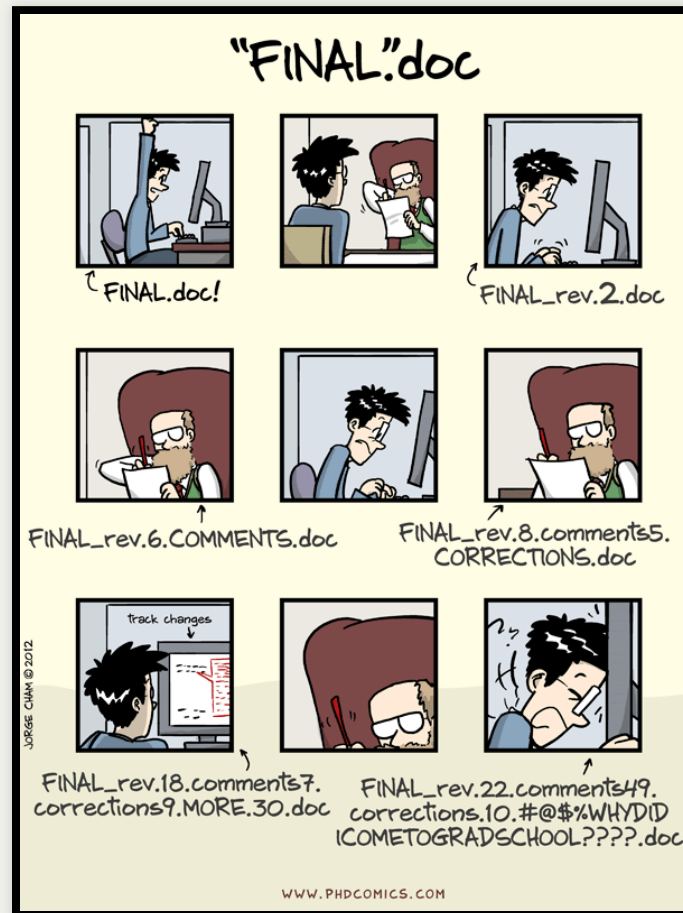


Version control with RStudio

An introduction to git and GitHub



Blake Seers

Topics covered

- Why version control?
- Git as a version control system.
- GitHub as a cloud-based management system.
- Git and GitHub basics.
- Examples to follow.

All that is required is RStudio, a browser, and an internet connection.

Version control

What is a version control system (VCS)?

- A system that tracks the changes to your files through time.
- ‘Track changes’ in Microsoft Word is a rudimentary type of version control.
- You’ve likely tried to implement your own system before (thesis_FINALLLL.docx).
- A good VCS should have a unique identifier showing:
 - What changed?
 - When?
 - Who?
 - Why?

Why should I use a version control system?

1. Safety

- Fail more safely (can go back in time).
- Everything is tracked so you can undo an undo.
- Fail more quickly (continuous integration).
- Get to a successful place much faster.

2. Community

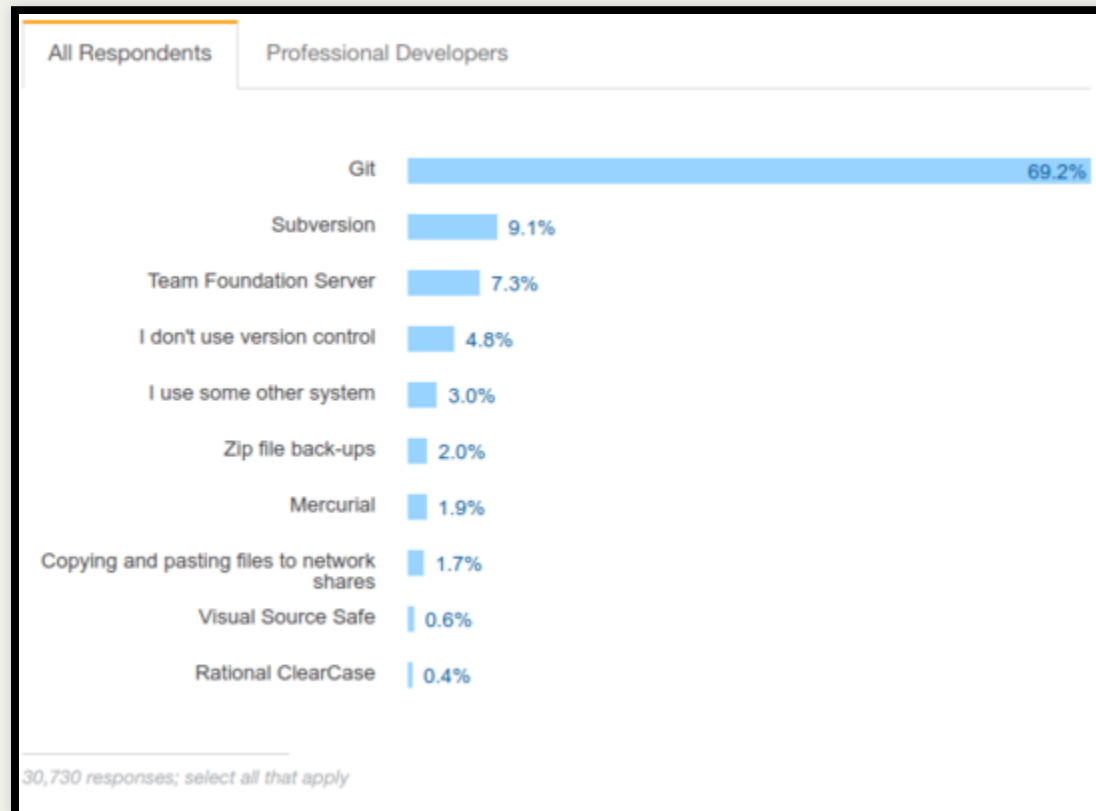
- There is a large community out there to help you and to benefit from your project.

For more on this, see the talk by Hadley.

Git as a version control system

Why should I git?

- Free, open-source VCS.
- Can work offline.
- Every developer has their own, complete codebase.
- By far the most popular:



Version control with git:

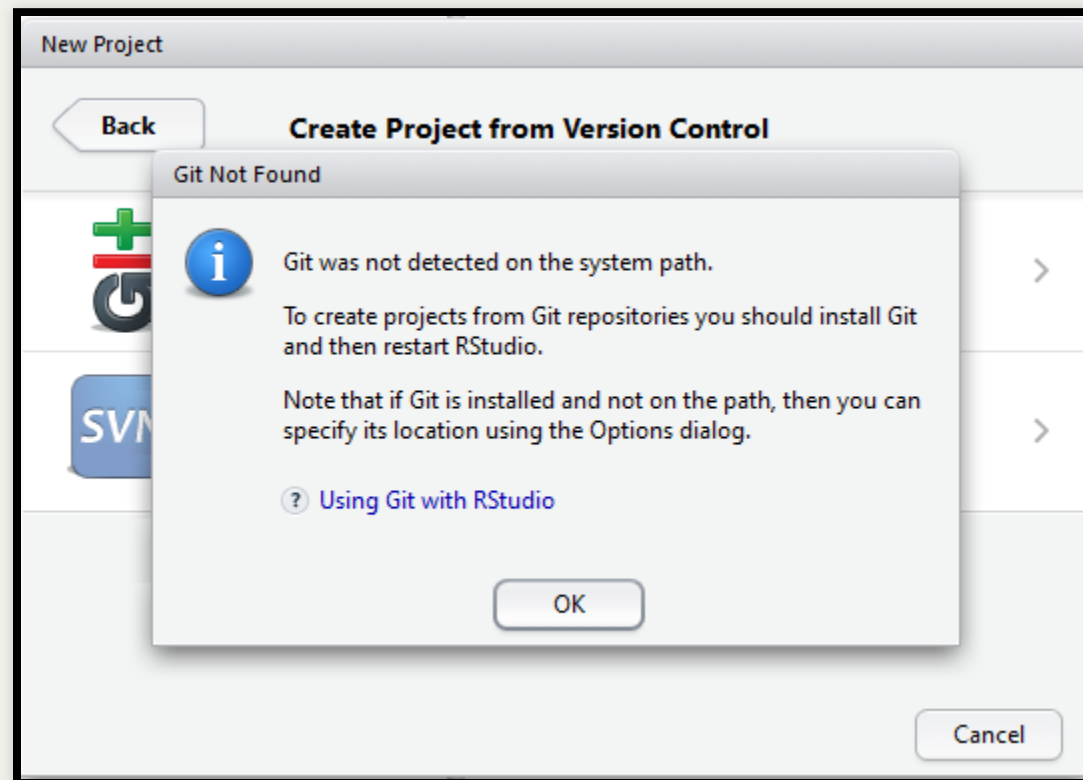
- A unique identifier (SHA for each 'save') gives info on:
 - What changed?
 - When?
 - Who?
 - Why?

Obtaining git

Check if you already have git installed

In RStudio:

File → New Project → Version Control → Git



Download git

On Windows:

1. go to <https://git-scm.com/download/win>.
2. This will automatically download latest version of git.
3. Choose to 'Run' exe to set up:
 - choose “Use Git from the Windows Command Prompt” when asked.

Download git

On Ubuntu:

```
# This PPA provides the latest stable upstream Git version:  
add-apt-repository ppa:git-core/ppa  
apt update  
apt install git
```

(may be adapted depending on your Linux/Unix flavour)

Download git

On Mac OS X:

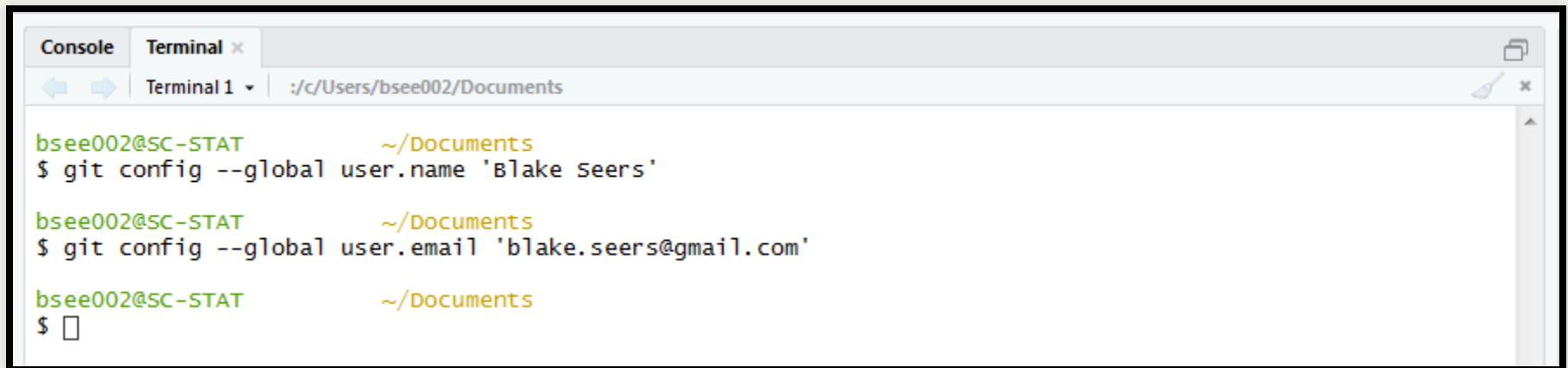
- Maverick 10.9 or higher - should already be installed
- Otherwise go to <https://git-scm.com/download/mac>

Introduce ourselves to git

Now that we have git installed, we want to introduce ourselves.

In RStudio, open the terminal tab in the console pane:

```
git config --global user.name 'Your Name'  
git config --global user.email 'your@email.com'
```

A screenshot of the RStudio interface, specifically the terminal pane. The terminal window is titled 'Terminal 1' and shows the current directory as '/c/Users/bsee002/Documents'. The user 'bsee002@SC-STAT' is shown at the prompt. The terminal displays the following commands and their execution: 1. '\$ git config --global user.name 'Blake Seers'' 2. '\$ git config --global user.email 'blake.seers@gmail.com'' 3. The prompt '\$ ' followed by a cursor, indicating the terminal is ready for the next command.

```
Console Terminal x  
Terminal 1 :/c/Users/bsee002/Documents  
bsee002@SC-STAT ~/Documents  
$ git config --global user.name 'Blake Seers'  
bsee002@SC-STAT ~/Documents  
$ git config --global user.email 'blake.seers@gmail.com'  
bsee002@SC-STAT ~/Documents  
$
```

GitHub

What is GitHub?

- An online interface for git.
- Allows for cloud-based repository management.
- Connects you to the online community of developers and end users.
- Allows you to access your files from any computer.

Sign up for GitHub

- Go to <https://github.com/join>
- Select the 'Unlimited public repositories' for the free option.
- You will get emailed a link to complete the verification process.

Username

Pick a username

Email

you@example.com

Password

Create a password

Make sure it's at least 7 characters, including a number, and a lowercase letter.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

Join GitHub Education

Discounted and free plans are available for educational use.

You have submitted 3 requests

Jun 26, 2018 for @sccuoa **Approved**

Jun 26, 2018 for @blasee **Approved**

May 15, 2015 for @blasee **Approved**

Tell us what you need

Tell us about you

Which best describes you? ?

☐ Student ☐ Teacher ☐ Researcher ☐ Administrator/staff ☐ Other

What kind of GitHub account are you using for education? ?

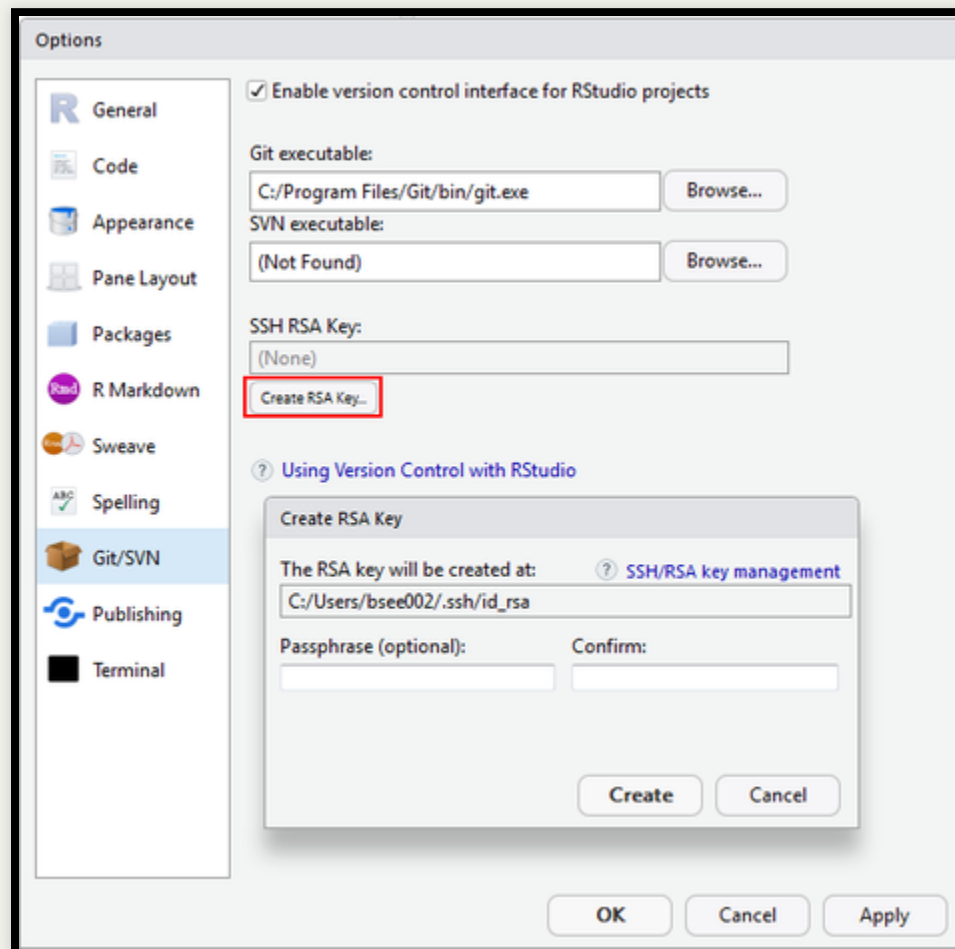
☐ Individual account ☐ Organization account

Next

Secure Shell (SSH) key (optional)

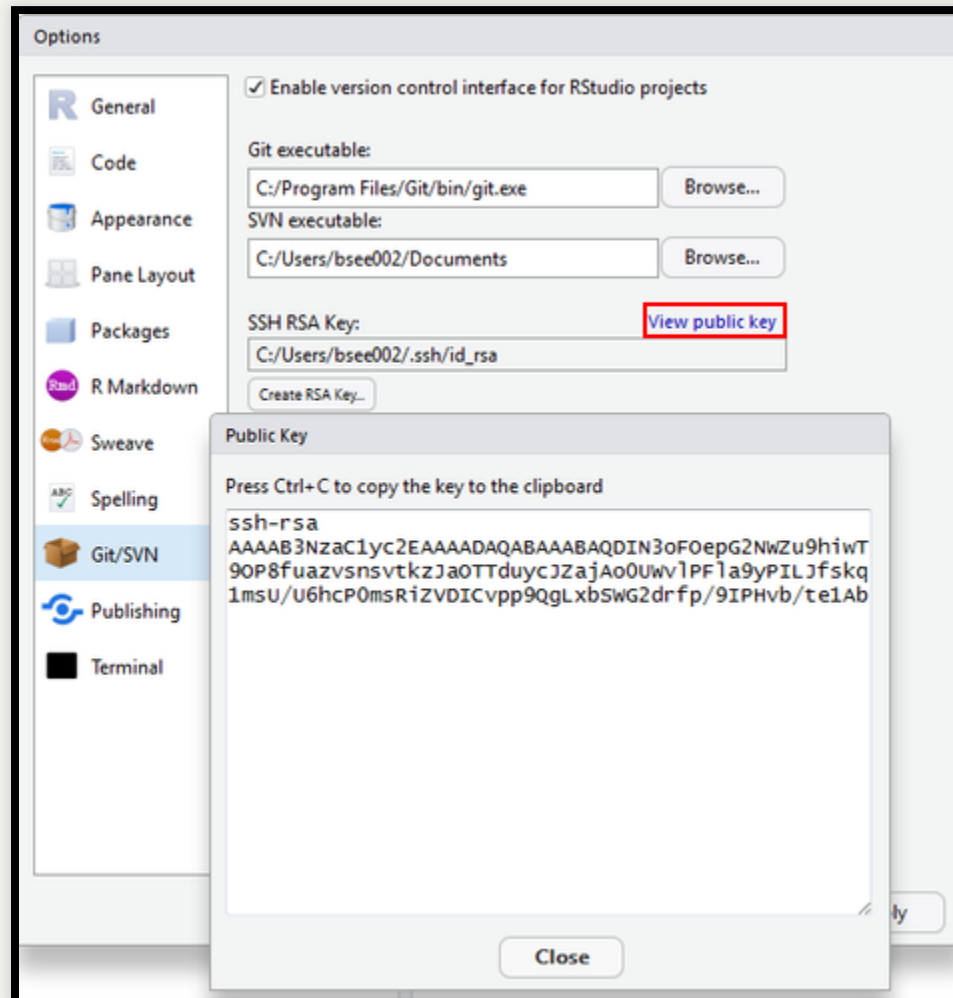
Create an SSH key in RStudio

- SSH allows you to connect and authenticate to remote servers.
- Tools → Global options → Git/SVN → Create RSA Key...



Connect SSH key to GitHub account

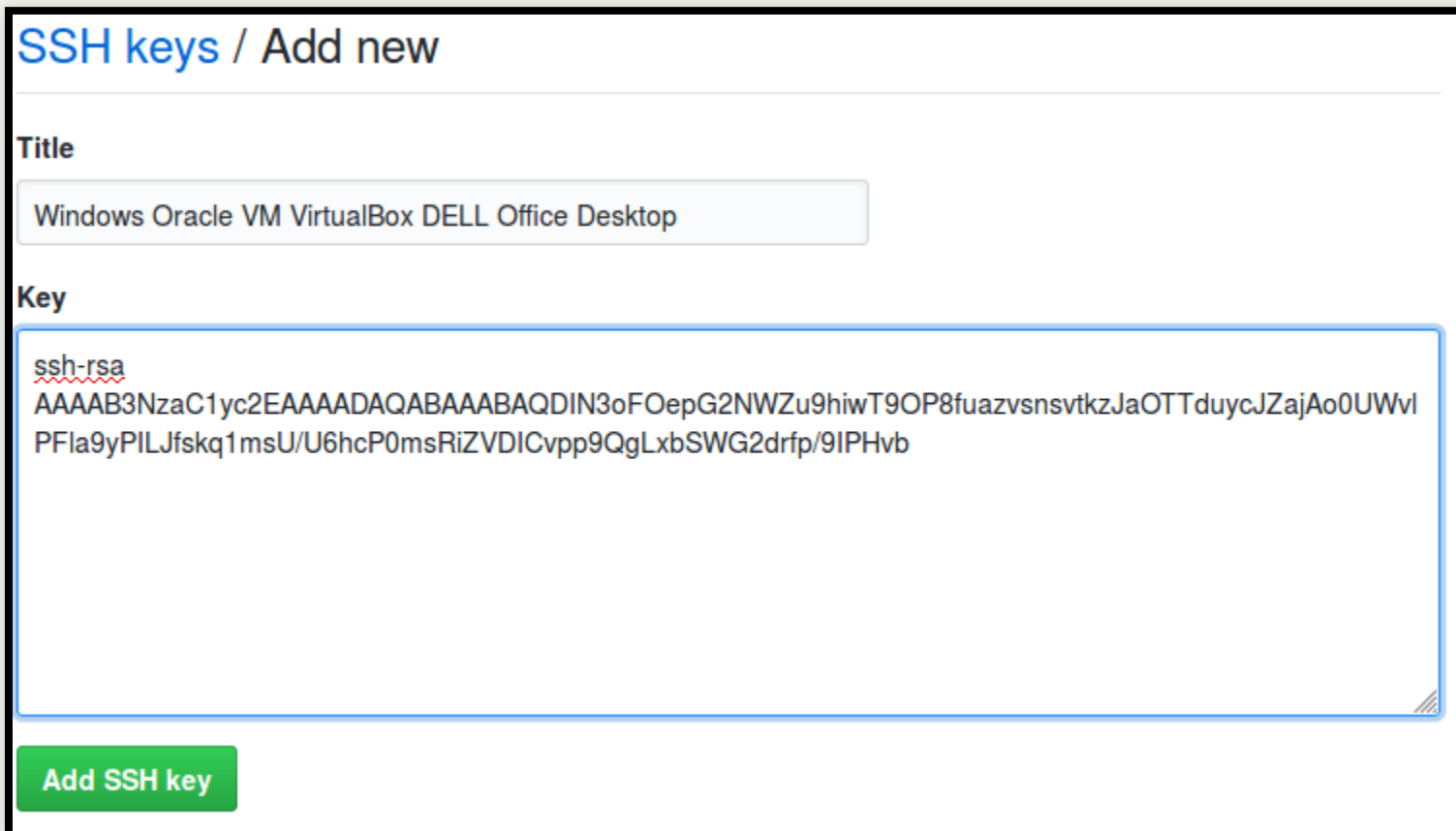
Click on “View Public Key” to copy the contents to clipboard.



Connect SSH key to GitHub account

In your browser:

- go to <https://github.com/settings/ssh/new>



SSH keys / Add new

Title

Windows Oracle VM VirtualBox DELL Office Desktop

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQADIN3oFOepG2NWZu9hiwT9OP8fuazvsnsvtkzJaOTTduycJZajAo0UWvl
PF1a9yPILJfskq1msU/U6hcP0msRiZVDICvpp9QgLxbSWG2drfp/9IPHvb
```

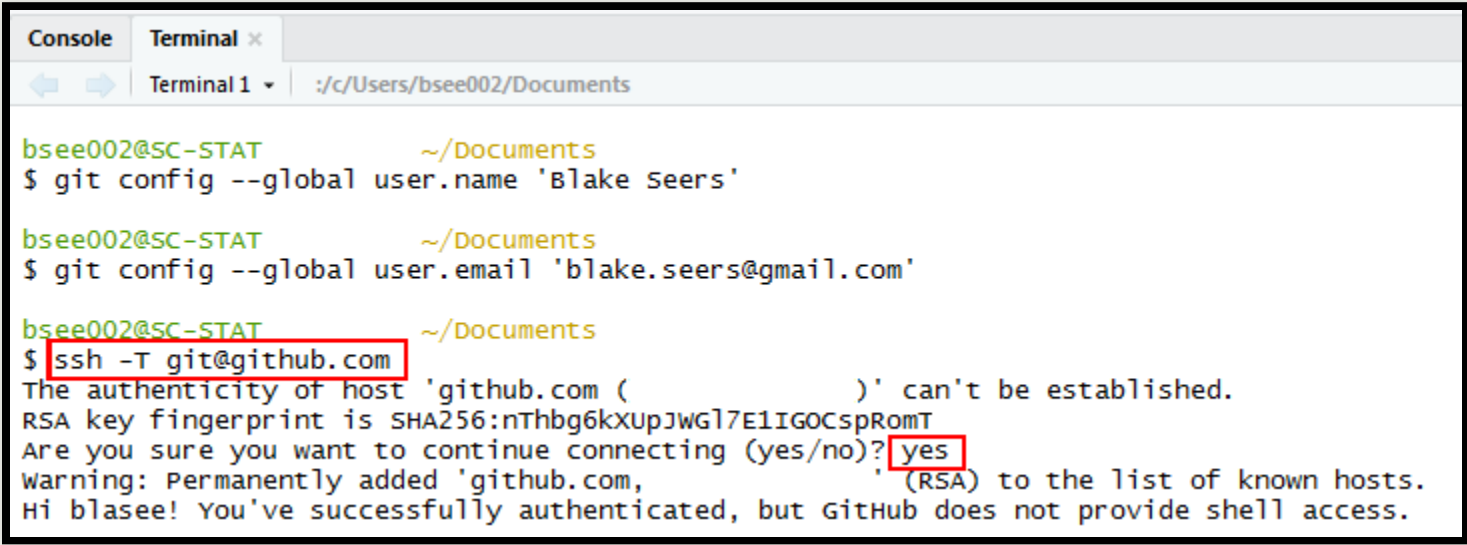
Add SSH key

Check connection

In the Terminal pane, in RStudio:

```
ssh -T git@github.com
```

... and then type `yes` when prompted.



The screenshot shows the RStudio interface with the 'Terminal' pane active. The terminal window title is 'Terminal 1' and the current directory is '/c/Users/bsee002/Documents'. The user 'bsee002@SC-STAT' is in the directory '~/Documents'. The terminal shows the following commands and output:

```
bsee002@SC-STAT ~/Documents
$ git config --global user.name 'Blake Seers'

bsee002@SC-STAT ~/Documents
$ git config --global user.email 'blake.seers@gmail.com'

bsee002@SC-STAT ~/Documents
$ ssh -T git@github.com
The authenticity of host 'github.com (192.168.1.1)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXupJWG17E1IGOCspR0mT
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com, 192.168.1.1' (RSA) to the list of known hosts.
Hi blasee! You've successfully authenticated, but GitHub does not provide shell access.
```

The command `ssh -T git@github.com` is highlighted with a red box. The prompt `Are you sure you want to continue connecting (yes/no)?` is also highlighted with a red box, and the response `yes` is entered.

Core concepts

Core concepts in git and GitHub

- **Repository**

- Contains all your files (and history) for the project.
- One-to-one mapping between **R** project and 'repo'.

- **Commit**

- Each commit is a snapshot of your files.
- Equivalent to a saving `FINAL_after_3rd_review.docx`.
- Can time travel between commits.

- **Staging**

- Files that have been selected (staged) for a commit.

Core concepts in git and GitHub

- **Push**

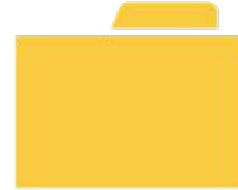
- Making the changes you have made locally (using git) available to the online repository (GitHub).

- **Pull**

- Updating your local repository to the current version in the online repository.

Core concepts in git and GitHub

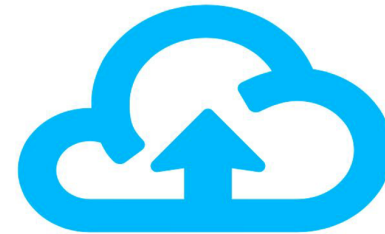
Repository, AKA Repo



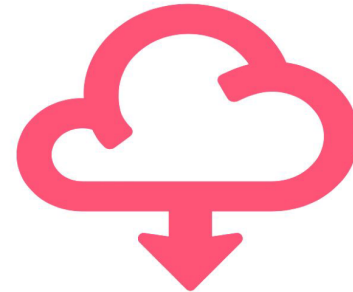
Commit



Push



Pull



Core concepts in git and GitHub

- **Clone**

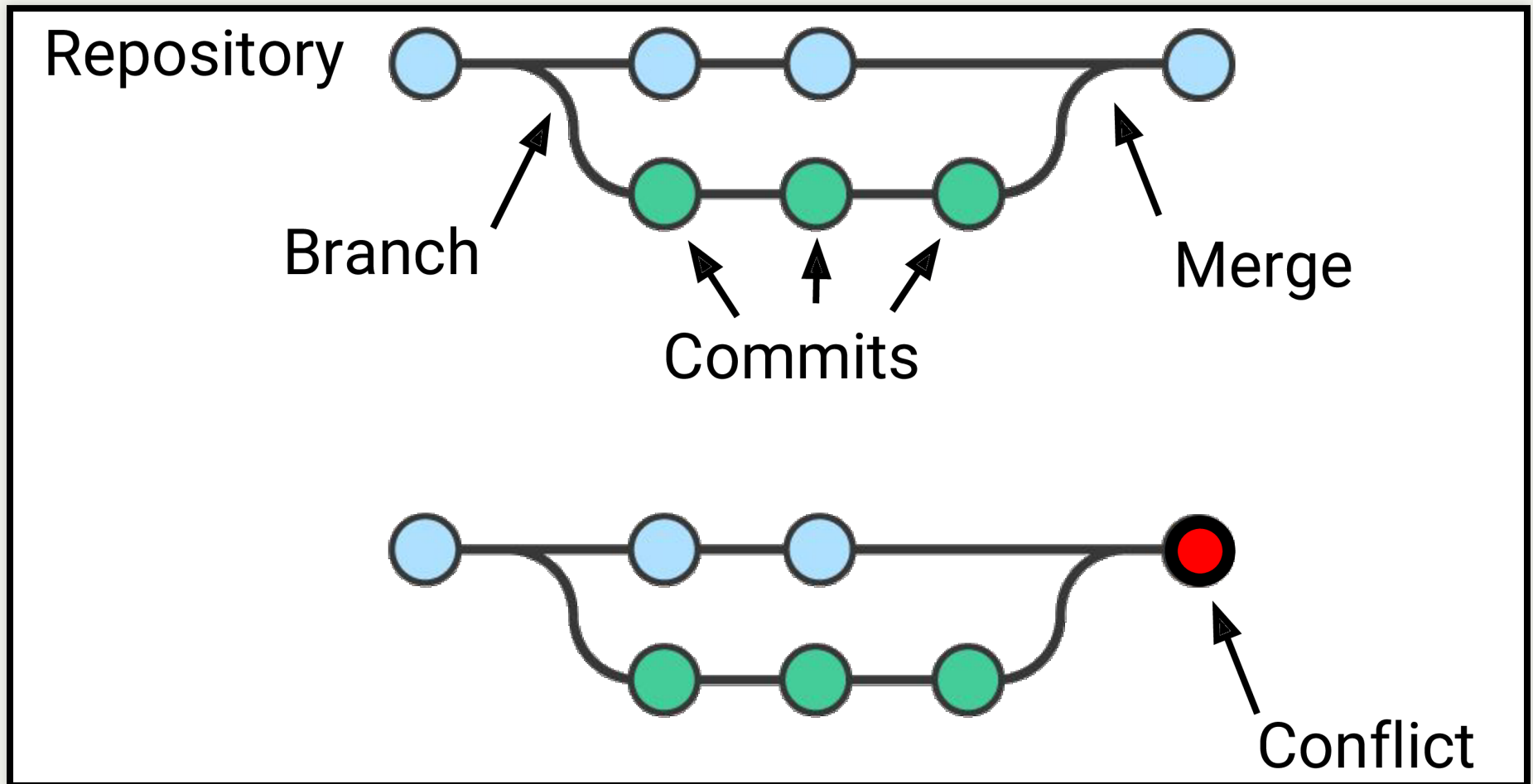
- Copy (clone) an existing repository locally.
- For example, working on a project from a different computer.

- **Fork**

- A personal copy of a repository ‘taken’ from another user.

Core concepts in git and GitHub

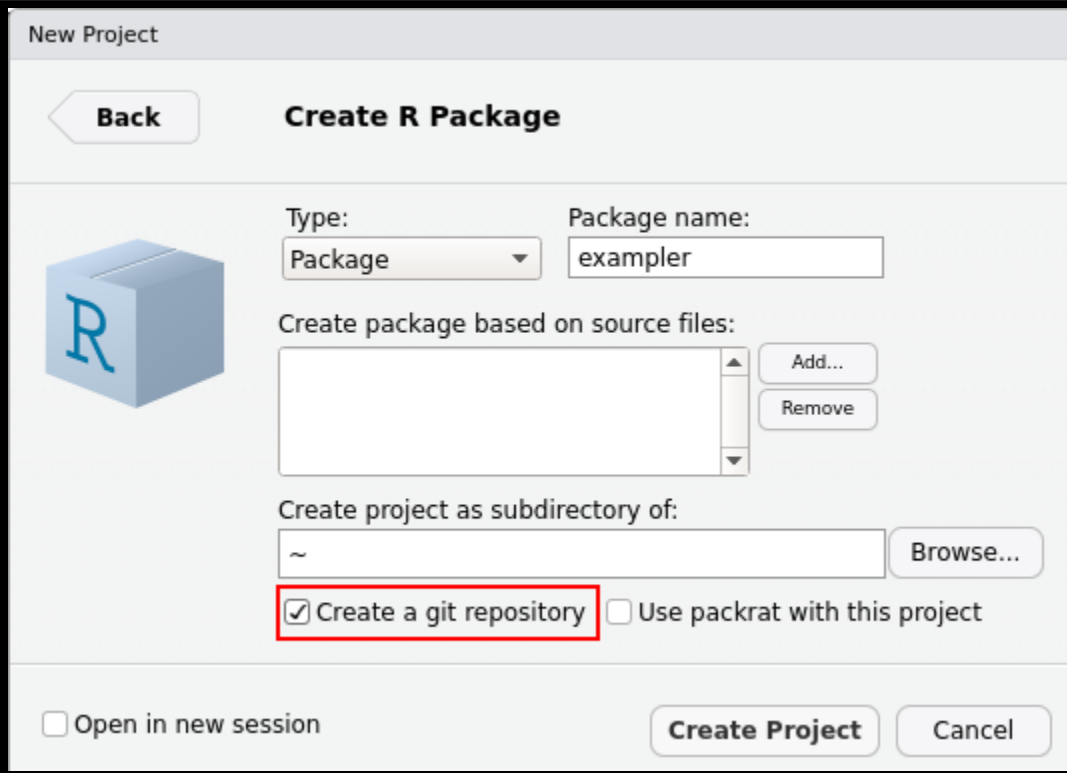
- **Branch, Merge, and Conflict**



Examples for getting started with git

Start a new R package

- New Project → R Package:



New Project

Back **Create R Package**

Type: Package Package name: exemplar

Create package based on source files:

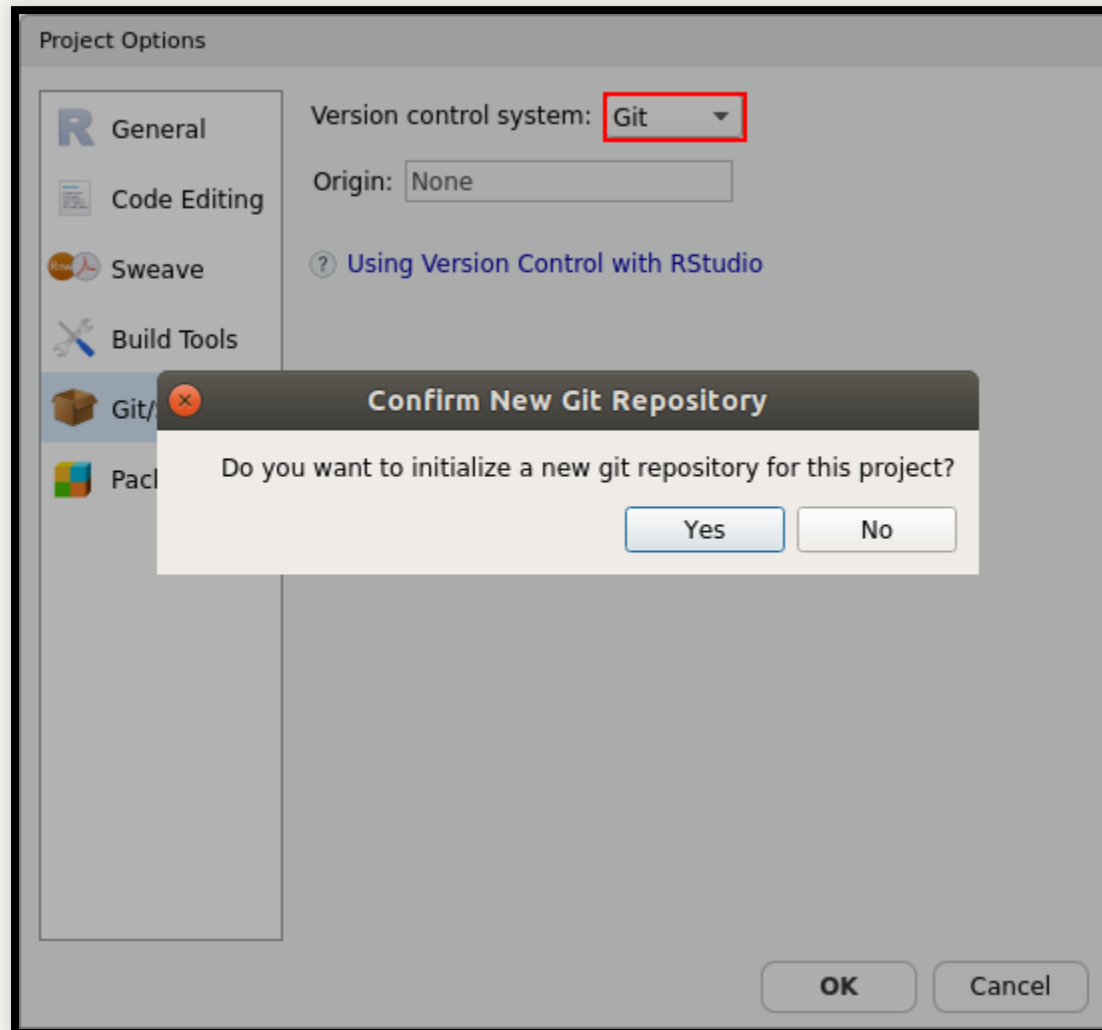
Create project as subdirectory of: ~ Browse...

☒ Create a git repository ☐ Use packrat with this project

☐ Open in new session **Create Project** Cancel

Initialize an existing repo with git

Tools → Project Options → Git/SVN:



The git tab

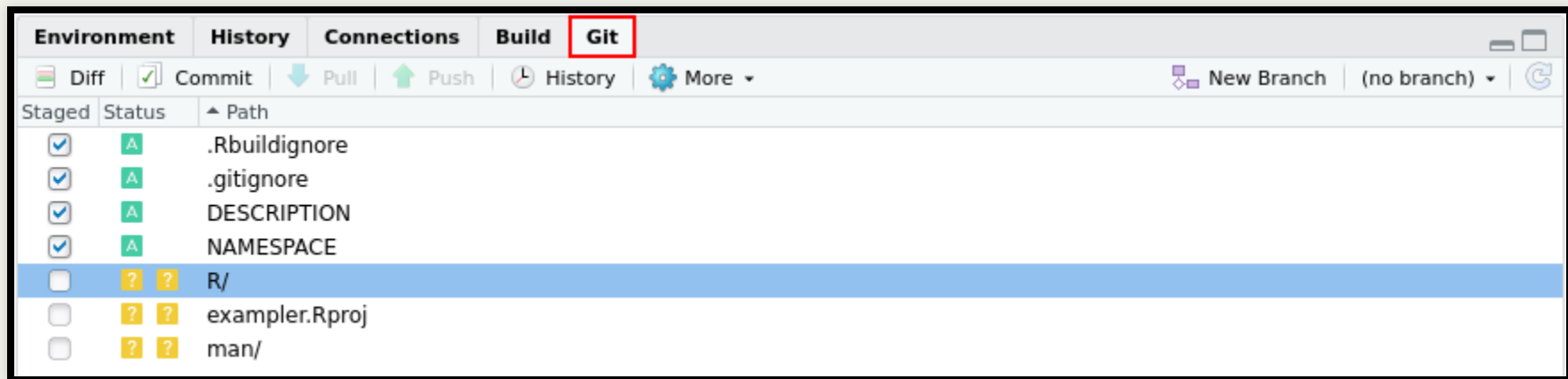
- Once initialized, there is a git tab in RStudio:



- This git tab in RStudio tab contains all the common git functionality we need.

Add files to the staging area

- In RStudio just check the boxes to add files to be staged.
- Here we will add all files in an ‘initial commit’.



- Press the ‘Commit’ button when you’re ready to commit.
- Shortcut in RStudio: Ctrl + Alt + M.

An initial commit

- It is common to use ‘initial commit’ for the first commit message.



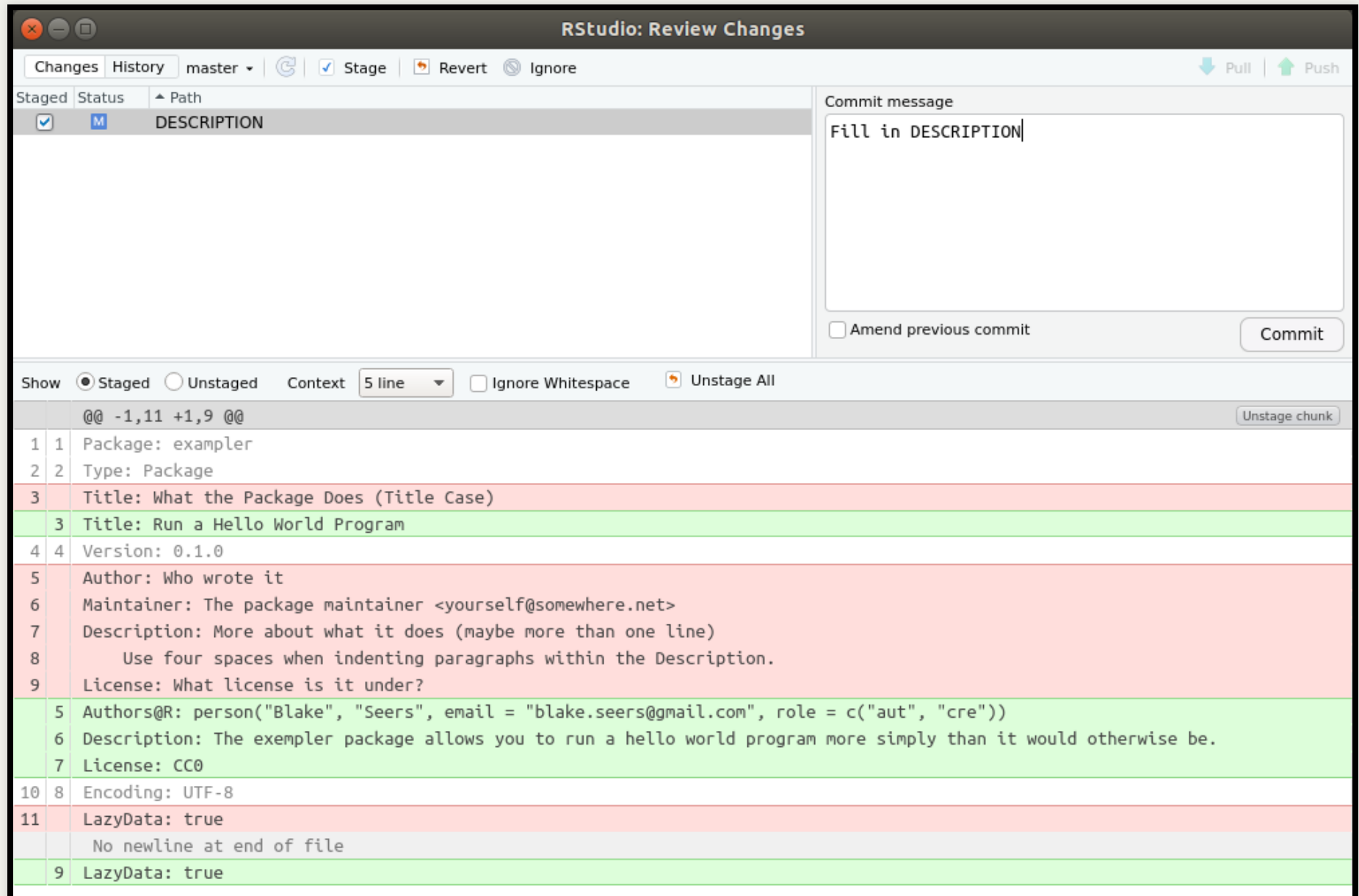
Commit message

Initial commit

☐ Amend previous commit

Commit

Modify a file and commit



RStudio: Review Changes

Changes | History | master | Stage | Revert | Ignore

Staged | Status | Path

Commit message

Fill in DESCRIPTION

☐ Amend previous commit

Commit

Show | Staged | Unstaged | Context | 5 line | Ignore Whitespace | Unstage All

@@ -1,11 +1,9 @@

1 1 Package: exemplar

2 2 Type: Package

3 Title: What the Package Does (Title Case)

3 Title: Run a Hello World Program

4 4 Version: 0.1.0

5 Author: Who wrote it

6 Maintainer: The package maintainer <yourself@somewhere.net>

7 Description: More about what it does (maybe more than one line)

8 Use four spaces when indenting paragraphs within the Description.

9 License: What license is it under?

5 Authors@R: person("Blake", "Seers", email = "blake.seers@gmail.com", role = c("aut", "cre"))

6 Description: The exemplar package allows you to run a hello world program more simply than it would otherwise be.

7 License: CC0

10 8 Encoding: UTF-8

11 LazyData: true

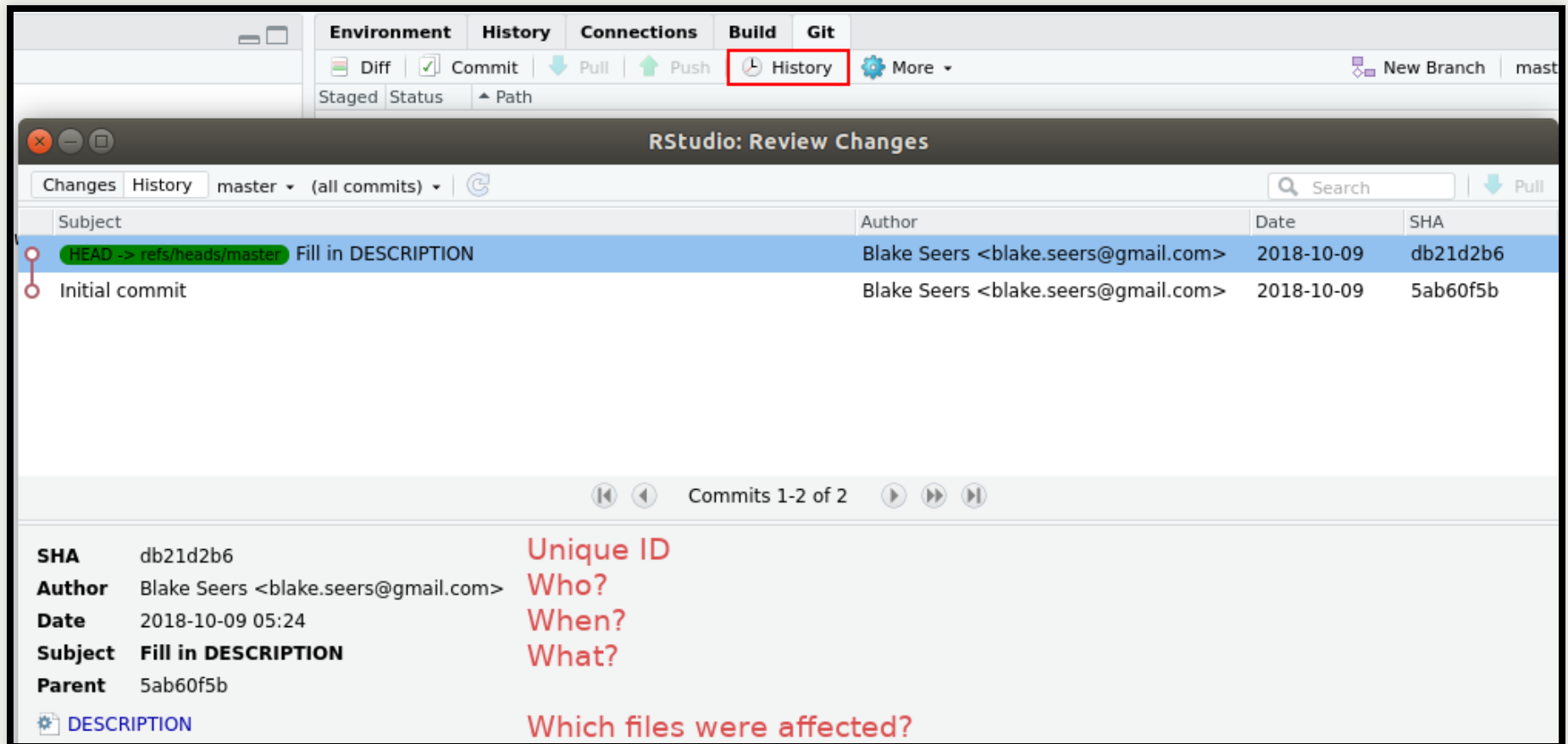
No newline at end of file

9 LazyData: true

Unstage chunk

View a history of commits in RStudio

Now we are ready to add our commits to GitHub.



The screenshot shows the RStudio interface with the Git pane open. The 'History' tab is selected, showing a list of commits. The first commit is highlighted, showing its details in the lower pane.

RStudio: Review Changes

Changes | History | master ▾ | (all commits) ▾ | Search | Pull

Subject	Author	Date	SHA
HEAD -> refs/heads/master Fill in DESCRIPTION	Blake Seers <blake.seers@gmail.com>	2018-10-09	db21d2b6
Initial commit	Blake Seers <blake.seers@gmail.com>	2018-10-09	5ab60f5b

Commits 1-2 of 2

SHA db21d2b6
Author Blake Seers <blake.seers@gmail.com>
Date 2018-10-09 05:24
Subject Fill in DESCRIPTION
Parent 5ab60f5b

[DESCRIPTION](#)

Unique ID
Who?
When?
What?
Which files were affected?

Create a repo on GitHub


- Go to <https://github.com/new> to create a new repo:

Create a new repository


A repository contains all the files for your project, including the revision history.

Owner

Repository name

 blasee ▾


 /

exampler 


Great repository names are short and memorable. Need inspiration? How about **musical-tribble**.

Description (optional)

An example R package

☒  **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**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

 |

Add a license: **None** ▾ 

Create repository

Create a repo on GitHub

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

or **HTTPS** **SSH** `https://github.com/blasee/exampler.git`



We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

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



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

```
git remote add origin https://github.com/blasee/exampler.git
git push -u origin master
```



- Simply copy-paste the code to the console in RStudio.

Package on GitHub

- Your files are now on GitHub!
- `devtools::install_github("blasee/exampler")`

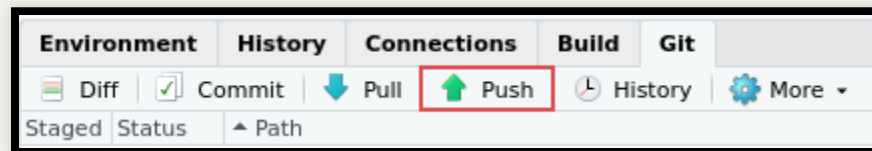
The screenshot shows the GitHub repository page for 'blasee/exampler'. The repository is described as 'An example R package'. It has 2 commits, 1 branch, 0 releases, and 1 contributor. The latest commit is 'db21d2b' from 3 hours ago. The repository contains the following files and folders:

File/Folder	Commit Message	Time
R	Initial commit	5 hours ago
man	Initial commit	5 hours ago
.Rbuildignore	Initial commit	5 hours ago
.gitignore	Initial commit	5 hours ago
DESCRIPTION	Fill in DESCRIPTION	3 hours ago
NAMESPACE	Initial commit	5 hours ago
exampler.Rproj	Initial commit	5 hours ago

At the bottom, there is a prompt to 'Add a README' to help people understand the project.

Push in RStudio

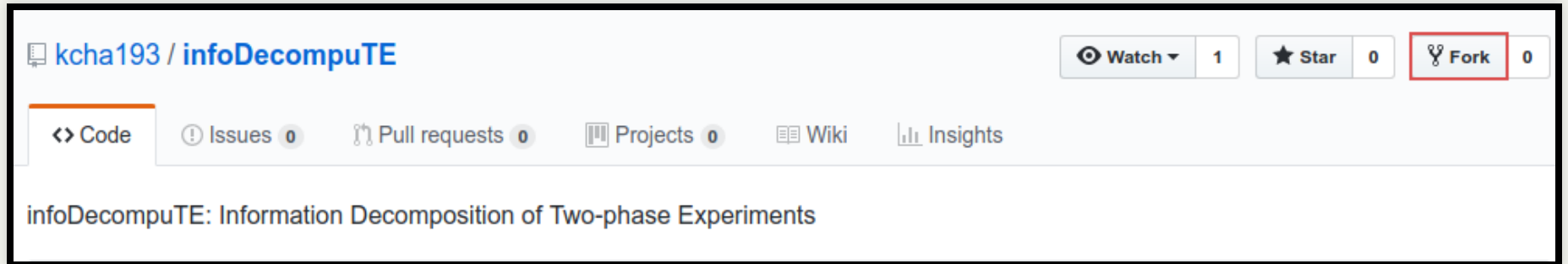
- Just because it is on GitHub, does not mean it is a finished product.
- Can go back to RStudio and continue working, committing, and pushing:
 - to push (or pull) in the future, you can press the “Push” button:



Forking and cloning repositories

Fork a repository

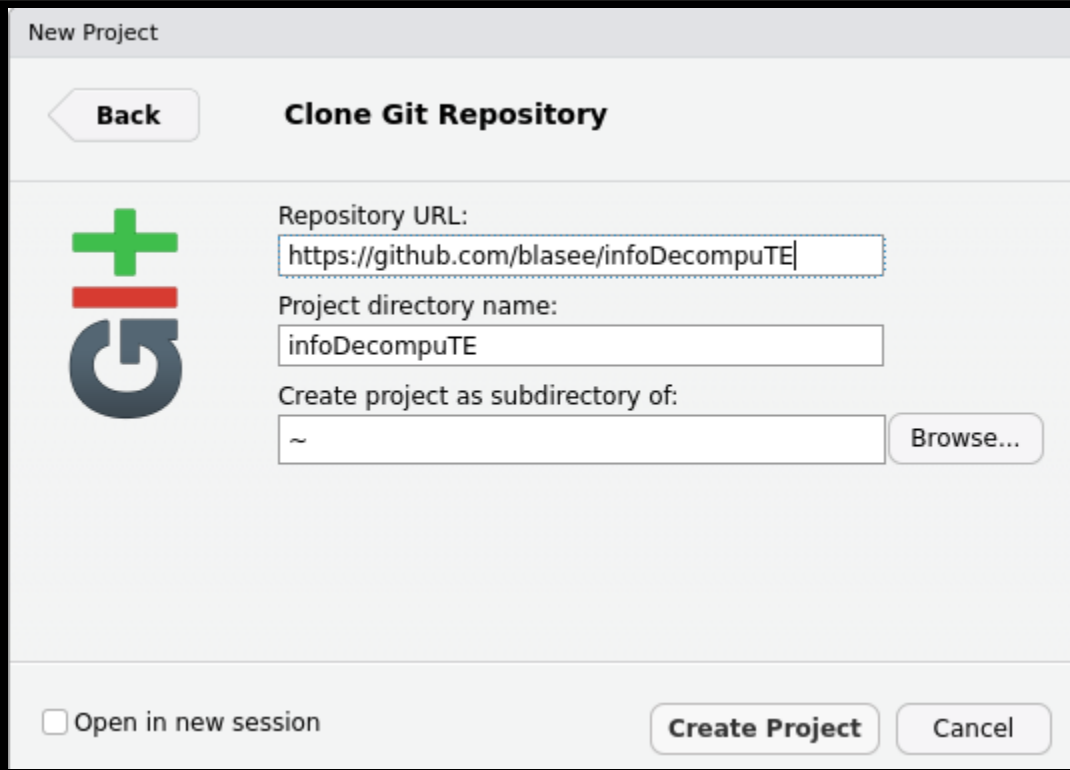
- Go to any repository and press ‘Fork’:



- This repository now lives at <https://github.com/blasee/infoDecompuTE>

Clone the repository from RStudio

File → New Project → Version Control → Git




The image shows the 'New Project' dialog box in RStudio, specifically the 'Clone Git Repository' tab. The dialog has a title bar 'New Project' and a 'Back' button. On the left is the Git logo (a green plus sign over a blue 'G'). The main area contains three input fields: 'Repository URL:' with the value 'https://github.com/blasee/infoDecompuTE', 'Project directory name:' with the value 'infoDecompuTE', and 'Create project as subdirectory of:' with the value '~'. A 'Browse...' button is next to the last field. At the bottom, there is a checkbox 'Open in new session' and two buttons: 'Create Project' and 'Cancel'.

New Project

Back

Clone Git Repository



Repository URL:

Project directory name:

Create project as subdirectory of:

☐ Open in new session

Concluding remarks

gitignore files

- Files that match the patterns in gitignore will be ignored by git.
- Similar to Rbuildignore.
- Patterns include:
 - ‘#’ for comments.
 - \ is the escape character.
 - dir/ ignores the directory ‘dir’.
 - * matches anything (except “/”)
 - ? matches anything (except “/” and “[”)

Best practices

Each commit should:

- address a single issue.
- have a helpful commit message.
- use imperative verbs in the present tense.
 - ‘*Add* print S4 method to person class’
 - ‘*Remove* ORIGINAL.csv file from top directory’
 - ‘*Update* dependency from XML to xml2’
 - ‘*Fix* installation problem on Windows’

Important things not covered

- Creating tags and releases (**R** packages).
- Adding README files to your repository.
- Branching and reverting
- Tracking issues on GitHub
- Pull request on GitHub
- Continuous integration services
- GitHub pages for a website for your repo

Summary

- Use version control for your projects:
 - R packages
 - Thesis / dissertation
 - Course notes
 - Book
 - Website
- Version control is easy with RStudio.
- Code and collaborate with RStudio, git and GitHub!

