Mellow Channel Views Analysis

Sage P.

For this analysis I was interested in extracting variables from Mellow Climbing's Youtube channel in order to make inference on what may effect views a video gets and to see if I can predict views a video will get. This is interesting as it potentially shows what people like seeing in the bouldering (and rope to some extent) world and I suppose could be of use in a business sense to Mellow or any other climbing channel looking for views if they are into that.

Mellow is a channel (and brand) ran by professional boulderers who wanted a place to post footage of hard climbing with a sentiment of "by climbers for climbers". It has a combination of bouldering and sport climbing, short "uncut"footage of sends and longer films of either the progress on a hard climb, multiple climbs, or multiple people with any combination. I thought it would be interesting to see if I can extract some basic info about the video and see if I can predict if a video will get a lot of views or not.

To start, I used Youtubes API to access Mellows videos, likes/comments/views, titles, description and date posted. I then did feature extraction to create potentially useful variables such as vgrade, uncut or not, gender of the climber, sport or boulder, time since video was posted and more. This was done in September 2023 and therefore only includes data from videos up until that point.

```
library(tidyverse)
library(psych)
library(corrplot)
library(patchwork)
library(randomForest)
library(Metrics)

raw <- read_csv("/Users/sagepletka/Documents/GitHub/mellow.analysis/channeldata.csv")


head(raw)
```

```
## # A tibble: 6 x 13
##   title     description viewCount likeCount commentCount date       uncut vgrade
##   <chr>     <chr>           <dbl>     <dbl>        <dbl> <date>     <chr> <chr>
## 1 "\"a val~ "Daniel Wo~     77658      3129          156 2023-01-04 no    <NA>
## 2 "uncut: ~ "Hugo Parm~     30005       718           41 2020-06-18 yes   <NA>
## 3 "antigra~ "Brainwash~     31397      1317           68 2023-07-10 no    <NA>
## 4 "uncut: ~ "Different~     83499      3435          189 2023-06-09 yes   v16
## 5 "blade r~ "Giuliano ~     90891      1777           86 2019-07-08 no    v15
## 6 "isabell~ "Isabelle ~     47779      1495           70 2020-09-07 no    v14
## # i 5 more variables: length_film <chr>, gender <chr>, group <chr>,
## #   climb_type <chr>, multi_climbs <chr>
```

Summary Stats on views:

```
##    vars   n     mean       sd median  trimmed      mad   min    max  range skew
```

```
## X1    1 277 77027.38 85845.01  48869 59038.04 29853.63 11795 699363 687568 3.66
##    kurtosis      se
## X1    17.27 5157.93
```

Data preparation:

```
prep <- raw %>% mutate(uncut = if_else(uncut == "no", 0, 1)) %>%
  mutate(vgrade = as.numeric(str_sub(vgrade, 2, 3))) %>%
  mutate(vgrade = if_else(is.na(vgrade), 1, vgrade)) %>%
  mutate(vgrade = as.factor(vgrade)) %>%
  mutate(length_film = if_else(length_film == "no", 0, 1)) %>%
  mutate(gender = case_match(gender, "Male" ~ 1,
                             "Female" ~ 2,
                             "Mix" ~ 3)) %>%
  mutate(gender = as.factor(gender)) %>%
  mutate(sport = if_else(climb_type == "sport", 1, 0)) %>%
  mutate(boulder = if_else(climb_type == "boulder", 1, 0)) %>%
  mutate(multi_climbs = if_else(multi_climbs == "no", 0, 1)) %>%
  mutate(group = if_else(group == "solo", 0, 1)) %>%
  mutate(days_since_upload = if_else(!is.na(date), as.numeric(as.Date("2023-10-01")-date), NA))
```

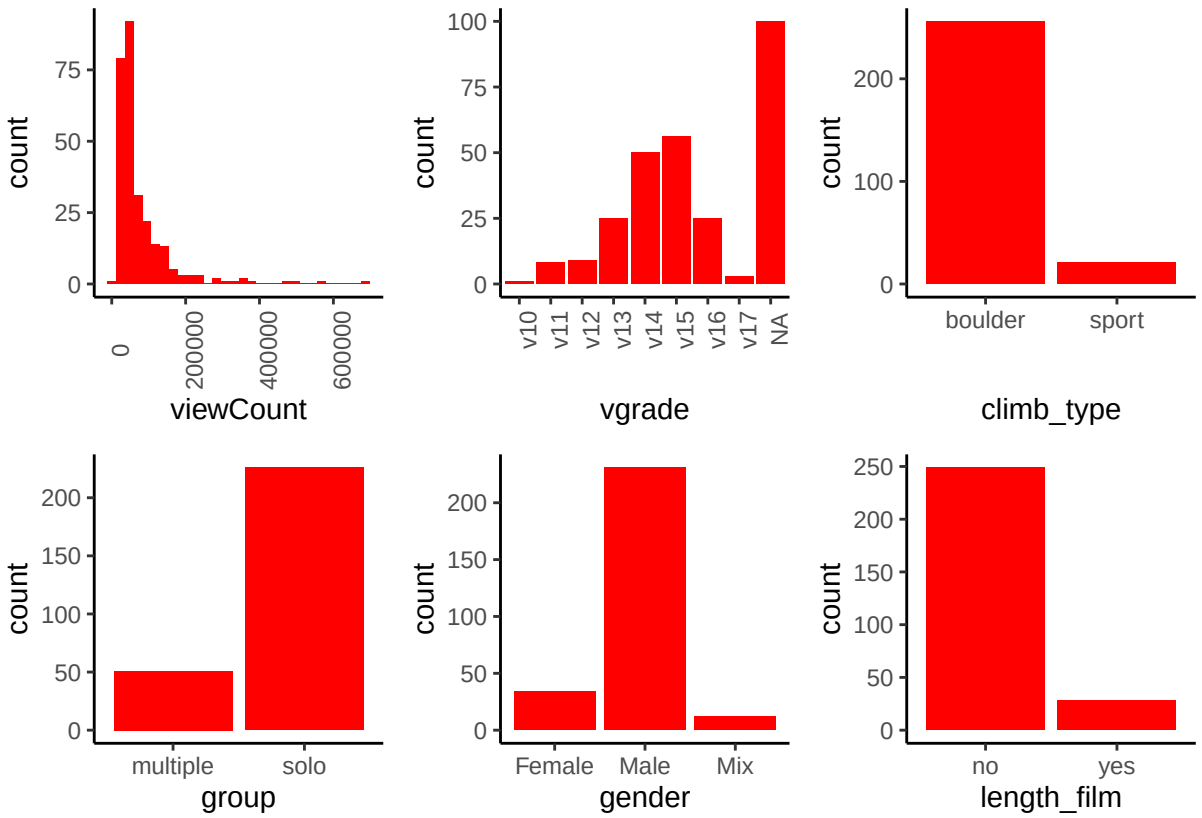
Independent variables used:

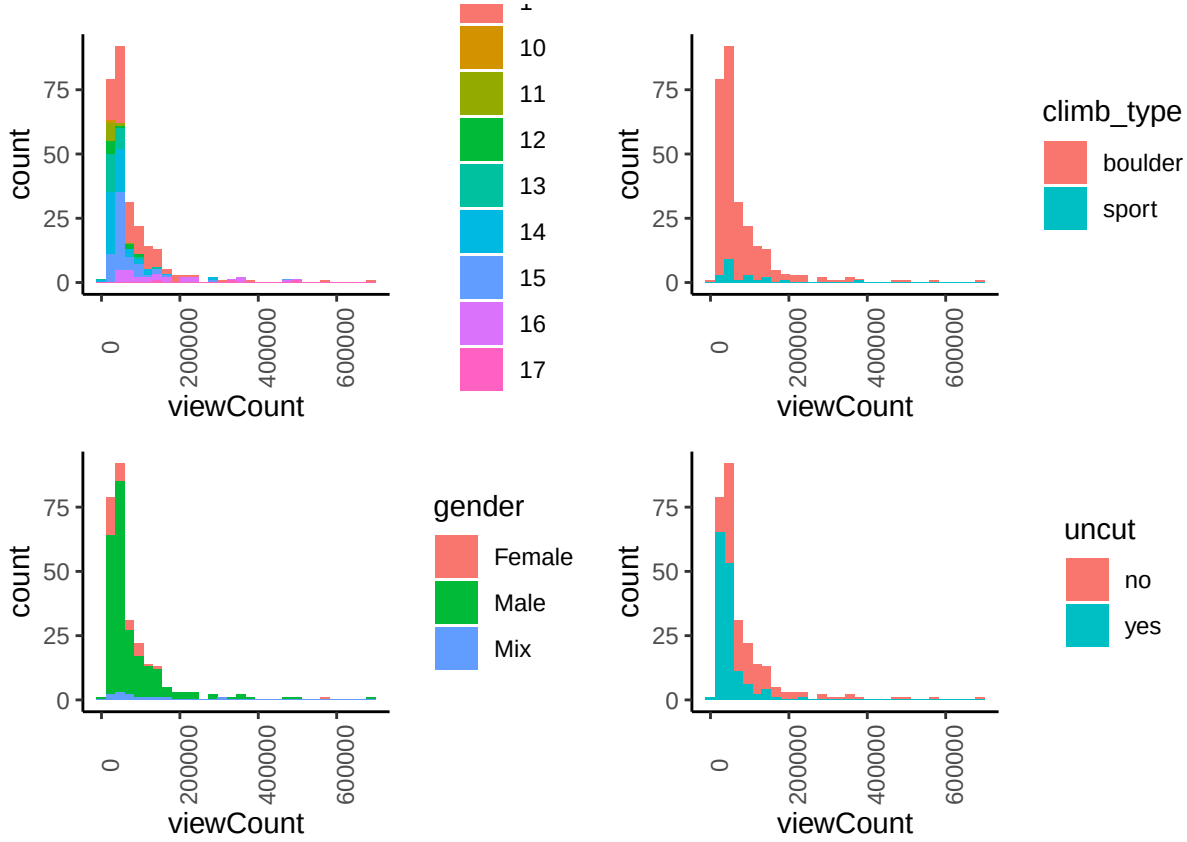
```
## [1] "uncut"             "vgrade"            "length_film"
## [4] "gender"            "group"             "climb_type"
## [7] "multi_climbs"      "days_since_upload"
```

Correlations between likes/views/comments were expectantly very high:

```
##              viewCount likeCount commentCount
## viewCount     1.000000 0.9245870    0.8281110
## likeCount     0.924587 1.0000000    0.9025076
## commentCount  0.828111 0.9025076    1.0000000
```

Some Data viz to better understand relationships between the dependents and independents:


count
75
50
25
0
0
200000
400000
600000
viewCount
100
75
50
25
0
v10
v11
v12
v13
v14
v15
v16
v17
NA
vgrade
200
100
0
boulder
sport
climb_type
200
150
100
50
0
multiple
solo
group
Female
Male
Mix
gender
250
no
yes
length_film



With exception to days since video upload, all of my dependents are categorical variables which is going to make precise prediction or explanation of the variation in views not possible, however it may still be interesting to consider what variables are important and in what direction and magnitude, and to what extent we could predict views.

To make inference on the independent variables, I used linear regression using the poisson link function for the count data with a heavy right skew

```
all_pois <- glm(viewCount ~ uncut + gender + sport + days_since_upload + group +
                  multi_climbs + length_film + vgrade,
                data = prep, family = "poisson")
```

```
##
## Call:
## glm(formula = viewCount ~ uncut + gender + sport + days_since_upload +
##     group + multi_climbs + length_film + vgrade, family = "poisson",
##     data = prep)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -363.33  -116.62   -37.69    52.00  1221.86
##
## Coefficients: (1 not defined because of singularities)
##                        Estimate    Std. Error  z value            Pr(>|z|)
## (Intercept)       11.2652078931  0.0007650547 14724.71 <0.0000000000000002 ***
## uncut             -0.7218982215  0.0005922159 -1218.98 <0.0000000000000002 ***
```

```
## gender2            0.4040746293  0.0008609002   469.36 <0.0000000000000002 ***
## gender3            0.0102565602  0.0010454736     9.81 <0.0000000000000002 ***
## sport              0.3268070962  0.0009092888   359.41 <0.0000000000000002 ***
## days_since_upload  0.0000656520  0.0000004786   137.19 <0.0000000000000002 ***
## group              0.1300984128  0.0006788965   191.63 <0.0000000000000002 ***
## multi_climbs       0.2452392792  0.0006230604   393.60 <0.0000000000000002 ***
## length_film                  NA            NA       NA                   NA
## vgrade10          -0.4504942306  0.0063064069   -71.43 <0.0000000000000002 ***
## vgrade11          -0.6758998380  0.0023699664  -285.19 <0.0000000000000002 ***
## vgrade12          -0.3248524670  0.0019502343  -166.57 <0.0000000000000002 ***
## vgrade13          -0.2456459039  0.0012953374  -189.64 <0.0000000000000002 ***
## vgrade14          -0.1428136911  0.0009404655  -151.85 <0.0000000000000002 ***
## vgrade15          -0.0579499938  0.0007457426   -77.71 <0.0000000000000002 ***
## vgrade16           0.8566982238  0.0007571049  1131.55 <0.0000000000000002 ***
## vgrade17           1.4602501920  0.0011634576  1255.10 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 16325789  on 276  degrees of freedom
## Residual deviance:  9010182  on 261  degrees of freedom
## AIC: 9013748
##
## Number of Fisher Scoring iterations: 5
```

All variables were significant contributors to view count. Of note is the large negative effect of uncut videos and of lower v-grades which even out around v15 and then become positive at v16. Female climber videos do better than males. Both having multiple climbers and boulders have a small effect. Interestingly sport climbing does better than boulders, although the number of sport videos is smaller. Length film not computed as it is linearly dependent. The poisson regression coefficients can be interpreted as for each unit change in the independent variables, the expected dependent variable logged will change by that coefficient.

Next I ran both random forest regression and poisson glm regression to predict views using the above variables minus length film. 25% of the data was pulled as a test set for the models and models were compared by RMSE and variance explained.

Creating train and test sets:

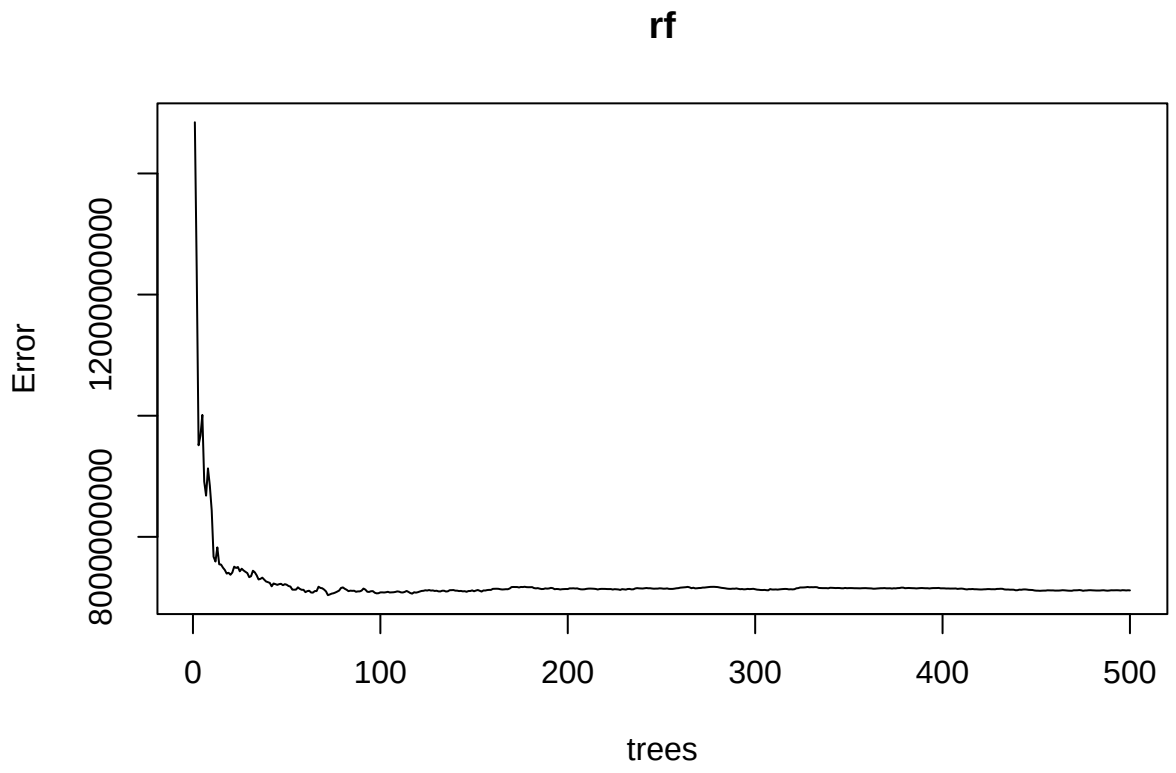
```
rf_data <- prep %>% select(viewCount,uncut, gender, climb_type, days_since_upload, group, multi_climbs,

set.seed(16)
sample_rf <- sample(1:nrow(rf_data), size = .75 * nrow(rf_data), replace = FALSE)
train_rf <- rf_data[sample_rf, ]
test_rf <- rf_data[-sample_rf, ]
```

Training and testing the random forest algorithm first with the default 500 trees

```
set.seed(500)
rf <- randomForest(viewCount ~ ., data = train_rf)
rf
```

```
##
## Call:
```

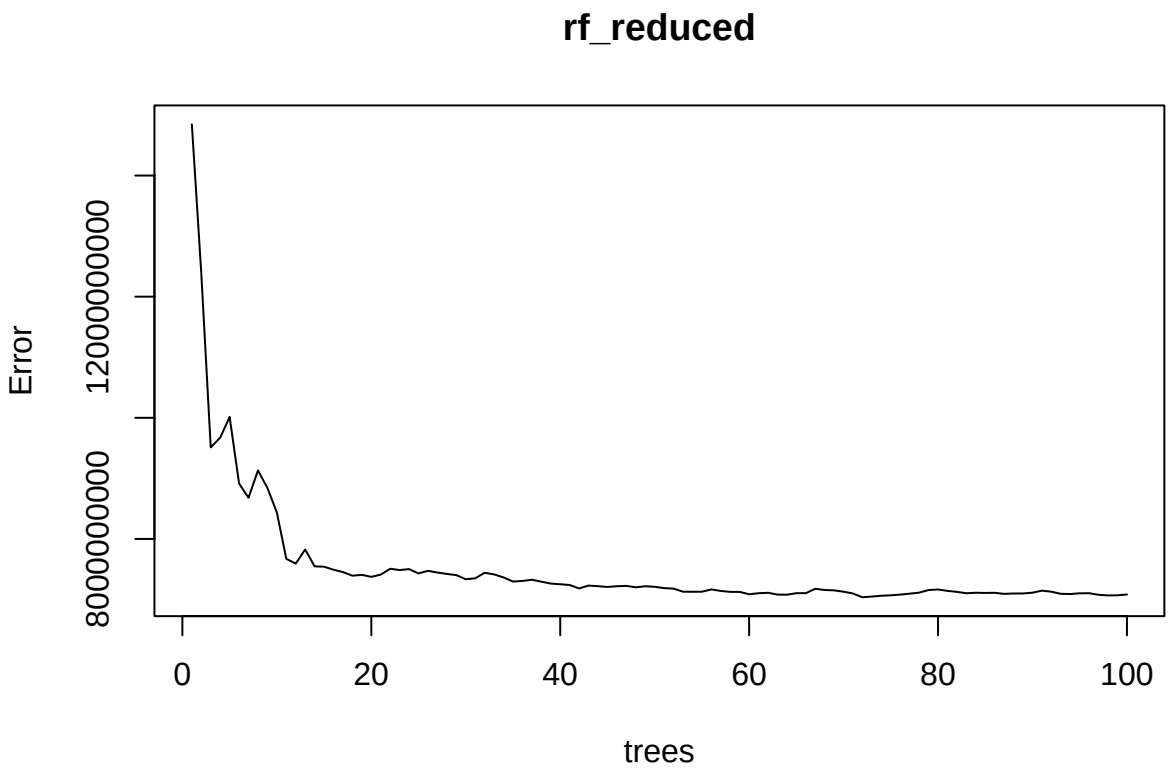
```
##  randomForest(formula = viewCount ~ ., data = train_rf)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 7115977950
##                     % Var explained: 13.62
```



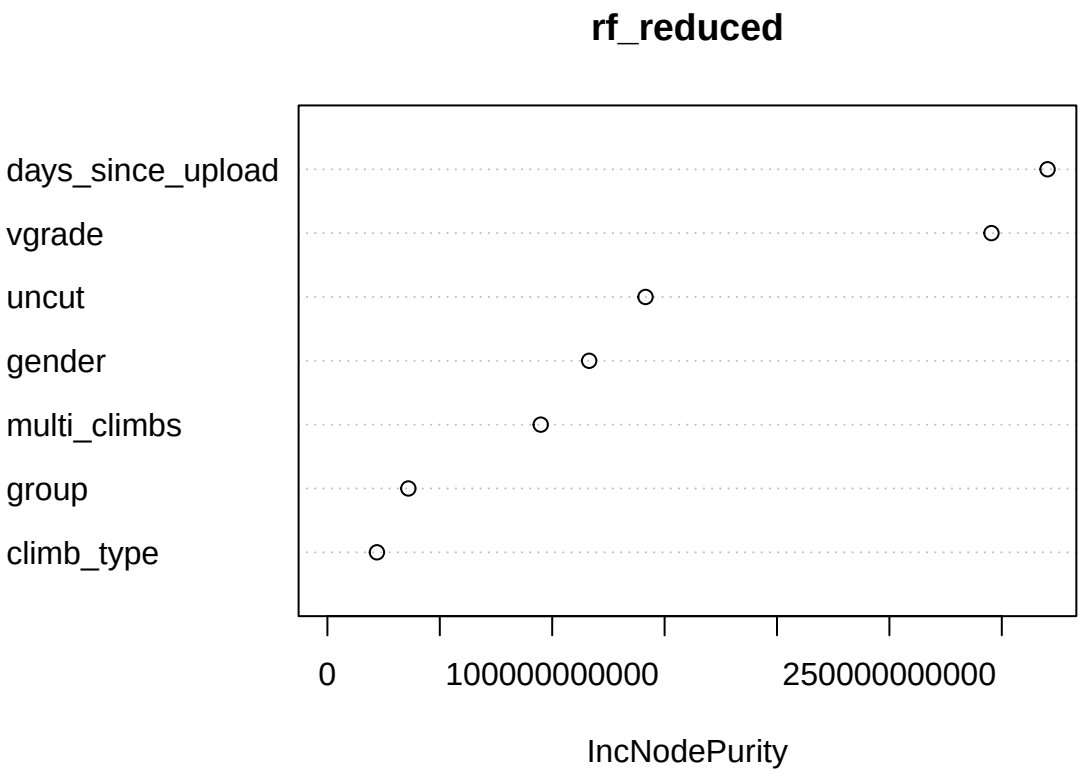
Reducing trees to 100

```
set.seed(500)
rf_reduced <- randomForest(viewCount ~ ., data = train_rf, ntree = 100)
rf_reduced
```

```
##
## Call:
##  randomForest(formula = viewCount ~ ., data = train_rf, ntree = 100)
##                Type of random forest: regression
##                      Number of trees: 100
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 7082621244
##                     % Var explained: 14.03
```



Variable importance in the random forest algorithm with 100 trees



Next I trained the poisson model from the training set

```
poisson_train <- glm(viewCount ~ ., data = train_rf, family = "poisson")
```

Running the models on the test set and computing RMSE:

```
#random forest
test_num <- tibble(predict(rf_reduced, test_rf[ , -1]))
comp <- test_num %>% mutate(actual = test_rf[ , 1])
mean_test_rf <- sum(comp$actual)/nrow(comp$actual)
rss_rf <- sum((comp$actual - comp$`predict(rf_reduced, test_rf[, -1])`)^2)
tss_rf <-sum((comp$actual - mean_test_rf)^2)
actual_rf <- pull(comp$actual)
predicted_rf <- comp$`predict(rf_reduced, test_rf[, -1])`
rmse_rf <- rmse(actual_rf, predicted_rf)

#poisson glm
pois_test <- tibble(predict(poisson_train, test_rf[ , -1], type = "response"))
pois_compare <- pois_test %>% mutate(actual = test_rf[ , 1])
actual_pois <- pull(pois_compare$actual)
predicted_pois <- pois_compare$`predict(poisson_train, test_rf[, -1], type = "response")`
rmse_pois <- rmse(actual_pois, predicted_pois)
```

Random forest RMSE:

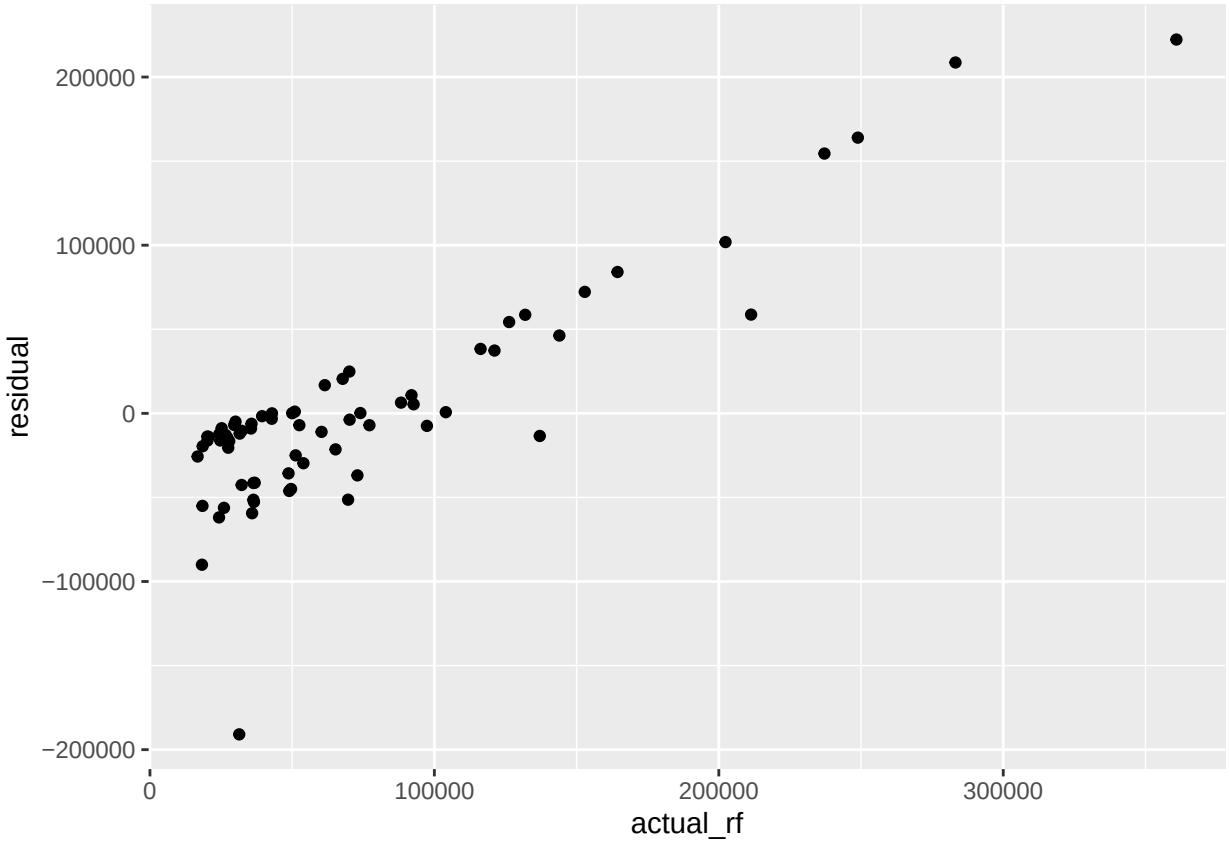
```
## [1] 61780.22
```

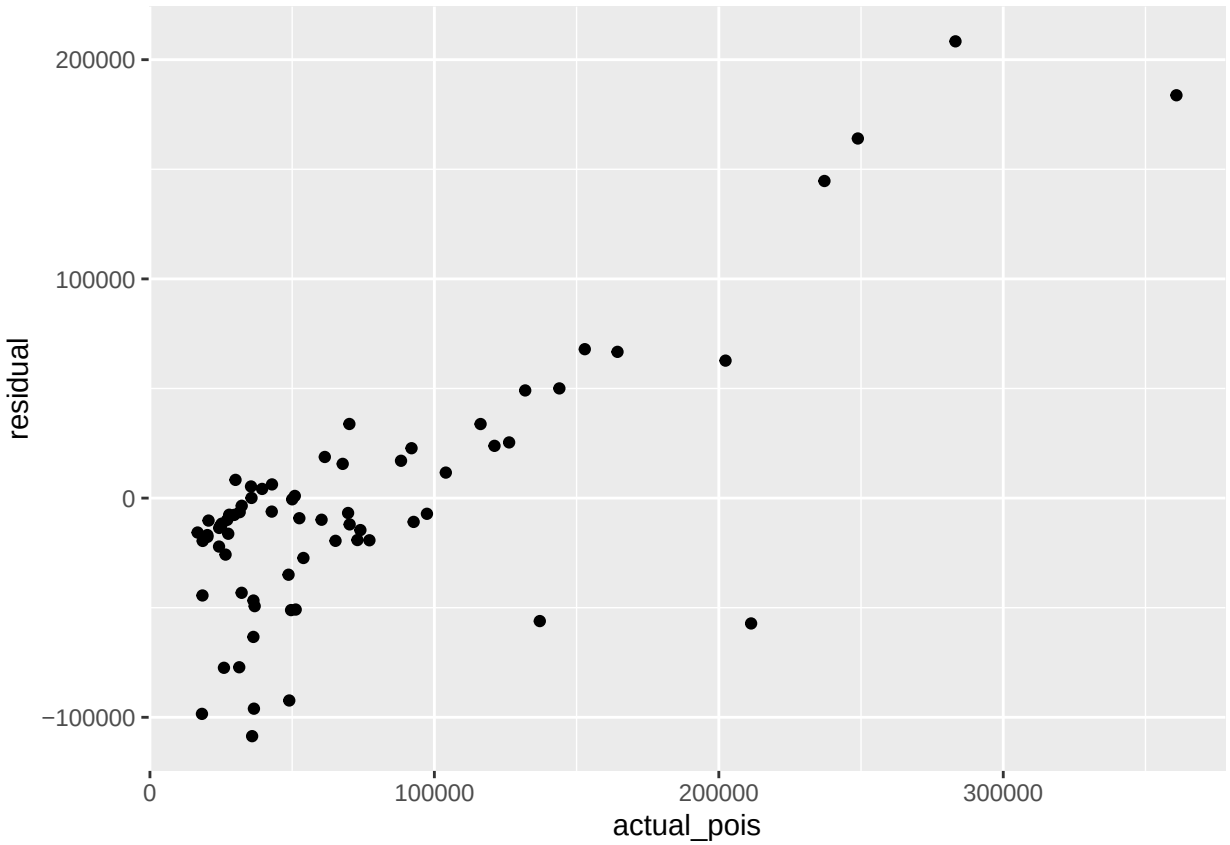
Poisson GLM RMSE:

```
## [1] 57435.88
```

Both are quite high and similar, thought the random forest algorithm is just barely better. The RMSE was nearly the standard deviation of views (85845)

I was also interested to understand the spread of residuals to see if there was patterned bias





Both models tended to overestimate at lower levels of the views and underestimate at higher levels. This suggests that to some extent neither model was a good reflection of the data. This may be because of sparce nature of some of the levels of various independent variables, and potentially the slight over dispersion for the poisson regression. Additionally, there was only so much power in my independent variables as most were nominal data with few levels.

In conclusion, there was some prediction power in things like grade sent, who sent, what type of video editing was used, what type of climb, and obviously how long it has been since the video had been uploaded. These variables did not explain much of the data however (15% according to the random forest output) which is to be expected to some degree. I was not surprised that boulders under v16 got less than over and that uncut videos don't do very well. I was a bit more surprised that sport climbs do generally well considering the channel and owners are known more as boulderers.

It might have been interesting to extract/use more variables such as home country of climber and even use Instagram follower numbers of the climbers in the videos as one could conceive that people with bigger followings and from countries that consume a lot of climbing media would tend to have higher views. This might help the the non-random residuals as well, as the residuals seemed to get higher with higher actual views.

Another thing to note is that I did not include interaction effects in my models. This was in part due to the small sample size and high number of variables already included. Interactions between vgrade and feature film or even interactions between those two and country of origin or climbers social media following would be interesting to try.