Des règles que le secteur vous a déjà données

Transformer « cela semble faux » en une table de règles avec une source, une gravité et un compte. Bornes dures face aux limites physiques, bornes souples face aux seuils du secteur, et la discipline du signalement plutôt que de la suppression.

Pierre Robentz Cassion

Leçon 5 sur 8

Nettoyage et validation des données — Des valeurs qui ne peuvent pas être vraies

Ce que couvre cette leçon

- « Cela semble faux » n'est pas une règle
- Les seuils existent déjà
- Des règles comme données, pas comme conditions
- Appliquer la table
- Bornes dures et bornes souples
- Des règles qui comparent deux colonnes
- Signalez, ne supprimez pas
- Le taux de signalement est lui-même un indicateur
- La suite

« Cela semble faux » n'est pas une règle

- Quiconque fait ce métier finit par acquérir l'œil.

Les seuils existent déjà

Domaine	Règle	Source
PB, 6-59 mois	En dessous de 80 mm ou au-dessus de 220 mm, ce n'est pas une mesure	OMS / pratique de terrain PCIMA
Score Z poids-taille	Hors de l'intervalle -5 à +5 de la référence	Normes OMS 2006 de croissance
Anthropométrie SMART	Signaler les scores Z à plus de 3 ET de la moyenne de l'enquête	Rapport de plausibilité SMART
Quantité d'eau	Moins de 15 litres par personne et par jour est sous le minimum	Sphere
Collecte d'eau	Un aller-retour de plus de 30 minutes est un service limité, pas élémentaire	Échelles de service JMP
Chlore résiduel libre	0,2 à 2,0 mg/L au point de consommation	Sphere / OMS
Couverture	Des doses administrées supérieures à la population cible demandent une explication	Guide d'indicateurs UNICEF

Les seuils existent déjà

- Citez la source dans la règle — Un seuil accompagné d'une référence survit à la contestation ; le même chiffre sans. . .

Des règles comme données, pas comme conditions — En Python (suite)

```
RULES = [
    {
        "id": "muac-range",
        "column": "muac_mm",
        "severity": "error",
        "source": "WHO / CMAM field practice",
        "test": lambda d: d["muac_mm"].between(80, 220) | d["muac_mm"].isna(),
        "message": "MUAC outside 80-220 mm",
    },
    {
        "id": "muac-unit-error",
        "column": "muac_mm",
        "severity": "warning",
        "source": "unit check, cm entered as mm",
        "test": lambda d: ~d["muac_mm"].between(1, 40),
        "message": "MUAC looks like centimetres",
```

Des règles comme données, pas comme conditions — En Python (suite)

```
    },
    {
        "id": "age-in-scope",
        "column": "age_months",
        "severity": "warning",
        "source": "CMAM screening protocol, 6-59 months",
        "test": lambda d: d["age_months"].between(6, 59) | d["age_months"].isna(),
        "message": "age outside the screening window",
    },
]
```

Des règles comme données, pas comme conditions — En R

```r
RULES <- tibble::tribble(
  ~id,               ~column,      ~severity,  ~source,                              ~message,
  "muac-range",      "muac_mm",    "error",    "WHO / CMAM field practice",          "MUAC
     outside 80-220 mm",
  "muac-unit-error", "muac_mm",    "warning",  "unit check, cm entered as mm",       "MUAC
     looks like centimetres",
  "age-in-scope",    "age_months", "warning",  "CMAM screening protocol, 6-59 mo",   "age
     outside the screening window"
)

TESTS <- list(
  `muac-range`      = function(d) dplyr::between(d$muac_mm, 80, 220) | is.na(d$muac_mm),
  `muac-unit-error` = function(d) !dplyr::between(d$muac_mm, 1, 40),
  `age-in-scope`    = function(d) dplyr::between(d$age_months, 6, 59) | is.na(d$age_months)
)
```

Des règles comme données, pas comme conditions

- Chaque test réussit sur une valeur manquante — C'est délibéré et cela prend constamment les gens en défaut : `NA < 80...`

Appliquer la table — En Python (suite)

```python
def apply_rules(df, rules):
    results = []
    flags = pd.DataFrame(index=df.index)
    for rule in rules:
        ok = rule["test"](df)
        flags[rule["id"]] = ~ok
        results.append({
            "rule": rule["id"],
            "severity": rule["severity"],
            "column": rule["column"],
            "failed": int((~ok).sum()),
            "share": round((~ok).mean(), 4),
            "source": rule["source"],
        })
    return pd.DataFrame(results), flags
```

Appliquer la table — En Python (suite)

```
report, flags = apply_rules(muac, RULES)
print(report.to_string(index=False))
```

Appliquer la table — En R

```r
apply_rules <- function(df, rules, tests) {
  flags <- purrr::map_dfc(rules$id, function(id) {
    tibble::tibble(!!id := !tests[[id]](df))
  })
  report <- rules |>
    dplyr::mutate(
      failed = purrr::map_int(id, ~ sum(flags[[.x]], na.rm = TRUE)),
      share  = round(failed / nrow(df), 4)
    )
  list(report = report, flags = flags)
}

out <- apply_rules(muac, RULES, TESTS)
out$report
```

Appliquer la table

Règle	Gravité	Échecs	Part
muac-range	error	7	0,17 %
muac-unit-error	warning	7	0,17 %
age-in-scope	warning	0	0,00 %

Bornes dures et bornes souples

- **Les bornes dures** sont physiquement ou logiquement impossibles. Une taille de ménage négative, un PB de 450 mm, une...
- **Les bornes souples** sont implausibles mais possibles. Dix litres par personne et par jour est sous le minimum...

Bornes dures et bornes souples — En Python

```
suspect = wash[wash["litres_per_person_day"] > 60][
    ["household_id", "community", "household_size", "litres_per_person_day", "water_source"]
]
suspect["implied_total"] = suspect["litres_per_person_day"] * suspect["household_size"]
print(suspect)
```

Bornes dures et bornes souples — En R

```
wash |>
  filter(litres_per_person_day > 60) |>
  mutate(implied_total = litres_per_person_day * household_size) |>
  select(household_id, community, household_size, litres_per_person_day,
         water_source, implied_total)
```

Des règles qui comparent deux colonnes — En Python

```
CROSS_RULES = [
    {
        "id": "referral-without-measurement",
        "severity": "warning",
        "test": lambda d: ~(d["outcome"].str.startswith("referred") & d["muac_mm"].isna()),
        "message": "referred but no MUAC recorded",
    },
    {
        "id": "outcome-contradicts-muac",
        "severity": "warning",
        "test": lambda d: ~((d["muac_mm"] >= 125) & (d["oedema"] != True)
                            & (d["outcome"] != "no-action")),
        "message": "no-action expected from the measurement",
    },
]
```

Des règles qui comparent deux colonnes — En R

```
CROSS_TESTS <- list(
  `referral-without-measurement` = function(d)
    !(startsWith(d$outcome, "referred") & is.na(d$muac_mm)),
  `outcome-contradicts-muac` = function(d)
    !(d$muac_mm >= 125 & !d$oedema & d$outcome != "no-action")
)
```

Signalez, ne supprimez pas — En Python

```
muac = muac.join(flags.add_prefix("flag_"))
muac["flag_any_error"] = flags[[r["id"] for r in RULES if r["severity"] == "error"]].any(axis
    =1)
```

Signalez, ne supprimez pas — En R

```
muac <- dplyr::bind_cols(muac, dplyr::rename_with(out$flags, ~ paste0("flag_", .x)))
muac$flag_any_error <- dplyr::if_any(dplyr::starts_with("flag_"), identity)
```

Signalez, ne supprimez pas

- **Le compte reste disponible.** « 4 218 dépistages, 7 exclus pour mesure implausible » exige les deux nombres, et un…
- **L'exclusion est réversible.** Un relecteur en désaccord avec une règle peut relancer l'analyse sans elle, en une…
- **Les signalements sont analysables.** Ce qui nous amène à la dernière section.

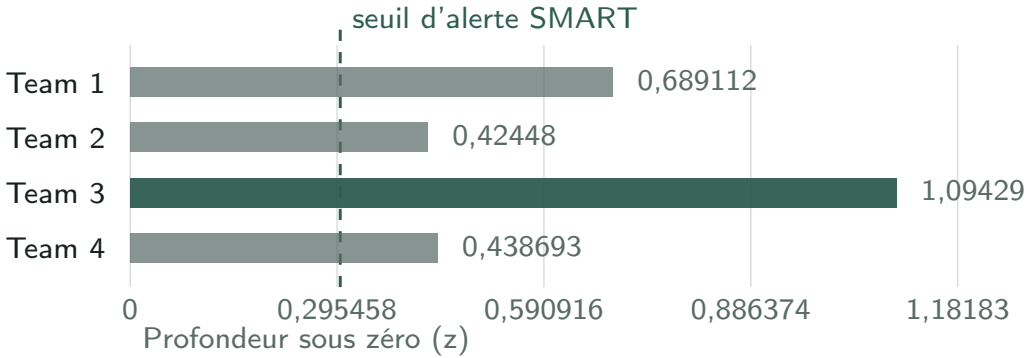
Signalez, ne supprimez pas — En Python

```
analysis = muac[~muac["flag_any_error"]]
print(f"{len(muac)} screenings, {len(analysis)} analysed, "
      f"{muac['flag_any_error'].sum()} excluded")
```

Signalez, ne supprimez pas — En R

```
analysis <- muac |> filter(!flag_any_error)
sprintf("%d screenings, %d analysed, %d excluded",
        nrow(muac), nrow(analysis), sum(muac$flag_any_error))
```

Le taux de signalement est lui-même un indicateur



Score Z poids-pour-taille moyen par équipe de mesure dans l'enquête SMART, tracé en profondeur sous zéro. Les grappes ayant été attribuées aux équipes indépendamment de l'état nutritionnel, un écart de cette ampleur entre équipes est un défaut de mesure et non une différence réelle entre populations.

Le taux de signalement est lui-même un indicateur — En Python

```
print(muac.groupby("commune")[["flag_muac_unit_error", "flag_any_error"]].mean())
```

Le taux de signalement est lui-même un indicateur — En R

```r
muac |>
  summarise(across(starts_with("flag_"), mean), .by = commune)
```

Le taux de signalement est lui-même un indicateur

Une règle qui se déclenche partout est une règle sur le monde. Une règle qui se déclenche à un seul endroit est une règle sur une équipe.

La suite

- Toutes les règles vues jusqu'ici comparent une valeur à un nombre.

Lire la leçon complète, avec le code exécutable

https://data-analysis.cassion.dev/fr/cours/data-cleaning-and-validation/cleaning-05-plausibility-rules