

Iteration - r4ds

Tomoya Fukumoto

2019-09-06

Iteration

繰り返し作業をどうやって自動化するための二つの手法

1. ループ
2. 関数型プログラミング (functional programming)

準備

```
library(tidyverse)
```

ループに関わるのは base ライブラリ

FP に関わるのは purrr ライブラリ

21.2 For loops

最も標準的なループ

例：各行の median を求める（ループなし）

```
df <- tibble(  
  a = rnorm(10),  
  b = rnorm(10),  
  c = rnorm(10),  
  d = rnorm(10)  
)  
median(df$a)  
median(df$b)  
median(df$c)  
median(df$d)
```

例：各行の median を求める（ループ）

```
output <- vector("double", ncol(df)) # 1. output
for (i in seq_along(df)) {           # 2. sequence
  output[[i]] <- median(df[[i]])      # 3. body
}
output
```

```
## [1] -0.23397728 -0.35668072  0.01067616 -0.31417659
```

ループの構成要素 output

```
output <- vector("double", ncol(df))
```

- ▶ ループの出力の器
- ▶ ループが始まる前に作る
- ▶ `vector` 関数で型と長さを指定する
- ▶ 長さを指定せずとも処理が遅くなる

ループの構成要素 sequence

```
i in seq_along(df)
```

- ▶ どうループを回すか
- ▶ 一周するたびに `i` がベクトル `seq_along(df)` の中で値を変化させる
- ▶ (`seq_along(df)` は `1:length(df)` とほぼ同じ)

ループの構成要素 body

```
output[[i]] <- median(df[[i]])
```

- ▶ ループで実際に処理する内容
- ▶ 1 回目は `output[[1]] <- median(df[[1]])`
- ▶ 2 回目は `output[[2]] <- median(df[[2]])`

参考文献