French given names per year per department

Lucas Mello Schnorr, Jean-Marc Vincent

October, 2022

The problem context

The aim of the activity is to develop a methodology to answer a specific question on a given dataset.

The dataset is the set of Firstname given in France on a large period of time. https://www.insee.fr/fr/statistiques/2540004, we choose this dataset because it is sufficiently large, you can't do the analysis by hand, the structure is simple

You need to use the *tidyverse* for this analysis. Unzip the file *dpt2020_txt.zip* (to get the **dpt2020.csv**). Read in R with this code. Note that you might need to install the `readr` package with the appropriate command.

Download Raw Data from the website

```
file = "dpt2021_csv.zip"
if(!file.exists(file)){
  download.file("https://www.insee.fr/fr/statistiques/fichier/2540004/dpt2021_csv.zip",
    destfile=file)
}
unzip(file)
```

Check if your file is the same as in the first analysis (reproducibility)

```
md5sum dpt2021.csv
```

```
## f18a7d627883a0b248a0d59374f3bab7  dpt2021.csv
```

expected : MD5 (dpt2021.csv) = f18a7d627883a0b248a0d59374f3bab7

Build the Dataframe from file

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
FirstNames <- read_delim("dpt2021.csv",delim=";")
```

```
## Rows: 3784673 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr (3): preusuel, annais, dpt
## dbl (2): sexe, nombre
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

All of these following questions may need a preliminary analysis of the data, feel free to present answers and justifications in your own order and structure your report as it should be for a scientific report.

1. Choose a firstname and analyse its frequency along time. Compare several firstnames frequency
2. Establish by gender the most given firstname by year. Analyse the evolution of the most frequent firstname.
3. Optional : Which department has a larger variety of names along time ? Is there some sort of geographical correlation with the data?

Analysis of a given firstname

We are going to analyze the frequency of the name "Thibaut", and other common spellings ("Thibault", "Thibaud") of the name to compare their popularity.

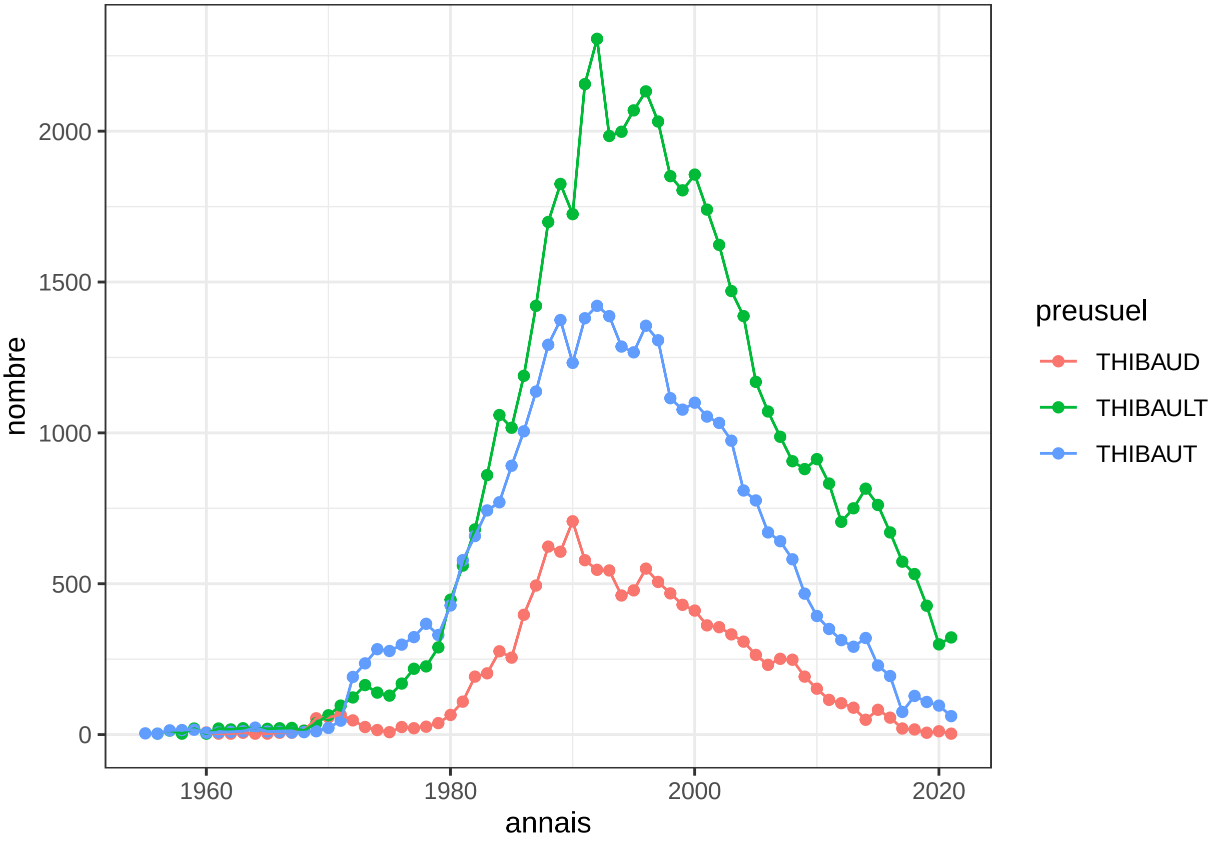
```
graph_freq <- function(list_names, y_scale_log) {
  # list_names is an array of names in capital
  # Only keep data for given names
  FirstNames %>% filter(preusuel %in% list_names) -> AllGivenNames

  # Plot frequency over time (year)
  AllGivenNames %>% filter(annais!="XXXX") %>% group_by(preusuel, annais) %>% summarize(nombre = sum(nor

  render = ggplot(AllGivenNamesYears, aes(annais, nombre, color=preusuel)) + theme_bw() + geom_point() +
  if (y_scale_log) {
    render + scale_y_log10()
  } else {
    render
  }
}

graph_freq(c("THIBAULT","THIBAUT","THIBAUD"), FALSE)
```

```
## `summarise()` has grouped output by 'preusuel'. You can override using the
## `.groups` argument.
```



As we can see the name was most popular between the 80s and 10s and is downgrading almost constantly this recent years. The different spellings follow the same trend overall, but we can observe "Thibault" is the most popular one and "Thibaud" the least since the 80s. In the 70s, "Thibaut" was the most popular spelling of the name in France.

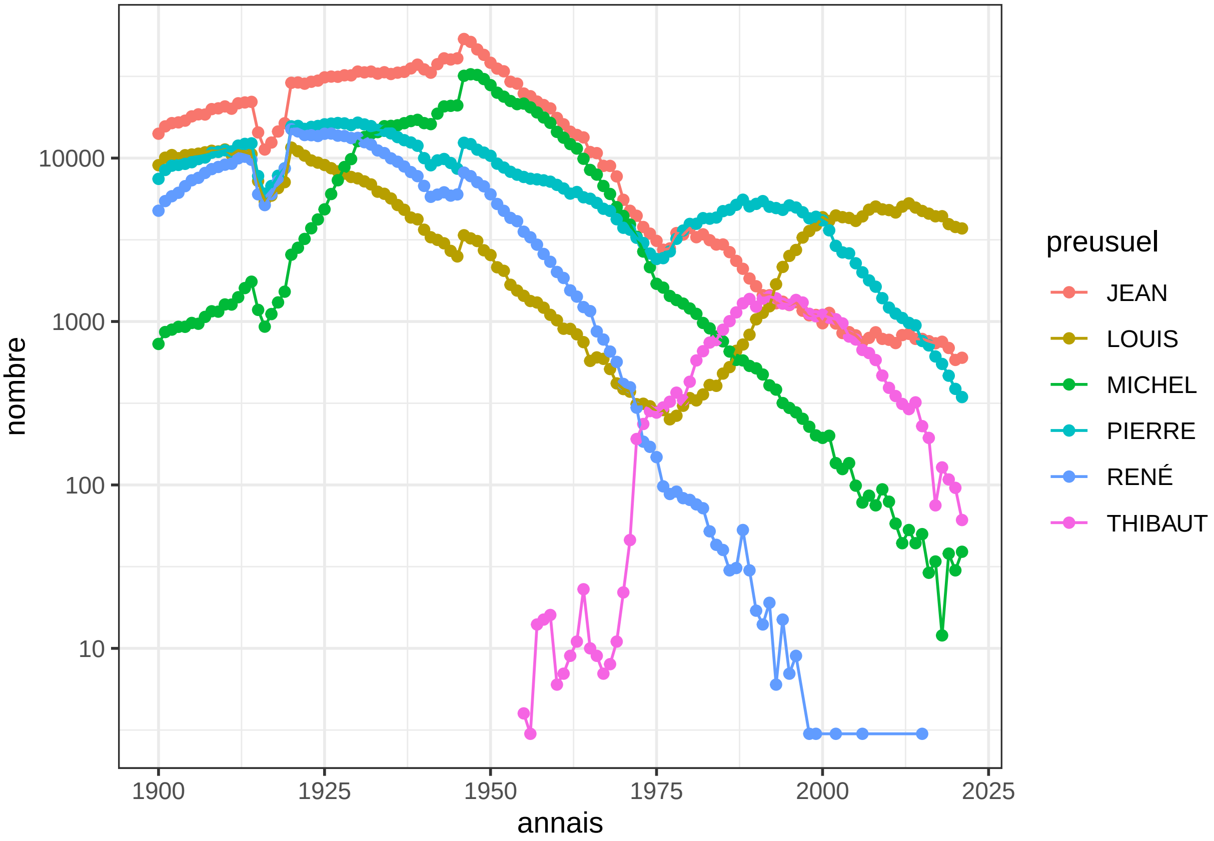
Now, let's see how "Thibaut" compares to other popular names in terms of frequency. This time we will use logarithmic scales because it's not possible to clearly see some parts of the graph if we do not use it. I choose to compare the name to other "masculine" first names among the top 10

```
# List of popular names
FirstNames %>% group_by(preusuel) %>% summarize(n = sum(nombre)) %>% arrange(desc(n))
```

```
## # A tibble: 36,171 x 2
##    preusuel             n
##    <chr>            <dbl>
##  1 MARIE          2259123
##  2 JEAN           1913968
##  3 _PRENOMS_RARES 1718677
##  4 PIERRE          892866
##  5 MICHEL          820611
##  6 ANDRÉ           712675
##  7 JEANNE          561151
##  8 PHILIPPE        538880
##  9 LOUIS           529131
## 10 RENÉ            516631
## # i 36,161 more rows
```

```
graph_freq(c("THIBAUT","JEAN","PIERRE","RENÉ","MICHEL","LOUIS"), TRUE)
```

```
## `summarise()` has grouped output by 'preusuel'. You can override using the
## `.groups` argument.
```



This graph tells us a lot of info. First of all, some parts of the curve are related to events which influenced the number of newborns (For example, World War 1 where every name has a drop between 1915 and 1920). But most importantly, we see that the names do not share the same tendencies.

For example, "René" is almost not given to anyone anymore, whereas "Louis" has regained popularity. Compared to "Thibaut", which was more given in the 80s, "Louis" is now >20 times as popular.

Most given firstname by year for two sexes

For the following study, we will consider the two sexes attributed at birth in France (male and female).

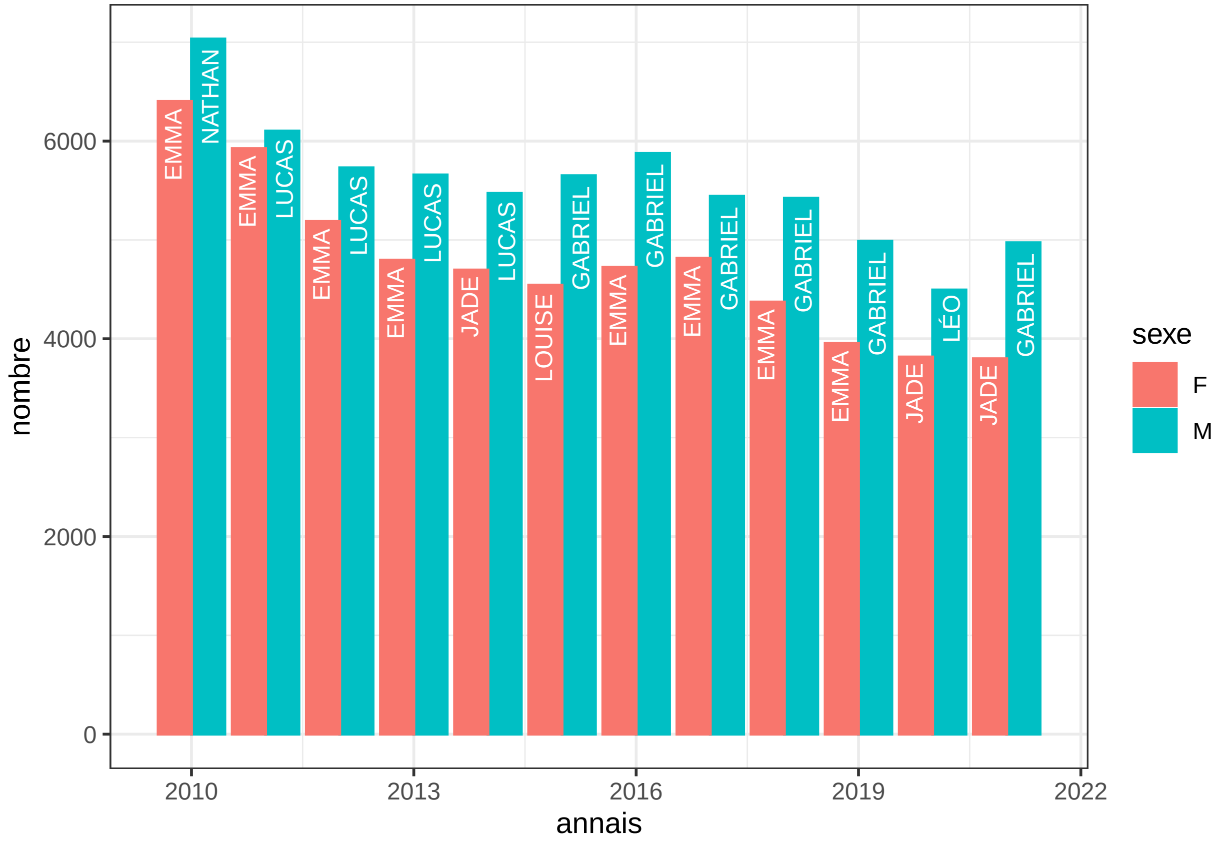
```
# Ignore data for unknown year & rare names and merge each department in France
FirstNames %>% filter(preusuel!="_PRENOMS_RARES") %>% group_by(sexe, preusuel, annais) %>% summarize(no
```

```
## `summarise()` has grouped output by 'sexe', 'preusuel'. You can override using
## the `.groups` argument.
```

```
# Only keep the most popular name for each sex and year
# We can use filter(nomber=max(nombre)) but I also wanted to try out sorting with arrange
FirstNamesFrance %>% group_by(sexe, annais) %>% arrange(desc(nombre)) %>% filter(row_number()==1) -> Mos

# We will only keep a slice of the data to show the graph
intToSex <- c("M","F")
MostGivenNamePerYear %>% mutate(annais = as.integer(annais), sexe = intToSex[as.integer(sexe)], nombre
plot <- ggplot(MostGivenNamePerYearStart2010, aes(x=annais, y=nombre, fill=sexe)) + geom_bar(stat="iden

# Render plot with text
plot + theme_bw() + geom_text(aes(label=preusuel), position=position_dodge(1), size=3, hjust = 1.1, col
```

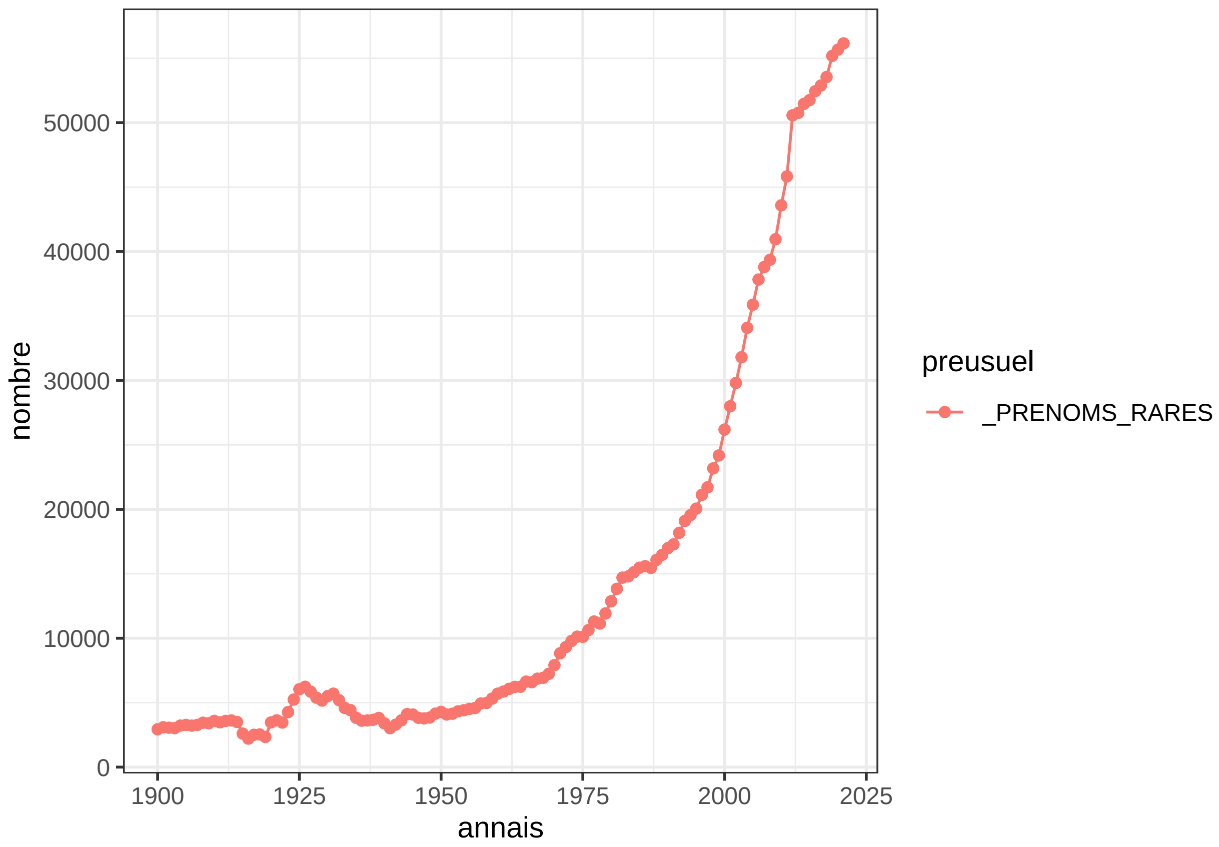


Since 2010, it seems “Lucas” and then Gabriel” are really popular for the male sex. For female, “Emma” and “Jade” are the most given names. We can also observe that the most popular name for male is always more given than the most popular female name, which is likely not due to the birth frequency between those two sexes because they are roughly the same. (We could explore some hypothesis, like female having more uncommon names, or maybe studying the top 5 names for each gender, to see if the top 5 given name for females have approx the same number as top 5 male).

Also, we can see that over time the most popular name for each sex is decreasing. This likely means that more different names are given to newborns (or different spelling), which can also be observed with the growing number of “rares names” in the data (when the name is too rare and given to less than 20 babies since 1900).

```
graph_freq(c("_PRENOMS_RARES"), FALSE)
```

```
## `summarise()` has grouped output by 'preusuel'. You can override using the
## `.groups` argument.
```



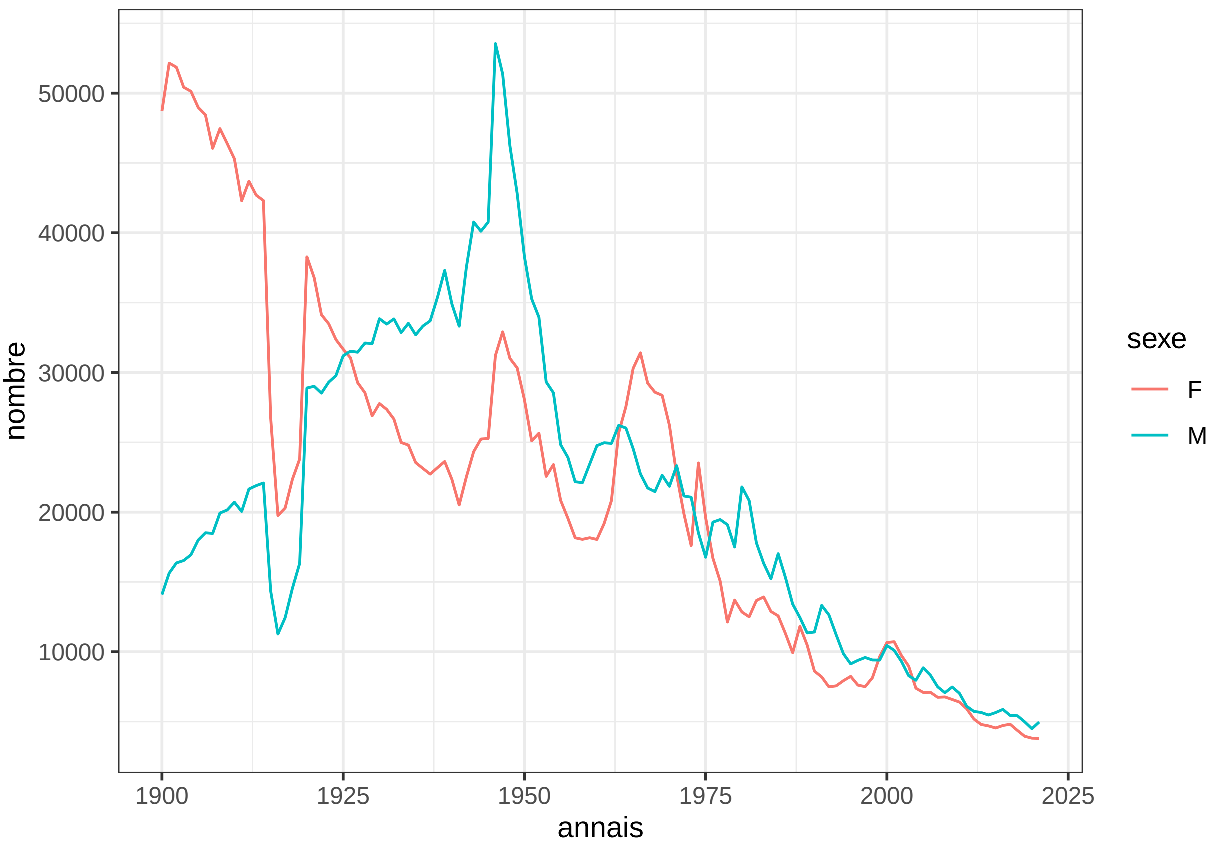
“Analyse the evolution of the most frequent firstname”

This part is not finished, as I am unsure if it is about the evolution of the first name given the most since 1900 (“Marie” and “Jean”), or the evolution of which name was the most given each year and its frequency (like the fact that over time the most given name is less frequent).

The following graph shows the evolution of how much the first name was given each year

```
MostGivenNamePerYear %>% select(-preusuel) %>% mutate(annais = as.integer(annais), sexe = intToSex[as.i

ggplot(EvolutionMostGivenName, aes(x=annais, y=nombre)) + geom_line(aes(color=sexe)) + theme_bw()
```



50000
40000
30000
20000
10000
nombre
1900
1925
1950
1975
2000
2025
annais
sexe
F
M