Baltimore Ceasefire 365

Estimated impact of a recurring community-led ceasefire on gun violence

Script by Peter L. Phalen

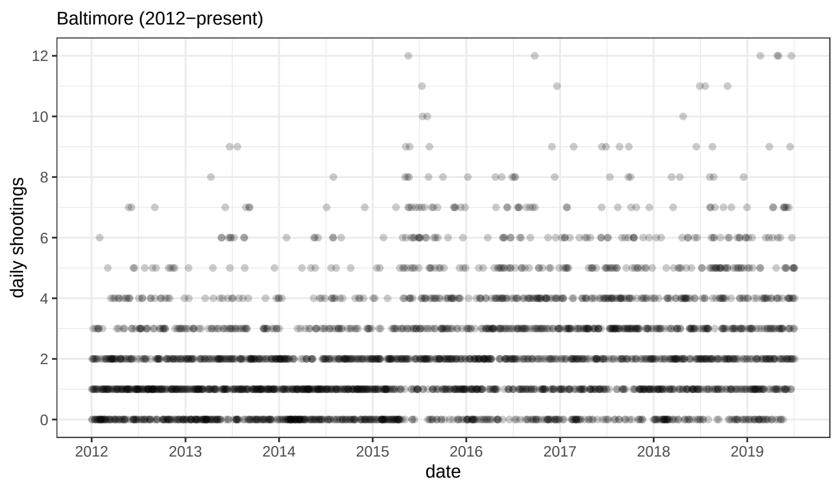
This markdown gives a complete annotated R script for the associated paper: "Baltimore Ceasefire 365: Estimated impact of a recurring community-led ceasefire on gun violence."

Shootings in Baltimore

Baltimore releases detailed data on issues relevant to the city, including victim-based crime. We begin by downloading this data, calculating daily number of shootings, and graphing the raw numbers for each day from January 1, 2012 to July 6, 2019.

```
library(knitr)
library(tidyverse)
library(scales)

setwd("/Users/peterphalen/Documents/ceasefire/")
bpd <- read_csv("BPD_Part_1_Victim_Based_Crime_Data.csv")

# subset to shootings or homicides with a firearm
bpd <- subset(bpd, Description == "SHOOTING" |
                (Description == "HOMICIDE" & Weapon == "FIREARM"))

bpd$CrimeDate <- as.Date(bpd$CrimeDate, format = "%m/%d/%y")

# there are many crimes per day. collapse to daily counts
daily <- bpd %>% group_by(CrimeDate) %>% summarise(shootings = n())

# fill missing dates, because some had no shootings
full.ts <- data.frame(CrimeDate = seq(daily$CrimeDate[1],
                                      daily$CrimeDate[nrow(daily)], by="day"))
daily <- full_join(full.ts,daily)
daily <- daily %>% group_by(CrimeDate) %>% mutate_all(funs(ifelse(is.na(.),0,.)))

ggplot(daily) +
  aes(x=CrimeDate, y=shootings) +
  geom_point(alpha=.2) +
  scale_x_date(breaks="year", date_labels="%Y") +
  scale_y_continuous(breaks=seq(0,20,2)) +
  xlab("date") +
```

```
  ylab("daily shootings") +
  ggtitle(" ",
          subtitle="Baltimore (2012-present)") +
  theme_bw()
```



Ceasefire weekends

Baltimore Ceasefire 365 has been organizing ceasefires four times per year since August 2017. These always occur on the first weekends (Friday through Sunday) of February, August, and November, as well as Mother's Day weekend in May. This code gets us a list of ceasefire dates.

```
# first days (Fridays) of each ceasefire weekend
ceasefire.fridays <-
  as.Date(
  c("08/04/2017",
    "11/03/2017",
    "02/02/2018",
    "05/11/2018",
    "08/03/2018",
    "11/02/2018",
    "02/01/2019",
    "05/10/2019",
    "08/02/2019"),
      format="%m/%d/%Y")

ceasefire.weekends <-
  lapply(ceasefire.fridays,
         function(x){
```

```
          seq(from=x,
              by="day",
              length.out=3)
          }
        )

ceasefire.weekends <- do.call("c",
                              ceasefire.weekends)
```

Statistical model

We want to include information about the date, the day of the week (Mon-Sun), the day of the year (to measure the effects of seasons and "special days" like Christmas), and a binary variable indicating whether a date occurs during ceasefire. We create those variables here:

```
library(lubridate)

# the julian calendar is a simple system for numeric dates
daily$jul <- julian(daily$CrimeDate)

daily$weekday <- factor(weekdays(daily$CrimeDate),
                        levels=c("Monday","Tuesday","Wednesday","Thursday",
                                 "Friday","Saturday","Sunday"))

daily$seasonal <- yday(daily$CrimeDate)

daily$calendar.day <- scales::date_format("%b-%d")(daily$CrimeDate)

daily$ceasefire <- factor(ifelse(daily$CrimeDate %in% ceasefire.weekends, 1, 0),
                          labels=c("Regular Day","Ceasefire Weekend"))
```

We also need to identify Mother's Day, because it floats (second Sunday in May) but always coincides with a ceasefire. This code identifies each Mother's Day in the dataset:

```
d <- daily$CrimeDate
sundays.in.may <- d[ format(d, '%a') == 'Sun' & format(d, '%B') == 'May' ]
sundays.in.may <- data.frame(sundays = sundays.in.may, year = format(sundays.in.may, '%Y'))
mothers.days <-
  sundays.in.may %>%
  group_by(year) %>%
  mutate(week2 = sundays[2]) %>%
  distinct(week2) %>%
  pull(week2)
```

```r
daily$mothers.day <- factor(ifelse(daily$CrimeDate %in% mothers.days, 1, 0))
```

You can see that there were a lot of shootings on Mother's Days in the years prior to Ceasefire, but since the introduction of Ceasefire there have been very few.

```r
kable(daily %>%
        subset(mothers.day == 1) %>%
        select(date=CrimeDate,weekday,ceasefire,shootings),
        align="c")
```

date	weekday	ceasefire	shootings
2012-05-13	Sunday	Regular Day	4
2013-05-12	Sunday	Regular Day	2
2014-05-11	Sunday	Regular Day	2
2015-05-10	Sunday	Regular Day	9
2016-05-08	Sunday	Regular Day	2
2017-05-14	Sunday	Regular Day	2
2018-05-13	Sunday	Ceasefire Weekend	0
2019-05-12	Sunday	Ceasefire Weekend	1

We're going to predict shootings using the following covariates:

- Overall time trend
  - We use a spline to estimate the overall time trend, which allows for "curvy" relationships while avoiding overfitting
- Yearly seasonality
  - We account for yearly seasonality using a spline with a cyclical constraint to ensure that the seasonal effect begins where it ended
- Day of the week
  - We allow intercepts to vary for days of the week in case weekends have different patterns of shootings than weekdays
- Calendar day
  - We allow varying intercepts for each day of the year in case "special days", like Christmas, show different patterns of shootings
- Mother's Day
  - We use a binary indicator for Mother's Day, which is the second Sunday in May and always coincides with ceasefire

- Ceasefire effect
  - We use a binary indicator for days occurring during ceasefire weekends to estimate the impact of ceasefire after accounting for all of the above

We fit the model using the `rstanarm` package below.

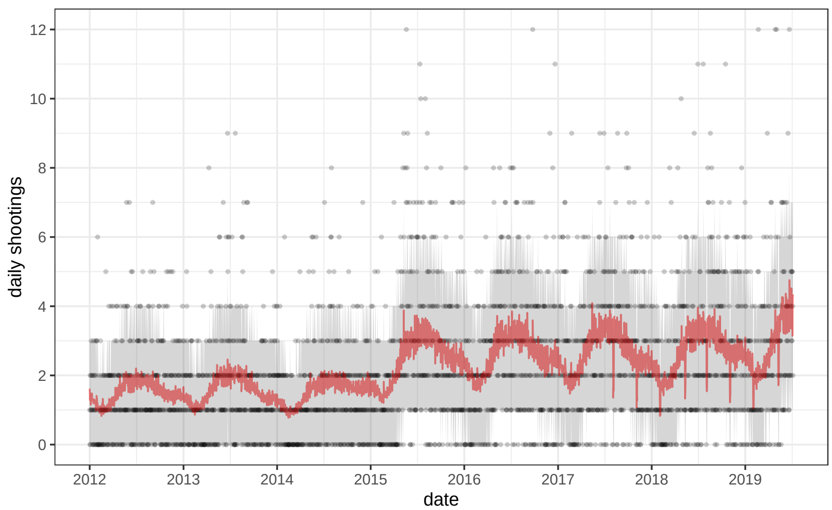
We use Poisson regression because the outcome is a count and follows a Poisson distribution fairly well. In fact there is evidence of overdispersion and posterior predictive checks confirm that a negative binomial model fits the data better, but parameter estimates of interest are basically unaffected (the most significant difference is that the `calendar.day` effect becomes even weaker) so we stick with the Poisson model for simplicity. Interested researchers can refit the model to a negative binomial distribution by replacing "poisson" in the `stan_gamm4` function with "neg_binomial_2". The rest of the code can then be run without further adjustments.

```
library(rstanarm)

model <- stan_gamm4(shootings ~
            s(jul) + # spline time trend
            s(seasonal, # seasonal effect
             bs="cc") + #cyclical constraint
            mothers.day + # mothers day indicator
            ceasefire, # ceasefire indicator
           random= ~ (1 | weekday) + # day of the week
                     (1 | calendar.day), # every day....
           data=daily,
           cores=4,
           iter=2000,
           family=poisson)
```

The following code creates a plot of the model against the observations. The red line represents the point prediction for the model. The grey area is the 80% predictive interval, which means that about 80% of days will have shooting counts that fall within the grey area. 10% of the time, the number of shootings will extend *above* the grey area.

```
daily$Estimate <- apply(posterior_linpred(model, transform=TRUE),
                        2, mean)

# 80% posterior predictive interval for main plot
preds <- posterior_predict(model, transform=TRUE)
preds <- apply(preds, 2, function(x){quantile(x, prob=c(.1, .9))})
```

```
daily$high <- preds["90%",]
daily$low <- preds["10%",]

daily %>%
  ggplot(aes(x = CrimeDate, y = shootings)) +
  geom_point(alpha=.2, size=.75) +
  geom_line(aes(y = Estimate), alpha=.5, color="red") +
  geom_ribbon(aes(ymin=low, ymax=high), alpha=.2) +
  scale_x_date(breaks="year", date_labels="%Y") +
  scale_y_continuous(breaks=seq(0,20,2)) +
  xlab("date") +
  ylab("daily shootings") +
  theme_bw()
```



Ceasefires are visible to the naked eye as eight dramatic downward red spikes beginning in 2017.

Model components We can plot the marginal effects to show the components that make up the above time series. These figures give you the right idea of the shape of the trends.

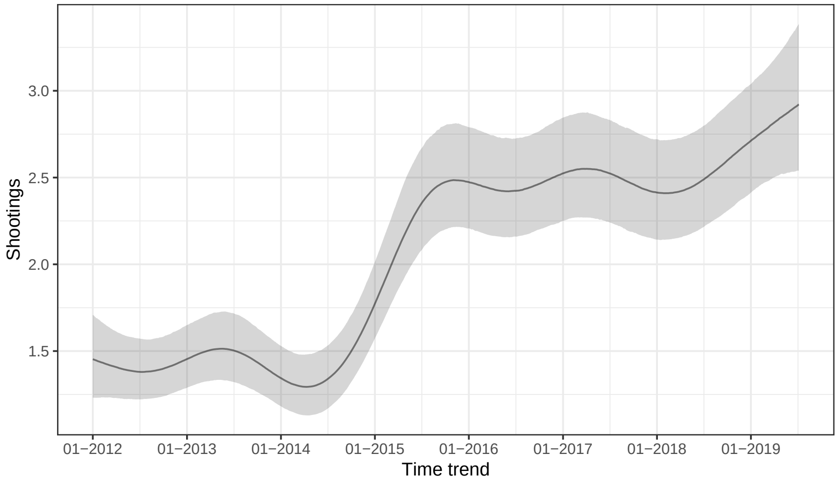
We begin by creating a plot of the overall time trend, with seasonality removed.

```
### Time trend plot
time.frame <- with(daily, # Ref: mid-April
                  data.frame(
                    jul=jul,
                    weekday=0, # not used for this prediction
                    ceasefire="Regular Day",
                    calendar.day= 0,
```

```
                    mothers.day = 0,
                    seasonal=yday(as.Date("2018-04-15"))))

post <- posterior_linpred(model,
                          newdata=time.frame,
                          transform=TRUE,
                          re.form = NA)
time.frame$Estimate <- apply(post,2, median)


# 95% CI
ci <- apply(post,2,function(x){quantile(x, prob=c(.025, .975))})
time.frame$low <- ci["2.5%",]
time.frame$high <- ci["97.5%",]

trend.axis.dates <- seq(from=as.Date("2012-01-01"),
                        by="year",
                        length.out=9)
time.plot <-
  time.frame %>%
  ggplot() +
  aes(x=jul, y=Estimate) +
  geom_line(aes(y = Estimate), alpha=.5) +
  geom_ribbon(aes(ymin=low, ymax=high), alpha=.2) +
  xlab("Time trend") +
  ylab("Shootings") +
  ggtitle(" ") +
  scale_x_continuous(
    breaks=julian(trend.axis.dates),
    labels=date_format("%m-%Y")(trend.axis.dates)) +
  theme_bw()
```

Here's the deseasonalized time trend we just created in the above block of code. Shootings increased in 2015 and have stayed high.

```
time.plot
```



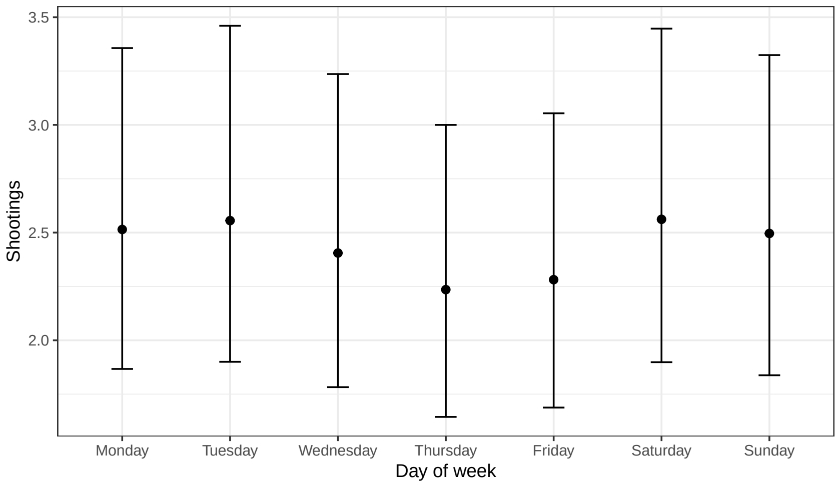
Now we're going to construct a plot for weekdays.

```
wday.frame <- with(daily, # Ref: regular day in mid-April
                  data.frame(
                    jul=julian(as.Date("2018-04-15"))[1],
                    weekday=unique(daily$weekday),
                    ceasefire="Regular Day",
                    mothers.day = 0,
                    calendar.day = 0,
                    seasonal=yday(as.Date("2018-04-15"))))

post <- posterior_linpred(model,
                          newdata=wday.frame,
                          transform=TRUE)
wday.frame$Estimate <- apply(post,2, median)

# 95% CI
ci <- apply(post,2,function(x){quantile(x, prob=c(.025, .975))})
wday.frame$low <- ci["2.5%",]
wday.frame$high <- ci["97.5%",]

wday.plot <-
  wday.frame %>%
  ggplot() +
  aes(x=weekday, y=Estimate) +
  geom_point(size=2) +
  geom_errorbar(aes(ymin=low, ymax=high),
                width=.2) +
  xlab("Day of week") +
  ylab("Shootings") +
  ggtitle(" ") +
```

```
  theme_bw()
```

Here is the weekday plot we just created, which shows essentially no weekday effect:

```
wday.plot
```



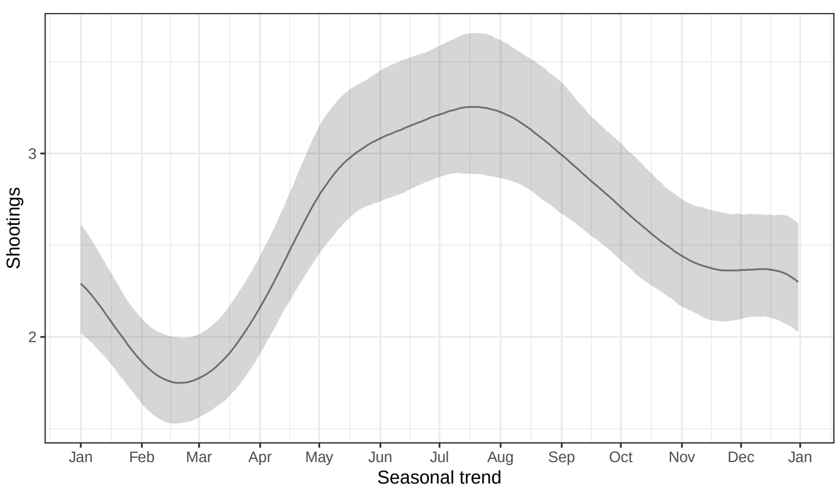
Next we construct a plot for the yearly seasonal effect.

```
seasonal.frame <- with(daily, # Ref: regular day in mid-2018
                data.frame(
                  jul=julian(as.Date("2018-04-15"))[1],
                  weekday=0, # weekday not used for this prediction
                  ceasefire="Regular Day",
                  mothers.day = 0,
                  calendar.day= 0,
                  seasonal=1:365))

post <- posterior_linpred(model,
                           newdata=seasonal.frame,
                           transform=TRUE,
                          re.form = NA)
seasonal.frame$Estimate <- apply(post,2, median)

# 95% CI
ci <- apply(post,2,function(x){quantile(x, prob=c(.025, .975))})
seasonal.frame$low <- ci["2.5%",]
seasonal.frame$high <- ci["97.5%",]

cal.day.axis.dates <- seq(as.Date("0-01-01"),by="month",length.out=12)
```

```
seasonal.plot <-
  seasonal.frame %>%
  ggplot() +
  aes(x=seasonal, y=Estimate) +
  geom_line(aes(y = Estimate), alpha=.5) +
  geom_ribbon(aes(ymin=low, ymax=high), alpha=.2) +
  xlab("Seasonal trend") +
  ylab("Shootings") +
  ggtitle(" ") +
  scale_x_continuous(
    breaks=yday(c(cal.day.axis.dates, as.Date("0-12-31"))),
    labels=date_format("%b")(c(cal.day.axis.dates, as.Date("0-01-01")))
  ) +
  scale_y_continuous(breaks=0:5) +
  theme_bw()
```

Looking at the yearly seasonal component of the model with 95% credible intervals, we do see a strong seasonal effect. Summers show up as particularly bad. We see the largest dips in shootings in February and March, which are cold and dark months in Baltimore.

```
seasonal.plot
```



Now we construct a plot showing the relative incidence of shootings on each calendar day of the year.

```
cal.day.for.single.year <- date_format("%b-%d")(seq(as.Date("0000-01-01"),
                                                    as.Date("0000-12-31"),
                                                    by = 1)) # this year includes a leap day


cal.day.frame <- data.frame(calendar.day=cal.day.for.single.year,
```

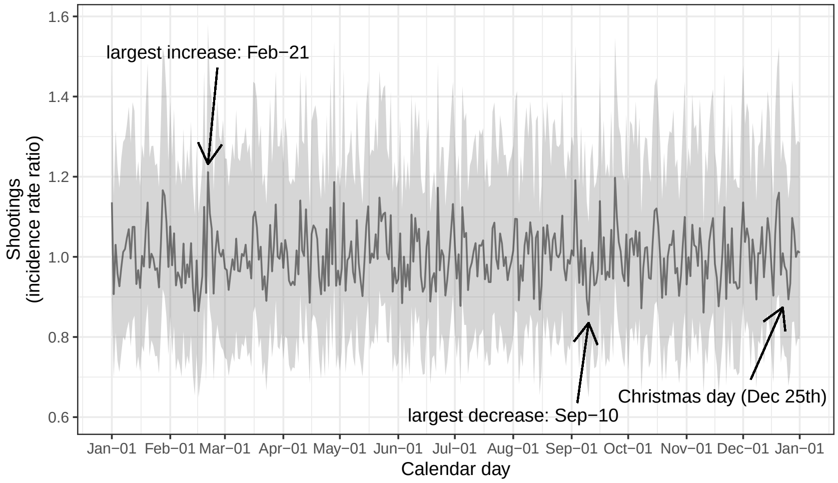
```
                              cal.day.index = 1:length(cal.day.for.single.year))

cal.day.samples <- as.matrix(model, regex_pars = " calendar.day")
# scale to an IRR by exponentiating
cal.day.samples <- exp(cal.day.samples)
# give coefficients more readable names
cal.alpha <- sort(cal.day.for.single.year)
colnames(cal.day.samples) <- cal.alpha
# order by calendar year (whereas stan alphabetizes)
cal.day.samples <-
  cal.day.samples[,order(match(colnames(cal.day.samples),cal.day.for.single.year))]

cal.day.effects <- apply(cal.day.samples, 2, median)
cal.day.frame$Estimate <- as.vector(cal.day.effects)

# 95% credible interval
ci <- apply(cal.day.samples, 2, function(x){quantile(x, prob=c(.025, .975))})
cal.day.frame$low <- ci["2.5%",]
cal.day.frame$high <- ci["97.5%",]

cal.day.axis.labels <- seq(as.Date("0-01-01"),by="month",length.out=12)

highest.day <- cal.day.frame %>% slice(which.max(Estimate))
lowest.day <- cal.day.frame %>% slice(which.min(Estimate))
xmas.day <- cal.day.frame %>% slice(which(calendar.day=="Dec-25"))

cal.day.plot <-
  cal.day.frame %>%
  ggplot() +
  aes(x=cal.day.index, y=Estimate) +
  geom_line(aes(y = Estimate), alpha=.5) +
  geom_ribbon(aes(ymin=low, ymax=high), alpha=.2) +
  xlab("Calendar day") +
  ylab("Shootings\n(incidence rate ratio)") +
  ggtitle(" ") +
  scale_x_continuous(
    breaks=yday(c(cal.day.axis.labels, as.Date("0-12-31"))),
    labels=date_format("%b-%d")(c(cal.day.axis.labels, as.Date("0-01-01")))
  ) +
  annotate("text",   #dynamically annotate highest day
           x = highest.day$cal.day.index,
           y = highest.day$Estimate + .3,
           label = paste("largest increase:",highest.day$calendar.day)) +
  geom_segment(aes(
```

```
            xend = highest.day$cal.day.index,
            x = highest.day$cal.day.index + 5,
            yend = highest.day$Estimate + .02,
            y = highest.day$Estimate + .26),
            arrow = arrow(length = unit(0.5, "cm"))) +
  # lowest day
  annotate("text", x = lowest.day$cal.day.index - 40,
        y = lowest.day$Estimate - .25,
        label = paste("largest decrease:",lowest.day$calendar.day)) +
  geom_segment(aes(
            xend = lowest.day$cal.day.index,
            x = lowest.day$cal.day.index - 6,
            yend = lowest.day$Estimate - .02,
            y = lowest.day$Estimate - .22),
            arrow = arrow(length = unit(0.5, "cm"))) +
  # christmas day
  annotate("text", x = xmas.day$cal.day.index - 35,
        y = xmas.day$Estimate - .24,
        label = "Christmas day (Dec 25th)") +
  geom_segment(aes(
            xend = xmas.day$cal.day.index - 3,
            x = xmas.day$cal.day.index - 20,
            yend = xmas.day$Estimate - .02,
            y = xmas.day$Estimate - .2),
            arrow = arrow(length = unit(0.5, "cm"))) +
  theme_bw()
```

I've programmatically marked the highest and lowest days (after accounting for slower-moving seasonal trends) on the above figure. I've also marked Christmas day, because it was interesting to me.

We don't see a lot of evidence for calendar day effects: the 95% credible intervals for all these days overlap. So we could make an argument for taking it out of the model, but including it lends credence to the idea that the ceasefire effect is special.

```
cal.day.plot
```



We can look at the 5 days with the lowest numbers of shootings (relative to their placement in the season).

```
day.effects.sorted <- cal.day.frame[order(cal.day.frame$Estimate),
                                     "calendar.day"]
kable(data.frame(five_lowest_days=head(day.effects.sorted,5)))
```

five_lowest_days
Sep-10
Nov-10
Feb-16
Feb-14
Aug-15

And the 5 days with highest numbers of shootings (relative to their placement in the season).

```
kable(data.frame(five_highest_days=rev(tail(day.effects.sorted,5))))
```

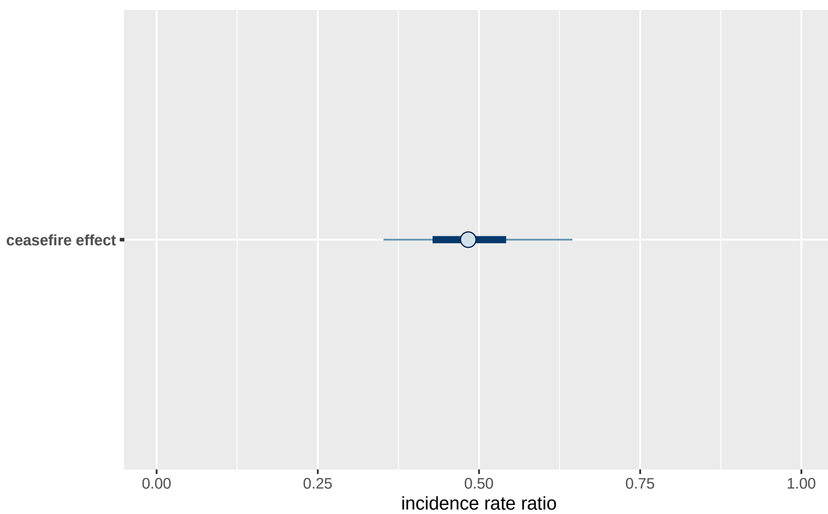
five_highest_days
Feb-21
Sep-24
Sep-03
Apr-28
Jun-22

None of the these days are holidays or otherwise notable. It's likely these calendar day effects are simply statistical noise…

Effect of Ceasefire

Finally, we can use this model to measure the effect of Ceasefires on shootings per day, after accounting for all these trends and seasonalities. The effect of the Ceasefire (plotted here as an incidence rate ratio with 95% credible intervals) is classically statistically significant.

```
ceasefire.effect <- as.array(model, regex_pars = "ceasefire")
# scale to an IRR by exponentiating
ceasefire.effect <- exp(ceasefire.effect)
library(bayesplot)
mcmc_intervals(ceasefire.effect) +
  scale_y_discrete(labels="ceasefire effect") +
  xlab("incidence rate ratio") +
  xlim(c(0,1))
```



We're looking at an approximate 50% reduction in shootings during ceasefire weekends, after accounting for all other effects (e.g., day of the week, holidays, etc).

```
round(1-median(ceasefire.effect),2)
```

```
## [1] 0.52
```

There's enough uncertainty in our estimates that the true effect of ceasefire could be anywhere from a one third to a two-thirds reduction in shootings.

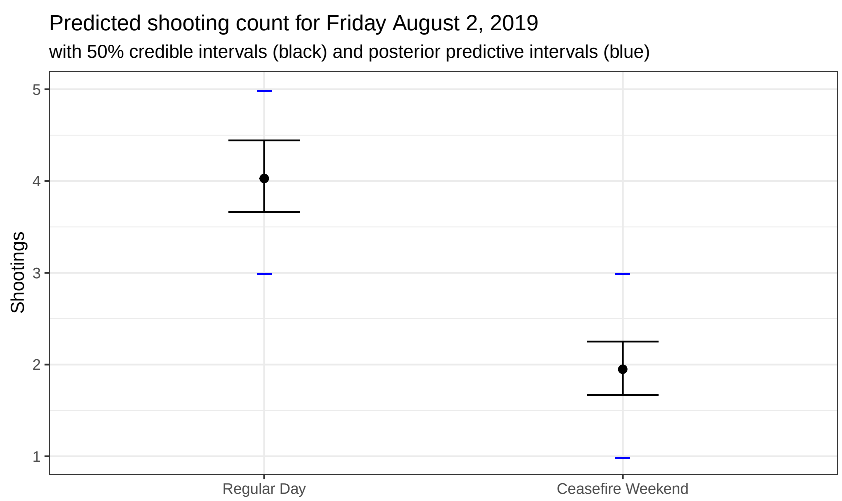
```
# 95% credible intervals
round(1-quantile(ceasefire.effect, probs=c(.975, .025)),2)
```

```
## 97.5%  2.5%
##  0.33  0.67
```

We can also use this model to see the impact of the ceasefire at specific points in time. For example, this block of code lets us see the model-estimated impact of an upcoming ceasefire on Friday August 2nd, 2019 (out-of-sample).

```
### Ceasefire plot
pred.day <- as.Date("2019-08-02")
ceasefire.frame <- with(daily,
                  data.frame(
                    jul=julian(pred.day)[1],
                    weekday="Friday",
                    mothers.day = 0,
                    ceasefire=factor(c("Regular Day",
                                "Ceasefire Weekend"),
                                levels=c("Regular Day",
                                         "Ceasefire Weekend")),
                    calendar.day=date_format("%b-%d")(pred.day),
                    seasonal= yday(pred.day)))

post <- posterior_linpred(model,
                          newdata=ceasefire.frame,
                          transform=TRUE)
ceasefire.frame$Estimate <- apply(post,2, median)

# 50% CI
ci <- apply(post,2,function(x){quantile(x, prob=c(.25, .75))})
ceasefire.frame$low <- ci["25%",]
ceasefire.frame$high <- ci["75%",]

# 50% posterior predictive interval for main plot
preds <- posterior_predict(model,
                          newdata=ceasefire.frame,
                          transform=TRUE)

ceasefire.frame$high.ppd <- apply(preds,2,function(x){quantile(x, prob=c(.75), na.rm=T)})
ceasefire.frame$low.ppd <- apply(preds,2,function(x){quantile(x, prob=c(.25), na.rm=T)})

ceasefire.frame %>%
  ggplot() +
  aes(x=ceasefire, y=Estimate) +
  geom_point(aes(y = low.ppd), col="blue", shape=95, size=5) +
  geom_point(aes(y = high.ppd), col="blue", shape=95, size=5) +
  geom_point(aes(y = Estimate),
             size=2) +
```

```
geom_errorbar(aes(ymin=low, ymax=high),
              width=.2) +

xlab("") +
ylab("Shootings") +
ggtitle("Predicted shooting count for Friday August 2, 2019",
        subtitle="with 50% credible intervals (black) and posterior predictive intervals (blue)") +
theme_bw()
```



Without a ceasefire, we'd expect about four people to get shot on this day. But because this will be a Ceasefire weekend, the model expects about half that many to be shot.

Notice the little blue horizontal lines drawn at 1 and 3 for the ceasefire weekend estimate. Those marks are the 50% posterior predictive intervals, representing the 50% window for the number of shootings you can expect on any given day. The fact that the tick marks rest at 1 and 3 means that there is about a 25% chance that we will see zero shootings on this day, and about a 25% chance that we will see >3 shootings on this day.

The ceasefire impact is real and at the same time, our model won't be surprised by the occurrence of several shootings on each day of ceasefire weekend.

Is it possible that violence is simply postponed?

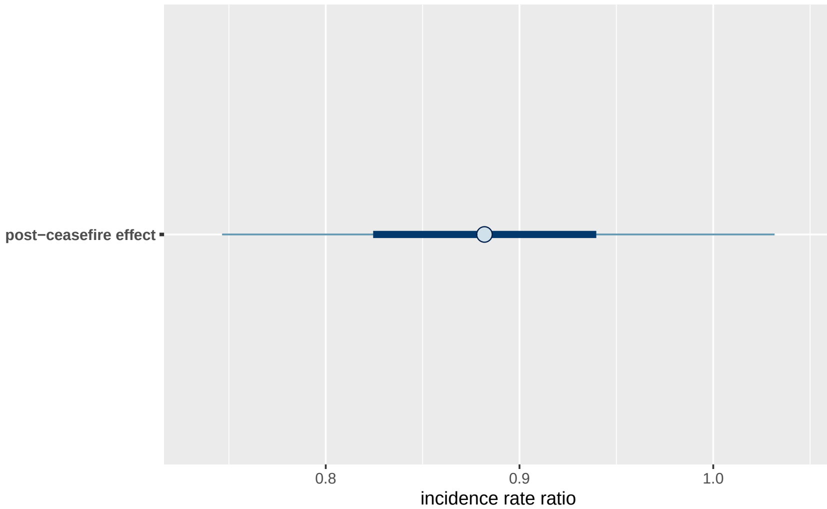
Some people have suggested the possibility that people who decide not to commit shootings during ceasefire may simply postpone that violence until after ceasefire is over. This is possible, but seems unlikely given that lapses in violence observed during Ceasefires have been observed to extend beyond them, with one Ceasefire marking the beginning of a 12-day period of zero shooting deaths.

We can test the hypothesis of postponed violence by looking at the pattern of shootings observed in the three days following each three-day ceasefire weekend, and also the three-day weekend following each ceasefire weekend. We start by creating a dummy variable for these three-day periods.

```
# Get the three days following ceasefire (by incrementing the ceasefire friday dates by 3)
ceasefire.subsequent.mondays <- ceasefire.fridays + 3

# Get the weekend following ceasefire (by incrementing the ceasefire friday dates by 7)
ceasefire.subsequent.fridays <- ceasefire.fridays + 7

three.day <-
  lapply(c(ceasefire.subsequent.mondays, ceasefire.subsequent.fridays),
         function(x){
           seq(from=x,
               by="day",
               length.out=3)
           })

following.ceasefire <- do.call("c",
                               three.day)

# mark days according to whether or not they're in the three days following ceasefire
daily$postceasefire <- ifelse(daily$CrimeDate %in% following.ceasefire, 1, 0)
```

We can now include this dummy variable in our larger model to test whether the three days following ceasefires show increased gun violence.

```
model.two <- update(model, . ~ . + postceasefire, iter=1000) # three days post ceasefire indicator

post.ceasefire.effect <- as.array(model.two, regex_pars = "postceasefire")
# scale to an IRR by exponentiating
post.ceasefire.effect <- exp(post.ceasefire.effect)
mcmc_intervals(post.ceasefire.effect) +
  scale_y_discrete(labels="post-ceasefire effect") +
  xlab("incidence rate ratio")
```




```
round(quantile(post.ceasefire.effect, probs=c(.025,.5,.975)), 2)
```

```
##  2.5%   50% 97.5%
##  0.72  0.88  1.06
```

The model does **not** suggest a higher number of shootings in the days following ceasefires. If anything there is a continued decrease (IRR=0.9, 95%CI: 0.7-1.1).