Getting started with Git and GitHuib

Becca Freese and Michael Shyne

October 31, 2018

Presentation Outline and Ideas

1. Why you should use version controls
   1. Work history, easily revert to an older version of a file
   2. Collaboration!
2. Getting Started
   1. Git – Type *git version* in a terminal window to see if you already have git installed
      1. Mac: an old version comes already installed, may want to get a newer version
      2. Windows: does not come installed.
   2. GitHub or Bitbucket
      1. A place to keep and manage files
   3. Source Tree
      1. Git GUI if you don’t want to use the command line
3. Essential Git Commands/Vocabulary/Concepts
   1. Repository
      1. A place locally or on GitHub to store your files.
   2. Keys
      1. How you connect your local repo to the remote (GitHub)
      2. SSH vs HTTPS
   3. git status – your new best friend!
      1. Check that your folder is being tracked, which files have been modified, etc.
   4. git clone
      1. get a copy of a copy of a git repository from GitHub
      2. useful in collaboration when someone already has a repo started
   5. git init
      1. initiate an existing file as a git repository
   6. git add
      1. specify which files you want to commit
   7. git commit – commit changes you make
      1. Explain ‘what’ and ‘why’, not ‘how’
      2. Commit frequently to easily identify where something went wrong
      3. <https://chris.beams.io/posts/git-commit/>
   8. git push
      1. add changes from your local repo to the remote
   9. git pull
      1. get changes other people made on your local copy from the remote
   10. git branch
       1. Master – always deployable
       2. Development – works in progress
4. Example: Starting a new repository from Github
5. Example: Starting a new repository in R Studio and adding to GitHub
6. Example: Start tracking an already existing project

(Just ideas of things I’ve had to learn over the years)

