# dplyr, plyr, and tidyr: Baby Names in the US

*Sam Castillo*

*April 15, 2017*

```
library(plyr)
library(dplyr)
library(tidyr)
library(ggplot2)
library(gridExtra)
library(googleVis)
library(reshape2)
```

A dataset of baby names used by Hadley Wickham for a workshop can be found at https://github.com/hadley/babynames/tree/master/data. It contains the 1000 most popular male and female baby names in the US, from 1880 to 2008. There are 258,000 records (1000 * 2 * 129) but only four variables: year, name, sex, and percent.

The task is to identify the top 5 boys and girls names for each year from 1880 to 2008 and put it into a dataframe.

```
bnames = read.csv('bnames.csv')
str(bnames)
```

```
## 'data.frame':    258000 obs. of  4 variables:
##  $ year   : int  1880 1880 1880 1880 1880 1880 1880 1880 1880 1880 ...
##  $ name   : Factor w/ 6782 levels "Aaden","Aaliyah",..: 3380 6632 3125 1174 2554 2449 3428
##  $ percent: num  0.0815 0.0805 0.0501 0.0452 0.0433 ...
##  $ sex    : Factor w/ 2 levels "boy","girl": 1 1 1 1 1 1 1 1 1 1 ...
```

```
#View(bnames)
#bnames = as_data_frame(bnames)
summary(bnames)
```

```
##       year              name           percent             sex        
##  Min.   :1880   Jessie   :   258   Min.   :0.0000260   boy :129000  
##  1st Qu.:1912   Leslie   :   247   1st Qu.:0.0000810   girl:129000  
##  Median :1944   Guadalupe:   244   Median :0.0001640                
##  Mean   :1944   Jean     :   244   Mean   :0.0008945                
##  3rd Qu.:1976   Lee      :   240   3rd Qu.:0.0005070                
##  Max.   :2008   James    :   239   Max.   :0.0815410                
##                 (Other)  :256528                                    
```

We can arrange by percent and by year with dplyr.

```
tmp = bnames

boys = tmp %>%
  filter(sex == "boy")%>%
  group_by(name, year) %>%
  summarise(sum_percent = sum(percent))%>%
  select(-year) %>%
  arrange(desc(sum_percent))

head(boys,10)
```

```
## Source: local data frame [10 x 2]
## Groups: name [2]
##
##        name sum_percent
##      <fctr>       <dbl>
## 1      John    0.081541
## 2      John    0.080975
## 3   William    0.080511
## 4      John    0.079066
## 5   William    0.078712
## 6      John    0.078314
## 7      John    0.076476
## 8   William    0.076191
## 9      John    0.075820
## 10     John    0.075517
```

```
tmp = bnames

girls = tmp %>%
  filter(sex == "girl")%>%
  group_by(name, year) %>%
  summarise(sum_percent = sum(percent))%>%
  select(-year) %>%
  arrange(desc(sum_percent))

head(girls,10)
```

```
## Source: local data frame [10 x 2]
## Groups: name [1]
##
##       name sum_percent
##     <fctr>       <dbl>
## 1     Mary    0.072381
## 2     Mary    0.070431
## 3     Mary    0.069986
## 4     Mary    0.066990
## 5     Mary    0.066737
## 6     Mary    0.064334
## 7     Mary    0.064300
## 8     Mary    0.063620
## 9     Mary    0.062041
## 10   Mary    0.061562
```

Finding the top 5 for each year and putting it in a wider format is more difficult. The function `spread` from tidyr would not work easily in this case, at least not from what I can tell. Here I split the dataframe into a list of dataframes and then apply a function over the entire list. Finally, I put everything back together.

```
boys_tmp = subset(bnames, sex =="boy")
boys_tmp = split(boys_tmp, boys_tmp$year)

girls_tmp = subset(bnames, sex =="girl")
girls_tmp = split(girls_tmp, girls_tmp$year)
```

```
top5 = function(dat){
  cur_year = dat$year[1]
  top_rows = top_n(dat,5, percent)
  out = c(cur_year, as.character(top_rows$name))
  # df = as_data_frame(matrix(ncol = 6, nrow = 1))
  # df[1,] = out
  return(out)
}
```

Applying over the whole dataset.

```
mynames = c("year", "1st.name" ,"2nd.name" , "3rd.name", "4th.name", "5th.name")

girls_final = NULL
girls_final = sapply(girls_tmp, top5)
boys_final = ldply(boys_tmp, top5)

df_girls_final = setNames(do.call(rbind.data.frame, girls_final), mynames)

df_girls_final = df_girls_final[1:6] %>% as_data_frame()

df_boys_final = boys_final %>%
  select(-1)

names(df_boys_final) = mynames

df_final = rbind(df_girls_final, df_boys_final)

head(df_final, 5)
```

```
## # A tibble: 5 × 6
##     year `1st.name` `2nd.name` `3rd.name` `4th.name` `5th.name`
##   <fctr>     <fctr>     <fctr>     <fctr>     <fctr>     <fctr>
## 1   1880       Mary       Anna       Emma  Elizabeth     Minnie
## 2   1881       Mary       Anna       Emma  Elizabeth   Margaret
## 3   1882       Mary       Anna       Emma  Elizabeth     Minnie
## 4   1883       Mary       Anna       Emma  Elizabeth     Minnie
## 5   1884       Mary       Anna       Emma  Elizabeth     Minnie
```

```r
tail(df_final, 5)
```

```
## # A tibble: 5 × 6
##     year `1st.name` `2nd.name` `3rd.name` `4th.name` `5th.name`
##   <fctr>     <fctr>     <fctr>     <fctr>     <fctr>     <fctr>
## 1   2004      Jacob    Michael     Joshua    Matthew      Ethan
## 2   2005      Jacob    Michael     Joshua    Matthew      Ethan
## 3   2006      Jacob    Michael     Joshua      Ethan    Matthew
## 4   2007      Jacob    Michael      Ethan     Joshua     Daniel
## 5   2008      Jacob    Michael      Ethan     Joshua     Daniel
```

We can try to identify the "trendiest" baby names by fitting linear regression models. We only use simple linear regression here.

```r
data <- bnames

#creates a function of temp
lm.fit <- function(temp){
  #fits a simple linear regression model with year as the predictor and percent as the respons
  #over the columns percent and year of the data temp which was input
  fit <- lm( percent ~ year, data = temp)
  #returns the intercept and slope of the regression line, and n, the number of rows
  return(data.frame(int=fit$coef[1],slope=fit$coef[2],

  n=dim(temp)[1]))
}
#For each boys name and girls name, apply the lm.fit function to return a row for each name.
#the columns are name, sex, intercept, slope, and n respectively.  See the output from the hea
#function below
inc.dec <- ddply(data,.(name,sex),lm.fit)

#Examine only those names with greater than 100 observations
inc.dec <- subset(inc.dec,n>100)

#subset again to only the most extreme cases.  This looks only at the top 1% and botton 1% of
#the na.rm options removes missing values.
inc.dec <- subset(inc.dec, (slope > quantile(slope, p=0.99,na.rm=T))|(slope < quantile(slope, p
head(inc.dec)
```

```
##             name  sex        int         slope   n
## 425         Anna girl  0.3869374 -0.0001942732 129
## 1290     Charles  boy  0.5540564 -0.0002749670 129
## 1377 Christopher  boy -0.3882577  0.0002038261 129
## 1685      Daniel  boy -0.1973127  0.0001058796 129
## 1765       David  boy -0.2519217  0.0001381987 129
## 2700       Frank  boy  0.4428781 -0.0002224752 129
```

```
dim(inc.dec)
```

```
## [1] 16  5
```

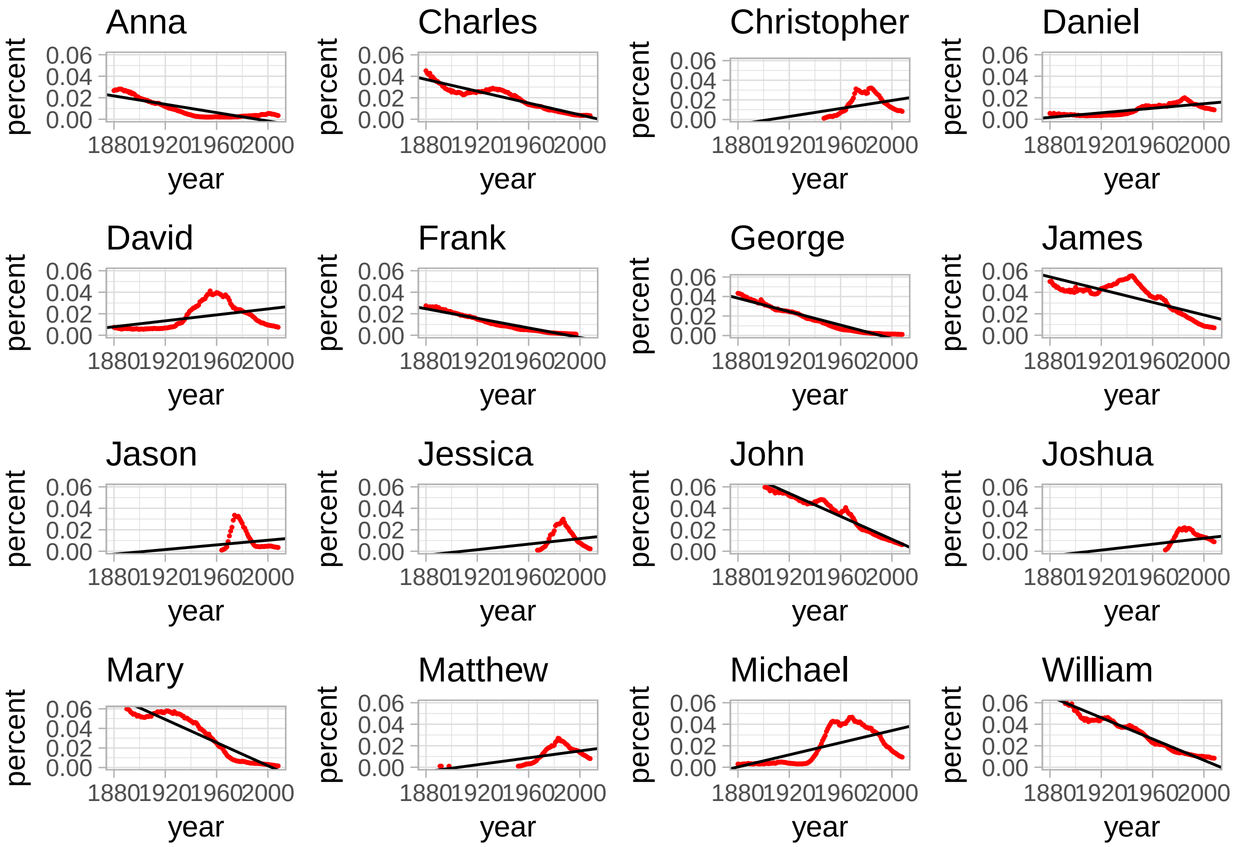
The data.frame inc.dec above has 16 rows. For each of those names, I make a scatterplot with year on the x-axis and percent on the y-axis. Then I label the plot with the name, and use abline() to add the least squares regression line.

```
outliers_df = filter(bnames, name %in% inc.dec$name)

ggplot_function = function(cur_name) {
  cur_dat = subset(outliers_df, name == cur_name)# %>%select(-sex, -name)
  cur_dat = cur_dat[cur_dat$percent > 0.001,] #Remove rows with percent equal to zero
  names(cur_dat) = c("year", "name", "percent", "sex")
  lm.dat = subset(inc.dec, name == cur_name)
  p1 = ggplot(data = cur_dat, aes(year,percent)) + geom_point(size = 0.2, colour = "red") + geo

  p1
}
```

```
plots = lapply(inc.dec$name, ggplot_function)
```

```
do.call("grid.arrange", plots)
```

```
## Warning: Removed 21 rows containing missing values (geom_point).

## Warning: Removed 10 rows containing missing values (geom_point).

## Warning: Removed 13 rows containing missing values (geom_point).
```



Have most babies had similar names in certain years? In other words, how has the sum of percentage of the top 100 baby names changed over time?

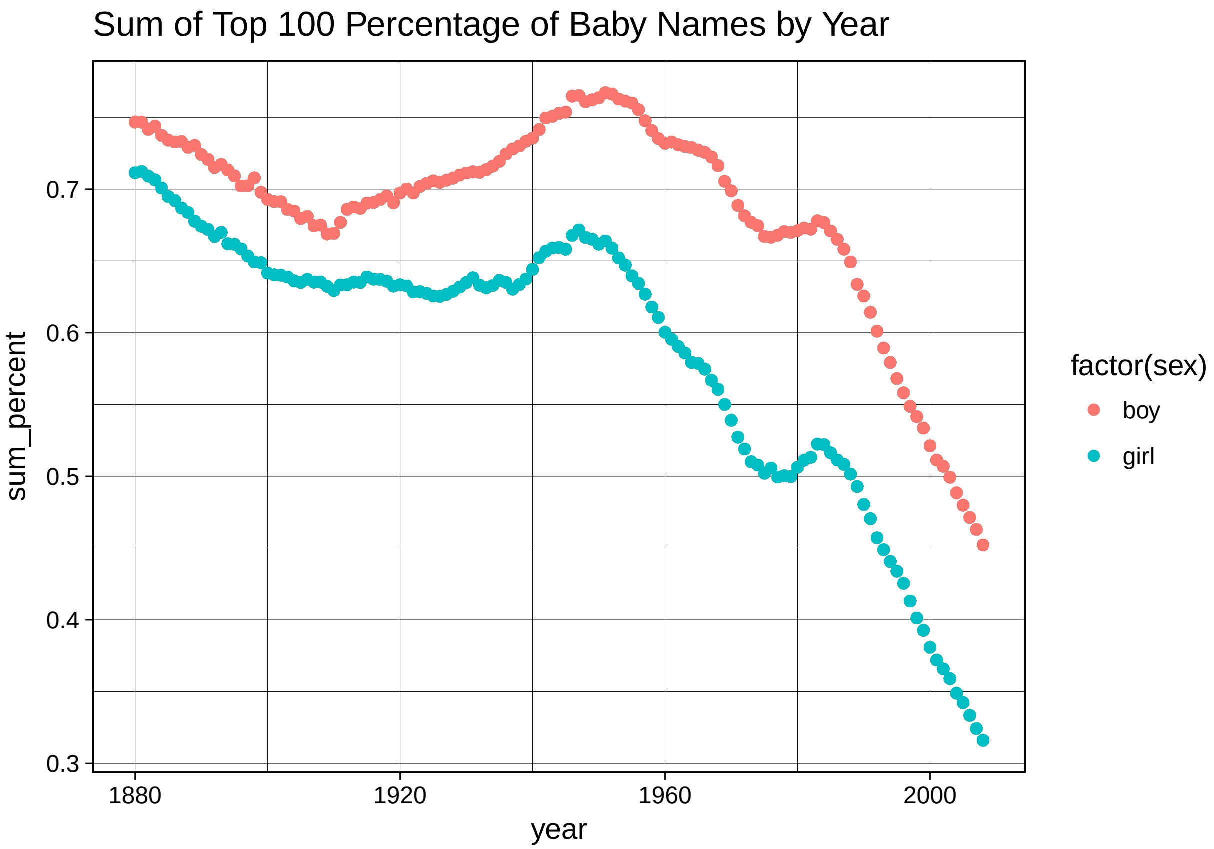
Here I create a plot that shows (by year and gender) the proportion of US children who have a name in the top 100. Proportion is on the y-axis, year on the x-axis, and two lines, one for each gender.

```
suppressPackageStartupMessages(library(googleVis))

df_boys = bnames %>%
  filter(sex =="boy") %>%
  group_by(year) %>%
  arrange(desc(percent)) %>%
  filter(percent > min(head(percent, 101))) %>%
  group_by(year)%>%
  mutate(sum_percent = sum(percent))

df_girls = bnames %>%
  filter(sex =="girl") %>%
  group_by(year) %>%
  arrange(desc(percent)) %>%
  filter(percent > min(head(percent, 101))) %>%
  group_by(year)%>%
  mutate(sum_percent = sum(percent))
head(df_girls)
```

Source: local data frame [6 x 5] Groups: year [6]

year name percent sex sum_percent 1 1880 Mary 0.072381 girl 0.711437 2 1882 Mary 0.070431 girl 0.709072 3 1881 Mary 0.069986 girl 0.712258 4 1884 Mary 0.066990 girl 0.700749 5 1883 Mary 0.066737 girl 0.706464 6 1886 Mary 0.064334 girl 0.692079

```
p2 = ggplot(data = rbind(df_boys, df_girls), aes(year, sum_percent, sex))

p2 + geom_point(aes(color = factor(sex))) + ggtitle("Sum of Top 100 Percentage of Baby Names by
```



```
df3 = rbind(df_boys, df_girls)

df3$year = as.Date(as.character(df3$year), "%Y")

Anno = gvisAnnotationChart(df3,
                           datevar = "year",
                           numvar = "sum_percent",
                           idvar = "sex"
                           )
```

```
## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for sex=boy: first taken
```

```
## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : multiple rows match for sex=girl: first taken
```

```
Anno
```

AnnotationChartID233c2c3b2218

Data: df3 • Chart ID: AnnotationChartID233c2c3b2218 • googleVis-0.6.2  R version 3.3.2 (2016-10-31) • Google Terms of Use • Documentation and Data Policy