

Une chaîne qui échoue vaut mieux qu'une chaîne qui devine

L'export amont gagne une colonne, en perd une, change un code de « yes » à « Y », ou arrive avec la moitié des lignes. Une chaîne qui continue produit un nombre faux et plausible. Sept vérifications l'arrêtent, et chacune a une défaillance réelle derrière elle.

Pierre Robentz Cassion

Leçon 6 sur 8

Chaînes d'analyse reproductibles — Un modèle, plusieurs sorties

Ce que couvre cette leçon

- La défaillance qui n'a pas de message d'erreur
- Sept vérifications, chacune avec une défaillance réelle derrière
- Échouez au point de défaillance, non à la fin
- Ce que fait cette plateforme
- La règle à retenir
- Rapportez-le en entier
- La suite

La défaillance qui n'a pas de message d'erreur

- Rien n'a échoué — Le graphique est tracé, le pourcentage est plausible, le fichier est daté d'aujourd'hui, et le nombre. . .
- C'est le mode de défaillance pour lequel cette leçon existe — et il est bien plus courant qu'un plantage

Sept vérifications, chacune avec une défaillance réelle derrière

- Une : les colonnes attendues sont présentes

Sept vérifications, chacune avec une défaillance réelle derrière — En Python

```
REQUIRED = {"household_id", "district", "water_source", "round_trip_minutes"}  
missing = REQUIRED - set(survey.columns)  
if missing:  
    raise ValueError(f"Export is missing columns: {sorted(missing)}")
```

Sept vérifications, chacune avec une défaillance réelle derrière — En R

```
stopifnot(all(required %in% names(survey)))
```

Sept vérifications, chacune avec une défaillance réelle derrière

- Deux : le nombre de lignes est dans la plage attendue

Sept vérifications, chacune avec une défaillance réelle derrière — En Python

```
if not 2_000 <= len(survey) <= 3_000:  
    raise ValueError(f"Expected 2,000-3,000 households, got {len(survey):,}")
```

Sept vérifications, chacune avec une défaillance réelle derrière

- La moitié d'un export est la défaillance silencieuse la plus courante — Un téléchargement tronqué, un filtre resté. . .
- Trois : les codes sont les codes que vous connaissez

Sept vérifications, chacune avec une défaillance réelle derrière — En Python

```
KNOWN = {"piped-into-dwelling", "piped-into-yard", "public-tap", "borehole",  
         "protected-well", "protected-spring", "unprotected-well",  
         "unprotected-spring", "surface-water", "tanker-truck", "rainwater"}  
unknown = set(survey["water_source"].dropna()) - KNOWN  
if unknown:  
    raise ValueError(f"Unknown water_source values: {sorted(unknown)}")
```

Sept vérifications, chacune avec une défaillance réelle derrière — En R

```
setdiff(unique(survey$water_source), known)
```

Sept vérifications, chacune avec une défaillance réelle derrière

- Un nouveau code est une décision, non une donnée — Quelqu'un a ajouté une option au formulaire et l'analyse doit. . .
- Quatre : l'identifiant est unique là où il doit l'être

Sept vérifications, chacune avec une défaillance réelle derrière — En Python

```
duplicates = survey["household_id"].duplicated().sum()
if duplicates:
    raise ValueError(f"{duplicates} duplicate household_id values")
```

Sept vérifications, chacune avec une défaillance réelle derrière

- Le registre scolaire de cette plateforme a exactement ce défaut — deux élèves y figurent deux fois après un transfert. . .
- Cinq : les valeurs manquantes sont là où vous les attendez

Sept vérifications, chacune avec une défaillance réelle derrière — En Python

```
completeness = survey.notna().mean()
if completeness["district"] < 0.99:
    raise ValueError(f"district is {completeness['district']:.1%} complete")
```

Sept vérifications, chacune avec une défaillance réelle derrière

- Six : les nombres sont dans une plage possible

Sept vérifications, chacune avec une défaillance réelle derrière — En Python

```
implausible = survey[(survey["litres_per_person_day"] < 0)
                      | (survey["litres_per_person_day"] > 200)]
if len(implausible) > 20:
    raise ValueError(f"{len(implausible)} implausible litres values")
```

Sept vérifications, chacune avec une défaillance réelle derrière

- Notez le seuil plutôt qu'une tolérance nulle — Onze ménages avec une erreur d'unité est un défaut documenté que cette. . .
- Sept : la sortie est celle que vous avez déclarée

Sept vérifications, chacune avec une défaillance réelle derrière — En Python

```
assert summary["n"].sum() == len(survey), "rows lost between input and summary"
```

Sept vérifications, chacune avec une défaillance réelle derrière

- Un nombre de lignes qui change au travers d'une jointure est l'assertion la plus utile de l'analyse de programme —...

Échouez au point de défaillance, non à la fin — En Python

```
def load_survey(path: pathlib.Path) -> pd.DataFrame:
    survey = pd.read_csv(path)
    check_columns(survey)
    check_rows(survey)
    check_codes(survey)
    return survey                # nothing downstream runs on a bad file
```

Échouez au point de défaillance, non à la fin — En R

```
load_survey <- function(path) {  
  survey <- readr::read_csv(path)  
  check_columns(survey); check_rows(survey); check_codes(survey)  
  survey  
}
```

Échouez au point de défaillance, non à la fin

- Vérifiez à la frontière — là où les données entrent, et après chaque jointure — Une vérification en fin de chaîne vous. . .
- Levez une erreur, n'avertissez pas — Un avertissement dans un journal que personne ne lit équivaut à aucune. . .

Ce que fait cette plateforme

Vérification	Attrape
Schéma Zod	Un champ manquant ou du mauvais type
Références inter-collections	Un chemin pointant vers un cours qui n'existe pas
Fichiers hors d'un répertoire de langue	Une entrée sans langue
Parité de traduction	Une entrée publiée dans une seule langue
Cohérence sujet-secteur	Un sujet étiqueté hors de son secteur
Synthétique uniquement	Un jeu de données non déclaré synthétique
Ossature du programme	Un cours publié absent de la carte du cursus

Ce que fait cette plateforme

- Quatre autres s'exécutent dans `pnpm test` plutôt que dans la compilation — parce qu'elles ont besoin de `node:fs` et...
- L'une d'elles vérifie une relation que le graphe de références ne peut structurellement pas atteindre — Les règles de...
- Toute règle du type « chaque X a au moins un Y » exige un test de cette forme — Un graphe de références est le mauvais...

La règle à retenir

- Tout champ nommant un fichier livré exige un drapeau `ready` et un test de système de fichiers à côté — Cette. . .
- Un chemin dans un frontmatter est une chaîne de caractères, et rien dans une compilation ne peut dire s'il pointe vers. . .

Rapportez-le en entier — Exemple

Pipeline checks

The loader validates every raw export before anything downstream runs: required columns present, 2,000-3,000 rows, water_source values within the known set, household_id unique, district at least 99% complete.

Implausible litres-per-person values are tolerated up to 20 rows, which is the documented unit-entry defect; above that the run stops.

Row counts are asserted across every join. A join that changes the row count stops the pipeline rather than producing a summary.

All checks raise rather than warn. The March run stopped on an unknown water_source value ("piped-shared"), which turned out to be a new form option added upstream; it is now mapped explicitly in src/clean.py.

- La dernière phrase est ce qui rend la section crédible — Une section de vérifications qui n'a jamais rien attrapé est. . .

- Tout ce qui précède suppose que vous êtes encore là.

Lire la leçon complète, avec le code exécutable

```
https://data-analysis.cassion.dev/fr/cours/reproducible-workflows/  
repro-06-fail-loudly
```