Reading an export without corrupting it

Type inference, leading zeros, dates, and the missing-value code that becomes a number. The damage happens at import, before you have looked at anything.

Pierre Robentz Cassion

Lesson 3 of 8

Data Analysis Foundations for M&E

What this lesson covers

- The damage happens before you look
- Missing-value codes become numbers
- Identifiers are not numbers
- Dates
- Categories with a fixed vocabulary
- A defensive read, end to end
- Reading the other formats
- What comes next

The damage happens before you look

- `read_csv(path)` is one line and it makes half a dozen decisions on your behalf.

Missing-value codes become numbers — In Python

```
import pandas as pd

naive = pd.read_csv("data/raw/muac-screening-artibonite-2024.v1.csv")
print(naive["muac_mm"].mean())
```

Missing-value codes become numbers — In Python

```
muac = pd.read_csv(
    "data/raw/muac-screening-artibonite-2024.v1.csv",
    na_values={"muac_mm": ["-99"]},
)

print(muac["muac_mm"].mean())
print(muac["muac_mm"].isna().sum())
```

Missing-value codes become numbers — In R

```
library(readr)

muac <- read_csv(
  "data/raw/muac-screening-artibonite-2024.v1.csv",
  na = c("", "NA", "-99")
)

mean(muac$muac_mm, na.rm = TRUE)
sum(is.na(muac$muac_mm))
```

Missing-value codes become numbers

Handling a sentinel is not the same as deciding what to do about the missing value. That decision is lesson 6. Here you are only making sure the code stops pretending to be a measurement.

Identifiers are not numbers — In Python

```
muac = pd.read_csv(
    path,
    dtype={"child_id": "string", "commune": "string"},
    na_values={"muac_mm": ["-99"]},
)
```

Identifiers are not numbers — In R

```
muac <- read_csv(
  path,
  col_types = cols(
    child_id = col_character(),
    commune  = col_character()
  ),
  na = c("", "NA", "-99")
)
```

Dates — In Python

```
muac["screening_date"] = pd.to_datetime(
    muac["screening_date"], format="%Y-%m-%d", errors="raise"
)
```

Dates — In R

```
muac <- muac |>
  dplyr::mutate(
    screening_date = as.Date(screening_date, format = "%Y-%m-%d")
  )

stopifnot(!any(is.na(muac$screening_date)))
```

Dates

- **State the format explicitly** rather than letting the parser infer it. An inferred format can change between files as...
- **Use** `errors="raise"`**.** The alternative, `errors="coerce"`, converts every unparseable date to missing — which means...

Categories with a fixed vocabulary — In Python

```
outcomes = pd.CategoricalDtype(
    ["no-action", "referred-tsfp", "referred-otp", "referred-sc"], ordered=False
)

muac["outcome"] = muac["outcome"].astype(outcomes)

# Anything outside the vocabulary is now NaN -- count it before moving on.
print(muac["outcome"].isna().sum())
```

Categories with a fixed vocabulary — In R

```
muac <- muac |>
  dplyr::mutate(
    outcome = factor(
      outcome,
      levels = c("no-action", "referred-tsfp", "referred-otp", "referred-sc")
    )
  )

sum(is.na(muac$outcome))
```

A defensive read, end to end — In Python (cont.)

```python
import pandas as pd

PATH = "data/raw/muac-screening-artibonite-2024.v1.csv"

muac = pd.read_csv(
    PATH,
    dtype={"child_id": "string", "commune": "string", "sex": "string"},
    na_values={"muac_mm": ["-99"]},
    keep_default_na=True,
)

muac["screening_date"] = pd.to_datetime(
    muac["screening_date"], format="%Y-%m-%d", errors="raise"
)

assert len(muac) == 4218, f"expected 4218 rows, got {len(muac)}"
```

A defensive read, end to end — In Python (cont.)

```
assert muac["child_id"].notna().all(), "every row needs an identifier"

print(muac.dtypes)
print(muac.isna().sum())
```

A defensive read, end to end — In R (cont.)

```r
library(readr)
library(dplyr)

PATH <- "data/raw/muac-screening-artibonite-2024.v1.csv"

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()
```

A defensive read, end to end — In R (cont.)

```
  ),
  na = c("", "NA", "-99")
)

stopifnot(nrow(muac) == 4218)
stopifnot(!any(is.na(muac$child_id)))

glimpse(muac)
colSums(is.na(muac))
```

Reading the other formats

Source	Python	R
CSV	`pd.read_csv`	`readr::read_csv`
Excel	`pd.read_excel`	`readxl::read_excel`
Stata	`pd.read_stata`	`haven::read_dta`
SPSS	`pd.read_spss`	`haven::read_sav`
Parquet	`pd.read_parquet`	`arrow::read_parquet`

What comes next

- The register is now in memory with its types intact.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/data-analysis-foundations/foundations-03-reading-exports