

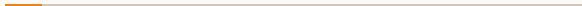
# BEST PRACTICES FOR THE POLITICAL SCIENTIST

---

J. Alexander Branham

Fall 2016

# INTRO



# WELCOME TO GRAD SCHOOL

---

- We're going to talk about things that aren't taught or talked about very often

# WELCOME TO GRAD SCHOOL

- We're going to talk about things that aren't taught or talked about very often
- How are you going to organize and manage your work?

# WELCOME TO GRAD SCHOOL

- We're going to talk about things that aren't taught or talked about very often
- How are you going to organize and manage your work?
- Tools for keeping track of what you do, how you do it, designed to *minimize error*

# WELCOME TO GRAD SCHOOL

- We're going to talk about things that aren't taught or talked about very often
- How are you going to organize and manage your work?
- Tools for keeping track of what you do, how you do it, designed to *minimize error*
- I'll give general advice that should apply for most, and then the tools I use for the specific problem

- Academic writing is messy

- Academic writing is messy
  - Keep track of the paper, figures, tables, references, cross-references, analysis, appendix, etc



- You want a system that keeps a record of your actions as you:

- You want a system that keeps a record of your actions as you:
  - Edit text

- You want a system that keeps a record of your actions as you:
  - Edit text
  - Analyze data

- You want a system that keeps a record of your actions as you:
  - Edit text
  - Analyze data
  - Present results

- You want a system that keeps a record of your actions as you:
  - Edit text
  - Analyze data
  - Present results
  - Do the above *reproducibly*

- You want a system that keeps a record of your actions as you:
  - Edit text
  - Analyze data
  - Present results
  - Do the above *reproducibly*
- We'll focus on free, open-source tools that enable this behavior and work on Linux, Mac, and Windows

RECORD



- You can use **version control** to keep track of changes to a file/folder/project



- You can use **version control** to keep track of changes to a file/folder/project
- The best version control systems let you easily see what has changed, when it changed, and who changed it

- You can use **version control** to keep track of changes to a file/folder/project
- The best version control systems let you easily see what has changed, when it changed, and who changed it
  - Word's “track changes” feature on steroids

- You can use **version control** to keep track of changes to a file/folder/project
- The best version control systems let you easily see what has changed, when it changed, and who changed it
  - Word's "track changes" feature on steroids
- Gets rid of `paper.doc`, `paper2.doc`, `paper-commented.doc`, `paper-FINAL.doc`, `paper-FINALFINAL.doc`

- You can use **version control** to keep track of changes to a file/folder/project
- The best version control systems let you easily see what has changed, when it changed, and who changed it
  - Word's "track changes" feature on steroids
- Gets rid of `paper.doc`, `paper2.doc`, `paper-commented.doc`, `paper-FINAL.doc`, `paper-FINALFINAL.doc`
- git is the best version-control system - it combines the benefits of "track changes" with that of backups

- Git is a formal *distributed version control system*

- Git is a formal *distributed version control system*
- It is an easy way to keep track of all the revisions you have saved

- Git is a formal *distributed version control system*
- It is an easy way to keep track of all the revisions you have saved
- You only have *one* version of a file at any one time

- Git is a formal *distributed version control system*
- It is an easy way to keep track of all the revisions you have saved
- You only have *one* version of a file at any one time
- You can see the entire history of a file easily



- Git is a formal *distributed version control system*
- It is an easy way to keep track of all the revisions you have saved
- You only have *one* version of a file at any one time
- You can see the entire history of a file easily
- You can see exactly what changed in each new commit

- Git is a formal *distributed version control system*
- It is an easy way to keep track of all the revisions you have saved
- You only have *one* version of a file at any one time
- You can see the entire history of a file easily
- You can see exactly what changed in each new commit
- There are *many* different GUIs so you don't have to deal with the command line

- Git is a formal *distributed version control system*
- It is an easy way to keep track of all the revisions you have saved
- You only have *one* version of a file at any one time
- You can see the entire history of a file easily
- You can see exactly what changed in each new commit
- There are *many* different GUIs so you don't have to deal with the command line
  - Rstudio can also do most things

- Git is a formal *distributed version control system*
- It is an easy way to keep track of all the revisions you have saved
- You only have *one* version of a file at any one time
- You can see the entire history of a file easily
- You can see exactly what changed in each new commit
- There are *many* different GUIs so you don't have to deal with the command line
  - Rstudio can also do most things
- ((show example git history))

## WHY USE GIT?

- Makes it easy to collaborate - no emailing files back and forth, no need to send an email so that you don't clobber coauthor's changes in dropbox

## WHY USE GIT?

- Makes it easy to collaborate - no emailing files back and forth, no need to send an email so that you don't clobber coauthor's changes in dropbox
- But mainly for yourself!

## WHY USE GIT?

- Makes it easy to collaborate - no emailing files back and forth, no need to send an email so that you don't clobber coauthor's changes in dropbox
- But mainly for yourself!
- You'll generate a documented record of your actions that is also a backup of your project (dissertation?) at every stage of development

## WHY USE GIT?

- Makes it easy to collaborate - no emailing files back and forth, no need to send an email so that you don't clobber coauthor's changes in dropbox
- But mainly for yourself!
- You'll generate a documented record of your actions that is also a backup of your project (dissertation?) at every stage of development
- When you return to your code six months from now, you won't have to wonder what it is or what you were thinking



- Github is the most popular online git service

- Github is the most popular online git service
  - There are many others, such as Gitlab

- Github is the most popular online git service
  - There are many others, such as Gitlab
- Each project gets a repository (“repo”)

- Github is the most popular online git service
  - There are many others, such as Gitlab
- Each project gets a repository (“repo”)
- Each repo is version-controlled (using git)

- Github is the most popular online git service
  - There are many others, such as Gitlab
- Each project gets a repository (“repo”)
- Each repo is version-controlled (using git)
- Default is open-source (public)

- Github is the most popular online git service
  - There are many others, such as Gitlab
- Each project gets a repository (“repo”)
- Each repo is version-controlled (using git)
- Default is open-source (public)
- You can make repos private (for a fee - students for free, though)

- Github is the most popular online git service
  - There are many others, such as Gitlab
- Each project gets a repository (“repo”)
- Each repo is version-controlled (using git)
- Default is open-source (public)
- You can make repos private (for a fee - students for free, though)
- This file is a part of my “math-camp” repo here

- Github is the most popular online git service
  - There are many others, such as Gitlab
- Each project gets a repository (“repo”)
- Each repo is version-controlled (using git)
- Default is open-source (public)
- You can make repos private (for a fee - students for free, though)
- This file is a part of my “math-camp” repo here
  - Feel free to fork-edit-pull request any changes!



EDIT TEXT

---

## WRITE USING PLAIN TEXT!

---

- What if we write an article now that gets famous?

## WRITE USING PLAIN TEXT!

---

- What if we write an article now that gets famous?
- 20 years later, some grad student wants to extend our work

## WRITE USING PLAIN TEXT!

- What if we write an article now that gets famous?
- 20 years later, some grad student wants to extend our work
- How did we make Figure 1?

## WRITE USING PLAIN TEXT!

- What if we write an article now that gets famous?
- 20 years later, some grad student wants to extend our work
- How did we make Figure 1?
- Non-plain text files may be unusable 20 years from now

## WRITE USING PLAIN TEXT!

- What if we write an article now that gets famous?
- 20 years later, some grad student wants to extend our work
- How did we make Figure 1?
- Non-plain text files may be unusable 20 years from now
- Bonus: plain text files are usually *much* smaller than their Word/pdf counterparts

- If you write using plain text, you'll want an editor

- If you write using plain text, you'll want an editor
- I like emacs, but it can be a hassle to get set up



- If you write using plain text, you'll want an editor
- I like emacs, but it can be a hassle to get set up
  - Working with R use ESS, rmarkdown use poly-mode

- If you write using plain text, you'll want an editor
- I like emacs, but it can be a hassle to get set up
  - Working with R use ESS, rmarkdown use poly-mode
- Other alternatives:

- If you write using plain text, you'll want an editor
- I like emacs, but it can be a hassle to get set up
  - Working with R use ESS, rmarkdown use poly-mode
- Other alternatives:
  - Sublime Text

- If you write using plain text, you'll want an editor
- I like emacs, but it can be a hassle to get set up
  - Working with R use ESS, rmarkdown use poly-mode
- Other alternatives:
  - Sublime Text
  - Vim

- If you write using plain text, you'll want an editor
- I like emacs, but it can be a hassle to get set up
  - Working with R use ESS, rmarkdown use poly-mode
- Other alternatives:
  - Sublime Text
  - Vim
  - RStudio (for R analysis)

## # Markdown

You can use markdown to write plain text that  
[contain links](https://google.com). Markdown can also manage your  
references and bibliography [@wlezien1995].

As a quick aside, we can also write really nice math in markdown. Inline math goes between single dollar signs: ( $\beta = 3$ ) and display math uses double dollar signs:

$$\frac{1}{3} x = \pi$$

REPRODUCE





- It's hard to reproduce a lot of research

- It's hard to reproduce a lot of research
- You'll sometimes have to modify a table in a paper months (years) later due to a reviewer or whatever

- It's hard to reproduce a lot of research
- You'll sometimes have to modify a table in a paper months (years) later due to a reviewer or whatever
- How did you produce that figure/table the first time?

- It's hard to reproduce a lot of research
- You'll sometimes have to modify a table in a paper months (years) later due to a reviewer or whatever
- How did you produce that figure/table the first time?
- Solution: Integrate code and paper with rmarkdown

Rmarkdown lets you put code inside your markdown document like so:

```
```{r}  
x <- c(1, 2, 3)  
mean(x)  
```
```

- Run data analysis in R, write with r/markdown, track with git, etc

- Run data analysis in R, write with r/markdown, track with git, etc
- It's complicated but

- Run data analysis in R, write with r/markdown, track with git, etc
- It's complicated but
- Everything is free



- Run data analysis in R, write with r/markdown, track with git, etc
- It's complicated but
- Everything is free
- Everything is open-source and runs on Linux, Mac, and Windows

- Run data analysis in R, write with r/markdown, track with git, etc
- It's complicated but
- Everything is free
- Everything is open-source and runs on Linux, Mac, and Windows
- Your work can be done in a portable, documented, reproducible way

- Run data analysis in R, write with r/markdown, track with git, etc
- It's complicated but
- Everything is free
- Everything is open-source and runs on Linux, Mac, and Windows
- Your work can be done in a portable, documented, reproducible way
- These programs work well with each other

How to install and setup this stuff?

# SETUP

How to draw an owl

1.



2.



How to install and setup this stuff?

1. Draw some circles

2. Draw the rest of the fucking owl