Homework 2 Part 2

AUTHOR
Thomas Zwiller

Analytic Tasks

The tasks below should live in a separate qmd from your prep work. You will want to `load` your data from the previous step.

```
#loading in my final product version (the clean, aggregated version, so 100 rows)
load("/Users/TomTheIntern/Desktop/Mendoza/Mod 1/Wrangling/Homework 2/final_movie_product.
#and then the cleaned, un-aggregated version (1000 rows)
load("/Users/TomTheIntern/Desktop/Mendoza/Mod 1/Wrangling/Homework 2/raw_movie_data.rda")
```

1. Which genre has the highest critic rating? Viewer rating?

```
#created a composite average rating here, so that both ratings were captured in the set a
raw_data$average_rating <- (raw_data$rating_first_watch + raw_data$rating_second_watch) /

#pulling critics only so I can cleanly aggregate it
critic_data <- raw_data[raw_data$reviewer_type == "Critic", ]
#aggregating the critic data
critic_genre_rating <- aggregate(critic_data$average_rating ~ critic_data$genre, data = c
#I then found this link https://www.reddit.com/r/rstats/comments/d8c0ae/how_do_i_return_a
#which suggested using which max
critic_genre_rating$`critic_data$genre`[which.max(critic_genre_rating$`critic_data$averag
```

```
[1] "Comedy"
```

```
#pulling the viewer data just like I did the critic
viewer_data <- raw_data[raw_data$reviewer_type == "Viewer", ]
#aggregating the viewer data just like the critic
viewer_genre_rating <- aggregate(viewer_data$average_rating ~ viewer_data$genre, data = v
viewer_genre_rating$`viewer_data$genre`[which.max(viewer_genre_rating$`viewer_data$averag
```

```
[1] "Drama"
```

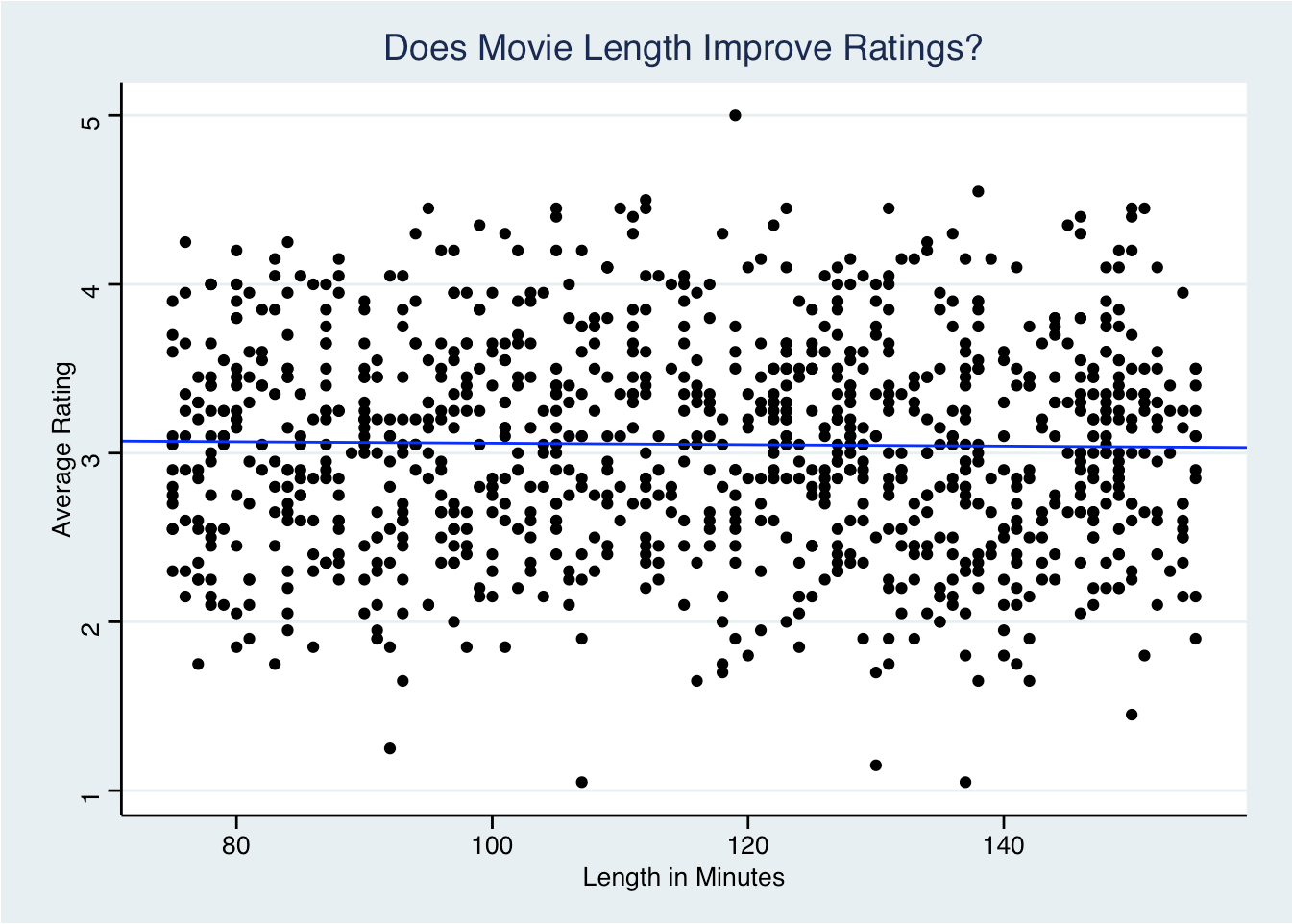
2. What is the relationship between movie length and average rating?

```
#I just ended up writing a function to make my charting a little bit easier because I'm k
lazy_plot_function <- function(data_name, x_plot, y_plot, x_name, y_name, title, intercep
  #imports ggplot2
  library(ggplot2)
  #imports gg themes
  library(ggthemes)
  #requires the data name, what to plot on the x, what to plot on the y
```

```
  chart_output <- ggplot(data_name, aes(x = x_plot, y = y_plot)) +
                  geom_point()+
    #then the x label, y label and title
                   labs(x = x_name,
                        y = y_name,
                       title = title)+
                       theme_stata()+
    #and a line of best fit to show the relationship
                  geom_abline(intercept = coef(intercept)[1], slope = coef(intercept)[2]
  return(chart_output)
}
#Cool, now that I have a function that in theory makes graphing easier, let's graph it!
#A requirement is that I need to ensure that I have a lm function calculated, so lets do
length_rating_relationship <- lm(raw_data$average_rating ~ raw_data$length_combined)
summary(length_rating_relationship)
```

```
Call:
lm(formula = raw_data$average_rating ~ raw_data$length_combined)

Residuals:
     Min       1Q   Median       3Q      Max
-2.00563 -0.45904  0.00972  0.43329  1.94949

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                3.1012342  0.1005441  30.845   <2e-16 ***
raw_data$length_combined -0.0004262  0.0008481  -0.503    0.615
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6296 on 998 degrees of freedom
Multiple R-squared:  0.0002531, Adjusted R-squared:  -0.0007487
F-statistic: 0.2526 on 1 and 998 DF,  p-value: 0.6153
```

```
#and plot the function
lazy_plot_function(raw_data, raw_data$length_combined, raw_data$average_rating,
                   "Length in Minutes", "Average Rating", "Does Movie Length Improve Rati
                   length_rating_relationship)
```



Just visually looking at the data, it's pretty safe to say that no, there is not really a correlation between a movie's length and the average rating it receives. The sample is pretty solid in terms of size, but the data is just a bit too random.

3. What is the relationship between the date delta and average rating?

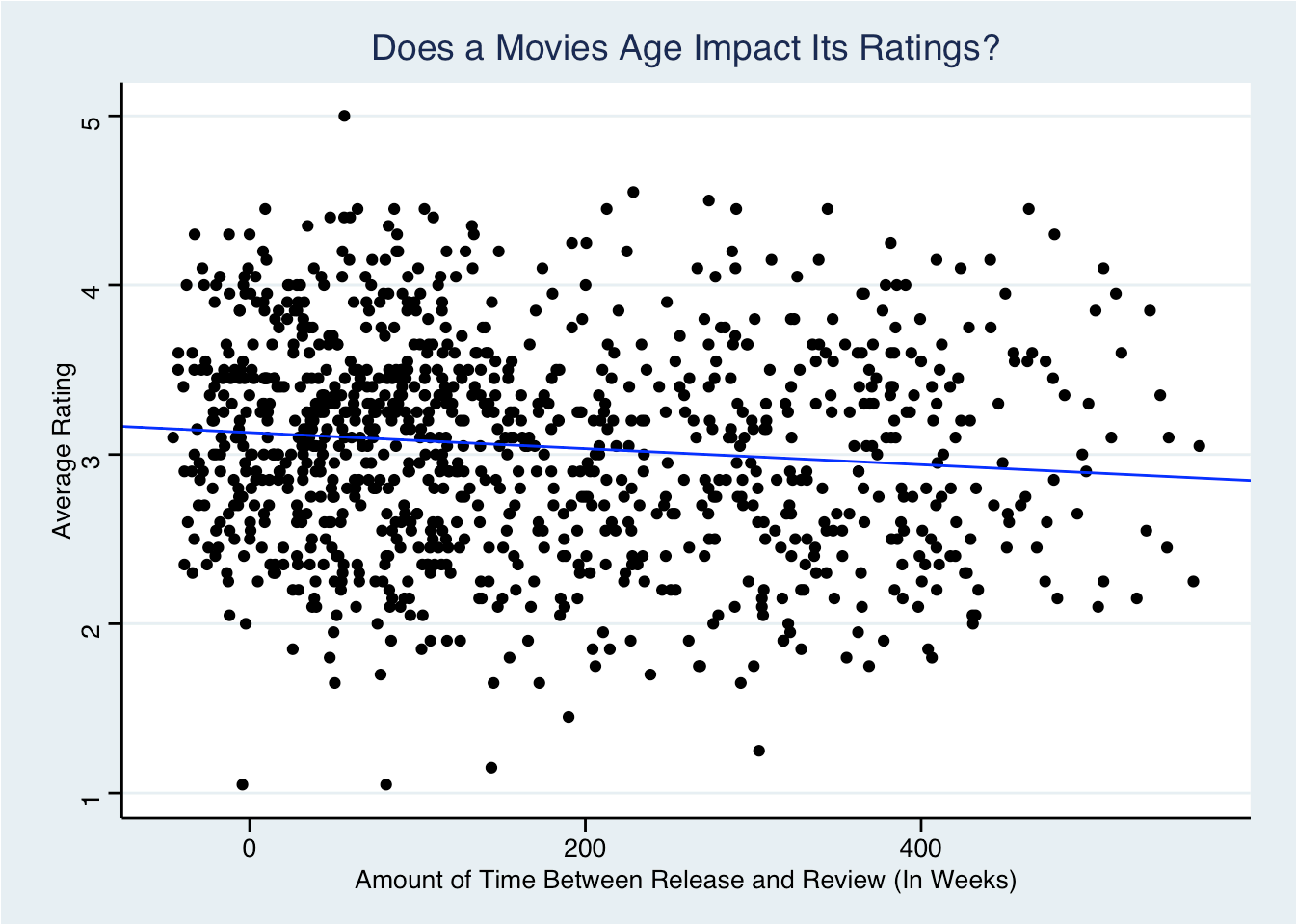
```
#Quickly make the relationship model
date_delta_rating <- lm(raw_data$average_rating ~ raw_data$date_delta)
summary(date_delta_rating)
```

```
Call:
lm(formula = raw_data$average_rating ~ raw_data$date_delta)

Residuals:
     Min       1Q   Median       3Q      Max
-2.08252 -0.45488  0.00271  0.41280  1.89639

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          3.1304822  0.0300349 104.228  < 2e-16 ***
raw_data$date_delta -0.0004761  0.0001365  -3.487 0.000509 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.6259 on 998 degrees of freedom
Multiple R-squared:  0.01204,   Adjusted R-squared:  0.01105
F-statistic: 12.16 on 1 and 998 DF,  p-value: 0.000509
```

```
#Time to graph! (I made it, might as well use it...)
lazy_plot_function(raw_data, raw_data$date_delta, raw_data$average_rating,
                   "Amount of Time Between Release and Review (In Weeks)",
                   "Average Rating", "Does a Movies Age Impact Its Ratings?",
                   date_delta_rating)
```

```
Don't know how to automatically pick scale for object of type <difftime>.
Defaulting to continuous.
```



Another one where you can say that there is a limited relationship between the rating and when the movie came out. There is a slight negative correlation, which does suggest there is a slight impact to ratings based on age, but it's so small that we would need to either collect data or just ignore the relationship because it's so small.

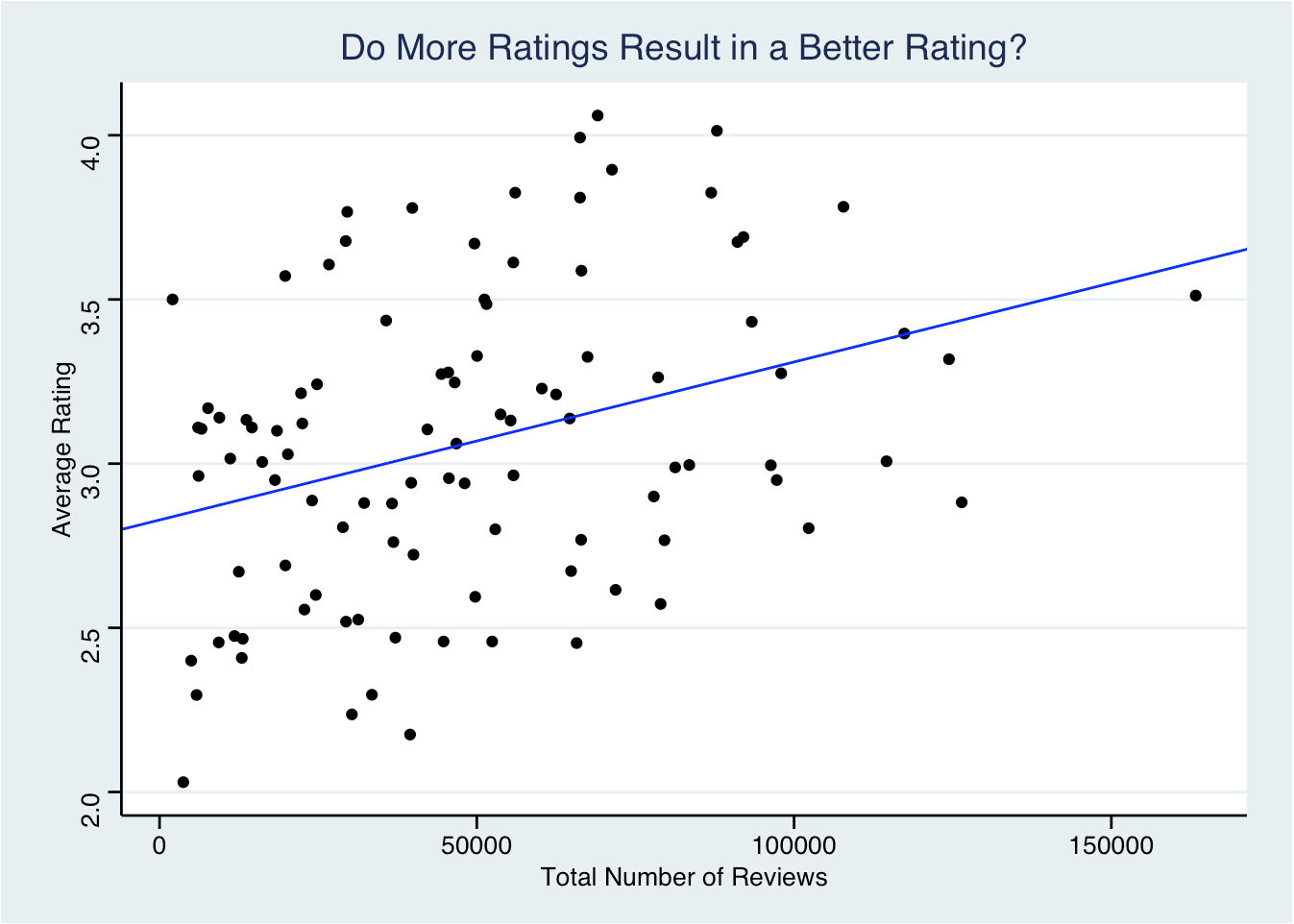
4. What is the relationship between total number of reviews and average?

```
#I had the total reviews in my final product data, but not my raw data file, which means
final_product$average_rating <- (final_product$`First Watch Rating` + final_product$`Seco
```

```
#You know the drill! Find the relationship
rating_to_reviews <- lm(final_product$average_rating ~ final_product$`Total Reviews`)
summary(rating_to_reviews)
```

```
Call:
lm(formula = final_product$average_rating ~ final_product$`Total Reviews`)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.84341 -0.37444  0.00525  0.26930  0.89934 

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)    
(Intercept)                   2.828e+00  7.898e-02   35.81  < 2e-16 ***
final_product$`Total Reviews` 4.812e-06  1.337e-06    3.60 0.000502 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.437 on 98 degrees of freedom
Multiple R-squared:  0.1168,    Adjusted R-squared:  0.1078 
F-statistic: 12.96 on 1 and 98 DF,  p-value: 0.0005021
```

```
#And then plot it (Okay this might have been worth it...)
lazy_plot_function(final_product, final_product$`Total Reviews`, final_product$average_ra
                   "Total Number of Reviews",
                   "Average Rating", "Do More Ratings Result in a Better Rating?",
                   rating_to_reviews)
```



Finally! we have an R-squared rating that is greater than .02! It's just 11.6, so a relatively small relationship, but there overall is a positive correlation between the number of reviews and the average rating which suggests that people who enjoyed a movie are more likely to rate the movie favorably.

5. Which movie contains your word of interest at the highest frequency. Does that word have any relationship with average rating?

```
#Unless I'm really missing something, this feels really straight forward.
#Quickly aggregate the word count based on the movie title
word_frequency_per_movie <- aggregate(raw_data$disappointing_count~ raw_data$title, FUN =
#Make the model
#Then group the aggregated movies into a data frame named final_with_word_frequency
final_with_word_frequency <- merge(final_product,
                              word_frequency_per_movie,
                              by.x = 'Movie Title',
                              by.y = 'raw_data$title')

#first things first, I found the high count of disappointing and stored it as a variable
high_count <- max(final_with_word_frequency$`raw_data$disappointing_count`)

#I then created a data frame by comparing the values in the disappointing column to the h
most_disappointing <- final_with_word_frequency$`Movie Title`[final_with_word_frequency$`
```

```
#and then wrote out the most disappointing
most_disappointing
```

```
[1] "Legends of Winter" "Mystic Shadows"    "The Last Crusade"
[4] "Whispers of Fate"
```

```
#Then I aggregated the average rating based on the number of times the word 'disappointin
#Which resulted in the table below
frequency_average <- aggregate(average_rating ~ `raw_data$disappointing_count`, data = fi
#Quickly clean up the column names
colnames(frequency_average) <- c("Disappointing Count", "Average Rating")
frequency_average
```

```
  Disappointing Count Average Rating
1                   0       3.152853
2                   1       3.045266
3                   2       3.043149
4                   3       3.032400
5                   4       3.151073
6                   5       2.980513
```

Those movies with a didn't count of 0 achieve the highest average rating of 3.152, narrowly beating the the movie ratings that had 4 mentions of the word didn't. However, 5 mention of the word didn't wound up having the lowest rating, and the only rating to dip below 3.0.

Also worth noting. I intially did make a graph and it was hard to read. I then made a box and whisker plot but honestly it still was harder to interpret than the table. Because of small amount of rows and the easy to understand average rating, I figured keeping it simple was the better move.