Problem Set 5

Regression

[YOUR NAME]

Due Date: 2023-10-22

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]_ps5.Rmd` to your `code` folder.

Copy and paste the contents of this file into your `[LAST NAME]_ps5.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 worth 10 total points, plus five extra credit points. The point values for each question are indicated in brackets below. To receive full credit, you must both have the correct code **and include a comment describing what each line does**. In addition, some questions ask you to provide a written response in addition to the code. Unlike the first two problem sets, some of the code chunks are totally empty, requiring you to try writing the code from scratch. Make sure to comment each line, explaining what it is doing!

You are free to rely on whatever resources you need to complete this problem set, including lecture notes, lecture presentations, Google, your classmates…you name it. However, the final submission must be complete by you. There are no group assignments. To submit, compiled the completed problem set and upload the PDF file to Brightspace by midnight on 2023/10/22. Also note that the TAs and professors will not respond to Campuswire posts after 5PM on Friday, so don't wait until the last minute to get started!

Good luck!

ChatGPT Link [Optional]

*Copy the link to ChatGPT you used here: _______________________.

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.)

```r
require(tidyverse)
```

```
## Loading required package: tidyverse
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.2     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
movies <- read_rds('../data/mv.Rds') #https://github.com/jbisbee1/DS1000_S2023/blob/main/Lecture
s/4_Uni_Multivariate/data/game_summary.Rds?raw=true')
```

Question 1 [1 point]

In this problem set, we will answer the following research question: do movies that score higher among audiences (`score`) make more money (`gross`). First, write out a **theory** that answers this question and transform it into a **hypothesis**.

> Theory: people want to see good movies. Hypothesis: therefore the higher a movie scores among audiences, the more money it should make.

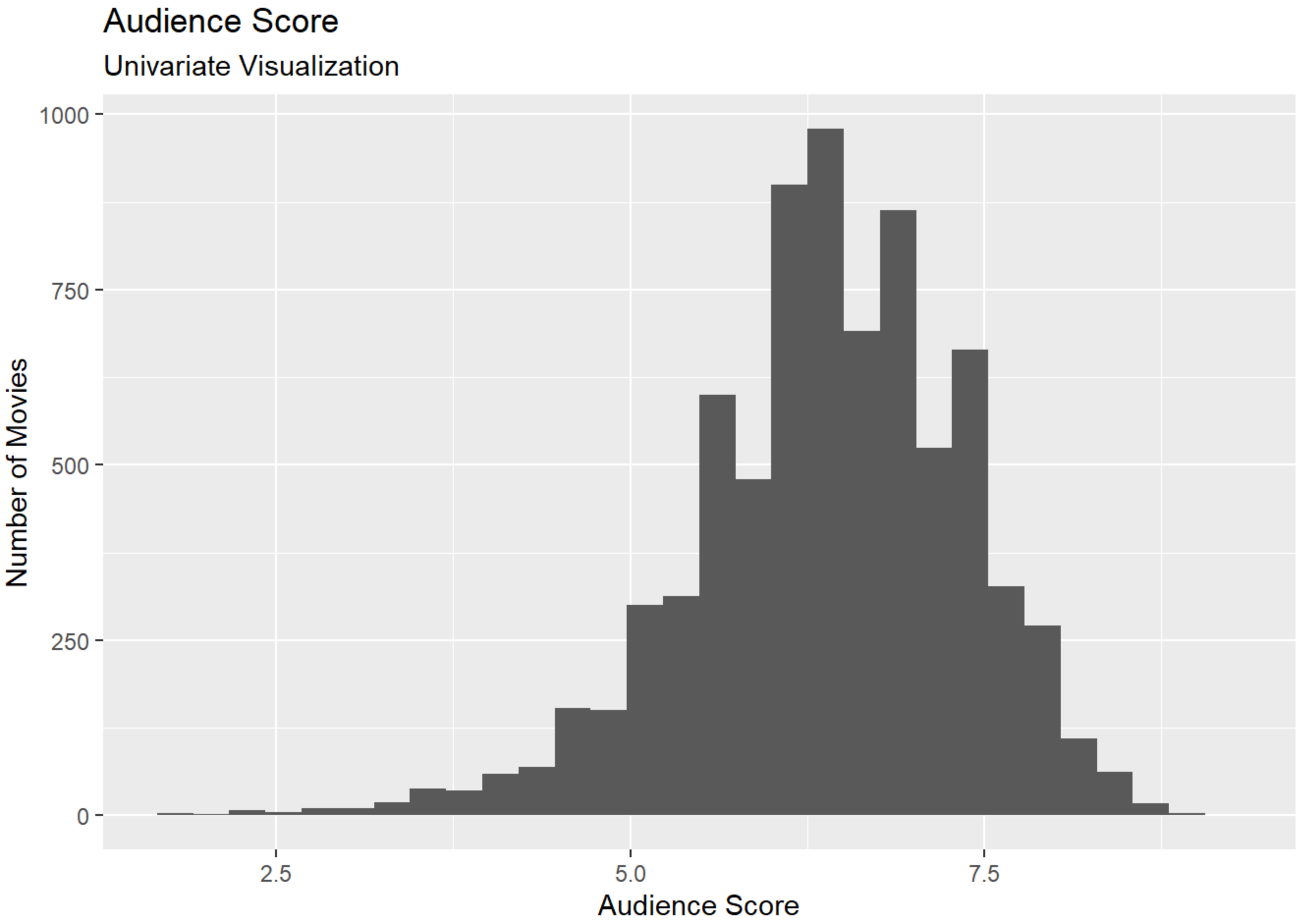
Question 2 [1 point]

Based on your theory, which variable is the X variable (i.e., the independent variable or the predictor)? Which variable is the Y variable (i.e., the dependent variable or the outcome)? Use **univariate** visualization to create two plots, one for each variable. Do you need to apply a log-transformation to either of these variables? Why?

```
# Predictor: score
movies %>%
  ggplot(aes(x = score)) +
  geom_histogram() +
  labs(title = 'Audience Score',
       subtitle = 'Univariate Visualization',
       x = 'Audience Score',
       y = 'Number of Movies')
```

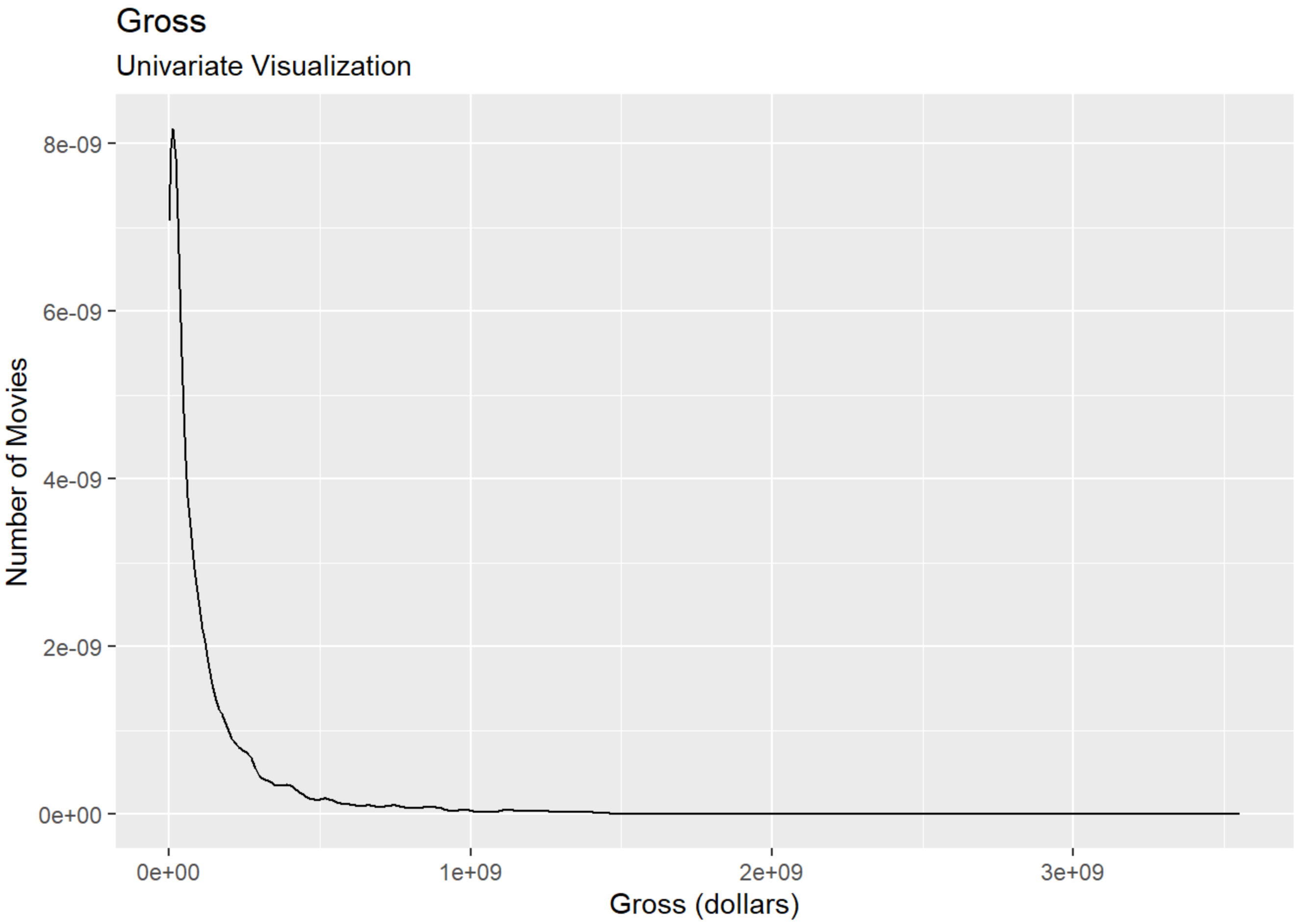
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

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



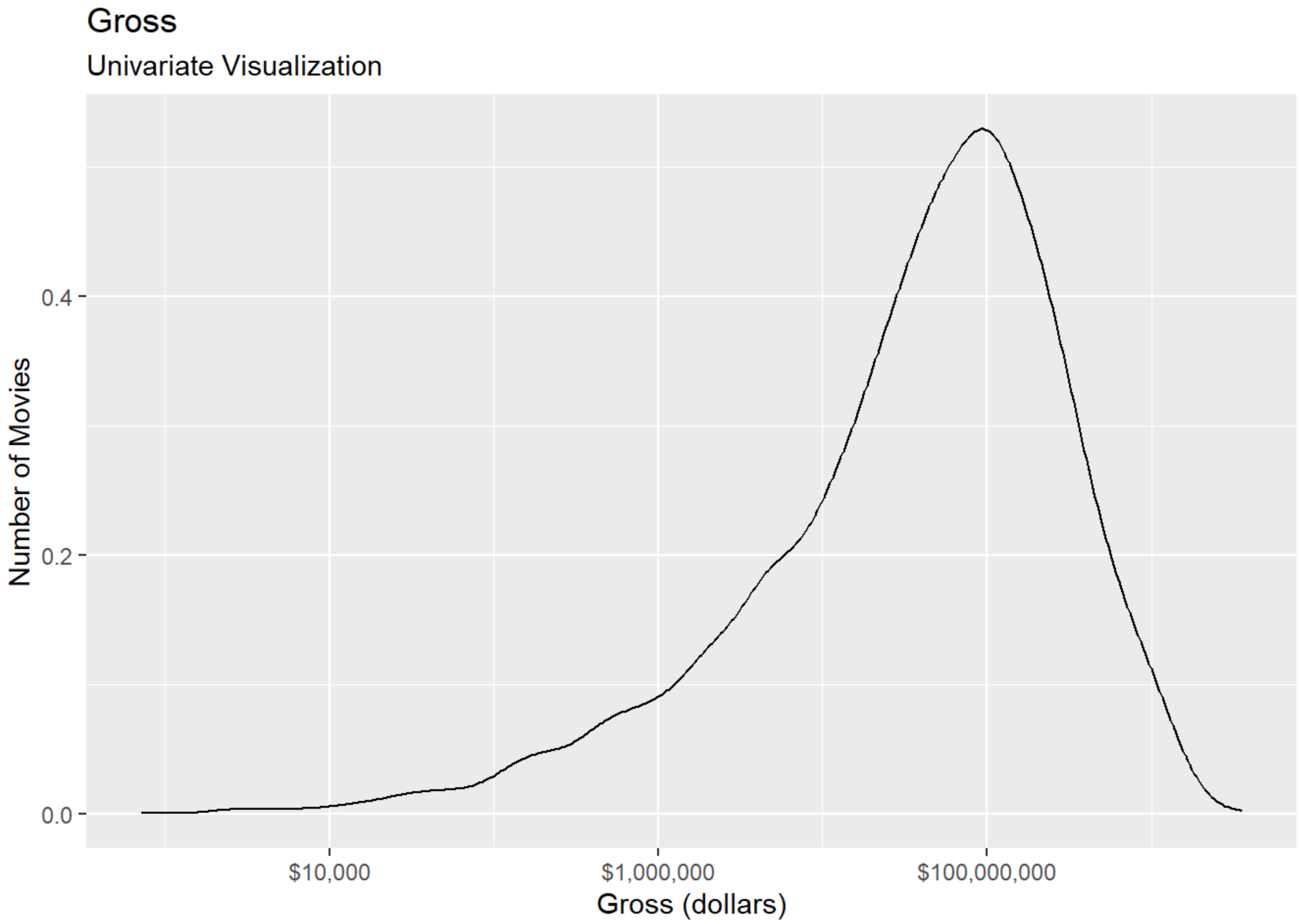
```
# Outcome: gross
movies %>%
  ggplot(aes(x = gross)) +
  geom_density() +
  labs(title = 'Gross',
       subtitle = 'Univariate Visualization',
       x = 'Gross (dollars)',
       y = 'Number of Movies')
```

```
## Warning: Removed 3668 rows containing non-finite values (`stat_density()`).
```



```
movies %>%
  ggplot(aes(x = gross)) +
  geom_density() +
  scale_x_log10(label = scales::dollar) +
  labs(title = 'Gross',
       subtitle = 'Univariate Visualization',
       x = 'Gross (dollars)',
       y = 'Number of Movies')
```

```
## Warning: Removed 3668 rows containing non-finite values (`stat_density()`).
```



> According to my theory, the audience score is the predictor / independent / X variable and the movie's gross is the outcome / dependent / Y variable. Univariate visualization revealed that the gross variable is highly skewed, meaning that I should use a log transformation.

Question 3 [2 points]

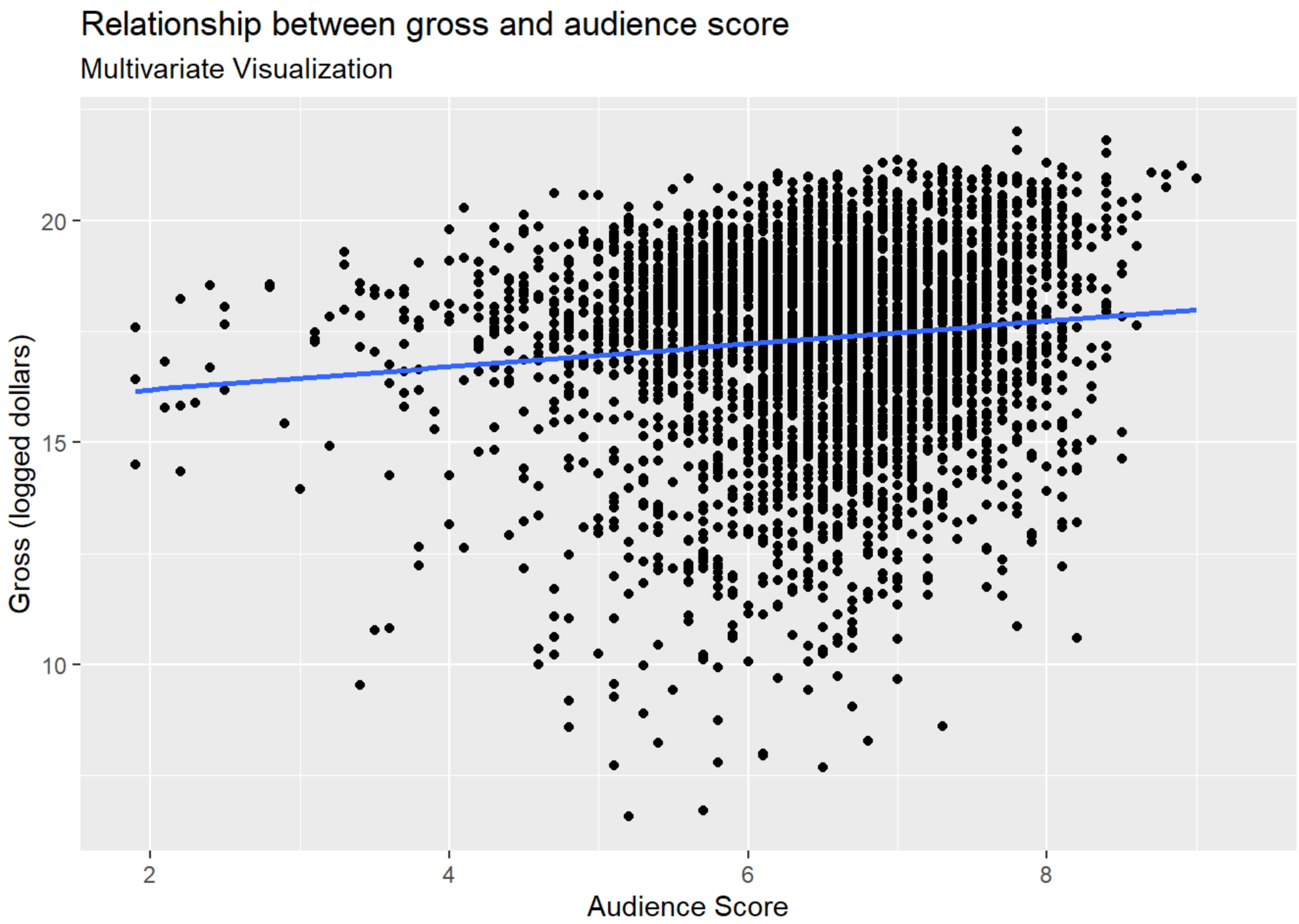
Now create a multivariate visualization of these two variables, making sure to put the independent variable on the x-axis and the dependent variable on the y-axis. Add the line of best fit. Make sure to log the data if you determined this was necessary in the previous question! Does the visualization support your hypothesis?

```
movies %>%
  mutate(log_gross = log(gross)) %>%
  ggplot(aes(x = score,y = log_gross)) +
  geom_point() +
  geom_smooth(method = 'lm',se = F) +
  labs(title = 'Relationship between gross and audience score',
       subtitle = 'Multivariate Visualization',
       x = 'Audience Score',
       y = 'Gross (logged dollars)')
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 3668 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 3668 rows containing missing values (`geom_point()`).
```



> The visualization does support my hypothesis by showing there is a positive relationship between the audience score and how much money the movie makes.

Question 4 [2 points]

Now estimate the regression using the `lm()` function. Describe the output of the model in English, talking about the intercept, the slope, and the statistical significance.

```
movies_analysis <- movies %>%
  mutate(log_gross = log(gross)) %>%
  drop_na(log_gross,score)

model_gross_score <- lm(formula = log_gross ~ score,
                        data = movies_analysis)

summary(model_gross_score)
```

```
##
## Call:
## lm(formula = log_gross ~ score, data = movies_analysis)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4413  -1.1336   0.4662   1.5388   4.3091
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.67373    0.23785  65.897  < 2e-16 ***
## score        0.25745    0.03639   7.075 1.76e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.197 on 4003 degrees of freedom
## Multiple R-squared:  0.01235,    Adjusted R-squared:  0.0121
## F-statistic: 50.05 on 1 and 4003 DF,  p-value: 1.762e-12
```

```
exp(15.67373)
```

```
## [1] 6412309
```

```
(exp(.25745)-1)*100
```

```
## [1] 29.36271
```

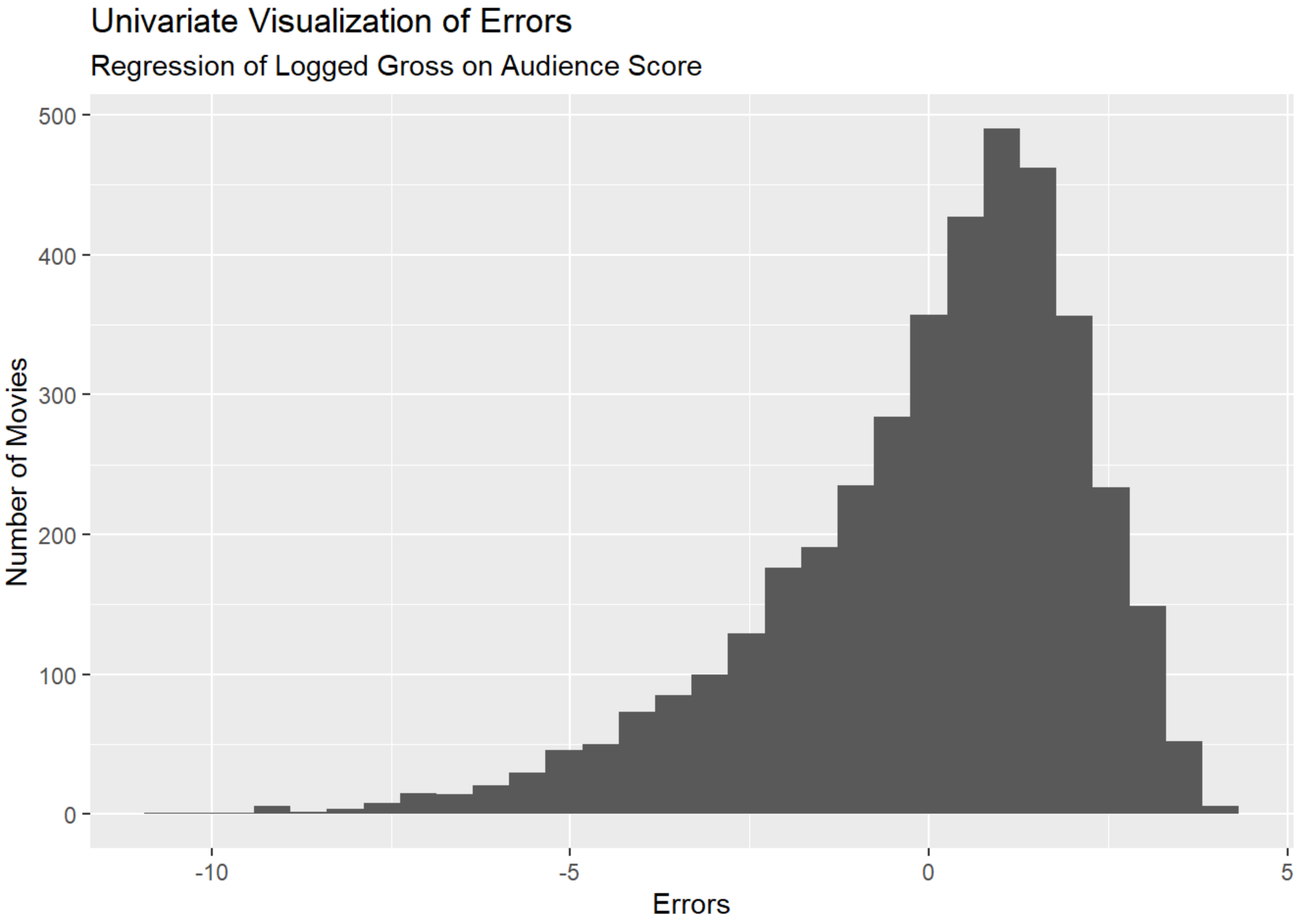
> The model indicates that movies which scored a zero had an average gross of 15.67 logged dollars, or $6.4 million. Each additional point of audience score is associated with an increase of 0.257 logged dollars, which is better expressed as a 29.36 percentage point increase. This relationship is highly significant, with a p-value that is almost zero, meaning we are above 99.9% confident that the relationship is not zero.

Question 5 [2 points]

Now calculate the model's prediction errors and create both a univariate and multivariate visualization of them. Based on these analyses, would you say that your model does a good job predicting how much money a movie makes? **Make sure to reference both the univariate and multivariate visualization of the errors!**

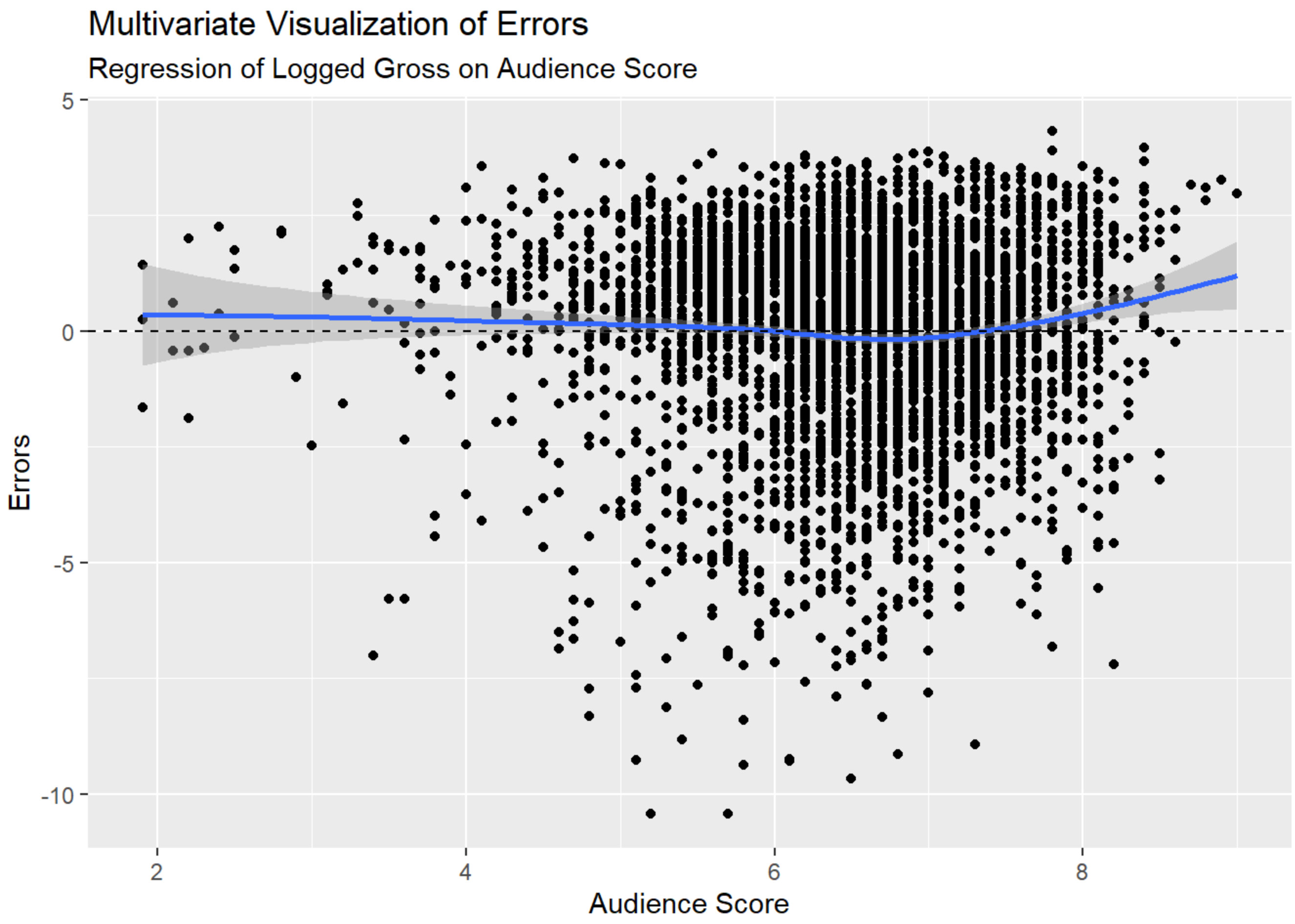
```
movies_analysis <- movies_analysis %>%
  mutate(preds = predict(model_gross_score)) %>%
  mutate(errors = log_gross - preds)

# Univariate
movies_analysis %>%
  ggplot(aes(x = errors)) +
  geom_histogram() +
  labs(title = 'Univariate Visualization of Errors',
       subtitle = 'Regression of Logged Gross on Audience Score',
       x = 'Errors',
       y = 'Number of Movies')
```

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



```
# Multivariate
movies_analysis %>%
  ggplot(aes(x = score,y = errors)) +
  geom_point() +
  geom_smooth() +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  labs(title = 'Multivariate Visualization of Errors',
       subtitle = 'Regression of Logged Gross on Audience Score',
       x = 'Audience Score',
       y = 'Errors')
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



> I would conclude that the model is poor based on the univariate and multivariate visualization of the errors. The univariate visualization of the errors is not symmetric around zero, meaning the model overpredicts the logged gross more than it underpredicts the logged gross. A good model should have symmetrically distributed errors around zero. The multivariate visualization errors is also poor, indicating that the model underpredicts logged gross for movies that score between a 6 and a 7, and overpredicts either the lowest or highest scoring movies. A good model should have a rectangular cloud of points centered on zero on the y-axis, and a horizontal line of best fit centered on zero on the y-axis.

Question 6 [2 points]

Calculate the RMSE in the full data. Then calculate the RMSE using 100-fold cross validation with an 80-20 split and take the average of the 100 estimates. Which value is larger? Why?

```
# RMSE Full Data
movies_analysis %>%
  mutate(se = errors^2) %>%
  summarise(mse = mean(se)) %>%
  mutate(rmse = sqrt(mse))
```

```
## # A tibble: 1 × 2
##     mse  rmse
##   <dbl> <dbl>
## 1  4.82  2.20
```

```
# RMSE 100-fold CV
set.seed(123)
cvRes <- NULL # Instantiate an empty object to store data from the loop
for(i in 1:100) { # Loop 100 times
  inds <- sample(1:nrow(movies_analysis),size = round(.8*nrow(movies_analysis)),replace = F)

  train <- movies_analysis %>% slice(inds)
  test <- movies_analysis %>% slice(-inds)

  m <- lm(formula = log_gross ~ score,data = train)

  test$preds <- predict(m,newdata = test)

  e <- test$log_gross - test$preds
  se <- e^2
  mse <- mean(se)
  rmse <- sqrt(mse)
  cvRes <- c(cvRes,rmse)
}

mean(cvRes)
```

```
## [1] 2.198704
```

> The average RMSE calculated from 100 cross validation steps is slightly larger than the RMSE calculated from the full data. This is because our model might overfit to the data, which yields an unrealistically good measure of RMSE. We use cross validation to overcome this issue, producing a slightly larger RMSE and a more realistic understanding of our model's poor performance.

Question 7 [5 EC Points]

Using the same process as described in the preceding questions, answer the following research question: "Do movies that have a higher Bechdel Score (`bechdel_score`) make more money (`gross`)?" Make sure to include:

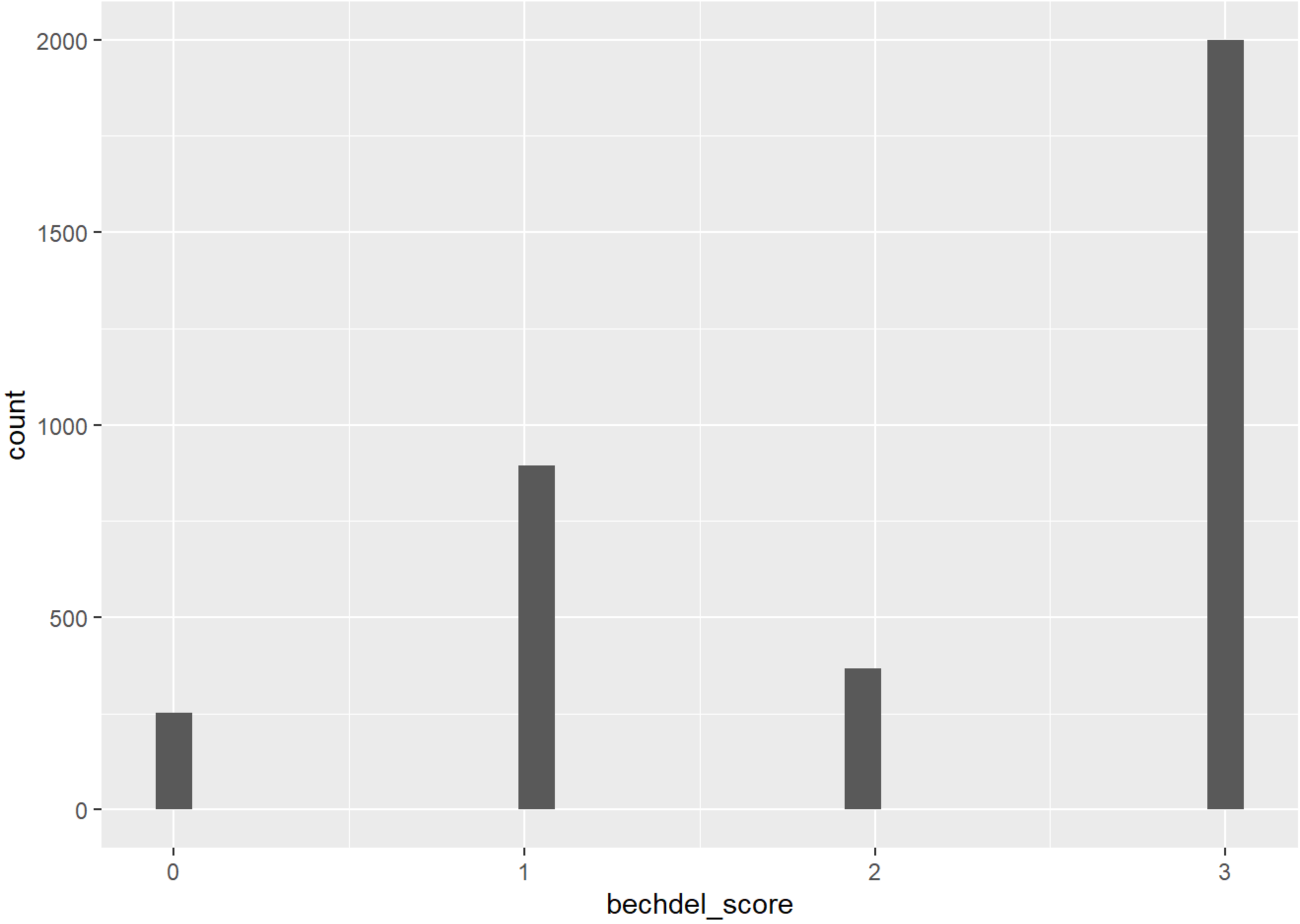
- A theory and hypothesis (write-up required: 2 to 3 sentences)
- Univariate AND multivariate visualizations of both the X and Y variables (no write-up required)
- A regression model using the `lm()` function (write-up required)
- Univariate and multivariate visualizations of the errors (write-up required: 2 to 3 sentences)
- Analysis of RMSE using 100-fold cross validation with an 80-20 split (no write-up required)

> I theorize that movies which have deeper female characters are appealing to women who comprise half of the population. Therefore, I hypothesize that the relationship between the Bechdel Score (X) and the movie's gross (Y) should be positive.

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

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

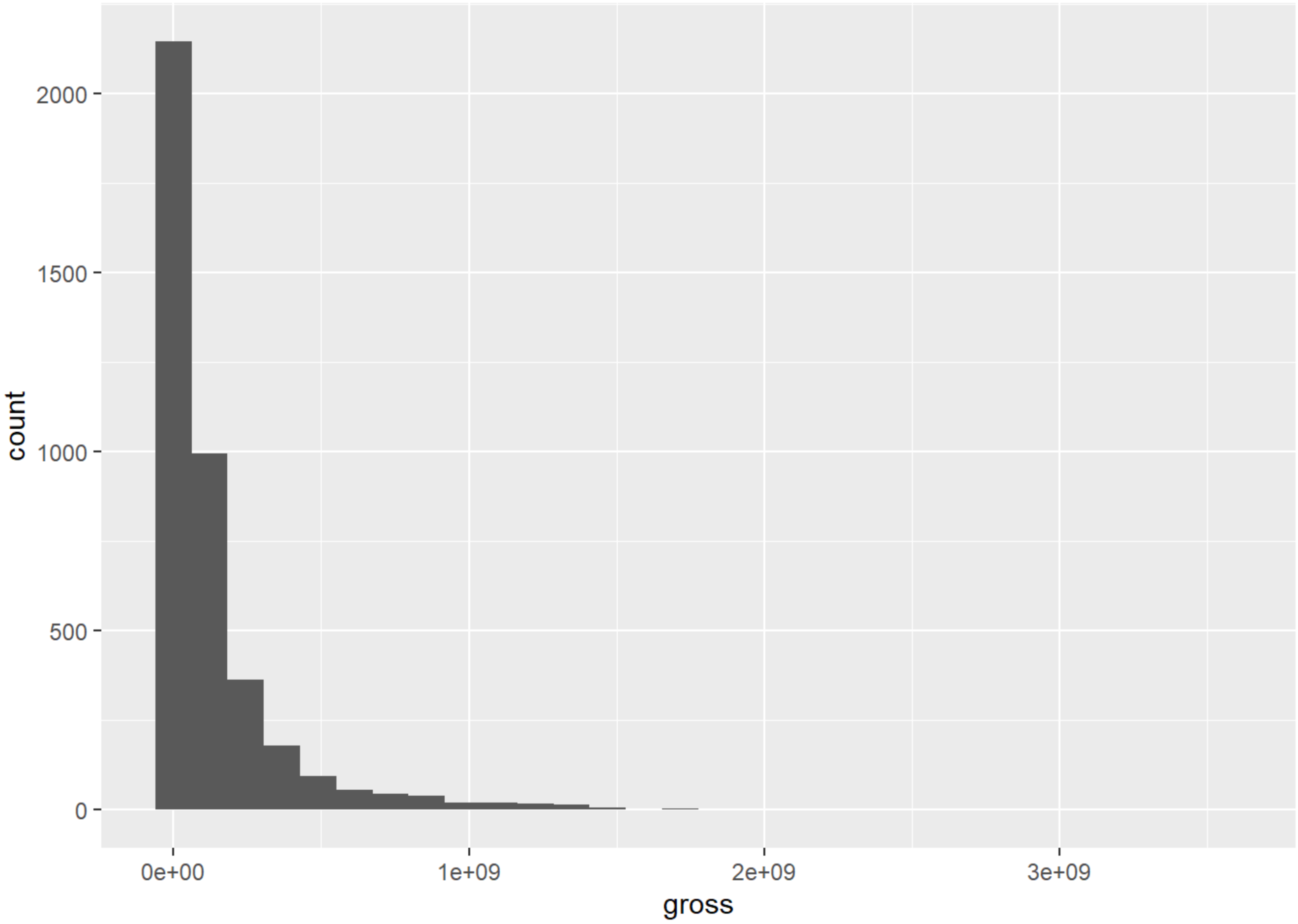
```
## Warning: Removed 4162 rows containing non-finite values (`stat_bin()`).
```



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

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

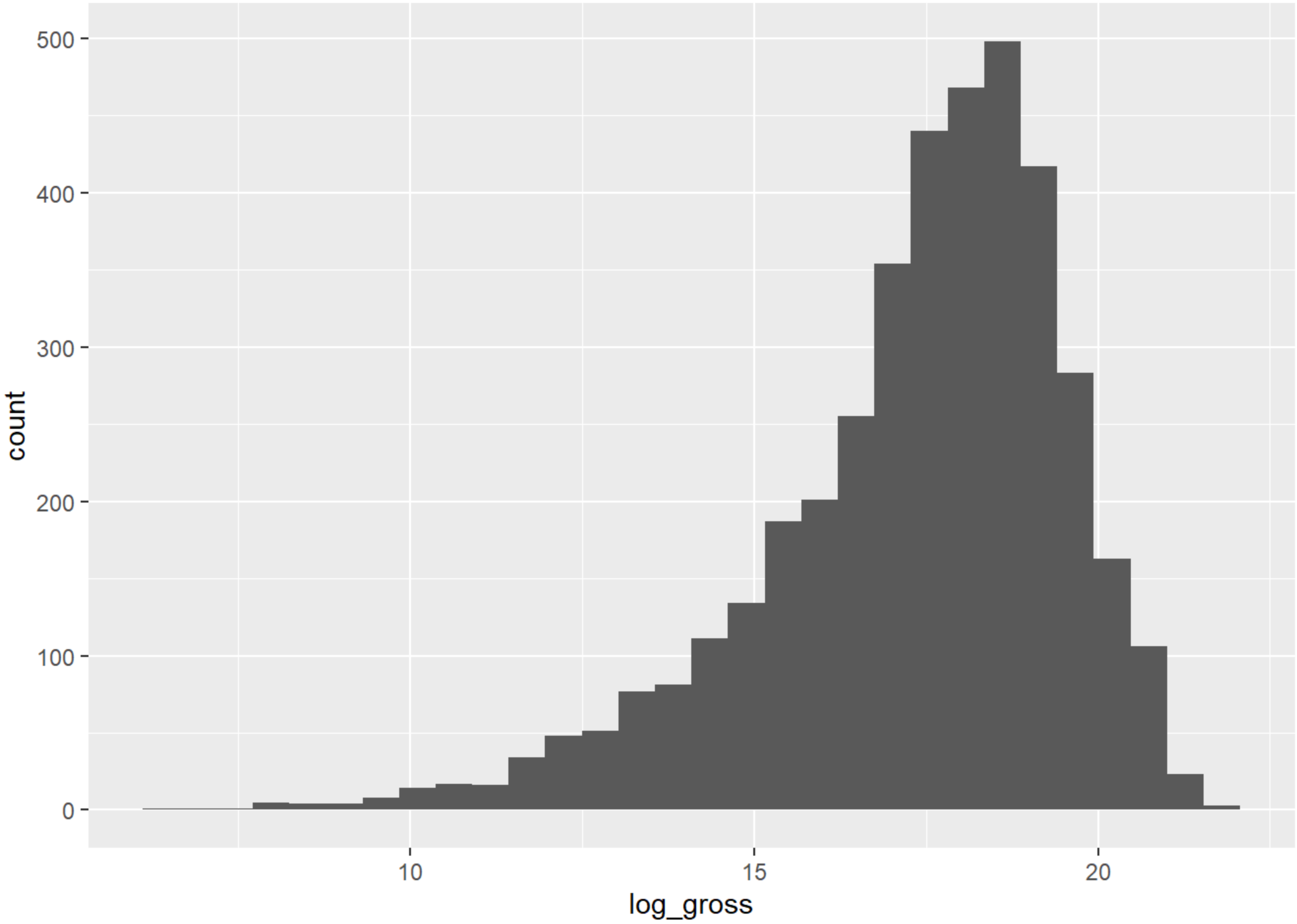
```
## Warning: Removed 3668 rows containing non-finite values (`stat_bin()`).
```



```
movies %>%
  mutate(log_gross = log(gross)) %>%
  ggplot(aes(x = log_gross)) +
  geom_histogram()
```

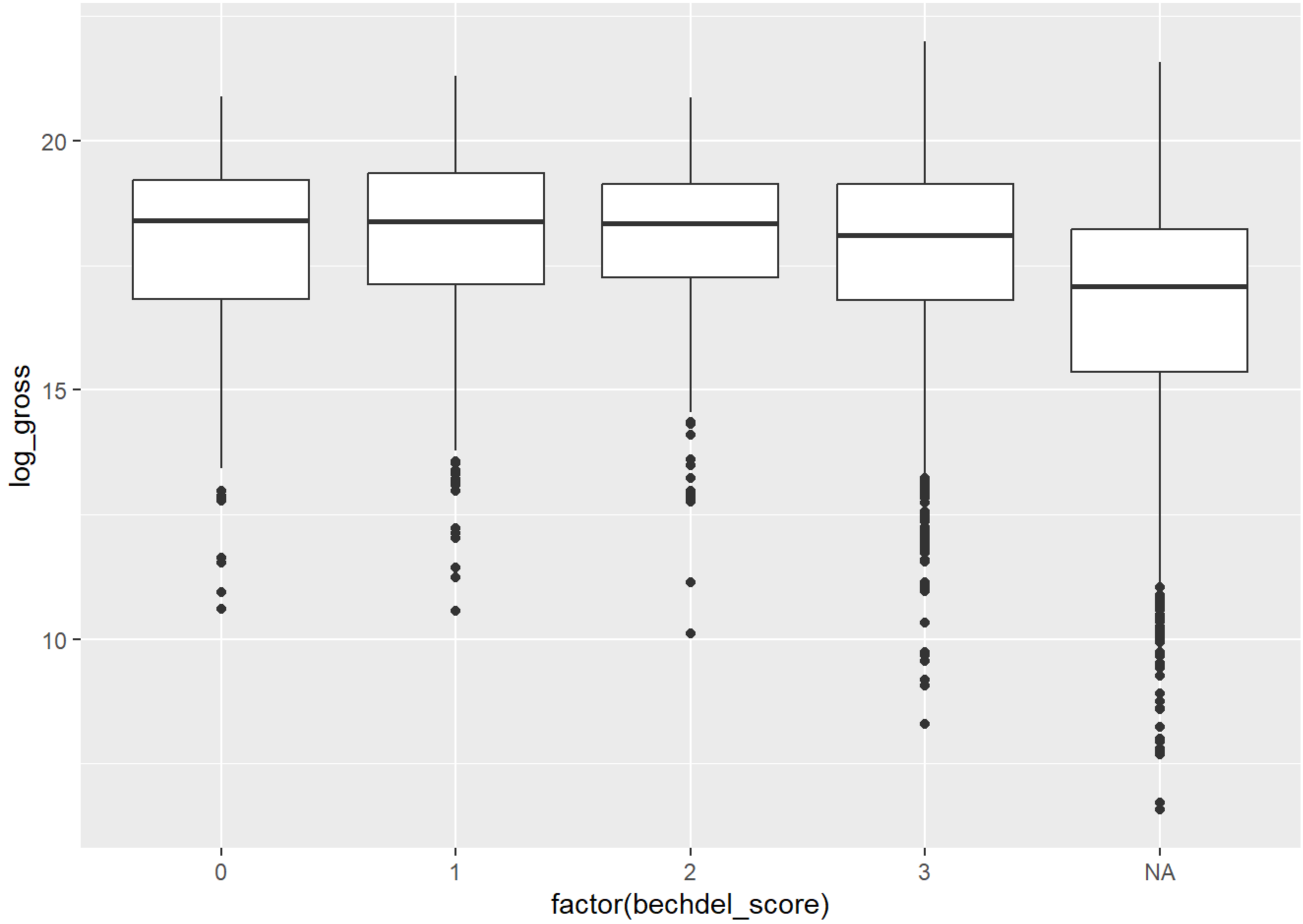
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 3668 rows containing non-finite values (`stat_bin()`).
```



```
movies %>%
  mutate(log_gross = log(gross)) %>%
  ggplot(aes(x = factor(bechdel_score),y = log_gross)) +
  geom_boxplot()
```

```
## Warning: Removed 3668 rows containing non-finite values (`stat_boxplot()`).
```

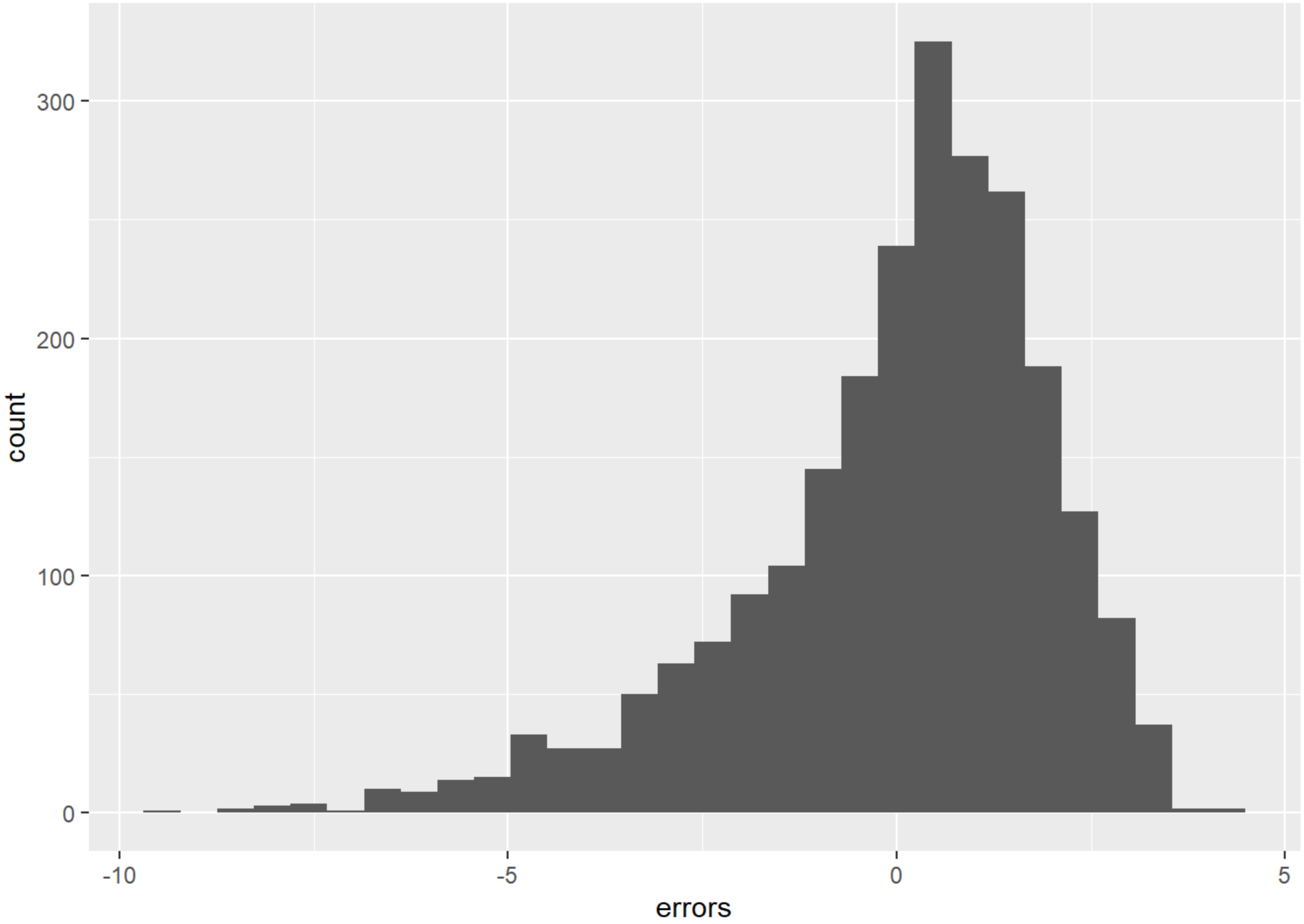


```
movies_analysis <- movies %>%
  mutate(log_gross = log(gross)) %>%
  drop_na(log_gross,bechdel_score)

model_gross_bechdel <- lm(formula = log_gross ~ bechdel_score,data = movies_analysis)

summary(model_gross_bechdel)
```

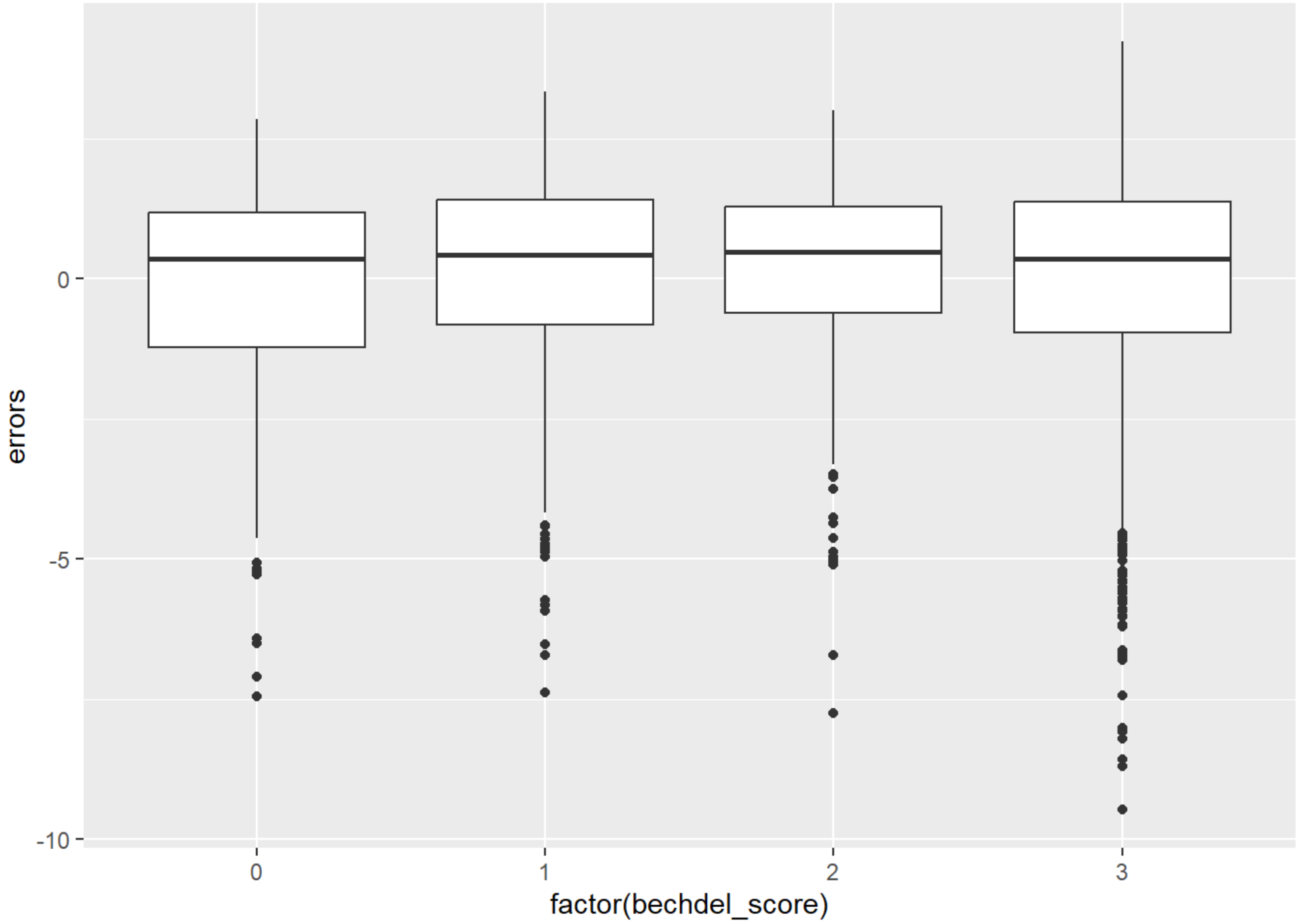
```
##
## Call:
## lm(formula = log_gross ~ bechdel_score, data = movies_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4747 -0.9196  0.3796  1.3447  4.2295
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   18.04380    0.09552 188.903   <2e-16 ***
## bechdel_score -0.09411    0.03927  -2.397   0.0166 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.981 on 2395 degrees of freedom
## Multiple R-squared:  0.002393,   Adjusted R-squared:  0.001976
## F-statistic: 5.744 on 1 and 2395 DF,  p-value: 0.01662
```

```
movies_analysis %>%
  mutate(preds = predict(model_gross_bechdel)) %>%
  mutate(errors = log_gross - preds) %>%
  ggplot(aes(x = errors)) +
  geom_histogram()
```

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



```
movies_analysis %>%
  mutate(preds = predict(model_gross_bechdel)) %>%
  mutate(errors = log_gross - preds) %>%
  ggplot(aes(x = factor(bechdel_score),y = errors)) +
  geom_boxplot()
```



```
movies_analysis %>%
  mutate(preds = predict(model_gross_bechdel)) %>%
  mutate(errors = log_gross - preds) %>%
  summarise(rmse = sqrt(mean(errors^2)))
```

```
## # A tibble: 1 × 1
##    rmse
##   <dbl>
## 1  1.98
```

```r
cvRes <- NULL # Instantiate an empty object to store data from the Loop
for(i in 1:100) { # Loop 100 times
  inds <- sample(1:nrow(movies_analysis),size = round(.8*nrow(movies_analysis)),replace = F)

  train <- movies_analysis %>% slice(inds)
  test <- movies_analysis %>% slice(-inds)

  m <- lm(formula = log_gross ~ bechdel_score,data = train)

  test$preds <- predict(m,newdata = test)

  e <- test$log_gross - test$preds
  se <- e^2
  mse <- mean(se)
  rmse <- sqrt(mse)
  cvRes <- c(cvRes,rmse)
}

mean(cvRes)
```

```
## [1] 1.976464
```

> Based on the regression output, we would conclude against my theory. Specifically, the output indicates that each additional point in the Bechdel score corresponds to a decline in the movie’s gross by 0.094 logged dollars. We are over 98% confident that this result is not zero.