

# TP: Data Wrangling

Eric Marcon

01 February 2024

# Données

Les données sont obtenues selon un protocole clair et reproductible.

Elles sont enregistrées dans des fichiers structurés :

- Données rectangulaires (*tidy*) ;
- Dans un tableur (Excel est le standard) ;
- Sans répétitions (principe DNNRY) : un fichier par objet ;
- Sécurisés : plan de gestion des données pour la sauvegarde et l'archivage.

Interdiction de modifier les données brutes après nettoyage.

Mesures de diamètres et hauteurs de Pins et Cèdres du Ventoux en 2020.

Ouvrir “data/Inv\_GEEFT\_Ventoux\_09-2020.xlsx”.

Règles :

- une colonne par variable, une ligne par individu ;
- toutes les valeurs d'une variable sont de même type ;
- figer les volets pour simplifier le travail ;
- éventuellement, mettre sous forme de tableau ;
- formater les nombres pour la lisibilité, utiliser les filtres et tris librement ;
- pas de fusion de cellules, de surtitre,...

Valider une version *définitive* du fichier venant du terrain (“Données brutes”).

Possible mais pas standard parce que le fichier de données est binaire.

```
library("readxl")
read_excel("data/Inv_GEEFT_Ventoux_09-2020.xlsx") |>
  print() ->
  ventoux_excel
```

```
## # A tibble: 223 x 3
##   Espèce `Diamètre (cm)` `Hauteur réelle (m)`
##   <chr>           <dbl>           <dbl>
## 1 P               16.5             14.8
## 2 P               13              11.2
## 3 P               23.8             8.71
## 4 C               16.5             8.9
## 5 P               35              21.4
## # i 218 more rows
```

Texte séparé par des virgules en anglais. Pour les pays latins, la virgule est le séparateur décimal : le séparateur de colonne devient le point-virgule.

Conserver le fichier Excel dans les archives de terrain et exporter les données dans le projet R :

- Fichier > Enregistrer sous... : CSV UTF-8.
- Fermer Excel pour déverrouiller le fichier.

Dans le tidyverse, utiliser `read_csv()` ou `read_csv2()`.

```
read_csv2("data/Inv_GEEFT_Ventoux_09-2020.csv") |>
  print() ->
  ventoux
```

```
## # A tibble: 223 x 3
##   Espèce `Diamètre (cm)` `Hauteur réelle (m)`
##   <chr>          <dbl>          <dbl>
## 1 P              16.5            14.8
## 2 P              13             11.2
## 3 P             23.8             8.71
## 4 C             16.5             8.9
## 5 P             35             21.4
## # i 218 more rows
```

Les noms de colonnes doivent être des noms d'objets R valides pour simplifier le code :

```
# Hauteur moyenne, difficile à manipuler  
ventoux$`Hauteur réelle (m)` |> mean()
```

```
## [1] 14.88399
```

Renommer les colonnes:

```
ventoux |>  
  rename(  
    espece = Espèce,  
    diametre = `Diamètre (cm)`,  
    hauteur = `Hauteur réelle (m)`  
  ) |>  
  print() ->  
  ventoux
```

```
## # A tibble: 223 x 3  
##   espece diametre hauteur  
##   <chr>      <dbl>   <dbl>  
## 1 P          16.5    14.8  
## 2 P          13     11.2
```



Les noms des espèces ne sont pas clairs.

```
ventoux |>
  mutate(
    espece = case_match(
      espece,
      "P" ~ "Pin",
      "C" ~ "Cèdre"
    )
  ) |>
  print() ->
  ventoux
```

```
## # A tibble: 223 x 3
##   espece diametre hauteur
##   <chr>      <dbl>   <dbl>
## 1 Pin        16.5    14.8
## 2 Pin         13    11.2
## 3 Pin        23.8     8.71
## 4 Cèdre      16.5     8.9
## 5 Pin        35     21.4
## # i 218 more rows
```

A ce stade, l'objet ventoux est nettoyé : il ne devra plus être modifié.

→ Réécrire le pipeline complet.

Les traitements ultérieurs devront créer de nouveaux objets.

# Transformation

Calculer la surface terrière.

```
ventoux |>
  mutate(G = diametre^2 * pi /40000) |>
  print() ->
  ventoux_g
```

```
## # A tibble: 223 x 4
##   espece diametre hauteur      G
##   <chr>      <dbl>   <dbl>  <dbl>
## 1 Pin        16.5    14.8  0.0214
## 2 Pin         13     11.2  0.0133
## 3 Pin        23.8     8.71  0.0445
## 4 Cèdre      16.5     8.9   0.0214
## 5 Pin        35     21.4  0.0962
## # i 218 more rows
```

## TP: Data Wrangling

Eric Marcon

Données

Transformation