Jouer avec les données

Thomas Chuffart

24 septembre 2018

dplyr Introduction

- Permet de manipuler des BDD assez larges
- Simple: tout est verbalisé
- Rapide: plus rapide que si l'on utilise les fonctions de base

```
library(dplyr)
```

Les données

```r
library(AER)
data("Fatalities")
```

US traffic fatalities panel data for the “lower 48”" US states (i.e., excluding Alaska and Hawaii), annually for 1982 through 1988. (NT = 336, 34 variables)

Les données

```r
data("Fatalities")
head(Fatalities, n = 3L)
```

```
##   state year spirits unemp   income   emppop  beertax baptist  mormon
## 1    al 1982    1.37  14.4 10544.15 50.69204 1.539379 30.3557 0.32829
## 2    al 1983    1.36  13.7 10732.80 52.14703 1.788991 30.3336 0.34341
## 3    al 1984    1.32  11.1 11108.79 54.16809 1.714286 30.3115 0.35924
##   drinkage     dry youngdrivers    miles breath jail service fatal nfatal
## 1       19 25.0063     0.211572 7233.887     no   no      no   839    146
## 2       19 22.9942     0.210768 7836.348     no   no      no   930    154
## 3       19 24.0426     0.211484 8262.990     no   no      no   932    165
##   sfatal fatal1517 nfatal1517 fatal1820 nfatal1820 fatal2124 nfatal2124
## 1     99        53          9        99         34       120         32
## 2     98        71          8       108         26       124         35
## 3     94        49          7       103         25       118         34
##    afatal     pop  pop1517  pop1820  pop2124 milestot unempus emppopus
## 1 309.438 3942002 208999.6 221553.4 290000.1    28516     9.7     57.8
## 2 341.834 3960008 202000.1 219125.5 290000.2    31032     9.6     57.9
## 3 304.872 3988992 197000.0 216724.1 288000.2    32961     7.5     59.5
##           gsp
## 1 -0.02212476
## 2  0.04655825
## 3  0.06279784
```

```r
df <- Fatalities
```

Les verbes: filter

filter sélectionne les ligne d'un df selon une condition:

```
filter(df, year == 1983 & state == "al")
```

```
##   state year spirits unemp  income   emppop  beertax baptist  mormon
## 1    al 1983    1.36  13.7 10732.8 52.14703 1.788991 30.3336 0.34341
##   drinkage     dry youngdrivers    miles breath jail service fatal nfatal
## 1       19 22.9942     0.210768 7836.348     no   no      no   930    154
##   sfatal fatal1517 nfatal1517 fatal1820 nfatal1820 fatal2124 nfatal2124
## 1     98        71          8       108         26       124         35
##    afatal     pop  pop1517  pop1820  pop2124 milestot unempus emppopus
## 1 341.834 3960008 202000.1 219125.5 290000.2    31032     9.6     57.9
##          gsp
## 1 0.04655825
```

Les verbes: filter

```r
filter(df, unemp == max(unemp))
```

```
##   state year spirits unemp   income  emppop   beertax baptist  mormon
## 1    wv 1983    0.93    18 10451.83 42.9932 0.4577901 1.57328 0.37133
##   drinkage dry youngdrivers    miles breath jail service fatal nfatal
## 1     18.5   0     0.195664 5958.217    yes  yes      no   425     93
##   sfatal fatal1517 nfatal1517 fatal1820 nfatal1820 fatal2124 nfatal2124
## 1     61        23          4        56         21        58         18
##   afatal     pop  pop1517  pop1820  pop2124 milestot unempus emppopus
## 1    153 1963003 96000.13 98565.66 134999.9    11696     9.6     57.9
##           gsp
## 1 -0.02779328
```

Les verbes: select

select sélectionne colonnes d'un df:

```
head(select(df, state, year, unemp))
```

```
##   state year unemp
## 1    al 1982  14.4
## 2    al 1983  13.7
## 3    al 1984  11.1
## 4    al 1985   8.9
## 5    al 1986   9.8
## 6    al 1987   7.8
```

Si on met un "-" devant, la colonne est supprimée.

Les verbes: select

On peut utiliser plein de fonctions associées à select:

- starts_width *c qui commence par*
- contains *c qui contient*
- c1:c2 *sélectionne toutes les colonnes entre c1 et c2*
- everything() *toutes les colonnes non sélectionnées*

Du coup, c'est trop bien pour trier et rechercher quelque chose dont on ne connait pas forcément le nom! On peut ensuite renommer une variable avec **rename**

Les verbes: mutate

mutate permet de créer de nouvelles colonnes dans le tableau de données

```
df <- mutate(df, total_fatal = fatal + nfatal + sfatal)
head(select(df,state,total_fatal,fatal, nfatal, sfatal), n = 3L)
```

```
##   state total_fatal fatal nfatal sfatal
## 1    al        1084   839    146     99
## 2    al        1182   930    154     98
## 3    al        1191   932    165     94
```

Les verbes: slice

slice: sélection les lignes du df selon leur position:

```
slice(df, 1:3)
```

```
##   state year spirits unemp   income   emppop  beertax baptist  mormon
## 1    al 1982    1.37  14.4 10544.15 50.69204 1.539379 30.3557 0.32829
## 2    al 1983    1.36  13.7 10732.80 52.14703 1.788991 30.3336 0.34341
## 3    al 1984    1.32  11.1 11108.79 54.16809 1.714286 30.3115 0.35924
##   drinkage     dry youngdrivers    miles breath jail service fatal nfatal
## 1       19 25.0063     0.211572 7233.887     no   no      no   839    146
## 2       19 22.9942     0.210768 7836.348     no   no      no   930    154
## 3       19 24.0426     0.211484 8262.990     no   no      no   932    165
##   sfatal fatal1517 nfatal1517 fatal1820 nfatal1820 fatal2124 nfatal2124
## 1     99        53          9        99         34       120         32
## 2     98        71          8       108         26       124         35
## 3     94        49          7       103         25       118         34
##    afatal     pop  pop1517  pop1820  pop2124 milestot unempus emppopus
## 1 309.438 3942002 208999.6 221553.4 290000.1    28516     9.7     57.8
## 2 341.834 3960008 202000.1 219125.5 290000.2    31032     9.6     57.9
## 3 304.872 3988992 197000.0 216724.1 288000.2    32961     7.5     59.5
##           gsp total_fatal
## 1 -0.02212476        1084
## 2  0.04655825        1182
## 3  0.06279784        1191
```

Les verbes: arrange

```r
head(arrange(df,unemp), n=3L)
```

```
##   state year spirits unemp   income   emppop   beertax baptist mormon
## 1    nh 1988    3.79   2.4 18704.52 70.83839 0.6496632     0.1    0.3
## 2    nh 1987    3.90   2.5 17906.00 71.26865 0.6750000     0.1    0.3
## 3    nh 1986    4.05   2.8 17132.10 69.96149 0.6965944     0.1    0.3
##   drinkage     dry youngdrivers    miles breath jail service fatal nfatal
## 1     21.0 0.15137     0.156750 8762.188    yes   no      no   166     22
## 2     21.0 0.15137     0.167907 8672.647    yes   no      no   179     25
## 3     20.5 0.15579     0.172651 8133.395    yes   no      no   172     27
##   sfatal fatal1517 nfatal1517 fatal1820 nfatal1820 fatal2124 nfatal2124
## 1     11        13          0        28          4        28          7
## 2     18        23          2        16          5        28          7
## 3     18        16          3        26          9        17          3
##   afatal     pop  pop1517  pop1820  pop2124 milestot unempus emppopus
## 1 43.716 1085003 48000.06 50999.96 71999.79     9507     5.5     62.3
## 2 53.200 1057001 48999.97 54999.83 66000.38     9167     6.2     61.5
## 3 48.037 1027000 48999.94 55000.00 65998.91     8353     7.0     60.7
##          gsp total_fatal
## 1 0.04980035         199
## 2 0.14236085         222
## 3 0.08774439         217
```

Ordonne de façon croissante. On peut aussi rajouter une variable.

```r
arrange(df,unemp, income)
```

Les verbes: arrange

```r
x <- arrange(df, unemp)
slice(x, 1:3)
```

```
##   state year spirits unemp   income   emppop   beertax baptist mormon
## 1    nh 1988    3.79   2.4 18704.52 70.83839 0.6496632     0.1    0.3
## 2    nh 1987    3.90   2.5 17906.00 71.26865 0.6750000     0.1    0.3
## 3    nh 1986    4.05   2.8 17132.10 69.96149 0.6965944     0.1    0.3
##   drinkage     dry youngdrivers    miles breath jail service fatal nfatal
## 1     21.0 0.15137     0.156750 8762.188    yes   no      no   166     22
## 2     21.0 0.15137     0.167907 8672.647    yes   no      no   179     25
## 3     20.5 0.15579     0.172651 8133.395    yes   no      no   172     27
##   sfatal fatal1517 nfatal1517 fatal1820 nfatal1820 fatal2124 nfatal2124
## 1     11        13          0        28          4        28          7
## 2     18        23          2        16          5        28          7
## 3     18        16          3        26          9        17          3
##   afatal     pop  pop1517  pop1820  pop2124 milestot unempus emppopus
## 1 43.716 1085003 48000.06 50999.96 71999.79     9507     5.5     62.3
## 2 53.200 1057001 48999.97 54999.83 66000.38     9167     6.2     61.5
## 3 48.037 1027000 48999.94 55000.00 65998.91     8353     7.0     60.7
##          gsp total_fatal
## 1 0.04980035         199
## 2 0.14236085         222
## 3 0.08774439         217
```

Le pipe

```r
library(magrittr)
```

- Permet d'enchainer plusieurs opérations sur le df
- Se note %>%
- Plus lisible que de tout emboiter.

Le pipe: exemple

```r
head(arrange(select(filter(df, state == "al"), year, unemp, income), unemp), n = 3L)
```

```
##   year unemp   income
## 1 1988   7.2 12368.62
## 2 1987   7.8 11944.00
## 3 1985   8.9 11332.63
```

```r
head(df %>% filter(state == "al") %>% select(year, unemp, income) %>% arrange(unemp), n = 3L)
```

```
##   year unemp   income
## 1 1988   7.2 12368.62
## 2 1987   7.8 11944.00
## 3 1985   8.9 11332.63
```

Des opérations bien utiles: group_by

group_by est votre plus grand allié. Elle permet de définir des groupes de lignes à partir d'une ou plusieurs colonnes:

```
df$year <- as.integer(levels(df$year))[df$year]
df %>% group_by(year)
```

```
## # A tibble: 336 x 35
## # Groups:   year [7]
##    state  year spirits unemp income emppop beertax baptist mormon drinkage
##    <fct> <int>   <dbl> <dbl>  <dbl>  <dbl>   <dbl>   <dbl>  <dbl>    <dbl>
##  1 al     1982    1.37 14.4  10544.   50.7   1.54    30.4   0.328     19
##  2 al     1983    1.36 13.7  10733.   52.1   1.79    30.3   0.343     19
##  3 al     1984    1.32 11.1  11109.   54.2   1.71    30.3   0.359     19
##  4 al     1985    1.28  8.90 11333.   55.3   1.65    30.3   0.376     19.7
##  5 al     1986    1.23  9.80 11662.   56.5   1.61    30.3   0.393     21
##  6 al     1987    1.18  7.80 11944    57.5   1.56    30.2   0.411     21
##  7 al     1988    1.17  7.20 12369.   56.8   1.50    30.2   0.430     21
##  8 az     1982    1.97  9.90 12309.   56.9   0.215    3.96  4.92      19
##  9 az     1983    1.90  9.10 12694.   57.6   0.206    3.89  4.83      19
## 10 az     1984    2.14  5    13266.   60.4   0.297    3.82  4.74      19
## # ... with 326 more rows, and 25 more variables: dry <dbl>,
## #   youngdrivers <dbl>, miles <dbl>, breath <fct>, jail <fct>,
## #   service <fct>, fatal <int>, nfatal <int>, sfatal <int>,
## #   fatal1517 <int>, nfatal1517 <int>, fatal1820 <int>, nfatal1820 <int>,
## #   fatal2124 <int>, nfatal2124 <int>, afatal <dbl>, pop <dbl>,
## #   pop1517 <dbl>, pop1820 <dbl>, pop2124 <dbl>, milestot <dbl>,
## #   unempus <dbl>, emppopus <dbl>, gsp <dbl>, total_fatal <int>
```

Des opérations bien utiles: group_by

```r
df <- df %>% group_by(year) %>% mutate(meanunemp = mean(unemp))
head(df)
```

```
## # A tibble: 6 x 36
## # Groups:   year [6]
##   state  year spirits unemp income emppop beertax baptist mormon drinkage
##   <fct> <int>   <dbl> <dbl>  <dbl>  <dbl>   <dbl>   <dbl>  <dbl>    <dbl>
## 1 al     1982    1.37 14.4  10544.   50.7    1.54    30.4  0.328     19
## 2 al     1983    1.36 13.7  10733.   52.1    1.79    30.3  0.343     19
## 3 al     1984    1.32 11.1  11109.   54.2    1.71    30.3  0.359     19
## 4 al     1985    1.28  8.90 11333.   55.3    1.65    30.3  0.376     19.7
## 5 al     1986    1.23  9.80 11662.   56.5    1.61    30.3  0.393     21
## 6 al     1987    1.18  7.80 11944    57.5    1.56    30.2  0.411     21
## # ... with 26 more variables: dry <dbl>, youngdrivers <dbl>, miles <dbl>,
## #   breath <fct>, jail <fct>, service <fct>, fatal <int>, nfatal <int>,
## #   sfatal <int>, fatal1517 <int>, nfatal1517 <int>, fatal1820 <int>,
## #   nfatal1820 <int>, fatal2124 <int>, nfatal2124 <int>, afatal <dbl>,
## #   pop <dbl>, pop1517 <dbl>, pop1820 <dbl>, pop2124 <dbl>,
## #   milestot <dbl>, unempus <dbl>, emppopus <dbl>, gsp <dbl>,
## #   total_fatal <int>, meanunemp <dbl>
```

```r
df$meanunemp[0:8]
```

```
## [1] 9.266667 9.270833 7.233333 7.060417 6.918750 6.220833 5.456250 9.266667
```

Des opérations bien utiles: group_by

```r
df %>% group_by(year) %>% filter(jail == "no")
```

```
## # A tibble: 241 x 36
## # Groups:   year [7]
##    state  year spirits unemp income emppop beertax baptist mormon drinkage
##    <fct> <int>   <dbl> <dbl>  <dbl>  <dbl>   <dbl>   <dbl>  <dbl>    <dbl>
##  1 al     1982    1.37 14.4  10544.   50.7   1.54     30.4  0.328     19
##  2 al     1983    1.36 13.7  10733.   52.1   1.79     30.3  0.343     19
##  3 al     1984    1.32 11.1  11109.   54.2   1.71     30.3  0.359     19
##  4 al     1985    1.28  8.90 11333.   55.3   1.65     30.3  0.376     19.7
##  5 al     1986    1.23  9.80 11662.   56.5   1.61     30.3  0.393     21
##  6 al     1987    1.18  7.80 11944    57.5   1.56     30.2  0.411     21
##  7 al     1988    1.17  7.20 12369.   56.8   1.50     30.2  0.430     21
##  8 ar     1982    1.19  9.80 10267.   54.5   0.650    23.0  0.328     21
##  9 ar     1983    1.20 10.1  10433.   53.8   0.675    23.0  0.343     21
## 10 ar     1984    1.22  8.90 10916.   54.7   0.599    23.0  0.359     21
## # ... with 231 more rows, and 26 more variables: dry <dbl>,
## #   youngdrivers <dbl>, miles <dbl>, breath <fct>, jail <fct>,
## #   service <fct>, fatal <int>, nfatal <int>, sfatal <int>,
## #   fatal1517 <int>, nfatal1517 <int>, fatal1820 <int>, nfatal1820 <int>,
## #   fatal2124 <int>, nfatal2124 <int>, afatal <dbl>, pop <dbl>,
## #   pop1517 <dbl>, pop1820 <dbl>, pop2124 <dbl>, milestot <dbl>,
## #   unempus <dbl>, emppopus <dbl>, gsp <dbl>, total_fatal <int>,
## #   meanunemp <dbl>
```

Des opérations bien utiles: group_by et arrange

```
df %>% group_by(year) %>%  arrange(desc(unemp), .by_group = TRUE)
```

```
## # A tibble: 336 x 36
## # Groups:   year [7]
##    state  year spirits unemp income emppop beertax baptist mormon drinkage
##    <fct> <int>   <dbl> <dbl>  <dbl>  <dbl>   <dbl>   <dbl>  <dbl>    <dbl>
##  1 mi     1982    1.88  15.5 13247.   53.5   0.546   0.625  0.200       21
##  2 al     1982    1.37  14.4 10544.   50.7   1.54   30.4    0.328       19
##  3 wv     1982    1.05  13.9 10748.   45.5   0.476   1.55   0.381       18
##  4 oh     1982    1.27  12.5 13039.   55.6   0.430   1.51   0.200       21
##  5 wa     1982    1.90  12.1 14342.   56.3   0.232   1.15   2.66        21
##  6 in     1982    1.44  11.9 12283.   56.4   0.309   1.80   0.328       21
##  7 tn     1982    1.35  11.8 10988.   54.0   0.338  25.8    0.200       19
##  8 or     1982    1.68  11.5 12626.   58.5   0.225   1.02   2.80        21
##  9 il     1982    2.04  11.3 14743.   58.1   0.189   2.35   0.233       21
## 10 ms     1982    1.5   11    9554.   52.2   1.15   30.1    0.300       21
## # ... with 326 more rows, and 26 more variables: dry <dbl>,
## #   youngdrivers <dbl>, miles <dbl>, breath <fct>, jail <fct>,
## #   service <fct>, fatal <int>, nfatal <int>, sfatal <int>,
## #   fatal1517 <int>, nfatal1517 <int>, fatal1820 <int>, nfatal1820 <int>,
## #   fatal2124 <int>, nfatal2124 <int>, afatal <dbl>, pop <dbl>,
## #   pop1517 <dbl>, pop1820 <dbl>, pop2124 <dbl>, milestot <dbl>,
## #   unempus <dbl>, emppopus <dbl>, gsp <dbl>, total_fatal <int>,
## #   meanunemp <dbl>
```

Des opérations bien utiles: sample_n et sample_frac

- **sample_n** et **sample_frac** sélectionne un *n* de ligne ou une *frac* des lignes d'un tableau aléatoire.

```
df %>% ungroup()
```

```
df %>% ungroup() %>% sample_frac(0.01)
```

```
## # A tibble: 3 x 36
##   state  year spirits unemp income emppop beertax baptist mormon drinkage
##   <fct> <int>   <dbl> <dbl>  <dbl>  <dbl>   <dbl>   <dbl>  <dbl>    <dbl>
## 1 mn     1984    2.06  6.30 14734.   67.2   0.319   0.100  0.200       19
## 2 ga     1982    1.94  7.80 11774.   60.1   2.72   24.7    0.426       19
## 3 mo     1985    1.35  6.40 14034.   61.1   0.308  14.7    0.588       21
## # ... with 26 more variables: dry <dbl>, youngdrivers <dbl>, miles <dbl>,
## #   breath <fct>, jail <fct>, service <fct>, fatal <int>, nfatal <int>,
## #   sfatal <int>, fatal1517 <int>, nfatal1517 <int>, fatal1820 <int>,
## #   nfatal1820 <int>, fatal2124 <int>, nfatal2124 <int>, afatal <dbl>,
## #   pop <dbl>, pop1517 <dbl>, pop1820 <dbl>, pop2124 <dbl>,
## #   milestot <dbl>, unempus <dbl>, emppopus <dbl>, gsp <dbl>,
## #   total_fatal <int>, meanunemp <dbl>
```

Des opérations bien utiles: summarise

summarise permet d'agréger les lignes du df en effectuant un **summary** sur une ou plusieurs colonnes.

```
df %>% group_by(year) %>% summarise(mean_unemp = mean(unemp))
```

```
## # A tibble: 7 x 2
##    year mean_unemp
##   <int>      <dbl>
## 1  1982       9.27
## 2  1983       9.27
## 3  1984       7.23
## 4  1985       7.06
## 5  1986       6.92
## 6  1987       6.22
## 7  1988       5.46
```

Des opérations bien utiles: summarise

```r
df %>% group_by(year) %>% summarise(nb =  n())
```

```
## # A tibble: 7 x 2
##    year    nb
##   <int> <int>
## 1  1982    48
## 2  1983    48
## 3  1984    48
## 4  1985    48
## 5  1986    48
## 6  1987    48
## 7  1988    48
```

Des opérations bien utiles: count

count permet de compter le nombre de ligne par groupe, plus rapide que précédement:

```r
df %>% count(year)
```

```
## # A tibble: 7 x 2
## # Groups:   year [7]
##    year     n
##   <int> <int>
## 1  1982    48
## 2  1983    48
## 3  1984    48
## 4  1985    48
## 5  1986    48
## 6  1987    48
## 7  1988    48
```

Des opérations bien utiles: quelques remarques

- On peut grouper selon plusieurs variables
- Compter selon plusieurs variables
- **ungroup()** permet de dégrouper, cela peut servir dans certains cas

Des opérations bien utiles: lead et lag

- **lead** et **lag** permettent de décalier les observations

$$y_t = \phi y_{t-1} + \varepsilon_t \tag{1}$$

```
## [1] -0.9709506  0.8273073  0.6082210
```

```
## [1]         NA -0.9709506  0.8273073
```

Finalement

- Si vous avez 3 dataframe (taux crimes par région en France par année), vous pouvez utilisz dplyr pour les merger
- **bind_rows**, **bind_cols**