BeginneR Session
-- Nested data handling --

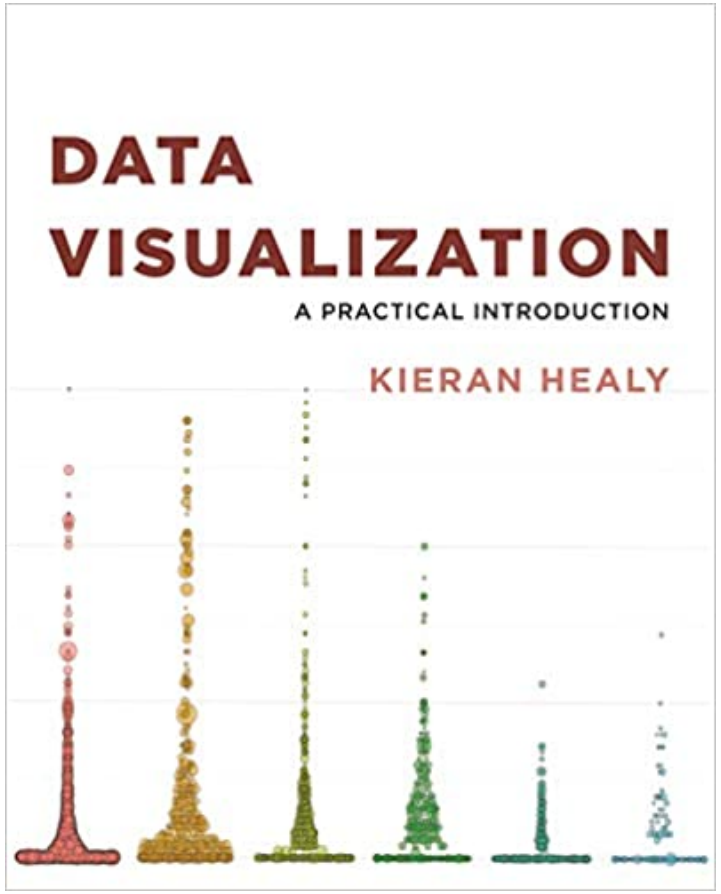
@kilometer00

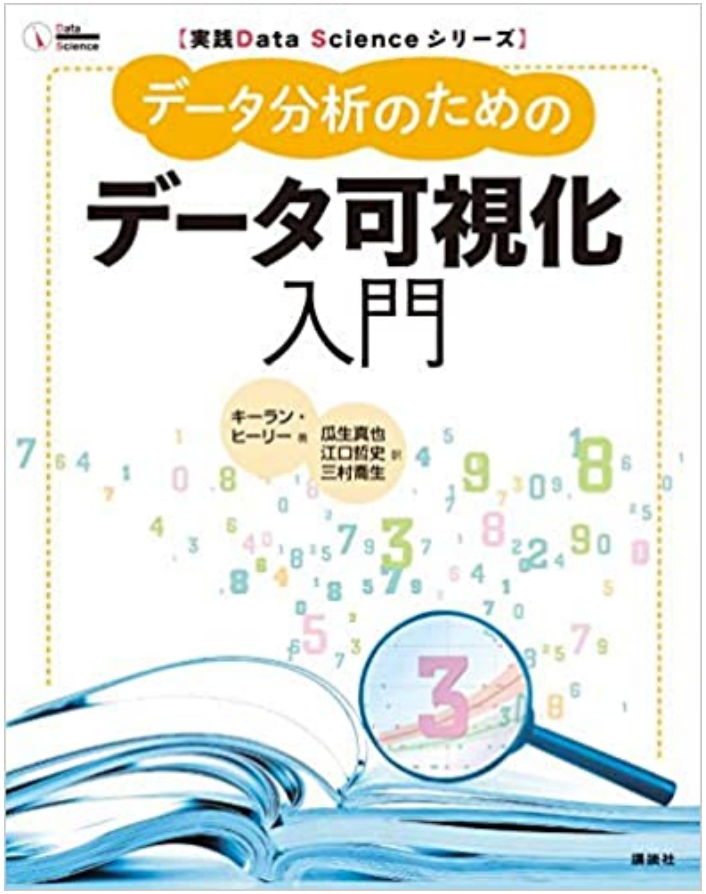
Who！？

Who！？

- @kilometer
- Postdoc Researcher (Ph.D. Eng.)
- Neuroscience
- Computational Behavior
- Functional brain imaging
- R： ~ 10 years

宣伝!!（書籍の翻訳に参加しました。）

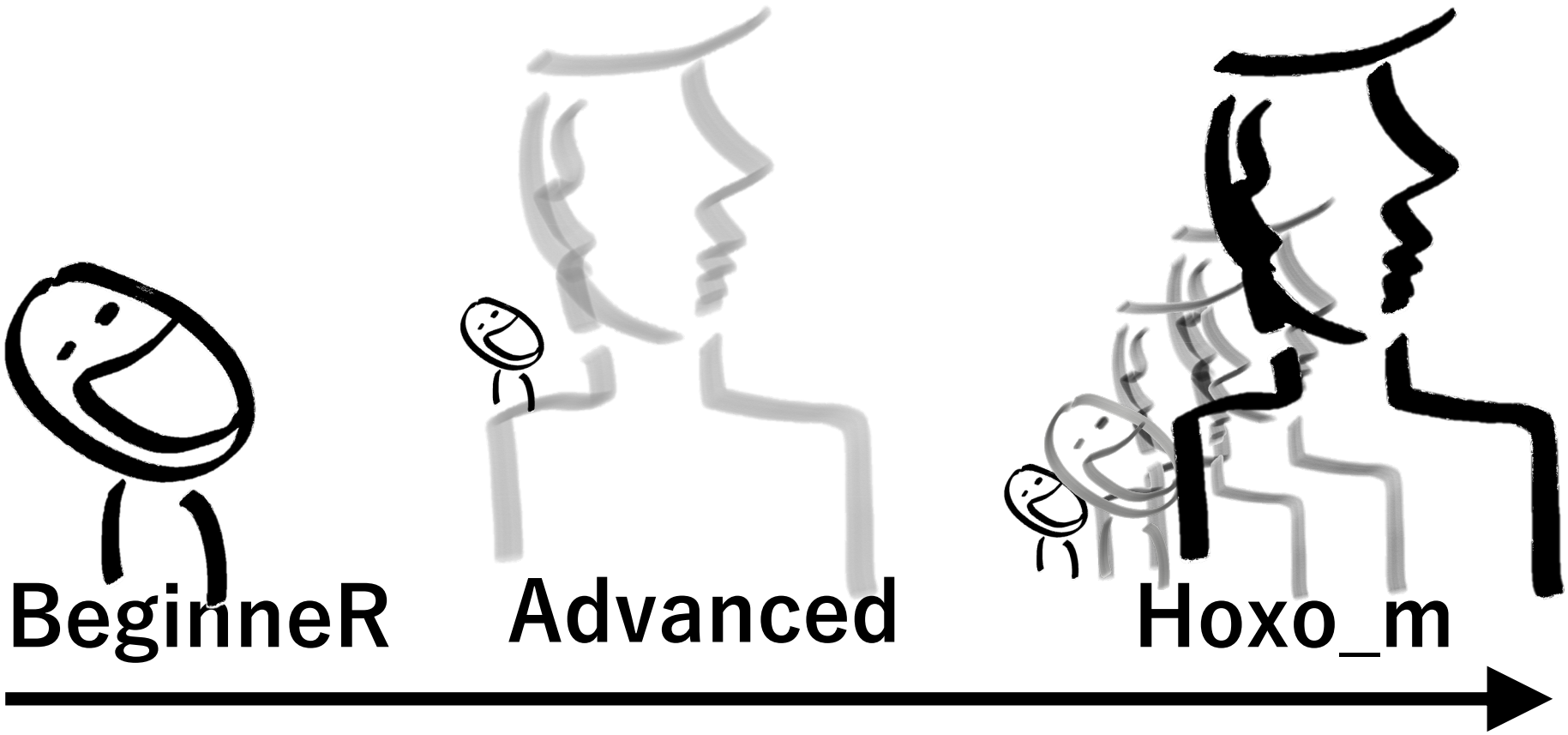




絶賛販売中！

BeginneR Session

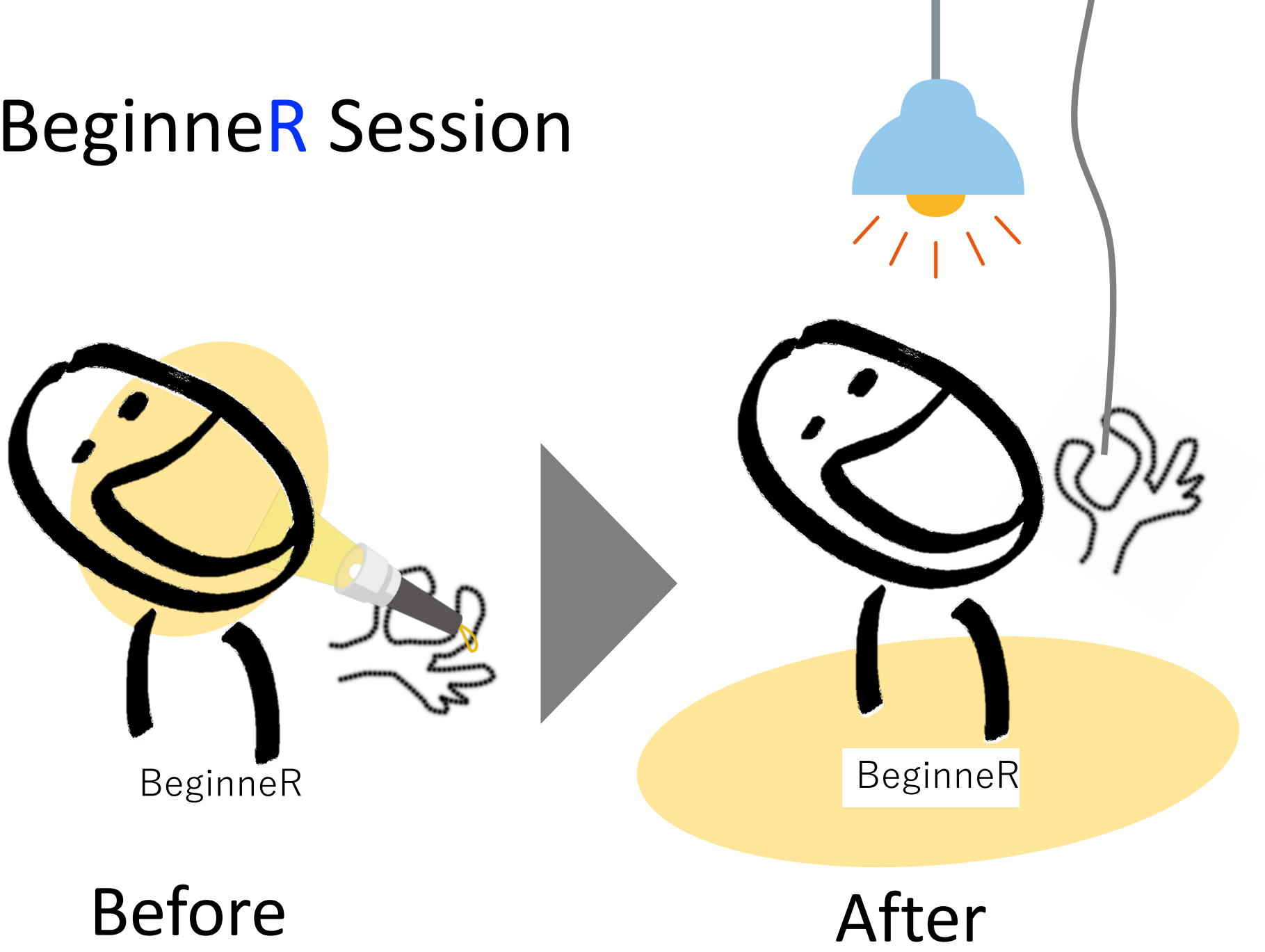
BeginneR



If I have seen further it is by standing on the shoulders of Giants.

-- Sir Isaac Newton, 1676

BeginneR Session



Before

After

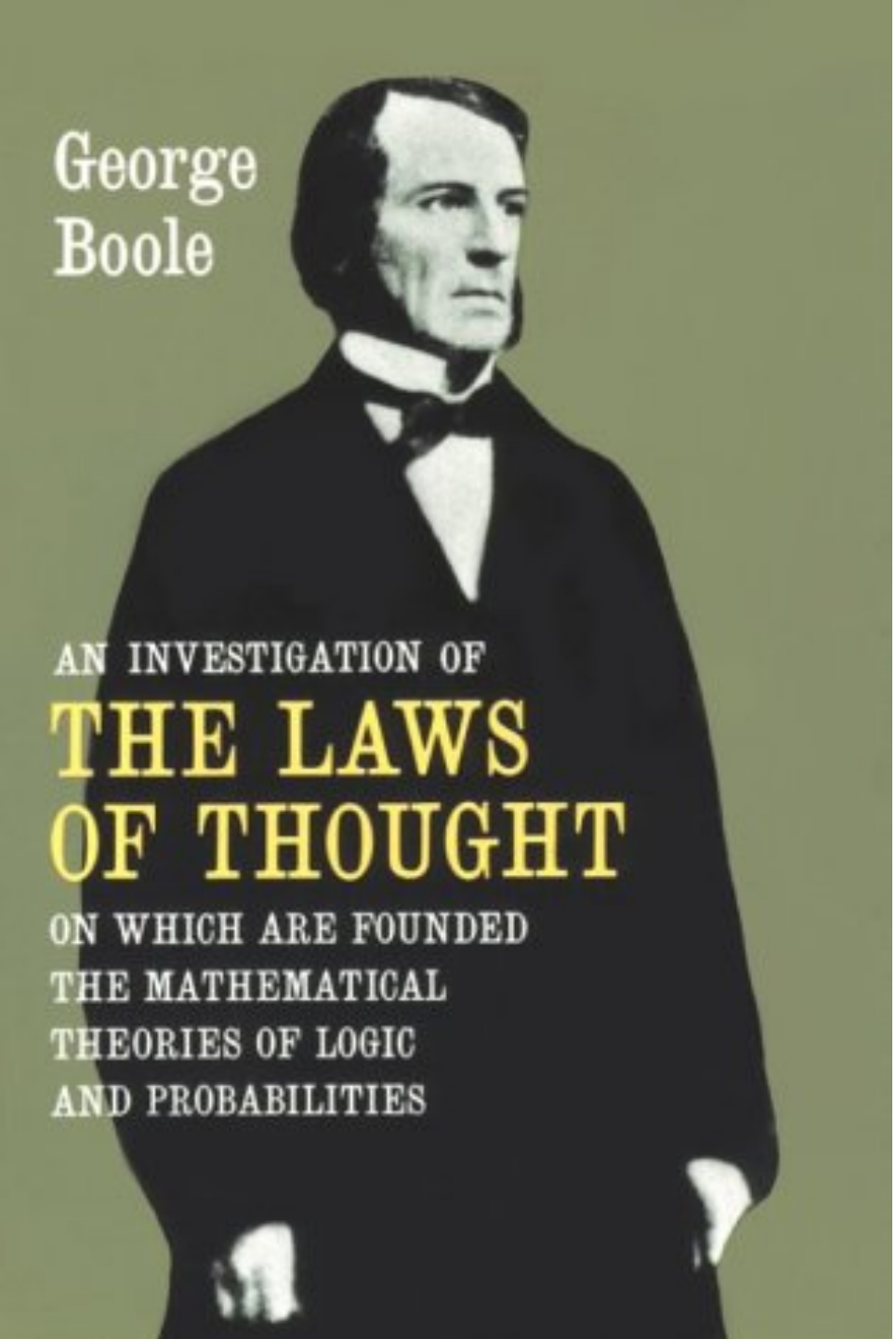
ブール演算子 Boolean Algebra

```
# is A in B?
    "a" != "b"
```

```
[1] TRUE
```

```
# is A in B?
    1 %in% 10:100
```

```
[1] FALSE
```



George Boole
1815 - 1864

Mathematician

&

Philosopher

A Class-Room Introduction to Logic
https://niyamaklogic.wordpress.com/category/laws-of-thoughts/

ブール演算子 Boolean Algebra

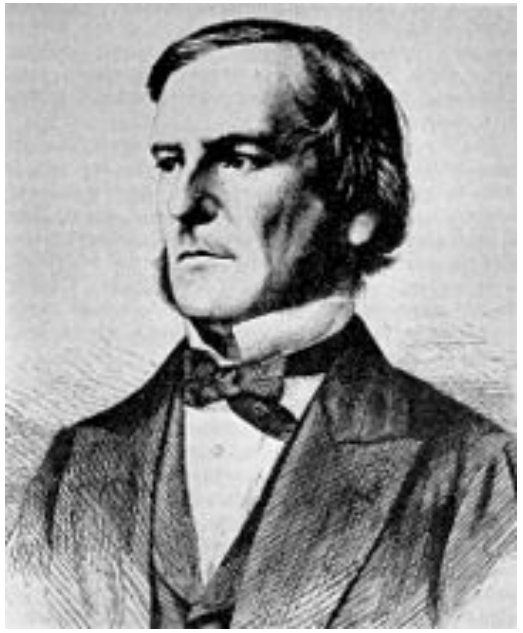
```
# equal to
A == B
```

```
# not equal to
A != B
```

```
# or
A | B
```

```
# and
A & B
```

```
# is A in B?
A %in% B
```



George Boole
1815 - 1864

wikipedia

Pipe algebra %>%

{magrittr}

X %>% f ⟷ f(X)

X %>% f(y) ⟷ f(X, y)

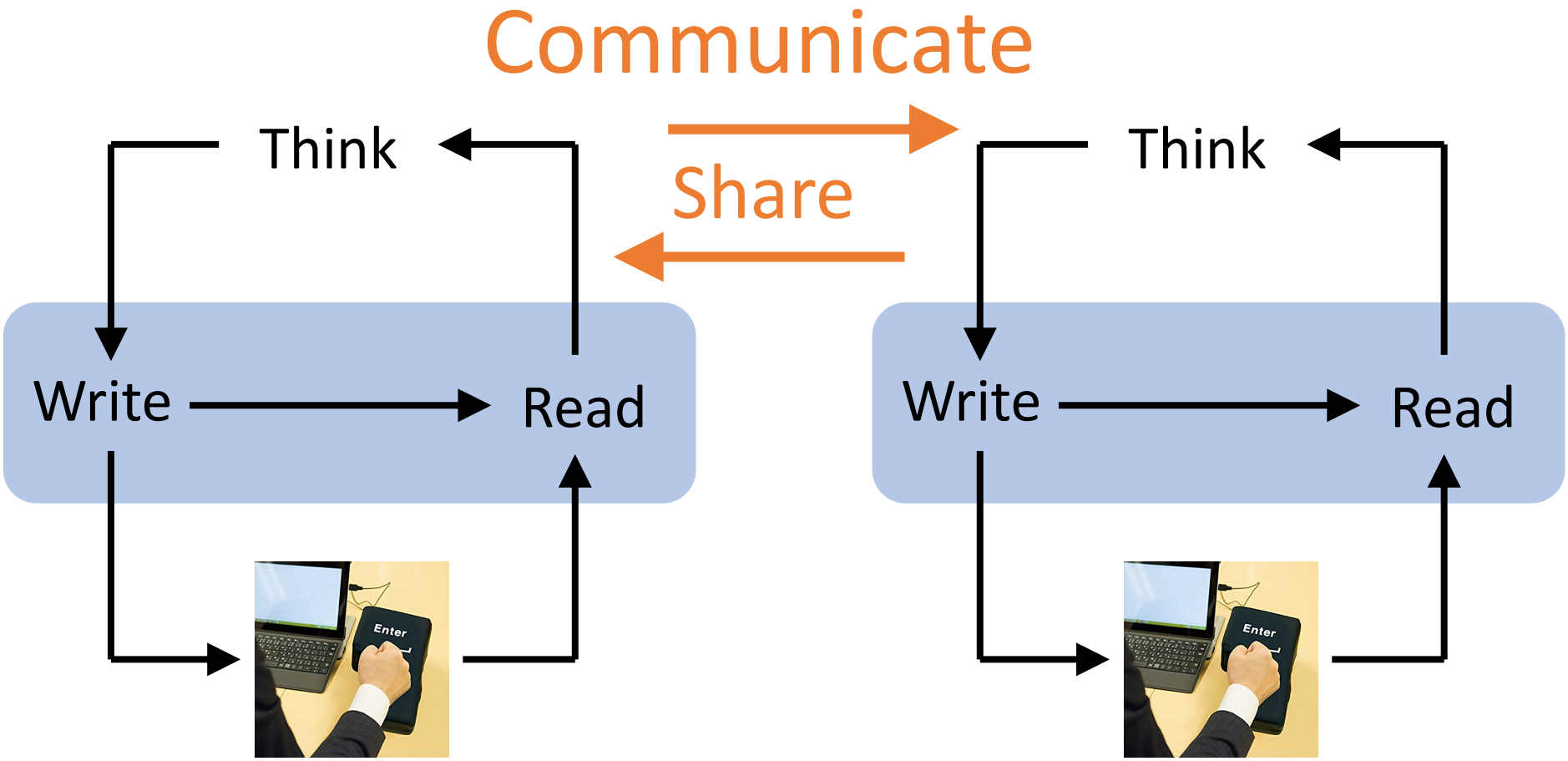
X %>% f %>% g ⟷ g(f(X))

X %>% f(y, .) ⟷ f(y, X)

「dplyr再入門（基本編）」yutanihilation

https://speakerdeck.com/yutannihilation/dplyrzai-ru-men-ji-ben-bian

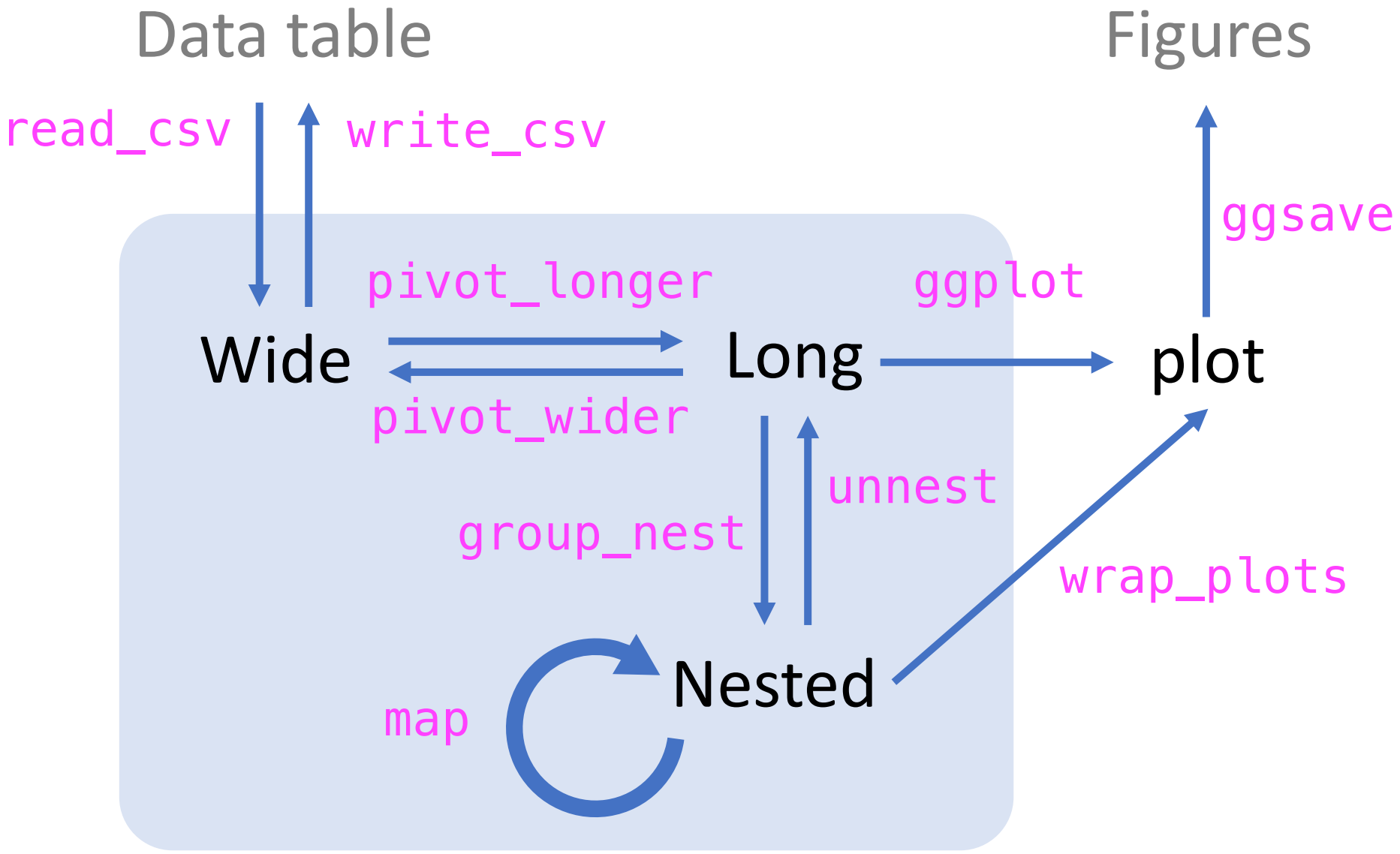
Programing


Communicate
Think
Share
Think
Write
Read
Write
Read
Enter
Enter

BeginneR Session
-- Nested data handling --

@kilometer00

data.frame


Data table
Figures
read_csv
write_csv
ggsave
pivot_longer
ggplot
Wide
Long
plot
pivot_wider
unnest
group_nest
wrap_plots
Nested
map

?data.frame

```
data.frame(
  x = c(1:3),
  y = letters[1:3],
  z = seq(3, 5, by = 1))
```

```
##   x y z
## 1 1 a 3
## 2 2 b 4
## 3 3 c 5
```

?data.frame

```
data.frame(
  x = c(1:3),
  y = letters[1:3],
  z = seq(3, 5, by = 1))
```

```
##   x y z
## 1 1 a 3
## 2 2 b 4
## 3 3 c 5
```

observation

variable

?data.frame

```
a <-
  data.frame(
    x = c(1:3),
    y = letters[1:3],
    z = seq(3, 5, by = 1))


a$x
```

```
## [1] 1 2 3
```

?data.frame

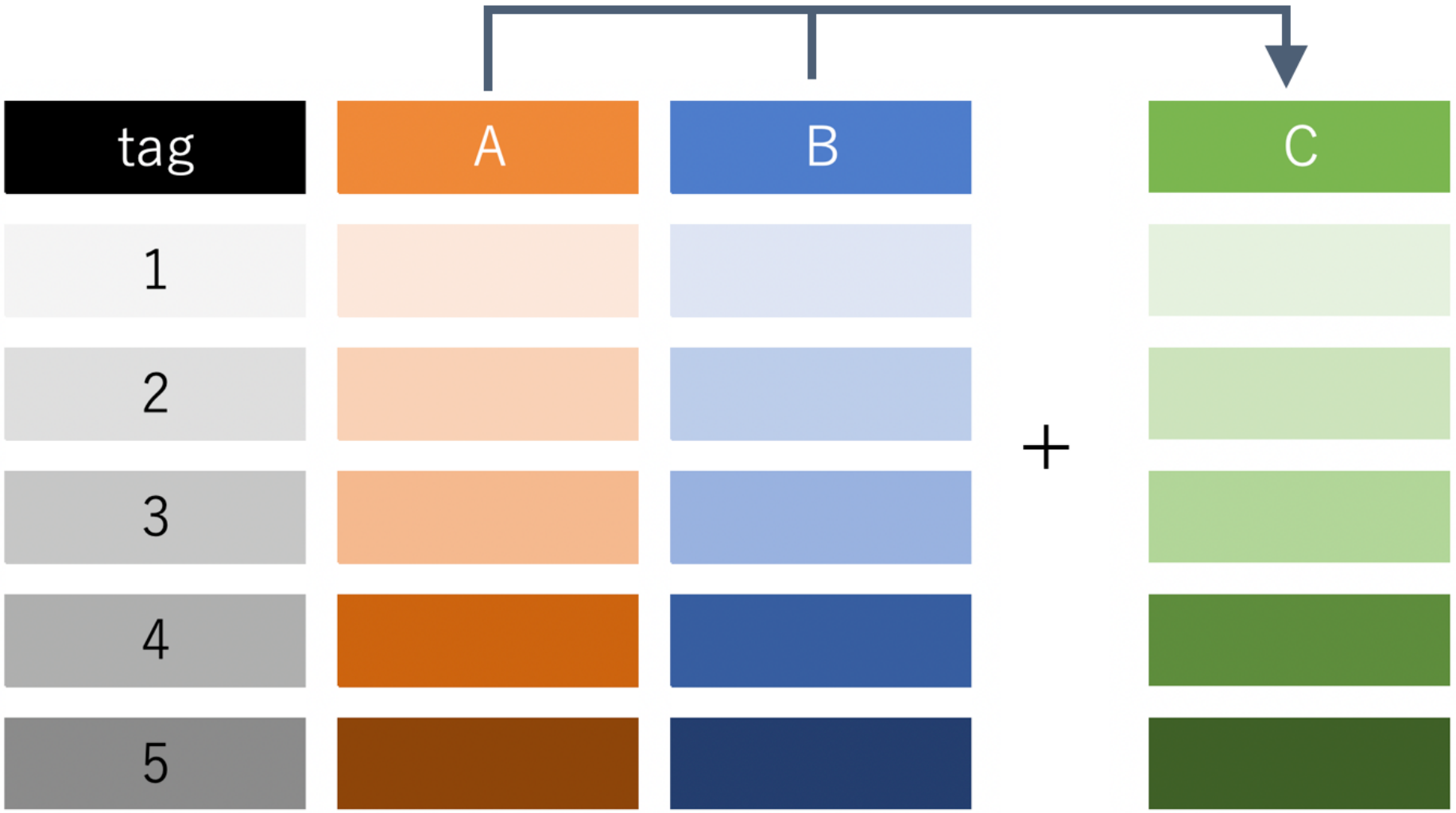
```
a %>% mutate(new = x + 1)
```

```
##   x y z new
## 1 1 a 3   2
## 2 2 b 4   3
## 3 3 c 5   4
```

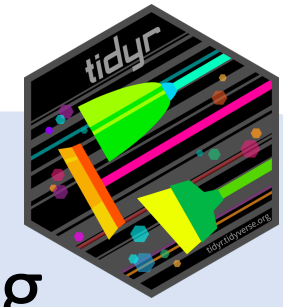
```
a %>% mutate(new = x + y)
```

```
##   x y z new
## 1 1 a 3   4
## 2 2 b 4   6
## 3 3 c 5   8
```

```
mutate(dat, C = fun(A, B))
dat %>% mutate(C = fun(A, B))
```



data.frame



pivot_longer

Wide ⇄ Long

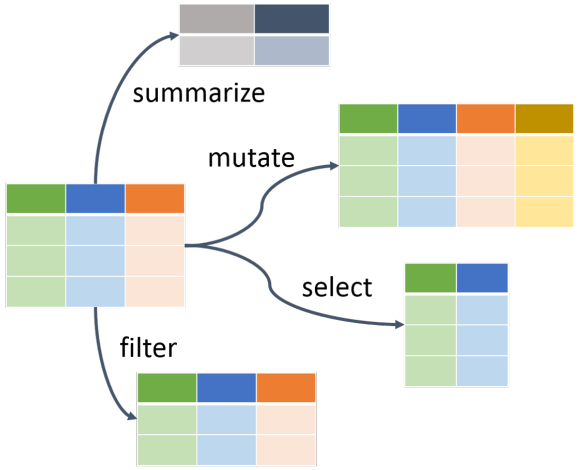
pivot_wider

group_nest ⇅ unnest

map ↻ Nested

Verbs

mutate
filter
select
rename
summarize

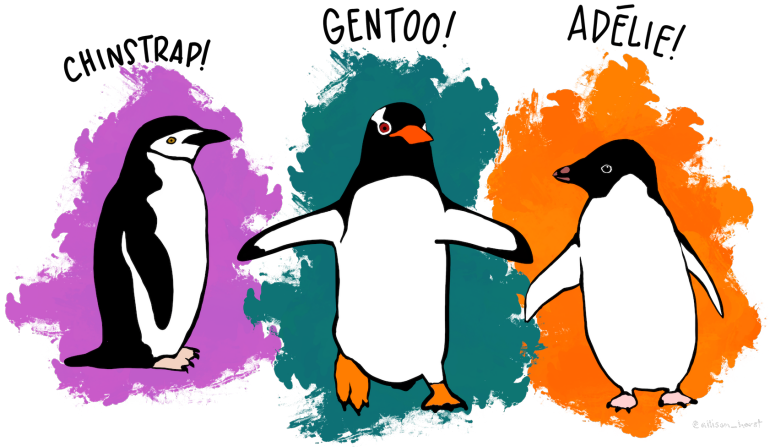


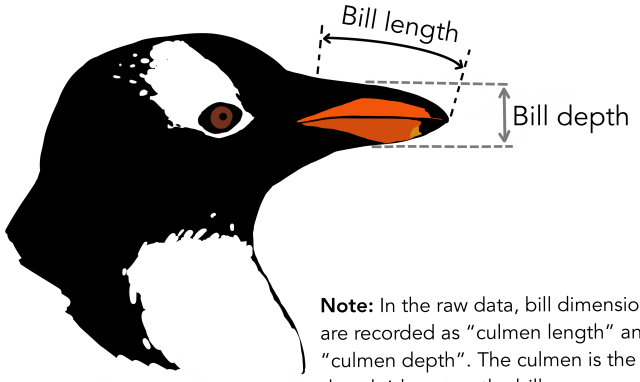

```
library(tidyverse)
```

"Data Manipulation in R with dplyr" Griesemer J. 2019

```
library(palmerpenguins)
penguins %>% head()
```


```
# A tibble: 6 x 8
  species island bill_length_mm bill_depth_mm flipper_length_…
  <fct>   <fct>           <dbl>         <dbl>             <int>
1 Adelie  Torge…            39.1          18.7               181
2 Adelie  Torge…            39.5          17.4               186
3 Adelie  Torge…            40.3          18                 195
4 Adelie  Torge…            NA            NA                  NA
5 Adelie  Torge…            36.7          19.3               193
6 Adelie  Torge…            39.3          20.6               190
# … with 3 more variables: body_mass_g <int>, sex <fct>,
#   year <int>
```



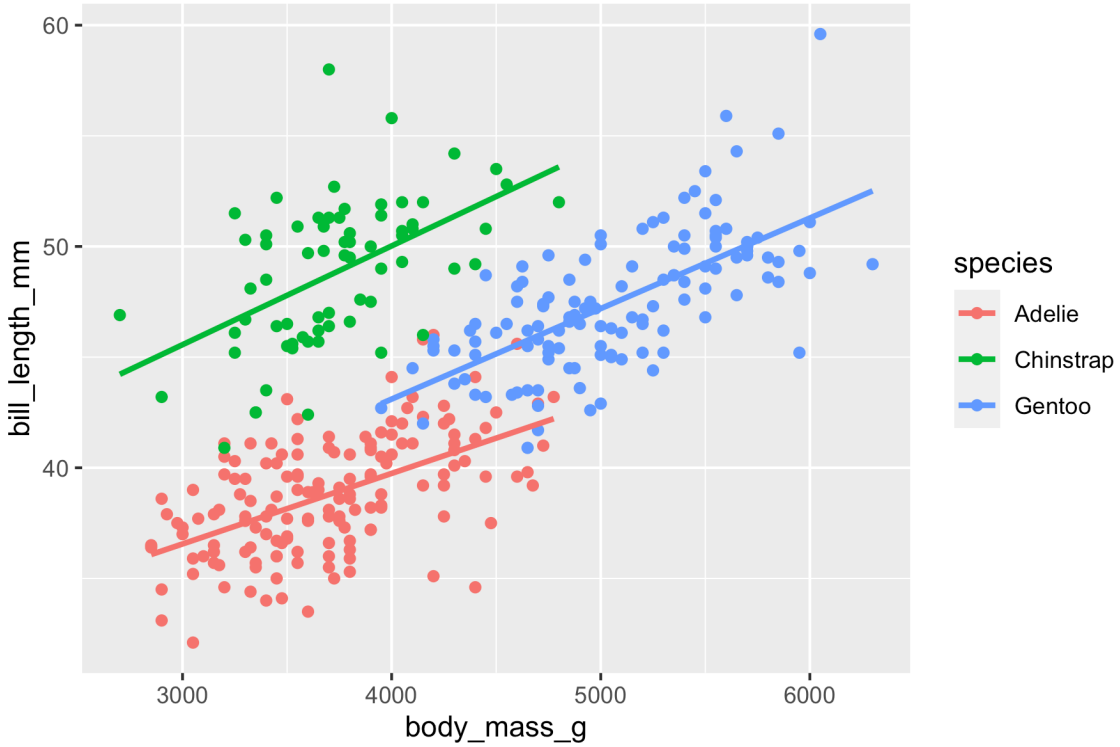


Note: In the raw data, bill dimensions are recorded as "culmen length" and "culmen depth". The culmen is the dorsal ridge atop the bill.

Artwork by @allison_horst

```
ggplot(data = penguins) +
  aes(x = body_mass_g,
      y = bill_length_mm,
      color = species) +
  geom_point()
```

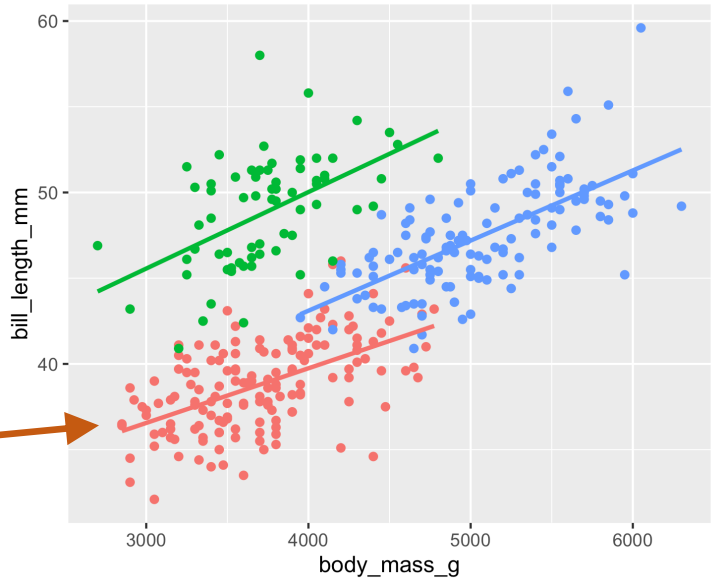
```
ggplot(data = penguins) +
  aes(x = body_mass_g,
      y = bill_length_mm,
      color = species) +
  geom_point() +
  geom_smooth(method = “lm”, se = F)
```



```
penguins_xy <-
  penguins %>%
  mutate(x = body_mass_g,
         y = bill_length_mm)
```

```
penguins_xy %>%
  filter(species == “Adelie”) %>%
  lm(y ~ x, data = .)
```

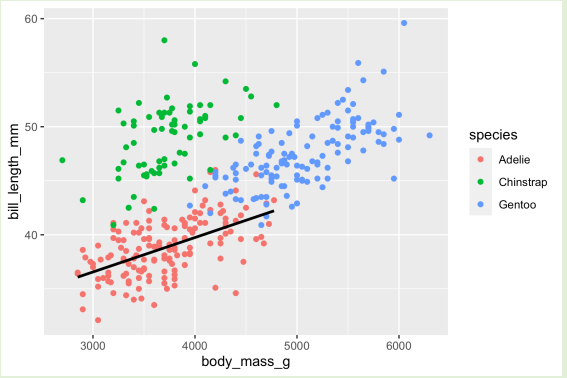
```
Call:
lm(formula = y ~ x, data = .)
Coefficients:
(Intercept)                x
  26.994139     0.003188
```



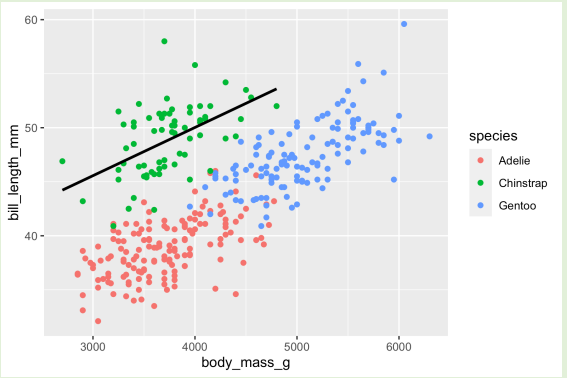
```r
penguins_xy %>%
  filter(species == "Adelie") %>%
  lm(y ~ x, data = .) %>%
  summary()
```

```
Call:
lm(formula = y ~ x, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4208 -1.3690  0.1874  1.4825  5.6168

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.699e+01  1.483e+00  18.201  < 2e-16 ***
x           3.188e-03  3.977e-04   8.015 2.95e-13 ***
---

Residual standard error: 2.234 on 149 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.3013,  Adjusted R-squared:  0.2966
F-statistic: 64.24 on 1 and 149 DF,  p-value: 2.955e-13
```

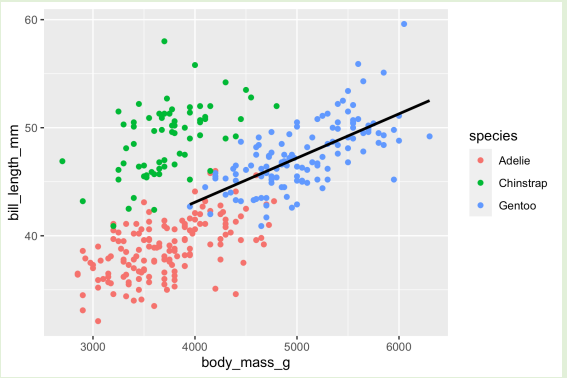
```
penguins_xy %>%
  filter(species == “Adelie”) %>%
  lm(y ~ x, data = .) %>%
  summary()
```




```
penguins_xy %>%
  filter(species == “Chinstrap”) %>%
  lm(y ~ x, data = .) %>%
  summary()
```




```
penguins_xy %>%
  filter(species == “Gentoo”) %>%
lm(y ~ x, data = .) %>%
  summary()
```



```
penguins_xy %>% group_nest(species)
```

```
# A tibble: 3 x 2
  species                    data
  <fct>     <list<tbl_df[,9]>>
1 Adelie               [152 × 9]
2 Chinstrap             [68 × 9]
3 Gentoo               [124 × 9]
```

```
penguins_xy %>% group_nest(species)
```

```
# A tibble: 3 x 2
  species                  data
  <fct>     <list<tbl_df[,9]>>
1 Adelie              [152 × 9]
2 Chinstrap            [68 × 9]
3 Gentoo              [124 × 9]
```

```
penguins_xy %>%
  group_nest(species, island)
```

```
# A tibble: 5 x 3
  species   island                    data
  <fct>     <fct>     <list<tbl_df[,8]>>
1 Adelie    Biscoe                [44 × 8]
2 Adelie    Dream                 [56 × 8]
3 Adelie    Torgersen             [52 × 8]
4 Chinstrap Dream                 [68 × 8]
5 Gentoo    Biscoe               [124 × 8]
```

```
penguins_xy %>% group_nest(species)
```

```
# A tibble: 3 x 2
  species                   data
  <fct>     <list<tbl_df[,9]>>
1 Adelie              [152 × 9]
2 Chinstrap            [68 × 9]
3 Gentoo              [124 × 9]
```

```
penguins_xy %>% group_nest(species) %>%
  .$data %>% .[[1]]
```

```
# A tibble: 152 x 9
   island bill_length_mm bill_depth_mm flipper_length_…
   <fct>           <dbl>         <dbl>             <int>
 1 Torge…           39.1          18.7               181
 2 Torge…           39.5          17.4               186
 3 Torge…           40.3          18                 195
 4 Torge…           NA            NA                  NA
 5 Torge…           36.7          19.3               193
 6 Torge…           39.3          20.6               190
```

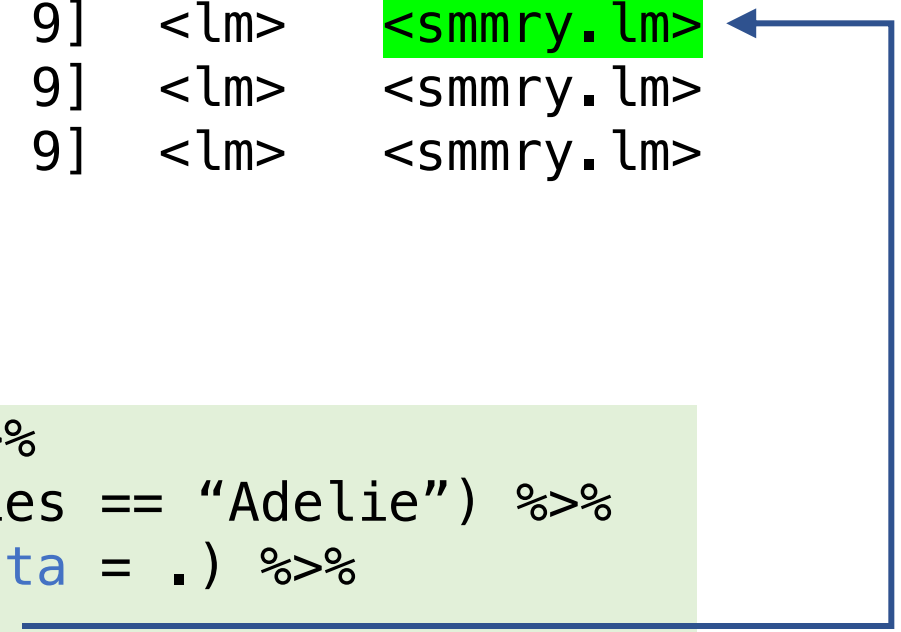
```
penguins_lm <-
  penguins_xy %>%
  group_nest(species) %>%
  mutate(fit = map(data, ~ lm(y ~ x, data = .)),
         summary = map(fit, summary))
```

```
# A tibble: 3 x 4
  species                  data    fit      summary
  <fct>     <list<tbl_df[,9]>> <list>      <list>
1 Adelie             [152 × 9]  <lm>   <smmry.lm>
2 Chinstrap           [68 × 9]  <lm>   <smmry.lm>
3 Gentoo             [124 × 9]  <lm>   <smmry.lm>
```

```
penguins_lm <-
  penguins_xy %>%
  group_nest(species) %>%
  mutate(fit = map(data, ~ lm(y ~ x, data = .)),
         summary = map(fit, summary))
```

```
# A tibble: 3 x 4
  species                  data    fit      summary
  <fct>     <list<tbl_df[,9]>> <list>      <list>
1 Adelie              [152 × 9]  <lm>    <smmry.lm>
2 Chinstrap            [68 × 9]  <lm>    <smmry.lm>
3 Gentoo              [124 × 9]  <lm>    <smmry.lm>
```

```
penguins_xy %>%
  filter(species == “Adelie”) %>%
  lm(y ~ x, data = .) %>%
  summary()
```



```r
penguins_lm <-
  penguins_xy %>%
  group_nest(species) %>%
  mutate(fit = map(data, ~ lm(y ~ x, data = .)),
         summary = map(fit, summary))
```

```r
penguins_lm$summary[[1]]
```

```
Call:
lm(formula = y ~ x, data = .)

Residuals:
   Min     1Q  Median     3Q    Max
-6.4208 -1.3690  0.1874  1.4825  5.6168

Coefficients:
             Estimate Std.         Error  t value   Pr(>|t|)
(Intercept) 2.699e+01  1.483e+00 18.201   < 2e-16 ***
x             3.188e-03  3.977e-04  8.015   2.95e-13 ***
```

```
penguins_lm <-
  penguins_xy %>%
  group_nest(species) %>%
  mutate(fit = map(data, ~ lm(y ~ x, data = .)),
         summary = map(fit, summary),
         a = map_dbl(fit, ~ .$coefficients[2]),
         R2 = map_dbl(summary, ~ .$r.squared))
```

```
# A tibble: 3 x 6
  species     data               fit      summary      a      R2
  <fct> <list<tbl_df[,9]>> <list>     <list>  <dbl>   <dbl>
1 Adelie        [152 × 9]   <lm>  <smmry.lm> 0.00319  0.301
2 Chinstrap      [68 × 9]   <lm>  <smmry.lm> 0.00446  0.264
3 Gentoo        [124 × 9]   <lm>  <smmry.lm> 0.00409  0.448
```

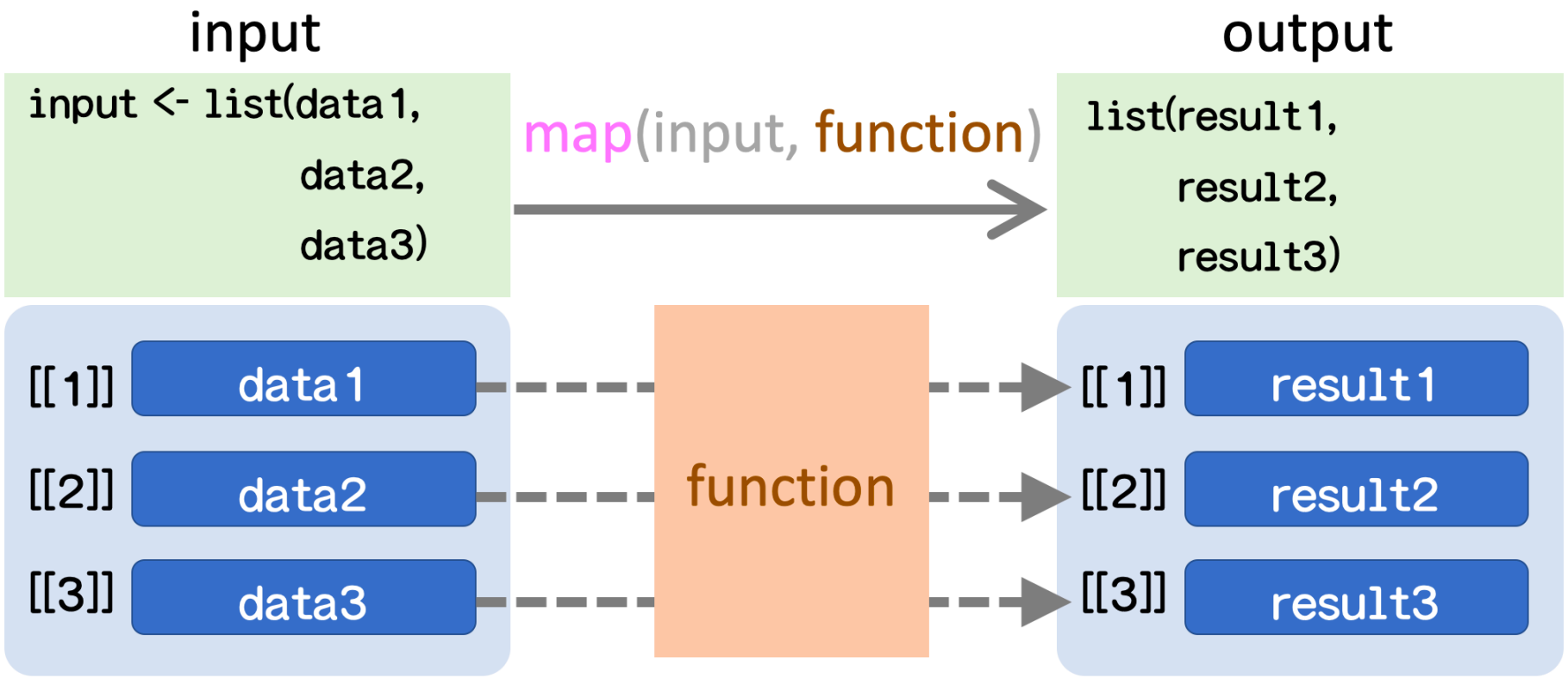
Wrapper functions

`map_dbl()`, `map_chr()`

`map_dfc()`, `map_dfr()`

1. group_nest()
2. mutate()
3. map()

?map


input
output
input <- list(data1,
data2,
data3)
map(input, function)
list(result1,
result2,
result3)
[[1]]
data1
[[2]]
data2
[[3]]
data3
function
[[1]]
result1
[[2]]
result2
[[3]]
result3

```
?map
```

```
f <- function(num){ num * 4 }
```

```
dat <- 1:4
f(num = dat)
```

```
[1] 4 8 12 16
```

```
?map
```

```
f <- function(num){ num * 4 }
```

```
dat <- 1:4
f(num = dat)
```

```
[1] 4 8 12 16
```

```
dat <- list(1:4, 7:4)
f(num = dat)
```

Error in num*4 : non-numeric argument to binary operator

```
?map
```

```
f <- function(num){ num * 4 }
```

```
dat <- list(1:4, 7:4)
```

```
f(num = dat)
map(.x = dat, .f = f)
```

```
[[1]]
[1]  4  8 12 16

[[2]]
[1] 28 24 20 16
```

```
?map
```

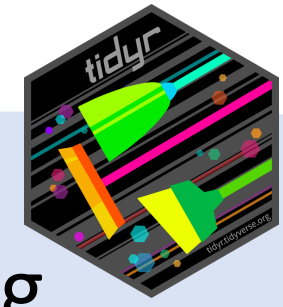
```
f <- function(num){ num * 4 }
```

```
dat <- list(1:4, 7:4)
map(.x = dat, .f = f)
```

by using `for`

```
result <- NULL
for(i in 1:length(dat)){
  result[[i]] <- f(dat[[i]])
}
```

data.frame



pivot_longer

Wide ⇄ Long

pivot_wider

group_nest ⇓ ⇑ unnest

map ↻ Nested

Verbs

mutate
filter
select
rename
summarize

```
?map
```

```
f <- function(num){ num * 4 }
```

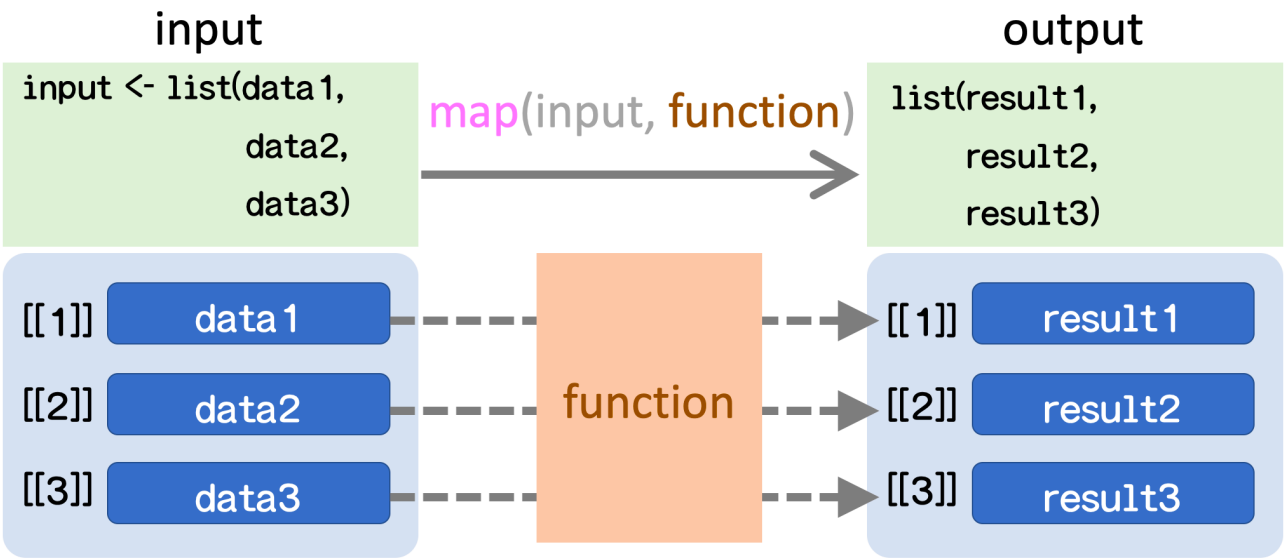
```
dat <- list(1:4, 7:4)
map(.x = dat, .f = f)
```

```
map(dat, f)
map(.x = dat, ~ f(num = .x))
map(.x = dat, function(num){num * 4})
map(dat, ~ {.x * 4})
map(dat, ~ {. * 4})
```

group_nest -> mutate -> map

```
penguins_xy %>%
  group_nest(species) %>%
  mutate(fit = map(data, ~ lm(y ~ x, data = .)))
```

```
# A tibble: 3 x 4
  species                  data                      fit
  <fct>     <list<tbl_df[,9]>>                   <list>
1 Adelie              [152 × 9] -> lm(y ~ x) ->   <lm>
2 Chinstrap            [68 × 9] -> lm(y ~ x) ->   <lm>
3 Gentoo              [124 × 9] -> lm(y ~ x) ->   <lm>
```



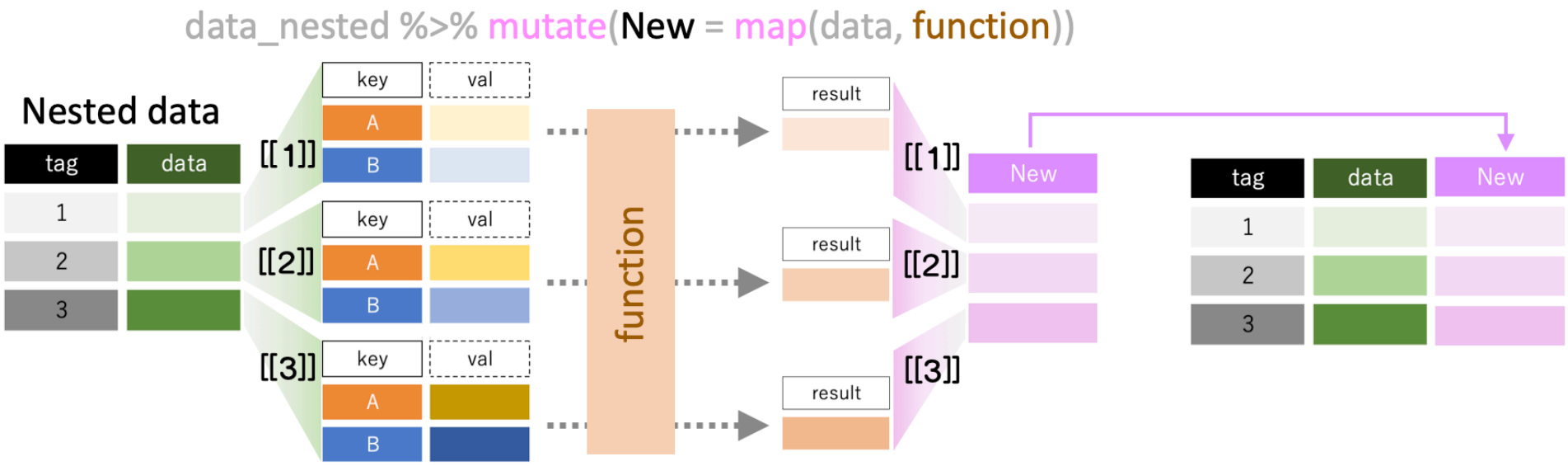
group_nest -> mutate -> map

```
penguins_xy %>%
  group_nest(species) %>%
  mutate(fit = map(data, ~ lm(y ~ x, data = .))
```

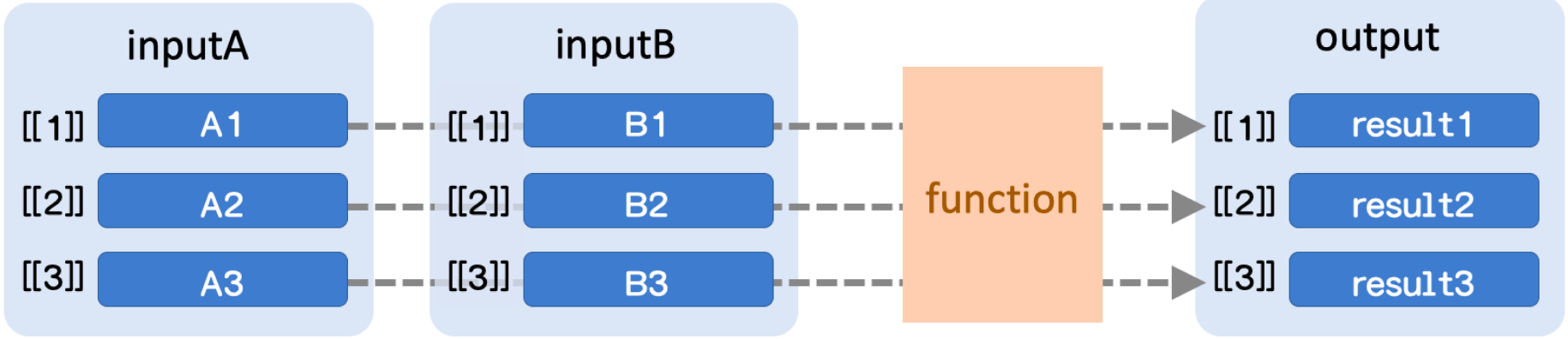
```
f <- function(dat){ lm(y ~ x, data = dat }

penguins_xy %>%
  group_nest(species) %>%
  mutate(fit = map(data, f))
```

```
penguins_xy %>%
  group_nest(species) %>%
  mutate(fit = map(data,
                   function(dat){
                     lm(y ~ x, data = dat)})
```

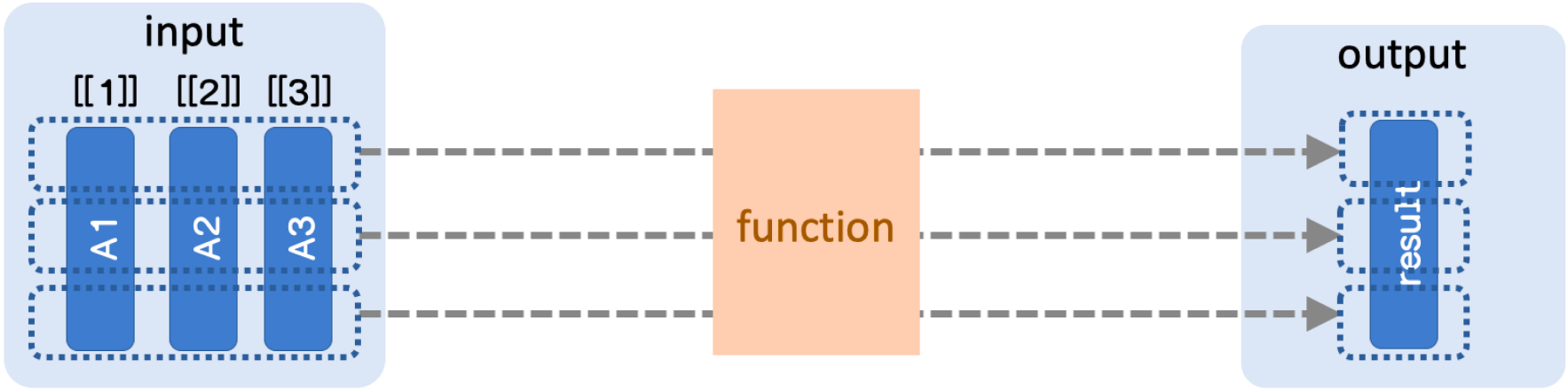
1. group_nest()
2. mutate()
3. map()



map2(.x = inputA, .y = inputB, function(.x, .y))



pmap(input, function(A1, A2, A3))



@kilometer が2020年05月09日に投稿

nested dataでggplot

R, map, ggplot2, Nest

LG TM 13

@kilometer が2019年11月19日に投稿

map脳になろう、もしくはnested dataのハンドリング

R

LG TM 40

@kilometer が2019年02月13日に投稿

{purrr} mapを導入しよう。

R, map, purrr

LG TM 37

@kilometer が2019年01月28日に投稿

{tidyr} nestしていこう。

R, purrr, tidyr

LG TM 51

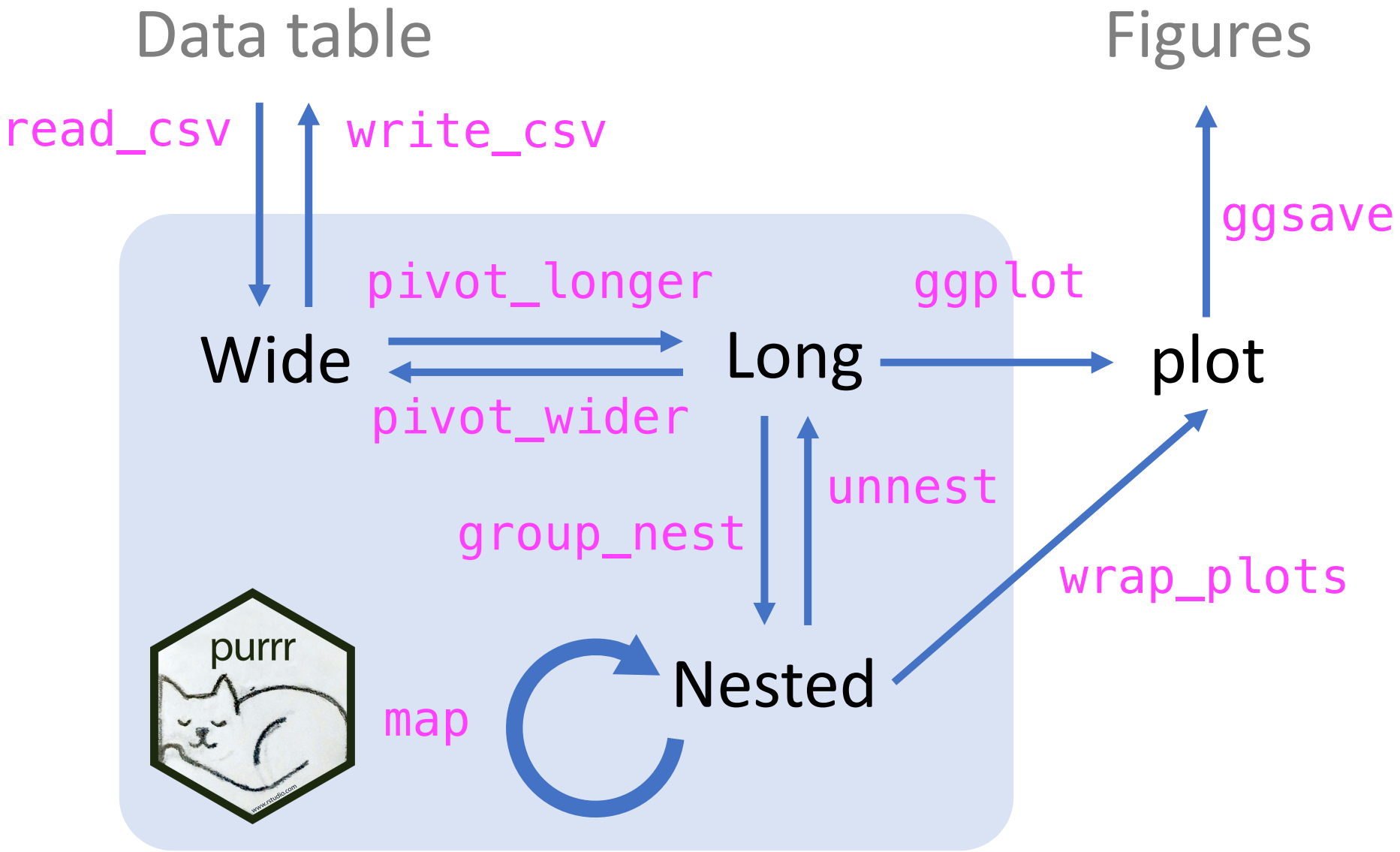
@kilometer が2018年12月22日に投稿

R言語入門（裏口）-- Landscape with R --

R, プログラミング言語, 裏口

LG TM 58

data.frame


Data table
Figures
read_csv
write_csv
ggsave
pivot_longer
ggplot
Wide
Long
plot
pivot_wider
unnest
group_nest
wrap_plots
purrr
map
Nested


Enjoy!!