

The dplyr verbs, used deliberately

filter, select, mutate and arrange — what each does with a missing value, why if_else() refuses what ifelse() silently accepts, and the across() that applies one rule to many columns.

Pierre Robentz Cassion

Lesson 5 of 8

R and the Tidyverse for Programme Data — The verbs

What this lesson covers

- Five verbs and a pipe
- `filter()` drops missing values, silently
- `select()` and the helpers
- `mutate()` and the two conditional functions
- `across()`: one rule, many columns
- `arrange()`
- `distinct()` and counting
- Joins, and the check that belongs beside them
- What comes next

Five verbs and a pipe — In 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()` drops missing values, silently — In R

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

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

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

`filter()` drops missing values, silently — In R

```
# Children measured below 125 mm.
```

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

```
# Children not known to be at or above 125 mm -- the unmeasured included.
```

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

```
#> 391 rows
```

select() and the helpers — In 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()` and the helpers — In R

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

mutate() and the two conditional functions — In R

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

`mutate()` and the two conditional functions — In R

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

mutate() and the two conditional functions — In R

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

`mutate()` and the two conditional functions

- `if_else()` rather than `ifelse()`

`mutate()` and the two conditional functions — In R

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

mutate() and the two conditional functions — In R

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

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

`across()`: one rule, many columns — In 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()`: one rule, many columns — In R

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

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

`across()`: one rule, many columns — In R

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

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

distinct() and counting — In R

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

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

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

distinct() and counting — In R

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

Joins, and the check that belongs beside them — In R

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

Joins, and the check that belongs beside them — In R

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

Joins, and the check that belongs beside them — In R

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

What comes next

- The verbs are in hand.

Read the full lesson, with runnable code

`https:`

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