Modeling_Katrina

Katrina Truebebach

March 18, 2019

```
rm(list = ls())
```

Load cleaned data

```
load(file = '~/DS5110/data/proj_cleaned_dta.RData')
```

Fit Model with Genre Variables vs Real Revenue

Step Wise Selection

End model includes (in order of steps): 'Adventure', 'Fantasy', 'Action', 'Thriller', 'Documentary', 'Horror', 'Drama', 'Comedy', 'War', 'Musical'

This model selection is initially surprising because some of the included variables are not significant and, according to Qiang's graphs in EDA, do not make a real difference to real_gross. Also, some genres that look like they would make a signficant difference are not included.

Thoughts:

- There are a few genres that define almost all of the movies (For example, Adventure, Action, and Drama identify 1341 out of 1885 movies). Thus, the relationship between revenue and some genres can be explained by other generes. For example, 78 out of 99 Animation movies are also Comedy. So Animation's effect on revenue may already by captured by Comedy.
  - Maybe not the best argument. . . 51 out of 62 War movies are Drama. But both included in the model (although War is close to the cutoff of not being included based on how much it decreases RMSE)
- On the flip side, Comedy and Thriller are included even though they seem to have a negligable effect on revenue based on the EDA bar graphs. One theory is that neither of these generes correlate well/are explained largely by another genre, thus making the relationship with genre more explanatory, even if the effect on revenue is small and insignificant.

Also, the residuals are debatably random vs included and excluded variables in model (not sure if these are not-random enough to matter – see graphs).
More concerning is the fact that the residuals themselves are not Normal. See QQ-Plot

```
train %>% filter(Animation == 1, Comedy == 1) %>% count() # 78
train %>% filter(Animation == 1, (Fantasy == 1 | Adventure == 1 | Comedy == 1)) %>% count() # 99
train %>% filter(Animation == 1) %>% count() # 99
train %>% count()

train %>% filter(War == 1, Drama == 1) %>% count() # 51
train %>% filter(War == 1) %>% count() # 62

train %>% filter(Thriller == 1) %>% count() # 508
train %>% filter(Thriller == 1, Adventure == 1) %>% count()
train %>% filter(Thriller == 1, Fantasy == 1) %>% count()
train %>% filter(Thriller == 1, Action == 1) %>% count()
train %>% filter(Thriller == 1, Documentary == 1) %>% count()
```

```r
train %>% filter(Thriller == 1, Horror == 1) %>% count()
train %>% filter(Thriller == 1, Drama == 1) %>% count()
train %>% filter(Thriller == 1, Comedy == 1) %>% count()
train %>% filter(Thriller == 1, War == 1) %>% count()
train %>% filter(Thriller == 1, Musical == 1) %>% count()


train %>% filter(Comedy == 1) %>% count() # 793
train %>% filter(Comedy == 1, Adventure == 1) %>% count()
train %>% filter(Comedy == 1, Fantasy == 1) %>% count()
train %>% filter(Comedy == 1, Action == 1) %>% count()
train %>% filter(Comedy == 1, Documentary == 1) %>% count()
train %>% filter(Comedy == 1, Horror == 1) %>% count()
train %>% filter(Comedy == 1, Drama == 1) %>% count()
train %>% filter(Comedy == 1, Thriller == 1) %>% count()
train %>% filter(Comedy == 1, War == 1) %>% count()
train %>% filter(Comedy == 1, Musical == 1) %>% count()
```

Which genres should we be using?
Note: not using the step() function because can't fit and find RMSE on different datasets (train, valid)

```r
# version of train set with just genre columns to loop through
train_genre <- train %>% select(Action, Adventure, Animation, Biography, Comedy, Crime, Documentary,
                Drama, Family, Fantasy, History, Horror, Music, Musical, Mystery,
                Romance, SciFi, Sport, Thriller, War, Western)

# function to automate each step of stepwise variable selection
step_wise_step <- function(genre_lst = NULL, formula = NULL) {
  # if first step
  if (length(genre_lst) == 0) {
    # rmse with each variable against real_gross
    rmse_genre <- sapply(names(train_genre), function(var) {
      rmse(lm(as.formula(str_c('real_gross ~', var)), data = train), data = valid)
    })
  # if > first step: exclude variables from genre_lst from data and include in model formula
  } else {
    rmse_genre <- sapply(names(train_genre %>% select(-genre_lst)), function(var) {
      rmse(lm(as.formula(str_c('real_gross ~', formula, ' + ', var)),
              data = train), data = valid)
    })
  }
  # return the name and value of the genre that resulted in the lowest RMSE
  return(rmse_genre[which.min(rmse_genre)])
}

# loop through each step wise loop
step_wise_loop <- function() {
  # list to store min RMSE from each step in
  rmse_lst <- c()

  # first step: no genre_lst or formula (default values NULL)
  min_genre <- step_wise_step()
  print(names(min_genre))
```

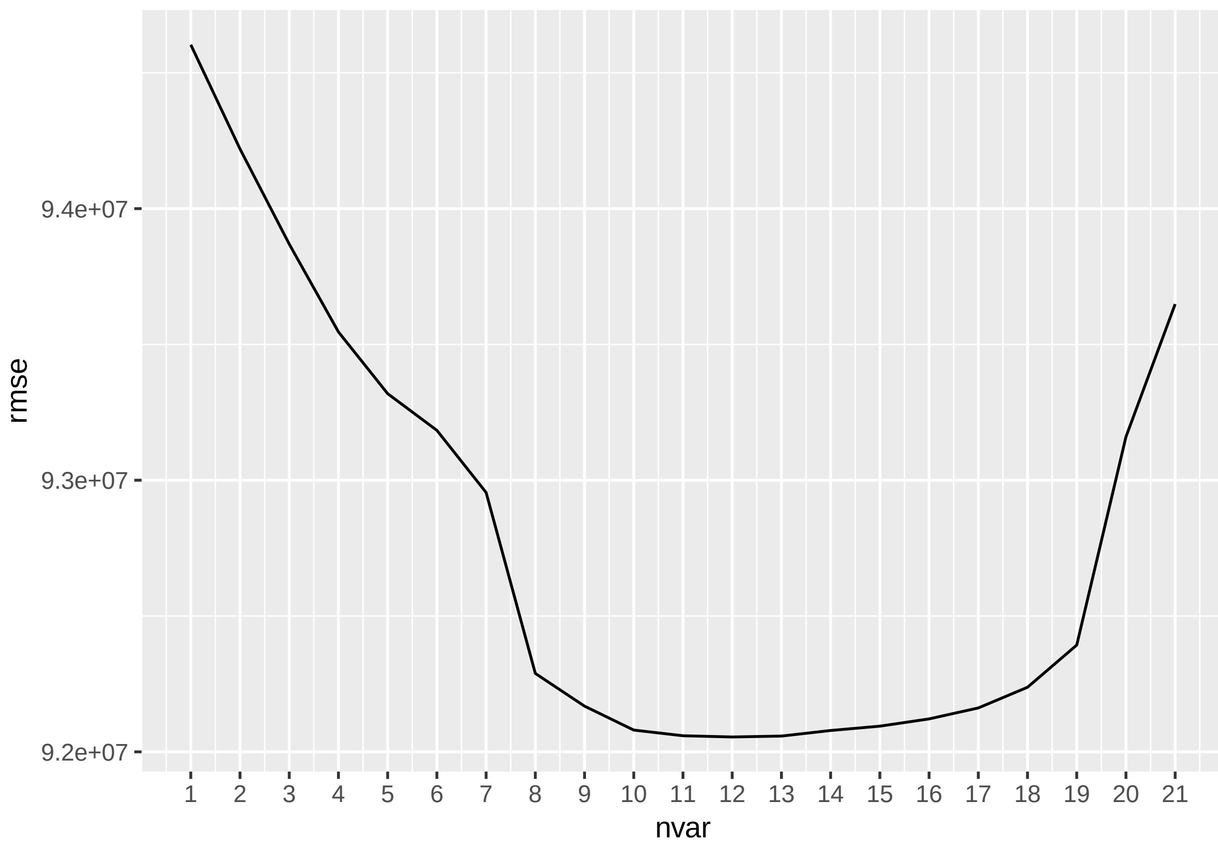
```r
  # add to list of genres, formula, and min RMSE list
  genre_lst <- names(min_genre)
  formula <- str_c(names(min_genre))
  rmse_lst <- c(rmse_lst, min(min_genre))

  # loop through until have considered every genre variable
  for (i in seq(1:(ncol(train_genre)-1))) {
    print(i)
    # step
    min_genre <- step_wise_step(genre_lst = genre_lst, formula = formula)
    print(min_genre)

    # add to lists
    genre_lst <- c(genre_lst, names(min_genre))
    formula <- str_c(formula, ' + ', names(min_genre))
    rmse_lst <- c(rmse_lst, min(min_genre))
  }
  return(rmse_lst)
}

# step wise implement
# return list of all min RMSE from each step -> graph
rmse_lst <- step_wise_loop()
```

```
## [1] "Adventure"
## [1] 1
##  Fantasy
## 94219722
## [1] 2
##   Action
## 93869425
## [1] 3
## Thriller
## 93546130
## [1] 4
## Documentary
##    93318784
## [1] 5
##   Horror
## 93183612
## [1] 6
##    Drama
## 92954767
## [1] 7
##   Comedy
## 92288768
## [1] 8
##      War
## 92168244
## [1] 9
##  Musical
## 92080106
## [1] 10
##    SciFi
```

```
## 92059199
## [1] 11
##  Mystery
## 92054694
## [1] 12
##    Sport
## 92058043
## [1] 13
##  Romance
## 92078513
## [1] 14
##    Crime
## 92094513
## [1] 15
##    Music
## 92121260
## [1] 16
## Biography
##  92161603
## [1] 17
##  Western
## 92237697
## [1] 18
##  History
## 92393043
## [1] 19
##   Family
## 93159768
## [1] 20
## Animation
##  93648551
```

Graph RMSE vs number of variables: how many to include?
Specify 'final' model

```r
# graph RMSE at each step
fit_rmse <- tibble(nvar = 1:length(rmse_lst),
                   rmse = rmse_lst)
ggplot(fit_rmse) + geom_line(aes(x = nvar, y = rmse))+
  scale_x_continuous(breaks = seq(1, length(rmse_lst), by = 1))
```




```r
# after var 10, decreases too small or increase

# model based off of step wise
# HOWEVER some of these variables are insignificant
  # (see pvalues and graphs from Qiang's EDA where barely any difference in revenue from genre)
mod <- lm(real_gross ~ Adventure + Fantasy + Action + Thriller +
            Documentary + Horror + Drama + Comedy + War + Musical,
          data = train)
summary(mod)
```

```
##
## Call:
## lm(formula = real_gross ~ Adventure + Fantasy + Action + Thriller +
##     Documentary + Horror + Drama + Comedy + War + Musical, data = train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -164552357  -44787756  -20325205   18977925  817098772
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 68172888    5824775  11.704  < 2e-16 ***
## Adventure   47351871    5736059   8.255 2.83e-16 ***
## Fantasy     30136443    6141247   4.907 1.00e-06 ***
## Action      21091460    5661485   3.725 0.000201 ***
## Thriller    -8379615    5241750  -1.599 0.110072
```

```
## Documentary -51113957   12915545  -3.958 7.85e-05 ***
## Horror      -23091856    7312057  -3.158 0.001614 **
## Drama       -21932011    4948817  -4.432 9.89e-06 ***
## Comedy       -9079349    4903266  -1.852 0.064227 .
## War          17680670   11462995   1.542 0.123142
## Musical      19955446   11947709   1.670 0.095040 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 86360000 on 1874 degrees of freedom
## Multiple R-squared:  0.1397, Adjusted R-squared:  0.1351
## F-statistic: 30.43 on 10 and 1874 DF,  p-value: < 2.2e-16
```

```r
rmse(mod, data = valid)
```

```
## [1] 92080106
```

```r
# list of these variables for future use
genre_xvar <- c('Adventure', 'Fantasy', 'Action', 'Thriller',
                'Documentary', 'Horror', 'Drama', 'Comedy', 'War', 'Musical')
```

Try alternate model with just genres that looked the most significant from eyeballing Qiang's graphs. RMSE is higher

```r
# alternate model just using the genres that looked correct based on eyeballing Qiang's graphs
mod2 <- lm(real_gross ~ Action + Adventure + Animation + Documentary + Family +
             Fantasy + SciFi, data = train)
rmse(mod2, data = valid) # higher rmse
```

```
## [1] 94815203
```

```r
summary(mod2)
```

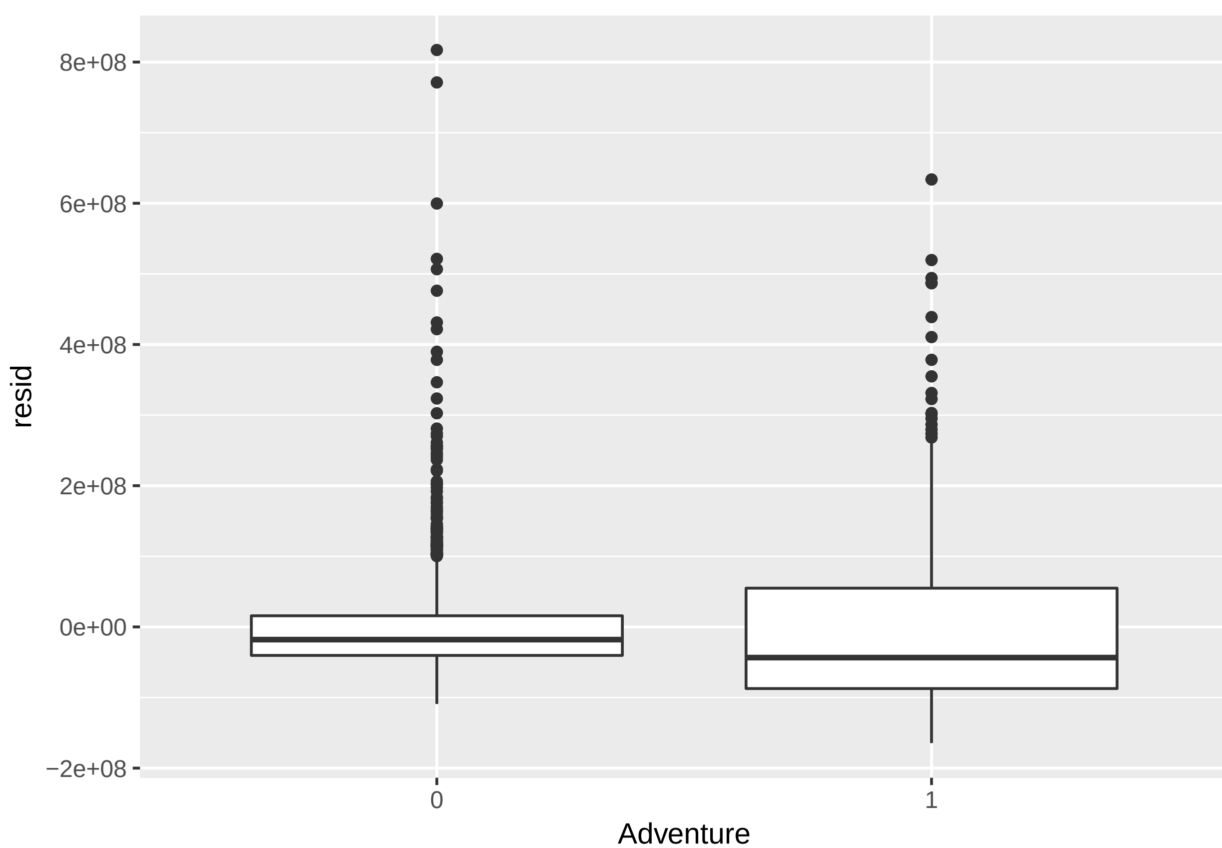
```
##
## Call:
## lm(formula = real_gross ~ Action + Adventure + Animation + Documentary +
##     Family + Fantasy + SciFi, data = train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -168828393  -42752666  -22956215   16692284  797610950
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  43221356    2522857  17.132  < 2e-16 ***
## Action       28066620    5198127   5.399 7.54e-08 ***
## Adventure    37978555    6026589   6.302 3.66e-10 ***
## Animation    36912510   10956603   3.369 0.000770 ***
## Documentary -28739345   12148523  -2.366 0.018099 *
## Family       22257312    7571034   2.940 0.003324 **
## Fantasy      25001737    6159565   4.059 5.13e-05 ***
## SciFi        22182042    6550397   3.386 0.000723 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 85890000 on 1877 degrees of freedom
## Multiple R-squared:  0.1478, Adjusted R-squared:  0.1446
```

```
## F-statistic: 46.49 on 7 and 1877 DF,  p-value: < 2.2e-16
```

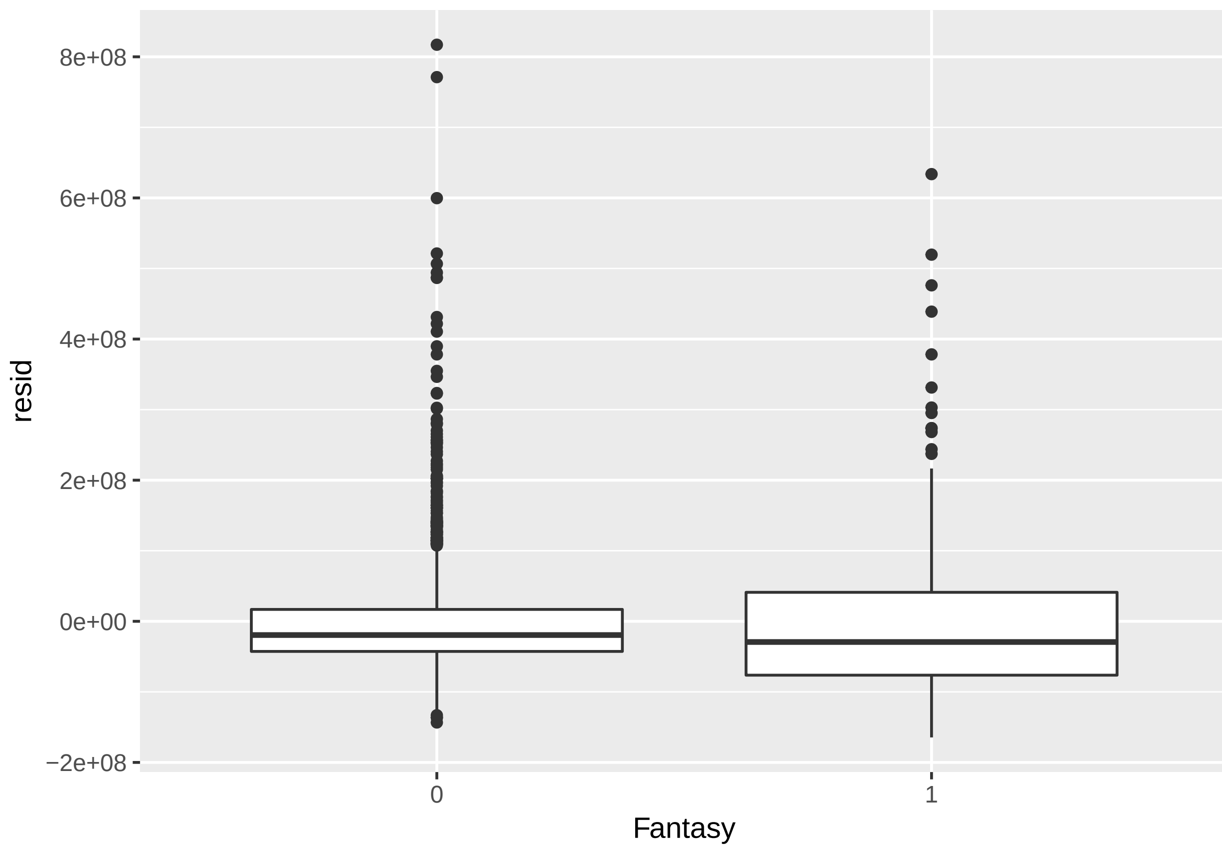
Graph variables in and out of model against residuals
There are some relationships (Adventure, Animation etc.) that don't look random. Not sure if big enough deviation to matter however.

```r
# graph residuals against each variable included in the model
# most look random except Adventure
train <- train %>%
  add_residuals(mod) %>%
  mutate_at(vars(Action, Adventure, Animation, Biography, Comedy, Crime, Documentary,
                 Drama, Family, Fantasy, History, Horror, Music, Musical, Mystery,
                 Romance, SciFi, Sport, Thriller, War, Western),
            funs(as.factor(.)))

lapply(genre_xvar, function(var) {
  train %>%
    ggplot() +
    geom_boxplot(aes_string(var, y = 'resid'))
})
```

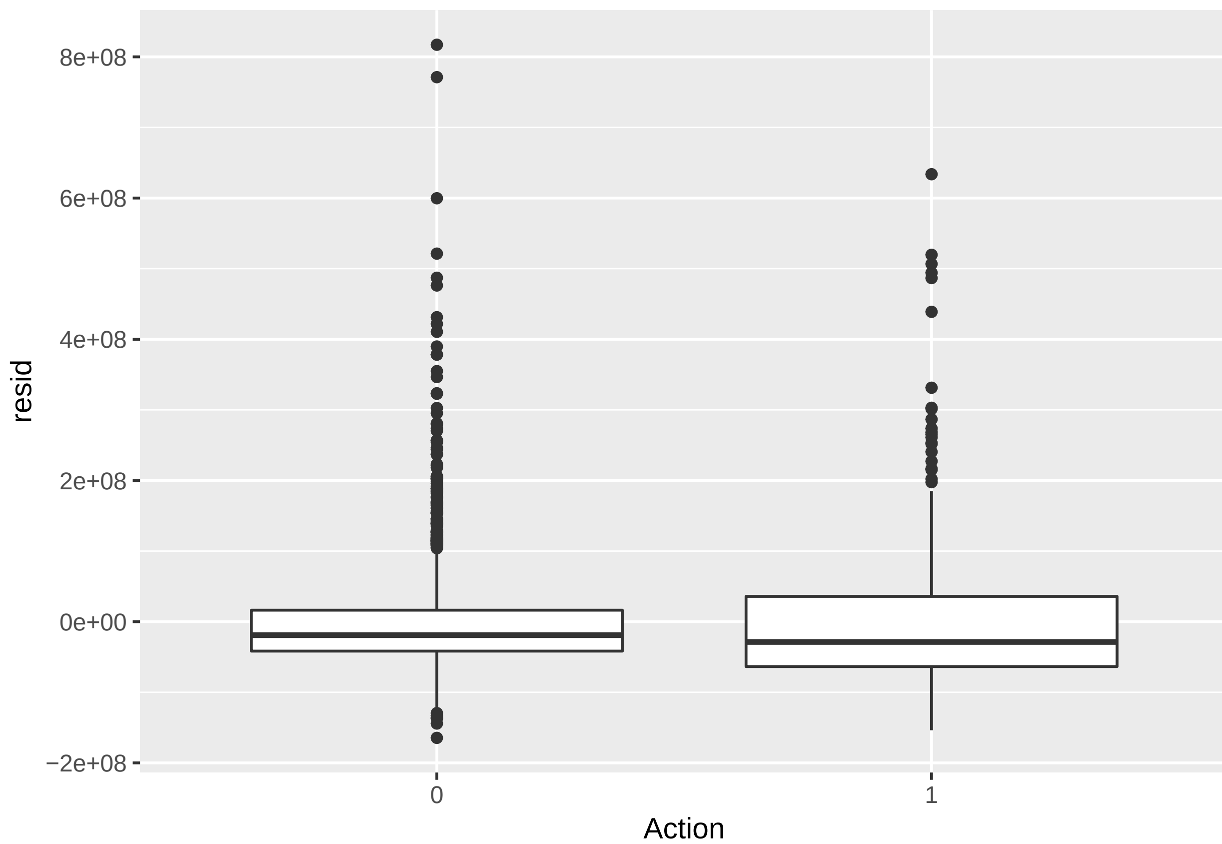
```
## [[1]]
```



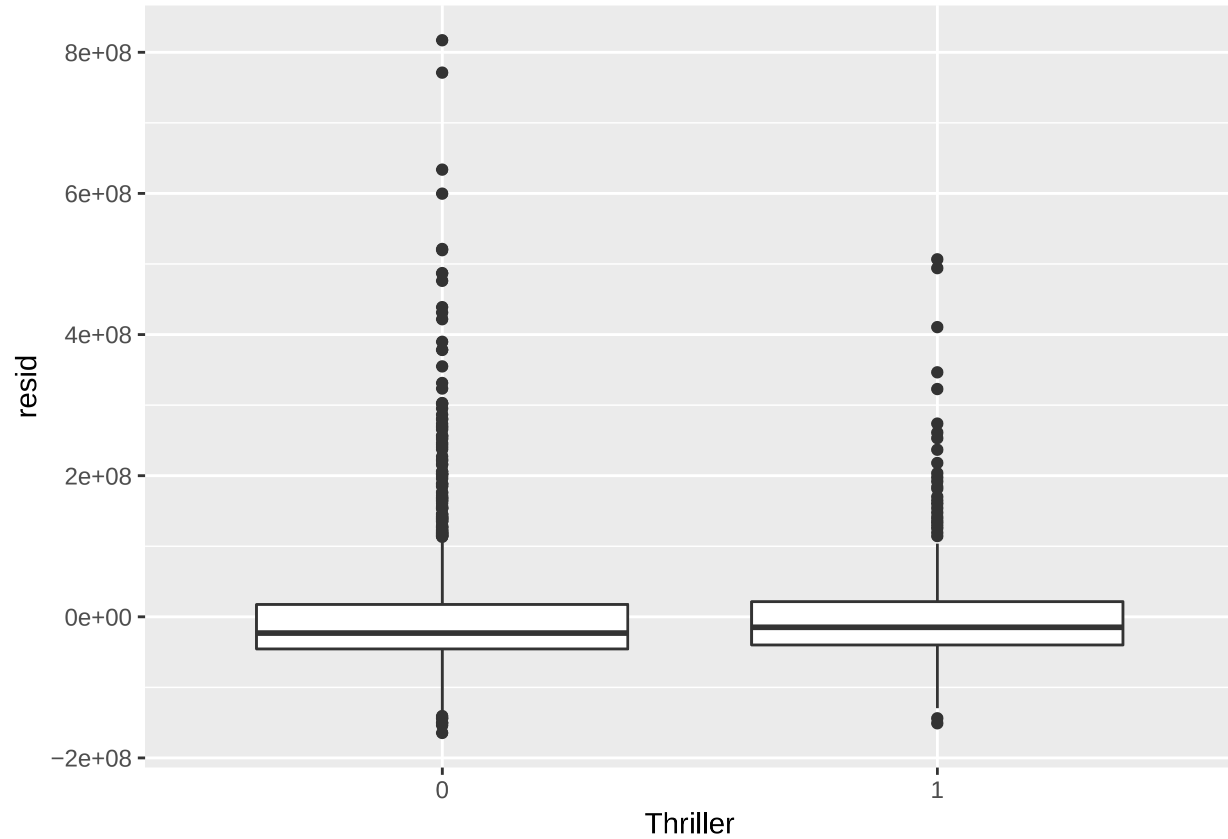

```
##
## [[2]]
```



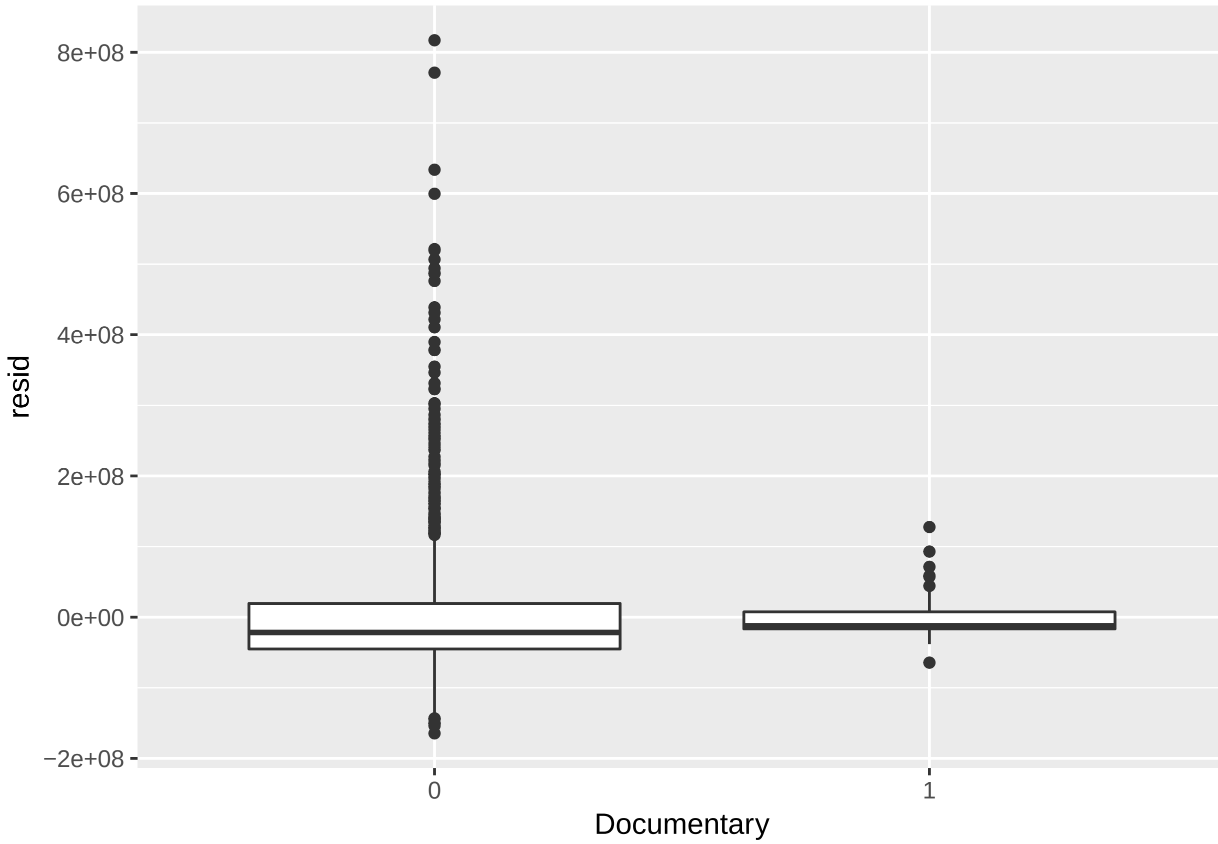

```
##
## [[3]]
```



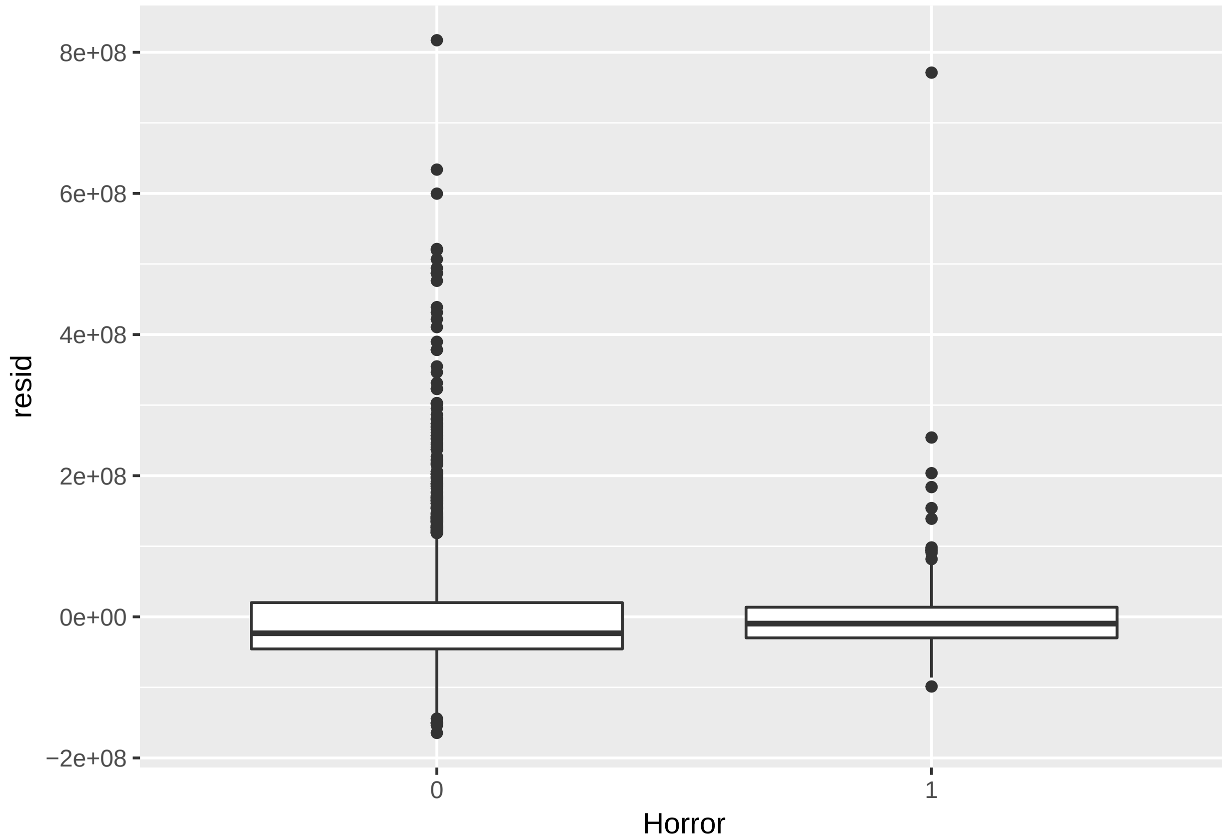

```
##
## [[4]]
```



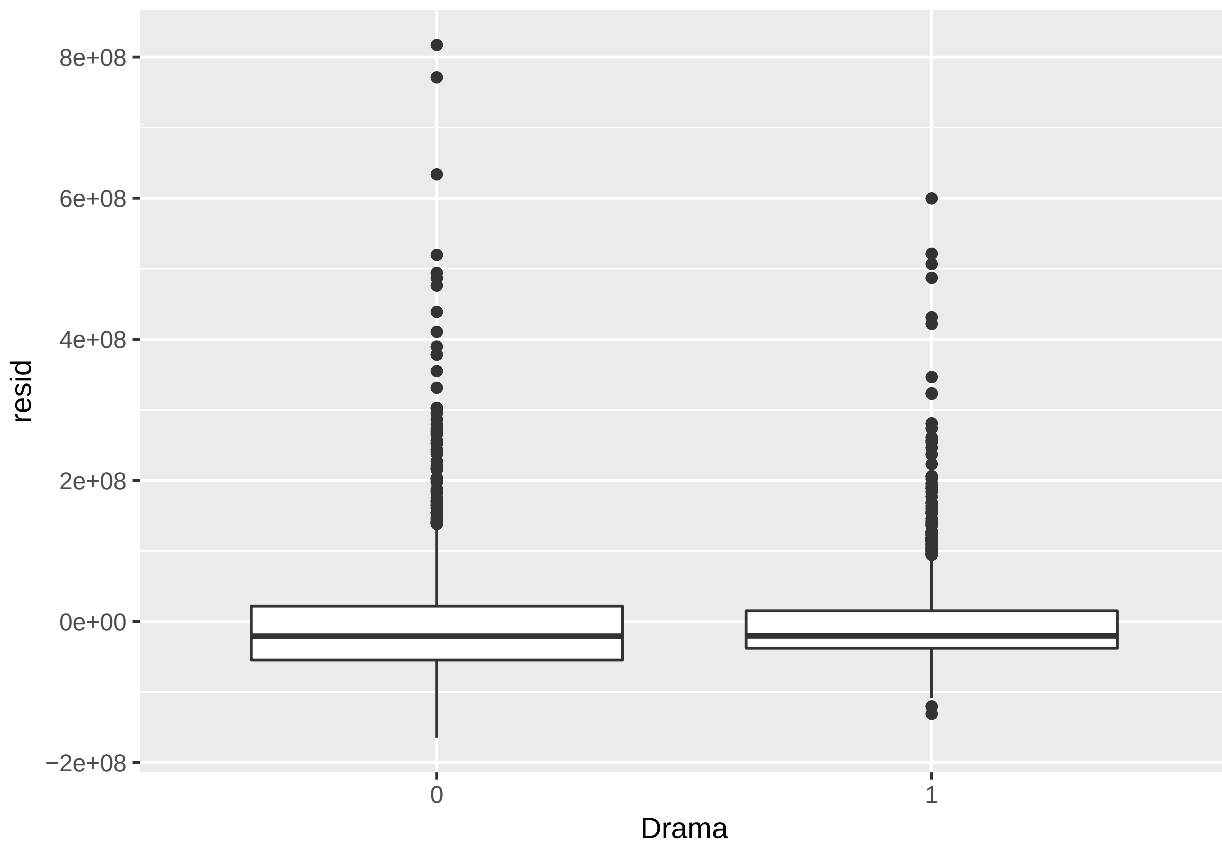

```
##
## [[5]]
```



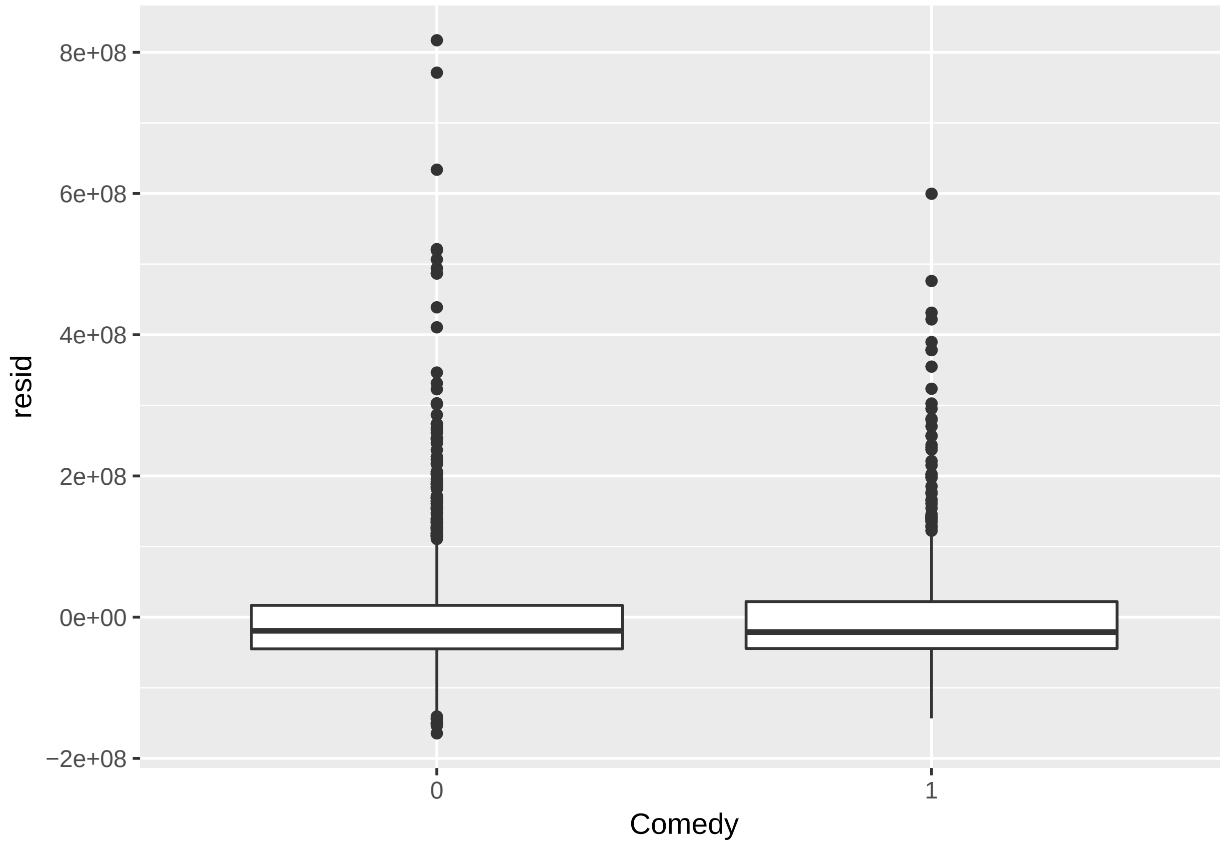

```
##
## [[6]]
```



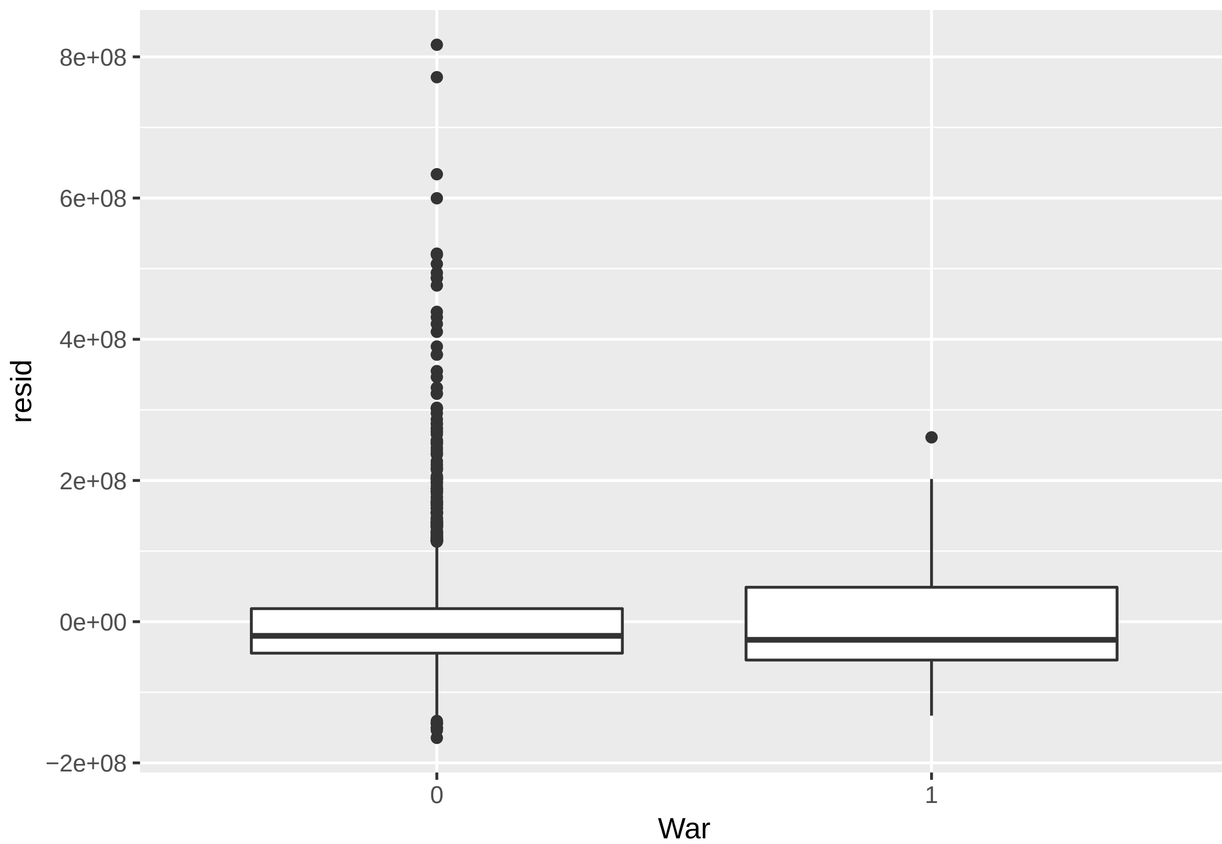

```
##
## [[7]]
```



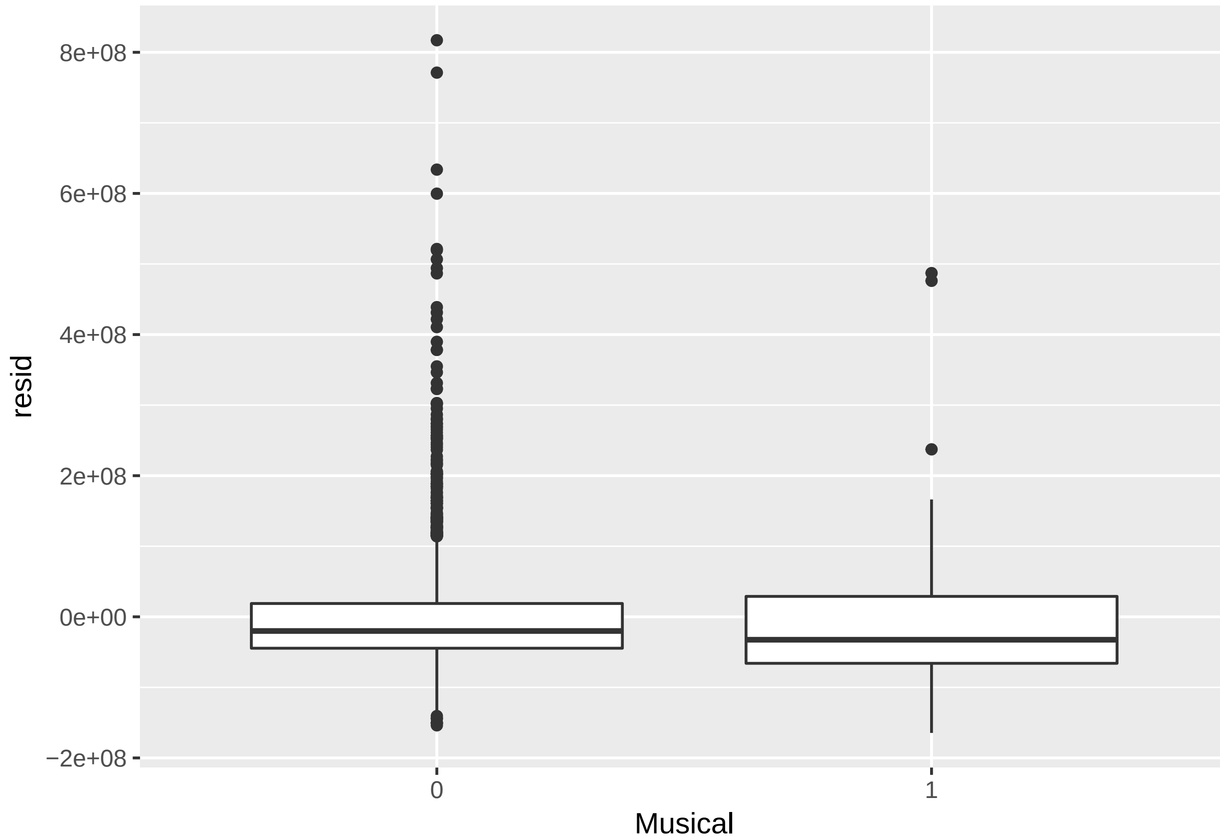

```
##
## [[8]]
```




```
##
## [[9]]
```

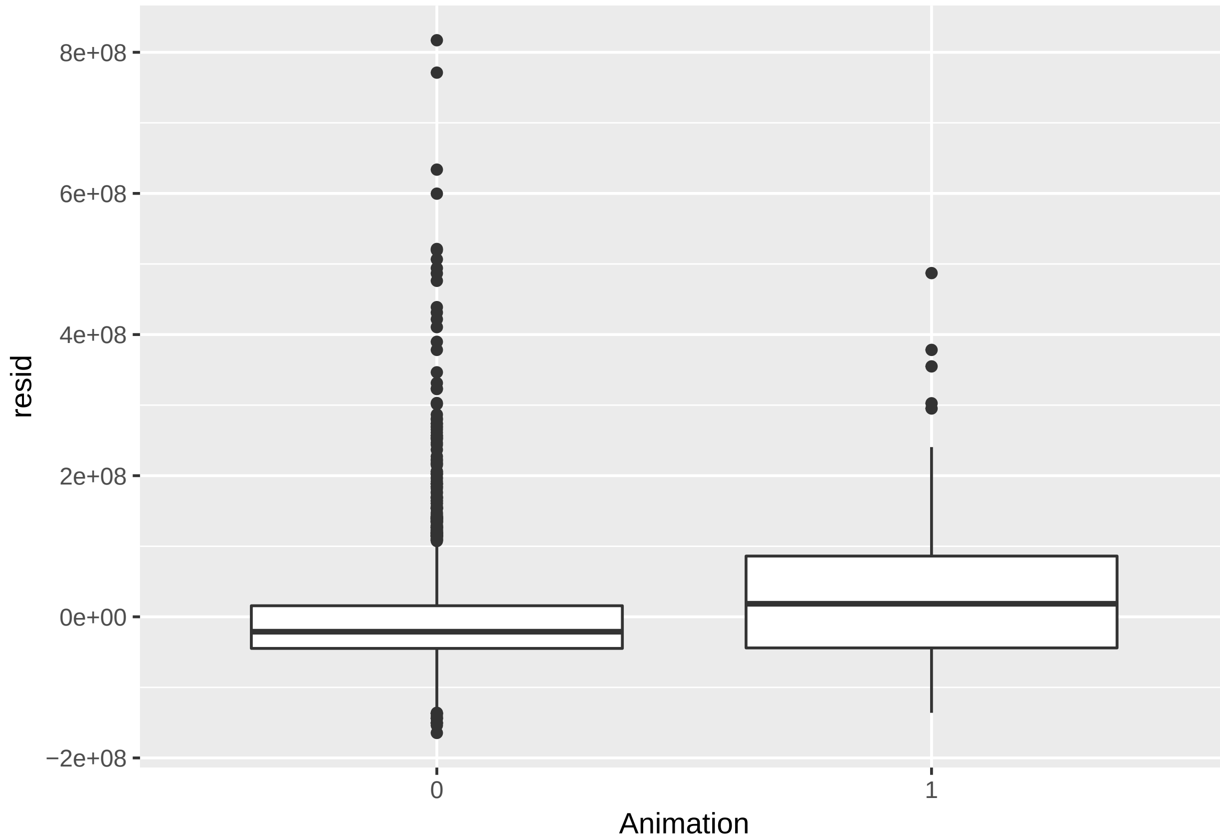



```
##
## [[10]]
```

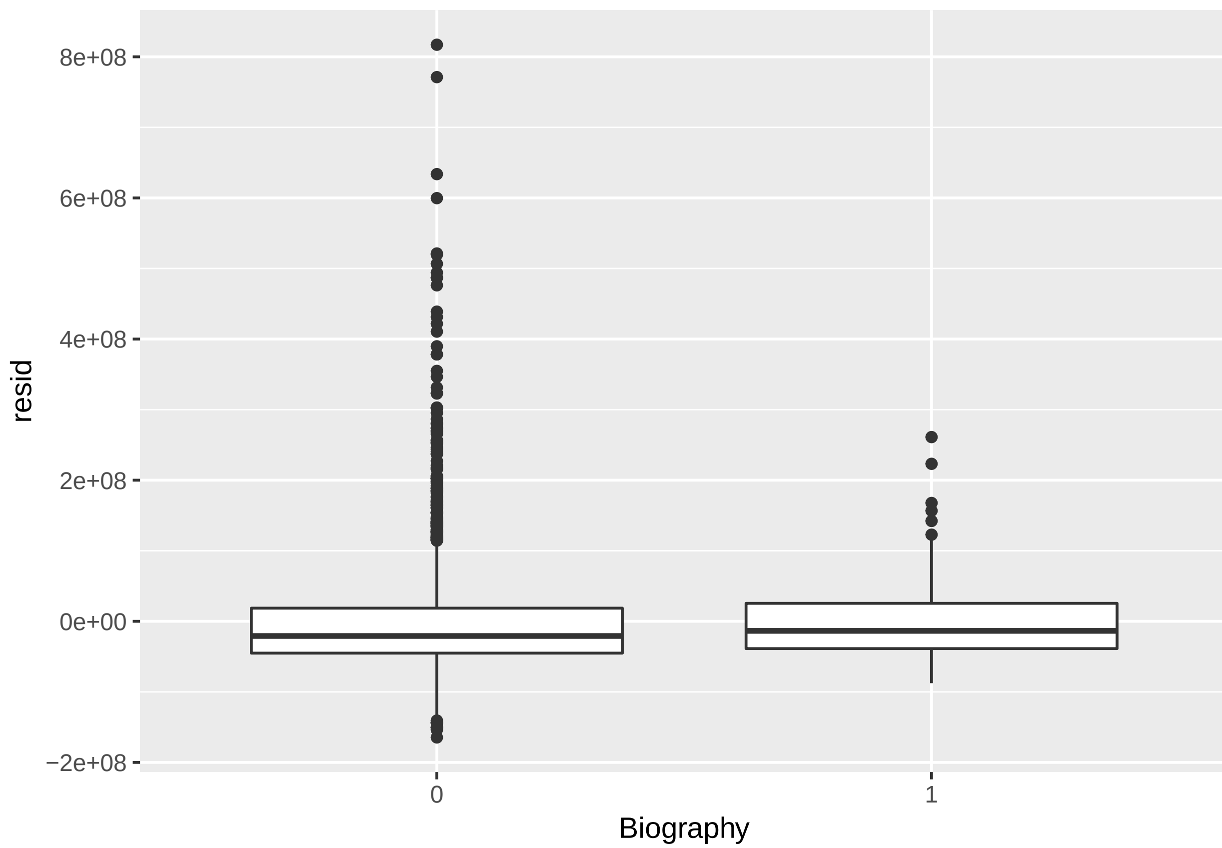



```
# graph residuals against each genre not included in the model
# several are questionable if random. Especially Animation.
lapply(names(train_genre %>% select(-genre_xvar)), function(var) {
  train %>%
    ggplot() +
    geom_boxplot(aes_string(var, y = 'resid'))
})
```

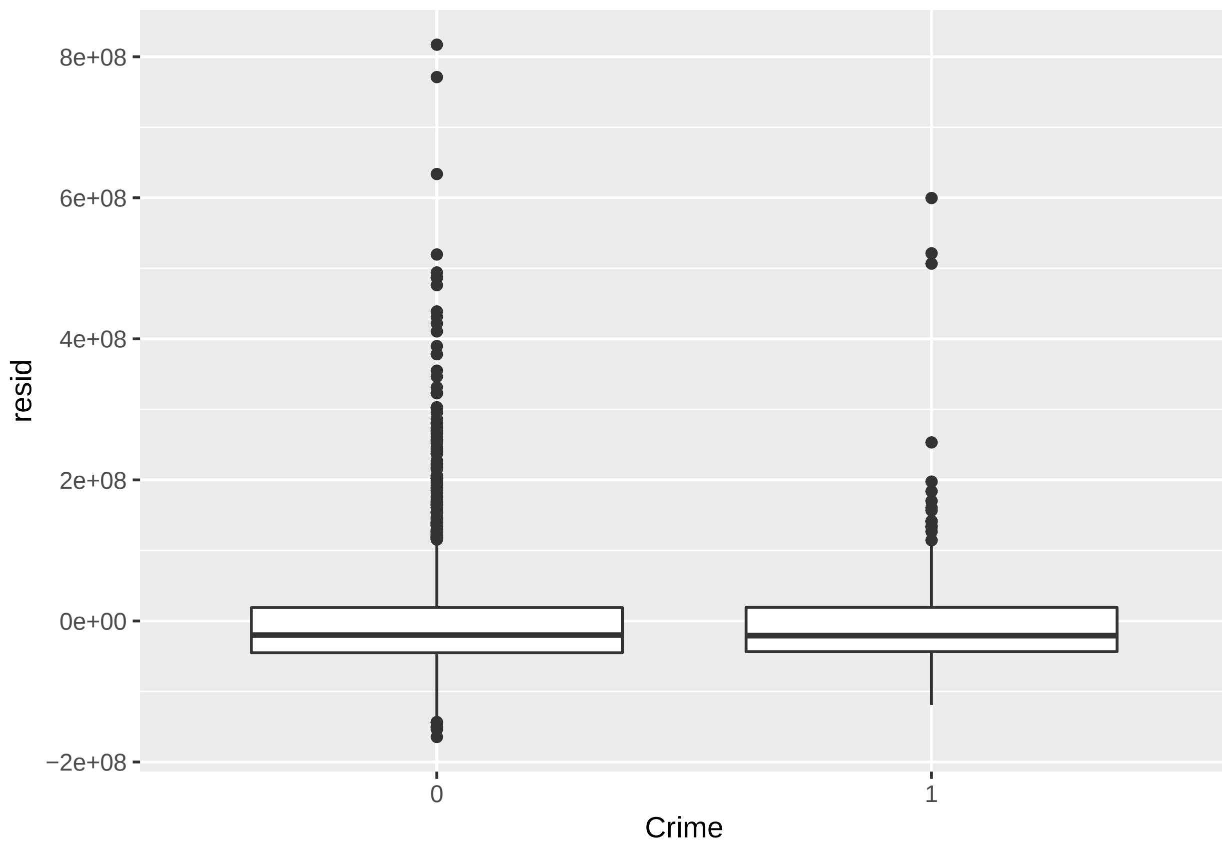
```
## [[1]]
```



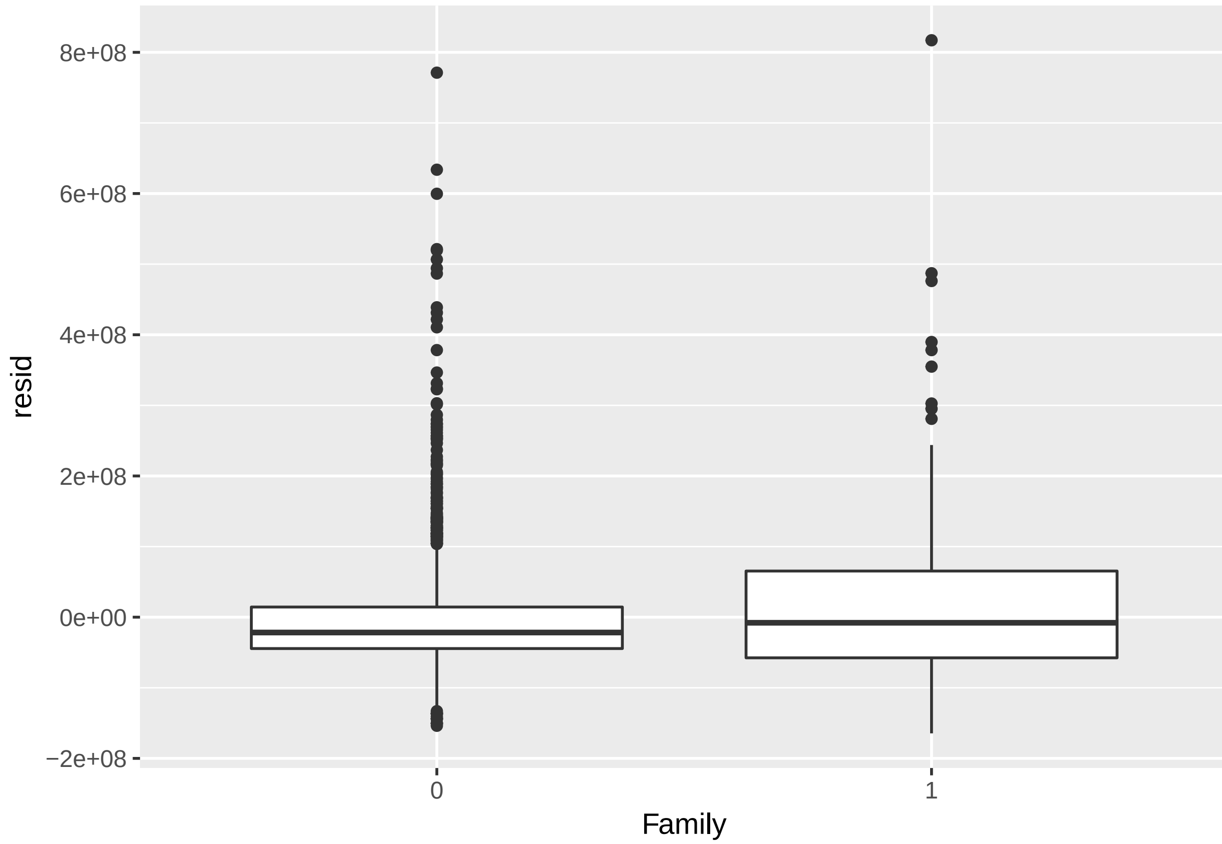

```
##
## [[2]]
```



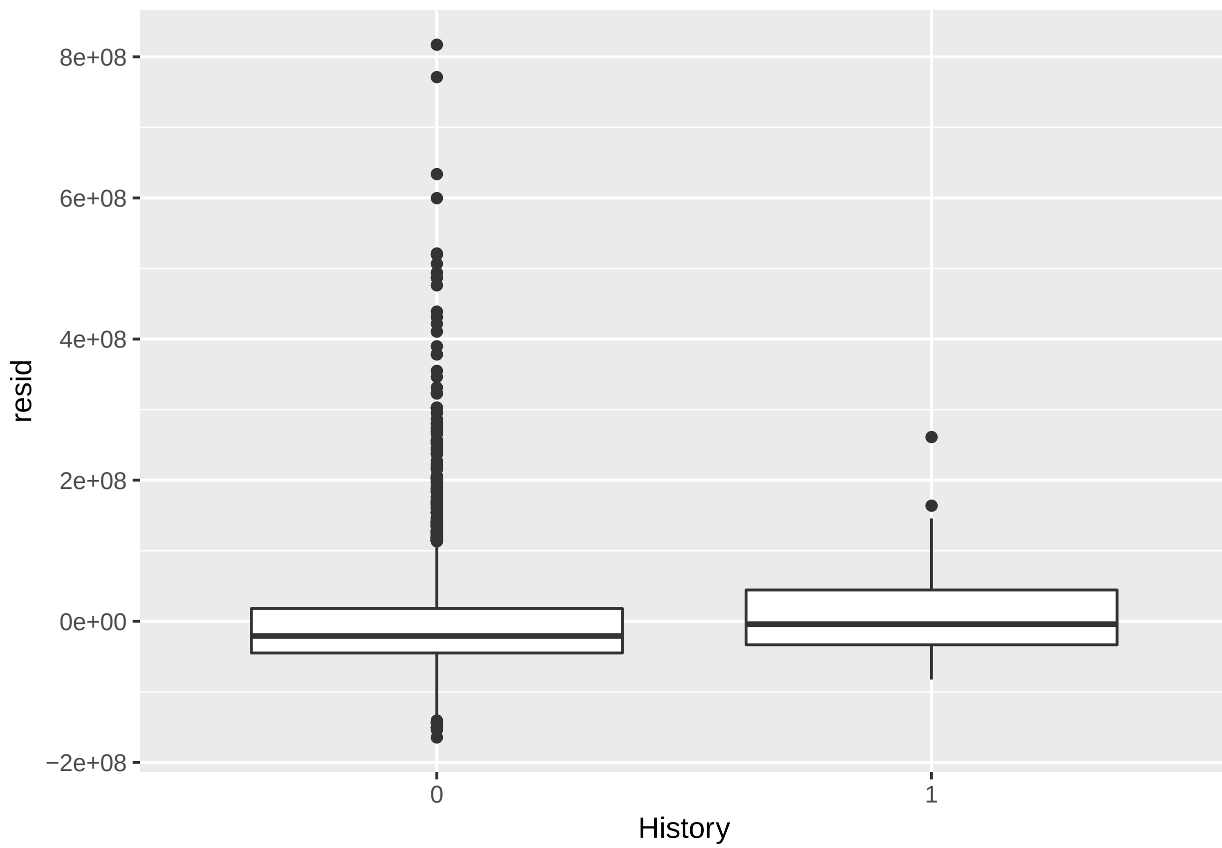

```
##
## [[3]]
```



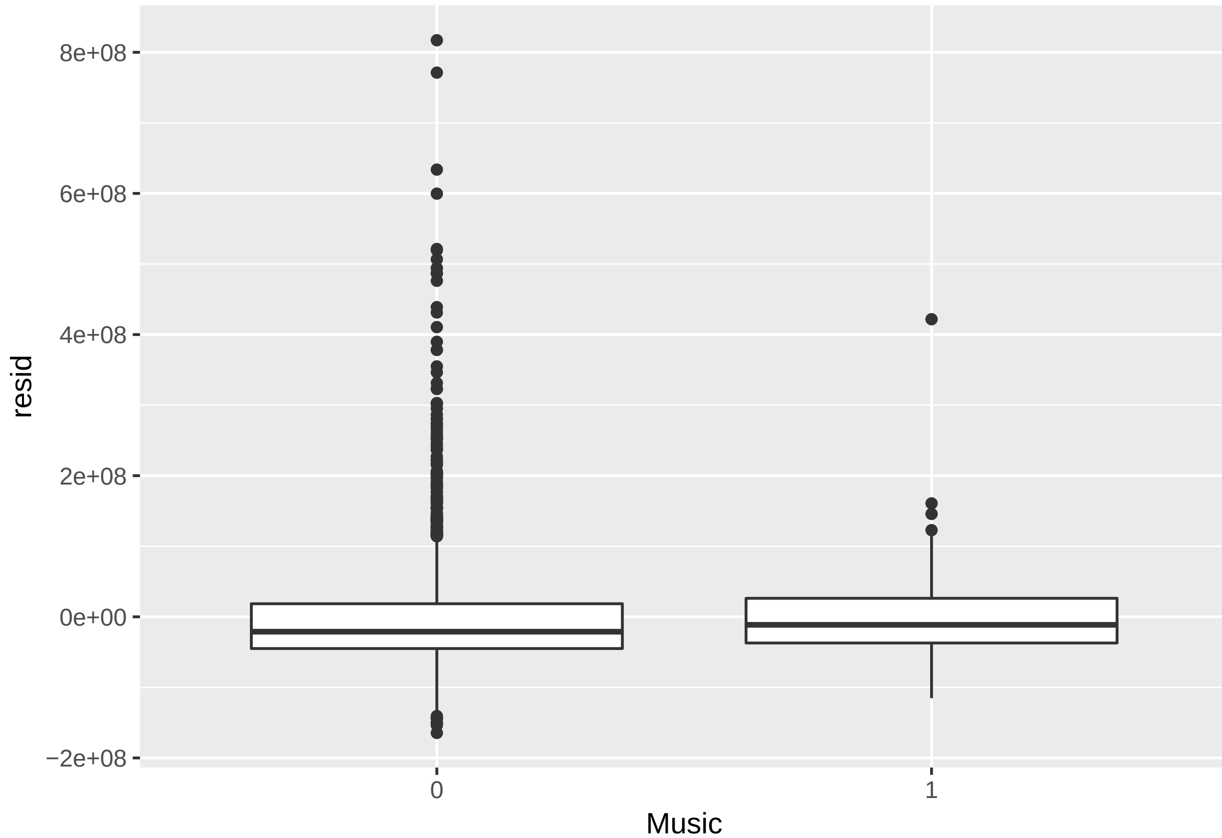

```
##
## [[4]]
```



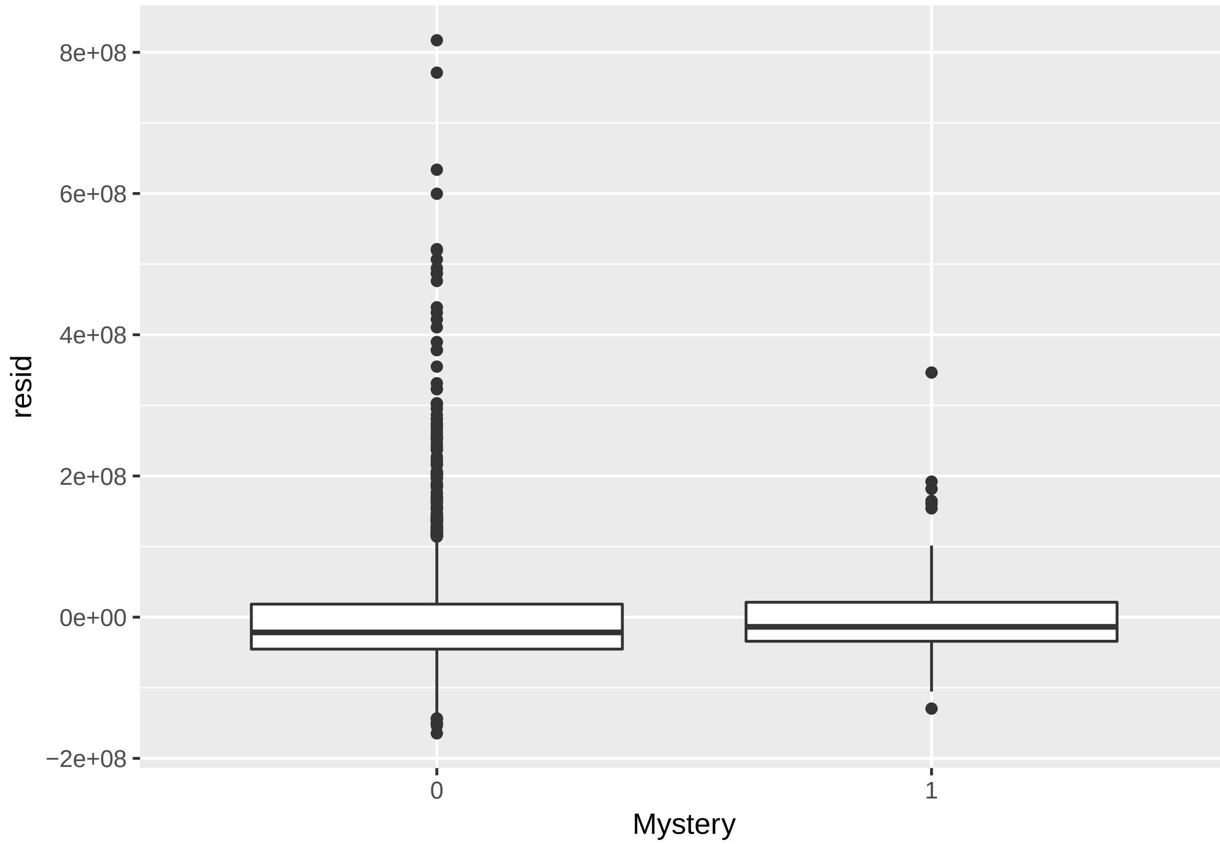

```
##
## [[5]]
```



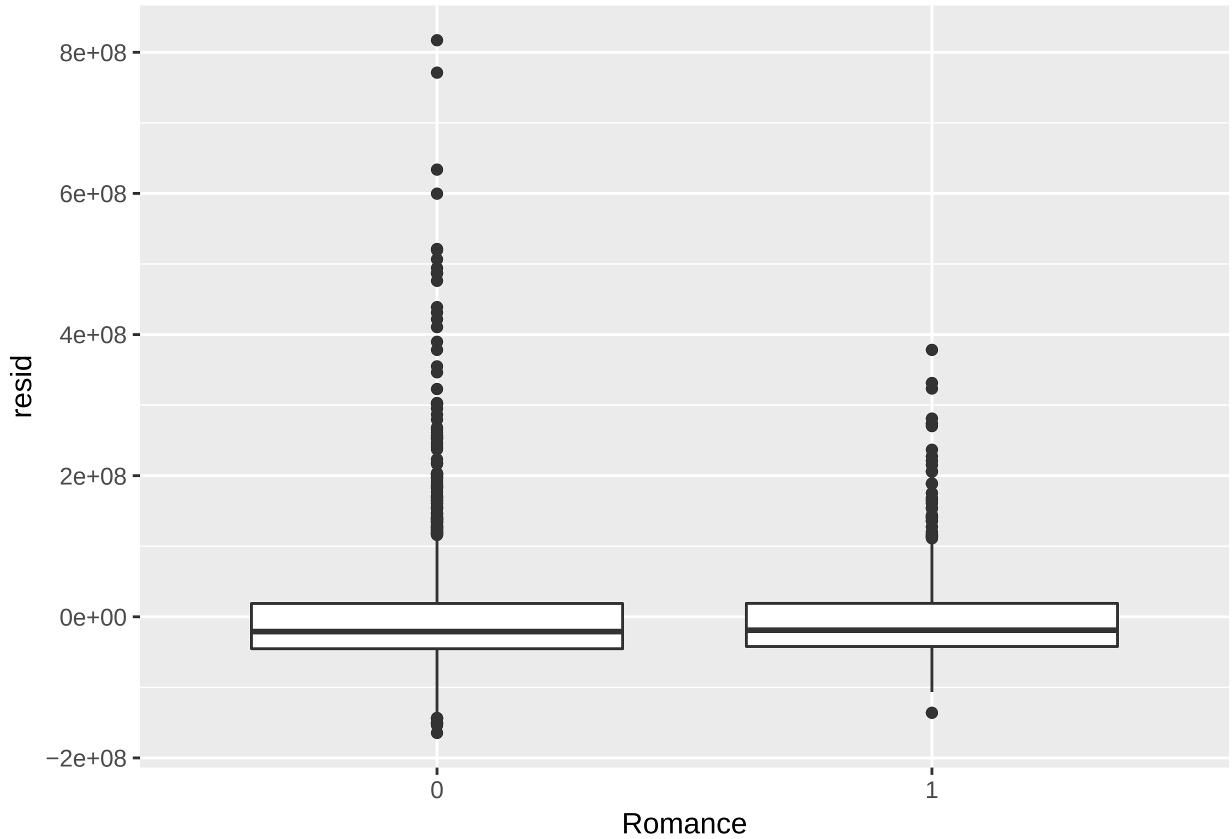

```
##
## [[6]]
```



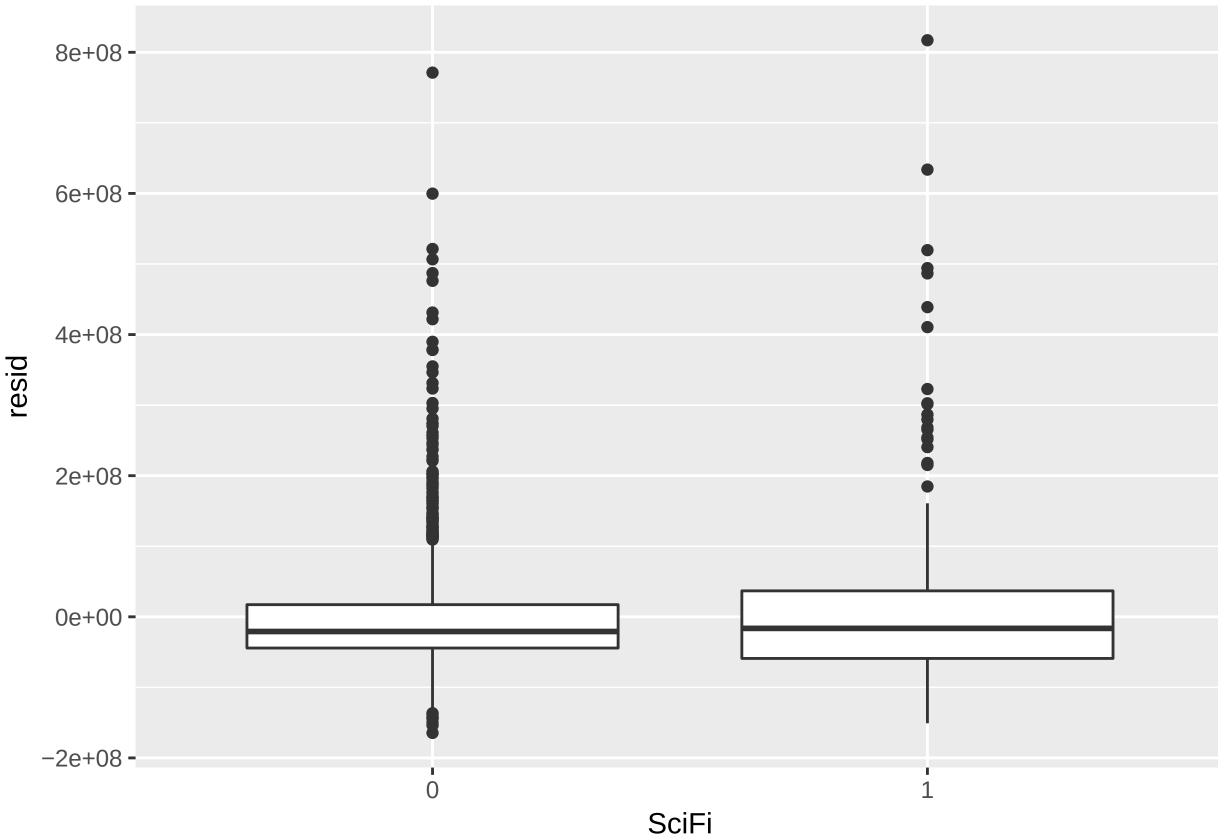

```
##
## [[7]]
```



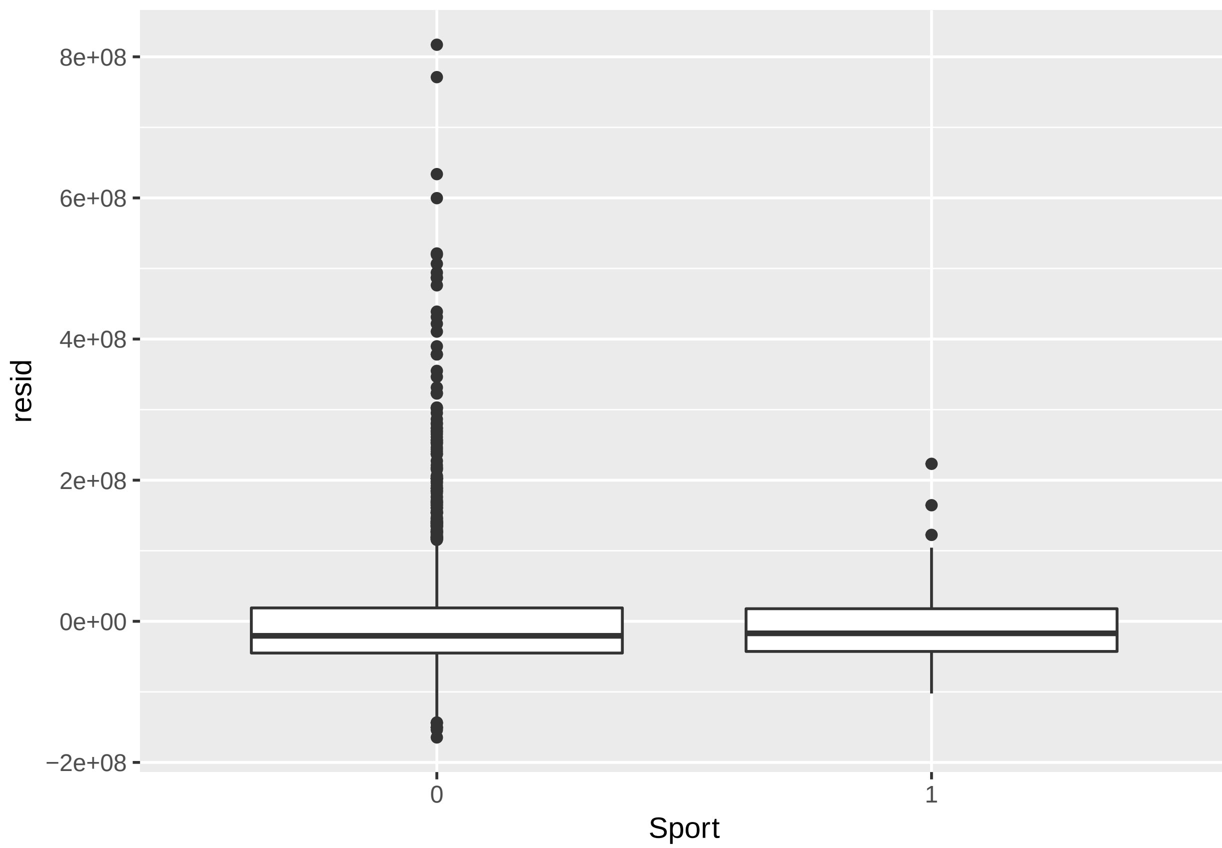

```
##
## [[8]]
```



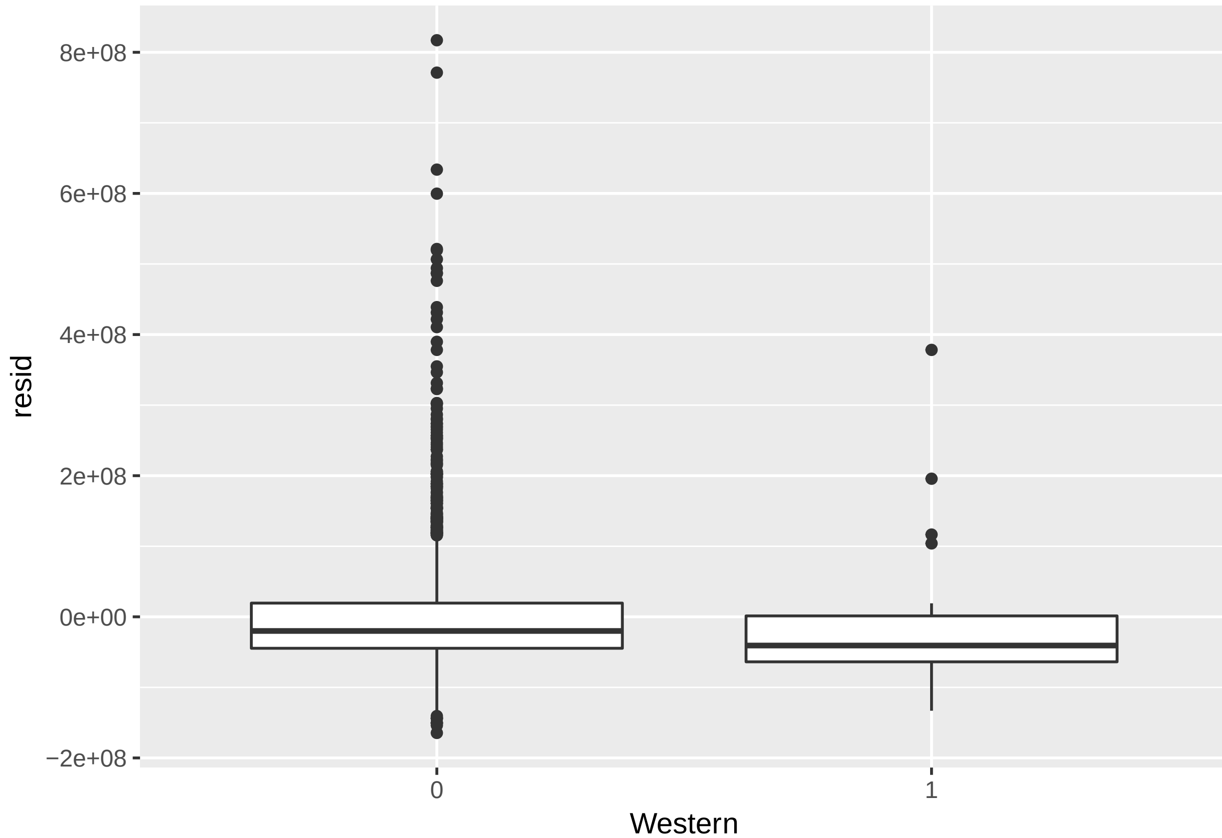

```
##
## [[9]]
```




```
##
## [[10]]
```

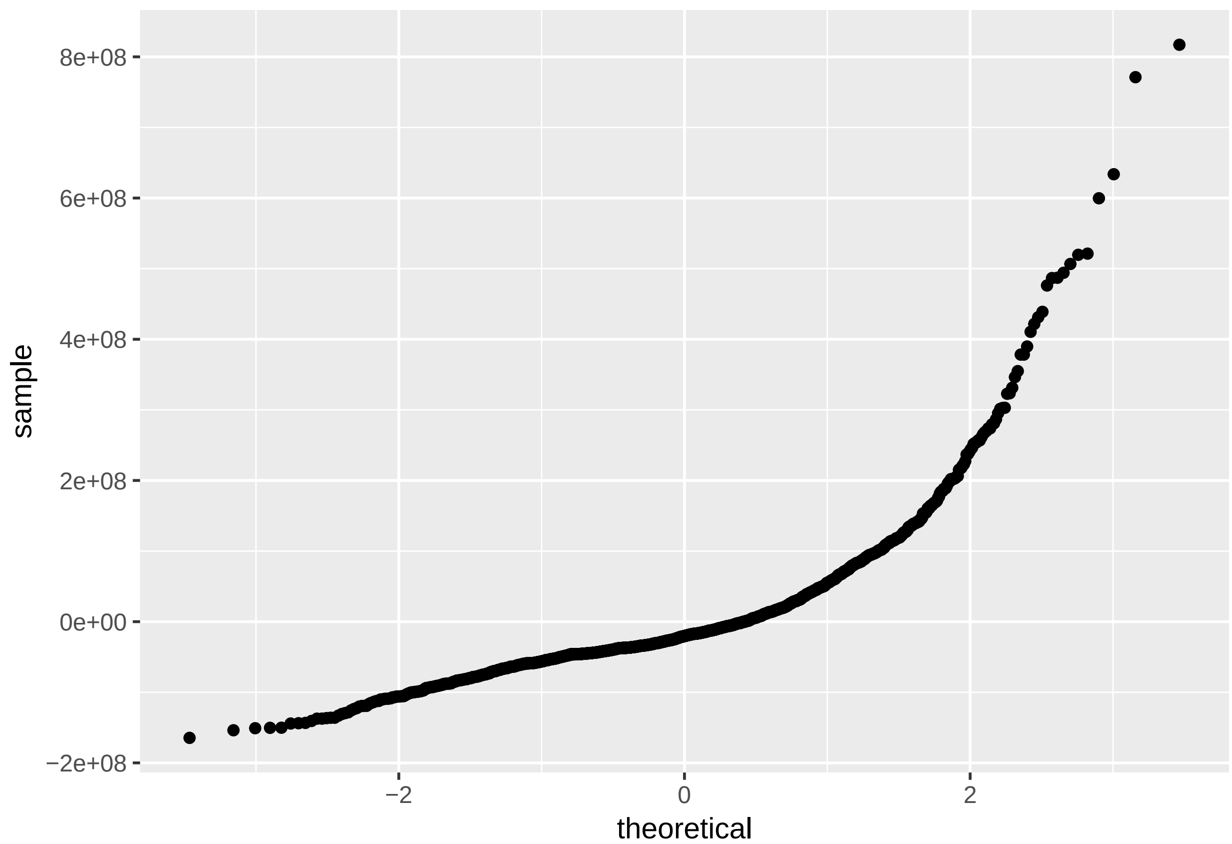



```
##
## [[11]]
```



Plot QQ plot for residuals. NOT normally distributed. Don't know what to do here.

```r
# residuals themselves are NOT normally distributed
# qq plot
train %>% ggplot() +
  geom_qq(aes(sample = resid))
```



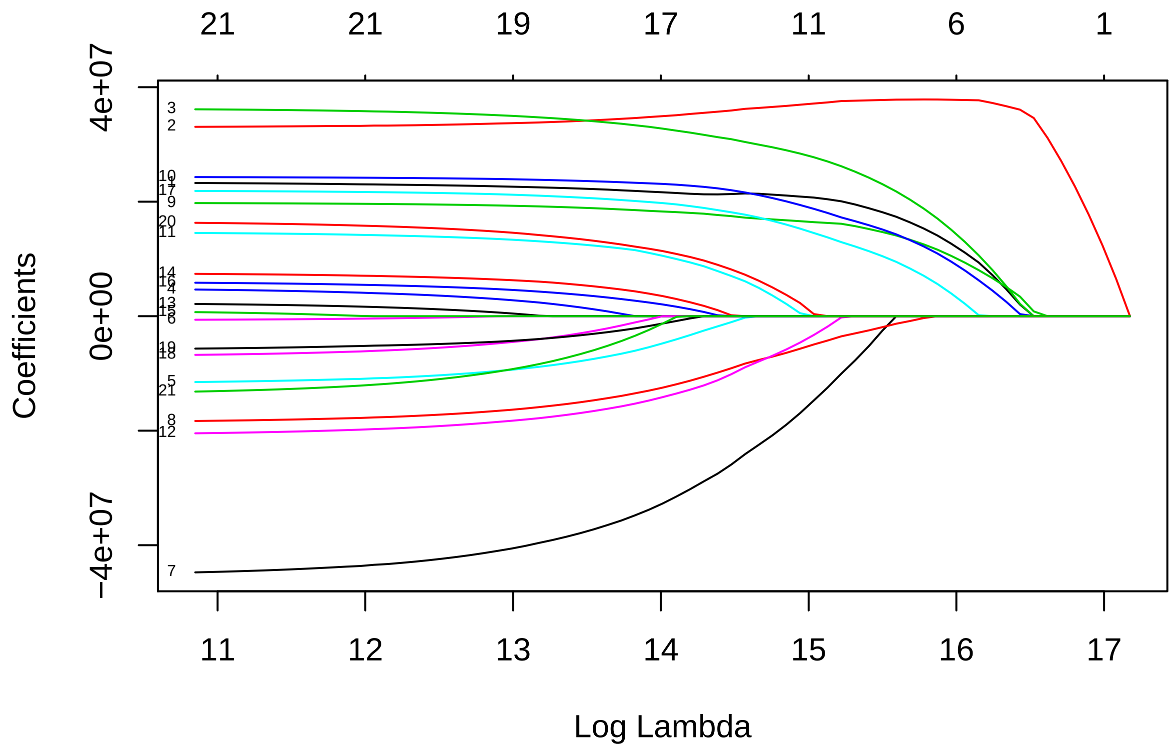
Glmnet: sparse

Quickly try this new method from class instead of stepwise.
However, doesn't eliminate many variables and can't do statistical testing, so not very useful.

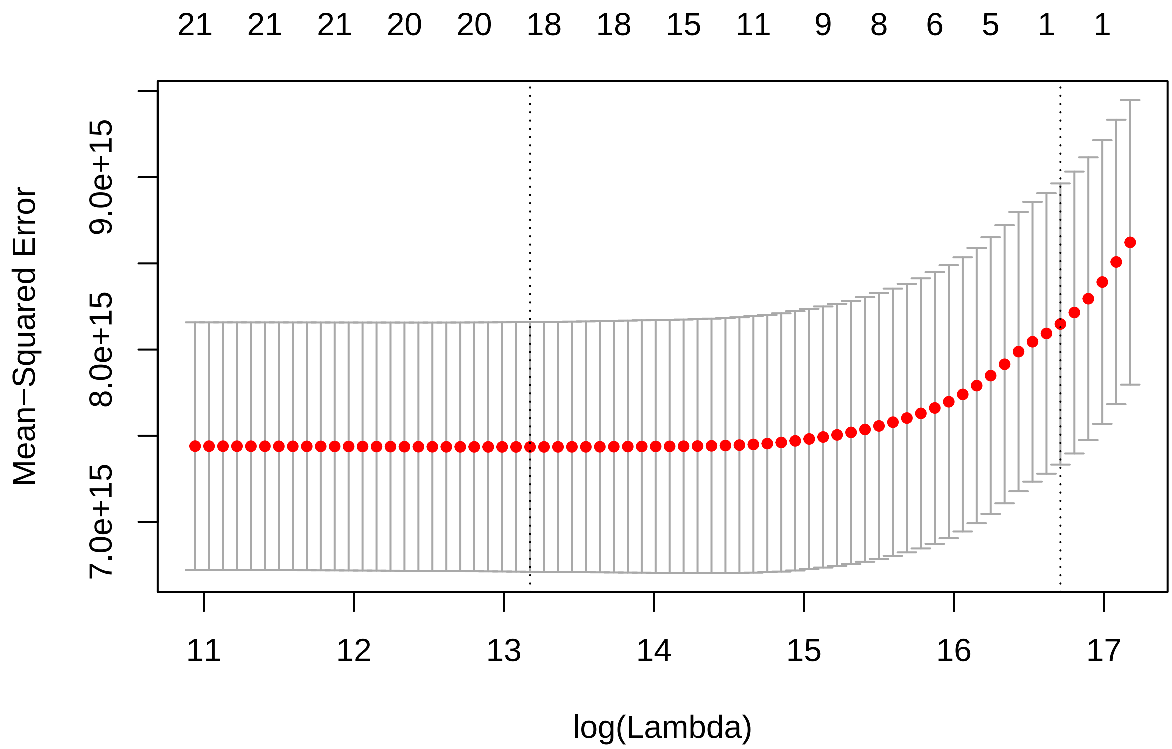
```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.5.3

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':
##
##     expand

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.5.3

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loaded glmnet 2.0-16
```

```r
# matrix of x and y variables
x <- as.matrix(train_genre)
y <- as.matrix(train$real_gross)

# glmnet process form class
mod <- glmnet(x, y, family = 'gaussian')
plot(mod, xvar = 'lambda', label = TRUE)
```




```r
mod2 <- cv.glmnet(x, y)
plot(mod2)
```




```r
coef(mod2, s = 'lambda.min') # use min lambda
```

```
## 22 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept)  58943362.0
## Action       22505933.7
## Adventure    33837527.8
## Animation    34732710.4
## Biography     2388860.3
## Comedy       -8846134.9
## Crime                 .
## Documentary -39606912.8
## Drama       -15886006.5
## Family       19175705.4
## Fantasy      23821138.7
## History      13081609.7
## Horror      -17817886.8
## Music           86457.8
## Musical       6037249.1
## Mystery               .
## Romance       4295040.6
## SciFi        21049072.9
## Sport        -4021512.4
## Thriller     -3988151.1
## War          14177477.2
## Western      -8369304.1
```

```r
coef(mod2, s = 'lambda.1se') # use most sparse
```

```
## 22 x 1 sparse Matrix of class "dgCMatrix"
##                    1
## (Intercept) 61603560
## Action             .
## Adventure   27089522
## Animation          .
## Biography          .
## Comedy             .
## Crime              .
## Documentary        .
## Drama              .
## Family             .
## Fantasy            .
## History            .
## Horror             .
## Music              .
## Musical            .
## Mystery            .
## Romance            .
## SciFi              .
## Sport              .
## Thriller           .
## War                .
## Western            .
```