

AstéRix_jean_souris_dplyr

Teddy Leandre

23/12/2020

I. Présentation

Dans ce dossier, nous chercherons à évaluer 12 travaux comportant du code R et des concepts mathématiques. L'évaluation se fera sur la base de 5 critères de notation qui, additionnés, donnera une note globale sur 20 points.

1. Qualité du visuel du document sur 4 points
2. Pluralité des fonctionnalités sur 4 points
3. Fonctionnement du code sur 4 points
4. Lisibilité du code sur 4 points
5. Qualité des explications sur 4 points

II. Description

Ici nous verrons l'outil `dplyr`, c'est un paquet qui permet d'effectuer certaines manipulations de données et donc un meilleur traitement. Le travail de Jean SOURIS se porte sur certaines des fonctionnalités principales du paquet.

[Le Github évalué](#)

a. Commentaire

Tout d'abord, il faut installer le package `dplyr` pour cette démonstration :

```
#install.packages("dplyr")  
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
```

Dplyr sert à la manipulation de bases de données sous forme de tableau, donc pouvoir les réarranger, les filtrer, les trier, plein beaucoup d'autres fonctions.

Mais, avant de pouvoir modifier une base de donnée, il faut en sélectionner une ; c'est pour cela que nous allons utiliser les fonctions ci-dessous :

```
#install.packages("nycflights13")  
library("nycflights13")
```

Après avoir installé le package contenant notre base de donnée, nous allons sélectionner 2 tableaux que nous utiliserons au cours de cette démonstration :

```
data(flights)  
data(airports)
```

Dans cette partie, nous allons voir 3 principaux verbes que nous pouvons utiliser sur dplyr.

Slice

Le premier verbe que nous allons voir est "slice" et permet globalement de sélectionner à notre guise différentes lignes d'un tableau afin de les afficher :

Nous allons afficher une certaine ligne de la colonne "airlines" et voir ce qui s'affiche :

```
slice(airports, 537)
```

```
## # A tibble: 1 x 8
##   faa   name          lat   lon   alt   tz dst   tzone
##   <chr> <chr>          <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 GGG   East Texas Rgnl  32.4 -94.7   365    -6 A   America/Chicago
```

Nous avons donc accès à plusieurs informations d'un aéroport précis, tel que ses coordonnées géographiques, à savoir latitude, longitude, même altitude mais aussi à son nom raccourcis et sa zone géographique.

La fonction `slice` nous permet également de sélectionner plusieurs lignes à la fois en utilisant un interval :

```
slice(airports, 9:27)
```

```
## # A tibble: 19 x 8
##   faa   name          lat   lon   alt   tz dst   tzone
##   <chr> <chr>          <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 0P2   Shoestring Aviation Airf~ 39.8 -76.6  1000   -5 U   America/New_Y~
## 2 0S9   Jefferson County Intl    48.1 -123.   108   -8 A   America/Los_A~
## 3 0W3   Harford County Airport   39.6 -76.2   409   -5 A   America/New_Y~
## 4 10C   Galt Field Airport       42.4 -88.4   875   -6 U   America/Chica~
## 5 17G   Port Bucyrus-Crawford Co~ 40.8 -83.0  1003   -5 A   America/New_Y~
## 6 19A   Jackson County Airport   34.2 -83.6   951   -5 U   America/New_Y~
## 7 1A3   Martin Campbell Field Ai~ 35.0 -84.3  1789   -5 A   America/New_Y~
## 8 1B9   Mansfield Municipal     42.0 -71.2   122   -5 A   America/New_Y~
## 9 1C9   Frazier Lake Airpark     54.0 -125.   152   -8 A   America/Vanco~
## 10 1CS   Clow International Airpo~ 41.7 -88.1   670   -6 U   America/Chica~
## 11 1G3   Kent State Airport       41.2 -81.4  1134   -5 A   America/New_Y~
## 12 1G4   Grand Canyon West Airport 35.9 -114.  4813   -7 A   America/Phoen~
## 13 1H2   Effingham Memorial Airpo~ 39.1 -88.5   585   -6 A   America/Chica~
## 14 10H   Fortman Airport          40.6 -84.4   885   -5 U   America/New_Y~
## 15 1RL   Point Roberts Airpark    49.0 -123.   10   -8 A   America/Los_A~
## 16 23M   Clarke CO                32.1 -88.4   320   -6 A   America/Chica~
## 17 24C   Lowell City Airport      43.0 -85.3   681   -5 A   America/New_Y~
## 18 24J   Suwannee County Airport   30.3 -83.0   104   -5 A   America/New_Y~
## 19 25D   Forest Lake Airport      45.2 -93.0   925   -6 A   America/Chica~
```

Ici, nous avons sélectionné les lignes 9 à 27 du tableau de données des aéroports.

Hormis la selection de lignes au choix d'un tableau, la fonction `slice` nous permet également d'en selectionner de manière aléatoire grâce au verbe “`slice_sample`” :

```
airports %>% slice_sample(n=6)
```

```
## # A tibble: 6 x 8
##   faa   name                lat   lon   alt   tz dst   tzone
##   <chr> <chr>                <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 FXE   Fort Lauderdale Executive 26.2 -80.2   13   -5 A   America/New_York
## 2 FYV   Drake Fld                 36.0 -94.2  1251  -6 A   America/Chicago
## 3 MSN   Dane Co Rgnl Truax Fld    43.1 -89.3   887  -6 A   America/Chicago
## 4 MNZ   Manassas                  38.7 -77.5   192  -5 A   America/New_York
## 5 ORF   Norfolk Intl              36.9 -76.2    26  -5 A   America/New_York
## 6 3G3   Wadsworth Municipal       41.0 -81.8   974  -5 A   America/New_York
```

```
slice(airports, 1:6)
```

```
## # A tibble: 6 x 8
##   faa   name                lat   lon   alt   tz dst   tzone
##   <chr> <chr>                <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 04G   Lansdowne Airport        41.1 -80.6  1044  -5 A   America/New_Y~
## 2 06A   Moton Field Municipal Airp~ 32.5 -85.7   264  -6 A   America/Chica~
## 3 06C   Schaumburg Regional      42.0 -88.1   801  -6 A   America/Chica~
## 4 06N   Randall Airport          41.4 -74.4   523  -5 A   America/New_Y~
## 5 09J   Jekyll Island Airport     31.1 -81.4    11  -5 A   America/New_Y~
## 6 0A9   Elizabethton Municipal Air~ 36.4 -82.2  1593  -5 A   America/New_Y~
```

Comme vous pouvez le constater, le premier tableau a généré aléatoirement 6 lignes du tableau aéroport, lorsque le second a sélectionné les 6 premières.

NB : Nous pouvons également tirer des lignes du tableau en partant du bas ou du haut grâce aux verbes “`slice_head`” et “`slice_tail`” :

```
airports %>% slice_head(n=3)
```

```
## # A tibble: 3 x 8
##   faa   name                lat   lon   alt   tz dst   tzone
##   <chr> <chr>                <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 04G   Lansdowne Airport      41.1 -80.6  1044   -5 A   America/New_Y~
## 2 06A   Moton Field Municipal Airp~ 32.5 -85.7   264   -6 A   America/Chica~
## 3 06C   Schaumburg Regional      42.0 -88.1   801   -6 A   America/Chica~
```

```
airports %>% slice_tail(n=3)
```

```
## # A tibble: 3 x 8
##   faa   name                lat   lon   alt   tz dst   tzone
##   <chr> <chr>                <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 ZWI   Wilmington Amtrak Station 39.7 -75.6    0   -5 A   America/New_York
## 2 ZWU   Washington Union Station 38.9 -77.0   76   -5 A   America/New_York
## 3 ZYP   Penn Station             40.8 -74.0   35   -5 A   America/New_York
```

De même, nous pouvons tirer au hasard 5% de lignes de notre tableau en utilisant la fonction “prop” tel que :

```
airports %>% slice_sample(prop = 0.05)
```

```
## # A tibble: 72 x 8
##   faa   name                lat   lon   alt   tz dst   tzone
##   <chr> <chr>                <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 003   Morgantown Airport      40.2 -75.9   600   -5 A   America/New_York
## 2 MKK   Molokai                 21.2 -157.   454  -10 N   Pacific/Honolulu
## 3 HKB   Healy River Airport      63.9 -149.   1263  -9 A   America/Anchorage
## 4 SCC   Deadhorse               70.2 -148.    64   -9 A   America/Anchorage
## 5 NGU   Norfolk Ns              36.9 -76.3    15   -5 A   America/New_York
## 6 MGE   Dobbins Arb             33.9 -84.5  1068   -5 A   America/New_York
## 7 OLS   Nogales Intl            31.4 -111.   3955  -7 A   America/Phoenix
## 8 FSM   Fort Smith Rgnl         35.3 -94.4   469   -6 A   America/Chicago
## 9 OQN   Brandywine Airport       39.6 -75.3   466   -5 A   America/New_York
## 10 MPV   Edward F Knapp State     44.2 -72.6  1165   -5 A   America/New_York
## # ... with 62 more rows
```

Il y a également des verbes tels que “slice_min” et “slice_max” qui prennent en compte un argument supplémentaire du tableau choisi afin de filtrer son choix. Par exemple, si je souhaite connaître les 7 aéroports étant le plus bas, donc ayant la plus faible altitude, j'utilise la fonction suivante :

```
airports %>% slice_max(alt, n=7)
```

```
## # A tibble: 7 x 8
##   faa   name                lat   lon   alt   tz dst   tzone
##   <chr> <chr>                <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 TEX   Telluride              38.0 -108.  9078   -7 A   America/Denver
## 2 TVL   Lake Tahoe Airport     38.9 -120.  8544   -8 A   America/Los_An~
## 3 ASE   Aspen Pitkin County Sardy~ 39.2 -107.  7820   -7 A   America/Denver
## 4 GUC   Gunnison - Crested Butte 38.5 -107.  7678   -7 A   America/Denver
## 5 BCE   Bryce Canyon          37.7 -112.  7590   -7 A   America/Denver
## 6 ALS   San Luis Valley Regional ~ 37.4 -106.  7539   -7 A   America/Denver
## 7 LAR   Laramie Regional Airport 41.3 -106.  7284   -7 A   America/Denver
```

```
summary(flights)
```

```
##      year      month      day      dep_time sched_dep_time
## Min.   :2013   Min.    : 1.000   Min.    : 1.00   Min.     : 1      Min.     : 106
## 1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 907     1st Qu.: 906
## Median :2013   Median : 7.000   Median :16.00   Median :1401     Median :1359
## Mean   :2013   Mean    : 6.549   Mean    :15.71   Mean     :1349     Mean     :1344
## 3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1744     3rd Qu.:1729
## Max.   :2013   Max.    :12.000   Max.    :31.00   Max.     :2400     Max.     :2359
##
##                      NA's    :8255
##   dep_delay   arr_time   sched_arr_time   arr_delay
## Min.      : -43.00   Min.      : 1      Min.      : 1      Min.      : -86.000
## 1st Qu.:  -5.00   1st Qu.:1104     1st Qu.:1124     1st Qu.: -17.000
## Median :  -2.00   Median :1535     Median :1556     Median :  -5.000
## Mean     : 12.64   Mean     :1502     Mean     :1536     Mean      :  6.895
## 3rd Qu.: 11.00   3rd Qu.:1940     3rd Qu.:1945     3rd Qu.: 14.000
## Max.     :1301.00   Max.     :2400     Max.     :2359     Max.     :1272.000
## NA's     :8255     NA's     :8713     NA's     :9430
##   carrier      flight      tailnum      origin
```

```
## Length:336776      Min.   : 1      Length:336776      Length:336776
## Class :character    1st Qu.: 553    Class :character    Class :character
## Mode :character     Median :1496   Mode :character     Mode :character
##                      Mean    :1972
##                      3rd Qu.:3465
##                      Max.    :8500
##
##      dest          air_time      distance      hour
## Length:336776      Min.    : 20.0   Min.    : 17      Min.    : 1.00
## Class :character    1st Qu.: 82.0   1st Qu.: 502     1st Qu.: 9.00
## Mode :character     Median :129.0   Median : 872     Median :13.00
##                      Mean    :150.7   Mean    :1040     Mean    :13.18
##                      3rd Qu.:192.0   3rd Qu.:1389     3rd Qu.:17.00
##                      Max.    :695.0   Max.    :4983     Max.    :23.00
##                      NA's    :9430
##      minute      time_hour
## Min.   : 0.00     Min.   :2013-01-01 05:00:00
## 1st Qu.: 8.00     1st Qu.:2013-04-04 13:00:00
## Median :29.00     Median :2013-07-03 10:00:00
## Mean   :26.23     Mean   :2013-07-03 05:22:54
## 3rd Qu.:44.00     3rd Qu.:2013-10-01 07:00:00
## Max.   :59.00     Max.   :2013-12-31 23:00:00
##
```

De même si je souhaite connaitre les 10 vols les plus courts effectués en 2013 :

```
flights %>% slice_min(distance, n=10)
```

```
## # A tibble: 50 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     7    27      NA           106         NA      NA           245
## 2  2013     1     3    2127         2129        -2     2222         2224
## 3  2013     1     4    1240         1200        40     1333         1306
## 4  2013     1     4    1829         1615       134     1937         1721
## 5  2013     1     4    2128         2129        -1     2218         2224
```

```
## 6 2013 1 5 1155 1200 -5 1241 1306
## 7 2013 1 6 2125 2129 -4 2224 2224
## 8 2013 1 7 2124 2129 -5 2212 2224
## 9 2013 1 8 2127 2130 -3 2304 2225
## 10 2013 1 9 2126 2129 -3 2217 2224
## # ... with 40 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 <dtm>
```

Select :

Le second verbe que nous allons utiliser s'intitule "select", et, comme son nom l'indique nous permet de sélectionner des données d'un tableau et plus précisément des collones de celui-ci tel que :

```
select(flights, origin, time_hour)
```

```
## # A tibble: 336,776 x 2
##   origin time_hour
##   <chr> <dtm>
## 1 EWR    2013-01-01 05:00:00
## 2 LGA    2013-01-01 05:00:00
## 3 JFK    2013-01-01 05:00:00
## 4 JFK    2013-01-01 05:00:00
## 5 LGA    2013-01-01 06:00:00
## 6 EWR    2013-01-01 05:00:00
## 7 EWR    2013-01-01 06:00:00
## 8 LGA    2013-01-01 06:00:00
## 9 JFK    2013-01-01 06:00:00
## 10 LGA    2013-01-01 06:00:00
## # ... with 336,766 more rows
```

Ici nous avons donc les collones nous indiquant l'origine et l'heure de nos vols.

Nommer toutes les colonnes peut paraître rébarbatif, nous pouvons donc sélectionner un interval contenant les colonnes que nous souhaitons tel que :


```
select(flights, dep_time:dep_delay)
```

```
## # A tibble: 336,776 x 3
##   dep_time sched_dep_time dep_delay
##   <int>      <int>      <dbl>
## 1     517         515         2
## 2     533         529         4
## 3     542         540         2
## 4     544         545        -1
## 5     554         600        -6
## 6     554         558        -4
## 7     555         600        -5
## 8     557         600        -3
## 9     557         600        -3
## 10    558         600        -2
## # ... with 336,766 more rows
```

Les colonnes situées entre “dep_time” et dep“delay” comprises sont donc affichées.

En revanche, si, avant le nom de chaque colonne nous faisons apparaître le symbole “-”, alors le tableau s’affiche entièrement en ayant soustrait les colonnes sélectionnées :

```
select(flights, -origin, -time_hour)
```

```
## # A tibble: 336,776 x 17
##   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     517         515         2     830        819
## 2  2013     1     1     533         529         4     850        830
## 3  2013     1     1     542         540         2     923        850
## 4  2013     1     1     544         545        -1    1004       1022
## 5  2013     1     1     554         600        -6     812        837
## 6  2013     1     1     554         558        -4     740        728
## 7  2013     1     1     555         600        -5     913        854
## 8  2013     1     1     557         600        -3     709        723
```

```
## 9 2013 1 1 557 600 -3 838 846
## 10 2013 1 1 558 600 -2 753 745
## # ... with 336,766 more rows, and 9 more variables: arr_delay <dbl>,
## # carrier <chr>, flight <int>, tailnum <chr>, dest <chr>, air_time <dbl>,
## # distance <dbl>, hour <dbl>, minute <dbl>
```

Il y a également la possibilité d'appliquer des sortes de filtres, ou des conditions à nos tableaux avec les termes "starts_with", "ends_with", "contains" ou encore "matches" :

```
select(airports, starts_with("A"))
```

```
## # A tibble: 1,458 x 1
##   alt
##   <dbl>
## 1 1044
## 2  264
## 3  801
## 4  523
## 5   11
## 6 1593
## 7  730
## 8  492
## 9 1000
## 10 108
## # ... with 1,448 more rows
```

Dans cet exemple, j'ai affiché la seule colonne de ma table "airports" qui commençait par un "a".

Rename :

Le troisième verbe que nous allons voir est un dérivé de select et se nomme "rename".

Il nous permet de choisir certaines colonnes et de les renommer afin qu'elle soit plus lisible.

Par exemple :

```
rename(airports, altitude = alt, time_zone = tzone)
```

```
## # A tibble: 1,458 x 8
##   faa   name      lat    lon altitude    tz dst  time_zone
##   <chr> <chr>      <dbl> <dbl>    <dbl> <dbl> <chr> <chr>
## 1 04G   Lansdowne Airport  41.1 -80.6    1044   -5 A   America/New_Yo~
## 2 06A   Moton Field Municipa~ 32.5 -85.7     264   -6 A   America/Chicago
## 3 06C   Schaumburg Regional  42.0 -88.1     801   -6 A   America/Chicago
## 4 06N   Randall Airport    41.4 -74.4     523   -5 A   America/New_Yo~
## 5 09J   Jekyll Island Airport 31.1 -81.4      11   -5 A   America/New_Yo~
## 6 0A9   Elizabethton Municip~ 36.4 -82.2    1593   -5 A   America/New_Yo~
## 7 0G6   Williams County Airp~ 41.5 -84.5     730   -5 A   America/New_Yo~
## 8 0G7   Finger Lakes Regiona~ 42.9 -76.8     492   -5 A   America/New_Yo~
## 9 0P2   Shoestring Aviation ~ 39.8 -76.6    1000   -5 U   America/New_Yo~
## 10 0S9   Jefferson County Intl 48.1 -123.     108   -8 A   America/Los_An~
## # ... with 1,448 more rows
```

Nous avons réussi à renommer 2 colonnes du tableau “airports” initiale.

Enfin, si les surnoms que nous souhaitons donner contiennent des espaces ou des caractères spéciaux tels que “é”, “è”, “ù”, etc, nous pouvons utiliser l’écriture ci-dessous :

```
rename(airports, "altitude du vol" = alt, "zone horaire" = tzone)
```

```
## # A tibble: 1,458 x 8
##   faa   name      lat    lon `altitude du vo~    tz dst  `zone horaire`
##   <chr> <chr>      <dbl> <dbl>    <dbl> <dbl> <chr> <chr>
## 1 04G   Lansdowne Ai~ 41.1 -80.6    1044   -5 A   America/New_Yo~
## 2 06A   Moton Field ~ 32.5 -85.7     264   -6 A   America/Chicago
## 3 06C   Schaumburg R~ 42.0 -88.1     801   -6 A   America/Chicago
## 4 06N   Randall Airp~ 41.4 -74.4     523   -5 A   America/New_Yo~
## 5 09J   Jekyll Islan~ 31.1 -81.4      11   -5 A   America/New_Yo~
## 6 0A9   Elizabethton~ 36.4 -82.2    1593   -5 A   America/New_Yo~
## 7 0G6   Williams Cou~ 41.5 -84.5     730   -5 A   America/New_Yo~
## 8 0G7   Finger Lakes~ 42.9 -76.8     492   -5 A   America/New_Yo~
## 9 0P2   Shoestring A~ 39.8 -76.6    1000   -5 U   America/New_Yo~
```

```
## 10 0S9 Jefferson Co~ 48.1 -123.  
## # ... with 1,448 more rows
```

```
108 -8 A America/Los_An~
```

b. Notation

Critère 1 : 4/4 Visuel agréable

Critère 2 : 4/4 Fonction bien expliqué

Critère 3 : 4/4 Le code fonctionne.

Critère 4 : 2/4 Présentation abrupte du paquet.

Critère 5 : 3/4 Explication assez claire.

III. Conclusion

Note globale de 17/20. Bon travail qui explique bien le package dplyr. Les commentaires auraient pu être mieux positionnés.