Modern Data Structures

QMSS G5072 - Columbia University

Fall 2017
Lecture: Wednesdays 6.10 - 8pm (but see weekly schedule)
Location: 207 Union Theological Seminary

Instructor: Thomas Brambor
thomas.brambor.com
tb2729@columbia.edu
IAB 509E Thurs 11-12pm

TA1: Shriya Balaji Palsamudram
sbp2148@columbia.edu
IAB 270 Mon 2-4pm

TA2: Sahil Manocha
sahil.manocha@columbia.edu
IAB 270 Fri 2-4pm

Quick Links

Course Description

This course is intended to provide a detailed tour on how to access, clean, "munge" and organize data, both big and small. (It should also give students a flavor of what would be expected of them in a typical data science interview.) Each week will have simple, moderate and complex examples in class, with code to follow. Students will then practice additional exercises at home. The end point of each project would be to get the data organized and cleaned enough so that it is in a data-frame, ready for subsequent analysis and graphing. Therefore, no analysis or visualization (beyond just basic tables and plots to make sure everything was correctly organized) will be taught; and this will free up substantial time for the "nitty-gritty" of all of this data wrangling.

Course Website

All lecture materials, exercises, and (links to) readings will be made available in the GitHub course repository.

This is a new course. The materials and topics indicated below are a provisional roadmap that will be adjusted to the needs of the students. I will let you know well ahead of time of any changes.

Communications

For all questions to the members of the teaching team, we will be using the discussion forum that is integrated into Columbia's Canvas. The forum will be used to exchange questions about lectures, assignments, software etc. Students are encouraged to help each other!

Students are asked to customize their Canvas notifications preferences to receive immediate (ASAP) notifications of messages and announcements through the third-party provider of choice (e.g. email, SMS/text). Students are also asked to log into the course regularly (more than twice a week) and check Announcements and the Canvas Inbox immediately upon logging in to stay on top of developments in the course as they occur.

Please send emails and messages to the instructor and teaching assistants through Canvas. Messages sent through the Canvas Inbox (Send a Message) feature will be answered within 24 hours during the week and within 48 hours on weekends. Please consider these response times when asking about assignments etc.

References and Resources

Books

There are no required books for the course. All required readings will be provided as PDFs or links. However, here are some books that you may find useful in addition to the lectures and course readings.

- Wickham, H., & Grolemund, G. (2017). R for Data Science: Import, Tidy, Transform, Visualize, and Model Data (1 edition). O'Reilly Media. – Great as introduction on how to use R. From the creator of many R packages that we use in the course, this will help with the usual tasks of data import and management, modeling, and some visualization. Book is available for free online.
- Wickham, H. (2014). Advanced R (1 edition). Boca Raton, FL: Chapman and Hall/CRC. Book is available for free online
- Boehmke, B. C. (2016). Data Wrangling with R (1st ed.). New York, NY: Springer.

Free Online Resources

R, RStudio, and R Markdown

- IDRE at UCLA has lots of tutorials, code examples, for R and other statistical packages.
- Try R. In-browser, interactive online tutorial. Particularly useful if you have not used R (much) before.
- Cheat sheets for data wrangling, data visualization, general use of R, R Studio, R Markdown etc.
- R Studio resources for R Markdown. Get started here with markdown.
- Awsome-R A curated list of great R packages and tools.

Git and GitHub

- Git
  - Official git command line and GUI clients, official documentation
- Clients
  - Desktop Client for Mac and Windows
  - Sourcetree - another free visual Git
- Tutorials
  - Setting up git
  - Try.github
  - Hello World - GitHub for the non-programming beginner.

  - Guides at GitHub
  - Git and Github guide from plot.ly Extensive screen-shot guided intro to Git, Github, Git in RStudio and GitHub pages.
  - Pro Git - a full book with lots of details
  - Datacamp - Introduction to Git for Data Science

Coding Help Sites

- http://stackoverflow.com/ Programming Q&A site. Excellent first stop if you have questions on coding. Searching for keywords, and restrict your queries by adding tags about the coding language or package in square brackets, e.g. `[R]`,`[ggplot]`, or `[shiny]`.
- http://stats.stackexchange.com/ A stackoverflow off-shoot with a bit more focus on conceptual questions in statistics.
- http://rseek.org/ Search engine for R-related stuff, including tutorials and code.

Requirements and Assessments

Requirements

This course will guide you through the data wrangling process using the software package `R` for most exercises. The program `R` itself can be downloaded for free at http://cran.r-project.org/.

Some familiarity with the software, in particular with regards to the base functions in R is assumed. Knowledge of specific packages and other software tools will be built throughout the course. If you have extensive experience with other similar programming tools, say Python or Matlab, you will be fine. However, if you are completely new to `R` and do not have compensatory experience in other coding languages, please consider in the QMSS course "Data Mining" instead.

You will need to have access to your own computer to install software and packages, do your assignments etc. I highly recommend bringing your laptop to class to follow along the coding tutorials and examples.

Assessments

Homework

Homework problems will be assigned on a weekly basis, and students are expected to work on them alone.

Exams

We will have an in-class final, which will require the students to generate code during class to perform common operations, just as they would find in a data science interview.

Grade Distribution

The distribution of the parts for your grade is as follows:

- Final Exam = 30%
- Homework Assignments = 60%
- Attendance and Participation = 10%

Policies

Attendance and Class Participation

Your attendance and participation are necessary at every meeting. This class will work best when students ask a lot of questions.

Academic Integrity

This course is based on the principles of academic integrity established by Columbia University and agreed to by each student. The same rules hold in this course. Academic dishonesty will not be tolerated. All submitted work must be your own work and properly cited.

The full guidelines on academic integrity as well as a review of how or what to cite, can be found here: http://gsas.columbia.edu/academic-integrity

Students found guilty of plagiarism or academic dishonesty will be subject to appropriate disciplinary action, which may include reduction of grade, a failure in the course, suspension or expulsion. This includes lab reports – if they are copied from another student, severe penalties may be applied. ** Note that plagarism is also possible when writing code, so be careful to write your own code.

Late Assignment Policy

Students will lose points for handing in late assignments, at the discretion of the instructor and teaching assistant.

Other

Turn off or silence your cell phones prior to the beginning of class. I reserve the right to answer all calls (your's, not mine) received during class time and let your friends know what you are learning that day.

Feel free to use laptops in class - in fact, I encourage it. Respecting your classmates and myself, please refrain from using Facebook, shopping sites or other random distractions during class.

Changes

There may be adjustments of readings, assignments, exams, and classrooms. Changes will be posted on Courseworks along with other announcements.

Slides

Lecture slides will be made available on the course website. However, I believe that learning and understanding is better served when you need to aggregate and structure your notes yourself, so I suggest you do so as well.

Lecture Topics

Week 1: (Sep 6) Introduction

- **On your own:** Install R and R Studio on your own computer. Try out R Markdown (use the tutorial to get familiar).

Part 1 - Data Manipulation

Week 2: (Sep 13) Github

- **On your own:**
  - Sign up for a GitHub account.
  - Install GitHub Desktop (if you are confident in using command-line Git or have a different software preference, feel free to skip this step.)
  - Claim your private repository connected with this class.
- **Reading**:
  - Hello World - GitHub for the non-programming beginner.
  - An Intro to Git and GitHub for Beginners (Tutorial), by Meghan Nelson.
- **Advanced Topics (optional, on your own only)**:
  - *Combining Shiny & RMarkdown* (Overview here):

    - RMarkdown also allows interactive applications with Shiny. Follow the introduction on the RStudio website to create an interactive document. Shiny apps can be embedded in a document or called from an externally saved shiny application.
    - In Shiny applications themselves, you can allow users to generate a report (based on markdown).
  - RMarkdown can be used directly from the command line or from within R. You can render `.R` scripts into reports.
  - *Report Automation*: The creation of report (as well as uploading/emailing) can be automated completely.
  - *Git*:
    - The in-class introduction to Git was centered around GitHub. To learn a bit more, get comfortable with command line git usage.
    - Also, make sure you understand how branches work and how to work with a group of people.
    - Submit something to a public repository on Github using a pull request.

Homework 1: Using RMarkdown and Github. Also see the homework submission instructions.

Week 3: (Sep 20) Basics of the tidyverse

- **On your own:** Install `tidyverse` package.
- **Reading**:
  - Why R is Hard to Learn, by Robert A. Muenchen
  - Wickham, H., & Grolemund, G. (2017). R for Data Science. Chapters 9-12
- **Advanced Topics (optional, on your own only)**:
  - Interested in text as data? Take a look at Silge, J., & Robinson, D. (2017). Text Mining with R: A Tidy Approach. O'Reilly Media.

Homework 2: Data Wrangling with the Tidyverse. Also see the homework submission instructions.

Week 4: (Sep 27) Functions I: the basic logic and simple steps

- **Reading**:
  - Functions in *Advanced R* by Hadley Wickham
  - Some basics of code styling. Style Guide in *R packages*, by Hadley Wickham

- **Advanced Topics (optional, on your own only)**:
  - We discussed how scoping depended on R environments. Learn more about how these enviroments are called, how to create new environments, how you can look up their content, and how to define the search path of a scoping operation. See Wickham, *Advanced R*, chapter on "Environments"
  - We only discussed `for` loops in lecture. Check out two other types of loops - `while` and `repeat` loops - and how they can be useful for programming. Datacamp has a tutorial on "A Tutorial on Loops in R - Usage and Alternatives".
  - A great way to hide additional arguments for advanced users of a function is the `...` (read dot-dot-dot) argument. Try to get familiar with it and hide some options of a function you created.

Homework 3: `for` loops and functions. Also see the homework submission instructions.

Week 5: (Oct 4) Functions II: nested operations and complex sets of commands. The `purrr` package

- **On your own:** Install `purrr` package.
- **Reading**:
  - Wickham, H., & Grolemund, G. (2017). *R for Data Science*. Chapter 21 on "Iteration" - For the introduction of the `purrr()` package, I follow this material quite closely.

  - *Optional:* For a more interactive approach try Swirl – R Programming – Lesson 9 – Functions, by Johnny Chan.

- **Advanced Topics (optional, on your own only)**:
  - Section on *Functional Programming* in *Advanced R* by Hadley Wickham. Note, that the `purrr` package did not exist yet when the book was written, so it is not discussed.

Homework 4: Functions II. Also see the homework submission instructions.

Week 6: (Oct 11) Functions III: Writing your own R Package

- **On your own:** Please follow the instructions in the *Intro* of Hadley Wickham's book on *R Packages* to make sure you have the required software installed.

- **Reading**:
  - Writing An R Package From Scratch, by Hilary Parker
  - Instructions for Creating Your Own R Package, by Song Kim, Phil Martin and Nina McMurry
  - *Further reading, not required!:* Wickham, H. (2015). R Packages: Organize, Test, Document, and Share Your Code (1st edition). Sebastopol, CA: O'Reilly Media. Available online for free.

- **Advanced Topics (optional, on your own only)**:
  - *Advanced R* by Hadley Wickham: There are several issues for which we have very little time in lecture. These include performance of R code and how to optimize it. Similarly, we spend no time on C++ programming, but R has some well-developed packages to create high-performance functions in Rcpp.

Homework 5: Writing an R Package. Also see the homework submission instructions.

Week 7: (Oct 18) Functions IV: Working with strings

- **On your own:** Install `stringr` and `rebus` packages.
- **Reading**:
  - Handling and Processing Strings in R, by Gaston Sanchez
  - Strings in *R for Data Science*, by Hadley Wickham and Garrett Grolemund
- **Advanced Topics (optional, on your own only)**:
  - There are a lot of tools to work with text as data. To get started with text mining and visualization, I recommend the following readings:
    - Silge, J., & Robinson, D. (2017). Text Mining with R: A Tidy Approach. O'Reilly Media.
    - Grimmer, J., & Stewart, B. M. (2013). Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. Political Analysis, 21(3), 267-297. https://doi.org/10.1093/pan/mps028
  - For packages to work with text, I recommend the following:
    - Quanteda package vignette on text analysis
    - tm package vignette

Homework 6: Working with Strings. Also see the homework submission instructions.

Part 2 - Getting Data In

Week 8: (Oct 25) APIs

- **On your own:** Install the `httr` package.

- **Reading**:
  - Using Data.gov APIs in R, University of Virginia Library

  - Scraping via APIs, by Bradley Boehmke
- **Advanced Topics (optional, on your own only)**:
  - *Writing a R API client*: We have learned how to write an R package before. So how about writing an R API package if none is available yet. CRAN provides some Best practices for API packages by Hadley Wickham.
  - *Creating an API*: We can go even further. The `plumber` package allows you to turn your existing R code to a web API.

Homework 7: Calling an API using `httr`. Also see the homework submission instructions.

Week 9: (Nov 1) Handling JSON and XML

- **Reading**:
  - Using R to download and parse JSON: an example using data from an open data portal, by Zev Ross
  - Better handling of JSON data in R?, by Rolf Fredheim
  - Introduction to `tidyjson`, by Jeremy Stanley

Homework 8: Writing a simple API client. Also see the homework submission instructions.

Week 10: (Nov 8) Web Scraping from HTML

- **On your own:** Install the `rvest` package.

- **Reading**:
  - Using rvest to Scrape an HTML Table, by Cory Nissen
  - How To Screen-scrape, by Chris Bail

Homework 9: Web Scraping from Wikipedia. Also see the homework submission instructions.

Week 11: (Nov 15) SQL

- **Reading**:
  - There are lots of great SQL Tutorials to go further. Here are a few pointers:
    - Codeacademy's SQL Tutorial
    - SQLzoo's SQL Tutorial
    - W3School.com SQL reference (also interactive)
    - Datacamp's SQL Intro Course
  - Practice using SQLZOO
- **Advanced Topics (optional, on your own only)**:
  - We did not cover joins in SQL. But they are essential, especially given the usually *relational* nature of the data.
    - Learn about join types and how they can be used in SQL.
    - Datacamp has a nice introduction to joins as well: Introduction to Joins in PostgreSQL
  - Try your hand on connecting to a remote SQL database of your choice.

Part 3 - Other "Big Data" Considerations

Week 12: (Nov 29) Amazon Web Services and Parallelization

- **Reading**:
  - A comprehensive beginner's guide to start ML with Amazon Web Services (AWS) by Aarshay Jain
  - Analyzing Your Data on the AWS Cloud (with R), by Tal Galili
  - Five ways to handle Big Data in R, by Oliver Bracht
- **Online Tutorials (only recommended)**:

  - Launch a Linux Virtual Machine with Amazon EC2
  - Store and Retrieve a File with Amazon S3
  - Create and Connect to a MySQL Database with Amazon RDS
- **Advanced Topics (optional, on your own only)**:
  - Learn more about efficient coding in R. I recommend the chapters on "Efficient Coding" and "Efficient Optimization" from Gillespie, C., & Lovelace, R. (2017). Efficient R Programming: A Practical Guide to Smarter Programming (1 edition). Sebastopol, CA: O'Reilly Media. Free online
  - Additional coverage of Parallelization in R: McCallum, Q. E., & Weston, S. (2011). Parallel R: Data Analysis in the Distributed World (1 edition). Beijing: O'Reilly Media. PDF free here
  - An extensive and well-written tutorial on parallelization: Going beyond single-core R by Jonathan Dursi.

Final Project Proposal: Final Project Proposal due on Dec 1.

Week 13: (Dec 6) Big data access and computation

- **Readings on Algorithms**:
  - Basic Introduction into Algorithms and Data Structures, by Frauke Liers
  - *Introduction to Pseudocode* by Carnegie Mellon's Robotics Academy
- **Online Tutorials (only recommended)**:
  - For Spark:
    - RStudio Website with resources for Spark integration
    - Apache Spark for R
    - Datacamp course Introduction to Spark in R
    - RStudio Webinar Using Spark with Shiny and R Markdown
  - For BigRQuery:
  - For GDELT:
    - Brendan Knapp's brief analysis of the Syria conflict using GDELT here
    - RStudio Webinar Working with Big Data in R
- **Advanced Topics (optional, on your own only)**:
  - Running a cluster on AWS using Spark

Final Project due on Dec 15: Final Project Description.