R Notebooks in a book

Yannick Le Corre

2018-12-31

Contents

Preface

TODO

1 R

1.1 conference 2018

```
library(knitr)
knitr::opts_chunk$set(warning=FALSE, message=FALSE)
```

by Emily Robinson (DataCamp) at the 2018 New York R Conference Les packages de l'écosystème tidyverse sont décrits dans le diagramme 1. Noter que leaflet ne fait pas partie de tidyverse.

1.1.1 make a toy dataset

```
library(tibble)
df <- tribble(
~date, ~whatever, ~numstuff1, ~numstuff2, ~multi_value,
"10-09-2017", .1, 5, NA, "val1, val2, val3",
"10-09-2017", .5, 1, 3, "val1",
"11-09-2017", .2, 3, 2, "val1, val2",
"12-09-2017", .5, Inf, 3, "val1, val2, val3, val4"
)
df
```

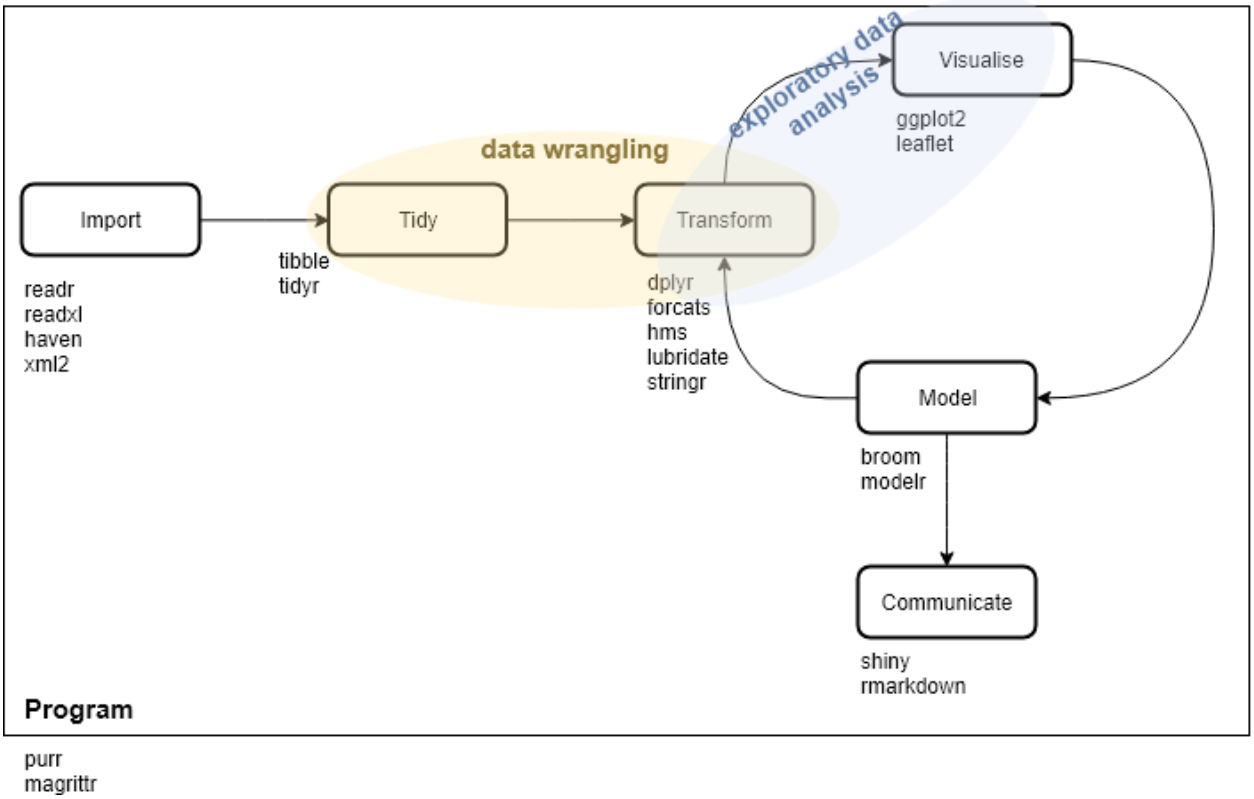


Figure 1: data anlysis workflow (draw.io)

```
## # A tibble: 4 x 5
##   date       whatever numstuff1 numstuff2 multi_value
##   <chr>         <dbl>     <dbl>     <dbl> <chr>
## 1 10-09-2017      0.1         5        NA val1, val2, val3
## 2 10-09-2017      0.5         1         3 val1
## 3 11-09-2017      0.2         3         2 val1, val2
## 4 12-09-2017      0.5       Inf         3 val1, val2, val3, val4
```

Examine your NAs and empty values

```
library(purrr)
df %>%
  purrr::map_df(~sum(is.na(.))) # count the NA for every columns of a DF
```

```
## # A tibble: 1 x 5
##    date whatever numstuff1 numstuff2 multi_value
##   <int>    <int>     <int>     <int>       <int>
## 1     0        0         0         1           0
```

```
library(dplyr)
df <- df %>%
  na_if(Inf) # convert empty values into NA values
df
```

```
## # A tibble: 4 x 5
##   date       whatever numstuff1 numstuff2 multi_value
##   <chr>         <dbl>     <dbl>     <dbl> <chr>
## 1 10-09-2017      0.1         5        NA val1, val2, val3
## 2 10-09-2017      0.5         1         3 val1
## 3 11-09-2017      0.2         3         2 val1, val2
## 4 12-09-2017      0.5        NA         3 val1, val2, val3, val4
```

1.1.2 Examine your numeric columns

```
library(skimr)
df %>%
  select_if(is.numeric) %>%
  skimr::skim() # summary of all the numeric columns
```

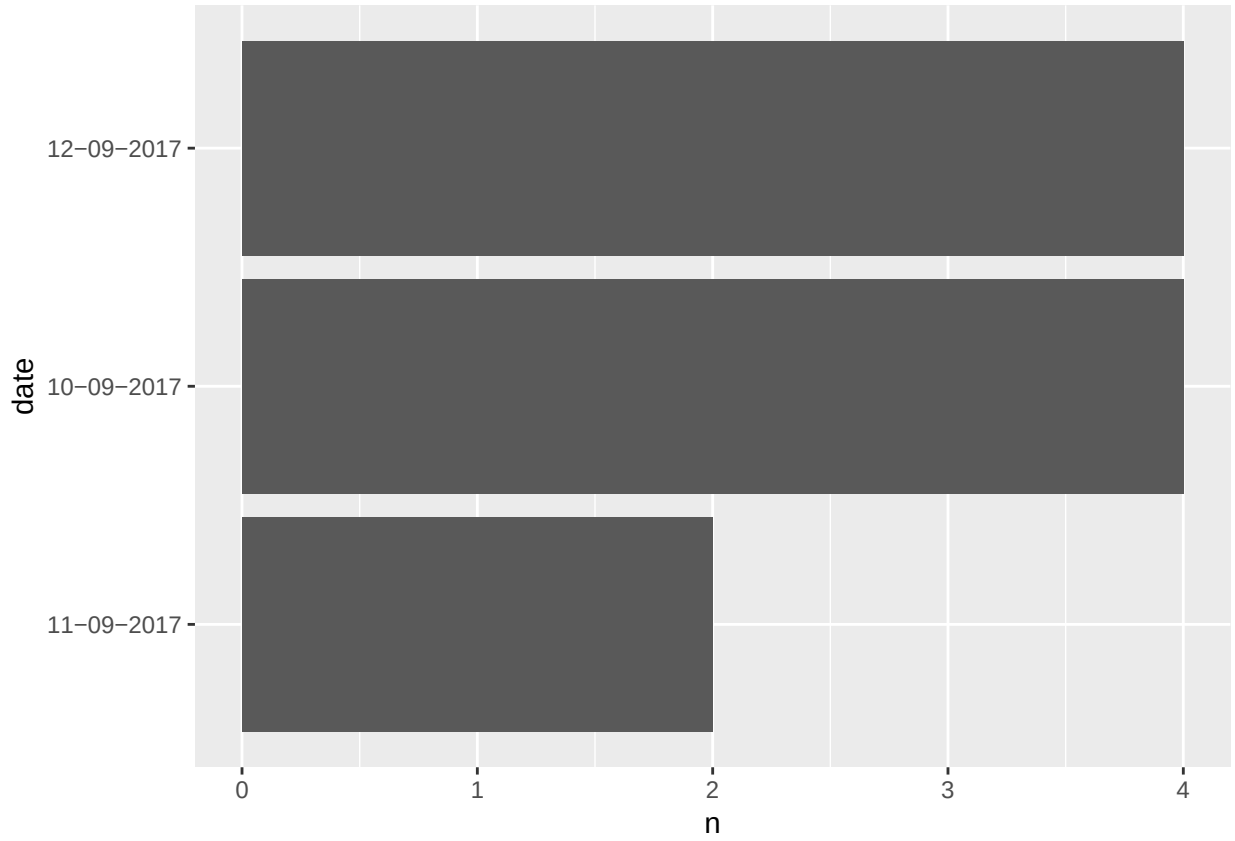
```
## Skim summary statistics
##  n obs: 4
##  n variables: 3
##
## -- Variable type:numeric ------------------------------------------------------------------------
##   variable missing complete n mean   sd  p0  p25  p50 p75 p100     hist
##  numstuff1       1        3 4 3    2    1   2    3    4    5   <U+2587><U+2581><U+2581><U+2587><U+258
##  numstuff2       1        3 4 2.67 0.58 2   2.5  3    3    3   <U+2583><U+2581><U+2581><U+2581><U+258
##   whatever       0        4 4 0.32 0.21 0.1 0.18 0.35 0.5  0.5 <U+2583><U+2583><U+2581><U+2581><U+258
```

1.1.3 Tidy columns containing colon separated values

```
library(tidyr)
library(stringr)
df <- df %>%
mutate(colonne = str_split(multi_value, ",")) %>% # transform into a list of characters
unnest() # split into seperate observations
```

1.1.4 plot uncluttered bar charts

```
library(ggplot2)
library(forcats)
df %>%
count(date) %>%
mutate(date = fct_reorder(date, n)) %>% # ordonne les colonnes en ordre décroissant
ggplot(aes(x = date, y = n)) + geom_col() + coord_flip()
```



1.2 Sams Teach Yourself R

By Andy Nicholls, Richard Pugh, & Aimee Gott (first edition dec 2015) object types

4 modes (fonction mode()): • numeric • character • boolean • complex

3 types (fonction class()): • vector (c'est le type de base) - attributes length et names • matrix/array - attribut dim • list

1.2.1 cut et split

cut est utilisé pour classer des données numériques en facteur

```
numeric_matrix <- 1:100
mycut1 <- cut(numeric_matrix, 3) # classifie les données selon trois classes de valeurs égales
head(mycut1)
```

```
## [1] (0.901,34] (0.901,34] (0.901,34] (0.901,34] (0.901,34] (0.901,34]
## Levels: (0.901,34] (34,67] (67,100]
```

```
table(mycut1)
```

```
## mycut1
## (0.901,34]    (34,67]   (67,100]
##         34         33         33
```

On peut aussi couper selon des classes numériques spécifiques

```
mycut2 <- cut(numeric_matrix, breaks = c(1, 10, 100), include.lowest = TRUE) # classes [1, 10], ]10, 10
head(mycut2)
```

```
## [1] [1,10] [1,10] [1,10] [1,10] [1,10] [1,10]
## Levels: [1,10] (10,100]
```

```
table(mycut2)
```

```
## mycut2
##   [1,10] (10,100]
##       10       90
```

La fonction split permet de découper un ensemble selon des facteurs. Donc on peut combiner les fonctions cut et split pour décomposer un vecteur ou DF en liste de vecteurs/DF groupées par facteur.

```
mysplit_list <- split(numeric_matrix, mycut2)
```

la fonction split fonctionne aussi sur des DF

```
df <- data.frame(value = sample(1:1000, size = 100), whatever = LETTERS[sample(1:length(LETTERS), size =
split(df, cut(df$value, 3))
```

```
## $`(13,342]`
##    value whatever
## 1    123        X
## 3    266        V
## 4    135        U
```

```
## 8     46        P
## 9    264        W
## 10   143        I
## 17   305        B
## 19   129        B
## 21   137        I
## 25    47        H
## 33   313        S
## 34   116        Q
## 38    38        P
## 39   342        Q
## 41   322        H
## 44   277        Z
## 46    69        G
## 49   132        L
## 52   164        Q
## 56   152        X
## 58   127        G
## 62   174        Z
## 66    14        V
## 71   149        Q
## 76   214        A
## 77   324        A
## 80   255        D
## 83   336        J
## 85    84        O
## 90   254        P
## 91   299        U
## 94    33        G
## 97   140        V
## 98   303        L
##
## $`(342,671]`
##    value whatever
## 2    532        Z
## 5    526        N
## 6    553        Z
## 11   393        V
## 13   388        A
## 14   456        R
## 16   520        T
## 20   551        D
## 23   361        E
## 24   475        L
## 29   396        O
## 32   496        Y
## 35   415        B
## 42   419        N
## 43   666        S
## 45   619        U
## 47   592        T
## 54   537        H
## 55   485        I
## 59   377        L
```

```
## 61   397        D
## 63   602        O
## 64   655        V
## 65   521        E
## 69   590        B
## 70   447        K
## 73   612        E
## 74   413        Z
## 82   364        E
## 84   556        R
## 87   513        O
## 88   650        C
## 89   435        Q
## 93   371        F
## 96   505        W
##
## $`(671,1e+03]`
##     value whatever
## 7     936        K
## 12    928        M
## 15    912        R
## 18    999        Q
## 22    704        L
## 26    681        K
## 27    904        S
## 28    732        S
## 30    736        A
## 31    982        A
## 36    781        I
## 37    700        A
## 40    947        E
## 48    804        S
## 50    858        G
## 51    974        D
## 53    939        G
## 57    744        V
## 60    702        L
## 67    945        F
## 68    898        B
## 72    996        H
## 75    812        Q
## 78    964        F
## 79    979        S
## 81    689        F
## 86    764        T
## 92    950        P
## 95    895        N
## 99    910        Y
## 100   698        U
```

1.3 Advanced R

The book Advanced R explains a lot of fundamental aspects of the R language.

The library **lobstr** allows to check the size of a variable

```
library(lobstr)
obj_size(1)
```

```
## 56 B
```

Main object types

```
t1 <- c(1:5)
```

```
show_type(t1)
```

```
## [1] 1 2 3 4 5
## [1] "integer"
##  int [1:5] 1 2 3 4 5
```

```
t2 <- c(1.1, 1e2)
show_type(t2)
```

```
## [1]   1.1 100.0
## [1] "double"
##  num [1:2] 1.1 100
```

```
t3 <- c("dog", "cat")
show_type(t3)
```

```
## [1] "dog" "cat"
## [1] "character"
##  chr [1:2] "dog" "cat"
```

```
t4 <- c(TRUE, FALSE)
show_type(t4)
```

```
## [1]  TRUE FALSE
## [1] "logical"
##  logi [1:2] TRUE FALSE
```

```
t5 <- list(1:2, 2:3)
show_type(t5)
```

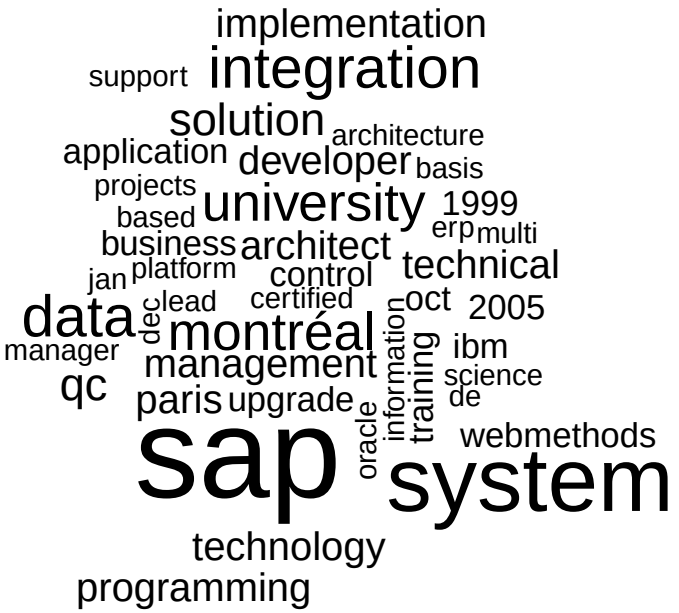
```
## [[1]]
## [1] 1 2
##
## [[2]]
## [1] 2 3
##
## [1] "list"
## List of 2
##  $ : int [1:2] 1 2
##  $ : int [1:2] 2 3
```

```r
t6 <- data.frame(col1 = c(1:5), col2 = letters[1:5])
show_type(t6)
```

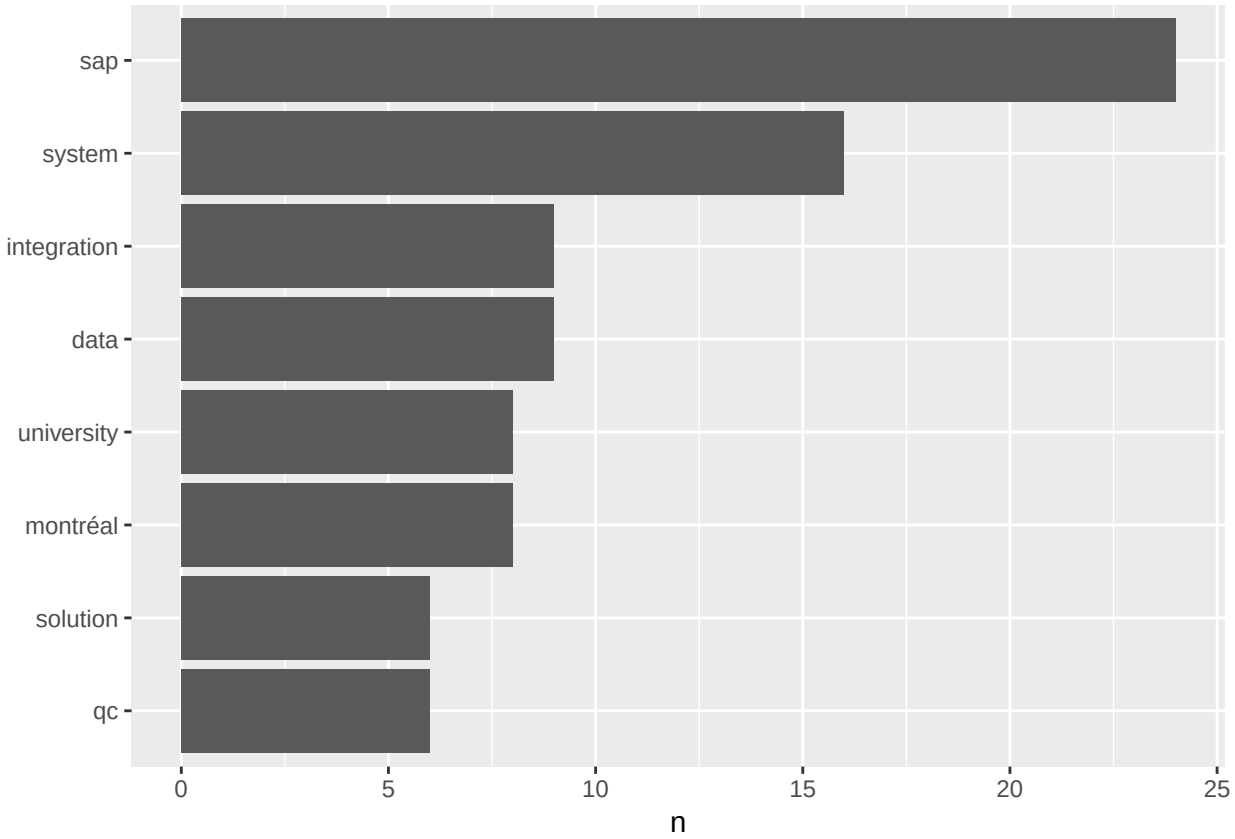
```
##   col1 col2
## 1    1    a
## 2    2    b
## 3    3    c
## 4    4    d
## 5    5    e
## [1] "list"
## 'data.frame':    5 obs. of  2 variables:
##  $ col1: int  1 2 3 4 5
##  $ col2: Factor w/ 5 levels "a","b","c","d",..: 1 2 3 4 5
```

2 Resume analysis

```r
FILENAME <- paste(getwd(), "/2019_last.docx", sep="")
```



Filtre des occurences > 5


sap
system
integration
data
university
montréal
solution
qc
0
5
10
15
20
25
n