

Codes, sentinelles et le 99 qui entre dans une moyenne

Valeurs sentinelles, vocabulaires catégoriels, booléens saisis de cinq façons et étiquettes de valeurs Stata — transformer ce que le formulaire a enregistré en ce que pandas peut calculer.

Pierre Robentz Cassion

Leçon 4 sur 8

Python pour les données de programme — Faire entrer l'export

Ce que couvre cette leçon

- Un nombre qui signifie « pas de réponse »
- Vocabulaires catégoriels
- Des booléens saisis de cinq façons
- Du texte qui devrait être une seule valeur
- Stata et SPSS transportent leurs étiquettes
- La fonction de lecture, assemblée
- Ce qui vient ensuite

Un nombre qui signifie « pas de réponse » — En Python

```
import pandas as pd

muac = pd.read_csv(RAW / "muac-screening-artibonite-2024.v1.csv")
print(muac["muac_mm"].mean())          # plusieurs millimetres sous la verite
```

Un nombre qui signifie « pas de réponse »

- Déclarer les sentinelles à la lecture

Un nombre qui signifie « pas de réponse » — En Python

```
muac = pd.read_csv(  
    RAW / "muac-screening-artibonite-2024.v1.csv",  
    dtype={"child_id": "string", "commune": "string"},  
    na_values={"muac_mm": ["-99"]},  
)  
  
print(muac["muac_mm"].mean())  
print(muac["muac_mm"].isna().sum())
```

Un nombre qui signifie « pas de réponse »

- Restreignez la sentinelle à sa colonne — `na_values=["-99"]` sans dictionnaire l'applique à toutes les colonnes du...
- La liste des sentinelles est une documentation, non une tradition orale

Un nombre qui signifie « pas de réponse » — En Python

```
SENTINELS = {  
    "muac_mm": ["-99"],          # non mesure  
    "age_months": ["99", "-1"],  # non repondu / sans objet  
}  
  
muac = pd.read_csv(path, na_values=SENTINELS)
```

Un nombre qui signifie « pas de réponse » — En Python

```
for column in ["muac_mm", "age_months"]:  
    print(column, sorted(muac[column].dropna().unique())[:5],  
          sorted(muac[column].dropna().unique())[-5:])
```

Un nombre qui signifie « pas de réponse »

Déclarer une sentinelle n'est pas décider quoi faire de la valeur manquante. Cette décision — supprimer, imputer, rapporter séparément — relève de l'analyse, et doit être consignée. Ici vous empêchez seulement un code de se faire passer pour une mesure.

```
OUTCOMES = pd.CategoricalDtype(  
    ["no-action", "referred-tsfp", "referred-otp", "referred-sc"], ordered=False  
)
```

```
muac["outcome"] = muac["outcome"].astype(OUTCOMES)  
print(muac["outcome"].isna().sum())
```

- Comptez les manquants immédiatement après la conversion — C'est là le piège : une valeur hors des catégories déclarées. . .

```
raw_values = set(muac_raw["outcome"].dropna().unique())  
declared = set(OUTCOMES.categories)  
print("inattendues :", raw_values - declared)
```

- Catégories ordonnées

Vocabulaires catégoriels — En Python

```
LADDER = pd.CategoricalDtype(  
    ["surface-water", "unimproved", "limited", "basic", "safely-managed"],  
    ordered=True,  
)  
  
wash["service"] = wash["service"].astype(LADDER)  
at_least_basic = wash["service"] >= "basic"
```

Des booléens saisis de cinq façons — En Python

```
print(muac["oedema"].value_counts(dropna=False))
```

Des booléens saisis de cinq façons — En Python

```
# Faux d'une maniere difficile a voir : toute chaine non vide est vraie,  
# donc "false" devient True.  
muac["oedema"] = muac["oedema"].astype(bool)
```

Des booléens saisis de cinq façons — En Python

```
BOOLEANS = {
    "true": True, "TRUE": True, "Y": True, "y": True, "yes": True, "1": True,
    "false": False, "FALSE": False, "N": False, "n": False, "no": False, "0": False,
}

muac["oedema"] = (
    muac["oedema"].astype("string").str.strip().str.lower()
    .map({k.lower(): v for k, v in BOOLEANS.items()})
)

print(muac["oedema"].isna().sum())    # ce que la correspondance n'a pas couvert
```

Des booléens saisis de cinq façons — En Python

```
muac["oedema"] = muac["oedema"].astype("boolean")    # True / False / <NA>
```

Du texte qui devrait être une seule valeur — En Python

```
print(wash["district"].value_counts())  
# Nord-Ouest      727  
# NORD-OUEST      33  
# Nord Ouest      22  
# nord-ouest      20
```

Du texte qui devrait être une seule valeur — En Python

```
wash["district"] = (  
    wash["district"].astype("string").str.strip().str.lower()  
    .str.replace(r"\s+", "-", regex=True)  
)  
print(wash["district"].nunique())      # 3
```

Du texte qui devrait être une seule valeur — En Python

```
CANONICAL = {  
    "gonaïves": "Gonaïves",  
    "st-marc": "Saint-Marc",  
    "saint-marc": "Saint-Marc",  
}  
wash["district"] = wash["district"].map(CANONICAL).fillna(wash["district"])
```

Stata et SPSS transportent leurs étiquettes — En Python

```
survey = pd.read_stata(path)                                # valeurs converties en etiquettes  
survey = pd.read_stata(path, convert_categoricals=False)    # codes bruts
```

Stata et SPSS transportent leurs étiquettes — En Python

```
with pd.io.stata.StataReader(path) as reader:
    labels = reader.value_labels()
    variables = reader.variable_labels()

print(variables["hh_size"])
print(labels.get("consent", {}))
```

La fonction de lecture, assemblée — En Python (suite)

```
from pathlib import Path
import pandas as pd

SENTINELS = {"muac_mm": ["-99"]}
OUTCOMES = pd.CategoricalDtype(
    ["no-action", "referred-tsfp", "referred-otp", "referred-sc"]
)
BOOLEANS = {"true": True, "y": True, "yes": True, "1": True,
            "false": False, "n": False, "no": False, "0": False}

def read_register(path: Path) -> pd.DataFrame:
    muac = pd.read_csv(
        path,
        dtype={"child_id": "string", "commune": "string", "sex": "string"},
        na_values=SENTINELS,
    )
```

La fonction de lecture, assemblée — En Python (suite)

```
muac["screening_date"] = pd.to_datetime(
    muac["screening_date"], format="%Y-%m-%d", errors="raise"
)
muac["outcome"] = muac["outcome"].astype(OUTCOMES)
muac["oedema"] = (
    muac["oedema"].astype("string").str.strip().str.lower()
    .map(BOOLEANS).astype("boolean")
)

assert muac["child_id"].notna().all(), "chaque ligne exige un identifiant"
return muac
```

Ce qui vient ensuite

- Le tableau est en mémoire et chaque colonne dit ce qu'elle signifie.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/python-for-programme-data/
python-04-codes-and-sentinels](https://data-analysis.cassion.dev/fr/cours/python-for-programme-data/python-04-codes-and-sentinels)