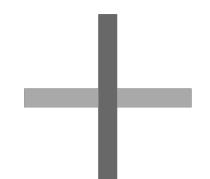
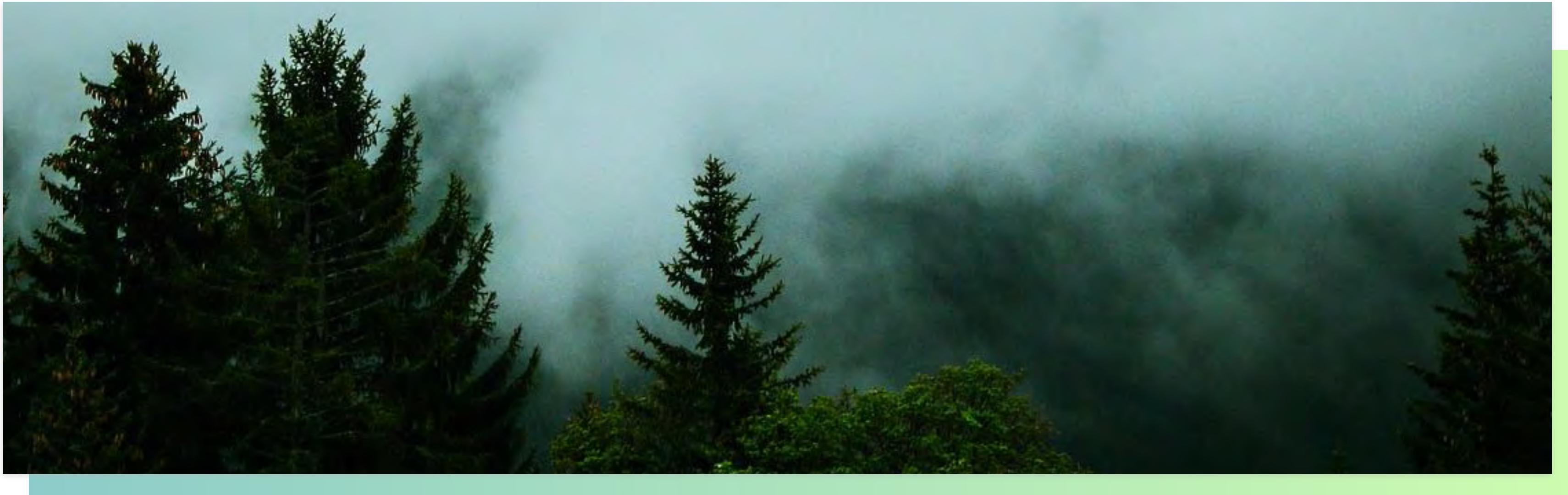




# PILGRIM'S PROGRESS

a journey from confusion to contribution



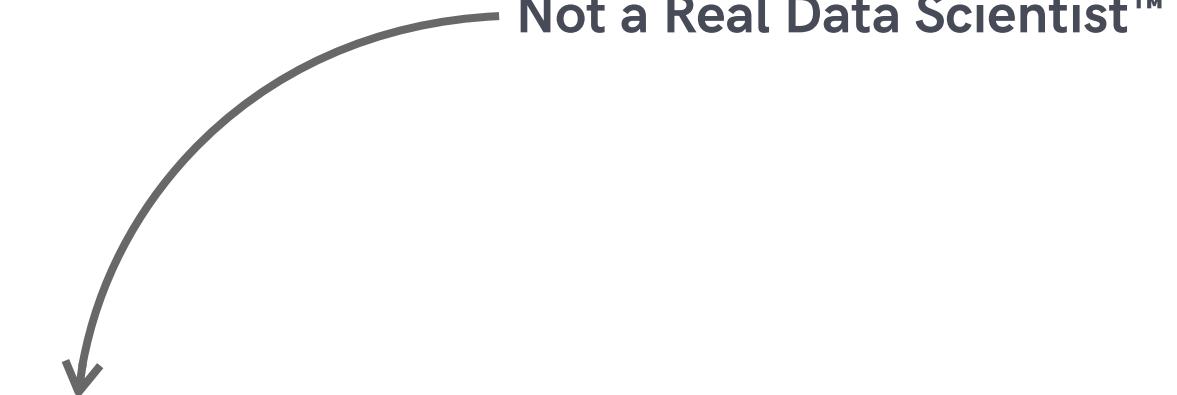


**Mara Averick**  
TIDYVERSE DEV ADVOCATE, RSTUDIO



**Mara Averick**  
TIDYVERSE DEV ADVOCATE, RSTUDIO

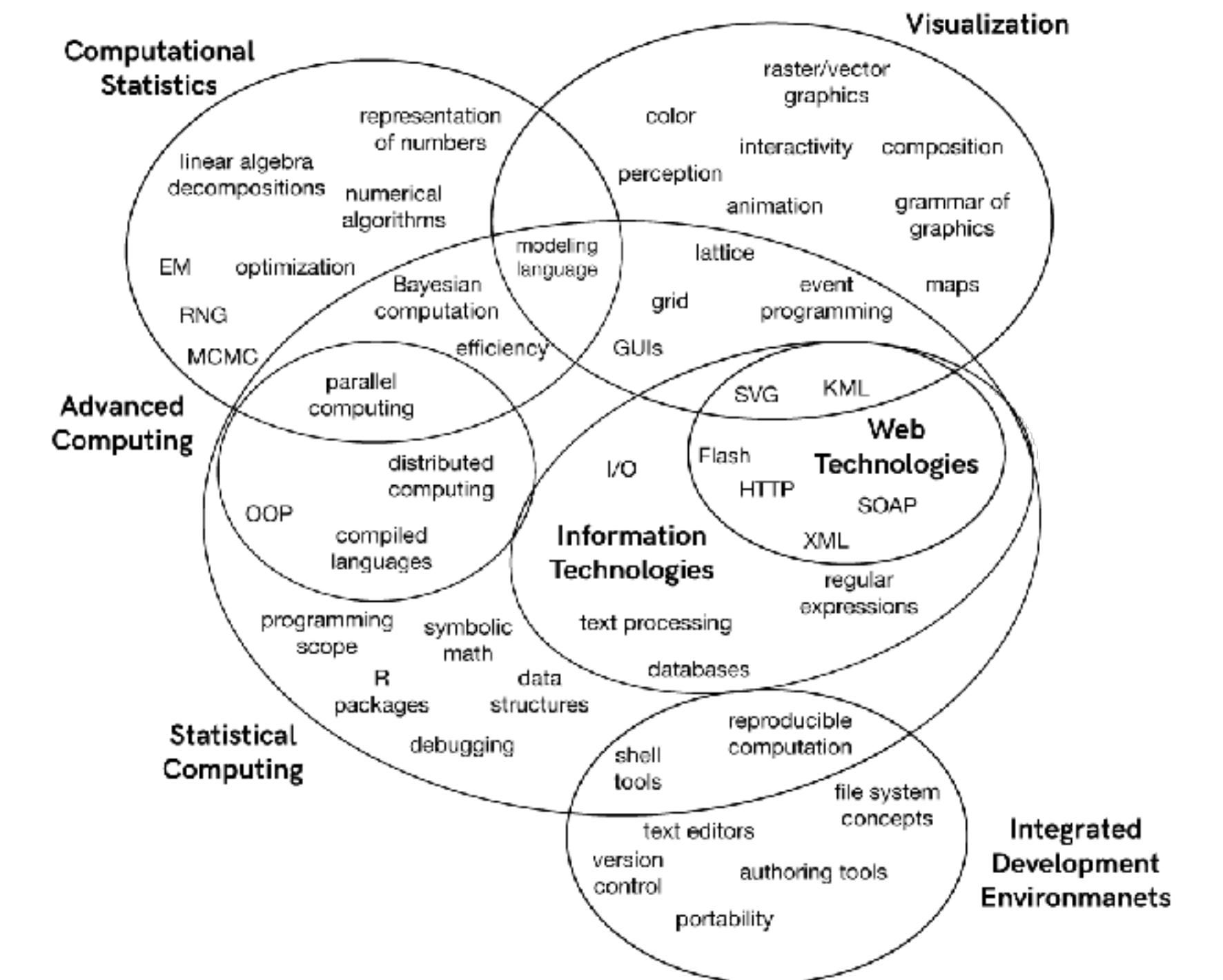
Not a Real Data Scientist™





# Mara Averick

TIDYVERSE DEV ADVOCATE, RSTUDIO





# 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



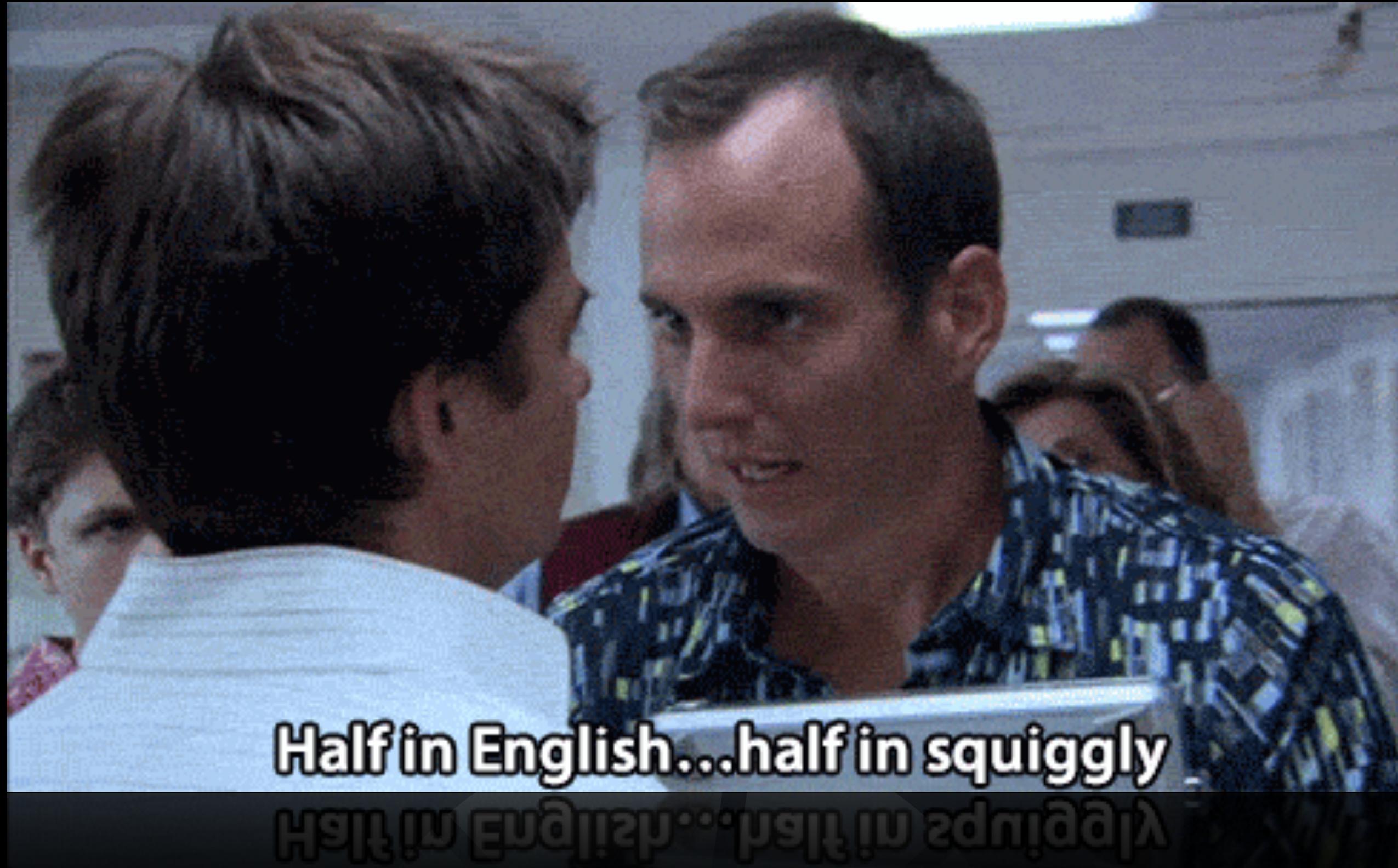
# An aside on the title



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

Following

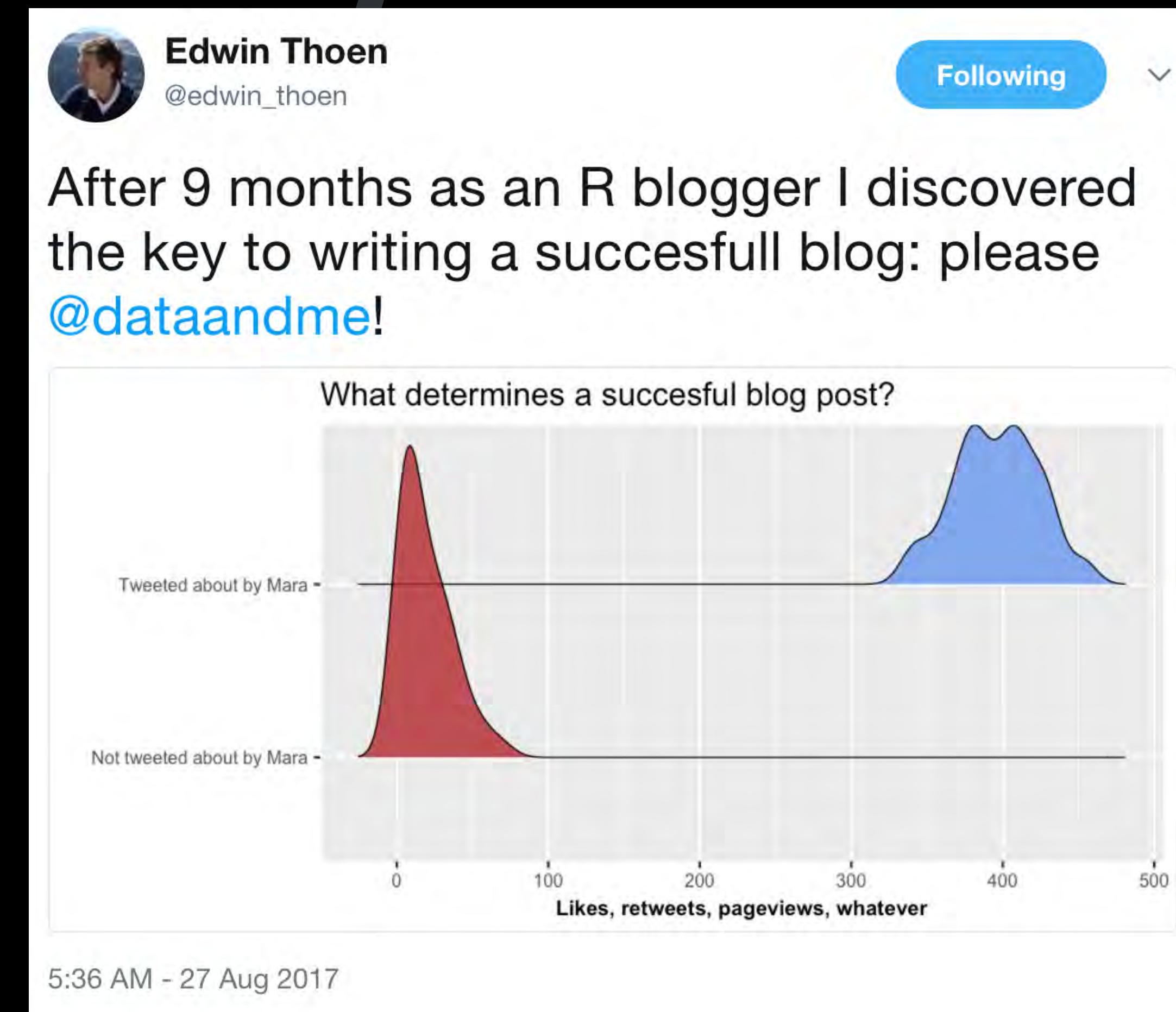
If you want to make a deep non-technical contribution to the field of data science, I can not think of a better role model than [@dataandme](#)

7:25 AM - 26 Jul 2017

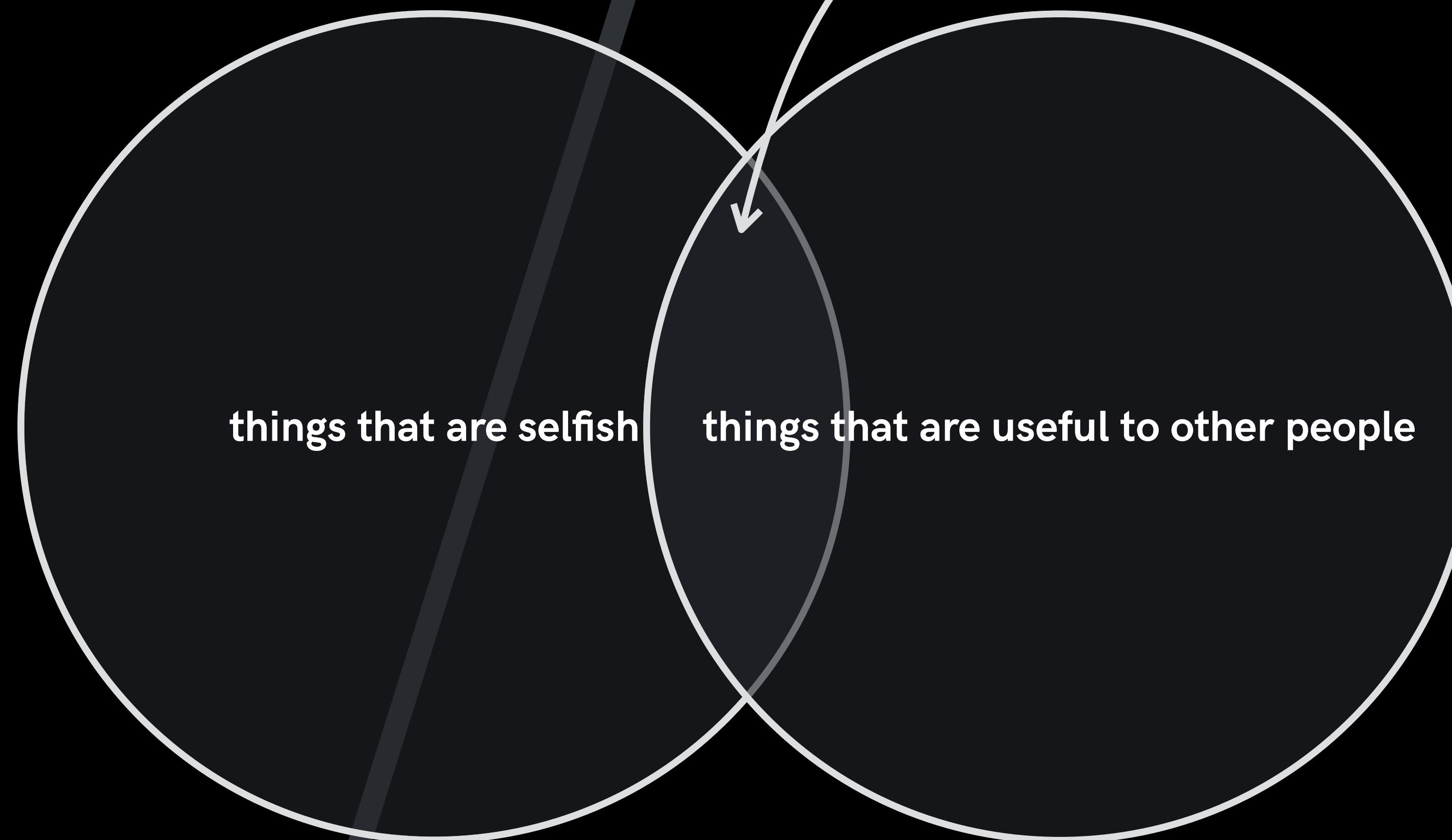
21 Retweets 169 Likes

6 21 169

# but...



# 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



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



Replying 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*



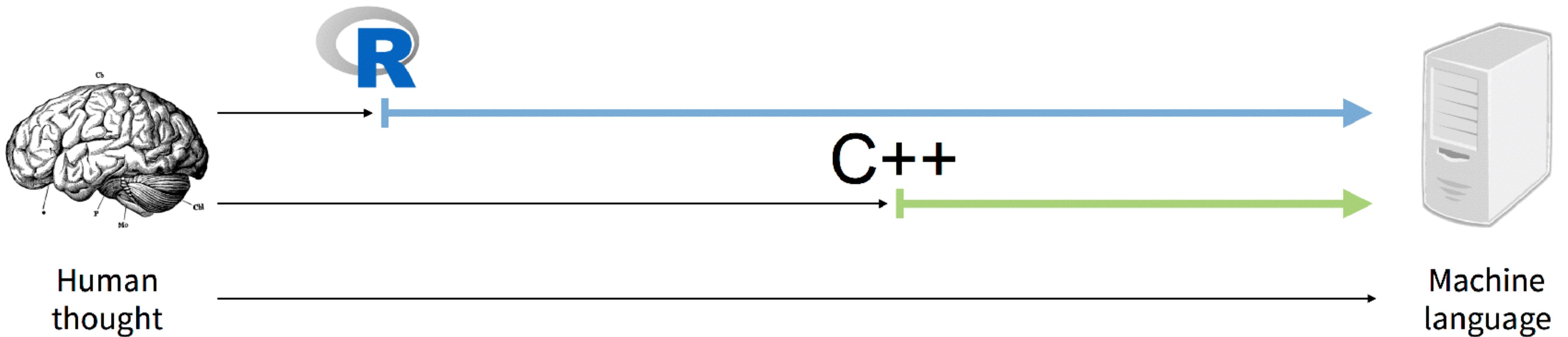
*Expert*

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.

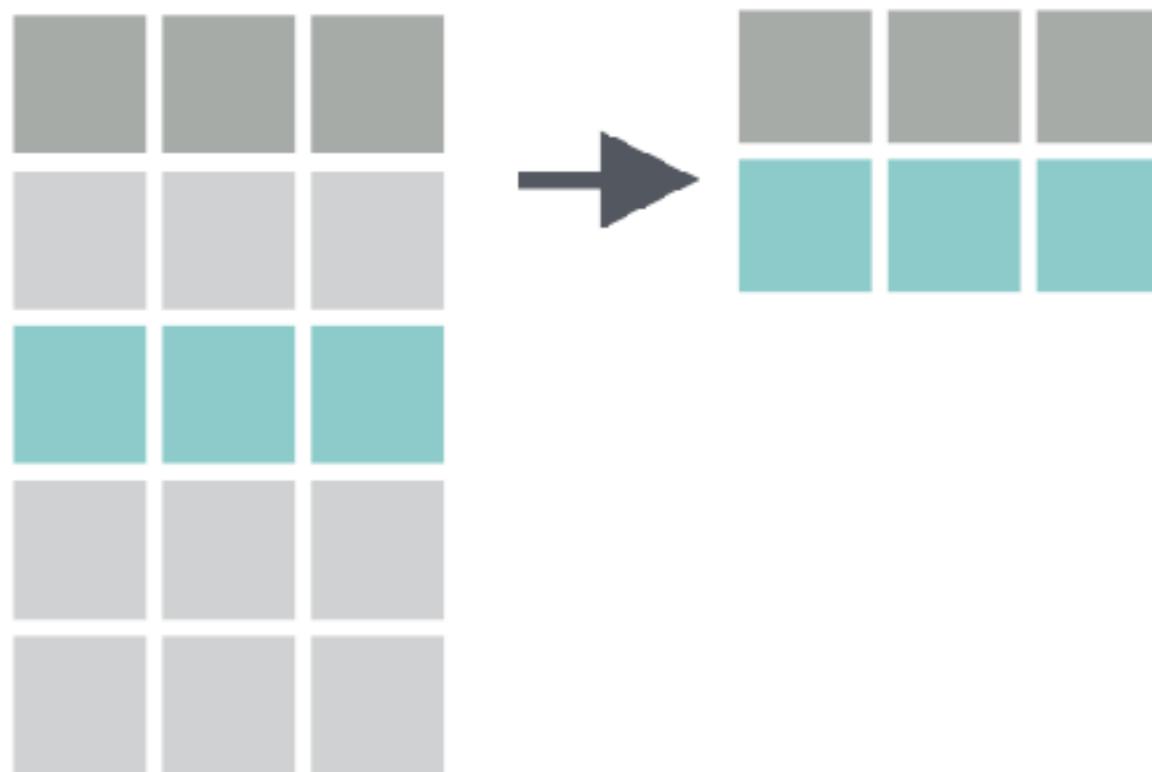
## COMPOSABLE

Combine with other functions for multi-step operations.

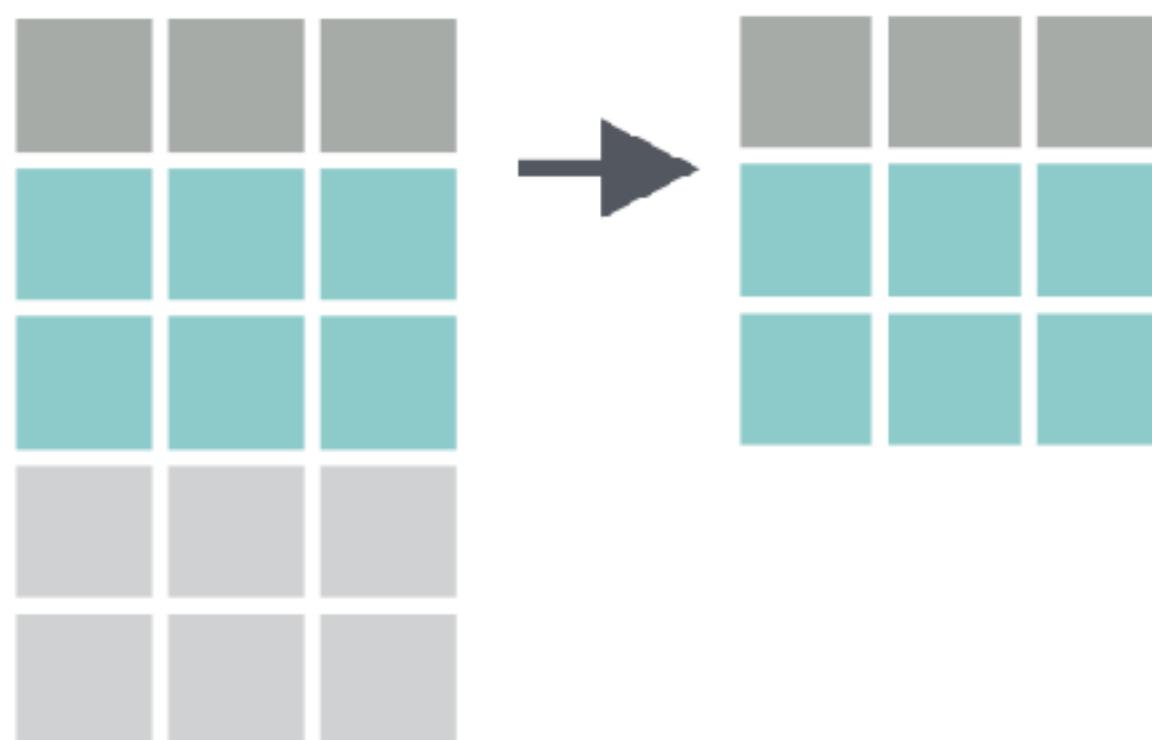
## DESIGNED FOR HUMANS

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



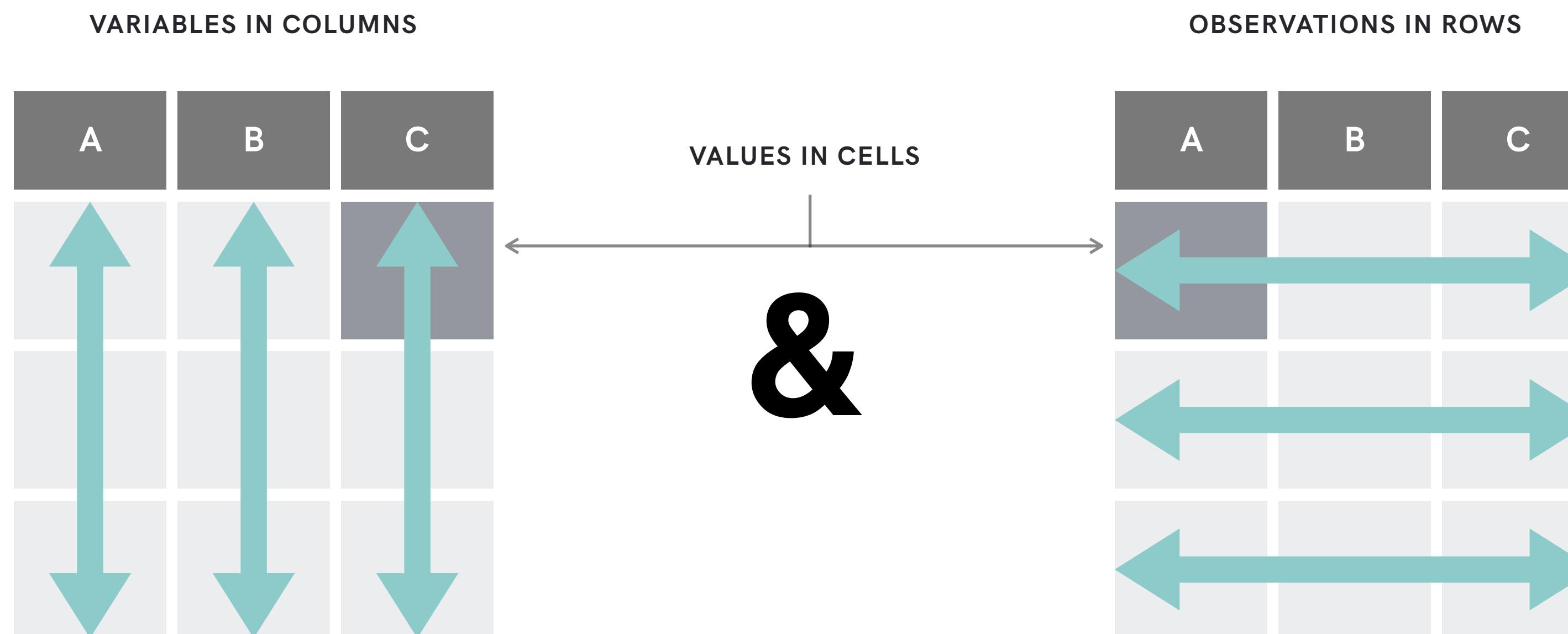
# COMPOSE WITH THE PIPE

```
iris %>%
```

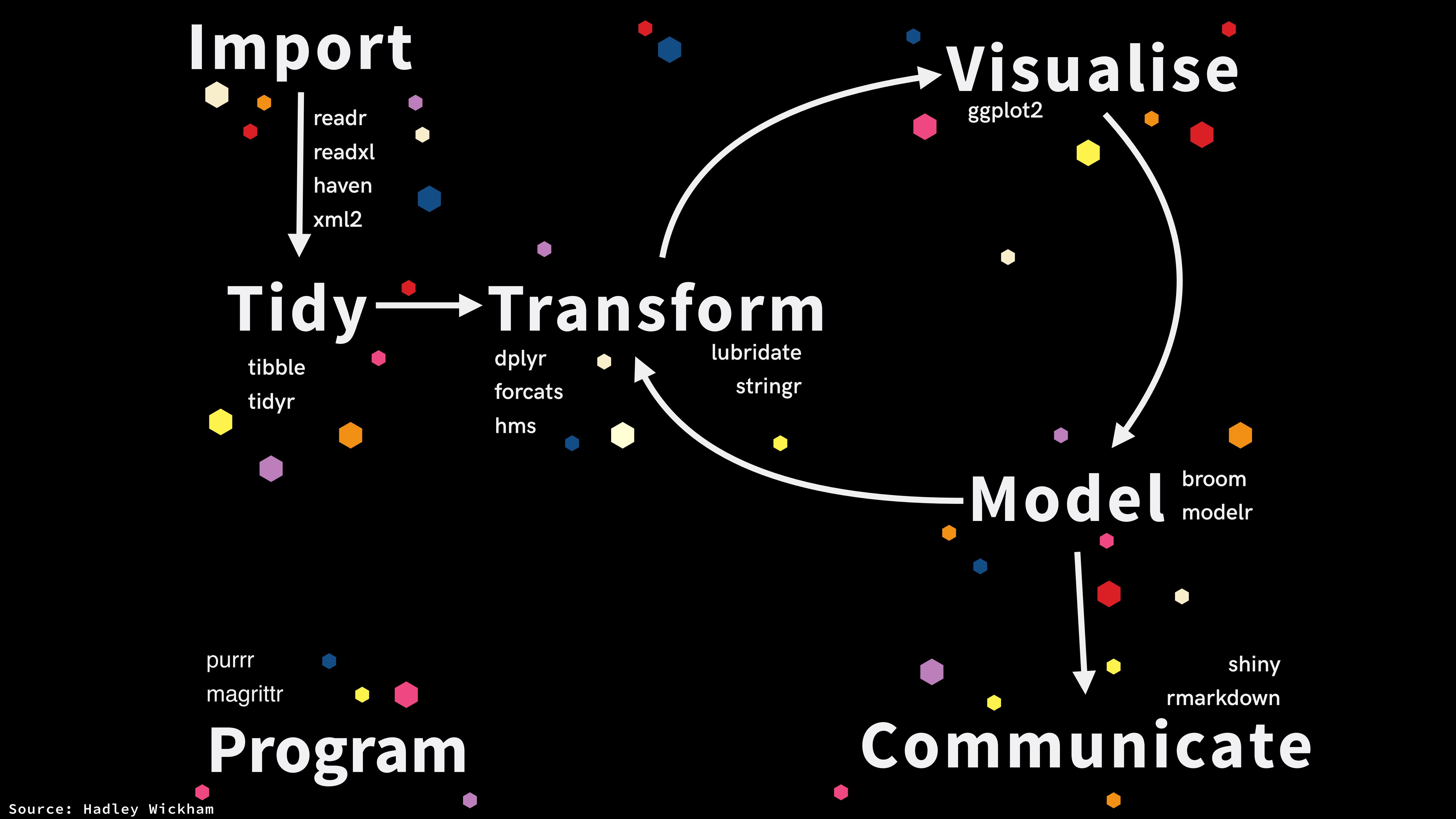
```
  filter(Sepal.Length > 7) %>%
```

```
  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>.



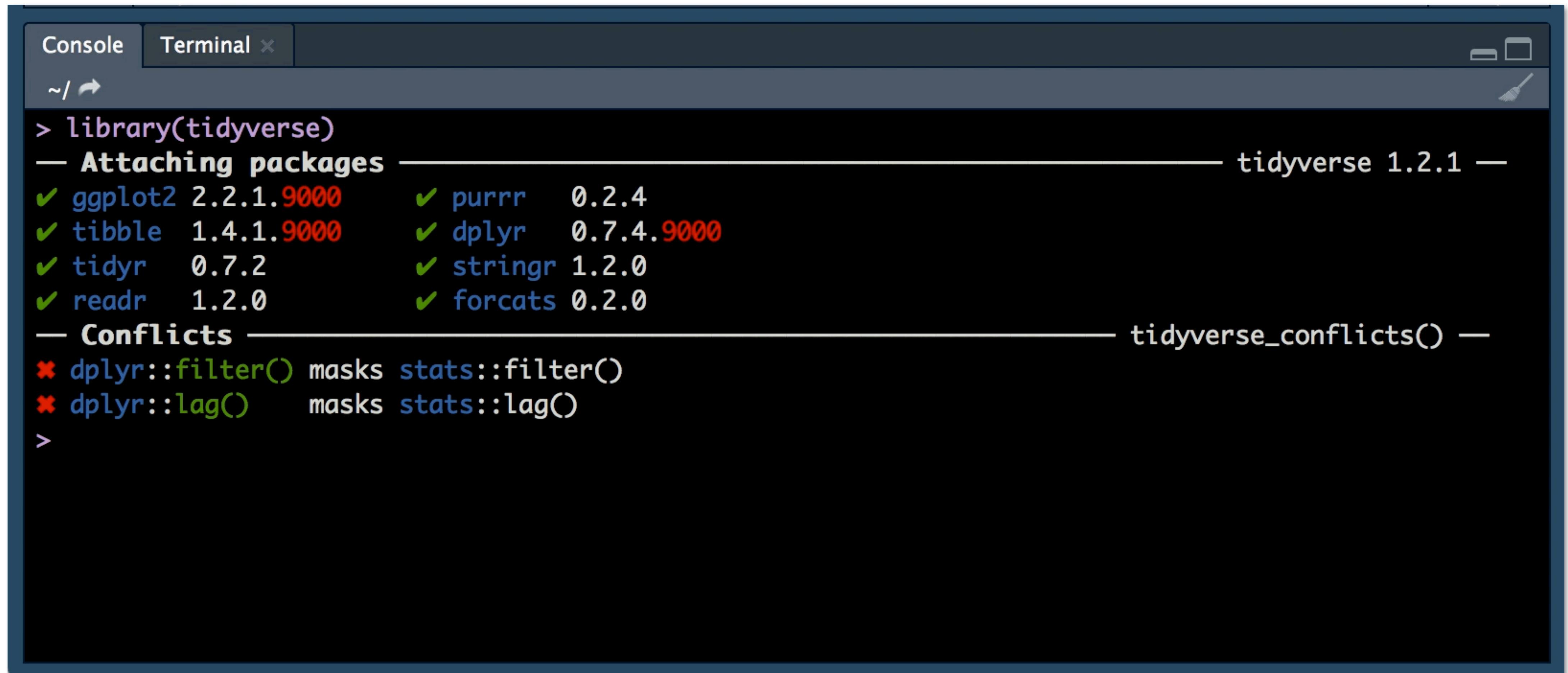
# install.packages("tidyverse")



The screenshot shows an RStudio interface with a dark theme. The top bar has tabs for 'Console' (which is selected) and 'Terminal'. The status bar indicates the current directory is '~ddtx\_demo/'. The main area of the window displays the R command `install.packages("tidyverse")` followed by the output of the package installation process.

```
> 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’, ‘tidyr’, ‘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.*”



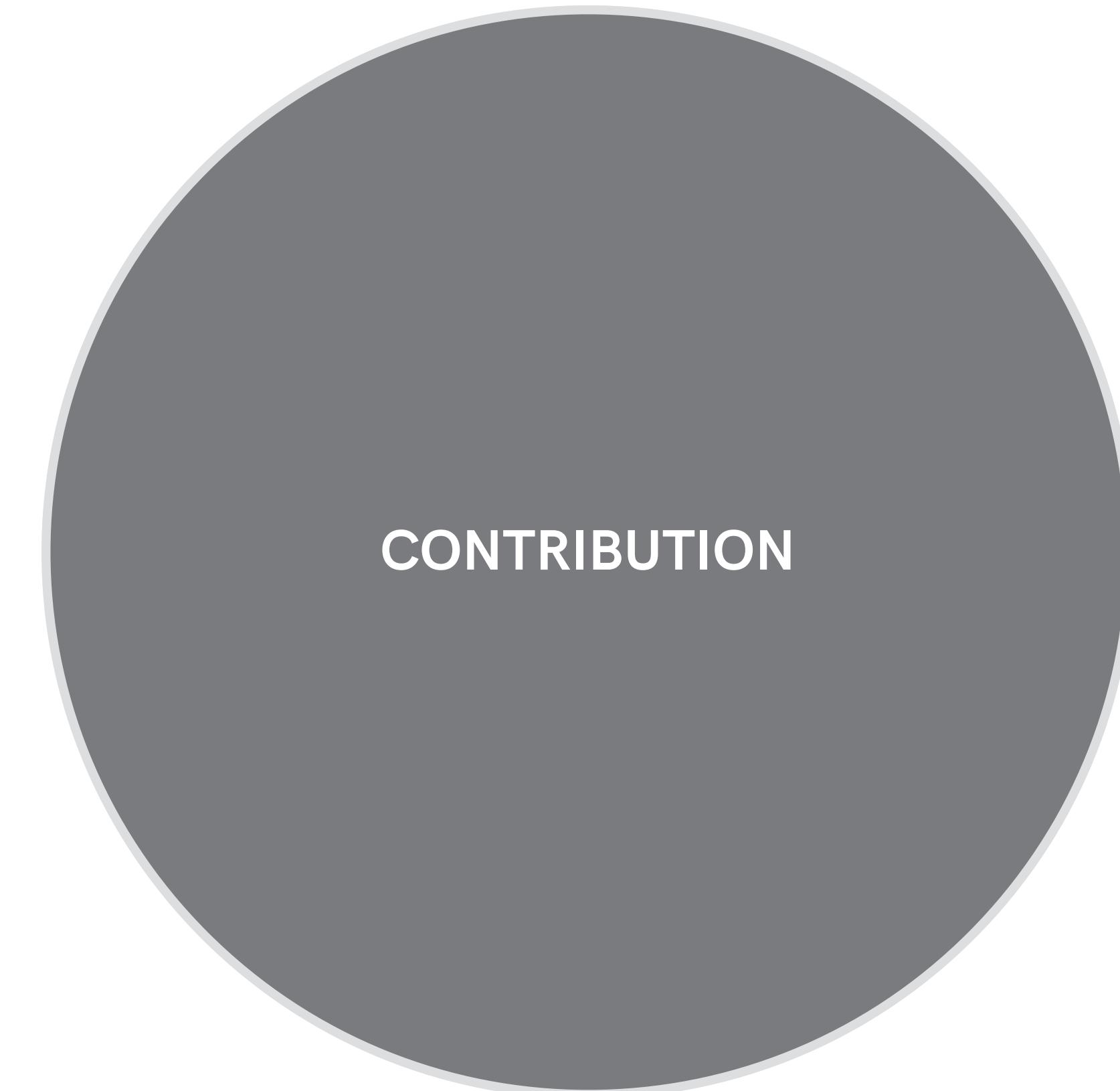
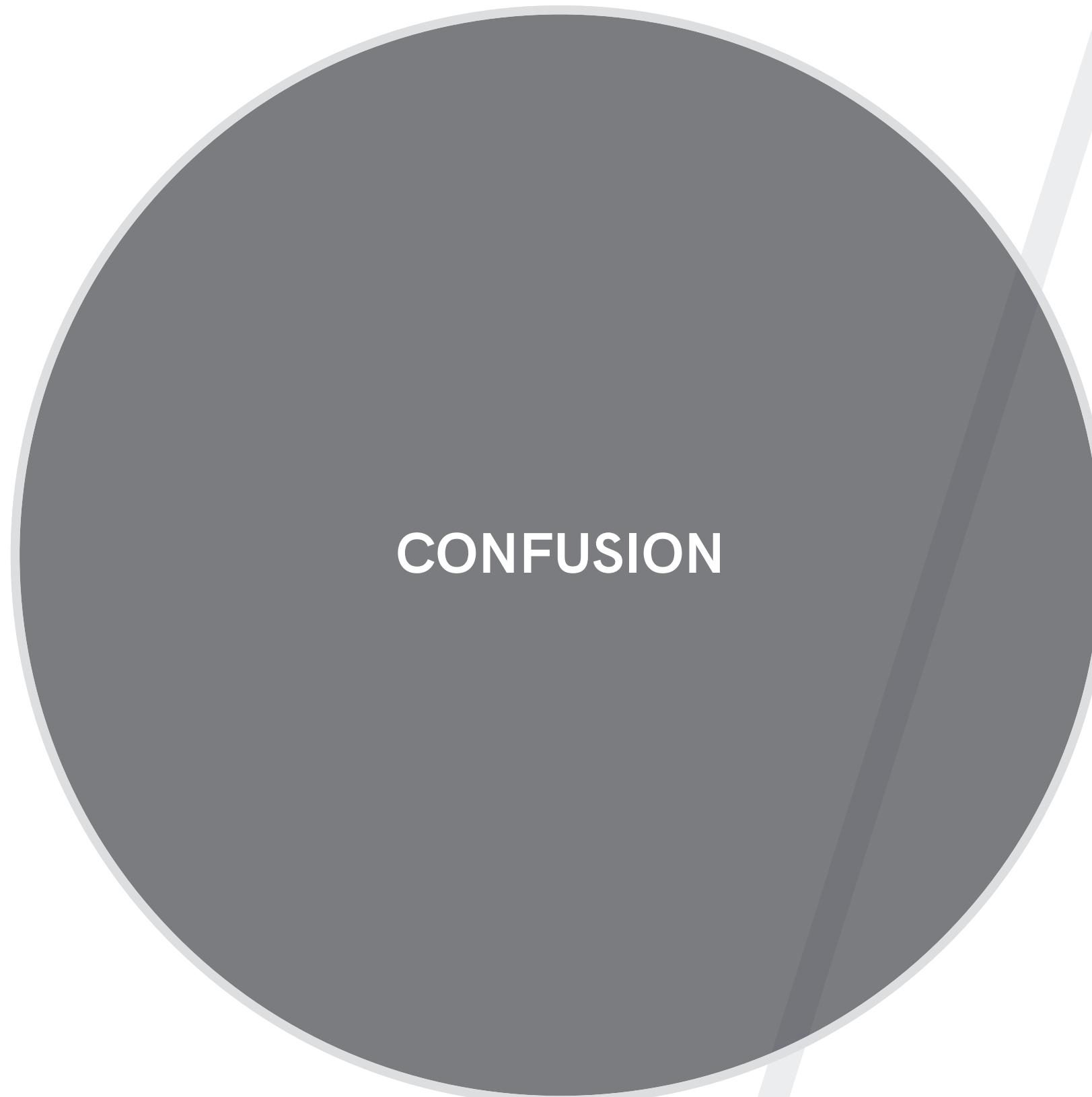
# Contributing to FOSS

## WHAT HOLDS PEOPLE BACK?

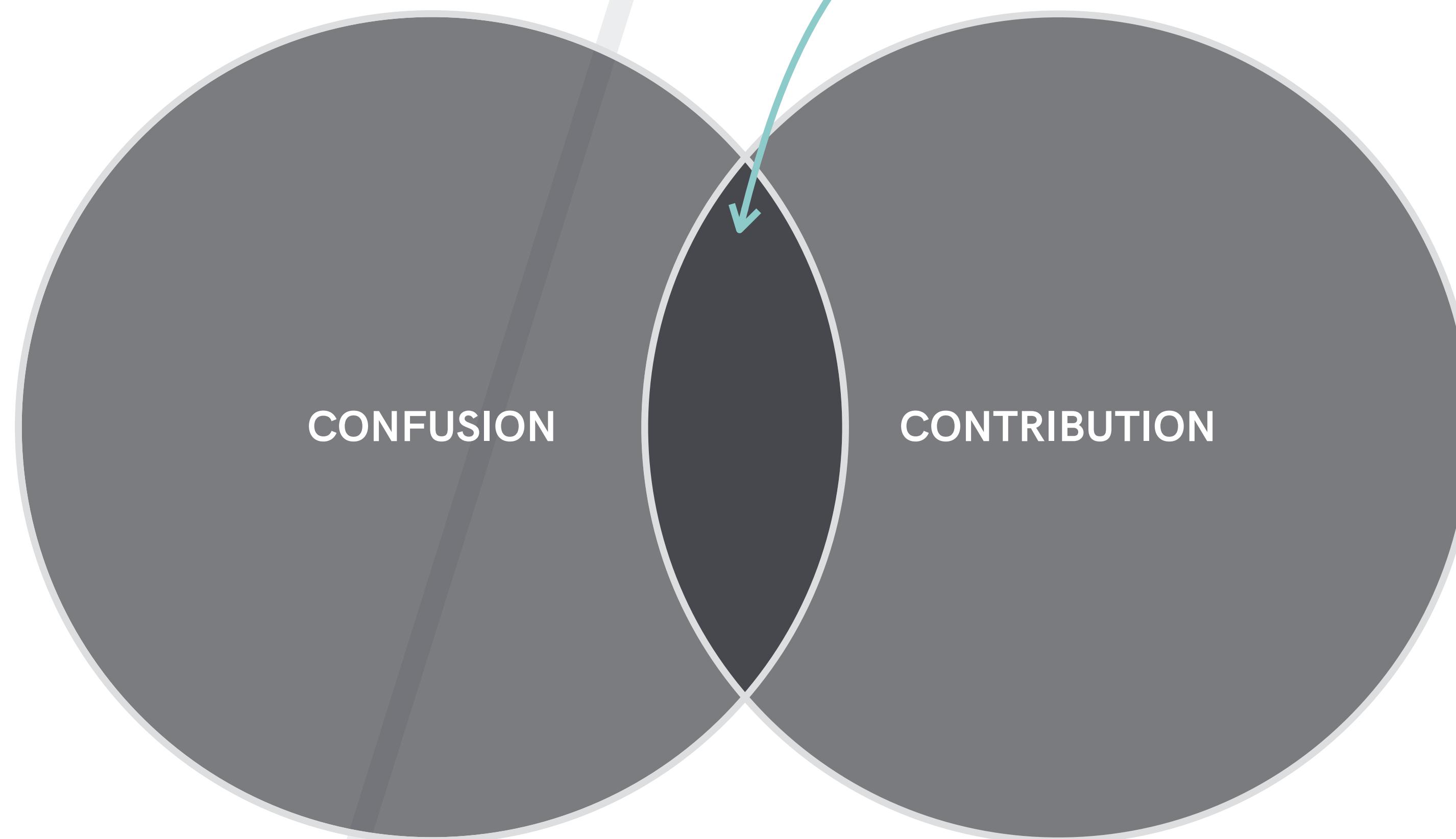
- “*I can't write code.*”
- “*I'm not really good at this.*”
- “*I'd just be a burden.*”
- “*They already have enough people smarter than me.*”



# 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



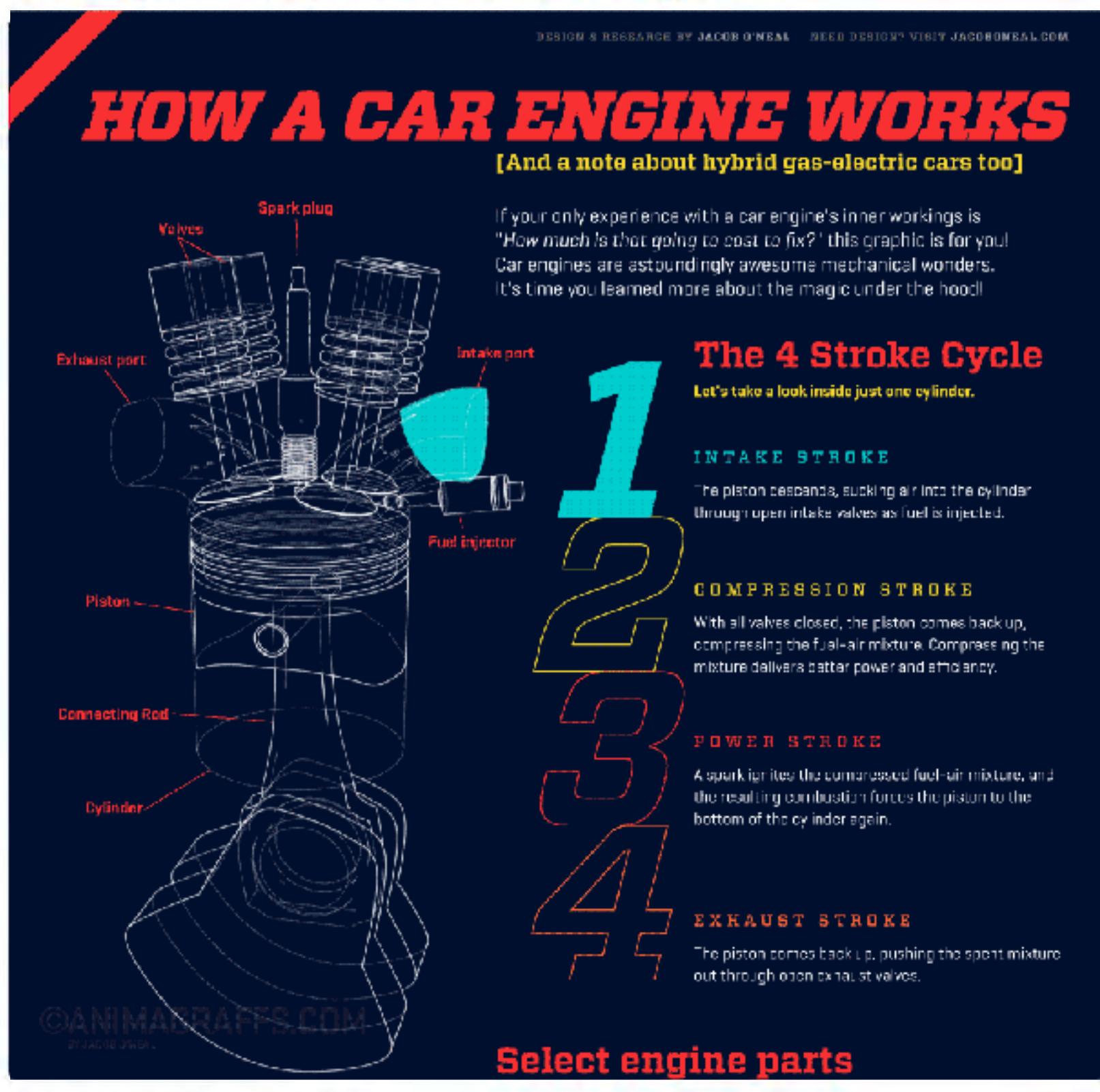
**Mara Averick**  
@dataandme

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

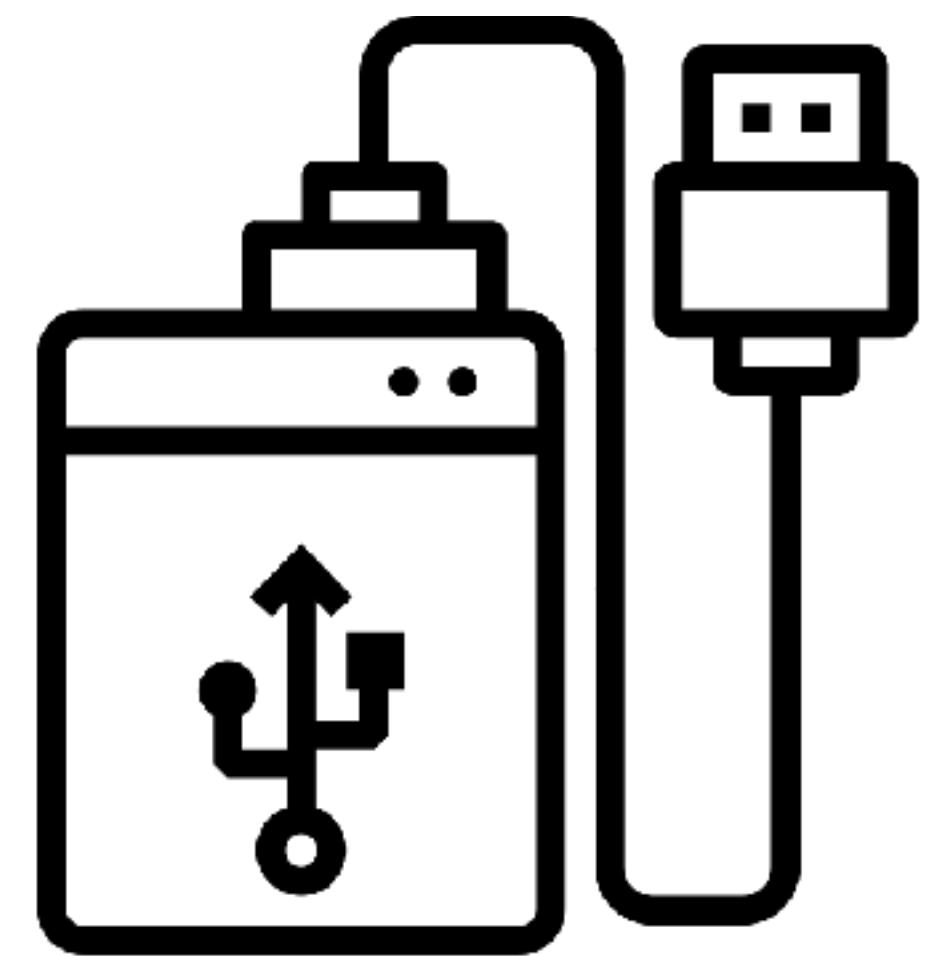
Internet: please disagree 🚗

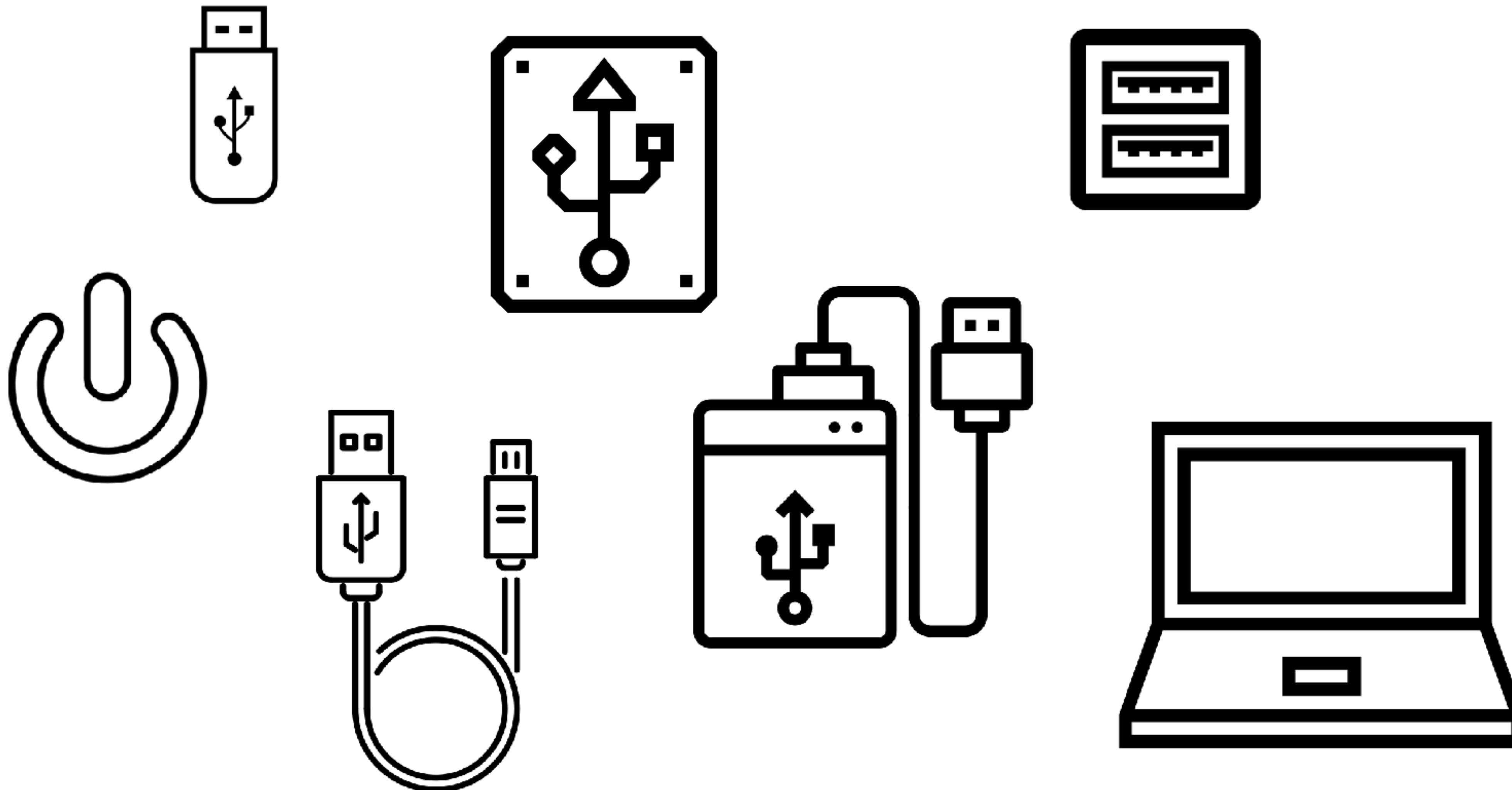


[buff.ly/2iteUyZ](https://buff.ly/2iteUyZ) @animagraffs

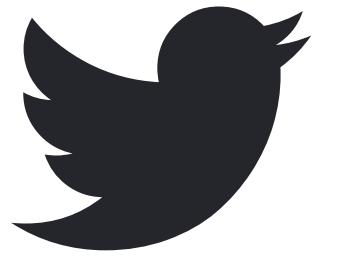


1:21 PM - 18 Nov 2017





# where to ask



Twitter



StackOverflow



RStudio Community

# the magic of reprex



**reproducible  
example**

# 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\*".
- 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

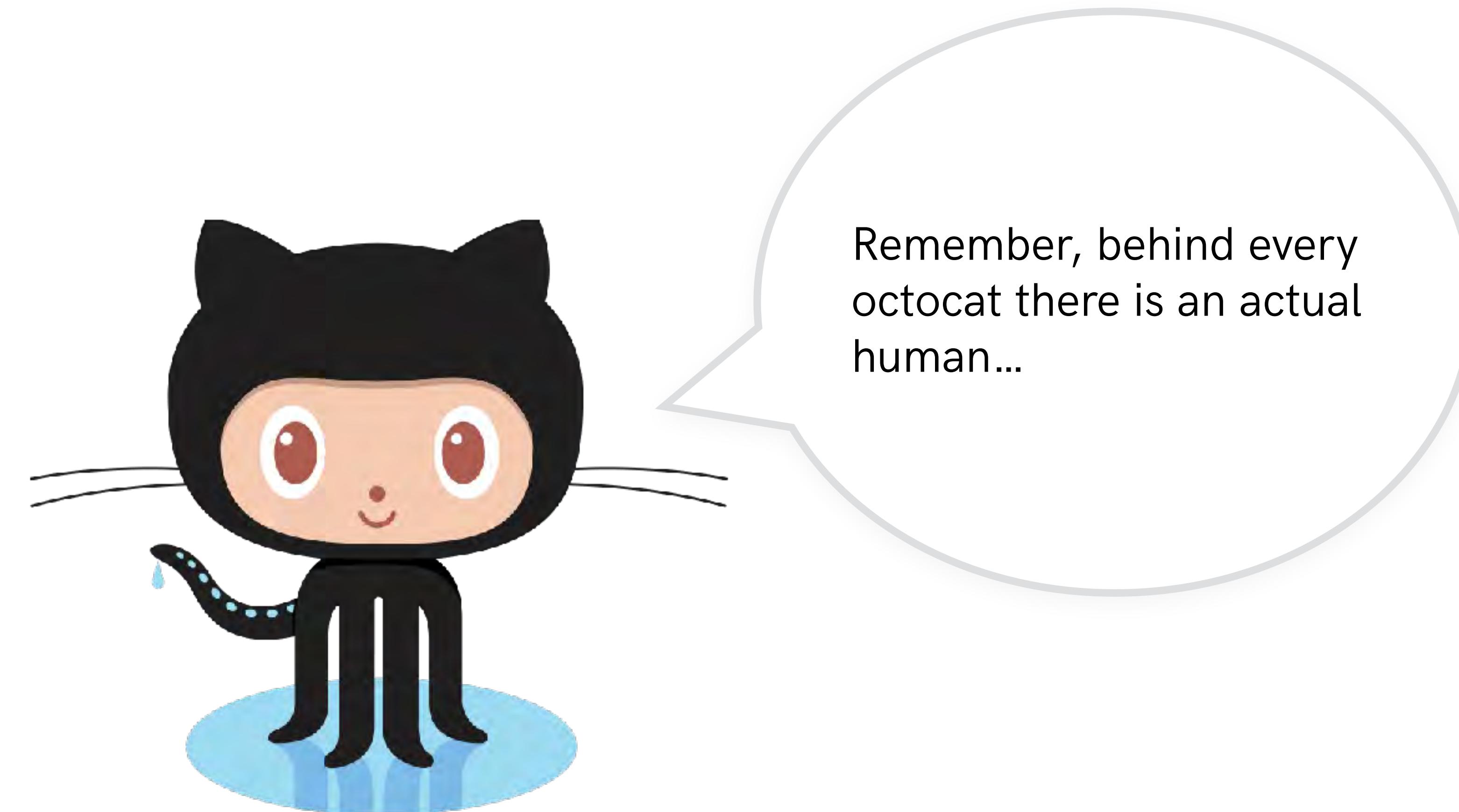


RStudio Community



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*

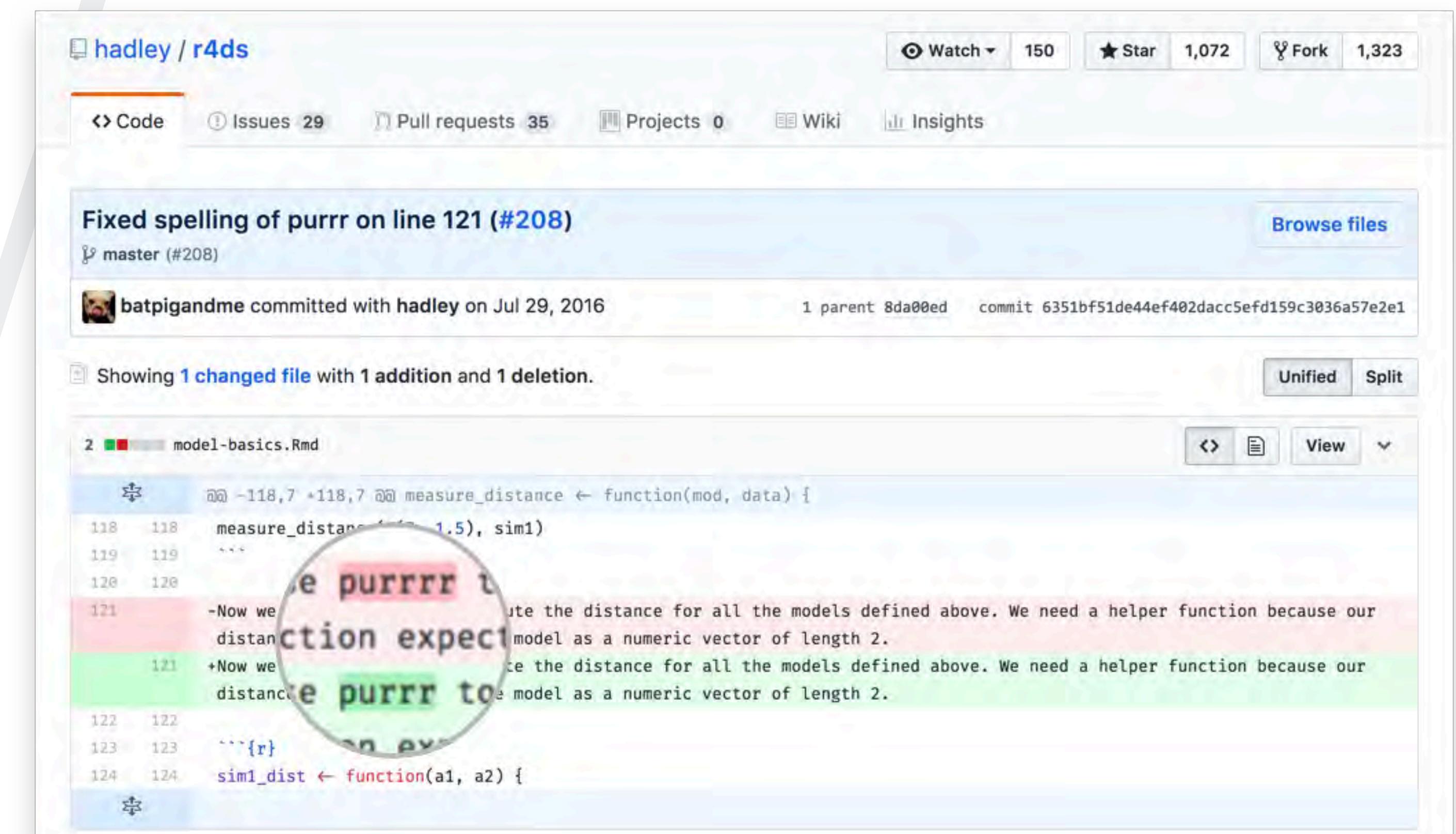
# My first “contribution”

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 a fix for the spelling of `purrr` on line 121. The code changes are shown in the `model-basics.Rmd` file, where line 121 was changed from `-Now we can use purrrr` to `+Now we can use purrr`. The pull request has 150 reviews and 1,072 forks.

```
@@ -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 purrrr to compute the distance for all the models defined above. We need a helper function because our  
 121 +Now we can use purrr to compute the distance for all the models defined above. We need a helper function because our  
 121 distance function expects the model as a numeric vector of length 2.  
 122 122  
 123 123  ``{r}  
 124 124  sim1_dist <- function(a1, a2) {  
 125 125  
 126 126  
 127 127  
 128 128  
 129 129  
 130 130  
 131 131  
 132 132  
 133 133  
 134 134  
 135 135  
 136 136  
 137 137  
 138 138  
 139 139  
 140 140  
 141 141  
 142 142  
 143 143  
 144 144  
 145 145  
 146 146  
 147 147  
 148 148  
 149 149  
 150 150  
 151 151  
 152 152  
 153 153  
 154 154  
 155 155  
 156 156  
 157 157  
 158 158  
 159 159  
 160 160  
 161 161  
 162 162  
 163 163  
 164 164  
 165 165  
 166 166  
 167 167  
 168 168  
 169 169  
 170 170  
 171 171  
 172 172  
 173 173  
 174 174  
 175 175  
 176 176  
 177 177  
 178 178  
 179 179  
 180 180  
 181 181  
 182 182  
 183 183  
 184 184  
 185 185  
 186 186  
 187 187  
 188 188  
 189 189  
 190 190  
 191 191  
 192 192  
 193 193  
 194 194  
 195 195  
 196 196  
 197 197  
 198 198  
 199 199  
 200 200  
 201 201  
 202 202  
 203 203  
 204 204  
 205 205  
 206 206  
 207 207  
 208 208  
 209 209  
 210 210  
 211 211  
 212 212  
 213 213  
 214 214  
 215 215  
 216 216  
 217 217  
 218 218  
 219 219  
 220 220  
 221 221  
 222 222  
 223 223  
 224 224  
 225 225  
 226 226  
 227 227  
 228 228  
 229 229  
 230 230  
 231 231  
 232 232  
 233 233  
 234 234  
 235 235  
 236 236  
 237 237  
 238 238  
 239 239  
 240 240  
 241 241  
 242 242  
 243 243  
 244 244  
 245 245  
 246 246  
 247 247  
 248 248  
 249 249  
 250 250  
 251 251  
 252 252  
 253 253  
 254 254  
 255 255  
 256 256  
 257 257  
 258 258  
 259 259  
 260 260  
 261 261  
 262 262  
 263 263  
 264 264  
 265 265  
 266 266  
 267 267  
 268 268  
 269 269  
 270 270  
 271 271  
 272 272  
 273 273  
 274 274  
 275 275  
 276 276  
 277 277  
 278 278  
 279 279  
 280 280  
 281 281  
 282 282  
 283 283  
 284 284  
 285 285  
 286 286  
 287 287  
 288 288  
 289 289  
 290 290  
 291 291  
 292 292  
 293 293  
 294 294  
 295 295  
 296 296  
 297 297  
 298 298  
 299 299  
 300 300  
 301 301  
 302 302  
 303 303  
 304 304  
 305 305  
 306 306  
 307 307  
 308 308  
 309 309  
 310 310  
 311 311  
 312 312  
 313 313  
 314 314  
 315 315  
 316 316  
 317 317  
 318 318  
 319 319  
 320 320  
 321 321  
 322 322  
 323 323  
 324 324  
 325 325  
 326 326  
 327 327  
 328 328  
 329 329  
 330 330  
 331 331  
 332 332  
 333 333  
 334 334  
 335 335  
 336 336  
 337 337  
 338 338  
 339 339  
 340 340  
 341 341  
 342 342  
 343 343  
 344 344  
 345 345  
 346 346  
 347 347  
 348 348  
 349 349  
 350 350  
 351 351  
 352 352  
 353 353  
 354 354  
 355 355  
 356 356  
 357 357  
 358 358  
 359 359  
 360 360  
 361 361  
 362 362  
 363 363  
 364 364  
 365 365  
 366 366  
 367 367  
 368 368  
 369 369  
 370 370  
 371 371  
 372 372  
 373 373  
 374 374  
 375 375  
 376 376  
 377 377  
 378 378  
 379 379  
 380 380  
 381 381  
 382 382  
 383 383  
 384 384  
 385 385  
 386 386  
 387 387  
 388 388  
 389 389  
 390 390  
 391 391  
 392 392  
 393 393  
 394 394  
 395 395  
 396 396  
 397 397  
 398 398  
 399 399  
 400 400  
 401 401  
 402 402  
 403 403  
 404 404  
 405 405  
 406 406  
 407 407  
 408 408  
 409 409  
 410 410  
 411 411  
 412 412  
 413 413  
 414 414  
 415 415  
 416 416  
 417 417  
 418 418  
 419 419  
 420 420  
 421 421  
 422 422  
 423 423  
 424 424  
 425 425  
 426 426  
 427 427  
 428 428  
 429 429  
 430 430  
 431 431  
 432 432  
 433 433  
 434 434  
 435 435  
 436 436  
 437 437  
 438 438  
 439 439  
 440 440  
 441 441  
 442 442  
 443 443  
 444 444  
 445 445  
 446 446  
 447 447  
 448 448  
 449 449  
 450 450  
 451 451  
 452 452  
 453 453  
 454 454  
 455 455  
 456 456  
 457 457  
 458 458  
 459 459  
 460 460  
 461 461  
 462 462  
 463 463  
 464 464  
 465 465  
 466 466  
 467 467  
 468 468  
 469 469  
 470 470  
 471 471  
 472 472  
 473 473  
 474 474  
 475 475  
 476 476  
 477 477  
 478 478  
 479 479  
 480 480  
 481 481  
 482 482  
 483 483  
 484 484  
 485 485  
 486 486  
 487 487  
 488 488  
 489 489  
 490 490  
 491 491  
 492 492  
 493 493  
 494 494  
 495 495  
 496 496  
 497 497  
 498 498  
 499 499  
 500 500  
 501 501  
 502 502  
 503 503  
 504 504  
 505 505  
 506 506  
 507 507  
 508 508  
 509 509  
 510 510  
 511 511  
 512 512  
 513 513  
 514 514  
 515 515  
 516 516  
 517 517  
 518 518  
 519 519  
 520 520  
 521 521  
 522 522  
 523 523  
 524 524  
 525 525  
 526 526  
 527 527  
 528 528  
 529 529  
 530 530  
 531 531  
 532 532  
 533 533  
 534 534  
 535 535  
 536 536  
 537 537  
 538 538  
 539 539  
 540 540  
 541 541  
 542 542  
 543 543  
 544 544  
 545 545  
 546 546  
 547 547  
 548 548  
 549 549  
 550 550  
 551 551  
 552 552  
 553 553  
 554 554  
 555 555  
 556 556  
 557 557  
 558 558  
 559 559  
 560 560  
 561 561  
 562 562  
 563 563  
 564 564  
 565 565  
 566 566  
 567 567  
 568 568  
 569 569  
 570 570  
 571 571  
 572 572  
 573 573  
 574 574  
 575 575  
 576 576  
 577 577  
 578 578  
 579 579  
 580 580  
 581 581  
 582 582  
 583 583  
 584 584  
 585 585  
 586 586  
 587 587  
 588 588  
 589 589  
 590 590  
 591 591  
 592 592  
 593 593  
 594 594  
 595 595  
 596 596  
 597 597  
 598 598  
 599 599  
 600 600  
 601 601  
 602 602  
 603 603  
 604 604  
 605 605  
 606 606  
 607 607  
 608 608  
 609 609  
 610 610  
 611 611  
 612 612  
 613 613  
 614 614  
 615 615  
 616 616  
 617 617  
 618 618  
 619 619  
 620 620  
 621 621  
 622 622  
 623 623  
 624 624  
 625 625  
 626 626  
 627 627  
 628 628  
 629 629  
 630 630  
 631 631  
 632 632  
 633 633  
 634 634  
 635 635  
 636 636  
 637 637  
 638 638  
 639 639  
 640 640  
 641 641  
 642 642  
 643 643  
 644 644  
 645 645  
 646 646  
 647 647  
 648 648  
 649 649  
 650 650  
 651 651  
 652 652  
 653 653  
 654 654  
 655 655  
 656 656  
 657 657  
 658 658  
 659 659  
 660 660  
 661 661  
 662 662  
 663 663  
 664 664  
 665 665  
 666 666  
 667 667  
 668 668  
 669 669  
 670 670  
 671 671  
 672 672  
 673 673  
 674 674  
 675 675  
 676 676  
 677 677  
 678 678  
 679 679  
 680 680  
 681 681  
 682 682  
 683 683  
 684 684  
 685 685  
 686 686  
 687 687  
 688 688  
 689 689  
 690 690  
 691 691  
 692 692  
 693 693  
 694 694  
 695 695  
 696 696  
 697 697  
 698 698  
 699 699  
 700 700  
 701 701  
 702 702  
 703 703  
 704 704  
 705 705  
 706 706  
 707 707  
 708 708  
 709 709  
 710 710  
 711 711  
 712 712  
 713 713  
 714 714  
 715 715  
 716 716  
 717 717  
 718 718  
 719 719  
 720 720  
 721 721  
 722 722  
 723 723  
 724 724  
 725 725  
 726 726  
 727 727  
 728 728  
 729 729  
 730 730  
 731 731  
 732 732  
 733 733  
 734 734  
 735 735  
 736 736  
 737 737  
 738 738  
 739 739  
 740 740  
 741 741  
 742 742  
 743 743  
 744 744  
 745 745  
 746 746  
 747 747  
 748 748  
 749 749  
 750 750  
 751 751  
 752 752  
 753 753  
 754 754  
 755 755  
 756 756  
 757 757  
 758 758  
 759 759  
 760 760  
 761 761  
 762 762  
 763 763  
 764 764  
 765 765  
 766 766  
 767 767  
 768 768  
 769 769  
 770 770  
 771 771  
 772 772  
 773 773  
 774 774  
 775 775  
 776 776  
 777 777  
 778 778  
 779 779  
 780 780  
 781 781  
 782 782  
 783 783  
 784 784  
 785 785  
 786 786  
 787 787  
 788 788  
 789 789  
 790 790  
 791 791  
 792 792  
 793 793  
 794 794  
 795 795  
 796 796  
 797 797  
 798 798  
 799 799  
 800 800  
 801 801  
 802 802  
 803 803  
 804 804  
 805 805  
 806 806  
 807 807  
 808 808  
 809 809  
 810 810  
 811 811  
 812 812  
 813 813  
 814 814  
 815 815  
 816 816  
 817 817  
 818 818  
 819 819  
 820 820  
 821 821  
 822 822  
 823 823  
 824 824  
 825 825  
 826 826  
 827 827  
 828 828  
 829 829  
 830 830  
 831 831  
 832 832  
 833 833  
 834 834  
 835 835  
 836 836  
 837 837  
 838 838  
 839 839  
 840 840  
 841 841  
 842 842  
 843 843  
 844 844  
 845 8
```

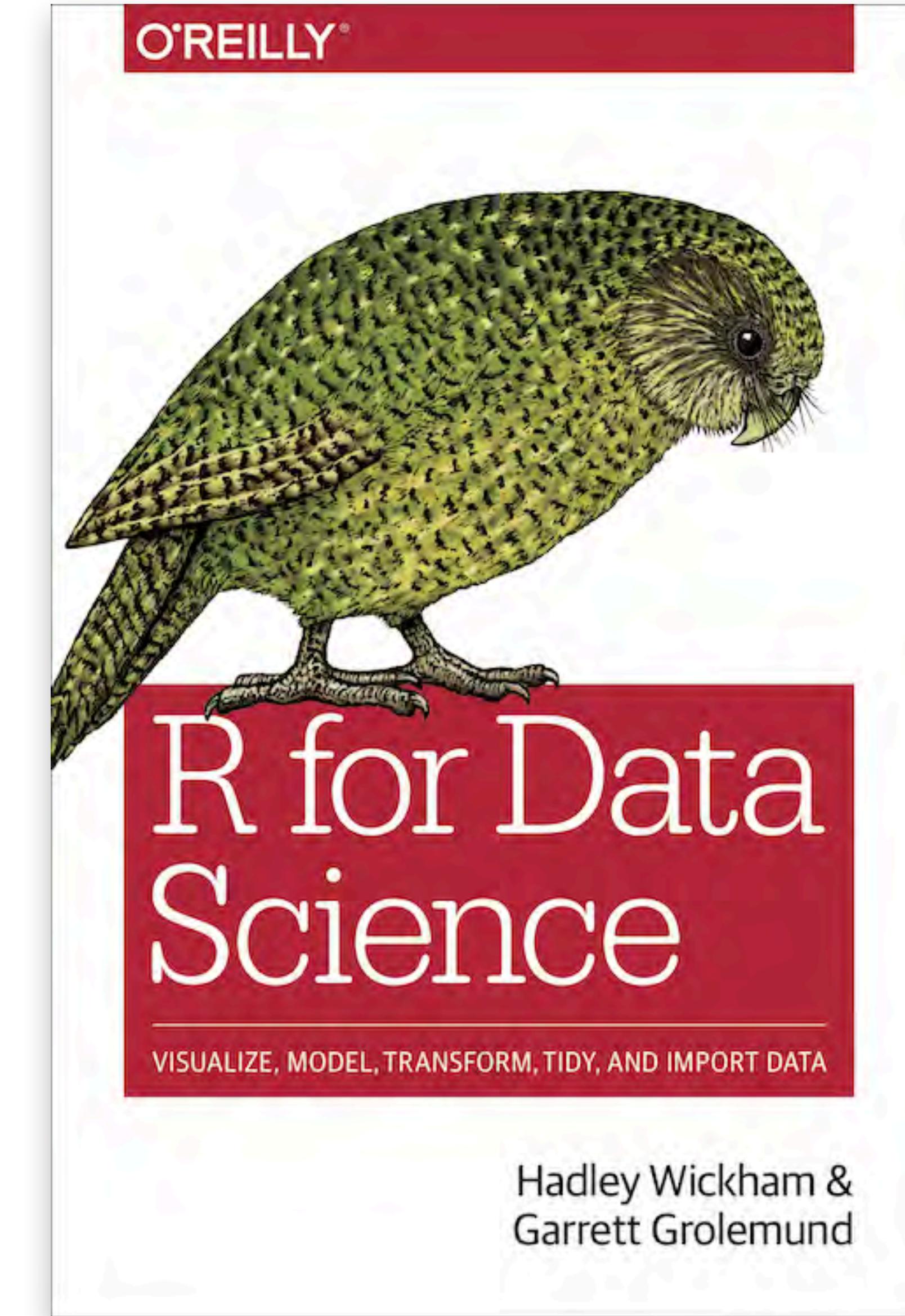
# My first “contribution”

46



# My first “contribution”

46



# every contribution counts...

The screenshot shows the Tidyverse website's "Acknowledgments" page. At the top, there is a navigation bar with links for "Tidyverse", "Packages", "Articles" (which is highlighted in white), "Learn", "Help", and "Contribute". Below the navigation bar, the title "Acknowledgments" is displayed. A large block of text follows, listing numerous GitHub handles of 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.

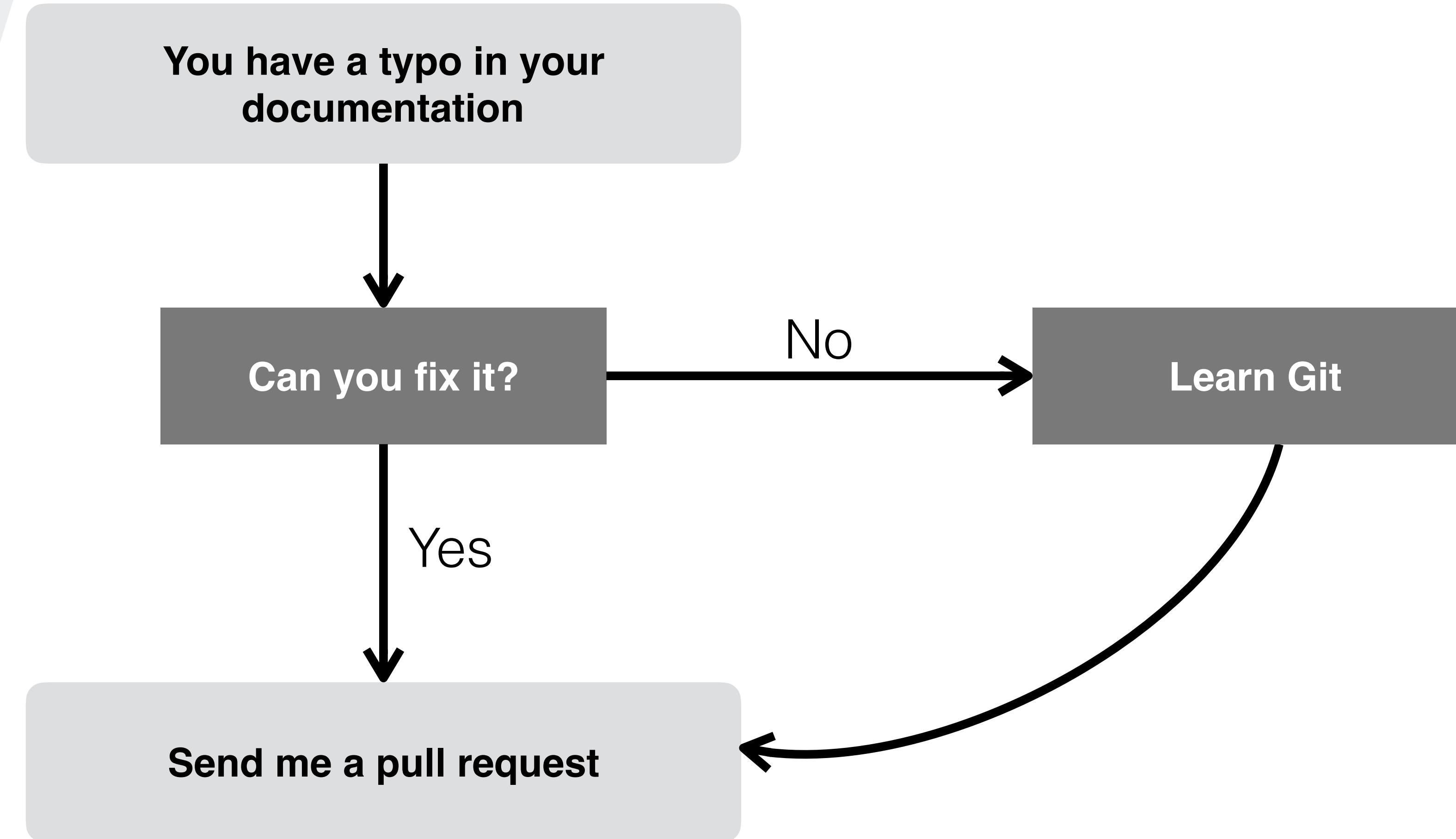
We received issues, pull requests, and comments from 109 people since tibble 1.2.0. Thanks to everyone: @alexandersson, @adnbps, @AkhilNairAmey, @alexhallam, @alibat, @amjuzl, @AndreMikulec, @andrewjpfeiffer, @ashiklm, @atribe, @bapfeld, @barnettjacob, @behrman, @BillDunlap, @BruceZhaoR, @cassiusoat,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

PULL REQUESTS

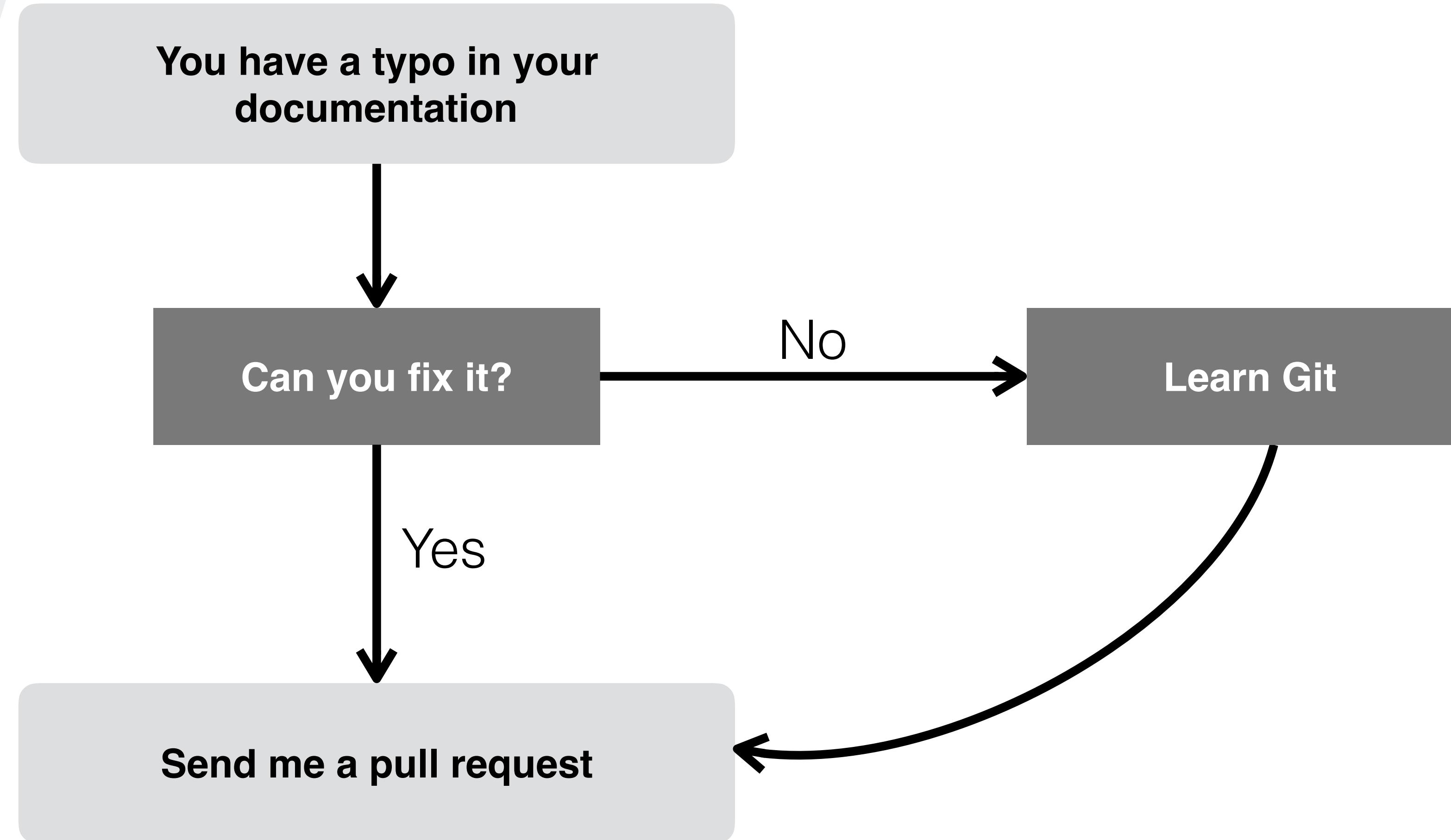
ISSUES

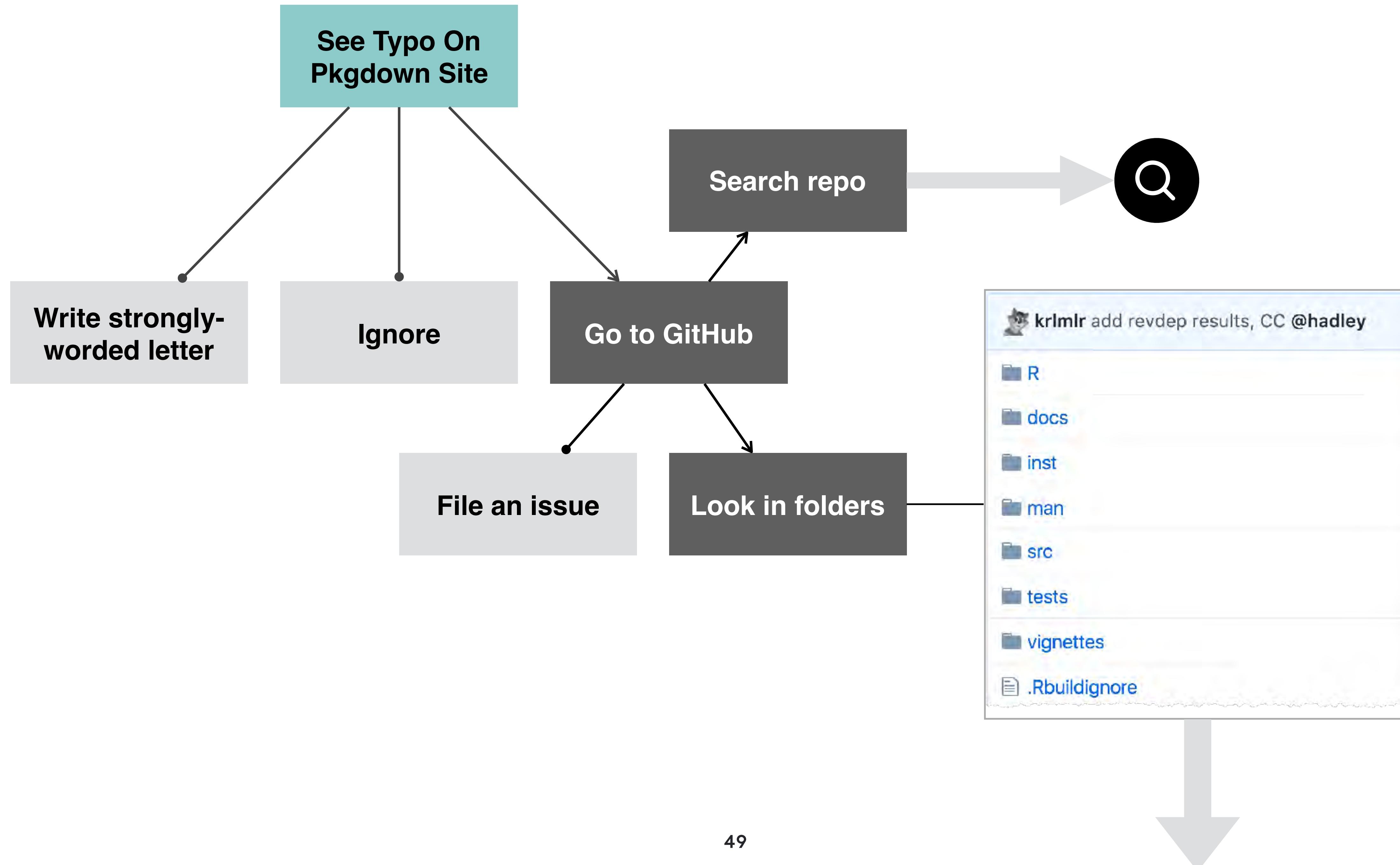
COMMENTS

# tpyos

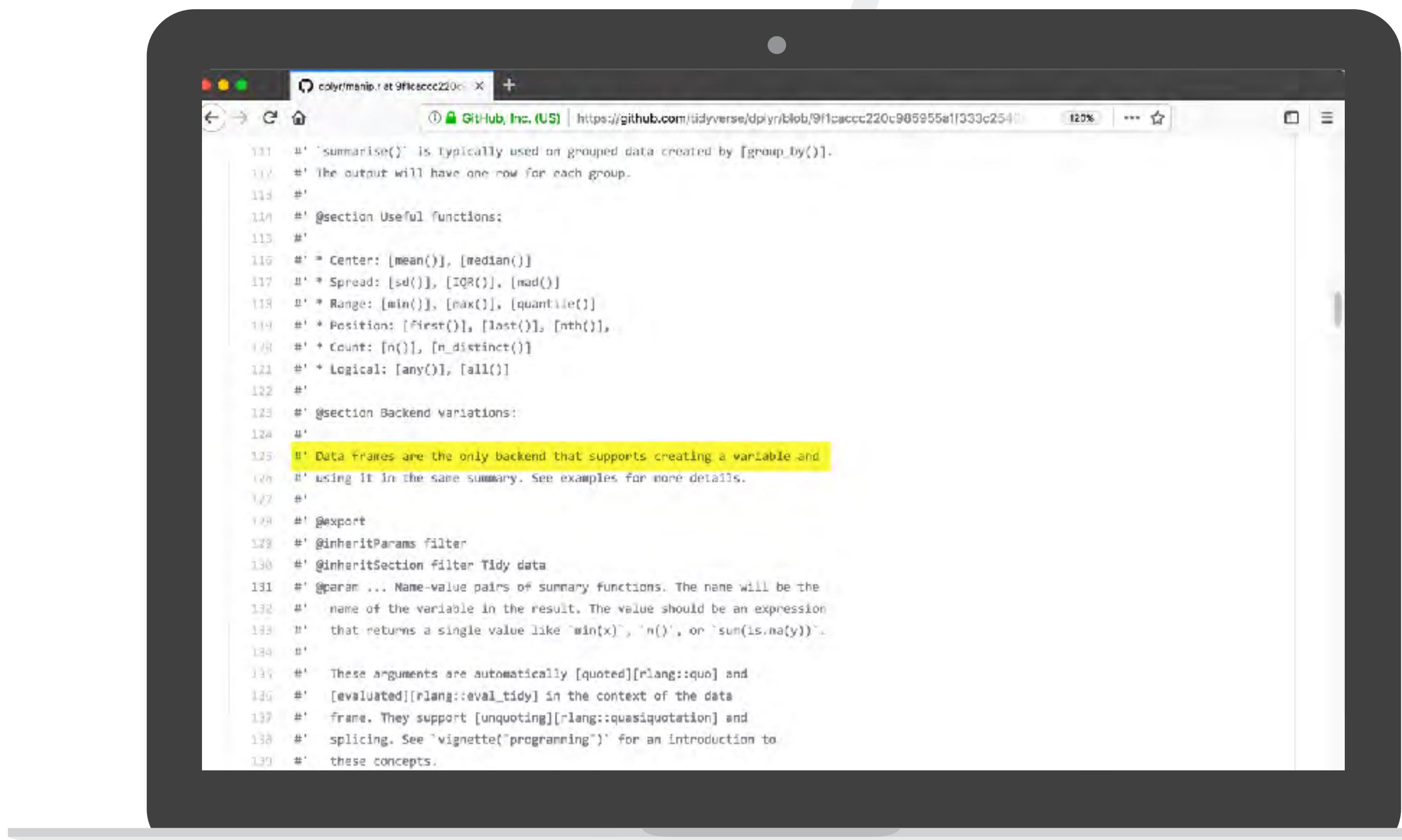
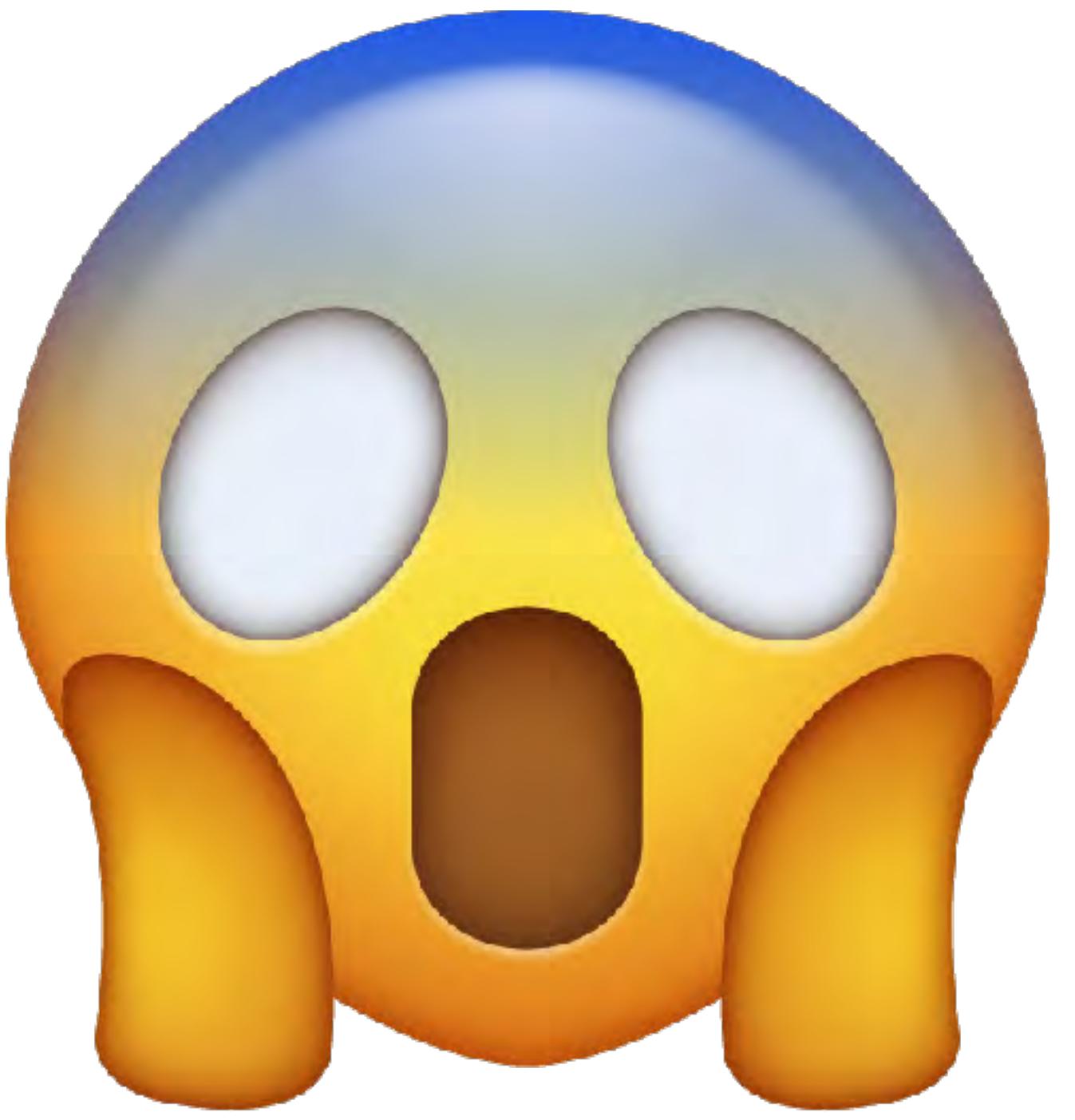


# typos





# Source code?!



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 Mac OS X style. The code is highlighted in yellow for specific sections, such as the documentation comments and the `@export` annotation. The URL in the address bar is <https://github.com/tidyverse/dplyr/blob/9f1ccccc220c905955a1f333c2540d0r/man/man/summarise.R>.

```
131 #' `summarise()` is typically used on grouped data created by `group_by()`.  
132 #' The output will have one row for each group.  
133 #'  
134 #' @section Useful functions:  
135 #'  
136 #' * Center: [mean()], [median()]  
137 #' * Spreads: [sd()], [IQR()], [mad()]  
138 #' * Range: [min()], [max()], [quantile()]  
139 #' * Position: [first()], [last()], [nth()],  
140 #' * Count: [n()], [n_distinct()]  
141 #' * Logical: [any()], [all()]  
142 #'  
143 #' @section Backend Variations:  
144 #'  
145 #' Data frames are the only backend that supports creating a variable and  
146 #' using it in the same summary. See examples for more details.  
147 #'  
148 #' @export  
149 #' @inheritParams filter  
150 #' @inheritSection filter Tidy data  
151 #' @param ... Name-value pairs of summary functions. The name will be the  
152 #' name of the variable in the result. The value should be an expression  
153 #' that returns a single value like `min(x)`, `n()`, or `sum(is.na(y))`.  
154 #'  
155 #' These arguments are automatically [quoted][rlang::quo] and  
156 #' [evaluated][rlang::eval_tidy] in the context of the data  
157 #' frame. They support [unquoting][rlang::quasiquotation] and  
158 #' splicing. See `vignette("programming")` for an introduction to  
159 #' these concepts.
```

# Source code?!

A screenshot of a Mac OS X browser window displaying a GitHub page. The URL in the address bar is https://github.com/tidyverse/dplyr/blob/9f1ccccc220c905955a1f333c2540. The page content shows R code for the `summarise` function. A specific line of code 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.`. The code is numbered from 111 to 139.

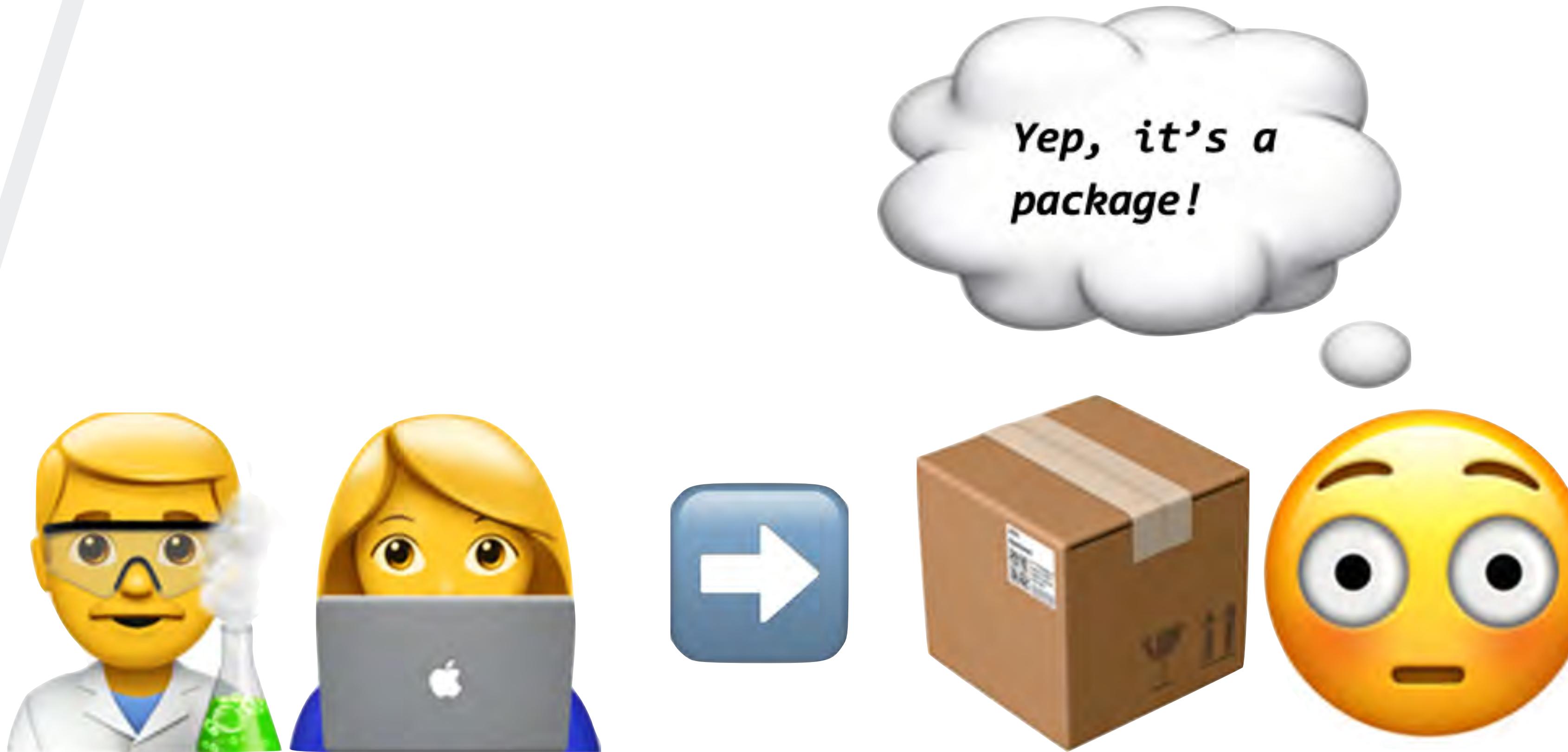
```
111 ## `summarise()` is typically used on grouped data created by `group_by()`.  
112 #' The output will have one row for each group.  
113 #'  
114 #' @section Useful functions:  
115 #'  
116 #' * Center: [mean()], [median()]  
117 #' * Spreads: [sd()], [IQR()], [mad()]  
118 #' * Range: [min()], [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.
```



# 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 highlighted with Highly

# BABY STEPS WITH MORE PACKAGES



**ROXYGEN2**

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



**DEVTOOLS**

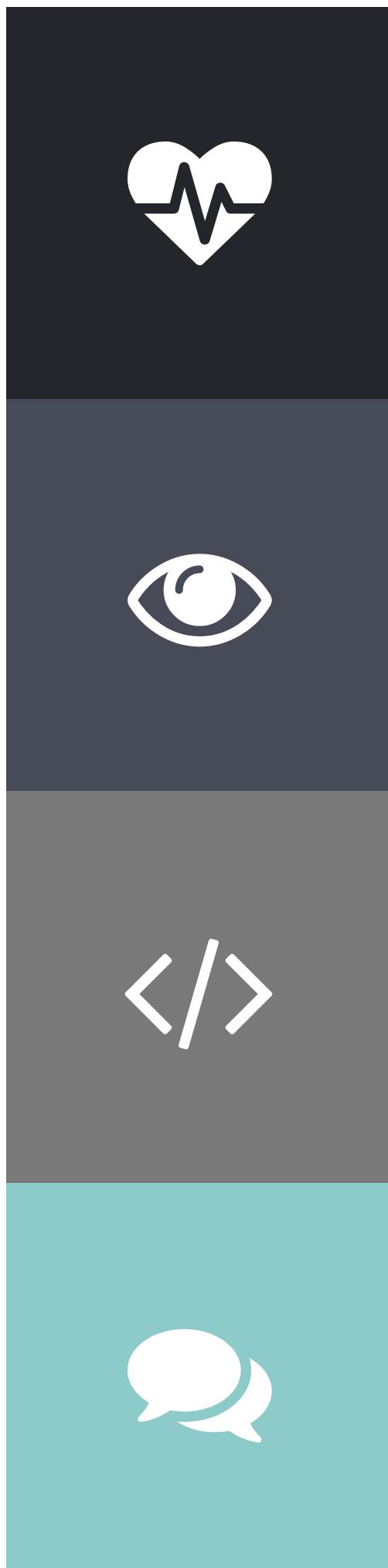
makes package development easier by  
providing R functions that simplify  
common tasks



**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

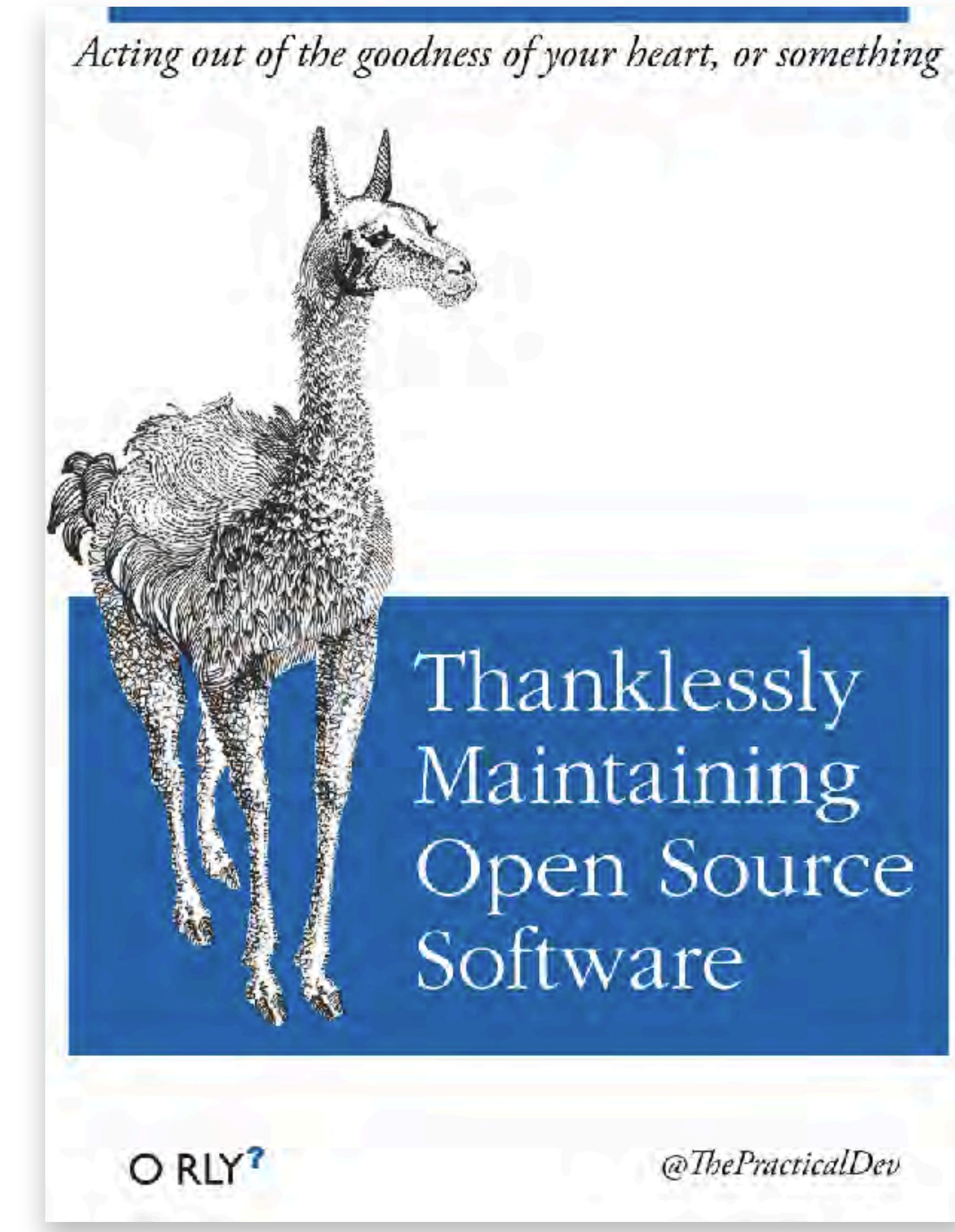
DISCUSS YOUR IDEAS

# leaRn out loud



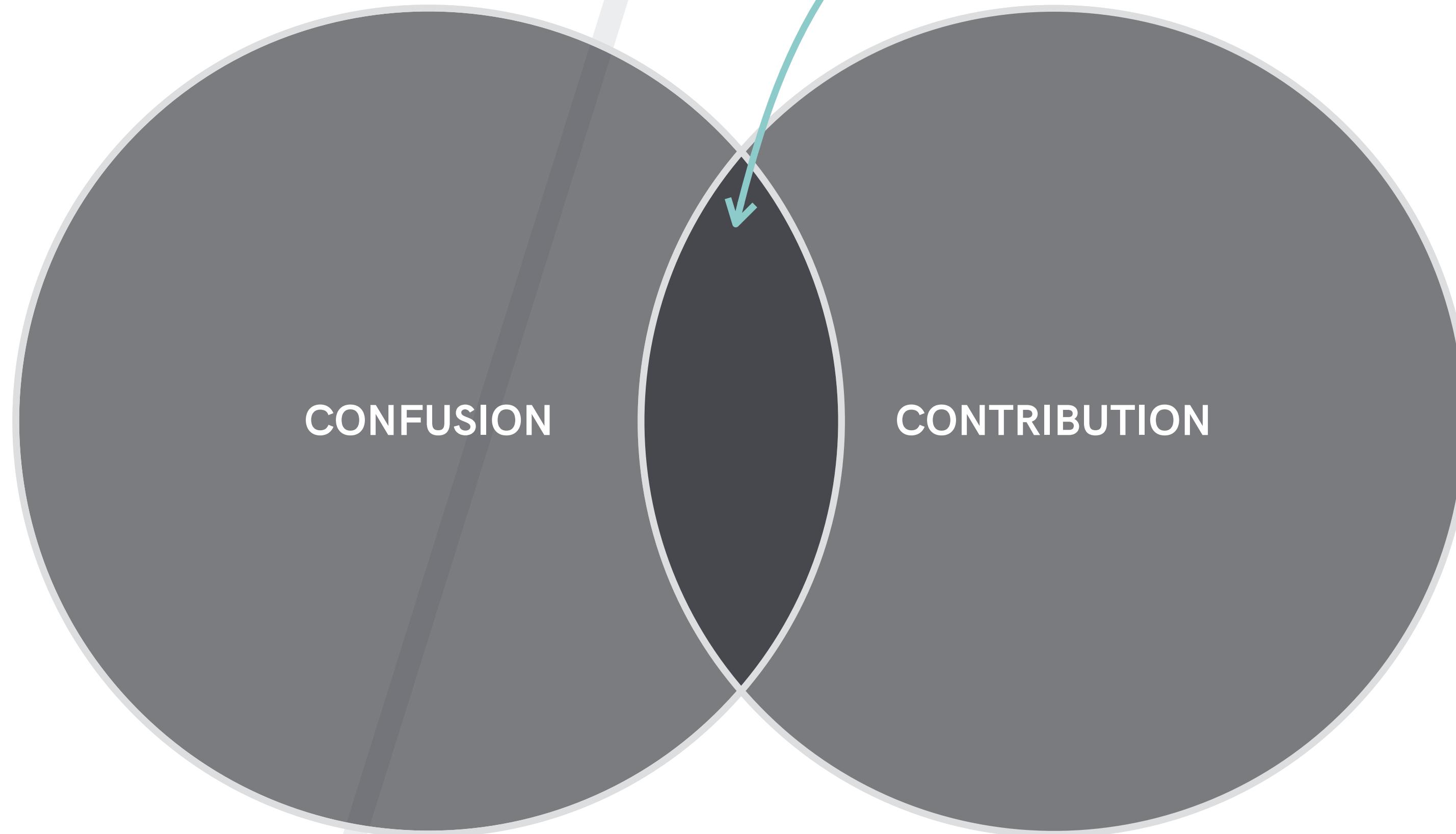
OMG, I just learned a thing!

# give back



# embrace it...

58



# Thank You

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



# Thank You

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

