Preliminary Results of an Anaylsis of Seasonal Patterns in Gun Deaths Including a Drop in Februaries

Thadryan J. Sweeney

Introduction

Last year, I read a project by FiveThirtyEight examining CDC gun deaths data for the years 2012-14. The dataset is publicly available so I decided to look through it myself out of curiosity. When I plotted the gun deaths over time, I noticed a suspicious looking dip in February of each year. I did a little digging and found that seasonality-related issues had gotten a little media attention here, here, here, and here. We'll visualized the dataset at a high level before investigating seasonal trends, with a particular interests in drops during February vs other winter months.

```r
library(tidyverse)    # obviously
library(outliers)     # chi-squared
library(reshape2)     # melt dataframe
library(kableExtra)   # pretty output
library(DataExplorer) # streamlined exploratory analysis
library(ggpubr)       # plot density
library(ggthemes)     # color schemes for ggplot
library(e1071)        # skewness
library(fma)          # seasonality
```

We begin by reading in the dataset, inspecting the first few rows, summarizing it, and getting a sense of where the missing values are.

```r
# read in the data, inspect and summaraize
dRaw <- read.csv("Data/full_data.csv", stringsAsFactors = FALSE)

# look at the first few rows
kable(head(dRaw)) %>% kable_styling()
```

X	year	month	intent	police	sex	age	race	hispanic	place	educa
1	2012	1	Suicide	0	M	34	Asian/Pacific Islander	100	Home	BA+
2	2012	1	Suicide	0	F	21	White	100	Street	Some
3	2012	1	Suicide	0	M	60	White	100	Other specified	BA+
4	2012	2	Suicide	0	M	64	White	100	Home	BA+
5	2012	2	Suicide	0	M	31	White	100	Other specified	HS/G
6	2012	2	Suicide	0	M	17	Native American/Native Alaskan	100	Home	Less t

```r
# typical summary
dim(dRaw)
```

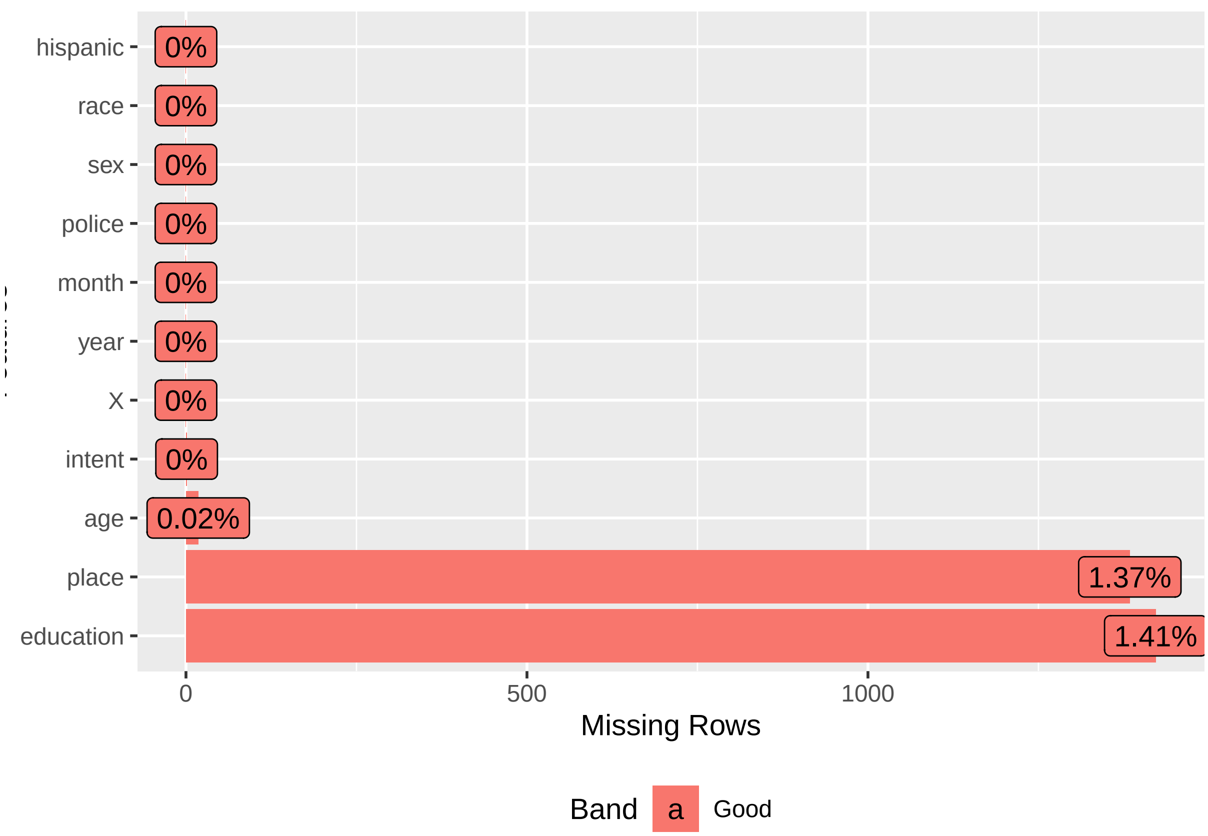
```
## [1] 100798     11
```

```r
summary(dRaw)
```

```
##        X               year          month           intent
##  Min.   :     1   Min.   :2012   Min.   : 1.000   Length:100798
##  1st Qu.: 25200   1st Qu.:2012   1st Qu.: 4.000   Class :character
##  Median : 50400   Median :2013   Median : 7.000   Mode  :character
##  Mean   : 50400   Mean   :2013   Mean   : 6.568
```

```
##  3rd Qu.: 75599   3rd Qu.:2014   3rd Qu.: 9.000
##  Max.   :100798   Max.   :2014   Max.   :12.000
##
##      police            sex                 age              race
##  Min.   :0.00000   Length:100798      Min.   :  0.00   Length:100798
##  1st Qu.:0.00000   Class :character   1st Qu.: 27.00   Class :character
##  Median :0.00000   Mode  :character   Median : 42.00   Mode  :character
##  Mean   :0.01391                      Mean   : 43.86
##  3rd Qu.:0.00000                      3rd Qu.: 58.00
##  Max.   :1.00000                      Max.   :107.00
##                                       NA's   :18
##     hispanic        place            education
##  Min.   :100.0   Length:100798      Length:100798
##  1st Qu.:100.0   Class :character   Class :character
##  Median :100.0   Mode  :character   Mode  :character
##  Mean   :114.2
##  3rd Qu.:100.0
##  Max.   :998.0
##
```

```
# get proportions of missing values
plot_missing(dRaw)
```




```
# what percentage of the data set do we keep if we simply drop NAs?
nrow(na.omit(dRaw))/nrow(dRaw)
```

```
## [1] 0.9723903
```

This suggests a fairly large data set without a lot of missing values. For simplicity, we will simply drop rows where there is information missing (more on this later).

Overview of the Dataset

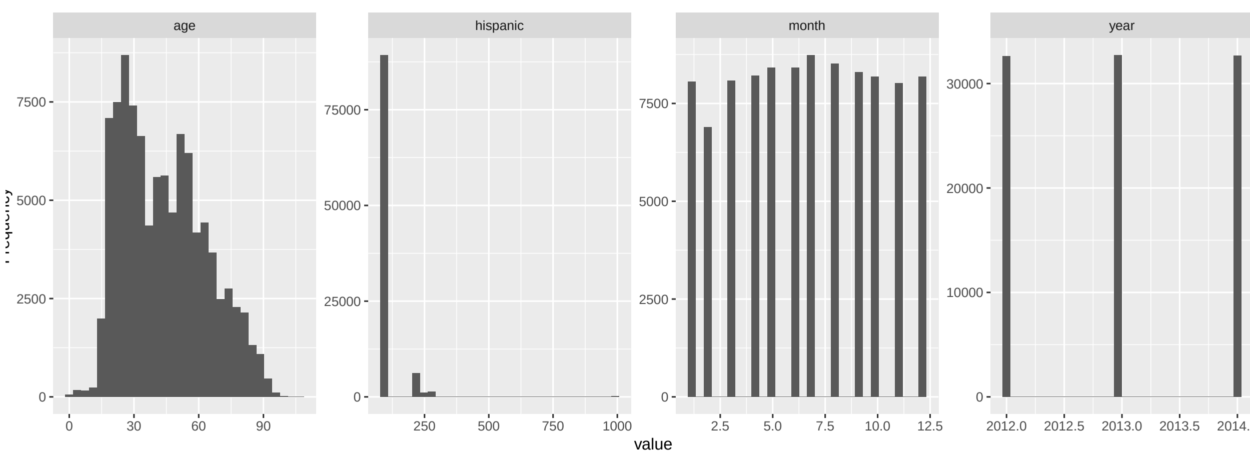
We'll clean up the dataset a bit and familiarize ourselves with its features before looking at the deaths over time.

```
# remove incomplete rows and the X column
d <- na.omit(dRaw[, 2:(ncol(dRaw))])

# convert police to factor
d$police[d$police == 1] <- "yes"
d$police[d$police == 0] <- "no"
d$police <- as.factor(d$police)

kable(head(d)) %>% kable_styling()
```

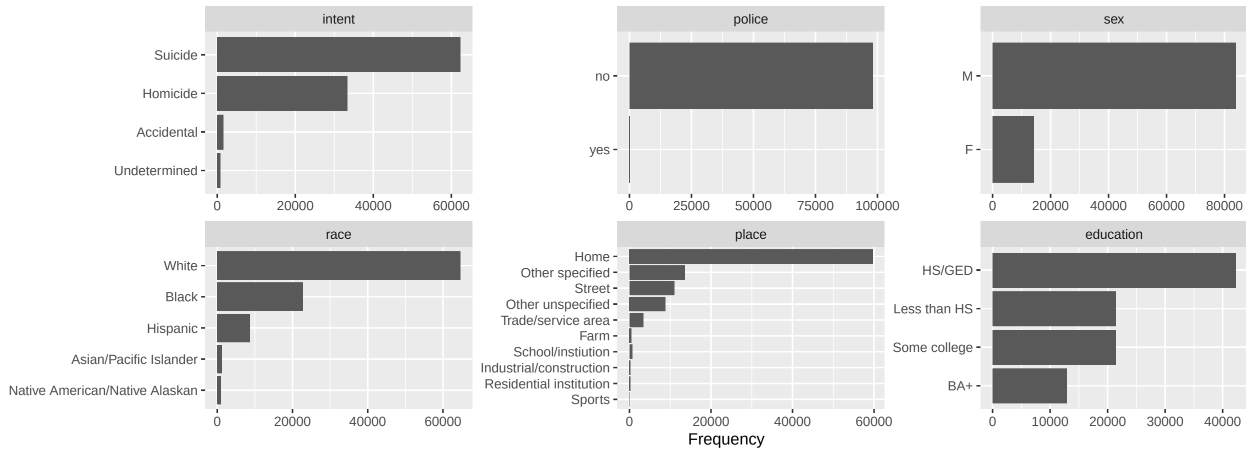
year	month	intent	police	sex	age	race	hispanic	place	education
2012	1	Suicide	no	M	34	Asian/Pacific Islander	100	Home	BA+
2012	1	Suicide	no	F	21	White	100	Street	Some colle
2012	1	Suicide	no	M	60	White	100	Other specified	BA+
2012	2	Suicide	no	M	64	White	100	Home	BA+
2012	2	Suicide	no	M	31	White	100	Other specified	HS/GED
2012	2	Suicide	no	M	17	Native American/Native Alaskan	100	Home	Less than

The DataExplorer packages gives ready made visualizations. Let's get some high level summaries of the categories and see what's worth taking a closer look at.

```
# continuous
plot_histogram(d)
```




```
# categorical
plot_bar(d)
```



It looks like the Hispanic column is redundant and contains little variation. The age of the subjects seems to exhibit some patterns, as does the deaths per months. There is little variation from year to year.

Let's write some custom functions to look at the categories in more nuanced way.

```
# basic plot for our continous variables
plotContinuous <- function(df, colString, annotate = FALSE)
{
  # create a histogram
  p <- ggplot(df, aes(df[, colString])) +
        geom_histogram(fill = "Dark Grey", binwidth = 1, col = "Black") +
        theme_economist() +
        scale_color_economist() +
        xlab(colString)

    # allow addition of an annotation and mean line if desired
    if(isTRUE(annotate)) {
      p <- p + geom_vline(xintercept = mean(d[, colString]), color = "Dark Blue" ) +
               ggtitle(paste("mean:", round(mean(d[, colString]), 2)))
    }
  p
}

# customized plotes for categorical variables
plotCategorical <- function(df, colString)
{
  # create a bar plot
  ggplot(df, aes(df[, colString])) +
    geom_bar(fill = "Dark Grey", col = "Black") +  xlab(colString) +
    theme_economist() +
    scale_color_economist() +
    theme(axis.text.x = element_text(angle = 75, hjust = 0, size = 12)) +
    scale_x_discrete(label = function(x) abbreviate(x, minlength = 10))
}
```

We're ready to zoom in on a few things.

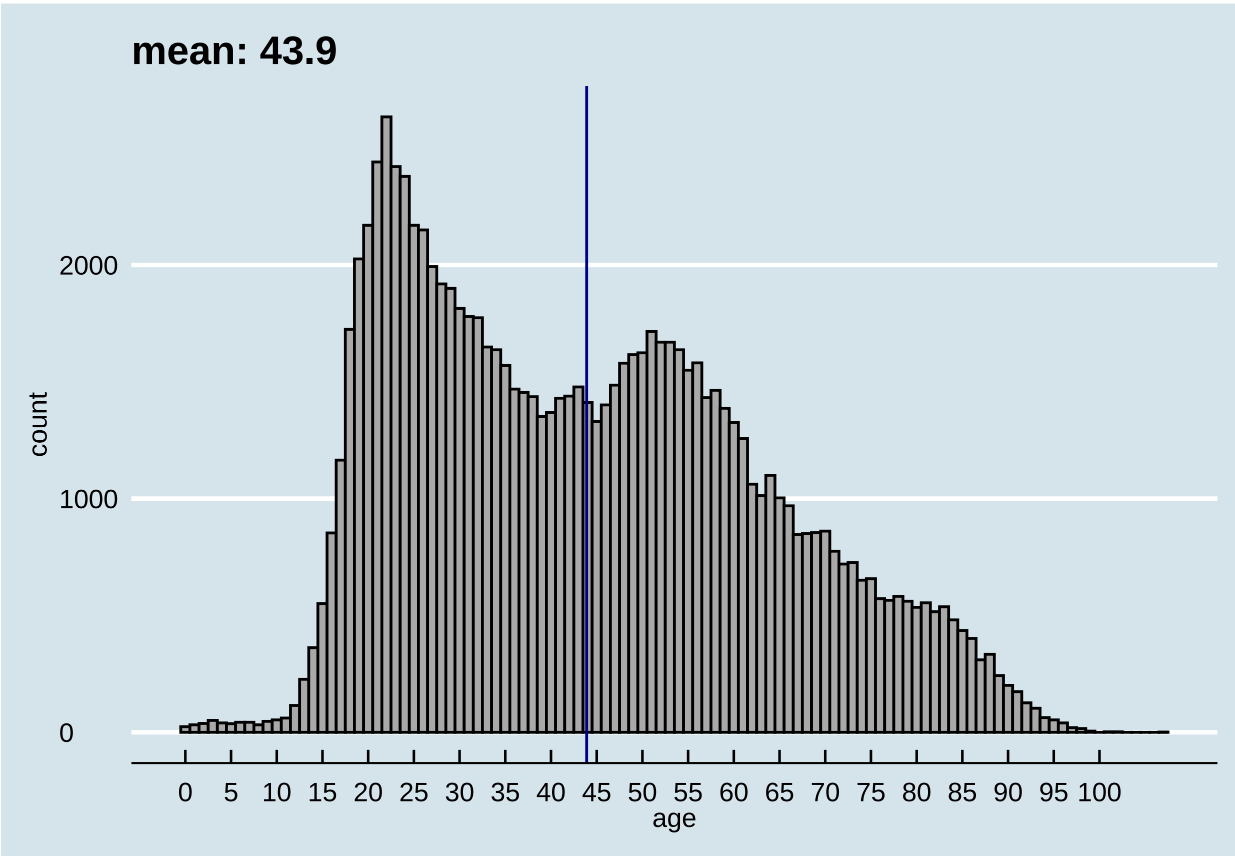
Inspect Continuous Features

We will start with the continuous variables.

Age

A closer looks at the ages of the subjects:

```
plotContinuous(d, c("age"), annotate = TRUE) +
  scale_x_continuous(breaks = seq(0, 100, by = 5))
```

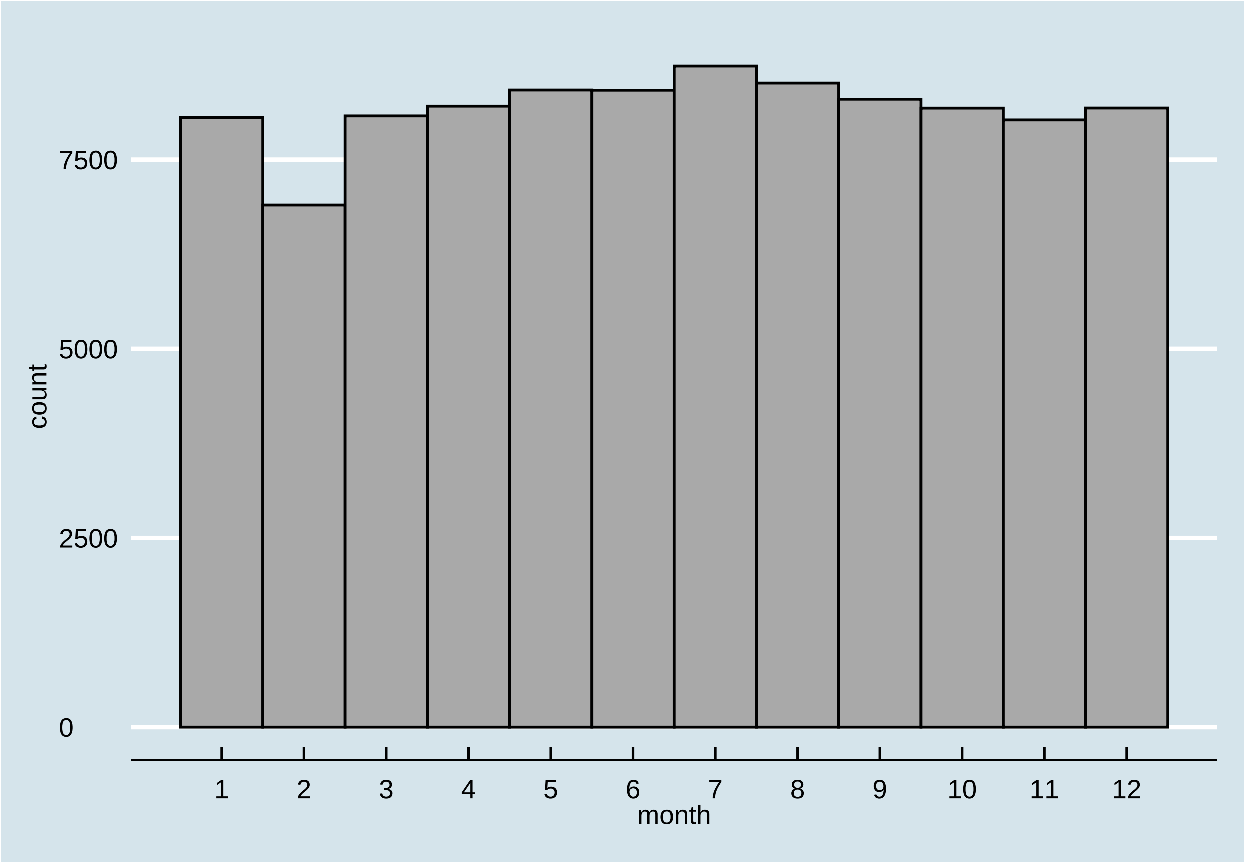


The amount of deaths by age sees a pronounced spike in early 20's, drops in the 30's, and rises again in the first half of the 50's.

Month

Deaths by month:

```
plotContinuous(d, c("month")) + scale_x_continuous(breaks = seq(from = 1, to = 12, by = 1))
```

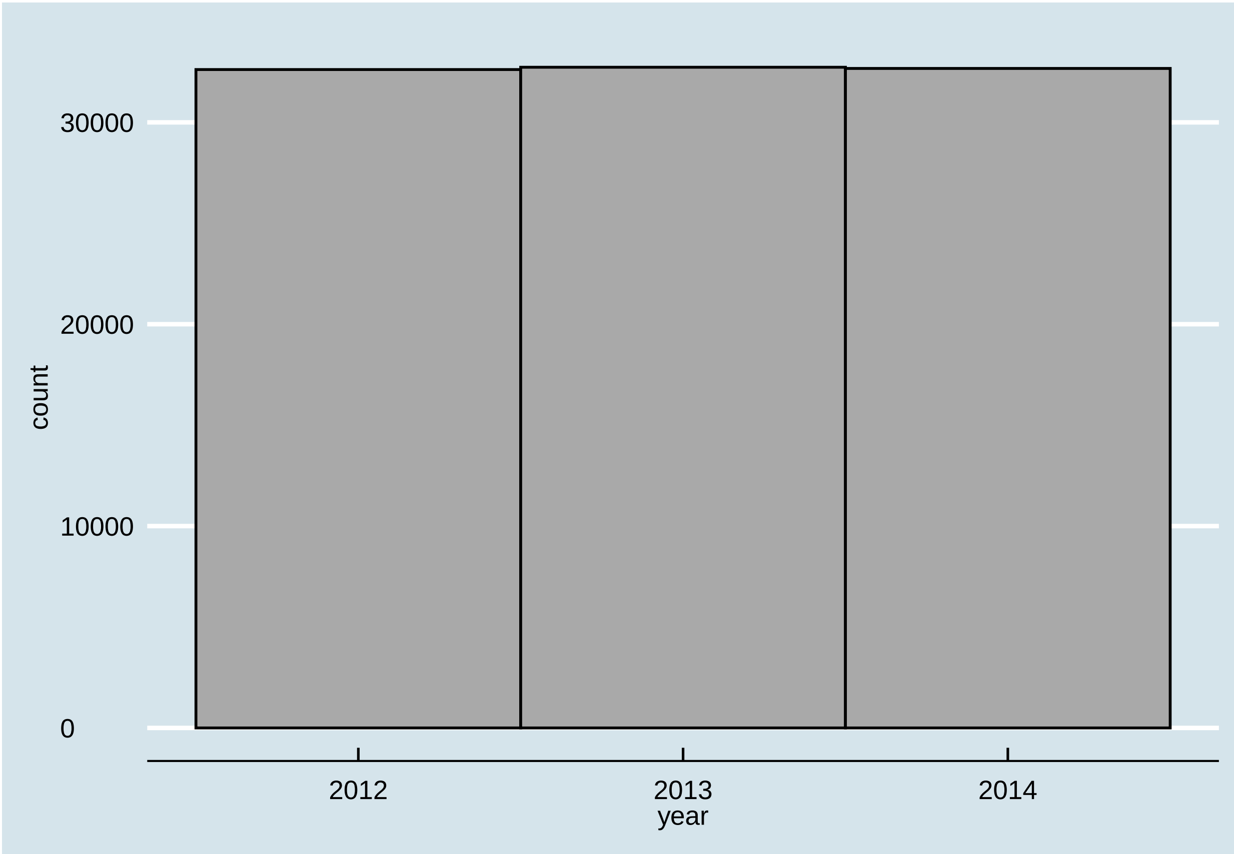


There appears to be a drop in February that may be a trend or an error in the data collection. Deaths curve up slightly in the summer time.

Year

Deaths over the years in the dataset:

```
plotContinuous(d, c("year"))
```



Year to year variation is basically null.

Continuous Features Takeaway

Age related trends emerge in the 20's and 50's (increase). There appears to be a dip in gun deaths in February, and a slight upward trend through the summer. The years in our dataset are very similar in totals.

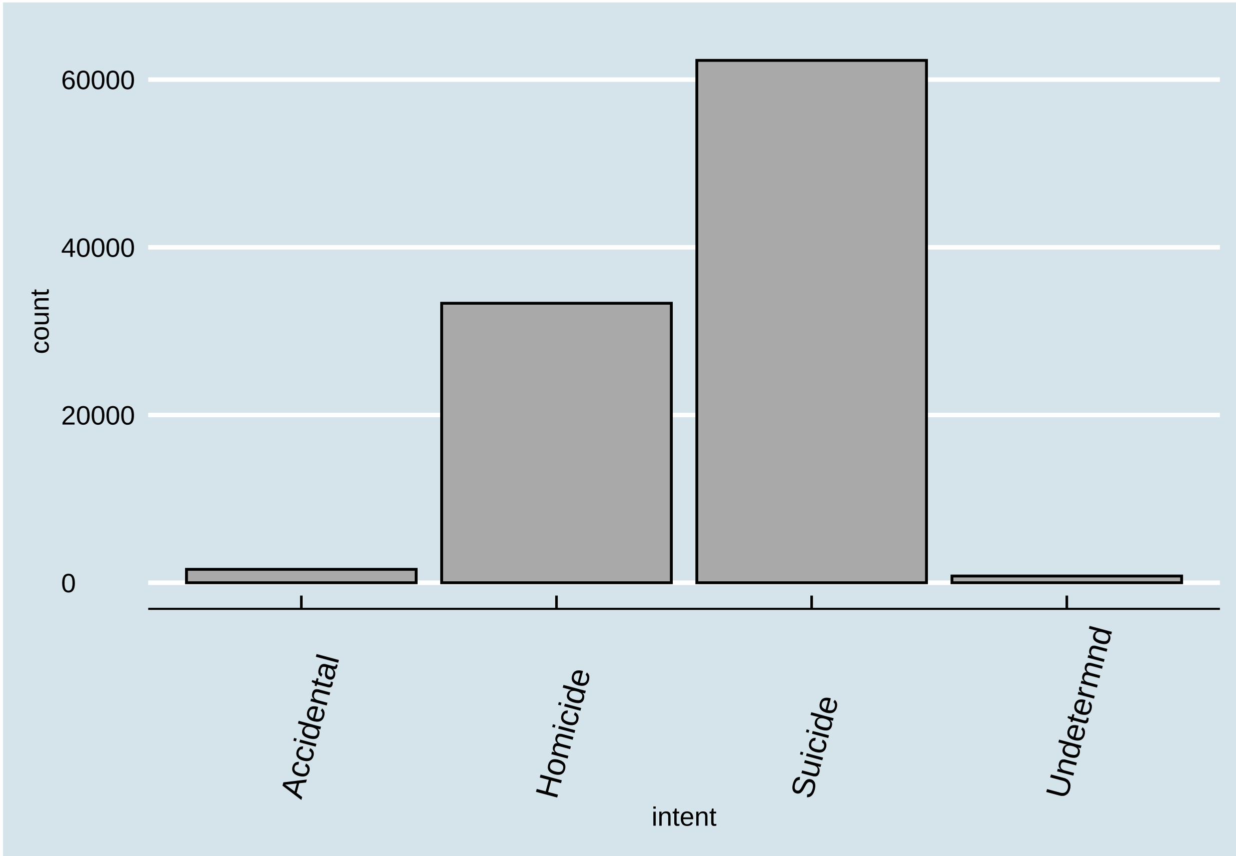
Inspect Categorical Features

An overview of the categorical features.

Intent

The determined intent behind the fatality:

```
# intent of gun death
plotCategorical(d, c("intent"))
```

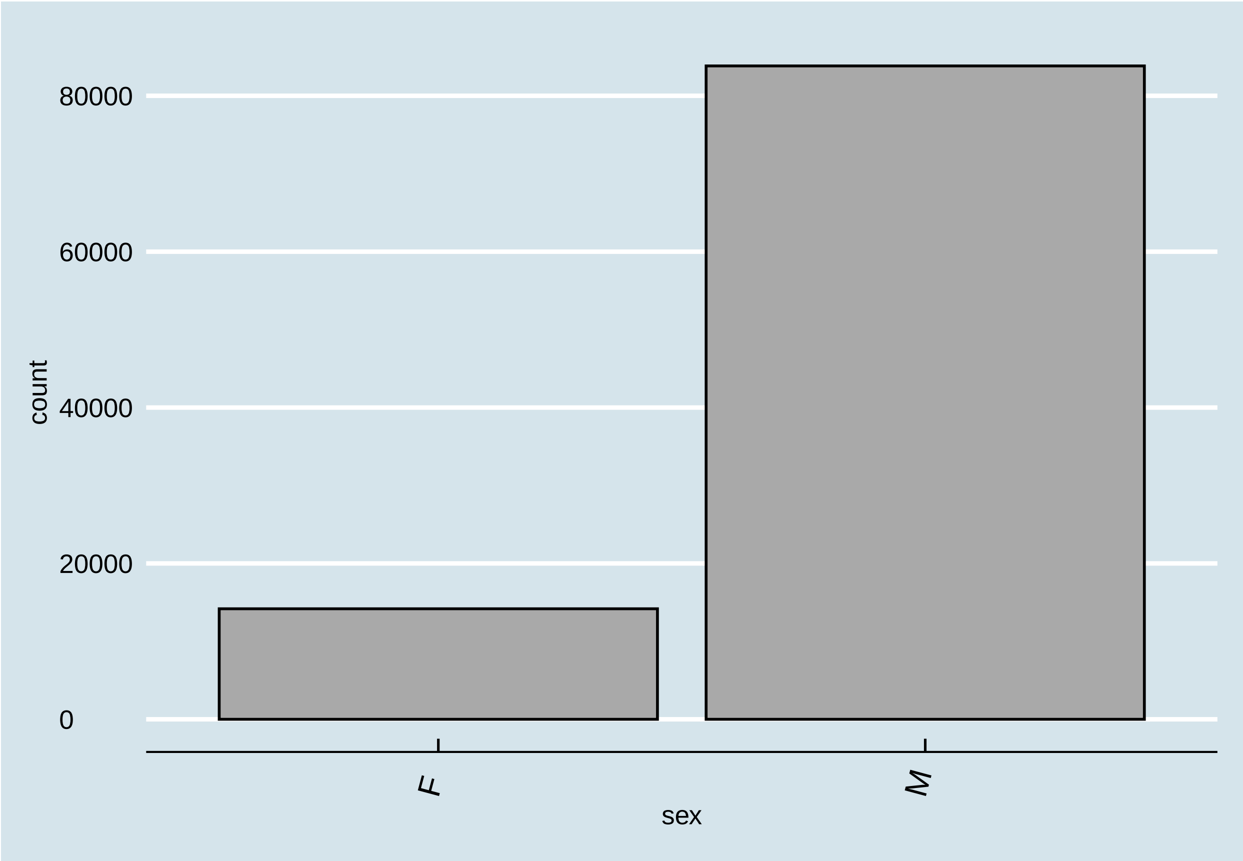


Very few of our observations are accidental or undetermined. Suicides readily outnumber homicides.

Sex

The sex of the subject:

```
# ditribution of sex
plotCategorical(d, c("sex"))
```

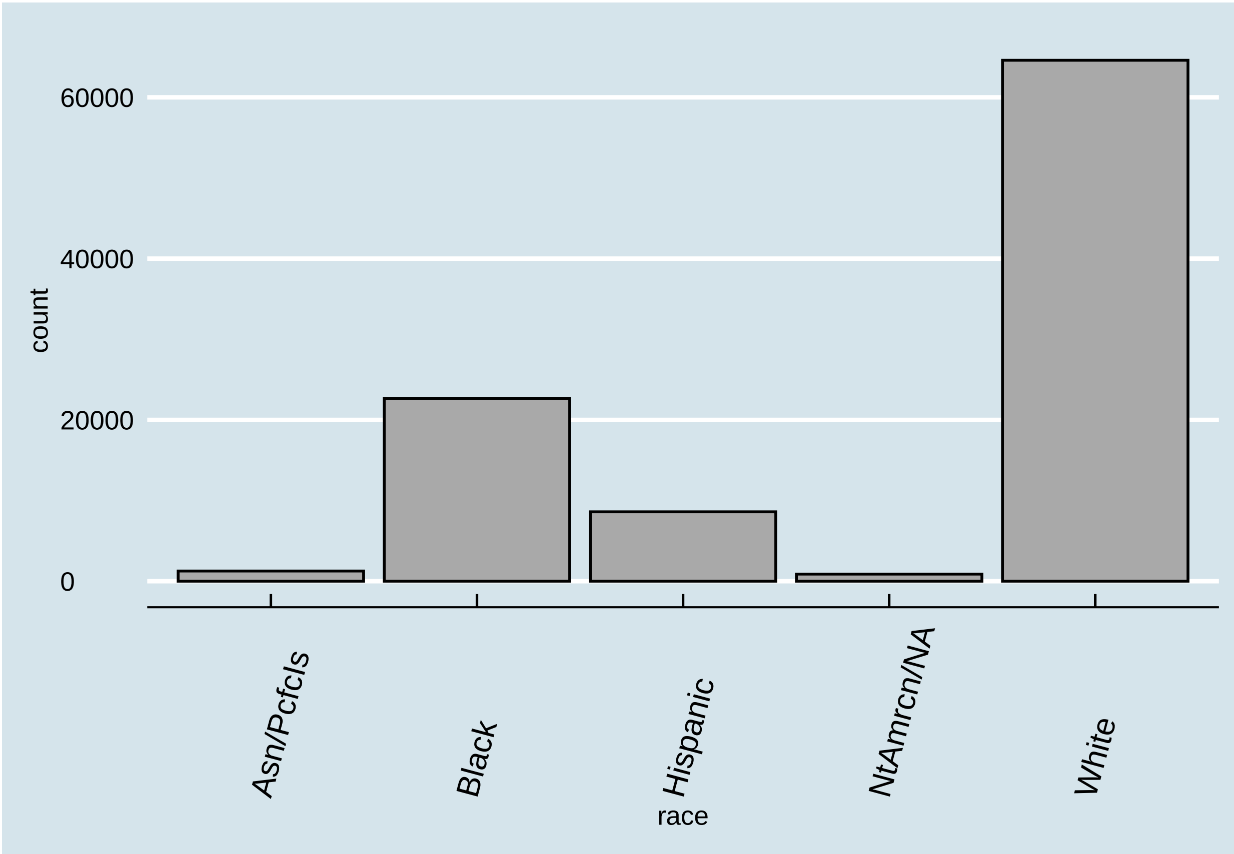


The dataset is strongly male dominated.

Race

The race of the subject:

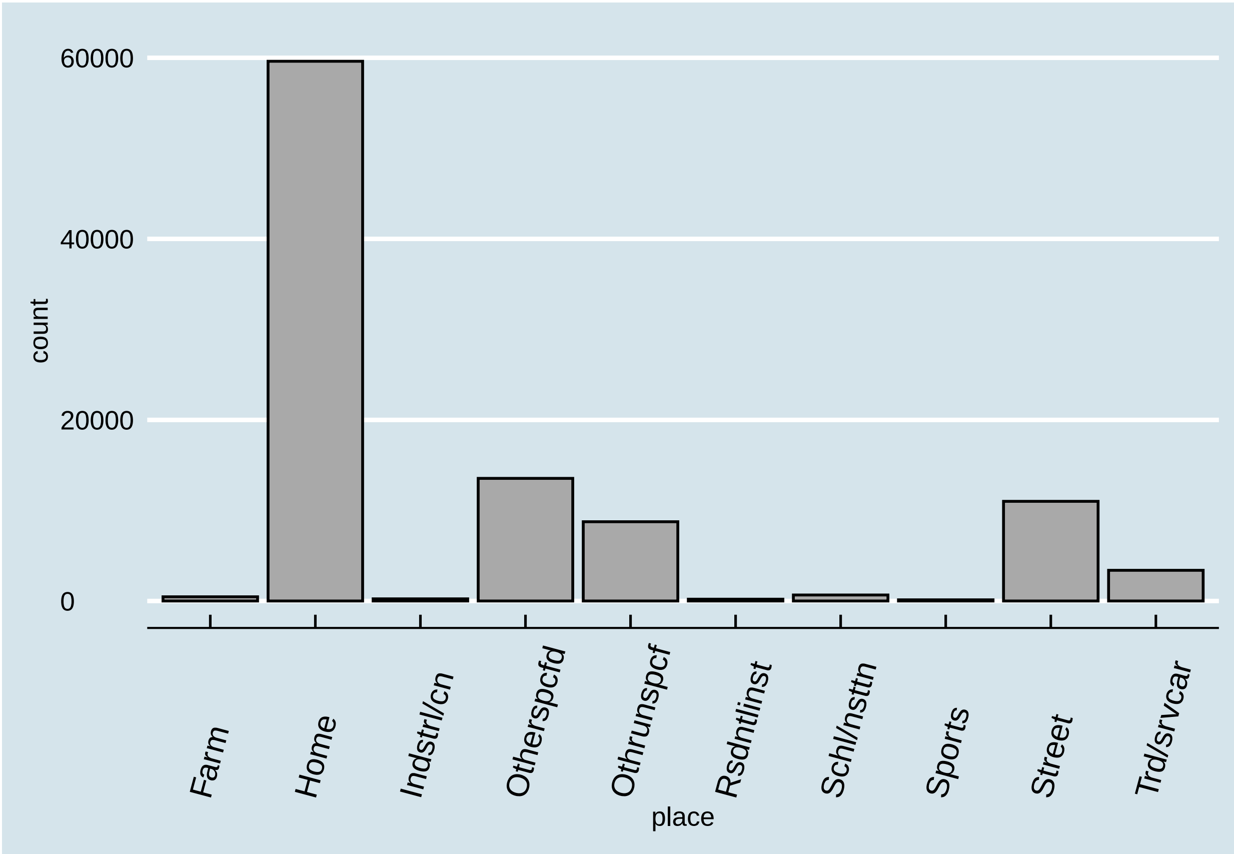
```
# distribution of race
plotCategorical(d, c("race"))
```



Most deaths are of White subjects, followed by Black then Hispanic. Native America and Asian deaths are much less common.

Place

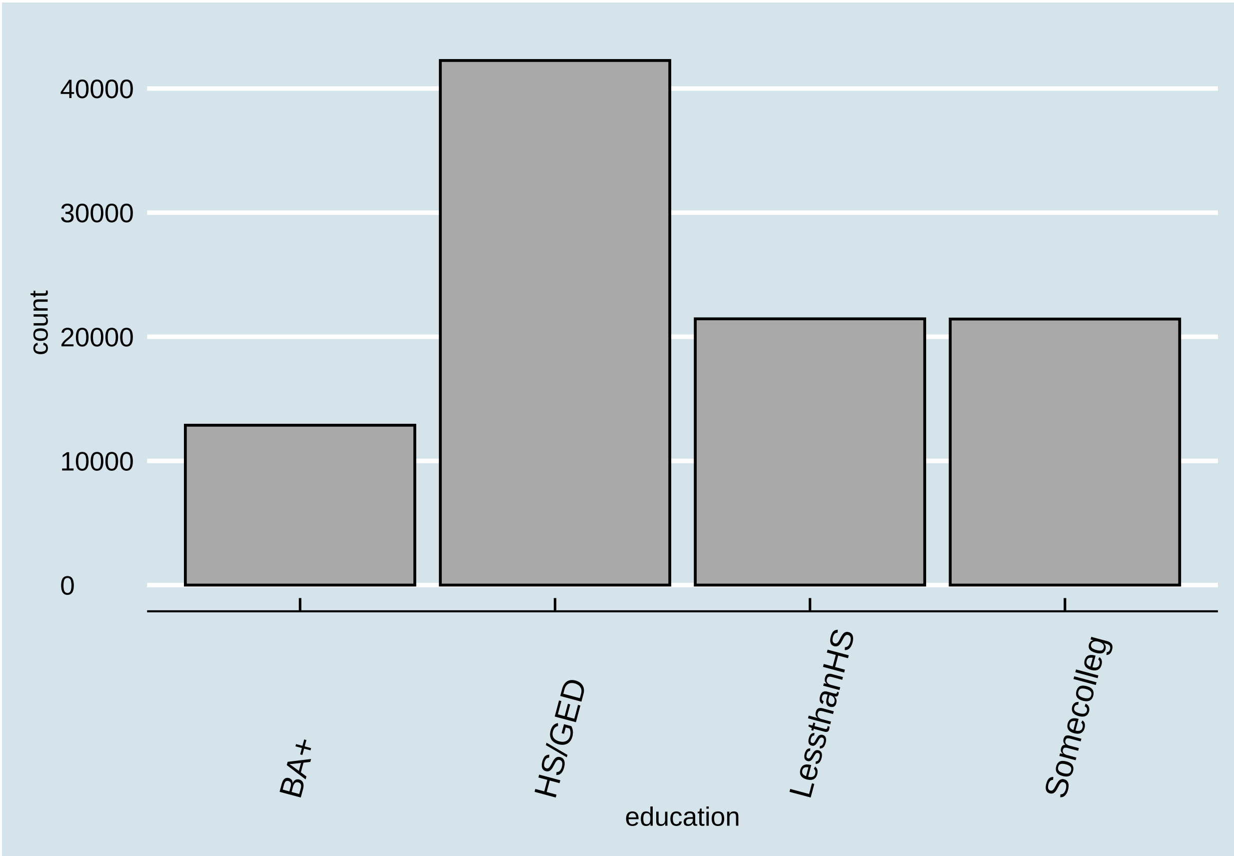
```
# location of shooting
plotCategorical(d, c("place"))
```



Most deaths take place in the home, without a strong second-place candidate.

Education

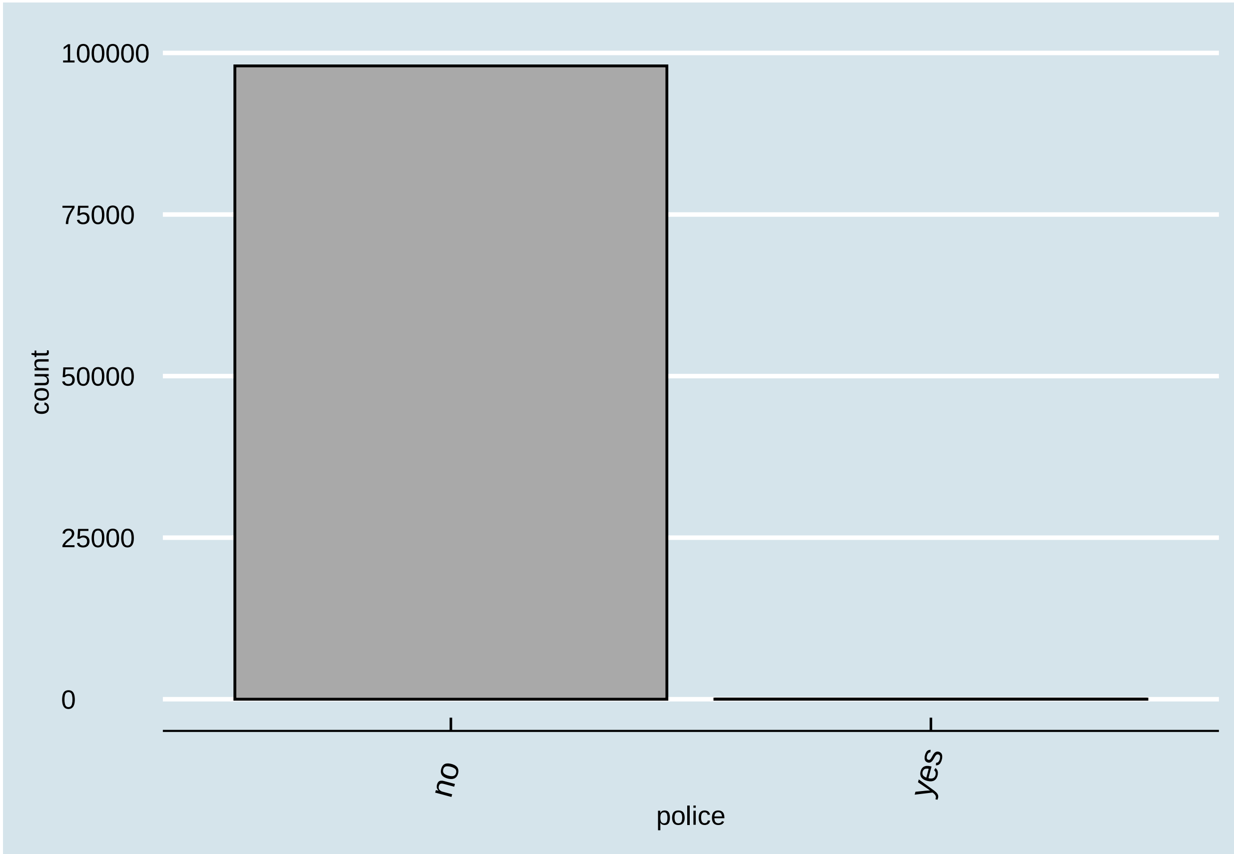
```
# education level of subject
plotCategorical(d, c("education"))
```



HS/GED educated subjects make up our largest groups, with college-educated subjects the smallest.

Police

```
# police involvment
plotCategorical(d, c("police"))
```



Very few of our observations involve Police.

Continuous Features Takeaway

The demographics of our dataset: race is predominantly white, overwhelmingly male, mostly of high school education. Suicides make up the majority of deaths. Most deaths occur inside the home, and do not involve police.

The Question of February

Having gotten familiar with the content of our dataset, we will begin our investigation of February-specific trends. Let's extract the data by year:

```
# sequence of 1-12
monthNumbers <- seq(from = 1, to = 12, by = 1)

# subset the deaths by year and count them by month, bind into dataframe
dDeathsByMonthByYear <- data.frame(
  cbind(
    monthNumbers,
    d %>% filter(year == 2012) %>% group_by(month) %>% count %>% .$n,
    d %>% filter(year == 2013) %>% group_by(month) %>% count %>% .$n,
    d %>% filter(year == 2014) %>% group_by(month) %>% count %>% .$n
  )
)

# set sensible column names
```

```
colnames(dDeathsByMonthByYear) <- c("month", "yr2012","yr2013","yr2014")

kable(dDeathsByMonthByYear) %>% kable_styling()
```

month	yr2012	yr2013	yr2014
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 now have an organized count by each month, and can plot them over the years in the data. Because the months vary in their number of days, we will also scale the counts to what they would have had if they were all 31 days long (IE, February deaths in a non-leap year are [x * (31/28))].

```
# creates a dataframe associating months with counts and years
dMelt <- melt(dDeathsByMonthByYear, id.vars = "month")

colnames(dMelt) <- c("month", "year", "deaths")

# inspect new frame
kable(head(dMelt)) %>% kable_styling()
```

month	year	deaths
1	yr2012	2695
2	yr2012	2281
3	yr2012	2674
4	yr2012	2719
5	yr2012	2921
6	yr2012	2730

```
# copy the melted data to scale it
dMeltNorm <- dMelt

# iterate over melted data
for(i in 1:nrow(dMeltNorm)){

  # if it's a month with 30 days, multiply deaths by 31/30
  if(dMeltNorm$month[i] %in% c(4,6,9,11)){
    dMeltNorm$deaths[i] <- dMeltNorm$deaths[i] * (31/30)

  # it is a february..
  }else if(dMeltNorm$month[i] == 2){
      # ... in a leap year
      if(dMeltNorm$year[i] == "yr2012"){
```

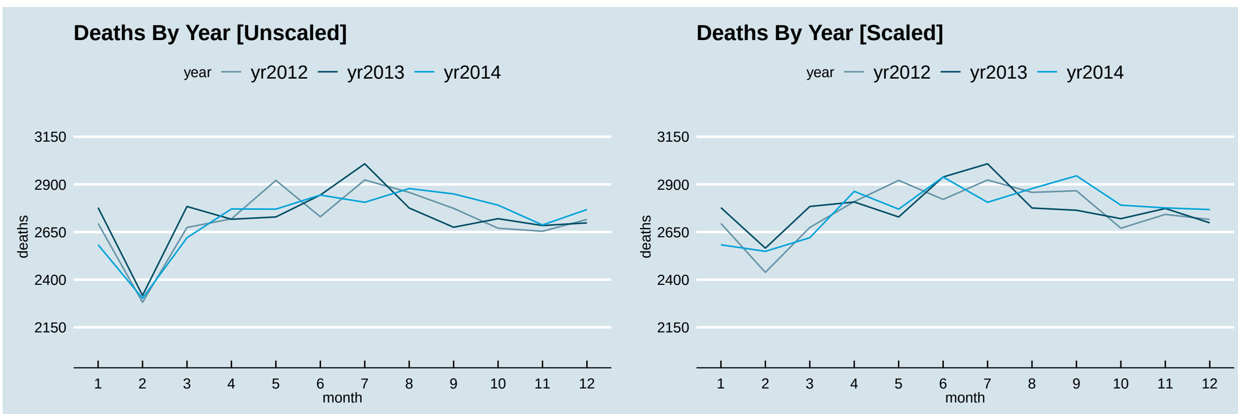
```
        dMeltNorm$deaths[i] <- dMeltNorm$deaths[i] * (31/29)
      } else {
        dMeltNorm$deaths[i] <- dMeltNorm$deaths[i] * (31/28)
    }
  }
}
kable(dMeltNorm) %>% kable_styling()
```

month	year	deaths
1	yr2012	2695.000
2	yr2012	2438.310
3	yr2012	2674.000
4	yr2012	2809.633
5	yr2012	2921.000
6	yr2012	2821.000
7	yr2012	2923.000
8	yr2012	2858.000
9	yr2012	2866.467
10	yr2012	2670.000
11	yr2012	2742.467
12	yr2012	2716.000
1	yr2013	2778.000
2	yr2013	2565.250
3	yr2013	2784.000
4	yr2013	2807.567
5	yr2013	2729.000
6	yr2013	2938.800
7	yr2013	3008.000
8	yr2013	2776.000
9	yr2013	2764.167
10	yr2013	2720.000
11	yr2013	2773.467
12	yr2013	2698.000
1	yr2014	2583.000
2	yr2014	2548.643
3	yr2014	2620.000
4	yr2014	2863.367
5	yr2014	2770.000
6	yr2014	2938.800
7	yr2014	2806.000
8	yr2014	2878.000
9	yr2014	2945.000
10	yr2014	2791.000
11	yr2014	2776.567
12	yr2014	2768.000

We will plot the scaled and unscaled data:

```
# plot the results on a line graph
deathsByYearPlot <-
  ggplotGrob(ggplot(dMelt, aes(month,deaths, col =  year)) +
    ggtitle("Deaths By Year [Unscaled]") +
```

```
    geom_line() +
    scale_y_continuous(limits = c(2000, 3250), breaks = seq(1650, 3350, by = 250)) +
    scale_x_continuous(breaks = monthNumbers) +
    scale_color_economist() + theme_economist())

# plot the results on a line graph
deathsByYearPlot31Scaled <-
  ggplotGrob(ggplot(dMeltNorm, aes(month,deaths, col =  year)) +
    ggtitle("Deaths By Year [Scaled]") +
    geom_line() +
    scale_y_continuous(limits = c(2000, 3250), breaks = seq(1650, 3350, by = 250)) +
    scale_x_continuous(breaks = monthNumbers) +
    scale_color_economist() + theme_economist())

ggarrange(deathsByYearPlot, deathsByYearPlot31Scaled)
```



There is a noticeable drop in total deaths in February in each year of the dataset. Before we assume there is something unusual about February, let's check for other reasons this could be happening.

February Missing Values

First, we make sure that the missing rows, while relatively few, don't cause the drop.

```
# what percent of the raw dataset is February
percentMissingFeb <- filter(dRaw, month == 2) %>% nrow/nrow(dRaw)

# what percent of the working dataset is February
percentCompleteFeb <- filter(d, month == 2) %>% nrow/nrow(d)

paste(percentMissingFeb, "vs", percentCompleteFeb, sep = " ")
```

```
## [1] "0.0703684596916605 vs 0.0703973881548743"
```

February takes up almost exactly the same proportion of the missing vs utilized dataset, suggesting there must be another reason for the drop.

Deaths per Day by Month in Dataset

Let's investigate the deaths by month as relates to the number of days the month accounts for in the dataset.

```
# calculate deaths total number of deaths per month
getDeaths <- function(df, monthNum, perDay = FALSE) { df %>% filter(month == monthNum) %>% nrow }
```

```r
# return the number of days of that month in the whole dataset
daysByMonth <- function(month)
{
  if(month %in% c(1,3,5,7,8,10,12)){
    return(31 * 3)
  }else if(month %in% c(4,6,9,11)){
    return(30 * 3)
  }else{
    # there is a leap year in the dataset
    return((28*3)+1)
  }
}

# get the number of deaths in each month
numberOfDeaths <- sapply(monthNumbers, getDeaths, df = d)

# number of deaths per day in that month across dataset ie, overall deaths in February
deathsPerMonth <- numberOfDeaths/sapply(monthNumbers, daysByMonth)

# z score of the deaths per month
zScoreDeathsPerMonth <- scale(deathsPerMonth)

# create a data frame of this information
dDeathsPerMonth <- data.frame(cbind(monthNumbers, numberOfDeaths,
                                    deathsPerMonth, zScoreDeathsPerMonth))

colnames(dDeathsPerMonth) <- c("month", "numberOfDeaths",
                               "overallDeathsByMonth", "zScoreOverallDeathsByMonth")

# print a pretty summary
kable(dDeathsPerMonth) %>% kable_styling(position = "center", full_width = TRUE)
```

month	numberOfDeaths	overallDeathsByMonth	zScoreOverallDeathsByMonth
1	8056	86.62366	-0.7746807
2	6900	81.17647	-2.2983847
3	8078	86.86022	-0.7085097
4	8207	91.18889	0.5023207
5	8420	90.53763	0.3201498
6	8418	93.53333	1.1581162
7	8737	93.94624	1.2736148
8	8512	91.52688	0.5968652
9	8299	92.21111	0.7882600
10	8181	87.96774	-0.3987087
11	8025	89.16667	-0.0633417
12	8182	87.97849	-0.3957010

By Z-score, February (-2.2983847) stands out in deaths adjusted by total number of days in the dataset, clocking in over a full standard deviation further from the mean that the next most deviant month (July, 1.2736148).

Statistical Test For Outliers

Let's apply some statistical heuristics for detecting outliers to see what sticks out.

```
# use scaled data
dDeathsByMonthByYear$yr2012 <- dMeltNorm[1:12, "deaths"]
dDeathsByMonthByYear$yr2013 <- dMeltNorm[13:24, "deaths"]
dDeathsByMonthByYear$yr2014 <- dMeltNorm[25:36, "deaths"]

kable(dDeathsByMonthByYear) %>% kable_styling()
```

month	yr2012	yr2013	yr2014
1	2695.000	2778.000	2583.000
2	2438.310	2565.250	2548.643
3	2674.000	2784.000	2620.000
4	2809.633	2807.567	2863.367
5	2921.000	2729.000	2770.000
6	2821.000	2938.800	2938.800
7	2923.000	3008.000	2806.000
8	2858.000	2776.000	2878.000
9	2866.467	2764.167	2945.000
10	2670.000	2720.000	2791.000
11	2742.467	2773.467	2776.567
12	2716.000	2698.000	2768.000

Chi-square Test

We will apply a simple Chi-square test for outliers to each year in the dataset.

```
# call chi-square outlier tests on each year in the dataset

# chisq for 2012
chisq.out.test(dDeathsByMonthByYear$yr2012,
               variance = var(dDeathsByMonthByYear$yr2012),
               opposite = FALSE)
```

```
##
##  chi-squared test for outlier
##
## data:  dDeathsByMonthByYear$yr2012
## X-squared = 5.5945, p-value = 0.01802
## alternative hypothesis: lowest value 2438.31034482759 is an outlier
```

```
# chisq for 2013
chisq.out.test(dDeathsByMonthByYear$yr2013,
               variance=var(dDeathsByMonthByYear$yr2013),
               opposite = FALSE)
```

```
##
##  chi-squared test for outlier
##
## data:  dDeathsByMonthByYear$yr2013
## X-squared = 4.2188, p-value = 0.03998
## alternative hypothesis: highest value 3008 is an outlier
```

```
# chisq for 2013 - lower range
chisq.out.test(dDeathsByMonthByYear$yr2013,
               variance=var(dDeathsByMonthByYear$yr2013),
               opposite = TRUE)
```

```
##
##  chi-squared test for outlier
##
## data:  dDeathsByMonthByYear$yr2013
## X-squared = 3.6439, p-value = 0.05628
## alternative hypothesis: lowest value 2565.25 is an outlier
```

```
# chisq for 2014
chisq.out.test(dDeathsByMonthByYear$yr2014,
               variance=var(dDeathsByMonthByYear$yr2014),
               opposite = FALSE)
```

```
##
##  chi-squared test for outlier
##
## data:  dDeathsByMonthByYear$yr2014
## X-squared = 2.9794, p-value = 0.08433
## alternative hypothesis: lowest value 2548.64285714286 is an outlier
```

This guideline tags February as the most notable outlier in 2012 and 2014. July surfaces again as the most deviant figure in 2013, though February comes up again if the function is instructed to look for the lowest outlier. The outlier package in R also allows for a p-value cutoff.

```
# 0.90
scores(dDeathsByMonthByYear$yr2012, type = "chisq", p = 0.90)
```

```
##  [1] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE
```

```
scores(dDeathsByMonthByYear$yr2013, type = "chisq", p = 0.90)
```

```
##  [1] FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## [12] FALSE
```

```
scores(dDeathsByMonthByYear$yr2014, type = "chisq", p = 0.90)
```

```
##  [1] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE
```

```
# 0.95
scores(dDeathsByMonthByYear$yr2012, type = "chisq", p = 0.95)
```

```
##  [1] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE
```

```
scores(dDeathsByMonthByYear$yr2013, type = "chisq", p = 0.95)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## [12] FALSE
```

```
scores(dDeathsByMonthByYear$yr2014, type = "chisq", p = 0.95)
```

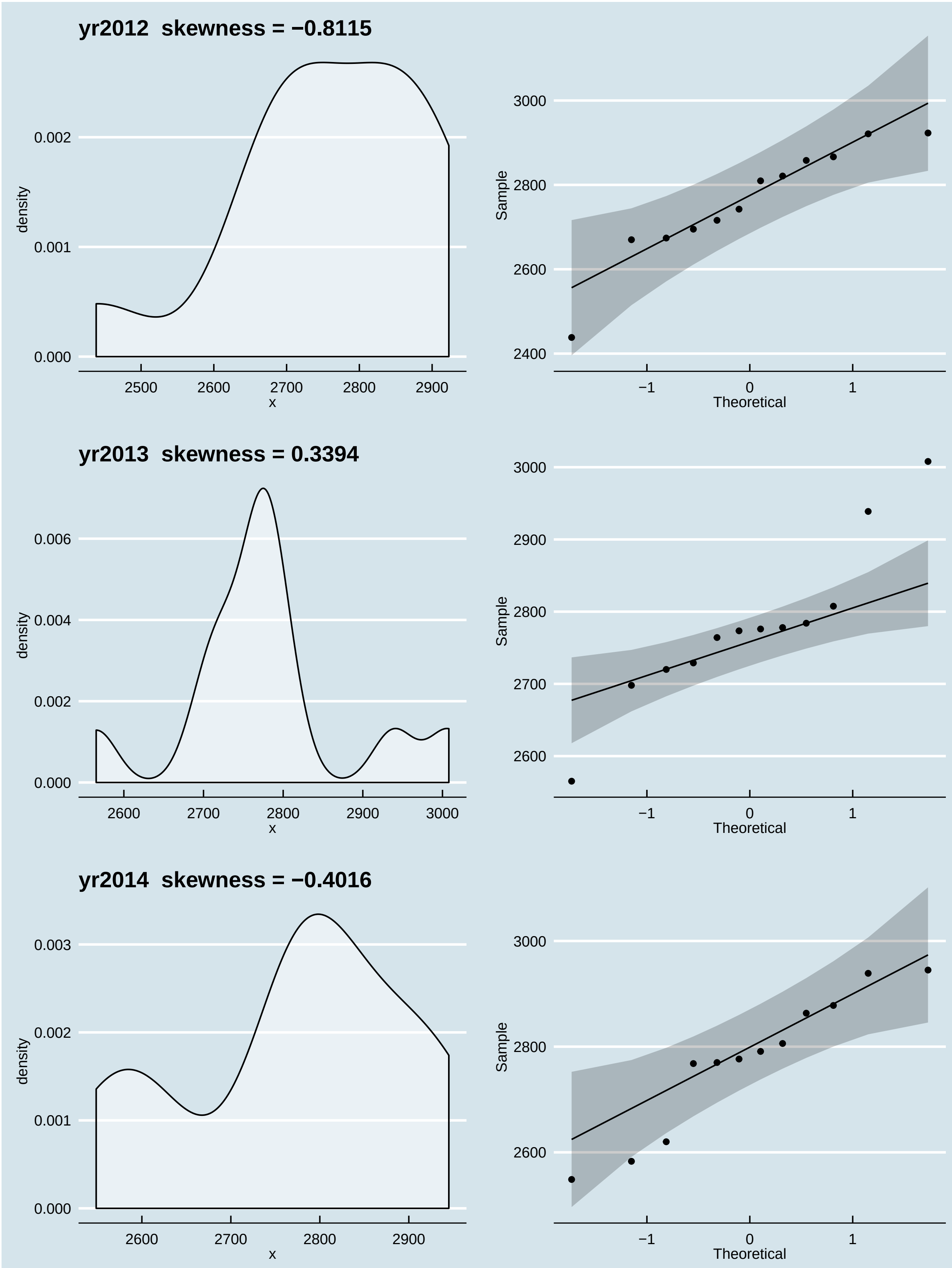
```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE
```

February passes a 0.90 cutoff in all three years and a 0.95 in 2012. July emerges again in two of the six tests as well. Interestingly, these five instances are the only ones flagged in either case. Next, let's get some estimates of the distribution of the data.

Distribution & Skewness of Deaths by Year

Let's inspect the distribution and skewness of the death in years using density & qqplots.

```
# custom wrapper for density
plotDensity <- function(df, year)
{
  ggplotGrob(
    ggdensity(
      df[[year]], main = paste(year, " skewness =", round(skewness(df[[year]]), 4))
    ) + theme_economist()
  )
}

# arrange 3x2 column of charts
ggarrange(

  # create density plots and qqplot for 2012
  plotDensity(dDeathsByMonthByYear, "yr2012"),

  ggplotGrob(
      ggqqplot(dDeathsByMonthByYear$yr2012) + theme_economist()
  ),


  # create density plots and qqplot for 2013
  plotDensity(dDeathsByMonthByYear, "yr2013"),

  ggplotGrob(
    ggqqplot(dDeathsByMonthByYear$yr2013) + theme_economist()
  ),


  # create density plots and qqplot for 2014
  plotDensity(dDeathsByMonthByYear, "yr2014"),

  ggplotGrob(
    ggqqplot(dDeathsByMonthByYear$yr2014) + theme_economist()
  ),

  nrow = 3, ncol = 2
)
```



The skew varies from year to year, with the qqplot suggesting largely normal distribution for 2012 and 2014 and a "heavy tailed" break from normality in 2013. We can further formalize this with a Shapiro-Wilk test:

```
shapiro.test(dDeathsByMonthByYear$yr2012)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dDeathsByMonthByYear$yr2012
## W = 0.90844, p-value = 0.2038
```

```
shapiro.test(dDeathsByMonthByYear$yr2013)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dDeathsByMonthByYear$yr2013
## W = 0.90799, p-value = 0.2011
```

```
shapiro.test(dDeathsByMonthByYear$yr2014)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dDeathsByMonthByYear$yr2014
## W = 0.91709, p-value = 0.2627
```

Given the p-values, we're unable to reject the null hypothesis the samples come from a normal distribution.

Modeling

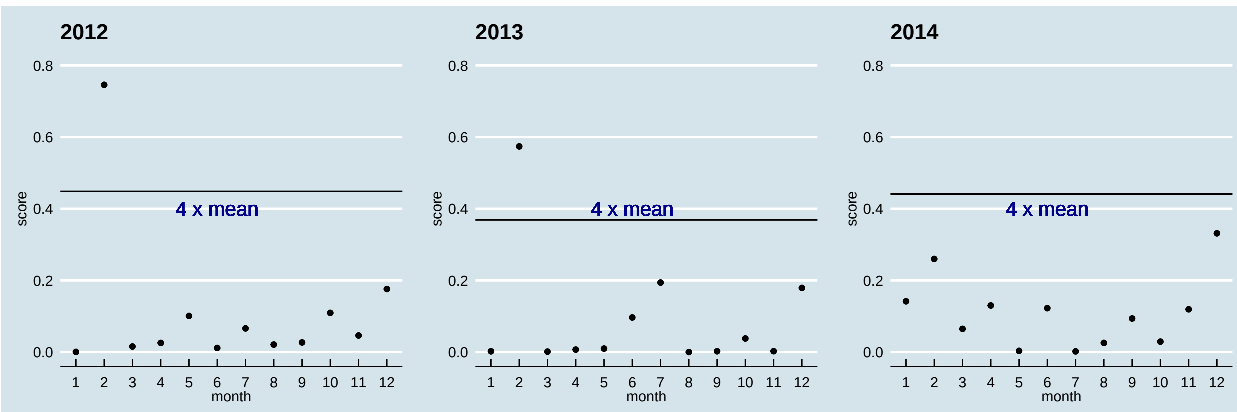
Lastly, let's train some linear models and examine the influence of the month on them by Cook's distance.

Cook's Distance

We'll write some code to plot the Cook's distance vs 4x the mean to see what emerges.

```
# custom Cook's distance plot code
plotCooksDistance <- function(df, name)
{
  # set months and names
  df$month <- seq(1,12,1)
  colnames(df) <- c("score", "month")
  # wrap in a rendering object for scaling purposes
  ggplotGrob(
    # make a point plot with the values of the model
    ggplot(df, aes(month, score)) +
      geom_point() +
      scale_y_continuous(limits = c(0,0.8)) +
      scale_x_continuous(breaks = seq(1,12,1)) +
      # denote 4 times the mean
      geom_hline(yintercept = mean(df$score * 4)) +
      geom_text(aes(x = 6, label = "4 x mean", y = 0.4),
                colour = "Dark Blue",
                angle = 0, size = 5) +
      ggtitle(name) +
      theme_economist() +
      scale_color_economist()
  )
```

```
}

# plot cook's distance charts for each year
ggarrange(
  plotCooksDistance(
    data.frame(cooks.distance(lm(yr2012 ~ month, dDeathsByMonthByYear))), "2012"
  ),
  plotCooksDistance(
    data.frame(cooks.distance(lm(yr2013 ~ month, dDeathsByMonthByYear))), "2013"
  ),
  plotCooksDistance(
    data.frame(cooks.distance(lm(yr2014 ~ month, dDeathsByMonthByYear))), "2014"
  ),
  nrow = 1, ncol = 3
)
```



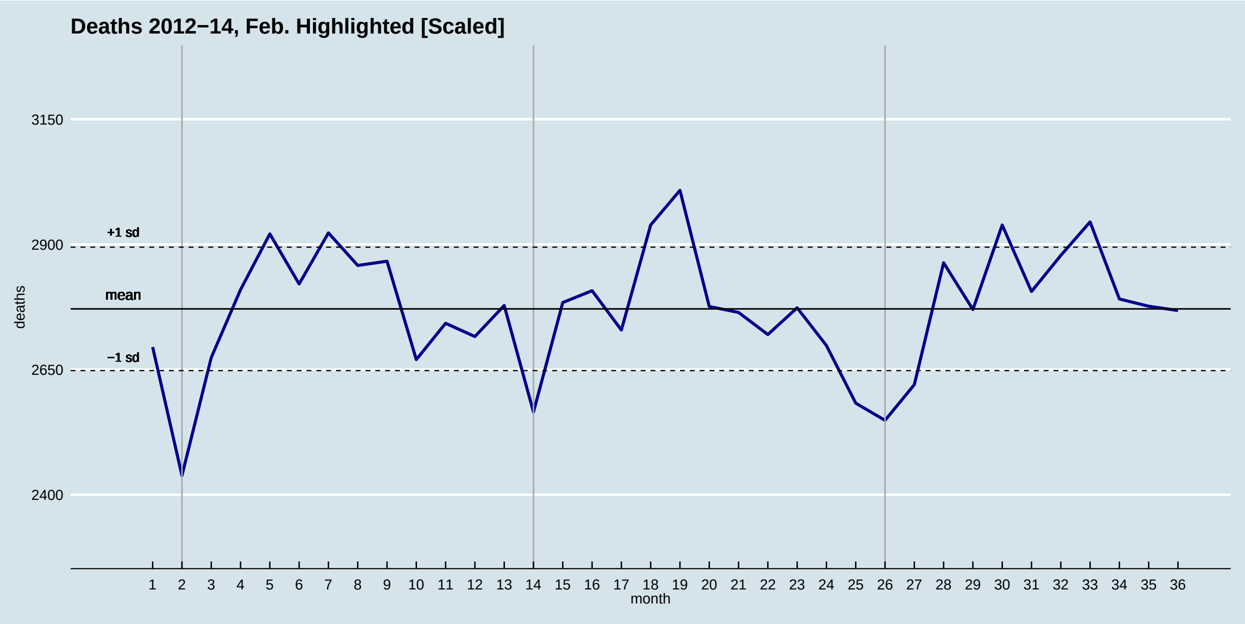
February easily exceeds 4 times the mean influence in a linear model by cooks distance in 2012, 2013, though no months do in 2014. Notably, February stands apart from other winter months.

Seasonality

Given we're looking at time data, we should also consider evidence of seasonality. It is frequently observed that crime rises in the summer. That arguably suggests that it returns to a baseline after the summer, it's less often noted that is rises in the summer *and* drops in the winter. This does not seem to be how the idea is framed (I get 85,300,000 hits on google when I search "crime rises in summer" vs only 20,000,000 for "crime drops in winter"). Perhaps the trends are less pronounced, leading to less media attention. To investigate, we will plot the deaths over the course of all three years in a row instead of over one another.

```
# copy the normalized melted data
dTimeSeries <- dMeltNorm

# set 3-year months
dTimeSeries$month <- seq(1,36,1)

# plot over 3 years
ggplot(dTimeSeries, aes(month, deaths)) +
  ggtitle("Deaths 2012-14, Feb. Highlighted [Scaled]") +
  geom_line(color = "Dark Blue", size = 1) +
  # highlight februaries
  geom_vline(xintercept = c(2,14,26), size = 0.5, color = "Dark Grey") +
  # denote the mean
```

```
  geom_hline(yintercept = mean(dTimeSeries$deaths)) +
  geom_hline(
    yintercept = mean(dTimeSeries$deaths) + sd(dTimeSeries$deaths),
    size = 0.40, linetype = "dashed"
  ) +
  geom_hline(
    yintercept = mean(dTimeSeries$deaths) - sd(dTimeSeries$deaths),
    size = 0.40, linetype = "dashed"
  ) +
  scale_y_continuous(limits = c(2300, 3250),
                     breaks = seq(1650, 3350, by = 250)) +
  scale_x_continuous(breaks = seq(1,36,1)) +
  # label mean and +/- standard deviations
  geom_text(aes(x = 0, label = "+1 sd", y = 2925),
            colour = "Black", angle = 0, size = 3) +
  geom_text(aes(x = 0, label = "mean", y = 2800),
            colour = "Black", angle = 0, size = 3.5) +
  geom_text(aes(x = 0, label = "-1 sd", y = 2675),
            colour = "Black", angle = 0, size = 3) +
  scale_color_economist() +
  theme_economist()
```




```
# look at summer vs winter deviants.
dSeasonVariants <- dMeltNorm %>%
  select(month, deaths) %>%
  mutate(zScore = scale(deaths)) %>%
  filter(abs(zScore) > 1) %>%
  arrange(month)

kable(dSeasonVariants) %>% kable_styling()
```

month	deaths	zScore
1	2583.000	-1.527256
2	2438.310	-2.701023
2	2565.250	-1.671249
2	2548.643	-1.805972
3	2620.000	-1.227100
5	2921.000	1.214705
6	2938.800	1.359104
6	2938.800	1.359104
7	2923.000	1.230930
7	3008.000	1.920476
9	2945.000	1.409401

Nearly as many winter months (5) stray more than a standard deviation from the mean as summer months, and they generally deviate futher when they do. This plot also suggests that a baseline could be hard to pin down - there is a six month period where the counts never vary more than one standard deviation (Aug of 2012 - Feb 2013) from the mean but it does not hold true in the other years. The other noticable steady pattern is just 4 months.

Statistical Test for Seasonality

A test for seasonality is described here. Essentially the idea is to train one model using a function that detects seasonality, if present, train another model specifying a non-seasonal method, and see if there is a statistically significant difference.

```
# train season model
seasonModel <- ets(ts(dTimeSeries$deaths, frequency = 12))

# aseasonal model
nonSeasonModel <- ets(ts(dTimeSeries$deaths, frequency = 12), model = "ANN")

# calculate significance
deviance <- 2*c(logLik(seasonModel) - logLik(nonSeasonModel))
df <- attributes(logLik(seasonModel))$df - attributes(logLik(nonSeasonModel))$df
1 - pchisq(deviance,df)
```

```
## [1] 2.301726e-06
```

The resulting p-value confirms our suspicion of seasonality.

Findings

In this dataset, the drop in the month of cannot be explained entirely due to it's smaller number of days. After scaling it is still flagged as an outlier in all three years, including as the most extreme outlier in two of the three years (90% probability cutoff). It also deviates strongly in overall deaths per day. In 2012 & 13 it easily exceeds 4x the mean in Cook's distance. No months do in 2014. Ultimately, more years are needed to confirm the possible trend is significant; this analysis cannot fully explain or reject the possible trend. The dataset exhibits statistically significant seasonality, with similar downward variation in the winter as upward in the summer.