Le {tidyverse} : {dplyr} et {tidyr}

Marie VAUGOYEAU

07/02/2023

Contents

Introduction

Historique

Le `{tidyverse}` s'appelait encore le `hadleyverse` il y a quelques année, c'est-à-dire `l'univers de Hadley` pour Hadley Wickham son génial créateur.
Le but de Hadley est de rendre l'analyse données plus facile, plus rapide et surtout **plus fun** et je trouve que cela transparaît dans ses packages !
Le `{tidyverse}` c'est l'ensemble des packages open-source développé par Hadley et son équipe (Hadley travaille maintenant pour RStudio en plus de plusieurs universités) qui partagent la même philosophie, la même structure de données (le fameux format `tidy`) et la même syntaxe.

Le format `tidy`

Le format `tidy`repose sur la répétition des lignes des individus afin de limiter le nombre de colonnes.
Dans le plus stricte cas, le format `tidy` ne présente que 3 colonnes :

_ Identification de l'individu, *par exemple* : `nom_du_pays`, `num_bague_identification`,...
_ Variables mesurées, *par exemple* : `variable` peut prendre comme modalités `superficie`, `taille_population`, `PIB` ou `masse`, `taille`, `longueur_du_bec`...
_ Valeur de la mesure. **Attention**, le format `tidy` ne supporte par plusieurs type de données dans la même colonne !

La syntaxe `tidyverse`

Non détaillée ici, je vous invite à consulter le tidyverse style guide.

Les packages concernés

_ `ggplot2` : Visulisation des données
_ `dplyr` : Manipulation des données (filtrer, trier,...) à ne pas confondre avec `tidyr` qui manipule le format du jeu de données
_ `tidyr` : Modification du format du jeu de données pour en faire un jeu de donnée `tidy`
_ `readr` : Lecture rapide de fichiers de données format `csv` et autres. **Attention** : format `xslx` non pris en charge, il faut utiliser le package `readxl` qui fait partie du `tidyverse` au sens large mais qui n'est pas attaché par défaut quand on fait `library(tidyverse)`
_ `purrr` : Permet le remplacement d'un grand nombre de boucles *A faire pour un prochain direct ?*
_ `tibble` : Format des données `tidy`
_ `stringr` : Manipulation des chaînes de caractères. *A faire pour un prochain direct ?*
_ `forcats` : Manipulation des variables facteurs `factors`. *A faire pour un prochain direct ?*

```
library(tidyverse)
```

Projecteur sur :

`{dplyr}`

Le package `{dplyr}` permet de manipuler les données, c'est-à-dire, les filtrer, sélectionner les colonnes d'intérêts ou autres...
Exercice : trouver pour chaque poussin le jour où il a pris le plus de masse à partir des données `ChickWeight` présente de base dans R

```
# visualisation des données
head(ChickWeight)
```

```
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
## 4     64    6     1    1
## 5     76    8     1    1
## 6     93   10     1    1
```

```
# création de la colonne de différence de poids avec la fonction lag qui permet de prendre les valeurs
ChickWeight %>%
  group_by(
    Chick
  ) %>%
  mutate(
    prise_poids = weight - lag(weight)
  ) %>%
  ungroup()
```

```
## # A tibble: 578 x 5
##    weight  Time Chick Diet  prise_poids
##     <dbl> <dbl> <ord> <fct>       <dbl>
##  1     42     0 1     1              NA
##  2     51     2 1     1               9
##  3     59     4 1     1               8
##  4     64     6 1     1               5
##  5     76     8 1     1              12
##  6     93    10 1     1              17
##  7    106    12 1     1              13
##  8    125    14 1     1              19
##  9    149    16 1     1              24
## 10    171    18 1     1              22
## # ... with 568 more rows
```

```
# trie avec la fonction arrange
## sur toutes les lignes
ChickWeight %>%
  group_by(
    Chick
  ) %>%
  mutate(
    prise_poids = weight - lag(weight)
  ) %>%
  arrange(prise_poids) %>%
  ungroup()
```

```
## # A tibble: 578 x 5
##    weight  Time Chick Diet  prise_poids
##     <dbl> <dbl> <ord> <fct>       <dbl>
##  1    125    20 8     1              -9
##  2    147    21 33    3              -9
##  3     66     8 24    2              -8
##  4    150    21 30    2              -7
##  5     90    12 9     1              -6
##  6    175    21 11    1              -6
##  7     51    10 16    1              -6
##  8    146    18 33    3              -5
##  9    220    21 36    3              -5
## 10    205    21 47    4              -5
## # ... with 568 more rows
```

```
## en prenant l'effet groupe
ChickWeight %>%
  group_by(
    Chick
  ) %>%
  mutate(
    prise_poids = weight - lag(weight)
  ) %>%
  arrange(prise_poids, .by_group = TRUE) %>%
  ungroup()
```

```
## # A tibble: 578 x 5
```

```
##    weight  Time Chick Diet  prise_poids
##     <dbl> <dbl> <ord> <fct>       <dbl>
##  1     35     2 18    1              -4
##  2     39     0 18    1              NA
##  3     51    10 16    1              -6
##  4     51     6 16    1               2
##  5     54    12 16    1               3
##  6     45     2 16    1               4
##  7     49     4 16    1               4
##  8     57     8 16    1               6
##  9     41     0 16    1              NA
## 10     67    12 15    1              -1
## # ... with 568 more rows
```

```
# Élimination des poussins qui ne sont pas mesurés pendant toute l'expérience
ChickWeight %>%
  count(
    Chick
  )
```

```
##    Chick  n
## 1     18  2
## 2     16  7
## 3     15  8
## 4     13 12
## 5      9 12
## 6     20 12
## 7     10 12
## 8      8 11
## 9     17 12
## 10    19 12
## 11     4 12
## 12     6 12
## 13    11 12
## 14     3 12
## 15     1 12
## 16    12 12
## 17     2 12
## 18     5 12
## 19    14 12
## 20     7 12
## 21    24 12
## 22    30 12
## 23    22 12
## 24    23 12
## 25    27 12
## 26    28 12
## 27    26 12
## 28    25 12
## 29    29 12
## 30    21 12
## 31    33 12
## 32    37 12
## 33    36 12
```

```
## 34    31 12
## 35    39 12
## 36    38 12
## 37    32 12
## 38    40 12
## 39    34 12
## 40    35 12
## 41    44 10
## 42    45 12
## 43    43 12
## 44    41 12
## 45    47 12
## 46    49 12
## 47    46 12
## 48    50 12
## 49    42 12
## 50    48 12
```

```
ChickWeight %>%
  count(
    Chick
  ) %>%
  filter(n != 12)
```

```
##   Chick  n
## 1    18  2
## 2    16  7
## 3    15  8
## 4     8 11
## 5    44 10
```

```
ChickWeight %>%
  count(
    Chick
  ) %>%
  filter(n != 12) %>%
  select(Chick)
```

```
##   Chick
## 1    18
## 2    16
## 3    15
## 4     8
## 5    44
```

```
ChickWeight %>%
  filter(
    Chick %in% (
      ChickWeight %>%
        count(Chick) %>%
        filter(n == 12)
    )$Chick
```

```
  ) %>%
  group_by(
    Chick
  ) %>%
  mutate(
    prise_poids = weight - lag(weight)
  ) %>%
  arrange(desc(prise_poids), .by_group = TRUE) %>%
  ungroup()
```

```
## # A tibble: 540 x 5
##    weight  Time Chick Diet  prise_poids
##     <dbl> <dbl> <ord> <fct>       <dbl>
##  1     81    18 13    1              10
##  2     91    20 13    1              10
##  3     48     2 13    1               7
##  4     60     6 13    1               7
##  5     53     4 13    1               5
##  6     65     8 13    1               5
##  7     96    21 13    1               5
##  8     71    12 13    1               4
##  9     67    10 13    1               2
## 10     71    16 13    1               1
## # ... with 530 more rows
```

```
# utilisation de slice pour sélectionner uniquement la plus forte valeur pour chaque poussin
ChickWeight %>%
  filter(
    Chick %in% (
      ChickWeight %>%
        count(Chick) %>%
        filter(n == 12)
    )$Chick
  ) %>%
  group_by(
    Chick
  ) %>%
  mutate(
    prise_poids = weight - lag(weight)
  ) %>%
  arrange(desc(prise_poids), .by_group = TRUE) %>%
  slice(1) %>%
  ungroup()
```

```
## # A tibble: 45 x 5
##    weight  Time Chick Diet  prise_poids
##     <dbl> <dbl> <ord> <fct>       <dbl>
##  1     81    18 13    1              10
##  2     85     8 9     1              17
##  3     89    14 20    1              12
##  4     63     6 10    1              11
##  5     72     6 17    1              11
```

```
##  6    144    20 19    1             24
##  7    136    16 4     1             28
##  8    124    10 6     1             27
##  9    168    12 11    1             29
## 10    163    16 3     1             25
## # ... with 35 more rows
```

```
# création d'un tableau résumé
ChickWeight %>%
  filter(
    Chick %in% (
      ChickWeight %>%
        count(Chick) %>%
        filter(n == 12)
    )$Chick
  ) %>%
  group_by(
    Chick
  ) %>%
  mutate(
    prise_poids = weight - lag(weight)
  ) %>%
  arrange(desc(prise_poids), .by_group = TRUE) %>%
  slice(1) %>%
  ungroup() %>%
  summarise(
    moyenne_jour = mean(prise_poids),
    min_jour = min(prise_poids),
    max_jour = max(prise_poids)
  )
```

```
## # A tibble: 1 x 3
##   moyenne_jour min_jour max_jour
##          <dbl>    <dbl>    <dbl>
## 1         31.3       10       59
```

{tidyr}

Le package `{tidyr}` permet de mettre en forme les données pour respecter le format `tidy`.
Exemple avec le jeu de données ***`mtcars`*** *clairement pas* ***`tidy`***

```
mtcars
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
```

```
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
# transformation en format tibble.
mtcars %>%
  as_tibble()
```

```
## # 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
```

```
# perte des noms des voitures

# enregistrement des noms dans une colonne
mtcars %>%
  rownames_to_column(var = "vehicule") %>%
  as_tibble()
```

```
## # A tibble: 32 x 12
##    vehicule      mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
```

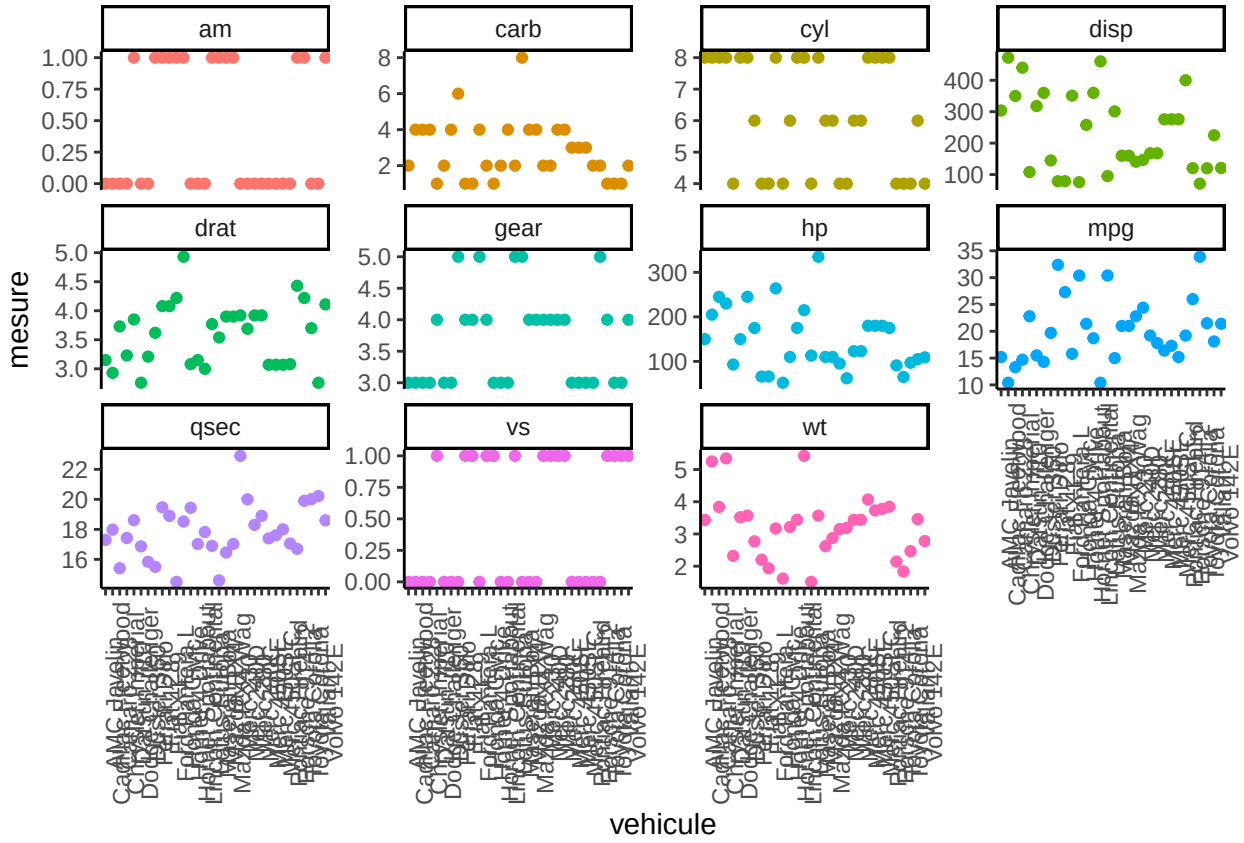
```
##    <chr>       <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 Mazda RX4    21       6  160    110  3.9   2.62  16.5     0     1     4     4
##  2 Mazda RX4 ~  21       6  160    110  3.9   2.88  17.0     0     1     4     4
##  3 Datsun 710   22.8     4  108     93  3.85  2.32  18.6     1     1     4     1
##  4 Hornet 4 D~  21.4     6  258    110  3.08  3.22  19.4     1     0     3     1
##  5 Hornet Spo~  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
##  7 Duster 360   14.3     8  360    245  3.21  3.57  15.8     0     0     3     4
##  8 Merc 240D    24.4     4  147.    62  3.69  3.19  20       1     0     4     2
##  9 Merc 230     22.8     4  141.    95  3.92  3.15  22.9     1     0     4     2
## 10 Merc 280     19.2     6  168.   123  3.92  3.44  18.3     1     0     4     4
## # ... with 22 more rows
```

```
# réalisation d'un pivot
mtcars %>%
  rownames_to_column(var = "vehicule") %>%
  as_tibble() %>%
  pivot_longer(
    - vehicule,
    names_to = "variable",
    values_to = "mesure"
  )
```

```
## # A tibble: 352 x 3
##    vehicule  variable mesure
##    <chr>     <chr>     <dbl>
##  1 Mazda RX4 mpg       21
##  2 Mazda RX4 cyl        6
##  3 Mazda RX4 disp     160
##  4 Mazda RX4 hp       110
##  5 Mazda RX4 drat       3.9
##  6 Mazda RX4 wt         2.62
##  7 Mazda RX4 qsec      16.5
##  8 Mazda RX4 vs         0
##  9 Mazda RX4 am         1
## 10 Mazda RX4 gear       4
## # ... with 342 more rows
```

```
# exemple d'utilisation
mtcars %>%
  rownames_to_column(var = "vehicule") %>%
  as_tibble() %>%
  pivot_longer(
    - vehicule,
    names_to = "variable",
    values_to = "mesure"
  ) %>%
  ggplot() +
  aes(y = mesure, x = vehicule, color = variable) +
  geom_point() +
  facet_wrap(~ variable, scales = "free_y") +
  theme_classic() +
  theme(
```

```
    axis.text.x = element_text(angle = 90),
    legend.position = "none"
    )
```



Exemple avec `ChickWeight` *pour la transformer en une table utilisable pour faire une ACP*

```
# pivoter
ChickWeight %>%
  pivot_wider(
    names_from = Time,
    values_from = weight
  )
```

```
## # A tibble: 50 x 14
##    Chick Diet    `0`   `2`   `4`   `6`   `8`  `10`  `12`  `14`  `16`  `18`  `20`
##    <ord> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 1     1        42    51    59    64    76    93   106   125   149   171   199
##  2 2     1        40    49    58    72    84   103   122   138   162   187   209
##  3 3     1        43    39    55    67    84    99   115   138   163   187   198
##  4 4     1        42    49    56    67    74    87   102   108   136   154   160
##  5 5     1        41    42    48    60    79   106   141   164   197   199   220
##  6 6     1        41    49    59    74    97   124   141   148   155   160   160
##  7 7     1        41    49    57    71    89   112   146   174   218   250   288
##  8 8     1        42    50    61    71    84    93   110   116   126   134   125
##  9 9     1        42    51    59    68    85    96    90    92    93   100   100
```

```
## 10 10    1        41    44    52    63    74    81    89    96   101   112   120
## # ... with 40 more rows, and 1 more variable: `21` <dbl>
```

```
# retirer les lignes où il manque au moins une mesure
ChickWeight %>%
  pivot_wider(
    names_from = Time,
    values_from = weight
  ) %>%
  drop_na()
```

```
## # A tibble: 45 x 14
##    Chick Diet    `0`   `2`   `4`   `6`   `8`  `10`  `12`  `14`  `16`  `18`  `20`
##    <ord> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 1     1        42    51    59    64    76    93   106   125   149   171   199
##  2 2     1        40    49    58    72    84   103   122   138   162   187   209
##  3 3     1        43    39    55    67    84    99   115   138   163   187   198
##  4 4     1        42    49    56    67    74    87   102   108   136   154   160
##  5 5     1        41    42    48    60    79   106   141   164   197   199   220
##  6 6     1        41    49    59    74    97   124   141   148   155   160   160
##  7 7     1        41    49    57    71    89   112   146   174   218   250   288
##  8 9     1        42    51    59    68    85    96    90    92    93   100   100
##  9 10    1        41    44    52    63    74    81    89    96   101   112   120
## 10 11    1        43    51    63    84   112   139   168   177   182   184   181
## # ... with 35 more rows, and 1 more variable: `21` <dbl>
```

```
# modifier les noms des colonnes
ChickWeight %>%
  pivot_wider(
    names_from = Time,
    names_glue = "jour_{Time}",
    values_from = weight
  ) %>%
  drop_na()
```

```
## # A tibble: 45 x 14
##    Chick Diet  jour_0 jour_2 jour_4 jour_6 jour_8 jour_10 jour_12 jour_14
##    <ord> <fct>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>   <dbl>   <dbl>   <dbl>
##  1 1     1         42     51     59     64     76      93     106     125
##  2 2     1         40     49     58     72     84     103     122     138
##  3 3     1         43     39     55     67     84      99     115     138
##  4 4     1         42     49     56     67     74      87     102     108
##  5 5     1         41     42     48     60     79     106     141     164
##  6 6     1         41     49     59     74     97     124     141     148
##  7 7     1         41     49     57     71     89     112     146     174
##  8 9     1         42     51     59     68     85      96      90      92
##  9 10    1         41     44     52     63     74      81      89      96
## 10 11    1         43     51     63     84    112     139     168     177
## # ... with 35 more rows, and 4 more variables: jour_16 <dbl>, jour_18 <dbl>,
## #   jour_20 <dbl>, jour_21 <dbl>
```