

The chart and the sentence come from the same file

Every rule in this course is one someone has to remember on the day. Generating the figure from the dataset removes the remembering — and it is the only way a dataset correction moves the chart and the sentence beside it together, rather than one of them.

Pierre Robentz Cassion

Lesson 8 of 8

Data Visualisation for Programmes — Charts that cannot drift

What this lesson covers

- The failure this prevents
- What this platform does
- The four properties worth copying
- Doing it in matplotlib or ggplot2
- What to commit and what to ignore
- The three-line habit for a report with no pipeline
- Report it whole
- What comes next

The failure this prevents

- Nothing in any build can detect it — The figure is a valid image, the report is a valid document, and the number in the. . .
- The fix is structural rather than procedural — If the figure is generated from the dataset by a script that runs on. . .

What this platform does — In Python

```
def food_security_instruments() -> Figure:
    survey = read("food-security-survey-2024.v1.csv")
    coping = {row["household_id"]: row for row in read("livelihood-coping-2024.v1.csv")}

    # ... compute the four prevalences from the rows ...

    return Figure(
        slug="food-security-instruments",
        kind="bar",
        title={"en": "Four instruments, one set of households", "fr": "..."},
        caption={"en": "Four food security instruments applied to the same 1,955 "
                        "households. The highest prevalence is seven times the lowest...",
                  "fr": "..."},
        x_label={"en": "Households flagged (%)", "fr": "Ménages signalés (%)"},
        bars=bars,
    )
```

What this platform does — In R

The same shape in R: a function returning data plus its labels, rendered once.

What this platform does

- No number is typed — The 7.4% in the caption is not written into the caption — the caption states the relationship and. . .
- Two outputs from one description — SVG for the web, TikZ for the PDF handout and the Beamer deck
- Standard library only — The generators need a stock Python and no install step, so a contributor can regenerate every. . .

The four properties worth copying

- One description, many renderings — The chart is data plus labels, and the renderer is separate
- Bilingual by construction — Title, caption and axis label are dictionaries keyed by locale, so a figure cannot exist in...
- Committed output — The SVG and TikZ files are in the repository, so the site builds with no Python on the deploy path...
- A test that fails when they diverge — `figures.test.ts` checks that every figure referenced by a lesson exists, that a...

The four properties worth copying — In Python

```
# The check that matters most, in one line:  
assert (OUT / f"{figure.slug}.{locale}.svg").exists()
```

The four properties worth copying — In R

A test is what turns a convention into a constraint.

Doing it in matplotlib or ggplot2 — In Python (cont.)

```
import pathlib
import pandas as pd
import matplotlib.pyplot as plt

FIGURES = pathlib.Path("outputs/figures")

def gam_by_commune() -> None:
    screening = pd.read_csv("data/muac-screening-artibonite-2024.v1.csv")
    summary = (screening.assign(case=screening["muac_mm"] < 125)
               .groupby("commune")["case"].agg(["mean", "size"])
               .sort_values("mean"))

    fig, ax = plt.subplots(figsize=(6.5, 4))
    ax.barh(summary.index, summary["mean"])
    ax.set_xlabel("GAM (%)")
    fig.savefig(FIGURES / "gam-by-commune.svg", bbox_inches="tight")
```

Doing it in matplotlib or ggplot2 — In Python (cont.)

```
plt.close(fig)

summary.to_csv(FIGURES / "gam-by-commune.csv")    # the numbers, beside the chart

if __name__ == "__main__":
    FIGURES.mkdir(parents=True, exist_ok=True)
    gam_by_commune()
```

Doing it in matplotlib or ggplot2 — In R

```
gam_by_commune <- function() {  
  summary <- screening |>  
  mutate(case = muac_mm < 125) |>  
  summarise(gam = mean(case), n = n(), .by = commune)  
  
  ggsave("outputs/figures/gam-by-commune.svg", width = 6.5, height = 4)  
  write_csv(summary, "outputs/figures/gam-by-commune.csv")  
}
```

Doing it in matplotlib or ggplot2

- Write the numbers out beside the chart — A CSV next to every figure is what lets a reviewer check the chart without. . .
- One function per figure, one script that runs them all — Not a notebook where the chart is cell 34 and depends on cell. . .
- Save as SVG, not PNG — wherever the destination allows it: it stays sharp when someone scales it, and it is text, so it. . .

What to commit and what to ignore

Commit	Ignore
The generating script	The exported PNG previews
The figure output, if the build cannot run Python	Notebook checkpoints
The numbers CSV beside each figure	Anything with a timestamp in it

What to commit and what to ignore

- Deterministic output or nothing — A figure that embeds the generation date changes on every run, and the diff becomes. . .

The three-line habit for a report with no pipeline

- One script that produces every figure in the report — run before the report is sent, in one command
- No chart made by hand in a spreadsheet — because that is the one that will not be regenerated
- The caption generated with the figure — or at least the numbers in it read from the same object the chart was drawn from

The three-line habit for a report with no pipeline — In Python

```
caption = (f"Global acute malnutrition by commune, {summary['size'].sum():,} children "  
          f"screened. {(summary['mean'] > 0.10).sum()} of {len(summary)} communes "  
          f"are above the 10% emergency threshold.")
```

The three-line habit for a report with no pipeline — In R

If the caption is an f-string, it cannot disagree with the chart.

The three-line habit for a report with no pipeline

- That single line is most of the benefit — A caption computed from the data is a caption that cannot survive the dataset. . .

Report it whole — Example

Figure production for this report

All figures are generated from the committed dataset files by scripts/figures.py, run as part of the report build. No figure is drawn by hand and no number in a caption is typed.

Each figure writes its underlying values to a CSV of the same name, so any figure can be checked without rerunning the analysis.

Figures are produced in English and French from one description, so a figure cannot exist in only one language.

Output is deterministic: rerunning the script on unchanged data produces byte-identical files, so a diff shows only real changes.

Report it whole

- The last paragraph is what makes the rest verifiable — A pipeline whose output churns on every run cannot be used to. . .

What comes next

- That is the course.

Read the full lesson, with runnable code

[https://data-analysis.cassion.dev/courses/data-visualisation/
viz-08-generate-the-figure](https://data-analysis.cassion.dev/courses/data-visualisation/viz-08-generate-the-figure)