

Dry is not broken

Protected wells run at 74.7% in the wet months and 36.4% in the dry ones. Handpump boreholes barely move. The status field cannot tell those two failures apart, and the sequence of visits can.

Pierre Robentz Cassion

Lesson 6 of 8

WASH Analysis — Service is a year, not a day

What this lesson covers

- One status, several different failures
- The seasonal signal, and where it is not
- Classifying each point from its own sequence
- The status value that looks like the answer and is not
- Downtime is a management statistic
- Report the shapes, not just the rate
- What comes next

One status, several different failures

Shape over the year	What it is	What to do
Down in the same months every year, up in between	Seasonal — the source runs dry	Deepen, or provide a dry-season alternative
Down once, then up after a gap	Breakdown — repaired	Look at the length of the gap
Down and stays down to the end	Abandoned or awaiting parts	Find out which; they are not the same
Up and down repeatedly	Chronic — under-maintained or overloaded	Management problem, not an asset problem

The seasonal signal, and where it is not — In Python

```
import pandas as pd

points = pd.read_csv("water-point-monitoring-2024.v1.csv", parse_dates=["visit_date"])
WORKING = {"functional", "partially-functional"}
DRY_MONTHS = {1, 2, 3, 11, 12}

points["dry_season"] = points["visit_date"].dt.month.isin(DRY_MONTHS)
points["ok"] = points["functional_status"].isin(WORKING)

print(points.groupby("dry_season")["ok"].agg(["mean", "size"]).round(3))
```

The seasonal signal, and where it is not — In R

```
library(dplyr)

points |>
  mutate(dry = lubridate::month(visit_date) %in% c(1, 2, 3, 11, 12),
         ok = functional_status %in% c("functional", "partially-functional")) |>
  summarise(functionality = mean(ok), n = n(), .by = dry)
```

The seasonal signal, and where it is not

- 67.8% in the dry months against 79.4% in the wet ones — An eleven-point swing, which a report comparing a March...

The seasonal signal, and where it is not — In Python

```
seasonal = (  
    points.groupby(["source_type", "dry_season"])["ok"].mean().unstack()  
)  
seasonal.columns = ["wet", "dry"]  
seasonal["swing"] = seasonal["wet"] - seasonal["dry"]  
print((seasonal * 100).round(1).sort_values("swing", ascending=False))
```

The seasonal signal, and where it is not — In R

```
points |>  
  summarise(functionality = mean(ok), .by = c(source_type, dry)) |>  
  tidyr::pivot_wider(names_from = dry, values_from = functionality)
```

The seasonal signal, and where it is not

Source type	Wet	Dry	Swing
Protected well	74.7%	36.4%	38.3
Protected spring	84.8%	41.0%	43.8
Handpump borehole	75.0%	76.1%	−1.1
Piped scheme tap	95.9%	98.8%	−2.9

The seasonal signal, and where it is not

- Two source types carry the entire seasonal effect and two do not move at all — Shallow wells and springs draw on a...

Classifying each point from its own sequence — In Python

```
def classify(group):
    group = group.sort_values("visit_date")
    failures = group.loc[~group["ok"]]
    if failures.empty:
        return "never failed"
    if not group["ok"].tail(3).any() and len(group) >= 3:
        return "down at year end"
    if failures["dry_season"].all():
        return "seasonal"
    return "intermittent"

shape = points.groupby("water_point_id").apply(classify, include_groups=False)
print(shape.value_counts())
```

Classifying each point from its own sequence — In R

```
points |>
  arrange(visit_date) |>
  summarise(
    shape = case_when(
      all(ok) ~ "never failed",
      !any(tail(ok, 3)) & n() >= 3 ~ "down at year end",
      all(dry[!ok]) ~ "seasonal",
      TRUE ~ "intermittent"
    ), .by = water_point_id) |>
  count(shape)
```

Classifying each point from its own sequence

Shape	Points
Never failed	82
Intermittent	80
Seasonal	46
Down at year end	34

Classifying each point from its own sequence

- The order of the tests matters and it is a judgement, not a detail — A point that fails only in dry months and is still. . .
- Write down which order you used — Two analysts with the same file and different orderings will produce different counts. . .

The status value that looks like the answer and is not — In Python

```
ever_abandoned = points.loc[points["functional_status"].eq("abandoned"),  
                             "water_point_id"].nunique()  
print(f"points ever recorded as abandoned: {ever_abandoned}")  
print(f"points down at the end of the year: 34")
```

The status value that looks like the answer and is not — In R

```
points |> filter(functional_status == "abandoned") |>  
  summarise(points = n_distinct(water_point_id))
```

The status value that looks like the answer and is not

- Derive the state from the history where you can, and use the recorded status as a cross-check — A field an enumerator. . .

Downtime is a management statistic — In Python

```
down = points.loc[~points["ok"] & points["days_since_breakdown"].notna()]
print(down.groupby("management")["days_since_breakdown"].agg(
    median="median", visits="size"
).sort_values("median"))
```

Downtime is a management statistic — In R

```
points |>  
  filter(!ok, !is.na(days_since_breakdown)) |>  
  summarise(median_days = median(days_since_breakdown), n = n(), .by = management)
```

Downtime is a management statistic

Management	Median days down at the visit
Private operator	29
Utility	32
NGO-managed	48
Community committee	68

Report the shapes, not just the rate — Example

Water point failure, 2024, 242 points

Never failed	82	33.9%
Intermittent	80	33.1%
Seasonal (dry months only)	46	19.0%
Down at year end	34	14.0%

Seasonality is concentrated in protected wells (74.7% wet, 36.4% dry) and protected springs (84.8% wet, 41.0% dry). Handpump boreholes and piped schemes show no seasonal effect.

Points classified as down at year end if not working at any of the last three visits; seasonal if every recorded failure fell in a dry month. Permanence tested before seasonality.

What comes next

- Every number in this lesson is computed over visits that were made.

Read the full lesson, with runnable code

```
https://data-analysis.cassion.dev/courses/wash-analysis/  
wash-06-seasonal-or-abandoned
```