Manipulation de données avec dplyr

Vincent Guillemot

Mardi 25 mai 2021

Enchaîner les commandes avec magrittr

- On utilise un opérateur

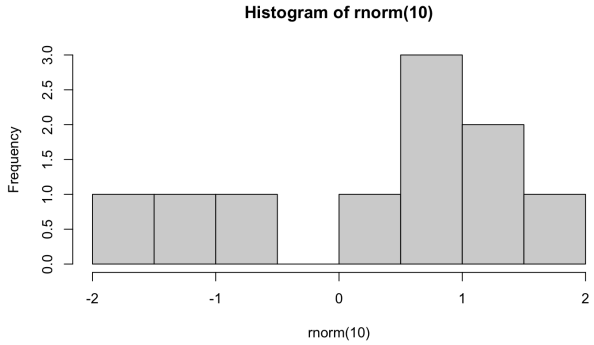
`%>%`

- Avant : `f(g(x))`
- Après : `g(x) %>% f()`

Exemple

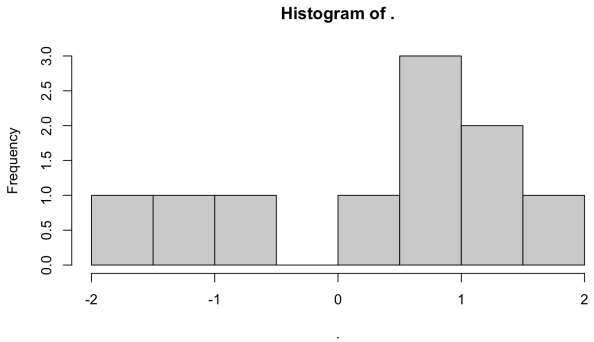
```
set.seed(7895)

hist(rnorm(10))
```




```
library(magrittr)
#>
#> Attachement du package : 'magrittr'
```

```
#> L'objet suivant est masqué depuis 'package:t
#>
#>     extract
set.seed(7895)
rnorm(10) %>% hist()
```



Le tidyverse


fig:
Tidyverse: https://www.tidyverse.org/

Avant toute chose

Charger le package `dplyr`...

```
library(dplyr) # ou require(dplyr)
```

Ou bien charger `tidyverse`...

```
library(tidyverse)
```

... mais cela chargera d'autres packages en plus

Et charger les données.

```
data("fruits", package = "tidyViz")
```

Le format “tibble”

Les données sont au format “tibble” : c’est comme des “data-frames” mais en mieux !

```
fruits
#> # A tibble: 51 × 18
#>    nom       groupe Energie   Eau Proteines Glucides Lipides
#>    <chr>     <chr>    <dbl> <dbl>     <dbl>    <dbl>   <dbl>
#>  1 Abricot   crus       194  87.1      0.81     9.01    0.25
#>  2 Abricot   secs      1010  24.7      2.88    59.1     0.5
#>  3 Ananas    exoti…     304  81.3      0.94    15.1     0.25
#>  4 Banane    exoti…     383  75.8      1.06    19.7     0.25
#>  5 Canneber… secs      1410  14.6      0.25    76.4     1
#>  6 Cerise    crus       235  85.7      0.81    13       0.25
#>  7 Citron    crus       118  91.3      0.25     1.56    0.25
#>  8 Clementi… crus       200  87        0.81     9.17    0.25
#>  9 CompoteM… compo…     279  82.9      0.25    15.3     0.08
#> 10 CompoteP… compo…     432  72.9      0.23    24.4     0.21
#> # … with 41 more rows, and 11 more variables: Sucres <dbl>,
#> #   Fructose <dbl>, Fibres <dbl>, Calcium <dbl>,
#> #   Magnesium <dbl>, Phosphore <dbl>, Potassium <dbl>,
#> #   Zinc <dbl>, BetaCarotene <dbl>, VitamineE <dbl>,
#> #   VitamineC <dbl>
```

Les fonctions de dplyr

Nous allons voir ensemble quelques fonctions très pratiques de la librairie `dplyr`.

#	Fonction (US)	Fonction (UK)	Description
1	`mutate`	`mutate`	Créer ou modifier des colonnes
2	`select`	`select`	Sélectionner des colonnes
3	`arrange`	`arrange`	Trier les lignes
4	`filter`	`filter`	Sélectionner des lignes
5	`group_by`	`group_by`	Grouper des lignes
6	`summarize`	`summarise`	Résumer des groupes

Créer ou modifier des colonnes

Avec la fonction `mutate`.

```
fruits2 <- fruits %>%
  mutate(Sucres_ratio = Sucres / 100)

head(fruits2[, "Sucres_ratio"])
#> # A tibble: 6 × 1
#>   Sucres_ratio
#>          <dbl>
#> 1        0.067
#> 2        0.343
#> 3        0.149
#> 4        0.156
#> 5        0.728
#> 6        0.1
```

Avec les fonctions classiques.

```
fruits2 <- fruits
fruits2$Sucres_ratio <-
  fruits2$Sucres / 100
head(fruits2[, "Sucres_ratio"])
#> # A tibble: 6 × 1
#>   Sucres_ratio
#>          <dbl>
#> 1        0.067
#> 2        0.343
#> 3        0.149
#> 4        0.156
#> 5        0.728
#> 6        0.1
```

Sélectionner des colonnes

Avec la fonction `select`.

```
fruits %>%
  select(
    Energie,
    Sucres,
    Lipides,
    Proteines)
```

Avec les fonctions classiques.

```
fruits[,
  c(
    "Energie",
    "Sucres",
    "Lipides",
    "Proteines")]
```

Trier des lignes

Avec les fonctions `arrange` et `desc`.

```
fruits %>%
  select(Energie, Sucres, Fibres) %>%
  arrange(desc(Fibres))
#> # A tibble: 51 × 3
#>    Energie Sucres Fibres
#>      <dbl>  <dbl>  <dbl>
#>  1    1010  34.3     8.3
#>  2     425   8.5     6.8
#>  3    1410  72.8     5.7
#>  4     198   6.1     5.2
#>  5     969  38.1     5.1
#>  6     289   6.63    4.6
#>  7     206   5.4     4.3
#>  8     170   2.1     4.3
#>  9    1360  70.3     4.2
#> 10     293  12.2     4.1
#> # … with 41 more rows
```

Avec les fonctions classiques

```
fruits[
  order(fruits$Fibres, decreasing = TRUE),
  c("Energie", "Sucres", "Fibres")]
#> # A tibble: 51 × 3
#>    Energie Sucres Fibres
#>      <dbl>  <dbl>  <dbl>
#>  1    1010  34.3     8.3
#>  2     425   8.5     6.8
#>  3    1410  72.8     5.7
#>  4     198   6.1     5.2
#>  5     969  38.1     5.1
#>  6     289   6.63    4.6
#>  7     206   5.4     4.3
#>  8     170   2.1     4.3
#>  9    1360  70.3     4.2
#> 10     293  12.2     4.1
#> # … with 41 more rows
```

Sélectionner des lignes

Avec la fonction `filter`.

```
fruits %>%
  filter(Sucres > 60)
#> # A tibble: 2 × 18
#>   nom        groupe Energie   Eau Proteines
#>   <chr>      <chr>    <dbl> <dbl>     <dbl>
#> 1 Canneberge secs      1410  14.6      0.25
#> 2 Raisin     secs      1360  16         3
#> # … with 11 more variables: Sucres <dbl>, Fr
#> #   Fibres <dbl>, Calcium <dbl>, Magnesium <
#> #   Phosphore <dbl>, Potassium <dbl>, Zinc <
#> #   BetaCarotene <dbl>, VitamineE <dbl>, Vitamine
```

Avec les fonctions classiques.

```
fruits[fruits$Sucres > 60, ]
#> # A tibble: 2 × 18
#>   nom        groupe Energie   Eau Proteines Glucides L
#>   <chr>      <chr>    <dbl> <dbl>     <dbl>    <dbl>
#> 1 Canneberge secs      1410  14.6      0.25     76.4
#> 2 Raisin     secs      1360  16         3       73.2
#> # … with 11 more variables: Sucres <dbl>, Fructose <db
#> #   Fibres <dbl>, Calcium <dbl>, Magnesium <dbl>,
#> #   Phosphore <dbl>, Potassium <dbl>, Zinc <dbl>,
#> #   BetaCarotene <dbl>, VitamineE <dbl>, VitamineC <db
```

Agréger des colonnes

Avec la fonction `group_by` :

```
fruits %>% group_by(groupe)
#> # A tibble: 51 × 18
#> # Groups:   groupe [4]
#>    nom       groupe Energie   Eau Proteines Glucides Lipides
#>    <chr>     <chr>    <dbl> <dbl>     <dbl>    <dbl>   <dbl>
#>  1 Abricot   crus       194  87.1      0.81     9.01    0.25
#>  2 Abricot   secs      1010  24.7      2.88    59.1     0.5
#>  3 Ananas    exoti…     304  81.3      0.94    15.1     0.25
#>  4 Banane    exoti…     383  75.8      1.06    19.7     0.25
#>  5 Canneber… secs      1410  14.6      0.25    76.4     1
#>  6 Cerise    crus       235  85.7      0.81    13       0.25
#>  7 Citron    crus       118  91.3      0.25     1.56    0.25
#>  8 Clementi… crus       200  87        0.81     9.17    0.25
#>  9 CompoteM… compo…     279  82.9      0.25    15.3     0.08
#> 10 CompoteP… compo…     432  72.9      0.23    24.4     0.21
#> # … with 41 more rows, and 11 more variables: Sucres <dbl>,
#> #   Fructose <dbl>, Fibres <dbl>, Calcium <dbl>,
#> #   Magnesium <dbl>, Phosphore <dbl>, Potassium <dbl>,
#> #   Zinc <dbl>, BetaCarotene <dbl>, VitamineE <dbl>,
#> #   VitamineC <dbl>
```

Les données sont prêtes à être “traitées” groupe par groupe. PS : L’opération `ungroup()` permet d’enlever les groupes.

Calculer une moyenne

Avec la fonction `summarize`.

```
fruits %>%
  group_by(groupe) %>%
  summarize(SucreMoyen = mean(Sucres))
#> # A tibble: 4 × 2
#>   groupe   SucreMoyen
#>   <chr>         <dbl>
#> 1 compote        15.5
#> 2 crus            9.68
#> 3 exotique       11.4
#> 4 secs           55.0
```

Avec les fonctions classiques.

```
aggregate(fruits$Sucres,
          by = list(fruits$groupe),
          FUN = mean)
#>    Group.1         x
#> 1  compote 15.533333
#> 2     crus  9.684242
#> 3 exotique 11.380000
#> 4     secs 54.980000
```

Exercice(s)

Calculer l’énergie moyenne, la teneur en sucres médiane et le maximum de la teneur en Fibres par groupe de fruits et trier le tout par ordre décroissant du maximum de la teneur en Fibres !

Deux autres fonctions

	Sélectionne	Ne sélectionne pas
Ne transforme pas	`select`	`rename`
Peut transformer	`transmute`	`mutate`

Mais il y en a tellement d’autres !