

Les facteurs, et l'ordre qu'exige un rapport

Ce qu'est réellement un facteur, le piège de `as.numeric()` qui renvoie silencieusement des positions de niveaux, et les verbes forcats qui mettent une échelle JMP dans l'ordre de l'échelle plutôt que dans celui de l'alphabet.

Pierre Robentz Cassion

Leçon 4 sur 8

R et le tidyverse pour les données de programme — Des données qui gardent leur sens

Ce que couvre cette leçon

- Un facteur est un entier assorti d'une table de correspondance
- Le piège de `as.numeric()`
- Fixer l'ordre voulu
- Les facteurs ordonnés
- Les niveaux inutilisés, et les deux comportements qui divergent
- Réduire une longue traîne
- Facteurs et valeurs manquantes
- Là où les facteurs mordent dans une jointure
- Ce qui vient ensuite

Un facteur est un entier assorti d'une table de correspondance — En R

```
service <- factor(c("basic", "limited", "unimproved", "basic"))
```

```
levels(service)
```

```
#> [1] "basic"      "limited"     "unimproved"
```

```
as.integer(service)
```

```
#> [1] 1 2 3 1
```

Le piège de `as.numeric()` — En R

```
age <- factor(c("10", "9", "8"))
```

```
as.numeric(age)
```

```
#> [1] 1 3 2
```

Le piège de `as.numeric()` — En R

```
as.numeric(as.character(age))  
#> [1] 10 9 8
```

Fixer l'ordre voulu — En R

```
library(forcats)

ladder <- fct_relevel(service, "unimproved", "limited", "basic")
levels(ladder)
#> [1] "unimproved" "limited"      "basic"
```

Fixer l'ordre voulu — En R

```
fct_infreq(source)      # le niveau le plus frequent en premier  
fct_reorder(name, value) # ordonner un facteur par une autre colonne
```

Fixer l'ordre voulu — En R

```
library(dplyr)

by_commune |>
  mutate(commune = fct_reorder(commune, gam_rate)) |>
  ggplot2::ggplot(ggplot2::aes(gam_rate, commune)) +
  ggplot2::geom_col()
```

```
service[1] >= service[2]  
#> Warning: '>=' not meaningful for factors  
#> [1] NA
```

```
ladder <- factor(  
  c("basic", "limited"),  
  levels = c("unimproved", "limited", "basic"),  
  ordered = TRUE  
)
```

```
ladder[1] >= ladder[2]  
#> [1] TRUE
```

- Employez les facteurs ordonnés avec parcimonie — Ils changent la façon dont les modèles traitent la variable. . .

Les niveaux inutilisés, et les deux comportements qui divergent — En R

```
d <- tibble::tibble(k = factor(c("a", "b", "a"), levels = c("a", "b", "c")))
```

```
nrow(dplyr::count(d, k))
```

```
#> [1] 2
```

```
length(table(d$k))
```

```
#> [1] 3
```

Les niveaux inutilisés, et les deux comportements qui divergent — En R

```
dplyr::count(d, k, .drop = FALSE)  # conserver le niveau vide  
droplevels(d$k)                    # retirer les niveaux sans ligne  
forcats::fct_drop(d$k)             # la meme chose, en forcats
```

Réduire une longue traîne — En R

```
source <- factor(wash$water_source)
length(levels(source))
#> [1] 10
```

```
levels(fct_lump_n(source, 4))
#> [1] "borehole" "piped-into-dwelling" "piped-into-yard" "public-tap" "Other"
```

Réduire une longue traîne

- Ne repliez pas avant de calculer un indicateur — Other mêle ici les sources protégées, qui sont améliorées, et l'eau...

Réduire une longue traîne — En R

```
fct_recode(source,  
  improved = "borehole",  
  improved = "protected-well",  
  unimproved = "surface-water"  
)
```

```
table(muac$outcome, useNA = "ifany")
```

```
muac <- muac |> mutate(outcome = fct_na_value_to_level(outcome, level = "non renseigné"))
```

Là où les facteurs mordent dans une jointure — En R

```
left_join(muac, sites, by = "commune")
```

Ce qui vient ensuite

- Les colonnes portent désormais leur sens et leur ordre.

Lire la leçon complète, avec le code exécutable

https:

`//data-analysis.cassion.dev/fr/cours/r-for-programme-data/r-04-factors`