French given names per year per department

Lucas Mello Schnorr, Jean-Marc Vincent

October, 2022

Marah Analyize 2023

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)
```

Build the Dataframe from file , I will use the file that I downloaded already.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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 errors
```

```
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.
```

Let see summary about the data.

```
# Summary of the data
data_summary <- summary(FirstNames)
print(data_summary)
```

```
##       sexe         preusuel            annais              dpt
##  Min.   :1.000   Length:3784673     Length:3784673     Length:3784673
##  1st Qu.:1.000   Class :character   Class :character   Class :character
##  Median :2.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1.535
##  3rd Qu.:2.000
##  Max.   :2.000
##      nombre
##  Min.   :   3.0
##  1st Qu.:   4.0
##  Median :   7.0
##  Mean   :  23.1
##  3rd Qu.:  18.0
##  Max.   :6307.0
```

```
head_table <- head(FirstNames,10)
print(head_table)
```

```
## # A tibble: 10 x 5
##     sexe preusuel       annais dpt   nombre
##    <dbl> <chr>          <chr>  <chr>  <dbl>
##  1     1 _PRENOMS_RARES 1900   02         7
##  2     1 _PRENOMS_RARES 1900   04         9
##  3     1 _PRENOMS_RARES 1900   05         8
##  4     1 _PRENOMS_RARES 1900   06        23
##  5     1 _PRENOMS_RARES 1900   07         9
##  6     1 _PRENOMS_RARES 1900   08         4
##  7     1 _PRENOMS_RARES 1900   09         6
##  8     1 _PRENOMS_RARES 1900   10         3
##  9     1 _PRENOMS_RARES 1900   11        11
## 10     1 _PRENOMS_RARES 1900   12         7
```

```
library(conflicted)

conflict_prefer("filter", "dplyr")
```

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

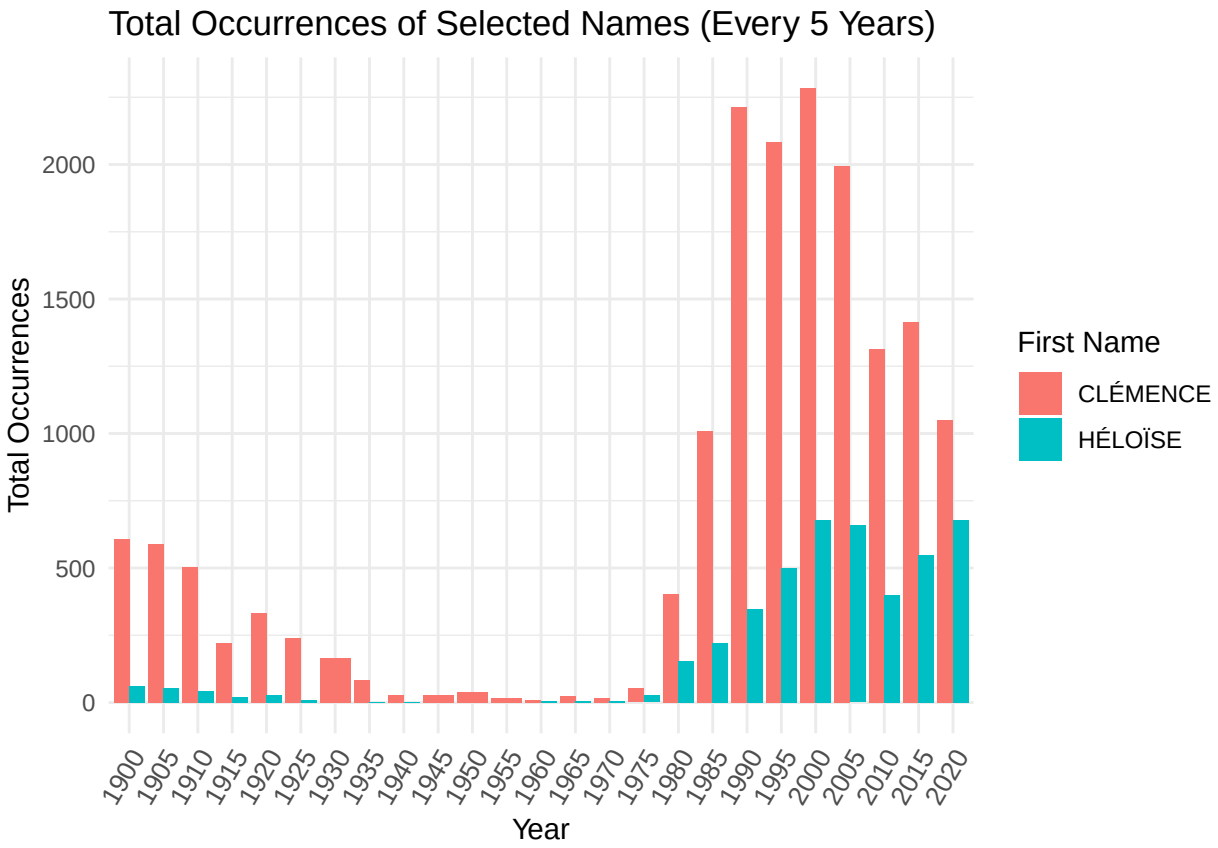
```
conflict_prefer("lag", "dplyr")
```

```
## [conflicted] Will prefer dplyr::lag over any other package.
```

1. Choose a firstname and analyse its frequency along time. Compare several firstnames frequency.

Answer 1:

```r
# Load required libraries
library(tidyverse)

# Read data with semicolon delimiter
FirstNames <- read_delim("dpt2021.csv", delim = ";", show_col_types = FALSE)

# Selected names
selected_names <- c("CLÉMENCE", "HÉLOÏSE")


# Filter data for the selected names and valid years
filtered_data <- FirstNames %>%
  filter(preusuel %in% selected_names) %>%
  filter(str_detect(annais, "^\\d{4}$")) %>%
  mutate(annais_numeric = as.numeric(annais)) %>%
  filter(annais_numeric %% 5 == 0)


# Group and summarize the filtered data
grouped_data <- filtered_data %>%
  group_by(annais, preusuel) %>%
  summarise(total_occurrences = sum(nombre, na.rm = TRUE), .groups = "drop")


# Create a bar chart with bars next to each other for each year
ggplot(grouped_data, aes(x = annais, y = total_occurrences, fill = preusuel)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Total Occurrences of Selected Names (Every 5 Years)",
       x = "Year",
       y = "Total Occurrences",
       fill = "First Name") +
  theme_minimal() +
  theme(axis.text.x = element_text(size = 10, angle = 60, hjust = 1))  # Adjust the size
```



We can see from this that CLEMENCE is the more frequent compared to HELOISE.

2. Establish by gender the most given firstname by year. Analyse the evolution of the most frequent firstname.

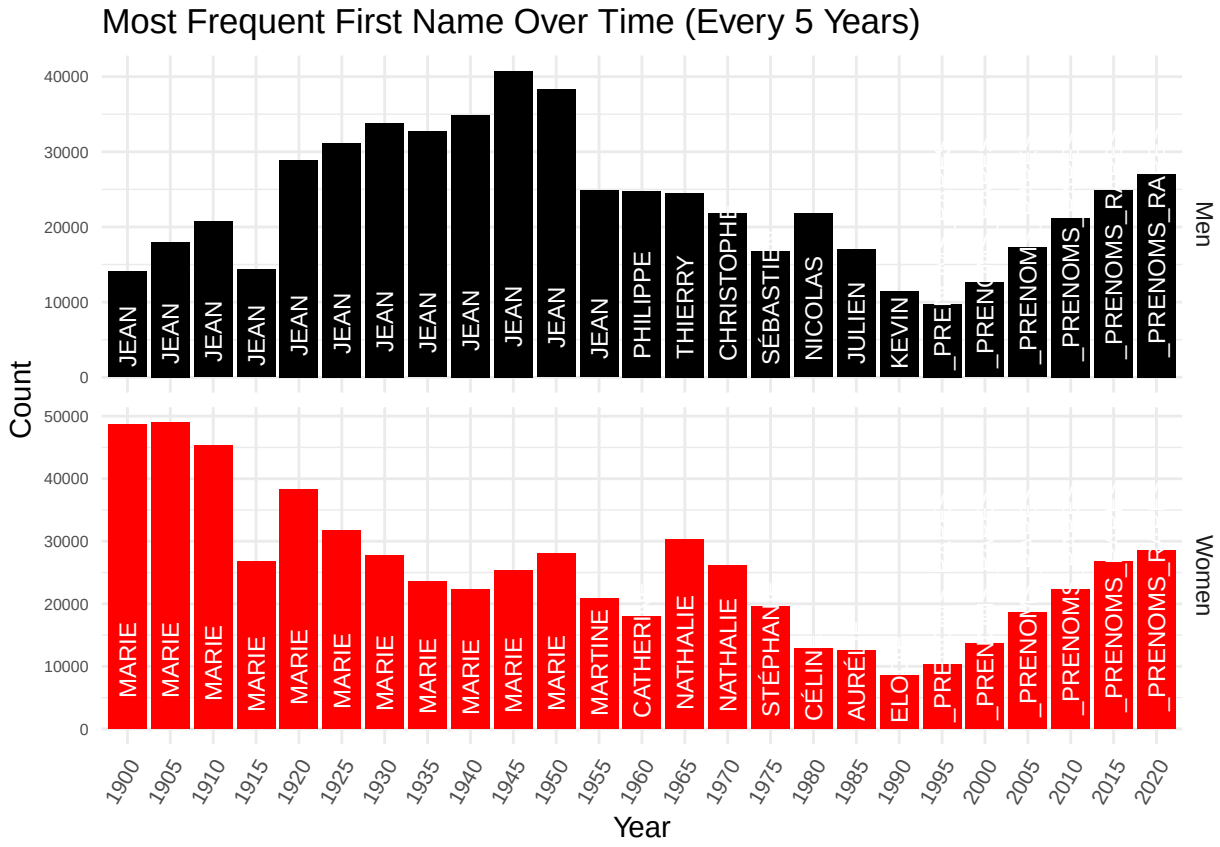
Answer 2:

```
# Remove rows with NA values in the 'annais' column
FirstNames <- FirstNames[complete.cases(FirstNames$annais), ]


# Find the most frequent name for each gender in each year
most_frequent_names <- FirstNames %>%
  group_by(sexe, annais, preusuel) %>%
  summarise(total_occurrences = sum(nombre), .groups = "drop") %>%
  group_by(sexe, annais) %>%
  arrange(desc(total_occurrences)) %>%
  slice(1) %>%
  ungroup()


# Convert 'annais' to numeric
most_frequent_names$annais <- as.numeric(as.character(most_frequent_names$annais))
```

```
## Warning: NAs introduits lors de la conversion automatique
```

```
most_frequent_names <- most_frequent_names %>%
  filter(annais %% 5 == 0)
```

```
# Determine the color for each year based on gender
most_frequent_names <- most_frequent_names %>%
  mutate(color = ifelse(sexe == 1, "black", "red"))


# Create a bar chart with facets for each gender
p<-ggplot(most_frequent_names, aes(x = as.factor(annais),
                                   y = total_occurrences, fill = color, label = preusuel)) +
  geom_bar(stat = "identity") +
  geom_text(position = position_stack(vjust = 0.1), size = 3,
            color = "white", angle = 90, hjust = 0) +  # Add labels vertically with white text
  labs(title = "Most Frequent First Name Over Time (Every 5 Years)",
       x = "Year",
       y = "Count") +
  scale_fill_identity() +
  theme_minimal() +
  theme(axis.text.x = element_text(size = 8, angle = 60, hjust = 1),
        axis.text.y = element_text(size = 6)) +  # Adjust text size for y-axis
  facet_grid(sexe ~ ., scales = "free_y" ,
             labeller = labeller(sexe = c("1" = "Men", "2" = "Women"))) +
  theme(legend.position = "none")  # Remove default legend

print(p)
```



We can see how Marie is the most frequent name in women , and Jean in men.

3. Optional : Which department has a larger variety of names along time ? Is there some sort of geographical correlation with the data?

Answer 3:

For this quuestion lets start by groupby on the dpt and annais column to see for each dpt in each year how many unique names the have.

```
FirstNames
```

```
## # A tibble: 3,784,673 x 5
##     sexe preusuel       annais dpt   nombre
##    <dbl> <chr>          <chr>  <chr>  <dbl>
##  1     1 _PRENOMS_RARES 1900   02         7
##  2     1 _PRENOMS_RARES 1900   04         9
##  3     1 _PRENOMS_RARES 1900   05         8
##  4     1 _PRENOMS_RARES 1900   06        23
##  5     1 _PRENOMS_RARES 1900   07         9
##  6     1 _PRENOMS_RARES 1900   08         4
##  7     1 _PRENOMS_RARES 1900   09         6
##  8     1 _PRENOMS_RARES 1900   10         3
##  9     1 _PRENOMS_RARES 1900   11        11
## 10     1 _PRENOMS_RARES 1900   12         7
## # i 3,784,663 more rows
```

```
FirstNames_grouped <- FirstNames %>%
  group_by(dpt, annais) %>%
  summarise(unique_names = n_distinct(preusuel)) %>%
  ungroup()
```

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

```
FirstNames_grouped
```

```
## # A tibble: 11,749 x 3
##    dpt   annais unique_names
##    <chr> <chr>         <int>
##  1 01    1900            162
##  2 01    1901            179
##  3 01    1902            172
##  4 01    1903            176
##  5 01    1904            176
##  6 01    1905            177
##  7 01    1906            181
##  8 01    1907            181
##  9 01    1908            175
## 10 01    1909            184
## # i 11,739 more rows
```

Now lets sum for each dpt over the year.

```
FirstNames_unique_names_per_dpt <- FirstNames_grouped %>%
  group_by(dpt) %>%
  summarise(total_unique_names = sum(unique_names))
```

```r
FirstNames_unique_names_per_dpt
```

```
## # A tibble: 101 x 2
##    dpt   total_unique_names
##    <chr>              <int>
##  1 01                 28061
##  2 02                 37926
##  3 03                 26479
##  4 04                 13231
##  5 05                 14909
##  6 06                 50192
##  7 07                 24792
##  8 08                 27050
##  9 09                 15272
## 10 10                 25588
## # i 91 more rows
```

```r
larger_variety_of_names <- max(FirstNames_unique_names_per_dpt$total_unique_names, na.rm = TRUE)

# Identify the row where total_unique_names equals the maximum value
max_rows_indices <- which(
  FirstNames_unique_names_per_dpt$total_unique_names == larger_variety_of_names)

# Subset the FirstNames_unique_names_per_dpt to get the row with the maximum total_unique_names
The_largest_variety_dpt <- FirstNames_unique_names_per_dpt[max_rows_indices, ]

The_largest_variety_dpt
```

```
## # A tibble: 1 x 2
##   dpt   total_unique_names
##   <chr>              <int>
## 1 75                127718
```

So it is department number 75.

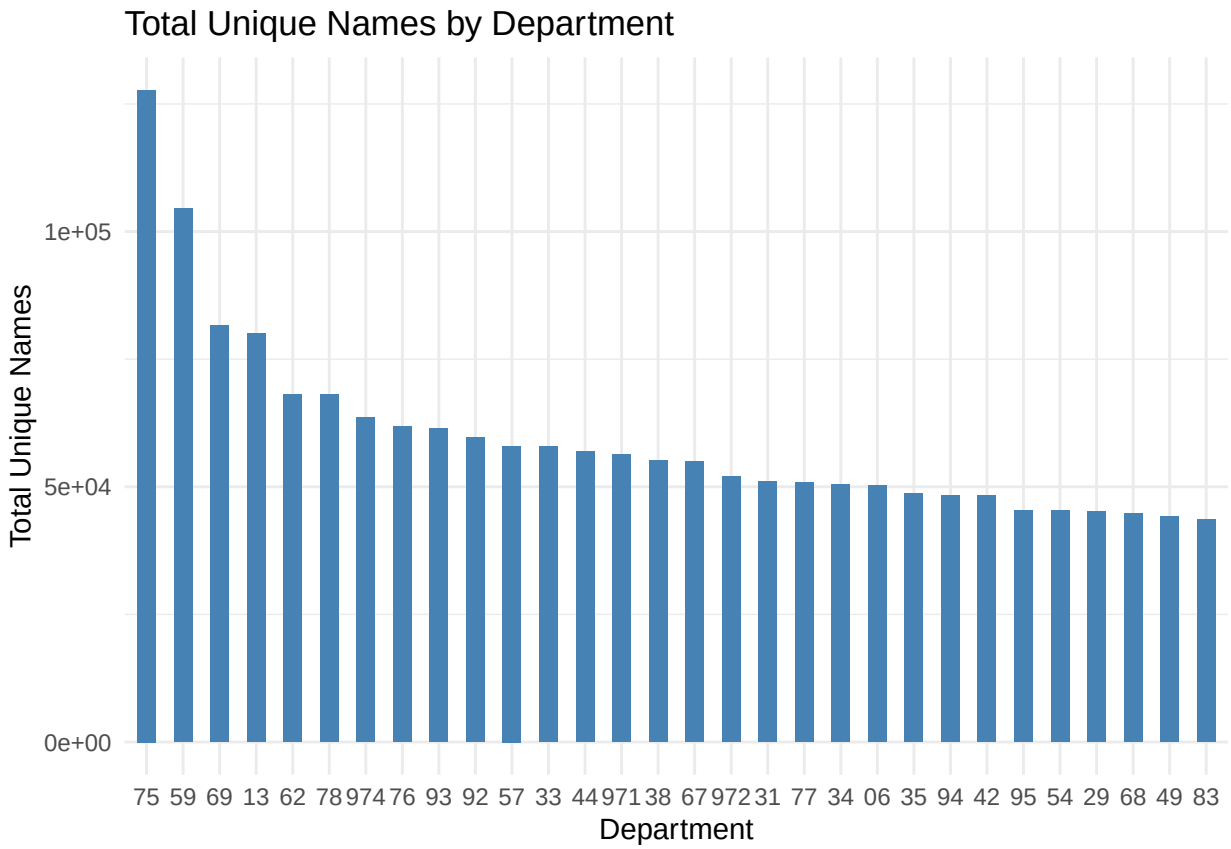
Let us draw the data .

```r
library(ggplot2)
top_30_df <- FirstNames_unique_names_per_dpt %>% slice_max(order_by = total_unique_names, n = 30)
top_30_df
```

```
## # A tibble: 30 x 2
##    dpt   total_unique_names
##    <chr>              <int>
##  1 75                127718
##  2 59                104495
##  3 69                 81655
##  4 13                 79962
##  5 62                 68155
##  6 78                 68072
##  7 974                63579
##  8 76                 61854
##  9 93                 61477
## 10 92                 59728
```

```
## # i 20 more rows
```

```
ggplot(top_30_df, aes(x = reorder(dpt, -total_unique_names), y = total_unique_names)) +
  geom_bar(stat = "identity", fill = "steelblue" , width=0.5) +
  theme_minimal() +
  labs(x = "Department", y = "Total Unique Names", title = "Total Unique Names by Department")
```



I don't think there is a geographical correlation ,to be honest, I didnt understand the meaning of the second part of the third question , I mean, I did not find any geographical data except the department and I do not know which department belongs to any city ,However I found different departments with the same distribution, so I answered that there is no geographical correlation.