EDA

Katrina Truebebach

March 16, 2019

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

Load cleaned data

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

Log

Note: when graphing, better to use the non-log version of a variable and add scale_x_continuous() and scale_y_continuous().
However, when we are graphing means, this is not possible. Would graph log(mean(var)) rather than mean(log(var)), which is what we prefer.

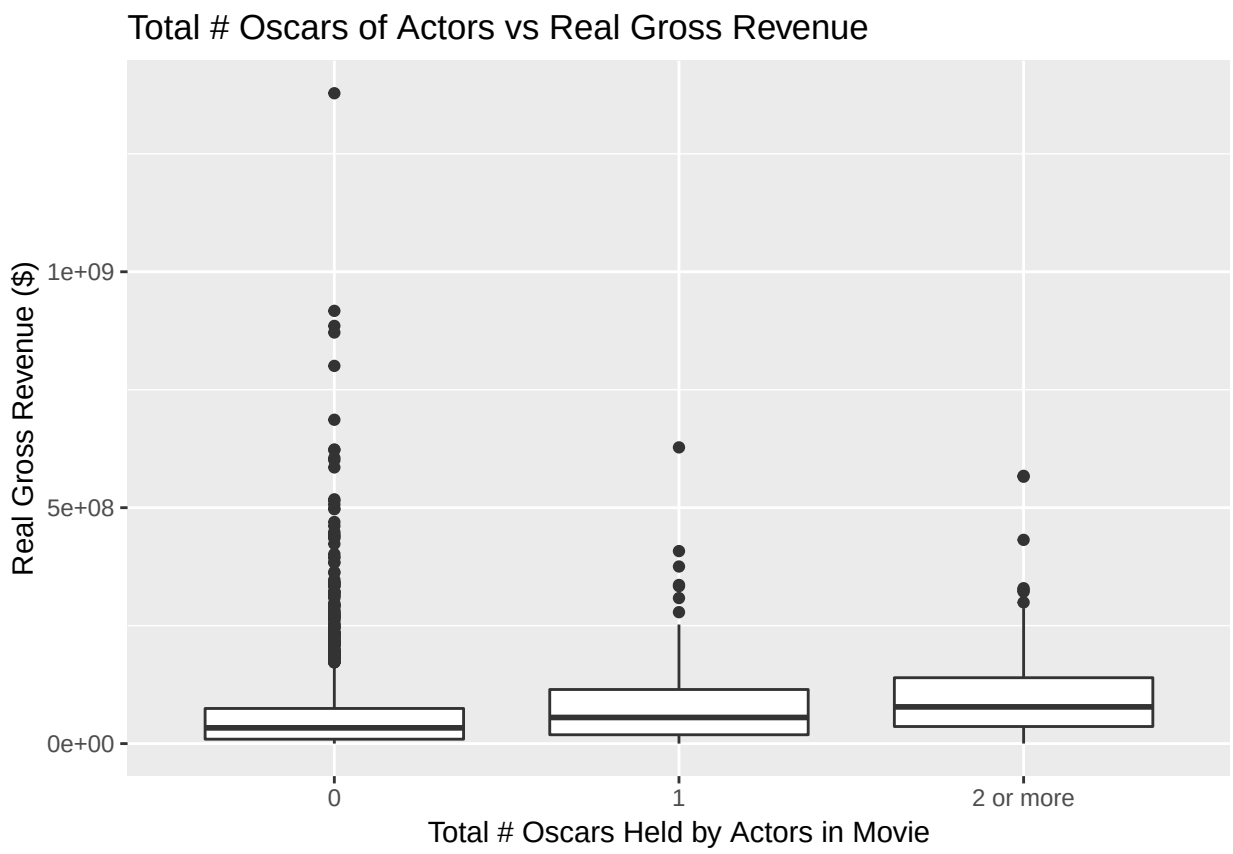
```
train <- train %>%
  mutate_at(vars(real_gross, real_budget, director_facebook_likes, cast_total_facebook_likes, imdb_score
```

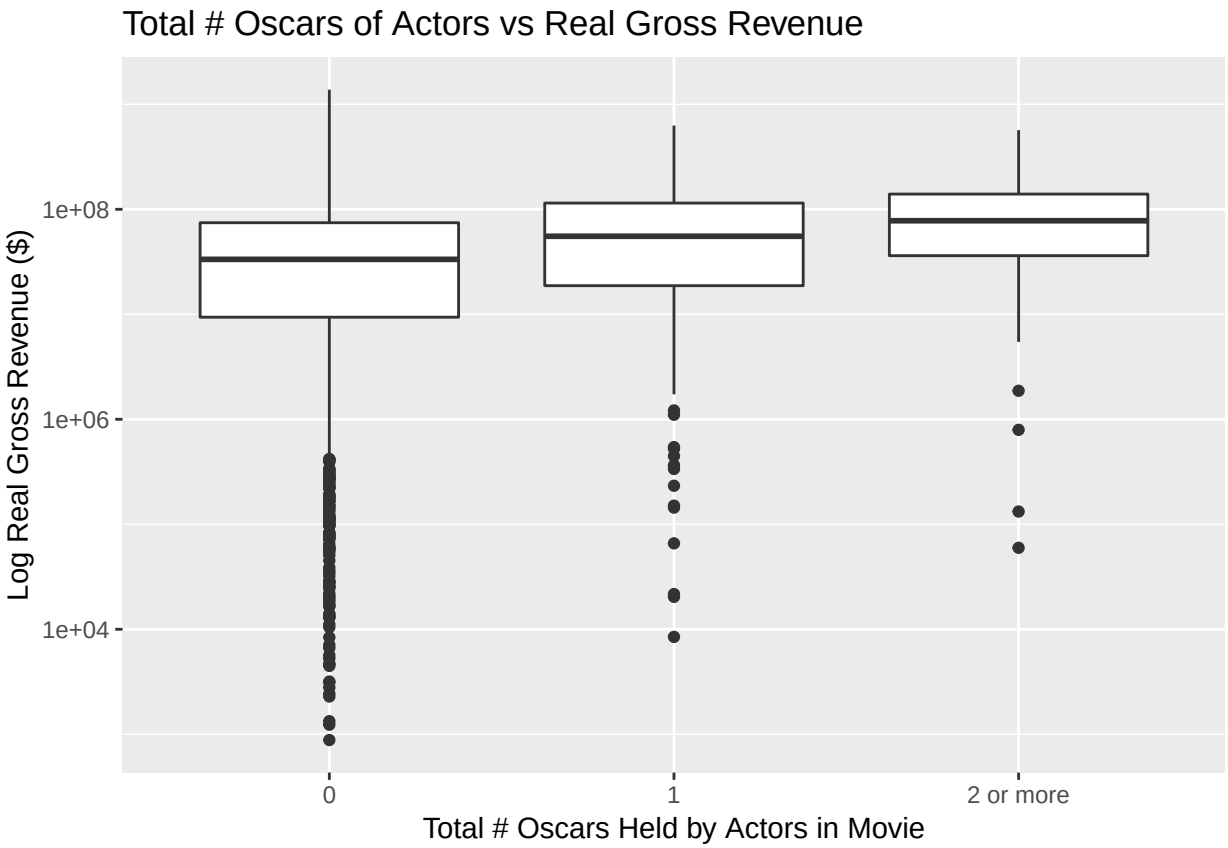
Oscars

Graph number of Oscars for actors and directors against real revenue. Boxplot and bar plot (average revenue)
Maybe linear relationships. Unclear.

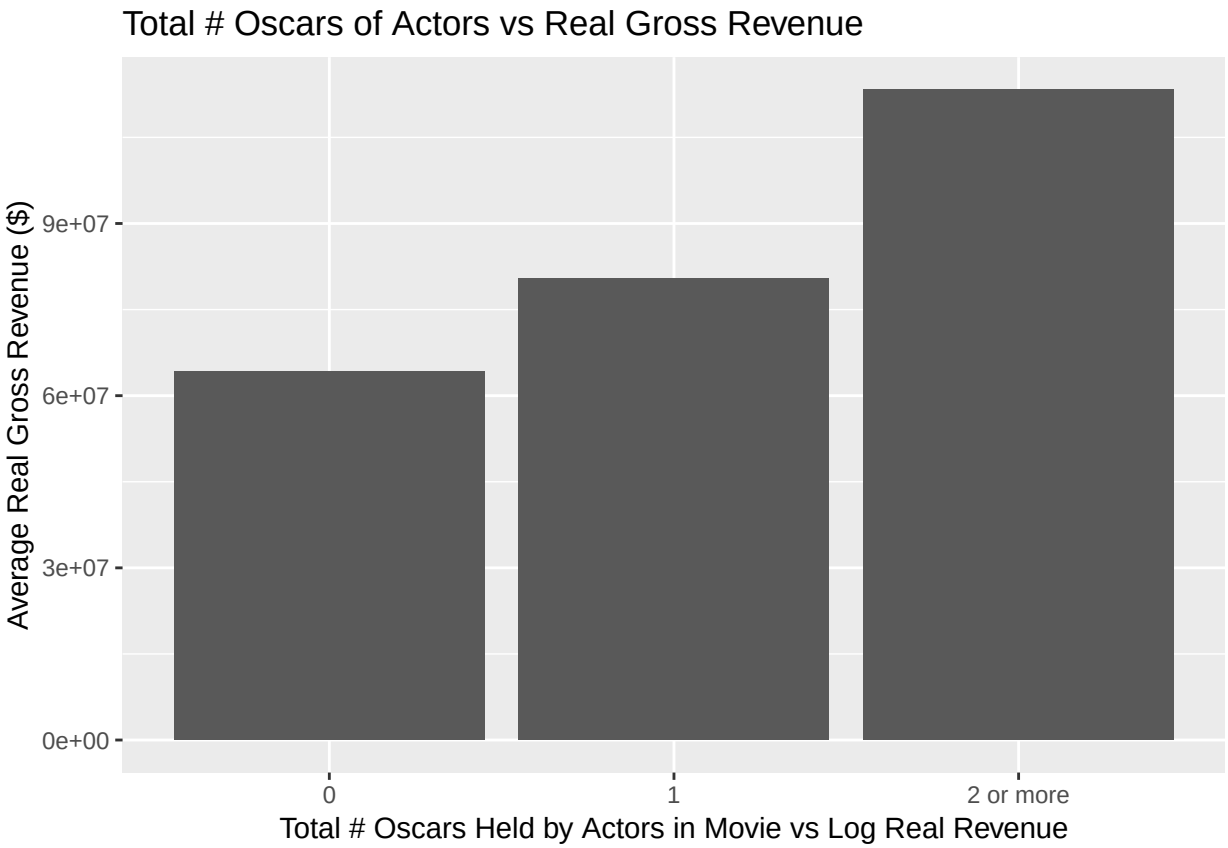
```
# Versions of data with average revenue by number of oscars
train_oscar_actor <- train %>%
  group_by(total_oscars_actor) %>%
  summarise(avg_real_gross = mean(real_gross),
            avg_real_gross_log = mean(real_gross_log))
train_oscar_director <- train %>%
  group_by(total_oscars_director) %>%
  summarise(avg_real_gross = mean(real_gross),
            avg_real_gross_log = mean(real_gross_log))

# Functions to graph number of Oscars held by actors in 1movie vs. real revenue
# boxplot
oscar_box <- function(df, var, title_str, x_str) {
  plt_base <- ggplot(df, aes_string(var, "real_gross")) +
    geom_boxplot()
  print(plt_base +
    labs(title = title_str, x = x_str, y = 'Real Gross Revenue ($)'))
  print(plt_base +
    labs(title = title_str, x = x_str, y = 'Log Real Gross Revenue ($)') +
      scale_y_log10())
}

# bar graph
# can't do scale_y_log10() with average
oscar_bar <- function(df, var, title_str, x_str) {
  print(ggplot(df, aes_string(var, "avg_real_gross")) +
```

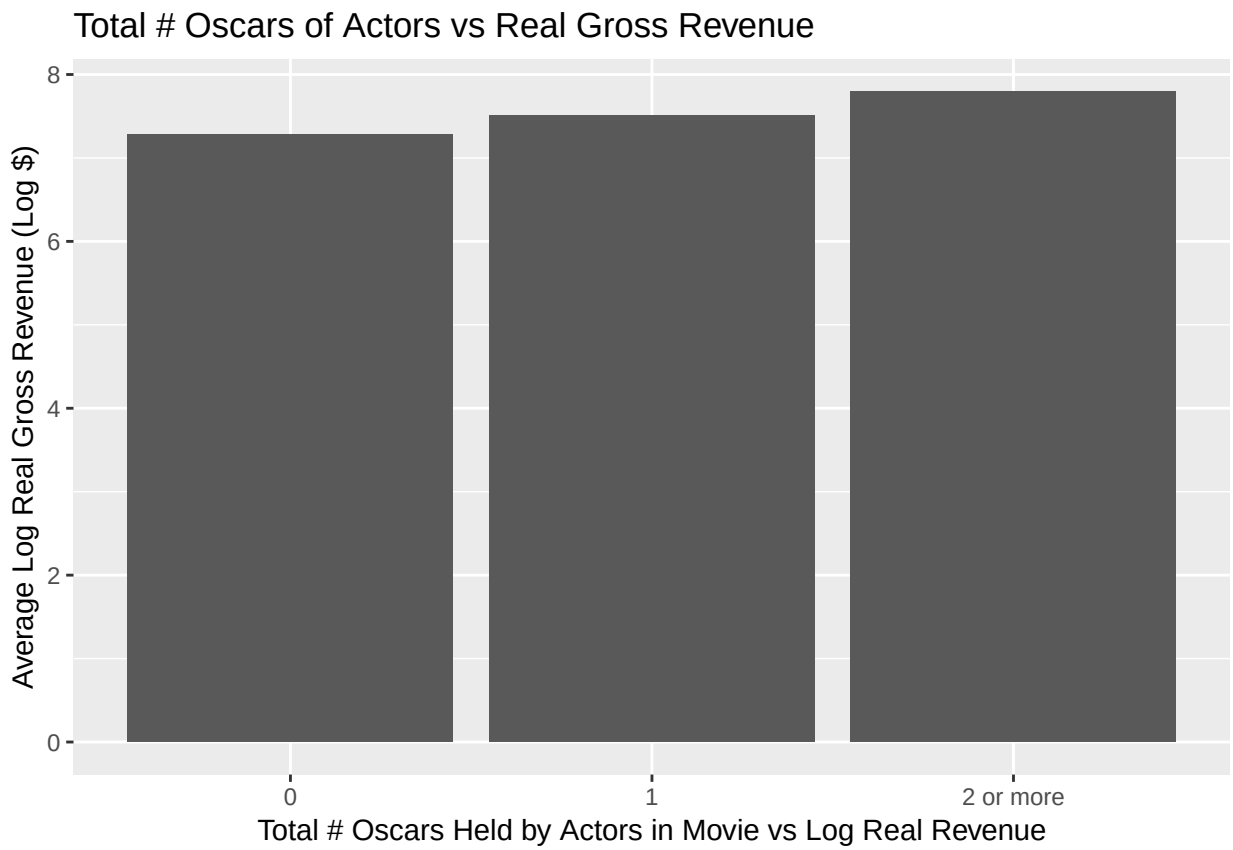
```
    geom_col() +
    labs(title = title_str, x = x_str, y = 'Average Real Gross Revenue ($)'))
  print(ggplot(df, aes_string(var, 'avg_real_gross_log')) +
    geom_col() +
    labs(title = title_str, x = x_str, y = 'Average Log Real Gross Revenue (Log $)'))
}

# actors
oscar_box(train, 'total_oscars_actor', 'Total # Oscars of Actors vs Real Gross Revenue', 'Total # Oscars
```





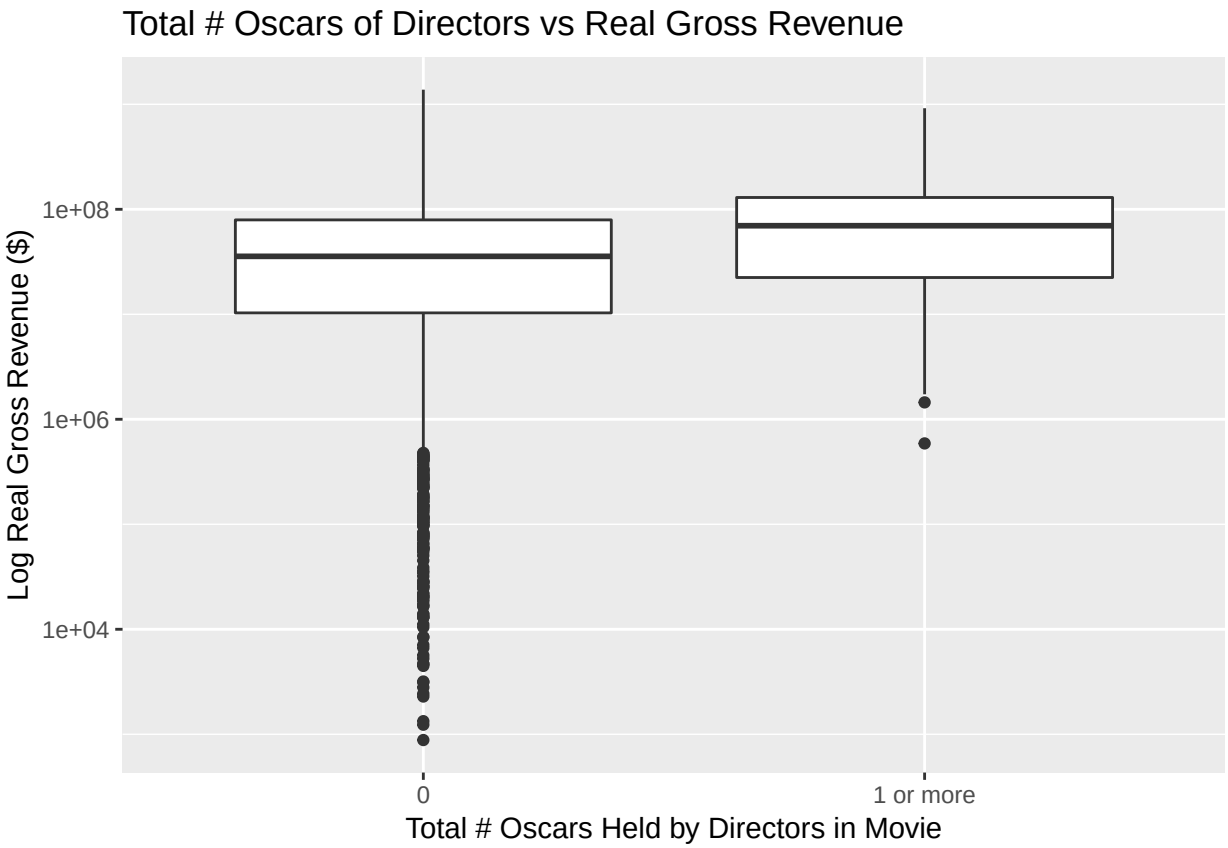

```
oscar_bar(train_oscar_actor, 'total_oscars_actor', 'Total # Oscars of Actors vs Real Gross Revenue', 'T
```


Total # Oscars of Actors vs Real Gross Revenue
Average Real Gross Revenue ($)
9e+07
6e+07
3e+07
0e+00
0
1
2 or more
Total # Oscars Held by Actors in Movie vs Log Real Revenue

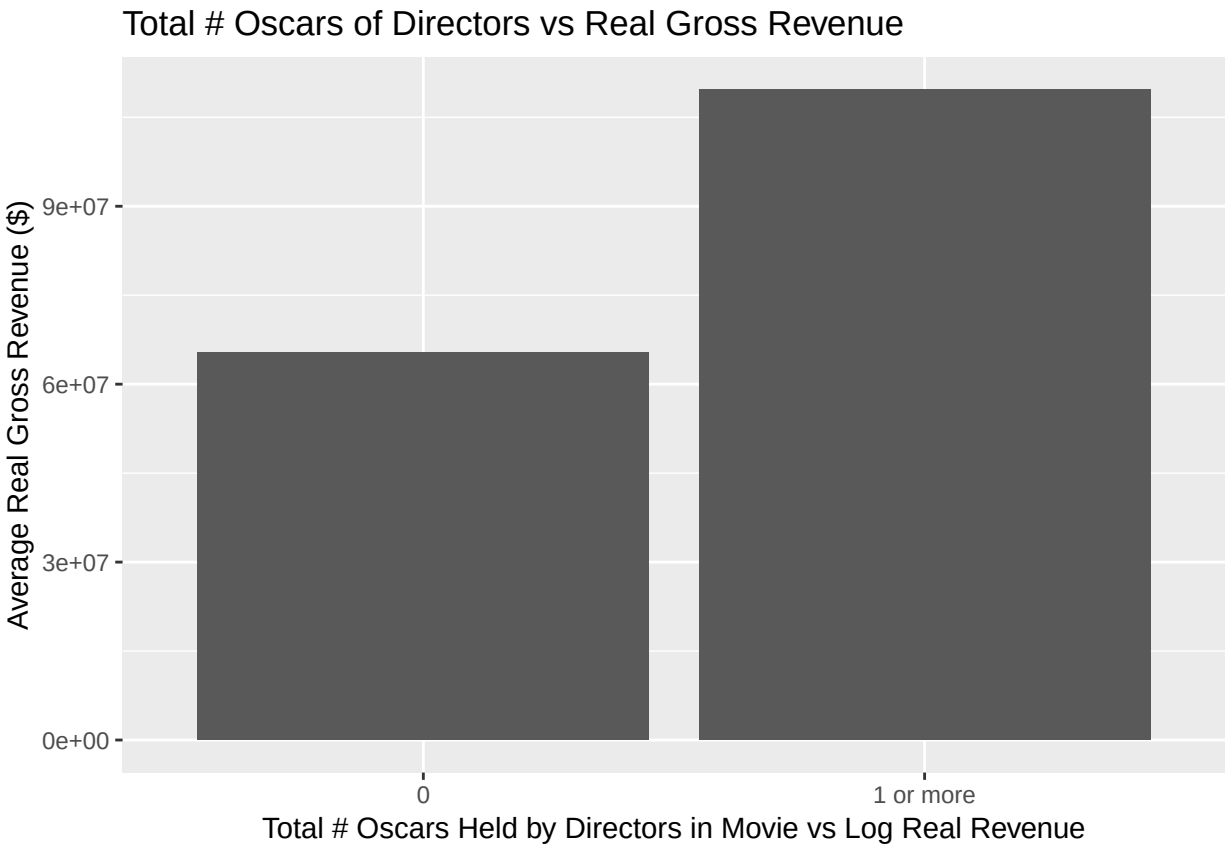


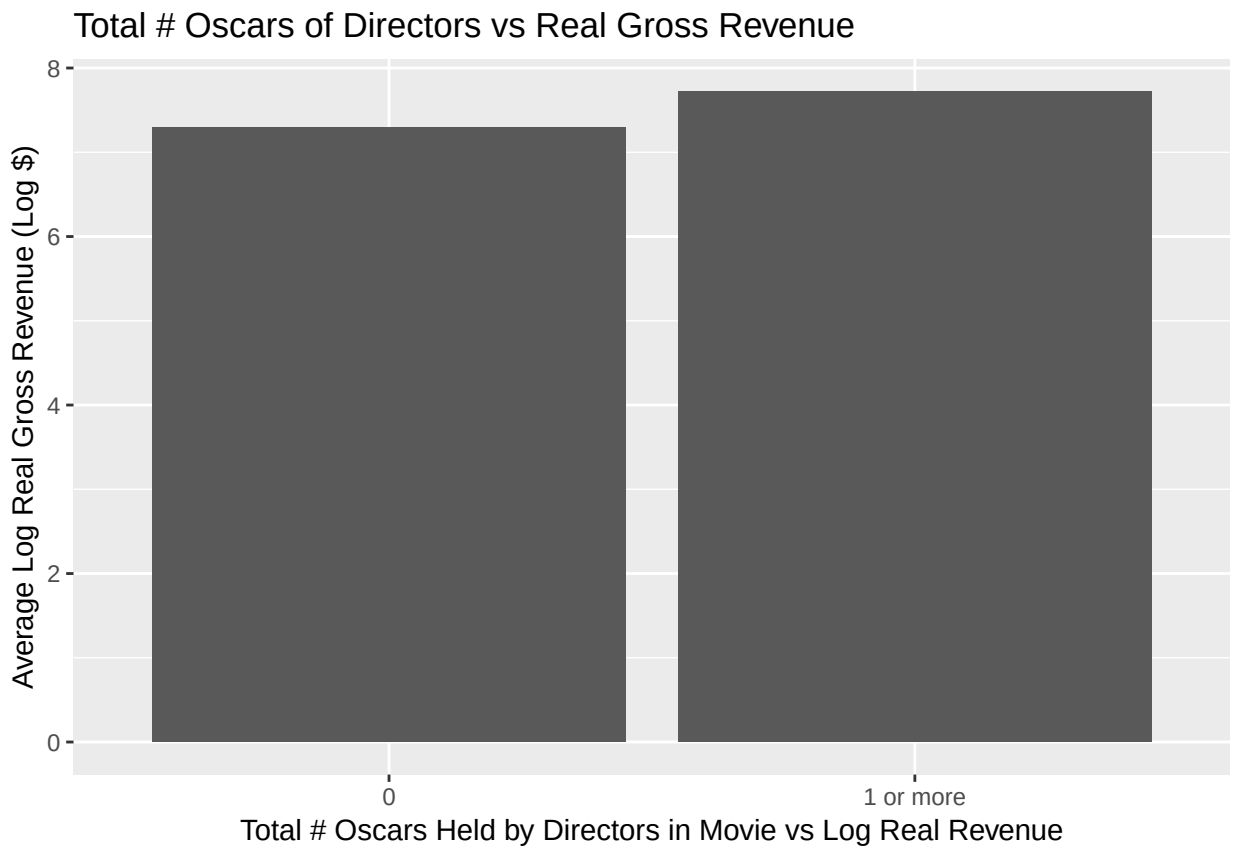

```
# directors
oscar_box(train, 'total_oscars_director', 'Total # Oscars of Directors vs Real Gross Revenue', 'Total #
```


Total # Oscars of Directors vs Real Gross Revenue
Real Gross Revenue ($)
1e+09
5e+08
0e+00
0
1 or more
Total # Oscars Held by Directors in Movie




```
oscar_bar(train_oscar_director, 'total_oscars_director', 'Total # Oscars of Directors vs Real Gross Rev
```


Total # Oscars of Directors vs Real Gross Revenue
Average Real Gross Revenue ($)
9e+07
6e+07
3e+07
0e+00
0
1 or more
Total # Oscars Held by Directors in Movie vs Log Real Revenue



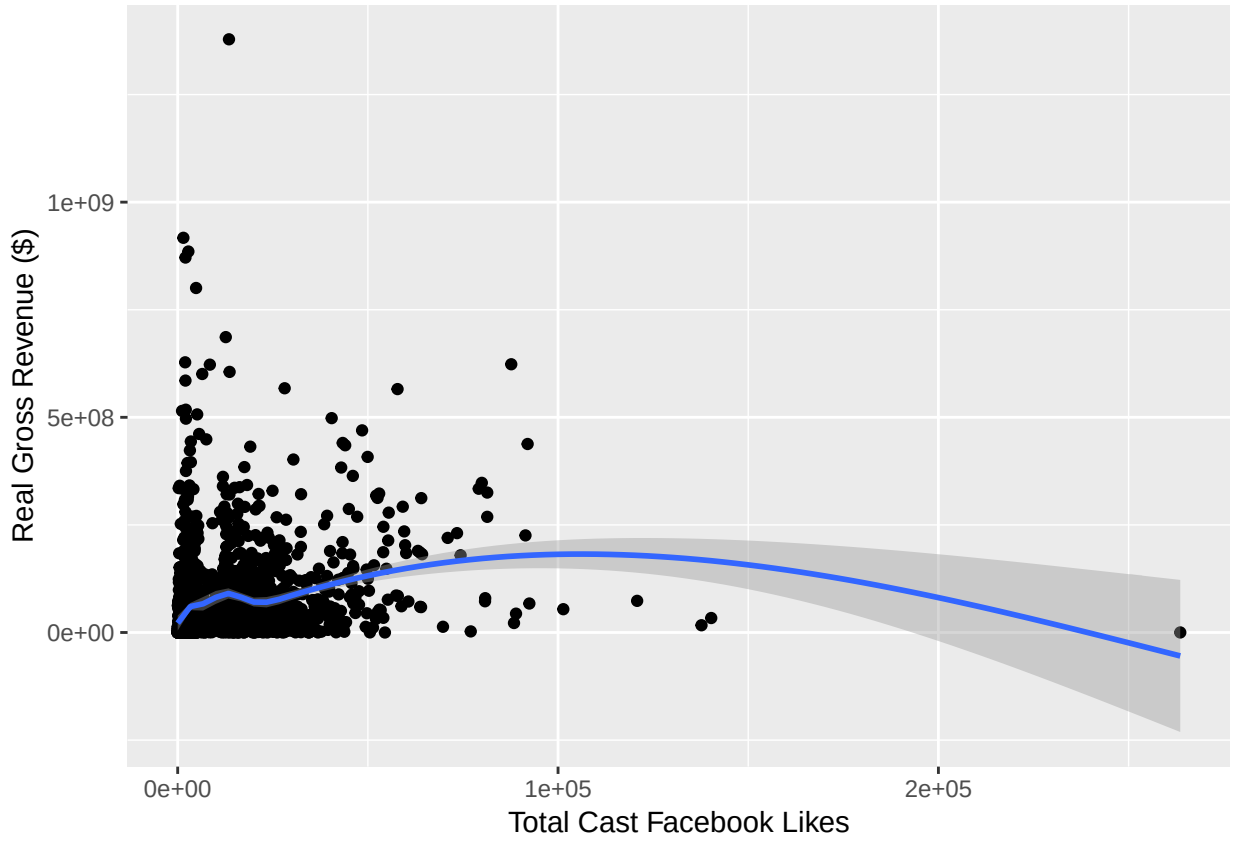
Facebook Likes

Graph total cast facebook likes and director facebook likes vs real gross revenue. Continuous, so scatter plot. Trying log versions too.

Nothing is very linear.

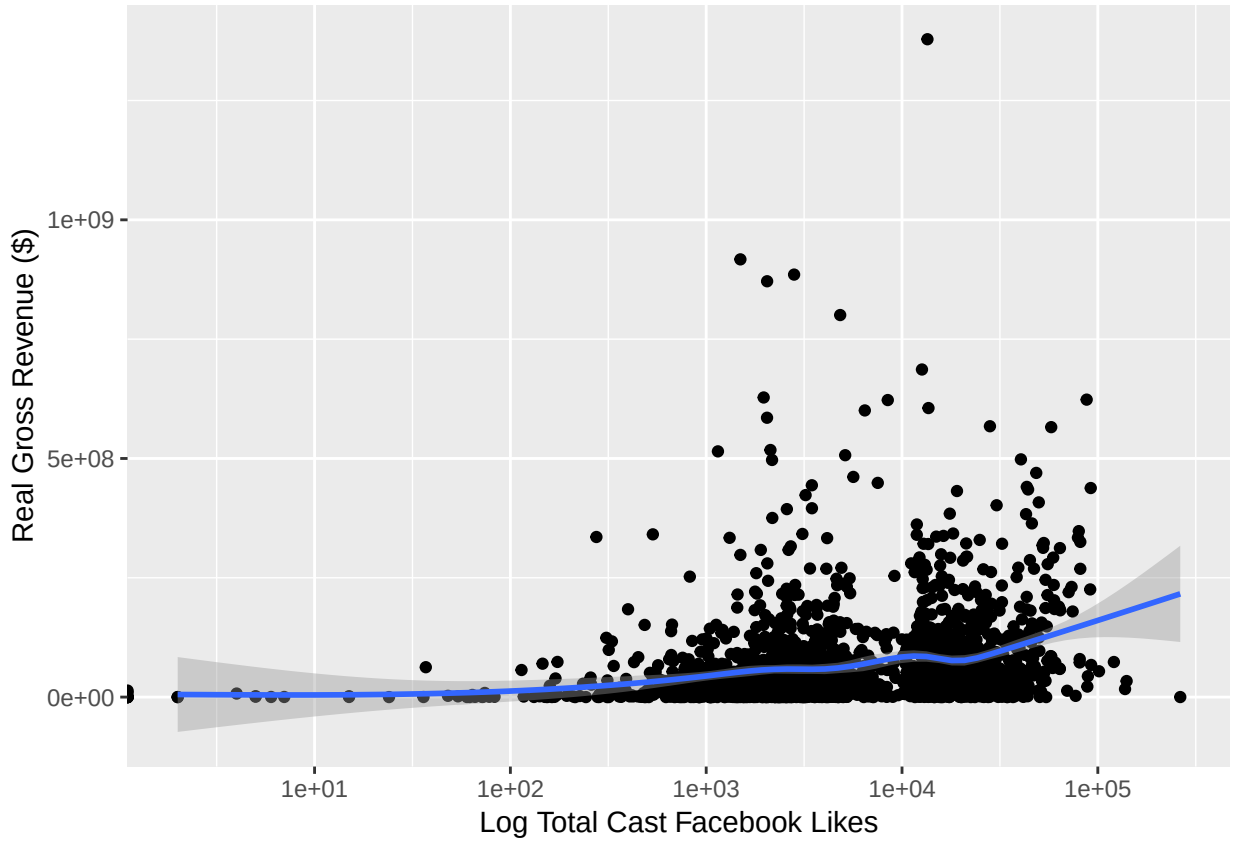
```
facebook_plot <- function(xvar, xlab) {
 base_plt <- train %>%
    ggplot(aes_string(x = xvar, y = 'real_gross')) +
    geom_point() +
    geom_smooth()

  print(base_plt +
    labs(x = xlab, y = 'Real Gross Revenue ($)'))
  print(base_plt +
          labs(x = str_c('Log ', xlab), y = 'Real Gross Revenue ($)') +
          scale_x_log10())
  print(base_plt +
          labs(x = xlab, y = 'Log Real Gross Revenue ($)') +
          scale_y_log10())
  print(base_plt +
          labs(x = str_c('Log ', xlab), y = 'Log Real Gross Revenue ($)') +
          scale_x_log10() + scale_y_log10())
}

facebook_plot('cast_total_facebook_likes', 'Total Cast Facebook Likes')
```

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

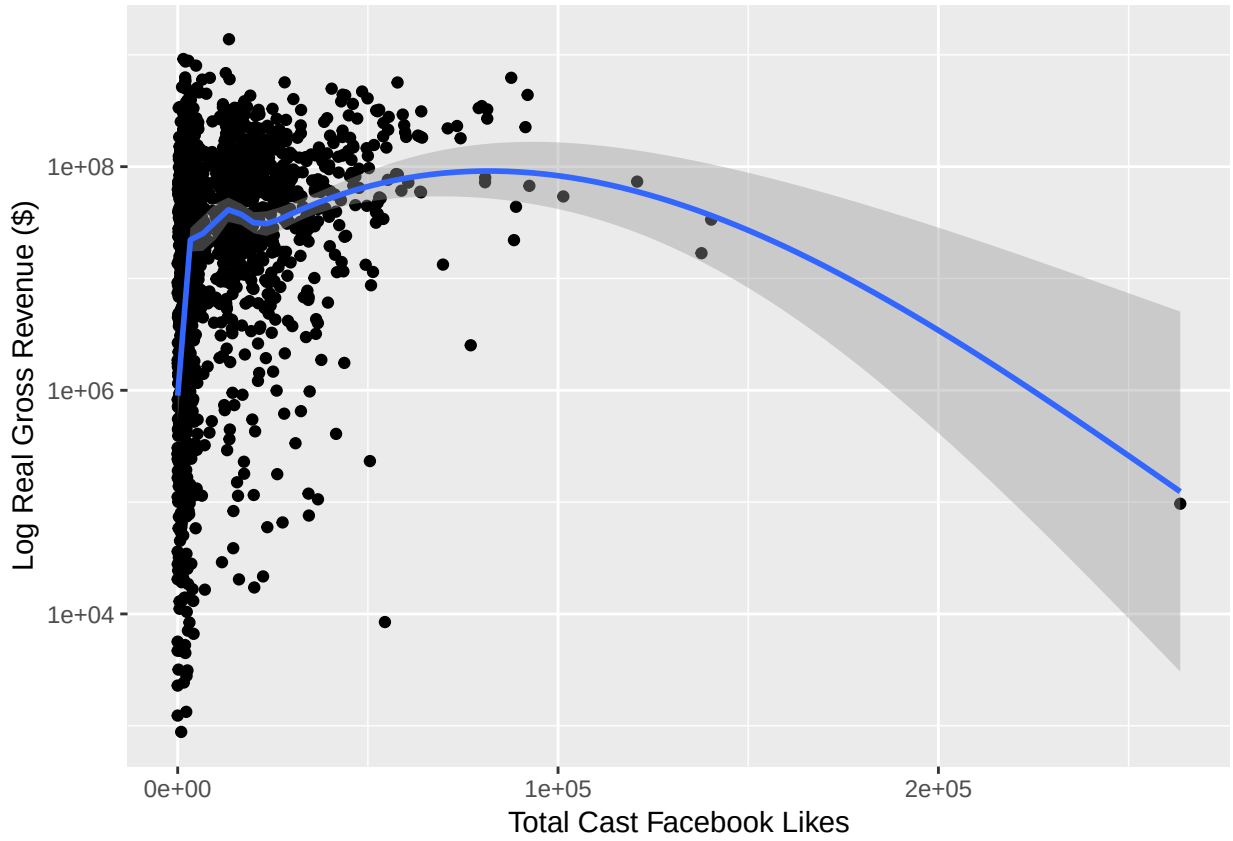



```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## Warning: Removed 8 rows containing non-finite values (stat_smooth).
```

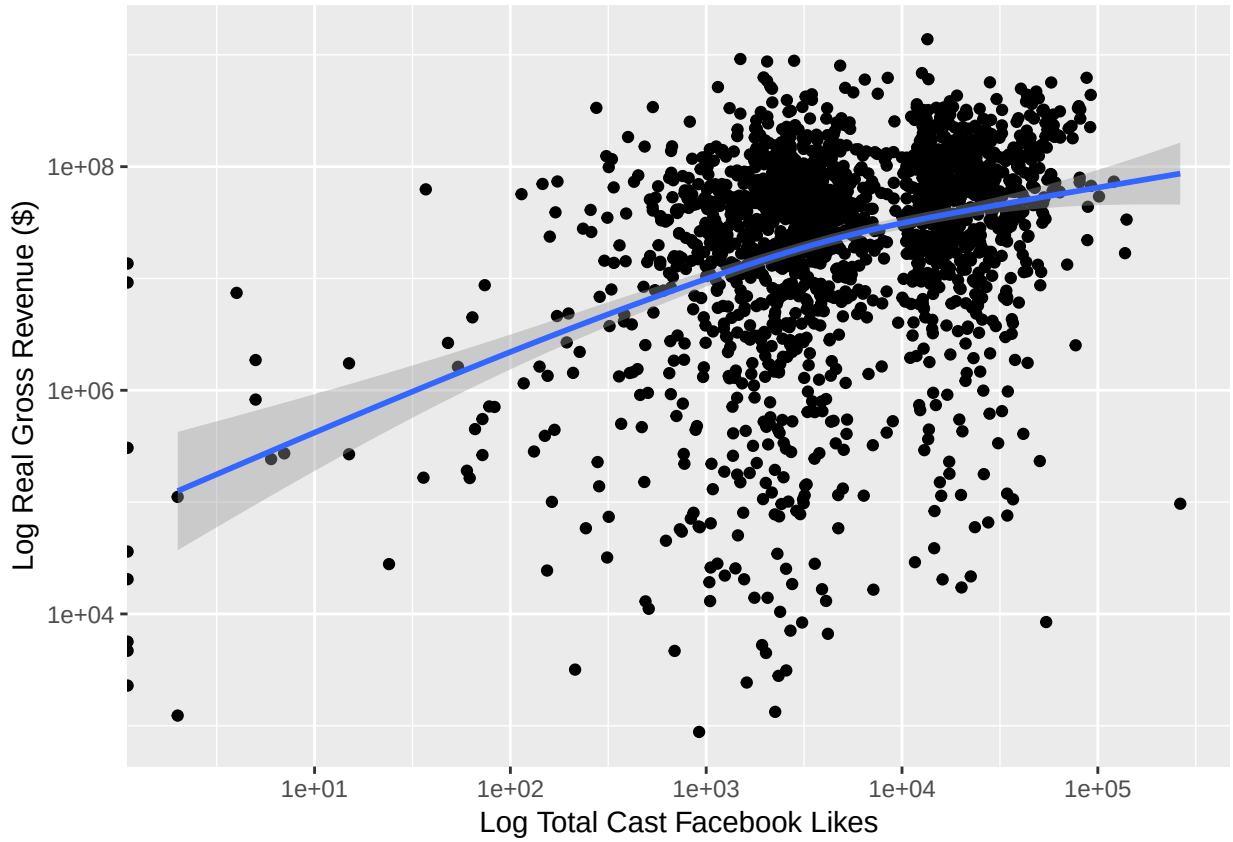



```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



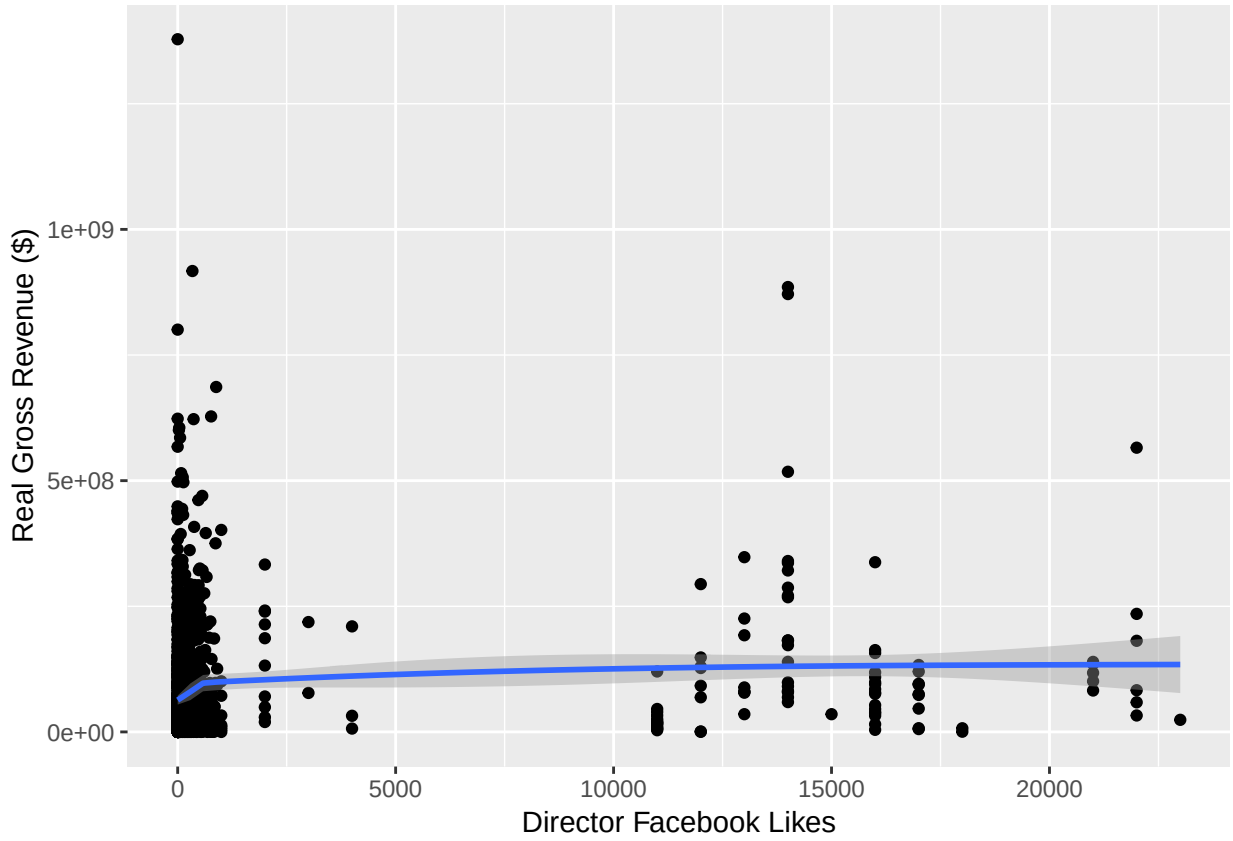

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## Warning: Removed 8 rows containing non-finite values (stat_smooth).
```




```
facebook_plot('director_facebook_likes', 'Director Facebook Likes')
```

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

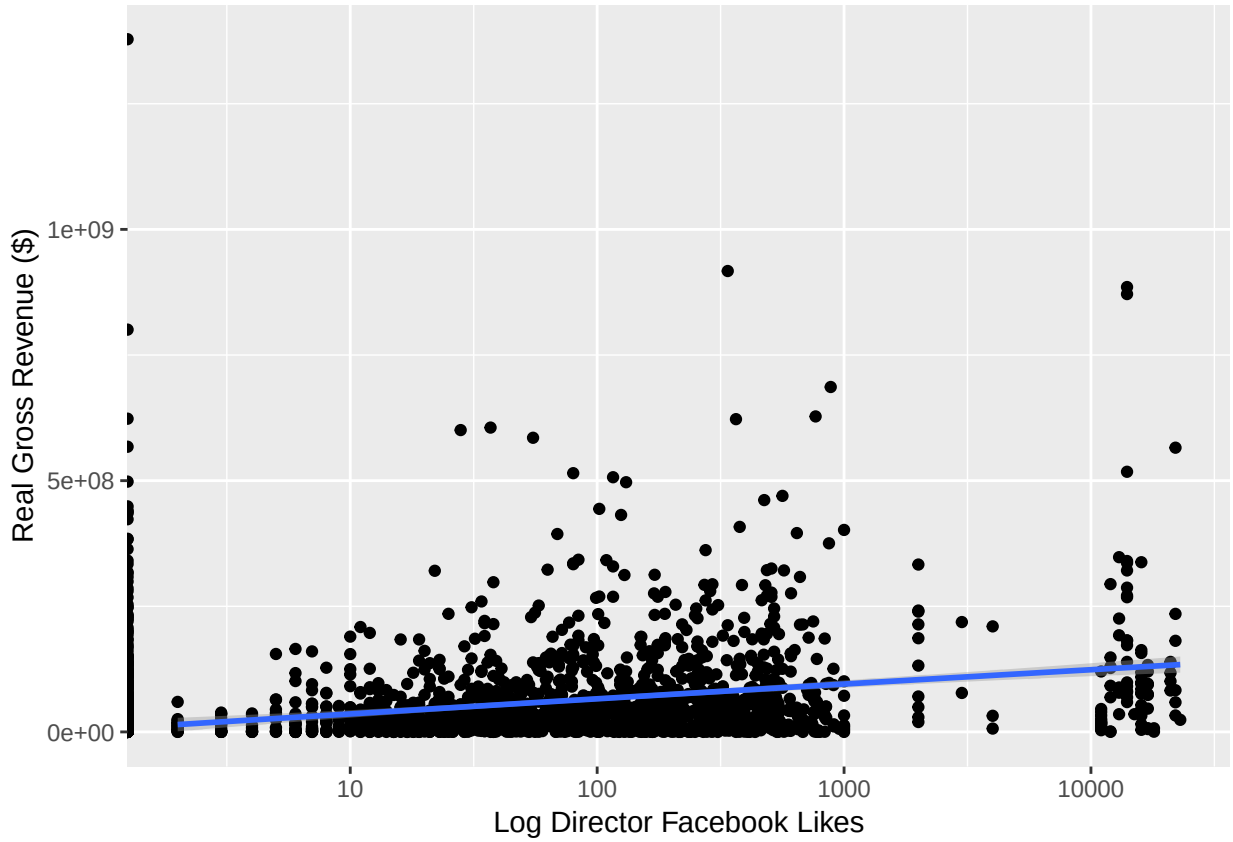



```
## Warning: Transformation introduced infinite values in continuous x-axis
```

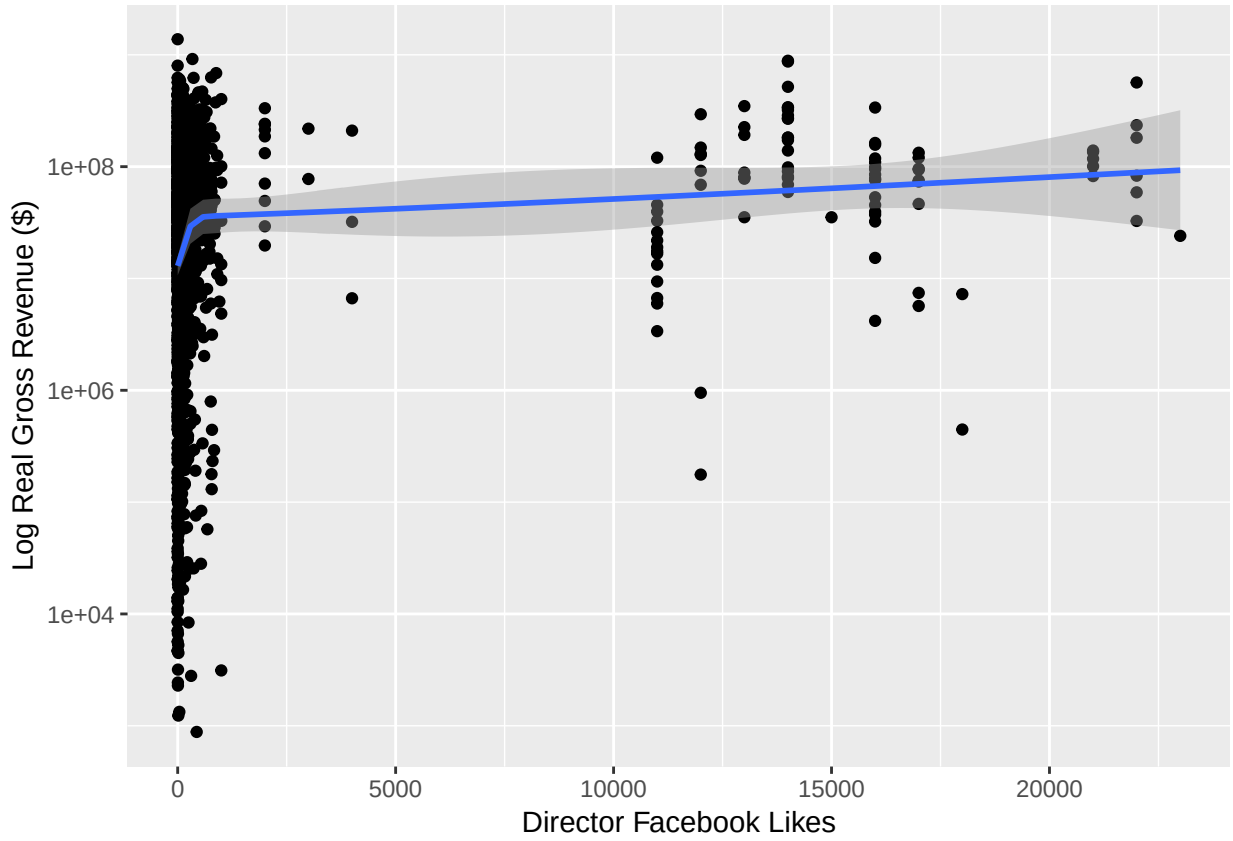
```
## Warning: Transformation introduced infinite values in continuous x-axis
```

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

```
## Warning: Removed 340 rows containing non-finite values (stat_smooth).
```




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

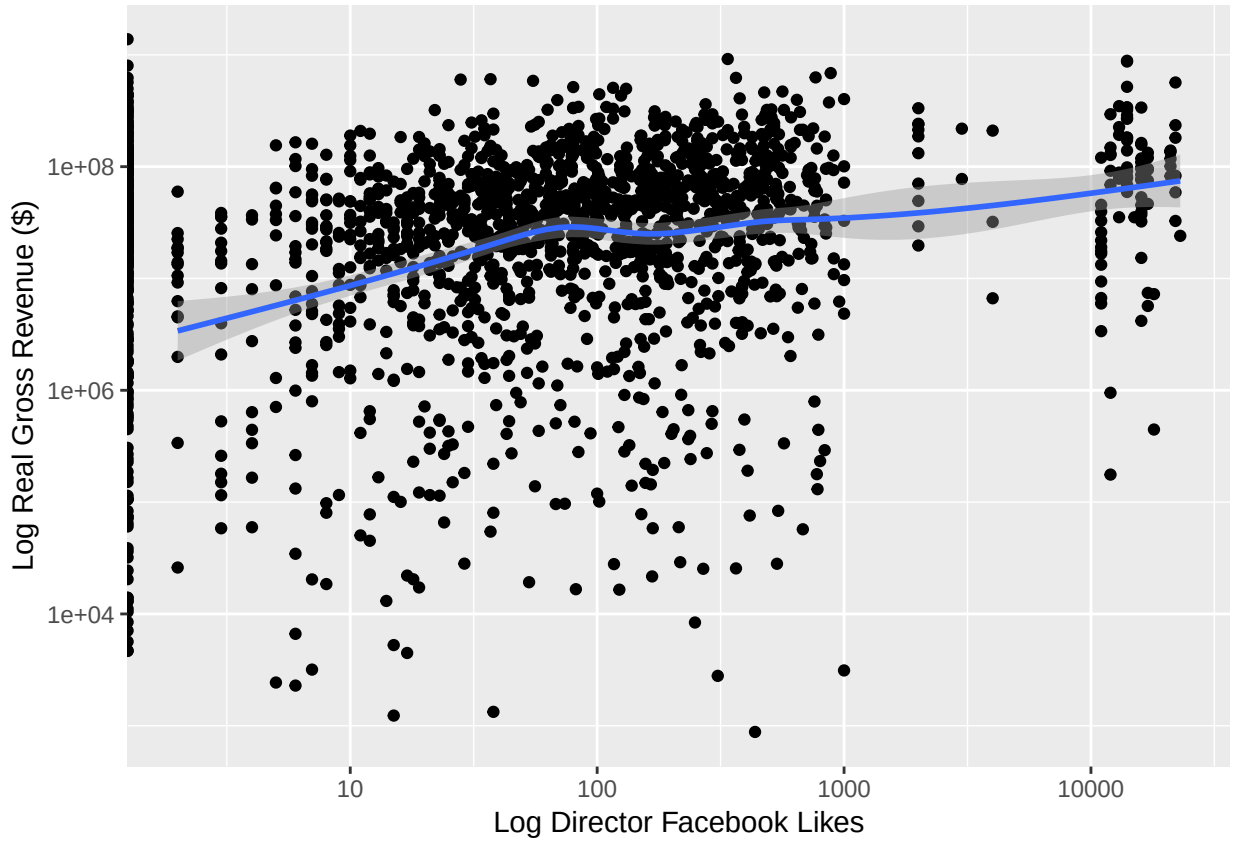



```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

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

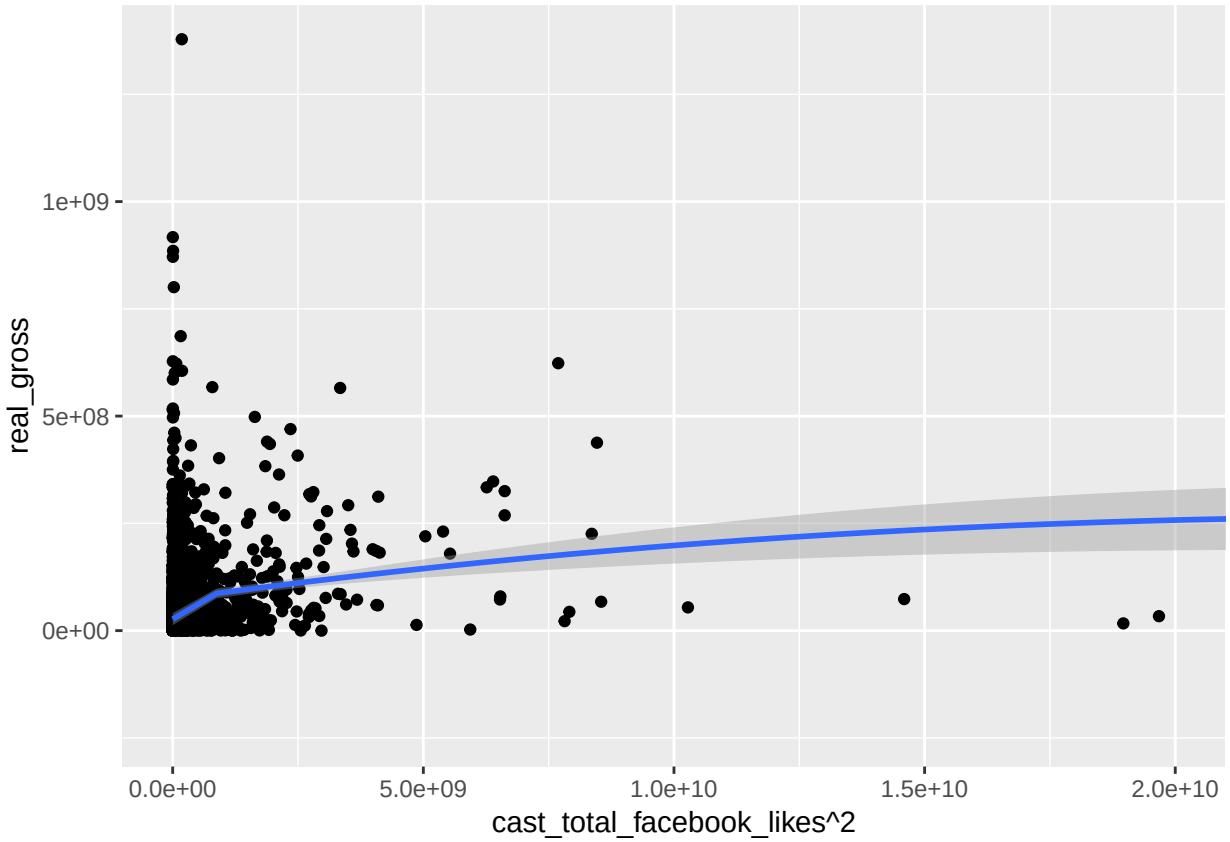
```
## Warning: Removed 340 rows containing non-finite values (stat_smooth).
```



Try squaring as well. . . still not linear

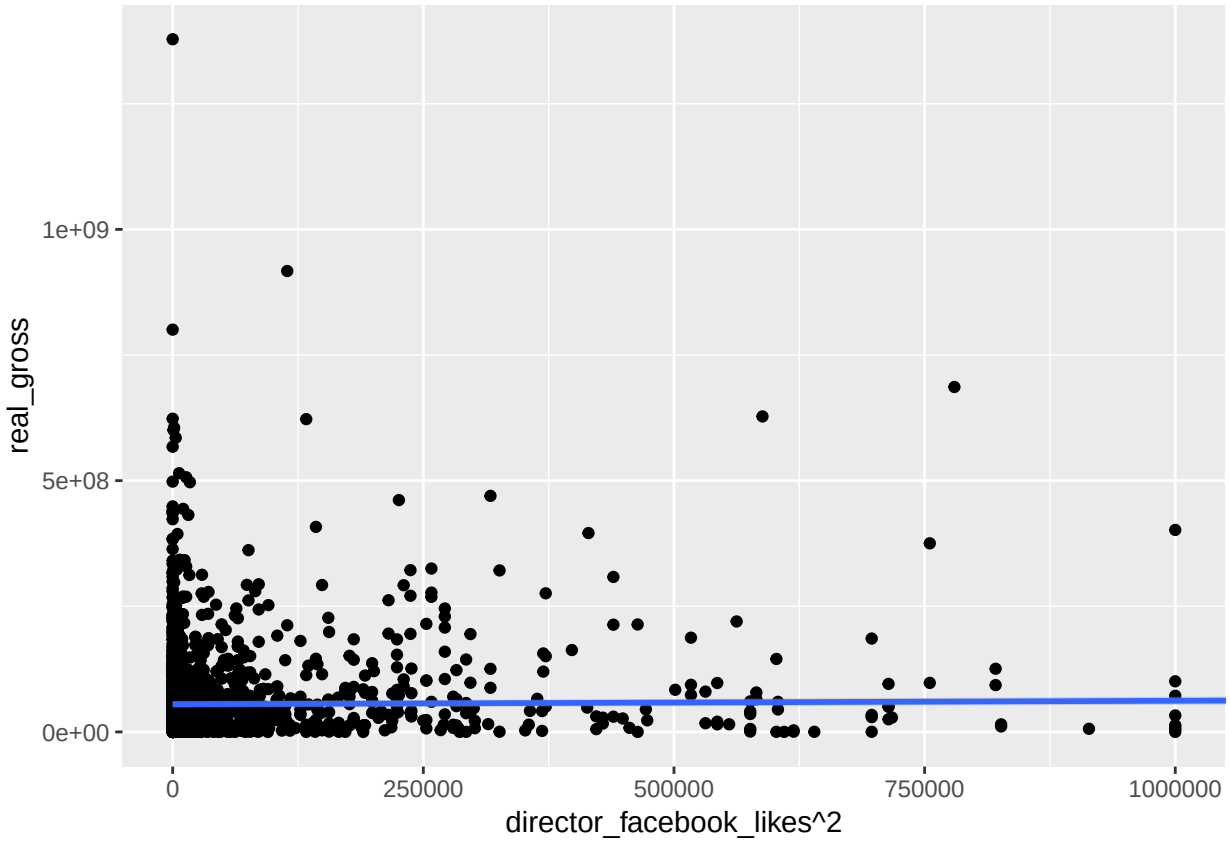
```
train %>% ggplot(aes(cast_total_facebook_likes^2, y = real_gross)) +
  geom_point() +
  geom_smooth() +
  coord_cartesian(xlim = c(0, 20000000000))
```

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




```
train %>% ggplot(aes(director_facebook_likes^2, y = real_gross)) +
  geom_point() +
  geom_smooth() +
  coord_cartesian(xlim = c(0, 1000000))
```

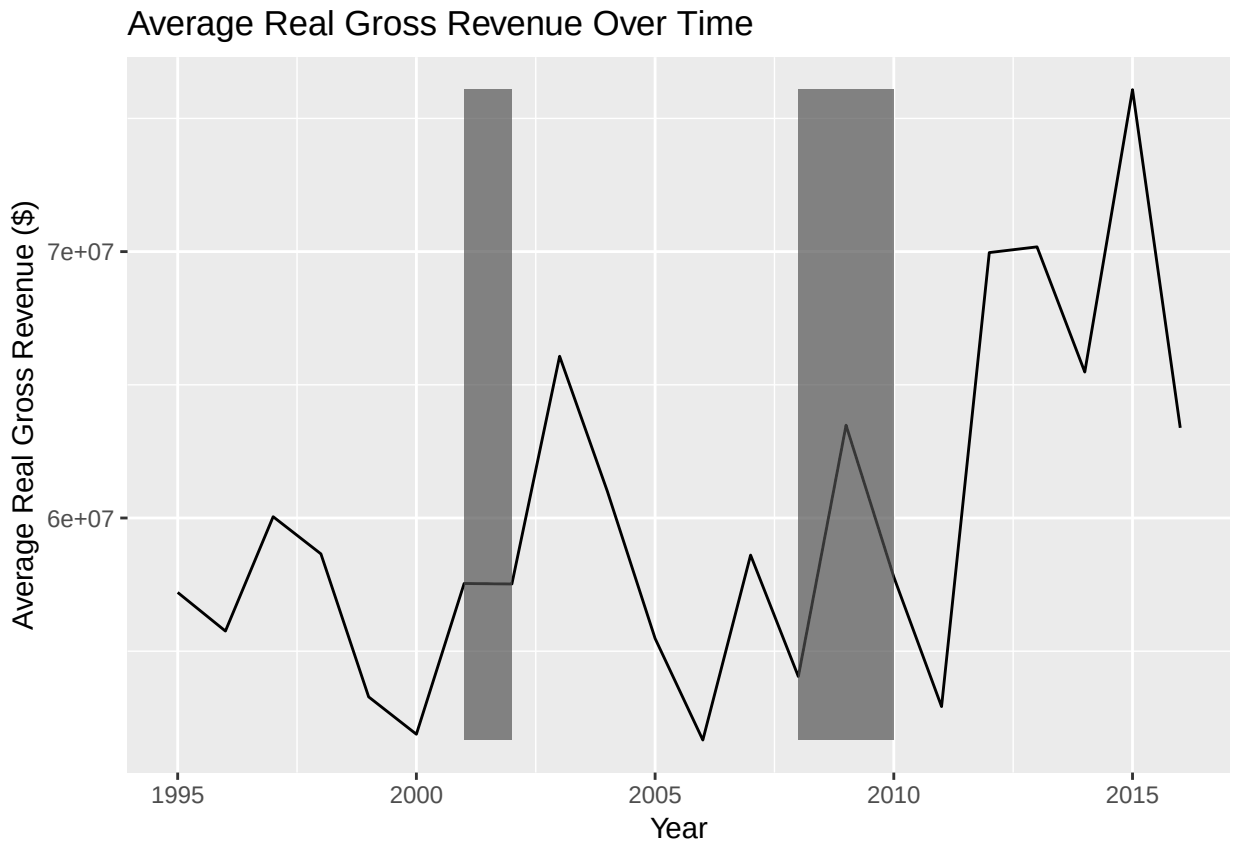
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



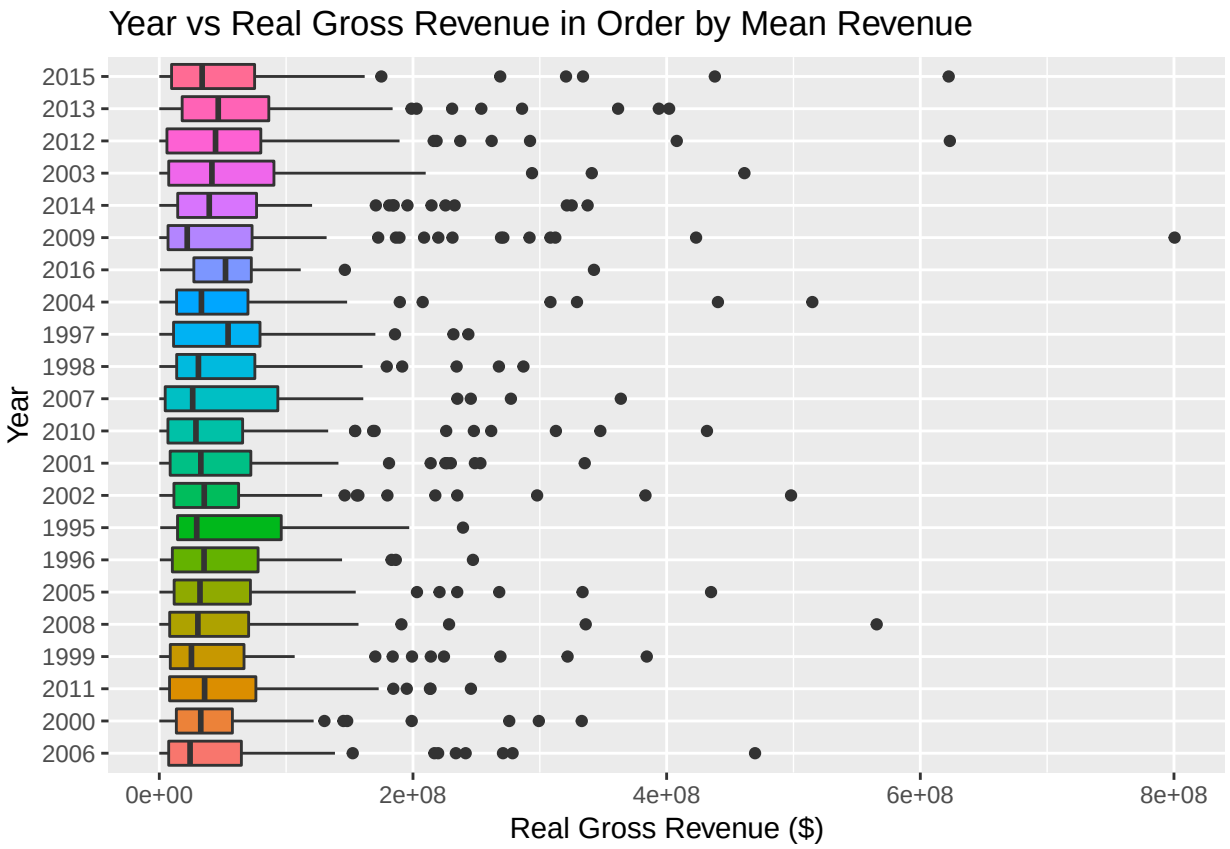
Year

Average real revenue vs year
Adeed APPROXIMATE recession shading. Annual data, so hard to do.
Real revenue increase during recessions, but then decreases as recession worsens? (have seen this before with Great Depression - numerous articles we can reference)
Some differences between years based on boxplots.

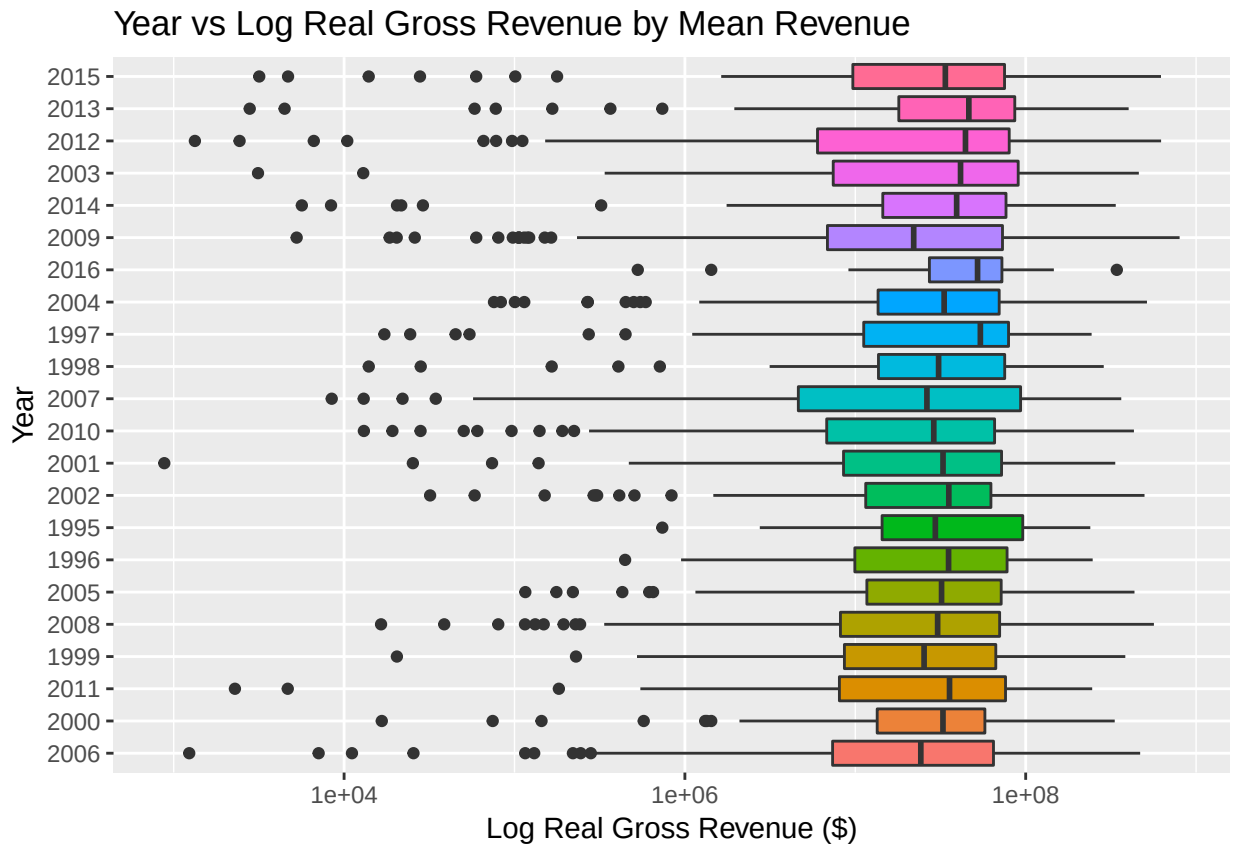
```
# take average of revenue per year: how does revenue evolve over time?
train_sum <- train %>%
  group_by(year) %>%
  summarise(real_gross_avg = mean(real_gross))

# need to limit because before 1995 there are very few observations per year (< 10 usually).
# this causes large spikes because one high earning or low earning movie influences the average heavily
# Starting at 1995, where have at least 30 (or very close) movies per year. Now can see movements over

ggplot(data = train_sum %>% filter(as.integer(as.character(year)) >= 1995)) +
  geom_line(aes(x = as.integer(as.character(year)), y = real_gross_avg)) +
  labs(title = 'Average Real Gross Revenue Over Time', x = 'Year', y = 'Average Real Gross Revenue ($)'
    geom_rect(aes(xmin = 2008, xmax = 2010,
                  ymin = min(real_gross_avg, na.rm = T),
                  ymax = max(real_gross_avg, na.rm = T)), alpha = .05) +
  geom_rect(aes(xmin = 2001, xmax = 2002,
                  ymin = min(real_gross_avg, na.rm = T),
                  ymax = max(real_gross_avg, na.rm = T)), alpha = .05)
```



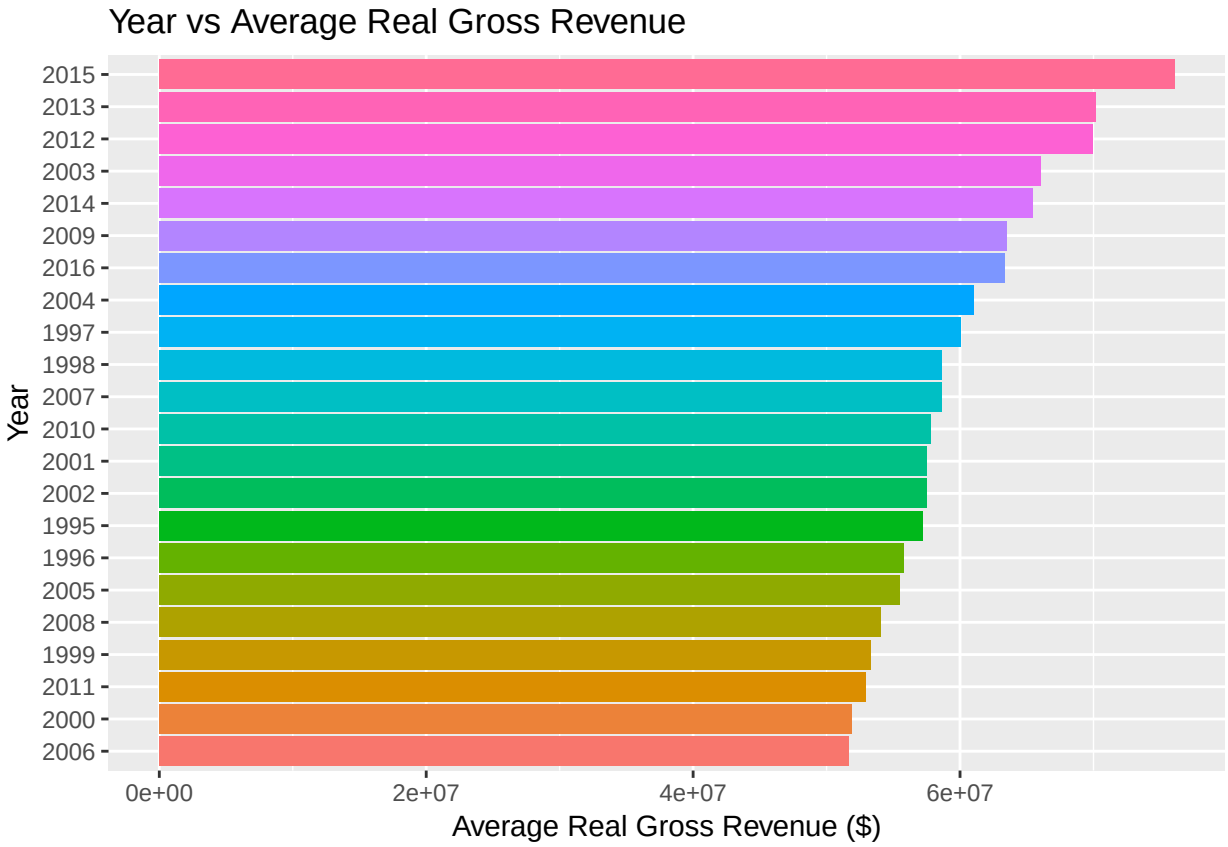

```
# now treat year as a factor and create box plots and average bar plots using log transformations
train_year <- train %>%
  filter(as.integer(as.character(year)) >= 1995) %>%
  group_by(year) %>%
  mutate(real_gross_avg = mean(real_gross)) %>%
  ungroup(year) %>%
  mutate(year = reorder(year, real_gross_avg))

# box plot
plt_base <- train_year %>%
  ggplot() +
  geom_boxplot(aes(x = year, y = real_gross, fill = year)) +
  theme(legend.position = 'none') +
  coord_flip()
plt_base +
  labs(x = 'Year', y = 'Real Gross Revenue ($)',
       title = 'Year vs Real Gross Revenue in Order by Mean Revenue')
```



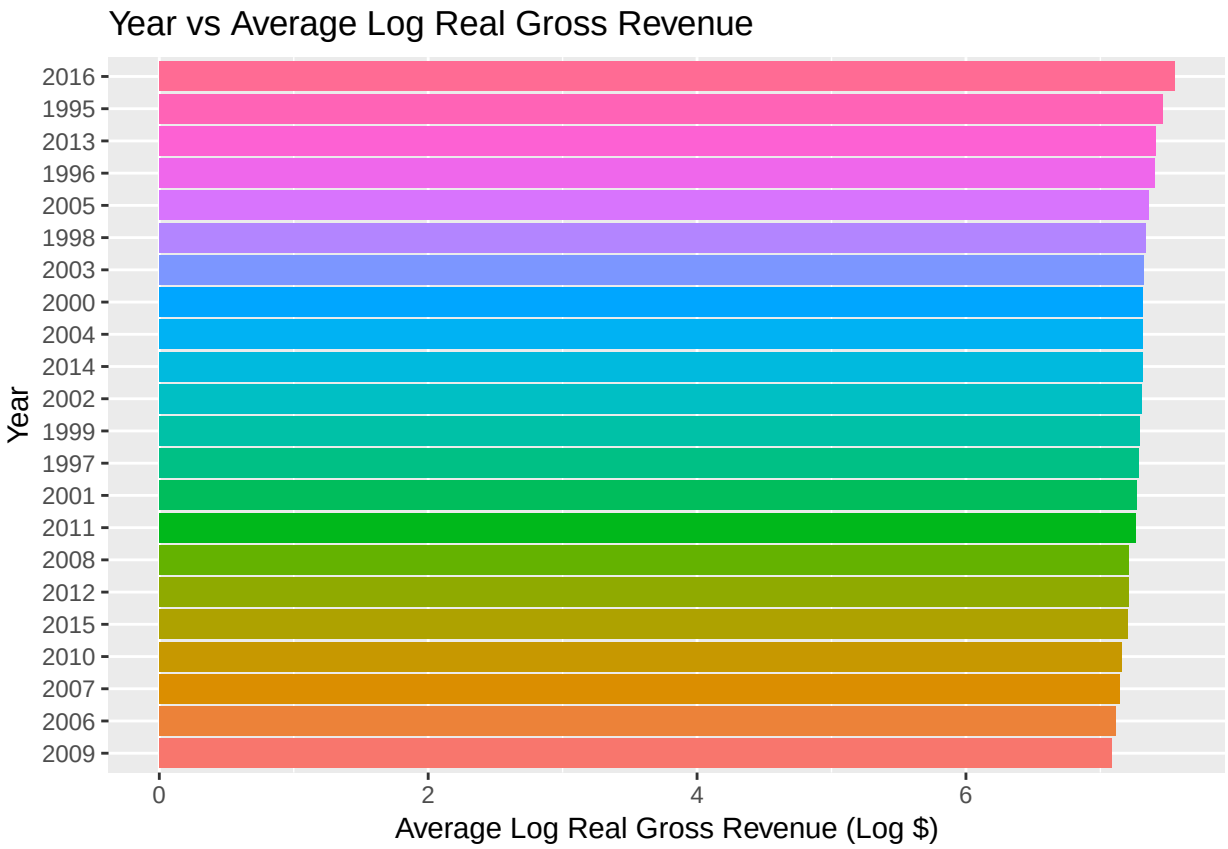

```
plt_base +
   labs(x = 'Year', y = 'Log Real Gross Revenue ($)',
        title = 'Year vs Log Real Gross Revenue by Mean Revenue') +
  scale_y_log10()
```




```r
# bar plot
train_year %>%
  group_by(year) %>%
  summarise(avg_real_gross = mean(real_gross)) %>%
  ggplot() +
  geom_col(aes(x = year, y = avg_real_gross, fill = year)) +
  labs(x = 'Year', y = 'Average Real Gross Revenue ($)',
  title = 'Year vs Average Real Gross Revenue') +
  theme(legend.position = 'none') +
  coord_flip()
```



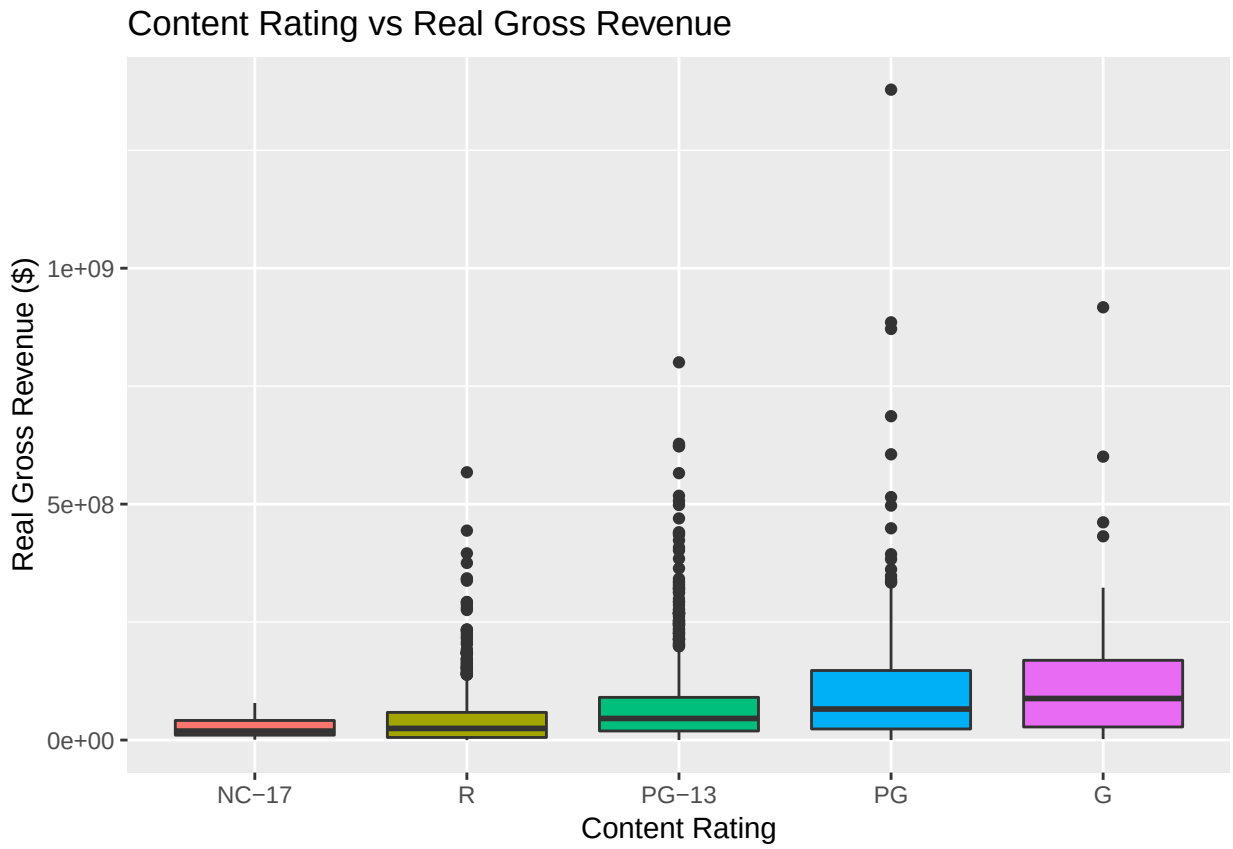

```r
# log
# would like to use scale_y_log10() but can't since mean
train_year %>%
  mutate(year = reorder(year, real_gross_log)) %>%
  group_by(year) %>%
  summarise(avg_real_gross_log = mean(real_gross_log)) %>%
  ggplot() +
  geom_col(aes(x = year, y = avg_real_gross_log, fill = year)) +
  labs(x = 'Year', y = 'Average Log Real Gross Revenue (Log $)',
  title = 'Year vs Average Log Real Gross Revenue') +
  theme(legend.position = 'none') +
  coord_flip()
```



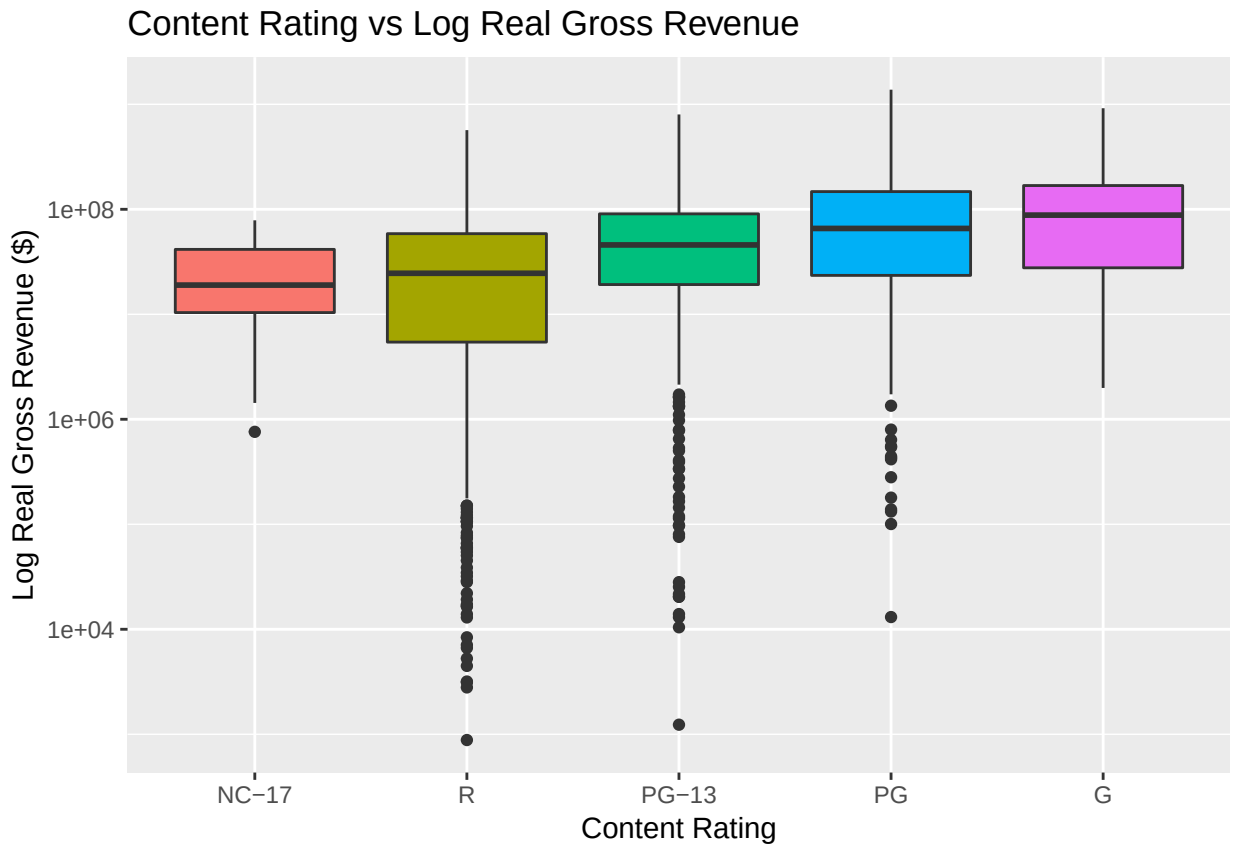
Content Rating

Bar graph of average real revenue and boxplot
Linear relationship. Good candidate to include in the model Each individual rating has an significantly different average mean real revenue

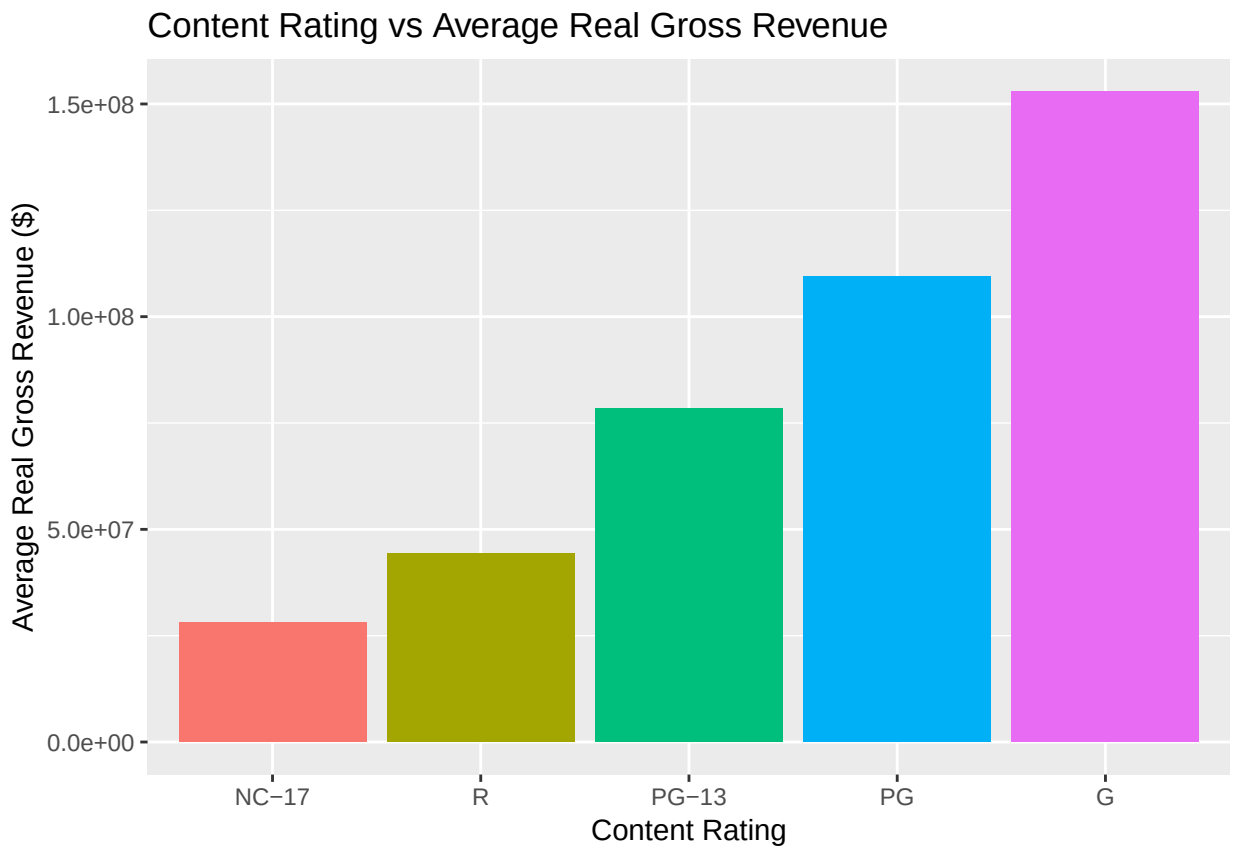
```
# data manipulation. Factor.
train_content <- train %>%
  # filter out missing
  filter(!is.na(content_rating)) %>%
  # reorder content rating based on gross
  mutate(content_rating = reorder(content_rating, real_gross))

# boxplot
plt_base <- train_content  %>%
  ggplot() +
  geom_boxplot(aes(x = content_rating, y = real_gross, fill = content_rating)) +
  theme(legend.position = 'none')
plt_base +
  labs(x = 'Content Rating', y = 'Real Gross Revenue ($)',
       title = 'Content Rating vs Real Gross Revenue')
```



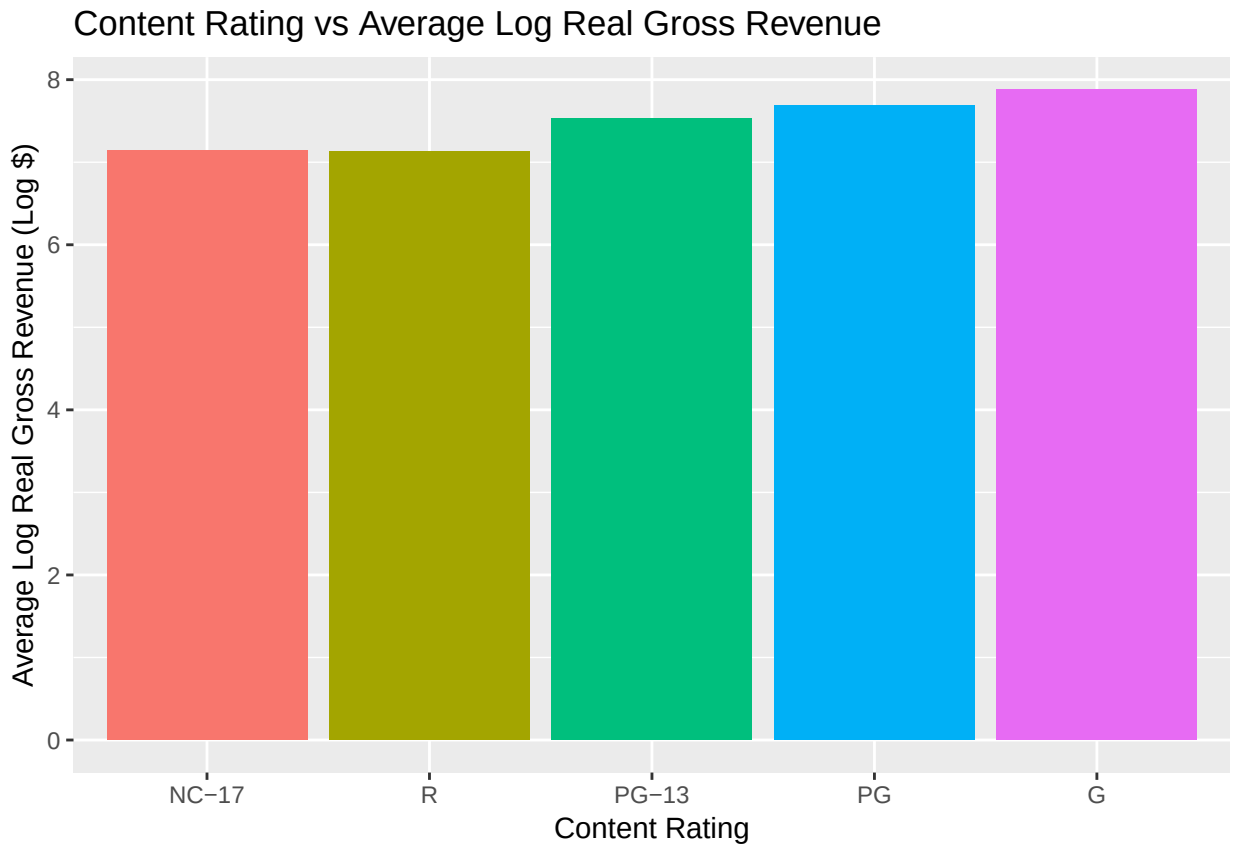

```
plt_base +
  labs(x = 'Content Rating', y = 'Log Real Gross Revenue ($)',
       title = 'Content Rating vs Log Real Gross Revenue') +
  scale_y_log10()
```




```
# bar graph
train_content %>%
  # average revenue by content rating
  group_by(content_rating) %>%
  summarise(avg_real_gross = mean(real_gross)) %>%
  ggplot() +
  geom_col(aes(x = content_rating, y = avg_real_gross, fill = content_rating)) +
  labs(x = 'Content Rating', y = 'Average Real Gross Revenue ($)',
  title = 'Content Rating vs Average Real Gross Revenue') +
  theme(legend.position = 'none')
```

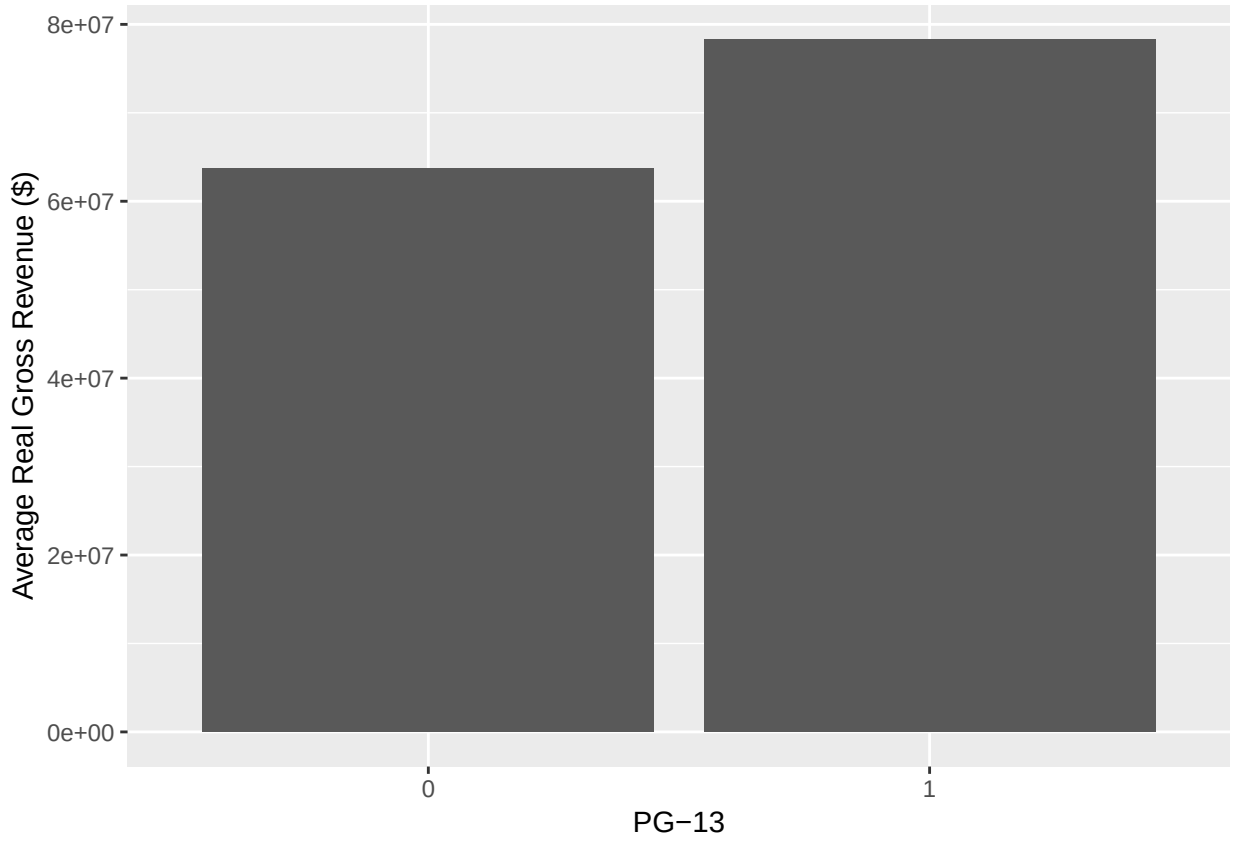



```
# log
# would like to use scale_y_log10() but can't since mean
train_content %>%
  # average revenue by content rating
  group_by(content_rating) %>%
  summarise(avg_real_gross_log = mean(real_gross_log)) %>%
  ggplot() +
  geom_col(aes(x = content_rating, y = avg_real_gross_log, fill = content_rating)) +
  labs(x = 'Content Rating', y = 'Average Log Real Gross Revenue (Log $)',
  title = 'Content Rating vs Average Log Real Gross Revenue') +
  theme(legend.position = 'none')
```

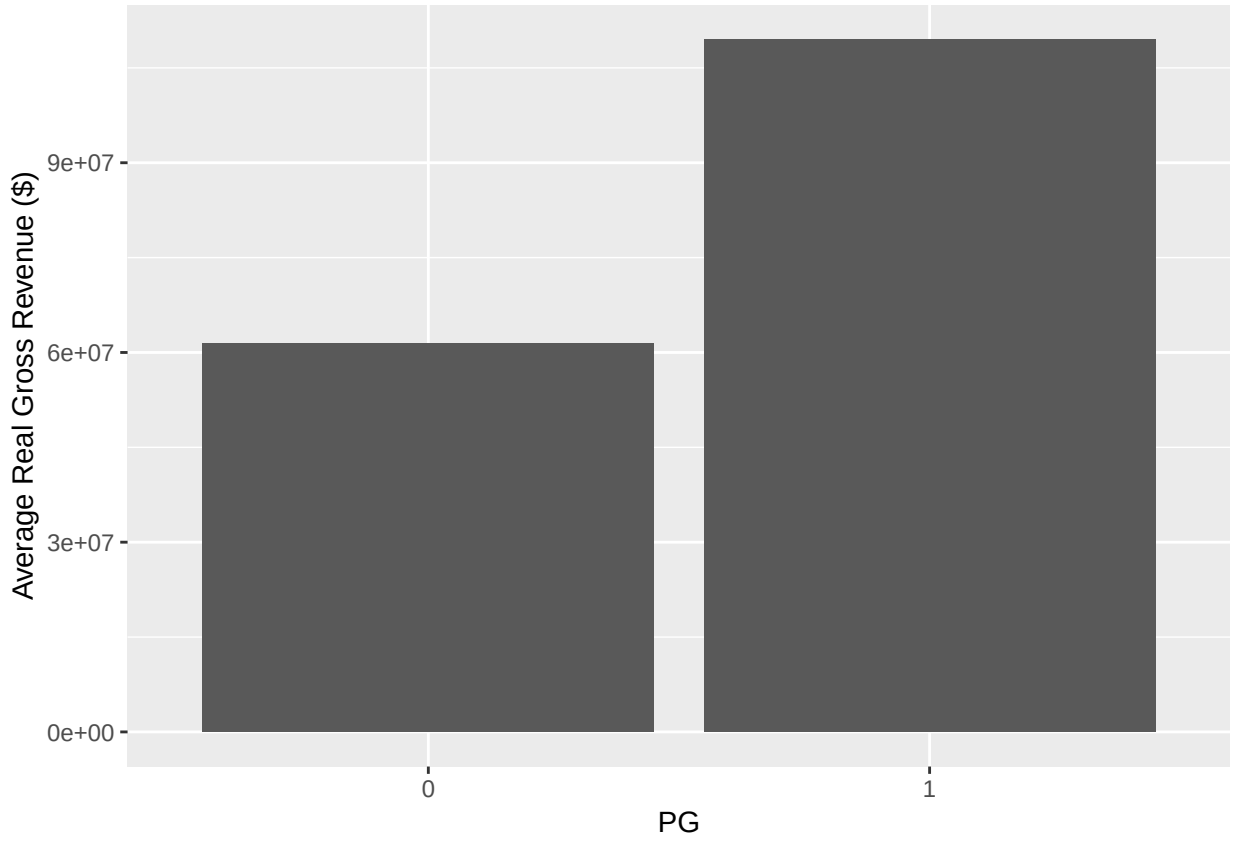



```
# graph each content rating 0/1 against mean revenue: is there a difference?
lapply(unique(train_content$content_rating), function(r) {
  train_content %>%
    # get 1 if this rating, 0 else. Make factor
    mutate(rating_dum = as.factor(ifelse(content_rating == r, 1, 0))) %>%
    # mean revenue for 0 vs 1 for that content rating
    group_by(rating_dum) %>%
    summarise(avg_real_gross = mean(real_gross)) %>%
    ggplot() +
    geom_col(aes(x = rating_dum, y = avg_real_gross)) +
    labs(x = r, y = 'Average Real Gross Revenue ($)')
})
```

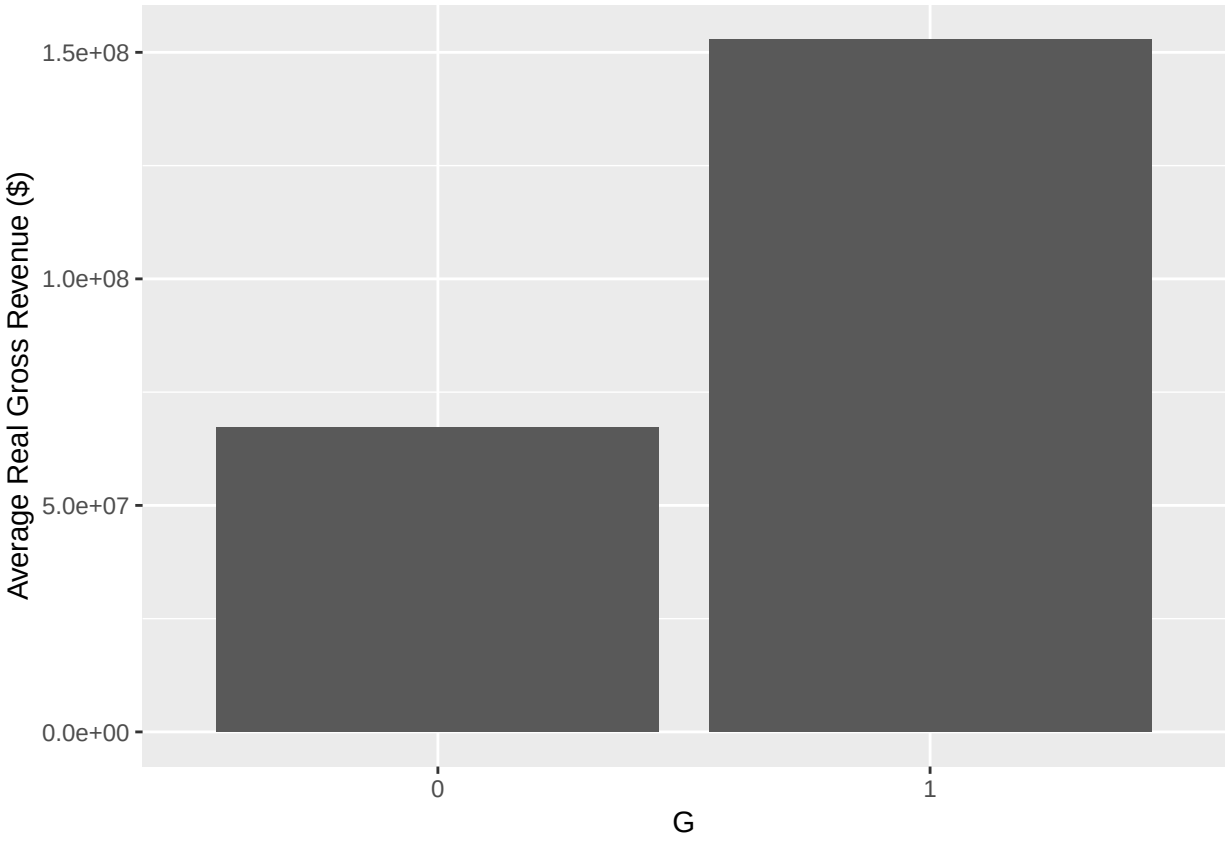
```
## [[1]]
```



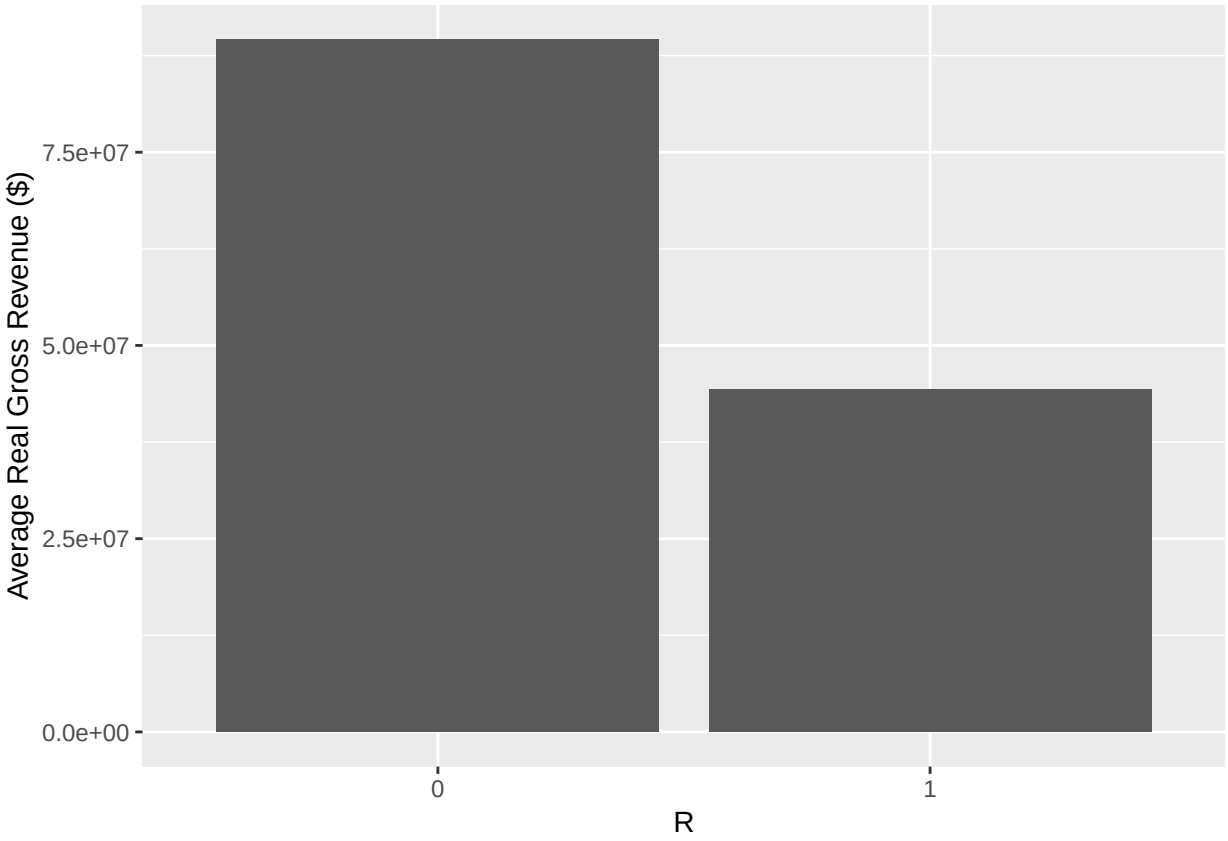

```
##
## [[2]]
```



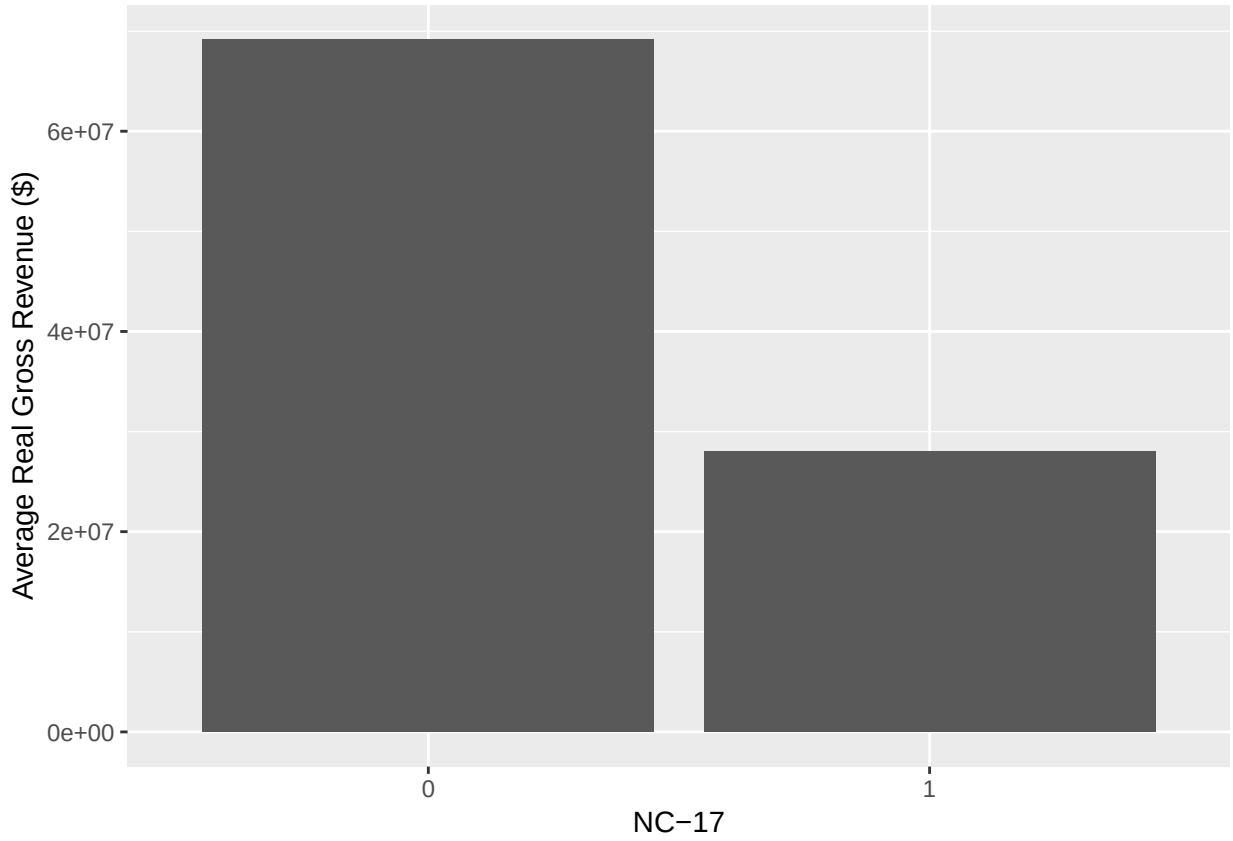

```
##
## [[3]]
```




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

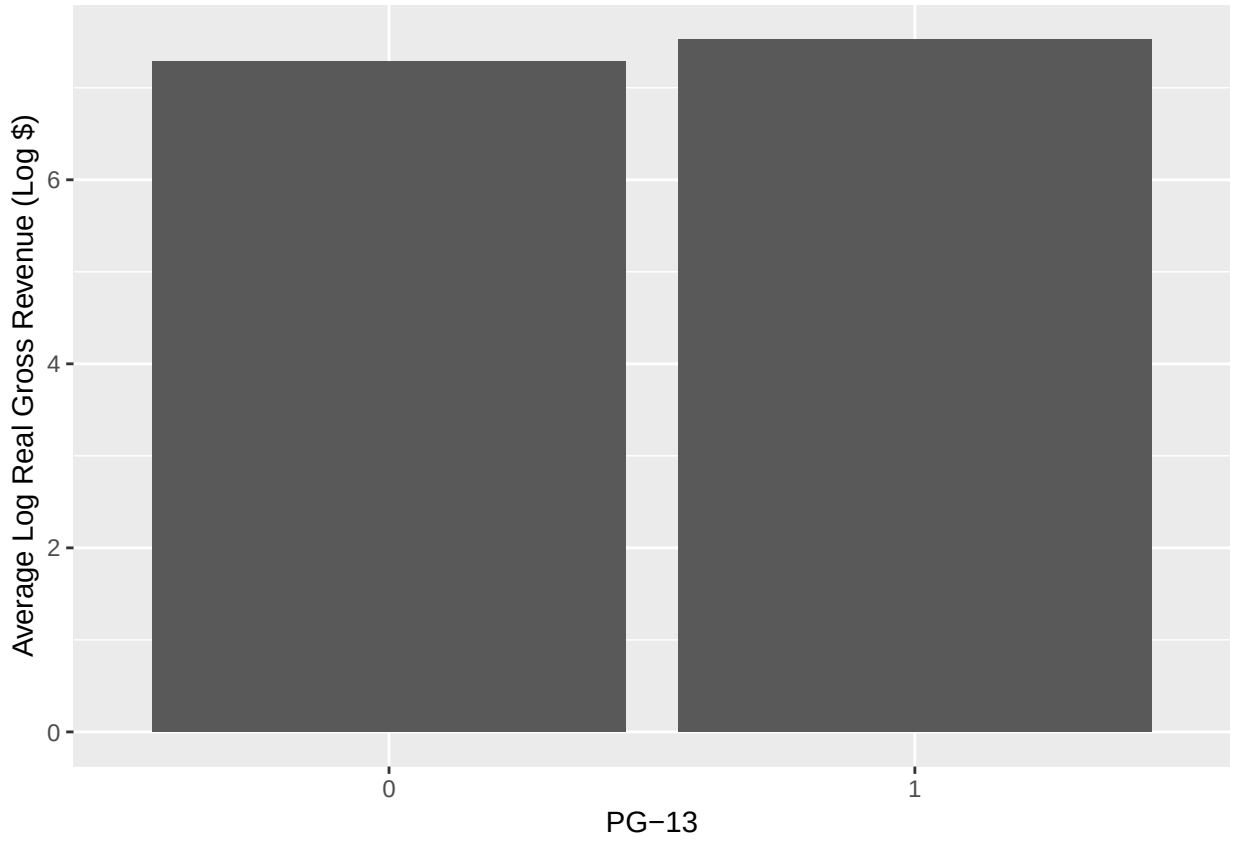



```
##
## [[5]]
```

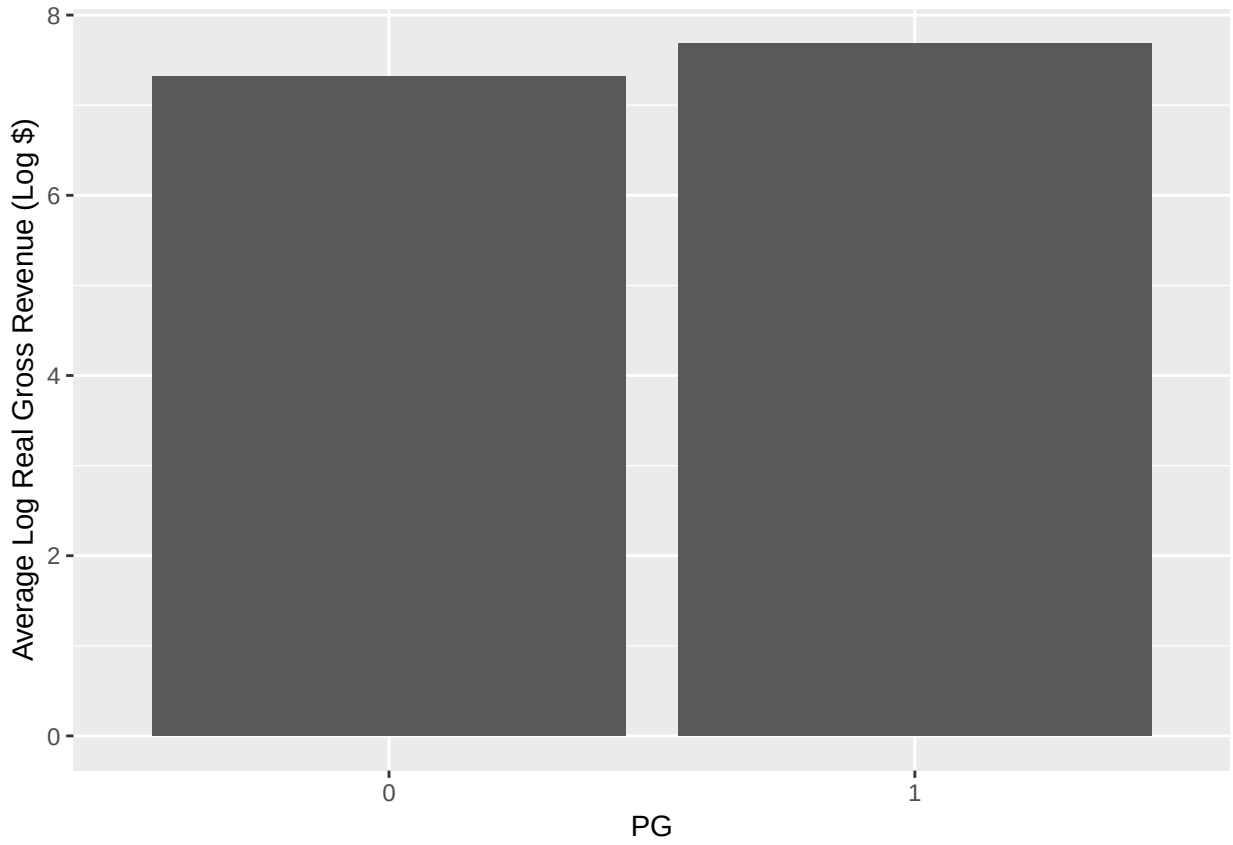



```
lapply(unique(train_content$content_rating), function(r) {
  train_content %>%
    # get 1 if this rating, 0 else. Make factor
    mutate(rating_dum = as.factor(ifelse(content_rating == r, 1, 0))) %>%
    # mean revenue for 0 vs 1 for that content rating
    group_by(rating_dum) %>%
    summarise(avg_real_gross_log = mean(real_gross_log)) %>%
    ggplot() +
    geom_col(aes(x = rating_dum, y = avg_real_gross_log)) +
    labs(x = r, y = 'Average Log Real Gross Revenue (Log $)')
})
```

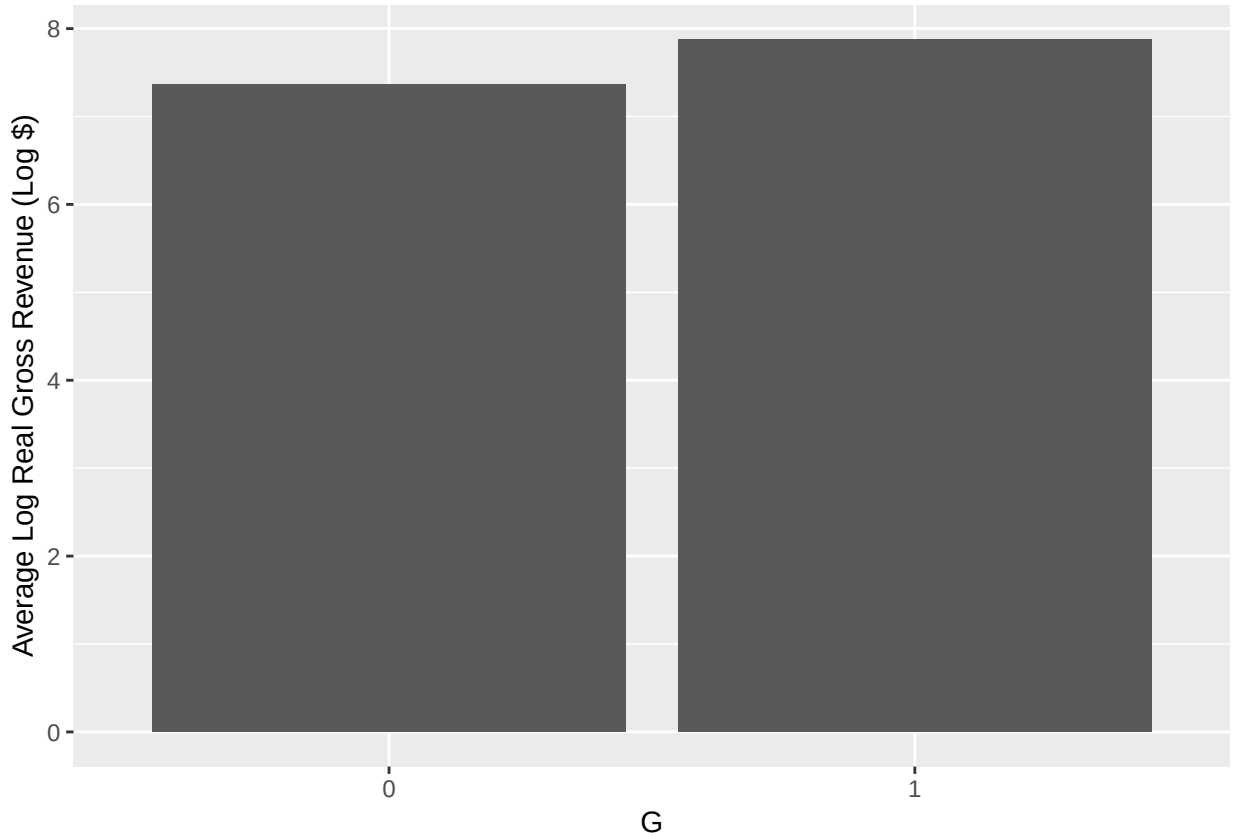
```
## [[1]]
```



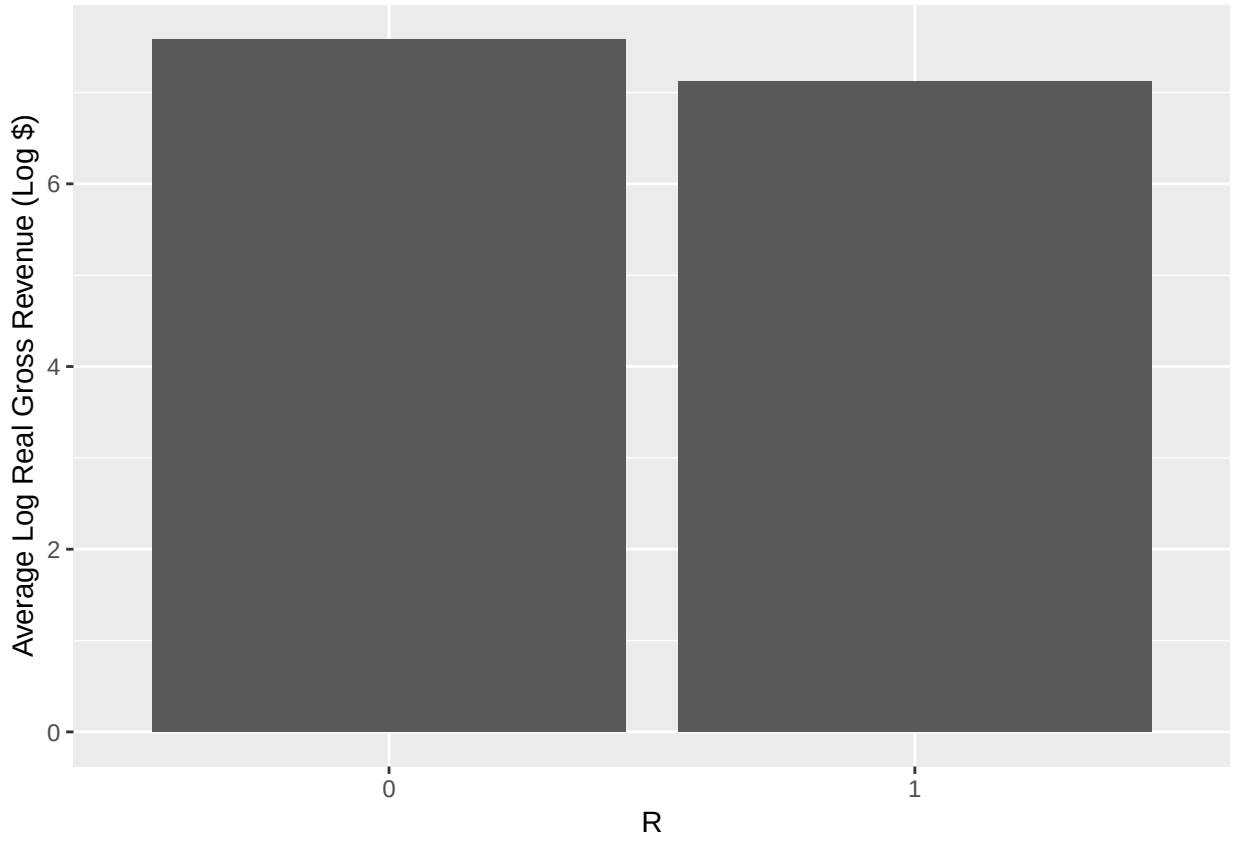

```
##
## [[2]]
```



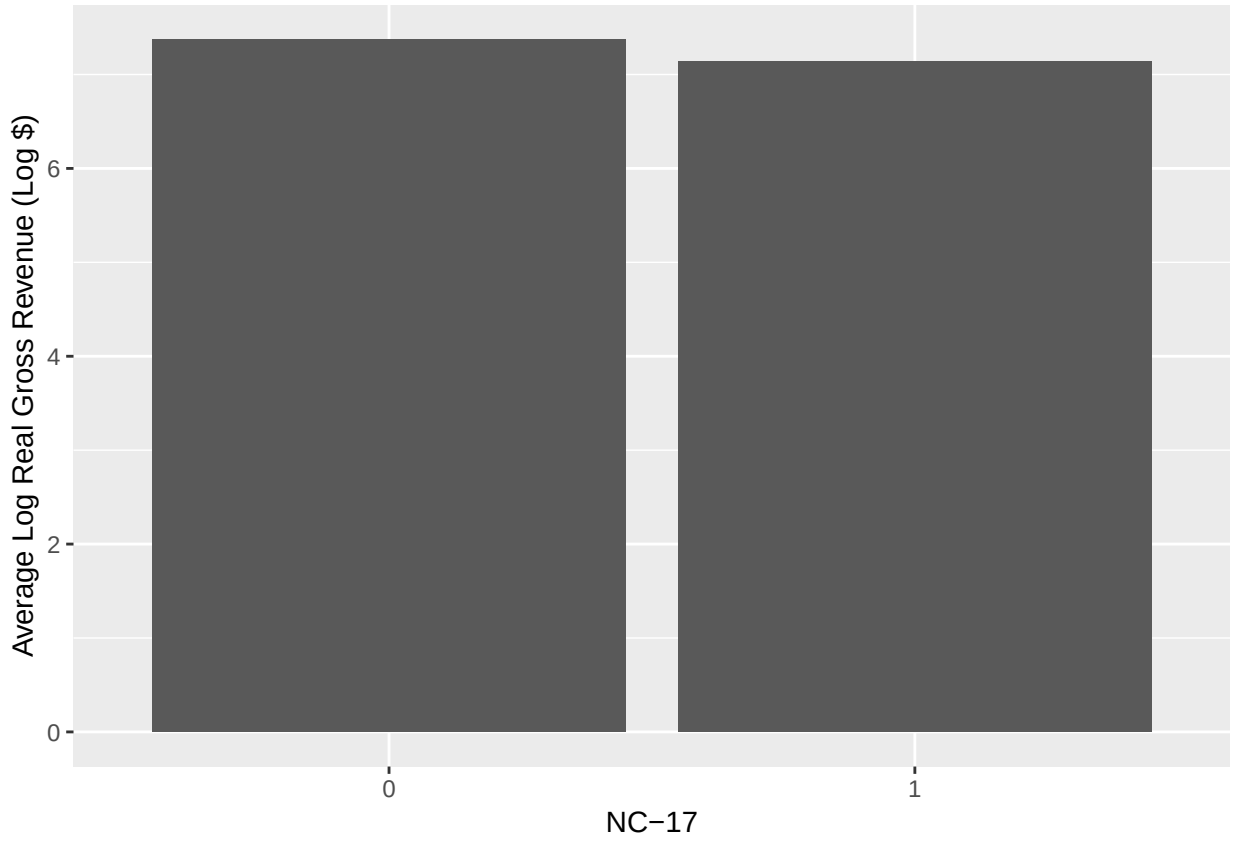

```
##
## [[3]]
```




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




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

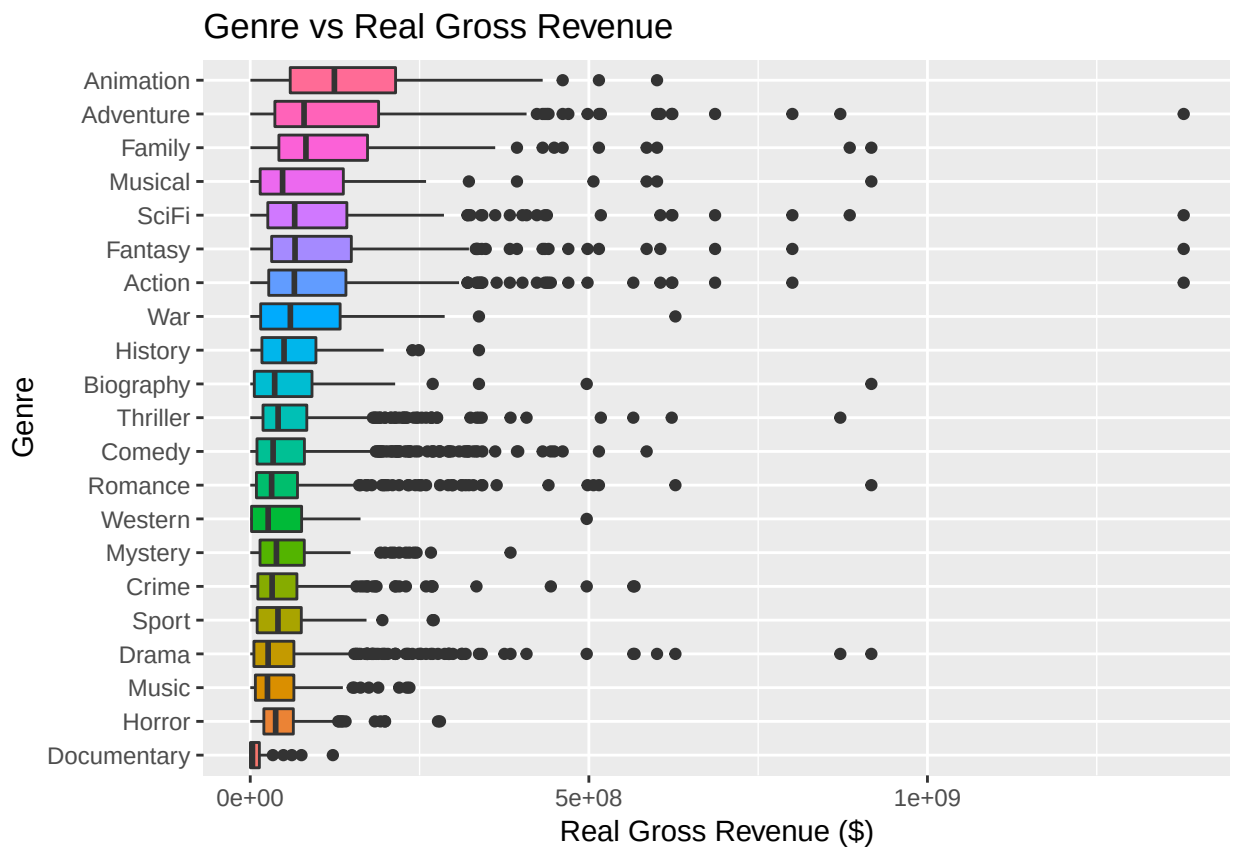


Genre

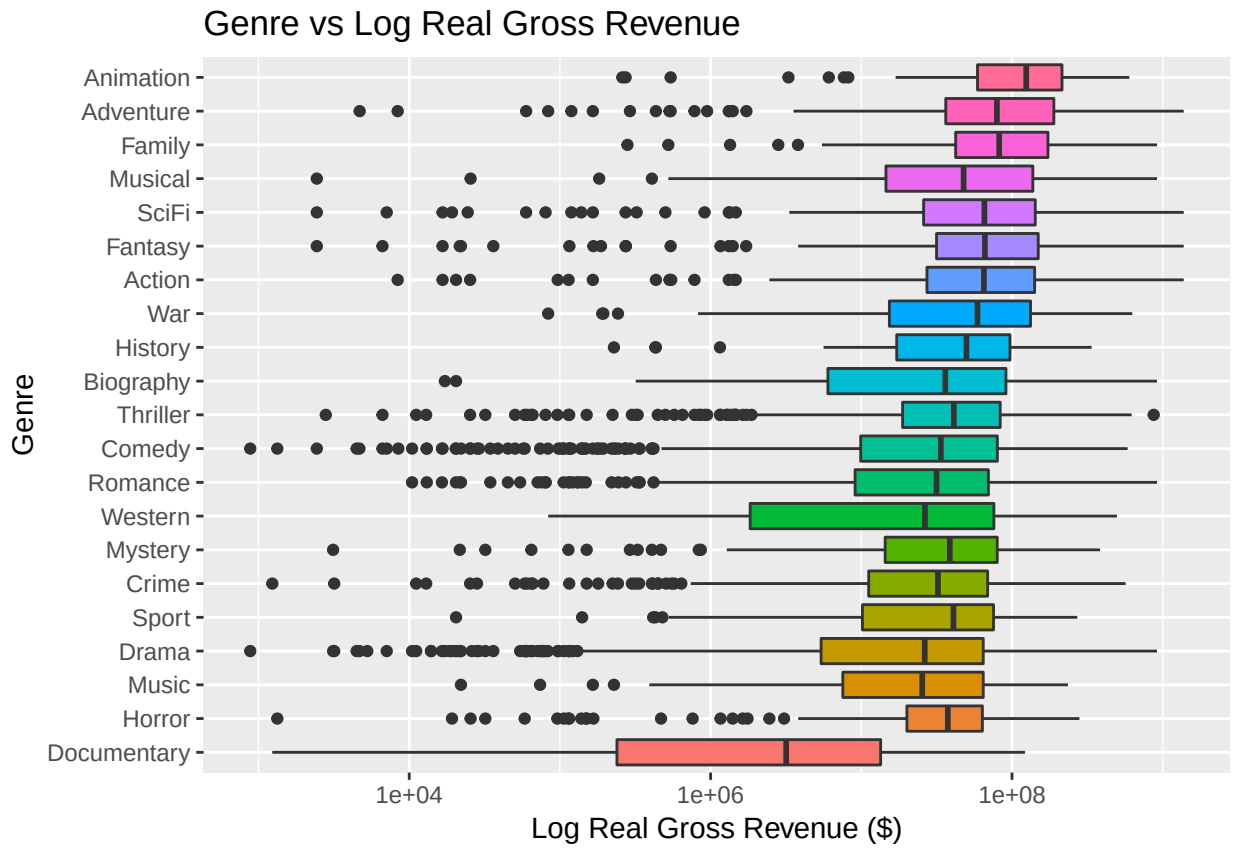
Bar graph of genre vs real revenue. Try boxplot and bar graph against average real revenue.
Fairly clear differences in relationship with revenue by genre. Good candidate to include in model.

```
# untidy the genre data such that one obseration is spread across many rows. Easier to graph
genre_cols <- c('Action', 'Adventure', 'Animation', 'Biography', 'Comedy', 'Crime', 'Documentary',
                'Drama', 'Family', 'Fantasy', 'History', 'Horror', 'Music', 'Musical', 'Mystery',
                'Romance', 'SciFi', 'Sport', 'Thriller', 'War', 'Western')
train_genre <- train %>%
  # gather: one row per genre-movie combo
  gather(genre_cols, key = genre, value = yes) %>%
  # only keep when 'yes' is 1 (yes it is of that genre) %>%
  filter(yes == 1) %>%
  # reorder genre by real_gross
  mutate(genre = reorder(genre, real_gross))

# boxplot
plt_base <- train_genre %>%
  ggplot() +
  geom_boxplot(aes(x = genre, y = real_gross, fill = genre)) +
  coord_flip() +
  theme(legend.position = 'none')

plt_base +
  labs(x = 'Genre', y = 'Real Gross Revenue ($)',
```

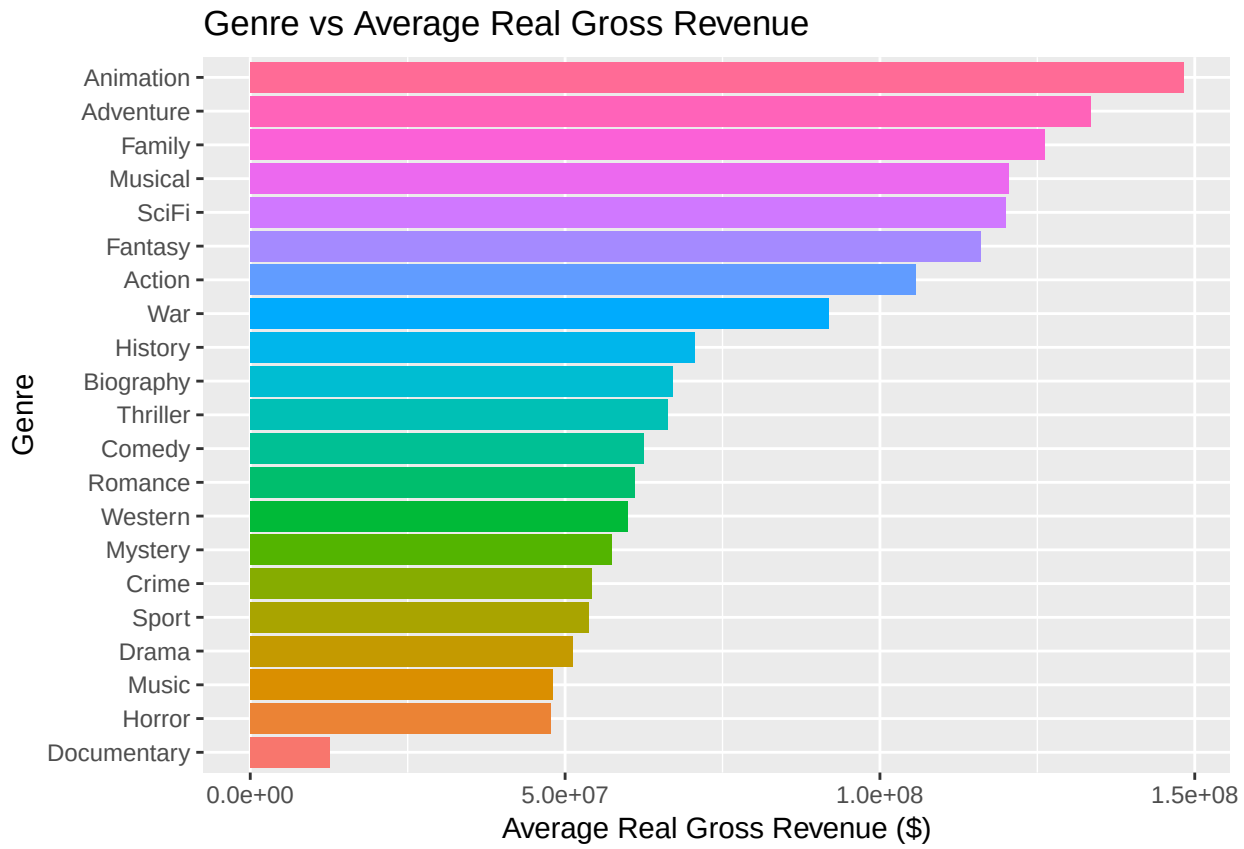
```
       title = 'Genre vs Real Gross Revenue')
```



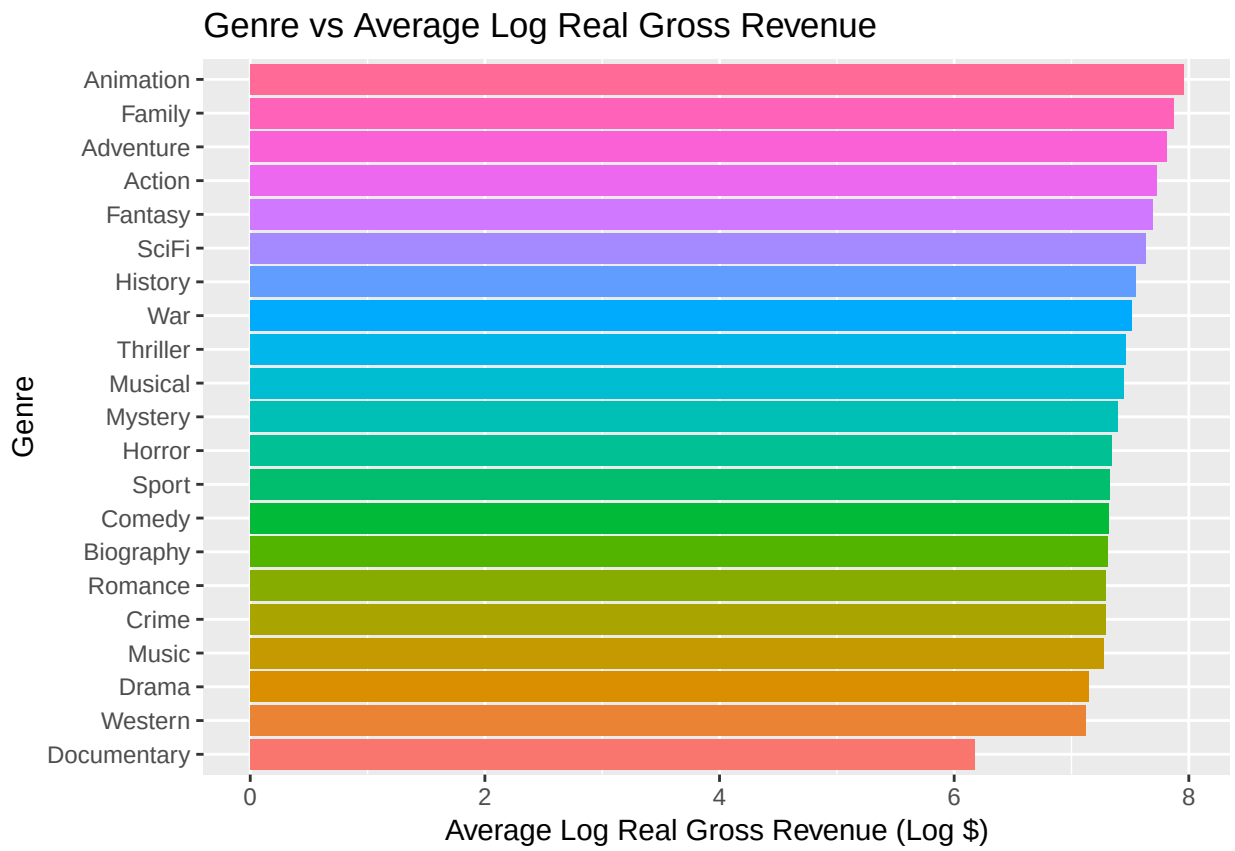

```
plt_base +
  labs(x = 'Genre', y = 'Log Real Gross Revenue ($)',
       title = 'Genre vs Log Real Gross Revenue') +
  scale_y_log10()
```




```
# bar graph
train_genre %>%
  # average by genre
  group_by(genre) %>%
  summarise(avg_real_gross = mean(real_gross)) %>%
  # graph
  ggplot() +
  geom_col(aes(x = genre, y = avg_real_gross, fill = genre)) +
  coord_flip() +
  labs(x = 'Genre', y = 'Average Real Gross Revenue ($)',
       title = 'Genre vs Average Real Gross Revenue') +
  theme(legend.position = 'none')
```

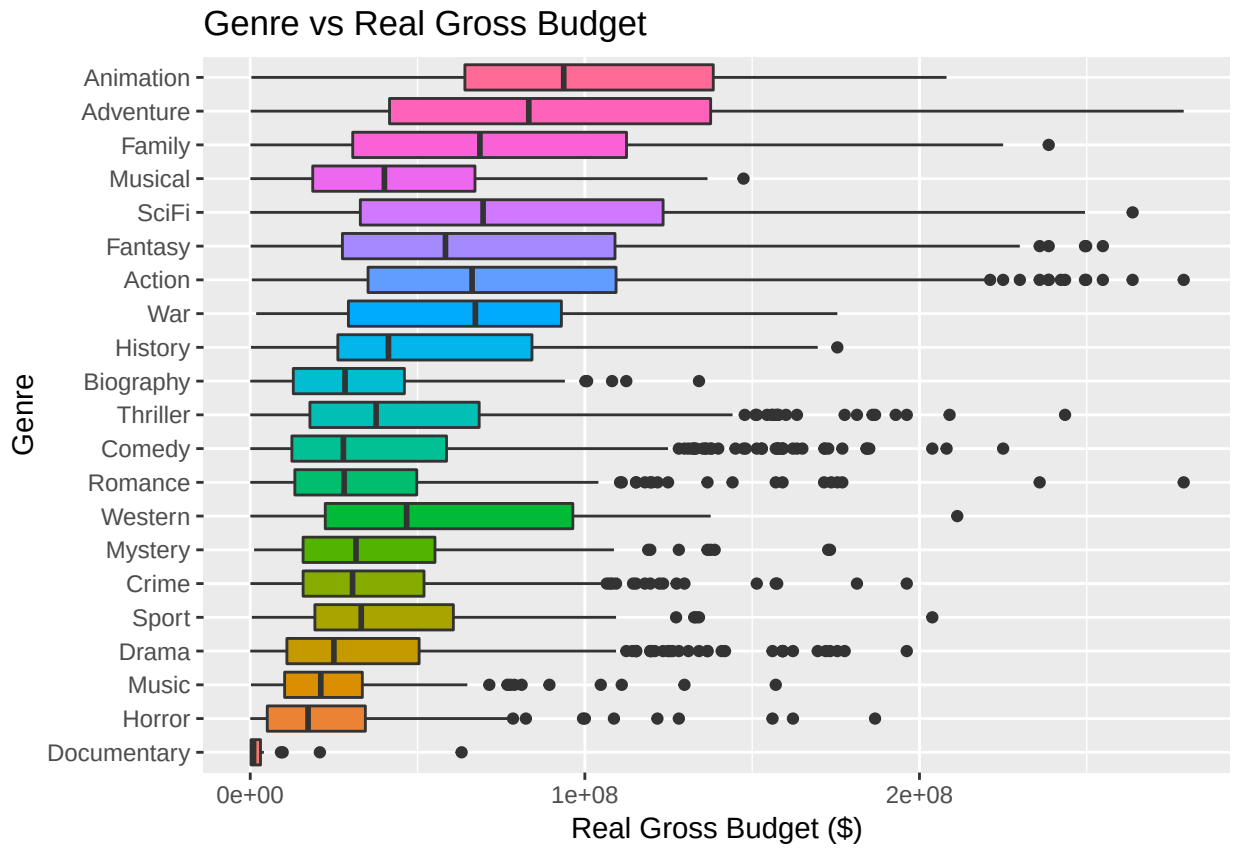



```
train_genre %>%
  # reorder factor by log
  mutate(genre = reorder(genre, real_gross_log)) %>%
  # average by genre
  group_by(genre) %>%
  summarise(avg_real_gross_log = mean(real_gross_log)) %>%
  # graph
  ggplot() +
  geom_col(aes(x = genre, y = avg_real_gross_log, fill = genre)) +
  coord_flip() +
  labs(x = 'Genre', y = 'Average Log Real Gross Revenue (Log $)',
       title = 'Genre vs Average Log Real Gross Revenue') +
  theme(legend.position = 'none')
```

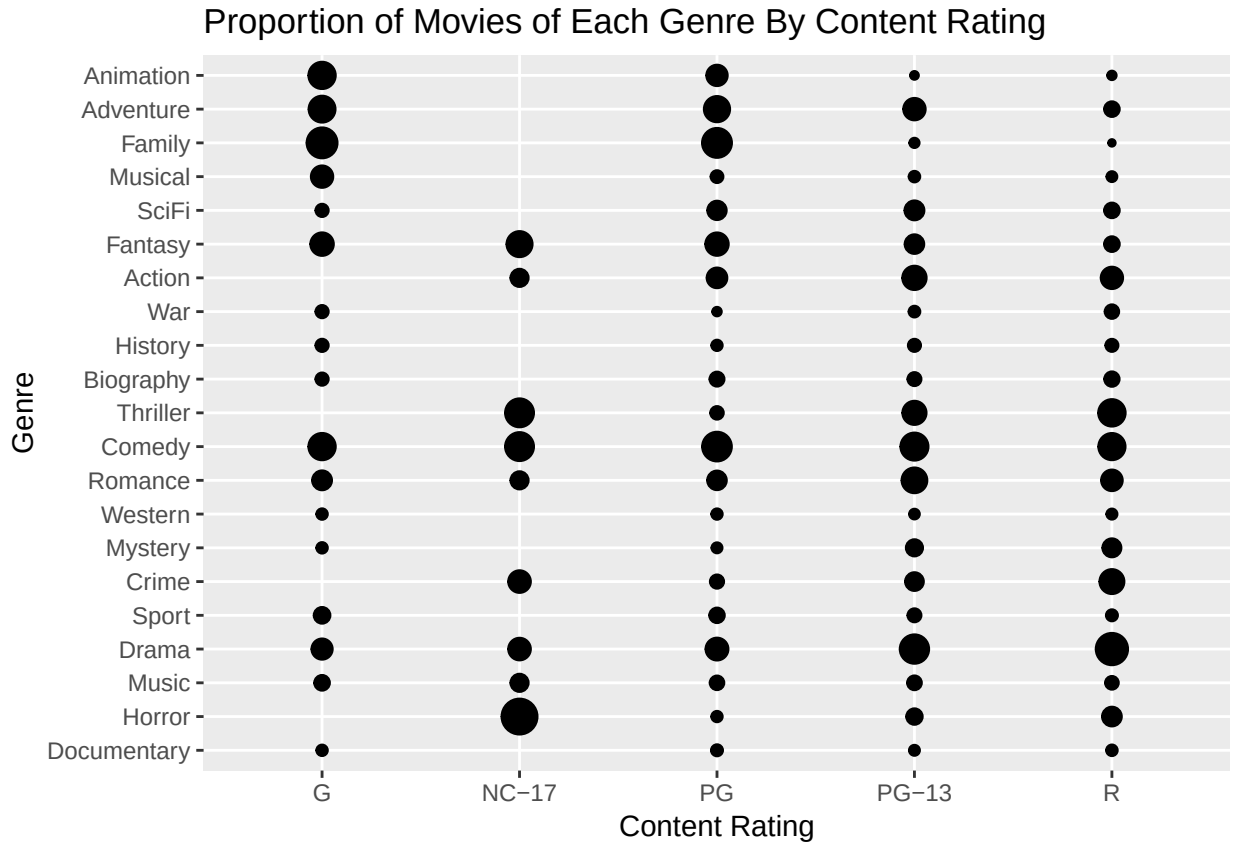



```
#### genre against budget:
# using this as a partial justification for why genre isn't in the final model
train_genre %>%
  mutate(genre = reorder(genre, real_budget)) %>%
  ggplot() +
  geom_boxplot(aes(x = genre, y = real_budget, fill = genre)) +
  coord_flip() +
  theme(legend.position = 'none') +
  labs(x = 'Genre', y = 'Real Gross Budget ($)',
       title = 'Genre vs Real Gross Budget')
```

```
## Warning: Removed 239 rows containing non-finite values (stat_boxplot).
```




```
#### genre against content rating:
# using this as a partial justification for why genre isn't in the final model
# proportion of movies within each content rating that are each genre
train_genre_content <- train_genre %>%
  group_by(genre, content_rating) %>%
  count() %>%
  group_by(content_rating) %>%
  mutate(prop = n / sum(n))
train_genre_content %>%
  filter(!is.na(content_rating)) %>%
  ggplot() +
  geom_count(aes(x = genre, y = content_rating, size = prop)) +
  coord_flip() +
  theme(legend.position = 'none') +
  labs(x = 'Genre', y = 'Content Rating', title = 'Proportion of Movies of Each Genre By Content Rating
```


Proportion of Movies of Each Genre By Content Rating
Genre
Animation
Adventure
Family
Musical
SciFi
Fantasy
Action
War
History
Biography
Thriller
Comedy
Romance
Western
Mystery
Crime
Sport
Drama
Music
Horror
Documentary
G
NC−17
PG
PG−13
R
Content Rating