Publishing Software, GitHub & Zenodo …

Megan Potterbusch ~ Dan Kerchner
GW Libraries & Academic Innovation

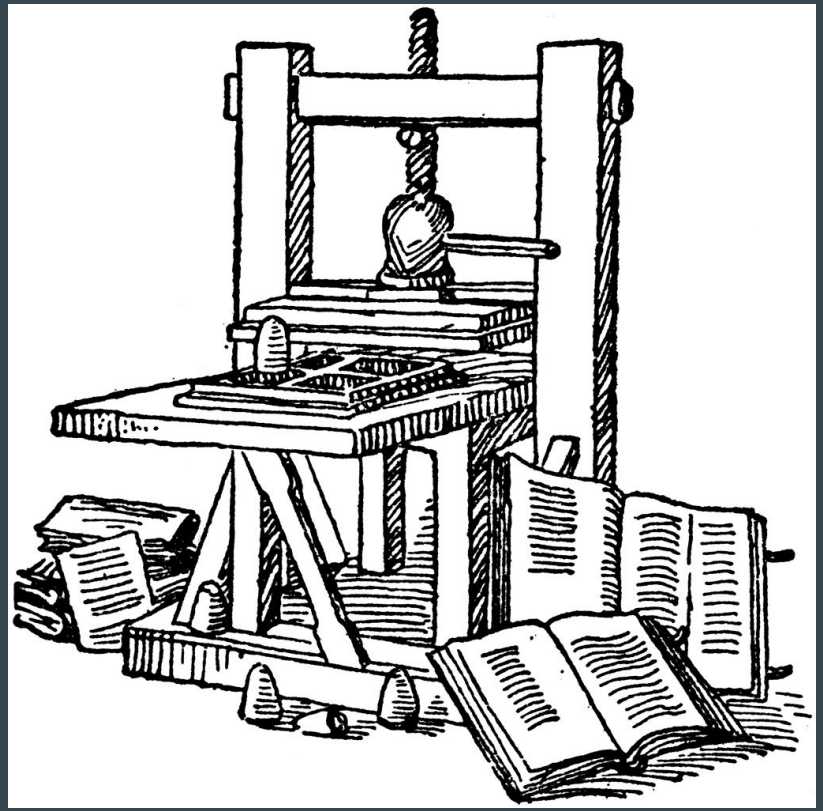
March 6, 2018

go.gwu.edu/swpub

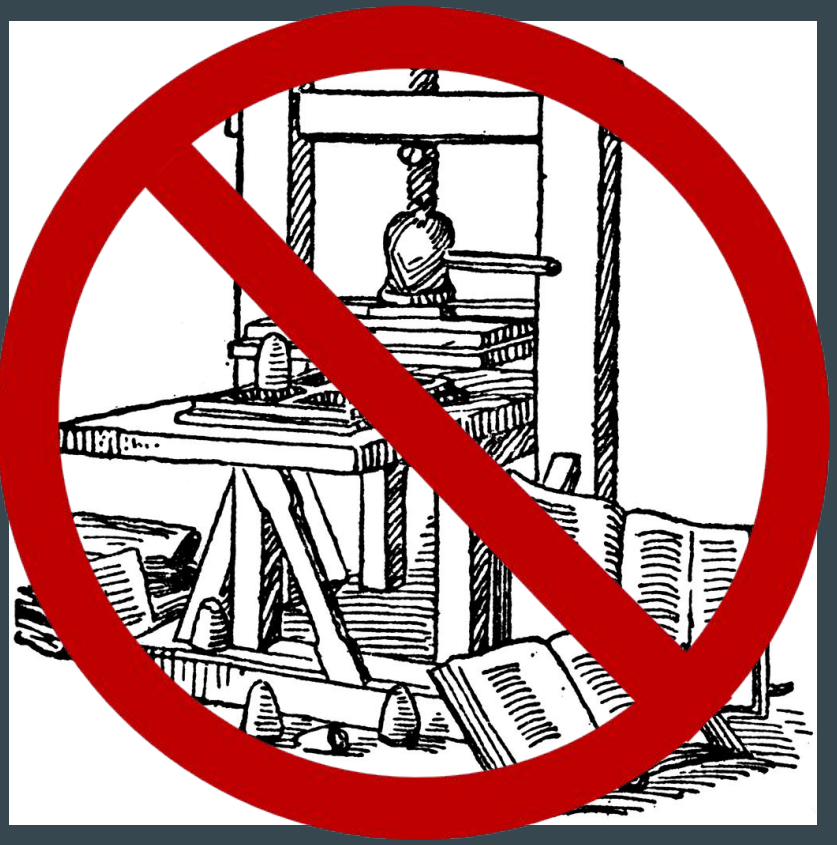
Agenda

- Get Scholarly Credit for Writing Software
  - Citable, durable, publishable software
- Collaborating with GitHub (Hands-on)

What is Software Publication?



What is Software Publication?



What is Software Publication?

- Accessible
  - Available
- Citable
  - Title; Author(s); Version; Unique Persistent Identifier;
- Usable
  - Documented
  - Commented; Openly Licensed; Dependencies Outlined

Make your work citable!

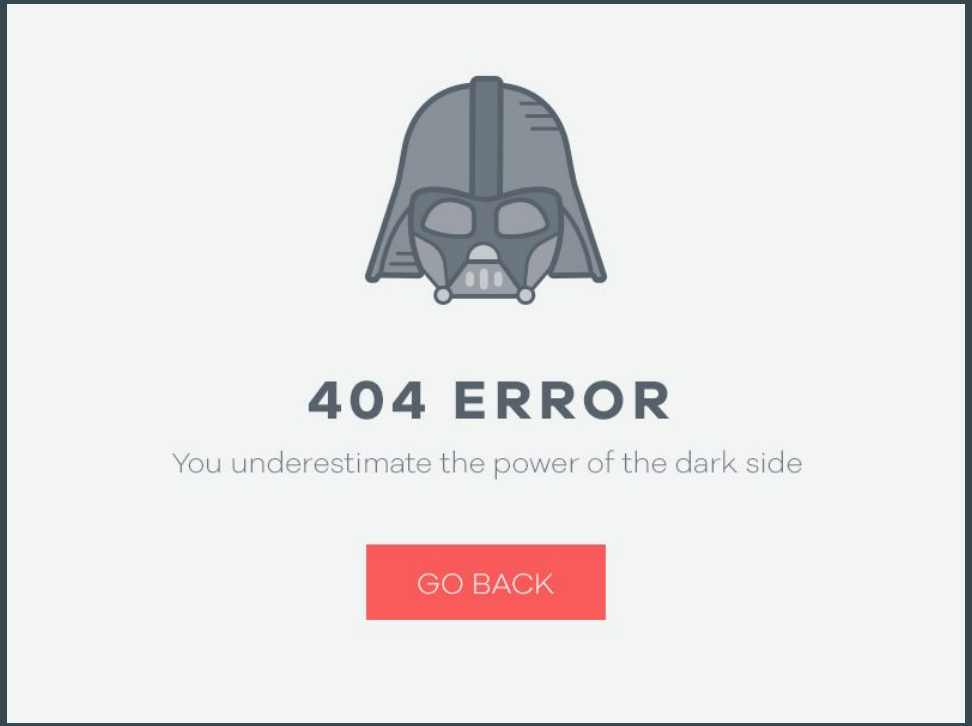


Image Credit: Susana Gonzalez

Make your work citable!

DOI == Digital Object Identifier

Persistent, externally managed, standardized

http://dx.doi.org/

Standard: ISO 26324

Levels of “Publication” for Software

- Supplement to traditional article publishing
- Citable and Peer reviewed
- Citable online resource
- Publically available generally

Software as: Supplemental to Publication

Authors:	Diego Franco Saldana, Yang Feng
Title:	**SIS: An R Package for Sure Independence Screening in Ultrahigh-Dimensional Statistical Models**
Abstract:	We revisit sure independence screening procedures for variable selection in generalized linear models and the Cox proportional hazards model. Through the publicly available R package SIS, we provide a unified environment to carry out variable selection using iterative sure independence screening (ISIS) and all of its variants. For the regularization steps in the ISIS recruiting process, available penalties include the LASSO, SCAD, and MCP while the implemented variants for the screening steps are sample splitting, data-driven thresholding, and combinations thereof. Performance of these feature selection techniques is investigated by means of real and simulated data sets, where we find considerable improvements in terms of model selection and computational time between our algorithms and traditional penalized pseudo-likelihood methods applied directly to the full set of covariates.
	Page views:: 345. **Submitted:** 2014-05-10. **Published:** 2018-02-22.
Paper:	SIS: An R Package for Sure Independence Screening in Ultrahigh-Dimensional Statistical Models Download PDF (Downloads: 153)
Supplements:	SIS_0.8-6.tar.gz: R source package Download (Downloads: 6; 2MB) v83i02-replication.zip: Replication materials Download (Downloads: 9; 5KB)
DOI:	10.18637/jss.v083.i02

Link to article: https://www.jstatsoft.org/article/view/v083i02

Software as: Supplemental to Publication

HALOGEN: a tool for fast generation of mock halo catalogues

Santiago Avila ✉, Steven G. Murray, Alexander Knebe, Chris Power, Aaron S. G. Robotham, Juan Garcia-Bellido

Monthly Notices of the Royal Astronomical Society, Volume 450, Issue 2, 21 June 2015, Pages 1856–1867, https://doi.org/10.1093/mnras/stv711

thousands of simulated realizations. We discuss the applicability of our method in this context, and conclude that it is well suited to mass production of appropriate halo catalogues. The code is publicly available at https://github.com/savila/halogen.

Link to article: https://www.jstatsoft.org/article/view/v083i02

Software as: Supplemental to Publication

HALOGEN: a tool for fast generation of mock halo catalogues

Santiago Avila ✉, Steven G. Murray, Alexander Knebe, Chris Power, Aaron S. G. Robotham, Juan Garcia-Bellido

Mon
185

Acknowledgments

If you find this code helpful in your research, please cite Avila, S. et al. 2014 -- http://arxiv.org/abs/1412.5228.

this context, and conclude that it is well suited to mass production of appropriate halo catalogues. The code is publicly available at https://github.com/savila/halogen.

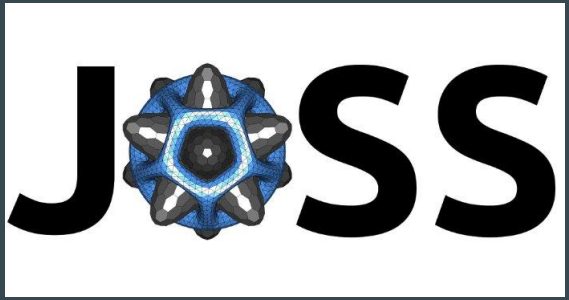
Link to article: https://www.jstatsoft.org/article/view/v083i02

Software as: Peer-Reviewed Scholarship

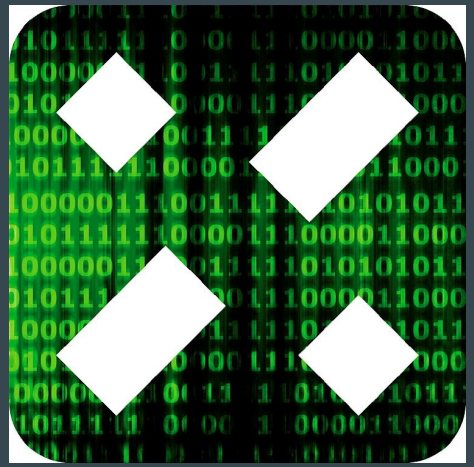
- Journal of Open Source Software (JOSS)
  - https://joss.theoj.org/
  - Example article: https://joss.theoj.org/papers/e33112ace30bb700c5e46baf9230df5c
- SoftwareX (part of *Research Elements*)
  - https://www.sciencedirect.com/journal/softwarex
  - Example article: https://www.sciencedirect.com/science/article/pii/S2352711018300165

Journal of Open Source Software (JOSS)

- Contents of a JOSS Paper:
  - A list of the authors of the software and their affiliations (i.e. George Washington University)
  - Summary of function and purpose of the software
    - Must be written for a diverse, non-specialist audience
  - A clear statement of need for the software. “Why does this software exist?”
  - References
  - Mentions (if applicable) to other research projects using this software
- Suggested Length: 250-1000 words
  - More details: https://joss.theoj.org/about#author_guidelines



SoftwareX



- Contents of a SoftwareX Paper:
  - Motivation and significance
  - Software description
  - Illustrative examples
  - Major assets
  - Impact
  - Conclusions
  - References
  - Keywords
  - (Code and Software) Metadata
- Max Length: 3000 words
  - More Details: https://www.elsevier.com/journals/softwarex/2352-7110/guide-for-authors#17106

Software as: Peer-Reviewed Scholarship

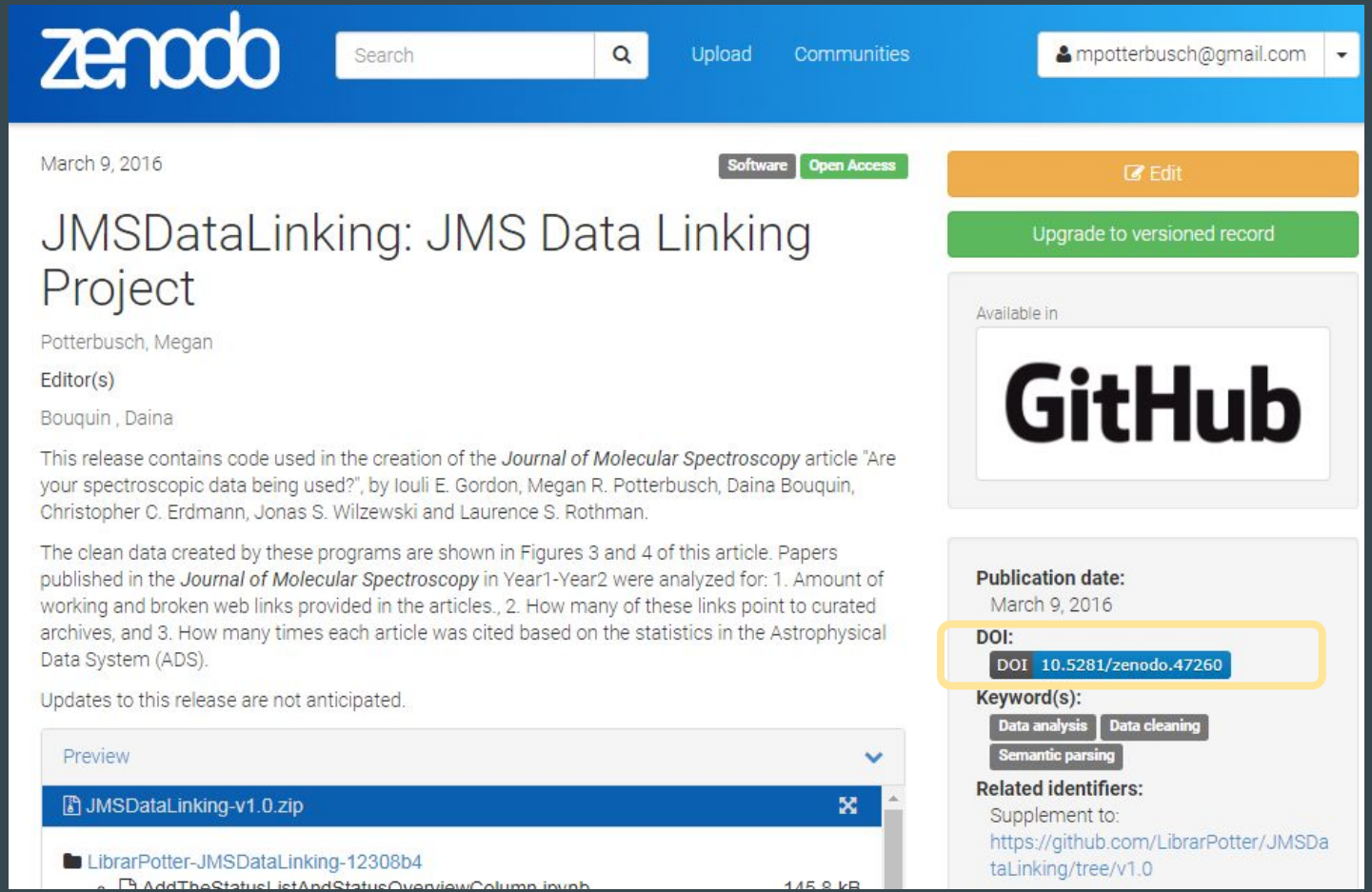
- Typically a “Software Paper”
- Citable resource
- Goes through Peer-review

Software as: A Citable Resource

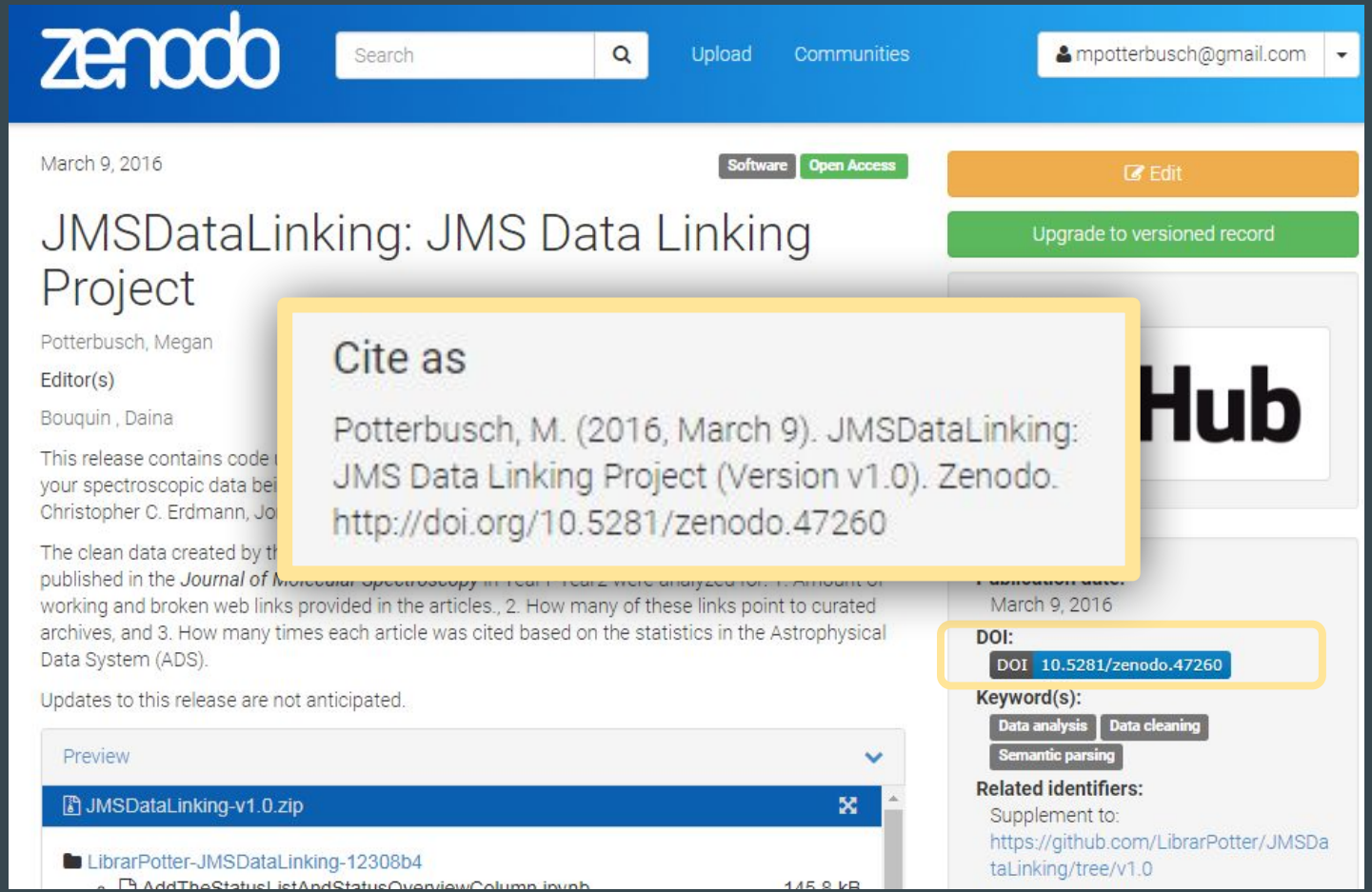
Upload Software to a Repository

They Preserve the Files

You Get a DOI



Link to resource: https://zenodo.org/record/47260#.Wp7442rwbIW



Link to resource: https://zenodo.org/record/47260#.Wp7442rwbIW

Journal of Molecular Spectroscopy 327 (2016) 232–238

Contents lists available at ScienceDirect

Journal of Molecular Spectroscopy

journal homepage: www.elsevier.com/locate/jms

Are your spectroscopic data being used?

Iouli E. Gordon [a,*], Megan R. Potterbusch [a], Daina Bouquin [a], Christopher C. Erdmann [a], Jonas S. Wilzewski [a,b,c], Laurence S. Rothman [a]

[a] Harvard–Smithsonian Center for Astrophysics, Atomic and Molecular Physics Division, Cambridge, MA 02138, USA
[b] Ludwig-Maximilians-Universität München, Faculty of Physics, Schellingstr. 4, 80799 Munich, Germany
[c] German Aerospace Center (DLR) Oberpfaffenhofen, 82234 Weßling, Germany

ARTICLE INFO

Article history:
Received 20 January 2016
In revised form 12 March 2016

ABSTRACT

The issue of availability of data and their presentation in spectroscopic publications is discussed. Different curr
policies and r

[20] M. Potterbusch, JMSDatalinking: JMS Data Linking Project. Zenodo, 2016. <https://zenodo.org/record/47260#.VuWB0ebcTIU>. doi:http://dx.doi.org/10.5281/zenodo.47260.

Link to resource: https://doi.org/10.1016/j.jms.2016.03.011

Software as: An Online Resource

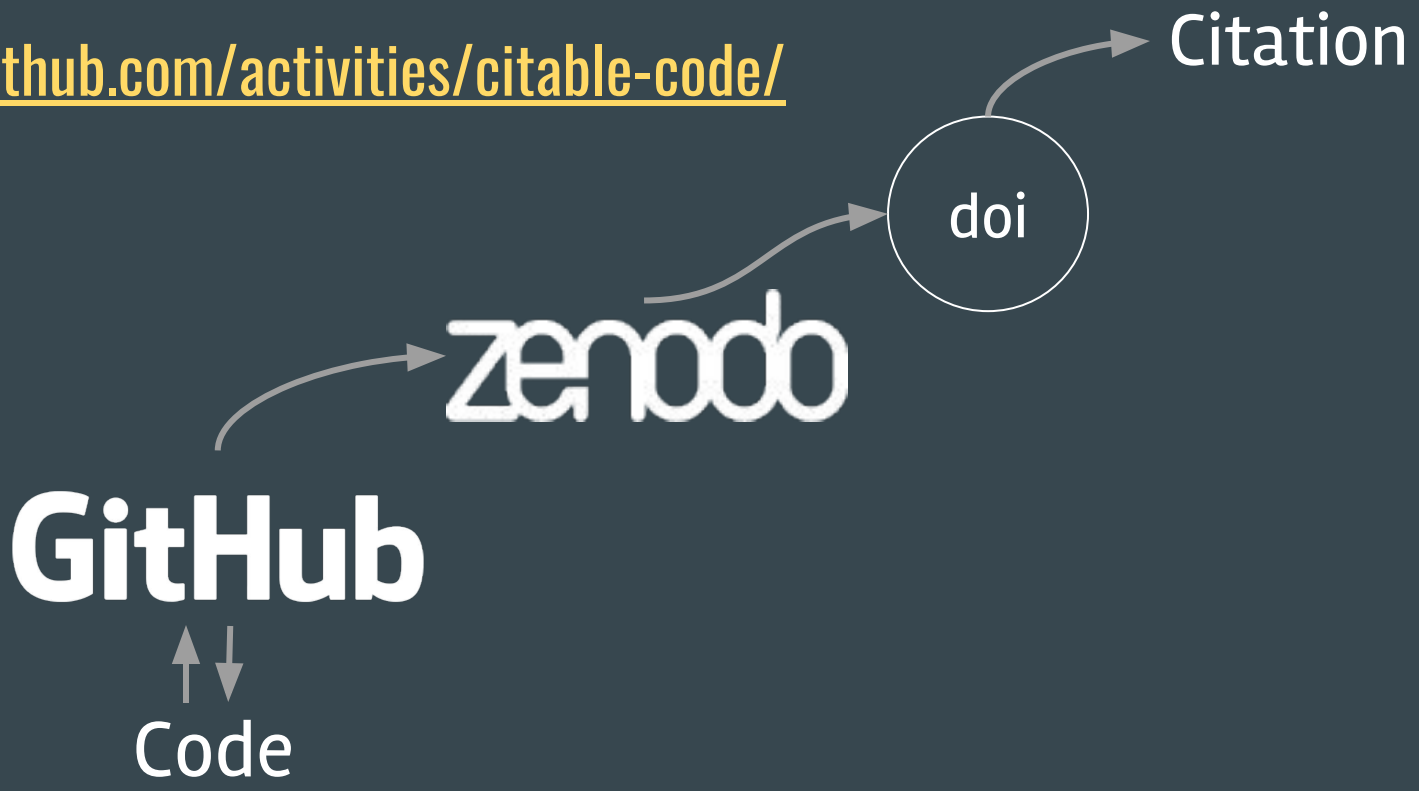
- Online Collaborative Software Development
  - GitHub, Bitbucket, GitLab etc.
- Websites
  - Personal, University, Industry
- Any Resources without a Preservation Plan
  - Hint: they won't issue DOIs

GitLab

Bitbucket

Context: Making Your Code Citable

guides.github.com/activities/citable-code/



Collaborating with GitHub

GitHub Agenda

- Why GitHub?
- A tour of a repository
- Hands-on:
  - Create your own new repository
  - Collaborate on a repository
  - Make a release of your code
  - Fork an existing repository

Why GitHub?

- Version Control - uses `git`
- Collaboration
- Open (or not)
- Pricing - free for open, cheap for private, and free for students education.github.com (and /pack)
- Code with context

Tour of a Repository: About your code

- **README.md** ← showcase your work, this is your home page
- **LICENSE** file ← Important! choosealicense.com
- (Optional) Wiki pages

These use Markdown - lightweight formatting

Tour of a Repository - The Code

- Code
  - history of commits on a branch
  - history of commits on a file
- Branches
  - Master branch vs. others
  - Visually represented with the Insights → Network graph

Tour of a Repository - Code as a Project

- Issues - tags, assignment, releases, comments
- Milestones and Releases
- Pull requests
- Comment threads - on issues, commits, etc.
- Insights (graphs)

Tour of a Repository - Connections

Between Github users/repositories

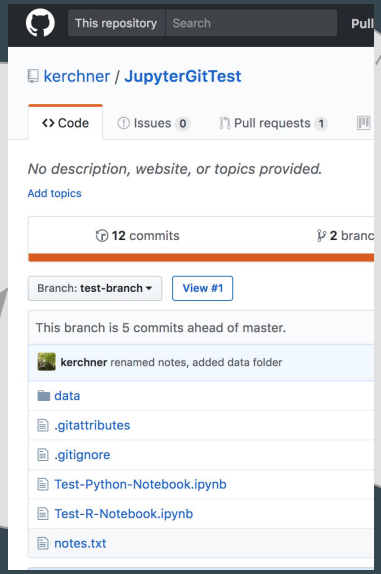
- Watch/Star/Follow
- Mentions

With other software development tools

- Integrations & Webhooks (under repository Settings)

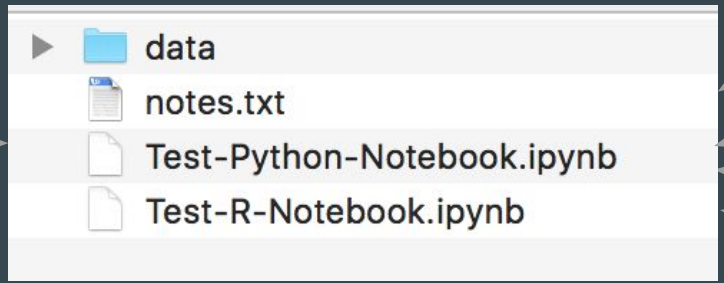
Your Code, Git, and GitHub

Your repository on GitHub



Apps you use that modify your files

Files on your computer - your *local* git repository



GitHub Desktop

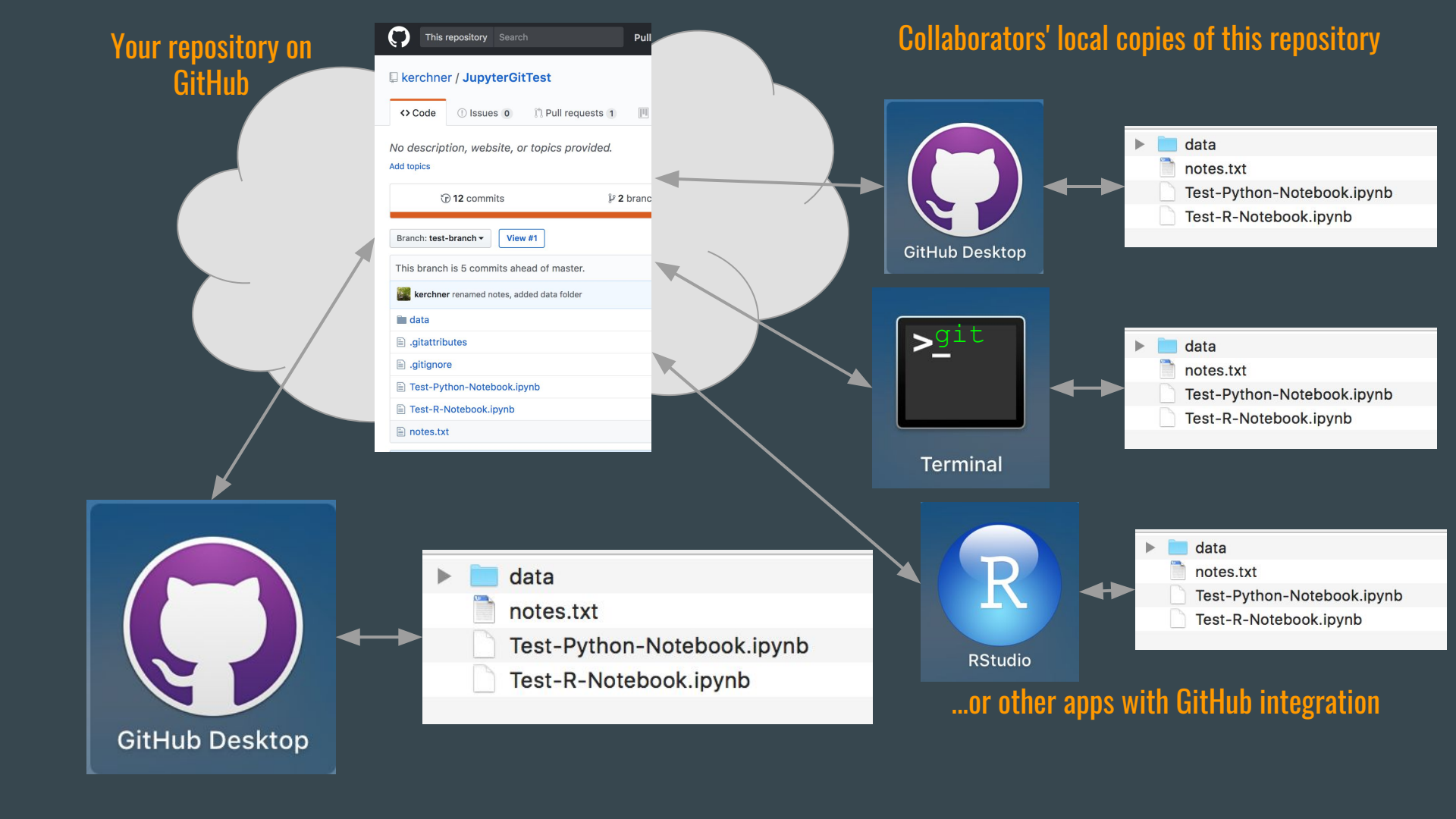
jupyter

Atom

RStudio

Microsoft Excel

** or command-line git, RStudio, Atom, other Git app, etc.*


Your repository on GitHub
This repository
Search
Pull
kerchner / JupyterGitTest
Code
Issues 0
Pull requests 1
No description, website, or topics provided.
Add topics
12 commits
2 branc
Branch: test-branch
View #1
This branch is 5 commits ahead of master.
kerchner renamed notes, added data folder
data
.gitattributes
.gitignore
Test-Python-Notebook.ipynb
Test-R-Notebook.ipynb
notes.txt
Collaborators' local copies of this repository
GitHub Desktop
data
notes.txt
Test-Python-Notebook.ipynb
Test-R-Notebook.ipynb
>_git
Terminal
data
notes.txt
Test-Python-Notebook.ipynb
Test-R-Notebook.ipynb
GitHub Desktop
data
notes.txt
Test-Python-Notebook.ipynb
Test-R-Notebook.ipynb
R
RStudio
data
notes.txt
Test-Python-Notebook.ipynb
Test-R-Notebook.ipynb
...or other apps with GitHub integration

Basic Terminology

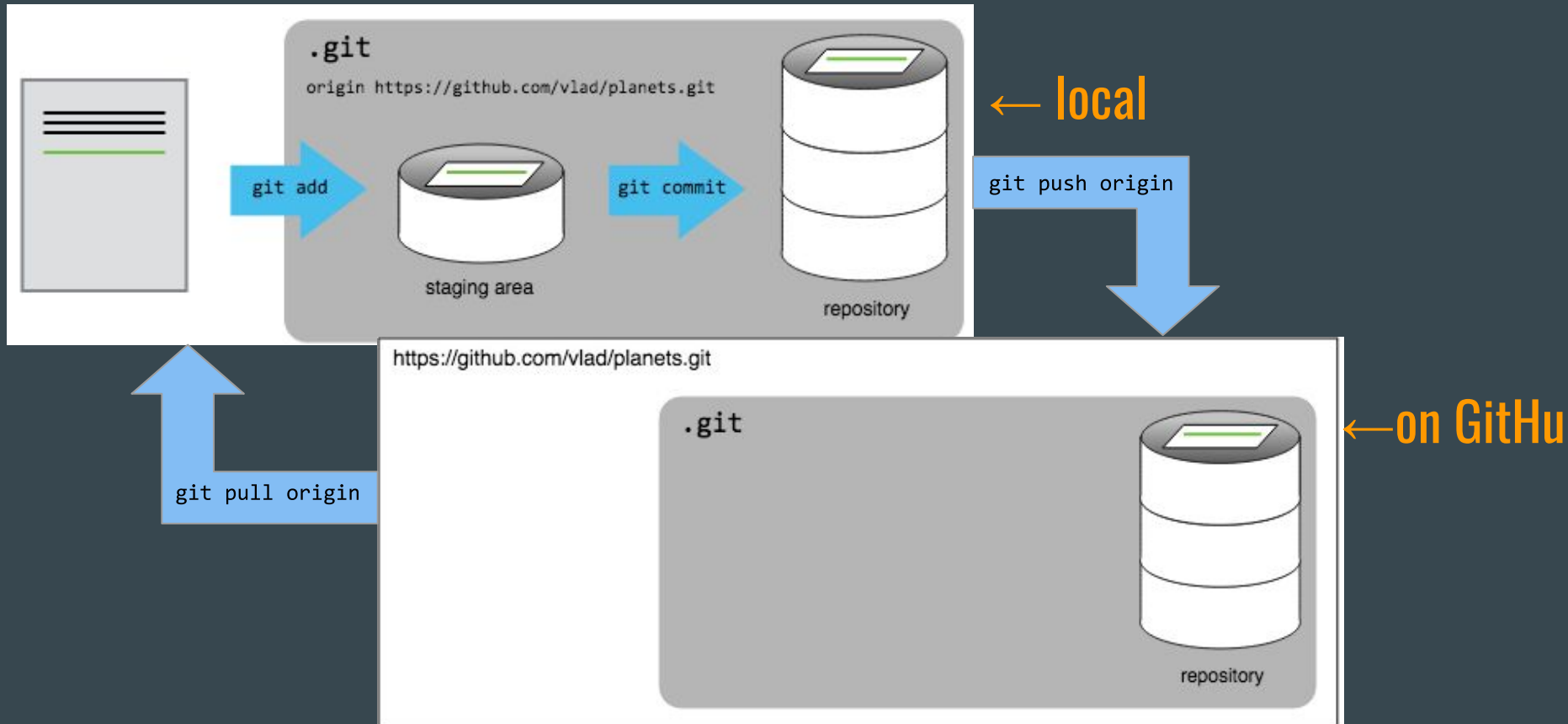
Add/remove - puts changes in a staging area

Commit - commits staged changes to the (local) repository

Push - pushes local repository changes to the remote repository

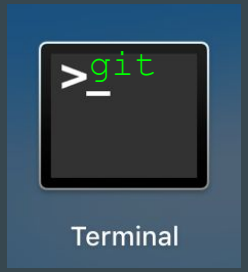
Pull - pulls remote repository changes to the local repository

add ~ commit ~ push ~ pull



Ways to use Git

*or other GUI



...and many IDEs

Ways To Use Git (cont.)

- Git command line
- GitHub Desktop App (or: git-scm.com/downloads/guis)
- GitHub Mobile App
- Integrations w/Editors, IDEs, and other apps
  - R Studio
  - Atom
  - ...many others

Git repository anatomy

- Open the folder containing your local git repository. It contains a `.git` directory which contains all the metadata about the repository (that is, the local copy)

Let's Try It!

Create your own GitHub repository

Prerequisites / Setup

Your Github account, plus **EITHER**:

- Install **GitHub Desktop app** via desktop.github.com
- Under *Preferences...* sign in to your GitHub account

OR:

- Install **Git command line** via git-scm.com/downloads
- ```
  git config --global user.email "you@youremail.com"
  git config --global user.name "Your Name"
  ```
  ```

1. Create a repository, make some changes

2. Collaborate on a repository

Form groups of 2+

3. Make a release of your code

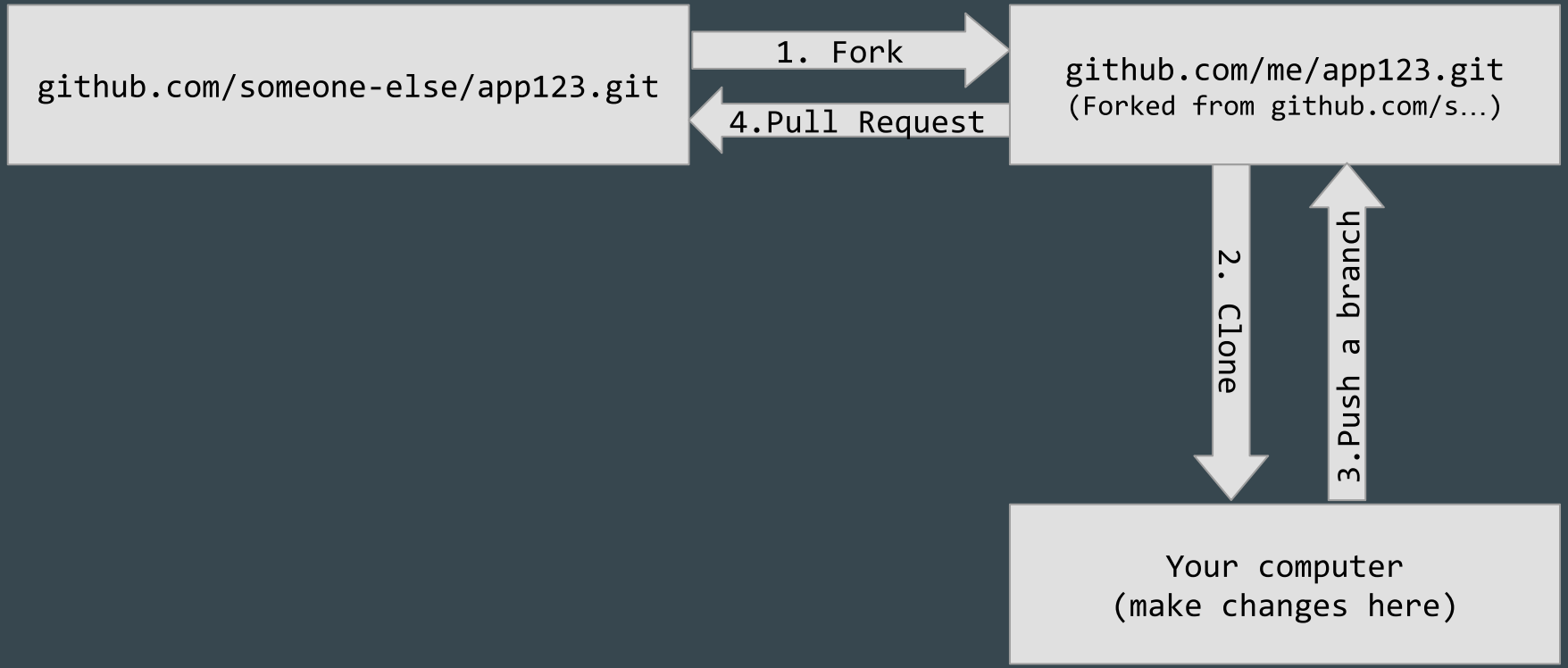
4. Fork a Repository

About forking...

Scenario: There is a Github repository on which you are *not* a collaborator, but you would like to:

- contribute a feature or fix to it, or
- use it, but with your own modifications

Forking a repository



Advanced Git Topics

- `.gitignore`
- Merges that require manual merging
- Resetting
- Re-basing
- Squashing
- Reverting

Git - Command Line - some common commands

```
git config --global user.email
git config --global user.password
git init <name>  # creates new repository in this folder
git clone <URL for repo>
git status       # -s gives a short version
git branch <branch-name>    # creates a branch
git checkout <branch-name>  # switches to a branch
git diff    /  git diff --staged
git add <file(s)>  /  git rm <file(s)>  /  git mv <file>
git commit -m 'Comment goes here'
git push origin <branch> # When working with forks, you might
git pull origin <branch> # use remotes other than "origin"
```

Git - Command Line - continued

```
git log
git merge <branch to merge from>
git rebase <branch to rebase off of>
git reset --hard  # Warning! You'll lose your changes!
git stash
git fetch   # like git pull, but without merging
```

Other Uses for Github Repositories (besides code)

- Sharing data
- "Document" collaboration (e.g. legislation)
- Course sites (e.g. DSCN 6279; ISTM 6212)
- Github Sites (e.g. gwu-libraries.github.io/sfm-ui/)
- Publishing and Citing your code (you can use Zenodo to mint a DOI for a Github repository)
- and more

Key takeaways

- Make your code presentable in GitHub:

  * README.md
  * LICENSE
  * Commit messages
  * Releases - release notes should describe new features/ changes, upgrade instructions, known issues
  * Use issues - helps explain WHY you changed code. Open issues communicate desired fixes/enhancements.

More takeaways

- The GitHub Desktop app seems useful, but mainly for "basic" things. Ultimately, it's probably good to learn and use git shell commands. (Especially good on your resume!)

- For team projects, one person can create the repository on their git account and add others as collaborators.
  Or, you can create an "organization" account - go to your profile's Settings -> Organizations -> New Organization

Even more takeaways

- When working together - or even alone - use branches. Merge a branch using a pull request.
- *Insights* → *Network* is a ~~good~~ great way to visualize what's going on with the branches.
- When you're working on a branch for an extended period, consider periodically merging (or rebasing) from the master branch, so your branch doesn't diverge too far from the code on master. This will minimize merge conflicts

To learn more about Git/GitHub...

- *Pro Git* book, free online: **git-scm.com**
- Lynda.com: *Up and Running with Git and GitHub*
- github.com:
  - help.github.com
  - guides.github.com
  - try.github.io (codeschool)
  - services.github.com/on-demand/
  - resources.github.com/webcasts/

Key take-home links

Github Guide to Making Your Code Citable:
guides.github.com/activities/citable-code/

Research Guide for Publishing a Scientific article:
libguides.gwu.edu/c.php?g=737748&p=5273908

Thanks!

- Dan Kerchner kerchner@gwu.edu
- Megan Potterbusch mpotterbusch@gwu.edu

Coding Consultations (GitHub, etc.) - schedule via:
go.gwu.edu/coding

Research Consultations (Data, etc.) - schedule with Megan via:
library.gwu.edu/reference/research-consultations