

Popular Names

Over the years and decades, names come and go in popularity. BabyNames provides a lot of information about this, but it is not in glyph-ready form.

Task:

Create a graph (like Figure A.1) showing the ups and downs in the popularity of names of interest to you. using names of interest to you. The raw material you have is the BabyNames data table in the DataComputing package.

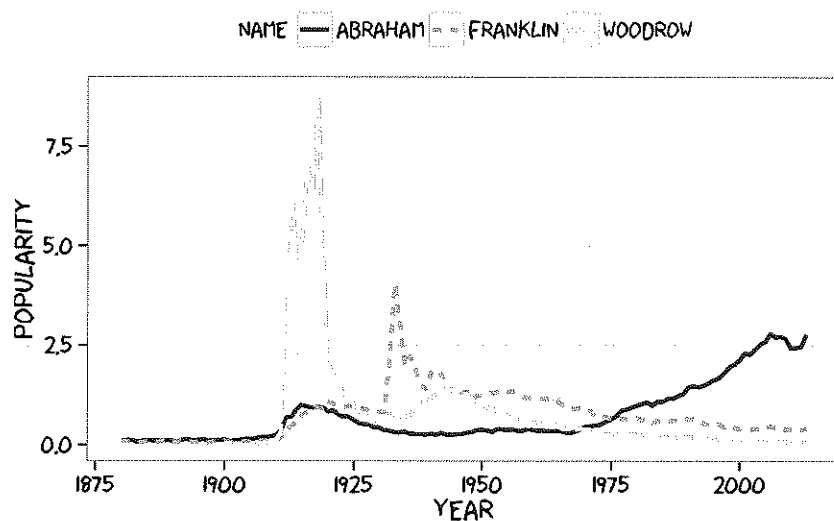


Figure A.1: A sketch of the popularity over time of a few names.

Developing a plan for wrangling

Step 1: Examine the data you have at hand — for this project, the data are the one table `BabyNames` — to find out what variables are available and what is the meaning of a case.

Step 2: Imagine what your end report will look like and sketch out your idea. Here, Figure A.1 will serve as the sketch of the goal.

Step 3: Analyze the graphic to figure out what a glyph-ready data table should look like. Mostly, this involves figuring out what variables are represented in the graph. Write down a small example of a glyph-ready data frame that you think could be used to make something in the form of the graphic.

- What variable(s) from the raw data table do not appear at all in the graph?
- What variable(s) in the graph are similar to corresponding variables in the raw data table, but might have been transformed in some way?

Step 4: Consider how the cases differ between the raw input and the glyph-ready table.

- Have cases been **filtered** out?
- Have cases been grouped and summarized within groups in any way?
- Have any new variables been introduced? If so, what's the relationship between the new variables and existing variables?

Step 5: Using English, write down a sequence of steps that will accomplish the wrangling from the raw data table to your hypothesized glyph-ready data table.

Step 6: Using paper and pen, translate your design, step by step, into R.

Step 7: Implement, test, and revise. Place your commands from Step (6) into a suitable document. Run them in R.

It's usual, particularly for beginners, but even for experts, that the wrangling sequence as first written won't work exactly as you anticipated. Among other reasons, this might be due to syntax errors in your R commands, or mis-matches when referring to variables, or because you unintentionally left out some or another data wrangling action.

Expect to follow a cycle of testing, diagnosing problems, revising, testing again, diagnosing, revising again, One effective strategy is divide and conquer; examine the output from early wrangling actions before adding in the actions to follow.

Once you have your *glyph-ready data*, make your graphic. You may find this template `ggplot()` command useful:

```
GlyphReadyForm %>%  
  ggplot(aes(x = year, y = total, group = name)) +  
  geom_line( size = 1, alpha = 0.5, aes(color = name)) +  
  ylab("Popularity") + xlab("Year")
```