

# Initializing and Synchronizing a Git Repo with GitHub Desktop

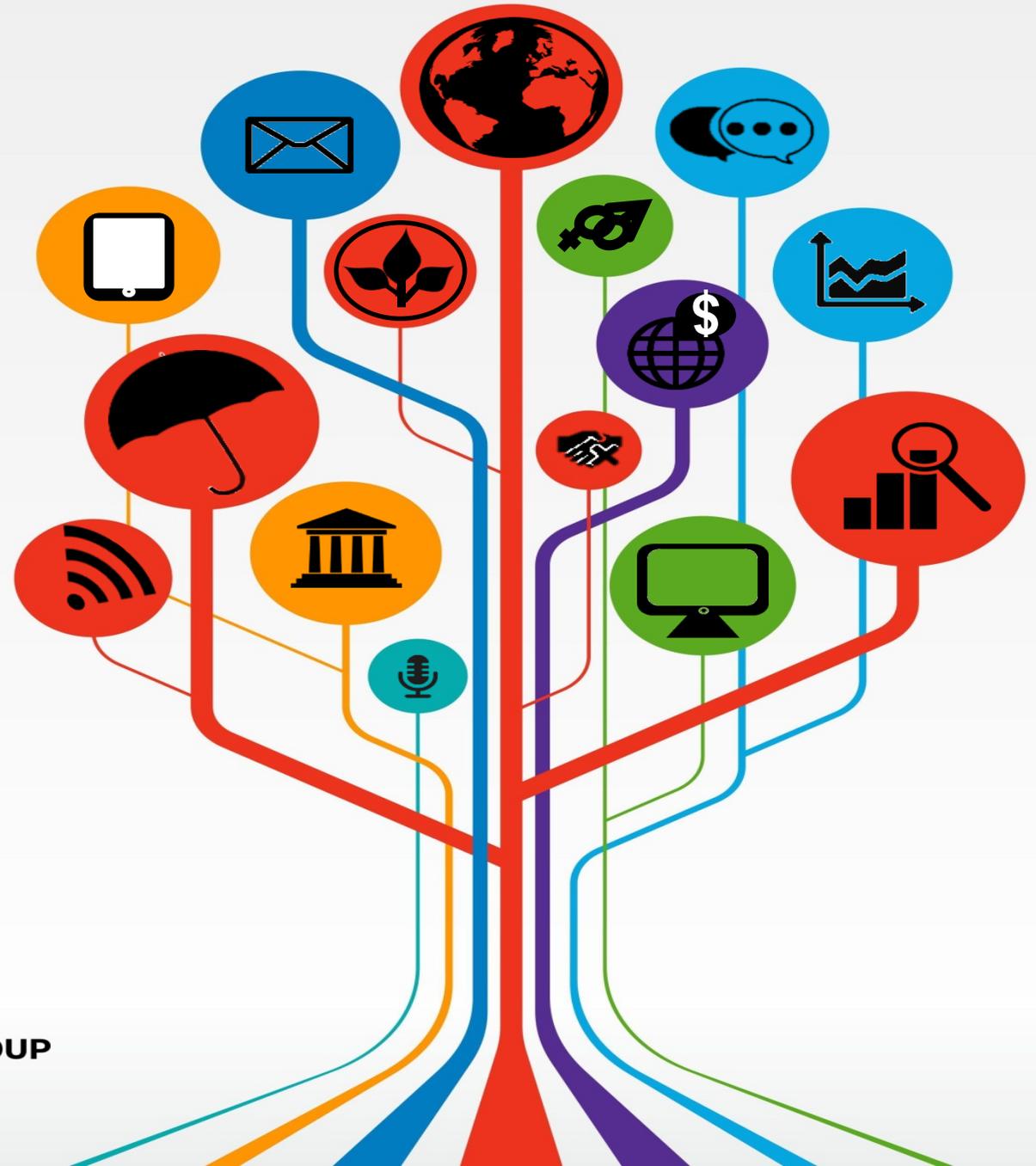
## Research Assistant Onboarding

Prepared by DIME Analytics

[dimeanalytics@worldbank.org](mailto:dimeanalytics@worldbank.org)

Presented by Benjamin Daniels

[bdaniels@worldbank.org](mailto:bdaniels@worldbank.org)



# Create a repository on GitHub

**Create a new repository**  
A repository contains all the files for your project, including the revision history.

**Owner** **Repository name**

/ 

Great repository names are short and memorable. Need inspiration? How about [jubilant-octo-journey](#).

**Description (optional)**

**Public**  
Anyone can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

**Initialize this repository with a README** ← 3  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

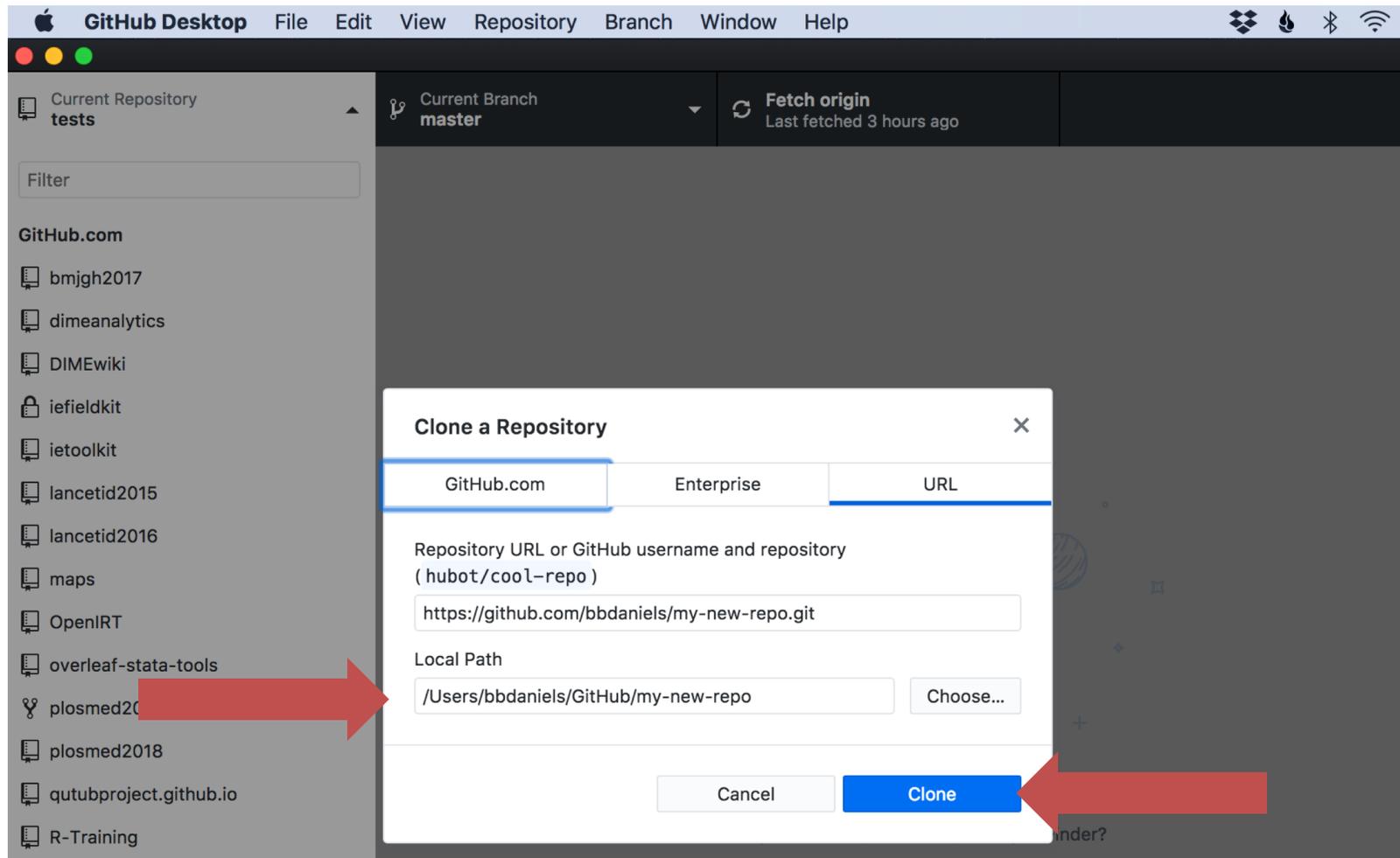
Add .gitignore:  | Add a license:  ⓘ

← 4

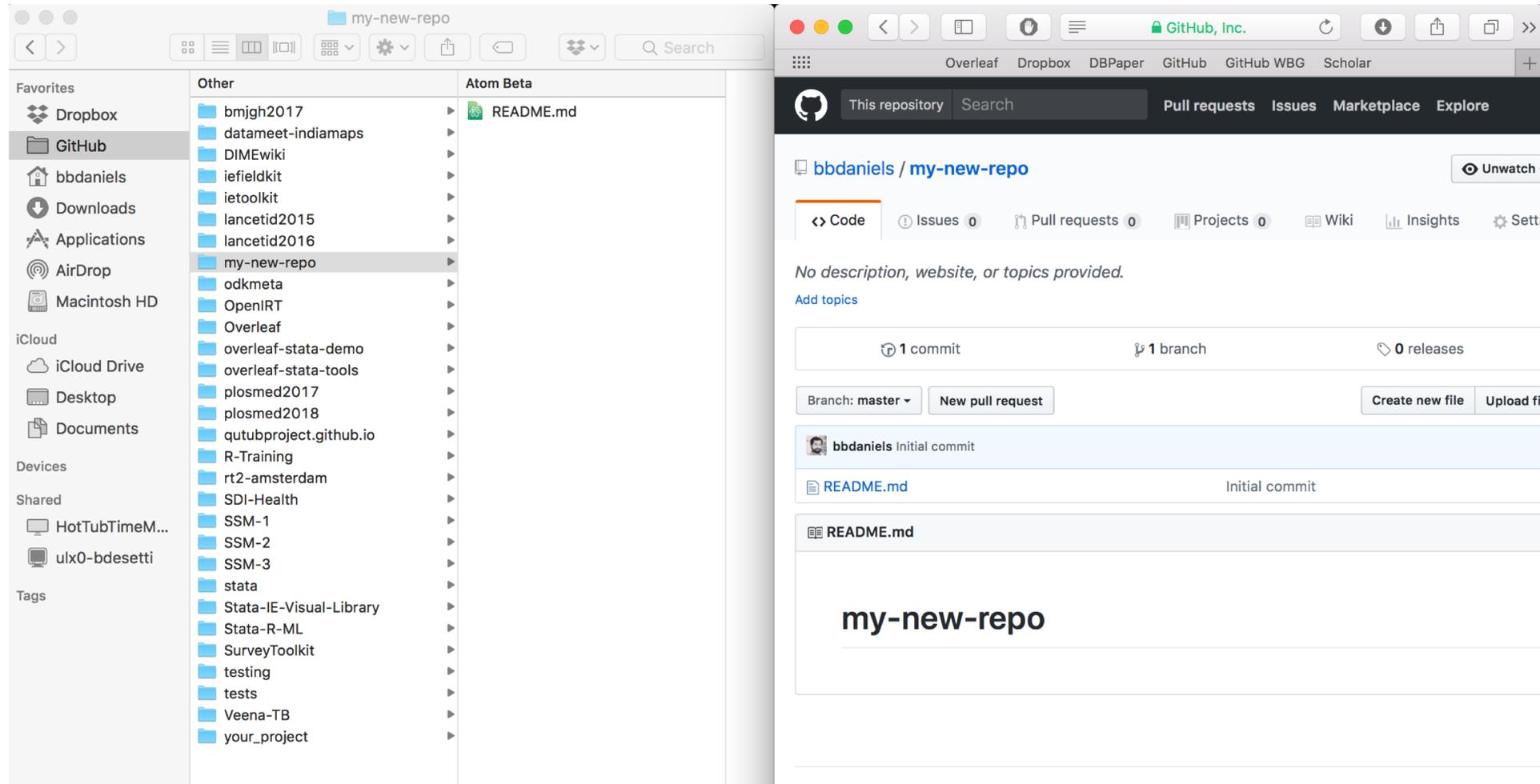
# Setup in GitHub Desktop

The screenshot shows the GitHub web interface for a repository named 'my-new-repo' by user 'bbdaniels'. The repository has 1 commit, 1 branch, 0 releases, and 1 contributor. The 'Clone or download' button is highlighted with a red arrow. The dropdown menu is open, showing the 'Clone with HTTPS' option selected, with the URL 'https://github.com/bbdaniels/my-new-r'. Below the URL are two buttons: 'Open in Desktop' (highlighted in blue) and 'Download ZIP'. A tooltip at the bottom of the dropdown reads: 'Clone bbdaniels/my-new-repo to your computer and use it in GitHub Desktop.'

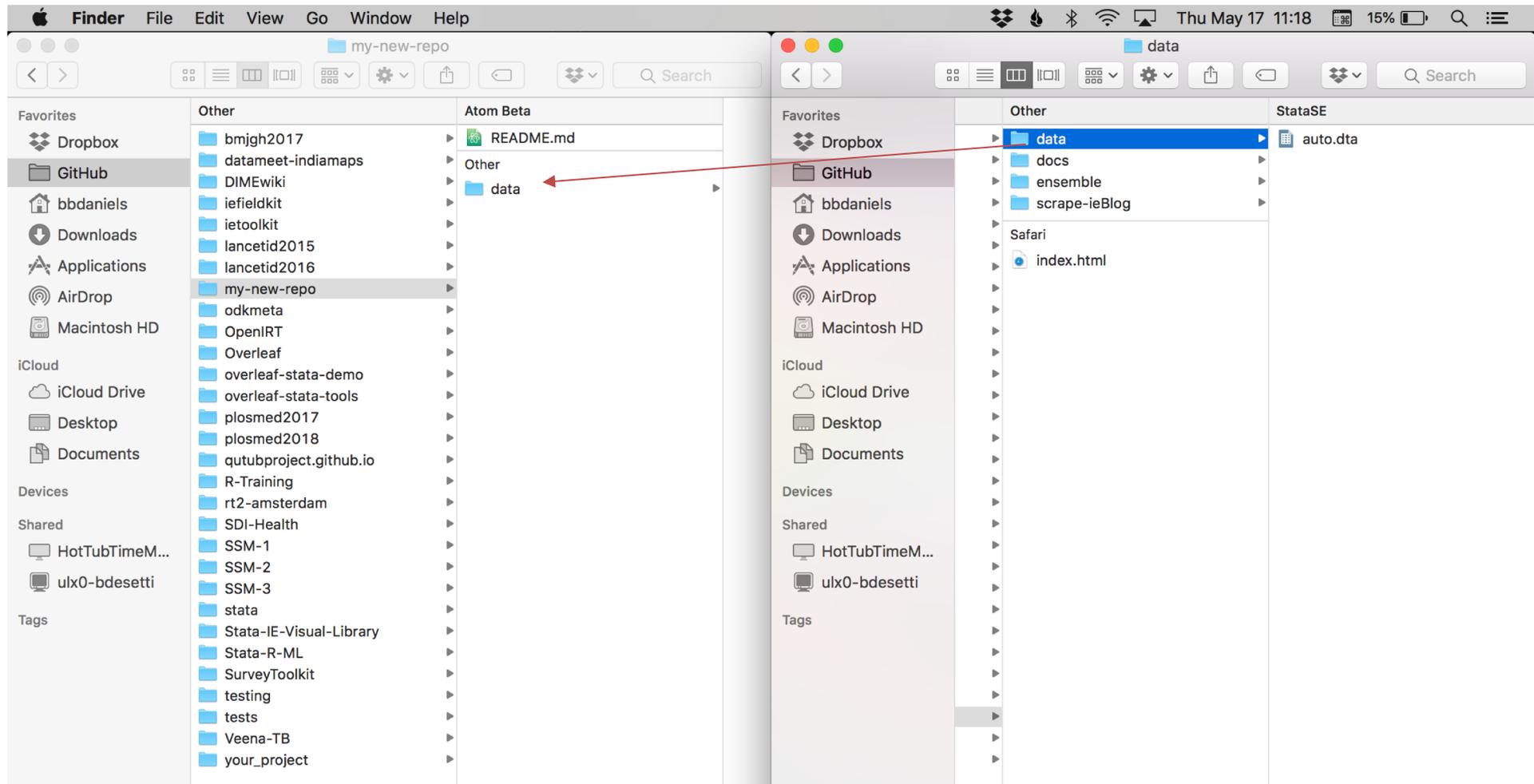
# Set local destination for “clone” repository



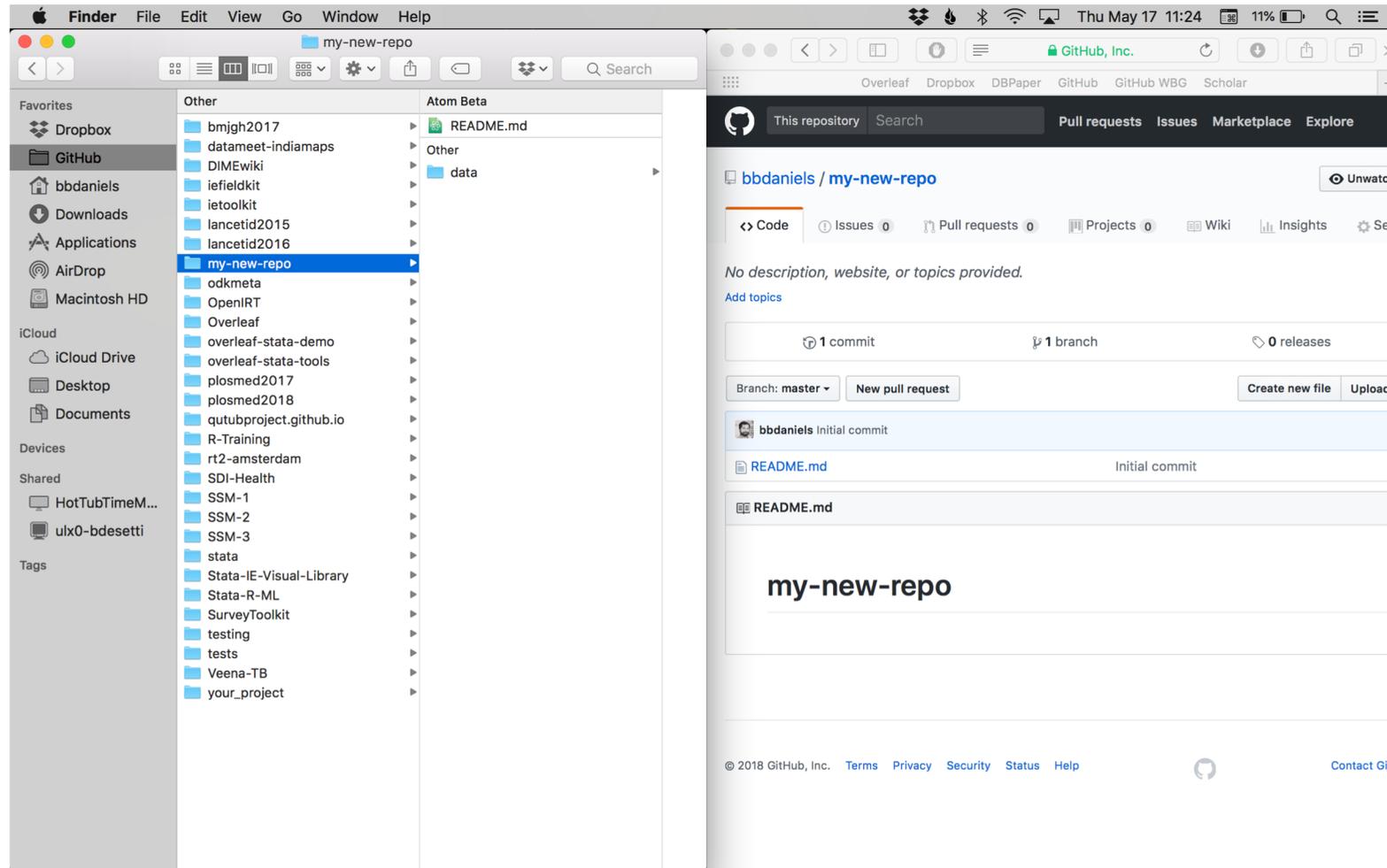
# The “local” and “remote” instances are identical



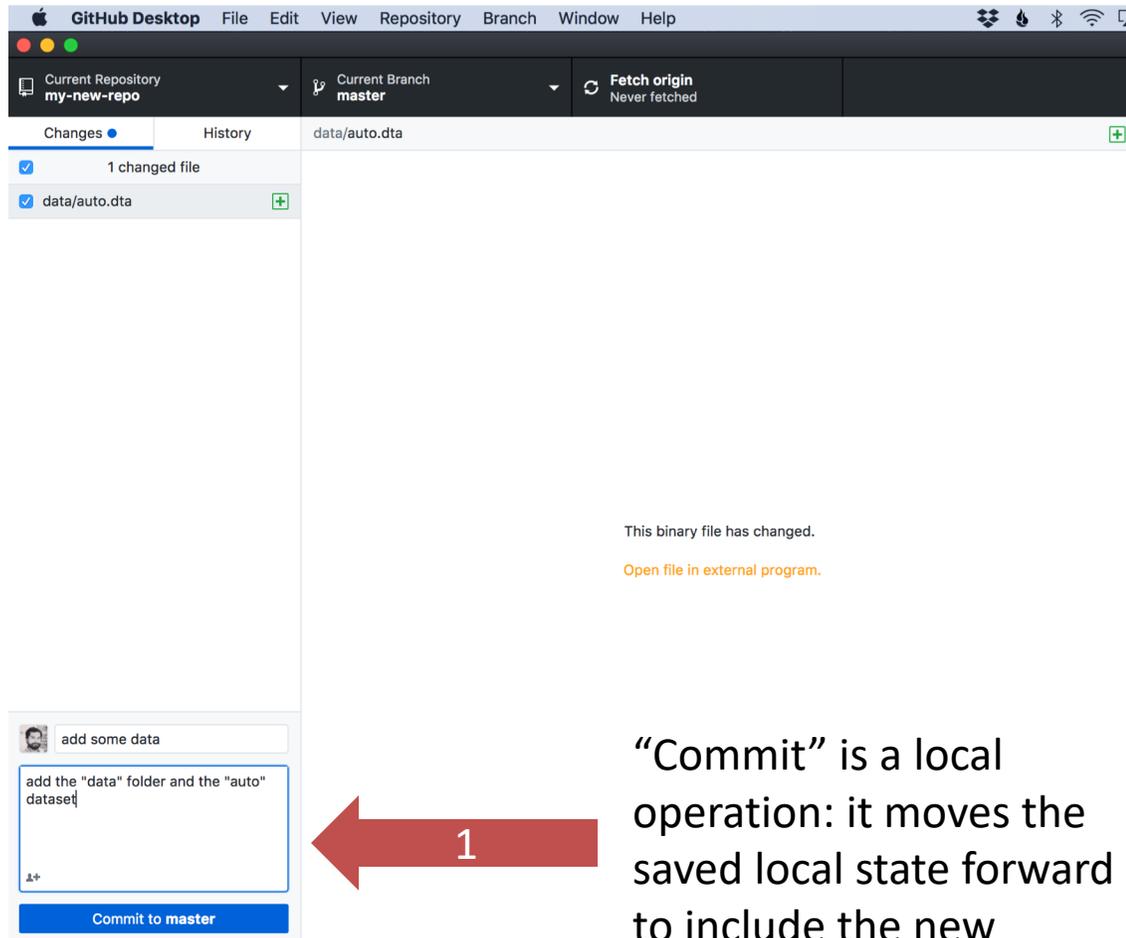
# Add some data locally...



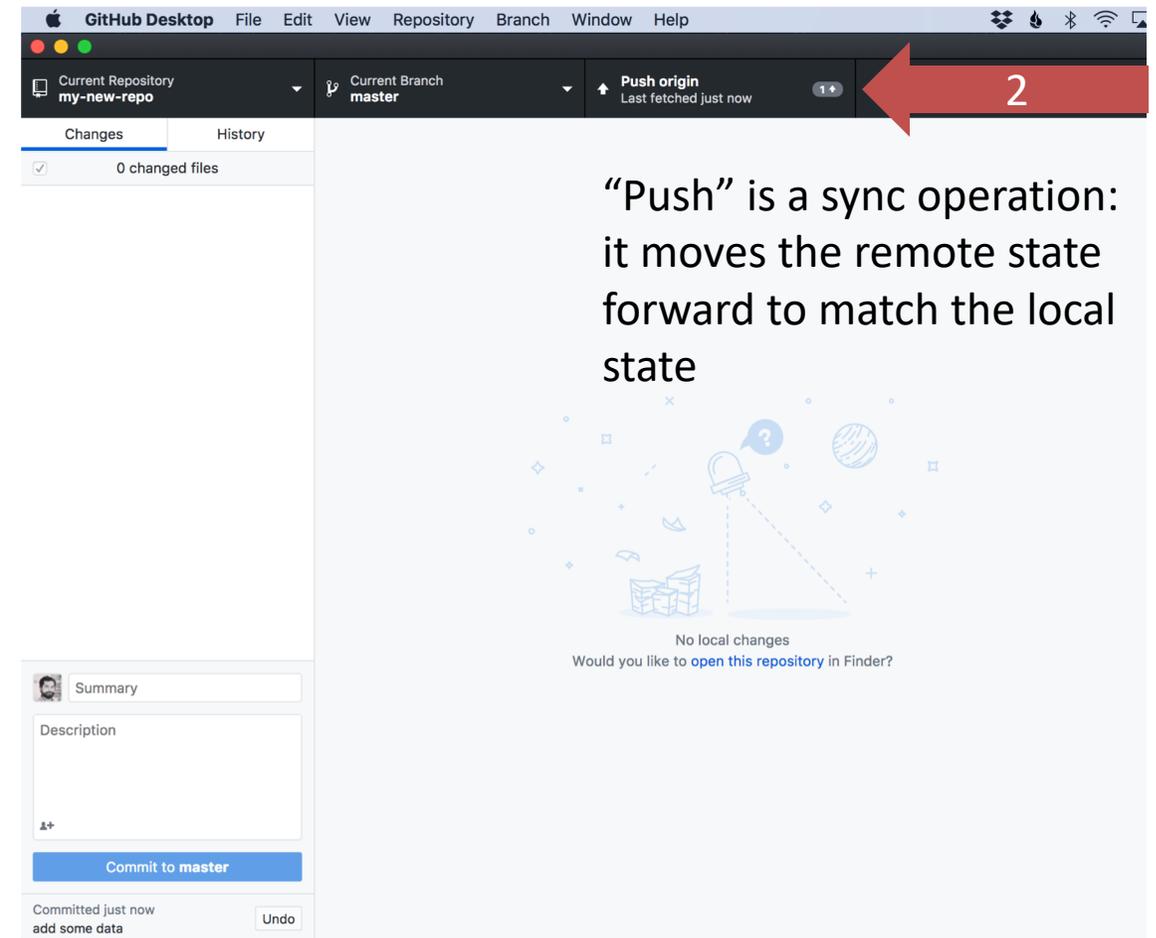
# The local and remote are *not* in sync



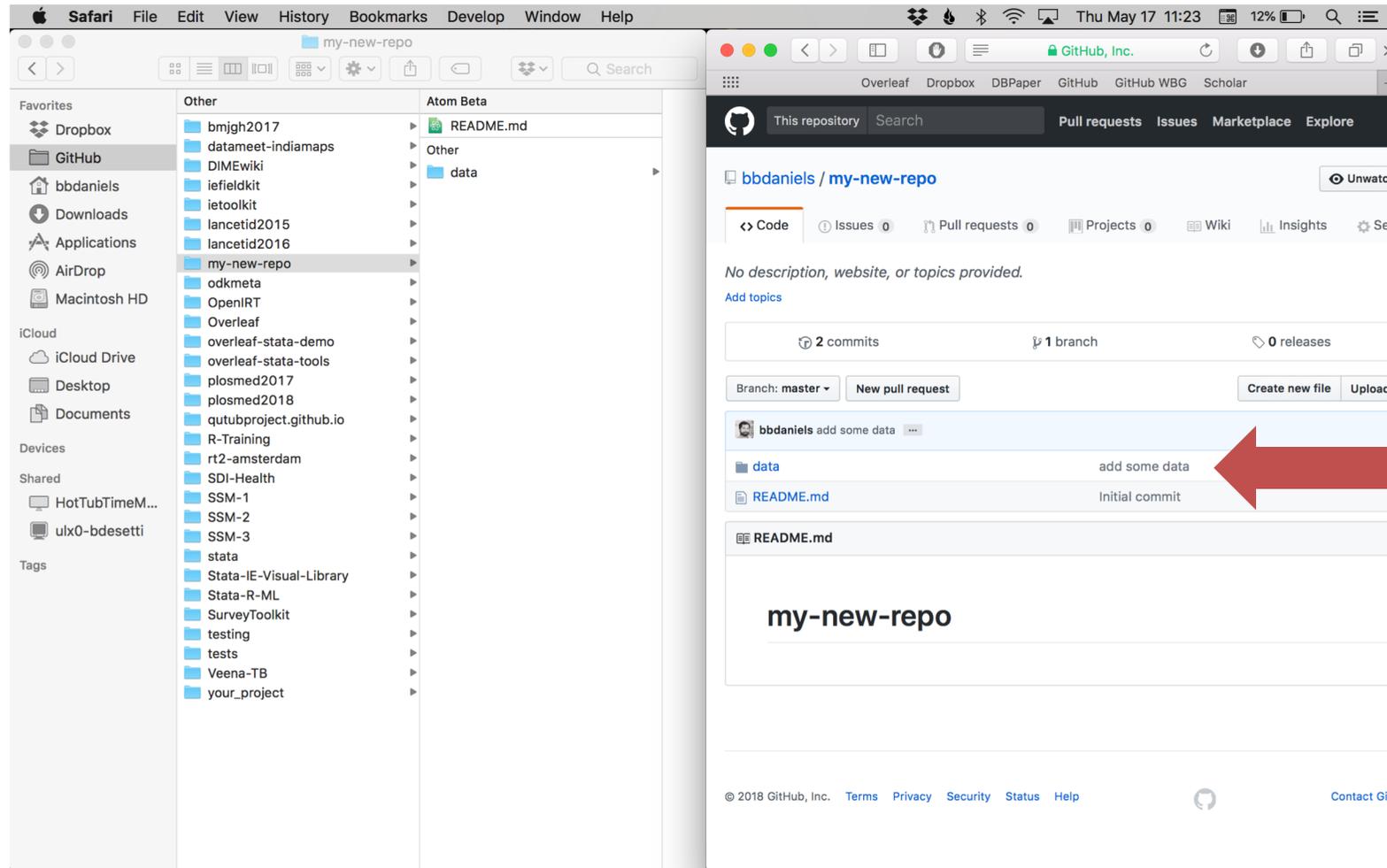
# GitHub Desktop notes the changes and allows you to push them to the remote (“origin”)



“Commit” is a local operation: it moves the saved local state forward to include the new changes



# The local and remote are again in sync



Each item records when it was last modified

# Git can do much more!

---

- For now, you have created a version history of your files with a publicly accessible remote backup
- Git is designed such that this remote backup serves as a coordination point for multiple people making changes simultaneously, through “branching”
- We will cover this later!
- [Pro Git](#) is a good introduction to all of Git’s many features
- And [GitKraken](#) is a better desktop client for interacting with the version history and the advanced features

Thank you!

