Facebook Birthday Analysis

Patrick Vo

June 25, 2017

Welcome to my Facebook birthday analysis.I decided to use the example birthdays.csv file to do this problem.

Load in the necessary libraries:

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

Let's start out by reading in the birthday data and seeing what it looks like.

```
#Read in the csv
bd <- read.csv('birthdaysExample.csv')

head(bd, n = 2)
```

```
##      dates
## 1 11/25/14
## 2   6/8/14
```

The dates are stored in a dataframe with 1 column, dates. The formatting of the birthdates themselves may be a little problematic. Let's check to see what kind of datatypes we're working with.

```
class(bd$dates[0])
```

```
## [1] "factor"
```

```
mode(bd$dates[0])
```

```
## [1] "numeric"
```

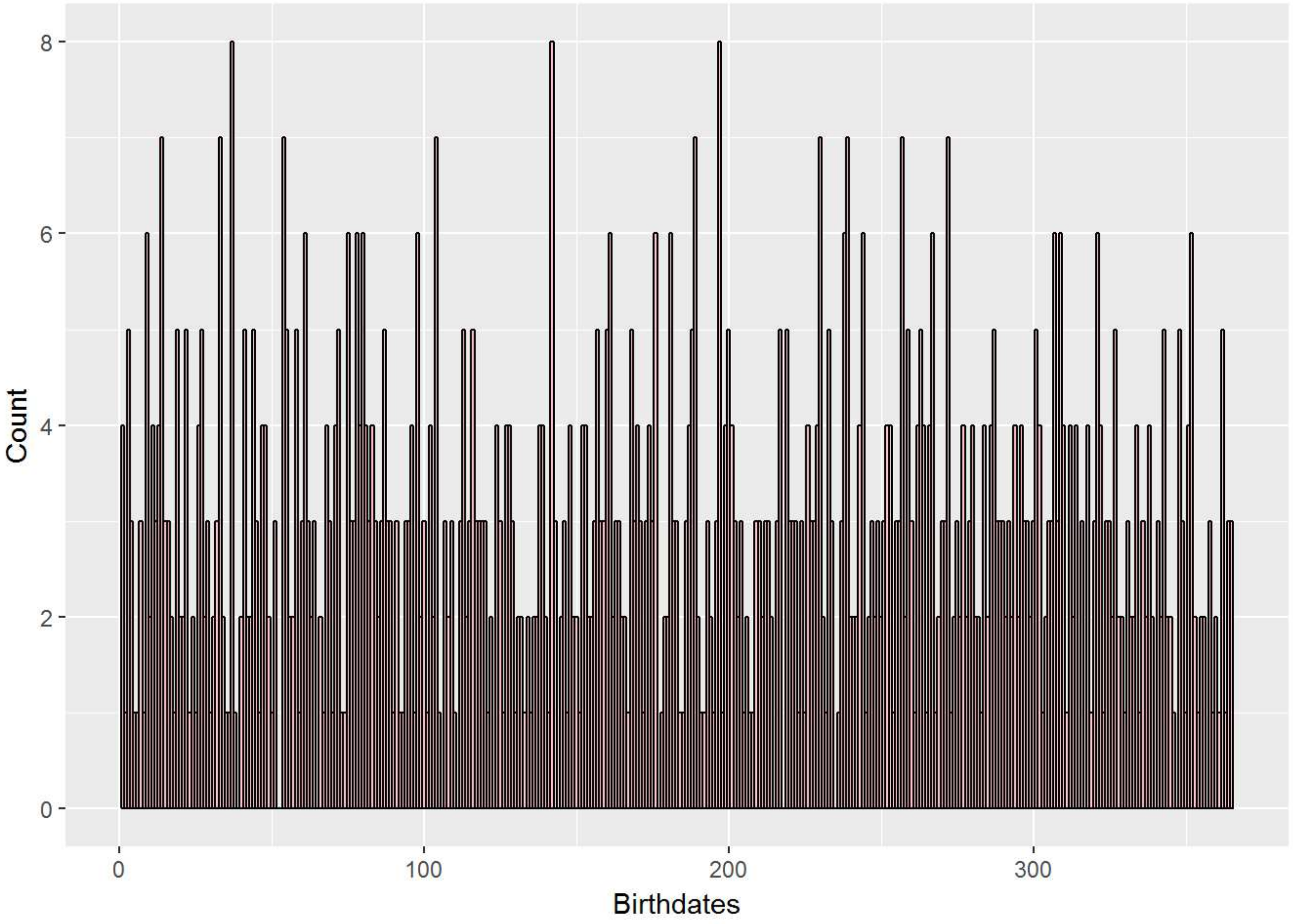
A quick search of datetime objects in R reveals that you can convert from various formats to an R datetime object (a POSIXlt object) using the strptime() function.

I decided to convert the dates, then plot the birthdates to see what kind of distribution the birthdays would have. The first time I tried this, I tried to plot directly from the vector of POSIXlt's, and R crashed my computer. I decided to convert the datetime value into integers from 1-365, each one representing a day of the year

```
#Convert the birthdays into R POSIXLt
bd_c <- strptime(bd$dates, "%m/%d/%y")

#Convert the new datetime vector into an integer vector
day <- strftime(bd_c, "%j")
day <- as.numeric(day)

qplot(day, xlab = 'Birthdates', ylab = 'Count', fill = I('pink'), color = I('black'), binwidt
h = 1)
```



Even without being able to tell which days correspond to which counts, I can answer a few of the sample questions. The largest number of births that appear on any given day is 8. The smallest number of births that on any given day is 0.

Now, I would normally expect the distribution of birthdates to be uniform. I wanted to do some additional testing of that assumption. Because the data are discrete, I decided to use Pearson’s chi-squared test. Typically, chi-squared goodness-of-fit testing requires that at least 20% of the expected data counts are above 5– the expected number of births each day is only 1033/365 = 2.83. To get around this, I decided to test the monthly distributions of birthdays instead.

Notice that the expected number of births in each month is based on the number of days in each month.

```
#Convert the POSIXlt data to months
birth_months = strftime(bd_c, '%m')

#Use the table function to put month counts together
month_counts <-table(birth_months)

#Find the expected number of births in each month
days_per_month <- c(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)
expected_births_per_month <- days_per_month * (1033/365)

#Apply the goodness of fit test
chisq.test(month_counts, expected_births_per_month)
```

```
## Warning in chisq.test(month_counts, expected_births_per_month): Chi-squared
## approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  month_counts and expected_births_per_month
## X-squared = 24, df = 18, p-value = 0.155
```

A p-value of 0.155 is pretty good for a goodness-of-fit test, and shows that our monthly counts are fairly close to those expected fo a uniform random distribution.

Lastly, I wanted to see if any of my friends share a birthday with me. I decided to convert my birthday to an integer and find all the matches:

```
#Declare my birthday and convert to the day of the year
my_birthday <- "2014-09-20"
my_birthday <-as.numeric(strftime(my_birthday, "%j"))


#Sum up all the days that in the integer vector that match my birthday
sum(day == my_birthday)
```

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

A whopping 5 of my friends share my birthday. I feel very ordinary.