

# DPLYR

## Chargement des packages dplyr et nycflights13

Tout d'abord pour dplyr, il faut telecharger le package via la commande suivante :

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

```
# NOUS ALLONS PRENDRE LE DATASHEET SUR LES VOLS INTERIEURS AU USA
rm(flights)
```

```
install.packages("tidyverse")
```

```
#install.packages("tidyverse")
#install.packages("nycflights13")
# chargement des librairies
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2      v purrr   0.3.4
## v tibble  3.0.4      v dplyr  1.0.2
## v tidyr   1.1.2      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.0
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
library(nycflights13)
# Nous allons imprimer le jeu de données de vols de nycflights13
head(flights)
```

```
## # A tibble: 6 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     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
## # ... 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 <dtm>
```

## Choosing columns: select, rename

```
# en plus d'utiliser simplement select () pour choisir des colonnes ...
flights %>% select(carrier, flight)
```

```
## # A tibble: 336,776 x 2
##   carrier flight
##   <chr>   <int>
## 1 UA      1545
## 2 UA      1714
## 3 AA      1141
## 4 B6       725
## 5 DL       461
## 6 UA      1696
## 7 B6       507
## 8 EV      5708
## 9 B6        79
## 10 AA      301
## # ... with 336,766 more rows
```

```
# ... on peut utiliser le signe - pour cacher des colonnes
flights %>% select(-month, -day)
```

```
## # A tibble: 336,776 x 17
##   year dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
##   <int> <int>         <int>    <dbl>   <int>         <int>    <dbl>
## 1 2013     517           515        2     830           819        11
## 2 2013     533           529        4     850           830        20
## 3 2013     542           540        2     923           850        33
## 4 2013     544           545       -1    1004          1022       -18
## 5 2013     554           600       -6     812           837       -25
## 6 2013     554           558       -4     740           728        12
## 7 2013     555           600       -5     913           854        19
## 8 2013     557           600       -3     709           723       -14
## 9 2013     557           600       -3     838           846        -8
## 10 2013     558           600       -2     753           745         8
## # ... with 336,766 more rows, and 10 more variables: carrier <chr>,
## #   flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
## #   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dtm>
```

```
# cacher une rangée de colonne
flights %>% select(-(dep_time:arr_delay))
# cacher toutes les colonnes qui contiennent un mot
flights %>% select(-contains("time"))
```

```
# Choisir une colonne qui permet de prendre des colonnes en utilisant un vecteur de caractères de noms de colonnes
cols <- c("carrier", "flight", "tailnum")
flights %>% select(one_of(cols))
```

```
## # A tibble: 336,776 x 3
##   carrier flight tailnum
##   <chr>     <int> <chr>
## 1 UA         1545 N14228
## 2 UA         1714 N24211
## 3 AA         1141 N619AA
## 4 B6          725 N804JB
## 5 DL          461 N668DN
## 6 UA        1696 N39463
## 7 B6          507 N516JB
## 8 EV        5708 N829AS
## 9 B6          79  N593JB
## 10 AA         301 N3ALAA
## # ... with 336,766 more rows
```

*# select () peut être utilisé pour renommer les colonnes, bien que toutes les colonnes non mentionnées soient supprimées*

```
flights %>% select(tail = tailnum)
```

```
## # A tibble: 336,776 x 1
##   tail
##   <chr>
## 1 N14228
## 2 N24211
## 3 N619AA
## 4 N804JB
## 5 N668DN
## 6 N39463
## 7 N516JB
## 8 N829AS
## 9 N593JB
## 10 N3ALAA
## # ... with 336,766 more rows
```

*# rename () fait la même chose, sauf que toutes les colonnes non mentionnées sont conservées*

```
flights %>% rename(tail = tailnum)
```

```
## # A tibble: 336,776 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     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 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tail <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dtm>
```

# Choix des lignes : filter, between, slice, sample\_n, top\_n, distinct

```
# filter() supporte l'utilisation de multiples conditions
flights %>% filter(dep_time >= 600, dep_time <= 605)
```

```
## # A tibble: 2,460 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     600           600         0     851           858
## 2  2013     1     1     600           600         0     837           825
## 3  2013     1     1     601           600         1     844           850
## 4  2013     1     1     602           610        -8     812           820
## 5  2013     1     1     602           605        -3     821           805
## 6  2013     1     2     600           600         0     814           749
## 7  2013     1     2     600           605        -5     751           818
## 8  2013     1     2     600           600         0     819           815
## 9  2013     1     2     600           600         0     846           846
## 10 2013     1     2     600           600         0     737           725
## # ... with 2,450 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>
```

```
# between() est une alternative concise pour déterminer si les valeurs numériques tombent dans une plage
flights %>% filter(between(dep_time, 600, 605))
# note latérale: is.na () peut également être utile lors du filtrage
flights %>% filter(!is.na(dep_time))
```

```
# slice() filtrer les lignes par position
flights %>% slice(1000:1005)
```

```
## # A tibble: 6 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     2     809           810        -1     950           948
## 2  2013     1     2     810           800         10    1008          1014
## 3  2013     1     2     811           815        -4    1100          1056
## 4  2013     1     2     811           815        -4    1126          1131
## 5  2013     1     2     811           820        -9     944           955
## 6  2013     1     2     815           815         0    1109          1128
## # ... 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 <dtm>
```

```
#Garde les trois premières rangées de ligne de chaque groupe
flights %>% group_by(month, day) %>% slice(1:3)
```

```
## # A tibble: 1,095 x 19
## # Groups:   month, day [365]
##   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     2      42          2359        43     518           442
## 5  2013     1     2     126          2250       156     233          2359
## 6  2013     1     2     458           500        -2     703           650
## 7  2013     1     3      32          2359        33     504           442
## 8  2013     1     3      50          2145       185     203          2311
## 9  2013     1     3     235          2359       156     700           437
## 10 2013     1     4      25          2359        26     505           442
## # ... with 1,085 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>
```

```
# echantillonner trois lignes de chaque groupe
flights %>% group_by(month, day) %>% sample_n(3)
```

```
## # A tibble: 1,095 x 19
## # Groups:   month, day [365]
##   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    1625          1550        35    2054          2050
## 2  2013     1     1    1742          1746        -4    2028          2052
## 3  2013     1     1    1327          1330        -3    1638          1655
## 4  2013     1     2    1732          1734        -2    1850          1852
## 5  2013     1     2    1719          1720        -1    1921          1915
## 6  2013     1     2    1714          1700        14    1947          1935
## 7  2013     1     3    1046          1050        -4    1223          1227
## 8  2013     1     3     755           800        -5     958          1011
## 9  2013     1     3     811           810         1    1053          1042
## 10 2013     1     4      NA          1830        NA      NA          2044
## # ... with 1,085 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>
```

```
# garder trois lignes de chaque groupe avec le dep_delay supérieur
flights %>% group_by(month, day) %>% top_n(3, dep_delay)
```

```
## # A tibble: 1,108 x 19
## # Groups:   month, day [365]
##   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     848           1835        853     1001           1950
## 2  2013     1     1    1815           1325        290     2120           1542
## 3  2013     1     1    2343           1724        379      314           1938
## 4  2013     1     2    1412            838        334     1710           1147
## 5  2013     1     2    1607           1030        337     2003           1355
## 6  2013     1     2    2131           1512        379     2340           1741
## 7  2013     1     3    2008           1540        268     2339           1909
## 8  2013     1     3    2012           1600        252     2314           1857
## 9  2013     1     3    2056           1605        291     2239           1754
## 10 2013     1     4    2058           1730        208         2           2110
## # ... with 1,098 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>
```

```
# trier également par dep_delay dans chaque groupe
flights %>% group_by(month, day) %>% top_n(3, dep_delay) %>% arrange(desc(dep_delay))
```

```
## # A tibble: 1,108 x 19
## # Groups:   month, day [365]
##   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     9     641            900       1301     1242           1530
## 2  2013     6    15    1432           1935       1137     1607           2120
## 3  2013     1    10    1121           1635       1126     1239           1810
## 4  2013     9    20    1139           1845       1014     1457           2210
## 5  2013     7    22     845           1600       1005     1044           1815
## 6  2013     4    10    1100           1900        960     1342           2211
## 7  2013     3    17    2321            810        911      135           1020
## 8  2013     6    27     959           1900        899     1236           2226
## 9  2013     7    22    2257            759        898      121           1026
## 10 2013    12     5     756           1700        896     1058           2020
## # ... with 1,098 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>
```

```
#Les lignes uniques peuvent être identifiées en utilisant unique () à partir de la base R
flights %>% select(origin, dest) %>% unique()
```

```
## # A tibble: 224 x 2
##   origin dest
##   <chr>  <chr>
## 1 EWR    IAH
## 2 LGA    IAH
## 3 JFK    MIA
## 4 JFK    BQN
## 5 LGA    ATL
## 6 EWR    ORD
## 7 EWR    FLL
## 8 LGA    IAD
## 9 JFK    MCO
## 10 LGA    ORD
## # ... with 214 more rows
```

```
# dplyr propose une alternative plus "efficace"
flights %>% select(origin, dest) %>% distinct()
# remarque: lors du chaînage, vous n'avez pas besoin d'inclure les parenthèses s'il n'y a pas d'arguments
flights %>% select(origin, dest) %>% distinct
```

## Ajouter de nouvelles variables : mutate, transmute, add\_rownames

```
# mutate () crée une nouvelle variable (et conserve toutes les variables existantes)
flights %>% mutate(speed = distance/air_time*60)
```

```
## # A tibble: 336,776 x 20
##   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 12 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>,
## #   speed <dbl>
```

```
# transmute() garde seulement les nouvelles variables
flights %>% transmute(speed = distance/air_time*60)
```

```
## # A tibble: 336,776 x 1
##   speed
##   <dbl>
## 1  370.
## 2  374.
## 3  408.
## 4  517.
## 5  394.
## 6  288.
## 7  404.
## 8  259.
## 9  405.
## 10 319.
## # ... with 336,766 more rows
```

```
# exemple data frame avec des noms de ligne
mtcars %>% head()
```

```
##           mpg  cyl  disp  hp drat    wt  qsec vs  am  gear  carb
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0   1    4    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0   1    4    4
## Datsun 710     22.8   4  108  93 3.85 2.320 18.61  1   1    4    1
## Hornet 4 Drive  21.4   6  258 110 3.08 3.215 19.44  1   0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0   0    3    2
## Valiant        18.1   6  225 105 2.76 3.460 20.22  1   0    3    1
```

```
# add_rownames() transforme le nom des lignes en variable explicite
mtcars %>% add_rownames("model") %>% head()
```

```
## Warning: `add_rownames()` is deprecated as of dplyr 1.0.0.
## Please use `tibble::rownames_to_column()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## # A tibble: 6 x 12
##   model      mpg  cyl  disp  hp drat    wt  qsec    vs    am  gear  carb
##   <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Mazda RX4      21     6   160   110  3.9   2.62  16.5     0     1     4     4
## 2 Mazda RX4 W~   21     6   160   110  3.9   2.88  17.0     0     1     4     4
## 3 Datsun 710     22.8    4   108    93  3.85  2.32  18.6     1     1     4     1
## 4 Hornet 4 Dr~   21.4    6   258   110  3.08  3.22  19.4     1     0     3     1
## 5 Hornet Spor~   18.7    8   360   175  3.15  3.44  17.0     0     0     3     2
## 6 Valiant       18.1    6   225   105  2.76  3.46  20.2     1     0     3     1
```

```
# side note: dplyr n'imprime plus les noms de lignes pour les trames de données locales
mtcars %>% tbl_df()
```

```
## Warning: `tbl_df()` is deprecated as of dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```



```
## # A tibble: 32 x 11
##   mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  21     6  160   110  3.9   2.62  16.5    0    1    4     4
## 2  21     6  160   110  3.9   2.88  17.0    0    1    4     4
## 3 22.8     4  108    93  3.85  2.32  18.6    1    1    4     1
## 4 21.4     6  258   110  3.08  3.22  19.4    1    0    3     1
## 5 18.7     8  360   175  3.15  3.44  17.0    0    0    3     2
## 6 18.1     6  225   105  2.76  3.46  20.2    1    0    3     1
## 7 14.3     8  360   245  3.21  3.57  15.8    0    0    3     4
## 8 24.4     4  147    62  3.69  3.19  20      1    0    4     2
## 9 22.8     4  141    95  3.92  3.15  22.9    1    0    4     2
## 10 19.2     6  168   123  3.92  3.44  18.3    1    0    4     4
## # ... with 22 more rows
```

## Fonction de groupe et de compte g: summarise, tally, count, group\_size, n\_groups, ungroup

```
# summarise() peut etre utiliser pour compter Le nombre de lignes de chaque groupe
flights %>% group_by(month) %>% summarise(cnt = n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 12 x 2
##   month  cnt
##   <int> <int>
## 1     1 27004
## 2     2 24951
## 3     3 28834
## 4     4 28330
## 5     5 28796
## 6     6 28243
## 7     7 29425
## 8     8 29327
## 9     9 27574
## 10    10 28889
## 11    11 27268
## 12    12 28135
```

```
# tally() et count() peuvent Le faire de manière plus précise
flights %>% group_by(month) %>% tally()
flights %>% count(month)
```

```
# vous pouvez trier par Le nombre
flights %>% group_by(month) %>% summarise(cnt = n()) %>% arrange(desc(cnt))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 12 x 2
##   month   cnt
##   <int> <int>
## 1     7 29425
## 2     8 29327
## 3    10 28889
## 4     3 28834
## 5     5 28796
## 6     4 28330
## 7     6 28243
## 8    12 28135
## 9     9 27574
## 10    11 27268
## 11     1 27004
## 12     2 24951
```

```
# tally () et count () ont un paramètre de tri à cet effet
flights %>% group_by(month) %>% tally(sort=TRUE)
flights %>% count(month, sort=TRUE)
```

```
# vous pouvez additionner une variable spécifique au lieu de simplement compter les lignes
flights %>% group_by(month) %>% summarise(dist = sum(distance))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 12 x 2
##   month   dist
##   <int>   <dbl>
## 1     1 27188805
## 2     2 24975509
## 3     3 29179636
## 4     4 29427294
## 5     5 29974128
## 6     6 29856388
## 7     7 31149199
## 8     8 31149334
## 9     9 28711426
## 10    10 30012086
## 11    11 28639718
## 12    12 29954084
```

```
#tally () et count () ont un paramètre wt à cet effet
flights %>% group_by(month) %>% tally(wt = distance)
flights %>% count(month, wt = distance)
```

```
#group_size () renvoie les décomptes sous forme de vecteur
flights %>% group_by(month) %>% group_size()
```

```
## [1] 27004 24951 28834 28330 28796 28243 29425 29327 27574 28889 27268 28135
```

```
# n_groups () rapporte simplement le nombre de groupes  
flights %>% group_by(month) %>% n_groups()
```

```
## [1] 12
```

```
# grouper par deux variables, résumer, organiser (la sortie est peut-être déroutante)  
flights %>% group_by(month, day) %>% summarise(cnt = n()) %>% arrange(desc(cnt)) %>% print(n  
= 40)
```

```
## `summarise()` regrouping output by 'month' (override with `.groups` argument)
```

```
## # A tibble: 365 x 3
## # Groups:   month [12]
##   month   day   cnt
##   <int> <int> <int>
##  1     11    27 1014
##  2      7    11 1006
##  3      7     8 1004
##  4      7    10 1004
##  5     12     2 1004
##  6      7    18 1003
##  7      7    25 1003
##  8      7    12 1002
##  9      7     9 1001
## 10      7    17 1001
## 11      7    31 1001
## 12      8     7 1001
## 13      8     8 1001
## 14      8    12 1001
## 15      7    22 1000
## 16      7    24 1000
## 17      8     1 1000
## 18      8     5 1000
## 19      8    15 1000
## 20     11    21 1000
## 21      7    15  999
## 22      7    19  999
## 23      7    26  999
## 24      7    29  999
## 25      8     2  999
## 26      8     9  999
## 27     11    22  999
## 28      8    16  998
## 29      7    23  997
## 30      7    30  997
## 31      8    14  997
## 32      7    16  996
## 33      8     6  996
## 34      8    19  996
## 35      9    13  996
## 36      9    26  996
## 37      9    27  996
## 38      4    15  995
## 39      6    20  995
## 40      6    26  995
## # ... with 325 more rows
```

```
# dégroupier () avant d'arranger pour organiser dans tous les groupes
flights %>% group_by(month, day) %>% summarise(cnt = n()) %>% ungroup() %>% arrange(desc(cnt))
```

```
## `summarise()` regrouping output by 'month' (override with `.groups` argument)
```

```
## # A tibble: 365 x 3
##   month   day   cnt
##   <int> <int> <int>
## 1     11    27  1014
## 2      7    11  1006
## 3      7     8  1004
## 4      7    10  1004
## 5     12     2  1004
## 6      7    18  1003
## 7      7    25  1003
## 8      7    12  1002
## 9      7     9  1001
## 10     7    17  1001
## # ... with 355 more rows
```

## Création de trames de données: data\_frame

`data_frame()` est un meilleur moyen que `data.frame()` pour créer des tableaux de données. Avantages de `data_frame()`:

- Vous pouvez utiliser des colonnes précédemment définies pour calculer de nouvelles colonnes.
- Il ne contraint jamais les types de colonnes.
- Il ne brouille jamais les noms de colonnes.
- Il n'ajoute jamais de noms de ligne.
- Il ne recycle que l'entrée de longueur 1.
- Il renvoie une trame de données locale (un `tbl_df`).

```
# data_frame()
data_frame(a = 1:6, b = a*2, c = 'string', 'd+e' = 1) %>% glimpse()
```

```
## Warning: `data_frame()` is deprecated as of tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## Rows: 6
## Columns: 4
## $ a      <int> 1, 2, 3, 4, 5, 6
## $ b      <dbl> 2, 4, 6, 8, 10, 12
## $ c      <chr> "string", "string", "string", "string", "string", "string"
## $ `d+e` <dbl> 1, 1, 1, 1, 1, 1
```

```
# data.frame()
data.frame(a = 1:6, c = 'string', 'd+e' = 1) %>% glimpse()
```

```
## Rows: 6
## Columns: 3
## $ a      <int> 1, 2, 3, 4, 5, 6
## $ c      <chr> "string", "string", "string", "string", "string", "string"
## $ d.e <dbl> 1, 1, 1, 1, 1, 1
```

# Joindre des tables (merging): left\_join, right\_join, inner\_join, full\_join, semi\_join, anti\_join

```
# creation deux si dataframes
(a <- data_frame(color = c("green","yellow","red"), num = 1:3))
```

```
## # A tibble: 3 x 2
##   color    num
##   <chr> <int>
## 1 green     1
## 2 yellow    2
## 3 red       3
```

```
(b <- data_frame(color = c("green","yellow","pink"), size = c("S","M","L")))
```

```
## # A tibble: 3 x 2
##   color  size
##   <chr> <chr>
## 1 green  S
## 2 yellow M
## 3 pink  L
```

```
# n'inclut que les observations trouvées à la fois dans "a" et "b" (joint automatiquement sur
les variables qui apparaissent dans les deux tableaux)
inner_join(a, b)
```

```
## Joining, by = "color"
```

```
## # A tibble: 2 x 3
##   color    num size
##   <chr> <int> <chr>
## 1 green     1 S
## 2 yellow    2 M
```

```
# inclure les observations trouvées dans «a» ou «b»
full_join(a, b)
```

```
## Joining, by = "color"
```

```
## # A tibble: 4 x 3
##   color    num size
##   <chr> <int> <chr>
## 1 green     1 S
## 2 yellow    2 M
## 3 red       3 <NA>
## 4 pink      NA L
```

```
# inclure les observations trouvées dans "a"
left_join(a, b)
```

```
## Joining, by = "color"
```

```
## # A tibble: 3 x 3
##   color    num size
##   <chr> <int> <chr>
## 1 green      1 S
## 2 yellow     2 M
## 3 red        3 <NA>
```

```
# inclure les observations trouvées dans "b"
right_join(a, b)
```

```
## Joining, by = "color"
```

```
## # A tibble: 3 x 3
##   color    num size
##   <chr> <int> <chr>
## 1 green      1 S
## 2 yellow     2 M
## 3 pink      NA L
```

```
# right_join (a, b) est identique à left_join (b, a) sauf pour l'ordre des colonnes
left_join(b, a)
```

```
## Joining, by = "color"
```

```
## # A tibble: 3 x 3
##   color size    num
##   <chr> <chr> <int>
## 1 green S         1
## 2 yellow M         2
## 3 pink  L         NA
```

```
# filtrer "a" pour seulement montrer les observation qui matchent avec "b"
semi_join(a, b)
```

```
## Joining, by = "color"
```

```
## # A tibble: 2 x 2
##   color    num
##   <chr> <int>
## 1 green      1
## 2 yellow     2
```

```
# filtrer "a" pour seulement montrer les observation qui matchent avec "b"
anti_join(a, b)
```

```
## Joining, by = "color"
```

```
## # A tibble: 1 x 2
##   color  num
##   <chr> <int>
## 1 red      3
```

```
# parfois les variables correspondantes n'ont pas de noms identiques
b <- b %>% rename(col = color)
# spécifier que la jointure doit se produire en faisant correspondre "couleur" dans "a" avec
"col" dans "b"
inner_join(a, b, by=c("color" = "col"))
```

```
## # A tibble: 2 x 3
##   color  num size
##   <chr> <int> <chr>
## 1 green     1 S
## 2 yellow    2 M
```

## Viewing more output: print, View

```
# indiquez que vous souhaitez voir plus de lignes
flights %>% print(n = 15)
```

```
## # A tibble: 336,776 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     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
## 11 2013     1     1     558             600          -2     849             851
## 12 2013     1     1     558             600          -2     853             856
## 13 2013     1     1     558             600          -2     924             917
## 14 2013     1     1     558             600          -2     923             937
## 15 2013     1     1     559             600          -1     941             910
## # ... with 336,761 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>
```



```
# spécifiez que vous voulez voir TOUTES les lignes (ne l'exécutez pas!)
flights %>% print(n = Inf)
```

```
#indiquez que vous souhaitez voir toutes les colonnes
flights %>% print(width = Inf)
```

```
## # A tibble: 336,776 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     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
##   arr_delay carrier flight tailnum origin dest air_time distance hour minute
##   <dbl> <chr>   <int> <chr>   <chr> <chr>   <dbl>   <dbl> <dbl> <dbl>
## 1      11 UA      1545 N14228 EWR   IAH     227    1400     5     15
## 2      20 UA      1714 N24211 LGA   IAH     227    1416     5     29
## 3      33 AA      1141 N619AA JFK   MIA     160    1089     5     40
## 4     -18 B6       725 N804JB JFK   BQN     183    1576     5     45
## 5     -25 DL       461 N668DN LGA   ATL     116     762     6      0
## 6      12 UA      1696 N39463 EWR   ORD     150     719     5     58
## 7      19 B6       507 N516JB EWR   FLL     158    1065     6      0
## 8     -14 EV      5708 N829AS LGA   IAD      53     229     6      0
## 9       -8 B6       79 N593JB JFK   MCO     140     944     6      0
## 10      8 AA       301 N3ALAA LGA   ORD     138     733     6      0
##   time_hour
##   <dtm>
## 1 2013-01-01 05:00:00
## 2 2013-01-01 05:00:00
## 3 2013-01-01 05:00:00
## 4 2013-01-01 05:00:00
## 5 2013-01-01 06:00:00
## 6 2013-01-01 05:00:00
## 7 2013-01-01 06:00:00
## 8 2013-01-01 06:00:00
## 9 2013-01-01 06:00:00
## 10 2013-01-01 06:00:00
## # ... with 336,766 more rows
```

```
# afficher jusqu'à 1000 lignes et toutes les colonnes
flights %>% View()
# définir l'option pour voir toutes les colonnes et moins de lignes
options(dplyr.width = Inf, dplyr.print_min = 6)
# réinitialiser les options (ou fermer simplement R)
options(dplyr.width = NULL, dplyr.print_min = 10)
```