Welcome & Syllabus

Lecture 01

Dr. Colin Rundel

Course Details

Course Team

Instrutor

- Dr. Colin Rundel

  - colin.rundel@duke.edu / cr173@duke.edu / rundel@gmail.com

  - Office hours: Wednesdays 2-3 pm, 204 Old Chem (or Zoom)

TAs

- Alonso Guerrero Castañeda

- Olivier Binette

Course website(s)

- GitHub pages - https://sta323-sp24.github.io

  - HTML, PDF, and qmds of Slides

  - Readings and other notes

- Canvas - https://canvas.duke.edu/courses/25196

  - Announcements

  - Gradebook

Course Timetable

- Lectures (weekly) – Old Chemistry 116
    - Wednesdays, 11:45 - 1:00 pm
    - Fridays, 11:45 - 1:00 pm
- Labs (weekly) – Reuben-Cooke Building 129
    - Tuesdays, 11:45 - 1:00 pm

Labs

- Attendance is expected

- Opportunity to work on course assignments with TA support

- Labs will begin in Week 2 (January 23rd)

Announcements

Will be posted on Canvas (Announcements tool)

- email notification

- also check Canvas regularly

Grading

This course is assessed 100% on your coursework (there is no exam). We will be assessing you based on the following assignments,

Assignment	Type	Value	n	Assigned
Homeworks	Team	50%	~6	~ Every other week
Midterms	Individual	40%	2	~ Week 6 and 14
Project	Team	10%	1	~ Week 10

Teams

- Team assignments

  - Roughly biweekly assignments

  - Open ended, ~5 - 15 hours of work

  - Peer evaluation after completion

- Expectations and roles:

  - Everyone is expected to contribute equal *effort*

  - Everyone is expected to understand *all* code turned in

  - Individual contribution evaluated by peer evaluation, commits, etc.

Collaboration policy

- Only work that is clearly assigned as team work should be completed collaboratively (Homeworks + Project).

- Individual assignments (Midterms) must be completed individually, you may not directly share or discuss answers / code with anyone other than the myself and the TAs.

- On Homeworks you should not directly share answers / code with other teams, however you are welcome to discuss the problems in general and ask for advice.

Sharing / reusing code policy

- We are aware that a huge volume of code is available on the web, and many tasks may have solutions posted.

- Unless explicitly stated otherwise, this course's policy is that you may make use of any online resources (e.g. Google, StackOverflow, etc.) but you must explicitly cite where you obtained any code you directly use or use as inspiration in your solution(s).

- Any recycled code that is discovered and is not explicitly cited will be treated as plagiarism, regardless of source.

- The same applies to the use of LLM like ChatGPT or GitHub Copilot - you are welcome to make use of these tools as the basis for your solutions but you must cite the tool when using it.

Academic integrity

To uphold the Duke Community Standard:

- I will not lie, cheat, or steal in my academic endeavors;

- I will conduct myself honorably in all my endeavors; and

- I will act if the Standard is compromised.

Course Tools

Accessing RStudio Workbench

To reduce friction, the preferred method is to use the department's RStudio server(s).

To access RStudio/Posit Workbench:

1. Navigate to https://rstudio.stat.duke.edu

2. Log-in with your Duke NetID and password.

If off campus, use the VPN to create a secure connection from your computer to Duke. If you are on campus, be

DSS RStudio alternatives

If you cannot access RStudio via the DSS servers:

- Make sure you are on authenticated Duke network (e.g. DukeBlue or VPN if off campus)

- Make sure you are not using a custom DNS server

  - e.g. `1.1.1.1` or `8.8.8.8`

- Use a Docker container from Duke OIT

  1. Go to https://cmgr.oit.duke.edu/ and login

  2. Select `Reserve a Container` and find `RStudio (statistics application with Rmarkdown and knitr support)`

  3. Click the link under my reservations to create your environment

Local R + RStudio

If working locally you should make sure that your environment meets the following requirements:

- latest R (4.3.2)

- latest RStudio (2023.12)

- working git installation

- ability to create ssh keys (for GitHub authentication)

- *All* R packages updated to their latest version from CRAN

Support policy for local installs - we will try to help you troubleshoot if we can but reserve the right to say go use the server.

GitHub

- We will be using an organization specifically to this course github.com/sta323-sp24

- All assignments will be distributed and collected via GitHub

- All of your work and your membership (enrollment) in the organization is private

- We will be distributing a survey this weekend to collection your GitHub account names

  - Before lab you will be invited to the course organization.

- All course related repositories will be created for you

Before next week

- Create a GitHub account if you don't have one

- Complete the course survey

- Make sure you can login in to the Department's RStudio server
  https://rstudio.stat.duke.edu

- Setup ssh key authentication with GitHub, see
  https://github.com/DukeStatSci/github_auth_guide

Account names

A few suggestions on picking a GitHub username:

- Incorporate your actual name! People like to know who they're dealing with. It makes your username easier for people to guess or remember.

- Pick a username you will be comfortable revealing to your future boss.

- Shorter is better than longer.

- Be as unique as possible in as few characters as possible. In some settings GitHub auto-completes or suggests usernames.

- Make it timeless. Don't highlight your current university, employer, or place of residence.

- Avoid words laden with special meaning in programming.

git and GitHub setup demo

Configure git

The following will tell git who you are, and other common configuration tasks.

```
1  usethis::use_git_config(
2    user.name = "Colin Rundel",
3    user.email = "rundel@gmail.com",
4    push.default = "simple"
5  )
```

You will need to do this configuration once on each machine in which you use git (e.g. the server and your laptop).

This can also be done via the terminal with,

All of these values will be saved in the `.gitconfig` file in your home directory

Configure git verification

To verify you configured git correctly, run

You should see output similar to above with your details.

Configure SSH and GitHub (authentication)

We will be authenticating with GitHub using SSH via public / private keys. We can create a new key pair (if necessary) by running the following in RStudio's *console*:

Testing things out

If you were successful with all of the above then you should be able to do the following,

- Create a new *private* repository on GitHub (e.g. `test_repo`)

- Add a `README` or similar file

- Clone the repository into RStudio

  - Using `File` > `New Project` > `Version Control` > `Git`

- Modify the `README` or other file

- Stage, commit, and push