Faire en sorte que cela tourne encore au trimestre suivant

Transformer une analyse qui fonctionne en une analyse qui se relance sur un nouvel export sans modification, échoue bruyamment quand l'entrée change, et se transmet à quelqu'un qui n'a rien de votre contexte.

Pierre Robentz Cassion

Leçon 8 sur 8

Fondamentaux de l'analyse de données pour le suivi-évaluation

Ce que couvre cette leçon

- Le test
- Découpez l'analyse en étapes qui se passent des fichiers
- Rassemblez les paramètres en un seul endroit
- Faites échouer la chaîne bruyamment
- Paramétrez au lieu de copier
- Ce qui ne va jamais dans le dépôt
- Le README de passation
- Pour aller plus loin

Le test

- L'export du trimestre suivant arrive.

Découpez l'analyse en étapes qui se passent des fichiers — Exemple

```
scripts/
  01-lecture.py       export brut   -> data/interim/muac-type.parquet
  02-nettoyage.py     typé          -> data/processed/muac-propre.parquet
                                       output/tables/journal-nettoyage.csv
  03-indicateurs.py   propre        -> output/tables/mag-par-commune.csv
```

Découpez l'analyse en étapes qui se passent des fichiers — En Python

```
muac.to_parquet("data/interim/muac-type.parquet", index=False)
```

Découpez l'analyse en étapes qui se passent des fichiers — En R

```
arrow::write_parquet(muac, "data/interim/muac-type.parquet")
```

Rassemblez les paramètres en un seul endroit — En Python

```python
# config.py
from pathlib import Path

BRUT = Path("data/raw/muac-screening-artibonite-2024.v1.csv")
INTERMEDIAIRE = Path("data/interim/muac-type.parquet")
TRAITE = Path("data/processed/muac-propre.parquet")
TABLEAUX = Path("output/tables")

MAS_MM = 115
MAG_MM = 125
PB_PLAUSIBLE = (80, 220)
AGE_MOIS = (6, 59)
CODE_MANQUANT = "-99"
```

Rassemblez les paramètres en un seul endroit — En R

```r
# config.R
BRUT          <- "data/raw/muac-screening-artibonite-2024.v1.csv"
INTERMEDIAIRE <- "data/interim/muac-type.parquet"
TRAITE        <- "data/processed/muac-propre.parquet"
TABLEAUX      <- "output/tables"

MAS_MM        <- 115
MAG_MM        <- 125
PB_PLAUSIBLE  <- c(80, 220)
AGE_MOIS      <- c(6, 59)
CODE_MANQUANT <- "-99"
```

Faites échouer la chaîne bruyamment — En Python (suite)

```python
def controler_entree(df):
    attendues = {
        "child_id", "commune", "screening_date", "age_months",
        "sex", "muac_mm", "oedema", "outcome",
    }
    absentes = attendues - set(df.columns)
    if absentes:
        raise ValueError(f"colonnes absentes de l'export : {sorted(absentes)}")

    if df["child_id"].isna().any():
        raise ValueError("lignes sans identifiant")

    mesures = df["muac_mm"].dropna()
    if len(mesures) == 0:
        raise ValueError("aucune mesure de PB -- mauvais fichier ?")
```

Faites échouer la chaîne bruyamment — En Python (suite)

```
part_manquante = df["muac_mm"].isna().mean()
if part_manquante > 0.20:
    raise ValueError(
        f"{part_manquante:.1%} de PB manquants -- au-dessus de la tolérance de 20 %. "
        "À investiguer avant tout rapportage."
    )
```

Faites échouer la chaîne bruyamment — En R (suite)

```r
controler_entree <- function(df) {
  attendues <- c("child_id", "commune", "screening_date", "age_months",
                 "sex", "muac_mm", "oedema", "outcome")
  absentes <- setdiff(attendues, names(df))
  if (length(absentes) > 0) {
    stop("colonnes absentes de l'export : ", paste(absentes, collapse = ", "))
  }

  if (any(is.na(df$child_id))) stop("lignes sans identifiant")

  part_manquante <- mean(is.na(df$muac_mm))
  if (part_manquante > 0.20) {
    stop(sprintf(
      "%.1f%% de PB manquants - au-dessus de la tolerance de 20%%. A investiguer avant tout
     rapportage.",
      part_manquante * 100
    ))
```

Faites échouer la chaîne bruyamment — En R (suite)

```
  }
}
```

Faites échouer la chaîne bruyamment

Écrivez l'assertion correspondant à la défaillance qui vous embarrasserait, pas à celle que vous jugez probable. Les probables, vous les remarquerez.

Paramétrez au lieu de copier — En Python (suite)

```
import sys
from pathlib import Path

def produire_rapport(commune: str | None = None):
    df = pd.read_parquet(TRAITE)
    if commune:
        df = df[df["commune"] == commune]
        if df.empty:
            raise ValueError(f"aucune ligne pour la commune {commune!r}")

    tableau = tableau_indicateurs(df, "tranche_age")
    nom = commune.lower().replace(" ", "-") if commune else "toutes"
    tableau.to_csv(TABLEAUX / f"mag-{nom}.csv", index=False)
    return tableau
```

Paramétrez au lieu de copier — En Python (suite)

```
if __name__ == "__main__":
    cible = sys.argv[1] if len(sys.argv) > 1 else None
    produire_rapport(cible)
```

Paramétrez au lieu de copier — En R

```r
produire_rapport <- function(commune = NULL) {
  df <- arrow::read_parquet(TRAITE)
  if (!is.null(commune)) {
    df <- filter(df, commune == !!commune)
    if (nrow(df) == 0) stop("aucune ligne pour la commune ", commune)
  }

  tableau <- tableau_indicateurs(df, "tranche_age")
  nom <- if (is.null(commune)) "toutes" else tolower(gsub(" ", "-", commune))
  readr::write_csv(tableau, file.path(TABLEAUX, paste0("mag-", nom, ".csv")))
  tableau
}

args <- commandArgs(trailingOnly = TRUE)
produire_rapport(if (length(args) > 0) args[1] else NULL)
```

Ce qui ne va jamais dans le dépôt

- **Ne versionnez jamais de données brutes de bénéficiaires.** Ni identifiées, ni pseudonymisées. Git conserve chaque…
- **Ne versionnez jamais d'identifiants de connexion.** Jetons DHIS2, clés d'API KoboToolbox, mots de passe de base de…
- **Ajoutez** `data/raw/` **au** `.gitignore` et documentez dans le README où se trouve le vrai fichier — un disque partagé,…

Ce qui ne va jamais dans le dépôt — Exemple

```
data/raw/
data/interim/
data/processed/
.env
*.Rhistory
__pycache__/
.Rproj.user/
```

Le README de passation — Exemple (suite)

```
# Analyse du dépistage PB, Artibonite

## Ce que cela produit
Taux trimestriels de MAG et de MAS par commune, sexe et tranche d'âge, avec
intervalles à 95 %. Alimente la décision d'implantation des sites PCIMA et le
rapport bailleur trimestriel.

## D'où viennent les données
Export CommCare, projet `artibonite-nutrition`, formulaire `Dépistage
communautaire`. Exporté chaque mois par l'assistant suivi-évaluation vers le
disque partagé à <chemin>. Absent de ce dépôt.

## Comment l'exécuter
    uv sync
    uv run python scripts/01-lecture.py
    uv run python scripts/02-nettoyage.py
```

Le README de passation — Exemple (suite)

```
    uv run python scripts/03-indicateurs.py

## Décisions à connaître
Voir output/tables/journal-nettoyage.csv. Celle qui compte : les lignes sans
âge sont conservées, car 60 % de la semaine du 10 au 14 juin à Gros-Morne est
sans âge et leur suppression modifie le classement des communes.

## Qui contacter
Définitions d'indicateurs : <nom>, responsable suivi-évaluation.
Problèmes de collecte : <nom>, coordinateur terrain.
```

Pour aller plus loin

- **Nettoyage et validation des données** approfondit le profilage et les corrections des leçons 5 et 6, avec...
- **Conception d'indicateurs et cadre logique** part de la fiche de référence de la leçon 7 et remonte jusqu'à la théorie...
- **Analyse d'enquêtes, échantillonnage et pondération** couvre ce que cette formation n'a pas abordé : ce qui change...

Lire la leçon complète, avec le code exécutable

https://data-analysis.cassion.dev/fr/cours/data-analysis-foundations/foundations-08-reproducing