**A grim research question**

The original research question was “what is the distribution of number of fatalities per incident in USA firearm-related violence?”. The motivation was to better understand how material mass shooting or multiple-shooting events are in a broader context, and what proportion of violence is in smaller events than those that get the headlines.

Of course, there’s a lot written on this topic already. Some of the important points made there:

* Mass shootings are rare despite the publicity they get; and the people doing them are different from most firearm violence perpetrators
* Two-thirds of firearm deaths are suicides (which may or may not be part of an incident involving other injuries and fatalities)
* Most firearm homicide victims are men between the ages of 15 and 34, and two-thirds are black; again, the profile of most firearm homicide victims differs from that of mass shootings (where 50% of victims are women, and around have of mass shootings involve domestic or family violence)

Putting aside the natural interest, importance and indeed horror that firearm violence in the USA attracts around the world, my question was more about the number of deaths as interesting extreme values. Most firearm incidents have few if any deaths, but some have very many; how can this be statistically modelled?

Some final bits of context. We saw about violent deaths as a percentage of population in an earlier post. The USA has more homicides per capita than other rich countries, although less than the front runners Colombia, Brazil and Russia. Here’s a key graphic from that post:

Code Chunks - violent deaths as a percentage of population

Arranging the data

Here’s how I import the data into R. If you know exactly what you’re looking for, you can import data directly from OECD.Stat with the [rsdmx](https://cran.r-project.org/web/packages/rsdmx/index.html) package by sending an appropriately structured URL (line 18 below) to their website. However, to work out that URL you’ll probably need to navigate  by hand and choose your particular ‘customising’, then choose the “export” and “SDMX” options, then copy the URL it generates for you. It’s useful for reproducibility purposes (for example, it meant I didn’t need to save a copy of the data anywhere to make the code below reproducible for others).

library(ggplot2)

library(scales)

library(grid)

library(gridExtra) *# added 27/11/2015, for grid.arrange() to work!*

library(dplyr)

library(tidyr)

library(rsdmx) *# for importing OECD data*

library(ISOcodes) *# for merging country codes with names*

library(rvest) *# for screen scraping data about Catholicism*

*#----------Import and mung the death from assault data---------------*

*# load deaths by assault from OECD.Stat*

**if**(**!**exists("viol\_sdmx")){

myUrl **<-** "http://stats.oecd.org/restsdmx/sdmx.ashx/GetData/HEALTH\_STAT/CICDHOCD.TXCMFETF+TXCMHOTH+TXCMILTX.AUS+AUT+BEL+CAN+CHL+CZE+DNK+EST+FIN+FRA+DEU+GRC+HUN+ISL+IRL+ISR+ITA+JPN+KOR+LUX+MEX+NLD+NZL+NOR+POL+PRT+SVK+SVN+ESP+SWE+CHE+TUR+GBR+USA+NMEC+BRA+CHN+COL+CRI+IND+IDN+LVA+LTU+RUS+ZAF/all?startTime=1960&endTime=2014"

dataset **<-** readSDMX(myUrl) *# takes about 30 seconds on a slow hotel internet connection*

viol\_sdmx **<-** as.data.frame(dataset) *# takes about 10 seconds on an i5*

}

The country codes provided by the OECD are ISO 3166 three digit country codes and we want to turn them into something more friendly for presentation purposes. We also want to rename the levels of our units (“TXCMILTX” is less friendly than “Deaths per 100 000 population”), order some factor levels for future graphics purposes, and knock out Turkey which only has a few year’s data.

*# load Country codes from ISOcodes R package*

data("ISO\_3166\_1")

*# mung:*

viol **<-** viol\_sdmx **%>%**

*# match country codes to country names:*

left\_join(ISO\_3166\_1[ , **c**("Alpha\_3", "Name")], by **=** **c**("COU" **=** "Alpha\_3")) **%>%**

*# more friendly names:*

rename(Country **=** Name,

Year **=** obsTime,

Value **=** obsValue) **%>%**

*# limit to columns we want:*

select(UNIT, Country, Year, Value) **%>%**

*# make graphics-friendly versions of Unit and Year variables:*

mutate(Unit **=** ifelse(UNIT **==** "TXCMILTX",

"Deaths per 100 000 population", "Deaths per 100 000 males"),

Unit **=** ifelse(UNIT **==** "TXCMFETF", "Deaths per 100 000 females", Unit),

Unit **=** factor(Unit, levels **=** unique(Unit)[**c**(2, 1, 3)]),

Year **=** **as.numeric**(Year)) **%>%**

*# not enough data for Turkey to be useful so we knock it out:*

filter(Country **!=** "Turkey")

Having created the original object viol I make a few summary and total objects to help with structuring my graphics. In particular, I need a data frame that provides average rates since 1990 for the first chart, and a data frame of totals so I can arrange the plots in order of increasing rates of death from assault.

*# create country x Unit summaries*

viol\_sum **<-** viol **%>%**

filter(Year **>** 1990) **%>%**

group\_by(Country, Unit) **%>%**

summarise(Value **=** mean(Value)) **%>%**

ungroup()

*# create country totals (for ordering in charts)*

totals **<-** viol\_sum **%>%**

group\_by(Country) **%>%**

summarise(Value **=** mean(Value)) **%>%**

arrange(Value)

*# create wider version, with one column per variable:*

viol\_spread **<-** viol\_sum **%>%**

mutate(Unit **=** **as.character**(Unit),

Unit **=** ifelse(grepl("female", Unit), "Female", Unit),

Unit **=** ifelse(grepl(" males", Unit), "Male", Unit),

Unit **=** ifelse(grepl("population", Unit), "Population", Unit)) **%>%**

spread(Unit, Value) **%>%**

mutate(Country **=** factor(Country, levels **=** totals**$**Country))

The actual plots

Now we’re ready to draw some plots. Here’s the code that makes the main two plots:

viol\_sum **%>%**

mutate(Country **=** factor(Country, levels **=** totals**$**Country)) **%>%**

mutate(Label **=** ifelse(grepl("population", Unit), **as.character**(Country), "|")) **%>%**

ggplot(aes(x **=** Value, y **=** Country)) **+**

geom\_segment(data **=** viol\_spread, aes(y **=** Country, yend **=** Country, x **=** Male, xend **=** Female),

colour **=** "white", size **=** 3) **+**

geom\_text(size **=** 4, aes(label **=** Label, colour **=** Unit), alpha **=** 0.8) **+**

labs(y **=** "") **+**

scale\_x\_log10(("Deaths per 100,000 (logarithmic scale)")) **+**

theme(legend.position **=** "bottom") **+**

scale\_colour\_manual("", values **=** **c**("red", "grey10", "blue")) **+**

labs(colour **=** "") **+**

ggtitle("Mean annual deaths from assault 1990 to 2013")

viol **%>%**

mutate(Country **=** factor(Country, levels **=** totals**$**Country)) **%>%**

ggplot(aes(x **=** Year, y **=** Value, colour **=** Unit)) **+**

facet\_wrap(**~**Country, scales **=** "free\_y", ncol **=** 5) **+**

geom\_smooth(se **=** **FALSE**, method **=** "loess") **+**

geom\_point(alpha **=** 0.8, size **=** 1) **+**

scale\_colour\_manual("", values **=** **c**("red", "grey10", "blue")) **+**

theme(legend.position **=** "bottom") **+**

labs(y **=** "Deaths per 100,000 per year - note changing vertical scale",

title **=** "Deaths from assault", x **=** "")

To investigate the Catholicism and gender ratio issues in the second set of plots, I had to pull down some data from the web using [rvest](https://cran.r-project.org/web/packages/rvest/index.html). It’s inspired by Python’s wonderful [Beautiful Soup](http://www.crummy.com/software/BeautifulSoup/) but I have to say is *astonishingly* user friendly.

Here’s my complete set of code for the gender dot plot, downloading the Catholicism data, and the resulting scatter plot. The web-scraping part occupies a total of two lines of code (21 and 22 in the code below) - amazing:

*# create a gender summary:*

viol\_gender **<-** viol **%>%**

filter(**!**grepl("population", Unit)) **%>%**

select(UNIT, Value, Year, Country) **%>%**

spread(UNIT, Value) **%>%**

mutate(ratio **=** TXCMHOTH **/** TXCMFETF) **%>%**

group\_by(Country) **%>%**

summarise(ratio **=** mean(ratio, tr **=** 0.2)) **%>%**

arrange(ratio) **%>%**

*# knock out Luxembourg and Iceland, too many NAs:*

filter(**!is.na**(ratio)) **%>%**

mutate(Country **=** factor(Country, levels **=** Country))

*# plot it:*

viol\_gender **%>%**

ggplot(aes(x **=** ratio, y **=** Country)) **+**

geom\_point() **+**

labs(x **=** "Trimmed mean annual ratio of male to female rates\nof deaths from assault over whole period", y **=** "")

*# download catholic proportion data*

cath\_page **<-** read\_html("http://www.catholic-hierarchy.org/country/sc1.html")

cath\_data **<-** html\_table(cath\_page)[[1]]

**names**(cath\_data) **<-** gsub(" ", "\_", **names**(cath\_data))

*# create vectors of country names for use in creating dummy variables:*

lat\_american **<-** **c**("Colombia", "Brazil", "Mexico", "Chile", "Costa Rica")

american **<-** **c**(lat\_american, **c**("Canada", "United States"))

*# mung:*

cath\_data **<-** cath\_data **%>%**

mutate(Country **=** gsub("M.+xico", "Mexico", Country),

Country **=** ifelse(Country **==** "USA", "United States", Country),

Country **=** ifelse(Country **==** "Great Britain", "United Kingdom", Country),

Country **=** ifelse(Country **==** "Korea (South)" , "Korea, Republic of", Country)) **%>%**

select(Country, Percent\_Catholic) **%>%**

mutate(Percent\_Catholic **=** **as.numeric**(gsub("%", "", Percent\_Catholic)),

Continent1 **=** ifelse(Country **%in%** american, "Americas", "Other"),

Continent2 **=** ifelse(Country **%in%** lat\_american, "Latin American", "Other"))

*# join the Catholic data to the assault data:*

viol\_gender\_cath **<-** viol\_gender **%>%**

left\_join(cath\_data, by **=** "Country")

*# set up base plot:*

cath1 **<-** viol\_gender\_cath **%>%**

ggplot(aes(x **=** Percent\_Catholic, y **=** ratio, label **=** Country)) **+**

geom\_smooth(method **=** "lm") **+**

geom\_text(family **=** "myfont", size **=** 3, alpha **=** 0.8) **+**

labs(x **=** "Percentage of country reported to be Catholic",

y **=** "Ratio of female to male\nassault death rates") **+**

xlim(-5, 110)

*# and the other two types of plots:*

cath2 **<-** cath1 **+** aes(colour **=** Continent1)

cath3 **<-** cath1 **+** aes(colour **=** Continent2)

*# draw plots:*

grid.arrange(cath1, cath2, cath3)

*# for reference, look explicitly at the underlying statistical models:*

mod1 **<-** lm(ratio **~** Percent\_Catholic, data **=** viol\_gender\_cath)

mod2 **<-** lm(ratio **~** Percent\_Catholic **\*** Continent1, data **=** viol\_gender\_cath)

mod3 **<-** lm(ratio **~** Percent\_Catholic **\*** Continent2, data **=** viol\_gender\_cath)

summary(mod1)

summary(mod2)

summary(mod3)

On the particular issue of suicides that make up so many firearm-related deaths, here is a new chart along similar lines to the assaults graphic from that earlier post:

Note that I’ve chosen to keep the vertical scales “fixed” this time, to give a better sense of comparison across countries. Also that the suicide numbers are higher for many countries than they are for deaths from assault, but less variable (both between countries, and over time), which is why fixed vertical scales is more of an option here.

Like the earlier post, this is based on the OECD’s synthesis of data from national governments. I can’t vouch for the data’s comparability across countries – there are some definite surprises here – other than to say that I’m sure every possible effort has been made by the various national statistical offices to get it right. Here’s the code to download and draw the chart of suicide rates:

library(tidyverse)

library(scales)

library(readr)

library(lubridate)

library(ggrepel)

library(viridis)

library(rstan)

library(rsdmx)

library(ISOcodes)

#----------contextual data on suicides---------------

url <- "http://stats.oecd.org/restsdmx/sdmx.ashx/GetData/HEALTH\_STAT/CICDHARM.TXCMFETF+TXCMHOTH+TXCMILTX.AUS+AUT+BEL+CAN+CHL+CZE+DNK+EST+FIN+FRA+DEU+GRC+HUN+ISL+IRL+ISR+ITA+JPN+KOR+LVA+LUX+MEX+NLD+NZL+NOR+POL+PRT+SVK+SVN+ESP+SWE+CHE+TUR+GBR+USA+NMEC+BRA+CHN+COL+CRI+IND+IDN+LTU+RUS+ZAF/all?startTime=2000&endTime=2020"

# Deaths per 100,000 population, standardised only

if(!exists("suic\_sdmx")){

dataset <- readSDMX(url)

suic\_sdmx <- as.data.frame(dataset)

}

data(ISO\_3166\_1)

suicide <- suic\_sdmx %>%

mutate(variable = ifelse(UNIT == "TXCMILTX", "All", ""),

variable = ifelse(UNIT == "TXCMHOTH", "Males", variable),

variable = ifelse(UNIT == "TXCMFETF", "Females", variable)) %>%

rename(year = obsTime, value = obsValue) %>%

select(year, COU, value, variable) %>%

left\_join(distinct(ISO\_3166\_1[ , c("Alpha\_3", "Alpha\_2", "Name")]), by = c("COU" = "Alpha\_3")) %>%

mutate(year = as.numeric(year),

country = Name) %>%

as\_tibble()

suicide %>%

mutate(

variable = fct\_reorder(variable, -value),

country = fct\_reorder(country, value)

) %>%

ggplot(aes(x = year, y = value, colour = variable)) +

facet\_wrap(~country) +

geom\_line() +

scale\_colour\_manual("", values = c("blue", "grey10", "red")) +

labs(x = "",

y = "Deaths per 100,000 population (standardised rates)") +

ggtitle("Suicide rates around the world (all methods)",

"USA is moderately high, but not as high as it features when compared on firearm-related deaths")

**Data sources**

The [Gun Violence Archive](http://www.gunviolencearchive.org/) aims to record every firearm violence event in the USA, certainly all those with fatalities. However, it won’t let you do a bulk download; and my requests for a way to do this have not gotten anywhere. The database query tool is set up to return individual cases rather than aggregate statistics, which is a shame for my purposes. In fact, only 500 incidents can be downloaded at once, and as far as I can tell it will give you the most recent 500 incidents matching your criteria. Because of the astonishing number of firearm violence incidents in the USA (the database includes non-lethal incidents), this makes downloading them all an impractical proposition without breaching the intent of the original limitation.

I tried a couple of things to get around this.

First, I queried the Gun Violence Archive database for each month in 2017 for incidents with at least one victim, and downloaded the c. 6,000 incidents it gave me as 12 CSV files (the latest 500 in the relevant month). This isn’t a simple random sample – events closer to the end of each month are more likely to be chosen; and events in shorter months eg February are more like to be chosen – but I thought it was close enough to act as though it was. I’ve saved all the data I got from this (and a few abortive additional efforts) in a zip file accessible from this website

Second, I queried the database for incidents with at least four victims (which could include both killed and injured) in each year from 2014 to 2018. Unless I misunderstood how the database worked, this returned less than 500 incidents each year. I then counted on screen (having first sorted by # Killed) the frequency of events with 4, 5, 6 or more people killed, and manually entered the data myself.

My plan was to treat 2017 as my year for a representative sample, then to supplement that with a special census of all the large events over a longer period. Then I could combine the two datasets for a model of the overall distribution that paid due attention to the rare, large events.

**The 2017 data**

Here’s how it looks if you just take the sample I got of 2017 data, from the first, simple method. In all the analysis that follows I exclude events with zero deaths. This is because I distrust the boundaries that need to be crossed for such events to get into the archive. I suspect there are large numbers of zero-death events that fail to make it to the archive due to the limitations of the reporting process. In fact, it is difficult even to define a zero-death violent event (how violent? how much does it need to be known to the authorities?); whereas once at least one death has taken place, the chances of reporting become much greater, and certainly the definition is clearer. This difficulty of counting zero-death events has big implications when we come to modelling the data later.

First, a histogram of the distribution of number of deaths in each event:

We can see that my sample missed the two rare events of 2017: 59 deaths at Las Vegas and 27 in the attack on at the church in Sutherland Springs, Texas. This is a nice illustration of how many events we have – even with a sample of 6,000 firearm incidents causing injury or death, we didn’t pick up the two highest profile such events in the year.

Then if we look at a scatterplot to get a sense of the basic question of what proportion of deaths come from different scale incidents:

Not only are single death incidents 80% of all firearm violence events with fatalities (remember, in this analysis I am excluding zero-death incidents), more than 80% of fatalities come from such events.

Or even a humble pie chart (yes, I believe they do have their purposes, if done properly):

Here’s the code to download the data and draw those charts.

#===============downloading gun violence archives==============

# to download these files you need to go to http://www.gunviolencearchive.org/query/

# and do a query, then choose download to CSV. It only downloads 500 at a time.

# "Save" actually means "Execute the query". Then you need to explicitly download CSV.

# And then you need to pick "download" (even though you're already in the "download CSV"

# environment and it has downloaded it = "download" here means "yes, really download it,

# and save it to my local drive")

# to save others doing this, a subset of the data is on my website

# download the data

try(unlink("gun-violence", recursive = TRUE))

dir.create("gun-violence")

download.file("http://freerangestats.info/data/gva.zip",

destfile = "gun-violence/gva.zip", mode = "wb")

unzip("gun-violence/gva.zip", exdir = "gun-violence")

# read into a list in R

files <- list.files("gun-violence", full.names = TRUE, pattern = ".csv$")

gva\_l <- lapply(files, read\_csv)

# convert into a data frame and do a bit of tidying up of dates etc

gva <- do.call("rbind", gva\_l) %>%

mutate(date = as.Date(`Incident Date`, format = "%B %d, %Y"),

year = year(date),

month = month(date)) %>%

# filter out zero deaths incidents, which we can't exclude from the original query

# (can only filter by number of victims, not number killed)

filter(`# Killed` > 0) %>%

# remove duplicates in case I downloaded some data twice or there is a database error:

distinct()

# this next graph not shown in the blog post, is just to illustrate that the data

# includes some non-2017 points to be wary of

gva %>%

ggplot(aes(x = date, y = `# Killed`)) +

geom\_jitter(alpha = 0.1) +

ggtitle("Fatal firearm incidents in the USA; 500 sampled from selected months",

"Points are jittered to give a sense of how much concentration there is in '1 death per incident'") +

labs(x = "", y = "Number killed in each incident",

caption = "Searches from Gun Violence Archive")

# Only 2014 onwards is complete data

# Only 2017 is at all representative; the rest includes various other downloads!

# Histogram

gva %>%

filter(year == 2017) %>%

ggplot(aes(x = `# Killed`)) +

geom\_histogram(binwidth = 1, aes(y = ..density..), fill = "steelblue", alpha = 0.8) +

scale\_x\_continuous(breaks = 1:8) +

scale\_y\_continuous("Percentage of events", label = percent) +

ggtitle("Random sample of firearm-events in 2017",

"Excludes events with zero people killed")

# Scatterplot

gva %>%

filter(year == 2017) %>%

mutate(n\_killed = paste0("'", ifelse(`# Killed` < 5, `# Killed`, "More than 4"), " death' incidents")) %>%

group\_by(n\_killed) %>%

summarise(incidents = n(),

deaths = sum(`# Killed`)) %>%

ungroup() %>%

mutate(incidents = incidents / sum(incidents),

deaths = deaths / sum(deaths)) %>%

ggplot(aes(x = incidents, y = deaths, label = n\_killed, colour = n\_killed)) +

geom\_point(size = 2) +

geom\_text\_repel() +

theme(legend.position = "none") +

scale\_x\_continuous("Proportion of incidents", label = percent) +

scale\_y\_continuous("Proportion of deaths", label = percent) +

ggtitle("The vast majority of firearm deaths come from single-death incidents.",

"Estimated firearms incidents and deaths in the USA in 2017, aggregated by number of deaths per incident"

) +

# scale\_color\_viridis(discrete = TRUE, option = "D")

scale\_color\_brewer(palette = "Set2")

# Pie chart

blank\_theme <- theme\_minimal() +

theme(

axis.title.x = element\_blank(),

axis.title.y = element\_blank(),

panel.border = element\_blank(),

panel.grid=element\_blank(),

axis.ticks = element\_blank(),

plot.title=element\_text(size=14, face="bold"),

axis.text.x = element\_blank()

)

gva %>%

filter(year == 2017) %>%

mutate(n\_killed = paste0("'", ifelse(`# Killed` < 5, `# Killed`, "More than 4"), " death' incidents")) %>%

group\_by(n\_killed) %>%

summarise(incidents = n(),

deaths = sum(`# Killed`)) %>%

ungroup() %>%

mutate(incidents = incidents / sum(incidents),

deaths = deaths / sum(deaths)) %>%

ggplot(aes(x = "", weight = deaths, fill = n\_killed, stat = "identity")) +

# make width 0.5 if prefer a donut...

geom\_bar(width = 1) +

coord\_polar("y", start = 0) +

scale\_fill\_brewer("Number of deaths per incident", palette = "Set2",

guide = guide\_legend(reverse = TRUE)) +

blank\_theme +

ggtitle("The vast majority of firearm deaths come from single-death incidents.",

"Estimated firearms deaths in the USA in 2017, aggregated by number of deaths per incident")

**Truncated negative binomial distribution**

My starting point for modelling the count of deaths per event was a negative binomial distribution. This is often a good model for counts that vary more than a Poisson distribution would predict, which I was certain would be the case here. In fact, one way of deriving a negative binomial distribution is as a whole collection of different Poisson distributions, whose means themselves have a Gamma distribution. This lets a negative binomial distribution cover quite a variety of types of count data.

I have two datasets to fit with this distribution:

* a sample from 2017 truncated so only those with one or more deaths were observed (ie I have discarded the zero-killed events as probably incomplete, as mentioned above)
* a census from 2014 to 2018 that is complete for high-death events but truncated so only those with four or more deaths were observed.

Both are cases of truncated data for which we are interested in the underlying negative binomial distribution that plausibly could have generated it. Even the 2017 dataset of incidents with one or more deaths is truncated, by failing to include the great number of zero death incidents. Remember, I had doubts about whether these could be archived sufficiently comprehensively, hence decided to exclude them systematically.

There are a variety of ways to fit this model, but I chose to write a simple Stan program. I started with a very broad prior distribution for my expectation of what the underlying negative binomial distribution might be, and refined this by fitting the first, large set of 2017 data to it. Then I took the posterior distribution for the parameters of that negative binomial distribution and used them as the prior when fitting the model to just the larger, rarer events of four deaths or more per incident. My strategy here was to first get a broad sense of the distribution from the random sample, then to refine it so it matches with the extreme events.

As things turned out, the estimation tactic worked fine but the model didn’t work particularly well. I think no simple negative binomial distribution will adequately model this data. Here’s a comparison of the two original datasets (truncated at 1+ deaths for 2017, and truncated at 4+ deaths for all available years) with data for 100,000 events simulated from the model after fitting it in this two-step process:

It fits badly in two aspects noticeable to the human eye:

* There are too *many* predicted incidents with between 2 and 10 people killed. This is most obvious in the left-most panels, where we can see the actual distribution has much more probability mass concentrated at 1 death, and drops off more rapidly, compared to the data simulated from the model.
* There are too *few* predicted incidents with 50+ people killed. Effectively none of the mass of the modelled distribution is to the right of 20 deaths, and it has no explanation for two events with 50+ people killed in the space of two years (Orlando and Las Vegas), or even three others between 15 and 30 in the past three years.

Instead, the response variable should have a distribution that varies conditional on which part of the mixture the event is. I suspect a plausible model might be something like five types of incidents:

* suicides that don’t involve homicides
* family and intimate partner related violence
* aimless non-family inter-personal violence (eg road rage, argument at the pub)
* related to crimes other than family violence (eg theft, drugs, turf wars)
* mass shootings

The last of these stands out as an event where the perpetrator aims for as many deaths as possible. However, there are obvious grey areas and overlaps between several of the categories (as noted earlier, family violence is a factor in about half of mass shootings). And for my purposes, I simply don’t have access to the explanatory variables necessary to do this kind of modelling.

I made one half-hearted attempt at modelling the data as a mixture of two distributions but it wasn’t much better than what I had originally. As well as the mixture question, I’m pretty sure a more skewed right-tailed distribution than the negative binomial will be needed, and that takes me well beyond this blog post.

Anyway, here’s how I did the two stage modelling. Perhaps it will be more useful in other situations with truncated count data. First, here’s the program written in Stan that fits a negative binomial model to truncated count data. It’s quite similar to the program in my last blog on fitting a model to truncated Poisson count data.

data {

int n;

int lower\_limit;

int x[n];

real mu\_prior\_mean;

real mu\_prior\_sd;

real phi\_prior\_mean;

real phi\_prior\_sd;

}

parameters {

realmu;

realphi;

}

model {

mu ~ normal(mu\_prior\_mean, mu\_prior\_sd);

phi ~ normal(phi\_prior\_sd, phi\_prior\_sd);

for(i in 1:n){

x[i] ~ neg\_binomial\_2(mu, phi) T[lower\_limit, ];

}

}

And here’s the R code that fits two sets of data to two models

#=================neg binomial modelling====================

options(mc.cores = 8)

#------------------2017 sample first--------------------

data <- list(

# x = sample(filter(gva, year == 2017)$`# Killed`, 100),

x = filter(gva, year == 2017)$`# Killed`,

lower\_limit = 1,

mu\_prior\_mean = 1,

mu\_prior\_sd = 10,

phi\_prior\_mean = 5,

phi\_prior\_sd = 20

)

data$n <- length(data$x)

fit1 <- stan("0119-trunc-negbin.stan", data = data)

#----------------------2016 data second-------------------

# hand comparisons of the CSV download data with that on the screen at GVA show that

# some kind of unusual sampling is going on with dispoprotionate numbers of

# incidents with 4 deaths being dropped from the data. It's not possible to download

# reliably as a CSV so we'll just enter the values.

other\_years <- rbind(

data\_frame(

killed = c(4, 5, 6,7,8),

freq = c(25, 8, 1, 1, 1),

year = 2014

),

data\_frame(

killed = c(4, 5, 6, 8, 9, 10, 16),

freq = c(31, 7, 2, 2, 3, 1, 1),

year = 2015

),

data\_frame(

killed = c(4,5,6,8,50),

freq = c(23,13,3,1,1),

year = 2016

),

data\_frame(

killed = c(4,5,6,8,9,27,59),

freq = c(33, 4, 2, 1, 1, 1, 1),

year = 2017

),

data\_frame(

killed = c(4, 5, 17),

freq = c(11, 3, 1),

year = 2018

)

)

data <- list(

x = rep(other\_years$killed, other\_years$freq),

lower\_limit = 4,

mu\_prior\_mean = summary(fit1)$summary["mu", "mean"],

mu\_prior\_sd = summary(fit1)$summary["mu", "sd"],

phi\_prior\_mean = summary(fit1)$summary["phi", "mean"],

phi\_prior\_sd = summary(fit1)$summary["phi", "sd"]

)

data$n <- length(data$x)

fit2 <- stan("0119-trunc-negbin.stan", data = data)

#---------------comparison of modelled to actual-----------

phi <- summary(fit2)$summary["phi", "mean"]

mu <- summary(fit2)$summary["mu", "mean"]

x\_sim <- rnbinom(100000, size = phi, mu = mu)

comp\_data <- rbind(

data\_frame(x = rep(other\_years$killed, other\_years$freq), source = "Actual, 4+ deaths"),

data\_frame(x = filter(gva, year == 2017)$`# Killed`, source = "Actual, 1+ deaths"),

data\_frame(x = x\_sim[x\_sim >= 1], source = "Simulated, 1+ deaths"),

data\_frame(x = x\_sim[x\_sim >= 4], source = "Simulated, 4+ deaths")

)

comp\_data %>%

ggplot(aes(x = x, y = ..density..)) +

facet\_wrap(~source) +

geom\_histogram(binwidth = 1) +

ggtitle("Comparison of actual and simulated deaths per incident",

"Negative binomial model fit iteratively to two sets of truncated data.

Simulated data based on the model does not feature the rare, large events observed in reality.")

**Summary**

To recap:

* I wanted to characterise the distribution of deaths per firearm incident the USA, and in particular try to model it with a single negative binomial distribution.
* I couldn’t get the full data I needed from the Gun Violence Archive, aggregated by deaths per incident. So I downloaded a representative sample of data from 2017 for incidents with at least 1 death, and a census sample of data for 2014 to present (early 2018) for incidents with at least 4 deaths.
* I used Bayesian methods to fit negative binomial distributions to the truncated data. I started with a very broad prior distribution for the mean and shape, and narrowed down estimates for those parameters by fitting to the data truncated at 1 death. Then I used the posterior distribution for the parameters from that model as the prior distribution for fitting to the data truncated at 4 deaths.
* The estimation method worked, but showed that a single negative binomial distribution is too simple a model to fit this data. There is a wider dispersal of counts of deaths per incident than the model allows for. In particular, in reality there are even more incidents with one death than modelled; and also a small number of high-death incidents that do not fit well into the model. An alternative model that takes into account the mixture of several different distributions, and that is better at predicting rare, large events, is needed.