A Drop in Gun Deaths in February?

Investigating a Curious Trend in CDC Gun Deaths 2012-2014

Thadryan Sweeney

April 15, 2018

This set of gun deaths, collected mostly from the CDC, spans 2012, 2013, and 2014. I was initially interested in building a classifier to see if a machine could predict, with reasonable to strong accuracy, a person's race based on how they died with a gun (initial findings, somewhat eerily? Yes). But in familiarizing myself with the dataset, I noticed something. Each year showed a pronounced drop in gun deaths in February. At first I dismissed this, thinking it was due to the fact that it's the shortest month. It's also only three years of data. I took a look just the same and the findings are a bit unusual given the strength of the pattern.

First, we load the data and make sure Feb doesn't have a disproportionate amount of missing values, skewing my analysis. If Feb gun deaths were more poorly documented, for instance, that might be why (I used a random forest imputation strategy in my analysis and wanted to make sure any errors in this weren't the cause of the issue). We will use the raw data.

Non-coders, fear not: there are written language chunks between each block, and yellow lines following a "#" explain what happens at each step.

```r
# some tools for generating pretty output
library("kableExtra")
library("knitr")
```

We will check to see if there is a difference if proportions of "Feb" entries with missing values vs complete values.

```r
# get the data
d <- read.csv("full_data.csv")

# complete data - omit all rows missing something
c.d <- na.omit(d)

# proportions of deaths in raw data by month
prop.table(table(d$month))
```

```
##
##          1          2          3          4          5          6
## 0.08207504 0.07036846 0.08223377 0.08388063 0.08600369 0.08608306
##          7          8          9         10         11         12
## 0.08917836 0.08713467 0.08440644 0.08339451 0.08177742 0.08346396
```

```r
# proportions of Feb deaths in complete data by month
prop.table(table(c.d$month))
```

```
##
##          1          2          3          4          5          6
## 0.08219150 0.07039739 0.08241596 0.08373208 0.08590522 0.08588481
##          7          8          9         10         11         12
## 0.08913942 0.08684385 0.08467071 0.08346682 0.08187522 0.08347702
```

We see proportions of 0.07036846 vs 0.07039739 for month 2. Feb makes up almost exactly as much of the dataset with or without missing records. So we can probably lay that to rest.

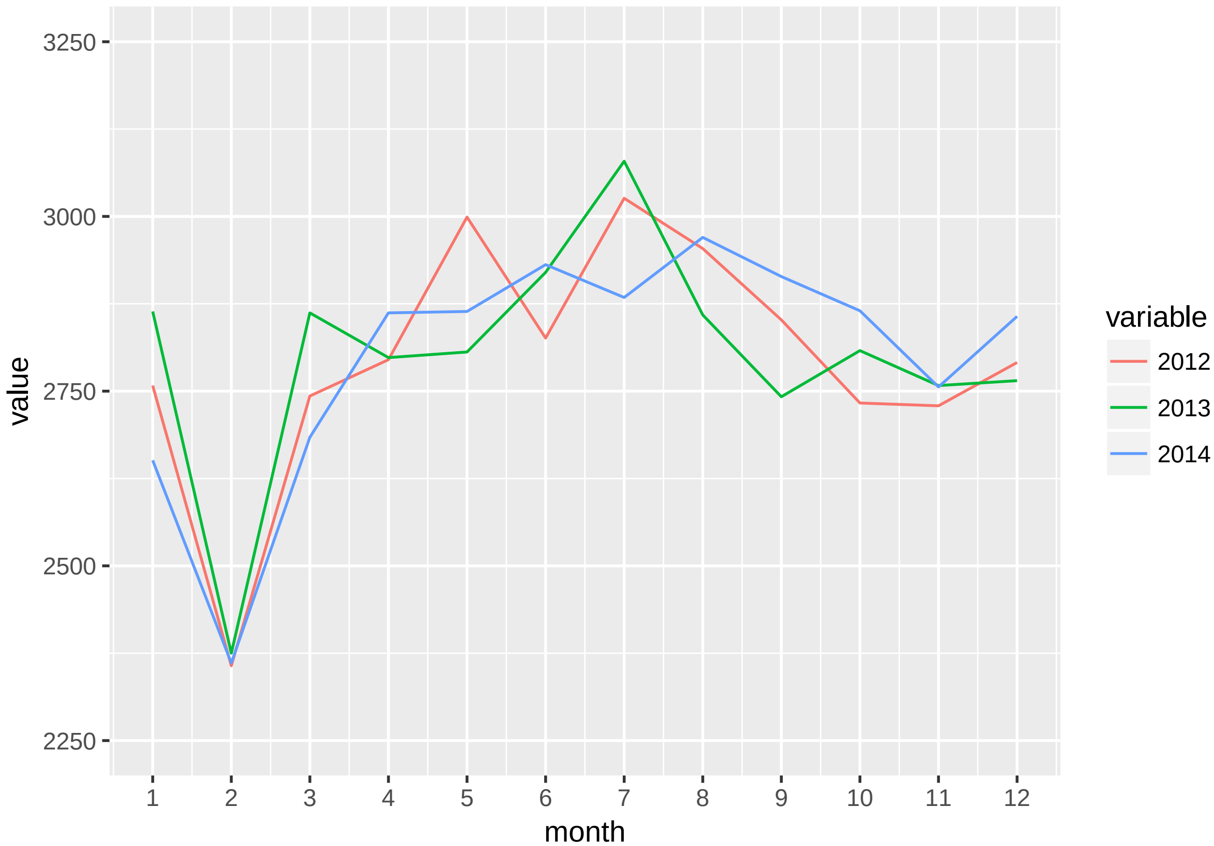
Visual Analysis

We'll now visualize the data for Feb. This is where I started to get suspicious:

```r
library(ggplot2)
library(reshape2)

# frame the data by year
data12 <- d[which(d$year == "2012"), ]
data13 <- d[which(d$year == "2013"), ]
data14 <- d[which(d$year == "2014"), ]

# extract month data
d12 <- data.frame(summary(as.factor(data12$month)))
d13 <- data.frame(summary(as.factor(data13$month)))
d14 <- data.frame(summary(as.factor(data14$month)))

# set months
month <- c(1,2,3,4,5,6,7,8,9,10,11,12)

# make a new dataframe of deaths per month
month.data <- cbind(month,d12,d13,d14)

# set new names
colnames(month.data) <- c("month", "2012","2013","2014")

# inspect the deaths/month data
kable(month.data) %>%
  kable_styling(position = "center", full_width = TRUE) %>%
  row_spec(0, bold = TRUE) %>%
  row_spec(2, bold = TRUE, color = "blue")
```

month	2012	2013	2014
1	2758	2864	2651
2	**2357**	**2375**	**2361**
3	2743	2862	2684
4	2795	2798	2862
5	2999	2806	2864
6	2826	2920	2931
7	3026	3079	2884
8	2954	2859	2970
9	2852	2742	2914
10	2733	2808	2865
11	2729	2758	2756
12	2791	2765	2857

```r
# melt the dataframe for easy visualization
month.data <- melt(month.data, id.vars = "month")

# plot the results on a line graph
ggplot(month.data, aes(month,value, col =  variable)) +
  geom_line() +
  # set x and y limits
  scale_y_continuous(limits = c(2250,3250), breaks = seq(2250, 3250, by = 250)) +
  scale_x_continuous(breaks = seq(1,12, by = 1))
```



There is a very obvious drop in Feb. This persists in an obvious way even when the scale of the graph is changed. Is it only due to the fact that it is the shortest month?

Numeric Analysis

To investigate, we will find the average death per day and use that to estimate what the Feb deaths would look like if they were normal.

```
# vector of months by name
months <- c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")

# list of days in months
days   <- c(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)

# restructure the dataframe to have days per month
month.data <- cbind(months, days, d12, d13, d14)

# set the names
colnames(month.data) <- c("month","days","d12","d13","d14")

# look at the data
kable(month.data) %>%
  kable_styling(position = "center", full_width = TRUE) %>%
  row_spec(0, bold = TRUE) %>%
  row_spec(2, bold = TRUE, color = "blue")
```

month	days	d12	d13	d14
Jan	31	2758	2864	2651
Feb	**28**	**2357**	**2375**	**2361**
Mar	31	2743	2862	2684
Apr	30	2795	2798	2862
May	31	2999	2806	2864
Jun	30	2826	2920	2931
Jul	31	3026	3079	2884
Aug	31	2954	2859	2970
Sep	30	2852	2742	2914
Oct	31	2733	2808	2865
Nov	30	2729	2758	2756
Dec	31	2791	2765	2857

We now have a dataframe where we can make a prediction of what the values would be if they simply followed the number of deaths per day.

Let's find the deaths per day:

```
# get deaths per day by year
d.per.day12 <- sum(month.data$d12)/365
d.per.day13 <- sum(month.data$d13)/365
d.per.day14 <- sum(month.data$d14)/365

# gun deaths per day in this data set
d.per.day <- mean(c(d.per.day12 , d.per.day13 , d.per.day14))

# show information by year
# 2012
d.per.day12
```

```
## [1] 91.95342
```

```
# 2013
d.per.day13
```

```
## [1] 92.15342
```

```
# 2013
d.per.day14
```

```
## [1] 92.05205
```

```
# deaths per day
d.per.day
```

```
## [1] 92.05297
```

Now we can simply multiply the number of days days in the month times the average deaths per day to see what it would be if it was following the trend. We'll call this the "expected" value. We will add another value called "diff.exp" that shows how far off from the expectation the reality is (the "reality"" being the average of the actual observations for that month)

```
# iterate by rows
for(i in 1:nrow(month.data)) {

  # the "expect" column is the number of days times the average per day
  month.data$expected[i]  <- month.data$days[i] * d.per.day
```

```
  # add "reality" - average of actual observations from each year in that month
  month.data$reality[i] <- mean(c(month.data$d12[i] ,  month.data$d13[i] ,  month.data$d14[i]))

  # the "diff.exp" - difference from expected and the actual average
  month.data$dif.exp[i] <- month.data$reality[i] - month.data$expect[i]
}

# 2012 was a leap year so we will add the average once more to it
month.data$d12[2] <- month.data$d12[2] + d.per.day

# look at the data
kable(month.data) %>%
  kable_styling(position = "center") %>%
  row_spec(0, bold = TRUE) %>%
  row_spec(2, bold = TRUE, color = "blue")
```

month	days	d12	d13	d14	expected	reality	dif.exp
Jan	31	2758.000	2864	2651	2853.642	2757.667	-95.97534
Feb	**28**	**2449.053**	**2375**	**2361**	**2577.483**	**2364.333**	**-213.14977**
Mar	31	2743.000	2862	2684	2853.642	2763.000	-90.64201
Apr	30	2795.000	2798	2862	2761.589	2818.333	56.74429
May	31	2999.000	2806	2864	2853.642	2889.667	36.02466
Jun	30	2826.000	2920	2931	2761.589	2892.333	130.74429
Jul	31	3026.000	3079	2884	2853.642	2996.333	142.69132
Aug	31	2954.000	2859	2970	2853.642	2927.667	74.02466
Sep	30	2852.000	2742	2914	2761.589	2836.000	74.41096
Oct	31	2733.000	2808	2865	2853.642	2802.000	-51.64201
Nov	30	2729.000	2758	2756	2761.589	2747.667	-13.92237
Dec	31	2791.000	2765	2857	2853.642	2804.333	-49.30868

We now have a table of the expected values based on the average, as well as the difference between the expected and actual averages.

February is still looking pretty weird. To increase the rigor of our poking around, we will look at the z-scores. We will now add a column representing the z-score of the "diff.expected" column. We will also re-frame the data so that only the columns we currently need are displayed so it's obvious whats going on. Typically z-scores of either +/- 3.0 or +/- 1.5 are used as starting points in outlier detection.

```
# we will see how differs in expected
month.data$z.expected <- scale(month.data$expected)

# frame the most relevant stats
feb.variance <- month.data[, c("month","expected","reality", "dif.exp","z.expected")]

# look at the data
kable(month.data) %>%
  kable_styling(position = "center") %>%
  row_spec(0, bold = TRUE) %>%
  row_spec(2, bold = TRUE, color = "blue")
```

month	days	d12	d13	d14	expected	reality	dif.exp	z.expected
Jan	31	2758.000	2864	2651	2853.642	2757.667	-95.97534	0.6479058
Feb	**28**	**2449.053**	**2375**	**2361**	**2577.483**	**2364.333**	**-213.14977**	**-2.6841812**
Mar	31	2743.000	2862	2684	2853.642	2763.000	-90.64201	0.6479058
Apr	30	2795.000	2798	2862	2761.589	2818.333	56.74429	-0.4627899
May	31	2999.000	2806	2864	2853.642	2889.667	36.02466	0.6479058
Jun	30	2826.000	2920	2931	2761.589	2892.333	130.74429	-0.4627899
Jul	31	3026.000	3079	2884	2853.642	2996.333	142.69132	0.6479058
Aug	31	2954.000	2859	2970	2853.642	2927.667	74.02466	0.6479058
Sep	30	2852.000	2742	2914	2761.589	2836.000	74.41096	-0.4627899
Oct	31	2733.000	2808	2865	2853.642	2802.000	-51.64201	0.6479058
Nov	30	2729.000	2758	2756	2761.589	2747.667	-13.92237	-0.4627899
Dec	31	2791.000	2765	2857	2853.642	2804.333	-49.30868	0.6479058

Feb's weirdness holds up pretty well to this test as well, clocking in with -2.64 (I didn't use absolute value so I could see which direction we were going in). This comfortably surpass the 1.5 threshold and approaches the 3.0. Which to use requires some discretion and context.

The next largest score deviations are around 0.64, less than a quarter of Feb's. None of them make it even half-way to 1.5, a solid case that 1.5 is more appropriate that 3.0. It appears by this measure, Feb is very much abnormal. If we go with 3.0 (which doesn't seem as contextually appropriate), it still certainly seems odd enough to warrant further investigation. We'll trying sidestepping some of this uncertainty about the relative appropriateness by using quartiles.

This gives an inter-quartile range of:

```
# the interquartle range:
IQR <- IQR(month.data$dif.exp)
```

One formal definition of outlier is a number found outside a certain range, defined as follows:

low end: Q1 - (1.5 x IQR)

high end: Q3 + (1.5 x IQR)

```
# get the summary data
quartiles <- as.vector(summary(month.data$dif.exp))

# get first and third quartiles
firstQ <- quartiles[2]
thirdQ <- quartiles[5]

# lower end of formal outlier range
low <- firstQ - (1.5 * IQR)

# higher end of the formal outlier range
high <- thirdQ + (1.5 * IQR)

low
```

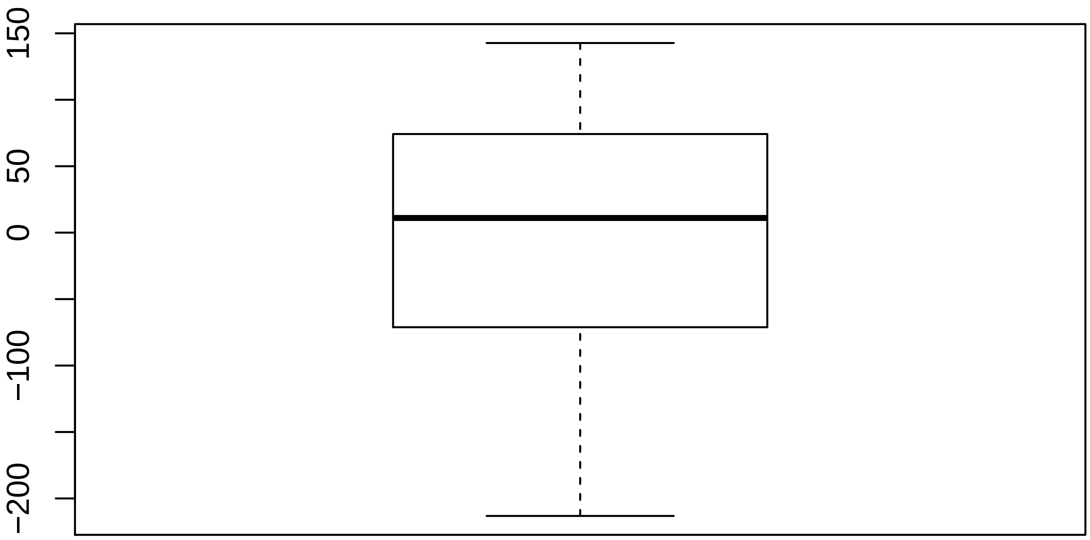
```
## [1] -264.6599
```

```
high
```

```
## [1] 277.3899
```

At -213.14977, Feb is not an outlier by this definition (though there is no once-and-for-all definition). This seems in keeping with R's boxplot function which a similar method to determine ranges and puts February right at the extreme but not over it:

```r
# create a boxplot of dif.exp
boxplot(month.data$dif.exp)
```



A proof reader of mine pointed out that I should look at the different type of gun deaths to see if there was an obvious change in an particular type that changed or if there was simply a change of volume. Let's see if the types of death (as measured by the "intent" feature) change.

We will first look at the data framed without Feb entries at all. We will compare this to the summaries with Feb and the summaries of only Feb.

```r
# intent proportions in general data
prop.table(table(d$intent))
```

```
##
##   Accidental     Homicide      Suicide Undetermined
##  0.016260405  0.348978640  0.626754765  0.008006191
```

```r
# frame data without Feb
non.feb.data <- d[which(d$month != 2), ]
prop.table(table(non.feb.data$intent))
```

```
##
##   Accidental     Homicide      Suicide Undetermined
##  0.016135917  0.352151456  0.623698028  0.008014599
```

```r
# frame data as only Feb
feb.data <- d[which(d$month == 2), ]
prop.table(table(feb.data$intent))
```

```
##
##   Accidental     Homicide      Suicide Undetermined
##  0.017904977  0.307063302  0.667136614  0.007895108
```

There is a slight shift in the proportions, with homicide decreasing and suicide increasing, but nothing as sharp as the deviation itself. Still, we will look at homicides.

```r
# frame the data by year
data12 <- na.omit(d[which(d$year == "2012"), ])
data13 <- na.omit(d[which(d$year == "2013"), ])
data14 <- na.omit(d[which(d$year == "2014"), ])


# extract month data
d12 <- data.frame(summary(as.factor(data12$intent)))
d13 <- data.frame(summary(as.factor(data13$intent)))
d14 <- data.frame(summary(as.factor(data14$intent)))


# make a new dataframe of deaths per month
intent.data <- cbind(d12,d13,d14)

# set new names
colnames(intent.data) <- c("2012","2013","2014")

# inspect the deaths/month data
kable(intent.data) %>%
  kable_styling(position = "center", full_width = TRUE) %>%
  row_spec(0, bold = TRUE)
```

	2012	**2013**	**2014**
Accidental	533	490	575
Homicide	11467	11073	10789
Suicide	20360	20892	21039
Undetermined	255	275	267

```r
# frame the data by year
non.feb.data12 <- na.omit(d[which(d$year == "2012"), ])
non.feb.data13 <- na.omit(d[which(d$year == "2013"), ])
non.feb.data14 <- na.omit(d[which(d$year == "2014"), ])


# extract month data
d12 <- data.frame(summary(as.factor(non.feb.data12$intent)))
d13 <- data.frame(summary(as.factor(non.feb.data13$intent)))
d14 <- data.frame(summary(as.factor(non.feb.data14$intent)))

# set months
#month <- c(1,2,3,4,5,6,7,8,9,10,11,12)

# make a new dataframe of deaths per month
non.feb.intent.data <- cbind(d12,d13,d14)

# set new names
colnames(non.feb.intent.data) <- c("2012","2013","2014")

# inspect the deaths/month data
kable(non.feb.intent.data) %>%
  kable_styling(position = "center", full_width = TRUE) %>%
  row_spec(0, bold = TRUE)
```

	2012	2013	2014
Accidental	533	490	575
Homicide	11467	11073	10789
Suicide	20360	20892	21039
Undetermined	255	275	267

I should probably just use apply abd overwite the whole thing

```r
# make colums for the z-score that start at zero
intent.data$z12 <- 0
intent.data$z13 <- 0
intent.data$z14 <- 0

# iterate through the dataframe
for(i in 1:nrow(intent.data)) {
  # put the scaled values in the new columns
  intent.data[i,4:6] <- scale(as.numeric(intent.data[i,1:3]))
}

# label the new data
colnames(intent.data) <-c("d12", "d13","d14","z12", "z13", "z14")
intent.data <- intent.data[, c("d12","z12","d13","z13","d14", "z14")]

# inspect the deaths/month data
kable(intent.data) %>%
  kable_styling(position = "center", full_width = TRUE) %>%
  row_spec(0, bold = TRUE)
```

	d12	z12	d13	z13	d14	z14
Accidental	533	0.007843	490	-1.0038984	575	0.9960555
Homicide	11467	1.049486	11073	-0.1076898	10789	-0.9417967
Suicide	20360	-1.129995	20892	0.3592470	21039	0.7707481
Undetermined	255	-1.059626	275	0.9271726	267	0.1324532

With these modest Z scores, it seems apparent that there is no particular type of gun death that accounts for this, rather a general drop across the board.

Conclusion

After all this I'm considering February "suspicious", and doing some further investigation.

A little online browsing reveals the following:

https://chicago.suntimes.com/news/chicago-gun-violence-february/

https://www.usatoday.com/story/news/2018/03/01/murders-shootings-down-chicago-1st-two-months-2018/385074002/

but nothing close to the time-frame of the original dataset or on a scale larger than a major city. It does however, make me wonder if the pattern held true in 2018 and the years between 2014 and 2018.

Have you heard anything about this? What would you look at next?