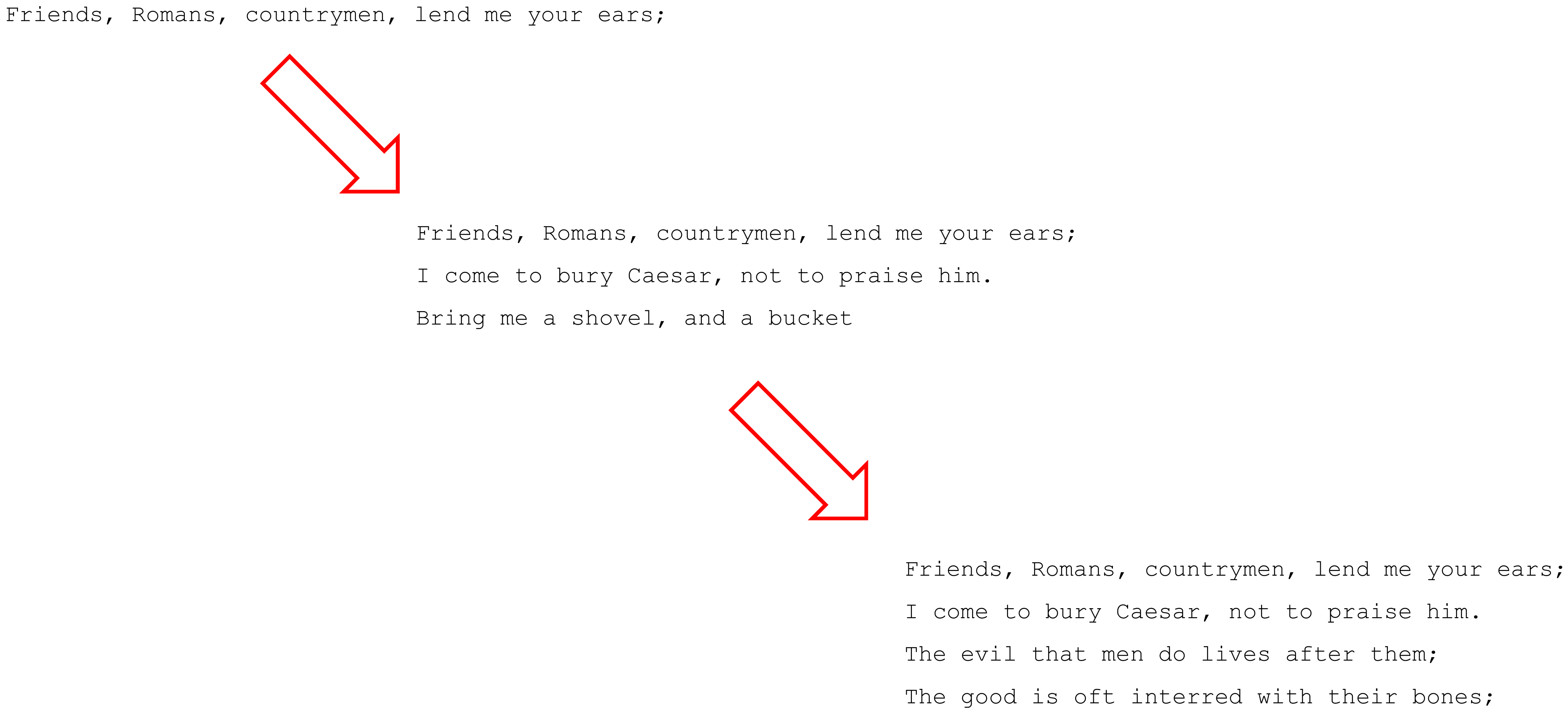
Introducing version control using Git, GitHub and Rstudio

v2020 11

Babraham Bioinformatics

Version Control



Friends, Romans, countrymen, lend me your ears;

Friends, Romans, countrymen, lend me your ears;
I come to bury Caesar, not to praise him.
Bring me a shovel, and a bucket

Friends, Romans, countrymen, lend me your ears;
I come to bury Caesar, not to praise him.
The evil that men do lives after them;
The good is oft interred with their bones;

Progressive Saving Options

- Save and overwrite
  - Only have the latest version
  - Can’t backtrack

```
caesar.txt
```

- Save to new files
  - Full history
  - Lots of duplication
  - No partial backtracking

```
caesar_1.txt
caesar_2.txt
caesar_3.txt
```

Version Control

Store differences to the last version

Version 1

```
+ Friends, Romans, countrymen, lend me your ears;
```

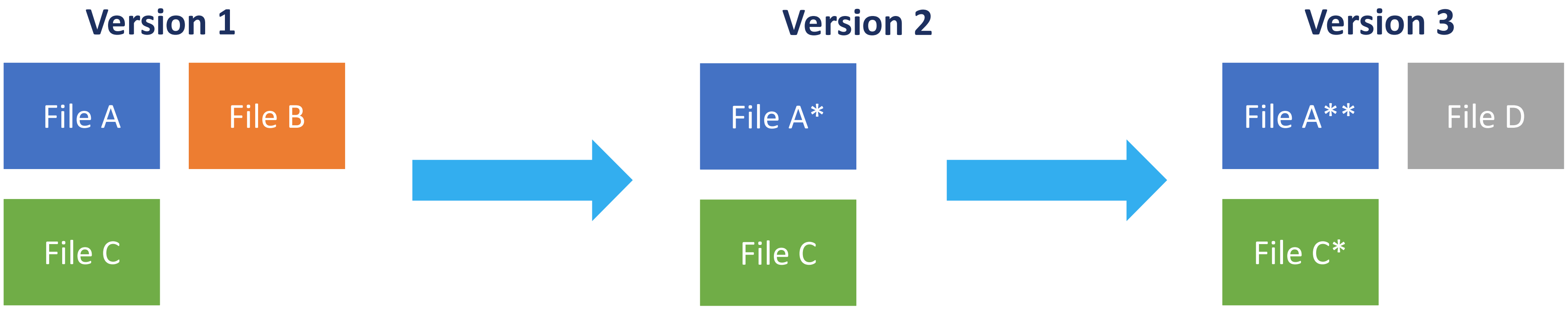
Version 2

```
+ I come to bury Caesar, not to praise him.
+ Bring me a shovel, and a bucket
```

Version 3

```
- Bring me a shovel, and a bucket
+ The evil that men do lives after them;
+ The good is oft interred with their bones;
```

Managing a project is difficult and requires version control



- Files edited
- Files added
- Files removed
- How to best manage this?

- Version control records each version of the project
- Previous versions can be accessed as and when required

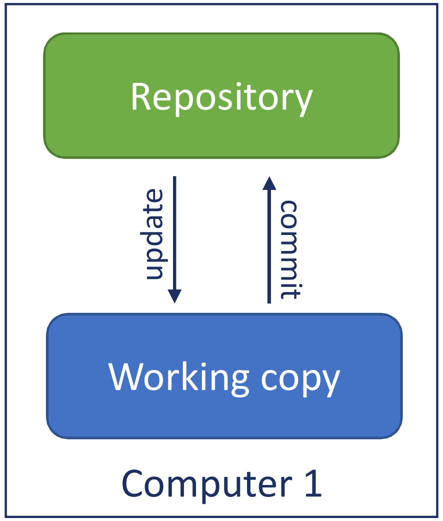
Version control is also useful because…

- It is a great way to keep track of how and why scripts were modified
- Back-up capabilities
- Collaboration
- Sharing

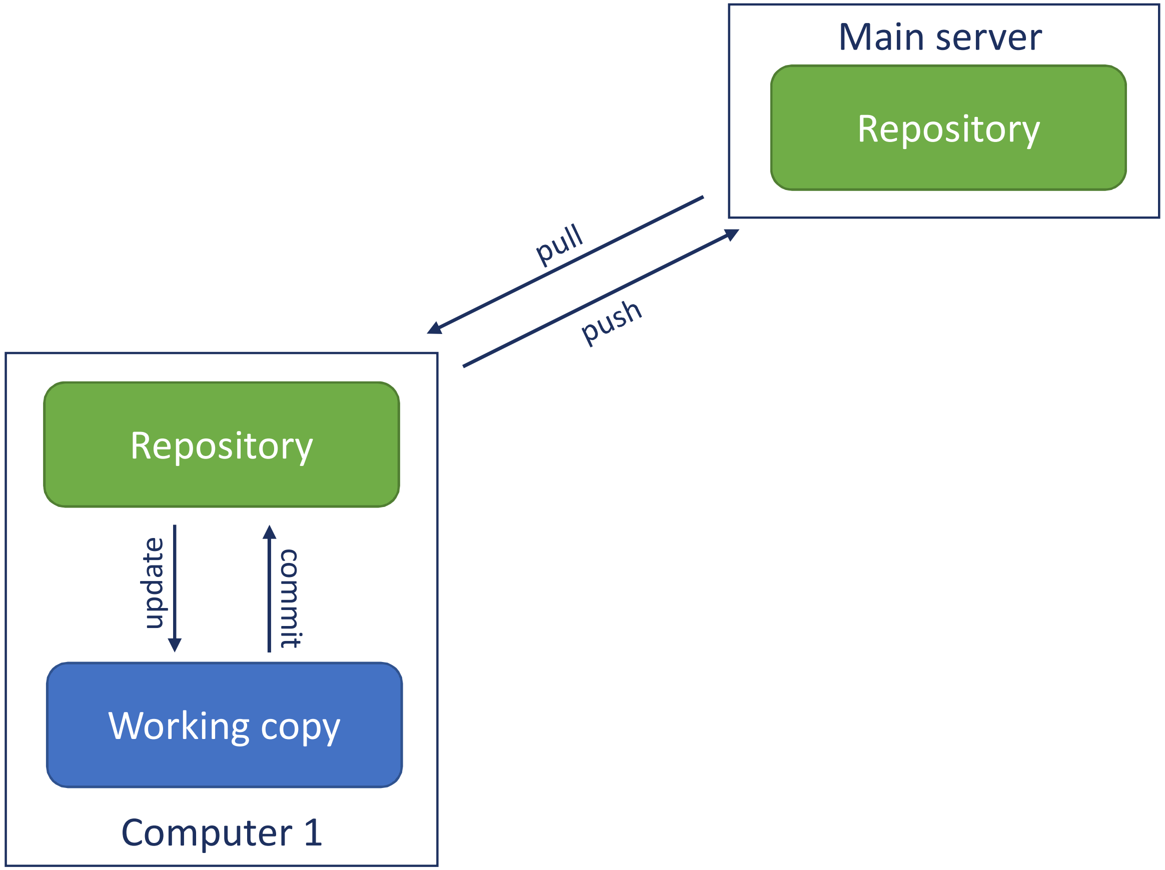
Distributed version control

Working copy
Computer 1

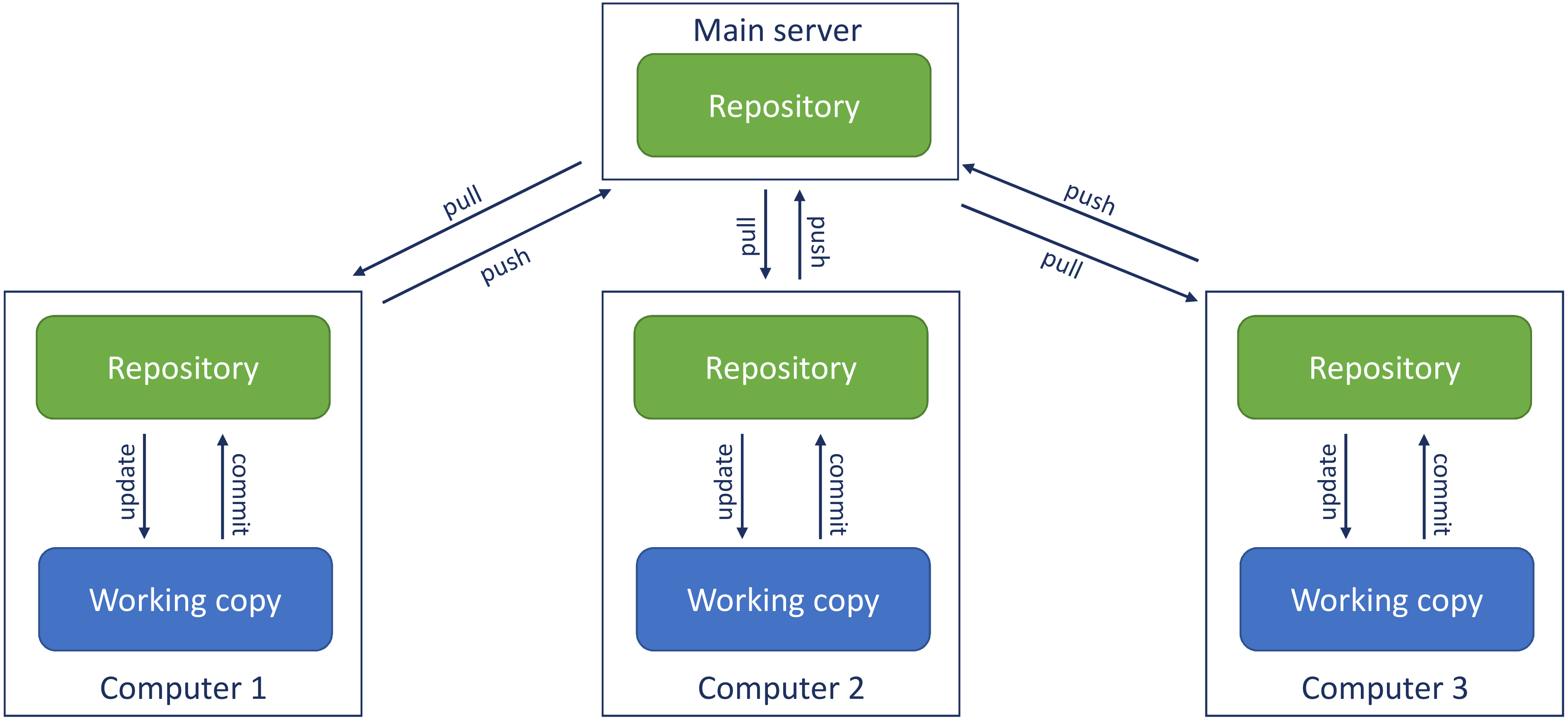
Distributed version control



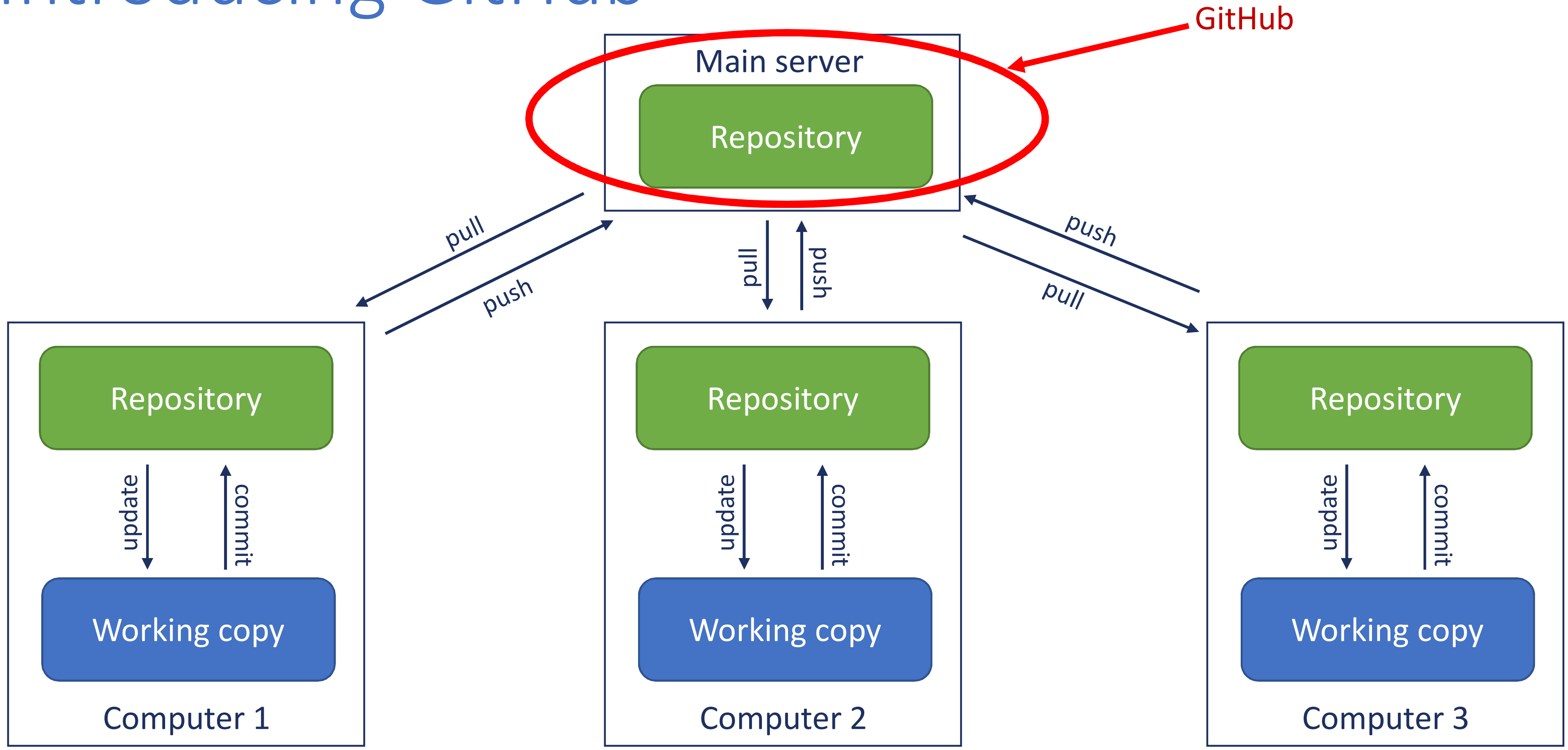
Distributed version control



Distributed version control

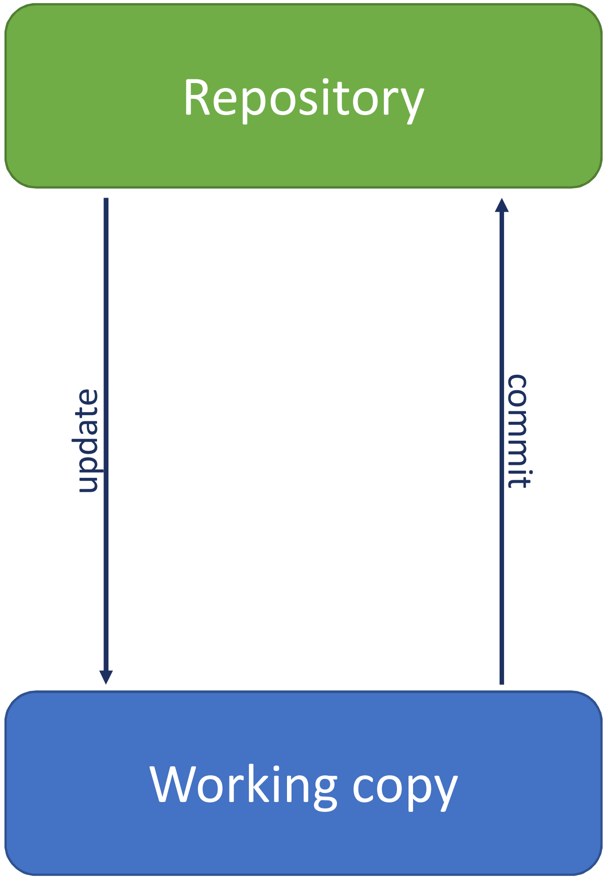


Introducing GitHub



GitHub

Committing files to a Git repository



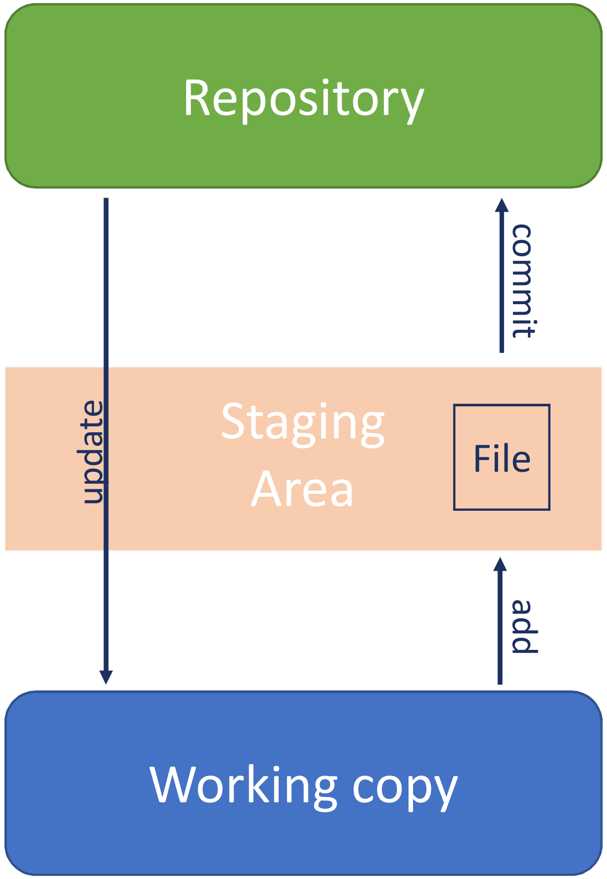
Repository

update

commit

Working copy

Committing files to a Git repository

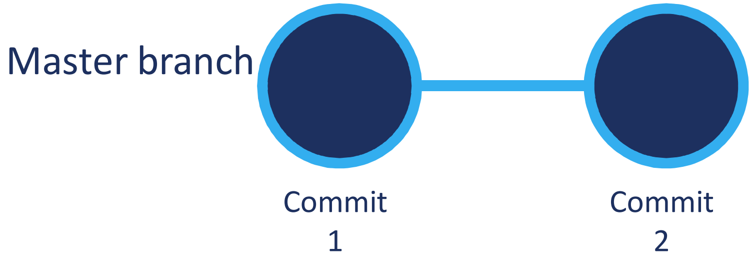


Branches

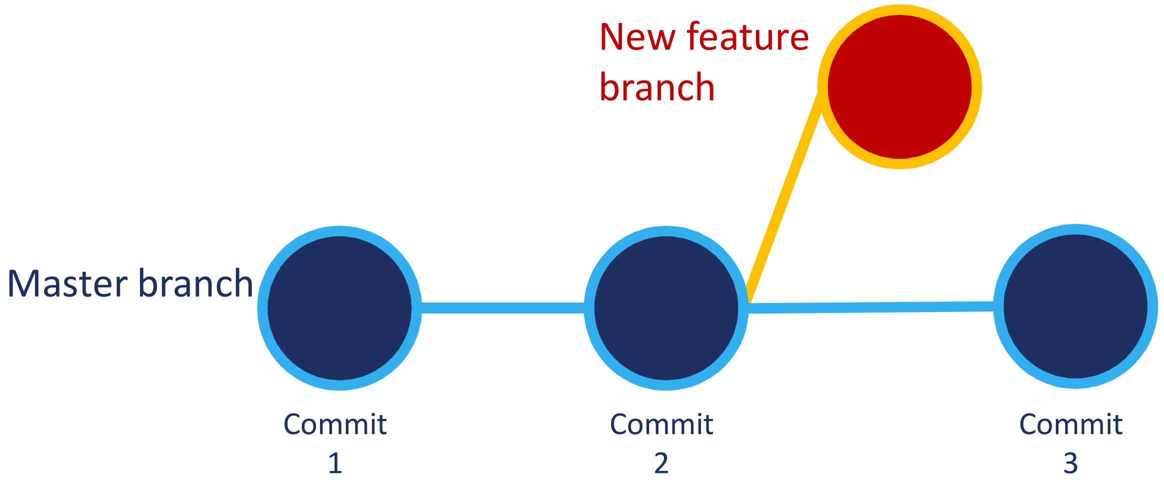
Master branch

Commit 1

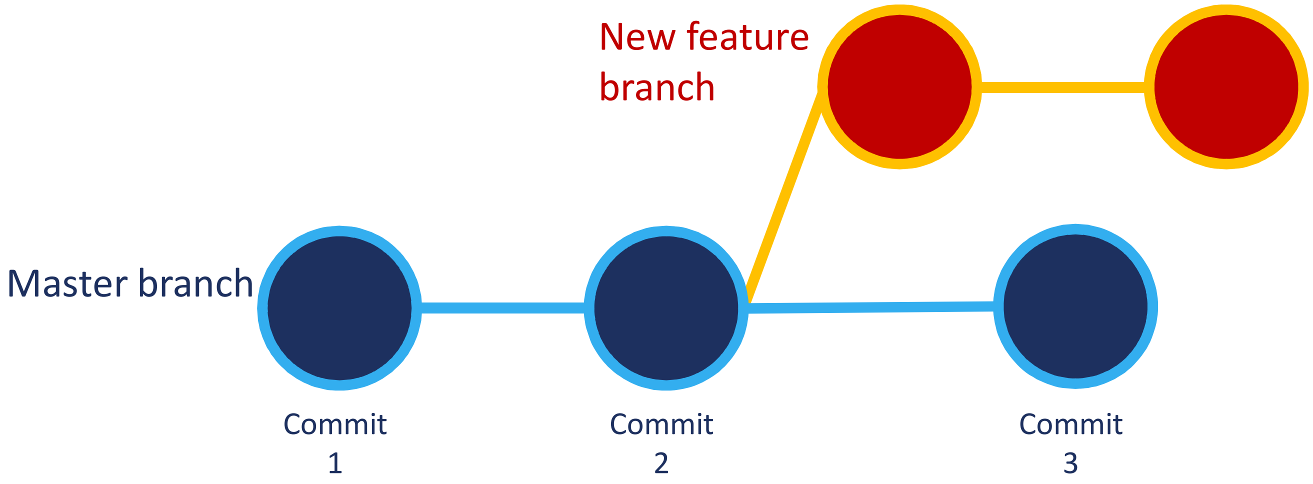
Branches



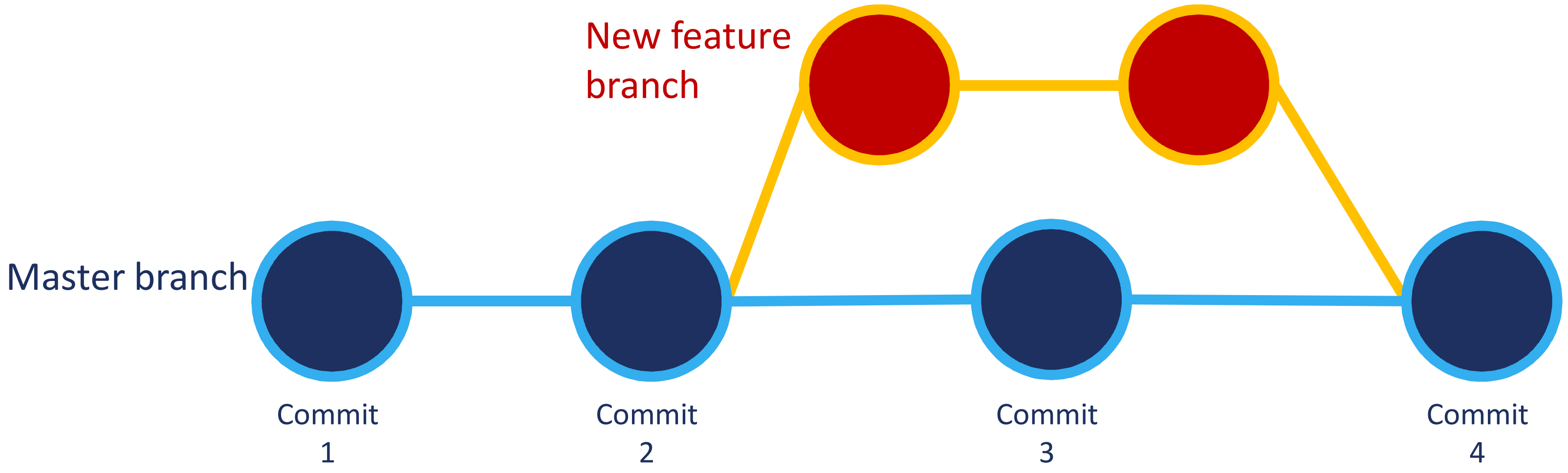
Branches



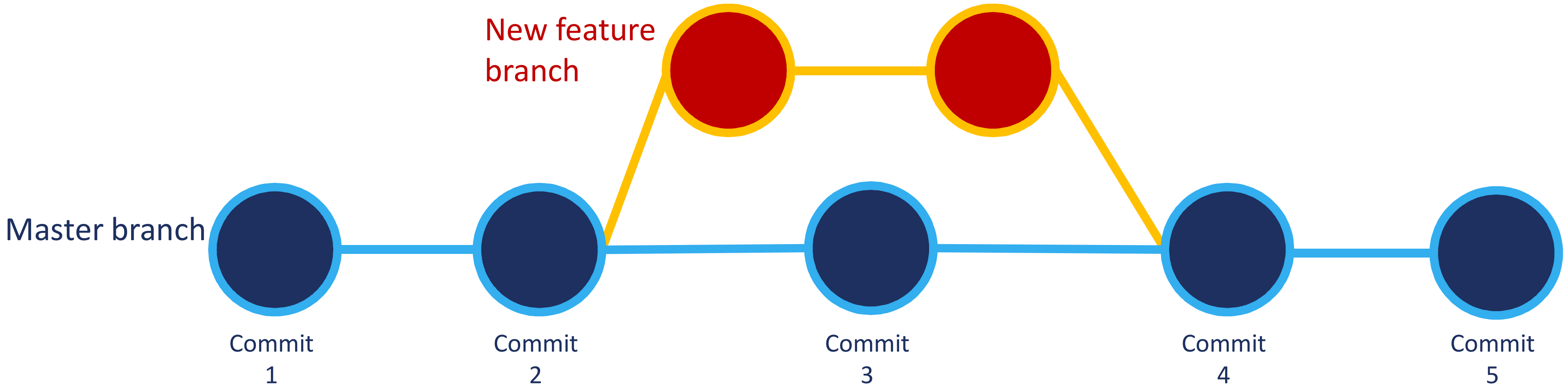
Branches



Branches


New feature branch
Master branch
Commit 1
Commit 2
Commit 3
Commit 4

Branches


New feature branch
Master branch
Commit 1
Commit 2
Commit 3
Commit 4
Commit 5

Checking out a previous version

Requires use of the terminal in RStudio

```
git checkout [revision] .
```

where [revision] is the commit hash, for example:

```
git checkout 668c6a44383e7381d0417143da12804f1ae8fb66 .
```

The . at the end means that all the files will be checked out.

To only checkout one file from a previous version:

```
git checkout [revision] [file_name]
```

e.g.

```
git checkout 668c6a44383e7381d0417143da12804f1ae8fb66 scatterplot.R
```

Other resources

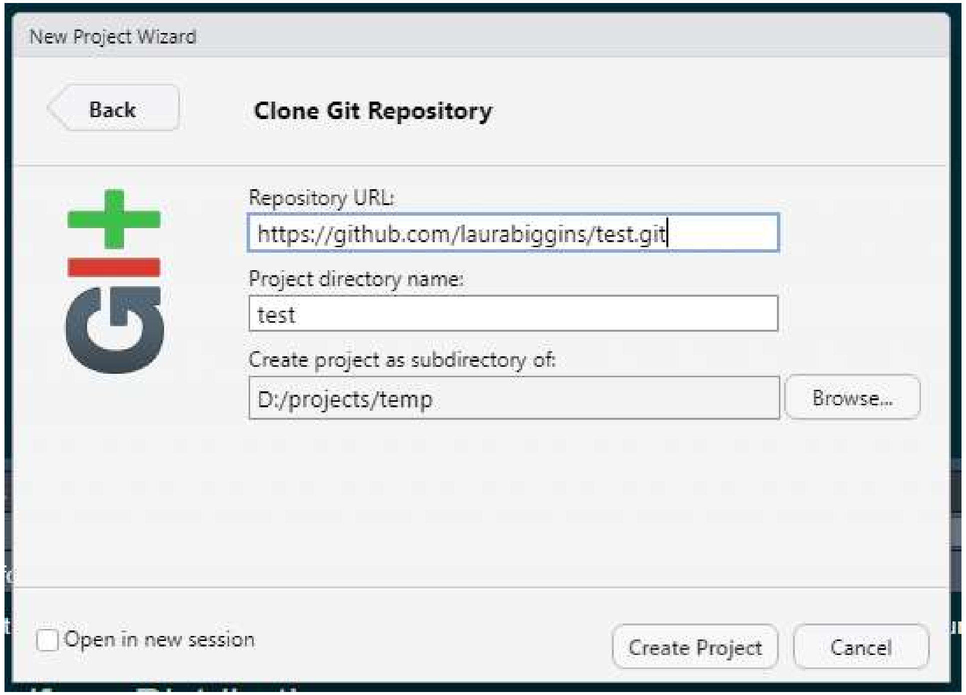
https://happygitwithr.com/

https://rstudio.com/resources/webinars/managing-part-2-github-and-rstudio/

https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN

Cloning/Importing repository from GitHub into RStudio

New project -> Version Control -> Git
Paste in URL from GitHub

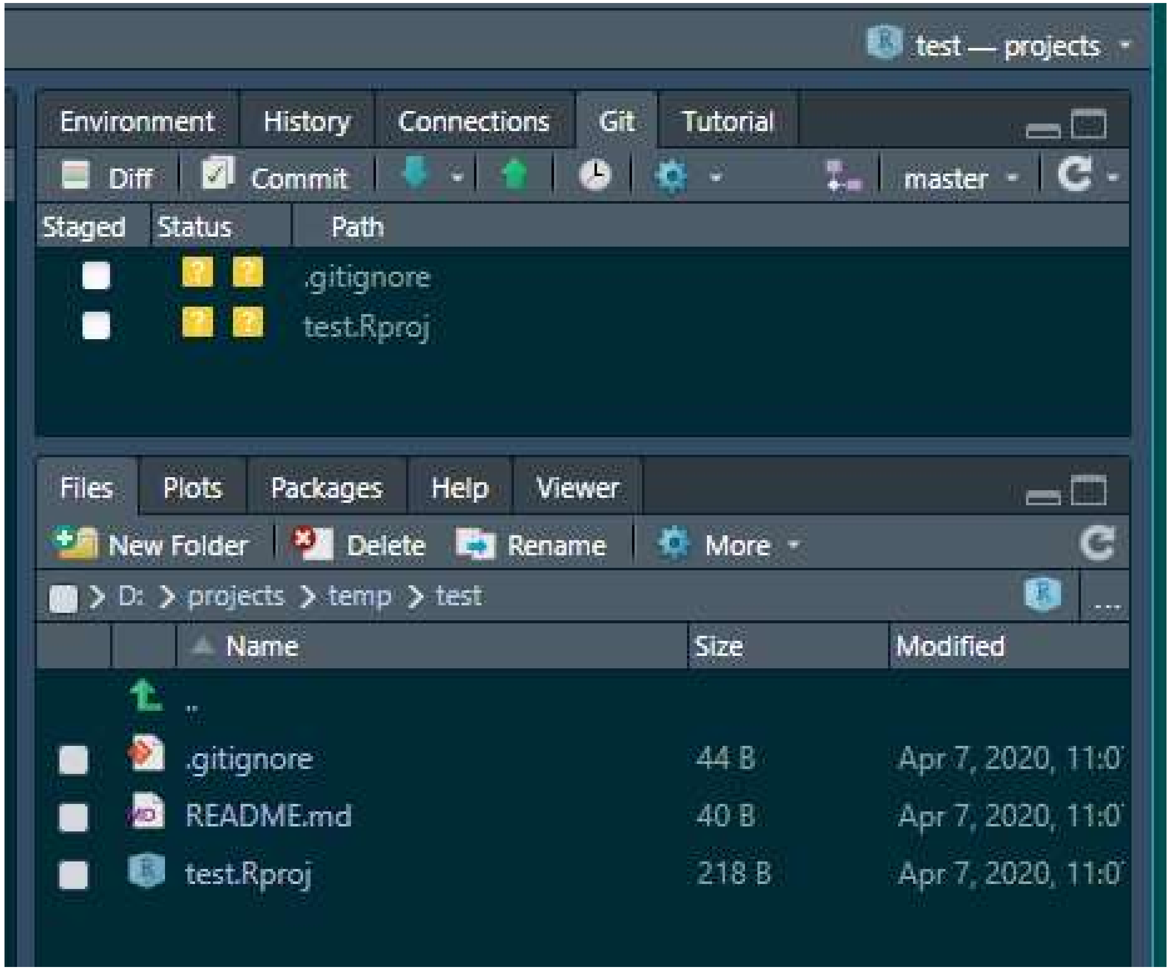


Repository URL: https://github.com/laurabiggins/test.git

Project directory name: test

Create project as subdirectory of: D:/projects/temp

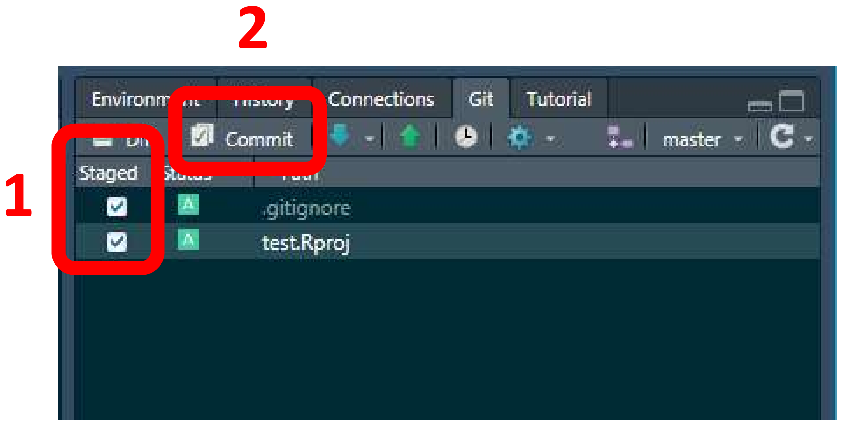
☐ Open in new session



Staged	Status	Path
□	? ?	.gitignore
□	? ?	test.Rproj

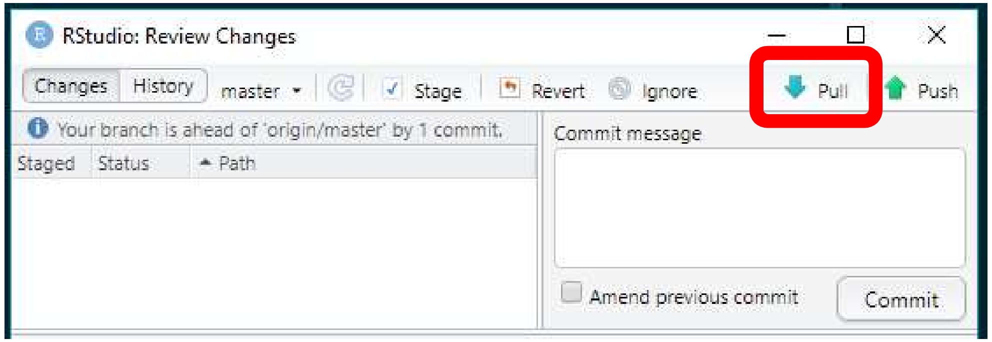
	Name	Size	Modified
	..		
□	.gitignore	44 B	Apr 7, 2020, 11:0
□	README.md	40 B	Apr 7, 2020, 11:0
□	test.Rproj	218 B	Apr 7, 2020, 11:0

In creating the new Rproject, a test.Rproj file was created, along with a .gitignore file.



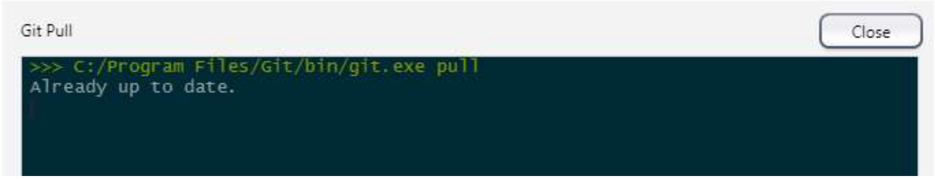
To stage the files – select the tick boxes under *Staged*

Click on Commit



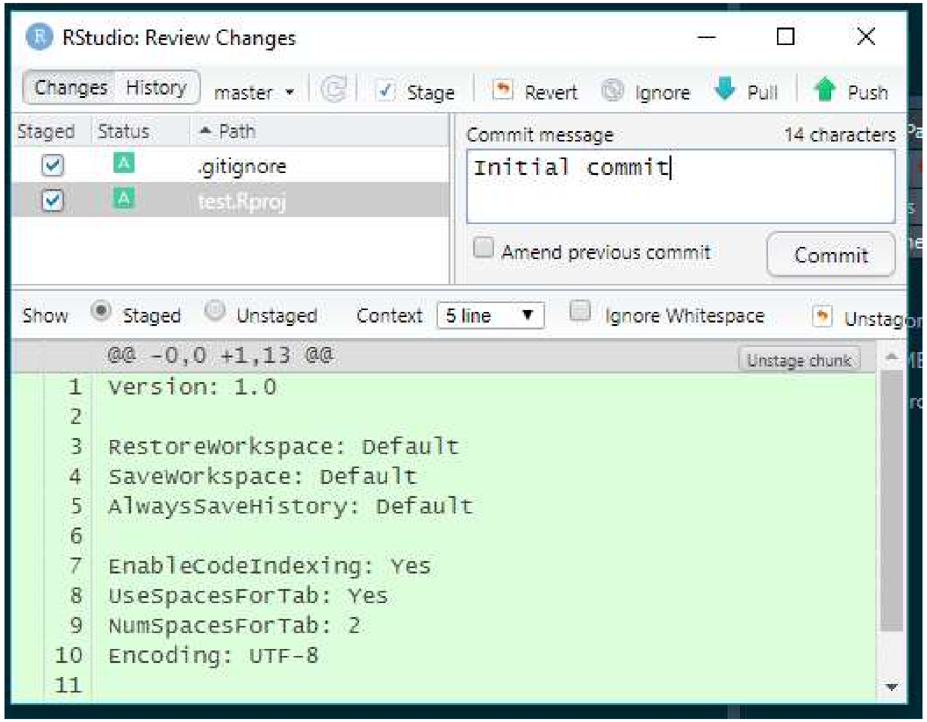
Your branch is ahead of 'origin/master' by 1 commit.

Always a good idea to do a Git pull



```
>>> C:/Program Files/Git/bin/git.exe pull
Already up to date.
```

Should see an information message saying that it’s already up to date



Staged	Status	Path
☑	A	.gitignore
☑	A	test.Rproj

Commit message: Initial commit

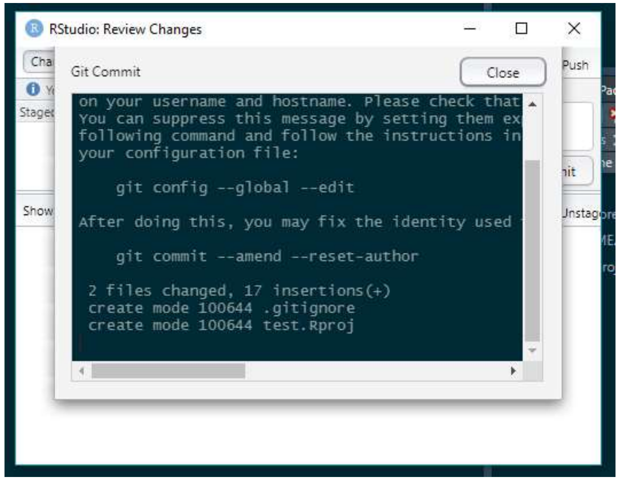
14 characters

☐ Amend previous commit

Show ● Staged ○ Unstaged Context 5 line ☐ Ignore Whitespace

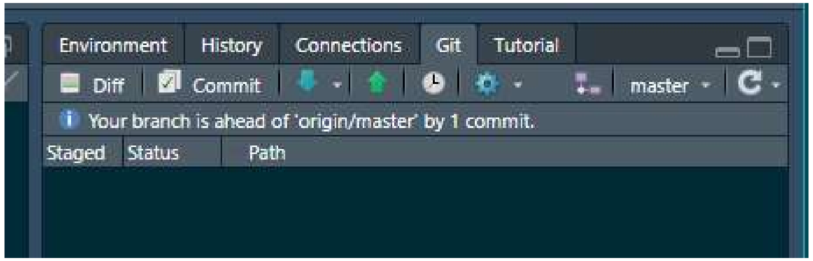
```
@@ -0,0 +1,13 @@
Version: 1.0

RestoreWorkspace: Default
SaveWorkspace: Default
AlwaysSaveHistory: Default

EnableCodeIndexing: Yes
UseSpacesForTab: Yes
NumSpacesForTab: 2
Encoding: UTF-8

```

Add a commit message and select commit



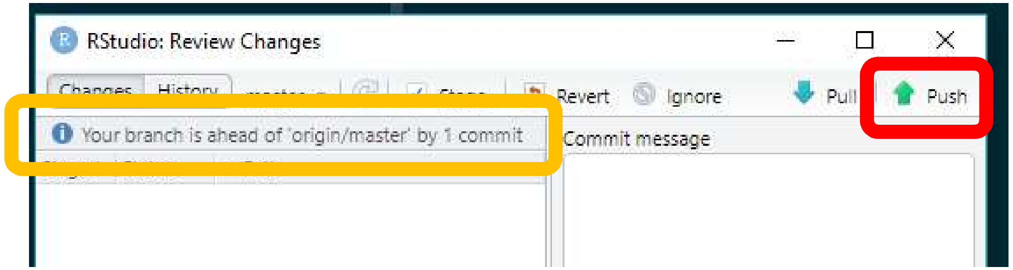
```
on your username and hostname. Please check that
You can suppress this message by setting them ex
following command and follow the instructions in
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used

    git commit --amend --reset-author

 2 files changed, 17 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 test.Rproj
```

Should get confirmation – select Close



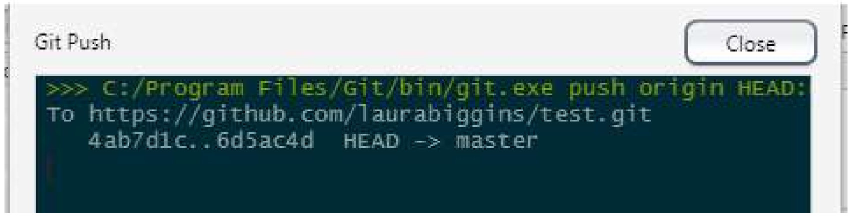
Your branch is ahead of 'origin/master' by 1 commit.

Should see an information message saying your branch is ahead of origin/master by 1 commit.
The changes – the creation of the new files - test.Rproj file and .gitignore, have been committed to the local git repo, but they haven't been pushed to the remote repository.



Your branch is ahead of 'origin/master' by 1 commit

The push button can now be clicked

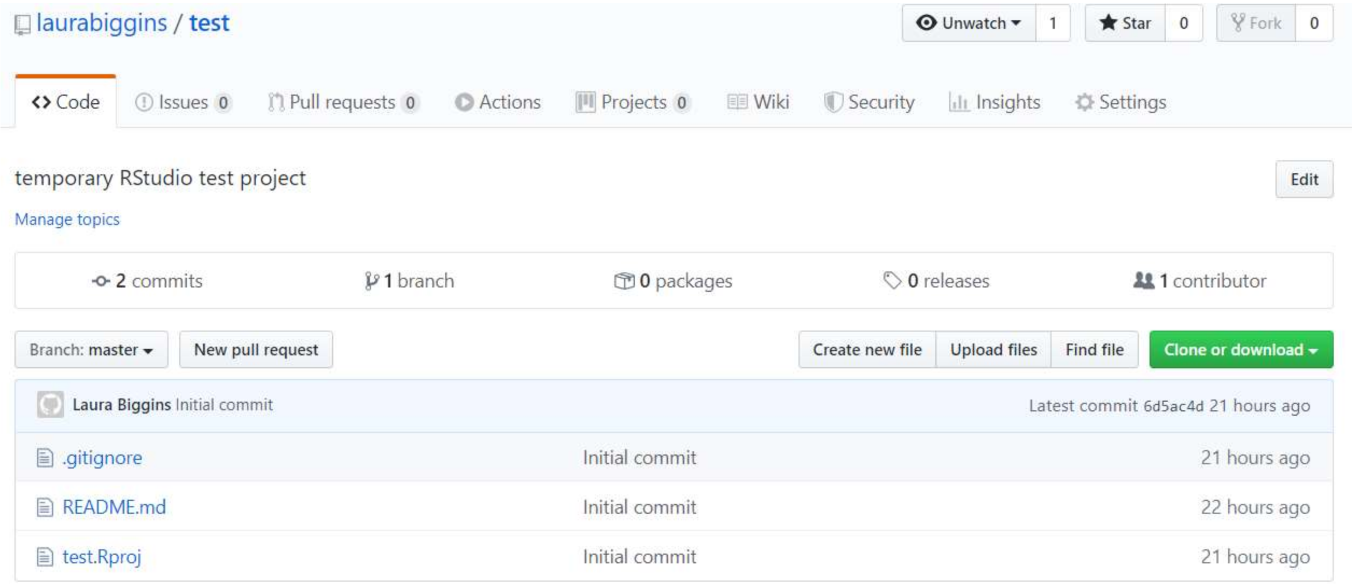


```
>>> C:/Program Files/Git/bin/git.exe push origin HEAD:
To https://github.com/laurabiggins/test.git
   4ab7d1c..6d5ac4d  HEAD -> master
```

This should then push the changes to the remote repository.

Back to GitHub

This should then push the changes to the remote repository and if the GitHub web page is refreshed, the new files should appear.



laurabiggins / test

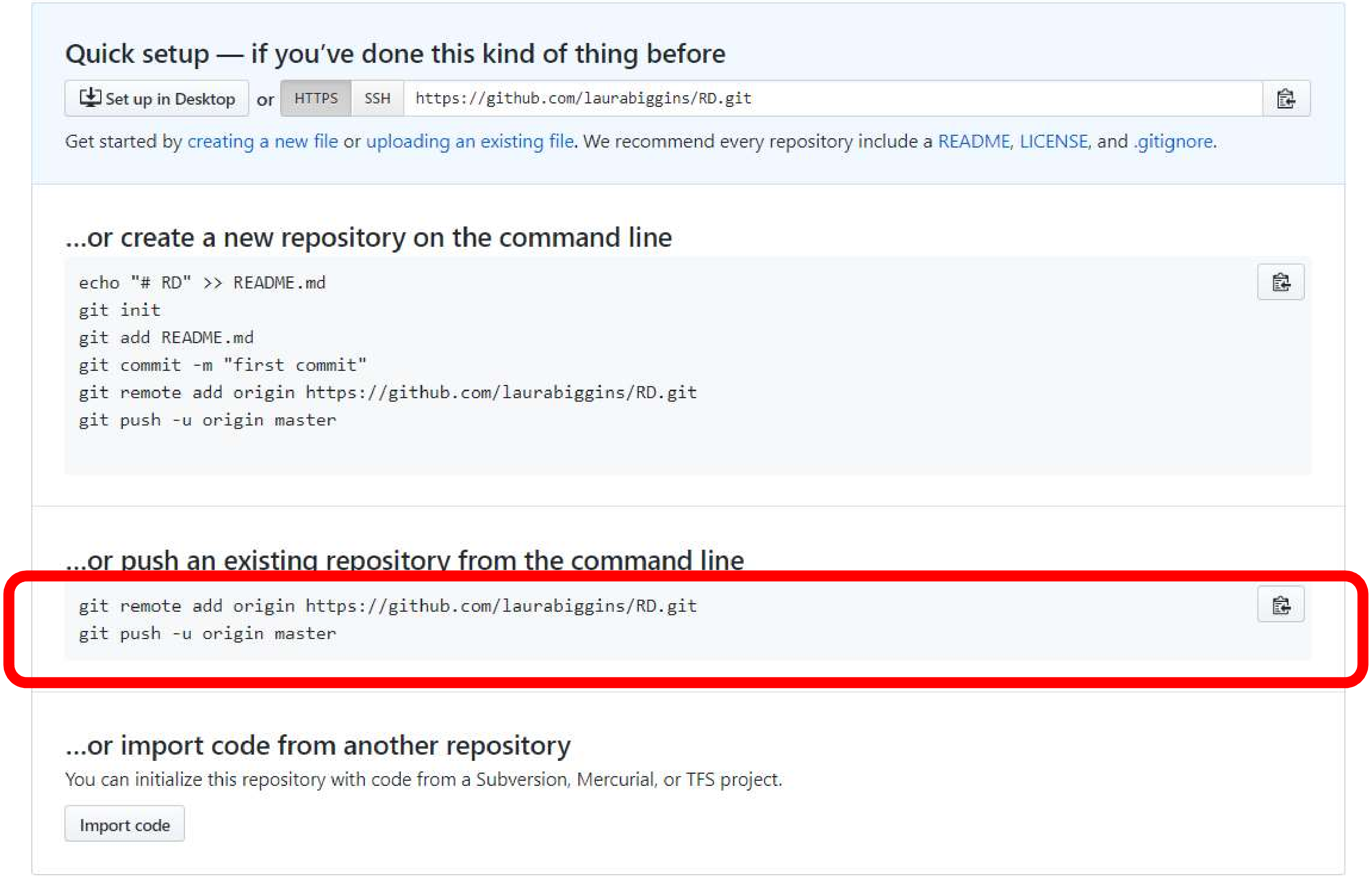
temporary RStudio test project

Laura Biggins Initial commit — Latest commit 6d5ac4d 21 hours ago

.gitignore	Initial commit	21 hours ago
README.md	Initial commit	22 hours ago
test.Rproj	Initial commit	21 hours ago

Existing R project not in GitHub

Export an R project to GitHub



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

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

…or push an existing repository from the command line

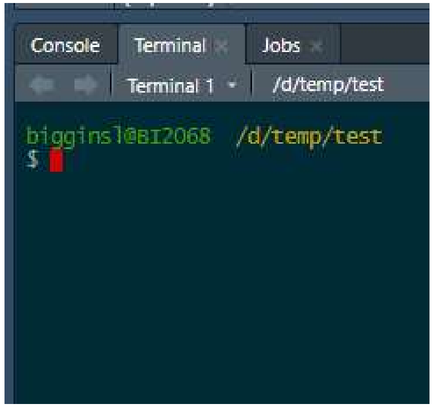
```
git remote add origin https://github.com/laurabiggins/RD.git
git push -u origin master
```

…or import code from another repository

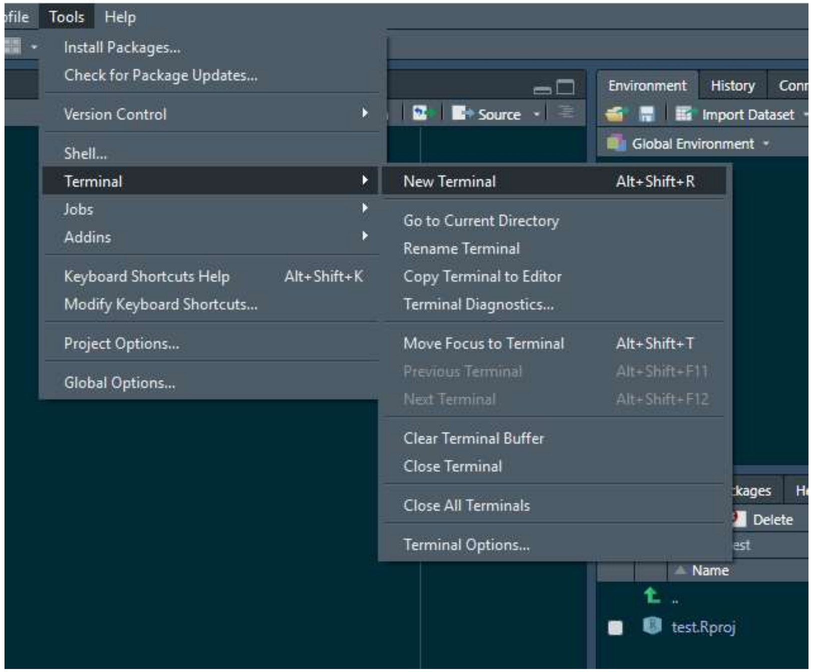
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Copy the 2
lines of code

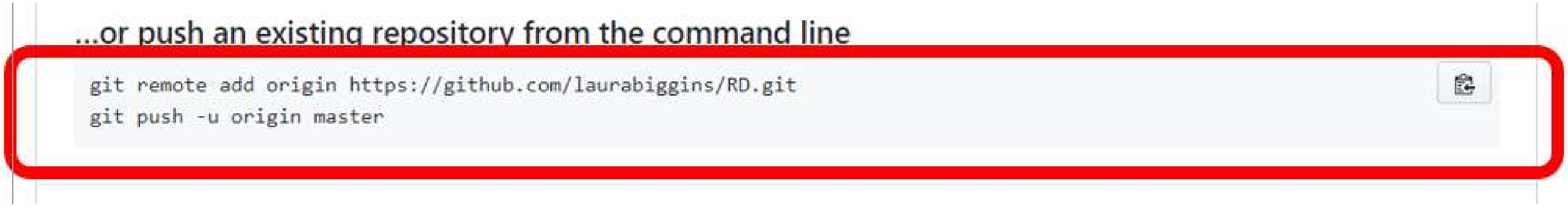
Using the terminal in RStudio



```
biggins1@BI2068  /d/temp/test
$
```

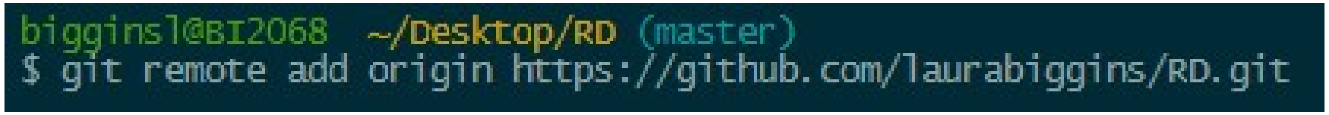


Using the terminal in RStudio

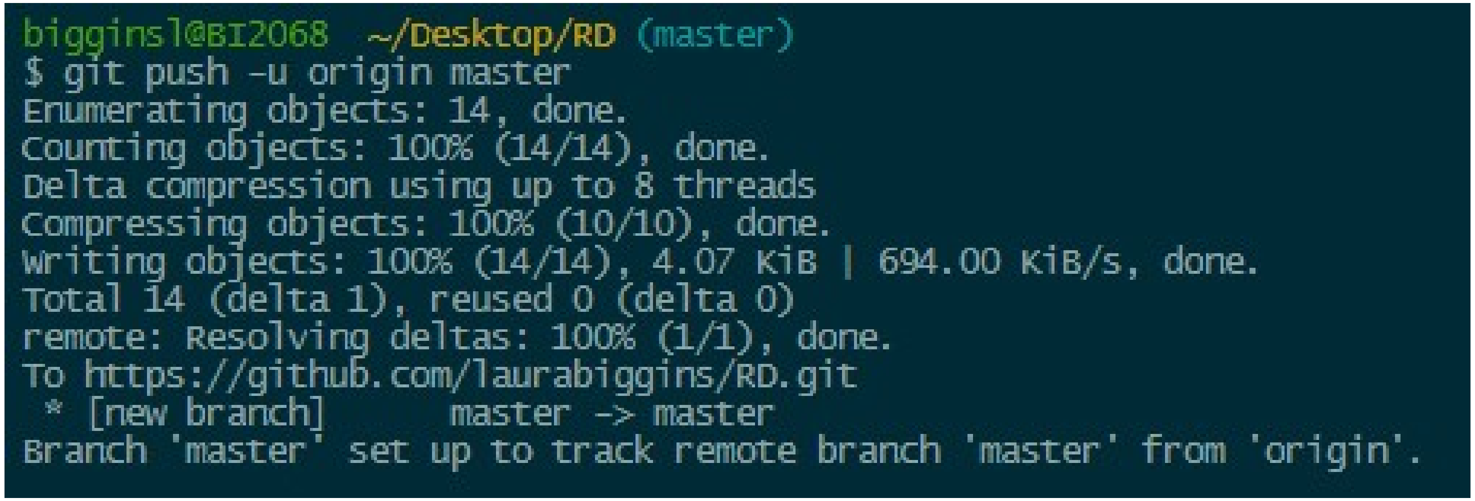


...or push an existing repository from the command line

```
git remote add origin https://github.com/laurabiggins/RD.git
git push -u origin master
```



```
biggins1@BI2068  ~/Desktop/RD (master)
$ git remote add origin https://github.com/laurabiggins/RD.git
```



```
biggins1@BI2068  ~/Desktop/RD (master)
$ git push -u origin master
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (14/14), 4.07 KiB | 694.00 KiB/s, done.
Total 14 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/laurabiggins/RD.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```