



# Workflows & Git/GitHub Basics

Justin Post

# What do we want to be able to do?

Data Science!

- Read in raw data and manipulate it
- Combine data sources
- Summarize data to glean insights
- Apply common analysis methods
- Communicate Effectively

# Workflow

Important considerations for data analysis workflow:

- **Reproducibility**
- **Version control**
- **Collaboration**

# What are git and github?

- **Git** is a version control software
- **Github** is an online hosting service for Git-based projects

# What are git and github?

- **Git** is a version control software
- **GitHub** is an online hosting service for Git-based projects

## Git Tracking

- You associate git with a folder (repo)
- Git keeps track of all files in the folder (repo)
- If you want to keep changes you've made, you **commit** and **push** the changes to the folder (repo)

# Github

- Github allows you to have a remote file repository (folder) tracked by git
  - Let's create a repository on github.com
  - **Add** some files and **commit** to the changes
  - Modify some files on github
  - Investigate the version control!

# Local vs Remote Work

Mostly you'll want to work on your local computer. Install `git` on your computer!

# Local vs Remote Work

Mostly you'll want to work on your local computer. Install `git` on your computer!

## Workflow

1. (Initially) **clone** the repo locally. (Later) **pull** to get most recent versions of files
2. Work and make changes
3. **add** and **commit** to changes you like
4. **push** changes to remote repo (on github)

Let's clone our repo and work on it locally!



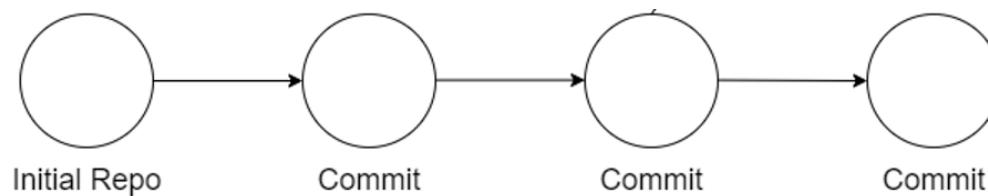
# Git & RStudio

Git and RStudio work great together!

- Works through **R Projects**
- Start a new project from git repo
  - Update with command line or git menu!

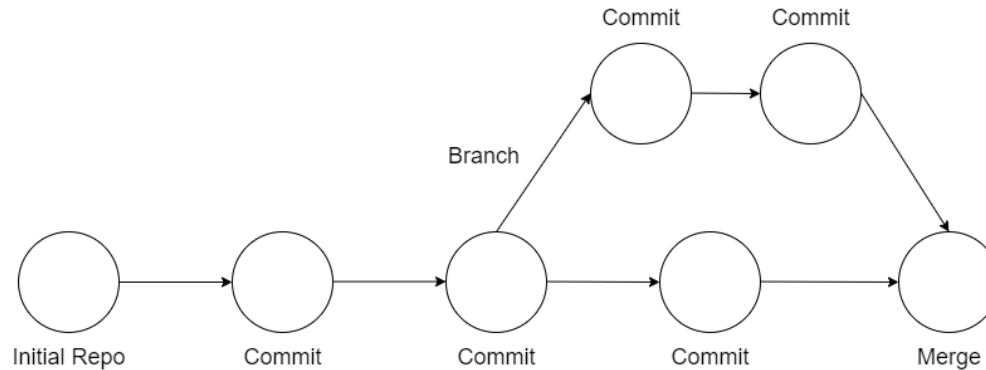
# Collaboration Idea

- Everyone can work on the same **branch** and just take turns working



# Working on Branches

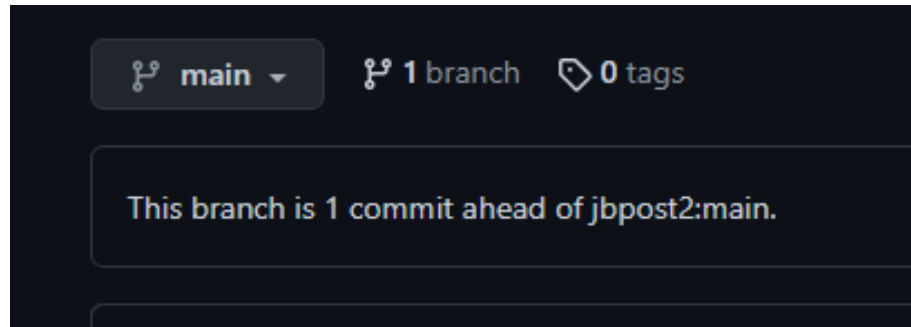
- Alternatively, you can have separate **branches** of the repo



- Work on a branch is similar to working on the main branch
- Can merge when happy!

# Forking

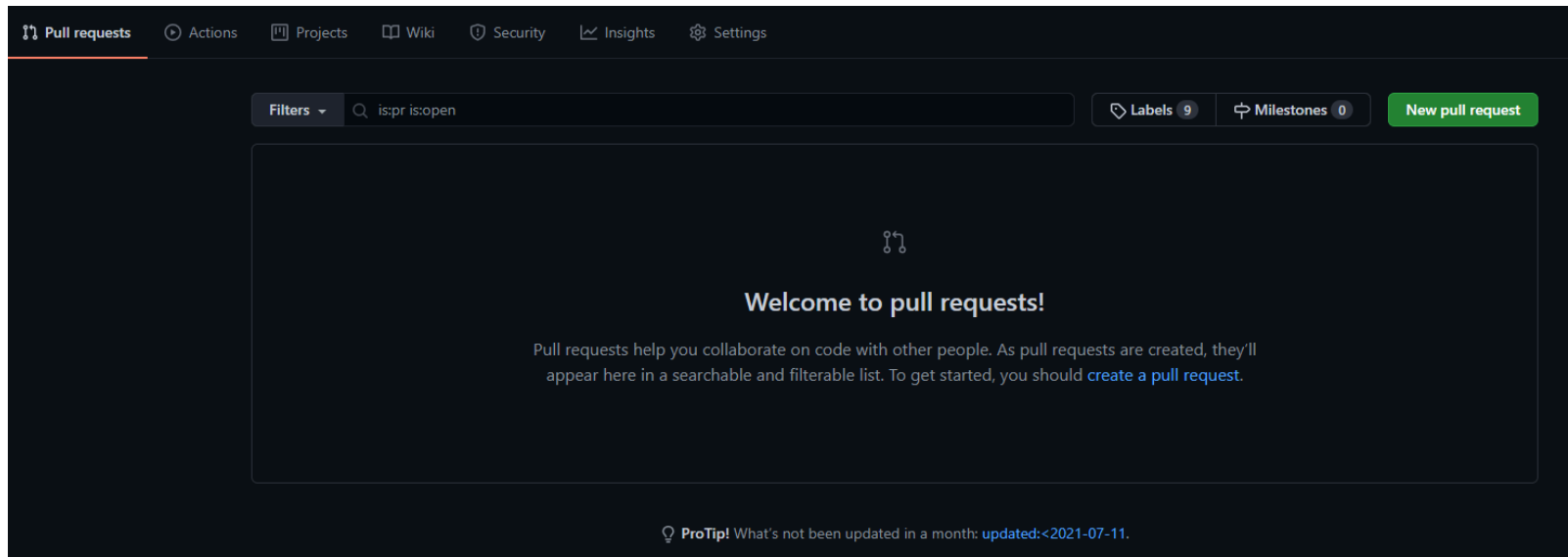
- People often `fork` the repo
- This creates a copy of the repo on your account
- You can then work as normal
- If you do a commit on your branch, you may notice something like this



# Merging Branches

Suppose you like your commit and you think the original owner will too!

- You can do a **pull** request



# Merging Branches

Suppose you like your commit and you think the original owner will too!

- You can do a **pull** request

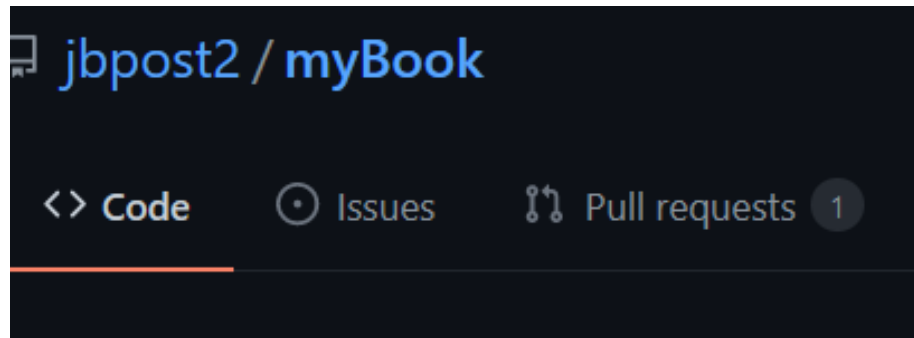
The screenshot shows the GitHub 'Comparing changes' interface. At the top, it says 'Comparing changes' and 'Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.' Below this, there are dropdown menus for 'base repository: jbpost2/myBook', 'base: main', 'head repository: JustinBPost/myBook', and 'compare: main'. A green checkmark indicates 'Able to merge. These branches can be automatically merged.' A 'Create pull request' button is visible. Below the button, it shows '1 commit', '1 file changed', '0 comments', and '1 contributor'. The commit history shows 'Commits on Aug 12, 2021' and 'Update 05-summary.Rmd' with a 'Verified' badge and commit hash '2f7ad8f'. The file diff shows 'Showing 1 changed file with 2 additions and 0 deletions.' The diff content is as follows:

```
2 05-summary.Rmd
@@ -1,3 +1,5 @@
1 1 # Final Words
2 2
3 3 We have finished a nice book.
4 +
5 + Or did we?
```

# Merging Branches

If you are lucky, there won't be any merge conflicts.

- Allows the owner of the original repo to accept the pull request without needing to modify things
- The owner will get a notification that a pull request has been made



# Merging Branches

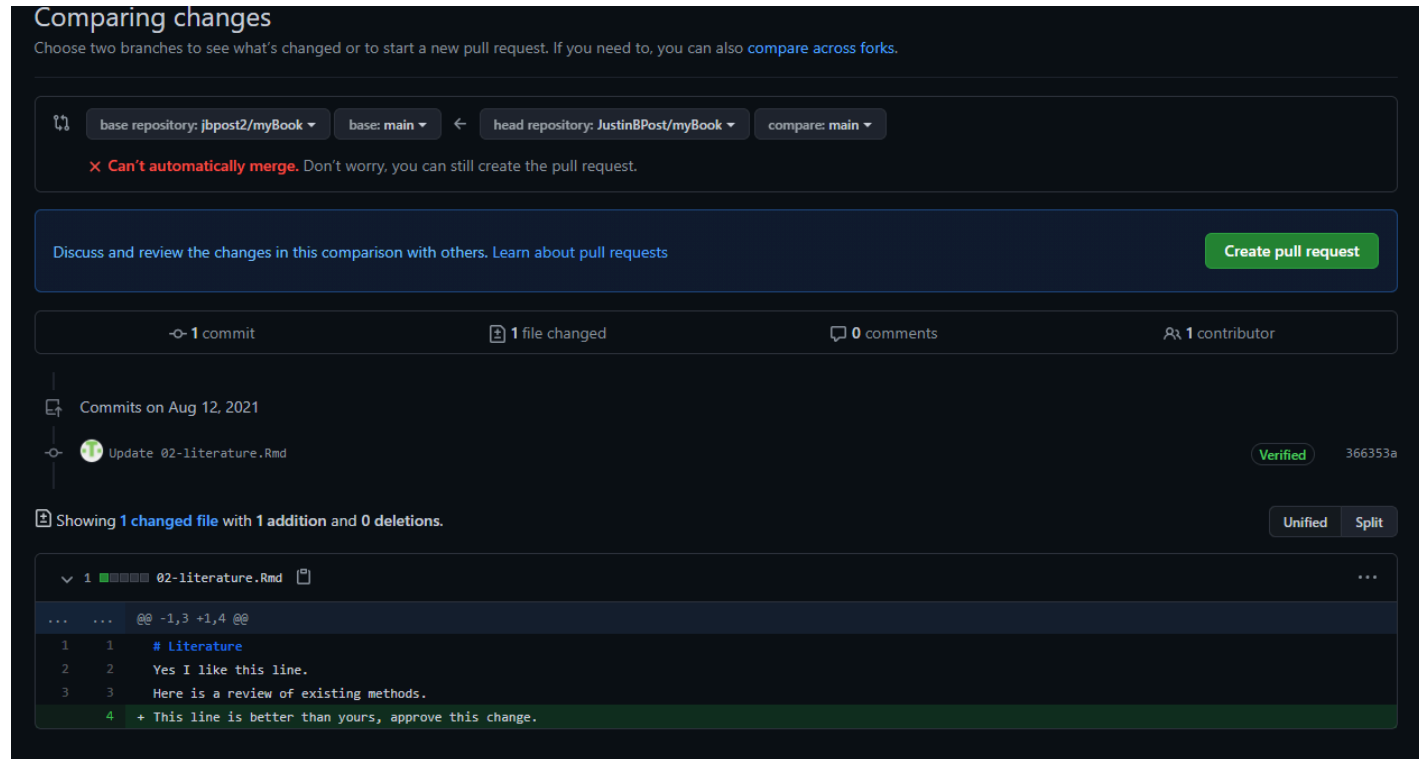
Owner can then investigate the request and choose whether or not to accept it or they can ask for more details

The screenshot shows a GitHub pull request interface. At the top, the title is "Update 05-summary.Rmd #1". Below the title, there is a green "Open" button and a status bar indicating "JustinBPost wants to merge 1 commit into jbpost2:main from JustinBPost:main". A navigation bar shows "Conversation 0", "Commits 1", "Checks 0", and "Files changed 1". A comment from JustinBPost, posted 4 minutes ago, states "No description provided." Below the comment, a commit titled "Update 05-summary.Rmd" is shown with a "Verified" badge and the hash "2f7ad8f". A message below the commit reads: "Add more commits by pushing to the main branch on JustinBPost/myBook." At the bottom, there are two status messages: "Continuous integration has not been set up" (with a warning icon) and "This branch has no conflicts with the base branch" (with a green checkmark icon). The bottom of the interface features a green "Merge pull request" button and a note: "You can also open this in GitHub Desktop or view command line instructions."



# Dealing with conflicts

- Sometimes changes requested conflict with changes already made



Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base repository: jbpost2/myBook base: main ← head repository: JustinBPost/myBook compare: main

✗ **Can't automatically merge.** Don't worry, you can still create the pull request.

Discuss and review the changes in this comparison with others. [Learn about pull requests](#) [Create pull request](#)

→ 1 commit 1 file changed 0 comments 1 contributor

Commits on Aug 12, 2021

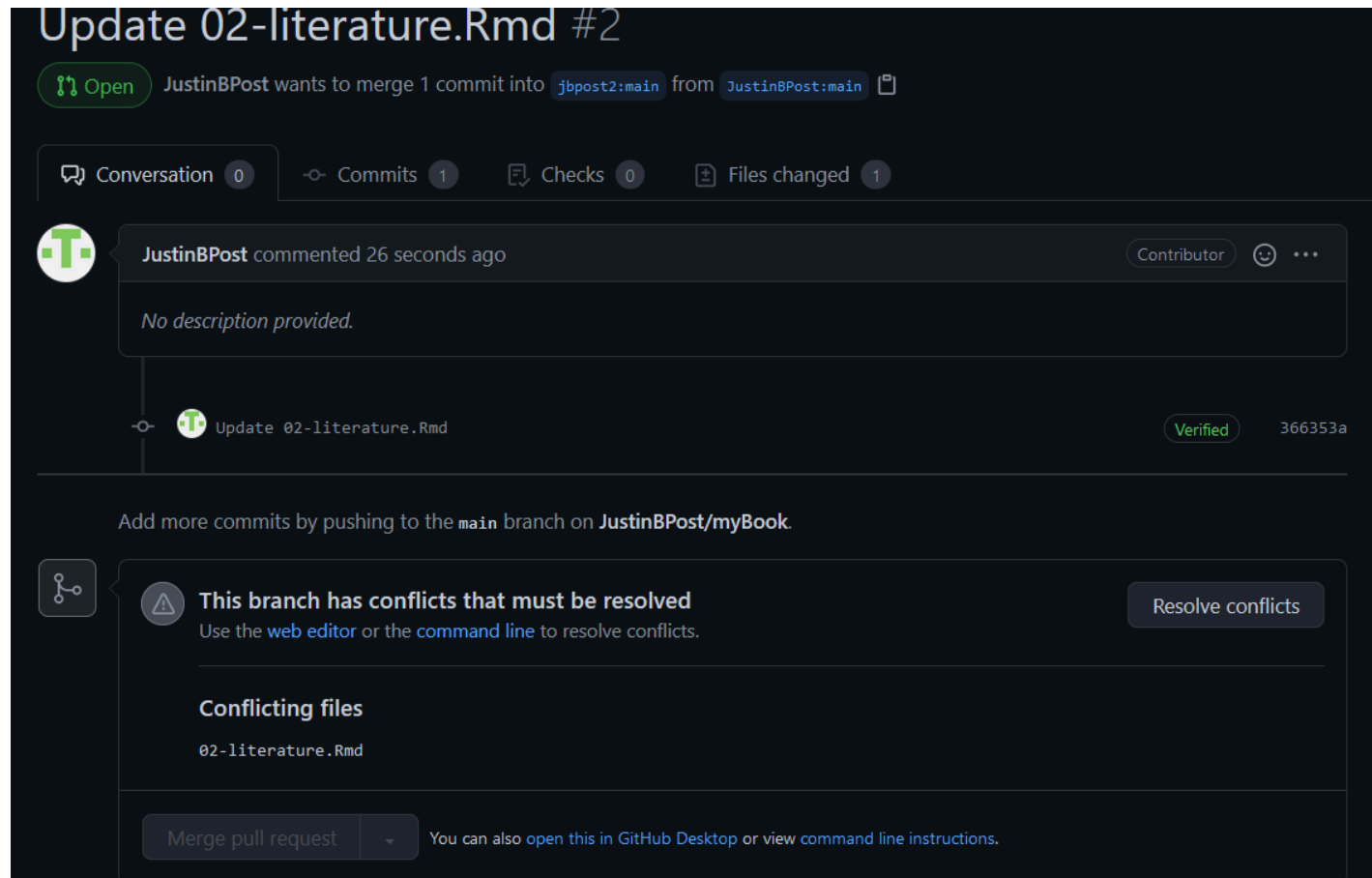
Update 02-literature.Rmd Verified 366353a

Showing 1 changed file with 1 addition and 0 deletions. [Unified](#) [Split](#)

```
1 # Literature
2 Yes I like this line.
3 Here is a review of existing methods.
4 + This line is better than yours, approve this change.
```

# Dealing with conflicts

Owner sees a notification about conflicts that must be resolved

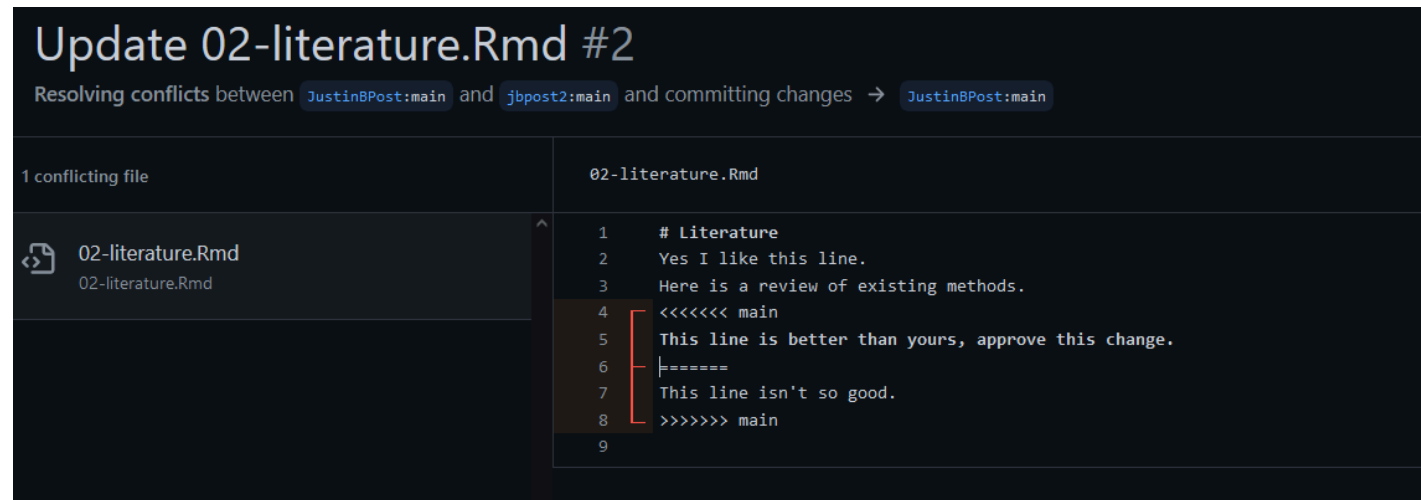


The screenshot shows a GitHub pull request titled "Update 02-literature.Rmd #2". At the top, it indicates that JustinBPost wants to merge 1 commit into the `jbpost2:main` branch from the `JustinBPost:main` branch. Below this, there are statistics for the pull request: Conversation (0), Commits (1), Checks (0), and Files changed (1). A comment from JustinBPost, posted 26 seconds ago, states "No description provided." Below the comment, a commit is shown: "Update 02-literature.Rmd" by JustinBPost, which is marked as "Verified" with the commit hash `366353a`. At the bottom of the pull request, a warning message states: "This branch has conflicts that must be resolved. Use the [web editor](#) or the [command line](#) to resolve conflicts." A "Resolve conflicts" button is provided next to this message. Below the warning, a section titled "Conflicting files" lists `02-literature.Rmd` as the file with conflicts. At the very bottom, there is a "Merge pull request" button and a note: "You can also [open this in GitHub Desktop](#) or [view command line instructions](#)."

# Dealing with conflicts

They can view the issues and pick which to include or to include both with a modification

<<<<<<< is a conflict marker



The screenshot shows a code editor window titled "Update 02-literature.Rmd #2". Below the title, it says "Resolving conflicts between JustinBPost:main and jbpost2:main and committing changes → JustinBPost:main". On the left, a sidebar shows "1 conflicting file" and a file named "02-literature.Rmd". The main editor area shows the content of "02-literature.Rmd" with line numbers 1 through 9. The content is as follows:

```
1 # Literature
2 Yes I like this line.
3 Here is a review of existing methods.
4 <<<<<< main
5 This line is better than yours, approve this change.
6 |=====
7 This line isn't so good.
8 >>>>>> main
9
```

- Figure out what to do and delete the <<< === >>> lines

# Recap

- **Git** is a version control software
  - Associated with a folder (repo)
  - Tracks changes to files
- **Github** is an online hosting service for Git-based projects
- Workflow:
  - Pull down most recent files (`git pull`) or do initial download (`git clone`)
  - Add files you want to keep changes to (`git add`)
  - Commit to the changes (`git commit`)
  - Push the changes to the remote repo (`git push`)