

# Reproducible (and collaborative) science through RStudio

A whirlwind tour with R, RMarkdown, Python, LaTeX, and more

Jenny Rieck & Derek Beaton

May 19 2019

# The big outline

- ▶ Part 0: Introduction, background, & RStudio
- ▶ Part 1: Setup & R
- ▶ Part 2: RMarkdown & more
- ▶ Part 3: Advanced, beyond, & our favorites

## Part 0: Introduction, Background, & RStudio

To dive right in

If you want to skip over the background & RStudio, go straight to  
**Part 1: Setup & R**

# Background

- ▶ This is a taste and to bring you into a bigger world
  - ▶ R, Python, SQL, and JavaScript are critical data science tools/languages
- ▶ R (language and community) strongly emphasizes
  - ▶ Centralization & standards
  - ▶ Rigor & reproducibility (packages, RMarkdown)
- ▶ An interesting language
  - ▶ Functional
  - ▶ With a sublanguage (or dialect?): the tidyverse

## R is a community (actually many communities!)

- ▶ Help and resources
- ▶ Package development and distribution
- ▶ An ideal example
  - ▶ Not quite always that way
  - ▶ Strong communal presence

## R: Help!

- ▶ So many websites e.g., <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
- ▶ Git
  - ▶ With open books (via bookdown)
- ▶ Twitter #rstats
- ▶ RStudio (website)
  - ▶ Videos, cheat sheets

# R Packages

- ▶ Packages are bundles of code made by someone (or many people) for everyone to use
  - ▶ There are packages for everything
  - ▶ We'll cover some of the diversity throughout
- ▶ Comprehensive & Reproducible
- ▶ Available primarily on CRAN
  - ▶ But also github (less so: r-forge)

# RStudio

- ▶ IDE: Integrated development environment
- ▶ RStudio: Does so much
  - ▶ We scratch the surface here
- ▶ Quick walk through
- ▶ Followed by specific set up
  - ▶ Generally, but
  - ▶ Also for this workshop

# RStudio Environment

The screenshot shows the RStudio interface with several windows open:

- Script Editor:** Displays R code for creating an ADNI dataset and merging it with a diverse dataset. The code includes data cleaning steps like removing rows with missing data and matching on modhash(RID).
- Console:** Shows the output of the R code execution.
- File Browser:** Shows the project structure with files like `ADNI.RData`, `ADNI_didyverse.R`, and `merge_subset.R`.
- Environment:** Shows the global environment with objects like `adni`, `adni$MOCA`, and `merge_subset`.

# RStudio Environment

The screenshot shows the RStudio interface with several windows open:

- Code Editor:** The main window displays R code for creating an ADNI data subset. The code includes library imports, data loading, merging, and filtering steps.
- Console:** A red box highlights the console output area. It shows statistical summaries for various variables like APOE4, FDG, AV45, CDRSB, ADAS13, and MOCA across different brain regions (Wholebrain, Hippocampus, Midtemp, nPACCtailB, and MSMScore).
- Environment:** The environment pane shows the global environment with objects like `anmerge_subset`, `ids`, and `MOCA`.
- File Browser:** The file browser pane shows the project structure with files like `Renviron`, `2019_Rstudio_Magic.Rproj`, `external`, `mac`, `output`, `R`, and `README.md`.

# RStudio Environment

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Displays the R script `create_ANOVA_data.R`. The code performs data cleaning and merging, including:
  - Specifying column names and participants.
  - Handling missing data.
  - Bringing in modified hashkeys.
  - Manually changing variable classes.
- Console:** Shows statistical summaries for various variables like APOE4, FDG, AV45, CDRSB, ADAS13, MOCA, and hippocampus.
- Environment:** Shows the global environment with objects like `merge_subset` (665 obs. of 17 variables), `ids` (chr vector), and `MOCA` (num vector).
- File Browser:** A red box highlights the file browser showing the project structure and files.
- Header Bar:** Includes tabs for Environment, History, Connections, and Git, along with a search bar and a tab for "2019\_Rstudio\_Magic".

**FILES, PLOTS, HELP**

# RStudio Environment

The screenshot shows the RStudio interface with several windows open:

- Code Editor:** Shows R code for creating an ADNI data subset. The code includes library imports, data loading, merging, and subset selection. A red box highlights the final command: `> view(merge_subset)`.
- Environment:** Shows the `merge_subset` object, which is a data frame with 665 observations and 17 variables. It lists variables like `ADAS13`, `CDRSB`, `MOCA`, and `PTEDUCAT`.
- File Browser:** Shows the project structure under `workshops > 2019_Rstudio_Magic`. It includes files like `README.md`, `environment.Rproj`, and `output`.
- Text Overlay:** A large red text overlay in the center-right area reads "VARIABLES, HISTORY, VERSION CONTROL".

# RStudio Environment

The screenshot displays the RStudio interface with several panes:

- Code pane:** Shows R code for creating an ADNI data subset. The code includes library imports, data loading, cleaning, and subset selection. It uses functions like `library`, `read.csv`, `subset`, and `complete.cases`.
- Console pane:** Displays statistical summaries for variables like APOE4, FDG, AV45, CDRSB, ADAS13, and MOCA. For example, APOE4 has a mean of 71.92 and a median of 70.00. The FDG variable has a range from 89.60 to 208.00.
- Environment pane:** Shows the global environment with objects like `anmerge\_subset` (665 obs., 17 variables), `variable\_type\_map`, `values`, and `functions` (e.g., `scatterplotter`).
- File browser:** Shows the project structure with files like `Renviron`, `2019.Rstudio\_MAGIC.Rproj`, `external`, `mac`, `output`, `R`, and `README.md`.

```
library(ADNImerGE)
#####
## Load and clean data
#####
## 0.1 Specify the column names and participants you want (ie, baseline visit for all participants with MOCA>=1
admin.cols <- c("RID", "VISCODE", "DX", "AGE", "PTGENDER", "PTEDUCAT", "PTETHCAT", "PTRACCAT", "APOE4", "FDG", "ADAS13", "CDRSB", "MOCA")
admin.rows <- c(adminmerge$VISCODE=="b1" & adminmerge$MOCA>=16)
anmerge_subset <- adminmerge[admin.rows,admin.cols]
#####
## remove participants with missing data
anmerge_subset <- anmerge_subset[complete.cases(anmerge_subset),]
#####
## 0.2 Bring in modified hachkins
anmerge_subset$MSMSCORE <- modhach$MSMSCORE[match(anmerge_subset$RID, modhach$RID)]
#####
## 0.3 Manually change variable classes (remove class 'labelled')
anmerge_subset$FDG <- as.numeric(as.character(anmerge_subset$FDG))
anmerge_subset$AV45 <- as.numeric(as.character(anmerge_subset$AV45))
anmerge_subset$ADAS13 <- as.numeric(as.character(anmerge_subset$ADAS13))
anmerge_subset$CDRSB <- as.numeric(as.character(anmerge_subset$CDRSB))
anmerge_subset$MOCA <- as.numeric(as.character(anmerge_subset$MOCA))

# Whole brain
# Hippocampus
# Midtemp
# nPACCtailB
# MSMSCORE

# APOE4
Min. : 0,00000 Min. :0.6983 Min. :0.8385 Min. :0,00000 Min. : 0,00000 Min. :16,36
1st Qu.:0,00000 1st Qu.:1,1100 1st Qu.:1,1100 1st Qu.:0,00000 1st Qu.: 8,40000 1st Qu.:17,60
Median :0,00000 Median :1,2802 Median :1,1105 Median :1,00000 Median :10,00000 Median :22,00
Mean   :0,5248 Mean   :1,2682 Mean   :1,1989 Mean   :1,2020 Mean   :11,80000 Mean   :23,89
3rd Qu.:1,00000 3rd Qu.:1,3620 3rd Qu.:1,3714 3rd Qu.:1,20000 3rd Qu.:18,00 3rd Qu.:26,00
Max.  :2,00000 Max.  :1,7013 Max.  :2,0256 Max.  :15,50000 Max.  :46,00 Max.  :30,00

# White matter
# Hippocampus
# Midtemp
# nPACCtailB
# MSMSCORE

# FDG
Min. :14,421 Min. :11,111 Min. :12,213 Min. :18,6883 Min. : 0,00000
1st Qu.: 984410 1st Qu.: 6510 1st Qu.: 5535 1st Qu.: 10,051 1st Qu.: 0,00000
Median :1051621 Median : 7223 Median :20186 Median : -2,5250 Median :1,00000
Mean   :1057026 Mean   : 7150 Mean   :20302 Mean   : -3,6882 Mean   :0,588
3rd Qu.:1120570 3rd Qu.: 7834 3rd Qu.:22088 3rd Qu.: -0,3482 3rd Qu.:1,00000
Max.  :1486036 Max.  :10602 Max.  :32189 Max.  : 5,3540 Max.  :3,00000

> view(anmerge_subset)
> |
```

# RStudio Environment

The screenshot displays the RStudio interface with several windows open:

- Data Viewer:** A central window titled "DATA VIEWER" showing a table of 665 observations across 17 variables. The variables include DX, AGE, PTGENDER, PTEDUCAT, PTRECAT, PTRACCAT, APOE4, FDG, AV45, CDRSB, ADAS13, MOCA, WholeBrain, and Hippocampus.
- Global Environment:** A window showing the global environment with objects like anerage\_subset, variable\_type\_map, values, and functions.
- File Browser:** A window showing the file structure under "workshops > 2019\_Rstudio\_Magic".
- Console:** A window showing R code and its output, including descriptive statistics for variables like APOE4, FDG, AV45, CDRSB, ADAS13, MOCA, WholeBrain, Hippocampus, Midtemp, nPACCtrailsB, and HMSCore.
- Terminal:** A window showing the command line interface.
- Jobs:** A window showing the current jobs.

# Some 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
  - ▶ Python, D3 (JavaScript), SQL, Shiny, LaTeX, Git/SVN, HTML/CSS, and so much more.
- ▶ This workshop
  - ▶ Will walk you through some of this (and more)
  - ▶ See [https://github.com/jennyrieck/workshops/tree/master/2019\\_Rstudio\\_Magic](https://github.com/jennyrieck/workshops/tree/master/2019_Rstudio_Magic)

## RStudio is more

- ▶ Not just an IDE
- ▶ A company
- ▶ A community
- ▶ A conference
- ▶ A centralized resource

# RStudio Resources

The screenshot shows the RStudio website homepage. At the top, there's a navigation bar with links for Products, Resources, Pricing, About Us, Blogs, and a search icon. Below the navigation is a decorative banner featuring a colorful, abstract graphic of overlapping colored bands.

**RStudio**: A screenshot of the RStudio IDE interface, showing the code editor, workspace, and plots.

**Shiny**: An image of a map of the United States with a "ZIP explorer" interface overlaid.

**R Packages**: Icons for several popular R packages: `markdown`, `Shiny`, `tidyverse`, `knitr`, and `ggplot2`.

**RStudio** description: RStudio makes R easier to use. It includes a code editor, debugging & visualization tools.

**Shiny** description: Shiny helps you make interactive web applications for visualizing data. Bring R data analysis to life.

**R Packages** description: Our developers create popular packages to expand the features of R. Includes `ggplot2`, `dplyr`, `R Markdown` & more.

At the bottom, there are download and learn more buttons for each section, and a horizontal orange progress bar.

# RStudio Resources

The screenshot shows a web browser window for 'Online Learning - RStudio' at the URL <https://www.rstudio.com/online-learning/>. The page features a navigation bar with links for Products, Resources (which is underlined), Pricing, About Us, Blogs, and a search icon. The main content area has a title 'Online learning' and a sidebar with links for R Programming, Shiny, R Markdown, Data Science, and Books. Below this is a section with a paragraph about learning R and its extensions, followed by four cards: 'R Programming' (with a heart icon), 'Shiny' (with a star icon), 'R Markdown' (with a document icon), and 'Data Science' (with a bar chart icon). Each card includes a 'Read More >' link.

Online Learning - RStudio

https://www.rstudio.com/online-learning/

R Studio

Products Resources Pricing About Us Blogs

## Online learning

- R Programming
- Shiny
- R Markdown
- Data Science
- Books

A wealth of tutorials, articles, and examples exist to help you learn R and its extensions. Scroll down or click a link below for a curated guide to learning R and its extensions.

R Programming  
Read More >

Shiny  
Read More >

R Markdown  
Read More >

Data Science  
Read More >

# RStudio Resources

Cheatsheets - RStudio x + - □ x

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

R Studio Products Resources Pricing About Us Blogs Q

## 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.

Use Python with R with reticulate :: CHEAT SHEET

The reticulate package makes it easy to have and use Python in R. It's a Python interface, just like R itself.

Python in R Markdown

Object Conversion

Helpers



## Part 1: Setup & R

# Project and Environment Setup

Somethign...?

## 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

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

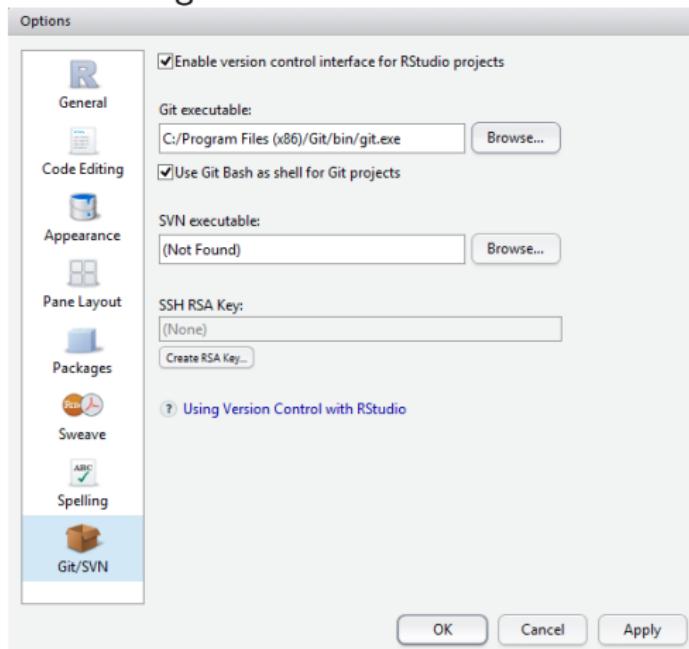
## For safety & collaboration

- ▶ Projects
  - ▶ SOMETHING?

# Git & Projects

## ► Git

- Download git and link executable within RStudio



# Projects through Git

- ▶ Create a new project File

New Project

Create Project

---

 **New Directory**  
Start a project in a brand new working directory >

---

 **Existing Directory**  
Associate a project with an existing working directory >

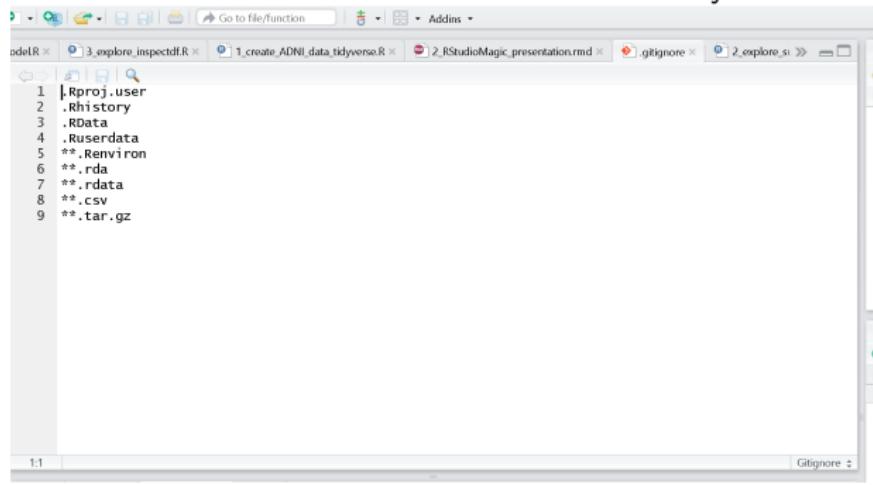
---

 **Version Control**  
Checkout a project from a version control repository >

Cancel

# Format .gitignore

- ▶ File types to ignore via version control
  - ▶ \*\* before each extension will match directories anywhere in the



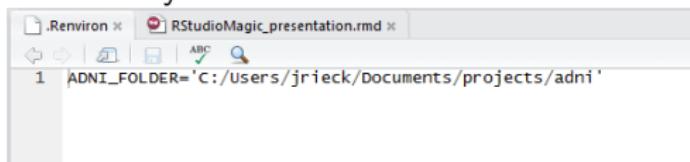
The screenshot shows the RStudio interface with multiple tabs open at the top. The current tab is ".gitignore". The content of the .gitignore file is displayed as follows:

```
1 .Rproj.user
2 .Rhistory
3 .RData
4 .Ruserdata
5 **,.Renvironment
6 **,.rda
7 **,.rdata
8 **,.csv
9 **,.tar.gz
```

The word "repo" is visible in the bottom left corner of the editor area.

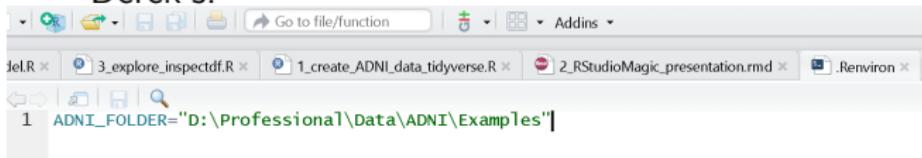
# Format environmental variables

- ▶ Set environmental variables (ie, directory location of data) to make code generalizable across computers
  - ▶ In **your** project folder create a `.Renvironment` file and define variables
  - ▶ Jenny's:



```
1 ADNI_FOLDER='C:/users/jrieck/Documents/projects/adni'
```

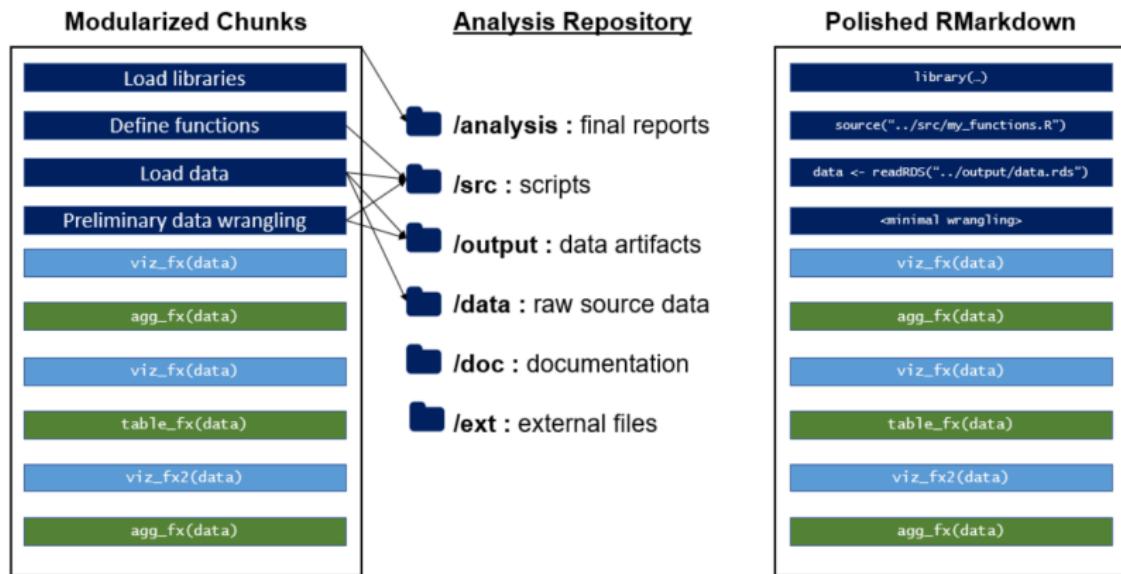
- ▶ Derek's:



```
1 ADNI_FOLDER="D:\Professional\Data\ADNI\Examples"
```

# Organize your project folders and markdown

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



# Organize your project folders and markdown

jennyrieck / workshops

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master workshops / 2019\_Rstudio\_Magic / Create new file Upload files Find file History

jennyrieck added our favoRite things ... Latest commit d818f26 6 hours ago

..

R	more updates to manuscript example!	23 hours ago
Rmd	added our favoRite things	6 hours ago
external/images	reorganizing pngs	6 hours ago
misc	reorganizing pngs	6 hours ago
2019_Rstudio_Magic.Rproj	initial folder structure	5 days ago
README.md	create readme	5 days ago

README.md

Rstudio magic for BrainHack Toronto 2019

# RStudio Setup

- ▶ Download R and Rstudio
- ▶ Add-on packages

```
#to install from CRAN
install.packages('devtools', dependencies = 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)
```

## 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 interface with the language
  - ▶ 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

## A little bit more about beer

- ▶ R is free (as in beer and speech)
  - ▶ No cost, no restrictions
  - ▶ Revolution/MRAN
  - ▶ etc...

- ▶ A bit of background, including idiosyncrasies and unique things about R
  - ▶ Especially packages & three ways to install (somewhat covered above) CRAN, Locally, Git & others (devtools)
  - ▶ It's a functional language
  - ▶ Data types Including data frames & alts like tibbles

R

Some more about R here...

## Tidyverse

- ▶ something here about tidy
- ▶ Learn it. But don't learn *only* the tidyverse; you'll be lost in base R

- ▶ A bit of background, including idiosyncrasies and unique things about R
  - ▶ Especially packages & three ways to install (somewhat 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

## Read in and create your dataframe

- ▶ ADNI Dataset adnimerge package
  - ▶ Reduce full dataset to only those participants (rows) and variables (columns) you're interested in
- ▶ Two methods to create your dataframe
  - ▶ using base R functions: 0\_create\_ADNI\_data\_base.R
  - ▶ Using tidyverse functions:  
`1_create_ADNI_data_tidyverse.R`

## Screenshots

Explanation

## Exploring your data

- ▶ Many packages to help explore and describe your data:
  - ▶ `summarytools`: `2_explore_summarytools.R`
  - ▶ `inspectdf`: `3_explore_inspectdf.R`
  - ▶ `DataExplorer`: `4_explore_DataExplorer_one_liner.R`

Code w/ eval=F

## Hard Break

- ▶ DataExplorer is dangerous
- ▶ Blind analyses can be *criminal*
  - ▶ de Leeuw paper quote
  - ▶ DEREK RANTS, PER USUAL.

## Analyze your data

- ▶ Linear models: 5\_linear\_model.R

## Screenshots / Code w/ eval=F

## Get experimental

- ▶ Explain motivation, not method
- ▶ covSTATIS: `6_covstatis_example.R`

## Part 2: RMarkdown

# 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

## RMarkdown Don(u)'ts

- ▶ Don't hardcode values
- ▶ Don't hardcode absolute file paths
- ▶ Don't do complicated database queries
- ▶ Don't litter
  - ▶ avoid eval=FALSE
  - ▶ reduce repeated code by making functions
- ▶ Don't load unnecessary libraries
- ▶ More at: <https://emilyriederer.netlify.com/post/rmarkdown-driven-development/>

## Part 3: Advanced R

## Some advanced/other things we're not covering

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

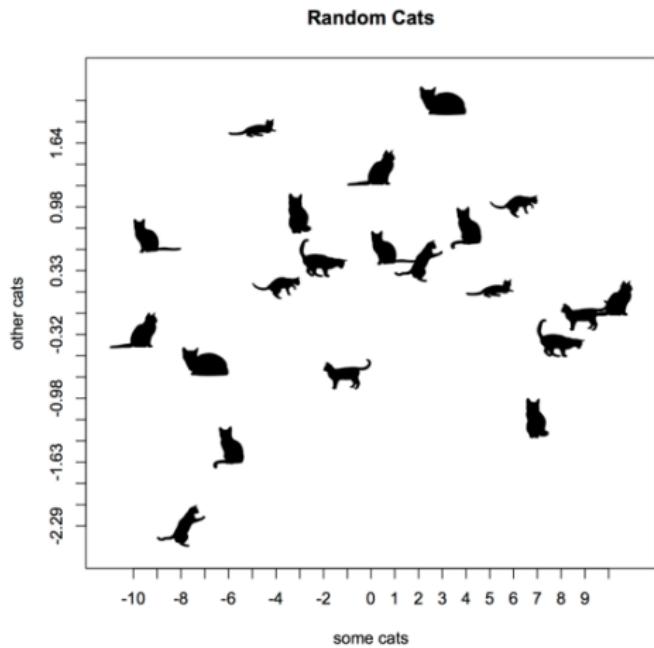
## A few of our favorite things

- ▶ Fun R do-dads

# CatterPlot for feline based graphics:

► <https://github.com/Gibbsdavidl/CatterPlots>

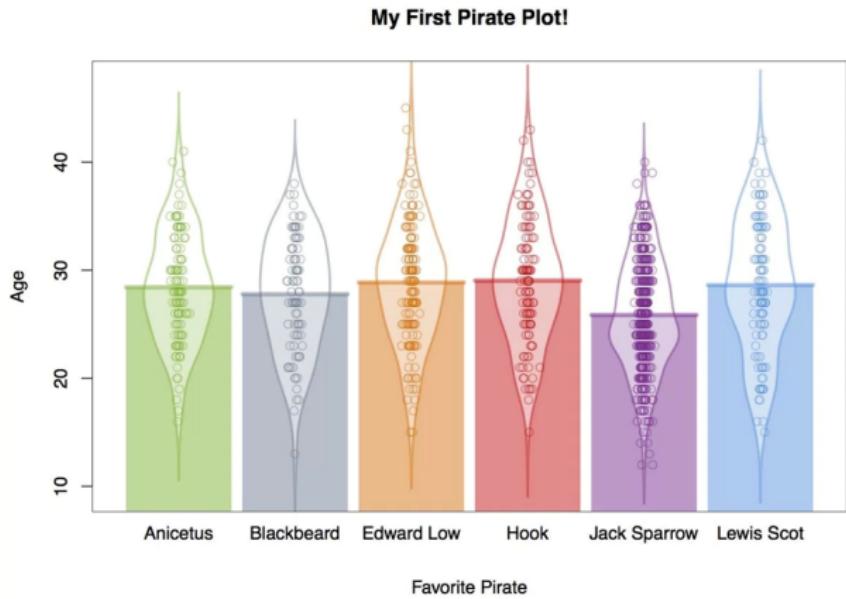
```
dev.tools::install_github(Gibbsdavidl/CatterPlots)
```



# What's a pirate's favorite programming language?

► <https://cran.r-project.org/web/packages/yarr/vignettes/pirateplot.html>

```
install.packages('yarr')
```



# Color palettes to fit your mood

► <https://github.com/karthik/wesanderson>

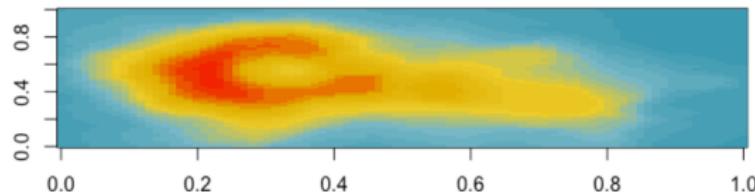
```
devtools::install_github(karthik/wesanderson)
```

The Life Aquatic with Steve Zissou (2004)

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

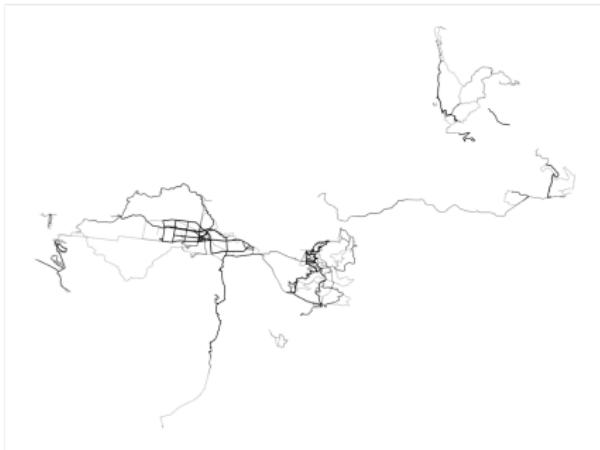
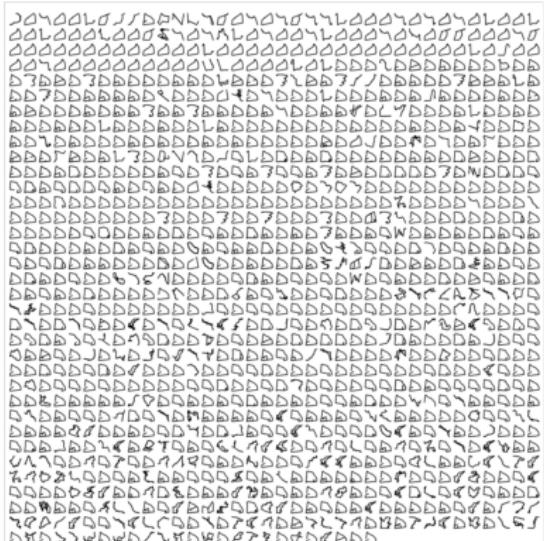


```
pal <- wes_palette("Zissou1", 21, type = "continuous")
image(volcano, col = pal)
```



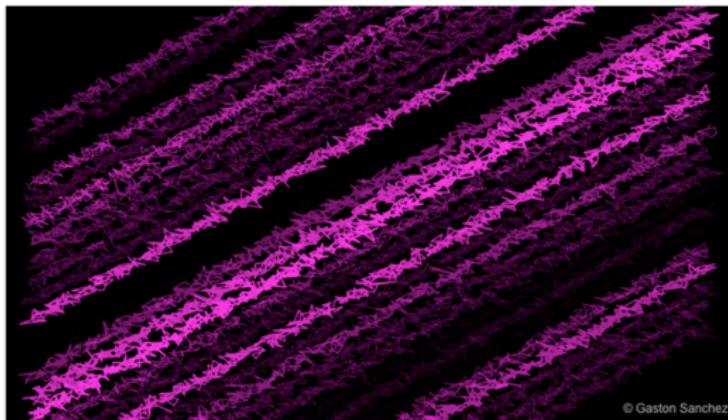
# Mapping your Strava routes

- ▶ <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 points x-y values
x <- seq(0, 100, length = 1000)
y <- x + rnorm(1000)

# -----
# Pink Barbs
# -----
# see graphical parameters
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, 0), pch = 19,
        lwd = rnorm(0.8), lty = 1, runif(1000),
        lwd = sample(seq(0.1, 2, length = 20), 1))
}
# signature
legend("bottomright", legend = "@ Gaston Sanchez", bty = "n",
       text.col = "gray77")
# reset par
par(op)
dev.off()
```