French given names per year per department

Raised by Lucas Mello Schnorr, Jean-Marc Vincent; reviewed by Gregory James

October, 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)
```

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

expected : MD5 (dpt2021.csv) = f18a7d627883a0b248a0d59374f3bab7

Build the Dataframe from file

```
library(tidyverse)
library(dplyr)
library(ggplot2)
df_loaded <- read_delim("dpt2021.csv",delim=";",show_col_types = FALSE)
```

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?

Cleaning the Dataframe

The preliminary step before running some analysis on the evolution of first names in France, we have to first analyze and ensure that the database is structured in the right way with necessary cleaning performed. We are therefore first printing different details on the "dataframe loaded"

```
print(head(df_loaded))
```

```
## # A tibble: 6 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
```

```
print(tail(df_loaded))
```

```
## # A tibble: 6 x 5
##    sexe preusuel annais dpt   nombre
##   <dbl> <chr>    <chr>  <chr>  <dbl>
## 1     2 ZYA      2018   59         3
## 2     2 ZYA      2021   35         5
## 3     2 ZYA      XXXX   XX       278
## 4     2 ZYNA     2013   93         3
## 5     2 ZYNA     XXXX   XX        68
## 6     2 ZYNEB    XXXX   XX       125
```

```
dim(df_loaded)
```

```
## [1] 3784673       5
```

```
length(df_loaded)
```

```
## [1] 5
```

```
summary(df_loaded)
```

```
##       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
```

The first table is showing the value "_PRENOMS_RARES" as the firstname which we will investigate after cleaning some other fields.

Titles of the Table

Titles are not really explicit and I have made the decision to rename those as per below: - “sexe” changed to “gender” - “preusuel” changed to “firstname” - “annais changed to”year"
- “dpt” changed to “local_department” - “nombre” changed to “count_of_name”

In order to keep a trace of the original load I have created a new table called “df_loaded_cleaned”

```
df_loaded_cleaned <- df_loaded %>% rename(gender = sexe, firstname = preusuel, year =  annais,
                                  local_department = dpt, count_of_name = nombre)
print(tail(df_loaded_cleaned))
```

```
## # A tibble: 6 x 5
##   gender firstname year  local_department count_of_name
##    <dbl> <chr>     <chr> <chr>                    <dbl>
## 1      2 ZYA       2018  59                           3
## 2      2 ZYA       2021  35                           5
## 3      2 ZYA       XXXX  XX                         278
## 4      2 ZYNA      2013  93                           3
## 5      2 ZYNA      XXXX  XX                          68
## 6      2 ZYNEB     XXXX  XX                         125
```

Understanding and Cleaning “XXXX” and “XX” Values

It appears also that there are some data that need to be cleaned out of the table. I have decided to start investigating the “XXXX” and “XX” values. To further the investigation, I have looked into the detail for 3 names that are shown with their values. I am also testing if there are any combination for which we have “XXXX” in year and/or “XX” in local_department.

```
df_count_names <- df_loaded_cleaned %>% count(firstname, sort = TRUE, name = "rows_per_name")
set.seed(1)
name_tested_A <- df_count_names %>% filter(rows_per_name == 6) %>%
  sample_n(1, replace = FALSE) %>% select(1) %>% as.character()
set.seed(1)
name_tested_B <- df_count_names %>% filter(rows_per_name == 7) %>%
  sample_n(1, replace = FALSE) %>% select(1) %>% as.character()
set.seed(1)
name_tested_C <- df_count_names %>% filter(rows_per_name == 8) %>%
  sample_n(1, replace = FALSE) %>% select(1) %>% as.character()

count(df_loaded_cleaned, year == "XXXX" & local_department != "XX")
```

```
count(df_loaded_cleaned, year != "XXXX" & local_department == "XX")
```

```
count(df_loaded_cleaned, year != "XXXX" & local_department != "XX")
```

```
count(df_loaded_cleaned, year == "XXXX" & local_department == "XX")
```

```
print(df_loaded_cleaned %>% filter(firstname == name_tested_A))
```

```
## # A tibble: 6 x 5
##   gender firstname year  local_department count_of_name
##    <dbl> <chr>     <chr> <chr>                    <dbl>
## 1      1 DAWID     2008  75                           3
## 2      1 DAWID     2010  75                           4
## 3      1 DAWID     2011  75                           3
## 4      1 DAWID     2014  75                           3
## 5      1 DAWID     2020  93                           4
```

```
## 6      1 DAWID     XXXX  XX                          79
```

```
print(df_loaded_cleaned %>% filter(firstname == name_tested_B))
```

```
## # A tibble: 7 x 5
##   gender firstname year  local_department count_of_name
##    <dbl> <chr>     <chr> <chr>                    <dbl>
## 1      1 NIHED     XXXX  XX                          37
## 2      2 NIHED     2009  59                           6
## 3      2 NIHED     2010  69                           3
## 4      2 NIHED     2012  13                           3
## 5      2 NIHED     2016  69                           3
## 6      2 NIHED     2017  66                           3
## 7      2 NIHED     XXXX  XX                         230
```

```
print(df_loaded_cleaned %>% filter(firstname == name_tested_C))
```

```
## # A tibble: 8 x 5
##   gender firstname year  local_department count_of_name
##    <dbl> <chr>     <chr> <chr>                    <dbl>
## 1      2 SIRYNE    2006  69                           3
## 2      2 SIRYNE    2006  93                           3
## 3      2 SIRYNE    2007  13                           3
## 4      2 SIRYNE    2010  13                           3
## 5      2 SIRYNE    2011  69                           3
## 6      2 SIRYNE    2012  93                           3
## 7      2 SIRYNE    2013  75                           4
## 8      2 SIRYNE    XXXX  XX                         272
```

The outcome of that investigation is that any logical information can established but an assumption can be made. I have considered that those records correspond, for a first name, to all the records for which the year and/or the department haven't been recorded. To avoid removing a data that might be useful in the analysis I have decided to replace "XXXX","XX" by "9999".

```
df_loaded_cleaned$year[df_loaded_cleaned$year == "XXXX"] <- "9999"
df_loaded_cleaned$local_department[df_loaded_cleaned$local_department == "XX"] <- "9999"
print(df_loaded_cleaned %>% filter(firstname == name_tested_A))
```

```
## # A tibble: 6 x 5
##   gender firstname year  local_department count_of_name
##    <dbl> <chr>     <chr> <chr>                    <dbl>
## 1      1 DAWID     2008  75                           3
## 2      1 DAWID     2010  75                           4
## 3      1 DAWID     2011  75                           3
## 4      1 DAWID     2014  75                           3
## 5      1 DAWID     2020  93                           4
## 6      1 DAWID     9999  9999                        79
```

Format of the Data

In addition I have also seen that the column that includes the year information is shown as "chr" which might generate wrong analysis if we want to use that base and compare to an evolution over time. I am therefore changing this column to become an integer ("int").

```
set.seed(7)
df_loaded_cleaned$year <- as.numeric(df_loaded_cleaned$year)
names_tested_A <- df_count_names %>% filter(rows_per_name == 7) %>%
```

```
  sample_n(1, replace = FALSE) %>% select(1) %>% as.character()
print(df_loaded_cleaned %>% filter(firstname == names_tested_A))
```

```
## # A tibble: 7 x 5
##   gender firstname  year local_department count_of_name
##    <dbl> <chr>     <dbl> <chr>                    <dbl>
## 1      2 MIKELA     1996 64                           3
## 2      2 MIKELA     1998 64                           3
## 3      2 MIKELA     2009 64                           3
## 4      2 MIKELA     2010 64                           4
## 5      2 MIKELA     2016 64                           3
## 6      2 MIKELA     2017 64                           4
## 7      2 MIKELA     9999 9999                        60
```

```
any(is.na(df_loaded_cleaned))
```

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

```
which(is.na(df_loaded_cleaned))
```

```
## [1] 7108014
```

It appears that there are some NA values in the table. I am therefore looking in every columns to see where there are some and identify the action to take with those values.

```
paste0("Test NA in column 'gender' =",any(is.na(df_loaded_cleaned$gender)))
```

```
## [1] "Test NA in column 'gender' =FALSE"
```

```
paste0("Test NA in column 'firstname' =",any(is.na(df_loaded_cleaned$firstname)))
```

```
## [1] "Test NA in column 'firstname' =TRUE"
```

```
paste0("Test NA in column 'year' =",any(is.na(df_loaded_cleaned$year)))
```

```
## [1] "Test NA in column 'year' =FALSE"
```

```
paste0("Test NA in column 'local_department' =",any(is.na(df_loaded_cleaned$local_department)))
```

```
## [1] "Test NA in column 'local_department' =FALSE"
```

```
paste0("Test NA in column 'count_of_name' =",any(is.na(df_loaded_cleaned$count_of_name)))
```

```
## [1] "Test NA in column 'count_of_name' =FALSE"
```

```
count(df_loaded_cleaned, is.na(df_loaded_cleaned$firstname))
```

```
which(is.na(df_loaded_cleaned$firstname))
```

```
## [1] 3323341
```

```
df_loaded_cleaned[3323341,]
```

Only one firstname is shown as NA and looking at the information it seems that there are some records for which names are missing. I am again updating this value with "9999" to make sure that we can then use the base appropriately.

```
df_loaded_cleaned[3323341,2] <- "9999"
df_loaded_cleaned[3323341,]
```

```
print(head(df_loaded_cleaned))
```

```
## # A tibble: 6 x 5
##   gender firstname       year local_department count_of_name
##    <dbl> <chr>          <dbl> <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
```

```
print(tail(df_loaded_cleaned))
```

```
## # A tibble: 6 x 5
##   gender firstname  year local_department count_of_name
##    <dbl> <chr>     <dbl> <chr>                     <dbl>
## 1      2 ZYA        2018 59                            3
## 2      2 ZYA        2021 35                            5
## 3      2 ZYA        9999 9999                        278
## 4      2 ZYNA       2013 93                            3
## 5      2 ZYNA       9999 9999                         68
## 6      2 ZYNEB      9999 9999                        125
```

```
dim(df_loaded_cleaned)
```

```
## [1] 3784673       5
```

```
length(df_loaded_cleaned)
```

```
## [1] 5
```

```
summary(df_loaded_cleaned)
```

```
##      gender       firstname              year      local_department
##  Min.   :1.000   Length:3784673     Min.   :1900   Length:3784673
##  1st Qu.:1.000   Class :character   1st Qu.:1949   Class :character
##  Median :2.000   Mode  :character   Median :1982   Mode  :character
##  Mean   :1.535                      Mean   :2056
##  3rd Qu.:2.000                      3rd Qu.:2004
##  Max.   :2.000                      Max.   :9999
##  count_of_name
##  Min.   :   3.0
##  1st Qu.:   4.0
##  Median :   7.0
##  Mean   :  23.1
##  3rd Qu.:  18.0
##  Max.   :6307.0
```

Rare First Names

When running the first review of the table it appears that the name "_PRENOMS_RARES" is shown as the first name. I have excluded this value and run a summary to see if it is linked to the count of names.

```
firstname_count_mapping <- df_loaded_cleaned %>% select(!(gender)) %>% select(!(year)) %>%
  select(!(local_department))
firstname_count_mapping_exc_rare <- firstname_count_mapping[firstname_count_mapping$firstname !=
                                                              "_PRENOMS_RARES", ]
summary(na.omit(firstname_count_mapping_exc_rare))
```

```
##   firstname         count_of_name
##  Length:3762419     Min.   :   3.00
##  Class :character   1st Qu.:   4.00
##  Mode  :character   Median :   7.00
##                     Mean   :  22.78
##                     3rd Qu.:  18.00
##                     Max.   :6307.00
```

We can clearly see that first names that are registered are the one that have more than 2 records in the given range (gender, year, department). All the others are registered as "_PRENOMS_RARES". To be sure that we are having all the data and numbers available for the analysis i have decided to keep the data as it but can still remove this value when needed.

After all those changes the inital dataframe structured is preserved but the data is in a format that can be used for making further analysis on the repartition of first names.

Analysis over the first name frequency along time.

For this analysis the local_department nor the gender information are needed. As a starting point I have created a new table and summed up all the count of names per year and remove the local_department and gender_data.

```
table_freq_analysis <- df_loaded_cleaned %>% select(!(gender)) %>% select(!(local_department)) %>%
  group_by(firstname,year,.add = TRUE,.drop = FALSE) %>%
  mutate(sum_per_year = sum(count_of_name)) %>% select(!(count_of_name))
table_freq_analysis <- table_freq_analysis[!duplicated(table_freq_analysis),]
summary(table_freq_analysis)
```

```
##   firstname              year       sum_per_year
##  Length:292254      Min.   :1900   Min.   :    3.0
##  Class :character   1st Qu.:1959   1st Qu.:    5.0
##  Mode  :character   Median :1993   Median :   19.0
##                     Mean   :2970   Mean   :  299.1
##                     3rd Qu.:2014   3rd Qu.:   86.0
##                     Max.   :9999   Max.   :56152.0
```

As we are going to look at the distribution per year I have also removed the data for which the year is '9999'

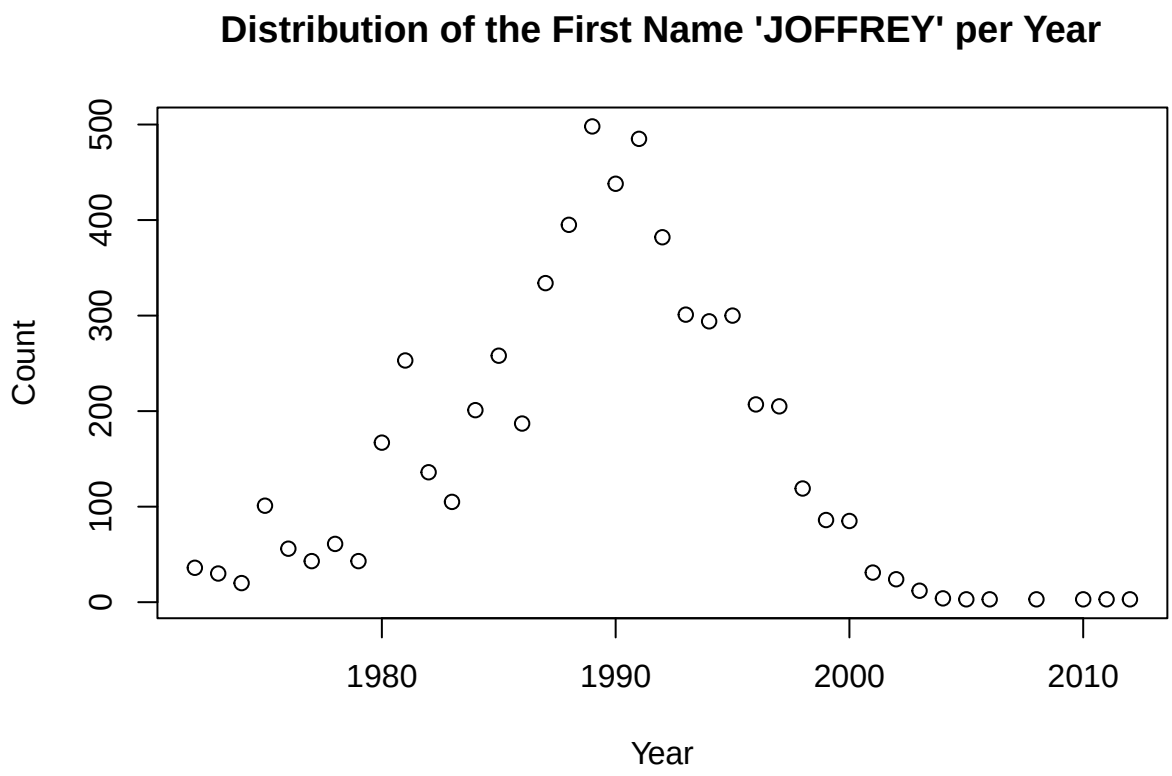
```
table_freq_analysis <- table_freq_analysis[table_freq_analysis$year != 9999, ]
summary(table_freq_analysis)
```

```
##   firstname              year       sum_per_year
##  Length:256083      Min.   :1900   Min.   :    3.0
##  Class :character   1st Qu.:1954   1st Qu.:    4.0
##  Mode  :character   Median :1986   Median :   13.0
##                     Mean   :1978   Mean   :  306.6
##                     3rd Qu.:2007   3rd Qu.:   71.0
##                     Max.   :2021   Max.   :56152.0
```

```
filter_table_name = function (name, tabx){
  FirstNameFiltered <- tabx %>% filter(firstname==name)
}
```

```
set.seed(6)
randomrows <- sample(nrow(table_freq_analysis))
table_freq_analysis_random <- table_freq_analysis[randomrows, ]
list_firstnames_random <- unique(table_freq_analysis_random[,1])
```

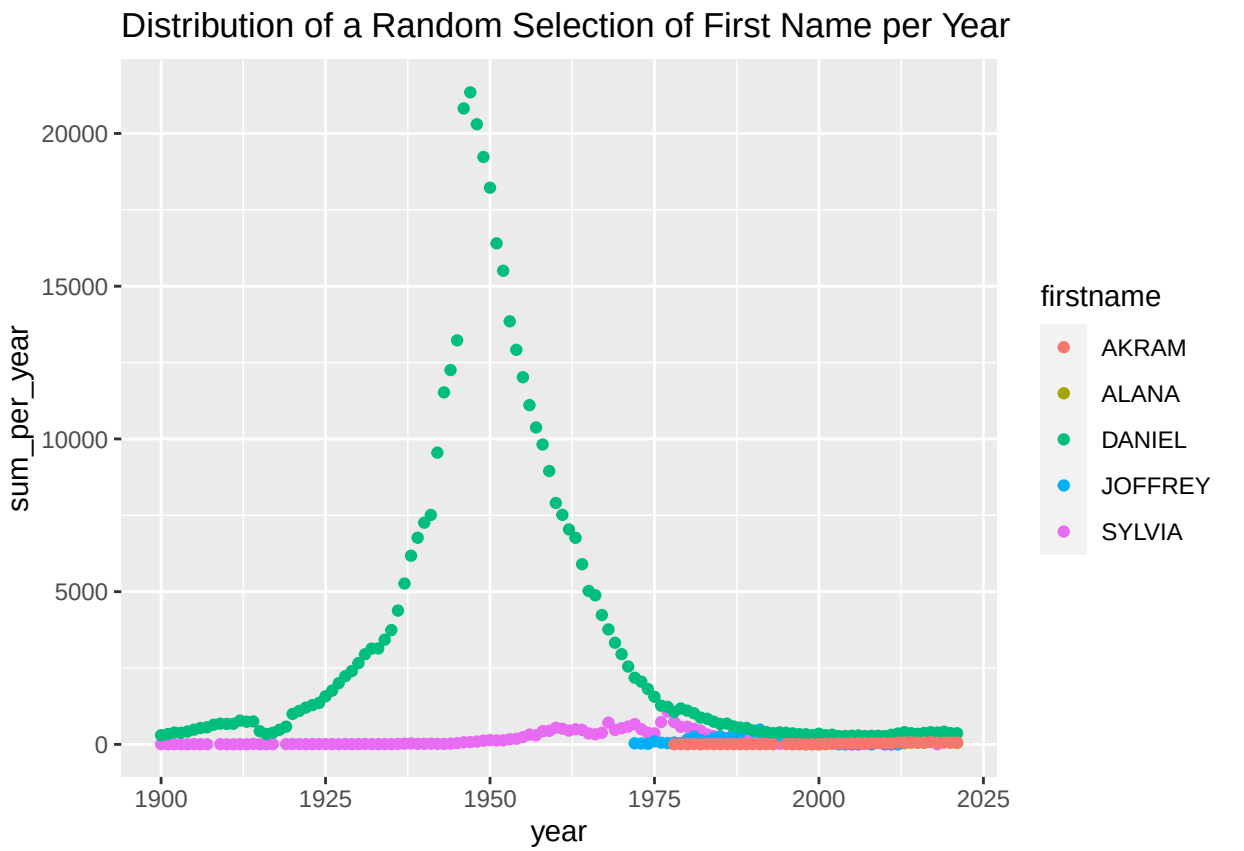
```
table_freq_analysis_names <- table_freq_analysis[0,]
for (each_rows in sample(5)){
  names_selected_x <- list_firstnames_random[each_rows,1] %>% as.character()
  table_freq_analysis_names <- table_freq_analysis_names %>%
    rbind(filter_table_name(names_selected_x,table_freq_analysis))
  }

table_freq_analysis_name_1 <- table_freq_analysis_names %>%
  filter(firstname == list_firstnames_random[1,1])

plot(table_freq_analysis_name_1$year,table_freq_analysis_name_1$sum_per_year,
     xlab="Year", ylab="Count", main = paste0("Distribution of the First Name '",
                   list_firstnames_random[1,1],"' per Year"))
```



A random first name "JOFFREY" is selected and a graph shows, using a plot, how that first name has evolved over time with a pic around 1990.

I have also made a random selection of 5 names and run the same graph.

```
ggplot(data = table_freq_analysis_names, aes(x=year, y=sum_per_year, color=firstname)) +
  geom_point() + ggtitle("Distribution of a Random Selection of First Name per Year")
```



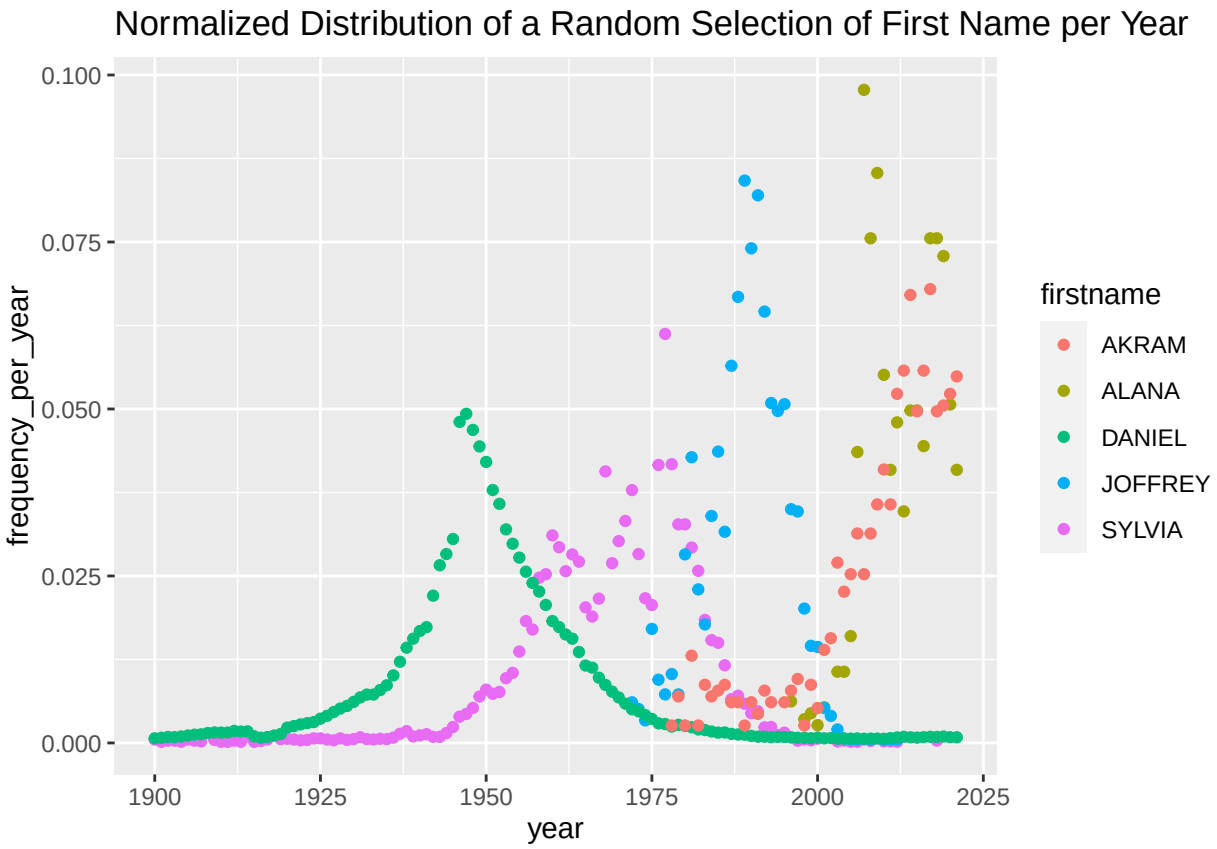
This type of graph allows to see that some first name are "stars" during some periods ("DANIEL" have been given a lot around 1950). We can also see that some are appearing/diseappiring along time. We still need to find another way to represent the data as when a name is having a big number, it is compressing all the other data. This is especially the around 1950's, due to baby boom, during when a much higher number of first names have been given.

Compare several first names frequency

In order to get a smoother vision over the data, I have added the frequency of a first name overtime. We can again plot the data to see the result.

```
table_freq_analysis <- table_freq_analysis %>% group_by(firstname) %>%
  mutate(sum_per_name = sum(sum_per_year)) %>%
  mutate(frequency_per_year = sum_per_year/sum_per_name)

set.seed(6)
randomrows <- sample(nrow(table_freq_analysis))
table_freq_analysis_random <- table_freq_analysis[randomrows, ]
list_firstnames_random <- unique(table_freq_analysis_random[,1])

table_freq_analysis_names <- table_freq_analysis[0,]
for (each_rows in sample(5)){
  names_selected_x <- list_firstnames_random[each_rows,1] %>% as.character()
  table_freq_analysis_names <- table_freq_analysis_names %>%
    rbind(filter_table_name(names_selected_x,table_freq_analysis))
  }
ggplot(data = table_freq_analysis_names, aes(x=year, y=frequency_per_year, color=firstname)) +
```

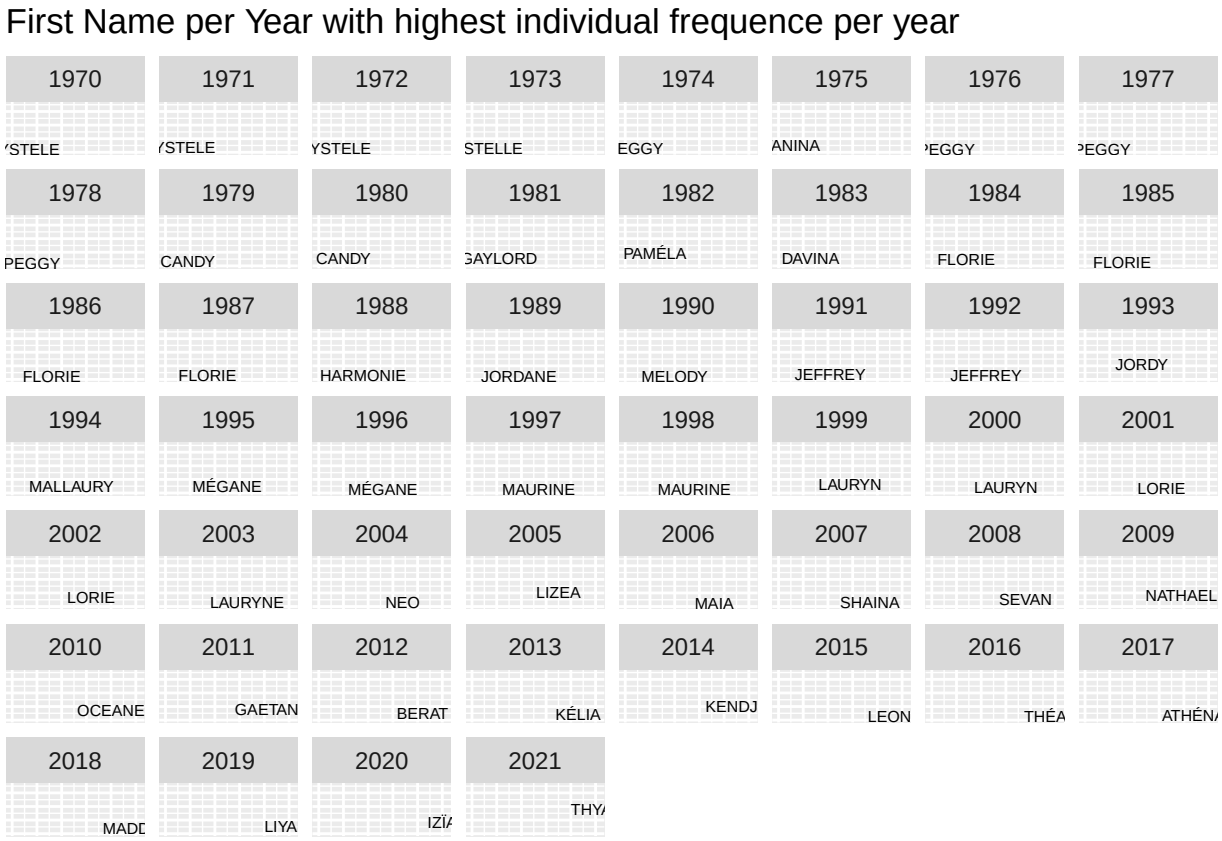
```
  geom_point() + ggtitle("Normalized Distribution of a Random Selection of First Name per Year")
```



We can see more details on each names and easily identify when they had their "star" moments.

We can also look at the first name that have been given more than 150 times in a year and where there is the higher frequence to have a look at the first names that had their "star moments" each year.

```
table_max_freq_analysis_names <- table_freq_analysis %>% filter(sum_per_year > 150) %>%
  group_by(year) %>% top_n(1, frequency_per_year)
ggplot(subset(table_max_freq_analysis_names, year >= 1970 & year <= 2021),
       aes(x=year, y=frequency_per_year,label = firstname)) +
  ggtitle("First Name per Year with highest individual frequence per year")+
  geom_text(size = 2,check_overlap = FALSE,position = position_stack(vjust = 0.5))+
  facet_wrap(~year)+theme(axis.title.x=element_blank(),axis.text.x=element_blank(),axis.ticks.x=element_
  theme(axis.title.y=element_blank(),axis.text.y=element_blank(),axis.ticks.y=element_blank())
```



Most Given First Name by Gender

We are now trying to find the most given first name by gender. The exact same steps done before are used without removing the gender information. The value with "_PRENOMS_RARES" will also impact the result so I am removing it. In addition, the table with the first name with the first name most given by gender in a year.

```
table_most_analysis <- df_loaded_cleaned %>% select(!(local_department)) %>%
  group_by(gender,firstname,year,.add = TRUE,.drop = FALSE) %>%
  mutate(sum_per_year = sum(count_of_name)) %>% select(!(count_of_name))
table_most_analysis <- table_most_analysis[!duplicated(table_most_analysis),]
table_most_analysis <- table_most_analysis[table_most_analysis$year != 9999, ]
table_most_analysis <- table_most_analysis[table_most_analysis$firstname != "_PRENOMS_RARES", ]
table_most_analysis <- table_most_analysis %>% ungroup() %>%
  group_by(gender,year) %>% top_n(1, sum_per_year)
print(tail(table_most_analysis))
```

```
## # A tibble: 6 x 4
## # Groups:   gender, year [6]
##   gender firstname  year sum_per_year
##    <dbl> <chr>     <dbl>        <dbl>
## 1      2 STÉPHANIE  1976        16697
## 2      2 STÉPHANIE  1977        15056
## 3      2 SYLVIE     1961        19189
## 4      2 SYLVIE     1962        20823
## 5      2 SYLVIE     1963        25669
## 6      2 SYLVIE     1964        27554
```

```
ggplot(subset(table_most_analysis, gender == 1), aes(x=gender, y=gender,label = firstname)) +
  geom_text(size = 2,check_overlap = FALSE,position = position_stack(vjust = 0.5))+
  ggtitle("Female: Most Given First Names per Year")+
  facet_wrap(~year)+theme(axis.title.x=element_blank()
                          ,axis.text.x=element_blank(),axis.ticks.x=element_blank())+
  theme(axis.title.y=element_blank(),axis.text.y=element_blank()
        ,axis.ticks.y=element_blank(),strip.text = element_text(size=6) )
```

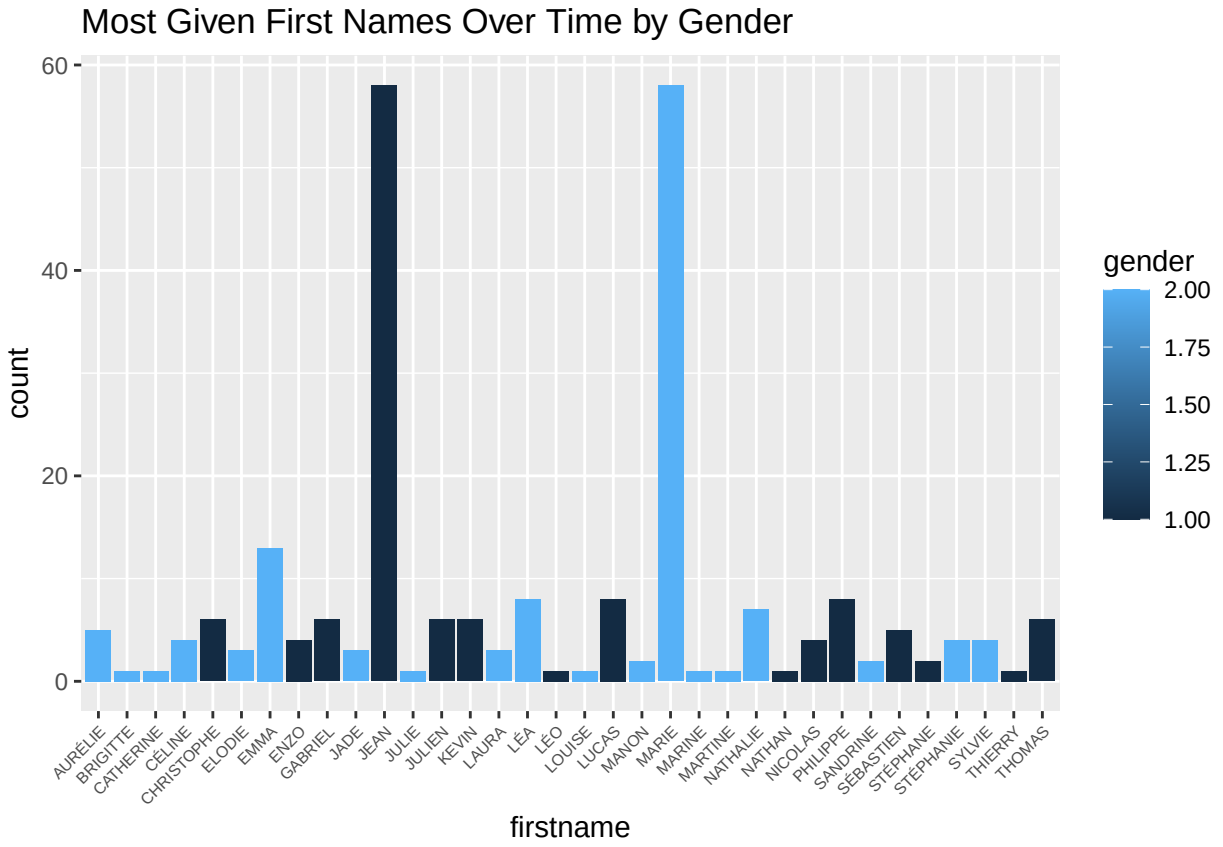

```
ggplot(subset(table_most_analysis, gender == 2), aes(x=gender, y=gender,label = firstname)) +
  ggtitle("Male: Most Given First Names per Year")+
  geom_text(size = 2,check_overlap = FALSE,position = position_stack(vjust = 0.5))+
  facet_wrap(~year)+theme(axis.title.x=element_blank()
                          ,axis.text.x=element_blank(),axis.ticks.x=element_blank())+
  theme(axis.title.y=element_blank(),axis.text.y=element_blank(),axis.ticks.y=element_blank()
        ,strip.text = element_text(size=6) )
```

Male: Most Given First Names per Year

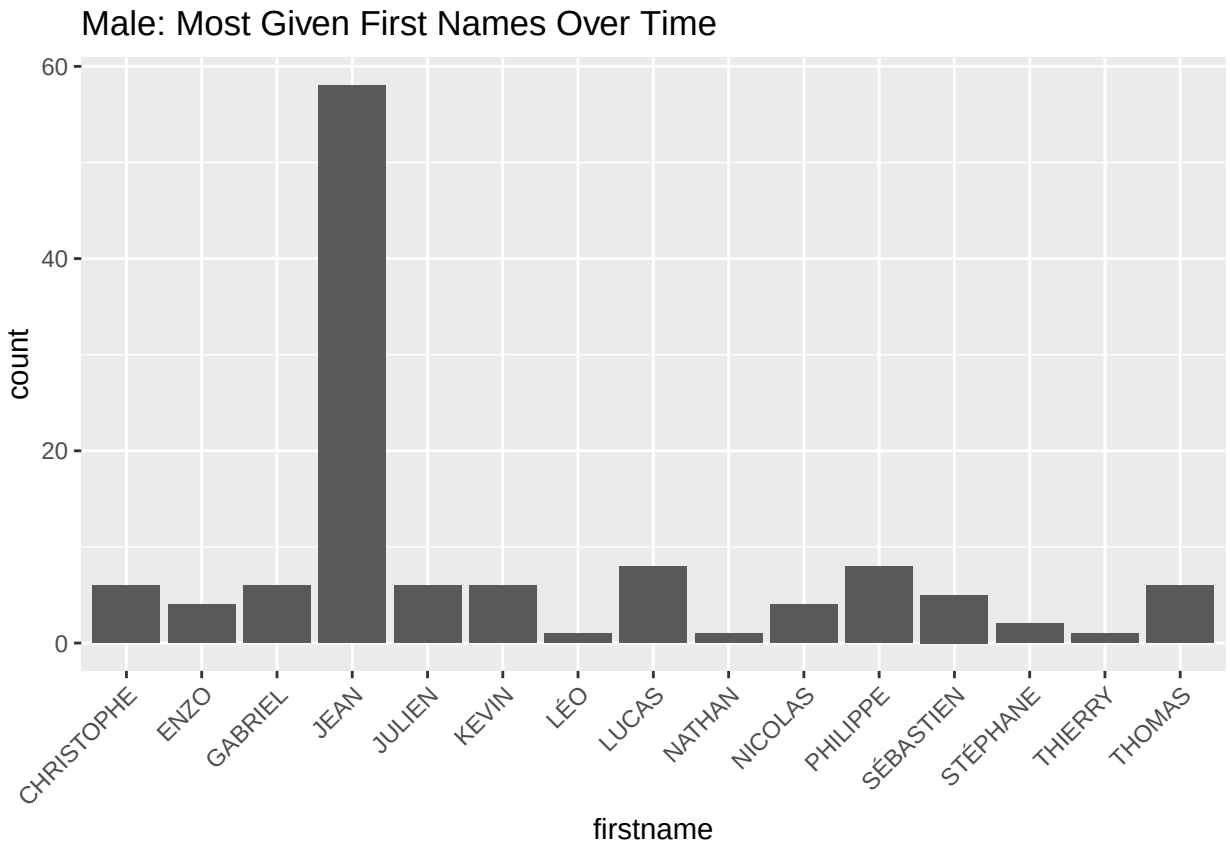
1900	1901	1902	1903	1904	1905	1906	1907	1908	1909	1910	1911
MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE
1912	1913	1914	1915	1916	1917	1918	1919	1920	1921	1922	1923
MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE
1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE
1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947
MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE
1948	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959
MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARIE	MARTINE	MARIE	MARIE	MARIE	BRIGITTE
1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971
CATHERINE	SYLVIE	SYLVIE	SYLVIE	SYLVIE	NATHALIE	NATHALIE	NATHALIE	NATHALIE	NATHALIE	NATHALIE	NATHALIE
1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
SANDRINE	SANDRINE	STÉPHANIE	STÉPHANIE	STÉPHANIE	STÉPHANIE	CÉLINE	CÉLINE	CÉLINE	CÉLINE	AURÉLIE	AURÉLIE
1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995
AURÉLIE	AURÉLIE	AURÉLIE	JULIE	ELODIE	ELODIE	ELODIE	MARINE	LAURA	LAURA	LAURA	MANON
1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007
MANON	LÉA	LÉA	LÉA	LÉA	LÉA	LÉA	LÉA	LÉA	EMMA	EMMA	EMMA
2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
EMMA	EMMA	EMMA	EMMA	EMMA	EMMA	JADE	LOUISE	EMMA	EMMA	EMMA	EMMA
2020	2021										
JADE	JADE										

We can see that there are some trends where the top first name is rarely at the top only one year. This is even more the case in the early of the 20th century when “JEAN” and “MARIE” have been given a lot during that period.

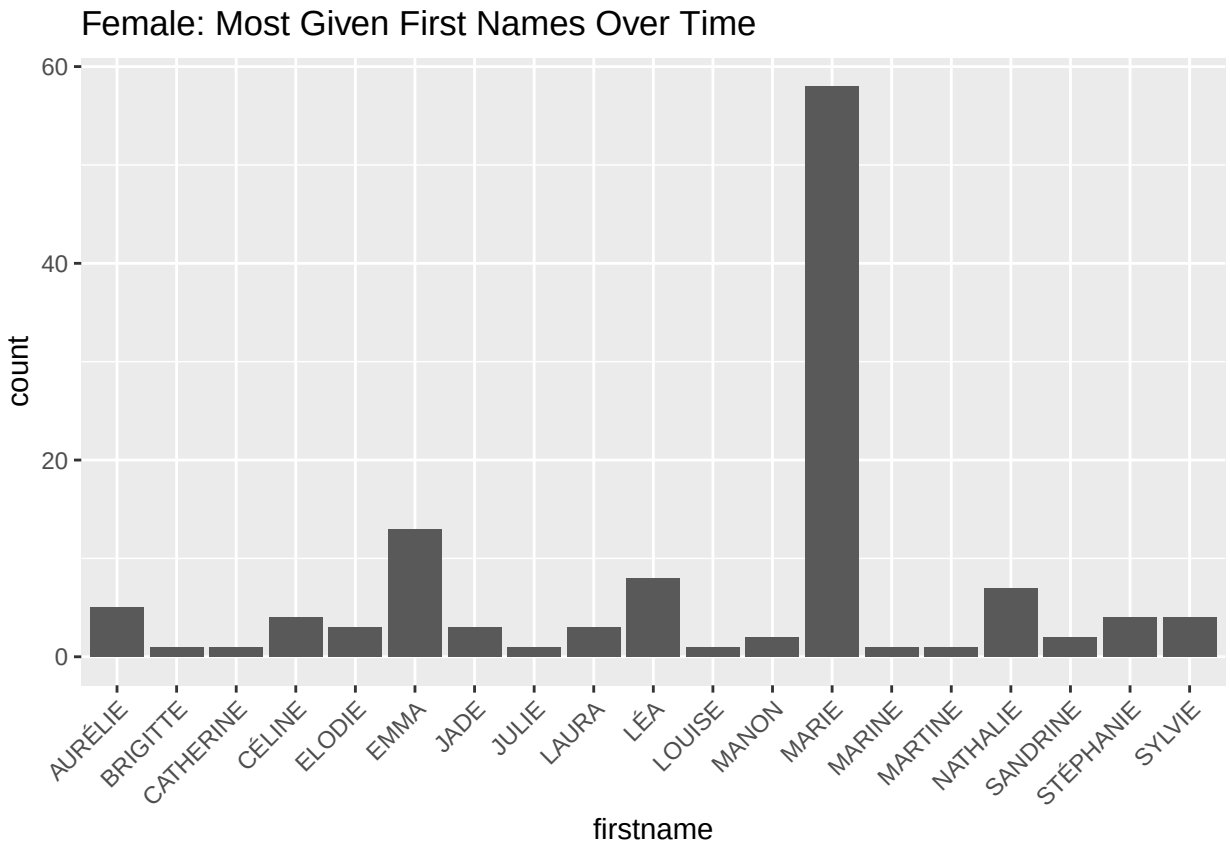
```
ggplot(subset(table_most_analysis), aes(x=firstname, fill=gender)) +
  geom_bar() + ggtitle("Most Given First Names Over Time by Gender")+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, size = 6))
```




```
ggplot(subset(table_most_analysis, gender == 1), aes(x=firstname)) +
  geom_bar() + ggtitle("Male: Most Given First Names Over Time")+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```




```
ggplot(subset(table_most_analysis, gender == 2), aes(x=firstname)) +
  geom_bar() + ggtitle("Female: Most Given First Names Over Time")+
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```



When grouping by first name we can see that there are not too many variations in the list the "most" given names.

Department With Largest Variety of Names

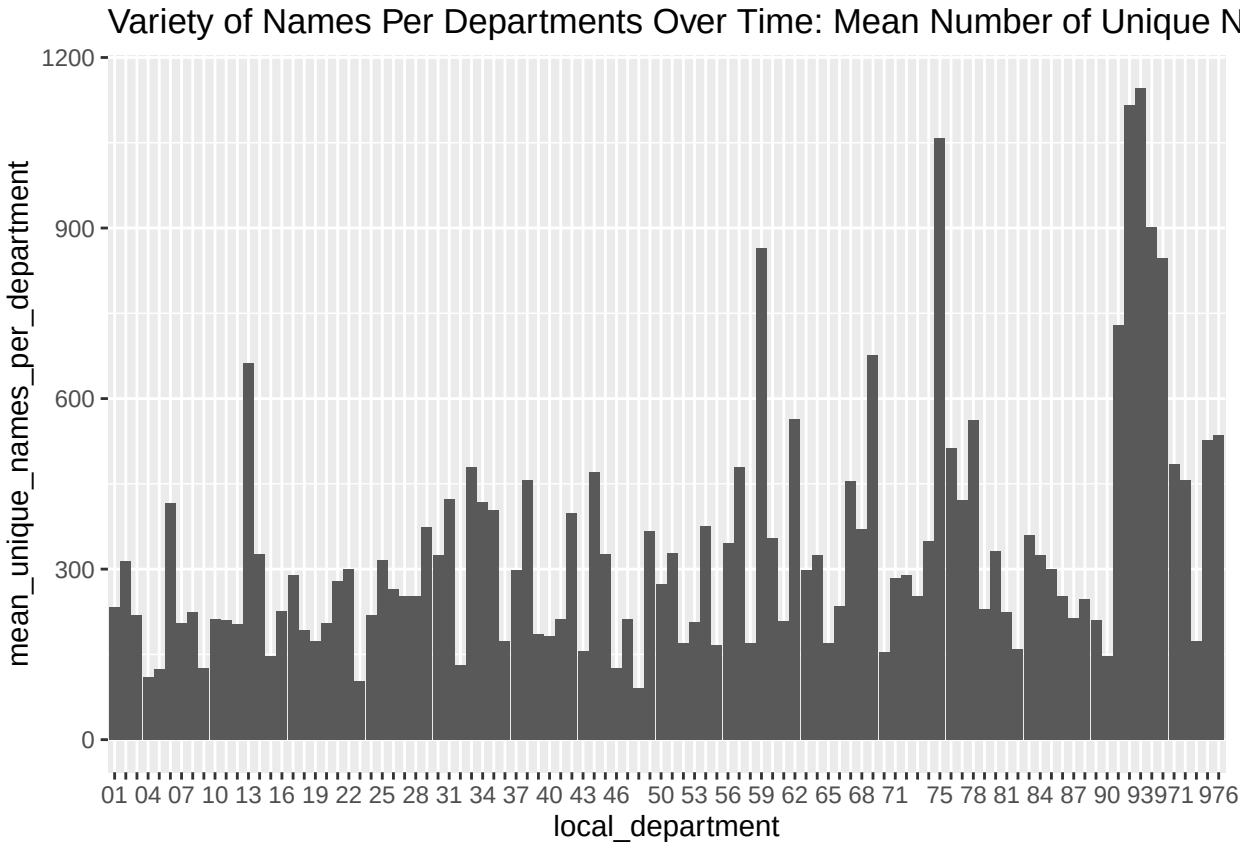
In order to look at how names are distributed within each departments we will again rebuilt a table using the same logic without the "99" and "9999" data. We are removing the count of each names and adding a new columns with: - total number of names given (number of birth) - number of different first names per year and department We can keep the "_PRENOMS_RARES" as it will be counted once only per department and will add information to the dataset.

We are searching which departments have had the largest variety over time by looking at the uniquess disctribution of first names over time.

```
table_by_dpt_analysis <- df_loaded_cleaned %>% select(!(gender)) %>%
  group_by(local_department,year,.add = TRUE,.drop = FALSE) %>%
   mutate(unique_names_per_department = n()) %>%
   mutate(total_birth_per_department = sum(count_of_name)) %>%
  select(!(firstname)) %>% select(!(count_of_name))
table_by_dpt_analysis <- table_by_dpt_analysis[!duplicated(table_by_dpt_analysis),]
table_by_dpt_analysis <- table_by_dpt_analysis %>% ungroup() %>%
  group_by(local_department,.add = TRUE,.drop = FALSE) %>%
   select(local_department , total_birth_per_department , unique_names_per_department) %>%
  summarize(iterations = n(),mean_unique_names_per_department = mean(unique_names_per_department),
            sd_unique_names_per_department = sd(unique_names_per_department))
table_by_dpt_analysis <- table_by_dpt_analysis[table_by_dpt_analysis$local_department != 9999, ]
print(table_by_dpt_analysis)
```

```
## # A tibble: 100 x 4
##    local_department iterations mean_unique_names_per_de~1 sd_unique_names_per_~2
##    <chr>                 <int>                       <dbl>                   <dbl>
##  1 01                      122                        232.                    65.5
##  2 02                      122                        314.                    75.9
##  3 03                      122                        219.                    47.5
##  4 04                      122                        110.                    15.2
##  5 05                      122                        123.                    15.9
##  6 06                      122                        415.                   226.
##  7 07                      122                        205.                    39.1
##  8 08                      122                        224.                    54.0
##  9 09                      122                        126.                    17.6
## 10 10                      122                        212.                    57.0
## # i 90 more rows
## # i abbreviated names: 1: mean_unique_names_per_department,
## #   2: sd_unique_names_per_department
```

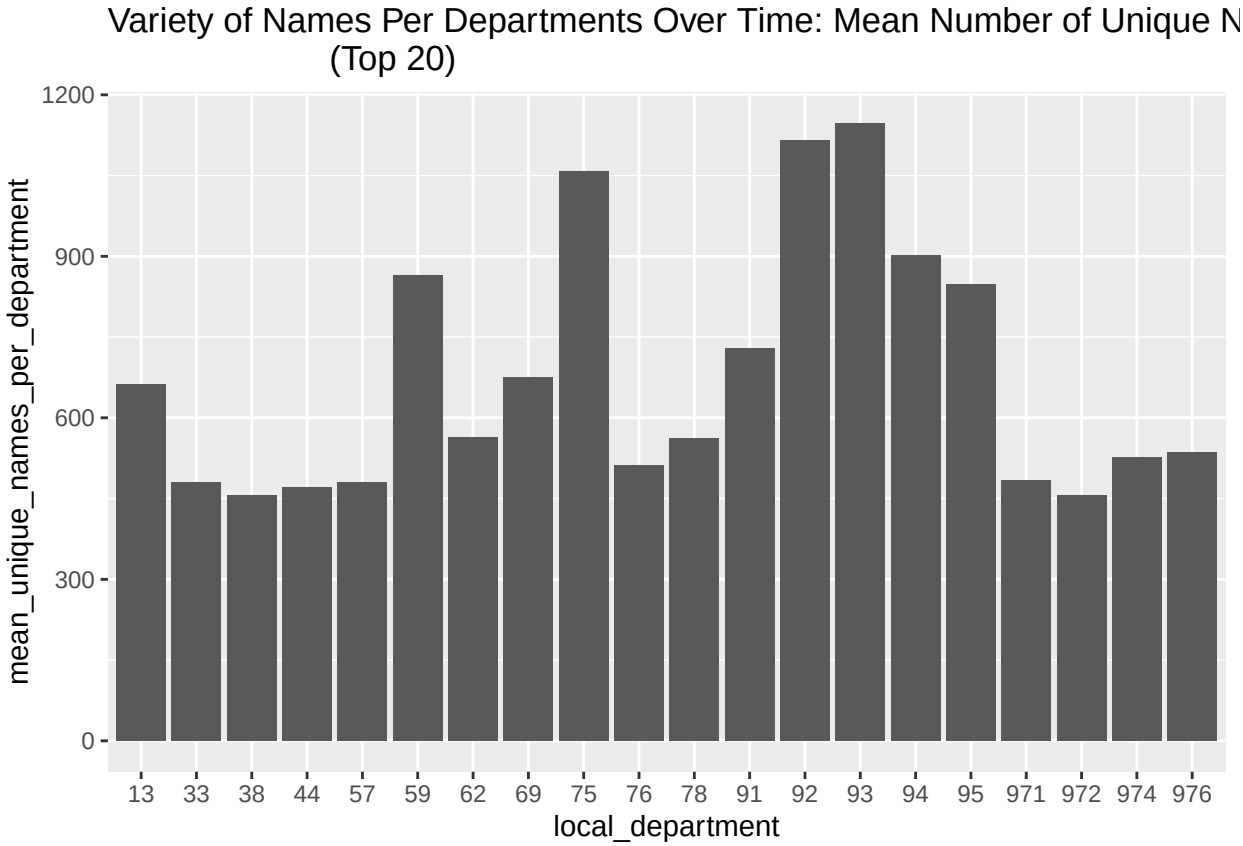
```
ggplot(data = table_by_dpt_analysis, aes(x=local_department, y=mean_unique_names_per_department)) +
  geom_col() + scale_x_discrete(guide = guide_axis(check.overlap = TRUE)) +
  ggtitle("Variety of Names Per Departments Over Time: Mean Number of Unique Names")
```



We can seems some variations over the number of department which could be also linked to where the biggest number of birth have happened.

```
table_by_dpt_analysis_top20 <- table_by_dpt_analysis %>%
  select(local_department,mean_unique_names_per_department)
table_by_dpt_analysis_top20 <- table_by_dpt_analysis_top20[!duplicated(table_by_dpt_analysis_top20),]
```

```r
table_by_dpt_analysis_top20 <- table_by_dpt_analysis_top20 %>%
  top_n(20, mean_unique_names_per_department)
ggplot(data = subset(table_by_dpt_analysis, local_department %in%
                       table_by_dpt_analysis_top20$local_department),
       aes(x=local_department, y=mean_unique_names_per_department)) +
  geom_col() + ggtitle("Variety of Names Per Departments Over Time: Mean Number of Unique Names
                       (Top 20)")
```



It is confirmed that the departments where the biggest average/mean numbers of birth have happened over time and we can search for more explanation by comparing with the number of birth. This will let us know how variable a given name is for each departments.