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.

```r
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)

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

"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.

```r
# 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 Phant
               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 sales from rescreenings/rereleases of movies
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)))

# Joining movie_ratings and movie_genres
movie_ratings <- movie_genres %>%
  left_join(movie_ratings, by="Title") %>%
  filter(Year >= 1995 & Year <= 2005)

# 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_ratings_temp <- movie_ratings %>%
  mutate(Year = Year+1)
movie <- movie_sales %>%
  left_join(movie_ratings, by=c("Title", "Year")) %>%
  left_join(movie_ratings_temp, by=c("Title", "Year")) %>%
  mutate(Genre = ifelse(is.na(Genre.x), Genre.y, Genre.x)) %>%
  mutate(Info = ifelse(is.na(Info.x), Info.y, Info.x)) %>%
  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(-Genre.x, -Info.x, -MPAA_Rating.x, -Violence.x, -viol_strong.x, -viol_mild.x, -viol_non.x,
         -Genre.y, -Info.y, -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)

# Cleanup
movie_ratings$Genre = as.factor(movie_ratings$Genre)
rm(movie_genres)
rm(movie_ratings_temp)
rm(ticket_prices)
```

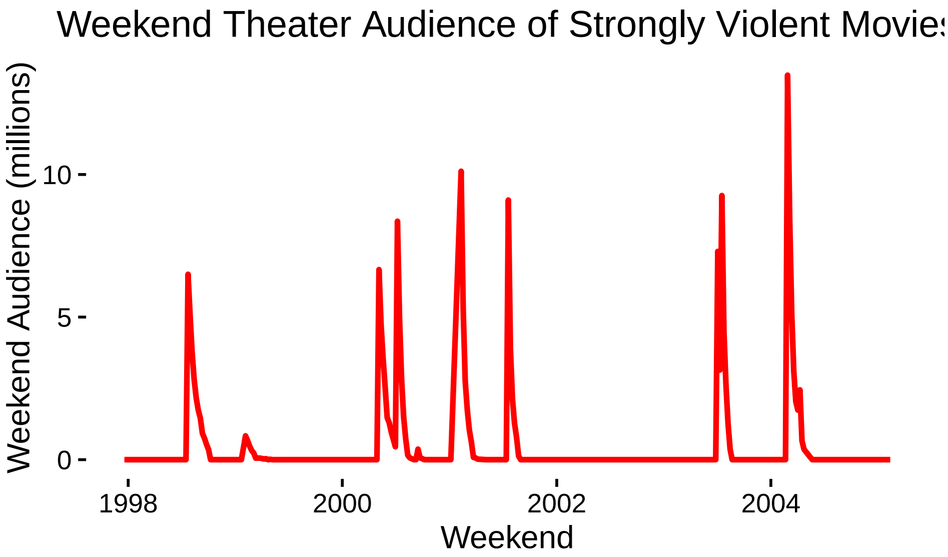
Finally, we can compute the daily exposure to movie violence.

```
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.

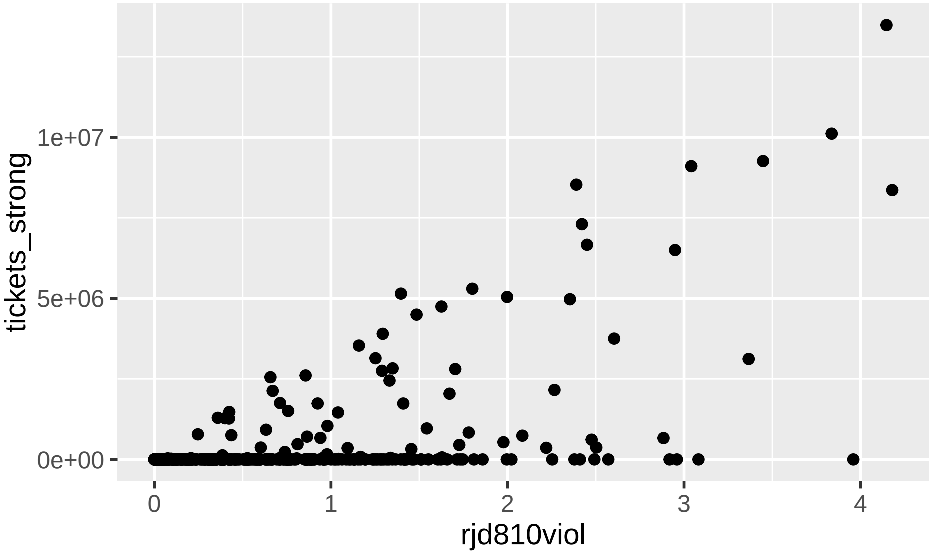
```
# Aggregate by weekend
weekend_exposure <- daily_exposure %>%
  ungroup() %>%
  filter(Weekday==1|Weekday==6|Weekday==7) %>%
  mutate(Date = ifelse(Weekday==7, Date-1,
                ifelse(Weekday==1, Date-2, Date))) %>%
  mutate(Date = as.Date(Date, origin="1970-01-01 UTC")) %>%
  group_by(Date) %>%
  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(colour="red", lwd=1) +
  labs(title="Weekend Theater Audience of Strongly Violent Movies",
       y="Weekend Audience (millions)",
       x="Weekend") +
  scale_y_continuous(labels=function(x){x/1000000})  +
  theme_bw() +
  theme(axis.line = element_line(colour = "black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank())
```



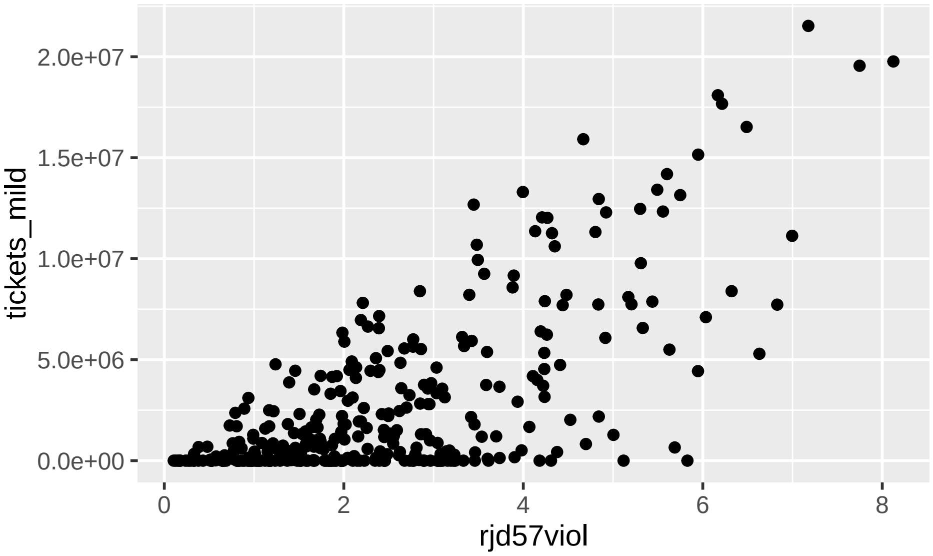
We note a few things. First, our exposure numbers are definitely sparser, which is expected, given that we only use top 10 movies. Second, our exposure numbers don't peak as highly (again expected).

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

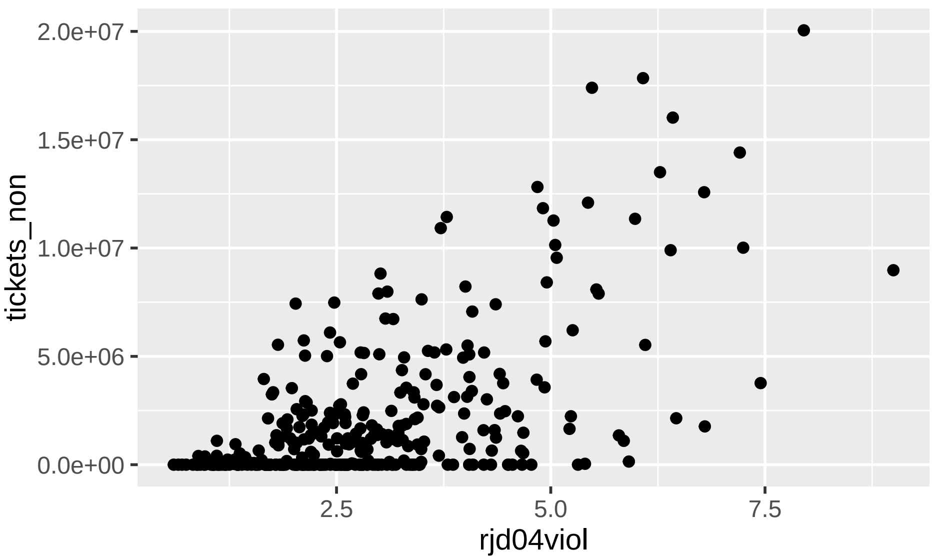
```r
dd <- read_csv("fulliblockday.csv")
dd_ev <- dd %>%
  select(mdy, rjd04viol, rjd57viol, rjd810viol) %>%
  # Origin date computed from information in data
  mutate(Date = as.Date(mdy, origin="1960-01-01 UTC"))
compare_ev <- weekend_exposure %>%
  left_join(dd_ev, by="Date")
  # not sure how exactly 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()
```



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



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



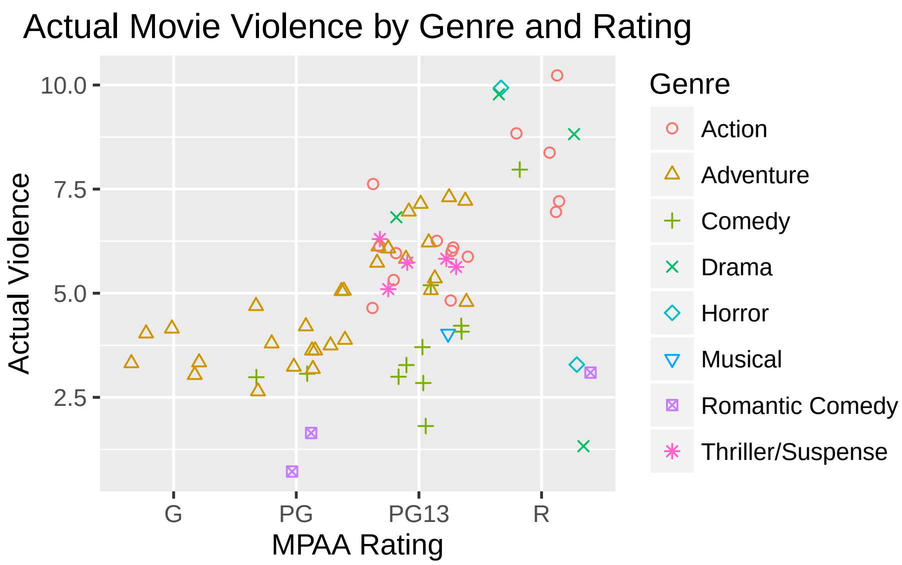
Obviously, these aren't perfect relationships. In particular, we can see how my data underestimates exposure on many days.

Computing Expected Exposure to Violence

Though my data are not perfect, they do not display any glaring discrepancies with Dahl and DellaVigna's data either. As such, I proceed to calculate expected exposure to violence using a movie's MPAA rating and genre.

First, I explore the variation in actual violence by MPAA rating and genre.

```r
ggplot(movie_ratings, aes(x=MPAA_Rating, y=Violence)) +
  geom_jitter(aes(color=Genre, shape=Genre)) +
  scale_shape_manual(values=1:nlevels(movie_ratings$Genre)) +
  labs(title="Actual Movie Violence by Genre and Rating",
       y="Actual Violence",
       x="MPAA Rating")
```



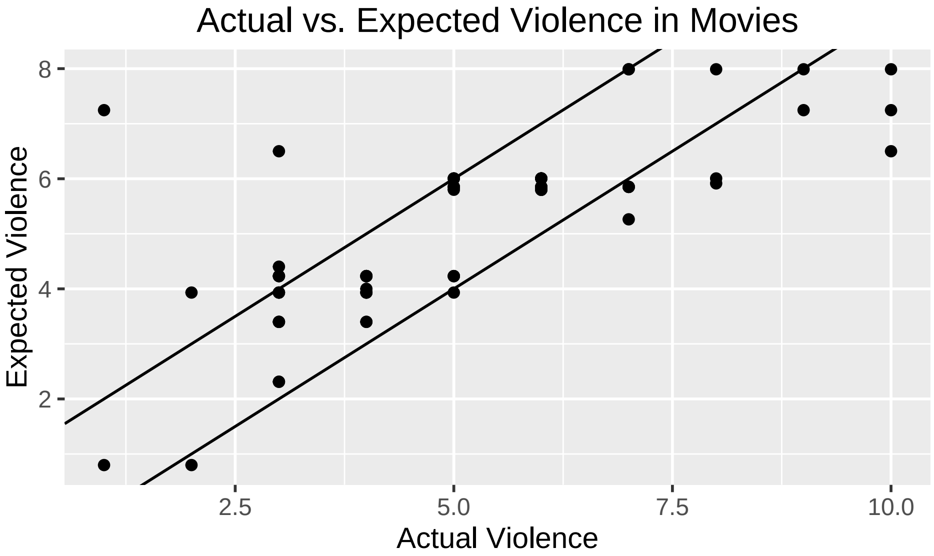
We see that there are definitely patterns in the data, but the relationships between MPAA Rating/Genre and Violence aren't perfect. Thus, we proceed to use regression to predict violence.

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

```
##
## Call:
## lm(formula = Violence ~ Genre + MPAA_Rating, data = movie_ratings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2458 -0.8389  0.0056  0.6878  3.5000
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.5537     0.8599   4.133 0.000115 ***
## GenreAdventure            -0.1537     0.5658  -0.272 0.786789
## GenreComedy               -2.0731     0.5979  -3.467 0.000987 ***
## GenreDrama                -0.7430     0.8538  -0.870 0.387700
## GenreHorror               -1.4889     1.1640  -1.279 0.205877
## GenreMusical              -2.0056     1.5092  -1.329 0.188987
## GenreRomantic Comedy      -3.5865     0.9759  -3.675 0.000515 ***
## GenreThriller/Suspense    -0.2056     0.7751  -0.265 0.791762
## MPAA_RatingPG              0.8315     0.7554   1.101 0.275478
## MPAA_RatingPG13            2.4518     0.7554   3.246 0.001933 **
## MPAA_RatingR               4.4351     0.9531   4.654 1.89e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.448 on 59 degrees of freedom
## Multiple R-squared:  0.5733, Adjusted R-squared:  0.501
## F-statistic: 7.928 on 10 and 59 DF,  p-value: 6.365e-08
```

We see that Genre and MPAA Rating capture 50 percent of the variation in kids-in-mind violence. We plot the relationship visually.

```
ggplot(movie_ratings, 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) +
  geom_abline(slope=1, intercept=1) +
  geom_abline(slope=1, intercept=-1)
```



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

```
## [1] 0.7571891
```

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_ratings %>%
  mutate(Resid_Violence = Exp_Violence-Violence) %>%
  arrange(Resid_Violence)
head(foo)
```

```
##                        Title  Genre
## 1                   Hannibal Horror
## 2 The Passion of the Christ  Drama
## 3                Scary Movie Comedy
## 4                  Gladiator Action
## 5            Jurassic Park 3 Action
## 6        Saving Private Ryan  Drama
##                                                                    Info
## 1                       R for strong gruesome violence, some nudity and language
## 2                                           R for sequences of graphic violence
## 3               R for strong crude sexual humor, language, drug use and violence
## 4                                                R for intense, graphic combat
## 5                                    PG-13 for intense sci-fi terror and violence
## 6 R for intense prolonged realistically graphic sequences of war violence, and for language
##   Year MPAA_Rating Violence viol_strong viol_mild viol_non Exp_Violence
## 1 2001           R       10        TRUE     FALSE    FALSE     6.500000
## 2 2004           R       10        TRUE     FALSE    FALSE     7.245825
## 3 2000           R        8        TRUE     FALSE    FALSE     5.915784
## 4 2000           R       10        TRUE     FALSE    FALSE     7.988866
## 5 2001        PG13        8        TRUE     FALSE    FALSE     6.005567
## 6 1998           R        9        TRUE     FALSE    FALSE     7.245825
##   Resid_Violence
## 1      -3.500000
## 2      -2.754175
## 3      -2.084216
## 4      -2.011134
## 5      -1.994433
## 6      -1.754175
```

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)
```

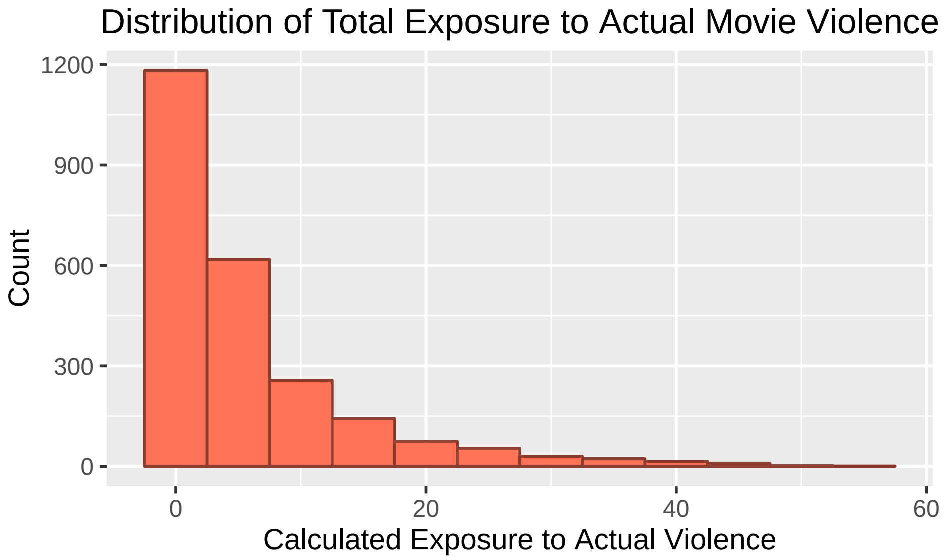
```
##                           Title           Genre
## 1              Erin Brockovich           Drama
## 2      The Blair Witch Project          Horror
## 3             Meet the Parents          Comedy
## 4 There's Something About Mary Romantic Comedy
## 5                      Ice Age       Adventure
## 6                      Shrek 2       Adventure
##                                                                                Info
## 1                                                                    R for language
## 2                                                                                 R
## 3                            PG-13 for sexual content, drug references and language
## 4                                    R for strong comic sexual content and language
## 5                                                                 PG for mild peril
## 6 PG for some crude humor, a brief substance reference and some suggestive content
##   Year MPAA_Rating Violence viol_strong viol_mild viol_non Exp_Violence
## 1 2000           R        1       FALSE     FALSE     TRUE     7.245825
## 2 1999           R        3       FALSE     FALSE     TRUE     6.500000
## 3 2000        PG13        2       FALSE     FALSE     TRUE     3.932486
## 4 1998           R        3       FALSE     FALSE     TRUE     4.402412
## 5 2002          PG        3       FALSE     FALSE     TRUE     4.231507
## 6 2004          PG        3       FALSE     FALSE     TRUE     4.231507
##   Resid_Violence
## 1       6.245825
## 2       3.500000
## 3       1.932486
## 4       1.402412
## 5       1.231507
## 6       1.231507
```

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.

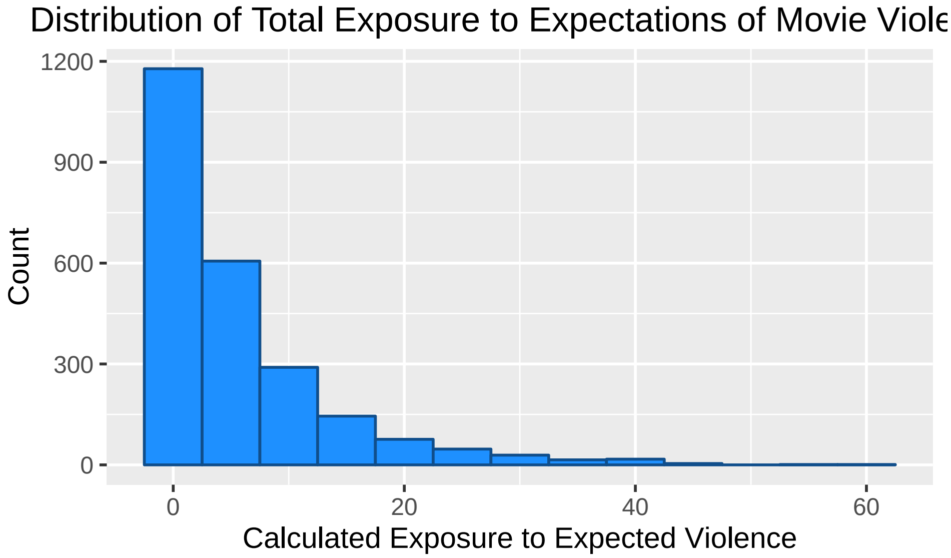
```
movie <- movie %>%
  left_join(select(movie_ratings, Title, Exp_Violence), by="Title")

movie <- movie %>%
  mutate(exp_viol_strong = Exp_Violence>=7,   # Note: 7, not 8 because of expected violence model
         exp_viol_mild = Exp_Violence>4 & Exp_Violence<7,
         exp_viol_non = Exp_Violence<=4) %>%
  mutate(more_violent = Exp_Violence-Violence <= -1,
         as_violent = (Exp_Violence-Violence >-1) & (Exp_Violence-Violence <1),
         less_violent = Exp_Violence-Violence >= 1)

daily_exposure <- movie %>%
  group_by(Date) %>%
  summarise(tickets_tot = sum(Tickets)/1000000,
            tickets_strong = sum(ifelse(viol_strong, Tickets, 0))/1000000,
            tickets_mild = sum(ifelse(viol_mild, Tickets, 0))/1000000,
            tickets_non = sum(ifelse(viol_non, Tickets, 0))/1000000,
            tickets_exp_strong = sum(ifelse(exp_viol_strong, Tickets, 0))/1000000,
            tickets_exp_mild = sum(ifelse(exp_viol_mild, Tickets, 0))/1000000,
            tickets_exp_non = sum(ifelse(exp_viol_non, Tickets, 0))/1000000,
            tickets_more_violent = sum(ifelse(more_violent, Tickets, 0))/1000000,
            tickets_as_violent = sum(ifelse(as_violent, Tickets, 0))/1000000,
            tickets_less_violent = sum(ifelse(less_violent, Tickets, 0))/1000000,
            tickets_to_violence = tickets_non + 2*tickets_mild + 3*tickets_strong,
            tickets_to_exp_violence = tickets_exp_non + 2*tickets_exp_mild + 3*tickets_exp_strong,
            tickets_to_aggviol = sum(Tickets*Violence)/1000000,
            tickets_to_aggexpviol = sum(Tickets*Exp_Violence)/1000000,
            ln_ttav = log1p(tickets_to_aggviol),
            ln_ttaev = log1p(tickets_to_aggexpviol))
```

Here, I check the distribution of the `tickets_to_aggviol` and `tickets_to_aggexpviol` variables (to demonstrate their strong right skew), and then compare those distributions to those of the variables logged.

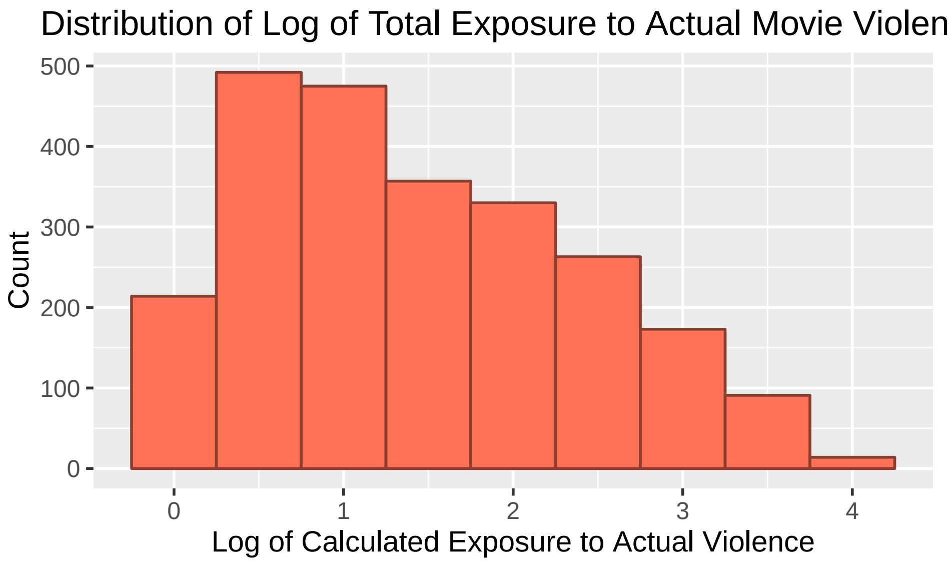
```
ggplot(daily_exposure, aes(x=tickets_to_aggviol)) +
  geom_histogram(binwidth=5, color="coral4", fill="coral1") +
  labs(title="Distribution of Total Exposure to Actual Movie Violence",
       y="Count",
       x="Calculated Exposure to Actual Violence")
```



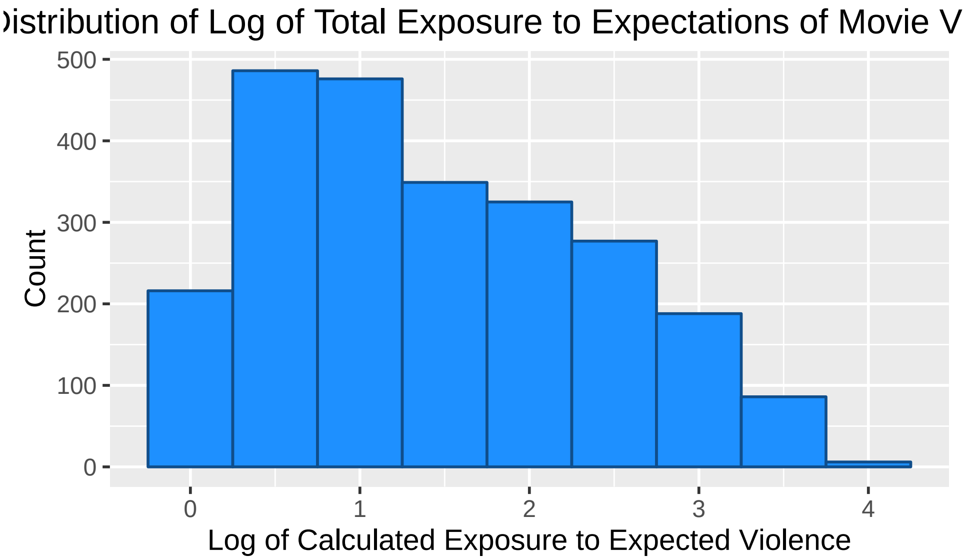
```
ggplot(daily_exposure, aes(x=tickets_to_aggexpviol)) +
  geom_histogram(binwidth=5, color="dodgerblue4", fill="dodgerblue1") +
  labs(title="Distribution of Total Exposure to Expectations of Movie Violence",
       y="Count",
       x="Calculated Exposure to Expected Violence")
```



```
ggplot(daily_exposure, aes(x=ln_ttav)) +
  geom_histogram(binwidth=0.5, color="coral4", fill="coral1") +
  labs(title="Distribution of Log of Total Exposure to Actual Movie Violence",
       y="Count",
       x="Log of Calculated Exposure to Actual Violence")
```



```r
ggplot(daily_exposure, aes(x=ln_ttaev)) +
  geom_histogram(binwidth=0.5, color="dodgerblue4", fill="dodgerblue1") +
  labs(title="Distribution of Log of Total Exposure to Expectations of Movie Violence",
       y="Count",
       x="Log of Calculated Exposure to Expected Violence")
```



Exporting Data

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

```r
master <- dd %>%
  mutate(Date = as.Date(mdy, origin="1960-01-01 UTC")) %>%
  left_join(daily_exposure, by="Date") %>%
  filter(year(Date) >= 1998)
master[is.na(master)] <- 0   # for days with 0 movies in my data
```

```r
write_csv(master, path="master.csv")
saveRDS(master, "master.rds")
write_csv(movie, path="movie.csv")
saveRDS(movie, "movie.rds")
write_csv(movie_ratings, path="movie_ratings_final.csv")
saveRDS(movie_ratings, "movie_ratings_final.rds")
```