Coding data science

March 2019

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

What is data analysis?

Data analysis is the process by which data becomes understanding, knowledge and insight

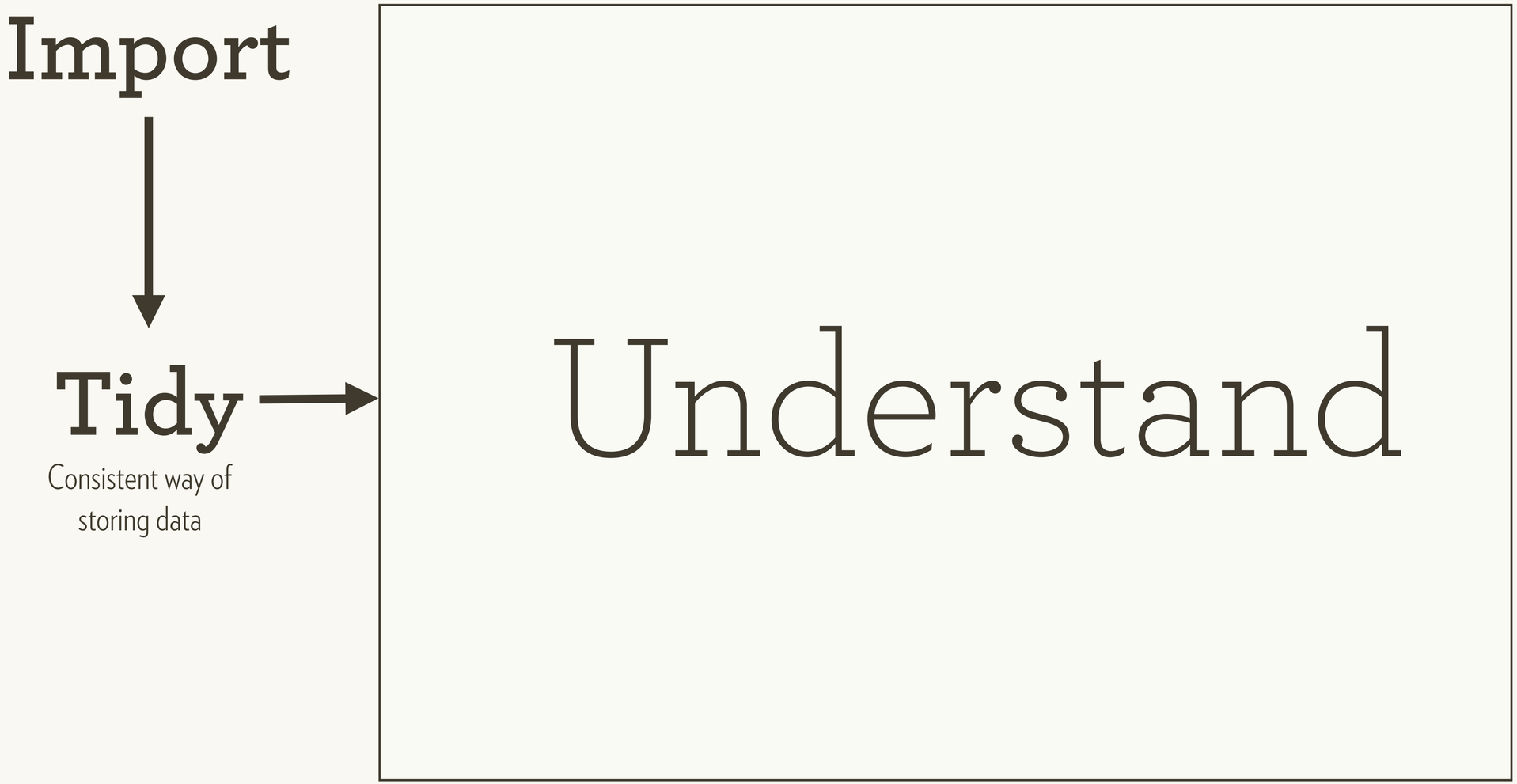
Data analysis is the process by which data becomes understanding, knowledge and insight

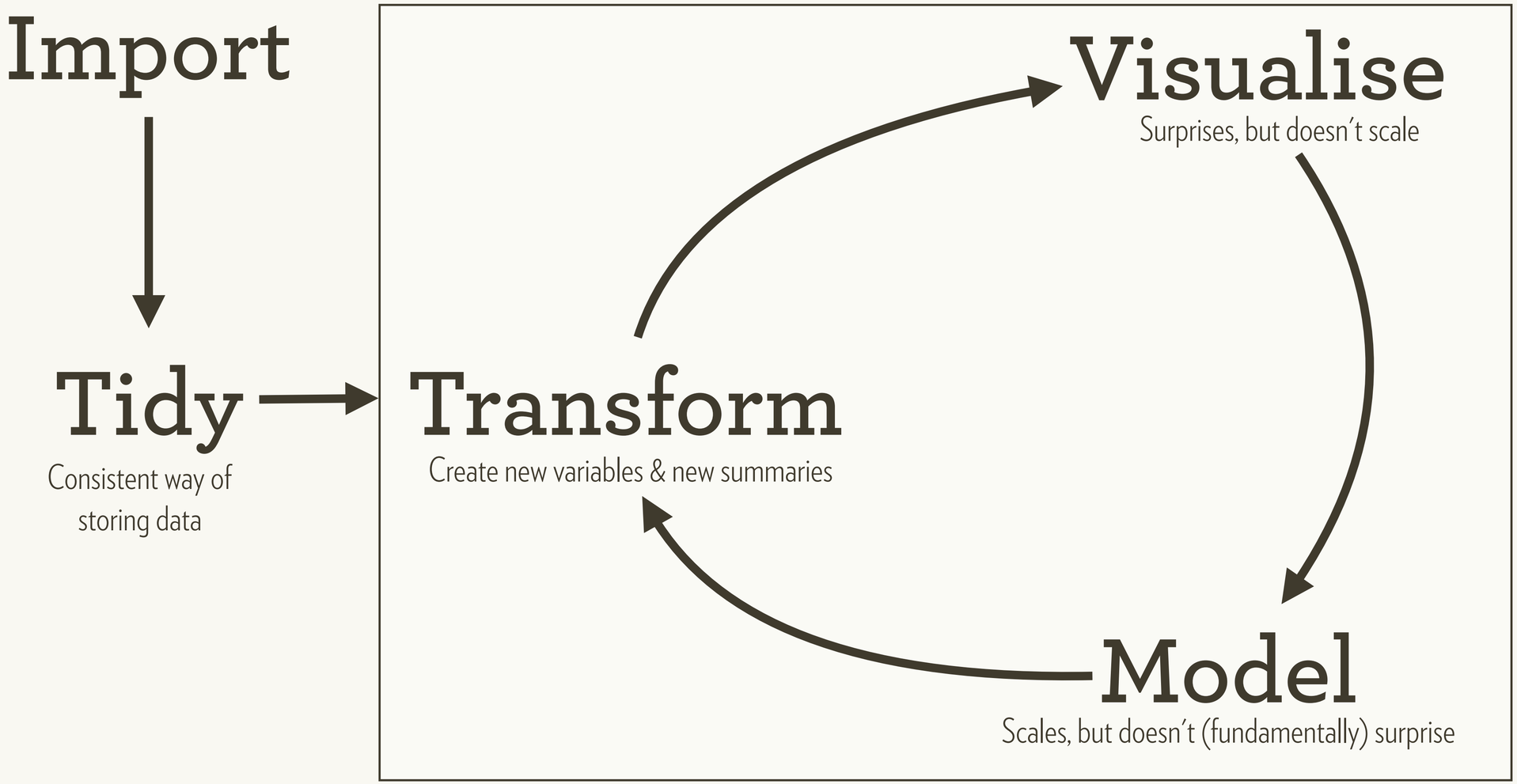
Data analysis is the process by which data becomes understanding, knowledge and insight

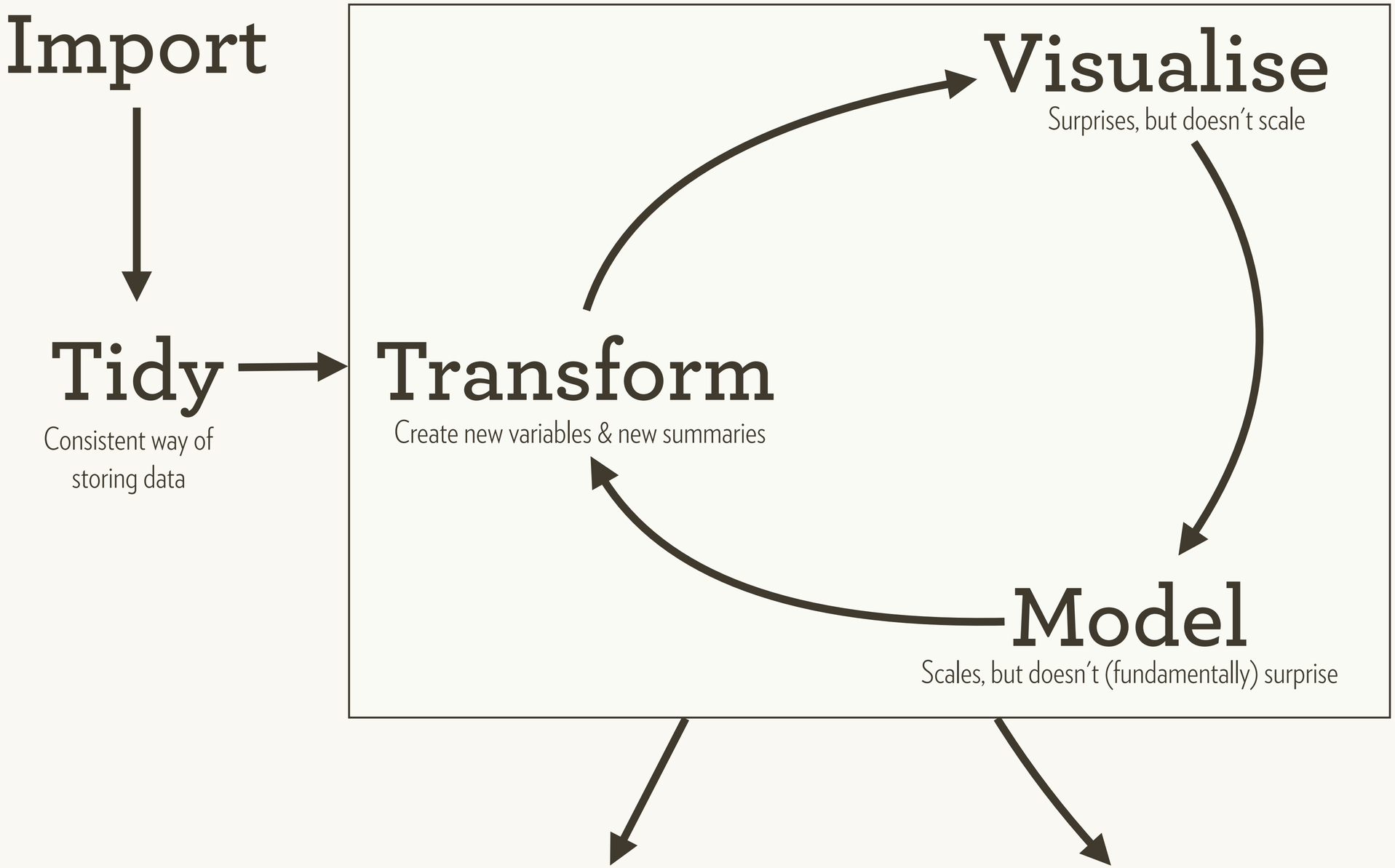
Import

Tidy

Consistent way of storing data


Import
Tidy
Consistent way of storing data
Understand


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

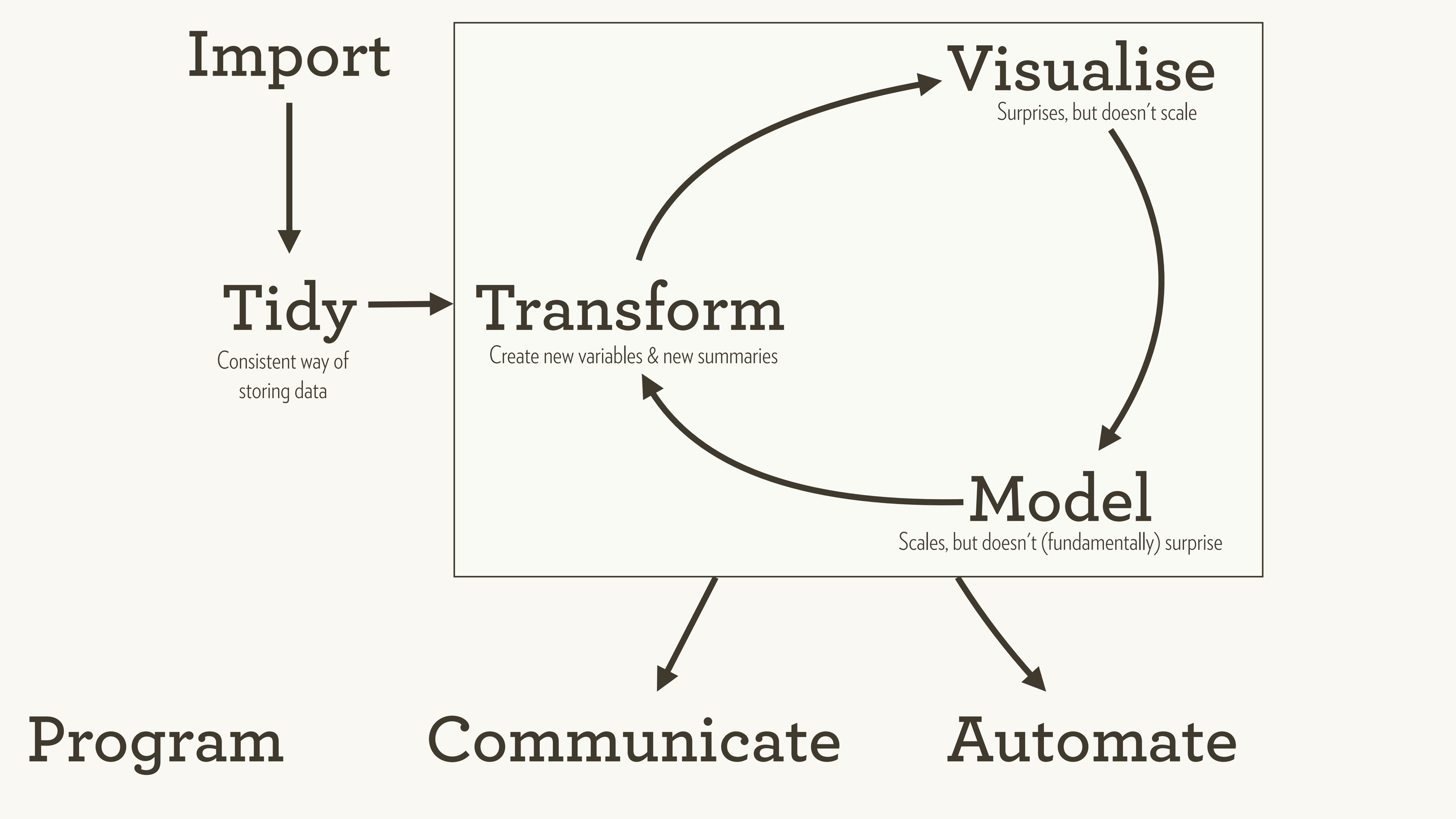

Import
Tidy
Consistent way of storing data
Transform
Create new variables & new summaries
Visualise
Surprises, but doesn't scale
Model
Scales, but doesn't (fundamentally) surprise
Communicate
Automate

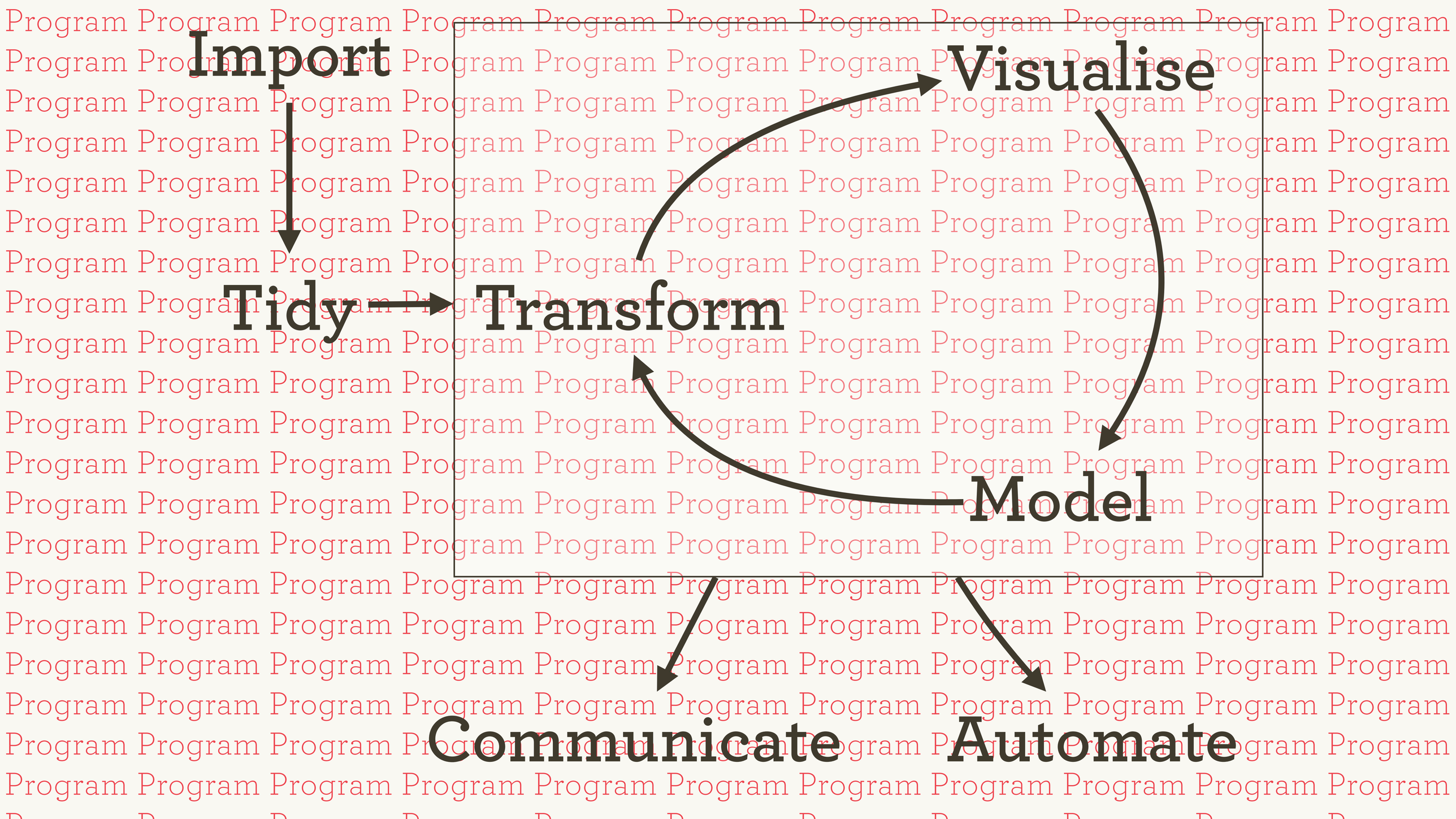
What is data science?

Data science =

data analysis +

programming


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
Automate


Import
Tidy
Transform
Visualise
Model
Communicate
Automate


tidyverse
www.rstudio.com

Import

readr
readxl
haven
xml2

Tidy

tibble
tidyr

Transform

dplyr
forcats
hms
lubridate
stringr

Visualise

ggplot2

Model

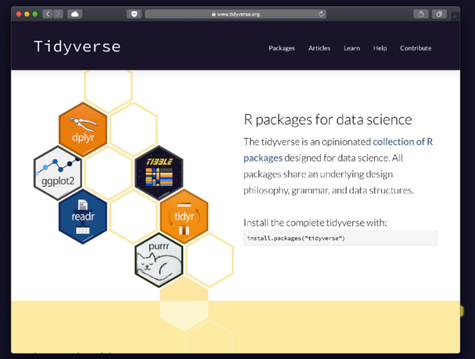
recipes
parsnip

purrr
magrittr

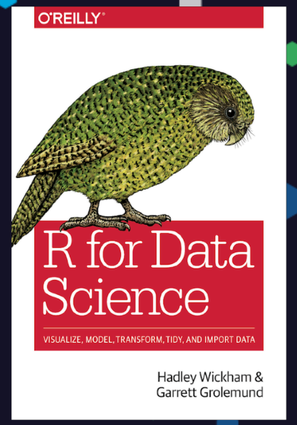
Program

shiny
rmarkdown

Communicate



tidyverse.org



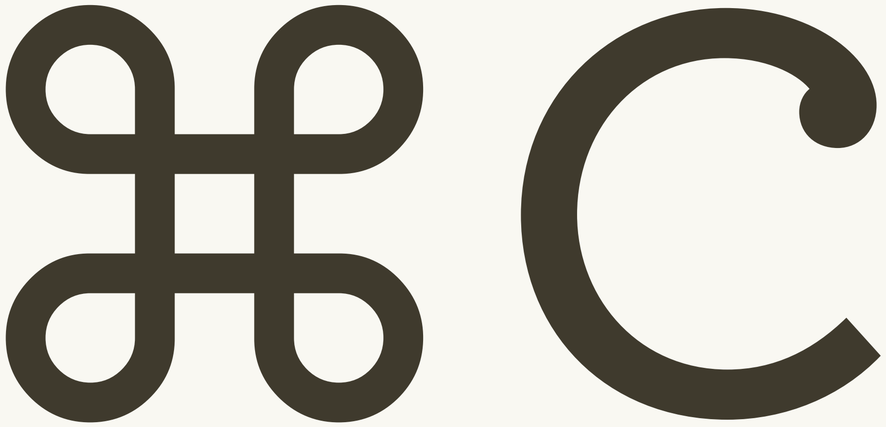
r4ds.had.co.nz

Why code?

The disadvantages of code are obvious

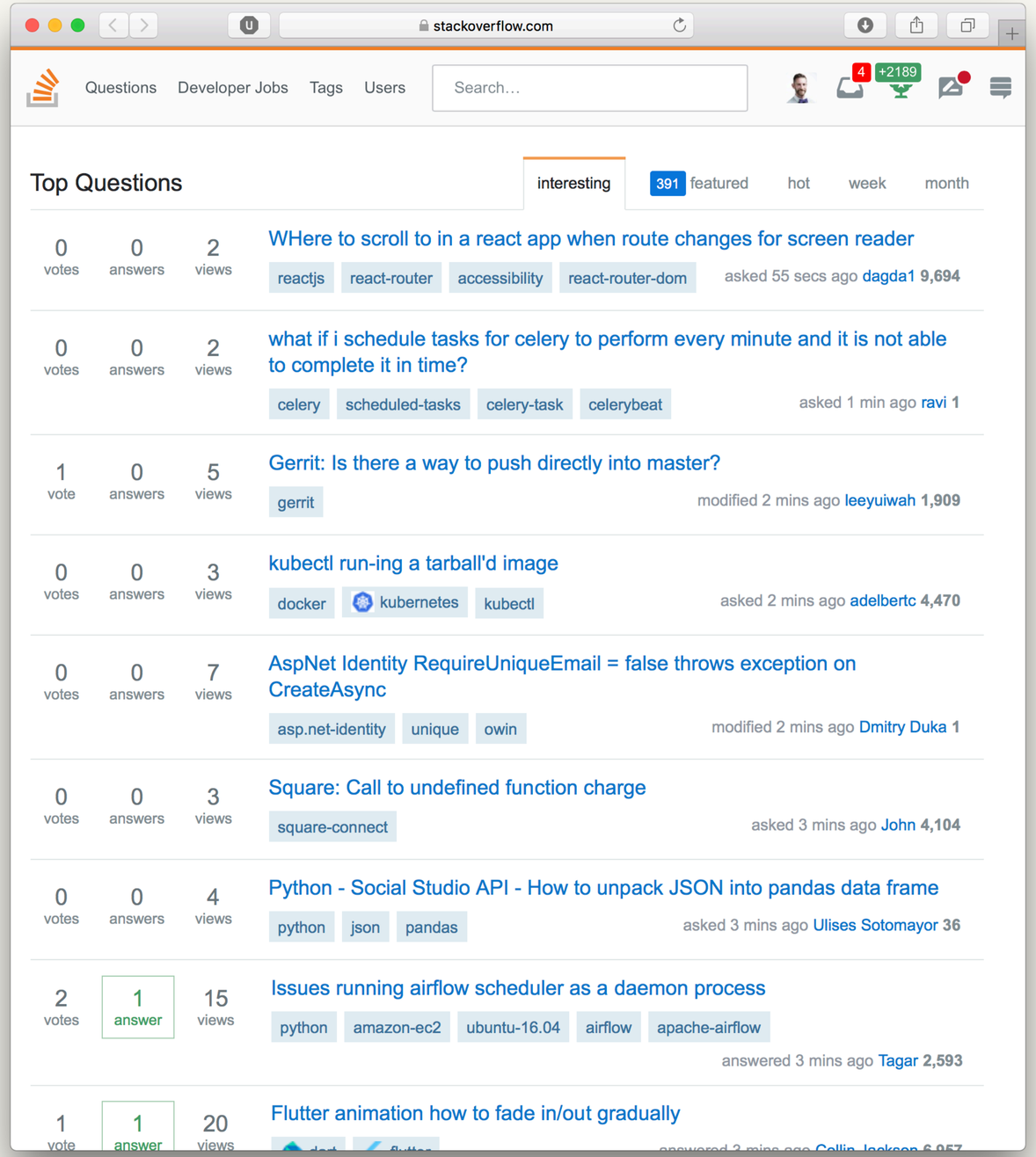
Why code?

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



Copy

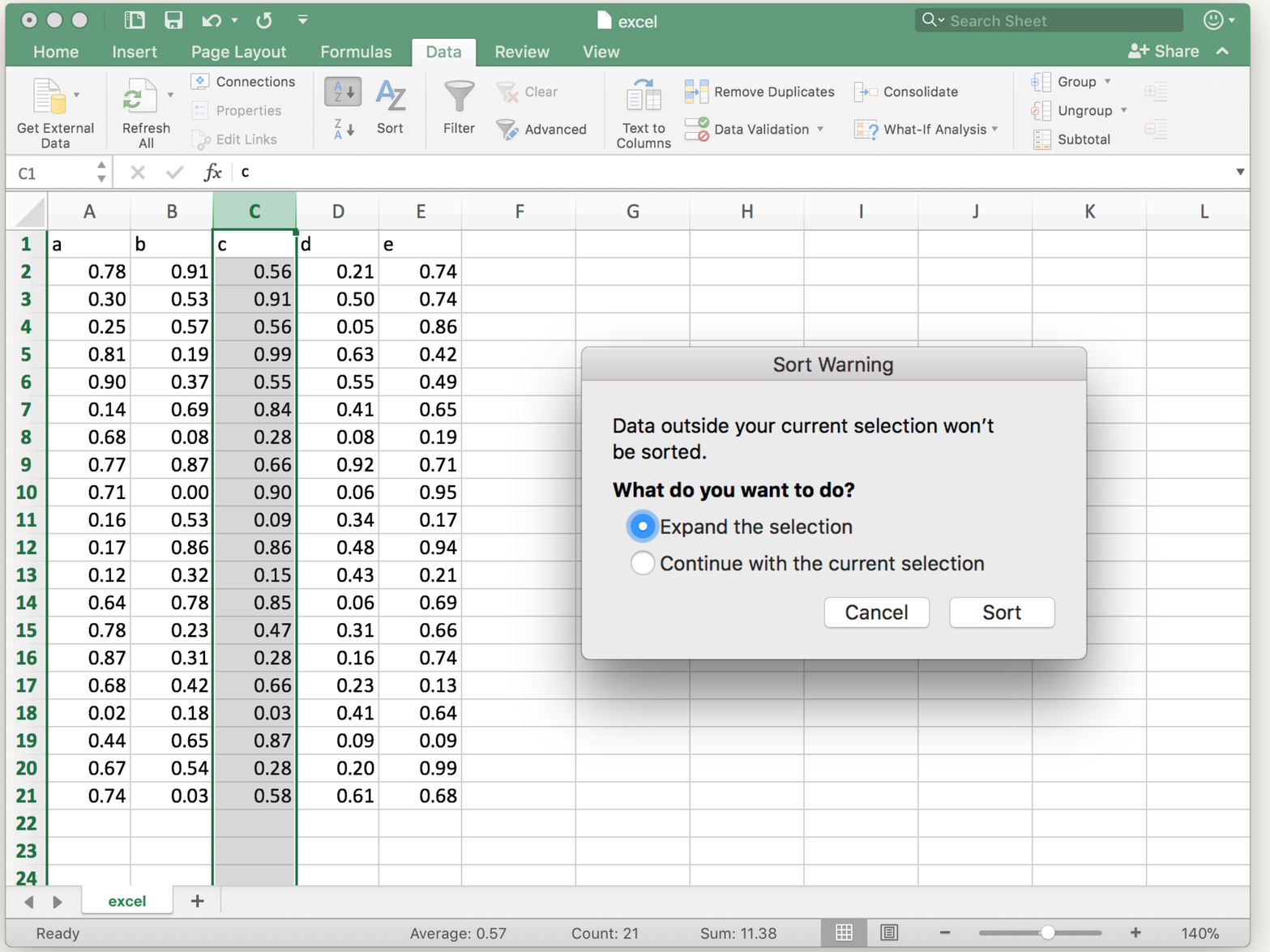
Paste


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

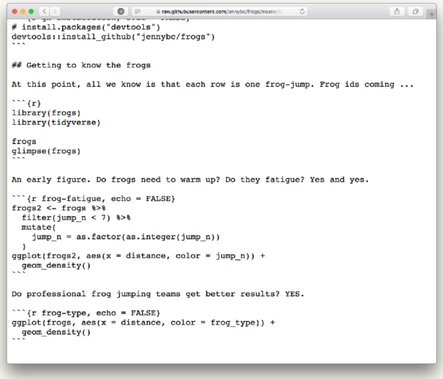
What have you done?



	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)
```

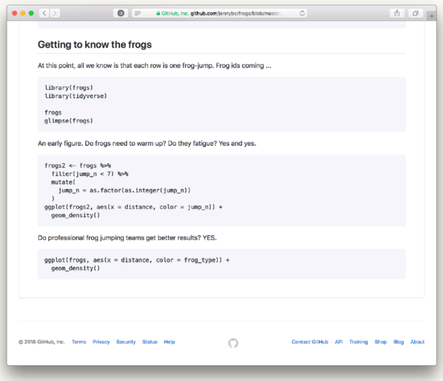
An early figure. Do frogs need to warm up? Do they fatigue? Yes and yes.

```{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)
```

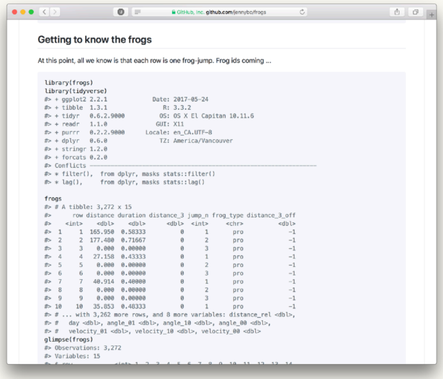
An early figure. Do frogs need to warm up? Do they fatigue? Yes and yes.

```
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()
```

.md — Prose and results



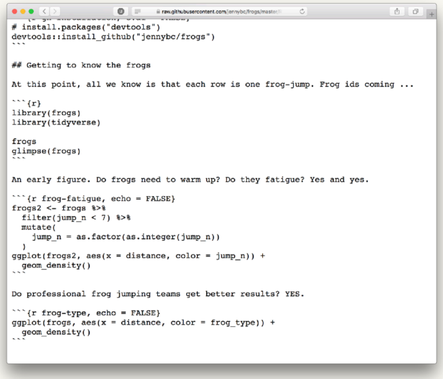
Getting to know the frogs

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   35.053  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
```

.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.

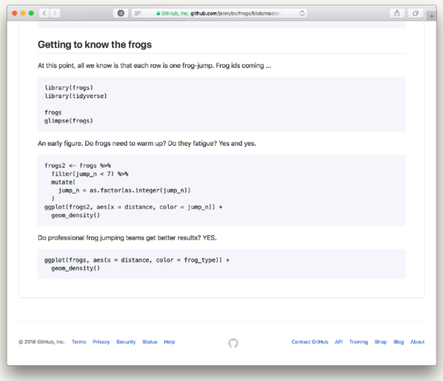
```{r frog-fatigue, echo = FALSE}
frogs2 <- frogs %>%
  filter(jump_n < 7) %>%
  mutate(
    jump_n = as.factor(as.integer(jump_n))
  )
ggplot(frogs2, aes(x = distance, color = jump_n)) +
  geom_density()
```

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.

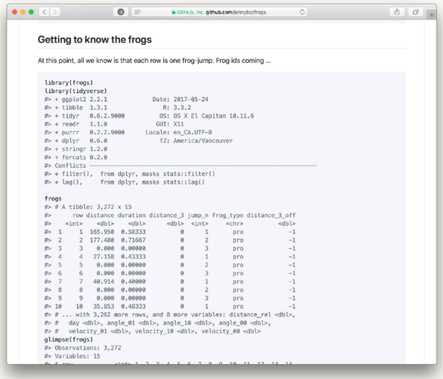
```
frogs2 <- frogs %>%
  filter(jump_n < 7) %>%
  mutate(
    jump_n = as.factor(as.integer(jump_n))
  )
ggplot(frogs2, aes(x = distance, color = jump_n)) +
  geom_density()
```

Do professional frog jumping teams get better results? YES.

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

.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   35.053  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
```

.html

.doc

.ppt

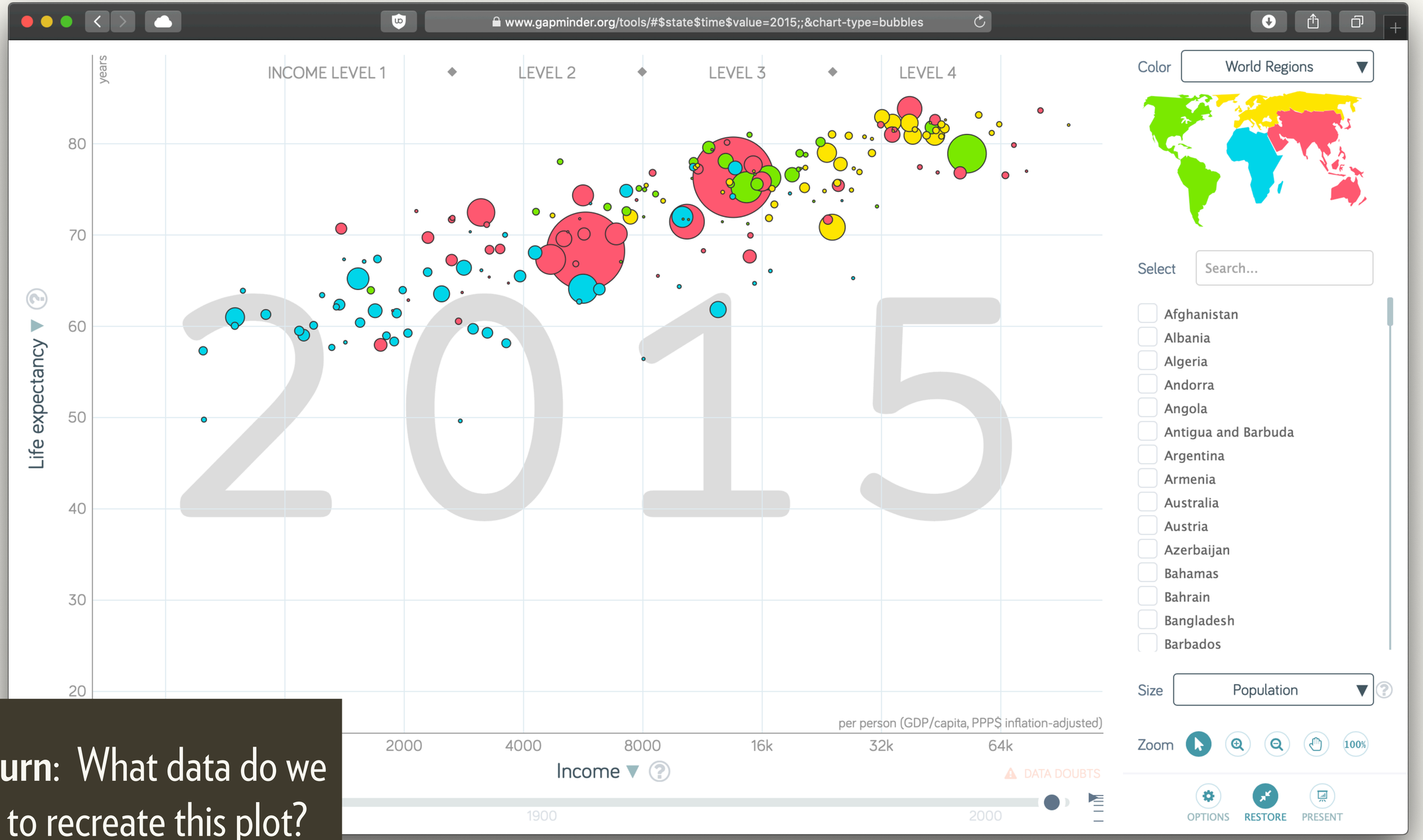
.tex

.pdf

...

What about non-programmers?

You don’t need to be a programmer to code!



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

Underlying data

```
# 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
```

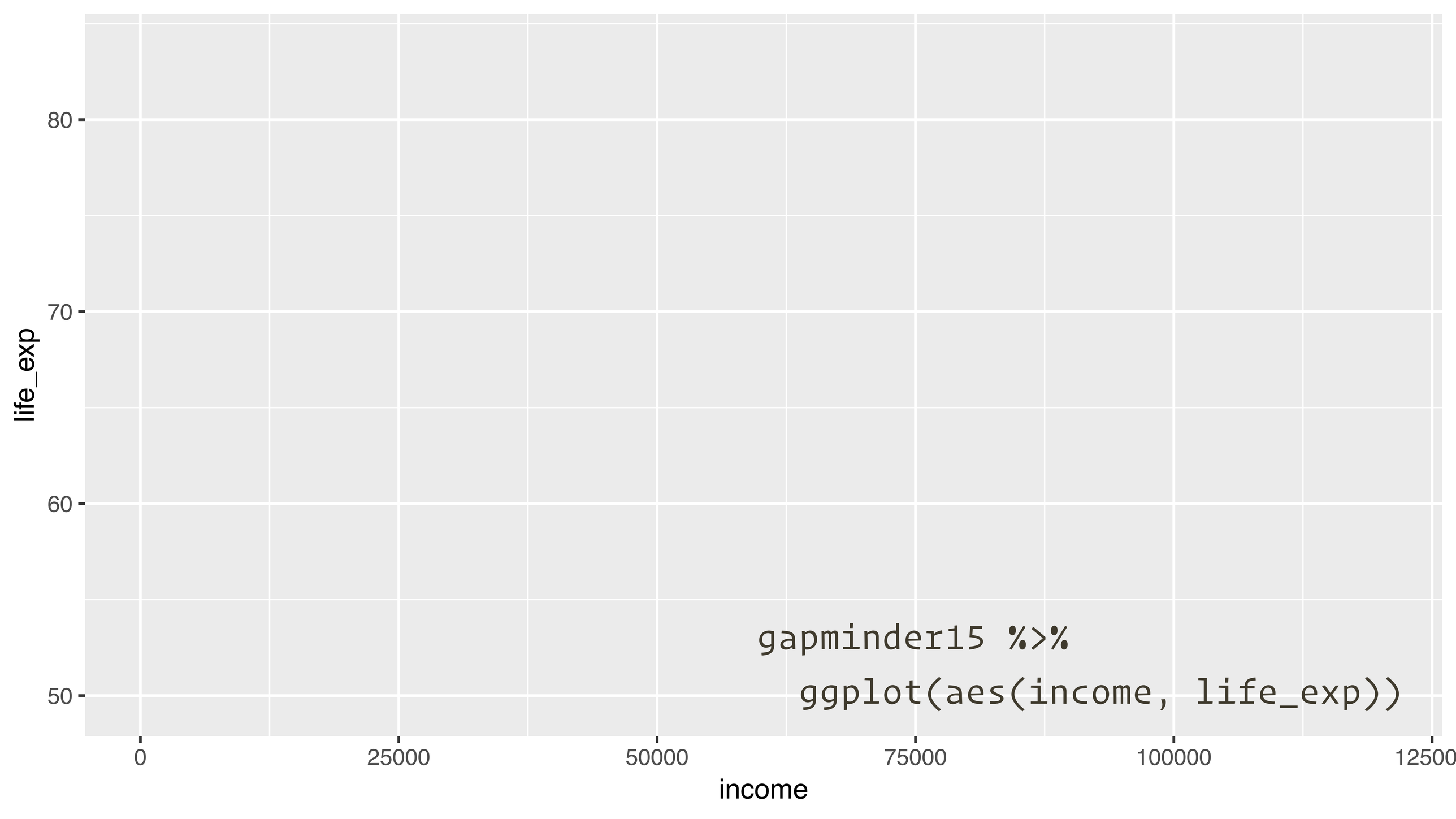
Phonics are important!

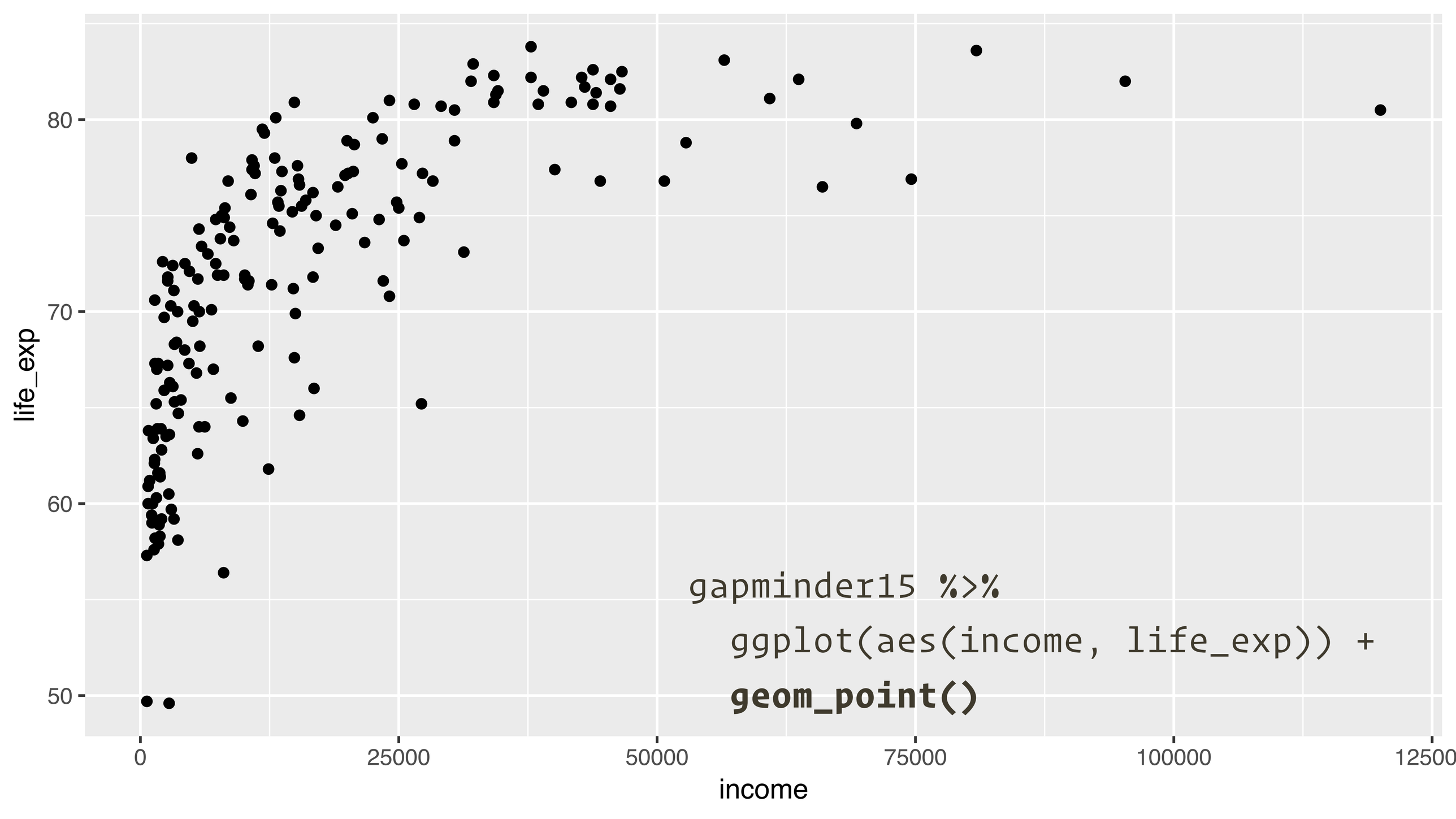
```
gapminder %>%
  filter(year == 2015) ->
  gapminder15
```

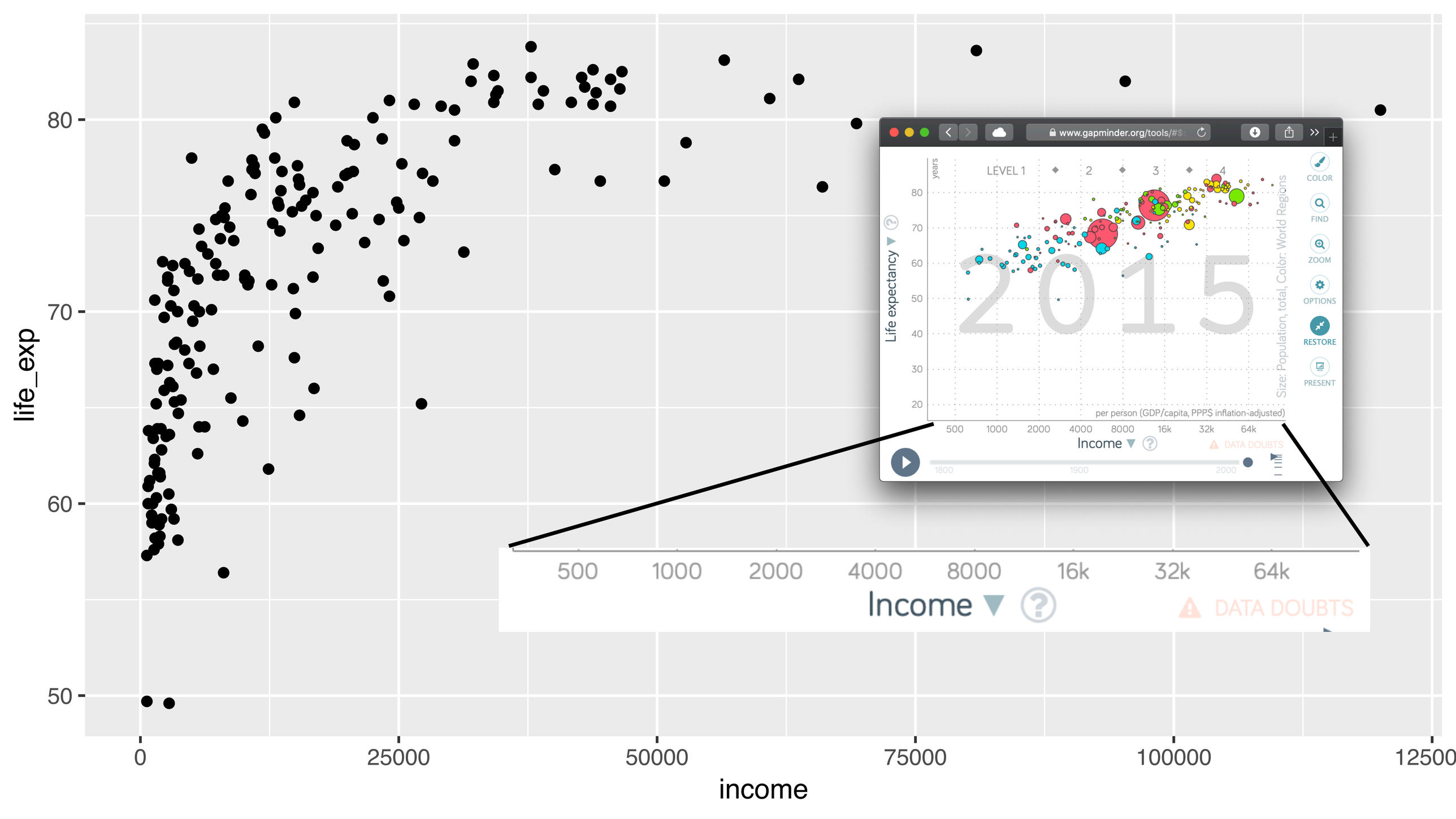
Take the gapminder data, then

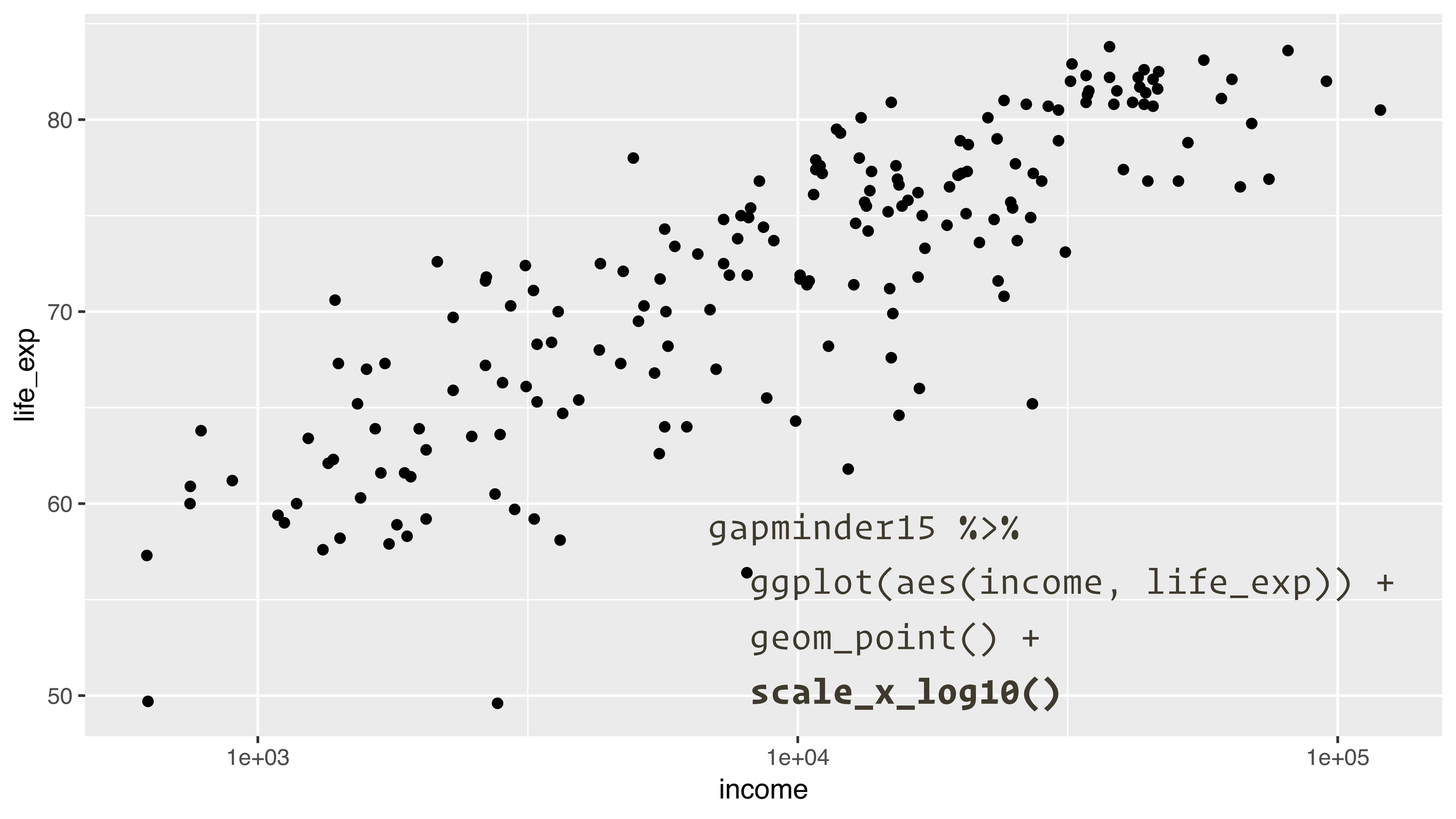
filter rows where year equals 2015, creating

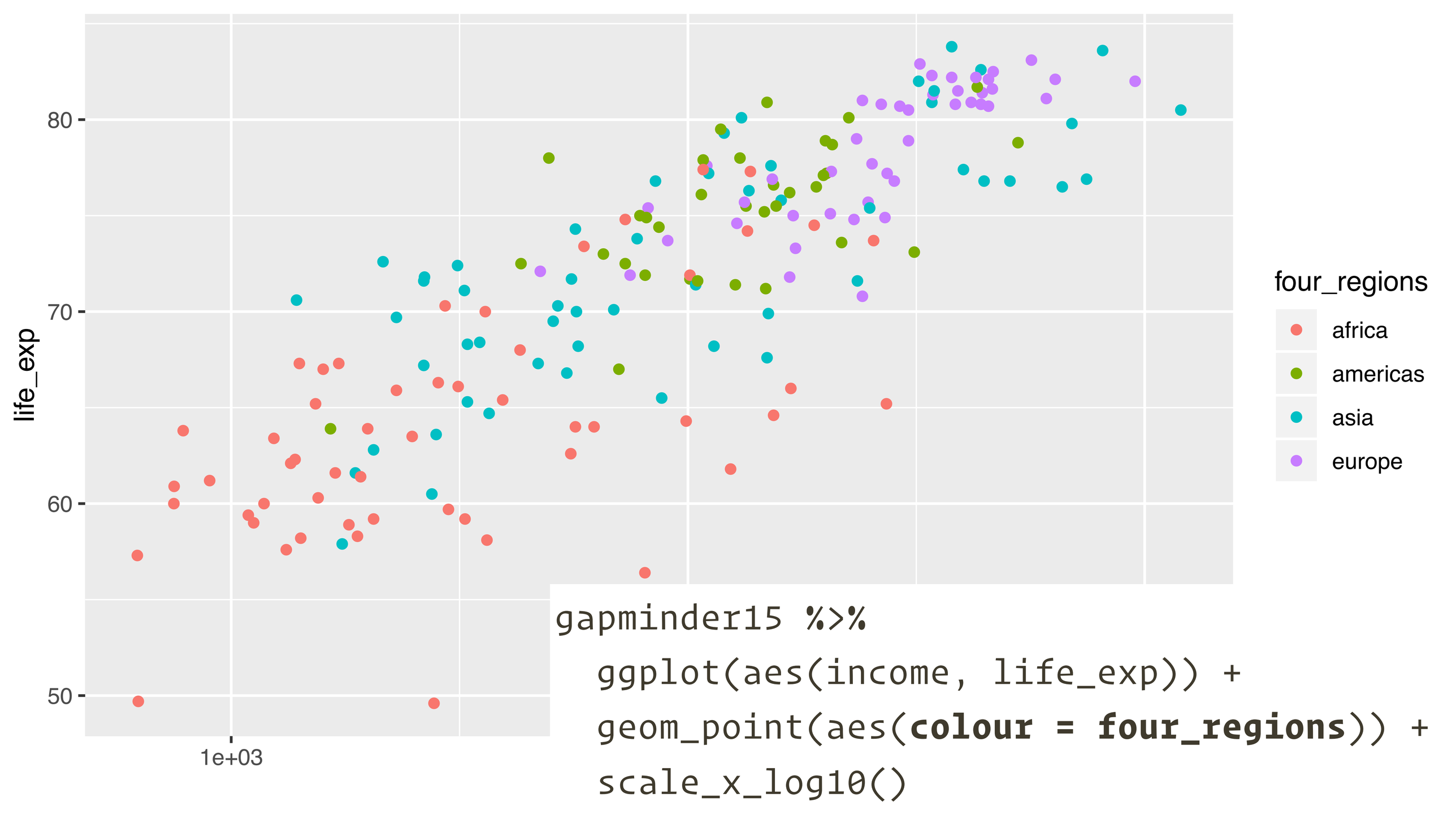
gapminder15 variable


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

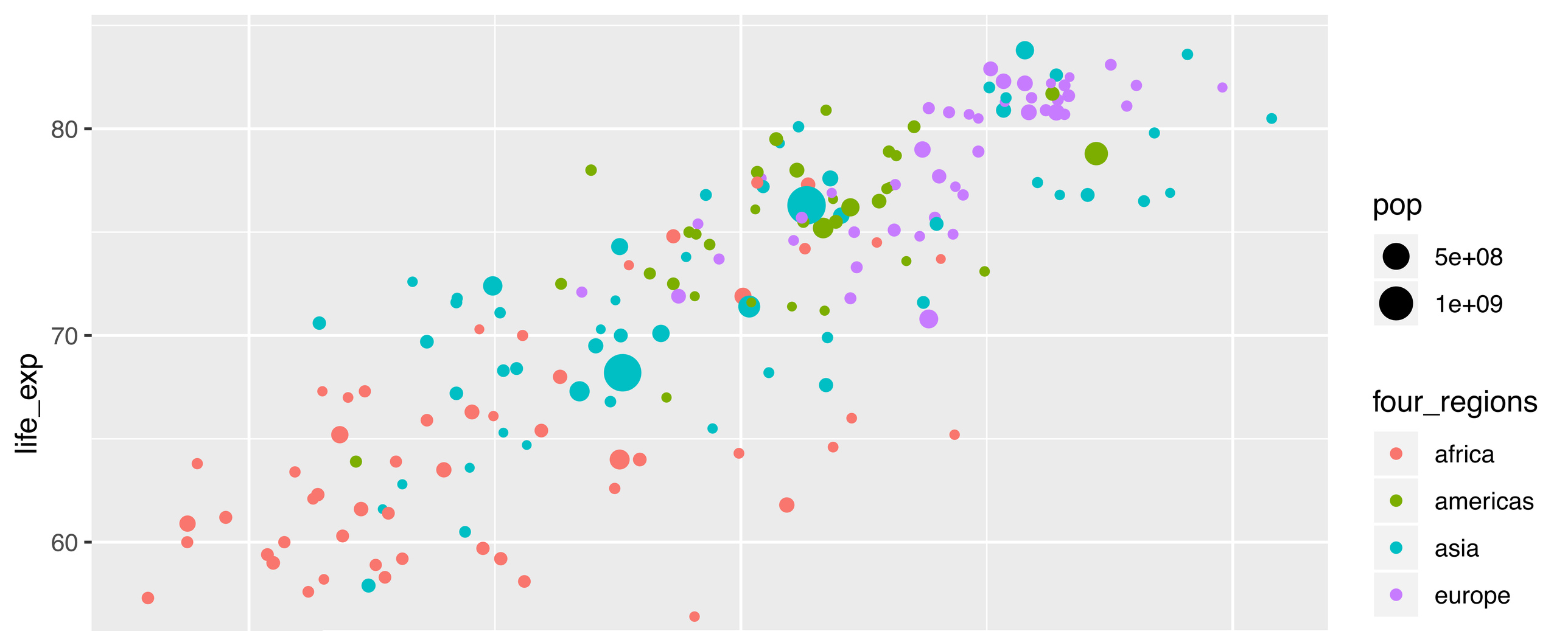

life_exp
80
70
60
50
0
25000
50000
75000
100000
12500
income
gapminder15 %>%
ggplot(aes(income, life_exp)) +
geom_point()


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

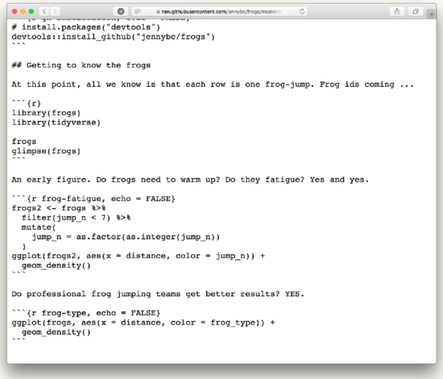

80
70
60
50
life_exp
1e+03
1e+04
1e+05
income
gapminder15 %>%
ggplot(aes(income, life_exp)) +
geom_point() +
scale_x_log10()




```
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()
```



```
# 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.

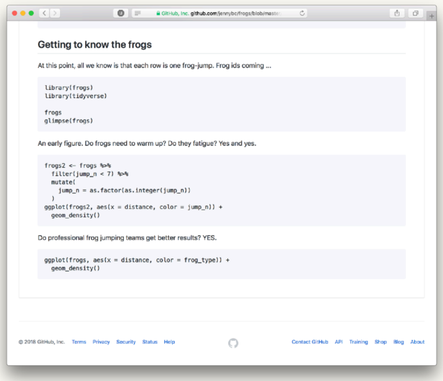
```{r frog-fatigue, echo = FALSE}
frogs2 <- frogs %>%
  filter(jump_n < 7) %>%
  mutate(
    jump_n = as.factor(as.integer(jump_n))
  )
ggplot(frogs2, aes(x = distance, color = jump_n)) +
  geom_density()
```

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.

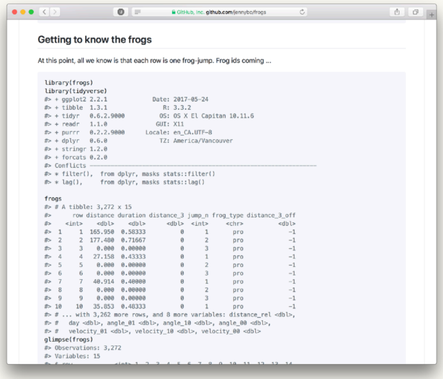
```
frogs2 <- frogs %>%
  filter(jump_n < 7) %>%
  mutate(
    jump_n = as.factor(as.integer(jump_n))
  )
ggplot(frogs2, aes(x = distance, color = jump_n)) +
  geom_density()
```

Do professional frog jumping teams get better results? YES.

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

.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   35.053  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
```

.html

.doc

.ppt

.tex

.pdf

...

But


40
30
20
hwy
2
3
4
5
6
7
displ

And this is painful!

```
df %>%
  rename(
    date = `Date Created`,
    name = Name,
    plays = `Total Plays`,
    loads = `Total Loads`,
    apv = `Average Percent Viewed`
  )
```

RStudio Addins

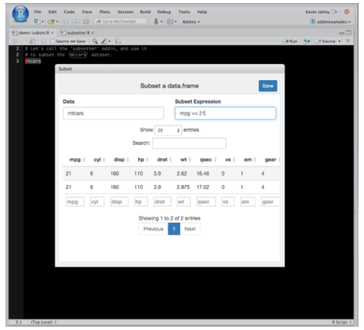
IMPORTANT NOTE: Support for addins is available only within the most recent release of RStudio (v0.99.878 or later). If you want to try out addins please be sure to download this release.

RStudio addins provide a mechanism for executing R functions interactively from within the RStudio IDE—either through keyboard shortcuts, or through the *Addins* menu.

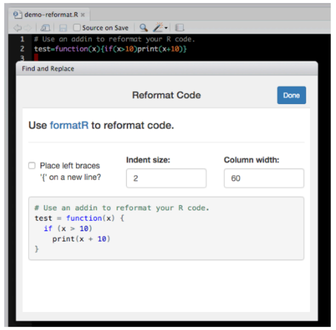
An addin can be as simple as a function that inserts a commonly used snippet of text, and as complex as a Shiny application that accepts input from the user, and later mutates a document open in RStudio. The sky is the limit!

Here are two examples of addins in action (click on the thumbnail to see a brief demonstration):

Subset a dataset



Reformat R code



Using Addins

This guide will walk you through the basics of installing addins, binding keyboard shorcuts to them, and finally developing your own addins.

Installation

RStudio Addins are distributed as R packages. Once you've installed an R package that contains addins, they'll be immediately become available within RStudio.

Let's start by playing around with a couple of the example addins provided by the addinexamples package. Within RStudio, install this package (plus its requisite dependencies) with:

```
devtools::install_github("rstudio/addinexamples", type = "source")
```

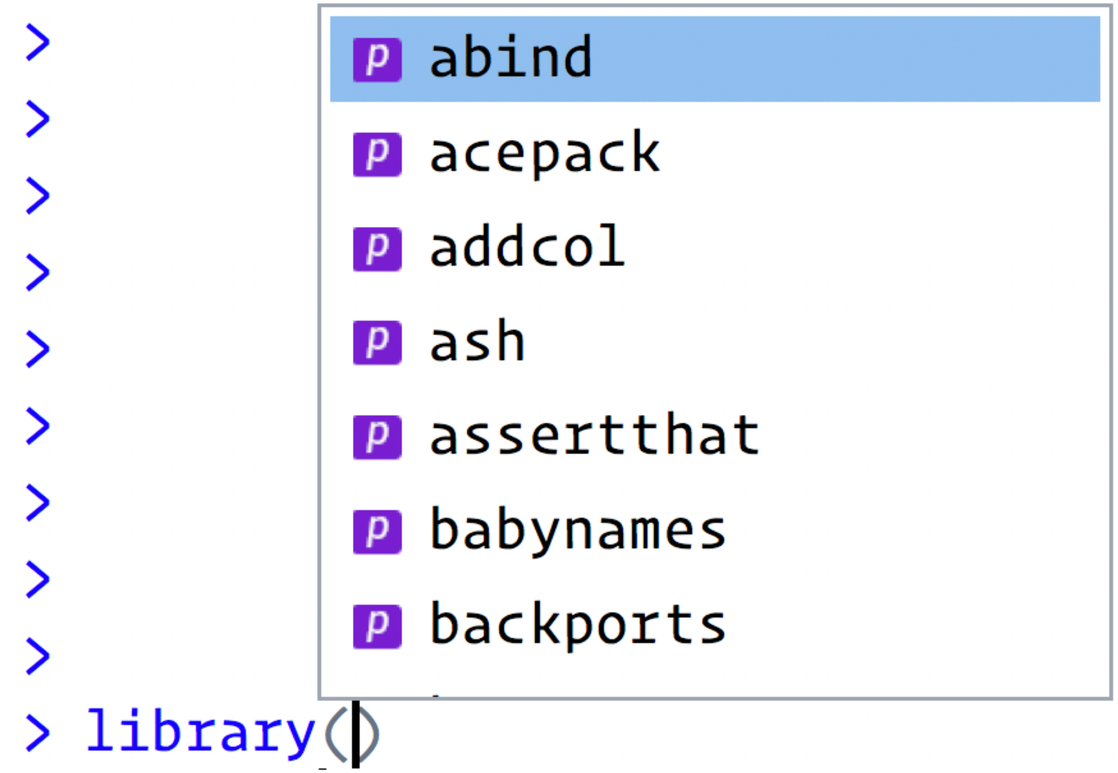
Running Addins

What next?

```
df %>%
  filter(n > 1e6) %>%
  mutate(x = f(y))) %>%
  ???

# How predictable is next step from
# previous steps?
```

Can we do more with autocomplete?



Where do dialogs and autocomplete intersect?

Learning from examples

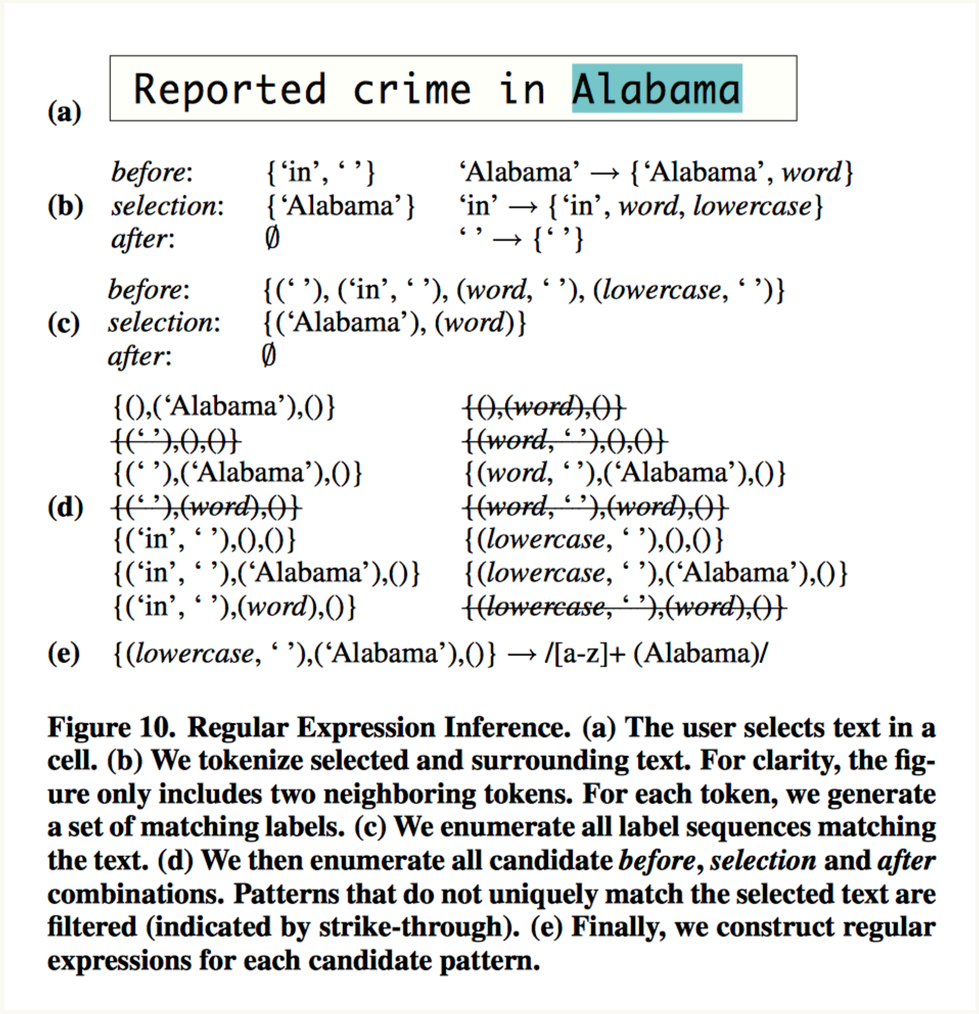
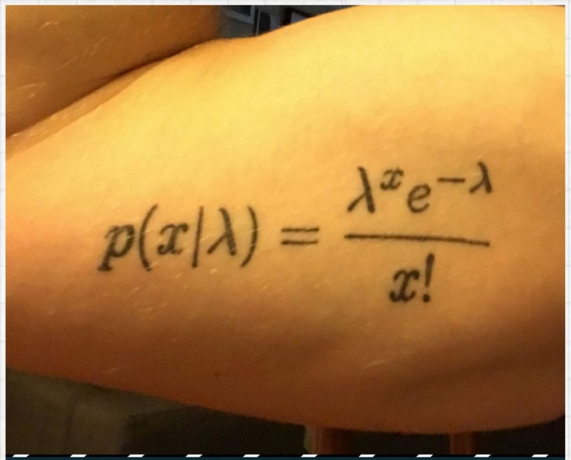
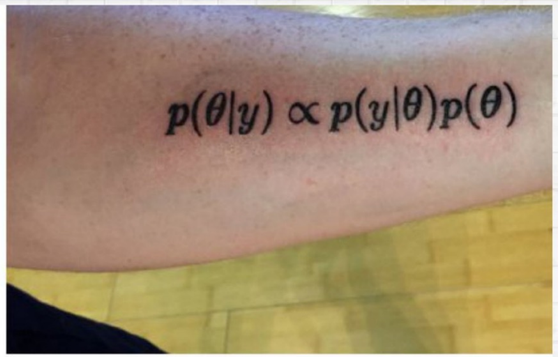


Figure 10. Regular Expression Inference. (a) The user selects text in a cell. (b) We tokenize selected and surrounding text. For clarity, the figure only includes two neighboring tokens. For each token, we generate a set of matching labels. (c) We enumerate all label sequences matching the text. (d) We then enumerate all candidate *before*, *selection* and *after* combinations. Patterns that do not uniquely match the selected text are filtered (indicated by strike-through). (e) Finally, we construct regular expressions for each candidate pattern.

http://vis.stanford.edu/papers/wrangler

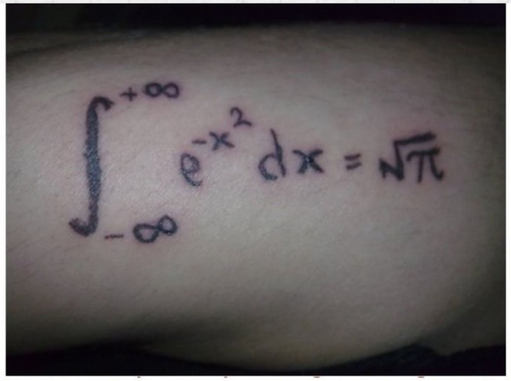



```
mathpix("~/Downloads/armeq1.jpg")
$$
 p ( x / \lambda ) = \frac { \lambda ^ { x } e ^ { - 1} }
$$
```




```
mathpix::mathpix("~/Desktop/DKg07ynVoAA5iMA.jpg")
```

```
$$
 p ( \theta | y ) \alpha p ( y | \theta ) p ( \thet
$$
```




```
> mathpix('c:/Users/Federico/Pictures/1927741_53993664350_665326
Using Mathpix APP_ID=jcarroll
> $$
   \int _ { - \infty } ^ { + \infty } e ^ { - x ^ { 2} } d x = \
  $$
```

https://twitter.com/carroll_jono/status/914254139873361920

Fin

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”

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

Import

readr
readxl
haven
xml2

Tidy

tibble
tidyr

Transform

dplyr
forcats
hms
lubridate
stringr

Visualise

ggplot2

Model

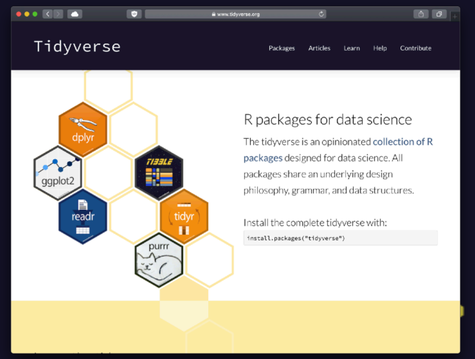
recipes
parsnip

purrr
magrittr

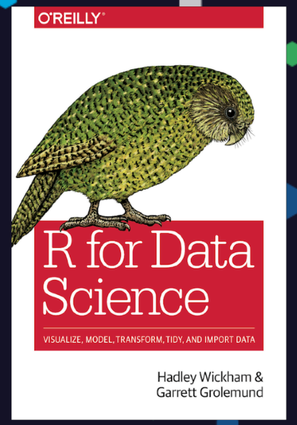
Program

shiny
rmarkdown

Communicate



tidyverse.org



r4ds.had.co.nz