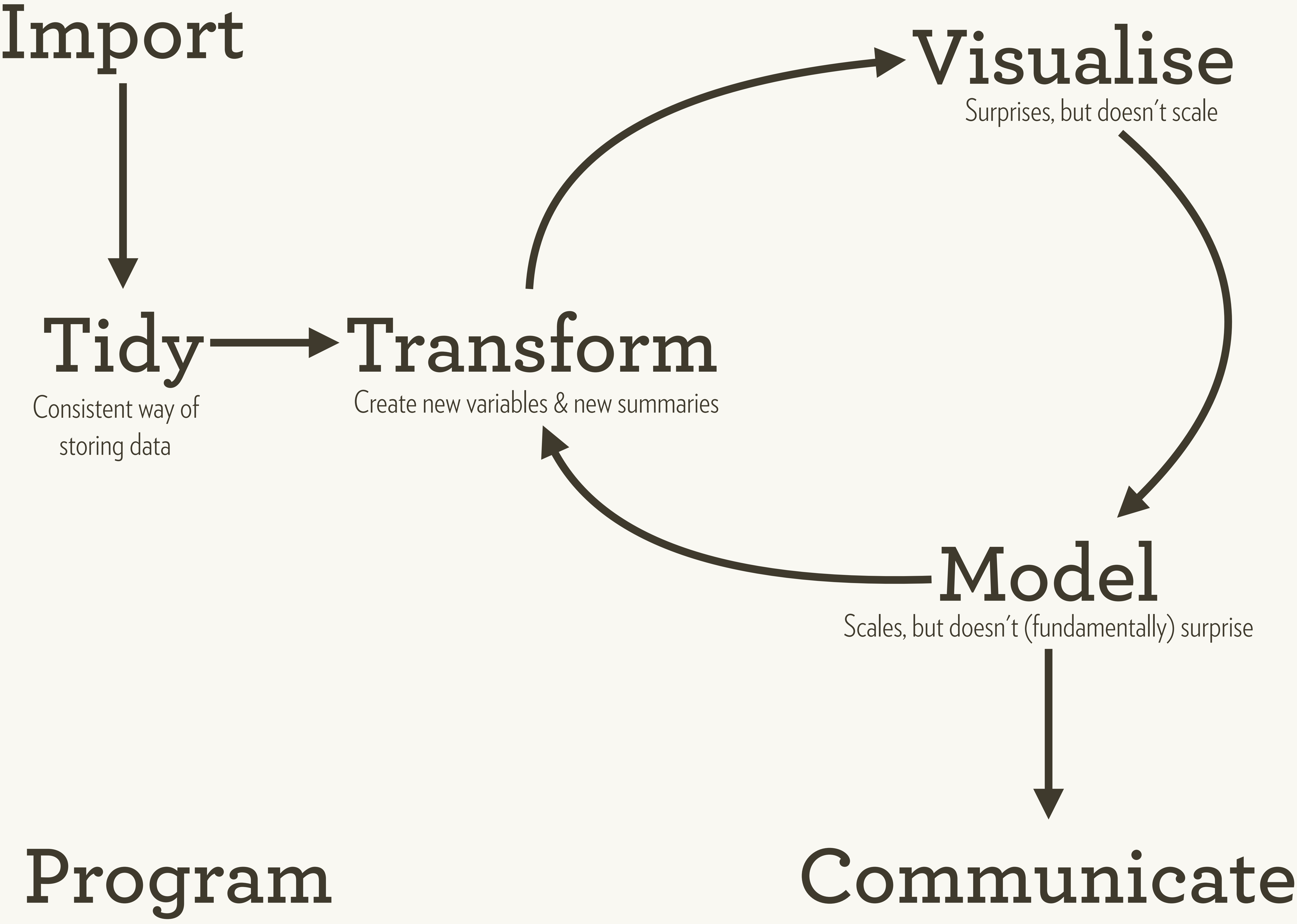
Welcome to the tidyverse

December 2017

Hadley Wickham
@hadleywickham
Chief Scientist, RStudio



Import

Tidy

Consistent way of storing data

Transform

Create new variables & new summaries

Visualise

Surprises, but doesn't scale

Model

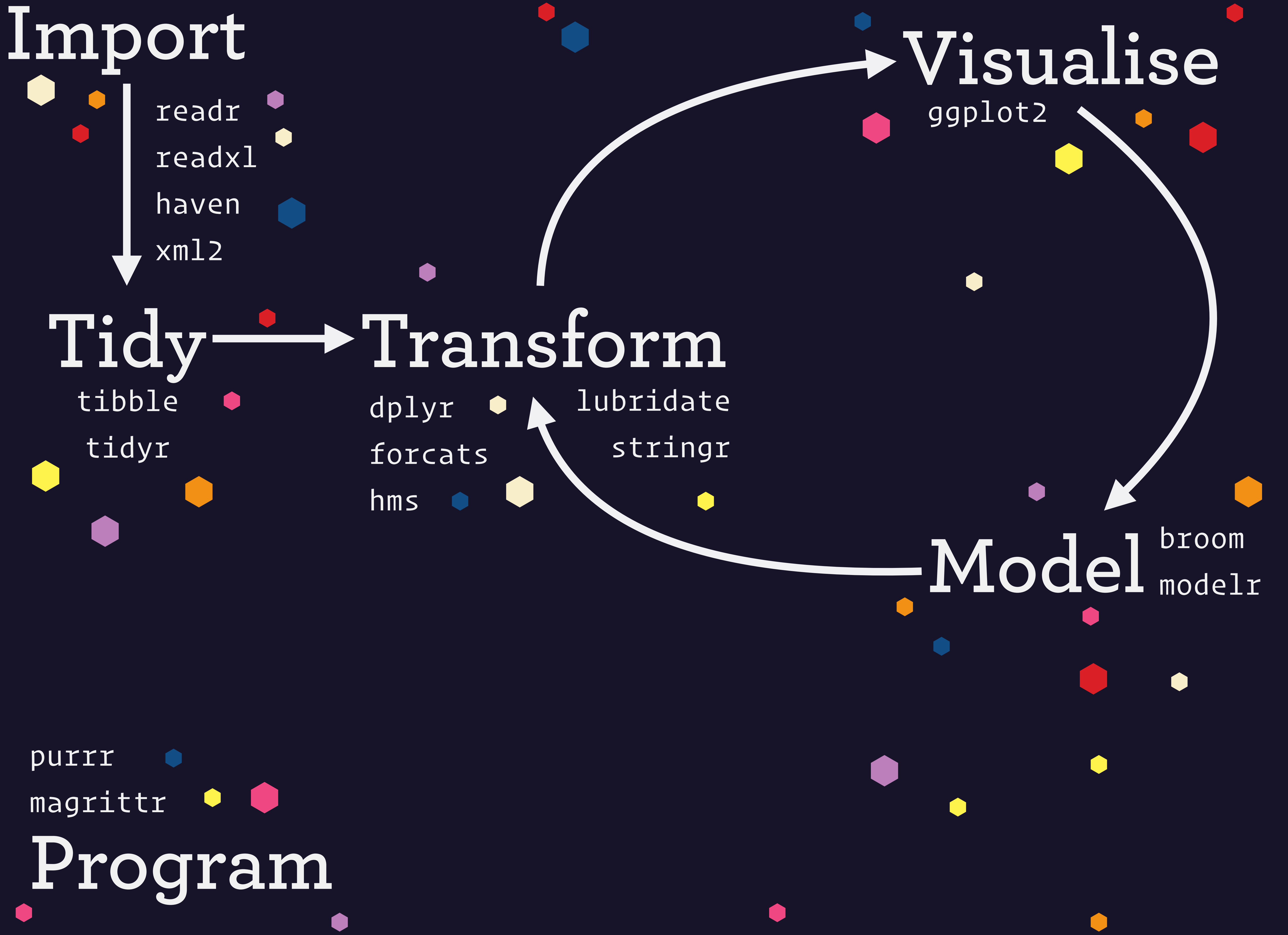
Scales, but doesn't (fundamentally) surprise

Program

Communicate

tidyverse
www.rstudio.com

@aaronwolen, @aghaynes, @ajdamico, @ajschumacher, @alberthkcheng, @alyst, @andrew, @andrewjlm, @apjanke, @arneschillert, @artemklevtsov, @arunsrinivasan, @asnr, @astamm, @austenhead, @baptiste, @bbolker, @bearloga, @benmarwick, @bhive01, @BioStatMatt, @bpbond, @bquast, @BrianDiggs, @briatte, @burchill, @casallas, @cb4ds, @cboettig, @cderv, @christophergandrud, @cmartin, @colinbrislawn, @coolbutuseless, @cosinequanon, @craigcitro, @csgillespie, @ctbrown, @daattali, @dandermotj, @danliIDEA, @DanRuderman, @davharris, @davidmorrison, @dchiu911, @dchudz, @dewittpe, @dgromer, @dgrtwo, @dhimmel, @dickoa, @diogocp, @djmurphy420, @dlebauer, @dmedri, @dmenne, @dougmitarotonda, @dpastoor, @dpocock, @dtelad11, @earino, @echasnovski, @ecortens, @eddelbuettel, @edgararuiz, @edwindj, @egnha, @ehrlinger, @eibanez, @eipi10, @ekstroem, @emojiencoding, @etiennebr, @evanmiller, @fpinter, @FvD, **@gaborcsardi**, @gagolews, **@garrettgman**, @gavinsimpson, @gergness, @gnustats, @gorcha, @goyalmunish, @gregmacfarlane, @guillett, @gvelasq2, @hannesmuehleisen, @has2k1, @helix123, @hmalmedal, @hoehleatsu, @hoesler, @holstius, @hrbrmstr, @ianmcook, @ijlyttle, @ilarischeinin, @imanuelcostigan, @Ironholds, @ismayc, @isomorphisms, @itsdalmo, @JakeRuss, @janschulz, @jasonelaw, @javierluraschi, @jayhesselberth, @jcheng5, @jdnewmil, @jefferis, **@jennybc**, @jenzopr, @jeremystan, @jeroen, @jgabry, @jhuovari, @jiho, **@jimhester**, @jirkalewandowski, @jjallaire, @jmarshallnz, @jmi5, @joethorley, @JoFrhwld, @jonboiser, @jonmcalder, @joranE, @joshkatz, @jrnold, @juba, @junkka, @justmarkham, @kalibera, **@karawoo**, @karthik, @Katiedaisey, @kbenoit, @Kevin-M-Smith, @kevinushey, @kmillar, @kohske, **@krlmlr**, @kwenzig, @kwstat, @KZARCA, @l-d-s, @LaDilettante, @larmarange, @leondutoit, @lepennec, @lindbrook, **@lionel-**, @lmullen, @lorenzwalthert, @lselzer, @luckyrandom, **@LucyMcGowan**, @lwjohnst86, @MarcusWalz, @markdly, @markriseley, @matthieugomez, @maurolepore, @mdlincoln, @mgacc0, @mgirlich, @michaelquinn32, @mikelove, @mkcor, @mkuehn10, @mkuhn, @mmparker, @msonnabaum, @ncarchedi, @NoahMarconi, @noamross, @npjc, @nutterb, @paternogbc, @paul-buerkner, @PedramNavid, @PeteHaitch, @pierucci, @pimentel, @pitakakariki, @pkq, @r2evans, @rbdixon, @richierocks, @RiRam, @rmsharp, @robertzk, @rohan-shah, **@romainfrancois**, @RoyalTS, @rsaporta, @rtaph, @rudazhan, @ruderphilipp, @s-fleck, @seaaan, @setempler, @sfirke, @shabbybanks, @sjackman, @sjPlot, @smbache, @statisfactions, @steromano, @t-kalinowski, @tareefk, @tdhock, @terrytangyuan, @thomasp85, @tjmahr, @tklebel, @tmshn, @tonytonov, @tuttinator, @tverbeke, @uribo, @vspinu, **@wch**, @webbedfeet, @wibeasley, @wligtenberg, @x0rshift, @xiaodaigh, @Yeedle, @yutannihilation, @zeehio, @zhaoy, and @zhilongjia



Import

readr

readxl

haven

xml2

Tidy

tibble

tidyr

Transform

dplyr

forcats

hms

lubridate

stringr

Visualise

ggplot2

Model

broom

modelr

purrr

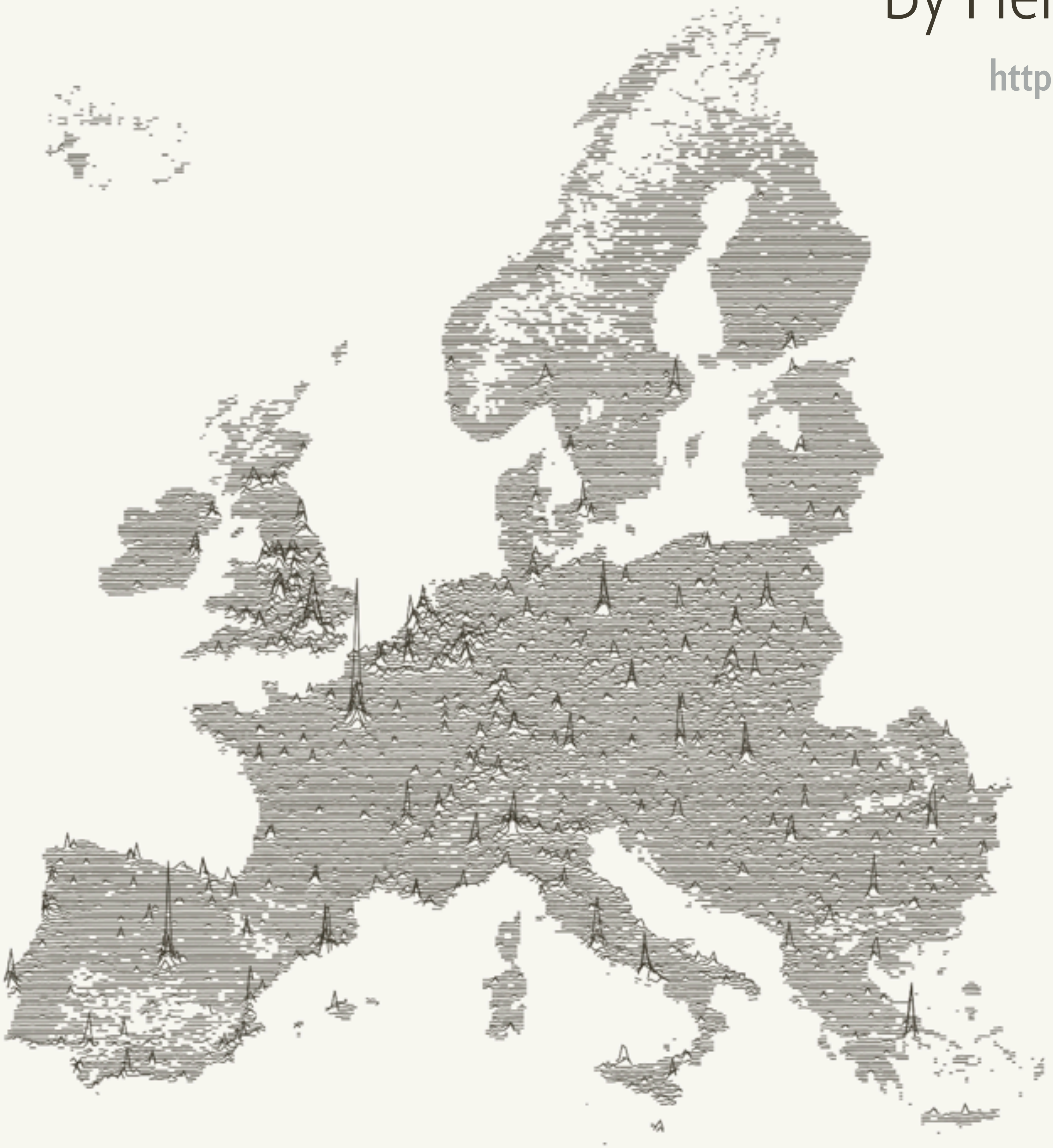
magrittr

Program

What does this plot show?

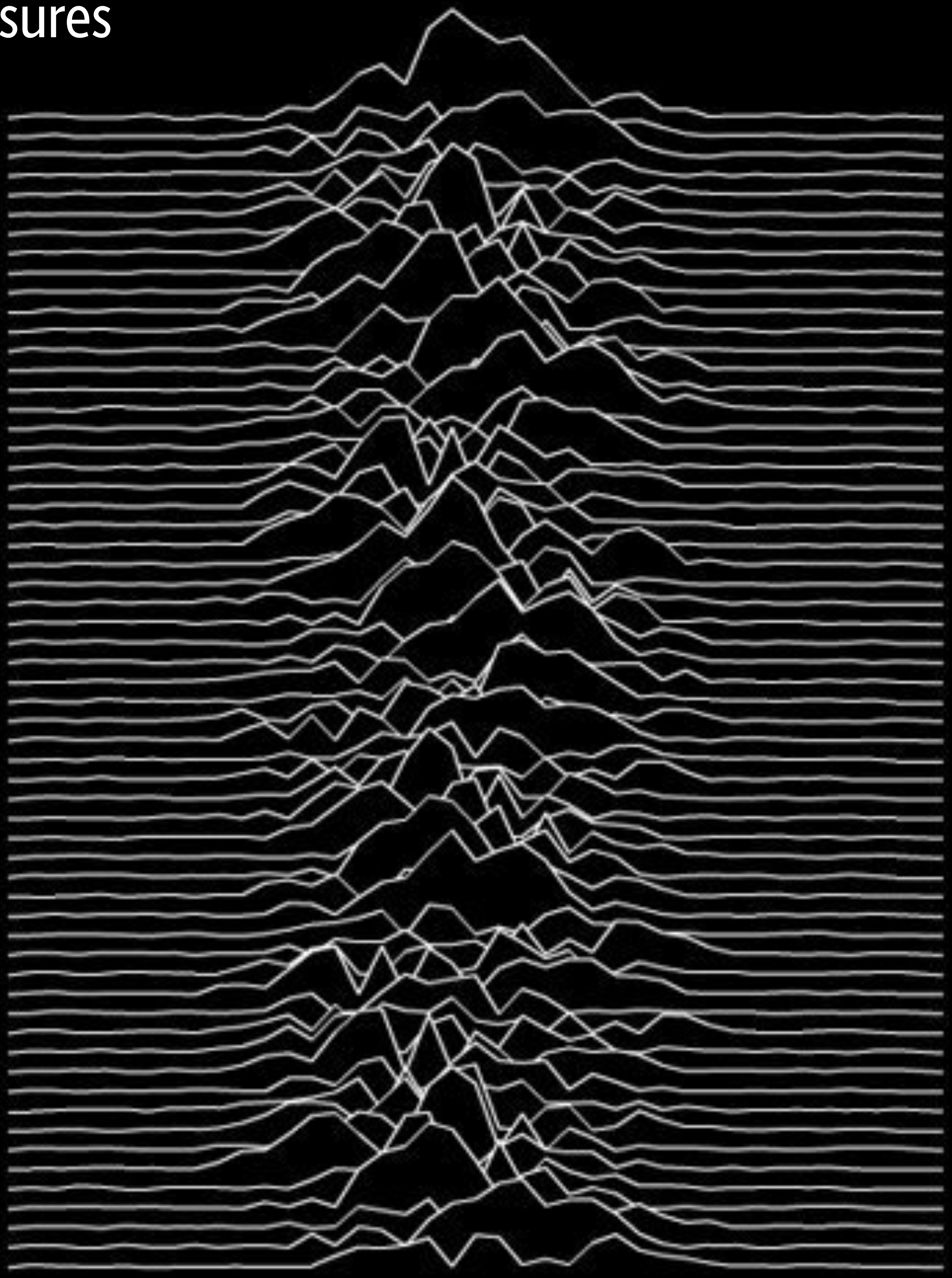
By Henrik Lindberg

https://github.com/halhen



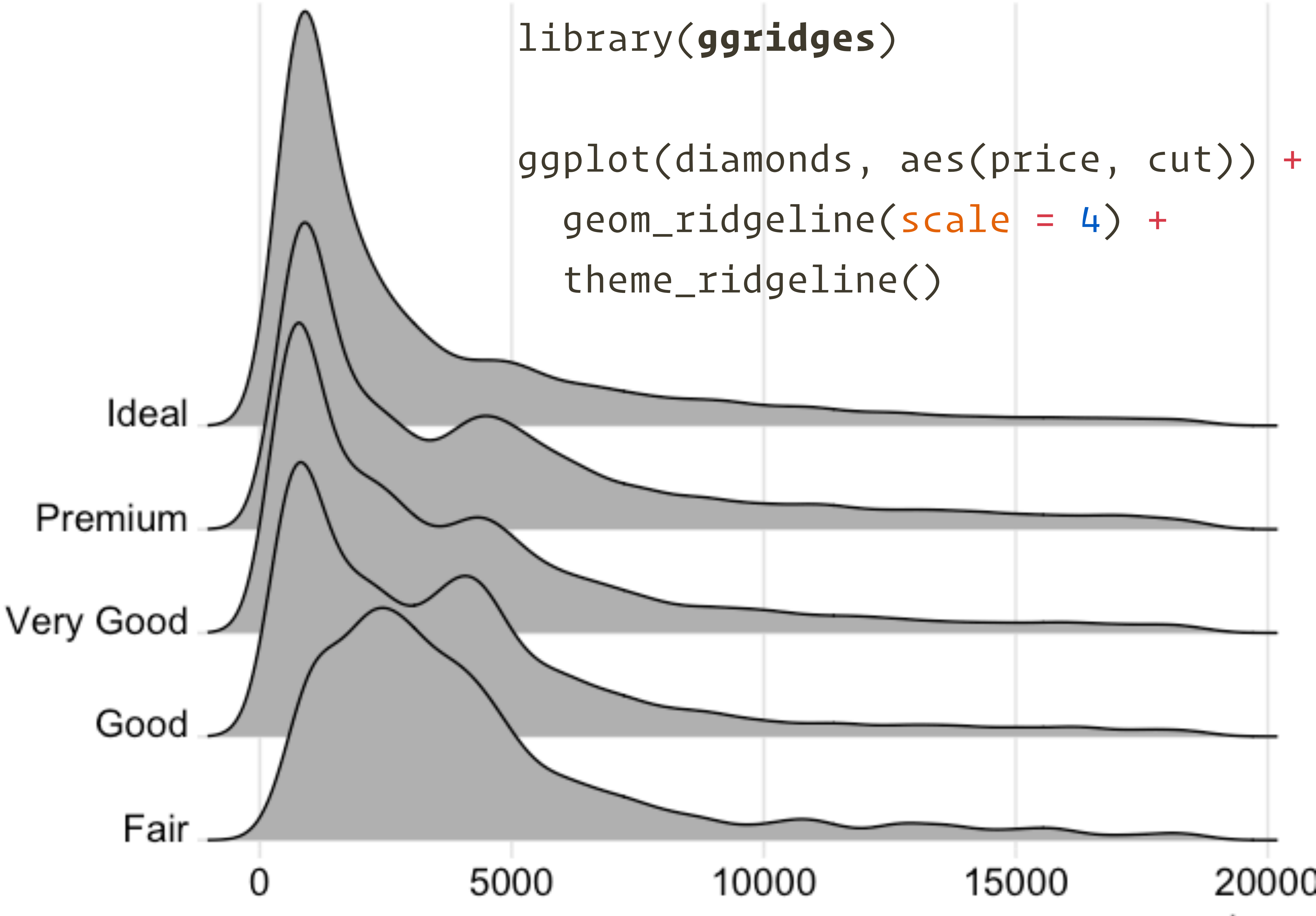
Unknown Pleasures

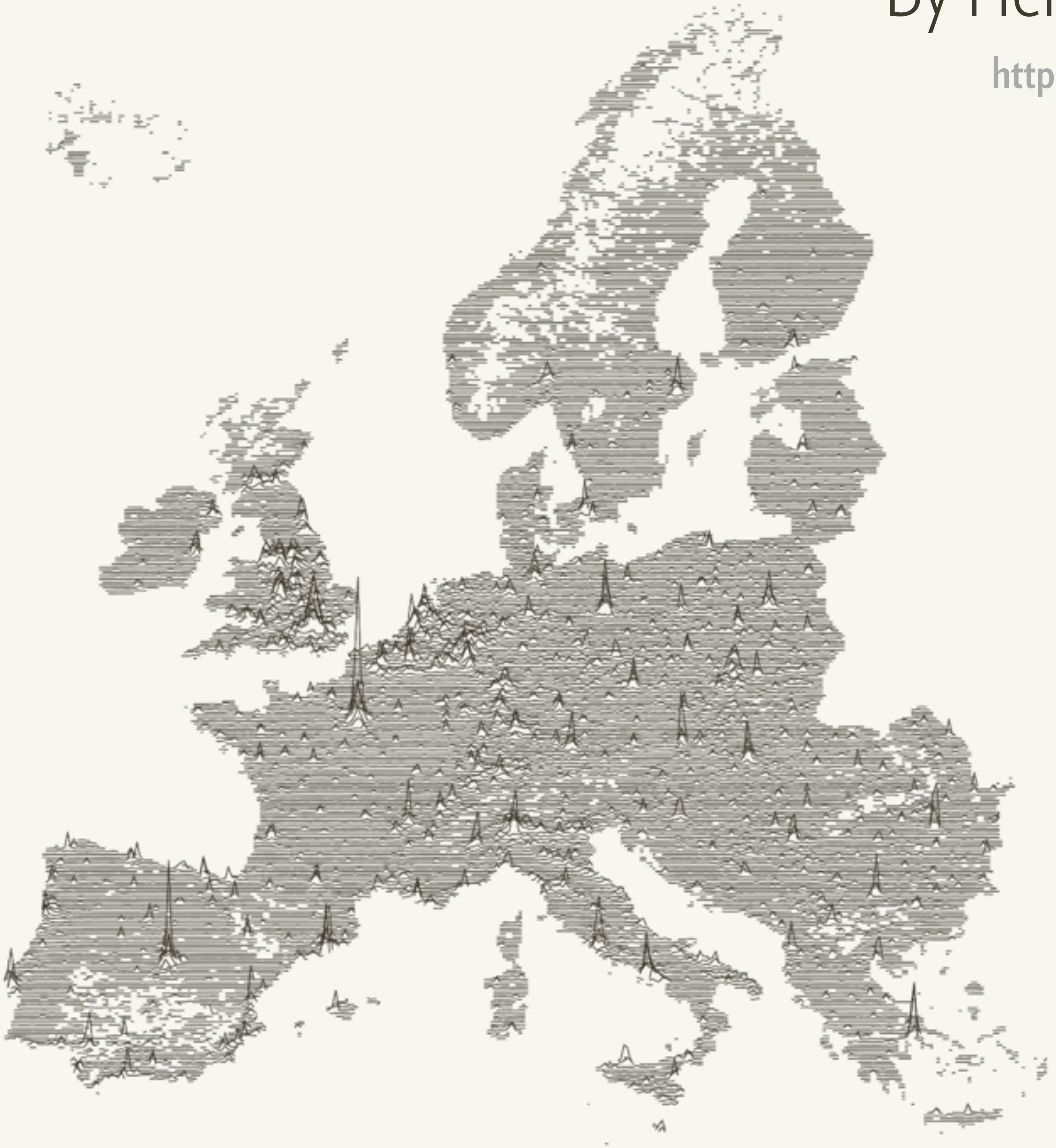
Joy Division



```
library(ggplot2)
library(ggridges)

ggplot(diamonds, aes(price, cut)) +
  geom_ridgeline(scale = 4) +
  theme_ridgeline()
```

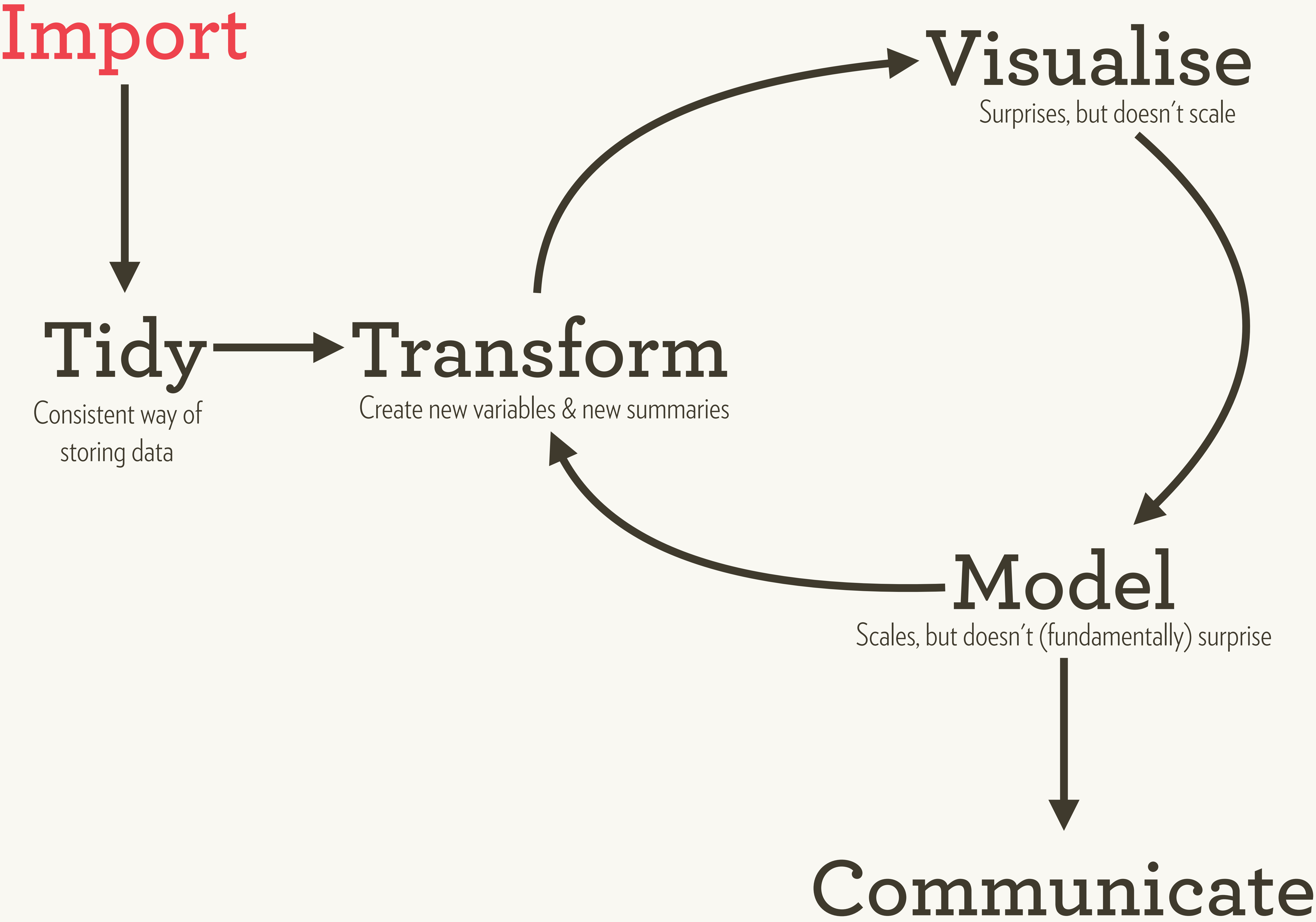




By Henrik Lindberg

https://github.com/halhen

Import



Import

Tidy

Consistent way of storing data

Transform

Create new variables & new summaries

Visualise

Surprises, but doesn't scale

Model

Scales, but doesn't (fundamentally) surprise

Communicate

Data import is 80% boredom

https://unsplash.com/photos/GNUcUx-iObg

And 20% endless screaming

https://unsplash.com/photos/RdHma30logY

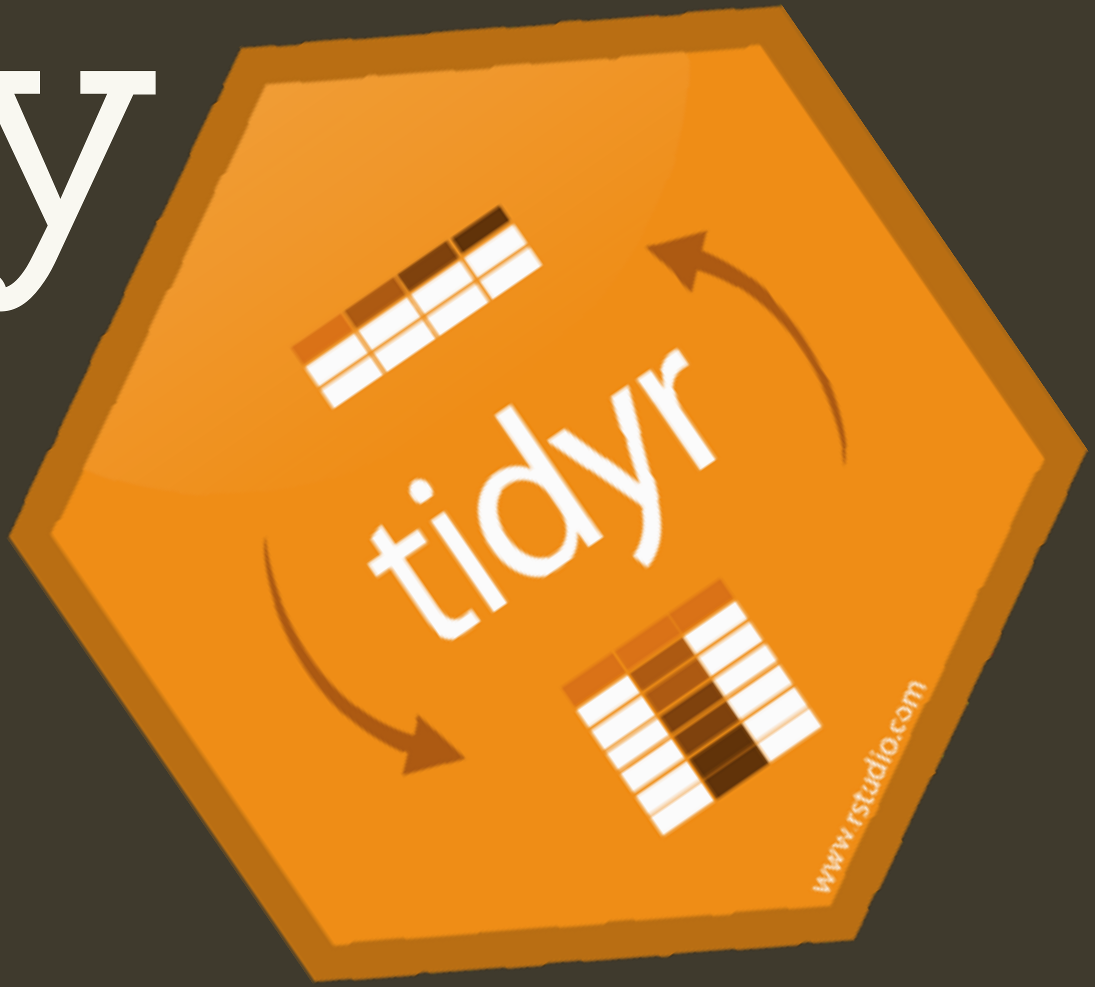
readr
www.rstudio.com
readxl
www.rstudio.com
feather
www.rstudio.com
haven
www.rstudio.com
rvest
www.rstudio.com

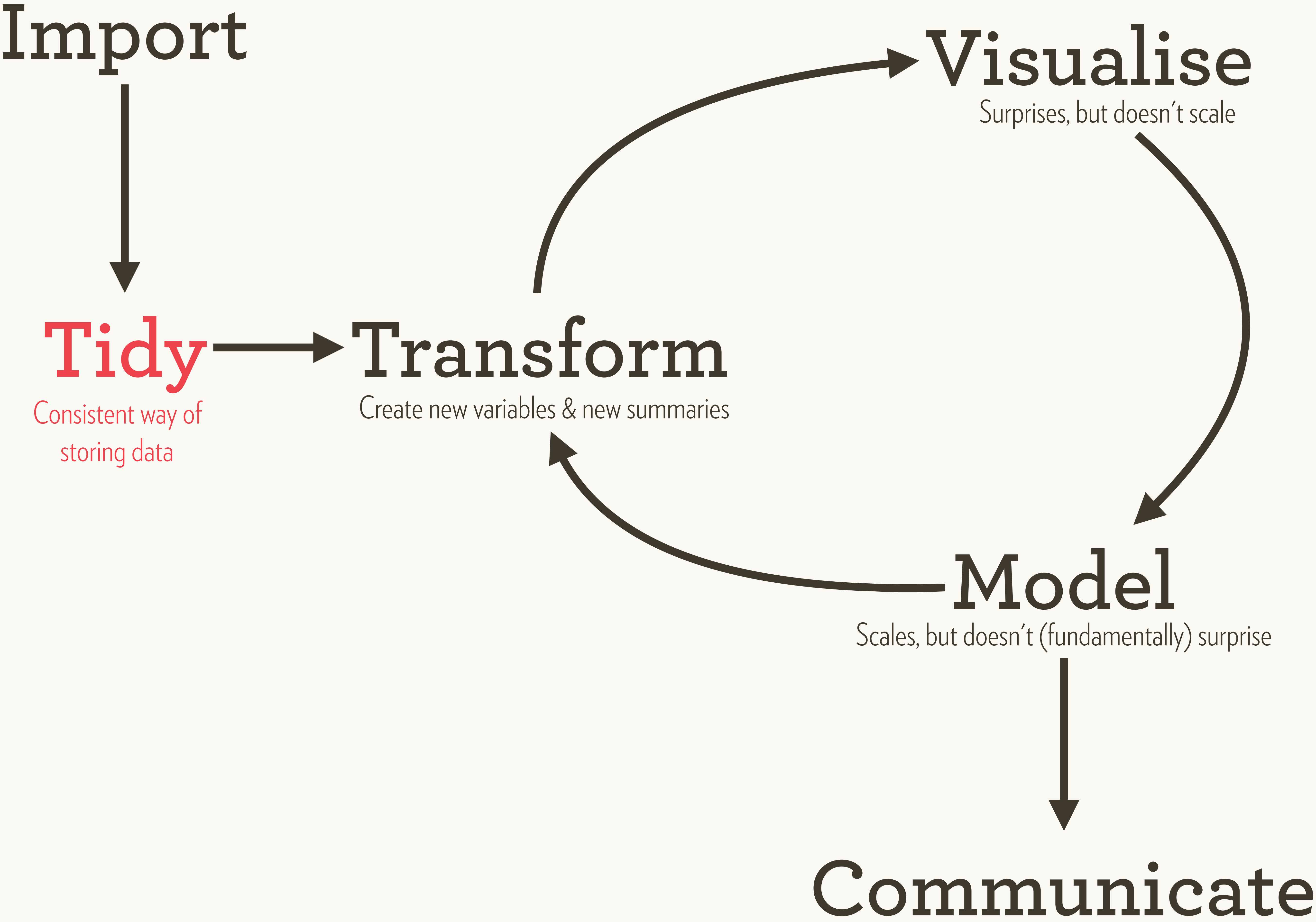
For Europe population map

```
pop_raw <- bind_rows(
  read_csv('europe-pop/GEOSTAT_grid_POP_1K_2011_V2_0_1.csv'),
  read_csv('europe-pop/JRC-GHSL_AIT-grid-POP_1K_2011.csv')
)
```

```
#> # A tibble: 2,108,409 × 7
#>    TOT_P        GRD_ID CNTR_CODE METHD_CL  YEAR DATA_SRC TOT_P_CON_DT
#>    <int>         <chr>     <chr>    <chr> <int>    <chr>        <chr>
#> 1      8 1kmN2689E4337        DE        A  2011       DE        other
#> 2      7 1kmN2689E4341        DE        A  2011       DE        other
#> 3      3 1kmN2690E4341        DE        A  2011       DE        other
#> 4      3 1kmN2691E4340        DE        A  2011       DE        other
#> 5     22 1kmN2691E4341        DE        A  2011       DE        other
#> 6     20 1kmN2692E4341        DE        A  2011       DE        other
#> 7      9 1kmN2692E4344        DE        A  2011       DE        other
#> 8     28 1kmN2693E4340        DE        A  2011       DE        other
#> # ... with 2,108,401 more rows
```

Tidy





Import

Tidy

Consistent way of
storing data

Transform

Create new variables & new summaries

Visualise

Surprises, but doesn't scale

Model

Scales, but doesn't (fundamentally) surprise

Communicate

Tidy data = data that makes data analysis easy

Storage	Meaning
Column	Variable
Row	Observation
Data frame	Data set

First we loaded

```
pop_raw <- bind_rows(
  read_csv("europe-pop/GEOSTAT_grid_POP_1K_2011_V2_0_1.csv"),
  read_csv("europe-pop/JRC-GHSL_AIT-grid-POP_1K_2011.csv")
)
```

```
#> # A tibble: 2,108,409 × 7
#>     TOT_P        GRD_ID CNTR_CODE METHD_CL  YEAR DATA_SRC TOT_P_CON_DT
#>     <int>         <chr>     <chr>    <chr> <int>    <chr>        <chr>
#> 1       8 1kmN2689E4337        DE        A  2011       DE        other
#> 2       7 1kmN2689E4341        DE        A  2011       DE        other
#> 3       3 1kmN2690E4341        DE        A  2011       DE        other
#> 4       3 1kmN2691E4340        DE        A  2011       DE        other
#> 5      22 1kmN2691E4341        DE        A  2011       DE        other
#> 6      20 1kmN2692E4341        DE        A  2011       DE        other
#> 7       9 1kmN2692E4344        DE        A  2011       DE        other
#> 8      28 1kmN2693E4340        DE        A  2011       DE        other
#> # ... with 2,108,401 more rows
```

GRD_ID contains multiple variables

What are they?

```
1kmN2689E4337
1kmN2689E4341
1kmN2690E4341
1kmN2691E4340
```

Latitude & longitude each in two variables

```
1kmN2689E4337
1kmN2689E4341
1kmN2690E4341
1kmN2691E4340
```

Can define variables by their positions

```
1kmN2689E4337
1kmN2689E4341
1kmN2690E4341
1kmN2691E4340
```

Now we tidy

```r
pop_raw2 <- pop_raw %>%
  separate(
    GRD_ID,
    c("grid", "NS", "lat", "EW", "lon"),
    c(3, 4, 8, 9),
    convert = TRUE
  )
```

Now we tidy

```
pop_raw2 <- pop_raw %>%
  separate(
    GRD_ID,
    c("grid", "NS", "lat", "EW", "lon"),
    c(3, 4, 8, 9),
    convert = TRUE
  ) %>%
  mutate(
    lat = lat / 100 * if_else(NS == "S", -1, 1),
    lon = lon / 100 * if_else(EW == "W", -1, 1)
  ) %>%
  select(-EW, -NS)
```

%>%
Ceci n'est pas un pipe.

Could have written as

```
pop_raw2 <- separate(pop_raw,
  GRD_ID,
  c("grid", "NS", "lat", "EW", "lon"),
  c(3, 4, 8, 9),
  convert = TRUE
)
pop_raw3 <- mutate(pop_raw2,
  lat = lat / 100 * ifelse(NS == "S", -1, 1),
  lon = lon / 100 * ifelse(EW == "W", -1, 1)
)
pop_raw4 <- select(pop_raw3, -EW, -NS)
```

The pipe is syntactic sugar

```
x %>%
  f(a) %>%
  g(b, c)

# Equivalent to
g(f(x, a), b, c)

# Or
tmp1 <- f(x, a)
g(tmp1, b, c)
```

Makes it easy to read unfamiliar code

What does this code do?

```
library(tidyverse)
library(magick)

dir(pattern = ".png") %>%
  map(image_read) %>%
  image_join() %>%
  image_animate(fps = 1, loop = 25) %>%
  image_write("my_animation.gif")
```

https://twitter.com/ricardokriebel/status/849626401611411458

https://twitter.com/ricardokriebel/status/849626401611411458

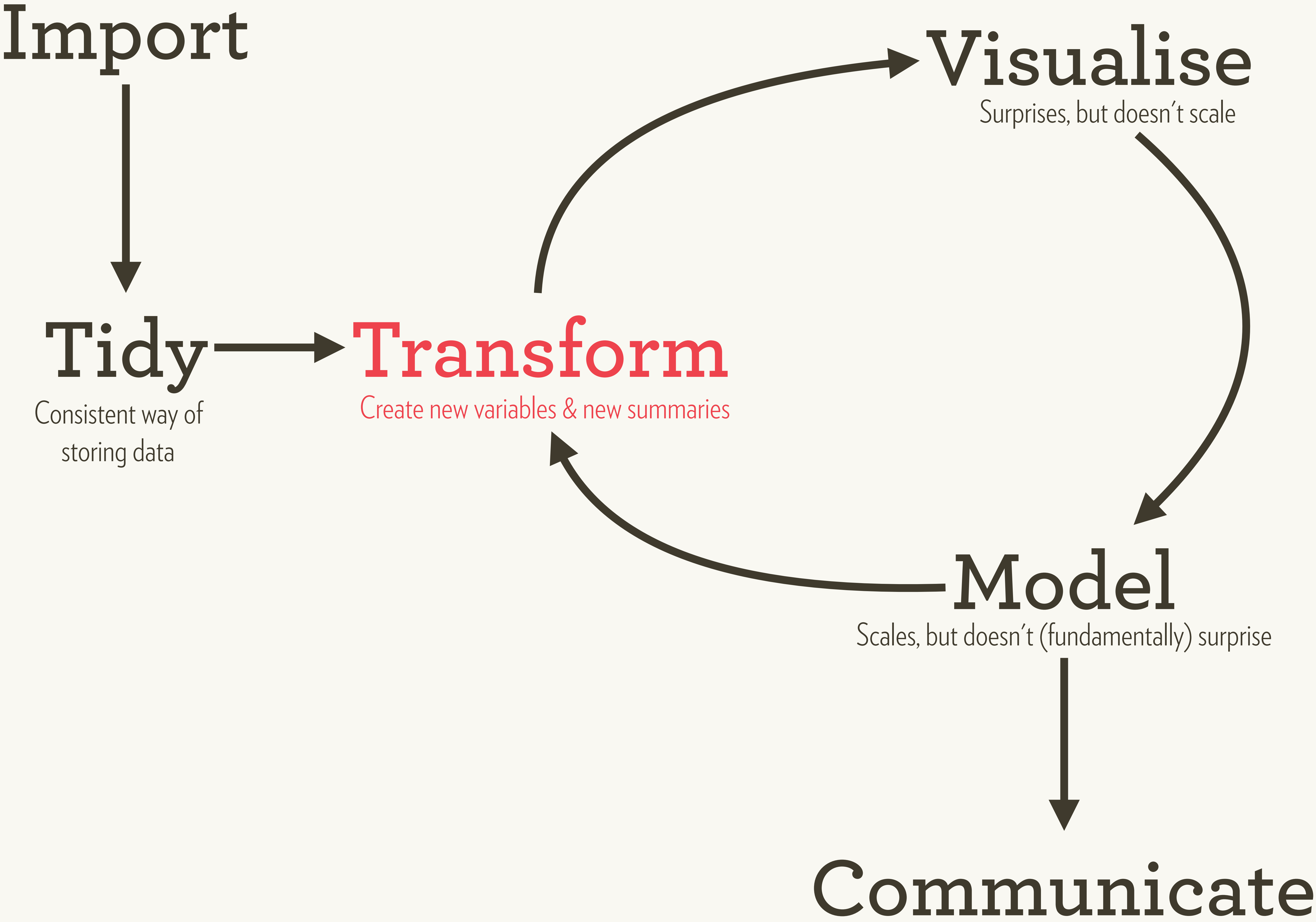
Back to the problem

```
pop_raw2 <- pop_raw %>%
  separate(
    GRD_ID,
    c("grid", "NS", "lat", "EW", "lon"),
    c(3, 4, 8, 9),
    convert = TRUE
  ) %>%
  mutate(
    lat = lat / 100 * ifelse(NS == "S", -1, 1),
    lon = lon / 100 * ifelse(EW == "W", -1, 1)
  ) %>%
  select(-EW, -NS)
```

Which yields:

```
# A tibble: 2,108,409 x 9
   TOT_P  grid   lat   lon CNTR_CODE METHD_CL  YEAR DATA_SRC TOT_P_CON_DT
   <int> <chr> <dbl> <dbl>     <chr>    <chr> <int>    <chr>        <chr>
 1     8   1km 26.89 43.37        DE        A  2011       DE        other
 2     7   1km 26.89 43.41        DE        A  2011       DE        other
 3     3   1km 26.90 43.41        DE        A  2011       DE        other
 4     3   1km 26.91 43.40        DE        A  2011       DE        other
 5    22   1km 26.91 43.41        DE        A  2011       DE        other
 6    20   1km 26.92 43.41        DE        A  2011       DE        other
 7     9   1km 26.92 43.44        DE        A  2011       DE        other
 8    28   1km 26.93 43.40        DE        A  2011       DE        other
 9     8   1km 26.93 43.41        DE        A  2011       DE        other
10     3   1km 26.93 43.43        DE        A  2011       DE        other
11    12   1km 26.94 43.40        DE        A  2011       DE        other
12    12   1km 26.94 43.43        DE        A  2011       DE        other
13    15   1km 26.95 43.40        DE        A  2011       DE        other
14     7   1km 26.95 43.43        DE        A  2011       DE        other
# ... with 2,108,395 more rows
```

Transform



Import

Tidy

Consistent way of storing data

Transform

Create new variables & new summaries

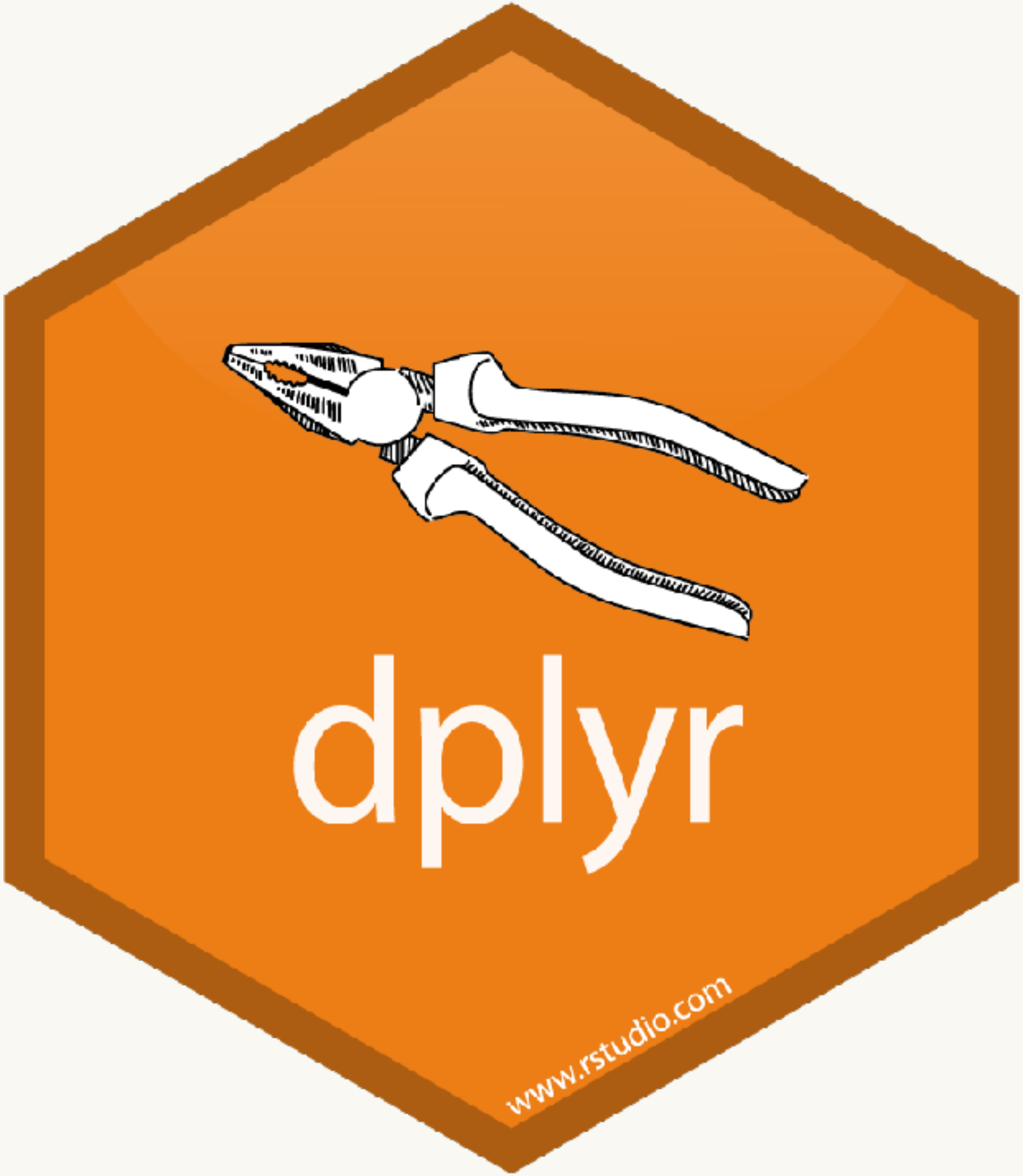
Visualise

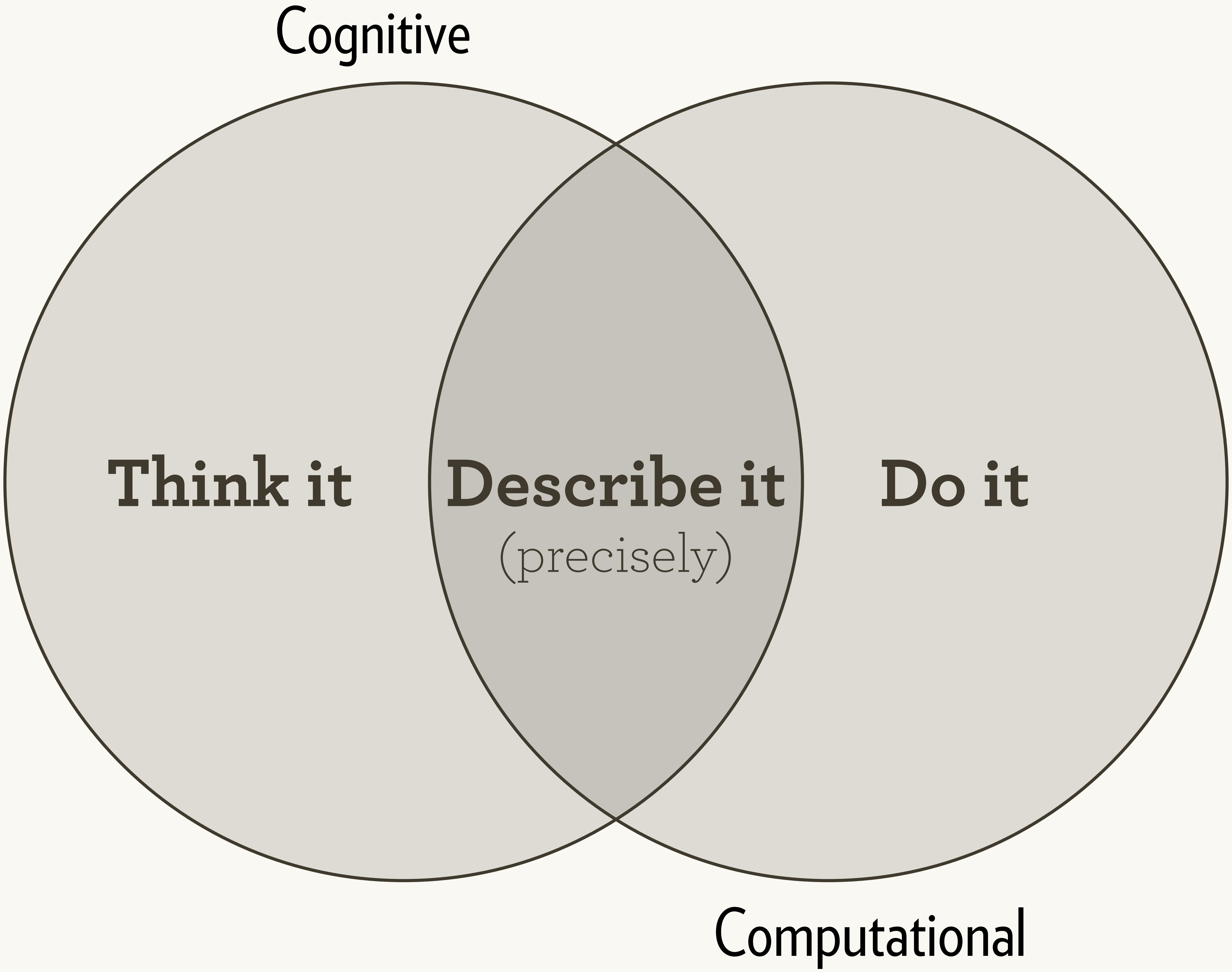
Surprises, but doesn't scale

Model

Scales, but doesn't (fundamentally) surprise

Communicate


dplyr
www.rstudio.com



Cognitive

Think it

Describe it

(precisely)

Do it

Computational

5 verbs solve 90% of data manipulation challenges

+ group by

select: subset variables by name

filter: subset observations by value

mutate: add new variables

summarise: reduce to a single obs

arrange: re-order the observations

We sum population in 0.1° × 0.1° bins

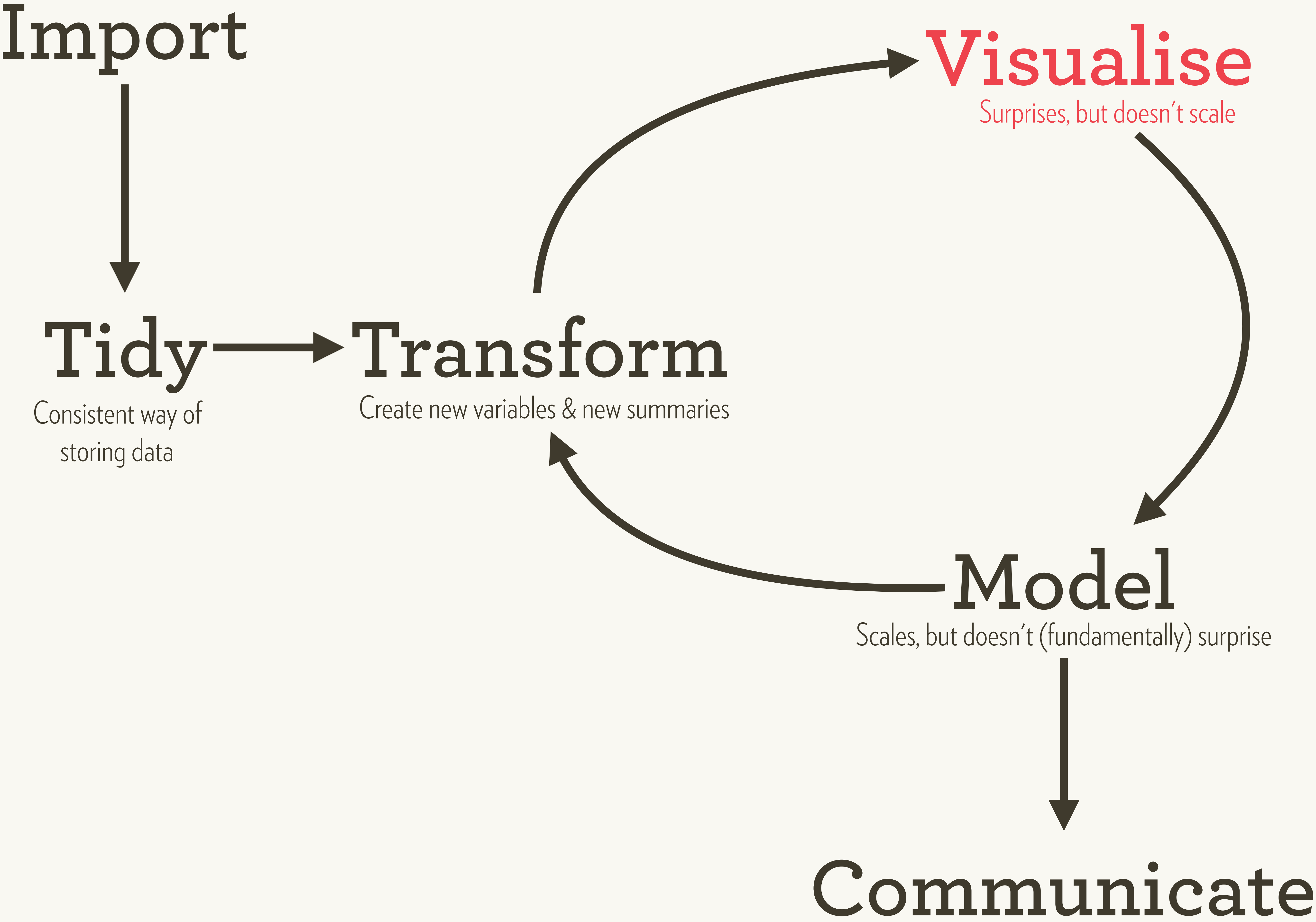
```
pop_sum <- pop_raw2 %>%
 group_by(
    lat = round(lat, 1),
    lon = round(lon, 1)
  ) %>%
  summarize(
    value = sum(TOT_P, na.rm = TRUE)
  )
```

This yields a much smaller dataset

```
Source: local data frame [49,974 x 3]
Groups: lat

     lat   lon  value
   <dbl> <dbl>  <int>
1   13.9  45.5     28
2   13.9  45.6   5659
3   14.3  45.8    416
4   14.3  47.2  24153
5   14.3  47.3  97686
6   14.3  47.4  14082
7   14.3  56.1     47
8   14.3  56.2    105
9   14.4  47.1      6
10  14.4  47.2  79548
# ... with 49,964 more rows
```

Visualise



Import

Tidy

Consistent way of
storing data

Transform

Create new variables & new summaries

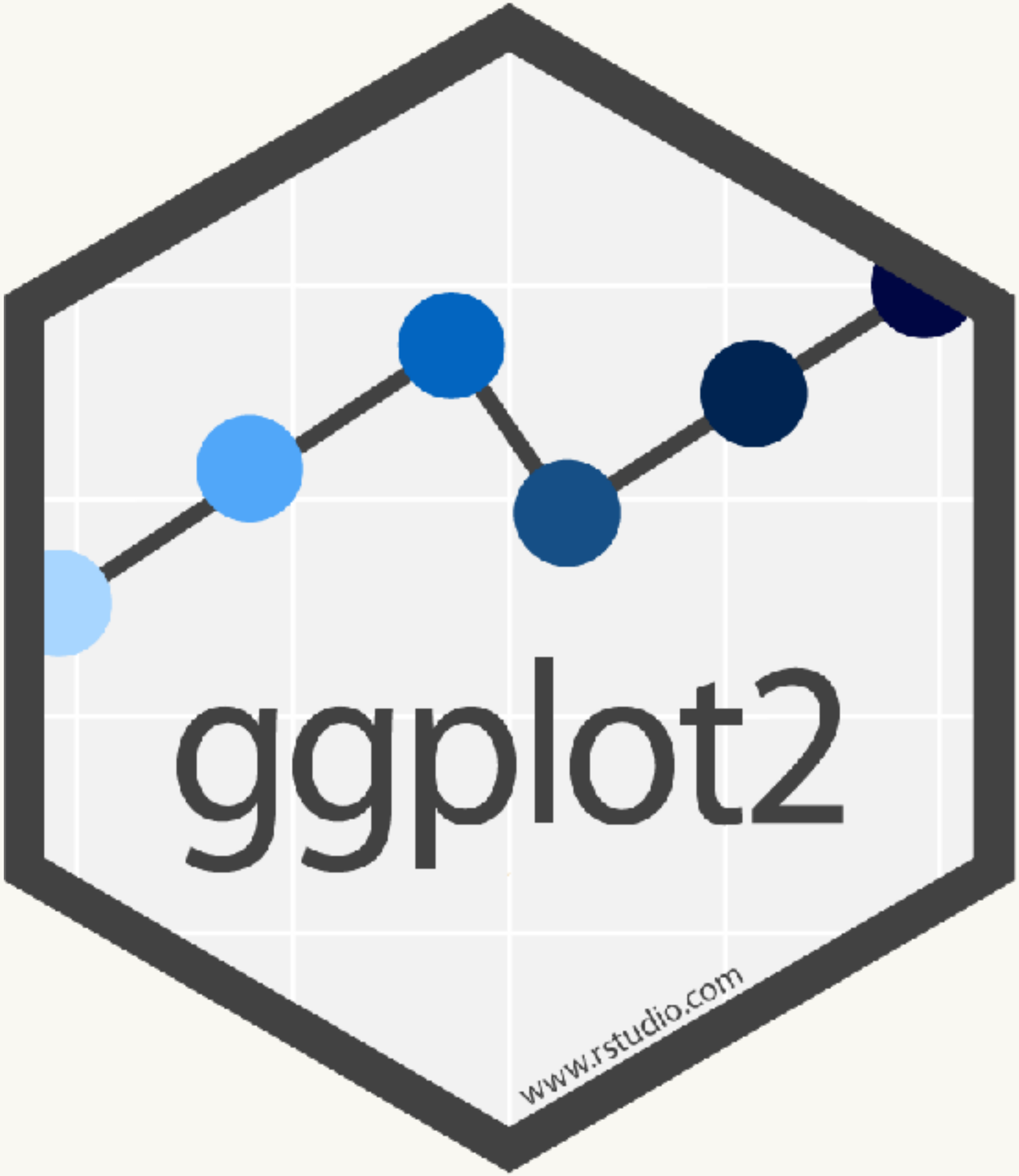
Visualise

Surprises, but doesn't scale

Model

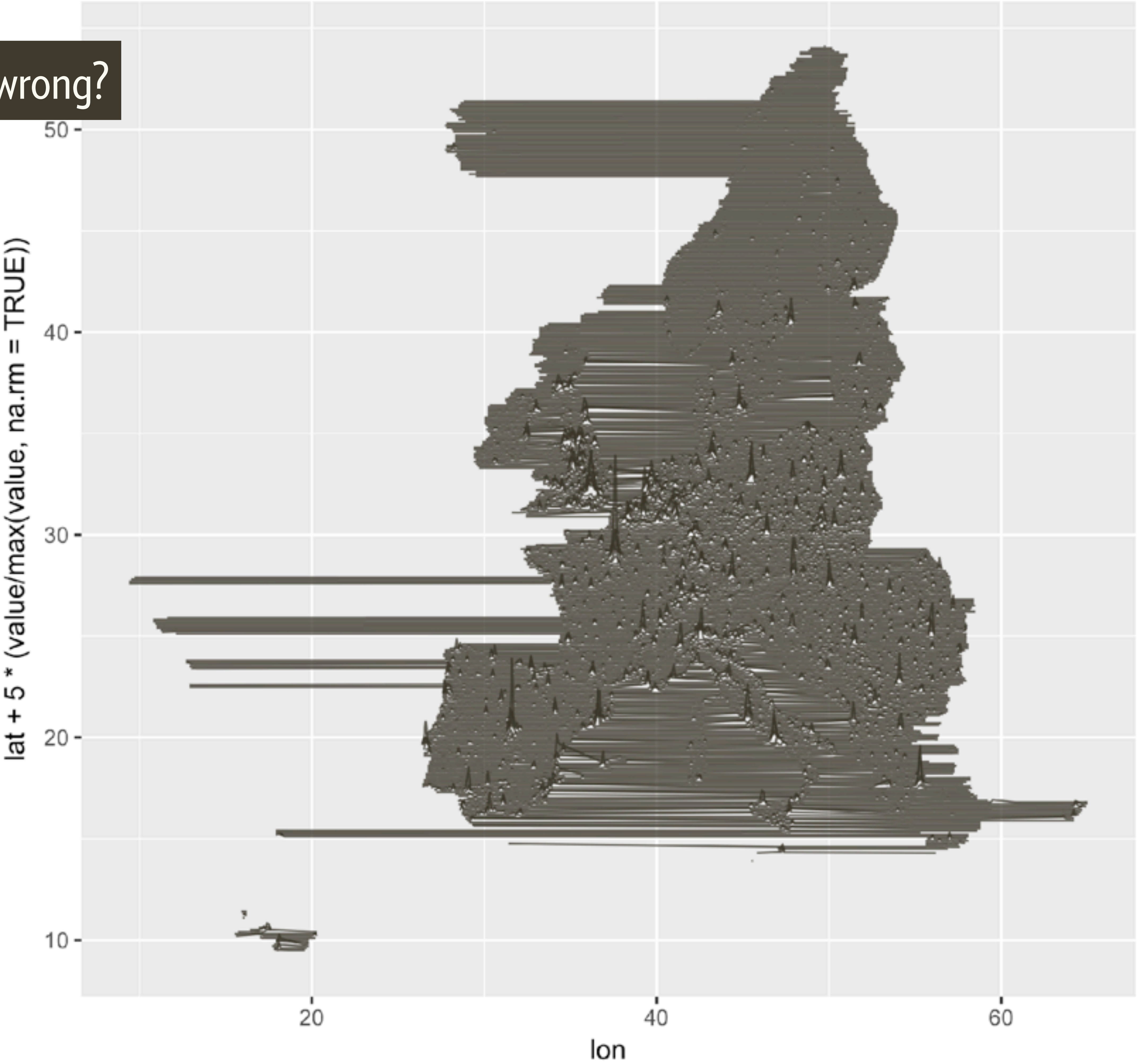
Scales, but doesn't (fundamentally) surprise

Communicate


ggplot2
www.rstudio.com

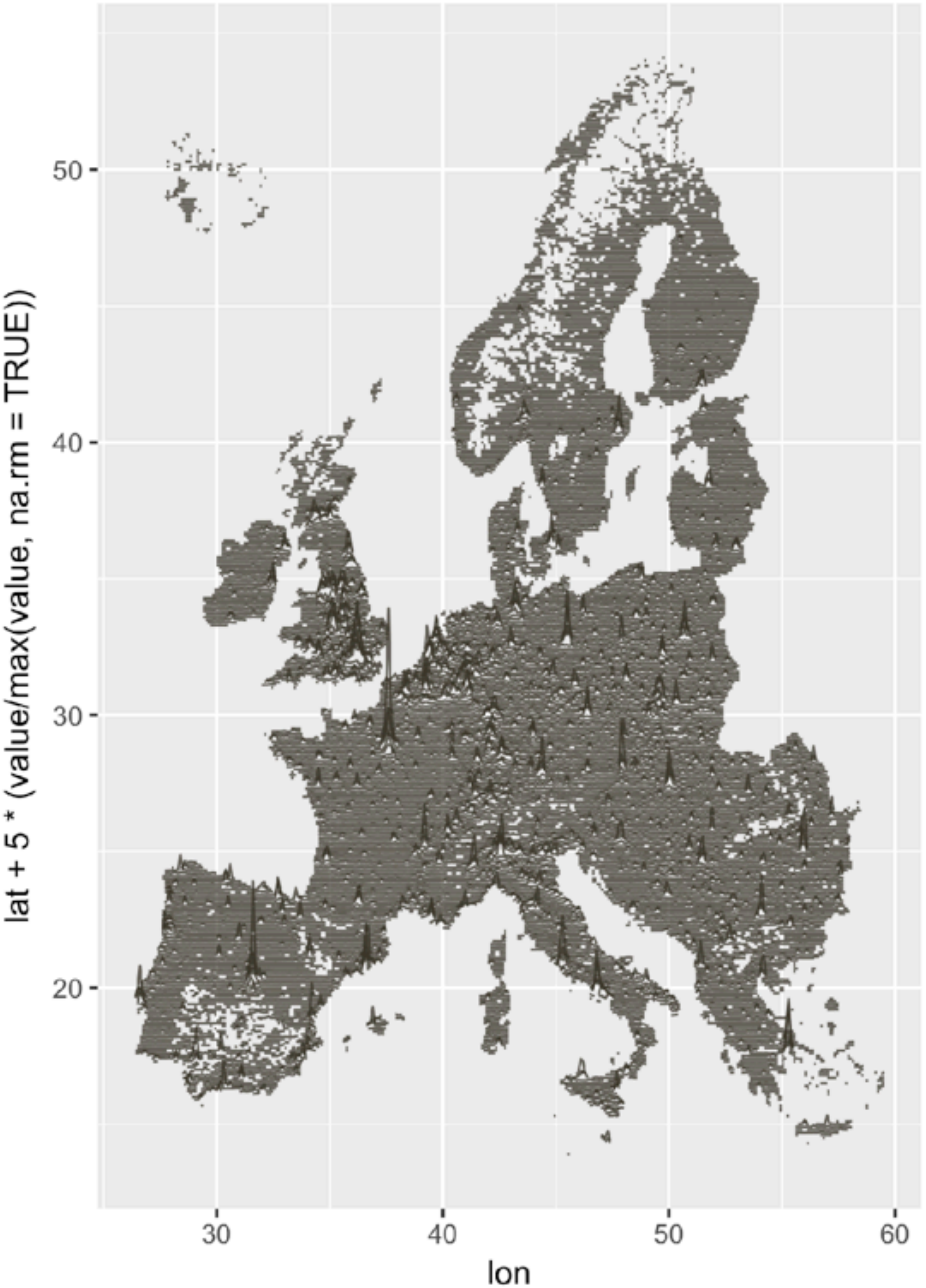
```r
pop_sum %>%
  ggplot(aes(
    x = lon,
    y = lat + 5 * rescale01(value),
    group = lat)
  ) +
  geom_line(
    size = 0.4,
    alpha = 0.8,
    color = "#3F3A2D"
  ) +
  coord_quickmap()
```

What’s wrong?



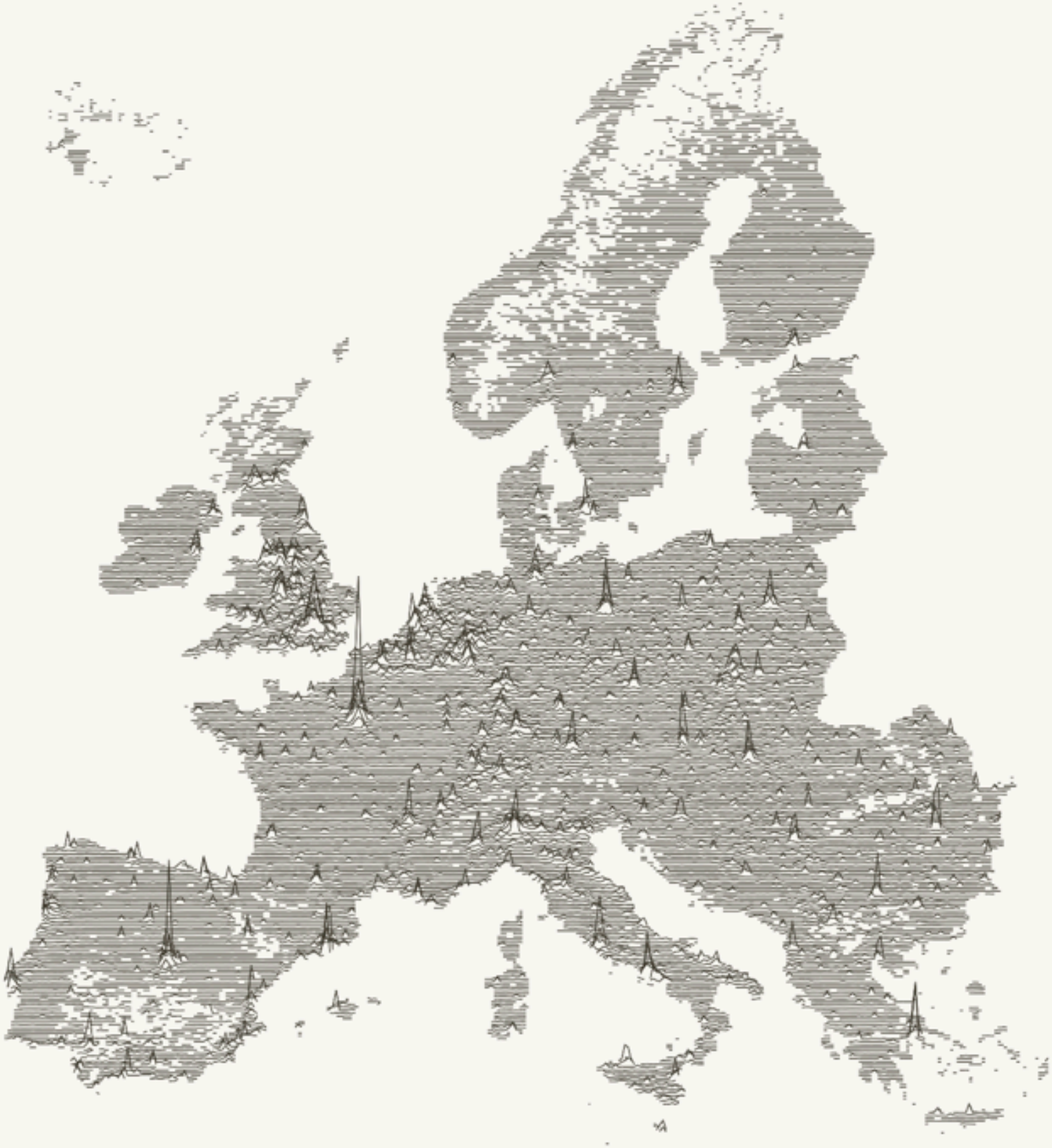
Inevitably, 1st visualisation reveals data problem

```
pop_sum2 <- pop_sum %>%
  ungroup() %>%
  filter(lon > 25, lon < 60) %>%
  complete(lat, lon)
```


lat + 5 * (value/max(value, na.rm = TRUE))
50
40
30
20
30
40
50
60
lon

From exploration to exposition

```
... +
    ggthemes::theme_map() +
    theme(
      panel.background = element_rect(
        fill = "#F9F8F2",
        colour = NA
      )
    ) +
    coord_equal(0.9)
```



Conclusion

Solve complex problems by combining simple pieces

https://unsplash.com/photos/tjX_sniNzgQ

Create a pit of success

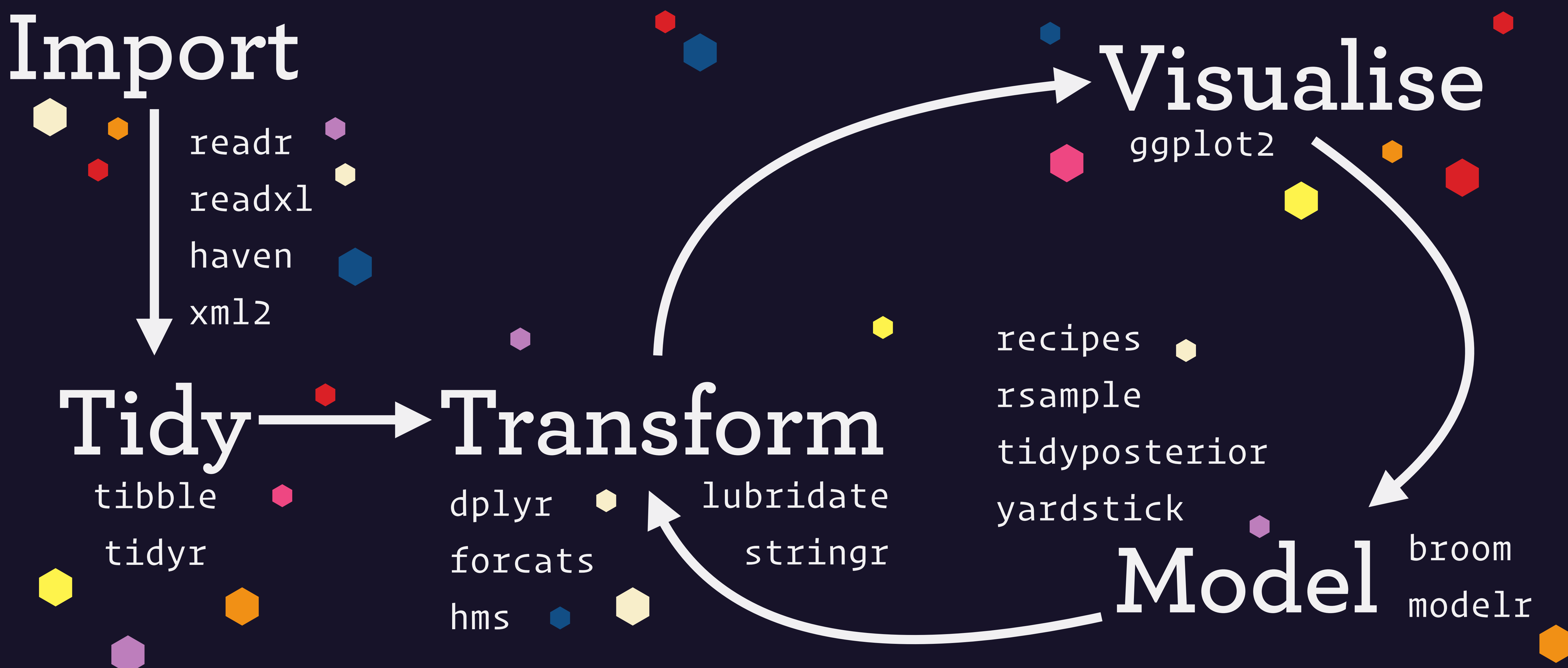
http://blog.codinghorror.com/falling-into-the-pit-of-success/

Embrace humanity

https://unsplash.com/@jamesmayer?photo=Wr3HGvx_RSM

https://unsplash.com/photos/LqNazx_i4UY

Communicate clearly



Import

readr

readxl

haven

xml2

Tidy

tibble

tidyr

Transform

dplyr

forcats

hms

lubridate

stringr

Visualise

ggplot2

recipes

rsample

tidyposterior

yardstick

Model

broom

modelr

purrr

magrittr

Program

R packages for data science

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying philosophy and common APIs.

Install the complete tidyverse with:

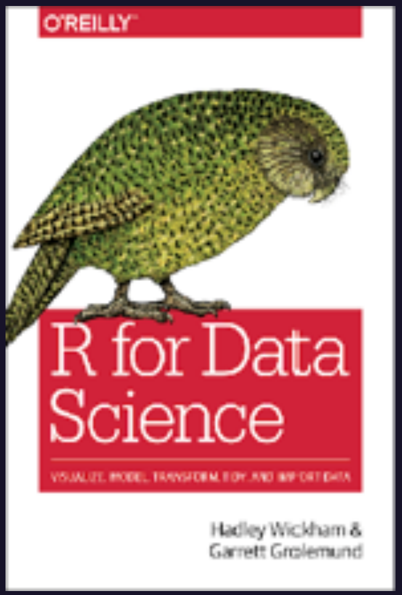
```
install.packages("tidyverse")
```

Learn the tidyverse

See how the tidyverse makes data science faster, easier and more fun with "R for Data

Building tidy tools

tidyverse.org



O'REILLY

R for Data Science

VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA

Hadley Wickham & Garrett Grolemund

r4ds.had.co.nz