Babynames I

2024-09-05

```r
require(patchwork)
require(httr)
require(glue)
require(ineq)
require(here)
require(skimr)
require(magrittr)
require(tidyverse)

old_theme <- theme_set(theme_minimal())
```

- **L3 MIASHS**
- **Université Paris Cité**
- Année 2024-2025
- Course Homepage

- Moodle

> **!** **Objectives**

Naming babies

French data

The French data are built and made available by INSEE (French Governement Statistics Institute)

- https://www.insee.fr/fr/statistiques/fichier/2540004/nat2021_csv.zip

This dataset has been growing for a while. It has been considered by social scientists for decades. Given names are meant to give insights into a variety of phenomena, including religious observance.

A glimpse at that body of work can be found in *L'archipel français* by Jérome Fourquet, Le Seuil, 2019

Read the File documentation

```r
path_data <- 'DATA'
fname <- 'nat2021_csv.zip'
fpath <- here(path_data, fname)

if (!file.exists(fpath)){
```

```
  url <- "https://www.insee.fr/fr/statistiques/fichier/2540004/nat2021_csv.zip"
  download.file(url, fpath, mode="wb")
}

df_fr <- readr::read_csv2(fpath)

# df_fr |> glimpse()
```

US data

US data may be gathered from

Baby Names USA from 1910 to 2021 (SSA)

See https://www.ssa.gov/oact/babynames/background.html

It can also be obtained by installing and loading the "babynames" package.

Full baby name data provided by the SSA. This includes all names with at least 5 uses.

```
if (!require("babynames")){
  install.packages("babynames")
  stopifnot(require("babynames"), "Couldn't install and load package 'babynames'")
}
```

```
?babynames
```

Tidy the French data

Rename columns according to the next lookup table:

```
lkp <- list(year="annais",
  sex="sexe",
  name="preusuel",
  n="nombre")
```

```
df_fr <- df_fr |>
  rename(!!!lkp) |>                                                    (1)
  mutate(country='fr') |>
  mutate(sex=as_factor(sex)) |>
  mutate(sex=fct_recode(sex, "M"="1", "F"="2")) |>
  mutate(sex=fct_relevel(sex, "F", "M")) |>
  mutate(year=ifelse(year=="XXXX", NA, year)) |>
  mutate(year=as.integer(year))

df_fr  |>
  sample(5) |>
  glimpse()
```

(1) `!!!` (bang-bang-bang) is offered by `rlang` package. Here, we use it to perform *list unpacking* (with the same intent and purposes we use dictionary unpacking in Python)

```
Rows: 686,538
Columns: 5
$ name    <chr> "_PRENOMS_RARES", "_PRENOMS_RARES", "_PRENOMS_RARES", "_PRENOM~
$ n       <dbl> 1249, 1342, 1330, 1286, 1430, 1472, 1451, 1514, 1509, 1526, 16~
$ year    <int> 1900, 1901, 1902, 1903, 1904, 1905, 1906, 1907, 1908, 1909, 19~
```

```
$ sex     <fct> M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M,~
$ country <chr> "fr", "fr", "fr", "fr", "fr", "fr", "fr", "fr", "fr", "fr", "f~
```

Download 'Naissances totales par sexe' from URL `https://www.ined.fr/fichier/s_rubrique/168/t35.fr` from INED.

```r
births_fr_path <- here(path_data, 't35.fr.xls')
births_fr_url <- 'https://www.ined.fr/fichier/s_rubrique/168/t35.fr.xls'

if (!file.exists(births_fr_path)) {
  download.file(births_fr_url, births_fr_path)
}
```

```r
births_fr <-  readxl::read_excel(births_fr_path, skip = 3)

births_fr <- births_fr[-1, ]


births_fr |>
  glimpse()
```

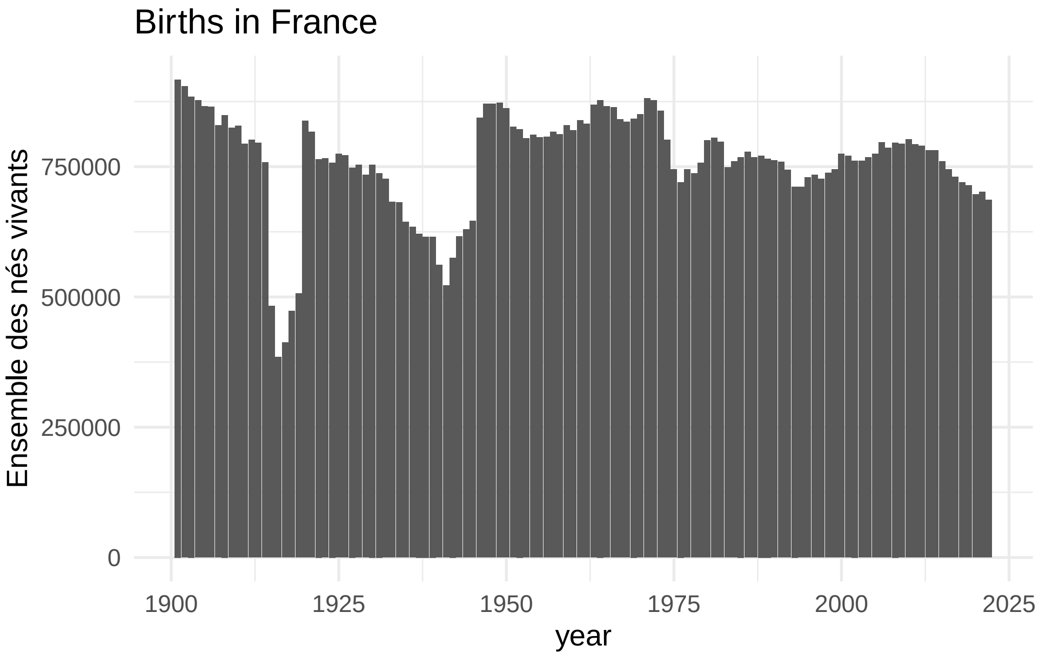
```
Rows: 130
Columns: 10
$ `Répartition par sexe et vie`                <chr> "1901", "1902", "1903", "~
$ `Ensemble des nés vivants`                   <dbl> 917075, 904434, 884498, 8~
$ `Nés vivants - Garçons`                      <dbl> 468125, 462097, 451510, 4~
$ `Nés vivants - Filles`                       <dbl> 448950, 442337, 432988, 4~
$ `Ensemble des enfants sans vie`              <dbl> 32410, 32000, 31076, 3067~
$ `Enfants sans vie - Garçons`                 <chr> "18522", "18172", "17875"~
$ `Enfants sans vie - Filles`                  <chr> "13888", "13828", "13201"~
$ `Garçons vivants pour 100 nés\nvivants`      <dbl> 51.0, 51.1, 51.0, 51.0, 5~
$ `Garçons vivants pour 100\nfilles vivantes`  <dbl> 104.3, 104.5, 104.3, 104.~
$ `Garçons sans vie pour 100\nfilles sans vie` <chr> "133.40000000000001", "13~
```

If you have problems with the excel reader, feel free to download an equivalent `csv` file from url

```r
names(births_fr)[1] <- "year"
```

```r
births_fr <- births_fr |>
  mutate(year=as.integer(year)) |>
  drop_na()
```

```r
births_fr |>
  ggplot() +
  aes(x=year, y=`Ensemble des nés vivants`) +
  geom_col() +
  labs(title="Births in France")
```

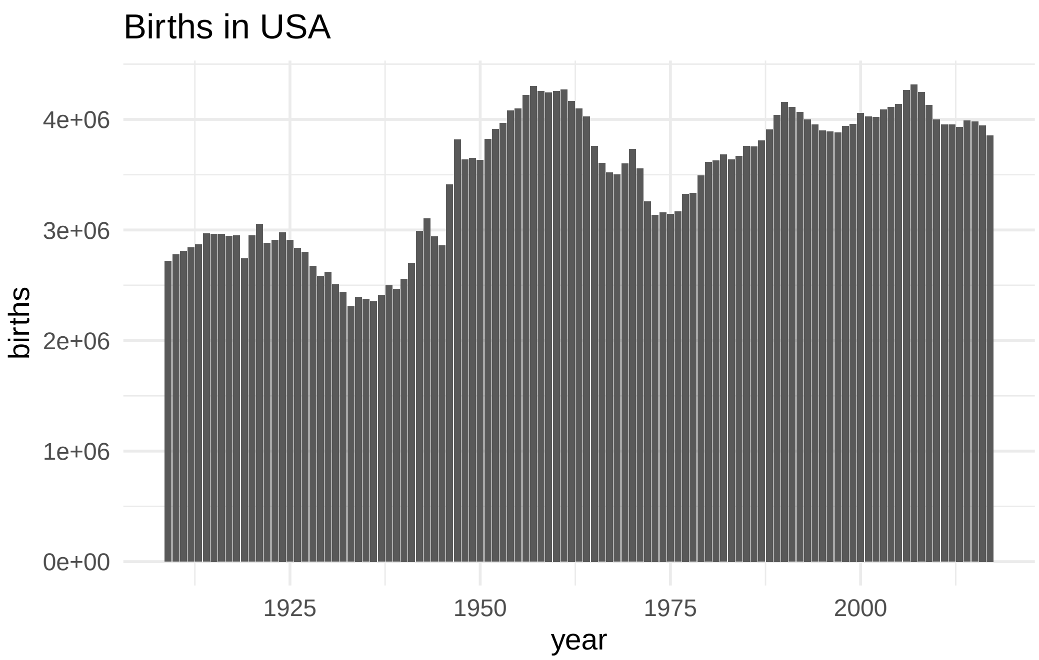


Tidy the American data

```
babynames <- babynames |>
  mutate(country='us') |>
  mutate(sex=as_factor(sex))

babynames |>
  glimpse()
```

```
Rows: 1,924,665
Columns: 6
$ year    <dbl> 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 18~
$ sex     <fct> F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F, F,~
$ name    <chr> "Mary", "Anna", "Emma", "Elizabeth", "Minnie", "Margaret", "Id~
$ n       <int> 7065, 2604, 2003, 1939, 1746, 1578, 1472, 1414, 1320, 1288, 12~
$ prop    <dbl> 0.07238359, 0.02667896, 0.02052149, 0.01986579, 0.01788843, 0.~
$ country <chr> "us", "us", "us", "us", "us", "us", "us", "us", "us", "us", "u~
```

```
births_us <- births

births_us  |>
  ggplot() +
  aes(x=year, y=births) +
  geom_col() +
  labs(title="Births in USA")
```



Sex ratios

> **Question**
>
> In dataset **df_fr** compute the total number of reported male and female births per year. Compute and plot the sex ratio.

```
df_accounted_births_fr <- df_fr |>
  group_by(year, sex) |>
  summarise(n=sum(n))

df_accounted_births_fr |>
  glimpse()
```

```
Rows: 246
Columns: 3
Groups: year [123]
$ year <int> 1900, 1900, 1901, 1901, 1902, 1902, 1903, 1903, 1904, 1904, 1905,~
$ sex  <fct> F, M, F, M, F, M, F, M, F, M, F, M, F, M, F, M, F, M, F, M, F, M,~
$ n    <dbl> 237653, 177387, 257492, 195964, 261437, 204354, 261450, 207360, 2~
```

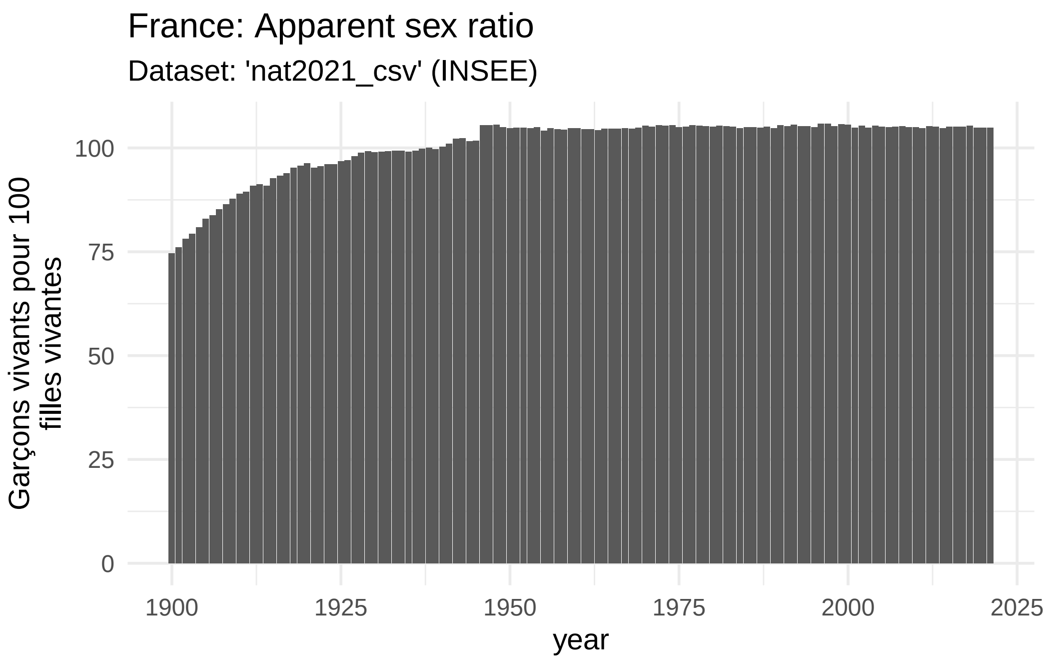
```
df_app_sex_ratio_fr <- df_accounted_births_fr |>
  pivot_wider(id_cols=year,
              names_from=sex,
              values_from=`n`) |>
  mutate(`Garçons vivants pour 100\nfilles vivantes`=100*M/F)

df_app_sex_ratio_fr |>
  glimpse()
```

```
Rows: 123
Columns: 4
Groups: year [123]
$ year                                       <int> 1900, 1901, 1902, 1903, 19~
$ F                                          <dbl> 237653, 257492, 261437, 26~
```

```
$ M                                          <dbl> 177387, 195964, 204354, 20~
$ `Garçons vivants pour 100\nfilles vivantes` <dbl> 74.64118, 76.10489, 78.165~
```

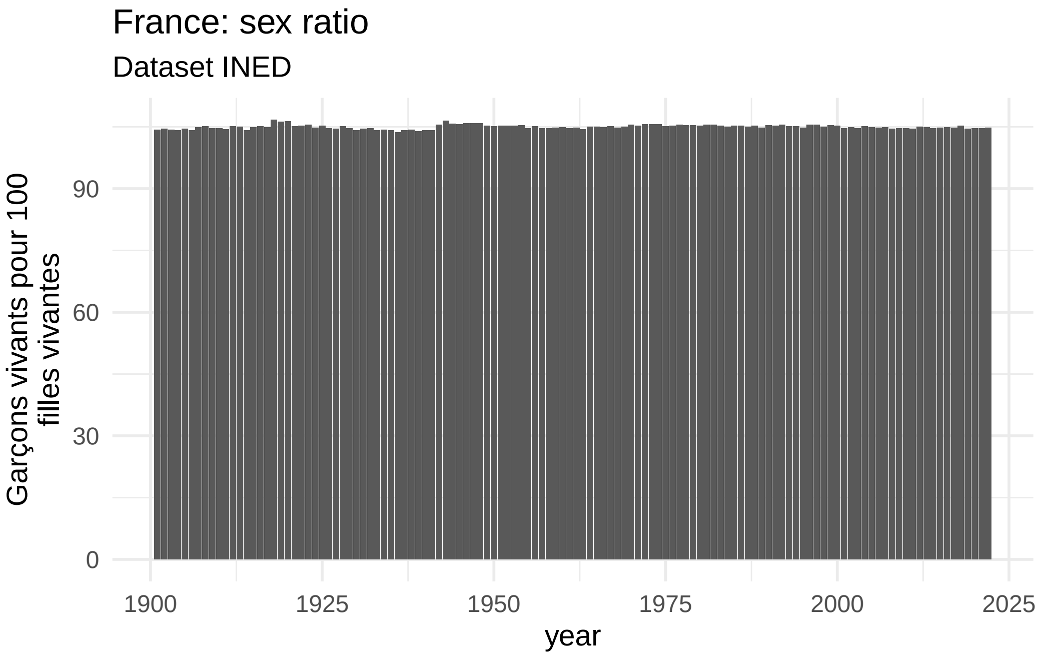
```
p_app_sex_ratio_fr <- df_app_sex_ratio_fr |>
  ggplot() +
  aes(x=year, y=`Garçons vivants pour 100\nfilles vivantes`) +
  geom_col() +
  theme_minimal()                                                          (1)


p_app_sex_ratio_fr  +
  labs(
    title="France: Apparent sex ratio",
    subtitle="Dataset: 'nat2021_csv' (INSEE)"
  )
```

(1) Should not be necessary

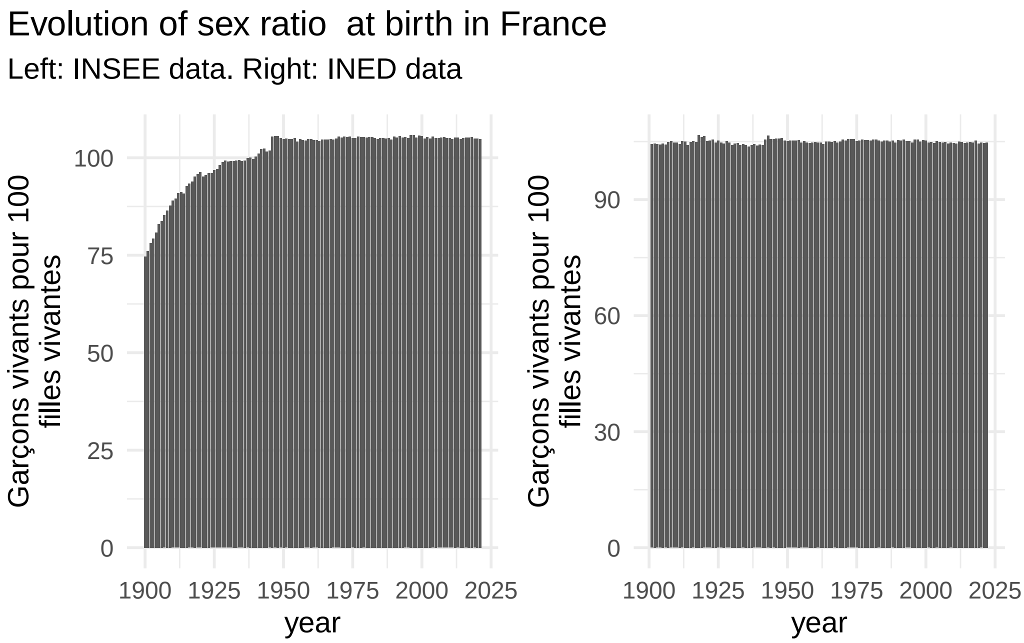


> **Question**
>
> Compare with sex ratio as given in dataset from INED

```
p_sex_ratio_fr <- p_app_sex_ratio_fr %+%
  births_fr

p_sex_ratio_fr + labs(
    title="France: sex ratio",
    subtitle="Dataset INED")
```




```
(p_app_sex_ratio_fr + p_sex_ratio_fr) +
  plot_annotation(
    title="Evolution of sex ratio  at birth in France",
    subtitle="Left: INSEE data. Right: INED data"
  )
```

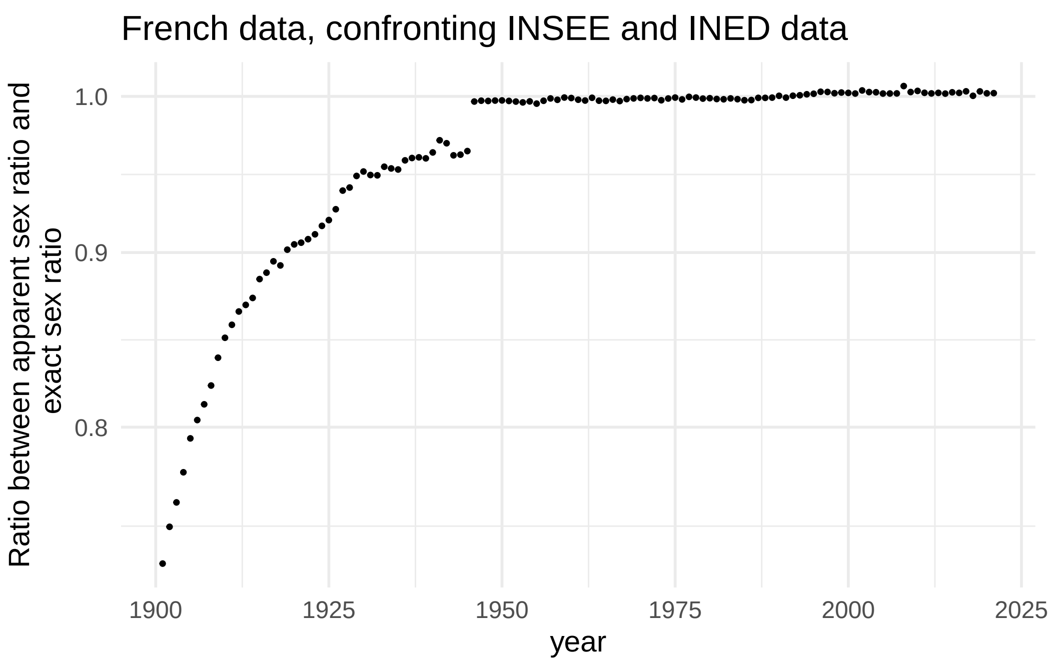



```
df_app_sex_ratio_fr |>
  inner_join(births_fr, by="year") |>
  glimpse()
```

```
Rows: 121
Columns: 13
Groups: year [121]
$ year                                         <int> 1901, 1902, 1903, 1904, ~
$ F                                            <dbl> 257492, 261437, 261450, ~
$ M                                            <dbl> 195964, 204354, 207360, ~
$ `Garçons vivants pour 100\nfilles vivantes.x` <dbl> 76.10489, 78.16568, 79.3~
```

```
$ `Ensemble des nés vivants`                       <dbl> 917075, 904434, 884498, ~
$ `Nés vivants - Garçons`                          <dbl> 468125, 462097, 451510, ~
$ `Nés vivants - Filles`                           <dbl> 448950, 442337, 432988, ~
$ `Ensemble des enfants sans vie`                  <dbl> 32410, 32000, 31076, 306~
$ `Enfants sans vie - Garçons`                     <chr> "18522", "18172", "17875~
$ `Enfants sans vie - Filles`                      <chr> "13888", "13828", "13201~
$ `Garçons vivants pour 100 nés\nvivants`          <dbl> 51.0, 51.1, 51.0, 51.0, ~
$ `Garçons vivants pour 100\nfilles vivantes.y`    <dbl> 104.3, 104.5, 104.3, 104~
$ `Garçons sans vie pour 100\nfilles sans vie`     <chr> "133.40000000000001", "1~
```

```
df_app_sex_ratio_fr |>
  inner_join(births_fr, by="year") |>
  ggplot() +
  aes(x=year, y=`Garçons vivants pour 100\nfilles vivantes.x`/`Garçons vivants pour 100\nf
  geom_point(size=.5) +
  scale_y_log10() +
  ylab('Ratio between apparent sex ratio and\n exact sex ratio') +
  labs(
    title="French data, confronting INSEE and INED data"
  )
```



> **Question**
>
> Consider the fluctuations of the sex ratio through the years.
> Are they consistent with the hypothesis: the sex of newborns are independently. identically distributed with the probability of getting a girl equal to .48?

> **Question**
>
> Consider again the fluctuations of the sex ratio through the years.
> Assume that for each year the sex of newborns are independently. identically distributed with the probability of getting a girl depending on the year.
> Are the data consistent with the hypothesis: the probability of getting a girl remains constant thoughout the years?

Picturing concentration of babynames distributions

Every year, in each country, for both sex, the name counts define a discrete probability distribution over the set of names (the universe).

This distribution, just as an income or wealth distribution, is (usually) far from being uniform. We want to assess how uneven it is.

We use the tools developed in econometrics.

Without loss of generality, we assume that we handle a distribution over positive integers $1, \ldots, n$ where n is the number of distinct names given during a year.

We assume that frequencies $p_1, p_2, \ldots, p_n$ are given in ascending order, ties are broken arbitrarily.

The `Lorenz function` (Lorenz not `Lorentz`) maps $[0,1] \to [0,1]$.

$$L(x) = \sum_{i=1}^{\lfloor nx \rfloor} p_i .$$

Note that this is a piecewise constant function.

> **Question**
>
> Compute and plot the Lorenz function for a given `sex`, `year` and `country`

```r
make_lorenz_df <- function(df) {
  df |>
  group_by(year, sex) |>
  arrange(n) |>
  mutate(rr=row_number()/n(), L=cumsum(n)/sum(n),  p=n/sum(n)) |>   # (1)
  ungroup()
}
```

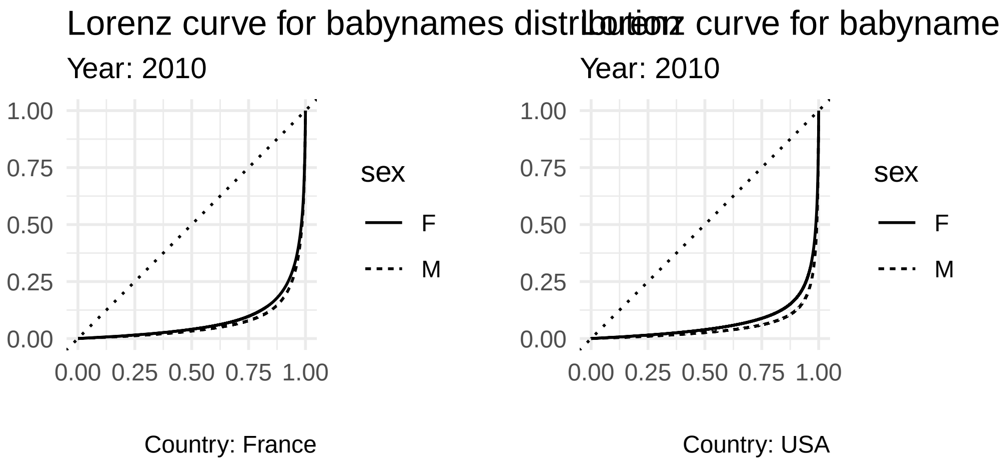
(1) The three expressions defining `rr`, `L` and `p` act as window functions. The window is defined by partitioning by `year, sex` and ordering by `n`. In SQL parlance: `WINDOW w AS (PARTITION BY year, sex ORDER BY n)`

```r
df_lorenz_fr <- df_fr |>
  filter(name != '_PRENOMS_RARES' &  !is.na(year)) |>
  make_lorenz_df()

df_lorenz_us <- babynames |>
  make_lorenz_df()
```

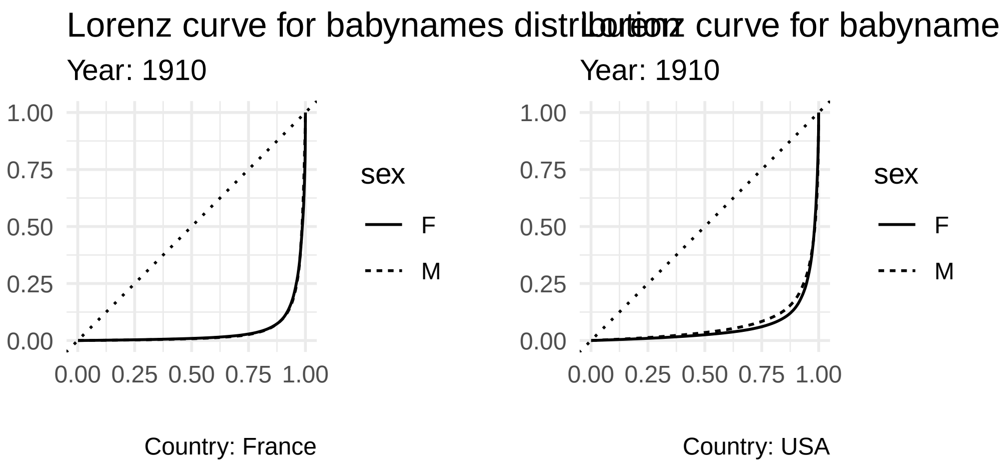
```r
plot_lorenz <- function(df, ze_year=2020, ze_country='fr'){
  df |>
  filter(year==ze_year) |>
  ggplot() +
    aes(x=rr, y=L, linetype=sex) +
    geom_line()  +
    coord_fixed() +
    xlab("") +
    ylab("") +
    geom_abline(intercept=0, slope=1, linetype="dotted") +
```

```r
    labs(title="Lorenz curve for babynames distribution",
         subtitle=glue("Year: {ze_year}"),
         caption=glue("Country: {ze_country}")
    )
}
```

```r
plot_lorenz(df_lorenz_fr, 2010, 'France') |
plot_lorenz(df_lorenz_us, 2010, 'USA')
```




```r
plot_lorenz(df_lorenz_fr, 1910, 'France') |
plot_lorenz(df_lorenz_us, 1910, 'USA')
```

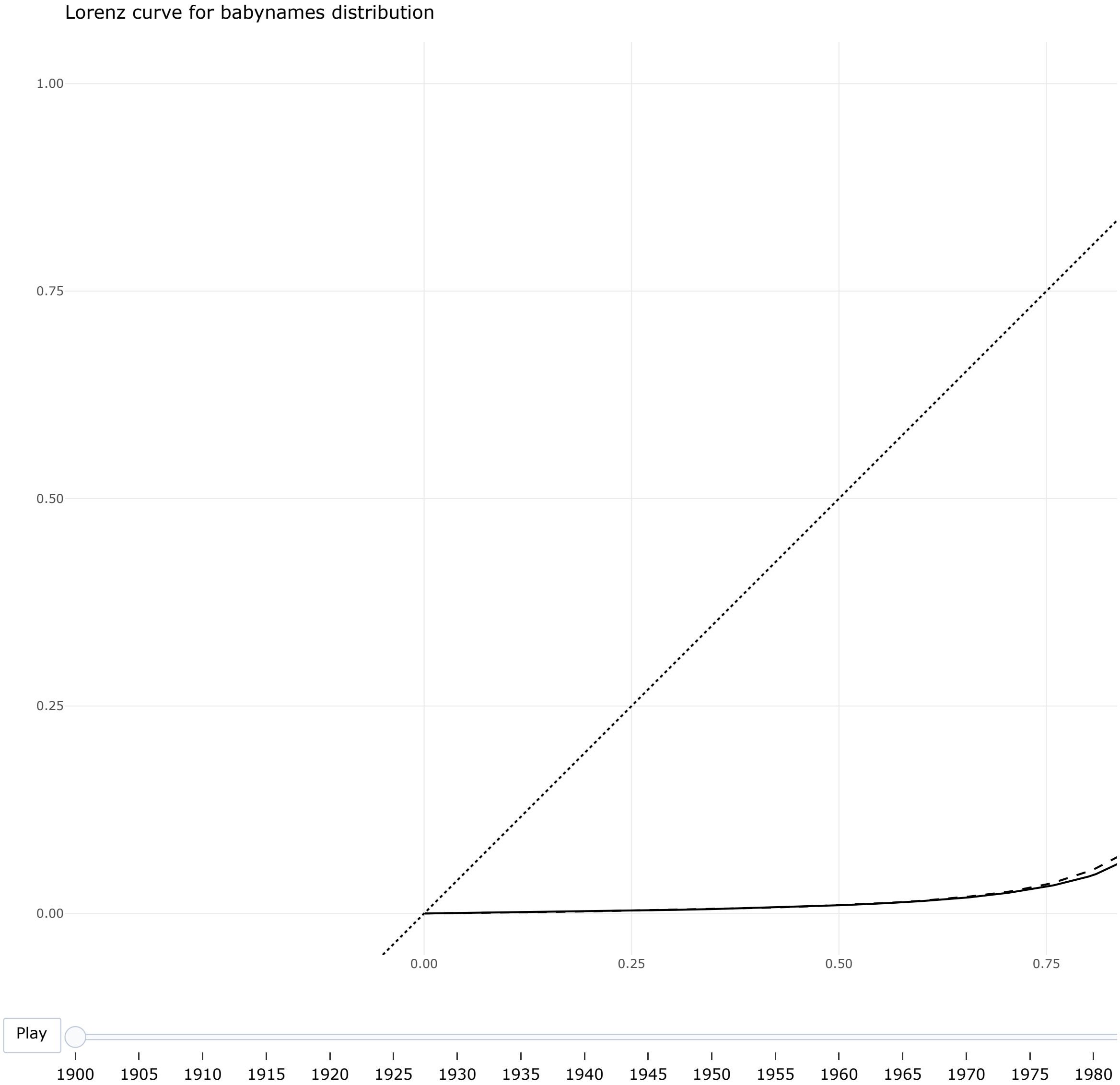


Question

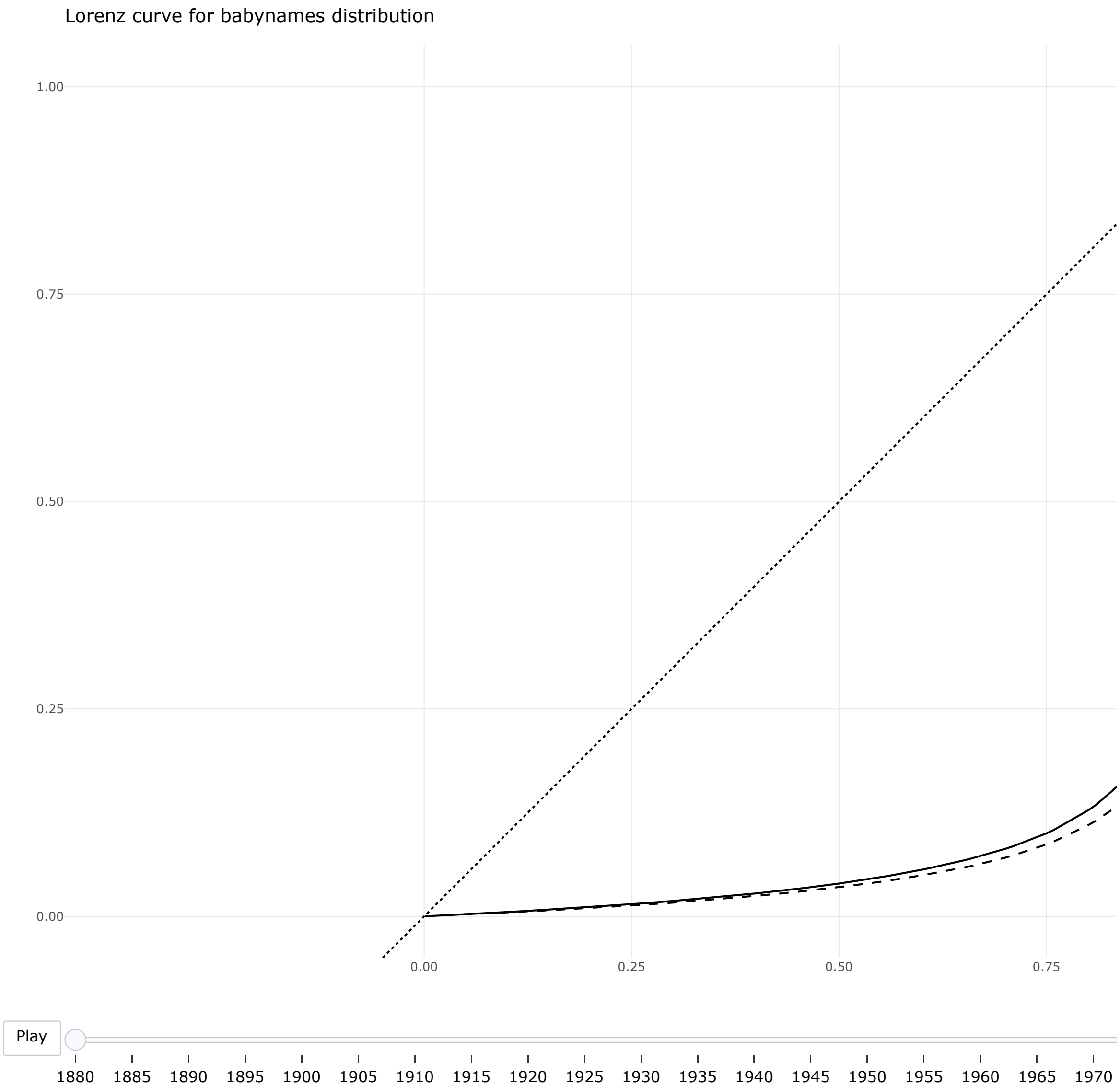
Design an animated plot that shows the evolution of the Lorenz curve of babynames distribution through the years for a given sex and country.

```r
p_inter <- filter(df_lorenz_fr,
                  year %% 5 ==0,
                  floor(rr*100)%% 5==0) |>
  ggplot() +
    aes(x=rr, y=L, linetype=sex, frame=year) +
    geom_line()  +
    coord_fixed() +
    xlab("") +
    ylab("") +
    geom_abline(intercept=0, slope=1, linetype="dotted")
```

```
(p_inter +
    labs(title="Lorenz curve for babynames distribution",
         caption=glue("Country: France")
    )) |> plotly::ggplotly()
```




```
(
  p_inter %+%
    filter(df_lorenz_us,
           year %% 5 ==0,
           floor(rr*100)%% 5==0)   +
    labs(title="Lorenz curve for babynames distribution",
         caption=glue("Country: US"))
) |> plotly::ggplotly()
```



Inequality indices

The Lorenz curve summarizes how far a discrete probability distribution is from the uniform distribution. This is a very rich summary and it is difficult to communicate this message to a wide audience. People tend to favor numerical indices (they don't really understand, but they get used to it): Gini, Atkinson, Theil, …

The Gini index is twice the surface of the area comprised between curves $y = x$ and $y = L(x)$.

$$G = 2 \times \int_0^1 (x - L(x))\mathrm{d}x$$

The next formula allows us to compute it efficiently.

$$G = \frac{2\sum_{i=1}^n i p_i}{n\sum_{i=1}^n p_i} - \frac{n+1}{n}.$$

Question

Compute and plot Gini index of names distribution over time for sex and countries

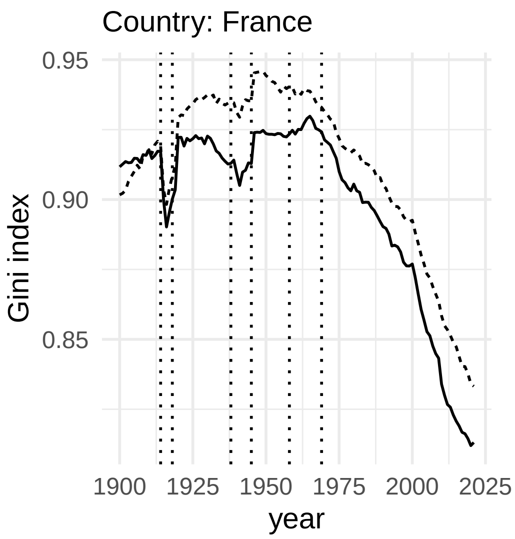
```
p_gini <- df_lorenz_fr |>
  group_by(year, sex) |>
  summarize(gini=2 * sum(rr*p) - 1 - 1/n()) |>
  ggplot() +
  aes(x=year, y=gini, linetype=sex) +
  geom_line() +
  theme(legend.position="none") +
  ylab("Gini index")

for(y in c(1914, 1918, 1938, 1945, 1958, 1969)) {
  p_gini <- p_gini +
    geom_vline(xintercept = y, linetype="dotted")
}

p_gini_fr <- p_gini +
  labs(subtitle="Country: France")
```

```
p_gini_us <- (
  p_gini %+%
    (df_lorenz_us |>
     group_by(year, sex) |>
     summarize(gini=2 * sum(rr*p) - 1 - 1/n(), .groups="drop")) +
     labs(
      subtitle="Country: US"
  )
)
```

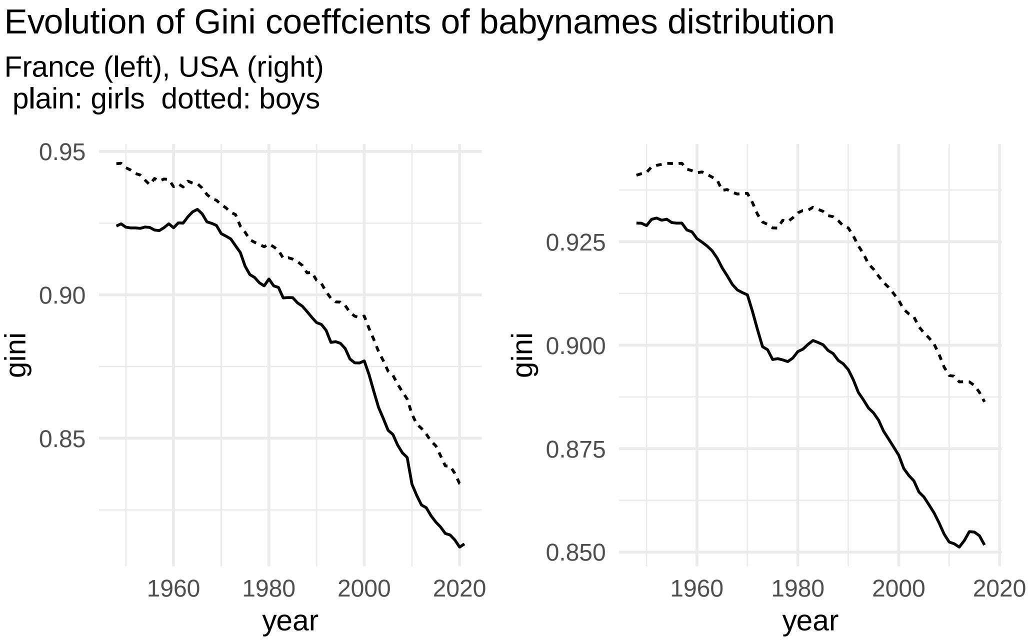
```
(p_gini_fr| p_gini_us) +
  plot_annotation(
    title="Gini index of names distributions",
    subtitle="..."
)
```

Gini index of names distributions

...




```r
giniplot <- function (df) {
  df  |>
  filter(name != '_PRENOMS_RARES' &  !is.na(year)) |>
  group_by(year, sex)  |>
  mutate(gini=ineq::ineq(n))  |>
  ggplot() +
  aes(x=year, y=gini, linetype=sex) +
  geom_line() +
  theme(legend.position = "none")
}

p1 <- giniplot(filter(df_fr, year> 1947))
p2 <- giniplot(filter(babynames, year>1947))


( p1 | p2 ) +
  plot_annotation(
    title = "Evolution of Gini coeffcients of babynames distribution",
    subtitle="France (left), USA (right) \n plain: girls  dotted: boys"
  )
```

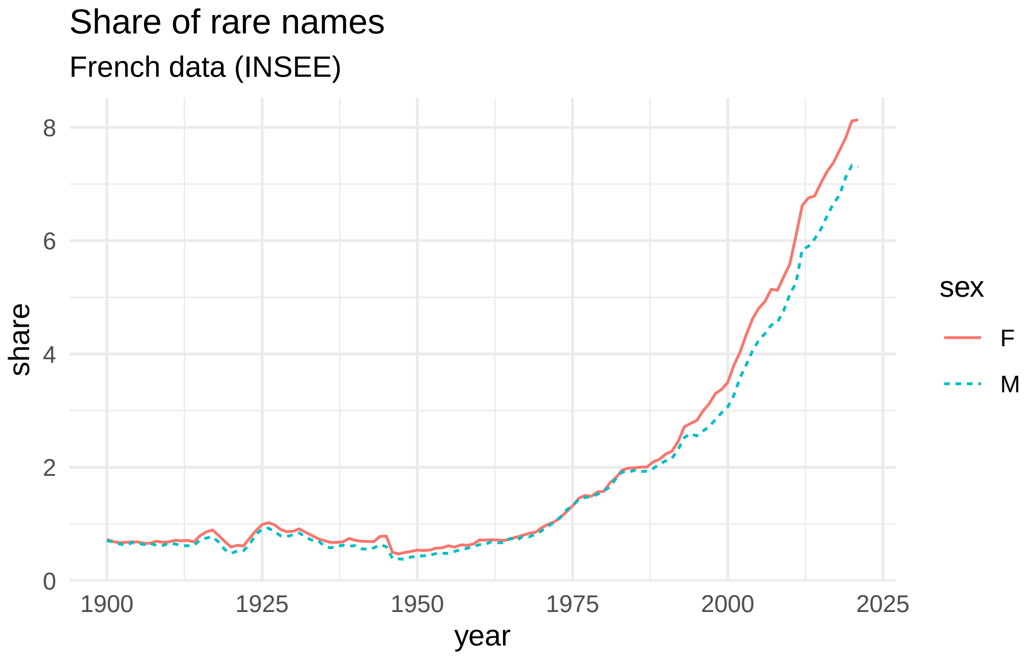


PRENOMS RARES in France

> **Question**
>
> For each sex, Plot the proportion of births given `_PRENOMS_RARES` as a function of year.

```
df_fr |>
  filter(!is.na(year)) |>
  group_by(year, sex) |>
  mutate(total=sum(n)) |>                                      (1)
  filter(name=='_PRENOMS_RARES') |>
  select(-name) |>
  mutate(share= 100*n/total) |>
  ungroup() |>
  ggplot() +
    aes(x=year, y=share, color=sex, linetype=sex) +
    geom_line() +
    labs(
      title="Share of rare names",
      subtitle="French data (INSEE)"
    ) +
  theme_minimal()                                              (2)
```

(1) Here `sum()` works as a window function over partition by `year, sex`.
(2) This should not be necessary. Inconsistency in quarto ?



Look for `Mary` in US Data

Marie, Jeanne and France in France

Question

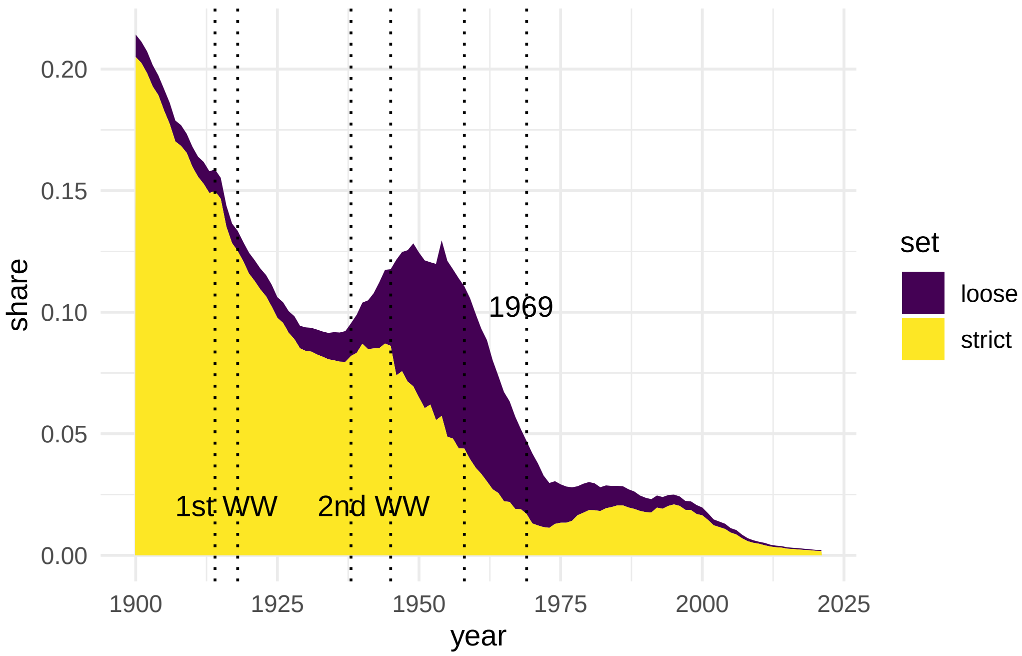
Plot the proportion of female births given name 'MARIE' or 'MARIE-...' as a function of year. Proceed in such a way that the reader can see the share of compounded names. We are expecting an *area plot*

Have a look at r-graph-gallery: stacked area and at ggplot documentation. Pay attention on the way you stack the area corresponding to names matching pattern 'MARIE-...' over or under the are corresponding to babies named 'MARIE'

```r
theme_set(theme_minimal())

share_name  <- function(data, .name_stem='MARIE', .sex='F'){
  data %>%
  dplyr::filter(sex==.sex, !is.na(year)) %>%
  select(-sex) %>%
  group_by(year) %>%
  summarize(strict=sum(ifelse(name==.name_stem, n, 0)),
            loose=sum(ifelse(stringr::str_starts(name, glue('{.name_stem}-')), n, 0)),
            total=sum(n)
  ) %>%
  transmute(year=year,
            strict=strict/total,
            loose=loose/total) %>%
  pivot_longer(strict:loose,
               names_to=c("set"),
               values_to="share") %>%
```

```
  mutate(set=factor(set,
                    levels=c("loose", "strict"),
                    ordered=TRUE))
}
```

```
decline_and_fall <- function(df, .name_stem = "MARIE", .sex = "F"){

  df <- share_name(df, .name_stem, .sex)
  maxshare <- max(pull(df, share), na.rm = T)

  p <- df |>
    ggplot(aes(x=year)) +
    geom_area(aes(y=share,
                  fill=set),
              position="stack") +
    ylab("share") +
    annotate('text',
            label="1st WW",
            x = 1916,
            y=0.1*maxshare) +
    annotate('text',
             label="2nd WW",
             x = 1942,
             y=0.1*maxshare) +
    annotate("text",
             label= "1969",
             x= 1968,
             y= 0.5*maxshare) +
    theme_minimal()

  for(y in c(1914, 1918, 1938, 1945, 1958, 1969)) {
    p <- p + geom_vline(xintercept = y, linetype="dotted")
  }
  p
}

decline_and_fall(df_fr, .name_stem = "MARIE", .sex="F")
```



See Graphique 3, page 48, de *L'archipel français* de J. Fourquet. Le Seuil. Essais. Vol. 898.

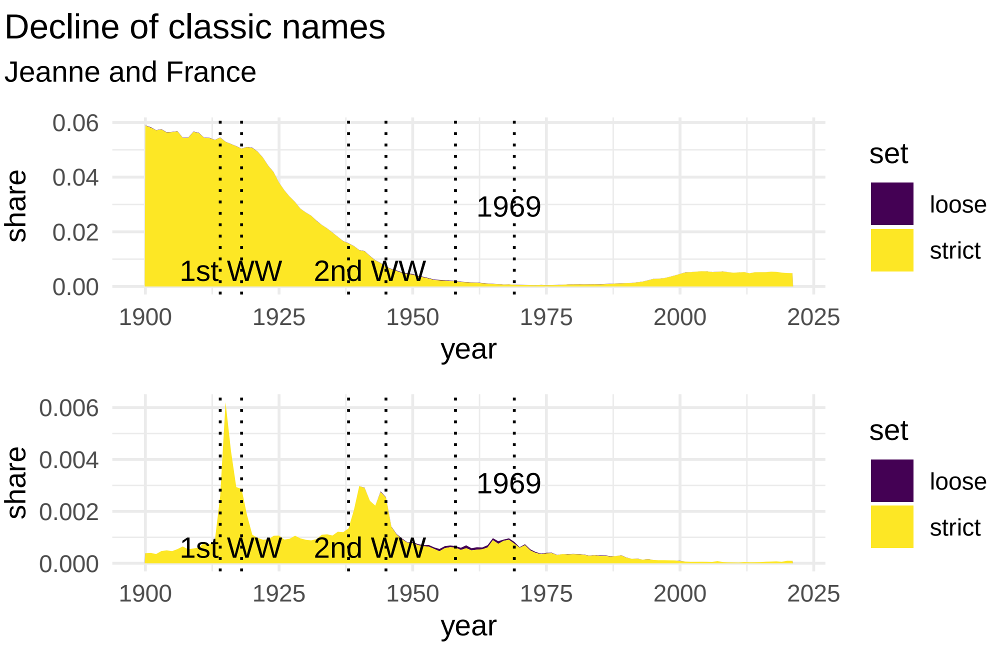
Question

Answer the same question for JEANNE and FRANCE

```
p_jeanne <- decline_and_fall(df_fr, "JEANNE")
# p_jeanne
```

```
p_france <- decline_and_fall(df_fr, "FRANCE")

# p_france
```

```
patchw <- p_jeanne / p_france

patchw + plot_annotation(
  title="Decline of classic names",
  subtitle="Jeanne and France"
)
```

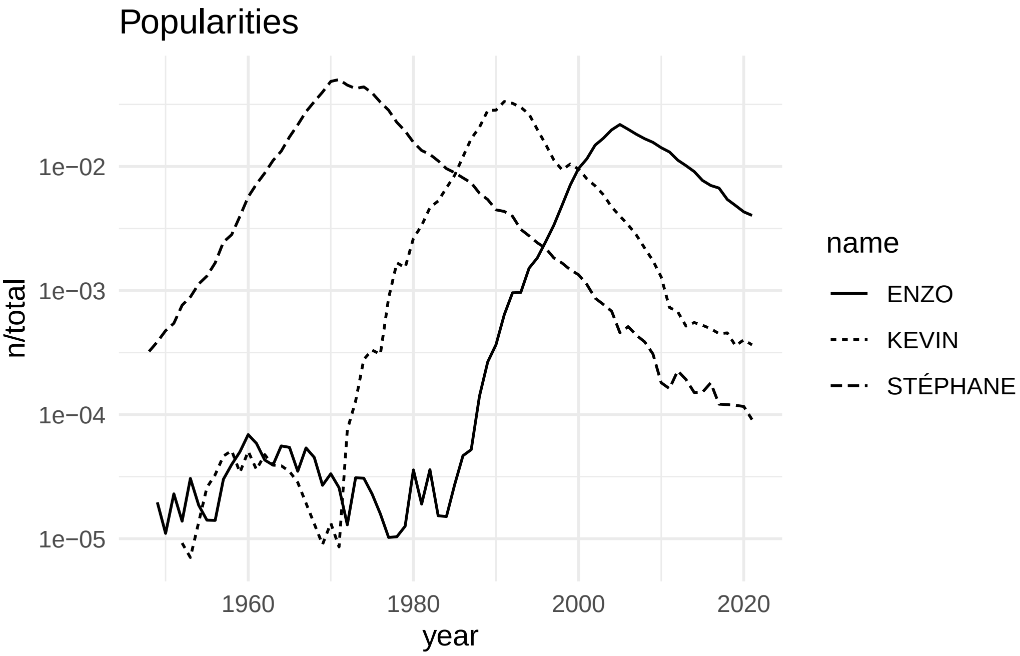


Patterns of popularity

> **Question**
>
> Plot the popularities of `KEVIN`, `ENZO`, `STÉPHANE` as a function of `year`.

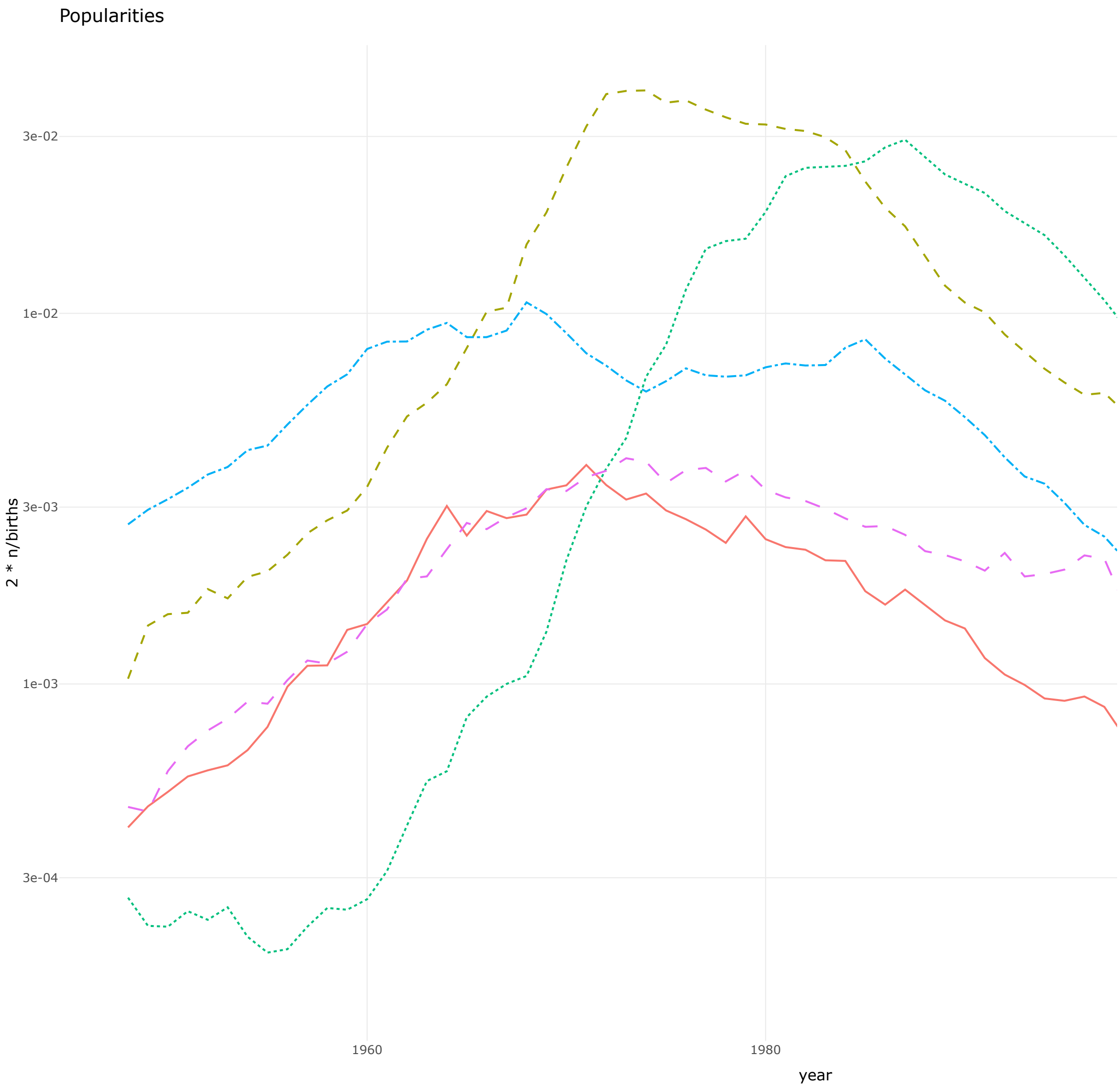
```
df_accounted_births_fr <- rename(df_accounted_births_fr, total=n)
```

```
prenoms <- c("STÉPHANE", "KEVIN", "ENZO")

df_fr |>
  filter(year>1947) |>
  filter(name %in% prenoms, sex=="M") %>%
  inner_join(df_accounted_births_fr, by=c("year", "sex")) %>%
  ggplot() +
  aes(x=year, y=n/total, linetype=name) +
  geom_line() +
  scale_y_log10() +
  ggtitle(glue("Popularities"))
```



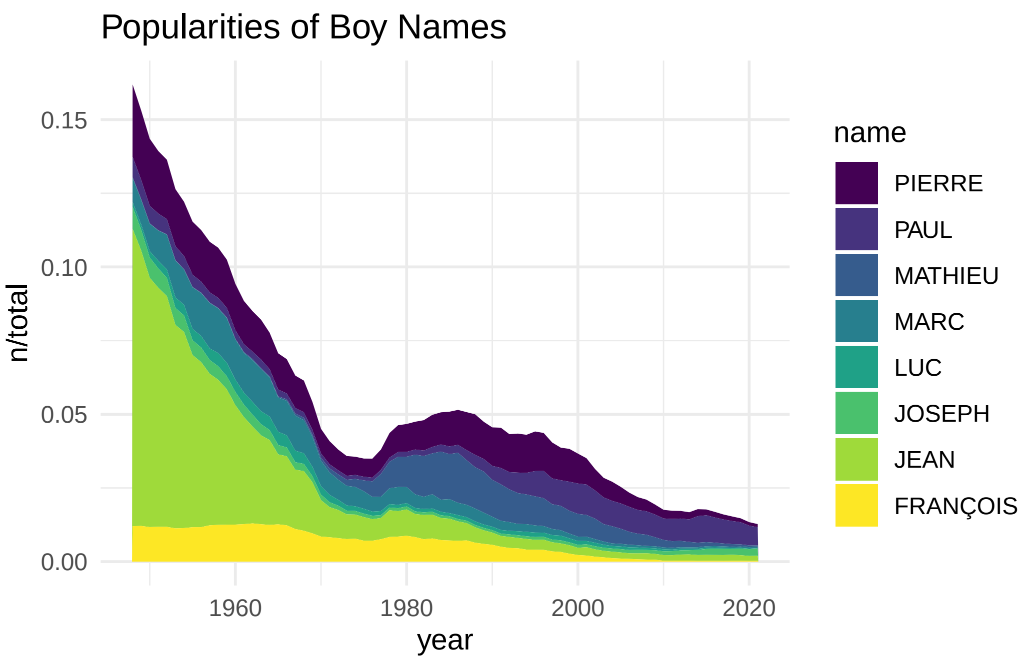
We can investigate surges of popularity for female English names in the way we did for male French names.

```
hypenames <- c('Jessica', 'Jennifer', 'Dana', 'Monica', 'Laura')

(
  babynames %>%
    filter(year > 1947) |>
    filter(name %in% hypenames, sex=='F') %>%
    inner_join(babynames::births, by=c("year")) %>%
    ggplot() +
      aes(x=year, y=2*n/births, linetype=name, colour=name) +
      geom_line() +
      scale_y_log10() +
      ggtitle(glue("Popularities"))
) |>
    plotly::ggplotly()
```



> **Question**
>
> Plot the popularities of "JEAN", "LUC", "MATHIEU", "MARC", "PAUL", "PIERRE", "JOSEPH", "FRANÇOIS" as a function of `year`. Use stacked area style plot.

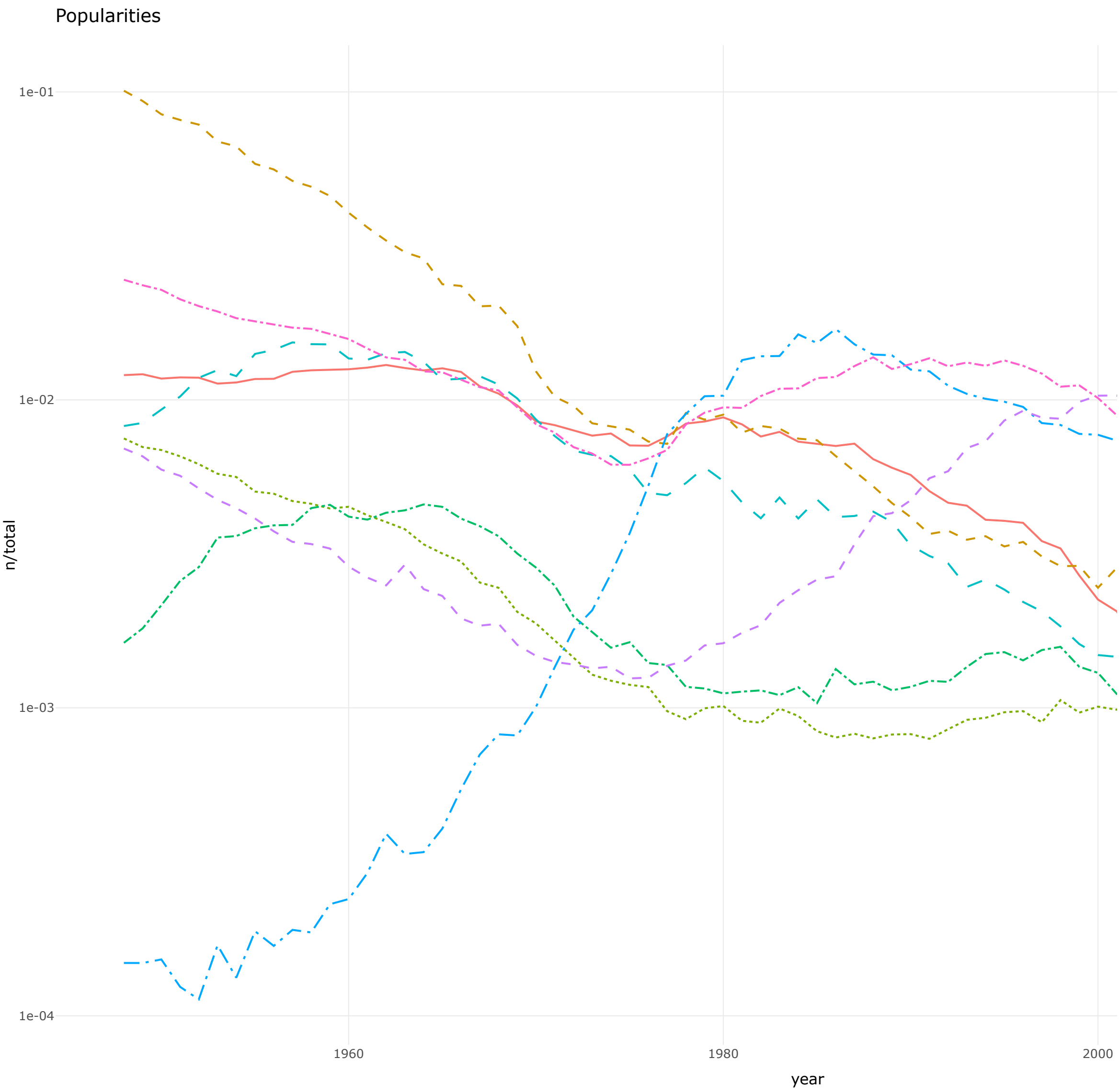
```
prenoms <- c("JEAN", "LUC", "MATHIEU", "MARC", "PAUL", "PIERRE", "JOSEPH", "FRANÇOIS")

df_fr %>%
  filter(year>1947) |>
  filter(name %in% prenoms, sex=="M") %>%
  mutate(name= as_factor(name)) %>%
  mutate(name= fct_rev(name)) %>%
  inner_join(df_accounted_births_fr, by=c("year", "sex")) %>%
  ggplot() +
  aes(x=year, y=n/total, linetype=name, fill=name) +
  scale_fill_viridis_d() +
  geom_area(position = "stack") +
#  scale_y_log10() +
  ggtitle(glue("Popularities of Boy Names"))
```



Question

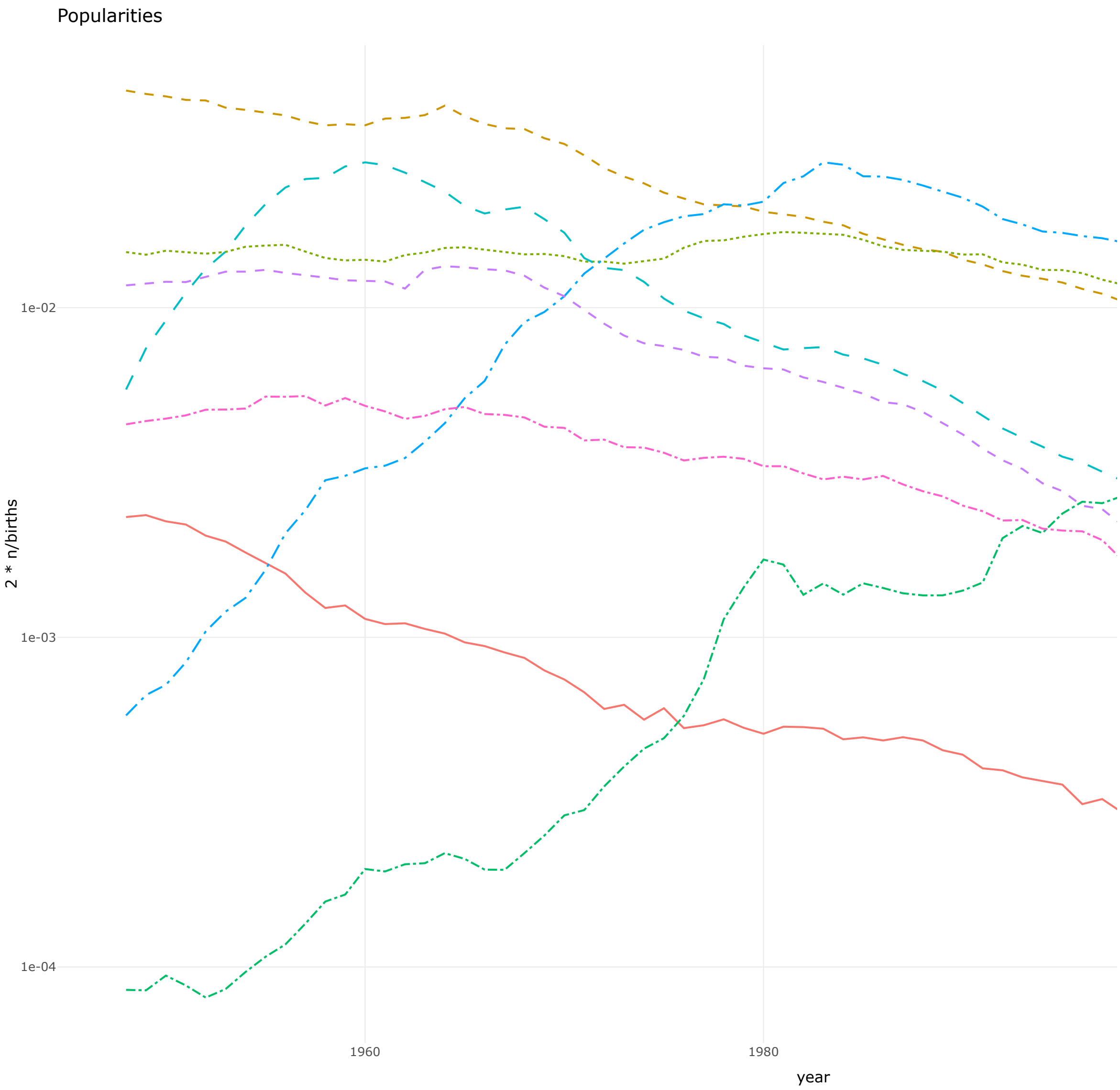
Plot the popularities of "JEAN", "LUC", "MATHIEU", "MARC", "PAUL", "PIERRE", "JOSEPH", "FRANÇOIS" as a function of `year`. Use line plot.

```
q <- (
  df_fr %>%
    filter(year > 1947) |>
    filter(name %in% prenoms, sex=="M") %>%
    inner_join(df_accounted_births_fr, by=c("year", "sex")) %>%
    ggplot() +
      aes(x=year, y=n/total, linetype=name, colour=name) +
      geom_line() +
      scale_y_log10() +
      ggtitle(glue("Popularities"))
) |>
    plotly::ggplotly()

q
```



> **Question**
>
> Look for the translation of these names in US Data

```
firstnames <- str_to_title(c("JOHN", "LUKE", "MATTHEW", "MARK", "PAUL", "PETER", "JOSEPH",

(babynames %>%
  filter(year > 1947) |>
  filter(name %in% firstnames, sex=='M') %>%
  inner_join(babynames::births, by=c("year")) %>%
  ggplot() +
  aes(x=year, y=2*n/births, linetype=name, colour=name) +
  geom_line() +
  scale_y_log10() +
  ggtitle(glue("Popularities"))) |>
    plotly::ggplotly()
```



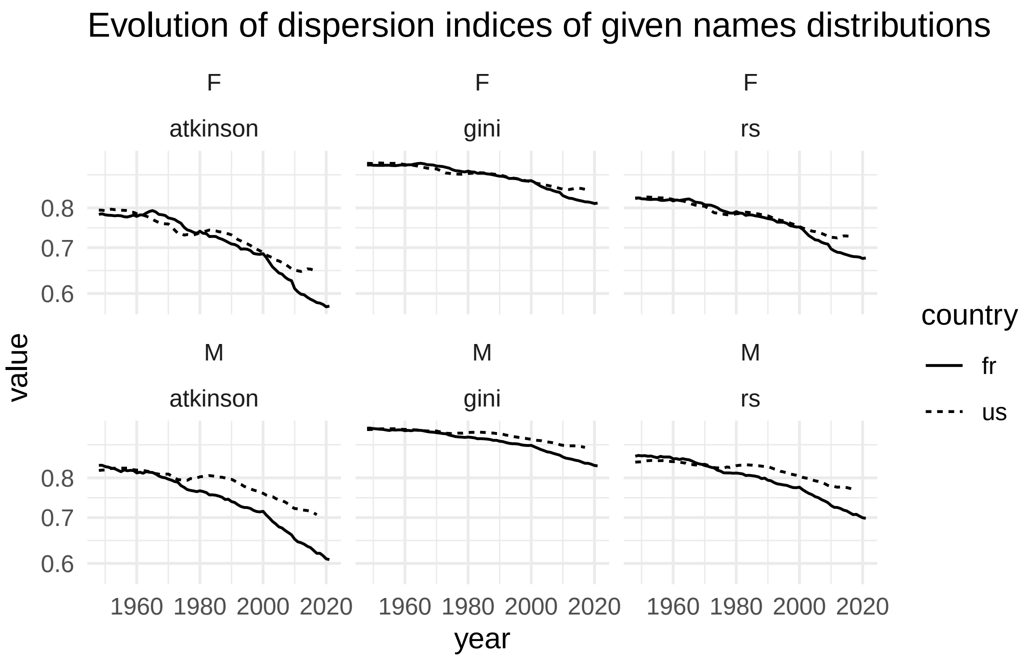
The variations of popularity exhibit different patterns

- Some names declined steadily after second world war.
- Other names started from a very low popularity and enjoyed a rapid increase in popularity over one or two decades. Afterwards, these names rapidly lost the public favor and returned to obscurity.

Grouping names by patterns of popularity

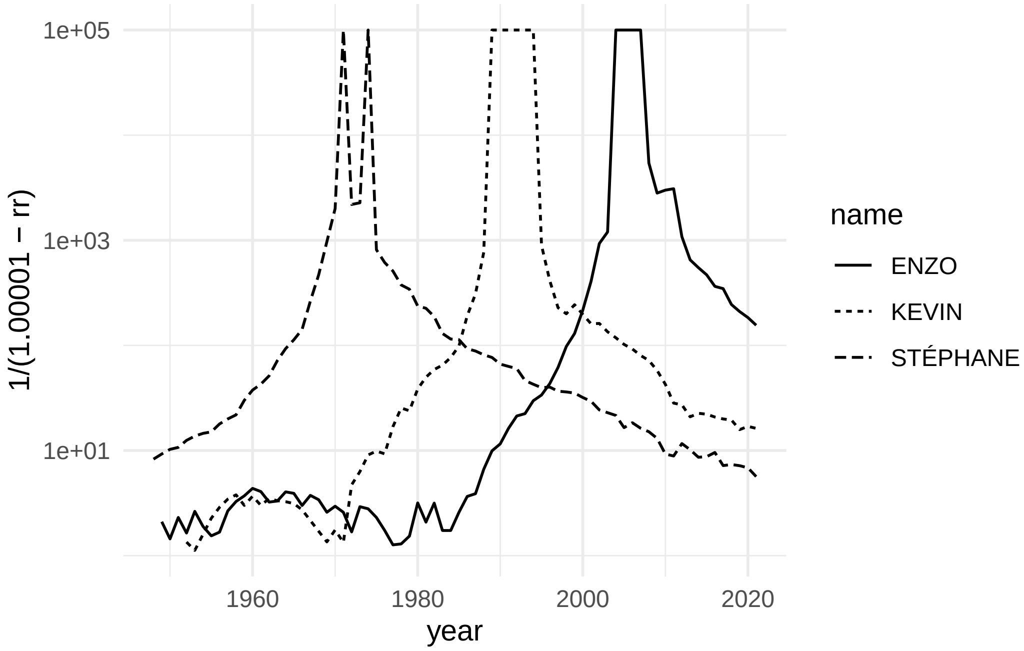
```
bind_rows(df_lorenz_fr, df_lorenz_us) |>
  filter(year> 1947, name != '_PRENOMS_RARES') |>
  group_by(country, year, sex) |>
  summarise(shannon=sum(p*log2(p)),
            gini=ineq(p, na.rm = T),
            atkinson=ineq(p, type="Atkinson", na.rm = T),
            theil=ineq(p, type="Theil", na.rm = T),
            entropy= ineq(p, type="entropy", na.rm = T),
            rs=ineq(p, type="RS", na.rm = T),
            .groups = "drop") |>
  pivot_longer(cols=-c(country,year,sex),
               names_to = "index",
```

```
                values_to = "value") |>
  filter(! index %in% c('entropy','theil', 'shannon')) |>
  ggplot() +
  aes(x=year, y=value, linetype=country) +
  geom_line() +
  scale_y_log10() +
  facet_wrap(~ sex + index) +
  ggtitle("Evolution of dispersion indices of given names distributions")
```

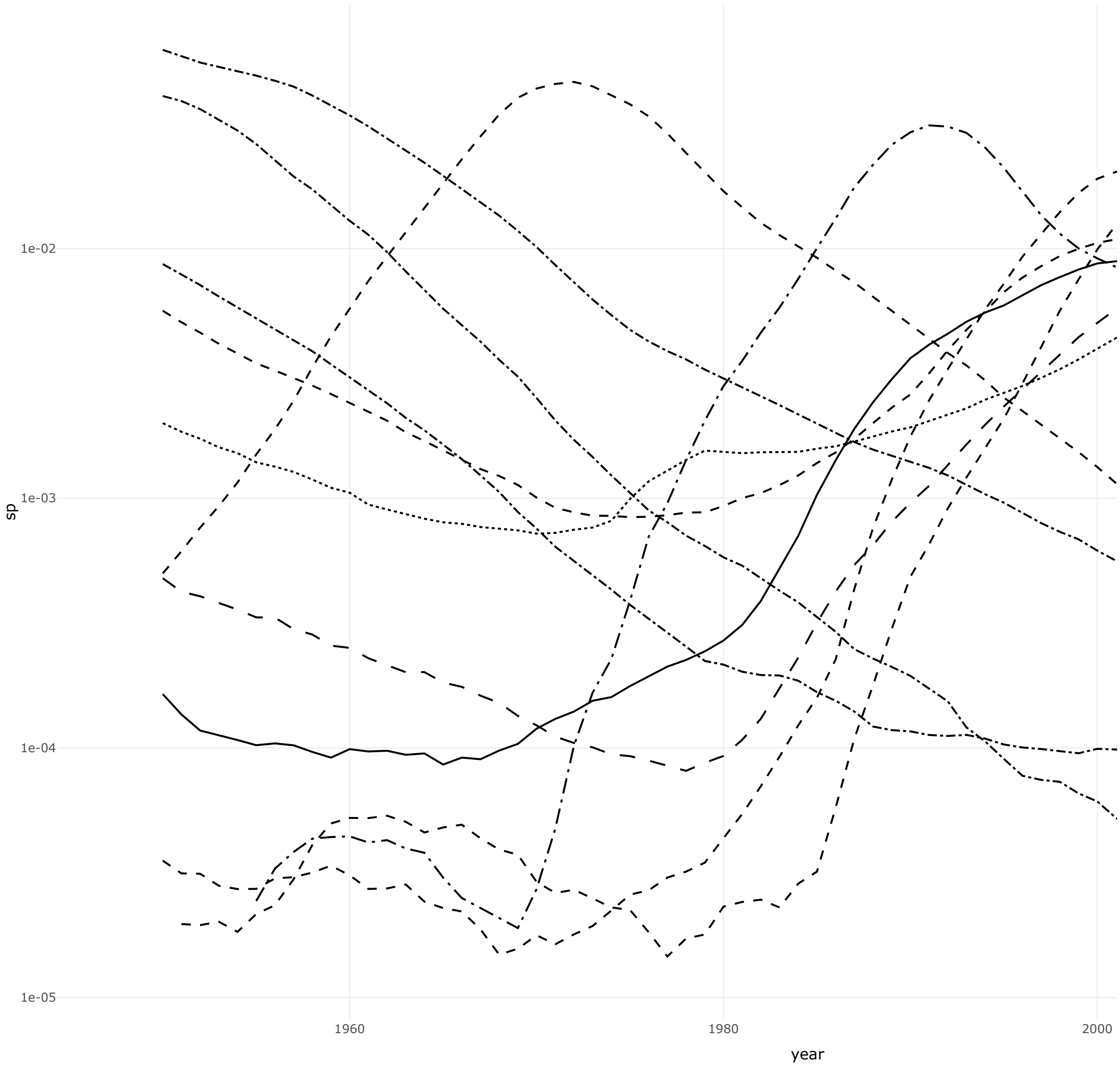


Patterns of popularity

```
df_lorenz_fr |>
  filter(year>1947) |>
  # group_by(sex, name) |>
  # arrange(year) |>
#  mutate(increase=log(rr/lag(rr))) |>
  # ungroup() |>
  filter(sex=='M', name %in% c('STÉPHANE', 'KEVIN', 'ENZO')) |>
  ggplot() +
  aes(x=year, y=1/(1.00001-rr), shape=name, linetype=name) +
  geom_line() +
  scale_y_log10()
```




```
(df_lorenz_fr |>
  filter(year>1947) |>
  group_by(sex, name) |>
  arrange(year) |>
  mutate(sp=slider::slide_vec(p, mean, .before = 2, .after = 2, .complete = T)) |>
  ungroup() |>
  filter(sex=='M', name %in% c('STÉPHANE', 'KEVIN', 'ENZO', 'THÉO', 'GABRIEL', 'ARTHUR', '
  ggplot() +
  aes(x=year, y=sp, shape=name, linetype=name) +
  geom_line() +
  scale_y_log10()) |> plotly::ggplotly()
```



Names that were rare in 1948 and made it to the top 10 afterwards

```
df_ratio_pop <- df_lorenz_fr |>
  filter(year>1947) |>
  group_by(sex, name) |>
  arrange(year) |>
  summarise(ratiop=max(p)/min(p),
            maxrr=max(rr),
            minp=min(p),
            maxp=max(p),
            year_max = min(year) + which.max(p) -1,
            .groups="drop")
```

```
df_ratio_pop |>
  filter(name %in% c('STÉPHANE', 'ENZO', 'KEVIN', 'THÉO'), sex=='M')
```

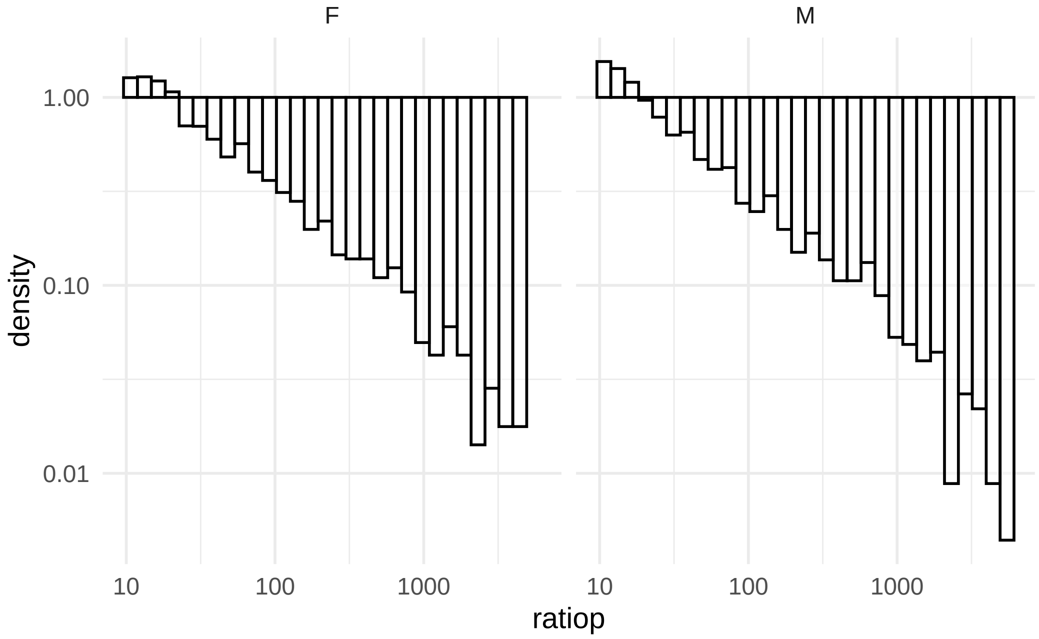
```
# A tibble: 4 x 7
  sex   name     ratiop maxrr       minp   maxp year_max
  <fct> <chr>     <dbl> <dbl>      <dbl>  <dbl>    <dbl>
1 M     ENZO      2187.   1    0.0000104  0.0227     2005
2 M     KEVIN     4803.   1    0.00000709 0.0341     1988
3 M     STÉPHANE   514.   1    0.0000984  0.0506     1971
```

```
4 M     THÉO       1995.   1.00 0.0000113  0.0226      2001
```

```
df_ratio_pop |>
  filter(sex=='M') |>
  arrange(desc(maxp)) |>
  head(200)
```

```
# A tibble: 200 x 7
   sex   name        ratiop maxrr      minp   maxp year_max
   <fct> <chr>        <dbl> <dbl>     <dbl>  <dbl>    <dbl>
 1 M     JEAN          55.3 1     0.00183   0.101      1948
 2 M     MICHEL       494.  0.999 0.000143  0.0707     1948
 3 M     PHILIPPE     396.  1     0.000145  0.0574     1963
 4 M     THIERRY     1789.  1     0.0000309 0.0553     1964
 5 M     ALAIN        830.  0.998 0.0000643 0.0533     1950
 6 M     NICOLAS      106.  1     0.000497  0.0530     1980
 7 M     SÉBASTIEN    656.  1     0.0000788 0.0517     1976
 8 M     CHRISTOPHE  784.  1     0.0000654 0.0513     1969
 9 M     STÉPHANE     514.  1     0.0000984 0.0506     1971
10 M     PATRICK      637.  0.999 0.0000789 0.0503     1956
# i 190 more rows
```

```
df_ratio_pop |>
  filter(ratiop > 10) |>
  ggplot() +
  aes(x=ratiop, y=after_stat(..density..)) +
  scale_y_log10() +
  scale_x_log10() +
  geom_histogram( fill="white", alpha=.5, color="black") +
#  stat_function()
  facet_wrap(~ sex)
```



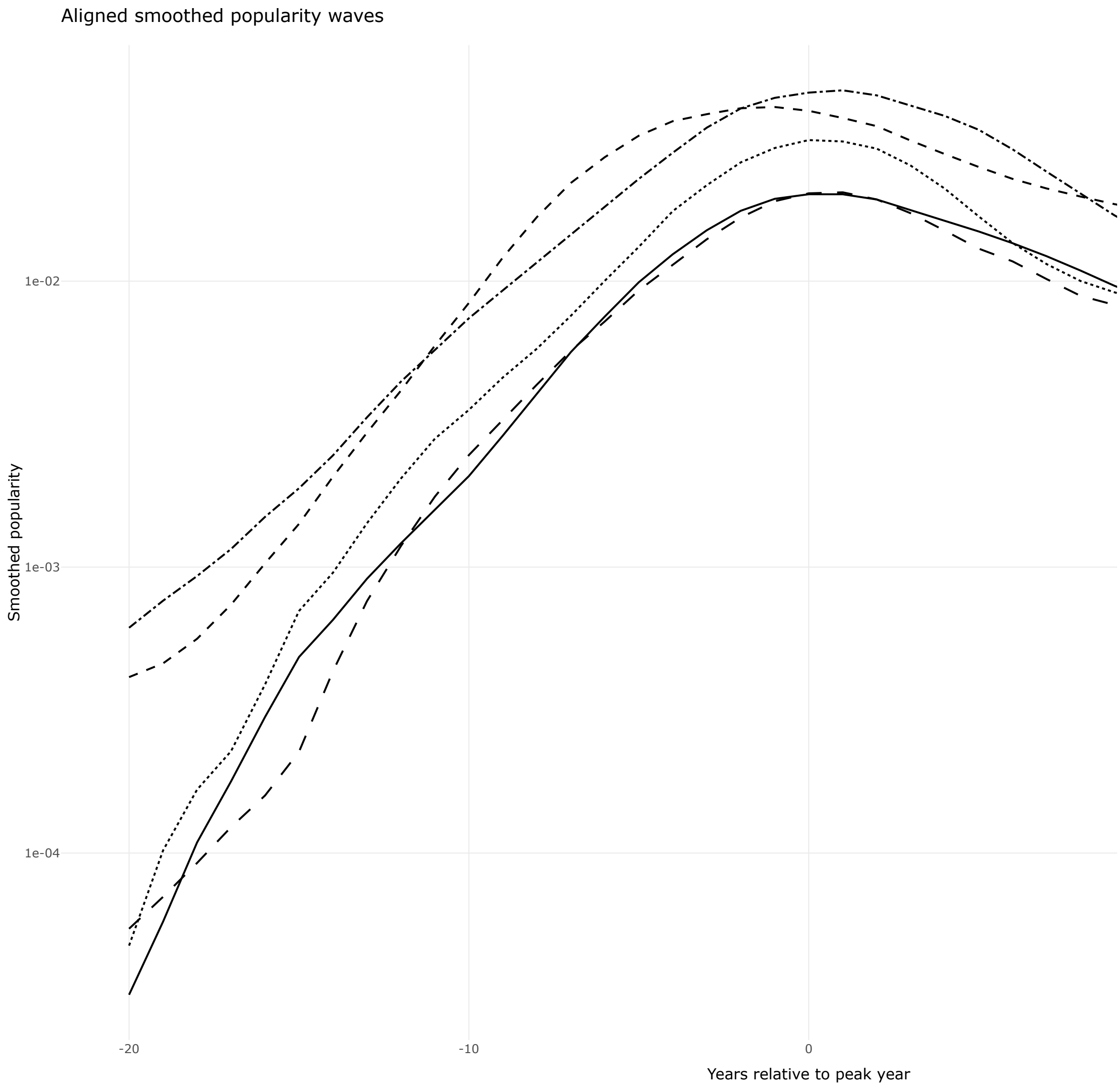

```
df_ratio_pop <- df_lorenz_fr |>
  filter(year>1947) |>
  group_by(sex, name) |>
  arrange(year) |>
```

```
  mutate(ymax= year[which.max(p)]) |>
  mutate(ryear = year-ymax) |>
  mutate(sp=slider::slide_vec(p, mean, .before = 2, .after = 2, .complete = T))  |>
  filter(between(ryear, -20, 20))
```

```
df_ratio_pop |>
  filter(name=='KEVIN', sex=='M')
```

```
# A tibble: 41 x 11
# Groups:   sex, name [1]
   sex   name   year     n country    rr       L          p  ymax ryear       sp
   <fct> <chr> <int> <dbl> <chr>   <dbl>   <dbl>      <dbl> <int> <int>    <dbl>
 1 M     KEVIN  1971     4 fr      0.251 0.00383 0.00000867  1991   -20  4.74e-5
 2 M     KEVIN  1972    35 fr      0.788 0.0334  0.0000763   1991   -19  1.02e-4
 3 M     KEVIN  1973    58 fr      0.841 0.0462  0.000130    1991   -18  1.67e-4
 4 M     KEVIN  1974   118 fr      0.890 0.0719  0.000283    1991   -17  2.27e-4
 5 M     KEVIN  1975   130 fr      0.899 0.0812  0.000336    1991   -16  3.88e-4
 6 M     KEVIN  1976   116 fr      0.892 0.0808  0.000311    1991   -15  7.02e-4
 7 M     KEVIN  1977   340 fr      0.941 0.146   0.000883    1991   -14  9.56e-4
 8 M     KEVIN  1978   645 fr      0.961 0.210   0.00170     1991   -13  1.42e-3
 9 M     KEVIN  1979   606 fr      0.958 0.199   0.00155     1991   -12  2.04e-3
10 M     KEVIN  1980  1101 fr      0.974 0.285   0.00267     1991   -11  2.81e-3
# i 31 more rows
```

```
(df_ratio_pop|>
  filter(name %in% c('STÉPHANE', 'ENZO', 'THÉO', 'KEVIN', 'JULIEN'), sex=='M') |>
  ggplot() +
  aes(x=ryear, y=sp,  shape=name, linetype=name, label=ymax) +
  geom_line() +
  scale_y_log10() +
  labs(title="Aligned smoothed popularity waves",
       subtitle="") +
  xlab("Years relative to peak year") +
  ylab("Smoothed popularity"))|>
  plotly::ggplotly()
```

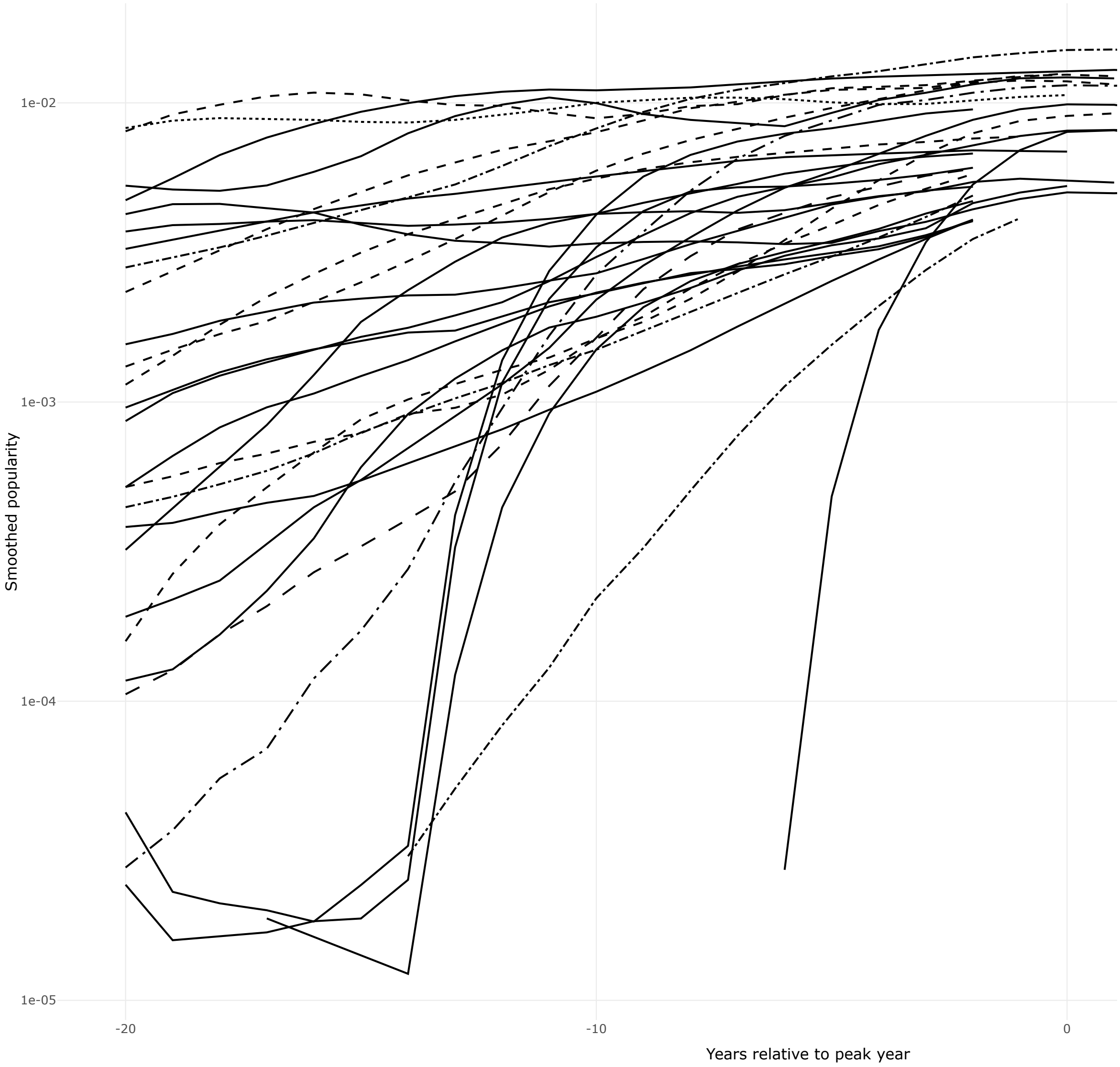



```
df_vieille_france <- df_ratio_pop  |>
  filter(min(ryear)>=-5, max(p)>1e-3)  |>
  distinct(sex, name)  |>
  arrange(sex, name)
```

```
df_nouvelle_france <- df_ratio_pop  |>
  filter(max(ryear)<=10, max(p)>5e-3)  |>
  distinct(sex, name)  |>
  arrange(sex, name)
```

```
trendy_names <- pull(filter(df_nouvelle_france, sex=='M'), name)
spam <- (df_ratio_pop|>
  filter(name %in% trendy_names, sex=='M'))

(spam  |>
  ggplot() +
  aes(x=ryear, y=sp,  shape=name, linetype=name, label=ymax) +
  geom_line() +
  scale_y_log10() +
  # labs(title="Aligned smoothed popularity waves",
#        subtitle="") +
```

```
  xlab("Years relative to peak year") +
  ylab("Smoothed popularity")) |>
  plotly::ggplotly()
```




```
df_ratio_pop |>
  filter(max(ryear)<=3, max(p)>5e-3) |>
  distinct(sex,name) |>
  arrange(sex, name)
```

```
# A tibble: 36 x 2
# Groups:   sex, name [36]
   sex   name
   <fct> <chr>
 1 F     ADÈLE
 2 F     AGATHE
 3 F     ALBA
 4 F     AMBRE
 5 F     ANNA
 6 F     CHARLIE
 7 F     INAYA
 8 F     IRIS
 9 F     JULIA
```

```
10 F      LOU
# i 26 more rows
```

Fitting a Zipf distribution

Choosing scales

Animation

Classifying names according to their pattern of popularity

Now, we focus on names that made it to the top 300 at least once since year 1948. We attempt to classify them according to their pattern of popularity,