STAT 209: DATA COMPUTING AND VISUALIZATION

SUMMER 2021

Instructor. Colin Reimer Dawson (*they/them*)
Email. cdawson@oberlin.edu
Course Website. colindawson.net/stat209/
Class Slack Workspace. stat209s2021.slack.com
RStudio Server. rstudio.oberlin.edu
Times and Locations. T/Th 9:30-10:50 in AJLC 102

Office Hours.
M 2:30pm-4:00pm (Group Office Hour in King 203 – Math Library)
T 11:00am-12:00pm (drop in, King 204)
W 9:30am-10:30am (Zoom, by appointment)
Th 11:00am-12:00pm (Zoom, by appointment)

If your schedule prevents you from attending any of the posted hours, please let me know as soon as possible.

Course Overview

The "big picture" learning goal of this course is for you to **develop the knowledge and skills needed to find, organize, manipulate, summarize, and identify patterns in data** from different sources and of various kinds, shapes, and sizes.

For the most part the end goal of each major assignment will be one or more graphical representations of data in order to highlight features and patterns of interest. However, before you can translate data into concise graphical representations, you need to be able to get it into a format that lends itself to visualization. The process of taking "wild" real-world data and "taming" it, which can involve pulling data from the web, merging data from multiple sources, cleaning data by removing or restructuring information, and creating new variables out of existing ones, is referred to as "data wrangling".

Date: Last Revised June 1, 2021.

Since the end product is more satisfying than the intermediate stages, the course is structured more or less "backwards" in terms of the sequence of steps taken to go from wild data to an interpreted visualization, so that at each stage we will have a "pipeline" that ends with a report that provides useful insight.

We will begin the semester with **interpretation and report writing** about existing visualizations, followed by **visualization** of already "clean" data, then **formatting, organizing, and summarizing messier data** so it can more easily be visualized, then automating **data-retrieval**, and will close with a handful of specialized topics like handling **geographical data**, **text data**, and some basic **machine learning** techniques which are useful to simplify large or complex datasets to make them easier to understand. Along the way we will also develop some data science best practices and habits, such as **documentation**, **reproducibility** and **version control**.

Note that although exploratory visualization is the "use case" for data that threads through this course, the data-cleaning and processing skills you will learn in the latter 2/3 of the course are also valuable for other pursuits in data science, such as statistical modeling. The modeling process itself is not covered in this class, but is the focus of other courses (for example, STAT 205, 213 or 237, or ECON 255). As such, **one way to think about this course is as a "practical complement"** to the more theoretical content of those other classes. Another is as an **introduction to data science**.

At every stage in the process from data-retrieval to report-writing, we will employ **professional-grade computing tools** in the form of the **R statistical language** and the **RStudio computing environment**.

Learning Goals. After completing this course, students should

1. Be able to **recognize what statistical elements** (quantities, categories, descriptive summary values) are captured by various graphical elements (such as color, size and position) in data visualizations encountered in the media or scientific publications

2. Be able to **create concise, informative, and accurate data visualizations** by translating statistical elements to graphical elements, with the aid of standard statistical software

3. Be capable of **preparing "wild" data** for visualization and analysis by merging, cleaning, filtering, recoding, and manipulating datasets, with the aid of standard statistical software

4. Be able to produce **transparent, and reproducible reports** that integrate code, output, and verbal explanation and interpretation in a smooth and aesthetically pleasing way
5. Develop good **data science work habits**, such as **version control** and **documentation of code**

Who This Course is For. Although this course is numbered at the 200 level, it has **no prerequisites**, and is intended to be **accessible to anyone**. The main requirement is simply that you have a **willingness to write some code** (but you need not have prior experience doing so!).

Since the course is **project-centered**, ideally you should have some idea about what kinds of questions you'd like to use data visualization to help address. If you don't have any project ideas coming in, that's fine, but you should start thinking about some as soon as possible!

Materials

Textbook. The main textbook is *Modern Data Science with R*, by Baumer, Kaplan and Horton. The paperback edition is about $70; there is an eBook only version available for slightly less (or hardcover for more).

A useful supplementary text is *R for Data Science*, by Hadley Wickham and Garrett Girolemund, which can be read online for free at `https://r4ds.had.co.nz/`, or purchased in hardcopy. This text is mostly useful for supllementary information on implementation of your wrangling and visualization goals in R (the "how"). Unfortunately it is a little light on explaining the reasoning behind the goals (the "what" and "why"), which is why I haven't adopted it as the main text, as much as I'd prefer to have a free text.

Software. We will be using the R computer language throughout the course, with the aid of the RStudio environment. Oberlin has an RStudio server that allows you to access RStudio from your browser (which you may have done if you took another STAT course). You can access it at `http://rstudio.oberlin.edu`. If you do not already, you will have an account set up with your ObieID (the short form of your email address, not including the domain: mine is `cdawson` for example) as your userID.

It is also possible to download and install your own local copies of R and RStudio, though this will require you to manage some things on your own that I am handling

on the server side. It's probably a good idea to do this at some point so you can more easily do everything after you leave Oberlin, but you will still need to use the server to turn in work and access feedback.

To install R and RStudio on your own computer (not recommended yet unless you already have a computing background), go first to `cran.r-project.org` to download R, then to `www.rstudio.com` to download the free Desktop version of RStudio.

Miscellany

Communication With Me and Each Other. I have set up a Slack group for communication related to the course. If you are taking the course you will get an invite at your oberlin.edu email address. **Please direct all course-related communication there, rather than email!** I have a much easier time keeping correspondence organized that way and you will get responses much more promptly.

I recommend installing the Desktop and/or Mobile app for Slack, rather than using it within your browser, but it's up to you. I will usually post classwide communications to the `#announcements` channel, so I recommend keeping subscribed to notifications, or at least checking it regularly.

If you have a question or comment that other students might be interested in, I encourage you to **post to one of the classwide channels** rather than PMing me. You might even get a faster response from one of your peers than from me!

I will try to respond to most questions posted on class days by the next class day. **If you need to ask me about something due the following morning, don't wait until the night before!** I have family and parenting responsibilities in the evenings and on weekends; and besides, it's just poor form.

Accommodations. If you have a disability of any sort that may require accommodations in order for you to do your best work in this class, please let me know as early as possible, and consult as well with the Office of Disability Services (ODS). By college policy, **all requests for accommodation require documentation from ODS.**

Honor Code. The Oberlin College Honor Code formalizes the idea that all work that you submit is your own and that you have given credit to the ideas and work of others when you incorporate them. You will be asked to write and sign the honor pledge on each graded assignment that you hand in. The honor pledge reads: "I have adhered to the Honor Code in this assignment."

What it means to adhere to the Honor Code depends on context. I describe what it means for each type of assignment below.

More information about the honor code can be found on the web at the Dean of Students site:
`https://www.oberlin.edu/dean-of-students/student-conduct/academic-integrity/students`

Assignments and Activities

Flipped Class. We will for the most part be employing a "flipped class" structure, with readings and/or (mostly short) lecture videos assigned to be completed before each class, the first part of class time dedicated to Q&A, and the rest dedicated to working in small groups on labs and projects. The readings and videos listed for each class should be completed/viewed *before* that class, as the lab will be based on that content. You will, therefore, need to **bring a computer to class**.

Labs. Most of our class time will be spent working hands on on guided explorations ("labs"). You'll work on these in pairs or threes (rotating partners every few weeks) with the goal that you can get at least close to finishing them during class, though if you need to finish outside of class you will have until midnight after either that class or the following one.

These involve a combination of coding exercises and written interpretations. Most coding exercises will have my solutions available (but behind a spoiler) while you are working on them, to keep you from getting stuck and so you can compare your solutions with mine as you go. In most cases, the last exercise will require integrating content from the rest of the lab and will not have a solution provided in advance.

There will be about 18 labs throughout the semester. The labs are intended as instructional assignments and not assessments, and accordingly most of the grade for each lab is completion-based, but the last problem will usually be graded for quality of the solution. The 15 highest lab grades will count toward the final grade.

Honor Code. You will work on labs collaboratively, however

1. You must write your own solutions and code
2. You must indicate in your submission what other students you worked with
3. You should not submit anything that you do not understand (this last criterion is admittedly fairly subjective; but the idea is that you should interrogate the

structure of what you are typing until you know why it's doing what it's doing, and not just imitate an example by shallow pattern-matching)

Online "Homework Quizzes". There will be five "takehome" quizzes on the basic concepts of the course: one on R basics, one on data visualization basics, and three on data-wrangling. These are intended to make sure that you have the most basic concepts mastered, and will consist of multiple choice and short answer responses submitted via a Google Form.

There will be one optional 'makeup' quiz covering content from the previous quizzes. This can be used to replace earlier grades on a "concept-by-concept" basis.

Honor Code. The quizzes must be done individually, but are 'open book and open notes'

1. You may not discuss your responses with anyone else
2. You may refer to course texts and notes
3. You may *not* use statistical software to answer the quiz questions. (Some questions will ask you to predict the output of a piece of code, for example; you need to actually predict it, by looking at it, not by trying to run it)

Projects. The main focus of the class is a progressive sequence of three visualization projects, each expanding on the content of the previous one. The first one will be done in randomly assigned small groups; the second in small groups of your choosing; the last one will be done individually.

These are fairly open-ended, to allow you to explore questions of personal interest.

The end product of each project is an "illustrated writeup", consisting of one or more data visualizations presented in the form of a short blog post-style article discussing what they show (FiveThirtyEight or Vox posts are a good model for this).

In order, the projects require you to

1. **Project 1** Use a dataset that is fairly "clean" to start with, with a minimum of "wrangling" required. Produce informative visualizations that highlight interesting properties of the data.
2. **Project 2** Merge multiple datasets with some non-trivial wrangling to clean the data and combine information. Produce visualizations that highlight patterns or relationships that would not be apparent from any individual dataset by itself

3. **Project 3** Combine the wrangling and visualization techniques from the first two projects with one or more of the more advanced or specialized techniques covered in the last part of the semester. Possibilities include:
    - creating visualizations for the web that allow the user to interact with them, for example by filtering or zooming in on the data with a slider, or hovering a cursor over parts of the visualization for more information
    - efficiently retrieving only needed data from a remote database where the full dataset would be too large to fit in memory
    - applying machine learning algorithms such as clustering or dimensionality reduction as a means of "pre-processing" a dataset too large or complex to visualize directly
    - working with data that has a specialized structure, such as geographic or text data

Honor Code. The same requirements apply to projects as to labs, except that it is assumed that you have collaborated with your group members, and you will turn in one set of files (via GitHub). However, in addition, **individual contributions should be recorded** in the form of individual commits to the submitted repository (this may not make sense now, but it will when we talk about version control).

Peer Feedback. For the first two projects, before turning in the final version, your peers will have a chance to provide you with constructive feedback and suggestions for improvement. Each project group will read draft versions of two other groups' writeups and rate and comment on the technical, aesthetic, and communication dimensions of the article, using the same rubric that I will use to grade the final product. We will use a single-blind design for this: the graders will know the authors of the project they are reading (this is unavoidable, since everyone has a different topic), but the authors will not know who wrote the feedback.

The group that created the writeup will then have a chance to make revisions before turning in the finished version that I will grade.

Honor Code. You need to write your own feedback.

Exams. There are no exams. There is no class meeting during the posted exam time, however Project 3 is due (electronically) at the end of the posted exam period, which is 4pm on August 30th.

Grading

60% of the the final grade will be based on mastery of specific learning objectives as demonstrated on labs, quizzes and projects. See the accompanying "Grading System" handout for details on how this gets broken down.

30% will be based on good faith, timely completion of assigned work, irrespective of technical correctness

10% will be based on participation in providing peer feedback on draft projects

Major Dates

6/17	Project 1 Topic Chosen
6/22	Project 1 In-Class Brief Presentations
6/24	Project 1 Draft Writeup Due
6/29	Project 1 Peer Feedback Due
7/06	NO CLASS
7/08	Project 1 Final Writeup Due
7/15	Project 2 Topic Chosen
7/27	Project 2 In-Class Brief Presentations
7/29	Project 2 Draft Writeup Due
8/03	Project 2 Peer Feedback Due
8/12	Project 2 Final Writeup Due
8/30	Project 3 Due

Approximate Schedule of Topics

See the "Schedule" tab on the course website