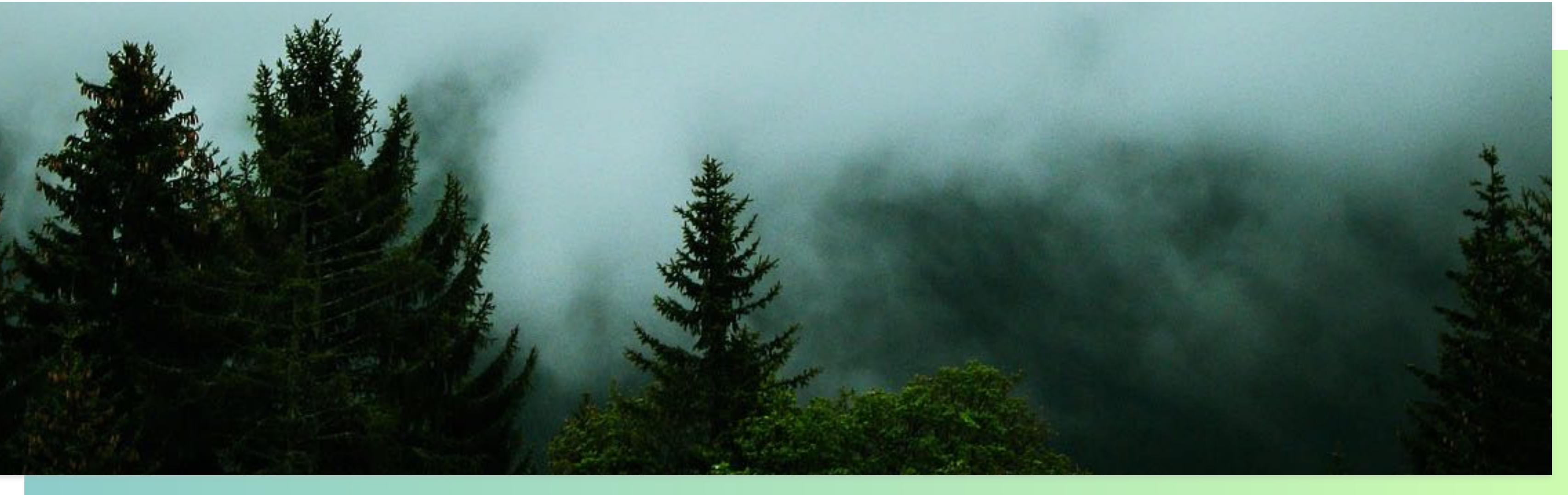


# PILGRIM'S PROGRESS

a journey from confusion to contribution





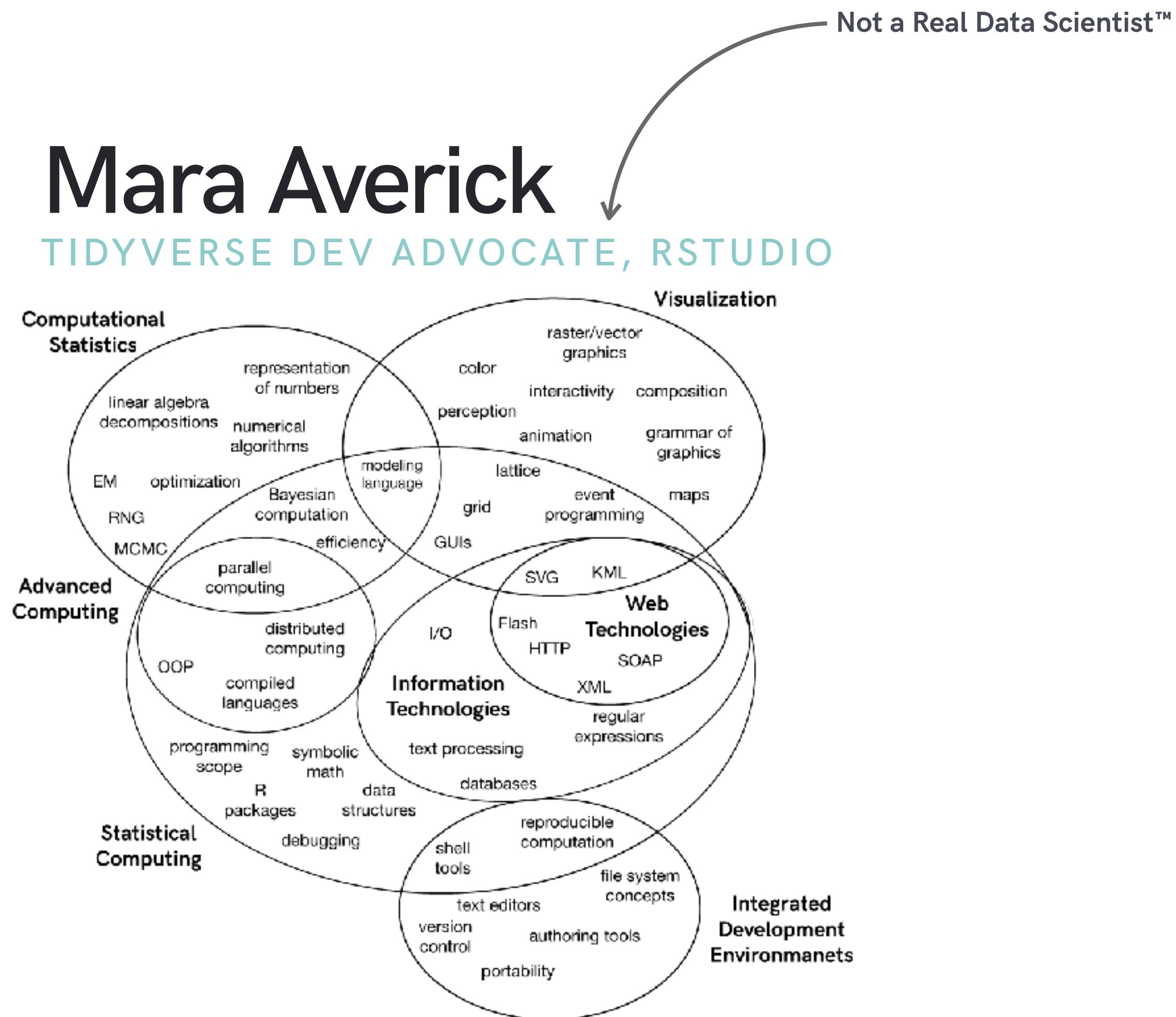
**Mara Averick**  
TIDYVERSE DEV ADVOCATE, RSTUDIO



**Mara Averick**  
TIDYVERSE DEV ADVOCATE, RSTUDIO

Not a Real Data Scientist™







# Mara Averick

TIDYVERSE DEV ADVOCATE, RSTUDIO



Less true these days!

# An aside on the title

*“Like many social groups that do not reproduce themselves biologically, the experimental particle physics community renews itself by training novices.”*

— Sharon Traweek, *Pilgrim's Progress: Male Tales Told During a Life in Physics*



# SCIENCE & SOCIETY

# An aside on the title



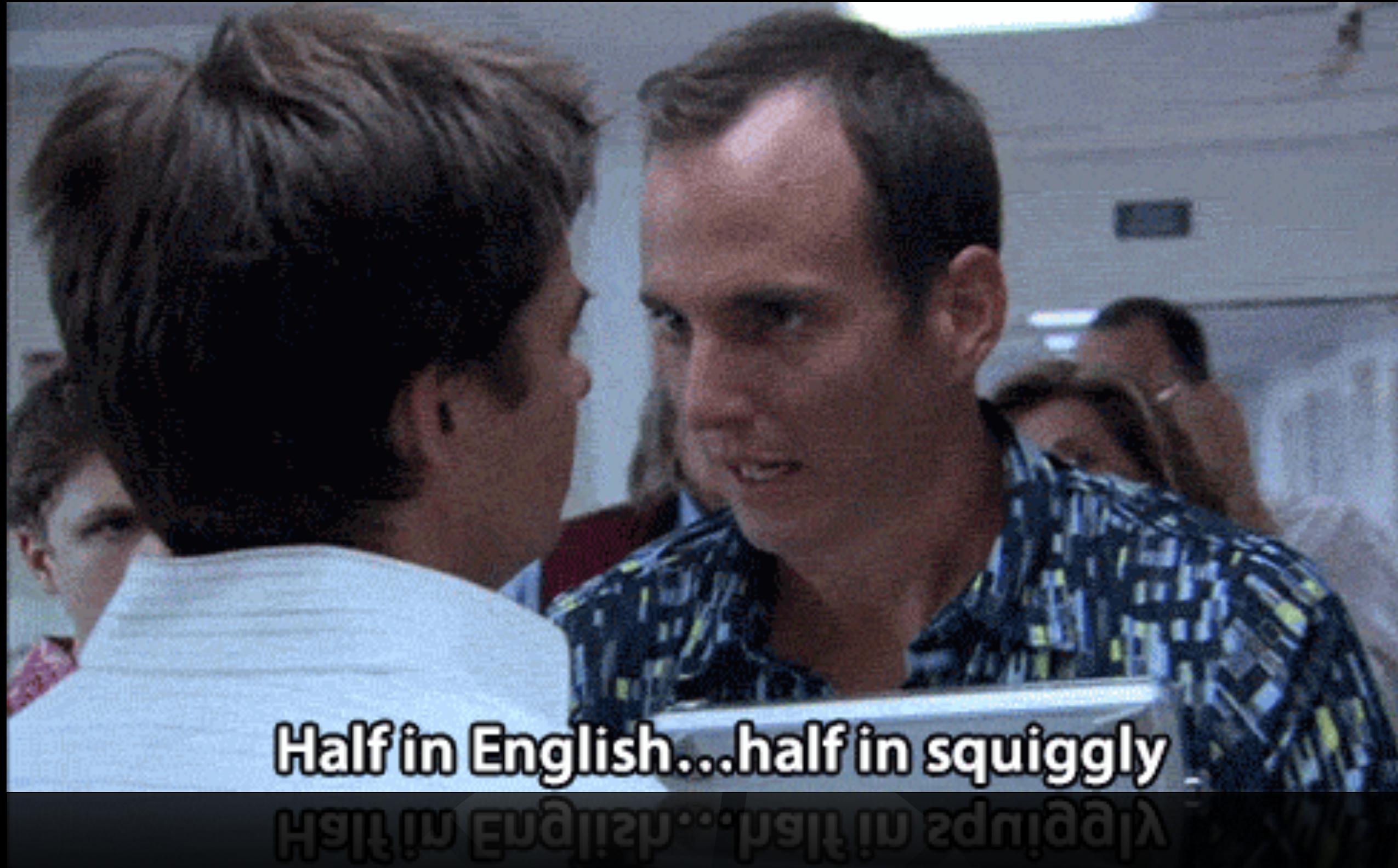
# An aside on the title



# my jouRney...



# my jouRney...



Half in English...half in squiggly  
Half in English...half in squiggly

# my jouRney...



OMG I just learned a thing!!

100% selfish

# but...

Hadley Wickham @hadleywickham

If you want to make a deep non-technical contribution to the field of data science, I can not think of a better role model than [@dataandme](#)

7:25 AM - 26 Jul 2017

21 Retweets 169 Likes

6 21 169

# but...

Edwin Thoen  
@edwin\_thoen

Following

After 9 months as an R blogger I discovered the key to writing a succesfull blog: please [@dataandme!](#)

What determines a succesful blog post?

The figure is a histogram titled "What determines a successful blog post?". It compares two distributions on a horizontal axis labeled "Likes, retweets, pageviews, whatever" ranging from 0 to 500. A red curve represents posts "Tweeted about by Mara", which have a low, sharp peak centered around 20 likes. A blue curve represents posts "Not tweeted about by Mara", which have a higher, broader peak centered around 400 likes. Two horizontal lines extend from the peaks of each curve to the axis.

Tweeted about by Mara -

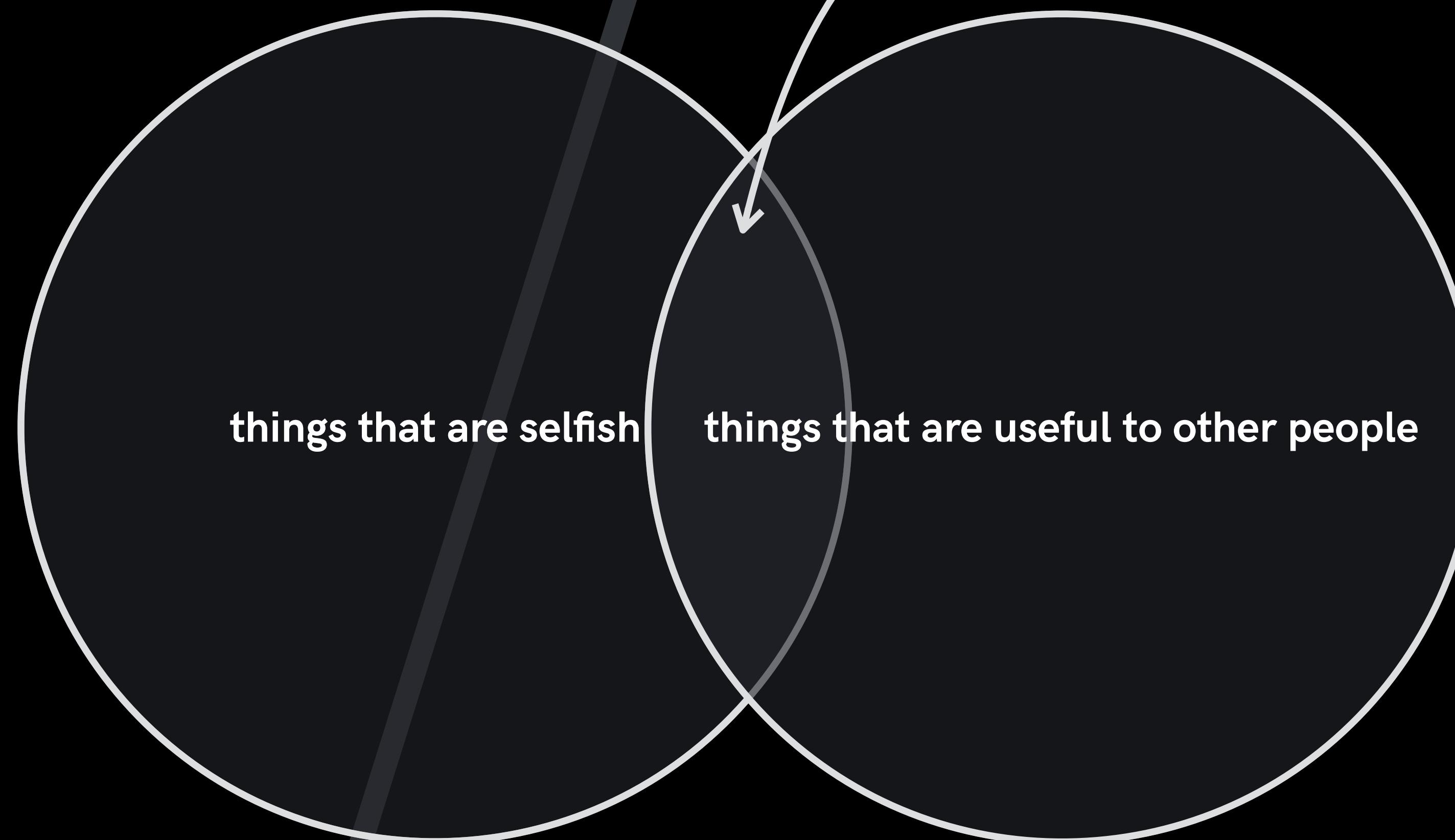
Not tweeted about by Mara -

0 100 200 300 400 500

Likes, retweets, pageviews, whatever

5:36 AM - 27 Aug 2017

**but...**





Hadley Wickham @hadleywickham

Follow



Replying to [@juliasilge](#) [@arnicas](#) [@dataandme](#)

I really didn't want to imply that Mara doesn't have awesome technical skills 😞 She does!

8:32 AM - 26 Jul 2017

8:35 AM - 26 Jul 2017



Hadley Wickham   
@hadleywickham

Follow

▼

Replying to @juliasilge @arnicas @dataandme

I really didn't want to imply that Mara doesn't have awesome technical skills 😞 She does!

8:32 AM - 26 Jul 2017

8:35 AM - 26 Jul 2017



Lucy   
@LucyStats

Follow

▼

Replying to @hadleywickham @juliasilge and 2 others

I read as @dataandme has the ability to translate quite technical material to a non-tech audience, so if you're non-tech she's a 🔥 go to! 🎉🏆

8:37 AM - 26 Jul 2017

# ex•o•ter•ic

*adj. understandable by outsiders*

*or the general public*



Lucy   
@LucyStats

Follow ▾

Replying to [@hadleywickham](#) [@juliasilge](#) and 2 others

I read as [@dataandme](#) has the ability to translate quite technical material to a non-tech audience, so if you're non-tech she's a 🔥 go to! 🚀🏆

8:37 AM - 26 Jul 2017

# you never know...

 **Hadley Wickham**  @hadleywickham · 18 Nov 2017  
If you're using databases with shiny, make sure to check out the pool 🎯:  
[blog.rstudio.com/2017/11/17/poo...](http://blog.rstudio.com/2017/11/17/poo...) Will make your life much easier! #rstats

7 120 395

 **Mara Averick** @dataandme · 18 Nov 2017  
Never forget to check the pool...



2 19

# you never know...



**Matthew Dickinson**

@midickinson

Follow

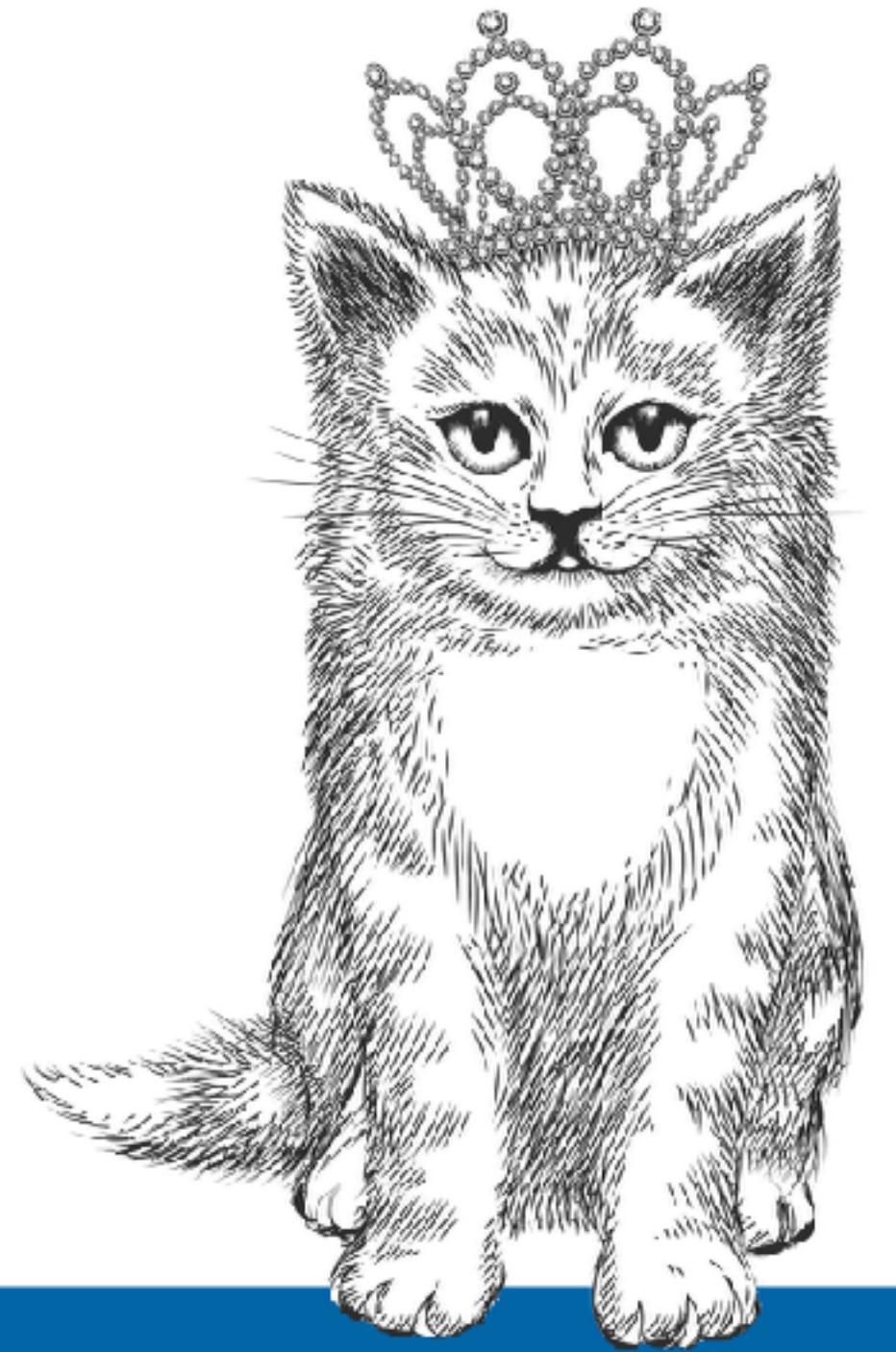


Replies to [@dataandme](#) [@hadleywickham](#)

If it weren't for the gif I would have totally just skipped over knowing about this package that looks to be a huge benefit to me.  
Thanks!

6:13 PM - 18 Nov 2017

*Your taste, experiences, and objectives are the absolute truth*



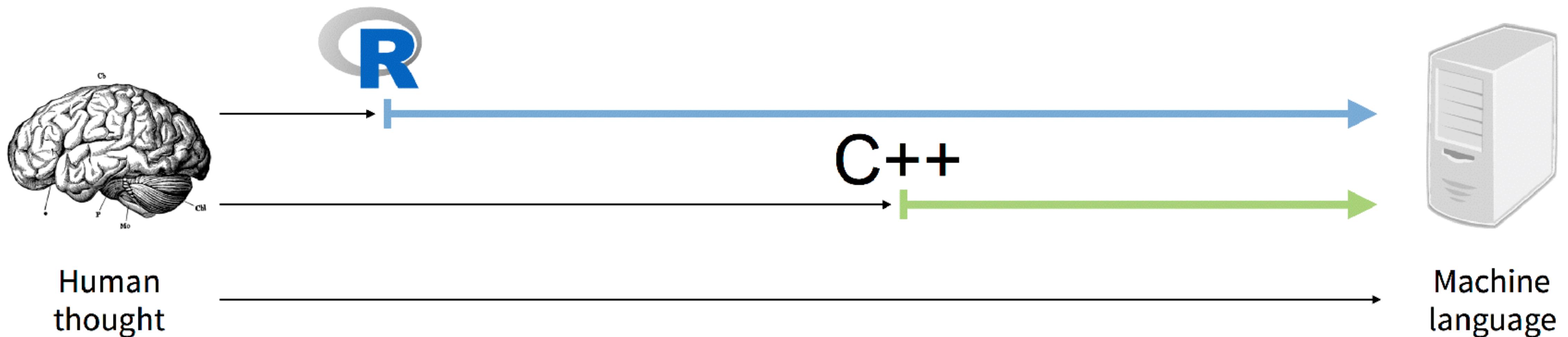
*Expert*

Hating on Languages  
You Don't Use

O RLY?

@ThePracticalDev

# R - a computer language for scientists





# What about this so-called tidyverse?

The tidyverse is an opinionated collection of R packages designed for data science.

All packages share an underlying design philosophy, grammar, and data structures.

# TIDY TOOLS

Functions  
should be...

## SIMPLE

Do one thing and do it well.

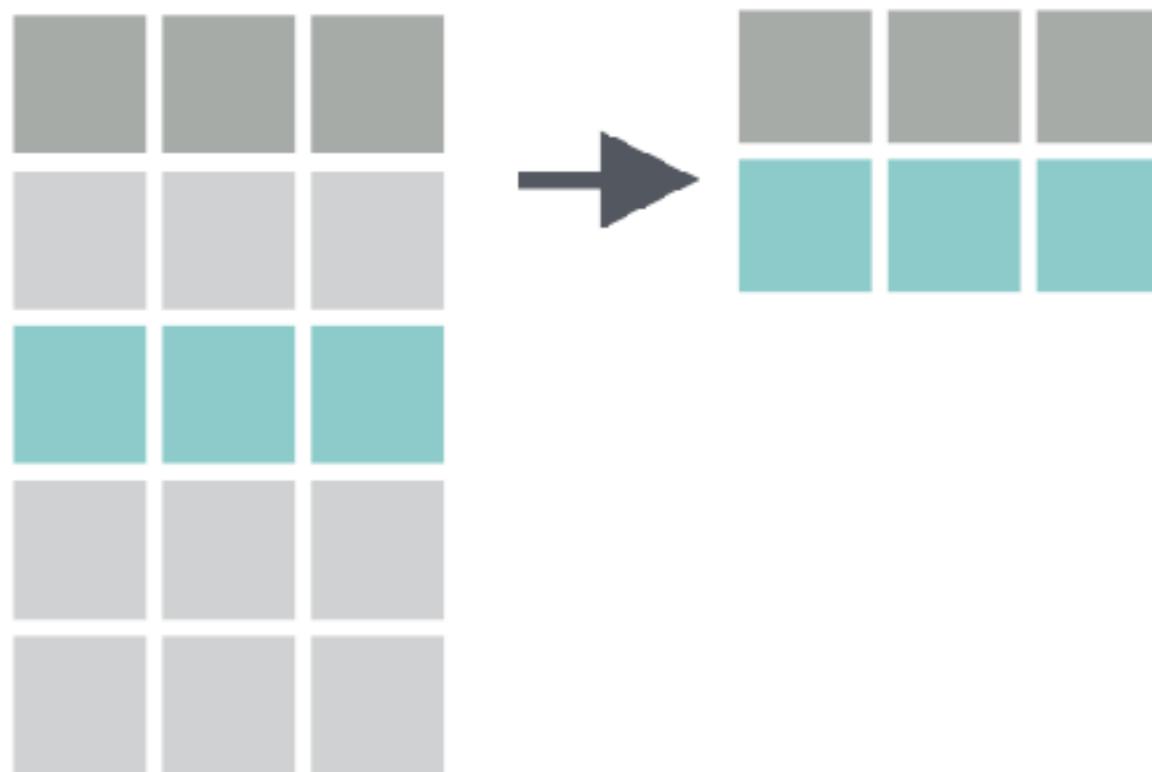
## COMPOSABLE

Combine with other functions for multi-step operations.

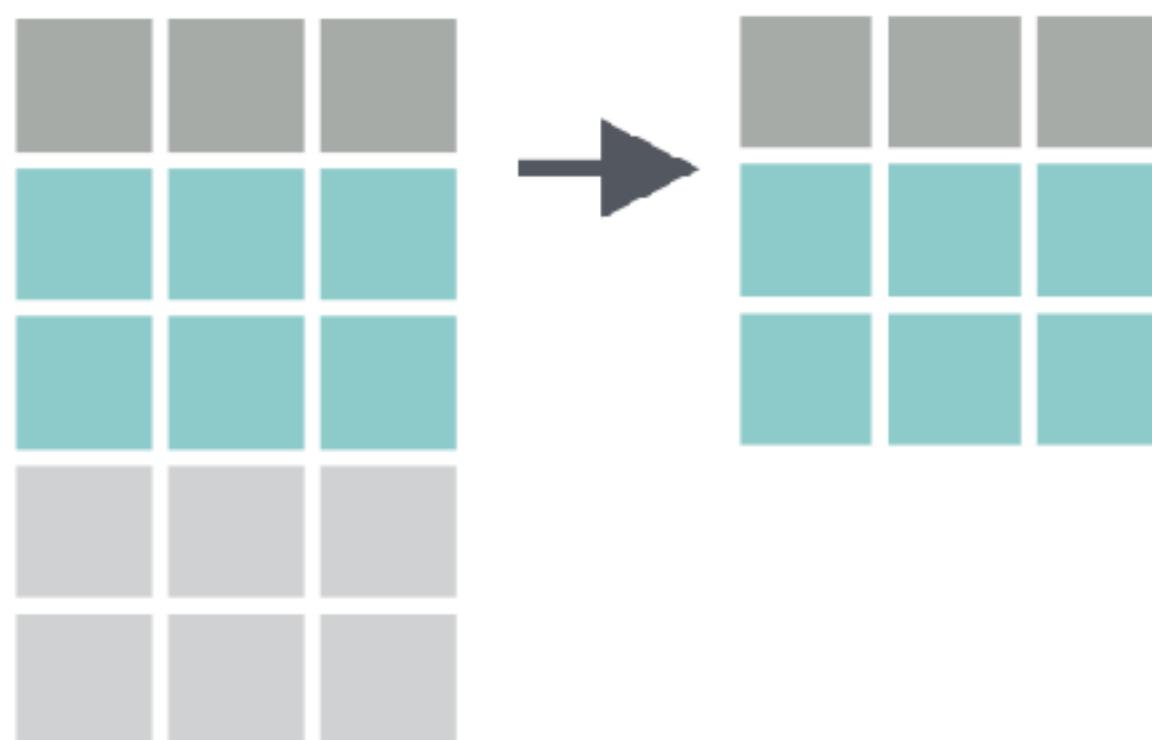
## DESIGNED FOR HUMANS

Use evocative verb names, making them easy to remember.

# FUNCTION EXAMPLES



**filter(.data, ...)** Extract rows that meet logical criteria. Also **filter\_()**. `filter(iris, Sepal.Length > 7)`



**top\_n(x, n, wt)** Select and order top n entries (by group if grouped data).  
`top_n(iris, 5, Sepal.Width)`

# COMPOSE WITH THE PIPE

```
iris %>%
```

```
  filter(Sepal.Length > 7) %>%
```

```
  top_n(5, Sepal.Width)
```

# COMPOSE WITH THE PIPE



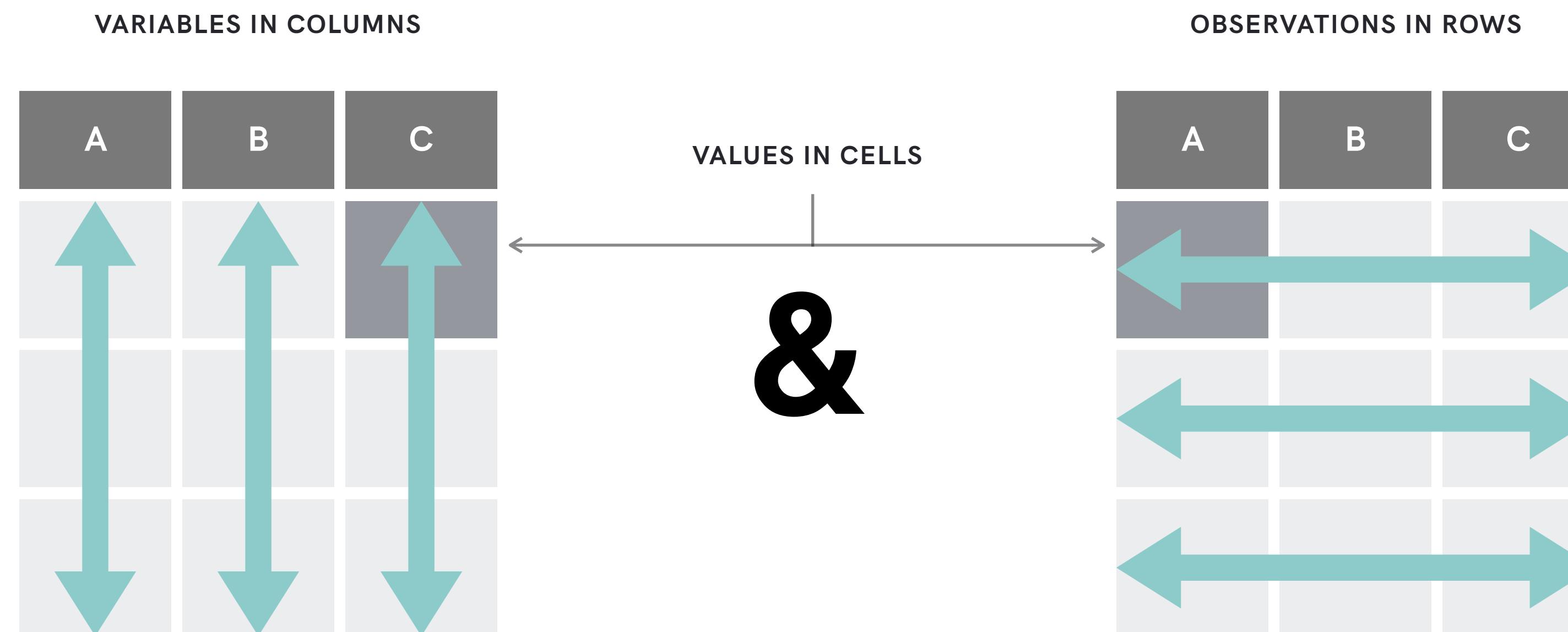
# COMPOSE WITH THE PIPE

```
iris %>%
```

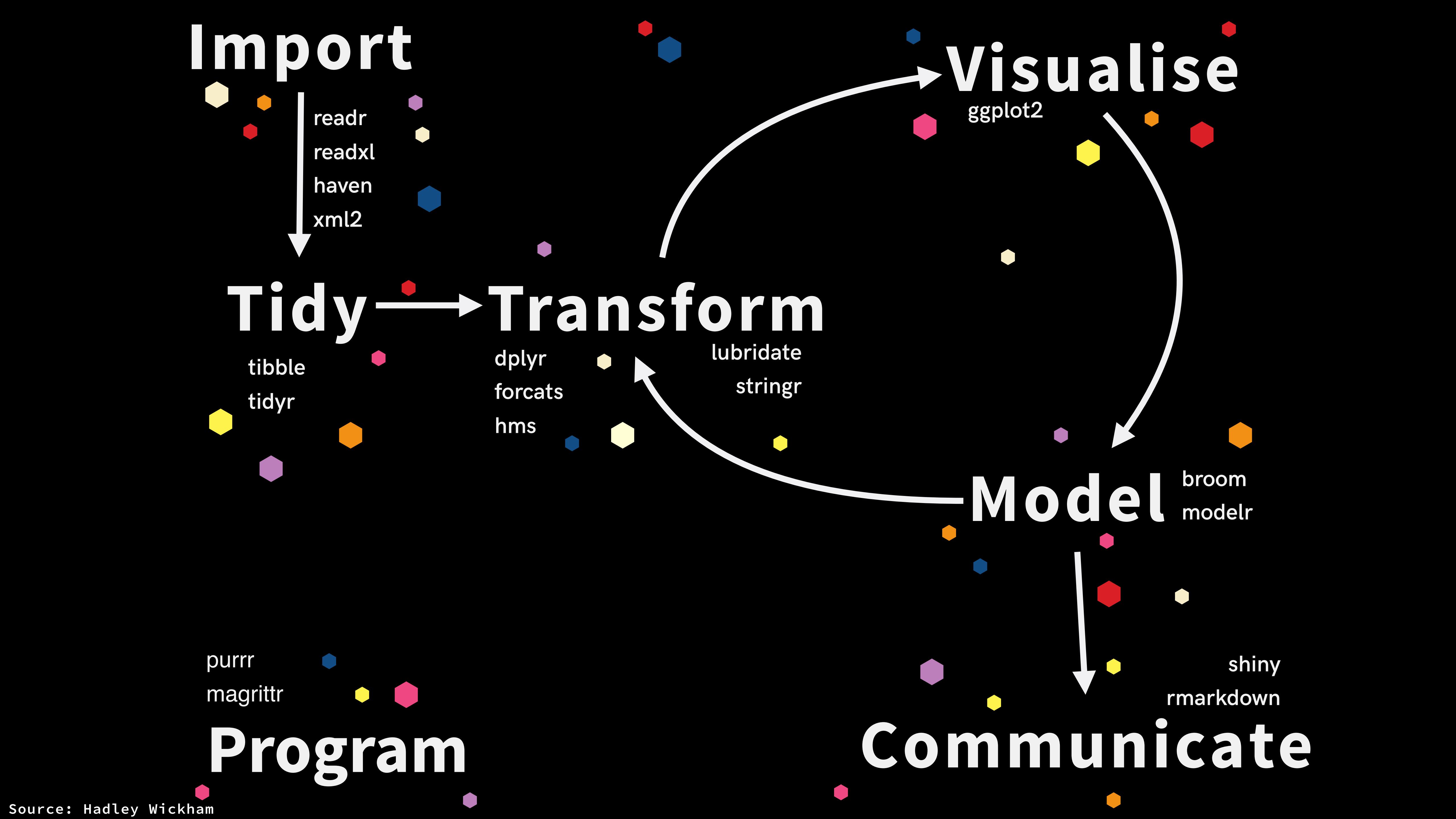
```
  filter(Sepal.Length > 7) %>%
```

```
  top_n(5, Sepal.Width)
```

# TIDY DATA



Source: Wickham, Hadley. 2014. "Tidy Data." *Journal of Statistical Software* 59 (10): 1-23. doi:<http://dx.doi.org/10.18637/jss.v059.i10>.



# install.packages("tidyverse")



The screenshot shows an RStudio interface with a dark theme. The top bar has tabs for 'Console' (which is selected) and 'Terminal'. The path in the top left is '~ddtx\_demo/'. The main area displays the R command `install.packages("tidyverse")` followed by its output. The output indicates that the package is being installed into the directory `/Users/maraaverick/ddtx\_demo/packrat/lib/x86\_64-apple-darwin15.6.0/3.4.3` (as 'lib' is unspecified). It also lists the dependencies being installed, which include a large number of packages from the tidyverse ecosystem.

```
> install.packages("tidyverse")
Installing package into ‘/Users/maraaverick/ddtx_demo/packrat/lib/x86_64-apple-darwin15.6.0/3.4.3’
(as ‘lib’ is unspecified)
also installing the dependencies ‘colorspace’, ‘backports’, ‘mnormt’, ‘bindr’, ‘RColorBrewer’, ‘dichromat’,
‘munsell’, ‘labeling’, ‘viridisLite’, ‘rematch’, ‘evaluate’, ‘highr’, ‘markdown’, ‘yaml’, ‘htmltools’,
‘base64enc’, ‘rprojroot’, ‘utf8’, ‘plyr’, ‘psych’, ‘reshape2’, ‘assertthat’, ‘bindrcpp’, ‘glue’, ‘pkgconf’,
‘R6’, ‘Rcpp’, ‘DBI’, ‘digest’, ‘gtable’, ‘scales’, ‘lazyeval’, ‘mime’, ‘curl’, ‘openssl’, ‘cellrange’,
‘callr’, ‘clipr’, ‘knitr’, ‘rmarkdown’, ‘whisker’, ‘selectr’, ‘stringi’, ‘pillar’, ‘tidyselect’, ‘broom’,
‘cli’, ‘crayon’, ‘dplyr’, ‘dbplyr’, ‘forcats’, ‘ggplot2’, ‘haven’, ‘hms’, ‘httr’, ‘jsonlite’, ‘lubridate’,
‘magrittr’, ‘modelr’, ‘purrr’, ‘readr’, ‘readxl’, ‘reprex’, ‘rlang’, ‘rstudioapi’, ‘rvest’, ‘stringr’,
‘tibble’, ‘tidy়’, ‘xml2’
```

# library(tidyverse)

The screenshot shows the RStudio interface with the 'Console' tab selected. The command `> library(tidyverse)` is entered, followed by the output of the tidyverse package loading process.

```
> library(tidyverse)
— Attaching packages — tidyverse 1.2.1 —
✓ ggplot2 2.2.1.9000    ✓ purrr   0.2.4
✓ tibble  1.4.1.9000    ✓ dplyr   0.7.4.9000
✓ tidyr   0.7.2         ✓ stringr 1.2.0
✓ readr   1.2.0         ✓forcats 0.2.0
— Conflicts — tidyverse_conflicts() —
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()   masks stats::lag()
```

# TIDYVERSE PACKAGES

THE CORE



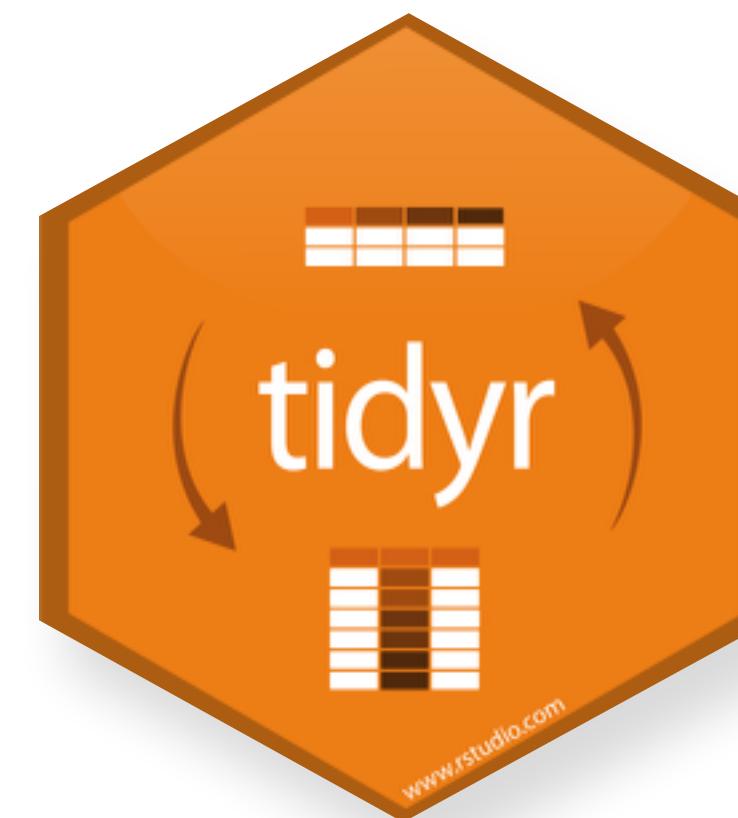
TIDYVERSE

a wrapper package that makes it easy  
to install and load core packages from  
the tidyverse in a single command



READR

a fast and friendly way to read in and  
parse rectangular data (like csv, tsv,  
and fwf)



TIDYR

a set of verbs that help you get to tidy  
data, allowing you to work with other  
tidyverse packages and store results

# TIDYVERSE PACKAGES

THE CORE



**TIBBLE**

a modern reimagining of data.frames  
that do less and complain more forcing  
you to confront problems earlier



**DPLYR**

a grammar of data manipulation with  
a set of verbs to solve common data  
wrangling problems



**GGPLOT2**

a system for declaratively creating  
graphics, based on The Grammar of  
Graphics

# TIDYVERSE PACKAGES

THE CORE



STRINGR

a cohesive set of functions designed to make working with strings as easy as possible



FORCATS

a suite of useful tools that solve common problems with factors, which R uses to handle categorical variables



PURRR

a consistent toolkit for enhancing R's functional programming, and working with functions and vectors

# TIDYVERSE PACKAGES

SOME NON-CORE



MAGRITTR

offers a set of operators (e.g. `%>%`)  
which make code more readable by  
structuring sequences of operations



READXL

makes it easy to get data out of Excel  
and into R, and work with tabular data  
in R



LUBRIDATE

provides robust methods for working  
with date-times in R, and functionality  
not offered in base R

# TIDYVERSE PACKAGES

SOME MORE NON-CORE



**HMS**

provides a simple class for storing durations or time-of-day values



**BROOM**

takes untidy model outputs of predictions and estimations to the tidy data we want to work with



**HAVEN**

enables R to read and write various data formats used by other statistical packages

# TIDYVERSE PACKAGES

SOME MORE NON-CORE



GOOGLEDRAVE

allows you to interact with files on  
Google Drive from R



RMarkdown

an authoring framework for data  
science that allows you to combine  
prose, code, and output



SHINY

makes it easy to build interactive web  
apps straight from R

CONTRIBUTING TO THE

tidyverse

# Contributing to FOSS



# Contributing to FOSS

**WHAT HOLDS PEOPLE BACK?**



# Contributing to FOSS

## WHAT HOLDS PEOPLE BACK?

- “*I can't write code.*”



# Contributing to FOSS

## WHAT HOLDS PEOPLE BACK?

- “*I can't write code.*”
- “*I'm not really good at this.*”



# Contributing to FOSS

## WHAT HOLDS PEOPLE BACK?

- “*I can't write code.*”
- “*I'm not really good at this.*”
- “*I'd just be a burden.*”



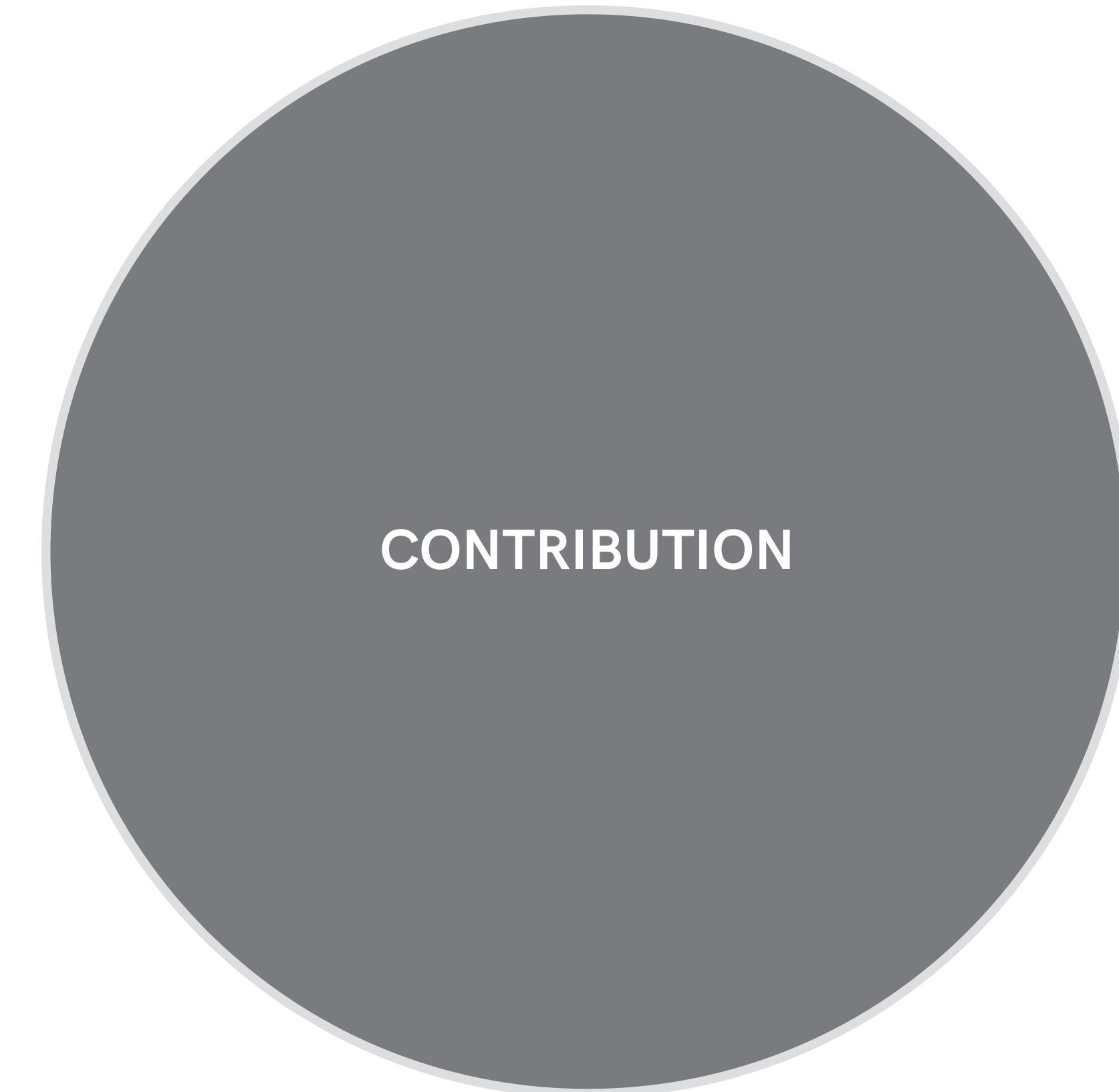
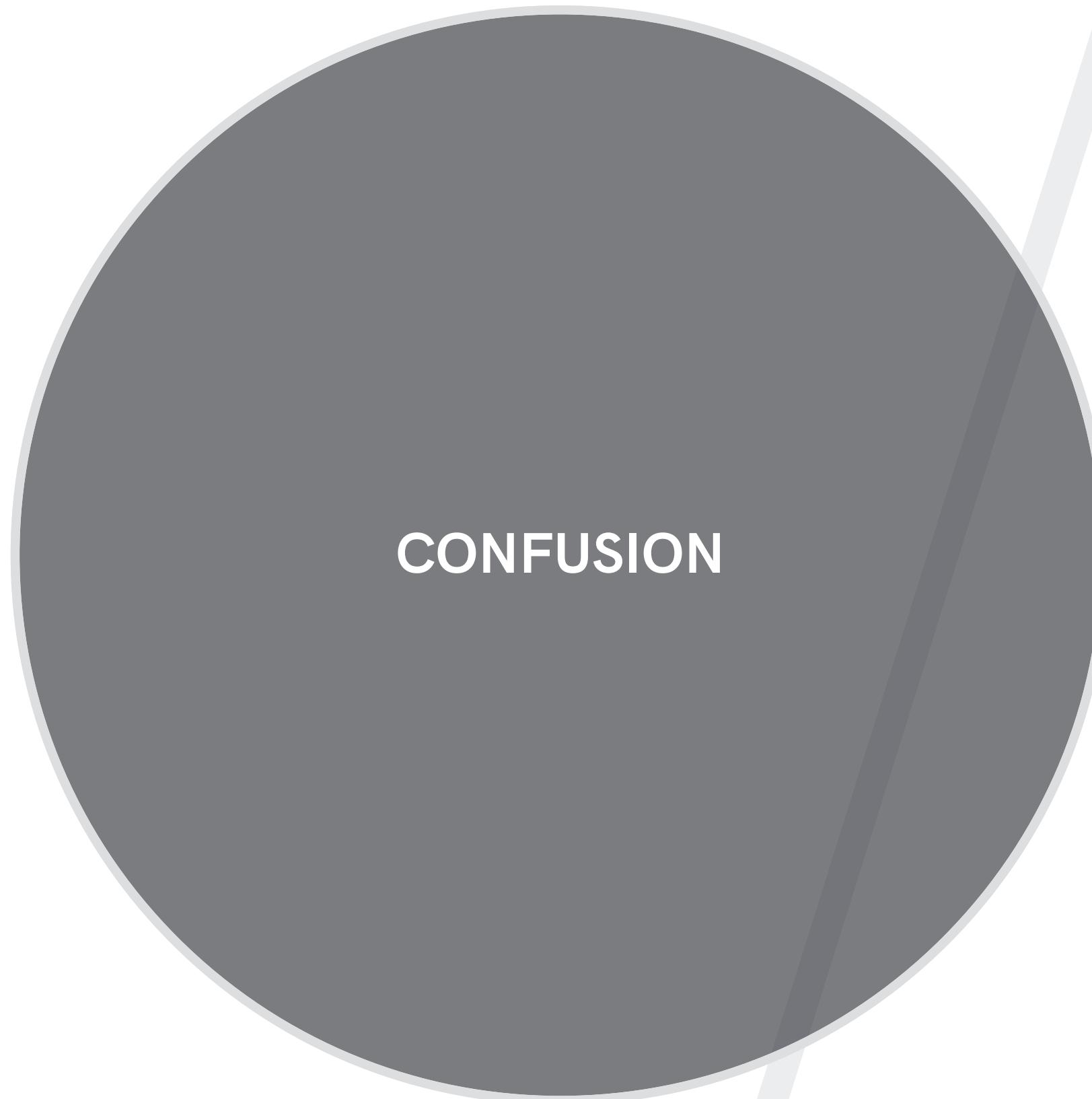
# Contributing to FOSS

## WHAT HOLDS PEOPLE BACK?

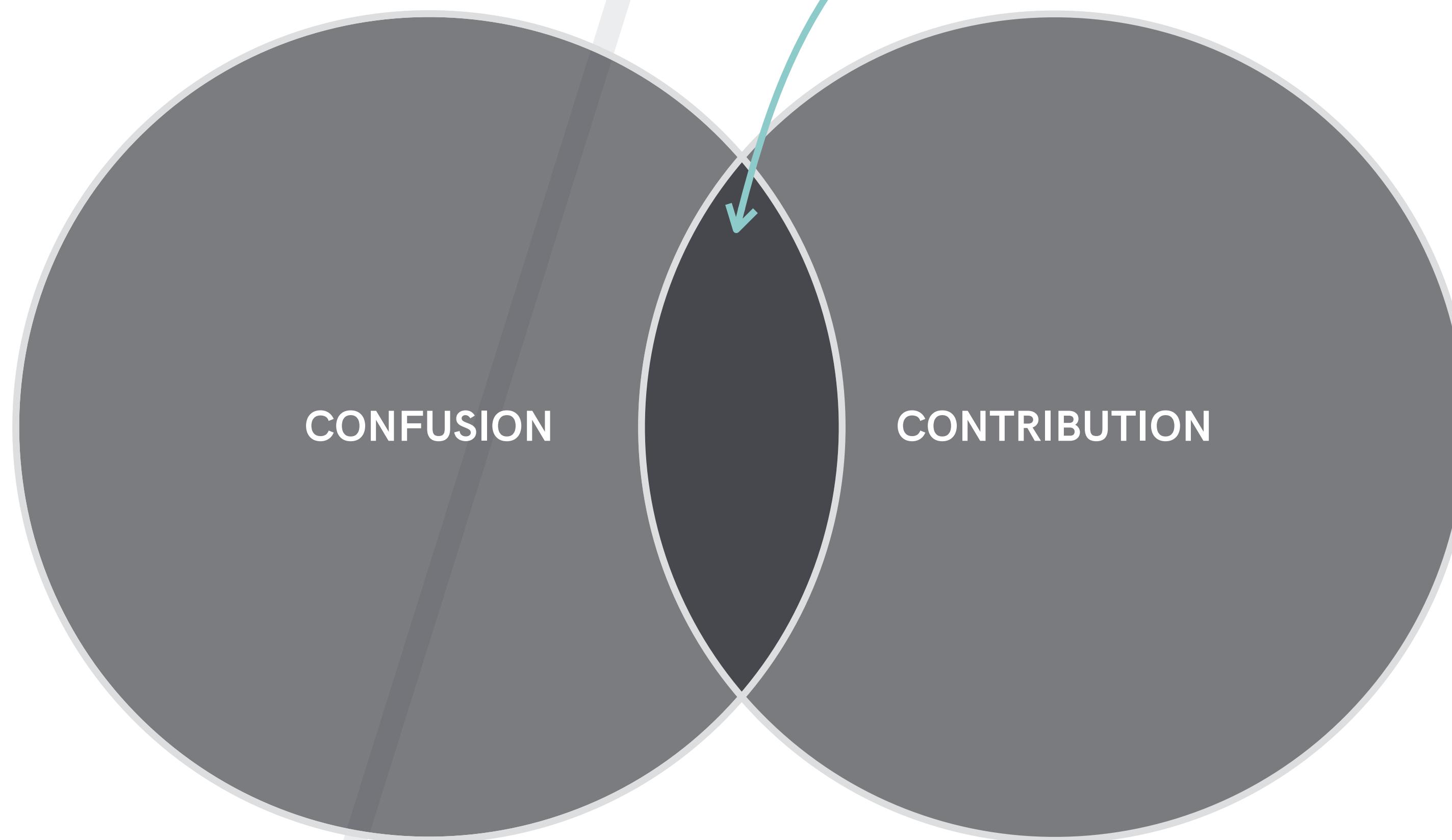
- “*I can't write code.*”
- “*I'm not really good at this.*”
- “*I'd just be a burden.*”
- “*They already have enough people smarter than me.*”



# Luckily...



# Luckily...



# Ask questions

*The most useless problem statement that one can face is "it doesn't work", yet we seem to get it far too often.*

- Thiago Maciera



# The newcomer's paradox...



*When you ask for help, some friendly soul will no doubt tell you that "it's easy, just do foo, bar and baz." Except for you, it is not easy, there may be no documentation for foo, bar is not doing what it is supposed to be doing and what is this baz thing anyway with its eight disambiguation entries on Wikipedia?*

— Leslie Hawthorne



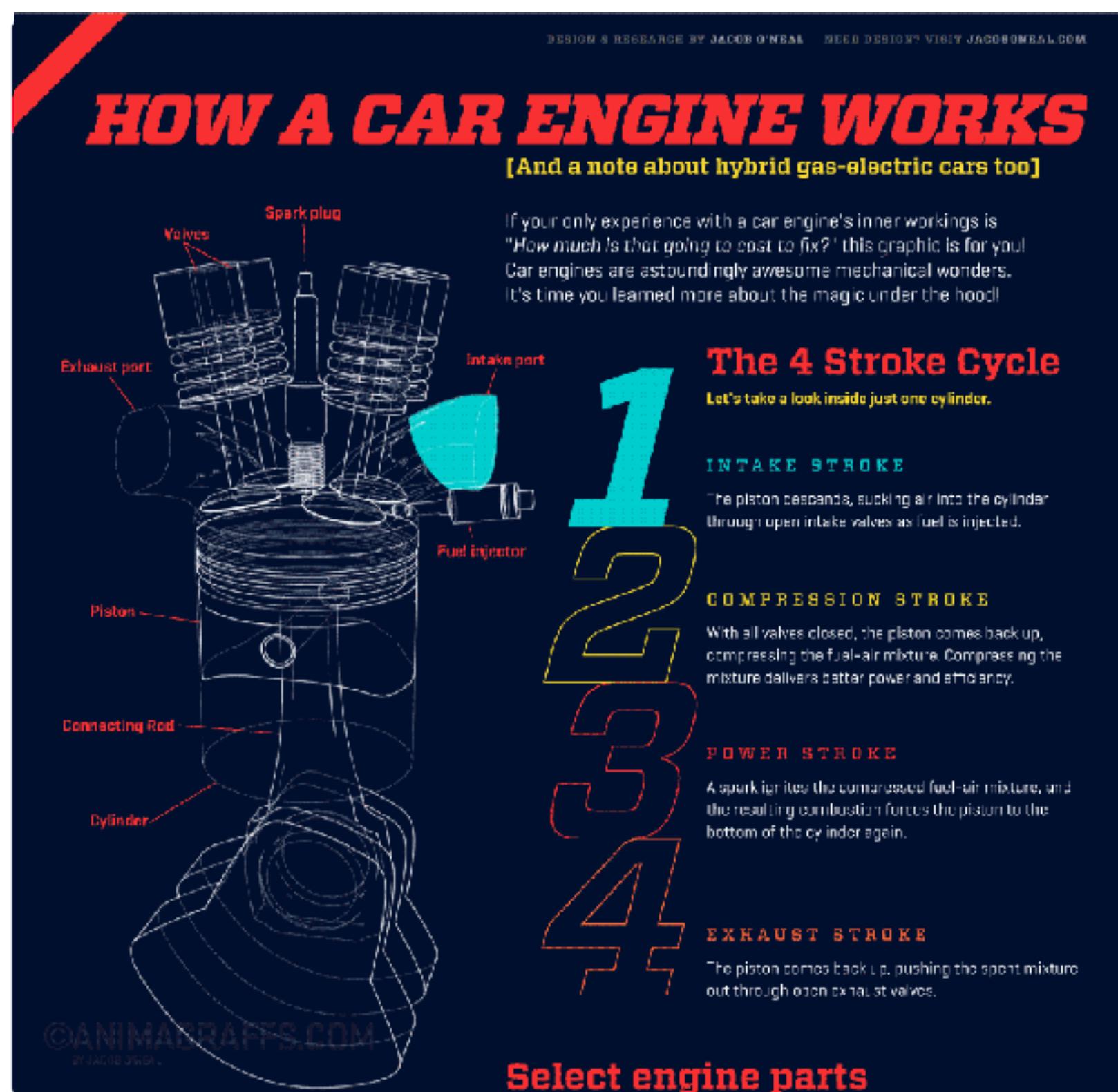
**Mara Averick**  
@dataandme

😩 my sister says my artfully-crafted analogy  
is worthless bc people don't know the  
4-stroke engine cycle...

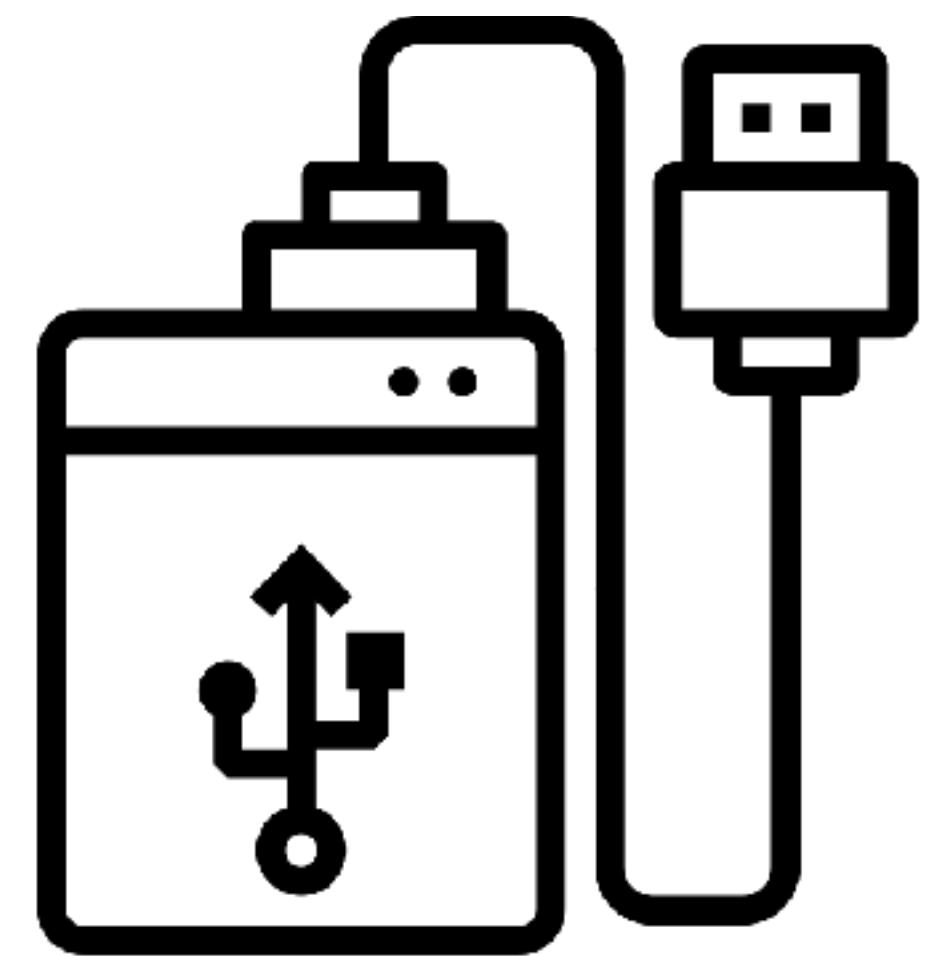
Internet: please disagree 🚗

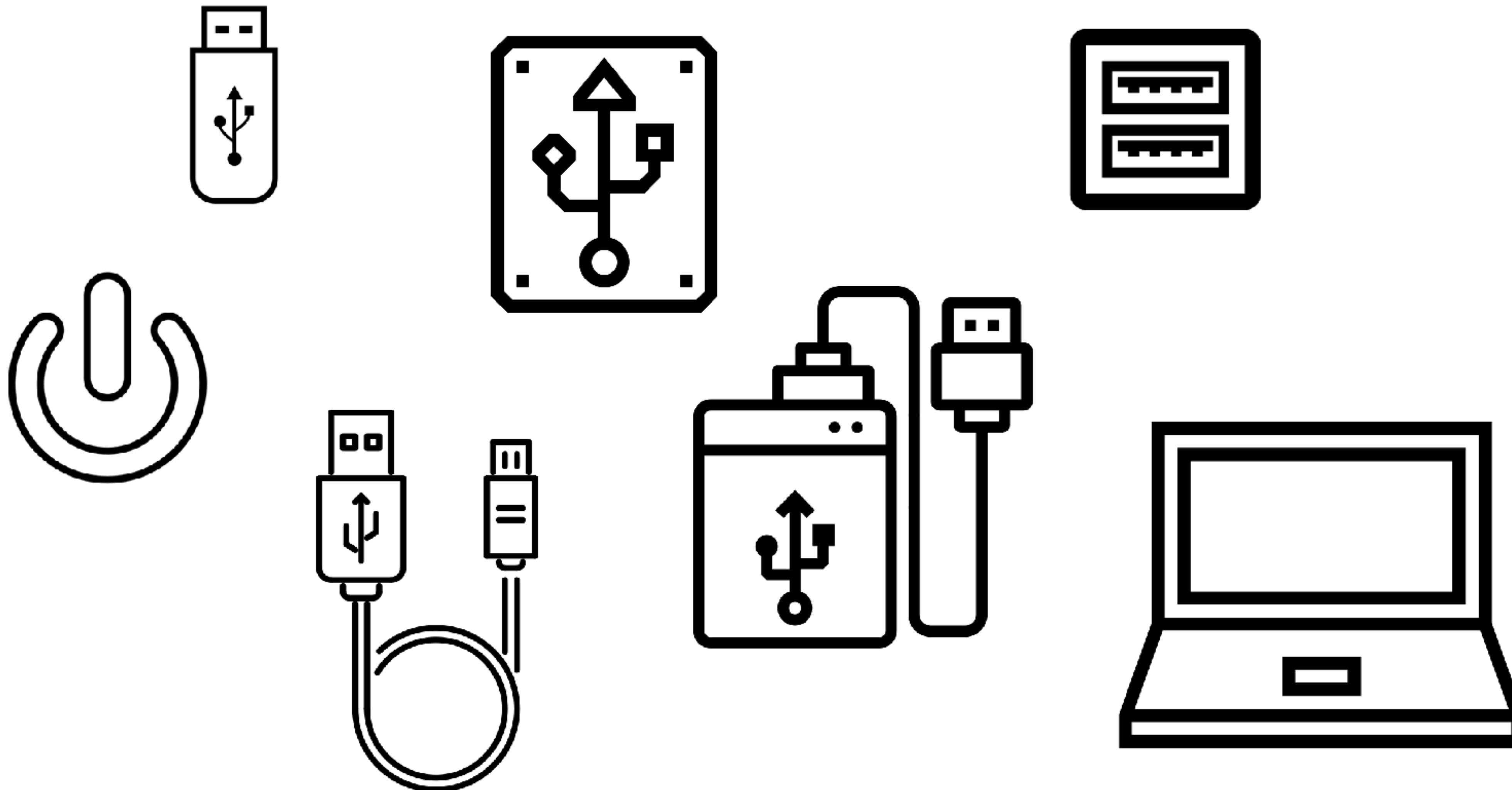


[buff.ly/2iteUyZ](https://buff.ly/2iteUyZ) 📸 @animagraffs

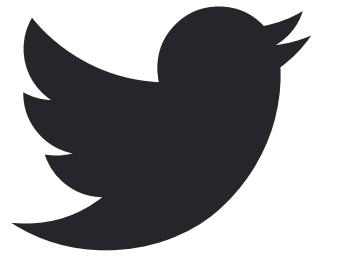


1:21 PM - 18 Nov 2017





# where to ask



Twitter



StackOverflow



RStudio Community

# the magic of reprex



# reprex raison d'être



# Keys to reprex-cellence

- ✓ Code that **actually runs**
- ✓ Code that **doesn't have to be run**
- ✓ Code that **can be easily run**

The screenshot shows an RStudio interface with the following components:

- Left Panel (Code Editor):** An R script titled "2017-01-03-reprex-magic.Rmd" containing the following code:

```
1 library(visdat)
2 
3 vis_miss(airquality)
4 
5 library(ggplot2)
6 
7 ggplot(airquality,
8       aes(x = Ozone,
9            y = Solar.R)) +
10   geom_point()
11 
12 library(naniar)
13 
14 ggplot(airquality,
15       aes(x = Ozone,
16            y = Solar.R)) +
17   geom_missing_point()
```
- Top Bar:** Shows tabs for "2017-01-03-reprex-magic.Rmd", "Untitled1\*", and "Untitled2\*". It also includes "Run", "Source", and "Console" buttons.
- Console:** Displays the output of running the script:

```
> reprex::reprex()
Rendered reprex ready on the clipboard.
> reprex::reprex()
Rendered reprex ready on the clipboard.

Restarting R session...

> reprex::reprex()
Rendered reprex ready on the clipboard.
> |
```
- Bottom Panel (Viewer):** Shows the rendered output of the code:

```
vis_miss(airquality)
#> Error in eval(expr, envir, enclos): could not find function "vis_miss"

ggplot(airquality,
       aes(x = Ozone,
            y = Solar.R)) +
  geom_point()
#> Error in eval(expr, envir, enclos): could not find function "ggplot"

ggplot(airquality,
       aes(x = Ozone,
            y = Solar.R)) +
  geom_missing_point()
#> Error in eval(expr, envir, enclos): could not find function "ggplot"
```

Source: Nick Tierney. "Magic reprex." 2017-01-11 <<http://www.njtierney.com/post/2017/01/11/magic-reprex/>>

# Nailing those reprexes?

- Help others ask questions.
- Answer questions.
- Write about it.

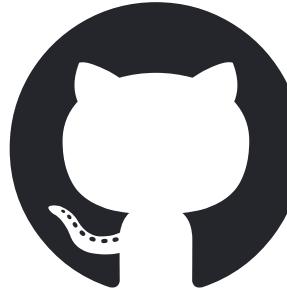
# where to answer



StackOverflow

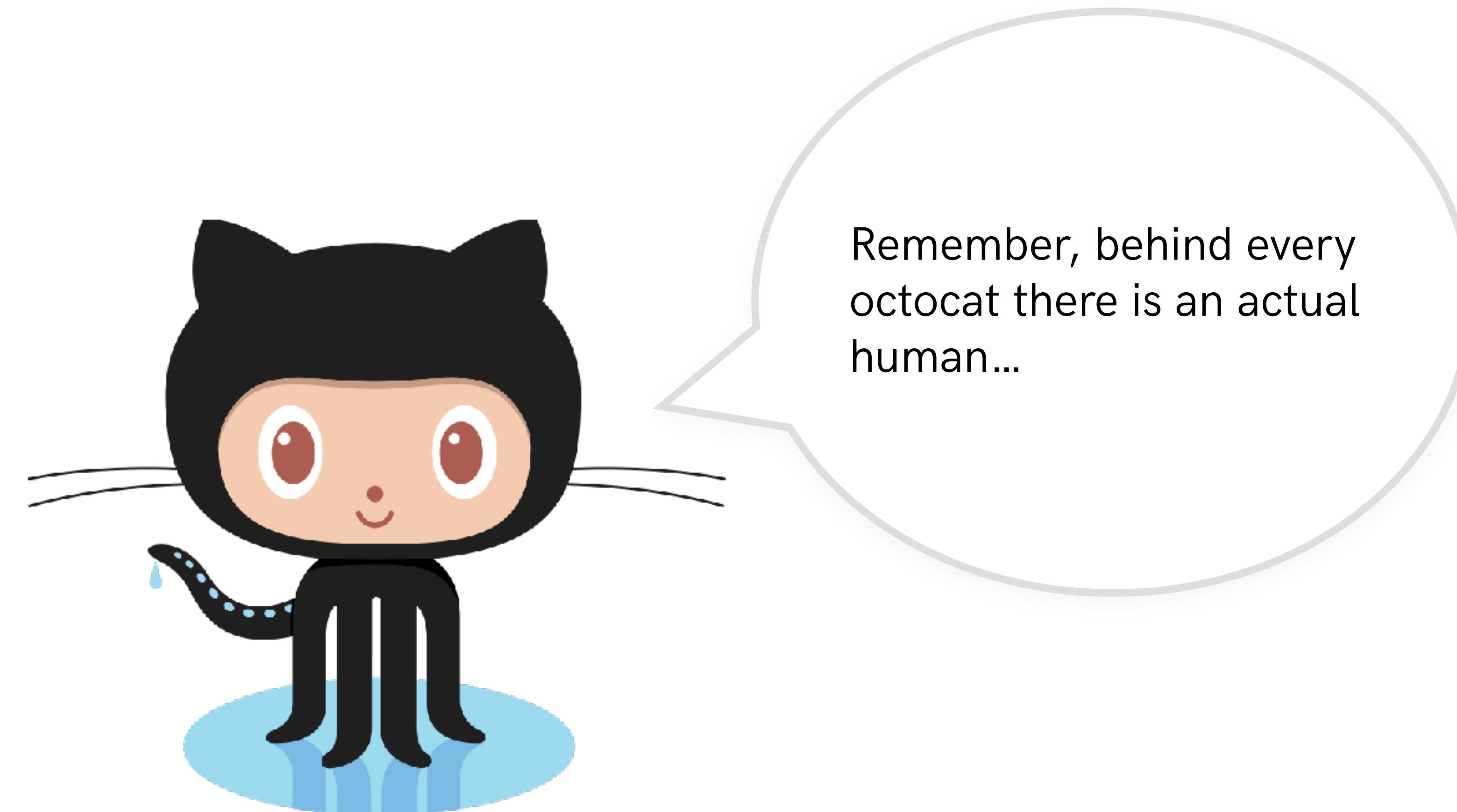


RStudio Community



GitHub

# File issues



# the anatomy of an issue



PROBLEM DESCRIPTION

EXPECTED BEHAVIOUR

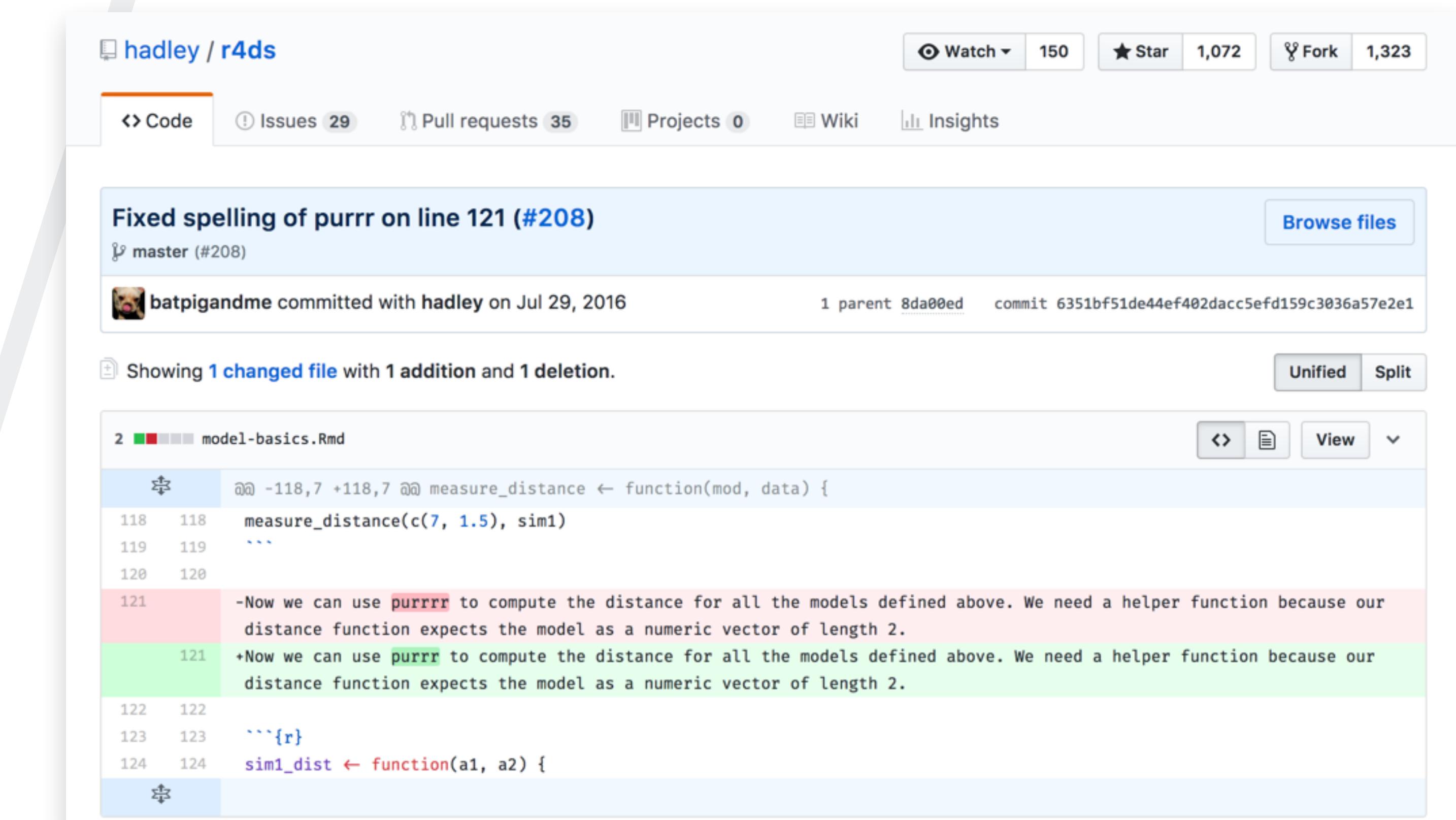
REPREX

# Contribute documentation

*“Innocence lost is not easily regained. The designer simply cannot predict the problems people will have, the misinterpretations that will arise, and the errors that will get made.”*

— Donald Norman, *The Design of Everyday Things*

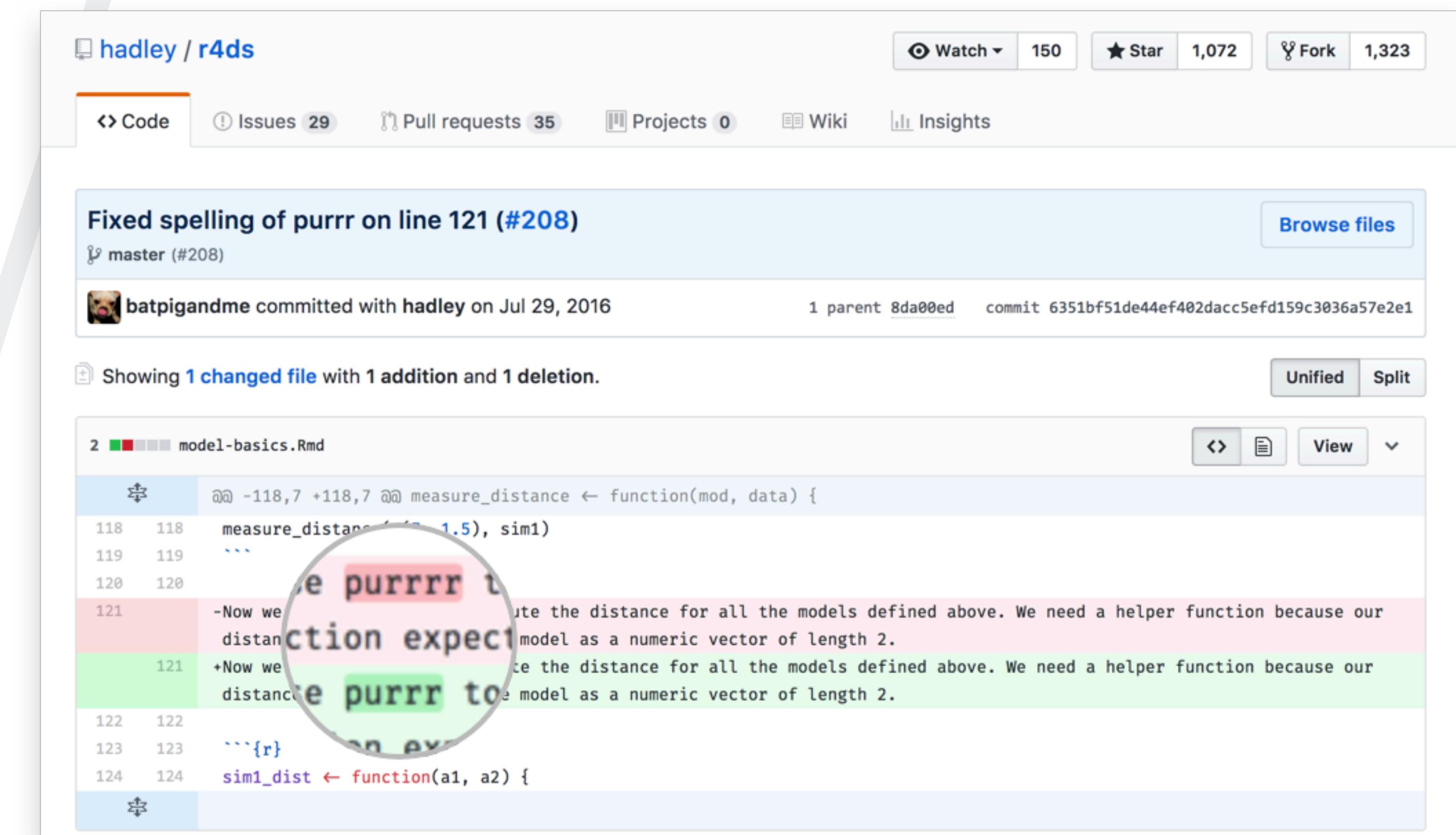
# My first “contribution”



The screenshot shows a GitHub pull request page for the repository "hadley / r4ds". The pull request is titled "Fixed spelling of purrr on line 121 (#208)". It was created by "batpigandme" and merged by "hadley" on July 29, 2016. The commit hash is 6351bf51de44ef402dacc5efd159c3036a57e2e1. The commit message indicates that the spelling of "purrr" was corrected to "purrr". The code changes are shown in the "model-basics.Rmd" file, specifically lines 121 and 122.

```
@@ -118,7 +118,7 @@ measure_distance <- function(mod, data) {  
 118 118    measure_distance(c(7, 1.5), sim1)  
 119 119    ...  
 120 120  
 121 -Now we can use purrr to compute the distance for all the models defined above. We need a helper function because our  
     distance function expects the model as a numeric vector of length 2.  
 122 122 +Now we can use purrr to compute the distance for all the models defined above. We need a helper function because our  
     distance function expects the model as a numeric vector of length 2.  
 123 123    ...{r}  
 124 124    sim1_dist <- function(a1, a2) {  
 125 125      ...  
 126 126    }  
 127 127  }
```

# My first “contribution”

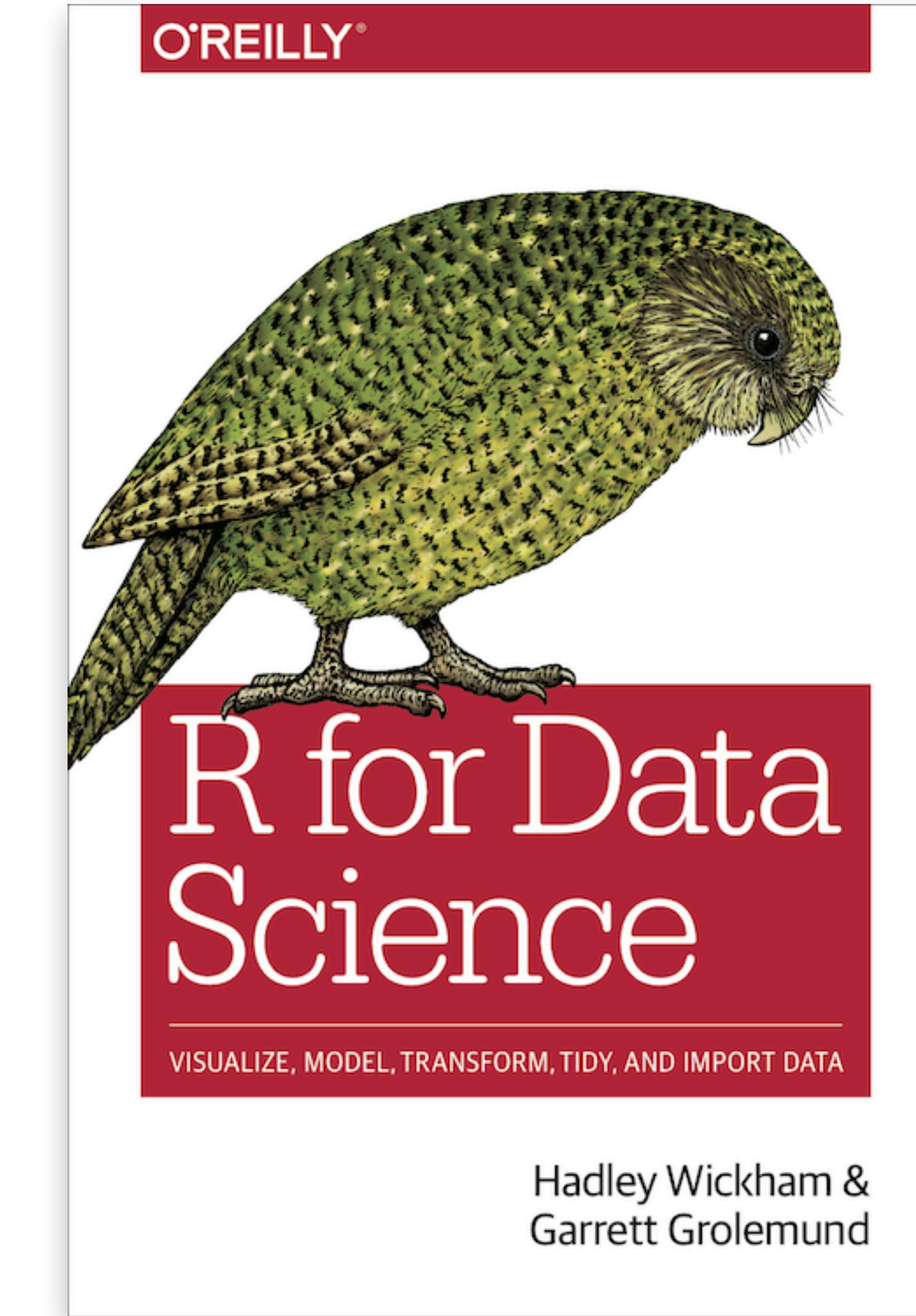


A screenshot of a GitHub pull request page for the repository "hadley / r4ds". The pull request is titled "Fixed spelling of purrr on line 121 (#208)". It was committed by "batpigandme" on Jul 29, 2016. The commit message is "Fixed spelling of purrr on line 121". The commit hash is 6351bf51de44ef402dacc5efd159c3036a57e2e1. The pull request has 1 parent and 1 addition. The code change is shown in the file "model-basics.Rmd". Line 121 was changed from "purrrr" to "purrr". A circular highlight is drawn around the word "purrrr" in the diff view.

```
diff --git a/model-basics.Rmd b/model-basics.Rmd
--- a/model-basics.Rmd
+++ b/model-basics.Rmd
@@ -118,7 +118,7 @@ measure_distance <- function(mod, data) {
 118   118     measure_distance(`^`[1.5], sim1)
 119   119   `^`[1.5]
 120   120 
 121 -Now we
 121 +Now we
 121   distance expect
 121 -Now we
 121 +distance expect
 121   purrr to
 122   122 
 123   123   `^`[r]
 124   124   sim1_dist <- function(a1, a2) {
```

# My first “contribution”

46



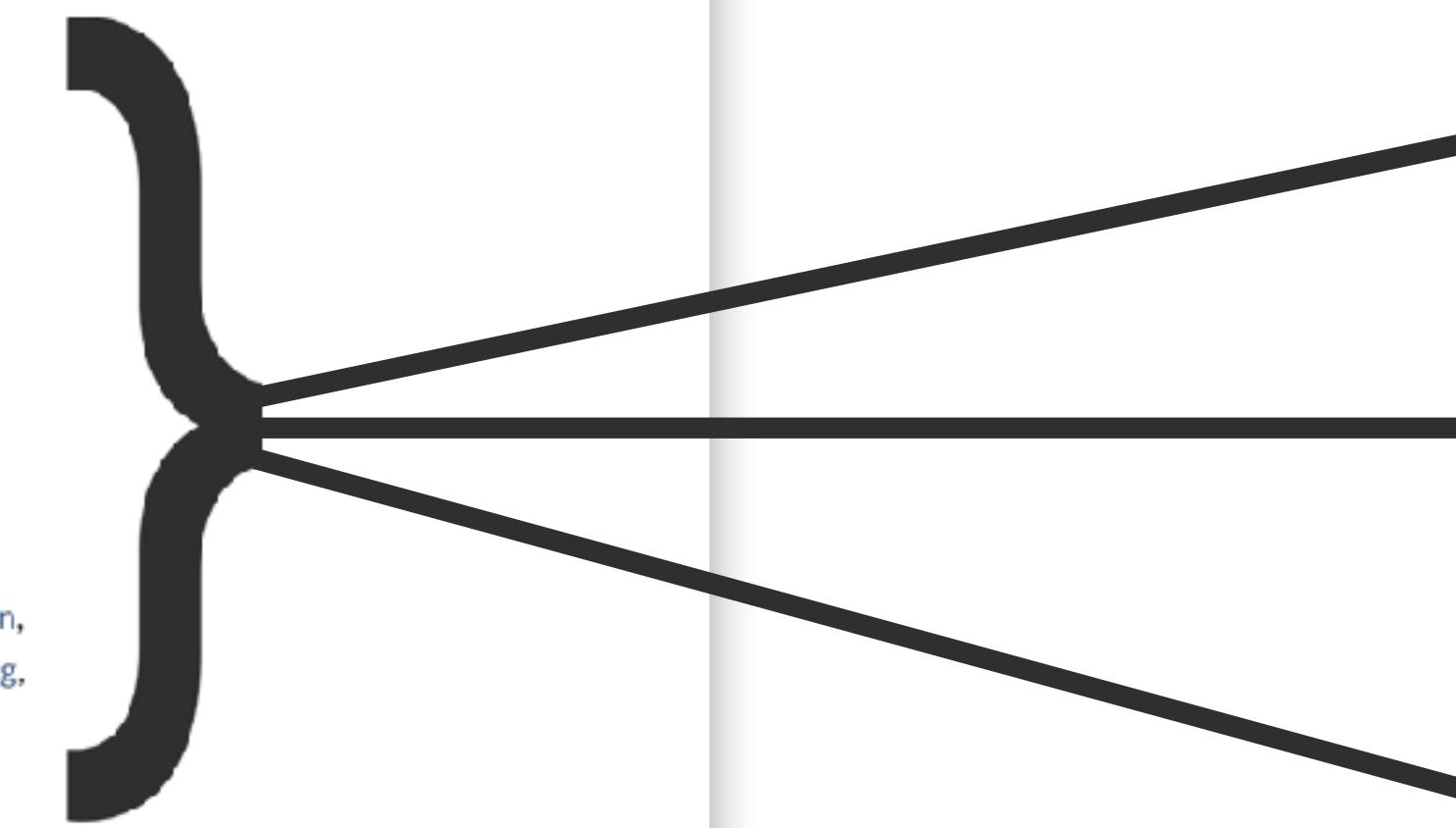
# every contribution counts...

The screenshot shows the Tidyverse Acknowledgments page. At the top, there is a navigation bar with links for Tidyverse, Packages, Articles (which is the active tab), Learn, Help, and Contribute. Below the navigation bar, the title "Acknowledgments" is displayed. A large text block lists the names of 109 contributors who have made issues, pull requests, or comments since tibble 1.2.0. At the bottom of the page, it says "The tidyverse is proudly supported by R Studio" and features social media icons for GitHub and Twitter.

Acknowledgments

We received issues, pull requests, and comments from 109 people since tibble 1.2.0. Thanks to everyone: @alexandersson, @adnbps, @AkhilNairAmey, @alexhallam, @alibat, @amjuzi, @AndreMikulec, @andrewjpfeiffer, @ashiklom, @atribe, @bapfeld, @barnettjacob, @behrman, @BillDunlap, @BruceZhaoR, @cassiusoat, @cboettig, @cderv, @ckluss, @ClaytonJY, @colearendt, @csgillespie, @dalejbarr, @dan87134, @DavisVaughan, @ddiez, @dhicks, @dldpd, @drewgendaleau, @drolejoel, @echasnovski, @edzer, @ElsLommelen, @etiennebr, @FabianRoger, @garrettgman, @gavinsimpson, @geotheory, @ginolhac, @hadley, @happyshows, @heavywatal, @helix123, @holstius, @huftis, @ianmcook, @manuelcostigan, @janschulz, @javierluraschi, @jennybc, @jimhester, @joelgombin, @jonathan\_g, @kendonB, @kevinushey, @khughitt, @kismsu, @krlmlr, @kwstat, @LaDilettante, @lcolladotor, @lionel-, @lpmarco, @m-sostero, @MarcusWalz, @matteodefelice, @mattfidler, @mgirlich, @michaellevy, @MikeBadescu, @mkearney, @mmuurr, @Monduiz, @mubeenarasack, @mundl, @nbenn, @ncarchedi, @NikNakk, @noamross, @ntguardian, @p0bs, @patperry, @pgensler, @phalexo, @pssguy, @r2evans, @rentrop, @richierocks, @Rongpeng, @s-fleck, @sainathadapa, @sebschub, @sibojan, @slonik-az, @sskim47, @t-kalinowski, @thercast, @thornend, @tjmahr, @trinker, @vnijjs, @vspinu, @vvrably, @wibom, @wpetry, @yedle, @yihui, @yutannihilation, and @Zedseyou.

The tidyverse is proudly supported by R Studio

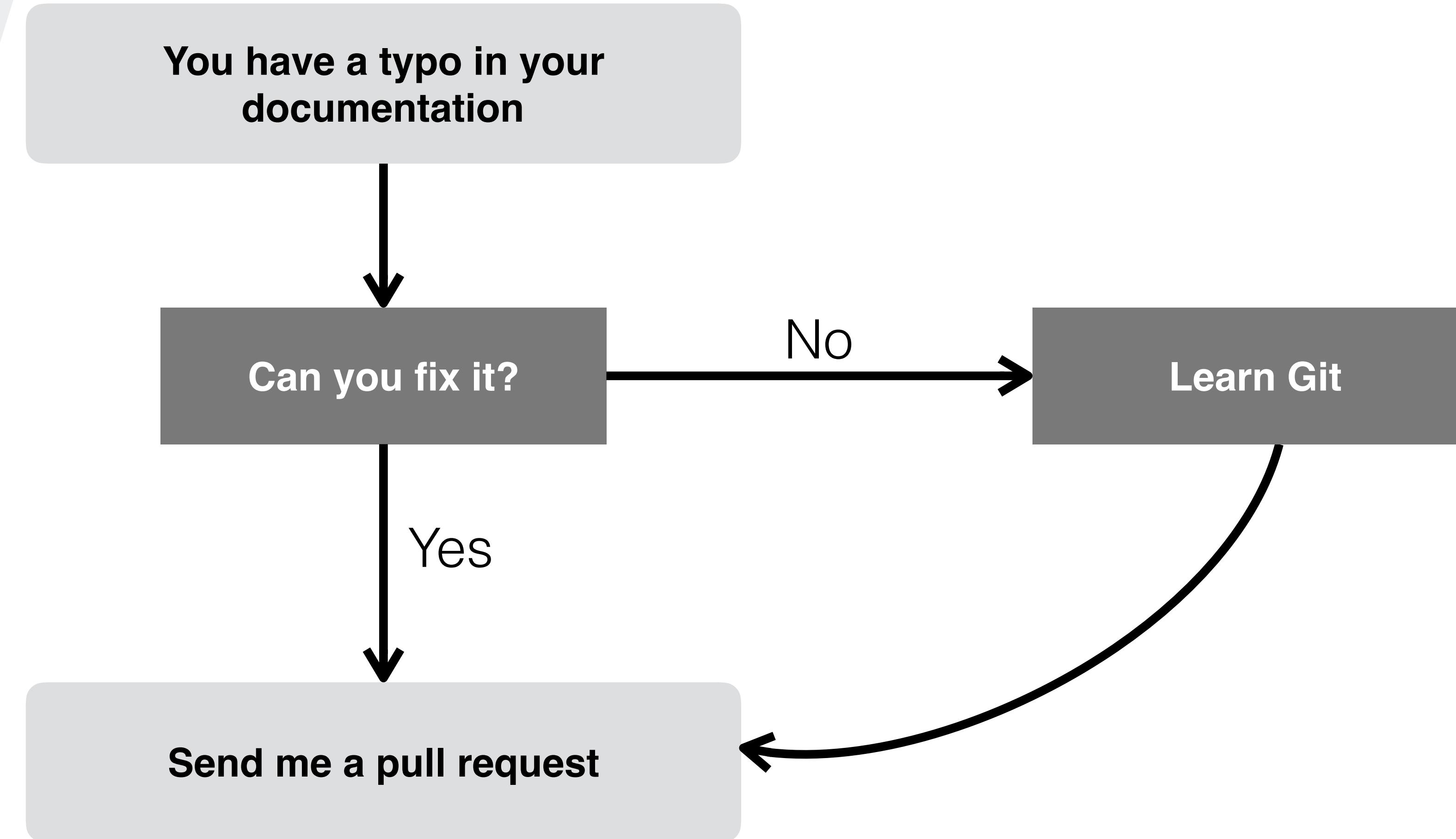


PULL REQUESTS

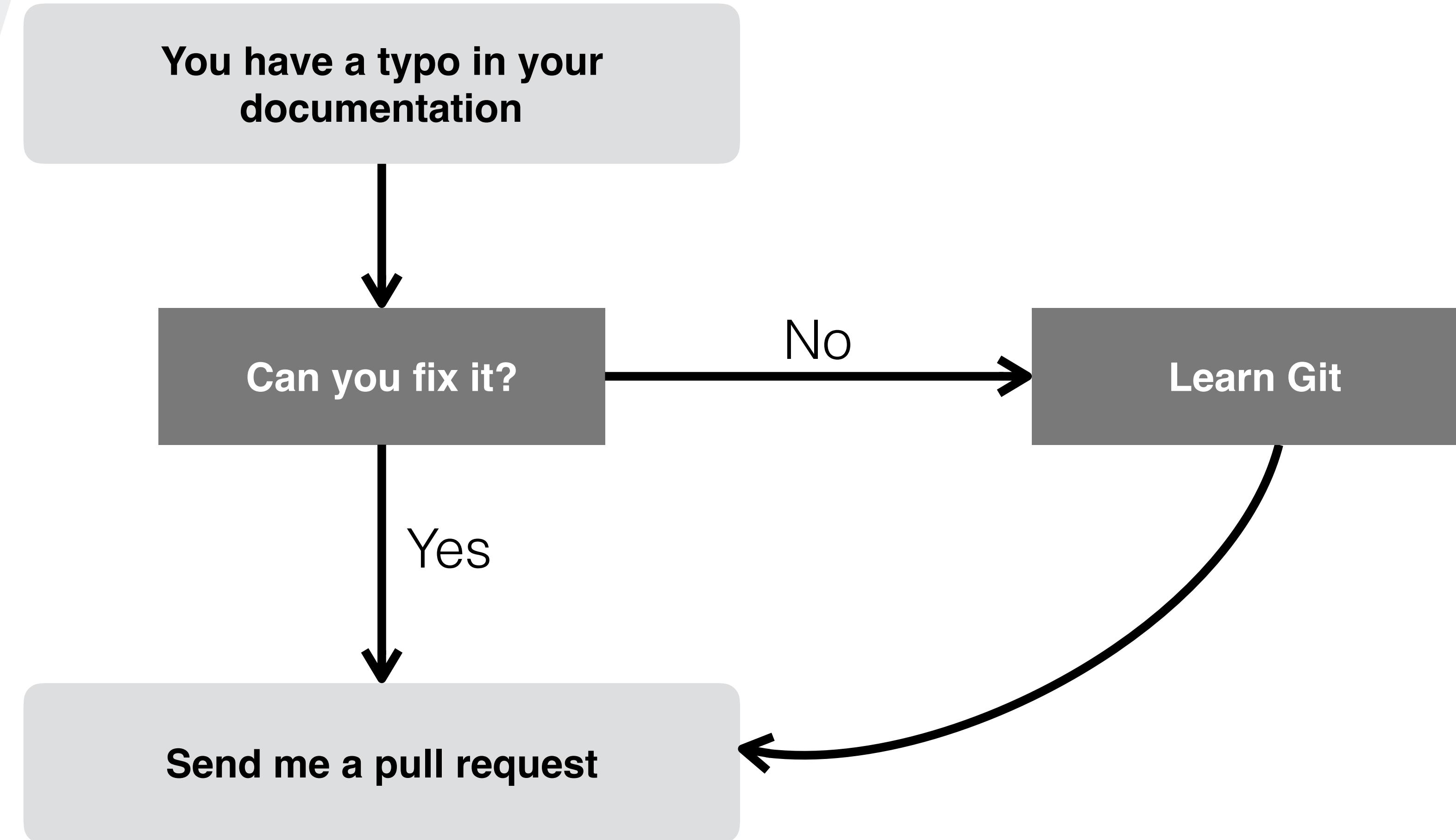
ISSUES

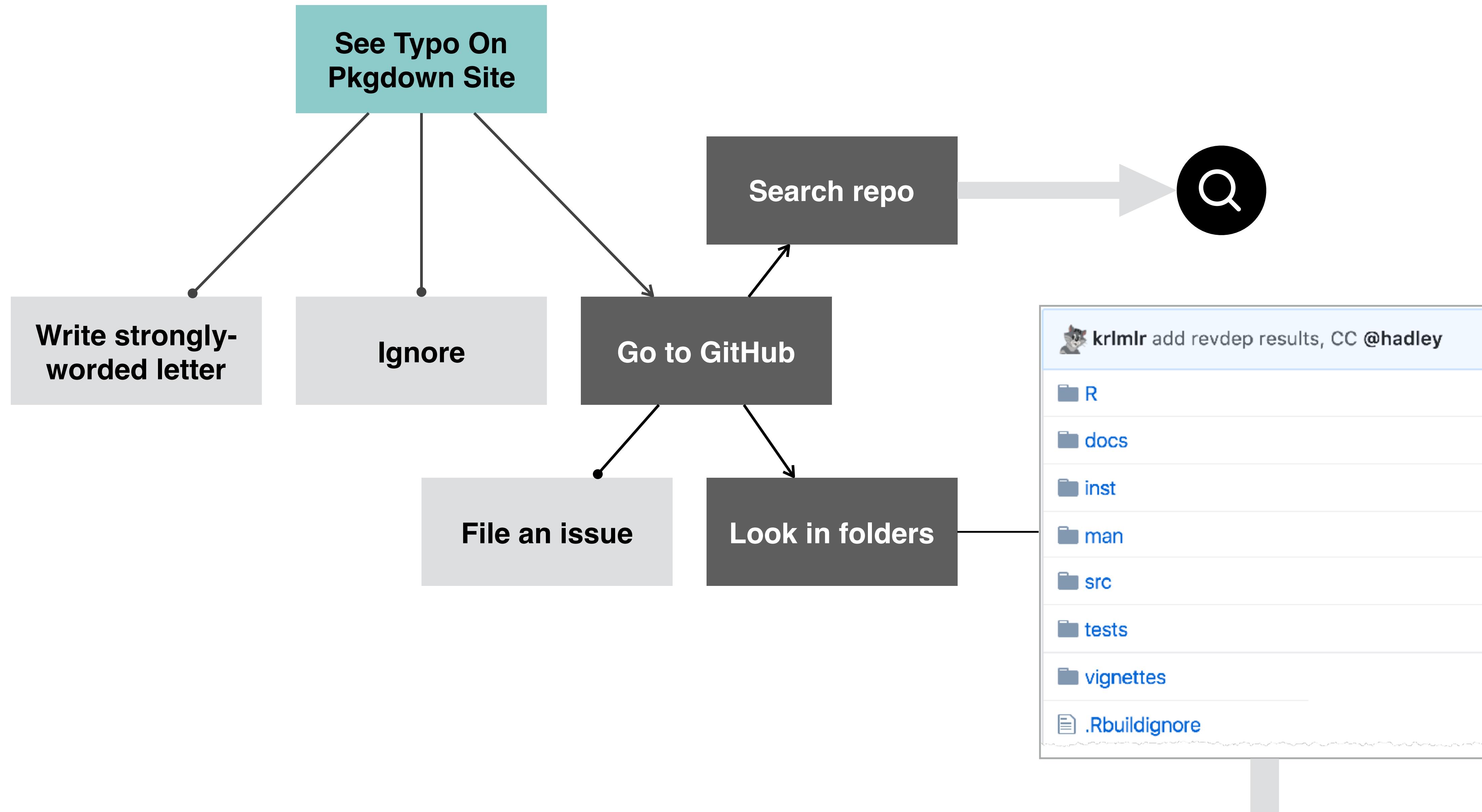
COMMENTS

# tpyos

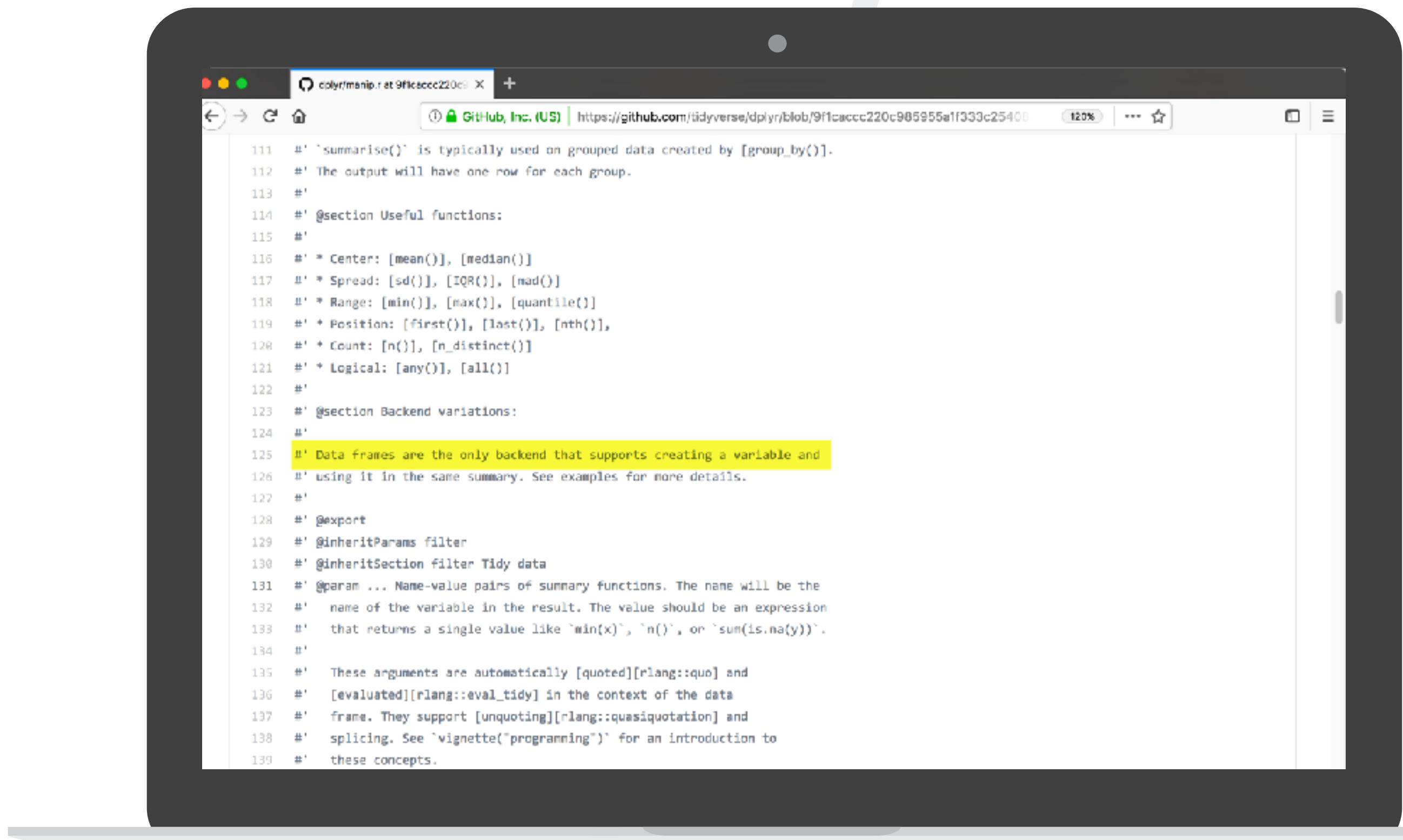
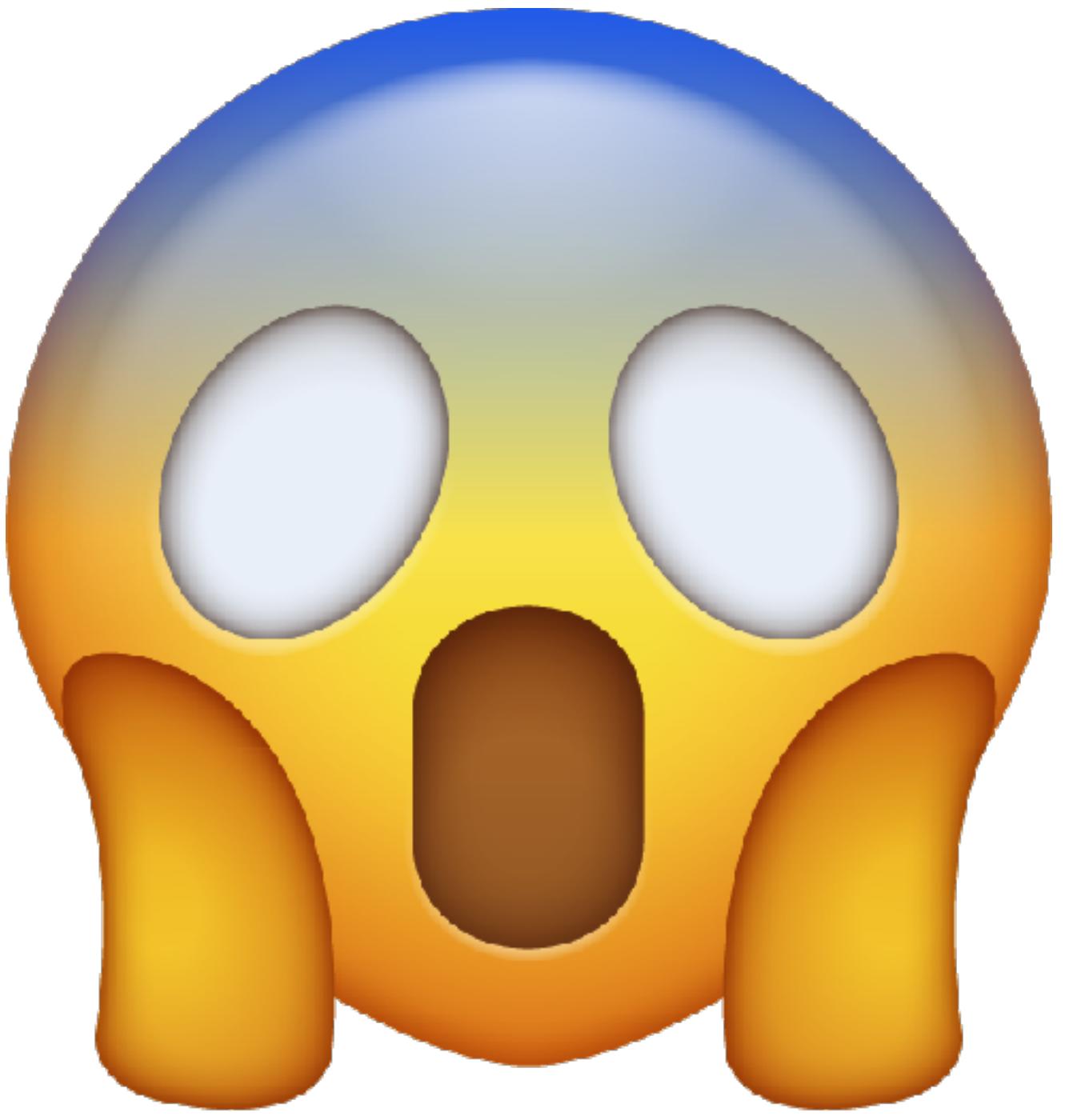


# typos





# Source code?!

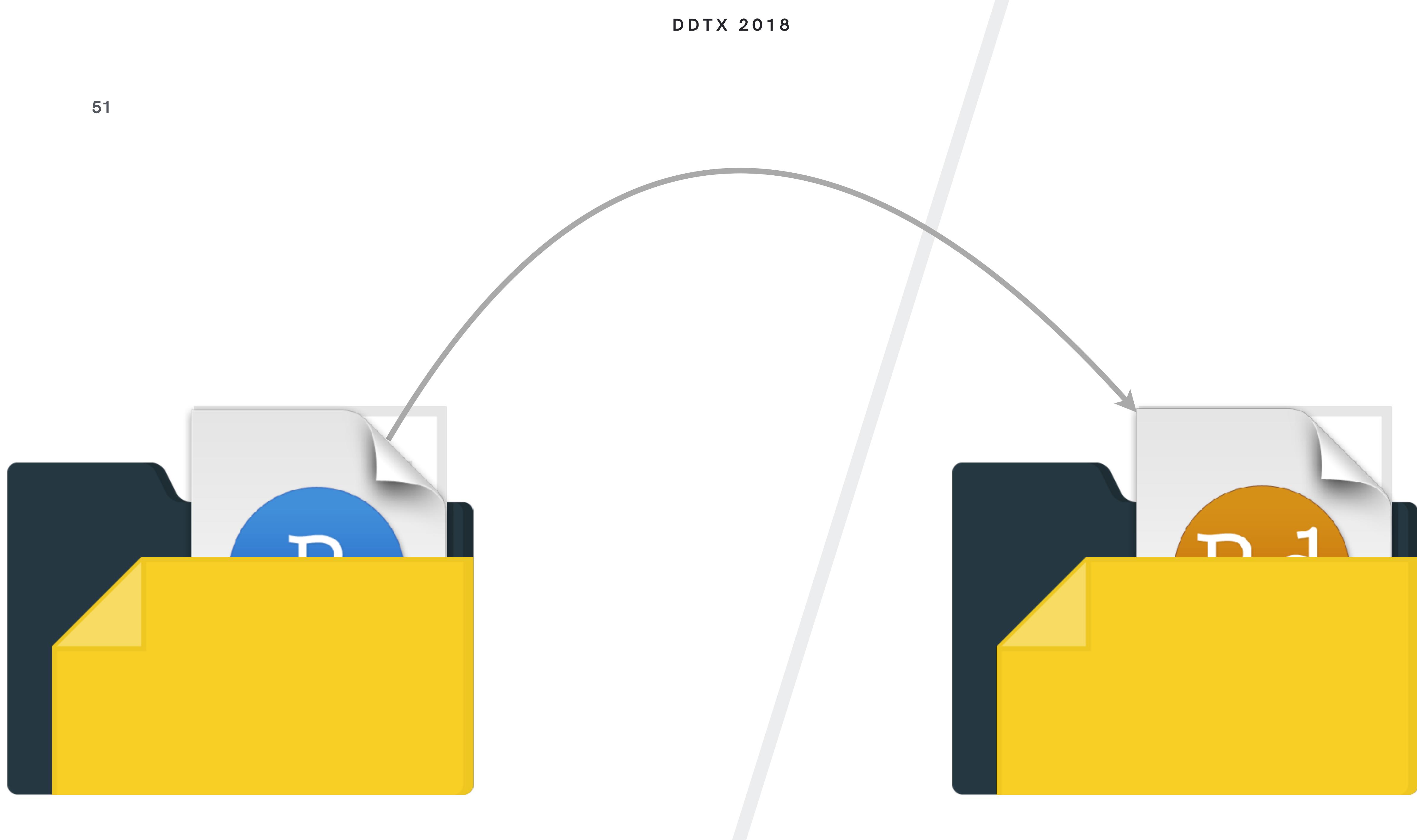


The image shows a screenshot of a web browser window displaying R source code on GitHub. The code is part of the `dplyr` package, specifically the `summarise` function. The browser is a dark-themed version of Safari, with the GitHub URL visible in the address bar: <https://github.com/tidyverse/dplyr/blob/9f1ccccc220c905955a1f333c25401d0r/man/man/summarise.R>. The code is annotated with line numbers and comments. A specific comment is highlighted in yellow: `## Data frames are the only backend that supports creating a variable and` `## using it in the same summary. See examples for more details.` This highlights a key feature of the `dplyr` package where variables can be created within the summarise function and used in subsequent calculations.

```
111 #' `summarise()` is typically used on grouped data created by `group_by()`.  
112 #' The output will have one row for each group.  
113 #'  
114 #' @section Useful functions:  
115 #'  
116 #' * Center: [mean()], [median()]  
117 #' * Spread: [sd()], [IQR()], [mad()]  
118 #' * Range: [min()], [max()], [quantile()]  
119 #' * Position: [first()], [last()], [nth()],  
120 #' * Count: [n()], [n_distinct()]  
121 #' * Logical: [any()], [all()]  
122 #'  
123 #' @section Backend variations:  
124 #'  
125 #' Data frames are the only backend that supports creating a variable and  
126 #' using it in the same summary. See examples for more details.  
127 #'  
128 #' @export  
129 #' @inheritParams filter  
130 #' @inheritSection filter Tidy data  
131 #' @param ... Name-value pairs of summary functions. The name will be the  
132 #' name of the variable in the result. The value should be an expression  
133 #' that returns a single value like `min(x)`, `n()`, or `sum(is.na(y))`.  
134 #'  
135 #' These arguments are automatically [quoted][rlang::quo] and  
136 #' [evaluated][rlang::eval_tidy] in the context of the data  
137 #' frame. They support [unquoting][rlang::quasiquotation] and  
138 #' splicing. See `vignette("programming")` for an introduction to  
139 #' these concepts.
```

# Source code?!

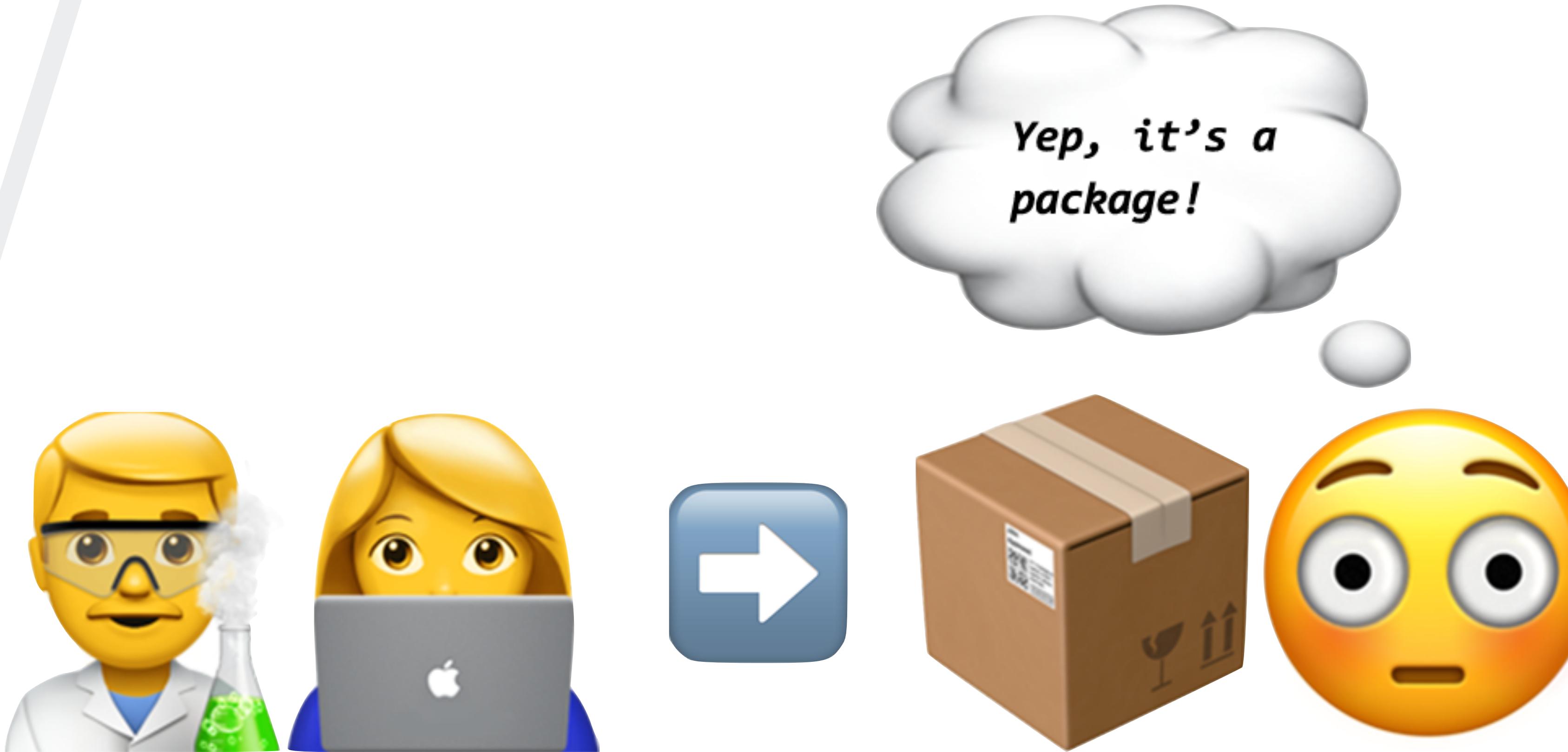
A screenshot of a web browser window displaying an R script from GitHub. The script is titled 'dplyr/manip.R' and has a commit hash '9f1ccccc220c905955a1f333c2540'. The code is annotated with several lines of explanatory text, notably line 125 which states: '#' Data frames are the only backend that supports creating a variable and '#' using it in the same summary. See examples for more details. The browser interface includes standard navigation buttons, a search bar, and a status bar indicating 120% zoom.



# Go to the source...



# require(n00bs)



# require(n00bs)

When assigning reviewers to a submission, we aim to pair experienced reviewers with new ones, or reviewers with expertise on a package's programming methods with those experienced in its field of application.



## **How rOpenSci uses Code Review to Promote Reproducible Science**

[ropensci.org](http://ropensci.org) highlighted with Highly

# BABY STEPS WITH MORE PACKAGES

**ROXYGEN2**

generates documentation from specially-formatted comments, used by all tidyverse packages

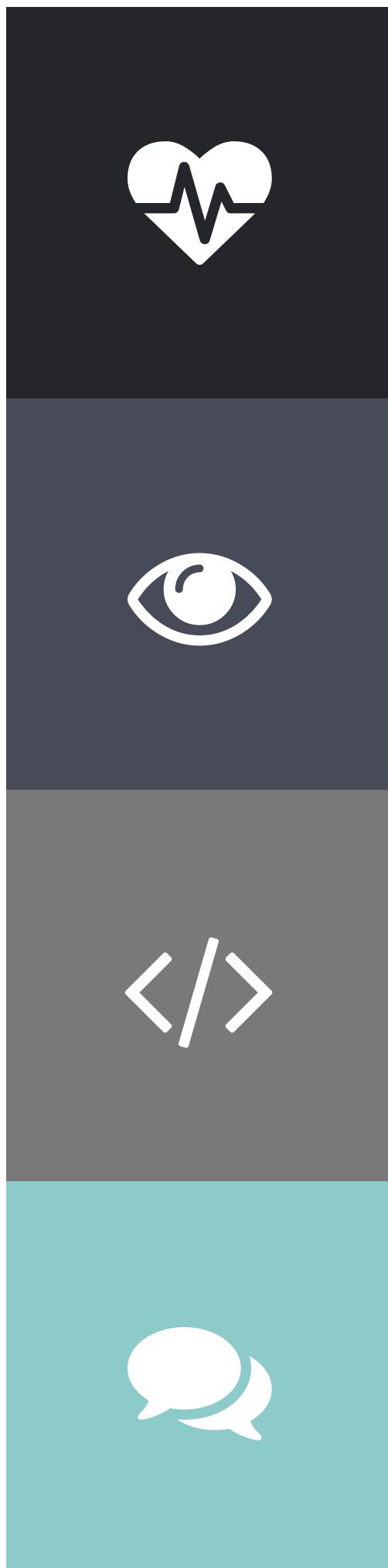
**DEVTOOLS**

makes package development easier by providing R functions that simplify common tasks

**TESTTHAT**

provides functions that make it easy to create unit tests for R packages, used throughout tidyverse

# Hints for happy contributing in the tidyverse



GET THE PULSE OF A PROJECT

WATCH THE REPO

READ THE CODE

DISCUSS YOUR IDEAS

# leaRn out loud



OMG, I just learned a thing!

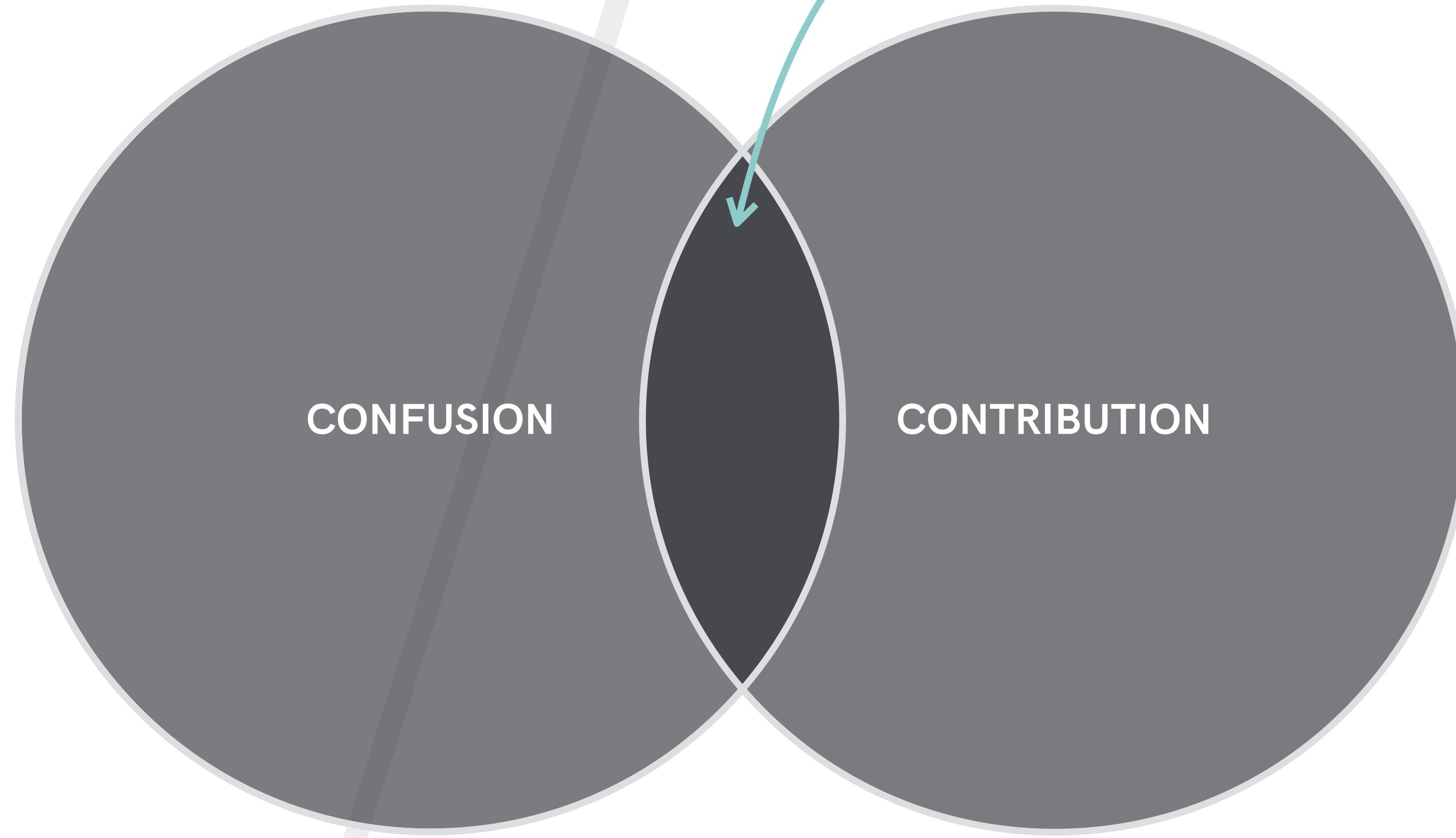
# give back

*Acting out of the goodness of your heart, or something*



# embrace it...

58



# Thank You

<http://bit.ly/mara-ddtx>



# Thank You

<http://bit.ly/mara-ddtx>

