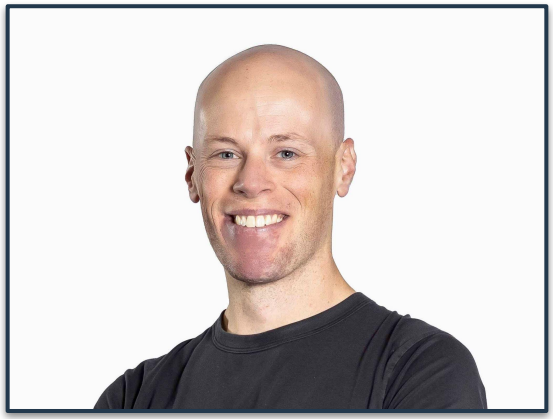
Intro to Version Control

(using)

git

RStudio®



Ryan Johnson

ryan@posit.co

Data Science Advisor

@ryjohnson09

---

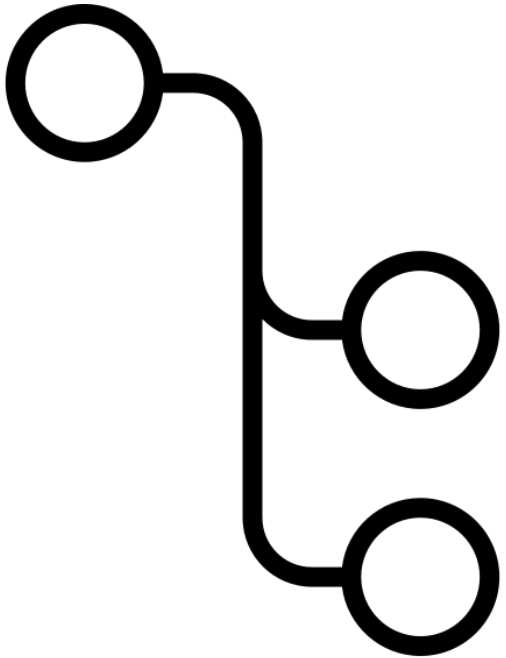
1. **Questions?** Pop them in the chat (Zoom / MS Teams).
2. Keep it **fun**, keep it **casual**, keep it **safe**!
3. Slides, material, and recording will be made available after.

posit™

Questions for the group!

Agenda

- What is **version control** and **Git**?
- Admin checklist
- RStudio Projects + Version Control
- RStudio Git integrations
- Git-Backed Deployment



Git + RStudio Resources

- Happy Git and GitHub for the useR
- RStudio: Version Control with Git and SVN
- Excuse me, do you have a moment to talk about version control?
- usethis R package

What is **version control** and **Git**?

Version Control: *Practice of **tracking** and **managing** changes to software code*

: Version Control System (VCS) that manages the evolution of a set of files – called a **repository**

Repositories can be **hosted and managed** using:

- **GitHub**
- **Bitbucket**
- **GitLab**
- **SourceForge**
- **Others…**

https://www.atlassian.com/git/tutorials/what-is-version-control

Using Git

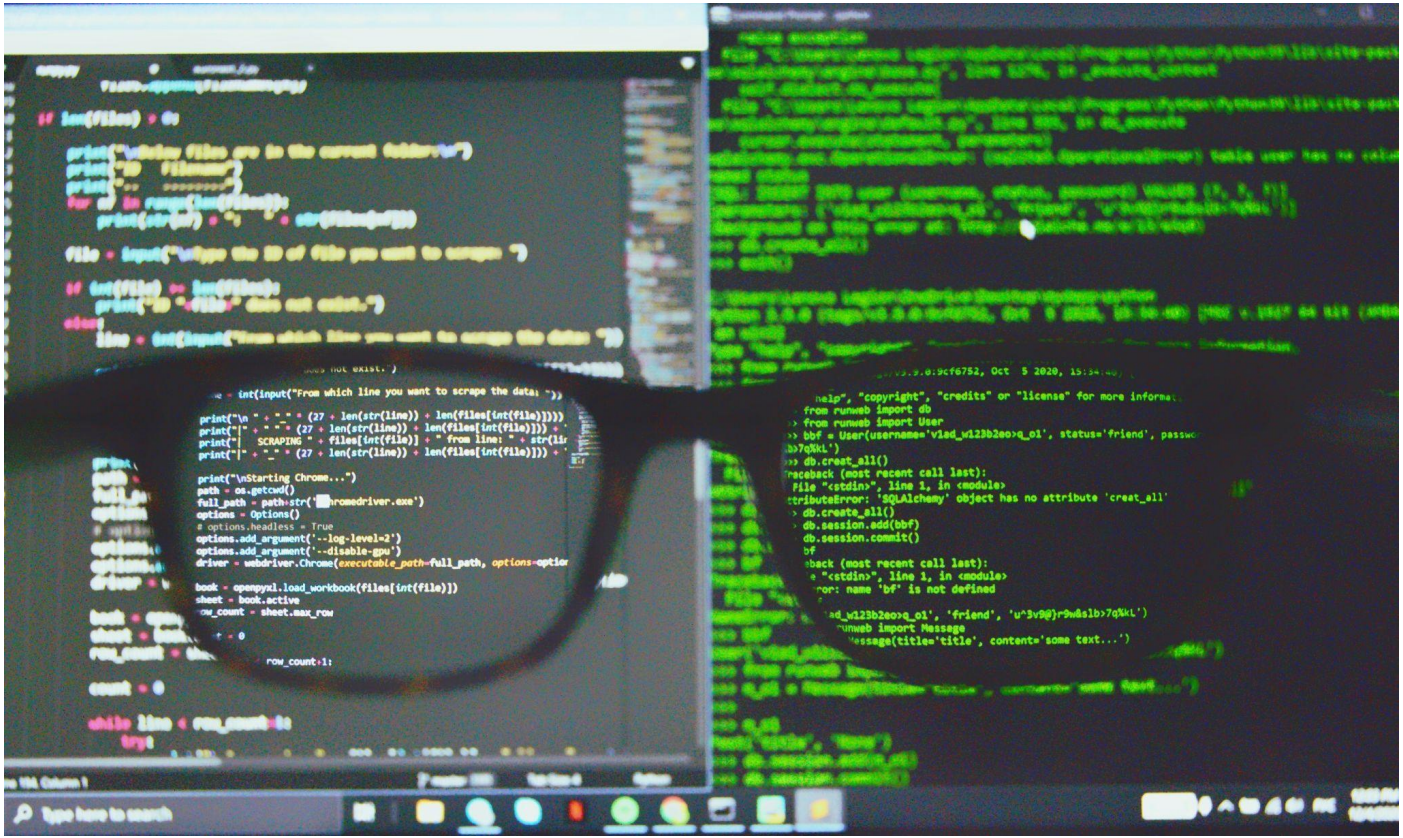
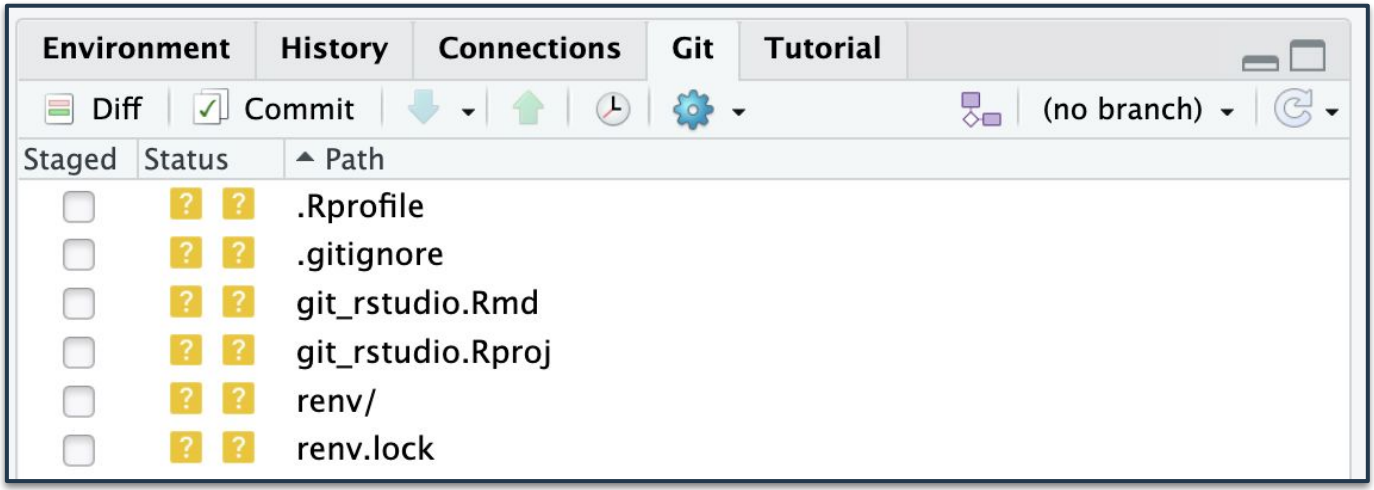


Photo by Alex Chumak on Unsplash

Using Git

Most Integrated Development Environments (IDEs) have integrated support for version control



But first!

We need to make sure Git can talk to our repository hosting service (e.g., GitHub)

1. Register an account on GitHub (or other service)
2. Install & update R/RStudio
3. Install Git
4. Configure Git - Make sure Git knows who you are!
5. Confirm you can push to and pull from repository hosting service

Things to help with Git configuration:

Happy Git and GitHub for the useR

Search

Table of contents

Let's Git started

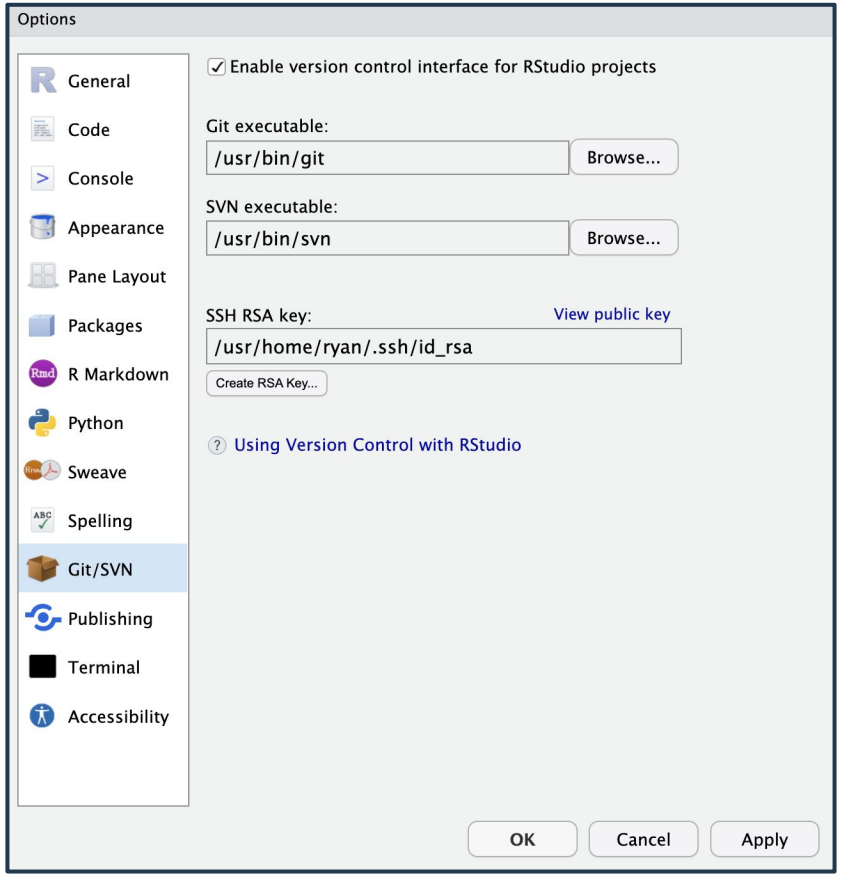
Still from Heaven King video

On this page

View source

Edit this page

Tools → Global Options → Git/SVN



usethis R package

usethis setup

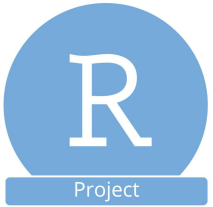
Source: `vignettes/articles/usethis-setup.Rmd`

You will get the most out of usethis if you do some setup. These setup tasks do not need to be done all at once or even done at all. But usethis can offer the most support for package development and Git/GitHub workflows with some advance configuration. usethis can even help you with this!

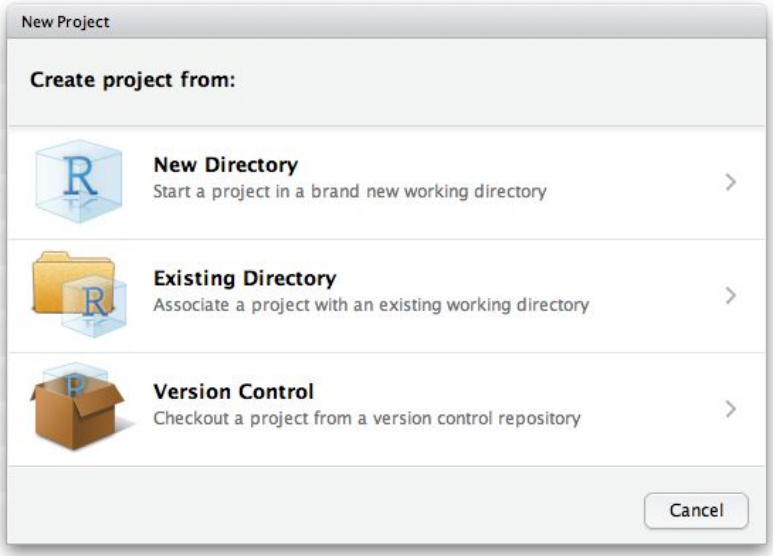
Key steps that accelerate your R development workflow (details on *how* to do all this follow):

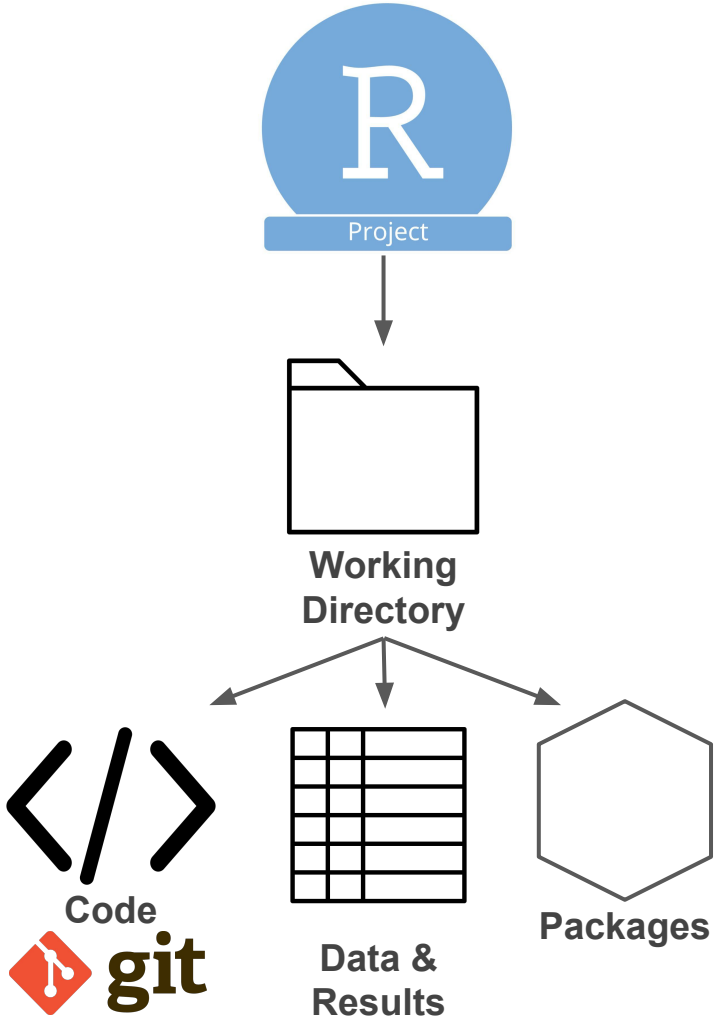
- Make usethis available in interactive R sessions.
- Provide usethis with info to use in all new R packages you create.
- Use the "sitrep" functions to get a health check or gather info when you're going to ask for help.
- Sign up for a free GitHub.com account, if you plan to use GitHub.
- Install Git.
- Configure your Git `user.name` and `user.email`.
- If you use RStudio, make sure RStudio can find your Git executable. If you use GitHub, make sure you can pull/push from your local computer to GitHub.com, in general and from RStudio.
- Get a personal access token from GitHub.com and make it available in R sessions.
- Prepare your system to build R packages from source.

ON THIS PAGE

In order to use **version control** features in  Studio® You must use 

In order to use **version control** features in RStudio® You must use R Project

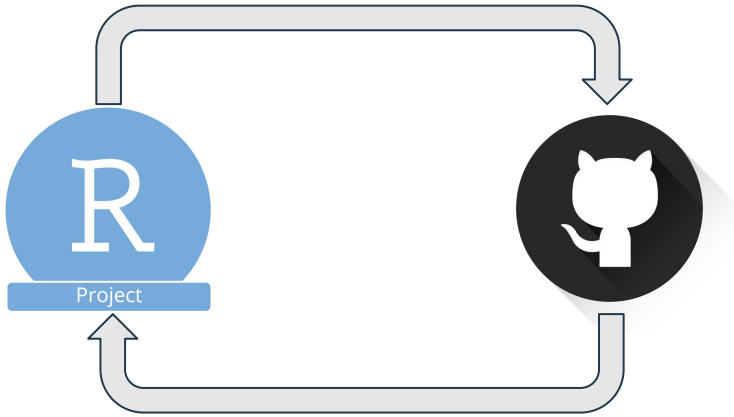



R
Project
Working
Directory
Code
git
Data &
Results
Packages

Order of Operations

Using GitHub as an example

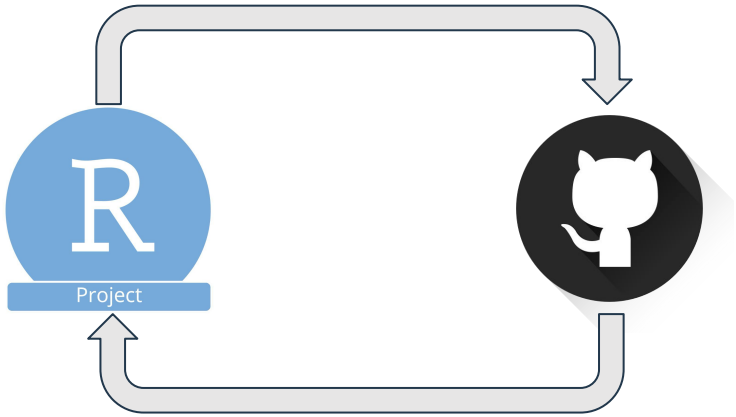
1. New Project, GitHub first
2. Existing Project, GitHub first
3. Existing Project, GitHub last



Order of Operations

Using GitHub as an example

1. **New Project, GitHub first**
2. Existing Project, GitHub first
3. Existing Project, GitHub last



Demo

1. Create a GitHub Repository
2. Pull in repository as an RStudio Project
3. Make a change to a file (e.g., README)
4. View changes in Git tab
5. Commit changes
6. Push to Github
7. Create a shiny app
8. Deploy app to Posit Connect from GitHub

Demo

1. Create a GitHub Repository
2. Pull in repository as an RStudio Project
3. Make a change to a file (e.g., README)
4. View changes in Git tab
5. Commit changes
6. Push to Github
7. Create a shiny app
8. Deploy app to Posit Connect from GitHub

Demo

1. Create a GitHub Repository
2. Pull in repository as an RStudio Project
3. Make a change to a file (e.g., README)
4. View changes in Git tab
5. Commit changes
6. Push to Github
7. Create a shiny app
8. Deploy app to Posit Connect from GitHub

Demo

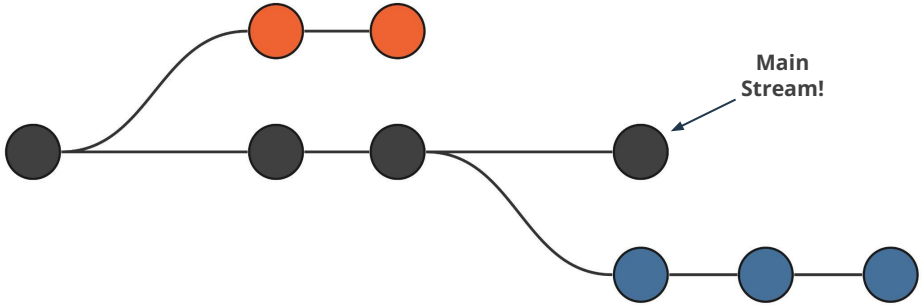
1. Create a GitHub Repository
2. Pull in repository as an RStudio Project
3. Make a change to a file (e.g., README)
4. View changes in Git tab
5. Commit changes
6. Push to Github
7. Create a shiny app
8. Deploy app to Posit Connect from GitHub

QUESTIONS?

git

RStudio®

A note on branches



- Branches allow you to take a detour from the main stream of development and do work without changing the main stream.
- Allows one or many people to work in parallel without overwriting each other's work.
- Allows someone working solo to work incrementally on an experimental idea, without jeopardizing the state of the main stream.