Gov 1005: Data

David Kane

Description

Data matters. How much money is donated to political campaigns? How do polls help us forecast election results? We need data to answer these questions.

This course, an introduction to data science, will teach you how to work with data, how to gather information from a variety of sources and in various formats, how to import that information into a project, how to tidy and transform the variables and observations, how to visualize the data, and how to communicate your findings in a sophisticated fashion. Each student will complete a final project, the first entry in their professional portfolio. Our main focus is data associated with political science, but we will also use examples from education, economics, public health, sociology, sports, finance, climate and any other topic which students find interesting.

We use the R programming language, RStudio, Git, GitHub and DataCamp. Although we will learn how to write code, this is not a course in computer science. Although we will learn how to work with data, this is not a course in statistics. We focus on practice, not theory. We make stuff.

Prerequisites: None. You must have a laptop with R, RStudio and Git installed.

Logistics: Class meets in Tsai Auditorium in the basement of CGIS South from 1:30 to 2:45 on T/TH.

Figure 1: *Ulysses and the Sirens*, 1891, by John William Waterhouse. "This dramatic painting illustrates an episode from the journeys of the Greek hero Odysseus (in Latin, Ulysses) told in the poet Homer's *Odyssey* in which the infamous Sirens lured unwary sailors towards perilous rocks and their doom by singing in the most enchanting manner. Odysseus wished to hear the Siren's song and ordered his crew to lash him to a mast and block their ears in order to ensure their safe passage. Waterhouse has depicted each Siren with the body of a bird and the head of a beautiful woman, which came as a surprise to Victorian audiences, who were more used to seeing these mythic creatures portrayed as comely mermaid-like nymphs. He borrowed the motif from an ancient Greek vase that he studied in the British Museum." The next stop on Odysseus's journey was Thrinacia.

Course Metaphor

The central metaphor for this class is *Ulysses and the Sirens*. You are Ulysses. Thrinacia is the future you want. The Sirens are the many distractions of the modern world. *I am the rope.*

Course Staff

- Preceptor David Kane; dkane@fas.harvard.edu; CGIS South 310; 646-644-3626; office hours Thursday from 8:30 to 11:00, generally held in Fisher Commons. Please address me as "Preceptor," not "David," nor "Preceptor Kane," nor "Professor Kane," nor "Mr. Kane," nor, worst of all, "Dr. Kane."
- Teaching Fellow Albert Rivero; arivero@g.harvard.edu; office hours Monday 2:00 PM – 5:00 PM, generally held in Fisher Commons.

- Teaching Fellow Jacob Brown; jrbrown@g.harvard.edu; office hours Tuesday 3:00 PM – 6:00 PM, generally held in Fisher Commons.

Course Philosophy

No Lectures: The worst method for transmitting information from my head to yours is for me to lecture you. There are no lectures. We work on problems together during class.

R Everyday: Learning a new programming language is like learning a new human language: You should practice every day.

Cold Calling: I call on students during class. This keeps every student involved, makes for a more lively discussion and helps to prepare students for the real world, in which you can't hide in the back row.

Speakers: We will have a variety of visitors to class, people performing professional data analysis, both inside and outside of academia, often using exactly the same tools that we use. If there is someone you would like to meet, talk to me about it and we can invite them!

Class Activities: Awkwardness in the pursuit of learning is no vice. We will do a variety of class activities that will sometimes take you out of your comfort zones.

Community: You will meet and work with many more of your fellow Harvard students than you would in a normal course. You will probably learn the names of more students in this course than you will in all your other courses put together.

Millism: Political disputes are not the focus of this class but, when such topics arise, I will insist that we follow John Stuart Mills' advice: "He who knows only his own side of the case, knows little of that. His reasons may be good, and no one may have been able to refute them. But if he is equally unable to refute the reasons on the opposite side; if he does not so much as know what they are, he has no ground for preferring either opinion."

Organized by House: We use geography to create a community. During class, you will sit with students from your house, grouped with other houses near yours. If you live in Adams, for example, you will sit with other Adams students, and nearby the students from Lowell and Quincy. Within your house, you will work with different peers each class. Don't want to meet a score or more of Harvard students? Don't take this class.

Professionalism: We use professional tools in a professional fashion. Your workflow will be very similar to the workflow involved in paid employment. Your problem sets and final project will be public, the better to interest employers in your abilities.

Monologues: On occasion, I give brief monologues, designed to explain specific topics that have confused students in the past. I hope to never talk for more than 5 minutes straight.

No Cost: Every tool we use and reading I assign is available for free. You don't have to spend any money on this class. Some activities, like DataCamp and GitHub, have paid options which provide more services, but you never have to use them. Don't give anyone your credit card number.

Workload: The course should take about 10 to 15 hours a week, outside of class meetings, exams and the final project. This is an expected average across the class as a whole. *It is not a maximum.* Some students will end up spending much less time. Others will spend **much, much more**.

Course Policies

Use your Harvard e-mail: Please use your official Harvard e-mail address for all aspects of this class, especially things like signing up for services like DataCamp, GitHub, and so on. Doing so makes it much easier for us to figure out who is doing what. This may not be easy if you already connect with these services but, even in that case, you should be able to add your Harvard e-mail address to your account.

Piazza: All general questions — those not of a personal nature — should be posted to Piazza so that all students can benefit from both the question and the answer(s).

Plagiarism: If you plagiarize, you will fail the course. See the Harvard College Handbook for Students for details.

Working with Others: Students are free (and encouraged) to discuss problem sets and their final projects with one another. However, you must hand in your own unique code and written work in all cases. Any copy/paste of another's work is plagiarism. In other words, you can work with your friend, sitting side-by-side and going through the problem set question-by-question, but you must **each type your own code**. Your answers may be similar (obviously) but they must not be identical, or even identical'ish.

R: You must use R and RStudio for this class. You are responsible for installing both on your laptop.

Git and GitHub: Analyzing data without using source control is like writing an essay without using a word processor — possible but not professional. Starting in week 3, we will do all our work using Git/GitHub. If Git is not already installed on your computer, please install it.

DataCamp: We make extensive use of lessons from DataCamp. All DataCamp courses are graded pass/fail. Each week's course(s) are due by Monday at 10:05 AM (except in the cases of holidays or exams). Class on Tuesday will assume the completion of this work.

R for Data Science *(R4DS):* Reading assignments from R4DS in a given week cover material that we will use that week. Some students prefer to do such readings ahead of time, the better to prepare for class. Some students prefer to do the readings after those classes, the better to reinforce the material. Some students prefer to never do the readings. No matter what path you select, know that, when constructing/grading the problem sets, exams and final projects, we will assume that you understand the material in R4DS. If you are struggling in the class, the best advice we can offer is to read R4DS cover-to-cover.

Optional Activities: The syllabus includes background readings and DataCamp assignments which students may find interesting.

Depth and Breadth: The course aims to teach specific topics in depth. Months after the class is complete, you should be able to — without looking anything up — create a GitHub project, connect it to your R session, read in some raw data and create a nice looking plot. But we also aim for breadth. We want to expose you to dozens of techniques in R, even if we only show them to you once. We don't expect you to internalize these topics, in the same way you have done with the core material. We just want to show you these topics so that, if the need arises a year from now, you will recall, not the details, but that there does exist an R trick for your current problem.

Computer Problems: If you are having problems with your computer, follow these steps. First, post the problem on Piazza, with details and screenshots. With luck, a fellow student will be able to solve it. (And students who help their peers with technical issues are guaranteed full participation points for grading purposes.) Second, if I and/or the TFs can't solve it, we will direct you toward the IQSS IT Client Support Services, located in the basement of CGIS Knafel. They are excellent! E-mail them with the details of your problem, mentioning your enrollment in this class, at help@iq.harvard.edu. Although I and the teaching fellows want to be helpful, we are not experts in troubleshooting computer problems. Third, once your problem is solved, tell us all the solution by responding to your own post on Piazza.

Missing Class: You expect me to be present for lecture. I expect the same of you. There is nothing more embarrassing, for both us, than for me to call your name and have you not be there to answer. But, at the same time, conflicts arise. It is never a problem to miss class if, for example, you are out of town or have a health issue. Simply let me know if you will be absent. Failure to do so will decrease your participation points, as will missing too many classes, even with notification.

Late Days: An assignment is a day late if it is turned in anytime after it was due (even 5 minutes after) but within 24 hours. After that, it is two days late, and so on. You have 5 *Late Days* which may be used for any assignment, except for the two midterms and final project Demo Day. **You should save your late days.** If you use them early in the semester for no particularly good reason and then, later in the semester, have an actual emergency, we will not be sympathetic. We will not give you extra late days in such a situation. (That isn't fair to your classmates, and we are all about fairness.) We will just, mentally, move the Late Days you wasted so that they cover your actually emergency. You will now be penalized for being late earlier in the semester, when you did not have a good reason for tardiness.

Major Emergencies: We are not monsters. If you are hit with a major emergency — the sort of thing that necessitates the involvement of your Resident Dean — we will be sympathetic. We require a signed letter (not an e-mail) from your Resident Dean as documentation.

Role of Teaching Fellows: The TFs run all aspects of grading for the course, keeping track of late days, dealing with emergencies and so on. Go to them first with any problems. (Feel free to cc the Preceptor if you want to keep me in the loop but I am very respectful of TF authority on these matters.)

Role of Course Assistants: The CAs run Study Halls. They are not involved in grading assignments and can make no commitments about how the TFs will grade. Never ask a CA a question about grading. Instead, ask on Piazza and a TF will respond, or come to a TF privately with your question.

Waite Rule: We don't wear hats in the classroom. (Obviously, this prohibition does not apply to headgear of a religious nature.)

Computer Emergencies: We are very unsympathetic to computer emergencies. You should keep all your work on GitHub, so it won't matter if your computer explodes. If it does explode, you will lose only the work after your last push. You can then restart your work on a public computer (the basement of CGIS Knafel has machines with R/RStudio installed) or on your roommate's computer.

Github Classroom: We use Github Classroom to distribute problem sets and midterms. You will receive an e-mail with a link. Click on that link and a repo, with instructions, will be created. *Do this as soon as you receive the e-mail.* We don't want GitHub problems to arise the night before the assignment is due.

Speakers: We follow the *No Laptop Rule* during speaker presentations. Close your laptops. Put down your phones. If you want to take notes, use a pen. We do this because we respect the speakers, want to give them our full attention, and are thankful that they have taken the time to talk with us. I will still need to look at my phone, but only to ensure that the class ends exacly on time. Also, do not start gathering your belongings until class ends.

Announcements: You are responsible for any assignment/exam/deadline updates/changes which are either announced in class or promulgated via the course Canvas e-mail list. You are not responsible for every random post on Piazza.

Grading

Participation: 10 points. I expect you to participate, both in class and online. Helping your fellow students, especially on Piazza, is the best form of participation, as is volunteering for a class role. Be a good class citizen. Missing class (without notifying us) or missing too many classes will cost you points.

DataCamp Lessons: 5 points. Grades are pass/fail only. **These are free points!** Given the level of the questions and the hints provided, it is essentially impossible not to get full credit as long as you make an honest effort. Each day late (beyond the five allowed) results in -1 point from the 5 points total allocated for DataCamp assignments. *If you use up more than 5 points, further days late will make a negative contribution to your final grade.*

Problem Sets: 25 points. The first two problem sets counts for 1/2 point each. The remaining 6, which are much more time consuming, count for 4 points each. Problem sets are distributed on Wednesday and then due the following Wednesday at 10:05 AM. You are welcome to work on them with your friends but, first, you must personally type in every character in the work you submit and, second, you must list all the people you worked with. You may only use one Late Days for a given problem set since, after one days, we will distribute the answers. *If you do not submit your problem set within 24 hours of its due date, you receive a zero for that assignment. You will (also!) still be "charged" with the late day.*

Midterms: 15 points each. The two midterms are take-home. They are open-book and open-web. Because students have different schedules, you can complete the midterm any time within a four-day window starting after midterm distribution. Late midterms will not be accepted.

Final Project: 30 points. Students will present their projects publicly during Reading Period. They will then have the opportunity to incorporate feedback before submitting the final version. There are several milestones for the projects. You may use your Late Days for them, just as you might for DataCamp assignments or the

Problem Sets. But, as with DataCamp, these milestones must be met. Negative points will accrue until the milestone is completed.

The end of this syllabus provides details on schedule and grading standards.

Resources

The text for the class is *R for Data Science* (R4DS) by Garrett Grolemund and Hadley Wickham. The primary resources below are also useful, but are not required reading. The secondary resources may also be helpful. All are free.

Primary

Data Visualization: A practical introduction by Kieran Healy
Happy Git and GitHub for the useR by Jenny Bryan
The Unix Workbench by Sean Kross
R Markdown: The Definitive Guide by Yihui Xie, J. J. Allaire, Garrett Grolemund

Secondary

Pro Git by Scott Chacon and Ben Straub
Statistical Inference via Data Science: A moderndive into R and the tidyverse by Chester Ismay and Albert Y. Kim
Introduction to Data Science by Rafael A. Irizarry
Handling Strings with R by Gaston Sanchez
Text Mining with R: A Tidy Approach by Julia Silge and David Robinson

Conclusion

If you had tried to complete a data analysis project before taking this class, you would have done X well. Now that you have taken the class – now that you have learned how gather information in various formats, how to import that information into a project, how to tidy and transform the variables and observations, how to visualize and model the data for both analysis and prediction, and how to communicate your findings in a sophisticated fashion – you will do Y well. The success (or failure) of the class can be measured by comparing Y with X.

Schedule

Rhythm of the Class

The class follows a steady weekly rhythm:

- Sunday, 2:00 – 5:00 PM, Study Hall with Claire Fridkin, Dunster House.
- Monday 10:05 AM. DataCamp exercises due, except for extensions because of holidays or midterms.
- Monday 2:00 PM – 5:00 PM. Office Hours in Fisher with Albert.
- Monday, 7:00 PM – 10:00 PM. Study Hall with Dillon Smith, Smith Center,

- Tuesday 1:30 PM – 2:45 PM. Class. Main focus of class will be interactive R session using material from DataCamp exercises you have just completed. We will start with an example of something that we will work the rest of the week towards learning how to create.
- Tuesday 3:00 PM – 6:00 PM. Office Hours in Fisher with Jacob.
- Tuesday, 6:30 PM – 9:30 PM. Study Hall with Charlie Flood, Leverett House.
- Wednesday 10:05 AM. Problem set (distributed last week) due.
- Wednesday 4:00 PM. Take-home midterm distributed.

- Thursday 8:30 AM – 11:30 AM. Office Hours in Fisher with Preceptor.
- Thursday 1:30 PM – 2:45 PM. Class. In addition to continuing with the new R commands from Tuesday's class, we will finish with you creating an example similar to the one I showed you on Tuesday.
- Thursday evening. Problem set due next week will be distributed.

- Friday 10:05 PM. Final project milestones are due.
- Sunday 10:05 PM. Midterm exams, if distributed on Wednesday, are due.

Course Assistant Study Halls

Claire Fridkin, Dunster House, Sunday, 2:00 to 5:00 PM.
Dillon Smith, Smith Center, Monday, 7:00 to 10:00 PM.
Charlie Flood, Leverett House, Tuesday, 6:30 to 9:30 PM.

Week 1: January 28: Shopping Week

Install R, RStudio and Git on your machine. Start on the DataCamp assignments. They are due on Monday, February 4 at 10:05 AM.

Readings

R4DS: Chapters 1, 2 and 3

Optional

- How Obama's Team Used Big Data to Rally Voters by Sasha Issenberg

- An Extremely Detailed Map of the 2016 Presidential Election

- The Left Side of Steve Kerr's Brain by Marc Stein

Week 2: February 4.

Remember: DataCamp assignments are due Monday at 10:05 AM.

Readings

R4DS: Chapters 4, 6, 8, 26 and 27.

DataCamp

- Introduction to the Tidyverse
- Visualization Best Practices in R
- Working with the RStudio IDE (Part 1)

packages: tidyverse, ggplot2, fivethirtyeight, ggthemes
commands: install.packages, library, help, data, print, View/view, glimpse, summary, %>%, count, filter, arrange, select, desc, mutate, group_by, summarize, ggplot, aes, facet_wrap, facet_grid, mean, max, min, n
arguments: data, mapping, x, y, color, shape, group, geom_point, geom_line, geom_smooth, geom_bar, geom_freqpoly, geom_density, geom_histogram, scale_{x,y}_continuous, labs, annotate, show.legend, title, subtitle, caption, theme_fivethirtyeight
misc: %in%, c()

Assignments

Problem Set #1 due February 6 at 10:05 AM. We will do this problem set in class. It's purpose is to ensure that everyone has a working computer set up. Students will hand this assignment in via Canvas.

Optional

- "Rich State, Poor State, Red State, Blue State: What's the Matter with Connecticut?" by Gelman et al.
- *The Cognitive Style of Powerpoint* by Edward Tufte

Week 3: February 11.

Readings

R4DS: Chapters 5 and 7.

DataCamp

- Working with Data in the Tidyverse
- Data Manipulation in R with dplyr
- Reporting with R Markdown

R Packages, Commands and Arguments

packages: dplyr, janitor, ggridges, viridis
commands: names, tally, stat_summary, tidyverse_update, is.na, desc, contains, matches, num_range, rename, transmute, everything, parse_*, lead, lag, cum*, min_rank, median, sum, sd, quantile, first, nth, last, n_distinct, count, ungroup, geom_histogram, fill, cut_width, coord_cartesian, geom_boxplot, coord_flip, fct_rev, geom_violin, geom_count, geom_tile, geom_hex, geom_jitter, near, enframe, geom_density_ridges, scale_fill_viridis
arguments: starts_with, ends_with, contains, weight
misc: int, dbl, chr, dttm, lgl, fctr, date, janitor::clean_names

Assignments

Problem Set #2 due February 13 at 10:05 AM. Students will hand in this assignment via Canvas. Please work with other students. You are all in this together!

Optional

- *Visual and Statistical Thinking: Displays of Evidence for Making Decisions* by Edward Tufte

Week 4: February 18.

Monday, February 18 is a holiday so DataCamp is due on Tuesday at 10:05 AM.

Readings

The Unix Workbench, chapters 1 – 6.
GitHub Classroom Guide for Students

DataCamp

- Introduction to Shell for Data Science
- Introduction to Git for Data Science
- Working with the RStudio IDE (Part 2) — Only do Chapter 2 Version Control.

Assignments

Problem Set #3 due February 20 at 10:05 AM. Students will hand in this assignment via Canvas.

Speaker

February 19: Natalia Urtubey, Executive Director, Imagine Boston 2030 and Kayla Patel, Data and Performance Analyst at City of Boston

R Packages, Commands and Arguments

packages: dplyr, broom, gganimate, gifski, gt
commands: row_number, dense_rank, percent_rank, cume_dist, ntile, glance, tidy, augment, transition_*, view_*, shadow_*, enter_*/exit_*, ease_aes, tab_header, tab_source_note, cols_label, fmt_percent, fmt_missing, as_raw_html
arguments: frame_time, binwidth, xlim, ylim, na.rm. alpha
misc: ls, ., .., ~, pwd, cp, mv

Optional

- Git Set Up: The Definitive Guide
- A Quick Introduction to Version Control with Git and GitHub

Week 5: February 25.

Readings

R4DS: Chapters 9, 10, 11, 12, 13 and 14.

DataCamp

- Joining Data in R with dplyr
- String Manipulation in R with stringr. Only chapters 1 and 2.

Assignments

Problem Set #4 due February 27 at 10:05 AM. This problem set will be distributed, collected and graded using GitHub Classroom.

Speaker

February 28: Dishant Rana, The Data Trust.

R Packages, Commands and Arguments

packages: dplyr, tibble, readr, readxl, haven, tidyr, stringr, fs
commands: tibble, as_tibble, print, View, .$, .[[“”]], read_csv, read_excel, write_csv, parse_{logical,integer,date,number,charac
col_*, problems, guess_encoding, {write,read}_rds, gather, spread, separate, unite, pull, left_join, bind_rows, anti_join, str_detect, file_delete
arguments: “ (back ticks), n, width, Inf, skip, comment, col_names, na, col_types, n_max, locale, levels, format, %b,%y,%Y,%*,

Optional

- Regular expressions
- String Manipulation in R with stringr

Week 6: March 4.

First midterm distributed March 6 and due Sunday March 10 at 10:05 PM. Focus will be everything in R4DS through chapter 14.

Readings

R4DS: Chapters 15 and 16.

DataCamp

You are only required to do the first chapter from four of these courses, thereby getting a tour of R's capabilities. Note that, because of limitations with Data Camp, the entire course for each of these is officially assigned, rather than just the first chapter. You only need to do the first chapter.

- Working with Web Data in R
- Interactive Maps with leaflet in R
- Analyzing US Census Data in R
- Analyzing Election and Polling Data in R
- Sentiment Analysis in R: The Tidy Way
- Modeling with Data in the Tidyverse
- Machine Learning in the Tidyverse
- Interactive Data Visualization with plotly in R

Speaker

March 5: Stephanie Zhang '09, Quantitative Analyst at Google

Assignments

Problem Set #5 due Wednesday March 6 at 10:05 AM.

R Packages, Commands and Arguments

packages: forcats, lubridate,
commands: factor, parse_factor, parse_date, levels, count, fct_reorder, fct_reorder2, fct_relevel, fct_infreq, fct_recode, fct_collapse, fct_lump, today, ymd, make_date, as_date, year, month, mday, wday, {round,floor,ceiling}_date, update, days, months, weeks, years,
arguments: levels, format

Optional

- A Guide to Working with US Census Data in R

Week 7: March 11.

Readings

R4DS: Chapters 17 and 18.

DataCamp

Choose your own adventure! Pick two of the DataCamp courses from Week 6 and complete them. Provide the TFs with your DataCamp course certificates to confirm by pasting them in the Google doc, a link to which we will provide. You may also choose a different DataCamp class, if you like, but you must confirm your choice with us ahead of time. Because of the midterm, these are due on Wednesday March 13 at 10:05 AM. If you choose the Tidyverse machine learning course, you should probably do the Tidyverse modeling course first.

R Packages, Commands and Arguments

There are not many new items in R4DS this week. So, we will review the commands from the first half of the course, with a special focus on the midterm. We may explore some new packages like `rayshader` and `reprex`.

Optional

- Chapter 44 Web Scraping from *Introduction to Data Science* by Rafael A. Irizarry
- The tsibble package is the best way to handle time series in R.

Week of March 18 is Spring Break.

Week 8: March 25.

Because of Spring Break, DataCamp assignments are not due until Wednesday March 27 at 10:05 AM.

Readings

- Causality, Chapter 2 of *Quantitive Social Science* by Kosuke Imai.

DataCamp

- Intro to SQL for Data Science
- Intro to Python for Data Science
- Intermediate Python for Data Science

Speakers

March 28: Stefanie Costa Leabo, Chief Data Officer and Matt Smith, Principal Data Scientist, City of Boston

Optional

- Data Organization in Spreadsheets by Karl W. Broman and Kara H. Woo

Week 9: April 1.

Readings

R4DS: Chapters 19, 20 and 21.

DataCamp

- Writing Functions in R

Assignments

Problem Set #6 due Wednesday April 3 at 10:05 AM.

packages: purrr
commands: function, if, else, all, any, identical, near, switch, cut, stop, stopifnot, return, typeof, length, as.*, is_*, is_*scalar, set_names, [], [[]], $, list, str, attributes, vector, seq_along, flatten* , while, map, map_*, split, safely, possibly, quietly, invoke_map, keep, discard, detect, {head,tail}while, reduce
arguments: x, na.rm, …, L, NA, NaN, Inf, -Inf, NA{integer,real,character}, type, length, .x, .f, .$, ~, .
tricks: Cmd/Ctrl-Shift-R, list(…), lazyeval, use [[]] in all loops

Optional

- Building DashBoards with flexdashboard

Week 10: April 8.

Readings

R4DS: Chapters 22 and 23.

DataCamp

Foundations of Functional Programming with purrr

Assignments

Problem Set #7 due Wednesday April 10 at 10:05 AM.

Second midterm distributed April 10 and due Sunday April 14 at 10:05 PM. This midterm will be cumulative.

Speaker

April 11: Mara Averick, RStudio

Optional

- The Quartz guide to bad data

Week 11: April 15.

During the last three weeks of class, our focus shifts to Shiny. DataCamp is due Tuesday at 10:05 AM because of the midterm.

Readings

R4DS: Chapters 24 and 25.

DataCamp

- Building Web Applications in R with Shiny
- Building Web Applications in R with Shiny: Case Studies Many students find it helpful to start with the first chapter of this DataCamp even if, in theory, you should complete the prior DataCamp first.

Optional

- Naming Things by Jenny Bryan
- Shiny Apps User Guide

Week 12: April 22.

Readings

Read these written Shiny tutorials.

Assignments

Problem Set #8 due Tuesday April 23 at 10:05 AM. (Note that this is one day earlier than normal because there are no DataCamps this week.)

Hackathon

There will be an *optional* hackathon on Tuesday April 23, starting at 3:00 PM and going as late as people want to stay. Location is the Bok Center Learning Lab at 50 Church Street, Suite 308. Jacob's regular office hours will occur here from 3:00 to 6:00. Charles's regular study halll will be here from 6:00 to 9:00. Class veteran Molly Leavens will be running the show.

The is the perfect time to work on your final project, a draft version of which is due on Friday.

Speaker

April 23: Zachary Wang, Manager for Resources Adoption and Impact, Harvard Initiative for Teaching and Learning

Optional

- A Compendium of Clean Graphs in R

Week 13: April 29.

No problem set or DataCamp.

Only Tuesday class. Students will present their final projects during either the last class session or in the course slot immediately thereafter.

Contacts

A variety of data science professionals have kindly volunteered to be available to talk with students, both about data science in general and about data availability for final projects:

- Mara Averick, mara@rstudio.com, RStudio
- Hunter Holmes, hunter.holmes@thedatatrust.com, The Data Trust
- Kayla Patel, kayla.patel@boston.gov, Data and Performance Analyst at City of Boston
- Dishant Rana, dishant.rana@thedatatrust.com, The Data Trust
- Matthew N.K. Smith, matthew.nk.smith@boston.gov, Principal Data Scientist at City of Boston
- Hugh Truslow, truslow@fas.harvard.edu, Head, Social Sciences and Visualization, Harvard University
- Natalia Urtubey, natalia.urtubey@boston.gov, Executive Director, Imagine Boston 2030
- Zachary Wang, zachary_wang@harvard.edu, Manager for Resources Adoption and Impact, Harvard Initiative for Teaching and Learning

Assignment Details

Participation

There are several ways to earn participation points in class.

Imperator: Each House will have a class Imperator, someone who helps to organize a House study group, coordinate activities with other Houses, set up working pairs in class, and so on. Adiya Abdilkhay and Shafi Rubbani (First Years), Andrea Lamas-Nino, Shivani Aggarwal, Tate Green, Celine Vendler and Simone Chu (Everywhere Else). Imperators make everyone feel welcome, first by learning everyone's name and, then, by introducing classmates to each other.

Magicum: The class will have several technical wizards, students who have volunteered to help their peers with computer problems either in person or on Piazza. The most difficult of these questions will involve Git/GitHub, so only volunteer for this role if you are comfortable with those tools. Seeam Noor, Hemanth Bharatha Chakravarthy, Benjamin Hoffner-Brodsky and Will Smiles.

Welcome Committee: We organize a Welcome Committee of four students for each speaker. See below for the duties associated with this job.

Piazza Participation: Answering your classmates questions on Piazza is the best way to earn participation points. Be a good class citizen! If you find a (meaningful!) typo in a problem set or exam, please post it to Piazza. The first student to do so earns many participation points.

Problem Sets

Solutions for the first three problem sets are submitted via Canvas. After that, you will use GitHub. In both cases, you will submit two files: ps_N.Rmd and ps_N.html, where N is replaced by the number of the problem set. You must use exactly these names.

- The two documents you are submitting are very different.
  - The Rmd file is a *technical* document, an accurate record of your work which allows you (and us!) to reproduce your html easily. It should be well-organized, nicely formatted and clean. Non-technical readers will not understand it, but that is OK.

  - The html file is a *presentation* document, designed for non-technical readers. No R code or weird warnings or obscure messages mar its pleasing appearance. It is a simple list of the answers to the questions. There is no need for you to write anything.
- **It must be possible for us to replicate your work.** That is, we will open your ps_N.Rmd file in an R project which already includes all the files (raw data, generally) associated with the assigment, files which we distribute to you. When we knit your .Rmd, we should produce your .html. If we can't, we will take off points. (The Course Assistants will be happy to test your work. Visit them during Study Hall!)

Question Types

There are only three types of questions on the problem sets: tables, graphics and Mad Libs. Outside of these, you do not write any prose.

A Mad Libs style question provides a sentence with an X which you must replace with the correct answer. For example, the problem set might state:

You copy/paste that sentence as your answer, but replace the X with *inline R code* that determines the correct replacement for X dynamically. Do not include the words in the parantheses. They are there for explanation. Do *not* simply copy/paste the correct answer. In your Rmd, you might write:

I realize that these backticks are not exactly correct, but you get the idea. When you knit your Rmd file, this will turn into:

This is (you hope!) the answer that we are looking for.

Obviously, `x` needs to be a tibble which you have already created and which has `state` as a variable name. Sometimes, so much code is needed to answer the Mad Lib that it is placed in its own code chunk, with the answer saved as an object.

But that object is still placed in the inline code:

Late Days

You may use your late days on the problem sets, with a maximum of 1 late day per problem set. When using GitHub, there is no longer a "submission" as with Canvas. Rather, we download the latest commit you've pushed as of 10:05 AM Wednesday and grade that. If you want to use a late day for a problem set, email Albert before 10:05 AM Wednesday. Otherwise, we will grade your latest commit as of the deadline.

Colleagues

Always list, at the very end, the names of any students with whom you worked on the problem set. If there were none, write None.

Grading Rubrics

- Make sure you follow all the instructions in the prompt.
- Ensure that your repo is clean (no unnecessary or duplicative files).
- At least 5 commits with sensible commit messages, i.e., not "stuff" or "update."
- Once we download your repo, can we replicate your work easily? When we knit your .Rmd file, does your code throw an error (for example, by referencing a file you have locally but which you didn't push to GitHub)? (It is OK if you use a library which we need to download, but your Rmd better include all the necessary `library()` commands.)
- List the colleagues you worked with, if any.
- Make your code readable. Formatting matters. Check the "Excellent"" column in this rubric for a detailed description of what we are looking for.
- Include comments in your code. Rough guideline: You should have as many lines of comments as you have lines of code.
- Make your comments meaningful. They should *not* be a simple description of what your code does.

- Code comments must be separated from code by one empty line.

- Format your code comments neatly. Cmd-Shift-/ is the easiest way to do that.

- Name your R code chunks.
- Follow the Tidyverse Style Guide.
- Spelling and punctuation matter.
- Use captions, titles, axis labels and so on to make it clear what your tables and graphics mean.
- Use your best judgment. For example, sometimes axis labels are unnecessary. Read *Data Visualization: A practical introduction* by Kieran Healy for guidance on making high quality graphics.

Final Project

Do you love soccer or wine or NYC politics? The final project provides you with an opportunity to study that topic in depth. Your goal is to gather data and present it in an engaging fashion. We are not necessarily investigating specific hypotheses or trying to fit a statistical model, although you can do those things if you want. Instead, imagine that your roommate also cares about soccer/wine/politics/whatever. You are building something that would interest her, something that will make her say, "That is cool! Let's spend 30 minutes poking around with your data." Projects without at least 10,000 data points are unlikely to be interesting enough.

Your final project will be, for most of you, the first item in your professional portfolio, something so impressive that you will be eager to show it to potential employers. You must show this work publicly, both on the web (viewable by all) and in person at our Demo Day. You will host your final project using ShinyApps, a free service provided by RStudio. Make use of free statistical consulting from the Harvard Statistics Department and IQSS.

Consider all the final projects from Fall 2018. Here are some highlights:

Maclaine Fields: Harvard Volleyball
Kemi Akenzua: Death Row Last Words
Cayanne Chachati: Syrian Civil War

Charlie Olmert: Harvard Mens Lacrosse
Richard Qui: Vaccines

Milestones

Final project milestones are always due at 10:05 PM of the designated date. You may use Late Days, except for Demo Day itself. All submissions are made via a Google spreadsheet, the url of which will be distributed on Piazza.

- Early March: Speaker with CA/TF/Preceptor about your Final Project. No need to prepare for this meeting. But it is important to start thinking about what you want to do.
- March 15: URL for (or short description of) your data. You may change your project completely, all the way until Demo Day. But you are still responsible for meeting these milestones, even if you know you are going to pivot. 1 point.
- March 29: Github repo with Rmd (and knitted html) which discusses pros and cons of at least two projects from last year. At least one project should be one which did extensive data gathering/cleaning. You should not select the same projects for commentary as your friends have. 1 point.
- April 5: Rough Github repo, with all necessary data, and reproducible Rmd document. The Rmd must contain two items. First, a brief description of the data, where you got, what you have done with it so far and what you plan to do. Second, a beautiful ggplot2 graphic using some of the data. 1 point.
- April 19: Working Shiny App, just to demonstrate that you can get something up and running. 1 point.
- April 26: Must have a working rough draft of your Shiny App. 1 point.
- April 30: Demo Day! Details TBD. 10 points.
- May 10: Final Project due. Fill out Google spreadsheet correctly! 15 points.

Grading Rubrics

The milestones are pass/fail. Keep the following in mind for Demo Day and for the submission of your final project.

- All the rubrics for problem sets apply here as well.
- You should have a one and four sentence summary of your project memorized for Demo Day. The one sentence summary is for listeners who want the briefest possible pitch. The four sentence summary is for those who want more details. Both should be smooth and persuasive. People are busy. Why should they pay attention to you?
- Your repo must be public.
- We will look at (and grade) your code in conjunction with the Demo Day evaluation.
- Give your repo and Shiny App a descriptive name. "Vaccine-Explorer" or "syrian_civil_war" is good. "Gov_1005_Final_Project" or "project_test" is not.
- Some students work with messy data which requires a great deal of cleaning. Good stuff! Those students can create a very "vanilla" Shiny App and still receive full credit for the final project. Other students just use a ready-made data set from someplace like 538. Good stuff! But, in that case, they need to do something special with the analysis and/or display.
- Apps should all have an "Info" or "About" tab which includes, your name, contact information, GitHub repo and data source information. Include other background information as you see fit.
- Apps should "open" on an interesting tab, which will usually not be the "About" tab.

- Apps should have at least one tab in which the user can select something and see a change.
- Apps often have "story" tabs which, although they do not allow for user selections, do highlight specific aspects of the data which are interesting, and which users are unlikely to discover by themselves.
- Fill out the Final Project Spreadsheet accurately. Failure to do so will cost you two points.

Study Halls and Office Hours

Study Halls are run by Course Assistants (CAs), undergraduates who have taken the class in the past. They are one of the most popular parts of the course. Office Hours are run by the Teaching Fellows. Students who make the most use of these resources do better in class, and enjoy it more, than students who do not.

Introductions

At every Study Hall and Office Hours (SH/OH), the CA/TF will ensure that everyone knows everyone else's name. This class is a community and community begins with names. The typical timeline starts with the first person arriving and sitting at the table. They and CA/TF chat. (It is always nice for the student to take the initiative and introduce themselves to the CA/TF. Remembering all your names is hard!) A second person arrives and sits at the same table, followed by introductions. Persons 3 and 4 arrive. More introductions. Help your CA/TF by introducing yourself, even if you are 75% sure they remember your name. Be friendly!

At this point, the table is filled. Another person arrives. Instead of that person starting a new table, the CA/TF gives the new student their spot and moves their belonging to a new table. No student ever sits alone. The CA/TF hovers around the table until more students arrive and start filling out table #2. And so on. At each stage, students are responsible for, at a minimum, introducing themselves to the CA/TF and, even better, to the other students. Even better is when students who are already present shower newly arriving students with welcomes and introductions.

I realize that this is not how things work in (m)any other course(s). But awkwardness is the pursuit of class community is no vice.

Help Us Help You

Course Assistants, Teaching Fellows and Preceptor will, to the greatest extent possible, never just give you the answer. Something like "Use annotate()" might solve your immediate problem, but it does not set you up for success during the midterms — when we won't be around to serve as your personal oRacles — much less for the rest of your life.

Instead, we will take the time to show you how to find the answer yourself. This starts with how to search for help, especially when you are not sure what you are looking for. This is more art than science, but adding certain strings — like "R", "tidyverse", or "ggplot" — to the search often helps. Then, we provide advice about which locations are the highest quality (anything to do with RStudio or tidyverse), which locations are less good than they initially appear (sthda.com, r-statistics.co, rdocumentation.org), and which are difficult to use (Stack Overflow). We then explain the best way to make use of what you find.

We also point you directly to the best resources, especially to *R for Data Science* by Garrett Grolemund and Hadley Wickham and to *Data Visualization: A practical introduction* by Kieran Healy. We won't say: "Just use starts_width()." Instead, we will ask, "Have you read Section 5.4 of R4DS, involving the use of select()?" Yes, this will require an extra five minutes of your time. But every minute you spend reading high quality references is a minute well-spent.

We also help you learn how to seek help from others. There is a good way to ask for help on Piazza or Stack Overflow — generally involving the use of simple code which highlights your precise problem — and a bad way.

Only if none of this works will we just tell you the answer.

Social Events

Socializing with students outside of class is fun. Joining/inviting me is optional and has no influence on your grade in the class, i.e., **it earns you no participation points.** The three main options are:

Restaurant Lunches

My wife and I host students in groups of 4 for lunch throughout the semester, sometimes via the Harvard Class-Room-to-Table program and sometimes on our own dime. We organize this by House at the start and then open up spots to everyone later. Invitations to come. Dress is casual. Please be on time. The reservation will be under "Kane." Just go straight to the table.

House Lunches

I enjoy having lunch with you before class, either in Annenberg or in your House. I will leave this to the Imperators to organize.

Faculty Dinners

I enjoy attending faculty dinners, so feel free to invite me to yours. My only request is that you also invite the other students in the class who also live in your House. It is often fun to take over a table with a group of 4 or 5 or . . .

Welcome Committee Duties

There are 4 students on the Welcome Committee. Student 1 is in charge, contacts the speaker, and offers the use of her laptop. Student 2 does the introduction and stays after class to ensure that the speaker is escorted out nicely. Student 3 prepares 5 questions and asks the first question. Student 4 handles Piazza and asks the second question. (Assign these roles amongst yourselves. If you lose a committee member, re-assign the responsibilities.)

Student 1 Duties

- Goes to lunch with Students 2, 3, and 4 to get to know them and takes a picture with them. [**When: 1 week before talk**]
- Spends at least 30 minutes learning about the speaker and using Google to learn about their recent work or that which might be relevant to the talk. This can be over lunch. [**When: 1 week before talk**]
- Sends email to speaker [**When: To be sent around 3 days before the event**]

- Thank them again for agreeing to visit. Include a link to the syllabus and to our speaker information file. (Make sure everything you say is consistent with that speaker information. Check with me about any confusions.)
- Provide them with your e-mail and phone number so that they can e-mail/text you (whichever is easiest for them) in case something comes up before class. (I, obviously, turn off my phone during class.)
- Even though this information is in the speakers link (which they might not read), remind them that we expect them at roughly 2:15 and give them our address and classroom location. Do not ask them to arrive early!
- Do not ask them for anything! (They are already doing enough to visit us.) In particular, do not ask them for links and/or details about their work. It is the Welcome Committee's job to find that information.
- Quote (politely!) the advice from the Speakers document about the two types of talks that are most successful. Do not portray this as your own advice. That would be rude! Instead, frame this as a comment from your professor.

- Remind me via email to bring water for the speaker... [**When: A few hours before class**]
- Fifteen minutes before they are scheduled to arrive, the entire Welcome Committee should leave class and go wait for the speaker. Find the speaker. (Maybe they forgot where the classroom is located. Maybe they are waiting in the lobby upstairs or at the building entrance. Maybe they got confused and are in CGIS Knafel.) Introduce yourselves. Chat them up. Thank them for coming. Offer them water. (I have water bottles which you should get from me before the start of class.)
- In general, we do NOT want the speaker to start early. I have material that I need to cover in the class. So, it is best if you keep them outside the classroom until their scheduled start time, which is almost always 2:15. Of course, if they really want to see my genius lectures, then, by all means, invite them in. But (surprisingly!) they usually don't care. They will be happy to chat with you until the start time.
- If they are planning to use their own laptop, help them get connected to Harvard Wifi, presumably as a guest. If they are using your laptop with some of their slides installed, make sure your settings work for them. Work out the kinks before coming into the room. You might want to clean up your laptop so that nothing embarrassing comes up . . .
- At the start time (almost always 2:15), come in the room and signal me. Then, even if I don't see you, come down to the lecture stand. I will unplug, move to the side and keep on lecturing. You (Student 1) bring the speaker to the lectern and help them get set up. (Nothing wrong with you checking out the lecture set up ahead of time!) In general, it should be easy for them to plug in their laptop and have their screen appear. (Or for you to plug in your laptop.) Once it does, tell me. I will wrap up and turn the floor over to you.
- Return to seat [**When: At 2:15**]
- Personally send a thank you email to speaker cc'ing me alone. Hand written notes in addition to the email are encouraged but optional. [**When: The evening of the talk**]

Student 2 Duties

- Goes to lunch with Students 1, 3, and 4 to get to know them and takes a picture with them. [**When: 1 week before talk**]
- Spends at least 30 minutes learning about the speaker and using Google to learn about their recent work or that which might be relevant to the talk. This can be over lunch. [**When: 1 week before talk**]
- Remind me via email to bring water for the speaker. [**When: A few hours before class**]

- Fifteen minutes before they are scheduled to arrive, the entire Welcome Committee should leave class and go wait for the speaker. Find the speaker. (Maybe they forgot where the classroom is located. Maybe they are waiting in the lobby upstairs or at the building entrance. Maybe they got confused and are in CGIS Knafel.) Introduce yourselves. Chat them up. Thank them for coming. Offer them water. (I have water bottles which you should get from me before the start of class.)
- Provides a three sentence introduction for the speaker. Make sure you pronounce their name correctly! No matter how famous they are, three sentences is enough. Write out your introduction ahead of time. Memorize it. Do not read it from your phone like an Eli. Sit down near the front when you are done. [**When: After the technical set up is complete**]
- Personally send a thank you email to speaker cc'ing me alone. Hand written notes in addition to the email are encouraged but optional. [**When: The evening of the talk**]

Student 3 Duties

- Goes to lunch with Students 1, 2, and 4 to get to know them and takes a picture with them. [**When: 1 week before talk**]
- Spends at least 30 minutes learning about the speaker and using Google to learn about their recent work or that which might be relevant to the talk. This can be over lunch. [**When: 1 week before talk**]
- Creates questions that you might use during the lecture. E-mail them to me at least 24 hours before the visit. Ideally, the students will have lots of questions and you won't need them all. But if the other students are asleep, the Welcome Committee must engage with the visitor. The worst outcome is to have me ask the questions. [**When: At least 24 hours before the visit**]
- Remind me via email to bring water for the speaker.. [**When: A few hours before class**]
- Fifteen minutes before they are scheduled to arrive, the entire Welcome Committee should leave class and go wait for the speaker. Find the speaker. (Maybe they forgot where the classroom is located. Maybe they are waiting in the lobby upstairs or at the building entrance. Maybe they got confused and are in CGIS Knafel.) Introduce yourselves. Chat them up. Thank them for coming. Offer them water. (I have water bottles which you should get from me before the start of class.)
- Return to seat [**When: At 2:15**]
- Student 3 asks one question within the first five minutes of the talk. (And, yes, I will be timing this.) The exact question almost (!) does not matter. A technical question — "Sorry for interrupting, but, I was just wondering: What programming tools did you use in this project?" — is often good. [**When: Within first 5 minutes of talk**]
- Personally send a thank you email to speaker cc'ing me alone. Hand written notes in addition to the email are encouraged but optional. [**When: The evening of the talk**]

Student 4 Duties

- Goes to lunch with Students 1, 2, and 3 to get to know them and takes a picture with them. [**When: 1 week before talk**]
- Spends at least 30 minutes learning about the speaker and using Google to learn about their recent work or that which might be relevant to the talk. This can be over lunch. [**When: 1 week before talk**]
- Posts picture taken with other students on Piazza [**When: 1 week before talk**]
- Remind me via email to bring water for the speaker. [**When: A few hours before class**]

- Fifteen minutes before they are scheduled to arrive, the entire Welcome Committee should leave class and go wait for the speaker. Find the speaker. (Maybe they forgot where the classroom is located. Maybe they are waiting in the lobby upstairs or at the building entrance. Maybe they got confused and are in CGIS Knafel.) Introduce yourselves. Chat them up. Thank them for coming. Offer them water. (I have water bottles which you should get from me before the start of class.)
- Return to seat [**When: At 2:15**]
- Student 4 asks one question within the first ten minutes of the talk. We just want to break the ice, make the talk more interactive, encourage other students to chime in. [**When: Within first 10 minutes of talk**]
- Personally send a thank you email to speaker cc'ing me alone. Hand written notes in addition to the email are encouraged but optional. [**When: The evening of the talk**]
- Posts a message to Piazza offering to provide contact information for the speaker. You would never just post someone's e-mail in a semi-public location. Instead, you are offering to provide that info to any student who wants to contact you for it. If the speaker volunteered to you that they would be interested in providing data for final projects or meeting with students, you should obviously mention that. [**When: The evening of the talk**]

Technical Advice

Follow this advice.

R

- When using `download.file()`, make sure to set `mode = "wb"`. This ensures that the download will work on all platforms.

- When loading libraries at the start of an Rmd file, load `tidyverse` last. This decreases the chance of confusing name conflicts, like getting `count()` from the `plyr` library rather than from the `dplyr` library, which is almost certainly the version you want. `dplyr` is a part of the `tidyverse` so, by loading the `tidyverse` last, you ensure that `dplyr` will take precedence over any other library with identically named functions.

Rmd

- Use two blank spaces at the end of a line to ensure a new line.

- You can add tabs in your RMarkdown document by adding `{.tabset}` in your header. All sub-headers will then appear in a tab instead of alone.
- An empty code chunk with — r ref.label=knitr::all_labels(), echo=TRUE, eval=FALSE — in the chunk header will magically print out all the code (to the pdf/html output) from the Rmd file in which it lives. I use this for the problem set and exam solutions.

- `code_download: true` added below `html_document:` in the YAML header will produce a Code button for downloading the underlying Rmd file.

Git

- If you have git problems, your first stop is *Happy Git and GitHub for the useR* by Jenny Bryan.

- Never check in your .Rproj file.
- Always check in your .gitignore file. I always add *.Rproj to my .gitignore file so that Git doesn't keep bothering me about the .Rproj file.
- Read your git error messages. They often tell you what to do!
- You often need to "pull" before you can "push" your code. Pulling and then pushing is good default workflow.
- If you commit something by mistake, you can, before you push it, undo the commit by typing, from the command line: `git reset HEAD~`. Background here and here.

Github

- Keep your collection of repos clean. You can delete repos that you are no longer using or want to keep. For a specific repo, look under "Settings" for instructions.
- These steps will often solve a weird GitHub problem, especially in the case of a problem set or midterm:
- Clone the repository from GitHub.com into a new local directory (same way we always do it, but just save it somewhere other than where you are currently working)
- You now have whatever the latest version is of your project on GitHub. Copy over any local changes since your last successful push — from the old local directory you were working in — to the new local directory you just created.
- Add, commit, push. Use this new local directory from now on.

RStudio

- Under Tools -> Global Options -> General, set the "Save workspace to .RData on exit:" to "Never".
- Under Tools -> Global Options -> Code -> Saving, set the "Default text encoding:" to "UTF-8". This is especially important for Windows users from non-English locales.
- Under Tools -> Global Options -> Code -> Saving, check the box for Ensure Source Files end with a Newline.
- Relow comment lines with Shift+CMD+/
- Reformat code with Shift+CMD+A

DataCamp

- If you have trouble finding a course directly from DataCamp, go to the syllabus and click on the link that we provide for each course. Indeed, this is often the easiest way of starting work.
- Clicking "Start Course for Free" may not work, but "Continue Chapter" almost always will.
- If DataCamp is behaving strangely, then restart your browser. This solves most problems. If that does not work, restart your computer. If that does not work, use a different browser. Chrome seems to work best.

- It is fine to use several "Take Hint" and a few "Show Answer" options each chapter. We never want you to get stuck.
- If weird things start happening — especially a failure of DataCamp to credit the right solution — try restarting your browser. You won't lose any work. Sometimes, DataCamp just needs to reset itself.
- If, despite choosing "Show Answer", DataCamp refuses to give you credit, don't worry! Just skip that question and do everything else. DataCamp will still think you have not completed the course — because of that one question — but that is OK. Just e-mail Albert when the assignment is due and tell him about the difficulty. He will give you full credit and not charge you a Late Day.

Acknowledgements

This course is inspired by STAT 545, created by the legendary Jenny Bryan. Some of the slides and exercises come from Data Science in a Box, by Mine Çetinkaya-Rundel. Some of the in-class exercises are from *Teaching Statistics: A Bag of Tricks* by Andrew Gelman and Deborah Nolan. Kudos to authors like Garrett Grolemund and Hadley Wickham (*R for Data Science*), Kieran Healy (*Data Visualization: A practical introduction*), Chester Ismay and Albert Y. Kim (*Statistical Inference via Data Science: A moderndive into R and the tidyverse*) and Sean Kross (*The Unix Workbench*) for making their books freely available. Thanks to Kosuke Imai for open sourcing several of the datasets from *Quantitative Social Science: An Introduction* and to Matt Blackwell and Xiang Zhou for sharing the data from their courses. Lecture slides were created via the R package **xaringan** by Yihui Xie. Many thanks to all the folks responsible for R, RStudio, Git, GitHub and DataCamp. This course would not be possible without their amazing contributions.