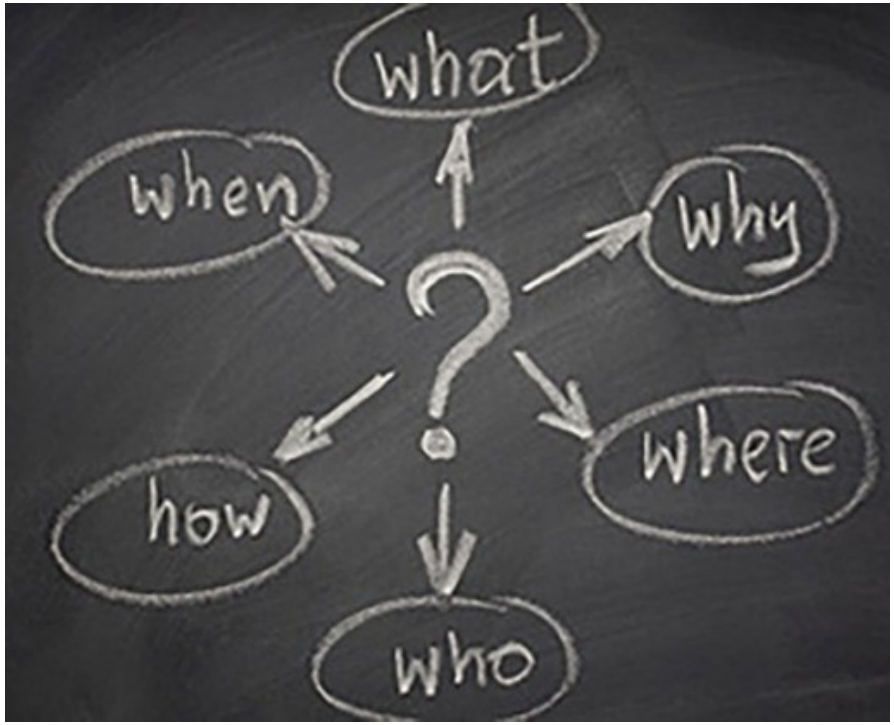
# Reproducible research

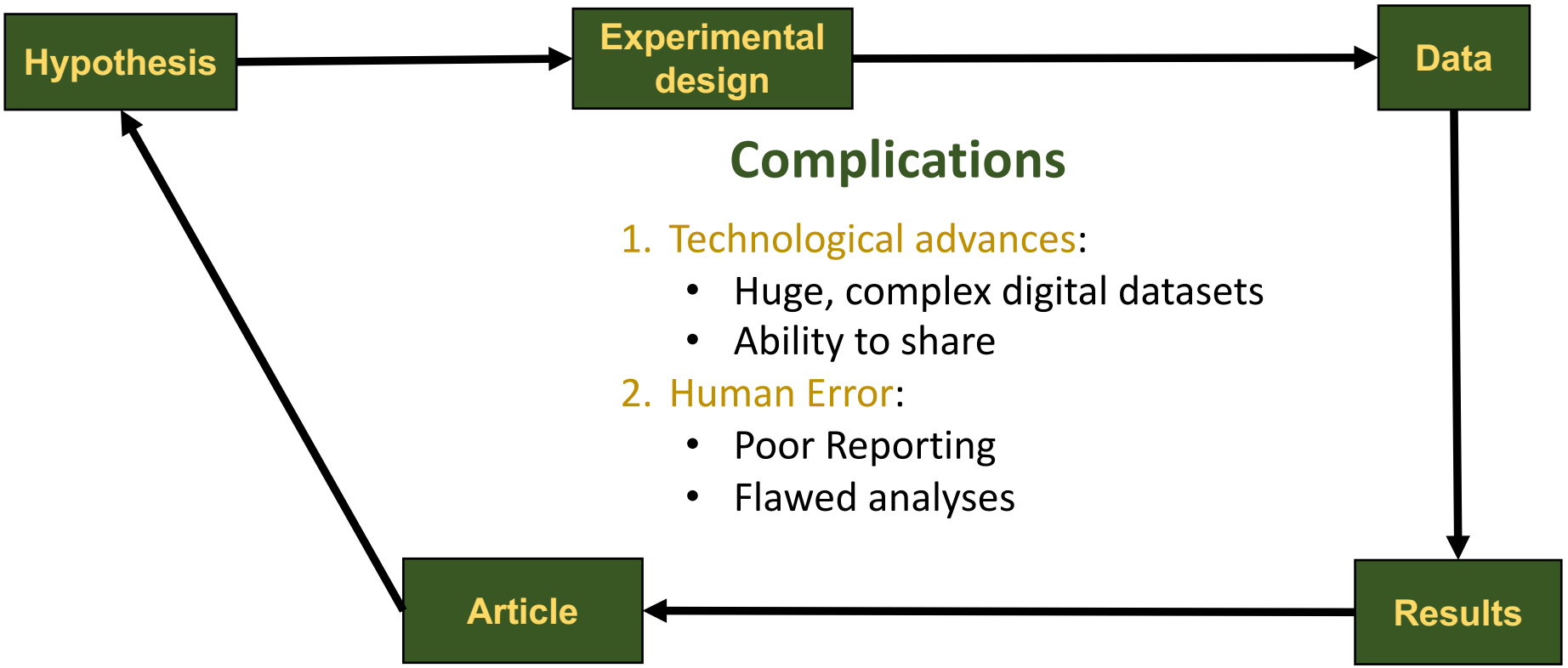
Tobin Magle, PhD
tobin.magle@colostate.edu
Data Management Specialist
Morgan Library
Colorado State University

# Outline

- What is reproducible research?
- Why would I do that?
- How? (in R Studio)
  - Automation
  - Git
  - R Markdown



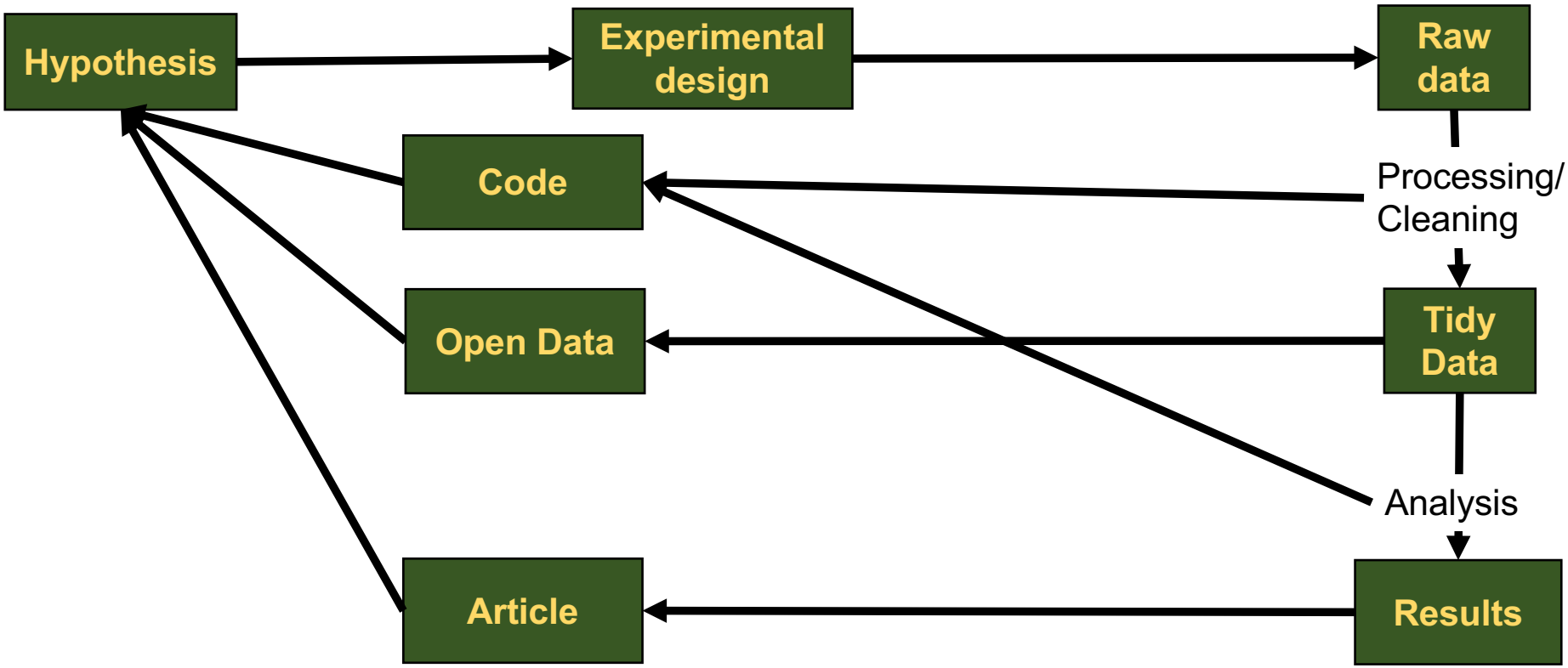
# The research cycle




## Complications

1. Technological advances:
   - Huge, complex digital datasets
   - Ability to share
2. Human Error:
   - Poor Reporting
   - Flawed analyses

# The research cycle



# Reproducible research

is the practice of distributing all data, software source code, and tools required to reproduce the results discussed in a research publication.

https://www.ctspedia.org/do/view/CTSpedia/ReproducibleResearchStandards

# Reproducible research

=

Data (with metadata)

+

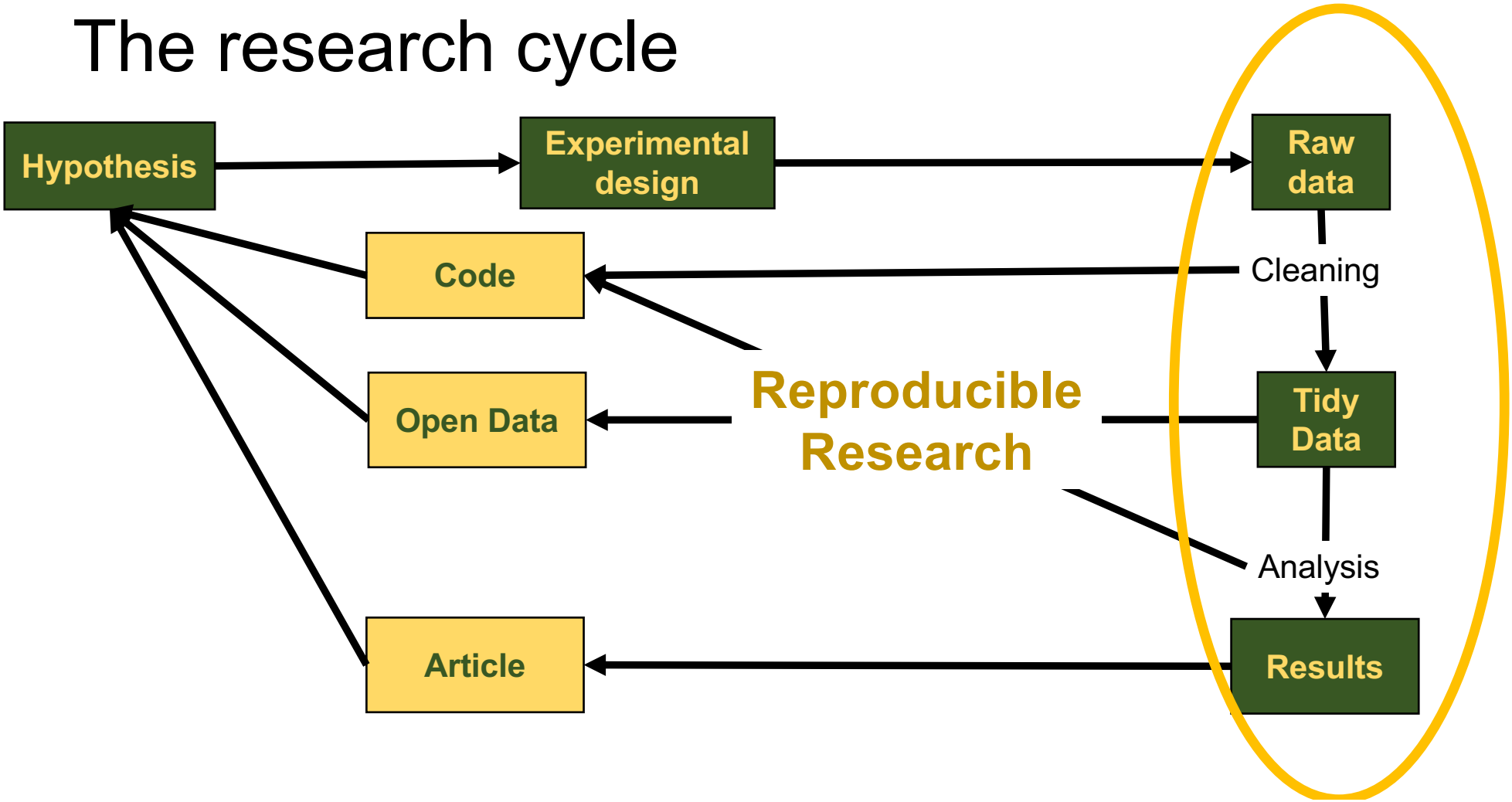
Code/Software

**Reproducible research**

**=**

Transparency

# The research cycle



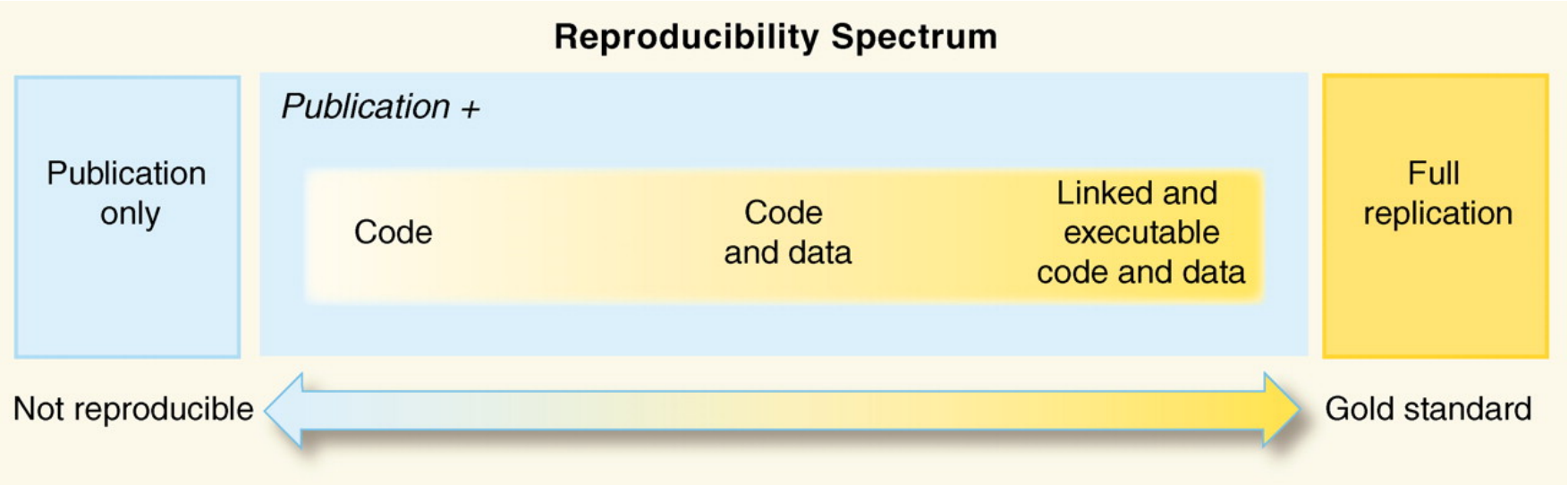
# Replication vs. Reproducibility

- **Replication**: Same conclusion new study (gold standard)
- "Again, and Again, and Again …" **BR Jasny et. al.** Science, 2011. 334(6060) pp. 1225 DOI: 10.1126/science.334.6060.1225

- **Replication isn't always feasible**: too big, too costly, too time consuming, one time event, rare samples

- **Reproducibility**: Same results from same data and code (minimum standard for validity)
- "Reproducible Research in Computational Science". **RD Peng** Science, 2011. 334 (6060) pp. 1226-1227 DOI: 10.1126/science.1213847

# Reproducibility spectrum




"Reproducible Research in Computational Science". **RD Peng** Science, 2011. 334 (6060) pp. 1226-1227 DOI: 10.1126/science.1213847

# Using markdown and version control in R Studio:

# Reproducible research checklist

- **Think about the entire pipeline**: are all the pieces reproducible?
- **Is your cleaning/analysis process automated?**– guarantees reproducibility
  - Are you doing things “by hand”? editing tables/figures; splitting/reformatting data
  - Does your software support log files or scripts?
  - If no, do you have a detailed description of your process?
- **Are you using version control?**
- **Are you keeping track of your software?**
  - Computer architecture;
  - OS/Software/tool/add ons (libraries/packages)/external databases
  - version numbers for everything (when available)
- **Are you saving the right files?**: if it’s not reproducible, it’s not worth saving
  - Save the data and the code
  - Data + Code = Output
- **Are your reports human and machine readable**?

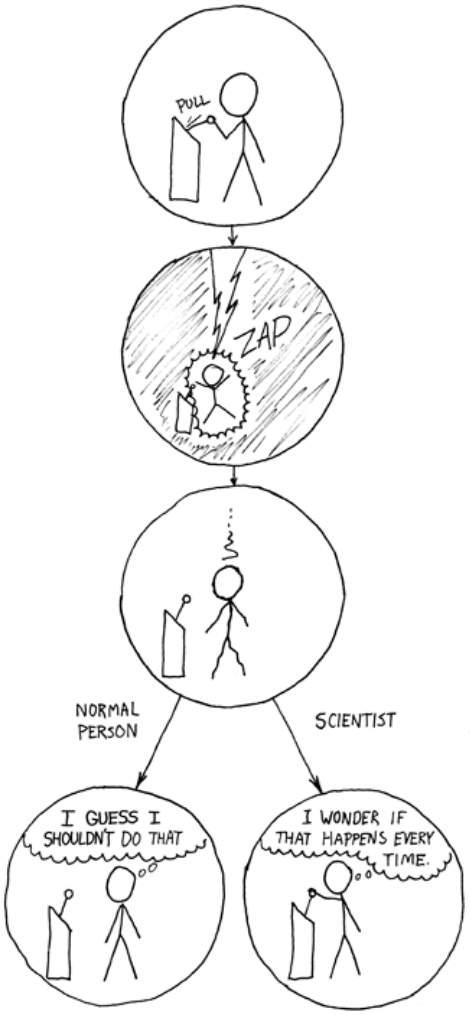
Adapted from: https://github.com/DataScienceSpecialization/courses/blob/master/05_ReproducibleResearch/Checklist/Reproducible%20Research%20Checklist.pdf

# Write instructions

- **Optimal**: the instructions should be an automated script file (ie, “code”)
- **Minimum**: Written instructions that allow for the complete reproduction of your analysis

# Research is repetitive

- Replication
- Same assay, different samples
- Longitudinal experiments




http://xkcd.com/242/

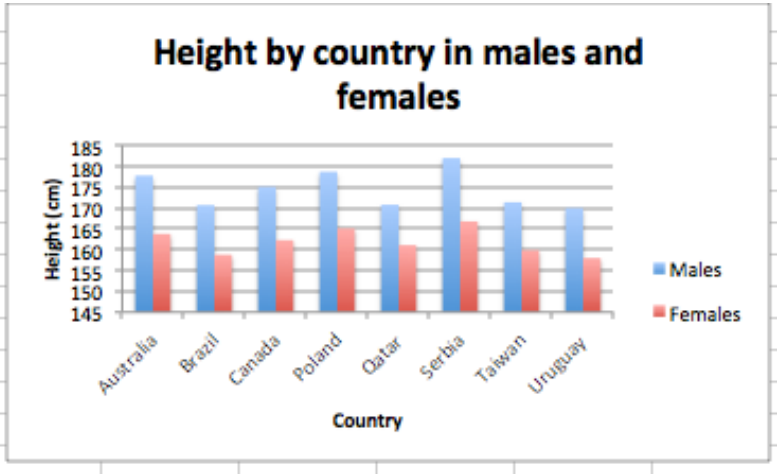
# Doing things by hand is…

- Slow
- Hard to document
- Hard to repeat

# Example: Making graphs

- Describe how to make a bar graph in excel

| | Height in cm | |
|---|---|---|
| | Males | Females |
| Australia | 177.8 | 163.8 |
| Brazil | 170.7 | 158.8 |
| Canada | 175.1 | 162.3 |
| Poland | 178.7 | 165.1 |
| Qatar | 170.8 | 161.1 |
| Serbia | 182 | 166.8 |
| Taiwan | 171.4 | 159.9 |
| Uruguay | 170 | 158 |



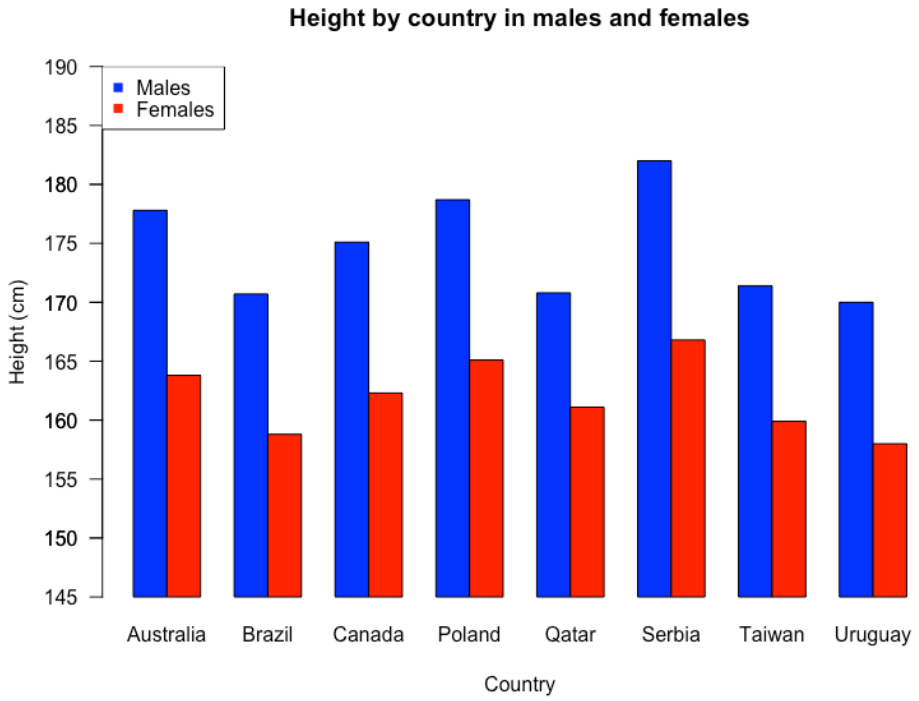
# Automation

- Fast
- Easy to document
- Easy to repeat
- Write scripts or save log files

# Example: Making graphs

- Use a script

```
#download the file
download.file(url = "http://libguides.colostate.edu/ld.php?content_id=27156359",
              destfile = "ex1.csv",
              method="libcurl")

#Load the data from the file into an R variable
height<-read.csv("ex1.csv", row.names="Country")

#Now let's plot the data:

counts<-t(as.matrix(height)) #converts the variable height to a format that
#can be plotted
counts<-counts-145            #transforms the data so it looks like the excel plot
barplot(counts,               #the height of the bar
        beside = TRUE,        #put cols next to eachother
        main="Height by country in males and females",  #plot title
        xlab="Country",         #X axis label
        ylab="Height (cm)",     #Y axis label
        col=c("blue", "red"),   #bar colors
        offset=145,             #shifts the axis to make it look like excel
        ylim=c(145,190),        #y axis limits
        las=1)                  #horizontal text
axis(side=2,                    #marks on the left of axis
     at=c(145,150,155,160,165,170,175,180,185),  #where you want ticks
     las=1) #horizontal text

legend(x=0, y=190, #coordinates of where you want the legend to go
       legend=c("Males", "Females"), #legend text label
       col=c("blue", "red"),         #colors
       pch=15)                       #shape of legend
```



# Details to record for processing/analysis

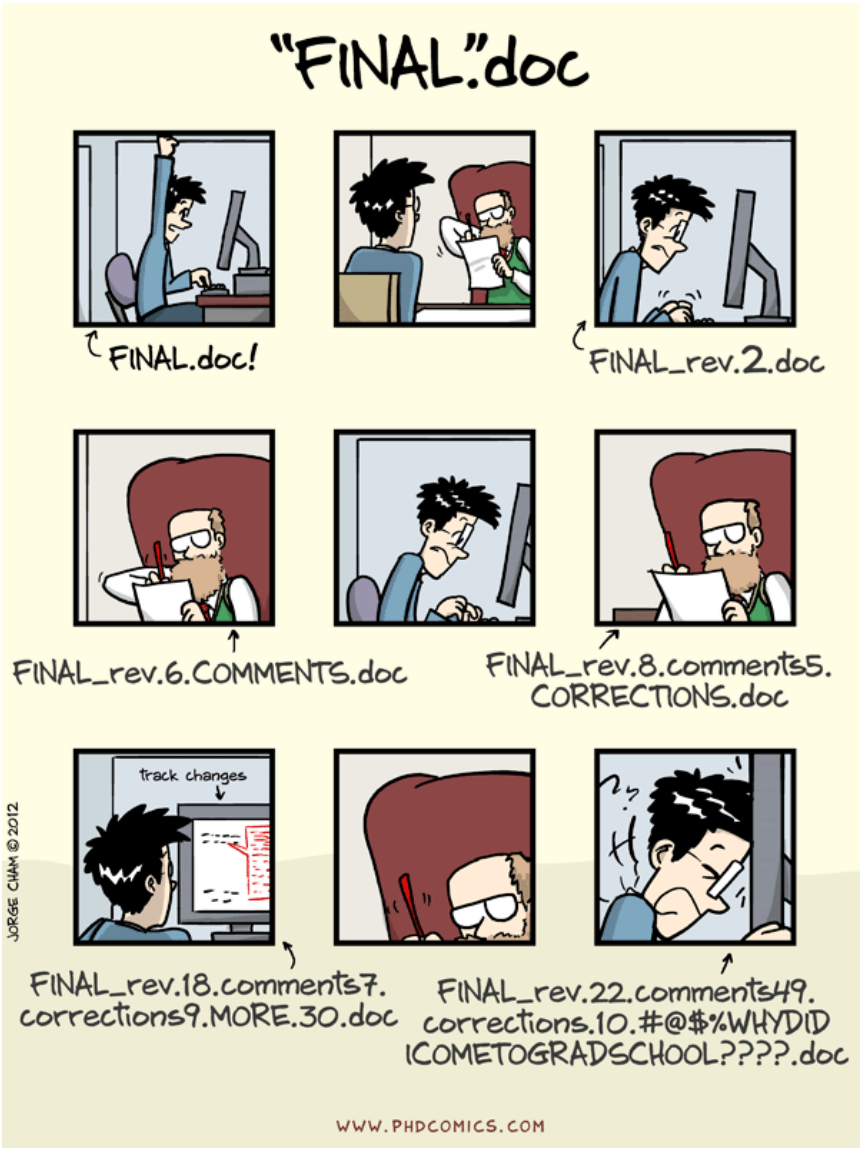
- What **software** was used? (R Studio, script)
- Does it support **log files/scripts**? (yes!)
- What **version # and settings** were used? (R version 3.3.2)
- **What else** does the software need to run?
  - Computer architecture
  - OS/Software/tool/add ons (libraries/packages)
  - External databases

# In R: the sessionInfo() command

```
> sessionInfo()
R version 3.3.2 (2016-10-31)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: macOS Sierra 10.12.1

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
[1] tools_3.3.2
```

# Intuitive version control

http://phdcomics.com/comics.php?f=1531



# Version Control

- A system that records changes to a file or set of files over time so that you can recall specific **versions** later
  - https://git-scm.com/doc
  - Saves EVERYTHING
  - Records who made what change
  - Identifies conflicting changes
- Not just for code

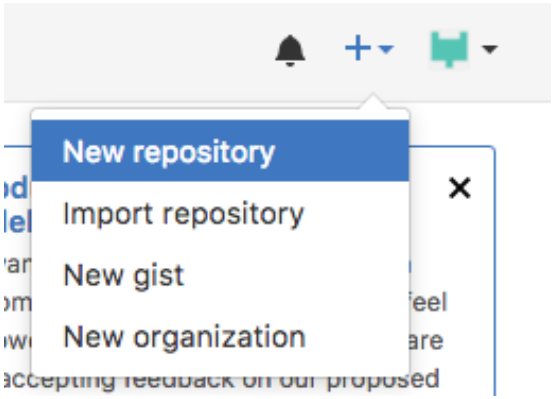
GitHub

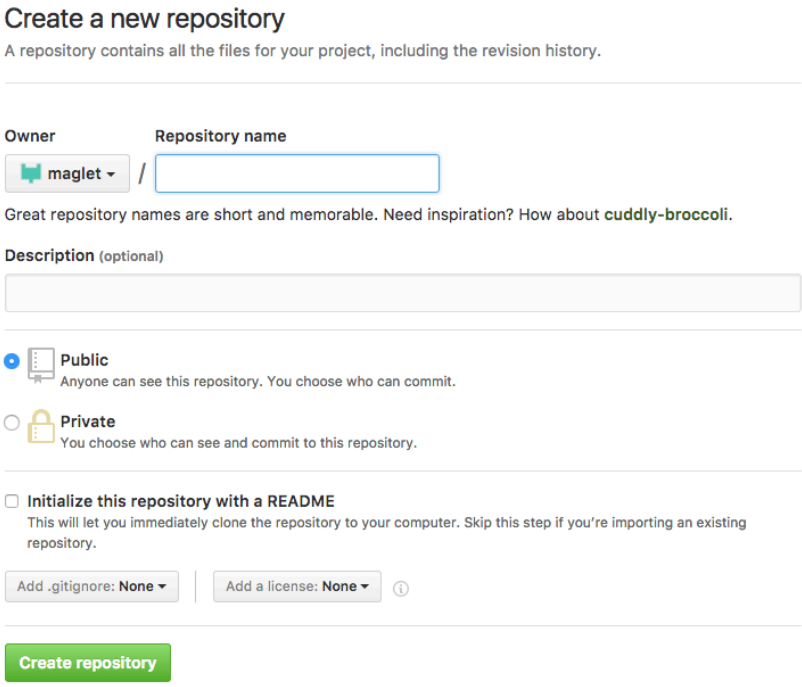
git

RStudio

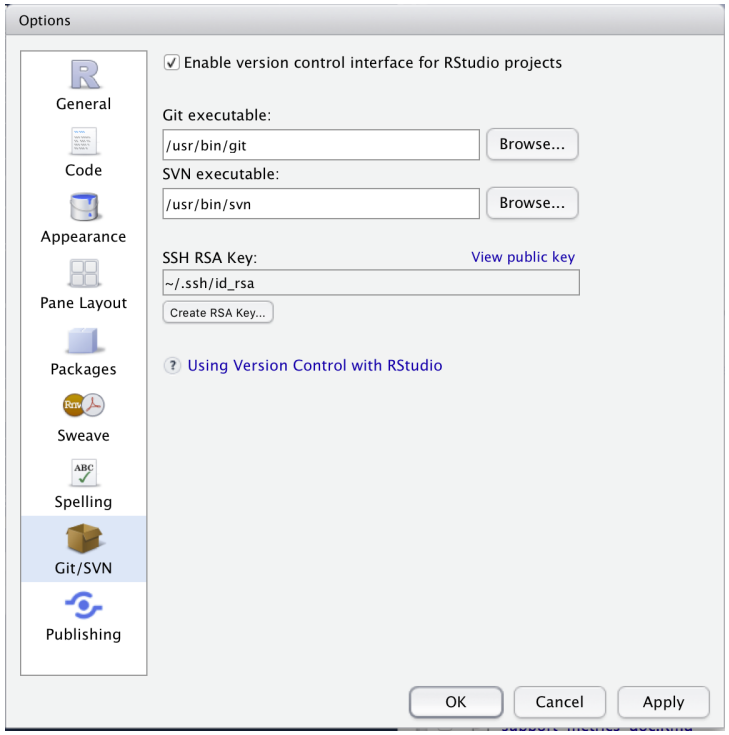
# Make an empty GitHub

- Github.com



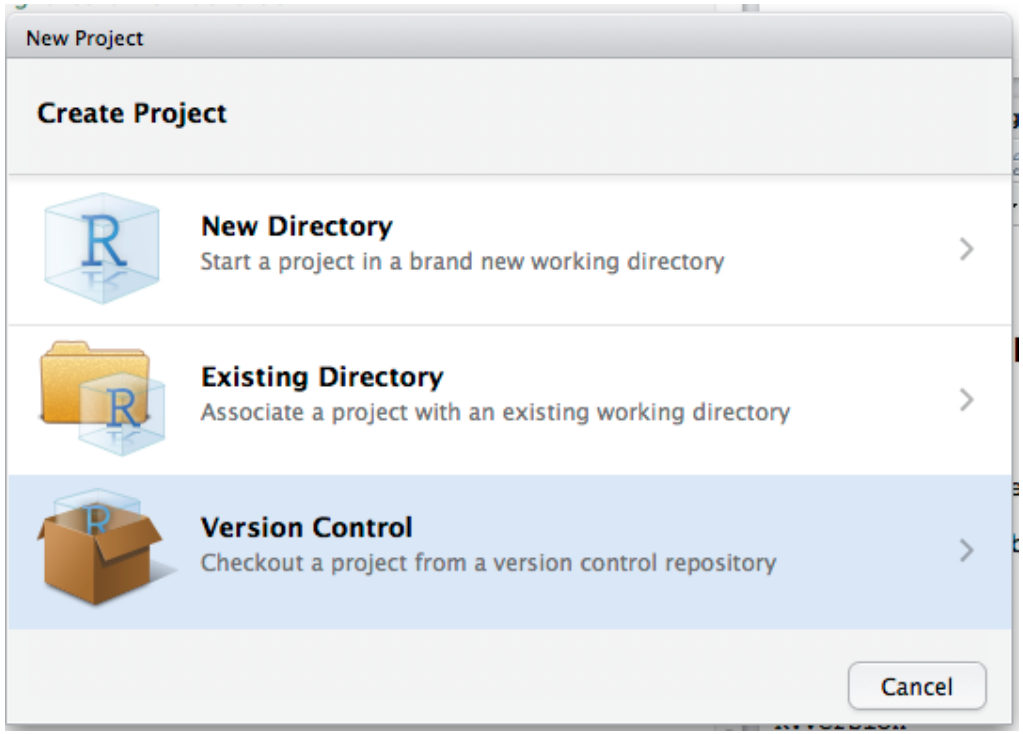
# Tell R Studio where git is




*if you don't' know where git is installed, try

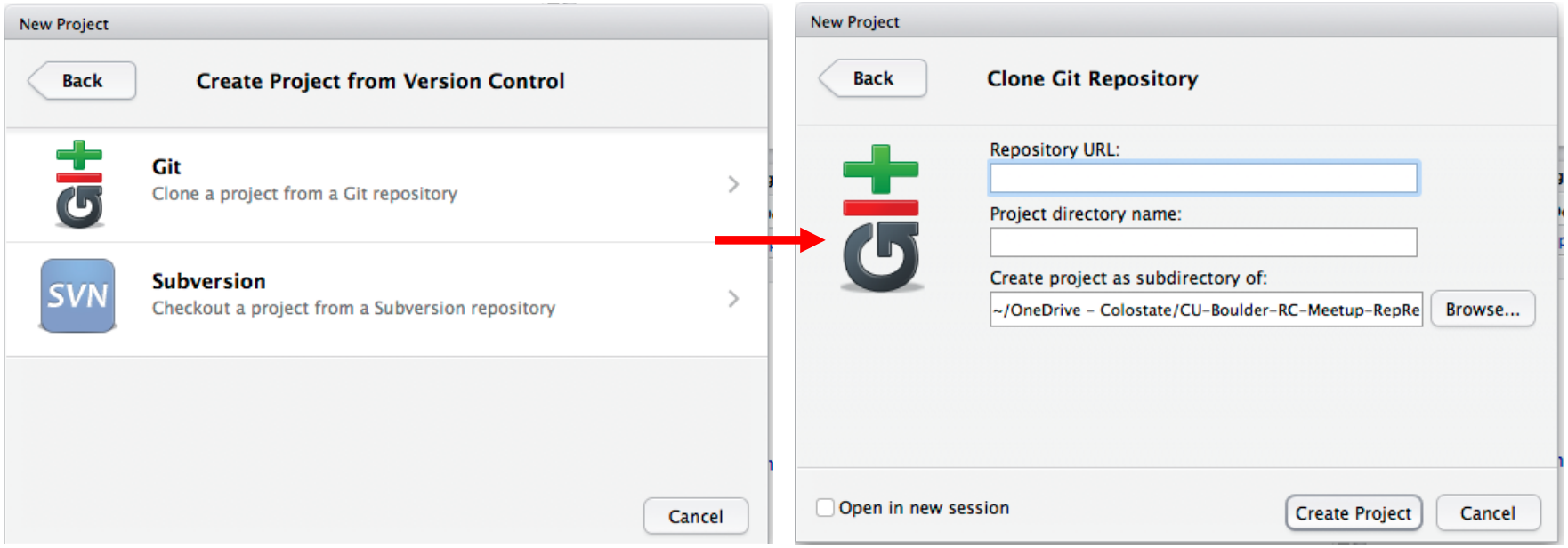
- Mac terminal: which git
- PC: where git.exe

# Make a new version control project




*auto-recognizes version control  if you select an existing directory

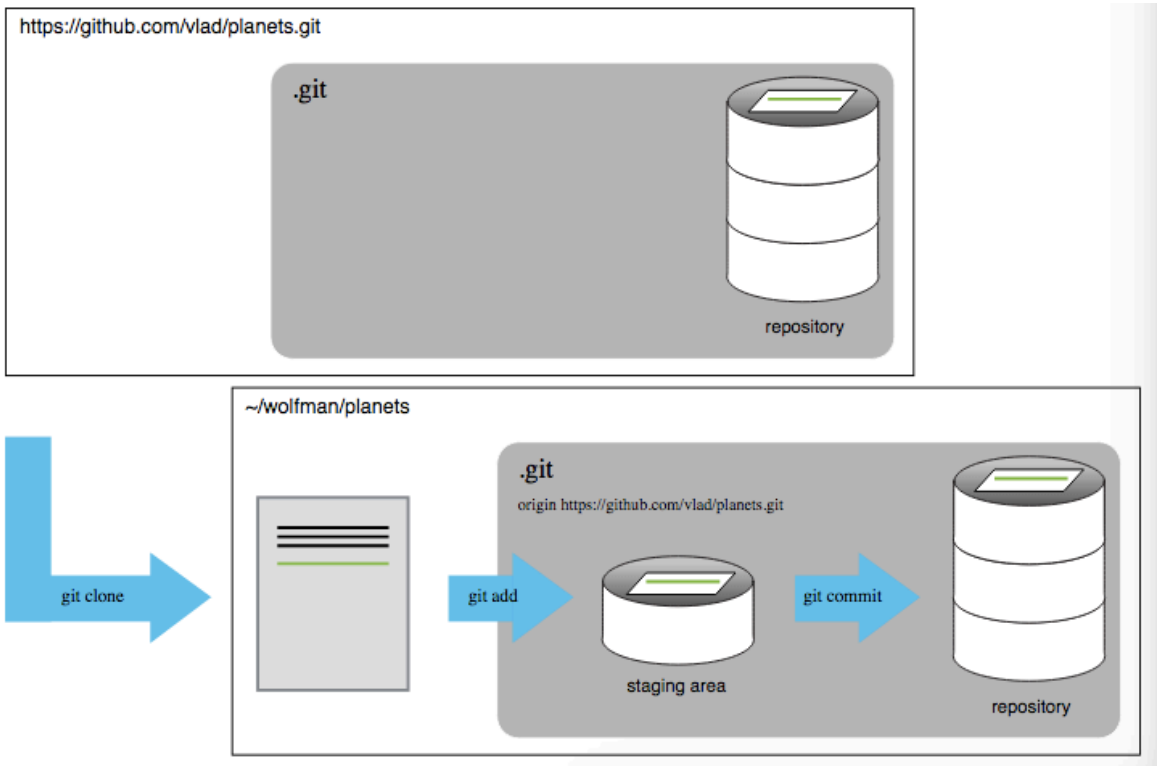
# Select git, select options



# Clone

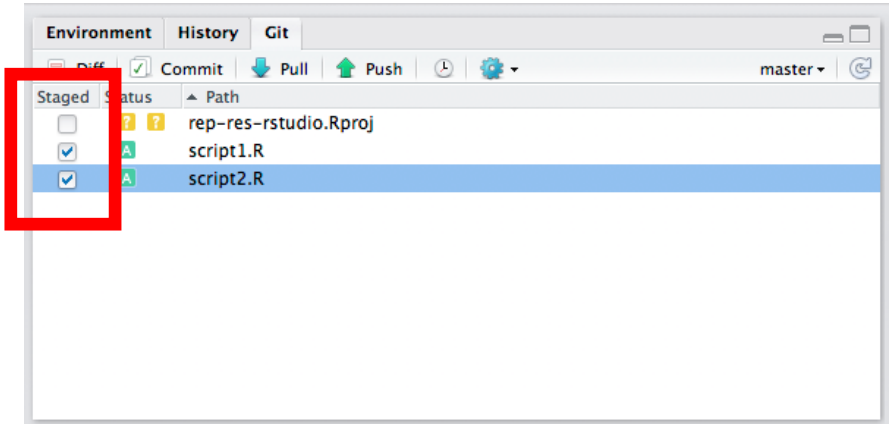
**Remote**

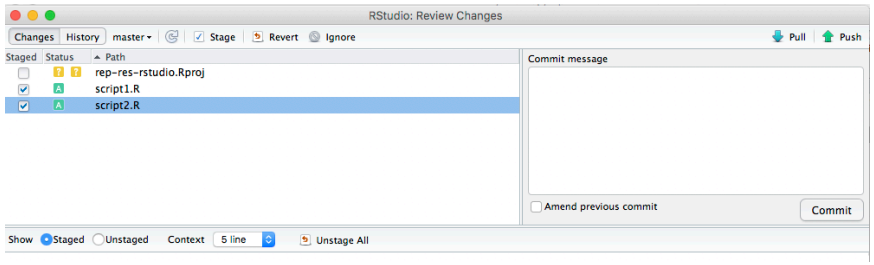
**Local**



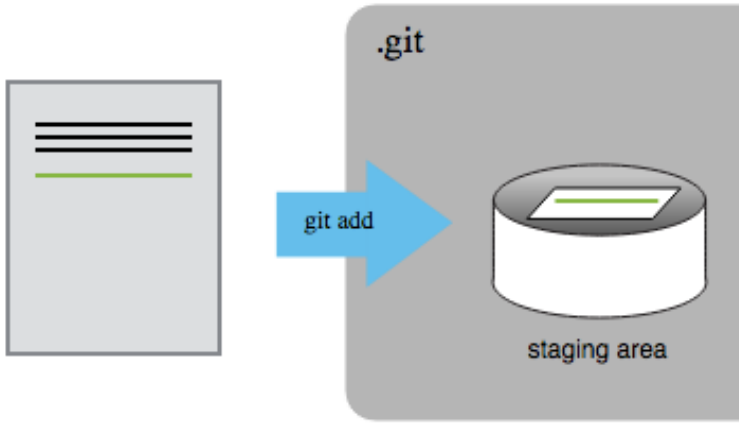
# Add files

- Check boxes to add files to the “staging area”
  - Gets them ready to be added (committed) to the repository



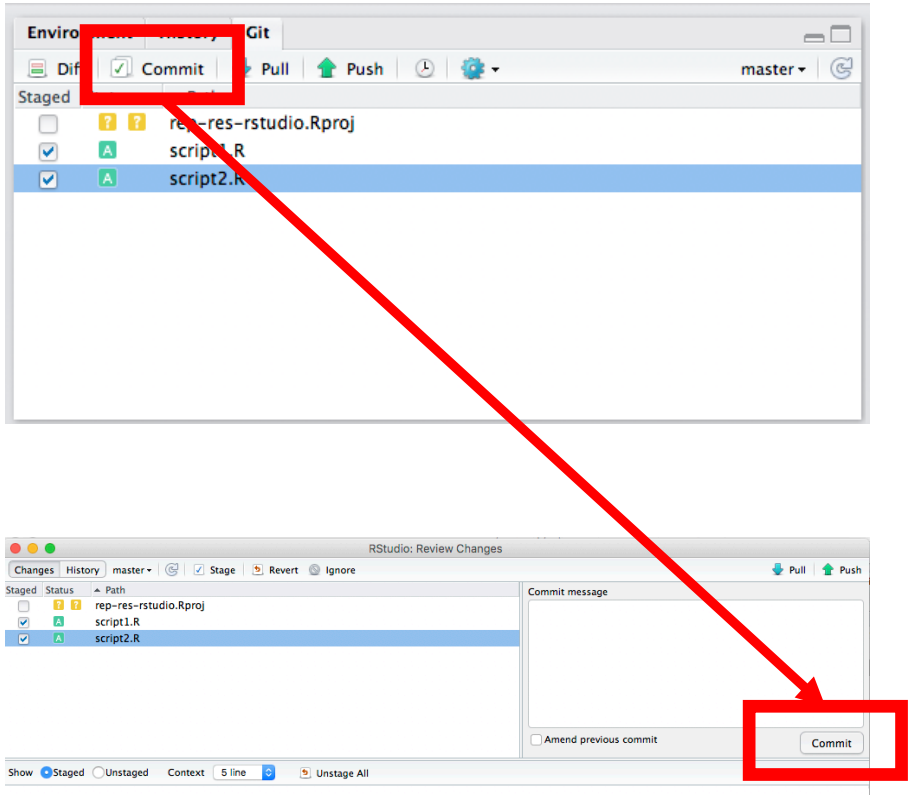



# Staging area vs. repository

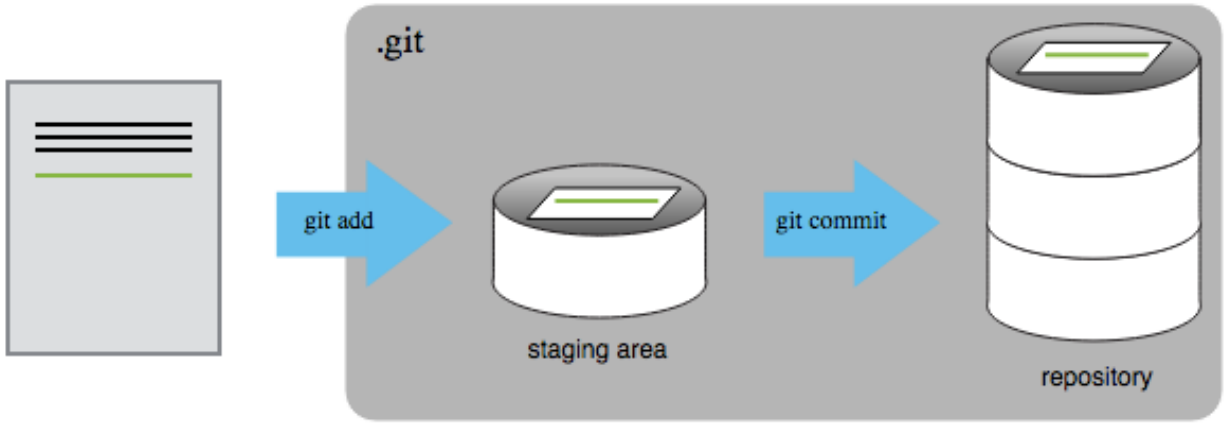


# Commit files

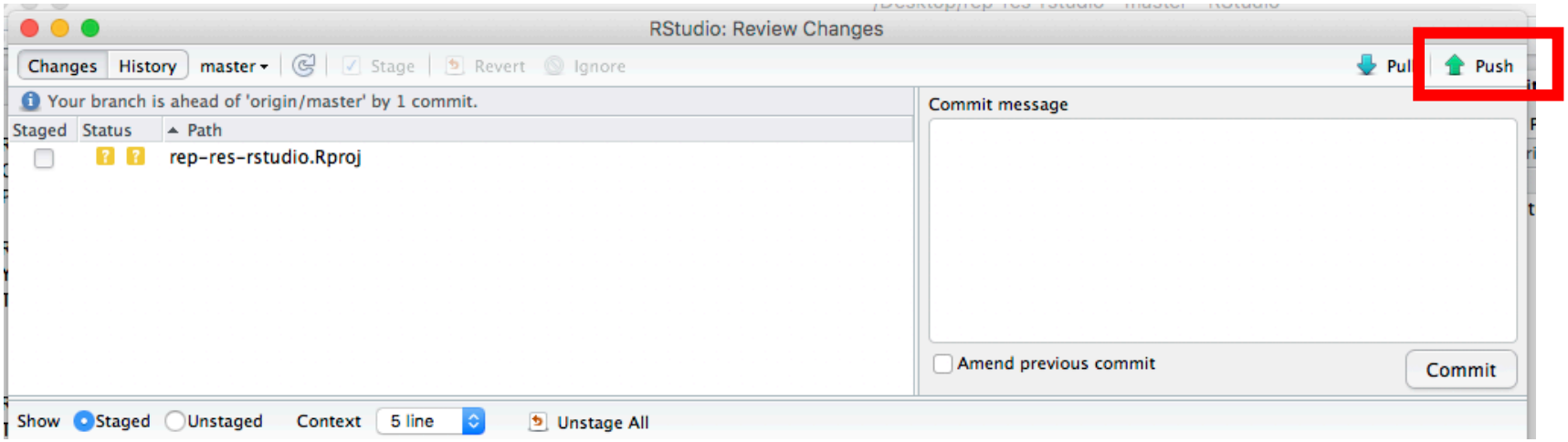
- Click “commit”
- Write a commit message
- Click “commit”
  - Adds the changes to the repository



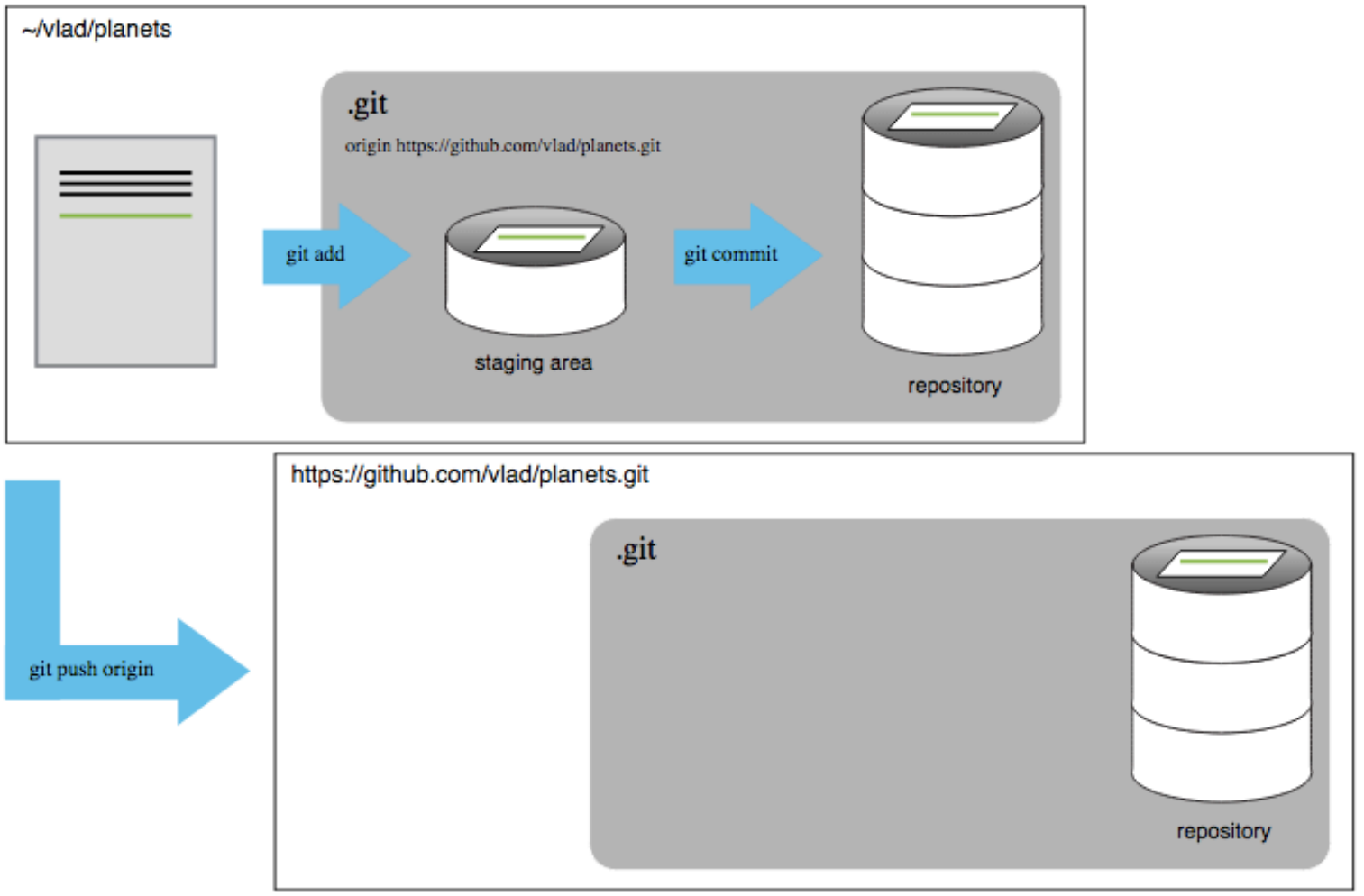
# Staging area vs. repository



# Push to remote

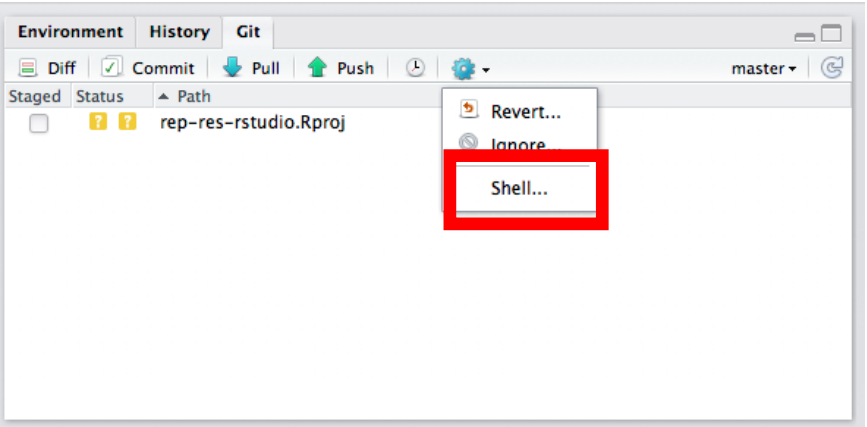


# Push


~/vlad/planets
.git
origin https://github.com/vlad/planets.git
git add
staging area
git commit
repository
git push origin
https://github.com/vlad/planets.git
.git
repository

# Limitation:

- Can’t play with history other than a simple revert
- Can access the command line



Literate programming

=

human readable (text)

+

machine readable (code)

R markdown

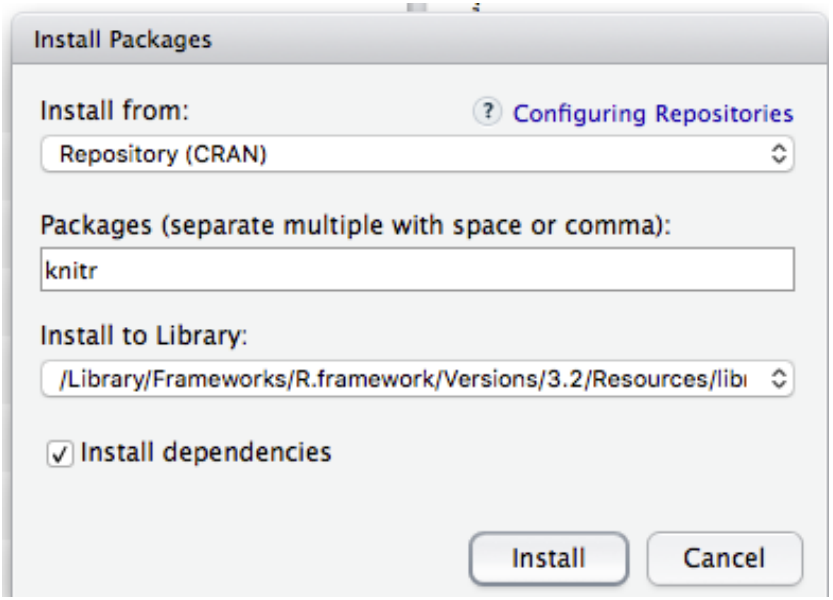
# R Markdown

- Open
- Write
- Embed
- Render

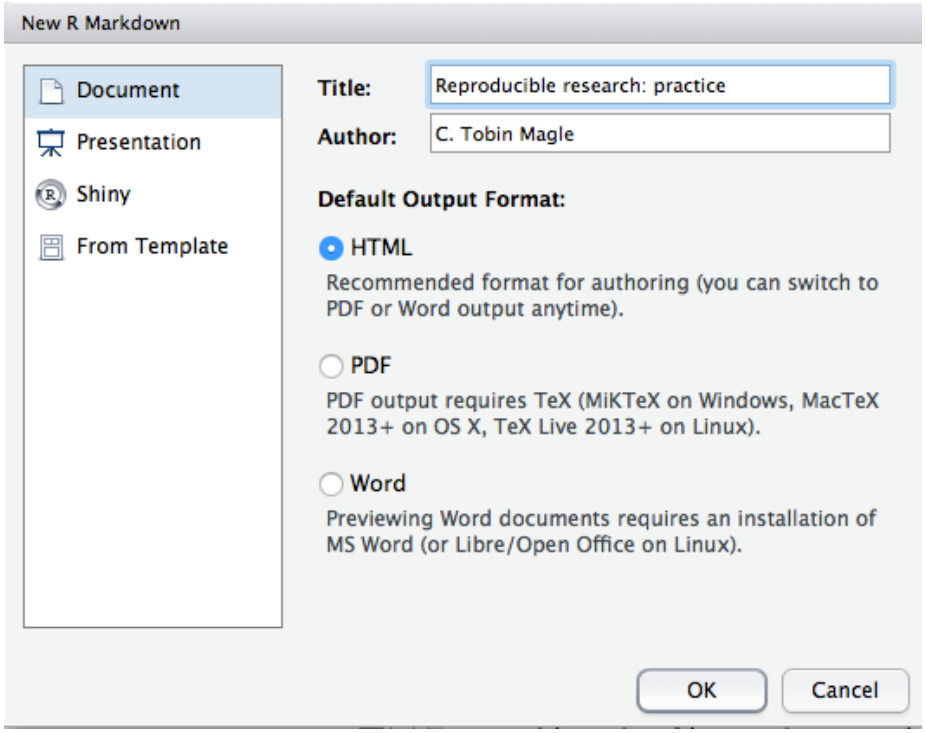

https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf

# Install knitr and markdown packages

- Tools > install packages
- Enter the package name (will autocomplete)
  - Knitr
  - Markdown
  - TeX (if you want to knit to PDF)
- OR install.packages("knitr")



# Open/Create a markdown document



# Write: useful syntax

- Plain text
- *italics* -> *italics*
- **bold** -> **bold**
- #Header -> Header (more # decreases size)
- Can also draw:
  - Insert pictures
  - Ordered and unordered list
  - Tables

# Embed code

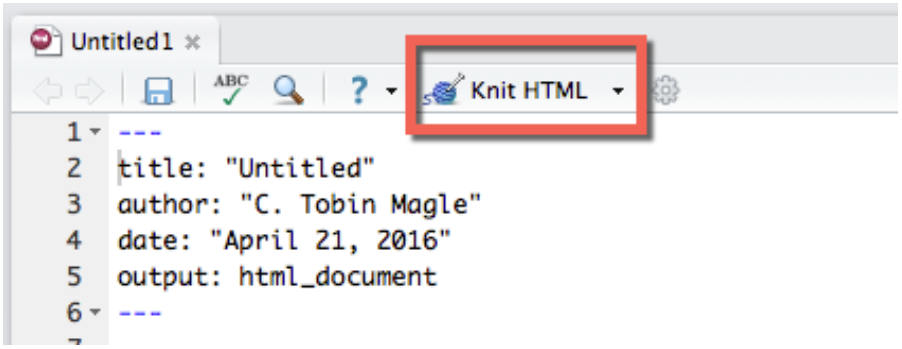
- **Inline –** Use variables in the human readable text
  - `r 2 + 2`
- **Code chunks** - Include working code that generates output
  - ```{r}
  - #Code goes here
  - ```
- **Display Options** –

| option | default | effect |
|---|---|---|
| eval | TRUE | Whether to evaluate the code and include its results |
| echo | TRUE | Whether to display code along with its results |
| warning | TRUE | Whether to display warnings |
| error | FALSE | Whether to display errors |
| message | TRUE | Whether to display messages |
| tidy | FALSE | Whether to reformat code in a tidy way when displaying it |
| results | "markup" | "markup", "asis", "hold", or "hide" |
| cache | FALSE | Whether to cache results for future renders |
| comment | "##" | Comment character to preface results with |
| fig.width | 7 | Width in inches for plots created in chunk |
| fig.height | 7 | Height in inches for plots created in chunk |

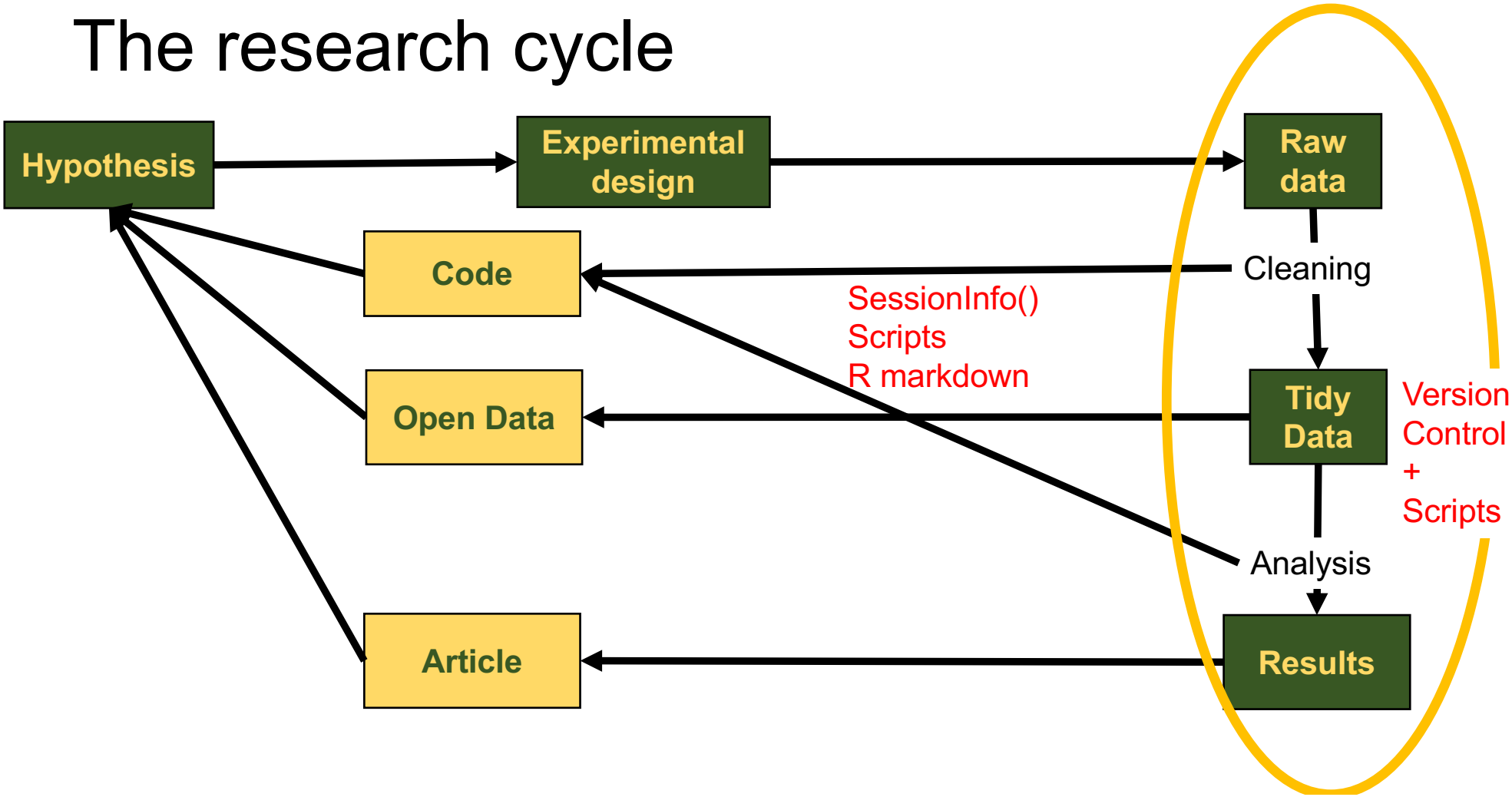
For more details visit yihui.name/knitr/

# Render

- Won't render unless the code runs with **no errors**
  - You know it should be reproducible
- Render using the **knit** function
- Output Formats
  - Knit HTML
  - Knit PDF – requires TeX
  - Knit Word



# The research cycle



# Reproducible research checklist

- **Think about the entire pipeline**: are all the pieces reproducible?
- **Is your cleaning/analysis process automated?**– guarantees reproducibility
  - Are you doing things “by hand”? editing tables/figures; splitting/reformatting data
  - Does your software support log files or scripts?
  - If no, do you have a detailed description of your process?
- **Are you using version control?**
- **Are you keeping track of your software?**
  - Computer architecture;
  - OS/Software/tool/add ons (libraries/packages)/external databases
  - version numbers for everything (when available)
- **Are you saving the right files?**: if it’s not reproducible, it’s not worth saving
  - Save the data and the code
  - Data + Code = Output
- **Are your reports human and machine readable**?

Adapted from: https://github.com/DataScienceSpecialization/courses/blob/master/05_ReproducibleResearch/Checklist/Reproducible%20Research%20Checklist.pdf