Analyzing IMDb Ratings of Scooby-Doo Episodes

Regression Rockstars: James Cai, Steph Reinke, Sarah Wu, Michael Zhou

2023-12-01

Introduction and Data:

Introduction

Scooby-Doo is a popular animated TV show that follows a group of teenagers and a talking Great Dane, Scooby-Doo, as they solve mysteries involving supernatural monsters and creatures. Each episode typically involves seeking and scheming to find the villain, ending with a dramatic unmasking of the monster. The show focuses on themes of friendship and teamwork. The show originally aired on CBS from 1969 - 1976, but there has been many subseries and reboots since.

We are interested in researching Scoody-Doo IMDb ratings because we all enjoyed Scooby-Doo in our childhoods. We also think that finding certain predictors of animated TV series ratings is useful for the entertainment industry. Specifically, our findings could be useful to anyone looking to create an animated TV series and wanting to know what aspects make up a successful episode. In the paper, "Determining and Evaluating The Most Popular Cartoons Among Children Between 4 and 6 Years of Age" published in 2017, the authors criticize the use of violence, vulgar language, and horror music in Scooby-Doo (Başal et. al. 2017), yet we can't ignore the huge impact and popularity of Scooby-Doo. In 2013, Scooby-Doo was ranked the fifth greatest cartoon of all time (TVGuide 2013). If Scooby-Doo continues to create spin-off shows, our findings about what makes a successful episode could inform their future episodes as well.

Our primary research question is what factors best explain the variability in the IMDb ratings of Scooby-Doo episodes? In other words, what elements tend to contribute to a successful episode? We want to investigate how various predictor variables in the dataset like `monster.amount`, `engagement`, character that unmasks the villain (`unmask.fred`, and such for all five characters part of the main group), `network`, and more, adequately explain the variability in IMDb ratings. We hypothesize that episodes with a higher monster count will have a better rating, since we think that there is more action and suspense in episodes with more monsters. We also hypothesize that the higher engagement, the worse the rating will be. This is because we think that people are more likely to write a review online when they dislike the episode rather than if they liked the episode. We think that episodes where Fred unmasked the villain will have a higher rating since he is the leader of the group and thus, we think that people will be more drawn to him. Finally, we think that episodes that aired on Cartoon Network will have a better rating, since we think that Cartoon Network has the ability to generate more positive responses since they specialize in cartoons and are pretty well-known. In our analysis, we would like to explore the interaction between these variables as well as consider other predictors in the dataset.

Data

This Scooby-Doo data was found on the TidyTuesday database on Github. The data originally comes from Kaggle and was manually aggregated by user Plummye in 2021. The curator took roughly one year to watch every Scooby-Doo iteration and track every variable in this dataset. It is noted that some of the values are subjective by nature of watching, but the original data data curator tried to keep the data collection consistent across the different episodes.

Each observation represents an episode from a rendition of the Scooby-Doo franchise up until February 25, 2021, including movies and specials. The variables that were measured include the series and episode name (which we will not use as predictor variables), network aired on, IMDb rating, engagement (represented by number of reviews on IMDb), and many details about what happened in each episode, such as how many monsters appeared, which character captured and unmasked the monster, the terrain of the episode, and more. There is a mix of both numerical and categorical characteristics.

The unmask variable is in the data as 6 separate columns with each column representing a person, such as `unmask.fred`, `unmask.velma`, etc. Before any of our analysis, we combined these columns into one singular column, `unmask_villain`. We also converted `imdb` from a `character` to a `double` as we want it to be a quantitative value.

Our response variable is `imdb`, while after careful consideration, our predictor variables are `unmask_villain`, `monster.amount`, and `network`.

`imdb`: double, represents the rating on IMDb

`unmask_villain`: character, represents which character unmasked the villain (if any)

`monster.amount`: double, represents the number of monsters in the episode

`network`: character, represents the network the episode was aired on

Exploratory data analysis

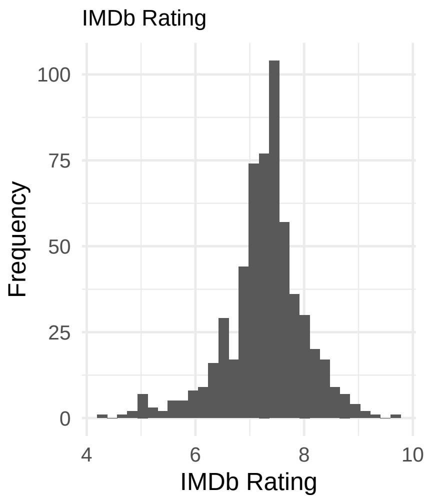


Figure 1

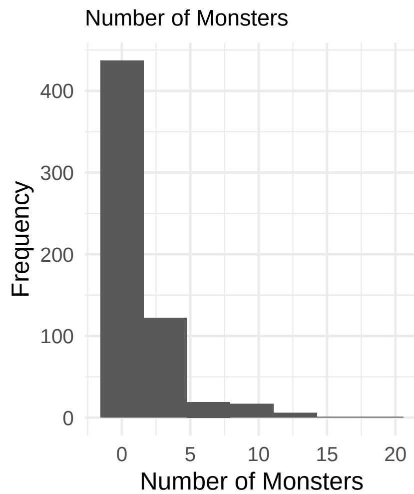


Figure 2

Figure 1: The distribution of our response variable IMDb ratings, `imdb`, is unimodal and roughly symmetrical. The mean is 7.278 and the standard deviation is 0.732. The minimum is 4.2 and the maximum is 9.6. There does not seem to be any significant outliers.

Figure 2: The distribution of the number of monsters, `monster.amount`, is unimodal and right skewed. The median is 1 monster and the IQR is 1 monster. The minimum is 0 monsters and the maximum is 19 monsters. There are a few episodes with notably high amounts of monsters, with 5 episodes having 13 or more monsters.

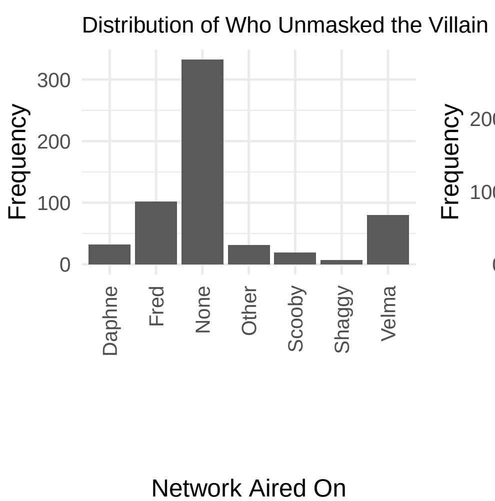


Figure 3

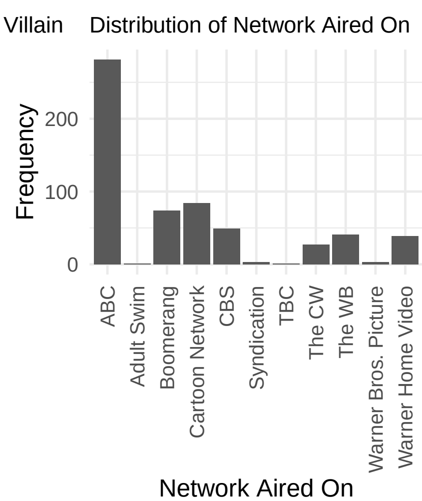


Figure 4

Figure 3: The distribution of who unmasks the villain, `unmask_villain`, shows that in a good majority of the episodes, no one unmasked the villain. However, out of the episodes where a villain was unmasked, Fred and Velma were the main characters that unmasked the villain.

Figure 4: The distribution of the network the episode aired on, `network`, shows that a good majority of the episodes aired on ABC. There were also considerable amounts of episodes that aired on Cartoon Network and Boomerang, while there are also networks that aired very few episodes, such as TBC and Adult Swim.

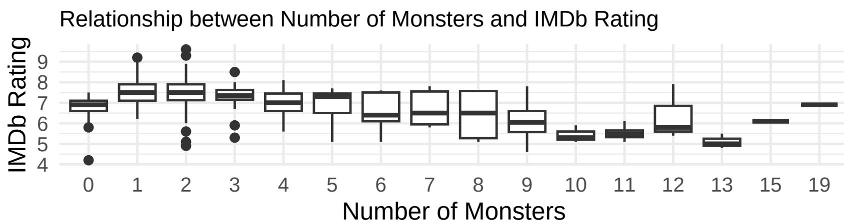


Figure 5

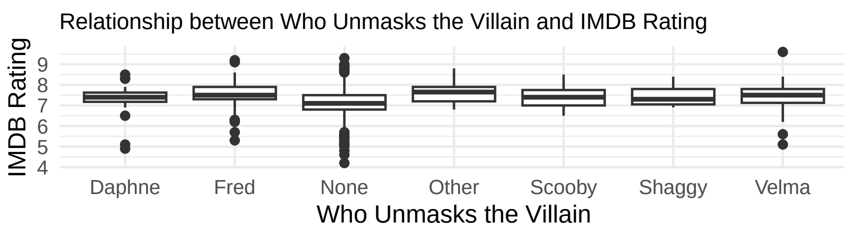


Figure 6

Figure 5: From the distribution of the different boxplots for each number of total monsters included in `monster.amount`, we observe that many of the interquartile intervals of the boxplots overlap, meaning that their IMDb ratings are quite similar. However, there are still quite a few boxplots that do not overlap with a couple of the other boxplots, signaling that `monster.amount` may have some significant effects.

Figure 6: From the distribution of the different boxplots for each character included in `unmask_villain`, we observe that many of the interquartile intervals of the boxplots overlap, meaning that their IMDb ratings are quite similar. Since they all overlap, we should consider whether this variable is important for our model.

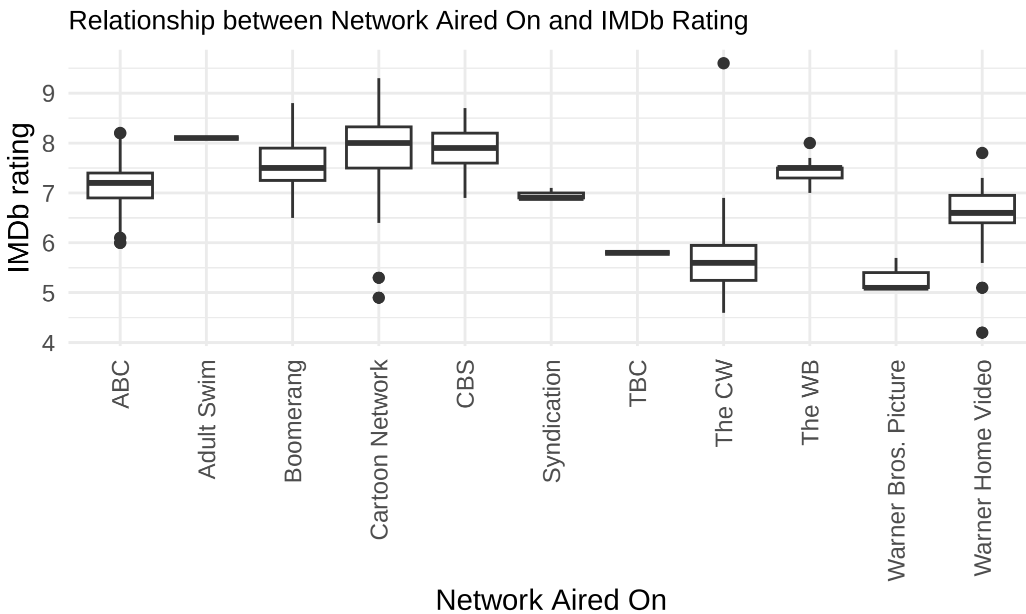


Figure 5

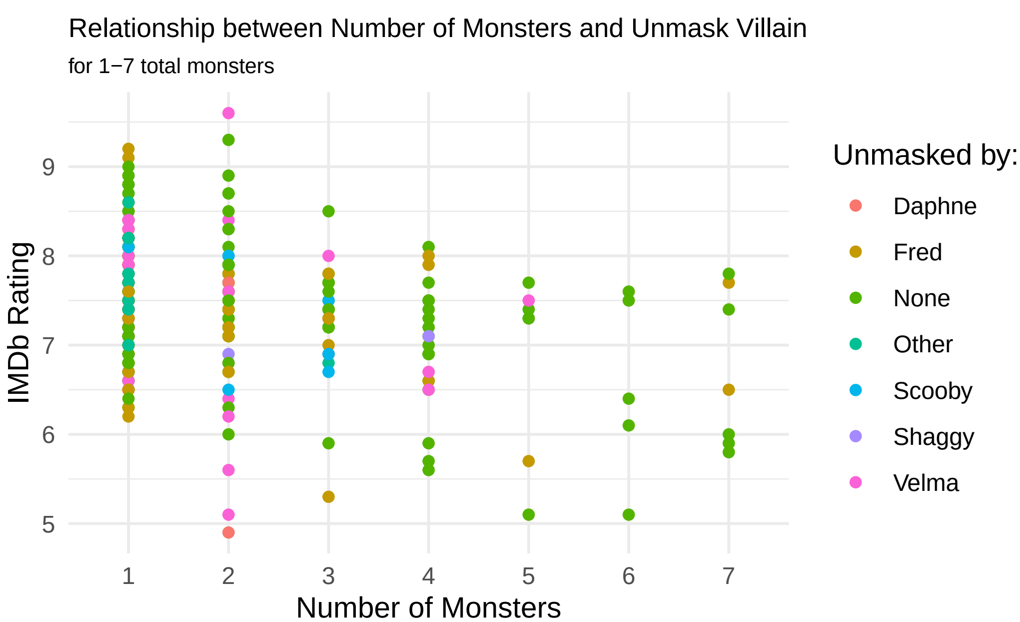


Figure 8

Figure 7: From the distribution of the different boxplots for each network, we observe that many of the interquartile intervals of the boxplots overlap, meaning that their IMDb ratings are quite similar. It seems that Cartoon Network generally received the best ratings, while Warner Bros. Picture and The CW generally received the worst ratings. We also observe a few

outliers in the distribution of IMDb ratings for some networks, such as Warner Home Video and ABC. Many of the networks have IMDb ratings that are pretty symmetrical, as the line representing the median is close to the middle of the box, such as in the case of CBS and The CW, but some are pretty skewed, such as in the case of Syndication, The WB, and Warner Bros. Picture.

Figure 8: Here, we explore the relationship between two of our predictor variables, `monster.amount` and `unmask_villain`. We are only showing a subset of the full relationship between the two variables, since as number of monsters increases, there is less and less data to observe a relationship. From this graph, we become interested in whether there is a significant relationship between these two variables on IMDb rating–for example, if there's one monster, is the effect on IMDb rating different for whether Fred unmasks the villain or if Velma unmasks the villain?

Methodology:

Since our response variable is quantitative, we decided to use multiple linear regression for modeling. We also split our original data set into a training (75%) and testing (25%) set to attempt to prevent model overfitting.

Given that our response variable is the IMDb rating of each episode, it is natural to choose predictor variables that have variability for each episode, meaning the predictors should characterize each episode. In this specific case, the plot of Scooby Doo can be broadly summarized as a team of friends trying to unmask the villain. Thus it is natural to choose variables such as monster.amount, unmask_villain. We also include network in the list of predictors because there is decent variability for this variable, and intuitively, the network an episode takes place in could affect the IMDb rating because there could be slight stylistic differences in each series for different networks. Additionally, we finalized on the listed variables because there were many variables that were difficult to use due to the nature of the input, such as monster.type, where episodes with multiple monsters had all the types listed as one character string with a comma separating the different types. Other variables did not make sense to use, such as title and culprit.name, as we do not think variables like these would make good predictors.Therefore, taking all of these factors into consideration, the predictor variables we settled on were `network`, `monster.amount`, and `unmask_villain`.

From our initial exploratory data analysis, we saw a lot of variation within the relationship between network and IMDb, so we definitely wanted to include `network` as part of our model. The relationships between `monster.amount` and `unmask_villain` each with IMDb seemed less strong, but since we were still interested in how these predictor variables affected IMDb as well as any interaction effects that could be made within the three variables, we included these two variables in our model as well. To compare our models, we plan on using 3-fold cross validation, and we also included a function to calculate adjusted R squared, AIC, and BIC, which we will use when choosing our final model.

For each model, we decided to create a recipe, where we performed these steps across all models:

1. Simplified the number of networks in `networks` by using step_other with a threshold of 30 in hopes of ending with a more parsimonious model.
2. Created dummy variables for all nominal predictors using step_dummy, allowing us to use linear regression for the different levels in each nominal predictor.
3. Removed predictors with zero variance using step_zv so that our model is only including variables with variability throughout the dataset.

We tested a total of four different models. Since we knew that we wanted to include the variables, `network`, `monster.amount`, and `unmask_villain`, we tested different interactions between combinations of two of the three predictor variables.

Model 1: imdb ~ network + monster.amount + unmask_villain with no interaction terms

Model 2: imdb ~ network + monster.amount + unmask_villain with interaction between monster.amount and unmask_villain

Model 3: imdb ~ network + monster.amount + unmask_villain with interaction between monster.amount and network

Model 4: imdb ~ network + monster.amount + unmask_villain with interaction between unmask_villain and network

Here are our results from our 3-fold cross validation:

Model #	Mean Adjusted R-Squared	Mean AIC	Mean BIC
1	0.269	610.982	676.177
2	0.333	580.208	633.100
3	0.265	606.73	649.782
4	0.303	603.144	694.162

From these statistics, it is clear that Model #2 (imdb ~ network + monster.amount + unmask_villain with interaction between monster.amount and unmask_villain) had the highest mean adjusted R-squared and the lowest AIC and BIC values. Therefore, we will use this model as our final model.

Results:

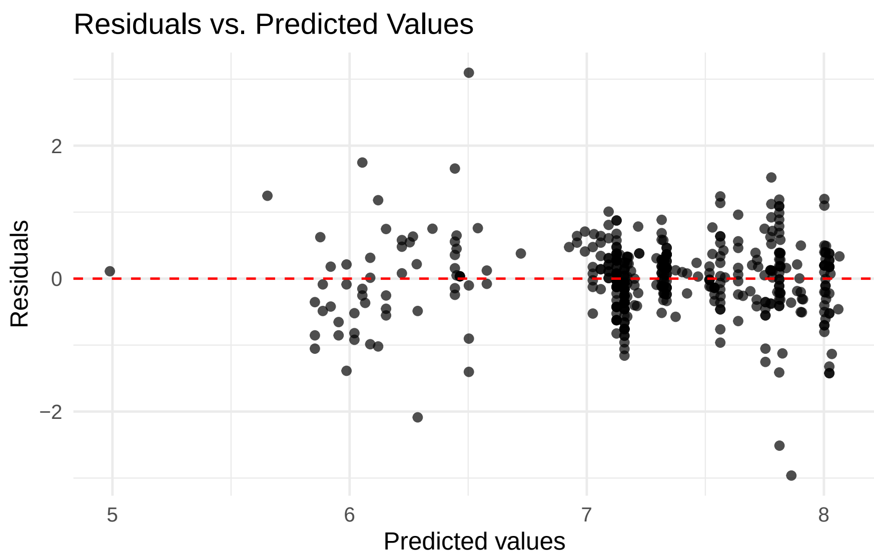
Here is the output from our final model:

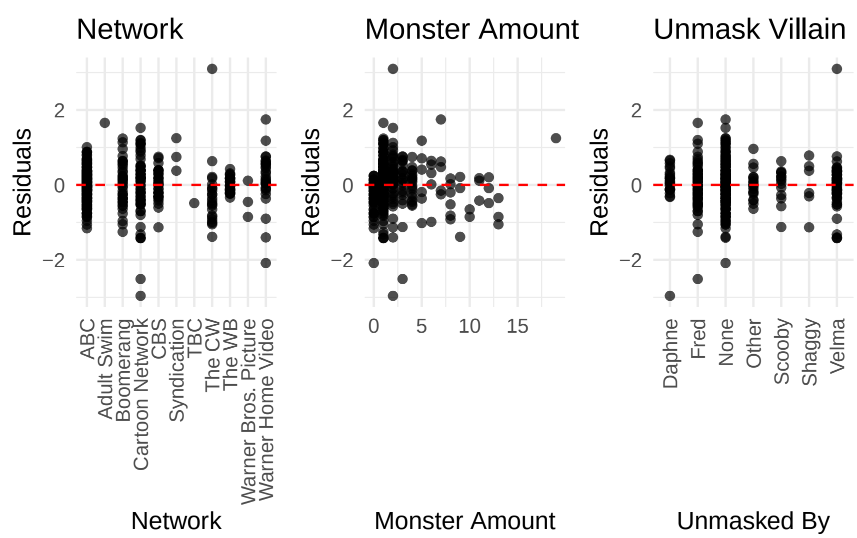
term	estimate	std.error	statistic	p.value
(Intercept)	7.469	0.144	51.764	0.000
monster.amount	-0.146	0.057	-2.560	0.011
monster.amount_x_unmask_villainFred	0.052	0.084	0.615	0.539
monster.amount_x_unmask_villainNone	0.113	0.058	1.949	0.052
monster.amount_x_unmask_villainOther	0.152	0.077	1.955	0.051
monster.amount_x_unmask_villainScooby	0.130	0.177	0.735	0.463
monster.amount_x_unmask_villainShaggy	0.272	0.218	1.248	0.213
monster.amount_x_unmask_villainVelma	0.184	0.085	2.171	0.031
network_Boomerang	0.438	0.102	4.274	0.000
network_Cartoon.Network	0.686	0.082	8.390	0.000
network_CBS	0.691	0.107	6.471	0.000
network_The.WB	0.202	0.115	1.756	0.080
network_other	-0.872	0.103	-8.439	0.000
unmask_villain_Fred	-0.059	0.187	-0.315	0.753
unmask_villain_None	-0.310	0.151	-2.057	0.040
unmask_villain_Other	-0.273	0.224	-1.220	0.223
unmask_villain_Scooby	-0.282	0.327	-0.863	0.389
unmask_villain_Shaggy	-0.378	0.446	-0.848	0.397
unmask_villain_Velma	-0.170	0.185	-0.921	0.358

This is the model output for our final model, where we predicted `imdb` with `network`+ `monster.amount` + `unmask_villain` with an interaction between `monster.amount` and `unmask_villain`. From the p-values of the coefficients, we see that 8 of 18 predictor variables have significant values when using a threshold of 0.05. Most of the low p-values in coefficients come from `network`. It is interesting that `monster.amount_x_unmask_villainVelma` was significant while none of the other interactions were. Another significant coefficient value that is worthy to note is `unmask_villain_None`. From our earlier EDA, we did see that 'None' was the largest category for `unmask_villain`, so it is possible the coefficient value is significant only because there was a large amount of data for it.

[insert coefficient interpretations^]

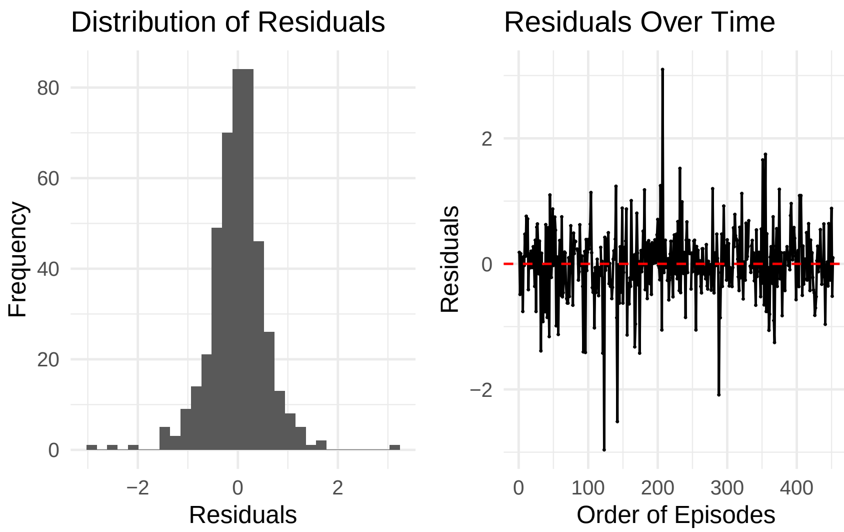
We proceed by checking model conditions for inference. Specifically, this includes linearity, constant variance, normality, and independence.





From the residuals vs. predicted values plot as well as the three plots of residuals vs. each of our three predictor variables, we observe that there does not seem to be a discernible, non-linear pattern in each of these plots. Therefore, the linearity condition is satisfied.

From the residuals vs. predicted values plot, the vertical spread seems to be approximately constant across the x-axis. Though there are definitely outliers in this plot, it is roughly the same for above and below the line. Therefore, the constant variance condition is satisfied.



From the histogram showing the distribution of residuals, we observe that the distribution of the residuals is approximately unimodal and symmetric. The sample size of 452 observations in our training data is also sufficiently large to relax this condition if it was not satisfied. Therefore, the normality condition is satisfied.

Since our data is collected over time, we examined a scatterplot of the residuals versus order in which the data were collected. However, no clear pattern was observed in the residuals vs. order of data collection plot. Therefore, the independence condition is satisfied.

Since all four conditions are satisfied, we continue with using our model for predictions in both our training and testing sets.

Dataset	R-Squared	RMSE
Training	0.483	0.554
Testing	0.506	0.423

The model's R-squared value on the training data is approximately 0.483, which means that approximately 48.3 percent of the variation in the response variable explained by our regression model. However, the model's R-squared value on the testing data is approximately 0.506, which is actually surprisingly higher than the value for our training data. Since the R-squared values are relatively close, we are not too concerned about our model having overfit the data. The same trend is observed in our RMSE values, as the RMSE for the training data is 0.554, while it is 0.423 for the testing data.

Discussion and Conclusion:

All in all, from our model, we see that `network` and `monster.amount` seem to be significant when predicting IMDb ratings for different Scooby-Doo episodes. We observe that the variable, `unmask_villain`, as well as the interaction of it with `monster.amount` carry very few significant coefficients, so we may reconsider including this variable as a predictor variable in future models. However, for this project, since we were especially interested in using `unmask_villain` as a predictor variable, we retain it in our final model.