Data Sciences Practicum

Course Description

A practical introduction to contemporary data analysis with a focus on improving programming literacy, research workflow, and reproducibility. A majority of the workload is designed to complement and advance the student's current research. Covers programming basics, exploratory data analysis, and common modeling tasks using the *R* programming language. Introduces cutting-edge data science concepts, including distributed version control, the grammar of graphics, and dynamic documents. Rather than focusing on the particulars of the language, learning objectives are targeted toward building the familiarity and confidence necessary for future self-study.

Course Logistics

Sessions

- 12 Weekly Sessions
  - 3 Hours In Length
  - First 1.5 Hours of Instruction
  - Final 1.5 hours are optional, and serve as open office hours with instructor and collaboration with classmates

Course Notes

- **Course notes** will be published online in both HTML and PDF format, at least one week prior to the relevent session
- Course notes will provide **explicit examples for the basic concepts** students are expected to be familiar with, and **vetted online references for advanced** concepts which may be helpful to their particular research

Homework

- **Weekly review exercises** of less than 2 hours in length will be assigned. They will be relevant to the course notes of the particular week. Some of these exercises will involve **peer review** of code, so it is important that the assignments be submitted on time.

- Optional **challenge exercises** will be occasionally posted. These exercises will be based on advanced concepts from previous sessions. Successful completion of a **challenge exercise** will earn extra credit equivalent to a **weekly review exercise**.

Project

- The course project will ideally be based on student's current research. The purpose of the project is to:
  - Resolve pre-existing data analysis issues
  - Refine the student's current research and data analysis workflow
  - Expose the student to best practices for documentation and reproducibility
  - Practice communication of experimental design and results
- The project will have with **three** milestones:
  1. **Proposal**: The proposal consists of two parts:
     - A general description of the student's current research and *a scope of work to be attempted* within the semester towards this research project.
     - In the context of your research project, identify a common frustration, existing limitation, or *bad habit* that you would like to improve on over the course of the semester. Work the instructor to develop a *course of action* to resolve the problem.
  2. **Presentation**: Starting on **week 7** each class will feature a student presentation. In the presentation the student will communicate:
     - the research question / hypothesis
     - the data analysis methods used or needed
     - describe the features of a relevant dataset to the class using exploratory data analysis
  3. **Report**: The project will culminate in a project report. The report will include all necessary code and data to reproduce the findings. The grade will be based partially on the reproduction of results, but also on style, documentation, and overall effort.

Learning Objectives by Session

Unit I: Getting Started With *R*

1. oRientation
   - **Become familiar** with the *R* and *RStudio* environments

- **Understand** the differences between basic data types
- **Understand** basic variable assignment
- **Exposure** to data importing and exporting functionality
- **Exposure** to *git* for version control

2. Programming Control Structures

   - **Understand** how to control programs using conditional statements
   - **Understand** Understand how to use *functions* and *loops* to compartmentalize code
   - **Exposure** to *apply* statements and *style* standards

3. **Topical**: Monte Carlo Simulation

   - **Understand** how to generate random numbers in R
   - **Understand** how to to use the *replicate* function
   - **Exposure** to *parallelization* to perform multiple computations simultaneously

4. Introduction to Graphics

   - **Become familiar** with basic graphics functions
   - **Understand** the basics of graphics devices and how to save outputs
   - **Understand** the ideas behind **the grammar of graphics**
   - **Exposure** to **ggplot2**, which implements **the grammar of graphics**

5. Debugging and Improving Performance

   - **Become familiar** with *RStudio* debugger
   - **Become familiar** with *lineprof* package
   - **Revisit** *apply* statements and *parallelization*
   - **Exposure** to common performance pitfalls

Unit II: Working with Datasets

6. **Topical**: Exploratory Data Analysis

   - **Become familiar** with methods for determine structure of a dataset
   - **Understand** how to check for missing data
   - **Understand** how to make quick diagnostic plots

7. Harvesting and Cleaning Data

   - **Understand** how to use *Rstudio* to import data
   - **Exposure** to tools for downloading live data from the web
   - **Exposure** to the *plyr* package

8. Working with 'Big Data'

- **Exposure** to the *data.table* package
- **Enhanced understanding** of *plyr* package

- **Become familiar** with efficient means of altering and summarizing data

9. **Topical**: Regression

    - **Become familiar** with *model objects* and accessor functions
    - **Become familiar** with *formulas*
    - **Understand** how to run a basic linear regression model with *lm*
    - **Understand** how to graph model outputs
    - **Exposure** to generalized linear models with *glm*
    - **Exposure** to *leaps* package for best subsets regression

Unit III: In the Driver's Seat

10. **Creating Dynamic Documents**

    - **Exposure** to *Rmarkdown*
    - **Understand** how to create a basic dynamic report
    - **Understand** how to print system information for reproducibility

11. **Creating Packages**

    - **Exposure** to *buildtools*
    - **Exposure** to *unit tests*
    - **Understand** how to make *Roxygen2* comments
    - **Understand** the basics of the *DESCRIPTION* file
    - **Understand** how to store package for later use

12. **TBD / Slack**