

Décisions de nettoyage et journal de nettoyage

Trancher le traitement de chaque défaut, chiffrer ce que la décision coûte au résultat final, et la consigner pour qu'un auditeur — ou vous-même dans six mois — puisse suivre le raisonnement.

Pierre Robentz Cassion

Leçon 6 sur 8

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

Ce que couvre cette leçon

- Le nettoyage est un ensemble de décisions, pas un script
- Les quatre options
- Traitement de ce registre
- Chiffrez ce que la décision coûte
- Le journal de nettoyage
- La seule chose à ne jamais faire
- La suite

Le nettoyage est un ensemble de décisions, pas un script

- Le profilage de la leçon précédente a produit une liste de défauts.

Les quatre options

Option	Quand elle s'applique	Ce qu'elle coûte
Corriger	Vous savez quelle valeur était visée	Rien, si vous avez raison
Passer en manquant	La valeur est fausse et irrécupérable	De la précision, et peut-être du biais
Supprimer la ligne	La ligne n'est pas une observation réelle	De la charge de cas, si vous vous trompez
Conserver et signaler	La valeur peut être juste, sans moyen de trancher	Rien, mais le lecteur doit la traiter

- Erreurs d'unité — sept enregistrements en centimètres — Récupérables

Traitement de ce registre — En Python

```
import numpy as np

erreur_unite = muac["muac_mm"].notna() & (muac["muac_mm"] < 40)
n_erreur_unite = int(erreur_unite.sum())

muac.loc[erreur_unite, "muac_mm"] = muac.loc[erreur_unite, "muac_mm"] * 10
```

Traitement de ce registre — En R

```
erreur_unite <- !is.na(muac$muac_mm) & muac$muac_mm < 40  
n_erreur_unite <- sum(erreur_unite)  
  
muac <- muac |>  
  mutate(muac_mm = if_else(!is.na(muac_mm) & muac_mm < 40, muac_mm * 10L, muac_mm))
```

- Valeurs implausibles qui ne sont pas des erreurs d'unité — Irrécupérables

Traitement de ce registre — En Python

```
implausible = muac["muac_mm"].notna() & (  
    (muac["muac_mm"] < 80) | (muac["muac_mm"] > 220)  
)  
n_implausible = int(implausible.sum())  
  
muac.loc[implausible, "muac_mm"] = np.nan
```

Traitement de ce registre — En R

```
implausible <- !is.na(muac$muac_mm) & (muac$muac_mm < 80 | muac$muac_mm > 220)
n_implausible <- sum(implausible)

muac <- muac |>
  mutate(muac_mm = if_else(implausible, NA_integer_, muac_mm))
```

- Doublons exacts — douze lignes — À supprimer

Traitement de ce registre — En Python

```
avant = len(muac)
muac = muac.drop_duplicates()
n_doublons_exacts = avant - len(muac)
```

Traitement de ce registre — En R

```
avant <- nrow(muac)
muac <- distinct(muac)
n_doublons_exacts <- avant - nrow(muac)
```

- Réenregistrements sous un nouvel identifiant — six suspectés — À conserver et signaler

Traitement de ce registre — En Python

```
cle = ["commune", "screening_date", "age_months", "sex", "muac_mm"]
muac["doublon_suspecte"] = muac.duplicated(subset=cle, keep=False) & muac[
    "muac_mm"
].notna()

n_suspectes = int(muac["doublon_suspecte"].sum())
```

Traitement de ce registre — En R

```
muac <- muac |>
  group_by(commune, screening_date, age_months, sex, muac_mm) |>
  mutate(doublon_suspecte = n() > 1 & !is.na(muac_mm)) |>
  ungroup()

n_suspectes <- sum(muac$doublon_suspecte)
```

- Codage incohérent des œdèmes — Récupérable

Traitement de ce registre — En Python

```
table_oedeme = {  
    "true": True, "TRUE": True, "Y": True, "y": True, "yes": True,  
    "false": False, "FALSE": False, "N": False, "n": False, "no": False,  
}  
muac["oedema"] = muac["oedema"].astype("string").str.strip().map(table_oedeme)  
n_oedeme_manquant = int(muac["oedema"].isna().sum())
```

Traitement de ce registre — En R

```
muac <- muac |>
  mutate(
    oedema = case_when(
      tolower(trimws(oedema)) %in% c("true", "y", "yes") ~ TRUE,
      tolower(trimws(oedema)) %in% c("false", "n", "no") ~ FALSE,
      TRUE ~ NA
    )
  )

n_oedeme_manquant <- sum(is.na(muac$oedema))
```

- Âge manquant concentré sur Gros-Morne — Voilà le cas qui exige une décision plutôt qu'une règle, et il fait l'objet de. . .

Chiffrez ce que la décision coûte — En Python (suite)

```
seuil_mag = 125

complet = muac.dropna(subset=["age_months", "muac_mm"])
tous_mesures = muac.dropna(subset=["muac_mm"])

def taux_mag(df):
    return (df["muac_mm"] < seuil_mag).mean()

comparaison = pd.DataFrame(
    {
        "en_supprimant": complet.groupby("commune").apply(taux_mag),
        "en_conservant": tous_mesures.groupby("commune").apply(taux_mag),
    }
)
comparaison["ecart"] = (
    comparaison["en_supprimant"] - comparaison["en_conservant"]
)
```

Chiffrez ce que la décision coûte — En Python (suite)

```
)  
print(comparaison.sort_values("ecart", ascending=False).round(4))
```

Chiffrez ce que la décision coûte — En R

```
taux_mag <- function(df) mean(df$muac_mm < 125, na.rm = TRUE)

complet <- muac |> filter(!is.na(age_months), !is.na(muac_mm))
tous_mesures <- muac |> filter(!is.na(muac_mm))

comparaison <- full_join(
  complet |> group_by(commune) |> summarise(en_supprimant = taux_mag(pick(everything()))),
  tous_mesures |> group_by(commune) |> summarise(en_conservant = taux_mag(pick(everything()))
),
  by = "commune"
) |>
  mutate(ecart = en_supprimant - en_conservant) |>
  arrange(desc(abs(ecart)))

comparaison
```

Chiffrez ce que la décision coûte

Principe général : supprimez des lignes pour une analyse donnée, jamais du jeu de données. Une ligne sans âge reste une donnée probante sur le PB. Filtrer au point d'utilisation maintient chaque analyse sur le plus grand échantillon qu'elle admette.

Le journal de nettoyage — En Python (suite)

```
journal_nettoyage = pd.DataFrame(  
    [  
        {  
            "regle": "Erreur d'unité PB corrigée (cm vers mm)",  
            "colonne": "muac_mm",  
            "action": "corrige",  
            "lignes": n_erreur_unite,  
            "justification": "Les valeurs sous 40 sont des centimètres non convertis. "  
            "Les plages plausibles en mm et en cm sont disjointes, la correction est donc  
sans ambiguïté.",  
        },  
        {  
            "regle": "PB implausible passé en manquant",  
            "colonne": "muac_mm",  
            "action": "passe-manquant",  
            "lignes": n_implausible,  
            "justification": "Hors de 80-220 mm, ce n'est pas une mesure pour 6-59 mois. "        }  
    ]  
)
```

```
    "Ligne conservée : le dépistage a bien eu lieu.",
},
{
    "regle": "Doubles soumissions exactes supprimées",
    "colonne": "toutes",
    "action": "supprime",
    "lignes": n_doublons_exacts,
    "justification": "Des lignes identiques correspondent à un formulaire soumis deux
fois sur une mauvaise connexion.",
},
{
    "regle": "Réenregistrement suspecté signalé",
    "colonne": "doublon_suspecte",
    "action": "signale",
    "lignes": n_suspectes,
    "justification": "Mêmes commune, date, âge, sexe et PB sous des identifiants diff
érents. "
    "Indiscernable d'une coïncidence sans le registre papier.",
```

Le journal de nettoyage — En Python (suite)

```
    },
    {
        "regle": "Codage des œdèmes harmonisé",
        "colonne": "oedema",
        "action": "corrige",
        "lignes": n_oedeme_manquant,
        "justification": "Ennery et Desdunes ont utilisé Y/N au T1. Les cases vides
restent manquantes.",
    },
    {
        "regle": "Âge manquant conservé",
        "colonne": "age_months",
        "action": "conserve",
        "lignes": int(muac["age_months"].isna().sum()),
        "justification": "60 % de manquants sur la semaine du 10 au 14 juin à Gros-Morne.
"
        "Supprimer modifie le classement des communes ; les seuils de PB n'exigent pas l'
âge.",
    },
    },
```

```
    ]  
)  
  
journal_nettoyage.to_csv("output/tables/journal-nettoyage.csv", index=False)  
print(journal_nettoyage[["regle", "action", "lignes"]])
```

Le journal de nettoyage — En R

```
journal_nettoyage <- tibble::tribble(  
  ~regle,                ~colonne,                ~action,                ~  
    lignes,  
  "Erreur d'unité PB corrigée (cm vers mm)", "muac_mm",            "corrige",            n_  
    erreur_unite,  
  "PB implausible passé en manquant",        "muac_mm",            "passe-manquant",    n_  
    implausible,  
  "Doubles soumissions exactes supprimées",  "toutes",              "supprime",          n_  
    doublons_exacts,  
  "Réenregistrement suspecté signalé",        "doublon_suspecte",    "signale",           n_  
    suspectes,  
  "Codage des œdèmes harmonisé",              "oedema",              "corrige",           n_  
    oedeme_manquant,  
  "Âge manquant conservé",                    "age_months",          "conserve",          sum(is  
    .na(muac$age_months))  
)
```

```
readr::write_csv(journal_nettoyage, "output/tables/journal-nettoyage.csv")  
journal_nettoyage
```

- **Les effectifs viennent du code**, pas de la mémoire. Si la règle change, l'effectif change avec elle.
- **Il est livré avec l'analyse** — en annexe du rapport, et non comme un fichier sur le portable de quelqu'un.
- **La justification est une phrase, pas une catégorie.** « Qualité des données » n'est pas une justification.
- **Il s'écrit pendant le nettoyage**, et non reconstitué après coup. Les journaux reconstitués sont faux précisément là...

La seule chose à ne jamais faire

- Ne modifiez pas `data/raw/`.

- Les données sont propres et les décisions sont consignées.

Lire la leçon complète, avec le code exécutable

`https://data-analysis.cassion.dev/fr/cours/data-analysis-foundations/
foundations-06-cleaning-decisions`