The calendar is bigger than the programme

Median maize is 75 gourdes at the December harvest and 130 at the July peak. Compare two survey rounds taken at different points in that cycle and you have measured the calendar — and one district's panel changed composition exactly when it mattered.

Pierre Robentz Cassion

What this lesson covers

- An independent line of evidence
- Clean three things before plotting anything
- The seasonal cycle
- The panel that changed composition
- Report the series with its panel
- What comes next

An independent line of evidence — In Python

```
import pandas as pd

prices = pd.read_csv("market-prices-2024.v1.csv")
print(f"{len(prices):,} observations")
print(f"{prices['market_id'].nunique()} markets, "
      f"{prices['commodity'].nunique()} series, "
      f"{prices['period'].nunique()} months")
```

An independent line of evidence — In R

```
library(dplyr)

prices |> summarise(n = n(), markets = n_distinct(market_id),
                    series = n_distinct(commodity), months = n_distinct(period))
```

Clean three things before plotting anything — In Python

```
prices["price_htg"] = pd.to_numeric(prices["price_htg"])

dollars = prices["price_htg"] < 10
marmite = (prices["commodity"] == "maize") & (prices["market_id"] == "MK005")
thin = prices["trader_quotes"] == 1

print(f"dollar entries: {dollars.sum()}")
print(f"marmite market maize rows: {marmite.sum()}")
print(f"single-quote observations: {thin.sum()}")
```

Clean three things before plotting anything — In R

```
prices |> summarise(
  dollars = sum(price_htg < 10),
  thin = sum(trader_quotes == 1)
)
```

Clean three things before plotting anything

- Eight prices are in dollars — with the currency column still saying HTG
- One market records maize by the marmite — a volume measure of roughly 2.7 kg, and files it in a column labelled per…

Clean three things before plotting anything — In Python

```
by_market = prices[prices["commodity"] == "maize"].groupby("market_id")["price_htg"].median()
print(by_market.round(1).sort_values())
```

Clean three things before plotting anything — In R

```
prices |> filter(commodity == "maize") |>
  summarise(median = median(price_htg), .by = market_id) |> arrange(median)
```

Clean three things before plotting anything

- The unit error is invisible in one market and obvious across twelve — That is the general rule for a unit slip: it...
- Thirteen observations rest on a single trader quote — They are not errors

The seasonal cycle — In Python

```
clean = prices[~dollars & ~marmite]
maize = clean[clean["commodity"] == "maize"]
series = maize.groupby("period")["price_htg"].median()
print(series.round(0))
```

The seasonal cycle — In R

```
prices |>
  filter(commodity == "maize", price_htg >= 10, market_id != "MK005") |>
  summarise(median = median(price_htg), .by = period) |> arrange(period)
```

The seasonal cycle

Month	2023	2024
January	66	70
April	97	110
July (lean peak)	**130**	**159**
October	92	104
December (harvest)	**75**	**81**

The seasonal cycle

- Maize costs 74% more at the July peak than at the December harvest in 2023, and 96% more in 2024 — That is the size of...
- The consequence for survey design is direct — A baseline in December and an endline in July would show a catastrophic...

The seasonal cycle — In Python

```
peaks = series.loc[["2023-07", "2024-07"]]
print(f"lean peak 2023: {peaks['2023-07']:.1f}")
print(f"lean peak 2024: {peaks['2024-07']:.1f}")
print(f"year on year: {peaks['2024-07'] / peaks['2023-07'] - 1:+.1%}")
```

The seasonal cycle — In R

```
# Same point in the season, one year apart. Everything else is the calendar.
```

The seasonal cycle

- 130.1 against 159.3 — a 22% rise at the same point in the season — *That* is the deterioration, and it is a quarter of...

The panel that changed composition — In Python

```
coverage = maize.pivot_table(index="period", columns="market_id",
                             values="price_htg", aggfunc="size")
print(coverage.isna().sum(axis=1)[lambda s: s > 0])
```

The panel that changed composition — In R

```
prices |> filter(commodity == "maize") |> count(period, market_id) |>
  count(period) |> filter(n < 11)
```

The panel that changed composition

- MK001 stops reporting from June to September 2024 — when its road is cut

The panel that changed composition — In Python

```
nord = maize[maize["district"] == "Nord-Ouest"]
reported = nord.groupby("period")["price_htg"].mean()

balanced = nord[nord["market_id"] != "MK001"].groupby("period")["price_htg"].mean()
print(pd.DataFrame({"reported": reported, "balanced": balanced}).round(1)
      .loc["2024-04":"2024-10"])
```

The panel that changed composition — In R

```
prices |>
  filter(commodity == "maize", district == "Nord-Ouest") |>
  summarise(reported = mean(price_htg), .by = period)
```

The panel that changed composition

Month	Whoever reported	The two that reported every month
May 2024	147.0	123.2
June 2024	172.1	172.1
May to June	**+17%**	**+40%**

The panel that changed composition

- The reported series understates the lean-season rise by more than half — and every number in it is a correct mean of…
- Use a balanced panel — only the markets present in every month you are comparing — or chain month-on-month changes…

Report the series with its panel — Example

Maize price, gourdes per kg, median across markets

2023 harvest (Dec)	75	2024 harvest (Dec)	81
2023 lean peak (Jul)	130	2024 lean peak (Jul)	159
Seasonal swing	+74%	Seasonal swing	+96%

Year on year at the lean peak: +22%

Balanced panel of 10 markets throughout. MK001 (Nord-Ouest) did not report June to September 2024 and is excluded from all periods; MK005 records maize by the marmite and is excluded from maize.
8 dollar-denominated entries converted at 132 HTG; 13 single-quote observations retained and flagged.

What comes next

- A price says what food cost.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/food-security-ipc/fs-05-price-series-and-seasonality