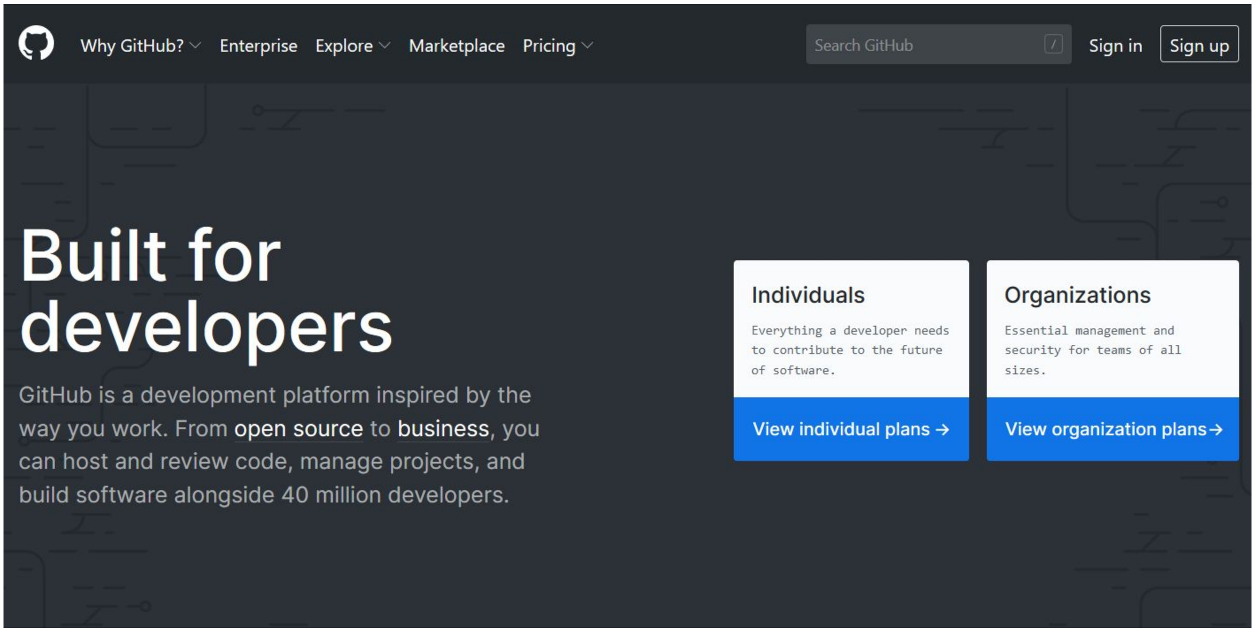
GitHub for Data and Methods Dissemination

Welcome to the GitHub primer. This document will show you how to put your data and analysis files onto GitHub so that they can be easily shared with the world. This demonstration is designed for people who want to use the web page interface to upload files to GitHub. That is probably the easiest for beginners. If you want to regularly use GitHub to share and disseminate data and code, it is worth the effort to learn how to use a more typical method, e.g. GitHub Desktop (https://desktop.github.com/), Git Bash (https://git-scm.com), commandline UI, or an IDE plugin (https://github.atom.io/).

There are three parts to the process: Creating a GitHub presence for your lab, creating a repository for a project, and uploading your software and data files.

Part 1: Creating a GitHub presence for your lab.

1. The first step is to go to https://github.com/ Sign in or sign up. When you sign up for the first time, you will be creating a personal account for you.



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 40 million developers.

Individuals

Everything a developer needs to contribute to the future of software.

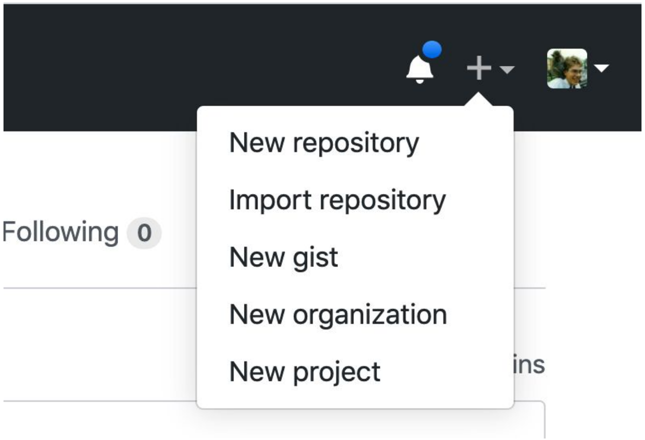
Organizations

Essential management and security for teams of all sizes.

2. Next, we recommend creating an “organization” account. This is different from your personal account. The organization account is helpful for lots of reasons. The two biggest are 1 - it keeps

all your projects together, and 2 - it has a flexible permission scheme built for fluid organizations like your lab. As the organizational lead, or PI, you can assign people to be admins on a rolling basis, which helps you control permission for students as they come into and then leave the lab.

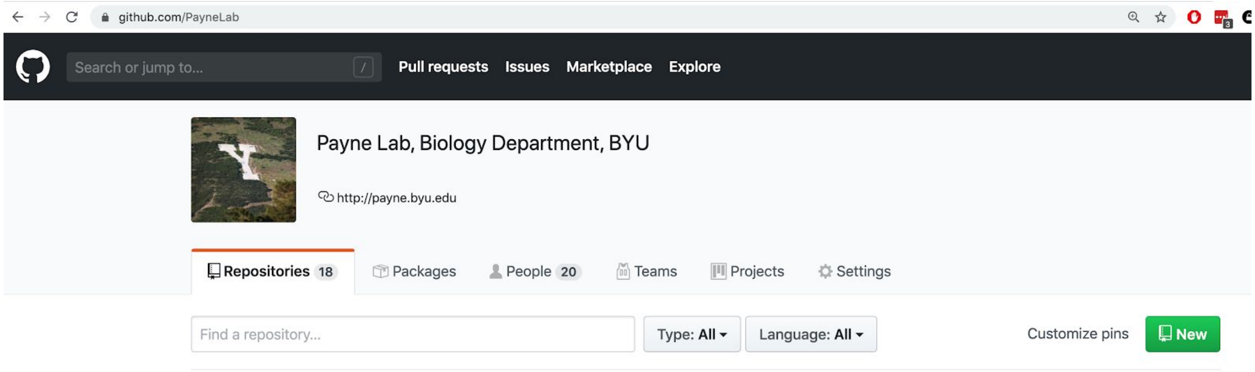
In the top right corner of your personal home page on GitHub, you'll see a + which has a "New organization".



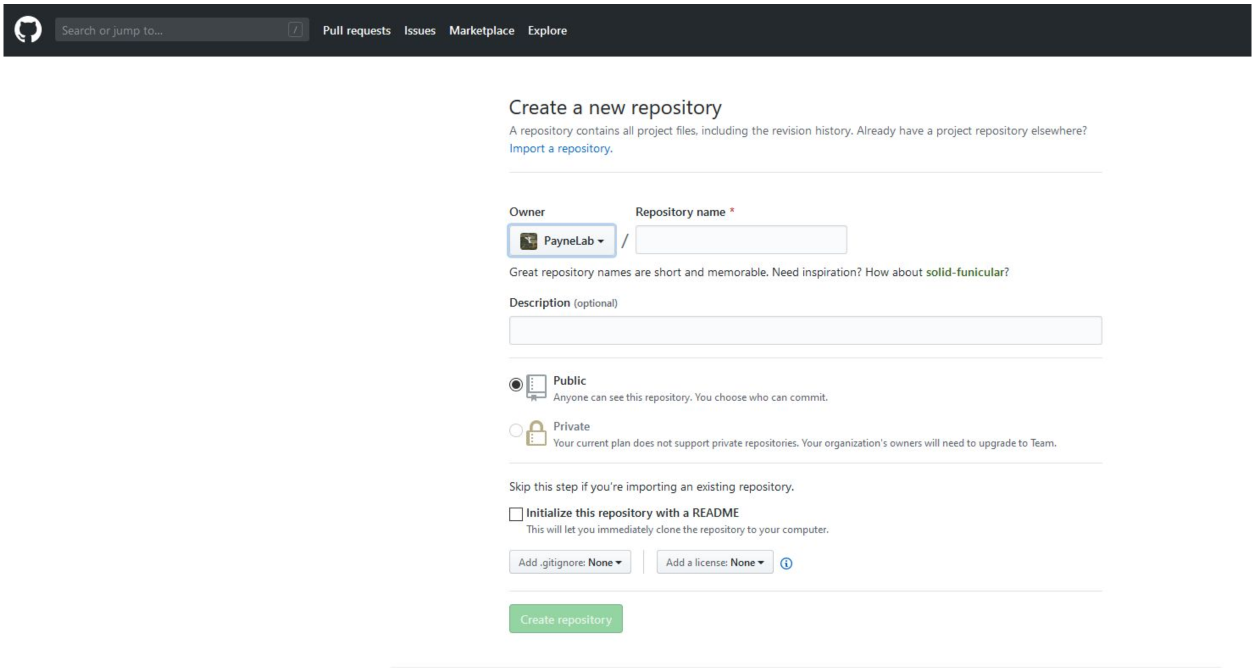
For more instructions on creating a team, visit: https://help.github.com/en/github/setting-up-and-managing-organizations-and-teams/creating-a-team

Part 2. Create a repository

1. Create a new repository. We recommend creating the repository under your lab organization. In the image below, you see the home page for the PayneLab organization, and on the right hand side is a green box labeled 'New'. This will create a new repository.



2. Add initial information about the repository.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner: PayneLab / Repository name *

Great repository names are short and memorable. Need inspiration? How about solid-funicular?

Description (optional)

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

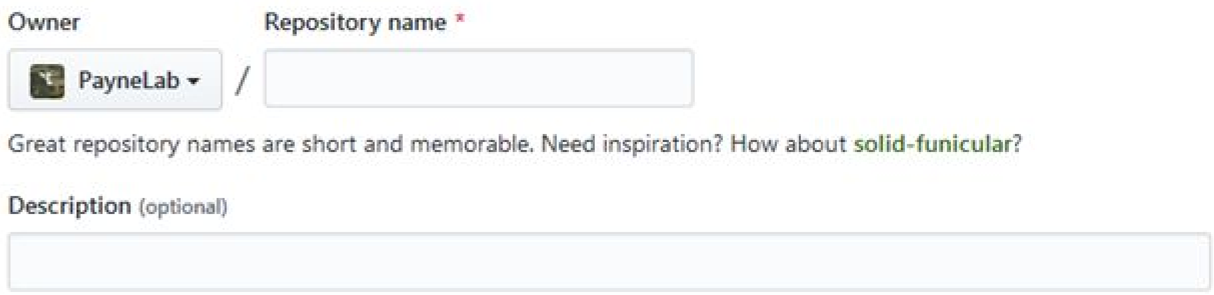
○ Private
Your current plan does not support private repositories. Your organization's owners will need to upgrade to Team.

Skip this step if you're importing an existing repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None

The first piece of information that they ask you for is the name of the repository. This name is generally brief, and must be unique from other project names. A more lengthy description may be provided if you want. By default, the owner of this repository is the organization.

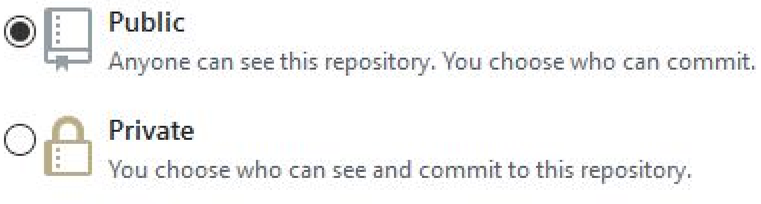


Owner: PayneLab / Repository name *

Great repository names are short and memorable. Need inspiration? How about solid-funicular?

Description (optional)

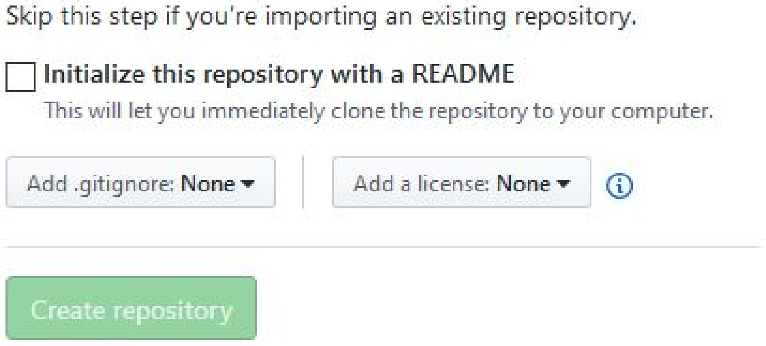
The next step is to choose whether the repository is public or private. For publication, this should be public.



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

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

There are three final options. If you want, you can totally ignore these for the moment. We'll get back to these later in Part 3. So go ahead and click the green button 'Create repository.'



Skip this step if you're importing an existing repository.

□ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

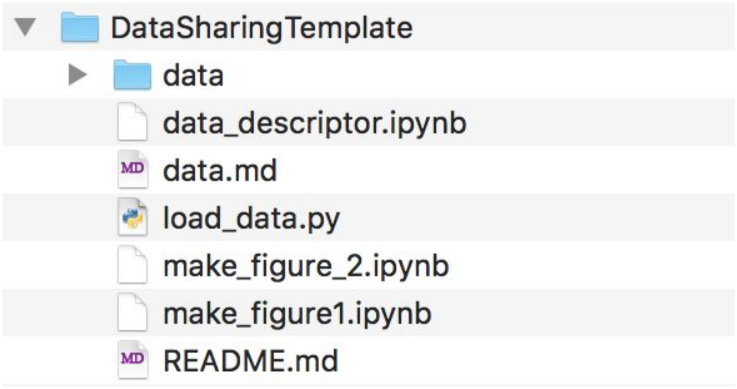
Add .gitignore: None ▾ | Add a license: None ▾

Create repository

Part 3. Upload your code

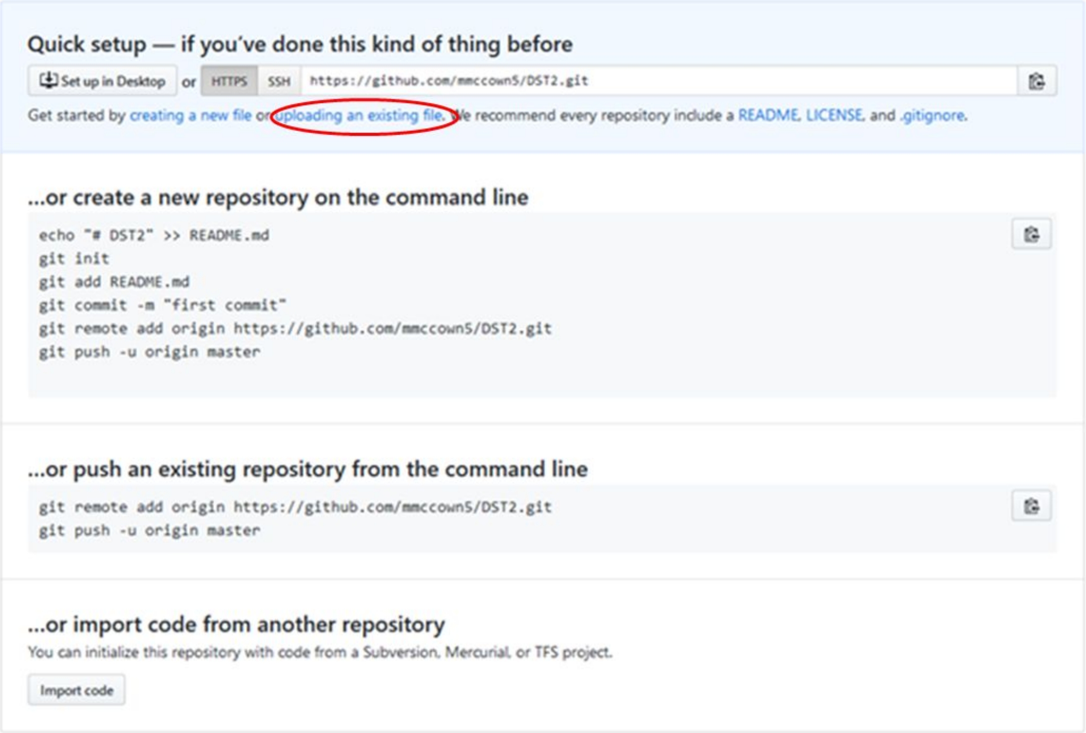
1. There are a lot of options for how to upload your code. This primer will just present the simplest, copying files from your computer to the repository. This is based on the assumption that you have organized your files according to the pattern suggested in the manuscript "Simple and Efficient Data Analysis Dissemination for Individual Labs."

Let's assume that you followed the pattern in the manuscript and have a project folder on your computer that looks like:



```
▼ DataSharingTemplate
    ▶ data
      data_descriptor.ipynb
      data.md
      load_data.py
      make_figure_2.ipynb
      make_figure1.ipynb
      README.md
```

Under Quick setup, follow the link for "uploading an existing file."



Quick setup — if you've done this kind of thing before

```
https://github.com/mmccown5/DST2.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# DST2" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/mmccown5/DST2.git
git push -u origin master
```

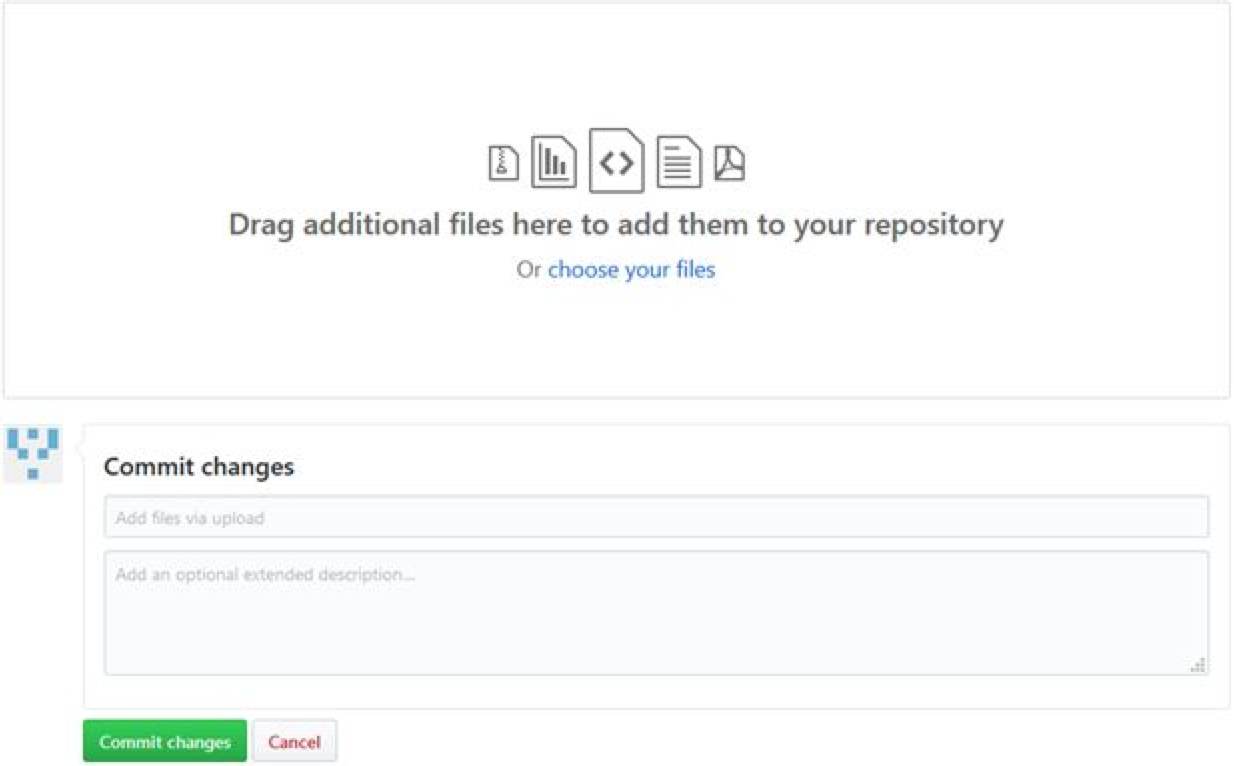
...or push an existing repository from the command line

```
git remote add origin https://github.com/mmccown5/DST2.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

This will give you the option (shown below) to drag and drop files into the repository. Drag and drop all of your desired files and folders. Note that you must use the drag and drop feature to upload entire folders (not choose your files feature).



Drag additional files here to add them to your repository

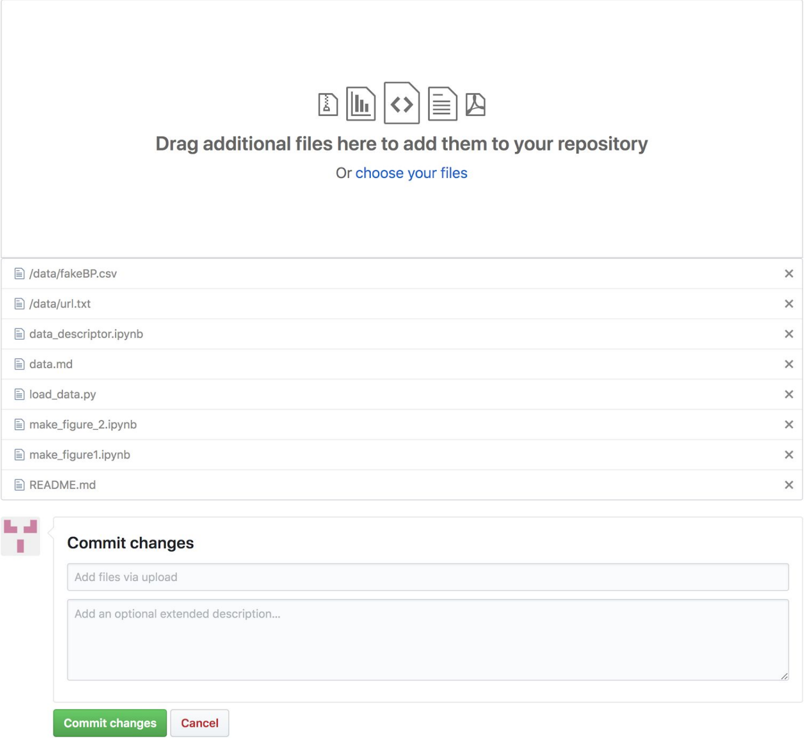
Or choose your files

Commit changes

Add files via upload

Add an optional extended description...

Once you have dropped your all the files and folders into the repository it will look like this:



Drag additional files here to add them to your repository

Or choose your files

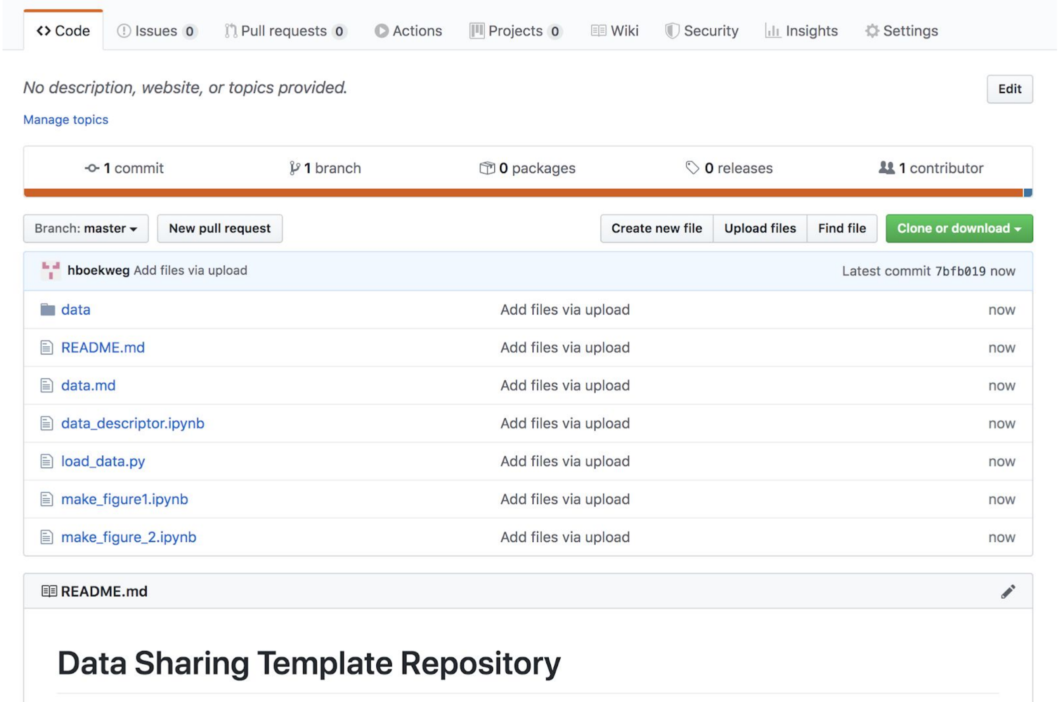
- /data/fakeBP.csv ×
- /data/url.txt ×
- data_descriptor.ipynb ×
- data.md ×
- load_data.py ×
- make_figure_2.ipynb ×
- make_figure1.ipynb ×
- README.md ×

Commit changes

Add files via upload

Add an optional extended description...

Click the "Commit changes" button, and it will upload the files to your repository, and look like this:



No description, website, or topics provided.

hboekweg Add files via upload — Latest commit 7bfb019 now

data	Add files via upload	now
README.md	Add files via upload	now
data.md	Add files via upload	now
data_descriptor.ipynb	Add files via upload	now
load_data.py	Add files via upload	now
make_figure1.ipynb	Add files via upload	now
make_figure_2.ipynb	Add files via upload	now

README.md

Data Sharing Template Repository

The repository is now complete. Provide the url in publications.