TP Tidyverse

Eric Marcon

2025-09-12

Table des matières

L'objectif de ce TP est de pratiquer les quelques commandes essentiels de la bagarre avec les données dans le tidyverse (Wickham, Çetinkaya-Rundel, et Grolemund 2023).

Données

L'inventaire de la parcelle 6 de Paracou est fourni. C'est un tableau qui contient une ligne par arbre mesuré sur le terrain, avec son identification botanique et ses coordonnées géographiques et sa taille.

C'est un fichier texte au format CSV latin: le séparateur décimal est la virgule. Pour le lire, on utilise la fonction `read_csv2()` du tidyverse, plus efficace que `read.csv2` de R base: le résultat est un tibble, qui est un dataframe amélioré.

```
library("tidyverse")
read_csv2("data/Paracou6.csv")
```

```
## # A tibble: 3,541 x 10
##   SubPlot idTree Xfield Yfield Family spName Genus
##     <dbl>  <dbl>  <dbl>  <dbl> <chr>  <chr>  <chr>
## 1       1 100655    7.5   180. Perac~ Pogon~ Pogo~
## 2       1 100657    8     184. Perac~ Pogon~ Pogo~
## 3       1 100658    5.5   182. Chrys~ Lican~ Lica~
## 4       1 100659    1.5   186  Eupho~ Sandw~ Sand~
## 5       1 100660    0.5   190  Fabac~ Eperu~ Eper~
## # i 3,536 more rows
## # i 3 more variables: Species <chr>,
## #   CircCorr <dbl>, species <chr>
```

Les colonnes sont la sous-parcelle (SubPlot, sans intérêt ici), l'identifiant de l'arbre (idTree, unique), sa position (XField et YField, dans le repère de la parcelle), son identification botanique (Family, Genus, Species et spName qui regroupe genre et espèce) et sa taille (CircCorr, circonférence à hauteur de poitrine, corrigée si l'arbre n'est pas rond).

Carte des Wapas

Les wapas (*Eperua*) sont un genre abondant dont la distribution des deux espèces présentes à Paracou a été très étudiée. Nous allons en faire une carte. Pour cela, nous allons utiliser les commandes classiques du tidyverse, enchâinées grâce au pipeline `%>%`. Chaque étape est ajoutée à la précédente quand celle-ci fonctionne correctement.

Filtrage du genre *Eperua*

Pour ne retenir que certaines lignes, on utilise la fonction `filter()`: on dit qu'on *filtre* les lignes, et on évite de parler de *sélection* des lignes, la fonction `select()` servant à sélectionner des colonnes.

```
# Lecture des données
read_csv2("data/Paracou6.csv") %>%
  # Filtrage des wapas
  filter(Genus == "Eperua")
```

```
## # A tibble: 333 x 10
##   SubPlot idTree Xfield Yfield Family spName Genus
##     <dbl>  <dbl>  <dbl>  <dbl> <chr>  <chr>  <chr>
## 1       1 100660    0.5    190 Fabac~ Eperu~ Eper~
## 2       1 100665    2.5    206 Fabac~ Eperu~ Eper~
## 3       1 100676    9      222 Fabac~ Eperu~ Eper~
## 4       1 100677    7      226 Fabac~ Eperu~ Eper~
## 5       1 100691    4      247 Fabac~ Eperu~ Eper~
## # i 328 more rows
## # i 3 more variables: Species <chr>,
## #   CircCorr <dbl>, species <chr>
```

Le code équivalent en R classique est:

```
# Lecture des données
paracou6 <- read.csv2("data/Paracou6.csv")
# Quels sont les wapas ?
est_wapa <- paracou6$Genus == "Eperua"
# Filtrage des lignes par le vecteur logique
paracou6_wapa <- paracou6[est_wapa, ]
# Affichage des premières lignes
head(paracou6_wapa)
```

```
##    SubPlot idTree Xfield Yfield   Family
## 5        1 100660    0.5  190.0 Fabaceae
## 9        1 100665    2.5  206.0 Fabaceae
## 19       1 100676    9.0  222.0 Fabaceae
## 20       1 100677    7.0  226.0 Fabaceae
## 32       1 100691    4.0  247.0 Fabaceae
## 33       1 100692    9.5  249.5 Fabaceae
##            spName  Genus Species CircCorr species
## 5  Eperua_falcata Eperua falcata     77.0 falcata
## 9  Eperua_falcata Eperua falcata    167.0 falcata
## 19 Eperua_falcata Eperua falcata    133.0 falcata
## 20 Eperua_falcata Eperua falcata    123.5 falcata
## 32 Eperua_falcata Eperua falcata    150.0 falcata
## 33 Eperua_falcata Eperua falcata     98.0 falcata
```

La syntaxe classique est plus difficile à lire et nécessite de nombreuses variables intermédiaires. En pratique, un pipeline est beaucoup plus efficace. D'autre part, l'affichage à l'écran des tibbles est bien plus lisible que celui des dataframes.

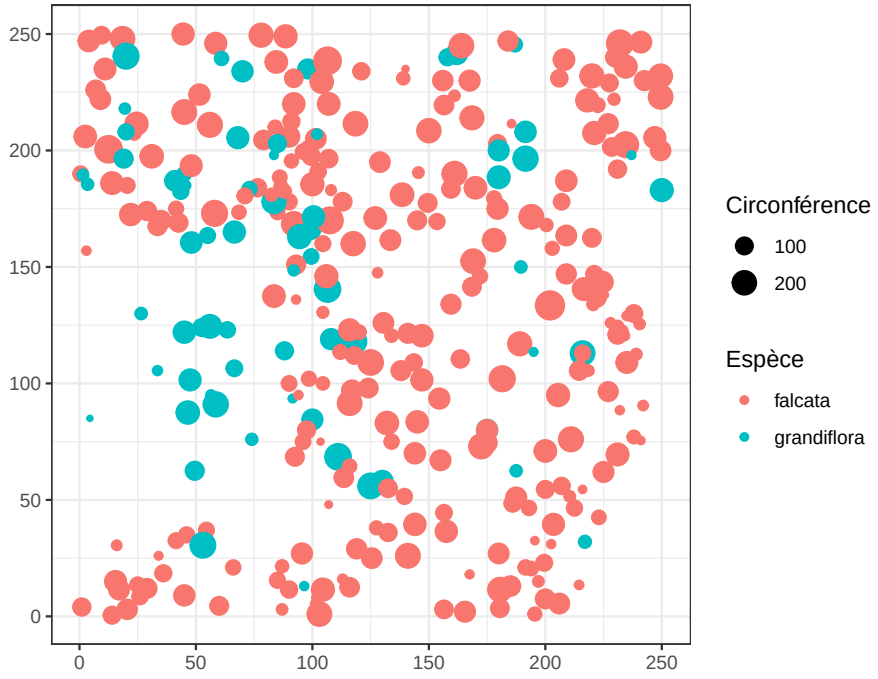
Graphique

Il reste à créer un graphique avec `ggplot()`. La géométrie définit le type de graphique: ici, des points avec `geom_point()`. L'esthétique (fonction `aes()`) consiste à décrire les caractéristiques de ces points:

- obigatoirement leur position `x` et `y`,
- éventuellement leur taille `size` et leur couleur `color`.

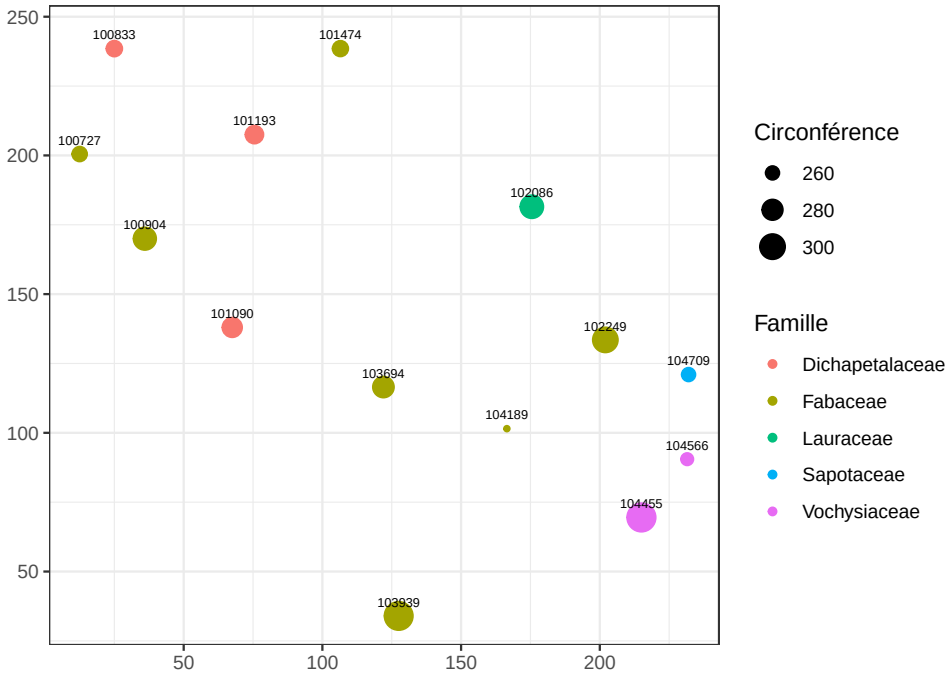
```
# Lecture des données
read_csv2("data/Paracou6.csv") %>%
  # Filtrage des wapas
  filter(Genus == "Eperua") %>%
  # Graphique. Les éléments du graphique sont ajoutés avec l'opérateur +
  ggplot() +
```

```
  # Un point par arbre
  geom_point(aes(x = Xfield, y = Yfield, size = CircCorr, color = Species)) +
  # Même échelle pour les abscisses et ordonnées pour ne pas déformer la carte
  coord_fixed() +
  # Texte des axes et de la légende
  labs(x = NULL, y = NULL, size = "Circonférence", color = "Espèce")
```



L'esthétique est héritable: elle peut être déclarée dans chaque géométrie ou au niveau global de `ggplot()` pour s'appliquer à toutes les géométries s'il y en a plusieurs. Le graphique suivant affiche la carte des très gros arbres (plus de 250cm de circonférence) avec leur identifiant. Les positions (esthétique `x` et `y`) sont communes aux deux géométries (points et texte), mais chacune des géométries a sa propre esthétique en complément: la taille et la couleur des points et l'étiquette.

```
# Lecture des données
read_csv2("data/Paracou6.csv") %>%
  # Filtrage des très gros arbres
  filter(CircCorr > 250) %>%
  # Graphique. Les éléments du graphique sont ajoutés avec l'opérateur +
  ggplot(aes(x = Xfield, y = Yfield)) +
  # Un point par arbre
  geom_point(aes(size = CircCorr, color = Family)) +
  # Etiquettes. Taille du texte et décalage au dessus du point
  geom_text(aes(label = idTree), size = 2, nudge_y = 5) +
  # Même échelle pour les abscisses et ordonnées pour ne pas déformer la carte
  coord_fixed() +
  # Texte des axes et de la légende
  labs(x = NULL, y = NULL, size = "Circonférence", color = "Famille")
```



Calculs

Des calculs peuvent être réalisés à partir des données du tableau:

- au niveau individuel (chaque ligne), en ajoutant des champs calculés (c'est-à-dire des colonnes) avec la fonction `mutate()`,
- en regroupant des lignes pour obtenir des statistiques descriptives, comme le nombre d'individus, la moyenne ou l'écart-type d'une valeur.

L'objectif est maintenant de lister les espèces les plus abondantes dans la parcelle en calculant leur effectif et leur surface terrière totale.

Sélection des colonnes

La première étape consiste à sélectionner les colonnes utiles. Elle est rarement indispensable mais permet de mieux voir les données. Ici, on conserve le nom de l'espèce et la circonférence.

```
# Lecture des données
read_csv2("data/Paracou6.csv") %>%
  # Sélection des colonnes utiles
  select(spName, CircCorr)
```

```
## # A tibble: 3,541 x 2
##   spName                    CircCorr
##   <chr>                        <dbl>
## 1 Pogonophora_schomburgkiana    44
## 2 Pogonophora_schomburgkiana    43.5
## 3 Licania_membranacea           53.5
## 4 Sandwithia_guyanensis         38.5
## 5 Eperua_falcata                77
## # i 3,536 more rows
```

Champs calculés

La surface terrière (en m^2) est calculée à partir de la circonférence (en cm).

```
# Lecture des données
read_csv2("data/Paracou6.csv") %>%
  # Sélection des colonnes utiles
  select(spName, CircCorr) %>%
  # Calcul de la surface terrière
  mutate(G = CircCorr^2 / 4 / pi / 10000)
```

```
## # A tibble: 3,541 x 3
##   spName                     CircCorr      G
##   <chr>                         <dbl>  <dbl>
## 1 Pogonophora_schomburgkiana     44   0.0154
## 2 Pogonophora_schomburgkiana     43.5 0.0151
## 3 Licania_membranacea            53.5 0.0228
## 4 Sandwithia_guyanensis          38.5 0.0118
## 5 Eperua_falcata                 77   0.0472
## # i 3,536 more rows
```

Statistiques synthétiques

On regroupe (`group_by()`) les individus par espèce et on calcule (`summarize()`) leur nombre et la somme des surfaces terrières.

```
# Lecture des données
read_csv2("data/Paracou6.csv") %>%
  # Sélection des colonnes utiles
  select(spName, CircCorr) %>%
  # Calcul de la surface terrière
  mutate(G = CircCorr^2 / 4 / pi / 10000) %>%
  # Grouper par espèce
  group_by(spName) %>%
  # Calculer le nombre de tiges et la surface terrière par ha
  summarize(Abondance = n(), Surface = sum(G) / 6.25)
```

```
## # A tibble: 335 x 3
##   spName               Abondance Surface
##   <chr>                    <int>   <dbl>
## 1 Abarema_jupunba             10 0.166
## 2 Abarema_mataybifolia         4 0.0214
## 3 Albizia_pedicellaris         3 0.109
## 4 Amaioua_guianensis           2 0.00471
## 5 Amanoa_congesta              1 0.00258
## # i 330 more rows
```

La fonction `n()` compte les lignes de chaque groupe. Les colonnes du tableau qui n'ont pas servi au regroupement sont éliminées: c'est la raison pour laquelle la sélection préalable des colonnes utiles `select()` est en général inutile.

Tri

Il reste à trier (`arrange()`) les espèces par surface terrière décroissante (appliquer la fonction `desc()` à la colonne).

```
# Lecture des données
read_csv2("data/Paracou6.csv") %>%
  # Sélection des colonnes utiles
  select(spName, CircCorr) %>%
  # Calcul de la surface terrière
  mutate(G = CircCorr^2 / 4 / pi / 10000) %>%
  # Grouper par espèce
  group_by(spName) %>%
  # Calculer le nombre de tiges et la surface terrière par ha
  summarize(Abondance = n(), G_ha = sum(G) / 6.25) %>%
  # Tri
  arrange(desc(G_ha))
```

```
## # A tibble: 335 x 3
##   spName                Abondance  G_ha
##   <chr>                     <int> <dbl>
## 1 Eperua_falcata              266  5.68
## 2 Eschweilera_sagotiana       151  1.69
## 3 Eperua_grandiflora           67  1.50
## 4 Vouacapoua_americana         91  1.44
## 5 Tapura_capitulifera          63  1.13
## # i 330 more rows
```

La surface de la parcelle est 6,25ha: la surface terrière totale est donc divisée par 6,25 pour obtenir sa valeur par hectare. L'espèce la plus abondante est le wacapou (*Eperua falcata*).

Stockage dans une variable

Le code complet ci-dessous enregistre le tableau final dans une variable.

```
# Lecture des données
read_csv2("data/Paracou6.csv") %>%
  # Sélection des colonnes utiles
  select(spName, CircCorr) %>%
  # Calcul de la surface terrière
  mutate(G = CircCorr^2 / 4 / pi / 10000) %>%
  # Grouper par espèce
  group_by(spName) %>%
  # Calculer le nombre de tiges et la surface terrière par ha
  summarize(Abondance = n(), G_ha = sum(G) / 6.25) %>%
  # Tri
  arrange(desc(G_ha)) ->
  # Variable de stockage du résultat
  paracou6_G_espece
```

La syntaxe la plus naturelle pour affecter le tableau à une variable utilise la flèche vers la droite à la fin du pipeline, dans le sens de la progression des données.

Extraction d'un résultat

Souvent, une valeur unique est attendue plutôt qu'un tableau.

Nous calculons maintenant le nombre d'arbres et la surface terrière par hectare toutes essences confondues. Le code est similaire au précédent avec des simplifications:

- la sélection des colonnes est omise,
- les arbres ne sont pas regroupés par espèce,
- le tri final n'a pas lieu d'être.

```
# Lecture des données
read_csv2("data/Paracou6.csv") %>%
  # Calcul de la surface terrière
  mutate(G = CircCorr^2 / 4 / pi / 10000) %>%
  # Calcul du nombre de tiges et la surface terrière par ha
  summarize(Densite = n() / 6.25, G_ha = sum(G) / 6.25)
```

```
## # A tibble: 1 x 2
##   Densite  G_ha
##     <dbl> <dbl>
## 1    567.  31.7
```

Pour extraire la surface terrière du tableau final et la placer dans une variable numérique (un vecteur à un seul élément), on utile la fonction `pluck()`. Attention:

- le tableau ne doit contenir qu'une seule ligne,

- le nom de la colonne est placé entre guillemets, contrairement à la syntaxe habituelle du tidyverse, par exemple dans la fonction `select()`.

```
# Lecture des données
read_csv2("data/Paracou6.csv") %>%
  # Calcul de la surface terrière
  mutate(G = CircCorr^2 / 4 / pi / 10000) %>%
  # Calculer la surface terrière par ha
  summarize(G_ha = sum(G) / 6.25) %>%
  # Extraire la valeur de G_ha
  pluck("G_ha") ->
  # Affectation à une variable numérique
  G_ha
# Affichage
G_ha
```

```
## [1] 31.68715
```

La valeur étant sauvegardée dans une variable, elle peut être incluse dans le texte d'un rapport: la surface terrière de la parcelle 6 de Paracou est égale à 31.7 m^2/ha.

Wickham, Hadley, Mine Çetinkaya-Rundel, et Garrett Grolemund. 2023. *R for data science*. 2nd éd. O'Reilly Media. http://r4ds.had.co.nz/.