dplyrJean

Jean Souris

10/12/2020

Introduction :

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

```
#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 selectionner 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 seelctionner 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 selectionner à notre guise différentes lignes d'un tableau afin de les afficher :

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

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

faa	name	lat	lon	alt	tz	dst	tzone
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
GGG	East Texas Rgnl	32.38401	-94.71149	365	-6	A	America/Chicago

1 row

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

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

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

…	name	lat	lon	alt	tz	…	tzone
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
0P2	Shoestring Aviation Airfield	39.79482	-76.64719	1000	-5	U	America/Ne
0S9	Jefferson County Intl	48.05381	-122.81064	108	-8	A	America/Lo
0W3	Harford County Airport	39.56684	-76.20240	409	-5	A	America/Ne
10C	Galt Field Airport	42.40289	-88.37511	875	-6	U	America/Ch
17G	Port Bucyrus-Crawford County Airport	40.78156	-82.97481	1003	-5	A	America/Ne
19A	Jackson County Airport	34.17586	-83.56160	951	-5	U	America/Ne
1A3	Martin Campbell Field Airport	35.01581	-84.34683	1789	-5	A	America/Ne
1B9	Mansfield Municipal	42.00013	-71.19677	122	-5	A	America/Ne
1C9	Frazier Lake Airpark	54.01333	-124.76833	152	-8	A	America/Va
1CS	Clow International Airport	41.69597	-88.12923	670	-6	U	America/Ch

1-10 of 19 rows Previous **1** 2 Next

Ici, nous avons selectionné 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)
```

…	name	lat	lon	alt	tz	…	tzone
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
CDS	Childress Muni	34.43378	-100.28799	1954	-6	A	America/Chicago
KBW	Chignik Bay Seaplane Base	56.29556	-158.40140	0	-9	A	America/Anchorage
GED	Sussex Co	38.68919	-75.35889	50	-5	A	America/New_York
CLC	Clear Lake Metroport	29.55690	-95.13750	35	-6	A	America/Chicago
TVF	Thief River Falls	48.06556	-96.18500	1116	-6	A	America/Chicago

…	name	lat	lon	alt	tz	…	tzone
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
SPG	Albert Whitted	27.76511	-82.62697	7	-5	A	America/New_York

6 rows

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

…	name	lat	lon	alt	tz	…	tzone
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
04G	Lansdowne Airport	41.13047	-80.61958	1044	-5	A	America/New_York
06A	Moton Field Municipal Airport	32.46057	-85.68003	264	-6	A	America/Chicago
06C	Schaumburg Regional	41.98934	-88.10124	801	-6	A	America/Chicago
06N	Randall Airport	41.43191	-74.39156	523	-5	A	America/New_York
09J	Jekyll Island Airport	31.07447	-81.42778	11	-5	A	America/New_York
0A9	Elizabethton Municipal Airport	36.37122	-82.17342	1593	-5	A	America/New_York

6 rows

Comme vous pouvez le constater, le premier tableau a généré aléatoirement 6 lignes du tableua aéroport, lorsque le second a slectionné 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)
```

…	name	lat	lon	alt	tz	…	tzone
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
04G	Lansdowne Airport	41.13047	-80.61958	1044	-5	A	America/New_York
06A	Moton Field Municipal Airport	32.46057	-85.68003	264	-6	A	America/Chicago
06C	Schaumburg Regional	41.98934	-88.10124	801	-6	A	America/Chicago

3 rows

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

…	name	lat	lon	alt	tz	…	tzone
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
ZWI	Wilmington Amtrak Station	39.73667	-75.55167	0	-5	A	America/New_York
ZWU	Washington Union Station	38.89746	-77.00643	76	-5	A	America/New_York
ZYP	Penn Station	40.75050	-73.99350	35	-5	A	America/New_York

3 rows

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)
```

…	name	lat	lon	alt	tz	…
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<c
BKX	Brookings Regional Airport	44.30480	-96.81690	1648	-6	A
HYG	Hydaburg Seaplane Base	55.20639	-132.82833	0	-9	A
IRC	Circle City Airport	65.82778	-144.07611	613	-9	A
ZBP	Penn Station	39.30722	-76.61556	66	-5	A
GLS	Scholes Intl At Galveston	29.26532	-94.86041	6	-6	A
49A	Gilmer County Airport	34.62822	-84.52658	1486	-5	A
60J	Ocean Isle Beach Airport	33.90851	-78.43667	32	-5	U
CYT	Yakataga Airport	60.08190	-142.49361	12	-9	A
CLT	Charlotte Douglas Intl	35.21400	-80.94314	748	-5	A
DIK	Dickinson Theodore Roosevelt Regional Airport	46.79750	-102.80194	2592	-7	A

1-10 of 72 rows | 1-7 of 8 columns Previous **1** 2 3 4 5 6 … 8 Next

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

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

…	name	lat	lon	alt	tz	…	tzone
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
TEX	Telluride	37.95376	-107.9085	9078	-7	A	America/Denver
TVL	Lake Tahoe Airport	38.89389	-119.9953	8544	-8	A	America/Los_Ange
ASE	Aspen Pitkin County Sardy Field	39.22320	-106.8690	7820	-7	A	America/Denver
GUC	Gunnison - Crested Butte	38.53389	-106.9331	7678	-7	A	America/Denver
BCE	Bryce Canyon	37.70644	-112.1458	7590	-7	A	America/Denver
ALS	San Luis Valley Regional Airport	37.43500	-105.8667	7539	-7	A	America/Denver
LAR	Laramie Regional Airport	41.31210	-105.6750	7284	-7	A	America/Denver

7 rows

```
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 connaitres les 10 vols les plus courts effectués en 2013 :

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

y…	mo…	…	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time	arr_delay	c
<int>	<int>	<int>	<int>	<int>	<dbl>	<int>	<int>	<dbl>	<
2013	7	27	*NA*	106	*NA*	*NA*	245	*NA*	U
2013	1	3	2127	2129	-2	2222	2224	-2	E
2013	1	4	1240	1200	40	1333	1306	27	E
2013	1	4	1829	1615	134	1937	1721	136	E
2013	1	4	2128	2129	-1	2218	2224	-6	E

y… <int>	mo… <int>	… <int>	dep_time <int>	sched_dep_time <int>	dep_delay <dbl>	arr_time <int>	sched_arr_time <int>	arr_delay <dbl>	c <…
2013	1	5	1155	1200	-5	1241	1306	-25	E
2013	1	6	2125	2129	-4	2224	2224	0	E
2013	1	7	2124	2129	-5	2212	2224	-12	E
2013	1	8	2127	2130	-3	2304	2225	39	E
2013	1	9	2126	2129	-3	2217	2224	-7	E

1-10 of 50 rows | 1-10 of 19 columns Previous **1** 2 3 4 5 Next

Select :

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

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

origin <chr>	time_hour <dttm>
EWR	2013-01-01 05:00:00
LGA	2013-01-01 05:00:00
JFK	2013-01-01 05:00:00
JFK	2013-01-01 05:00:00
LGA	2013-01-01 06:00:00
EWR	2013-01-01 05:00:00
EWR	2013-01-01 06:00:00
LGA	2013-01-01 06:00:00
JFK	2013-01-01 06:00:00
LGA	2013-01-01 06:00:00

1-10 of 10,000 rows Previous **1** 2 3 4 5 6 … 1000 Next

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 selectionner un interval contenant les colonnes que nous souhaitons tel que :

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

dep_time <int>	sched_dep_time <int>	dep_delay <dbl>
517	515	2

dep_time <int>	sched_dep_time <int>	dep_delay <dbl>
533	529	4
542	540	2
544	545	-1
554	600	-6
554	558	-4
555	600	-5
557	600	-3
557	600	-3
558	600	-2

1-10 of 10,000 rows Previous **1** 2 3 4 5 6 … 1000 Next

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 selectionnées :

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

y… <int>	mo… <int>	… <int>	dep_time <int>	sched_dep_time <int>	dep_delay <dbl>	arr_time <int>	sched_arr_time <int>	arr_delay <dbl>	c <chr>
2013	1	1	517	515	2	830	819	11	U
2013	1	1	533	529	4	850	830	20	U
2013	1	1	542	540	2	923	850	33	A
2013	1	1	544	545	-1	1004	1022	-18	B
2013	1	1	554	600	-6	812	837	-25	D
2013	1	1	554	558	-4	740	728	12	U
2013	1	1	555	600	-5	913	854	19	B
2013	1	1	557	600	-3	709	723	-14	B
2013	1	1	557	600	-3	838	846	-8	B
2013	1	1	558	600	-2	753	745	8	A

1-10 of 10,000 rows | 1-10 of 17 columns Previous **1** 2 3 4 5 6 … 1000 Next

Il y a également la possibilité d’appliquer des sortes de filtres, ou des conditions à nos tableaux avec les tremes “starts_with”, “ends_with”, “contains” ou encore “matches” :

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

alt <dbl>
1044
264
801
523
11
1593
730
492
1000
108

1-10 of 1,458 rows Previous **1** 2 3 4 5 6 … 146 Next

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)
```

… name <chr>	lat <dbl>	lon <dbl>	altitude <dbl>	tz <db
04G Lansdowne Airport	41.13047	-80.61958	1044	-5
06A Moton Field Municipal Airport	32.46057	-85.68003	264	-6
06C Schaumburg Regional	41.98934	-88.10124	801	-6
06N Randall Airport	41.43191	-74.39156	523	-5
09J Jekyll Island Airport	31.07447	-81.42778	11	-5
0A9 Elizabethton Municipal Airport	36.37122	-82.17342	1593	-5
0G6 Williams County Airport	41.46731	-84.50678	730	-5
0G7 Finger Lakes Regional Airport	42.88356	-76.78123	492	-5
0P2 Shoestring Aviation Airfield	39.79482	-76.64719	1000	-5
0S9 Jefferson County Intl	48.05381	-122.81064	108	-8

1-10 of 1,458 rows | 1-7 of 8 columns Previous **1** 2 3 4 5 6 … 146 Next

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)
```

…	name	lat	lon	altitude du
	<chr>	<dbl>	<dbl>	<
04G	Lansdowne Airport	41.13047	-80.61958	
06A	Moton Field Municipal Airport	32.46057	-85.68003	
06C	Schaumburg Regional	41.98934	-88.10124	
06N	Randall Airport	41.43191	-74.39156	
09J	Jekyll Island Airport	31.07447	-81.42778	
0A9	Elizabethton Municipal Airport	36.37122	-82.17342	
0G6	Williams County Airport	41.46731	-84.50678	
0G7	Finger Lakes Regional Airport	42.88356	-76.78123	
0P2	Shoestring Aviation Airfield	39.79482	-76.64719	
0S9	Jefferson County Intl	48.05381	-122.81064	

1-10 of 1,458 rows | 1-7 of 8 columns Previous **1** 2 3 4 5 6 … 146 Next

Je tenais à remercier cette source pour sa grande aide ! Source (https://juba.github.io/tidyverse/10-dplyr.html#autres-fonctions-utiles)

Vous pouvez retrouver tous mes dossiers juste ici !

 Mon Github (https://github.com/jeansouris/PSBX)