

Une seule table, assemblée à dessein

Partir d'une ossature des unités sur lesquelles vous devez rapporter, attacher une mesure à la fois, affirmer à chaque raccord, et porter des colonnes de provenance pour qu'un relecteur puisse remonter tout chiffre jusqu'à son fichier d'origine.

Pierre Robentz Cassion

Leçon 8 sur 8

Jointures et restructuration des données de programme — La table que vous analysez

Ce que couvre cette leçon

- La table dont tout le reste se calcule
- Décidez l'unité d'analyse avant d'écrire une ligne
- Construisez d'abord l'ossature
- Attachez une mesure à la fois, et affirmez à chaque raccord
- Portez la provenance
- Affirmez ce qui doit être vrai de la table finie
- Ce qui n'a pas sa place dans la table
- Enregistrez-la avec sa granularité dans le nom
- Ce que ce cours vous laisse

La table dont tout le reste se calcule

- À l'issue d'un assemblage, vous voulez exactement une table : une ligne par unité d'analyse, une colonne par mesure, et rien d'autre.

Décidez l'unité d'analyse avant d'écrire une ligne — En Python

```
# unit of analysis: one school x month  
# rows: 24 schools x 3 months = 72
```

Décidez l'unité d'analyse avant d'écrire une ligne — En R

```
# unit of analysis: one school x month  
# rows: 24 schools x 3 months = 72
```

Construisez d'abord l'ossature — En Python

```
schools = roster["school_id"].drop_duplicates()
months = pd.Series(["2024-02", "2024-03", "2024-04"], name="month")

analysis = (
    pd.MultiIndex.from_product([schools, months], names=["school_id", "month"])
    .to_frame(index=False)
)
print(len(analysis), "rows in the spine")
```

Construisez d'abord l'ossature — En R

```
analysis <- tidyr::expand_grid(  
  school_id = unique(roster$school_id),  
  month = c("2024-02", "2024-03", "2024-04")  
)
```

Construisez d'abord l'ossature

- L'ossature d'abord est l'habitude la plus utile de cette leçon — Un couple école-mois sans données existe désormais. . .

Attachez une mesure à la fois, et affirmez à chaque raccord — En Python

```
def attach(base, addition, on, name):
    before = len(base)
    out = base.merge(addition, on=on, how="left", validate="one_to_one")
    if len(out) != before:
        raise ValueError(f"{name}: row count changed {before} -> {len(out)}")
    filled = out[addition.columns.difference(on)].notna().any(axis=1).mean()
    print(f"{name}: attached, {filled:.1%} of spine rows matched")
    return out

analysis = attach(analysis, attendance_by_school_month, ["school_id", "month"], "attendance")
analysis = attach(analysis, enrolment_by_school, ["school_id"], "enrolment")
analysis = attach(analysis, feeding_by_school, ["school_id"], "feeding programme")
```

Attachez une mesure à la fois, et affirmez à chaque raccord — En R

```
attach_measure <- function(base, addition, by, name) {  
  before <- nrow(base)  
  out <- dplyr::left_join(base, addition, by = by, relationship = "one-to-one")  
  if (nrow(out) != before) {  
    stop(sprintf("%s: row count changed %d -> %d", name, before, nrow(out)))  
  }  
  message(sprintf("%s: attached", name))  
  out  
}  
  
analysis <- analysis |>  
  attach_measure(attendance_by_school_month, c("school_id", "month"), "attendance") |>  
  attach_measure(enrolment_by_school, "school_id", "enrolment")
```

Portez la provenance — En Python

```
analysis["source_register"] = "school-attendance-2024.v1.csv"  
analysis["denominator_source"] = "roster 2024, enrolled students"  
analysis["marks_definition"] = "true/false only; Y/N excluded"  
analysis["extracted_on"] = "2026-07-27"
```

```
analysis <- analysis |>
  mutate(
    source_register      = "school-attendance-2024.v1.csv",
    denominator_source  = "roster 2024, enrolled students",
    marks_definition    = "true/false only; Y/N excluded",
    extracted_on        = "2026-07-27"
  )
```

Affirmez ce qui doit être vrai de la table finie — En Python

```
assert analysis.duplicated(subset=["school_id", "month"]).sum() == 0
assert analysis["attendance_rate"].between(0, 1).all()
assert (analysis["days_present"] <= analysis["days_marked"]).all()
assert len(analysis) == 24 * 3

print(f"analysis table: {len(analysis)} rows, "
      f"{analysis['attendance_rate'].isna().sum()} without an attendance rate")
```

Affirmez ce qui doit être vrai de la table finie — En R

```
stopifnot(  
  !any(duplicated(analysis[c("school_id", "month")])),  
  all(dplyr::between(analysis$attendance_rate, 0, 1), na.rm = TRUE),  
  all(analysis$days_present <= analysis$days_marked, na.rm = TRUE),  
  nrow(analysis) == 24 * 3  
)
```

Ce qui n'a pas sa place dans la table

- **Les colonnes intermédiaires.** `present_x`, `_merge`, `key_clean` et les six colonnes nécessaires à un seul calcul...
- **Les lignes sous votre seuil de rapportage.** Si vous supprimeriez une cellule de quatre cas dans un tableau publié,...
- **Tout ce qui est au niveau de la personne.** Une table d'analyse à la granularité école-mois n'a rien à faire d'un...

Enregistrez-la avec sa granularité dans le nom — En Python

```
analysis.to_csv("outputs/tables/attendance_by_school_month.csv", index=False)
```

Enregistrez-la avec sa granularité dans le nom — En R

```
readr::write_csv(analysis, here::here("outputs", "tables", "attendance_by_school_month.csv"))
```

Enregistrez-la avec sa granularité dans le nom — Exemple

scripts/

01_read_and_validate.R	contract and validation	(cleaning course)
------------------------	-------------------------	-------------------

02_clean.R	rules and cleaning log	(cleaning course)
------------	------------------------	-------------------

03_assemble.R	spine, joins, assertions	(this course)
---------------	--------------------------	---------------

04_report.R	tables and figures	
-------------	--------------------	--

Ce que ce cours vous laisse

- Vous savez prendre plusieurs fichiers décrivant le même programme et produire une table unique que vous défendriez colonne par colonne — jointures prouvées, granularité énoncée, dénominateur sourcé, calendrier complet, et l'écart avec ce qui a été remonté consigné plutôt que discuté.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/joining-and-reshaping/
joining-08-the-analysis-table](https://data-analysis.cassion.dev/fr/cours/joining-and-reshaping/joining-08-the-analysis-table)