

Du poids et de la taille au score z

La méthode LMS, la recherche dans la table de référence, les trois scores z et ce que chacun mesure. Calculé sur 930 enfants, il donne une moyenne de -0,67 et un écart-type de 1,22 — et ce second nombre est déjà un constat.

Pierre Robentz Cassion

Leçon 2 sur 8

Analyse nutritionnelle : PCIMA et SMART — Mesurer un enfant

Ce que couvre cette leçon

- Ce que dit un score z
- Trois scores z , trois questions
- La méthode LMS
- La clé de recherche a trois parties
- Le calcul
- Le signalement — deux règles, deux réponses
- L'écart-type est déjà un constat
- Enregistrez les scores z avec leurs entrées
- La suite

Ce que dit un score z

- Un score z répond à une seule question — à quelle distance cet enfant est-il de la médiane d'une population de référence en bonne santé, en écarts-types ?

Trois scores z, trois questions

Score	Compare	Déficit détecté	Récupérable ?
Poids-pour-taille (WHZ)	Le poids à la taille	Aigu — émaciation	Oui, en semaines
Taille-pour-âge (HAZ)	La taille à l'âge	Chronique — retard de croissance	Non
Poids-pour-âge (WAZ)	Le poids à l'âge	Les deux à la fois — insuffisance pondérale	En partie

Trois scores z, trois questions

- Le poids-pour-âge est celui dont il faut se méfier — Il mélange les deux déficits, si bien qu'un WAZ bas ne dit pas si...

- **L** — la puissance de Box-Cox qui normalise une distribution asymétrique
- **M** — la médiane
- **S** — le coefficient de variation

La méthode LMS — Exemple

$$z = ((\text{weight} / M)^L - 1) / (L * S)$$

La méthode LMS — En 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")
```

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

La clé de recherche a trois parties — En 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)
```

La clé de recherche a trois parties — En 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))  
}
```

La clé de recherche a trois parties

- Le standard suit l'âge — non la position
- L'ajustement porte sur la mesure, non sur le standard — Vous convertissez la mesure en celle que le standard attend
- L'arrondi est à une décimale, des deux côtés, en un seul endroit — Un flottant issu d'un ajustement n'est pas le même. . .

```
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")
```

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

- **Poids ou âge manquant** — 32 enfants, et ce sont des données manquantes.
- **Taille hors de la plage de référence** — le standard de longueur va de 45,0 à 110,0 cm et celui de taille de 65,0 à...
- **Une mesure dans la mauvaise unité** — quatre tailles enregistrées en mètres, que le laboratoire du cours sur les...
- **Rapportez les trois séparément** — « 66 enfants exclus » masque le fait que certains étaient des erreurs corrigibles

Le signalement — deux règles, deux réponses

- **Valeurs aberrantes OMS** — bornes fixes, exclure les z hors de -5 à +5. Absolu, comparable entre enquêtes.
- **Valeurs aberrantes SMART** — relatif, exclure les observations à plus de 3 ET de la moyenne *de l'enquête elle-même*...

Le signalement — deux règles, deux réponses — En 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()}")
```

Le signalement — deux règles, deux réponses — En 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)))
```

L'écart-type est déjà un constat — En Python

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

L'écart-type est déjà un constat — En R

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

Enregistrez les scores z avec leurs entrées — En 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)
```

Enregistrez les scores z avec leurs entrées — En R

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

- Vous avez un score z pour chaque enfant analysable.

Lire la leçon complète, avec le code exécutable

```
https://data-analysis.cassion.dev/fr/cours/nutrition-analysis-cmam-smart/  
nutrition-02-z-scores
```