Copy the Master

CODE

SDS 192: MP3

Emily Rhyu

Smith College

Julia Walker

Smith College

Kitty Chen

Smith College

June 30, 2022

Our Approach

The goal of this project was to mimic the data graphic from the Flowing Data post by finding and then plotting the most unisex names (meaning equally assigned to female and male babies) found in the babynames package. A general strategy we employed was to go through the Jessie start code line by line to figure out what each part was doing, which helped us successfully generalize it later.

To start, we found the root mean squared error (RMSE) for the name "Jessie". Then, to find the RMSE for all names, we first considered how we could make the data from the babynames package more manageable. First, we filtered by the years in between 1930 and 2012 and then removed rows in which there was a NA value (meaning that the name was not assigned to either males or females.) We also reshaped the data by applying pivot_wider. After altering the data, we created a function called find_rmse that adds columns error and squared error and then summarizes that data to give the mean squared error (mse) and root mean squared error (rmse). We then applied this function to the altered data, called all_babies, that we grouped by name.

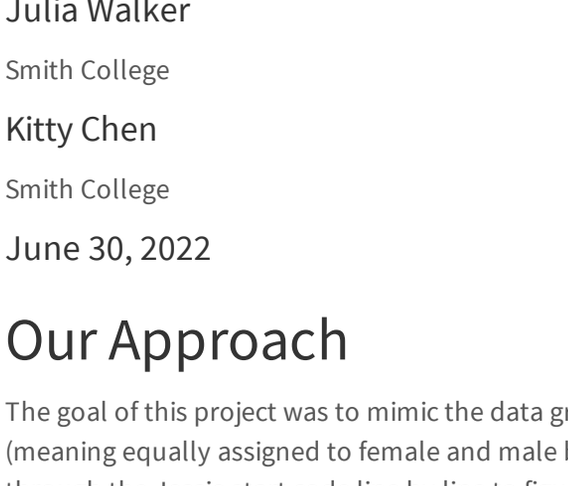
We tried various things when filtering to get the top 35 most unisex names: filtering out names with NA in the M or F column, finding the total occurrences of each name and considering the most popular names, and finding the total years in which each name appeared and including names that occurred in at least 70 years. We were not entirely successful at matching the original data graphic.

To gather the data for the time series, we filtered the original babynames data frame to include only the years of interest. We rearranged and mutated the data based on the "Jessie" plot code. To limit the names included to only the top 35 most unisex ones, we did an inner join between our modified babynames data frame and the data frame we created earlier with the 35 most unisex names.

To draw the points representing the most unisex years, we started with the code to draw the point for the name "Jessie" and wrote a function to generalize the process. Then we used map_dfr() to iterate those steps over the list of the top 35 most unisex names.

To create the annotations for the exhibit, we used the tribble() or transposed tibble command to manually create the descriptions for each notable name trend. This enabled us to construct a new data frame in which each row corresponds to a single segment, such as the year and the composition of boys to girls ratio.

Finally, to draw the plot, we used a combination of the line, area, point, path, and text geoms. We mimicked a lot of elements from the sample Jessie plot, including the fill and scale of the y axis, but changed other elements including but not limited to adding a facet wrap based on name and adding annotations with segments to the plots of certain names.



Copy the Master Assignment

Step 1: Make the plot for "Jessie"

Step 1A: Gather the data for "Jessie"

HIDE

HIDE

```
jessie <- babynames %>%
  filter(
    name == "Jessie",
    year >= 1930 & year < 2012
  ) %>%
  select(-prop) %>%
  pivot_wider(names_from = sex, values_from = n) %>%
  mutate(pct_girls = F / (F + M))
jessie
```

year <dbl>	name <chr>	F <int>	M <int>	pct_girls <dbl>
1930	Jessie	2196	1330	0.6228020
1931	Jessie	1930	1267	0.6036910
1932	Jessie	1895	1282	0.5964747
1933	Jessie	1807	1077	0.6265603
1934	Jessie	1793	1091	0.6217060
1935	Jessie	1618	1103	0.5946343
1936	Jessie	1586	1013	0.6102347
1937	Jessie	1552	1040	0.5987654
1938	Jessie	1475	971	0.6030253
1939	Jessie	1396	1058	0.5688672

1-10 of 82 rows   Previous 1 2 3 4 5 6 ... 9 Next

Step 1B: Compute the "most unisex year"

HIDE

HIDE

```
jessie_unisex_year <- jessie %>%
  mutate(distance = abs(pct_girls - 0.5)) %>%
  arrange(distance) %>%
  head(1)
jessie_unisex_year
```

year <dbl>	name <chr>	F <int>	M <int>	pct_girls <dbl>	distance <dbl>
1949	Jessie	1031	1023	0.5019474	0.00194742

1 row

Step 1C: Add the annotations for "Jessie"

HIDE

HIDE

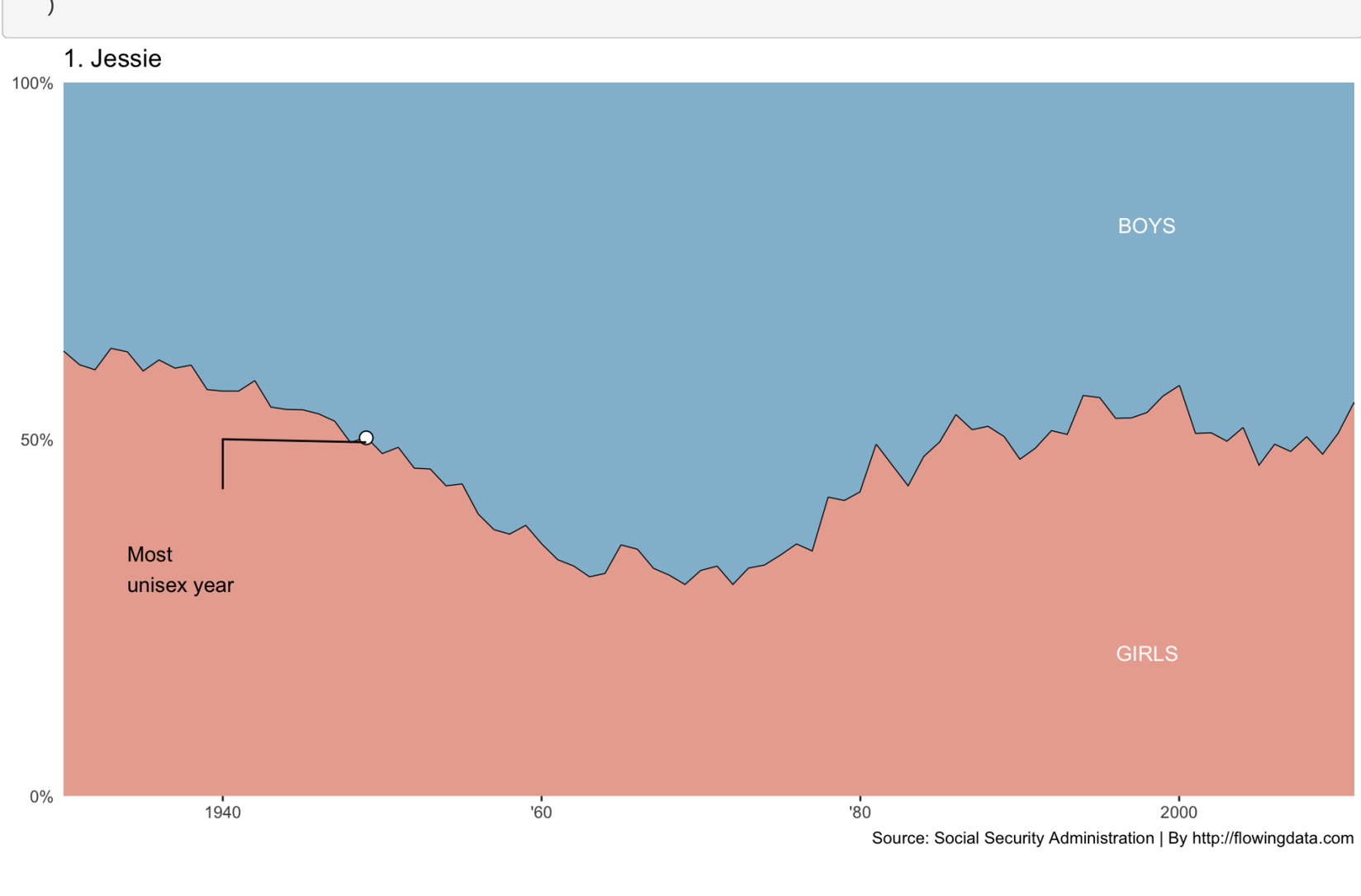
```
jessie_context <- tribble(
  ~year_label, ~vpos, ~hjust, ~name, ~text,
  1934, 0.35, "left", "Jessie", "Most\nunisex year"
)

jessie_segments <- tribble(
  ~year, ~pct_girls, ~name,
  1940, 0.43, "Jessie",
  1940, 0.5, "Jessie",
  1949, 0.4956897, "Jessie"
)

jessie_labels <- tribble(
  ~year, ~name, ~pct_girls, ~label,
  1998, "Jessie", 0.8, "BOYS",
  1998, "Jessie", 0.2, "GIRLS"
)
```

Step 1D: Draw the plot for "Jessie"

HIDE

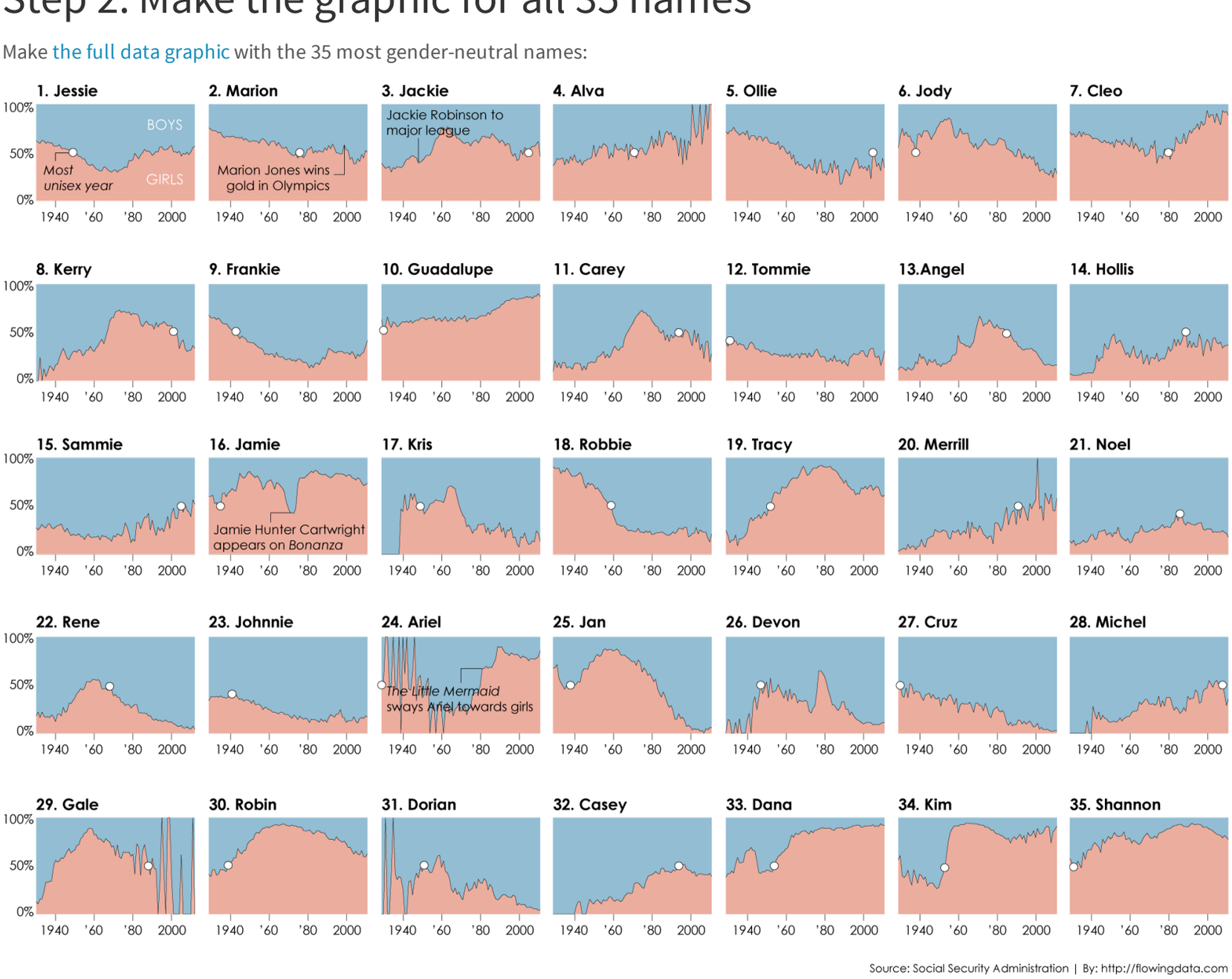
HIDE

```
ggplot(jessie, aes(x = year, y = pct_girls)) +
  geom_line() +
  geom_area(fill = "#eaac9e") +
  geom_point(data = jessie_unisex_year, fill = "white", pch = 21, size = 3) +
  geom_path(data = jessie_segments) +
  geom_text(
    data = jessie_labels,
    aes(label = label),
    color = "white"
  ) +
  geom_text(
    data = jessie_context, family = "Century Gothic",
    aes(x = year_label, y = vpos, label = text, hjust = hjust), vjust = "top"
  ) +
  scale_y_continuous(NULL,
    limits = c(0, 1),
    breaks = c(0, 0.5, 1),
    labels = scales::percent,
    expand = c(0,0)
  ) +
  scale_x_continuous(breaks = c(1940, 1960, 1980, 2000),
                     labels = c("1940", "'60", "'80", "2000"),
                     expand = c(0,0),
                     NULL) +
  scale_fill_manual(values = c("#eaac9e", "black")) +
  theme(
    panel.background = element_rect(fill = "#92bdd3"),
    axis.ticks.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
#    text = element_text(family = "Century Gothic"),
    strip.background = element_blank(),
    strip.text = element_text(hjust = 0, face = "bold", size = 14)
  ) +
  guides(fill = FALSE) +
  labs(
    title = "1. Jessie",
    caption = "Source: Social Security Administration | By http://flowingdata.com"
  )
```

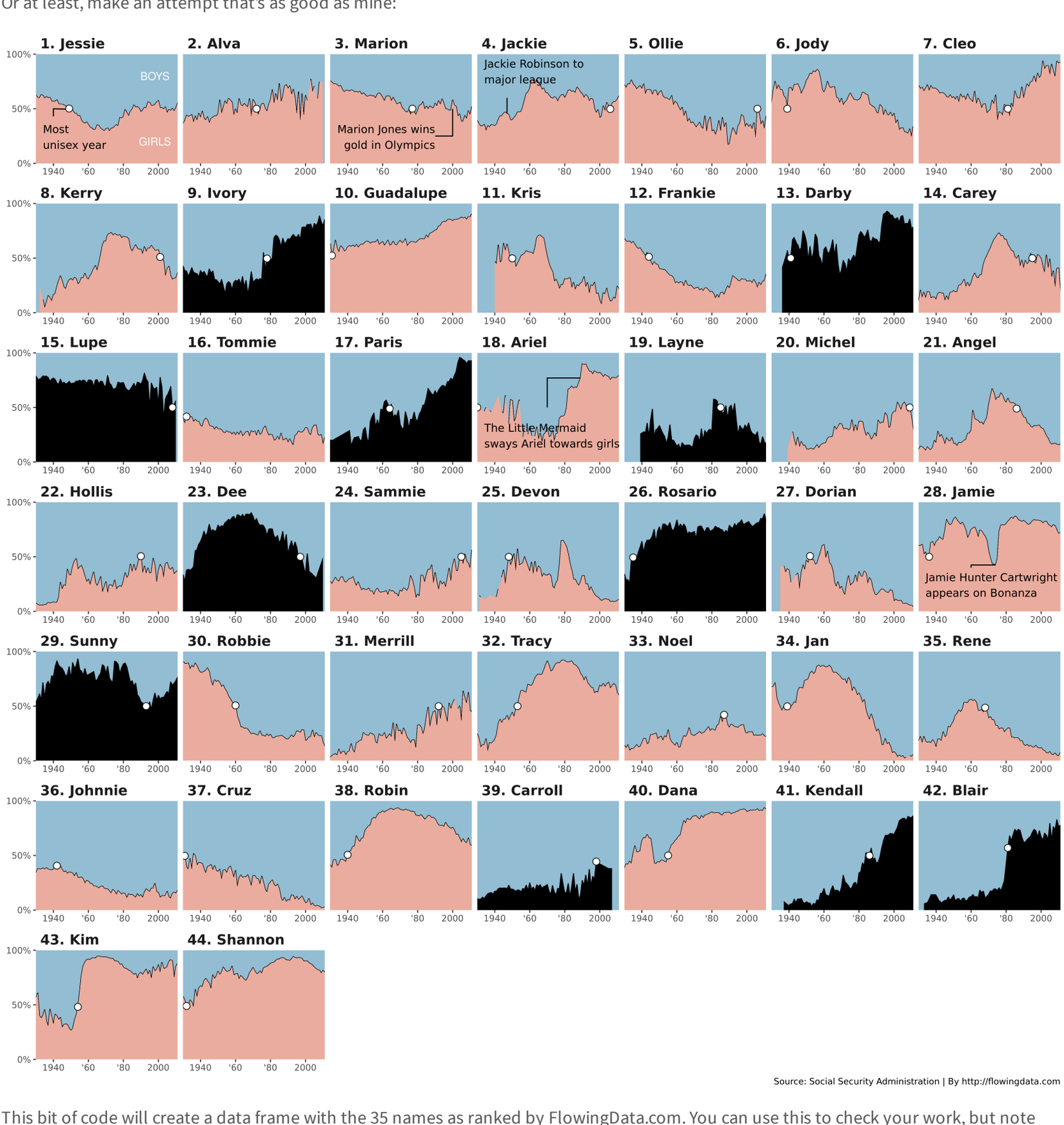


Step 2: Make the graphic for all 35 names

Make the full data graphic with the 35 most gender-neutral names:



Or at least, make an attempt that's as good as mine:



This bit of code will create a data frame with the 35 names as ranked by FlowingData.com. You can use this to check your work, but note that to meet the standard for *computing* the names, you need to discover these names algorithmically.

HIDE

HIDE

```
fd_names <- c(
  "Jessie", "Marion", "Jackie", "Alva", "Ollie",
  "Jody", "Cleo", "Kerry", "Frankie", "Guadalupe",
  "Carey", "Tommie", "Angel", "Hollis", "Sammie",
  "Jamie", "Kris", "Robbie", "Tracy", "Merrill",
  "Noel", "Rene", "Johnnie", "Ariel", "Jan",
  "Devon", "Cruz", "Michel", "Gale", "Robin",
  "Dorian", "Casey", "Dana", "Kim", "Shannon"
) %>%
  enframe(name = "fd_rank", value = "name")
```

Step 2A: Compute the RMSE for Jessie

HIDE

HIDE

```
jessie %>%
  mutate(
    error = pct_girls - 0.5,
    squared_error = error^2
  ) %>%
  summarize(
    mse = mean(squared_error),
    rmse = sqrt(mse)
  )
```

mse <dbl>	rmse <dbl>
0.009980733	0.09990362

1 row

Step 2B: Compute the RMSE for all names

HIDE

HIDE

```
#collect all baby data, filter and reshape it
all_babies <- babynames %>%
  filter( year >= 1930 & year < 2012) %>%
  select(-prop) %>%
  pivot_wider(names_from = sex, values_from = n) %>%
   mutate(
  pct_girls = F / (F + M))%>%
  filter(!is.na(F) & !is.na(M))
```

HIDE

HIDE

```
find_rmse <- function(x) {
  x %>%
    mutate(
           error = pct_girls - 0.5,
           squared_error = error^2
    ) %>%
    summarize(
      mse = mean(squared_error),
      rmse = sqrt(mse)
    )
}
```

HIDE

HIDE

```
all_babies %>%
  group_by(name)%>%
  group_modify(~find_rmse(.x))
```

Step 2C: Rank and filter the list of names

HIDE

HIDE

```
#finds 1000 most popular names
popular_names <- all_babies %>%
  group_by(name)%>%
  summarize(
    total_years = n(),
    total_occurences = sum(F+M)
  ) %>%
  filter(total_years >= 70 & name != "Unknown")%>%
  arrange(desc(total_occurences))%>%
 head(1000)
popular_names
```

name <chr>	total_years <int>	total_occurences <int>
Michael	82	4202264
James	82	4125512
Robert	82	3890601
John	82	3872824
David	82	3397125
William	82	2974869
Mary	78	2540721
Richard	81	2229191
Joseph	82	2028334
Christopher	74	1962547

1-10 of 269 rows   Previous 1 2 3 4 5 6 ... 27 Next

HIDE

HIDE

```
pop_rmse <- all_babies%>%
  inner_join(popular_names, by = "name")
pop_rmse
```

year <dbl>	name <chr>	F <int>	M <int>	pct_girls <dbl>	total_years <int>	total_occurences <int>
1930	Mary	64146	340	0.994727538	78	2540721
1930	Patricia	15752	52	0.996709694	73	1480993
1930	Joan	15480	68	0.995626447	82	430557
1930	Jean	11984	288	0.976531943	82	320636
1930	Elizabeth	10995	48	0.995653355	82	1219786
1930	Frances	10646	127	0.988211269	79	285830
1930	Evelyn	9536	50	0.994784060	74	253394
1930	Anna	9079	52	0.994305114	81	459306
1930	Nancy	9069	25	0.997250935	73	916855
1930	Catherine	6298	32	0.994944708	71	426077

1-10 of 10,000 rows   Previous 1 2 3 4 5 6 ... 1000 Next

HIDE

HIDE

```
most_unisex_names <- pop_rmse %>%
  group_by(name)%>%
  group_modify(~find_rmse(.x))%>%
  arrange((rmse))%>%
  head(35)
most_unisex_names
```

name <chr>	mse <dbl>	rmse <dbl>
Jessie	0.009980733	0.09990362
Alva	0.012205835	0.11048002
Marion	0.012376831	0.11125121
Carlin	0.013678273	0.11695415
Natividad	0.013696564	0.11703232
Michal	0.014960015	0.12231114
Jackie	0.016838970	0.12976506
Arie	0.018801023	0.13711682
Trinidad	0.020006373	0.14144388
Lorenza	0.020881748	0.14450518

1-10 of 35 rows   Previous 1 2 3 4 Next

Step 2D: Gather the data you need to draw the time series

HIDE

HIDE

```
data <- babynames %>%
  filter(
    year >= 1930 & year < 2012
  ) %>%
  select(-prop) %>%
  pivot_wider(names_from = sex, values_from = n) %>%
  mutate(pct_girls = F / (F + M)) %>%
  inner_join(most_unisex_names, by = c("name" = "name"))
```

Step 2E: Gather the data you need to draw the points

HIDE

HIDE

```
most_unisex_yr <- function(name_arg) {
  all_babies %>%
    filter(name == name_arg) %>%
    mutate(distance = abs(pct_girls - 0.5)) %>%
    arrange(distance) %>%
    head(1)
}

names_list <- most_unisex_names %>%
  select(-mse, -rmse) %>%
  deframe()

unisex_years <- map_dfr(names_list, most_unisex_yr)

unisex_years
```

year <dbl>	name <chr>	F <int>	M <int>	pct_girls <dbl>	distance <dbl>
1949	Jessie	1031	1023	0.5019474	0.001947420
1972	Alva	29	29	0.5000000	0.000000000
1977	Marion	229	228	0.5010941	0.001094092
1945	Carlin	9	9	0.5000000	0.000000000
1987	Natividad	15	15	0.5000000	0.000000000
1990	Michal	69	69	0.5000000	0.000000000
2006	Jackie	118	119	0.4978903	0.002109705
1960	Arie	11	11	0.5000000	0.000000000
1934	Trinidad	43	43	0.5000000	0.000000000
1983	Lorenza	22	22	0.5000000	0.000000000

1-10 of 35 rows   Previous 1 2 3 4 Next

Step 2F: Polish the data

HIDE

HIDE

```
all_babies <- all_babies %>% filter(name != "Unknown")

most_unisex_yr <- function(name_arg) {
  all_babies %>%
    filter(name == name_arg) %>%
    mutate(distance = abs(pct_girls - 0.5)) %>%
    arrange(distance) %>%
    head(1)
}

names_list <- most_unisex_names %>%
  select(-mse, -rmse) %>%
  deframe()

unisex_years <- map_dfr(names_list, most_unisex_yr)

unisex_years
```

year <dbl>	name <chr>	F <int>	M <int>	pct_girls <dbl>	distance <dbl>
1949	Jessie	1031	1023	0.5019474	0.001947420
1972	Alva	29	29	0.5000000	0.000000000
1977	Marion	229	228	0.5010941	0.001094092
1945	Carlin	9	9	0.5000000	0.000000000
1987	Natividad	15	15	0.5000000	0.000000000
1990	Michal	69	69	0.5000000	0.000000000
2006	Jackie	118	119	0.4978903	0.002109705
1960	Arie	11	11	0.5000000	0.000000000
1934	Trinidad	43	43	0.5000000	0.000000000
1983	Lorenza	22	22	0.5000000	0.000000000

1-10 of 35 rows   Previous 1 2 3 4 Next

Step 2G: Create the annotations

HIDE

HIDE

```
map_dfr(c("Jessie", "Marion", "Jackie", "Ariel", "Jamie"), most_unisex_yr)
```

year <dbl>	name <chr>	F <int>	M <int>	pct_girls <dbl>	distance <dbl>
1949	Jessie	1031	1023	0.5019474	0.001947420
1977	Marion	229	228	0.5010941	0.001094092
2006	Jackie	118	119	0.4978903	0.002109705
1930	Ariel	8	8	0.5000000	0.000000000
1936	Jamie	49	49	0.5000000	0.000000000

5 rows

HIDE

HIDE

```
general_context <- tribble(
  ~year_label, ~vpos, ~hjust, ~name, ~text,
  1934, 0.35, "left", "Jessie", "Most\nunisex year",
  1977, 0.35, "right", "Marion", "Marion Jones wins\ngold in Olympics",
  2006, 0.35, "top", "Jackie", "Jackie Robinson to\nmajor league",
  1930, 0.35, "right", "Ariel", "The Little Mermaid\nsways Ariel towards girls",
  1936, 0.35, "top", "Jamie", "Jamie Hunter Cartwright\nappears on Bonanza"
)

general_segments <- tribble(
  ~year, ~pct_girls, ~name,
  1940, 0.43, "Jessie",
  1940, 0.5, "Jessie",
  1949, 0.4956897, "Jessie",
  1940, 0.23, "Marion",
  1940, 0.5, "Marion",
  1977, 0.5, "Marion",
  1980, 0.33, "Jackie",
  1980, 0.498, "Jackie",
  2006, 0.498, "Jackie",
  1923, 0.23, "Ariel",
  1923, 0.498, "Ariel",
  1930, 0.498, "Ariel",
  1928, 0.23, "Jamie",
  1936, 0.5, "Jamie",
  1936, 0.5, "Jamie"
)
```

Step 2H: Order the facets

HIDE

HIDE

```
ranked_names <- most_unisex_names %>%
  mutate(
        fct_rmse = factor(rmse),
        name_rank= dense_rank(fct_rmse),
        name_label = paste(name_rank, name, sep = ".")
    )
ranked_names
```

name <chr>	mse <dbl>	rmse <dbl>	fct_rmse <fct>	name_rank <int>	name_label <chr>
Jessie	0.009980733	0.09990362	0.099903617034947	1	1.Jessie
Alva	0.012205835	0.11048002	0.11048002267235	1	1.Alva
Marion	0.012376831	0.11125121	0.111251205087006	1	1.Marion
Carlin	0.013678273	0.11695415	0.116954149907021	1	1.Carlin
Natividad	0.013696564	0.11703232	0.117032321402407	1	1.Natividad
Michal	0.014960015	0.12231114	0.122311139525075	1	1.Michal
Jackie	0.016838970	0.12976506	0.129765055294801	1	1.Jackie
Arie	0.018801023	0.13711682	0.137116821925788	1	1.Arie
Trinidad	0.020006373	0.14144388	0.141443884758966	1	1.Trinidad
Lorenza	0.020881748	0.14450518	0.144505183512645	1	1.Lorenza

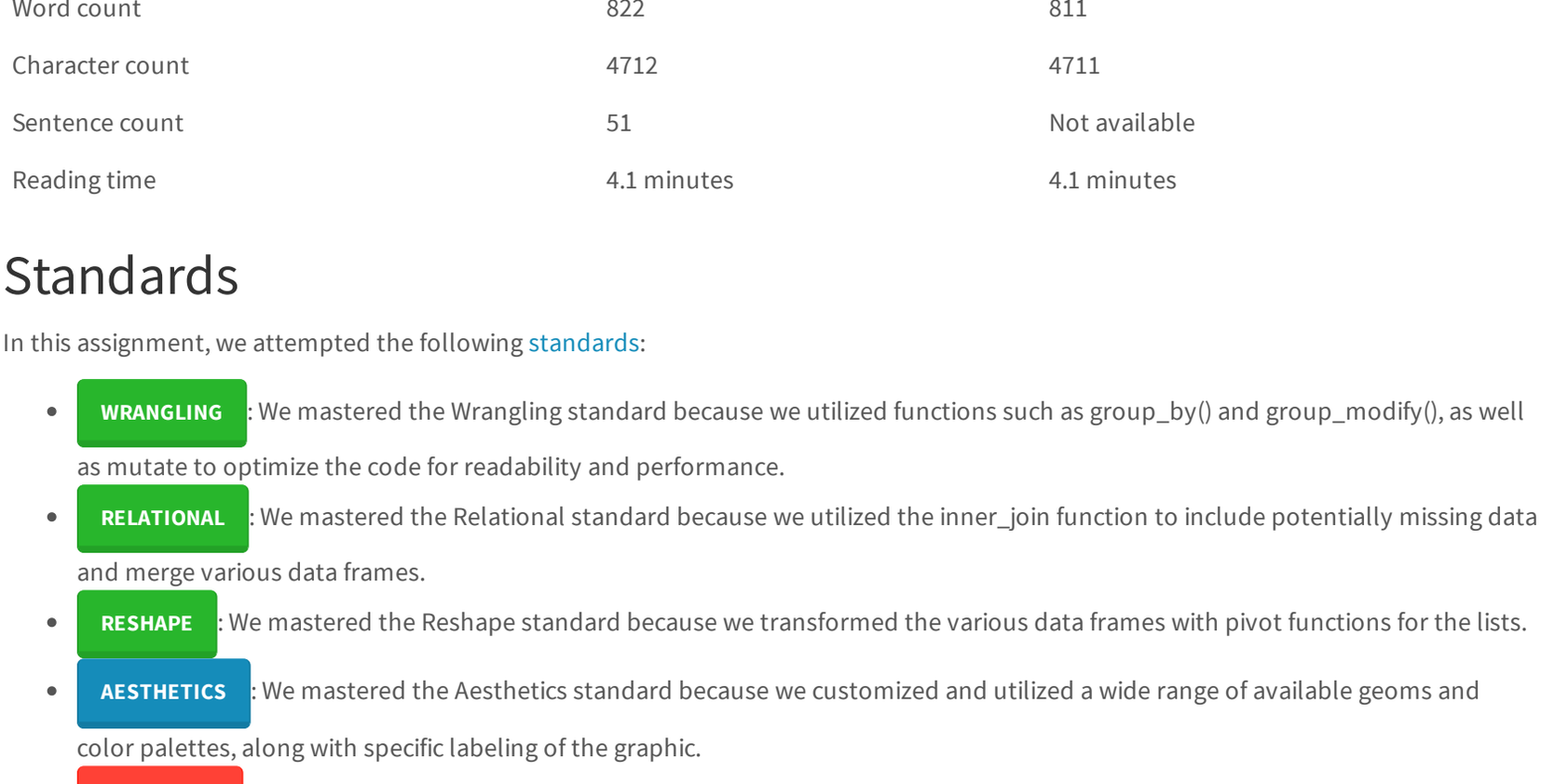
1-10 of 35 rows   Previous 1 2 3 4 Next

Step 2I: Draw the plot

HIDE

HIDE

```
ggplot(data, aes(x = year, y = pct_girls)) +
  geom_line() +
  geom_area(fill = "#eaac9e") +
  facet_wrap(~name, scales='free_x', ncol = 7) +
  geom_point(data = unisex_years, fill = "white", pch = 21, size = 2.8) +
  geom_text(
    data = jessie_labels,
    aes(label = label),
    color = "white"
  ) +
  geom_text(
  data = general_context, family = "Century Gothic",
    aes(x = year_label, y = vpos, label = text, hjust = hjust),
    vjust = "top",size=2
  ) +
  scale_y_continuous(NULL,
    limits = c(0, 1),
    breaks = c(0, 0.5, 1),
    labels = scales::percent,
    expand = c(0,0)
  ) +
    geom_path(data = general_segments, aes(x = year, y = pct_girls)) +
  scale_x_continuous(breaks = c(1940, 1960, 1980, 2000),
                     labels = c("1940", "'60", "'80", "2000"),
                     expand = c(0,0),
                     NULL
                     ) +
  scale_fill_manual(values = c("#eaac9e", "black")) +
  theme(
    panel.background = element_rect(fill = "#92bdd3"),
    axis.ticks.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    strip.background = element_blank(),
    strip.text = element_text(hjust = 0, face = "bold", size = 14)
  ) +
  guides(fill = FALSE) +
  labs(
    caption = "Source: Social Security Administration | By http://flowingdata.com"
  )
```



Word count

Method	koRpus	stringi
Word count	822	811
Character count	4712	4711
Sentence count	51	Not available
Reading time	4.1 minutes	4.1 minutes

Standards

In this assignment, we attempted the following standards:

- WRANGLING: We mastered the Wrangling standard because we utilized functions such as group_by() and group_modify(), as well as mutate to optimize the code for readability and performance.
- RELATIONAL: We mastered the Relational standard because we utilized the inner_join function to include potentially missing data and merge various data frames.
- RESHAPE: We mastered the Reshape standard because we transformed the various data frames with pivot functions for the lists.
- AESTHETICS: We mastered the Aesthetics standard because we customized and utilized a wide range of available geoms and color palettes, along with specific labeling of the graphic.
- MARKDOWN: We mastered the R Markdown standard because we employed formatting within the Markdown file to include a variety of code chunks, text, and graphics.

References

-http://flowingdata.com