Designing data science

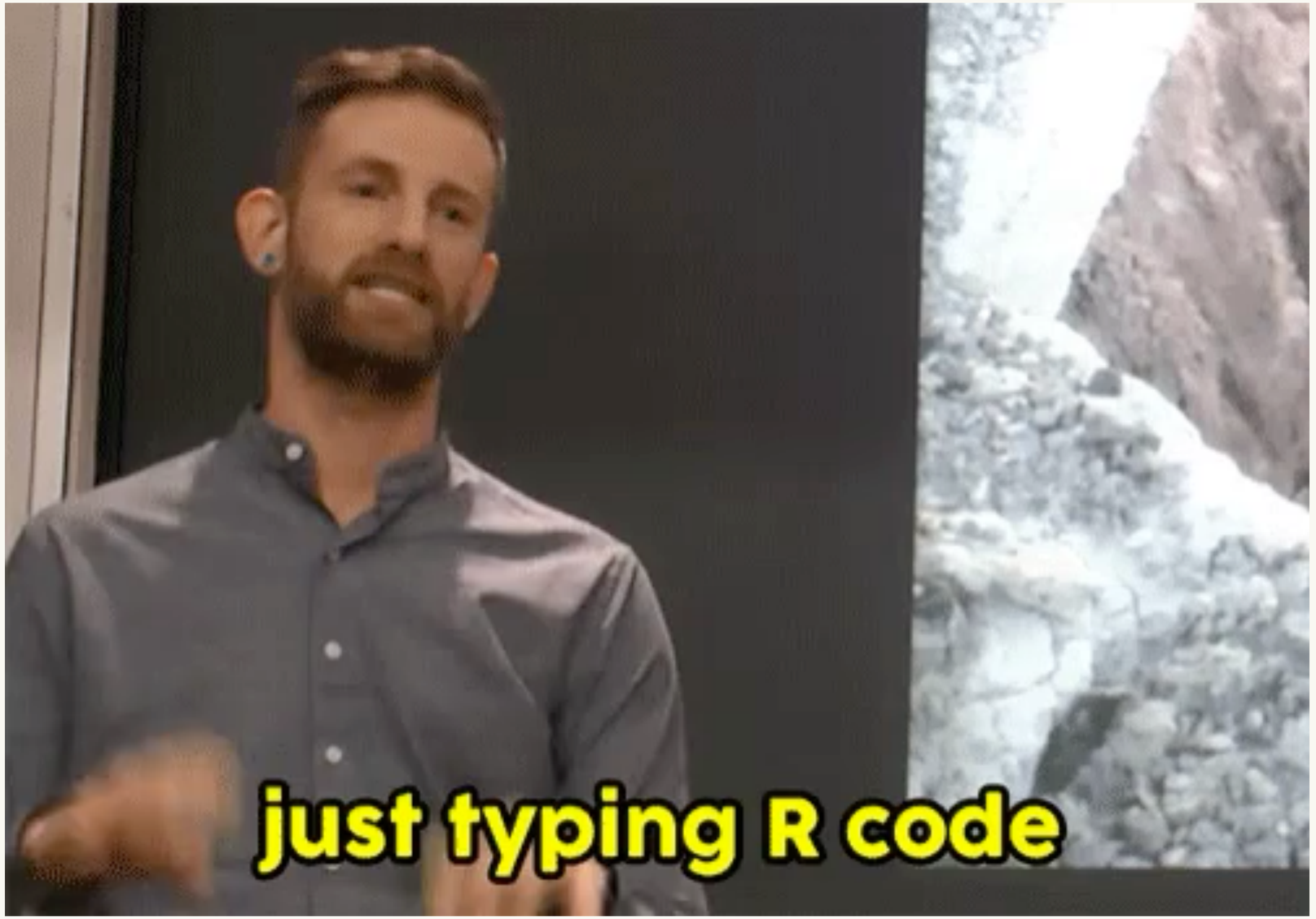
November 2019

Hadley Wickham
@hadleywickham
Chief Scientist, **RStudio**

What do I do?

$$\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

https://unsplash.com/photos/8HyrGTYPQ68 — Eric Muhr


just typing R code

We wanted users to be able to begin in an interactive environment, where they did not consciously think of themselves as programming. Then as their needs became clearer and their sophistication increased, they should be able to **slide gradually into programming**, when the language and system aspects would become more important.

— John Chambers, “Stages in the Evolution of S”

Import

readr
readxl
haven
xml2

Tidy

tibble
tidyr

Transform

dplyr
forcats
hms
lubridate
stringr

Visualise

ggplot2

Model

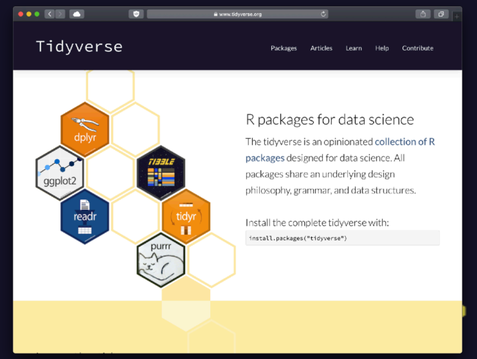
tidymodels

Program

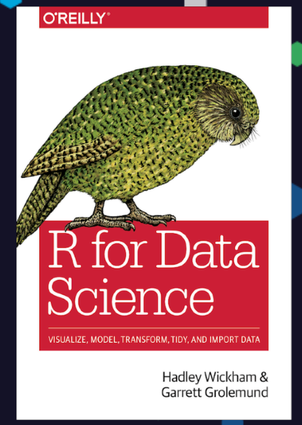
purrr
magrittr

Communicate

shiny
rmarkdown



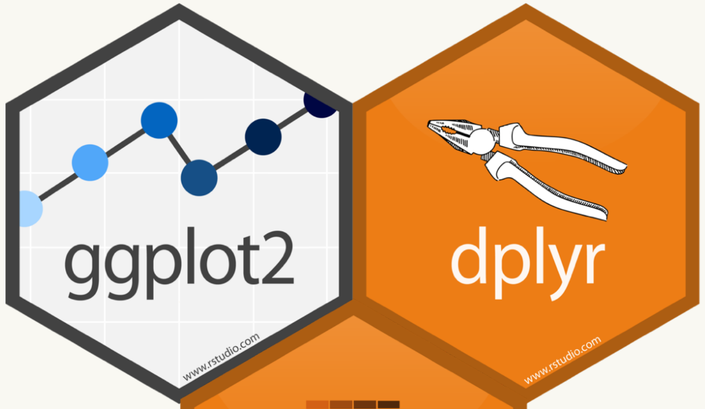
tidyverse.org

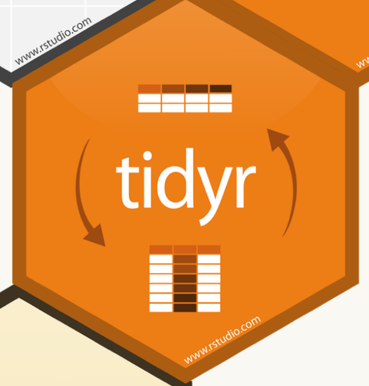


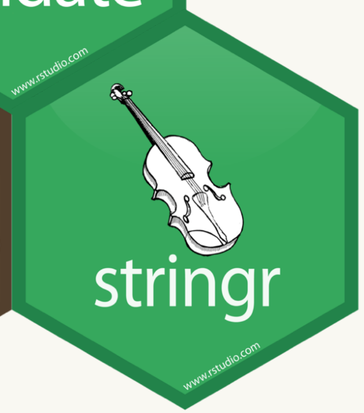
r4ds.had.co.nz

Design

Software engineering







What is design?

https://99percentinvisible.org/article/norman-doors

Designerly ways of knowing by Nigel Cross

Education in any of these cultures entails the following three aspects:

- the transmission of **knowledge** about a phenomenon of study
- a training in the appropriate **methods** of enquiry
- an initiation into the **belief systems** and values of the culture

Design studies, 1982, 3(4):221–227

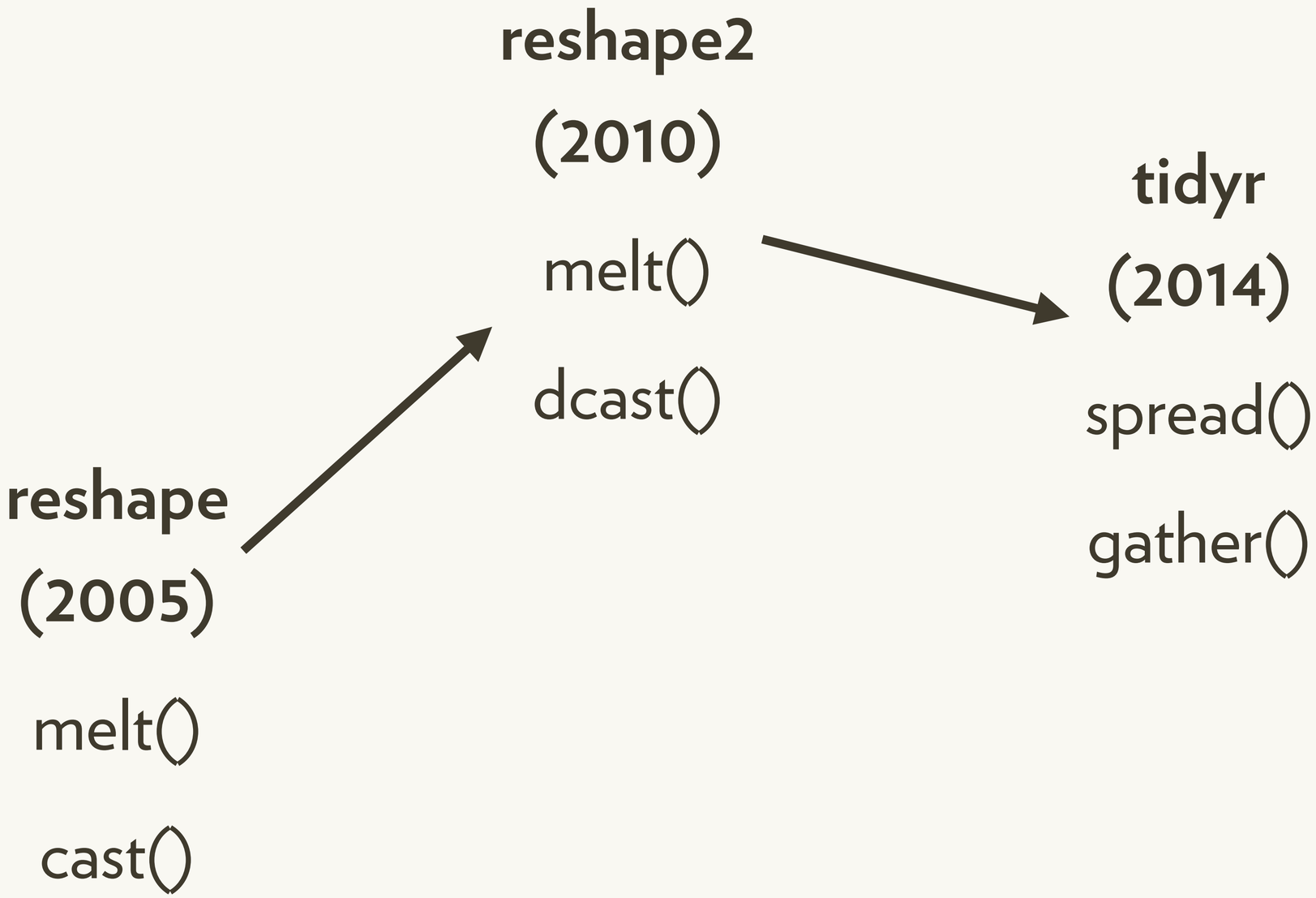
	Phenomenon of study	Values	Appropriate methods
Humanities	human experience	subjectivity, imagination, commitment, and a concern for 'justice'	analogy, metaphor, criticism, evaluation
Sciences	the natural world	objectivity, rationality, neutrality, and a concern for 'Truth'	controlled experiment, classification, analysis
Design	the constructed world	practicality, ingenuity, empathy, and a concern for 'appropriateness'	modelling, pattern-formation, synthesis

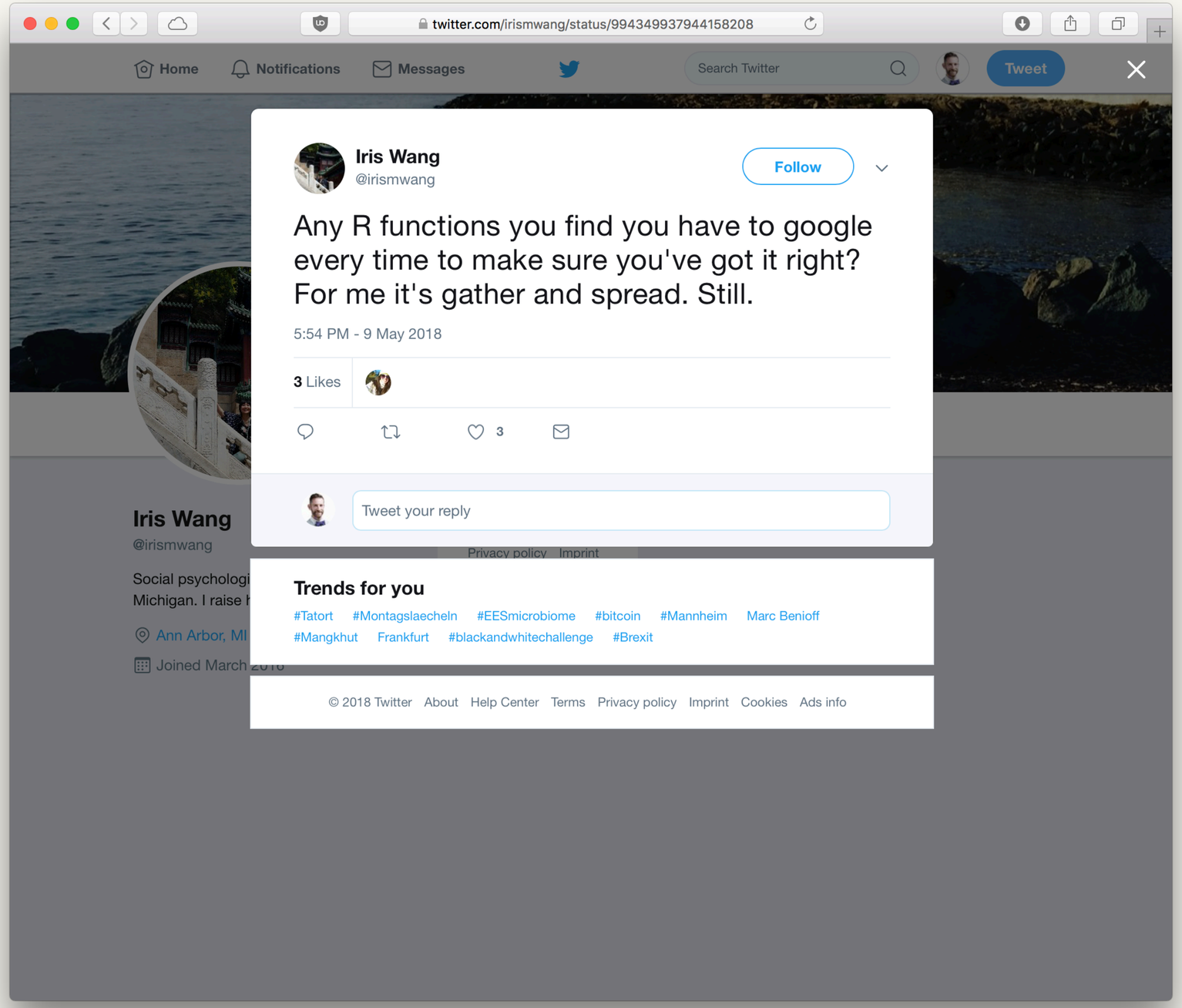
1. Design is iterative
2. Design embraces constraints
3. Design is affective
4. Design is taking the blame

Design is iterative

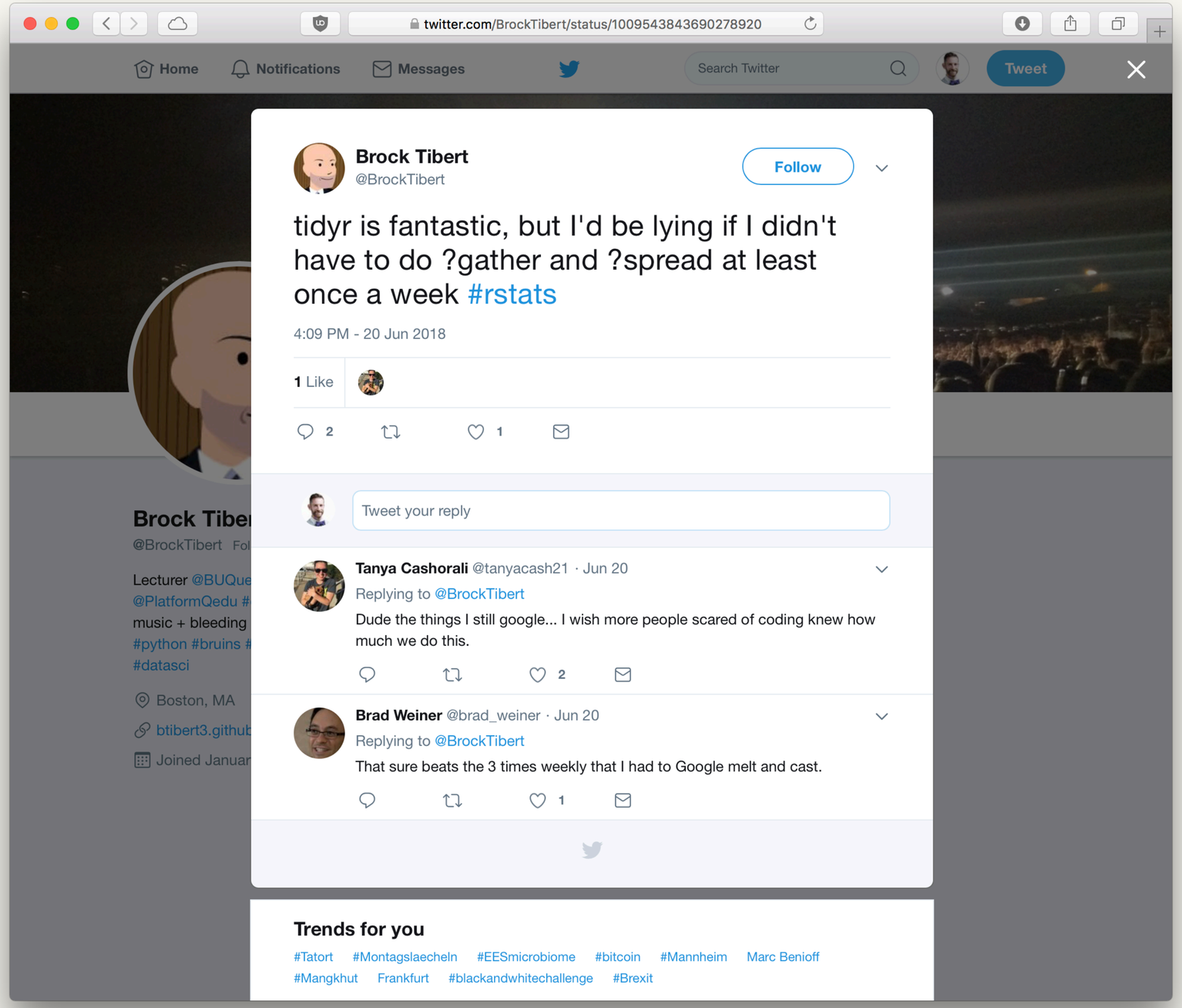
What is tidy data?

1. Each variable in a column
2. Each observation in a row
3. Each cell is one value


reshape2
(2010)
melt()
dcast()
tidyr
(2014)
spread()
gather()
reshape
(2005)
melt()
cast()

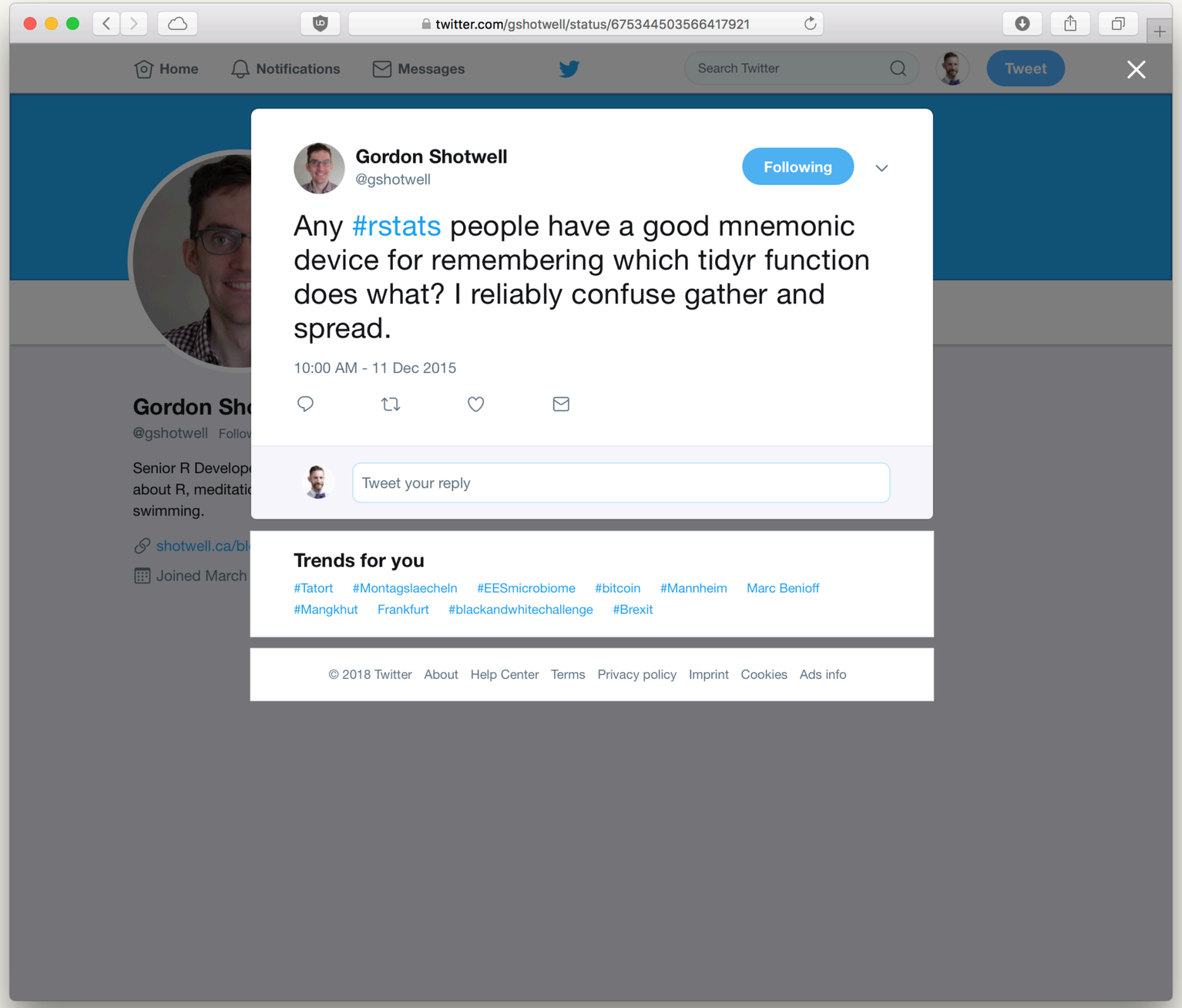


https://twitter.com/irismwang/status/994349937944158208

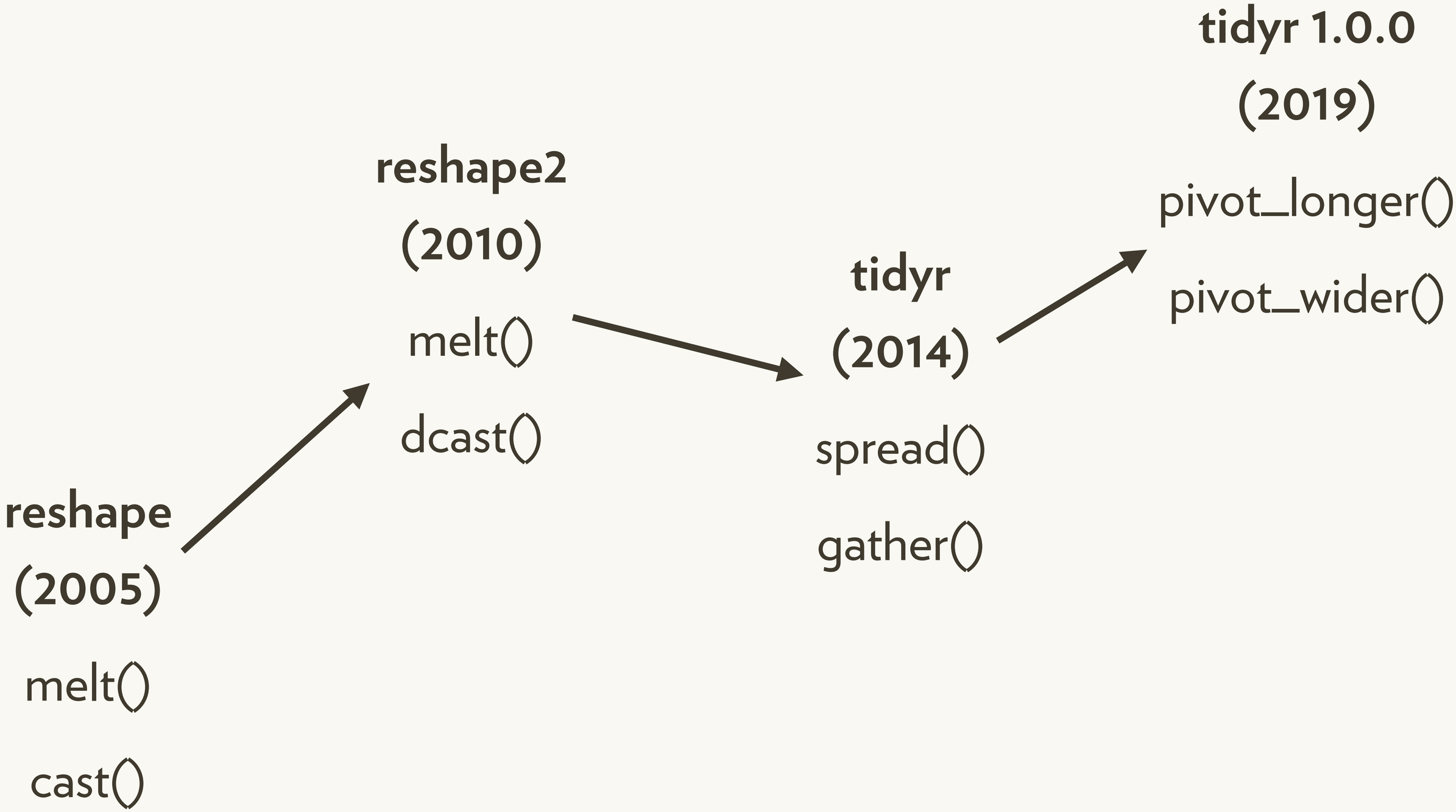


https://twitter.com/BrockTibert/status/1009543843690278920

https://twitter.com/mattfrost/status/768078660200898560



https://twitter.com/gshotwell/status/675344503566417921


reshape
(2005)
melt()
cast()
reshape2
(2010)
melt()
dcast()
tidyr
(2014)
spread()
gather()
tidyr 1.0.0
(2019)
pivot_longer()
pivot_wider()

JD Long
@CMastication

I finally held my breath and updated {tidyr} just to play with the new pivot functions. I read the help file on `pivot_wider` and that damn thing worked *on the first try*. That's preposterous. #rstatstidyr.tidyverse.org/articles/pivot…

Pivoting
tidyr.tidyverse.org

♡ 222 2:07 PM - Oct 31, 2019

41 people are talking about this

Travis Dawry
@tdawry

Some of my students had problems with gather() and spread(), but with pivot_longer() and pivot_wider() the tables have turned. #rstats

♡ 182 2:14 PM - Oct 12, 2019

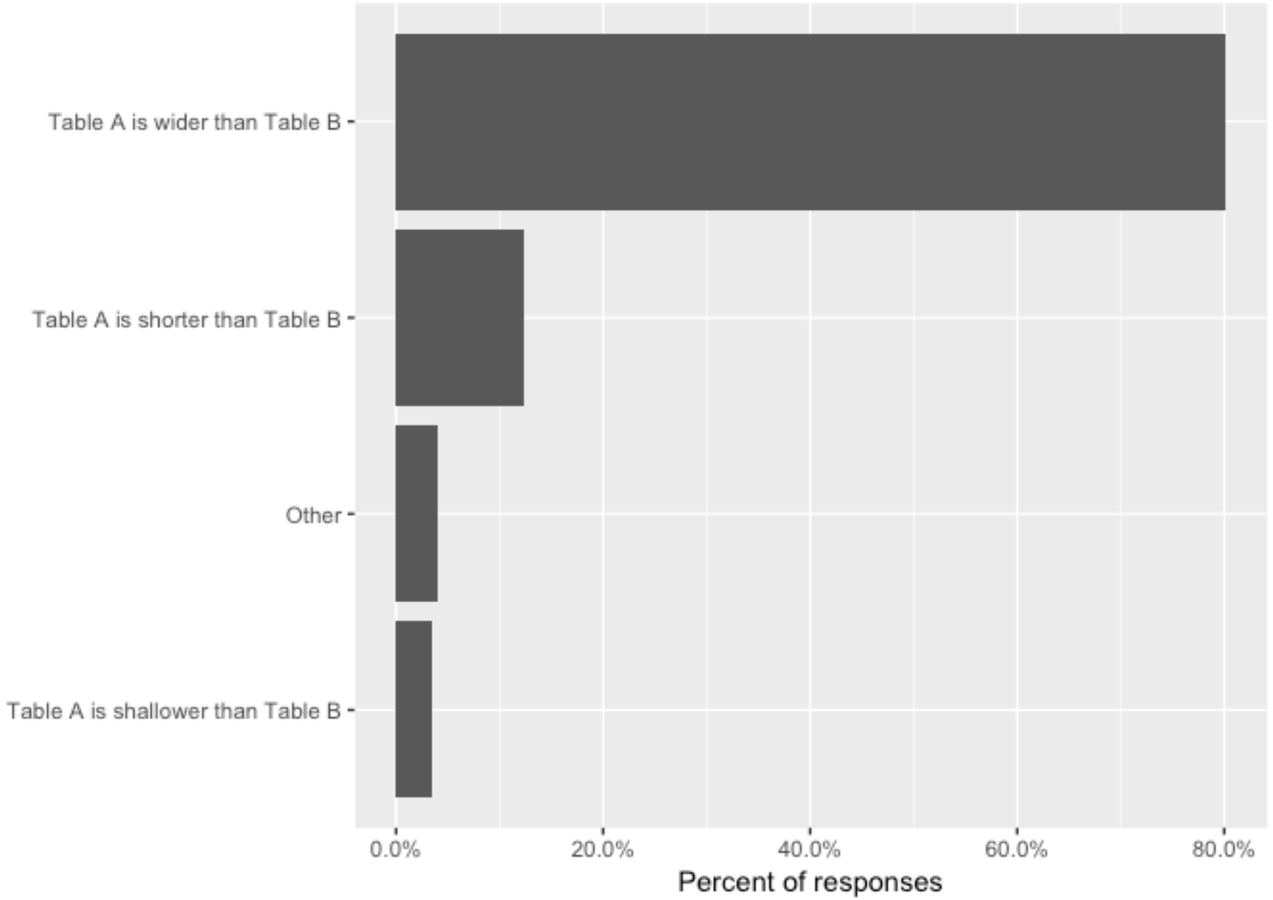
22 people are talking about this

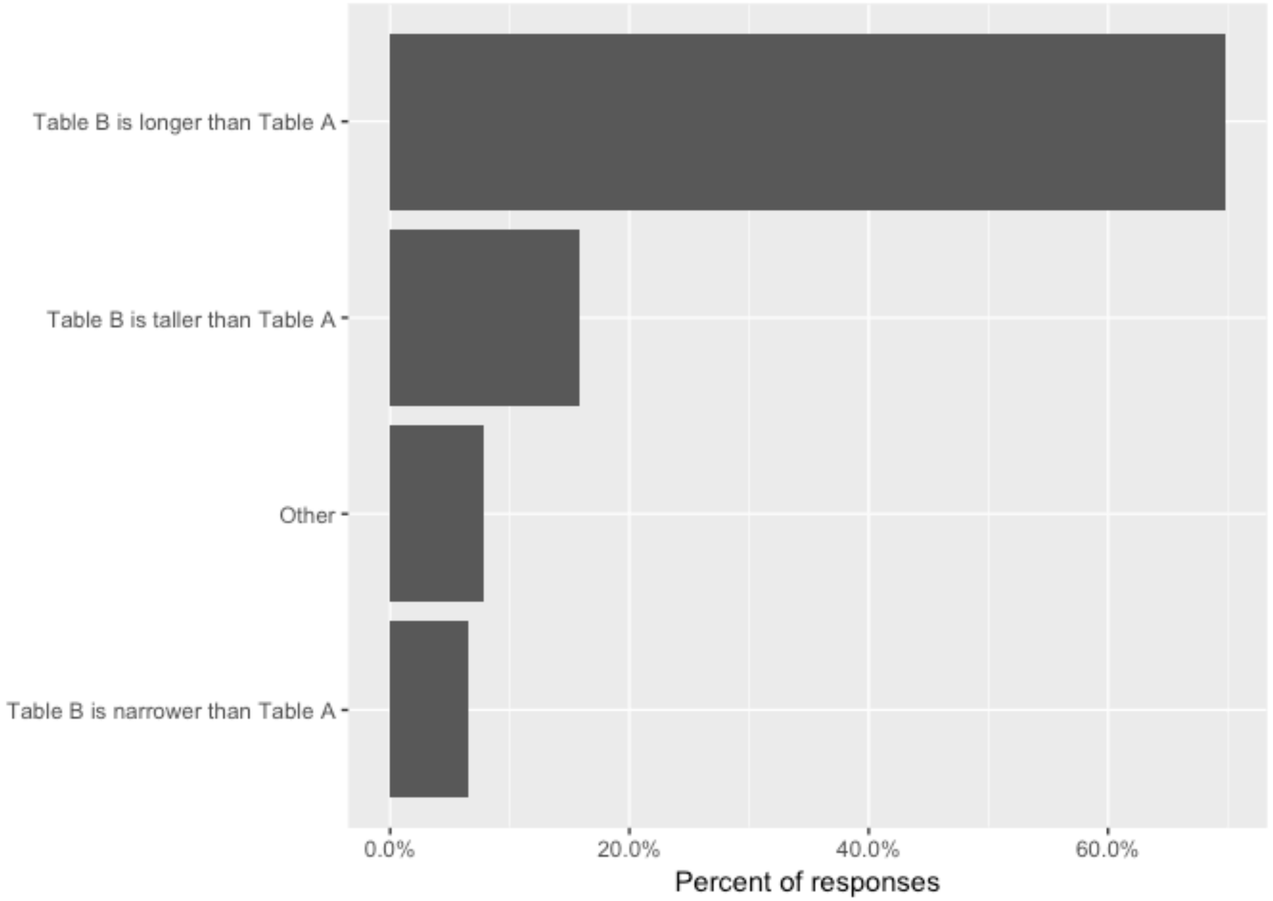
Table A

id	x	y	z
1	a	b	c
2	d	e	f

Table B

id	n	v
1	x	a
1	y	b
1	z	c
2	x	d
2	y	e
2	z	f





https://github.com/hadley/table-shapes

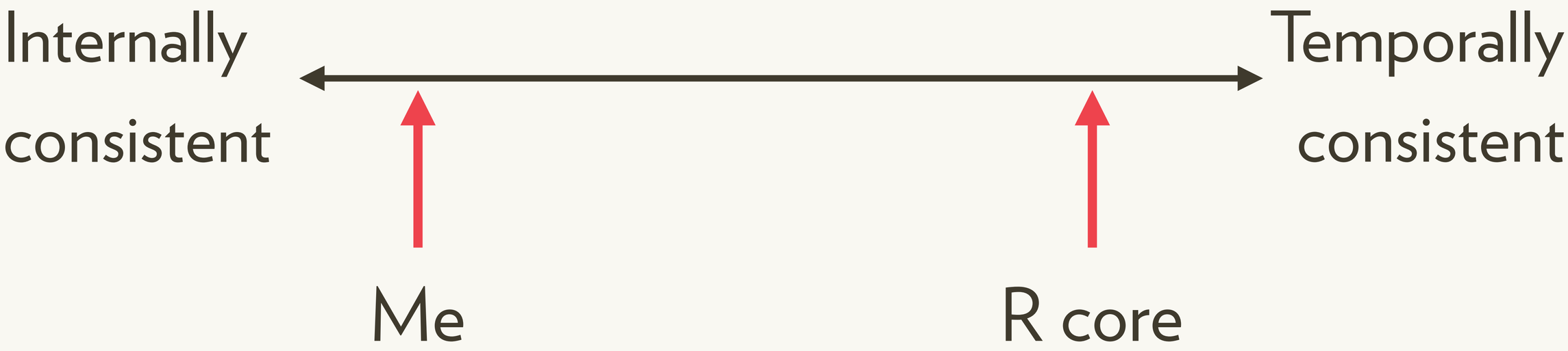
“The cost of never making a mistake is very often never making a change. It’s just too incredibly hard to be sure.”

— GeePaw Hill http://geepawhill.org/try-different-not-harder/

Design accepts constraints

What happens if you make a design mistake?

I think this is the greatest challenge in programming



I think this is the greatest challenge in programming

Both have pros and cons

Utopian ⟷ Conservative

Easier to learn the first time

Existing users may need to re-learn

Optimised for future users

Old code keeps working

Memorise more special cases

Optimised for existing users

Can you predict what this returns?

```
df <- data.frame(xyz = "a")
df$x
```

Data frames

```
df <- data.frame(xyz = "a")
df$x
#> [1] a
#> Levels: a
```

Can’t change 20 year behaviour, can but introduce new classes

```
df <- tibble(xyz = "a")
df$x
#> NULL
#> Warning message:
#> Unknown or uninitialised column: 'x'.
```

Tibbles

```
df <- tibble(xyz = "a")
df$xyz
#> [1] "a"
```

Is it worth it?

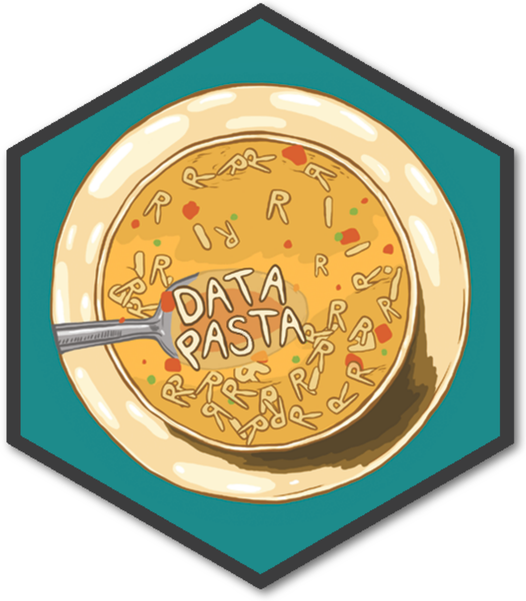
Design is affective


auditor


GGFORCE


tsibble

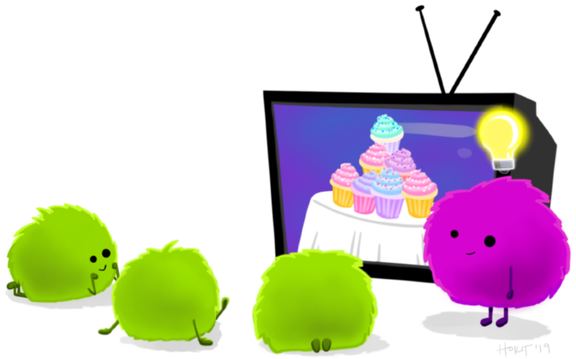

R
ladies
xalapa


DATA
PASTA


naniar

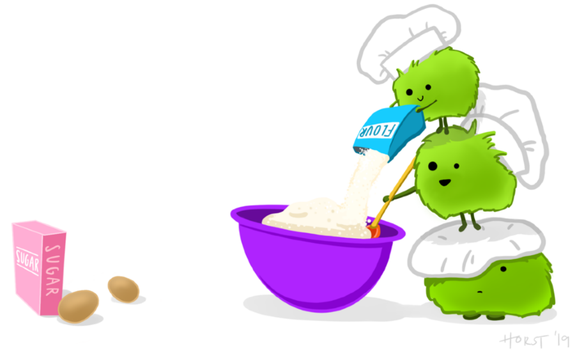

R-LADIES
AUSTIN

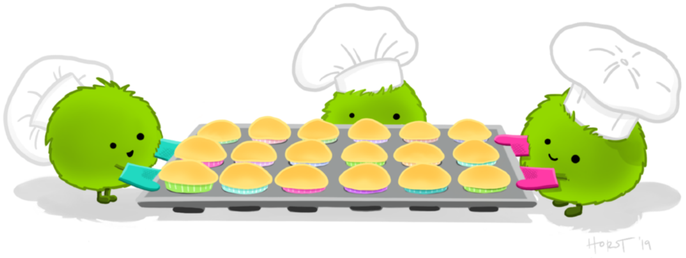

DALEX

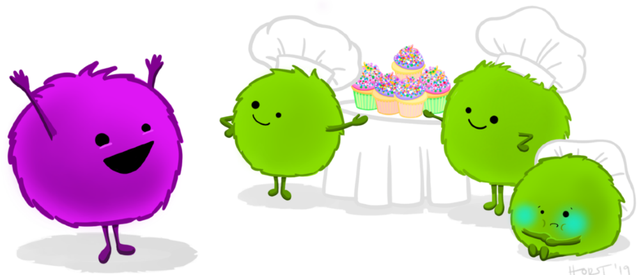


The art of Allison Horst

https://github.com/allisonhorst

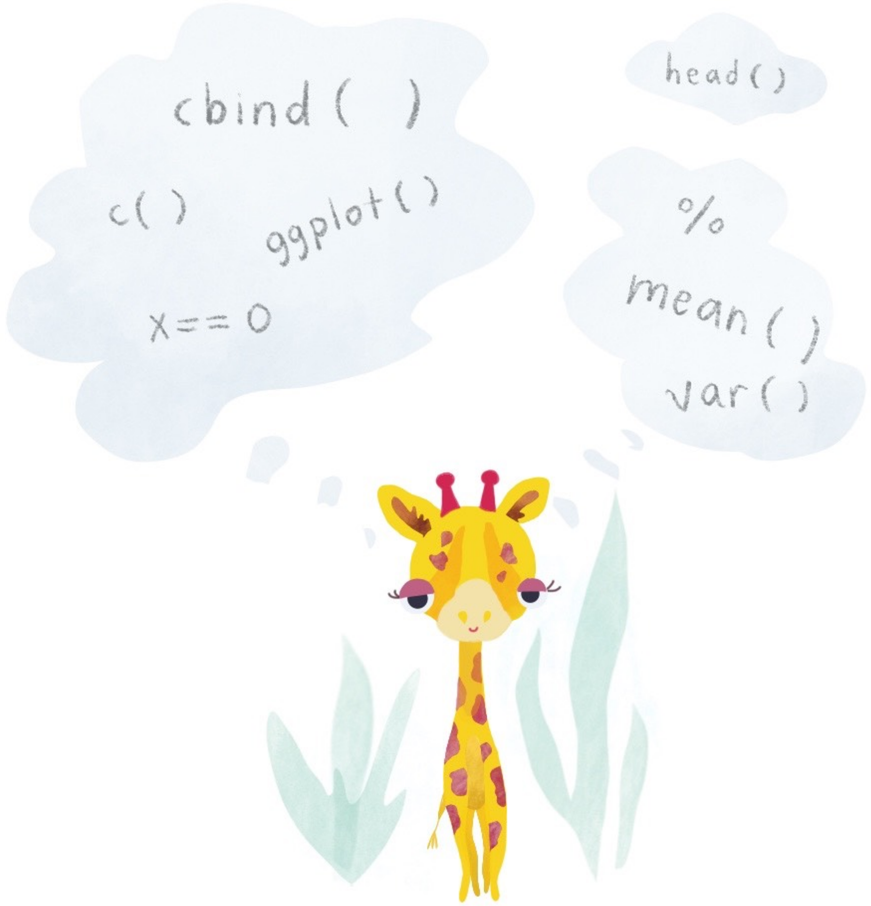


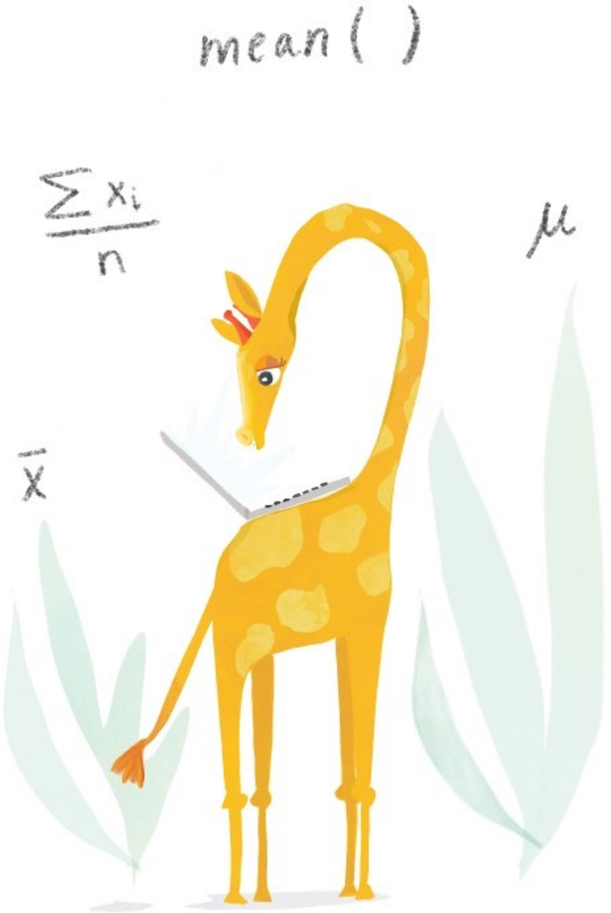


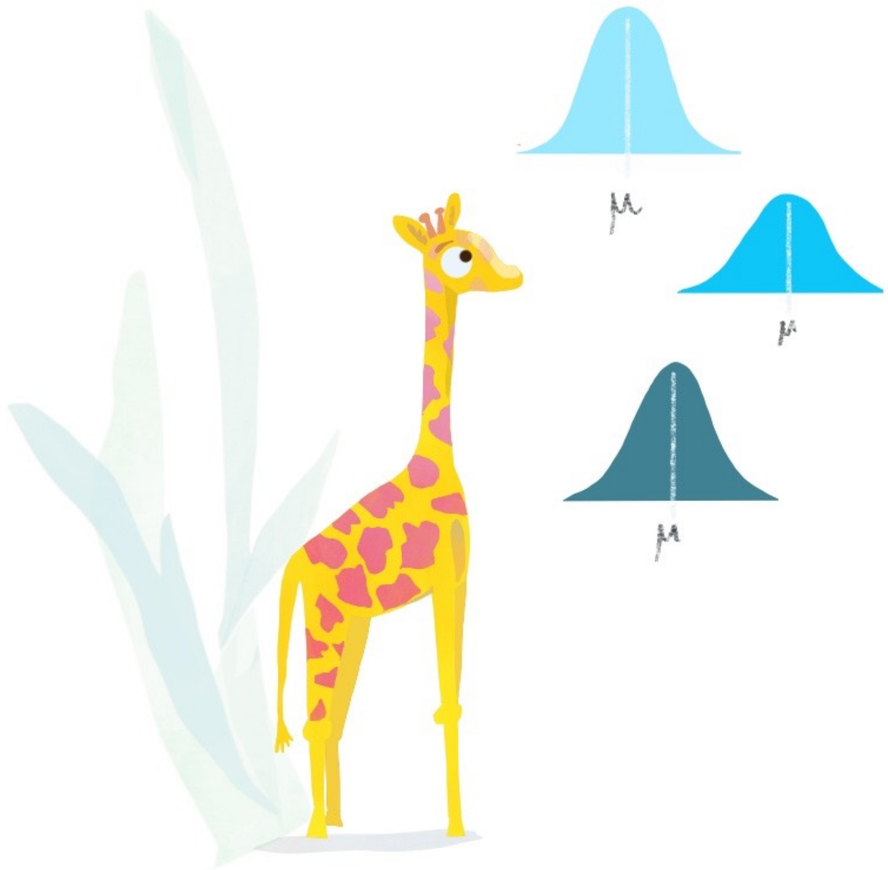


Teacups, Giraffes, and Statistics

Hasse Walum and Desirée de Leon







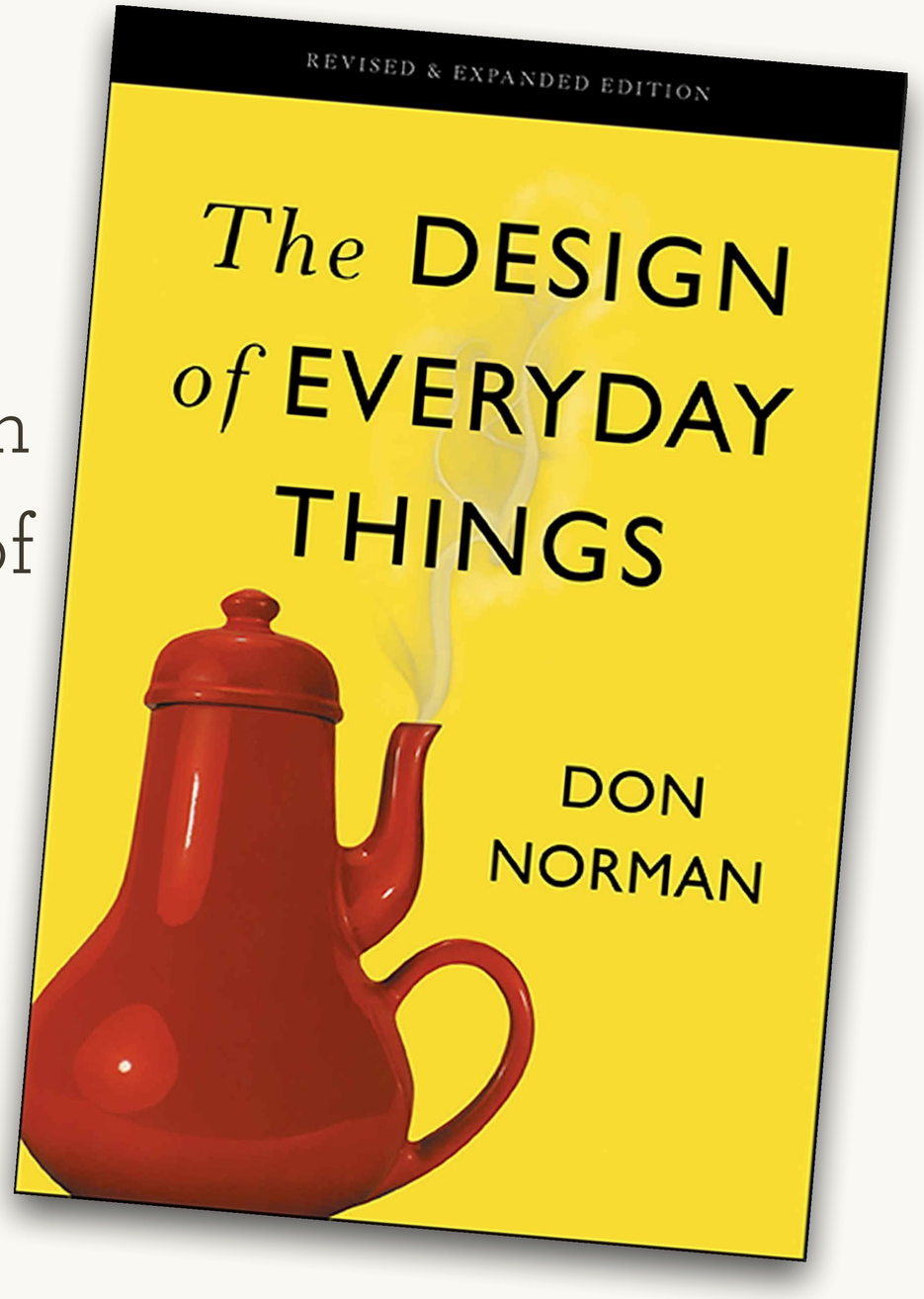
https://tinystats.github.io/teacups-giraffes-and-statistics/

Design is taking the blame


There’s

“When you have trouble with things—whether it’s figuring out whether to push or pull a door or the arbitrary vagaries of the modern computer and electronics industries—it’s not your fault. Don't blame yourself: blame the designer...”

— *Donald A. Norman*



But the idea of a door already exists in your head. How do new metaphors get in there?

New idea  Old idea

Fits perfectly

Might need to force it to fit

Surprise

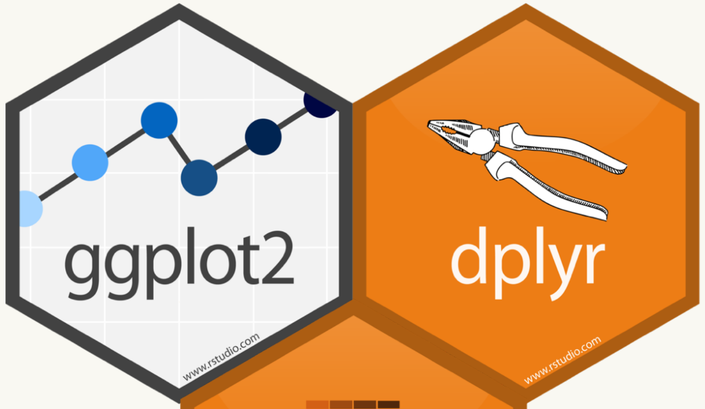
Comfort

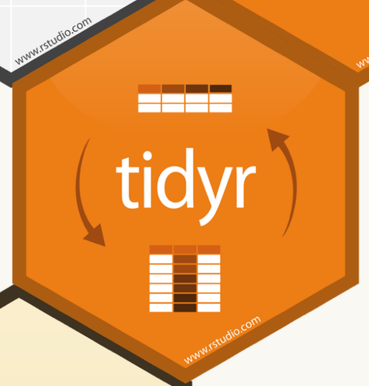
Conclusion

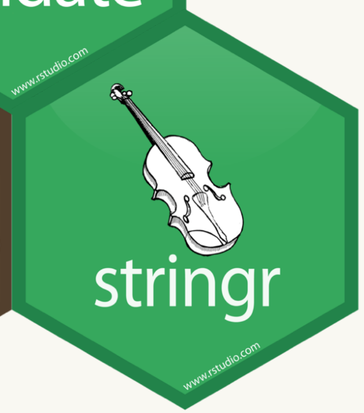
https://unsplash.com/photos/8HyrGTYPQ68 — Eric Muhr

Design

Software engineering







1. Design is iterative
2. Design embraces constraints
3. Design is affective
4. Design is taking the blame

R	Python
Apple	Linux
	Programmers + engineers
CRAN	pip
Specialised	General purpose
Exploration	Automation
REPL + Rmd	Jupyter notebooks