A whirlwind tour of Rstudio, R, and Rmarkdown Magic for behavioral and brain scientists

Jenny Rieck & Derek Beaton

May 17 2019

The big outline

- Part 0: Background and Community
- Part 1: Rstudio
- Part 2: Project and environment setup
- Part 3: R (et al., eg Python)
- Part 4: Rmarkdown
- Part 5: Advanced R and beyond
- Part 6: A few of our favorite things

Section 1

Part 0: Background and Community

Part 0: Background and Community

- What this is & isn't; a bunch of things we aren't covering but you should be aware of
  - This is a taste and to bring you into a bigger world
- Centralization, standards
- Help
- Including rigor & reproducibility of packages
- The "tidyverse"
  - Learn it. But don't learn *only* the tidyverse; you'll be lost in base R

R Background

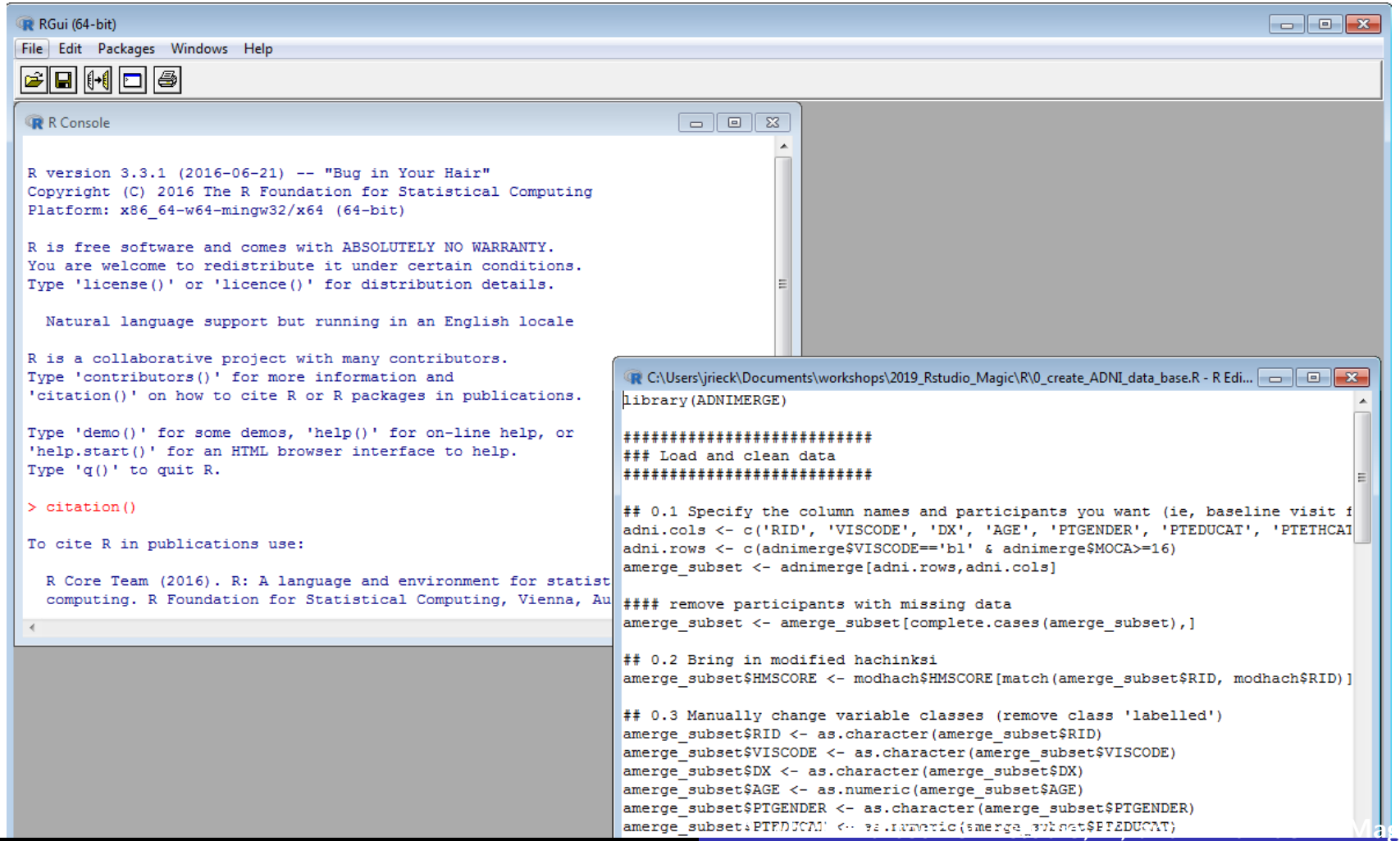
- Created in 1992 by Gentleman & Ihaka

[we] considered the problem of obtaining decent statistical software for our undergraduate Macintosh lab. After considering the options, we decided that the most satisfactory alternative was to write our own. [. . .] Finally we added some syntactic sugar to make it look somewhat like S. We call the result "R".

What is R?

- R is for stats and general purpose programming
- R is a functional language
  - Turing complete – can do anything other languages can do
- R is an environment to inferface with the langauge
  - Console based
  - Type in commands
  - No point-and-click
- R is a collection of tools
  - Pre-packaged software at your disposal
- R is free (as in beer and speech)
  - No cost, no restrictions

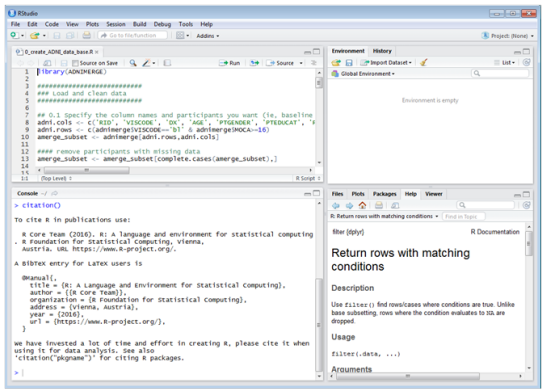
R is a bit ugly



But R has many interfaces

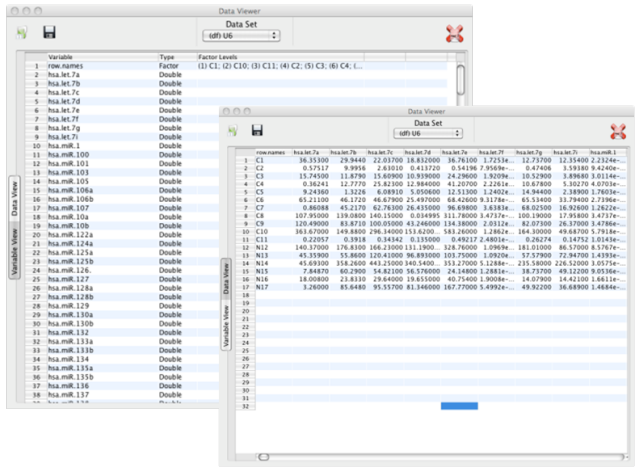
- Today we focus on RStudio (MatLab-like)
- But see also Deducer, RCommander (SPSS-like)

RStudio



RCommander

Deducer



R is a community (actually many communities!)

- Help and resources
- Package development and distribution

R: Help!

- https://www.statmethods.net/
- Online forums (Stack Exchange, r-lists)
- SpringerLink
  - All R books for free (pdf format) or for minimal cost (printed)
- Vignettes
  - step-by-step instruction guides for packages

R Packages

- Packages are bundles of code made by someone (or many people) for everyone to use
  - If you can think of a stats problem, there is a package for it
- Available primarily on CRAN
  - But also github, r-forge

Tidyverse

- something here about tidy

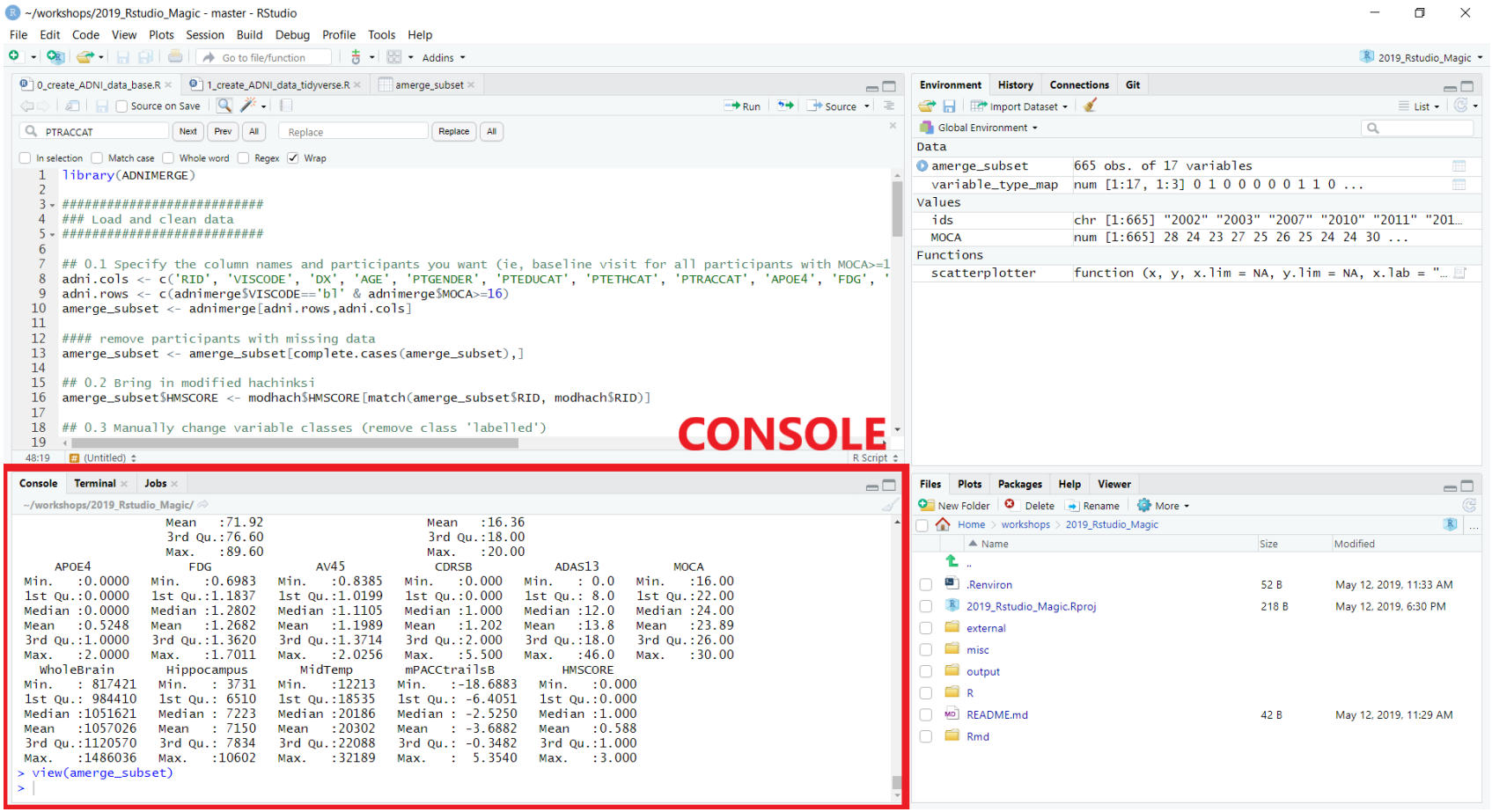
Section 2

Part 1: Rstudio

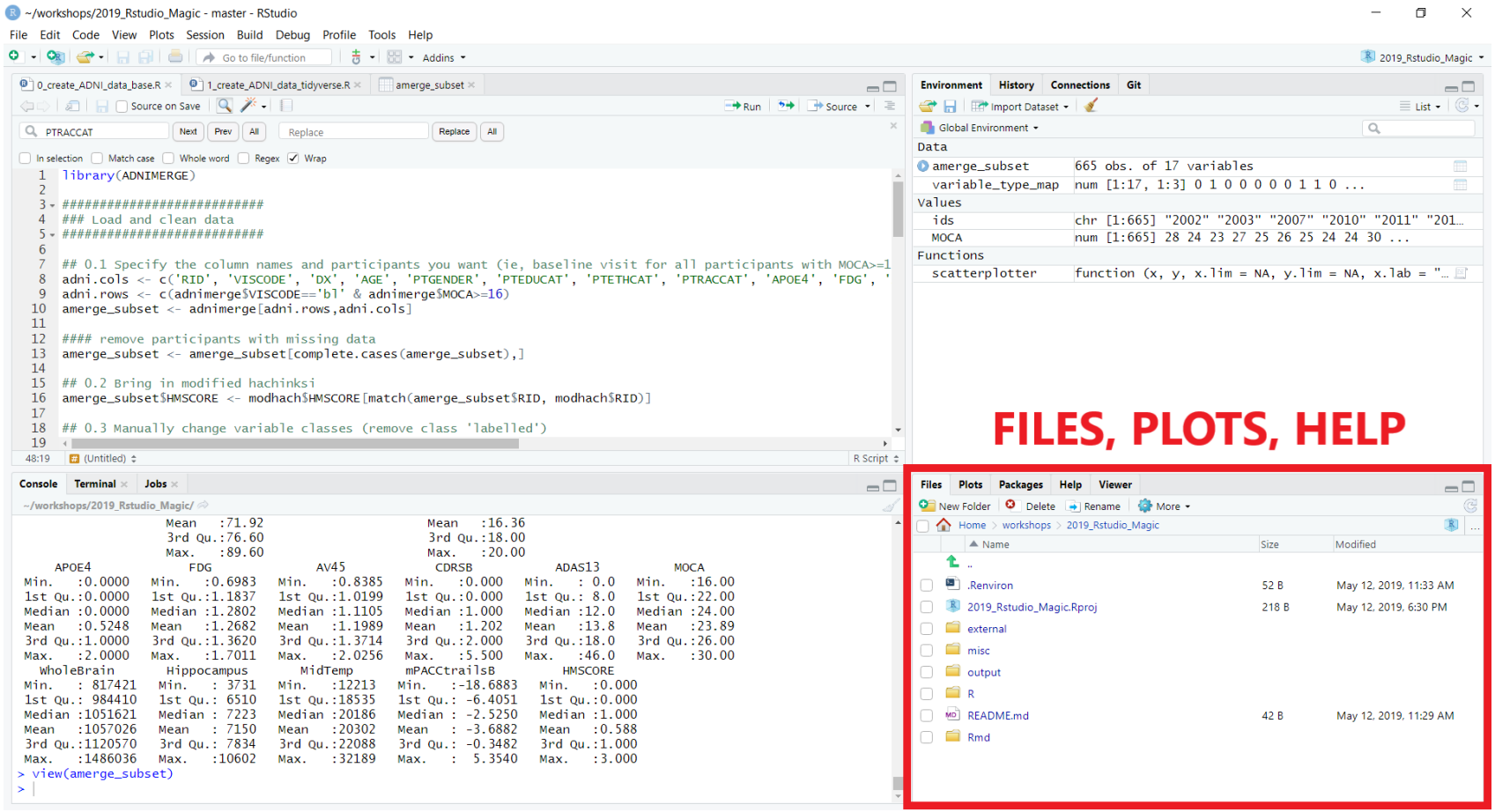
Part 1: RStudio

- Settings, a quit tour through stuff, features
- Examples on getting setup

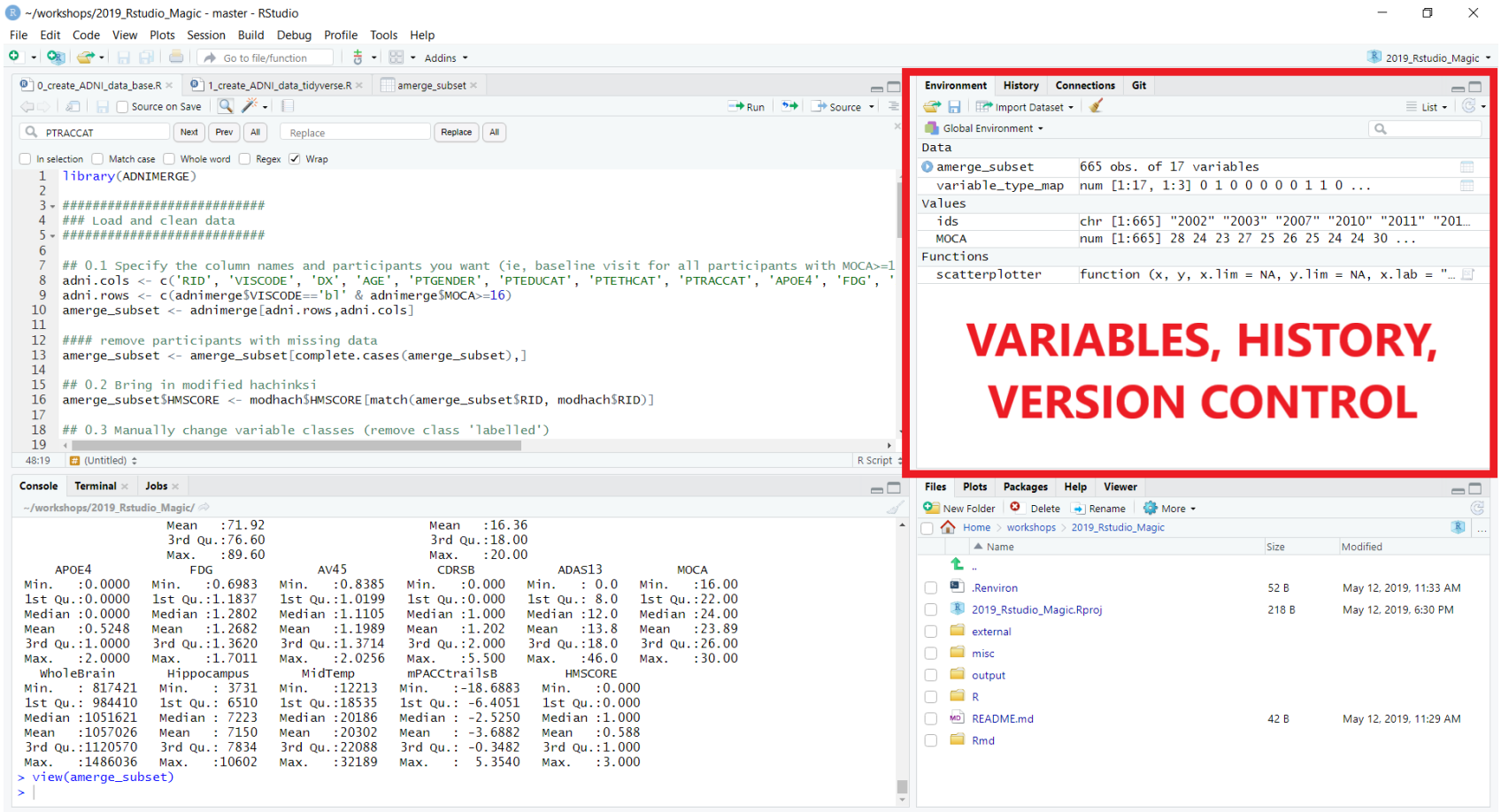
RStudio Environment



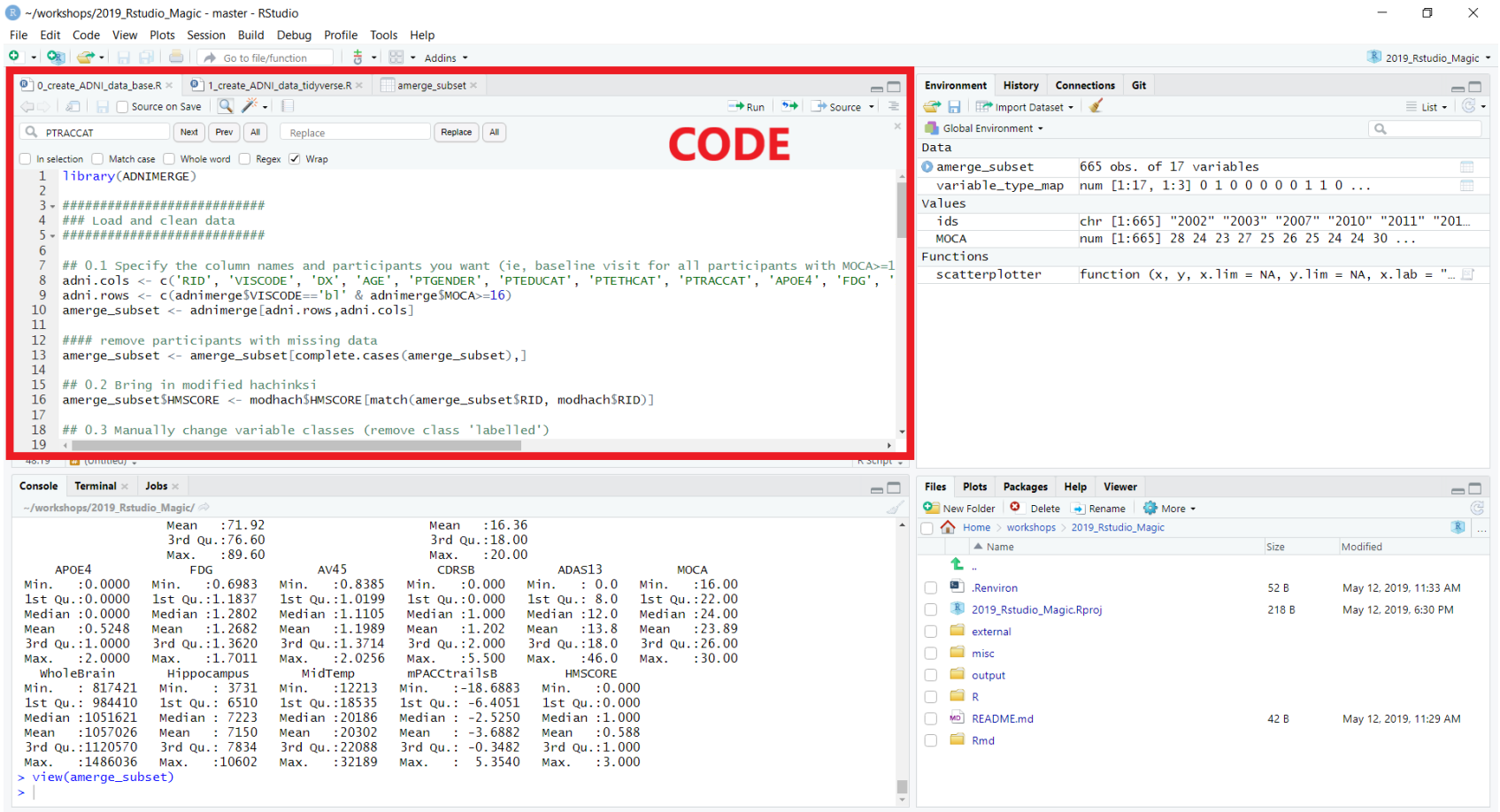
RStudio Environment



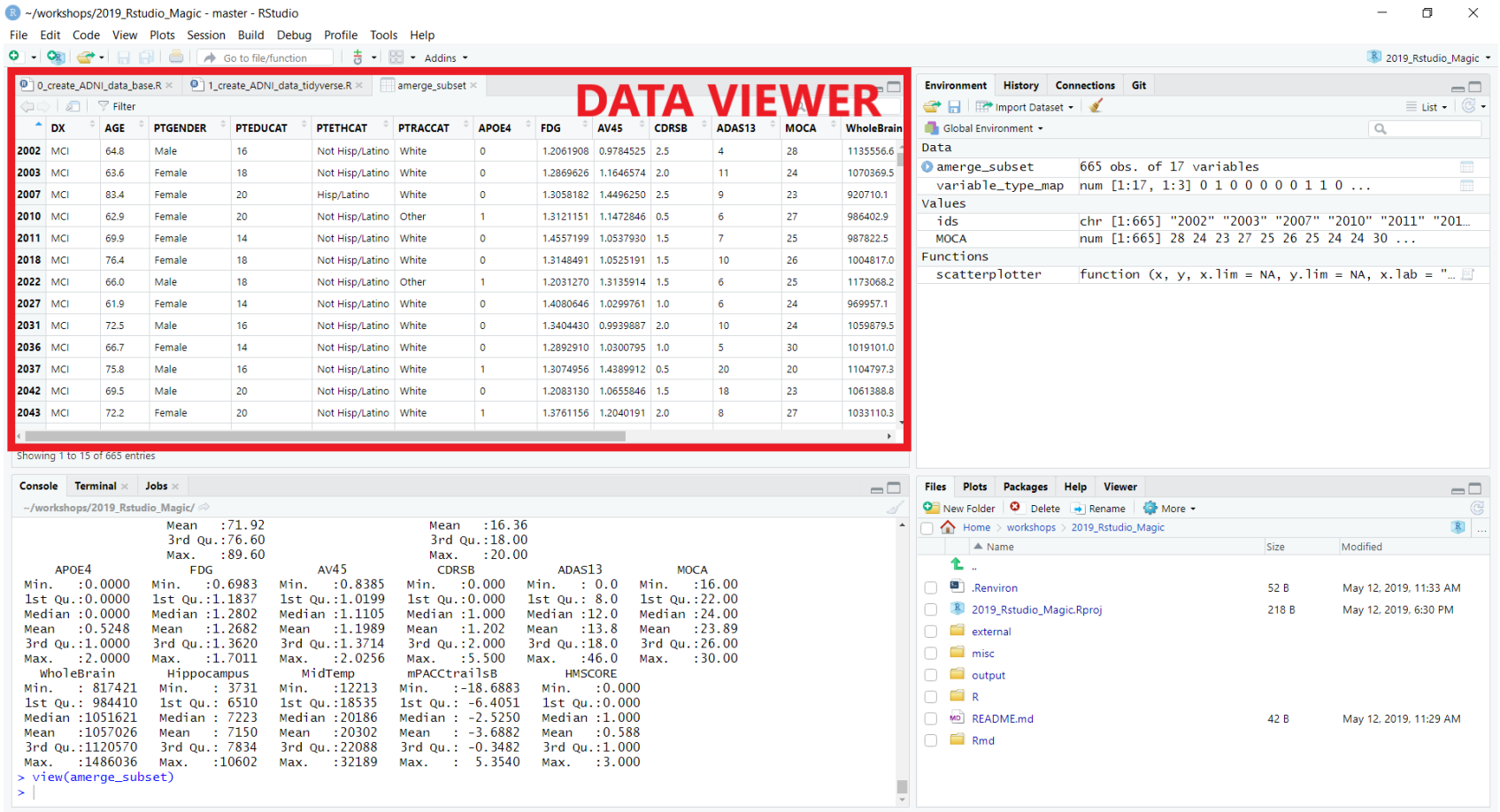
RStudio Environment



RStudio Environment



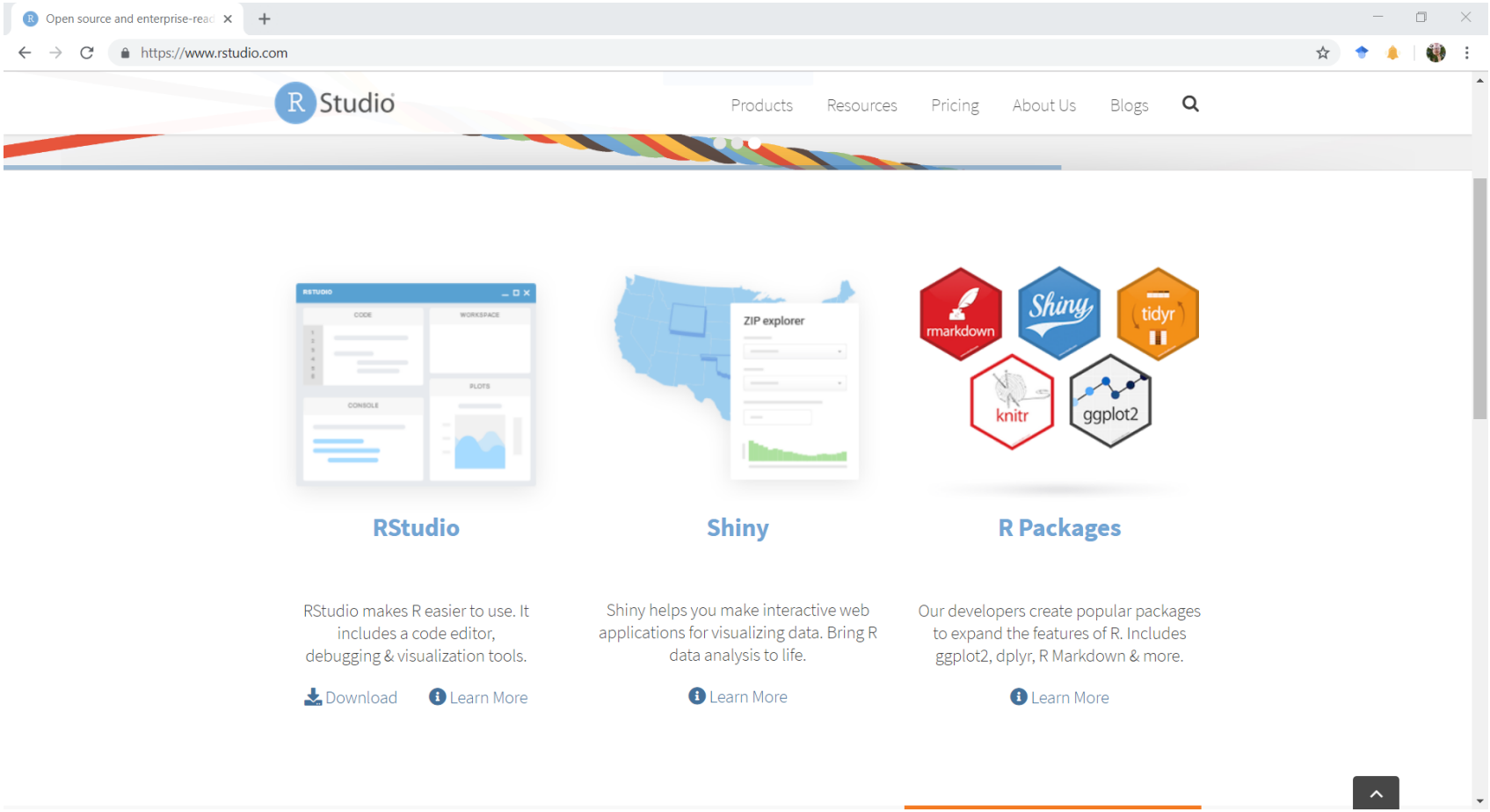
RStudio Environment



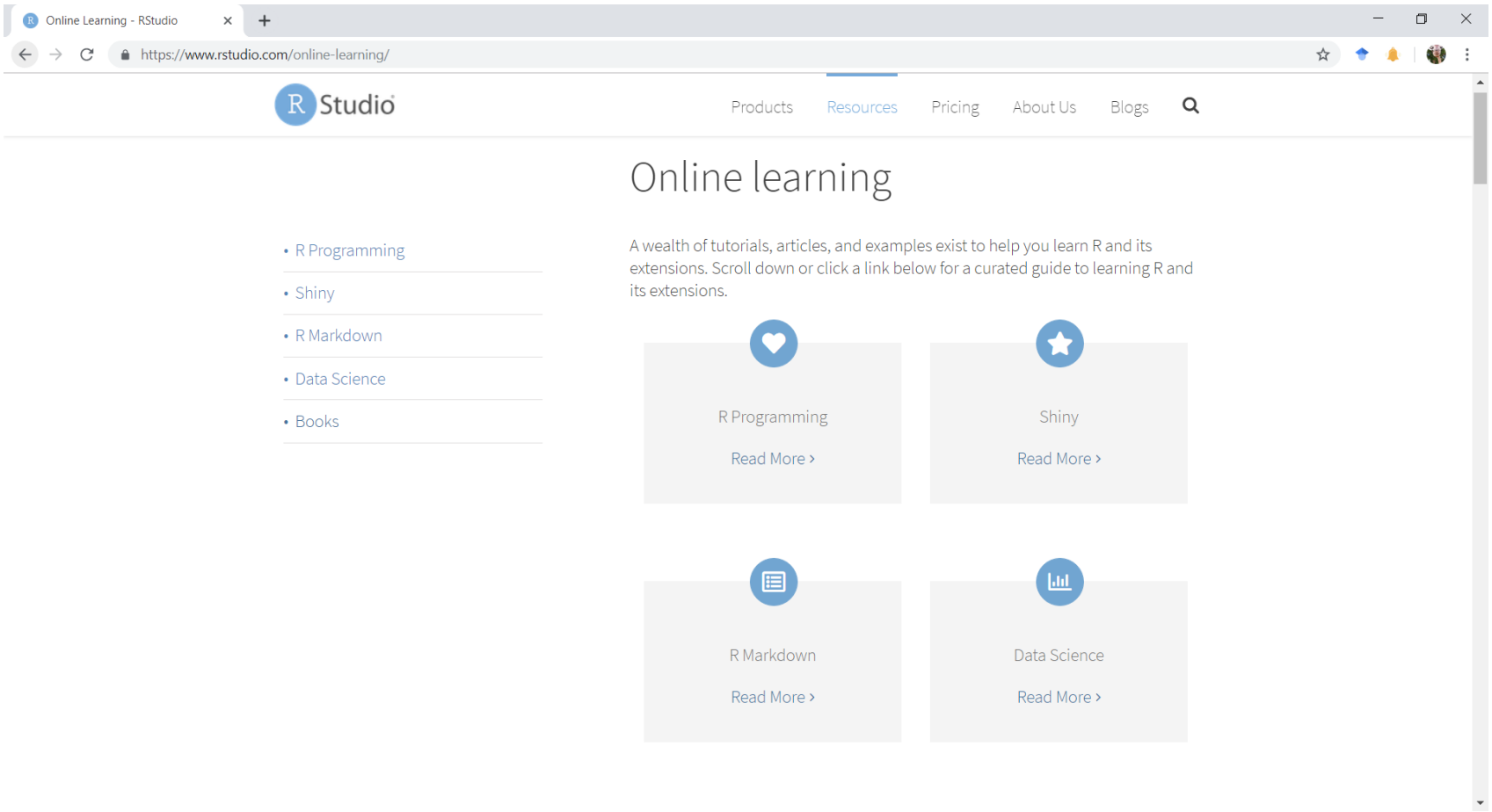
Benefits of RStudio

- Built-in integration with version control (git or SVN)
- Package and documentation generation
- Reproducible science!
  - R Markdown documents
    - Save and execute code
    - Generate high quality reports that can be shared
  - Create presentations (like this one!)
  - Even write papers

RStudio Resources



RStudio Resources



RStudio Resources

Cheatsheets - RStudio

https://www.rstudio.com/resources/cheatsheets/

RStudio | Products | Resources | Pricing | About Us | Blogs

RStudio Cheat Sheets

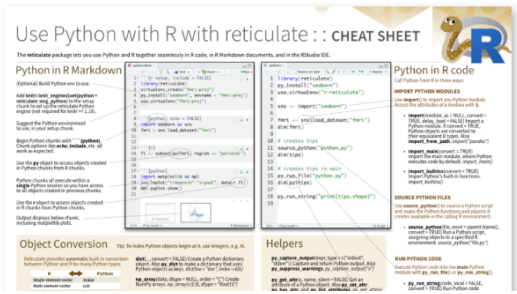
The cheat sheets below make it easy to learn about and use some of our favorite packages. From time to time, we will add new cheat sheets to the gallery. If you'd like us to drop you an email when we do, let us know by clicking the button to the right.

SUBSCRIBE TO CHEAT SHEET UPDATES HERE

- RStudio IDE
- R Markdown
- Shiny
- Package Development
- Data Import
- Data Transformation with dplyr
- Data Visualization with ggplot2
- Apply functions with purrr
- Deep Learning with Keras
- Data Science in Spark with Sparklyr
- String manipulation with stringr
- Dates and times with lubridate

Python with R and Reticulate Cheat Sheet

The reticulate package provides a comprehensive set of tools for interoperability between Python and R. With reticulate, you can call Python from R in a variety of ways including importing Python modules into R scripts, writing R Markdown Python chunks, sourcing Python scripts, and using Python interactively within the RStudio IDE. This cheatsheet will remind you how. Updated 4/19.



Section 3

Part 2: Setup

Part 2: Project and Environment Setup

- Hidden files & whatnot
- Have a structure ready to go on Github
- Explain/walk through
- Discuss the helpful packages above

RStudio Setup

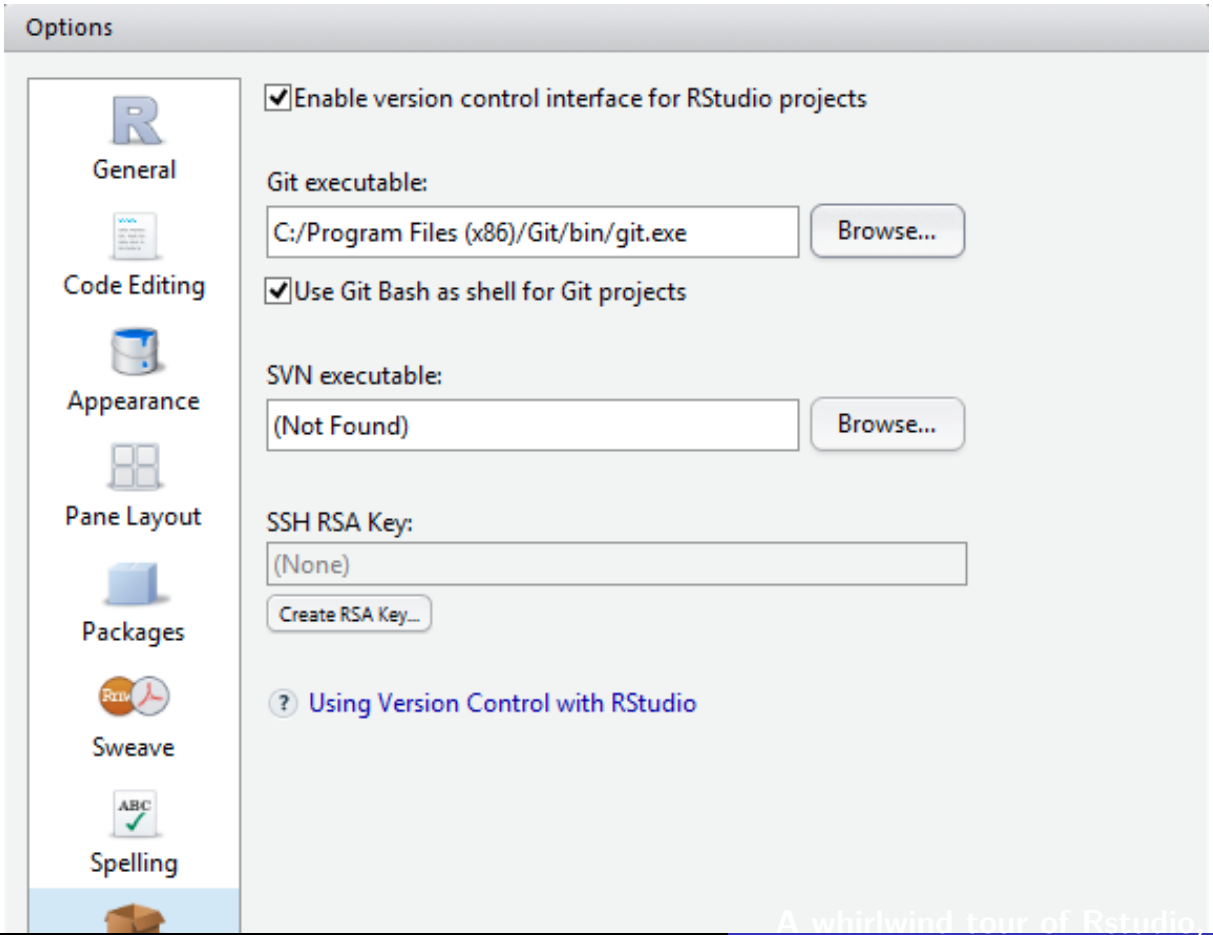
- Download R and Rstudio
- Add-on packages

```
#to install from CRAN
install.pacakges('devtools', depenedencies = TRUE)
#to install from a file
install.packages('/mypath/to/package/ADNIMERGE.tar.gz',
                 type='source', repos=NULL)
#to install from a git  (requires the devtools package)
dev.tools::install_github(Gibbsdavidl/CatterPlots)
```

- See https://jennybc.github.io/2014-05-12-ubc/r-setup.html for a detailed guide

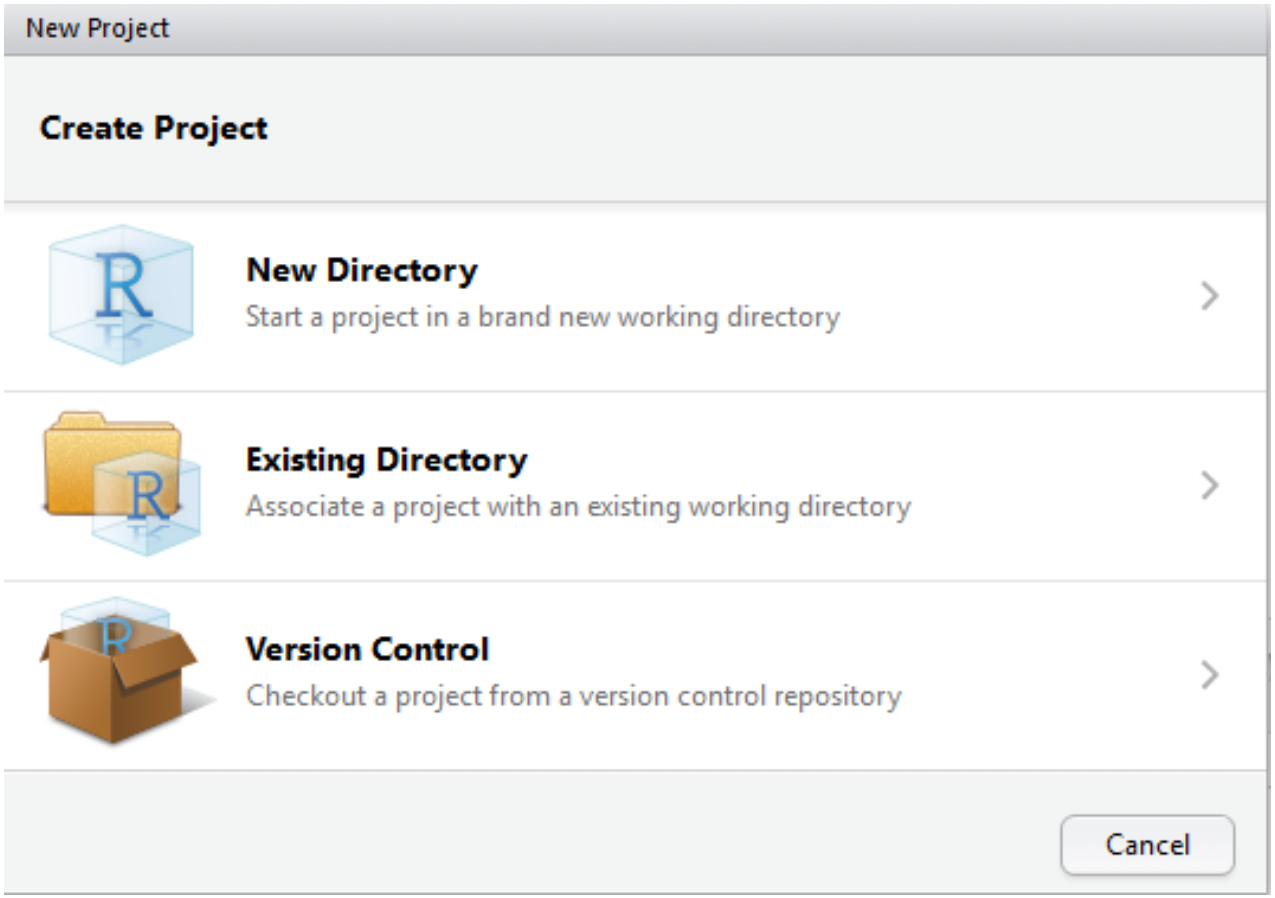
Rstudio Setup: Projects & Git

- Download git and link to RStudio



Rstudio Setup: Projects & Git

- Create a new project File

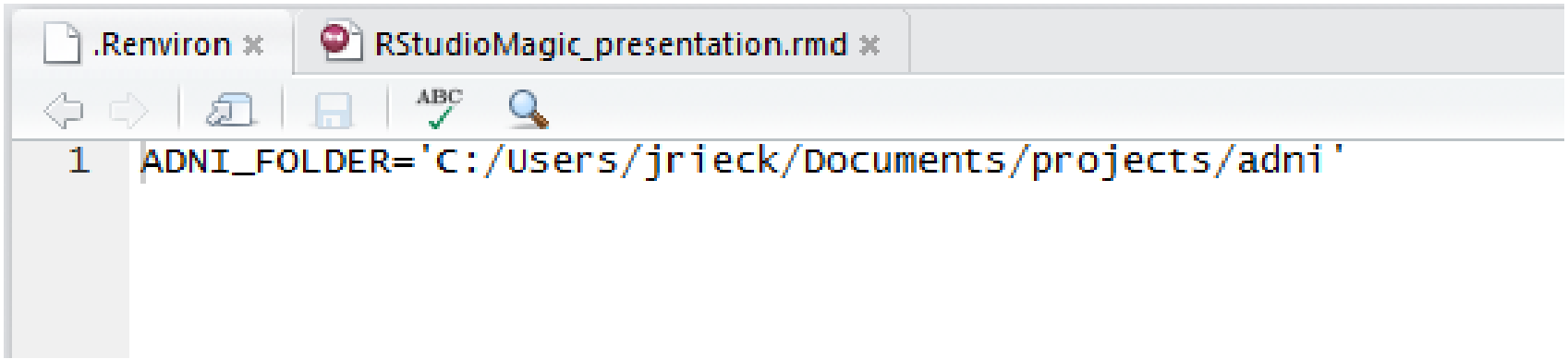


Format .gitignore

- File types to ignore:
  - `.Rproj.user`
  - `.Rhistory`
  - `.Ruserdata`
  - `.Renviron`
  - `.rda` & `.Rdata` (to avoid pushing potentially sensitive data files to git)
  - ** before each extentions will match directories anywhere in the repo

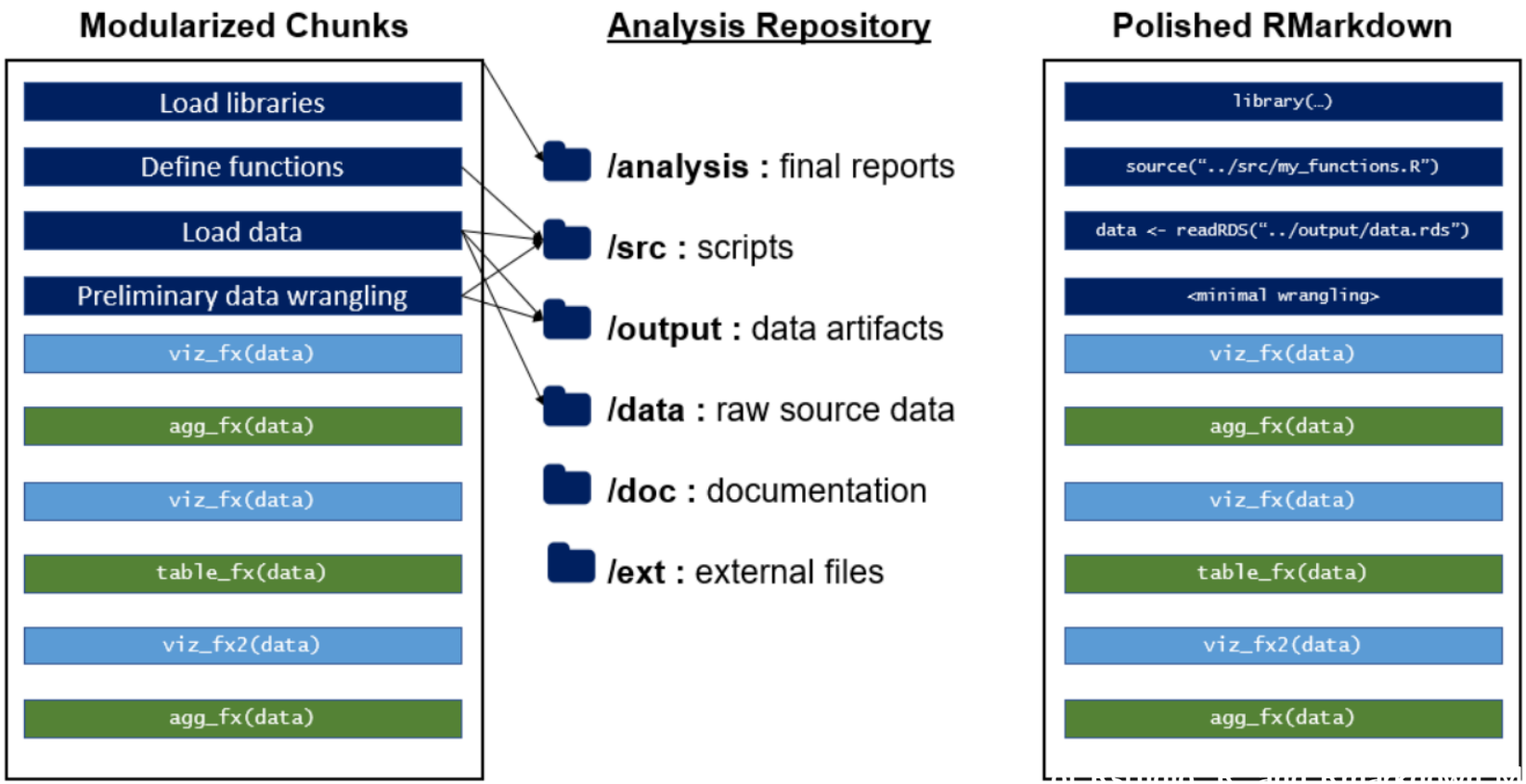
Format environmental variables

- Set environmental variables (ie, directory location of data) to make code generalizable across computers
  - In your project folder create a `.Renviron` file and define variables

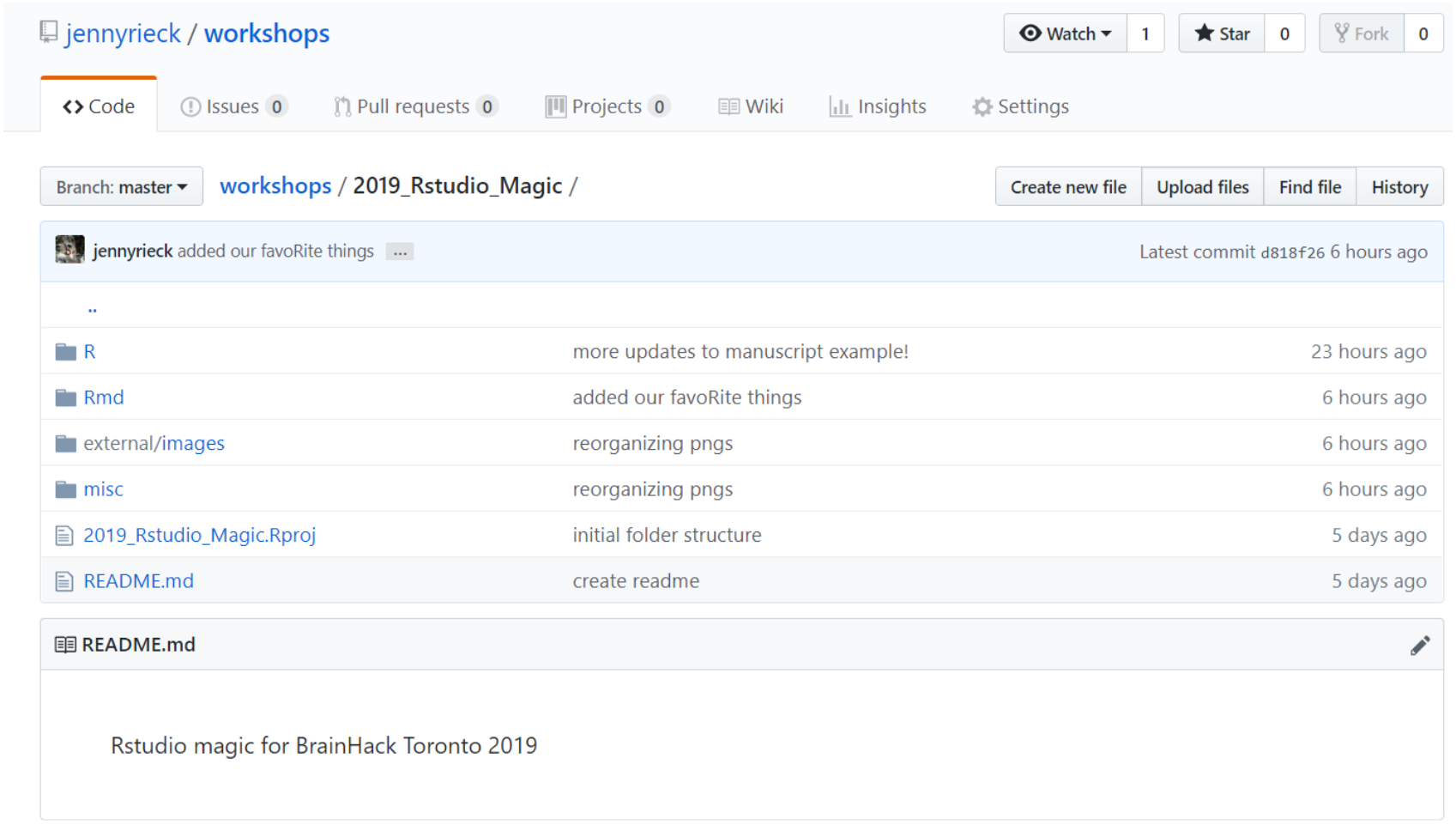


Organize your project folders and markdown

*https://emilyriederer.netlify.com/post/rmarkdown-driven-development/



Organize your project folders and markdown



Section 4

Part 3: R et al

Part 3: R et al

- A bit of background, including idiosyncrasies and unique things about R
  - Especially packages & three ways to install (somehwhat covered above) CRAN, Locally, Git & others (devtools)
  - It's a functional language
  - Data types Including data frames & alts like tibbles
- Read/explore
  - explore .R scripts
- Clean/export
  - Show 0_Create from PCA/MCA with Base, Tidyverse, Plyr (NOT dplyr), data.table
  - Reimport?
  - Analyze With MCA & covstatis

Section 5

Part 4: RMarkdown

Part 4: RMarkdown

- What it is /why to use it
- A short deviation for LaTeX, and new helpers: kable & kableExtra
    - A taxonomy and how to approach this *Tying it all together through here 1: simple RMD* Plot-based visuals
        - Base, gt, ggplot, grobTable()/grid/gridExtra
        - 2: Slides (these ones here)
        - 3: Manuscripts!!
- Reporting/presentin

Section 6

Part 5: Advanced R

Part 5: Some advanced/other things we're not covering

- package development
- Shiny
- SQL
- C/C++
- R2D3

Section 7

Part 6: Extras

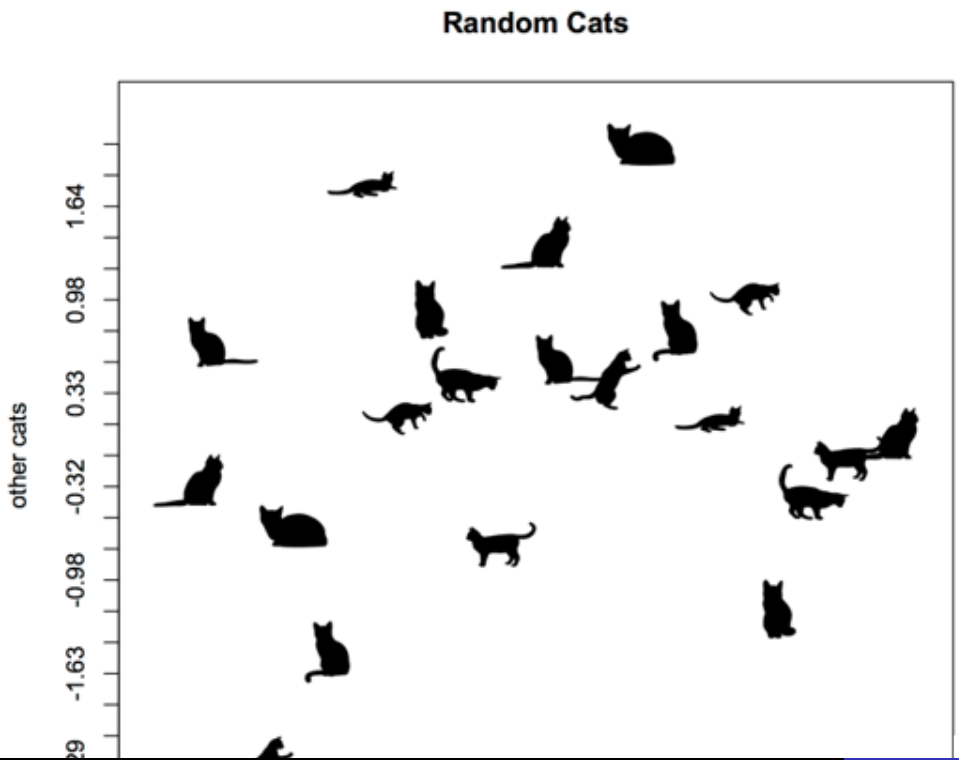
Part 6: A few of our favorite things

- Fun R do-dads

CatterPlot for feline based graphics:

- https://github.com/Gibbsdavidl/CatterPlots

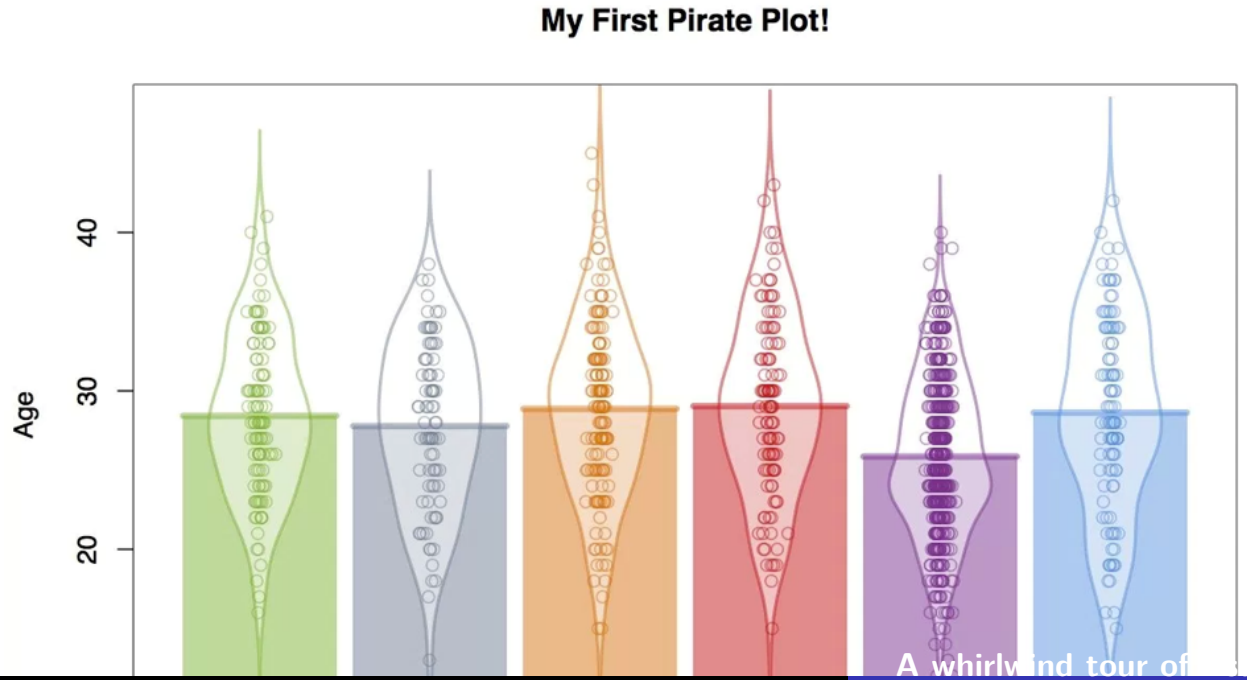
```
dev.tools::install_github(Gibbsdavidl/CatterPlots)
```



What's a pirate's favorite programming language?

- https://cran.r-project.org/web/packages/yarrr/vignettes/pirateplot.html

```
install.packages('yarrr')
```



Color palettes to fit your mood

- https://github.com/karthik/wesanderson

```
dev.tools::install_github(karthik/wesanderson)
```

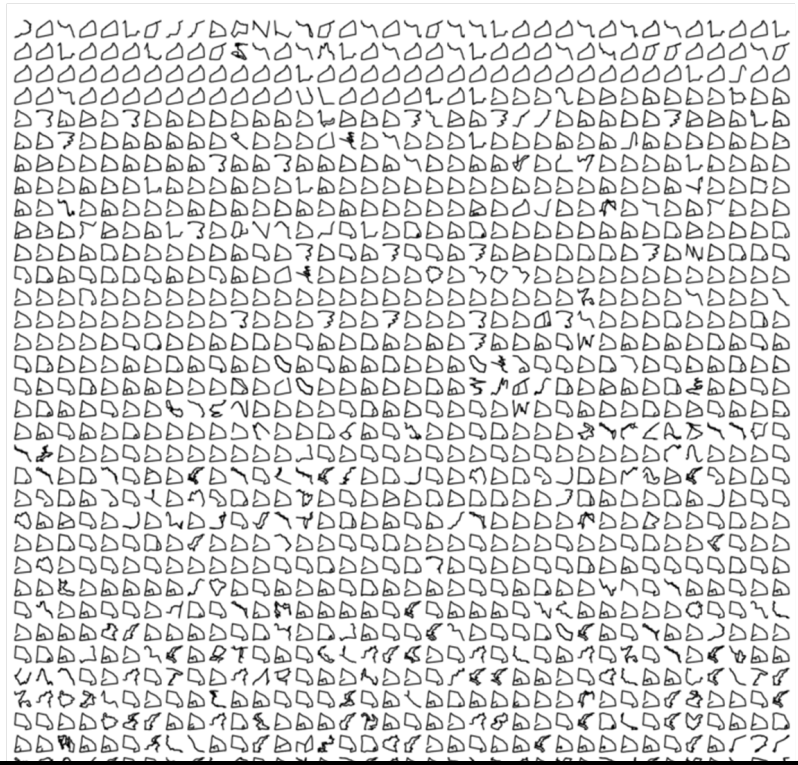
The Life Aquatic with Steve Zissou (2004)

```
wes_palette("Zissou1")
```


```
pal <- wes_palette("Zissou1", 21, type = "continuous")
image(volcano, col = pal)
```

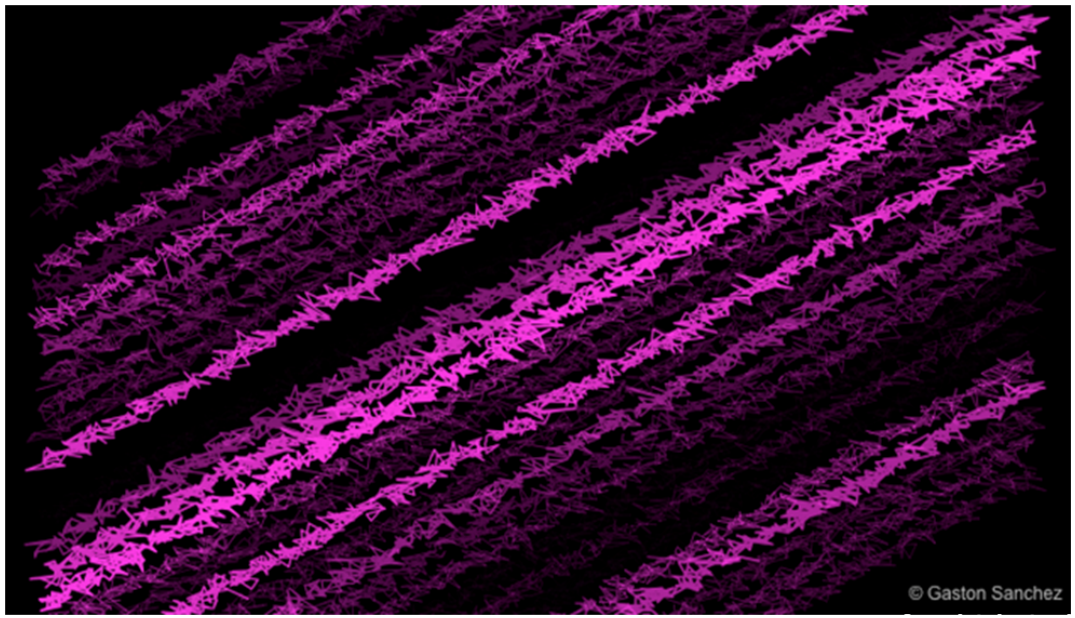
Mapping your Strava routes

- https://www.r-bloggers.com/strava-rides-map-in-r/
- ALSO https://marcusvolz.com/?p=4068
  - `dev.tools::install_github(marcusvolz/strava)`



Make aRt!

- R Graph Gallery
  - http://www.r-graph-gallery.com/
- Rtist: Gaston Sanchez
  - http://gastonsanchez.com/Rtist/




```
# ============================================================================
# Pink Barbs
# ============================================================================
# generate pairs of x-y values
x <- seq(1, 100, length = 1000)
y <- x + rnorm(1000)


png("pink_barbs.png", width = 700, height = 400)
# set graphical paramters
op <- par(bg = "black", mar = rep(0, 4))
# plot
plot(x, y, type = 'n')
for (i in seq(-80, 70, by = 5))
{
  lines(x + rnorm(1000), x + i + rnorm(1000, 2), pch = 19,
        col = hsv(0.85, 1, 1, runif(1000)),
        lwd = sample(seq(0.3, 2, length = 20), 1))
}
# signature
legend("bottomright", legend = "© Gaston Sanchez", bty = "n",
       text.col = "gray70")
# reset par
par(op)
dev.off()
```