Scientific Computing in Python Lecture 2

Functions, control flow, and threshold statistics

Zan Ahmad

Winter 2026

A three-valued classification rule

$$c(x;\theta) = \begin{cases} \text{missing}, & x \text{ absent}, \\ \text{at or below}, & x \leq \theta, \\ \text{above}, & x > \theta. \end{cases}$$

A three-valued classification rule

$$c(x;\theta) = \begin{cases} \text{missing}, & x \text{ absent}, \\ \text{at or below}, & x \leq \theta, \\ \text{above}, & x > \theta. \end{cases}$$

At $\theta = 40$, the values 40 and 40.1 belong to different classes.

A function implementing the partition

```
def classify(value, target):
    if value is None:
        return "missing"
    if value <= target:
        return "at_or_below"
    return "above"
```

Arguments, returns, and local names

```
category = classify(35, 40)
print(category)
```

Arguments, returns, and local names

```
category = classify(35, 40)
print(category)
```

The returned string is a value. The function can be called again with a different threshold without changing its definition.

Boundary cases

Reading	Threshold	Category
`None`	40	missing
40	40	at or below
40.1	40	above

Boundary cases

Reading	Threshold	Category
`None`	40	missing
40	40	at or below
40.1	40	above

Testing the boundary establishes whether the implemented inequality agrees with the definition.

Indicators and accumulated counts

$$N_{\leq}(\theta) = \sum_{i:\, x_i \text{ known}} \mathbf{1}_{\{x_i \leq \theta\}}.$$

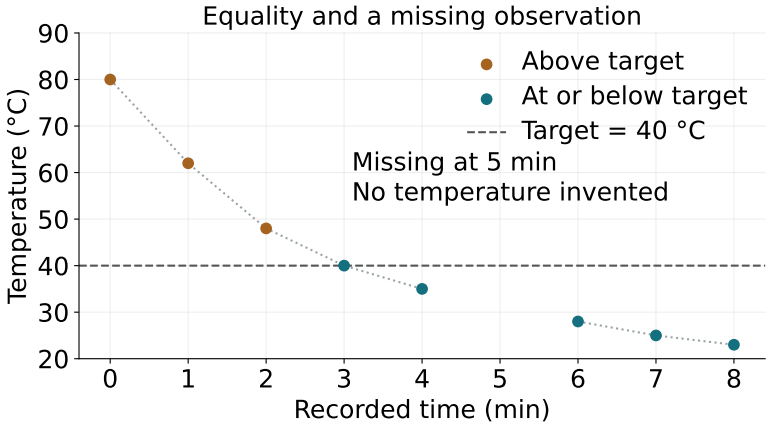
Indicators and accumulated counts

$$N_{\leq}(\theta) = \sum_{i:\, x_i \text{ known}} \mathbf{1}_{\{x_i \leq \theta\}}.$$

After k observations,

$$N_{\leq}^{(k)} + N_{>}^{(k)} + N_{\text{missing}}^{(k)} = k.$$

Recorded threshold attainment



The nine-reading example

$$80, 62, 48, 40, 35, \text{missing}, 28, 25, 23.$$

At threshold 40:

The nine-reading example

$$80, 62, 48, 40, 35, \text{missing}, 28, 25, 23.$$

At threshold 40:

$$N_{\leq} = 5, \quad N_{>} = 3, \quad N_{\text{missing}} = 1.$$

The counter identity gives $5 + 3 + 1 = 9$.

The denominator of a proportion

$$f(\theta) = \frac{N_{\leq}(\theta)}{N_{\text{known}}}, \qquad N_{\text{known}} > 0.$$

The denominator of a proportion

$$f(\theta) = \frac{N_{\leq}(\theta)}{N_{\text{known}}}, \qquad N_{\text{known}} > 0.$$

For the supplied record,

$$f(40) = \frac{5}{8} = 0.625.$$

Changing the threshold

At threshold 35, only $35, 28, 25, 23$ qualify:

$$f(35) = \frac{4}{8} = 0.5.$$

Changing the threshold

At threshold 35, only $35, 28, 25, 23$ qualify:

$$f(35) = \frac{4}{8} = 0.5.$$

The known-reading denominator remains 8 because the data and missingness did not change.

The first recorded qualifying time

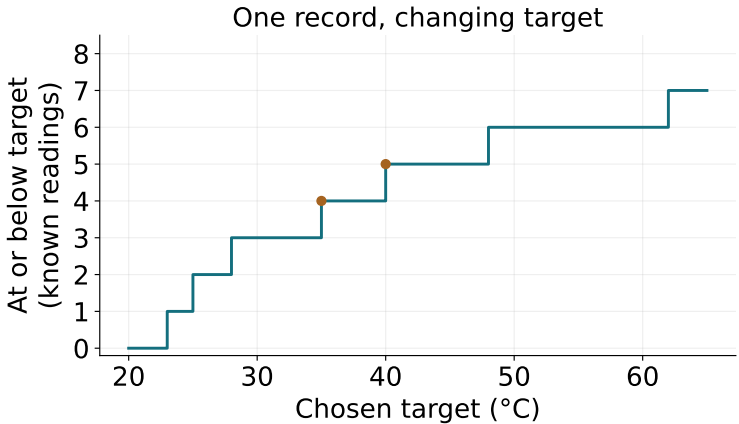
$$t_{\text{first}} = \min\{t_i : x_i \text{ known and } x_i \leq \theta\}.$$

The first recorded qualifying time

$$t_{\text{first}} = \min\{t_i : x_i \text{ known and } x_i \leq \theta\}.$$

For observations at integer minutes and $\theta = 40$, $t_{\text{first}} = 3$ min.

Threshold count as a step function



Recorded time and continuous crossing

The record contains $T(2) = 48$ and $T(3) = 40$.

Recorded time and continuous crossing

The record contains $T(2) = 48$ and $T(3) = 40$. The first qualifying observation occurs at 3 min. An exact continuous crossing requires a model between observations. With oscillation, even earlier unobserved crossings are possible.

Monotonicity of threshold counts

$$\{i : x_i \leq \theta_1\} \subseteq \{i : x_i \leq \theta_2\} \quad (\theta_1 \leq \theta_2).$$

Monotonicity of threshold counts

$$\{i : x_i \le \theta_1\} \subseteq \{i : x_i \le \theta_2\} \quad (\theta_1 \le \theta_2).$$

Therefore $N_{\le}(\theta_1) \le N_{\le}(\theta_2)$.

A loop with paired observations

```
for time, value in zip(times, readings):
    category = classify(value, target)
    counts[category] += 1
    if category == "at_or_below" and first is None:
        first = time
```

Equal list lengths must be checked: `zip` stops at the shorter input.