# NYPD Shootings: Multi-victim Incidents

Ron Sielinski

2024-05-16

## Step 1: Start an Rmd Document

The first step to analyzing New York City's shooting data set is to import the data in a reproducible manner.

I started by loading the tidyverse package, which contains multiple functions that will simplify many of the tasks involved in end-to-end analysis. For example, the `read_csv()` function allows us to import the data directly from the City of New York, which makes its data freely available at https://data.cityofnewyork.us.

```
# load the tidyverse package
library(tidyverse)

# load data from the City of New York
data_file <- 'https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD'
ny_dat <- read_csv(data_file)
```

## Step 2: Tidy and Transform the Data

Next, I profiled the data to understand what the data set contains and to identify question(s) that I might want to answer.

```
# display a high-level summary of the data
summary(ny_dat)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME           BORO
##  Min.   :  9953245   Length:28562       Length:28562      Length:28562
##  1st Qu.: 65439914   Class :character   Class1:hms        Class :character
##  Median : 92711254   Mode  :character   Class2:difftime   Mode  :character
##  Mean   :127405824                      Mode  :numeric
##  3rd Qu.:203131993
##  Max.   :279758069
##
##  LOC_OF_OCCUR_DESC     PRECINCT     JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:28562       Min.   :  1.0   Min.   :0.0000    Length:28562
##  Class :character   1st Qu.: 44.0   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 67.0   Median :0.0000    Mode  :character
##                     Mean   : 65.5   Mean   :0.3219
##                     3rd Qu.: 81.0   3rd Qu.:0.0000
##                     Max.   :123.0   Max.   :2.0000
##                                     NA's   :2
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
```

```
##  Length:28562       Mode :logical          Length:28562      
##  Class :character   FALSE:23036            Class :character  
##  Mode  :character   TRUE :5526             Mode  :character  
##                                                              
##                                                              
##                                                              
##                                                              
##    PERP_SEX          PERP_RACE         VIC_AGE_GROUP        VIC_SEX         
##  Length:28562       Length:28562       Length:28562       Length:28562      
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##    VIC_RACE           X_COORD_CD        Y_COORD_CD        Latitude    
##  Length:28562       Min.   : 914928   Min.   :125757   Min.   :40.51  
##  Class :character   1st Qu.:1000068   1st Qu.:182912   1st Qu.:40.67  
##  Mode  :character   Median :1007772   Median :194901   Median :40.70  
##                     Mean   :1009424   Mean   :208380   Mean   :40.74  
##                     3rd Qu.:1016807   3rd Qu.:239814   3rd Qu.:40.82  
##                     Max.   :1066815   Max.   :271128   Max.   :40.91  
##                                                        NA's   :59     
##    Longitude        Lon_Lat         
##  Min.   :-74.25   Length:28562      
##  1st Qu.:-73.94   Class :character  
##  Median :-73.92   Mode  :character  
##  Mean   :-73.91                     
##  3rd Qu.:-73.88                     
##  Max.   :-73.70                     
##  NA's   :59
```

To do that, I needed know what the individual columns in the data set tell us about the shootings. Fortunately, the city also publishes a guide to the data: https://data.cityofnewyork.us/Public-Safety/NYPD-Shooting-Incident-Data-Historic-/833y-fsy8/about_data. From that, I learned that the first column, INCIDENT_KEY, is a persistent identifier for each shooting incident.

A quick query confirms that an individual incident can have multiple victims.

```r
# for my convenience, convert column names to lower case (to make typing easier)
colnames(ny_dat) <- colnames(ny_dat) %>%
  str_to_lower()

# display the number of shooting records per unique incident
ny_dat %>%
  group_by(incident_key) %>%
  summarize(cnt = n()) %>%
  ungroup() %>%
  arrange(desc(cnt))
```

```
## # A tibble: 22,394 x 2
##    incident_key   cnt
##           <dbl> <int>
##  1    173354054    18
```

```
## 2     263503175    16
## 3      23749375    12
## 4      24717013    12
## 5      33478089    12
## 6      33706902    12
## 7      35803777    12
## 8      66027258    12
## 9      72195829    12
## 10     72616285    12
## # i 22,384 more rows
```

The results show that one incident had 18 victims!

However, there is only one 18-victim incident out of more than 22k, which suggested that multi-victim incidents would be an interesting topic to explore.

To simplify the analysis, I reduced the data set to just the columns that I was likely to need: incident_key, occur_date, and statistical_murder_flag. To facilitate a variety of perspectives, I changed the occur_date from a character to a date and introduced occur_year and month_name as factors.

I intentionally chose *not* to convert statistical_murder_flag to a factor because the field was already a Boolean, and R has built-in functions that make dealing with Boolean values straightforward.

```r
# keep only fields needed to analyze multi-victim incidents
ny_dat <- ny_dat %>%
  select(incident_key, occur_date, statistical_murder_flag)

# convert the date field from a character data type to a date
ny_dat$occur_date <- mdy(ny_dat$occur_date)

# coarsen the granularity of the occurrence date to monthly and yearly and convert
# them to factors, which will help to simplify analysis and visualizations
ny_dat <- ny_dat %>%
  mutate(occur_year = as.factor(year(occur_date)),
         occur_month = floor_date(occur_date, unit = 'month'),
         calendar_month = month(occur_date),
         month_name = factor(month.name[calendar_month], levels = month.name)
  )
```

Finally, I created a data frame of multi-vicitim incidents and reduced the source data (ny_dat) to just those rows that contained multi-victim incidents.

```r
# create a data frame that contains multi-victim incidents
# for each incident, include the number of victims and murders per incident
incident_cnts <- ny_dat %>%
  group_by(incident_key) %>%
  summarize(victims = n(), murders = sum(statistical_murder_flag)) %>%
  ungroup() %>%
  filter(victims > 1)

# reduce ny_dat to just those rows with a matching incident_key in incident_cnts
ny_dat <- ny_dat |>
  inner_join(select(incident_cnts, incident_key), by = 'incident_key')
```

Summarizing the data again confirms that the reduced data set is complete: None of columns in the reduced data set has missing data.

```r
# display a high-level summary of the data
summary(ny_dat)
```

```
##   incident_key         occur_date         statistical_murder_flag
##  Min.   :  9953250   Min.   :2006-01-01   Mode :logical
##  1st Qu.: 62860035   1st Qu.:2009-06-17   FALSE:7533
##  Median : 90905822   Median :2013-05-30   TRUE :2322
##  Mean   :126161269   Mean   :2014-04-26
##  3rd Qu.:206568218   3rd Qu.:2019-12-16
##  Max.   :279547333   Max.   :2023-12-26
##
##    occur_year    occur_month          calendar_month       month_name
##  2006   : 777   Min.   :2006-01-01   Min.   : 1.000   July     :1241
##  2008   : 713   1st Qu.:2009-06-01   1st Qu.: 5.000   August   :1163
##  2007   : 712   Median :2013-05-01   Median : 7.000   June     :1036
##  2010   : 701   Mean   :2014-04-11   Mean   : 6.766   May      : 991
##  2021   : 684   3rd Qu.:2019-12-01   3rd Qu.: 9.000   September: 934
##  2011   : 677   Max.   :2023-12-01   Max.   :12.000   October  : 808
##  (Other):5591                                         (Other)  :3682
```

Had there been missing data, there are a number of techniques that I could have used to address the missing data:

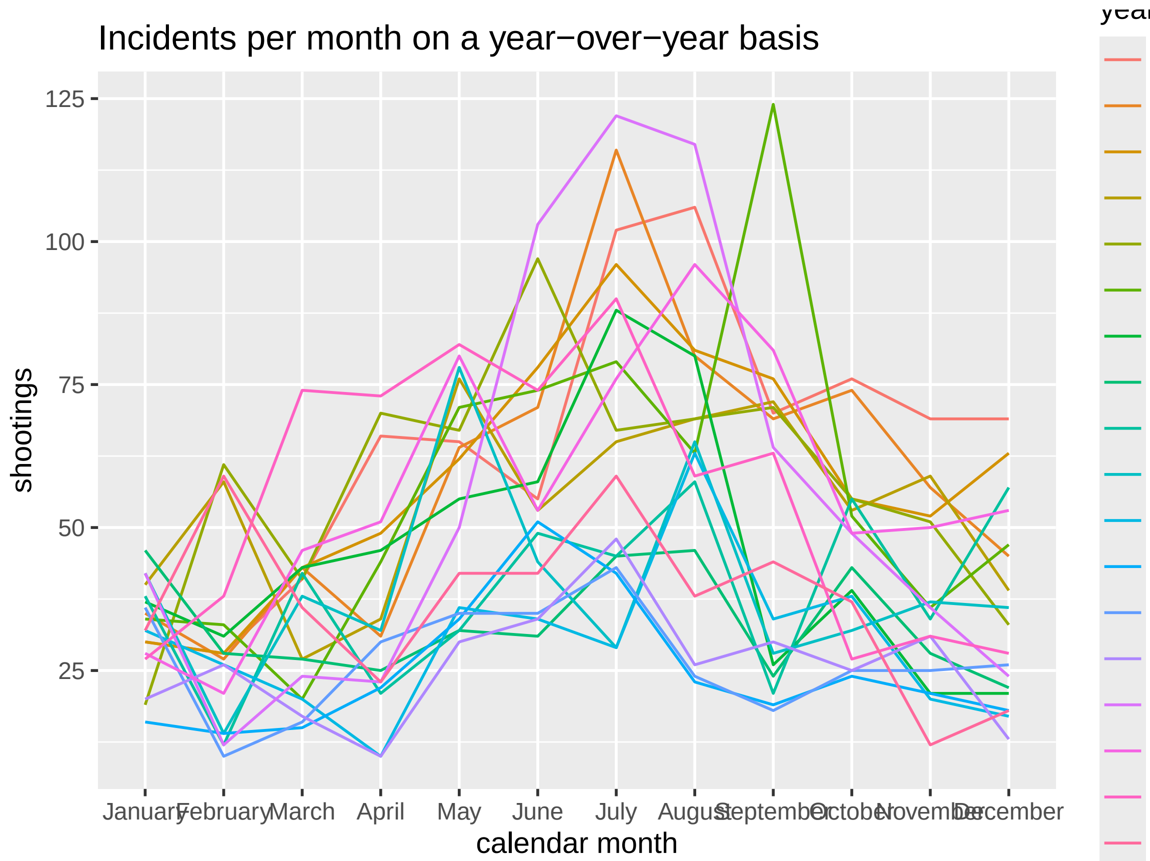
- Cross validation: Since individual incidents can have multiple victims, I could have checked the other records from the same incident to see if they had the missing data.
- Ignore the missing data: jurisdiction_code, for example, has only 2 records with NA values. Ignoring the NA's would have minimal impact on most statistical analyses, although that fact would need to be confirmed and called out in any summary/report of the analysis.
- Reclass the missing data: In some cases (e.g, perp_sex), it might be sufficient to reclass the NA's as "U" (unknown), because, depending on the analysis, NA and "U" might be functionally identical.

Additional other techniques exist, but that topic is beyond the scope of this analysis.

## Step 3: Add Visualizations and Analysis

To get a better understanding at incident data, I took look at a number of visualizations. The first was a chart that showed the number of multi-victim incidents over time.

```r
# count the number of shootings by month and year
ny_dat %>%
  group_by(month_name, occur_year) %>%
  summarize(shootings = n()) %>%
  ungroup() %>%
  # plot the result
  ggplot(aes(x = month_name)) +
  geom_line(aes(y = shootings, group = occur_year, color = occur_year)) +
  labs(title = 'Incidents per month on a year-over-year basis', x = 'calendar month', color = 'year')
```

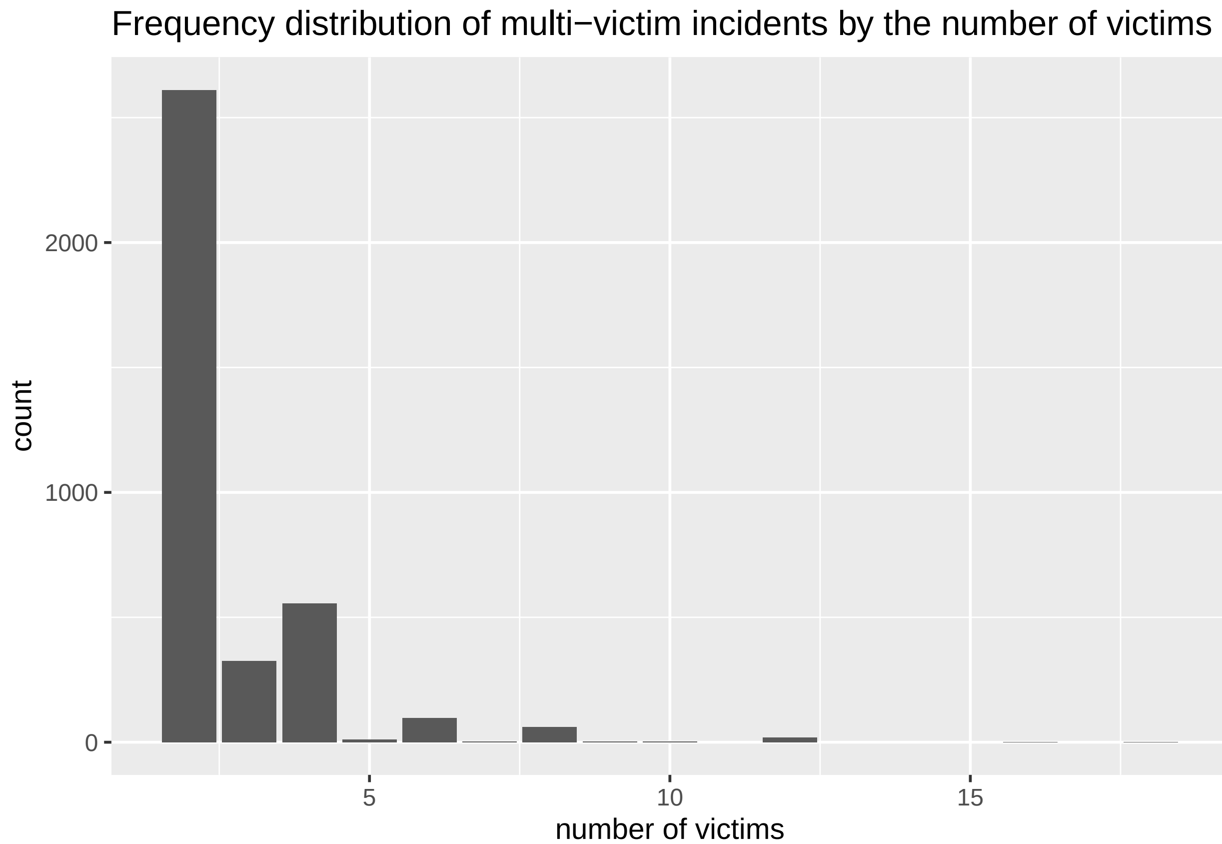



The yearly trends show a recurring pattern of an increased number of incidents in the summer months. Conversely, February often has the lowest number of incidents, but that's a byproduct of the fact that February has the fewest number of days (i.e., the fewest number of opportunities for an incident to occur).

To avoid potential issues with date-related and seasonal bias, I chose to focus on rate-based metrics independent of time.

To that end, I took a look at a histogram that showed a count of incidents by the number of victims.

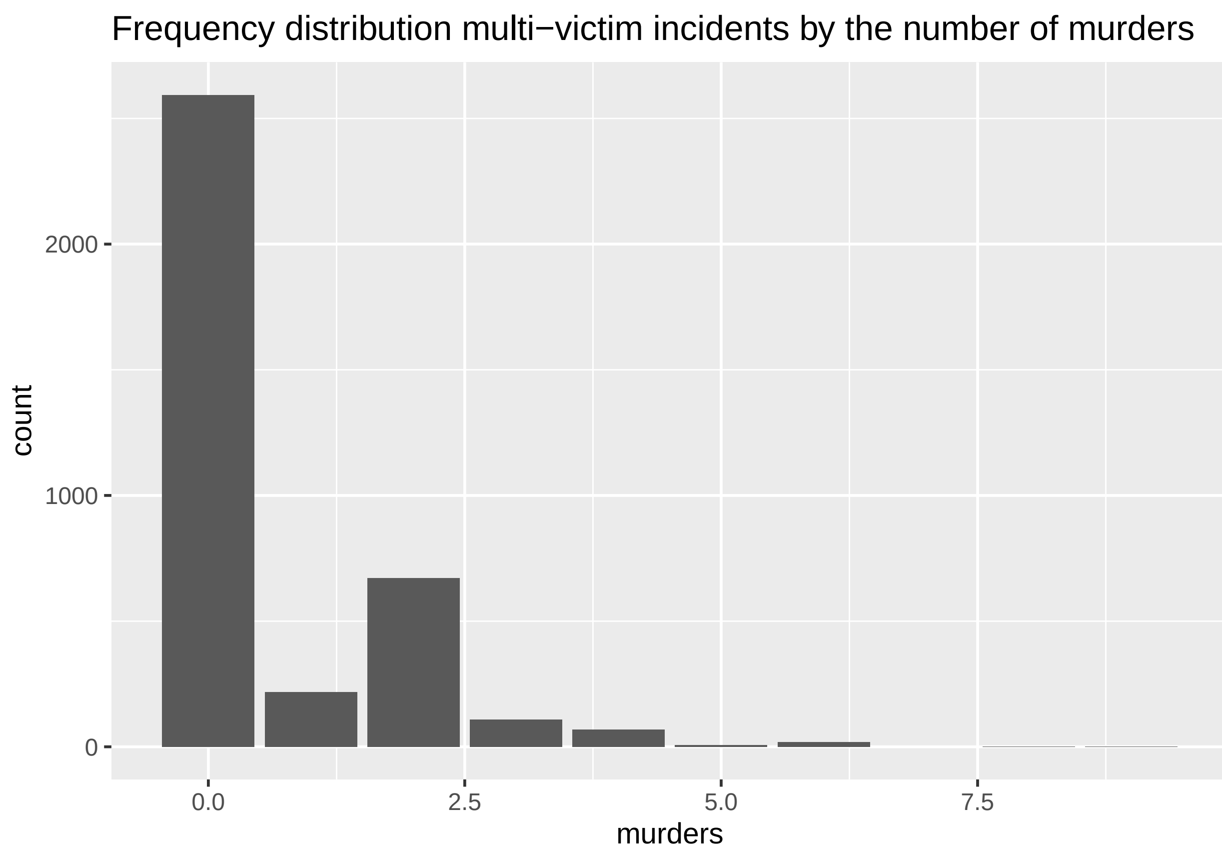
```r
# look at the frequency distribution of multi-victim incidents
incident_cnts %>%
  ggplot() +
  geom_histogram(aes(x = victims), stat = 'count') +
  labs(title = 'Frequency distribution of multi-victim incidents by the number of victims',
       x = 'number of victims')
```




There are over 2,500 incidents with two (2) victims, far more than any other number of victims.

I also looked at a histogram that showed a count of multi-victim incidents by the number of murders.

```
# look at the frequency distribution of murders for multi-victim incidents
incident_cnts %>%
  ggplot() +
  geom_histogram(aes(x = murders), stat = 'count') +
  labs(title = 'Frequency distribution multi-victim incidents by the number of murders')
```




Similarly, there are over 2,500 incidents with zero murders. It's not clear from these separate perspectives, however, how many of the 2-victim incidents had zero murders: Any incident could have zero murders, regardless of the number of victims. (If the number of victims in an incident is $n$, an $n$-victim incident could have from 0 to $n$ murders. A 2-victim incident could have zero murders, and a 12-victim incident could have zero murders.) This led to a new question: What is the typical murder rate of multi-victim incidents? A simple average is a straightforward way to estimate the murders-to-victims ratio.

```
# average number of murders per shooting
global_rate <- sum(incident_cnts$murders) / sum(incident_cnts$victims)
print(global_rate)
```

```
## [1] 0.2356164
```

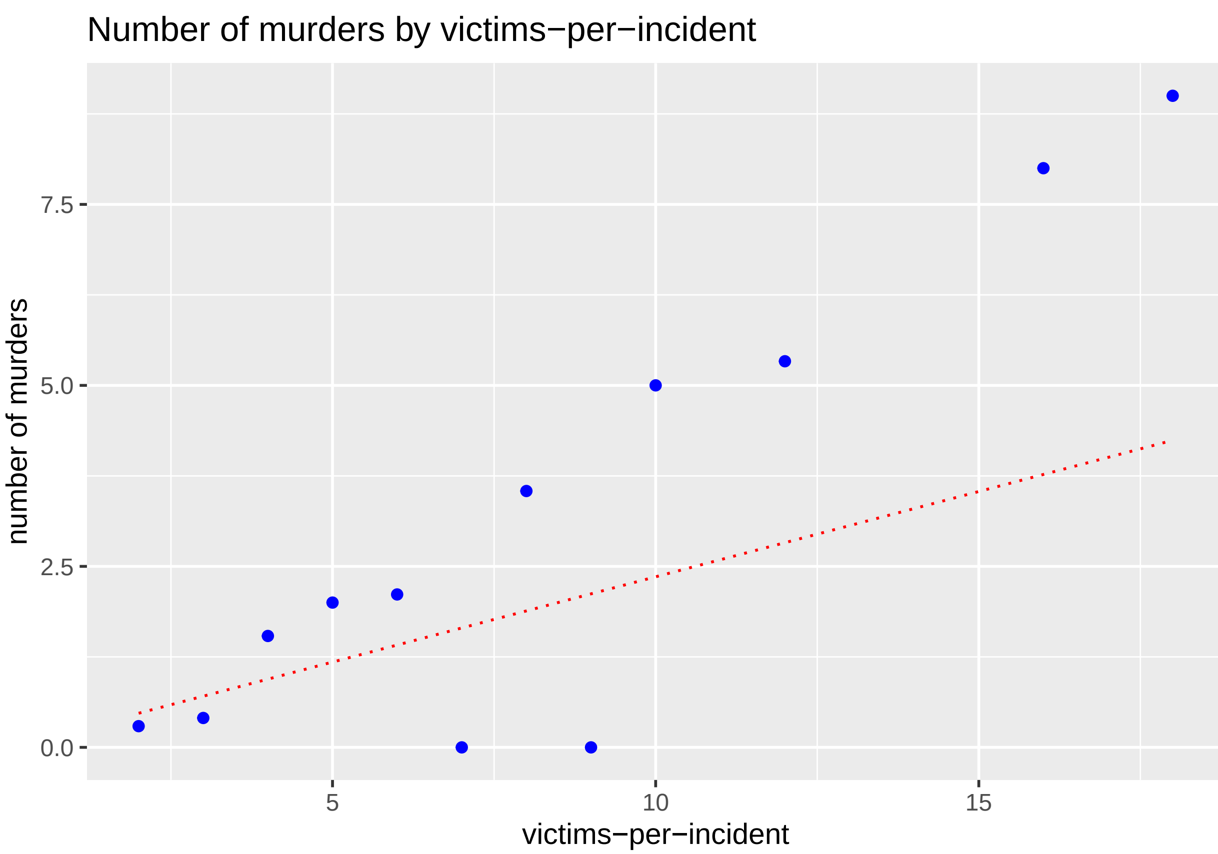
The global average is 0.24 murders per victim.

Averages can be deceiving, however. In fact, a model that estimates the number of murders based on that average would perform poorly. The following chart shows that, in most cases, an average-based model either over-estimates or under-estimates the murders-to-victims ratios.

```
# calculate the actual ratio of murders-to-victims based on the number of
# victims-per-incident
murder_ratio <- incident_cnts %>%
  group_by(victims) %>%
  summarize(incidents = n(), mean_murders = mean(murders)) %>%
  ungroup() %>%
  mutate(murder_ratio = mean_murders / victims)
```

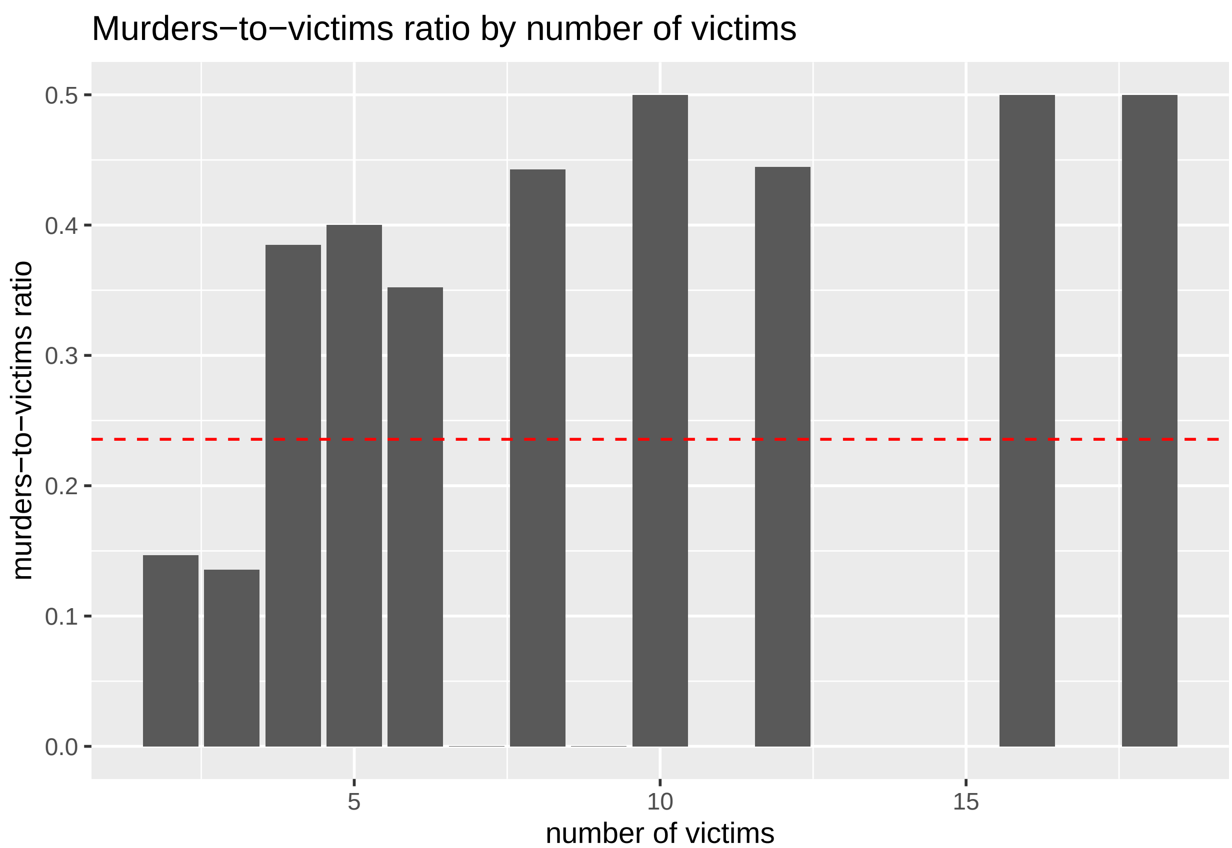
```r
# use the global_rate to predict the number of murders
murder_ratio$global_rate <- global_rate * murder_ratio$victims

# plot the difference between the actual and predicted number of murders
ggplot(murder_ratio, aes(x = victims)) +
  geom_point(aes(y = mean_murders), color = 'blue') +
  # add the average model
  geom_line(aes(y = global_rate), color = 'red', linetype = 'dotted') +
  labs(title = 'Number of murders by victims-per-incident',
       y = 'number of murders',
       x = 'victims-per-incident')
```




The reason is that the murders-to-victims ratio changes with the number of victims. The following chart illustrates that point:

```r
# plot the murders-to-victims ratio
ggplot(murder_ratio) +
  geom_col(aes(x = victims, y = murder_ratio)) +
  # show the average murder ratio as a dashed red line
  geom_hline(yintercept = global_rate, color = 'red', linetype = 'dashed') +
  labs(title = 'Murders-to-victims ratio by number of victims',
       x = 'number of victims',
       y = 'murders-to-victims ratio')
```




Incidents that involve 2 or 3 victims have below-average murders-to-victims ratios, and incidents that involve 4 or more victims have above-average murders-to-victims ratios.

Because most multi-victims incidents involve only 2 victims, they have the most influence on the calculation of the average.

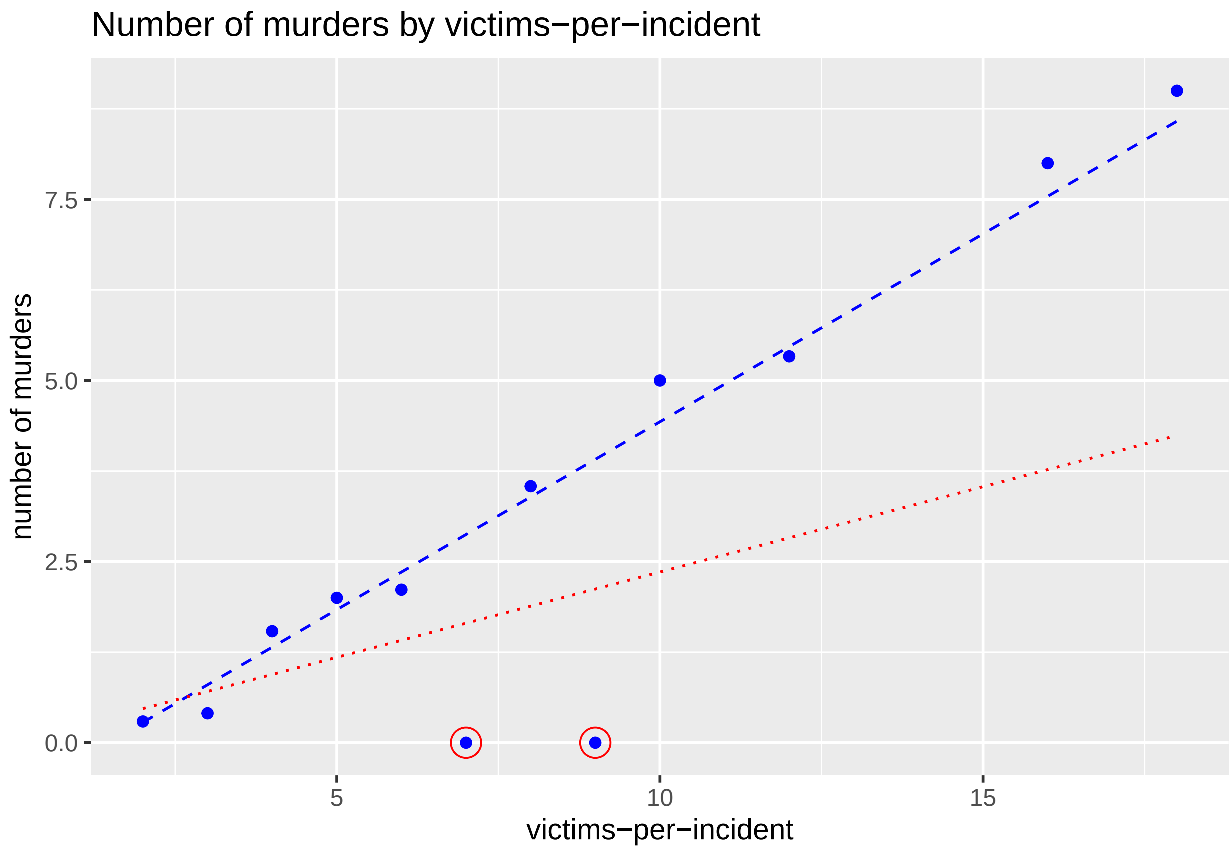
As such, a model that incorporates the number of victims per incident would produce more accurate results.

```r
# build a model to predict the number of murders based on victims-per-incident
lm_fit <- lm(murders ~ victims, data = select(incident_cnts, murders, victims))

# use the model to predict the number of murders
murder_ratio$predicted <- predict(lm_fit, newdata = murder_ratio)

# plot the difference between the actual and predicted number of murders
ggplot(murder_ratio, aes(x = victims)) +
  geom_point(aes(y = mean_murders), color = 'blue') +
  # add the linear model
  geom_line(aes(y = predicted), color = 'blue', linetype = 'dashed') +
  # add the average model
  geom_line(aes(y = global_rate), color = 'red', linetype = 'dotted') +
  # emphasize the outliers
  geom_point(
    data = filter(murder_ratio, mean_murders == 0),
    aes(y = mean_murders),
    color = 'red',
    shape = 1,
```

```
    size = 5
  ) +
  labs(title = 'Number of murders by victims-per-incident',
       y = 'number of murders',
       x = 'victims-per-incident')
```




The chart above shows that a simple linear model is much more accurate.

### Conclusion

An initial exploration of NYPD Shootings involved summarizing and transforming the data to better understand its structure and contents. This led to a focus on multi-victim incidents and their murder rates.

By grouping the data by incident_key, it was possible to visualize and better understand the distribution of multi-victim incidents. A global average for the murder-to-victim ratio was determined to be 0.24. Further analysis, however, led to the discovery that 2- and 3-victim incidents have a lower murder rates than incidents that involve more victims.

The importance of this insight was illustrated by showing the difference in accuracy between models that predict a number of murders using 1) a global average and 2) averages based on the number of victims per incident.

#### Potential Sources of Bias

Both models were least accurate for 7- and 9-victim incidents (the points circled in red in the final chart). The reason for that is the data set did not include *any* 7- and 9-victim incidents that also involved murders.

The small number of multi-victim incidents, especially incidents with a higher number of victims, introduces small-sample bias. I have chosen to handle that bias simply by calling attention to it.

There are multiple other sources of bias:

- This analysis looks at the number of murders per incident. However, the definition of "murder" is not clear. For example, does murder depend on a plea or verdict of guilty? Would the historic count of murders change if a perpetrator were acquitted? Similarly, is (unintentional) manslaughter handled differently from (intentional) murder—or not counted at all? A clear definition of murder is important for both a correct framing of the analysis and an interpretation of the results.
- The online guide to the data specifically refers to the INCIDENT_KEY as a "randomly generated persistent ID for each arrest". That suggests that an incident might be counted only if a perpetrator is arrested. That might make the analysis blind to incidents where an arrest did *not* occur.
- I avoided potential date-related and seasonal biases by looking at the data independent of calendar dates. Had a per-month perspective been required, I would have needed to adjust for the number of days-per-month and the seasonal differences in incident rates.
- My personal bias had a definite effect on this analysis. I have long been leery of summary statistics (like the mean/average), particularly when they can lead to mistaken conclusions about the data. In this case, that bias has resulted in my directing the analysis to a result that confirms my bias. Had I chosen to do so, I could have mitigated that bias by focusing on a different aspect of the shooting incident data.