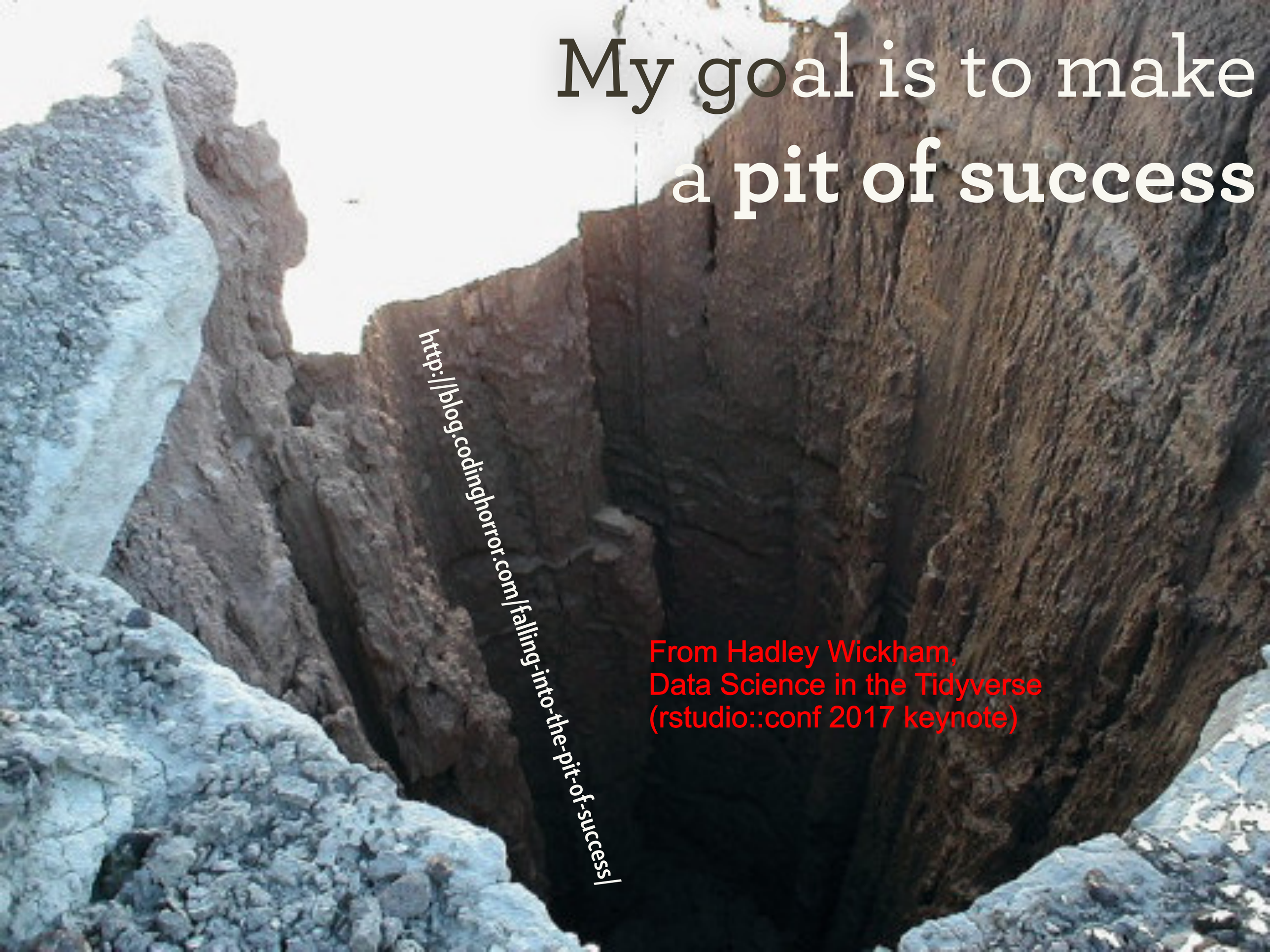
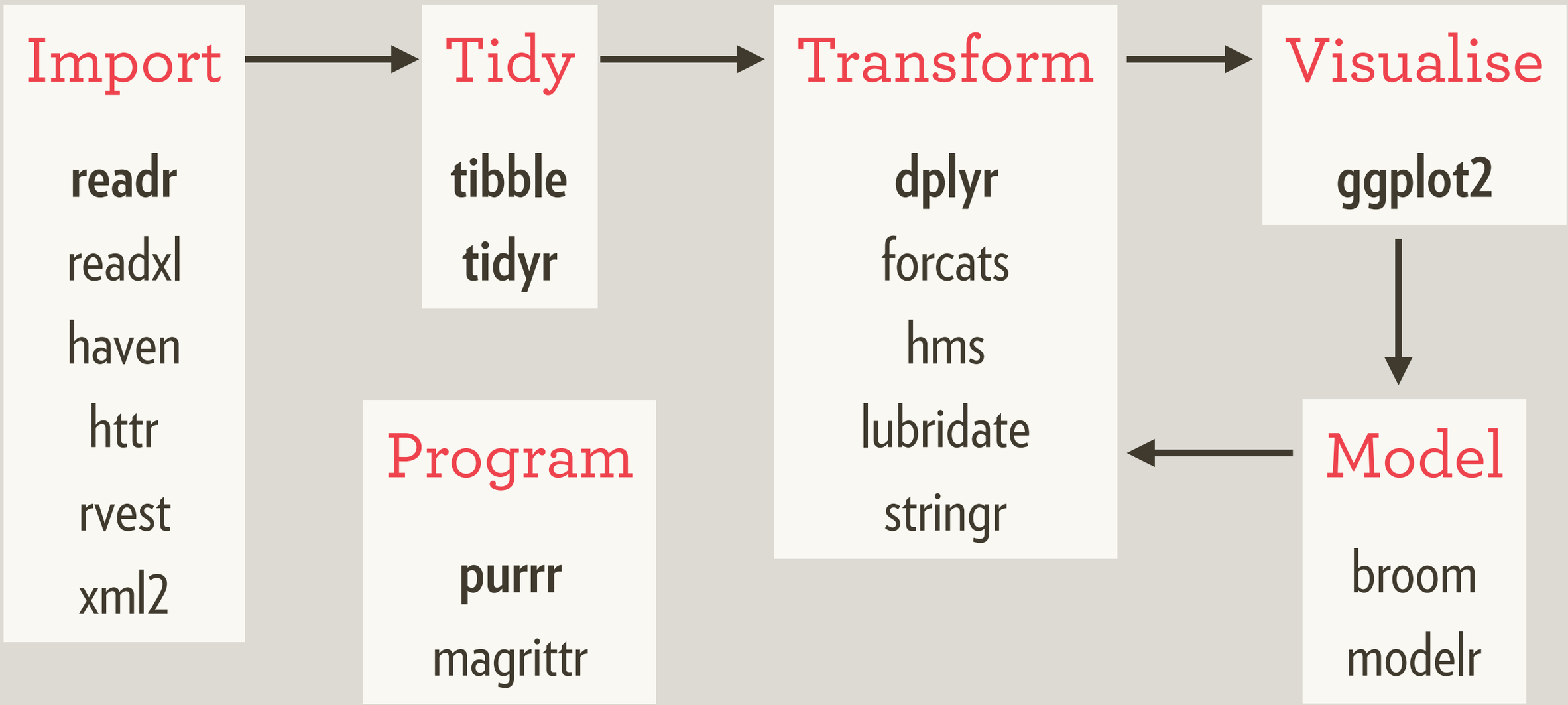

My goal is to make
a **pit of success**
http://blog.codinghorror.com/falling-into-the-pit-of-success/
From Hadley Wickham,
Data Science in the Tidyverse
(rstudio::conf 2017 keynote)


Import
readr
readxl
haven
httr
rvest
xml2
Tidy
tibble
tidyr
Transform
dplyr
forcats
hms
lubridate
stringr
Visualise
ggplot2
Model
broom
modelr
Program
purrr
magrittr

Installing tidyverse installs everything

```
install.packages("tidyverse")

# Instead of
install.packages(c(
  "broom", "dplyr", "feather",
  "forcats","ggplot2", "haven",
  "httr", "hms", "jsonlite",
  "lubridate", "magrittr",
  "modelr", "purrr", "readr",
  "readxl", "stringr", "tibble",
  "rvest", "tidyr", "xml2"
))
```

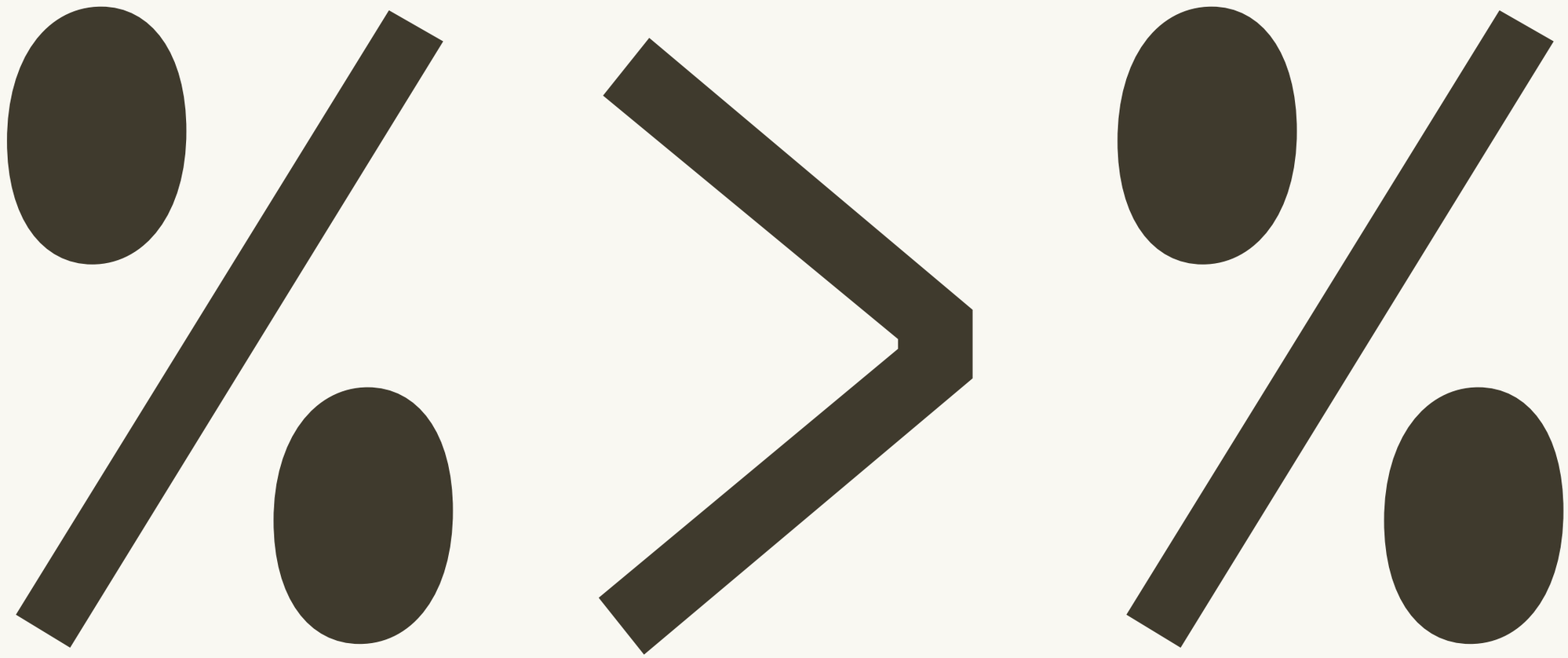
Loading it loads the **core** tidyverse

```
library(tidyverse)

# Instead of:
library(ggplot2)
library(tibble)
library(tidyr)
library(readr)
library(purrr)
library(dplyr)

# These are the packages you use in almost
# every analysis
```

magrittr::



The command-query distinction is useful for pipes

The body is made up of **queries**

Every pipe is ended by a **command**

Where is the command function?

```
flights %>%
  group_by(dest) %>%
  summarise(
    delay = mean(dep_delay, na.rm = TRUE),
    n = n()
) %>%
filter(n > 100) %>%
arrange(desc(delay))
```

In the absence of a command, R prints

```
flights %>%
  group_by(dest) %>%
  summarise(
    delay = mean(dep_delay, na.rm = TRUE),
    n = n()
) %>%
filter(n > 100) %>%
arrange(desc(delay)) %>%
print()
```

Another common command is **assign**

```
flights %>%
  group_by(dest) %>%
  summarise(
    delay = mean(dep_delay, na.rm = TRUE),
    n = n()
) %>%
filter(n > 100) %>%
arrange(desc(delay)) ->
dest_delays
```

But leading with assignment improves readability

```
dest_delays <- flights %>%
  group_by(dest) %>%
  summarise(
    delay = mean(dep_delay, na.rm = TRUE),
    n = n()
  ) %>%
  filter(n > 100) %>%
  arrange(desc(delay))
```

Functions fit best into a pipe when:

1. The first argument is the “data”
2. The data is the same type across a family of functions

Tidy data

Goal: Solve complex problems by combining simple, **uniform pieces**.

Tidy data is a consistent way of storing data

1. Each **dataset** goes in a **data frame**.

2. Each **variable** goes in a **column**.

Happy families are all alike;
every unhappy family is
unhappy in its own way

— *Leo Tolstoy*

Tidy datasets are all alike;
every messy dataset is
messy in its own way

— *Hadley Wickham*

Messy data has a varied shape

```
# A tibble: 5,769 × 22
   iso2  year   m04  m514  m014 m1524 m2534 m3544 m4554 m5564   m65    mu   f04  f514  f014 f1524
   <chr> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
1  AD    1989    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
2  AD    1990    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
3  AD    1991    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
4  AD    1992    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
5  AD    1993    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
6  AD    1994    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
7  AD    1996    NA    NA     0     0     0     4     1     0     0    NA    NA    NA     0     1
8  AD    1997    NA    NA     0     0     1     2     2     1     6    NA    NA    NA     0     1
9  AD    1998    NA    NA     0     0     0     1     0     0     0    NA    NA    NA    NA    NA
10 AD    1999    NA    NA     0     0     0     1     1     0     0    NA    NA    NA     0     0
11 AD    2000    NA    NA     0     0     1     0     0     0     0    NA    NA    NA    NA    NA
12 AD    2001    NA    NA     0    NA    NA     2     1    NA    NA    NA    NA    NA    NA    NA
13 AD    2002    NA    NA     0     0     0     1     0     0     0    NA    NA    NA     0     1
14 AD    2003    NA    NA     0     0     0     1     2     0     0    NA    NA    NA     0     1
15 AD    2004    NA    NA     0     0     0     1     1     0     0    NA    NA    NA     0     0
16 AD    2005     0     0     0     0     1     1     0     0     0     0     0     0     0     1
17 AD    2006     0     0     0     1     1     2     0     1     1     0     0     0     0     0
# ... with 5,752 more rows, and 6 more variables: f2[illegible]
#   f5564 <int>, f65 <int>, fu <int>
```

What are the variables in this dataset?

(Hint: f = female, u = unknown, 1524 = 15-24)

Tidy data has a uniform shape

```
# A tibble: 35,750 × 5
   country  year   sex   age       n
     <chr> <int> <chr> <chr> <int>
1       AD  1996     f   014       0
2       AD  1996     f  1524       1
3       AD  1996     f  2534       1
4       AD  1996     f  3544       0
5       AD  1996     f  4554       0
6       AD  1996     f  5564       1
7       AD  1996     f    65       0
8       AD  1996     m   014       0
9       AD  1996     m  1524       0
10      AD  1996     m  2534       0
# ... with 35,740 more rows
```

tidytext

by Julia Silge & David Robinson

The family of Dashwood had long been settled in Sussex. Their estate was large, and their residence was at Norland Park, in the centre of their property, where, for many generations, they had lived in so respectable a manner as to engage the general good opinion of their surrounding acquaintance.

— *Sense & Sensibility*, **Jane Austen**

tidytext provides an answer

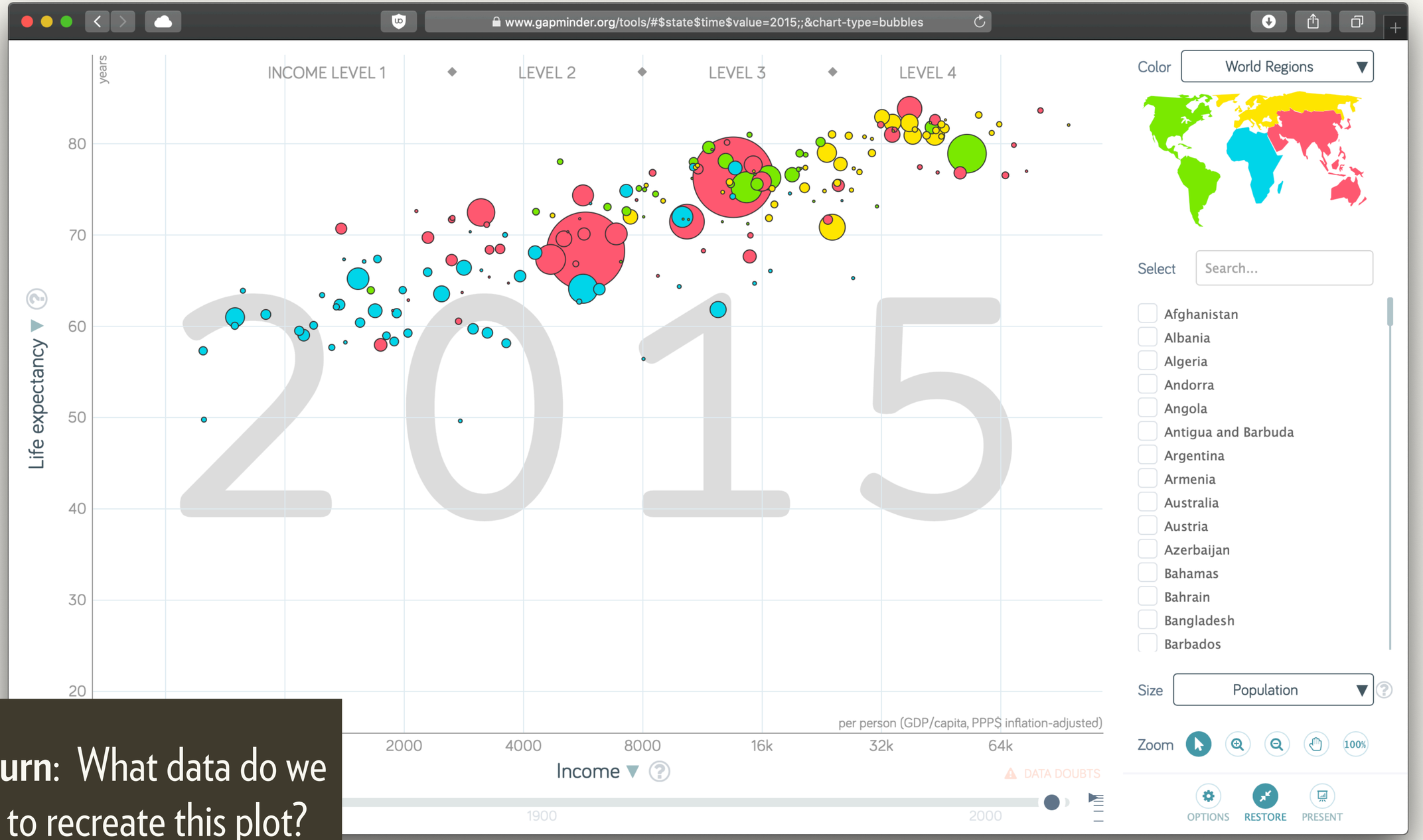
```
# A tibble: 724,880 × 4
                  book linenumber chapter     word
                <fctr>      <int>   <int>    <chr>
1  Sense & Sensibility         10       1  chapter
2  Sense & Sensibility         10       1        1
3  Sense & Sensibility         13       1      the
4  Sense & Sensibility         13       1   family
5  Sense & Sensibility         13       1       of
6  Sense & Sensibility         13       1 dashwood
7  Sense & Sensibility         13       1      had
8  Sense & Sensibility         13       1     long
9  Sense & Sensibility         13       1     been
10 Sense & Sensibility         13       1  settled
# ... with 724,870 more rows
```

Welcome to the tidyverse

February 2019

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

The tidyverse is a **language** for solving data science challenges with R code.

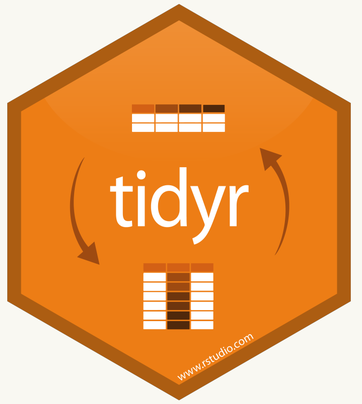


Your turn: What data do we need to recreate this plot?

We need five variables

```
# A tibble: 193 x 6
   country              four_regions  year income life_exp      pop
   <chr>                <chr>        <int>  <int>    <dbl>    <int>
 1 Afghanistan          asia          2015   1750     57.9 33700000
 2 Albania              europe        2015  11000     77.6  2920000
 3 Algeria              africa        2015  13700     77.3 39900000
 4 Andorra              europe        2015  46600     82.5    78000
 5 Angola               africa        2015   6230     64   27900000
 6 Antigua and Barbuda  americas      2015  20100     77.2    99900
 7 Argentina            americas      2015  19100     76.5 43400000
 8 Armenia              europe        2015   8180     75.4  2920000
 9 Australia            asia          2015  43800     82.6 23800000
10 Austria              europe        2015  44100     81.4  8680000
# … with 183 more rows
```

This is “tidy” data



Observation

```
# A tibble: 193 x 6
```

	country	four_regions	year	income	life_exp	pop
	<chr>	<chr>	<int>	<int>	<dbl>	<int>
1	Afghanistan	asia	2015	1750	57.9	33700000
2	Albania	europe	2015	11000	77.6	2920000
3	Algeria	africa	2015	13700	77.3	39900000
4	Andorra	europe	2015	46600	82.5	78000
5	Angola	africa	2015	6230	64	27900000
6	Antigua and Barbuda	americas	2015	20100	77.2	99900
7	Argentina	americas	2015	19100	76.5	43400000
8	Armenia	europe	2015	8180	75.4	2920000
9	Australia	asia	2015	43800	82.6	23800000
10	Austria	europe	2015	44100	81.4	8680000

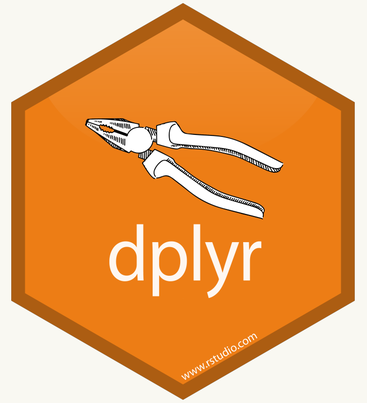
```
# … with 183 more rows
```

Variable

The original data looked like this

```
# A tibble: 193 x 220
   country `1800` `1801` `1802` `1803` `1804` `1805` `1806` `1807` `1808` `1809` `1810` `1811` `1812` `1813`
   <chr>    <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>
 1 Afghan…    603    603    603    603    603    603    603    603    603    603    604    604    604    604
 2 Albania    667    667    667    667    667    668    668    668    668    668    668    668    668    668
 3 Algeria    715    716    717    718    719    720    721    722    723    724    725    726    727    728
 4 Andorra   1200   1200   1200   1200   1210   1210   1210   1210   1220   1220   1220   1220   1220   1230
 5 Angola     618    620    623    626    628    631    634    637    640    642    645    648    651    654
 6 Antigu…    757    757    757    757    757    757    757    758    758    758    758    758    758    758
 7 Argent…   1510   1510   1510   1510   1510   1510   1510   1510   1510   1510   1510   1510   1510   1510
 8 Armenia    514    514    514    514    514    514    514    514    514    514    514    515    515    515
 9 Austra…    814    816    818    820    822    824    825    827    829    831    833    835    837    839
10 Austria   1850   1850   1860   1870   1880   1880   1890   1900   1910   1920   1920   1930   1940   1950
# … with 183 more rows, and 205 more variables: `1814` <int>, `1815` <int>, `1816` <int>, `1817` <int>,
#   `1818` <int>, `1819` <int>, `1820` <int>, `1821` <int>, `1822` <int>, `1823` <int>, `1824` <int>,
#   `1825` <int>, `1826` <int>, `1827` <int>, `1828` <int>, `1829` <int>, `1830` <int>, `1831` <int>,
#   `1832` <int>, `1833` <int>, `1834` <int>, `1835` <int>, `1836` <int>, `1837` <int>, `1838` <int>,
#   `1839` <int>, `1840` <int>, `1841` <int>, `1842` <int>, `1843` <int>, `1844` <int>, `1845` <int>,
#   `1846` <int>, `1847` <int>, `1848` <int>, `1849` <int>, `1850` <int>, `1851` <int>, `1852` <int>, ...
```

Start with single year



Called the pipe; pronounced “then”

Called the reverse assignment operator; pronounced “creates”

```
gapminder %>%
  filter(year == 2015) ->
  gapminder15
```

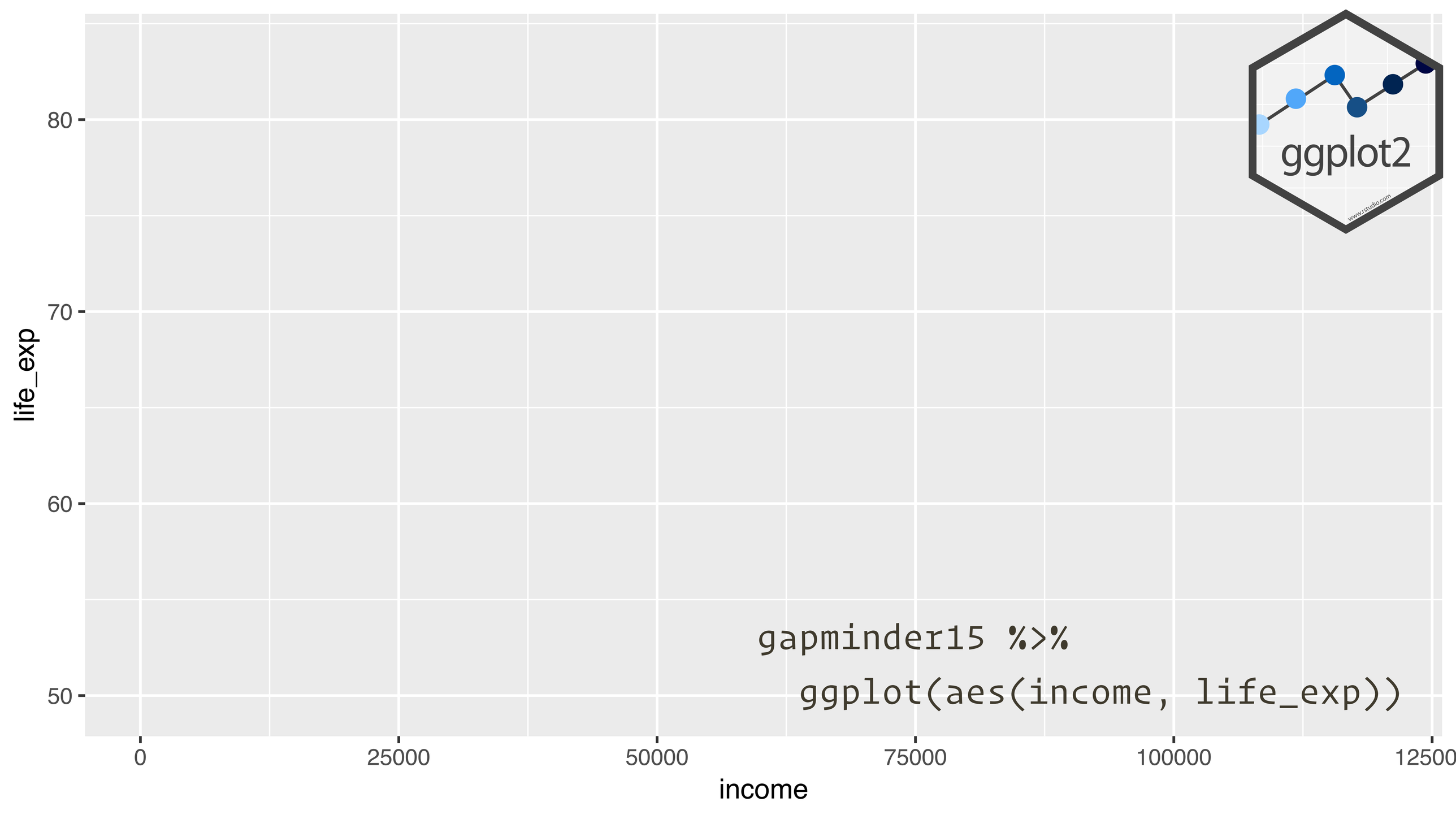
Phonics are important!

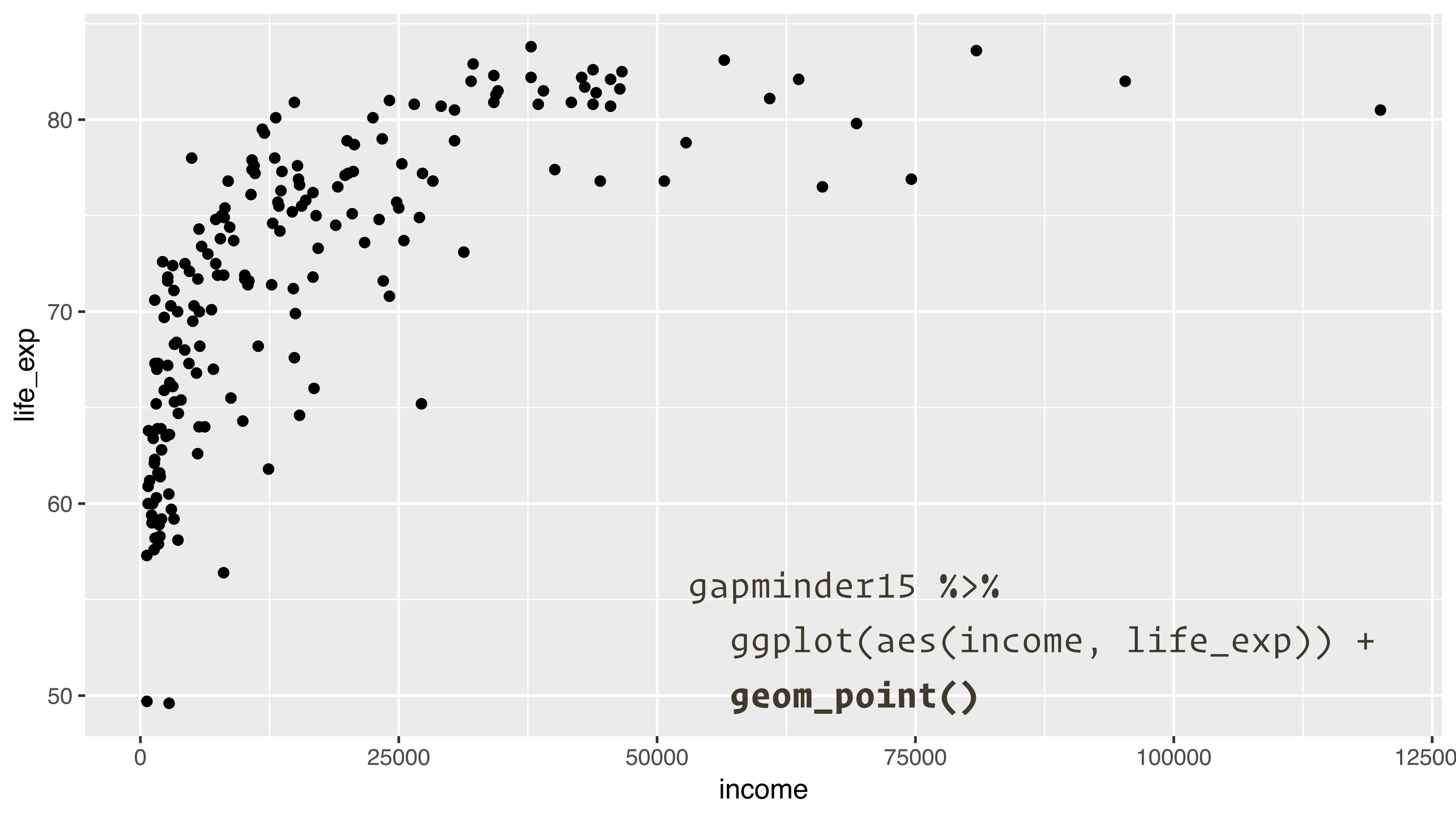
```
gapminder %>%
  filter(year == 2015) ->
  gapminder15
```

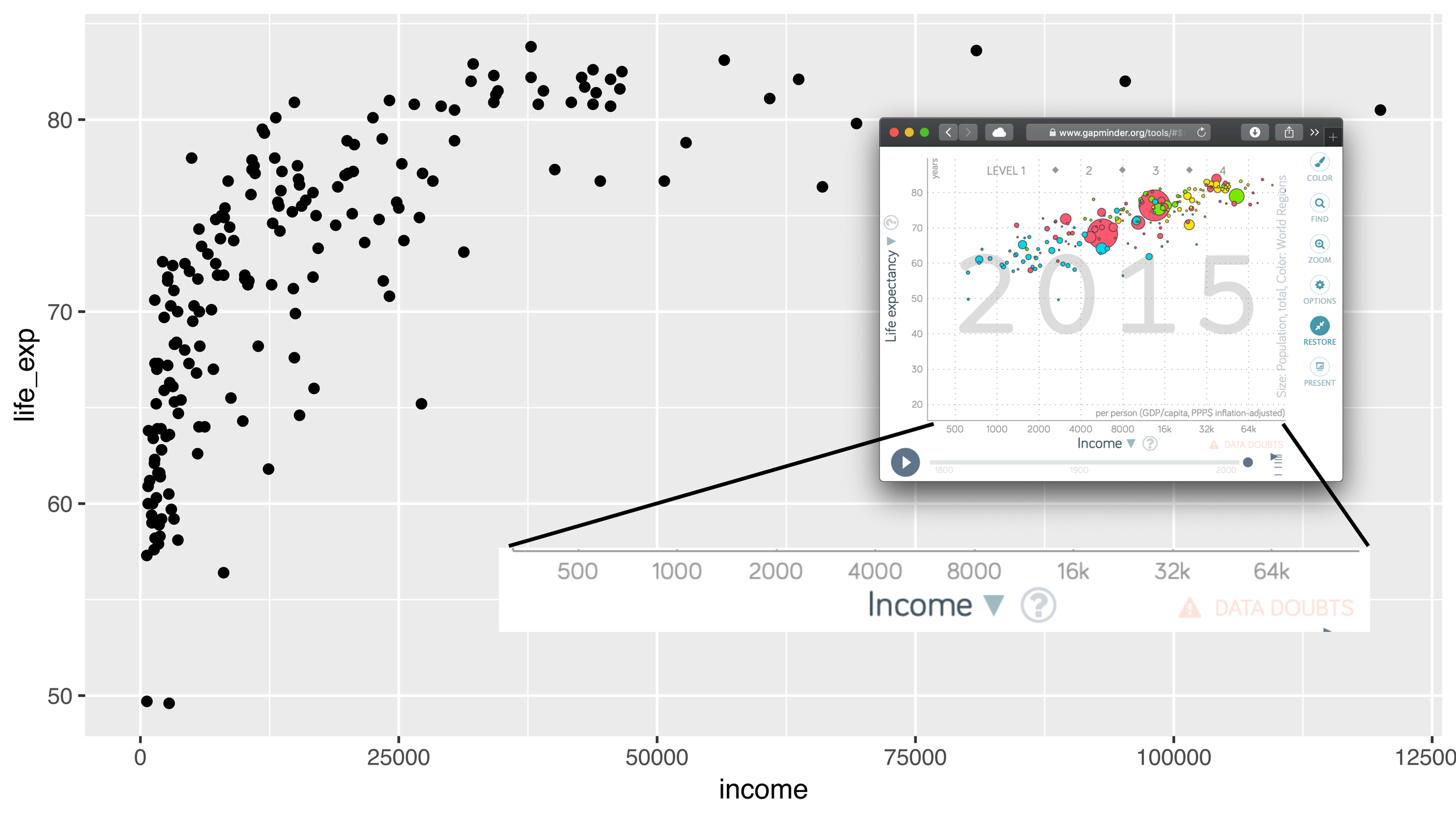
Take the gapminder data, then

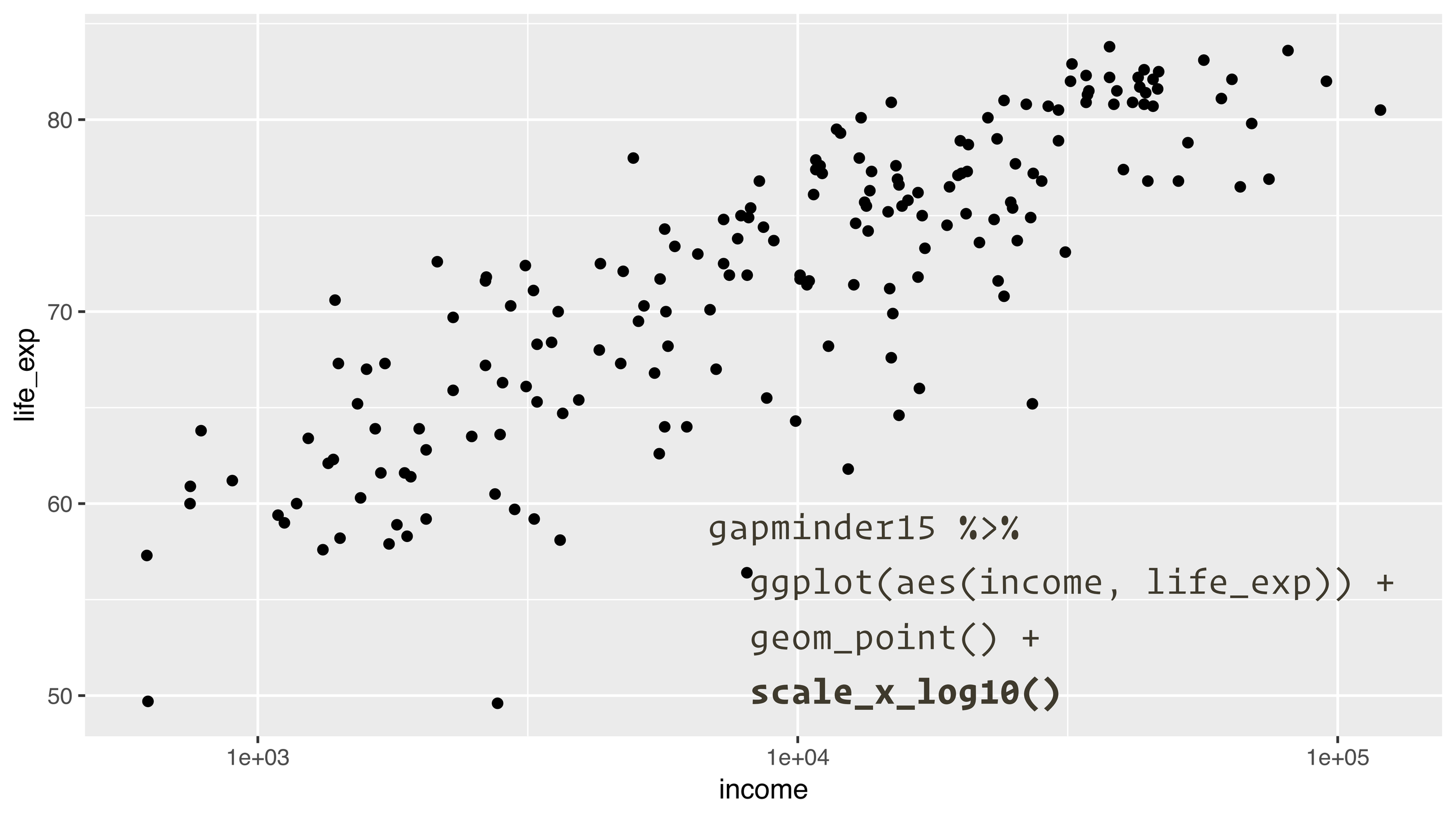
filter rows where year equals 2015, creating

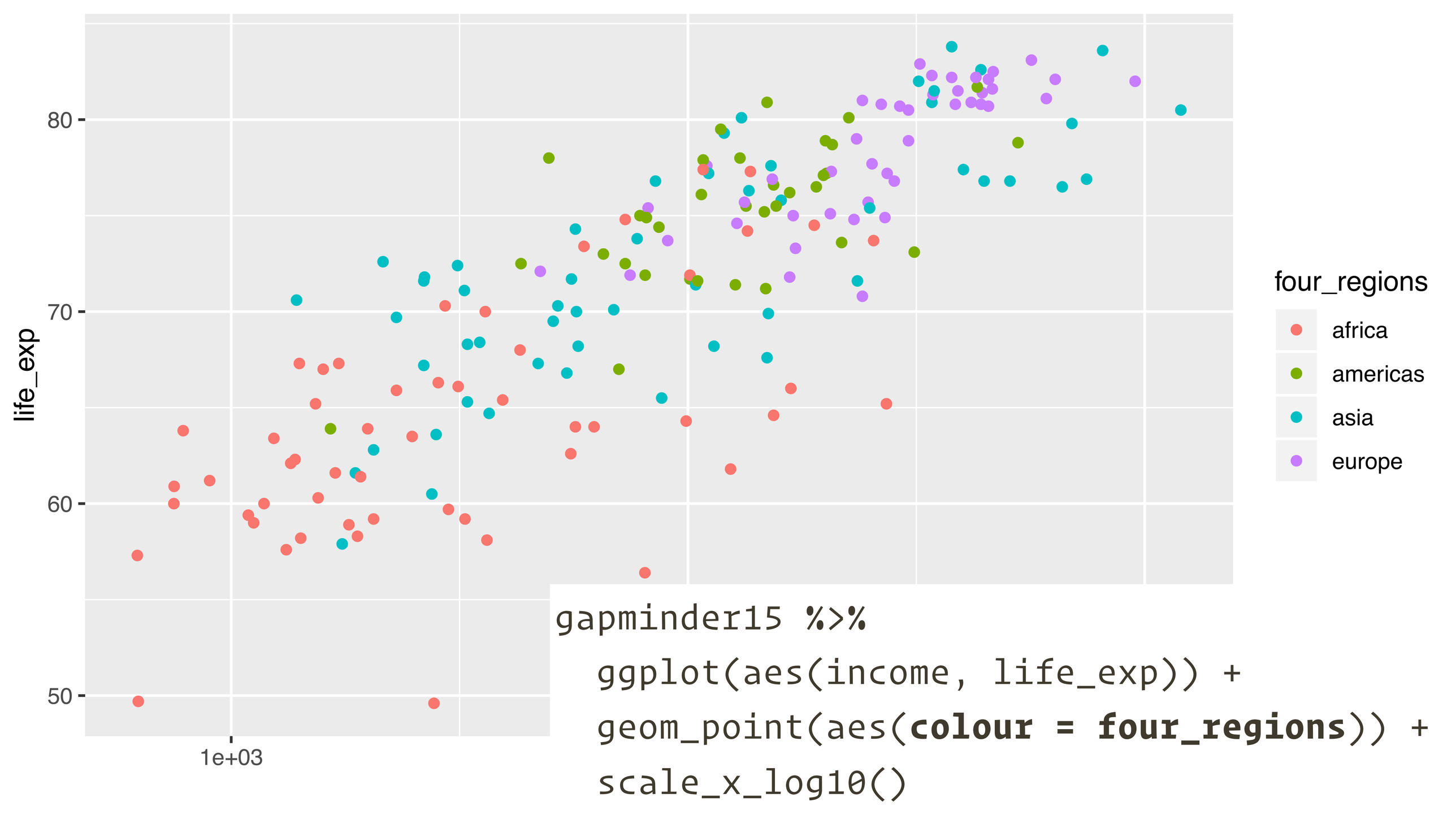
gapminder15 variable


gapminder15 %>%
ggplot(aes(income, life_exp))
life_exp
80
70
60
50
0
25000
50000
75000
100000
12500
income
ggplot2
www.rstudio.com

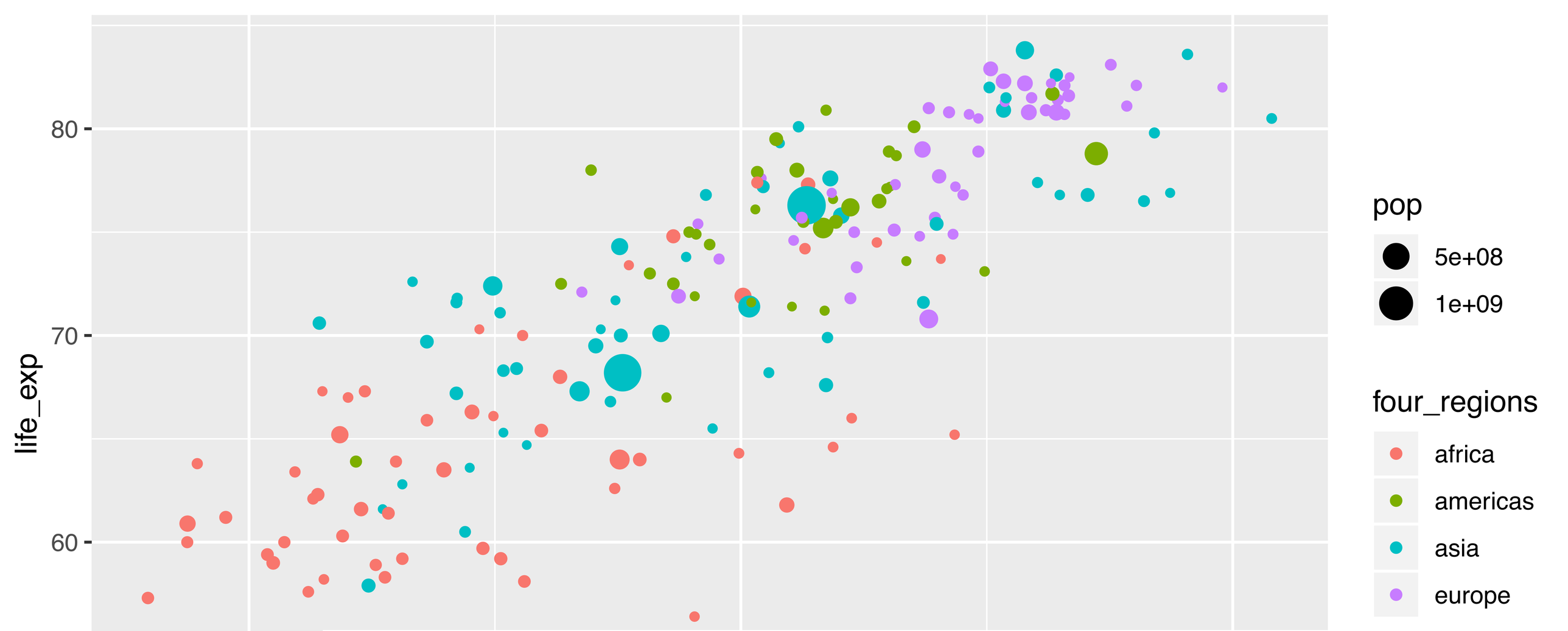

life_exp
80
70
60
50
0
25000
50000
75000
100000
12500
income
gapminder15 %>%
ggplot(aes(income, life_exp)) +
geom_point()


life_exp
80
70
60
50
0
25000
50000
75000
100000
12500
income
www.gapminder.org/tools/#$
LEVEL 1
2
3
4
years
Life expectancy
2015
Size: Population, total, Color: World Regions
COLOR
FIND
ZOOM
OPTIONS
RESTORE
PRESENT
per person (GDP/capita, PPP$ inflation-adjusted)
500
1000
2000
4000
8000
16k
32k
64k
Income
DATA DOUBTS
1800
1900
2000


80
70
60
50
life_exp
1e+03
1e+04
1e+05
income
gapminder15 %>%
ggplot(aes(income, life_exp)) +
geom_point() +
scale_x_log10()

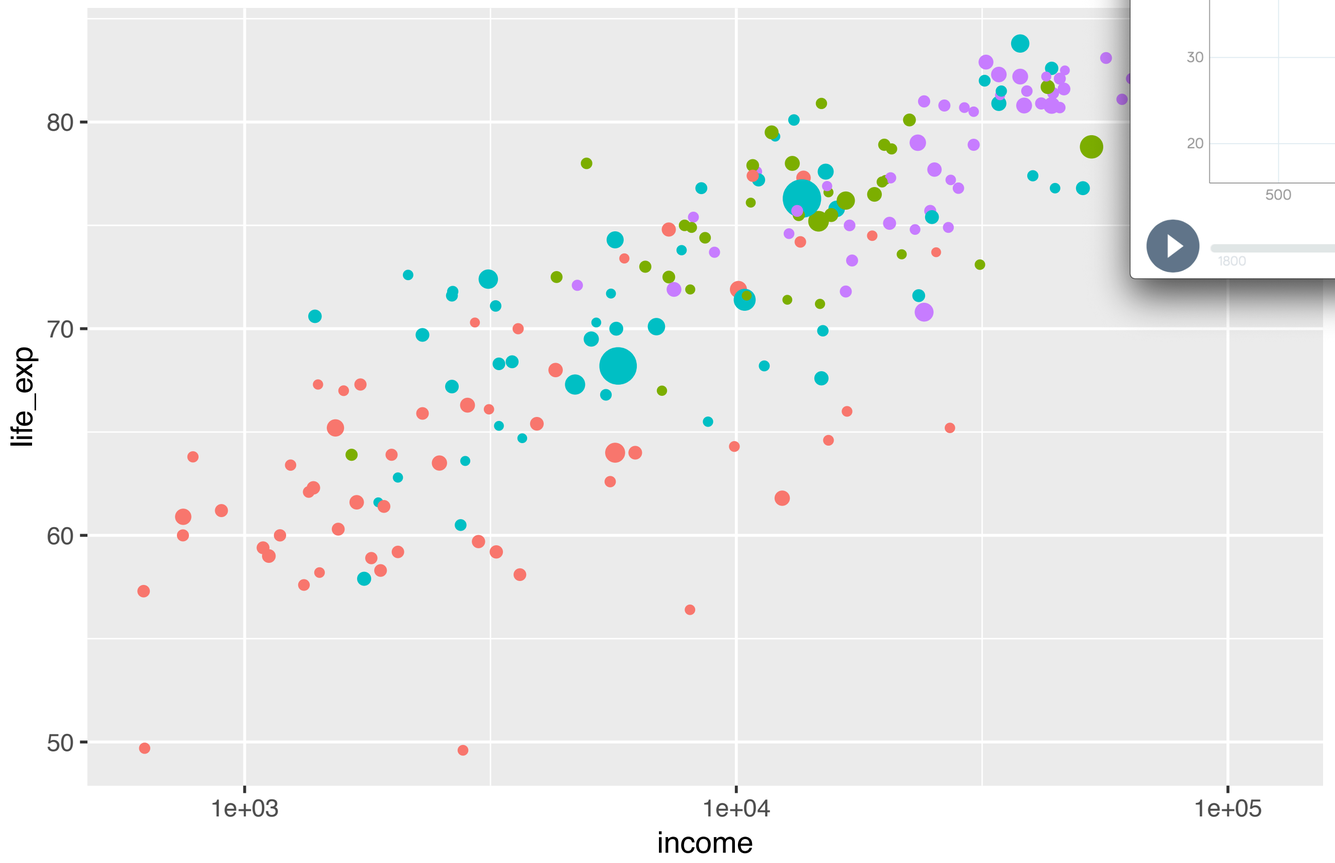


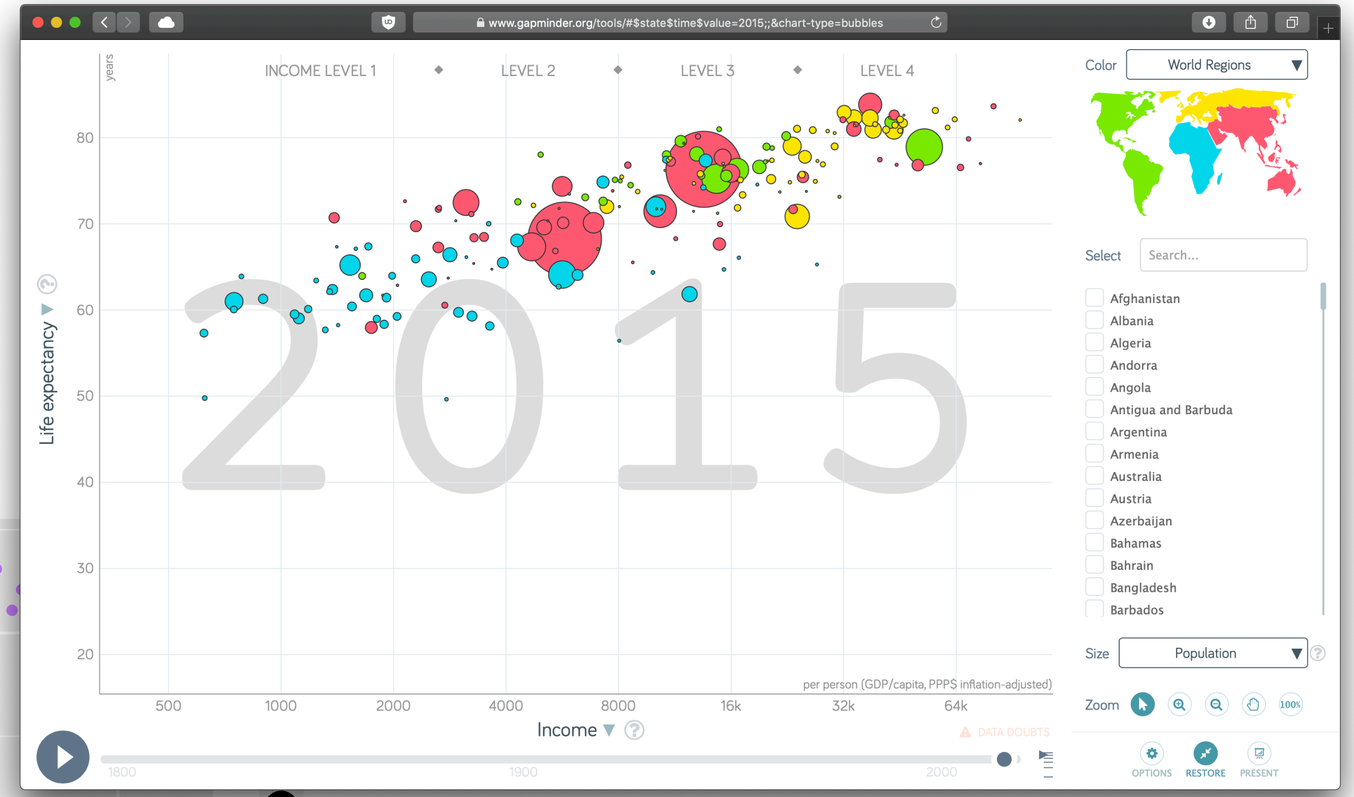

```
gapminder15 %>%
  ggplot(aes(income, life_exp)) +
  geom_point(aes(colour = four_regions)) +
  scale_x_log10()
```




```
gapminder15 %>%
  ggplot(aes(income, life_exp)) +
  geom_point(aes(colour = four_regions, size = pop)) +
  scale_x_log10()
```

Your turn: What's missing?





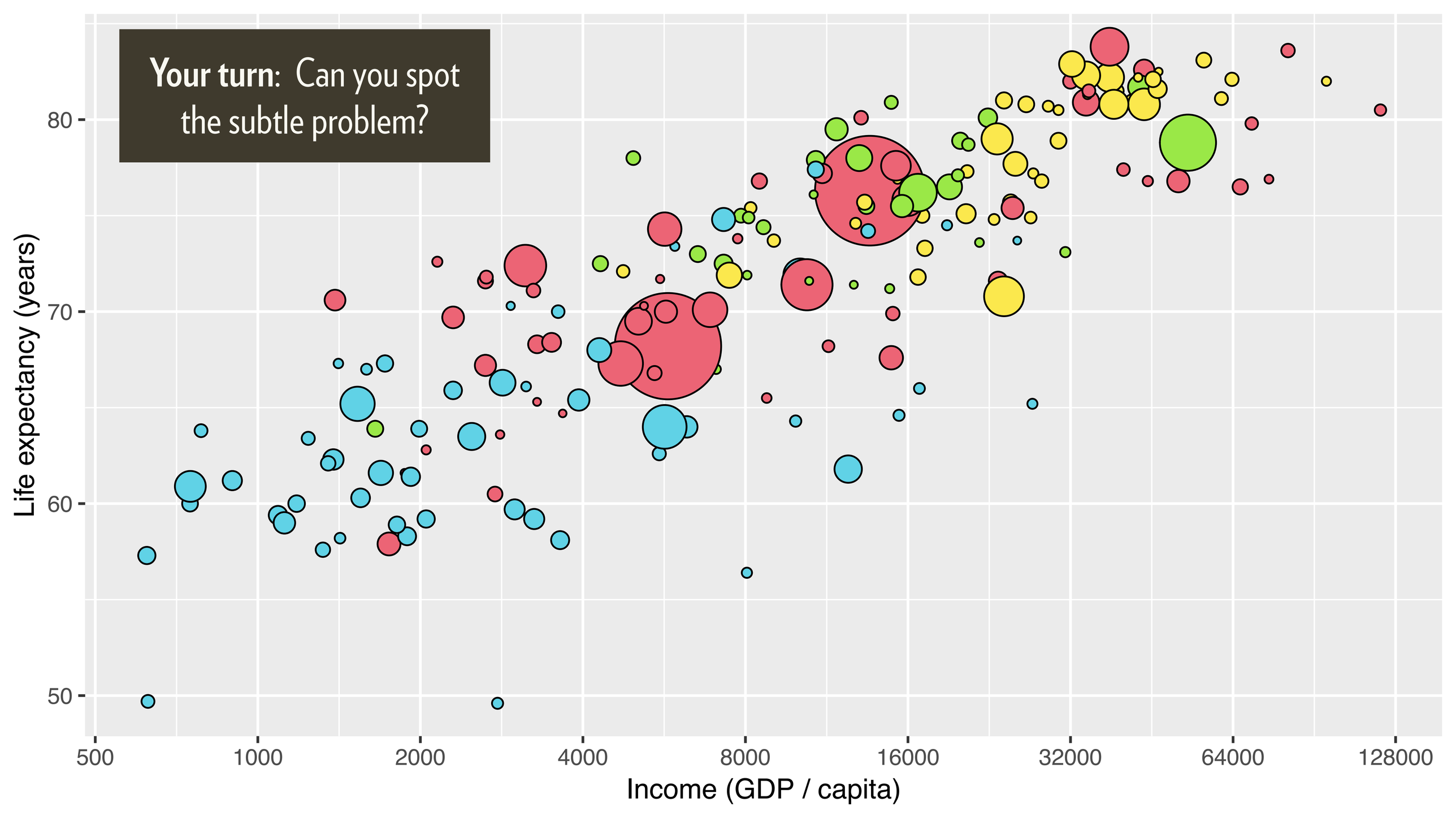
1e+09

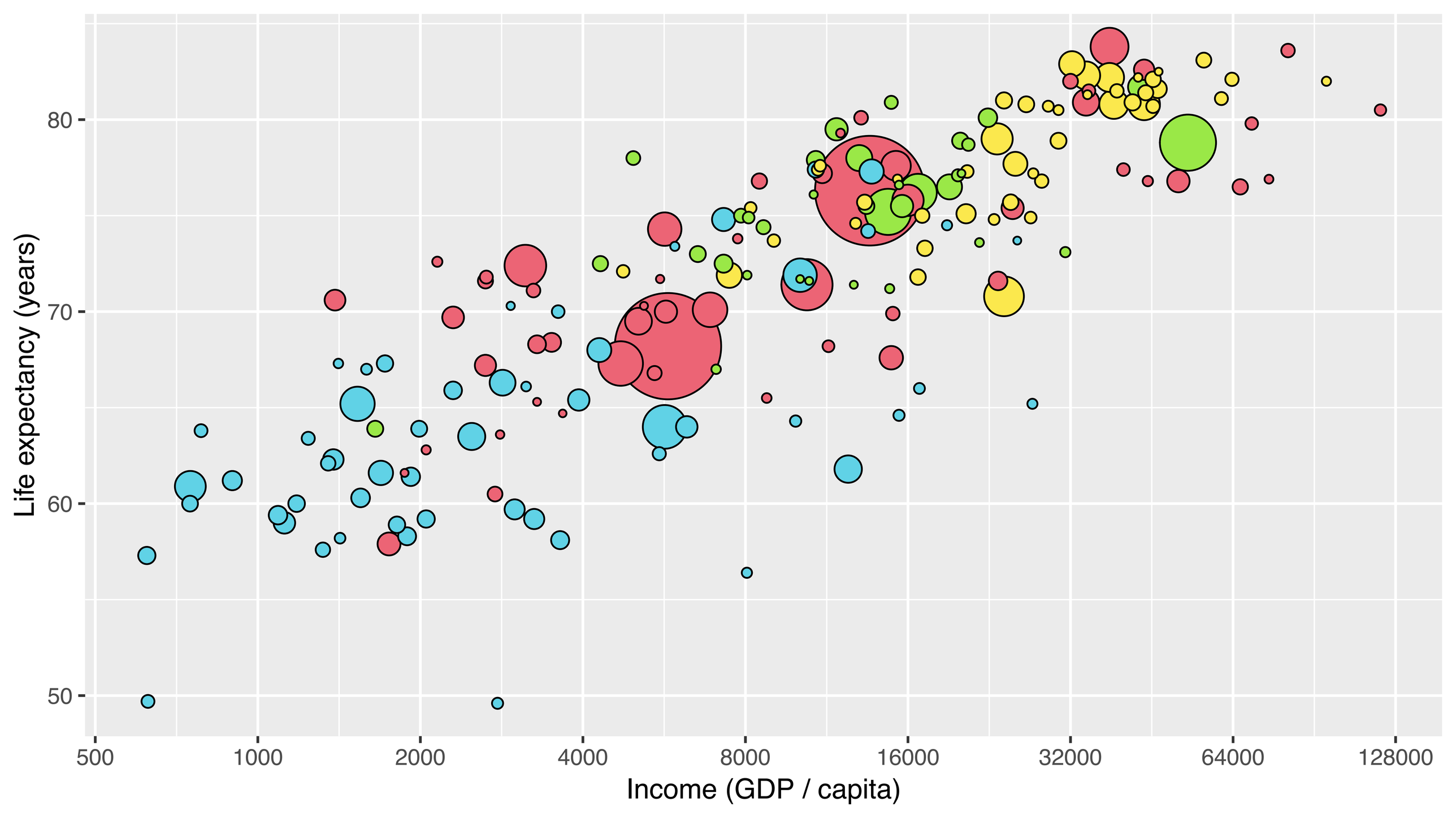
four_regions

- africa
- americas
- asia
- europe

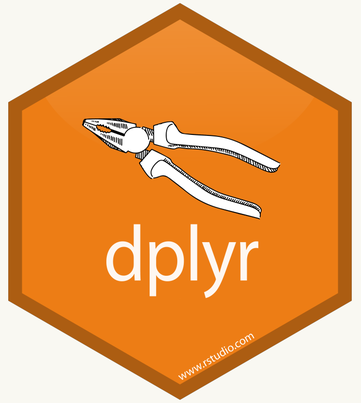
It’s a lot more work to make a expository graphic

```
gapminder15 %>%
  ggplot(aes(income, life_exp)) +
  geom_point(aes(fill = four_regions, size = pop), shape = 21) +
  scale_x_log10(breaks = 2^(-1:7) * 1000) +
  scale_size(range = c(1, 20), guide = FALSE) +
  scale_fill_manual(
    guide = FALSE,
    values = c(
      africa = "#60D2E6",
      americas = "#9AE847",
      asia = "#EC6475",
      europe = "#FBE84D"
    )
  ) +
  labs(
    x = "Income (GDP / capita)",
    y = "Life expectancy (years)"
  )
```


Your turn: Can you spot the subtle problem?
Life expectancy (years)
80
70
60
50
500
1000
2000
4000
8000
16000
32000
64000
128000
Income (GDP / capita)


Life expectancy (years)
80
70
60
50
500
1000
2000
4000
8000
16000
32000
64000
128000
Income (GDP / capita)

A little motivation




```
data %>%
  arrange(desc(pop)) %>%
  ggplot(aes(income, life_exp)) +
  ...
```

You can also turn your code into a function

```
gap_plot <- function(data) {
  data %>%
    arrange(desc(pop)) %>%
    ggplot(aes(income, life_exp)) +
    geom_point(aes(fill = four_regions, size = pop), shape = 21) +
    scale_x_log10(breaks = 2^(-1:7) * 1000) +
    scale_size(range = c(1, 20), guide = FALSE) +
    scale_fill_manual(values = c(
      africa = "#60D2E6",
      americas = "#9AE847",
      asia = "#EC6475",
      europe = "#FBE84D"
    )) +
    labs(
      x = "Income (GDP / capita)",
      y = "Life expectancy",
      fill = "Region"
    )
}
```

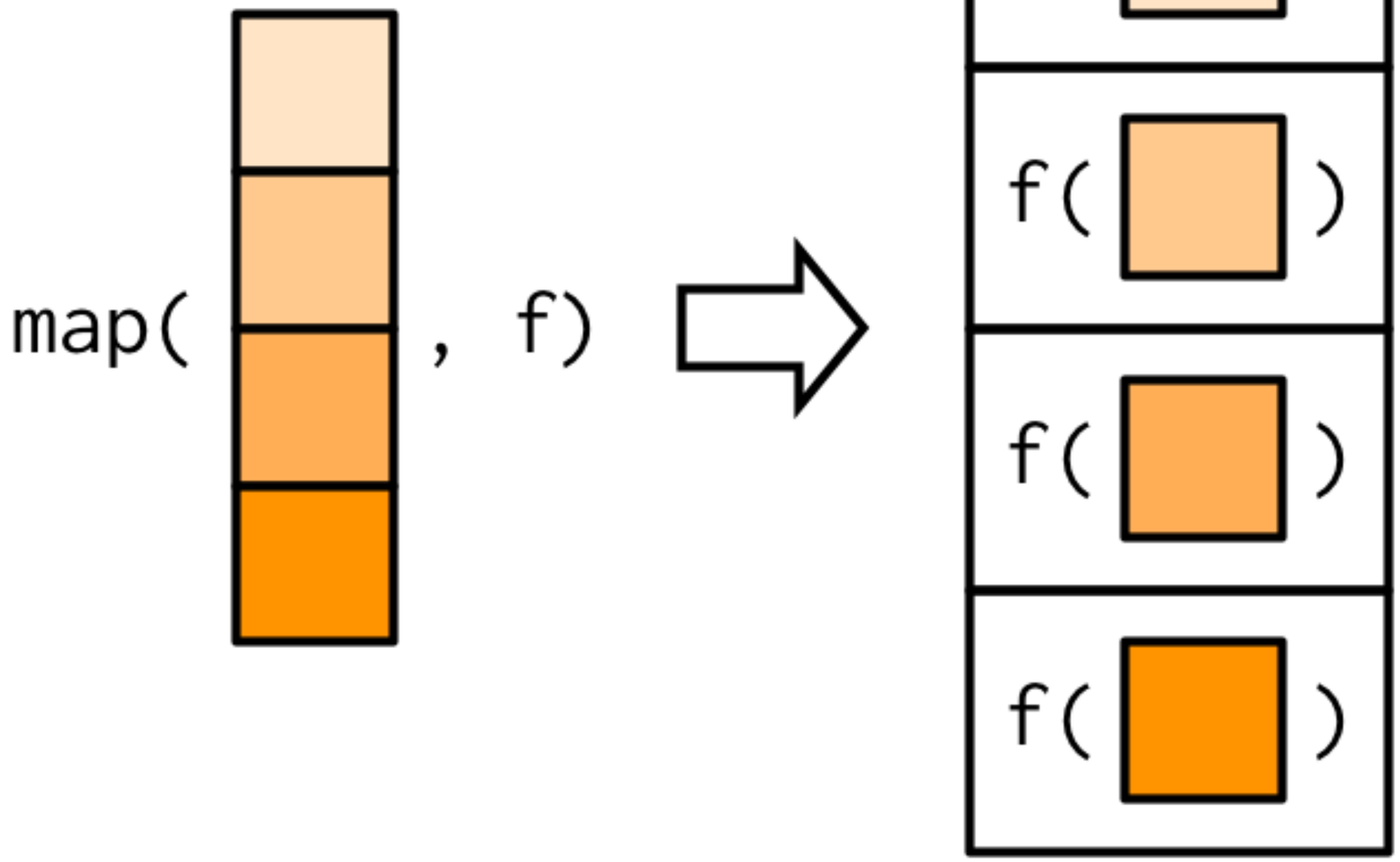

gapminder %>%
filter(year == 1900) %>%
gap_plot()
Life expectancy
50
40
30
20
500
1000
2000
4000
8000
Income (GDP / capita)
Region
africa
americas
asia
europe

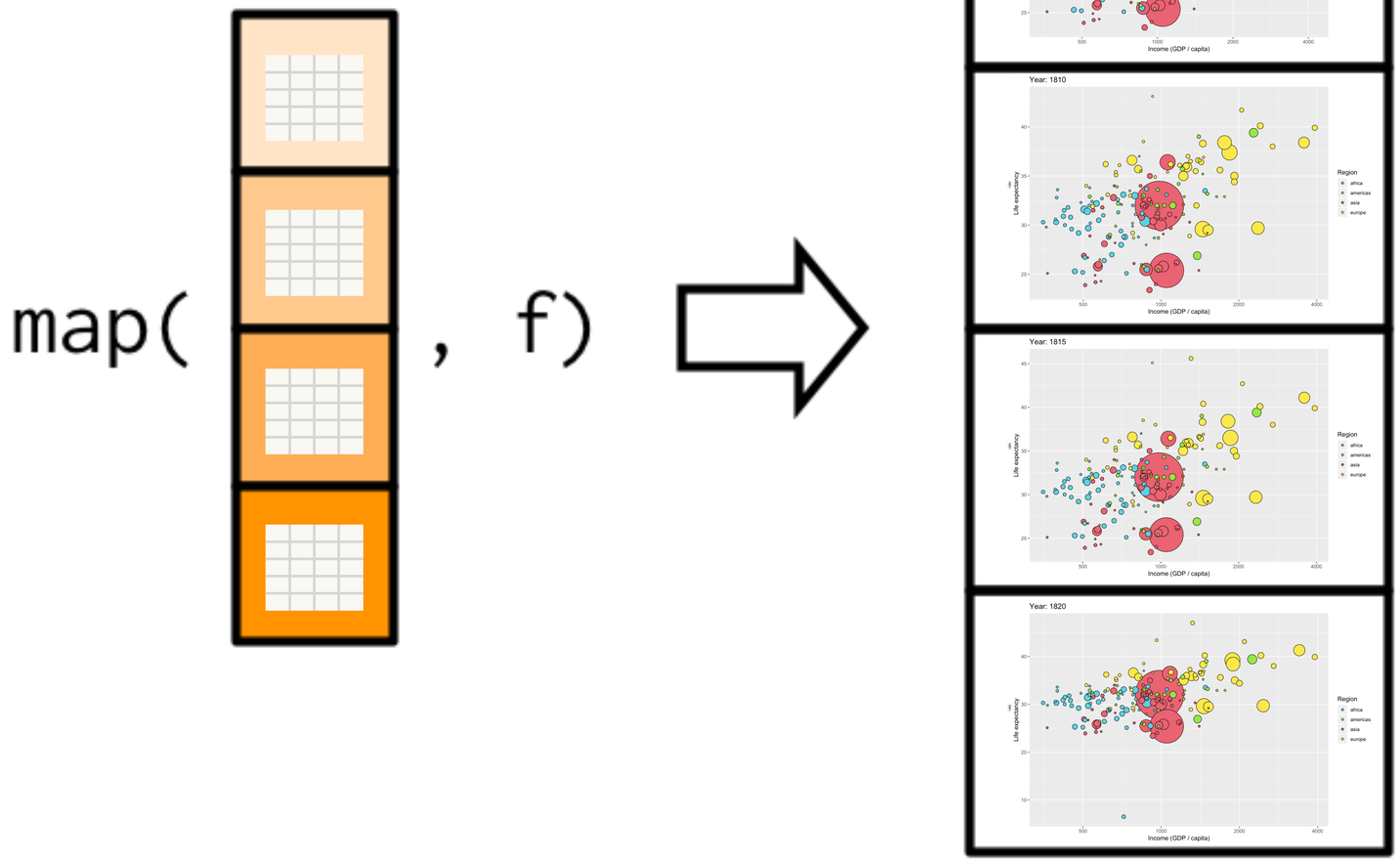

gapminder %>%
filter(year == 1905) %>%
gap_plot()
Life expectancy
50
40
30
20
10
500
1000
2000
4000
8000
Income (GDP / capita)
Region
africa
americas
asia
europe

How can we do this with every plot?

```
by_year <- gapminder %>%
  filter(year %% 5 == 0) %>%
  group_split(year)

plots <- map(by_year, ~ gap_plot(.x))
```


map(, f)
f( )
f( )
f( )
f( )



Datasets

Plots

Import

readr
readxl
haven
xml2

Tidy

tibble
tidyr

Transform

dplyr
forcats
hms
lubridate
stringr

Visualise

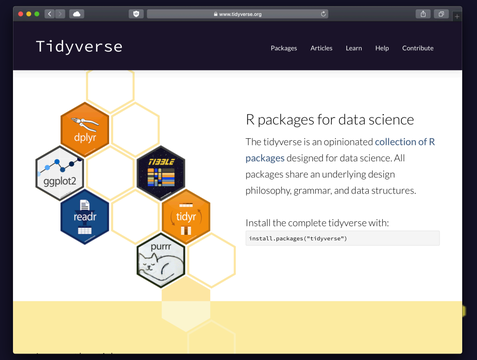
ggplot2

Model

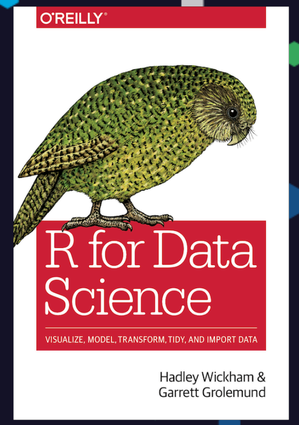
recipes
parsnip

purrr
magrittr

Program



tidyverse.org



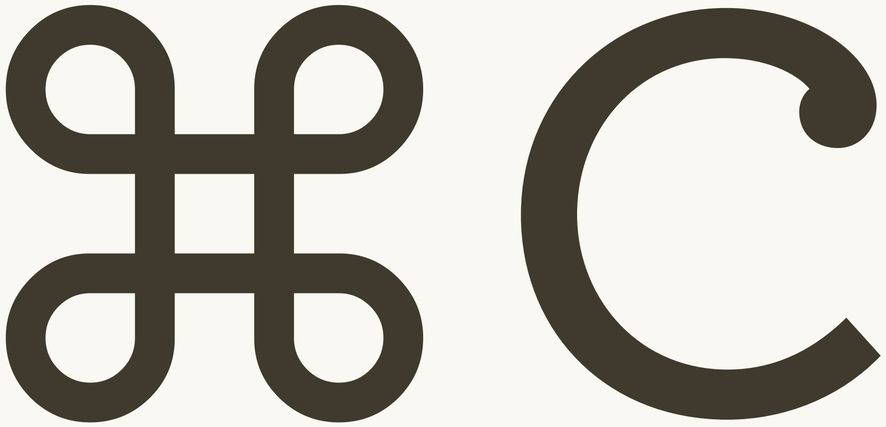
r4ds.had.co.nz

The tidyverse is a language for solving data science challenges with R **code**.

The disadvantages of code are obvious

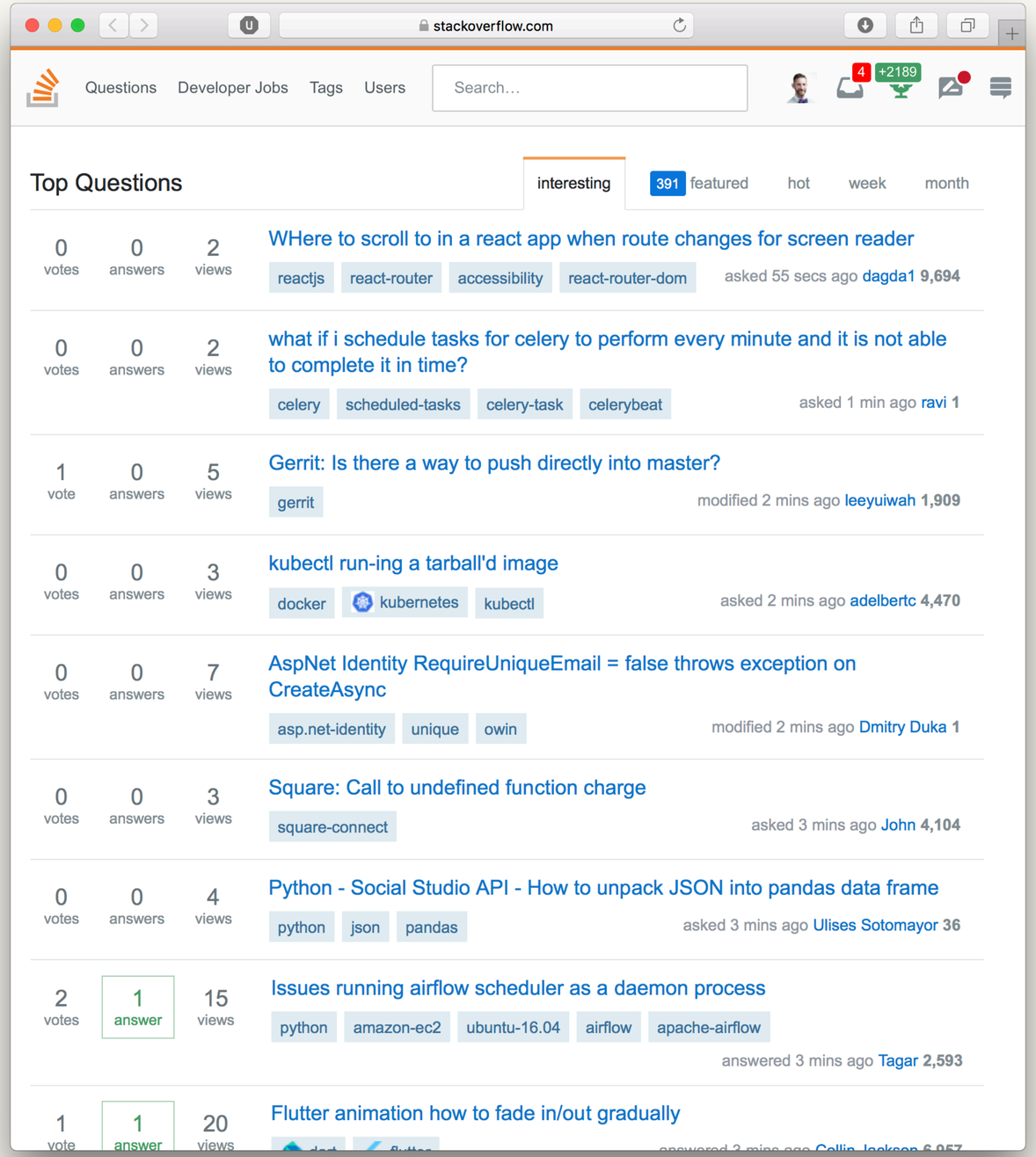
Why code?

1. Code is text
2. Code is read-able
3. Code is reproducible



Copy

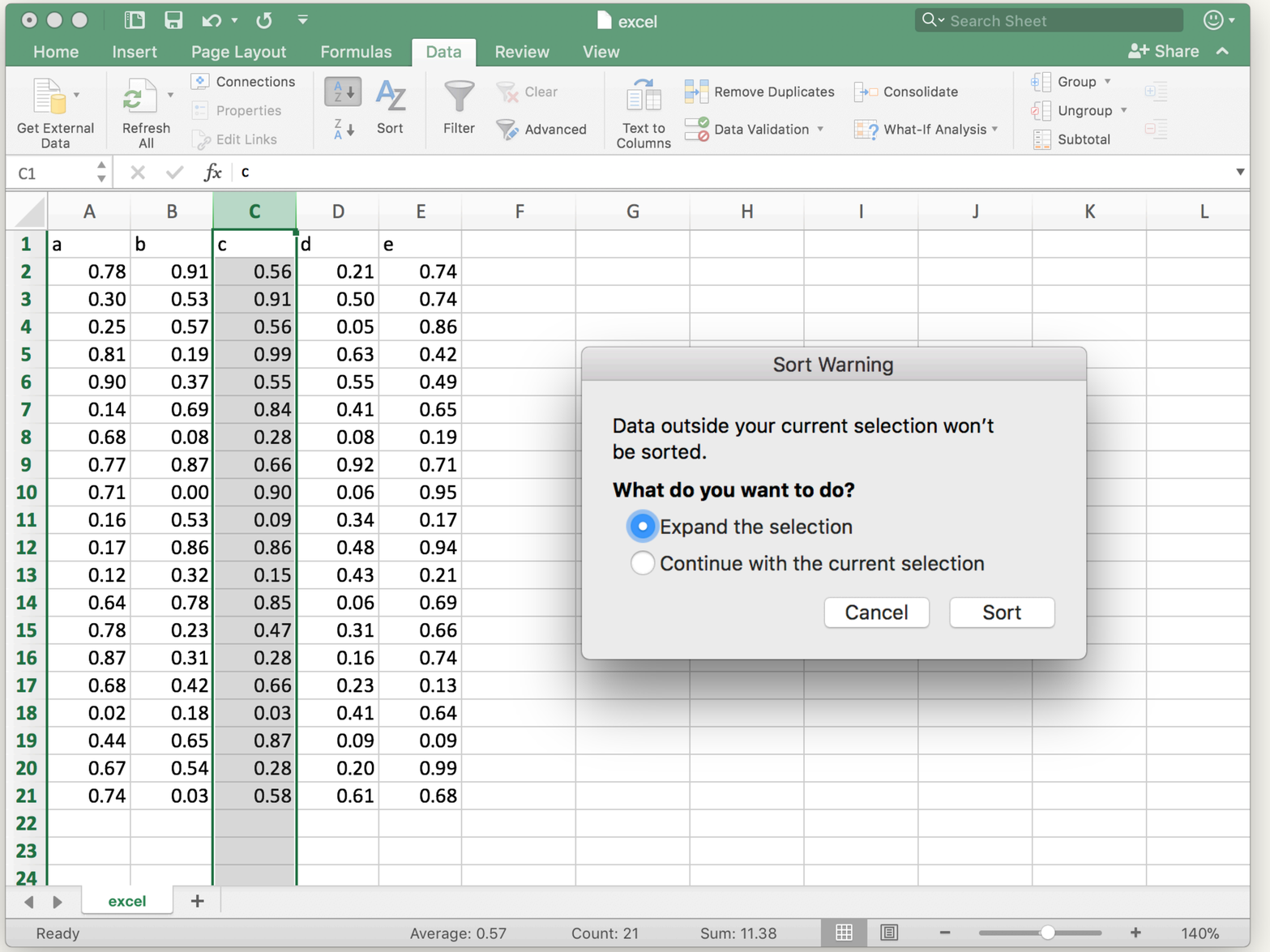
Paste


stackoverflow.com
Questions
Developer Jobs
Tags
Users
Search…
4
+2189
Top Questions
interesting
391 featured
hot
week
month
0 votes
0 answers
2 views
WHere to scroll to in a react app when route changes for screen reader
reactjs
react-router
accessibility
react-router-dom
asked 55 secs ago dagda1 9,694
0 votes
0 answers
2 views
what if i schedule tasks for celery to perform every minute and it is not able to complete it in time?
celery
scheduled-tasks
celery-task
celerybeat
asked 1 min ago ravi 1
1 vote
0 answers
5 views
Gerrit: Is there a way to push directly into master?
gerrit
modified 2 mins ago leeyuiwah 1,909
0 votes
0 answers
3 views
kubectl run-ing a tarball'd image
docker
kubernetes
kubectl
asked 2 mins ago adelbertc 4,470
0 votes
0 answers
7 views
AspNet Identity RequireUniqueEmail = false throws exception on CreateAsync
asp.net-identity
unique
owin
modified 2 mins ago Dmitry Duka 1
0 votes
0 answers
3 views
Square: Call to undefined function charge
square-connect
asked 3 mins ago John 4,104
0 votes
0 answers
4 views
Python - Social Studio API - How to unpack JSON into pandas data frame
python
json
pandas
asked 3 mins ago Ulises Sotomayor 36
2 votes
1 answer
15 views
Issues running airflow scheduler as a daemon process
python
amazon-ec2
ubuntu-16.04
airflow
apache-airflow
answered 3 mins ago Tagar 2,593
1 vote
1 answer
20 views
Flutter animation how to fade in/out gradually

Why code?

1. Code is text
2. Code is read-able
3. Code is reproducible

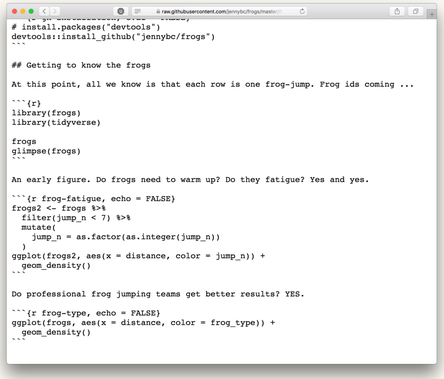
What have you done?



	A	B	C	D	E
1	a	b	c	d	e
2	0.78	0.91	0.56	0.21	0.74
3	0.30	0.53	0.91	0.50	0.74
4	0.25	0.57	0.56	0.05	0.86
5	0.81	0.19	0.99	0.63	0.42
6	0.90	0.37	0.55	0.55	0.49
7	0.14	0.69	0.84	0.41	0.65
8	0.68	0.08	0.28	0.08	0.19
9	0.77	0.87	0.66	0.92	0.71
10	0.71	0.00	0.90	0.06	0.95
11	0.16	0.53	0.09	0.34	0.17
12	0.17	0.86	0.86	0.48	0.94
13	0.12	0.32	0.15	0.43	0.21
14	0.64	0.78	0.85	0.06	0.69
15	0.78	0.23	0.47	0.31	0.66
16	0.87	0.31	0.28	0.16	0.74
17	0.68	0.42	0.66	0.23	0.13
18	0.02	0.18	0.03	0.41	0.64
19	0.44	0.65	0.87	0.09	0.09
20	0.67	0.54	0.28	0.20	0.99
21	0.74	0.03	0.58	0.61	0.68
22					
23					
24					

Why code?

1. Code is text
2. Code is read-able
3. Code is reproducible



```
# install.packages("devtools")
devtools::install_github("jennybc/frogs")
```

Getting to know the frogs

At this point, all we know is that each row is one frog-jump. Frog ids coming ...

```{r}
library(frogs)
library(tidyverse)

frogs
glimpse(frogs)
```

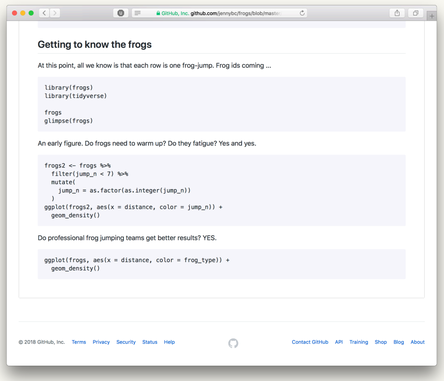
An early figure. Do frogs need to warm up? Do they fatigue? Yes and yes.

```{r frog-fatigue, echo = FALSE}
frogs2 <- frogs %>%
  filter(jump_n < 7) %>%
  mutate(
    jump_n = as.factor(as.integer(jump_n))
  )
ggplot(frogs2, aes(x = distance, color = jump_n)) +
  geom_density()
```

Do professional frog jumping teams get better results? YES.

```{r frog-type, echo = FALSE}
ggplot(frogs, aes(x = distance, color = frog_type)) +
  geom_density()
```

.Rmd — Prose and code



Getting to know the frogs

At this point, all we know is that each row is one frog-jump. Frog ids coming ...

```
library(frogs)
library(tidyverse)

frogs
glimpse(frogs)
```

An early figure. Do frogs need to warm up? Do they fatigue? Yes and yes.

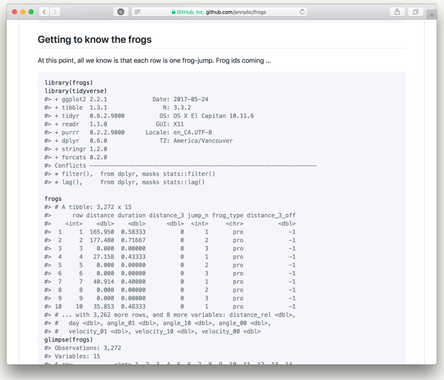
```
frogs2 <- frogs %>%
  filter(jump_n < 7) %>%
  mutate(
    jump_n = as.factor(as.integer(jump_n))
  )
ggplot(frogs2, aes(x = distance, color = jump_n)) +
  geom_density()
```

Do professional frog jumping teams get better results? YES.

```
ggplot(frogs, aes(x = distance, color = frog_type)) +
  geom_density()
```

© 2018 GitHub, Inc. Terms Privacy Security Status Help — Contact GitHub API Training Shop Blog About

.md — Prose and results



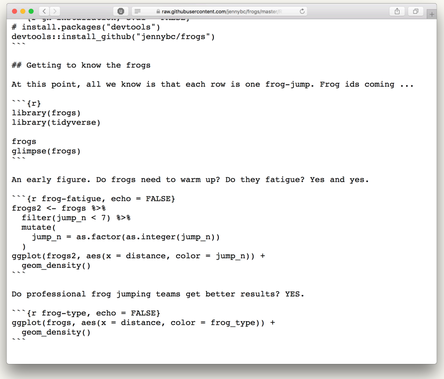
Getting to know the frogs

At this point, all we know is that each row is one frog-jump. Frog ids coming ...

```
library(frogs)
library(tidyverse)
#> ── Attaching packages ─────────────────────────────────────── tidyverse 1.2.1 ──
#> ✔ ggplot2 2.2.1          Date: 2017-05-24
#> ✔ tibble  1.3.1            R: 3.3.2
#> ✔ tidyr   0.6.2.9000       OS: OS X El Capitan 10.11.6
#> ✔ readr   1.1.0          GUI: X11
#> ✔ purrr   0.2.2.9000    Locale: en_CA.UTF-8
#> ✔ dplyr   0.6.0            TZ: America/Vancouver
#> ✔ stringr 1.2.0
#> ✔ forcats 0.2.0
#> ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
#> ✖ dplyr::filter(),  from dplyr, masks stats::filter()
#> ✖ dplyr::lag(),     from dplyr, masks stats::lag()

frogs
#> # A tibble: 3,272 x 15
#>     row distance duration distance_3 jump_n frog_type distance_3_off
#>   <int>    <dbl>    <dbl>      <dbl>  <int>     <chr>          <dbl>
#> 1     1   165.950  0.58333          0      1       pro           -1
#> 2     2   177.480  0.71667          0      2       pro           -1
#> 3     3     0.000  0.00000          0      3       pro           -1
#> 4     4    27.158  0.43333          0      1       pro           -1
#> 5     5     0.000  0.00000          0      2       pro           -1
#> 6     6     0.000  0.00000          0      3       pro           -1
#> 7     7    40.914  0.40000          0      1       pro           -1
#> 8     8     0.000  0.00000          0      2       pro           -1
#> 9     9     0.000  0.00000          0      3       pro           -1
#> 10   10    35.853  0.40333          0      1       pro           -1
#> # ... with 3,262 more rows, and 8 more variables: distance_rel <dbl>,
#> #   day <dbl>, angle_01 <dbl>, angle_10 <dbl>, angle_00 <dbl>,
#> #   velocity_01 <dbl>, velocity_10 <dbl>, velocity_00 <dbl>
glimpse(frogs)
#> Observations: 3,272
#> Variables: 15
#> $ row          <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,...
```

.html — Human shareable



```
# install.packages("devtools")
devtools::install_github("jennybc/frogs")
```

Getting to know the frogs

At this point, all we know is that each row is one frog-jump. Frog ids coming ...

```{r}
library(frogs)
library(tidyverse)

frogs
glimpse(frogs)
```

An early figure. Do frogs need to warm up? Do they fatigue? Yes and yes.

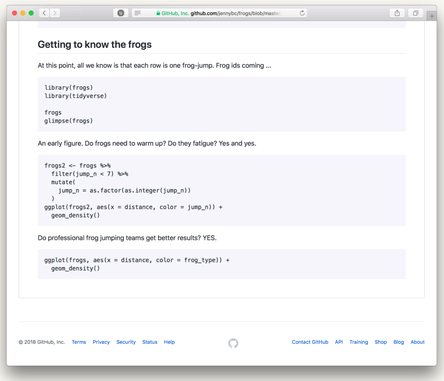
```{r frog-fatigue, echo = FALSE}
frogs2 <- frogs %>%
  filter(jump_n < 7) %>%
  mutate(
    jump_n = as.factor(as.integer(jump_n))
  )
ggplot(frogs2, aes(x = distance, color = jump_n)) +
  geom_density()
```

Do professional frog jumping teams get better results? YES.

```{r frog-type, echo = FALSE}
ggplot(frogs, aes(x = distance, color = frog_type)) +
  geom_density()
```

.Rmd

Prose and code



Getting to know the frogs

At this point, all we know is that each row is one frog-jump. Frog ids coming ...

```
library(frogs)
library(tidyverse)

frogs
glimpse(frogs)
```

An early figure. Do frogs need to warm up? Do they fatigue? Yes and yes.

```
frogs2 <- frogs %>%
  filter(jump_n < 7) %>%
  mutate(
    jump_n = as.factor(as.integer(jump_n))
  )
ggplot(frogs2, aes(x = distance, color = jump_n)) +
  geom_density()
```

Do professional frog jumping teams get better results? YES.

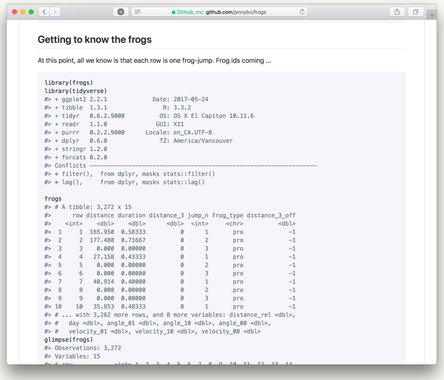
```
ggplot(frogs, aes(x = distance, color = frog_type)) +
  geom_density()
```

© 2018 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About

.md

Prose and results



Getting to know the frogs

At this point, all we know is that each row is one frog-jump. Frog ids coming ...

```
library(frogs)
library(tidyverse)
#> ✔ ggplot2 2.2.1           Date: 2017-05-24
#> ✔ tibble  1.3.1              R: 3.3.2
#> ✔ tidyr   0.6.2.9000       OS: OS X El Capitan 10.11.6
#> ✔ readr   1.1.0           GUI: X11
#> ✔ purrr   0.2.2.9000   Locale: en_CA.UTF-8
#> ✔ dplyr   0.6.0            TZ: America/Vancouver
#> ✔ stringr 1.2.0
#> ✔ forcats 0.2.0
#> Conflicts ----------------------------------------------------------------------
#> ✖ filter(),  from dplyr, masks stats::filter()
#> ✖ lag(),     from dplyr, masks stats::lag()

frogs
#> # A tibble: 3,272 x 15
#>      row distance duration distance_3 jump_n frog_type distance_3_off
#>    <int>    <dbl>    <dbl>      <dbl>  <int>     <chr>          <dbl>
#>  1     1  165.950  0.58333          0      1       pro             -1
#>  2     2  177.480  0.71667          0      2       pro             -1
#>  3     3    0.000  0.00000          0      3       pro             -1
#>  4     4   27.158  0.43333          0      1       pro             -1
#>  5     5    0.000  0.00000          0      2       pro             -1
#>  6     6    0.000  0.00000          0      3       pro             -1
#>  7     7   40.914  0.40000          0      1       pro             -1
#>  8     8    0.000  0.00000          0      2       pro             -1
#>  9     9    0.000  0.00000          0      3       pro             -1
#> 10    10   35.853  0.48333          0      1       pro             -1
#> # ... with 3,262 more rows, and 8 more variables: distance_rel <dbl>,
#> #   day <dbl>, angle_01 <dbl>, angle_10 <dbl>, angle_00 <dbl>,
#> #   velocity_01 <dbl>, velocity_10 <dbl>, velocity_00 <dbl>
glimpse(frogs)
#> Observations: 3,272
#> Variables: 15
```

.html

.doc

.ppt

.tex

.pdf

...

Import

readr
readxl
haven
xml2

Tidy

tibble
tidyr

Transform

dplyr
forcats
hms

lubridate
stringr

Visualise

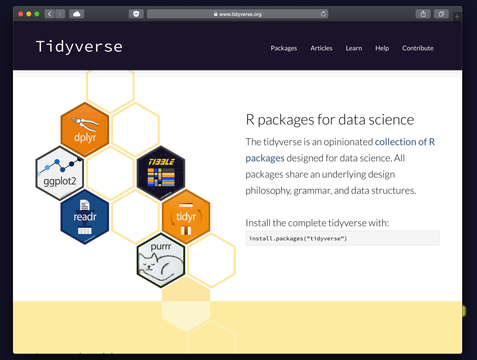
ggplot2

Model

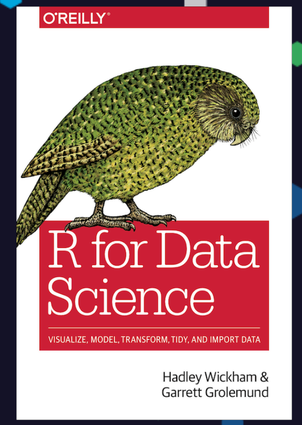
recipes
parsnip

purrr
magrittr

Program



tidyverse.org



r4ds.had.co.nz