Aggregation of Name Data

January 20, 2021

1 Introduction

Names are an integral part of our society as it gives an identifier to not only us humans but to all things around us in existence. We all have given names at birth determined by possibly many different factors in our parents' lives such as history, tradition, religion, and beliefs. By studying the trends and naming habits in the United States over the course of many years, we can hope to learn/make conclusions about the history and happenings country.

One of the useful sets of data that we have to help us make conclusions is a list of names along with the gender and frequency associated with that given name for each year starting from 1880 to 2018. This span of 139 years hold an abundant amount of data for us to manipulate and use to find some interesting tidbits of information about how our country has changed.

One of those interesting bits of information is the change in the trend of given names across those 139 years. One could observe that the frequency of given names having biblical origins being lower and lower with each passing year.

2 Importing Files

We can start by using the glob function to import all of the files that hold the name data. We are also importing the numpy library to use later.

[6]:
```
from glob import glob
#imports the glob module from the glob library to grab files with the name␣
 ↪pattern of the files

import numpy as np
#imports the numpy library

files = sorted(glob("names/yob*.txt"))
```

3 Creating the Dictionary

We then will need to identify the first,last, and number of years we are working with from the data set.

[7]:
```
names = {} #create an empty dictionary in the form of {names:{gender:count}}
firstyear = 1880
lastyear = 2018
```

```
nyears = lastyear - firstyear + 1
```

Now we can read the files line by line and split the data seperated by commas to obtain purely the values we want. We then create entries in the dictionary.

[9]:
```
for file in files:
    year = int(file[-8:-4]) # for each file we will extract the year from the␣
 ↪file name
    with open(file) as f:
        lines = f.read().split('\n')
    lines = [line for line in lines if len(line)>2]
    for line in lines:
        name,gender,count = line.split(',')
        if name not in names:
            # create a new default entry in d for name an entry for the name␣
 ↪doesn't already exist
            names[name] = {"F": np.zeros(nyears,dtype = int), "M" : np.
 ↪zeros(nyears, dtype = int)}
        names[name][gender][year-firstyear] = int(count)
```

4 Scraping the Web for Names

After establishing our dictionary made of our data set, we now need a set of biblical names to compare bring up the data of from the dictionary "names". This can be done by searching for a list of biblical names and scraping the website for the data. For this instance I chose two websites which had a list of 100 biblical names each for each gender. Another option would be to scrape all the proper nouns out of the bible itself.

We need to import the requests library to help us access the website with the names. Then turn the aquired response into a large string of text made of the website's html. Because a websites html has an abundance of text that we do not need, we need to split the text by certain strings.

[96]:
```
import requests
urlB = "https://www.everydayknow.com/biblical-boy-names/"
rB = requests.get(urlB) #get the request from the website
new = rB.text.split("<p><strong>")[2:] #turn it into a string and split by a␣
 ↪certain string

print("\n Here we see a portion of difficult to read unusuable information that␣
 ↪we do not need with valuable information mixed in.\n")
print("[\n\n\n" + new[1][0:50] + "\n\n\n]")
```

```
 Here we see a portion of difficult to read unusuable information that we do not
need with valuable information mixed in.

[
2. James</strong></p>
<p>This Hebrew name appears


]
```

The html we requested and converted to text has been filtered out slightly but we still need to filter out some more to have a completely pure list of just names.

[47]:
```
count = 0
names100B = []
for i in new:
    names100B.append(i.split("<")[0].split(". ")[1]) #split by the "<"␣
 ↪character then by the ". " character
    count += 1
    if count > 99:
        break

print("Heres a sample of our list of biblical boy names \n")
for i in range(5):
    print(str(i+1) + "." + names100B[i] + "\n")
```

```
Heres a sample of our list of biblical boy names

1.Ezra

2.James

3.Jordan

4.Lot

5.Omar
```

We need to now do the same for girl names

[95]:
```
names100G = []
count = 0
urlG = "https://foreverymom.com/maternity/
 ↪baby-girl-names-meanings-diy-baby-name-wall-art-2/"
rG = requests.get(urlG)
x = '<h3><strong><a data-mil="51810" href="https://foreverymom.com/
 ↪family-parenting/babies-toddlers/
 ↪baby-girl-names-meanings-diy-baby-name-wall-art/'
newG = rG.text.split(x)
for i in newG:
    names100G.append(i.split(">")[1].split("<")[0]) #split by the ">" character␣
 ↪then by the "<" character
```

```
names100G = names100G[1:]

print("Heres a sample of our list of biblical girl names \n")
for i in range(5):
    print(str(i+1) + "." + names100G[i] + "\n")
```

```
Heres a sample of our list of biblical girl names

1.Emma

2.Sophia

3.Olivia

4.Isabella

5.Ava
```

These blocks of code are just to get the sum of males and females in a given year in the form of a list with each index representing a year. This is done for both males and females. We can then put them together in a new list for the total of people in each year.

[54]:
```
nOb = [0] * 139
for name in names:
    for i in range(139):
        nOb[i] += names[name]["M"][i]

nOg = [0] * 139
for name in names:
    for i in range(139):
        nOg[i] += names[name]["F"][i]

nOp = []
for i in range(139):
    nOp.append(nOg[i] + nOb[i])
```

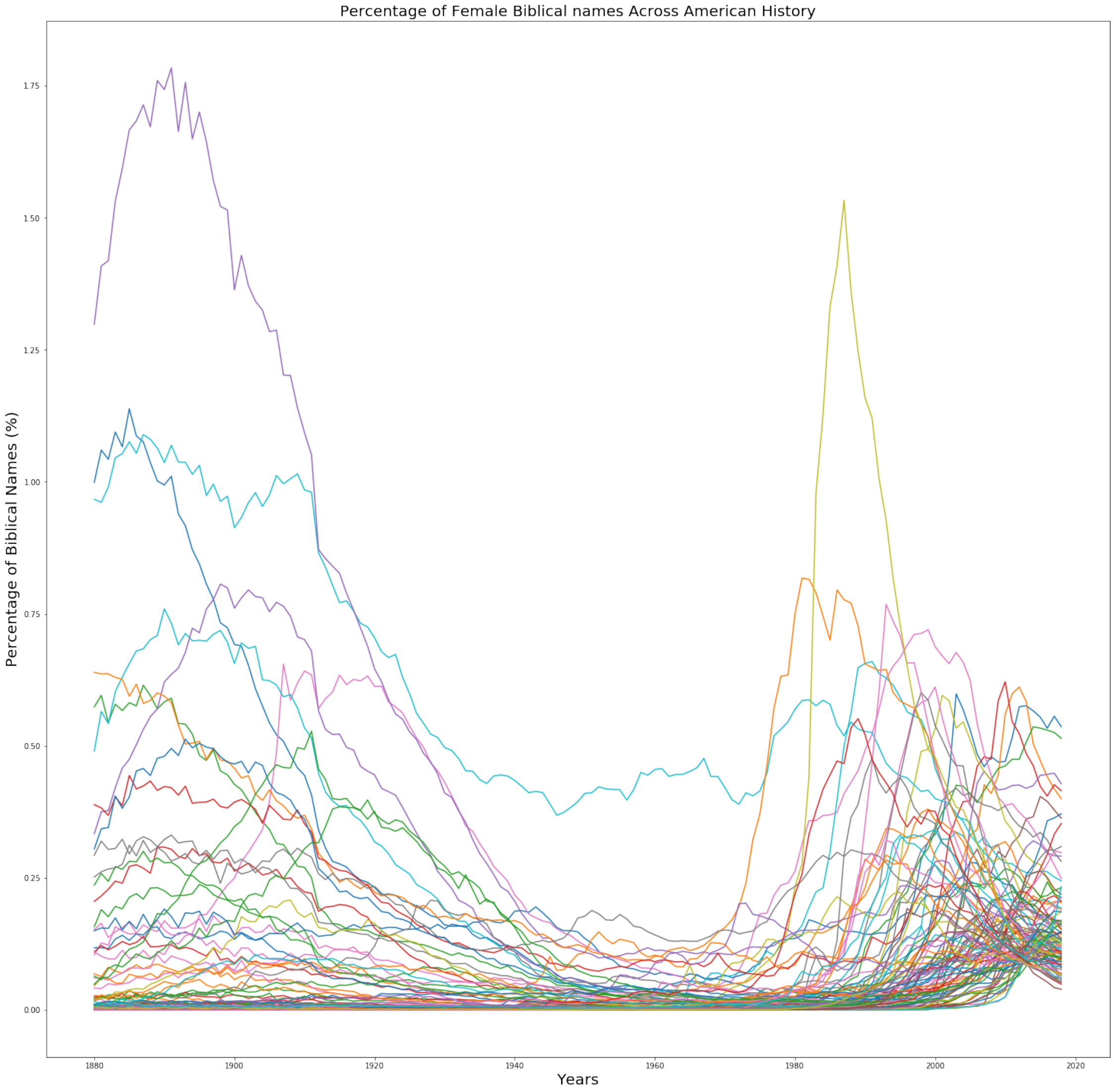
Importing the matplotlib library will allow us to start to plot the data that we have aquired to have a visual representation of our data. Creating a graph with the ranges of years from the files as the x-axis and the percentage of the total number of people actually has a biblical name in context of girl biblical names.

[92]:
```
import matplotlib.pyplot as plt



plt.figure(figsize = (25,25))
for i in names100G:
    plt.plot(range(firstyear,lastyear + 1), (names[i]["M"] + names[i]["F"])/nOp␣
 ↪* 100, alpha = 1,markersize = 3) # plot data
```

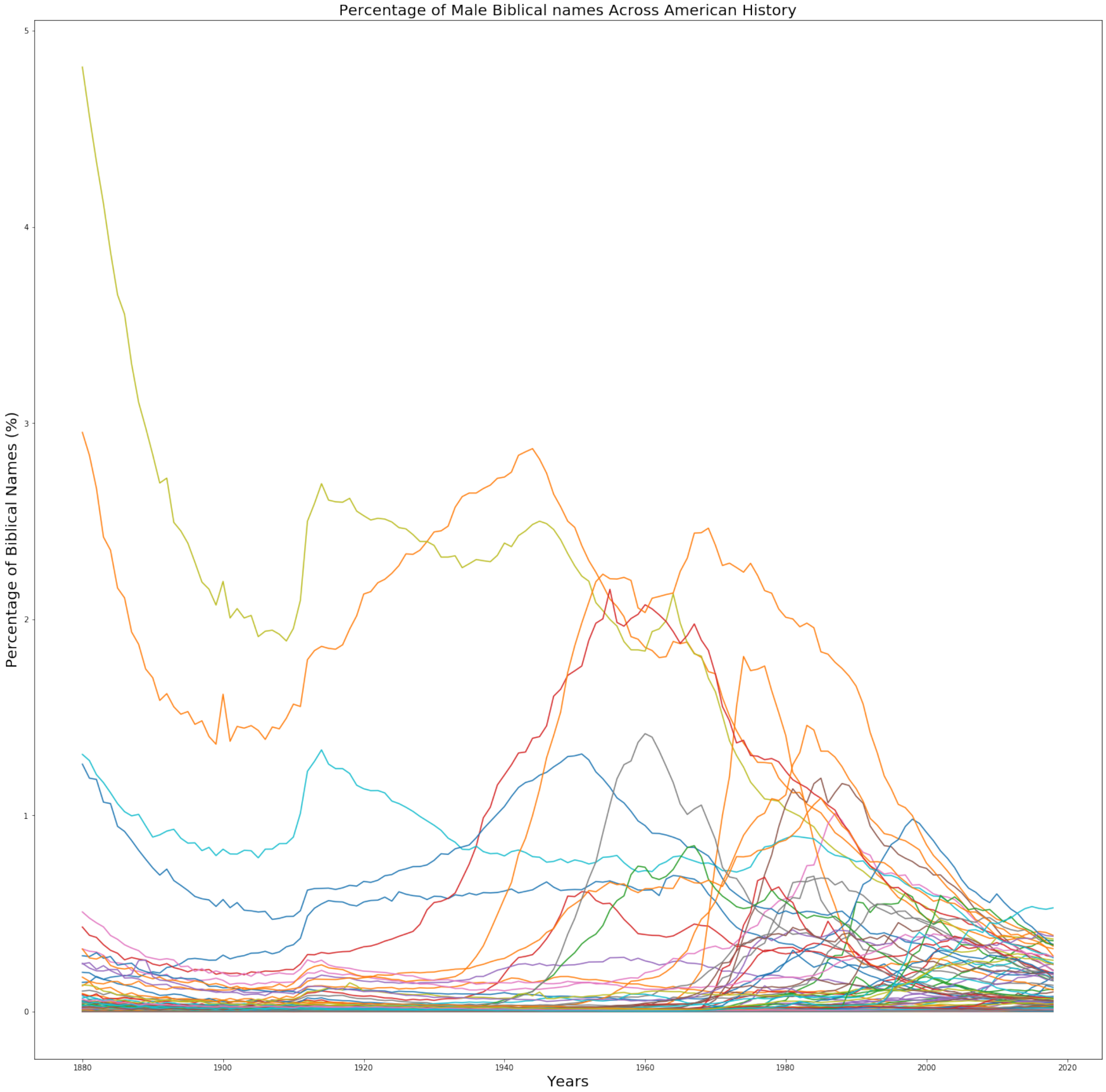
```
plt.title("Percentage of Female Biblical names Across American History",␣
 ↪fontsize = 20) #title
plt.xlabel("Years", fontsize = 20) # x label
plt.ylabel("Percentage of Biblical Names (%)", fontsize = 20) # y label
plt.show()
```



This is done the same way for the list of male names.

[94]:
```
plt.figure(figsize = (25,25))
for i in names100B:
    plt.plot(range(firstyear,lastyear + 1), (names[i]["M"] + names[i]["F"])/nOp␣
 ↪* 100, alpha = 1,markersize = 3) # plot data
plt.title("Percentage of Male Biblical names Across American History", fontsize␣
 ↪= 20) #title
```

```
plt.xlabel("Years", fontsize = 20) # x label
plt.ylabel("Percentage of Biblical Names (%)", fontsize = 20) # y label
plt.show()
```



We can see here that that many of the popular biblical names were higher in percentage in the very beginning of the time range in 1880. As time passed the percentage of biblical names started to decrease gradually with a few certain names making a resurgence in significant moments in history.

Interesting that the patterns for male and female are quite different. A logarithmic vertical scale would unsquish all those curves at the bottom of your plots.

5 Conclusion

With the increase in population in native and foreign people, as time goes on, we can expect a departure from classic biblical names. (Native and foreign? What else is there?) With the population in 1880 being composed of more religious in its entirely. As time went on and more immigrants arrived in the country, the more culture specific and non-traditional names started to appear in the data. In history those new immigrants had a desire to keep their own non-traditional given names which may explain the decline in biblical names.

I don't think you should ignore in your Conclusion the fact that biblical names seem to have become more popular again in recent decades.

I'd like to see a little deeper analysis: for example what is the *total fraction* of males and of females with biblical names - over time.

5.0.1 Sources

https://en.wikipedia.org/wiki/Demographic_history_of_the_United_States#Immigration_1850_to_1965 https://en.wikipedia.org/wiki/Naming_in_the_United_States https://foreverymom.com/maternity/baby-girl-names-meanings-diy-baby-name-wall-art-2/ https://www.everydayknow.com/biblical-boy-names/