STOR 320: Introduction to Data Science
Summer session one, 2020

Instructor: Brendan Brown
E-mail: `bb@live.unc.edu`
Office hour: M 1:00-2:00 pm
Office half-hour: W 8:30 am - 9:00 am

Lectures: **First lecture held live at 9:45 am, Eastern, on Wednesday May 13.**
Zoom link provided through Sakai message.
Remainder of course uses asynchronous instruction.
See 'Instruction' section below.

Website: `brendanrbrown.github.io`

Assistants: *Haixu Ma*
E-mail: `haixuma@live.unc.edu`
Office hour: 1:00-2:00 pm Thursday
Wan Zhang
E-mail: `wanz63@live.unc.edu`
Office hour: 10:00-11:00 am Tuesday

Description: This course is an application-driven introduction to data science. Statistical and computational tools are valuable from Silicon Valley startups, to marine biology labs, to news organizations. More importantly, they give us means to explore our world through data—even if you do not end up with a job in data science per se. These tools require technical skills such as programming and statistics. They also require professional skills such as communication, teamwork, problem solving, and critical thinking.

You will learn these tools and hone these skills through hands-on experience working with datasets provided in class and downloaded from certain public websites. More than half of the course is dedicated to fundamental technical skills in data science: the manipulation, preparation and exploration through summaries and graphs of data. We will cover additional topics to round out the skill set, such as modeling, web scraping and more advanced data presentation.

Come to every class with your computer and be ready to work with others. Using resources around you is a key component of successful data analysis.

Textbook: **R for data science**, by *Hadley Wickham*
available free online `https://r4ds.had.co.nz/`
and
An introduction to statistical learning, by *James et al*
available free online `https://faculty.marshall.usc.edu/gareth-james/ISL/`

Prerequisites: STOR 155 or an equivalent introductory statistics course.

Grade:

Tutorials (15 %)
Homework (30%)
Project (55%)

Instruction:

All lectures after the first will be pre-recorded and available on Sakai. Accompanying course slides, regular tutorial exercises, and homework are posted on the course github website.

I will not hold live lectures, and therefore attendance is not expected or required.

Each video will represent a single set of slides and the accompanying tutorials. However, **often multiple videos will represent a single 'class period', associated with a single day of the course**.

This is necessary because of the summer school's condensed timeline. Which slides relate to which 'class periods' will be made clear on the course webpage.

Contacting me:

- **If you email me by 5:00 pm Eastern on a given day, I will respond that day.**
- Otherwise, I usually will not respond until the following day.

Tutorials:

The best way to learn coding is by imitation and repetition. Tutorials encourage you to imitate methods presented in class and will help build coding skills necessary for homework, the project and beyond.

There will be one tutorial per slide/video, which means usually **one to two tutorials each day**.

These are small tasks asking you to imitate methods presented in lecture.

They are intended to be helpful, rather than to put pressure on you to perform. However, I will grade your submissions as described below.

- **Tutorials are intended to be completed without the instructor's help, aside from that provided in the lecture videos.** Rely on internet searches, lecture and textbook materials.

- **Must be submitted in an html markdown file to the IA Wan Zhang at wanz63@live.unc.edu**

- I will select 10 people at random and ask their teams to email me the html files with their work, in class. Over the course of the semester, I will ensure each person has had a chance to submit tutorial work.

- As time allows, I will pick one of the submissions to review immediately in class.

- **Tutorials will be graded on effort and completion:** If you have demonstrated a valid attempt to complete the task correctly, you will receive full credit.

Homework:

- Homework is assigned weekly, in general.

- Homework is due by the beginning of class, on the given due date, by uploading the submission to Sakai.

- Each **homework point** is worth the same amount toward your final grade. So an assignment worth 80 points will be worth twice an assignment worth 40 points in your final grade.

- You may discuss homework with classmates and teaching staff. But you must submit your own work.

- You may and often should search online for solutions to coding problems. This is perfectly fine and encouraged.

- However, copying responses from students who have taken the course, including from sources online, is unacceptable and could be treated as an honor code violation.

- **Homework must be submitted as the html output from an R Markdown file.** In other words, your homework submission must be a .html file with all code and writing, as produced in R Markdown.

  Submissions that do not ‘knit‘ to html will not be accepted. Such cases most often result from errors in the code, which students must correct before submission.

- Late homework submitted less than 24 hours from when it was due will have its **score reduced** 50%.

- Homework later than 24 hours or a failure to adhere to the rules above will result in a **score of zero for that assignment**.

Project: The final project is done in groups of two and worth a total of 100 points. Groups will be randomly assigned.

- **Part one:** Project Proposal and Exploration. Worth 30 points and will be due approximately half-way through the semester.
- **Part two:** Final Paper. Worth 40 points and must be submitted on Sakai by 11:55 PM on the last day of classes, June 15.
- **Part three:** Final Presentation. Worth 30 points and will take place during the course's designated final exam time according to the university calendar.

Topics

Core programming and data science skills
– R Markdown
– data frame creation and manipulation
– summary statistics
– visualization
– exploratory data analysis
– 'tidy' and relational data
– functions and functional programming
– string manipulation and regular expressions
Modeling
– cross-validation
– linear and generalized linear models
– classification techniques
– clustering
Advanced topics
– Shiny
– more advanced modeling with support vector machines and tree-based methods
– web scraping

Honor Code: `http://instrument.unc.edu/`