

Un environnement de travail en Python et en R

Installer Python et R de façon à ce qu'ils fonctionnent encore sur une mauvaise connexion, organiser un dossier de projet qui survit à une passation, et exécuter le même premier script dans les deux langages.

Pierre Robentz Cassion

Leçon 2 sur 8

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

Ce que couvre cette leçon

- Pourquoi cette leçon n'est pas facultative
- Python : utilisez uv, pas l'interpréteur du système
- R : utilisez les Projets et renv
- Une organisation de projet qui survit à une passation
- Le même premier script, dans les deux langages
- Ce sur quoi les deux langages divergent
- La suite

Pourquoi cette leçon n'est pas facultative

- La raison la plus fréquente pour laquelle une analyse de suivi-évaluation cesse d'être reproductible n'est pas une erreur statistique.

Python : utilisez uv, pas l'interpréteur du système — Shell

```
# Installer uv (macOS et Linux)  
curl -LsSf https://astral.sh/uv/install.sh | sh  
  
# Créer un projet et fixer un interpréteur  
uv init muac-analysis  
cd muac-analysis  
uv python install 3.13  
  
# Ajouter ce que cette formation utilise  
uv add pandas pyarrow matplotlib
```

Python : utilisez uv, pas l'interpréteur du système — Shell

```
uv sync
```

R : utilisez les Projets et renv — En R

```
# Une fois par machine  
install.packages("renv")
```

```
# Une fois par analyse, depuis un Projet RStudio  
renv::init()
```

```
# Ajouter ce que cette formation utilise  
install.packages(c("readr", "dplyr", "tidyr", "ggplot2", "janitor"))
```

```
# Consigner exactement ce qui est installé  
renv::snapshot()
```

- **Travaillez dans un Projet.** Le répertoire de travail est alors la racine du projet, pour tout le monde, toujours...
- **N'écrivez jamais `setwd()`.** Cela inscrit votre nom d'utilisateur dans l'analyse et garantit son échec chez la...

Une organisation de projet qui survit à une passation — Exemple

```
muac-analysis/  
  data/  
    raw/          # exactement tel qu'arrivé. Lecture seule. Jamais modifié.  
    interim/      # résultats intermédiaires. Jetables.  
    processed/    # prêt à l'analyse. Régénérable depuis raw + code.  
  scripts/  
    01-lecture.py  
    02-nettoyage.py  
    03-indicateurs.py  
  output/  
    figures/  
    tables/  
  README.md
```


Une organisation de projet qui survit à une passation

Un test utile : supprimez entièrement `data/processed/` et `output/`, relancez vos scripts, et vérifiez que vous retrouvez les mêmes résultats. Si vous ne vous résolvez pas à essayer, vous connaissez déjà la réponse.

Le même premier script, dans les deux langages — En Python

```
import pandas as pd

muac = pd.read_csv("data/raw/muac-screening-artibonite-2024.v1.csv")

print(muac.shape)
print(muac.head())
print(muac.dtypes)
```

Le même premier script, dans les deux langages — En R

```
library(readr)
library(dplyr)

muac <- read_csv("data/raw/muac-screening-artibonite-2024.v1.csv")

dim(muac)
head(muac)
glimpse(muac)
```

Ce sur quoi les deux langages divergent

Aspect	pandas	dplyr / readr
Valeur manquante	NaN, et None pour les objets	NA, typé par colonne
Lecture d'un CSV	pd.read_csv devine les types	read_csv devine et le signale
Enchaînement d'étapes	Chaînage de méthodes ou réaffectation	Le tube, \>
Synthèse par groupe	groupby().agg()	group_by() \> summarise()
Catégories	Type category, sur demande	Les facteurs, et ils mordent

- La leçon suivante lit l'export correctement — ce qui est plus difficile que `read_csv(chemin)`, car les valeurs par défaut abîmeront discrètement les identifiants, les dates et le code de valeur manquante avant que vous n'ayez regardé quoi que ce soit.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/data-analysis-foundations/
foundations-02-environment](https://data-analysis.cassion.dev/fr/cours/data-analysis-foundations/foundations-02-environment)