

A Data Scientist's Guide to Code Reviews

PyCon/PyData 2022



Doing code reviews

Pros

- They'll improve the code clarity
- They might uncover errors
- I'll probably learn something while doing it
- Literally everyone says that I should

Cons

- I don't want to

⇒ I will ****not**** do code reviews

Doing code reviews on data science work

Pros

- They'll improve the code clarity
- They might uncover errors
- I'll probably learn something while doing it

Cons

- Hardly anyone says that I should
- The code will likely not end up in a live system as-is
- The work is a one-off thing

⇒ I will ****not**** do code reviews

However, having someone else review your work is as important in data science as in software engineering.

What are code reviews for?

- Verifying that the specified goal is achieved
- Uncovering errors and misunderstandings
- Knowledge transfer
- Feedback for architectural or design decisions
- Improving your code & coding practice

The traditional code review practice is not applicable to “typical” data science work.

Different focus

Software Engineering

- Is the artifact functional?
- Are there bugs?
- Are coding guidelines & quality standards met?
- Can someone else than the author work on the artifact?


⇒ Code Review

Data Science

- Is the chosen approach comprehensible & clear?
- Have data peculiarities been taken into account?
- Are the results plausible?
- Can someone else than the author explain the concept?

⇒ Peer Review

What are code reviews for in data science?

- Verifying that the specified goal is achieved 
- Uncovering errors and misunderstandings
- Knowledge transfer
- Feedback for architectural or design decisions
- Improving your code & coding practice

What are code reviews for in data science?

- Verifying that the specified goal is achieved ✓
- Uncovering **logical** errors and misunderstandings ✓
- Knowledge transfer ✓
- Feedback for architectural or design decisions
- Improving your code & coding practice

What are code reviews for in data science?

- Verifying that the specified goal is achieved ✓
- Uncovering **logical** errors and misunderstandings ✓
- Knowledge transfer ✓
- Feedback for ~~architectural or design decisions~~ **approach** ✓
- Improving your code & coding practice

What are code reviews for in data science?

- Verifying that the specified goal is achieved ✓
- Uncovering **logical** errors and misunderstandings ✓
- Knowledge transfer ✓
- Feedback for ~~architectural or design decisions~~ **approach** ✓
- ~~Improving your code & coding practice~~ **Reproducibility** ↻

Code review checklist

- Overview over present files and the task
 - changelist
 - MR's description
 - accompanying ticket (when working with a ticket system, e.g. JIRA)
- Run the code and reproduce the results
 - [optional] if GitLab CI is used it might be worth checking the pipeline
 - ! fixing the pipeline is the author's responsibility
- Ensure comprehension: ask, ask, ask

Why has the author decided to do XY, chosen package A instead of B, selected model 42 as baseline,...?

github.com/awoerner92/talks

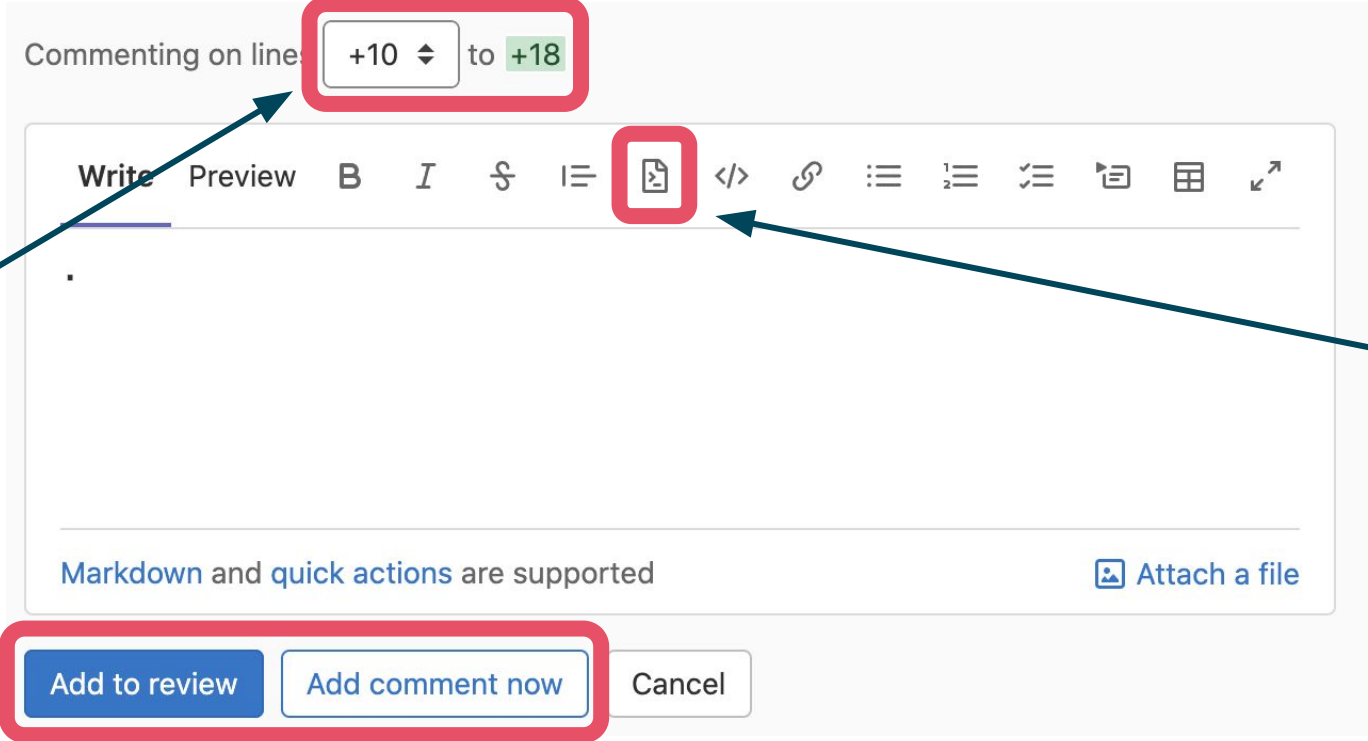
How to do code reviews?

- How to Do Code Reviews Like a Human
 - <https://mtlynch.io/human-code-reviews-1/>
 - <https://mtlynch.io/human-code-reviews-2/>
- How to Make Your Code Reviewer Fall in Love With You
 - <https://mtlynch.io/code-review-love/>

Useful Git functionality: pre-commit hooks

- Runs pre-defined set of tools with every commit
- Tools:
 - Jupyter notebook conversion: nbconvert
 - Code formatter: black, isort
 - Linter: flake8

Useful GitLab functionalities: Comment field



The screenshot shows the GitLab comment field interface. At the top, it says "Commenting on line" followed by a range selector showing "+10" and "+18" (both highlighted with red boxes). Below this is a rich text editor with a toolbar containing icons for bold, italic, link, code, and other formatting options. The code icon (a document with a code symbol) is highlighted with a red box. At the bottom of the editor, there are three buttons: "Add to review", "Add comment now", and "Cancel". The "Add to review" and "Add comment now" buttons are highlighted with a red box. Arrows point from text labels to these elements: "comment & mark multiple lines" points to the line range selector; "code suggestion" points to the code icon; and "add comment to batch or comment immediately" points to the "Add to review" and "Add comment now" buttons.

comment & mark multiple lines

code suggestion

(GitHub: ```` suggestion ````)

add comment to batch or comment immediately

Useful GitLab functionalities: Mark viewed

The screenshot shows the GitLab interface for a repository. At the top, there are navigation tabs: Overview (55), Commits (19), Pipelines (14), and Changes (43). A status bar indicates 'All threads resolved'. Below this, there are controls for comparing versions (version 1 and latest version) and a 'Show latest version' button. The file list shows 39 files with a total of +939 and -909 changes. A file named 'tests/test_model.py' is highlighted, showing a change of +252 and -0. A red box highlights the 'Viewed' checkbox, which is currently checked.

- Collapses the file
- Helps to keep an overview

Thank you!



Alexandra Wörner

Data Scientist

alexandra.woerner@scieneers.de



@alex_woerner

Questions?