

Sélectionner et filtrer, et l'avertissement devenu erreur

`loc`, `iloc` et masques booléens utilisés délibérément — plus ce que signalait le `SettingWithCopyWarning`, pourquoi pandas 3 l'a remplacé par `ChainedAssignmentError`, et ce que cela change pour le code que vous avez déjà.

Pierre Robentz Cassion

Leçon 5 sur 8

Python pour les données de programme — Travailler le tableau

Ce que couvre cette leçon

- Trois façons de sélectionner, et quand chacune convient
- Masques booléens
- L'avertissement devenu erreur
- Ajouter et modifier des colonnes
- Trier n'est pas classer
- Ce qui vient ensuite

Trois façons de sélectionner, et quand chacune convient — En Python

```
import pandas as pd
```

```
muac = read_register(RAW / "muac-screening-artibonite-2024.v1.csv")
```

Trois façons de sélectionner, et quand chacune convient

- Par étiquette, avec `loc` — Les lignes par valeur d'index, les colonnes par nom

Trois façons de sélectionner, et quand chacune convient — En Python

```
muac.loc[muac["commune"] == "Gonaives", ["child_id", "muac_mm", "outcome"]]
```

Trois façons de sélectionner, et quand chacune convient

- Par position, avec `iloc` — Lignes et colonnes par position entière

Trois façons de sélectionner, et quand chacune convient — En Python

```
muac.iloc[0]           # premiere ligne  
muac.iloc[:5, :3]      # cinq premieres lignes, trois premieres colonnes
```

Trois façons de sélectionner, et quand chacune convient

- Une colonne seule — est une Series ; une liste de colonnes est un DataFrame :

Trois façons de sélectionner, et quand chacune convient — En Python

```
muac["muac_mm"]           # Series
muac[["muac_mm"]]         # DataFrame a une colonne
muac[["commune", "muac_mm"]] # DataFrame a deux colonnes
```

Masques booléens — En Python

```
severe = muac["muac_mm"] < 115
print(severe.sum())           # combien
print(severe.mean())         # quelle part
muac.loc[severe]
```

- Combiner des conditions

Masques booléens — En Python

```
# Les parenthesèses sont obligatoires : & lie plus fort que <  
under_five = (muac["age_months"] < 60) & (muac["muac_mm"] < 125)
```

```
# Utilisez & / ~ -- non and / or / not, qui ne savent pas operer element par element  
either = (muac["muac_mm"] < 115) | (muac["oedema"] == True)
```

- Les valeurs manquantes ne sont pas False

Masques booléens — En Python

```
print(len(muac))  
print((muac["muac_mm"] < 125).sum())  
print(muac["muac_mm"].isna().sum())
```

```
# 4218  
# ne compte que les enfants mesures  
# ceux exclus silencieusement
```

Masques booléens — En Python

```
measured = muac["muac_mm"].notna()
gam = measured & (muac["muac_mm"] < 125)

print(f"evalues : {measured.sum()}, cas de MAG : {gam.sum()}")
print(f"taux de MAG : {gam.sum() / measured.sum():.3f}")
```

- Le dénominateur est tout l'enjeu — Diviser par `len(muac)` au lieu de `measured.sum()` rapporte une prévalence plus...
- `isin`, `between`, `query`

Masques booléens — En Python

```
north = muac["commune"].isin(["Gonaives", "Terre-Neuve", "Anse-Rouge"])  
plausible = muac["muac_mm"].between(80, 220)
```

Masques booléens — En Python

```
impossible = muac["muac_mm"].notna() & ~muac["muac_mm"].between(80, 220)
```

Masques booléens — En Python

```
muac.query("age_months < 60 and muac_mm < 125")
```

L'avertissement devenu erreur — Exemple

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer, col_indexer] = value` instead

L'avertissement devenu erreur — En Python

```
muac[muac["commune"] == "Gonaives"]["muac_mm"] = 0
```

- Ce que fait pandas 3 à la place
- Le comportement est désormais défini — Une sélection n'écrit jamais dans son parent — non pas parfois, jamais
- L'affectation chaînée vous dit qu'elle n'a rien fait

L'avertissement devenu erreur — En Python

```
import pandas as pd

df = pd.DataFrame({"a": [1, 2, 3], "b": [10, 20, 30]})
df[df["a"] > 1]["b"] = 0
```

L'avertissement devenu erreur — Exemple

```
ChainedAssignmentError: A value is being set on a copy of a  
DataFrame or Series through chained assignment. Such chained  
assignment never works ...
```

L'avertissement devenu erreur — En Python

```
import warnings

warnings.simplefilter("error", pd.errors.ChainedAssignmentError)
```

- La forme correcte, et ce qui change pour le code existant

L'avertissement devenu erreur — En Python

```
# Un seul .loc, les deux axes, une seule operation.  
muac.loc[muac["commune"] == "Gonaïves", "muac_mm"] = pd.NA
```

- `.copy()` et les cas où vous le voulez encore

L'avertissement devenu erreur — En Python

```
# Ce sous-ensemble va etre modifie et n'est la vue de rien.  
gonaives = muac.loc[muac["commune"] == "Gonaives"].copy()  
gonaives["flag"] = gonaives["muac_mm"] < 115
```

Ajouter et modifier des colonnes — En Python

```
summary = (  
    muac  
    .loc[muac["muac_mm"].notna()]  
    .assign(  
        gam=lambda d: d["muac_mm"] < 125,  
        sam=lambda d: d["muac_mm"] < 115,  
    )  
)
```

Ajouter et modifier des colonnes — En Python

```
import numpy as np

muac["band"] = np.select(
    [muac["muac_mm"] < 115, muac["muac_mm"] < 125],
    ["severe", "moderate"],
    default="normal",
)
```

```
muac.loc[muac["muac_mm"].isna(), "band"] = pd.NA
```

Trier n'est pas classer — En Python

```
muac.sort_values(["commune", "screening_date"], ascending=[True, False])
```

Trier n'est pas classer — En Python

```
muac.groupby("commune")["muac_mm"].mean().nsmallest(5)
```

Ce qui vient ensuite

- Vous savez sélectionner un sous-ensemble et le modifier sans vous demander si la modification a eu lieu.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/python-for-programme-data/
python-05-selection-and-filtering](https://data-analysis.cassion.dev/fr/cours/python-for-programme-data/python-05-selection-and-filtering)