# My Anime List Analysis

Aaron Yu

2025-04-14

#Loading Necessary Libraries

```
library(ggplot2)
library(MASS)
library(faraway)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.4     v tibble    3.2.1
## v purrr     1.0.4     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
##
## Attaching package: 'GGally'
##
## The following object is masked from 'package:faraway':
##
##     happy
```

```
library(dplyr)
```

#Loading in Data and Basic Structure

```
animelist = read.csv("AnimeList.csv")
animelist = animelist %>%
  select(-title_japanese) #For rendering purposes
head(animelist)
```

```
##   anime_id               title             title_english
## 1    11013       Inu x Boku SS Inu X Boku Secret Service
## 2     2104    Seto no Hanayome     My Bride is a Mermaid
## 3     5262  Shugo Chara!! Doki        Shugo Chara!! Doki
## 4      721       Princess Tutu             Princess Tutu
## 5    12365 Bakuman. 3rd Season                  Bakuman.
## 6     6586 Yume-iro Pâtissière
##                                                                  title_synonyms
## 1                                                                  Youko x Boku SS
## 2                                                               The Inland Sea Bride
## 3                                  Shugo Chara Ninenme, Shugo Chara! Second Year
## 4
## 5                                                                Bakuman Season 3
## 6 Yumeiro Patissiere, YumePati, Dream-Colored Pastry Chef, Yumeiro Pâtissière
##                                                 image_url type   source
## 1 https://myanimelist.cdn-dena.com/images/anime/12/35893.jpg   TV    Manga
## 2 https://myanimelist.cdn-dena.com/images/anime/13/58383.jpg   TV    Manga
## 3 https://myanimelist.cdn-dena.com/images/anime/11/10645.jpg   TV    Manga
## 4 https://myanimelist.cdn-dena.com/images/anime/13/32209.jpg   TV Original
## 5  https://myanimelist.cdn-dena.com/images/anime/6/41845.jpg   TV    Manga
## 6 https://myanimelist.cdn-dena.com/images/anime/12/21674.jpg   TV    Manga
##   episodes          status airing                aired_string
## 1       12 Finished Airing  False Jan 13, 2012 to Mar 30, 2012
## 2       26 Finished Airing  False   Apr 2, 2007 to Oct 1, 2007
## 3       51 Finished Airing  False  Oct 4, 2008 to Sep 25, 2009
## 4       38 Finished Airing  False Aug 16, 2002 to May 23, 2003
## 5       25 Finished Airing  False  Oct 6, 2012 to Mar 30, 2013
## 6       50 Finished Airing  False  Oct 4, 2009 to Sep 26, 2010
##                                       aired        duration
## 1 {'from': '2012-01-13', 'to': '2012-03-30'} 24 min. per ep.
## 2 {'from': '2007-04-02', 'to': '2007-10-01'} 24 min. per ep.
## 3 {'from': '2008-10-04', 'to': '2009-09-25'} 24 min. per ep.
## 4 {'from': '2002-08-16', 'to': '2003-05-23'} 16 min. per ep.
## 5 {'from': '2012-10-06', 'to': '2013-03-30'} 24 min. per ep.
## 6 {'from': '2009-10-04', 'to': '2010-09-26'} 24 min. per ep.
##                      rating score scored_by rank popularity members favorites
## 1 PG-13 - Teens 13 or older  7.63    139250 1274        231  283882      2809
## 2 PG-13 - Teens 13 or older  7.89     91206  727        366  204003      2579
## 3             PG - Children  7.55     37129 1508       1173   70127       802
## 4 PG-13 - Teens 13 or older  8.21     36501  307        916   93312      3344
## 5 PG-13 - Teens 13 or older  8.67    107767   50        426  182765      2082
## 6              G - All Ages  8.03     21618  526       1630   45625       826
##
## 1 Inu x Boku SS was licensed by Sentai Filmworks for North America, while MVM Films licensed it for
## 2
## 3
## 4                 Princess Tutu aired in two parts. The first part included 13 25-minute-long episo
## 5
## 6
##     premiered          broadcast
## 1 Winter 2012 Fridays at Unknown
## 2 Spring 2007            Unknown
## 3   Fall 2008            Unknown
## 4 Summer 2002 Fridays at Unknown
```

```
## 5   Fall 2012            Unknown
## 6   Fall 2009            Unknown
##
## 1
## 2 {'Adaptation': [{'mal_id': 759, 'type': 'manga', 'url': 'https://myanimelist.net/manga/759/Seto_no
## 3
## 4
## 5                                                                    {'Adaptation': [{'mal_id': 9711, '
## 6
##                                                                                                 producer
## 1 Aniplex, Square Enix, Mainichi Broadcasting System, Movic, Inu x Boku SS Production Partners
## 2                                                                  TV Tokyo, AIC, Square Enix, Sotsu
## 3                                                                                  TV Tokyo, Sotsu
## 4                                                                Memory-Tech, GANSIS, Marvelous AQL
## 5                                                                                      NHK, Shueisha
## 6                                                 Yomiuri Telecasting, DAX Production, Shueisha
##           licensor                        studio
## 1 Sentai Filmworks             David Production
## 2       Funimation                       Gonzo
## 3                                    Satelight
## 4        ADV Films               Hal Film Maker
## 5                                    J.C.Staff
## 6                 Studio Pierrot, Studio Hibari
##                                     genre
## 1   Comedy, Supernatural, Romance, Shounen
## 2 Comedy, Parody, Romance, School, Shounen
## 3            Comedy, Magic, School, Shoujo
## 4   Comedy, Drama, Magic, Romance, Fantasy
## 5          Comedy, Drama, Romance, Shounen
## 6                     Kids, School, Shoujo
##
## 1
## 2
## 3 ['#1: "Minna no Tamago (    )" by Shugo Chara Egg (eps 1-13)', '#2: "Shugo Shugo! (   !)" by Shugo
## 4
## 5
## 6
##
## 1 ['#1: "Nirvana" by MUCC (eps 1, 11-12)', '#2: "Rakuen no Photograph (  Photograph)" by Soushi Miket
## 2
## 3
## 4
## 5
## 6
```

```
dim(animelist)
```

```
## [1] 14478    30
```

#Data Cleaning Here we clean the data by converting the genre list into columns with booleans for each genre. We can also see there are 5 shows that have not yet aired with a score of 0.0 we will drop these to prevent them from skewing our data.

```r
animelist_with_genres = animelist %>%
  filter(!(status == "Not yet aired")) %>%
  mutate(genre_list = str_split(genre, ", ")) %>%
  unnest(genre_list) %>%
  filter(!is.na(genre_list), genre_list != "") %>%
  mutate(has_genre = 1) %>%
  pivot_wider(
    names_from = genre_list,
    values_from = has_genre,
    values_fill = 0
  )
```

In order to make the genre data more manageable and meaningful we will analyze the which genres are the most popular and decide a cutoff based on that

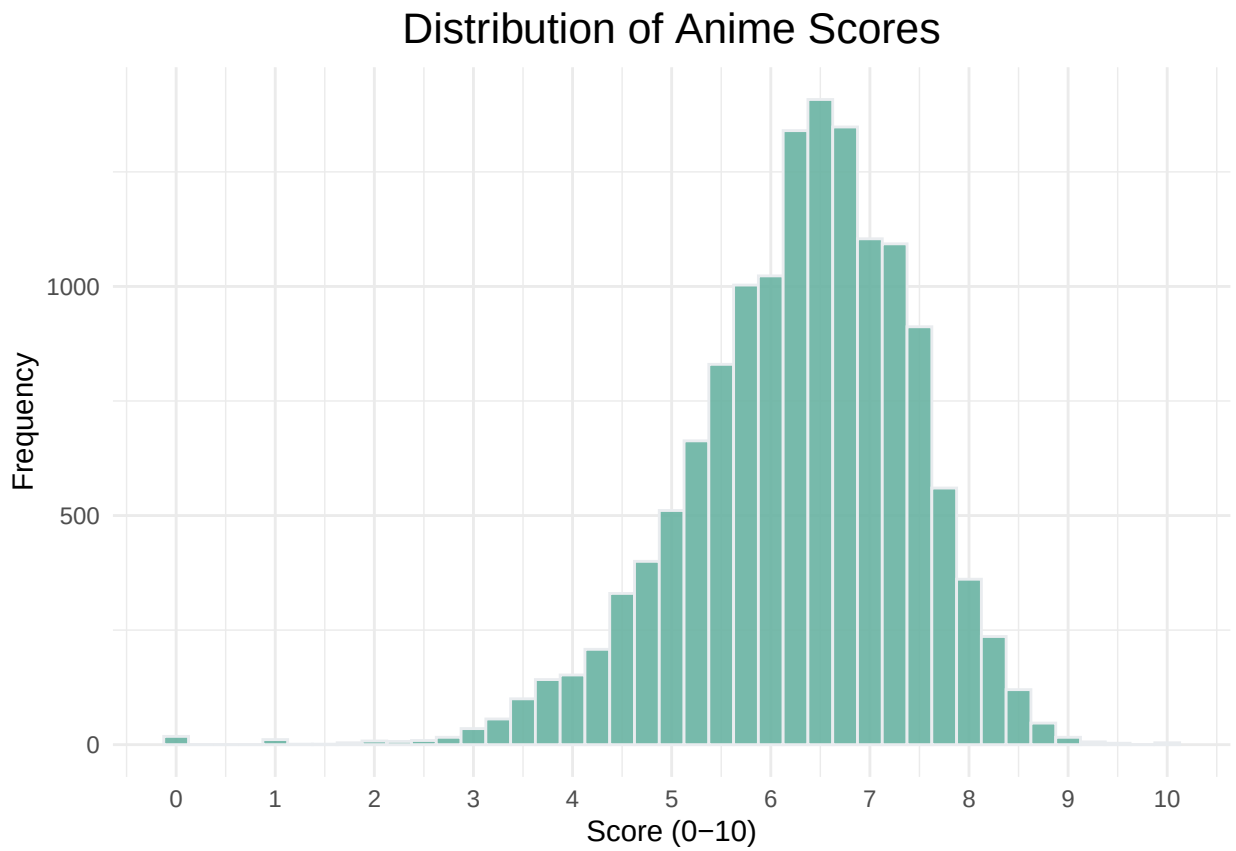
```r
genre_columns = names(animelist_with_genres)[!(names(animelist_with_genres) %in% names(animelist))]
genre_counts = sapply(animelist_with_genres[genre_columns], sum)
genre_counts = sort(genre_counts, decreasing = TRUE)
print(genre_counts)
```

```
##        Comedy        Action       Fantasy     Adventure         Drama
##          5147          3115          2621          2523          2229
##        Sci-Fi          Kids       Shounen       Romance Slice of Life
##          2209          1997          1733          1598          1488
##        School         Music        Hentai  Supernatural         Mecha
##          1356          1324          1211          1171           986
##    Historical         Magic         Ecchi        Seinen        Shoujo
##           942           896           676           670           630
##        Sports       Mystery        Parody   Super Power      Military
##           610           584           530           511           470
##         Space        Horror        Demons         Harem  Martial Arts
##           423           399           367           344           302
##      Dementia          Game Psychological        Police       Samurai
##           290           277           277           214           170
##       Vampire      Thriller          Cars         Josei    Shounen Ai
##           120            94            92            80            74
##     Shoujo Ai          Yuri          Yaoi
##            63            41            36
```

Based on the data I have decided to keep the all the genres, another thing is we want to get rid of shows that aren't scored by a large enough number of people.

#Histogram of the Score

```r
ggplot(animelist_with_genres, aes(x = score)) +
  geom_histogram(binwidth = 0.25, fill = "#69b3a2", color = "#e9ecef", alpha = 0.9) +
  labs(title = "Distribution of Anime Scores",
       x = "Score (0-10)",
       y = "Frequency") +
  theme_minimal() +
  scale_x_continuous(breaks = seq(0, 10, by = 1)) +
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```



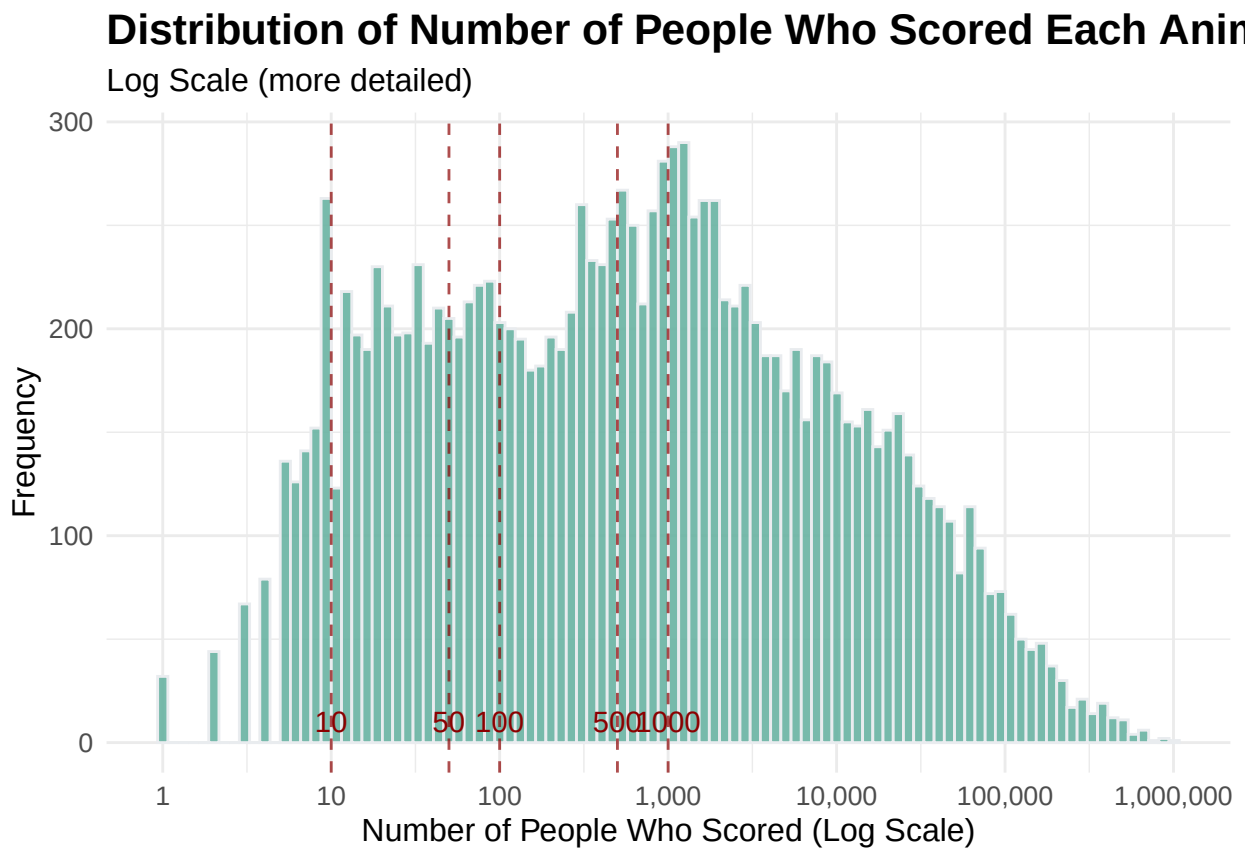

Based on this graph we have a lot of outliers at 0 so we will implement a threshold for anime we will considered if not enough users have viewed it before.

```
ggplot(animelist_with_genres, aes(x = scored_by)) +
  geom_histogram(bins = 100, fill = "#69b3a2", color = "#e9ecef", alpha = 0.9) +
  scale_x_log10(
    breaks = scales::trans_breaks("log10", function(x) 10^x),
    labels = scales::label_comma()
  ) +
  labs(
    title = "Distribution of Number of People Who Scored Each Anime",
    subtitle = "Log Scale (more detailed)",
    x = "Number of People Who Scored (Log Scale)",
    y = "Frequency"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10)
  ) +
  # Add vertical reference lines for potential thresholds
  geom_vline(xintercept = c(10, 50, 100, 500, 1000),
             linetype = "dashed",
             color = "darkred",
             alpha = 0.7) +
```

```
  # Add annotations for the reference lines
  annotate("text",
           x = c(10, 50, 100, 500, 1000),
           y = rep(0, 5),
           label = c("10", "50", "100", "500", "1000"),
           vjust = -0.5,
           color = "darkred")
```

```
## Warning in scale_x_log10(breaks = scales::trans_breaks("log10", function(x)
## 10^x), : log-10 transformation introduced infinite values.
```

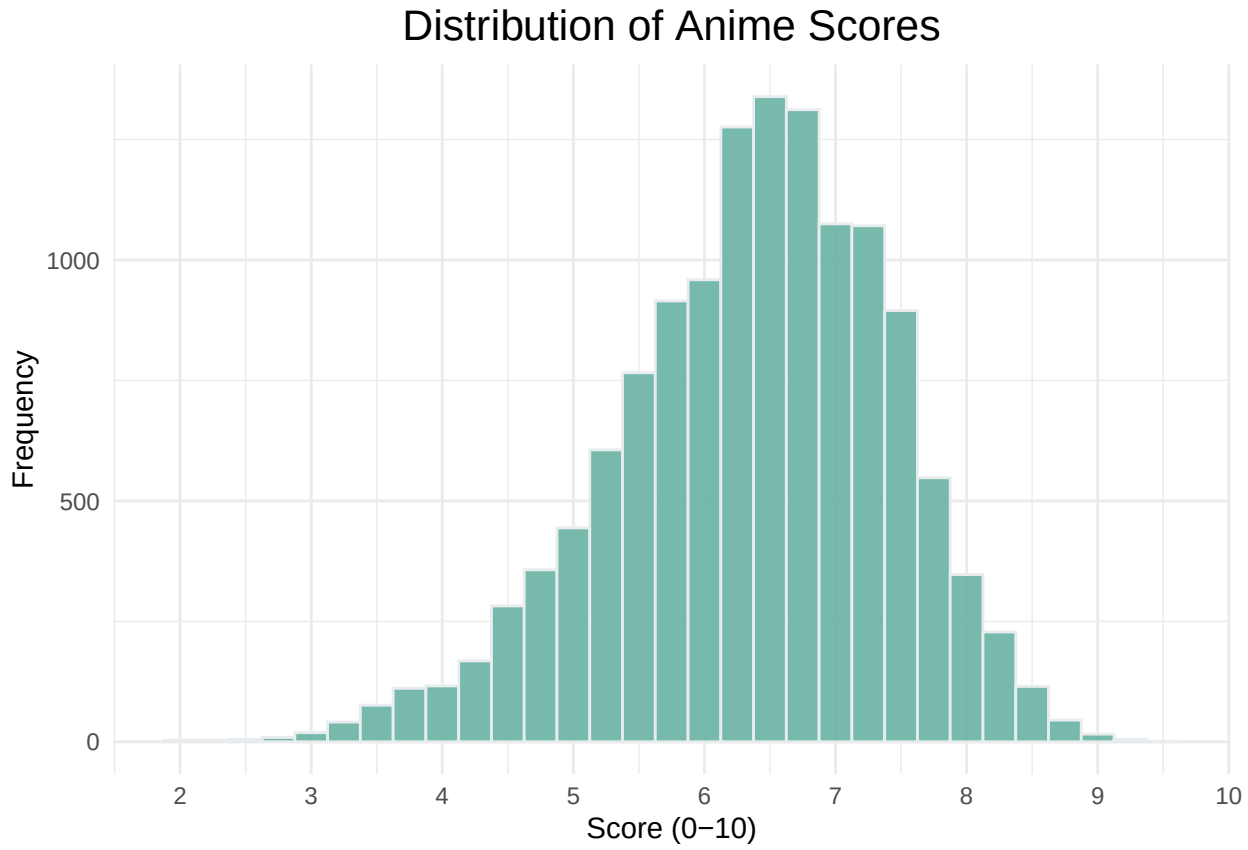
```
## Warning: Removed 18 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




We will start by getting rid of shows that have been scored less than 10 times and see if the number of 0 outliers change.

```
min_ratings_threshold = 10
animelist_with_genres = animelist_with_genres %>%
  filter(scored_by >= min_ratings_threshold)

ggplot(animelist_with_genres, aes(x = score)) +
  geom_histogram(binwidth = 0.25, fill = "#69b3a2", color = "#e9ecef", alpha = 0.9) +
  labs(title = "Distribution of Anime Scores",
       x = "Score (0-10)",
```

```
      y = "Frequency") +
  theme_minimal() +
  scale_x_continuous(breaks = seq(0, 10, by = 1)) +
  theme(plot.title = element_text(hjust = 0.5, size = 16))
```




We can see this eliminates a lot of the 0 scores which makes our data much more reasonable.

#Summary Statistics

```
summary(animelist_with_genres$score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.900   5.700   6.450   6.363   7.120   9.520
```

```
sd(animelist_with_genres$score, na.rm = TRUE)
```

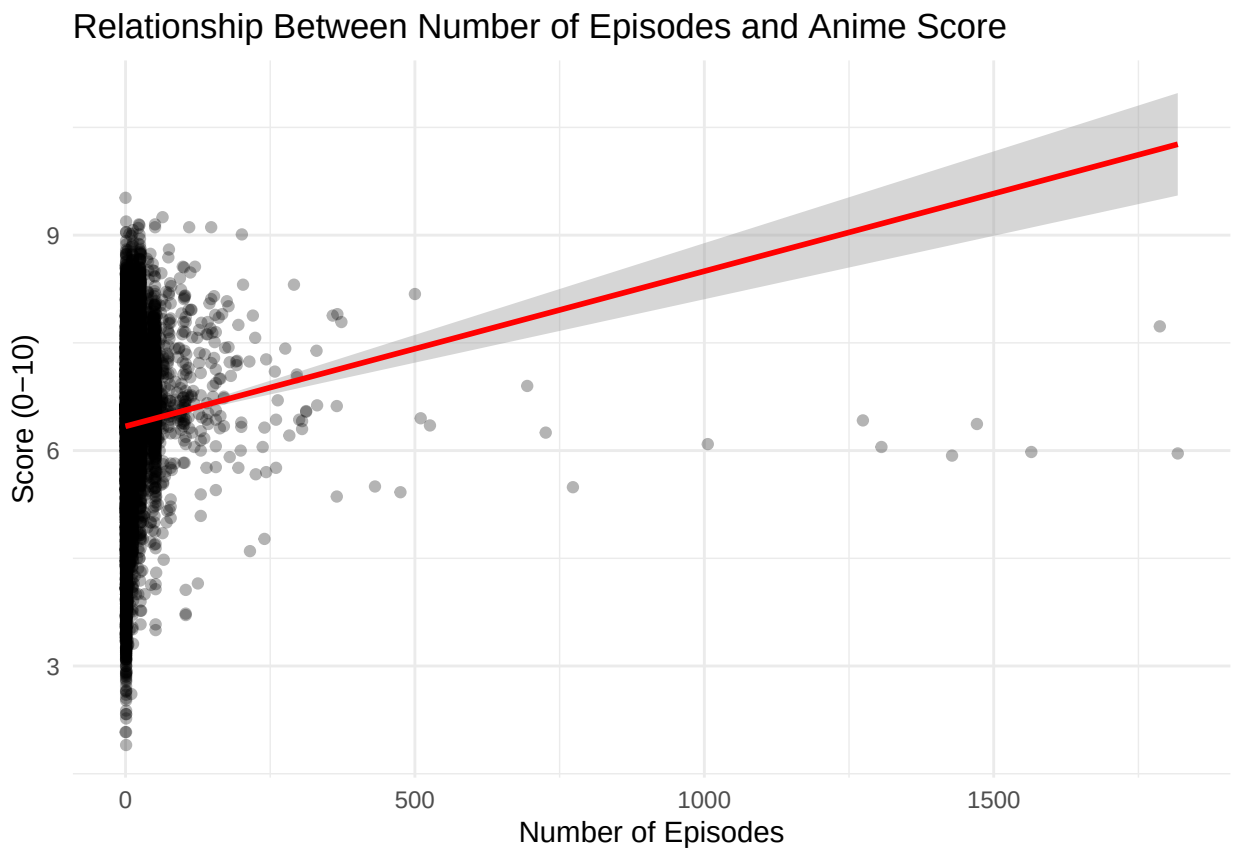
```
## [1] 1.051128
```

As we can see, most anime tend to score around 6.5, the exact median being 6.45 with a standard deviation of 1.058221. We had a set of outliers at 0 which seemed to be due to being rated by a small number of users. The data seems to have a longer tail on the left with a steeper drop as the scores get higher, showing that users seem to be more reluctant to give higher scores.

#Predictor Variables For our quantitative predictors we will use the number of episodes, scored_by, as I suspect that shows with a large number of ratings will have a more homogenized overall score, evening out to be very average. For categorical variables we will use Type, Source, Rating, and Studio. We will use genre data in later analysis.

```
ggplot(animelist_with_genres, aes(x = episodes, y = score)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", color = "red") +  # Linear trend line
  labs(
    title = "Relationship Between Number of Episodes and Anime Score",
    x = "Number of Episodes",
    y = "Score (0-10)"
  ) +
  theme_minimal()
```

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

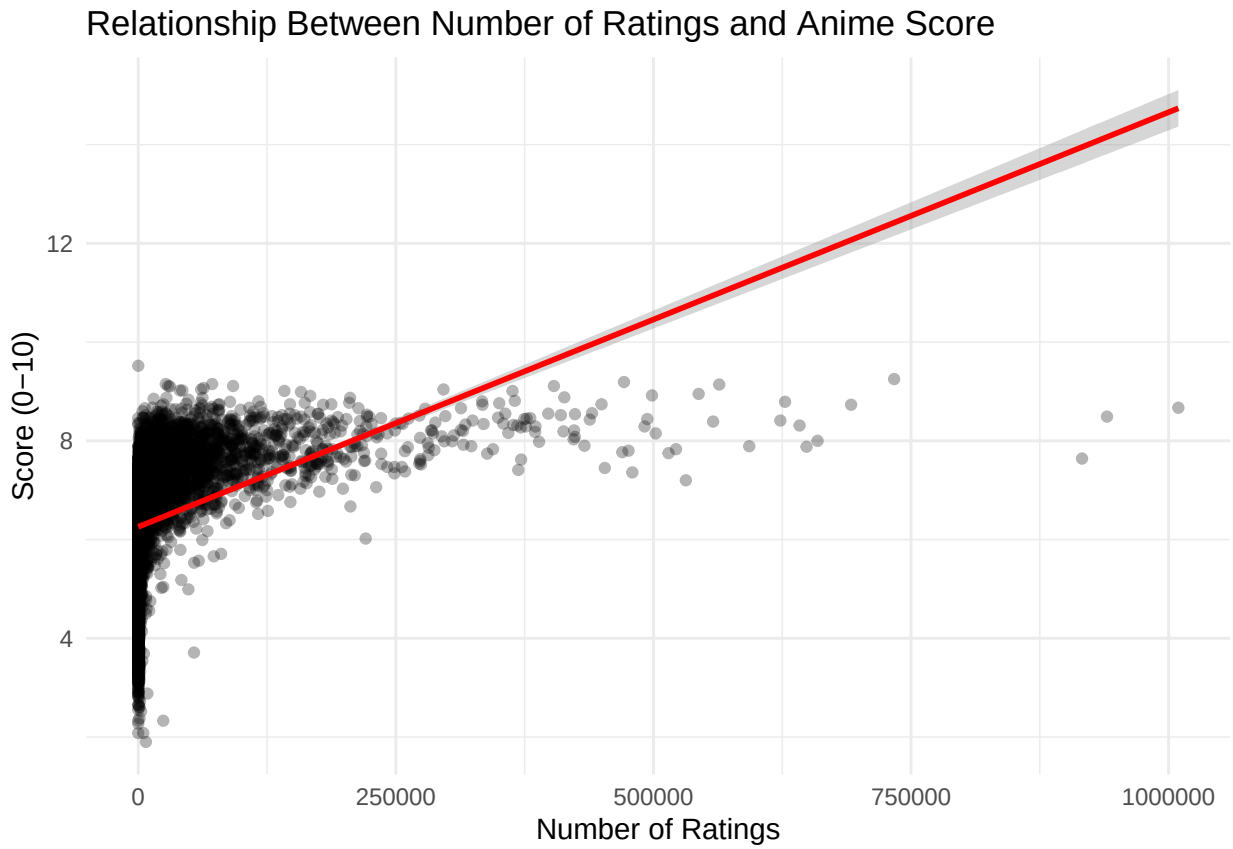



We can see that the number of episodes doesn't seem to have a very linear relationship, but as predicted, shows with larger numbers of episodes seem to have a very average score. That said there does still seem to be a concentration of higher scores given the higher number of episodes. This makes sense as only well-received shows would be expected to receive a longer runtime or else its likely the show would be canceled or pulled from air.

```
ggplot(animelist_with_genres, aes(x = scored_by, y = score)) +
  geom_point(alpha = 0.3) +

  geom_smooth(method = "lm", color = "red") +  # Linear trend line
  labs(
    title = "Relationship Between Number of Ratings and Anime Score",
    x = "Number of Ratings",
```

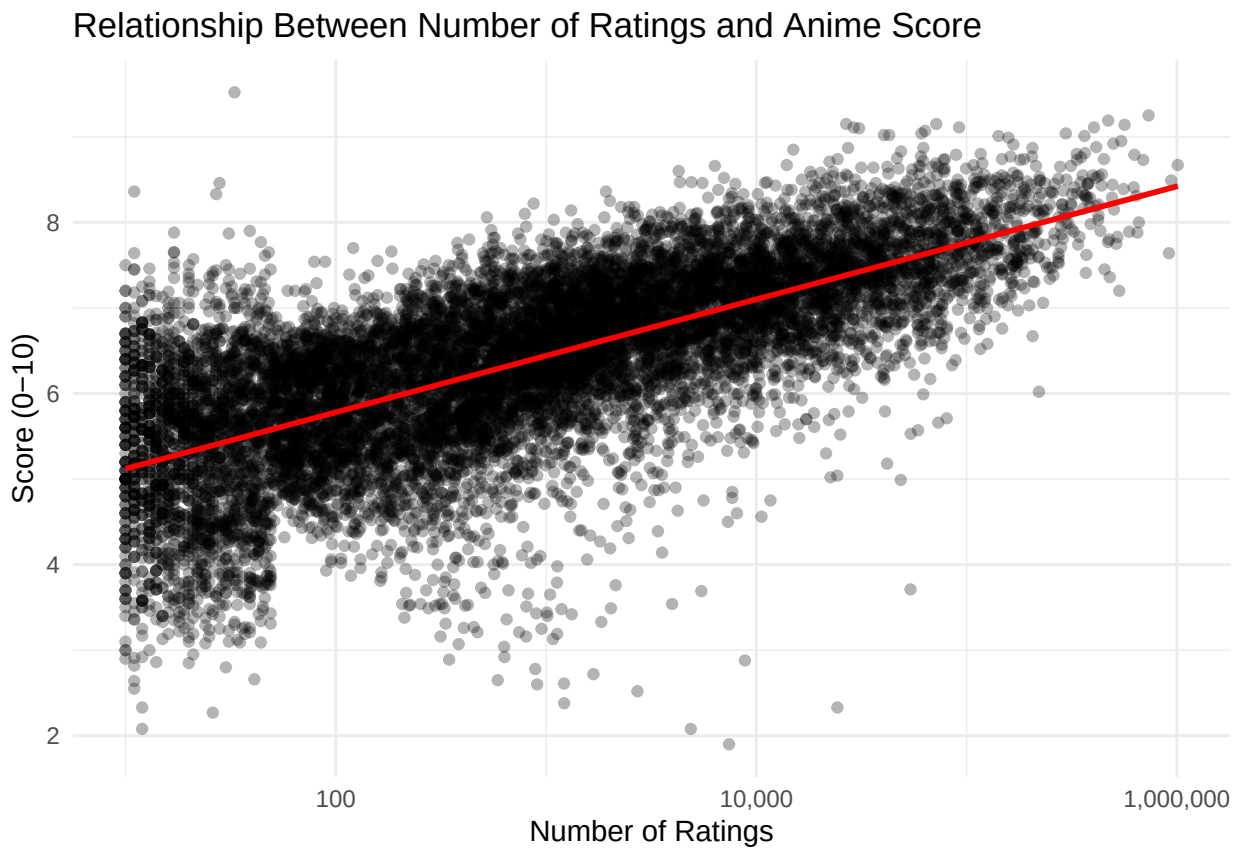
```
    y = "Score (0-10)"
  ) +
  theme_minimal()
```

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




```
ggplot(animelist_with_genres, aes(x = scored_by, y = score)) +
  geom_point(alpha = 0.3) +
  scale_x_log10(labels = scales::comma) +  # Log scale for x-axis
  geom_smooth(method = "lm", color = "red") +  # Linear trend line
  labs(
    title = "Relationship Between Number of Ratings and Anime Score",
    x = "Number of Ratings",
    y = "Score (0-10)"
  ) +
  theme_minimal()
```

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




The number of ratings has a much more linear relationship with the anime score, obviously capping out at a max score of 10. With a log scale for the number of rating, we see a much more linear relationship. This seems to suggest diminishing returns or an effect of crowd think, where users are influenced by an existing score that a show has.

#Initial Model

```
anime_model = lm(score ~ episodes + log(scored_by) + type + source + rating,
                 data = animelist_with_genres)
summary(anime_model)
```

```
##
## Call:
## lm(formula = score ~ episodes + log(scored_by) + type + source +
##     rating, data = animelist_with_genres)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0473 -0.3806  0.0513  0.4480  4.4686
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     4.6728318  0.0553949  84.355  < 2e-16 ***
## episodes                        0.0012006  0.0001419   8.463  < 2e-16 ***
## log(scored_by)                  0.2590552  0.0032934  78.659  < 2e-16 ***
## typeMusic                      -0.5327810  0.0346053 -15.396  < 2e-16 ***
```

```
## typeONA                                -0.4127626  0.0266893 -15.466  < 2e-16 ***
## typeOVA                                -0.0253645  0.0217328  -1.167 0.243190
## typeSpecial                            -0.0271771  0.0218762  -1.242 0.214143
## typeTV                                  0.0401051  0.0195253   2.054 0.039995 *
## sourceBook                              0.2133352  0.0934049   2.284 0.022388 *
## sourceCard game                         0.0701427  0.1067184   0.657 0.511019
## sourceDigital manga                    -1.0162035  0.2495412  -4.072 4.68e-05 ***
## sourceGame                             -0.0880912  0.0564989  -1.559 0.118981
## sourceLight novel                       0.0057602  0.0569293   0.101 0.919408
## sourceManga                             0.2311415  0.0496813   4.652 3.31e-06 ***
## sourceMusic                            -0.1347141  0.0692689  -1.945 0.051820 .
## sourceNovel                             0.3580154  0.0616191   5.810 6.39e-09 ***
## sourceOriginal                         -0.1697836  0.0500627  -3.391 0.000697 ***
## sourceOther                            -0.2228625  0.0623579  -3.574 0.000353 ***
## sourcePicture book                      0.0574859  0.0999175   0.575 0.565076
## sourceRadio                            -0.5955888  0.2496124  -2.386 0.017044 *
## sourceUnknown                           0.0286903  0.0501030   0.573 0.566907
## sourceVisual novel                      0.0765259  0.0556375   1.375 0.169019
## sourceWeb manga                         0.1521575  0.0776080   1.961 0.049948 *
## ratingNone                              0.1483322  0.0413315   3.589 0.000333 ***
## ratingPG-13 - Teens 13 or older         0.0782757  0.0182638   4.286 1.83e-05 ***
## ratingPG - Children                     0.1462560  0.0241517   6.056 1.44e-09 ***
## ratingR - 17+ (violence & profanity) -0.0447216  0.0283356  -1.578 0.114524
## ratingR+ - Mild Nudity                 -0.4058101  0.0289358 -14.024  < 2e-16 ***
## ratingRx - Hentai                      -0.2790237  0.0313060  -8.913  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6924 on 13124 degrees of freedom
## Multiple R-squared:  0.567,  Adjusted R-squared:  0.5661
## F-statistic: 613.8 on 28 and 13124 DF,  p-value: < 2.2e-16
```

#Model Interpritation

```
summary(anime_model)
```

```
##
## Call:
## lm(formula = score ~ episodes + log(scored_by) + type + source +
##     rating, data = animelist_with_genres)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0473 -0.3806  0.0513  0.4480  4.4686
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         4.6728318  0.0553949  84.355  < 2e-16 ***
## episodes                            0.0012006  0.0001419   8.463  < 2e-16 ***
## log(scored_by)                      0.2590552  0.0032934  78.659  < 2e-16 ***
## typeMusic                          -0.5327810  0.0346053 -15.396  < 2e-16 ***
## typeONA                            -0.4127626  0.0266893 -15.466  < 2e-16 ***
## typeOVA                            -0.0253645  0.0217328  -1.167 0.243190
```

```
## typeSpecial                            -0.0271771  0.0218762  -1.242 0.214143
## typeTV                                  0.0401051  0.0195253   2.054 0.039995 *
## sourceBook                              0.2133352  0.0934049   2.284 0.022388 *
## sourceCard game                         0.0701427  0.1067184   0.657 0.511019
## sourceDigital manga                    -1.0162035  0.2495412  -4.072 4.68e-05 ***
## sourceGame                             -0.0880912  0.0564989  -1.559 0.118981
## sourceLight novel                       0.0057602  0.0569293   0.101 0.919408
## sourceManga                             0.2311415  0.0496813   4.652 3.31e-06 ***
## sourceMusic                            -0.1347141  0.0692689  -1.945 0.051820 .
## sourceNovel                             0.3580154  0.0616191   5.810 6.39e-09 ***
## sourceOriginal                         -0.1697836  0.0500627  -3.391 0.000697 ***
## sourceOther                            -0.2228625  0.0623579  -3.574 0.000353 ***
## sourcePicture book                      0.0574859  0.0999175   0.575 0.565076
## sourceRadio                            -0.5955888  0.2496124  -2.386 0.017044 *
## sourceUnknown                           0.0286903  0.0501030   0.573 0.566907
## sourceVisual novel                      0.0765259  0.0556375   1.375 0.169019
## sourceWeb manga                         0.1521575  0.0776080   1.961 0.049948 *
## ratingNone                              0.1483322  0.0413315   3.589 0.000333 ***
## ratingPG-13 - Teens 13 or older         0.0782757  0.0182638   4.286 1.83e-05 ***
## ratingPG - Children                     0.1462560  0.0241517   6.056 1.44e-09 ***
## ratingR - 17+ (violence & profanity) -0.0447216  0.0283356  -1.578 0.114524
## ratingR+ - Mild Nudity                 -0.4058101  0.0289358 -14.024  < 2e-16 ***
## ratingRx - Hentai                      -0.2790237  0.0313060  -8.913  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6924 on 13124 degrees of freedom
## Multiple R-squared:  0.567,  Adjusted R-squared:  0.5661
## F-statistic: 613.8 on 28 and 13124 DF,  p-value: < 2.2e-16
```

```
anime_model$xlevels
```

```
## $type
## [1] "Movie"   "Music"   "ONA"     "OVA"     "Special" "TV"
##
## $source
##  [1] "4-koma manga"  "Book"          "Card game"     "Digital manga"
##  [5] "Game"          "Light novel"   "Manga"         "Music"
##  [9] "Novel"         "Original"      "Other"         "Picture book"
## [13] "Radio"         "Unknown"       "Visual novel"  "Web manga"
##
## $rating
## [1] "G - All Ages"                    "None"
## [3] "PG-13 - Teens 13 or older"       "PG - Children"
## [5] "R - 17+ (violence & profanity)" "R+ - Mild Nudity"
## [7] "Rx - Hentai"
```

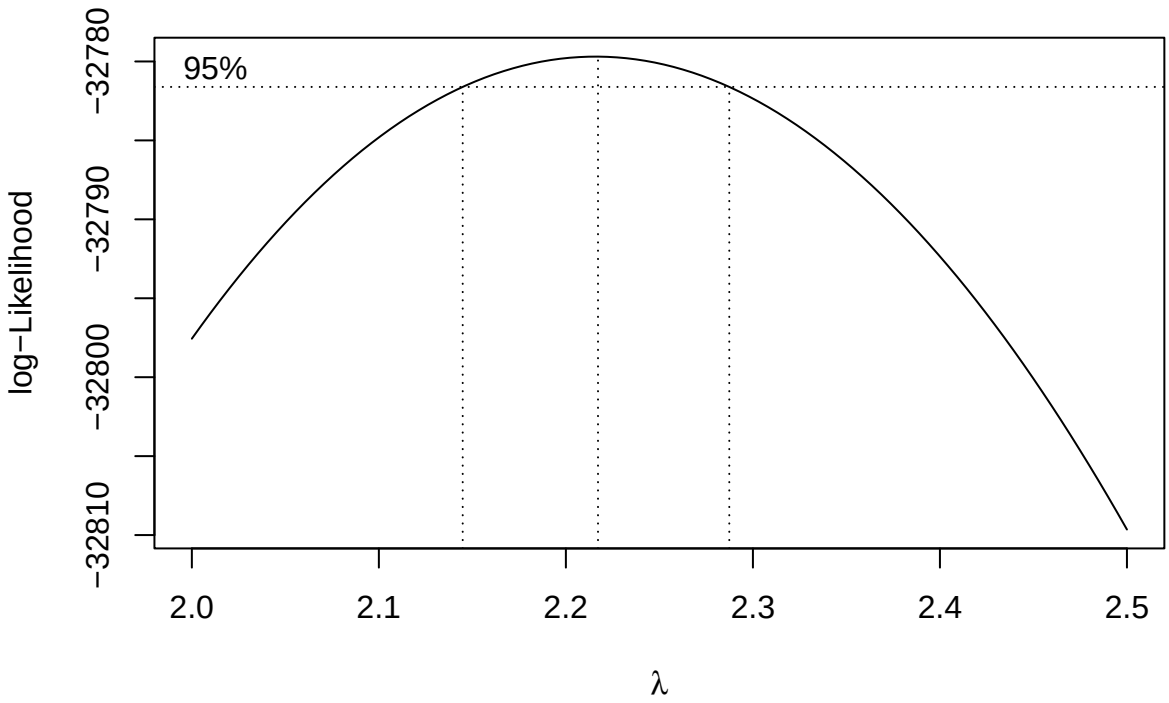
##Model Baseline Our model baseline would be an anime that has type=Movie, source=4-koma manga, and a rating=G - All Ages, thus our intercept, 4.6728318, would be the expected score for an anime that has 0 episodes, was scored by 1 person (since log), is a Movie, was adapted from a 4-koma manga, and has a rating of G - for all ages.

##Quantitative Predictor Interpretation We will interpret the coefficient for the episodes predictor, which is 0.0012006, this means when accounting for all other predictors and holding them constant, we expect the score of the anime to increase by 0.0012282 for each additional episode.

##Categorical Predictor Interpretation For the type variable when holding all other variables constant, when the anime is a Movie, the score is accounted for in the intercept, when the anime is a Music video, this lowers the score by 0.5327810, when it is an ONA (Original Net Animation) it lowers the score by 0.4127626, when it is an OVA (Original Video Animation) it lowers the score by 0.0253645, when it is a Special it lowers the score by 0.0271771, and when it is a TV series it increases the score by 0.0401051 We have 5 levels not including our baseline level, so this contributes 5 parameters towards p.

#Box-Cox Transformation

```
boxcox(anime_model, lambda = seq(2, 2.5, by = 0.05))
```



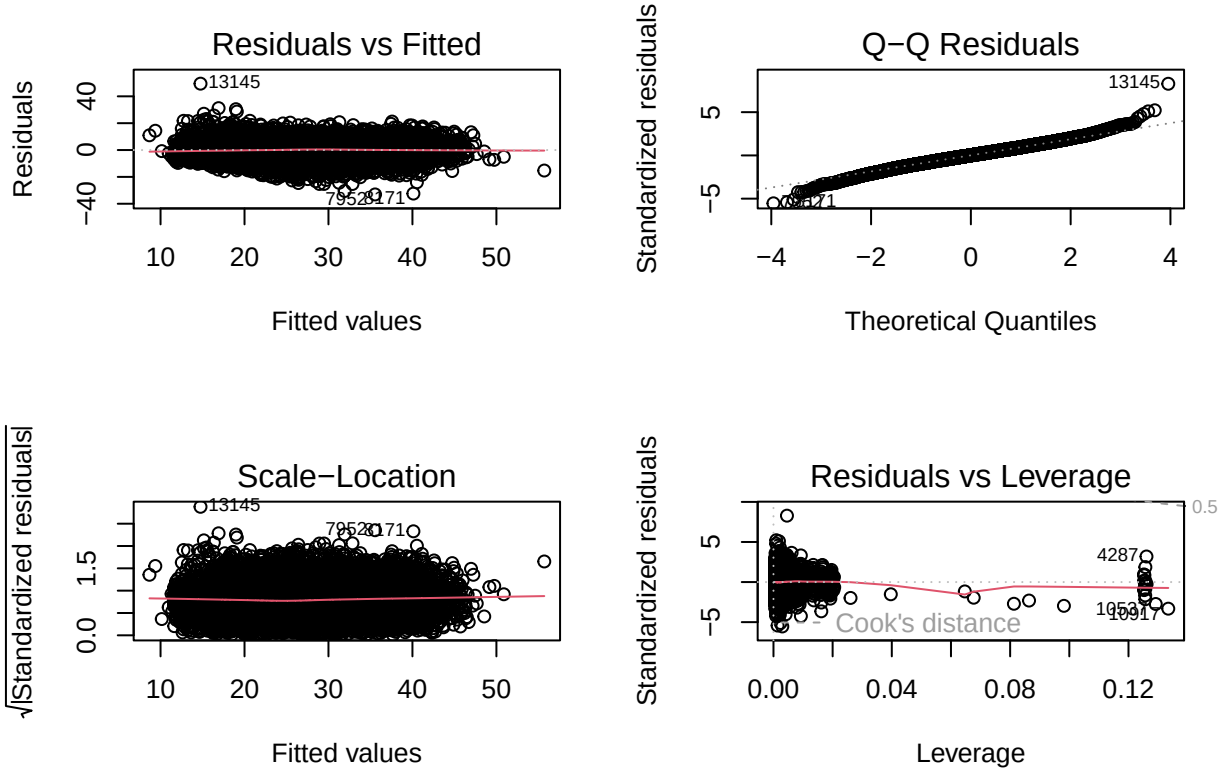

Looking at this we can see the 95% confidence interval looks to range from around 2.15 to 2.29, given that for simplicities sake we can try a 2.2 power transformation and compare against our previous model.

```
lambda = 2.2
animelist_with_genres = animelist_with_genres |>
  mutate(score_bc = (score^lambda - 1) / lambda)

model_bc = lm(score_bc ~ episodes + log(scored_by) + type + source + rating,
              data = animelist_with_genres)

summary(model_bc)
```

```
##
## Call:
## lm(formula = score_bc ~ episodes + log(scored_by) + type + source +
```

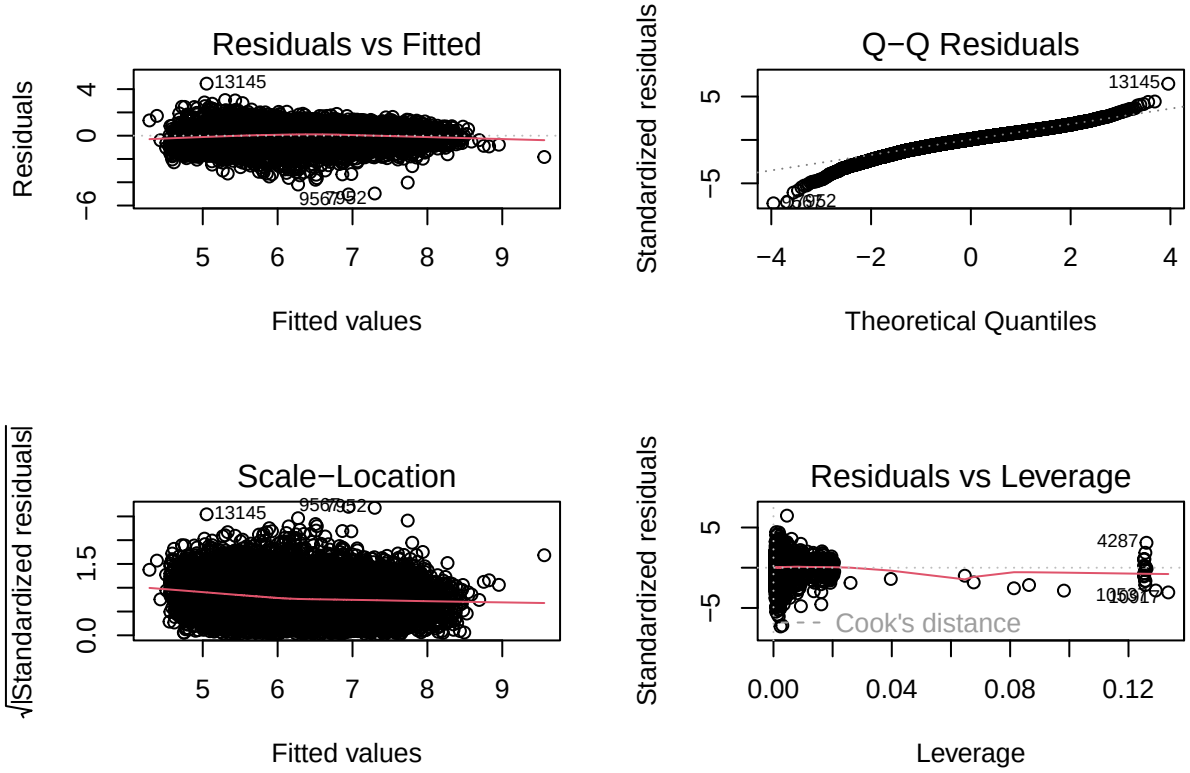
```
##     rating, data = animelist_with_genres)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.081  -3.693   0.077   3.816  49.437
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        11.535044   0.478688  24.097  < 2e-16 ***
## episodes                            0.010613   0.001226   8.657  < 2e-16 ***
## log(scored_by)                      2.421002   0.028459  85.069  < 2e-16 ***
## typeMusic                          -4.375983   0.299037 -14.634  < 2e-16 ***
## typeONA                            -3.464850   0.230632 -15.023  < 2e-16 ***
## typeOVA                            -0.666690   0.187802  -3.550 0.000387 ***
## typeSpecial                        -0.650464   0.189041  -3.441 0.000582 ***
## typeTV                              0.061537   0.168725   0.365 0.715326
## sourceBook                          1.830478   0.807146   2.268 0.023355 *
## sourceCard game                     0.298755   0.922193   0.324 0.745972
## sourceDigital manga                -8.706035   2.156378  -4.037 5.44e-05 ***
## sourceGame                         -1.033987   0.488228  -2.118 0.034208 *
## sourceLight novel                   0.276290   0.491947   0.562 0.574382
## sourceManga                         2.328886   0.429315   5.425 5.91e-08 ***
## sourceMusic                        -0.899981   0.598578  -1.504 0.132726
## sourceNovel                         3.363774   0.532474   6.317 2.75e-10 ***
## sourceOriginal                     -1.111261   0.432611  -2.569 0.010218 *
## sourceOther                        -1.771907   0.538857  -3.288 0.001011 **
## sourcePicture book                  0.648614   0.863424   0.751 0.452539
## sourceRadio                        -4.662685   2.156993  -2.162 0.030662 *
## sourceUnknown                       0.229077   0.432958   0.529 0.596747
## sourceVisual novel                  0.546766   0.480784   1.137 0.255460
## sourceWeb manga                     1.445966   0.670640   2.156 0.031094 *
## ratingNone                          1.484142   0.357161   4.155 3.27e-05 ***
## ratingPG-13 - Teens 13 or older     0.508688   0.157824   3.223 0.001271 **
## ratingPG - Children                 1.080163   0.208704   5.176 2.31e-07 ***
## ratingR - 17+ (violence & profanity) -0.323052   0.244858  -1.319 0.187076
## ratingR+ - Mild Nudity             -3.915957   0.250045 -15.661  < 2e-16 ***
## ratingRx - Hentai                  -2.801182   0.270526 -10.355  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.983 on 13124 degrees of freedom
## Multiple R-squared:  0.5936, Adjusted R-squared:  0.5928
## F-statistic: 684.7 on 28 and 13124 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(model_bc)
```




##Comparison + Linearity Assumptions

```
par(mfrow = c(2, 2))
plot(anime_model)
```




```
pred_orig = predict(anime_model)
pred_bc   = (predict(model_bc) * lambda + 1)^(1 / lambda)

rmse_orig = sqrt(mean((animelist_with_genres$score - pred_orig)^2))
rmse_bc   = sqrt(mean((animelist_with_genres$score - pred_bc)^2))

c(RMSE_untransformed = rmse_orig,
  RMSE_BoxCox        = rmse_bc)
```

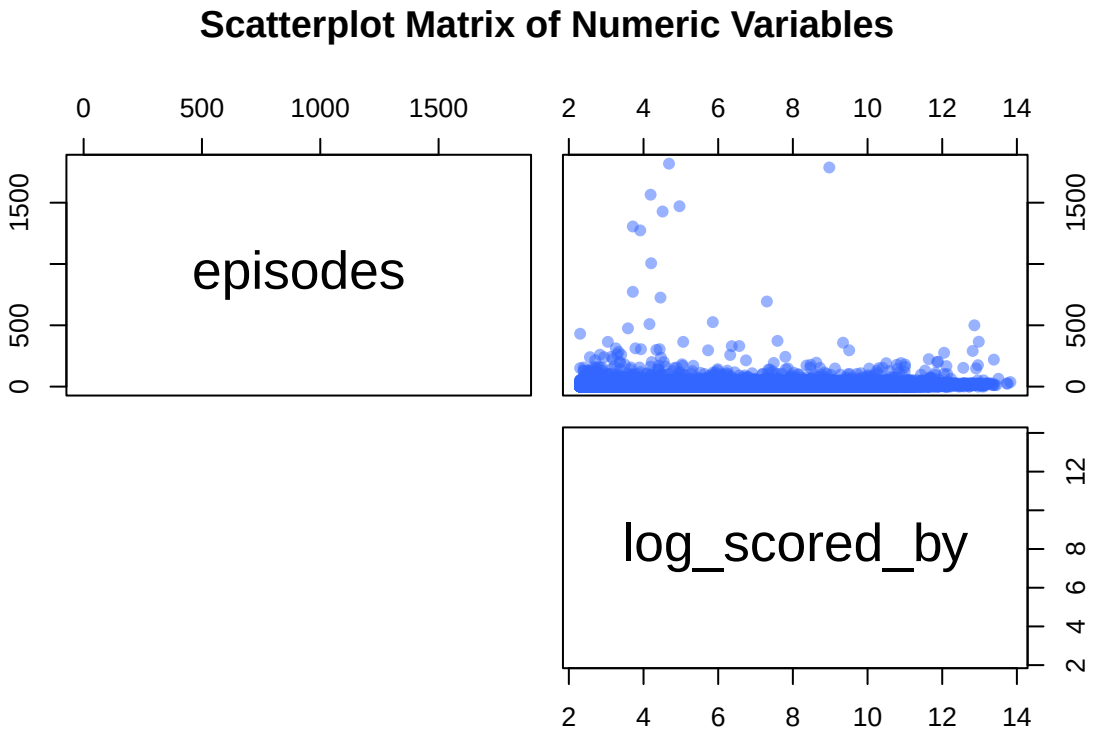
```
## RMSE_untransformed        RMSE_BoxCox
##          0.6916332          0.6912988
```

We can see that when comparing the residuals vs fitted graphs, our Power-Transformation model is better distributed horizontally, where as our original model was more weighted towards the loweer end of the scale. The same happens with the Q-Q residuals, with the power-transformation model fitting the diagonal line better than the old model. Although the difference in RMSE and $R^2$ is negligible, I think the difference in heteroscedasticity and tail departure from normality make the transformed model a valid choice for us to use.

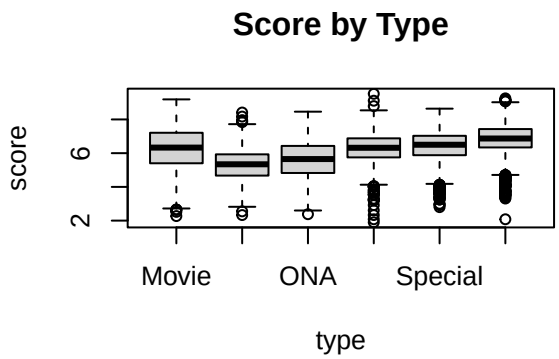
#Scatterplot Matrix

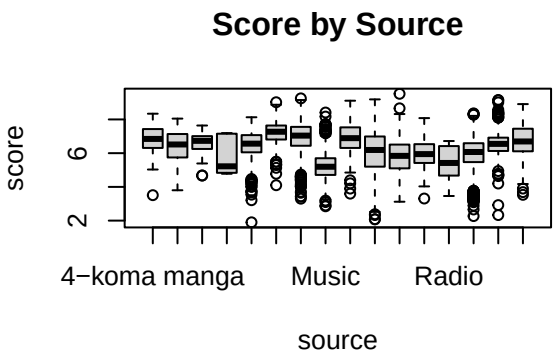
```
numeric_data = animelist_with_genres %>%
  select(episodes, scored_by) %>%
  mutate(log_scored_by = log(scored_by)) %>%
  select(-scored_by)
```

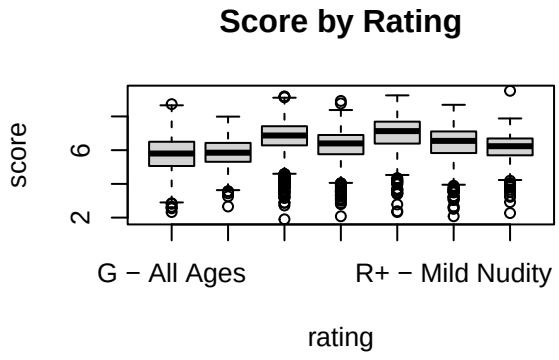
```r
pairs(numeric_data,
      main = "Scatterplot Matrix of Numeric Variables",
      pch = 16,
      col = "#3366FF80",
      lower.panel = NULL)
```




```r
par(mfrow = c(2, 2))
boxplot(score ~ type, data = animelist_with_genres, main = "Score by Type")
boxplot(score ~ source, data = animelist_with_genres, main = "Score by Source")
boxplot(score ~ rating, data = animelist_with_genres, main = "Score by Rating")
```




As we can see we don't really have any sort of relationship between the number of episodes and the amount of people scored by, the distribution is pretty equal with some outliers where there were shows with more episodes but scored by fewer people.

#Fitting a Second Model We plan to fit a second model including a new term for each of the top 5 most popular genres, to see if they have any influence on whether a show has a high score. The idea is that comedy shows may be more popular than another genre like action shows and we want to capture that data.

```
genre_model = lm(score_bc ~ episodes + log(scored_by) + type + source + rating + Comedy + Action + Fanta
             data = animelist_with_genres)
summary(genre_model)
```

```
##
## Call:
## lm(formula = score_bc ~ episodes + log(scored_by) + type + source +
##     rating + Comedy + Action + Fantasy + Adventure + Drama, data = animelist_with_genres)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.531  -3.694   0.062   3.795  49.183
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     11.665811   0.480387  24.284  < 2e-16 ***
## episodes                         0.010413   0.001209   8.614  < 2e-16 ***
## log(scored_by)                   2.343011   0.028436  82.395  < 2e-16 ***
## typeMusic                       -3.455274   0.300333 -11.505  < 2e-16 ***
```

```
## typeONA                                -2.892358   0.229778 -12.588  < 2e-16 ***
## typeOVA                                -0.496124   0.185408  -2.676 0.007463 **
## typeSpecial                            -0.262564   0.188512  -1.393 0.163698
## typeTV                                  0.237800   0.167231   1.422 0.155054
## sourceBook                              0.448480   0.804265   0.558 0.577109
## sourceCard game                        -0.171119   0.912059  -0.188 0.851179
## sourceDigital manga                    -8.631770   2.124566  -4.063 4.88e-05 ***
## sourceGame                             -2.016868   0.492096  -4.099 4.18e-05 ***
## sourceLight novel                      -0.368621   0.489544  -0.753 0.451470
## sourceManga                             1.456227   0.428113   3.401 0.000672 ***
## sourceMusic                            -1.673535   0.594742  -2.814 0.004902 **
## sourceNovel                             2.068773   0.536615   3.855 0.000116 ***
## sourceOriginal                         -1.910663   0.433784  -4.405 1.07e-05 ***
## sourceOther                            -2.472822   0.537749  -4.598 4.30e-06 ***
## sourcePicture book                     -0.076938   0.854054  -0.090 0.928221
## sourceRadio                            -5.543368   2.125997  -2.607 0.009133 **
## sourceUnknown                          -0.603131   0.433519  -1.391 0.164174
## sourceVisual novel                     -0.312778   0.480453  -0.651 0.515054
## sourceWeb manga                         1.013052   0.662406   1.529 0.126202
## ratingNone                              1.383184   0.353267   3.915 9.07e-05 ***
## ratingPG-13 - Teens 13 or older         0.287937   0.157969   1.823 0.068365 .
## ratingPG - Children                     0.851055   0.207812   4.095 4.24e-05 ***
## ratingR - 17+ (violence & profanity) -0.526484   0.250245  -2.104 0.035408 *
## ratingR+ - Mild Nudity                 -3.935232   0.249351 -15.782  < 2e-16 ***
## ratingRx - Hentai                      -1.875443   0.274181  -6.840 8.26e-12 ***
## Comedy                                  0.418327   0.120843   3.462 0.000538 ***
## Action                                  0.499463   0.136177   3.668 0.000246 ***
## Fantasy                                 0.513766   0.140915   3.646 0.000267 ***
## Adventure                               1.279817   0.144111   8.881  < 2e-16 ***
## Drama                                   2.573537   0.151705  16.964  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.895 on 13119 degrees of freedom
## Multiple R-squared:  0.6057, Adjusted R-squared:  0.6047
## F-statistic: 610.7 on 33 and 13119 DF,  p-value: < 2.2e-16
```

### Model Selection

```
step_bc = stepAIC(
  genre_model,
  scope = list(
    lower = model_bc,
    upper = genre_model
  ),
  direction = "both",
  trace = FALSE
)
summary(step_bc)
```

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

```
## lm(formula = score_bc ~ episodes + log(scored_by) + type + source +
##     rating + Comedy + Action + Fantasy + Adventure + Drama, data = animelist_with_genres)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.531  -3.694   0.062   3.795  49.183
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         11.665811   0.480387  24.284  < 2e-16 ***
## episodes                             0.010413   0.001209   8.614  < 2e-16 ***
## log(scored_by)                       2.343011   0.028436  82.395  < 2e-16 ***
## typeMusic                           -3.455274   0.300333 -11.505  < 2e-16 ***
## typeONA                             -2.892358   0.229778 -12.588  < 2e-16 ***
## typeOVA                             -0.496124   0.185408  -2.676 0.007463 **
## typeSpecial                         -0.262564   0.188512  -1.393 0.163698
## typeTV                               0.237800   0.167231   1.422 0.155054
## sourceBook                           0.448480   0.804265   0.558 0.577109
## sourceCard game                     -0.171119   0.912059  -0.188 0.851179
## sourceDigital manga                 -8.631770   2.124566  -4.063 4.88e-05 ***
## sourceGame                          -2.016868   0.492096  -4.099 4.18e-05 ***
## sourceLight novel                   -0.368621   0.489544  -0.753 0.451470
## sourceManga                          1.456227   0.428113   3.401 0.000672 ***
## sourceMusic                         -1.673535   0.594742  -2.814 0.004902 **
## sourceNovel                          2.068773   0.536615   3.855 0.000116 ***
## sourceOriginal                      -1.910663   0.433784  -4.405 1.07e-05 ***
## sourceOther                         -2.472822   0.537749  -4.598 4.30e-06 ***
## sourcePicture book                  -0.076938   0.854054  -0.090 0.928221
## sourceRadio                         -5.543368   2.125997  -2.607 0.009133 **
## sourceUnknown                       -0.603131   0.433519  -1.391 0.164174
## sourceVisual novel                  -0.312778   0.480453  -0.651 0.515054
## sourceWeb manga                      1.013052   0.662406   1.529 0.126202
## ratingNone                           1.383184   0.353267   3.915 9.07e-05 ***
## ratingPG-13 - Teens 13 or older      0.287937   0.157969   1.823 0.068365 .
## ratingPG - Children                  0.851055   0.207812   4.095 4.24e-05 ***
## ratingR - 17+ (violence & profanity) -0.526484   0.250245  -2.104 0.035408 *
## ratingR+ - Mild Nudity              -3.935232   0.249351 -15.782  < 2e-16 ***
## ratingRx - Hentai                   -1.875443   0.274181  -6.840 8.26e-12 ***
## Comedy                               0.418327   0.120843   3.462 0.000538 ***
## Action                               0.499463   0.136177   3.668 0.000246 ***
## Fantasy                              0.513766   0.140915   3.646 0.000267 ***
## Adventure                            1.279817   0.144111   8.881  < 2e-16 ***
## Drama                                2.573537   0.151705  16.964  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.895 on 13119 degrees of freedom
## Multiple R-squared:  0.6057, Adjusted R-squared:  0.6047
## F-statistic: 610.7 on 33 and 13119 DF,  p-value: < 2.2e-16
```

We do stepwise model selection using AIC as our metric, with our original model as the baseline and our genre model as the upper. We can see that we end selecting our genre model as the best model and that our new model has a higher Adjusted R-squared of 0.6047 compared to our previous model which had an Adjusted R-squared of 0.5928. We can further validate this with an ANOVA Test.

##Anova Test

```
anova(model_bc, genre_model)
```

```
## Analysis of Variance Table
##
## Model 1: score_bc ~ episodes + log(scored_by) + type + source + rating
## Model 2: score_bc ~ episodes + log(scored_by) + type + source + rating +
##     Comedy + Action + Fantasy + Adventure + Drama
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1  13124 469831
## 2  13119 455842  5     13989 80.518 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our null hypothesis $H_0$ is that the additional genre variables has no effect on the score of the anime and our Alternate hypothesis is that it does have an effect. Here we have an F-statistic, not a test statistic which is 80.518. We can see that our p-value is $2.2e-16$ which is far below 0.05, meaning we reject the null hypothesis, indicating that the genre variables provides statistically significant information beyond the existing predictors.

#Model Analysis We settle upon the genre_model which includes the five most popular genres and whether a show has that genre as our best model. We can see from the previous ANOVA test and our model selection that adding these genres provide valuable information, even when penalizing for model complexity.

##Fitted Model Since we have many levels for our categorical variables, we will write a specific model for a TV show that was adapted from a Novel with a PG-13 rating. Our model would be $\hat{y}$ (power-transformed score) = 11.665811+0.237800 (since TV show)+2.068773 (since adapted from Novel)+ 0.287937 (since PG-13) + 0.010413 $*$ number of episodes + 2.343011 $*$ log(number of people scored by) + 0.418327(if comedy)+0.499463(if action)+0.513766(if fantasy)+1.279817(if adventure)+2.573537(if drama)
Or: $\hat{y}$ (power-transformed score) = 14.260321+0.010413$*$number of episodes+2.343011$*$log(number of people scored by)+ 0.418327(if comedy)+0.499463(if action)+0.513766(if fantasy)+1.279817(if adventure)+2.573537(if drama)

##Model Size For this model our we have n = 13153 and p = 34 . Since we want to have at least 5-10 observations per coefficient, model complexity is not a concern since we have $\frac{13153}{34} \approx 386.9$ observations per coefficient, which is well above the suggested 5-10 threshold.

```
c(n = nobs(genre_model),
  p = length(coef(genre_model)))
```

```
##     n     p
## 13153    34
```

## Standard Deviation/Variance Confirmation

```
sd_y  = sd(animelist_with_genres$score_bc, na.rm = TRUE)
sd_yest = summary(genre_model)$sigma
c(sd_y, sd_yest)
```

```
## [1] 9.375855 5.894636
```

```r
1 - (sd_yest/sd_y)^2
```

```
## [1] 0.6047314
```

The standard deviation of y shows how much spread the scored_bc variable has and the standard deviation of the error shows how much spread the error has after our model showing that we have cut down on unexplained spread from 9.4 to 5.9, we expect these values to correspond to the adjusted $R^2$ of the model, which they do

##Collinearity We covered this early but we will do this again with vif.

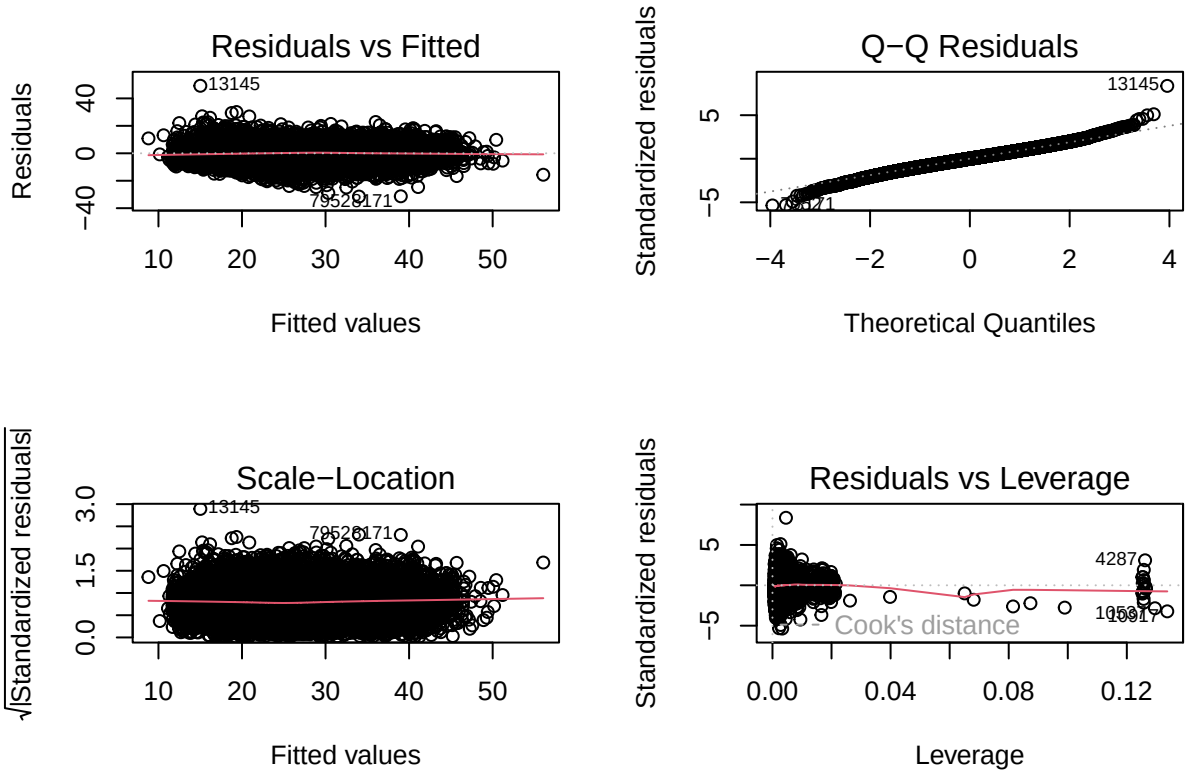
```r
num_only_model = lm(
  score_bc ~ episodes + log(scored_by),
  data = animelist_with_genres
)
vif(num_only_model)
```

```
##       episodes log(scored_by)
##       1.000134       1.000134
```

Since both our VIF values are well below 5, multicollinearity is not a cause of concern for us.

##Model Assumption We also covered this earlier but again

```r
par(mfrow = c(2, 2))
plot(genre_model)
```




We can see that our residuals are evenly distributed, and the scale location plot is also nearly horizontal,

indicating linearity and homoscedasticity. The normality of errors looks good from the Q-Q residuals as they hug the line except for the tails, and we will investigate outliers/leverage points later but we don't have too many issues. We do have points of high leverage, these correspond to popular shows that have many more people scoring them compared to regular shows. I think this is still important data, as if these shows achieve that mainstream appeal, there should be some indicators from the data.

#Error Estimation

```
res   = resid(genre_model)
h     = hatvalues(genre_model)
e_loo = res / (1 - h)

rmse_loo = sqrt(mean(e_loo^2))
rmse_loo
```

```
## [1] 5.905177
```

```
score0  = 6
rmse_raw = rmse_loo / (lambda * score0^(lambda - 1))
rmse_raw
```

```
## [1] 0.3126286
```

Our error is 5.905177 and after we convert it back from the transformation we get a roughly 0.313 error which is quite small.