Collaborative Code Review

Overview

Many projects led by academic researchers are supported by a lead statistical analyst. The majority of the research projects are reliant on analysts for their own quality check, and the practice of self-review varies across individual analysts and is impacted by factors such as project budget and timelines. Not all academic research teams require a formal code review process involving an independent analyst.

Objectives for Code Review Process

- Improve research replicability and transparency
- Identify errors or imprecision related to data and statistical programming before publishing or reporting to important stakeholders
- Promote coding skill and improve style and efficiency through collaborative development

Proposal

The collaborative code review process is comprised of several components:

- The research lead will decide if collaborative code review will be needed for a project. The research lead can deliberate on one of the following options:
    - Independent parallel coding by another analyst to fully replicate results
    - Collaborative code review process (described in this proposal)
    - No code review (criteria are still needed to gain confidence in work quality, such as clearly documented data analysis plan, code report and data report)
- The collaborative code review process generally takes place toward the end of the project
- Standard code review process
    - Include the reviewing analyst to IRB application if PHI data will be reviewed
    - Determine the time allocated to code review, ranging from 1 to 3 workdays (or longer if needed) depending on complexity of the project
    - Select a review analyst based on considerations such as bandwidth, skill-level (e.g, programming languages used), prior project-related experiences, etc.
    - Encourage the lead analyst to share related study documentation and the data analysis plan before code review starts
    - The reviewing analyst will start going through the documents independently to form initial impressions, opinions, and questions.
    - Plan collaborative code review over several sessions between the analysts to spell out the work as it takes time to fully digest other coder's workflow; Line-by-line review is recommended, partitioned into multiple sessions. We encourage in-person walkthrough for these code review sessions featuring real-time conversations
    - Reviewing analysts are expected to raise questions; seek clarification; critically review coding logic and organization, choice of procedures and options, and program efficiency; or provide suggestions on program optimization, etc.

    - The analysts are encouraged to also review program logs and program outputs thoroughly (data management best practices call for documenting program logs)

- Other considerations:
    - This optional code review process does not involve independent verification or replication of results. The focus is on checking coding quality and accuracy, verifying working assumptions, and identifying/improving logic flow, although more intensive review aimed at replication can be planned if deemed appropriate.
    - Acknowledge the contribution by the reviewing analyst in publication or deliverables if possible (and a good thing to show!)
    - Research team can prioritize naming a reviewing analyst within the overall research group led by the same PI. If no eligible candidates can be found within group, the Sr. Data Manager can assist with identifying an analyst from other research groups
    - Each project analyst will receive credit by contributing to code review for peer analysts. Similarly, the analyst can receive services by others reciprocally
    - The plan is that over time, most analysts will equally benefit from having their work reviewed, while also contributing to other people's work. The Sr. Data Manager will be facilitating analyst matching and effort tracking. This should alleviate needs for explicitly budgeting for code review effort.

Additional Benefits

- Anticipating code review by an independent analyst will encourage the lead study analyst to adopt best practices in programming, such as preparing robust data analysis plans, commenting and annotating their codes, reducing proliferation in codes, and improving programming efficiency and code readability
- Collaborative code review benefits analysts through promoting colleagueship and accelerating professional skills development

Next Steps

Please reach out to the Sr. Data Manager to discuss your project needs.