

Les verbes dplyr, employés délibérément

filter, select, mutate et arrange — ce que chacun fait d'une valeur manquante, pourquoi if_else() refuse ce que ifelse() accepte en silence, et le across() qui applique une règle à plusieurs colonnes.

Pierre Robentz Cassion

Leçon 5 sur 8

R et le tidyverse pour les données de programme — Les verbes

Ce que couvre cette leçon

- Cinq verbes et un tube
- `filter()` écarte les valeurs manquantes, en silence
- `select()` et ses assistants
- `mutate()` et les deux fonctions conditionnelles
- `across()` : une règle, plusieurs colonnes
- `arrange()`
- `distinct()` et le comptage
- Les jointures, et le contrôle qui les accompagne
- Ce qui vient ensuite

Cinq verbes et un tube — En R

```
library(dplyr)

muac |>
  filter(!is.na(muac_mm)) |>
  mutate(gam = muac_mm < 125) |>
  group_by(commune) |>
  summarise(assessed = n(), cases = sum(gam)) |>
  arrange(desc(cases / assessed))
```

`filter()` écarte les valeurs manquantes, en silence — En R

```
nrow(muac)
#> [1] 4218
```

```
nrow(filter(muac, muac_mm < 125))
#> [1] 319
```

```
sum(is.na(muac$muac_mm))
#> [1] 72
```

`filter()` écarte les valeurs manquantes, en silence — En R

```
# Enfants mesures en dessous de 125 mm.
```

```
filter(muac, muac_mm < 125)
```

```
# Enfants dont on ne sait pas qu'ils sont a 125 mm ou plus -- non mesures compris.
```

```
filter(muac, is.na(muac_mm) | muac_mm < 125)
```

```
#> 391 lignes
```

select() et ses assistants — En R

```
muac |> select(child_id, commune, muac_mm)
muac |> select(-oedema)
muac |> select(starts_with("muac"), ends_with("_date"))
muac |> select(where(is.numeric))
```

`select()` et ses assistants — En R

```
epi |> select(facility = facility_id, month = period, doses = doses_administered)
```

```
muac <- muac |>
  mutate(
    gam = muac_mm < 125,
    sam = muac_mm < 115,
    band = case_when(
      muac_mm < 115 ~ "severe",
      muac_mm < 125 ~ "moderate",
      .default = "normal"
    )
  )
```

```
case_when(c(110, 130, NA) < 125 ~ "case", .default = "not")  
#> [1] "case" "not"  "not"
```

```
band = case_when(  
  is.na(muac_mm) ~ NA_character_,  
  muac_mm < 115  ~ "severe",  
  muac_mm < 125  ~ "moderate",  
  .default = "normal"  
)
```

- `if_else()` plutôt que `ifelse()`

```
ifelse(c(TRUE, FALSE), 1L, "x")  
#> [1] "1" "x"
```

```
if_else(c(TRUE, FALSE), 1L, "x")
```

```
#> Error: Can't combine `true` <integer> and `false` <character>.
```

across() : une règle, plusieurs colonnes — En R

```
muac |>
  summarise(across(c(muac_mm, age_months), ~ mean(.x, na.rm = TRUE)))
#> # A tibble: 1 × 2
#>   muac_mm age_months
#> 1    140.      30.9
```

across() : une règle, plusieurs colonnes — En R

```
survey |> mutate(across(where(is.character), stringr::str_squish))
```

```
survey |> mutate(across(starts_with("fcs_"), ~ if_else(.x > 7, NA_real_, .x)))
```

across() : une règle, plusieurs colonnes — En R

```
survey |> mutate(across(starts_with("fcs_"), \(x) if_else(x > 7, NA_real_, x)))
```

```
muac |> arrange(commune, desc(screening_date))
```

distinct() et le comptage — En R

```
n_distinct(muac$commune)
```

```
#> [1] 12
```

```
muac |> distinct(child_id, .keep_all = TRUE)
```

distinct() et le comptage — En R

```
muac |> count(commune, sort = TRUE)
```

Les jointures, et le contrôle qui les accompagne — En R

```
daily <- attendance |>  
  left_join(roster, by = "student_id")
```

Les jointures, et le contrôle qui les accompagne — En R

```
before <- nrow(attendance)
daily <- attendance |> left_join(roster, by = "student_id")
stopifnot(nrow(daily) == before)
```

Les jointures, et le contrôle qui les accompagne — En R

```
attendance |> anti_join(roster, by = "student_id") |> distinct(student_id)
```

Ce qui vient ensuite

- Les verbes sont acquis.

Lire la leçon complète, avec le code exécutable

`https:`

`//data-analysis.cassion.dev/fr/cours/r-for-programme-data/r-05-dplyr-verbs`