

# Version control with Git

Pariksheet

January 29, 2016

## 1 Setup before class

### 1.1 Software

You will need:

1. Git installed,
2. An account with *GitHub*, and
3. SSH keys

It's possible that you have most of this already setup per Robi's *GettingStarted* instructions, however you would likely need to complete the last item, namely registering your ssh keys with *GitHub*.

Registering your ssh keys with *GitHub* allows you to upload your code, etc changes to *GitHub* without having to type in your username and password each time.

To check whether your keys are registered, see whether *GitHub* greets you with your registered username per: <https://help.github.com/articles/testing-your-ssh-connection/> (Windows users should open the *Git Bash* terminal app; just hit the Windows key, type *Git Bash* and hit enter)

If the terminal command output indicates that *GitHub* does not recognize your username, check whether your SSH keys exist. One way to do this is check inside of RStudio via **Tools > Global Options... > Git/SVN** and you should see SSH RSA Key with something like `~/.ssh/id_rsa` (basically, that field should not be blank). If that SSH RSA Key field is blank, click the button to create the RSA key, and you can skip entering a passphrase. Windows users may encounter an error here since the Windows filesystem does not like to create anything starting with a dot, but one can create the `.ssh` directory with one command in *Git Bash*: `mkdir -p ~/.ssh`

### 1.2 15 minutes of practise

Get a taste of quick Git with the *GitHub* web tutorial: <https://try.github.io/>

As you type out and run each command, scroll down to see your changes in the file browser (lower left) and read the advice box (lower right).

If you don't understand everything that's happening, that's totally fine; we will discuss the concepts on Tuesday.

## 2 Tuesday's Lesson Plan

Getting started and on your way to using Git is mostly about learning the concepts of version control. To learn the concepts we will use Git from the terminal, since *RStudio*'s Git integration requires some moving around, which starts to make more sense after one is better acquainted with Git.

We will cover:

- 9 essential commands for Git
- Collaborating on *GitHub*: issues, comments, integrations
- (If we have time) Resolving conflicts, hooks

## 3 Useful Links

### 3.1 Keep these next to your pillow

- Man pages: <http://www.git-scm.com/docs>
- Git Flight Rules: <https://github.com/k88hudson/git-flight-rules>

### 3.2 Tutorials and Refreshers

- Software Carpentry: <http://swcarpentry.github.io/git-novice/>
- Atlassian docs: <https://www.atlassian.com/git/>