Open Science and Reproducible Research in R

Lecture Series Feedback and Evaluation for 2023

Ernest Guevarra

2023-03-23

Contents

List of Figures

1 Introduction

This report provides my feedback, comments, and insight from a self-evaluation of the Open Science and Reproducible Research in R Lecture Series (https://oxford-ihtm.io/open-reproducible-science) as the lecture series organiser and facilitator. This report focuses on evaluating the lecture series based on its two main learning objectives:

1. Provide students with an introductory/foundational understanding and practical skills in using R (https://cran.r-project.org) for data management and analysis; and,

2. Provide students with an introductory/foundational understanding and practical skills in open science and reproducible research using R.

In addition, this report provides insight and perspective on the evolution of the lecture series since its inception and introduction to the MSc in International Health and Tropical Medicine course in 2022 and highlights impacts of changes implemented over the years in relation to the two main objectives stated above.

1.1 Background on changes in the lecture series over time

Following were the major changes implemented in this lecture series since its initiation in 2022.

- The lecture series is required for all students in the MSc for 2023 compared to it being a choice for students to participate during its implementation year.

- The lecture series has been expanded to 11 sessions compared to 6 sessions in 2022. This expansion allowed for the specific R learning components of the previous series to be made into their own sessions (5 sessions) at the start of the series to address the first main learning objective and the original 6 sessions followed thereafter to address the second main learning objective.

- Coding assignments were still delivered and managed through GitHub Classroom (https://classroom.github.com) using an education GitHub account via Oxford that provides features to create, manage, and grade/review coding assignments. The main difference now is that only one coding assignment was provided which the students worked on starting in session 2 up to session 8 (7 sessions) compared to 2 assignments which students covered in 2 session (about 1 session per assignment).

- R laboratory sessions outside of the 11 sessions were also provided as with previous year. However, all R laboratory sessions were one-on-one 30-minute sessions all throughout compared to initial group sessions (up to 3 students for 30 minutes) throughout the duration of the lecture series in the inaugural year and shifted to one-on-one 30 minute sesssions after the lecture series was completed. R laboratory sessions are sessions that the students can book if they felt they needed more informtion/discussion regarding a topic presented in class. These sessions have also been used by students who have missed a class session and wanted to catch up with the topic that was discussed then.

- A handbook was created specifically for this year's lecture series which was accessible both as an online handbook (see https://oxford-ithm/ihtm-handbook) and as a downloadable PDF (https://oxford-ihtm.io/ihtm-handbook/ihtm

-handbook/ihtm-handbook.pdf) or as an Ebook (https://oxford-ihtm.io/ihtm-handbook/ihtm-handbook.epub). The aim of the handbook is to serve as a reference for students on basic and overview types of topics that may help them currently or in the future when trying to apply the skills they've learned from this lecture series.

- A three-session (one month period) hackathon (https://github.com/OxfordIHTM/ihtm-hackathon-2023) which was conducted in the period of the last three sessions of the lecture series. The hackathon was designed more as a problem-based learning exercise to culminate the entire lecture-series. The hackathon offered a challenge that would require the students to put together all the R coding and open science/reproducible research concpets and skills they have learned and/or are learning so far to address a real-life problem/topic with actual research data. Within the three face-to-face sessions, topics relevant to the problem were presented and discussed whilst the students were working through their problem sets. This hackathon was a unique feature of this year's edition of the lecture series.

- A discussion board (https://github.com/orgs/OxfordIHTM/discussions) specific to the lecture series was initiated. The discussion board was primarily a forum to address and discuss very specific questions and/or topics that the students may have. It was also a means for the lecture series presenter/facilitator to share solutions/answers to the coding assignments and exercises. Whilst the discussion board was also there in the inaugural year, this year's iteration had a much clearer usage and purpose and was actively used.

- Asynchronous code review process using git and GitHub functionalities. Whilst these functionalities were already

available and usable in the inaugural year, this year saw the full use of this feature.

2 Evaluation approach

In this evaluation, I use two sources of information to guide my evaluation of the lecture series. First, I utilise the available user data via the lecture series' GitHub (https://github.com/OxfordIHTM) platform to gain insight into how the students engaged and interacted with the topics, exercises, and activities. Specifically, I used the **commit** and **push**[1], **pull**[2], and **merge**[3] metrics of the whole class to characterise as objectively as possible the students' level of engagement with the topics discussed while at the same time provides an assessment of the students' ability to demonstrate and apply the practical skills presented in class.

2.1 The git commit and push, pull request, and merge process metrics

A `git commit` is a fundamental operation in the Git version control system, which is used to save changes made to a repository. When you make changes to files in a Git repository, you can create a commit to record those changes. A git commit is essentially a snapshot of the state of the repository at a particular point in time. It includes a unique identifier (called a "hash") that identifies the commit, as well as metadata such as the author, date, and commit

[1] See https://github.com/git-guides/git-commit and https://github.com/git-guides/git-push

[2] See https://github.com/git-guides/git-pull

[3] See https://www.git-tower.com/learn/git/commands/git-merge

message. The commit message is a brief description of the changes made in that commit, which helps other developers understand the purpose of the commit. When a `git commit` is created, a permanent record of one's changes are recorded, which can be used to track the history of the repository and collaborate with other developers. `git push`, on the other hand, uploads the commits made in the local repository to a remote repository, such as GitHub, GitLab, or Bitbucket. When you push changes to a remote repository, git compares the differences between your local repository and the remote repository, and then sends the new changes to the remote repository. This allows other developers working on the same project to see and access your changes. By this characteristic, students' `git commit and push` is a good record of students' work during the lecture series and a good baseline indicator of their ability to write code and script (regardless of whether this code is syntactically and operationally correct).

A `git pull request` is a feature that allows developers (in this case students) to propose changes to a codebase hosted on a git repository. It enables developers/students to share their changes with other members of the team and solicit feedback or approval before merging their changes into the main branch. When a developer/student makes changes to a branch in their local repository, they can create a pull request to request that those changes be pulled into another branch or repository. The pull request shows the differences between the two branches, and allows other developers/students or myself (the facilitator) to review the changes and leave comments or suggestions for improvement.

In the case for the lecture series, once the pull request is submitted, it is reviewed by myself, commenting on the changes or requesting further modifications. The pull request was also used to trigger automated tests to ensure that the proposed changes do not introduce any errors or regressions. The `git pull request` is a good indicator of a students' confidence in their code and their

ability to check their own code locally which is a one-step higher order of knowledge and understanding of both R coding and use of git as a tool for reproducible science.

Finally, a `git merge` takes place once I approve the student's code in their `pull request`. The `pull request` can be merged into the target branch, allowing the changes to become part of the codebase. If not, the developer can make further modifications and submit a new pull request. `git merge` of a pull request is the highest level of accomplishment for the students as this means they were able to create R code that were syntactically correct while at the same time R code that actually outputs/produces the intended/required output. A merged pull request also means that students were able to engage with the review process demonstrating their ability to use git as a tool for reproducible science.

2.2 The manual code review and subjective assessment process

The **commit**, **push**, and **merge** metric were then corroborated/triangulated with my own qualitative and subjective observations during my interactions with the students during the lecture sessions, R laboratory sessions, and other interactions via the GitHub review process and discussion board.

I deemed this quantitative and qualitative/objective and subjective approach to this evaluation the most appropriate approach to assess the performance of the lecture series in reaching its learning objectives and reflects much closely and realistically what has transpired throughout the lecture series process.

3 Evaluation results

3.1 Commits

Table 3.1 presents the overall `commit` metrics for IHTM class 2022 and 2023.

Table 3.1: git commit metrics for class 2022 and class 2023

IHTM Class	No. of students	No. of commits	Commits rate
class-2022	16	111	6.94
class-2023	23	728	31.65

This year, the students made a total of **728** commits giving a commit rate of **31.65 commits per student** compared to **6.94 commits per student** last year. Commits rate is a low-level indicator of student engagement to the lecture series topics. Specifically, it would show that 1) students are writing R code as specified in the assignments and exercise provided to them; and that 2) students are able to perform the basic skills of a *git commit and push* which were the first topics to be taught the class in both years.

Figure 3.1 shows commits rate trend over time for class 2022 and class 2023.

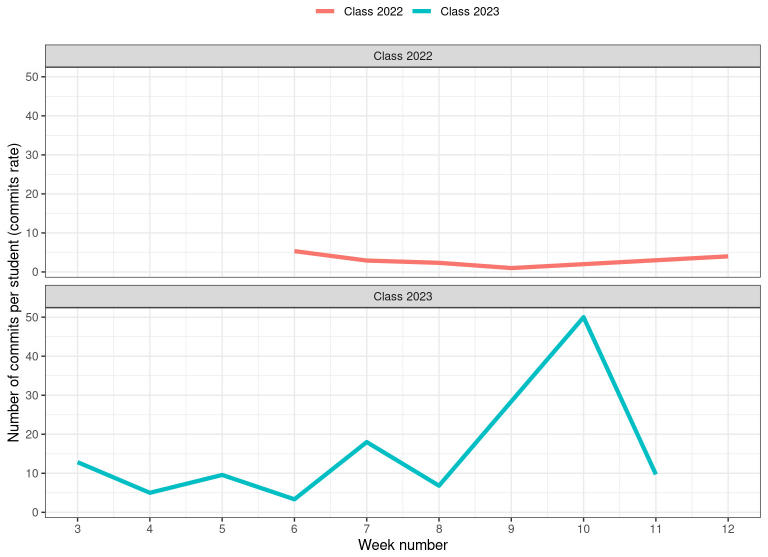

Figure 3.1: Commits rate per week by IHTM class

A similar trend of initial high commits rate in the starting week of the lecture series followed by a dip in subsequent weeks and then a ramping up of commits in latter weeks can be noted for both class of 2022 and 2023. However, There is a clear significantly higher volume of commits per student throughout the whole lecture series period for the class of 2023 as compared to the class of 2022.

The difference in number of sessions (6 for year 2022 and 11 for year 2023) would not explain this difference in commits rate as the rate was calculated week on week. Capacity of students in both years were also similar with all of those in attendance not having any experience of R prior to the lecture series and less than a handful having had experience with other relevant statistical software such as STATA or SPSS. Also, the assignment provided to each class was roughly the same[4].

[4]One assignment given to class 2022 was the same as what was given to class 2023. The second assignment given to class 2022 was not given to class 2023. Instead, class 2023 had a hackathon for the last 3 sessions. However, the second assignment given to class 2022 was issued in addition to the first assigment and the students were expected to work on both assignment simultaneously during the middle 2 sessions in 2022.

The possible explanations for the difference in commits rate are two factors:

1. The lecture series for 2023 started much earlier in the year (in January 2023) when students may have been more fresh and relaxed, and less busy with course work coming off the Christmas holiday compared to mid-February in 2022 when students well into the new term and with a lot more competing priorities for the students; and,

2. The re-organisation of the topics in 2023 wherein focused R sessions were taught starting on the second up to the fifth session before adding in topics on open science and reproducibility for the remaining sessions compared to a more mixed R and open science and reproducibility sessions throughout the 6 sessions in 2022.

3.2 Pull requests

Table 3.2 shows the overall `pull request` metrics for IHTM class 2022 and 2023.

Table 3.2: git pull request metrics for class 2022 and class 2023

IHTM Class	No. of pull requests	Commits rate	Pull request rate
class-2022	5	6.94	1.39
class-2023	33	31.65	0.96

This year, the students made a total of **33** pull requests giving a pull request rate[5] of **0.96 commits per student per pull request** compared to **1.39 commits per student per pull request** last year. This indicates that class 2023 students are requiring lesser commits per student

[5]Pull request rate is the number of commits per person per pull request. $ ~ = ~ $

before they are ready to issue a pull request compared to those in class 2022. This indicates that students of class 2023 are getting their code syntax right much earlier.

Pull request rate is an indicator of both student engagement to the lecture series topics and student confidence in having their code work assessed and reviewed by the facilitator. In addition to the competencies that a student would have demonstrated for making a commit, a student who then issues a pull request demonstrates 1) their ability to self-evaluate their code work and either correct their own work accordingly are appropriately ask for help until any errors are resolved; and, 2) their ability to appropriately issue a pull request in order to receive formal code feedback and review from the facilitator. These are higher order competencies.

Figure 3.2 shows pull request rate trend over time for class 2022 and class 2023.

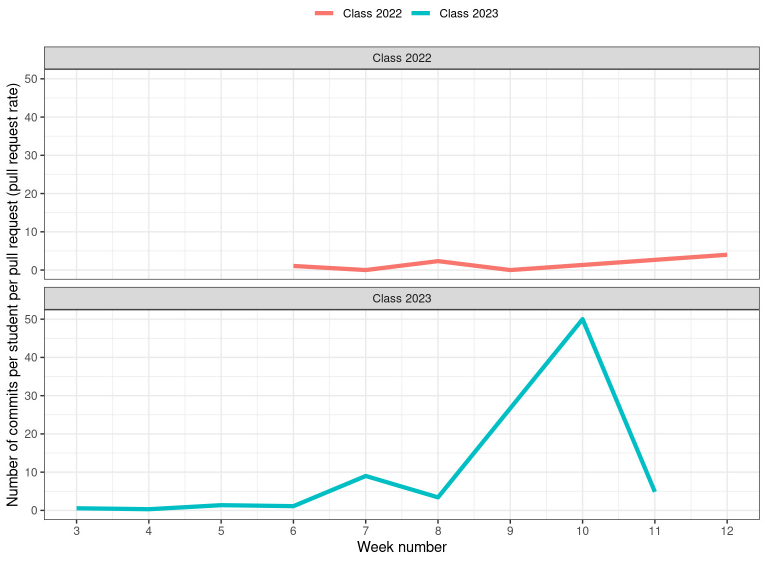

Figure 3.2: Pull request rate per week by IHTM class

Weekly trend of pull request rate are similar for both years with low commits per student per pull request at the start of the lecture series which progressively gets higher commits per student per pull request in the latters sessions. However, for class 2023, the last week of the lecture series saw a massive drop in the commits per person per pull request rate (though not near to their rate in the early sessions).

For this indicator, it is possible that the lengthier period of the lecture series in 2023 has had an impact such that students were still commiting and pushing their code but are not quite getting to a stage where they are able to get their code to work or answer the problem provide. Competing course work priorities may be prohibiting students to focus on their code work. However, the more likely explanation is the type of assignment given to the students in class of 2023 in the latter part of the lecture series. The highest commits per person per pull request rate was in the last 3 sessions which is when the hackathon was initiated. Compared to the previous assignment, the hackathon topic involved a much larger dataset on a topic that most of the students were less familiar with (nutrition). In addition, the hackathon mechanics provided the students with just enough information to begin with and were expected to assess what information they don't know and ask for these accordingly (as part of a problem-based learning approach). In the hackathon, they were being challenged both in terms of their code work but also with how they tackle a real-life public health data scenario while learning how to work as a team (compared to previous assignment which they did on their own). Despite this, the high commits per student per pull request during this hackathon period meant that students kept on working on the task at hand that was more challenging than their earlier assignment.

3.3 Merge

Table 3.3 shows the overall `merge` metrics for IHTM class 2022 and 2023.

Table 3.3: git pull request merge metrics for class 2022 and class 2023

IHTM Class	No. of pull requests	No. of pull requests merged	Pull request merge rate
class-2022	5	4	1.25
class-2023	33	19	1.74

This year, the students made a total of **19** pull requests were merged giving a pull request merge rate[6] of **1.74 pull requests per pull request merged** compared to **1.25 pull requests per pull request merged** last year. This indicates that class 2023 students are able to merge more pull requests compared to those in class 2022.

Pull request merge rate is an indicator of student engagement to the lecture series topics, student confidence in having their code work assessed and reviewed by the facilitator, and most importantly the quality and effectiveness of their code. In addition to the competencies that a student would have demonstrated for making a commit and issuing a pull request, a student whose code gets approved through the pull request review process demonstrates 1) their ability to write coherent, syntactically correct, and operationally accurate code; and, 2) their ability to navigate the

[6] Pull request merge rate is the number of pull requests per pull request merged. $ ~ = ~ $

strict and highly detailed code feedback and review process. These are highest order competencies and indicate a good foundational understanding of R programming, and concurrently a good foundational understanding of open science and reproducible research.

Figure 3.3 shows pull request merge rate trend over time for class 2022 and class 2023.

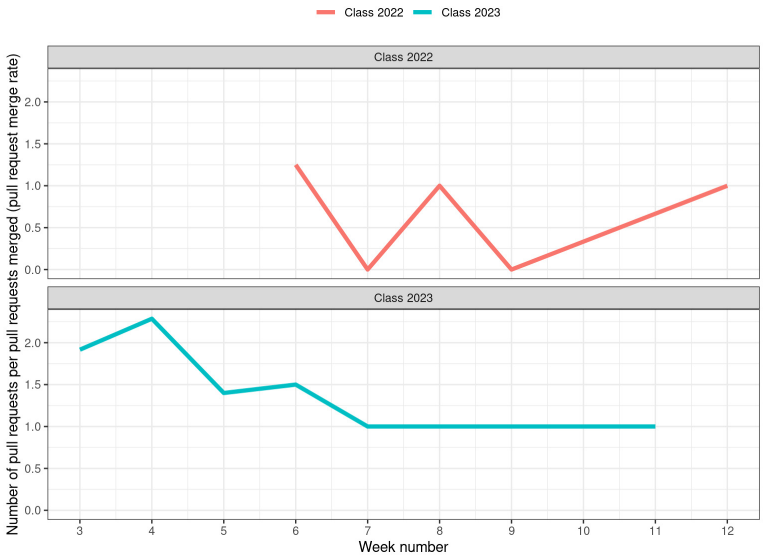

Figure 3.3: Pull request merge rate per week by IHTM class

The weekly trend of pull request merge rate for class 2022 shows alternating peaks and troughs from week to week with lowest points of no merge pull requests happening and values ranging from between 1 to 1.25 pull requests merged per week. For class 2023, the picture is more of a early peak of between 2 to 2.25 then slowly dipping eventually plateauing at 1 merged pull request per week until the end of the lecture series. This would indicate a more sustained achievement of approved code work week-on-week for the year 2023

compared to the year before with the dip in the latter sessions explained by the more challenging second activity/task given to the students.

3.4 Observations

Following were my observations based on the code review I performed on both commits/push and pull requests made by the students:

- There was much richer interactions between myself and the individual students in the review process in 2023 mainly due to the large volume of commits and pushes made by the students and by the students having been oriented more effectively on the use of the pull request for this purpose. A lot of additional learning happened within the notes of the pull request paricularly in 2023 and students were equipped to use the process to further their learning. The extra sessions were the main reason this was possible;

- It took about 3 sessions to get most of the students properly setup with the tools we used for the 2023 sessions. In 2022, we would have probably needed about the same time but given the fewer sessions, we had to prioritise the theoretical teaching even if a good number of the students were still not fully setup and that the topics on use of these tools were covered very quickly in half a session. Whilst the R laboratory sessions in 2022 made up for this limitation, not everyone had the time to participate in these sessions which were held outside of course work hours/period. Comparatively however, more students in 2022 took advantage of the R laboratory sessions compared to the students in 2023.

- Timing of the first 5 sessions of the 2023 lecture series was crucial as those happened in straight days early in January 2023 and covered mainly R coding topics and skills training. This was also at at time when the students were still quite energetic and had a bit more focus as other course work priorities were not as urgent yet. The metrics described above seem to support this observation.

- The time in which the sessions were scheduled posed a challenge. The early start was difficult for the students and the first few sessions when foundational learning were the main topics, it was a choice of starting even though some students are not yet around or waiting until most are already present. In my opinion, the outcomes of these two choices ended up being the same - that we lost at least 10-15 minutes in the first 5 sessions. Even if we didn't wait, those students who have arrived late would eventually need a bit more extra time and attention for the practical components of the session as they missed the introductory elements brought up at the start.

- From my engagement with the students, a good number (at least half) of the students commented that they believed these to be important skills that they can use for their thesis/placement project but more importantly beyond the MSc. These students were also the ones who took the time to participate in R laboratory sessions. This is likely reflected in the still low levels of pull requests and merges completed in 2023 (thought it has significantly increased from previous year).

- On the other hand, there were a good number of students who were interested in the sessions and participated as much as they can but were also seeing this as just something they needed to attend rather than something that they will draw

on in the future. It would be good to know from these types of students whether there was any learning beyond the coding that they felt translated into skills and knowledge that will apply to things that they were interested in. Some of those that I've talked with about this mentioned how the topics felt so strange and different to what they have been learning, almost similar to learning a new language for the first time and as such would require them more than the time allocated to really learn the top[ics properly and get to appreciate how the topics can be useful to them.