Rajeev_Trendy_Names

February 10, 2018

1 Determine "TRENDY" names

- There are several ways one could find "trendy" names, i.e. ones that rise quickly from obscurity to popularity, then fade quickly. One could look at percentage changes, but others have already done this, and one would have to use a second algorithm to distinguish truly popular names
- This method is borrowed from peak analysis in chemistry chromatography; it weighs the results in favour of the most popular names with the sharpest peaks. The names with the lowest ratio of maximum popularity (normalized frequency, i.e. peak height) to peak width a a certain percentage (e.g. 10%) of peak height are the most "spiky".

1.0.1 Load dataframes; run this before every subsequent section

```
In [7]: save_path = "rajeev_data/trendy_names" # files created by this notebook will be saved
        import time

        import os
        if not os.path.isdir(save_path): # creates path if it does not exist
            os.makedirs(save_path)

        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        %run Rajeev_download_process_files.py

        last_year = years.year.max()
```

```
Data already downloaded.
Data already extracted.
Processing.
Tail of dataframe 'yob':

Tail of dataframe 'names':

Tail of dataframe 'years':
Tail of dataframe 'yob1900':
```

```
Tail of dataframe 'names1900':
Tail of dataframe 'years1900':
```

1.0.2 Find largest ratios of peak height to width at 10 percent height for 1000 most popular names of each sex

Low-popularity names are not included because their low signal-to-noise ratio would create false positives

In [8]:
```
# create dataframes of most popular names

top_cutoff = 1000 #consider only this number of most popular names of each sex;
                  #it saves calculation time and does not change final result

peak_height_cutoff = 0.1  # ten percent of peak height

###

start = time.time()

dfnamesm = names[names.sex == 'M']
dfnamesf = names[names.sex == 'F']
dfnamesm = dfnamesm.sort_values(by='pct_sum', ascending=False)
dfnamesf = dfnamesf.sort_values(by='pct_sum', ascending=False)
dfnamesm = dfnamesm[:top_cutoff]
dfnamesf = dfnamesf[:top_cutoff]

# create dataframe of peak analyses

dfresult = pd.DataFrame()

print("Countdown F, then M to zero: ",)

sx = 'F'
print(sx,)
total = len(dfnamesf)
for nm in list(dfnamesf.name):
    total -= 1
    if total % 100 == 0: print(total,)
    df = yob[(yob.name == nm) & (yob.sex == sx)]
    pct_sum = names[(names.name == nm) & (names.sex == sx)].pct_sum.iloc[0]
    year_count = names[(names.name == nm) & (names.sex == sx)].year_count.iloc[0]
    year_min = names[(names.name == nm) & (names.sex == sx)].year_min.iloc[0]
    year_max = names[(names.name == nm) & (names.sex == sx)].year_max.iloc[0]
    pct_max = df.pct.max()
    df = df.sort_values(by='year')
    yrcutstart = 0
```

```
    yrcutend= 0
    pctcut_sum = 0

    for idx, row in df.iterrows():
        currpct =df.pct[idx]
        curryr = df.year[idx]
        if currpct >= peak_height_cutoff*pct_max:
            pctcut_sum += currpct
            if yrcutstart == 0:
                yrcutstart = curryr
            yrcutend = curryr

    tail_front = yrcutstart - year_min
    tail_end = year_max - yrcutend

    yrcutspan = yrcutend-yrcutstart
    yrcutratio = 1.0*yrcutspan/year_count

    spikiness = pct_max / yrcutspan

    dfresult = dfresult.append(pd.DataFrame({'name':[nm],
                     'sex':[sx],
                     'year_count':[year_count],
                     'year_min':[year_min],
                     'year_max':[year_max],
                     'pct_max':[pct_max],
                     'pct_sum':[pct_sum],
                     'yrcutstart':[yrcutstart],
                     'yrcutend':[yrcutend],
                     'yrcutspan':[yrcutspan],
                     'yrcutratio':[yrcutratio],
                     'pctcut_sum':[pctcut_sum],
                     'tail_front':[tail_front],
                     'tail_end':[tail_end],
                     'spikiness':[spikiness] }))

sx = 'M'
print(sx,)
total = len(dfnamesm)
for nm in list(dfnamesm.name):
    total -= 1
    if total % 100 == 0: print(total,)
    df = yob[(yob.name == nm) & (yob.sex == sx)]
    pct_sum = names[(names.name == nm) & (names.sex == sx)].pct_sum.iloc[0]
    year_count = names[(names.name == nm) & (names.sex == sx)].year_count.iloc[0]
    year_min = names[(names.name == nm) & (names.sex == sx)].year_min.iloc[0]
    year_max = names[(names.name == nm) & (names.sex == sx)].year_max.iloc[0]
    pct_max = df.pct.max()
```

```
    df = df.sort_values(by='year')
    yrcutstart = 0
    yrcutend= 0
    pctcut_sum = 0

    for idx, row in df.iterrows():
        currpct =df.pct[idx]
        curryr = df.year[idx]
        if currpct >= peak_height_cutoff*pct_max:
            pctcut_sum += currpct
            if yrcutstart == 0:
                yrcutstart = curryr
            yrcutend = curryr

    tail_front = yrcutstart - year_min
    tail_end = year_max - yrcutend

    yrcutspan = yrcutend-yrcutstart
    yrcutratio = 1.0*yrcutspan/year_count

    spikiness = pct_max / yrcutspan

    dfresult = dfresult.append(pd.DataFrame({'name':[nm],
                     'sex':[sx],
                     'year_count':[year_count],
                     'year_min':[year_min],
                     'year_max':[year_max],
                     'pct_max':[pct_max],
                     'pct_sum':[pct_sum],
                     'yrcutstart':[yrcutstart],
                     'yrcutend':[yrcutend],
                     'yrcutspan':[yrcutspan],
                     'yrcutratio':[yrcutratio],
                     'pctcut_sum':[pctcut_sum],
                     'tail_front':[tail_front],
                     'tail_end':[tail_end],
                     'spikiness':[spikiness] }))

picklepath = save_path + 'trendiness_'+ str(int(100*peak_height_cutoff))+'.pickle'
csvpath = save_path + 'trendiness_'+ str(int(100*peak_height_cutoff))+'.csv'

dfresult = dfresult[(dfresult.tail_end != 0 ) & (dfresult.tail_front != 0)] # remove n
df.reset_index(drop=True, inplace=True)
dfresult.to_pickle(picklepath)
dfresult.to_csv(csvpath)

print('\nFiles saved.')
```

```
Countdown F, then M to zero:
F
900
800
700
600
500
400
300
200
100
0
M
900
800
700
600
500
400
300
200
100
0

Files saved.
```

```
In [9]: print(dfresult[dfresult.sex == 'F'].sort_values(by='spikiness', ascending=False).reset_
```

```
   index      name   pct_max    pct_sum  pctcut_sum sex  spikiness  tail_end  \
0      0     Linda  5.666702  86.555278   78.125872   F   0.182797        47
1      0  Brittany  2.050001  19.963709   18.403812   F   0.120588        16
2      0     Debra  2.585571  29.137515   26.780622   F   0.117526        44
3      0   Shirley  4.039516  53.522958   46.610027   F   0.112209        59
4      0    Ashley  3.155632  47.449055   44.545735   F   0.105188         6
5      0  Jennifer  4.301120  88.448228   82.479855   F   0.104905        14
6      0   Deborah  2.815900  40.420231   36.584367   F   0.104293        42
7      0      Lisa  3.414340  55.329716   52.214487   F   0.100422        27
8      0   Jessica  3.221220  60.657733   57.355505   F   0.092035        10
9      0     Betty  3.396358  84.950758   76.300854   F   0.069313        57

   tail_front  year_count  year_max  year_min  yrcutend  yrcutratio  \
0          58         137      2016      1880      1969    0.226277
1          20          54      2016      1963      2000    0.314815
2          36         102      2016      1914      1972    0.215686
3          41         136      2016      1880      1957    0.264706
4          63          78      2016      1917      2010    0.384615
5          45          99      2016      1916      2002    0.414141
```

```
6          67          137       2016       1880       1974    0.197080
7          69          113       2016       1886       1989    0.300885
8          91          137       2016       1880       2006    0.255474
9          30          137       2016       1880       1959    0.357664

   yrcutspan  yrcutstart
0         31        1938
1         17        1983
2         22        1950
3         36        1921
4         30        1980
5         41        1961
6         27        1947
7         34        1955
8         35        1971
9         49        1910
```

```
In [10]: print(dfresult[dfresult.sex == 'M'].sort_values(by='spikiness', ascending=False).rese
```

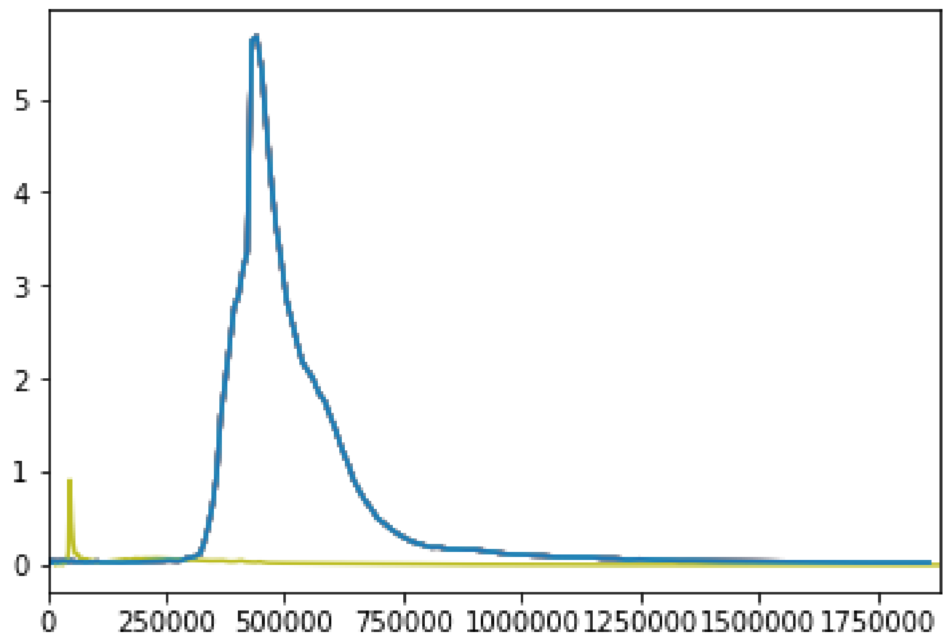
```
   index     name   pct_max     pct_sum  pctcut_sum sex  spikiness  tail_end  \
0      0    Dewey  0.908795    5.135351    2.104963   M   0.151466       113
1      0    Jason  3.481697   58.825162   54.843751   M   0.084919         7
2      0   Grover  0.712437    6.551928    3.901046   M   0.059370       121
3      0     Mark  2.754202   74.030527   65.988047   M   0.051966        17
4      0     Gary  2.026481   51.795390   48.723030   M   0.038971        31
5      0    Brian  2.290481   63.852754   61.003633   M   0.038822         7
6      0    Larry  1.909725   49.613461   44.996293   M   0.037446        34
7      0    Scott  1.747625   42.881843   39.137994   M   0.037184        19
8      0   Donald  2.945288  106.781089  100.561545   M   0.036362        33
9      0  Woodrow  0.455162    4.011130    2.757148   M   0.035012        92

   tail_front  year_count  year_max  year_min  yrcutend  yrcutratio  \
0          10         130      2016      1887      1903    0.046154
1          88         137      2016      1880      2009    0.299270
2           3         137      2016      1880      1895    0.087591
3          66         137      2016      1880      1999    0.386861
4          53         135      2016      1880      1985    0.385185
5          41         107      2016      1909      2009    0.551402
6          51         137      2016      1880      1982    0.372263
7          70         137      2016      1880      1997    0.343066
8          22         137      2016      1880      1983    0.591241
9          11         113      2016      1900      1924    0.115044

   yrcutspan  yrcutstart
0          6        1897
1         41        1968
2         12        1883
```

3	53	1946
4	52	1933
5	59	1950
6	51	1931
7	47	1950
8	81	1902
9	13	1911

1.1 Plot Female and Male names with highest spikiness

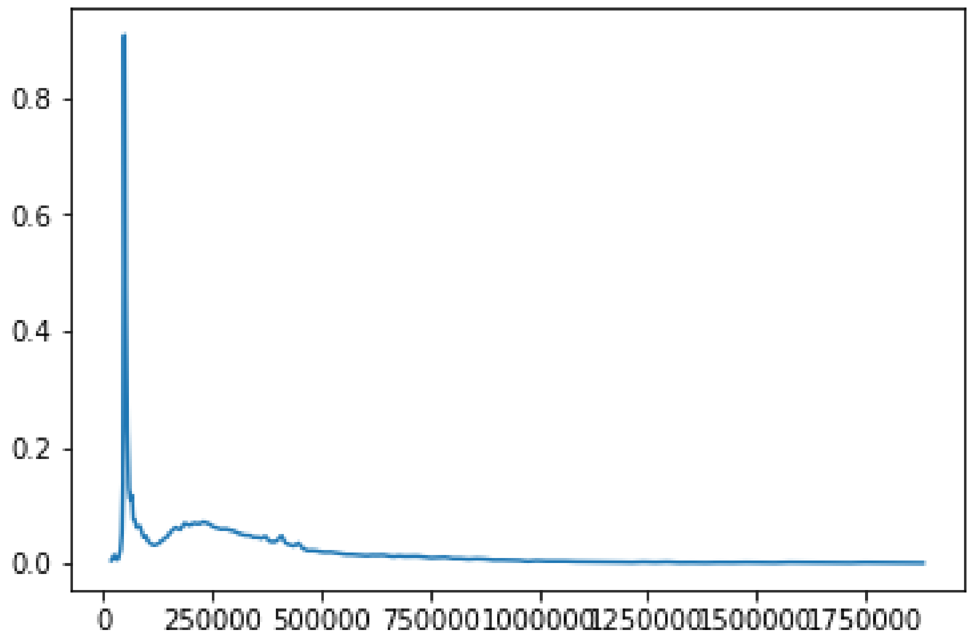
Female Name="Linda" and max spikiness

```
In [25]: plt.plot(yob[(yob.name=='Linda')&(yob.sex=='F')].sort_values(by='year')['pct'])
         plt.show()
```



Male Name="Dewey" and max spikiness

```
In [26]: plt.plot(yob[(yob.name=='Dewey')&(yob.sex=='M')].sort_values(by='year')['pct'])
         plt.show()
```


0.8
0.6
0.4
0.2
0.0
0
250000
500000
750000
1000000
1250000
1500000
1750000