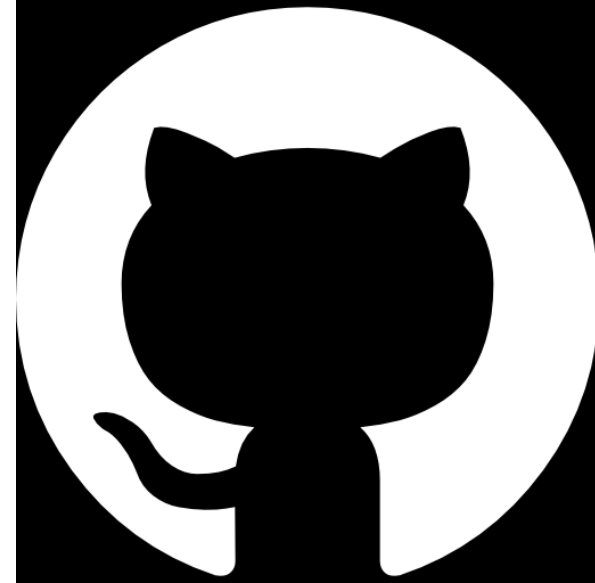


GitHub

- track changes on steroids



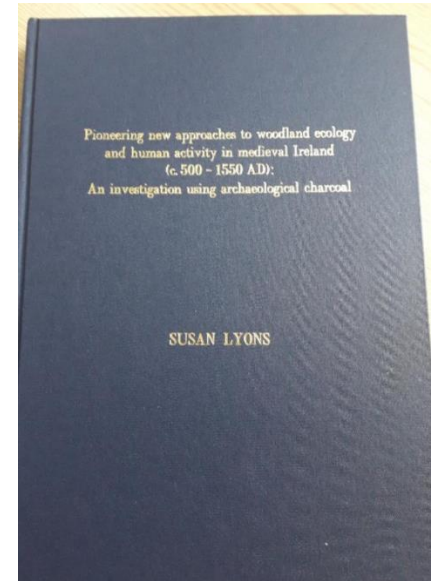
Brendan Palmer,

Statistics & Data Analysis Unit,

Clinical Research Facility - Cork

How is research presented?

Theses



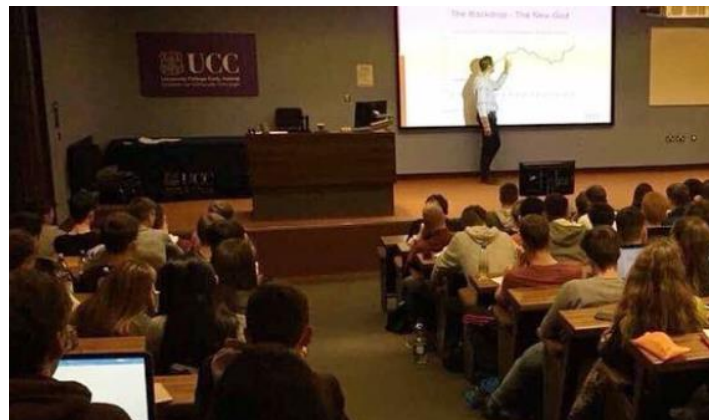
Books



Posters



Talks



Papers

Journal of Virology

Network Analysis of the Chronic Hepatitis C Virome Defines Hypervariable Region 1 Evolutionary Phenotypes in the Context of Humoral Immune Responses

Brendan A. Palmer,* Daniel Schmidt-Martin,* Zoya Dimitrova,* Pavel Skums,* Orla Crosbie,* Elizabeth Kenny-Walsh,* Liam J. Fanning*

Molecular Biology Diagnostic & Research Laboratory, Department of Medicine, University College Cork, Ireland†; Division of Viral Hepatitis, Centers for Disease Control and Prevention, Atlanta, Georgia, USA‡; Department of Hepatology, Cork University Hospital, Cork, Ireland§

ABSTRACT

Hypervariable region 1 (HVR1) of hepatitis C virus (HCV) comprises the first 27 N-terminal amino acid residues of E2. It is classically seen as the most heterogeneous region of the HCV genome. In this study, we assessed HVR1 evolution by using ultradeep pyrosequencing for a cohort of treatment-naïve, chronically infected patients over a short, 16-week period. Organization of the sequence set into connected components that represented single nucleotide substitution events revealed a network dominated by highly connected, centrally positioned master sequences. HVR1 phenotypes were observed to be under strong purifying (stationary) and strong positive (antigenic drift) selection pressures, which were coincident with advancing patient age and cirrhosis of the liver. It followed that stationary viromes were dominated by a single HVR1 variant surrounded by minor variants comprised from conservative single amino acid substitution events. We present evidence to suggest that neutralization antibody efficacy was diminished for stationary-virome HVR1 variants. Our results identify the HVR1 network structure during chronic infection as the preferential dominance of a single variant within a narrow sequence space.

IMPORTANCE

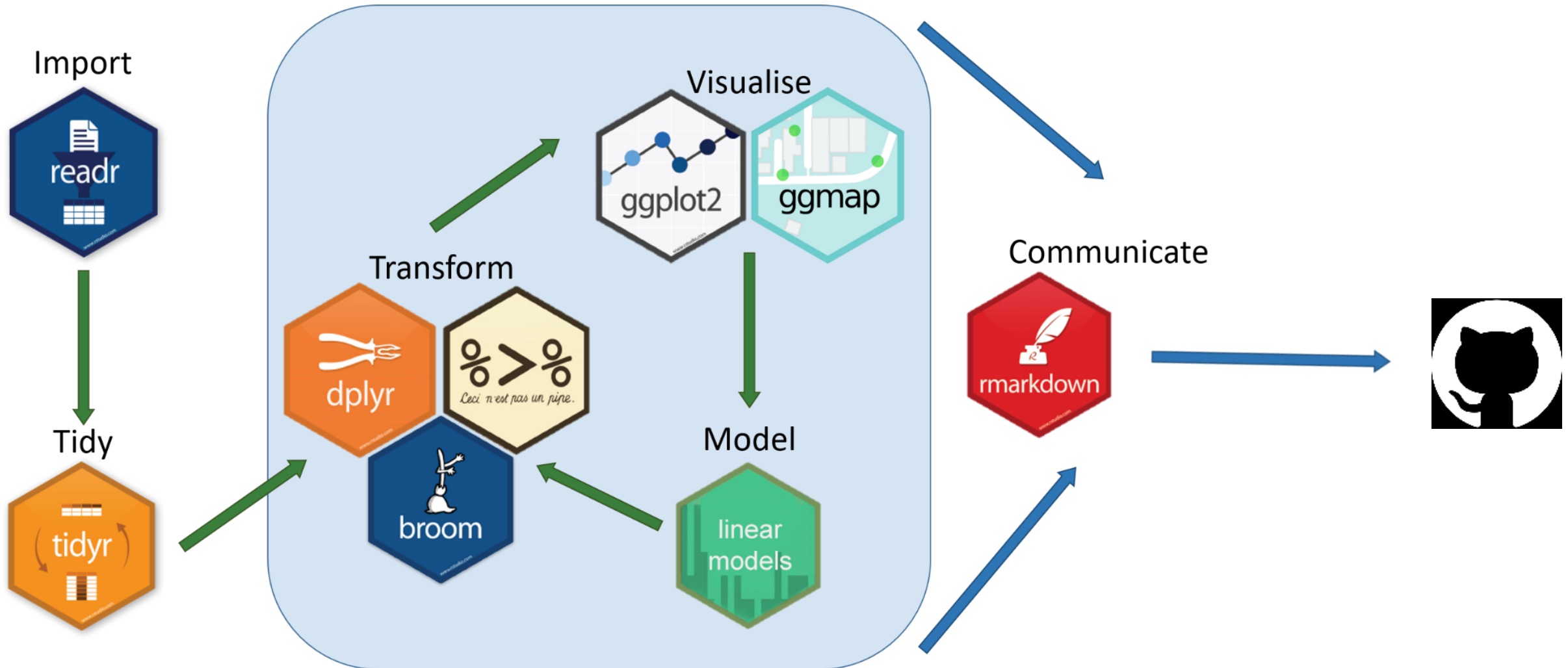
HCV infection is often asymptomatic, and chronic infection is generally well established in advance of initial diagnosis and subsequent treatment. HVR1 can undergo rapid sequence evolution during acute infection, and the variant pool is typically seen to diverge away from ancestral sequences as infection progresses from the acute to the chronic phase. In this report, we describe HVR1 viromes in chronically infected patients that are defined by a dominant epitope located centrally within a narrow variant pool. Our findings suggest that weakened humoral immune activity, as a consequence of persistent chronic infection, allows for the acquisition and maintenance of host-specific adaptive mutations at HVR1 that reflect virus fitness.

Hepatitis C virus (HCV) infection is a global health issue and is recognized as a major etiological agent of liver-related diseases (1). It has been estimated that the current prevalence of HCV represents approximately 2% of the global adult (15 years of age and older) population (2). Following transmission, HCV infection may remain asymptomatic for decades, resulting in the majority of infections initially passing undetected (3). It is estimated that up to 1 million Americans are living with the virus, the majority of whom became infected prior to the isolation and identification of the virus (4, 5). Consequently, the U.S. Centers for Disease Control and Prevention now recommend that Americans born from 1945 to 1965 be screened for the presence of the virus notwithstanding the presence of clinical symptoms (3, 5).

HCV is a single-stranded positive-sense RNA virus of considerable genomic heterogeneity. A recent reclassification defined the HCV global distribution into 7 genotypes and 67 subtypes, with genotypes 1 and 3 accounting for the majority of infections worldwide (6, 7). An error-prone RNA-dependent RNA polymerase, together with an inherent tolerance of defined hypervariable regions (HVR), accounts for much of this variability. Three HVRs are located within the envelope glycoprotein E2. The greatest heterogeneity has been identified at the 27-amino-acid HVR1 (residues 384 to 410 of the H77 reference strain), located at the amino-terminal end of the E2 glycoprotein (8). Recent studies indicated that the central region of E2 (residues 456 to 656) is globular and surprisingly compact, whereas the first 80 amino acids (including

Received 21 November 2015; Accepted 22 December 2015
Accepted manuscript posted online 10 December 2015
Citation: Palmer BA, Schmidt-Martin D, Dimitrova Z, Skums P, Crosbie O, Kenny-Walsh E, Fanning LJ (2016) Network analysis of the chronic hepatitis C virome defines hypervariable region 1 evolutionary phenotypes in the context of humoral immune responses. J. Virol. 90:3218–3228. doi:10.1128/JVI.02090-15
Editor: M. S. Diamond
Address correspondence to Liam J. Fanning, lfanning@ucc.ie.
BA-P and ZS-M contributed equally to this article.
Copyright © 2016, American Society for Microbiology. All Rights Reserved.

Putting the pieces together



Sounds complicated

- It isn't
- To get up and running takes only a few steps
- Any idiot could do it...



Brendan Palmer

bapalmer

Twitter: @B_A_Palmer

Edit bio

Organizations



Overview Repositories 4 Stars 0 Followers 1 Following 6

Popular repositories

[SDAU-Spring-2018](#)

Introductory R workshop: The tidyverse and beyond

HTML ★ 1 1

[SDAU-Autumn-2017](#)

First version of introductory R workshop: The tidyverse and beyond

R

Customize your pinned repositories

[propr](#)

Forked from ggloor/propr

Ways to measure strength of proportionality

R

[reproducible-workflows](#)

R

24 contributions in the last year

Contribution settings



What is git, gist, GitHub?



git is a version control system

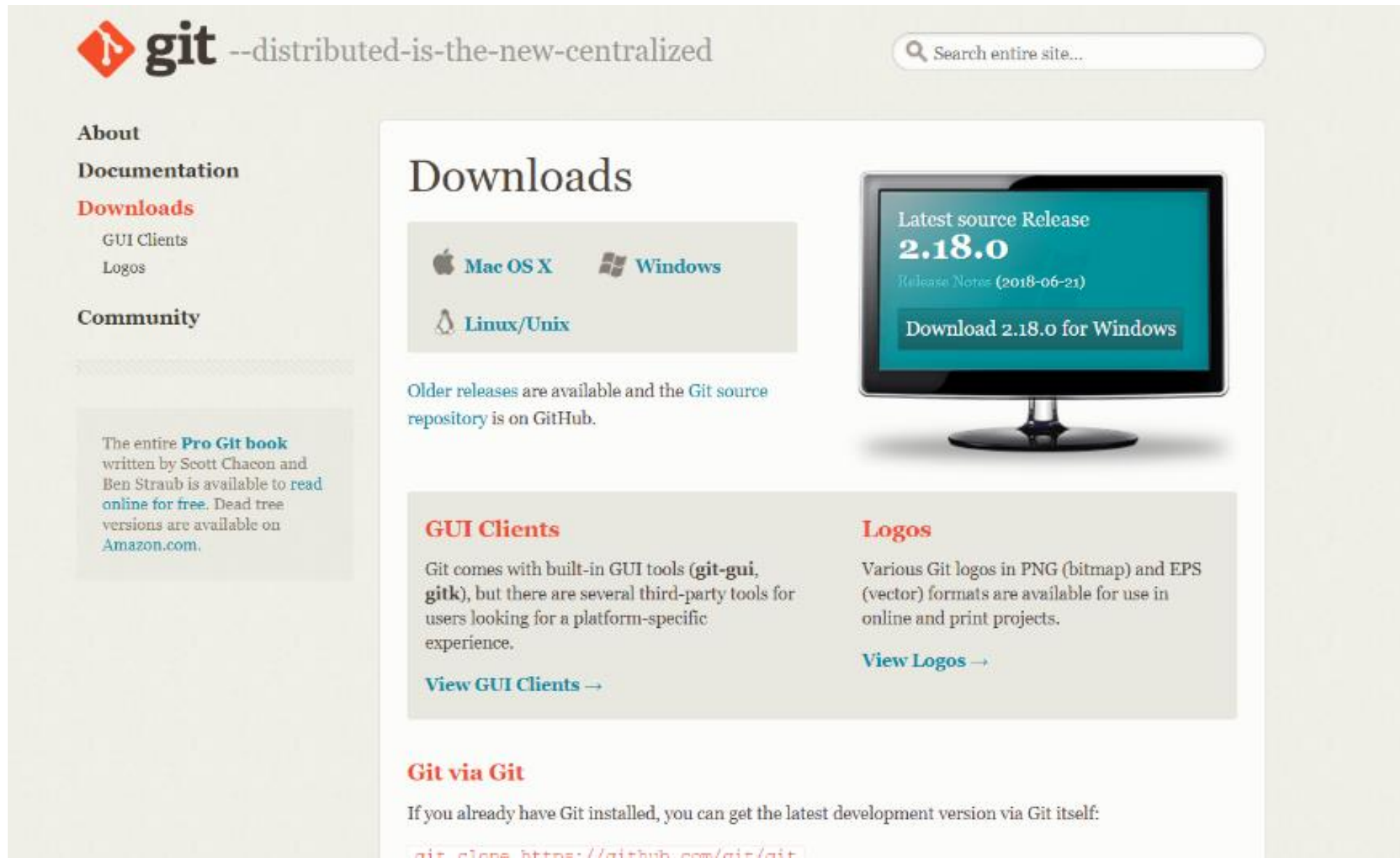
- originally designed to allow developers share code
- since repurposed to manage whole data projects
- gists are individual files or part of files that can be shared



GitHub is a hosting service for your Git projects

- remotely stored

Step 1: Install git after step 0, install R/RStudio



The screenshot shows the Git website homepage. At the top, the Git logo is followed by the tagline "--distributed-is-the-new-centralized". A search bar is located in the top right corner. The left sidebar contains navigation links: "About", "Documentation", "Downloads" (highlighted in red), "GUI Clients", "Logos", and "Community". Below these links, a box mentions the "Pro Git book" by Scott Chacon and Ben Straub, available for free online. The main content area features a "Downloads" section with icons for Mac OS X, Windows, and Linux/Unix. To the right of this is a monitor displaying the "Latest source Release 2.18.0" with a "Download 2.18.0 for Windows" button. Below the download section, there are links for "Older releases" and the "Git source repository" on GitHub. Further down, there are sections for "GUI Clients" and "Logos", each with a "View" link. At the bottom, a "Git via Git" section provides instructions on how to install Git using the Git command-line interface.

git --distributed-is-the-new-centralized

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read [online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads

Mac OS X Windows Linux/Unix

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.18.0
Release Notes (2018-06-21)
Download 2.18.0 for Windows

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

Git via Git

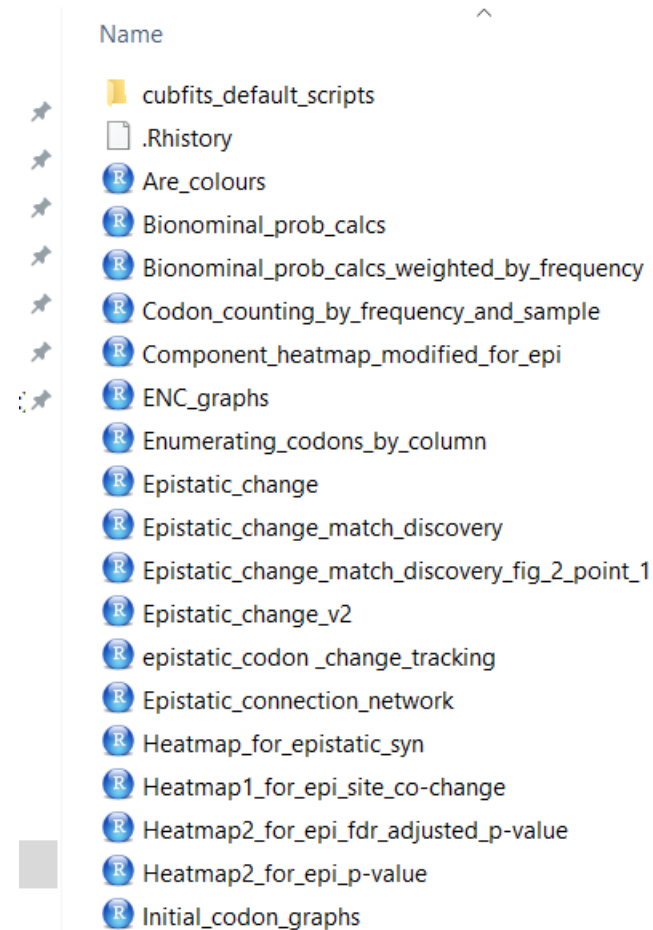
If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```


What's there to gain

- A record of your research
- Avoid that chaotic state where you have multiple copies of the same file, but without a stupid and annoying file naming convention.

Documents > MSc_Bioinformatics > MSc_Project > R_scripts

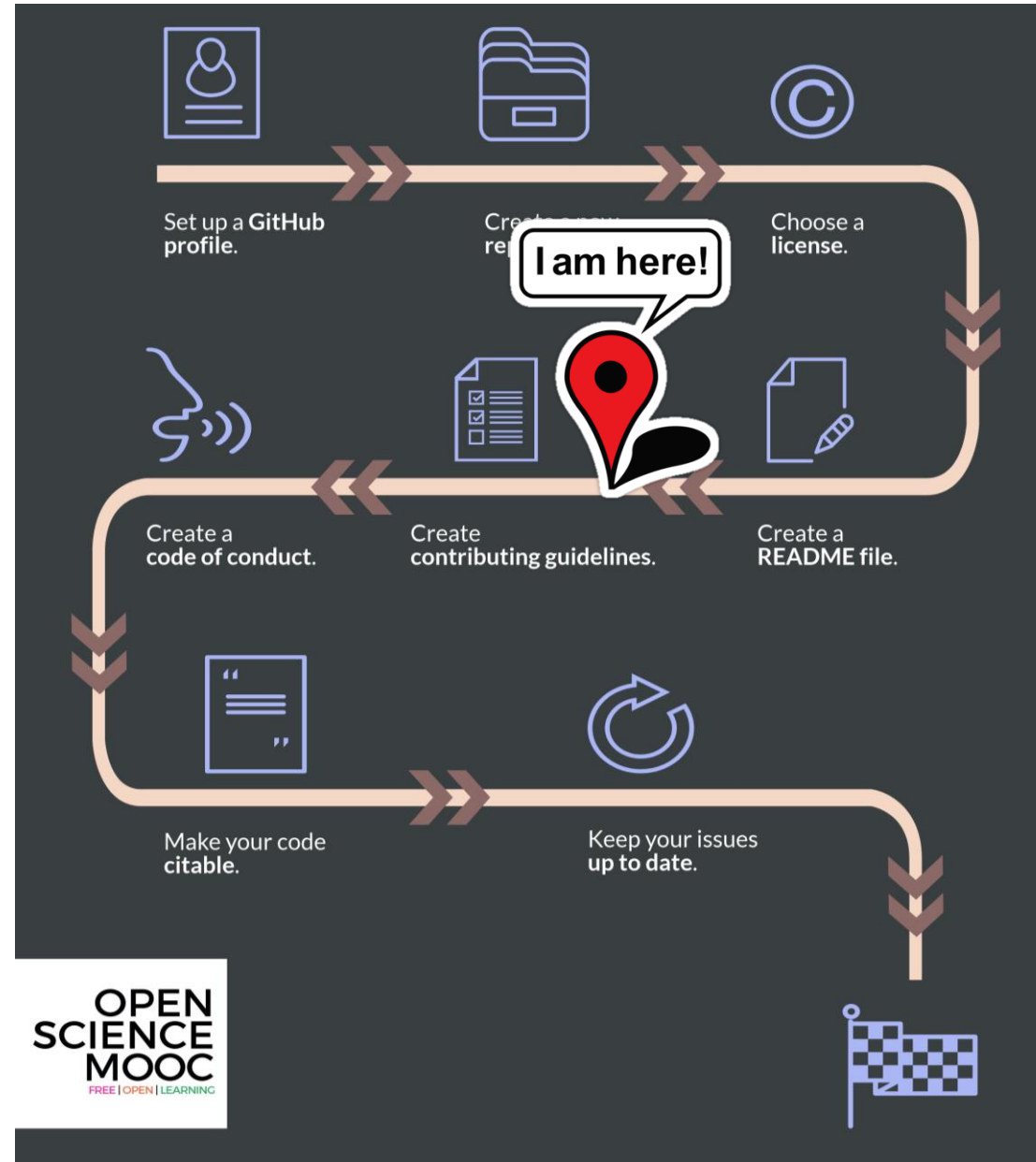


Final Final version

File Home Share View

← → ↕ ⬆ ⬇ This PC > B_Palmer_Medicine_Files > 4a Project > Pyrosequencing_analysis > Pyrosequencing_Paper > Draft_Paper_incl_Figs > Submission > JVI_Resubmission > JVI_resubmission_files > Final Final version Final Final version

Getting started with GitHub



Step 2: Go to GitHub and create a repository

Overview **Repositories 4** Stars 0 Followers 2 Following 6




Find a repository...

Type: All ▼

Language: All ▼

 New

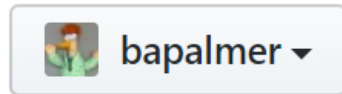
reproducible-workflows

 R  2  MIT License Updated 3 hours ago



Step 3: Give it a useful name


Owner




Repository name

Great repository names are short and memorable. Need inspiration? How about **probable-octo-potato**.

Description (optional)

☒  **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.

Step 4: Populate the README file

README.md



SDAU-Spring-2018

Introductory R workshop: The tidyverse and beyond

The lecture notes and R scripts given below are from an introductory R workshop hosted by the HRB CRF-C in the Spring of 2018. The lecture notes are punctuated with prompts to jump into the R scripts as you go and to see what we're talking about. All the scripts are composed in such a way, so that you can run each line yourself and get the outputs you expect. At certain points, there are exercises for you to complete yourself, which hopefully will allow you to get a feel for writing your own code.

Workshop 1:

First time users are given a crash course in this session, broadly along the lines of that described by David Robinson here (<http://varianceexplained.org/r/tidy-genomics/>). We proceed to take it on from there. The lecture notes discuss the need for learning R, especially if you are engaging in research in the life sciences. We follow this up with an overview of R, R packages and the basic steps of data analysis.

The structure of the tidyverse code is outlined and the fundamental ideas of tidy data are described. You won't learn how to use R in this session, but hopefully you will get a sense of how it can do powerful things with very little effort.

Workshop 2:

We go through the, must know, basics of the R language including data types, indexing and having a go at importing data into RStudio.

Step 5: Add a licence

bapalmer / reproducible-workflows

Watch

2

Star

8

Fork

0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

Branch: master

reproducible-workflows / LICENSE

Find file

Copy path



bapalmer/reproducible-workflows is licensed under the
MIT License

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions

- ✓ Commercial use
- ✓ Modification
- ✓ Distribution
- ✓ Private use

Limitations

- ✗ Liability
- ✗ Warranty

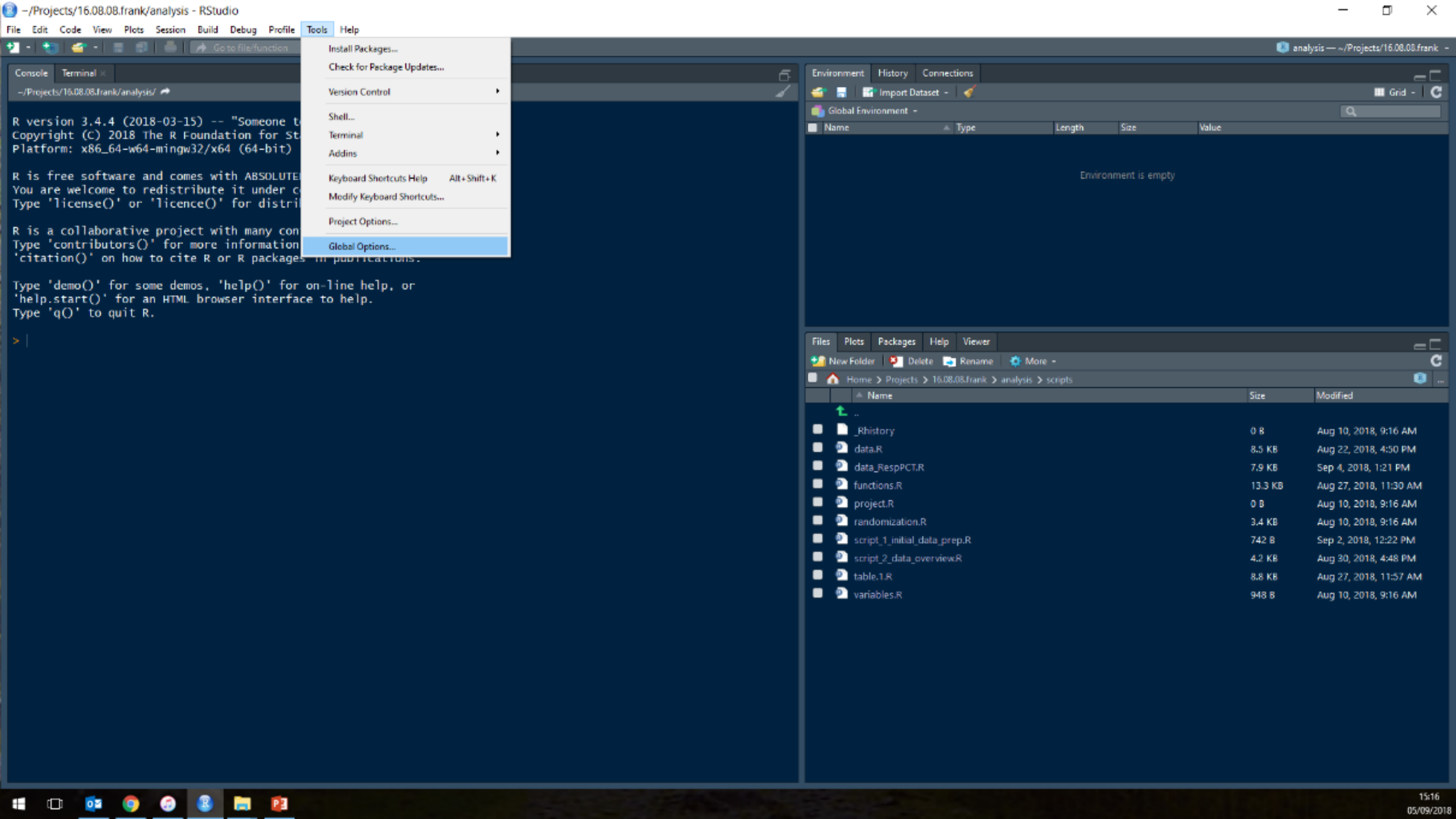
Conditions

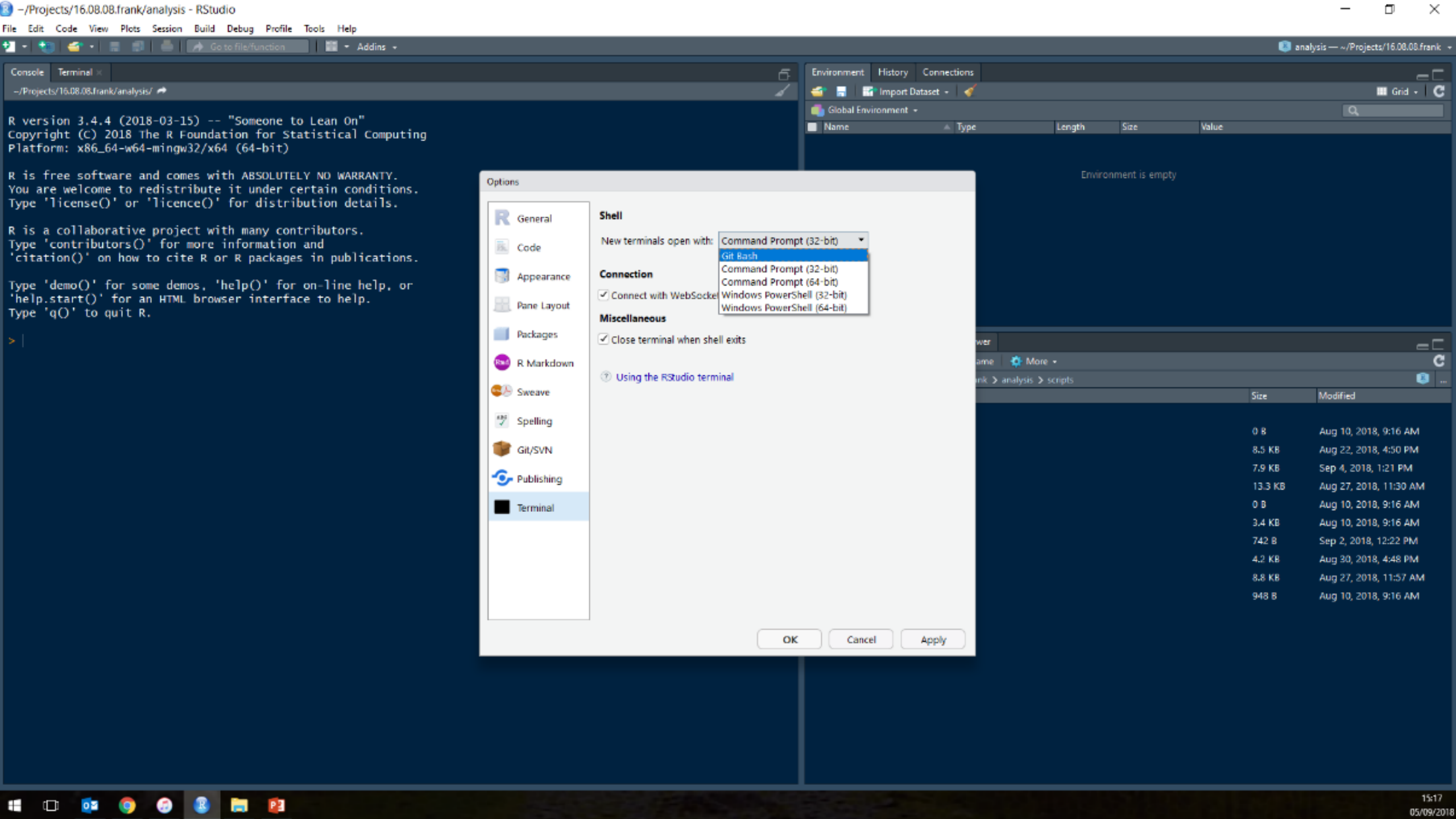
- ℹ License and copyright notice

This is not legal advice. [Learn more about repository licenses.](#)

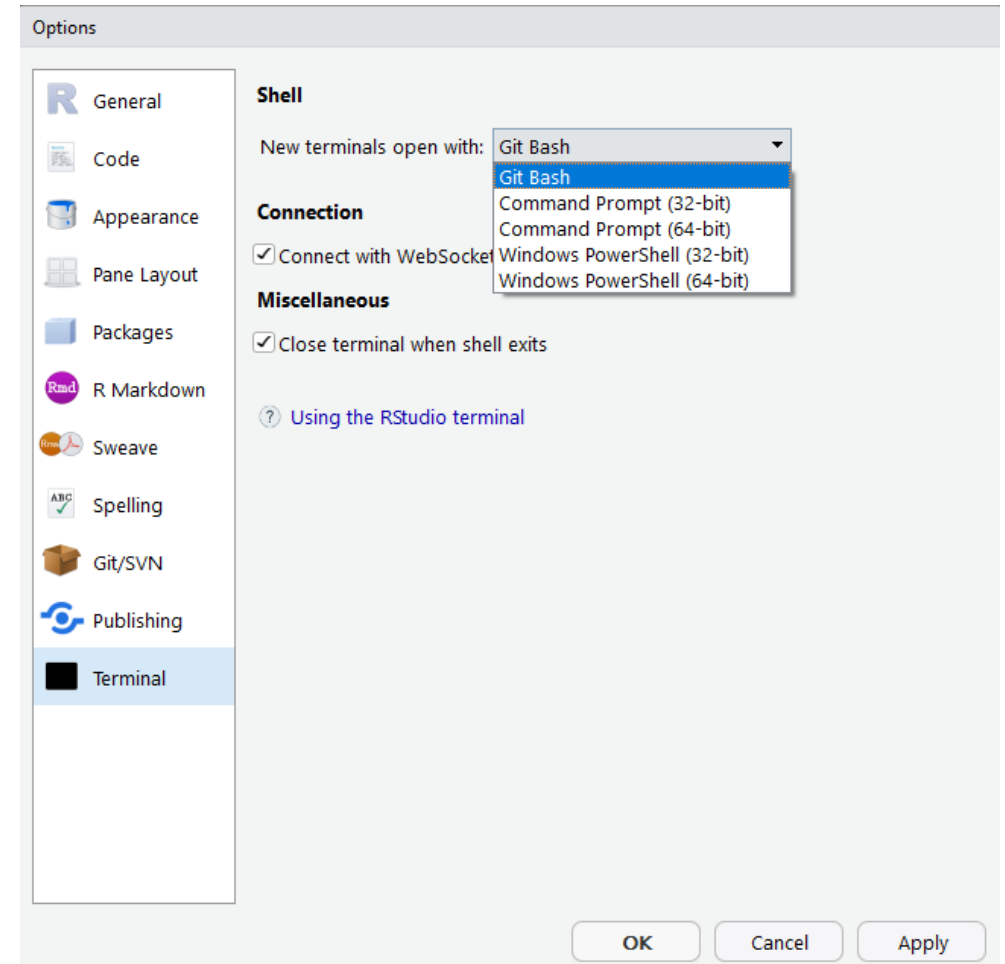
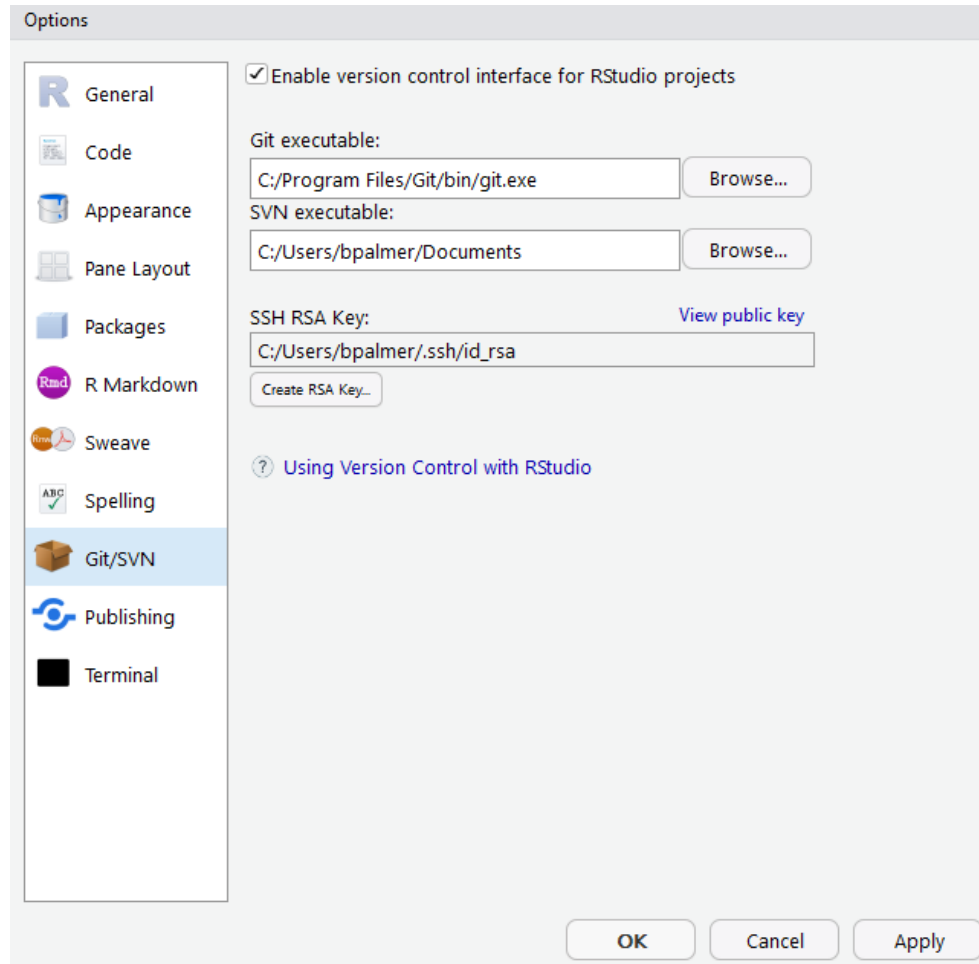
Step 6: Interface git with RStudio

- The key here is to select the correct shell
 - Windows users have four options
 - Only one of these is correct



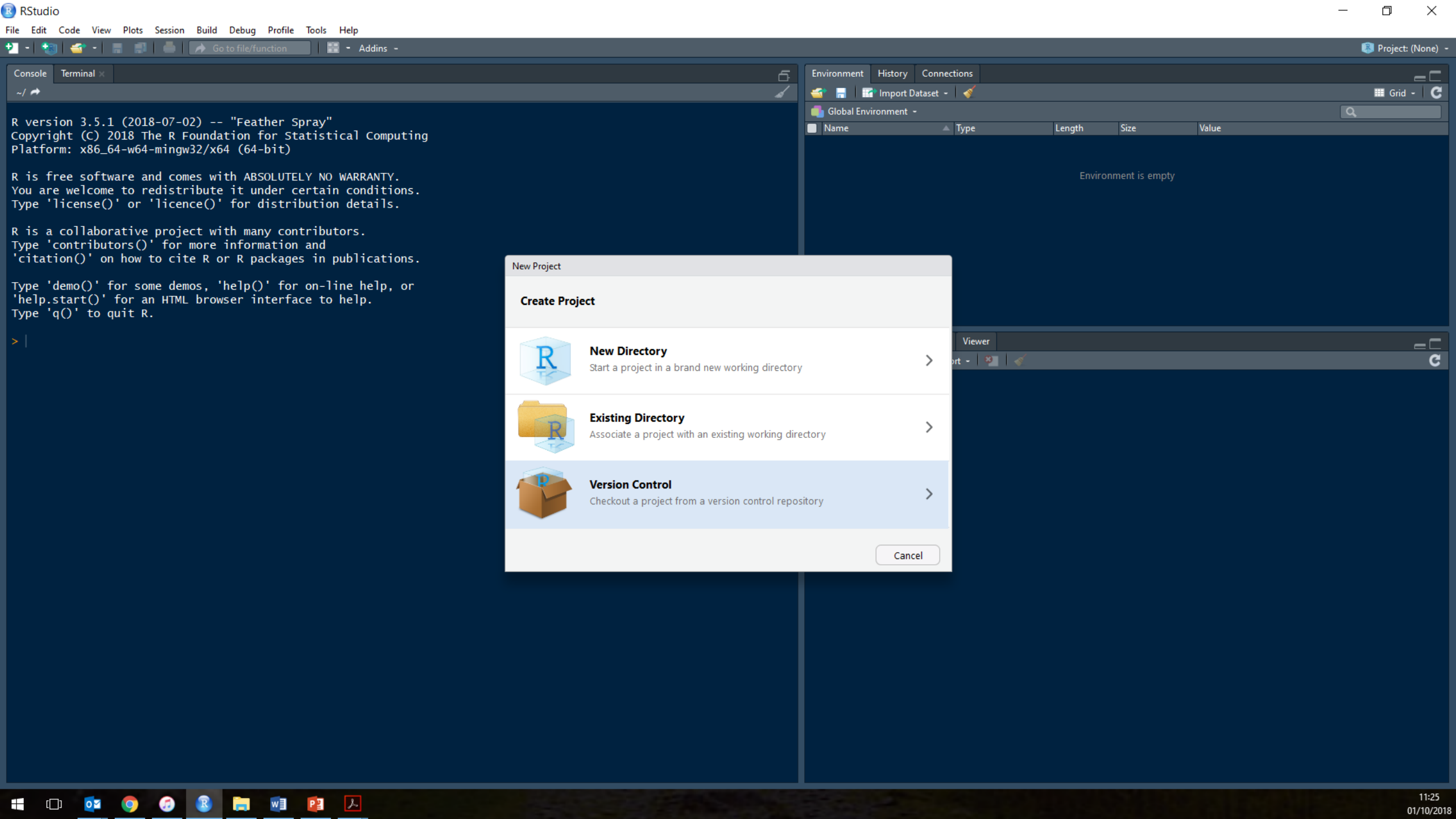


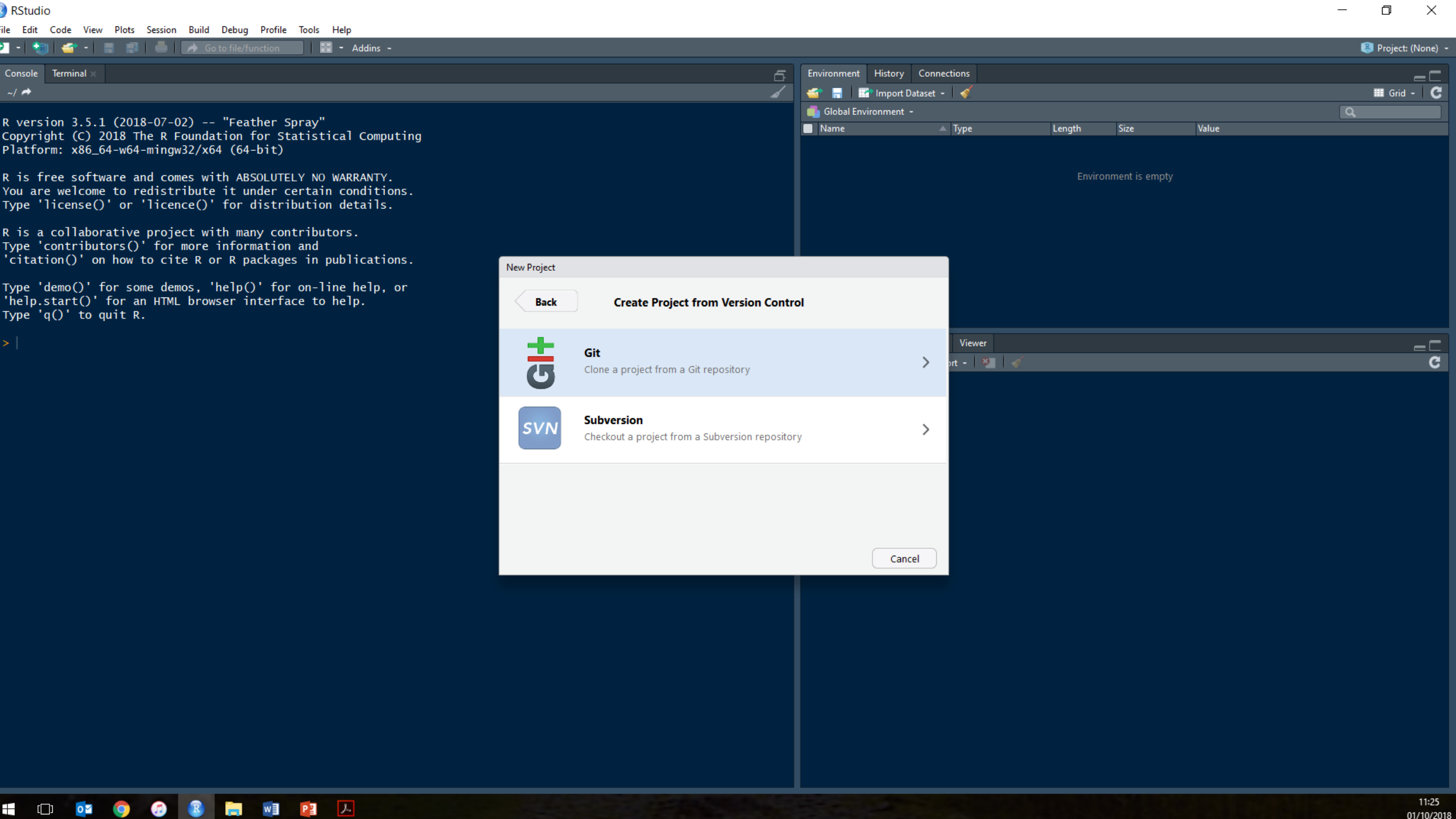
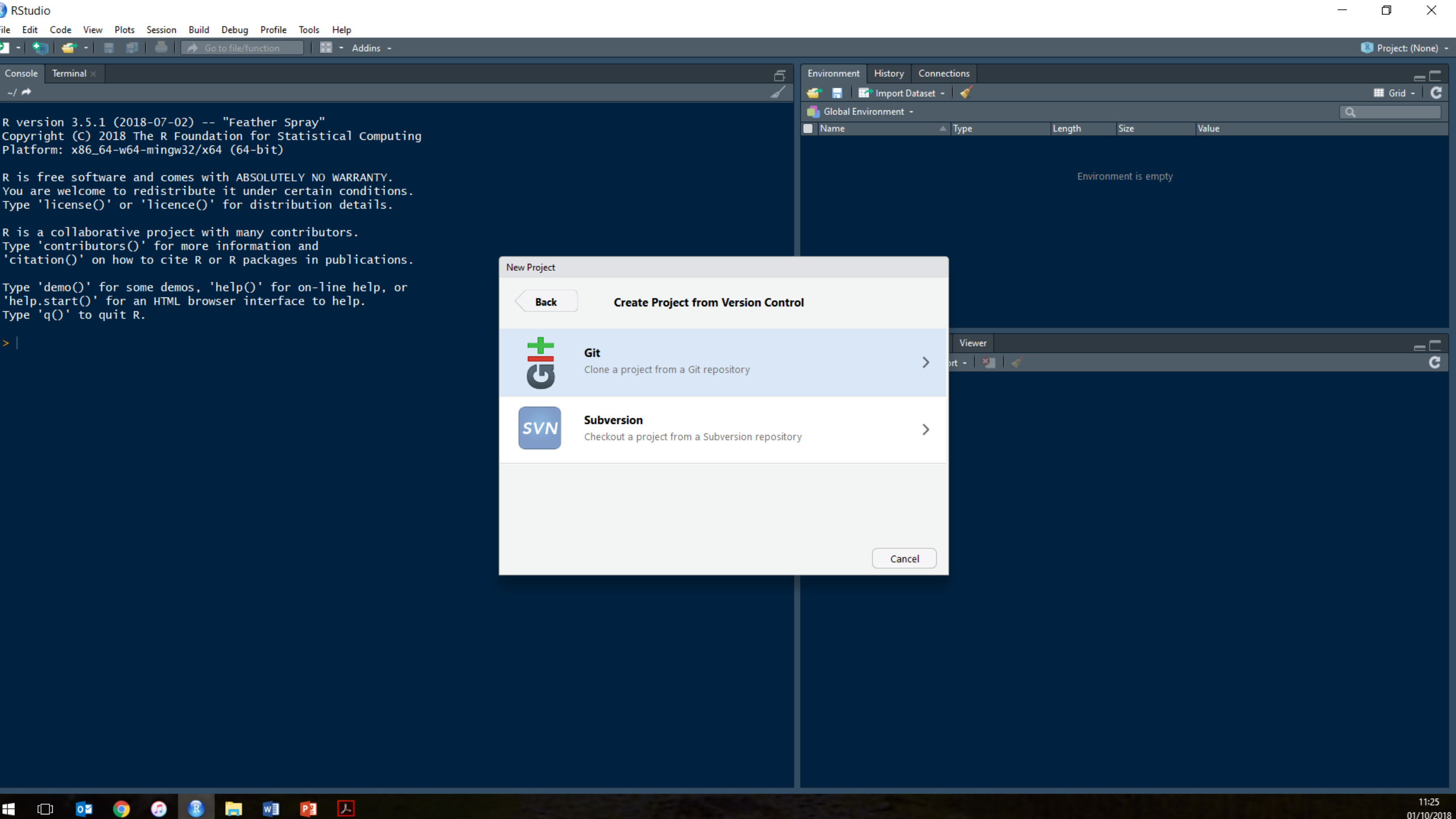
Step 6: Interface git with RStudio

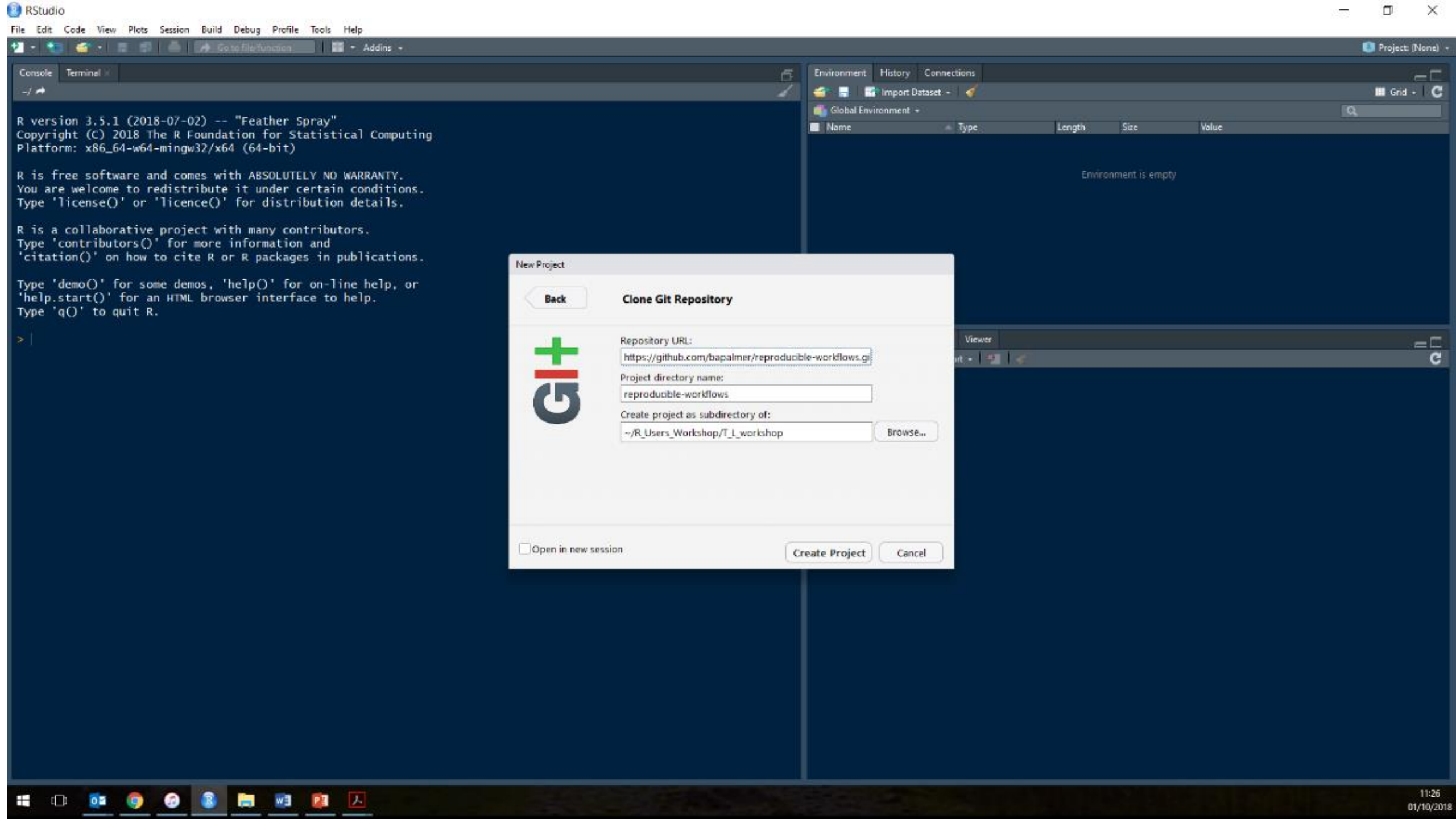


Step 6: Interface git with RStudio

- Setup a version control R project to interface with the newly created repository

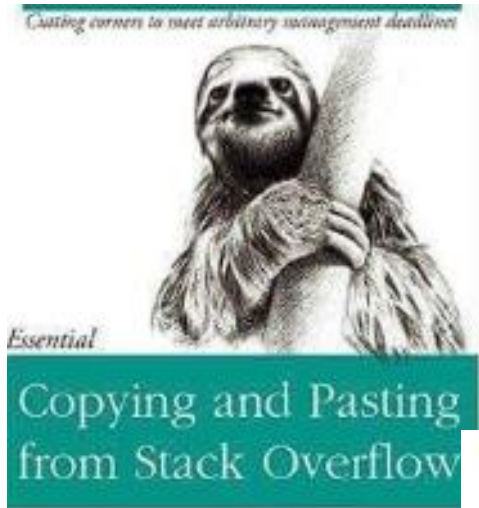






The screenshot displays the RStudio interface. The top pane, labeled 'Environment', is empty, showing a table with columns 'Name', 'Type', 'Length', 'Size', and 'Value'. Below this, the 'Files' pane shows the directory structure: 'Home > R_Users_Workshop > T_L_workshop > reproducible-workflows'. The file list includes '..' (green upward arrow), '.gitignore' (44 B, Oct 1, 2018, 11:27 AM), 'LICENSE' (1.1 KB, Oct 1, 2018, 11:27 AM), 'README.md' (24 B, Oct 1, 2018, 11:27 AM), 'reproducible-workflows.Rproj' (218 B, Oct 1, 2018, 11:27 AM), and 'reproducible-workflows' (folder).

From the people who brought you.....



O'REILLY*

*The Practical Developer
@ThePracticalDev*

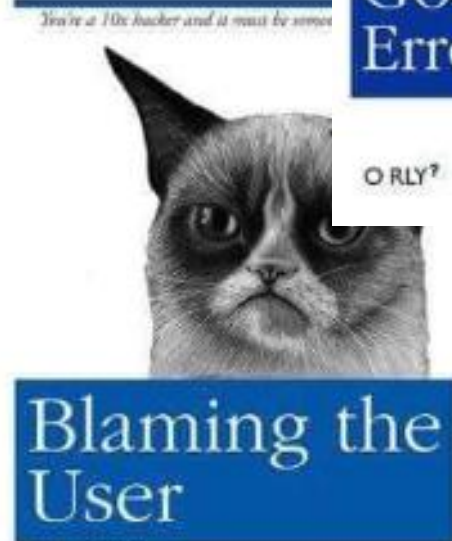


O RLY?

*The Practical Developer
@ThePracticalDev*

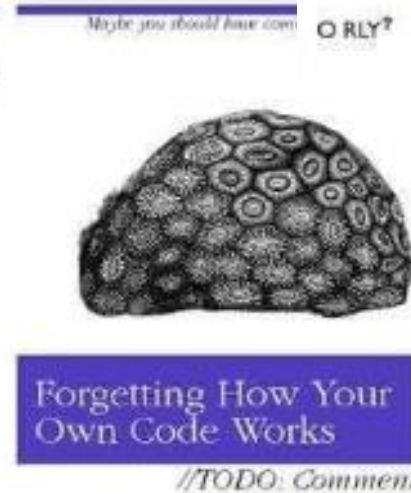


*The Practical Developer
@ThePracticalDev*



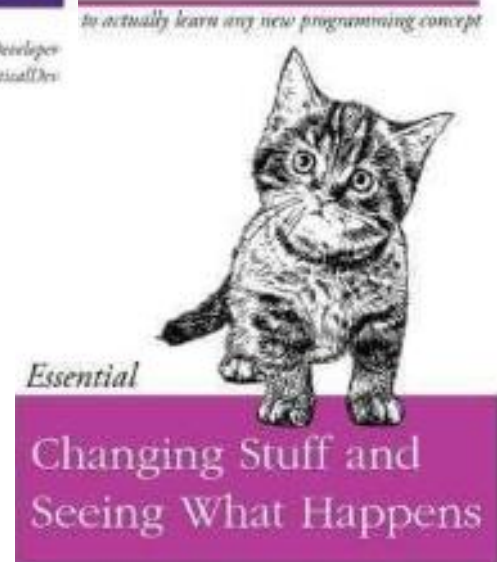
O RLY?

@ThePracticalDev



O RLY?

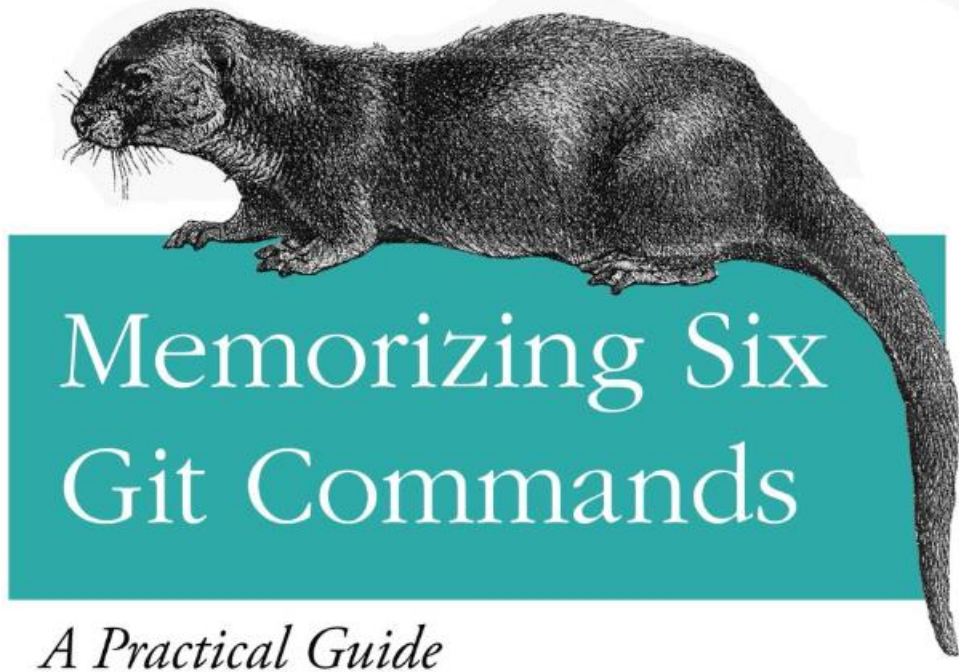
FunctionZero



O RLY?

@ThePracticalDev

I know git, depending on what you mean by “know”



JD Long
@CMastication

Following

Um... So what are the other 2?



DEV Community 🧑🧑 @ThePracticalDev
I know Git, depending on what you mean by "know".

6:39 PM - 19 Aug 2018

1 Retweet 36 Likes



💬 5

↻ 1

❤️ 36



JD speaks the truth

4 main commands just to get started with.....

- **Add:** This is where you identify the files you want to upload
- **Commit:** This is like "saving" your work by creating a new version or copy
- **Push:** This is how you send files from your local project to the online repository
- **Pull:** This is how you get files from your online repository to your local project

THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

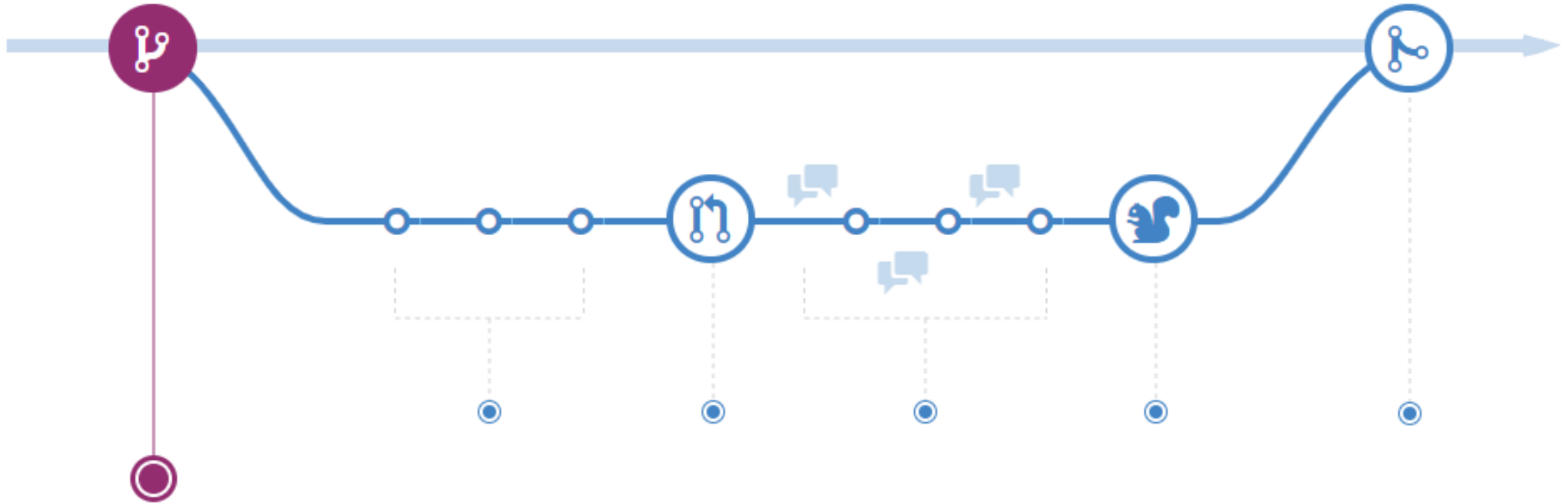
NO IDEA. JUST MEMORIZE THESE SHELL
COMMANDS AND TYPE THEM TO SYNC UP.
IF YOU GET ERRORS, SAVE YOUR WORK
ELSEWHERE, DELETE THE PROJECT,
AND DOWNLOAD A FRESH COPY.



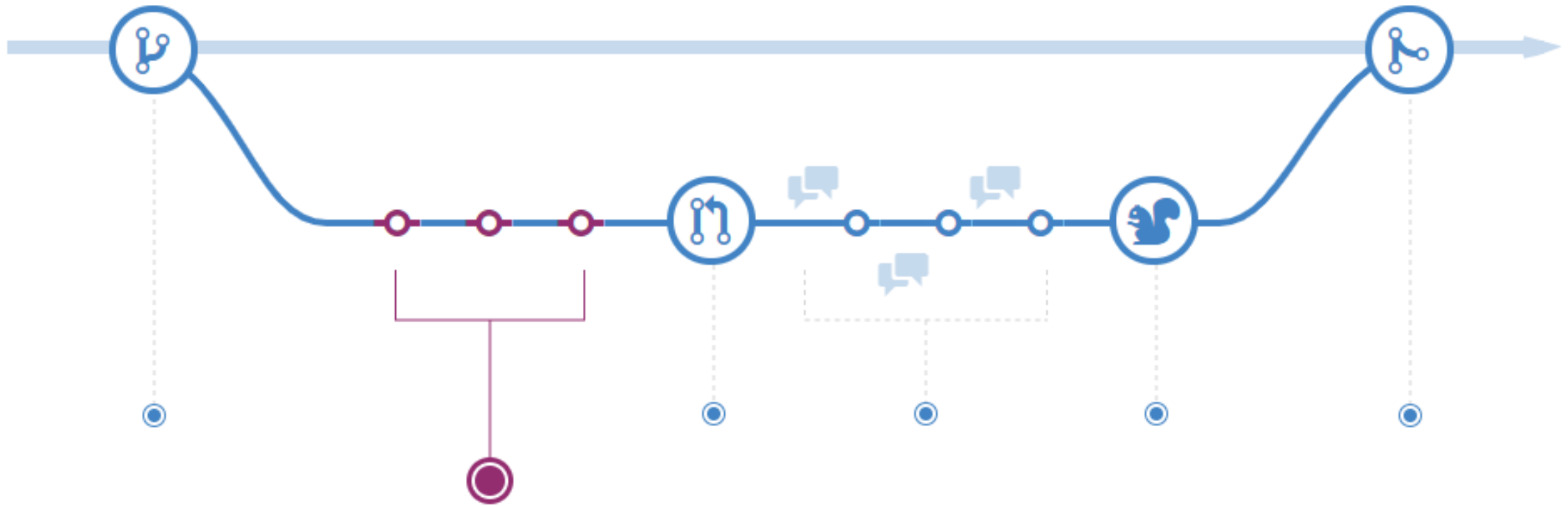
Obviously there are others

Initialise
Checkout
Clone
Fork
Branch
Master
Head
Index
Merge
Stage
Issue
Fetch
Clean
Check
Status
Commit message
Working directory

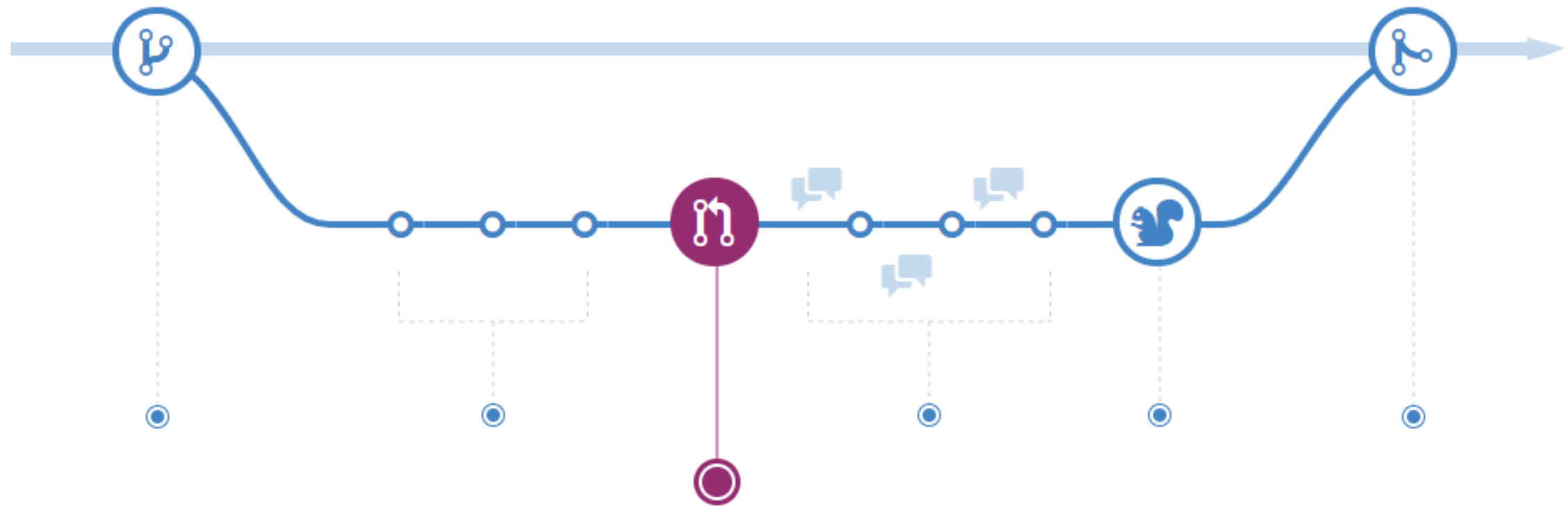
Understanding the GitHub flow: Branch



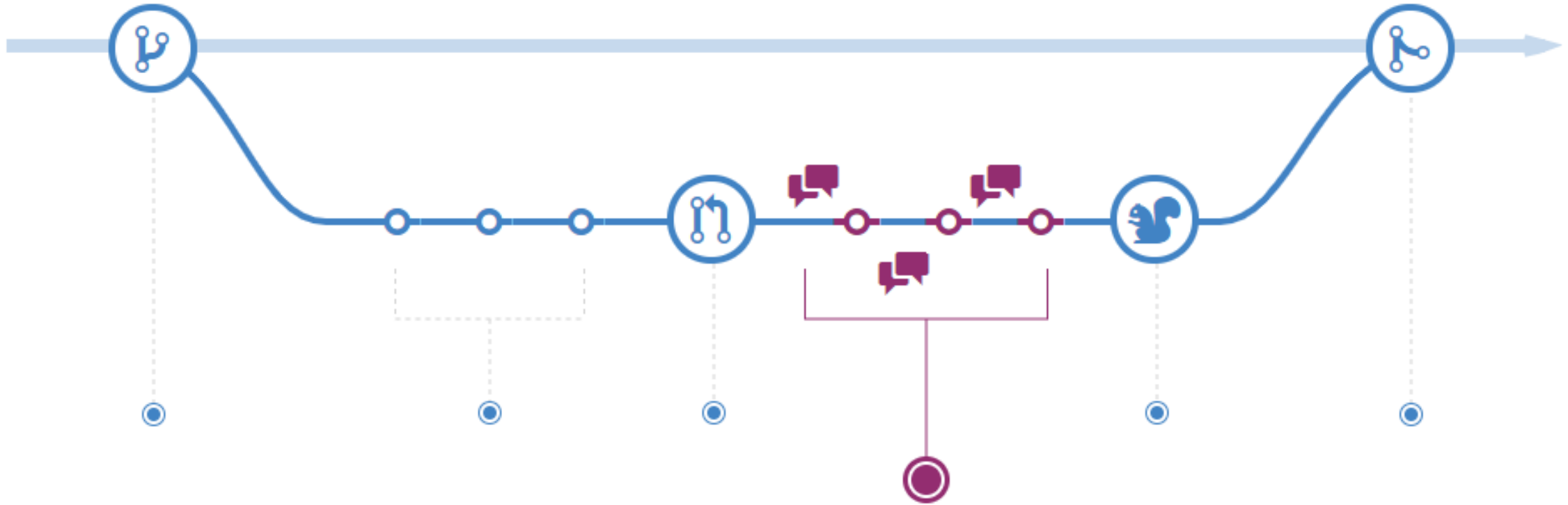
Understanding the GitHub flow: Commit



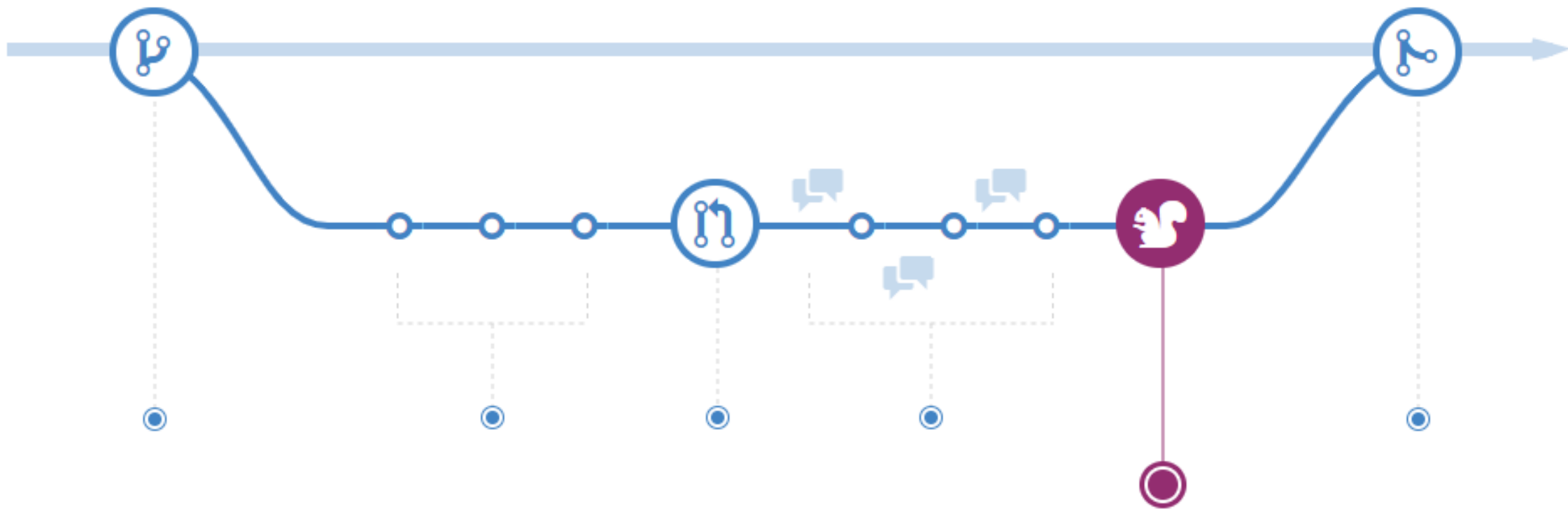
Understanding the GitHub flow: Pull request



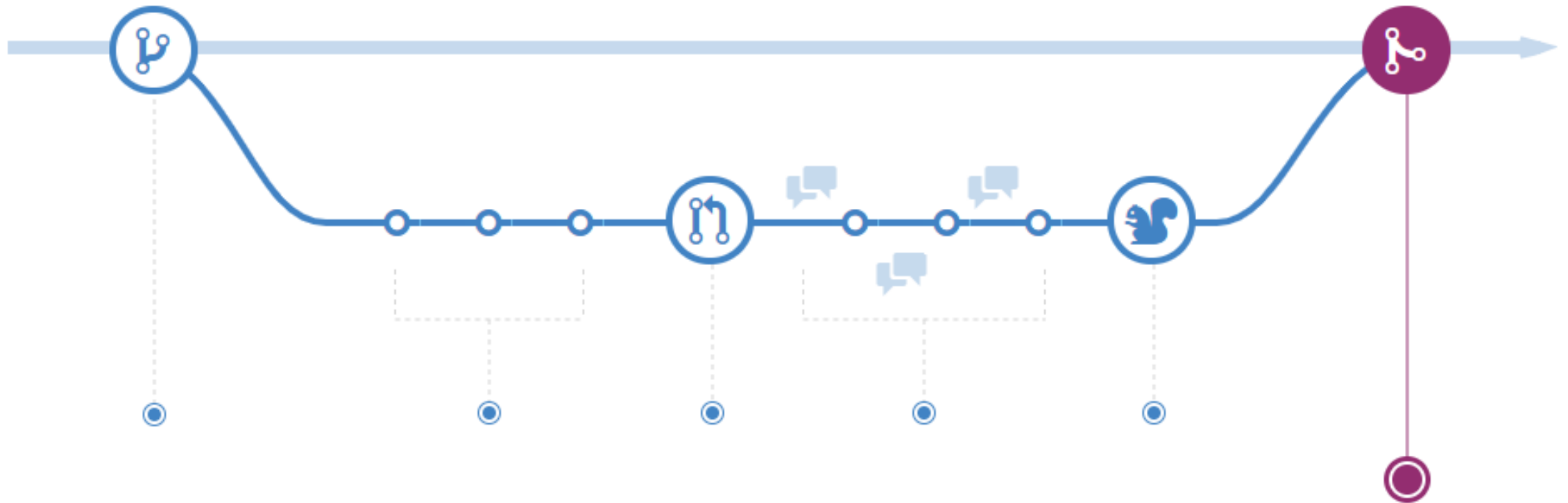
Understanding the GitHub flow: Review



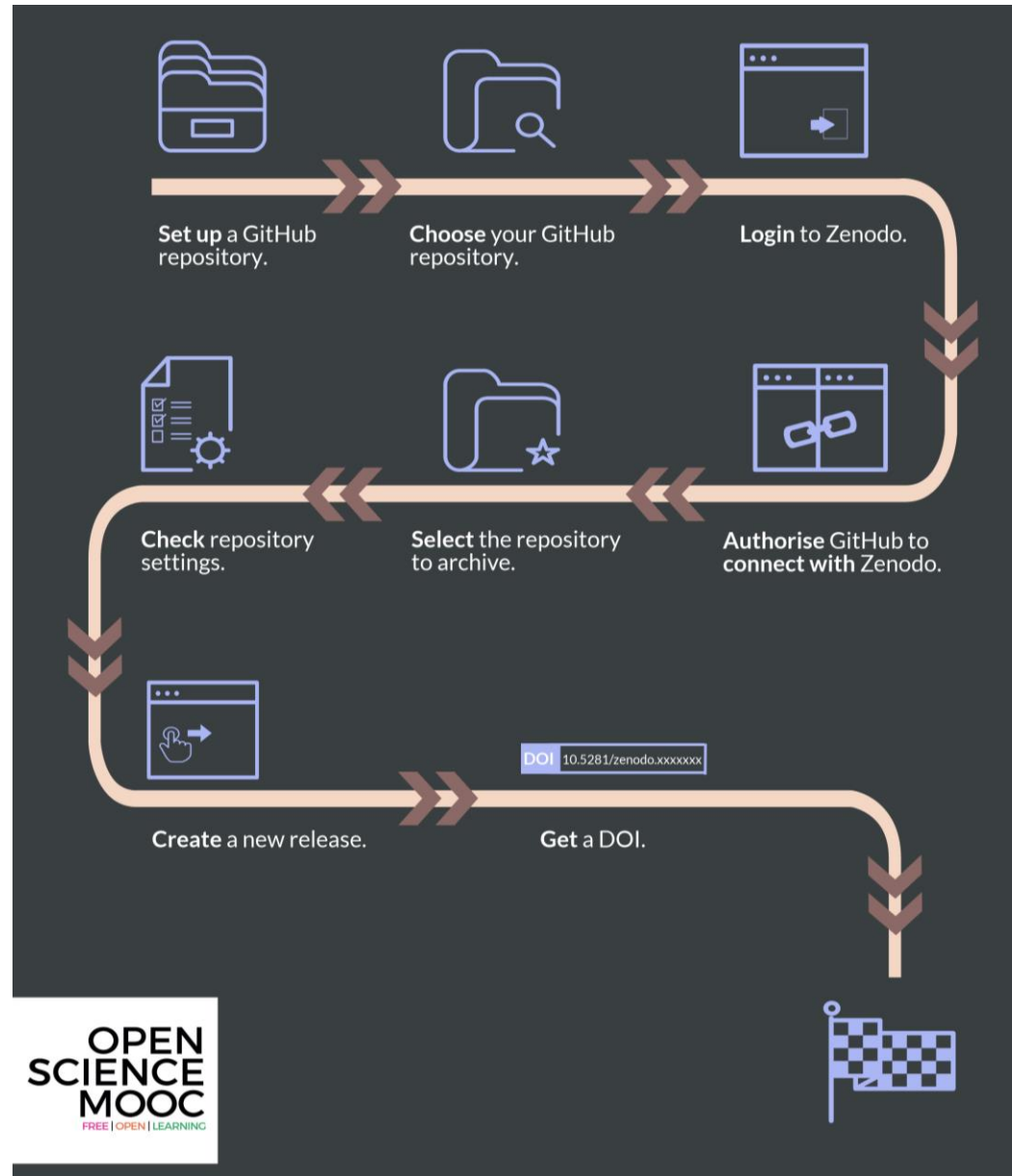
Understanding the GitHub flow: Deploy



Understanding the GitHub flow: Merge



Finishing up



Try it for yourself: Tasks 1-3

OPEN SCIENCE MOOC

FREE | OPEN | LEARNING