

Un modèle, douze districts, aucune copie

Douze rapports de district produits par copier-coller sont douze documents qui divergeront d'ici mars. Un modèle paramétré rendu douze fois est un document qui ne le peut pas, et le mécanisme tient en une ligne de frontmatter.

Pierre Robentz Cassion

Leçon 5 sur 8

Chaînes d'analyse reproductibles — Un modèle, plusieurs sorties

Ce que couvre cette leçon

- Le rapport qui devient douze rapports
- Quarto, que cette plateforme emploie déjà
- Rendre les douze
- Écrire un modèle qui survit à ses propres paramètres
- Ce qui va dans le modèle et ce qui va dans la chaîne
- Le paramètre qu'il faut toujours avoir
- Là où cette plateforme fait la même chose
- Rapportez-le en entier
- La suite

Le rapport qui devient douze rapports

- D'ici mars, ces douze fichiers divergent — Quelqu'un corrige la normalisation des noms de communes dans l'un
- Un modèle paramétré est un seul document — Le corriger corrige douze rapports, et une différence n'a nulle part où se. . .

Quarto, que cette plateforme emploie déjà — Exemple

```
---  
title: "Nutrition surveillance: `r params$district`"  
params:  
  district: "Artibonite"  
  as_of: "2025-03-31"  
format: pdf  
---
```

Quarto, que cette plateforme emploie déjà — Shell

```
quarto render report.qmd -P district:Nord-Ouest -P as_of:2025-03-31
```

Quarto, que cette plateforme emploie déjà

- Le document déclare ses paramètres et le moteur de rendu les fournit — Le corps renvoie à `params$district` — en R — ou...

Quarto, que cette plateforme emploie déjà — En Python

```
# / tags: [parameters]
district = "Artibonite"
as_of = "2025-03-31"
```

Quarto, que cette plateforme emploie déjà — En R

```
params$district
```

Quarto, que cette plateforme emploie déjà

- La valeur par défaut du frontmatter est ce qui rend le modèle prévisualisable — Vous l'ouvrez, il se rend pour. . .

Rendre les douze — En Python

```
import subprocess, pathlib

DISTRICTS = ["Artibonite", "Centre", "Grande-Anse", "Nippes", "Nord",
             "Nord-Est", "Nord-Ouest", "Ouest", "Sud", "Sud-Est"]

for district in sorted(DISTRICTS):
    slug = district.lower().replace(" ", "-")
    subprocess.run([
        "quarto", "render", "report.qmd",
        "-P", f"district:{district}",
        "-P", f"as_of:{AS_OF}",
        "--output", f"outputs/reports/{slug}.pdf",
    ], check=True)
```

```
for (d in sort(districts)) {  
  quarto::quarto_render(  
    "report.qmd",  
    execute_params = list(district = d, as_of = as_of),  
    output_file = paste0(tolower(gsub(" ", "-", d)), ".pdf")  
  )  
}
```

Rendre les douze

- `check=True` est l'argument important — Sans lui, un rendu échoué est un fichier manquant que la boucle enjambe. . .
- Triez la liste des districts — pour la raison qu'a donnée la leçon précédente : l'ordre de sortie de la boucle ne. . .

Écrire un modèle qui survit à ses propres paramètres

- Aucun nombre codé en dur dans la prose — Chaque chiffre du texte vient des données

Écrire un modèle qui survit à ses propres paramètres — Exemple

Coverage in ``r params$district`` is

``r scales::percent(coverage, accuracy = 0.1)``, against a national figure of

``r scales::percent(national, accuracy = 0.1)``.

Écrire un modèle qui survit à ses propres paramètres

- Gérez le district sans données — L'un des douze aura un tableau vide, et un modèle qui suppose des lignes produit une...

Écrire un modèle qui survit à ses propres paramètres — En Python

```
if summary.empty:  
    print(f"No screening was conducted in {district} during this period.")  
else:  
    ...
```

Écrire un modèle qui survit à ses propres paramètres — En R

```
if (nrow(summary) == 0) cat("No screening was conducted in this period.")
```

Écrire un modèle qui survit à ses propres paramètres

- Gérez le singulier — « 1 communes » dans onze rapports est le signe d'un modèle que personne n'a relu
- Gérez le petit dénominateur — Un district avec 40 dépistages obtient un intervalle trois fois plus large qu'un district. . .

Ce qui va dans le modèle et ce qui va dans la chaîne

Dans le modèle	Dans la chaîne
Mise en page, prose, forme de l'argumentation	La lecture des données brutes
Un tableau de synthèse calculé depuis des données préparées	Nettoyage et normalisation
Les figures, tracées depuis des données préparées	Toute transformation de plus d'une ligne ou deux
Réserves, seuils et définitions	Tout ce que deux rapports partageraient

Ce qui va dans le modèle et ce qui va dans la chaîne

- Le modèle devrait lire un fichier préparé, non l'export brut — Si douze rendus refont chacun le nettoyage, le nettoyage. . .

Ce qui va dans le modèle et ce qui va dans la chaîne — En Python

```
# run.py
clean()                # once
build_figures()        # once
for district in DISTRICTS:  # twelve times, from prepared data
    render(district)
```

Ce qui va dans le modèle et ce qui va dans la chaîne — En R

```
# run.R, same shape
```

Le paramètre qu'il faut toujours avoir — Exemple

```
params:  
  district: "Artibonite"  
  as_of: "2025-03-31"  
  data_version: "v1"
```

Le paramètre qu'il faut toujours avoir

- Un paramètre de version de données fait dire au rapport quel fichier il a lu — ce qui est la question de provenance. . .
- Et un paramètre `as_of` plutôt que l'horloge — pour la raison qu'a donnée la leçon 4 : un rapport qui dit « produit le. . .

Là où cette plateforme fait la même chose

- `slide-plan.mjs` planifie le diaporama une fois depuis le corps de la leçon — et le PDF Beamer, le fichier PowerPoint...
- Le raisonnement est celui des douze districts, à une autre échelle — Un tableau `slides`: dans le frontmatter...
- La conséquence pour les auteurs est réelle et vaut d'être nommée — Parce que le diaporama est dérivé, les leçons sont...

Rapportez-le en entier — Exemple

District reports

Twelve district reports are rendered from one template, `report.qmd`, by `run.py`. No district-specific text exists outside the data.

Parameters: `district`, `as_of` (2025-03-31), `data_version` (v1). Each report prints all three in its colophon.

Cleaning and figure generation run once, before rendering; the template reads prepared data and performs no transformation.

Districts with no screening in the period render a stated "no data" section rather than an empty table.

Rendering is verified by `--check`: a failed render stops the run rather than leaving a missing file.

- La dernière ligne est ce qui transforme douze fichiers rendus en douze rapports qu'on peut envoyer — Une boucle qui. . .

- Une chaîne qui rend douze rapports depuis des données amont cassées rend douze rapports cassés.

Lire la leçon complète, avec le code exécutable

```
https://data-analysis.cassion.dev/fr/cours/reproducible-workflows/  
repro-05-one-template-twelve-districts
```