Data Scrapping & Wrangling

Léo Dange

5/8/2020

Introduction

Here is an exemple of Data Scrapping and Wrangling from a pdf file located in the dslabs package in our computre. We will first load the tidyverse, stringr & pdftools libraries that we will need for this work.

```
library(tidyverse)
library(tinytex)
library(pdftools)
library(stringr)
library(ggplot2)
options(digits = 3)
```

Extracting data

First we will extract data from the pdf and open it to have a look at it.

```
fn <- system.file("extdata", "RD-Mortality-Report_2015-18-180531.pdf", package="dslabs")
system2("open", args = fn)
```

We are going to create a tidy dataset with each row representing one observation. The variables in this dataset will be year, month, day and deaths.

First we extract all the text from the pdf and store it as it is in an object "txt" then We extract the ninth page from the tidy dataset with the "str_split()" function and store them in the object "x". We can see that it is a "list" with 1 entry with lost of data to get rid of.

```
txt <- pdf_text(fn)
x <- str_split(txt[9], "\n")
class(x)
```

```
## [1] "list"
```

```
length(x)
```

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

we extract the first entry of "x" and store it in "s". Carrefull, if you forget on of the two "[ ]" it doest not creat a character vector but still a list.

```r
s <- x[[1]]
class(s)
```

```
## [1] "character"
```

```r
length(s)
```

```
## [1] 40
```

Data Wrangling

Now that we have extracted the data we are ready to work on the ninth page. Before starting we will "trim" the "s" object to remove white space before and after character using the command str_trim().

```r
s <- str_trim(s)
```

Before extracting the number from the string we want to preserve the header. We will use the "str_which()" function to detect where is located the column header by searching the year "2015" and we will save the column in an object called "header_index".

```r
header_index <- str_which(s, "2015")[1]
```

Then we will extract the header in two object "month" to store the month and "header" to store the column name.

If we just extract it as it is with the following code the header will include space that we do not want. In this exemple the header is not usable :

```r
header <- str_split(s[header_index],"\\s")
```

To do it properly We will need to create a temporary file "temp" and look for "one or more space" and using the simplify argument. Then we split it into the two object

```r
temp <- str_split(s[header_index], "\\s+", simplify = TRUE)
month <- temp[1]
header <- temp[-1]
month
```

```
## [1] "SEP"
```

Now that we saved the month and header we well create index that store row we will delete later. Start by creating the "tail_index" object sotring the row with the "Total" value and create an object "n" storing the row the count of number in each row.

```r
tail_index  <- str_which(s, "Total")
n <- str_count(s,"\\d+")
```

Now we can start cleaning the data. We will remove : - The "header_index" and the row before - The row in "n" with only one number - The "tail_index" and all the row after

To do that we will creat a list of those vector like that :

```
out <- c(1:header_index, which(n==1), tail_index:length(s))
out
```

```
##  [1]  1  2  6  9 35 36 37 38 39 40
```

Then we remove it from s that as now a length of 30 instead of 40

```
s <- s[-out]
length(s)
```

```
## [1] 30
```

It is time to remove all text that is not digit or space.

```
s <- str_remove_all(s, "[^\\d\\s]")
head(s)
```

```
## [1] "1     75    75     94      0"
## [2] "2     77    67     69      0
## [3] "3     67    78     80      0"
## [4] "4     71    99     84      0"
## [5] "5     62    89     74      0"
## [6] "6     77    74     83      0"
```

We still have on the row [19] a sequence of number from 1 to 30 with some years that come from the graph on the same page. We will remove them by only keeping data from the table and converting it into a data matrix with just the day and death count :

```
s <- str_split_fixed(s, "\\s+", n = 6)[,1:5]
head(s)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] "1"  "75" "75" "94" "0"
## [2,] "2"  "77" "67" "69" "0"
## [3,] "3"  "67" "78" "80" "0"
## [4,] "4"  "71" "99" "84" "0"
## [5,] "5"  "62" "89" "74" "0"
## [6,] "6"  "77" "74" "83" "0"
```

Tidy transformation

Before being done we will change the column name as “day” should be the first name and the “header” should follow:

```
tab <- s %>%
    as_data_frame() %>%
    setNames(c("day", header)) %>%
    mutate_all(as.numeric)
```

```
## Warning: `as_data_frame()` is deprecated, use `as_tibble()` (but mind the new semantics).
## This warning is displayed once per session.
```

Final step to have a tidy format we will use "gather" to rearange the data frame with column "day","year","deaths"

```
tab <- tab %>% gather(year, deaths, -day) %>%
    mutate(deaths = as.numeric(deaths))
```

Graph

We are done with the tidy data, we can now plote deaths vs day excluding 2018.

```
tab %>% filter(year < 2018) %>%
          ggplot(aes(day, deaths, color = year)) +
          geom_smooth() +
          geom_vline(xintercept = 20, alpha=0.4) +
          geom_point()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```