**Did the average rate of Australian mass-shooting decline after 1996, or was the drop just chance?**

The question under investigation is whether the drop in mass-shooting events in Australia since the change in the firearm regulatory environment in 1996 could be a result of chance or not. “Mass shootings” are defined as homicides in which at least five persons died, not including the perpetrator. There were 13 of these events in the 18 years from 1979 up to the time of the National Firearms Agreement, and none afterwards.

Chapman et al model the events with a [Poisson point process](https://en.wikipedia.org/wiki/Poisson_point_process) and to replicate their findings. However, the code they provide is somewhat compact and tersely commented, and having familiarised myself with how it works I thought it worthwhile blogging about in a somewhat more verbose manner.

We are putting aside controversies about whether five persons is the correct threshold to consider, whether these events should be restricted only to deaths from rapid fire weapons, and analysis of general trends and possible confounding factors. For the purpose of this blog I am just approaching this as an edifying illustration of the modelling of Poisson point processes.

**Data familiarisation**

Let’s start with visualising the data. Here’s a chart showing each mass-shooting event as a vertical strip in comparison to the timing of the regulatory changes:

That’s pretty dramatic and it passes Tukey’s intra-ocular impact significance test (ie hits you between the eyes)[1](http://freerangestats.info/blog/2019/09/07/mass-shootings-oz#fn:1). Here’s the code that sets up our data for that visulaisation and the more formal tests to come:

*Post continues after R code*

library(tidyverse)

library(NHPoisson)

library(scales)

#----------------Data prep and first visualisation----------------

mass\_shootings <- data.frame(

mon = c(9, 1, 6, 8, 10, 12, 9, 8, 8, 10, 3, 1, 4),

yr = c(1981, 1984, 1987, 1987, 1987, 1987, 1988, 1990, 1991, 1992, 1993, 1996, 1996)

)

mass\_shootings <- mass\_shootings %>%

mutate(months = 12 \* (yr - 1979) + mon,

approx\_date = as.Date(paste(yr, mon, 15, sep = "-")),

interval = c(NA, diff(approx\_date)))

p1 <- ggplot(mass\_shootings, aes(xend = approx\_date, x = approx\_date)) +

geom\_rect(xmin = as.Date("1996-07-15"), xmax = Inf, ymin = -Inf, ymax = Inf,

fill = "steelblue", alpha = 0.01) +

geom\_rect(xmax = as.Date("1996-07-15"), xmin = -Inf, ymin = -Inf, ymax = Inf,

fill = "red", alpha = 0.01) +

geom\_segment(y = -Inf, yend = Inf) +

scale\_x\_date(limits = c(min(mass\_shootings$approx\_date - 50), as.Date("2018-02-15"))) +

theme(panel.grid = element\_blank(),

panel.border = element\_blank(),

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

axis.ticks.y = element\_blank()) +

labs(x = "Approximate date of mass shooting",

y = "") +

annotate("text", x = as.Date("2008-01-01"), y = 1, label = "After the gun buy-back",

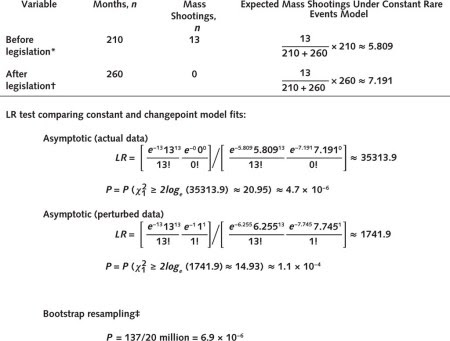
colour = "steelblue") +

ggtitle("Firearm-related homicides in Australia",

"Events in which at least 5 persons other than perpetrator died")

**Using likelihood ratio to compare two hypothesis**

Chapman et al provide this table of results in their main article:



The numbers above the line in that image are used to define:

* their null hypothesis (that the underlying rate of mass shootings is the same over the whole period); and
* an alternative (that there are two different underlying rates, once in the first 210 months and once in the second:

Under the null hypothesis we would expect to see 5.809 events in the first period of 210 months, then 7.191 in the second period of 260 months. Under the alternative hypothesis (which is purely driven by the data), we expect to see 13 in the first period and zero in the second.

**Simple calculation of the likelihood ratio test**

The likelihood ratio calculated under the heading of “Asymptotic (actual data)” comes directly from the known properties of a homogenous Poisson process. In such a process, events occur independently at intervals in time (and or space – these processes generalise to more than one dimension) which follow an exponential distribution. In any fixed amount of time, the count of events seen in such a process has a Poisson distribution; and the formula beginning “LR =” comes directly from calculating the likelihood of the observed data with such a distribution.

The ratio of 35,313.9 is how much more “likely” the data are to have been generated by the alternative hypothesis than the null. The calculation of the p value afterwards comes from conferting that ratio into a drop in “deviance” and comparing that to a Chi-squared distribution with one degree of freedom; the “alternative” hypothesis is one parameter more complex than the null (because there are two average rates of shootings over the full period, rather than one), so the drop in deviance we would expect to see from pure randomness if the null were true would follow that particular Chi-squared distribution.

It turns out that the actual drop in deviance (20.95) is far higher than can plausibly come from chance, with a p-value of 0.0000047. All of this is textbook statistical inference and the calculations are produced with this code:

*Post continues after R code*

#-------------------------Modelling and likelihood ratio test-----------

# number of mass shootings in first period:

n\_massacres <- nrow(mass\_shootings)

# Expected mass shootings in each of the two periods, under constant rare events model:

lam0 <- (210 / 470) \* n\_massacres

lam1 <- (260 / 470) \* n\_massacres

# Put those two expected values into a table:

cbind(lam0, lam1)

# Reduction in deviance comparing the two hypotheses

logLRobs <- 2 \* (dpois(n\_massacres, n\_massacres, log = TRUE) -

(dpois(n\_massacres, lam0, log = TRUE) + dpois(0, lam1, log = TRUE)))

# Print reduction in deviance to screen:

logLRobs

# Calculate p-value if null hypothesis of a random drop in deviance:

1 - pchisq(logLRobs, df = 1)

That reproduces the results shown in the top half of the results

**Robustness check – “one more” event**

The results reported under “Asymptotic (perturbed data)” are the first robustness check conducted by Chapman et al. They considered “what if there had been one more massacre in the period after the regulatory changes – for example, as our article goes to print?”. This is a very sensible check.

The calculations are the same as in the original case, except that some zeroes become ones; the average rate under the null is now 14 / (210 + 260), and the expected number of events in the two periods goes to 6.255 and 7.745. The reduction in deviance in this case is much less than previously, but the p value is still far below conventional threshold needed to dismiss the null hypothesis of a constant rate over the full period.

*Post continues after R code*

#--------------------------------Robustness - would an extra, late massacre matter?---------------

# Robustness check - what if there was an extra massacre happening in the post-buyback period

mu0 <- ((n\_massacres + 1) / (210 + 260)) \* 210

mu1 <- ((n\_massacres + 1) / (210 + 260)) \* 260

cbind(mu0,mu1)

logLRperturb <- 2 \* ((dpois(n\_massacres, n\_massacres, log = TRUE) + dpois(1, 1, log = TRUE))-

(dpois(n\_massacres, mu0, log = TRUE) + dpois(1, mu1, log = TRUE)))

logLRperturb

1 - pchisq(logLRperturb, df = 1)

**Robustness check – resampling**

The above calculations depend upon the large sample properties of a homogenous Poisson point process. However, 13 events over 39 years is not a very large sample. So Chapman et al rightly did a further check of comparing the observed drop in deviance from null to alternative hypothesis, not with the theoretical Chi-square distribution but with the distribution of drops in deviance from a large set of data generated by computer under the null hypothesis. The result is the slightly higher but still vanishingly small p value of 0.0000069.

The original authors don’t report it, but the same comparison done to the drop in deviance under the “perturbed” set of data (with an extra mass shooting in the late period) gives a p value of 0.0002 – nearly twice the reported p value for the perturbed data from the asymptotic distribution, but still far too small to think that the reduction of mass shootings in the later period could plausibly be from chance with an average rate over the entire time period.

*Post continues after R code*

#------------Comparison with resampling----------------

# Force R to use non-uniform Rounding sampler, as per older versions of R, to get exact results

RNGkind(sample.kind="Rounding")

set.seed(20180226)

n\_sim <- 20000000

logLRsim <- 0

for (i in 1:n\_sim){

x <- rpois(1, lam0)

y <- rpois(1, lam1)

lam0sim <- (x + y) / (210 + 260)

logL0 <- dpois(x, lam0sim \* 210, log = TRUE) + dpois(y, lam0sim \* 260, log = TRUE)

logL1 <- dpois(x, x, log = TRUE) + dpois(y, y, log = TRUE)

logLRsim[i] <- 2 \* (logL1 - logL0)

}

# Compare the drop of deviance actually observed with that simulated under the null hypothesis:

no\_exceeding <- sum(logLRsim >= logLRobs)

no\_exceeding

no\_exceeding / n\_sim

# same comparison, with the "one more recent massacre" perturbed data's drop in deviance:

no\_exceeding\_perturb <- sum(logLRsim >= logLRperturb)

no\_exceeding\_perturb / n\_sim

**Investigating clumping**

The final piece of analysis by the original authors was an investigation into whether “clumping” of events might invalidate their results. The above calculations, including those that compared the drop in deviance with simulated results that account for small sample, all rely on the model of the data as coming from a Poisson point process in the first place. A key part of that model is that events occur independently, in time intervals that follow an exponential distribution. If we look at the actual time intervals, we see that the exponential distribution is only an approximation, as of course is to be expected with real life data and a fairly small sample:

The grey shaded area is the empirical distribution of the intervals between events and the blue line is the theoretical exponential distribution under the “two different underlying rate” alternative hypothesis. That graphic was made with this code:

ggplot(mass\_shootings, aes(x = jitter(interval, 2.5))) +

geom\_density(fill = "grey", alpha = 0.5, colour = NA) +

geom\_rug() +

stat\_function(fun = dexp,

args = list(rate = 1 / mean(mass\_shootings$interval, na.rm = TRUE)),

colour = "blue",

size = 2) +

labs(x = "Interval between mass shootings in Australia 1981 to 1996",

title = "Comparison of intervals between mass shooting with theoretical independence",

subtitle = "There are fewer close-together shootings, and more far-apart, than expected from exponential distribution") +

scale\_x\_continuous(label = comma)

Interestingly, four very evenly spaced mass shooting events in 1987 were each 61 days apart (after my approximation of saying all events are on the 15th of the month – I don’t have the exact dates, only the month of occurrence) so I had to jitter the data a bit for it all to show up in the “rug marks” along the bottom of that plot.

A critique of some of Chapman’s earlier work in this space had suggested that the pre-1996 shootings were a cluster of non-independent events. In analysis of stochastic processes this is called “clumping”, a term that is more intuitive when considering a two dimensional Poisson point process for the positioning of (for example) trees than events in time, but the principle is the same. If, for example, mass shootings led to copycat events at relatively short intervals, followed by long periods when no-one breaks the ice with a shooting, then the statistical tests in the analysis so far would be over-stating the evidence against a constant underlying rate of mass shootings.

To check against this, Chapman et al used some elegant techniques to compare the clumping in the observed data to the amount of clumping seen in genuine Poisson processes. For this step, the null hypothesis is that the data are from a genuine Poisson process, and we are looking for evidence against that null hypothesis in the form of a p value suggesting that the observed data are unlikely (too clumpy) to have come from such a process. This is all done with simulation methods.

They start this check by observing the highest number of events per window in simulated data (up to a window of 18 months), and storing this in an object called max\_stat\_mat in the code below (differently named in their original code). Then, with the actual data, they calculate for each possible window the highest number of events taking place within that window – and compare this to its place in the distribution of the simulated data. This gives us a set of raw p values for how unlikely the clumping is for each window:

| **window** | **stat\_obs** | **p\_vals** |
| --- | --- | --- |
| 1 | 1 | 1.0000 |
| 2 | 1 | 1.0000 |
| 3 | 2 | 0.7522 |
| 4 | 2 | 0.8542 |
| 5 | 3 | 0.2143 |
| 6 | 3 | 0.3067 |
| 7 | 4 | 0.0488 |
| 8 | 4 | 0.0734 |
| 9 | 4 | 0.1058 |
| 10 | 4 | 0.1434 |
| 11 | 4 | 0.1842 |
| 12 | 4 | 0.2258 |
| 13 | 4 | 0.2693 |
| 14 | 4 | 0.3074 |
| 15 | 4 | 0.3531 |
| 16 | 5 | 0.0990 |
| 17 | 5 | 0.1226 |
| 18 | 5 | 0.1482 |

The most suspect window length is 7 months. For this window, the observed clumping was more than 95.1% of the simulations. However, we can’t use this p value of 0.049 to reject the null hypothesis of no clumping yet, because we have chosen that null hypothesis only after observing the data (that is, we picked 7 months as the window most likely to show clumping from the data). To get a “proper” p value we need to adjust this again by comparing to what we would have seen by chance. That is, some window is always going to generate the lowest p value by this method; how often will it be as low as 0.049? Adjusting it this way gets us an actual p value of 0.095 – not low enough to dismiss the null hypothesis of the data coming from a genuine Poisson process.

An interesting point about reproducibility here. When I first ran the code directly from the supplement to the original article, I got different results at this point to those reported, although still in line with the substantive conclusions. One of the original authors, Michael Stewart, was able to put me on to the reason why. In the code for this blog, I used RNGkind(sample.kind="Rounding") early in the script to revert to the old behaviour. This is certainly something worth knowing about when trying to reproduce pre-2019 simulation results.

I think I further complicate reproducibility through my code additions – and in particular the use of randomness in jitter to help show up the rugs in one of my plots. My eventual results are close enough to the published not to worry about this, but it’s something worth remembering when going for strict reproducibility, that randomness comes in a lot of ways. For robust reproducibility, it would be sensible to set the random seed with set.seed() before each key piece of simulation analysis, to control for dangers coming from restructuring code.

*Post continues after R code*

#' Calculate the highest number of events in a given window

#'

#' @param months a vector of times at which events took place

#' @param window length of window

#' @return The highest number of events occuring in the given window of time, in the given list of intervals of events

scan\_stat <- function(months, window){

sum\_window <- 0

for (j in 1:(210 - window + 1)){

sum\_window[j] = sum((j <= months) & (months < j + window))

}

max(sum\_window)

}

#### Simulated data from a theoretical poisson process

n\_sim <- 10000

max\_window <- 18

# matrix to hold results for simulated data

max\_stat\_mat <- matrix(0, n\_sim, max\_window)

for (j in 1:n\_sim){

N <- rpois(1, n\_massacres)

months\_sim <- sort(sample(1:210, size=N, repl=FALSE))

for (i in 1:max\_window){

max\_stat\_mat[j,i] <- scan\_stat(months\_sim, i)

}

}

#### Compare what actually happened to the simulated data

p\_vals <- 0

stat\_obs <- 0

window <- 1:max\_window

for (k in window) {

stat\_obs[k] <- scan\_stat(mass\_shootings$months, k)

p\_vals[k] <- mean(max\_stat\_mat[, k] >= stat\_obs[k])

}

# all unadjusted p values:

cbind(window, stat\_obs, p\_vals)

# the lowest unadjusted p value:

unadj\_pval <- min(p\_vals)

# adjust that p value for all the data dredging we've done so far by comparing our result (lowest p value)

# to what we get by simulating everything from an actual Poisson process

M\_sim <- 10000

pvals\_sim <- 0

min\_pval\_sim <- 0

for (a in 1:M\_sim){

N <- rpois(1, n\_massacres)

months\_sim <- sort(sample(1:210, size = N,repl = FALSE))

stat\_obs\_sim <- 0

for (b in window){

stat\_obs\_sim[b] <- scan\_stat(months\_sim,b)

pvals\_sim[b] <- mean(max\_stat\_mat[,b] >= stat\_obs\_sim[b])

}

min\_pval\_sim[a] <- min(pvals\_sim)

}

# On average, how often is the minimum p value we just simulated less than the unadjusted p value

# we got from the comparison of data to simulations?

adj\_pval <- mean(min\_pval\_sim <= unadj\_pval)

adj\_pval

**Conclusion**

This analysis is pretty robust. As the original authors state:

a standard rare events model provides strong evidence against the hypothesis that this prolonged absence simply reflects a continuation of a preexisting pattern of rare events.

So taken as a given issues such as whether the 13 events are the right ones to count and what to do about other confounding trends, we can be pretty confident in that conclusion.

**The R Script**

* Structure, sequencing and content
  + bring the definition of the data (number of shootings and the month they are in) up to the front and make a visualisation of it before we get into the analysis
  + change the order of some of the analysis to match the presentation of results eg put the calculation of the likelihood ratio from pertured data ahead of the results from bootstrap resampling of the original
* Discipline with objects and variables
  + with data that is intrinsically equal lengthed (such as the month and the year of each shooting – stored as vectors mon and yr in the original) store them in a data frame or tibble which provides the discipline of assuring that they are equally lengthed columns of data
  + replace some magic constants (such as “13”, the number of massacres) with variables calculated by R (eg n\_massacres <- nrow(mass\_shootings), then use n\_massacres instead of 13 from then onwards) – for maintainability and portability of the code to other use cases, and also for readability (it took me a while to spot the 13s in the code, whereas I find n\_massacres very readable)
  + change T and F to TRUE and FALSE throughout
* Documentation, readability and style
  + add section and subsection markers
  + more comments explaining the “why” of each step
  + document the purpose and parameters of functions with “roxygen2 – style” (eg @param)
  + spaces after commas, arithmetic operators, assignment operators etc – just for readability
  + replace dots in variable names with underscores

|  |
| --- |
| library(tidyverse) |
|  | library(NHPoisson) |
|  | library(scales) |
|  | library(knitr) |
|  | library(clipr) |
|  | library(kableExtra) |
|  |  |
|  | #----------------Data prep and first visualisation---------------- |
|  |  |
|  | mass\_shootings <- data.frame( |
|  | mon = c(9, 1, 6, 8, 10, 12, 9, 8, 8, 10, 3, 1, 4), |
|  | yr = c(1981, 1984, 1987, 1987, 1987, 1987, 1988, 1990, 1991, 1992, 1993, 1996, 1996) |
|  | ) |
|  |  |
|  | mass\_shootings <- mass\_shootings %>% |
|  | mutate(months = 12 \* (yr - 1979) + mon, |
|  | approx\_date = as.Date(paste(yr, mon, 15, sep = "-")), |
|  | interval = c(NA, diff(approx\_date))) |
|  |  |
|  | p1 <- ggplot(mass\_shootings, aes(xend = approx\_date, x = approx\_date)) + |
|  | geom\_rect(xmin = as.Date("1996-07-15"), xmax = Inf, ymin = -Inf, ymax = Inf, |
|  | fill = "steelblue", alpha = 0.01) + |
|  | geom\_rect(xmax = as.Date("1996-07-15"), xmin = -Inf, ymin = -Inf, ymax = Inf, |
|  | fill = "red", alpha = 0.01) + |
|  | geom\_segment(y = -Inf, yend = Inf) + |
|  | scale\_x\_date(limits = c(min(mass\_shootings$approx\_date - 50), as.Date("2018-02-15"))) + |
|  | theme(panel.grid = element\_blank(), |
|  | panel.border = element\_blank(), |
|  | axis.text.y = element\_blank(), |
|  | axis.ticks.y = element\_blank()) + |
|  | labs(x = "Approximate date of mass shooting", |
|  | y = "") + |
|  | annotate("text", x = as.Date("2008-01-01"), y = 1, label = "After the gun buy-back", |
|  | colour = "steelblue") + |
|  | ggtitle("Firearm-related homicides in Australia, to February 2018", |
|  | "Events in which at least 5 persons other than perpetrator died") |
|  |  |
|  | frs::svg\_png(p1, "../img/0158-events") |
|  |  |
|  |  |
|  | #-------------------------Modelling and likelihood ratio test----------- |
|  | # number of mass shootings in first period: |
|  | n\_massacres <- nrow(mass\_shootings) |
|  |  |
|  | # Expected mass shootings in each of the two periods, under constant rare events model: |
|  | lam0 <- (210 / 470) \* n\_massacres |
|  | lam1 <- (260 / 470) \* n\_massacres |
|  |  |
|  | # Put those two expected values into a table: |
|  | cbind(lam0, lam1) %>% kable() %>% kable\_styling() %>% write\_clip() |
|  |  |
|  | # Reduction in deviance comparing the two hypotheses |
|  | logLRobs <- 2 \* (dpois(n\_massacres, n\_massacres, log = TRUE) - |
|  | (dpois(n\_massacres, lam0, log = TRUE) + dpois(0, lam1, log = TRUE))) |
|  |  |
|  | # Print teducation in deviance to screen: |
|  | logLRobs |
|  |  |
|  | # Calculate p-value if null hypothesis of a random drop in deviance: |
|  | 1 - pchisq(logLRobs, df = 1) |
|  |  |
|  | #--------------------------------Robustness - would an extra, late massacre matter?--------------- |
|  | # Robustness check - what if there was an extra massacre happening in the post-buyback period |
|  | mu0 <- ((n\_massacres + 1) / (210 + 260)) \* 210 |
|  | mu1 <- ((n\_massacres + 1) / (210 + 260)) \* 260 |
|  | cbind(mu0,mu1) |
|  |  |
|  | logLRperturb <- 2 \* ((dpois(n\_massacres, n\_massacres, log = TRUE) + dpois(1, 1, log = TRUE))- |
|  | (dpois(n\_massacres, mu0, log = TRUE) + dpois(1, mu1, log = TRUE))) |
|  | logLRperturb |
|  |  |
|  | 1 - pchisq(logLRperturb, df = 1) |
|  |  |
|  |  |
|  | #------------Comparison with resampling---------------- |
|  | # Force R to use non-uniform Rounding sampler, as per older versions of R, to get exact results |
|  | RNGkind(sample.kind="Rounding") |
|  | set.seed(20180226) |
|  |  |
|  | n\_sim <- 20000000 |
|  | logLRsim <- 0 |
|  | for (i in 1:n\_sim){ |
|  | x <- rpois(1, lam0) |
|  | y <- rpois(1, lam1) |
|  | lam0sim <- (x + y) / (210 + 260) |
|  | logL0 <- dpois(x, lam0sim \* 210, log = TRUE) + dpois(y, lam0sim \* 260, log = TRUE) |
|  | logL1 <- dpois(x, x, log = TRUE) + dpois(y, y, log = TRUE) |
|  | logLRsim[i] <- 2 \* (logL1 - logL0) |
|  | } |
|  |  |
|  | # Compare the drop of deviance actually observed with that simulated under the null hypothesis: |
|  | no\_exceeding <- sum(logLRsim >= logLRobs) |
|  | no\_exceeding |
|  | no\_exceeding / n\_sim |
|  |  |
|  | # same comparison, with the "one more recent massacre" perturbed data's drop in deviance: |
|  | no\_exceeding\_perturb <- sum(logLRsim >= logLRperturb) |
|  | no\_exceeding\_perturb / n\_sim |
|  |  |
|  |  |
|  |  |
|  |  |
|  | #---------------Robustness - was there clumping to invalidate our assumptions--------------- |
|  |  |
|  | p2 <- ggplot(mass\_shootings, aes(x = jitter(interval, 2.5))) + |
|  | geom\_density(fill = "grey", alpha = 0.5, colour = NA) + |
|  | geom\_rug() + |
|  | stat\_function(fun = dexp, |
|  | args = list(rate = 1 / mean(mass\_shootings$interval, na.rm = TRUE)), |
|  | colour = "blue", |
|  | size = 2) + |
|  | labs(x = "Interval between mass shootings in Australia 1981 to 1996", |
|  | title = "Comparison of intervals between mass shooting with theoretical independence", |
|  | subtitle = "There are fewer close-together shootings, and more far-apart, than expected from exponential distribution") + |
|  | scale\_x\_continuous(label = comma) |
|  |  |
|  | frs::svg\_png(p2, "../img/0158-exp-distribution") |
|  |  |
|  | #' Calculate the highest number of events in a given window |
|  | #' |
|  | #' @param months a vector of times at which events took place |
|  | #' @param window length of window |
|  | #' @return The highest number of events occuring in the given window of time, in the given list of intervals of events |
|  | scan\_stat <- function(months, window){ |
|  | sum\_window <- 0 |
|  | for (j in 1:(210 - window + 1)){ |
|  | sum\_window[j] = sum((j <= months) & (months < j + window)) |
|  | } |
|  | max(sum\_window) |
|  | } |
|  |  |
|  |  |
|  | #### Simulated data from a theoretical poisson process |
|  | n\_sim <- 10000 |
|  | max\_window <- 18 |
|  |  |
|  | # matrix to hold results for simulated data |
|  | max\_stat\_mat <- matrix(0, n\_sim, max\_window) |
|  |  |
|  | for (j in 1:n\_sim){ |
|  | N <- rpois(1, n\_massacres) |
|  | months\_sim <- sort(sample(1:210, size=N, repl=FALSE)) |
|  | for (i in 1:max\_window){ |
|  | max\_stat\_mat[j,i] <- scan\_stat(months\_sim, i) |
|  | } |
|  | } |
|  |  |
|  |  |
|  | #### Compare what actually happened to the simulated data |
|  | p\_vals <- 0 |
|  | stat\_obs <- 0 |
|  | window <- 1:max\_window |
|  | for (k in window) { |
|  | stat\_obs[k] <- scan\_stat(mass\_shootings$months, k) |
|  | p\_vals[k] <- mean(max\_stat\_mat[, k] >= stat\_obs[k]) |
|  | } |
|  |  |
|  | # all unadjusted p values: |
|  | cbind(window, stat\_obs, p\_vals) %>% kable() %>% kable\_styling() %>% write\_clip() |
|  |  |
|  | # the lowest unadjusted p value: |
|  | unadj\_pval <- min(p\_vals) |
|  |  |
|  | # adjust that p value for all the data dredging we've done so far by comparing our result (lowest p value) |
|  | # to what we get by simulating everything from an actual Poisson process |
|  | M\_sim <- 10000 |
|  | pvals\_sim <- 0 |
|  | min\_pval\_sim <- 0 |
|  | for (a in 1:M\_sim){ |
|  | N <- rpois(1, n\_massacres) |
|  | months\_sim <- sort(sample(1:210, size = N,repl = FALSE)) |
|  | stat\_obs\_sim <- 0 |
|  | for (b in window){ |
|  | stat\_obs\_sim[b] <- scan\_stat(months\_sim,b) |
|  | pvals\_sim[b] <- mean(max\_stat\_mat[,b] >= stat\_obs\_sim[b]) |
|  | } |
|  | min\_pval\_sim[a] <- min(pvals\_sim) |
|  | } |
|  |  |
|  | # On average, how often is the minimum p value we just simulated less than the unadjusted p value |
|  | # we got from the comparison of data to simulations? |
|  | adj\_pval <- mean(min\_pval\_sim <= unadj\_pval) |
|  | adj\_pval |
|  |  |
|  | #-------------Addendum - including the Osmington solution--------- |
|  | mass\_shootings2 <- data.frame( |
|  | mon = c(9, 1, 6, 8, 10, 12, 9, 8, 8, 10, 3, 1, 4,5), |
|  | yr = c(1981, 1984, 1987, 1987, 1987, 1987, 1988, 1990, 1991, 1992, 1993, 1996, 1996, 2018) |
|  | ) |
|  |  |
|  | mass\_shootings2 <- mass\_shootings2 %>% |
|  | mutate(months = 12 \* (yr - 1979) + mon, |
|  | approx\_date = as.Date(paste(yr, mon, 15, sep = "-")), |
|  | interval = c(NA, diff(approx\_date))) |
|  |  |
|  | p3 <- ggplot(mass\_shootings2, aes(xend = approx\_date, x = approx\_date)) + |
|  | geom\_rect(xmin = as.Date("1996-07-15"), xmax = Inf, ymin = -Inf, ymax = Inf, |
|  | fill = "steelblue", alpha = 0.01) + |
|  | geom\_rect(xmax = as.Date("1996-07-15"), xmin = -Inf, ymin = -Inf, ymax = Inf, |
|  | fill = "red", alpha = 0.01) + |
|  | geom\_segment(y = -Inf, yend = Inf) + |
|  | scale\_x\_date(limits = c(min(mass\_shootings$approx\_date - 50), as.Date("2019-09-15"))) + |
|  | theme(panel.grid = element\_blank(), |
|  | panel.border = element\_blank(), |
|  | axis.text.y = element\_blank(), |
|  | axis.ticks.y = element\_blank()) + |
|  | labs(x = "Approximate date of mass shooting", |
|  | y = "") + |
|  | annotate("text", x = as.Date("2008-01-01"), y = 1, label = "After the gun buy-back", |
|  | colour = "steelblue") + |
|  | ggtitle("Firearm-related homicides in Australia, to September 2019", |
|  | "Events in which at least 5 persons other than perpetrator died") |