Reading an export with readr and readxl

Column specifications instead of guesses, problems() as a habit rather than a rescue, and the Excel traps — multiple sheets, a header that is not row one, merged cells and serial dates.

Pierre Robentz Cassion

Lesson 2 of 8

R and the Tidyverse for Programme Data — A project that reopens

What this lesson covers

- readr guesses, and tells you what it guessed
- The guess uses the first 1,000 rows
- Declare the columns
- `mean()` returns NA, and that is a feature
- `na` is file-wide, not per column
- `problems()` is a habit, not a rescue
- Verify the read before doing anything with it
- Separators, encodings and thousands
- Excel
- The read function, assembled
- What comes next

readr guesses, and tells you what it guessed — In R

```r
library(readr)
library(here)

PATH <- here("data", "raw", "muac-screening-artibonite-2024.v1.csv")
muac <- read_csv(PATH)
```

readr guesses, and tells you what it guessed — Example

```
child_id       character
commune        character
screening_date date
age_months     numeric
sex            character
muac_mm        numeric
oedema         character
outcome        character
```

The guess uses the first 1,000 rows — In R

```r
# Deliberate: read everything as text, look, then convert.
peek <- read_csv(PATH, col_types = cols(.default = col_character()), n_max = 200)
print(peek)
```

Declare the columns — In R

```r
muac <- read_csv(
  PATH,
  col_types = cols(
    child_id       = col_character(),
    commune        = col_character(),
    screening_date = col_date(format = "%Y-%m-%d"),
    age_months     = col_integer(),
    sex            = col_character(),
    muac_mm        = col_integer(),
    oedema         = col_character(),
    outcome        = col_character()
  ),
  na = c("", "NA", "-99")
)
```

Declare the columns

- Leading zeros survive — `col_character()` on an identifier is the whole lesson: a facility code `007` read as a number…
- The date format is stated — An inferred format can change between files as the mix of values changes
- The sentinel is declared — This register codes an unmeasured MUAC as `-99`

Declare the columns — In R

```r
mean(muac$muac_mm)
#> [1] NA
```

`mean()` returns NA, and that is a feature — In R

```r
mean(muac$muac_mm)
#> [1] NA

mean(muac$muac_mm, na.rm = TRUE)
#> [1] 139.92
```

`mean()` returns NA, and that is a feature

- R is loud where pandas is silent — The `NA` is a question: *do you know there are missing values here, and have you...

`mean()` returns NA, and that is a feature — In R

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

`na` is file-wide, not per column — In R

```r
muac <- read_csv(PATH, na = c("", "NA", "-99"))
```

`na` is file-wide, not per column — In R

```r
muac <- read_csv(PATH, na = c("", "NA")) |>
  dplyr::mutate(muac_mm = dplyr::na_if(muac_mm, -99L))
```

`problems()` is a habit, not a rescue — In R

```r
issues <- problems(muac)
nrow(issues)
#> [1] 0
```

`problems()` is a habit, not a rescue — In R

```r
if (nrow(problems(muac)) > 0) {
  print(problems(muac))
  stop("Input did not match its column specification.")
}
```

Verify the read before doing anything with it — In R

```r
library(dplyr)

check <- function(df, expected_rows = NULL) {
  cat("rows/cols ", nrow(df), ncol(df), "\n")
  print(dplyr::glimpse(df))
  print(colSums(is.na(df)))
  if (!is.null(expected_rows)) {
    stopifnot(nrow(df) == expected_rows)
  }
}

check(muac, expected_rows = 4218)
```

Separators, encodings and thousands — In R

```
epi <- read_csv2(PATH)                      # ; separator, , decimal
epi <- read_delim(PATH, delim = "\t")       # tabs
```

Separators, encodings and thousands — In R

```
muac <- read_csv(PATH, locale = locale(encoding = "UTF-8"))
muac <- read_csv(PATH, locale = locale(encoding = "latin1"))    # what Excel often writes
```

Separators, encodings and thousands — In R

```
epi <- read_csv(PATH, locale = locale(grouping_mark = ","))
```

Excel — In R

```
library(readxl)

excel_sheets(path)
#> [1] "Cover" "Data" "Codebook" "Sheet3"

data <- read_excel(path, sheet = "Data")
```

Excel

- The header is not row one

Excel — In R

```r
data <- read_excel(path, sheet = "Data", skip = 3)
```

Excel

- Merged cells

Excel — In R

```
data <- data |> tidyr::fill(district, .direction = "down")
```

Excel

- Serial dates

Excel — In R

```
data$screening_date <- as.Date(as.numeric(data$screening_date), origin = "1899-12-30")
```

Excel

- Types in Excel

Excel — In R

```
data <- read_excel(
  path,
  sheet = "Data",
  col_types = c("text", "text", "date", "numeric", "text", "numeric", "text", "text")
)
```

The read function, assembled — In R (cont.)

```r
# R/read_register.R
read_register <- function(path) {
  muac <- readr::read_csv(
    path,
    col_types = readr::cols(
      child_id       = readr::col_character(),
      commune        = readr::col_character(),
      screening_date = readr::col_date(format = "%Y-%m-%d"),
      age_months     = readr::col_integer(),
      sex            = readr::col_character(),
      muac_mm        = readr::col_integer(),
      oedema         = readr::col_character(),
      outcome        = readr::col_character()
    ),
    na = c("", "NA", "-99")
  )
```

The read function, assembled — In R (cont.)

```
  if (nrow(readr::problems(muac)) > 0) {
    print(readr::problems(muac))
    stop("Input did not match its column specification.")
  }
  stopifnot(!any(is.na(muac$child_id)))

  muac
}
```

What comes next

- The CSV is in and its types are right.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/r-for-programme-data/r-02-reading-exports