Earth 125/225: Statistics and Data Analysis in the Geosciences Winter 2019

Course goals

Geoscience is becoming increasingly data-driven and it is more important than ever to know how to analyze and interpret the meaning of your data. The goal of this course is to introduce you to a variety of data analysis methods that you can use in research, graduate school, or during your future career. At the end of this course, you will know which test to use to answer particular types of questions and you will be able to interpret the results and their meaning. This will require you to understand your data and recognize the requirements, assumptions, and limitations of statistical analysis.

This course will also introduce you to coding, an extremely practical skill for research and something that can set you apart with employers. You will learn to use R, a powerful open-source programming language designed for data science and widely used in both academia and business. At the end of this course, you will be able to manipulate data, perform statistical tests, program functions for data analysis, and plot and interpret results using the R programming language.

Course structure and approach

The best way for you to succeed is by spending hands-on time coding in R, so there are no lectures and class time is intended for you to work through exercises and get help. Coding will come more easily to some people, while the learning curve will be steeper for others, so this course is designed so that you can work at your own pace. Instead of a set schedule of topics and deadlines, you will instead work through modules on Canvas that will build up your coding skills and statistical knowledge. Modules include videos to introduce concepts and exercises to apply those concepts and gain coding skills.

Evaluation uses a mastery-based framework, which assigns grades based on demonstration of mastery and progress through the series of modules, rather than from timed assessments like problem sets or a final exam. You can spend as much time as you need on each module, but you must complete all exercises within a module and answer all questions correctly before moving on to the next module. It’s best to do the exercises in order, but if you get stuck you can skip ahead within a module. You’re also able to submit exercises as often as is necessary to demonstrate mastery. This means that even if the learning curve is steep, everyone will have the chance to succeed without having to worry about falling behind.

Module exercises will be available and can be submitted at any time before 5 PM on Friday March 22.

Getting help

This course is intended for students with no prior statistics or coding background. While the material will require hard work on your part, my goal is to support everyone’s learning. You can work through the modules independently, whenever and wherever you want, but I will be available to help you during class time, in office hours (either in person or virtually with the Zoom meeting software), or by email. If you email, please reference the dataset you’re using, include the code you ran, and copy the error message.

Grading

You can gain points in two ways. First, you will earn 4 participation points for each day that you submit answers to a quiz or assignment, up to a maximum of 20 points per week. Second, you will earn points for each exercise you complete. Point values for each exercise are listed in the outline of modules on subsequent pages. The table below shows how points will correspond to final letter grades.

Points	<240	240	280	320	360	400	440	480	520	560	600	640	680
Grade	F	D-	D	D+	C-	C	C+	B-	B	B+	A-	A	A+

Outline of modules

Foundation

Descriptive statistics module:

A brief introduction to R (5 points)
Reading data files (5 points)
Object classes in R (5 points)
Working with data frames (5 points)
Plotting histograms (5 points)
Central tendency: theory (5 points)
Central tendency in R (10 points)
Dispersion: theory (5 points)
Dispersion in R (10 points)
Standard error and confidence intervals: theory (5 points)
Standard error and confidence intervals in R (10 points)

Level 1

Univariate tests module:

Selecting rows from a data frame (5 points)
The basics of null-hypothesis significance testing (5 points)
The t test (10 points)
The F test (10 points)
Add-on packages (5 points)
Grouping and summarizing data (10 points)
Analysis of variance (ANOVA) and Tukey HSD test (10 points)

Non-parametric tests module:

Shapiro-Wilk test (10 points)
Q-Q plots (5 points)
Graphing with ggplot (5 points)
Cumulative density function plots (5 points)
Selecting rows from data frames with more complex criteria (5 points)
Kolmogorov-Smirnov test (10 points)
Box and whisker plots (5 points)
Mann-Whitney U test (10 points)
Kruskal-Wallis test (10 points)
Levene’s test (10 points)

Categorical tests module:

Count data and contingency tables (10 points)
Creating count data (5 points)
Goodness-of-fit: exact binomial test, exact multinomial test (10 points)
Creating contingency tables: matrices (5 points)
Tests of independence: chi-squared test, Fisher’s exact test (10 points)

Correlation and regression module 1:

Scatter plots (5 points)
Parametric and non-parametric correlation (10 points)

- Researcher degrees of freedom (5 points)
- Linear regression: ordinary least squares (10 points)

Level 1 summary assessment (36 points)

Level 2

Correlation and regression module 2:

- Data transformations (5 points)
- Multiple regression (10 points)
- Partial correlation (10 points)
- Best practices for data organization (5 points)
- Logistic regression (10 points)

Multivariate tests module:

- Hotelling T^2 test, including Mahalanobis distance (10 points)

Ordination module:

- Creating and joining data frames (10 points)
- Selecting columns from data frames (5 points)
- Principal component analysis - PCA (10 points)
- Non-metric multidimensional scaling – NMDS (10 points)

Level 2 summary assessment (36 points)

Level 3

Resampling methods module:

- Bootstrapping (10 points)
- Two-sample resampling tests (10 points)
- Null models (10 points)

Maximum likelihood estimation module:

- Estimating parameters with maximum likelihood estimation (10 points)
- Model selection with AIC (10 points)
- Exploratory statistics vs. hypothesis testing (5 points)

Correlation and regression module 3:

- Relaxing regression assumptions (5 points)
- Generalized least squares regression (10 points)
- Time series and first differences (10 points)
- Quantile regression (10 points)

Level 3 summary assessment (36 points)

Above and Beyond

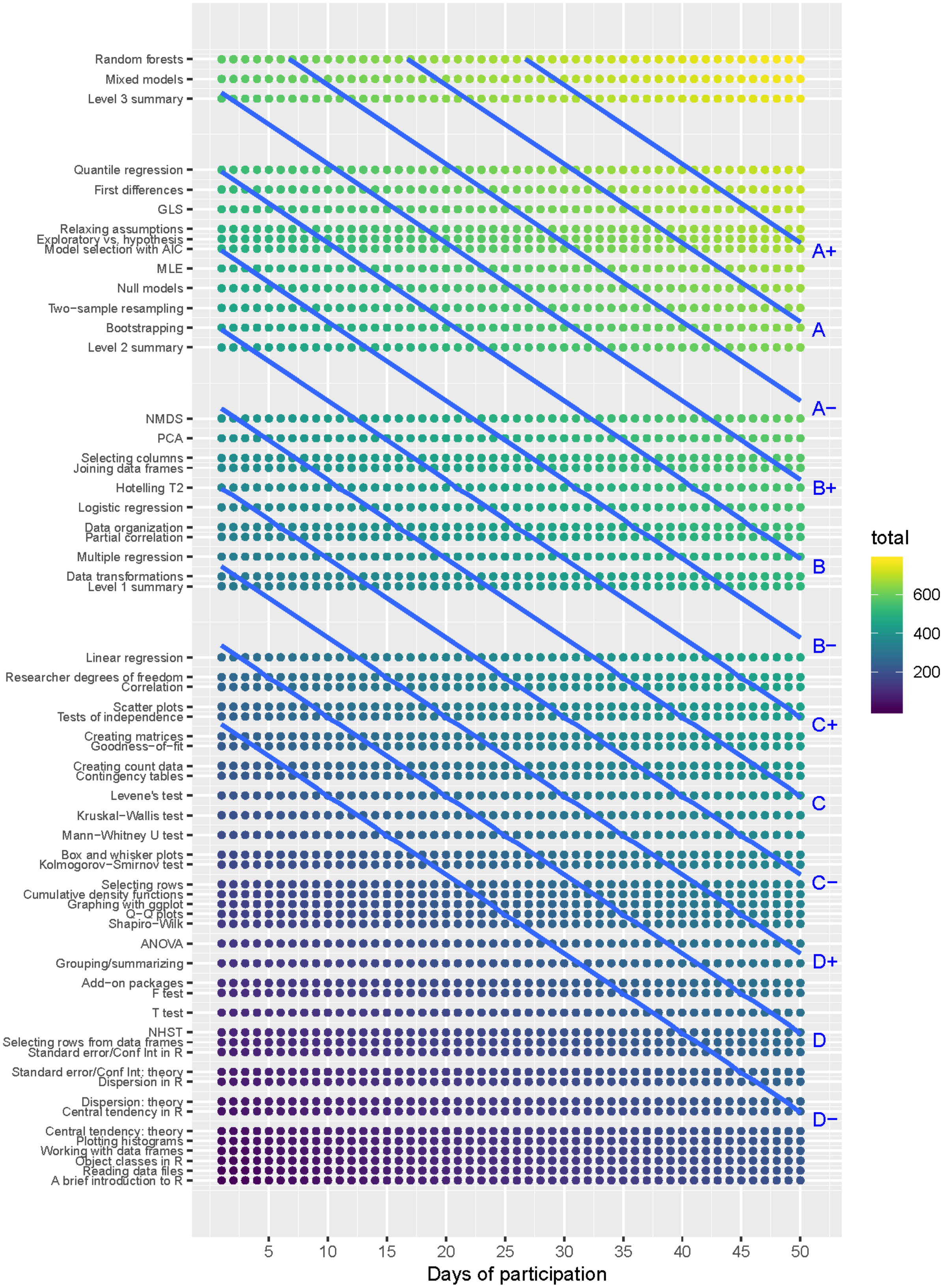
Correlation and regression module 4:

- Mixed models (10 points)

Machine learning classification module:

- Decision trees and random forests (10 points)

You can use this graph to track your progress in the course, based on the number of days that you have submitted responses to an exercise (x axis) and the number of exercises you have completed (y axis).



Contact info/Office hours

Email: mclapham@ucsc.edu, office: A208, phone: 459-1276

Office Hours: Will be held in the VisLab (EMS building A170) to provide space for more people. The room is booked and I will be there from 10-11:30 on Tuesdays and 3-4:30 on Wednesdays. You can also find me at my office at other times if you have more questions.

Academic integrity

Coding can be very frustrating – believe me, I know! I've been using R fairly seriously for 10 years and I still get angry at times, and I still have to google how to do things. But it's crucial for you to work through these challenges if you want to learn coding skills. However, you don't have be completely self-sufficient; working with classmates allows you to elaborate on the ideas and to explain your thought processes, which can also help learning. It's OK to discuss a problem with someone else in the class if you are both working on it. But you shouldn't give the solution to someone if you've already solved the problem, and you shouldn't accept answers from classmates if you're working on the problem. Likewise, you can work with classmates to figure out the appropriate statistical test for a particular problem. But you shouldn't tell someone what to do if you've already completed that exercise, and you shouldn't ask for answers from classmates. It's also OK to help someone debug their code, but you shouldn't share your code with others or use code from classmates. Remember, you can always get help from me in class, during office hours (in person or via chat), or by email.

Tips for success

This class is challenging, but if you hold yourself to high standards and if you put in the work you will become proficient at coding and data analysis. The grading is designed to reward effort and to give everyone a chance to excel. I'm also here to help everyone succeed. Here are some tips that might help you get the most out of the class:

1. Set aside time to work through the modules outside of the regular class meetings, and come to class or office hours if you have questions. There aren't deadlines for homework, papers, or exams, so you will need to be self-directed to make progress. It may help to schedule set times on your calendar, and to take advantage of times when workload in your other classes is lighter. You should try to work on the class at least 5 days a week, and at least an hour or two per day. You will receive an automated reminder email if you haven't submitted any exercises for a few days.

2. Stick with it! Coding can be inherently frustrating because you spend most time trying to figure out why it isn't working, especially as you tackle more complicated tasks. You may feel confused (or even angry!) at times, but it will all start to come together.

3. When your code returns an error message, you will need to debug it. Run your code in the smallest chunks possible, even running just a small part of a single line, to diagnose which part might be causing the problem. If you've found the problem but don't know what to do, or if you can't find the problem, don't hesitate to ask. Like any language, learning to code is hard but I aim to provide a supportive environment where everyone can succeed.

4. Modules are grouped into levels based on the sophistication of the method and the complexity of coding, so you should budget more time per exercise than on the earlier ones. In order to finish all of the material, aim to finish all level 1 modules by the end of week 3, level 2 by the end of week 6, and level 3 by the end of week 9.