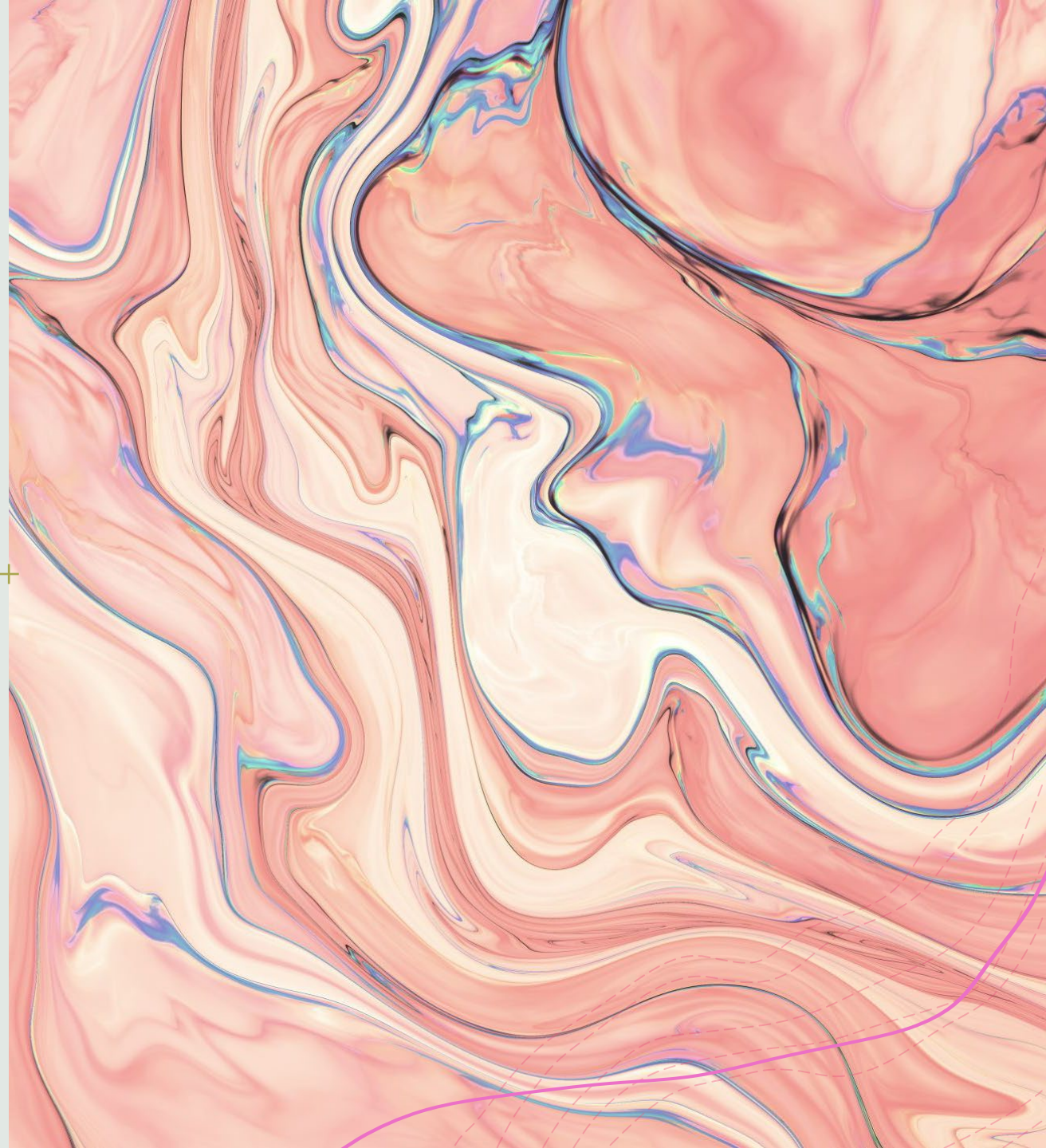




# Using RStudio and GitHub for Reproducible Science



# AGENDA

Importance of Reproducible Science

A light red arrow pointing downwards, indicating the flow from the first item to the second.

Project Organization

A light red arrow pointing downwards, indicating the flow from the second item to the third.

Rprojects and GitHub

A light red arrow pointing downwards, indicating the flow from the third item to the fourth.

Connecting to GitHub

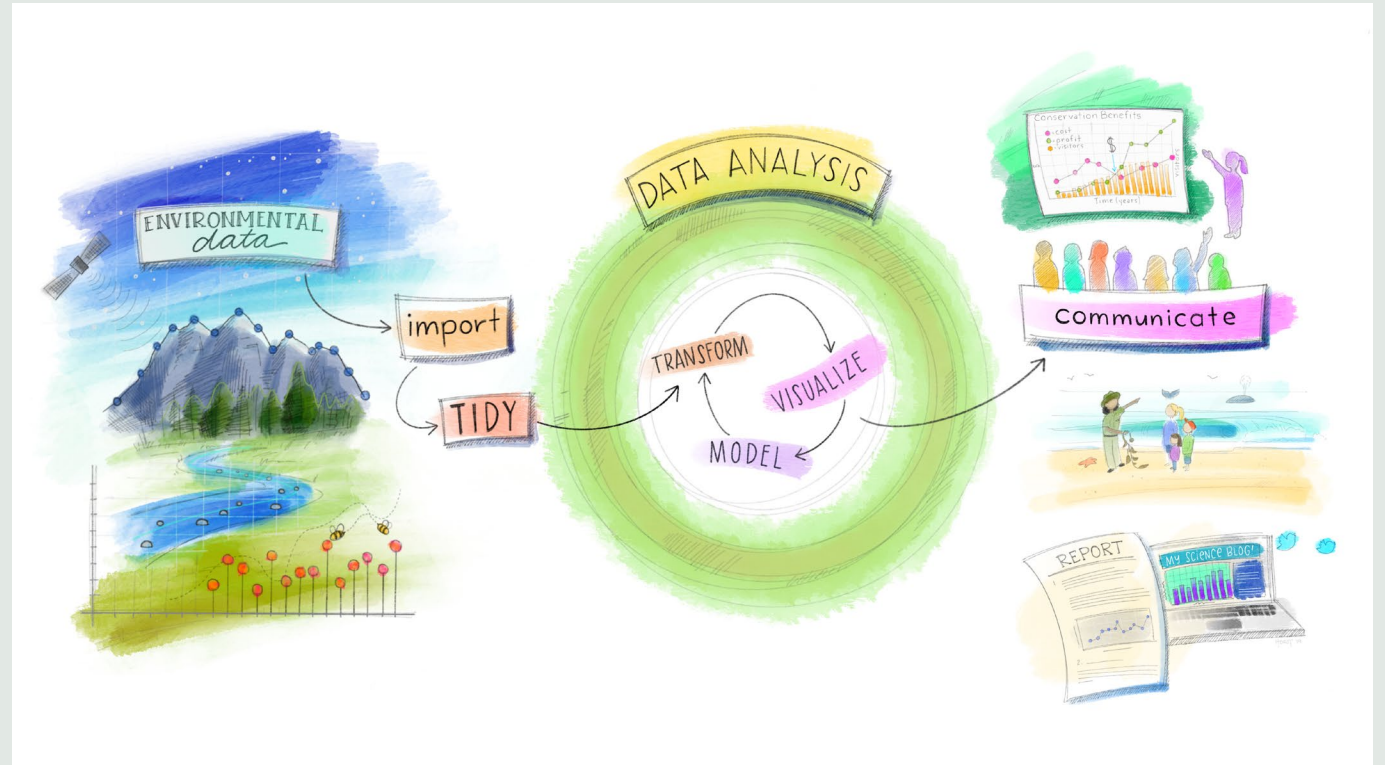


All the slides and additional  
resources available

[https://github.com/kpgund/github\\_workshop](https://github.com/kpgund/github_workshop)

# Importance of Reproducible Science

- ***“Reproducibility is a pillar of the scientific method” – Dr. Picardi***
- Being proficient in the use of programming tools and effectively apply them to store, process, manage, analyze, and visualize data have become **must-have skills** to take part in the scientific discourse.

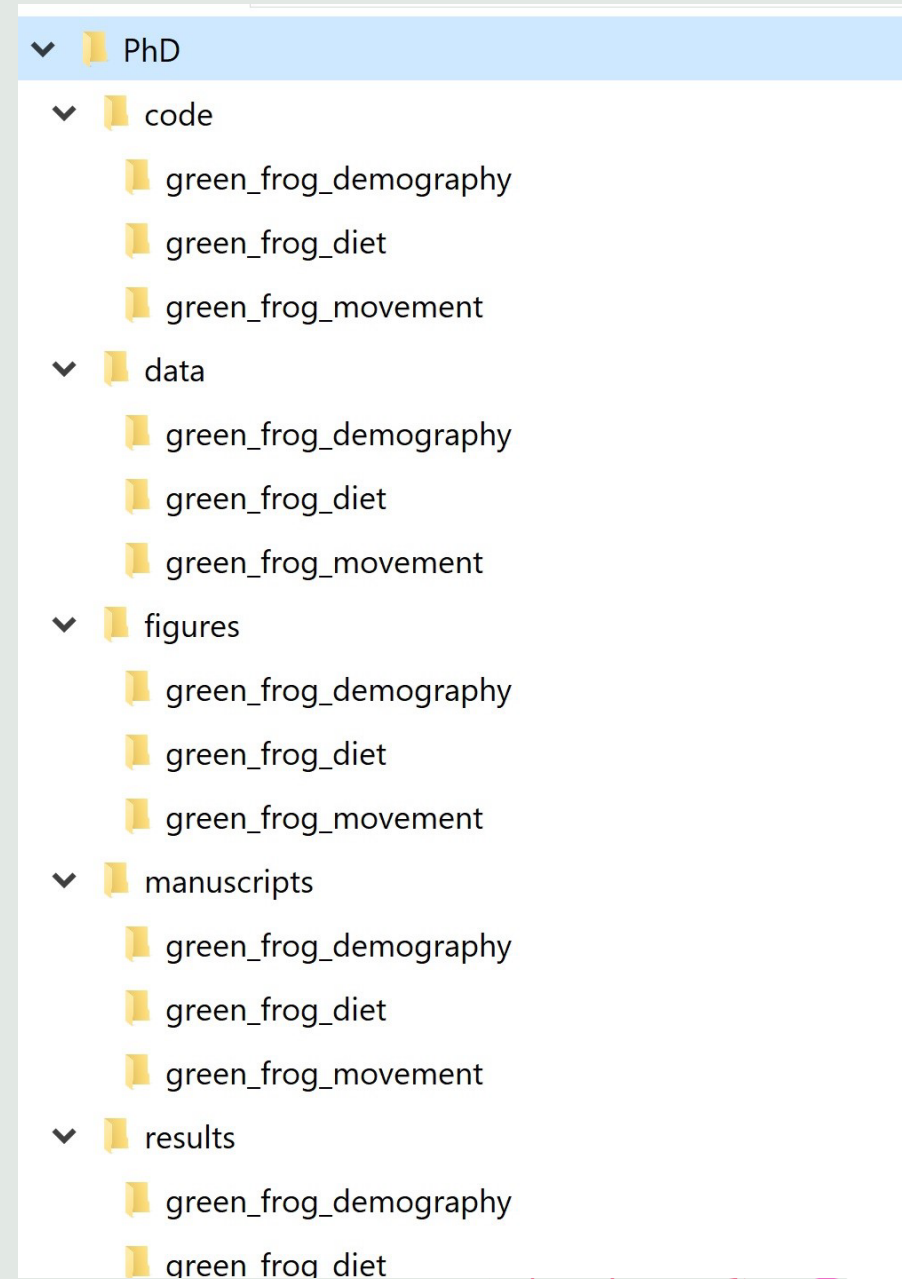


Artwork by Allison Horst



# Project Organization

- Everything meaningful in one folder structure
- Folder Structure
  - Project-based
  - Activity-based
- File Naming



# Golden rules

1. **Raw data should never be changed.** Save it into a “data” folder and treat it as immutable. You can even set it as read-only to make sure there is no room for accidents.
2. The processed, clean version of your data will go into a dedicated “processed\_data” folder.
3. Anything that can be generated from code goes into its own folder. This includes basically everything but the raw data and the code itself. You can have an “output” folder, or separate folders for output files and figures (e.g., “output” and “figures”)
4. If there are text documents, put them in their own folder (e.g., “docs”)
5. Code also has its own folder. If you write a lot of functions, it can be helpful to have a “funcs” folder to store those and a “src” (for ‘source’) folder to save processing/analysis code.
6. If processing/analysis scripts are meant to be used in a certain order, you can number them (more on this in a minute). Sometimes the pipeline is not linear but branched, so numbering may not always make sense. Function scripts should not be numbered.
7. Modularize your code: instead of having a giant script to run your entire analysis from data cleaning to final figures, break up your workflow into several short, single-purpose scripts with well-defined inputs and outputs.

# File Naming

- Good file names are computer-readable, human-readable, and work well with default ordering.

## **Ex: What we don't want:**

data.csv

data\_cleaned\_March-22-2012.csv

analysis code.R

Green Frogs Manuscript\_Final\_edits.docx

final.docx

## **Ex: What we do want:**

20221009\_GitHub-workshop\_powerpoint.ppt

01\_DataOrganization.R

02\_DataCleaning.R

03\_Analysis.R



## **Simona Picardi**

I am an ecologist with expertise in the analysis of animal movement, behavior, and space-use.

My research aims to quantify wildlife responses to environmental change and anthropogenic pressures with the goal of informing their management and conservation.

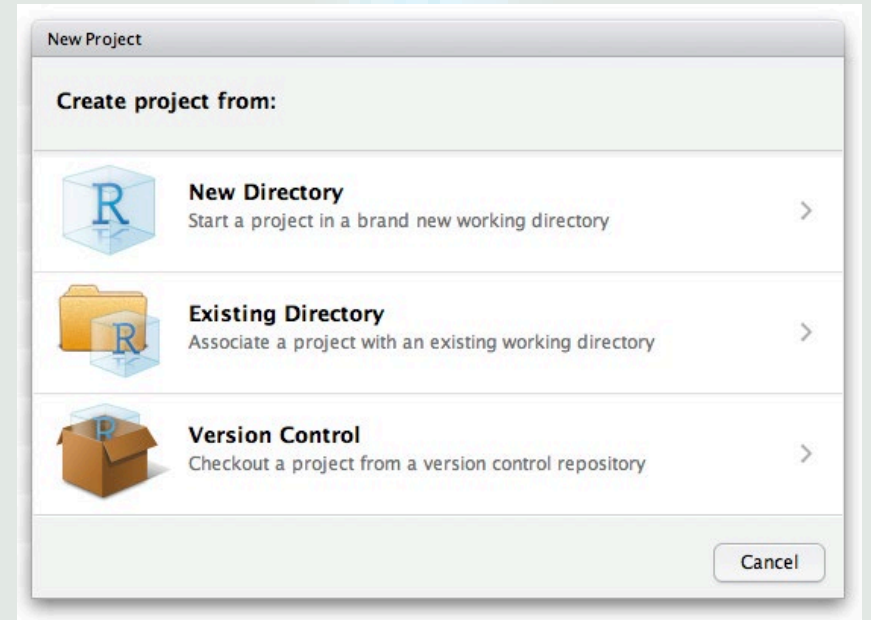
I am a passionate teacher and educator with several years of successful experience teaching data science and programming to ecologists at all levels.

- + For more info:
- + <https://ecorepsci.github.io/reproducible-science/>
- + <https://www.picardiecology.com/teaching>

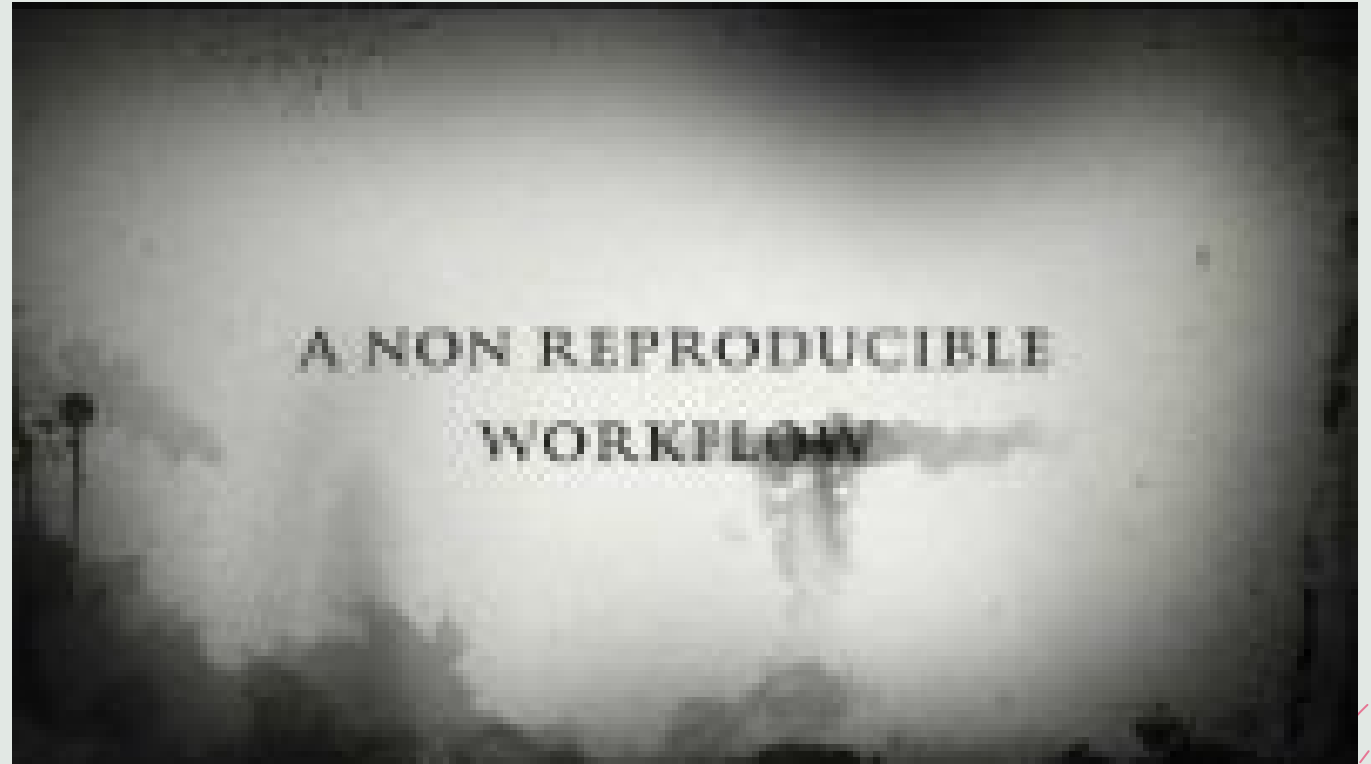


# Rprojects (.Rproj)

- + Working within the project directory
- + Self-contained
  - + Inputs, outputs, and code are all in one place
- + Relative paths



# Why use GitHub



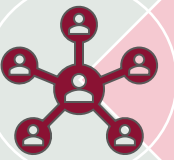
# Why use GitHub



Backup of your project



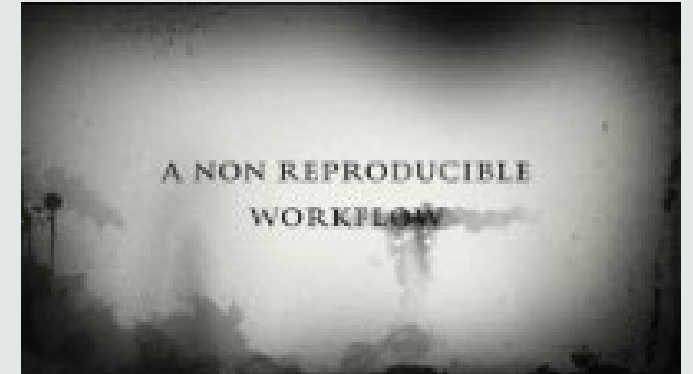
Keep track of changes



Network



Share research  
Collaborate with others



# Rstudio + GitHub

## Version Control with Git or SVN



Turn on at **Tools > Project Options > Git/SVN**

Stage files:

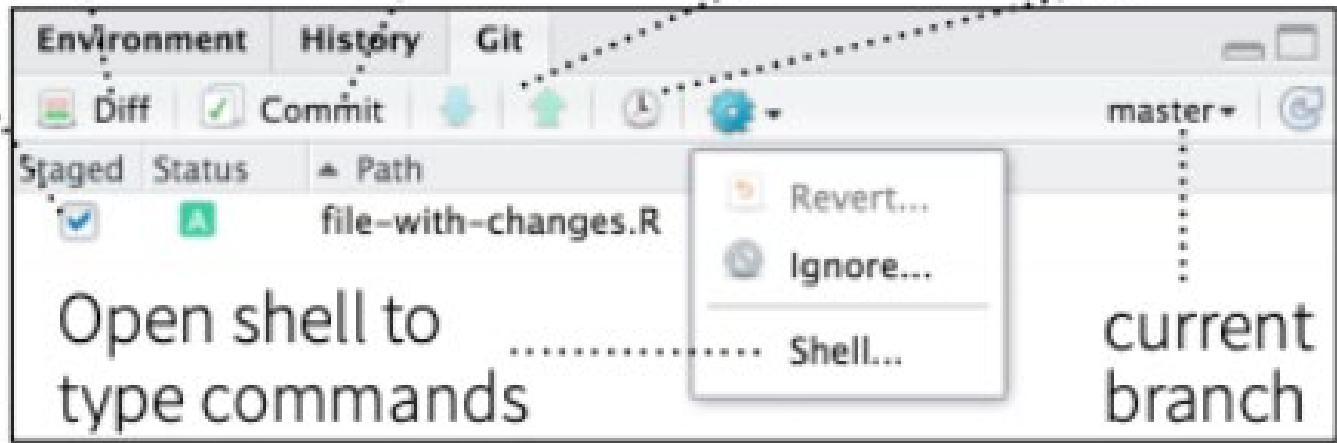
Show file diff

Commit staged files

Push/Pull to remote

View History

**A** Added  
**D** Deleted  
**M** Modified  
**R** Renamed  
**?** Untracked



# Rstudio + GitHub

- + Track what updates are happening
- + Find previous version

File Name	Line	Code	Commit
00_se			2
01_pa		@@ -23,7 +23,7 @@ deer.bcp <- read.csv("outputs/bcp/final_bcp_deer_wtaudiff_20220206.csv",header=T	days ago
02_br	23	dplyr::mutate(Species = "deer")	days ago
	24	elk.gcp <- read.csv("outputs/gcp/final_gcp_elk_wtaudiff_20220419.csv",header=T) %>%	days ago
05_ge	25	dplyr::mutate(Species = "elk")	months ago
	26	- elk.bcp <- read.csv("outputs/bcp/final_bcp_elk_wtaudiff_20220419.csv",header=T)%>%	days ago
05_lo	26	+ elk.bcp <- read.csv("outputs/bcp/final_bcp_elk_wtaudiff_20220826.csv",header=T)%>%	days ago
	27	dplyr::mutate(Species = "elk")	
06_ca	28	## add in 2023	last month
	29	elk.2023F <- read.csv(paste("outputs/bcp/2023F_bcp_elk_wtaudiff_",date_print,".csv",sep=""))%>%	days ago





CREATE  
SSH KEY



CONNECT  
SSH KEY  
TO GITHUB  
ACCOUNT



CREATE A  
REPOSITORY  
IN GITHUB



CREATE AN  
RPROJECT  
AND  
CONNECT IT  
TO THE  
GITHUB  
REPOSITORY



COMMIT/  
PUSH/  
PULL

# Let's get started!