STAT 447: Data Science Programming Methods

Dirk Eddelbuettel

Department of Statistics, University of Illinois

Syllabus for Spring 2024 Term – Last updated 09 Feb 2024
Website at `https://stat447.com` may be more current

> Data science is emerging as a field that is revolutionizing science and industries alike. Work across nearly all domains is becoming more data driven, affecting both the jobs that are available and the skills that are required. As more data and ways of analyzing them become available, more aspects of the economy, society, and daily life will become dependent on data.
>
> *Source:* Envisioning the Data Science Discipline: The Undergraduate Perspective. National Acadamies, 2018.

Overview

This course, provides the principal *programming* foundations for *working with data at scale*.

Data analysts are in demand, and particularly those who can *walk the walk* and not only *talk the talk*. This course aims for a "hands-on, roll-up-your-sleeves" learning-by-doing approach which can be highly rewarding to those willing to put in the required effort.

This printable version offers an alternate for the course site.

This course builds upon its predecessor STAT 430 Topics Course *Data Science Programming Methods*.

Learning Objectives

After this course, students should be able to ...

Key features below

- *analyze* code in multiple data science programming languages;
- *write* short programs in several relevant languages;
- *manipulate* files and data using the command-line;
- *access* data from relational databases using basic SQL commands;
- *utilize* git for version control, collaboration and publishing;
- *use* R as a language and environment for *programming with data*;
- *solve* new data science problems using these tools;
- *know* a variety a tools based on first-hand experience;
- *show your skills* via a group-project with a topic of your choice.

Credits

The course counts for *three credits* for undergraduate students, and for *four credits* for graduate students. Graduate students are required to submit more extensive homework assignments.

Key Facts

Core Content

- `shell` for managing files, commands, information flow, . . .
- `git` for modern version control supporting social computing
- `sql` as a base layer for data management and control
- `markdown` for programmatic control of html, pdf, . . . communication
- R for programming with data, and our core building block
- plus some extras such as Docker and more

See the website stat447.com for more.

Instructional Staff

Name	Location	Hours	Type / Booking
Dirk Eddelbuettel	Zoom	Mon 11am - noon	Open
Instructor	Zoom	Mon 6pm - 7pm	Open
	Calendly	Thu 7pm - 8:30pm	15m, one-on-one
Linjun Huang	Zoom	Wed 11am - noon	Open
TA	Zoom	Fri 11am - noon	Open

We offer two types of office hours. The first type is *open* with an open door where you can walk in and out, attend every week, or never—as you see fit. The second type are individual one-on-one office hours that fifteen minutes each, and which you book via the calendly link above. We ask that you limit your use of these to two or three per term to allow everybody a turn. Under *genuinely exceptional circumstances*, additional visits can be scheduled on demand. (Note that the Zoom links above differ per time slot. Make sure you pick the correct one.)

Lecture Location

What	When
Location	online
Times	no fixed times, aiming for weekly availability
Hours	Office hours as scheduled, see below

Lecture Schedule

For each week, deliverables consists of

- two pdf lecture note slide decks,
- (generally) two (or three) short videos
- a weekly overview video.

A list of lectures, generally two per week over a full (Spring) term, follows:

Week	Topics
1	Course Overview, RStudio, GitHub, General Setup
2	Shell Lectures I and II
3	Lecture on sed and awk; Markdown
4	Git Lectures I and II
5	SQL Lectures I an II
6	R Foundations; R Data Input/Output
7	R Data Wrangling; R Scripting
8	data.table; dplyr
9	Parallel R; Efficient R
10	Visualization I and II
11	Shiny; Guest Lecture (TBD)
12	R Packages Lectures I and II
13	GitHub Actions; Docker
14	No lectures – time for project

Weeks 12 and later will not be part of quizzes. This gives extra time for the optional (but *recommended*) project work.

Homework Schedule

Homework assignments (generally) cover the preceding four lectures, and are not cumulative. They prepare for the quiz (see next section) covering the same period, and permit students to do rigorous exercises which are graded electronically using PrairieLearn and PrairieTest.

Week	Given	Due
HW 1 (Shell, Markdown) – Week 3	Feb 1	Feb 7
HW 2 (Git,SQL) – Week 5	Feb 15	Feb 21
HW 3 (R Part I) – Week 7	Feb 27	Mar 4
HW 4 (R Part II) – Week 9	Mar 20	Mar 27
HW 5 (Visualization, Shiny) – Week 11	Apr 4	Apr 10

These are indicative dates which may be adjusted as needed.

Homeworks are generally released at noon, and due a week later at noon. Note that as the spring break, as well as 'busier' times at the CBTF site have to be accomodated, not all homeworks follow the Thursday to Thursday schedule. Graduate students receive (generally two) additional required questions. These questions are typically more substantial in nature and require extra effort than the regular questions for both undergraduate and graduate students. Undergraduates may opt to answer one or both of these questions for additional points, or challenges. Scoring is however capped at 100%.

PrairieTest – Computer-Based Testing Exams

The following dates have been (tentatively) reserved (but are as always subject to change):

Quiz	First Date	Last Date	Reserve	Weeks Covered
Q 1 (Shell, MD)	Feb 8	Feb 11	Jan 25	Weeks 2, 3
Q 2 (Git, SQL)	Feb 22	Feb 25	Feb 8	Weeks 4, 5
Q 3 (R Part I)	Mar 5	Mar 8	Feb 22	Weeks 6, 7
Q 4 (R Part II)	Mar 28	Mar 31	Feb 29	Weeks 8, 9
Q 5 (Vis, Shiny)	Apr 11	Apr 14	Mar 28	Weeks 10, 11

Quizzes follow the bi-weekly schedule of the homework, and cover the same (typically two week) set of lectures, and are also not cumulative. You can schedule your exam time starting the *Reserve* data (at 01:00h Central time per CBTF standards) via the PrairieTest site. Each exam will be a session of 50 minutes. These are *in-person* exams.

Under exceptional circumstance, accomodations may be made by course staff upon written request (also see email etiquette) *with proof of exceptional circumstances* to allow for online exams for fully-remote students not residing in Urbana-Champaign for the full length of term. Again, proof of such cirumstances will be required as this *must be* need-based and is not an elective choice for Urbana-Champaign based students who are expected to test at the CBTF facility in person. Requesting online testing when you were able to attend the CBTF in person may be treated an academic integrity violation with its full consequences.

Please consult the PrairieTest and CBTF site sites for full details.

Prerequisites

The course has no formal prerequisite.

Prior to taking this course, students should have:

- Taken one or more statistics courses for some general familiarity with data and analysis;
- Some general idea what basic descriptive statistics, probabilities and distributions, as well as linear regression are;
- Motivation for participation in an online class: readings, exercises, …
- Basic computer skills yet no formal programming background is *required* though it is surely helpful.

Online Access and Identification

The course is delivered primarily online and tested online. Students use Single-Sign-On with the University of Illinois 'netid' to access

- all lectures and videos stored on uofi.app.box.com
- access to R and RStudio on your personal computer, plus (likely) also on departmental server via SSO as additional computing resources
- GitHub via a U of I-administered instance also behind SSO
- GitHub via a University of Iillinois-administered SSO using GitHub 'cloud' resources
- CBTF and PrairieLearn to access homeworks and quizzes
- Canvas for grade and other course information

All material is linked from the course website.

In the past, CBTF Online quizzes used CBTF proctors for student identity verification. In the Fall 2022 term, we switched to PrairieTest for (on-line) exams/quizzes. The project requires a (recorded) presentation.

Office Hours

This course offers office hours from different members of the course staff that are held at throughout the week at pre-scheduled times.

GitHub Forum

For class discussion, we will use a GitHub repository and its issue system. This forum will be private and restricted to those in the course.

It is very important that each student

- registers a Github account (unless they already have one); since the Fall 2020 term we have been using a University of Illinois Single-Sign-On administered GitHub instance.

- let the instructors know about the Github id so that we can invite the student to the (private, controlled via Single-Sign-On with NetId) course discussion project

Email

Before you start writing an e-mail to a member of the *course staff* please make sure your question is *not*:

A list of Do Not Do This items.

- Already answered in this syllabus or course FAQ: the syllabus serves as the guiding document for the course.
- About exercises or homework: Questions should be asked via GitHub issues so that all students have access to the answer.
- A technical issue or code error: Try to *google* the error (*i.e.* copy and paste into Google). StackOverflow and similar forums can be helpful.

But please ensure your e-mails meet the following criteria:

A list of Do This items.

- The e-mail must be sent from an `@illinois.edu` account.
- The start of the subject line should contain the tag: `[STAT 447]`
- It should be followed by a space and a brief description.
- Good: '[STAT 447] Cannot load data file: error ...'
- Bad: '[STAT 447] Need help' or '[STAT 447] Code not working...'.
- Use the *course Staff E-Mail address*: `instructors@stat447.com` (or if you prefer `help@stat447.com`).
- Use professional tone as you would in other *written* communication.

We try our best to respond within 24 hours. Homework questions sent the same day homework is due will likely not receive a response before the homework is due. Plan accordingly.

Make sure the email does not contain homework code. The campus rules on academic integrity apply to all communication, including email.

Lastly, professional tone and written style matter, in email as in other *written* communication. Proper titles when addressing recipients is common style and recommended.

External Tutors

Please see the FAQ item on for hire tutors.

Assessments

Attendance

As an on-line course there is no attendance count. You are strongly encouraged to follow all the lecture slides and video, study the readings and possibly some or most of the extra readings. Most importantly, you need to *try* the examples and code we show, and experiment with it. As a proxy for class participation, we consider participation in the Github issue topic discussion which, for an online class, is the closest we have to class discussions.

Some online tasks may be offered for extra points. Examples are timely Github signup, or formation of a project team.

Homework

Homework assignments serve as a way to interact with the material outside of the classroom. Homework will be due at either *10:00 AM* on the *assigned due date*, which should generally be *Thursday*. We score the mean of the top four homeworks, *i.e.* with *the lowest homework score being dropped*. As this gives one automatic "out", late homework will generally *not* be accepted.

We score *best four out of five*.

In general, there will be no exceptions to this policy. Please start early, make sure your environment is working correctly, and that you are able to produce a working document. We have a teaching assistant with on-campus office hours, but in order to ask meaningful questions you need to *try* answering the material first.

That is the stated preference. In *truly exceptional* circumstances we also accommodate student requests.

Collaboration Policy

While working on *homework*, students are encouraged to study in groups. But students should strive to independently supply answers to the homework problems. As we use an automated platform, submissions can be compared easily. Academic integrity standards apply.

Copy code from on-line forums is fine provided it is *cited*. Any uncited code matched with available works online will be treated as plagiarism and may lead to an academic integrity investigation.

Distribution Policy

Each homework will be distributed via PrairieLearn where you are identified via your NetId so your submissions will be stored as combination of your NetId and the question.

Assignment Submission

Here are a few *do* and *don't* tips for the PrairieLearn web submission. Consider the following stanza from an actual homework:

```r
# Enter your code below: Do not alter the function signature:
# ensure it remains named 'iris_summary' and takes one argument.
# Ensure you return a data.frame as indicated in the question.

iris_summary <- function(irisdata) {

  # Enter code here

}
```

Consider the following recommendations carefully:

- *Do* follow the structure of the provided function.
- *Do* enter code where it says `# Enter code here`.
- *Do not* write code before the opening brace.
- *Do not* write code after the closing brace.
- *Do* use the supplied `irisdata` object. The function signature clearly states that that is the (only) input you need and are given.
- *Do not* load other data. You do not need `data(something)`. You *do not* need to load anything (unless specifically asked when a question is about data loading or saving).
- *Do* use the stated variable names: when the interface (or our instructions) say `irisdata`, do not deviate to `iris` or `iris_data` or any other form. *Do* write code to match the name exactly.
- *Do not* load other packages unless asked to do so. We generally expect you to use an explicitly named package, or just the functions already in R, *i.e.* what is called 'base R'.
- *Do* follow the instructions. When it asks to return a `data.frame` do not return a `matrix` or `data.table`. Return a `data.frame`.
- *Do* use the GitHub issue ticket linked to each question.
- *Do not* post code or (partial or complete) answers at GitHub.

Grading

Each homework assignment will be a variable number of points; however, each homework assignment will have equal weight towards your final grade. As stated above, we count *best four out of five*.

Quizzes

Instead of examinations, there will be to five weekly quizzes—see the section Schedule. The quizzes, just like the homework, will (generally) focus on the preceding (two weeks of) lectures and are (generally) not cumulative over the full course content.

And just like with the homework, you can drop *one* quiz grade over the course of the semester. We aim for five quizzes in total, and with *the lowest quiz score being dropped* the score will be the mean of the top four quiz scores.

The policies of the CBTF are the policies of this course, and academic integrity infractions related to the CBTF are infractions in this course.

If you have accommodations identified by the Division of Rehabilitation-Education Services (DRES) for exams, please take your Letter of Accommodation (LOA) to the CBTF proctors in person before you make your first quiz reservation. The proctors will advise you as to whether the CBTF provides your accommodations or whether you will need to make other arrangements with your instructor.

Any problem with testing in the CBTF *must* be reported to CBTF staff at the time the problem occurs. If you do not inform a proctor of a problem during the test then you *forfeit* all rights to redress.

Project

There are several components associated with the final project:

Also see the corresponding GitHub repository accessible to enrolled students.

Project Proposal The repository should contain an outline of what is planned, the sources of the data, possible transformation and possible modeling strategies and/or possible data visualizations. This can be provided via the README.md file of the repository.

Project Report The project report can be thought of as an (informal) *paper*. Guided by the format of an academic paper, it describes the projects in a succinct yet complete fashion along with references. Markdown should be used to write it, the result can be either in html or pdf format.

Project Presentation and Slides At the end of terms, a short recorded group video presentation, akin to a *lightning talk*, should introduce, present and summarize the work of the project in a form that is suitable for a general audience. A length of five minutes is a goal. The presentation should be supported by five to six slides, also produced in Markdown.

Evaluation of Peers, and Evaluations from Peers (if done as a Group Project) We require a short informal statement of *each* team member briefly stating who within in the team did (roughly) what percentage of the work.

The Project provides an *excellent* opportunity to “shine” and to demonstrate your passion, skill, and capabilities for *data science programming* work. It provides a great chance to make a mark to create something special and distinguished.

The group projects have to be finalized by noon (12:00h, Central) time on the due date which is Reading Day, May 2.

Exams

There are no midterm or final examinations in this course. Instead, we have homework, quizzes, and a group project.

Late or Missing Work

Late work will not be accepted for either homework or the group project.

Course Grades

Without a Project:

Type	Weight
Homework	One Half
Quizzes	One Half

With a Project:

Type	Weight
Homework	40%
Quizzes	40%
Project	25%

so we score out of 105% and the project permits to gain some extra credit.

Grading is discretionary, and performed by the instructor and the course assistant(s). There are no retakes; we mark 'best five out of six' for homework and quizzes so everybody gets to drop one each.

Grading Scale

Minimum Grade	Points
A- to A+	90 to 100
B- to B+	80 to 89.99
C- to C+	70 to 79.99
D- to D+	60 to 69.99

Minimum Grade	Points
F	below 60

Each ten point range is equally split over the three components (*i.e.* from minus to plus). Grades may be curved at the end of term before being finalized.

University Policies

Academic Integrity

The official University of Illinois policy related to academic integrity can be found in Article 1, Part 4 of the Student Code. Section 1-402 in particular outlines behavior which is considered an infraction of academic integrity. These sections of the Student Code will be upheld in the STAT 430 classroom. Any violations will be dealt with in a swift, fair and strict manner.

You may discuss methods for completing assignments with other students, but the execution of these methods and the preparation of the document must be done independently. Furthermore, there can be no discussion with other students or collaboration of any kind on exams. Sufficient evidence of sharing results, collaborating on written assignments, or simply relying on internet resources will generally result in:

- *First offense:* receiving an *undroppable zero* on the assignment and being written up for an academic integrity violation.
- *Second offense:* receiving an *F* in the course, an academic integrity violation, and recommendation for expulsion from the University.

If the evidence is indicative of a larger pattern, then the harshest penalty will be pursued.

Note that cheating includes both obtaining others' work, as well as distributing your own work.

- You may discuss the assignment with your classmates, but your final answers must be your own. Your final document should be created independently.
- To avoid any issues, *do note copy and paste code.* (With an exception for code provided for the course.)
- *Do not share RMarkdown or other submission files.*

If we detect academic integrity violations, we will contact you through the FAIR system.

In short, please do not cheat.

Support resources and supporting fellow students in distress

As members of the Illinois community, we each have a responsibility to express care and concern for one another. If you come across a classmate whose behavior concerns you, whether in regards to their well-being or yours, we encourage you to refer this behavior to the Student Assistance Center (333-0050) or online. Based upon your report, staff in the Student Assistance Center reaches out to students to make sure they have the support they need to be healthy and safe.

Further, we understand the impact that struggles with mental health can have on your experience at Illinois; significant stress, strained relationships, anxiety, excessive worry, alcohol/drug problems, a loss of motivation, or problems with eating and/or sleeping can all interfere with optimal academic performance. We encourage all students to reach out to talk with someone, and want to make sure you are aware that you can access mental health support at the Counseling Center or McKinley Health Center. For mental health emergencies, you can call 911 or walk-in to the Counseling Center, no appointment needed.

Accessibility

To obtain disability-related academic adjustments and/or auxiliary aids, students with disabilities must contact the course instructor and the Disability Resources and Educational Services (DRES) as soon as possible. To contact DRES, you may visit 1207 S. Oak St., Champaign, call 333-4603, e-mail `disability@illinois.edu` or go to the DRES website.

Disclaimer

The instructor reserves the right to make changes that are academically advisable. Such changes, if any, will be announced in class. Please note that it is your responsibility to attend the class and keep track of the proceedings.