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?
- Academic writing is messy

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?
- 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

Version control

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

Version control

- 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

Version control

- 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

Version control

- 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`

Version control

- 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

- Git is a formal *distributed version control system*

Git

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

Git

- 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

- 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

- 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

- 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

- 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

- 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

- Github is the most popular online git service

Github

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

Github

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

Github

- 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

- 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

- 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

- 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

- 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

Editor

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

Editor

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

Editor

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

Editor

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

Editor

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

Editor

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

Editor

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

```markdown
# Markdown

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

Markdown - math

```
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

Reproducible research

- It’s hard to reproduce a lot of research

Reproducible 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

Reproducible 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
- How did you produce that figure/table the first time?

Reproducible 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
- How did you produce that figure/table the first time?
- Solution: Integrate code and paper with rmarkdown

Rmarkdown

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

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

Overview

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

Overview

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

Overview

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

Overview

- 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

Overview

- 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

Overview

- 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

Setup

How to install and setup this stuff?

Setup

How to install and setup this stuff?