Reproducibility

Frank Aragona

2023-04-09

Objectives

- Data and Code Democratization
- Github Codespaces
- Package reproducibility with virtual environments
- Github Releases
- Documentation

Data and Code Democratization

Data and code in our repositories need to be accessible to end users and developers. There should be no bottlenecks or difficulties with installing software, executing code, finding documentation, and using test datasets.

The goal is for any user to run code without needing to install anything on their personal machine and run your code with minimal set up. This may not be possible in every scenario, but there are tools available in Github to make this possible for the majority of our repos.

Github Codespaces

Github Codespaces are virtual machines (VMs) owned by Github that are connected to each repository. They let a user open the repo in a browser IDE (Integrated Development

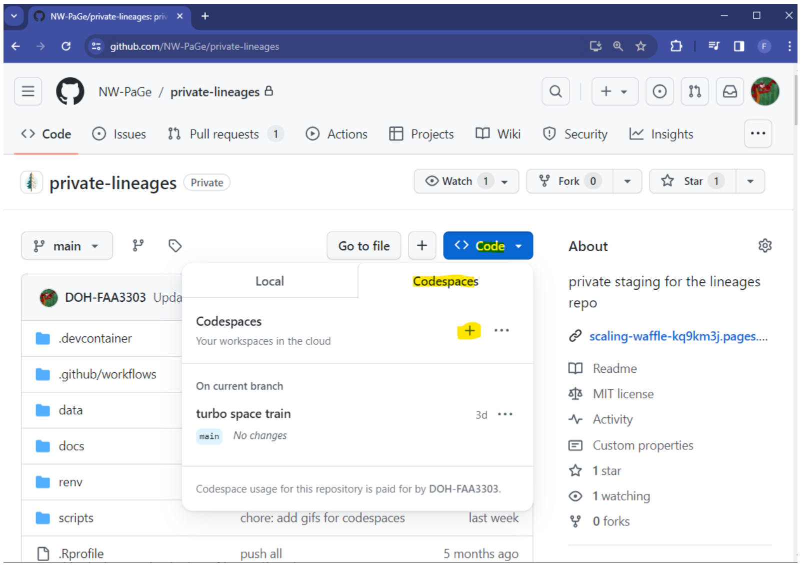
Environment) and execute the code in that environment. There is no set up or installation necessary for them.

The VMs are free for up to 60 hours a month of use and there are more hours added for Github users with paid memberships. 60 hours/month should be plenty for our purposes. Users are responsible for their own Codespace, so if they go over the limit they will be responsible for adding more hours and paying for the service.

Open a Codespace

At the root of the repo, click on the **Code** drop down button

1. On the right there is a tab called Codespaces.
2. Click the + sign and a Codespace will launch



Codespace usage for this repository is paid for by DOH-FAA3303.

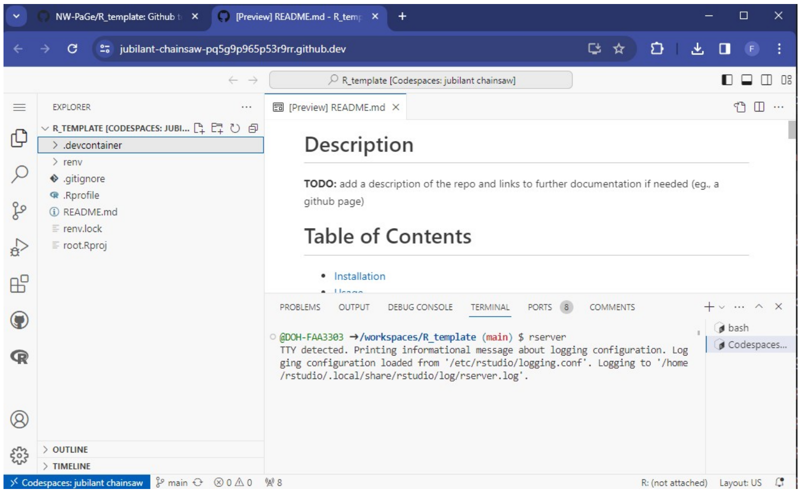
private staging for the lineages repo

chore: add gifs for codespaces

push all

Figure 1: open up a codespace

This will open up a VS Code window in your browser. There are also options to open up a Jupyter Notebook or Jetbrains IDE (Pycharm). You can also install an Rstudio IDE into the codespace. It will look something like this - note that the repository is already linked and checked out into the codespace:



Description

TODO: add a description of the repo and links to further documentation if needed (eg., a github page)

Table of Contents

- Installation

```
@DOH-FAA3303 ➜/workspaces/R_template (main) $ rserver
TTY detected. Printing informational message about logging configuration. Log
ging configuration loaded from '/etc/rstudio/logging.conf'. Logging to '/home
/rstudio/.local/share/rstudio/log/rserver.log'.
```

Figure 2: VS Code IDE in Github Codespaces

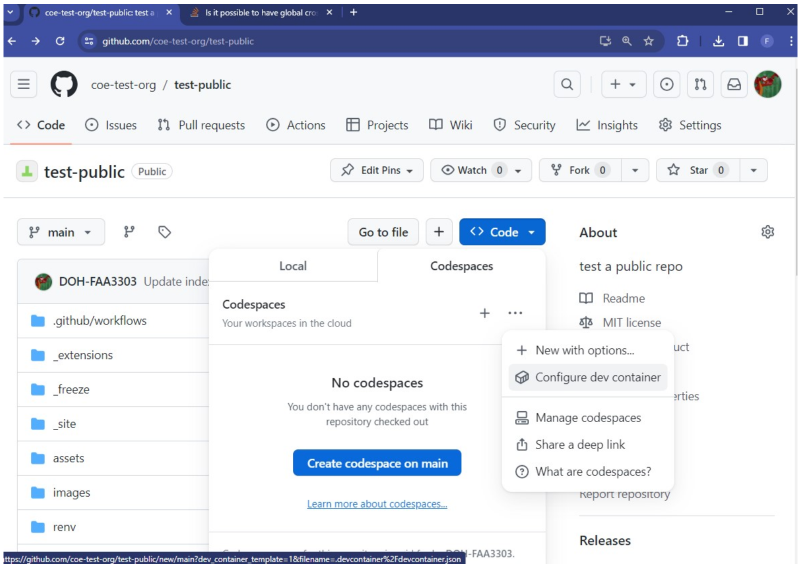
Here you can install most software. You can also customize the Codespace so that whenever someone opens one in your repo it will come with software pre-installed. More on that in the devcontainers section

Devcontainers

Devcontainers are a way to install software into a Codespace so that whenever a user opens up the Codespace they won't need to install anything themselves. Making a container can be a little tricky, so we've made Github templates that have devcontainers already made. See templates. There are R, Python, and general default templates. These containers will install R, Rstudio, Python, and all the packages in the repo's virtual environments (venv, conda, pip, renv, etc) so that the user can run all the code in your repo within a couple minutes.

To set up a devcontainer for yourself;

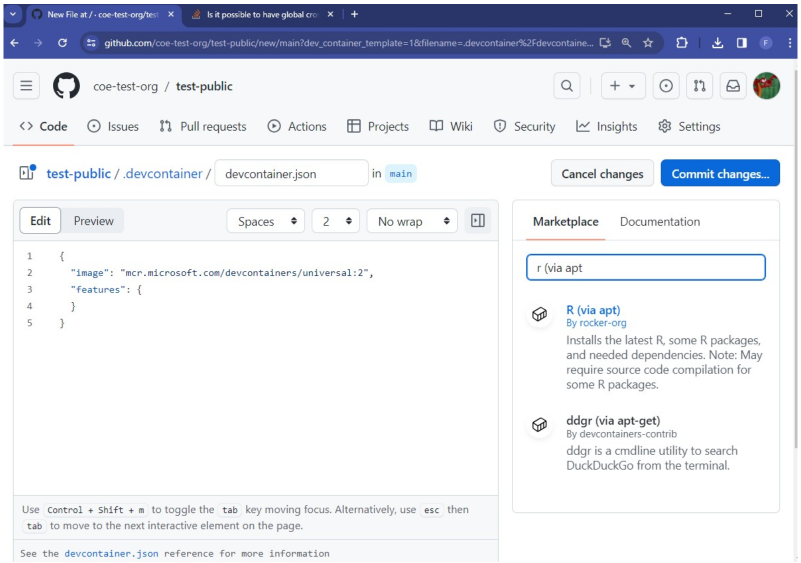
1. Click on 'Code > Codespaces > Configure dev container



test a public repo

You don't have any codespaces with this repository checked out

2. This will make a folder named `.devcontainer` at the root of your repo
3. In that folder it will make a file named `devcontainer.json`
4. On the right there is a searchable marketplace for software to add to your container


New File at / · coe-test-org/test
Is it possible to have global cro
github.com/coe-test-org/test-public/new/main?dev_container_template=1&filename=.devcontainer%2Fdevcontaine...
coe-test-org / test-public
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
test-public / .devcontainer / devcontainer.json in main
Cancel changes
Commit changes...
Edit
Preview
Spaces
2
No wrap
Marketplace
Documentation
r (via apt

```
{
  "image": "mcr.microsoft.com/devcontainers/universal:2",
  "features": {
  }
}
```

Use `Control + Shift + m` to toggle the `tab` key moving focus. Alternatively, use `esc` then `tab` to move to the next interactive element on the page.

See the devcontainer.json reference for more information

R (via apt)
By rocker-org
Installs the latest R, some R packages, and needed dependencies. Note: May require source code compilation for some R packages.

ddgr (via apt-get)
By devcontainers-contrib
ddgr is a cmdline utility to search DuckDuckGo from the terminal.

5. Each one comes with instructions on how to add the software to the `.devcontainer.json`

For more information about Codespaces, see the guides here

Virtual Environments

Virtual Environments are another great way to make sure aspects of your repo are reproducible. They are commonly used to record package versions that the code/project uses. For more on virtual environments, please see the venv guide.

Github Releases

Github Releases save code snapshots, versions, and changelogs of your repo. They are a great way for end users and developers to use different versions of their code and visualize changes that happened with each version. Please see the Github Releases guide for more information.

Documentation

Your code should be well documented so that end users (and developers) can understand what code is doing, how to install the software, and the utility of the project.

In general, you should have a README.md file in your repo that explains at least a high level summary of the code in the repo and what it does, how to install the code, outputs, and how to contribute to the repo. In addition, it may be a good idea to make a Github Page (a static website hosted in your repo) that explains the code in more detail. See the documentation guides here.

Having a Github Page is necessary if you have a package. Consider using software like `pkgdown` for R or `quartodoc` for Python (or other related software that helps link code to your documentation automatically). See more about package documentation here.