Gaining Insight from Online Customer Reviews:

A Non-traditional Analysis for a Very Traditional Product

Jacey Planteen
Kennesaw State University
STAT 8940: Applied Analysis
Summer 2016

Table of Contents

Executive Summary

In today's age of smartphones, social media, and online connectivity, an ever-growing list of products can be purchased online, which now includes lead-acid batteries. These very traditional products can be purchased from online retailers like Amazon.com, where customers can freely rate and review their new items. For now, most people who have a dead car battery are not going to order a replacement online and wait patiently for delivery; however, these people may consult online reviews as they make purchasing decisions in-store. Additionally, battery manufacturers may be able to gain valuable insight into customer preferences and opinions on their products as well as their competitors' from online reviews.

This study sought to determine whether meaningful insights could be gathered from data readily available on Amazon for three different battery brands: Exide, Optima, and Odyssey. Two different groups of data were extracted from Amazon and were analyzed with different techniques:

- Sales rank, price, number of customer reviews, and average reviews rating were tracked for 108 different products for a period of two weeks. Ordinary least squares (OLS) regression was used to understand the relationship of brand, price, and customer reviews on sales rank, which was used as a surrogate measure for sales volume.
- Individual customer reviews for the 108 battery types were also gathered. Term frequencies for individual words and two word phrases were used to identify key terms of interest for further analysis. Various categorical techniques were used to analyze relationships in the reviews, including the Pearson chi-square test for independence and log linear models for contingency tables.

Many insights were indeed gained from this research. Some of the key findings included:

- The average customer review rating for a given battery type has the most impact on sales volume for products with few customer reviews.
- More Amazon users find negative customer reviews to be helpful than positive reviews, making minimization of negative feedback all the more important.
- Exide branded products had more negative customer reviews than the other two brands. Negative reviews have high occurrences of words like "warranty" and "dead". Additionally, shorter periods of time are mentioned in Exide reviews versus reviews for the other brands. These findings suggest that Exide batteries may have shorter battery life or more premature battery failures than the competition.
- Toyota Prius batteries are relatively popular on Amazon.com, and this market segment appears to be dominated by Optima today.

This study did reveal that online retailer websites can provide a wealth of information to manufacturers. Approaches of this type could provide a direct customer feedback pipeline to understand not only their own products' performance but also that of other brands.

Introduction

Assessing customer satisfaction and needs can be an extremely challenging and costly task for manufacturers. The explosion of online business has provided a potentially rich customer feedback pipeline of uncensored product reviews to the people who design, engineer, and market these products. Additionally, customer reviews give manufacturers the ability to gauge not only the reception of their own products but also their competitors'.

As consumers conduct more and more business online, the range of goods available for purchase over the web continues to expand as well. Even products traditionally only purchased at brick-and-mortar stores have found their way online, from perishable groceries to used cars. This research sought to evaluate customer satisfaction and feedback for a product not traditionally sold online: valve-regulated lead-acid (VRLA) batteries.

The lead-acid battery industry has been around for over a century and does not have a reputation for being on the cutting edge of technology. However, even this industry must adapt to the ever-changing needs and preferences of the almighty consumer. A direct customer feedback channel, as through online product reviews, could accelerate the evolution of product design and marketing.

This research sought to evaluate the importance of customer reviews and opinion on battery sales as well as the utility of information contained in free-form customer reviews. Three major brands of batteries sold at the online behemoth Amazon were evaluated in this study. Is it possible to extract meaningful and actionable insights by analyzing online product information and customer reviews on Amazon?

Background

It is doubtful purchasing lead-acid car batteries from Amazon and other online retailers will become a mainstream method in the near future. First, only premium batteries are being offered through this direct ship method. Lead-acid batteries are a hazardous commodity. Only valve regulated lead-acid (VRLA) batteries are considered non-spillable; dangerous sulfuric acid may escape from the standard lead-acid battery if it is not kept upright. The premium VRLA batteries that are available online come at a much higher price point, and thus consumers not willing to pay that premium will continue to purchase their batteries elsewhere.

Secondly, a stranded motorist with a dead battery is not going to order online and then wait a couple of days to receive a replacement battery. A dead battery is generally a problem which requires immediate resolution, and that resolution will mainly happen in a traditional manner – through brick-and-mortar stores – for the time being. If business is being conducted offline, do online product reviews and ratings even matter?

In a 2007 survey, online reviews were accessed by 24% of Internet users prior to purchasing a service delivered offline (Zhu & Zhang, 2012). More recently, a 2012 survey found that over half (52%) of shoppers in the United States consider online reviews from a retailer website to be one of the three most important information sources for purchasing decisions – an increase over 2010 survey results where only 44% of participants found these to be an important source. Advice from store employees (12%) and in-store product packaging and displays (20%) did not rank nearly as well, and these numbers had decreased from the 2010 survey results (Fretwell et al., 2013). Furthermore, online consumers

generally trust online reviews. A 2012 Nielsen study found that 70% of online consumers said they trusted online product reviews (Floyd et al., 2014).

Thus, though most lead-acid battery purchases may continue to be conducted offline, customers may use digital sources of information, such as online product reviews, to decide what battery to buy and where to buy it from. Even if the revenue generated from online retailers such as Amazon is small for manufacturers, the potentially far-reaching influence of their products' ratings and reviews should not be ignored. It is both possible and feasible for customers to access online product reviews from their mobile devices as they browse product offerings in stores.

Beyond the possible impact of product ratings and reviews on sales, online product reviews may be more directly beneficial to manufacturers by providing insight into customer opinions on various product features. Traditionally, this kind of feedback has been captured through surveys, either online or with paper-and-pencil, which can be time consuming and expensive. Various factors such as questionnaire design and the willingness of participants can greatly impact the resulting data quality (Decker & Trusov, 2010). Thus, extraction of meaningful data from online product reviews is an appealing possibility.

Sample Information

Data Collection Process

All data for this research were acquired by scraping Amazon.com. Three different VRLA battery brands were chosen for this analysis: Optima, Odyssey, and Exide. All available product offerings for these three brands were originally gathered from search results. Non-battery items, such as testing equipment, chargers, and cables, were removed. Batteries which were only available through Amazon marketplace were also removed.

Various information including price, average customer review rating, volume of customer reviews, and each product's sales rank in the Automotive Replacement Parts: Batteries category were extracted for each battery type. The collection of this information was conducted over a period of 15 days, from June 15 to June 29, 2016. Information was scraped every other day for each battery type during this time period. Additionally, all customer product reviews and associated information were extracted once during this time period. Thus, ultimately two data sets were created: one longitudinal product information data set and one customer reviews data set. The extraction of all data from Amazon was conducted by scraping the HTML source code. This task was completed in R, and the associated code and functions can be found in Appendix B.

Longitudinal Product Information Sample

As previously mentioned, this data set contained information for 108 different battery types at 8 different collection time points for a total of 864 observations. The response variable in this study was the product's sales rank in the Automotive Replacement Parts: Batteries category. This sales rank was used as a surrogate measure for sales volume since that information was not available for public use. A lower sales rank number is indicative of higher sales volume. Explanatory variables of interest in this study included brand name, average customer review rating (on a scale from 1 to 5), and volume of

customer reviews. Price was also included in the analysis, even though it was not one of the predictors of interest. A summary of the product information data by brand is given below in Table I.

Table I. Summary Statistics of Product Information by Brand

	Exide	Odyssey	Optima
N Products	19	40	49
Mean Sales Rank	1,059	1,318	1,263
Min Sales Rank	32	35	5
Max Sales Rank	4,176	5,855	6,697
Mean Price	$198.38	$224.92	$191.42
Mean Review Volume	12.6	14.6	73.6
Mean Review Rating	4.23	4.33	4.59

As Table I shows, there were some notable differences between the brands. The number of products available for the Exide brand was about half of that for either Odyssey or Optima. Optima had both the lowest and highest sales rank products and also appeared to have substantially more product reviews (mean of 73.6 reviews per product) than the other two brands (means of 12.6 and 14.6 reviews). Optima also had the highest average customer review rating at a mean of 4.59 versus Exide's mean of 4.23 and Odyssey's of 4.33.

Examination of patterns in sales rank over time for individual batteries revealed some intriguing patterns. Sales ranks for two groups of select individuals are shown in Figures 1 and 2. While the sales ranks for the individuals in Figure 1 appeared to be relatively stable over the study period, all of the individuals in Figure 2 had a characteristic sharp drop in sales rank, followed by a gradual rise in rank. Though the full mechanism behind this drop is unknown, a likely explanation is that a sale was made for these product types just prior to the drop in rank. These large fluctuations would suggest that, in general, batteries are low volume products on Amazon except for perhaps a small subset of the lowest ranking (highest volume) battery types.

Figure 1. Sales Rank over Time for Select Stable Individuals

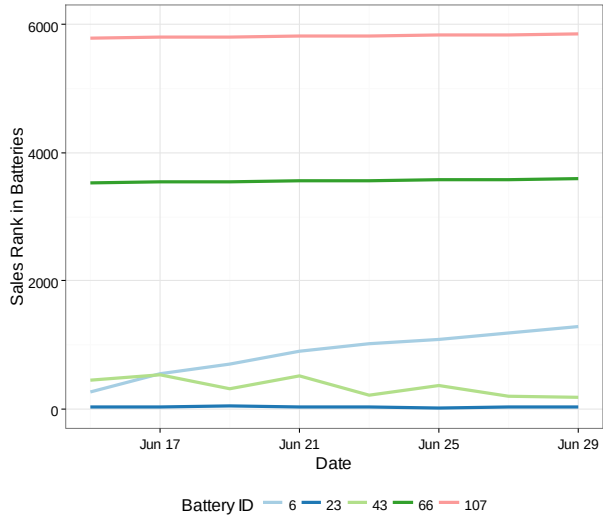
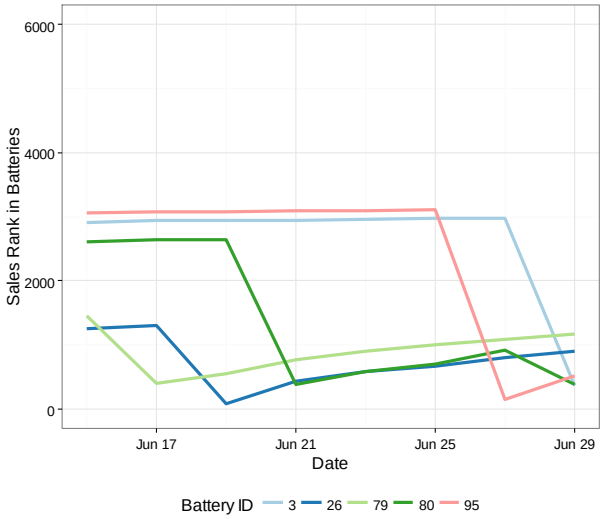


Figure 2. Sales Rank over Time for Select Variable Individuals

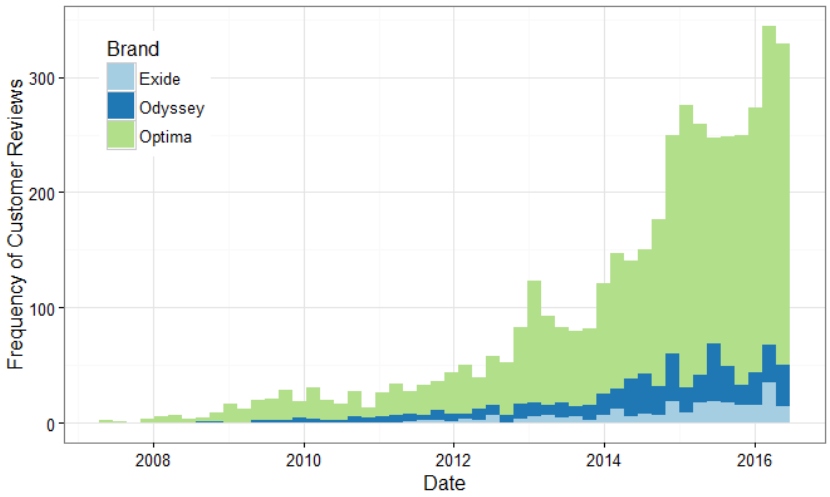


Customer Reviews Sample

Only 73 of the 108 battery types sampled had any customer reviews at the time of this study. The complete text of all customer reviews for those batteries was collected, yielding a total of 4,409 customer reviews. Other information that was collected with the review text included the actual review rating (from 1 to 5), the review title, the review date, the number of people who found the review helpful, and whether the purchase was verified or not.

The earliest of the gathered customer reviews dated back to June 2007. Figure 3 below shows the distribution of these reviews over time across the three brands. The volume of customer feedback has increased greatly over time for all three brands, but particularly for Optima.

Figure 3. Frequency of Customer Reviews over Time



It was questionable whether all the customer feedback was still relevant, with some reviews dating back nine years. Thus, it was decided to focus on feedback from only 2013 to present – roughly three and a half years of reviews. Due to the pattern of increasing feedback over time, this subset actually captured 82.9% of all reviews. The breakdown of the selected customer reviews by brand and rating is given in Table II below.

Table II. Customer Reviews by Rating and Brand

Rating	Exide	Odyssey	Optima	*Total*
1 star	31	33	215	*279*
2 stars	6	13	78	*97*
3 stars	8	17	84	*109*
4 stars	24	41	250	*315*
5 stars	149	363	2,345	*2,857*
Total	*218*	*467*	*2,972*	*3,657*

The totals in Table II reveal that most of the reviews have been positive since 2013, with 78.1% of reviews giving a 5 star rating. Optima product reviews dominate the feedback, accounting for 81.3% of the total reviews.

Methodology

Different analyses were conducted on the two data sets which were gathered off Amazon. All analyses were conducted in R, and Appendices C and D have the associated analysis code. A significance level of 0.05 was used throughout unless otherwise noted.

Sales Volume Analysis

The original intent with this data was to conduct a longitudinal analysis of sales volume via the observed sales ranks for each product and investigate the impact of branding and customer reviews over time. Ultimately, this approach was found to be inappropriate. Sales ranks were found to be highly variable and prone to drastic changes, as Figure 2 above shows. Additionally, the study period of only two weeks was likely not sufficient to capture meaningful trends over time. Thus, the longitudinal dataset was collapsed into a cross-sectional sample, with only one observation per battery type. Mean values for sales rank, customer reviews quantity, customer review rating, and price for each type were used to provide more robust estimates.

The cross-sectional data had only 108 observations, a relatively small sample size. Even so, the raw data were randomly split into training and test data sets for model development and validation respectively. Due to the limited amount of data, only 25% of the observations were reserved for model validation. The predictor variables of interest in this study included price, customer reviews rating, volume of customer reviews, and brand. Including both reviews rating and reviews volume presented a unique challenge as 35 of the 108 battery types had no customer reviews at the time of this study. Consequently, these 35 types did not have a customer reviews rating. Rather than exclude these observations, the customer review rating was set to zero. Additionally, an indicator variable for observations that had not yet been rated was included to capture any unique behavior related to this subset of the population.

Models were developed using ordinary least squares (OLS) regression. Up to three-way interactions were considered between the variables of interest. The response variable, sales rank, is by its nature a positive integer value. Thus, it would be possible for models to predict negative sales ranks, especially for observations with low fitted values. A natural logarithm transformation was found to be appropriate to resolve these issues and to better meet the assumption of a normally distributed response. Transformations of some of the predictor variables, particularly the reviews volume, were also found to be necessary from examination of partial regression plots and residuals analysis. The fourth root of the reviews volume was found to be more linearly related to the response than the untransformed values.

Another key assumption for OLS regression is that the observations are independent of each other. Sales rank arguably violates this assumption because if Battery A has a sales rank of 1, none of the other batteries can have a sales rank of 1. However, since sales ranks were an average of eight different observations (and thus batteries could have the same mean sales rank) and because the 108 batteries ranged in sales ranks from 5 to over 6,000 (relatively disperse), it was deemed that the independence assumption was reasonable.

The mean structure for the models was determined by first fitting the maximal model, including all main effects and up to three-way interactions between variables. Model simplification was

conducted using a backwards-selection type approach. The adequacy of the simplified model versus the maximal model was assessed using a likelihood ratio test. Variance inflation factors (VIFs) were reviewed to ensure minimal multi-collinearity existed; VIFs under 10 were considered to be acceptable. Validity of the final models were assessed by conducting residuals analysis, reviewing influence diagnostics, and lastly, by evaluating model performance with the reserved test data.

Two models were ultimately developed: one including customer reviews volume as a predictor and one excluding this variable. Products with low sales rankings inherently have had more customers purchase these batteries, and thus there have been more opportunities for customers to review that product. It would be expected for sales volume and customer reviews volume to be at least somewhat related, and it would be impossible from this cross-sectional study to determine if customer reviews volume actually caused more sales. This was the reasoning behind building a model without the reviews volume. All code for this portion of the analysis can be found in Appendix C.

Customer Reviews Analysis

High-level analysis was conducted on the customer reviews sample to understand how different variables were related to each other. Tests for association between brand name and review rating (1 to 5 stars) were conducted. Both the Pearson chi-square (X^2) test and log likelihood test (G^2) for independence were conducted; the expected cell counts were large enough (greater than 5) to meet the assumption for a chi-square distributed test statistic. The review rating is an ordinal value, and these tests for independence did not take this ordinal nature into account. Ignoring the ordinal nature of the rating may have decreased the test power; however, the power was sufficient for the purposes of this analysis. The relationship between review rating and the number of users who found the review to be helpful was also considered. The existence of a trend between rating and the helpfulness of the review was evaluated using Spearman's correlation and Kendall's tau. These particular tests were used to check for a general trend, not necessarily a linear trend. Spearman's correlation uses ranks instead of the raw values; Kendall's tau examines the number of concordant and discordant pairs of points.

The bulk of the textual analysis, however, was related to the raw text of the user reviews. The goal was to extract meaningful insights from the customer review text. Customer reviews are a blank slate, and these may be riddled with misspelled words, poor grammar, slang, and sarcasm. While a human reader is generally able to understand the overall message of the review, objective en masse analysis of all 3,657 reviews required significant preparation of the text for analysis.

Some of the basic text cleaning operations included removing punctuation and extra white space and converting all text to lower case. Typically numbers would also be stripped from customer reviews, but numbers were an important in this analysis. Many customers provided feedback on how long their battery (or previous batteries) had lasted: one month, one year, seven years, and so forth. The longevity of batteries was a characteristic of interest in this study, and thus integer values from one to twelve were converted to their textual equivalents. Any remaining numeric values were then removed. Stop words were also removed from the reviews. Stop words are words extremely common in the English language that do not convey much meaningful information, for example "the", "of", and "is". Additional words specific to this analysis were also removed. These included words such as "battery", "car", and "Amazon", among others, as well as the three brand names and associated terms.

Normally, word stemming would also be a part of the text preparation process. In stemming, words are transformed to their root. For instance, “purchase”, “purchasing”, “purchases”, and “purchased” would all be stemmed to the same term as they are derivations of the same word. Given that the text body of interest was comprised of opinions, stemming was found to cause issues. Terms like “expensive” and “inexpensive” were stemmed to the same root, and that was not desired. Thus, a custom dictionary of words and their stems was crafted to allow some, but not all stemming for this analysis. This approach was possible because of the relatively small number of reviews; it would likely not be a feasible tactic with larger bodies of text.

Frequent two word combinations were identified in the cleaned customer reviews. These increased the depth of understanding of the reviews information. For example, “great” was one of the most frequent terms. Examining pairs of words gave more specific context: “great price”, “works great”, or “great product”. The most frequent word pairs were consolidated into single terms. Finally, the entire corpus of texts was transformed into a document term matrix. At the end of this cleaning process, only the core substance of the reviews was left. This meant for a few reviews, no text was left. These reviews did not have much substance to begin with, for example “ok” or “A+++++”. Not much information was lost by the elimination of these texts.

Term frequencies were calculated and examined. It was originally desired to extract product features by clustering reviews based on the document term matrix, but this approach had limited success. Instead, prevalent terms of interest were identified and further analysis was conducted on the usage of those words in reviews. Since analysis of relationships between three different variables (the word or phrase of interest, brand, and rating) was needed, log linear regression was used to model the contingency table cell counts. This tactic used a generalized linear model, using a log link function and a Poisson distributional assumption. Rating was treated as an ordinal rather than a purely nominal factor. The deviance was used to assess the goodness-of-fit of each final model and ensure its validity. Additionally, the residuals were reviewed to ensure no outliers existed.

Finally, various time durations were extracted since many reviews had mentions of N years (N being an integer value). The number of years mentioned in each review (if mentioned at all) was quantified. OLS regression was used to model the mentioned duration against the review rating and brand name, using similar techniques as used with the sales rank modeling (except no data were reserved for model validation). It was found that a square root transformation of the time duration was needed to achieve homogeneous variance in the residuals.

All code for the reviews analysis of the analysis can be found in Appendix D.

Discussion of Results

Importance of Branding and Customer Reviews to Sales Volume

Model A: Inclusion of Customer Reviews Volume

The first model for log sales rank constructed included reviews volume as a potential explanatory variable. Although brand and price were also initially included, the final model was found to be adequate without these variables and without any interactions of variables by the likelihood ratio test [$p = 0.455$]. Model A ultimately only included the fourth root of reviews volume, the average

customer review rating, and an indicator variable for whether the product had been reviewed yet (equal to 1 if there were no customer reviews and 0 otherwise).

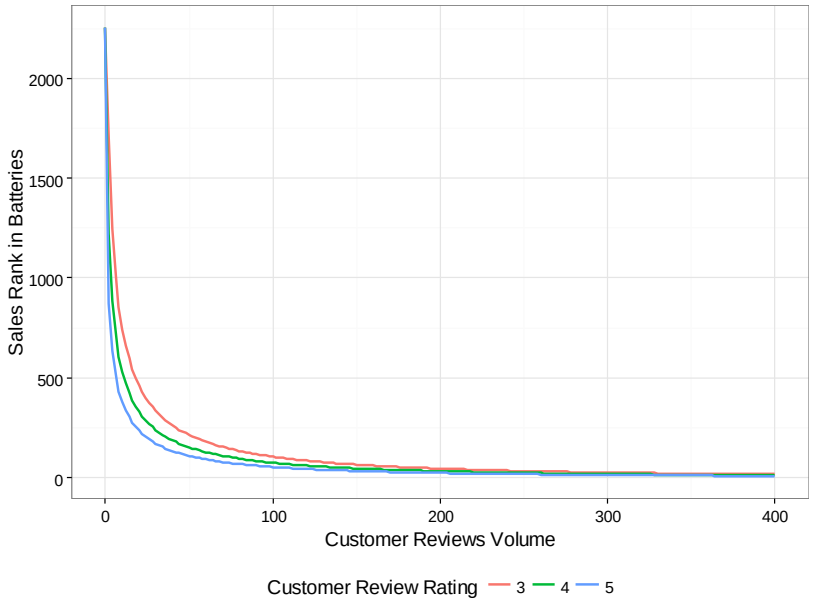
Not surprisingly, the no reviews indicator variable had high multi-collinearity with the other predictors, particularly the customer review rating (since this was set to 0 for products with no customer reviews). High multi-collinearity can add variability to parameter estimates and make them less stable; thus, this model may have fared poorly if used with future observations. To counter this, review ratings for products with no reviews were set to 3 (a neutral rating) instead of the 0 originally input. This decreased the multi-collinearity with negligible impact on the interpretation of the model. The variance inflation factors (VIFs) for both approaches can be reviewed in Appendix A, and the parameter estimates for this final model are in Table III below.

Table III. Parameter Estimates for Model A

Effect	Estimate	Std Error	t Value	p Value
Intercept	10.1	0.578	17.6	< 0.001
[Review Volume]$^{1/4}$	-1.42	0.0731	-19.4	<0.001
Rating	-0.338	0.128	-2.64	0.010
No Reviews	-1.41	0.263	-5.35	<0.001

Interpretation of the model parameters was somewhat complicated due to the transformations of various model parameters. Reviews volume staying constant, an increase of one star in the review rating was estimated to decrease the sales rank by 28.7% on average. Increases in reviews volume had more impact on sales rank at lower starting reviews volumes. For example, keeping reviews rating constant, an increase in customer review volume from 10 to 20 was estimated to decrease the sales rank by 37.9% on average; a similar increase from 20 to 30 reviews was estimated to decrease the sales rank by 27.4%. The biggest change was estimated to occur when moving from no reviews to a single customer review. Assuming that the first customer review had a three star rating, the sales rank was still estimated to decrease by 75.6% on average. These findings are presented visually in Figure 4 below.

Figure 4. Predicted Sales Rank vs Customer Reviews Volume by Reviews Rating



As mentioned previously, batteries which had higher sales volumes were likely to have had more product reviews due to the increased opportunities for such reviews to be provided. Thus, the finding that customer reviews volume and sales volume were related for the batteries studied did not necessarily mean that more customer reviews caused higher sales. However, what was evident was that the average reviews rating appeared to have a stronger effect at lower reviews volumes. Thus, initial positive feedback on new products may be crucial to growing sales.

Model B: Exclusion of Customer Reviews Volume

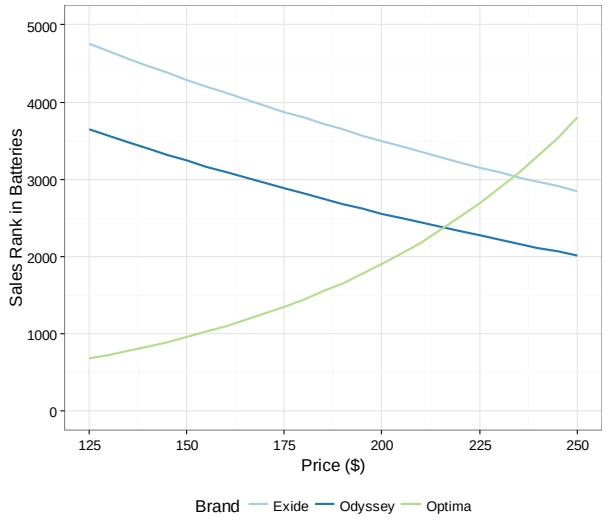
Fitting a model without customer reviews volume as a predictor led to a substantially different model form for the log sales rank. The indicator for no reviews did remain in the model to account for reviews which did not have a customer review rating. The final simplified model included main effects for brand, price, rating, and the no reviews indicator, in addition to two-way interactions between brand and price as well as price and rating. The parameter estimates are given in Table IV below.

Table IV. Parameter Estimates for Model B

Effect	Brand	Estimate	Std Error	t Value	p Value
Intercept		11.6	2.58	4.50	<0.001
Brand	Odyssey	-0.180	1.50	-0.12	0.905
Brand	Optima	-4.19	1.64	-2.55	0.013
Price		-0.0241	0.0114	-2.10	0.039
Rating		-1.49	0.548	-2.72	0.008
No Reviews		1.85	0.515	3.59	<0.001
Brand*Price	Odyssey	-0.0007	0.00747	-0.09	0.929
Brand*Price	Optima	0.0179	0.00835	2.15	0.035
Price*Rating		0.00667	0.00246	2.71	0.008

Batteries with no reviews were estimated to have 4.43 times higher sales ranks on average than batteries with a rating of 3 stars, everything else being equal. The predicted sales rank versus brand and price for batteries with no customer reviews is shown in Figure 5. Sales ranks, in general, were predicted to be high for batteries with no reviews as would be expected. The predicted sales rank increased with increasing price for the Optima brand; perhaps high price batteries with no reviews are considered high risk by consumers. Interestingly, however, the trend was reversed for Exide and Odyssey brands. Sales rank decreased with increasing price. Notably, Exide and Odyssey had fewer products at the low end of the price spectrum, so perhaps sparseness of data may have played some part in this predicted trend.

Figure 5. Predicted Sales Rank versus Price by Brand, for Products with No Customer Reviews



Predicted differences between the three brands became smaller for products with a customer rating. Figure 6 shows predicted sales ranks by brand and price for products with a 3 star rating, and Figure 7 shows similar trends for products with 5 stars. Five star rated products tended to have lower sales ranks, similar to what was found with Model A. At both the 3 and 5 star ratings, Optima batteries had the lowest sales ranks at lower price points, but Optima was surpassed by Odyssey at higher prices (roughly above $215) and Exide at even higher prices (roughly above $235).

Figure 6. Predicted Sales Rank versus Price by Brand, Customer Review Rating of 3 Stars

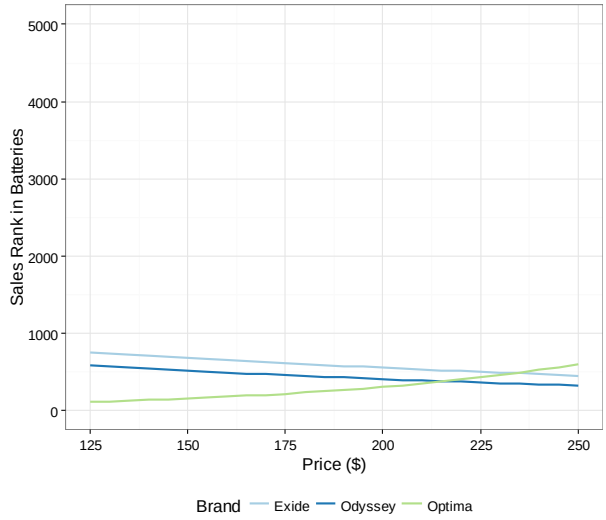
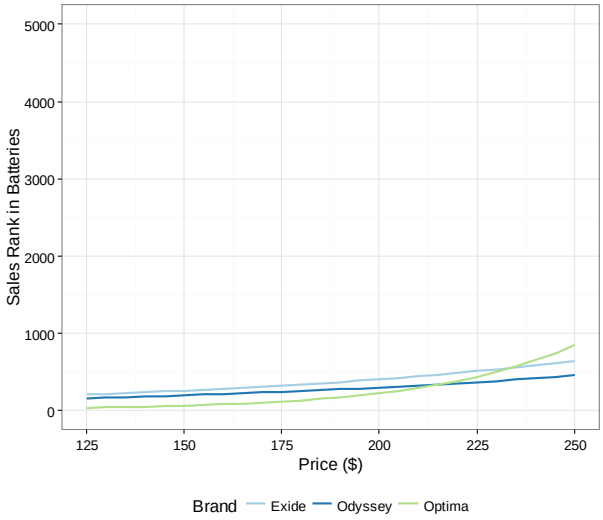


Figure 7. Predicted Sales Rank versus Price by Brand, Customer Review Rating of 5 Stars



In general, Exide had the highest sales ranks (lowest sales volume) for most rating and price combinations. Interestingly, Exide has the lowest average sales rank in the raw data (see Table I above), but controlling for price and customer reviews with this model gave different insight. Optima batteries were more sensitive to the effect of price at all rating levels.

Model Performance & Comparison

The two models developed provided different insights into the relationship of customer reviews, price, and branding with sales volume. The degree to which the models fit the data differed considerably. Model A explained more of the sales rank variation than did Model B and surpassed Model B in all performance metrics considered (see Table V). Model A had a considerably lower AIC value (lower is better), a smaller mean square error, and a better R^2, with Model A accounting for roughly 90% of the variation in log sales rank, compared to only 57% for Model B. Validation of both models with the test data set (the predicted values in Table V) showed no issues; the fit of the model to these observations was similar to that for the training data set.

Table V. Summary of Model Performance

Model	No. Terms	R^2	$R^2_{adjusted}$	MSE	AIC	$R^2_{predicted}$	$MSE_{predicted}$
Model A	4	0.9039	0.9002	0.2614	131.2	0.8827	0.2605
Model B	9	0.6156	0.5728	1.046	253.5	0.5866	0.9182

Residuals analysis was also completed for both models. Residual plots can be found in Appendix A. No major concerns were seen with the residuals for Model A; the residuals were approximately normally distributed with constant variance and a mean centered near zero versus all fitted values. The residuals for Model B had some potential concerns. Though the residuals were approximately normal, the residuals tended to be negative at low fitted values, indicating some possible issues with the mean specification. Clearly, the evidence has shown that Model B did not fit the data as well as Model A.

Finally, assessment of various influence diagnostics revealed no major concerns for Model A, and the associated plots can be found in the Appendix. Model B did have a few influential observations, which are summarized in Table VI. Battery ID 108 had a relatively large negative DFFIT value and a high covariance ratio; this battery had the highest price among Optima batteries by nearly $100 and one of the highest sales ranks overall. This high leverage point was likely largely responsible for the relatively steep upwards trend of sales ranks at high prices for Optima specifically.

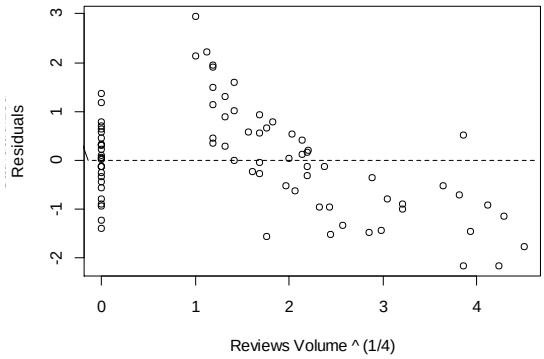
Table VI. Influential Observations for Model B

ID	DFFIT	Cov Ratio	Brand	Rating	Review Volume	Price	Sales Rank
14	0.67	0.43	Optima	5.0	1	$177.73	2,254
25	-1.16	1.12	Odyssey	5.0	9.5	$339.63	204
26	-0.13	1.93	Odyssey	N/A	0	$429.86	753
29	0.31	1.40	Exide	3.1	8	$159.01	1,088
42	-0.17	1.61	Odyssey	4.5	6.6	$389.86	694
44	0.21	1.59	Exide	5.0	3	$251.33	3,562
69	-0.13	1.41	Exide	4.9	8	$152.45	857
105	0.15	1.50	Odyssey	N/A	0	$375.40	1,406
108	-1.02	1.78	Optima	N/A	0	$354.51	6,602

Many of the influential observations had low, non-zero reviews volumes. Examining the residuals for Model B versus reviews volume, which was not included in the model, reveals that the residuals trended strongly versus reviews volume (see Figure 8). Observations with low review volumes tended to have

large positive residuals and those with high review volumes tended to have negative residuals. This may explain why many of the influential points have low reviews volume.

Figure 8. Model B Residuals versus Reviews Volume



Extraction of Feedback from Customer Reviews

Many different analyses were performed on the customer reviews data. Non-textual analysis of the reviews produced interesting results. First, the possible relationship between battery brand and the rating of the review was assessed. Both the Pearson chi-squared test [$X^2(8)$ = 19.0, p = 0.015] and log likelihood test [$G^2(8)$ = 16.5, p = 0.036] indicated brand and rating were related. An examination of the Pearson residuals (Table VII) revealed that reviews of Exide branded product contributed most strongly to the rejection of the null hypothesis of independence. There were nearly twice as many one star customer reviews for Exide products as what was expected; the quantity of five star reviews was also less than expected for Exide. Excluding Exide reviews, insufficient evidence existed to conclude that review rating and product brand (Odyssey or Optima) were related [$X^2(4)$ = 1.1, p = 0.894].

Table VII. Brand vs. Rating in Reviews
Pearson Test for Independence Expected Counts and Residuals

Brand	Count	1 star	2 stars	3 stars	4 stars	5 stars
Exide	*Observed*	31	6	8	24	149
	Expected	16.6	5.8	6.5	18.8	170.3
	Residual	**3.52**	0.09	0.59	1.21	**-1.63**
Odyssey	*Observed*	33	13	17	41	363
	Expected	35.6	12.4	13.9	40.2	364.8
	Residual	-0.44	0.17	0.83	0.12	-0.10
Optima	*Observed*	215	78	84	250	2,345
	Expected	226.7	78.8	88.6	256.0	2,321.8
	Residual	-0.78	-0.09	-0.49	-0.37	0.48

The review rating and the number of people who found the review to be helpful were also found to be related. A Spearman's correlation of -0.249 was found to exist between the two variables [$S = 1.02 \times 10^{10}$, $p < 0.0001$]. Kendall's tau found similar results, with $\tau = -0.230$ [$z = -15.1$, $p < 0.0001$], indicating the number of discordant pairs of points was significantly higher than what was expected by chance. Both of these results indicated the same overall trend: more people found lower rated reviews

to be helpful. This finding suggests that customers may pay more attention to negative reviews than to positive ones.

Analysis of the actual text in the reviews provided some interesting insights. The most frequently used terms were examined for negative (one to two stars), neutral (three stars), and positive (four to five stars) reviews. The top 75 terms for each group are presented visually in the word clouds in Figures 9 through 11 (note that the font size and color indicate word frequency, with more frequent words being larger and darker). The language in the negative and neutral reviews appeared to be somewhat similar, at least at a high level, with words like “dead”, “warranty”, and “replaced” being very frequent in both groups. This suggests these customers may have had issues with short battery life and perhaps the warranty policy.

Figure 9. Most Frequent Terms in Negative Reviews

Figure 10. Most Frequent Terms in Neutral Reviews

Figure 11. Most Frequent Terms in Positive Reviews

Frequent terms in the positive reviews were noticeably different, with "great" being the most prevalent. Another word that was surprisingly common in the positive reviews was "Prius". It was interesting that a specific vehicle model, the Toyota Prius, would be mentioned so frequently.

Some of the most interesting and prevalent words were taken from the word cloud analysis for further evaluation. Seven terms were selected: "Prius", "great", "price", "great price", "fit perfect", "dead", and "warranty" (note that "fit perfect" and "perfect fit" were considered to be equivalent and both counted). Associations between the term of interest, brand name, and rating were examined using log linear regression. The results are summarized in Table VIII, but the analysis of deviance for each model can be found in Appendix A.

Table VIII. Summary of Key Terms and Loglinear Models for Association

Term	N Reviews	Proportion of Reviews	Mean Rating	Model	Deviance	DF	p Value
Prius	290	0.0793	4.58	(BT, BR)	13.0	13	0.450
Great	967	0.2644	4.76	(RT, BR)	18.0	18	0.453
Price	457	0.1250	4.69	(BT, RT, BR)	6.1	15	0.978
Great price	101	0.0276	4.91	(RT, BR)	6.4	9	0.702
Perfect fit	157	0.0429	4.89	(BT, BR)	4.7	9	0.863
Dead	313	0.0856	3.54	(RT, BR)	22.2	17	0.178
Warranty	204	0.0558	3.09	(RT, BR)	27.8	18	0.066

Note: for the model specification, B = Brand, R = Rating, and T = Term

The association between brand and rating was found to be significant in all models, which agreed with previous testing for this association. Four of the terms ("great", "great price", "dead", and "warranty") were conditionally independent of brand name, controlling for rating. In other words, taking rating into consideration, there was insufficient evidence to conclude these phrases were more prevalent in reviews for one brand over the others. Not surprisingly, the words "dead" and "warranty" were more frequent in negative reviews, and the terms "great" and "great price" were more frequent in positive reviews. Keeping brand name constant, "dead" had 7.05 times higher odds of being in 1 star reviews than in 5 star reviews; "warranty" had 12.3 times higher odds for the same scenario.

Two of the terms ("Prius" and "perfect fit") were conditionally independent of rating, controlling for brand name. Brand, however, was associated with these terms. "Prius" was surprisingly mentioned in nearly 8% of the customer reviews and was the ninth most common term in all reviews. None of the Odyssey reviews mentioned the Prius model, and only five Exide reviews mentioned it. Optima had most of the Prius reviews – 9.6% of all Optima reviews mentioned the Prius. Further investigation revealed that the majority of these reviews were for a single part number, the Optima 8171-767 Yellow Top Prius Battery. Clearly this battery was specifically designed for the vehicle, and this approach appears to have been lucrative for Optima, at least on Amazon. The average sales rank for this battery was 10.8 during the study period, the second lowest rank of all the batteries sampled (indicating high sales volume). Toyota Prius owners may be one of the types of consumers who buy batteries online. It was also found that "perfect fit" was related to brand name. At first glance, it appeared that this term was less frequent in reviews for Odyssey products. However, it was found that the terms "Prius" and "perfect fit" were themselves related. Brand name and "perfect fit" were conditionally independent

when controlling for the presence of the word "Prius". Thus, Odyssey's lower occurrence of "perfect fit" may potentially be a function of a not having a product offering for the Toyota Prius. Apparently batteries tend to fit perfectly in Toyota Prius vehicles.

The word "price" had the most complex relationships and was found to have homogeneous association with brand and rating. "Price" was less common in Odyssey reviews at a given rating level. For any given review rating value, the odds that "price" appeared in an Optima review were 1.76 times that of it appearing in an Odyssey review. Surprisingly, the word "price" was more frequent in positive reviews for a given brand. Within the same brand's reviews, "price" had 2.37 greater odds of being written in a 5 star review than being in a 1 star review. Odyssey had the highest priced items, on average, and if price was mentioned more frequently in positive reviews, perhaps Odyssey products were not perceived to have the value of other brands. Interestingly, however, the two word phrase "great price" was not found to be related to brand, which might have been expected given these results.

Time in Battery Reviews

Lastly, time durations mentioned in customer reviews were assessed. Numerous customer reviews mentioned some amount of time in years. Though the exact context of this mention of duration was unknown, analysis was conducted on the information available. Roughly 13.1% of all the reviews mentioned some duration in years. Reviews with less than five stars were more likely to mention some time duration, as Table IX below shows.

Table IX. Summary of Reviews with Duration in Years Provided

Rating	N Mention Years	Proportion Mention Years	Mean Years
1 star	57	0.204	4.05
2 stars	31	0.320	3.45
3 stars	24	0.220	3.79
4 stars	47	0.149	4.96
5 stars	321	0.112	6.44
Total	*480*	*0.131*	*5.69*

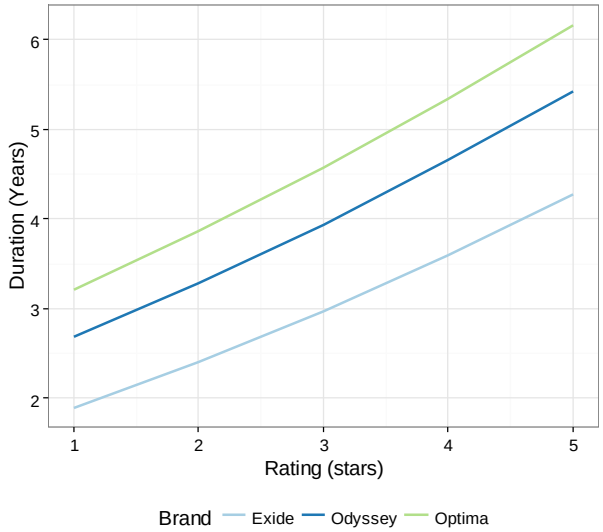
The number of years mentioned in the reviews correlated positively with the reviews rating. A significant model for the square root of the duration was found using OLS regression, with both review rating and brand name as predictors. The model parameter estimates are shown in Table X, and residuals plots, which showed no major issues, can be found in Appendix A. This model, though significant, did not explain much of the variation in the square root of the time duration. The $R^2_{adjusted}$ was only 0.149; in other words, the model only explained about 15% of the variation in the response.

Table X. Parameter Estimates for Durations Mentioned in Customer Reviews

Effect	Brand	Estimate	Std Error	t Value	p Value
Intercept		1.205	0.142	8.5	< 0.001
Brand	Odyssey	0.262	0.140	1.9	0.061
Brand	Optima	0.416	0.122	3.4	0.001
Rating		0.173	0.0202	8.5	< 0.001

Even though the model explained only a small part of the variation, it gave some indication on trends with rating and brand name. The predicted durations by review rating and brand name are plotted in Figure 12. The estimated number of years mentioned in reviews increased with review rating. Battery lifetime is important to customers, and thus the finding that the length of duration mentioned increased with review rating was not surprising. Differences also existed between the brands, with Exide having the lowest predicted durations mentioned on average and Optima having the highest.

Figure 12. Predicted Duration by Review Rating and Brand



Conclusions

This research showed that manufacturers can gain meaningful insights from data gathered off online retailer sites, like Amazon.com. While this study did not reveal customer preferences for specific technical product features, it did show some interesting differences in how various battery brands fare in the online marketplace. Thus, such analyses may be a good conduit of customer feedback for competitive comparisons.

The Exide branded batteries appear to have the most ground to make up, despite having the lowest average sales rank in the raw data. Exide has more one star and fewer five star customer reviews than the other two brands. To make matters worse, Amazon users tend to find more low rated reviews to be helpful than the higher rated reviews. Although the occurrence of negative terms like “dead” and “warranty” in customer reviews is not related to brand name *when review rating is held constant*, the Exide brand has more negative reviews than the other two brands. This finding suggests that Exide batteries could have more premature failures than the other brands. Further investigation is necessary to validate that hypothesis, but there is additional evidence from the customer reviews to suggest that short battery life is an issue for Exide. Exide generally has lower time durations (in years) mentioned in its reviews than either Optima or Odyssey.

One of Optima’s biggest strengths in the online marketplace may be its specific battery model for the Toyota Prius. This battery has one of the highest sales volumes for automotive batteries on Amazon, and numerous customer reviews talk about the Prius. The words “perfect fit” are also often

mentioned in conjunction with the Prius more than in other reviews. Perhaps customers enjoy the comparatively small footprint of this battery type.

Odyssey may need to review the pricing of their batteries to become more competitive. Odyssey has the highest priced batteries on average. The word "price" is mentioned less frequently in Odyssey reviews, and this word is more prevalent in positive reviews for all brands. Thus, one conclusion might be that the comparatively lower prices for the Exide and Optima brands are seen as a favorable characteristic that Odyssey does not share.

Finally, sales volume is strongly linked to the number of customer reviews. Whether the high volume of customer reviews promotes sales or whether higher sales volume creates more customer reviews is unknown. What is known is that review ratings have more impact on sales rank for batteries with few customer reviews – calling to mind the adage "first impressions are everything". The original intent with the sales rank data was to analyze the trends longitudinally over time, but the two week study period was too short for this purpose. An area for future research would be to scrape product information off Amazon for this same sample after more time has elapsed – perhaps three to six months – and then conduct a longitudinal study. Are first impressions really everything? Do battery types with a few extremely positive reviews increase in sales volume more rapidly than batteries with a few lackluster reviews? Such an investigation would be an interesting extension of this study.

References

Amazon.com, Inc. (2016, June 15-29). Product information and customer reviews [HTML source code]. Retrieved from http://amazon.com

Decker, R. & Trusov, M. (2010). Estimating aggregate consumer preferences from online product reviews. *International Journal of Research in Marketing*, *27*, 293-307.

Floyd, K., Freeling, R., Alhoqail, S., Cho, H. Y., & Freling, T (2014). How online product reviews affect retail sales: a meta-analysis. *Journal of Retailing, 90*(2), 217-232.

Fretwell, L., Stine, J., Sethi, H., & Noronha, A. Cisco Internet Business Solutions Group (2013). 'Catch and keep' digital shoppers: how to deliver retail their way. Retrieved from http://www.cisco.com/c/dam/en_us/about/ac79/docs/retail/Catch-and-Keep-the-Digital-Shopper_PoV.pdf

Zhu, F. & Zhang X. (2010). Impact of online consumer reviews on sales: the moderating role of product and consumer characteristics. *Journal of Marketing*, *74*(2), 133-148.

Appendix A – Supplementary Figures and Tables

Table A1. Variance Inflation Factors for Model A

Effect	Rating of No Reviews = 0	Rating of No Reviews = 3
[Review Volume]$^{1/4}$	2.77	2.77
Rating	21.80	3.11
No Reviews	22.52	4.54

Figure A1. Residual Plots for Model A

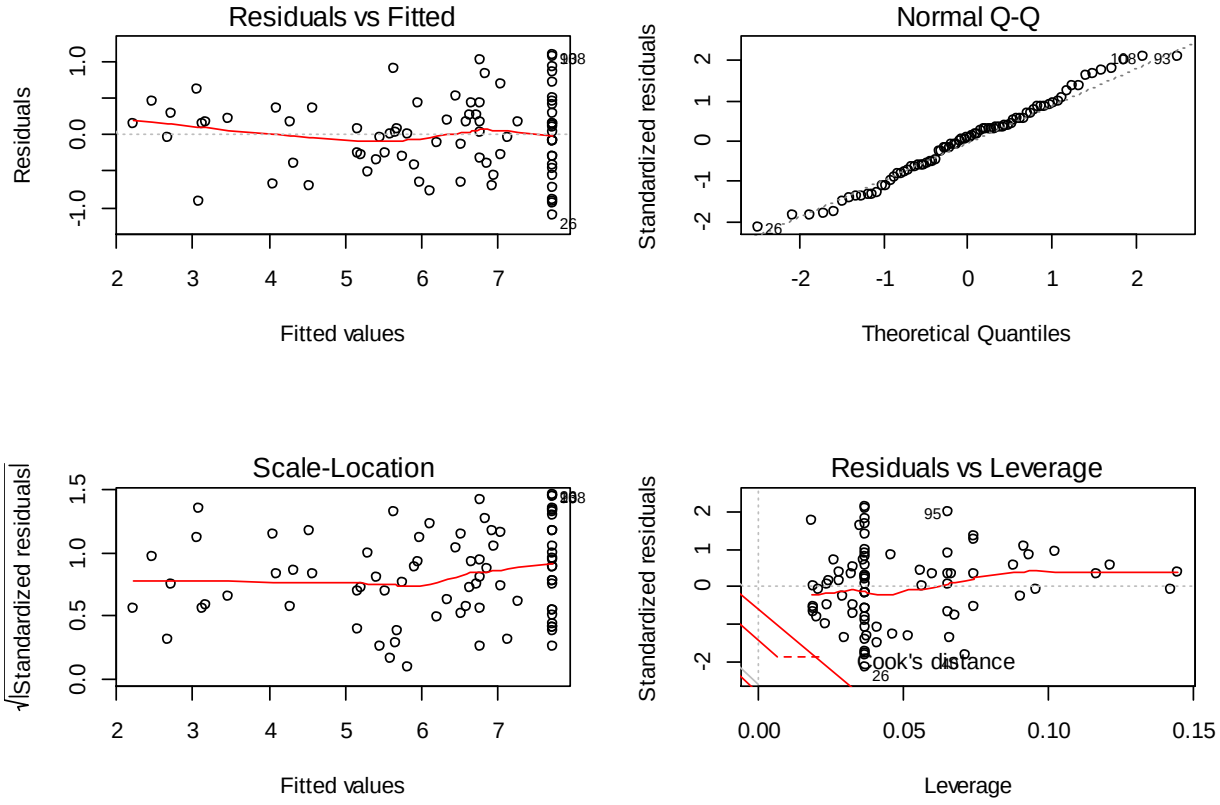


Figure A2. Influence Diagnostic Plots for Model A

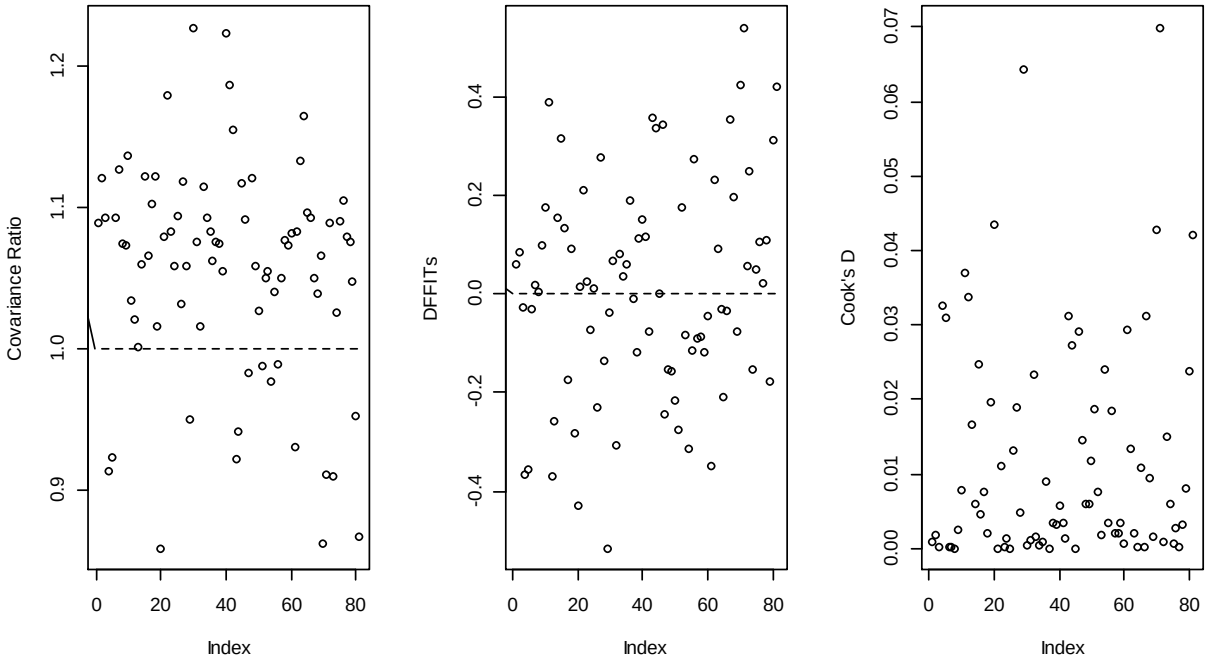


Table A2. Variance Inflation Factors for Model B

Effect	VIF
Brand	4.08
Price	6.09
Rating	3.65
No Reviews	2.02
Brand*Price	4.71
Price*Rating	5.37

Figure A3. Residual Plots for Model B

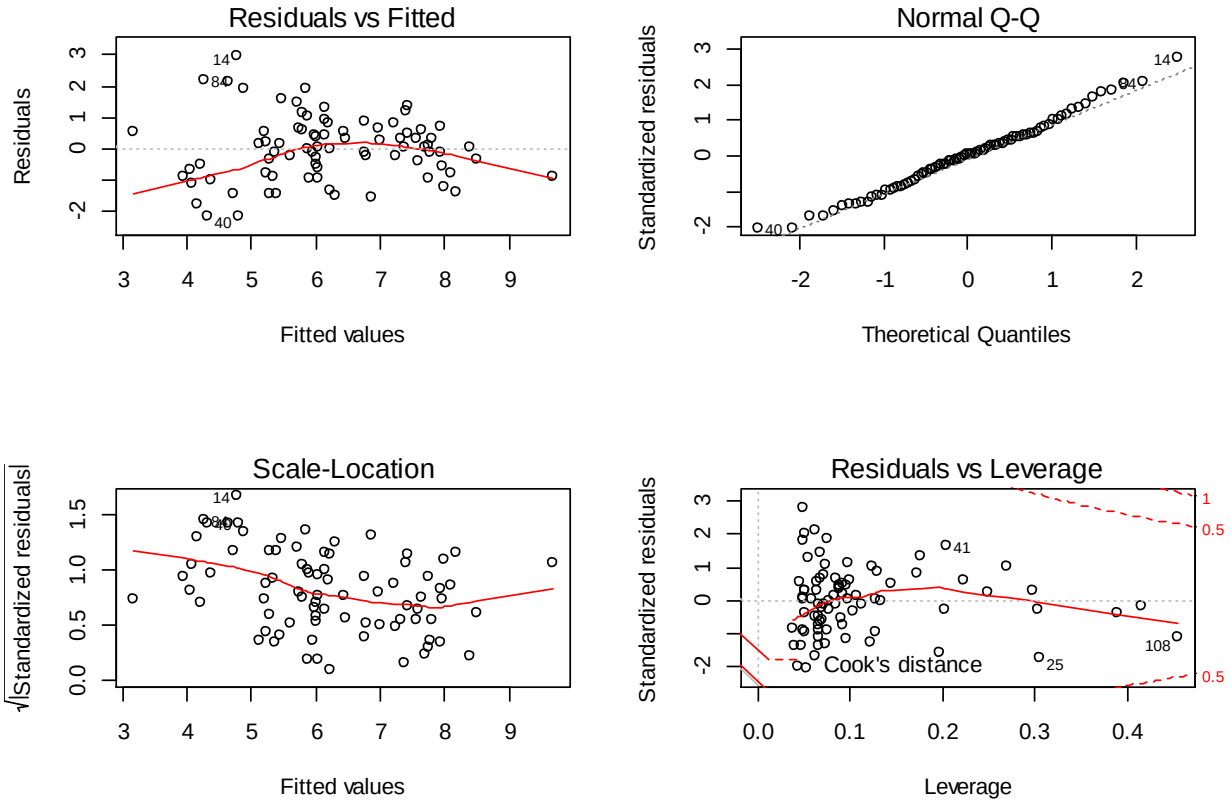


Figure A4. Influence Diagnostic Plots for Model B

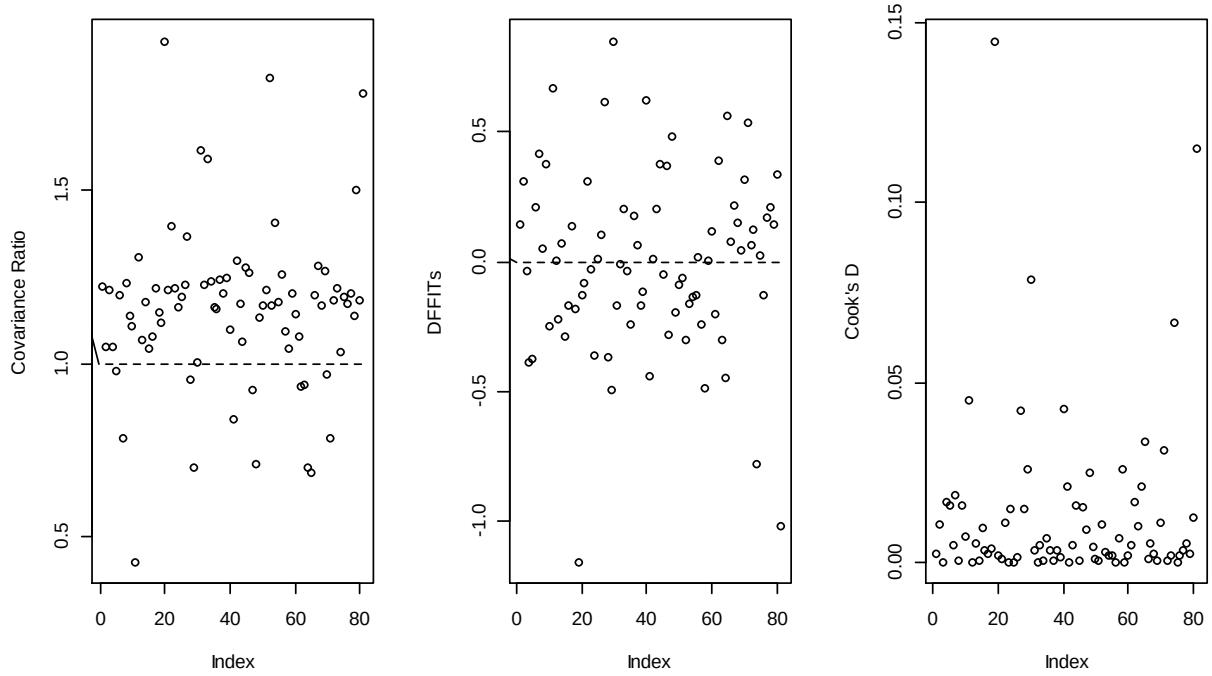


Table A3. Analysis of Deviance Table
Loglinear Model with Brand, Rating, and "Prius"

Effect	G^2	DF	p Value
Brand	3491.5	2	< 0.001
Rating	1310.5	3	<0.001
Prius	2471.1	1	<0.001
Brand*Rating	13.7	2	0.001
Brand*Prius	16.6	1	<0.001

Table A4. Analysis of Deviance Table
Loglinear Model with Brand, Rating, and "Great"

Effect	G^2	DF	p Value
Brand	3645.8	2	< 0.001
Rating	1222.7	3	<0.001
Great	838.4	1	<0.001
Brand*Rating	13.5	2	0.001
Rating*Great	93.2	1	<0.001

Table A5. Analysis of Deviance Table
Loglinear Model with Brand, Rating, and "Price"

Effect	G^2	DF	p Value
Brand	3643.1	2	< 0.001
Rating	1292.4	3	<0.001
Price	2296.8	1	<0.001
Brand*Rating	13.4	2	0.001
Rating*Price	18.6	1	<0.001
Brand*Price	11.5	2	0.003

Table A6. Analysis of Deviance Table
Loglinear Model with Brand, Rating, and "Great Price"

Effect	G^2	DF	p Value
Brand	3641.3	2	< 0.001
Rating	1288.7	3	<0.001
Great Price	3723.8	1	<0.001
Brand*Rating	13.7	2	0.001
Rating*Great Price	9.3	1	0.002

Table A7. Analysis of Deviance Table
Loglinear Model with Brand, Rating, and "Perfect Fit"

Effect	G^2	DF	p Value
Brand	3639.5	2	< 0.001
Rating	1311.6	3	<0.001
Perfect Fit	3211.4	1	<0.001
Brand*Rating	14.2	2	0.001
Brand*Perfect Fit	8.4	2	0.015

Table A8. Analysis of Deviance Table
Loglinear Model with Brand, Rating, and "Dead"

Effect	G^2	DF	p Value
Brand	3625.6	2	< 0.001
Rating	1168.0	3	<0.001
Dead	2907.6	1	<0.001
Brand*Rating	14.3	2	0.001
Rating*Dead	157.0	1	<0.001

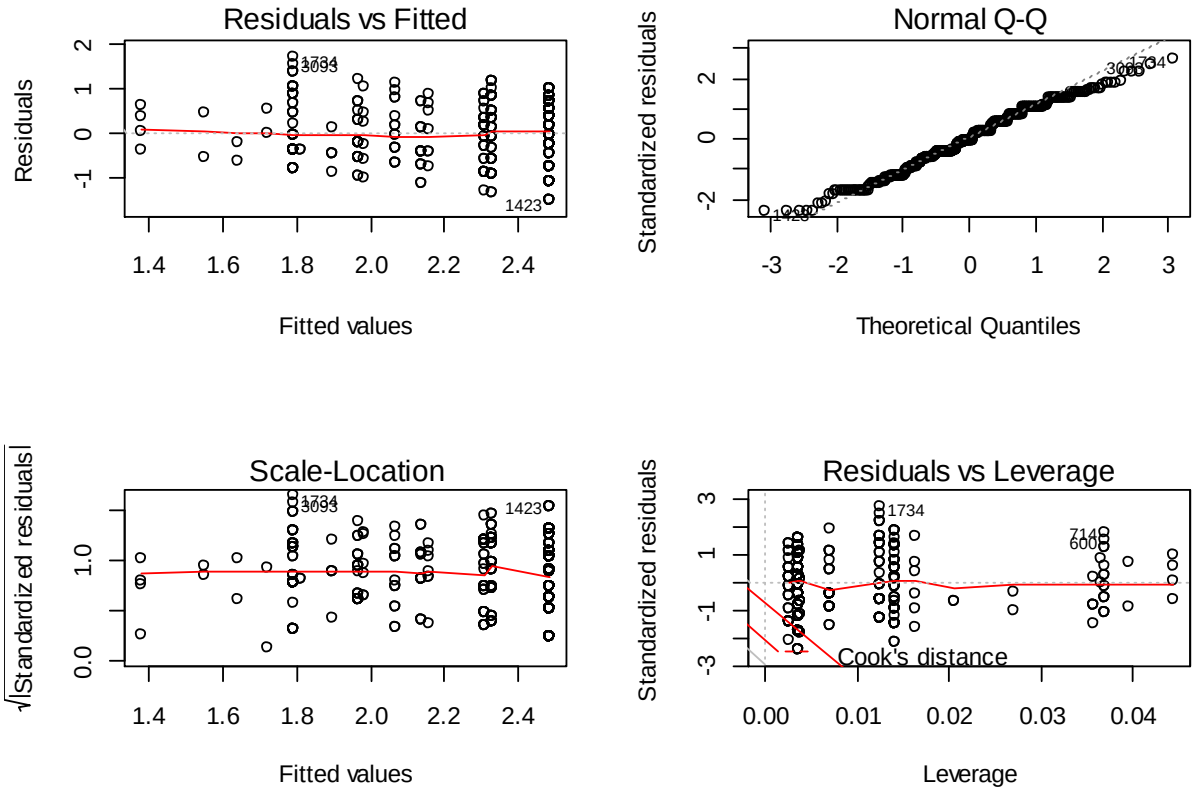
Table A9. Analysis of Deviance Table
Loglinear Model with Brand, Rating, and "Warranty"

Effect	G^2	DF	p Value
Brand	3644.8	2	< 0.001
Rating	1149.1	3	<0.001
Warranty	3488.1	1	<0.001
Brand*Rating	13.9	2	0.001
Rating*Warranty	198.1	1	<0.001

Table A10. Analysis of Deviance Table
Loglinear Model with Brand, "Prius", and "Perfect Fit"

Effect	G^2	DF	p Value
Brand	3498.3	2	< 0.001
Prius	2478.7	1	<0.001
Perfect Fit	3774.0	1	<0.001
Brand*Prius	17.8	1	<0.001
Prius*PerfectFit	99.7	1	<0.001

Figure A5. Residual Plots for Durations in Reviews Model



Appendix B – R Code for Web Scraping

```
################################################################
### AMAZON WEB SCRAPING CODE
################################################################

## Scraping Functions
################################################################

##Function to scrape product listing
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
getURLs <- function(searchURL, start=1){
#Gather URLs for individual products by web scraping search results

	urlcontents <- readLines(searchURL)

	#Initialize links vector
	links <- rep("", times=24)

	#Iterate through each search result
	for (i in start:(start+23)){
		#Find line with i-th search result
		keyline <- grep(paste('result', i, sep='_'),urlcontents)

		#Determine if i-th search result was found
		if (length(keyline)!=0){
			#Extract relevant line
			line <- urlcontents[keyline[1]]
			#Split before and after links "href="
			templinks <- strsplit(line, 'href="')[[1]]
			#Determine which segment contains "result#" & grab previous
			reslink <- grep(paste('result', i, sep='_'),templinks)
			links[i+1-start] <- strsplit(templinks[reslink+1], 'ref=')[[1]][1]
		}
	}
	links <- links[links!=""]

	#Find next page
	nextpage1 <- grep('title="Next Page"', urlcontents)
	#Does next page exist?
	if (length(nextpage1)==0){
		return(links)
	} else {
		nextarea <- urlcontents[nextpage1:(nextpage1+10)]
		nextpage2 <- grep('href=', nextarea)
		p2.address <- nextarea[nextpage2]
		#Clean up and format address
		p2.address <- gsub(" ", "", p2.address) #remove spaces
		p2.address <- gsub("&amp;", "&", p2.address) #replace ampersands
		p2.address <- gsub('href="', "", p2.address) #remove beginning
		p2.address <- gsub('">', "", p2.address) #remove end
		nextlink <- paste('https://www.amazon.com', p2.address, sep="")
		return(c(links, getURLs(nextlink, start=i+1)))
	}
}

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


##Function to scrape product details
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
productDetails <- function(URL){
#Scrape product name, price, rating, ranking, and customer review info from URL
```

```
if (runif(1)>0.5){
	mywait()
}
urlcontents <- readLines(URL, warn=FALSE)
visit.time <- Sys.time()

#Find title
	nameline <- grep('meta name="title"', urlcontents)
	lines <- strsplit(urlcontents[nameline], ":")[[1]]
	Name <- lines[2]

#Find price
	priceline <- grep('"priceblock_ourprice"', urlcontents)
	if (length(priceline)==0){
		Price <- NA
	} else {
		lines <- strsplit(urlcontents[priceline], ">")[[1]]
		Price <- as.numeric(gsub("[^0-9.]", "", lines[2]))
	}

#Average rating
	ratinglines <- urlcontents[grep('out of 5 stars', urlcontents)]
	ratingline <- grep('Histogram', ratinglines)
	if (length(ratingline)==0){
		Rating <- NA
	} else {
		lines <- strsplit(ratinglines[ratingline[1]], "out of")[[1]]
		Rating <- as.numeric(gsub("[^0-9.]", "", lines[1]))
	}


#Ranking
	bestline <- grep('Best Sellers Rank', urlcontents)
	if (length(bestline)==0){
		RankAuto <- NA
		RankBatteries <- NA
	} else {
		bestlines <- urlcontents[bestline:(bestline+15)]
		ranklines <- bestlines[grep('#', bestlines)]
		autoline <- ranklines[grep('See top 100', ranklines)]
		if (length(autoline) == 0){
			RankAuto <- NA
		} else {
			lines <- strsplit(autoline, "in")[[1]]
			RankAuto <- as.numeric(gsub("[^0-9]", "", lines[1]))
		}
		battline <- ranklines[grep('Replacement Parts', ranklines)]
		lines <- strsplit(battline, "in")[[1]]
		RankBatteries <- as.numeric(gsub("[^0-9]", "", lines[1]))
	}

#Customer Reviews
	if (length(grep("Be the first to review this item", urlcontents))>0){
		ReviewLink <- ""
		ReviewQty <- 0
	} else {
		reviewline <- grep("see_all_summary", urlcontents)
		lines <- strsplit(urlcontents[reviewline],'"')[[1]]
		ReviewLink <- lines[4]
		ReviewQty <- as.numeric(gsub("[^0-9]", "", lines[5]))
		if (is.na(ReviewQty)){
			if (length(grep("both", lines[5]))>0){
				ReviewQty <- 2
```

```
                        }else if (length(grep("See the customer review",
                        lines[5]))>0){
                              ReviewQty <- 1
                        }
                  }
            }

      #Combine into data frame and return
            product <- data.frame(URL, Name, Price, Rating, RankAuto, RankBatteries,
                  ReviewLink, ReviewQty, visit.time)
            return(product)
}
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

##Function to scrape product reviews
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
grabComments <- function(commentsURL){
#Grab all comments for a given product
      guts <- readLines(commentsURL, warn=FALSE)

      #Find next page of comments
      next.inds <- grep(">Next<", guts)
      if (length(next.inds)==0){
            return(commentDetails(guts))
      } else {
            next.line <- guts[next.inds]
            link.pages <- strsplit(next.line, '<li')[[1]]
            next.link <- link.pages[grep(">Next<", link.pages)]
            if (length(grep("a-disabled", next.link))>0){
                  return(commentDetails(guts))
            } else {
                  mywait()
                  temp <- strsplit(next.link, '">')[[1]]
                  temp <- temp[grep('href="', temp)][1]
                  next.URL <- paste("https://amazon.com", gsub('<a href="', "",
                  temp), sep="")
                  return(rbind(commentDetails(guts), grabComments(next.URL)))
            }
      }
}
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


##Helper function to scrape review details
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
commentDetails <- function(URLcontents){
#Get comment details such as ID, title, author, date, text, rating, verified
#purchase, and number of people who voted helpful

      commentlines <- grep("review-title", URLcontents)
      top.review <- grep('>Top positive review<', URLcontents)
      critical.review <- grep('>Top critical review<', URLcontents)
      commentlines <- commentlines[!(commentlines %in% c(top.review,
                        critical.review))]
      temp <- rep(NA, length(commentlines))

      comments <- data.frame(ID=temp, Title=temp, Author=temp, Date=temp,
                  Rating=temp, Text=temp, Verified=temp, Votes=temp)

      for (i in 1:length(commentlines)){
            templine <- paste(URLcontents[commentlines[i]],
                        URLcontents[commentlines[i]+1],
                        URLcontents[commentlines[i]+2], sep="")
```

```
templine <- gsub("<br />","",templine)
segments <- strsplit(templine, "><")[[1]]

#Review ID
      pos.segs <- segments[grep("div id=", segments)]
      ID.seg <- pos.segs[grep("a-section review", pos.segs)]
      temp <- strsplit(ID.seg, '=\"')[[1]][2]
      temp <- gsub('\" ', "", temp)
      comments[i,"ID"] <- gsub("class", "", temp)

#Review title
      title.seg <- segments[grep("review-title", segments)]
      temp <- strsplit(title.seg, ">")[[1]][2]
      comments[i,"Title"] <- gsub("</a", "", temp)

#Review author
      author.seg <- segments[grep("author", segments)]
      temp <- strsplit(author.seg, '">')[[1]][2]
      comments[i, "Author"] <- gsub("</a", "", temp)

#Get date
      date.seg <- segments[grep("review-date", segments)]
      temp <- strsplit(date.seg, ">on ")[[1]][2]
      comments[i,"Date"] <- gsub("</span","",temp)

#Get rating
      rating.seg <- segments[grep("out of 5 stars", segments)]
      temp <- strsplit(rating.seg, "out of")[[1]][1]
      comments[i,"Rating"] <- as.numeric(gsub("[^0-9.]", "", temp))

#Main text
      text.seg <- segments[grep("review-text", segments)]
      temp <- strsplit(text.seg, 'review-text\">')[[1]][2]
      comments[i,"Text"] <- gsub("</span","",temp)

#Verified Purchase
      verify <- grep("Verified Purchase", segments)
      if (length(verify)>0){
            comments[i,"Verified"] <- 1
      } else {
            comments[i,"Verified"] <- 0
      }

#Found helpful
      vote.loc <- grep("review.votes", segments)
      if (length(vote.loc)>0){
            vote.seg <- segments[vote.loc]
            temp <- strsplit(vote.seg, 'review-votes\">')[[1]][2]
            if (length(grep("One", temp))>0){
                  comments[i,"Votes"] <- 1
            } else {
                  comments[i,"Votes"] <- as.numeric(gsub("[^0-9]", "",
                  temp))
            }
      } else {
            vote.lines <- grep("review.votes", URLcontents)
                        not.used <- vote.lines[!(vote.lines %in%
                        c(commentlines, top.review, critical.review))]
            if (i == nrow(comments)){
                  vote.line <- not.used[not.used > commentlines[i]]
            } else {
                  vote.line <- not.used[not.used > commentlines[i] &
                              not.used < commentlines[i+1]]
```

```
                        }
                        if (length(vote.line==1)){
                              templine <- URLcontents[vote.line]
                              templine <- gsub("<br />","",templine)
                              segments <- strsplit(templine, "><")[[1]]
                              vote.seg <- segments[grep("review.votes", segments)]
                              temp <- strsplit(vote.seg, 'review-votes\">')[[1]][2]
                              if (length(grep("One", temp))>0){
                                    comments[i,"Votes"] <- 1
                              } else {
                                    comments[i,"Votes"] <- as.numeric(gsub("[^0-
                                    9]", "", temp))
                              }
                        } else {
                              comments[i, "Votes"] <- 0
                        }

                  }
      }
      return(comments)
}

#################################################################


## Scrape product list
#################################################################
##URLs to brand specific product listings
optima <-
"https://www.amazon.com/s/ref=lp_15719921_nr_p_89_2?fst=as%3Aoff&rh=n%3A15684181%2Cn%3
A!15690151%2Cn%3A15719731%2Cn%3A15719911%2Cn%3A15719921%2Cp_89%3AOptima&bbn=15719921&i
e=UTF8&qid=1465767410&rnid=2528832011"

odyssey <- "https://www.amazon.com/s/ref=sr_in_-
2_p_89_34?fst=as%3Aoff&rh=n%3A15684181%2Cn%3A!15690151%2Cn%3A15719731%2Cn%3A15719911%2
Cn%3A15719921%2Cp_89%3AOdyssey&bbn=15719921&ie=UTF8&qid=1465767448&rnid=2528832011"

exide <- "https://www.amazon.com/s/ref=sr_in_-
2_p_89_21?fst=as%3Aoff&rh=n%3A15684181%2Cn%3A!15690151%2Cn%3A15719731%2Cn%3A15719911%2
Cn%3A15719921%2Cp_89%3AExide&bbn=15719921&ie=UTF8&qid=1465767484&rnid=2528832011"

##Get individual product URLs for each brand
optima.links <- getURLs(optima)
odyssey.links <- getURLs(odyssey)
exide.links <- getURLs(exide)

##Concatenate all three groups of URLs into one vector
battery.links <- c(optima.links, odyssey.links, exide.links)

#Remove items that are not batteries
battery.links <- battery.links[-grep("Maintainer", battery.links)]
battery.links <- battery.links[-grep("Protectors", battery.links)]
battery.links <- battery.links[-grep("HK", battery.links)]
battery.links <- battery.links[-grep("Gauge", battery.links)]
battery.links <- battery.links[-grep("EHM327BK", battery.links)]

#Save battery URL vector
save(battery.links, file="links.RData")



## Scrape details for each product
#################################################################
```

```
##Randmize order of battery URLs
temp.links <- battery.links[sample(1:length(battery.links), length(battery.links))]

#Iterate through each battery URL and extract details
for (i in 1:length(temp.links)){
       temp <- productDetails(temp.links[i])
       if (i == 1){
              battery.details <- temp
       } else {
              battery.details <- rbind(battery.details, temp)
       }
}

#Eliminate batteries not currently being sold
battery.details <- battery.details[!is.na(battery.details$Price), ]

#Eliminate batteries which have no sales rank
battery.details <- battery.details[!is.na(battery.details$RankAuto), ]

#Save results
save(battery.details, file="jun29.RData")


## Scrape product reviews for each battery
############################################################
#Extract only batteries that have product reviews
comment.types <- battery.details[battery.details$ReviewQty > 0, ]

#Determine total number of reviews to be extracted
tot.comments <- sum(comment.types$ReviewQty)

#Convert factor classes to strings
comment.types$URL <- as.character(comment.types$URL)
comment.types$Name <- as.character(comment.types$Name)

#Initiate blank data frame
temp <- rep(NA, tot.comments)
all.comments <- data.frame(Prod.URL=temp, Prod.Name=temp, ID=temp, Title=temp,
                     Author=temp, Date=temp, Rating=temp, Text=temp, Verified=temp,
                     Votes=temp)

#Iterate through each battery type's comments
k <- 1
for (i in 1:nrow(comment.types)){
       #Extract all comments for a given battery
       temp.comments <- grabComments(as.character(comment.types[i,
                     "ReviewLink"]))
       #Determine number of comments extracted and properly advance index
       #for comments data frame
       N <- nrow(temp.comments)
       all.comments[k:(k+N-1), 1] <- as.character(comment.types[i, 1])
       all.comments[k:(k+N-1), 2] <- as.character(comment.types[i, 2])
       all.comments[k:(k+N-1), 3:10] <- temp.comments
       k <- k+N
}

#Save product reviews
save(all.comments, file="Comments.RData")
```

Appendix C – R Code for Linear Regression of Sales Rank Data

```
library(sqldf)
library(ggplot2)
library(car)

## Data Compilation & Sample Statistics / Plots
#############################################################
#Read in all raw data files and compile
#Create vector of file names
days <- seq(15, 29, by=2)
filenames <- paste("jun", days, ".RData", sep="")

#Combine data from all days into one file
for (i in 1:length(filenames)){
       load(filenames[i])
       battery.details$Wave <- i
       if (i==1){
              raw.data <- battery.details
       } else {
              raw.data <- rbind(raw.data, battery.details)
       }
}

#Assign brands
raw.data$Brand[grep("Odyssey", raw.data$Name, ignore.case=TRUE)] <- "Odyssey"
raw.data$Brand[grep("Exide", raw.data$Name, ignore.case=TRUE)] <- "Exide"
raw.data$Brand[grep("Optima", raw.data$Name, ignore.case=TRUE)] <- "Optima"
raw.data$Brand[grep("Hawker", raw.data$Name, ignore.case=TRUE)] <- "Odyssey"
raw.data$Brand[grep("Odyssey", raw.data$URL, ignore.case=TRUE)] <- "Odyssey"

#Assign Product IDs
raw.data$ID <- as.numeric(raw.data$Name)


## Creation of summary statistics by brand
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
brand.summary <- sqldf('select Brand, count(ID)/8 as N, avg(Price) as AvgPrice,
                           avg(Rating)  as AvgRating, avg(ReviewQty) as AvgReviewQty,
                           avg(RankBatteries) as AvgRank, min(RankBatteries) as
                           MinRank, max(RankBatteries) as MaxRank from "raw.data"
                           group by Brand')

## Creation of individual level graphics
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
big.changes <- raw.data[raw.data$ID %in% c(3, 26, 79, 80, 95),]
stable.IDs <- raw.data[raw.data$ID %in% c(6, 23, 43, 66, 107),]

#Plot individuals with large swings in sales rank over time
rank.BIG <- ggplot(big.changes, aes(x=Wave, y=RankBatteries, group=as.factor(ID))) +
             coord_cartesian() + geom_line(aes(color=as.factor(ID)), size=1.1) +
             theme_bw() + labs(x="Date", y="Sales Rank in Batteries") +
             scale_x_continuous(breaks=c(2,4,6,8), labels=c('Jun 17', 'Jun 21', 'Jun
             25', 'Jun 29')) + theme(legend.position="bottom",
```

```
            legend.key=element_blank()) + scale_y_continuous(limits = c(0,6000)) +
            scale_color_brewer(name="Battery ID", type="qual", palette=3)

#Plot individuals with relatively stable sales rank over time
rank.stable <- ggplot(stable.IDs, aes(x=Wave, y=RankBatteries, group=as.factor(ID))) +
            coord_cartesian() + geom_line(aes(color=as.factor(ID)), size=1.1) +
theme_bw() +
            labs(x="Date", y="Sales Rank in Batteries") + scale_x_continuous(
            breaks=c(2,4,6,8), labels=c('Jun 17', 'Jun 21', 'Jun 25', 'Jun 29')) +
            theme(legend.position="bottom", legend.key=element_blank()) +
            scale_y_continuous(limits = c(0,6000)) + scale_color_brewer(
            name="Battery ID", type="qual", palette=3)


## Data Preparation for Modeling
#############################################################
#Create an average summary by battery ID
raw.summary <- sqldf('select ID, Brand, avg(Price) as Price,
                          avg(Rating)  as Rating, avg(ReviewQty) as ReviewQty,
                          avg(RankAuto) as RankAuto, avg(RankBatteries) as
                          RankBatteries from "raw.data" group by ID, Brand')

#Create additional variables
raw.summary$LRank <- log(raw.summary$RankBatteries) #Log rank response
raw.summary$Rating[is.na(raw.summary$Rating)] <- 0 #Set no reviews to 0
raw.summary$NoReviews <- as.numeric(raw.summary$ReviewQty==0) #Indicator
raw.summary$LReviewQty <- (raw.summary$ReviewQty)^(1/4) #4th root volume
raw.summary$Rating2 <- raw.summary$Rating #Create a copy of rating
raw.summary$Rating2[raw.summary$Rating==0] <- 3 #Set no reviews to 3

##Split into test and training data sets
set.seed(152983)
test.inds <- sample(1:nrow(raw.summary), 0.25*nrow(raw.summary))
train.data <- raw.summary[-test.inds,]
test.data <- raw.summary[test.inds,]


## MODEL A DEVELOPMENT & ANALYSIS - Review Volume Included
#############################################################
##Fit full model, including reviews volume as a predictor
full.model <- lm(RankBatteries ~ Brand + Price + LReviewQty + Rating +
                    Brand:Price + Brand:LReviewQty + Brand:Rating +
                    Price:LReviewQty + Price:Rating + LReviewQty:Rating +
                    Brand:Price:LReviewQty + Brand:Price:Rating +
                    Brand:LReviewQty:Rating + Price:LReviewQty:Rating,
                    data=train.data)
#Check for appropriate response transformation
boxCox(full.model)#Indicates log transformation appropriate

#Fit full model with log transformation
full.model.A <- lm(LRank ~ Brand + Price + LReviewQty + Rating2 + NoReviews +
                    Brand:Price + Brand:LReviewQty + Brand:Rating2 + Brand:NoReviews +
                    Price:LReviewQty + Price:Rating2 + Price:NoReviews +
LReviewQty:Rating +
                    Brand:Price:LReviewQty + Brand:Price:Rating2 +
```

```
                    Brand:LReviewQty:Rating2 + Price:LReviewQty:Rating2,
                    data=train.data)
summary(full.model.A)

#Reduce model as possible
reduced.model.A <- lm(LRank ~ LReviewQty + Rating2 + NoReviews, data=train.data)
summary(reduced.model.A)
vif(reduced.model.A)
avPlots(reduced.model.A)
anova(reduced.model.A) #Type I Effects Analysis
anova(reduced.model.A, full.model.A) #Likelihood ratio test

mean(reduced.model.A$residuals^2) #Calculate MSE
AIC(reduced.model.A)
par(mfrow=c(2,2))
plot(reduced.model.A) #Residuals plots


#Influence diagnositics
infl.A <- influence.measures(reduced.model.A)
summary(infl.A)
par(mfrow=c(1,3))
cov.A <- infl.A$infmat[,"cov.r"]
plot(cov.A, ylab="Covariance Ratio")
abline(h=1, lty=2)
dffit.A <- infl.A$infmat[,"dffit"]
plot(dffit.A, ylab="DFFITs")
abline(h=0, lty=2)
cookD.A <- infl.A$infmat[,"cook.d"]
plot(cookD.A, ylab="Cook's D")


## MODEL B DEVELOPMENT & ANALYSIS - Review Volume Excluded
############################################################
#Fit full model
full.model.B <- lm(LRank ~ Brand + Price + Rating2 + NoReviews +
                    Brand:Price + Brand:Rating2 + Brand:NoReviews +
                    Price:Rating2 + Price:NoReviews + Brand:Price:Rating2 +
                    Brand:Price:NoReviews,
                    data=train.data)
summary(full.model.B)

#Reduce model as possible
reduced.model.B <- lm(LRank ~ Brand + Price + Rating2 + NoReviews +
                    Brand:Price + Price:Rating2, data=train.data)
summary(reduced.model.B)
vif(reduced.model.B)
summary(aov(reduced.model.B)) #Type I analysis of effects
anova(reduced.model.B, full.model.B) #Likelihood ratio test
avPlots(reduced.model.B)

MSE.B <- mean(reduced.model.B$residuals^2) #MSE calculation
AIC(reduced.model.B)
par(mfrow=c(2,2))
plot(reduced.model.B) #residuals plots
```

```
#Influence diagnositics
infl.B <- influence.measures(reduced.model.B)
summary(infl.B)
par(mfrow=c(1,3))
cov.B <- infl.B$infmat[,"cov.r"]
plot(cov.B, ylab="Covariance Ratio")
abline(h=1, lty=2)
dffit.B <- infl.B$infmat[,"dffit"]
plot(dffit.B, ylab="DFFITs")
abline(h=0, lty=2)
cookD.B <- infl.B$infmat[,"cook.d"]
plot(cookD.B, ylab="Cook's D")

#Residuals versus reviews volume
par(mfrow=c(1,1))
plot(train.data$LReviewQty, reduced.model.B$residuals, ylab="Standardized
       Residuals", xlab="Reviews Volume ^ (1/4)")
abline(h=0, lty=2)


## Validation of Models with Test Data
#################################################################

#Calculate predicted values
test.A <- predict(reduced.model.A, newdata=test.data)
test.B <- predict(reduced.model.B, newdata=test.data)

#Calculate residuals
resid.A <- test.data$LRank - test.A
resid.B <- test.data$LRank - test.B

#Calculate MSE on new observations
MSE.test.A <- mean(resid.A^2)
MSE.test.B <- mean(resid.B^2)

#Calculate approximate R^2 on new data
SST <- sum(test.data$LRank^2)-sum(test.data$LRank)^2/
             nrow(test.data) #Sum of squares total
R2.A <- 1 - sum(resid.A^2)/SST
R2.B <- 1 - sum(resid.B^2)/SST


## Predicted Value Plots
#################################################################

#Model A: Predicted Sales Rank vs. Rating & Reviews Volume
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Create new observations for predictions
ReviewQty <- rep(seq(0, 400, by=2), times=3)
Rating <- rep(seq(3, 5, by=1), each = 201)
new.vals.A <- data.frame(ReviewQty=ReviewQty, Rating=Rating)
new.vals.A$LReviewQty <- (new.vals.A$ReviewQty)^(1/4)
new.vals.A$NoReviews <- as.numeric(new.vals.A$ReviewQty==0)
new.vals.A$Rating2 <- new.vals.A$Rating
```

```
new.vals.A$Rating2[new.vals.A$NoReviews==1] <- 3

#Predict sales ranks
temp <- predict(reduced.model.A, newdata=new.vals.A)
new.vals.A$Pred.LRank <- temp
new.vals.A$Pred.Rank <- exp(new.vals.A$Pred.LRank)

#Plot sales rank versus reviews volume, grouped by rating
modelA.pred <- ggplot(new.vals.A, aes(x=ReviewQty, y=Pred.Rank, group=Rating)) +
              geom_line(aes(color=as.factor(Rating)), size=1) + theme_bw() +
              labs(x="Customer Reviews Volume", y="Sales Rank in Batteries") +
              theme(legend.position="bottom", legend.key=element_blank()) +
              scale_colour_discrete(name="Customer Review Rating")


#Model B: Predicted Sales Rank vs. Price by Brand & Rating
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Create new observations for predictions
Price <- rep(seq(125, 250, by=5),times=9)
Brand <- rep(rep(c("Odyssey", "Optima", "Exide"), each=26), times=3)
Rating <- rep(c(3,3,5), each=78)
NoReviews <- rep(c(1,0,0),each=78)
new.vals.B <- data.frame(Brand=Brand, Price=Price, Rating2=Rating,
NoReviews=NoReviews)

#Predict new values
temp <- predict(reduced.modelB, newdata=new.vals.B)
new.vals.B$Pred.LRank <- temp
new.vals.B$Pred.Rank <- exp(new.vals.B$Pred.LRank)

#Plot sales rank versus price, grouped by brand for NO REVIEWS
modelB.pred.0 <- ggplot(new.vals.B[new.vals.B$NoReviews==1,], aes(x=Price,
y=Pred.Rank,
              group=Brand)) + geom_line(aes(color=Brand), size=1) + theme_bw() +
              labs(x="Price ($)", y="Sales Rank in Batteries") +
              theme(legend.position="bottom", legend.key=element_blank()) +
              scale_color_brewer(name="Brand", type="qual", palette=3) +
              scale_y_continuous(limits = c(0,5000))

#Plot sales rank versus price, grouped by brand for 3 STAR Rating
modelB.pred.3 <- ggplot(new.vals.B[new.vals.B$Rating==3 & new.vals.B$NoReviews==0,],
              aes(x=Price, y=Pred.Rank, group=Brand)) + geom_line(aes(color=Brand),
              size=1) + theme_bw() + labs(x="Price ($)", y="Sales Rank in Batteries") +
              theme(legend.position="bottom", legend.key=element_blank()) +
              scale_color_brewer(name="Brand", type="qual", palette=3) +
              scale_y_continuous(limits = c(0,5000))

#Plot sales rank versus price, grouped by brand for 5 STAR Rating
modelB.pred.5 <- ggplot(new.vals.B[new.vals.B$Rating==5,], aes(x=Price, y=Pred.Rank,
              group=Brand)) + geom_line(aes(color=Brand), size=1) + theme_bw() +
              labs(x="Price ($)", y="Sales Rank in Batteries") +
              theme(legend.position="bottom", legend.key=element_blank()) +
              scale_color_brewer(name="Brand", type="qual", palette=3) +
              scale_y_continuous(limits = c(0,5000))
```

Appendix D – R Code for Analysis of Customer Reviews

```
load("Comments.RData")
library(tm); library(wordcloud); library(sqldf); library(ggplot2); library(Deducer)
library(RWeka); library(car)

## Text Cleaning & Analysis Functions
#################################################################

##Function yields sorted list of word frequencies in a doc term matrix
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
getWordList <- function(dtm){
#Function that takes in a document text matrix and returns
#a sorted list of word frequency

       words <- colnames(dtm)
       freqs <- apply(dtm, 2, sum)

       word.list <- data.frame(Word=words, Freq=freqs)

       w.order <- sort.int(word.list$Freq, decreasing=TRUE,
                           index.return=TRUE)

       word.list <- word.list[w.order$ix,]
       rownames(word.list) <- 1:nrow(word.list)

       return(word.list)
}
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

##Word stemming function that uses custom dictionary
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
customStemmer <- function(Texts, Dictionary){
       for (i in 1:nrow(Dictionary)){
              Texts <- gsub(Dictionary$Original[i], Dictionary$Replacement[i],
                            Texts, ignore.case=TRUE)
       }
       return(Texts)
}
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

##Number substitutions
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
numsToText <- function(Texts){
       integers <- c(' 1 ',' 2 ',' 3 ',' 4 ',' 5 ',' 6 ',' 7 ',' 8 ',
                     ' 9 ',' 10 ',' 11 ',' 12 ')
       numbers <- c(' one ',' two ',' three ',' four ',' five ',' six ',
                     ' seven ',' eight ',' nine ',' ten ',' eleven ',' twelve ')
       for (i in 1:length(integers)){
              Texts <- gsub(integers[i], numbers[i], Texts,)
       }
       return(Texts)
}
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
##Extract two word phrases (bi-grams)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
gram2Tokenizer <- function(x){
       NGramTokenizer(x, Weka_control(min=2,max=2))
}
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

##Convert bi-grams into single token
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
NGramFixer <- function(Texts, NGrams){
       NGrams.new <- gsub(' ', '_', NGrams)
       for (i in 1:length(NGrams)){
              Texts <- gsub(NGrams[i], NGrams.new[i], Texts,
              ignore.case=TRUE)
       }
       return(Texts)
}
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

## Add a column to a data frame indicating if Flag word is found
## in associated text
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
wordFlagger <- function(data, text, Flag){
       C <- ncol(data) + 1
       inds <- grep(Flag, text)
       data[,C] <- 0
       data[inds, C] <- 1
       names(data)[C] <- gsub(" ", ".", Flag)
       return(data)
}

#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


## Basic Data Preparation & Overall Analysis
#############################################################
##Assign brands
all.comments[grep("Odyssey", all.comments$Prod.Name),"Brand"] <- "Odyssey"
all.comments[grep("Optima", all.comments$Prod.Name),"Brand"] <- "Optima"
all.comments[grep("Exide", all.comments$Prod.Name),"Brand"] <- "Exide"
all.comments[grep("Hawker", all.comments$Prod.Name),"Brand"] <- "Odyssey"

#Convert date to POSIXt
all.comments$Date2 <- as.Date(all.comments$Date, "%B %d, %Y")

#Plot comment frequencies over time
date.hist <- ggplot(all.comments, aes(x=Date2, fill=Brand)) +
             geom_histogram(bins=50) + scale_fill_brewer(
             name="Brand", type="qual", palette=3) + theme_bw() +
             labs(x="Date", y="Frequency of Customer Reviews") +
             theme(legend.position=c(0.12,0.8))

##Reduce to only more recent reviews
comments.recent <- all.comments[all.comments$Date2 >= "2013-01-01",]
```

```
##Summarize selected comments by brand and rating
table(comments.recent$Brand)
table(comments.recent$Rating)
table(comments.recent$Rating, comments.recent$Brand)

##Test for association between brand and rating
pearson <- chisq.test(comments.recent$Rating,comments.recent$Brand)
pearson$observed
pearson$expected
pearson$residuals

G2.likelihood <- likelihood.test(comments.recent$Brand,comments.recent$Rating)

#Test for association between only Odyssey & Optima
pearson2 <- chisq.test(comments.recent$Rating[comments.recent$Brand!="Exide"],
             comments.recent$Brand[comments.recent$Brand!="Exide"])

#Create variable for whether comment was found helpful by anyone
comments.recent$Helpful <- as.numeric(comments.recent$Votes>0)
table(comments.recent$Rating, comments.recent$Helpful)

#Test for relationship between rating and helpfulness
cor.test(comments.recent$Rating, comments.recent$Votes, method="kendall")
cor.test(comments.recent$Rating, comments.recent$Votes, method="spearman")


## Text Processing & Cleaning & Word Cloud Creation
############################################################
#Read-in custom dictionary for stemming
doc.stems <- read.csv("CustomStems.csv")

#Replace punctuation with spaces
comments.recent$Text2 <- gsub("[.!;,?/]", " ", comments.recent$Text)

#Stem reviews
comments.recent$Text2 <- customStemmer(comments.recent$Text2, doc.stems)

#Replace integers
comments.recent$Text2 <- numsToText(comments.recent$Text2)

#Find frequent two word phrases
corpus <- VCorpus(DataframeSource(data.frame(comments.recent$Text2)))
corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removeWords, stopwords("english"))
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)
#Remove brand names & other custom stop words
corpus <- tm_map(corpus, removeWords, c("exide","optima","hawker","odyssey",
                    "optimas", "battery", "batteries", "batterys"))
corpus <- tm_map(corpus, stripWhitespace)

#Tokenize two word phrases and create list of most frequent terms
dtm.2gram <- DocumentTermMatrix(corpus, control=list(tokenize=gram2Tokenizer))
freq.2grams <- removeSparseTerms(dtm.2gram, 0.996)
bigrams <- getWordList(freq.2grams)
```

```
#Output to .csv file for later use
write.csv(bigrams, file="Common2Grams.csv", row.names=FALSE)

##Read in modified N-gram lists (nonsense pairings removed) and replace in reviews
list.2grams <- read.csv("Common2Grams.csv")
comments.modified <- data.frame(Text=unlist(sapply(corpus, '[', "content")),
                                stringsAsFactors=FALSE)
comments.modified$Text2 <- NGramFixer(comments.modified$Text, list.2grams$Word)

#Create new corpus
corpus2 <- VCorpus(DataframeSource(data.frame(comments.modified$Text2)))
#Remove specific product types & other custom stop words
corpus2 <- tm_map(corpus2, removeWords, c("red_top", "yellow_top", "redtop",
                    "yellowtop", "amazon", "one", "car", "will", "two", "three",
                    "four","buy","get","just","can","put","five", "com", "amazons",
                    "use", "also", "say", "said","purchase","vehicle"))

#Create document term matrices
dtm.tf <- DocumentTermMatrix(corpus2)
dtm.tf <- removeSparseTerms(dtm.tf, 0.9995)

##Make some word clouds for different review rating groups
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Create separate document term matrices by review rating
low.ratings <- dtm.tf[comments.recent$Rating<=2, ]
high.ratings <- dtm.tf[comments.recent$Rating>=4, ]
mid.ratings <- dtm.tf[comments.recent$Rating==3, ]

#Create word frequency data frames
low.words <- getWordList(low.ratings)
mid.words <- getWordList(mid.ratings)
high.words <- getWordList(high.ratings)

#Color pallettes for word clouds
low.cols <- brewer.pal(9, "YlOrRd")[-(1:4)]
high.cols <- brewer.pal(9, "PuBuGn")[-(1:4)]
brand.cols <- brewer.pal(9, "BuPu")[-(1:4)]

#Word cloud (by review rating) plotting
wordcloud(words=low.words$Word, freq=low.words$Freq, max.words=75,
             random.order=FALSE, random.color=FALSE, colors=low.cols)
wordcloud(words=mid.words$Word, freq=mid.words$Freq, max.words=75,
             random.order=FALSE, random.color=FALSE, colors=brand.cols)
wordcloud(words=high.words$Word, freq=high.words$Freq, max.words=75,
             random.order=FALSE, random.color=FALSE, colors=high.cols)

##Word clouds by brand
Exide.words <- getWordList(dtm.tf[comments.recent$Brand=="Exide",])
wordcloud(words=Exide.words$Word, freq=Exide.words$Freq, max.words=100,
             random.order=FALSE, random.color=FALSE, colors=brand.cols)

Optima.words <- getWordList(dtm.tf[comments.recent$Brand=="Optima",])
wordcloud(words=Optima.words$Word, freq=Optima.words$Freq, max.words=100,
             random.order=FALSE, random.color=FALSE, colors=brand.cols)
```

```
Odyssey.words <- getWordList(dtm.tf[comments.recent$Brand=="Odyssey",])
wordcloud(words=Odyssey.words$Word, freq=Odyssey.words$Freq, max.words=100,
              random.order=FALSE, random.color=FALSE, colors=brand.cols)


## Analysis of Key Terms in Reviews & Relationship to Brand, Rating
#############################################################
#Create new data frame copy with only key variables
key.comments <- comments.recent[,c(2:7,9:12)]
#Create list of key words
key.words <- c("prius", "dead", "great", "price", "warranty", "great price",
                     "fit perfect", "perfect fit")
#Indicate which reviews have which key words
for (j in 1:length(key.words)){
       key.comments <- wordFlagger(key.comments,comments.modified$Text,
                            key.words[j])
}

#Combine fit perfect and perfect fit variables
key.comments$fit.perfect <- key.comments$fit.perfect + key.comments$perfect.fit
key.comments$fit.perfect[key.comments$fit.perfect>1] <- 1
key.comments <- key.comments[,1:(ncol(key.comments)-1)]

#Determine number of reviews containing each of the key words
apply(key.comments[,11:17], 2, sum)

#Determine average customer review rating for each of the key words
key.words <- gsub(" ", ".", key.words)
m.ratings <- c()
for (j in 1:(length(key.words)-1)){
       m.ratings <- c(m.ratings, mean(key.comments$Rating[key.comments[,
                            key.words[j]]==1]))
}

##Analysis of the prevalence of "Prius"
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Create summarized data set
prius.tab <- sqldf('select Brand, Rating, prius, count(prius) as N
                     from "key.comments" group by Brand, Rating, prius')
prius.tab$Rating2 <- as.factor(prius.tab$Rating)

#Model contingency table with log linear model
prius.fit <- glm(N ~ Brand + Rating2 + prius + Brand:Rating + Rating:prius +
                     Brand:prius, data=prius.tab, family=poisson())
Anova(prius.fit)

#Reduce to final model
prius.fit2 <- glm(N ~ Brand + Rating2 + prius + Brand:Rating +
                     Brand:prius, data=prius.tab, family=poisson())
summary(prius.fit2)
pchisq(prius.fit2$deviance, df=prius.fit2$df.residual, lower.tail=FALSE)
Anova(prius.fit2)
anova(prius.fit2, prius.fit)
table(key.comments$prius, key.comments$Brand)
```

```
table(key.comments$prius, key.comments$Prod.Name)


##Analysis of the prevalence of "Great"
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Create summarized data set
great.tab <- sqldf('select Brand, Rating, great, count(great) as N
                    from "key.comments" group by Brand, Rating, great')
great.tab$Rating2 <- as.factor(great.tab$Rating)

#Model contingency table with log linear model
great.fit <- glm(N ~ Brand + Rating2 + great + Brand:Rating +
                   Rating:great + Brand:great, data=great.tab,
                   family=poisson())
Anova(great.fit)

#Reduce to final model
great.fit2 <- glm(N ~ Brand + Rating2 + great + Brand:Rating +
                   Rating:great, data=great.tab, family=poisson())
summary(great.fit2)
Anova(great.fit2)
pchisq(great.fit2$deviance, df=great.fit2$df.residual, lower.tail=FALSE)
anova(great.fit2, great.fit)
great.tab$fits <- great.fit2$fitted.values

#Odds of great in 5 stars vs 1 star
temp <- great.tab$fits[great.tab$Brand =='Optima' & great.tab$Rating %in% c(1,5)]
great.OR.stars <- temp[1]*temp[4]/(temp[2]*temp[3])


##Analysis of the prevalence of "Price"
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Create summarized data set
price.tab <- sqldf('select Brand, Rating, price, count(price) as N
                    from "key.comments" group by Brand, Rating, price')
price.tab$Rating2 <- as.factor(price.tab$Rating)

#Model contingency table with log linear model
price.fit <- glm(N ~ Brand + Rating2 + price + Brand:Rating +
                   Rating:price + Brand:price, data=price.tab,
                   family=poisson())
summary(price.fit)
Anova(price.fit)
pchisq(price.fit$deviance, df=price.fit$df.residual, lower.tail=FALSE)
price.tab$fits <- price.fit$fitted.values

##Calculate some odds ratios
#Odds of price =  1 in Optima vs Odyssey
temp <- price.tab$fits[price.tab$Brand %in% c('Optima','Odyssey') & price.tab$Rating
== 5]
price.OR.brand <- temp[1]*temp[4]/(temp[2]*temp[3])

#Odds of price in 5 stars vs 1 star
temp <- price.tab$fits[price.tab$Brand =='Optima' & price.tab$Rating %in% c(1,5)]
price.OR.stars <- temp[1]*temp[4]/(temp[2]*temp[3])
```

```
##Analysis of the prevalence of "Great Price"
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Create summarized data set
gp.tab <- sqldf('select Brand, Rating, "great.price", count("great.price")
                    as N from "key.comments" group by Brand, Rating,
                    "great.price"')
gp.tab$Rating2 <- as.factor(gp.tab$Rating)

#Model contingency table with log linear model
gp.fit <- glm(N ~ Brand + Rating2 + great.price + Brand:Rating +
                    Rating:great.price + Brand:great.price, data=gp.tab,
                    family=poisson())
Anova(gp.fit)

#Reduce to final model
gp.fit2 <- glm(N ~ Brand + Rating2 + great.price + Brand:Rating +
                    Rating:great.price, data=gp.tab, family=poisson())
summary(gp.fit2)
Anova(gp.fit2)
pchisq(gp.fit2$deviance, df=gp.fit2$df.residual, lower.tail=FALSE)
anova(gp.fit2, gp.fit)
table(key.comments$great.price, key.comments$Rating)


##Analysis of the prevalence of "Perfect Fit"
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Create summarized data set
fit.tab <- sqldf('select Brand, Rating, "fit.perfect", count("fit.perfect")
                    as N from "key.comments" group by Brand, Rating,
                    "fit.perfect"')
fit.tab$Rating2 <- as.factor(fit.tab$Rating)

#Model contingency table with log linear model
fit.fit <- glm(N ~ Brand + Rating2 + fit.perfect + Brand:Rating +
                    Rating:fit.perfect + Brand:fit.perfect, data=fit.tab,
                    family=poisson())
Anova(fit.fit)

#Reduce to final model
fit.fit2 <- glm(N ~ Brand + Rating2 + fit.perfect + Brand:Rating +
                    Brand:fit.perfect, data=fit.tab, family=poisson())
summary(fit.fit2)
Anova(fit.fit2)
pchisq(fit.fit2$deviance, df=fit.fit2$df.residual, lower.tail=FALSE)
anova(fit.fit2, fit.fit)


##Analysis of the prevalence of "Dead"
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Create summarized data set
dead.tab <- sqldf('select Brand, Rating, dead, count(dead) as N
                    from "key.comments" group by Brand, Rating, dead')
dead.tab$Rating2 <- as.factor(dead.tab$Rating)
```

```
#Model contingency table with log linear model
dead.fit <- glm(N ~ Brand + Rating2 + dead + Brand:Rating + Rating:dead +
                    Brand:dead, data=dead.tab, family=poisson())
Anova(dead.fit)

#Reduce to final model
dead.fit2 <- glm(N ~ Brand + Rating2 + dead + Brand:Rating + Rating:dead,
                    data=dead.tab, family=poisson())
summary(dead.fit2)
Anova(dead.fit2)
pchisq(dead.fit2$deviance, df=dead.fit2$df.residual, lower.tail=FALSE)
anova(dead.fit2, dead.fit)
dead.tab$fits <- dead.fit2$fitted.values

#Odds of dead = 1 in 1 stars vs 5 star
temp <- dead.tab$fits[dead.tab$Brand=='Optima' & dead.tab$Rating %in% c(1,5)]
dead.OR.stars <- temp[2]*temp[3]/(temp[1]*temp[4])


##Analysis of the prevalence of "Warranty"
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Create summarized data set
warr.tab <- sqldf('select Brand, Rating, warranty, count(warranty) as N
                    from "key.comments" group by Brand, Rating, warranty')
warr.tab$Rating2 <- as.factor(warr.tab$Rating)

#Model contingency table with log linear model
warr.fit <- glm(N ~ Brand + Rating2 + warranty + Brand:Rating + Rating:warranty +
                    Brand:warranty, data=warr.tab, family=poisson())
Anova(warr.fit)

#Reduce to final model
warr.fit2 <- glm(N ~ Brand + Rating2 + warranty + Brand:Rating +
                    Rating:warranty, data=warr.tab, family=poisson())
summary(warr.fit2)
Anova(warr.fit2)
pchisq(warr.fit2$deviance, df=warr.fit2$df.residual, lower.tail=FALSE)
anova(warr.fit2, warr.fit)
warr.tab$fits <- warr.fit2$fitted.values

#Odds of warranty = 1 in 1 stars vs 5 star
temp <- warr.tab$fits[warr.tab$Brand=='Optima' & warr.tab$Rating %in% c(1,5)]
warr.OR.stars <- temp[2]*temp[3]/(temp[1]*temp[4])


##Analysis of the prevalence of "Prius" & "Perfect Fit"
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Create summarized data set
priusfit.tab <- sqldf('select Brand, prius, "fit.perfect", count("fit.perfect")
                    as N from "key.comments" group by Brand, prius,
                    "fit.perfect"')

#Model contingency table with log linear model
priusfit.fit <- glm(N ~ Brand + prius + fit.perfect + Brand:prius +
```

```
                    prius:fit.perfect + Brand:fit.perfect, data=priusfit.tab,
                    family=poisson())
Anova(priusfit.fit)

#Reduce to final model
priusfit.fit2 <- glm(N ~ Brand + prius + fit.perfect + Brand:prius +
                    prius:fit.perfect, data=priusfit.tab,
                    family=poisson())
summary(priusfit.fit2)
Anova(priusfit.fit2)
pchisq(priusfit.fit2$deviance, df=priusfit.fit2$df.residual,
             lower.tail=FALSE)


## Analysis of Time Mentions in Reviews
#############################################################
#Create new data frame with only key columns reserved
time.comments <- comments.recent[,c(2:7,9:12)]
time.words <- c('one years','two years','three years','four years',
             'five years', 'six years', 'seven years','eight years',
             'nine years', 'ten years', 'eleven years','twelve years')
#Initialize number of years to 0; 0 == no specific time metioned
time.comments$Years <- 0

#Iterate through possible numbers of years and update Years variable accordingly
for (i in 1:length(time.words)){
       inds <- grep(time.words[i], comments.modified$Text)
       time.comments$Years[inds] <- i
}

#Create binary variable to indicate if specific number of years mentioned
time.comments$Mention <- as.numeric(time.comments$Years>0)

#Examine contingency tables
table(time.comments$Years)
table(time.comments$Mention)
table(time.comments$Mention, time.comments$Rating)
table(time.comments$Mention, time.comments$Brand)

#Create a new data frame with only reviews which mention time
time.comments2 <- time.comments[time.comments$Mention==1,]

##Model time with OLS
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
full.fit <- lm(Years ~ Brand + Rating + Brand:Rating, data=time.comments2)
boxCox(full.fit) #Indicates square root transformation

#Create a variable for square root of Years
time.comments2$Years2 <- sqrt(time.comments2$Years)
time.fit <- lm(Years2 ~ Brand + Rating, data=time.comments2)
summary(time.fit)
vif(time.fit)
anova(time.fit))
#Reisduals analysis
par(mfrow=c(2,2))
```

```
plot(time.fit)

##Create predicted value charts
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#Initialize new variables
Rating <- rep(1:5, times=3)
Brand <- rep(c("Exide", "Odyssey", "Optima"), each=5)
predix <- data.frame(Brand=Brand, Rating=Rating)
temp <- predict(time.fit, newdata=predix)
predix$Years <- temp^2

#Plot predicted values
pred.duration <- ggplot(predix, aes(x=Rating, y=Years, group=Brand)) +
             geom_line(aes(color=Brand),size=1) +
             theme_bw() + labs(x="Rating (stars)", y="Duration (Years)") +
             theme(legend.position="bottom", legend.key=element_blank()) +
             scale_colour_brewer(name="Brand", type="qual", palette=3)
```