Practice Problem Set

Regression

[YOUR NAME]

Getting Set Up

Open `RStudio` and create a new RMarkDown file (`.Rmd`) by going to `File -> New File -> R Markdown...` . Accept defaults and save this file as `[LAST NAME]_psEC1.Rmd` to your `code` folder.

Copy and paste the contents of this file into your `[LAST NAME]_psEC1.Rmd` file. Then change the `author: [YOUR NAME]` (line 4) to your name.

All of the following questions should be answered in this `.Rmd` file. There are code chunks with incomplete code that need to be filled in.

This problem set is only for review purposes. You are not expected to complete this, and you will not be graded on it. Nevertheless, I encourage you all to at least look over this problem set, as it embodies the style of questions that will be asked on the midterm. It can be a useful tool to help you prepare.

Question 0

Require `tidyverse` and load the `mv.Rds` (https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/5_Regression/data/mv.Rds?raw=true') data to an object called `movies` . (Tip: use the `read_rds()` function with the link to the raw data.)

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

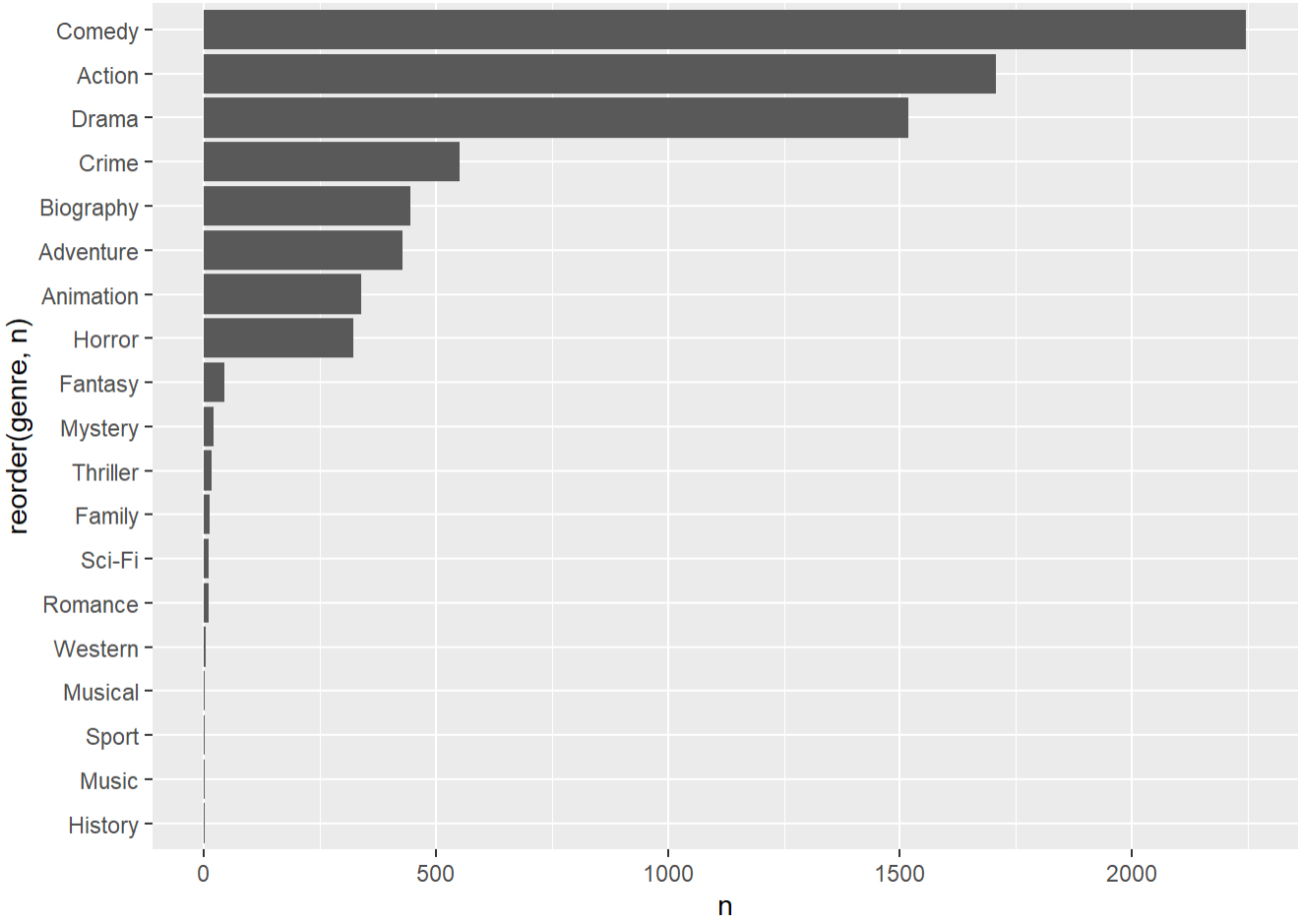
```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6     ✓ purrr   0.3.4
## ✓ tibble  3.1.7     ✓ dplyr   1.0.9
## ✓ tidyr   1.2.0     ✓ stringr 1.4.0
## ✓ readr   2.1.2     ✓ forcats 0.5.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
movies <- read_rds('../data/mv.Rds') #https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/5_Regression/data/mv.Rds?raw=true
```

Question 1 [1 points]

Which movie genres make the most money at the box office? To answer this question, begin by **looking** at the data. Specifically, create a univariate visualization of the `genre` category. Which categories appear the most? Which appear the least?

```
movies %>%
  count(genre) %>%
  ggplot(aes(y = reorder(genre,n),x = n)) +
  geom_bar(stat = 'identity')
```

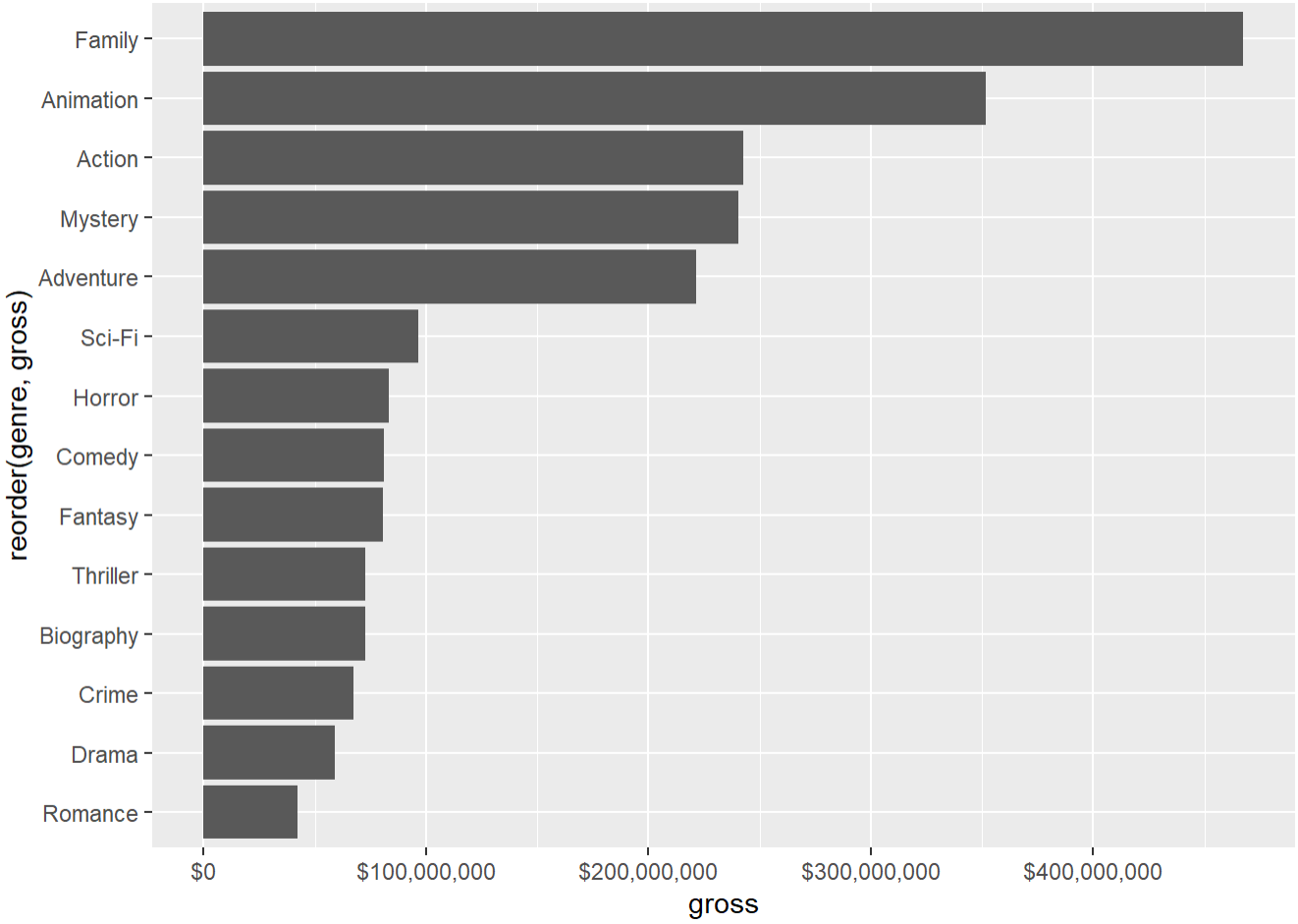


> The Comedy, Action, and Drama categories appear the most, the Sport, Music, and History categories appear the least.

Question 2 [1 point]

Drop any genre that appears fewer than 10 times in the data. Now plot a multivariate visualization comparing `genre` with `gross`. Which `genre` has the highest grossing movies (on average)?

```
movies %>%
  group_by(genre) %>%
  summarise(n = n(),
            gross = mean(gross,na.rm=T)) %>%
  ungroup() %>%
  filter(n >= 10) %>%
  ggplot(aes(y = reorder(genre,gross),x = gross)) +
  geom_bar(stat = 'identity') +
  scale_x_continuous(labels = scales::dollar)
```



> The Family and Animation genres make the most money on average, while the Crime, Drama, and Romance categories make the least.

Question 3 [1 point]

Estimate a regression that predicts `gross` as a function of `genre`, dropping any genre that appears fewer than 10 times. Make sure to `log()` the outcome! What is the hold-out category? Which genres are you more than 95% confident make **more** money than this category? Which genres are you 95% confident make **less** money than this category?

```
mv_analysis <- movies %>%
  group_by(genre) %>%
  mutate(n=n()) %>%
  ungroup() %>%
  filter(n >= 10) %>%
  mutate(log_gross = log(gross),
         genre = factor(genre)) %>%
  drop_na(genre,log_gross)

summary(m1 <- lm(log_gross ~ genre,mv_analysis))
```

```
##
## Call:
## lm(formula = log_gross ~ genre, data = mv_analysis)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3162  -1.0008   0.4133   1.4032   5.0777
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    18.27258    0.06493 281.431  < 2e-16 ***
## genreAdventure -0.22466    0.15555  -1.444   0.1487
## genreAnimation  0.63134    0.14612   4.321 1.59e-05 ***
## genreBiography -1.22844    0.13824  -8.886  < 2e-16 ***
## genreComedy    -1.23818    0.09184 -13.481  < 2e-16 ***
## genreCrime     -1.75002    0.13980 -12.518  < 2e-16 ***
## genreDrama     -1.94773    0.09808 -19.858  < 2e-16 ***
## genreFamily    -2.30422    1.18718  -1.941   0.0523 .
## genreFantasy   -1.38034    0.57314  -2.408   0.0161 *
## genreHorror    -1.23948    0.16907  -7.331 2.74e-13 ***
## genreMystery   -0.33293    0.72881  -0.457   0.6478
## genreRomance   -2.37684    1.02864  -2.311   0.0209 *
## genreSci-Fi    -0.11873    1.18718  -0.100   0.9203
## genreThriller  -0.27502    1.02864  -0.267   0.7892
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.053 on 3988 degrees of freedom
## Multiple R-squared:  0.1394, Adjusted R-squared:  0.1366
## F-statistic: 49.67 on 13 and 3988 DF,  p-value: < 2.2e-16
```

The hold-out category is the Action genre. Relative to this genre, only Animation makes more money. All the rest make less money, although we are not confident in this conclusion at the 95% level for the Adventure, Family, Mystery, Sci-Fi, and Thriller genres.

Question 4 [1 point]

Now add the budget (logged) to this regression. Again, what is the hold-out category? Which genres are you more than 95% confident make **more** money than this category? Which genres are you 95% confident make **less** money than this category?

```
mv_analysis <- mv_analysis %>%
  mutate(log_budget = log(budget))
summary(lm(log_gross ~ genre + log_budget,mv_analysis))
```

```
##
## Call:
## lm(formula = log_gross ~ genre + log_budget, data = mv_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1024 -0.6049  0.1440  0.7586  7.6442
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.63260    0.35790   4.562 5.27e-06 ***
## genreAdventure  0.12581    0.10280   1.224 0.221098
## genreAnimation  0.46781    0.09520   4.914 9.38e-07 ***
## genreBiography -0.21917    0.09671  -2.266 0.023505 *
## genreComedy    -0.06145    0.06485  -0.948 0.343438
## genreCrime     -0.37563    0.09943  -3.778 0.000161 ***
## genreDrama     -0.20946    0.07431  -2.819 0.004851 **
## genreFamily     0.41365    0.88966   0.465 0.641999
## genreFantasy   -0.35293    0.35225  -1.002 0.316449
## genreHorror     0.74749    0.11669   6.406 1.72e-10 ***
## genreMystery    0.48346    0.47698   1.014 0.310862
## genreRomance   -1.89953    0.63002  -3.015 0.002590 **
## genreSci-Fi    -0.06774    0.88966  -0.076 0.939311
## genreThriller   0.99620    0.63067   1.580 0.114300
## log_budget      0.94326    0.01978  47.697  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.257 on 3164 degrees of freedom
##   (823 observations deleted due to missingness)
## Multiple R-squared:  0.5001, Adjusted R-squared:  0.4979
## F-statistic: 226.1 on 14 and 3164 DF,  p-value: < 2.2e-16
```

The hold-out category is still the Action genre. After adding logged budget, our conclusions change. We are still confident that the Animation genre makes more money than Action, but we now also find that the Horror genre does as well. We can also now only say with 95% confidence that the Bio, Crime, Drama, and Romance genres make less money than Action movies.

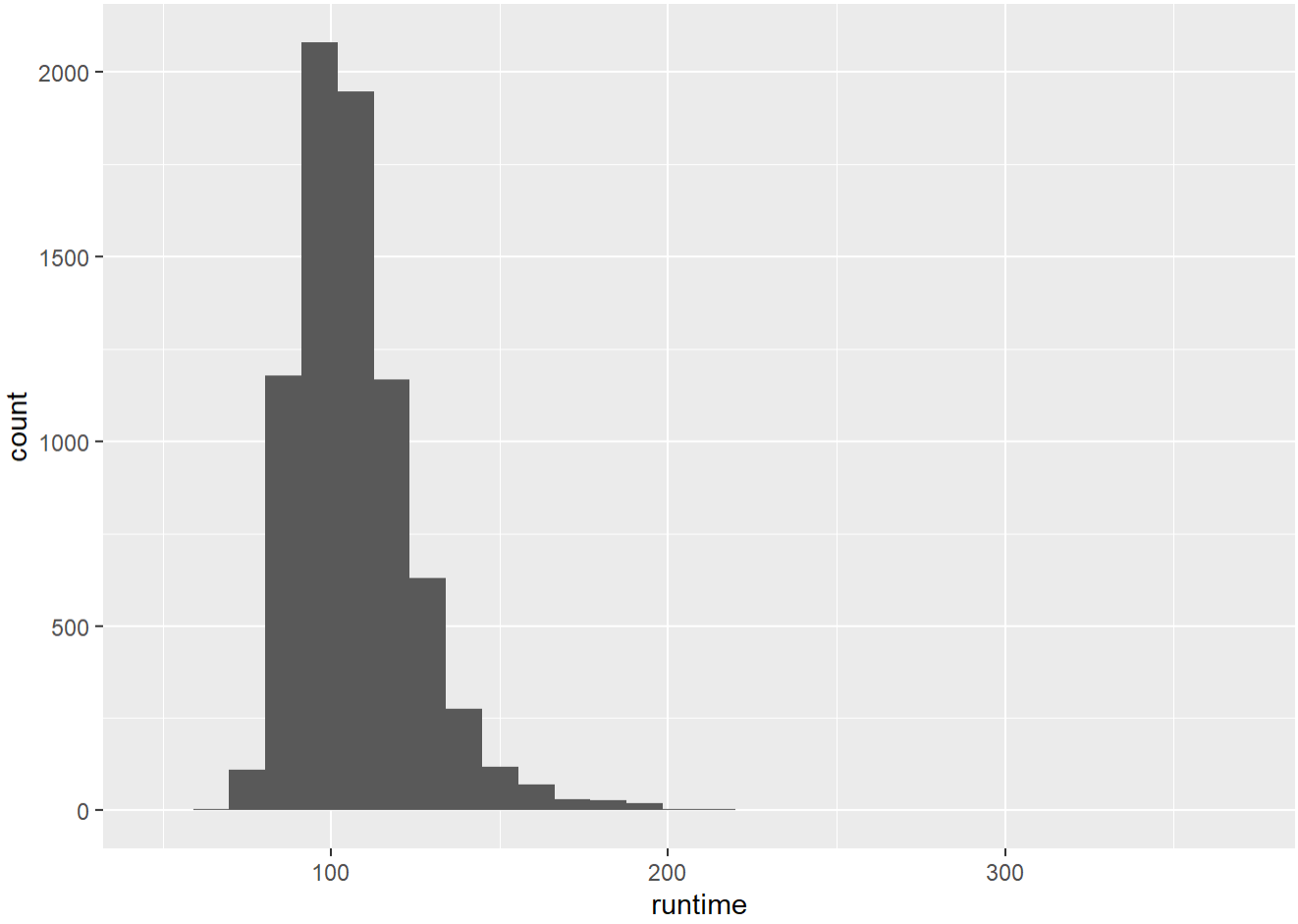
Question 5 [1 point]

Let's consider some other predictors: length (`runtime`), audience score (`score`), budget (`budget`), and rating (`rating`). Visualize each of these with a univariate visualization. Do you see any with sufficient skew to justify a log transformation?

```
movies %>%
  ggplot(aes(x = runtime)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

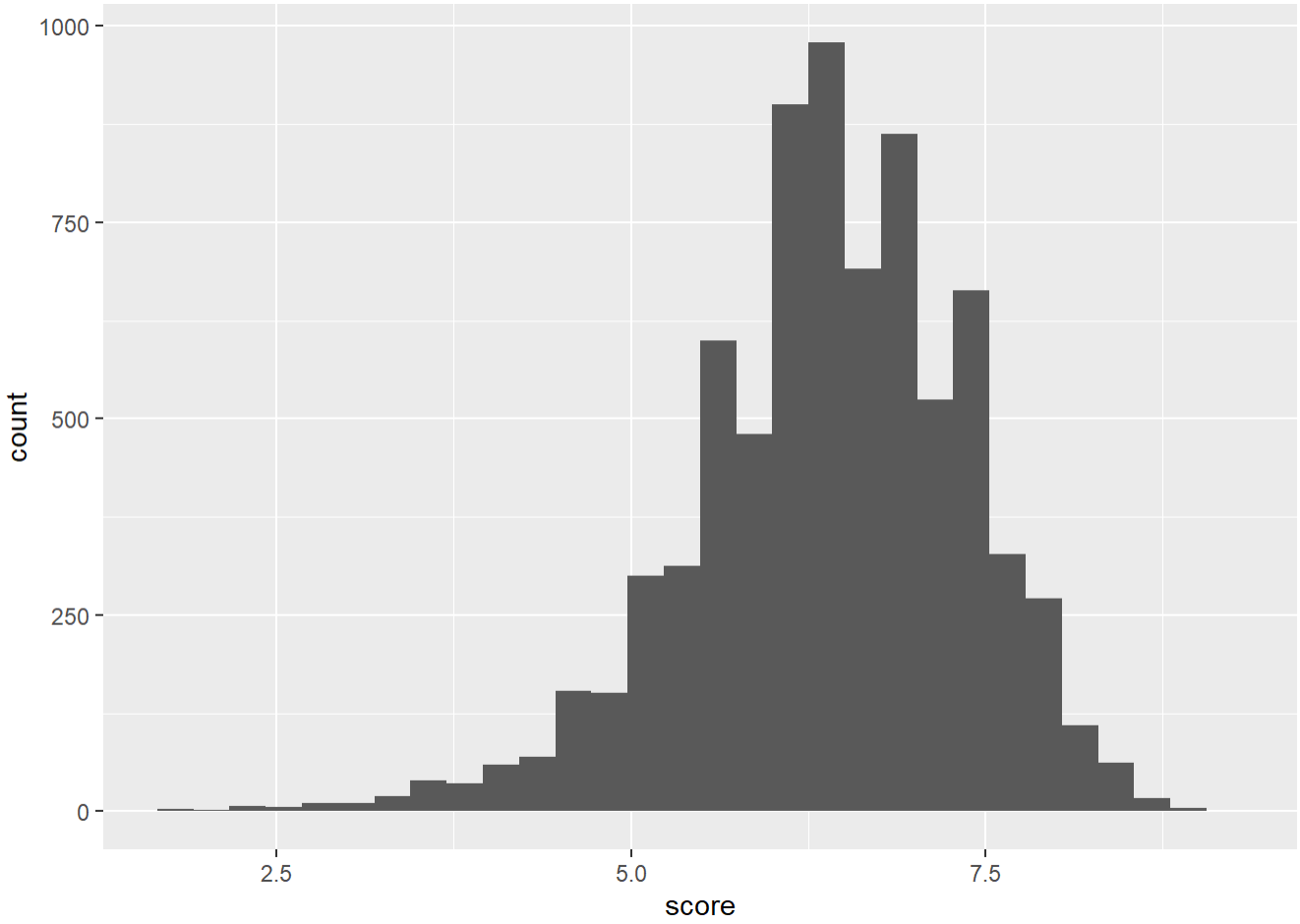
```
## Warning: Removed 4 rows containing non-finite values (stat_bin).
```



```
movies %>%
  ggplot(aes(x = score)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

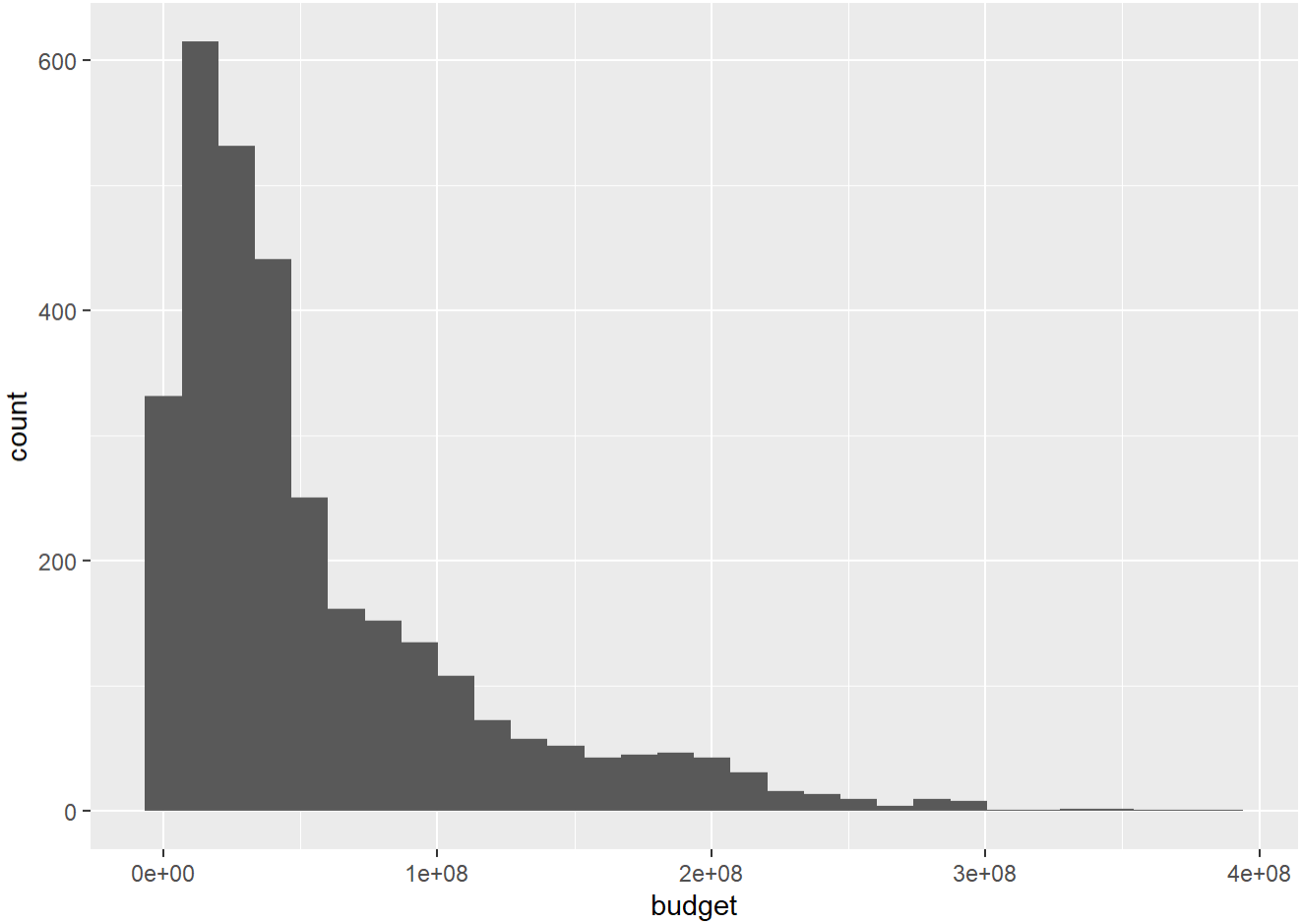
```
## Warning: Removed 3 rows containing non-finite values (stat_bin).
```



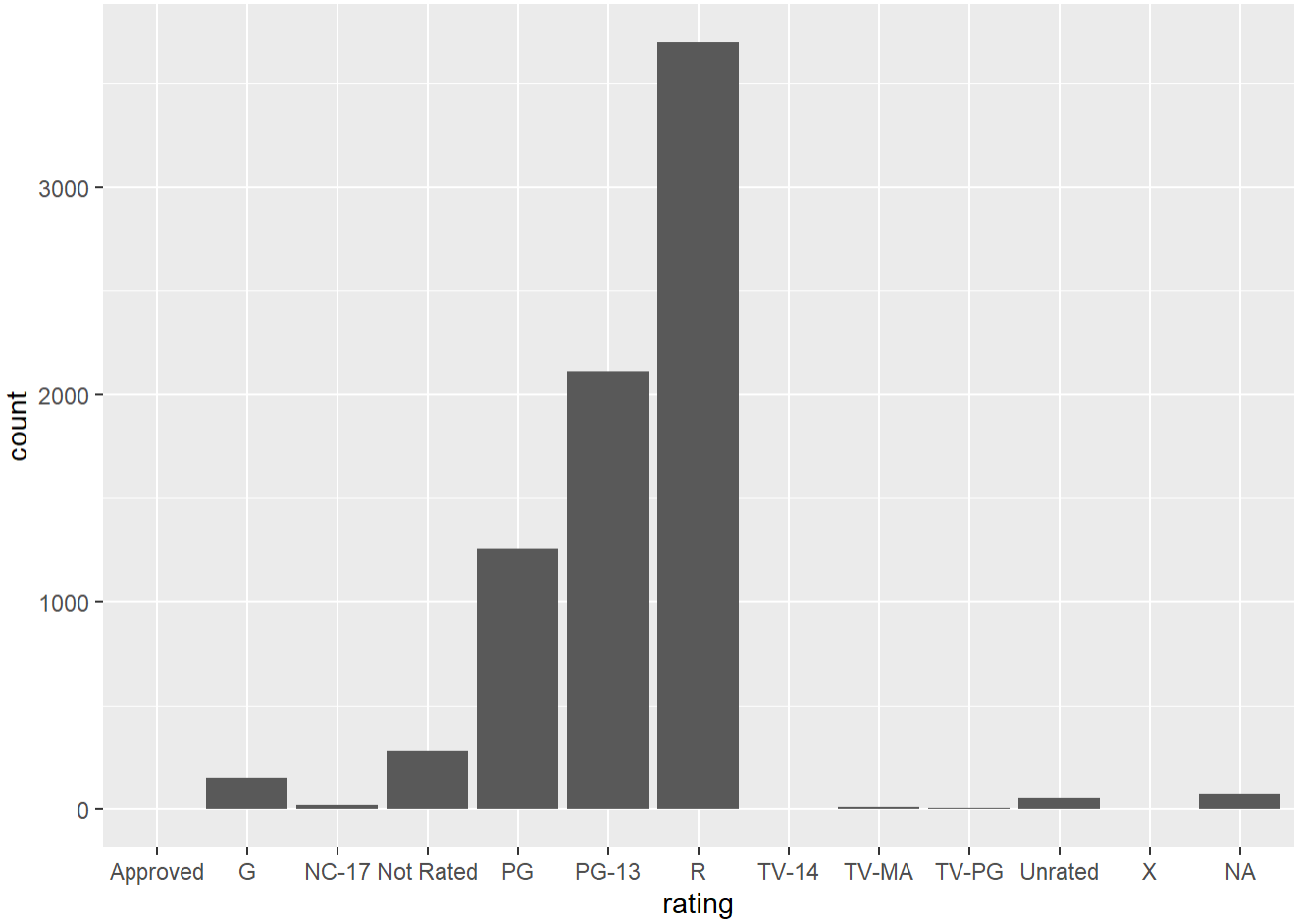

```
movies %>%
  ggplot(aes(x = budget)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 4482 rows containing non-finite values (stat_bin).
```




```
movies %>%
  ggplot(aes(x = rating)) +
  geom_bar()
```



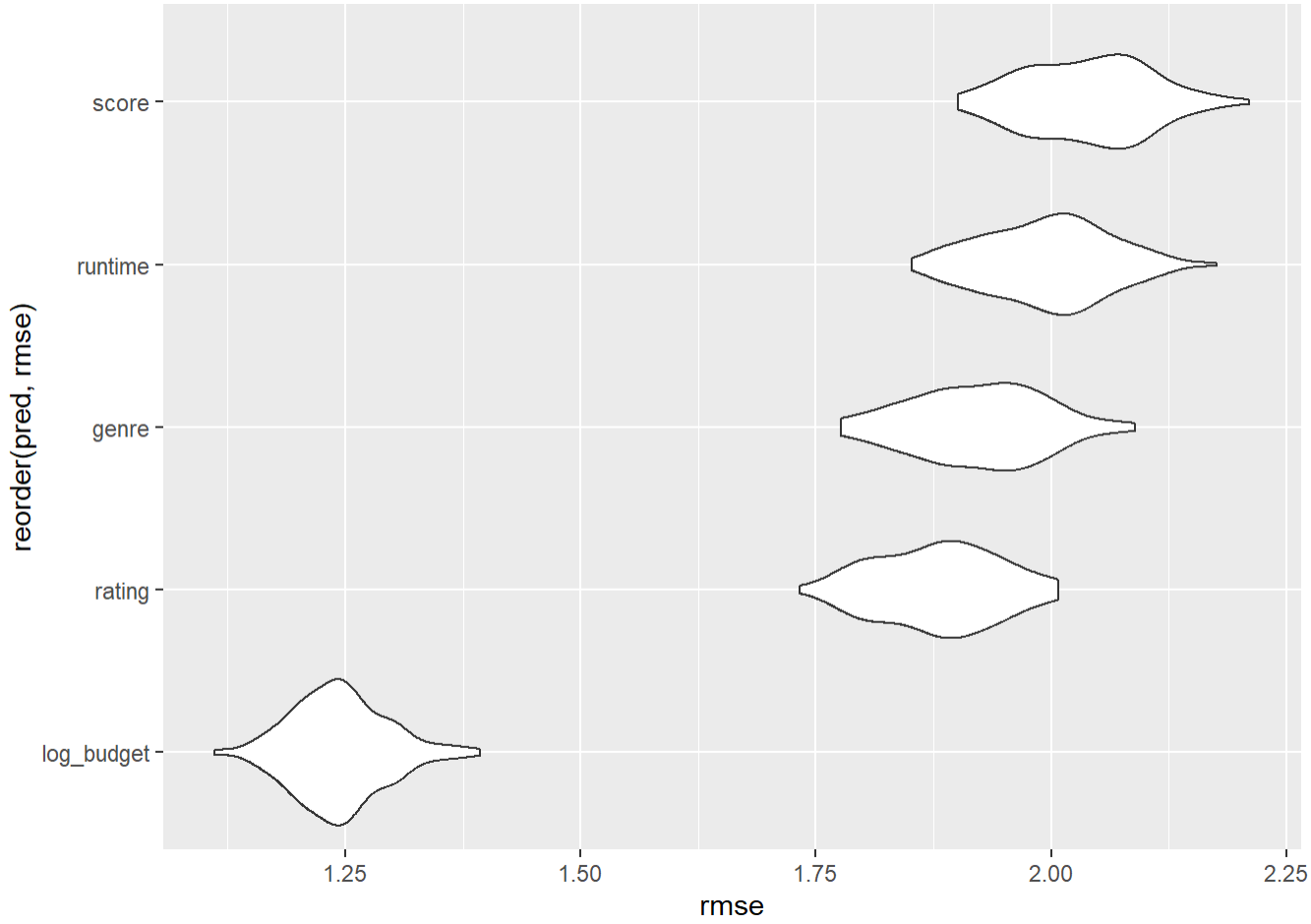
The only variable with sufficient skew to justify a log transformation is Budget.

Question 6 [3 points]

Apply the log transformation to the variable(s) that you identified, and drop any movies that weren't rated G, PG, PG-13, and R. Now run regressions predicting `log_gross` with each of these variables one-by-one (`genre` , `score` , `runtime` , `rating` , `budget`), and calculate the RMSE with 20-fold cross validation with an 80-20 split. **HINT**: make sure to `group_by(rating,genre)` prior to `sample_n` , and use the `anti_join()` solution to creating the `train` and `test` sets (see the notes from the 03/01 review session). Which variable best predicts `log_gross` ? Visualize your answer with `geom_violin()` . **HINT #2**: You can put every regression inside a single `for()` loop to save time (i.e., instead of running `for()` loops for each variable).

```
mv_analysis <- mv_analysis %>%
  filter(rating %in% c('G','PG','PG-13','R')) %>%
  select(log_gross,log_budget,rating,runtime,score,genre)

rmseRes <- NULL
for(i in 1:100) {
  train <- mv_analysis %>%
    group_by(rating,genre) %>%
    sample_n(size = round(n()*.8),replace = F)

  test <- mv_analysis %>%
    anti_join(train)

  m <- lm(log_gross ~ genre,train)

  rmseRes <- test %>%
    mutate(preds = predict(m,test)) %>%
    summarise(rmse = sqrt(mean((log_gross - preds)^2,na.rm=T))) %>%
    mutate(cvInd = i,
           pred = 'genre') %>%
    bind_rows(rmseRes)

  m <- lm(log_gross ~ rating,train)

  rmseRes <- test %>%
    mutate(preds = predict(m,test)) %>%
    summarise(rmse = sqrt(mean((log_gross - preds)^2,na.rm=T))) %>%
    mutate(cvInd = i,
           pred = 'rating') %>%
    bind_rows(rmseRes)

  m <- lm(log_gross ~ score,train)

  rmseRes <- test %>%
    mutate(preds = predict(m,test)) %>%
    summarise(rmse = sqrt(mean((log_gross - preds)^2,na.rm=T))) %>%
    mutate(cvInd = i,
           pred = 'score') %>%
    bind_rows(rmseRes)

  m <- lm(log_gross ~ log_budget,train)
```

```
  rmseRes <- test %>%
    mutate(preds = predict(m,test)) %>%
    summarise(rmse = sqrt(mean((log_gross - preds)^2,na.rm=T))) %>%
    mutate(cvInd = i,
           pred = 'log_budget') %>%
    bind_rows(rmseRes)

  m <- lm(log_gross ~ runtime,train)

  rmseRes <- test %>%
    mutate(preds = predict(m,test)) %>%
    summarise(rmse = sqrt(mean((log_gross - preds)^2,na.rm=T))) %>%
    mutate(cvInd = i,
           pred = 'runtime') %>%
    bind_rows(rmseRes)
}

rmseRes %>%
  ggplot(aes(x = rmse,y = reorder(pred,rmse))) +
  geom_violin()
```

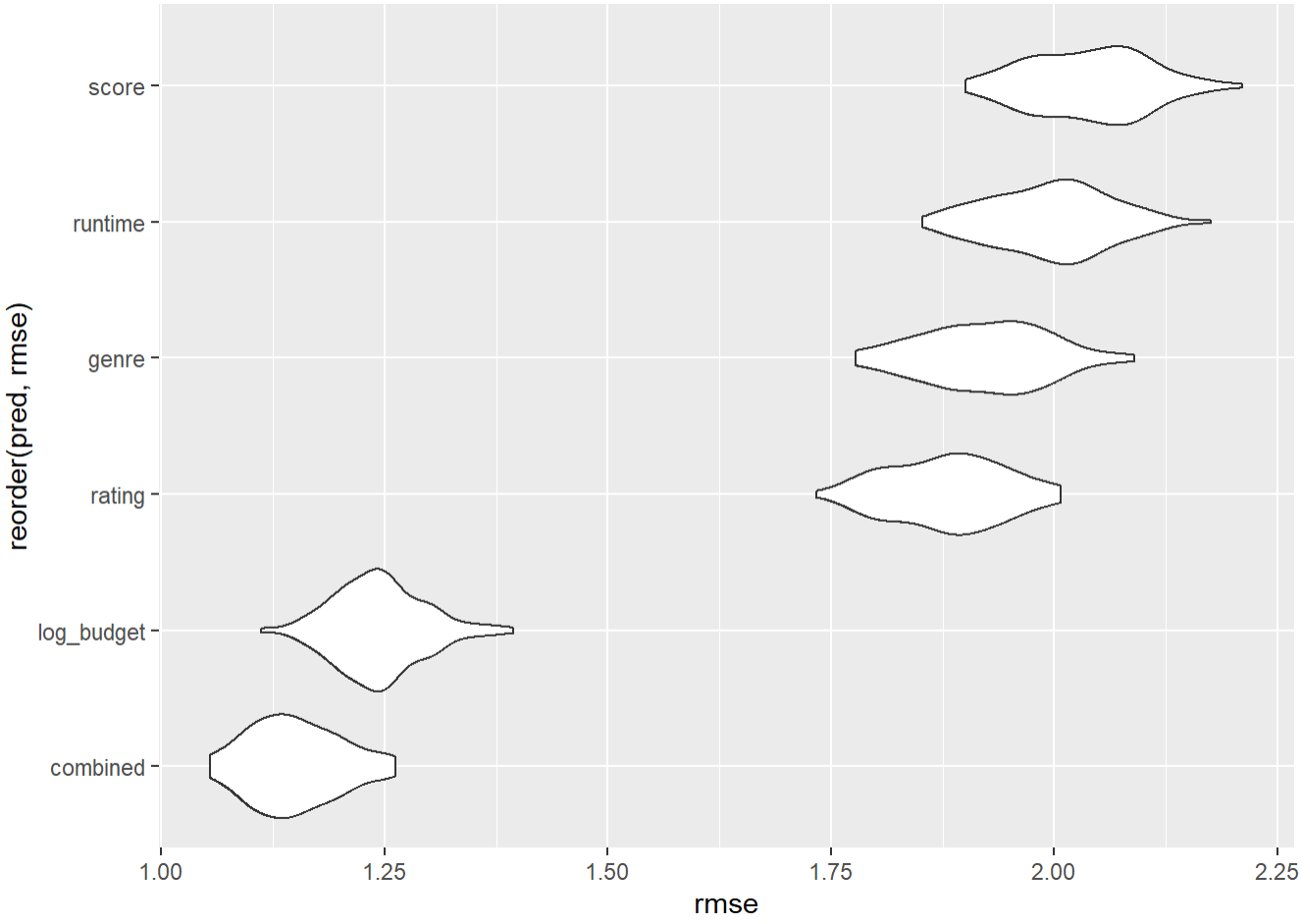


Budget is the variable that best predicts gross, as indicated with a much smaller RMSE than the other predictors.

Question 7 [1 point]

Now run a single regression in which `log_gross` is predicted as a function of all of the preceding predictors. Calculate the RMSE with 100-fold cross validation, again with an 80-20 split. Does the model fit improve relative to the best-performing single predictor from above?

```
for(i in 1:100) {
  train <- mv_analysis %>%
    group_by(rating,genre) %>%
    sample_n(size = round(n()*.8),replace = F)

  test <- mv_analysis %>%
    anti_join(train)

  m <- lm(log_gross ~ rating + score + log_budget + runtime + genre,train)

  rmseRes <- test %>%
    mutate(preds = predict(m,test)) %>%
    summarise(rmse = sqrt(mean((log_gross - preds)^2,na.rm=T))) %>%
    mutate(cvInd = i,
           pred = 'combined') %>%
    bind_rows(rmseRes)
}

rmseRes %>%
  ggplot(aes(x = rmse,y = reorder(pred,rmse))) +
  geom_violin()
```




```
rmseRes %>%
  group_by(pred) %>%
  summarise(rmse = mean(rmse))
```

```
## # A tibble: 6 × 2
##   pred        rmse
##   <chr>      <dbl>
## 1 combined    1.15
## 2 genre       1.92
## 3 log_budget  1.24
## 4 rating      1.88
## 5 runtime     2.00
## 6 score       2.04
```

> Yes the model fit improves when we include all the predictors together, moving from 1.25 with log_budget alone, to 1.15 to the model that includes every predictor.

Question 8 [1 point]

Re-estimate the preceding multiple regression model on the full data. Then use it to generate a prediction for an investor wanting to know how much money they will make if they invest $50m in a fantasy movie, rated PG-13, with a 120 minute run-time and an audience score of 6. Use the RMSE estimate from the preceding question to give the lower and upper bounds on this prediction. Would you recommend that the investor invest in this project?

```
m <- lm(log_gross ~ rating + score + log_budget + runtime + genre,mv_analysis)

rmseSum <- rmseRes %>%
  group_by(pred) %>%
  summarise(rmse = mean(rmse))

test <- data.frame(rating = 'PG-13',score = 6,runtime = 120,genre = 'Fantasy',log_budget = log(50000000))

scales::dollar(exp(predict(m,newdata = test)))
```

```
##             1
## "$168,888,935"
```

```
scales::dollar(exp(predict(m,newdata = test) - rmseSum$rmse[1]))
```

```
##            1
## "$53,527,000"
```

```
scales::dollar(exp(predict(m,newdata = test) + rmseSum$rmse[1]))
```

```
##             1
## "$532,880,091"
```

Yes, I would recommend that the investor pursue this project. For a $50m investment, they will get – on average – $169m. At the low end, our model still predicts that they will make $53.7m, or a $3.7m return. Their upside is as large as $531m!