**Dataframe schemas:**

Note dataframes have only an arbitrary ordinal index. Indexes and multi-indexes are added later where needed.

yob = a dataframe with each record comprising a unique name, sex and year. Length: approx. 1.76 million records; pickle = ~100 MB

name String

sex M or F

births Number of birth with that name of that sex during that year;

names with fewer than 5 births in a given year are omitted due to privacy concerns

year 1880-2016

pct Percentage of births of that sex during that year with that name (float)

ranked Rank of number of births of that name among all births of that sex during that year

names = a dataframe with each record comprising a unique name and sex, with data for individual years discarded but summary and additional data added. Length: approx. 101,000 records; pickle = ~ 7 MB

name Same as in df

sex Same as in df

year\_count Number of different years in which that name appears in dataframe, from 1 to 133.

year\_min First year name appears in database

year\_max Last year name appears in database

pct\_sum Sum of pct field for that name for all years. Not a statistically meaningful number

(because the underlying distribution of names varies from year to year),

but I have found it a useful rough metric during development

pct\_max Maximum value in pct field for all years, indicating the most popular that name has ever been in the database.

years = a dataframe with each record comprising a unique year, with individual name data discarded but summary and additional data added. Length: 133 records; pickle = ~ 8 kB

year Same as in df

births\_f Number of female births during that year

births\_m Number of male births during that year

births\_t Total number of births during that year

new\_names Number of names that appear for the first time during that year

unique\_names Number of different names that appear during that year

sexratio Number of boys born per hundred girls

# Project Objective:

Our project will examine trends of baby names over the past century. We’ll examine change in gender of names over time; difference in popularity of names by state/region compared to national; influence of the “famous people” (e.g., sports, entertainment, or any other trigger, etc.) on names; ethnicity/origin changes over time by state/region; and one-time names that pop up (minimum 5 for our dataset) with an investigation of reason for that name selection. We will use an existing dataset for names by state and time. We will attempt to acquire a dataset that identifies ethnicity of first names. And we will attempt to identify correlation between popular celebrities and increase in name popularity.

If time allows we will also look at fun presentation methods and other usages of the data (e.g., utilizing a twitter bot).

Discovery Phase:

* Researched multiple websites like Kaggle.com, DataWorld.com, Google Cloud for datasets which are source for US baby names data. Every source we found was sourcing data from US Social Security Administration(SSA) website.
* While researching data we found that SSA provides two data files one by name and year and second one with name, year and state.

Requirements Phase:

* Breaking the project objective into workable requirements for the team members to work on work distributed objectives towards the single goal of achieving the objectives within the limitation of time and resources.
* Collecting latest raw data from US SSA website.
* Best practices for working on raw data files and creating clean data sets for the baby names based on the various ways like name, year of birth, sex, number of births and state of birth.
* Setting the analysis objective for each team member based on defined objectives, choice and interest.

Design and Coding phase:

* Programmatically downloading the raw data files for US Social Security Administration(SSA) website.
* Defining the data schema for the dataframes:

1. yob = a dataframe with each record comprising a unique name, sex and year. Length: approx. 1.76 million records; pickle = ~100 MB

* name String
* sex M or F
* births Number of birth with that name of that sex during that year;
  + - * names with fewer than 5 births in a given year are omitted due to
      * privacy concerns
* year 1880-2017
* pct Percentage of births of that sex during that year with that name (float)
* ranked Rank of number of births of that name among all births of that sex during that
  + year

1. names = a dataframe with each record comprising a unique name and sex, with data for individual years discarded but summary and additional data added. Length: approx. 101,000 records; pickle = ~ 7 MB

* name Same as in df
* sex Same as in df
* year\_count Number of different years in which that name appears in dataframe, from 1 to 133.
* year\_min First year name appears in database
* year\_max Last year name appears in database
* pct\_sum Sum of pct field for that name for all years. Not a statistically meaningful number
  + - (because the underlying distribution of names varies from year to year),
    - but I have found it a useful rough metric during development
* pct\_max Maximum value in pct field for all years, indicating the most popular that name has ever been in the database.

1. years = a dataframe with each record comprising a unique year, with individual name data discarded but summary and additional data added. Length: 136 records; pickle = ~ 8 kB

* year Same as in df
* births\_f Number of female births during that year
* births\_m Number of male births during that year
* births\_t Total number of births during that year
* new\_names Number of names that appear for the first time during that year
* unique\_names Number of different names that appear during that year
* sexratio Number of boys born per hundred girls
* Steps to create the data frame:
* Read individual files, yob1880.txt, yob1881.txt, etc. and assemble into a dataframe
* Add column 'pct': the number of births of that name and sex in that year divided by the total number of births of that sex in that year, multiplied by 100 to turn into a percentage and reduce leading zeroes.
* Add rank of each name each year each sex.
* Names dataframe: discards individual birth or pct values, and instead collects data on unique names. There is one row per unique combination of name and sex.
* Create years dataframe. Discards individual name data, aggregating by year