "categorical variable." (http://en.wikipedia.org/wiki/Categorical_variable) It's also possible to consider syntax, if you need to. Computational linguists are getting pretty good at parsing sentences; many of their insights have been packaged accessibly in projects like the Natural Language Toolkit. (http://www.nltk.org) And there will certainly be research questions — involving, for instance, the concept of character (http://www.ark.cs.cmu.edu/literaryCharacter/) — that require syntactic analysis. But they tend not to be questions that are appropriate for people just starting out.

3) Identify distinctive vocabulary.

It can be pretty easy, on the other hand, to produce useful insights on the level of diction. These are claims of a kind that literary scholars have long made: *The Norton Anthology of English Literature* proves that William Wordsworth emblematizes Romantic alienation, for instance, by saying that "the words 'solitary,' 'by one self,' 'alone' sound through his poems" [Greenblatt et. al., 16].

Of course, literary scholars have also learned to be wary of these claims. I guess Wordsworth does write "alone" a lot: but does he really do so more than other writers? "Alone" is a common word. How do we distinguish real insights about diction from specious cherry-picking?

Corpus linguists have developed a number of ways to identify locutions that are really overrepresented in one sample of writing relative to others. One of the most widely used is Dunning's log-likelihood: Ben Schmidt has explained why it works (http://sappingattention.blogspot.com/2011/10/comparing-corpuses-by-word-use.html), and it's easily accessible online through Voyant (http://voyant-tools.org) or downloaded in the AntConc application already mentioned. (http://www.laurenceanthony.net/software.html) So if you have a sample of one author's writing (say Wordsworth), and a reference corpus against which to contrast it (say, a collection of other poetry), it's really pretty straightforward to identify terms that typify Wordsworth relative to the other sample. (There are also other ways to measure overrepresentation; Adam Kilgarriff recommends a Mann-Whitney test. (https://tedunderwood.com/2011/11/09/identifying-the-terms-that-characterize-an-author-or-genre-why-dunnings-may-not-be-the-best-method/)) And in fact there's pretty good evidence that "solitary" is among the words that distinguish Wordsworth from other poets.



(https://tedunderwood.files.wordpress.com/2012/08/wordsworthwordle1.jpg
)

Words that are consistently more common in works by William Wordsworth than in other poets from 1780 to 1850. I've used Wordle's graphics, but the words have been selected by a Mann-Whitney test, which measures overrepresentation relative to a context — not by Wordle's own (context-free) method.

It's also easy to turn results like this into a word cloud — if you want to. People make fun of word clouds, with some justice; they're eye-catching but don't give you a lot of information. I use them in blog posts, because eye-catching, but I wouldn't in an article.

4) Find or organize works.

This rubric is shorthand for the enormous number of different ways we might use information technology to organize collections of written material or orient ourselves in discursive space. Humanists already do this all the time, of course: we rely very heavily on web search, as well as keyword searching in library catalogs and full-text databases.

But our current array of strategies may not necessarily reveal all the things we want to find. This will be obvious to historians, who work extensively with unpublished material. But it's true even for printed books: works of poetry or fiction published before 1960, for instance, are often not tagged as "poetry" or "fiction."

(https://tedunderwood.files.wordpress.com/2015/05/harlemren.jpg)

A detail from Fig 7 in So and Long, "Network Analysis and the Sociology of Modernism."

Even if we believed that the task of simply finding things had been solved, we would still need ways to map or organize these collections. One interesting thread of research over the last few years has involved mapping the concrete social connections that organize literary production. Natalie Houston has mapped connections between Victorian poets and publishing houses; Hoyt Long and Richard Jean So have shown how writers are related by publication in the same journals [Houston 2014; So and Long 2013].

There are of course hundreds of other ways humanists might want to organize their material. Maps are often used to visualize references to places (http://www.nytimes.com/2015/04/14/books/stanford-literary-lab-maps-emotions-in-victorian-london.html?_r=0), or places of publication. (http://viraltexts.org) Another obvious approach is to group works by some measure of textual similarity.

There aren't purpose-built tools to support much of this work. There are tools for building visualizations, (http://gephi.github.io) but often the larger part of the problem is finding, or constructing, the metadata you need.

5) Model literary forms or genres.

Throughout the rest of this post I'll be talking about "modeling"; underselling the centrality of that concept seems to me the main oversight in the 2012 post I'm fixing.

(https://www.flickr.com/photos/zakh/360265041/in/photolist-xQsfe-5S3YW8-8b6fRs-7Etg1A-jtcgmj-9udFwq-79Lh3V-9CG2uW-o963Ew-7xsWd3-cxb32q-7BT2Y2-5S8jAJ-7BT2XV-7BT2Ye-79LmMZ-4GLXxt-4GR9rq-4GLYbr-4GLZjn-4GLYv2-4GR9cy-4GRafN-4GLY36-4GR7Z9-4GLXC8-4GLYqz-4GLXf2-4GR97u-4GRaxU-4GR8y1-4GR9LN-4GLYF2-4GLXt8-4GLXjB-4GRa6A-4GLYK8-4GRaiS-4GR7RG-4GLZ9M-4GR8G1-4GLXMT-7QzymC-4GR9S7-4GR8eU-4GR7V3-4GLZ5K-4GR8WJ-4GLXXg-4GLZBD)

A model treehouse, by Austin and Zak — CC-NC-SA.

A model is a simplified representation of something, and in principle models can be built out of words, balsa wood, or anything you like. In practice, in the social sciences, *statistical* models are often equations that describe the probability of an association between variables. Often the "response variable" is the thing you're trying to understand (literary form, voting behavior, or what have you), and the "predictor variables" are things you suspect might help explain or predict it.

This isn't the only way to approach text analysis; historically, humanists have tended to begin instead by first choosing some aspect of text to measure, and then launching an argument about the significance of the thing they measured. I've done that myself, and it can work. But social scientists prefer to tackle problems the other way around: first identify a concept that you're trying to understand, and then try to model it. There's something to be said for their bizarrely systematic approach.

Building a model can help humanists in a number of ways. Classically, social scientists model concepts in order to understand them better. If you're trying to understand the difference between two genres or forms, building a model could help identify the features that distinguish them.

The tension between modeling-to-explain and modeling-to-predict has been discussed at length in other disciplines [Shmueli, 2010]. (https://www.stat.berkeley.edu/~aldous/157/Papers/shmueli.pdf) But statistical models haven't been used extensively in historical research yet, and humanists may well find ways to use them that aren't common in other disciplines. For instance, once we have a model of a phenomenon, we may want to ask questions about the diachronic stability of the pattern we're modeling. (Does a model trained to recognize this genre in one decade make equally good predictions about the next?)

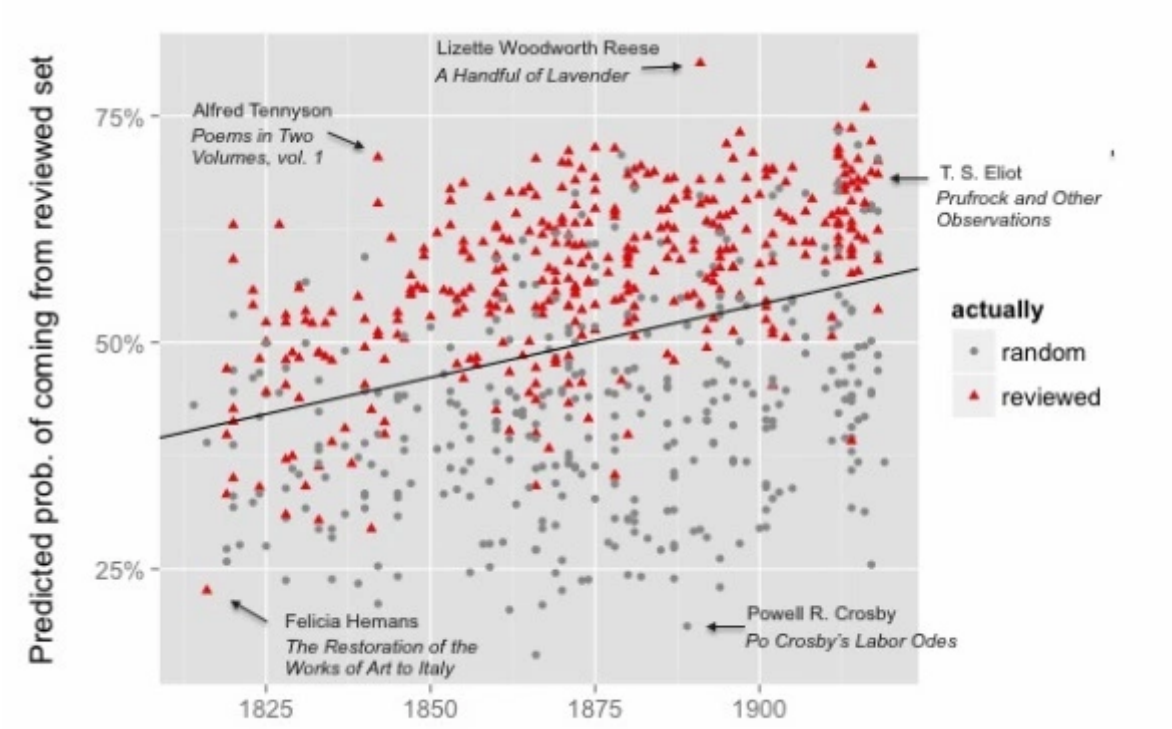
There are lots of software packages (http://www.ats.ucla.edu/stat/r/dae/logit.htm) that can help you infer models of your data. (http://scikit-learn.org/stable/tutorial/statistical_inference/supervised_learning.html) But assessing the validity and appropriateness of a model is a trickier business. It's important to fully understand the methods we're borrowing, and that's likely to require a bit of background reading. One might start by understanding the assumptions implicit in simple linear models, (http://onlinestatbook.com/2/regression/intro.html) and work up to the more complex models produced by machine learning algorithms [Sculley and Pasanek

2008] (http://llc.oxfordjournals.org/content/23/4/409.full.pdf?keytype=ref&ijkey=z3J27nmCeJkZuTz). In particular, it's important to learn something about the problem of "overfitting." (http://www.willamette.edu/~gorr/classes/cs449/overfitting.html) Part of the reason statistical models are becoming more useful in the humanities is that new methods make it possible to use hundreds or thousands of variables, which in turn makes it possible to represent unstructured text (those bags of words tend to contain a lot of variables). But large numbers of variables raise the risk of "overfitting" your data, and you'll need to know how to avoid that. (http://www.quora.com/What-are-ways-to-prevent-over-fitting-your-training-set-data)

6) Model social boundaries.

There's no reason why statistical models of text need to be restricted to questions of genre and form. Texts are also involved in all kinds of social transactions, and those social contexts are often legible in the text itself.

For instance, Jordan Sellers and I have recently been studying the history of literary distinction (https://tedunderwood.com/2015/05/18/how-quickly-do-literary-standards-change/) by training models to distinguish poetry reviewed in elite periodicals from a random selection of volumes drawn from a digital library. There are a lot of things we might learn by doing this, but the top-line result is that the implicit standards distinguishing elite poetic discourse turn out to be relatively stable across a century.



(https://tedunderwood.files.wordpress.com/2015/05/plotmainmodelannotate.jpeg)Similar questions could be framed about political or legal history. (https://people.cs.umass.edu/~wallach/workshops/nips2011css/papers/OConnor.pdf)

7) Unsupervised modeling.

The models we've discussed so far are *supervised* in the sense that they have an explicit goal. You already know (say) which novels got reviewed in prominent periodicals, and which didn't; you're training a model in order to discover whether there are any patterns in the texts themselves that might help us explain this social boundary, or trace its history.

But advances in machine learning have also made it possible to train *unsupervised* models. Here you start with an unlabeled collection of texts; you ask a learning algorithm to organize the collection by finding clusters or patterns of some loosely specified kind. You don't necessarily know what patterns will emerge.

If this sounds epistemologically risky, you're not wrong. Since the hermeneutic circle doesn't allow us to get something for nothing, unsupervised modeling does inevitably involve a lot of (explicit) assumptions. It can nevertheless be extremely useful as an exploratory heuristic, and sometimes as a foundation for argument. A family of unsupervised algorithms called "topic modeling" (https://tedunderwood.com/2012/04/07/topic-modeling-made-just-simple-enough/) have attracted a lot of attention in the last few years, from both social scientists (http://www.sciencedirect.com/science/journal/0304422X/41/6) and humanists. (http://journalofdigitalhumanities.org/2-1/) Robert K. Nelson has used topic modeling, for instance, to identify patterns of publication in a Civil-War-era newspaper from Richmond.
(http://dsl.richmond.edu/dispatch/pages/home)

(http://dsl.richmond.edu/dispatch/Topics)
But I'm putting unsupervised models at the end of this list because they may almost be too seductive. Topic modeling is perfectly designed for workshops and demonstrations, since you don't have to start with a specific research question. A group of people with different interests can just pour a collection of texts into the computer, gather round, and see what patterns emerge. Generally, interesting patterns do emerge: topic modeling can be a powerful tool for discovery. But it would be a mistake to take this workflow as paradigmatic for text analysis. Usually researchers begin with specific research questions, and for that reason I suspect we're often going to prefer supervised models.

* * *

In short, there are a lot of new things humanists can do with text, ranging from new versions of things we've always done (make literary arguments about diction), to modeling experiments that take us fairly deep into the methodological terrain of the social sciences. Some of these projects can be crystallized in a push-button "tool," but some of the more ambitious projects require a little familiarity with a data-analysis environment like Rstudio (http://www.rstudio.com), or even a programming language like Python, (https://www.python.org) and more importantly with the assumptions underpinning quantitative social science. For that reason, I don't expect these methods to become universally diffused in the humanities any time soon. In principle, everything above is accessible for undergraduates, with a semester or two of preparation — but it's not preparation of a kind that English or History majors are guaranteed to have.

Generally I leave blog posts undisturbed after posting them, to document what happened when. But things are changing rapidly, and it's a lot of work to completely overhaul a survey post like this every few years, so in this one case I may keep tinkering and adding stuff as time passes. I'll flag my edits with a date in square brackets.

SELECTED BIBLIOGRAPHY

Elson, D. K., N. Dames, and K. R. McKeown. "Extracting Social Networks from Literary Fiction." (http://www1.cs.columbia.edu/~delson/pubs/ACL2010-ElsonDamesMcKeown.pdf) Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden, 2010. 138-147.

Greenblatt, Stephen, et. al., *Norton Anthology of English Literature* 8th Edition, vol 2 (New York: WW Norton, 2006.

Houston, Natalie. “Towards a Computational Analysis of Victorian Poetics.” (http://www.jstor.org/stable/10.2979/victorianstudies.56.3.498) *Victorian Studies* 56.3 (Spring 2014): 498-510.

Nowviskie, Bethany. “Speculative Computing: Instruments for Interpretive Scholarship.” (http://nowviskie.org/dissertation.pdf) Ph.D dissertation, University of Virginia, 2004.

O’Connor, Brendan, David Bamman, and Noah Smith, “Computational Text Analysis for Social Science: Model Assumptions and Complexity,” (https://people.cs.umass.edu/~wallach/workshops/nips2011css/papers/OConnor.pdf) NIPS Workshop on Computational Social Science, December 2011.

Piper, Andrew. “Novel Devotions: Conversional Reading, Computational Modeling, and the Modern Novel.” (https://muse.jhu.edu/journals/new_literary_history/toc/nlh.46.1.html) *New Literary History* 46.1 (2015).

Sculley, D., and Bradley M. Pasanek. “Meaning and Mining: The Impact of Implicit Assumptions in Data Mining for the Humanities.” (http://llc.oxfordjournals.org/content/23/4/409.full.pdf?keytype=ref&ijkey=z3J27nmCeJkZuTz) *Literary and Linguistic Computing* 23.4 (2008): 409-24.

Shmueli, Galit. “To Explain or to Predict?” (http://arxiv.org/pdf/1101.0891.pdf) *Statistical Science* 25.3 (2010).

So, Richard Jean, and Hoyt Long, “Network Analysis and the Sociology of Modernism,” (http://boundary2.dukejournals.org/content/40/2/147.abstract) *boundary 2* 40.2 (2013).

Stallybrass, Peter. “Against Thinking.” (http://athanasius.stanford.edu/Readings/Responses.pdf) *PMLA* 122.5 (2007): 1580-1587.

Williams, Jeffrey. “The New Modesty in Literary Criticism.” (http://www.chroniclecareers.com/article/The-New-Modesty-in-Literary/150993/) *Chronicle of Higher Education* January 5, 2015.

tedunderwood.com (https://tedunderwood.com/2015/06/04/seven-ways-humanists-are-using-computers-to-understand-text/) · by tedunderwood · June 4, 2015

Against Cleaning

curatingmenus.org (http://curatingmenus.org/articles/against-cleaning/) · by Katie Rawson · July 6, 2016

- Trevor Muñoz, @trevormunoz (https://twitter.com/trevormunoz)

Practitioners, critics, and popularizers of new methods of data-driven research treat the concept of "data cleaning" as integral to such work without remarking on the oddly domestic image the term makes—as though a corn straw broom were to be incorporated, Rube-Goldberg-like, into the design of the Large Hadron Collider. In reality, "data cleaning" is a consequential step in the research process that we often make opaque by the way we talk about it. The phrase "data cleaning" is a stand in for longer and more precise descriptions of what people are doing in the initial phases of data-intensive research. If you work with data or pay attention to discussions among practitioners who do, you've probably heard or read somewhere that 80 percent of that work is "cleaning".[1] Subsequently you likely realize that, there is not one single understanding of what "data cleaning" means. Many times the specifics of "data cleaning" are not described anywhere but reside in the general professional practices, materials, personal histories, and tools of the researchers. That we employ obscuring language like "data cleaning" should be a strong invitation to scrutinize, perhaps reimagine, and almost certainly rename this part of our practice.

The persistence of an element that is "out of focus" in discussions of data-intensive research does not invalidate the findings of such research, nor is it meant to cast researchers using these methods under suspicion. Rather, the collective acceptance of a connotative term, "cleaning," suggests two

assumptions: first, that researchers in many domains consider the consequences of whatever is done during this little-discussed 80 percent of the process devoted to "cleaning" as sufficiently limited or bounded so as not to threaten the ultimate value of any findings; and second, relatedly, that there is little to be gained from more precise description of those elements of the research process that currently fall under the rubric of "cleaning."

As researchers working in the relatively new domain of data-intensive research in the humanities, we have found that these assumptions do not serve us well. In fields where data intensive work has a longer history, researchers have developed paradigms and practices that *de facto* define "data cleaning." However, in the humanities, these bounds are still unformed. Yet the humanities cannot import paradigms and practices whole from other fields, whether from "technoscience" or the nearer "social" sciences, without risking the foreclosure of specific and valuable humanistic modes of producing knowledge. If we are interested in working with data and we accept that there is something in our work with data that is like what other fields might call "data cleaning," we have no choice but to try to articulate both what it is and what it means in terms of how humanists make knowledge.

This may be a only a current issue, a tax on those humanities researchers who wish to adopt new methods, asking them to over-explain their work processes in order to hash out new regimes for research in this domain. Once new methods are more widely practiced, the data-intensive humanities researcher may also be able to toss off the shorthand of "data cleaning." For now, there is value in being arrested by the obfuscation of this phrase. Trying to more precisely say what we mean by "data cleaning" can be fruitful because this effort directs our attention to an unresolved conversation about data and reductiveness. In turn, this might help us to develop new work that blends the tradition of cultural criticism from the humanities with research that is also digital and data-intensive.

Humanities Data and Suspicions of Reduction

When humanities scholars recoil at data-driven research, they are often responding to the reductiveness inherent in this form of scholarship. This reductiveness can feel intellectually impoverishing to scholars who have spent their careers working through particular kinds of historical and cultural complexity. The modern humanities have invested mental and moral energy into, and reaped insights from, studying difference. Bethany Nowviskie summarizes this tradition in her contribution to the (forthcoming) 2017 edition of *Debates in the Digital Humanities*, "Capacity Through Care." (http://nowviskie.org/2016/capacity-through-care/) Nowviskie writes:

> The finest contribution of the past several decades of humanities research has been to broaden, contextualize, and challenge canonical collections and privileged views. Scholars do this by elevating instances of neglected or alternate lived experience—singular human conditions, often revealed to reflect the mainstream.

From within this worldview, data cleaning is then maligned because it is understood as a step that inscribes a normative order by wiping away what is different. The term "cleaning" implies that a data set is "messy." "Messy" suggests an underlying order. It supposes things already have a rightful place, but they're not in it—like socks on the bedroom floor rather than in the wardrobe or the laundry hamper.

Understood this way, suspicions about "cleaning" are suspicions that researchers are not recognizing or reckoning with the framing orders to which they are subscribing as they make and manipulate their data. In fields where researchers have long been explicit about their framing orders, the limits of results are often understood and articulated using specialized discourses. For example, in climate science, researchers confine their claims to the data they can work with and report results with margins of error.[2] While humanities researchers do have discourses for limiting claims (acknowledging to the choice of an archive or a particular intellectual tradition), the move into data intensive

research asks humanists to modify such discourses or develop new ones suitable for these projects. The ways in which humanities engage these challenges may both open up new practices for other fields and allow humanities researchers who have made powerful critiques of the existing systems of data analysis to undertake data-intensive forms of research in ways that don't require them to abandon their commitments to such critiques.

To contribute to the development of new discourses and the practice of critically-attuned data work, we scrutinize "cleaning" through a reflection on our own work with *Curating Menus*. *Curating Menus* is a research project that aims to curate and analyze the open data from New York Public Library's *What's on the Menu?* (http://menus.nypl.org/).

The Value of a Naive Tool

We set off to answer questions like: can we see the effect of wartime food rationing in what appeared on menus during World War I? or, can we track the changing boundaries of what constituted a "dish" over time? To do this, we thought we needed to "clean" the "messy" data. What became evident was that "cleaning up" or "correcting" values was a misleading—and even unproductive—way to think about how to make the data more useful for our own questions and for other scholars studying food.

Under the rubric of "cleaning," we began with a technical solution to what we'd imagined was a technical problem. Variation in the strings of text transcribed from menus was obscuring our ability to do things as simple as count how many dishes were in the data set. Trevor, along with Lydia Zvygintseva, began trying to reduce the number of variations using Open Refine (http://openrefine.org/). When the scale of the problem overwhelmed the capabilities of that tool, Trevor discovered that it was possible to run the clustering algorithms popularized by Open Refine using custom Python scripts (http://nbviewer.jupyter.org/gist/trevormunoz/8358810). The output of one of these scripts were lists of variant values such as:

```
id
2759        Potatoes, au gratin
7176         Potatoes au Gratin
8373        Potatoes--Au gratin
35728       Potatoes: au gratin
44271        Au Gratin Potatoes
84510      Au Gratin (Potatoes)
94968     Potatoes, au Gratin,
97166      POTATOES:- Au gratin
185040     Au Gratin [potatoes]
313168      Au Gratin  Potatoes
315697     (Potatoes) Au Gratin
325940       Au Gratin Potatoes
330420       au-Gratin Potatoes
353435     Potatoes:  Au gratin
373639       Potatoes Au Gratin
```

We were very excited to get lists that looked this way because we could easily imagine looping over such lists and establishing one normalized value for each set of variants. We hadn't yet recognized that the data model around which the data set was organized was not the data model we needed to answer our research questions. The main challenge seemed to be processing enough values quickly enough to "get on with it."

At this point, the Python scripts we were using were small, purpose-built command line programs. After some deliberation, we decided to build a simple web application to provide the task-specific user interfaces we needed to tackle the challenge of NYPL's menu data.[3]

The piece of software we built does, in some ways, the opposite of what one might expect. A cluster of values like the one for "Potatoes Au Gratin" above is presented to the user, and he or she (Trevor or Katie) have to make a decision about how to turn that cluster of variants into a single value. Our tool sorts the variants by the number of times they appear across the data set. So the decision may be to simply save the most commonly occurring value as the normalized

form: “potatoes au gratin”. Or it might be to modify that value based on a set of rules we have created (more on that later). Or it might be to supply a new value. The process can end up looking like this:

… What would be the authoritative spelling of Buzzards Bay oysters? Let me Google that.

… Oh, it collapsed an orange juice and an orange sherbet under “orange”; let me flag that.

… A jelly omelet!?

The tool surfaces near-matches, but it does not automate the work of collapsing them into normalized values. Instead, it focuses one’s attention and labor on exactly that activity. In our initial version of computer-assisted data curation, you still have to touch each data point.

In the process of normalizing values, we found ourselves faced with questions about the foods themselves. Choosing the “correct” string was not a self-contained problem, but an issue that required returning to our research questions and investigating the foods themselves. Since the research questions we were asking required us to maintain information about brands and places, we often had to look up items to see which was the correct spelling of the brand or place name.[4] Shellfish and liquor were two particularly interesting and plagued areas for these questions. The process revealed kinds of “messiness” that we had not yet even considered. We realized the data points we were making were not “corrections” “cleaning up” the original data set, but their rather formed an additional contribution of information with its own data model. What we were developing was not an updated version of the NYPL’s data set. What we thought were data points were, in fact, a synthesis or mash-up of different kinds of information within a half-finished or half-articulated data model. We needed to create our own data set, which would work in context with the NYPL data set.

This approach is made possible by and explicitly rests on a structure of linked data. The NYPL data was created to be linkable—it uses unique URLs for dishes and menus. Our data can include links referencing these URLs. The linked data paradigm thus encourages us to build our own data and to (inter-)link it with the original.

Diversity in Data

The Curating Menus data set is an organized hierarchy of concepts from the domain of food. To make it, we attach labels that we believe will best facilitate researchers' understanding of the scope, diversity, and value of the NYPL's data set for research. Our interaction with the NYPL data set became a process of evaluating variants. Which variants in the names of dishes revealed new information we should account for in our own data, and which variants were simply accidents of transcription or typesetting? The process freed us to attend to difference and detail rather than only attempting to hide it or clean it away. We can be sensitive to questions about time, place, and subject. This kind of attention is imperative if humanities researchers are to find the menus data legible.

As we considered methods for preserving diversity within our large data set, the work of anthropologist Anna Tsing offered a valuable theoretical framework to approach these issues. In "On Nonscalability: The Living World is Not Amenable to Precision-Nested Scales," Tsing critiques scalability as an overarching paradigm for organizing systems (whether world trade, scientific research, or colonial economies).[5] By scalability, Tsing means the quality that allows things to be traded out for each other in a totalizing system without regard to the unique or individual qualities of those things—like many stalks of sugarcane (which are biological clones of one another), or, subsequently, workers in a factory. From this definition of scalability, she goes on to argue for a theory of nonscalability. Tsing writes, "The definition of nonscalability is in the negative: scalability is a distinctive design feature; nonscalability refers to everything that is without that feature … Nonscalability theory is an analytic

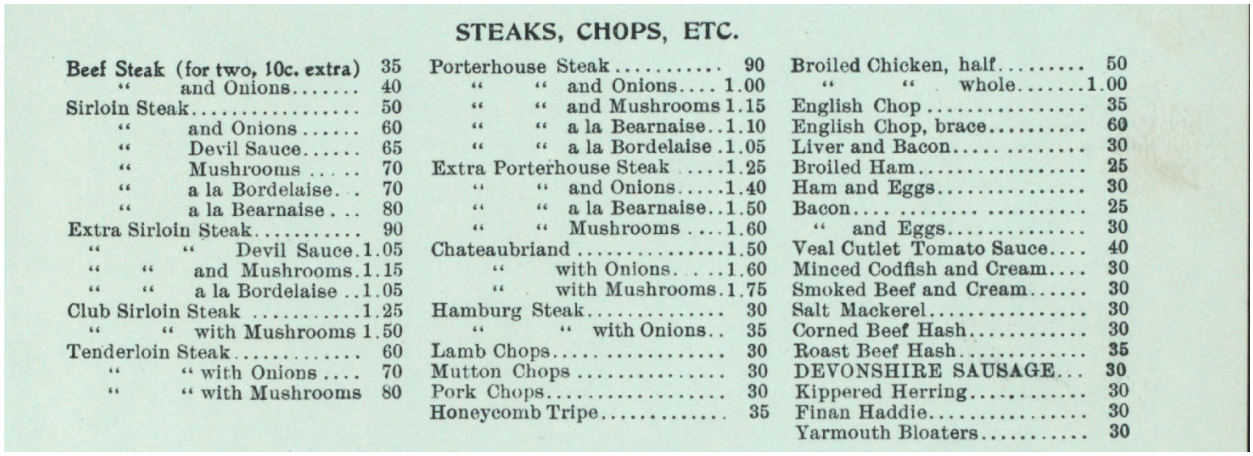
apparatus that helps us notice nonscalable phenomena." While scalable design creates only one relationship between elements of a system (what Tsing calls "precision nesting"), nonscalable phenomena are enmeshed in multiple relationships, outside or in tension with the nesting frame. "Scales jostle and contest each other. Because relationships are encounters across difference, they have a quality of indeterminacy. Relationships are transformative, and one is not sure of the outcome. Thus diversity-in-the-making is always part of the mix."

Currently, the imagination of the cultural heritage world has been captured by crowdsourced information production on the one hand and large-scale institutional aggregation on the other—the *What's On the Menu?* project exemplifies both of these trends. Our difficulties working with the open data from this project suggest that it is a vital moment to consider the virtues of non-scalability theory in relation to digital scholarship. Engineering crowdsourced cultural heritage projects usually involves making object transcription, identification, and the development of metadata scalable. For example, the makers of the *What's On the Menu?* project designed their system to divide the work into parcels that could be done quickly by users while reducing friction that arise from differences in the menus (the organization of the information on the page, other evidence of physical manifestations like handwriting and typeface variations).[6] The images of menus and the metadata about them are also being republished through projects like the Digital Public Library of America (https://dp.la/) (DPLA), another example of how things get shaped and parsed for purposes of scaling up to ever wider distribution. Tsing reminds us, "At best, scalable projects are articulations between scalable and nonscalable elements, in which nonscalable effects can be hidden." She argues that the question is not whether we do or do not engage in scalable or non-scalable processes. To explore the articulations between scalable and nonscalable, Tsing tells the story of the contemporary matsutake industry, which encompasses both foraging (by immigrant harvesters in the ruins of large-scale industrial forestry in the U.S. Pacific Northwest) and global supply chains serving Japanese markets. Tsing's account focuses our attention on how "scales... arise from the relationships that

inform particular projects, scenes, or events" (509). The elements of nonscalable systems enter into "transformative relationships" and these "contact[s] across difference can produce new agendas" (510). Following Tsing, we came to see points of articulation which had previously been invisible to us as would-be consumers of scaled data. Beginning from the creation of the original, physical menus and tracing the development of the crowd-created data, we identify and account for "nonscalable elements"—and consequently, edge further and further from the terminology of "cleaning."

Seeing Nonscalability in NYPL's Crowdsourced Menus Project

Making menus is a scalable process. Although menus are sometimes handwritten or elaborately printed on ribbon-sewn silk, the format of a menu is designed to be scalable. Menus are an efficient typographical vehicle for communicating a set of offerings for often low-margin dining enterprises. Part of the way that we know that menus are scaleable is how alike they appear. "Eggs Benedict" or "caviar", with their accompanying prices may fit interchangeably into the "slots" of the menu's layout. Within the menus themselves, we also see evidence of the nexus of printing scalability, dish scalability, and cost in, for example, the use of ellipses to express different options: eggs with ... cheese, ... ham, ... tomatoes, etc. The visual evidence of *What's on the Menu?* shows us how headings, cover images, epigraphs—for all their surface variations—follow recognizable patterns. These strong genre conventions and the mass production of the menus as physical objects allow us to see and treat them as scaled and scalable.[7]



STEAKS, CHOPS, ETC.

Beef Steak (for two, 10c. extra)	35	Porterhouse Steak	90	Broiled Chicken, half	50
" and Onions	40	" " and Onions	1.00	" " whole	1.00
Sirloin Steak	50	" " and Mushrooms	1.15	English Chop	35
" and Onions	60	" " a la Bearnaise	1.10	English Chop, brace	60
" Devil Sauce	65	" " a la Bordelaise	1.05	Liver and Bacon	30
" Mushrooms	70	Extra Porterhouse Steak	1.25	Broiled Ham	25
" a la Bordelaise	70	" " and Onions	1.40	Ham and Eggs	30
" a la Bearnaise	80	" " a la Bearnaise	1.50	Bacon	25
Extra Sirloin Steak	90	" " Mushrooms	1.60	" and Eggs	30
" " Devil Sauce	1.05	Chateaubriand	1.50	Veal Cutlet Tomato Sauce	40
" " and Mushrooms	1.15	" with Onions	1.60	Minced Codfish and Cream	30
" " a la Bordelaise	1.05	" with Mushrooms	1.75	Smoked Beef and Cream	30
Club Sirloin Steak	1.25	Hamburg Steak	30	Salt Mackerel	30
" " with Mushrooms	1.50	" with Onions	35	Corned Beef Hash	30
Tenderloin Steak	60	Lamb Chops	30	Roast Beef Hash	35
" " with Onions	70	Mutton Chops	30	DEVONSHIRE SAUSAGE	30
" " with Mushrooms	80	Pork Chops	30	Kippered Herring	30
		Honeycomb Tripe	35	Finan Haddie	30
				Yarmouth Bloaters	30

Scalability in the original physical menus.

However, the menus also express nonscalable elements—historical contingencies and encounters across difference. Some of these nonscalable elements are revealed by the kind of questions we find ourselves asking about the experience of ordering from these menus. How were they understood as part of interactions between purveyors, customers, and diners? How did diners navigate elements like the pervasive use of French in the late nineteenth and early twentieth centuries? How did they interpret the particular style and content of cover images or quotations? Evidence for these questions manifests in the menus as objects but does not fit within the scalable frames of menu production nor the menu data we have at hand. The nonscalable elements cannot be disregarded and have the potential to impact how we interpret and handle the scalable data. Nonscalability theory encourages us to grapple with this dynamic at each point of articulation in the process of making scalable objects.

The collection of these menus was also scalable. The system set up for their accession and processing not only treated the menus as interchangeable objects; it also treated them like the many other paper materials that entered the collections of the New York Public Library in the twentieth century. Perhaps the clearest evidence of this is in the cataloging. The catalog cards fit each menu into the same frame—with fields for the dining establishment, the date of creation and date of accession, and the sponsor of the meal if available. Cataloging is a way of suppressing or ignoring the great differences in the menus, choosing one

type of data to attend to. The cards index the collection so that the institution has a record of its holdings and so that a user can find a particular object. The menus, with their scalable and nonscalable features, become scalable library inventory through this process.[8]

Cataloging's aim is to find a way to make items at least interchangeable enough not to break the system. The practice is rife with examples of catalogers navigating encounters with difference. Catalogers practice nonscalability theory constantly. Sometimes the answer is institutionally-specific fields in MARC records (https://en.wikipedia.org/wiki/MARC_standards); sometimes the solution is overhauling subject headings or creating a new way of making records (like the BIBFRAME initiative (https://www.loc.gov/bibframe/)). However, the answer is almost never to treat each object as a unique form; instead the object is to find a way to keep the important and usable information while continuing to use socially and technologically embedded forms of classifying and finding materials.

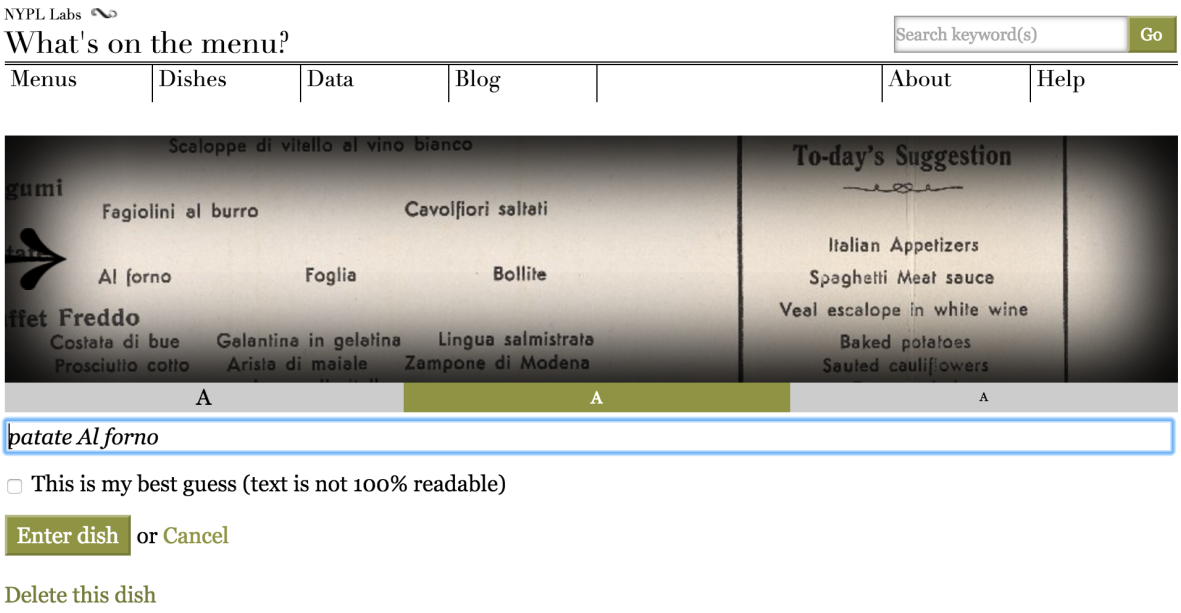
Digitization is also a process designed for scalability. As long as an object can fit into the focal area of an imaging device, its size, texture, and other material features are reduced to a digital image file. The zooming enabled by high-resolution digital images is one of Tsing's prime examples of design and engineering for scalability. In the distribution of digitized images, the properties of the digital surrogate which are suited to scalability are perpetuated, while the properties of the original which are nonscalable (the feel of the paper, its heft or daintiness) are lost.[9]

The point at which certain objects are selected for digitization is one of the moments of articulation Tsing describes between the scalable and nonscalable. Digitization transforms of diverse physical materials—brittle, acidic paper or animal parchment, large wooden covers or handstitched bindings, leaves or inserts—into standardized grids of pixels. From the point of digitization forward, the logic of scalability permeates projects like *What's On the Menu?*. The

transcription platform is constructed to nest precisely within the framework of how cultural heritage organizations like NYPL create digital objects from their original materials.

Paul Beaudoin from the NYPL Labs team discusses some of the logic behind their approach to these kind of projects in a blog post (http://www.nypl.org/blog/2015/11/23/scribe-framework-community-transcription) announcing Scribe, an open-source software platform released in 2015 but derived from the library's experience with crowdsourced transcription projects. Beaudoin describes how the Scribe platform (http://scribeproject.github.io/) is based on "simplification [that] allows us to reduce complex document transcription to a series of smaller decisions that can be tackled individually. ... the atomicity of tasks makes projects less daunting for volunteers to begin and easier to continue." *What's on the Menu?*, for example, presents visitors with a segment of a digitized image of a menu and a single text input box to record what they see.[10] The NYPL Labs team is explicit about its commitments to designing for scalability. We know from work in the domain of scholarly editing that what comprises "transcription" is not self-evident.[11] It could be modeled and implemented in software in a number of ways. The menus project uses Optical Character Recognition (OCR) software to generate bounding boxes that determine what human volunteers will transcribe. In this, we can see the "precision nesting" of scales at work. OCR software is designed to scalably produce machine-readable, machine-processable digital text from images of printed material. In the case of the menus, the software can detect regions of text within the digital images; however, due to the variation in typefaces, the ageing of inks and paper, and other nonscalable elements, the output of the OCR algorithm is not a legible set of words. Using the bounding boxes but discarding the OCR text in favor of text supplied by human volunteers is a clever and elegant design. It constructs the act of transcription in such a way that it matches the scalable process of digitization and ways of understanding the content of a menu that privilege scalable data.

Yet, even here, now that we know to look for them, the nonscalable effects cannot be completely hidden. The controls allow users to zoom through three levels of an image, a feature that evidences slippage in the segmentation algorithm. This element of the tool acknowledges that someone might need to zoom out to complete a transcription—often because the name of a dish has to be understood through the relation between the line of text in the bounding box and other nearby text, like a heading. Further, the text box on the transcription screen is unadorned, implying that what to do is self-evident, but the help page is full of lengthy instructions for how to "navigate some of the more commonly encountered issues," revealing the ways that transcription is not a self-evident, scalable process.



The physical layout of text often requires zooming out the image presented for transcription to decipher a dish.

In addition, the project was designed so that people did not have to create accounts or sign in to submit transcriptions. This reflects a view of volunteers as interchangeable, and embedded in this assumption is the hope that it allows more work to get done more quickly. However, what is is construed as a scalable workforce is, in fact, made up of people who have different levels of understanding or adherence to the guidelines and different perceptions or interpretations of the materials. When we understand this workforce as a

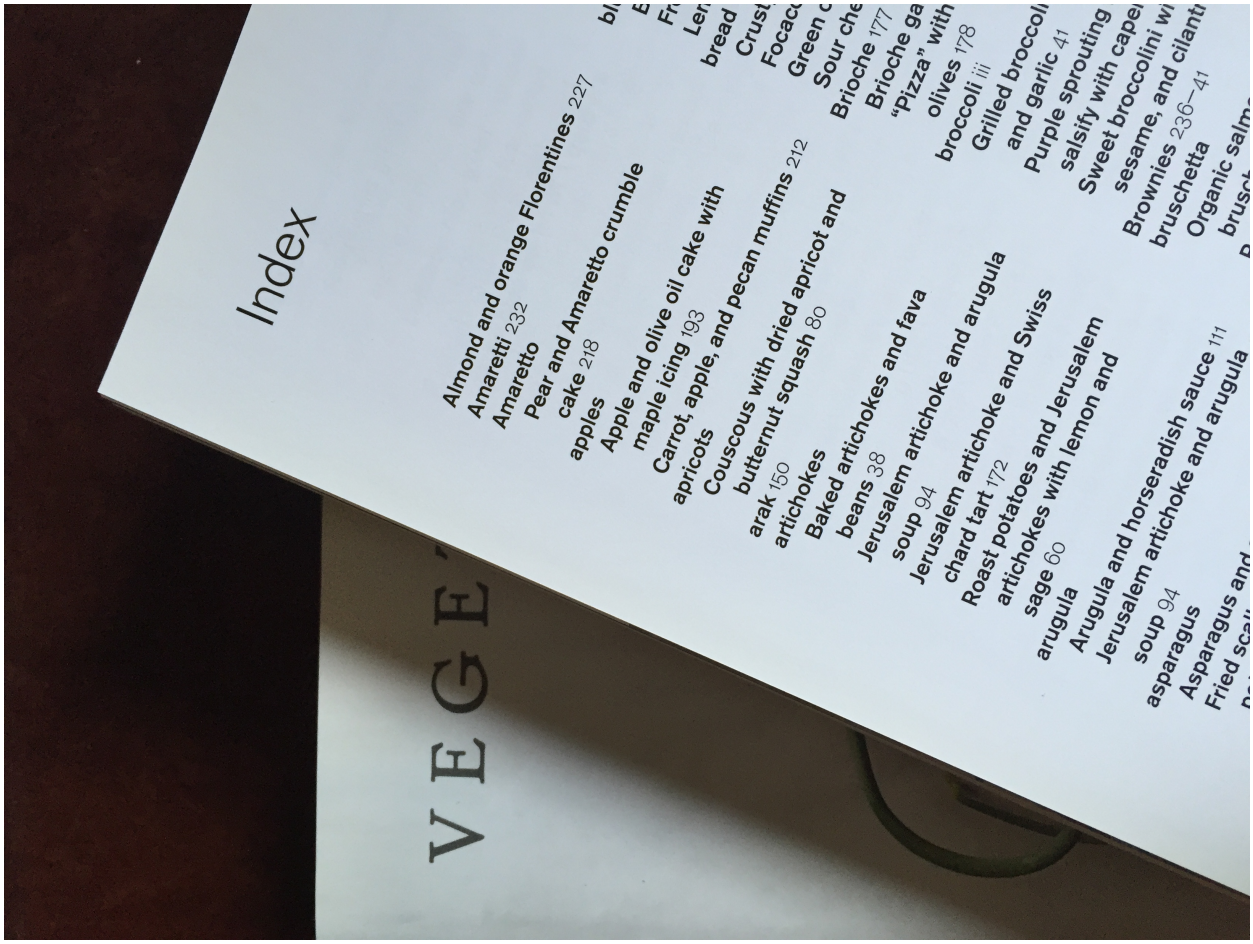
collection of individuals, we can see how any crowd as large as the one that has worked on the menus project will contain such diversity. The analytic apparatus of Tsing's nonscalability theory makes all these design choices visible and allows us to see the transcription task, as framed within *What's on the Menu?*, as another moment of articulation between scalable and nonscalable elements.

When we download the open data from the *What's on the Menu?* site and open up the files, we are presented with the results of all this activity—menu collection and digitization and transcription. Instead of seeing mess, we see the ways in which diversity has seeped or broken into what were designed to be smoothly scaling systems. Now we are better prepared to envision how our work—creating new data organized around concepts of historical food practices—begins from what the NYPL has released, which is transcription data (words volunteers typed into the boxes at *What's on the Menu?* linked to metadata from their digital library systems). In both of these data sets there is something called "dish." In NYPL's data, "dish" is the name of the field in which a transcribed string from a menu is stored in the project's database. In Curating Menus's data, "dish" is a representation created to reflect and name an arrangement of foods and culinary practices in a particular historical moment. This is an example of, as Tsing puts it, the ways that "scales jostle and contest." We know that the response to this friction is not to retreat from working at scale. Instead we have to find ways of working while aware that *precision* nesting hides diversity and that there are stakes to things being hidden.

Indexes: Making Scalability Explicit and Preserving Diversity

Our answer this challenge is an index. We're suggesting that indexing is a more precise replacement for some of the work that travels under the name of "cleaning." An index is an information structure designed to serve as a system of pointers between two bodies of information, one of which is organized to provide access to concepts in the other. The lists of terms and associated page numbers from the back of a book is one familiar example. An array of other terms that people use alongside "cleaning" (wrangling, munging, normalizing,

casting) name other important parts of working with data, but indexing best captures the crucial interplay of scalability and diversity that we are trying to trace in this piece.



A cookbook index

We began to think of the work we were doing as building something like a back-of-the-book index for the *What's On the Menu?* data. We would create additional data structures around and atop the existing data, generating links between a set of names or categories we created and the larger and more heterogeneous set of data from NYPL. We are interested in ultimately building two interconnected indexes, one focused on historical food concepts and one on the organizations connected to the menus (businesses, community organizations, fraternal societies, etc.). We have begun with the food index, and we are developing a framework that echoes cookbook indexes in order to structure our data: ingredients, cooking methods, courses, cuisines.

If we felt no unease continuing the lineage of precision nesting that link the scales of digitization and crowd-sourced transcription, we could proceed with a completely algorithmic approach—"cleaning" our data using scripts, linguistic rules, and even machine learning. These methods yield results by providing an approximation—which we suspected might hide diversity. We could imagine trusting that an approximation could be good enough or using algorithmic approaches. However, to understand what was being approximated—and what was being smoothed together—we needed to create a grounded approach to making a data model for our index.

Now when we look at those lists of variations on "Potatoes au Gratin" or some other group of transcriptions, we are focused on the task of choosing a label that will be a node in our data set and will serve as a pointer to all of the varying transcribed values in the NYPL data set. We are building the set of labels from the set of data rather than beginning by writing some large, hierarchical domain model. We want to represent the concept of two eggs and bacon not caring if it was written "bacon and two eggs" or "2 Eggs and Bacon."

To get from transcription to concept, we began with a set of simple rules: spell out numbers, use lower case letters. Actually engaging with the menu transcriptions quickly raised other questions. For example, on the question of place names, we decided to apply capitalization rules (in accord with style guides like the *Chicago Manual of Style* (http://www.chicagomanualofstyle.org/16/ch08/ch08_sec060.html)) that say that you capitalize when the reference to place is literal, but not when the reference makes a generic association: yes to Virginia ham or Blue Point oysters but no to swiss cheese or scotch. We also found many single transcriptions containing multiple concepts, like "steak, sauce béarnaise." Since we want a way to be able to systematically find multiple components of a dish, we're opting to standardize how we labeled the addition of sauces, garnishes, and other added

ingredients. Here is one instance where we plan to use algorithmic tools to help us analyze some of this big data after we have grounded it in a specific data model.

In building an index, we are engaged in creating scalability. We know that scalability is a process of articulations between different scales; however, Tsing suggests—and we believe—that those articulations are often hidden. Conversely, indexes are tools of scalability that make these articulations explicit.

Our index is about ingredients, meal structures, and cooking techniques. Someone else could re-index the menus material in a different way. Variations might involve attending to the species of the plants and animals that are in foods or taking a nutritional approach that classifies food based on calories, vitamins, carbohydrates. We can also imagine projects that attend to language use in ways that our index suppresses. As libraries and researchers move forward in making and curating data, instead of the constant refrain of "cleaning," we want to encourage indexing, which allows us to build up explicit and flexible bases of knowledge that people can continue to access and understand.

Sharing Control of Authority

One of the mechanisms that librarians and archivists have used to build and maintain large, distributed information systems is a set of practices referred to as authority control. In brief, these practices involve creating defined and agreed upon taxonomies as well as guidelines for the application of such arrangements of terms. The Library of Congress Subject Headings (http://id.loc.gov/authorities/subjects.html) represent one instance of authority control. Maintaining such a system is labor intensive and has been used only for supporting core library activities like managing collections and supporting patrons in finding materials. Libraries and archives are trying to take advantage of technological developments in linked data—merging their centuries-old authority control practices with the affordances of the World Wide Web.

However, what relatively few have seized on are new opportunities to apply the practices of authority control outside the original core needs of collection organization and wayfinding.

These new opportunities fall somewhere between digital library practices and digital humanities research, but the gap is one that more projects should embrace the opportunity to fill. There is a need for projects that take "authority work" as an invitation to new creativity; an invitation for making and building. In such a model, multiple regimes of authorities might be built up from critically-aware and engaged intellectual communities to meet their own specific needs while also participating in larger technological and information systems.

We imagine those communities will contain librarians and scholars. Though librarians and humanities scholars have frequently intersected, they have rarely interacted in the ways we are calling for. Simplifying to the point of caricature, these existing interactions go something like this: humanities scholars point out that the structure and content of a specific archive or collection represents even recreates certain cultural logics. For example, the systems of describing collections—such as the widely-used Library of Congress Subject Headings—reify concepts about persons or cultures that really ought to be interrogated more closely or perhaps discredited and dismantled all together. For the most part, librarians, archivists, and information scientists acknowledge these flaws and perhaps even work to remedy them in the course of maintaining systems that preserve whatever partial archives do exist or helping patrons to find information they need.

We are looking for new forms of collective action that can be expressed through the professional work of humanities scholars and librarians. This is not simply a call for the production of more and more data—attempting to subvert the work of categorization and classification through the production of ever more local, finely-wrought distinctions, details, qualifications. Our aim is to develop ways of working that validate local experiences of data without removing them from a more global network of information exchange. These practices, as we imagine

them, resonate with Bethany Nowviskie's interpretation (http://nowviskie.org/2016/everywhere-every-when/) of the Afrofuturist philosophy of Sun Ra (as expressed by Shabaka Hutchings), which claims: "Communities that have agency are able to form their own philosophical structures." The transition to working in a linked data paradigm should be valued not principally for the ways in which it might make large-scale information systems operate more smoothly, but rather for the ways in which it can create localized communities of authority, within which people can take control of the construction of data and the contexts in which it lives. In a keynote presentation (http://matienzo.org/2016/to-hell-with-good-intentions/) at the 2015 LITA Forum Mx (Mark) A. Matienzo articulated a parallel version of this view, saying:

> We need to begin having some serious conversations about how we can best serve our communities not only as repositories of authoritative knowledge or mere individuals who work within them. We should be examining the way in which we can best serve our communities to support their need to tell stories, to heal, and to work in the process of naming.

Discussions of "cleaning" data fails to capture this need. The cleaning paradigm assumes an underlying, "correct" order. However tidy values may look grouped into rows or columns or neatly-delimited records, this tidiness privileges the structure of a container rather than the data inside it. This is the same diversity-hiding trick that nonscalability theory encourages us to recognize.

It is not enough to recognize; we also wish to offer a way of working. In arguing against cleaning, we propose index-making. In this approach, the first things we would do with our data sets, rather than normalize them, is find the communities within which our data matters. With those communities in mind and even in dialogue, we would ask, what are the concepts that structure this data? And how can this data, structured in this way, point to other people's data?

This way of thinking allows us to see the messiness of data not as a block to scalability but as a vital feature of the world which our data represents and from which it emerges.

Please cite as

Katie Rawson and Trevor Muñoz, “Against Cleaning,” Curating Menus, July 7, 2016, http://www.curatingmenus.org/articles/against-cleaning/.

Notes

1.

Cf. Hadley, Wickham, "Tidy Data." *Journal of Statistical Software*, 59.10 (2014): 1 - 23. doi: 10.18637/jss.v059.i10 (http://dx.doi.org/10.18637/jss.v059.i10).

2.

The fact that these communities have developed discourses for describing the boundaries of the claims they make does not inoculate them from critique about the costs and shortcomings of their methods. Cf. Bruno, Latour, "Why has critique run out of steam? From matters of fact to matters of concern." *Critical Inquiry* 30.2 (2004): 225-248.

3.

For a variety of reasons, we would not recommend this course of action to others without serious deliberation.There is a reason why applications like Open Refine are so popular and useful. If you would like to know more, contact us.

4.

If we had a dictionary to compare these materials too, the process may had been more automatable; however, from what we have found thus far, that particular language resource—Wordnet for Oysters!—doesn't exist.

5.

Anna Lowenhaupt Tsing. "On Nonscalability: the Living World Is Not Amenable to Precision-Nested Scales." *Common Knowledge*. 18.3 (2012): 505-524.

6.

Michael Lascarides and Ben Vershbow, "What's On the Menu?: Crowdsourcing at the New York Public Library," Crowdsourcing our Cultural Heritage, ed. Mia Ridge (Surrey, UK: Ashgate, 2014).

7.

The NYPL menu collector Frank E. Buttolph's acquisition practices reinforce the role and scale of printers in menu production in the twentieth century. In addition to restaurants and customers, she went straight to the menu source—printers—to fill out her collection.

8.

Cf. Tsing 519

9.

Andrew Stauffer. "The Nineteenth-Century Archive in the Digital Age," *European Romantic Review*. 23:3 (2012): 335-341. doi: 10.1080/10509585.2012.674264 (http://dx.doi.org/10.1080/10509585.2012.674264).

10.

Early versions of the project interface featured a social-media-style call-to-action below the image snippet (“What does this say?”), as well as brief instructions below the text input box: “Please type the text of the indicated dish EXACTLY as it appears. Don't worry about accents” (see an example at the Internet Archive (https://web.archive.org/web/20120102212103/http://menus.nypl.org/menu_items/664698/edit)). This accompanying text was quickly dropped—presumably because the task seemed self-evident enough from the layout of the transcription screen.

curatingmenus.org (http://curatingmenus.org/articles/against-cleaning/) · by Katie Rawson · July 6, 2016

Do Digital Humanists Need to Understand Algorithms?

BENJAMIN M. SCHMIDT

Algorithms and Transforms

Ian Bogost recently published an essay[1] arguing that fetishizing algorithms can pollute our ability to accurately describe the world we live in. "Concepts like 'algorithm,'" he writes, "have become sloppy shorthands, slang terms for the act of mistaking multipart complex systems for simple, singular ones" (Bogost). Even critics of computational culture succumb to the temptation to describe algorithms as though they operate with a single incontrovertible beauty, he argues; this leaves them with a "distorted, theological view of computational action" that ignores human agency.

As one of the few sites in the humanities where algorithms are created and deployed, the digital humanities are ideally positioned to help humanists better understand the operations of algorithms rather than blindly venerate or condemn them. But too often, we deliberately occlude understanding and meaning in favor of an instrumental approach that simply treats algorithms as tools whose efficacy can be judged intuitively. The underlying complexity of computers makes some degree of ignorance unavoidable. Past a certain point, humanists certainly do *not* need to understand the algorithms that produce results they use; given the complexity of modern software, it is unlikely that they could.

But although there are elements to software we can safely ignore, some basic standards of understanding remain necessary to practicing humanities data analysis as a scholarly activity and not merely a technical one. While some algorithms are indeed byzantine procedures without much coherence or purpose, others are laden with assumptions that we are perfectly well equipped to understand. What an algorithm does is distinct from, and more important to understand, than how it does it. I want to argue here that a fully realized field of humanities data analysis can do better than to test the validity of algorithms from the outside; instead, it will explore the implications of the assumptions underlying the processes described in software. Put simply: digital humanists do not need to understand algorithms *at all*. They do need, however, to understand the transformations that algorithms attempt to bring about. If we do so, our practice will be more effective and more likely to be truly original.

The core of this argument lies in a distinction between *algorithms* and *transformations*. An algorithm is a set of precisely specifiable steps that produce an output. "Algorithms" are central objects of study in computer science; the primary intellectual questions about an algorithm involve the resources necessary for those steps to run (particularly in terms of time and memory). "Transformations," on the other hand, are the reconfigurations that an algorithm might effect. The term is less strongly

tions that can be taken on a shape) and linguistics (where it forms the heart of Noam Chomsky's theory of "transformational grammar").

Computationally, algorithms create transformations. Intellectually, however, people design algorithms in order to automatically perform a given transformation. That is to say: a transformation expresses a coherent goal that can be understood independently of the algorithm that produces it. Perhaps the simplest example is the transformation of sorting. "Sortedness" is a general property that any person can understand independently of the operations that produce it. The uses that one can make of alphabetical sorting in humanities research—such as producing a concordance to a text or arranging an index of names—are independent of the particular algorithm used to sort. There are, in fact, a multitude of particular algorithms that enable computers to sort a list. Certain canonical sorting algorithms, such as quicksort, are fundamental to the pedagogy in computer science. (The canonical collection and explanation of sorting algorithms is the first half of Knuth's canonical computer science text.) It would be ludicrous to suggest humanists need to understand an algorithm like quicksort to use a sorted list. But we *do* need to understand sortedness itself in order to make use of the distinctive properties of a sorted list.

The alternative to understanding the meaning of transformations is to use algorithms instrumentally; to hope, for example, that an algorithm like Latent Dirichlet Allocation will approximate existing objects like "topics," "discourses," or "themes" and explore the fissures where it fails to do so. (See, for example, Rhody; Goldstone and Underwood; Schmidt, "Words Alone.") This instrumental approach to software, however, promises us little in the way of understanding; in hoping that algorithms will approximate existing meanings, it in many ways precludes them from creating new ones. The signal criticism of large-scale textual analysis by traditional humanists is that it tells scholars nothing they did not know before. This critique is frequently misguided; but it does touch on a frustrating failure, which is that distant reading as commonly practiced frequently fails to offer any new ways of understanding texts.

Far more interesting, if less immediately useful, will be to marry large-scale analysis to what Stephen Ramsay calls "algorithmic criticism": the process of using algorithmic transformations as ways to open texts for new readings (Ramsay). This is true even when, as in some of the algorithms Ramsay describes, the transformation is inherently meaningless. But transformations that embody a purpose themselves can help us to create new versions of text that offer fresh or useful perspectives. Seeking out and describing how those transformations function is a type of work we can do more to recognize and promote.

The Fourier Transform and Literary Time

A debate between Annie Swafford and Matt Jockers over Jockers's "Syuzhet" package[2] for exploring the shape of plots through sentiment analysis offers a useful case study of how further exploring a transformation's purpose can enrich our vo-

ing conversation with Jockers centered on the appropriateness of his use of a low-pass filter from signal processing as a "smoothing function." Jockers argued it provided an excellent way to "filter out the extremes in the sentiment trajectories." Swafford, on the other hand, argued that it was often dominated by "ringing artifacts" which, in practice, means the curves produced place almost all their emphasis "at the lowest point only and consider rises or falls on either side irrelevant" (Jockers, "Revealing Sentiment"; Swafford "Problems"; Swafford, "Why Syuzhet Doesn't Work").

The Swafford and Jockers debate hinged over not just an algorithm, but a concretely defined transformation. The discrete Fourier transform undergirds the low-pass filters that Jockers uses to analyze plot. The thought that the Fourier transform might make sense as a formation for plot is an intriguing one; it is also, as Swafford argues, quite likely wrong. The ringing artifacts that Swafford describes are effects of a larger issue: the basic understanding of time embodied in the transformation itself.

The purpose of the Fourier transform is to represent cyclical events as *frequencies* by breaking complex signals into their component parts. Some of the most basic elements of human experience—most notably, light and sound—physically exist as repeating waves. The Fourier transform offers an easy way to describe these infinitely long waves as a short series of frequencies, constantly repeating. The pure musical note "A," for example, is a constant pulsation at 440 cycles per second; as actually produced by a clarinet, it has (among other components) a large number of regular "overtones," less powerful component notes that occur at a higher frequency and enrich the sound beyond a simple tone. A filter like the one Jockers uses strips away these regularities; it is typically used in processes like MP3 compression to strip out notes too high for the human ear to hear. When applied even more aggressively to such a clarinet tone, it would remove the higher frequencies, preserving the note "A" but attenuating the distinctive tone of the instrument.[3]

The idea that plots might be represented in the frequency domain is fascinating, but makes some highly questionable assumptions. Perhaps the most striking assumption is that plots, like sound or light, are composed of endlessly repeating signals. A low-pass filter like the one Jockers employs ignores any elements that seem to be regularly repeating in the text and instead focuses on the longest-term motions; those that take place over periods of time greater than a quarter or a third the length of the text. The process is analogous to predicting the continuing sound of the clarinet based on a sound clip of the note "A" just 1/440th of a second long, a single beat of the base frequency. This, remarkably, is feasible for the musical note, but only because the tone repeats endlessly. The default smoothing in the Syuzhet package assumes that books do the same; among other things, this means the smoothed versions assume the start of every book has an emotional valence that continues the trajectory of its final sentence. (I have explained this at slightly greater length in Schmidt, "Commodius Vici.")

For some plots, including Jockers's primary example, *Portrait of the Artist as a Young Man,* this assumption is not noticeably

from Project Gutenberg. William Dean Howell's *The Rise of Silas Lapham* is a story of ruination; *Ragged Dick,* by Horatio Alger, is the archetypal "Rags to Riches" novel of the nineteenth century; *Madame Bovary* is a classically tragic tale of decline. Three different smoothing functions are shown: a weighted moving average, among the simplest possible functions; a loess moving average, which is one of the most basic and least assumption-laden algorithms used in exploratory data analysis; and the low-pass filter included with Syuzhet.4

The problems with the Fourier transform here are obvious. A periodic function forces Madame Bovary to be "as well off" after her death as before her infidelity. The less assumption-laden methods, on the other hand, allow her fate to collapse at the end and for *Ragged Dick*'s trajectory to move upward instead of ending on the downslope. Andrew Piper suggests5 that it may be quite difficult to answer the question, "How do we know when a curve is 'wrong'?" (Piper, "Validation"). But in this case, the wrongness is actually quite apparent; only the attempt to close the circle can justify the downturn in Ragged Dick's fate at the end of the novel.

What sort of evidence is this? By Jockers's account,6 the Bovary example is simply a negative "validation" of the method, by which I believe he means a sort of empirical falsification of the claim that this is the best method in all cases (Jockers, "Requiem"). Swafford's posts imply similarly that case-by-case validation and falsification are the gold standard. In her words, the package (and perhaps the digital humanities as a whole) need "more peer review and rigorous testing—designed to confirm or refute hypotheses" (Swafford, "Continuing").

Seen in these terms, the algorithm is a process whose operations are fundamentally opaque; we can poke or prod to see if it matches our hopes, but we can never truly *know* it. But when the algorithm is a means of realizing a meaningful transformation, as in the case of the Fourier transform, we can do better than this kind of quality assurance testing; we can interpretively *know* in advance where a transformation will fail. I did not choose *Madame Bovary* at random to see if it looked good enough; instead, the implications of the smoothing method made it obvious that the tragedy, in general, was a *type* of novel that this conception of sentiment that Syuzhet's smoothing could not comprehend. I will admit, with some trepidation, that I have never actually read either *Madame Bovary* or *Ragged Dick*; but each is the archetype of a plot wholly incompatible with low-pass filter smoothing. Any other novel that ends in death and despair or extraordinary good fortune would fail in the same way.

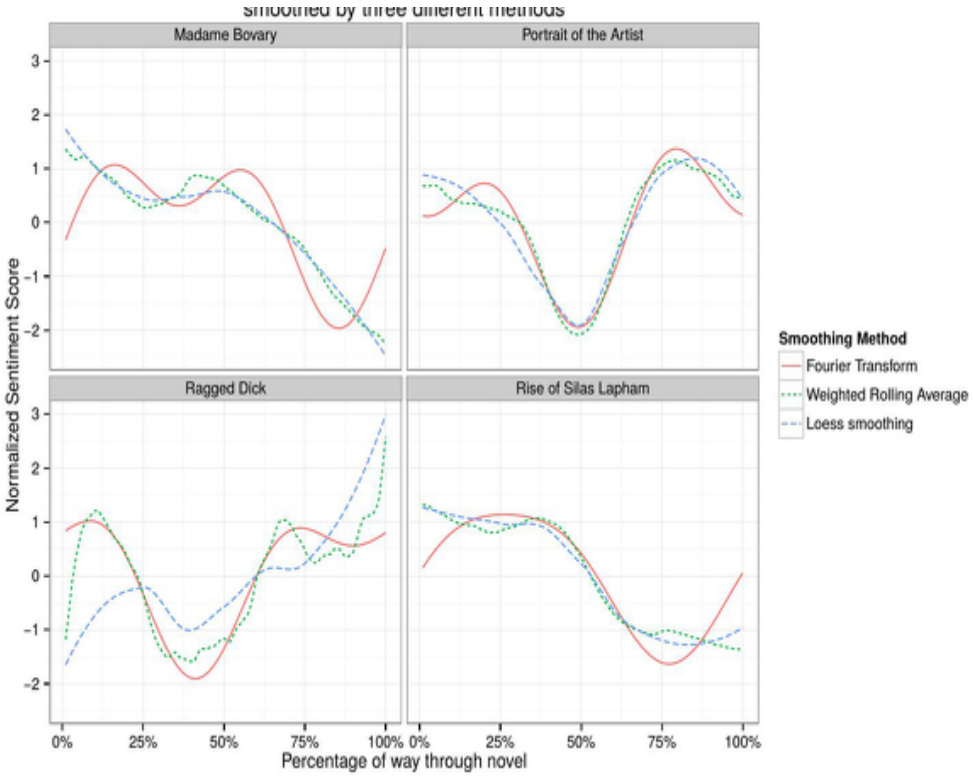


Figure 48.1. Four plot trajectories.

These problems carry through to Jockers's set of fundamental plots: all begin and end at exactly the same sentiment. But the obvious problems with this assumption were not noted in the first two months of the package's existence (which surely included far more intensive scrutiny than any peer-review process might have). One particularly interesting reason that these failings were not immediately obvious is that line charts, like Figure 48.1, do not fully embody the assumptions of the Fourier transform. The statistical graphics we use to represent results can *themselves* be thought of as meaningful transformations into a new domain of analysis. And in this case, the geometries and coordinate systems we use to chart plots are themselves emblazoned with a particular model. Such line charts assume that time is linear and infinite. In general, this is far and away the easiest and most accurate way to represent time on paper. It is not, though, true to the frequency domain that the Fourier transform takes for granted. If the Fourier transform is the right way to look at plots, we should be plotting in polar coordinates, which wrap around to their beginning. I have replotted the same data in Figure 48.2, with percentage represented as an angle starting from 12:00 on a clock face and the sentiment defined not by height but by distance from the center.

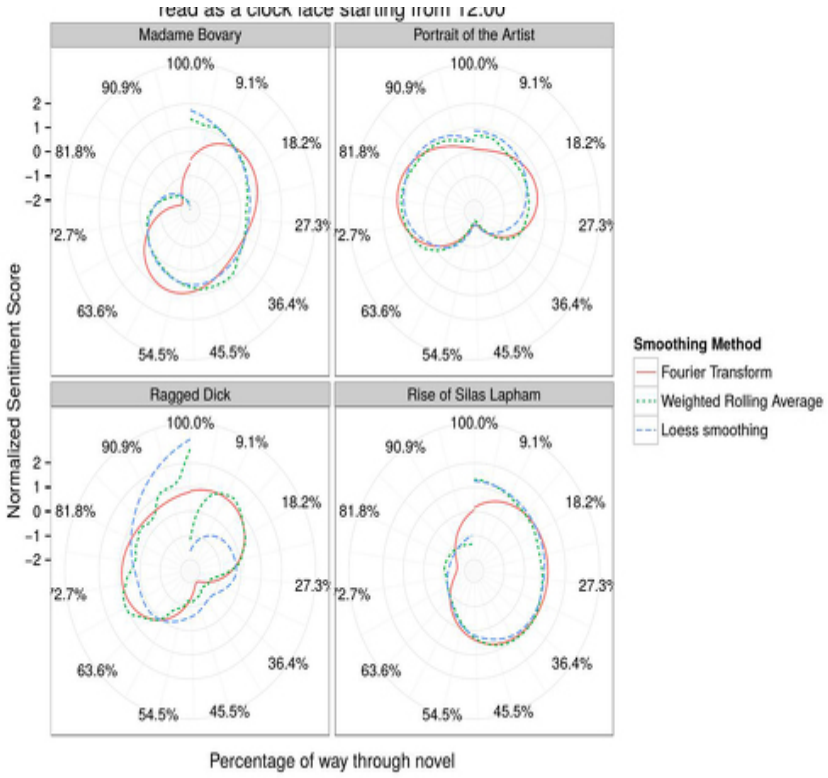


Figure 48.2. Four plot trajectories plotted in polar coordinates.

Here, the assumptions of the Fourier transform are much more clear. For all of the novels here, time forms a closed loop; the ending points distort themselves to line up with the beginning, and vice versa. The other algorithms, on the other hand, allow great gaps: the *Madame Bovary* arc circles inward as if descending down a drain, and *Ragged Dick* propels outward into orbit.

These circular plots are more than falsifications. Fully embracing the underlying assumptions of the transform in this way does not only highlight problems with the model; it suggests a new perspective for thinking about plots. This view highlights the gap between the beginning and end as a central feature of the novel; in doing so, it challenges us to think of the time that plots occupy as something other than straightforwardly linear.

This is a conversation worth having, in part because it reminds us to question our other assumptions about plots and time. The infinite time that the Cartesian plot implies is, in some ways, just as false as the radial one. Many smoothing methods (including the one I would like to see used in Syuzhet, loess regression), can easily extrapolate past the beginning and end of the plot. That this is possible shows that they are, in some ways, equally unsuitable for the task at hand. The heart of the distinction between *fabula* and *syuzhet,* in fact, is that there is no way to speak about "before the beginning" of a novel, or what words Shakespeare might have written if he had spent a few more hours working past the end of *Hamlet*. Any model that implies such phrases exist is obviously incorrect.

But even when arguably false, these transformations may yet be productive of new understandings and forms of analysis.

crally. By asking what sorts of plots of the frequency domain might be useful for, we can abstractly identify whole domains where new applications may be more appropriate.

For example: the ideal form of the three-camera situation comedy is written so that episodes can air in any arbitrary order in syndication. That is to say, along some dimensions they *should* be cyclical. For sitcom episodes, cyclicality is a useful framework to keep in mind. The cleanness of the fit of sentiment, theme, or other attributes may be an incredibly useful tool both to understand how commercial implications intertwine with authorial independence, or for understanding the transformation of a genre over time. Techniques of signal processing could be invaluable in identifying, for example, when and where networks allow writers to spin out multi-episode plot lines.[7]

Though the bulk of the Swafford and Jockers conversation centered on the issue of smoothing, many digital humanists seem to have found a second critique Swafford offered far more interesting. She argued that the sentiment analysis algorithms provided by Jockers's package, most of which were based on dictionaries of words with assigned sentiment scores, produced results that frequently violated "common sense." While the first issue seems blandly technical, the second offers a platform for digital humanists to talk through how we might better understand the black boxes of algorithms we run. What does it mean for an algorithm to accord to common sense? For it to be useful, does it need to be right 100 percent of the time? 95 percent? 50.1 percent? If the digital humanities are to be a field that appropriates tools created by others, these are precisely the questions it needs to practice answering.

To phrase the question this way, though, is once again to consider the algorithm itself as unknowable. Just as with the Fourier transform, it is better to ask consciously what the transformation of sentiment analysis does. Rather than thinking of the sentiment analysis portion of Syuzhet as a set of word lists to be tested against anonymous human subjects, for example, we should be thinking about the best way to implement the underlying algorithms behind sentiment analysis—logistic regression, perhaps—to distinguish between things other than the binary of "positive" and "negative." Jockers's inspiration, Kurt Vonnegut, for example, believed that the central binary of plot was fortune and misfortune, not happiness and sadness; while sentiment analysis provides a useful shortcut, any large-scale platforms might do better to create a classifier that actually distinguishes within that desired binary itself. Andrew Piper's work on plot structure involves internal comparisons within the novel itself (Piper, "Novel Devotions"). Work like this can help us to better understand plot by placing it into conversation with itself *and* by finding useful new applications for transformations from other fields.

Doing so means that digital humanists can help to dispel the myths of algorithmic domination that Bogost unpacks, rather than participating in their creation. When historians applied psychoanalysis to historical subjects, we did not suggest they "collaborate" with psychoanalysts and then test their statements

were seen as themselves meaningful. It is good and useful for humanists to be able to push and prod at algorithmic black boxes when the underlying algorithms are inaccessible or overly complex. But when they are reduced to doing so, the first job of digital humanists should be to understand the goals and agendas of the transformations and systems that algorithms serve so that we can be creative users of new ideas, rather than users of tools the purposes of which we decline to know.

Notes

1. http://www.theatlantic.com/technology/archive/2015/01/the-cathedral-of-computation/384300.

2. http://www.matthewjockers.net/%202015/02/02/syuzhet.

3. It may be worth emphasizing that a low-pass filter removes all elements above a certain frequency; it does not reduce to its top five or ten frequencies, which is a different, equally sensible compression scheme.

4. For all three filters, I have used a span approximating a third of the novel. The loess span is one-third; the moving average uses a third of the novel at a time; and the cutoff for the low-pass filter is three. To avoid jagged breaks at outlying points, I use a sine-shaped kernel to weight the moving average so that each point weights far-away points for its average less than the point itself.

5. http://txtlab.org/?p=470.

6. http://www.matthewjockers.net/2015/04/06/epilogue.

7. This does not necessarily mean that Fourier transform is the best way to think of plots as radial. Trying to pour plot time into the bottle of periodic functions, as we are seeing, produces extremely odd results. As Scott Enderle points out, even if a function is completely and obviously cyclical, it may not be regular enough for the Fourier transform to accurately translate it to the frequency domain (Enderle).

Bibliography

Bogost, Ian. "The Cathedral of Computation." *The Atlantic,* January 15, 2015. http://www.theatlantic.com/technology/archive/2015/01/the-cathedral-of-computation/384300/.

Enderle, Scott. "What's a Sine Wave of Sentiment?" *The Frame of Lagado* (blog), April 2, 2015. http://www.lagado.name/blog/?p=78.

Goldstone, Andrew, and Ted Underwood. "The Quiet Transformations of Literary Studies: What Thirteen Thousand Scholars Could Tell Us." *New Literary History* 45, no. 3 (2014): 359–84. doi:10.1353/nlh.2014.0025.

Jockers, Matthew. "Requiem for a Low Pass Filter." *Matthewjockers.net,* April 6, 2015. http://www.matthewjockers.net/2015/04/06/epilogue/.

—. "Revealing Sentiment and Plot Arcs with the Syuzhet Package." *Matthewjockers.net,* February 2, 2015. http://www.matthewjockers.net/2015/02/02/syuzhet/.

Knuth, Donald E. *The Art of Computer Programming: Volume 3: Sorting and Searching*. Reading, Mass.: Addison-Wesley Professional, 1998.

Piper, Andrew. "Novel Devotions: Conversional Reading, Computational Modeling, and the Modern Novel." *New Literary History* 46, no. 1 (2015): 63–98. doi:10.1353/nlh.2015.0008.

Ramsay, Stephen. *Reading Machines: Toward an Algorithmic Criticism.* Urbana: University of Illinois Press, 2011.

Rhody, Lisa M. "Topic Modeling and Figurative Language." *Journal of Digital Humanities* 2, no. 1 (2013). http://journalofdigitalhumanities.org/2%E2%80%931/topic-modeling-and-figurative-language-by-lisa-m-rhody/.

Schmidt, Benjamin. "Commodius Vici of Recirculation: The Real Problem with Syuzhet." Author's blog, April 13, 2015. http://benschmidt.org/2015/04/03/commodius-vici-of-recirculation-the-real-problem-with-syuzhet/.

—. "Words Alone: Dismantling Topic Models in the Humanities." *Journal of Digital Humanities* 2, no. 1 (2013). http://journalofdigitalhumanities.org/2-1/words-alone-by-benjamin-m-schmidt/.

Swafford, Annie. "Problems with the Syuzhet Package." *Anglophile in Academia: Annie Swafford's Blog,* March 2, 2015. https://annieswafford.wordpress.com/2015/03/02/syuzhet/.

—. "Continuing the Syuzhet Discussion." *Anglophile in Academia: Annie Swafford's Blog,* March 7, 2015. https://annieswafford.wordpress.com/2015/03/07/continuingsyuzhet/.

—. "Why Syuzhet Doesn't Work and How We Know." *Anglophile in Academia: Annie Swafford's Blog,* March 30, 2015. https://annieswafford.wordpress.com/2015/03/30/why-syuzhet-doesnt-work-and-how-we-know/.

API and Atom Feed

Data about this text is available via read-only JSON API endpoints: sentences, annotations, comments, and index keywords.

Comments posted on this text can be followed by subscribing to this text's Atom Feed.

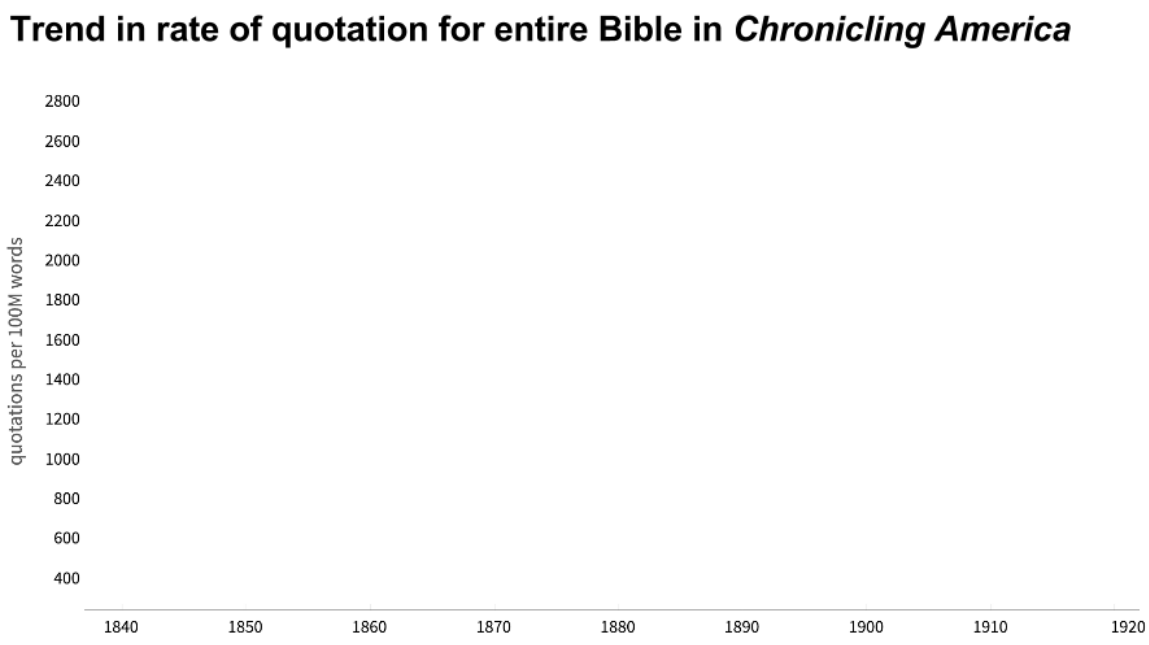
Isn't it obvious? | Lincoln Mullen

lincolnmullen.com (https://lincolnmullen.com/blog/isnt-it-obvious/)

A common response to digital history research is that has failed to make an argumentative or interpretative payoff (http://dhdebates.gc.cuny.edu/debates/text/77) commensurate with the amount of effort that has been put into it. Broadly speaking, I'm sympathetic (https://rrchnm.org/argument-white-paper/) to that claim. But there is a particular form that this claim sometimes takes which I think is mistaken: the idea that even when interpretations or arguments from digital history work are presented, they do not tell us anything new. Scott Weingart has written perceptively (http://scottbot.net/digital-history-can-never-be-new/) and more generally about the problem that "digital history can never be new." I want to add only a small piece to that discussion.

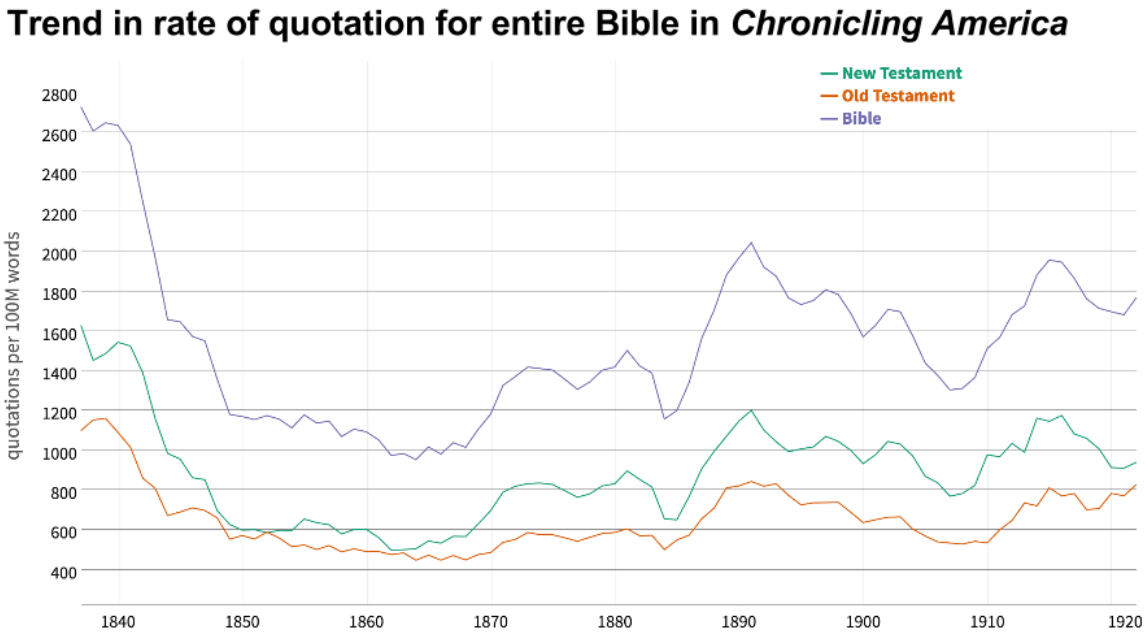
When I present the results of my work as a visualization, audiences sometimes react by saying that they can immediately explain what the visualization shows and that it merely reflects what they already knew. Matthew Lincoln has written about the "confabulation (https://matthewlincoln.net/2015/03/21/confabulation-in-the-humanities.html)" or "just-so stories" that readers of visualizations can come up with in order to explain them. And it's not just the audiences for visualizations who do this; I do it myself whenever I create visualizations for my own consumption. The sense that a visualization is immediately explainable is the result, I think, of the ability of visualizations to rapidly and persuasively communicate large amounts of information.

The problem is that often it is not possible to know what a visualization would look like in advance, and so it should not be possible immediately after seeing it to explain it as something that we already know. To convey this point, I've been trying a new technique when giving talks. Before showing the audience a visualization, I first show them a blank visualization with just the axes and title, and ask them to sketch out what they think the trend will be. For instance, here is a blank visualization of the trends in the rate at which the Bible was quoted in nineteenth-century U.S. newspapers, from my *America's Public Bible* (http://americaspublicbible.org/) project.



What would you predict the trends should be for the rates at which the Bible was quoted in nineteenth-century U.S. newspapers?

Before I made the visualization, I couldn't predict the trend. I would have said something like "vaguely downwards." I certainly could not have predicted the scale. When I have given this talk to various audiences, no one has been able to predict what the actual visualization should look like. Nor for that matter could anyone guess what the top ten most frequently quoted Bible verses were with any degree of accuracy—certainly not me.



The actual trend lines are not easily guessed.

I've found that this simple approach, used both for myself and with audiences, helps dispel the sense that what is learned from visualizations was already obvious.

Updates, 10 January 2018

1. An additional thought that occurs to me after discussion with my collaborator, Kellen Funk. In many instances, scholars might already know that a phenomenon happened, but that is different than the details of how it happened. For instance, before Ryan Cordell et al. did the work of the *Viral Texts* project, scholars knew that newspapers reprinted one another, but that project showed the actual relationships created by those borrowings. Similarly, scholars have long known that states borrowed their codes of civil procedure from New York, but Kellen and I have created a network of those borrowings. Knowing that there is a network and knowing what the network is are two different kinds of knowledge.
2. Hadley Wickham pointed out that it is possible to create a p-value by showing an audience different plots and letting them try to identify the

actual plot. He has an article on the subject. H. Wickham, D. Cook, H. Hofmann and A. Buja, "Graphical inference for infovis (http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5613434&isnumber=5613414)," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 973-979, Nov.-Dec. 2010. doi: 10.1109/TVCG.2010.161

lincolnmullen.com (https://lincolnmullen.com/blog/isnt-it-obvious/)

Representation Matters

xpmethod.plaintext.in (http://xpmethod.plaintext.in/torn-apart/reflections/moacir_p_de_sa_pereira_2.html) · by Moacir P. de Sá Pereira

by

> Diagrams are of great utility for illustrating certain questions of vital statistics by conveying ideas on the subject through the eye, which cannot be so readily grasped when contained in figures.

> Not only is it easy to lie with maps, it's essential.

Florence Nightingale had a mission. It was crucial to persuade the British Army, over the protestations of its generals who were convinced they knew better than some woman, that terrible hygiene was killing off their troops faster than enemy bullets. Half a year after her arrival at the Scutari Barracks, now the Selimiye Barracks in the Üsküdar district of Istanbul, the mortality rate of British soldiers dropped from 42.7% to 2.2%. She was certain it was the introduction of sanitation measures that caused the decrease, but how could she show it to drive home the difference her efforts made?

For Nightingale, familiar with the pioneering statistical work of Adolphe Quetelet and William Farr, aggregating data about individual outcomes to draw a larger picture seemed the best way to make a convincing argument, reversing the adage about tragedy and statistics typically misattributed to Stalin. Instead of flattening each individual's tragedy in dying from poor sanitation, Nightingale

compounded the tragedies into one greater than the sum of its parts, an emergent entity that carried extra affective weight from its simple proposals: more medical supplies, more nurses, less filth.

But as Nightingale explains in the opening of 1858 pamphlet on the mortality of the British Army, a diagram will be more useful to carry this affective weight, prompting the design of the "batwings"—diagrams that treat the year as a cycle and drew a line around the calendar to show trends in mortality.

Diagrams representing the relative Mortality from ZYMOTIC DISEASES, from WOUNDS, and from ALL OTHER CAUSES in the HOSPITALS of the ARMY in the EAST

April 1855 – March 1856

April 1854 – March 1855

Despite the fact that, for some reason, the years go from right to left (reproduced from Nightingale's original) and that the visualizations begin on April 1, or what is 9 o'clock on the calendar wheels, these diagrams demonstrate *very* clearly that the bulk of soldiers were dying of infection, or "zymotic diseases." Wounds and other causes are barely a blip in the chart. Yet Nightingale engages in a bit of sleight of hand to make the salient political point here. During the time period covered in these diagrams, 14,476 soldiers died from zymotic diseases, as opposed to 3,486 of wounds and other causes. In other words, about four times as many soldiers died of zymotic diseases.

Does the purple batwing look four times larger than the other two batwings combined?

With an *n* of one (my mom), the answer is clearly no. Nightingale is not wrong to suggest that too many soldiers are dying of infectious disease. But this chart makes it look like soldiers are effectively *only* dying of infectious disease, with a

negligible number dying from causes more proximate to war. To achieve this effect, Nightingale mobilizes an ancient "trick" of the visualization trade, using two dimensions to represent a one-dimensional variable.

When the "batwings" are converted to a more familiar visualization, that of the stacked bar chart, it's still clear that soldiers are dying mostly of zymotic diseases. However, the role of the other two causes of death is not as obliterated as in the batwings. The reason is clear: the *area* of the purple rectangles is proportionate to how much larger the mortality rate is to that of the other two causes. Though I rely on a second dimension to give the bars width, every observation enjoys the same, constant width, meaning the areas remain proportional. We can assume each bar is one unit wide and *n* units tall, meaning the only important value is the height, as the width cancels out when multiplying area.

It's important to put "trick" in scare quotes, because though Nightingale would improve on the batwings, it remains the case that Nightingale wanted to use the data, in their visual forms, to not just tell a story, but, rather, *make an argument*.

April 1855 – March 1856
April 1854 – March 1855

Nightingale returns in 1859 and "corrects" the batwing diagrams with the diagrams above, now known as polar area charts or Nightingale roses. This is more "correct" in that the one-dimensional variable of mortality rate is reproduced as the single variable of area, meaning that the wedges are appropriately proportionate to each other. However, unlike with the stacked bar chart, here area is measured from the center, meaning part of many of the purple wedges is covered up by orange and green. So while, still, zymotic disease remains a runaway killer, now it's, oddly, underrepresented. Furthermore, expecting people to accurately compare areas as opposed to lengths is a tall order. We can tell which line is longer than which, and typically we can tell about how much longer (two times, three times). With area, it's a bit trickier.

Which circle is two times larger than the leftmost? And how much larger is the other circle? It's hard to tell, since we can't visually stack the circles, unlike the way the bars can become mental Cuisenaire rods. In both visualizations—batwings and roses—Nightingale gets an effect from the fuzzy visual area calculations. In the first set, the result is massive distortion in favor of her argument. In the second, it's a slight distortion against.

But it is a fiction to assume that one rendering is more "correct" than the other; rather, the question is what distortions are within the realm of the reasonable in terms of making the narrative and persuasive point one wants to. As Mark Monmonier explains in opening *How to Lie with Maps*, maps are only readable and coherent because they have distortions. As Lewis Carroll, Jorge Luis Borges, and Umberto Eco (among others?) remind us, if maps were "true," they would be unusable.

Volume 1

The first volume of *Torn Apart / Separados* challenged us with several questions in terms of visualization, some of which were then discussed in several of the first reflections. There was a temptation, for example, to size the various dots in proportion with certain data associated with them, such as average daily population of detention facilities used by ICE.

But that would have distracted somewhat from the other, competing desires. If the goal is to show that ICE is everywhere, then scaling the markers based on average daily population will make some facilities jump out and others disappear into view. The story becomes different and more focused. In "Clinks" and "Charts," by making all the in-use facilities look the same on first blush, the banal repeatability of ICE as it infects our national body seems more thorough.

On the other hand, in “Banned,” we rely on cartographic distortion to overstate a case. This visualization draws a map of contiguous United States whose combined population is close to but less than the approximate number people banned from entry into the United States at the moment, blocked by the upheld “Muslim Ban.” Yet as sparsely populated mountain or plains state after state is added to the growing black shape, it soon seems like nearly all of America would be banned. We trick the eye, then, into thinking that American population is evenly distributed across the United States. But it isn’t.

This is, unfortunately, a common design choice in choropleth maps, and it resembles the “error” in Nightingale’s batwing, as well. I, at least, would not make such a map under normal circumstances.

But these are not normal circumstances. In this way, “Banned” is a response to the famous map of the 2016 Presidential Election hanging in the West Wing, as tweeted by Trey Yingst (https://twitter.com/TreyYingst/status/862669407868391424):

So just as fewer people voted for the red candidate than the blue, “Banned” exaggerates the size of the US that would be banned from entry.

The red candidate understands that this map is making an argument, that the totality of red space gives the visual illusion of an America standing behind him as President, as the true voice of the majority, and of an overwhelming majority

at that.

Volume 2

The second volume, however, also presented challenges in terms of representation. “Lines,” for example, shows how pervasive the American removal engine is. Yet the number of deportations at a specific port of entry can be single digits or it can be over 1,000. Furthermore, simply drawing lines devalues the fact that each individual removal is its own story that’s the most important in the world to the deportee and their family and loved ones. But somehow the difference between one and 1,000 also needs to be indicated. The solution to me was to use a logarithmic scale. Now, 1,000 is a line that’s four times longer than the line for a single removal, and every soul gets a wedge cutting across the face of the United States as it continues to expel.

In this way, “Lines” has it both ways. It shows the extent of the problem of removals across the world as a whole (after all, the US has no problem removing souls from points of entry like Ireland), but also makes clear that even one removal is a scar on the national image.

The Banality of ICE

Both Freezer visualizations work from literally the same data set, indicating that the same hierarchically grouped data can represent radically different modes of being and interacting with the government contracting machine. In one visualization, a messy ball of yarn shows how interconnected contractors are with the various governmental needs in the awards they win. A sprawling monster like CoreCivic provides “security guards and patrol services” (which we categorize under “The Threat of Violence”), “correctional institutions” (“Walls”), and “facilities support services” (“Surveillance”). Firms like CoreCivic, or even more abstract management and consulting firms like Booz Hamilton, have their fingers in various ICE contracting pies.

Nevertheless, there are not actually that many contractors that are that promiscuously pie-fingering. A company like Spectrum Security Services simply provides "security guards and patrol services" over and over until they have made $180m in ICE awards for 2018. In "ICE Tray," that focus is shown in having Spectrum Security services with just one square (but a large one) inside the "The Threat of Violence" category, where all "security guards and patrol services" awards land. CoreCivic, on the other hand, has several boxes throughout the tray in several differently colored areas, to show its relatively broad reach.

In the other visualization, however, our nine invented categories of government awards are more or less laid atop each other, giving the impression of a Gordian knot of governmental impenetrability. Neither visualization is right; they just aim to highlight different aspects of the same data to make a similar, underlying argument.

In short, ICE funding can be stultifyingly banal. The outrage over the zero tolerance policy in part relies on characterizing ICE as a bunch of special forces cosplayers, kicking people of color with steel-tipped boots and throwing their children in cages. But it's not just that, else ICE funding would go exclusively to boots and cages. It's also the massive apparatus that keeps Hillary-voting, lanyard-wearing, *Pod Save America*–listening Northern Virginia technocrats fed, technocrats largely represented by Democrats in Congress.

The oversized role of these technocrats is hinted at in every visualization for volume 2 save "Lines." In "Districts," we see that ICE funding is not evenly distributed around the US, with 16 Congressional districts taking home almost 90% of the ICE budget since 2014. Of these 16, several are DC-adjacent, where companies like Phacil and WidePoint Integrated Solutions pull in millions of dollars from ICE despite (and because of) being, basically, IT consultants. Unsurprisingly this explosion of money to the IT sector happens to be a trademark of Representative Gerry Connolly's work in Congress (https://fcw.com/articles/2015/03/30/gerry-connolly-fitara-force-

multiplier.aspx). Small wonder, then, that his district is also the most remunerated by ICE. Since 2014, $1.3B has been showered over the Clinton-by-39-points district, much of it to IT companies.

This visualization, showing the distribution of the over $9B in ICE spending since 2014, recontextualizes how those 16 districts chew up the budget. In 17th place, marked in orange above, is Washington, DC., which while another ICE fat cat, also marks the bend towards quickly shrinking turns at the authoritarian trough. And this visualization skips the quarter of all Congressional districts that haven't seen a dime from ICE. Perhaps they need to get in the consulting, computing, constructing, and coercing businesses.

The bipartisan nature of the grotesque gluttony also expresses itself in these top districts, where the pitiful Democrats, shut out of every avenue of access in Washington, or so they tell us, still manage to bring home the fat in thicker, richer slices than their GOP counterparts.

Finally, I should in passing, tying "Districts" to other current events, note that Duncan Hunter may be woeful with his family budgeting (https://www.nytimes.com/2018/08/26/us/duncan-hunter-corruption-scandal.html), but his district has done well under the ICE regime, as Spectrum Security Services has brought in $860M to parts of San Diego and Riverside Counties. California's 50th is the fourth most remunerated district.

The concentrated ICE spending also lets minority- and women-owned companies shine in the glow of government awards, as we see in "Gain." Much of the $890M going to Alaska ends up in the coffers of businesses that are registered with the government as "Alaska Native Corporations" (ANC). One such company, Barling Bay, which is a subsidiary of the Old Harbor Native Coporation, describes the benefits of doing business with an ANC by pointing out that "Alaska Native Villages suffer from some of the worst poverty in this country (http://www.barlingbay.com/family.html)," and, hence, it's important to support ANCs who enjoy special rights in the Federal procurement process.

Barling Bay seems to provide ICE only with IT-related services, but the reach of the biggest ANC bringing money to Alaska is much greater. The NANA Regional Corporation represents over 14,000 Iñupiat shareholders, who largely participate in subsistence activities and full-time or part-time employment (http://www.nana.com/regional/about-us/our-shareholders/) above the Arctic Circle. Nevertheless, NANA's subsidiary, Akima, has grossed those subsistence shareholders over $200M in ICE awards since 2014, mostly in relation to running the Krome detention center in Miami (https://www.ice.gov/detention-facility/krome-service-processing-center) and the Buffalo facility in Batavia, NY (https://www.ice.gov/detention-facility/buffalo-federal-detention-facility). Akima's ability to win awards for various detention-related services contribute to making Alaska's At-Large District the third most flush with ICE cash.

Back in the Beltway, Phacil, based in Virginia's 8th District (5th best remunerated), has the dubious distinction of being the most-remunerated "Black American–owned" (the government's designation) business regarding ICE, which has paid it $310M since 2014 for various IT-related services. The Phacil website recalls the company's start as "a small, minority owned business" that has, obviously, "always embraced diversity and inclusion (https://www.phacil.com/about/diversity-inclusion/)," but the idea that the money ICE has paid them has gone towards encouraging racial equality in the United States is an offensive joke. These minority-owned companies receiving over $1B from ICE since 2014 are, it seems, largely in the various banal businesses of logistics and IT, enriching workers already in these largely white-collar sectors. Though contractors like Akima are actively in the business of jailing their "fellow" people of color, most simply contribute to the more abstract carceral regimes of the state, facilitating the ways in which a laser printer helps keep another child separated from their parents.

Speaking of laser printers, finally, in "Rain," we see how ICE's awarding has grown since 2014. But the tiny dots in the visualization hide a simple detail visible only if one happens to mouse over them in just the right way that causes

nearly the whole visualization to suddenly become darker. Of the 5,500 awards represented in "Rain" (and in "Gain" and "Districts," for that matter), over 1,000 of them went to one company, North Carolina's Net Direct Systems. The $28M Net Direct has brought to North Carolina's 2nd district is comparable chump change on the ICE scale, but the idea that someone in the ICE office, one out of every six times, has awarded a contract to Net Direct boggles the mind and fully delineates how normal the ICE funding apparatus is.

If there is an entity to which nearly 20% of my purchases (in terms of number, not in terms money spent) goes, it would probably be a grocery store. Those trips feature the near daily purchases that provide the fuel that keeps me running. Similarly, the laptops, tablets, and desktops ICE constantly buys from Net Direct are also, then, the fuel that keeps this regime of terror running.

Moacir P. de Sá Pereira (http://moacir.com) (@muziejus (http://twitter.com/muziejus)) is a scholar of literature and space.

1.

Florence Nightingale, *Mortality of the British Army, at Home, at Home and Abroad, and During the Russian War, as Compared with the Mortality of the Civil Population in England* (London: Harrison and Sons, 1858), 1. ↩

2.

Mark Monmonier, *How to Lie with Maps* (Chicago, University of Chicago Press, 1991), 1. ↩

3.

I. Bernard Cohen, "Florence Nightingale," *Scientific American* 250 (1984), 131. Most of my retelling of the Nightingale story is based on Cohen's account. ↩

4.

In more detail, “Lines” relies on its own dataset, whereas the four other visualizations take our master file of 5,500 ICE awards and filter it in different ways for each visualization. For “Freezer,” it takes only 2018 awards, with values greater than 0, and then builds a network graph linking parent companies, companies, NAICS (https://www.census.gov/eos/www/naics/) descriptions of the awards, and our own, decolonial ontology of 9 super-categories of NAICS descriptions. While it was relatively easy to build a one-to-many link between our categories and the NAICS descriptions, the parent/subsidiary relationships on the contractor side of things ended up being occasionally many-to-many, accounting for changes in ownership, corporate restructuring, or even, I suspect, about 20 times, simple user error up at ICE Towers. ↩

5. 1
6. 2
7. 3
8. 4

CA Journal of Cultural Analytics

Critical Search: A Procedure for Guided Reading in Large-Scale Textual Corpora

Jo Guldi

12.20.18

Peer-Reviewed By: Ted Underwood, Anon.

Clusters: Theory

Article DOI: 10.22148/16.030

Dataverse DOI: 10.7910/DVN/BJNAPD

Journal ISSN: 2371-4549

Cite: Jo Guldi, "Critical Search: A Procedure for Guided Reading in Large-Scale Textual Corpora," Journal of Cultural Analytics. *December 20, 2018.*

Whether they work with pages hand-selected from an archive, or keywords hand-selected from a digital corpus, scholars of all kinds wrestle with the issue of exemplarity.[1] Cherry-picked examples leave the scholar's conclusions vulnerable to charges of, at best, irrelevance, and at worst, malfeasance. In digital research, even seemingly simple queries are plagued with these pitfalls, which can only be addressed by critical thinking about the queries we propose, the algorithms we use, and the questions we ask of them. If digital research is approached responsibly, however, it allows the scholar to choose exemplary texts for reading with a precision and clarity unavailable to previous generations.

Consider the case of searching for change in the language of property in the parliamentary debates. Displacement is a theme in modern Britain since the beginnings of the enclosure of traditional peasant commons in the middle ages,

[1] The research documented in this article was graciously supported by the National Science Foundation IBSS Grant 1822091 and a fellowship from the Center of Creative Computation, Southern Methodist University.

but in the nineteenth century, evictions and other displacements became more common, linked to an ideology of botanical improvement, the rise of utilitarian economics, the proliferation of race-based modalities of governance, and the creation of legal mechanisms to aid landlord-led improvement.[2] A dozen different secondary sources have located major changes in the understanding of property in nineteenth-century Britain, but no definitive method exists to synthesize these rival claims about when property changed and how.[3]

Looking for scholar-supplied keywords over time lends particular insights. The scholar, knowing the rising number and length of parliamentary speeches over the century, might choose to count keywords over time as a proportion of all words spoken each year.[4] The timeline shows an apparent eruption of debates about property rights after 1875, between the Landlord and Tenant Act (Ireland) of 1870 and the peasant insurrections known as the Irish Land War of the 1880s, followed by a further explosion of the terms after 1900. The importance of the yellow bar for "eviction" in the timeline (Figure 1) suggests that increasing mentions of tenants and landlords on the floor of parliament may have been driven by discussions of eviction at a time when cases of evicted peasants were being heard on the floor of parliament with great regularity.[5]

[2] Eric Stokes, *The English Utilitarians and India* (Oxford: Clarendon Press, 1959); Thomas M. Devine, *Clanship to Crofters' War* (Manchester: Manchester University Press, 1994); Fredrik Albritton Jonsson, *Enlightenment's Frontier* (New Haven: Yale University Press, 2013).

[3] Important texts include A. V. Dicey, "The Paradox of Land Law," *Law Quarterly Review* 21 (1905): 221-232; Avner Offer in *Property and Politics, 1870-1914* (Cambridge [Cambridgeshire]: Cambridge University Press, 1981); Paul Readman and Matthew Cragoe, eds., *The Land Question in Britain, 1750-1950* (Basingstoke, England; New York: Palgrave Macmillan, 2010); Paul Readman, *Land and Nation in England* (Woodbridge, UK: Boydell Press, 2008); David Steele, *Irish Land and British Politics* (London: Cambridge University Press, 1974); and L. Perry Curtis, *Depiction of Eviction in Ireland 1845-1910* (Dublin: University College Dublin Press, 2011).

[4] Ryan Vieira, *Time and Politics* (Oxford: Oxford University Press, 2015).

[5] Figures 1-3 were coded by Jo Guldi.

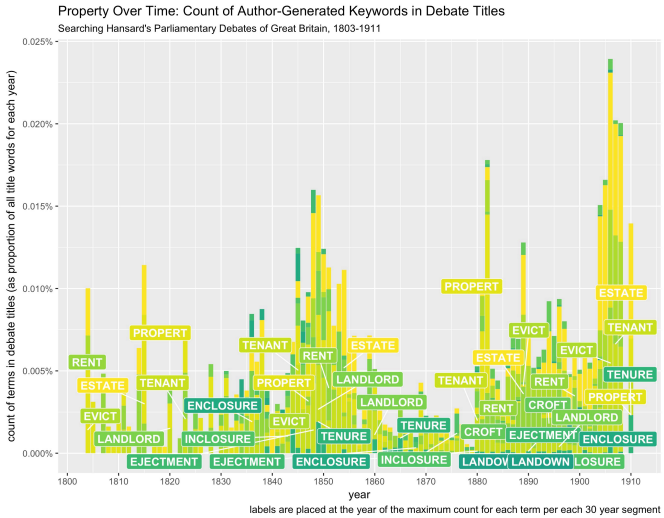


Figure 1.

There are problems with this search, however, for the visualization as a whole obscures conflations and exceptions. The keyword "tenant" pulls in not only those results about eviction, but also those about agricultural produce, property taxes, tithes, the franchise, and many other subjects only loosely related to relationships to property. Other general terms, for instance, "estate" or "landlord," similarly cast too wide a net, and the resulting list tells us very little about what changed when, how and why. Another omission is geographical and racial: the absence, from this list, of any words having to do with Indian or African tenure as opposed to Scottish, Irish, and English.

The scholar might try to enhance the results of Figure 1 by attempting to understand better the fate of individual words. Adding more information can refine this process towards a more reflective approach by, for example, using another scholarly apparatus—in this case the Oxford English Dictionary—to suggest the full variety of terms for property. In the revision, the search follows terms that include local relics of feudal culture (for instance "udal") and the lexicon of imperialism (for instance "zemindar"). From the OED keywords, the top ten most frequently-occurring terms will be plotted over time.

Revising the search process allows the scholar to investigate some additional questions of method and interpretation. Because the research question concerns displacement, the inquiry is unrelated to mentions of "rent" or "tenants" that pertain agricultural commodity prices, household taxation, or the franchise. One way to

zero in on discussions of debates is to limit the inquiry to the ten debates in which each term is present the *most,* rather than every mention of landlords. Such a shift of perspective would have the benefit of examining the changing language in those debates that hinged the most on a lexicon of property.

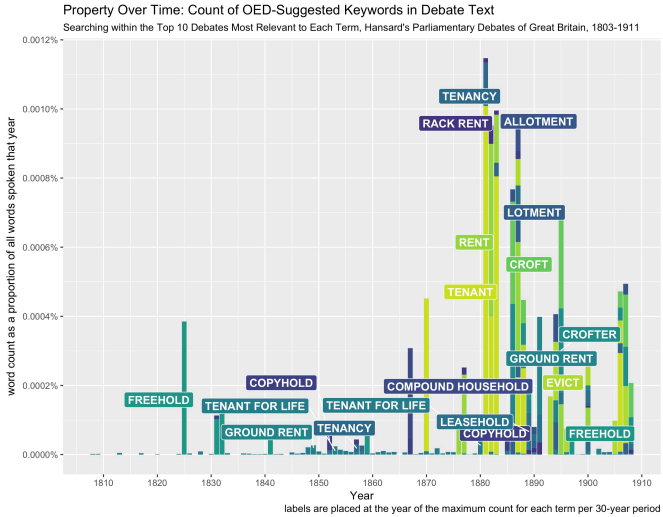


Figure 2.

The results of this third iteration (Figure 2) reinforce the sense of a transition from market-related terminology early in the century, where the language of "freehold" and "landhold" early in the century gave way to a different conversation about the "tenant" and "rent," mediated by the emergent concept of "ground rent." The later lexicon of property is more diverse, with a prominent discussion of the "crofter" and other tenant victims of "eviction," together with remedies such as "allotment," cresting between 1875 and 1890. The revised search reinforces the impression of an enormous explosion in mentions of tenant-landlord relationships around rent after the Irish Land Act of 1881), with a corresponding two-to-ten-fold increase in the usage of the new terms over the old, measured as a proportion of all words spoken in the period.

Like the first figure, however, Figure 2 is marked by the eurocentrism that typifies most debates in parliament, even in the age of empire. While the original list of keywords included "zemindar" and "ryot," counting the most frequently used words returned no Indian terms for landholding. Further approaches to the lexicon would be necessary to interpreting when and how any shift occurred in the lexicon of Indian, African, Pacific, or North American property.

From the viewpoint of a historian trying to recover the history of eviction in different times and places, the search still remains incomplete. The scholar could proceed by attempting another version of keyword search, perhaps selecting only the Indian terminology from the OED, or sampling only those debates that refer to India, Africa, Australia, or Canada by name in the title. Indeed, term-based search may be the wrong key to unlock the door of global eviction. Such a problem as this one requires the scholar to carefully consider the nature of the document base, to reckon with her choice of keywords, and even to consider other possible algorithms that might solve the riddle in a different way.

The foregoing discussion demonstrates both the necessity and the difficulty of critical thinking about digital searches. Neither the process of forming a question, nor the interpretation of that question, is automatic. Every word in a simple keyword search is open to unpacking; each group of words obscures others that might be elucidated by further searching, study, and reading.

Critical thinking about the words that supply a digital search lends strength and rigor to our research process. A process of critical engagement allows the scholar to correct for the proclivity to overinterpret a particular chart, that is, the tendency to construct a thesis from a single illustration of discontinuity. Iterative approaches and multiple tools are essential for controlling for the scholar's own subjectivity in encounters with the archive.

The two graphs in the introduction to this article demonstrate not a trajectory towards some ultimate treatment, but rather the ambling, iterative course of exploration that a scholar might take. Certain truths are revealed by one graph and other truths by another. Each illustration has limits which give way to new research questions, and the scholar must ultimately reconcile the findings of all the interventions to her original research project. The succession of charts forms a path through the data, as the scholar explores dimensions of the archive, explaining those findings to the reader. In the journey of critical search, the scholar engages with critical thinking at every step. The foregoing example also illustrates how multiple measures complement each other, enhancing the scholar's sense of *what* is being measured and *how* a particular search illuminates and disguises various dimensions of a canon.

This article calls for a critically-informed strategy for negotiating digital archives that is aligned with an understanding of how different algorithms determine particular perspectives on textual corpora from the past. Recently, for instance, Daniel Shore has shown how a dozen different algorithms produce a dozen different versions of the past.[6] Understanding digital algorithms as having this per-

[6] Daniel Shore, *Cyberformalism: Histories of Linguistic Forms in the Digital Archive* (Baltimore:

spectival ability to open up different dimensions of an archive reminds us that no search is complete until all of its aspects—the choice of keywords, the algorithm, the exceptions, and the particular texts taken as exemplary evidence of the result—have been subjected to iterative examination.

A call for critical thinking throughout the search works at odds with an empirical and scientific posture often taken by humanists who engage with digital tools, for instance in what we might call "proof of concept" articles in which a new tool is introduced to a new field. Such articles often stress the scientific correspondence between computerized generalizations and the reality of the archive, in order to validate a new method in the field.[7] In so doing, and stressing the "discovery" aspect of a new method, tools typically stress the unified nature of the reality produced by a particular archive and tool. It is unsurprising then that readers attached to suppressed voices from below might resist such tools, wondering indeed if they are instruments of a renewed imperialism of history by the pseudo-scientific fact.[8]

How digitally-enabled historians engage with macrohistory thus raises important issues about the interpretation of digital findings. Is the role of digital tools in "distant reading" necessarily to reveal a single, Apollonian, and definitive perspective on an archive or period of time? This article attempts to model a general process by which a scholar can approach the perspectival nature of algorithms. It argues for a critical, interpretive approach to digital tools based on iteration, where the scholar constantly uses the results of digital inquiry to investigate the question of what different options propose or produce. It asserts that each digital tool and the parameters with which is it is used provides its own perspectival approach on the vying lexicons, grammar, and ideas of the past. It urges a critical approach to the use of these digital tools, which is modeled in terms of three "macro-steps" in a process of engagement, choosing a seed text from the secondary literature, winnowing the results of the search, and guided reading in the results of the winnowing.

This article, therefore, proposes that the solution for better text-mining is not another algorithm, but a new attitude among scholars engaging with digital tech-

Johns Hopkins University Press, 2018).

[7] For instance, Kellen Funk and Lincoln A. Mullen, "The Spine of American Law: Digital Text Analysis and U.S. Legal Practice," *The American Historical Review* 123, no. 1 (February 1, 2018): 132-64; Matthew L. Jockers and David Mimno, "Significant Themes in 19th-Century Literature," *Poetics* 41, no. 6 (2013): 750-769; Lauren Klein and Jacob Eisenstein, "Reading Thomas Jefferson with TopicViz: Towards a Thematic Method for Exploring Large Cultural Archives," *Scholarly and Research Communication* 4, no. 3 (2013).

[8] For instance, Roopika Risam, "Beyond the Margins: Intersectionality and the Digital Humanities," *Digital Humanities Quarterly* 9:2 (2015).

niques, one that is modeled here under the phrase, "Critical Search." Critical search, like critical thinking, employs archives from a set of pre-existing social and political concerns, brokered through skepticism about the shifting meanings and hidden voices within any archive. As this article will argue, scholarship requires, above all, a careful process of iterative examination of the corpus, and iterative investigation, and the world of research by algorithm should follow these practices as well. No single text-mining technique—whether topic modeling, keyword searching, or statistical measures—is by itself sufficient for filling out social categories such as eviction, which was referred to under a variety of terms, many of them so general as to be eluding. Such an approach would fit entirely within a strain of historical method that sees historical agency as multiple and overlapping, and the task of historical interpretation as resurrecting and identifying the competing strains of agents vying for the past. Another perspective might assert that digital tools are generative and perspectival. They create visions and versions of the past that let us see other things.

Defined as an approach that incorporates critical thinking into the research process, Critical Search does not depend on a particular algorithm or set of algorithms, but rather suggests how questions of interpretation and scholarly selection permeate the entire process of applying digital tools and using their results. It argues, generally, for multiple and iterative engagement with different algorithms and tools for the purpose of generating a multiple and overlapping perspective on how different actors' lexicons, grammars, and discourses were changing across different periods of time.

The digitally-engaged process of Critical Search described in this article is designed to mirror a traditional history seminar, where students move from an assigned syllabus to a broader set of readings around a topic, followed by the identification of particular case studies for further reading. The student of history typically begins with some process of grouping together reading material based on her interests. Gradually, her reading moves wider, usually by using a variety of prostheses whose nature she understands well, including a card catalog (or now, more typically, its digitized version), or the research assistant tasked with assembling some primary sources for my next syllabus. If she does her job well, the resulting base of sources she uses is perfectly tailored to reflect her research question.[9] The social historian is thus faced with an enormous mass of

[9]The virtues of this expansive contextual reading, broad sifting of sources, and synthesis of evidence from different points of view have recently been articulated in the many-authored "Tuning Project" of the American Historical Association. See "AHA History Tuning Project: 2016 History Discipline Core," (accessed June 1, 2016). The critical search process proposed here, while crucial to the process of summarizing and dating events in history, is nevertheless fairly irrelevant to many fields in the humanities and information sciences. Critical search, as described here, would be alto-

possible records from which meaning could be extracted; hence the legitimacy of microhistorical approaches to social questions that follow individual lives and families as microcosms of larger dynamics of gender and class.[10]

Faced with imponderable archives, scholars need to use tools both to generalize and to narrow. Digital tools may help them to generalize about a corpus. Other tools may help them to identify particular texts that are symptomatic of larger trends, and to speak in specific ways about how a particular passage or set of words is exemplary of a larger whole. They may need to divide up massive archives into subcorpora, and to generalize within these smaller corpuses about the voices and trends they find there. For instance, historians who deal with official corpora need to be able to characterize what Robinson, Gallagher, and Denny called the "official mind" of the state, as well as to identify the texts recorded by the official mind, in the form of testimony, survey materials, and anthropological description, that reference social experience as it both resisted, remade, and was refashioned by the state.[11] Such tools as these would afford the digitally-aided scholar a set of advantageous techniques for the recovery and analysis of social experience through the mass-digitized archives so widely available today.

Humanistic research increasingly operates on collections where the scale of texts involved defies indexing by hand and results, increasingly, in the reliance upon technologies such as topic modeling as an intermediary between the researcher

gether unnecessary for a student of canonical politics who already knows the names of the actors who matter to him. Likewise, a student interrogating the female literary voice may only need to collect fifteen examples of novels by women to generalize her conclusions. The social historian, however, is responsible for portraying the range of voices related to a particular category, as well as adequately understanding the period for which her query is relevant, ideally by dating the first and peak expressions of her subject. For the literary scholar, mass extraction is irrelevant, and for the sociologist, broad winnowing of the scholarly record is unnecessary. For these reasons, the model of critical search proposed here differs from a more humanistic conception of research, for instance the one formulated by John Unsworth, where the choice and analysis of passages text from an already constrained sample – rather than the discovery of an appropriate subcorpus from an unreadable mass – is the critical factor under consideration. Unsworth describes a seven-fold list of unordered primitives, including"discovering," "annotating," "comparing," "referring," "sampling," "illustrating," and "representing," tasks that are suitable to a small collection such as the Blake Archive on which he was working at the time.

[10] An excellent recent example being Seth Koven, *The Matchgirl and the Heiress* (Princeton: Princeton University Press, 2016). The opposite approach, of course, also has validity: approaching the official record with the intent of extracting a case of how the assorting and abstracting mechanisms of modern government remade the life of the peasant. "Paradoxically, history from below may be (as mostly it has to be) achieved by examination from above," mused historian Peter Robb of his use of British state records to study the peasant of India. Peter G. Robb, *Ancient Rights and Future Comfort: Bihar, the Bengal Tenancy Act of 1885, and British Rule in India* (Richmond: Curzon, 1997), xxi.

[11] Ronald Robinson, John Gallagher, and Alice Denny, *Africa and the Victorians: The Climax of Imperialism* (Garden City, N.Y.: Anchor Books Doubleday, 1968).

and archival truth.[12] Scholars such as John Unsworth and Timothy Tangherlini have modelled the fit of the digital in the humanities by focus on the tasks of collecting and indexing.[13] As more researchers turn towards such technological intermediaries, certain agreements about the importance of critical awareness, the conventions of reviewing algorithm findings, and the documentation thereof become critical if researchers in the humanities are to function as a community.

Information retrieval is the subject of an extensive literature in library science, typically reduce the retrieval process to a single algorithm (for instance, tf-idf, the measure of terms relatively sparsely disseminated overall that are expressed in particular articles).[14] One of its conventions is the profiling of particular tools, one at a time. Journal articles about the digital humanities have frequently treated one digital toolkit at a time: consider how Lauren Klein introduced the topic model with the letters of Thomas Jefferson, or how Funk and Mullen explained the analysis of textual re-use in the American legal code.[15] In order to accept new knowledge provided by the abstraction of the of the topic model, readers need to first be persuaded that the abstraction of the topic model in some way provides new knowledge and that that knowledge can be verified in comparison with other, traditional means of learning about historical corpora. Most digital history articles to date, including some that I have written, conform to this extremely reductive convention of proving that a single tool is useful for understanding the past. But that convention need not dominate how we publish about the humanities, and it should not reduce our capacities to think in methodologically plural ways about the past.

In contrast to digital scholarship that profiles a single approach to the archive, this article will emphasize what happens as scholars move between questions, tools, texts, and provisional answers. This emphasis on the praxis of investigation represents an adjustment in approach from the strategy typically laid out in journal articles about digital history where a single tool is recommended for abstracting

[12] Gheorghe Muresan and David J. Harper, "Topic Modeling for Mediated Access to Very Large Document Collections," *Journal of the American Society for Information Science and Technology* 55, no. 10 (August 1, 2004): 892-910.

[13] John Unsworth, "Scholarly Primitives: what methods do humanities researchers have in common, and how might our tools reflect this?", in Symposium on "Humanities Computing: Formal methods, experimental practice" (King's College, London, May 13, 2000); Timothy R. Tangherlini, "The Folklore Macroscope: Challenges for a computational folkloristics," *Western Folklore*, 72(1) (2013): 7-27.

[14] Karen Spärck Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval," *Journal of Documentation* 28 (1972): 11-21; Stephen Robertson, "Understanding Inverse Document Frequency: On theoretical arguments for IDF," *Journal of Documentation* 60:5 (2004): 503-20.

[15] Klein and Eisenstein, "Reading Thomas Jefferson with TopicViz"; Funk and Mullen,"The Spine of American Law".

knowledge. To the degree to which we have a sense that it's appropriate to only apply one digital method at a time in analyzing digital corpora, that way of thinking reflects a convention that has arisen in journals around our conventions of explaining truth, a convention that is at odds with more sophisticated historical practice.

To interpret the range and depth of social experience requires an embedding of reductive techniques in a multiple-dimensional perspective on the past, which practice constitutes what I am calling critical search. What this means, in practice, is the integration of digital tools that *abstract* and *reduce* to a particular dimension—for example topic modeling with is abstraction of discourses of related words—with other tools that draw the reader's attention to still other dimensions of the text. A complementary and iterative approach thus provides an antidote to digital history as reductionism: when consulting topic models, it pays to use secondary sources to consider which texts to topic model and how to interpret the results; when using keyword search, it pays to use secondary sources to interpret the keyword search. Similarly, when topic modeling, it pays to also to reference keyword searches and keywords in context. Complementary and complicating abstractions and simplifications help the historian to convey the depth and breadth of their understanding of past events for the reader: and that has always been the challenge and promise of writing history.

Recently, certain scholars have pressed back against the reductionism of their methods, especially the assumption that scholars enter an archive already armed with an exhaustive understanding of the keywords, personal names, places, and dates that matter in a research project.[16] Indeed, recent standoffs over methods and theory have frequently taken the form of scholars insisting that new insight about historical change depends not merely upon the collection of new facts, but also upon insight into the changing history of human agency and institutions, realizations often opened up through engagement with critical theory.[17]

Critical search, as this article defines it, emerges from in between polarized positions of debate where some parties insist on critical thinking to the exclusion of new methods, and other voices focus strictly on the praxis of the method. One

[16] For instance, Nina Tahmasebi and others have investigated more generally the role of *a priori* knowledge in specifying the insights to be gained from the analysis of texts. Nina Tahmasebi et al., "Visions and Open Challenges for a Knowledge-Based Culturomics," *International Journal on Digital Libraries* 15, no. 2-4 (April 1, 2015): 169-87. There also exists a critique of this reductionism from within library science, for instance, Caleb Puckett, "Oh, the Humanities: Understanding information behavior to foster information literacy," *Emporia State Research Studies*, Vol. 46, no. 2 (2010): pp. 33-43.

[17] Ethan Kleinberg, Joan Wallach Scott, Gary Wilder, *Theses on Theory and History* (http://theoryrevolt.com, accessed June 1, 2018).

perspective is a reasoned appreciation of the power of algorithms in the humanistic research process. Another perspective is the critique of reductionism and the dangers of technological dependence given the multifarious nature of humanistic research. A third is the insistence upon the importance of critical perspective on agency and identity in the human past. From these three points of view emerges the necessity for critical search.

The Critical Search Process

In the process described in the rest of this article, an ideal set of stages alternating between algorithms, reading, and reflection formulate the humanist's research process. Critical search begins with critical reading of the sources, which then undergirds algorithmic modeling. Scholarly interpretation and computer-aided modeling alternate throughout the scholar's task. Modeling is then subjected to supervision, boundary-making, and guided reading, wherein a scholar then inspects the results of the algorithmic model for its accuracy as well as the vantages it opens on the material at hand. Statistical inspection of the results reveals the bias implicit in particular models. Documenting the state of the project in each of these categories opens the door to a truly transparent model of the scholarly project in a digital age.

The approach offered in this article—critical search—advocates that researchers proceed past preliminary reductionist models such as keyword searching and topic modeling, weaving together statistics, information theory, hermeneutics, critical theory, and critical reasoning into a model process, each step of which is open to inspection. The point of modeling a "critical search" is to offer reflections on a process of *narrowing* common to traditional research projects, which can guide the digital world in which researchers routinely need to constrain a large corpus around a particular question.

The categories of *Seeding, Winnowing,* and *Guided Reading* describe a sequence of research familiar to many professionals, who under the influence of method or theory constrain and broaden their reading on the basis of their findings. The resultant process typifies three general stages of opportunity for critical reflection on the search process and how the scholar has engaged with algorithms and primary and secondary sources—three places where those choices and can be usefully documented and described for other practitioners of history. In greater detail, the suggested categories are as follows:

Seeding.

The first question is what archive one addresses, and which known primary sources, dates, figures, or concepts govern the orientation to that archive. Considering these questions lends itself to a metaphor of inspecting and planting keywords, dates, and ideas passed on from elsewhere. The scholar's choices about which to engage necessarily change the shape of the later inquiry: patriarchal words will rarely reveal subaltern attitudes, for instance. In a digital process, the search process is generally also seeded with a choice of algorithm(s)—a topic model, keyword search, or statistical measurement of significance according to some abstraction. This process too will tend to shift the search in one direction or another, lumping or splitting the corpus according to some general mathematical or theoretical concept of discourse, lexicon, or cluster. Carefully documenting and discussing the choice of seeds—whether conceptual, semantic or algorithmic—represents a first opportunity for making transparent the search process.

Broad Winnowing.

Winnowing suggests work with the maturing fruit of a first round of searches, roughly working over the returns of some query to sort the wheat from the chaff, and discarding the less relevant options. In traditional research, the scholar chooses particular exemplary texts or characters for close reading only after engaging a wide variety of primary and secondary texts that allow her to map out how unusual they are; the researcher proceeds by working the source base to present other examples. In digital research, where an enormous corpus is generally present in every case, a researcher winnows with the algorithm, tuning and applying it to the results, testing how consistent are the results and how adequately they can be interpreted. In the process of working available algorithms and queries, the researcher will likely throw away many false positives or pieces of messy data, possibly trying the same search a dozen times with cleaner data and clearer results. Winnowing presents an opportunity for transparency in the documentation of how specific questions or algorithms work with a particular dataset and question.

Guided Reading.

Only at a mature stage of research does a researcher "harvest" the fruits of a research process as evidence of a shift over time, just as a gardener turns through tomatoes, discarding the ones too unripe, too moldy or bruised for consumption. In traditional research, a scholar inevitably discards or saves for later episodes from her work that are irrelevant to the research question as it becomes more and more targeted. In digital research, scholars must also choose which results bear not only upon the readership but also upon some historical question for her readership. The harvesting process in itself is laden with bias, and presents an opportunity for a scholar to explain in passing what was left out, and how much of the results as shown are the work of human sorting rather than the automatic detection-work of some algorithm.

Critical search thus humbly models the everyday interventions of traditional research and digital research, dividing them into the course of relatively natural seasons, each of which demands work, affords results, and offers an opportunity for transparent documentation of the choices made by scholars.

The process of critical search may be highly eclectic, and need not copy the steps laid out here: to engage in critical search is merely to insist on inquiring into the biases of different digital tools and their results at every step, constantly testing them and revising them with documentation. Later parts of this article will discuss, for example, an iterative research process that required successively reseeding, re-winnowing, and re-reading resulting samples of text from a corpus. To refer to portions of the model as stages or "seasons" is not intended to delimit or constrain. The resultant process may be either replicated simply (the way that by repeating a cookie recipe one procures cookies), repeated iteratively, or worked into the flows of inquiry that fork and take on new shapes with each pass. To divide them in three is merely to signal the many opportunities for documenting scholarly choice, and the biases that come with choice, as they are passed onto the next phases of work with data.

The bulk of this article explains, in greater detail, what the seasons of research look like in traditional and digital forms, and how algorithms and their the "fitting" to exploratory data analysis becomes part of the model. The article also follows the process of a critical search for texts about property in the parliamentary debates of nineteenth-century Britain as a case study. Provisional technical solutions form part of each of the three stages of critical search, but the algorithms presented here are deliberately chosen for their interpretability rather than as an assessment of the best algorithm from the continually evolving world of computer science.

Seeding with Words and Documents

Most scholars in some way start with names, dates, concepts, and words passed on from elsewhere, like those seeds that Indian peasant women sew into the hem of their garments for safe keeping from generation to generation. A strong component of scholarly research is likewise informed by tradition. Traditional scholars frequently renew a line of questions that previous generations posed before them, as when Peter Mandler opened one volume on reform with questions about the Constitution proposed by Oliver MacDonaugh.[18] Digital scholars, likewise, frequently consult earlier generations of Victorianists when determining which hand-picked keywords to follow for a quantitative study of Victorian moral ideals and how they changed.[19] Other scholars have started with contemporary documents, rather than words, using algorithmic matching to generate another set of documents linked by explicit textual re-use or similar thematic ideas.[20]

When a gardener goes to plant, she looks over the seeds and carefully chooses a few from a store. As those seeds begin to grow, she examines the hardier and weaker ones. Just so, the historian of Britain approaching Hansard's parliamentary debates may find herself pondering the categories that earlier generations of scholars have employed, and asking: Are the fundamental questions for text-mining parliament those of party, of individual personalities, of gender, race, or class, or of democracy or some other concept in general? Her choice is necessarily critically informed by changing theories in the discipline as well as her own temperament, politics, and interest.

The curation of the words and documents in any scholarly process tilt the results of the inquiry with the bias of the scholar and her world. From theory, from secondary readings, from prior knowledge of the canon, and from her own politics, the scholar approaches the vast unread with bias, that is, with certain ideas

[18] Peter Mandler, "Introduction," in Mandler, ed., *Liberty and Authority in Victorian Britain* (Oxford; New York: Oxford University Press, 2006).

[19] Frederick W. Gibbs and Daniel J. Cohen, "A Conversation with Data: Prospecting Victorian Words and Ideas," *Victorian Studies* 54, no. 1 (2011): 69-77; Bob Nicholson, "Counting Culture; or, How to Read Victorian Newspapers from a Distance," *Journal of Victorian Culture* 17, no. 2 (June 1, 2012): 238-46; Thomas Lansdall-Welfare, Saatviga Sudhahar, Justin Lewis, James Thompson, and Nello Cristianini, "Content Analysis of 150 Years of British Periodicals," *PNAS (Proceedings of the National Academy of the Sciences)* 114, no. 4 (January 9, 2017): E457-E465.

[20] Tangherlini and Leonard, "Trawling in the Sea of the Great Unread"; Funk and Mullen, *"The Spine of American Law"*.

about which documents are the most important, which category of ideas, feelings, or names the best witness to change which tend to color the results. Those choices, in turn, govern all later computational output, and constrain the results of analysis in an important direction. The reality of bias is in no way different in traditional and digital search, for at every instance where the work of curation takes place, the scholar is generating an a priori reference point that shapes the search process.

Digital scholars have replicated this process of relating the "great unread" to a known canon of texts, figures, and events, sometimes by measuring the difference between canon and archive in aggregate, and sometimes by beginning with a familiar reference text and using algorithms to discover the most similar documents.[21] The latter approach implies, in Timothy Tangherlini's metaphor, that the researcher uses a canonical text as the "hook" to go "trawling" in the "ocean of the great unread."[22] In the proposed process, one known text becomes the basis for using other topic models as a finding aid by which to collect an assortment of other novels with similar content or themes.

In critical search, the biases that governed the choice of keywords, documents, and dates need to be made explicit and self-reflective as possible because they have such strong downstream effects. The work of bias in digital search may be examined by the process of "seeding" a search with particular keywords, where the scholar begins by counting keywords over time and then uses those counts to identify particular trends, documents, or passages for further reading.

A search seeded by keywords provides ample material for historical analysis, where a research question is already extremely narrow, for example, if we want to find every instance where newspapers mentioned Gladstone or to count conservative invocations of "manliness," as Luke Blaxill has done.[23] Critical thinking about Victorian gender and its presentation in parliament has provided a narrow set of seeds, appropriate to the time and place covered by the data, and the scholar has only to plant the seed and reap interesting results. In very specialized cases of this kind, further winnowing may not even be necessary before moving from keyword counts to further conclusions.

When scholars engage an abstact concept such as property or eviction, they may have a hard time distilling a broad discourse about property into a single keyword

[21] Mark Algee-Hewitt et al., *Canon/Archive: Large-Scale Dynamics in the Literary Field*, 2016.

[22] Timothy R. Tangherlini and Peter Leonard, "Trawling in the Sea of the Great Unread: Sub-Corpus Topic Modeling and Humanities Research," *Poetics*, Topic Models and the Cultural Sciences, 41, no. 6 (December 1, 2013): 725.

[23] Luke Blaxill, "Quantifying the Language of British Politics, 1880-1910," *Historical Research* 86, no. 232 (May 1, 2013): 313-41.

that measurably changed over time with interpretable results. The choice of any particular keyword tilts the results in particular directions. A scholar may, therefore, choose to begin by following a broader class of keywords, for instance "rent" and "tenure," but these approaches cause new problems. The keyword "tenant" includes not only debates about eviction, but also those about agricultural produce, property taxes, tithes, the franchise, and many other subjects only loosely related to relationships to property. Other general terms, for instance, "estate" or "landlord," similarly cast too wide a net, and the resulting list tells us very little about what changed when, how and why. Using these words as an index for debates draws attention to the wordiest debates, or the discussions characterized by their use of scholar-defined terminology, but not necessarily the ones in which the words' usage changed the most.

	debate	year	speechdate	wordcount
1	IMPROVEMENTS.	1883	1883-07-17	674
2	COMMITTEE. [FIFTH NIGHT.]	1881	1881-06-02	539
3	LAND TENURE BILL.	1906	1906-11-12	526
4	IRISH LAND LAW BILL [Lords]. [BILL 308.]	1887	1887-08-01	479
5	LAND PURCHASE (IRELAND) BILL.	1888	1888-11-27	479
6	IMPROVEMENTS,	1883	1883-07-19	462
7	SECOND READING. [FIRST NIGHT.]	1881	1881-08-01	452
8	ARREARS OF RENT (IRELAND) (recommitted) BILL. [BILL 213.]	1882	1882-07-11	415
9	SECOND READING. ADJOURNED DEBATE. [SECOND NIGHT.]	1870	1870-03-08	398
10	SECOND READING.	1870	1870-06-14	397

Table 1. Debates with highest wordcounts for "tenant" in Hansard, 1803-1908.

	debate	year	speechdate	wordcount
1	INDIA TENURE OF LAND IN MADRAS.	1854	1854-07-11	48
2	GOVERNMENT OF INDIA.	1853	1853-06-03	9
3	OBSERVATIONS.	1861	1861-05-31	7
4	TORTURE IN MADRAS.	1856	1856-04-14	6
5	GOVERNMENT OF INDIA ADJOURNED DEBATE.	1853	1853-06-06	4
6	REVENUES OF INDIA.	1856	1856-04-18	4
7	GOVERNMENT OF INDIA BILL ADJOURNED DEBATE (FOURTH NIGHT).	1853	1853-06-30	3
8	Madras Ryotwari System.	1902	1902-07-29	3
9	THE GOVERNMENT OF INDIA.	1853	1853-02-25	3
10	INDIAN FAMINE COMMISSION.	1902	1902-02-03	2

Table 2. Debates with highest wordcounts for "ryot" in Hansard, 1803-1908.

	debate	year	speechdate	wordcount
1	IMPROVEMENTS.	1883	1883-07-17	674
2	COMMITTEE. [FIFTH NIGHT.]	1881	1881-06-02	539
3	LAND TENURE BILL.	1906	1906-11-12	526
4	IRISH LAND LAW BILL [Lords]. [BILL 308.]	1887	1887-08-01	479
5	LAND PURCHASE (IRELAND) BILL.	1888	1888-11-27	479
6	IMPROVEMENTS,	1883	1883-07-19	462
7	SECOND READING. [FIRST NIGHT.]	1881	1881-08-01	452
8	ARREARS OF RENT (IRELAND) (recommitted) BILL. [BILL 213.]	1882	1882-07-11	415
9	SECOND READING. ADJOURNED DEBATE. [SECOND NIGHT.]	1870	1870-03-08	398
10	SECOND READING.	1870	1870-06-14	397

Table 3. Debates with highest wordcounts for "croft" in Hansard, 1803-1908.

"Tenant" also highlights a peculiarly Irish subject-matter, where the Irish Catholic tenants of English landlords became the subject of debate in the Land Laws of 1881 and 1887 and their later amendments. Careful seeding, therefore, requires at a minimum that the scholar use words that highlight some of the geographical diversity of the property issue in parliament, for example keyword searching for Scottish "crofts," and Indian "ryots" (Tables 1-3).[24] Even so, the scholar's bias towards well-documented territories like Scotland, India, and Ireland necessarily constrains the inquiry, with the native tenures of South Africa, New Zealand, and Jamaica nowhere in this list. The bias of the search terms still structures the results.

It becomes incumbent on the researcher to devise a way of moving from a larger list of terms for property to particular turning points. The researcher cannot move hastily from a list of keywords to the full collection of debates about property, then extracting place-names and proper nouns, for a reliable and exhaustive search of a particular corpus.

In some processes, it is not a word or a name that operates as a "seed," but rather a collection of documents whose lexicon will be "matched" by some algorithmic process. In this kind of work, the justification of the seed texts unambiguously colors the results that will later be returned, and the scholar must justify those choices in a discussion. In the case of studying texts about property and eviction in Hansard's parliamentary debates of the UK, four reports relatively well-cited in the secondary literature were chosen as a "seed." They were chosen to eliminate bias, given relatively geographic representation - two reports for Ireland, one for Scotland, and one for England—and chronological representation—one from the 1840s, and three from the 1880s. A scholar writing about the results of the algorithm in question cannot present her work as final: "seeding" the process with slightly different texts or excerpts from those documents would result in totally different findings.

A seed can even go beyond a definite keyword or text, to be an evolving category. For Tim Tangherlini, the seed was known works of fiction in Danish literature—a static category—and topic models were used to harvest less-known works on the same subject. For Simon DeDeo and Rebecca Spang, the seed was "the future" relative to each year in the *debats* of the French Revolution—an evolving category—and statistical divergence was used as a measure of how much any speaker anticipated what the French government would be talking about in months and years to come. The choice of seed documents—whether canonical reports, or

[24] Tables 1-3 were coded by Jo Guldi.

futurity itself—governs intimately the analysis that follows.

The point of identifying seeding as a particular phase in the process is to call scholars' attention to the need to document and justify the determinative choices made at the beginning of research. The goal is not to eliminate bias altogether, but to raise the reader's critical awareness of the exclusions implied by certain choices of words or documents, thus opening up later inquiry by others about how the algorithms might have been programmed to return different results.

A critical search with keywords or documents moves from naïvely proposing a word or document and the "answer" to an algorithmic search, to a rich description of why *these words* or *that document* will illuminate further inspection of the past. The process remains potentially fraught and given to further argumentation and inquiry at every stage, and the problem of interpretation never disappears.

The documentation of these choices is not standard practice in history or in the humanities in general. Neither in traditional research or digital research is it common to reference the card catalogue used, the key terms searched for, the dead ends taken, and the routes that were most profitable along the way. Traditional scholars may have felt that their reading habits were more important. In digital scholarship, however, documenting the choice of seeds is crucial to making a query reproducible.

"Reproducibility" of a query may be a new virtue for some humanists, especially for those who work in the realm of interpretation, polysemy, and affect; but even those communities too should appreciate the role that footnotes have long played in documenting conversations and allowing communities of consensus to emerge. In the discipline of History at least, the reproducibility of individual visits to the archive has long been a standard of truth-making. Digital scholars, who regularly share both data and code with others, have an opportunity to convert that standard of truthful, replicable analysis by demonstrating, as they lay out their analysis, that the choices of seed texts and algorithms correspond with both the ideas at stake in their analysis and with the analysis actually performed by their code. Radical transparency about the choices upon which our arguments hinge promises to solidify nothing less than the humanities' role as defenders of mutually-agreed-upon truth—and of multiple possible avenues towards that truth.

In any event, making the invisible choices behind an analysis visible also offers a pedagogical opportunity for authors to educate their readers about how skillful scholarship is done. Transparency also makes the choices of scholarly analysis radically accessible to the classroom, where students of code should be instructed to try out, for themselves, alterative choices of seed texts and algorithms, so as

to better understand the consequences of particular inquiries. For all of these reasons, documenting the choice of seed words, texts, and queries opens up the route for mere search to become critical search.

Seeding with Algorithms

The digital scholar typically uses the computational "match" in the way that earlier generations of historians used a card catalog or a bibliography or book indices: all of these technologies are tools for more accurately discerning an overview of available materials and also finding particular texts that will later form the basis of an argument.

In both traditional and digital research, the process of searching can, in theory, go on forever. Choices have to be made; the process is constrained. One tries a topic model and a keyword search, but no scholar tries out every possible algorithm or variation thereupon.

Trying out more than one approach opens the door to a critical perspective on how each algorithm, or each setting on an algorithm, tilts the results of research in a different direction. The possibility of reading through different prisms allows the scholar to begin to draw conclusions about the corpus that are based, inherently, on the breadth of sampling texts from the entire range of debate.

"Seeding" the process with other kinds of algorithms can illuminate different dimensions of the data, although it cannot escape entirely from reductionism. For instance, the technique of topic modeling has been routinely used to index and analyze digitized textual corpora, from Thomas Jefferson's letters, to American newspapers, to ads for runaway slaves.[25] Topic models identify semantic similarities in collections of words that are used together, and they can even identify words that are used in multiple senses. For this reason, topic models are ideal for dating overlapping, competing discourses that use many of the same terms in slightly different ways. In a 500-topic model of the Hansard debates, eviction shows up among the top keywords for three topics relating to Ireland (Table

[25] David J. Newman and Sharon Block, "Probabilistic Topic Decomposition of an Eighteenth-Century American Newspaper," *Journal of the American Society for Information Science and Technology* 57, no. 6 (April 1, 2006): 753-67; Lauren Klein and Jacob Eisenstein, "Reading Thomas Jefferson with TopicViz: Towards a Thematic Method for Exploring Large Cultural Archives," *Scholarly and Research Communication* 4, no. 3 (2013); Cameron Blevins, "Space, Nation, and the Triumph of Region: A View of the World from Houston," *Journal of American History* 101, no. 1 (June 1, 2014): 122-47.

4),[26] one linked to the disputes over tenants' rights to compensation in the 1850s, another linked to evictions for arrears of rent during the "Land War" of the 1880s, and a third related to the Estates Purchase Act of 1903 and the Estates Commissioners who heard previously evicted tenants' claims to the right of reinstatement. The same abstract keywords—"tenant," "estate," "property," and "landlord"—are employed in slightly different senses in each topic. The topic model thus picks up patterns that would be invisible to the scholar armed only with keyword search.

The power of viewing those words through the topic model is the computer's ability to pull apart slightly different discourses into regular patterns of keyword co-occurrence, each of which has its own chronology. Comparing different topics to each other suggests an evolution of the changing focus of discussions about land reform in Ireland after 1880, as it moved from the evicted tenants themselves, to the use of police and their clashes with tenants delinquent in their rents, to the eventual resolution in the form of state-led buyouts that provisioned former tenants with farms. Topic model thus becomes an aid to discerning discourse rather than keyword usage, and topic models once compared allow the scholar some insight into the life-cycle of long-term competing discourses.

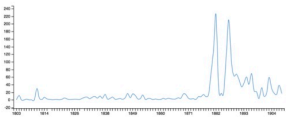
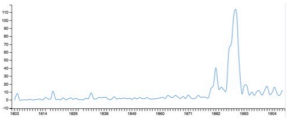
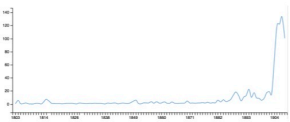
Computer-assigned topic number	Percentage of all debates represented by topic	Scholar-assigned name	Temporality	Top keywords
2461	00.29%	Evictions for past due rent		Tenant-, landlord-, rent-, land-, evict-, case-, pay-, arrear-, hold-, act-, year-, court-, judici-, fix-, farm-, fair-, farmer-, estat-, ani-, law-
886	.0.13%	Police and violence especiall y with regard to eviction		Polic- constable-, chief-, secretary-, policeman-, assault-, policemen-, order-constabulary-, peopl-, men-, man-, evict- occas-, charg-, case-, forc-, arrest-, inspector-, inform-
2415	00.12%	The Birrell Commiss ion and the rights of Irish tenants		Estat-, tenant-, commission-, evict-, land-, purchas-, chief-, hold-, farm-, secretary-, lord-, counti-, sale-lieuten-, reinstat-, case- applic-inform-, receiv-, agreement-, landlord-

Table 4. Eviction topics from a 500-topic model of Hansard, 1803-1908.

Even while the topic model illuminates some aspects of a historical transition,

[26] Table 4 was developed by Jo Guldi on the basis of an interface developed by Poom Chiarawongse, undergraduate, Brown University.

however, it simultaneously masks others. As with keyword search, topic models in this case have obscured the geographical diversity of British empire in the debates. A fine-grained topic model of all Hansard at 500 topics evidences discourses about the British occupation of Ireland, and not in Scotland's Highland Clearances, England's enclosure of the peasant commons, and India's railroad building, but nothing, once more, about Jamaican former slaves and their small farms, or African native tenure. Topic models can produce simplicity by indexing discourses over time, but the simplicity of reduction can be both a strength and a weakness.

The topic model, like the keyword search, becomes most useful in the course of synthesis, when a scholar wants to reduce a research question to a single typical episode of history that may be followed more deeply. When investing in an archive, scholars consult secondary sources, critical thinking, and interpretation; digital scholars need to do as much as well, moving from topic modeling to other dimensions of textual reduction, including other secondary sources and the keyword search. Keyword search and topic model enhance each other by representing separate aspects of a digital corpus: where a topic model abstracts a set of discourses whose representation changes over time, the keyword search, helps the user of the topic model to understand the particular life story of each individual keyword within the topic model. The scholar who wants to understand one dimension of a text is aided by moving back and forth between each abstraction.

The choice of method used to plant seed texts creates biases that will reverberate through the rest of the search process, for instance, modeling the domain of texts as a galaxy in which particular systems of affinity emerge, or as a yardstick with two poles. The booming search engine industry has funded a flourishing tide of studies about matching text to similar texts in machine learning, and here scholars have many tools that they can adopt, from packages for automatically "matching" documents based on a black-box similarity ranking, to software packages that provide similar matches for the technically adept.

In galaxy-type analysis, affinities between texts are represented as clusters, which can be traced by tools such as k-means clustering or topic modeling, where the algorithm has been designed to create probable clusters of documents that mirror human discourses, some large and some small, a principle that typically works for those interested in the discursive nature of a corpus.[27] Humanistic scholars have identified topic modeling with the scholarly reading of "discourses," and defended its logic as compatible with scholarly projects in the humanities.[28] Tools

[27] Tangherlini and Leonard, "Trawling in the Sea of the Great Unread"; Roe et al.,"Discourses and Disciplines in the Enlightenment".

[28] Roe and Gladstone winnow the rhetorical from substantive by working only with the nouns in

that operate on the level of figures of speech, for example, collocation analysis, may be more useful for identifying a *rhetorical* match that finds similar ways of speaking can be differentiated from the *substantive* match about particular subject fields.[29] Another set of tools, classified as "word embeddings," promise to transcend both categories, but have been less routinely studied in the digital humanities.[30] The weakness of all cluster-type analyses is—as with topic modeling—the reduction of texts into generalizations. To counteract the limits of clustering, a scholar can use a yardstick-type algorithm that simplifies the corpus according to a fundamentally different metaphor of abstraction.

Yardstick-type analysis include measures from information theory that impose a spectrum of order onto a corpus, arranging documents according to their linear proximity to some pole represented by another body of text.[31] In many fields, divergence measures have served as a fundamental metric of difference where difference is comprised of many factors whose expression is hard to describe. Divergence measures treat any two texts as a distribution of probabilities and arrive at an artificial *number* representing the distance, based on similar expression of the lexicon as a whole. The flexibility of creating a metric where none previously existed affords the making of structural comparisons in domains where comparison was hitherto available solely on a qualitative basis.

The scholar's choice of algorithm colors the results by revealing different dimensions of the experience of reading, whether topic-driven affinity or divergence-driven polarity. Where a topic model will provide a clustered relationship characterized by affinity groups, using divergence produces a more polarized answer, i.e. the "most close" texts to the seed texts are those that linearly ranked on a spectrum of distance according to how much they literally share the same word-

the *Éncyclopedie.* Would this strategy work as well in a legal context, when verbs and adverbs govern a field of procedures? Glenn Roe, Clovis Gladstone, and Robert Morrissey, "Discourses and Disciplines in the Enlightenment: Topic Modeling the French Encyclopédie," *Frontiers in Digital Humanities* 2 (2016).

[29] Alix Rule, Jean-Philippe Cointet, and Peter S. Bearman, "Lexical shifts, substantive changes, and continuity in State of the Union discourse, 1790-2014," *PNAS* 112:35 (September 1, 2015): 10837-10844.

[30] Ryan Heuser, "Word Vectors in the Eighteenth Century, Episode 1: Concepts." *Adventures of the Virtual* (14 Apr 2016), and "Word Vectors in the Eighteenth Century, Episode 2: Methods," *Adventures of the Virtual* (1 Jun 2016); Andrey Borisovich Kutuzov, Elizaveta Kuzmenko, and Anna Marakasova, "Exploration of Register-Dependent Lexical Semantics Using Word Embeddings," in *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, 2016, 26-34.

[31] Alexander T. J. Barron et al., "Individuals, Institutions, and Innovation in the Debates of the French Revolution," *Proceedings of the National Academy of Sciences* 115, no. 18 (May 1, 2018): 4607-12; Sara Klingenstein, Tim Hitchcock, and Simon DeDeo, "The Civilizing Process in London's Old Bailey," *Proceedings of the National Academy of Sciences* 111, no. 26 (July 1, 2014): 9419-24.

usages. Divergence is the more transparent of the two, as it allows the scholarly to directly explain the relationship between two "similar" texts and the words they share in common.

Whichever approach the scholar chooses, the choice of algorithm is likely to produce entirely different results. Careful comparison of seed algorithms ideally enhances the scholar's sensitivity to different possible dimensions of the archive, for instance, the synthesis of discourses by topic model and the comparison of changing lexicons by divergence. Selecting different algorithms, if done critically and carefully, will direct the scholar to an archival base that does not merely reduplicate the language or concerns of other scholars. To become aware of the particular dimensions of each algorithm is to raise the possibility of documenting for other scholars a critical dimension of scholarship in terms of the particular choices and biases at play and how they can be reworked around different questions.

Each part of this process is potentially open to debate. The choice of keyword search, topic modeling, and divergence measure is to some extent arbitrary, but these three were selected as covering a wide ground of supervised and unsupervised work, and "galaxy" and "yardstick" type of measures.

In the case study of searching for eviction in parliament, both keyword search, topic modeling, and divergence measurement were used for a first pass through the Hansard corpus. Keyword search and topic modeling provided different, complementary answers to when and how the discourse of property changed over time. Divergence measurement resulted in a potential subcorpus of parliamentary texts that were judged by the computer to be close in lexicon to known primary sources about property, and this subcorpus was reserved for inspection later.

Within each algorithm, the scholar selects particular settings, and these too are open to inspection: there's the question of which scholarly terms are searched as keywords; of how many topics the algorithm is asked to return; of how the text is stemmed and stop-worded. If divergence is used, the scholar must decide on a mathematical formula for measuring similarity, whether the object of measurement will be the lexicon as opposed to ngram, skip-gram, or topic; and whether all words will be used or a particular lexicon with general vocabulary stop-worded out.[32]

In the case of the property question, measurements of similarity were taken on the basis of how common each parliamentary debate was based on the measure-

[32] The *philentropy* package in R collects at least twenty algorithms that could be used with different results). See Hajk-Georg Drost, "Introduction," accessed November 5, 2018.

ment of the probability expression of individual words. Measuring the distance between documents in terms of words, it was hoped, would capture a shifting vocabulary of property, whereas multiple-length n-grams might make a more appropriate measurement for rhetorical similarity. Place names and personal names were stop-worded out to prevent the returns from reduplicating the Irish-Scottish geography of the seed texts. Individual debates were compared, as opposed to individual speeches or paragraphs, in the hopes of finding subjects of debate that matched the question of property. Three common mathematical formulae were used that matched those commonly used by other scholars of text mining.

Selecting the seeds is followed by another stage in the process that entails the choice of algorithm for finding patterns in the broader corpus. The point of the next step is to begin the process of winnowing the full corpus of documents to a smaller subcorpus.

Broad Winnowing

Winnowing is an agricultural metaphor for sorting what is valuable to the scholar from what is not. To highlight it as part of a process of critical search is to underscore the fact that in some processes, the results of pattern-recognition are more useful than others, and any choice entails adding layers of bias. Several kinds of winnowing may be required, including adjusting the algorithm and its mathematics to reveal the biases implicit in one assessment over others, and down sampling the results of an algorithmic search. In either case, critical assessment of the winnowing process requires the scholar to explain the choices made in analyzing, preferably taking some measures to document how much the bias was tilted by a particular set of choices.

In this form of scholarship, winnowing typically takes the form of a discursive encounter that justifies the scholar's attention to certain objects in the archive. Traditional scholars make critical choices about their work by engaging in critical theory, sometimes more explicitly than others; in some cases, for instance, historians may depart from a critical reading based on social theory, but at other times, concern with theories of gender, race, and class may inform their approach to historiography and drive them to ask new questions. On the basis of this orientation, they may seek out particular lost voices in the archive, or trace particular encounters. Inn traditional research, winnowing is driven by critical theory and

shaped by scholarly attention.

Winnowing with digital tools, by contrast, begins as a technical question: what adjustments can be made to this algorithm to "fit" my question or my data? What bias does each adjustment confer onto the results of my research? This technical question may be informed by an engagement with critical theory or political ethics, as for the traditional scholar. Indeed, in the case study of a search for eviction in parliament, the original engagement with eviction has its roots in a question about how policy reacted to an era of displacement, and whether the experiences of dispossession reached the corridors of power. That critical question, in turn, drives a technical problem of identifying the legislative debates that concern property and dispossession. In digital research, winnowing is motivated by critical theory and guided by the scholar's skill at matching research questions with the tools of information retrieval.

The scholar may proceed by comparing the results of different algorithms, or by adjusting the settings of particular algorithms, for instance, trying out different divergence measures to get a sense of how varied their results might be. In no case it is clear that there is an objectively *right* setting—a single right metric or right tool, an objectively "right" granularity of topic modeling. Rather, by thrashing the data with different tools, the digital scholar obtains insight into the bias of the tools themselves, and the variety of answers they can produce. Because algorithms are rarely a perfect fit for scholarly questions, the scholar may adjust algorithms repeatedly in the process of honing in on a particular question, all the while learning more about the "fit" of a particular tool. At the heart of this matter is choosing a method that is tight enough that the results usefully answer a scholarly question, while loose enough that the scholar may potentially be made aware of unknown discoveries. These issues bring us immediately to the next two categories.

Winnowing may entail comparing variant settings on a particular algorithm, for example, the precise mathematical formula used to calculate similarity in a divergence measure, so that the scholar understands the bias related to using particular mathematical choices. In the case study of a search for eviction in parliament, three common divergence algorithms were used to measure lexicon similarity between documents. In order to explore the dimensions of chronology opened up by each choice of measurement, the results of the three mathematical formulae were compared. The results of divergence measure also depend upon the user's choice of constraints for boundedness, so different cut-offs of similarity were examined in order to understand some of the different interpretations that might result.

Dedicating a phase of research to studying the effects of the first pass of broad winnowing gives the researcher an important opportunity to specify, critique, and interpret the relationship between the seed texts and the full corpus regarding the raw difference encoded by some algorithm. In the course of this iterative narrowing, the scholar has repeated opportunities to study the consequences of each pass with the algorithm. In the case study about property in Hansard, divergence between the four reports used as seed texts and Hansard as a whole can be represented as a histogram of speeches each of which has a different distance from the seed text. The resulting distribution groups the most similar debates to the left of the x-axis, and the least similar to the right of the x-axis (Figure 3).[33] The y-axis tells us how many debates are in each category: a few are very similar to the seed texts, many are somewhere in between, and a few are very far away. The KL measure classifies three out of the four seed documents as "more similar" to the rest of Hansard compared with the Bessborough Report, shown in yellow in Figure 3.

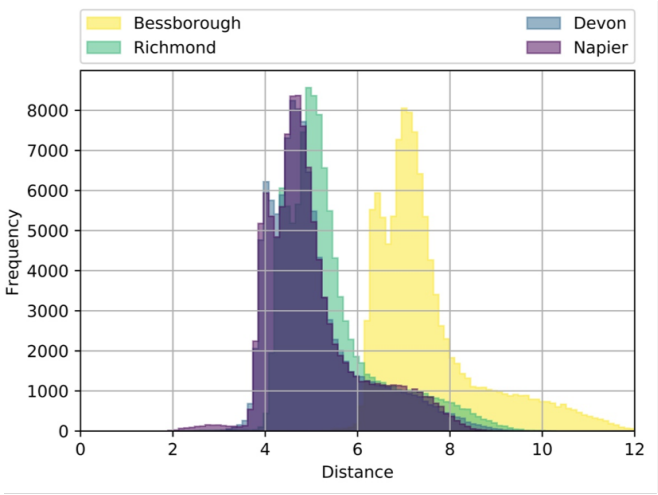


Figure 3. Kullback-Leibler divergence between Hansard debates as a whole and each seed report (distinguished by color).

As soon as a distribution of similarity has been created, the scholar obtains new information about how a particular metric describes the corpus as a whole. This information is critical for transparently presenting the choice of measure.

[33] Figures 3-4 were coded by Ashley Lee of Brown Data Science.

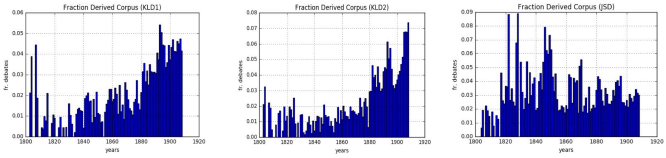


Figure 4. When did parliament debate property during the nineteenth century? Three different measures give three different answers.

A blue histogram gives the number of parliamentary debates classified as the 1% most similar to the Seed Corpus. The x-axis gives the year (1800-1910); the y-axis gives the fraction of all debates that were classed as 1% most similar to the Seed Corpus within that year. The three measures applied are: (A) Kullback-Leibler Divergence, (B) Symmetrical Kullback-Leibler Divergence, and (C) Jenson-Shannon Divergence.

Chronological diagrams help the researcher to understand how choosing a particular measure of similarity may influence the resulting chronology of documents returned. While mathematically similar, different measures of divergence will produce an utterly different sample of historical texts. For instance, compare how three different divergence measures answer the question, "When was property debated by parliament?" (Figure 4). In each case, the histogram shows how many debates are classified as being amongst the debates that are the 1% most similar in their distribution of language when compared to the seed corpus. Three different measures of "similarity" produce three utterly different chronologies of when property was debated. For instance, two formulae for measuring similarity returned results weighted towards the end of the century (Figure 4: a-b), while another formula returned results marked by four outbursts over the same period (Figure 4: c). In some cases, comparing these results will be sufficient to tell the scholar which measure "fits" their question. Broadly winnowing offers an opportunity for the scholar to "check" that the computer's first pass at the corpus is reasonable.

Diagrams that show the distribution of similarity the chronology of the most similar texts can be used along with topic models, top words, and other reductive synopses to judge different algorithms with regard to their "fit" for the question at hand. The chief problem satisfied by distributions and chronological histograms is to raise the scholar's consciousness of a *general bias,* reasonable or unreasonable, associated with unsupervised research associated with one particular similarity measure.

In approaching a problem through critical search, it is the scholar's duty to document those choices and the inevitable bias thereby entailed. Highlighting the bias of different algorithms does not necessarily resolve scholarly questions: it may leave them unanswered for later work. In the case of the search for "property" in *Hansard,* the results of modeling and abstracting the returns from three different divergence measures were far from conclusive. Distributions and chronological diagrams offered little guidance about which mathematical measure was ultimately better suited to the question of distance. The abstraction of mathematical clustering and measuring is simply too far from the scholar's *techne* of critical reading and discourse identification for a best "fit" to be determined by yet another abstraction of the same corpus. Winnowing the mathematics of the algorithm, in this case, may require the scholar to choose a particular approach to the corpus without being able to defend it, while making clear that the resulting chronology is definitively biased thereby.

None of these choices add up to a single model process; the choices made for a case study of property might not work for another inquiry or another archive. In arguing for critical search, the point is not to define an ideal set of algorithms or approaches, but to illustrate how each metric of measurement opens up a potential new window for the interpretation of the past. Whatever the choice of algorithm - whether divergence or another measure of similarity - the approach to the algorithm is crucial. In fact, any classification of texts—including topic modeling and word collocation—could be interpreted in a similar light, as a preliminary binary division of texts that could be refined, over time, through repeated passes through the process of critical search proposed here. Again, the responsibility of the scholar in critical search is that of documentation.

Guided Reading

Just as the gardener picks over moldy and damaged fruit for those good for eating and those good for pie, so too the scholar chooses the quotations with the clearest bearing on her questions. Digital scholars too must reckon with the choice of which findings to present. At this stage in the process, the scholar carefully inspects the results returned by a search process, sometimes sampling them, sometimes generalizing about them (for instance by counting keywords again or topic modeling).

The point of calling the last stage "guided reading" is, of course, to underscore

that the end of critical search is actual *reading* of particular texts, including close examination of particular episodes, characters, turns of phrase, or tensions in the original primary-source documents. The visualizations and timelines produced at earlier points in the analysis may provide context, but they usually do not offer the end goal: an understanding of history. For most scholars, that understanding will only feel complete once they have both a macroscopic overview of change as well as some understanding of the individual lives and struggles of some micro-historical encounter.

The guided reading stage of critical search thus refers to the process of moving from algorithms to particular texts, followed by reading and interpreting them with the skills of a traditional researcher, that is, with critical thinking. The researcher uses the contextual overview of some algorithm to identify particular documents for inspection. A minimal exercise in guided reading would be to follow a keyword search to the year when a particular term took off, that is, the first year that saw exponential growth in the appearance of that term. Within that year, the researcher might locate the three documents of the corpus where those keywords form the highest proportion of words. She would read with confidence that these documents were likely candidates for having influenced others in their use of the term.

The promise of guided reading, which follows algorithmic contextualization, is the detection of documents that bear a significant relationship to a question. Based on the foregoing exercise in contextualization, the researcher can encounter the documents with confidence that they offer a potentially meaningful expression of the words in question.

At each stage of sampling and reading, the researcher gained insight into the promise and shortcomings of different algorithms. At the same time, she also gathered and read documents that the algorithm has classified as possible versions of an exemplary answer to her research question.

Through sampled reading, she gained sensitivity into the usefulness of working with the algorithm's thresholds, which indeed classified documents more closely related to known primary sources and more surprising. In the course of this research process, the researcher came to understand that some level of surprise in the algorithm (the 20-25% threshold of similarity) produced texts that were more instructive for guided reading. Iterative encounters with the algorithm and reading allowed the researcher to find documents that fit best with her questions.

Resampling, or Iteration of the Process

At the end of the second world war, the statistician George Box argued that "iteration" at the heart of statistical inquiry into science. Box set forward a general theory of critical reasoning, investigated by way of Francis Bacon and Ovid, in which a crucial component of good science was the ability to take measures to tailor a process such that new discoveries were possible. In Box's conceit, Pygmalion the scientist must not fall in love with his model. Resampling, in Box's view, was the psychological prophylactic that would prevent the researcher from entering into a "feedback loop" where the results were predetermined from the beginning of the experiment. Iterative resampling of the data at different levels allowed the scientist to judge the full breadth of the data. Each resampling would allow the researcher to constrain the experiment to eliminate error, but only in such minimal ways as to keep the experiment consistently open to unforeseen results.[34]

Across the disciplines, recent scholarship has insisted on the importance of systematic iteration where data-driven questions are at stake. Sociologists James Evans and Pablo Aceves more recently have emphasized the importance in social science of iterating between "theory confirmation" and "theory discovery."[35] Literary scholar Richard Jean So has also persuasively argued for the importance of learning about iteration in data-driven processes in the digital humanities.[36] Documented iteration has become increasingly crucial to the research process as humanities and social science scholars engage algorithms and quantitative thinking.

Critical search in itself attunes the scholar's sensitivity to the bias and perspectival nature of particular algorithms. In many cases, however, one pass through the algorithms is not enough. Keyword search, topic models, and divergence measures may all be used to narrow a corpus down to a smaller body of texts, for example identifying a particular decade of interest. In order to precisely "tune" the algorithms to the researcher's question, successive rounds of the critical search process may be necessary.

Iterative seeding and winnowing provides safety barrier against naïvely embracing the results of computational algorithm. At present, it is unclear how dependable most of our best tools for modeling text are, and where careful limits need to

[34] George E. P. Box, "Science and Statistics," *Journal of the American Statistical Association* 71, no. 356 (1976): 791-799.

[35] James A. Evans and Pedro Aceves, "Machine Translation: Mining Text for Social Theory," *Annual Review of Sociology* 42, no. 1 (2016): 21-50, 29.

[36] Richard Jean So, " 'All Models are Wrong,' " *PMLA* 132.3 (2017), 668-673.

be provided. For instance, computer scientists who deal with topic models have themselves called for more studies of whether, why, and how the topic model aligns with insights gained in traditional approaches. Eric Baumer and his colleagues have warned that there is "little reason to expect that the word distributions in topic models would align in any meaningful way with human interpretations."[37] Iterative winnowing and reading offer insurance against embracing foolhardy conclusions from digital processes. A truly critical search requires human supervision wherever the fit between algorithms and humanistic questions is unclear.

Seeding, winnowing and guided reading together may take the form of iterative encounters with the results of an algorithmic search. The researcher may begin with one query, sample the results, and use the best samples to "re-seed" the search with more specific texts. For instance, in the case of research about property, a similarity measure was used to rank all Hansard debates as more or less similar to seed debates that were known texts about property in England, Scotland, and Ireland. The researcher sampled the results that were classified by the algorithm at different thresholds of similarity—the 1%, 5%, 10%, etc. most similar to the seed texts. From those results, the scholar collected by hand a set of exemplary texts for later analysis. This new collection of texts was both material for guided reading, and was used to "re-seed" the search process anew.

The process of continuously "checking" the work of the computer allows the expert to judge better whether and how the resulting subcorpus fits the scholarly questions at hand. Sampling the results in a structured, regular process allows the scholar to assess the results of a search confidently. Repeating the process of unsupervised matching and human sampling creates a virtuous cycle whereby the scholar gradually approaches a subcorpus ideally suited to her research question. Thus a process of critical search for digital history must fix iterative modeling, reading, and analysis of the data into the *habitude* of the modern scholar.

Results in Brief

In the case study of property in Britain, secondary sources supply a dozen possible candidates for the moment when the discourse of property changed in Britain:

[37] Eric P. S. Baumer et al., "Comparing Grounded Theory and Topic Modeling: Extreme Divergence or Unlikely Convergence?," *Journal of the Association for Information Science and Technology* 68, no. 6 (June 1, 2017): 1397-1410.

1815, with the publication of major utilitarian pamphlets on property; 1837, with the Anti-Corn Law League; 1849, with the Encumbered Estates Act; the 1850s, with the aftermath of famine in Ireland; 1870 with the Landlord and Tenant Act for Ireland; 1881 and 1886 with the Irish Land Acts, and so on. Each history of property tends to assert that its unique clique of writers or its particular regional tradition was the most important for Britain, and there is little way for a scholar to choose between them except according to individual instinct or habit.

Digital tools promise to free the scholar from idiosyncracy of interpretation, by arming her with specific information about how these events were situated within a shifting lexicon of property. Yet, as we have seen, naïve use of the digital tools produces multiple conclusions about when and how the lexicon of property changed: perhaps because culture is multiple rather than unitary, and different parts of the lexicon were, in fact, evolving at every point in history.

Through critical search, the scholar gains some insight into which parts of the property question were evolving at which time, and thus gains confidence and insight about explaining the apparent explosion of a lexicon of property around 1880. When hand-picked titles were selected during the process of "Guided Reading," the resulting list of titles amplified a sense of a discontinuity around 1880 and allowed an even closer definition of what the relevant turning points were. The vast majority of these hand-selected titles, where parliamentary debates seemed to directly address questions of eviction, rent, and ownership, came from the period after 1881, after the Irish Land League and its "No Rent Manifesto" articulated a program of nationally-coordinated social action to force down property prices in Great Britain. There were surges of selected debates in particular years during the 1880s: in 1881 as the Land League organized; a trough in 1884, then peaking for the entire century in 1886-7 (the beginning of the "Plan of Campaign," an Irish program of joint social and parliamentary activism for land reform and the repeal of acts criminalizing public protest and the freedom of the press), dying down again in 1890 and rising almost linearly from 1890 until 1904. Examination of documents after iteration through various searches allows the researcher to assert a particular periodization with greater precision and clarity.

In the case study of a search for property in parliament, critical search produced in a very different set of material for guided reading than did either simple keyword search or topic modeling do when unaccompanied by an iterative process of seeding, winnowing, and guided reading. Naïve keyword searching tended to return, for further reading, an ungainly list of pieces of legislation related to rent, property, and agriculture, with the Irish and English legislation profiled, which is useful for establishing a chronology of the most intense debates in which rent and eviction were named, but less useful for documenting the variety of debates

in which issues of rent, eviction, and property transpired. The results of naive processes reproduced a geographical bias towards England, Scotland, and Ireland, a bias inherent in the archive itself.

Unlike naïve search, the process of critical search guided the reader to a much richer caricature of the property question. The results of this process contained debates that show a marked international footprint. A debate about property that took place around the entire geography of British empire, encompassing a debate about the Egyptian sale of domain lands, and many details about police actions in Ireland at the time of the land reform, about tenancy in Bengal, Zululand farm allotments, Zanzibar land disputes, access to mountains in Scotland, the cadastral survey of Bihar, and the need for a land title registry in Ceylon, the infamous "hut tax" in central and British East Africa, the cultivation of wastelands in India, the settlement of colonists on new lands opened up after the Boer War in South Africa, Welsh colonists in Patagonia, irrigation in India, evicted ryots in Madras, the sale of public lands in the Straits settlement of Singapore, and street improvements in Kingston, Jamaica. Critical search allowed the scholar to navigate from the vast sea of agricultural and taxation documents about property to those texts that were "surprising" enough according to the algorithm to involve the plurality of ways that property was handled around emprie. This new subcorpus of texts—an "imperial property" subcorpus—can then be analysed with relative confidence of its exemplarity.

Critical search also enhances the reader's confidence in the periodization she believes to typify a conversation. The "Winnowing" of different methods allows the scholar to carefully compare the bias that different algorithms bring to questions of period. As figures 3-4 show, different divergence measures suggest different chronologies, so divergence cannot be trusted to supply a verdict about when the property question emerged and how. In this case, counting keywords underscored the sense of a historical discontinuity around 1880, which was dramatized in a keyword search of debate titles; the keyword count is relatively transparent, as a marker of a new lexicon of property, and so in the case of periodization, keyword count is preferred.

As hoped, the critical search process indeed returns parameters for an overview of social experience. It gives the advantage of a wide, contextual background to whatever close reading results at the end of the process. To define corpus, subcorpus, and research question precisely enough that scholars may be confident in the results of any models based on them is to raise the bar of knowledge.

Critical Search and Scholarly Transparency

Perhaps the most profound question raised by the use of digital tools in history is what it means to be fully transparent about our interpretive choices. In the world of social and cultural history, transparency about the scholar's bias typically took the form of a trail of footnotes to cultural anthropology or feminist theory wherein the scholar laid bare her intellectual influences, and perhaps announced an agenda for recovering the silenced voices of the past. Digital scholars too may come with announce such perspectives. But they also have the opportunity to explain how that orientation guided their maneuvers through the digital archives, caused the selection of a particular algorithms or a search for a particular lexicon, with the potential results of correcting for the biases of the past with an enhanced sensitivity that is not entirely their own.

Critical search means adopting algorithms to the research agendas we already have—feminist, subaltern, environmental, diplomatic, and so on—and searching out those tools and parameters that will enhance our prosthetic sensitivity to the multiple dimensions of the archive. Documenting the choice of seed, algorithm, cut-offs, and iteration can go a long way towards a disciplinary practice of transparency about how we understand the canon, how we develop a sensitivity to new research agendas, and how we as a field pursue the refinement of our understanding of the past.

By calling for the documentation of choices around different algorithms and their results, critical search can form the basis for a rigorous, statistically diverse overview of subject matter and time periods, making visible and transparent choices about research such as the use of secondary sources and canonical texts. In this way, digital research can build upon the findings of earlier generations, generalizing upon them or problematizing them at enormous scale.

Critical search promises transparency in these findings, making good on the commitment of earlier generations of scholars to replicability in humanistic research and even radically extending that commitment to the every-day choices made by scholars. Traditionally, the scholar plucks events, characters, and research questions out of the archive by a combination of individual proclivity, expert guidance, happy accident, and close reading; a good research project is one where a wide enough variety of sources materialize to make thick reading possible. As this article demonstrates, the virtues of iterative rigor, broad contextual reading, and curious questioning as the marks of inspiring scholarship, and these will remain important qualifiers of individual talent in a digital age.

In explaining how each visualization is the result of particular choices in the

accumulation of starting point, keywords, secondary sources, algorithms, and their deployment, critical search will tend to move the reading and interpretation of data-driven visualizations away from a naïve reading, where visualizations appear to propose the view from nowhere, performing what Donna Harraway dubbed "the god trick."[38] Instead, some visualizations may be used to compare the bias of different measurements (Figure 4). Others will to explore different dimensions of a corpus rendered visible by algorithms, but visualizations juxtaposed will illuminate how the same question can be asked different ways (Tables 1-3, 4).

In calling for transparent documentation of the choices that go into research, however, critical search thus does not propose to eliminate the scholar's personal biases or to render all historical research transcendentally objective. The scholar, after all, chooses the seed texts, algorithms, and their cut-offs; and it is only the scholar who chooses and reads the new texts supplied by this process. Subjectivity and opportunities of individual insight remain at every level.

[38] Donna Haraway, "Situated Knowledges: The Science Question in Feminism and the Privilege of Partial Perspective," *Feminist Studies* 14, no. 3 (1988): 581.