Continuous Integration

CI for research & data science

What is Continuous Integration (CI)?

A software development practice to **frequently** integrate changes to a project into a remote version

- Help find problems and conflicts early
- Keep all team members up to date

Each integration is put through an automated build process (e.g. tests)

Ensure integration takes place, and does so frequently

Continuous...

- ... integration: combine work from individuals into main, shared version
- ... delivery: run steps required to build and test the project
- deployment: run the actual code and build its outputs

The CI process

Each time a commit is pushed to a repository...

- 1. Clone a copy of the project
- 2. Recreate its computational environment on a new (virtual) machine
- 3. Build the project within the new environment
- 4. Run tests, deploy project, etc.
- 5. Report the results

CI providers

Hosted (free for open source)

- Travis CI
- GitLab CI/CD
- GitHub Actions
- + easy to use
- less flexible

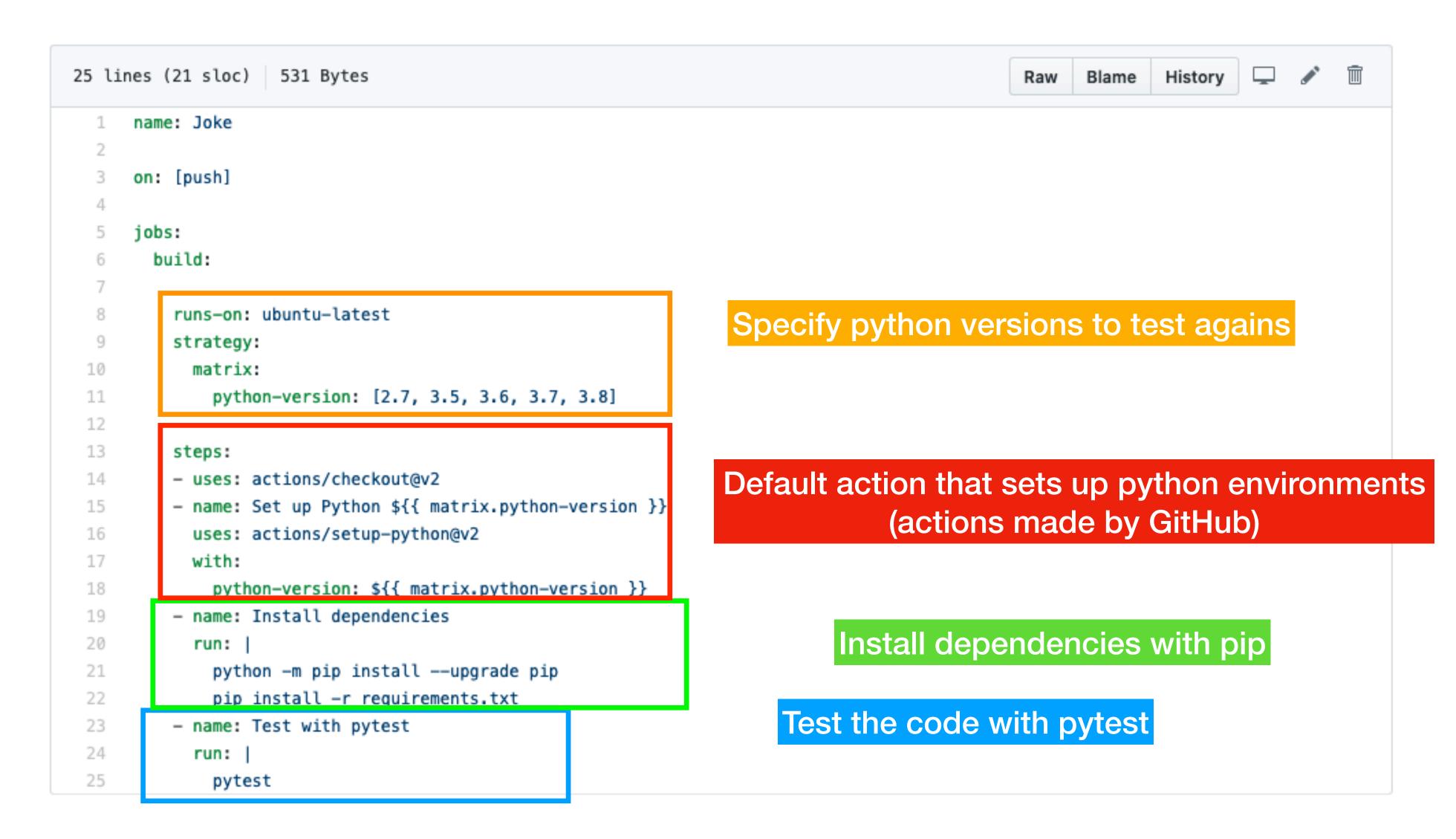
DIY

Jenkins

- configure and host yourself
- + more powerful

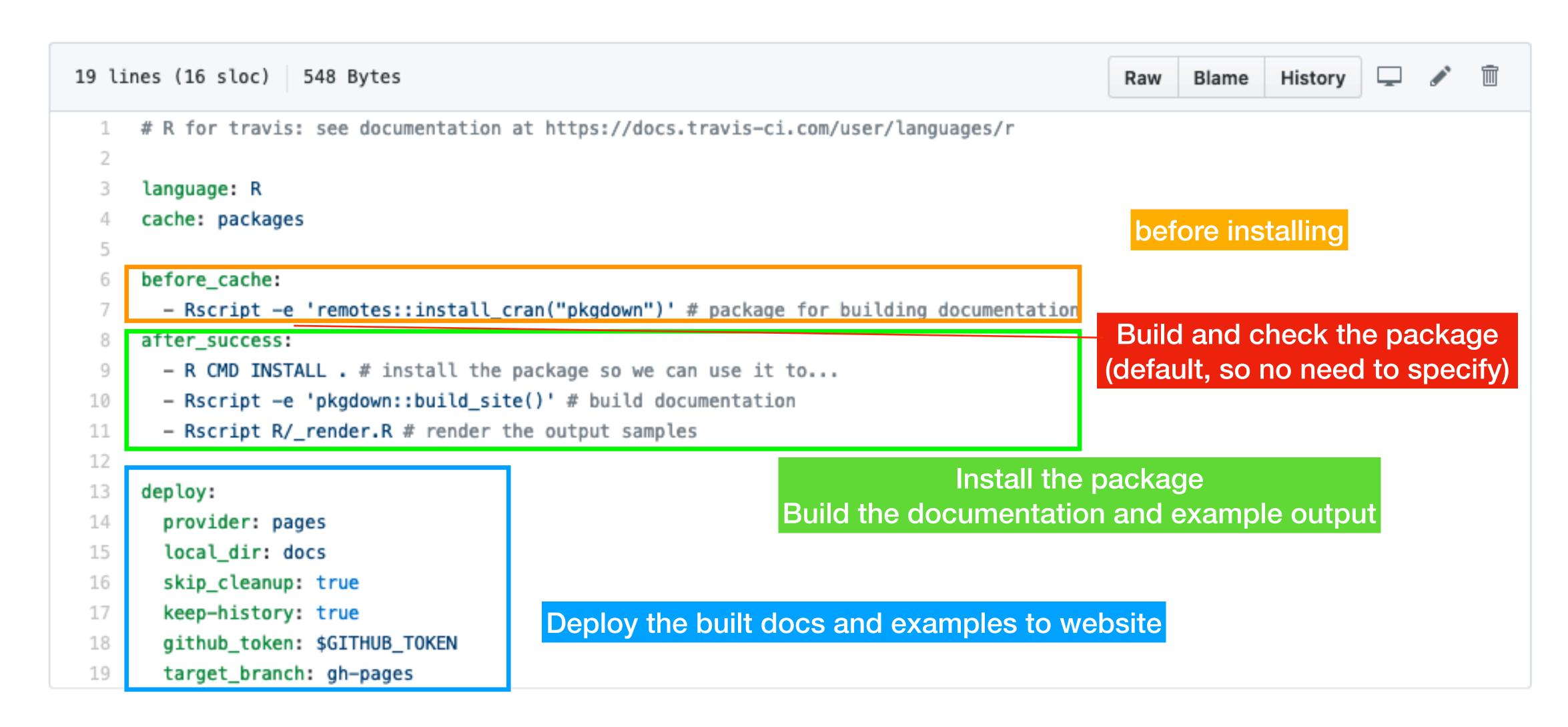
A toy example

With Github Actions: https://github.com/lcreteig/ci-python-example/blob/master/.github/workflows/test.yml



A (more) real-world example

With Travis: https://github.com/lcreteig/amsterdown/blob/master/.travis.yml



Applications?

- Running unit tests for your code
- Checking whether your code runs on a machine other than yours
- Building the output of your research: an analysis report, a paper, an ML model, a (Shiny/Dash) app, etc.

Limitations

- CI service needs access to your code/data
- Deployment can take a long time for complex products (huge dataset)

References

- Chapter 14 of "The Turing Way: A Handbook for Reproducible Data Science": https://the-turing-way.netlify.app
- https://help.github.com/en/actions/language-and-framework-guides/githubactions-for-python
- https://docs.travis-ci.com/user/languages/r/