

Selected Advanced Topics in Biology: R-Based Analyses & Programming for Community Ecology

Course Number: BIOL709B

Instructor: Nate Swenson; swenson@umd.edu; BPS 4207

Location: BPS 1208

Time: Thursday 1-3pm

Course GitHub Repository: <https://github.com/NGSwenson/Swenson709Bfall2017>

Overview: This course is designed for grad students in the BEES realm that are interested in learning to use the R programming language for their research and future careers. The course is not explicitly a “stats course”. Rather, the course is to develop a student from a basic/general R user that can utilize existing functions to a more advanced user that can write and read R code. As it is expected many/most students will have little to no experience writing their own code, we will take this opportunity to develop those skills more formally. Specifically, we will focus on writing versioned code that is well commented thereby making it easily replicated and understood in the future by others and yourself. Additionally, we will develop skills relating to how to break apart computational problems, speeding up code and reporting output. At the beginning of the course I will provide an overview of community ecology analyses and R packages that are those most heavily used to form a baseline of knowledge of analyses and basic R. We will use example datasets, but can also utilize any data a student may want to analyze from their own work. We will then transition towards the more ‘code-y’ part of the course where we will learn the fundamentals of R programming and utilize those skills for a class project that will be presented at the end of the semester.

Class Project: In my experience, learning R rapidly accelerates when there is a problem (of interest) to be solved. Specifically, learning by following a book is often tedious and often not useful unless the examples in that book are very relevant to ones current research. In other words, a student is far more likely to figure out how to solve a problem if it is for the sake of advancing their research than for the sake of simply learning it. Thus, I will ask students to identify a moderately- to substantially-size research objective from his/her thesis or dissertation. This problem will be outlined to the instructor in an email late in September. If the problem is suitable for the class, then the student will present it to the class early in October in as much detail as possible (i.e. what the problem is, why you want to address it and any ideas you may have on the types of analyses and/or code you would need to develop/use to solve it). Through the next 1.5 months the student will work on the project providing regular updates/versions of the work to GitHub while also attending class meetings to continue learning R programming. Two class meetings before Thanksgiving are devoted to discussing projects, identifying problematic code or obstacles and troubleshooting. After Thanksgiving, our last two class meetings will be a formal presentation of the class project by each student roughly 15 minutes long. NOTE: If a student does not have an available dataset, then they can discuss with the instructor regarding publicly available data that is similar to that to be collected in their thesis or dissertation. Additionally, the projects do NOT have to be community ecology *per se*.

Some Useful Texts (probably):

- Bocard, D., F. Gillet, and P. Legendre 2011. *Numerical Ecology with R*. Springer Press, New York, New York.
- Boehmke, B.C. 2016. *Data Wrangling with R*. Springer Press, New York, New York.
- Stevens, M.H.H. 2010. *A Primer of Ecology*. Springer Press, New York, New York.
- Wickham, H., and G. Grolemund. 2016. *R for Data Science*. O'Reilly Media Inc., New York, New York.

Timeline for course

| Meeting | Date | Topic | Homework |
|---------|-----------|--|---|
| 1 | 31 – Aug | R Intro, TextWrangler/BBedit | Running existing functions; Write a new function |
| 2 | 7 – Sept | More R Intro & Basic Population & Community Ecology Analyses | Analyze example datasets & email Swenson a dissection of a function |
| 3 | 14 – Sept | More Advanced Population & Community Ecology Analyses | Analyze example datasets & email Swenson a dissection of a function; Email Swenson a problem that you will work on for the class project |
| 4 | 21 – Sept | GitHub, Version Control, Terminal, Summarizing, Visualizing & Wrangling Data I | Set up GitHub account and use it, Run R in terminal |
| 5 | 28 – Sept | Summarizing, Visualizing & Wrangling Data II | Write code to read in, visualize and summarize several files provided and commit |
| 6 | 5 – Oct | Project idea presentation, how to tackle problems, & modular programming | Write a schematic detailing the workflow, the input/output, goals and functions (existing or not) and commit |
| 7 | 12 – Oct | Speeding things up, sampling and re-sampling | Meet with Swenson to discuss strategies to solve problem, address first one (or several) steps in workflow and commit |
| 8 | 19 – Oct | Markdown via Knitr | Commit a Markdown document for the first several steps in workflow |
| 9 | 26 – Oct | Simulating Data | Simulate random and non-random datasets and run through your workflow - commit |
| 10 | 2 – Nov | Troubleshooting Projects & Specialized Analyses: I | Commit all updates on your project and document achievements and known obstacles or errors |

| | | | |
|----|----------|---|--|
| 11 | 9 – Nov | Troubleshooting Projects & Specialized Analyses: II | Commit all updates on your project and document achievements and known obstacles or errors |
| | 16 – Nov | NO CLASS | Commit all updates on your project and document achievements and known obstacles or errors |
| | 23 – Nov | THANKSGIVING | Commit Presentations (Group 1) |
| 12 | 30 – Nov | Project Presentations Group 1 | Commit Presentations (Group 2) |
| 13 | 7 – Dec | Project Presentations Group 2 | |

COMPUTATION OF GRADES

| | |
|------------------------|--------------|
| In Class Participation | 130*10 = 130 |
| Homework | 12*10 = 120 |
| Project Topic | 10 |
| Project Final Commit | 200 |
| Project Presentations | 40 |
| Total | 500 |

Grading scale

| | |
|-------------|----|
| 100-98% | A+ |
| 97-93% | A |
| 92-90% | A- |
| 89-87% | B+ |
| 86-83% | B |
| 82-80% | B- |
| 79-77% | C+ |
| 76-73% | C |
| 72-70% | C- |
| 69-67% | D+ |
| 66-63% | D |
| 62-60% | D- |
| 59% or less | F |