From weight and height to a z-score

The LMS method, the reference table lookup, the three z-scores and what each measures. Computed on 930 children it gives a mean of -0.67 and a standard deviation of 1.22 — and that second number is already a finding.

Pierre Robentz Cassion

Lesson 2 of 8

Nutrition Analysis: CMAM and SMART — Measuring a child

What this lesson covers

- What a z-score says
- Three z-scores, three questions
- The LMS method
- The lookup key has three parts
- Computing it
- Flagging: two rules, two answers
- The standard deviation is already a finding
- Save the z-scores with their inputs
- What comes next

What a z-score says

- A z-score answers one question: how far is this child from the median of a healthy reference population, in standard deviations?

Three z-scores, three questions

Score	Compares	Deficit it detects	Recovers?
Weight-for-height (WHZ)	Weight against height	Acute — wasting	Yes, in weeks
Height-for-age (HAZ)	Height against age	Chronic — stunting	No
Weight-for-age (WAZ)	Weight against age	Both together — underweight	Partly

Three z-scores, three questions

- Weight-for-age is the one to be careful with — It mixes the two deficits, so a low WAZ does not say whether the child...

The LMS method

- **L** — the Box-Cox power that normalises a skewed distribution
- **M** — the median
- **S** — the coefficient of variation

The LMS method — Example

```
z = ((weight / M)^L - 1) / (L * S)
```

The LMS method — In Python

```
import pandas as pd

reference = pd.read_csv("who-2006-weight-for-lenhei.csv")
print(reference.head(3))
print(f"{len(reference)} rows: sex x standard x lenhei in 0.1 cm steps")
```

The LMS method — In R

```
reference <- readr::read_csv("who-2006-weight-for-lenhei.csv")
nrow(reference)
```

The lookup key has three parts — In Python

```python
def lookup_key(row):
    lying = row["measured_lying"] == "true"
    standard = "L" if row["age_months"] < 24 else "H"

    lenhei = row["height_cm"]
    if standard == "L" and not lying:
        lenhei += 0.7
    elif standard == "H" and lying:
        lenhei -= 0.7

    return row["sex"], standard, round(lenhei, 1)
```

The lookup key has three parts — In R

```r
lookup_key <- function(sex, age, height, lying) {
  standard <- if (age < 24) "L" else "H"
  lenhei <- height +
    if (standard == "L" && !lying) 0.7 else if (standard == "H" && lying) -0.7 else 0
  list(sex = sex, lorh = standard, lenhei = round(lenhei, 1))
}
```

The lookup key has three parts

- The standard follows the age — not the position
- The adjustment goes on the measurement, not the standard — You are converting the measurement into the one the standard...
- The rounding is to one decimal, on both sides, in one place — A float computed from an adjustment is not the same float...

Computing it — In Python

```
reference["key"] = list(zip(reference["sex"], reference["lorh"],
                            reference["lenhei"].round(1)))
lms = reference.set_index("key")[["l", "m", "s"]]

def whz(row):
    key = lookup_key(row)
    if key not in lms.index or pd.isna(row["weight_kg"]):
        return None
    l, m, s = lms.loc[key]
    return (((row["weight_kg"] / m) ** l) - 1) / (l * s)

smart["whz"] = smart.apply(whz, axis=1)
print(f"{smart['whz'].notna().sum()} of {len(smart)} computable")
```

Computing it — In R

```
# In R, use the official package rather than the table:
# anthro::anthro_zscores(sex = ..., age = ..., weight = ..., lenhei = ..., measure = ...)
```

Computing it

- **Missing weight or age** — 32 children, and they are missing data.
- **Height outside the reference range** — the length standard runs 45.0 to 110.0 cm and the height standard 65.0 to...
- **A measurement in the wrong unit** — four heights recorded in metres, which the joining course's lab found by...
- Report the three separately — "66 children excluded" hides that some were data errors you could have fixed

Flagging: two rules, two answers

- **WHO flags** — fixed bounds, exclude z outside -5 to +5. Absolute, comparable across surveys.
- **SMART flags** — relative, exclude observations more than 3 SD from the *survey's own* mean. Adapts to the survey, and...

Flagging: two rules, two answers — In Python

```
who_flagged = smart["whz"].between(-5, 5)

mean, sd = smart["whz"].mean(), smart["whz"].std()
smart_flagged = smart["whz"].between(mean - 3 * sd, mean + 3 * sd)

print(f"WHO flags keep {who_flagged.sum()}, SMART flags keep {smart_flagged.sum()}")
```

Flagging: two rules, two answers — In R

```
smart <- smart |>
  mutate(who_ok = between(whz, -5, 5),
         smart_ok = between(whz, mean(whz, na.rm = TRUE) - 3 * sd(whz, na.rm = TRUE),
                                 mean(whz, na.rm = TRUE) + 3 * sd(whz, na.rm = TRUE)))
```

The standard deviation is already a finding — In Python

```
analysable = smart.loc[who_flagged, "whz"]
print(f"n = {len(analysable)}, mean = {analysable.mean():.2f}, "
      f"sd = {analysable.std():.2f}")
```

The standard deviation is already a finding — In R

```
smart |> filter(who_ok) |> summarise(n = n(), mean = mean(whz), sd = sd(whz))
```

Save the z-scores with their inputs — In Python

```
smart[["child_id", "cluster", "team", "age_months", "sex", "weight_kg",
       "height_cm", "measured_lying", "oedema", "whz"]].to_csv(
    "outputs/smart_with_zscores.csv", index=False)
```

Save the z-scores with their inputs — In R

```
readr::write_csv(smart, here::here("outputs", "smart_with_zscores.csv"))
```

What comes next

- You have a z-score for every analysable child.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/nutrition-analysis-cmam-smart/nutrition-02-z-scores