genderreport

Jennifer Young

7/31/2021

Introduction

One important decision parents make is naming their children. In this study, we will look at popular names and gender neutral names. A soon-to-be parent who is researching such an important decision may want to consider data on a name to see how neutral the name is considered to be. Choosing a name that is almost equally chosen for both sexes can be the goal for parents. We will consider several names that have been labeled gender neutral and consider how they have been used by both biological sexes historically and we will use a model that predicts when the name is considered male or female based on it's use in the US. Need to download packages if not done already

```
local({r <- getOption("repos")
r["CRAN"] <- "http://cran.r-project.org"
options(repos=r)
})

install.packages("remotes") # if necessary
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```
remotes::install_github("lmullen/gender")
```

```
## Downloading GitHub repo lmullen/gender@HEAD
```

```
##
##      checking for file '/private/var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T/Rtmpq71kwE/remotes5
##   -  preparing 'gender':
##      checking DESCRIPTION meta-information ...  v  checking DESCRIPTION meta-information
##   -  checking for LF line-endings in source and make files and shell scripts
##   -  checking for empty or unneeded directories
##   -  building 'gender_0.5.4.1000.tar.gz'
##
##
```

```
install.packages("rTool")#or install through RStudio
```

```
## Warning: package 'rTool' is not available (for R version 4.0.2)
```

```
install.packages('plyr', repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```r
install.packages("babynames")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```r
install.packages("dplyr")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```r
install.packages("tidyr")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```r
install.packages("ggplot2")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```r
install.packages("gridExtra")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```r
install.packages("magrittr")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```r
install.packages("devtools")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```r
install.packages("tidyverse")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```r
install.packages("caret")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```r
install.packages("e1071")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

```r
install.packages("randomForest")
```

```
##
## The downloaded binary packages are in
##  /var/folders/m7/mqtw78w13fzf7324by81zj7r0000gn/T//Rtmpq71kwE/downloaded_packages
```

I had to install psych, naivebayes, gender, randomForest, tinytex and genderdata through RStudio instead

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(plyr)
```

```
## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:purrr':
##
##     compact
```

```r
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```r
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(gender)
library(tibble)
library(devtools)
```

```
## Loading required package: usethis
```

```r
library(babynames)
library(dplyr)
library(tidyr)
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
## The following object is masked from 'package:purrr':
##
##     set_names
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
library(e1071)
library(tinytex)
data(babynames)
head(babynames)
```

```
## # A tibble: 6 x 5
##    year sex   name          n   prop
##   <dbl> <chr> <chr>     <int>  <dbl>
## 1  1880 F     Mary       7065 0.0724
## 2  1880 F     Anna       2604 0.0267
## 3  1880 F     Emma       2003 0.0205
## 4  1880 F     Elizabeth  1939 0.0199
## 5  1880 F     Minnie     1746 0.0179
## 6  1880 F     Margaret   1578 0.0162
```

```r
tail(babynames)
```

```
## # A tibble: 6 x 5
##    year sex   name       n       prop
##   <dbl> <chr> <chr>  <int>      <dbl>
## 1  2017 M     Zyhier     5 0.00000255
## 2  2017 M     Zykai      5 0.00000255
## 3  2017 M     Zykeem     5 0.00000255
```

```
## 4  2017 M     Zylin      5 0.00000255
## 5  2017 M     Zylis      5 0.00000255
## 6  2017 M     Zyrie      5 0.00000255
```

Methods

Data visualization was used to look at specific names that are often considered to be gender neutral through various baby name web sites. We can look at the names and graph their use for male and female babies and see their use for either gender in a historical context.

Drawn from Social Security Administration data, a sample of random names were taken from websites that identify gender neutral names the prospective parents could visit using a Google search.

From the earlier analysis on each name, 7 names were chosen that seemed the most neutral based on male and female trendlines in the charts.

Logistic regression, Random Forest and Naive Bayes were used to create models of accurate classification of names for being male, female, or somewhere in between, or gender neutral.

Finding out how many people were named X name is year X (sample)

```r
entered_name <- "Charlie"
entered_year <- 2017
result <- babynames %>% filter(name == entered_name) %>%
  filter(year == entered_year) %>%
  summarize(count = sum(n))
result
```

```
##   count
## 1  3676
```

Number of male and female names in dataset

```r
babynames %$%
  split(., sex) %>%
  lapply(. %$% length(unique(name)))
```

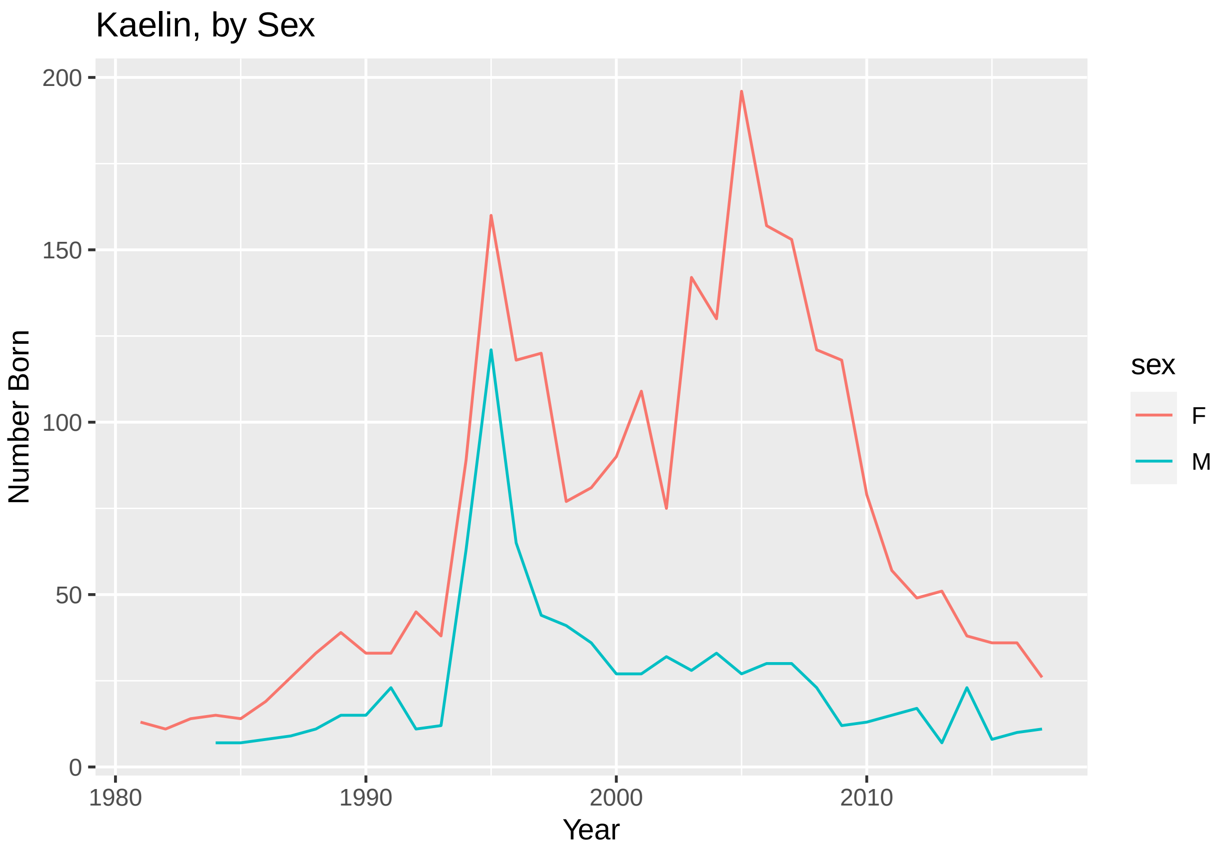
```
## $F
## [1] 67046
##
## $M
## [1] 40927
```

Gender Neutral Names by Sex from 1880-2017

For each chart, you can view the popularity of the name for use in both biological sexes between 1880-2017. I took a sample of random names from websites that identify gender neutral names the prospective parents could visit using a Google search The names that were tested were taken from a few popular websites, as that is likely the place where expectant parents would look. Some examples are: https://www.popsugar.com/family/Gender-Neutral-Baby-Names-34485564 https://www.mother.ly/child/top-50-gender-neutral-baby-names-youll-obsess-over-

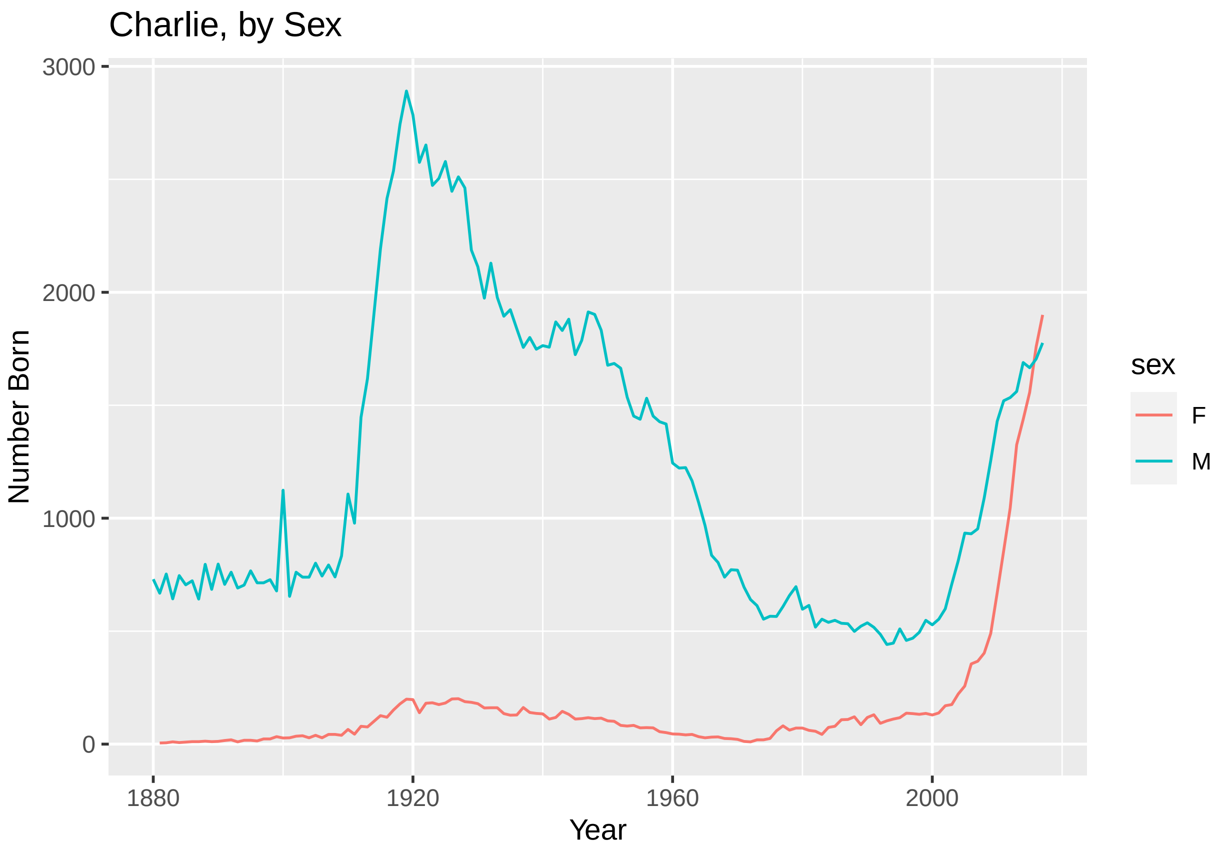
The name Kaelin seems to be used by both sexes but has fallen in popularity.

```r
babynames %>%
  filter(name == "Kaelin") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                    title = "Kaelin, by Sex")
```



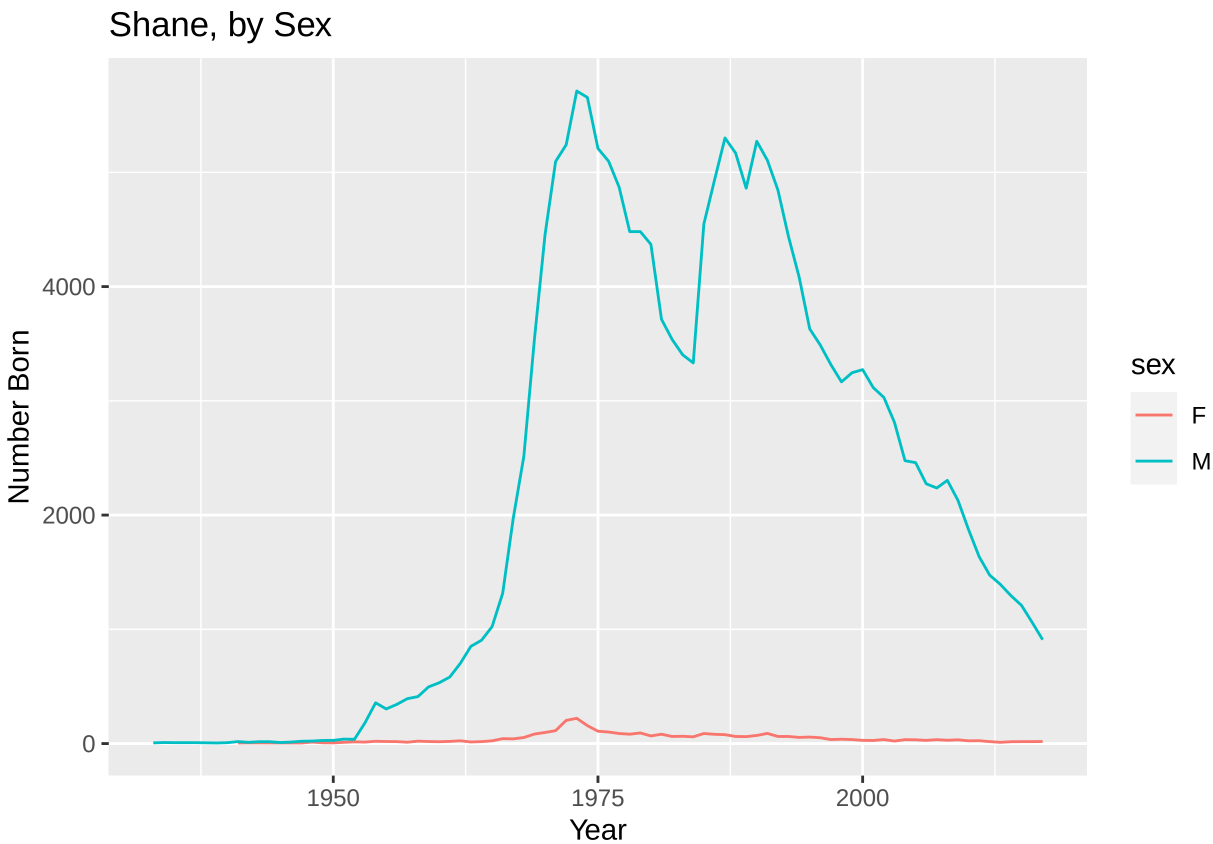
Charlie is another name for Charles and was traditionally used by males.
However, it has grown in popularity for both genders

```r
babynames %>%
  filter(name == "Charlie") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                    title = "Charlie, by Sex")
```



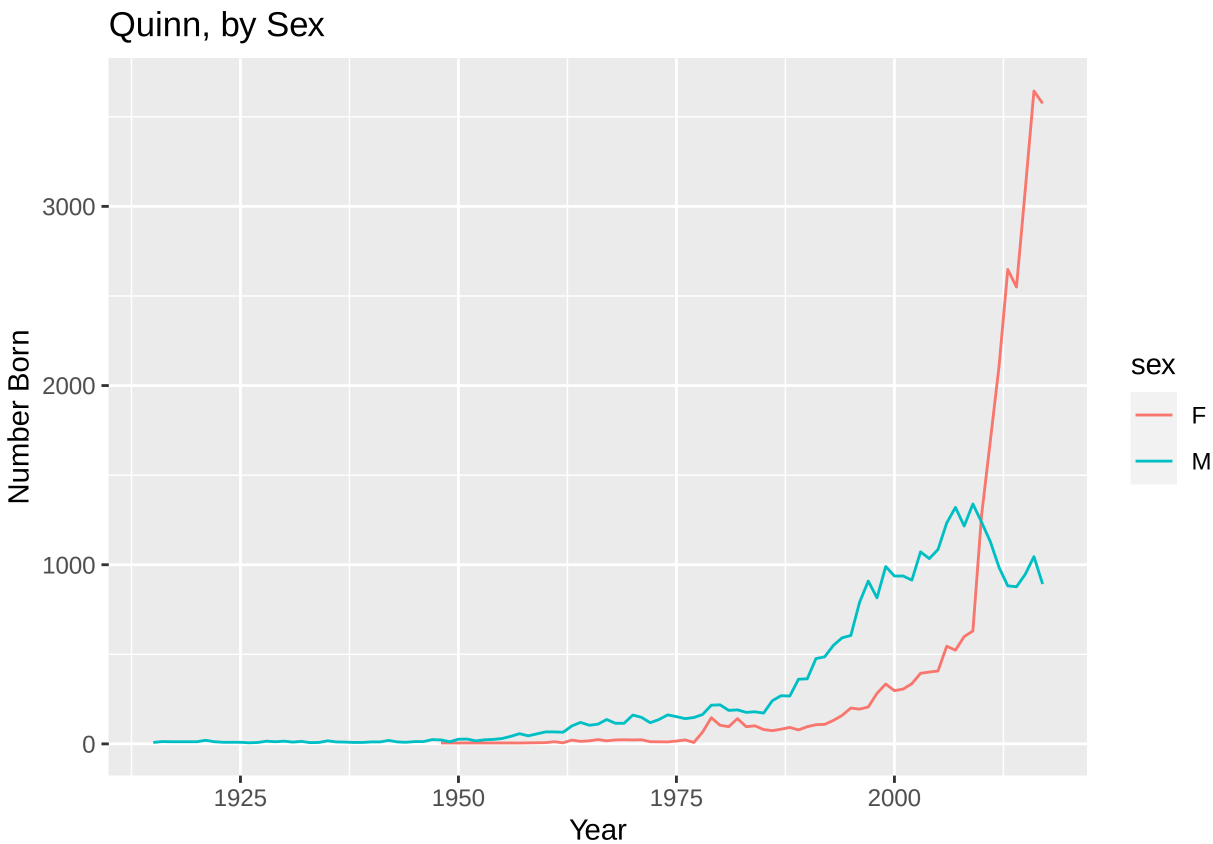
Shane is a name that was traditionally given to males but has decreased in popularity

```
babynames %>%
  filter(name == "Shane") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                      title = "Shane, by Sex")
```



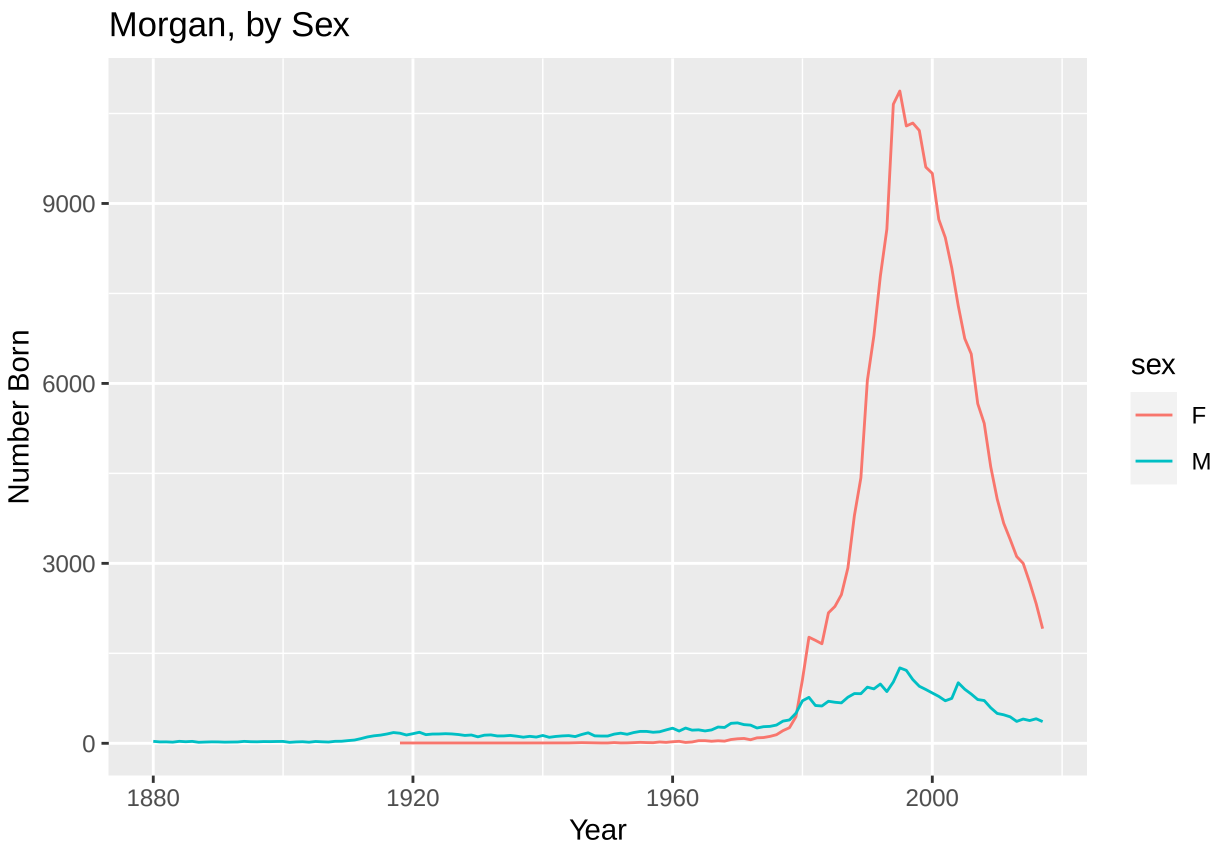
Quinn is a name that has been used by box sexes, but has grown in popularity in females

```
babynames %>%
  filter(name == "Quinn") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Quinn, by Sex")
```



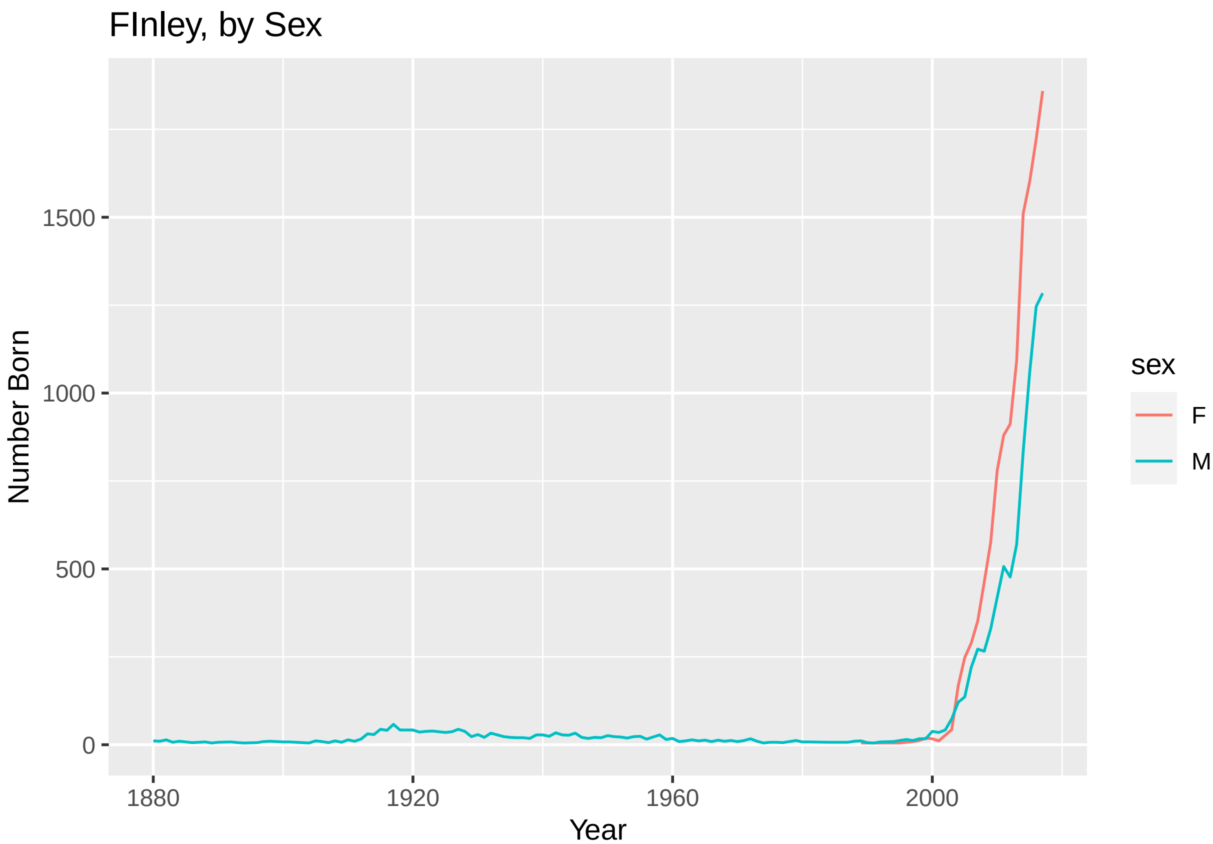
Morgan is a name that has historically been used by both sexes, but sharply rose among females 20 years ago. It has fallen in usage in females since then to meet male usage

```
babynames %>%
  filter(name == "Morgan") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Morgan, by Sex")
```



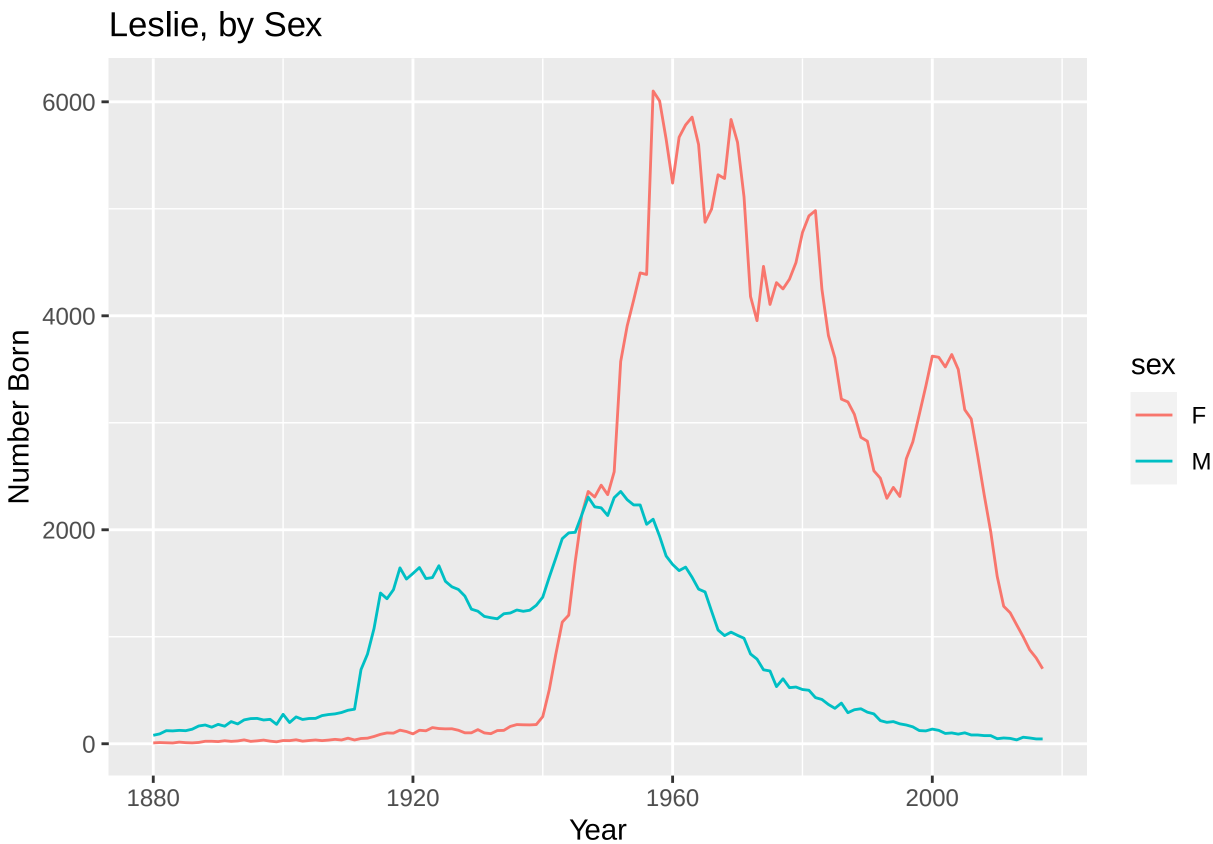
Finley has grown in usage for both sexes, but more for females

```
babynames %>%
  filter(name == "Finley") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "FInley, by Sex")
```



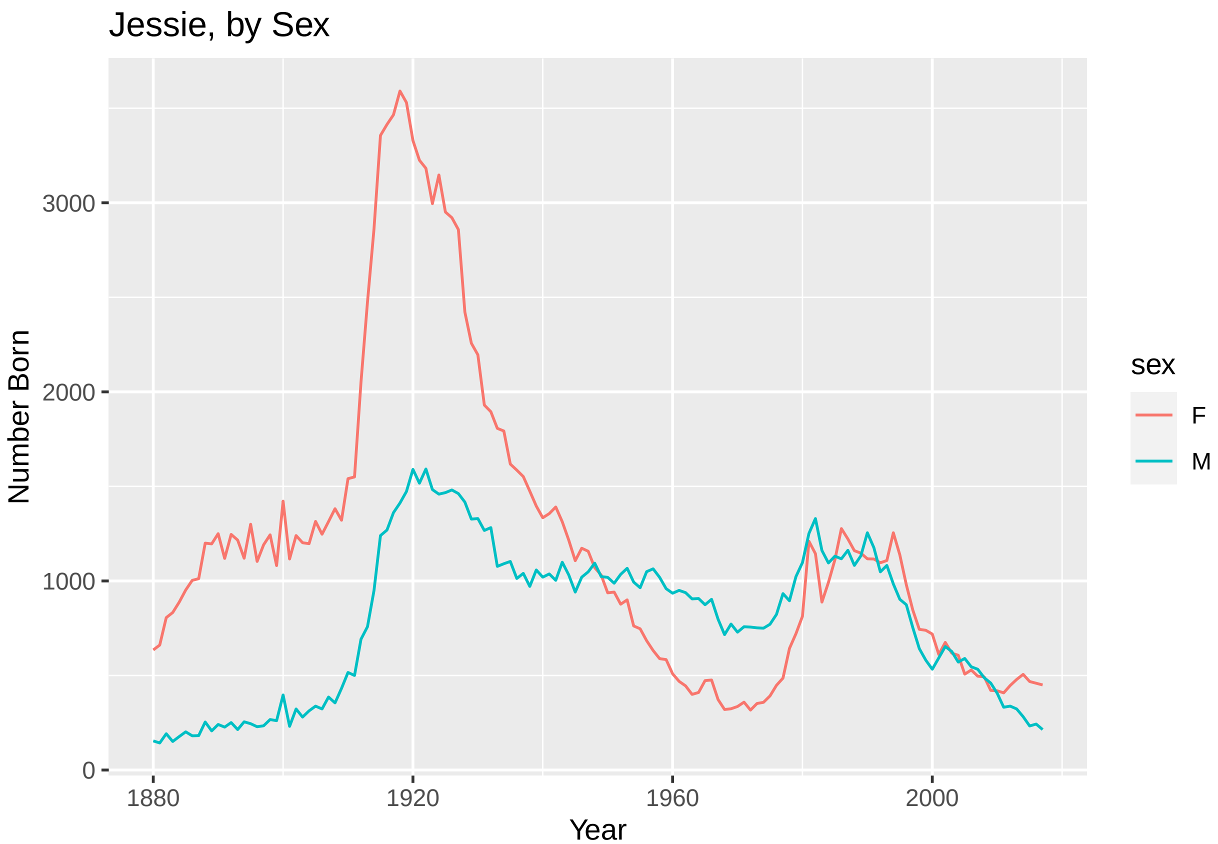
Leslie is a name that was historically used in both genders, although it's use in males has decreased over the last 60 years. It was popular for females in the last half of the last century. It has fallen in popularity overall.

```
babynames %>%
  filter(name == "Leslie") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Leslie, by Sex")
```



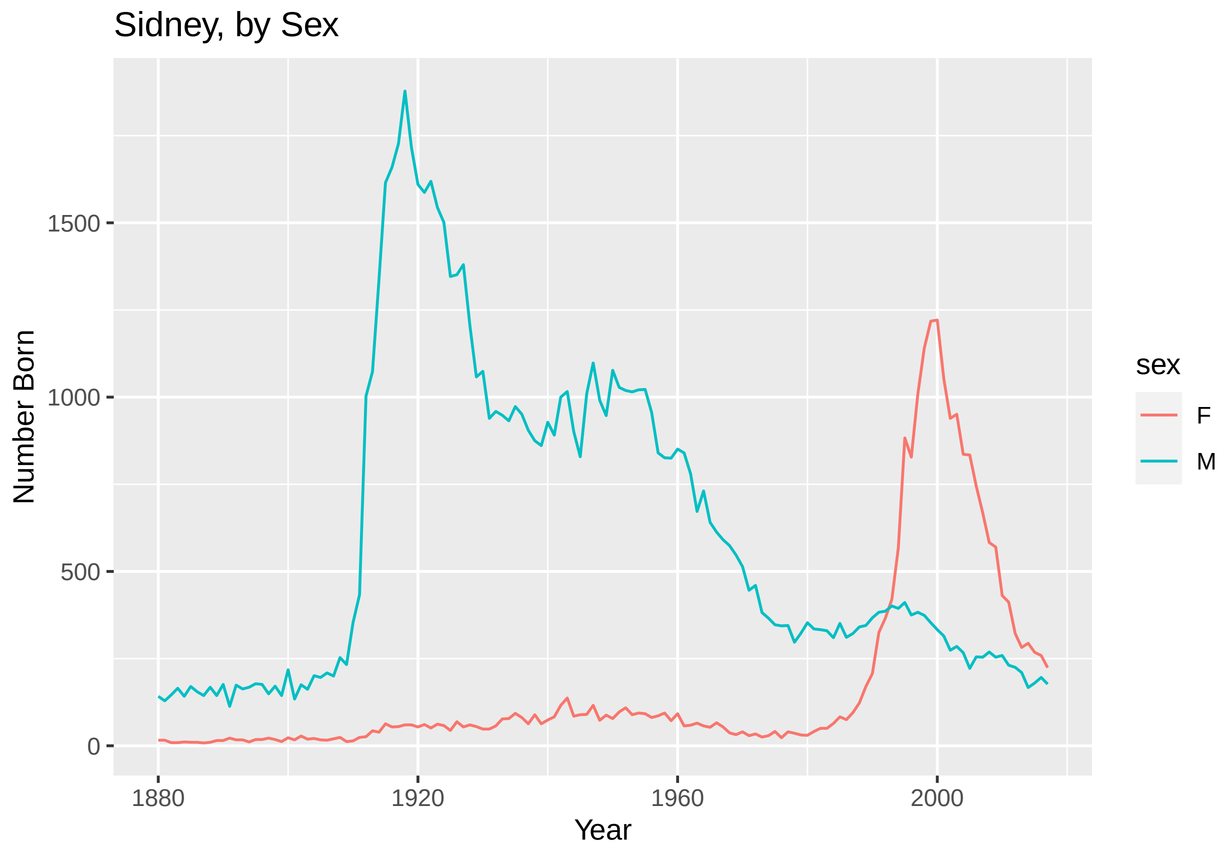
Jessie a name that was historically used in both genders and has fallen in popularity

```
babynames %>%
  filter(name == "Jessie") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Jessie, by Sex")
```



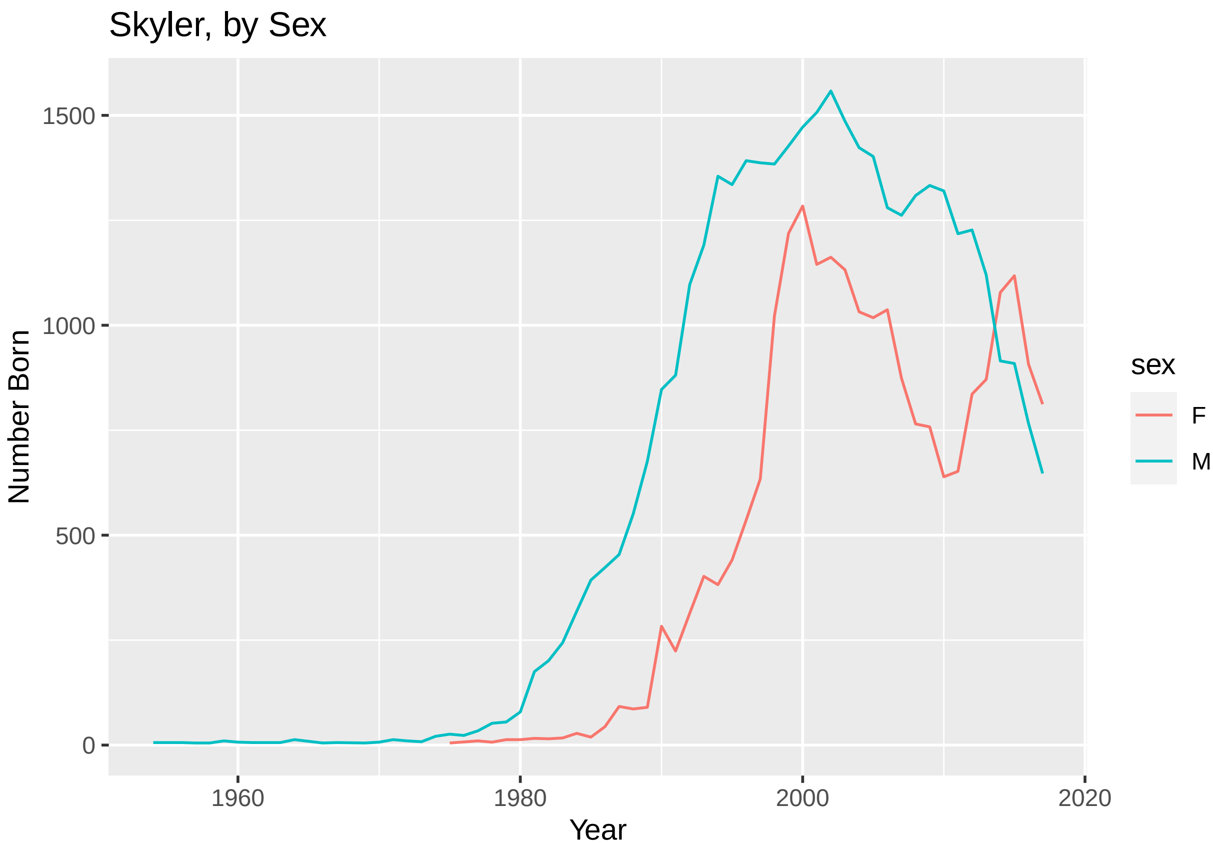
Sidney is a name that was historically used in both genders and has fallen in popularity for both genders

```
babynames %>%
  filter(name == "Sidney") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                      title = "Sidney, by Sex")
```



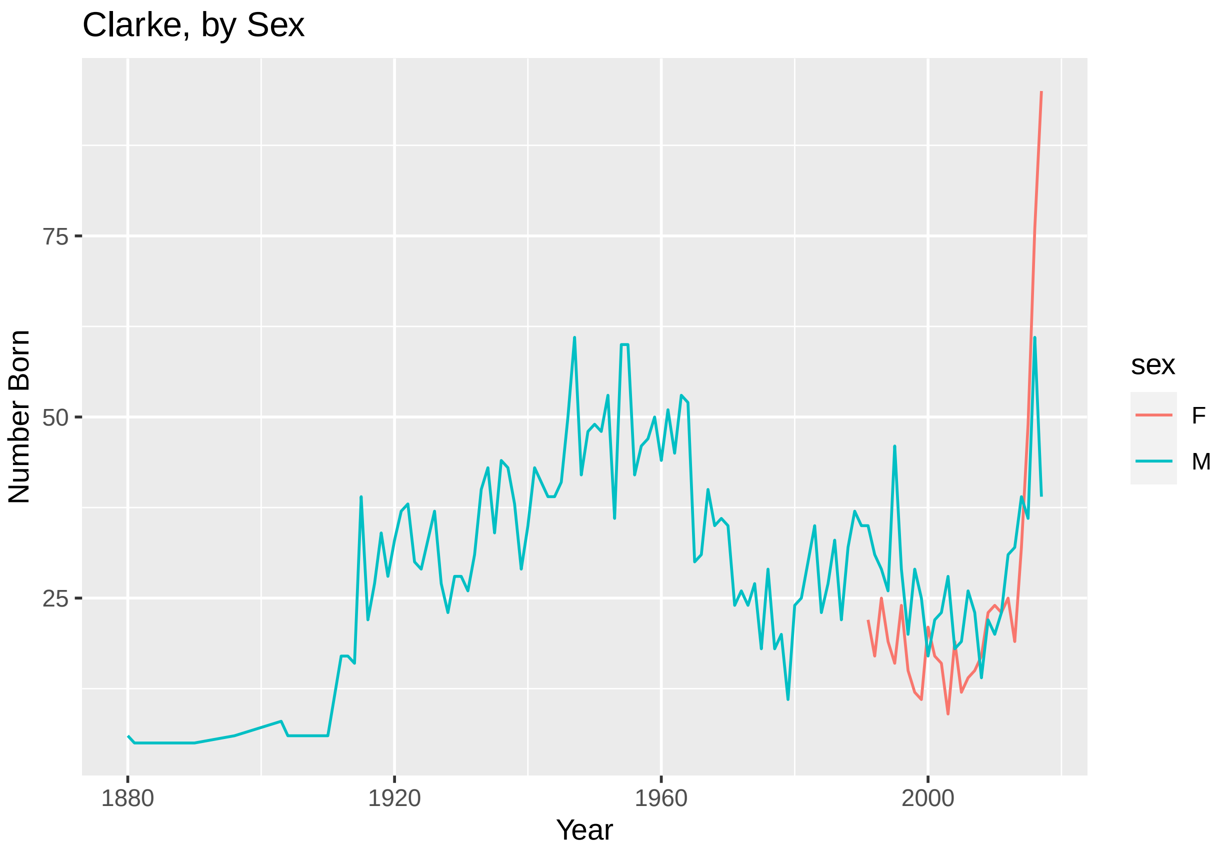
Skyler a name that was historically used in both genders and has risen in popularity in the last two decades

```
babynames %>%
  filter(name == "Skyler") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Skyler, by Sex")
```



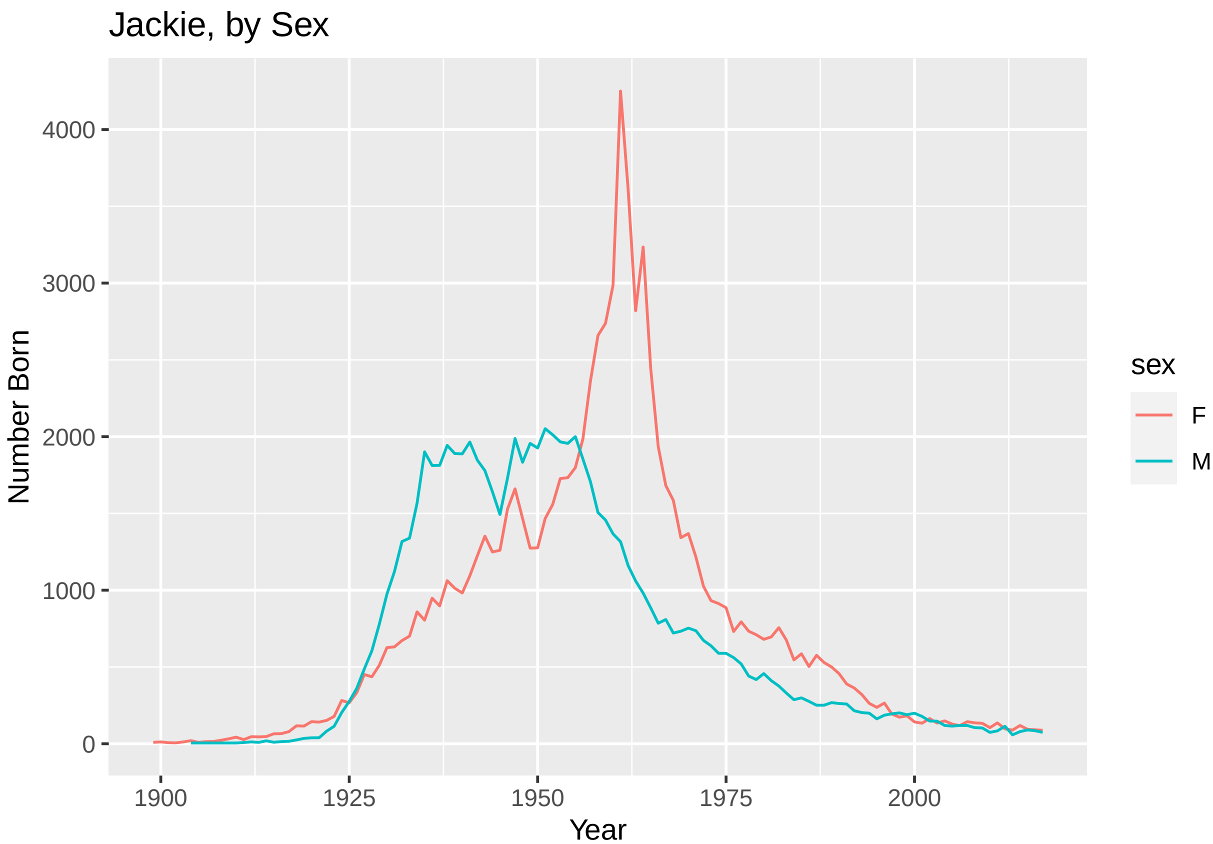
Clarke is a name that was historically used for males but has increased in popularity for females

```
babynames %>%
  filter(name == "Clarke") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Clarke, by Sex")
```



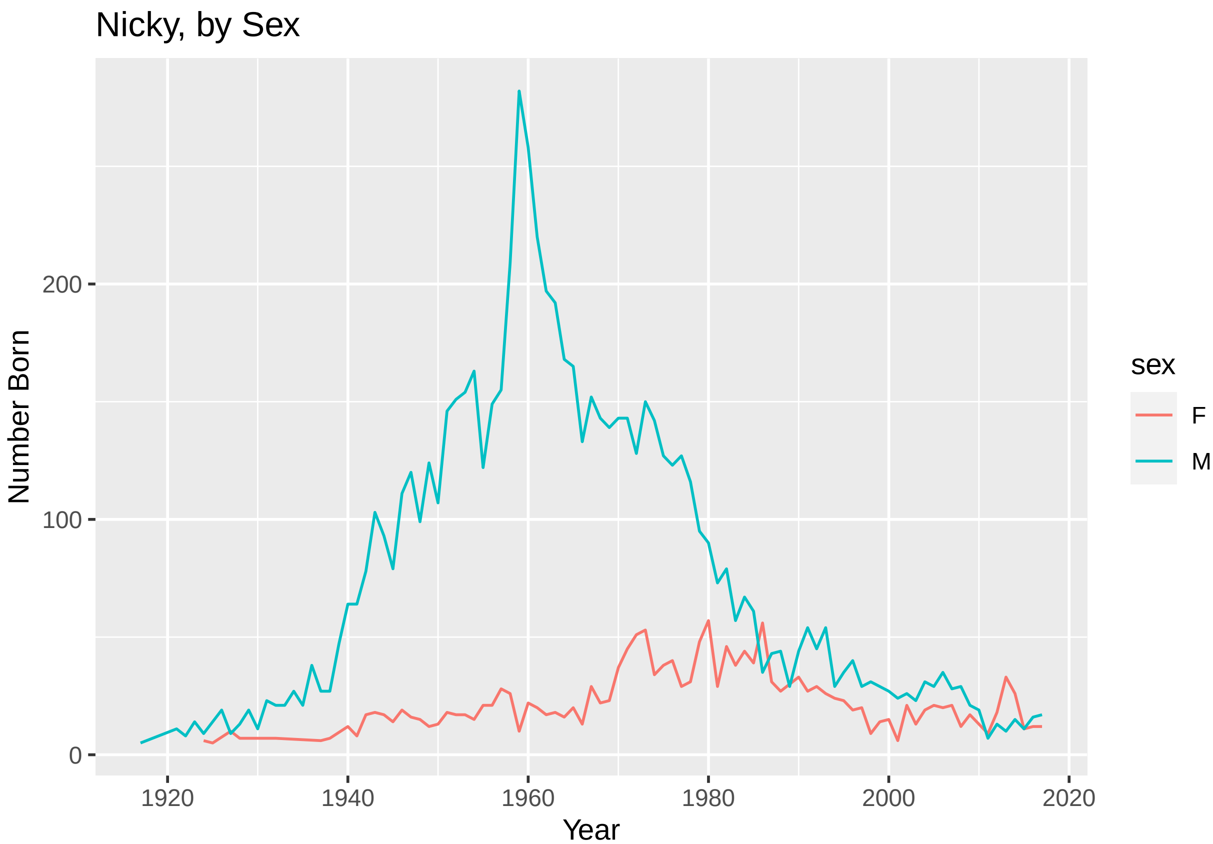
Jackie is a name that was historically used in both genders and has fallen in popularity for both genders

```
babynames %>%
  filter(name == "Jackie") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Jackie, by Sex")
```



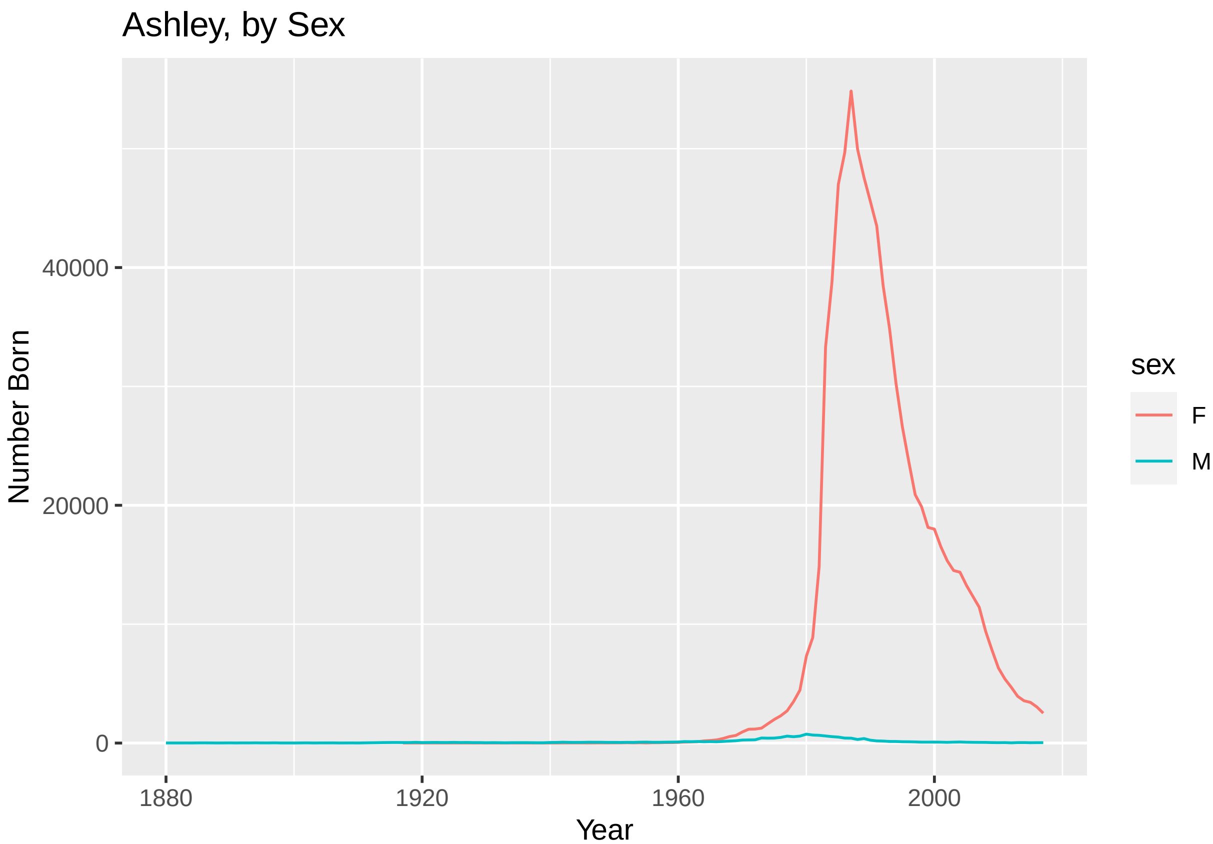
Nicky is a name that was historically used in both genders and has fallen in popularity for both genders

```
babynames %>%
  filter(name == "Nicky") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Nicky, by Sex")
```



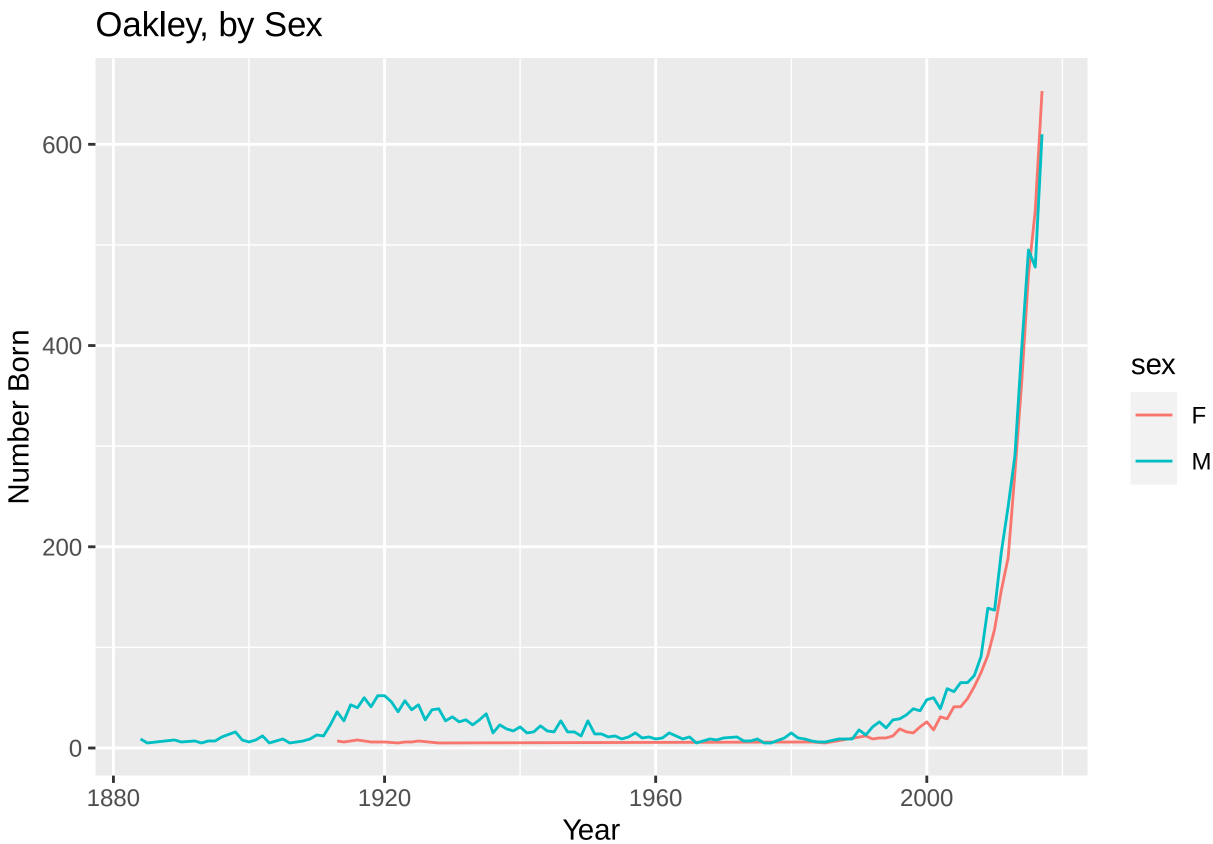
Ashley is a name that has generally been given to females. Gone With the Wind was an anomaly.

```
babynames %>%
  filter(name == "Ashley") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Ashley, by Sex")
```



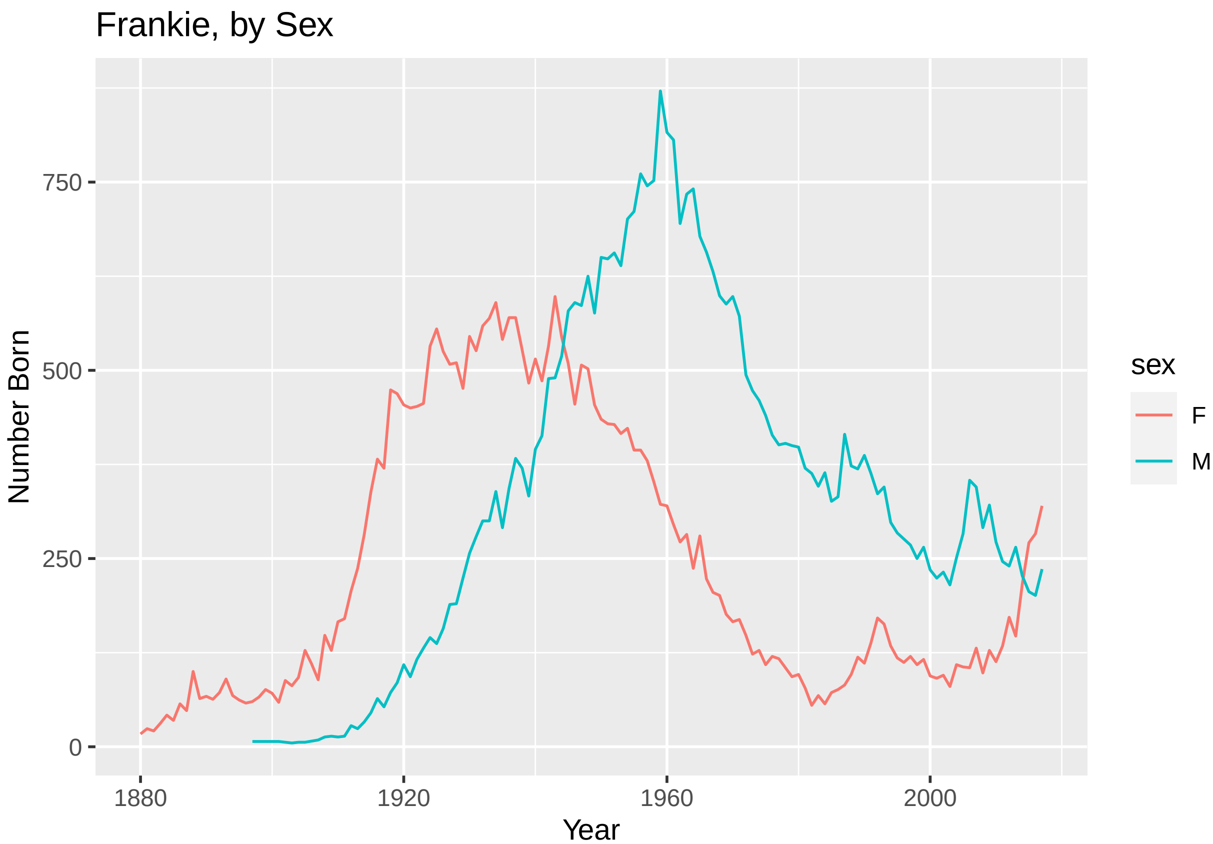
Oakley is the closest to gender neutral out of this data analysis and is extremely popular.

```
babynames %>%
  filter(name == "Oakley") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Oakley, by Sex")
```



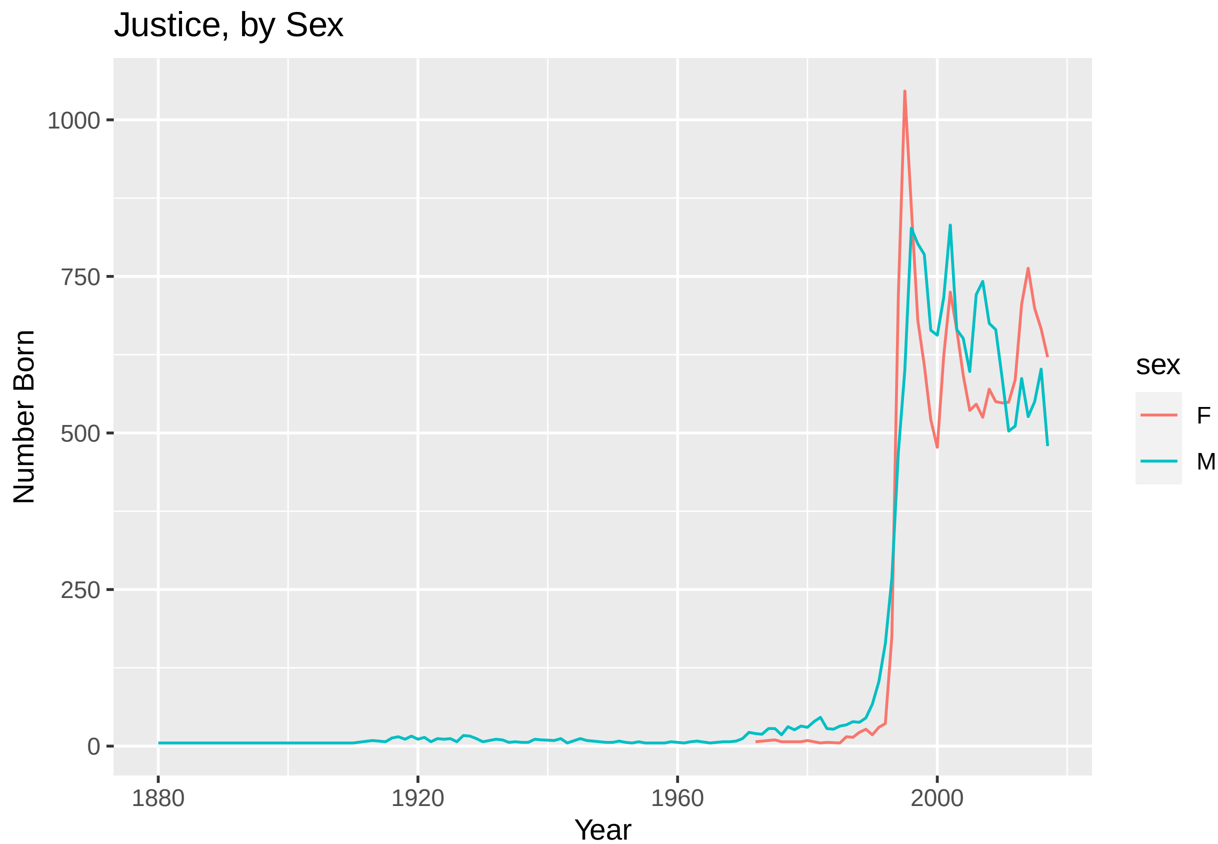
Frankie is a name that was historically used in both genders and is rising in popularity in females.

```
babynames %>%
  filter(name == "Frankie") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Frankie, by Sex")
```



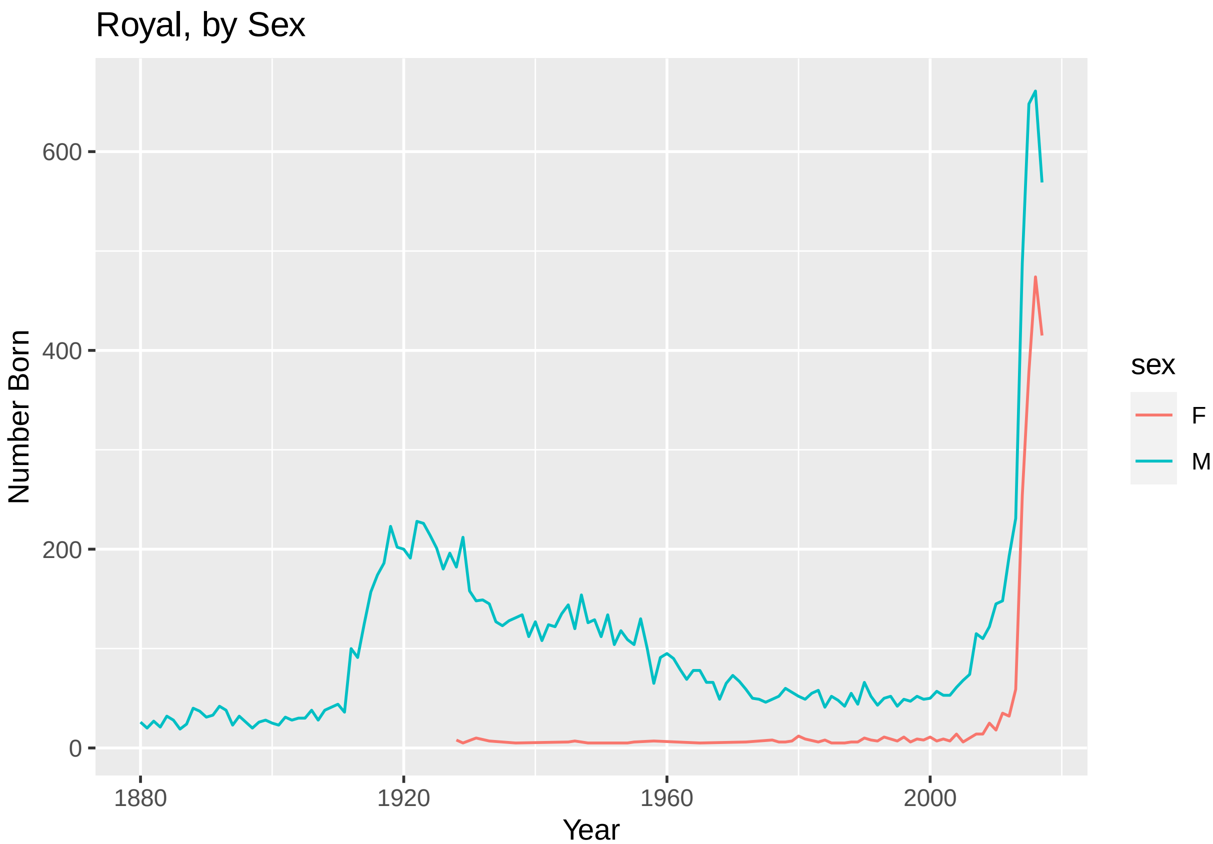
Justice is a name that was historically used in both genders and is a newer name compared to many others.

```
babynames %>%
  filter(name == "Justice") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                   title = "Justice, by Sex")
```



Royal is a name that was historically used for males but has risen in female in the past decade.

```
babynames %>%
  filter(name == "Royal") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Royal, by Sex")
```



What name has been the most popular over time for males? For females?

```
babynames %>% group_by(sex, name) %>%
  dplyr::summarize(median_prop = median(prop)) %>%
  top_n(1)
```

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

```
## Selecting by median_prop
```

```
## # A tibble: 2 x 3
## # Groups:   sex [2]
##   sex   name  median_prop
##   <chr> <chr>       <dbl>
## 1 F     Mary       0.0387
## 2 M     John       0.0442
```

```
namesinyear <- function(myyear){
  require(dplyr)
  yearnames <- babynames %>% filter(year == myyear) %>% distinct(name)
  yearnames <- sapply(yearnames[,"name"], as.character)
  return(length(yearnames))}
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

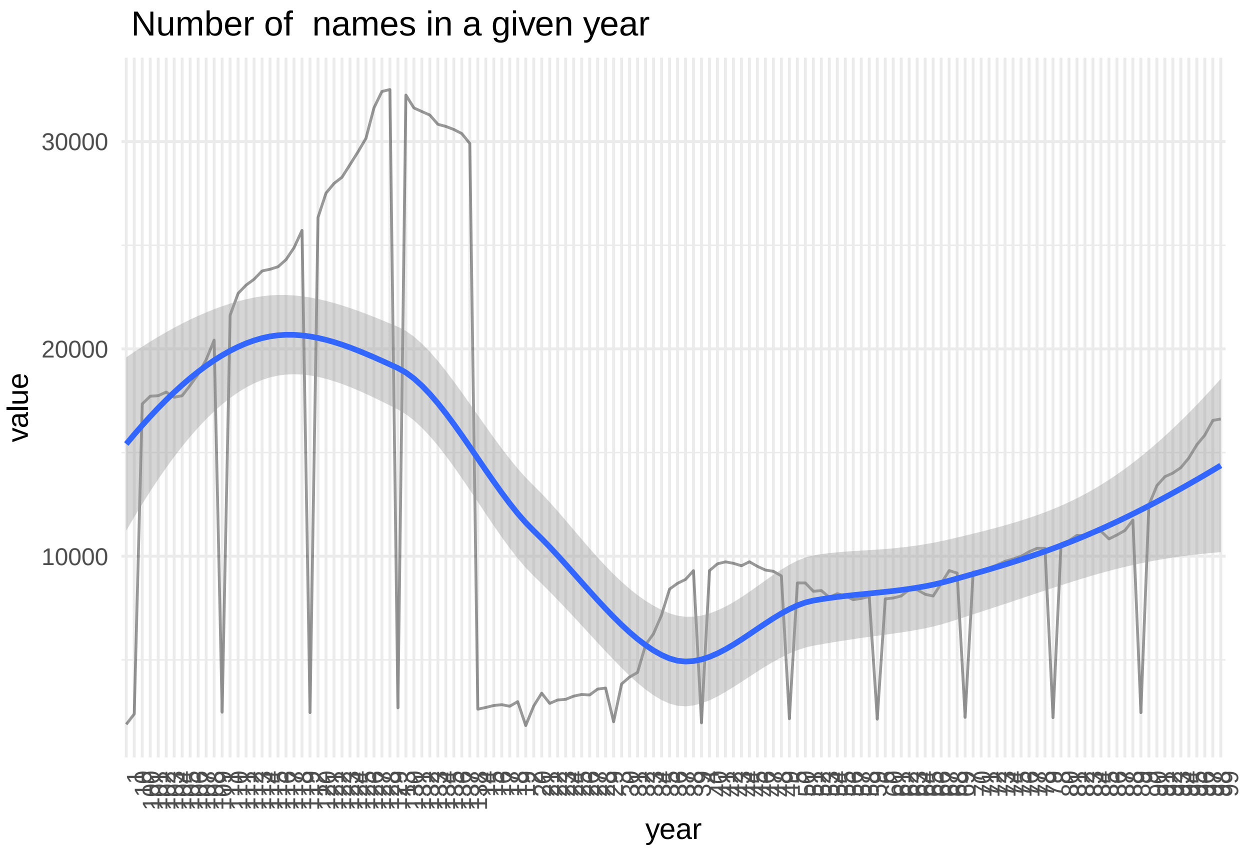
```r
namescount <- c()
for (year in 1880:2017){namescount <- c(namescount,namesinyear(year))}
namescount <- as.data.frame(namescount)
namescount$year <- rownames(namescount)
namescount <- melt(namescount)
```

```
## Using year as id variables
```

This is the number of names given each year in US (1880-2017). The number is rising, which means more names will be given for our data point.

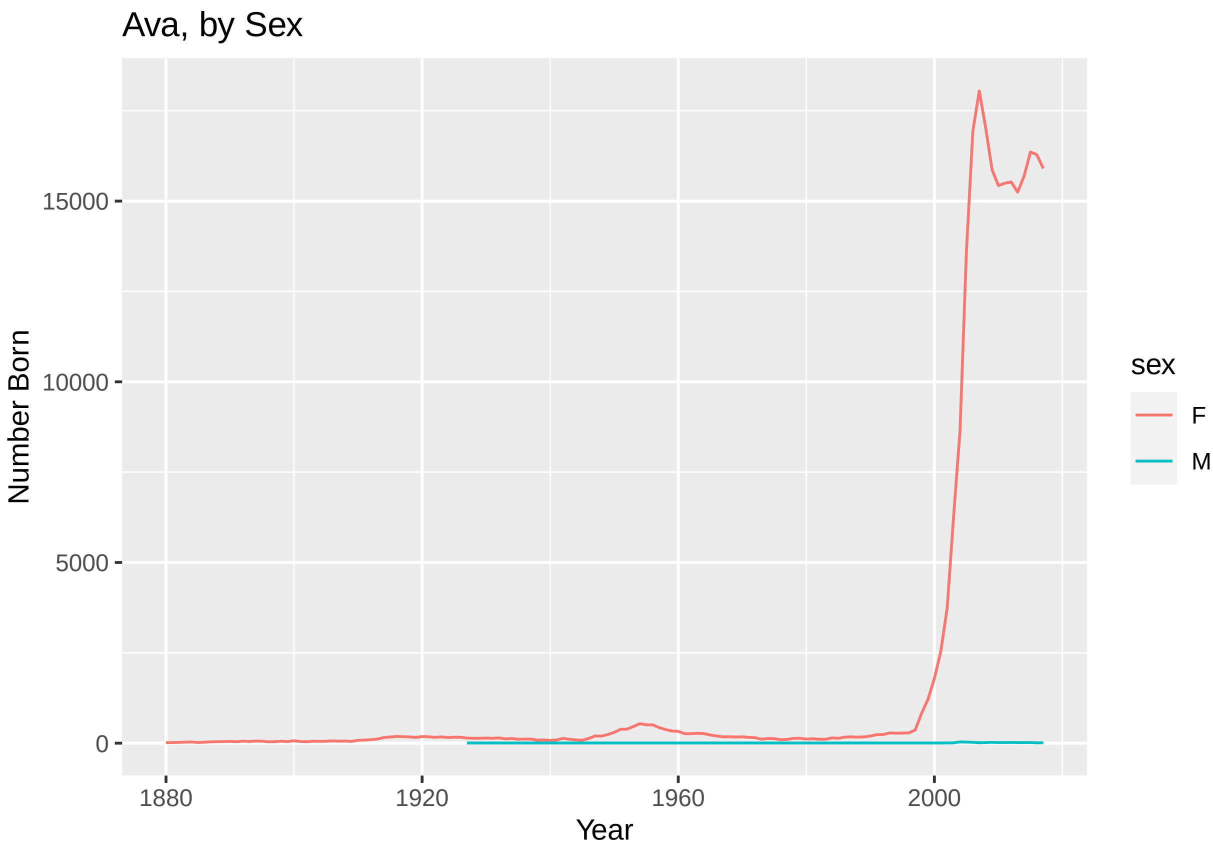
```r
ggplot(namescount, aes(x = year,y = value, group="variable")) + geom_line(alpha = 0.4) + theme_minimal()
```

```
## `geom_smooth()` using formula 'y ~ x'
```

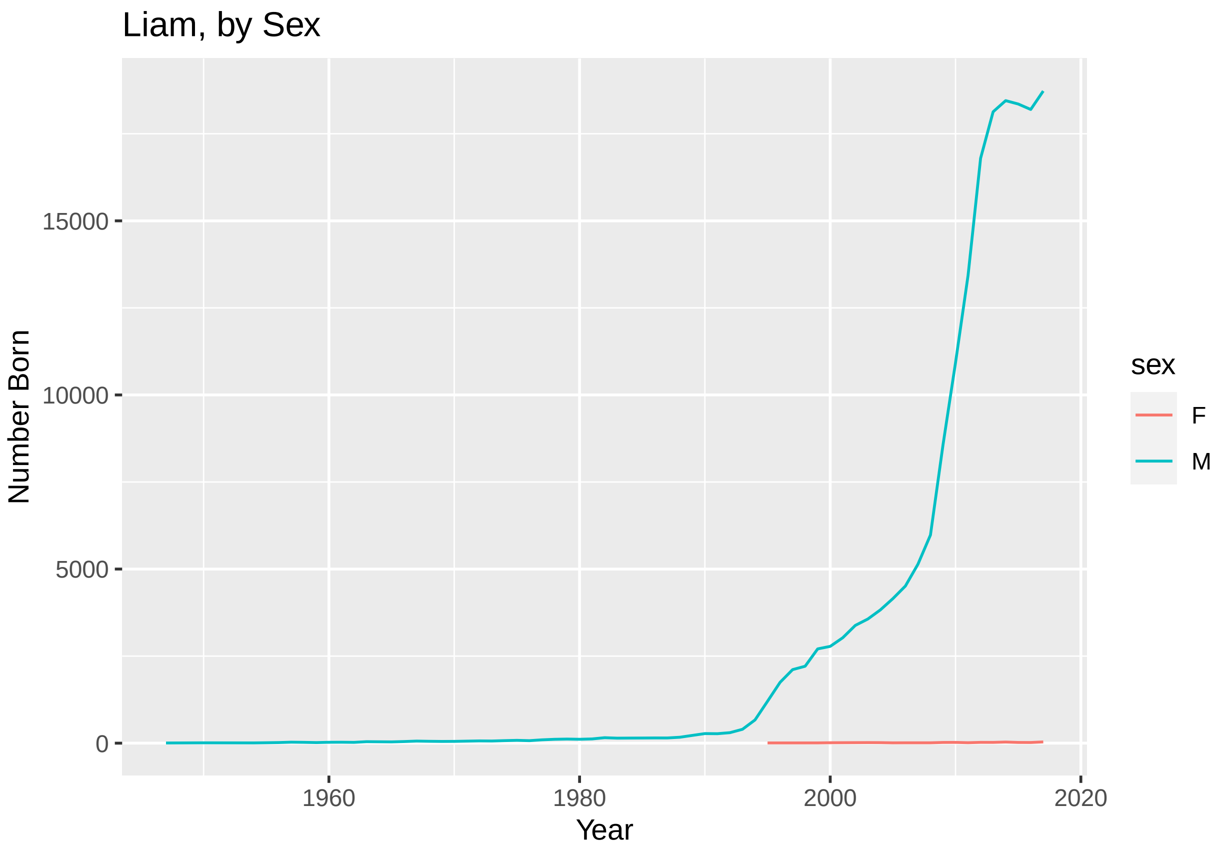


We can look at the popular names and see how gender neutral they appear.

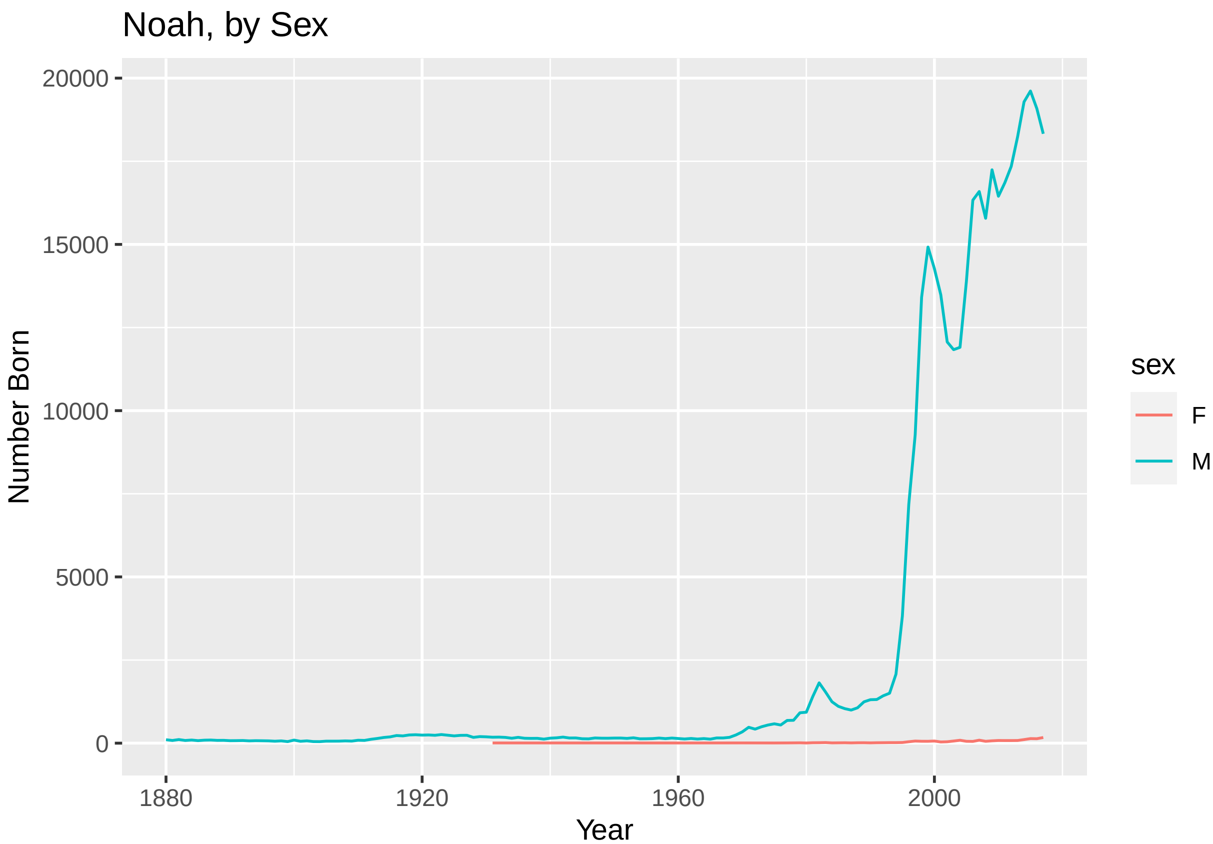
```r
babynames %>%
  filter(name == "Ava") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Ava, by Sex")
```



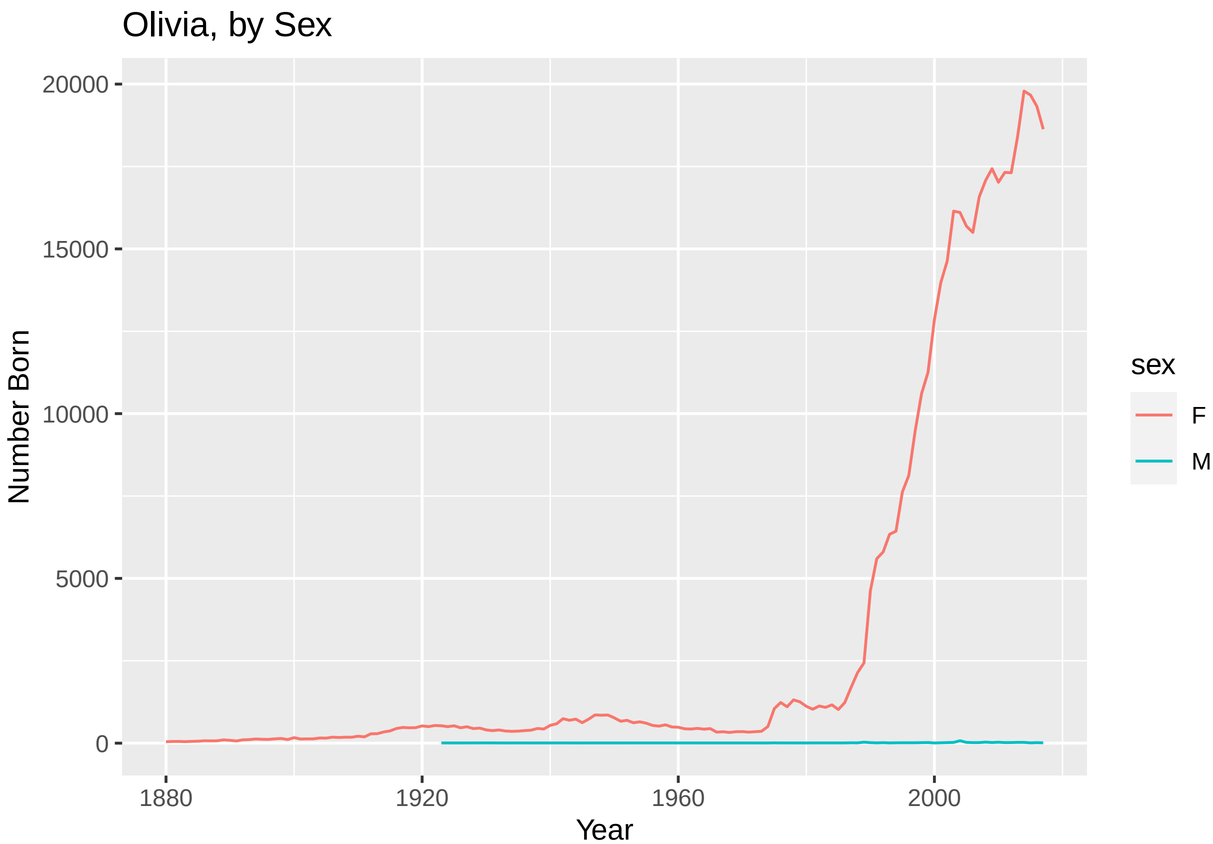

```r
babynames %>%
  filter(name == "Liam") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Liam, by Sex")
```




```r
babynames %>%
  filter(name == "Noah") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Noah, by Sex")
```




```
babynames %>%
  filter(name == "Olivia") %>%
  ggplot(aes(x = year, y = n)) +
  geom_line(aes(color = sex)) + labs(x = "Year", y = "Number Born",
                                     title = "Olivia, by Sex")
```



The most popular names in 2017 are not considered gender neutral. A parent would would be concerned about this would be unikely to choose these names.

Prediction of gender by name

I used (method = "ssa"): United States from 1930 to 2012. Drawn from Social Security Administration data.I took a sample of random names from websites that identify gender neutral names the prospective parents could visit using a Google search and graphed them earlier.
From the earlier analysis on each name, I chose 7 names that seemed the most neutral based on male and female trendlines in the charts.

```
head(gender)
```

```
##
## 1 function (names, years = c(1932, 2012), method = c("ssa", "ipums",
## 2     "napp", "kantrowitz", "genderize", "demo"), countries = c("United States",
## 3     "Canada", "United Kingdom", "Denmark", "Iceland", "Norway",
## 4     "Sweden"))
## 5 {
## 6     method <- match.arg(method)
```

```
ssa_names <- c("Charlie", "Royal", "Morgan", "Skyler",
               "Frankie", "Oakley", "Justice")
ssa_years <- c(rep(c(2009, 2012), 3), 2012)
ssa_df <- tibble(first_names = ssa_names,
                 last_names = LETTERS[1:7],
                 years = ssa_years,
                 min_years = ssa_years - 3,
```

```r
                  max_years = ssa_years + 3)

ssa_df
```

```
## # A tibble: 7 x 5
##   first_names last_names years min_years max_years
##   <chr>       <chr>      <dbl>     <dbl>     <dbl>
## 1 Charlie     A           2009      2006      2012
## 2 Royal       B           2012      2009      2015
## 3 Morgan      C           2009      2006      2012
## 4 Skyler      D           2012      2009      2015
## 5 Frankie     E           2009      2006      2012
## 6 Oakley      F           2012      2009      2015
## 7 Justice     G           2012      2009      2015
```

This dataset connects first names to years but there are columns for minimum and maximum years for possible age range since birth dates are not always exact. We pass this to gender_df() function, which assigns the method that we wish to use and the names of the columns that contain the names and the birth years. The result is a tibble of predictions.

```r
results <- gender_df(ssa_df, name_col = "first_names", year_col = "years",
                     method = "ssa")
```

```
## Warning: The `.dots` argument of `group_by()` is deprecated as of dplyr 1.0.0.
```

```r
results
```

```
## # A tibble: 7 x 6
##   name    proportion_male proportion_female gender year_min year_max
##   <chr>             <dbl>             <dbl> <chr>     <dbl>    <dbl>
## 1 Charlie           0.704             0.296 male       2009     2009
## 2 Frankie           0.702             0.298 male       2009     2009
## 3 Morgan            0.106             0.894 female     2009     2009
## 4 Justice           0.444             0.556 female     2012     2012
## 5 Oakley            0.539             0.461 male       2012     2012
## 6 Royal             0.852             0.148 male       2012     2012
## 7 Skyler            0.576             0.424 male       2012     2012
```

```r
ssa_df %>%
  left_join(results, by = c("first_names" = "name", "years" = "year_min"))
```

```
## # A tibble: 7 x 9
##   first_names last_names years min_years max_years proportion_male
##   <chr>       <chr>      <dbl>     <dbl>     <dbl>           <dbl>
## 1 Charlie     A           2009      2006      2012           0.704
## 2 Royal       B           2012      2009      2015           0.852
## 3 Morgan      C           2009      2006      2012           0.106
## 4 Skyler      D           2012      2009      2015           0.576
## 5 Frankie     E           2009      2006      2012           0.702
## 6 Oakley      F           2012      2009      2015           0.539
## 7 Justice     G           2012      2009      2015           0.444
## # ... with 3 more variables: proportion_female <dbl>, gender <chr>,
## #   year_max <dbl>
```

```r
gender_df(ssa_df, name_col = "first_names",
          year_col = c("min_years", "max_years"), method = "ssa")
```

```
## Warning in gender(.[[name_col]], years = c(.[[year_col[1]]][1], .[[year_col[2]]]
## [1]), : The year range provided has been trimmed to fit within 1880 to 2012.

## # A tibble: 7 x 6
##   name    proportion_male proportion_female gender year_min year_max
##   <chr>             <dbl>             <dbl> <chr>     <dbl>    <dbl>
## 1 Charlie           0.658             0.342 male       2006     2012
## 2 Frankie           0.685             0.315 male       2006     2012
## 3 Morgan            0.106             0.894 female     2006     2012
## 4 Justice           0.484             0.516 female     2009     2012
## 5 Oakley            0.541             0.459 male       2009     2012
## 6 Royal             0.837             0.163 male       2009     2012
## 7 Skyler            0.621             0.379 male       2009     2012
```

Now, we use gender_df() to predict gender by passing it the columns minimum and maximum years to be used for each name

```r
ssa_df %>%
  left_join(results, by = c("first_names" = "name", "years" = "year_min"))
```

```
## # A tibble: 7 x 9
##   first_names last_names years min_years max_years proportion_male
##   <chr>       <chr>      <dbl>     <dbl>     <dbl>           <dbl>
## 1 Charlie     A           2009      2006      2012           0.704
## 2 Royal       B           2012      2009      2015           0.852
## 3 Morgan      C           2009      2006      2012           0.106
## 4 Skyler      D           2012      2009      2015           0.576
## 5 Frankie     E           2009      2006      2012           0.702
## 6 Oakley      F           2012      2009      2015           0.539
## 7 Justice     G           2012      2009      2015           0.444
## # ... with 3 more variables: proportion_female <dbl>, gender <chr>,
## #   year_max <dbl>
```

```r
gender_df(ssa_df, name_col = "first_names",
          year_col = c("min_years", "max_years"), method = "ssa")
```

```
## Warning in gender(.[[name_col]], years = c(.[[year_col[1]]][1], .[[year_col[2]]]
## [1]), : The year range provided has been trimmed to fit within 1880 to 2012.

## # A tibble: 7 x 6
##   name    proportion_male proportion_female gender year_min year_max
##   <chr>             <dbl>             <dbl> <chr>     <dbl>    <dbl>
## 1 Charlie           0.658             0.342 male       2006     2012
## 2 Frankie           0.685             0.315 male       2006     2012
## 3 Morgan            0.106             0.894 female     2006     2012
## 4 Justice           0.484             0.516 female     2009     2012
## 5 Oakley            0.541             0.459 male       2009     2012
## 6 Royal             0.837             0.163 male       2009     2012
## 7 Skyler            0.621             0.379 male       2009     2012
```

```r
ssa_df %>%
  distinct(first_names, years) %>%
  rowwise() %>%
  do(results = gender(.$first_names, years = .$years, method = "ssa")) %>%
  do(bind_rows(.$results))
```

```
## # A tibble: 7 x 6
## # Rowwise:
```

```
##   name    proportion_male proportion_female gender year_min year_max
##   <chr>             <dbl>             <dbl> <chr>     <dbl>    <dbl>
## 1 Charlie           0.704             0.296 male       2009     2009
## 2 Royal             0.852             0.148 male       2012     2012
## 3 Morgan            0.106             0.894 female     2009     2009
## 4 Skyler            0.576             0.424 male       2012     2012
## 5 Frankie           0.702             0.298 male       2009     2009
## 6 Oakley            0.539             0.461 male       2012     2012
## 7 Justice           0.444             0.556 female     2012     2012
```

```
ssa_df %>%
  distinct(first_names, years) %>%
  group_by(years) %>%
  do(results = gender(.$first_names, years = .$years[1], method = "ssa")) %>%
  do(bind_rows(.$results))
```

```
## # A tibble: 7 x 6
## # Rowwise:
##   name    proportion_male proportion_female gender year_min year_max
##   <chr>             <dbl>             <dbl> <chr>     <dbl>    <dbl>
## 1 Charlie           0.704             0.296 male       2009     2009
## 2 Frankie           0.702             0.298 male       2009     2009
## 3 Morgan            0.106             0.894 female     2009     2009
## 4 Justice           0.444             0.556 female     2012     2012
## 5 Oakley            0.539             0.461 male       2012     2012
## 6 Royal             0.852             0.148 male       2012     2012
## 7 Skyler            0.576             0.424 male       2012     2012
```

Logistic Regression Model

```
neutral_names <- babynames %>%
  select(-prop) %>%
  #filter only names between years 1930 and 2012
  filter(year >= 1930, year <= 2012)  %>%
  #get the number of female and male for each name per year
  spread(key = sex, value = n, fill = 0)   %>%
  #Calculate the measure of gender-neutrality
  mutate(prop_F = 100 * F / (F+M), se = (50 - prop_F)^2) %>%
  group_by(name) %>%
  #per name, find the total number of babies and measure of gender-neutrality
  dplyr::summarise(n = n(), female = sum(F), male=sum(M), total = sum(F + M),
          mse = mean(se)) %>%
  #take only names that occurs every year and occurs greater than 9000 times
  filter(n == 83, total > 9000) %>%
  #sort by gender neutrality
  arrange(mse) %>%
  #get only the top 10
  head(10)

neutral_names
```

```
## # A tibble: 10 x 6
##    name        n female  male  total   mse
##    <chr>   <int>  <dbl> <dbl>  <dbl> <dbl>
```

```
##   1 Jessie     83  73714 76122 149836  99.2
##   2 Marion     83  73240 40298 113538 122.
##   3 Jackie     83  86418 74870 161288 167.
##   4 Unknown    83   8814  8366  17180 167.
##   5 Alva       83   4304  4703   9007 265.
##   6 Ollie      83  12743  7769  20512 266.
##   7 Jody       83  55538 30814  86352 303.
##   8 Cleo       83  10327  5330  15657 400.
##   9 Ivory      83   6515  6427  12942 457.
##  10 Kerry      83  48356 49198  97554 463.
```

Random Forest Classification

```r
library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:psych':
##
##     outlier

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
neutral_names <- babynames %>%
  select(-prop) %>%
  #Filter only names between years 1930 and 2012
  filter(year >= 1930, year <= 2012) %>%
  #Get the number of female and male for each name per year
  spread(key = sex, value = n, fill = 0) %>%
  #Calculate the measure of gender-neutrality
  mutate(prop_F = 100 * F / (F+M), se = (50 - prop_F)^2) %>%
  group_by(name) %>%
  #Find the total number of babies and measure of gender-neutrality per name
  dplyr::summarise(n = n(), female = sum(F), male=sum(M), total = sum(F + M),
          mse = mean(se)) %>%
  #Take only names that occurs every year and occurs greater than 9000 times
  filter(n == 83, total > 9000) %>%
  #Sort by gender neutrality
  arrange(mse) %>%
  #Add variable to represent gender neutral namse. Assumes an mse <= 2000
  mutate(isNeutral = ifelse(mse <= 2000,1,0))
```

```r
neutral_names$isNeutral <- as.factor(neutral_names$isNeutral)

set.seed(100)
train <- sample(nrow(neutral_names), 0.7*nrow(neutral_names), replace = FALSE)
TrainSet <- neutral_names[train,]
ValidSet <- neutral_names[-train,]
summary(TrainSet)
```

```
##      name                 n           female              male        
##  Length:1037        Min.   :83   Min.   :      0   Min.   :      0  
##  Class :character   1st Qu.:83   1st Qu.:    531   1st Qu.:    199  
##  Mode  :character   Median :83   Median :  10908   Median :   9245  
##                     Mean   :83   Mean   :  73563   Mean   :  89364  
##                     3rd Qu.:83   3rd Qu.:  48031   3rd Qu.:  48099  
##                     Max.   :83   Max.   :2545718   Max.   :4121292  
##      total              mse           isNeutral
##  Min.   :   9034   Min.   :  99.19   0:920    
##  1st Qu.:  18902   1st Qu.:2410.96   1:117    
##  Median :  46987   Median :2467.45            
##  Mean   : 162927   Mean   :2322.93            
##  3rd Qu.: 151072   3rd Qu.:2488.15            
##  Max.   :4138958   Max.   :2500.00            
```

```r
summary(ValidSet)
```

```
##      name                 n           female             male        
##  Length:445         Min.   :83   Min.   :     0   Min.   :      0  
##  Class :character   1st Qu.:83   1st Qu.:   458   1st Qu.:    194  
##  Mode  :character   Median :83   Median : 10674   Median :   9376  
##                     Mean   :83   Mean   : 49556   Mean   :  74116  
##                     3rd Qu.:83   3rd Qu.: 45849   3rd Qu.:  44825  
##                     Max.   :83   Max.   :604343   Max.   :4197382  
##      total              mse          isNeutral
##  Min.   :   9007   Min.   : 265.3   0:389    
##  1st Qu.:  19793   1st Qu.:2409.6   1: 56    
##  Median :  46070   Median :2469.6            
##  Mean   : 123672   Mean   :2308.8            
##  3rd Qu.: 125822   3rd Qu.:2490.5            
##  Max.   :4218445   Max.   :2500.0            
```

```r
model1 <- randomForest(isNeutral ~ ., data = TrainSet, importance = TRUE)
model1
```

```
## 
## Call:
##  randomForest(formula = isNeutral ~ ., data = TrainSet, importance = TRUE) 
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
## 
##         OOB estimate of  error rate: 0%
## Confusion matrix:
##     0   1 class.error
## 0 920   0           0
## 1   0 117           0
```

```r
predTrain <- predict(model1, TrainSet, type = "class")
caret::confusionMatrix(predTrain, TrainSet$isNeutral)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 920   0
##          1   0 117
##
##                Accuracy : 1
##                  95% CI : (0.9964, 1)
##     No Information Rate : 0.8872
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.8872
##          Detection Rate : 0.8872
##    Detection Prevalence : 0.8872
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

Train data accuracy is 100% that indicates all the values classified correctly.

Predicting on test data

```r
predTest <- predict(model1, ValidSet, type = "class")
caret::confusionMatrix(predTest, ValidSet$isNeutral)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 389   0
##          1   0  56
##
##                Accuracy : 1
##                  95% CI : (0.9917, 1)
##     No Information Rate : 0.8742
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
```

```
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.8742
##          Detection Rate : 0.8742
##    Detection Prevalence : 0.8742
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

Validation data accuracy is 100% that indicates all the values classified correctly.

Naive Bayes Classification

Comparing model 1 of Random Forest with Naive Bayes model and prediction using naive bayes on training data

```r
model <- naive_bayes(isNeutral ~ ., data = TrainSet, usekernel = T)
```

```
## Warning: naive_bayes(): Feature name - zero probabilities are present. Consider
## Laplace smoothing.
```

```r
model
```

```
##
## ================================== Naive Bayes ==================================
##
##  Call:
## naive_bayes.formula(formula = isNeutral ~ ., data = TrainSet,
##     usekernel = T)
##
## ---------------------------------------------------------------------------------
##
## Laplace smoothing: 0
##
## ---------------------------------------------------------------------------------
##
##  A priori probabilities:
##
##         0         1
## 0.8871745 0.1128255
##
## ---------------------------------------------------------------------------------
##
##  Tables:
##
## ---------------------------------------------------------------------------------
##  ::: name (Categorical)
## ---------------------------------------------------------------------------------
##
## name                     0           1
##   Aaron       0.001086957 0.000000000
##   Abby        0.001086957 0.000000000
##   Abel        0.001086957 0.000000000
##   Abigail     0.001086957 0.000000000
```

```
## Abraham 0.001086957 0.000000000
## Abram 0.001086957 0.000000000
## Ada 0.001086957 0.000000000
## Adam 0.001086957 0.000000000
## Adelaide 0.001086957 0.000000000
## Adele 0.001086957 0.000000000
## Adeline 0.001086957 0.000000000
## Adrian 0.000000000 0.008547009
## Adriana 0.001086957 0.000000000
## Adrianne 0.001086957 0.000000000
## Agustin 0.001086957 0.000000000
## Aida 0.001086957 0.000000000
## Aileen 0.001086957 0.000000000
## Aimee 0.001086957 0.000000000
## Al 0.001086957 0.000000000
## Alberta 0.001086957 0.000000000
## Alberto 0.001086957 0.000000000
## Aldo 0.001086957 0.000000000
## Alec 0.001086957 0.000000000
## Alejandra 0.001086957 0.000000000
## Alejandro 0.001086957 0.000000000
## Alex 0.001086957 0.000000000
## Alexander 0.001086957 0.000000000
## Alfonso 0.001086957 0.000000000
## Alice 0.001086957 0.000000000
## Alicia 0.001086957 0.000000000
## Allan 0.001086957 0.000000000
## Allen 0.001086957 0.000000000
## Alma 0.001086957 0.000000000
## Alonzo 0.001086957 0.000000000
## Alphonso 0.001086957 0.000000000
## Althea 0.001086957 0.000000000
## Alton 0.001086957 0.000000000
## Alyce 0.001086957 0.000000000
## Amanda 0.001086957 0.000000000
## Amber 0.001086957 0.000000000
## Amelia 0.001086957 0.000000000
## America 0.001086957 0.000000000
## Amie 0.001086957 0.000000000
## Amos 0.001086957 0.000000000
## Amy 0.001086957 0.000000000
## Ana 0.001086957 0.000000000
## Anastasia 0.001086957 0.000000000
## Anderson 0.001086957 0.000000000
## Andre 0.001086957 0.000000000
## Andres 0.001086957 0.000000000
## Andy 0.001086957 0.000000000
## Angel 0.000000000 0.008547009
## Angela 0.001086957 0.000000000
## Angelica 0.001086957 0.000000000
## Angelina 0.001086957 0.000000000
## Angelita 0.001086957 0.000000000
## Angie 0.001086957 0.000000000
## Anita 0.001086957 0.000000000
```

```
##    Ann          0.001086957 0.000000000
##    Anna         0.001086957 0.000000000
##    Annabelle    0.001086957 0.000000000
##    Annamarie    0.001086957 0.000000000
##    Anne         0.001086957 0.000000000
##    Annette      0.001086957 0.000000000
##    Annie        0.001086957 0.000000000
##    Antonio      0.001086957 0.000000000
##    Antony       0.001086957 0.000000000
##    April        0.001086957 0.000000000
##    Archie       0.001086957 0.000000000
##    Armand       0.001086957 0.000000000
##    Arnold       0.001086957 0.000000000
##    Arthur       0.001086957 0.000000000
##    Asher        0.001086957 0.000000000
##    Ashley       0.000000000 0.008547009
##    Ashton       0.000000000 0.008547009
##    Athena       0.001086957 0.000000000
##    Aubrey       0.000000000 0.008547009
##    Audra        0.001086957 0.000000000
##    Augustine    0.000000000 0.008547009
##    Aurora       0.001086957 0.000000000
##    Austin       0.001086957 0.000000000
##    Ava          0.001086957 0.000000000
##    Avery        0.000000000 0.008547009
##    Bailey       0.000000000 0.008547009
##    Barbara      0.001086957 0.000000000
##    Barrett      0.001086957 0.000000000
##    Beatrice     0.001086957 0.000000000
##    Beatriz      0.001086957 0.000000000
##    Becky        0.001086957 0.000000000
##    Belinda      0.001086957 0.000000000
##    Bella        0.001086957 0.000000000
##    Ben          0.001086957 0.000000000
##    Benita       0.001086957 0.000000000
##    Benito       0.001086957 0.000000000
##    Bennie       0.000000000 0.008547009
##    Benny        0.001086957 0.000000000
##    Bentley      0.000000000 0.008547009
##    Bernadette   0.001086957 0.000000000
##    Bernice      0.001086957 0.000000000
##    Bert         0.001086957 0.000000000
##    Beth         0.001086957 0.000000000
##    Bethany      0.001086957 0.000000000
##    Betsy        0.001086957 0.000000000
##    Betty        0.001086957 0.000000000
##    Beverly      0.001086957 0.000000000
##    Blake        0.001086957 0.000000000
##    Blanca       0.001086957 0.000000000
##    Blanche      0.001086957 0.000000000
##    Bob          0.001086957 0.000000000
##    Bobby        0.001086957 0.000000000
##    Bonnie       0.001086957 0.000000000
##    Boyd         0.001086957 0.000000000
```

```
## Brad        0.001086957 0.000000000
## Brady       0.001086957 0.000000000
## Braxton     0.001086957 0.000000000
## Brent       0.001086957 0.000000000
## Brian       0.001086957 0.000000000
## Brice       0.001086957 0.000000000
## Bridget     0.001086957 0.000000000
## Bruce       0.001086957 0.000000000
## Bryan       0.001086957 0.000000000
## Buddy       0.001086957 0.000000000
## Burton      0.001086957 0.000000000
## Byron       0.001086957 0.000000000
## Caleb       0.001086957 0.000000000
## Callie      0.001086957 0.000000000
## Calvin      0.001086957 0.000000000
## Camilla     0.001086957 0.000000000
## Camille     0.001086957 0.000000000
## Cara        0.001086957 0.000000000
## Carey       0.000000000 0.008547009
## Carl        0.001086957 0.000000000
## Carlee      0.000000000 0.008547009
## Carlene     0.001086957 0.000000000
## Carlie      0.000000000 0.008547009
## Carlo       0.001086957 0.000000000
## Carmela     0.001086957 0.000000000
## Carmella    0.001086957 0.000000000
## Carmen      0.000000000 0.008547009
## Carol       0.001086957 0.000000000
## Carolina    0.001086957 0.000000000
## Caroline    0.001086957 0.000000000
## Carolyn     0.001086957 0.000000000
## Carrie      0.001086957 0.000000000
## Carter      0.001086957 0.000000000
## Casey       0.000000000 0.008547009
## Cassandra   0.001086957 0.000000000
## Cassie      0.001086957 0.000000000
## Cathleen    0.001086957 0.000000000
## Cathryn     0.001086957 0.000000000
## Cathy       0.001086957 0.000000000
## Cecil       0.001086957 0.000000000
## Cecile      0.001086957 0.000000000
## Cecilia     0.001086957 0.000000000
## Cedric      0.001086957 0.000000000
## Celia       0.001086957 0.000000000
## Celina      0.001086957 0.000000000
## Cesar       0.001086957 0.000000000
## Chad        0.001086957 0.000000000
## Chandler    0.000000000 0.008547009
## Charlene    0.001086957 0.000000000
## Charles     0.001086957 0.000000000
## Charley     0.000000000 0.008547009
## Charlie     0.000000000 0.008547009
## Charmaine   0.001086957 0.000000000
## Cheri       0.001086957 0.000000000
```

```
## Cherry       0.001086957 0.000000000
## Cheryl       0.001086957 0.000000000
## Chester      0.001086957 0.000000000
## Chloe        0.001086957 0.000000000
## Chris        0.000000000 0.008547009
## Christa      0.001086957 0.000000000
## Christian    0.000000000 0.008547009
## Christie     0.001086957 0.000000000
## Christina    0.001086957 0.000000000
## Christine    0.001086957 0.000000000
## Christopher 0.001086957 0.000000000
## Christy      0.000000000 0.008547009
## Chrystal     0.001086957 0.000000000
## Claire       0.001086957 0.000000000
## Clara        0.001086957 0.000000000
## Clare        0.000000000 0.008547009
## Clarence     0.001086957 0.000000000
## Clarice      0.001086957 0.000000000
## Clarissa     0.001086957 0.000000000
## Claudette    0.001086957 0.000000000
## Claudia      0.001086957 0.000000000
## Clay         0.001086957 0.000000000
## Clifford     0.001086957 0.000000000
## Clifton      0.001086957 0.000000000
## Clint        0.001086957 0.000000000
## Clinton      0.001086957 0.000000000
## Clyde        0.001086957 0.000000000
## Cody         0.001086957 0.000000000
## Colby        0.000000000 0.008547009
## Colette      0.001086957 0.000000000
## Colin        0.001086957 0.000000000
## Collin       0.001086957 0.000000000
## Connie       0.001086957 0.000000000
## Conrad       0.001086957 0.000000000
## Consuelo     0.001086957 0.000000000
## Cora         0.001086957 0.000000000
## Corbin       0.001086957 0.000000000
## Cordell      0.001086957 0.000000000
## Corina       0.001086957 0.000000000
## Corinne      0.001086957 0.000000000
## Cornelia     0.001086957 0.000000000
## Cornelius    0.001086957 0.000000000
## Corrine      0.001086957 0.000000000
## Coy          0.001086957 0.000000000
## Craig        0.001086957 0.000000000
## Cristina     0.001086957 0.000000000
## Crystal      0.001086957 0.000000000
## Cullen       0.001086957 0.000000000
## Curt         0.001086957 0.000000000
## Curtis       0.001086957 0.000000000
## Cynthia      0.001086957 0.000000000
## Daisy        0.001086957 0.000000000
## Dallas       0.000000000 0.008547009
## Dalton       0.001086957 0.000000000
```

```
## 	Damian 	0.001086957 0.000000000
## 	Damon 	0.001086957 0.000000000
## 	Dan 	0.001086957 0.000000000
## 	Daniel 	0.001086957 0.000000000
## 	Danna 	0.001086957 0.000000000
## 	Danny 	0.001086957 0.000000000
## 	Dante 	0.001086957 0.000000000
## 	Daphne 	0.001086957 0.000000000
## 	Darius 	0.001086957 0.000000000
## 	Darlene 	0.001086957 0.000000000
## 	Darnell 	0.000000000 0.008547009
## 	Darrel 	0.001086957 0.000000000
## 	Darrell 	0.001086957 0.000000000
## 	Darryl 	0.001086957 0.000000000
## 	Daryl 	0.000000000 0.008547009
## 	David 	0.001086957 0.000000000
## 	Davis 	0.001086957 0.000000000
## 	Dawson 	0.001086957 0.000000000
## 	Dayton 	0.001086957 0.000000000
## 	Dean 	0.001086957 0.000000000
## 	Deanna 	0.001086957 0.000000000
## 	Deanne 	0.001086957 0.000000000
## 	Debbie 	0.001086957 0.000000000
## 	Deborah 	0.001086957 0.000000000
## 	Debra 	0.001086957 0.000000000
## 	Delia 	0.001086957 0.000000000
## 	Delilah 	0.001086957 0.000000000
## 	Della 	0.001086957 0.000000000
## 	Denice 	0.001086957 0.000000000
## 	Denis 	0.000000000 0.008547009
## 	Dennis 	0.001086957 0.000000000
## 	Denny 	0.001086957 0.000000000
## 	Denver 	0.001086957 0.000000000
## 	Deon 	0.000000000 0.008547009
## 	Desmond 	0.001086957 0.000000000
## 	Devon 	0.000000000 0.008547009
## 	Dewayne 	0.001086957 0.000000000
## 	Dianna 	0.001086957 0.000000000
## 	Dianne 	0.001086957 0.000000000
## 	Dion 	0.000000000 0.008547009
## 	Dirk 	0.001086957 0.000000000
## 	Dolly 	0.001086957 0.000000000
## 	Dolores 	0.001086957 0.000000000
## 	Domingo 	0.001086957 0.000000000
## 	Dominic 	0.001086957 0.000000000
## 	Donald 	0.001086957 0.000000000
## 	Donna 	0.001086957 0.000000000
## 	Donnell 	0.000000000 0.008547009
## 	Donnie 	0.000000000 0.008547009
## 	Donny 	0.001086957 0.000000000
## 	Doreen 	0.001086957 0.000000000
## 	Doris 	0.001086957 0.000000000
## 	Dorothea 	0.001086957 0.000000000
## 	Dorothy 	0.001086957 0.000000000
```

```
##    Dorthy      0.001086957 0.000000000
##    Douglas     0.001086957 0.000000000
##    Doyle       0.001086957 0.000000000
##    Duane       0.001086957 0.000000000
##    Dwight      0.001086957 0.000000000
##    Earl        0.001086957 0.000000000
##    Earnest     0.001086957 0.000000000
##    Eddie       0.001086957 0.000000000
##    Eddy        0.001086957 0.000000000
##    Edgar       0.001086957 0.000000000
##    Edith       0.001086957 0.000000000
##    Edmond      0.001086957 0.000000000
##    Edmund      0.001086957 0.000000000
##    Edna        0.001086957 0.000000000
##    Eduardo     0.001086957 0.000000000
##    Edward      0.001086957 0.000000000
##    Edwina      0.001086957 0.000000000
##    Efrain      0.001086957 0.000000000
##    Eileen      0.001086957 0.000000000
##    Elaine      0.001086957 0.000000000
##    Elbert      0.001086957 0.000000000
##    Eleanor     0.001086957 0.000000000
##    Elena       0.001086957 0.000000000
##    Eli         0.001086957 0.000000000
##    Elias       0.001086957 0.000000000
##    Elijah      0.001086957 0.000000000
##    Elisa       0.001086957 0.000000000
##    Elisabeth   0.001086957 0.000000000
##    Elise       0.001086957 0.000000000
##    Elissa      0.001086957 0.000000000
##    Eliza       0.001086957 0.000000000
##    Elizabeth   0.001086957 0.000000000
##    Ellie       0.000000000 0.008547009
##    Elliot      0.001086957 0.000000000
##    Elmer       0.001086957 0.000000000
##    Elsie       0.001086957 0.000000000
##    Elton       0.001086957 0.000000000
##    Elva        0.001086957 0.000000000
##    Elvira      0.001086957 0.000000000
##    Elvis       0.001086957 0.000000000
##    Elwood      0.001086957 0.000000000
##    Emanuel     0.001086957 0.000000000
##    Emerson     0.001086957 0.000000000
##    Emilie      0.001086957 0.000000000
##    Emilio      0.001086957 0.000000000
##    Emily       0.001086957 0.000000000
##    Emma        0.001086957 0.000000000
##    Emmanuel    0.001086957 0.000000000
##    Emmett      0.001086957 0.000000000
##    Emory       0.000000000 0.008547009
##    Eric        0.001086957 0.000000000
##    Erich       0.001086957 0.000000000
##    Erik        0.001086957 0.000000000
##    Erika       0.001086957 0.000000000
```

```
## 	Erin	0.001086957 0.000000000
## 	Erma	0.001086957 0.000000000
## 	Ernest	0.001086957 0.000000000
## 	Ernie	0.001086957 0.000000000
## 	Ervin	0.001086957 0.000000000
## 	Erwin	0.001086957 0.000000000
## 	Esmeralda	0.001086957 0.000000000
## 	Esperanza	0.001086957 0.000000000
## 	Estella	0.001086957 0.000000000
## 	Estelle	0.001086957 0.000000000
## 	Ester	0.001086957 0.000000000
## 	Esther	0.001086957 0.000000000
## 	Ethan	0.001086957 0.000000000
## 	Etta	0.001086957 0.000000000
## 	Eugene	0.001086957 0.000000000
## 	Eugenia	0.001086957 0.000000000
## 	Eunice	0.001086957 0.000000000
## 	Eva	0.001086957 0.000000000
## 	Evangeline	0.001086957 0.000000000
## 	Evelyn	0.001086957 0.000000000
## 	Everett	0.001086957 0.000000000
## 	Ezekiel	0.001086957 0.000000000
## 	Ezequiel	0.001086957 0.000000000
## 	Ezra	0.001086957 0.000000000
## 	Fabian	0.001086957 0.000000000
## 	Fannie	0.001086957 0.000000000
## 	Fay	0.001086957 0.000000000
## 	Felicia	0.001086957 0.000000000
## 	Felipe	0.001086957 0.000000000
## 	Fern	0.001086957 0.000000000
## 	Fernando	0.001086957 0.000000000
## 	Fidel	0.001086957 0.000000000
## 	Flora	0.001086957 0.000000000
## 	Forrest	0.001086957 0.000000000
## 	Frances	0.001086957 0.000000000
## 	Francis	0.000000000 0.008547009
## 	Frank	0.001086957 0.000000000
## 	Frankie	0.000000000 0.008547009
## 	Franklin	0.001086957 0.000000000
## 	Fred	0.001086957 0.000000000
## 	Freda	0.001086957 0.000000000
## 	Freddie	0.000000000 0.008547009
## 	Frederick	0.001086957 0.000000000
## 	Fredrick	0.001086957 0.000000000
## 	Gabriel	0.001086957 0.000000000
## 	Gabrielle	0.001086957 0.000000000
## 	Gail	0.001086957 0.000000000
## 	Galen	0.001086957 0.000000000
## 	Garland	0.001086957 0.000000000
## 	Garrett	0.001086957 0.000000000
## 	Garry	0.001086957 0.000000000
## 	Gary	0.001086957 0.000000000
## 	Gayle	0.001086957 0.000000000
## 	Gena	0.001086957 0.000000000
```

```
## Gene       0.001086957 0.000000000
## Geneva     0.001086957 0.000000000
## Geoffrey   0.001086957 0.000000000
## George     0.001086957 0.000000000
## Georgette  0.001086957 0.000000000
## Gerald     0.001086957 0.000000000
## Gerard     0.001086957 0.000000000
## Gerardo    0.001086957 0.000000000
## Gerry      0.000000000 0.008547009
## Gilberto   0.001086957 0.000000000
## Ginger     0.001086957 0.000000000
## Gino       0.001086957 0.000000000
## Giovanna   0.001086957 0.000000000
## Gladys     0.001086957 0.000000000
## Glen       0.001086957 0.000000000
## Glenn      0.001086957 0.000000000
## Glenna     0.001086957 0.000000000
## Goldie     0.001086957 0.000000000
## Gordon     0.001086957 0.000000000
## Grace      0.001086957 0.000000000
## Gracie     0.001086957 0.000000000
## Graciela   0.001086957 0.000000000
## Grady      0.001086957 0.000000000
## Greg       0.001086957 0.000000000
## Gregg      0.001086957 0.000000000
## Greta      0.001086957 0.000000000
## Gretchen   0.001086957 0.000000000
## Griffin    0.001086957 0.000000000
## Grover     0.001086957 0.000000000
## Guadalupe  0.000000000 0.008547009
## Guillermo  0.001086957 0.000000000
## Gunnar     0.001086957 0.000000000
## Gus        0.001086957 0.000000000
## Guy        0.001086957 0.000000000
## Gwen       0.001086957 0.000000000
## Gwendolyn  0.001086957 0.000000000
## Hal        0.001086957 0.000000000
## Hallie     0.000000000 0.008547009
## Hanna      0.001086957 0.000000000
## Hannah     0.001086957 0.000000000
## Hans       0.001086957 0.000000000
## Harlan     0.001086957 0.000000000
## Harley     0.000000000 0.008547009
## Harold     0.001086957 0.000000000
## Harriet    0.001086957 0.000000000
## Harris     0.001086957 0.000000000
## Harrison   0.001086957 0.000000000
## Harvey     0.001086957 0.000000000
## Hayden     0.000000000 0.008547009
## Hazel      0.001086957 0.000000000
## Heather    0.001086957 0.000000000
## Hector     0.001086957 0.000000000
## Helen      0.001086957 0.000000000
## Helena     0.001086957 0.000000000
```

```
## Helene 0.001086957 0.000000000
## Herbert 0.001086957 0.000000000
## Heriberto 0.001086957 0.000000000
## Herman 0.001086957 0.000000000
## Hilary 0.000000000 0.008547009
## Hillary 0.000000000 0.008547009
## Hollis 0.000000000 0.008547009
## Holly 0.001086957 0.000000000
## Hope 0.001086957 0.000000000
## Horace 0.001086957 0.000000000
## Houston 0.001086957 0.000000000
## Howard 0.001086957 0.000000000
## Hudson 0.001086957 0.000000000
## Hugh 0.001086957 0.000000000
## Hugo 0.001086957 0.000000000
## Humberto 0.001086957 0.000000000
## Ian 0.001086957 0.000000000
## Ida 0.001086957 0.000000000
## Ina 0.001086957 0.000000000
## Iris 0.001086957 0.000000000
## Irma 0.001086957 0.000000000
## Irving 0.001086957 0.000000000
## Isaac 0.001086957 0.000000000
## Isabel 0.001086957 0.000000000
## Isabella 0.001086957 0.000000000
## Isaias 0.001086957 0.000000000
## Isiah 0.001086957 0.000000000
## Ismael 0.001086957 0.000000000
## Issac 0.001086957 0.000000000
## Iva 0.001086957 0.000000000
## Ivy 0.000000000 0.008547009
## Jack 0.001086957 0.000000000
## Jackie 0.000000000 0.008547009
## Jackson 0.001086957 0.000000000
## Jaclyn 0.001086957 0.000000000
## Jacqueline 0.001086957 0.000000000
## Jacquelyn 0.001086957 0.000000000
## Jaime 0.000000000 0.008547009
## Jake 0.001086957 0.000000000
## James 0.001086957 0.000000000
## Jamie 0.000000000 0.008547009
## Jan 0.000000000 0.008547009
## Jane 0.001086957 0.000000000
## Janell 0.001086957 0.000000000
## Janelle 0.001086957 0.000000000
## Janet 0.001086957 0.000000000
## Janette 0.001086957 0.000000000
## Janie 0.001086957 0.000000000
## Janis 0.001086957 0.000000000
## Jaqueline 0.001086957 0.000000000
## Jared 0.001086957 0.000000000
## Jarrett 0.001086957 0.000000000
## Jarvis 0.001086957 0.000000000
## Jason 0.001086957 0.000000000
```

```
##    Jasper      0.001086957 0.000000000
##    Jay         0.001086957 0.000000000
##    Jayne       0.001086957 0.000000000
##    Jean        0.000000000 0.008547009
##    Jeanette    0.001086957 0.000000000
##    Jeanine     0.001086957 0.000000000
##    Jeanne      0.001086957 0.000000000
##    Jeannie     0.001086957 0.000000000
##    Jeff        0.001086957 0.000000000
##    Jeffery     0.001086957 0.000000000
##    Jeffrey     0.001086957 0.000000000
##    Jenna       0.001086957 0.000000000
##    Jennifer    0.001086957 0.000000000
##    Jenny       0.001086957 0.000000000
##    Jerald      0.001086957 0.000000000
##    Jeremy      0.001086957 0.000000000
##    Jeri        0.001086957 0.000000000
##    Jerome      0.001086957 0.000000000
##    Jerri       0.001086957 0.000000000
##    Jerry       0.001086957 0.000000000
##    Jesse       0.001086957 0.000000000
##    Jessica     0.001086957 0.000000000
##    Jessie      0.000000000 0.008547009
##    Jesus       0.001086957 0.000000000
##    Jewel       0.000000000 0.008547009
##    Jill        0.001086957 0.000000000
##    Jim         0.001086957 0.000000000
##    Jimmy       0.001086957 0.000000000
##    Jo          0.000000000 0.008547009
##    Joanna      0.001086957 0.000000000
##    Joanne      0.001086957 0.000000000
##    Joaquin     0.001086957 0.000000000
##    Jocelyn     0.001086957 0.000000000
##    Jodie       0.000000000 0.008547009
##    Joel        0.001086957 0.000000000
##    Joesph      0.001086957 0.000000000
##    Joey        0.000000000 0.008547009
##    Johanna     0.001086957 0.000000000
##    John        0.001086957 0.000000000
##    Johnathan   0.001086957 0.000000000
##    Johnie      0.000000000 0.008547009
##    Johnny      0.001086957 0.000000000
##    Jolene      0.001086957 0.000000000
##    Jon         0.001086957 0.000000000
##    Jonah       0.001086957 0.000000000
##    Jonas       0.001086957 0.000000000
##    Jonathan    0.001086957 0.000000000
##    Jordan      0.000000000 0.008547009
##    Jorge       0.001086957 0.000000000
##    Josh        0.001086957 0.000000000
##    Joshua      0.001086957 0.000000000
##    Joyce       0.001086957 0.000000000
##    Juan        0.001086957 0.000000000
##    Juana       0.001086957 0.000000000
```

```
##    Jude         0.000000000 0.008547009
##    Julia        0.001086957 0.000000000
##    Julian       0.001086957 0.000000000
##    Juliana      0.001086957 0.000000000
##    Julie        0.001086957 0.000000000
##    Juliet       0.001086957 0.000000000
##    Juliette     0.001086957 0.000000000
##    Julius       0.001086957 0.000000000
##    June         0.001086957 0.000000000
##    Justin       0.001086957 0.000000000
##    Karen        0.001086957 0.000000000
##    Karl         0.001086957 0.000000000
##    Karla        0.001086957 0.000000000
##    Karol        0.000000000 0.008547009
##    Katharine    0.001086957 0.000000000
##    Katherine    0.001086957 0.000000000
##    Katheryn     0.001086957 0.000000000
##    Kathie       0.001086957 0.000000000
##    Kathleen     0.001086957 0.000000000
##    Kathrine     0.001086957 0.000000000
##    Kathy        0.001086957 0.000000000
##    Katie        0.001086957 0.000000000
##    Kay          0.001086957 0.000000000
##    Kaye         0.001086957 0.000000000
##    Kelley       0.000000000 0.008547009
##    Kelly        0.000000000 0.008547009
##    Kelsey       0.000000000 0.008547009
##    Kendall      0.000000000 0.008547009
##    Kenneth      0.001086957 0.000000000
##    Kenny        0.001086957 0.000000000
##    Kent         0.001086957 0.000000000
##    Kermit       0.001086957 0.000000000
##    Kerry        0.000000000 0.008547009
##    Kim          0.000000000 0.008547009
##    Kirby        0.000000000 0.008547009
##    Kurt         0.001086957 0.000000000
##    Kyle         0.001086957 0.000000000
##    Lacey        0.000000000 0.008547009
##    Ladonna      0.001086957 0.000000000
##    Lamar        0.001086957 0.000000000
##    Lamont       0.001086957 0.000000000
##    Lana         0.001086957 0.000000000
##    Lane         0.000000000 0.008547009
##    Lara         0.001086957 0.000000000
##    Larry        0.001086957 0.000000000
##    Laura        0.001086957 0.000000000
##    Laurel       0.001086957 0.000000000
##    Laurence     0.001086957 0.000000000
##    Lavern       0.000000000 0.008547009
##    Layne        0.000000000 0.008547009
##    Leann        0.001086957 0.000000000
##    Leanna       0.001086957 0.000000000
##    Leanne       0.001086957 0.000000000
##    Lee          0.000000000 0.008547009
```

```
##   Leila        0.001086957 0.000000000
##   Leilani      0.001086957 0.000000000
##   Lela         0.001086957 0.000000000
##   Leland       0.001086957 0.000000000
##   Lenora       0.001086957 0.000000000
##   Lenore       0.001086957 0.000000000
##   Leonard      0.001086957 0.000000000
##   Leonardo     0.001086957 0.000000000
##   Leonel       0.001086957 0.000000000
##   Leroy        0.001086957 0.000000000
##   Lesley       0.000000000 0.008547009
##   Leslie       0.000000000 0.008547009
##   Lester       0.001086957 0.000000000
##   Leticia      0.001086957 0.000000000
##   Levi         0.001086957 0.000000000
##   Lewis        0.001086957 0.000000000
##   Lexie        0.000000000 0.008547009
##   Libby        0.001086957 0.000000000
##   Lila         0.001086957 0.000000000
##   Lilia        0.001086957 0.000000000
##   Lilian       0.001086957 0.000000000
##   Lilly        0.001086957 0.000000000
##   Lily         0.001086957 0.000000000
##   Lina         0.001086957 0.000000000
##   Linda        0.001086957 0.000000000
##   Lindsay      0.000000000 0.008547009
##   Lindsey      0.000000000 0.008547009
##   Linwood      0.001086957 0.000000000
##   Lionel       0.001086957 0.000000000
##   Lisa         0.001086957 0.000000000
##   Liza         0.001086957 0.000000000
##   Lizzie       0.001086957 0.000000000
##   Lloyd        0.001086957 0.000000000
##   Logan        0.001086957 0.000000000
##   Lois         0.001086957 0.000000000
##   Lola         0.001086957 0.000000000
##   Lonnie       0.000000000 0.008547009
##   Lora         0.001086957 0.000000000
##   Lorelei      0.001086957 0.000000000
##   Loren        0.000000000 0.008547009
##   Lorena       0.001086957 0.000000000
##   Lorene       0.001086957 0.000000000
##   Lorenzo      0.001086957 0.000000000
##   Lorna        0.001086957 0.000000000
##   Lottie       0.001086957 0.000000000
##   Louie        0.001086957 0.000000000
##   Louis        0.001086957 0.000000000
##   Louise       0.001086957 0.000000000
##   Lowell       0.001086957 0.000000000
##   Luann        0.001086957 0.000000000
##   Lucas        0.001086957 0.000000000
##   Lucia        0.001086957 0.000000000
##   Lucinda      0.001086957 0.000000000
##   Lucy         0.001086957 0.000000000
```

```
##    Luis         0.001086957 0.000000000
##    Luke         0.001086957 0.000000000
##    Lupe         0.000000000 0.008547009
##    Lydia        0.001086957 0.000000000
##    Lyla         0.001086957 0.000000000
##    Lyle         0.001086957 0.000000000
##    Lynette      0.001086957 0.000000000
##    Lynne        0.001086957 0.000000000
##    Lynnette     0.001086957 0.000000000
##    Mabel        0.001086957 0.000000000
##    Mable        0.001086957 0.000000000
##    Mack         0.001086957 0.000000000
##    Macy         0.000000000 0.008547009
##    Madeleine    0.001086957 0.000000000
##    Madelyn      0.001086957 0.000000000
##    Mae          0.001086957 0.000000000
##    Malachi      0.001086957 0.000000000
##    Malcolm      0.001086957 0.000000000
##    Malinda      0.001086957 0.000000000
##    Mamie        0.001086957 0.000000000
##    Manuel       0.001086957 0.000000000
##    Mara         0.001086957 0.000000000
##    Marc         0.001086957 0.000000000
##    Marcel       0.000000000 0.008547009
##    Marcella     0.001086957 0.000000000
##    Marco        0.001086957 0.000000000
##    Marcos       0.001086957 0.000000000
##    Marcus       0.001086957 0.000000000
##    Marcy        0.001086957 0.000000000
##    Margaret     0.001086957 0.000000000
##    Margarita    0.001086957 0.000000000
##    Margie       0.001086957 0.000000000
##    Mari         0.001086957 0.000000000
##    Maria        0.001086957 0.000000000
##    Mariah       0.001086957 0.000000000
##    Mariana      0.001086957 0.000000000
##    Marianne     0.001086957 0.000000000
##    Marilyn      0.001086957 0.000000000
##    Mario        0.001086957 0.000000000
##    Marion       0.000000000 0.008547009
##    Marjorie     0.001086957 0.000000000
##    Mark         0.001086957 0.000000000
##    Marlene      0.001086957 0.000000000
##    Marlon       0.001086957 0.000000000
##    Marquis      0.001086957 0.000000000
##    Marquita     0.001086957 0.000000000
##    Marsha       0.001086957 0.000000000
##    Marshall     0.001086957 0.000000000
##    Marta        0.001086957 0.000000000
##    Martin       0.001086957 0.000000000
##    Martina      0.001086957 0.000000000
##    Marvin       0.001086957 0.000000000
##    Mary         0.001086957 0.000000000
##    Maryann      0.001086957 0.000000000
```

```
##    Maryanne    0.001086957 0.000000000
##    Marybeth    0.001086957 0.000000000
##    Maryellen   0.001086957 0.000000000
##    Maryjane    0.001086957 0.000000000
##    Mason       0.001086957 0.000000000
##    Matilda     0.001086957 0.000000000
##    Maura       0.001086957 0.000000000
##    Maureen     0.001086957 0.000000000
##    Maurice     0.001086957 0.000000000
##    Mauricio    0.001086957 0.000000000
##    Mavis       0.001086957 0.000000000
##    Melanie     0.001086957 0.000000000
##    Melissa     0.001086957 0.000000000
##    Melvin      0.001086957 0.000000000
##    Meredith    0.000000000 0.008547009
##    Merle       0.000000000 0.008547009
##    Merlin      0.000000000 0.008547009
##    Merrill     0.000000000 0.008547009
##    Merry       0.001086957 0.000000000
##    Micheal     0.001086957 0.000000000
##    Michel      0.000000000 0.008547009
##    Michelle    0.001086957 0.000000000
##    Mickey      0.000000000 0.008547009
##    Miguel      0.001086957 0.000000000
##    Mildred     0.001086957 0.000000000
##    Millie      0.001086957 0.000000000
##    Milton      0.001086957 0.000000000
##    Miriam      0.001086957 0.000000000
##    Mitchell    0.001086957 0.000000000
##    Moises      0.001086957 0.000000000
##    Mollie      0.001086957 0.000000000
##    Monique     0.001086957 0.000000000
##    Monte       0.001086957 0.000000000
##    Morris      0.001086957 0.000000000
##    Moses       0.001086957 0.000000000
##    Muriel      0.001086957 0.000000000
##    Murray      0.001086957 0.000000000
##    Myles       0.001086957 0.000000000
##    Myra        0.001086957 0.000000000
##    Myrna       0.001086957 0.000000000
##    Nadia       0.001086957 0.000000000
##    Nadine      0.001086957 0.000000000
##    Nancy       0.001086957 0.000000000
##    Natalia     0.001086957 0.000000000
##    Natalie     0.001086957 0.000000000
##    Nathalie    0.001086957 0.000000000
##    Nathanial   0.001086957 0.000000000
##    Neil        0.001086957 0.000000000
##    Nellie      0.001086957 0.000000000
##    Nelson      0.001086957 0.000000000
##    Nicholas    0.001086957 0.000000000
##    Nick        0.001086957 0.000000000
##    Nickolas    0.001086957 0.000000000
##    Nicolas     0.001086957 0.000000000
```

```
##    Nina        0.001086957 0.000000000
##    Nita        0.001086957 0.000000000
##    Noah        0.001086957 0.000000000
##    Noe         0.001086957 0.000000000
##    Nolan       0.001086957 0.000000000
##    Nora        0.001086957 0.000000000
##    Norah       0.001086957 0.000000000
##    Norbert     0.001086957 0.000000000
##    Noreen      0.001086957 0.000000000
##    Norman      0.001086957 0.000000000
##    Norris      0.001086957 0.000000000
##    Nyla        0.001086957 0.000000000
##    Octavio     0.001086957 0.000000000
##    Ola         0.001086957 0.000000000
##    Olga        0.001086957 0.000000000
##    Olive       0.001086957 0.000000000
##    Olivia      0.001086957 0.000000000
##    Ollie       0.000000000 0.008547009
##    Omar        0.001086957 0.000000000
##    Opal        0.001086957 0.000000000
##    Ora         0.000000000 0.008547009
##    Orlando     0.001086957 0.000000000
##    Oscar       0.001086957 0.000000000
##    Otis        0.001086957 0.000000000
##    Otto        0.001086957 0.000000000
##    Owen        0.001086957 0.000000000
##    Paige       0.000000000 0.008547009
##    Pamela      0.001086957 0.000000000
##    Paris       0.000000000 0.008547009
##    Parker      0.001086957 0.000000000
##    Patience    0.001086957 0.000000000
##    Patrice     0.001086957 0.000000000
##    Patricia    0.001086957 0.000000000
##    Patrick     0.001086957 0.000000000
##    Patsy       0.001086957 0.000000000
##    Patty       0.001086957 0.000000000
##    Paul        0.001086957 0.000000000
##    Paula       0.001086957 0.000000000
##    Paulette    0.001086957 0.000000000
##    Paulina     0.001086957 0.000000000
##    Pauline     0.001086957 0.000000000
##    Pearl       0.001086957 0.000000000
##    Pedro       0.001086957 0.000000000
##    Peggy       0.001086957 0.000000000
##    Penelope    0.001086957 0.000000000
##    Penny       0.001086957 0.000000000
##    Perry       0.001086957 0.000000000
##    Pete        0.001086957 0.000000000
##    Peter       0.001086957 0.000000000
##    Peyton      0.000000000 0.008547009
##    Phil        0.001086957 0.000000000
##    Philip      0.001086957 0.000000000
##    Phillip     0.001086957 0.000000000
##    Pierre      0.001086957 0.000000000
```

```
## Polly       0.001086957 0.000000000
## Porter      0.001086957 0.000000000
## Portia      0.001086957 0.000000000
## Precious    0.001086957 0.000000000
## Presley     0.000000000 0.008547009
## Preston     0.001086957 0.000000000
## Prince      0.001086957 0.000000000
## Princess    0.001086957 0.000000000
## Priscilla   0.001086957 0.000000000
## Quentin     0.001086957 0.000000000
## Quincy      0.000000000 0.008547009
## Rachael     0.001086957 0.000000000
## Rafael      0.001086957 0.000000000
## Ramiro      0.001086957 0.000000000
## Ramon       0.001086957 0.000000000
## Randal      0.001086957 0.000000000
## Randall     0.001086957 0.000000000
## Randolph    0.001086957 0.000000000
## Randy       0.001086957 0.000000000
## Raphael     0.001086957 0.000000000
## Raquel      0.001086957 0.000000000
## Ray         0.001086957 0.000000000
## Reba        0.001086957 0.000000000
## Rebeca      0.001086957 0.000000000
## Rebecca     0.001086957 0.000000000
## Rebekah     0.001086957 0.000000000
## Reece       0.001086957 0.000000000
## Reed        0.001086957 0.000000000
## Reese       0.001086957 0.000000000
## Reggie      0.001086957 0.000000000
## Reginald    0.001086957 0.000000000
## Rena        0.001086957 0.000000000
## Renae       0.001086957 0.000000000
## Renee       0.001086957 0.000000000
## Reuben      0.001086957 0.000000000
## Reva        0.001086957 0.000000000
## Rex         0.001086957 0.000000000
## Reynaldo    0.001086957 0.000000000
## Rhea        0.001086957 0.000000000
## Rhoda       0.001086957 0.000000000
## Rhonda      0.001086957 0.000000000
## Ricardo     0.001086957 0.000000000
## Richard     0.001086957 0.000000000
## Rick        0.001086957 0.000000000
## Rickey      0.001086957 0.000000000
## Ricky       0.001086957 0.000000000
## Rigoberto   0.001086957 0.000000000
## Rita        0.001086957 0.000000000
## Robby       0.001086957 0.000000000
## Robert      0.001086957 0.000000000
## Roberto     0.001086957 0.000000000
## Robin       0.000000000 0.008547009
## Rochelle    0.001086957 0.000000000
## Rodger      0.001086957 0.000000000
```

```
## 	Rodney 	0.001086957 0.000000000
## 	Rodolfo 	0.001086957 0.000000000
## 	Rodrigo 	0.001086957 0.000000000
## 	Rolando 	0.001086957 0.000000000
## 	Roman 	0.001086957 0.000000000
## 	Romeo 	0.001086957 0.000000000
## 	Ron 	0.001086957 0.000000000
## 	Ronald 	0.001086957 0.000000000
## 	Ronny 	0.001086957 0.000000000
## 	Roosevelt 	0.001086957 0.000000000
## 	Rosa 	0.001086957 0.000000000
## 	Rosalie 	0.001086957 0.000000000
## 	Rosalind 	0.001086957 0.000000000
## 	Rosalinda 	0.001086957 0.000000000
## 	Rosanna 	0.001086957 0.000000000
## 	Rosanne 	0.001086957 0.000000000
## 	Rosario 	0.000000000 0.008547009
## 	Roscoe 	0.001086957 0.000000000
## 	Rose 	0.001086957 0.000000000
## 	Rosemarie 	0.001086957 0.000000000
## 	Rosemary 	0.001086957 0.000000000
## 	Rosetta 	0.001086957 0.000000000
## 	Rosie 	0.001086957 0.000000000
## 	Roslyn 	0.001086957 0.000000000
## 	Ross 	0.001086957 0.000000000
## 	Roxana 	0.001086957 0.000000000
## 	Roxanna 	0.001086957 0.000000000
## 	Roy 	0.001086957 0.000000000
## 	Royce 	0.000000000 0.008547009
## 	Ruben 	0.001086957 0.000000000
## 	Ruby 	0.001086957 0.000000000
## 	Rudy 	0.001086957 0.000000000
## 	Rufus 	0.001086957 0.000000000
## 	Russel 	0.001086957 0.000000000
## 	Russell 	0.001086957 0.000000000
## 	Ruth 	0.001086957 0.000000000
## 	Sadie 	0.001086957 0.000000000
## 	Salvador 	0.001086957 0.000000000
## 	Samantha 	0.001086957 0.000000000
## 	Sammie 	0.000000000 0.008547009
## 	Samuel 	0.001086957 0.000000000
## 	Sandra 	0.001086957 0.000000000
## 	Santos 	0.000000000 0.008547009
## 	Sara 	0.001086957 0.000000000
## 	Sarah 	0.001086957 0.000000000
## 	Saul 	0.001086957 0.000000000
## 	Savannah 	0.001086957 0.000000000
## 	Scott 	0.001086957 0.000000000
## 	Scottie 	0.000000000 0.008547009
## 	Scotty 	0.001086957 0.000000000
## 	Selena 	0.001086957 0.000000000
## 	Selina 	0.001086957 0.000000000
## 	Serena 	0.001086957 0.000000000
## 	Sharon 	0.001086957 0.000000000
```

```
## Sharron     0.001086957 0.000000000
## Sheila      0.001086957 0.000000000
## Shelby      0.000000000 0.008547009
## Shelia      0.001086957 0.000000000
## Shelley     0.001086957 0.000000000
## Shelly      0.000000000 0.008547009
## Shelton     0.001086957 0.000000000
## Sherrie     0.001086957 0.000000000
## Sherry      0.001086957 0.000000000
## Shirley     0.001086957 0.000000000
## Sidney      0.000000000 0.008547009
## Silvia      0.001086957 0.000000000
## Simon       0.001086957 0.000000000
## Simone      0.001086957 0.000000000
## Sofia       0.001086957 0.000000000
## Sonia       0.001086957 0.000000000
## Sophia      0.001086957 0.000000000
## Sophie      0.001086957 0.000000000
## Spencer     0.001086957 0.000000000
## Stephan     0.001086957 0.000000000
## Stephanie   0.001086957 0.000000000
## Stephany    0.001086957 0.000000000
## Stephen     0.001086957 0.000000000
## Sterling    0.000000000 0.008547009
## Steve       0.001086957 0.000000000
## Stuart      0.001086957 0.000000000
## Sunny       0.000000000 0.008547009
## Susan       0.001086957 0.000000000
## Susana      0.001086957 0.000000000
## Susie       0.001086957 0.000000000
## Sybil       0.001086957 0.000000000
## Sydney      0.000000000 0.008547009
## Sylvester   0.001086957 0.000000000
## Sylvia      0.001086957 0.000000000
## Tamara      0.001086957 0.000000000
## Tanya       0.001086957 0.000000000
## Taylor      0.000000000 0.008547009
## Terence     0.001086957 0.000000000
## Teresa      0.001086957 0.000000000
## Terrell     0.000000000 0.008547009
## Terrence    0.001086957 0.000000000
## Terry       0.000000000 0.008547009
## Thaddeus    0.001086957 0.000000000
## Thalia      0.001086957 0.000000000
## Theodore    0.001086957 0.000000000
## Theresa     0.001086957 0.000000000
## Theron      0.001086957 0.000000000
## Thomas      0.001086957 0.000000000
## Thurman     0.001086957 0.000000000
## Timothy     0.001086957 0.000000000
## Tina        0.001086957 0.000000000
## Tobias      0.001086957 0.000000000
## Toby        0.000000000 0.008547009
## Todd        0.001086957 0.000000000
```

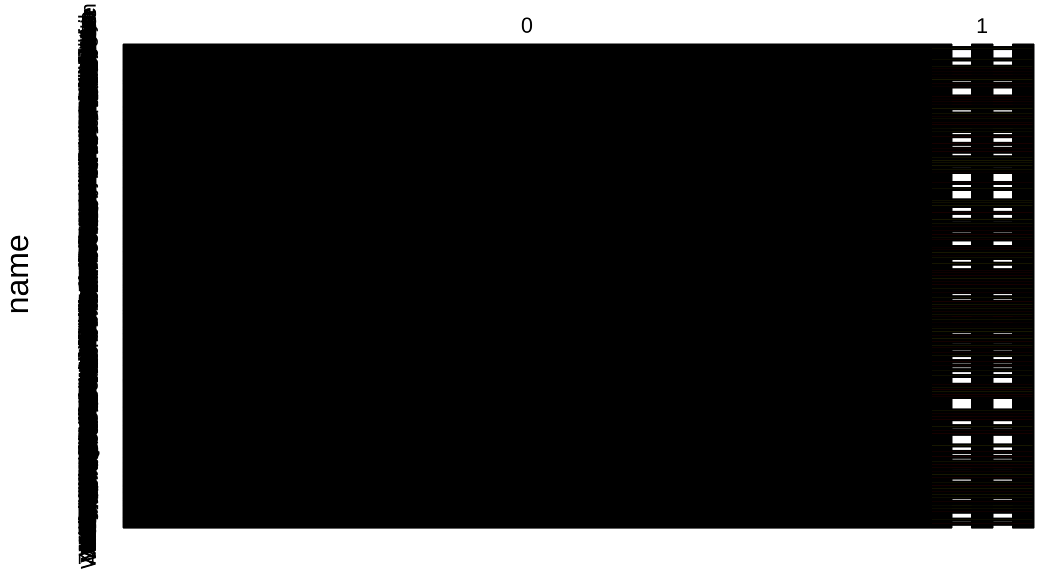
```
##   Tomas       0.001086957 0.000000000
##   Tommy       0.001086957 0.000000000
##   Tony        0.001086957 0.000000000
##   Tracy       0.000000000 0.008547009
##   Trenton     0.001086957 0.000000000
##   Trudy       0.001086957 0.000000000
##   Tucker      0.001086957 0.000000000
##   Tyler       0.001086957 0.000000000
##   Ulysses     0.001086957 0.000000000
##   Unknown     0.000000000 0.008547009
##   Ursula      0.001086957 0.000000000
##   Valarie     0.001086957 0.000000000
##   Valentina   0.001086957 0.000000000
##   Valerie     0.001086957 0.000000000
##   Van         0.000000000 0.008547009
##   Vance       0.001086957 0.000000000
##   Vaughn      0.001086957 0.000000000
##   Velma       0.001086957 0.000000000
##   Verna       0.001086957 0.000000000
##   Vernon      0.001086957 0.000000000
##   Veronica    0.001086957 0.000000000
##   Vicente     0.001086957 0.000000000
##   Vicki       0.001086957 0.000000000
##   Vickie      0.001086957 0.000000000
##   Vicky       0.001086957 0.000000000
##   Victoria    0.001086957 0.000000000
##   Vince       0.001086957 0.000000000
##   Viola       0.001086957 0.000000000
##   Vito        0.001086957 0.000000000
##   Walker      0.001086957 0.000000000
##   Wallace     0.001086957 0.000000000
##   Wanda       0.001086957 0.000000000
##   Ward        0.001086957 0.000000000
##   Warren      0.001086957 0.000000000
##   Waylon      0.001086957 0.000000000
##   Wayne       0.001086957 0.000000000
##   Wendell     0.001086957 0.000000000
##   Wendy       0.001086957 0.000000000
##   Wesley      0.001086957 0.000000000
##   Whitney     0.000000000 0.008547009
##   Wilbert     0.001086957 0.000000000
##   Wilbur      0.001086957 0.000000000
##   Wiley       0.001086957 0.000000000
##   Wilfred     0.001086957 0.000000000
##   Will        0.001086957 0.000000000
##   Willa       0.001086957 0.000000000
##   Willard     0.001086957 0.000000000
##   William     0.001086957 0.000000000
##   Willie      0.000000000 0.008547009
##   Willow      0.001086957 0.000000000
##   Wilma       0.001086957 0.000000000
##   Wilson      0.001086957 0.000000000
##   Winifred    0.001086957 0.000000000
##   Winston     0.001086957 0.000000000
```

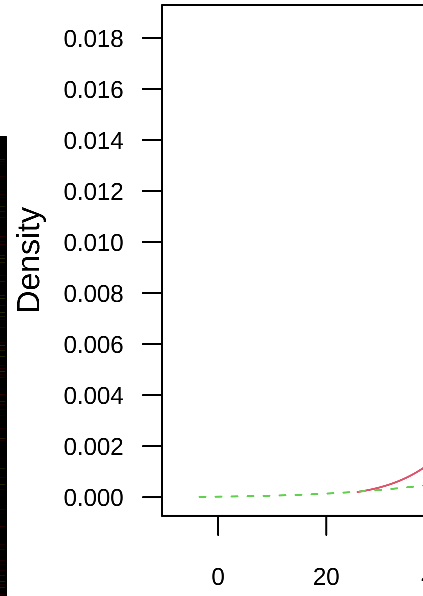
```
##   Woodrow     0.001086957 0.000000000
##   Xavier      0.001086957 0.000000000
##   Yolanda     0.001086957 0.000000000
##   Yvette      0.001086957 0.000000000
##   Zachary     0.001086957 0.000000000
##   Zane        0.001086957 0.000000000
##   Zoe         0.001086957 0.000000000
##
## ------------------------------------------------------------------------------
##  ::: n::0 (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (920 obs.);  Bandwidth 'bw' = 19.08
##
##        x                 y
##  Min.   : 25.76   Min.   :0.0002345
##  1st Qu.: 54.38   1st Qu.:0.0016675
##  Median : 83.00   Median :0.0067750
##  Mean   : 83.00   Mean   :0.0087036
##  3rd Qu.:111.62   3rd Qu.:0.0157389
##  Max.   :140.24   Max.   :0.0209057
##
## ------------------------------------------------------------------------------
##  ::: n::1 (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (117 obs.);  Bandwidth 'bw' = 28.82
##
##        x                  y
##  Min.   : -3.458   Min.   :0.0001553
##  1st Qu.: 39.771   1st Qu.:0.0011039
##  Median : 83.000   Median :0.0044853
##  Mean   : 83.000   Mean   :0.0057620
##  3rd Qu.:126.229   3rd Qu.:0.0104197
##  Max.   :169.458   Max.   :0.0138403
##
## ------------------------------------------------------------------------------
##  ::: female::0 (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (920 obs.);  Bandwidth 'bw' = 7685
##
##        x                  y
##  Min.   : -23054   Min.   :0.000e+00
##  1st Qu.: 624902   1st Qu.:0.000e+00
```

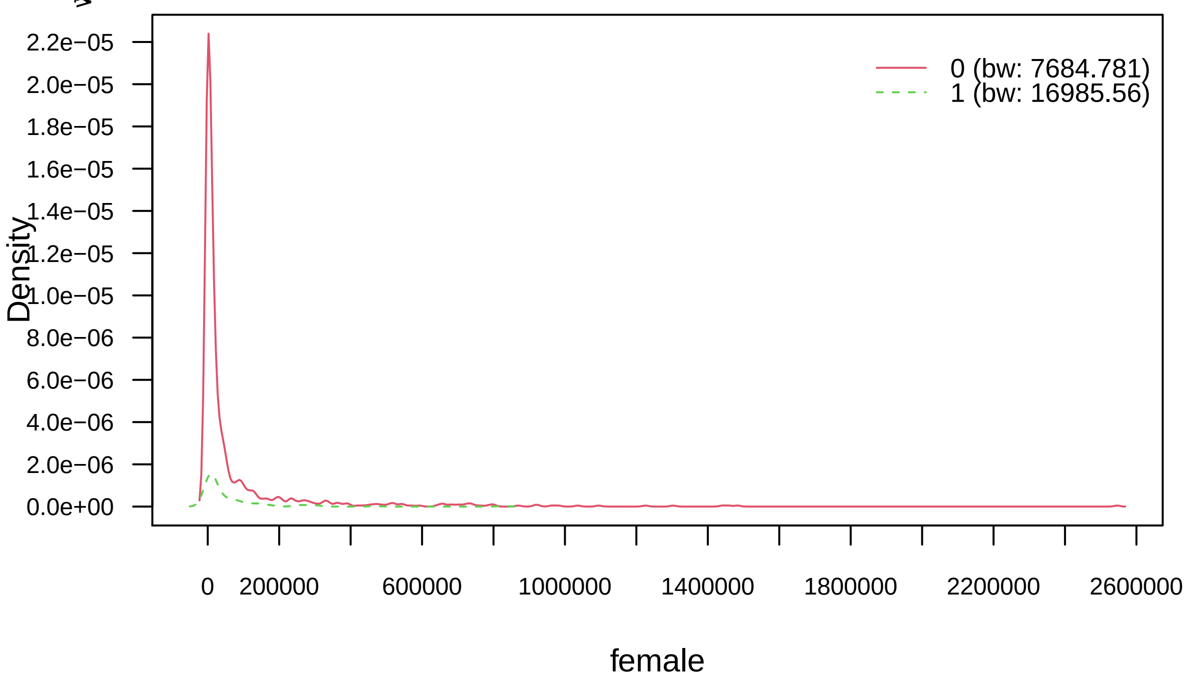
```
##  Median :1272859   Median :1.415e-09
##  Mean   :1272859   Mean   :3.852e-07
##  3rd Qu.:1920816   3rd Qu.:9.365e-08
##  Max.   :2568772   Max.   :2.524e-05
##
## ------------------------------------------------------------------------------
##  ::: female::1 (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (117 obs.);  Bandwidth 'bw' = 1.699e+04
##
##        x                y
##  Min.   :-50368   Min.   :0.000e+00
##  1st Qu.:181800   1st Qu.:2.300e-11
##  Median :413968   Median :7.720e-08
##  Mean   :413968   Mean   :1.075e-06
##  3rd Qu.:646136   3rd Qu.:6.262e-07
##  Max.   :878304   Max.   :1.365e-05
##
## ------------------------------------------------------------------------------
##  ::: male::0 (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (920 obs.);  Bandwidth 'bw' = 8246
##
##        x                 y
##  Min.   : -24737   Min.   :0.000e+00
##  1st Qu.:1017954   1st Qu.:0.000e+00
##  Median :2060646   Median :0.000e+00
##  Mean   :2060646   Mean   :2.394e-07
##  3rd Qu.:3103338   3rd Qu.:3.365e-08
##  Max.   :4146029   Max.   :2.559e-05
##
## ------------------------------------------------------------------------------
##  ::: male::1 (KDE)
## ------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (117 obs.);  Bandwidth 'bw' = 1.078e+04
##
##        x                y
##  Min.   :-32133   Min.   :1.000e-11
##  1st Qu.: 88444   1st Qu.:1.436e-07
##  Median :209020   Median :3.046e-07
##  Mean   :209020   Mean   :2.071e-06
##  3rd Qu.:329597   3rd Qu.:1.158e-06
```

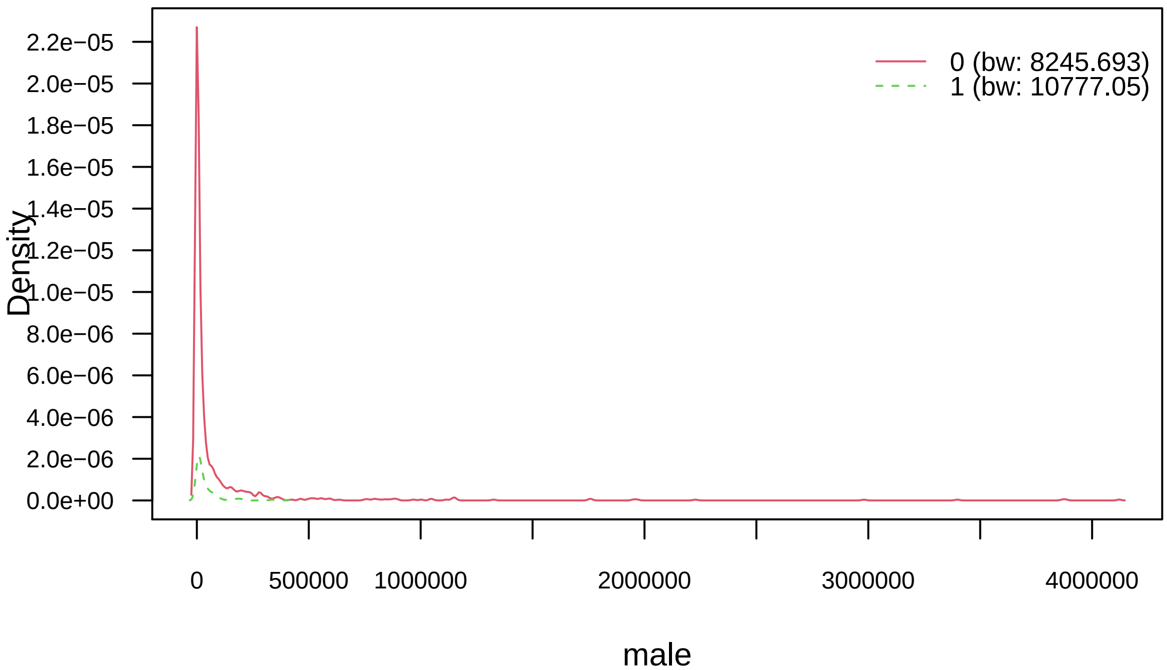
```
##  Max.   :450174   Max.   :1.882e-05
##
## --------------------------------------------------------------------------------
##  ::: total::0 (KDE)
## --------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (920 obs.);  Bandwidth 'bw' = 2.304e+04
##
##        x                 y
##  Min.   : -60096   Min.   :0.000e+00
##  1st Qu.:1006950   1st Qu.:4.000e-12
##  Median :2073996   Median :7.332e-09
##  Mean   :2073996   Mean   :2.341e-07
##  3rd Qu.:3141042   3rd Qu.:6.020e-08
##  Max.   :4208088   Max.   :8.285e-06
##
## --------------------------------------------------------------------------------
##  ::: total::1 (KDE)
## --------------------------------------------------------------------------------
##
## Call:
##  density.default(x = x, na.rm = TRUE)
##
## Data: x (117 obs.);  Bandwidth 'bw' = 2.425e+04
##
##        x                y
##  Min.   :-63474   Min.   :0.000e+00
##  1st Qu.:181006   1st Qu.:3.292e-08
##  Median :425486   Median :2.105e-07
##  Mean   :425486   Mean   :1.021e-06
##  3rd Qu.:669967   3rd Qu.:7.281e-07
##  Max.   :914447   Max.   :8.295e-06
##
## --------------------------------------------------------------------------------
##
## # ... and 1 more table
##
## --------------------------------------------------------------------------------
```

```r
plot(model)
```

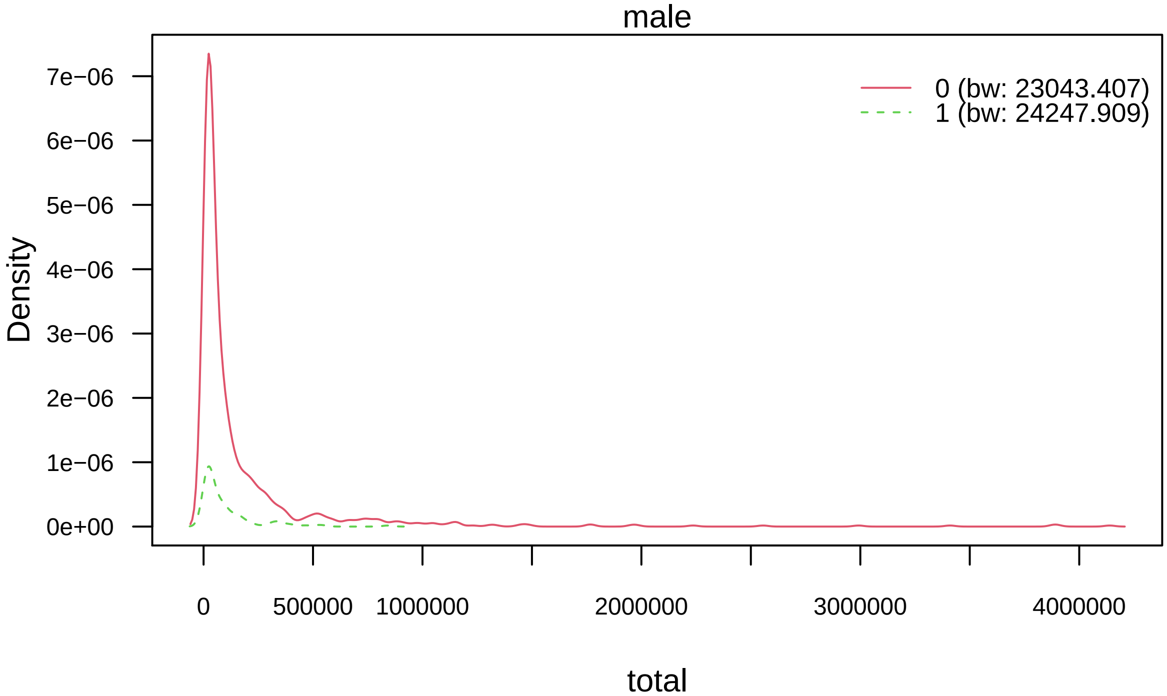

name
0
1


Density
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
20

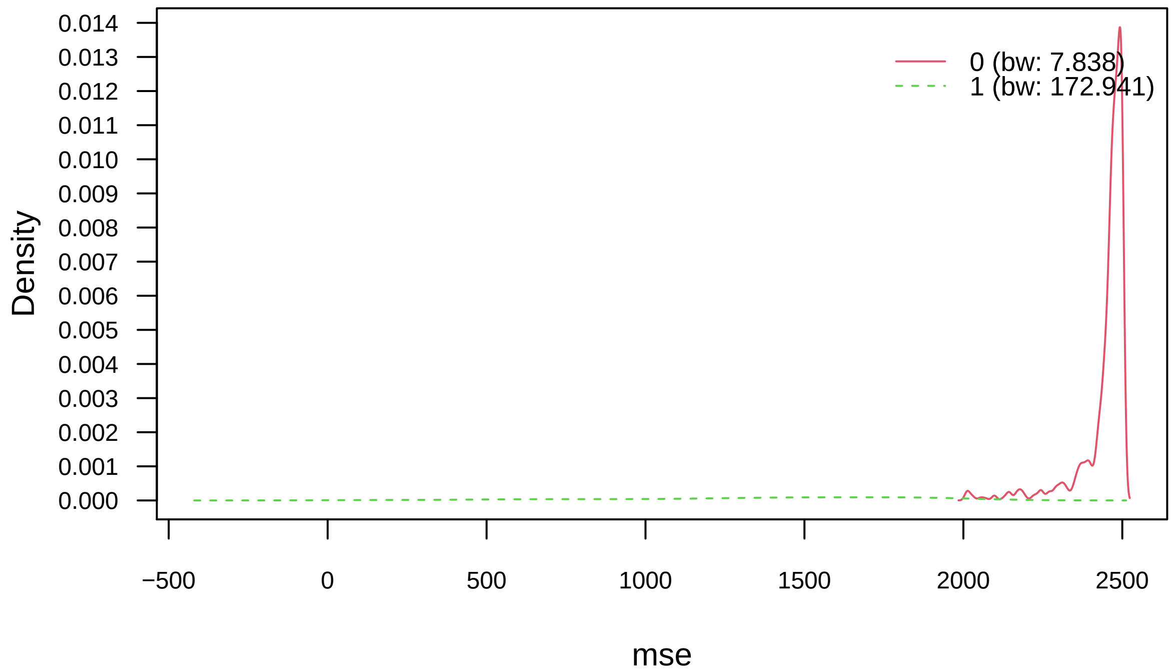

Density
2.2e−05
2.0e−05
1.8e−05
1.6e−05
1.4e−05
1.2e−05
1.0e−05
8.0e−06
6.0e−06
4.0e−06
2.0e−06
0.0e+00
0 (bw: 7684.781)
1 (bw: 16985.56)
0
200000
600000
1000000
1400000
1800000
2200000
2600000
female



0 (bw: 8245.693)
1 (bw: 10777.05)
Density
2.2e−05
2.0e−05
1.8e−05
1.6e−05
1.4e−05
1.2e−05
1.0e−05
8.0e−06
6.0e−06
4.0e−06
2.0e−06
0.0e+00
0
500000
1000000
2000000
3000000
4000000
male



0 (bw: 23043.407)
1 (bw: 24247.909)
Density
7e−06
6e−06
5e−06
4e−06
3e−06
2e−06
1e−06
0e+00
0
500000
1000000
2000000
3000000
4000000
total




```r
p <- predict(model, TrainSet, type = 'prob')
```

```
## Warning: predict.naive_bayes(): more features in the newdata are provided as
## there are probability tables in the object. Calculation is performed based on
## features to be found in the tables.
```

```r
head(cbind(p, TrainSet))
```

```
##           0            1    name  n female   male  total      mse isNeutral
## 1 0.9998937 1.062817e-04     Zoe 83  91028    383  91411 2446.768         0
## 2 0.9999924 7.609936e-06 Camilla 83  14413      0  14413 2500.000         0
## 3 0.9999781 2.190985e-05  Marsha 83 103510    250 103760 2482.349         0
## 4 0.9999855 1.448226e-05  Elbert 83     54  18348  18402 2482.788         0
## 5 0.9999678 3.217844e-05   Wilma 83  73774    417  74191 2478.787         0
## 6 0.9999542 4.578960e-05  Miguel 83    968 158576 159544 2441.390         0
```

Confusion matrix for train data, Calculate misscalculation/error,and model accuracy

```r
p1 <- predict(model, TrainSet)
```

```
## Warning: predict.naive_bayes(): more features in the newdata are provided as
## there are probability tables in the object. Calculation is performed based on
## features to be found in the tables.
```

```r
(tab1 <- table(p1, TrainSet$isNeutral))
```

```
##
## p1    0   1
##   0 919   0
##   1   1 117
```

```r
miscalc <- (1 - sum(diag(tab1)) / sum(tab1)) * 100

accuracy <- (100- miscalc)
accuracy
```

```
## [1] 99.90357
```

The model has an accuracy of 99.90357 on training data for the correct classification of gender neutral names.

Results

We can use logistic regression to make a prediction of gender from a name, we can use Random Forest Classification and Naive Bayes to make whether a name is gender neutral with close to 100% and over 99% accuracy, respectively. These methods are effective in determining whether a name is considered gender neutral based on its usage between genders. Using these methods indicate that the methods of classification between genders is highly accurate.

Conclusion

The results indicate the name and the proportion of each biological sex given that name and a prediction of whether the name is generally considered male or female. By using this data, a prospective parent can consider how names are viewed regarding gender neutrality based on statistical data from the SSA dataset.