Connecting R to GitHub Outline

* Going to be talking about using R Projects and GitHub
  + Together, like a good wine and cheese or peanut butter and bananas
  + Acknowledging sources
    - There are many ways to do this efficiently
* Why R Projects?
  + Everything meaningful in one folder structure
    - a folder on your computer that holds all the files relevant to that particular piece of work. I’m
  + Working directory is automatically set with each R Proj
    - The chance of the setwd() command having the desired effect – making the file paths work – for anyone besides its author is 0%. It’s also unlikely to work for the author one or two years or computers from now.
  + Allows for reproducibility (for others AND FOR YOURSELF)
  + How to structure
    - Project-based directory organization
    - Activity-based directory organization
* Why Use GitHub?
  + Version Control
    - Ever made an error in your code or deleted some portion of important analysis or data cleaning?
    - Academics
      * A picture containing timeline

        Description automatically generated
    - Data analysis
      * <https://www.youtube.com/watch?v=s3JldKoA0zw>
  + Connecting to GitHub
    - Pre-Req – have a GitHub account; download git/gitbash; idea of project you want to do

 3 main commands:

* pull: update your local project with the latest version of the main project
* commit: snapshot of your code at a specified point in time
  + - * A commit message is descriptive text that is added to the commit object by the developer who made the commit.
        + To help a future reader quickly understand what changed and why it changed
        + To assist with easily undoing specific changes
        + To prepare change notes or bump versions for a release
      * Graphical user interface, text, application, email

        Description automatically generated
      * Imperative mood – make abc do xyz. Verb in the present tense
* push: merge your local modifications with the main project

 Simple (solo) use of git to prevent merge conflicts:

* after opening a project, always pull
* before closing a project, always commit/push
* Create a GitHub Account (if you do not already have one)
  + Advice on creating a username - <https://happygitwithr.com/github-acct.html>
  + You will need to know your username, email, and password for GitHub
* Download Git (if you do not already have it installed)
* https://git-scm.com/downloads
  + Windows users additional instructions - <https://appuals.com/what-is-git-bash/>

**DO ABSOLUTELY NOTHING ELSE**

* Connect git/gitbash with Github
  + Set up SSH KEY - <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>
* Connect with R (https://cfss.uchicago.edu/setup/git-configure/)
  + Identify yourself (only have to do this once per machine)
    - usethis::use\_git\_config(user.name = "Benjamin Soltoff", user.email = "ben@bensoltoff.com")(email address needs to be associated with your GitHub account)
  + Cache credentials
    - Need to authenticate yourself
    - SSH key pair
      * credentials::ssh\_setup\_github()
      * prompted to generate a new SSH key. Tell the computer Yes
      * then copy and paste the public key (the whole line of text) into the resulting browser window
    - HTTPS (if you are running R and Git on your personal computer, this is the recommended method)
      * usethis::create\_github\_token() – creates a personal access token
      * I usually set the token to never expire. GitHub will wanr you this is not as secure an option
      * I store this in a .csv
      * Store your PAT
        + gitcreds::gitcreds\_set()
        + This way you don’t have to reenter it every time you interact with Git
      * Confirm your PAT is saved
        + Gh::gh\_whoami()
        + Usethis::git\_siterep()
* Create repository on GitHub
  + Create description
  + Select create README.md
  + Click on Code in upper right hand corner
    - HTTPS
      * copy
* In RStudio
  + New Project
  + Select Version Control
  + Git
    - Paste https code copied from GitHub into Repository URL
    - Project directory name should automatically generate to repo name
    - Select where the project will be housed
    - Click create project
* Project is created!
* Collaborating on projects
  + In gitbash
  + (you should already be connected to GitHub)
  + Set working directory where you want repo to go (cd “directoryname”)
  + Git clone git@github.com:username/repo.git