Mise en forme et ingénierie des données

Daniel Pont

08/06/2020

Contents

Dans ce paragraphe, on souhaite étudier

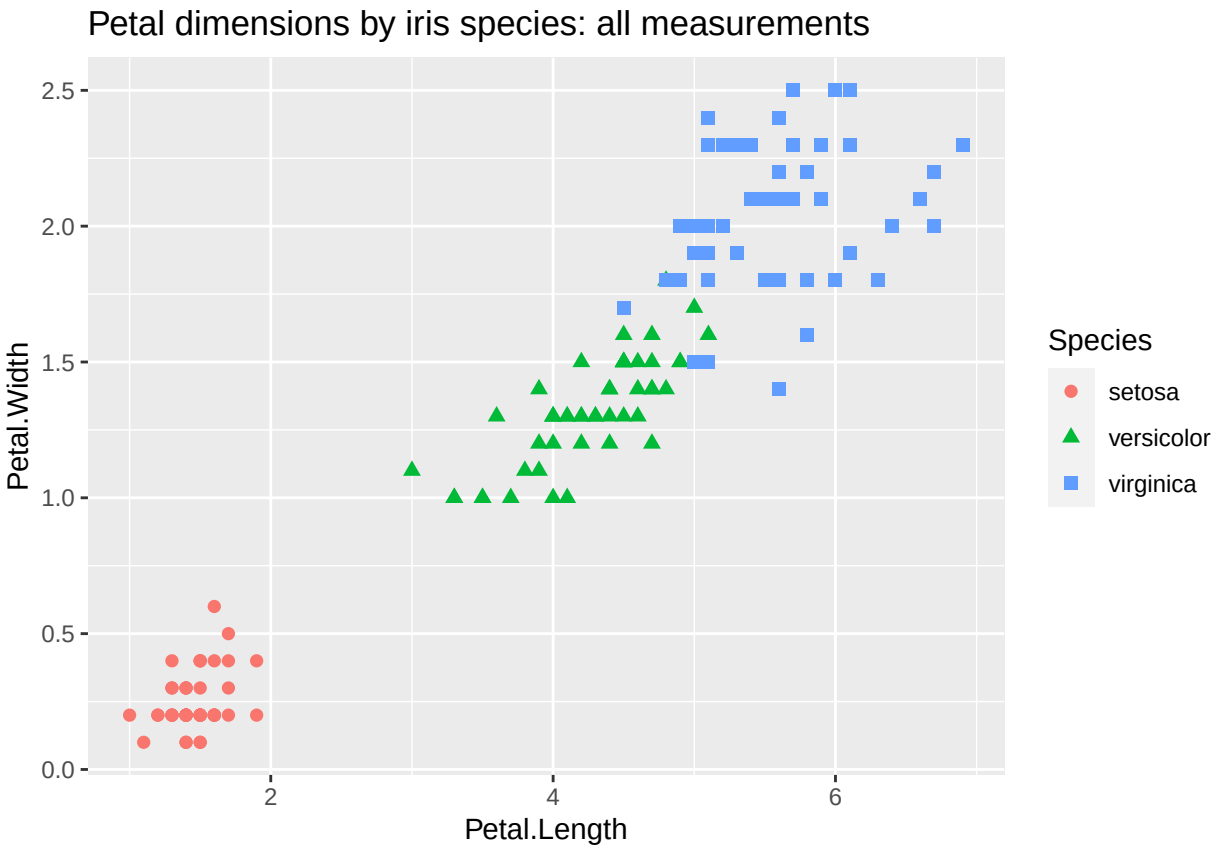
```
library(ggplot2)
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
ggplot(iris, aes(x=Petal.Length,y=Petal.Width,
                 shape= Species, color = Species)) +
    geom_point(size=2) +
    ggtitle("Petal dimensions by iris species: all measurements")
```




```
columns_we_want <- c("Petal.Length", "Petal.Width", "Species")
rows_we_want <- iris$Petal.Length > 2

#AVANT
head(iris)
```

1.1.1 Avec les méthodes de base en R

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
#APRES
#(drop=false permet d'obtenir un data.frame et non un vector si on ne sélectionne qu'une seule colonne)
iris_base <- iris[rows_we_want,columns_we_want,drop=FALSE]
head(iris_base)
```

```
##    Petal.Length Petal.Width    Species
## 51          4.7         1.4 versicolor
## 52          4.5         1.5 versicolor
## 53          4.9         1.5 versicolor
## 54          4.0         1.3 versicolor
```

```
## 55          4.6         1.5 versicolor
## 56          4.5         1.3 versicolor
```

1.1.2 Avec une data.table Quelques points clés sur les data.tables :

- ce sont des méthodes d'indexation puissantes (ex. : ".." ci-dessous)
- elles constitient la solution R la plus efficace en terme de rapidité et de de mémoire pour une large plage d'échelles
- FAQ : https://cran.r-project.org/web/packages/data.table/vignettes/datatable-faq.html
- Cheat sheet : https://www.datacamp.com/community/tutorials/data-table-cheat-sheet

```
library(data.table)
iris_data.table <- as.data.table(iris)
columns_we_want <- c("Petal.Length", "Petal.Width", "Species")
rows_we_want <- iris$Petal.Length > 2
# .. indique que columns_we_want n'est pas un nom de colonne mais une varaible contenant les colonnes
iris_data.table <- iris_data.table[rows_we_want,..columns_we_want]
head(iris_data.table)
```

```
##    Petal.Length Petal.Width    Species
## 1:          4.7         1.4 versicolor
## 2:          4.5         1.5 versicolor
## 3:          4.9         1.5 versicolor
## 4:          4.0         1.3 versicolor
## 5:          4.6         1.5 versicolor
## 6:          4.5         1.3 versicolor
```

Pour mieux comprendre la notation "..", voici un exemple :

```
library(data.table)
df <- data.frame(x=1:2,y=3:4)
# ERREUR (x non défini) :
#df[,x]

#Fonctionnement avec une data.table :
dt <- data.table(df)
x <- "y"

# sélectionne la colonne "x"
dt[,x]
```

```
## [1] 1 2
```

```
# sélectionne la colonne "y"
dt[,..x]
```

```
##    y
## 1: 3
## 2: 4
```

NB : Avec les packages qui ne les supportent pas, les data.tables se comportent comme des data.frames.

1.1.3 Avec dplyr Pour sélectionner :

- des *colonnes*, on utilise *dplyr::select*
- des *lignes*, on utilise *dplyr::filter*

```r
library(dplyr)

iris_dplyr <- iris %>%
               select( Petal.Length, Petal.Width, Species)  %>%
               filter( iris$Petal.Length > 2)
# NB : on peut aussi utiliser  select( c("Petal.Length", "Petal.Width", "Species"))

head(iris_dplyr)
```

```
##   Petal.Length Petal.Width    Species
## 1          4.7         1.4 versicolor
## 2          4.5         1.5 versicolor
## 3          4.9         1.5 versicolor
## 4          4.0         1.3 versicolor
## 5          4.6         1.5 versicolor
## 6          4.5         1.3 versicolor
```

1.2 Suppression des enregistrements avec des données manquantes

```r
library(ggplot2)
data(msleep)
str(msleep)
```

```
## tibble [83 x 11] (S3: tbl_df/tbl/data.frame)
##  $ name        : chr [1:83] "Cheetah" "Owl monkey" "Mountain beaver" "Greater short-tailed shrew" ..
##  $ genus       : chr [1:83] "Acinonyx" "Aotus" "Aplodontia" "Blarina" ...
##  $ vore        : chr [1:83] "carni" "omni" "herbi" "omni" ...
##  $ order       : chr [1:83] "Carnivora" "Primates" "Rodentia" "Soricomorpha" ...
##  $ conservation: chr [1:83] "lc" NA "nt" "lc" ...
##  $ sleep_total : num [1:83] 12.1 17 14.4 14.9 4 14.4 8.7 7 10.1 3 ...
##  $ sleep_rem   : num [1:83] NA 1.8 2.4 2.3 0.7 2.2 1.4 NA 2.9 NA ...
##  $ sleep_cycle : num [1:83] NA NA NA 0.133 0.667 ...
##  $ awake       : num [1:83] 11.9 7 9.6 9.1 20 9.6 15.3 17 13.9 21 ...
##  $ brainwt     : num [1:83] NA 0.0155 NA 0.00029 0.423 NA NA NA 0.07 0.0982 ...
##  $ bodywt      : num [1:83] 50 0.48 1.35 0.019 600 ...
```

```r
summary(msleep)
```

```
##      name              genus               vore              order
##  Length:83          Length:83          Length:83          Length:83
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  conservation        sleep_total      sleep_rem      sleep_cycle
##  Length:83          Min.   : 1.90   Min.   :0.100   Min.   :0.1167
##  Class :character   1st Qu.: 7.85   1st Qu.:0.900   1st Qu.:0.1833
##  Mode  :character   Median :10.10   Median :1.500   Median :0.3333
##                     Mean   :10.43   Mean   :1.875   Mean   :0.4396
##                     3rd Qu.:13.75   3rd Qu.:2.400   3rd Qu.:0.5792
##                     Max.   :19.90   Max.   :6.600   Max.   :1.5000
##                                     NA's   :22      NA's   :51
```

```
##      awake           brainwt            bodywt       
##  Min.   : 4.10   Min.   :0.00014   Min.   :   0.005  
##  1st Qu.:10.25   1st Qu.:0.00290   1st Qu.:   0.174  
##  Median :13.90   Median :0.01240   Median :   1.670  
##  Mean   :13.57   Mean   :0.28158   Mean   : 166.136  
##  3rd Qu.:16.15   3rd Qu.:0.12550   3rd Qu.:  41.750  
##  Max.   :22.10   Max.   :5.71200   Max.   :6654.000  
##                  NA's   :27
```

1.2.1 Avec les méthodes de base en R