

Organizing Your Research Workflow with Rstudio

A Practical Guide to Save You From Hating Yourself Later

Eli Kravitz

2/28/2019

Introduction

- I will give an **opinionated** way for you to manage your research workflow in Rstudio.
 - I've learned this through trial and error... mostly error.
- I spend most of my day coding in Rstudio. Why not do *everything* there.
- Take parts of this you like, ignore parts you don't
- Many people are more qualified to talk about this
 - none of them will come here and do it for free. You're stuck with me

Scenarios (totally not based on personal experience)

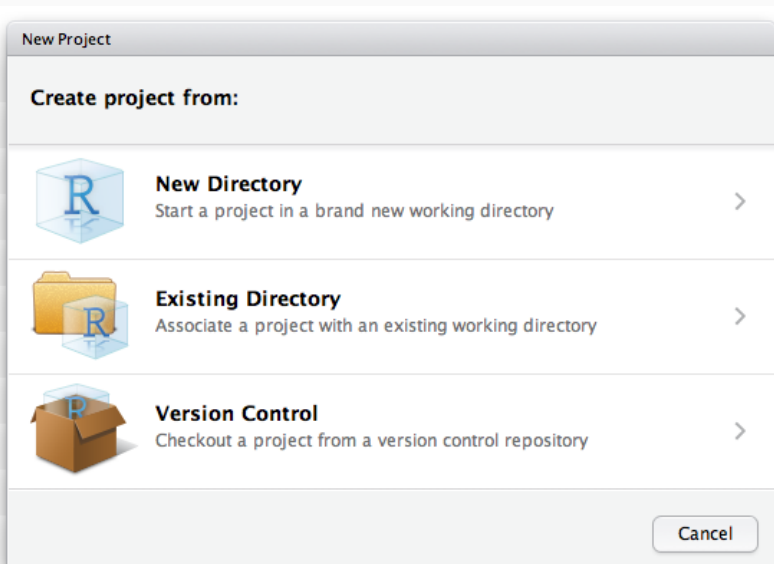
- **Scenarios 1:** You work on a project, complete 90% of it, then stop working on it for 2 years.
 - **Nothing works** on your new computer, you can't locate all the data and files you need.
- **Scenarios 2:** You have a simulation working perfectly. You make a lot of changes. **A few days later** you can't remember why you made these changes, and now **your simulation does not work correctly**.
- **Scenarios 3:** You decide to rewrite most of the paper, show it to your adviser, and they tell you it was better before.
 - You only have the old draft of the paper in PDF. You DO NOT want to retype it.

Storing Everything: Rprojects

- I organize all my research projects the same way.
 - Show folder structure
- Make the directory a Rproject.
- R projects let you:
 - Divide your work naturally
 - Store environments for every project
 - “freeze” R in a state that works (advanced, won't cover)
- Integrates nicely other things we'll cover

Making an R project

- Go to File → New Project



Making an R project

- Follow the prompts to make your project.
- You're done. Easy.
- Now do this **every time** you start a new research project.

Relative Filepaths and here()

- Does your R code start with this?

```
setwd("C:\\Users\\Eli\\path\\that\\only\\I\\have")  
mydata = read.csv(data.csv)
```

- This is **bad**. As soon as you rename or move directories, it breaks.
- What if someone else wants to use your code? What if you want to use a different computer?

Relative Filepaths and here()

- Putting your files in an Rproject already helps. The directory is to to where the .Rproj file is.

```
getwd()
```

```
## [1] "/Users/kravitze/Documents/SafeCafe/Code"
```

- Let's do better with the here package.

Relative Filepaths and here()

- Here looks for a .Rproj file and makes file paths relative

```
here()
```

```
## [1] "/Users/kravitze/Documents/SafeCafe"
```

```
here("Inner_Directory")
```

```
## [1] "/Users/kravitze/Documents/SafeCafe/Inner_Directory"
```

- Code won't need to change between computers
- Example: Load data

```
my_data = here("Data", "my_file.csv")
```

Saving Progress with Github

- Now nothing is tied to the filepaths on our computer.
- How can we **back everything up** and **make it available on multiple computers**?
- Github!

Saving Progress with Github

- Setting up Github is simple, but many steps.
 - See links at the end for instructions
- I will go over what it can do for you, and why you should work with it.

Saving Progress with Github

- Github let's you track versions on your file
- Github saves the history of your file
- Made changes to your code → you can **revert back to an earlier**. version.
- Works with .tex too!
- Show example on Github (code and .tex)

Saving Progress with Github

- Github integrates really well with Rstudio
 - Comes pre-installed on current version of Rstudio
- Point-and-click integration
- Use this and make your life easier

Make your presentations in Rstudio

- What else can we do in Rstudio? **Presentations** in Rmarkdown
- Beamer is the worst. Does anyone find this intuitive?

```
\begin{itemize}
```

```
\item
```

I organize all my research projects the same way.

```
\begin{itemize}
```

```
\tightlist
```

```
\item
```

Show folder structure

```
\end{itemize}
```

```
\item
```

Make the directory a Rproject.

```
\item
```

Make your presentations in Rstudio

- Look at how much more readable this is:

```
### Slide Title
```

```
* Bullet
```

```
  - Sub bullet
```

```
* Bullet
```

Make your presentations in Rstudio

- Compare tables
- Latex:

```
\begin{table}  
\begin{center}  
  \begin{tabular}{| l | c | r }  
    \hline  
    1 & 2 & 3 \\ \hline  
    4 & 5 & 6 \\ \hline  
    7 & 8 & 9 \\  
    \hline  
  \end{tabular}  
\end{center}  
\end{table}
```


Make your presentations in Rstudio

- Rmarkdown:

Table Header	Second Header	Third Header
Table Cell	Cell 2	Cell 3
Cell 4	Cell 45	Cell 6

Make your presentations in Rstudio

- You can create .tex files too. Just add to beginning of Rmarkdown

output:

```
beamer_presentation:  
  includes:  
    keep_tex: yes
```

- Many of you will need to do parallel computing for your research.
- Setting seeds is tricky
- Draw two examples on board

- Fix this with DoParallel and DoRNG
- Automatically changes RNG type, feeds correct RNG to workers
- Saves the **seeds as attribute**.
 - You can recreate these results!

DoRNG and Reproducibility

- So easy, no reason not to do this
- Setup:

```
library("parallel"); library("doParallel"); library("doRNG")  
cl = makeForkCluster(nnodes = detectCores()-1)  
registerDoParallel(cl)  
registerDoRNG(seed = 123)
```

DoRNG and Reproducibility

- Running reproducible parallel computing (easy too!)

```
results = foreach(i = 1:Nsim) %dorng% {  
  thing_I_want_in_parallel()  
}
```

- Saves seed as attribute:

```
attr(results, which = "rng")
```

```
## [[1]]
```

```
## [1]          407    365146966    287832543 -1097700716    196
```

```
## [7]    773696795
```

```
##
```

```
## [[2]]
```

```
## [1]          407 2125690035 1592067806    788949561 14931958
```

```
## [7] -261291987
```

- I did not go over anything in detail, why anything works
- I wanted to demonstrate *why* you want to use these tools
- Next slides have resources for implementing all of this. They can teach you better than I can.

- [Rprojects](#)
- [Github](#)
- [Github and Rstudio](#)
- [Here function](#)

- [Presentations in Rmarkdown](#)
 - I really, really like this website
- [Reproducible Parallel](#)