# Computing Expected Violence Exposure with the Bootstrap

*Jonathan Che*

*10 April 2017*

## Overview

Instead of using simple regression to produce a measure of "expected violence", I implement a bootstrap procedure.

## Method

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

```
movie_ratings <- readRDS("movie_ratings_final.rds")
movie <- readRDS("movie.rds")
dd <- read_csv("fulliblockday.csv")
```

Then, I randomly partition the movies into 5 groups.

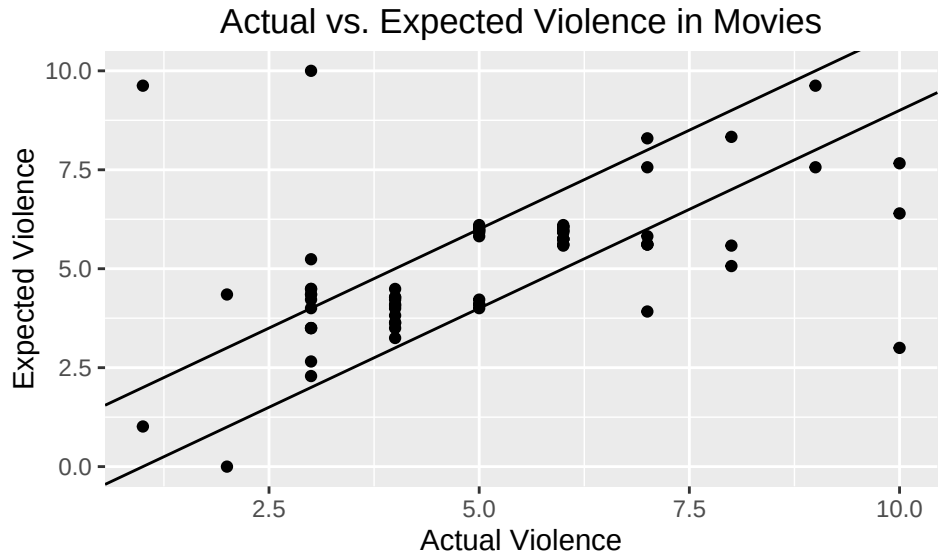
```
set.seed(392)
df <- movie_ratings[sample(nrow(movie_ratings)),]   # Shuffle rows

m1 <- lm(Violence ~ Genre+MPAA_Rating, data=df[15:70,])
p1 <- predict(m1, df[1:14,])
m2 <- lm(Violence ~ Genre+MPAA_Rating, data=df[c(1:14, 29:70),])
p2 <- predict(m2, df[15:28,])
m3 <- lm(Violence ~ Genre+MPAA_Rating, data=df[c(1:28, 43:70),])
p3 <- predict(m3, df[c(29:37, 39:42),])  # 38 is the musical
m4 <- lm(Violence ~ Genre+MPAA_Rating, data=df[c(1:42, 57:70),])
p4 <- predict(m4, df[43:56,])
m5 <- lm(Violence ~ Genre+MPAA_Rating, data=df[1:56,])
p5 <- predict(m5, df[57:70,])

Exp_Viol_Boot <- c(p1,p2,p3,p4,p5)
Exp_Viol_Boot <- append(Exp_Viol_Boot, 4, after=37)
Exp_Viol_Boot <- ifelse(Exp_Viol_Boot<0, 0, Exp_Viol_Boot)

df <- df %>%
  bind_cols(data.frame(Exp_Viol_Boot))
```

```
ggplot(df, aes(x=Violence, y=Exp_Viol_Boot)) +
  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(Exp_Violence~Exp_Viol_Boot, data=df)
```

```
## [1] 0.8962002
```

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 <- df %>%
  mutate(Resid_Violence = Exp_Viol_Boot-Violence) %>%
  arrange(Resid_Violence)
head(foo)
```

```
##                       Title  Genre
## 1                   Hannibal Horror
## 2 The Passion of the Christ  Drama
## 3         The Perfect Storm  Drama
## 4               Scary Movie Comedy
## 5            Jurassic Park 3 Action
## 6                 Gladiator Action
##                                                                  Info Year
## 1         R for strong gruesome violence, some nudity and language 2001
## 2                              R for sequences of graphic violence 2004
## 3                           PG-13 for language and scenes of peril 2000
## 4 R for strong crude sexual humor, language, drug use and violence 2000
## 5                      PG-13 for intense sci-fi terror and violence 2001
## 6                                    R for intense, graphic combat 2000
##   MPAA_Rating Violence viol_strong viol_mild viol_non Exp_Violence
## 1           R       10        TRUE     FALSE    FALSE     6.500000
## 2           R       10        TRUE     FALSE    FALSE     7.245825
## 3        PG13        7       FALSE      TRUE    FALSE     5.262526
## 4           R        8        TRUE     FALSE    FALSE     5.915784
## 5        PG13        8        TRUE     FALSE    FALSE     6.005567
## 6           R       10        TRUE     FALSE    FALSE     7.988866
##   Exp_Viol_Boot Resid_Violence
## 1      3.000000      -7.000000
## 2      6.397261      -3.602739
```

```
## 3      3.919039      -3.080961
## 4      5.067278      -2.932722
## 5      5.584124      -2.415876
## 6      7.663662      -2.336338
```

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                   Shark Tale       Adventure
## 6                      Ice Age       Adventure
##                                                    Info Year MPAA_Rating
## 1                                   R for language 2000           R
## 2                                             R 1999           R
## 3 PG-13 for sexual content, drug references and language 2000        PG13
## 4         R for strong comic sexual content and language 1998           R
## 5                  PG for mild language and crude humor 2004          PG
## 6                                    PG for mild peril 2002          PG
##   Violence viol_strong viol_mild viol_non Exp_Violence Exp_Viol_Boot
## 1        1       FALSE     FALSE     TRUE     7.245825      9.624386
## 2        3       FALSE     FALSE     TRUE     6.500000     10.000000
## 3        2       FALSE     FALSE     TRUE     3.932486      4.349131
## 4        3       FALSE     FALSE     TRUE     4.402412      5.239582
## 5        3       FALSE     FALSE     TRUE     4.231507      4.489142
## 6        3       FALSE     FALSE     TRUE     4.231507      4.489142
##   Resid_Violence
## 1       8.624386
## 2       7.000000
## 3       2.349131
## 4       2.239582
## 5       1.489142
## 6       1.489142
```

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(df, Title, Exp_Viol_Boot), by="Title")

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

```
         less_violent = Exp_Viol_Boot-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_Viol_Boot)/1000000)
```

### Exporting Data

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

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

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

## Sizing effects

```
m6 <- lm(rjd04viol ~ tickets_non, data=master_boot)
m7 <- lm(rjd57viol ~ tickets_mild, data=master_boot)
m8 <- lm(rjd810viol ~ tickets_strong, data=master_boot)
summary(m6)
```

```
##
## Call:
## lm(formula = rjd04viol ~ tickets_non, data = master_boot)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6572 -0.9342 -0.4618  0.6817  7.3859
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.42130    0.02721   52.23   <2e-16 ***
## tickets_non  1.19894    0.02986   40.16   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.239 on 2553 degrees of freedom
## Multiple R-squared:  0.3871, Adjusted R-squared:  0.3869
## F-statistic:  1613 on 1 and 2553 DF,  p-value: < 2.2e-16
```

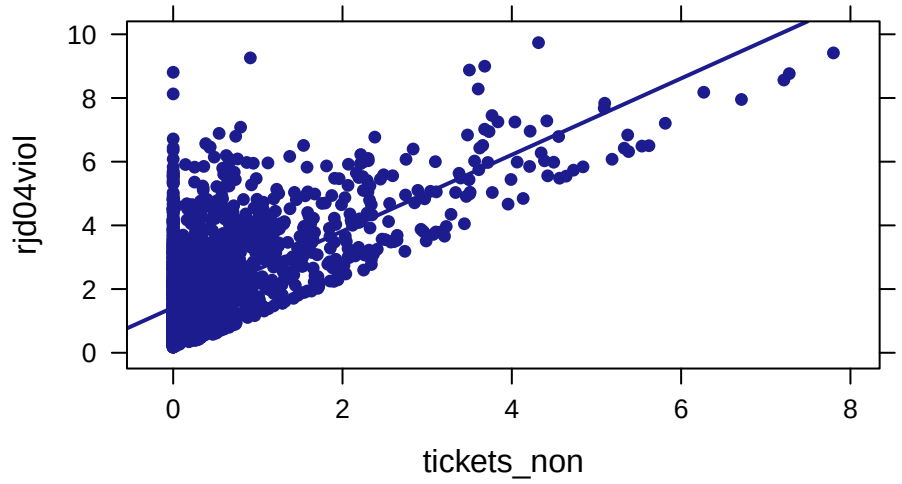
```
summary(m7)
```

```
##
## Call:
## lm(formula = rjd57viol ~ tickets_mild, data = master_boot)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1711 -0.7233 -0.3982  0.4369  5.3332
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.03898    0.02311   44.95   <2e-16 ***
## tickets_mild  1.04496    0.01964   53.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.034 on 2553 degrees of freedom
## Multiple R-squared:  0.5259, Adjusted R-squared:  0.5257
## F-statistic:  2832 on 1 and 2553 DF,  p-value: < 2.2e-16
```

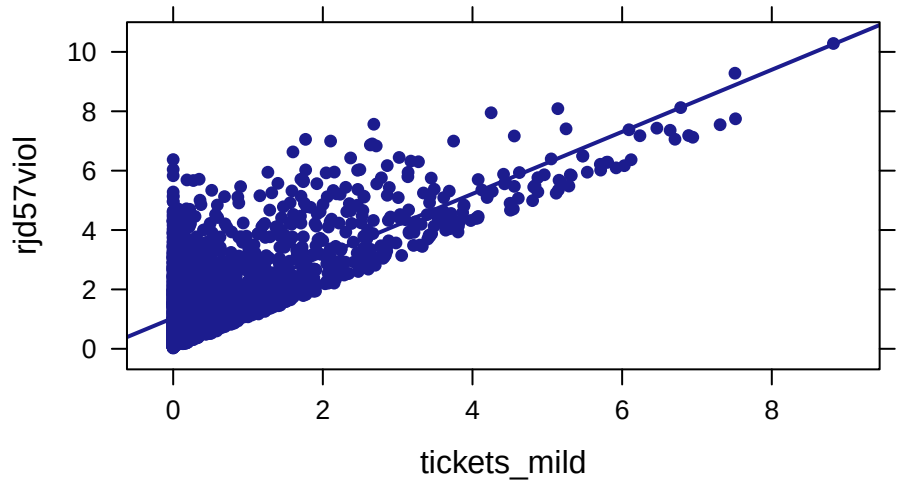
```
summary(m8)
```

```
##
## Call:
## lm(formula = rjd810viol ~ tickets_strong, data = master_boot)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7084 -0.3330 -0.1936  0.1106  3.5511
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.40778    0.01069   38.13   <2e-16 ***
## tickets_strong  1.02855    0.02615   39.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5231 on 2553 degrees of freedom
## Multiple R-squared:  0.3774, Adjusted R-squared:  0.3772
## F-statistic:  1548 on 1 and 2553 DF,  p-value: < 2.2e-16
```

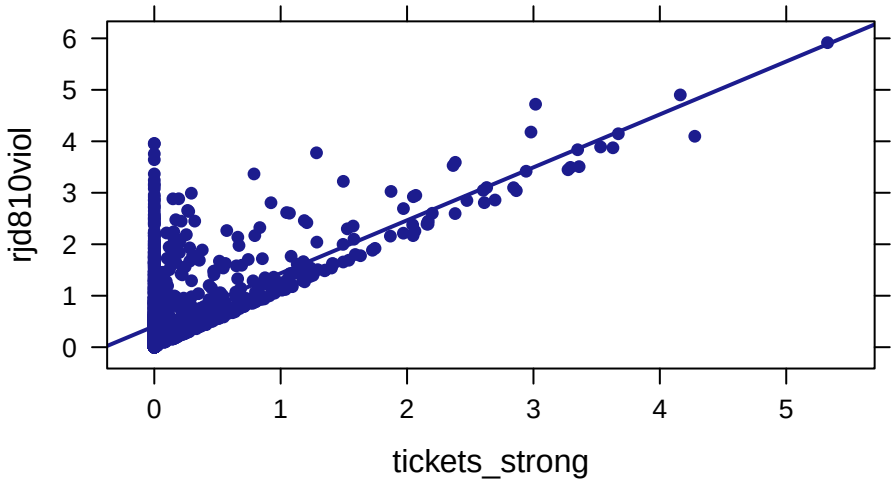
```
xyplot(rjd04viol ~ tickets_non, type=c("p","r"),data=master_boot)
```




```
xyplot(rjd57viol ~ tickets_mild, type=c("p","r"),data=master_boot)
```




```
xyplot(rjd810viol ~ tickets_strong, type=c("p","r"),data=master_boot)
```


rjd810viol
tickets_strong
0
1
2
3
4
5
6