Travail DPLYR

vincent obertelli

12/6/2020

Nous allons voir dans ce document comment utiliser le package dplyr. Ce package nous sera utile pour traiter ou manipuler des bases de données. Ce package nous permettra de modifier des data.frame ou tibble (tableaux de données).

Pour commencer nous allons installer le package. Pour ce faire on peut soit installer tout tidyverse, soit juste dplyr.

install.packages("tidyverse")

install.packages("dplyr")

Pour notre exemple, nous allons utiliser une base de donnée qui contient les informations sur les vols des trois aéroports de New York en 2013 (après avoir installer le package, nycflights13) :

```
library(nycflights13)
## Chargement des trois tables du jeu de données (se sont des tibbles)
data(flights)
data(airports)
data(airlines)
```

Maintenant, nos 3 tibbles sont dans l'environnement de RStudio. Maintenant nous pouvons voir les fonctions proposées par ce package :

-La fonction Slice. Elle permet de séléctionner des lignes du tableau grâce à leur position.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
slice(flights, 14)
```

```
## # A tibble: 1 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
## 1  2013     1     1      558            600        -2      923            937
## # ... with 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

On peut aussi séléctionner la première ligne :

```
library(dplyr)
slice_head(airlines)
```

```
## # A tibble: 1 x 2
##   carrier name
##   <chr>   <chr>
## 1 9E      Endeavor Air Inc.
```

On peut séléctionner des lignes akéatoires :

```
library(dplyr)
slice_sample(airports)
```

```
## # A tibble: 1 x 8
##   faa   name                   lat   lon   alt    tz dst   tzone
##   <chr> <chr>                <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 BMI   Central Illinois Rgnl  40.5 -88.9   871    -6 A     America/Chicago
```

On peut aussi séléctionner un nombre de ligne compris entre deux valeurs :

```
library(dplyr)
slice(airports, 100:105)
```

```
## # A tibble: 6 x 8
##   faa   name                        lat    lon   alt    tz dst   tzone
##   <chr> <chr>                     <dbl>  <dbl> <dbl> <dbl> <chr> <chr>
## 1 ADW   Andrews Afb                38.8  -76.9   280    -5 A     America/New_Yo~
## 2 AET   Allakaket Airport          66.6 -153.    441    -9 A     America/Anchor~
## 3 AEX   Alexandria Intl            31.3  -92.5    89    -6 A     America/Chicago
## 4 AFE   Kake Airport               57.0 -134.    172    -9 A     America/Anchor~
## 5 AFW   Fort Worth Alliance Airp~  33.0  -97.3   722    -6 A     America/Chicago
## 6 AGC   Allegheny County Airport   40.4  -79.9  1252    -5 A     America/New_Yo~
```

Il est possible d'utiliser la fonction slice pour d'autre but, mais vous verrez cela par vous même en pratiquant. Pour l'instant, grâce à ces connaissances, il est possible de travailler avec slice.

-La fonction filter. Elle permet de séléctionner des lignes qui suivent une condition particulière. Par exemple filtrer les vols sur le mois et plus précisément sur le mois de décembre.

```
library(dplyr)
filter(flights, month == 12)
```

```
## # A tibble: 28,135 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013    12     1       13           2359        14      446            445
##  2  2013    12     1       17           2359        18      443            437
##  3  2013    12     1      453            500        -7      636            651
##  4  2013    12     1      520            515         5      749            808
##  5  2013    12     1      536            540        -4      845            850
##  6  2013    12     1      540            550       -10     1005           1027
##  7  2013    12     1      541            545        -4      734            755
##  8  2013    12     1      546            545         1      826            835
##  9  2013    12     1      549            600       -11      648            659
## 10  2013    12     1      550            600       -10      825            854
## # ... with 28,125 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

-La fonction select. Elle permet de séléctionner les colonnes du tableau. Par exemple si l'on veut la colonne année du tableau flights, on fait :

```
library(dplyr)
select(flights, year)
```

```
## # A tibble: 336,776 x 1
##     year
##    <int>
##  1  2013
##  2  2013
##  3  2013
##  4  2013
##  5  2013
##  6  2013
##  7  2013
##  8  2013
##  9  2013
## 10  2013
## # ... with 336,766 more rows
```

-La fonction mutate. Elle permet de créer de nouvelles colonnes. Par exemple, la table airports contient l'altitude de l'aéroport en pieds. Si on veut créer une nouvelle variable alt_m avec l'altitude en mètres, on peut faire :

```
library(dplyr)
airports <- mutate(airports, alt_m = alt / 3.2808)
select(airports, name, alt, alt_m)
```

```
## # A tibble: 1,458 x 3
##    name                            alt  alt_m
##    <chr>                         <dbl>  <dbl>
##  1 Lansdowne Airport              1044 318.
##  2 Moton Field Municipal Airport   264  80.5
##  3 Schaumburg Regional             801 244.
```

```
##  4 Randall Airport                   523 159.
##  5 Jekyll Island Airport              11   3.35
##  6 Elizabethton Municipal Airport  1593 486.
##  7 Williams County Airport           730 223.
##  8 Finger Lakes Regional Airport     492 150.
##  9 Shoestring Aviation Airfield     1000 305.
## 10 Jefferson County Intl             108  32.9
## # ... with 1,448 more rows
```

Maintenant, il vous est possible de travailler avec le package dplyr, nous venont de voir les fonctions de base. Il reste cependant beaucoup à voir sur ce package, rien ne vous empêche de chercher plus par vous-même.