

Quatre choses qui changent la réponse alors que rien n'a changé

L'horloge, la graine, l'ordre des fichiers et la locale. Chacune produit un résultat différent depuis un code identique et des données identiques, chacune est invisible dans un diff, et cette plateforme en a rencontré trois sur quatre en production.

Pierre Robentz Cassion

Leçon 4 sur 8

Chaînes d'analyse reproductibles — La même réponse deux fois

Ce que couvre cette leçon

- Un : l'horloge
- Deux : la graine
- Trois : l'ordre dans lequel les fichiers reviennent
- Quatre : la locale
- Le test qui attrape les quatre
- L'habitude de la sortie déterministe
- Rapportez-le en entier
- La suite

Un : l'horloge — En Python

```
from datetime import date
report_date = date.today()           # different every day, by design
```

- Tout ce qui lit l'heure courante fait différer chaque exécution — Un pied de page « produit le », un nom de fichier. . .
- Le remède est de prendre la date des données ou d'un paramètre

Un : l'horloge — En Python

```
import pandas as pd

survey = pd.read_csv("data/raw/household-survey-2025.v1.csv")
as_of = pd.to_datetime(survey["interview_date"]).max()    # from the data
```

```
as_of <- max(as.Date(survey$interview_date))
```

- Cette plateforme s'est trompée là-dessus et l'échec est instructif — Les documents de cours et les diaporamas de leçon. . .
- La date est désormais rattachée à l'artefact — Celle d'un document vient de son propre cours et de ses leçons, celle. . .

Deux : la graine — En Python

```
import numpy as np

rng = np.random.default_rng(20260729)      # stated, committed, reproducible
sample = rng.choice(households, size=200, replace=False)
```

Deux : la graine — En R

```
set.seed(20260729)  
sample(households, 200)
```

Deux : la graine

- Tout ce qui est aléatoire exige une graine et la graine a sa place dans le code — Intervalles bootstrap,...
- Fixez-la une fois, en tête, visiblement — Une graine enfouie trois fonctions plus bas est une graine que quelqu'un...
- Et dites-le dans la sortie — L'intervalle bootstrap du cours de régression est rapporté avec « 400 rééchantillonnages,...
- Chaque jeu de données de cette plateforme est produit par un script à graine fixe — ce qui fait de 'pnpm...

Trois : l'ordre dans lequel les fichiers reviennent — En Python

```
import glob
for path in glob.glob("data/raw/*.csv"):    # order is filesystem-dependent
    ...
```

Trois : l'ordre dans lequel les fichiers reviennent

- `glob` et `os.listdir` renvoient les fichiers dans un ordre qui diffère selon les systèmes et parfois d'une exécution à...

Trois : l'ordre dans lequel les fichiers reviennent — En Python

```
for path in sorted(glob.glob("data/raw/*.csv")):  
    ...
```

Trois : l'ordre dans lequel les fichiers reviennent — En R

```
for (path in sort(list.files("data/raw", full.names = TRUE))) { }
```

Trois : l'ordre dans lequel les fichiers reviennent

- Triez. Toujours. Cela coûte six caractères
- La même chose vaut pour l'ordre des dictionnaires et des groupes — `groupby` dans `pandas` trie par défaut et. . .

Quatre : la locale — En Python

```
float("1,234")
```

```
# ValueError, or 1.234, depending on where you are
```

- Le séparateur décimal, le séparateur de milliers, le format de date et l'ordre de tri des caractères accentués. . .

Quatre : la locale — En Python

```
survey = pd.read_csv(path, decimal=".", thousands=None)      # stated, not inferred  
dates = pd.to_datetime(survey["date"], format="%Y-%m-%d")    # explicit
```

Quatre : la locale — En R

```
readr::read_csv(path, locale = locale(decimal_mark = ".", date_format = "%Y-%m-%d"))
```

Quatre : la locale

- Énoncez le format plutôt que de laisser le lecteur l'inférer — `pd.to_datetime` sans format est une fonction qui...
- Le tri est le plus subtil — `sorted()` sur des noms de communes français place Étroit après Zone sous une locale et...

Le test qui attrape les quatre — Shell

```
python run.py && cp -r outputs outputs-first  
python run.py && diff -r outputs outputs-first
```

Le test qui attrape les quatre

- Exécutez deux fois et comparez — Tout ce qui diffère est l'une des quatre, et le diff nomme le fichier
- Exécutez deux fois sur des machines différentes — et vous attrapez en plus la locale et l'ordre des fichiers, ce qu'une. . .
- Mettez-le dans l'intégration continue — qui est une machine propre à chaque fois, et la vérification s'exécute que. . .

L'habitude de la sortie déterministe

Source de bruit	Remède
Horodatages dans les métadonnées de fichier	SOURCE_DATE_EPOCH, ou les retirer
Une date de production dans un pied de page	La prendre du contenu
Une compression avec horodatage	Une archive dont les dates d'entrée sont fixées
L'ordre de sommation en virgule flottante	Trier avant de réduire là où cela compte
Un identifiant aléatoire incorporé	Le dériver d'une empreinte du contenu

L'habitude de la sortie déterministe

- Cette plateforme fixe `SOURCE_DATE_EPOCH` pour pdfTeX et pour pandoc — parce qu'un `.pptx` est une archive dont les...
- Le bénéfice est qu'un diff signifie quelque chose — Quand reconstruire ne produit aucun changement, un changement dans...

Rapportez-le en entier — Exemple

Determinism

All randomness is seeded: seed 20260729, set in `src/config.py` and reported with every bootstrap interval in this report.

Dates are taken from the data (the latest interview date) rather than from the clock. No output contains a generation timestamp.

File iteration is sorted. Group order is stated explicitly rather than inherited from the grouping library's default.

CSV reading states the decimal mark and the date format rather than inferring them.

Verified: running the pipeline twice produces byte-identical outputs, and CI runs it on a clean machine on every push.

Rapportez-le en entier

- La dernière ligne est l'affirmation et les quatre au-dessus sont la façon dont elle a été obtenue — Un rapport qui...

- Une chaîne reproductible qui produit un rapport vaut la peine d'exister.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/reproducible-workflows/
repro-04-what-makes-a-rerun-differ](https://data-analysis.cassion.dev/fr/cours/reproducible-workflows/repro-04-what-makes-a-rerun-differ)