

# SMEA 500: Coding in R for Natural and Social Sciences

Fall 2018

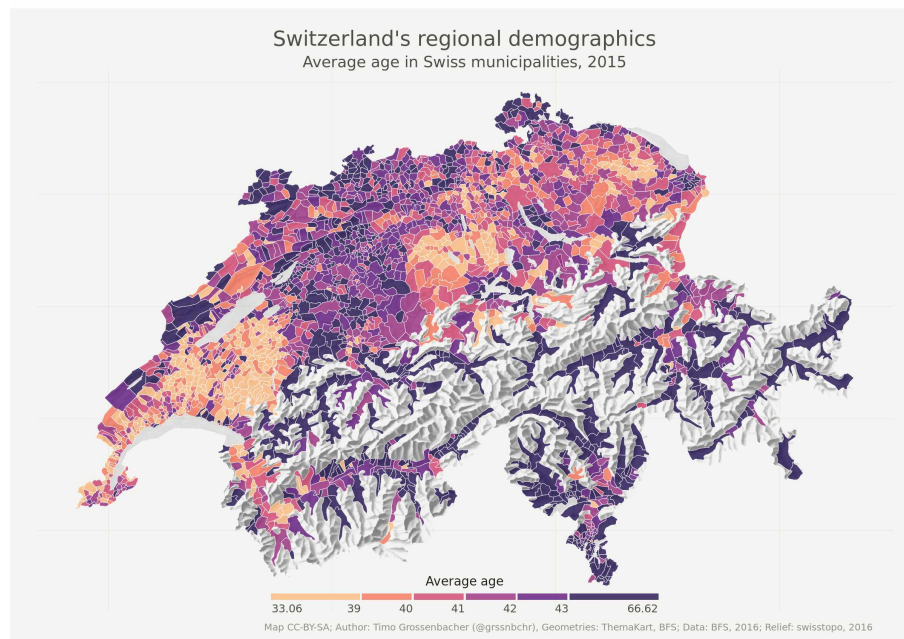
Wednesdays 12:30pm -- 2:20pm

OUG 141

Ryan Kelly & Ramón Gallego

Office Hours: By appointment

✉ rpkelly@uw.edu ; rgallego@uw.edu



Some degree of computer coding is increasingly a required skill for all kinds of careers. As natural and social scientists, we often have to analyze large datasets or undertake repetitive analytical tasks that do not lend themselves well to working with spreadsheets. We generally would also like to create powerful graphics that clearly express the conclusions we reach. The free, open-source statistical coding language R has become a common way for academics and others to tackle these challenges. This course will provide an introduction to R and give students the tools to become autonomous users if they so desire. Each week will feature a topical presentation, followed by workshop time during which students can troubleshoot and practice their skills on work of their own choosing (e.g., stats problem sets, thesis analyses, or graphics for capstones). No prior coding experience is required, but an open and exploratory mindset will be essential.

## Learning Goals

---

- 1) to become familiar with fundamentals of R
- 2) to become a more productive and efficient user of data

- 3) to overcome any fear of computer coding that may be lingering in your mind
- 4) to do work you already have to do, but do it better

## Format\_\_\_\_\_

We will spend the first half of each class session in lecture, and the half in workshop to practice the newly acquired skills (or to do your homework). You should bring a laptop to class with you, with a version of R running. If you do not have a laptop, you can borrow one from SMEA or from the library. All books/materials/etc. will be provided, also for free.

## Book\_\_\_\_\_

We weren't planning to use a book for this class, but it turns out there's a great one that's free and fun. It's YaRrr! The Pirate's Guide to *R*. You can download both the book and the source code used in the book [here](#). It's by a guy named Nathaniel Phillips, and it would make a good deal of sense to send him some money to support his good work. He has a paypal link on the page where you get the book/code.

## Grading\_\_\_\_\_

This class is Pass/Fail. You will Pass as long as you show up and try. You may be asked to drop for more than one unexcused absence, or for violations of the College's code of academic conduct (see below).

## How Do I Get and Install R ?\_\_\_\_\_

There are choices.

- Most people like to use RStudio, which is a fully-integrated development environment (that is, it's pretty, and has all kinds of useful functions). To get that, go here: <https://www.rstudio.com/products/RStudio/#Desktop>, and download/install the free version, according to the instructions. For this class, just use this one, so that everyone is on the same page.
- For the record: those of you who want a more stripped-down, customizable version can use plain old R. Go here: <https://cran.r-project.org/>. Follow the instructions.
- For those of you who want to fully live in the Matrix, and use R only at the command-line, you probably aren't taking this class. But there are instructions for this [here](#).

R has a large and vibrant community of users, contributors, and supporters. We will soon see that many of the functions of R come from users themselves, who contribute code to the overall project. You then download and install these contributed functions (called "packages"), extending the base-R that you got when you first downloaded it. In this way, R is a community-wide project that is always expanding.

And this means that if you have a question, Google it. You will find answers to nearly every question on one of a few sites, such as <https://stackoverflow.com>.

## Computers, Phones, etc.\_\_\_\_\_

Normally we effectively ban laptops from our classes, but for this class they're indispensable, obviously. So we do ask that you use them with due respect for the class and for your classmates. Please do not be buying shoes, using Facebook, or talking on the phone in class.

## Academic Integrity

---

You are responsible for understanding the College of the Environment's rules on academic misconduct. See <http://coenv.washington.edu/intranet/academics/academic-policies/academic-misconduct/>.

Note that it's not obvious what constitutes plagiarism in the case of computer coding (see [this article](#)). But for our purposes, can we just agree that you're going to do your own work? Certainly, copying code from others is often an accepted part of computer coding -- particularly in an open-source environment -- but there is a difference between using 2 lines of code and using a whole assignment.

# Schedule

---

## Week 1

September 26 • Introduction to the course and to R . Objects, functions, and getting to be a bit more comfortable with R .

Reading: Pirate's Guide Chapters 1 and 2. Always remember that clear thinking leads to sensible analysis, rather than the other way around.

Assigned: Problem Set 1. Due October 9.

## Week 2

October 3 • Basic computation and stats, and getting data into (and out of) R .

Reading: Pirate's Guide Chapters 5 and 9.

## Week 3

October 10 • Plotting 1 (base R ).

Reading: Pirate's Guide Chapter 11

Assigned: Problem Set 2. Due October 23.

## Week 4

October 17 • Data Cleaning and Management.

Reading: Pirate's Guide Skim/Review Chapter 9.

## Week 5

October 24 • Loops and other ways to iterate.

Reading: Pirate's Guide Chapter 17.

Assigned: Problem Set 3. Due November 6.

## Week 6

October 31 • Testing Hypotheses.

Reading: Pirate's Guide Chapter 13.

## Week 7

November 7 • Plotting 2 (ggplot).

Reading: Introduction chapter of ggplot2 (book), by Hadley Wickham. And this [tutorial](#) is pretty useful.

Assigned: Problem Set 4. Due December 4.

## Week 8

November 14 • Dipping a toe into the Tidyverse.

Reading: David Robinson, "Teach the Tidyverse to Beginners"

## Week 9

November 21 • Aggregation and Grouping

Reading: None

## Week 10

November 28 • Tidyverse 2

Reading: Skim ggplot2 book for useful nuggets.

## Week 11

December 5 • Project Management and Workflow: putting everything together from setup to cleaning, importing, calculating, testing, plotting, and presenting.

Reading: None