

The babynames data

DATA MANIPULATION WITH DPLYR



Chris Cardillo

Data Scientist at DataCamp

The babynames data

babynames

```
# A tibble: 332,595 x 3
  year name    number
<dbl> <chr>    <int>
1  1880 Aaron      102
2  1880 Ab         5
3  1880 Abbie      71
4  1880 Abbott      5
5  1880 Abby        6
6  1880 Abe       50
7  1880 Abel        9
8  1880 Abigail    12
9  1880 Abner      27
10 1880 Abraham     81
# ... with 332,585 more rows
```

Frequency of a name

```
babynames %>%  
  filter(name == "Amy")
```

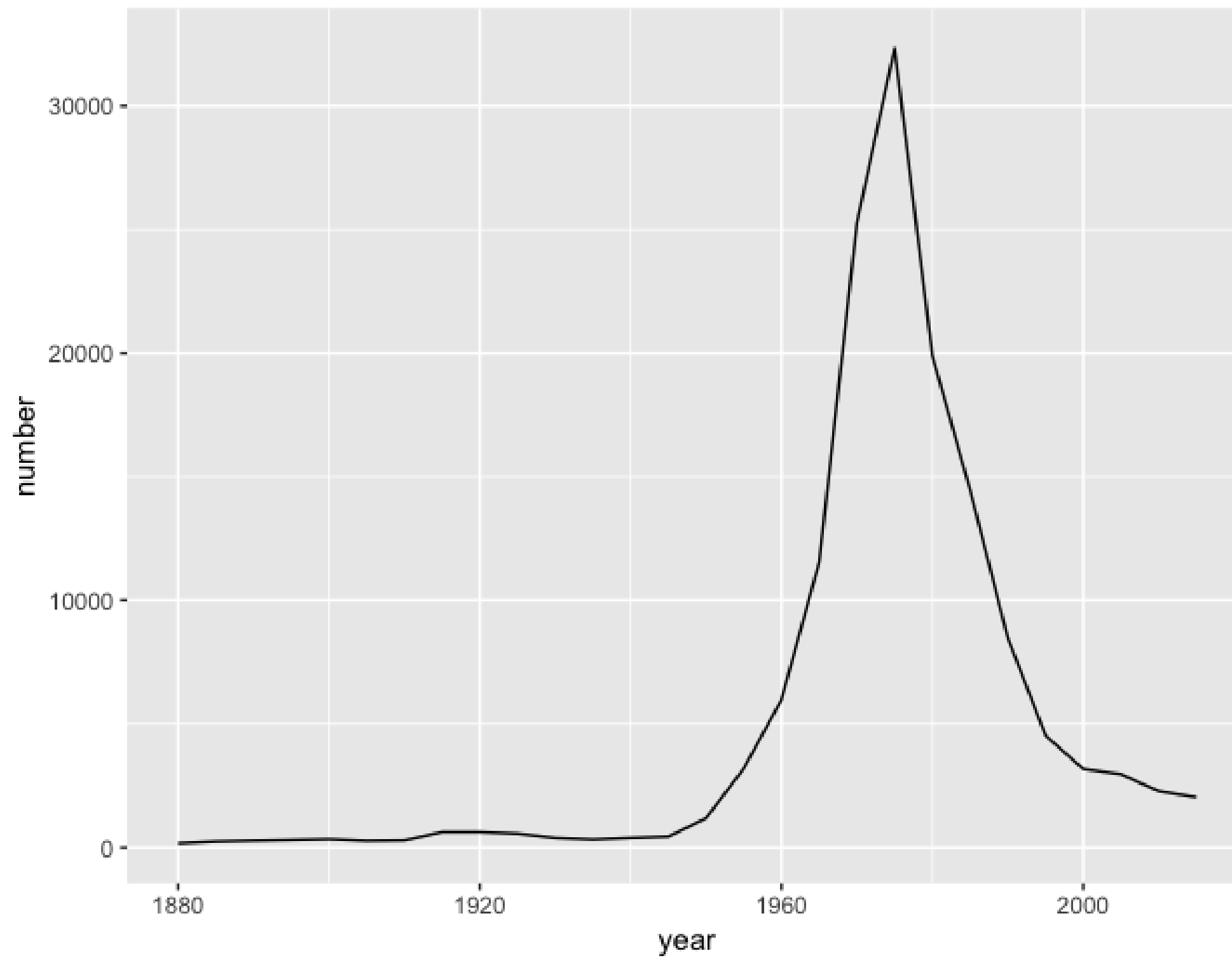
```
# A tibble: 28 x 3  
  year name  number  
  <dbl> <chr>  <int>  
1  1880 Amy    167  
2  1885 Amy    240  
3  1890 Amy    275  
4  1895 Amy    303  
5  1900 Amy    335  
6  1905 Amy    269  
7  1910 Amy    287  
8  1915 Amy    624  
9  1920 Amy    624  
10 1925 Amy    560  
# ... with 18 more rows
```

Amy plot

```
library(ggplot2)
```

```
babynames_filtered <- babynames %>%  
  filter(name == "Amy")
```

```
ggplot(babynames_filtered, aes(x = year, y = number)) +  
  geom_line()
```



Filter for multiple names

```
babynames_multiple <- babynames %>%  
  filter(name %in% c("Amy", "Christopher"))
```

When was each name most common?

```
babynames %>%  
  group_by(name) %>%  
  top_n(1, number)
```

```
# A tibble: 54,881 x 3  
# Groups:   name [48,040]  
   year name      number  
   <dbl> <chr>      <int>  
1  1880 Arch         61  
2  1880 Bird         17  
3  1880 Ednah         6  
4  1880 Erasmus        5  
5  1880 Garfield    122  
6  1880 Harve        17  
7  1880 Lidie         7  
8  1880 Loula        13  
9  1880 Lovisa         5  
10 1880 Lulie         8  
# ... with 54,871 more rows
```

Let's practice!

DATA MANIPULATION WITH DPLYR

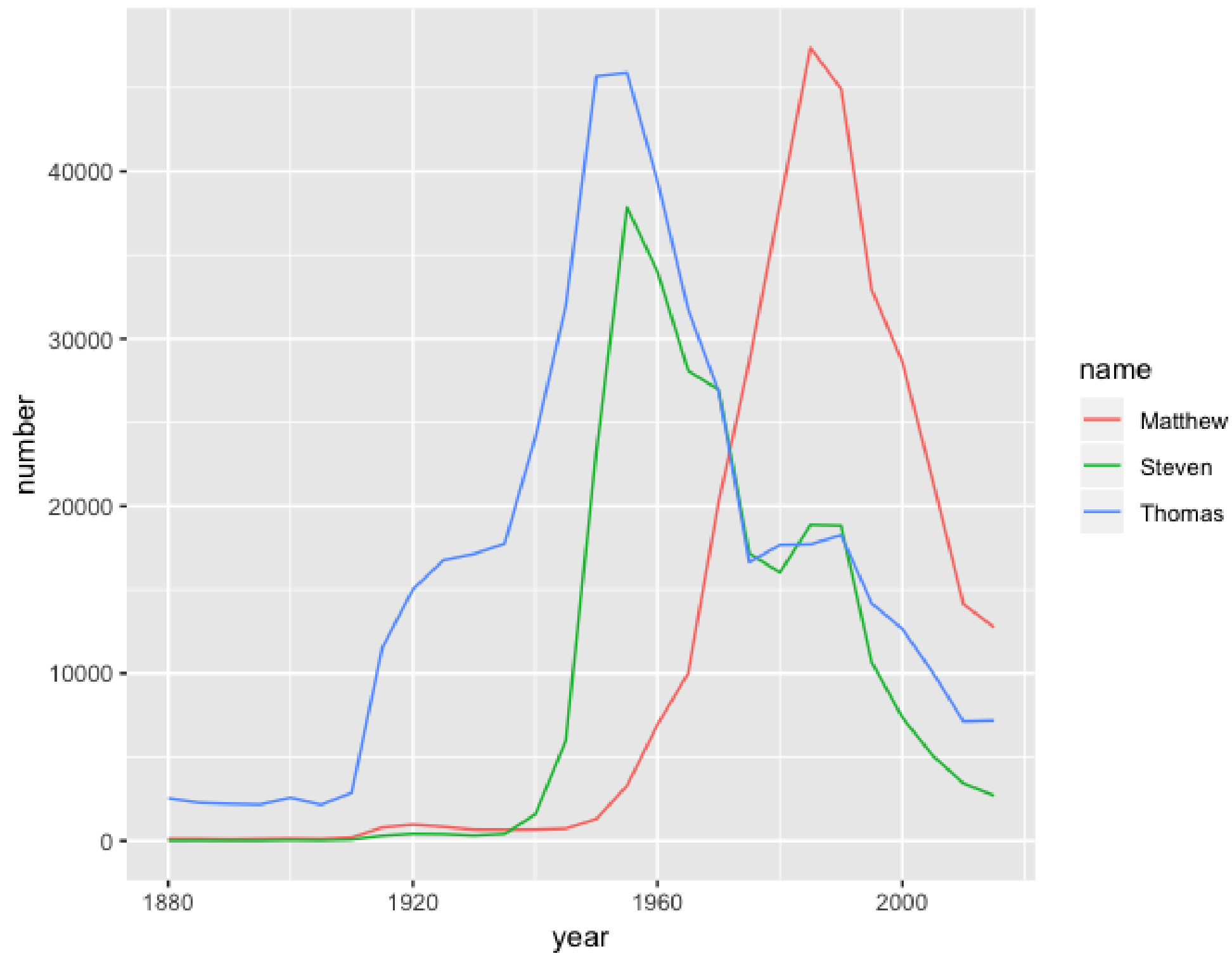
Grouped mutates

DATA MANIPULATION WITH DPLYR



Chris Cardillo

Data Scientist at DataCamp



Review: group_by() and summarize()

```
babynames %>%  
  group_by(year) %>%  
  summarize(year_total = sum(number))
```

```
# A tibble: 28 x 2  
  year year_total  
  <dbl>     <int>  
1  1880     201478  
2  1885     240822  
3  1890     301352  
4  1895     350934  
5  1900     450148  
6  1905     423875  
7  1910     590607  
8  1915    1830351  
9  1920    2259494  
10 1925    2330750  
# ... with 18 more rows
```

Combining group_by() and mutate()

```
babynames %>%  
  group_by(year) %>%  
  mutate(year_total = sum(number))
```

```
# A tibble: 332,595 x 4  
# Groups:   year [28]  
   year name      number year_total  
   <dbl> <chr>    <int>    <int>  
1  1880 Aaron      102    201478  
2  1880 Ab         5     201478  
3  1880 Abbie      71     201478  
4  1880 Abbott      5     201478  
5  1880 Abby       6     201478  
6  1880 Abe       50     201478  
7  1880 Abel       9     201478  
8  1880 Abigail    12     201478  
9  1880 Abner     27     201478  
10 1880 Abraham    81     201478  
# ... with 332,585 more rows
```

ungroup()

```
babynames %>%  
  group_by(year) %>%  
  mutate(year_total = sum(number)) %>%  
  ungroup()
```

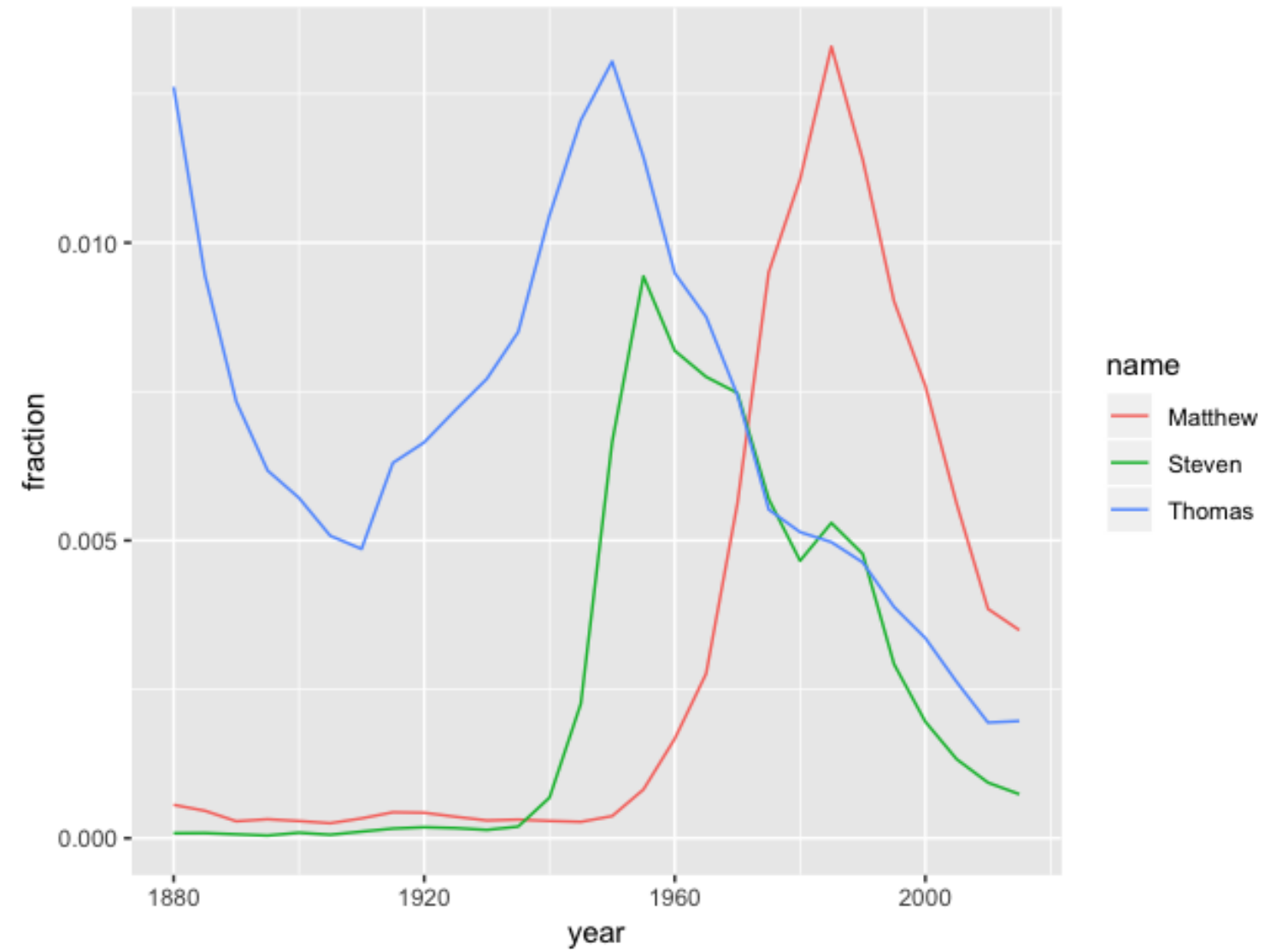
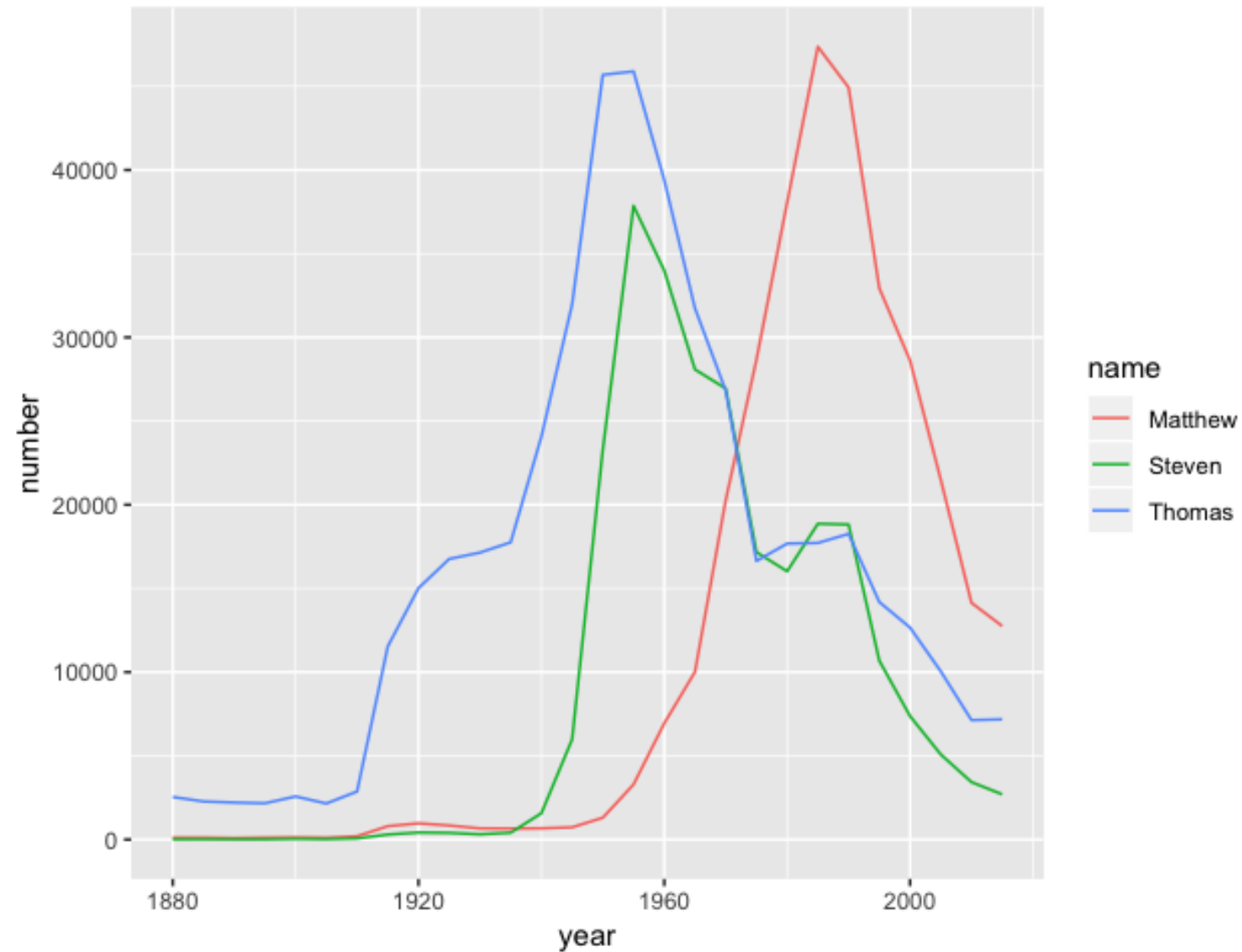
```
# A tibble: 332,595 x 4  
   year name      number year_total  
   <dbl> <chr>    <int>    <int>  
1  1880 Aaron      102    201478  
2  1880 Ab         5     201478  
3  1880 Abbie      71     201478  
4  1880 Abbott      5     201478  
5  1880 Abby        6     201478  
6  1880 Abe       50     201478  
7  1880 Abel        9     201478  
8  1880 Abigail    12     201478  
9  1880 Abner     27     201478  
10 1880 Abraham    81     201478  
# ... with 332,585 more rows
```

Add the fraction column

```
babynames %>%  
  group_by(year) %>%  
  mutate(year_total = sum(number)) %>%  
  ungroup() %>%  
  mutate(fraction = number / year_total)
```

```
# A tibble: 332,595 x 5  
  year name      number year_total fraction  
  <dbl> <chr>    <int>    <int>    <dbl>  
1  1880 Aaron      102    201478 0.000506  
2  1880 Ab         5     201478 0.0000248  
3  1880 Abbie      71     201478 0.000352  
4  1880 Abbott      5     201478 0.0000248  
5  1880 Abby       6     201478 0.0000298  
6  1880 Abe       50     201478 0.000248  
7  1880 Abel       9     201478 0.0000447  
8  1880 Abigail    12     201478 0.0000596  
9  1880 Abner     27     201478 0.000134  
10 1880 Abraham    81     201478 0.000402  
# ... with 332,585 more rows
```

Comparing visualizations



Let's practice!

DATA MANIPULATION WITH DPLYR

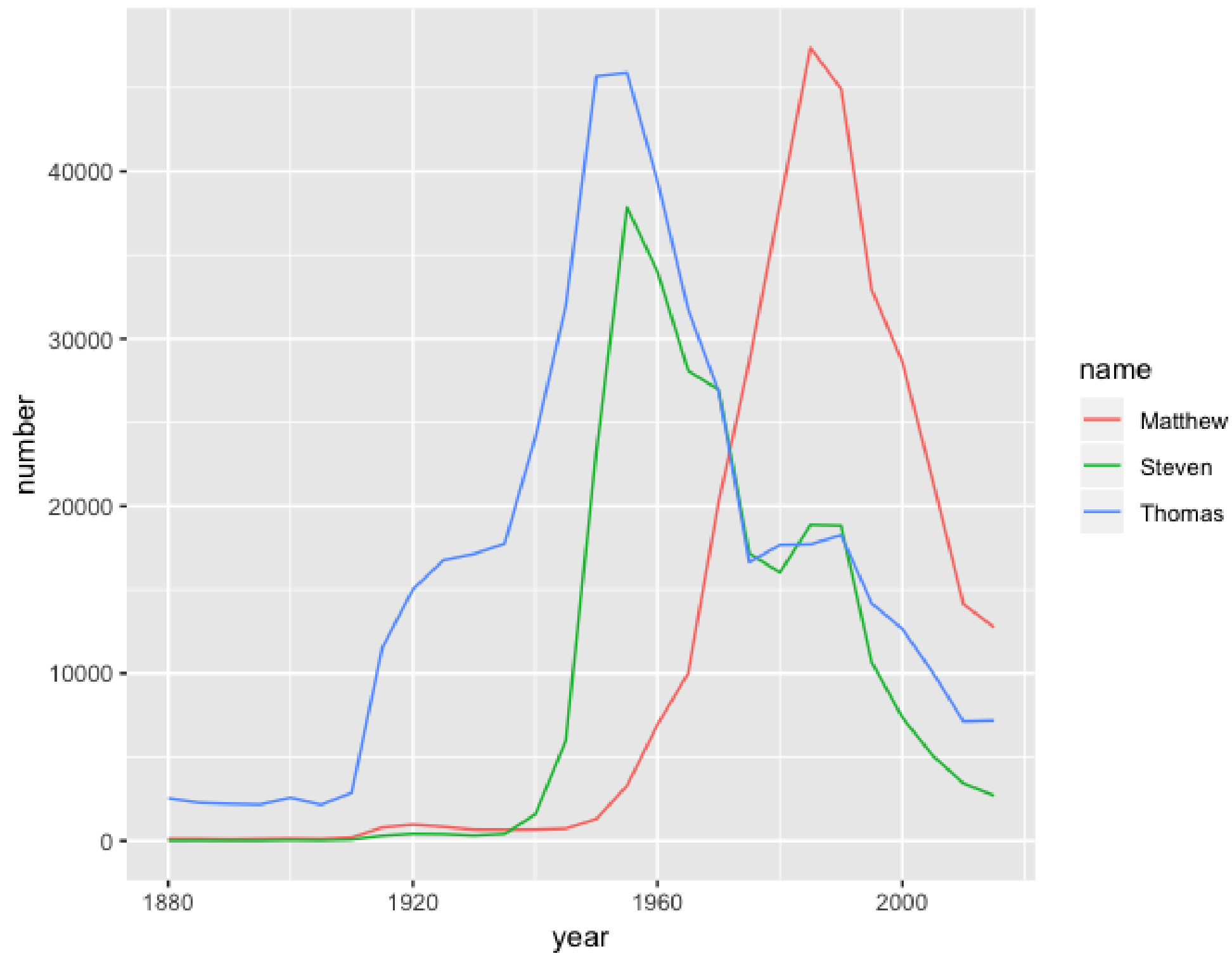
Window functions

DATA MANIPULATION WITH DPLYR



Chris Cardillo

Data Scientist at DataCamp



Window function

```
v <- c(1, 3, 6, 14)
```

```
v
```

```
[1] 1 3 6 14
```

```
lag(v)
```

```
[1] NA 1 3 6
```

Compare consecutive steps

```
v - lag(v)
```

```
[1] NA  2  3  8
```

Changes in popularity of a name

```
babynames_fraction <- babynames %>%  
  group_by(year) %>%  
  mutate(year_total = sum(number)) %>%  
  ungroup() %>%  
  mutate(fraction = number / year_total)
```

Matthew

```
babynames_fraction %>%  
  filter(name == "Matthew") %>%  
  arrange(year)
```

```
# A tibble: 28 x 5  
   year name    number year_total fraction  
   <dbl> <chr>    <int>      <int>    <dbl>  
1  1880 Matthew    113    201478 0.000561  
2  1885 Matthew    111    240822 0.000461  
3  1890 Matthew     86    301352 0.000285  
4  1895 Matthew    112    350934 0.000319  
5  1900 Matthew    130    450148 0.000289  
6  1905 Matthew    107    423875 0.000252  
7  1910 Matthew    197    590607 0.000334  
8  1915 Matthew    798   1830351 0.000436  
9  1920 Matthew    967   2259494 0.000428  
10 1925 Matthew    840   2330750 0.000360  
# ... with 18 more rows
```

Matthew over time

```
babynames_fraction %>%  
  filter(name == "Matthew") %>%  
  arrange(year) %>%  
  mutate(difference = fraction - lag(fraction))
```

```
# A tibble: 28 x 6  
  year name    number year_total fraction difference  
  <dbl> <chr>    <int>    <int>    <dbl>    <dbl>  
1  1880 Matthew    113    201478 0.000561 NA  
2  1885 Matthew    111    240822 0.000461 -0.0000999  
3  1890 Matthew     86    301352 0.000285 -0.000176  
4  1895 Matthew    112    350934 0.000319  0.0000338  
5  1900 Matthew    130    450148 0.000289 -0.0000304  
6  1905 Matthew    107    423875 0.000252 -0.0000364  
7  1910 Matthew    197    590607 0.000334  0.0000811  
8  1915 Matthew    798   1830351 0.000436  0.000102  
9  1920 Matthew    967   2259494 0.000428 -0.00000801  
10 1925 Matthew    840   2330750 0.000360 -0.0000676  
# ... with 18 more rows
```

Biggest jump in popularity

```
babynames_fraction %>%  
  filter(name == "Matthew") %>%  
  arrange(year) %>%  
  mutate(difference = fraction - lag(fraction)) %>%  
  arrange(desc(difference))
```

```
# A tibble: 28 x 6  
  year name      number year_total fraction difference  
  <dbl> <chr>    <int>      <int>    <dbl>      <dbl>  
1  1975 Matthew  28665    3014943 0.00951    0.00389  
2  1970 Matthew  20265    3604252 0.00562    0.00286  
3  1985 Matthew  47367    3563364 0.0133     0.00223  
4  1980 Matthew  38054    3439117 0.0111     0.00156  
5  1965 Matthew  10015    3624610 0.00276    0.00109  
6  1960 Matthew   6942    4152075 0.00167    0.000853  
7  1955 Matthew   3287    4012691 0.000819   0.000447  
8  1915 Matthew    798    1830351 0.000436   0.000102  
9  1950 Matthew   1303    3502592 0.000372   0.0000967  
10 1910 Matthew    197     590607 0.000334   0.0000811  
# ... with 18 more rows
```


Changes within every name

```
babynames_fraction %>%  
  arrange(name, year) %>%  
  mutate(difference = fraction - lag(fraction)) %>%  
  group_by(name) %>%  
  arrange(desc(difference))
```

```
# A tibble: 332,595 x 6  
# Groups:   name [48,040]  
   year name      number year_total fraction difference  
   <dbl> <chr>      <int>      <int>      <dbl>      <dbl>  
1  1880 John        9701      201478    0.0481    0.0481  
2  1880 William     9562      201478    0.0475    0.0475  
3  1880 Mary        7092      201478    0.0352    0.0352  
4  1880 James       5949      201478    0.0295    0.0295  
5  1880 Charles     5359      201478    0.0266    0.0266  
6  1880 George      5152      201478    0.0256    0.0256  
7  1880 Frank       3255      201478    0.0162    0.0162  
8  1935 Shirley    42790     2088487    0.0205    0.0137  
9  1880 Joseph      2642      201478    0.0131    0.0131  
10 1880 Anna        2616      201478    0.0130    0.0129  
# ... with 332,585 more rows
```

Let's practice!

DATA MANIPULATION WITH DPLYR

Congratulations!

DATA MANIPULATION WITH DPLYR



Chris Cardillo

Data Scientist at DataCamp

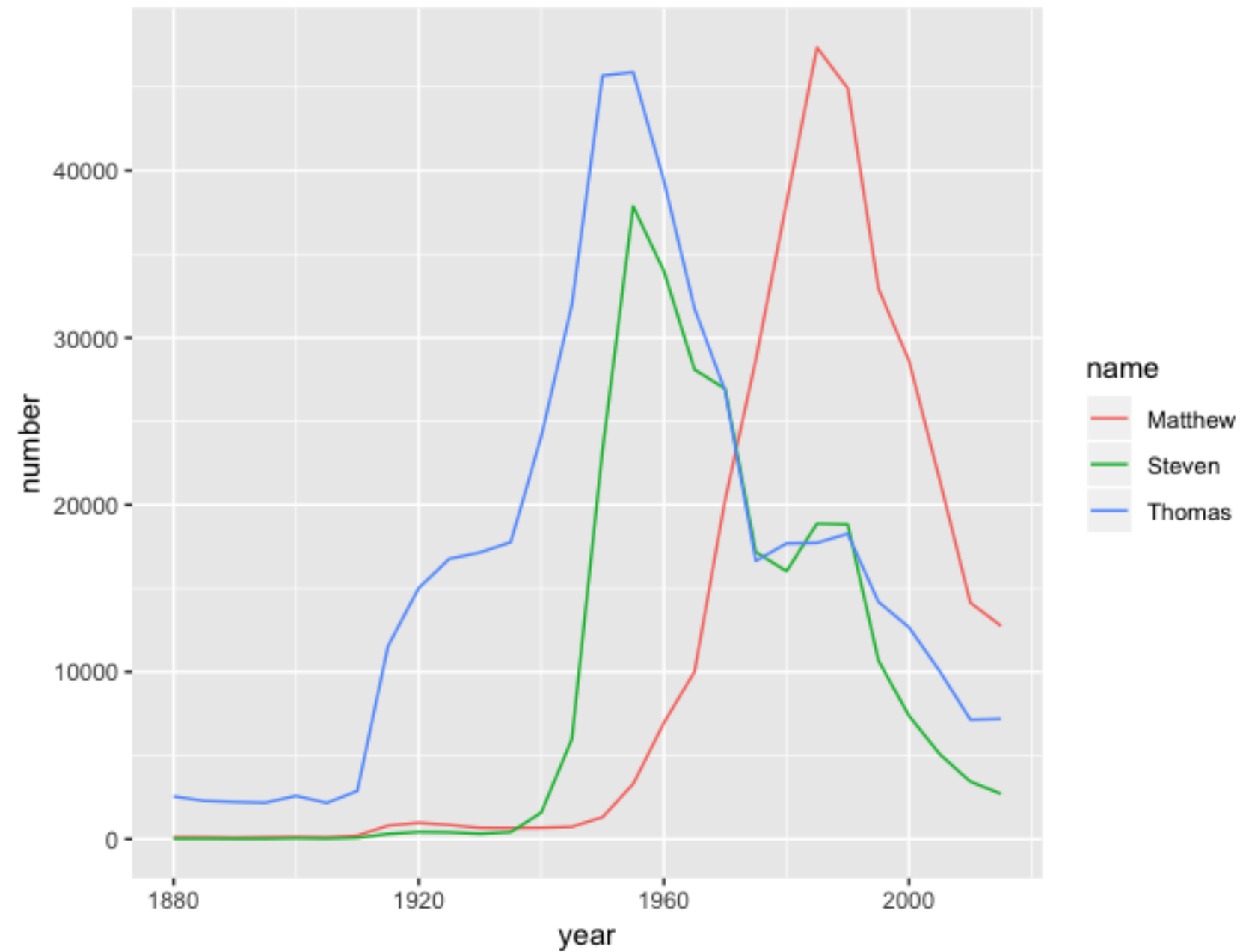
Summary

- `select()`
- `filter()`
- `mutate()`
- `arrange()`
- `count()`
- `group_by()`
- `summarize()`

Verbs table

	Keeps only specified variables	Keeps other variables
Can't change values	select	rename
Can change values	transmute	mutate

babynames data



Other DataCamp courses

- Exploratory Data Analysis in R: Case Study
- Working with Data in the Tidyverse
- Machine Learning in the Tidyverse
- Categorical Data in the Tidyverse

Congratulations!

DATA MANIPULATION WITH DPLYR