Computing Expected Violence Exposure

Jonathan Che

10 April 2017

Overview

In Dahl and DellaVigna's paper "Does Movie Violence Increase Violent Crime?", they find that the "incapacitation effect" outweighs the "arousal effect" (details in paper summary). In their analyses, they use kids-in-mind.com's violence ratings to measure public exposure to movie violence.

To get a better understanding of the differen effects of incapacitation and arousal, I build a model to predict "expected" exposure to violence, as opposed to kids-in-mind.com's "actual" exposure to violence scores. The variation between "expected" and "actual" violence will drive my regression analyses.

In this document, I will first perform a proxy replication of how Dahl and DellaVigna calculate public exposure to movie violence to see how my methods compare to theirs. I do this for a few reasons. First, I cannot directly view the methods that Dahl and DellaVigna use to calculate public exposure to movie violence. Though I have the final computed values, I don't have the specifics of how they were calculated. Thus, before I calculate my "expected exposure" scores, I want to check that my methods are at least similar to theirs. Second, I will not (at least for now) be imputing any box office sales like Dahl and DellaVigna do (Appendix I). Thus, my results will definitely be different from theirs. I want to check that the magnitude of this difference is not too extreme before proceeding. To check for similarity, I will just examine a scatterplot (and the R^2).

Then, I will use a regression methodology to compute "expected exposure" scores. I will base these predictions on the movie's MPAA rating, and its genre.

Method

First, I pull in all the data that I will need.

```
movie_ratings <- readRDS("movie_ratings.rds")
movie_sales <- readRDS("movie_sales.rds")
movie_genres <- readRDS("movie_genres.rds")
ticket_prices <- read_csv("ticket_prices.csv")
```

"We deflate... the daily box-office sales by the average price of a ticket" (Dahl and DellaVigna 690)

```
movie_sales <- movie_sales %>%
  left_join(ticket_prices, by="Year") %>%
  mutate(Tickets = Gross/Price)
```

```
## Warning in FUN(X[[i]], ...): failed to assign NativeSymbolInfo for lhs
## since lhs is already defined in the 'lazyeval' namespace
```

```
## Warning in FUN(X[[i]], ...): failed to assign NativeSymbolInfo for rhs
## since rhs is already defined in the 'lazyeval' namespace
```

"We match the box-office data to violence ratings from kids-in-mind.com... we group movies into three categories: strongly violent, mildly violent, and nonviolent" (690). 0-4 is nonviolent, 5-7 is mildly violent, 8-10 is violent.

```
# Some data cleaning first
movie_ratings <- movie_ratings %>%
```

```
  select(Title, Year, MPAA_Rating, Violence) %>%
  mutate(Year = as.numeric(as.character(Year))) %>%
  mutate(Violence = as.numeric(as.character(Violence))) %>%
  mutate(MPAA_Rating = as.character(MPAA_Rating)) %>%
  mutate(MPAA_Rating = str_replace_all(MPAA_Rating, "[-\\[\\]]", ""))

movie_ratings <- movie_ratings %>%
  mutate(viol_strong = Violence >= 8) %>%
  mutate(viol_mild = (Violence >=5 & Violence <= 7)) %>%
  mutate(viol_non = Violence <= 4)
```

Now, we merge the two data frames by movie title.

```
# Kids in mind parses "The ___" movies as "___, The" (same as "A ___")
movie_ratings <- movie_ratings %>%
  mutate(Title = ifelse(str_detect(Title, ", The"),
                        str_c("The ", str_replace(Title, ", The", "")),
                        as.character(Title))) %>%
  mutate(Title = ifelse(str_detect(Title, ", A"),
                        str_c("A ", str_replace(Title, ", A", "")),
                        as.character(Title)))
# Manual method of matching some more difficult names
movie_ratings <- movie_ratings %>%
  mutate(Title = ifelse(Title == "Dr. Dolittle", "Doctor Dolittle",
                 ifelse(Title == "Star Wars: Episode II - Attack of the Clones", "Star Wars Ep. II: Atta
                 ifelse(Title == "Star Wars Episode I: The Phantom Menace", "Star Wars Ep. I: The Phanto
                 ifelse(Title == "The Lord of the Rings: Return of the King", "The Lord of the Rings: Th
                 ifelse(Title == "Jurassic Park III", "Jurassic Park 3",
                 ifelse(Title == "Men In Black II", "Men in Black 2",
                 ifelse(Title == "X2: X-Men United", "X2", Title)))))))
# The-numbers.com parsed this title strangely
movie_sales <- movie_sales %>%
  mutate(Title = ifelse(str_detect(Title, "Harry Potter and the Sorcerer"),
                        "Harry Potter and the Sorcerer's Stone",
                        as.character(Title))) %>%
  filter(Year <= 2005)   # To avoid rescreening sales
movie_genres <- movie_genres %>%
  mutate(Title = ifelse(str_detect(Title, "Harry Potter and the Sorcerer"),
                        "Harry Potter and the Sorcerer's Stone",
                        as.character(Title)))

# Unusual coding to deal with movies with same title in different years, or same movie spanning differe
# Assume that two movies with the same title don't come out in consecutive years
movie_ratings2 <- movie_ratings %>%
  mutate(Year = Year+1)
movie <- movie_sales %>%
  left_join(movie_genres, by="Title") %>%
  left_join(movie_ratings, by=c("Title", "Year")) %>%
  left_join(movie_ratings2, by=c("Title", "Year")) %>%
  mutate(MPAA_Rating = ifelse(is.na(MPAA_Rating.x), MPAA_Rating.y, MPAA_Rating.x)) %>%
  mutate(Violence = ifelse(is.na(Violence.x), Violence.y, Violence.x)) %>%
  mutate(viol_strong = ifelse(is.na(viol_strong.x), viol_strong.y, viol_strong.x)) %>%
  mutate(viol_mild = ifelse(is.na(viol_mild.x), viol_mild.y, viol_mild.x)) %>%
  mutate(viol_non = ifelse(is.na(viol_non.x), viol_non.y, viol_non.x)) %>%
```

```
  select(-MPAA_Rating.x, -Violence.x, -viol_strong.x, -viol_mild.x, -viol_non.x,
         -MPAA_Rating.y, -Violence.y, -viol_strong.y, -viol_mild.y, -viol_non.y)
movie$MPAA_Rating = as.factor(movie$MPAA_Rating)
movie$Genre = as.factor(movie$Genre)
```

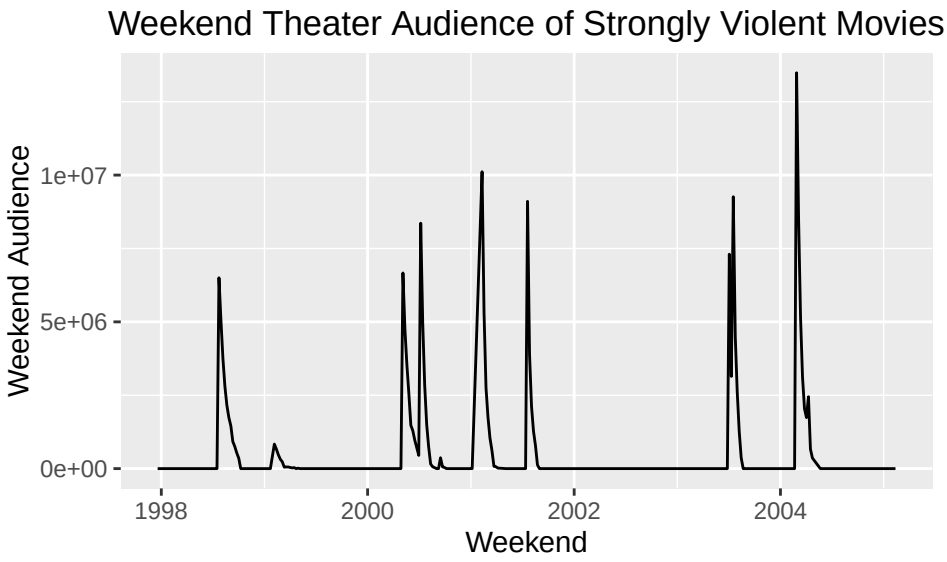
Finally, we can compute the daily exposure to movie violence.

```
daily_exposure <- movie %>%
  group_by(Date, Weekday) %>%
  summarise(tickets_tot = sum(Tickets),
            tickets_strong = sum(ifelse(viol_strong, Tickets, 0)),
            tickets_mild = sum(ifelse(viol_mild, Tickets, 0)),
            tickets_non = sum(ifelse(viol_non, Tickets, 0)))
```

Comparing to D&D

First, we construct a scatterplot similar to Dahl and DellaVigna's Figure 1a.

```
# Aggregate by weekend
weekend_exposure <- daily_exposure %>%
  ungroup() %>%
  filter(Weekday==1|Weekday==6|Weekday==7) %>%
  mutate(Date = ifelse(Weekday==7, Date-1,
                ifelse(Weekday==1, Date-2, Date))) %>%
  mutate(Date = as.Date(Date, origin="1970-01-01 UTC")) %>%
  group_by(Date) %>%
  summarise(tickets_tot = sum(tickets_tot),
            tickets_strong = sum(tickets_strong),
            tickets_mild = sum(tickets_mild),
            tickets_non = sum(tickets_non))

ggplot(weekend_exposure, aes(x=Date, y=tickets_strong)) +
  geom_line() +
  labs(title="Weekend Theater Audience of Strongly Violent Movies",
       y="Weekend Audience",
       x="Weekend")
```

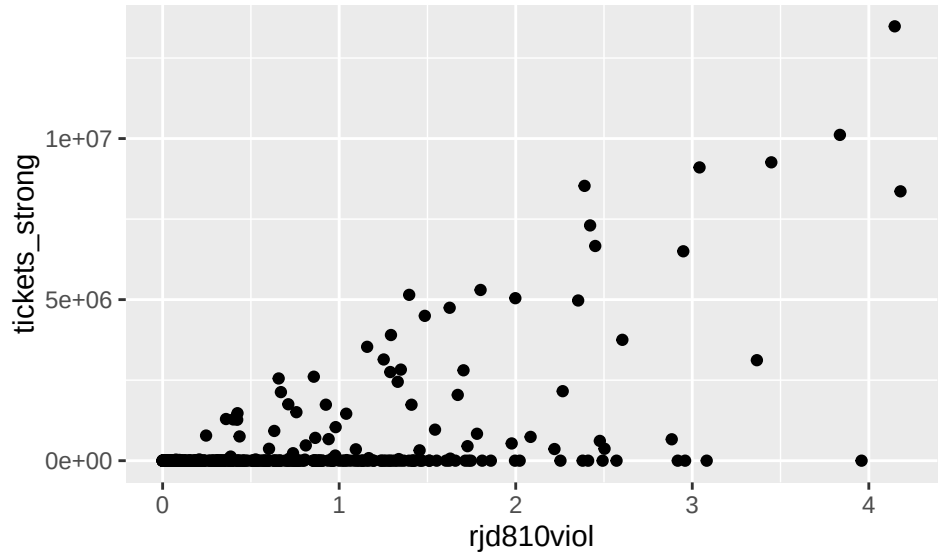


We note a few things. First, our exposure numbers are definitely sparser, which is expected, given that we only use top 10 movies. Second, our exposure numbers don't peak as highly (again expected).

Now, we pull in Dahl and DellaVigna's numbers to compare.

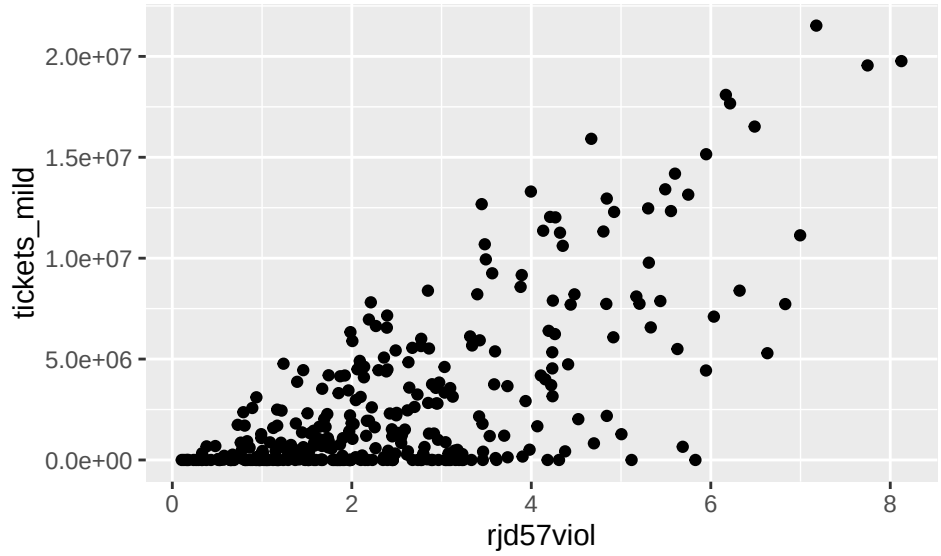
```
dd <- read_csv("fulliblockday.csv")
dd_ev <- dd %>%
  select(mdy, rjd04viol, rjd57viol, rjd810viol) %>%
  # Origin date computed from information in data
  mutate(Date = as.Date(mdy, origin="1960-01-01 UTC"))
compare_ev <- weekend_exposure %>%
  left_join(dd_ev, by="Date")
  # not sure how exactly D&D compute values
  # mutate(ln_viol = log(tickets_strong),
  #        ln_mild = log(tickets_mild),
  #        ln_non = log(tickets_non))

ggplot(compare_ev, aes(x=rjd810viol, y=tickets_strong)) +
  geom_point()
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```



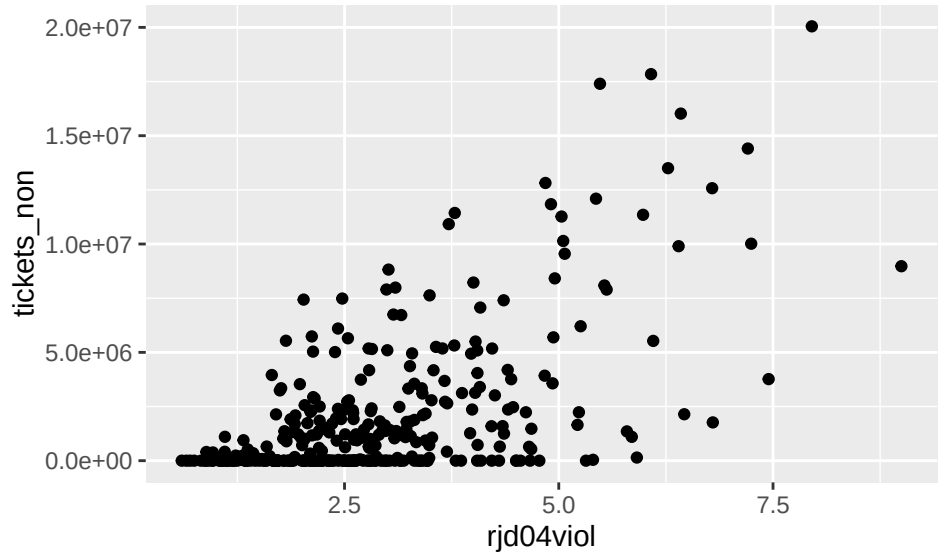

```
ggplot(compare_ev, aes(x=rjd57viol, y=tickets_mild)) +
  geom_point()
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```




```
ggplot(compare_ev, aes(x=rjd04viol, y=tickets_non)) +
  geom_point()
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```



Obviously, these aren't perfect relationships. In particular, we can see how my data underestimates exposure on many days.

Computing Expected Exposure to Violence

Though my data are not great, they do not display any glaring discrepancies with Dahl and DellaVigna's data either. As such, I proceed to calculate expected exposure to violence.

First, I run some summary statistics to explore the variation in my data.

```
##            MPAA_Rating
## viol_strong    G   PG PG13    R
##       TRUE     0    0   44  520
##       FALSE  372 1193 2547  422
```

```
##          MPAA_Rating
## viol_mild    G   PG PG13    R
##     TRUE     0  310 2049  148
##     FALSE  372  883  542  794
```

```
##         MPAA_Rating
## viol_non    G   PG PG13    R
##    TRUE   372  883  498  274
##    FALSE    0  310 2093  668
```

```
##            Genre
## viol_strong Action Adventure Comedy Drama Horror Musical Romantic Comedy
##       TRUE     222         0     66   222     54       0               0
##       FALSE    630      2118    562   143     97     115             312
##            Genre
## viol_strong Thriller/Suspense
##       TRUE                  0
##       FALSE               557
```

```
##          Genre
## viol_mild Action Adventure Comedy Drama Horror Musical Romantic Comedy
##     TRUE     630      1189     71    60      0       0               0
```

```
##     FALSE    222       929    557   305    151     115             312
##          Genre
## viol_mild Thriller/Suspense
##     TRUE                557
##     FALSE                 0
```

```
##         Genre
## viol_non Action Adventure Comedy Drama Horror Musical Romantic Comedy
##    TRUE       0       929    491    83     97     115             312
##    FALSE    852      1189    137   282     54       0               0
##         Genre
## viol_non Thriller/Suspense
##    TRUE                  0
##    FALSE               557
```

We see that there are definitely patterns in the data, but the relationships between MPAA Rating/Genre and Violence aren't perfect. Thus, we proceed to use regression to predict violence.

```
m1 <- lm(Violence ~ Genre+MPAA_Rating, data=movie)
movie <- movie %>%
  mutate(Exp_Violence = fitted(m1))
summary(m1)
```

```
##
## Call:
## lm(formula = Violence ~ Genre + MPAA_Rating, data = movie)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3543 -0.3984  0.0323  0.6237  4.4967
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.49418    0.09743  35.863  < 2e-16 ***
## GenreAdventure           -0.11783    0.06720  -1.753  0.07960 .
## GenreComedy              -2.12417    0.07445 -28.532  < 2e-16 ***
## GenreDrama               -0.63653    0.09101  -6.994 3.01e-12 ***
## GenreHorror              -2.48752    0.12791 -19.448  < 2e-16 ***
## GenreMusical             -2.08553    0.13788 -15.125  < 2e-16 ***
## GenreRomantic Comedy     -3.51497    0.09816 -35.809  < 2e-16 ***
## GenreThriller/Suspense   -0.23993    0.07897  -3.038  0.00239 **
## MPAA_RatingPG             1.02203    0.08278  12.347  < 2e-16 ***
## MPAA_RatingPG13           2.59136    0.08264  31.356  < 2e-16 ***
## MPAA_RatingR              4.49665    0.10598  42.428  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.361 on 5087 degrees of freedom
## Multiple R-squared:  0.5714, Adjusted R-squared:  0.5706
## F-statistic: 678.3 on 10 and 5087 DF,  p-value: < 2.2e-16
```

We see that Genre and MPAA Rating capture 57 percent of the variation in kids-in-mind violence. We plot the relationship visually.

```
ggplot(movie, aes(x=Violence, y=Exp_Violence)) +
  geom_point() +
  labs(title="Actual vs. Expected Violence in Movies",
```

```
       y="Expected Violence",
       x="Actual Violence") +
  geom_abline(slope=1, intercept=0)
```


```
cor(Violence~Exp_Violence, data=movie)
```

```
## [1] 0.7559395
```

We see that there is a decent amount of variation in the measures, which is good for our analyses. As a final note, we confirm that some examples of movies with large residuals are actually unexpectedly violent/nonviolent.

```
foo <- movie %>%
  mutate(Resid_Violence = Exp_Violence-Violence) %>%
  group_by(Title) %>%
  summarise(Resid_Violence=mean(Resid_Violence)) %>%
  arrange(Resid_Violence)
head(foo)
```

```
## # A tibble: 6 x 2
##                         Title Resid_Violence
##                         <chr>          <dbl>
## 1                   Hannibal      -4.496689
## 2 The Passion of the Christ      -2.645705
## 3                Scary Movie      -2.133339
## 4                  Gladiator      -2.009172
## 5            Jurassic Park 3      -1.914467
## 6       Saving Private Ryan      -1.645705
```

Looking at the top 6 movies that are "more violent than expected", we see that our measure performs decently well. Many of these films are in fact more violent than one would perhaps anticipate. We note that some movies, such as Hannibal and Gladiator, should be expected to be violent, and thus should not really be on this list. More controls may produce a better measure, but for now, I proceed with the current results.

```
foo <- foo %>%
  arrange(desc(Resid_Violence))
head(foo)
```

```
## # A tibble: 6 x 2
##                          Title Resid_Violence
##                          <chr>          <dbl>
## 1              Erin Brockovich       6.354295
## 2      The Blair Witch Project       2.503311
## 3             Meet the Parents       1.961366
## 4 There's Something About Mary       1.475858
## 5                      Ice Age       1.398374
## 6                   Shark Tale       1.398374
```

The top 6 movies that are "less violent than expected" seem to make sense as well, though somewhat less so than the "more violent" movies. R-rated movies with little to no violence seem to cause the model some issues. Again, more controls may help here, but for now I'll proceed with the current results.

```
movie <- movie %>%
  mutate(exp_viol_strong = Exp_Violence>=8,
         exp_viol_mild = Exp_Violence>4 & Exp_Violence<8,
         exp_viol_non = Exp_Violence<=4)

daily_exposure <- movie %>%
  group_by(Date) %>%
  summarise(tickets_tot = sum(Tickets),
            tickets_strong = sum(ifelse(viol_strong, Tickets, 0)),
            tickets_mild = sum(ifelse(viol_mild, Tickets, 0)),
            tickets_non = sum(ifelse(viol_non, Tickets, 0)),
            tickets_exp_strong = sum(ifelse(exp_viol_strong, Tickets, 0)),
            tickets_exp_mild = sum(ifelse(exp_viol_mild, Tickets, 0)),
            tickets_exp_non = sum(ifelse(exp_viol_non, Tickets, 0)))
```

Exporting Data

With the full data frame compiled, I export the data for analysis.

```
write_csv(daily_exposure, path="master.csv")
saveRDS(daily_exposure, "master.rds")
```