STA6714 Case Study String Distances

Aaron Smith

4/8/2020

For a case study, let’s look at how different string distances perform when comparing baby names to the name Aaron. The data comes from the R package, babynames; which took its data from the Social Security Administration and the US Census.

# Preliminary time series

require(ggplot2)

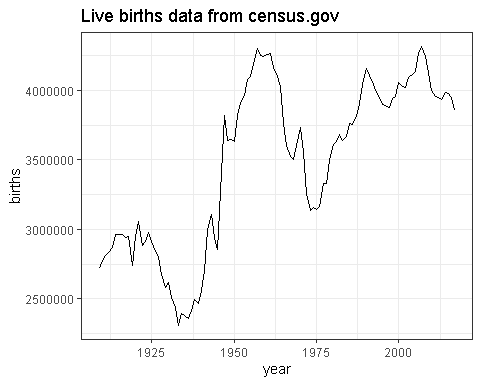
## Loading required package: ggplot2

First let’s look at the number of live births for each year.

babynames::births

## # A tibble: 109 x 2  
## year births  
## <int> <int>  
## 1 1909 2718000  
## 2 1910 2777000  
## 3 1911 2809000  
## 4 1912 2840000  
## 5 1913 2869000  
## 6 1914 2966000  
## 7 1915 2965000  
## 8 1916 2964000  
## 9 1917 2944000  
## 10 1918 2948000  
## # ... with 99 more rows

ggplot(babynames::births) +   
 aes(x = year,y = births) +   
 geom\_line() +   
 theme\_bw() +   
 ggtitle("Live births data from census.gov")

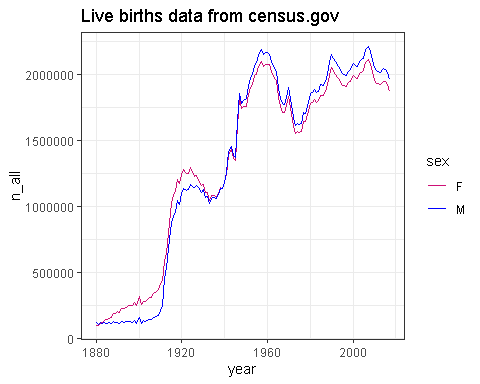


Now, let’s look at SSA name applications by gender.

babynames::applicants

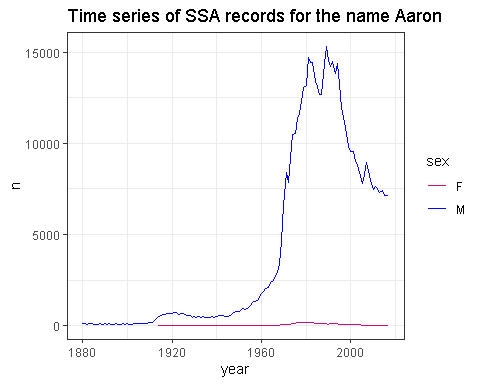
## # A tibble: 276 x 3  
## year sex n\_all  
## <int> <chr> <int>  
## 1 1880 F 97605  
## 2 1880 M 118400  
## 3 1881 F 98855  
## 4 1881 M 108282  
## 5 1882 F 115695  
## 6 1882 M 122031  
## 7 1883 F 120059  
## 8 1883 M 112477  
## 9 1884 F 137586  
## 10 1884 M 122738  
## # ... with 266 more rows

ggplot(babynames::applicants) +   
 aes(x = year,y = n\_all,group = sex,color = sex) +   
 geom\_line() +   
 theme\_bw() +   
 ggtitle("Live births data from census.gov") +   
 scale\_color\_manual(values = c("deeppink3","blue"))



Let’s see how common the name Aaron is by year and gender.

M <- as.data.frame(babynames::babynames)  
M\_Aaron <- M[M$name == "Aaron",]  
ggplot(M\_Aaron) +   
 aes(x = year,y = n,group = sex,color = sex) +   
 geom\_line() +   
 theme\_bw() +   
 ggtitle("Time series of SSA records for the name Aaron ") +   
 scale\_color\_manual(values = c("deeppink3","blue"))



# String distances

To get hands-on-experience with string distances, let’s compute the distance between “Aaron” and all other names in the data set.

v\_names <- sort(unique(babynames::babynames$name))  
v\_method <- c(  
 #"hamming",  
 "lcs","lv","osa","dl",  
 "qgram","jw","jaccard","cosine","soundex"  
)  
stringdist\_Aaron <- sapply(  
 X = v\_method,  
 FUN = function(x){  
 stringdist::stringdist(  
 a = "Aaron",  
 b = v\_names,  
 method = x  
 )  
 }  
)  
rownames(stringdist\_Aaron) <- v\_names  
colnames(stringdist\_Aaron) <- v\_method  
head(stringdist\_Aaron)

## lcs lv osa dl qgram jw jaccard cosine soundex  
## Aaban 4 2 2 2 4 0.2666667 0.5000000 0.3238766 1  
## Aabha 6 3 3 3 6 0.4000000 0.7142857 0.4929074 1  
## Aabid 6 3 3 3 6 0.4000000 0.7500000 0.6000000 1  
## Aabir 4 3 3 3 4 0.4000000 0.5714286 0.4000000 1  
## Aabriella 8 6 6 6 8 0.3555556 0.6666667 0.5038611 1  
## Aada 5 3 3 3 5 0.3666667 0.6666667 0.4522774 1

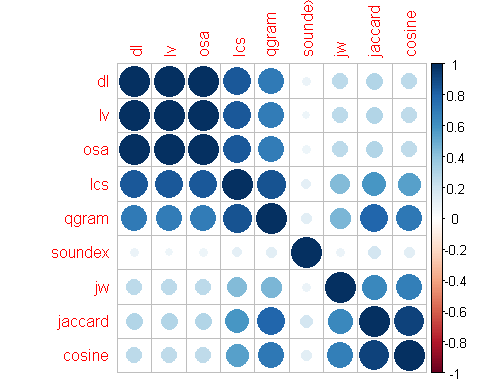
Let’s see which metrics give similar results. Let’s use Pearson correlation, and then visualize the correlations.

|  |  |  |
| --- | --- | --- |
| Description | method | dist operations or penalty |
| Hamming distance (a and b must have same nr of characters). | “hamming” | s |
| Longest common substring distance. | “lcs” | di |
| Levenshtein distance (as in R’s nativeadist). | “lv” | dis |
| Optimal string aligment, (restricted Damerau-Levenshtein distance). | “osa” | dist once |
| Full Damerau-Levenshtein distance. | “dl” | dist multiple times |
| q-gram distance. | “qgram” |  |
| Jaro distance | “jw” | p = 0 |
| Jaro-Winkler distance | “jw” | p = 0.1 |
| Jaro-Winkler distance | “jw” | p = 0.25 |
| Jaccard distance betweenq-gram profiles | “jaccard” |  |
| cosine distance betweenq-gram profiles | “cosine” |  |
| Distance based on soundex encoding | “soundex” |  |

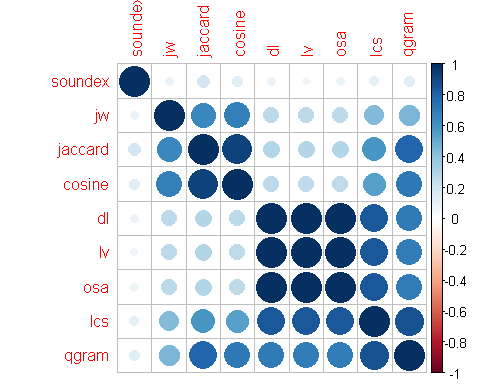
round(cor(stringdist\_Aaron),3)

## lcs lv osa dl qgram jw jaccard cosine soundex  
## lcs 1.000 0.843 0.843 0.844 0.866 0.434 0.587 0.542 0.114  
## lv 0.843 1.000 0.999 0.998 0.698 0.261 0.290 0.257 0.076  
## osa 0.843 0.999 1.000 0.999 0.700 0.261 0.292 0.259 0.076  
## dl 0.844 0.998 0.999 1.000 0.703 0.263 0.296 0.263 0.081  
## qgram 0.866 0.698 0.700 0.703 1.000 0.452 0.799 0.714 0.128  
## jw 0.434 0.261 0.261 0.263 0.452 1.000 0.640 0.686 0.081  
## jaccard 0.587 0.290 0.292 0.296 0.799 0.640 1.000 0.924 0.180  
## cosine 0.542 0.257 0.259 0.263 0.714 0.686 0.924 1.000 0.129  
## soundex 0.114 0.076 0.076 0.081 0.128 0.081 0.180 0.129 1.000

corrplot::corrplot(  
 corr = cor(stringdist\_Aaron),  
 order = "hclust",  
 hclust.method = "ward.D"  
)



corrplot::corrplot(  
 corr = cor(stringdist\_Aaron),  
 order = "hclust",  
 hclust.method = "single"  
)



Now let’s find the ten names with the most similar spelling to “Aaron” and create time series for these names.

v\_names <- v\_names[order(stringdist\_Aaron[,"dl"])]  
M\_2 <- M[M$name %in% v\_names[1:10],]  
M\_2 <- M\_2[M\_2$sex == "M",]  
ggplot(M\_2) +   
 aes(x = year,y = n,group = name,color = name) +   
 geom\_line() +   
 theme\_bw() +   
 ggtitle("Time series of SSA records for the names similar to Aaron","Males only")

