A Drop in Gun Deaths in February?

Investigating a Curious Trend in CDC Gun Deaths 2012-2014

Thadryan Sweeney

April 15, 2018

Introduction

This dataset of gun deaths, collected mostly from the CDC by FiveThirtyEight, 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 and consistency of the pattern in context, bringing to mind a significant trend or data quality issue.

Abstract

First, we loaded the dataset and made sure February didn't have a disproportionate amount of missing values, skewing my analysis. If February gun deaths were more poorly documented, for instance, that might cause the skew when missing values were discarded. The results were then plotted by deaths per month for each year to visualize any trends. The dataset was then adjusted to normalize the results based on the number of days in each month. An "expected value" was calculated for each month based on the number of days it contains and the average deaths per day of the dataset as a whole to see if it breaks from the overall trend of the dataset for a month of appropriate length. February's Z-score for this figure was -2.01, easily the most abnormal (the next higehst was July with +1.37). The dataset was then analyzed using Z-score normalization, inter-quartile range, and a box plot, showing the drop in February to be significantly unusual compared to the rest of the months, while not conforming to strict definitions of an outlier. The dataset was analyzed based on the "intent" value, and it was observed that February had a decrease in all types of deaths, with "Homicides" being the most deviant with a Z-score of -2.47 compared to other months.

Note for non-coders

It's my intent that this document be useful to those who don't code. There are written language chunks between each block of code, and yellow lines following a "#" explain what happens at each step of the program.

Preparing and Initial Inspection

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

Missing Values

First, we will check to see if there is a difference in proportions of February entries with missing values vs complete values to make sure it's not that missing values happened to be concentrated in February.

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

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

# proportions of deaths in raw data by month
prop.table(table(o.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
```

```
# 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. We will use only complete records for the analysis having established this.

Visual Analysis

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

```
library(ggplot2)
library(reshape2)

# get complete records only
d <- c.d

# 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 from summaries
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
```

```r
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")
```

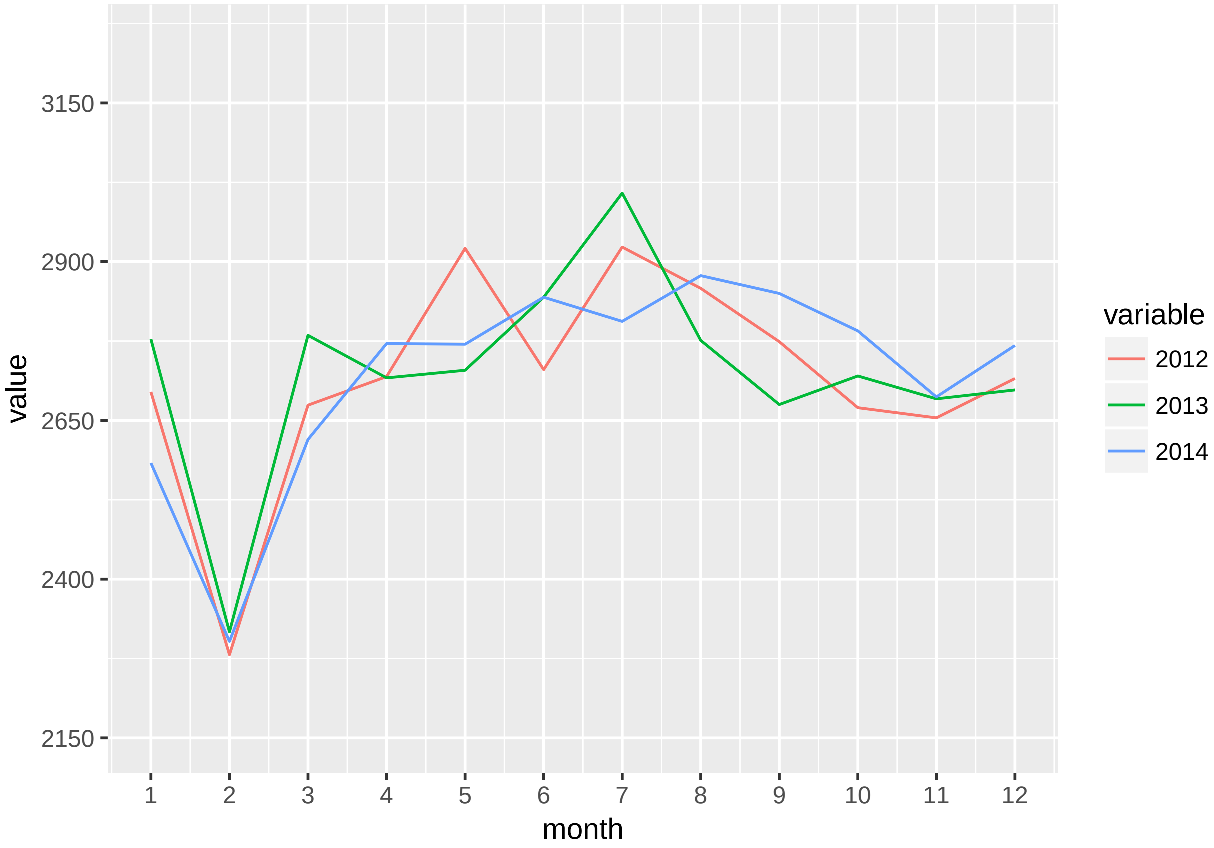
month	**2012**	**2013**	**2014**
1	2695	2778	2583
2	**2281**	**2317**	**2302**
3	2674	2784	2620
4	2719	2717	2771
5	2921	2729	2770
6	2730	2844	2844
7	2923	3008	2806
8	2858	2776	2878
9	2774	2675	2850
10	2670	2720	2791
11	2654	2684	2687
12	2716	2698	2768

We can see that we have a well formed dataframe based on the summaries of the year. Now to prepare a graph.

Graphing February Deaths

```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(2150,3250), breaks = seq(1650, 3350, by = 250)) +
  scale_x_continuous(breaks = seq(1,12, by = 1))
```



There is a very obvious drop in February. 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?

Adjusting for Month Length

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	2695	2778	2583
Feb	**28**	**2281**	**2317**	**2302**
Mar	31	2674	2784	2620
Apr	30	2719	2717	2771
May	31	2921	2729	2770
Jun	30	2730	2844	2844
Jul	31	2923	3008	2806
Aug	31	2858	2776	2878
Sep	30	2774	2675	2850
Oct	31	2670	2720	2791
Nov	30	2654	2684	2687
Dec	31	2716	2698	2768

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 calculate that figure.

Deaths Per Day

```r
# 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))

# deaths per day in 2012...
d.per.day12
```

```
## [1] 89.35616
```

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

```
## [1] 89.67123
```

```r
# 2014
d.per.day14
```

```
## [1] 89.50685
```

```r
# deaths per day in dataset
d.per.day
```

```
## [1] 89.51142
```

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 in the three years)

“Expected” Deaths

```r
# 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	2695.000	2778	2583	2774.854	2685.333	-89.52055
Feb	**28**	**2370.511**	**2317**	**2302**	**2506.320**	**2300.000**	**-206.31963**
Mar	31	2674.000	2784	2620	2774.854	2692.667	-82.18721
Apr	30	2719.000	2717	2771	2685.342	2735.667	50.32420
May	31	2921.000	2729	2770	2774.854	2806.667	31.81279
Jun	30	2730.000	2844	2844	2685.342	2806.000	120.65753
Jul	31	2923.000	3008	2806	2774.854	2912.333	137.47945
Aug	31	2858.000	2776	2878	2774.854	2837.333	62.47945
Sep	30	2774.000	2675	2850	2685.342	2766.333	80.99087
Oct	31	2670.000	2720	2791	2774.854	2727.000	-47.85388
Nov	30	2654.000	2684	2687	2685.342	2675.000	-10.34247
Dec	31	2716.000	2698	2768	2774.854	2727.333	-47.52055

Z-Score of Expected Deaths

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. Often, z-scores of either +/- 3.0 or +/- 1.5 are used as starting points in outlier detection in data science projects, so we will start there.

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

# frame the most relevant stats
```

```r
feb.variance <- month.data[, c("month","expected","reality", "dif.exp","z.diff.exp")]

# 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.diff.exp
Jan	31	2695.000	2778	2583	2774.854	2685.333	-89.52055	-0.8974181
Feb	**28**	**2370.511**	**2317**	**2302**	**2506.320**	**2300.000**	**-206.31963**	**-2.0682959**
Mar	31	2674.000	2784	2620	2774.854	2692.667	-82.18721	-0.8239035
Apr	30	2719.000	2717	2771	2685.342	2735.667	50.32420	0.5044859
May	31	2921.000	2729	2770	2774.854	2806.667	31.81279	0.3189142
Jun	30	2730.000	2844	2844	2685.342	2806.000	120.65753	1.2095576
Jul	31	2923.000	3008	2806	2774.854	2912.333	137.47945	1.3781926
Aug	31	2858.000	2776	2878	2774.854	2837.333	62.47945	0.6263388
Sep	30	2774.000	2675	2850	2685.342	2766.333	80.99087	0.8119105
Oct	31	2670.000	2720	2791	2774.854	2727.000	-47.85388	-0.4797216
Nov	30	2654.000	2684	2687	2685.342	2675.000	-10.34247	-0.1036803
Dec	31	2716.000	2698	2768	2774.854	2727.333	-47.52055	-0.4763800

Feb's weirdness holds up pretty well to this test as well, clocking in with -2.068 (I didn't use absolute value so I could see which direction we were going in). This comfortably surpass the 1.5 threshold but not the 3.0. 2 other months break 1.0 (but not 1.5), June and July, with 1.209 and 1.378 respectively.

We'll trying sidestepping some of this uncertainty about the relative appropriateness by using quartiles.

IQR

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

low end: Q1 - (1.5 x IQR)

high end: Q3 + (1.5 x IQR)

We will use that as another possibly useful metric.

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

# 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] -241.7568
```

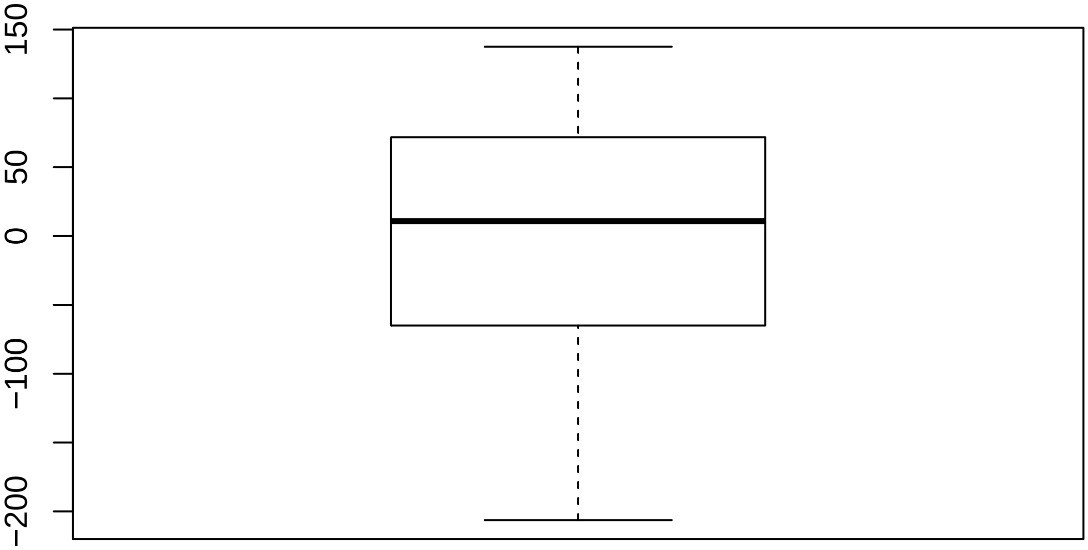
```
high
```

```
## [1] 252.4268
```

Boxplot

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 box plot function which a similar method to determine ranges and puts February right at the extreme but not over it:

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



Proportions with/without Feb, etc

One of the people I asked for input on this projected pointed out to me 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.

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

```
##
##   Accidental     Homicide      Suicide Undetermined
##  0.016303627  0.340039790  0.635525175  0.008131408
```

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

```
##
##   Accidental     Homicide      Suicide Undetermined
##  0.016199309  0.343071942  0.632585195  0.008143555
```

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

```
##
##   Accidental     Homicide      Suicide Undetermined
##  0.017681159  0.300000000  0.674347826  0.007971014
```

While the proportions of the various intents are fairly close with and without February, looking at February in isolation is a little more telling; it appears that the February-only dataset has a noticeably different proportion of homicides vs. Suicides.

Digging Further into "intent"

Let's build a dataframe entirely around "intent" data and see if we notice any interesting patterns.

```
# make a dataframe of the summary of the first month
d.intent <- d[which(d$month == 1), ]

# replace it with it's summary
d.intent <- summary(d.intent$intent)

# add the rest of the months iteratively
for(i in 2:12) {

  # get the month
  current.month <- d[which(d$month == i), ]

  # add the summary of it to the end of the df
  d.intent <- rbind(d.intent, summary(current.month$intent))
}

# create a table of d.intent
kable(d.intent)
```

	Accidental	Homicide	Suicide	Undetermined
d.intent	149	2682	5155	70
	122	2070	4653	55
	128	2629	5256	65
	96	2687	5352	72
	114	2813	5421	72
	112	2946	5292	68
	144	3095	5443	55
	162	2936	5341	73
	115	2828	5276	80
	125	2831	5174	51
	158	2769	5022	76
	173	3043	4906	60

Keep in mind we need to adjust for the variation in the length of the months:

Adjusted and Scaled "intent" Data

We will now adjust the intent data to account for the difference in the months buy dividing the number of each type of deaths by the number of days their month has in the dataset. For months that aren't February, this is simply 3 times the days in the month (times 3 because there are 3 years in the dataset). For February it's (28 x 2) + 29 because of our leap year.

```r
# convert to full-on dataframe
d.intent <- as.data.frame(d.intent)

# copy frame to manipulate
d.intent.adj <- d.intent

# for each row in the dataframe...
for(i in 1:nrow(d.intent.adj)) {
  # is it isn't Feb,
  if(i != 2) {
    # divide it by 3 times it's days value,
    d.intent.adj[i, ] <- d.intent.adj[i, ] / (days[i] * 3)
  }
  else {
    # if it is is Feb. do the same but with a leap year
    d.intent.adj[i, ] <- d.intent.adj[i, ] / (29 + 28 + 28)
  }
}

# scale the data
d.intent.scaled <- scale(d.intent.adj)

# add the months back
d.intent.scaled <- cbind(months, d.intent.scaled)

# make a table
kable(d.intent.scaled) %>%
  row_spec(2, bold = TRUE, color = "blue")
```

	months	Accidental	Homicide	Suicide	Undetermined
d.intent	Jan	0.592079711540243	-0.629311895307561	-0.694577169566176	0.243365896226467
	Feb	**-0.0864609143488781**	**-2.46718884407062**	**-1.0353716497167**	**-0.782805914766325**
	Mar	-0.326187928499242	-0.86280415853874	-0.157353406787581	-0.278937121344714
	Apr	-1.58552640626768	-0.212697083491931	1.30219313288021	0.702992551689108
	May	-0.938366355192233	-0.0521895088304981	0.720289373989333	0.45228710325494
	Jun	-0.86257283569691	0.966365420585512	0.972412209194336	0.271222057163597
	Jul	0.373444559149889	1.1901655521354	0.837308411426253	-1.32354315648708
	Aug	1.16053110775516	0.48968876244186	0.294765601491435	0.556747706769177
	Sep	-0.727019041214891	0.429186364673781	0.884470629544771	1.56653354074013
	Oct	-0.457369019933455	0.0271097503800915	-0.593515273597926	-1.74138557054403
	Nov	1.21591867969405	0.160596836717916	-0.511601947392081	1.13476304621462
	Dec	1.64152844301394	0.961078803304805	-2.01901991146588	-0.801240138915897

It appears that February has fewer deaths overall, but much more noticeably in Homicides.

Outside Research and Comments

After all this I'm considering February "suspicious", and doing some further investigation. A little online browsing reveals the following:

http://www.baltimoresun.com/news/maryland/crime/bs-md-ci-february-homicides-20180301-story.html

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/

There are a few results discussing this as part of a nationwide decrease in gun deaths, and some talking about isolated observations of February (and some January) guns deaths decreasing, but none about this specifically. It makes me wonder if the pattern (if genuine) held true in 2018 and the years between 2014 and 2018. Or if there is a discrepancy in the collection/recording methods or an error in the dataset somewhere. Have you heard anything about this? What would you look at next?

Quality Control

Some tests to make sure that the code did what it was supposed to.

```r
# for each month, is the total of the summary the same as in the dataset?
for(i in 1:12) {
  # does the total from the dataset match our summary total?
  print(length(which(d$month == i)) == sum(d.intent[i, ]))
}
```

```
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
```

```r
# do they data by year totals add up to the total length?
sum(c(nrow(data12) + nrow(data13) + nrow(data14))) == nrow(d)
```

```
## [1] TRUE
```

```r
# do the extracted homicide counts match the dataset?
for(i in 1:12) {
  # does the count of that mounth were intent is homicide match extracted value?
  print(length(which(d$month == i & d$intent == "Homicide")) == d.intent$Homicide[i])
}
```

```
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
```