

Le journal qui répond à la question avant qu'elle soit posée

Une ligne par règle appliquée — laquelle, combien de lignes touchées, ce qui a changé, qui a validé. Produit par le code plutôt que rédigé après coup, et livré comme l'annexe qui met fin au débat.

Pierre Robentz Cassion

Leçon 8 sur 8

Nettoyage et validation des données — La trace qui vous survit

Ce que couvre cette leçon

- La question qui vous sera posée
- Un journal est un tableau, pas un récit
- Les sept colonnes, et ce qui justifie chacune
- Produisez-le à partir du code
- Le tableau avant-après est le livrable
- Ce qui ne doit pas figurer dans le journal
- Où vit le journal
- L'annexe dont le rapport a besoin
- Ce que ce cours vous laisse

La question qui vous sera posée

- Lors d'une visite de suivi d'un bailleur — *« Votre charge de cas a baissé de 4 % entre le projet et le rapport final... »
- Par le responsable de programme — *« Le trimestre dernier Gros-Morne était deuxième plus touchée. Elle est maintenant... »
- Par votre successeur, dans un courriel — « *Il y a ici un script qui supprime des lignes. Savez-vous pourquoi ?* »

Un journal est un tableau, pas un récit

- Une ligne par règle appliquée. Chaque ligne porte un compte

Un journal est un tableau, pas un récit

règle	gravité	lignes	action	effet sur l'indicateur	décidé par	date
muac-range	error	7	exclues	MAG 8,62 % à 8,63 %	R. Cassion	2026-07-14
exact-duplicate	error	12	un exemplaire gardé	charge 366 à 360	R. Cassion	2026-07-14
re-registration	warning	6	fusionnées après relecture	charge 360 à 354	M. Joseph, superviseur	2026-07-16
oedema-yn-coding	warning	25	recodées en false	MAG inchangée	R. Cassion	2026-07-14
age-missing	note	226	gardées, signalées	tableau par âge seulement	R. Cassion	2026-07-14
district-mapping	warning	75	rattachées à HT07	Nord-Ouest 14,7 % à 15,0 %	A. Pierre, bureau terrain	2026-07-15

Les sept colonnes, et ce qui justifie chacune

- **règle** — l'identifiant de la table de règles de la leçon 5, pour que le journal et le code emploient le même nom. Si...
- **gravité** — ce que le contrôle signifiait, pour qu'un lecteur puisse survoler jusqu'aux erreurs.
- **lignes** — le compte. Ni « quelques-unes », ni « des valeurs aberrantes ». Un nombre.
- **action** — exclues, corrigées, recodées, fusionnées, gardées et signalées. Cinq verbes couvrent presque...
- **effet sur l'indicateur** — l'avant et l'après. C'est la colonne que l'on saute et la seule qu'un bailleur lise.
- **décidé par** — une personne. Pas « l'équipe ». La règle sans nom en face est celle qui se révèle fausse.

Les sept colonnes, et ce qui justifie chacune (suite)

- **date** — quand, pour que le journal se lise en regard de la version du résultat qu'il a produite.

Produisez-le à partir du code — En Python (suite)

```
LOG = []
```

```
def apply_step(df, rule, mask, action, note=""):
    before = indicator(df)
    if action == "exclude":
        out = df[~mask]
    elif action == "flag":
        out = df.assign(**{f"flag_{rule}": mask})
    else:
        raise ValueError(action)
    after = indicator(out)

    LOG.append({
        "rule": rule,
        "rows": int(mask.sum()),
```

Produisez-le à partir du code — En Python (suite)

```
    "action": action,  
    "before": before,  
    "after": after,  
    "effect": round(after - before, 4),  
    "note": note,  
})  
return out
```

```
muac = apply_step(muac, "muac-range", ~muac["muac_mm"].between(80, 220), "exclude")  
muac = apply_step(muac, "age-missing", muac["age_months"].isna(), "flag",  
                  "kept; age-disaggregated tables only")
```

```
log = pd.DataFrame(LOG)  
log.to_csv("outputs/cleaning-log.csv", index=False)
```

Produisez-le à partir du code — En R (suite)

```
LOG <- list()

apply_step <- function(df, rule, mask, action, note = "") {
  before <- indicator(df)
  out <- switch(action,
    exclude = df[!mask, ],
    flag     = dplyr::mutate(df, "flag_{rule}" := mask),
    stop("unknown action: ", action)
  )
  after <- indicator(out)

  LOG[[length(LOG) + 1]] <- tibble::tibble(
    rule = rule, rows = sum(mask, na.rm = TRUE), action = action,
    before = before, after = after, effect = round(after - before, 4), note = note
  )
  out
```

Produisez-le à partir du code — En R (suite)

```
}

muac <- apply_step(muac, "muac-range", !dplyr::between(muac$muac_mm, 80, 220), "exclude")
muac <- apply_step(muac, "age-missing", is.na(muac$age_months), "flag",
                  "kept; age-disaggregated tables only")

readr::write_csv(dplyr::bind_rows(LOG), here::here("outputs", "cleaning-log.csv"))
```

Produisez-le à partir du code

- Une étape ne peut pas avoir lieu sans être consignée — puisque consigner est ce que fait la fonction
- L'effet est mesuré, pas estimé — before et after proviennent de la même fonction `indicator()`, évaluée sur les...

Le tableau avant-après est le livrable — En Python

```
print(f"Records received:  {len(raw):>6,}")  
print(f"Excluded:          {len(raw) - len(analysis):>6,}")  
print(f"Analysed:          {len(analysis):>6,}")  
print(f"GAM, raw:           {indicator(raw):>6.1%}")  
print(f"GAM, cleaned:       {indicator(analysis):>6.1%}")
```

Le tableau avant-après est le livrable — En R

```
cat(sprintf("Records received: %6d\n", nrow(raw)))  
cat(sprintf("Excluded:           %6d\n", nrow(raw) - nrow(analysis)))  
cat(sprintf("Analysed:           %6d\n", nrow(analysis)))  
cat(sprintf("GAM, raw:           %6.1f%%\n", 100 * indicator(raw)))  
cat(sprintf("GAM, cleaned:       %6.1f%%\n", 100 * indicator(analysis)))
```

Ce qui ne doit pas figurer dans le journal

- **Aucun identifiant direct.** Ni noms, ni numéros de téléphone, ni coordonnées GPS. Une ligne disant 'fusion de CH03315. . .
- **Aucun petit effectif dans une catégorie sensible.** « 3 lignes recodées, catégorie VBG viol, Marmelade » identifie. . .
- **Aucun détail en texte libre recopié d'une note de dossier.** La règle est ce que vous avez appliqué, pas ce que. . .

Où vit le journal — Exemple

outputs/

profiles/muac-2024-q4.json	arrival profile	(lesson 1)
validation/muac-2024-q4.json	validation findings	(lesson 7)
cleaning-log.csv	decisions	(this lesson)
review/duplicate-candidates.csv	reviewed and signed	(lesson 4)
tables/gam_by_commune.csv	the result	
tables/definitions.csv	numerator, denominator, source	

L'annexe dont le rapport a besoin

- Source et complétude des données — D'où vient le fichier, combien d'enregistrements, quelle part des unités de...
- Nettoyage et exclusions — Le tableau du journal, ou son résumé avec le tableau complet en pièce jointe
- Limites — Ce que le nettoyage n'a pas pu corriger

L'annexe dont le rapport a besoin

L'annexe n'est pas un aveu. C'est la raison pour laquelle le chiffre du résumé exécutif peut être défendu, et la différence professionnelle entre un chiffre et une affirmation.

Ce que ce cours vous laisse

- Vous savez prendre un export brut de programme, décrire honnêtement ce qui est arrivé, y trouver les défauts de quatre manières différentes, décider du traitement de chacun au regard d'une norme publiée, faire tourner ces décisions à nouveau le trimestre suivant sans vous, et remettre l'ensemble à un auditeur avec la démarche jointe.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/data-cleaning-and-validation/
cleaning-08-the-cleaning-log](https://data-analysis.cassion.dev/fr/cours/data-cleaning-and-validation/cleaning-08-the-cleaning-log)