R Notebook

Introduction

Project inspiring this project: https://fivethirtyeight.com/features/gun-deaths/

Data obtained: https://github.com/fivethirtyeight/guns-data

```
library(tidyverse)    # everything
```

```
## -- Attaching packages -------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ----------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(reshape2)     # melt dataframe
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(kableExtra)   # pretty output
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(DataExplorer) # streamlined exploratory analysis
library(ggthemes)     # color schemes for ggplot
```

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

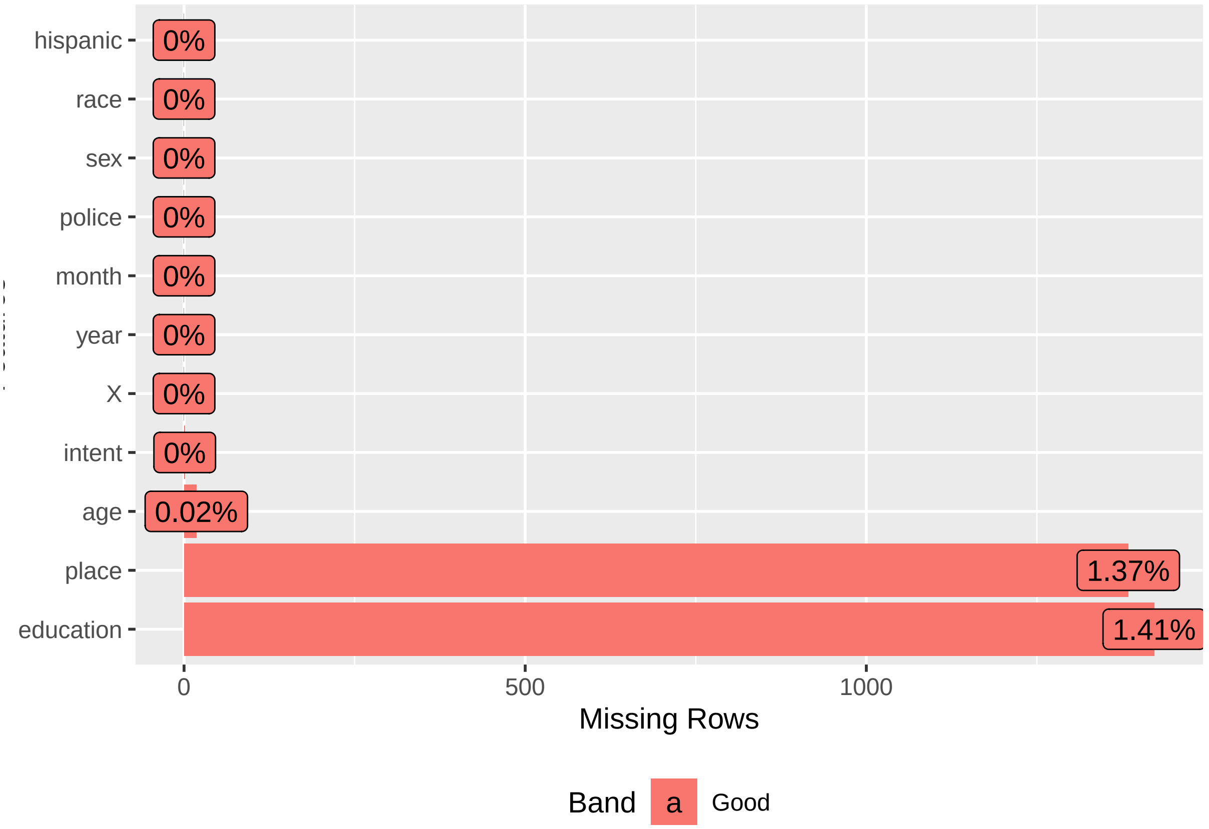
```
# read in the data, inspect and summaraize
dRaw <- read.csv("Data/full_data.csv", stringsAsFactors = FALSE)

# look at the first few rows
head(dRaw)
```

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

```
# typical summary
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
##
```

```
# visualize summary
plot_str(dRaw)

# 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.

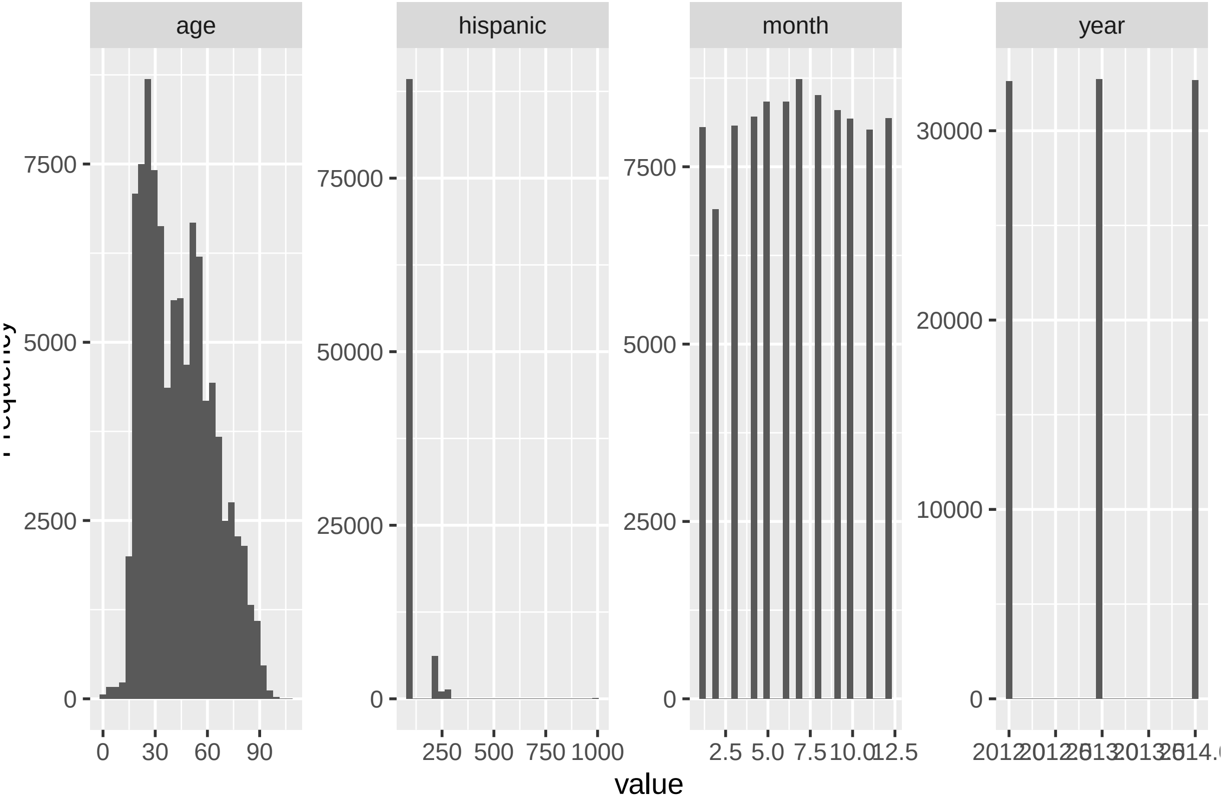
Overview of the Dataset

```
# 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)

head(d)
```

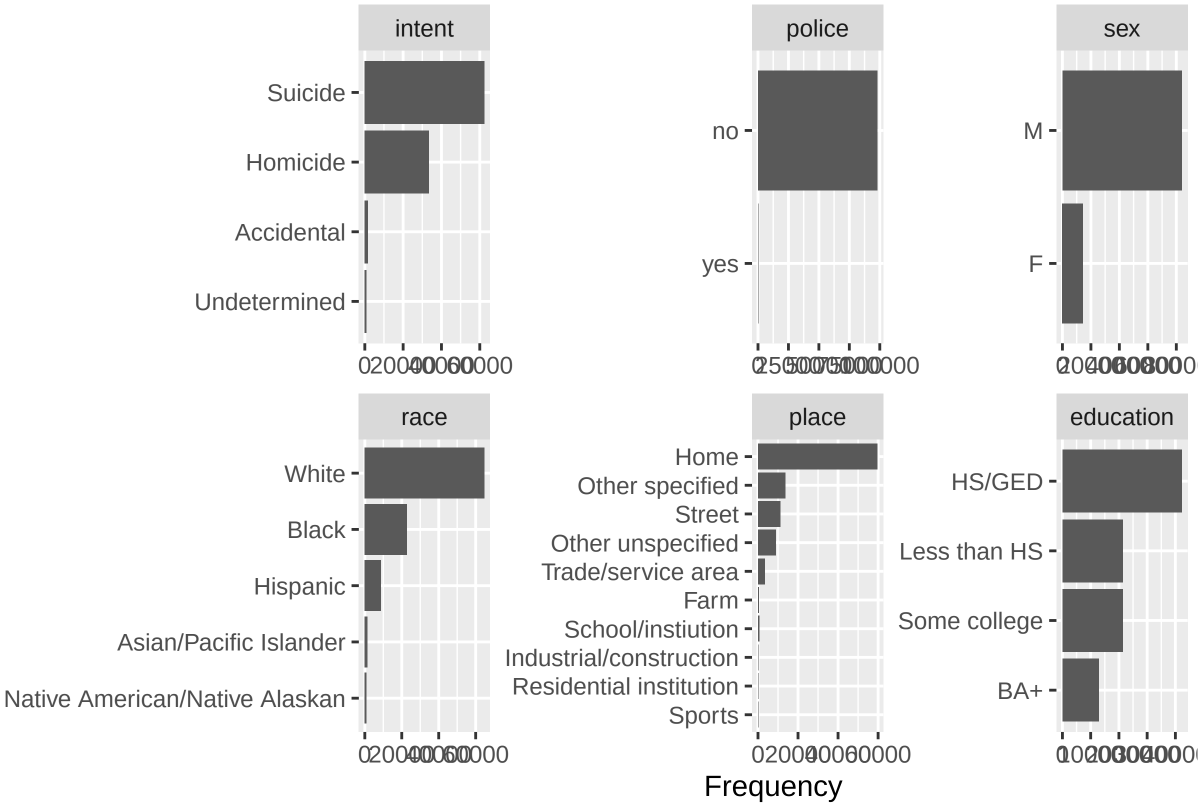
```
##   year month  intent police sex age                            race
## 1 2012     1 Suicide     no   M  34          Asian/Pacific Islander
## 2 2012     1 Suicide     no   F  21                           White
## 3 2012     1 Suicide     no   M  60                           White
## 4 2012     2 Suicide     no   M  64                           White
## 5 2012     2 Suicide     no   M  31                           White
## 6 2012     2 Suicide     no   M  17 Native American/Native Alaskan
##   hispanic           place    education
## 1      100            Home          BA+
## 2      100          Street Some college
## 3      100 Other specified          BA+
## 4      100            Home          BA+
## 5      100 Other specified       HS/GED
## 6      100            Home Less than HS
```

Let's get some high level summaries of the categories:

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



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

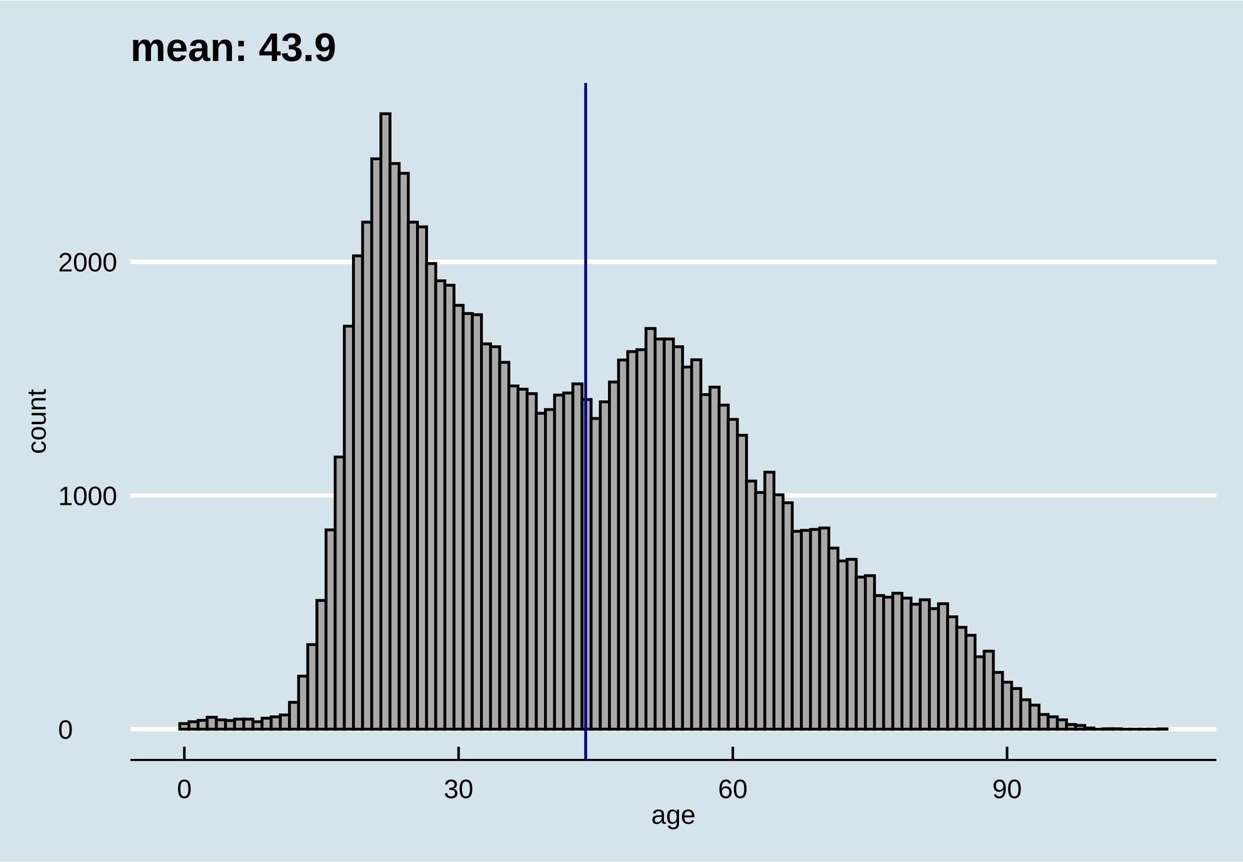


```r
# basic plot for our continous variables
plotContinuous <- function(df, colString, annotate = FALSE)
{
  p <- ggplot(df, aes(df[, colString])) +
        geom_histogram(fill = "Dark Grey", binwidth = 1, col = "Black") +
        theme_economist() + scale_color_economist() + xlab(colString)

    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)
{
  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))
}
```

Inspect Continuous Features

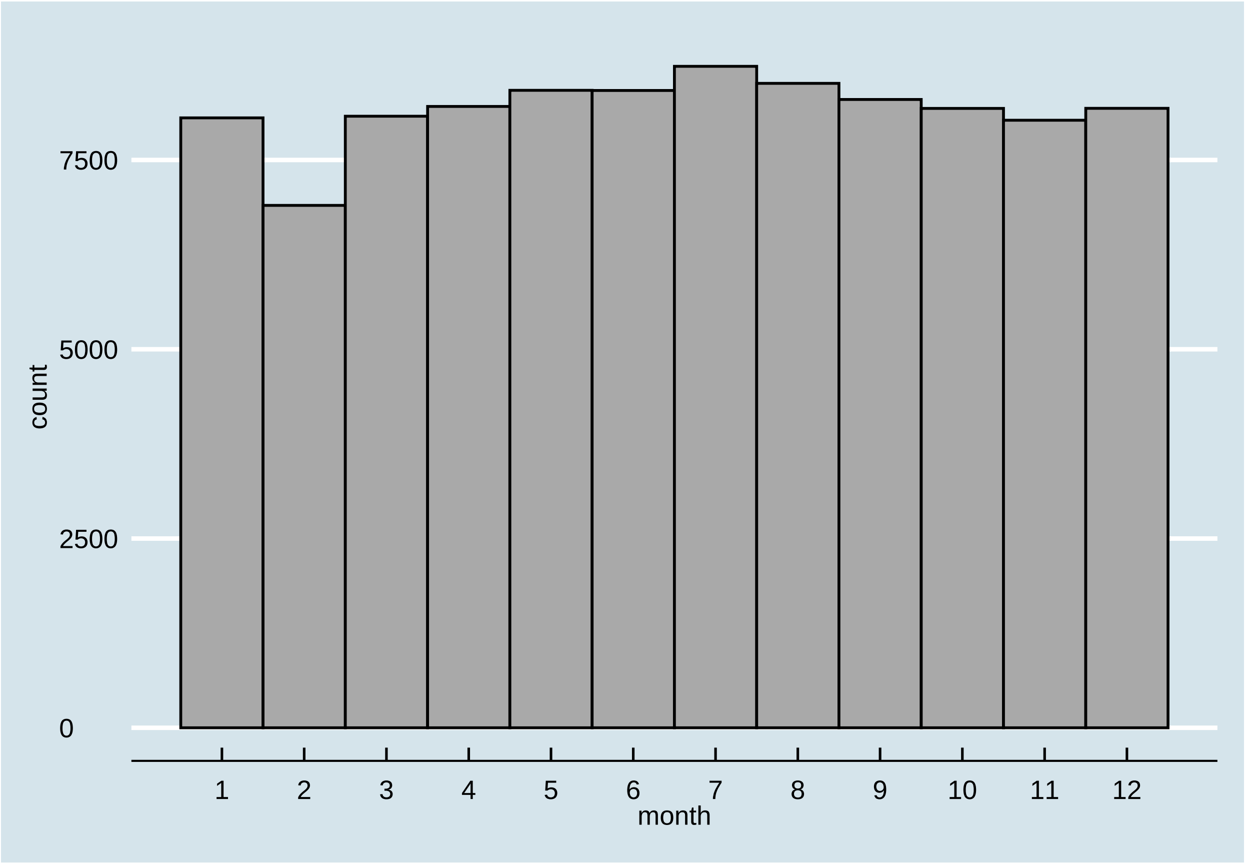
Age

```
plotContinuous(d, c("age"), annotate = TRUE)
```



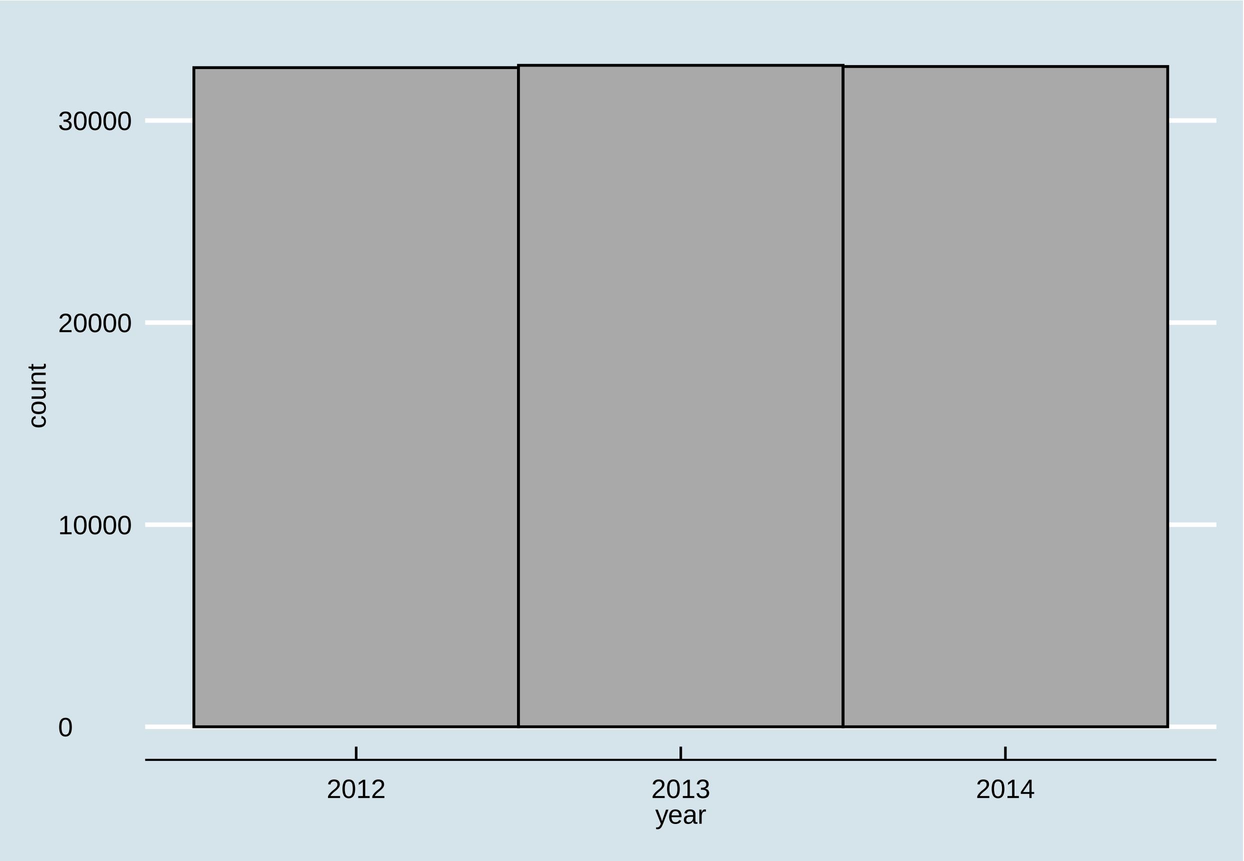
Month

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



Year

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



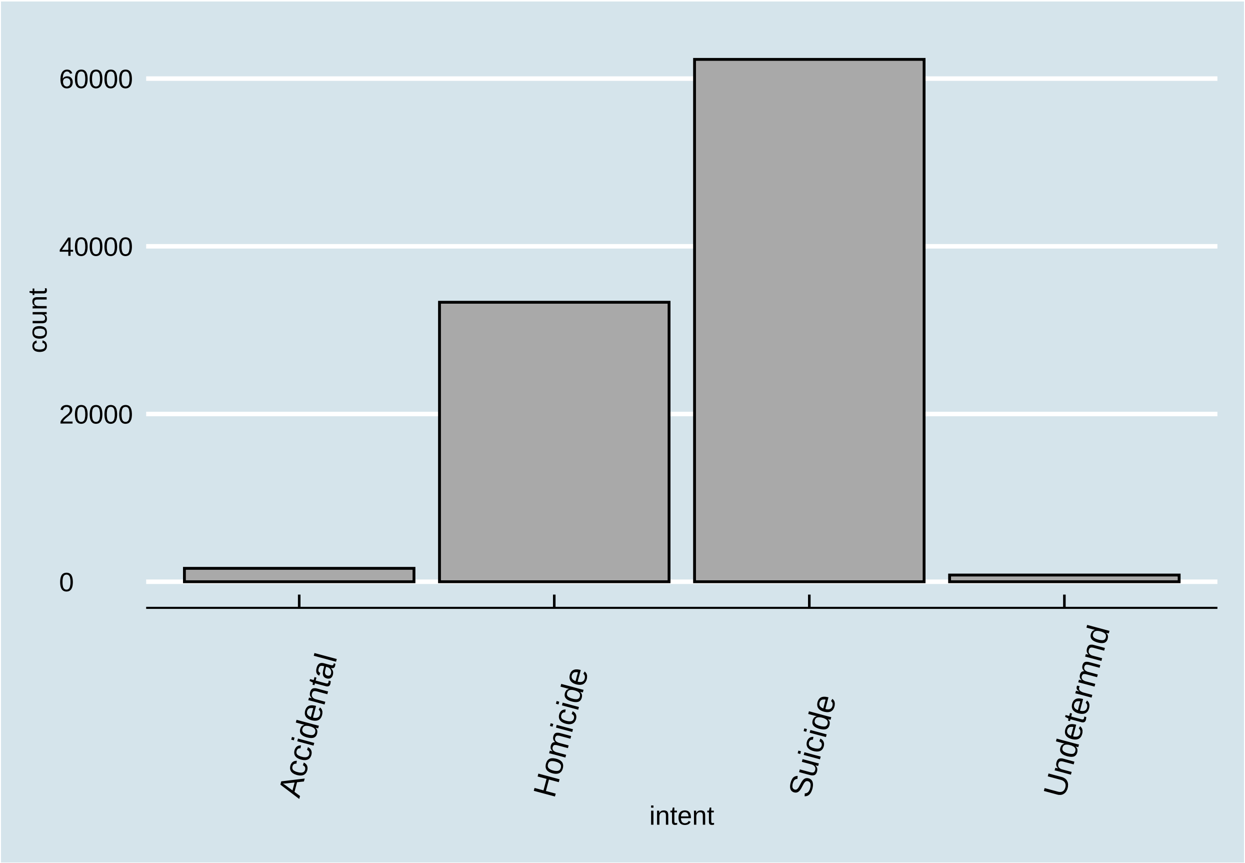
Contiuous Features Takeaway

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

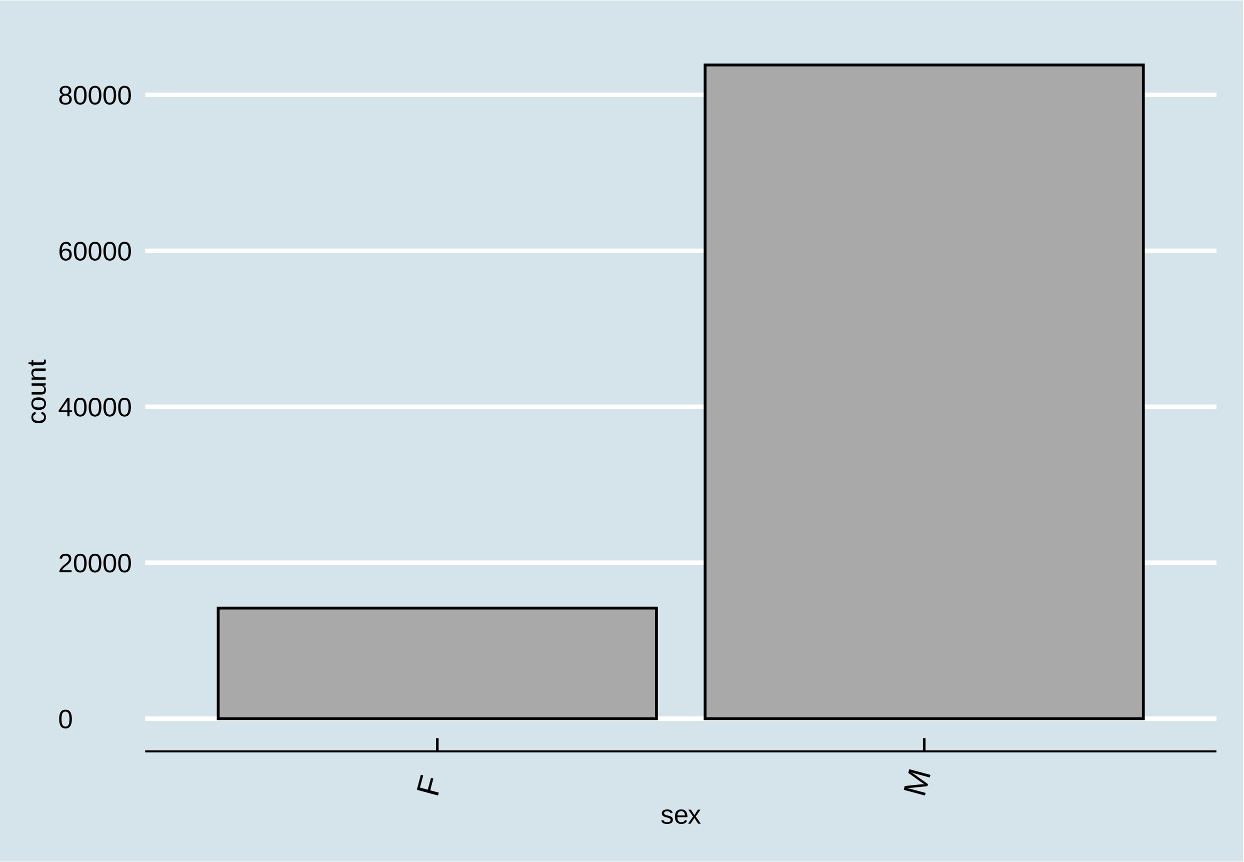
Intent

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



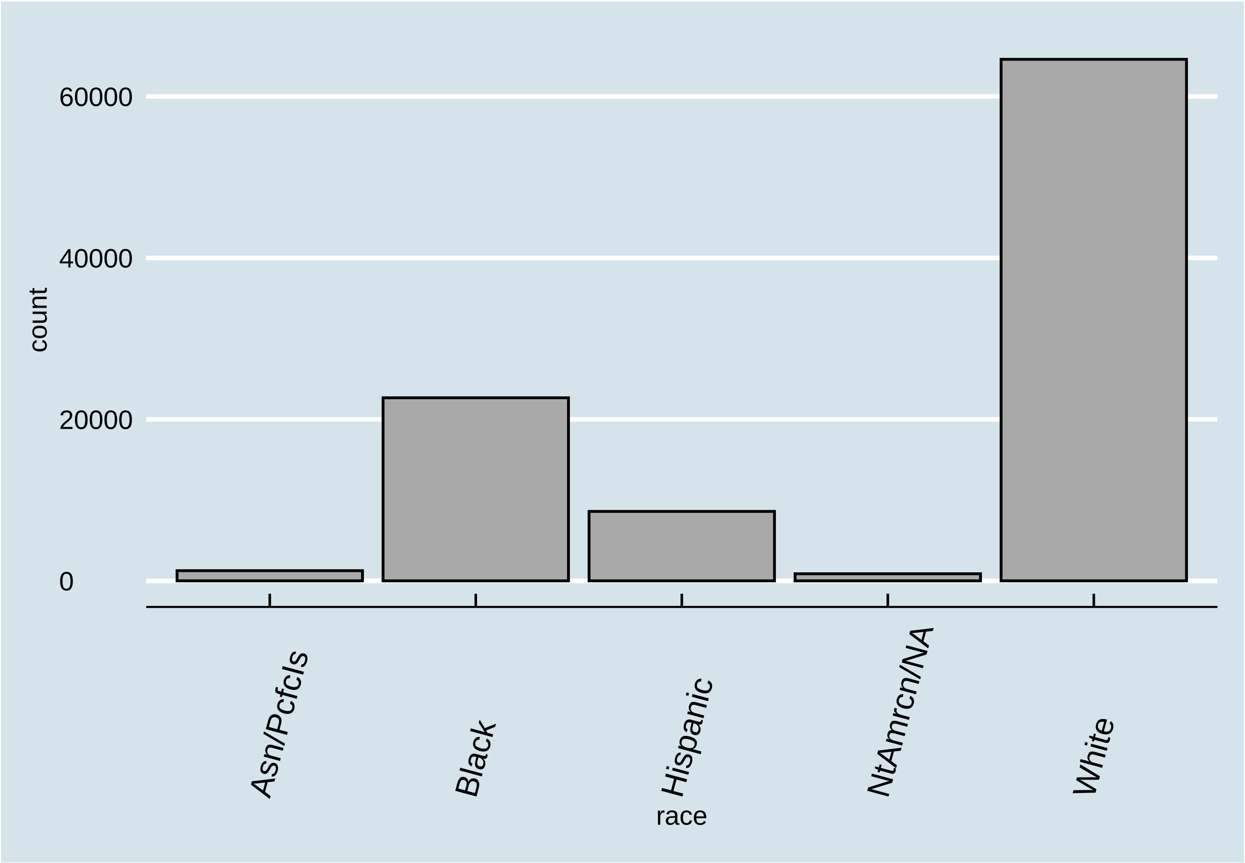
Sex

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



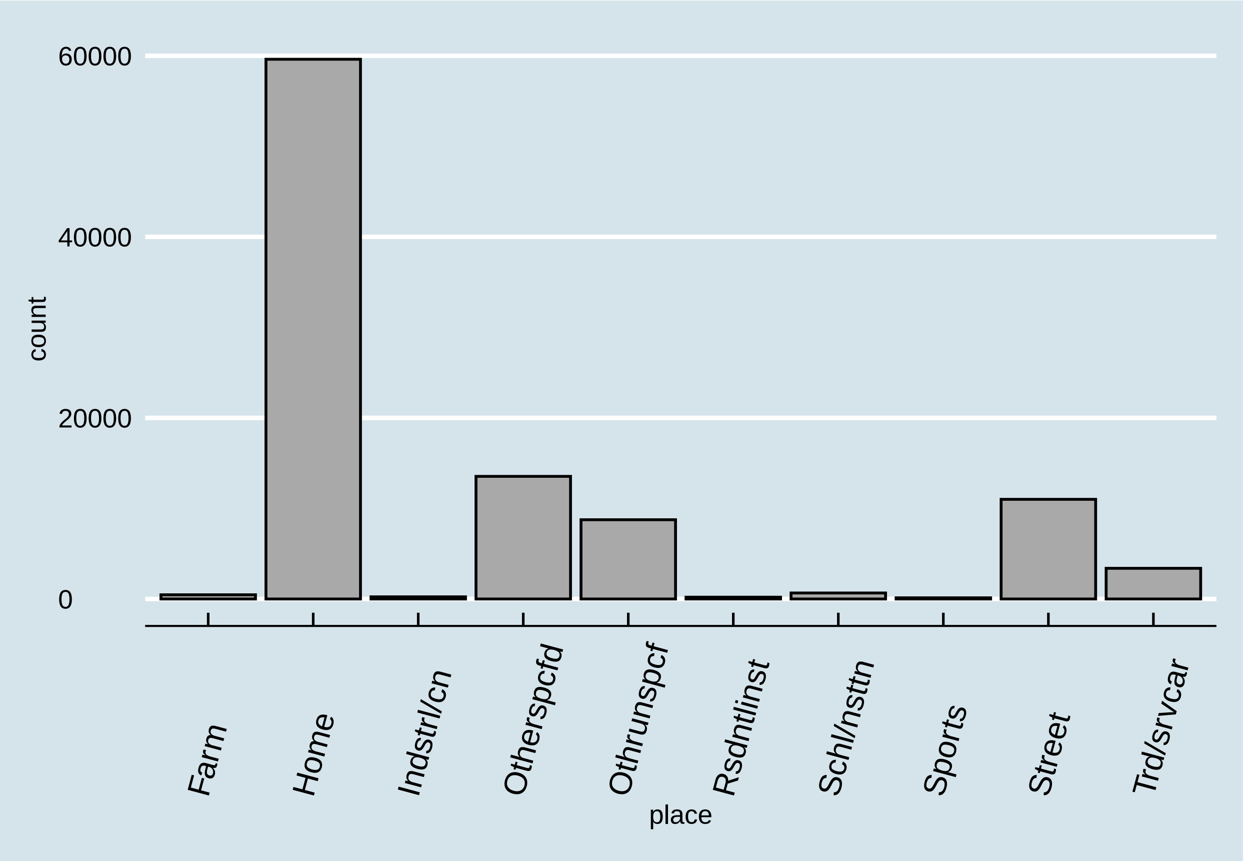
Race

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



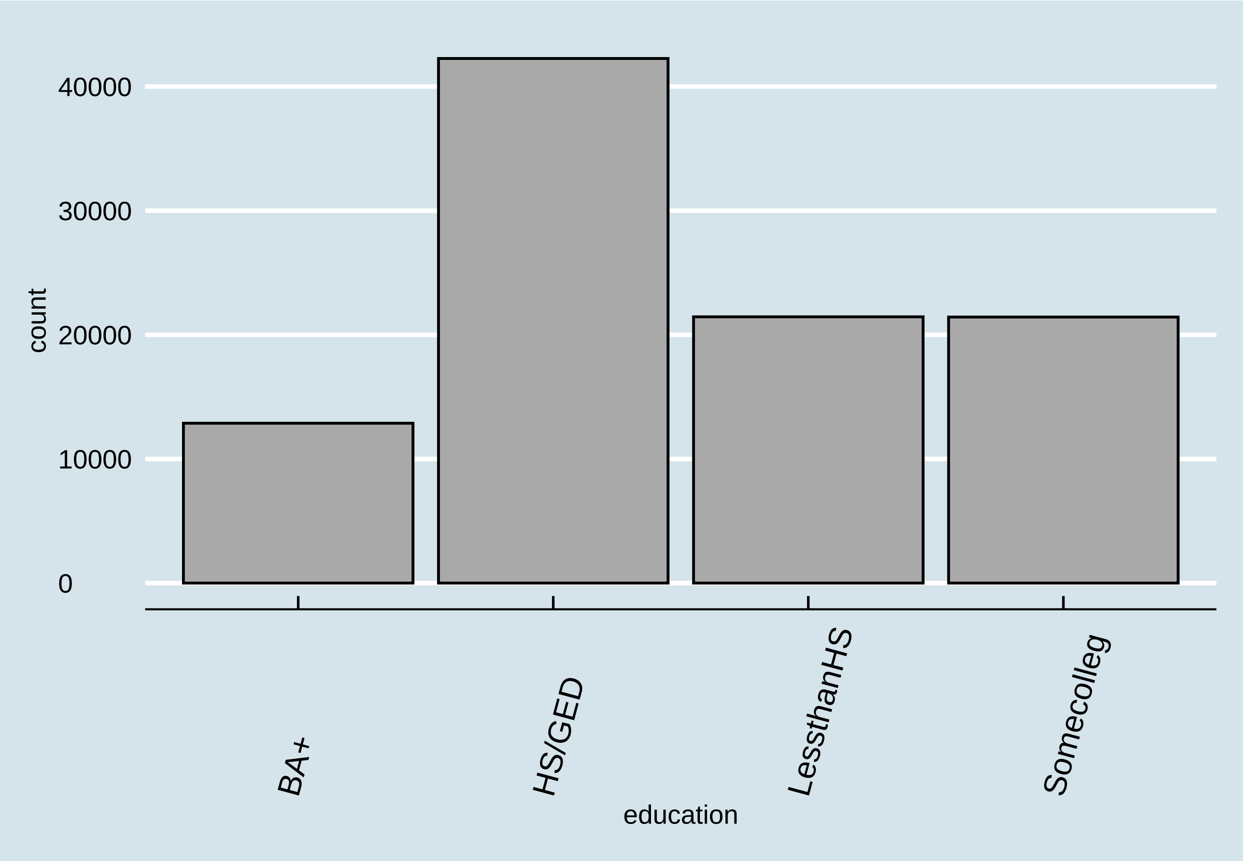
Place

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



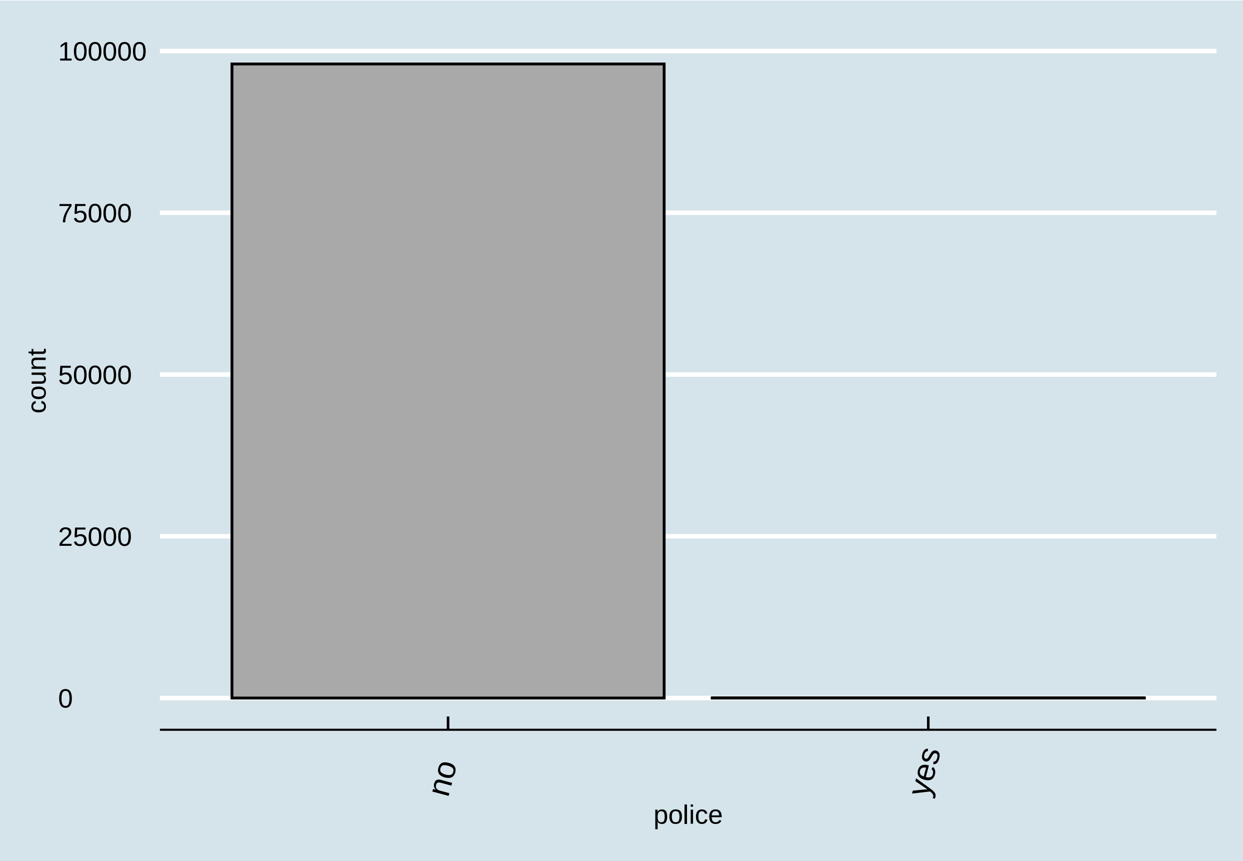
Education

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



Police

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



Continous Features Takeaway

The demographics of our dateset: 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

The following chart is the result of plotting the numbers of deaths per month in each year.

```r
# 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
  )
)

colnames(dDeathsByMonthByYear) <- c("month", "2012","2013","2014")

dDeathsByMonthByYear
```

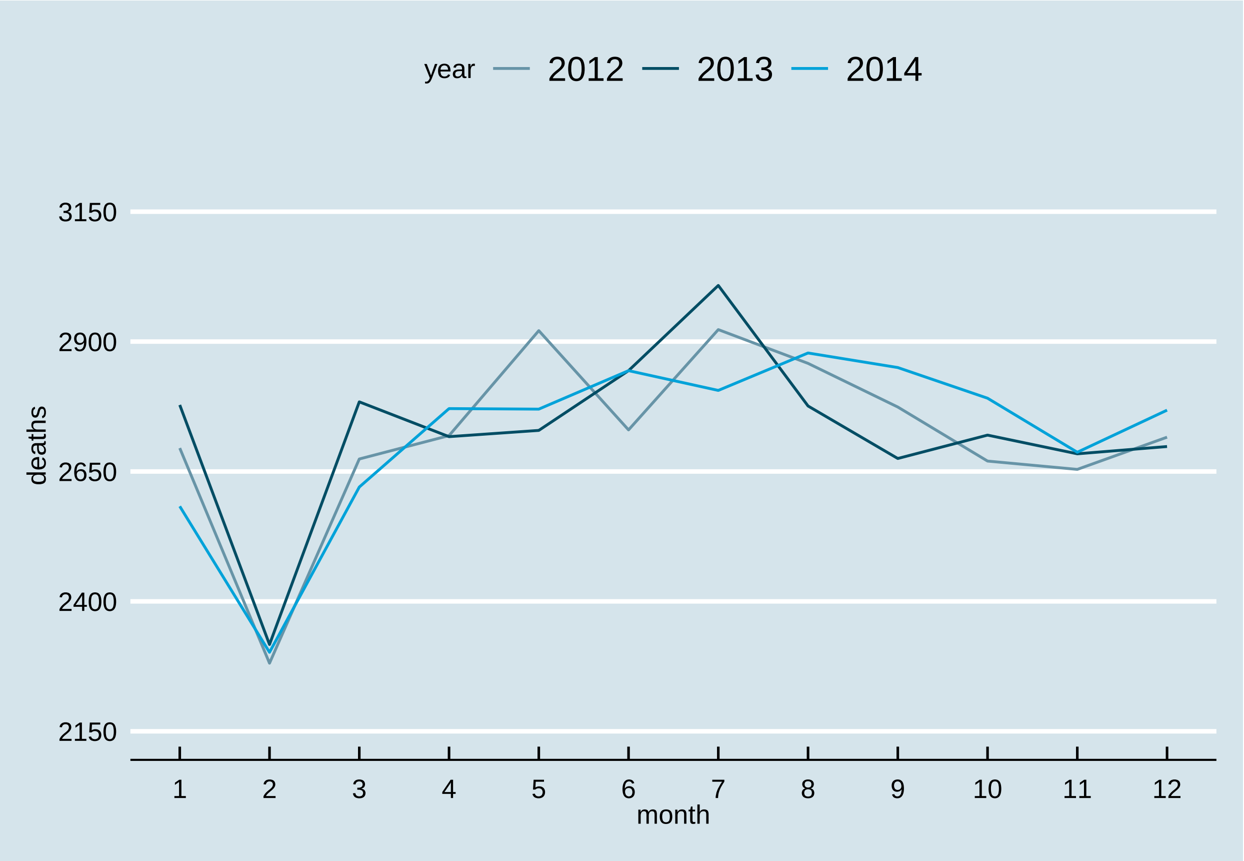
```
##    month 2012 2013 2014
## 1      1 2695 2778 2583
## 2      2 2281 2317 2302
## 3      3 2674 2784 2620
## 4      4 2719 2717 2771
## 5      5 2921 2729 2770
## 6      6 2730 2844 2844
## 7      7 2923 3008 2806
## 8      8 2858 2776 2878
## 9      9 2774 2675 2850
## 10    10 2670 2720 2791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

We now have an organized count by each month.

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

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

# inspect new frame
head(dMelt)
```

```
##   month year deaths
## 1     1 2012   2695
## 2     2 2012   2281
## 3     3 2012   2674
## 4     4 2012   2719
## 5     5 2012   2921
## 6     6 2012   2730
```

```
# plot the results on a line graph
ggplot(dMelt, aes(month,deaths, col =  year)) +
  geom_line() +
  scale_y_continuous(limits = c(2150, 3250), breaks = seq(1650, 3350, by = 250)) +
  scale_x_continuous(breaks = monthNumbers) +
  scale_color_economist() + theme_economist()
```



There is a noticable 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)

percentMissingFeb
```

```
## [1] 0.07036846
```

```
percentCompleteFeb
```

```
## [1] 0.07039739
```

February takes up almost excactly the same proportion of the missing vs utilized datset.

Number of Days in February

The next question that must be addressed is that February is the shortest month. We calculate the numbers of death per day to account for this.

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

# 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 od deaths in each month
numberOfDeaths <- sapply(monthNumbers, getDeaths, df = d)
numberOfDeaths
```

```
##  [1] 8056 6900 8078 8207 8420 8418 8737 8512 8299 8181 8025 8182
```

```
# number of deaths in that month across dataset
deathsPerMonth <- numberOfDeaths/sapply(monthNumbers, daysByMonth)
deathsPerMonth
```

```
##  [1] 86.62366 81.17647 86.86022 91.18889 90.53763 93.53333 93.94624
##  [8] 91.52688 92.21111 87.96774 89.16667 87.97849
```

```
# 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", "deathsPerMonth","zScoreOfDeathsPerMonth")

kable(dDeathsPerMonth) %>%
  kable_styling(position = "center", full_width = TRUE) %>%
  row_spec(0:nrow(dDeathsByMonthByYear), background = "#C0C0C0") %>%
  row_spec(0:nrow(dDeathsByMonthByYear)) %>%
  row_spec(0, bold = TRUE) %>%
  row_spec(2, bold = TRUE, color = "blue")
```

month	numberOfDeaths	deathsPerMonth	zScoreOfDeathsPerMonth
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