

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

Part 0: Background

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 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

R is a bit ugly

The screenshot shows the RGui (64-bit) interface. The R Console window displays the standard R startup message, including the version (R 3.3.1), copyright information, and a note about the "Bug in Your Hair". It also shows basic help text for running R, natural language support, and citation information. A command-line entry for 'citation()' is present. The script editor window shows a script titled 'C:\Users\jrieck\Documents\workshops\2019_Rstudio_Magic\R\0_create_ADNI_data_base.R'. The script contains R code for data manipulation, specifically merging datasets and modifying variable classes.

```
R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'licence()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> citation()

To cite R in publications use:

R Core Team (2016). R: A language and environment for statist
computing. R Foundation for Statistical Computing, Vienna, Au

## Load and clean data
#####
### 0.1 Specify the column names and participants you want (ie, baseline visit f
adni.cols <- c("RID", "VISCODE", "DX", "AGE", "PTGENDER", "PTEDUCAT", 'PTETHCAT'
adni.rows <- c(adnimerge$VISCODE=="D1" & adnimerge$MOCA>=16)
merge_subset <- adnimerge[adni.rows,adni.cols]

### remove participants with missing data
merge_subset <- merge_subset[complete.cases(merge_subset),]

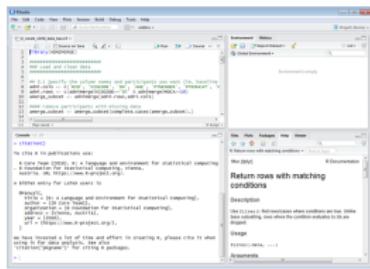
## 0.2 Bring in modified hachinksi
merge_subset$HMSCORE <- modhach$HMSCORE[match(merge_subset$RID, modhach$RID)]

## 0.3 Manually change variable classes (remove class 'labelled')
merge_subset$RID <- as.character(merge_subset$RID)
merge_subset$VISCODE <- as.character(merge_subset$VISCODE)
merge_subset$DX <- as.character(merge_subset$DX)
merge_subset$AGE <- as.numeric(merge_subset$AGE)
merge_subset$PTGENDER <- as.character(merge_subset$PTGENDER)
merge_subset$PTEDUCAT <- as.numeric(merge_subset$PTEDUCAT)
merge_subset$PTETHCAT <- as.character(merge_subset$PTETHCAT)
```

But R has many interfaces

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

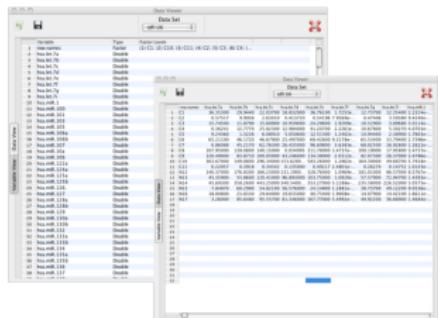
RStudio



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

Part 1: Rstudio

Part 1: RStudio

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

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 several windows open:

- Script Editor:** Displays the R script `create_ADNI_data.R`. The code performs the following steps:
 - Imports required packages: `tidyverse`, `lapply`, `data.table`, and `stringr`.
 - Creates a function `PTRACCAT` that takes a list of data frames and merges them into a single data frame.
 - Specifies column names for the merged data frame.
 - Loads and cleans data from `ADNI_merge` and `MOCA` datasets.
 - Specifies column names and participants for the baseline visit.
 - Removes participants with missing data.
 - Brings in modified hachimaki functions.
 - Manually changes variable classes (removing class `'labelled'`).
- Console:** Shows statistical summaries for various variables across different datasets (e.g., APOE4, FDG, AV45, CDRSB, ADAS13, MOCA, Hippocampus, Midtemp, nPACCtrailsB, HMSCORE). For example, the APOE4 dataset has the following summary statistics:

	Mean	3rd Qu.	Min.	Max.
APOE4	:71.92	:76.60	:69.60	:83.85
FDG	:16.36	:18.00	:20.00	:20.00
AV45				
CDRSB				
ADAS13				
MOCA				

- Environment:** Shows the global environment with objects like `merge_subset` (665 obs. of 17 variables), `variable_type_map`, `Values`, `ids`, and `MOCA`.
- Plots:** A scatter plot titled "FILEs, PLOTS, HELP" is shown in the plots pane.
- File Browser:** Shows the project structure under `workshops > 2019_Rstudio_Magic`. The `2019_Rstudio_Magic.Rproj` file is highlighted.

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 also handles missing data and manually changes variable classes.
- 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`, `variable_type_map`, `Values`, and `Functions`.
- File browser pane:** Shows the project structure with files like `README.md`, `Renviron`, and `2019_Rstudio_Magic.Rproj`.

```
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))
```

	APOE4	FDG	AV45	CDRSB	ADAS13	MOCA
Min.	:0.0000	Min. :0.6983	Min. :0.8385	Min. :0.0000	Min. :0.0	Min. :16.00
1st Qu.	:0.0000	1st Qu.:17.60	1st Qu.:17.60	1st Qu.:0.0000	1st Qu.:8.0	1st Qu.:22.00
Median	:0.0000	Median :28.02	Median :28.02	Median :0.0000	Median :12.00	Median :25.00
Mean	:0.5248	Mean :31.2682	Mean :31.2682	Mean :1.202	Mean :13.8	Mean :23.89
3rd Qu.	:1.0000	3rd Qu.:31.3620	3rd Qu.:31.3714	3rd Qu.:2.0000	3rd Qu.:18.0	3rd Qu.:26.00
Max.	:2.0000	Max. :31.7013	Max. :32.0256	Max. :5.500	Max. :46.0	Max. :30.00

> view(anmerge_subset)
> |

RStudio Environment

~\workshops\2019_RStudio_Magic - master - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

o 0_create_ADNI_data.RData 1_create_ADNI_data_tidyverse.R amerge_subset

DATA VIEWER

DX	AGE	PTGENDER	PTEDUCAT	PTRECAT	PTRACCAT	APOE4	FDG	AV45	CDRSB	ADAS13	MOCA	WholeBrain	
2002	MCI	64.0	Male	18	Not Hisp/Latino	White	0	1.2091938	0.9754323	2.5	4	28	1135568.8
2003	MCI	63.6	Female	18	Not Hisp/Latino	White	0	1.2889628	1.1645374	2.0	11	24	1070369.3
2007	MCI	85.4	Female	20	Hisp/Latino	White	0	1.305182	1.4495259	2.5	9	23	920710.1
2010	MCI	62.9	Female	20	Not Hisp/Latino	Other	1	1.3121151	1.1472844	0.5	6	27	966402.9
2011	MCI	69.9	Female	14	Not Hisp/Latino	White	0	1.4571991	1.0579399	1.5	7	25	987825.3
2018	MCI	76.4	Female	18	Not Hisp/Latino	White	0	1.3148491	1.052191	1.5	10	26	1004817.0
2022	MCI	66.0	Male	18	Not Hisp/Latino	Other	1	1.2031270	1.3135914	1.5	6	25	1173068.2
2027	MCI	61.9	Female	14	Not Hisp/Latino	White	0	1.4000448	1.0297671	1.0	6	24	969957.1
2031	MCI	72.5	Male	16	Not Hisp/Latino	White	0	1.3404430	0.9939887	2.0	10	24	1059879.5
2036	MCI	66.7	Female	14	Not Hisp/Latino	White	0	1.2959310	1.0307979	1.0	5	30	1019101.0
2037	MCI	75.8	Male	16	Not Hisp/Latino	White	1	1.3074956	1.4389912	0.5	20	20	1104797.3
2042	MCI	68.5	Male	20	Not Hisp/Latino	White	0	1.2081130	1.0555841	1.5	18	23	1061388.8
2043	MCI	72.2	Female	20	Not Hisp/Latino	White	1	1.3761158	1.2040191	2.0	8	27	1039110.3

Showing 10 of 15 1665 entries

Console Terminal Jobs

~\workshops\2019_RStudio_Magic

```
Mean : 71.92 Mean : 16.36
3rd Qu.: 76.60 3rd Qu.: 18.00
Max. : 89.60 Max. : 20.00

APOE4 FDG AV45 CDRSB ADAS13 MOCA
Min. : 0.0000 Min. : 0.6983 Min. : 0.8385 Min. : 0.0000 Min. : 0.0 Min. : 16.00
1st Qu.: 0.0000 1st Qu.: 1.0000 1st Qu.: 1.1000 1st Qu.: 0.0000 1st Qu.: 8.0 1st Qu.: 22.00
Median : 0.0000 Median : 1.2802 Median : 1.1105 Median : 0.0000 Median : 10.0 Median : 25.00
Mean : 0.5248 Mean : 1.2682 Mean : 1.1989 Mean : 0.1200 Mean : 13.8 Mean : 23.89
3rd Qu.: 1.0000 3rd Qu.: 1.3620 3rd Qu.: 1.3714 3rd Qu.: 2.0000 3rd Qu.: 18.0 3rd Qu.: 26.00
Max. : 2.0000 Max. : 1.7012 Max. : 2.0256 Max. : 15.5000 Max. : 46.0 Max. : 30.00

WholeBrain Hippocampus MidTemp nPACCtrailsB HMSCore
Min. : 114.421 Min. : 1.011 Min. : 12213 Min. : -18.6883 Min. : 0.0000
1st Qu.: 984410 1st Qu.: 6510 1st Qu.: 2535 1st Qu.: -1.051 1st Qu.: 0.0000
Median : 1051621 Median : 7223 Median : 20186 Median : -2.5250 Median : 1.0000
Mean : 1057026 Mean : 7150 Mean : 20302 Mean : -3.6882 Mean : 0.5888
3rd Qu.: 1120570 3rd Qu.: 7834 3rd Qu.: 22088 3rd Qu.: -0.3482 3rd Qu.: 1.0000
Max. : 1486036 Max. : 10602 Max. : 32189 Max. : 5.3540 Max. : 3.0000
> view(amerge_subset)
> |
```

Environment History Connections Git

Global Environment

anmerge_subset 665 obs. of 17 variables
variable_type_map num [1:17]: 0 1 0 0 0 0 0 0 1 0 ...
ids chr [1:665]: "2002" "2003" "2007" "2010" "2011" "2012" ...
MOCAs num [1:665]: 28 24 23 27 25 26 25 24 24 30 ...
Functions scatterplotter function (x, y, x.lim = NA, y.lim = NA, x.lab = "...") {

Files Plots Packages Help Viewer

Home workshops : 2019_RStudio_Magic

Name	Size	Modified
Renviron	52 B	May 12, 2019, 11:33 AM
2019_RStudio_Magic.Rproj	210 B	May 12, 2019, 6:30 PM
external		
mice		
output		
R		
README.md	42 B	May 12, 2019, 11:29 AM
Rmd		

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

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 small inset showing a ZIP code explorer application.

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

Online Learning - RStudio

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

R Studio

Products Resources Pricing About Us Blogs

Online learning

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
- Shiny
- R Markdown
- Data Science
- Books

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 SEARCH

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 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

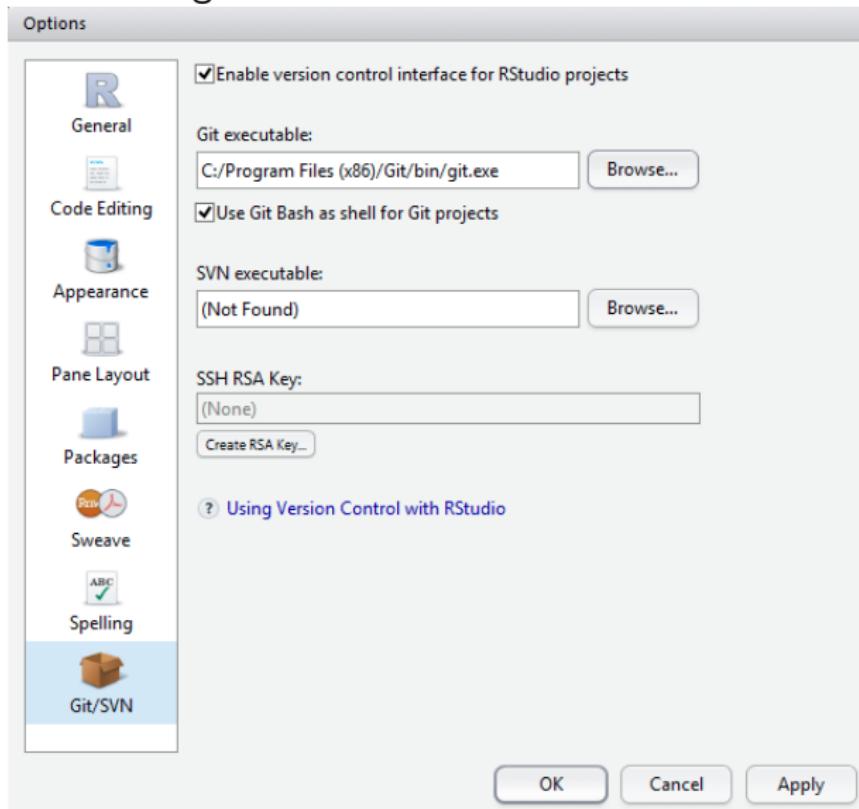
- ▶ Download R and Rstudio
- ▶ Add-on packages

```
#to install from CRAN
install.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)
```

- ▶ 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

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 >

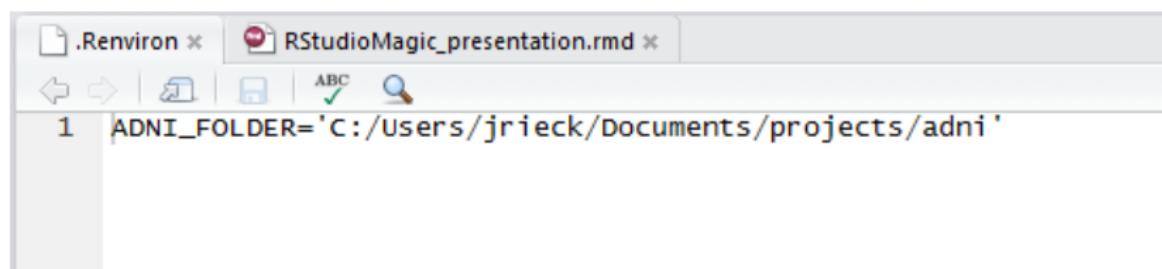
Cancel

Format .gitignore

- ▶ File types to ignore:
 - ▶ `.Rproj.user`
 - ▶ `.Rhistory`
 - ▶ `.Ruserdata`
 - ▶ `.Renviron`
 - ▶ `.rda` & `.Rdata` (to avoid pushing potentially sensitive data files to git)
 - ▶ `**` before each extention 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 `.Renvironment` file and define variables



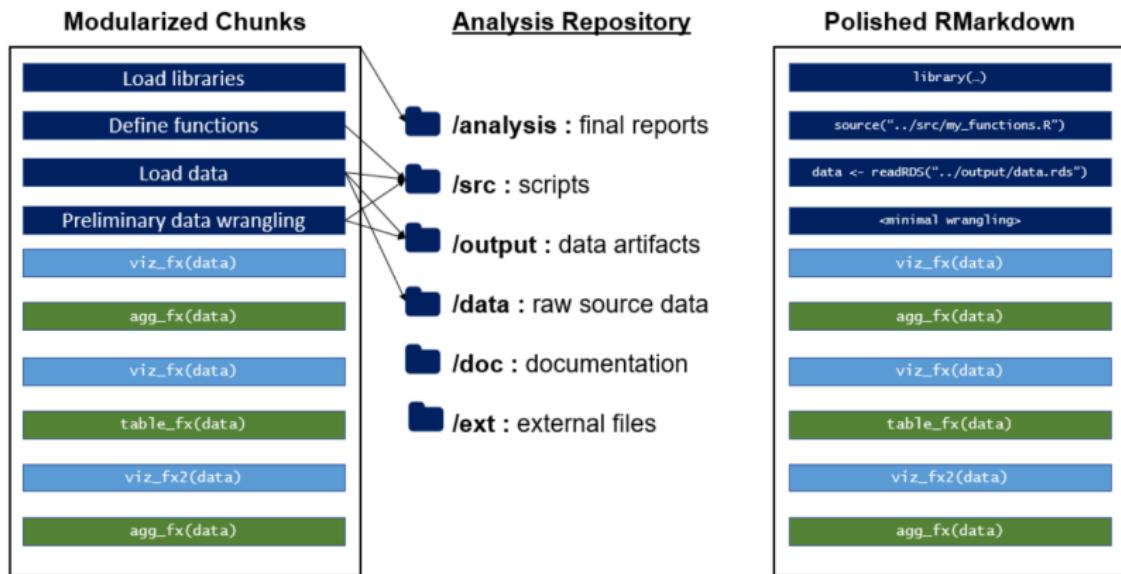
The screenshot shows the RStudio interface with two tabs visible: ".Renvironment" and "RStudioMagic_presentation.rmd". The ".Renvironment" tab is active, displaying the following code:

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

The code defines a variable `ADNI_FOLDER` with the value `'C:/Users/jrieck/Documents/projects/adni'`.

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

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 (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

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

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 5: Advanced R

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

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

Part 6: Extras

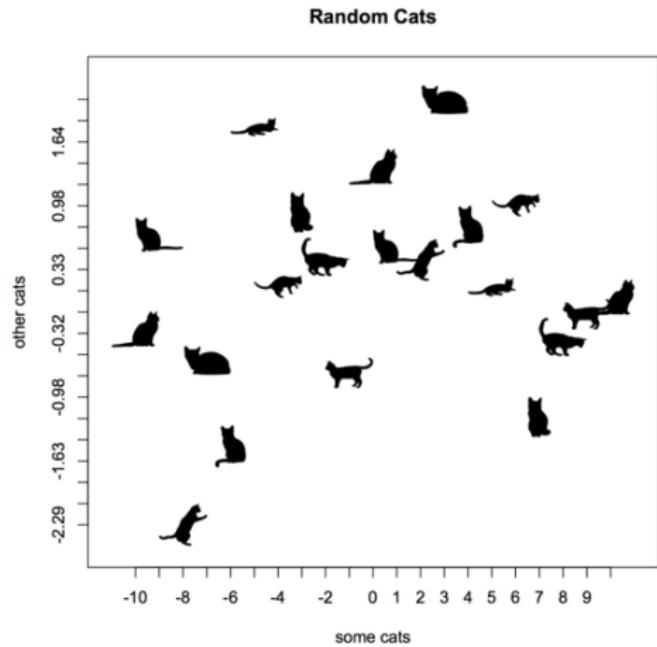
Part 6: A few of our favorite things

- ▶ Fun R do-dads

CatterPlot for feline based graphics:

- ▶ <https://github.com/Gibbsdavidl/CatterPlots>

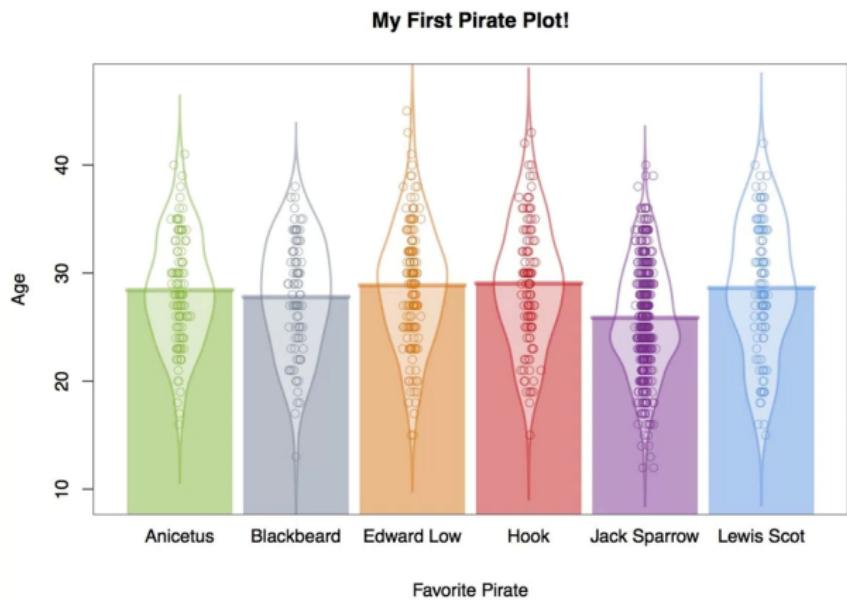
```
dev.tools::install_github(Gibbsdavidl/CatterPlots)
```



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>

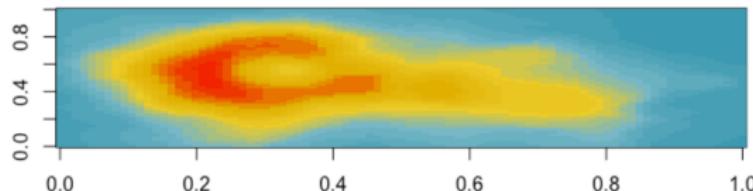
```
devtools::install_github(karthik/wesanderson)
```

The Life Aquatic with Steve Zissou (2004)

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

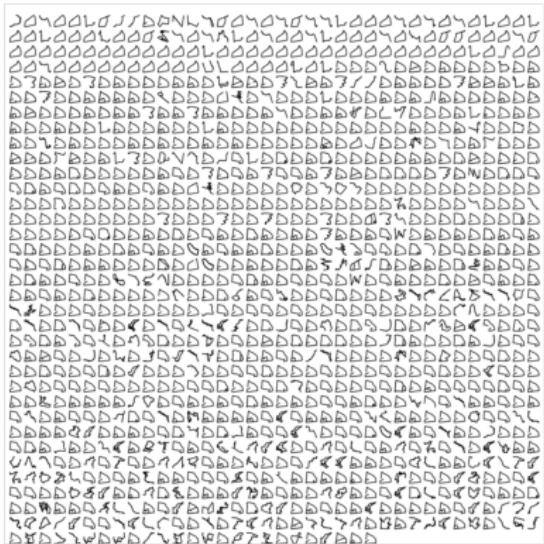


```
pal <- wes_palette("Zissou1", 21, type = "continuous")
image(volcano, col = pal)
```



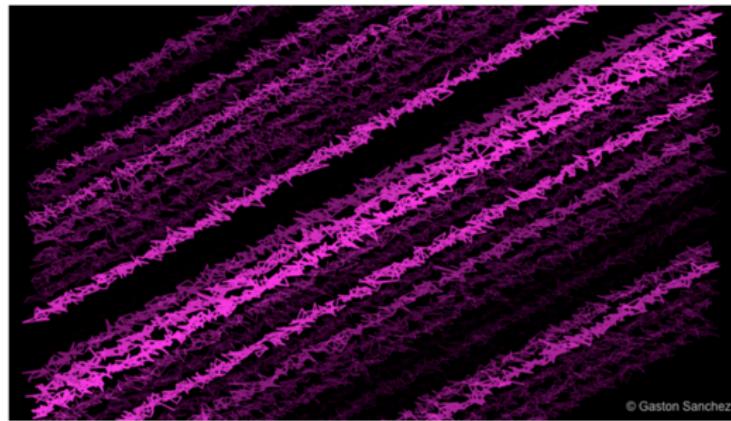
Mapping your Strava routes

- ▶ <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 pairs of x-y values
x <- seq(1, 100, length = 1000)
y <- x + rnorm(1000)

# -----
# Pink Barbs
# -----
# set graphical parameters
op <- par(bg = "black", mar = rep(0, 4))
plot(x, y, type = "n")
for (i in seq(-80, 70, 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()
```