

# GETTING STARTED AND BEST PRACTICES

Jeff Goldsmith, PhD  
Department of Biostatistics

# What is R?

- Language and environment for statistical computing
- Based on the (proprietary) S language, but open source and open development



# Why is R good?

- Powerful
- Flexible
- Extendable – “base” R vs the collection of R packages
- Active community
- Free
- RStudio

# Why is R bad?

- Not easy to learn
- Not designed for “modern” challenges
- No central support
- No central coordination of extensions / packages
- No “guarantees”
- Not always fast

# Why are we using R?

- One of the recognized “data science” languages (with good reason)
- Extensions matter a lot, and we’ll use them extensively

# Why are we using RStudio?

- Makes life much easier for useRs (not a typo – people who use R are sometimes referred to as useRs...)
- The RStudio folks are also leading the development of a new analytic framework within R, and that work is integrated into RStudio



# Working in R

- Console – where commands are executed
- Scripts – where sequences of commands are saved for reproducibility
- Functions – operations performed on inputs, usually producing outputs

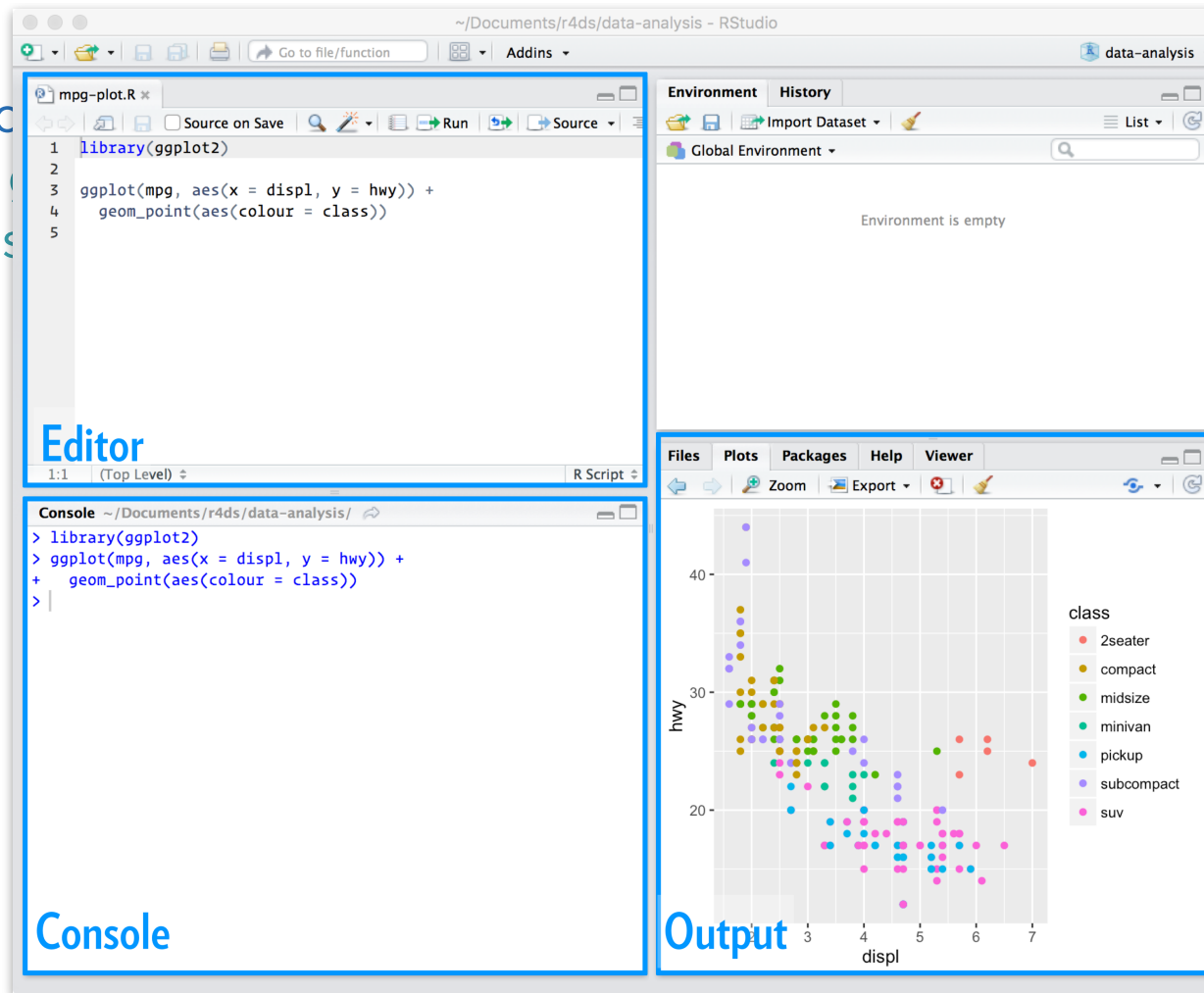
# Working in RStudio

- Rstudio is an Integrated Development Environment (IDE)
  - It's got everything you need to do data science in R
  - This IDE is one of the better reasons to use R ...



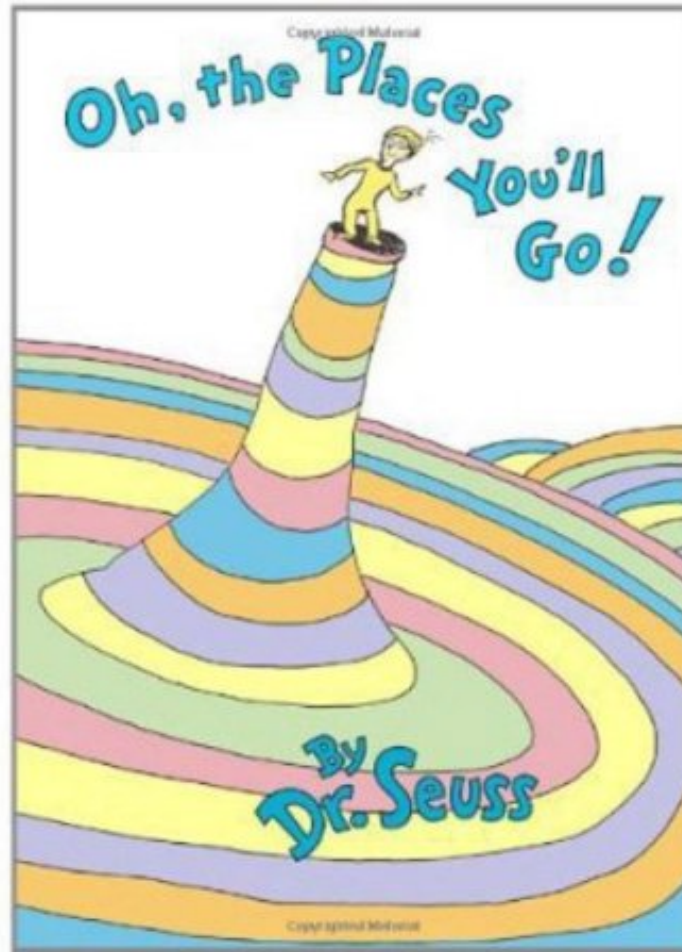
# Working in RStudio

- Rstudio
  - It's
  - This



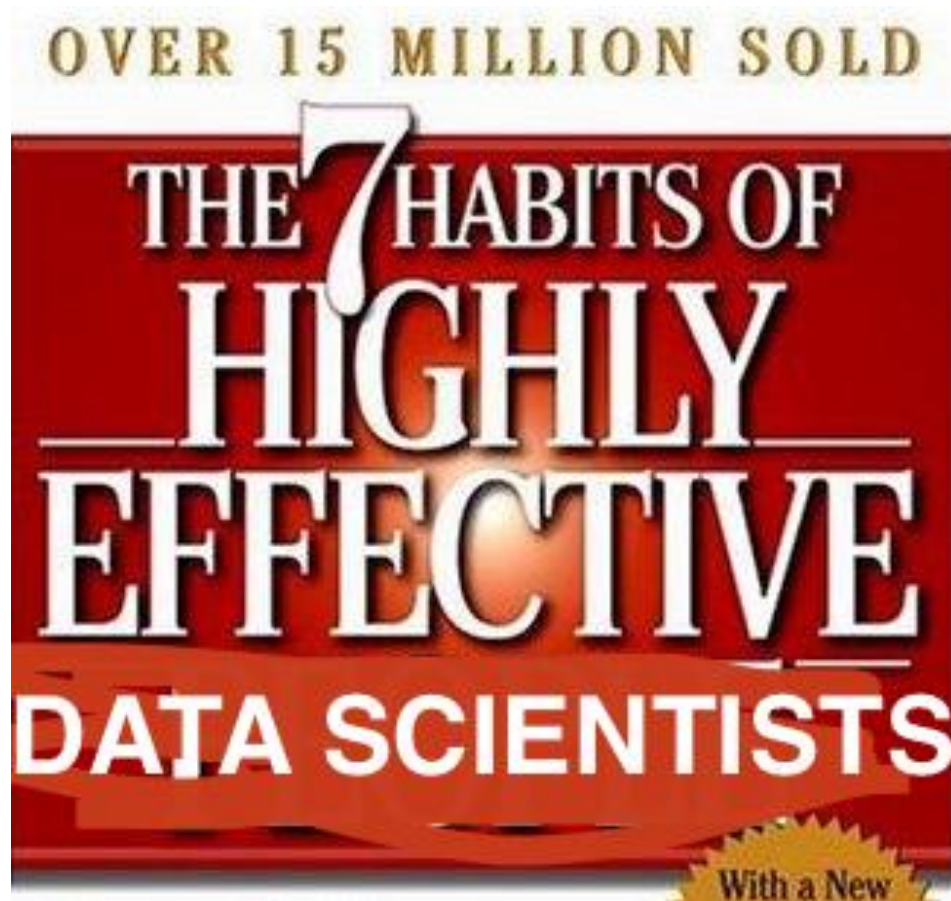
R for Data Science

# You'll have big projects...



# ... someday.

- Better get ready by establishing good habits now!



# Code

- Code is case sensitive
- There is no autocorrect
- Establish a variable naming convention
  - `this_is_snake_case`
  - `this.is.period.case`
  - `thisIsLowerCamelCase`
  - `ThisIsUpperCamelCase`
  - `ThisIsNoTaNaMiNgCoNvEnTiOn`
- Your names should match your regex skills
  - If you don't have regex skills, your variable and file names should be as simple as possible.
- Extensive comments will save you headache

# Code

- Code is case sensitive
- There is no autocorrection
- Establish a variable naming convention
  - `this_is_snake_case`
  - `this.is.period.case`
  - `thisIsLowerCamelCase`
  - `ThisIsUpperCamelCase`
  - `ThIsIsNoTaNaMiNg`
- Your names should make sense
  - If you don't have real names, they should be as simple as possible
- Extensive comments will save you headache



Actual  
programming

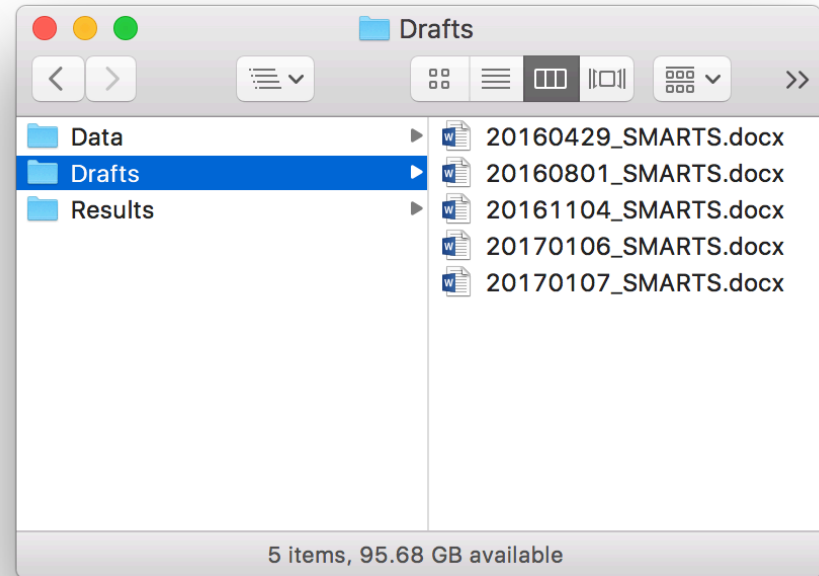
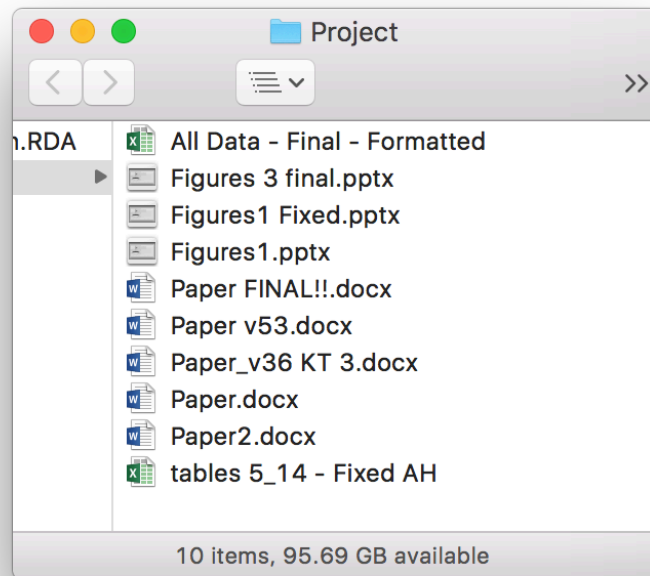


Debating for  
30 minutes on  
how to name a  
variable

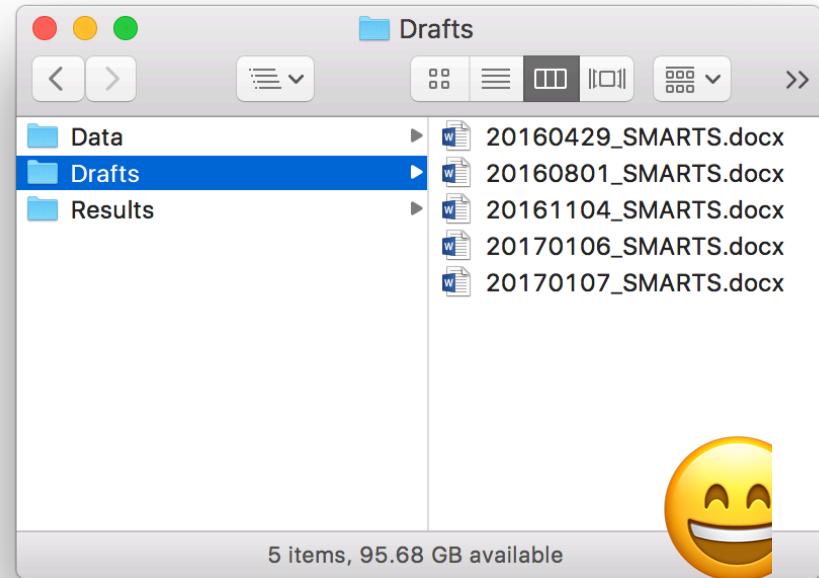
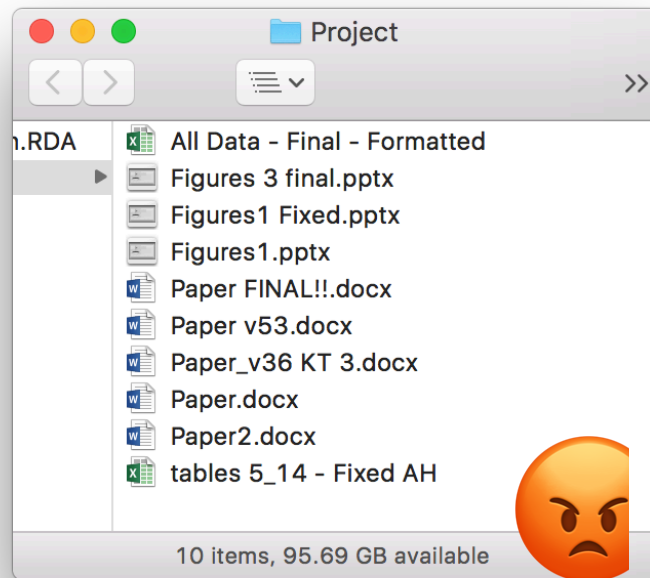
# Some perspective on code

- Treat your inputs (e.g. raw data) and code as “real”
  - Your results are created by input and code, and you can always reproduce your results from these if you need to
- Your code matters
  - It’s one of the most central ways you will communicate. Do it well.
- Plan for mistakes
  - You will make them, and that’s fine. Write code that makes it easy to fix mistakes without breaking the rest of your analysis

# Organizing files

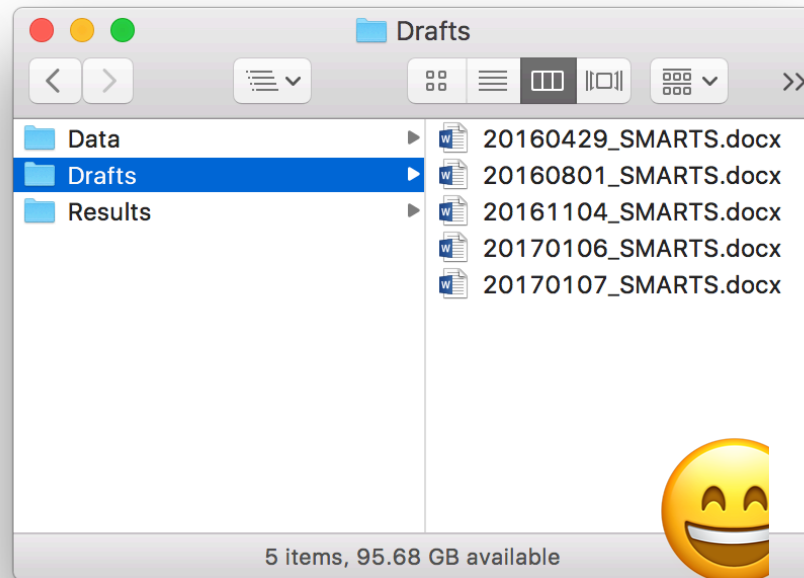
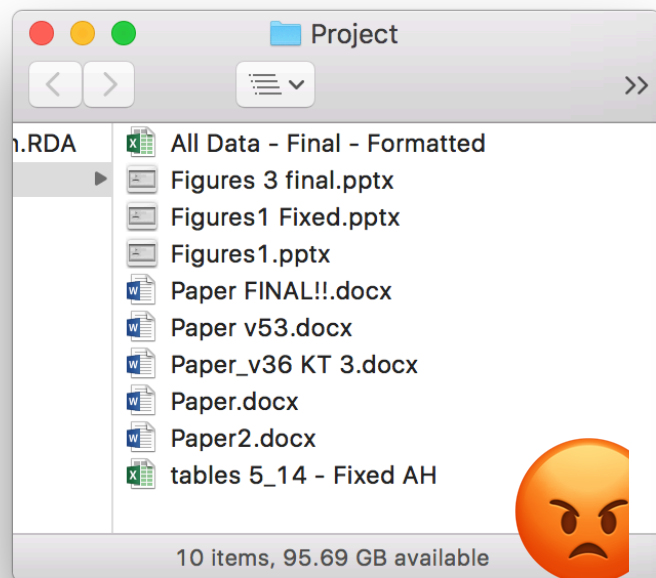


# Organizing files





# Organizing files



the life-changing  
magic of tidying up  
the Japanese art of decluttering  
and organizing

marie kondo

# Some perspective on files

- You will need to find everything again someday. Make sure it's easy to find.
  - Name your files reasonable things
  - Avoid special characters and spaces
  - Put everything for a project in the same place

# Why organization matters

Being organized will frequently make your life easier

- “Your most frequent collaborator is you from six months ago, but you don’t reply to emails”<sup>1</sup>
- Eventually, someone other than you (or even future you) will need to reproduce your results
  - Be ready for that.

1. This version of the quote comes from Karl Broman, who traced it to a tweet: [http://bit.ly/motivate\\_git](http://bit.ly/motivate_git)