The Simpsons by the Data

Anahita Bahri

October 23, 2016

Description of the Datasets

Overall, these datasets contain the characters, locations, episode details, and script lines for approximately 600 Simpsons episodes, dating back to 1989. You can find this data on Kaggle.

Join the Datasets

```
# using dplyr's left_join, join all the datasets
simpsons_data <- left_join(lines, locations, by = c(location_id = "id"))
simpsons_data <- left_join(simpsons_data, episodes, by = c(episode_id = "id"))
simpsons_data <- left_join(simpsons_data, characters, by = c(character_id = "id"))

# there are way too many columns. let's trim it down.
simpsons_data <- dplyr::select(simpsons_data, id, episode_id, number, timestamp_in_ms,
    speaking_line, raw_character_text, raw_location_text, normalized_text, word_count,
    title, original_air_date, season, number_in_season, number_in_series, us_viewers_in_millions,
    views, imdb_rating, gender)
```

Manipulate Variables

Change data type of word count to numeric. Convert gender and IMDb rating to binary, for logistic regression purposes. After doing some research, I was able to find that a rating of 7 or above is considered "good," which is why I've chosen this in the below code, when converting rating to binary.

```
simpsons_data$word_count <- as.numeric(simpsons_data$word_count)

# convert gender to binary variable
simpsons_data$female <- ifelse(simpsons_data$gender == "f", 1, ifelse(simpsons_data$gender ==
    "m", 0, "NA"))
simpsons_data$female <- as.numeric(simpsons_data$female)

# convert rating to binary (research shows 7 and above)
simpsons_data$rating <- ifelse(simpsons_data$imdb_rating >= 7, 1, 0)
```

Create Episode, Character, and Words Subsets

```
episodes_subset <- sqldf("SELECT episode_id, title, original_air_date, season, number_in_season, number
  FROM simpsons_data
  GROUP BY episode_id")

character_subset_NA <- sqldf("SELECT raw_character_text, female, SUM(word_count) as sum_word_count, COU
  FROM simpsons_data
  GROUP BY raw_character_text
  ORDER BY 3 DESC, raw_character_text")
```

```
character_subset <- na.omit(character_subset_NA)
character_subset$female <- as.integer(character_subset$female)

words_by_character <- sqldf("SELECT raw_character_text, gender, SUM(word_count) as sum_word_count, COUN
  FROM simpsons_data
  WHERE gender IS NOT \"NA\"
  GROUP BY raw_character_text
  ORDER BY 3 DESC, raw_character_text")
```

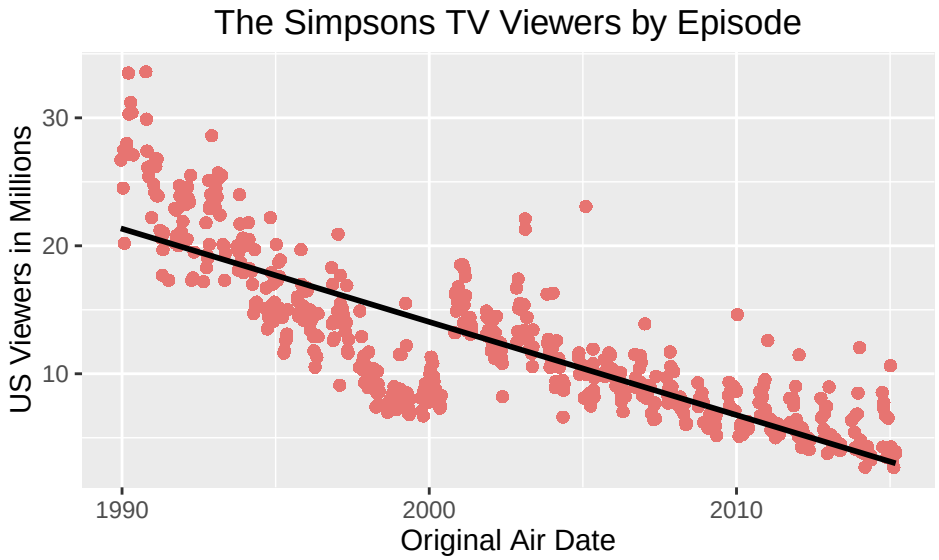
As someone who is a fan of all things entertainment, once I discovered this dataset, I knew I had to get my hands dirty with the data at hand.

While doing some exploratory analysis, I noticed gender imbalance in terms of total words spoken throughout all episodes: only 3 of the top 20 characters (by spoken word) were female. With this mind, I wanted to answer some gender-related questions. Also, as someone who has been at the center of "disruptive innovation," particularly when it comes to music, I also wanted to explore how this played out in the TV realm; streaming TV like Netflix, Hulu, among others, would be considered "disruptive innovation" here.

I also wanted to add another layer to my analysis by comparing ratings and viewership for The Simpsons to a similar TV show that's been considered competition for well over a decade: Family Guy. After running a quick Google search on "The Simpsons vs. Family Guy," I see many Buzzfeed style articles on the many reasons why The Simpsons is better than Family Guy. Do the numbers agree?

Here are some questions I'd like to answer through this project:
1. How has The Simpsons performed over the years, rating and viewership wise?
2. How does The Simpsons compare to ratings and viewers of Family Guy?
3. Is there a particular combination of things that can increase the IMDb rating, i.e. more words, more unique locations in an episode?
4. Can you predict the gender of the character based on the total number of words spoken, total episodes they've been featured on, unique locations they've visited while on the show, in addition to a combination of specific words spoken?

Regression Analysis: Singe Predictor

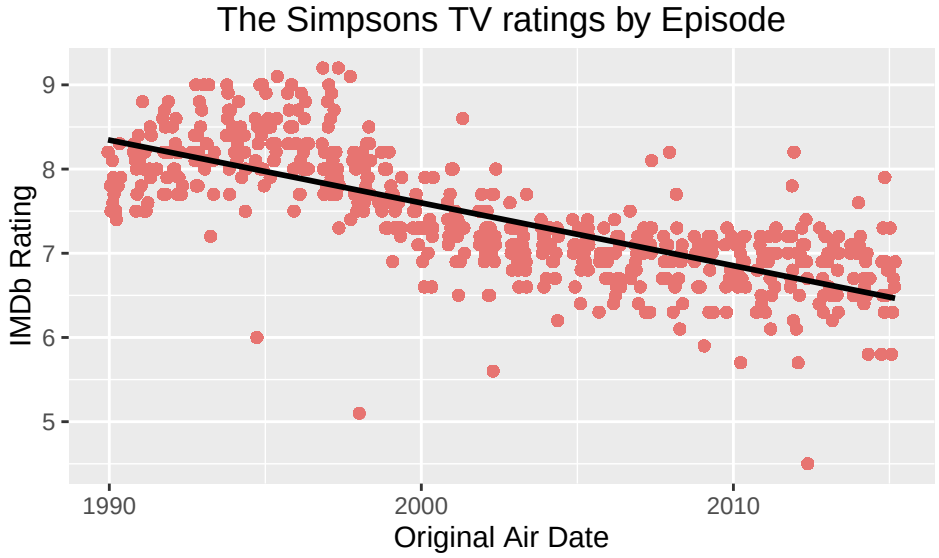


	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	35.885	0.04	895.900	0

	Estimate	Std. Error	t value	Pr(>|t|)
original_air_date	-0.002	0.00	-599.853	0

Predicted number of viewers for all episodes, dating back to 1989, by original air date. For every additional episode (or date), you can expect the viewership to decrease by an average of 0.002.

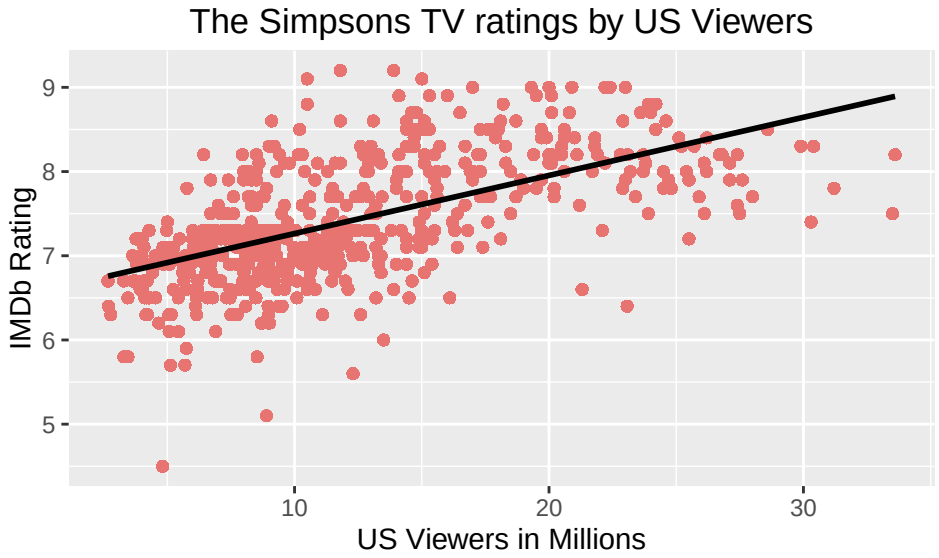
Viewership has gone down substantialy over the years. How about episode ratings over the years?



	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	9.836	0.005	1835.363	0
original_air_date	0.000	0.000	-459.077	0

Predicted IMDb ratings for all episodes, dating back to 1989, by original air date. For every additional episode (or date), you can expect the rating to decrease by an average of 0.0002.

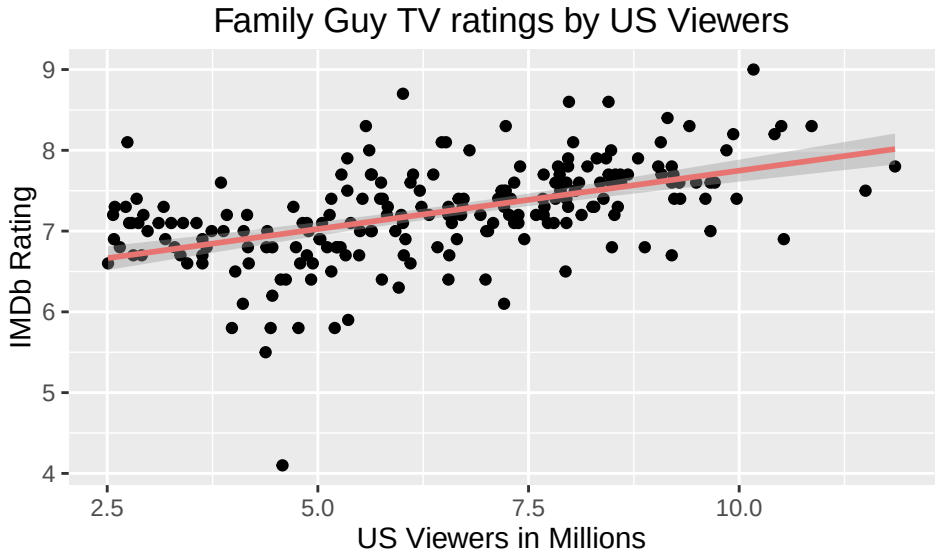
Ratings have also gone down slightly over the years, but not as drastically as viewers. Why may this be the case? The rise of TV streaming!



	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	6.574	0.003	2083.214	0
us_viewers_in_millions	0.069	0.000	305.988	0

Predicted IMDb ratings for all episodes, dating back to 1989, by US Viewers in Millions. For every additional US viewer (in millions), you can expect the rating to increase by an average of 0.069. We expect a positive linear relationship vs. negative linear relationship here, since both US Viewership and IMDb rating have gone down over the years, as we saw in the previous 2 plots.

How does this compare to Family Guy?



	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	6.30	0.11	55.44	0
us_viewers_in_millions	0.14	0.02	8.62	0

Predicted IMDb ratings for episodes since season 4, by US Viewers in Millions. For every additional US viewer (in millions), you can expect the rating to increase by an average of 0.14, higher than that of The Simpsons. However, when you compare ranges of the Viewership in Millions with The Simpsons, one quickly realizes that fewer people are watching Family Guy.

How about logistic regression? Let's predict a "good" IMDb rating by US Viewers.

```
rating_fit <- glm(rating ~ us_viewers_in_millions, family=binomial(link="logit"), data=episodes_subset)

kable(summary(rating_fit)$coef, digits=2)
```

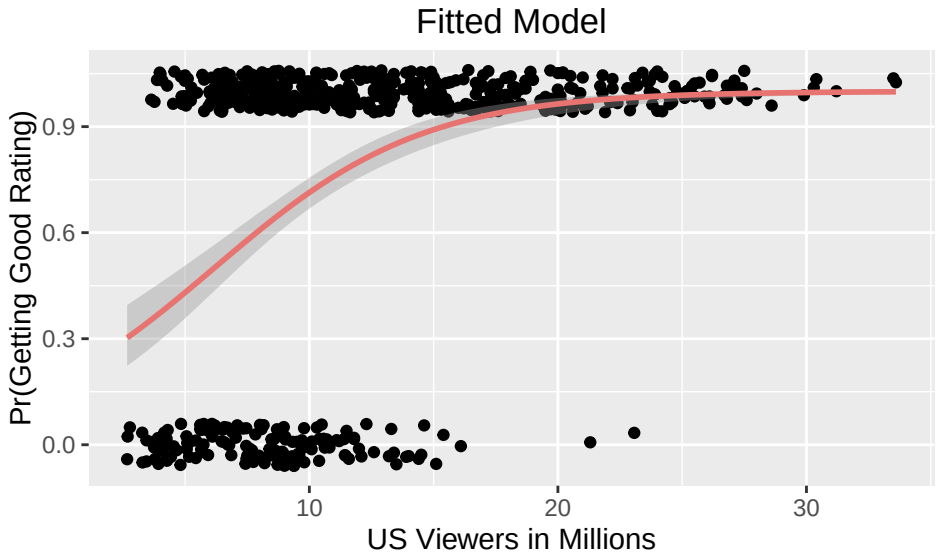
	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1.47	0.28	-5.30	0
us_viewers_in_millions	0.24	0.03	8.41	0

```
invlogit(-1.47)
```

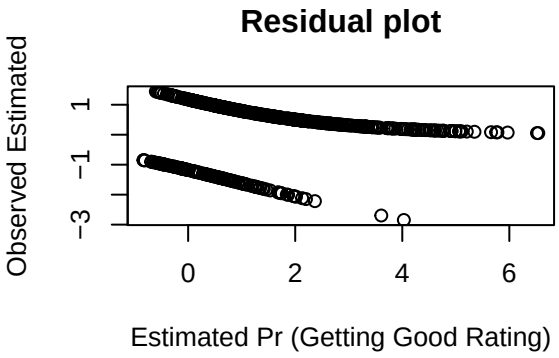
```
## [1] 0.1869426
```

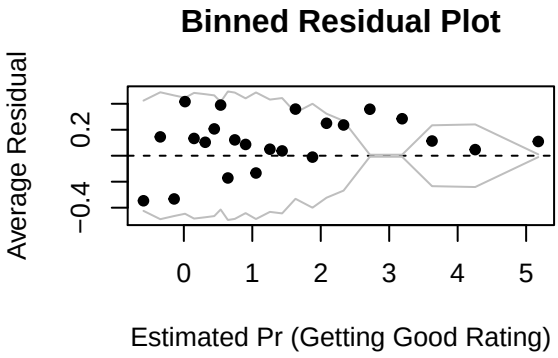
This model estimates a 19% probability of getting a "good" rating if no viewers are watching. I suppose this could make sense, since our viewership data relies on those watching actual TV. Instead of watching actual TV, they watch on streaming TV. An increase in a million viewers would indicate a 6% increase in probability of getting a "good" rating.

Let's plot this!



Let's run some diagnostics by making a residual plot and binned residual plot.





There's a strong pattern in the residual plot, which is why we rely on the binned residual plot instead.

Regression Analysis: Multiple Predictors

Linear Regression: IMDb Rating Predicted by US Viewers, Sum of Word Count, and Number in Series

```
rating_fit_1 <- lm(imdb_rating ~ us_viewers_in_millions + sum_word_count + number_in_series, data=episo

arm::display(rating_fit_1)
```

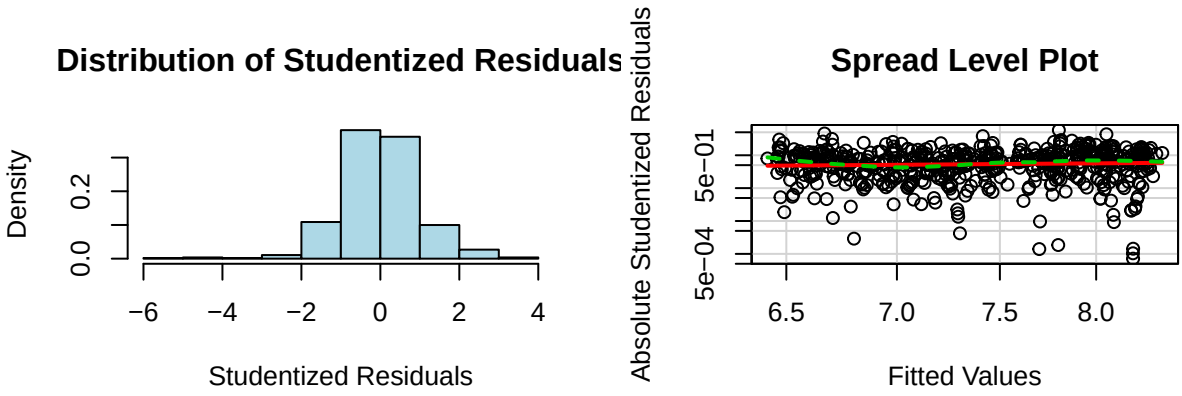
```
## lm(formula = imdb_rating ~ us_viewers_in_millions + sum_word_count +
##     number_in_series, data = episodes_subset)
##                         coef.est coef.se
## (Intercept)              8.58     0.14
## us_viewers_in_millions -0.01     0.01
## sum_word_count           0.00     0.00
## number_in_series         0.00     0.00
## ---
## n = 561, k = 4
## residual sd = 0.48, R-Squared = 0.57
```

FYI – I could've made a table using kable or xtable for summary(fit), however, I like the display output much more. Unfortunately, I'm not able to convert this into a table.

The fit of the model can be interpreted by the R^2 (explained variance) value, which is a decent 57%, however, the coefficients for these variables don't tell a meaningful story. The intercept at 8.58 is the average rating for 0 US Viewers, 0 word count, and 0 in the number of series. This isn't a useful prediction. This prediction, coupled with the coefficients close to 0, may not make this the best model. The number in series may not be the best predictor either, although it is a decent substitute for original air date. Regardless of this, let's run some regression diagnostics.

Regression Diagnostics:
Check for non-normality and non-constant error variance (heteroscedasticity)

```
##
## Suggested power transformation:  0.2388304
```




```
ncvTest(rating_fit_1)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.819929    Df = 1     p = 0.1773217
```

The distribution of studentized residuals looks roughly normal, approximately independently distributed with a mean of 0.

The non-constant variance score test has a p-value more than a significance level of 0.05, therefore we can accept the null hypothesis that the variance of the residuals is constant and infer that heteroscedasticity is not present, confirmed by graphical inference from the Spread Level Plot.

Let's replace number in series with count of unique locations in our model.

```
rating_fit_2 <- lm(imdb_rating ~ us_viewers_in_millions + sum_word_count + location_count, data=episodes

arm::display(rating_fit_2)
```

```
## lm(formula = imdb_rating ~ us_viewers_in_millions + sum_word_count +
##     location_count, data = episodes_subset)
##                         coef.est coef.se
## (Intercept)             6.61     0.10
## us_viewers_in_millions 0.07     0.00
## sum_word_count          0.00     0.00
## location_count          0.00     0.00
## ---
## n = 561, k = 4
## residual sd = 0.58, R-Squared = 0.37
```

The R^2 (explained variance) value has gone down to 37%, indicating that the fit of our first model was better. Perhaps we can log transform some of our variables.

```
log_us_viewers <- log(episodes_subset$us_viewers_in_millions)
log_sum_word <- log(episodes_subset$sum_word_count)
log_number_series <- log(episodes_subset$number_in_series)

rating_fit_3 <- lm(imdb_rating ~ log_us_viewers + log_sum_word + log_number_series, data=episodes_subse

arm::display(rating_fit_3)
```

```
## lm(formula = imdb_rating ~ log_us_viewers + log_sum_word + log_number_series,
##     data = episodes_subset)
##                   coef.est coef.se
## (Intercept)        6.27     0.87
## log_us_viewers     0.54     0.07
## log_sum_word       0.14     0.10
## log_number_series -0.23     0.04
## ---
## n = 561, k = 4
## residual sd = 0.55, R-Squared = 0.44
```

The R^2 (explained variance) value has gone up to 44%, better than our 2nd model, however, still not as good as our first. Our next step is to experiment with interactions.

```
rating_fit_4 <- lm(imdb_rating ~ log_us_viewers + log_sum_word + log_number_series, log_us_viewers:log_s

arm::display(rating_fit_4)
```

```
## lm(formula = imdb_rating ~ log_us_viewers + log_sum_word + log_number_series,
##     data = episodes_subset, subset = log_us_viewers:log_sum_word)
##                   coef.est coef.se
## (Intercept)       -15.87    37.05
## log_us_viewers      0.66     2.09
## log_sum_word        2.67     4.03
## log_number_series   0.44     0.54
## ---
## n = 5, k = 4
## residual sd = 0.34, R-Squared = 0.49
```

The R^2 (explained variance) value has gone up to 49%! This is great progress, but still isn't as good a fit as our initial model. We'll stick with our first linear regression model for the time being.

Logistic Regression: IMDb "Good" Rating Predicted by US Viewers, Sum of Word Count, and Number in Series

```
rating_fit_5 <- glm(rating ~ us_viewers_in_millions + sum_word_count + number_in_series, family=binomial

arm::display(rating_fit_5)
```

```
## glm(formula = rating ~ us_viewers_in_millions + sum_word_count +
##     number_in_series, family = binomial(link = "logit"), data = episodes_subset)
##                         coef.est coef.se
## (Intercept)              5.02     0.99
## us_viewers_in_millions -0.04     0.04
## sum_word_count          0.00     0.00
## number_in_series       -0.01     0.00
## ---
##   n = 561, k = 4
##   residual deviance = 467.9, null deviance = 651.5 (difference = 183.6)
```

```
invlogit(5.02)
```

```
## [1] 0.9934388
```

The intercept, 5.02, which turns into 0.99 represents the probability of getting a "good" rating with 0 viewers, sum of word count, and number in series, which doesn't really make sense. Basically, the baseline is that there's a 99% probability of getting a "good" rating. Another million viewers corresponds to a 1% (-0.04/4) lower probability of getting a good rating. If more people are watching, one would assume that it would be a good episode. However, if there are many more people watching and voting, there's more room for bad ratings in a sense.

Word count doesn't seem to make much of a difference, while as a new episode is released, there's a 0.25% lower probability of getting a good rating. We saw this trend earlier when we explored the trend of ratings throughout time.

How does the residual deviance change when you replace sum of word count with location count to the mix?

```
rating_fit_6 <- glm(rating ~ us_viewers_in_millions + number_in_series + location_count, family=binomial

arm::display(rating_fit_6)
```

```
## glm(formula = rating ~ us_viewers_in_millions + number_in_series +
##     location_count, family = binomial(link = "logit"), data = episodes_subset)
##                         coef.est coef.se
## (Intercept)              4.98     1.02
## us_viewers_in_millions -0.04     0.04
## number_in_series       -0.01     0.00
## location_count          0.01     0.02
## ---
##   n = 561, k = 4
##   residual deviance = 468.1, null deviance = 651.5 (difference = 183.4)
```

The residual deviance goes up slightly, but there's hardly a change. As Gelman states, the location count is most likely just "random noise." Although, we can tell that location count can positively affect rating, in a sense, vs. the other variables. We can consider a log transformation or interactions.

```
rating_fit_7 <- glm(rating ~ log_us_viewers + log_number_series + log_sum_word, family=binomial(link="l

arm::display(rating_fit_7)
```

```
## glm(formula = rating ~ log_us_viewers + log_number_series + log_sum_word,
##     family = binomial(link = "logit"), data = episodes_subset)
##                    coef.est coef.se
## (Intercept)       19.29     5.35
## log_us_viewers    -0.44     0.40
## log_number_series -3.67     0.54
## log_sum_word       0.49     0.44
## ---
##   n = 561, k = 4
##   residual deviance = 454.9, null deviance = 651.5 (difference = 196.5)
```

The residual deviance goes down by ~13! This is a better model. It seems like the only thing that can positively affect a rating through this model is the number of words used per episode. We do have a biased dataset in a way, since the number of viewers are skewed towards those who watch on TV, vs. streaming services.

Logistic Regression: Gender (1 for female, 0 for male) Predicted by Sum of Word Count, Episode Count, and Unique Location Count

```
gender_fit_1 <- glm(female ~ sum_word_count + episode_count + location_count, family=binomial(link="log

arm::display(gender_fit_1)
```

```
## glm(formula = female ~ sum_word_count + episode_count + location_count,
##     family = binomial(link = "logit"), data = character_subset)
##                coef.est coef.se
## (Intercept)    -1.20     0.14
## sum_word_count  0.00     0.00
## episode_count  -0.01     0.01
## location_count  0.01     0.01
## ---
##   n = 356, k = 4
##   residual deviance = 370.4, null deviance = 376.9 (difference = 6.4)
```

```
invlogit(-1.20)
```

```
## [1] 0.2314752
```

This is interesting, as I would have thought that the sum of the words spoken would have an effect on whether the character would be female or not. The intercept indicates that the probability of a character being female with 0 word count, episode count, and location count (which doesn't make sense, since the character nedes to be located somewhere), is 23%. This isn't a very useful prediction.

An extra word doesn't affect the probability of the character's gender being female. If the episode count goes up by 1, there's 0.25% lower probability that the character would be female, while if the location count goes up, there's a 0.25% higher probability that the character would be female. One thing that I could explore is the patterns of location count for female characters as time goes on. Perhaps female characters show up in a variety of locations more often vs. the male characters, even though they may not speak as much.

What happens when we log transform the word count?

```
log_sum_word_count <- log(character_subset$sum_word_count)

gender_fit_2 <- glm(female ~ log_sum_word_count + episode_count + location_count, family=binomial(link=

arm::display(gender_fit_2)
```

```
## glm(formula = female ~ log_sum_word_count + episode_count + location_count,
##     family = binomial(link = "logit"), data = character_subset)
##                    coef.est coef.se
## (Intercept)        -0.16     1.06
## log_sum_word_count -0.17     0.18
## episode_count       0.00     0.00
## location_count      0.00     0.00
## ---
##   n = 356, k = 4
##   residual deviance = 373.1, null deviance = 376.9 (difference = 3.8)
```

The residual deviance goes up slightly, which indicates our first gender-related model is better. How about when we log transform all variables?

```
log_episode_count <- log(character_subset$episode_count)
log_location_count <- log(character_subset$location_count)

gender_fit_3 <- glm(female ~ log_sum_word_count + log_episode_count + log_location_count, family=binomia

arm::display(gender_fit_3)
```

```
## glm(formula = female ~ log_sum_word_count + log_episode_count +
##     log_location_count, family = binomial(link = "logit"), data = character_subset)
##                     coef.est coef.se
## (Intercept)          0.47     0.92
## log_sum_word_count  -0.49     0.22
## log_episode_count   -0.25     0.16
## log_location_count   0.71     0.32
## ---
##   n = 356, k = 4
##   residual deviance = 369.9, null deviance = 376.9 (difference = 6.9)
```

The residual deviance went down, but is still quite close to our original model. All variables "log transformed" could just be considered "random noise," so we can stick with our original model. How about if we add a particular combination of words to the mix?

Logistic Regression: Gender (1 for female, 0 for male) Predicted by Word Count and Whether Particular Words are in the Line or Not

```
simpsons_data$words <- ifelse(simpsons_data$normalized_text%in%c("love","please","right","help","good",

gender_fit_4 <- glm(female ~ word_count + words, family=binomial(link="logit"), data=simpsons_data)

arm::display(gender_fit_4)
```

```
## glm(formula = female ~ word_count + words, family = binomial(link = "logit"),
##     data = simpsons_data)
##             coef.est coef.se
## (Intercept) -1.00     0.01
## word_count   0.00     0.00
## words        0.40     0.24
## ---
##   n = 111736, k = 3
##   residual deviance = 128121.5, null deviance = 128161.7 (difference = 40.3)
```

```
invlogit(-1)
```

```
## [1] 0.2689414
```

The words in "words" were drawn from wordclouds of Homer and Marge Simpson (found below) with a layer of "sentiment". This was definitely not the ideal way for me to implement this. I wasn't entirely sure how to create these wordclouds properly, but found a Kaggle kernel that did this for me.

The residual deviance is unbelievably high, implying that this model may not be the best. However, the combination of words coefficient indicates that there is 10% higher probability that the character speaking those lines would actually be female, which is quite interesting.

I'd like to repeat this, but by finding a particularly interesting combination of words, rather than just looking at Homer and Marge Simpson.

Conclusion

Let's revisit my initial questions...

1. How has The Simpsons performed over the years, rating and viewership wise?

The Simpsons' ratings and viewership has gone down over the years. Viewership has a steeper decline than that of ratings. Viewership in our data is in terms of those who watch The Simpsons on the TV, vs. streaming TV, which is why this makes sense; there have been many cord-cutters since the rise of streaming TV.

2. How does The Simpsons compare to ratings and viewers of Family Guy?

Additional viewership has a stronger effect on Family Guy over The Simpsons. However, There are definitely more people watching The Simpsons. We did not have a temporal variable in our model for this analysis. A next step could be how the viewers and ratings have changed over time for each show.

3. Is there a particular combination of things that can increase the IMDb rating, i.e. more words, more unique locations in an episode?

In our linear regression model, we only found that additional US viewers would have a small but negative effect, while more words and more unique locations had no effect.

In our logistic regression model, we found similar conclusions. US viewers would, again, have a small but negative effect, as would a new episode. We also noted that the count of unique location would have a small but positive effect on the rating. Once we log transformed our variables, the sum of spoken word positively influenced a "good" rating as well.

4. Can you predict the gender of the character based on the total number of words spoken, total episodes they've been featured on, unique locations they've visited while on the show, in addition to a combination of specific words spoken?

I had thought that the sum of words spoken would influence the predicted gender, but this wasn't the case unless we log transformed our variables. Without the log transformation, we notice that if the episode count goes up by 1, there's a 0.25% lower probability that the character would be female, while if the location count goes up, there's a 0.25% higher probability that the character would be female.

The logistic regression I created that predicted a character's gender based on word count and a combination of particular words wasn't the best model, since the residual deviance was very high. However, the combination of words that I did choose showed that there would be a 10% higher probability that the character speaking those lines would actually be female.

Moving forward, I could build a model predicting ratings of episodes using a combination of words that have positive sentiment.

Winning Models with Multiple Predictors

Here are the models that I stated were the "best," after exploring transformations and interactions.

```r
# Linear Regression: IMDb Rating Prediction
winning_fit_1 <- lm(imdb_rating ~ us_viewers_in_millions + sum_word_count + number_in_series, data=episo

# Logistic Regression: Rating Prediction
winning_fit_2 <- glm(rating ~ log_us_viewers + log_number_series + log_sum_word, family=binomial(link="

# Logistic Regression: Character's Gender Prediction
winning_fit_3 <- glm(female ~ sum_word_count + episode_count + location_count, family=binomial(link="log
```

Experimentation with WordClouds and Sentiments