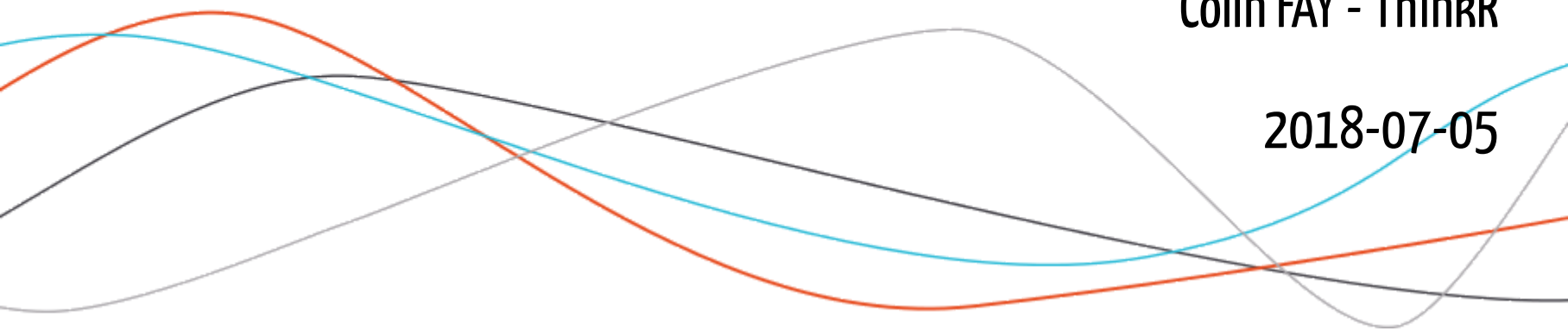
Vous allez aimer

avoir {purrr}

Colin FAY - ThinkR

2018-07-05



{purrr} ?

- Core tidyverse
- "Functional Programming Tools"
- 0.0.0.9000 <- "29 Nov 2014"

"You can't just treat everything as a list."

Iterate

```
lapply(X, FUN, ...)
sapply(X, FUN, ..., simplify = TRUE, USE.NAMES = TRUE)
vapply(X, FUN, FUN.VALUE, ..., USE.NAMES = TRUE)
tapply(X, INDEX, FUN = NULL, ..., default = NA, simplify = TRUE)
mapply(FUN, ..., MoreArgs = NULL, SIMPLIFY = TRUE, USE.NAMES = TRUE)
eapply(env, FUN, ..., all.names = FALSE, USE.NAMES = TRUE)
```

VS

```
map(.x, .f, ...)
map_if(.x, .p, .f, ...)
map_at(.x, .at, .f, ...)
map_lgl(.x, .f, ...)
map_chr(.x, .f, ...)
map_int(.x, .f, ...)
map_dbl(.x, .f, ...)
map_dfr(.x, .f, ..., .id = NULL)
map_dfc(.x, .f, ...)
```

Extract

```
lapply(list, function(x) x$tweets)

lapply(list, function(x) x[2])

lapply(list, function(x) nchar(x))

do.call( rbind,lapply(list, function(x) x$df) )
```

VS

```
map(list, "tweets")

map(list, 2)

map(list, nchar)

map_dfr(list, "df")
```

Lambda functions

```
lapply(list, function(x) x * 10)
```

VS

```
map(list, ~ .x + 2)
```

```
mapply(function(x, y) x + y, list1, list2)
```

VS

```
map2(list1, list2, ~ .x + .y)
```

Type stable

```
sapply(iris$Sepal.Length, as.data.frame) %>% class()
```

```
#> [1] "list"
```

```
sapply(iris$Sepal.Length, as.numeric) %>% class()
```

```
#> [1] "numeric"
```

VS

```
map_dfr(iris$Sepal.Length, as.data.frame) %>% class()
```

```
#> [1] "data.frame"
```

```
map_dbl(iris$Sepal.Length, as.numeric) %>% class()
```

```
#> [1] "numeric"
```

Selected actions

```
sapply(iris[, sapply(iris, is.numeric)], mean)
```

VS

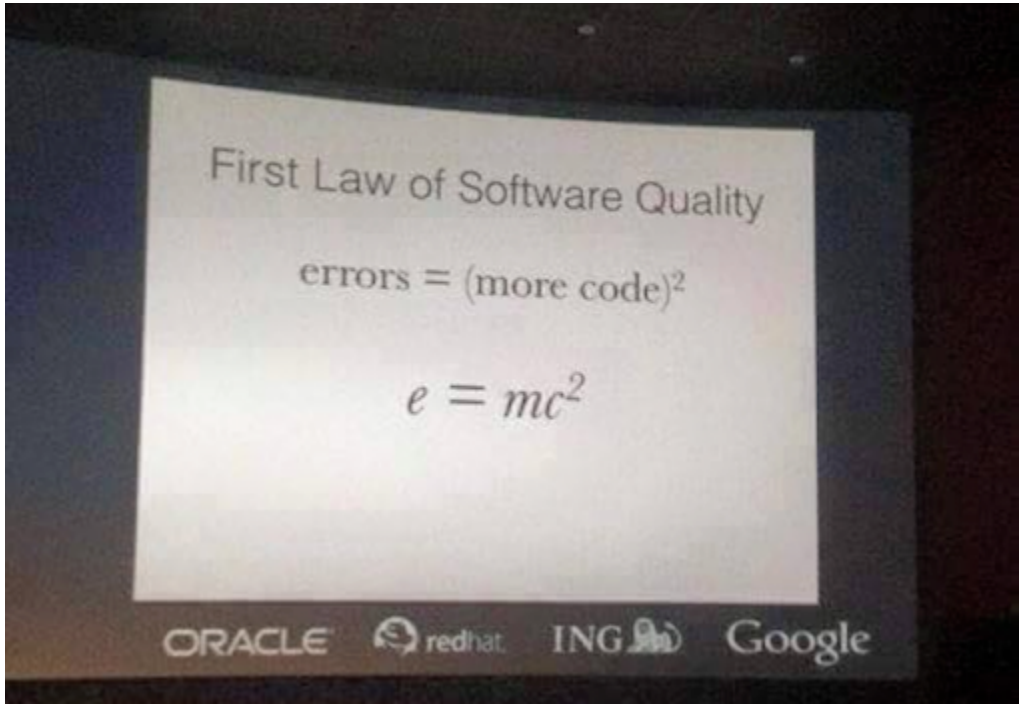
```
map_if(iris, is.numeric, mean)
```

```
sapply(iris[, c("Sepal.Length", "Sepal.Width")], mean)
```

VS

```
map_at(iris, c("Sepal.Length", "Sepal.Width"), mean)
```

e = mc2


First Law of Software Quality
errors = (more code)²
$e = mc^2$
ORACLE
redhat.
ING
Google

Cleaner code

```
coef(summary(lm(Sepal.Length ~ Species, data = iris)))
coef(summary(lm(Pepal.Length ~ Species, data = iris)))
coef(summary(lm(Sepal.Width ~ Species, data = irirs)))
coef(summary(lm(Sepal.Length ~ Species, data = iris)))
```

VS

```
coef_lm <- compose(coef, summary, lm)
coef_lm(Sepal.Length ~ Species, data = iris)
coef_lm(Petal.Length ~ Species, data = iris)
coef_lm(Sepal.Width ~ Species, data = iris)
coef_lm(Petal.Width ~ Species, data = iris)
```

Less code, more rock


```
sapply(airquality, mean, trim = 2, na.rm = TRUE)
sapply(mtcars, mean, trim = 2, na.rm = TRUE)
sapply(volcano, mean, trim = 2, na.rm = TRUE)
```

VS

```
my_mean <- partial(mean, trim = 2, na.rm = TRUE)
map_dbl(airquality, my_mean)
map_dbl(mtcars, my_mean)
map_dbl(volcano, my_mean)
```

I Am Groot


```
sapply(iris, max)
sapply(airquality, max)
sapply(volcano, max)
sapply(iris, max)
```

VS

```
possible_max <- possibly(max, otherwise = NULL)
map(iris, possible_max)
map(airquality, possible_max)
map(volcano, possible_max)
map(iris, possible_max)
```

Predicates

```
iris[ , sapply(iris, is.numeric) ]
```

VS

```
keep(iris, is.numeric)
```

```
iris[, ! sapply(iris, is.numeric) ]
```

VS

```
discard(iris, is.numeric)
```

Pipeline


```
rounded_mean <- compose(
  partial(round, digits = 1),
  partial(mean, trim = 2, na.rm = TRUE)
  )
map(
  list(airquality, mtcars),
  ~ map_dbl(.x, rounded_mean)
)
```

```
#> [[1]]
#>   Ozone Solar.R    Wind    Temp   Month     Day
#>    31.5   205.0     9.7    79.0     7.0    16.0
#>
#> [[2]]
#>   mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
#>  19.2   6.0 196.3 123.0   3.7   3.3  17.7   0.0   0.0   4.0   2.0
```

Merci !

Colin Fay

colin@thinkr.fr

http://twitter.com/_colinfay

http://twitter.com/thinkr_fr

https://github.com/ColinFay

https://github.com/Thinkr-open

https://thinkr.fr/

https://rtask.thinkr.fr/

http://colinfay.me/