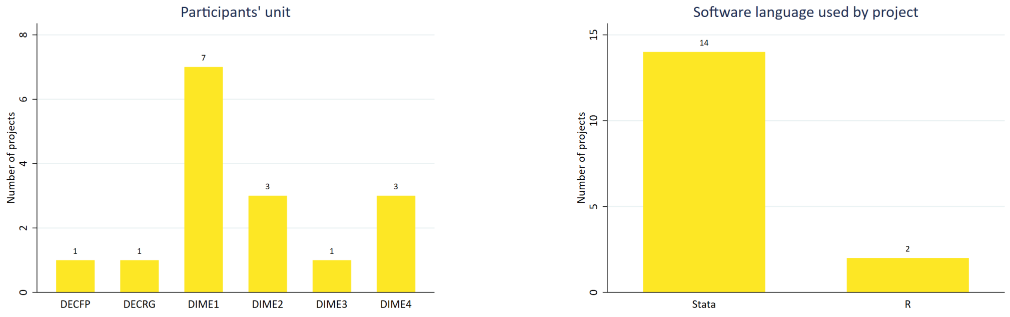
Peer Code Review Summary - FY23 Q3

DIME Analytics

A total of **16 research assistants** joined the peer code review held in the week of May 22nd, 2023, and reviewed code from **16 different projects**. DIME1 was the most-represented unit, followed by DIME2 and DIME4. Most projects used Stata as the main coding language.



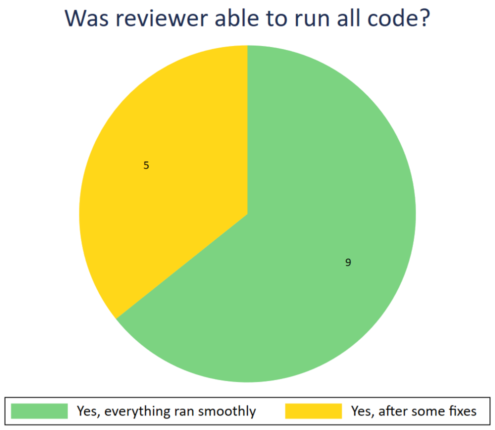
(a) Participant Units

(b) Participant Software Used

Reproducibility

Out of the 16 code packages exchanged, **14 included de-identified data, and were evaluated for reproducibility.**

Encouragingly, each of these code packages were reproducible: the code file could be run by the reviewer with either minor fixes, or no changes at all!

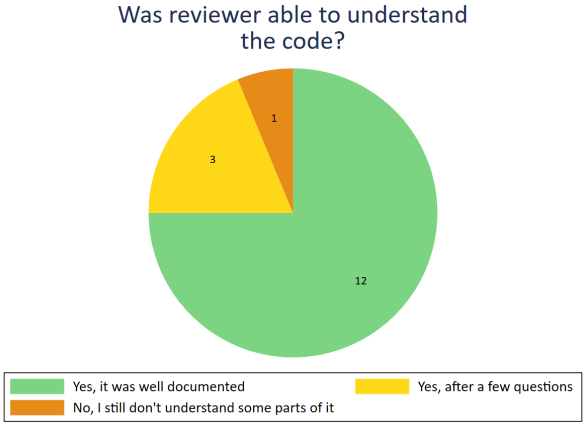


Ease of Use

75% reviewers said that the code they received was **easy to understand and well-documented**.

The remaining 25% indicated that more comments in the code and additional details in the GitHub README would be helpful.

In terms of **transferability**, 63% reviewers said the material provided would be enough to take over the project without any additional communication with the original coder.



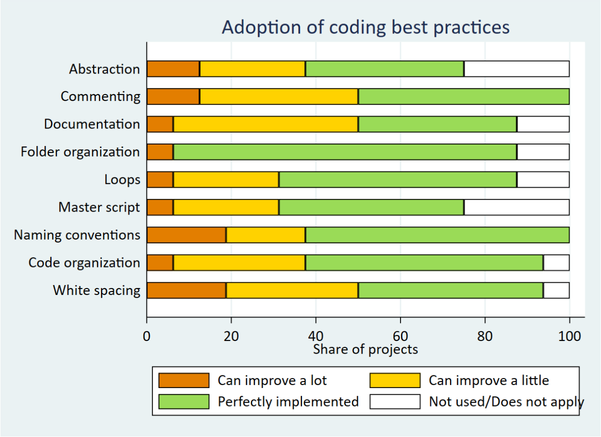
Moreover, 68% reviewers said that it would take them **1 day or less** to be able to understand the code well enough to make contributions to it. The code for 75% projects was rated **easy to maintain**. There was **only 4 projects** for which making adaptations would require changes in multiple places, making it hard to build on existing code.

Adoption of Coding Best Practices

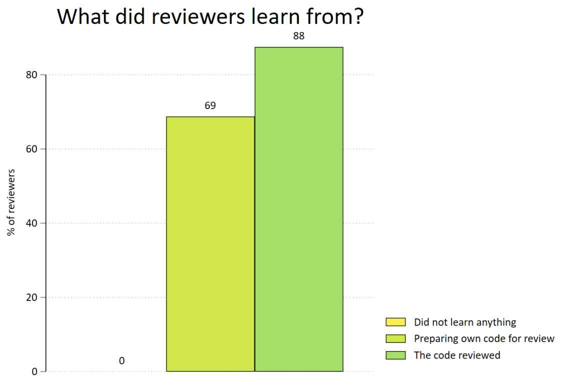
The **mean number of best practices adopted was a very healthy 8.0** - out of 9 in total. Further, 68% of projects **correctly implemented** each of the best practices.

The reviewers identified the **most room for improvement** in use of master scripts, and abstraction - that is, use of special commands to simplify repetitive tasks.

The **most widely adopted best practices**, include proper naming conventions for files, and well-commented code.



Feedback and Challenges



All participants reported learning something new from the code review exercise. 14 reviewers said they learned from the code reviewed, 11 reviewers reported learning from preparing their own code for submission, and 9 reviewers reported learning from both. In addition, **participants also reported learning** the following:

- New ways to visualize data
- Better systems of organizing code
- New commands and functions to perform tasks more efficiently
- Use of master scripts to define settings

Finally, the **primary challenges** identified during the exercise include **time constraints** to work on the code review, and **lack of clear communication** with partners who were not able to join the work review session. Participants acknowledged the extended flexibility in submitting feedback based on previous rounds. For future rounds of code review, we aim to enforce stricter attendance requirements from participants, with greater involvement of TTLs.

Participant Comments

"We were provided enough flexibility on the feedback submission to be able organize our workload between the code review and our projects."

"Through this process, I learned about my own code, how to ensure best practices, organization tips in code, and new ways to make the code cleaner."