Data Science Boot Camp

Arthur Small, Principal Scientist, Weldon Cooper Center for Public Service, University of Virgini

June 8-10, 2021

Contents

Course coverage **5**
0.1 Core texts and resources 5
Schedule 5

1 Computing setup **7**
1.1 R and related resources 7
1.2 Git and Github 8
1.3 Other resources 9

2 Introduction **11**
2.1 Reproducible workflows 12
2.2 Working collaboratively in a team 13

3 Materials from Basecamp **15**

4 Data Science with R **17**

5 Data pipelines **19**
5.1 Retrieving data 19
5.2 Data types 20
5.3 Data wrangling 20
5.4 Managing data 20

6 Git and Github **21**
6.1 Version control: what and why 22
6.2 How git works 22
6.3 Git in R Studio 22
6.4 Github 22
6.5 Best practices 22

7 R Markdown **23**

8 Shared resources **25**
8.1 Network File Server 25

9 Bibliographic resources **27**
9.1 Zotero 27
9.2 Bibtex 27

Course coverage

This mini-course is designed to …

Topics:

- Overview: what is data science; reproducible workflows & literate programming; working collaboratively in a team.

0.1 Core texts and resources

R for Data Science by Hadley Wickham and Garrett Grolemund. An excellent introduction, available for free online.

Schedule

June 8

Time	Topic	Readings
9:00–9:30	Welcoming remarks	
9:30–10:30	Data science on a team	
10:30–11:00	– *break* –	
11:00–12:00	R Markdown	
12:00–1:00	– *lunch break* –	
1:00–1:45		
1:45–2:30		
2:30–3:00	– *break* –	
3:00–4:00		

June 9

Time	Topic	Readings
9:00–9:30		
9:45–10:30		
10:30–11:00	– *break* –	
11:00–12:00		
12:00–1:00	– *lunch break* –	
1:00–1:45		
1:45–2:30		
2:30–3:00	– *break* –	
3:00–4:00		

June 10

Time	Topic	Readings
9:00–9:30		
9:45–10:30		
10:30–11:00	– *break* –	
11:00–12:00		
12:00–1:00	– *lunch break* –	
1:00–1:45		
1:45–2:30		
2:30–3:00	– *break* –	
3:00–4:00		

Credits: These course materials were generated using the **bookdown** package (Xie, 2020), which was built on top of R Markdown and **knitr** (Xie, 2015).

Chapter 1

Computing setup

For the Cooper Center Data Science Boot Camp, please install the software described below on your local machine.

1.1 R and related resources

We do much of our coding in R, a programming language especially well-suited to statistical computing. For more background, see: R for Data Science, Section 1.4, "Prerequisites".

1.1.1 Install R

- Download and install R. Follow directions for your machine's operating system.[1]

1.1.2 Install R Studio

R Studio is an integrated development environment (IDE) for R. It offers a variety of utilities to enhance the experience of coding and generating documents. Unless you already have another IDE you prefer, we suggest you use R Studio.

- Download and install R Studio, v. 1.4.1+.

[1] If you already have R installed on your machine, make sure you're version is later than v. 3.0.1. You can check your current version by running the command `R.Version()` from the R console. The most recent version is: R version 4.0.2 (2020-06-22).

R Markdown is installed automatically when you install R Studio.[2]

1.1.3 Install the Tidyverse packages for R

The utility of R can dramatically enhanced by the ability to install additional packages that extend its capabilties. The Tidyverse is a collection of packages that extend the capabilities of R for doing data science.

- Install the Tidyverse packages for R: From the Console tab in R Studio (or from R running in a Terminal window), enter:

```
install.packages("tidyverse")
```

- Alternatively, you may install these and other packages via the `Packages` tab in R Studio.

1.1.4 Install LaTeX

In order to generate PDF documents from your R Markdown sessions, you will need to install LaTeX. Several versions (or *distributions*) of LaTeX are available. If you do not already have LaTeX installed on your machine, and don't have any preference between options, then I recommend you install the TinyTeX distribution. This is easily done: simply enter the following two lines of code into your R console:

```
install.packages('tinytex')
tinytex::install_tinytex()
```

1.2 Git and Github

Git is software for version control. Github is a web service that provides remote storage and access to files via git. By using git and Github together, we greatly facilitate collaboration between multiple individuals working on the same code base, a.k.a. collaborative coding.

If you have not worked with git and Github before, this short YouTube video provides an orientation to git and Github: Git and GitHub for an Organized Project (STAT 545 Episode 2-A) from the University of British Columbia. See also: Happy Git and GitHub for the useR.

[2] If you do not install R Studio on your machine, you will need to install and load the `rmarkdown` package for R manually.

1.2.1 Install git and link tit to R Studio

Follow these instructions to download and install git and to link git with R Studio.

Optional: Download and install a local Github desktop client, or an alternative GUI client.

The git operations you need for this course can be managed within R Studio, from the `Git` tab. Some more advanced operations require using either a Terminal window, or a Git desktop client.

As you get going, you will likely want to learn more about how to work with git and Github. Review the documentation for git and this Github Guide. Learn the basics.

1.3 Other resources

- Github organization site, a hub for our work.
- Collab site, primarily as a shared space to access Zoom for group meetings and recordings..

Chapter 2

Introduction

- What is data science? Translation of raw data into insight, actionable intelligence.
  - Can usefully divide the work into stages of *data*, *analytics*, and *communications*.
    * *Data*: acquiring, organizing, managing, curating data (= symbols representing some measurements about the world); reformating and staging data to make it ready for analysis.
      · In practice, these activities absorb ~ 80% of the time of a working data scientist. (Your best friend is a good data manager!)
      · Hence the value of *data standards*: the more consistency in the way *meaning* is represented in data, the less time, tedium, and expense spent on data mapping and transformations.
    * *Analytics*: statistics, machine learning, modeling, other techniques for extracting summary intelligence from data.
      · In general, data science practitioners don't work on developing new techniques, nor on writing code to efficiently implement techniques. Those are specializations. Most of the work involves identifying the appropriate techniques, and the right software to implement those techniques, given the data and the goals. In some cases, work will be done in close collaboration with domain area specialists, e.g., climate scientists.
    * *Communications*: creating visualizations, narratives, etc., to reveal complexity, to convey insights. Also includes the creation of reports, dashboards, other *data products*.
- Doing data science requires a mix of skills: computing; statistics and other analytics; visual design and communications.

2.1 Reproducible workflows

Essential for us that workflows be *reproducible*, auditable, transparent.

2.1.1 The concept of reproducibility

Idea: when you present your results, you present *all* of the data and the code that another practitioner (of reasonable skill) would need to reproduce your work.[1]

We are very much a "reproducible workflow" shop. In ideal practice this means:

- If you generate a report or other data product, you should be able to reproduce it entirely by re-running your code. If you were to delete your report, you should be able to regenerate it exactly, simply by re-running your code. This process should require no manual inputs from you, no cutting-and-pasting.
- If you provide someone else with your code and your data, they should be able to reproduce your report by running your code on their machine. This should be true even if they are using a different type of machine, with a different operating system and directory structure. They should not need to edit your code, make manual inputs, or perform any cut-and-paste operations.

2.1.2 How we create reproducibility in practice

These conditions are ideals. In practice, we want to get as close as we can, within reason. Many of the tools that we use, and work processes we adopt, are designed to make our workflows hew closely to this ideal of full reproducibility.

R Markdown is designed specifically to enable the generation of reports and other products reproducibly.

Git and Github enable frequent backup of code and other files, and collaboration between colleagues without losing version control.

Cloud servers for maintaining files and databases help assure that our work is available to each other, properly backed up and curated, and organized in a structured way to facilitate reuse, without loss of version control.

We seek to avoid as much as possible exchanges of the form, "I don't know why my code won't work on your machine. It runs just fine on mine!"

[1] We distinguish *reproducibility* versus *replicability*. Your work is *reproducible* if someone else can regenerate exactly your same results, using your data and your code. Your research results are *replicable* if another researcher can get essentially the same results when applying your techniques to different but comparable data.

In particular: except possibly in the course of single work session, you should never have work-related data or code that exists only on your own machine. Ever. Your work, including data and code, should always be backed up to resources that are accessible by other team members.

2.2 Working collaboratively in a team

When working on a personal project, such as a student assignment, it is not necessary to assure that others can access, learn from, critique, reproduce, and build on your work. In a team setting, it is mandatory. We have adopted tools, processes, workflows to try to assure that your own work can be used by others and vice versa.

Chapter 3

Materials from Basecamp

R and R Studio Datacamp Cheatsheet for ggplot2 R for Data Science Coursera R Documentation Stack Overflow
Word Press R Shiny Tutorial

git and Github Background Info Version Control Info Please edit your Github profile to show your name. A photo is nice, too.

SQL and PostgreSQL Background Info How to access database? Doc&Files -> Instructions for DB access

Zotero Team Link Background Info

Panopto Team Link

Markdown and R Markdown Background More Background Info R Markdown: The Definitive Guide

One Drive Syncing R and OneDrive: Download and install the OneDrive desktop app that syncs cloud files with a folder on your local machine. Then point the R code to that folder. Note: One can use OneDrive as the home for several repos, especially if project data files are too large to post on Github. Add an appropriate line to the .gitignore file to tell git not to sync the large data files. Thus the data files can be kept locally together with the other files in the repo, and can be accessed by other team members, without syncing them to github. Link to OneDrive

Chapter 4

Data Science with R

Main reference: R for Data Science by Hadley Wickham and Garrett Grolemund.

Chapter 5

Data pipelines

5.1 Retrieving data

5.1.1 From local files

5.1.2 From APIs

5.1.3 From databases

Instructions for database access

(1) Install DBeaver (unless you already have desktop database client software you prefer): Here is a link to download for both Windows and Mac OS X. We will be using the community edition 7.0.0. for accessing PostgreSQL databases.

The Community Edition is free. Note that DBeaver uses and requires Java. If you install it via the Windows or MacOS installer then you don't need to install Java separately.

(2) Download and install Cisco Mobility VPN client: See instructions here.

After launch, select the "UVA Anywhere" network.

You need to use the VPN if and only if you are access the database from off-Grounds. When you are on-Grounds, skip this step.

(3) Log into Postgres DB:

Host: va-energy2.postgres.database.azure.com Username: [your UVA id]**?** Password: [Contact Chloe Fauvel Chloe to get your individual password] Port: 5432 Initial database: postgres

Please don't share your individual credentials.

5.1.4 From web resources

5.2 Data types

5.2.1 Data types in R

5.2.2 Conversion on read-in

5.3 Data wrangling

To learn how to wrangle and visualize data using the Tidyverse packages, you may find it useful to go through the Tidyverse Fundamentals with R modules on Datacamp. - Datacamp also offers a range of other learning modules for developing data science skills in R.

5.3.1 Tidy data

5.3.2 Dplyr

5.4 Managing data

5.4.1 DOs and DON'Ts

Chapter 6

Git and Github

A collection of files associated with a single project is in git-speak called a "repository" or *"repo"*.

Depending on your project, there may already be one or more repos on Github that you will work on. The next step is to copy ("clone") this remote repo to your local machine.

- Clone your course repo on Github to a new R Studio project on your local machine.
  - Navigate to the course website on Github. Select your repo.
  - Click on the green button labeled "Code". Copy the URL.
  - In the R Studio window, from the pull-down menu in the upper-right corner, select `New Project...`, `Version Control`, `Git`. Paste the URL into the dialog box labeled `Repository URL`.
  - Optional: Change the name of the project folder, and the location of this folder on your local directory tree.
  - Click on `Create Project`. The files from your remote repo should be copied to your local machine in a new folder with the name you chose.

6.1 Version control: what and why

6.2 How git works

6.2.1 The .gitignore file

6.3 Git in R Studio

6.4 Github

6.4.1 Using personal tokens to access Github

Github is phasing out the use of passwords for authorizations.

```
---- Forwarded Message -----
From: GitHub <noreply@github.com>
To: Arthur Small <asmall@virginia.edu>
Sent: Sunday, February 21, 2021, 6:20:58 AM EST
Subject: [GitHub] Deprecation Notice

Hi @arthursmalliii,

You recently used a password to access the repository at uva-eng-time-series-sp21/coronatc

Basic authentication using a password to Git is deprecated and will soon no longer work. Vis

Thanks,
The GitHub Team
```

Instead, you must create a personal access token. See the Github documentation.
The Cooper Center Organization on Github

6.5 Best practices

Chapter 7

R Markdown

Markdown is a markup language: a set of formatting instructions for rendering documents. R Markdown is an extension of Markdown that allows for embedding chunks of R code into a Markdown document. In this course, we will write our work in R Markdown within the R Studio environment, then use the `knitr` package to generate HTML and PDF output files.

For a nice introduction to Markdown and R Markdown, watch the short YouTube video Reproducible Reports with R Markdown (STAT 545 Episode 3-A) from the University of British Columbia.

As you proceed in creating your documents, you will probably want to access additional resources:

- From within R Studio, you can access an R Markdown Cheat Sheet via `Help/Cheatsheets`.
- Markdown reference: https://www.markdownguide.org/
- R Markdown reference: https://rmarkdown.rstudio.com/

Chapter 8

Shared resources

8.1 Network File Server

8.1.1 Instructions for mounting network file server on your own machine

For sharing certain files, especially larger data files, the project uses a 200 GB Premium network file server (NFS) hosted by UVA Information Technology Services. This network file server is pre-mounted on the project virtual machine, making the files available to all team members. If you are doing your work on a VM, you are all set: you don't need these instructions.

It sometimes can be useful to be able to access the NFS directly from your own machine, outside the VM. The instructions here tell you how to mount the NFS on your own local machine.

Credentials and network access: To access the NFS, you must be a member of the ceps-cleanenergyva group registered with UVA ITS. If you encounter a permissions issue when trying to access the NFS, check mygroups.virginia.edu to confirm you are a member of ceps-cleanenergyva; if you are not, contact Chloe Fauvel Chloe, David Hill David, or Arthur Small Arthur to request you be added. To access the NFS, you must be logged on to the UVA Anywhere network. If you are off-Grounds, you will need to access the UVA Anywhere network using a VPN. If you are physically on Grounds, you should have access automatically.

[INSERT HERE: Instructions for mounting NFS on Windows, Mac.]

Chapter 9

Bibliographic resources

9.1 Zotero

9.2 Bibtex

Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.21.