Ari Anisfeld
anisfeld@uchicago.edu

Accelerated Coding Camp Schedule

Math & Coding Camp

Summer 2021

We learn coding because it is an integral toolkit for data analysis and statistics, which are in turn fundemental to evidence-based policy. Coding camp should provide you a baseline of R skills, but will not make you an expert. Your core coursework will still be challenging as the statistics professors will push you to apply R beyond the level we cover here.
The accelerated course is aimed at those students who have previous experience learning computer science, programming as a data analyst or who have completed a fair amount of self-study.[1]
Each "class" involves

1. working through pre-recorded lectures.[2]
2. developling skills further in guided practice with Ari
3. practicing with lab material during a smaller lab section with your TA.

There will be several points during the camp where we ask students to submit work to gradescope.

1 Prerequisites

- **Mandatory:** Review lesson 0 materials: R Motivation + Intro to R Studio and Tidyverse. (A future version of this document will have a link to the material).
- **Mandatory:** Install R and R Studio prior to the start of this class. If that is daunting for you, consider joining the non-accelerated coding camp.

Class 1: Reading data files and manipulating data with dplyr

We begin the course very fast! First we learn how to read in data from several different tabular data types (csv, xlsx, dta). Then, we introduce a suite of functions for quickly understanding data. These functions will be used over and over again throughout the course with increasing levels of sophistication.

Learning Objectives

SWBAT:

- set their working directory and read in data.
- view attributes of data.
- use dplyr verb functions select(), arrange(), summarize(), filter(), and mutate().
- understand how to use the pipe operator to write longer and more organized code.

Class 2: Vectors and Data types

In this lesson, we look under the hood of tibbles and learn they are constructed of vectors. Vectors are the fundemental data structure in R. We discuss how to construct vectors, tibbles and other data structures such as lists and data.frames. Then we learn to distinguish between the four most common data types in R: integer, double, character and logical.

[1] We cover basic R skills in 6 lessons compared to the beginner section which takes 11 lessons starting (including 5 weeks in the Fall quarter). The material covered in both sections is very similar.
[2] Pause and practice the material along with the videos.

Learning Objectives

SWBAT:

- create vectors and tibbles in various manners.
- do vectorize operations and math with vectors.
- identify how to subset tibbles using base R bracket notation.
- explain data types, type coercion and solve common errors involving types.

Class 3: Using if() and ifelse() for conditional expressions.

In this lesson, we'll review logical operators (==, !=, ¿, ¡ etc) to compare two vectors item by item. We will then introduce conditional expressions, which allow us to do things based on context. We will use conditional expressions in a data analysis context using ifelse() with mutate() in tibbles to create new columns. We will also introduce the non-vectorized if() and else pattern which is used in a programming context.

Learning Objectives

SWBAT:

- use control flow with if() and if() / else() statements.
- use ifelse() and case_when() statements in conjunction with mutate() to create columns based on conditional statements.

Class 4: Grouped analysis with dplyr and

In this lesson, we gain a powerful tool called group_by() which allows us to conduct analyses on multiple subsets of our data simultaneously. For example, consider a data set with multiple rows of data for each country. group_by() allows us to do the same analysis for all the countries at one time. We will use grouped data to create summaries and to add columns to the complete data set based on group specific information like within group rankings.

Learning Objectives

SWBAT:

- summarize data by group with group_by() %¿% summarize()
- create new data by group with group_by() %¿% mutate()

Class 5: Functions

In the final two class, we shift focus from data analysis to core programming skills. First we introduce functions. Functions allow us to avoid rewriting (or copying and pasting) code to create cleaner and more useful code. We introudce how to write functions with a focus on best-practices for writing code including using names that are clear and consistent and debugging code while writing it.

Learning Objectives

SWBAT:

- explain when and why to use functions.
- identify that functions consist of arguments and a body and are usually assigned to a name.
- coherently name functions and their arguments.
- write simple functions.

Class 6: For-loops

Writing for-loops is a core programming skill. Loops are a way to repeatedly call code over a specific set of inputs. In R, we have so many vectorized functions that loops are often overkill. Yet, we still need for-loops on occassion. In this lesson, we motivate the use of for-loops and discuss the steps to writing a for-loop which includes

- defining what you will iterate over
- preallocating an object in memory to store output
- writing the body of the for-loop

As in the previous lesson, we will highlight coding practices such as debugging with print() statements.

Learning Objectives

SWBAT:

- explain that iteration is useful when we are repeatedly calling the same block of code or function while changing one (or two) inputs.
- recognize whether vectorized functions or loops are more approriate.
- write for-loops by defining what to iterate over, preallocating space and then writing parameterized code

Bonus Class: Data Visualization

If we had a 7th class it would be focused on data visualization. This material is dispersed throughout the earlier lessons. Here we provide material that helps you understand how to use ggplot() to create a variety of plots for data exploration and communication. ggplot() is built on the idea of mapping columns of data to aesthetics (e.g. x-axis, y-axis, color) and then layering aesthetics together.

Learning Objectives

SWBAT

- explain how to tell ggplot to map data to aesthetics.
- distinguish between 'continuous' vs. 'discrete' variables.
- use geoms to create different types of plots (e.g. histograms, bar chart, dot plot)
- explore data with simple visualizations.