Statistics 133 Project Final Report:
An Analysis of the Factors that
Affect the Helpfulness of an Amazon Review

Group Prime: Namrata Das, Vaibhav Ramamoorthy, Salil Vanvari, Jonathan Xu

1. Introduction

With online shopping becoming more popular because of its ease and convenience, users are researching their purchases more than ever before committing. Along with word of mouth, reading online reviews is quite important to many users because it can mean saving money that would have been wasted on a poor quality item. The power of crowdsourcing in reviews has become extremely popular and it can be argued that the system of online reviews have helped online vendors thrive in a competitive online market (for example Etsy). If a product has many reviews; however, this creates another problem for the user: the user now needs to evaluate whether the review was actually **helpful** to read.

In this project, we analyze the data collected from Amazon's food reviews. With this dataset, we examine the following questions:

1. When users look at reviews, what types of reviews do they find helpful?
2. How do things like the score rating of a review, word length of review, or specific words within the review affect a review's helpfulness?

To answer these two questions, we looked at the multitude of Amazon's food reviews to see if there were certain trends that made some food reviews more popular to read than others.

To analyze our key questions, we looked at all the components of a review (i.e. length of the review, the score given, the words used) to see how they influenced the helpfulness of the review. In the Amazon review system, a user can either vote positively or negatively on a review to show if the review was helpful or not. Taking this into consideration, we analyzed these components against the calculated helpfulness. To visualize the results, we used scatter plots (along with boxplots) to characterize the relation between helpfulness and major components of a review.

When viewing food product reviews, the "most helpful" reviews are often displayed at the top and our results showed common trends in these helpful reviews across different products. In a way, these reviews could be seen as the common sentiment of the community towards the product. In this sense, the results we discovered showed important biases and predilections that many consumers have while shopping for food online.

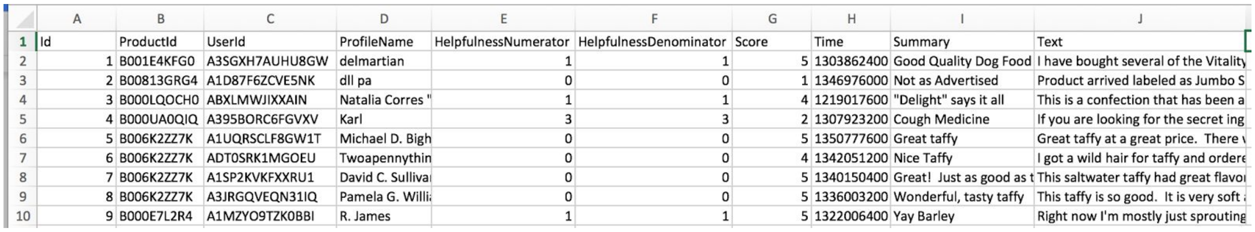
2. Dataset Description

The dataset that we worked with was taken from Kaggle.com and it consists of 568,454 reviews of food items written by Amazon users from October 1999 to October 2012 (Amazon Fine Foods, Kaggle). Being college students, we saw this dataset had the potential to be incredibly relevant to us. However, we also know the conflicts that can arise when we have too many choices, as stated by Barry Schwartz in his book The Paradox of Choice (The Paradox of Choice, Wikipedia).

To alleviate this stress from choosing the perfect product, this is where reading online reviews come into play. Keeping this in mind, we wanted to analyze these reviews to see what users thought about before placing an order. Upon downloading and unzipping the large 250 MB file, we fortunately gained access to a .csv file that was tidy and relatively ready to use. Analyzing this data further, we are given 10 columns that each represent something unique about the review:

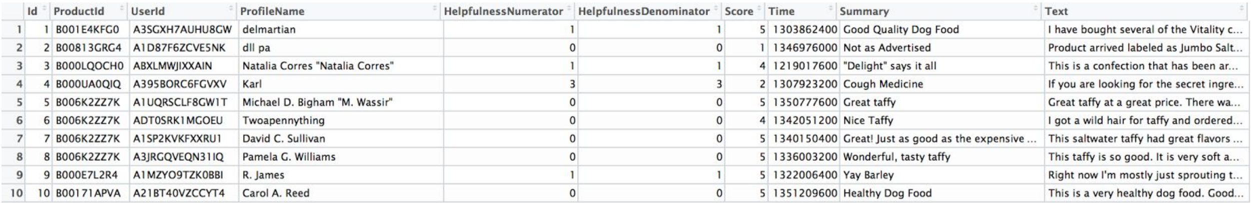
- **`Id`:** the actual id of that particular review in our dataset
- **`ProductId`:** Amazon's unique identifier for each product
- **`UserId`:** the unique identifier for the Amazon user who left the review
- **`ProfileName`:** the alphanumeric name for the Amazon user who left the review
- **`HelpfulnessNumerator`:** the number of users who marked the review as helpful
- **`HelpfulnessDenominator`:** the number of users who marked the review as either helpful or unhelpful
- **`Score`:** a rating from 1-5 given by the Amazon user who left the review
- **`Time`:** the timestamp of the review
- **`Summary`:** a brief summary provided by the Amazon user who left the review
- **`Text`:** the content of the review left by the Amazon user who left the review

Here is the .csv file in Excel:



	A	B	C	D	E	F	G	H	I	J
1	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
2	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality
3	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo S
4	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "	1	1	4	1219017600	"Delight" says it all	This is a confection that has been a
5	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ing
6	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigh	0	0	5	1350777600	Great taffy	Great taffy at a great price. There \
7	6	B006K2ZZ7K	ADT0SRK1MGOEU	Twoapennythin	0	0	4	1342051200	Nice Taffy	I got a wild hair for taffy and ordere
8	7	B006K2ZZ7K	A1SP2KVKFXXRU1	David C. Sullivai	0	0	5	1340150400	Great! Just as good as t	This saltwater taffy had great flavoi
9	8	B006K2ZZ7K	A3JRGQVEQN31IQ	Pamela G. Willi;	0	0	5	1336003200	Wonderful, tasty taffy	This taffy is so good. It is very soft ;
10	9	B000E7L2R4	A1MZYO9TZK0BBI	R. James	1	1	5	1322006400	Yay Barley	Right now I'm mostly just sprouting

Here is the data table in RStudio:



	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
1	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality c...
2	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salt...
3	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been ar...
4	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingre...
5	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There wa...
6	6	B006K2ZZ7K	ADT0SRK1MGOEU	Twoapennything	0	0	4	1342051200	Nice Taffy	I got a wild hair for taffy and ordered...
7	7	B006K2ZZ7K	A1SP2KVKFXXRU1	David C. Sullivan	0	0	5	1340150400	Great! Just as good as the expensive ...	This saltwater taffy had great flavors ...
8	8	B006K2ZZ7K	A3JRGQVEQN31IQ	Pamela G. Williams	0	0	5	1336003200	Wonderful, tasty taffy	This taffy is so good. It is very soft a...
9	9	B000E7L2R4	A1MZYO9TZK0BBI	R. James	1	1	5	1322006400	Yay Barley	Right now I'm mostly just sprouting t...
10	10	B00171APVA	A21BT40VZCCYT4	Carol A. Reed	0	0	5	1351209600	Healthy Dog Food	This is a very healthy dog food. Good...

We will proceed to clean and wrangle this data.

3. Data Cleaning and Wrangling

To make the data frame glyph-ready, we took several crucial data-cleaning steps. The first step was to clean up the `reviews` data table itself to ensure that all of the columns were formatted properly, with the right object class. We performed the following manipulations:

- Converted the `time` variable column from numeric format to `POSIXct`, which calculated the date and time of each entry of the column based on the number, which represented the number of seconds after 1/1/1970.
- Renamed the columns to make them more intuitive for our analysis.
- Added a column called `unhelpful`, which represents the difference between the total number of helpfulness ratings and the number of ratings that marked the review as helpful.
- We noticed that some reviews were unfortunately duplicates, with the exact same timestamp and text and summary. To eliminate these, we first needed to remove the `Id` column from the data frame and then filter out the duplicate cases. Then, we put back the `Id` column to ensure that it remained in the data frame.

Additionally, we needed to create some other manipulated data tables to create some of our graphs. To create the table for graph 4.1 (`text_count`), we used the `stri_count()` function from the `stringi` package to count the number of words in the text of each review, using the regex `"\\S+"` as the delimiter. We mutated the table to add this as a column in the data frame and called this variable `count`.

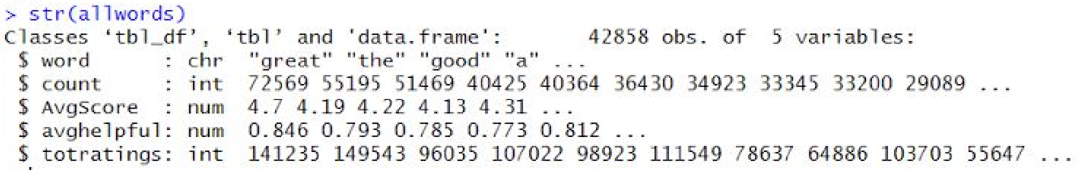
For all of the `graphs,` we needed to plot the `helpfulness` fraction of each review, so we created a new column in those instances that divided the number of `helpful` votes by the number of total votes to get the `helpfulness` for each case.

Finally, we performed textual analysis of the `summary` variable that required significant data wrangling, outlined below:

- We created a smaller data table called "`small_reviews`" that included only the columns for `Id, Helpful, Unhelpful, TotalHelpfulnessRatings, Score, Time,` and `Summary`.
- We applied the `strsplit()` function from the `stringr` package, which separated each word from each `summary` in the data table and put the words into a list, with the first word in each review `summary` in the first vector of the list and so on until the 42nd word in each review `summary` (for those that had 42 - which very few did).
- We converted that list to a matrix, a step that was necessary to eventually make the list into a data frame.
- We then converted that matrix into a data frame, ensuring that all strings were taken is as characters and not factors.
- Finally, we used the `cbind()` function to combine the data frame of `summary` words with the original data frame, such that there were 42 additional columns to the `small_reviews` data frame, each with the words of the `summary` separated into columns. We called this data frame `wordsplit`.

We noticed that many of the words in wordsplit contained extraneous punctuation or were not uniform in case, so we wanted to smooth out these disparities. As such, we wrote a function `wordclean()` that removed anything that wasn't an alphanumeric from each string (using `gsub()`) and converted all letters to lowercase. We ran this on the entire data frame (which took around 4 hours) and then the data frame was clean!

We still need to manipulate the data frame such that each case was each word that appeared in summaries of reviews, so certain additional steps were undertaken. We used the `gather()` and `summarize()` functions multiple times on `wordsplit` to switch the orientation of the table and calculate the `count` of each word, the average score of reviews containing each word, the average `helpfulness` fraction (`Helpful/TotalHelpfulRatings`) of reviews containing each word, and the sum of the number of helpfulness ratings given for all the reviews containing that word. All of these separate data tables containing each of these metrics were joined into one neat data table called `allwords`, with the following structure:



```
> str(allwords)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':	42858 obs. of  5 variables:
 $ word      : chr  "great" "the" "good" "a" ...
 $ count     : int  72569 55195 51469 40425 40364 36430 34923 33345 33200 29089 ...
 $ AvgScore  : num  4.7 4.19 4.22 4.13 4.31 ...
 $ avghelpful: num  0.846 0.793 0.785 0.773 0.812 ...
 $ totratings: int  141235 149543 96035 107022 98923 111549 78637 64886 103703 55647 ...
```

As you can see, this data table contained all of the words that were present in the summaries of the `reviews` across all reviews in our original data set. We now wanted to isolate only food words, to observe and identify trends regarding those. We identified a food word list online and converted that to .csv before importing it as a data frame to R.

We filtered the `allwords` data table to include only food words and called this table `foodwords_data.` We added a column to `foodwords_data` called `posneg` that identified whether the word was a positive one or a negative one based on its average score. If the average score was above 4, it was positive and if not, it was negative. For graphing purposes, we further filtered this table to include the 40 most common positive words and the 40 most common negative words. This data frame, which we called `small_foodword_data,` was the one we ultimately used for our last graph.

4. Analysis and Data Visualization

We analyzed this Kaggle dataset along with our own tables created (as shown above) to answer our two key questions, (1) When users look at reviews, what types of reviews do they find helpful? (2) How do things like the score rating of a review, word length of review, or specific words within the review affect a review's helpfulness?

We broke down our two key questions into smaller, more pointed queries, which would help us analyze the food reviews to a finer granularity. The four questions we looked at were:

1. How does the length of the review affect the helpfulness of that review?
2. How does the number of votes that the review received affect its helpfulness?
3. How does the rating of the product (given by the review) affect that review's helpfulness?
4. Do certain words used in the review affect its helpfulness?

With these four questions, we further wrangled the tables and then created visualizations to help us answer our key questions. Each visualization helped us realize trends in the most helpful reviews.

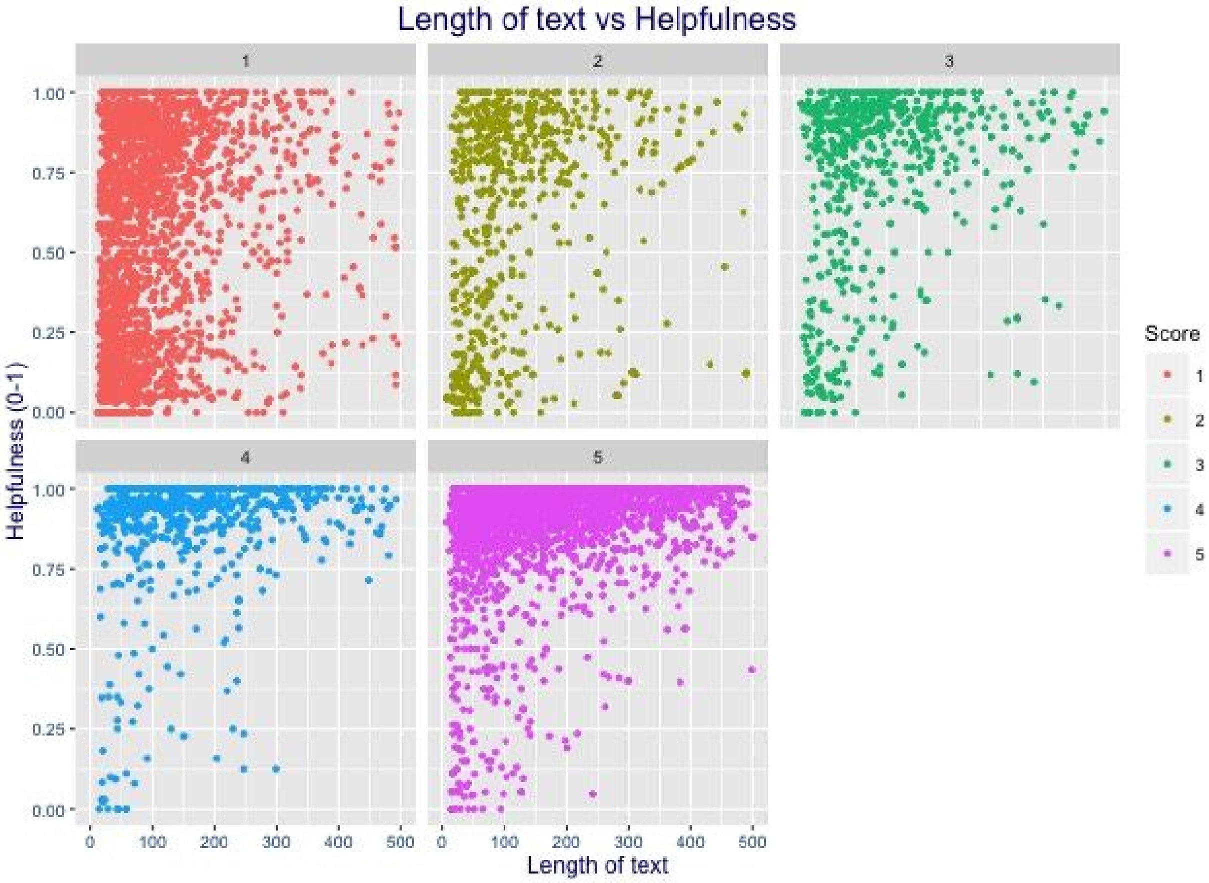
4.1 Length of Review vs. Helpfulness

1. Rationale for Question

   An important correlation we were looking for in our data was how the length of the review affected how helpful the review was. Initially before checking the data, our assumption was that on average, the longer reviews would be more helpful than the shorter reviews, simply because these longer reviews would have more details. With more details, the user would be able to make a more informed decision. We assumed more raw text was indicative of a greater quality of the reviews.

2. Graph Visualization

   To create this graph, we simply mapped the `helpfulness` fraction of the review to its length in words. We faceted the visualization by different scores given by the reviewer to also see if score also played an effect on the reviews. Finally we pruned the scores to only have scores with greater than 15 people having voted on its helpfulness. This way, removed the noise in the data that would be present otherwise.



3. Analysis

A very distinct thing we noticed was that the score heavily affected whether or not the text was actually useful or not. For example, if you look at the graph with the review scores of one, a very clear insight is that that regardless of whether or not you wrote a lot of text, the chance that the text was actually helpful to those looking to buy products was about uniform. This means that if you gave a score of one, no matter whether you wrote a 500 word essay or less than 100 words, there was no guarantee that your review would be viewed as helpful.

To rationalize that, first it is important to think about why a product would receive a score of one. More likely than not, scores of one are given when a user has found something very wrong with the product or the product has frustrated the user. This could mean that a lot of the reviews including the long ones, are actually just really long rants about the product and those are less likely to be helpful.

On the other hand, looking at the graphs with reviews between 4 and 5 shows a very different scenario. Notice that in the graph for 4, reviews that are larger than 300 words are almost all more than .75 helpful. This goes to show that if you give a four and you write more than 300 words to justify your review, there is a decent chance others will find your review helpful to read. This fact is also very similarly seen in the graph where reviewers gave a rating of 5. One possible idea of why this may be true is that on average, those who are reading reviews of a product are anyway looking to buy the product but just want to confirm that it is in fact a good product. In seeking this approval, they will probably find reviews with thought explanations about why a given product is good to be very helpful and thus they would probably prefer to read those reviews.

Overall though, our general hypothesis that just raw walls of text make for good reviews is incorrect. Instead it is nuanced by the score which divides the reviews up and changes their overall helpfulness.

4.2 Number of Votes vs. Helpfulness

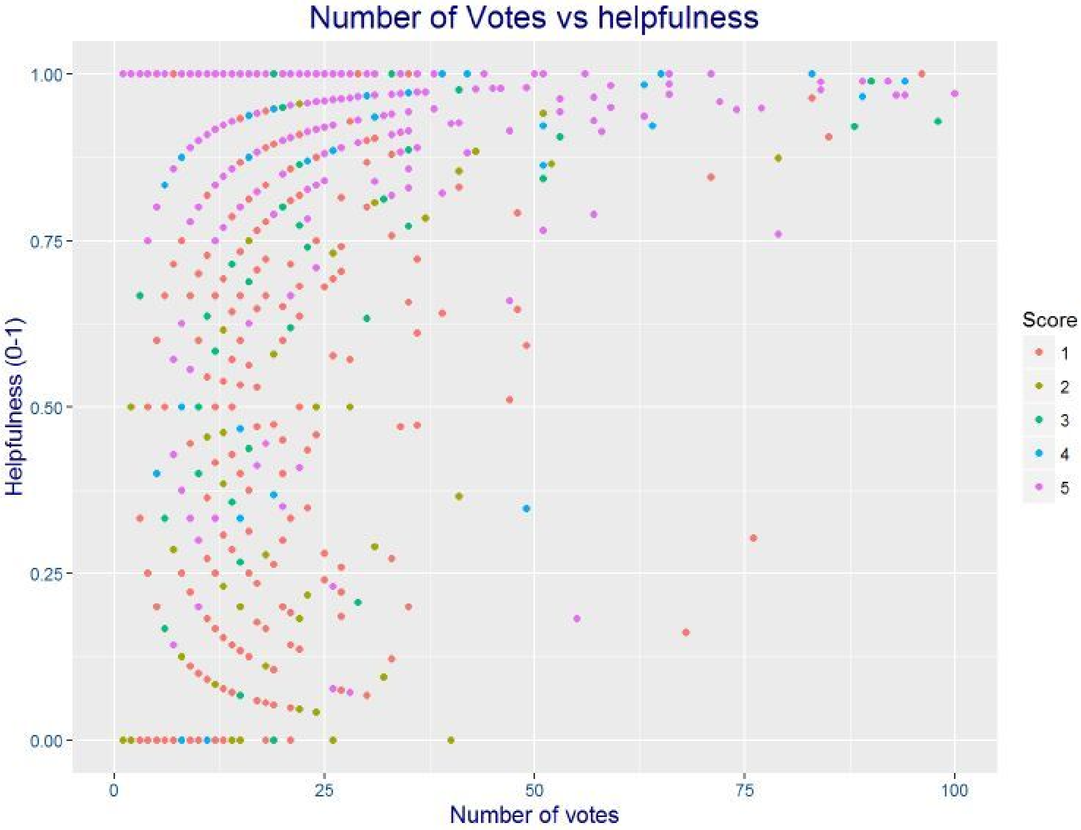
1. Rationale for Question

   For a review to be deemed helpful overall, it is better to have a wider range of distribution of negative and positive votes. With a few number of votes, the calculated helpfulness of a review can be heavily skewed. In this question, we looked at how the number of total votes (whether it be positive or negative) influenced how helpful a review is.

   Our initial assumption was that reviews with more votes would have a calculated helpfulness that would be around 0.5, since there would be a wider range of positive or negative votes. With this belief, we also assumed that reviews with a lower number of votes would be more skewed towards the polar ends on our scale of 0.0 to 1.0

2. Graph Visualization

   To create the graph, we mapped the `helpfulness` fraction to the number of votes that review received. Calculating the helpfulness fraction was simple: divide the number of positives votes by the total number of votes it received. In order to keep our graph readable, we also colored each review with the score it had received.



3. Analysis

The graph created above shows some very interesting trends and thus changed our initial hypotheses. We can see that most of the points congregate between the 0 - 25 bucket. This simply shows that most products have about 0 to 25 total ratings. However, this bucket also had the most variance in the helpfulness factor. The fractions for these reviews were all over the 0.0 – 1.0 scale. Keeping this in mind, if we compare this bucket to the other buckets, such as 50 – 75 and 75 – 100 votes, we see that these two buckets do not have as much variety in the helpfulness fraction. Moreover, the fraction range for the products in these two buckets is consistently within the upper quartile (with the exception of a few outliers explained later).

This graph shows that if a review has more votes, the community of users will generally deem that review to be quite helpful. With this analysis, our initial hypothesis was proven wrong. We believe this result could be a part of the bandwagon effect: if a user sees that many people already voted positively on a review, he or she will also vote positively despite the user (possibly) having conflicting beliefs (Bandwagon Effect, Wikipedia).

4.3 Score vs. Helpfulness

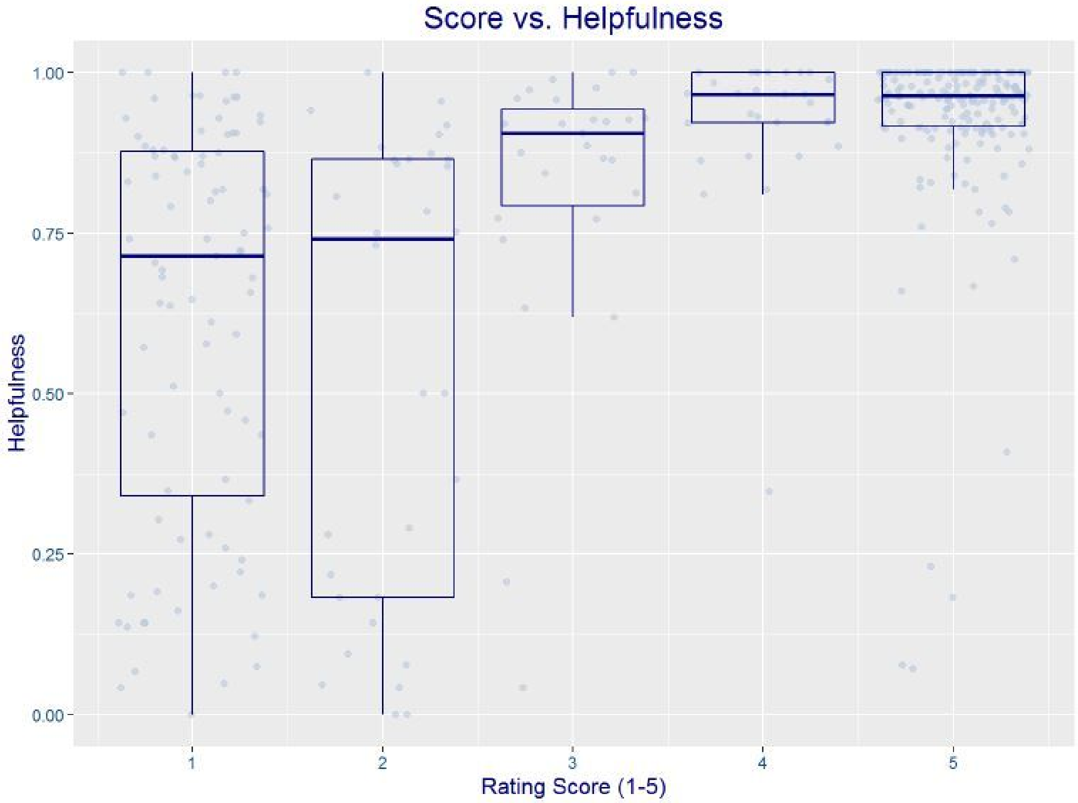
1. Rationale for Question

In analyzing this question, we wanted to see whether certain scores were more helpful to other users and to potentially analyze the rationale behind this. Before we analyzed this question, we all thought about why certain people would leave reviews. Typically, the reviews are polarizing: either someone really enjoys the product or someone really hates the product and wants to write a great or scathing review.

Following that, we thought about how Amazon algorithmically chooses to display its products. Its best-sellers and most popular products often have either the highest scores or the most number of reviews, and most people inherently gravitate towards these scores. Therefore, we hypothesized that there would be the narrowest spread for reviews that were given a score of 1 or 5 because these would be very helpful, and we sought to prove this through our graph visualization.

2. Graph Visualization

To create the graph, we mapped the `helpfulness` fraction to the score that the review received. Calculating the `helpfulness` fraction was simple: divide the number of positives votes by the total number of votes it received. To simplify the observation and analysis, we used the boxplot feature within ggplot on the Rating Score.



3. Analysis

We noticed that this graph actually disputes our claim that there would be a narrow spread for both scores 1 and 5. For score 1, the graph has the second largest spread and was therefore considered helpful and not helpful by a relatively equal number of people. For the score 2, the spread was even larger. However, for scores 3-5, the spread progressively goes down and is more concentrated within the 25th to 75th percentiles. This means that for the higher rated scores, more people agreed on the helpfulness of the review than they did for the lower rated scores. Even though this contradicted our claim, we realized an interesting phenomenon.

The results of the above graph actually point to an interesting phenomenon within the commercial world known as the **confirmation bias**. (Confirmation bias, Wikipedia) This means that when people look at products they like and are willing to purchase, they try to find different reasons to support their cause. In this case, consumers on Amazon browsed products that they enjoyed and were interested in purchasing, and after they read the reviews for each of the products, they found that the ones that rated the product highly were most helpful. This is a possible explanation for the small spread within the higher scores because each time a consumer read these high reviews, they noticed statements that reaffirmed their belief in the product and were most helpful in allowing them to make their decision.

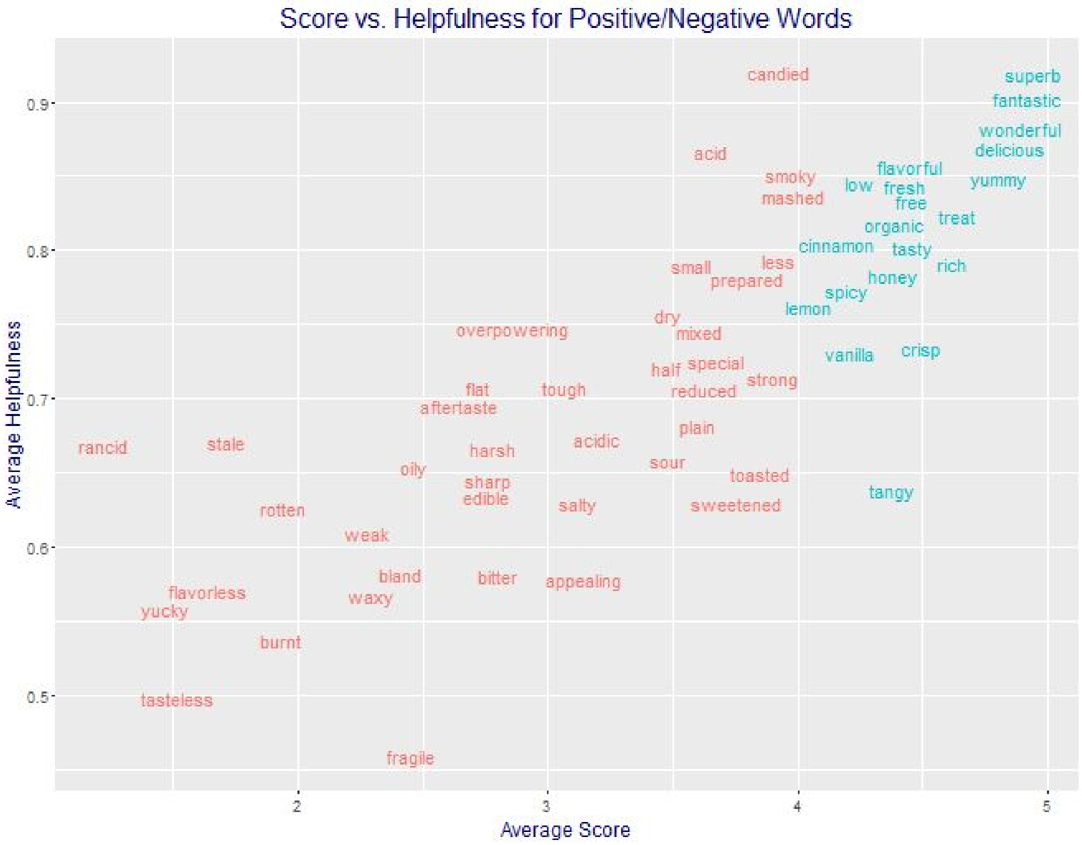
There may be other explanations for this phenomenon aside from the confirmation bias, but this data contradicted our initial hypothesis. Instead, it provides evidence that people tend to disagree on the helpfulness of reviews with lower scores and that people tend to agree on the helpfulness of reviews with higher scores.

4.4 Popular Food Words

1. Rationale for Question

We wanted to identify certain food words that did not follow the trend we had observed earlier, where score and helpfulness were pretty clearly positively associated. Findings words that were exceptions to this general association might give us a hint as to what kind of words make reviews interesting.

2. Graph Visualization



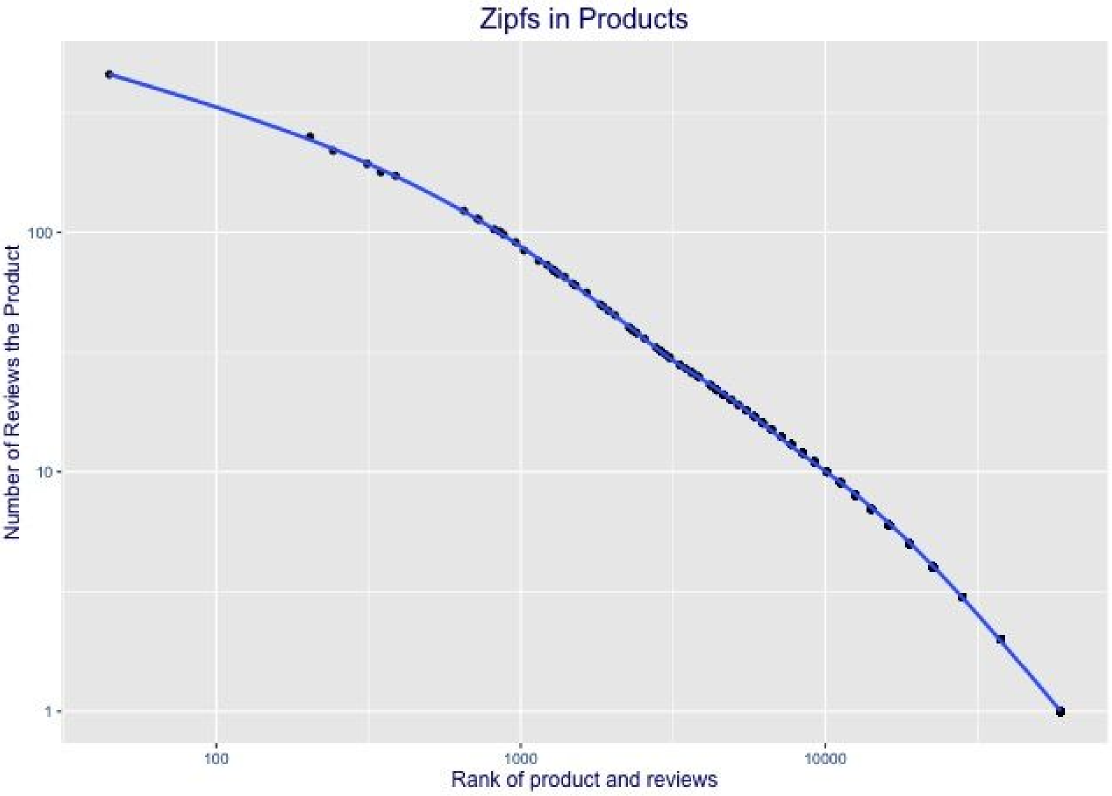
3. Analysis

The graph reinforces our notion earlier of the strong association between score and helpfulness, in this instance for words as the cases instead of reviews. Still, some words jump out interestingly enough. For example, the word "rancid", while obviously associated with lowly scored reviews, seems to be a rather helpful one. This may be because the word is specific to foods with poor smells, and is thus quite descriptive despite its negative connotation.

5. Other Interesting Observations

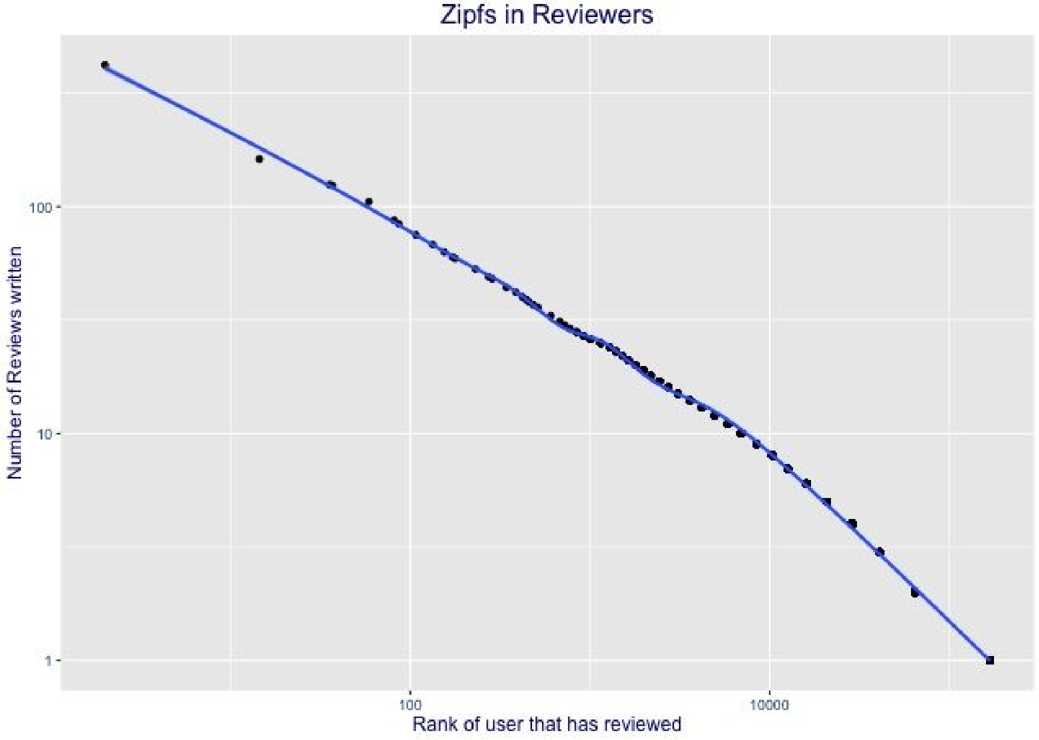
5.1 Zipf's Law

Another unique and interesting thing we found about our data set is that number of reviews to different products as well as the number of reviews written by a person followed a Zipfian distribution. First let's start with how the number of reviews of different products followed a Zipfian distribution. To be specific, we were first simply looking at the different products to see their respective popularities based on the number of reviews for the products. What we noticed was slightly strange, in that many products had a relatively small number of reviews but a few products had a really large number of reviews. After plotting and scaling a graph we arrived at the following visualization.



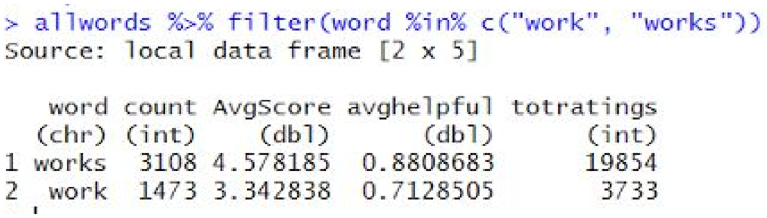
The graph above indicates that in general, the most popular products had a lot of reviews but the less popular products had far fewer reviews. There were hundreds of products that only had a single review whereas there were very few products that had hundreds of reviews. The graph above shows a sample from the 40,000+ products in our dataset and is also in log base 10 for both number of reviews as well as the Rank of the Products. A cool insight from this is that the the top 20% of the products had roughly 80% of the reviews.

Similarly, we also found a very nice Zipfian distribution in the number of reviews written by different reviewers. The top few reviewers wrote around 400 reviews while the next hundred all wrote less than 100 reviews each. Again this was a very clearly idea that the top 20% of reviewers comprised of around 80% of the reviews on these food reviews.



5.2 Unique words that did not follow the trend

One unique discovery we made was that the words “work” and “works” have completely different positivities and negatives:



```
> allwords %>% filter(word %in% c("work", "works"))
Source: local data frame [2 x 5]

   word count AvgScore avghelpful totratings
  (chr) (int)    (dbl)      (dbl)      (int)
1 works  3108 4.578185  0.8808683      19854
2  work  1473 3.342838  0.7128505       3733
```

This is quite interesting; the word with an ‘s’ at the end has a significantly higher `AvgScore` and `avghelpful` than without. Though surprising, this can be explained rather easily. The word “work” likely appears only in summaries in a negative context, as in “this product didn’t work” or “I can’t get this to work”; on the other hand, the plural form likely appears in a more positive context more of the time, in phrases like ”this product works great!” and “works like a charm.” Therefore, the discrepancy in scores and ratings can be explained by simple grammatical rules and intuitive explanations.

6. Conclusion

Overall across this report, we found many interesting trends about helpfulness as it relates to reviews of Amazon products. Several factors affect how helpful a review is including score, length, and specific words that appear within the review. All of these factors that we have analyzed can now be used by future reviewers to write better reviews for the different products in order for the community to have a better idea of what the products are like. From all of this data and analysis though, in order to make the guide better and broader, we would love to do similar analysis on broader sets of Amazon data.

In this report specifically we only had access to food reviews, and while they had a telling tale, some trends may differ given a different set of reviews from other departments. Furthermore, if we had similar data from different services besides Amazon that allow for users to review products we could find macro trends that extend beyond just Amazon and our report may have been able to provide a proper guide to writing the most helpful reviews. One reason that the certain trends of Amazon may not extend to the totality of reviews might include differences including Amazon filtering out reviews that are not helpful or Amazon promoting certain products by showing more good reviews than bad reviews. A larger and more diverse dataset would definitely help to comprehensively create a guide of best practices of writing perfect reviews.

7. Works Cited

"Amazon Fine Food Reviews | Kaggle." Amazon Fine Food Reviews | Kaggle. Web. 09 May 2016. <https://www.kaggle.com/snap/amazon-fine-food-reviews>.

"Bandwagon Effect." Wikipedia. Wikimedia Foundation. Web. 09 May 2016. <https://en.wikipedia.org/wiki/Bandwagon_effect>.

"Confirmation Bias." Wikipedia. Wikimedia Foundation. Web. 09 May 2016. <https://en.wikipedia.org/wiki/Confirmation_bias>.

"The Paradox of Choice." Wikipedia. Wikimedia Foundation. Web. 09 May 2016. <https://en.wikipedia.org/wiki/The_Paradox_of_Choice>.

8. Appendix

```
library(DataComputing)
library(data.table)
library(stringr)
library(stringi)

# Importing and Cleaning Data Table

  ## .csv file was downloaded from https://www.kaggle.com/snap/amazon-fine-food-reviews

rev <- read.csv("Reviews.csv")

clean <- function(table) {
  table$Time <- as.POSIXct(table$Time, origin="1970-01-01")
  names(table) <- c("Id", "ProductId", "UserId", "ProfileName", "Helpful", "TotalHelpfulRatings",
"Score", "Time", "Summary", "Text")
  table <- table %>%
    mutate(Unhelpful = TotalHelpfulRatings - Helpful) %>%
    select(Id, ProductId, UserId, ProfileName, Helpful, Unhelpful, TotalHelpfulRatings, Score, Time,
Summary, Text)
  table <- unique(select(table, -Id))
  table <- table %>%
    mutate(Id = 1)
  table$Id <- seq.int(from=1, to=nrow(table))
  setcolorder(table, c("Id", "ProductId", "UserId", "ProfileName", "Helpful", "Unhelpful",
"TotalHelpfulRatings", "Score", "Time", "Summary", "Text"))
  return(table)
}

reviews <- clean(rev)

# Graphs

  ## Graph 1: Word Count of Review vs. Helpfulness

text_count <- reviews %>%
  mutate(count = stri_count(Text, regex = "\\S+"))
text_count$Score <- as.factor(text_count$Score)
text_count %>%
  mutate(helpfulness=Helpful/TotalHelpfulRatings) %>%
  ggplot(aes(x = count, y = helpfulness)) +
  geom_point(aes(color= Score), size = 3) +
  xlim(0,500) +
  labs(title="Length of text vs Helpfulness", x="Length of text", y="Helpfulness (0-1)") +
labs(linetype='Score of review') +
  theme(
    axis.title = element_text(size=12, color='navyblue'),
    plot.title = element_text(size=16, color='navyblue'),
    axis.text = element_text(color='dodgerblue4')
  )
```

```r
  ## Graph 2: Number of Votes vs Helpfulness

temp <- reviews %>%
  mutate(helpfulness=Helpful/TotalHelpfulRatings)
temp$Score <- as.factor(temp$Score)

temp %>%
  ggplot(aes(x=TotalHelpfulRatings, y=helpfulness)) +
  geom_point(aes(color=Score)) +
  xlim(0, 100) +
  labs(title="Number of votes vs. Helpfulness", x="Number of votes given", y="Helpfulness (0-1)") +
  theme(
    axis.title = element_text(size=12, color='navyblue'),
    plot.title = element_text(size=16, color='navyblue'),
    axis.text = element_text(color='dodgerblue4')
  )

  ## Graph 3: Score vs. Helpfulness

reviews %>%
  filter(TotalHelpfulRatings > 20) %>%
  mutate(helpfulness=Helpful/TotalHelpfulRatings) %>%
  ggplot(aes(x=Score, y=helpfulness)) +
  geom_jitter(alpha=0.5, color='lightsteelblue') +
  geom_boxplot(aes(group=Score), color='navyblue', fill=NA, outlier.color=NA) +
  labs(x="Rating Score (1-5)", y="Helpfulness", title="Score vs. Helpfulness") +
  theme(
    axis.title = element_text(size=12, color='navyblue'),
    plot.title = element_text(size=16, color='navyblue'),
    axis.text = element_text(color='dodgerblue4')
  )

  ## Graph 4: Score vs. Helpfulness for Positive/Negative Words

foodwords <- read.csv("foodwords.csv", header=FALSE)
names(foodwords) <- c("word")

small_reviews <- reviews %>% select(Id, Helpful, Unhelpful, TotalHelpfulRatings, Score, Time,
Summary)
ls <- tstrsplit(small_reviews$Summary, split=" ")
mat  <- matrix(unlist(ls), ncol=length(ls), byrow=FALSE)
words <- as.data.frame(mat, stringsAsFactors = FALSE)
wordsplit <- cbind(small_reviews, words)

wordclean <- function(word) {
  g <- gsub("[^[:alnum:]]", "", word)
  return(tolower(g))
}
```

```r
for(i in 8:ncol(wordsplit)) {
  wordsplit[,i] <- sapply(wordsplit[,i], wordclean)
}

commons <- wordsplit %>%
  gather(key=col, value=word, -Id, -Helpful, -Unhelpful, -TotalHelpfulRatings, -Score, -Time,
-Summary) %>%
  group_by(word) %>%
  summarize(count=n()) %>%
  filter(! is.na(word)) %>%
  arrange(desc(count))

avgscores <- wordsplit %>%
  gather(key=col, value=word, -Id, -Helpful, -Unhelpful, -TotalHelpfulRatings, -Score, -Time,
-Summary) %>%
  group_by(word) %>%
  summarize(AvgScore=mean(Score)) %>%
  filter(! is.na(word)) %>%
  arrange(desc(AvgScore))

avghelpful <- wordsplit %>%
  mutate(H_percent=Helpful/TotalHelpfulRatings) %>%
  filter(! TotalHelpfulRatings==0) %>%
  gather(key=col, value=word, -Id, -Helpful, -Unhelpful, -TotalHelpfulRatings, -H_percent, -Score,
-Time, -Summary) %>%
  group_by(word) %>%
  summarize(avghelpful=mean(H_percent), totratings=sum(TotalHelpfulRatings)) %>%
  filter(! is.na(word)) %>%
  arrange(desc(avghelpful))

allwords <- commons %>%
  left_join(avgscores, by=c("word"="word")) %>%
  left_join(avghelpful, by=c("word"="word"))

foodwords_data <- foodwords %>%
  filter(! grepl(" ", word)) %>%
  left_join(allwords, by=c("word"="word")) %>%
  filter(! is.na(count)) %>%
  arrange(desc(count)) %>%
  mutate(posneg=ifelse(AvgScore>4, "pos", "neg"))

pos_foodwords <- foodwords_data %>%
  filter(posneg=="pos")

neg_foodwords <- foodwords_data %>%
  filter(posneg=="neg")

small_foodwords_data <- rbind(head(pos_foodwords, 40), head(neg_foodwords, 40))

small_foodwords_data %>%
  ggplot(aes(x=AvgScore, y=avghelpful)) +
  geom_text(aes(label=word, color=posneg), check_overlap = TRUE) +
  labs(x="Average Score", y="Average Helpfulness", title="Score vs. Helpfulness for
Positive/Negative Words") +
  theme(
       plot.title = element_text(size=16, color='navyblue'),
       axis.title = element_text(size=12, color='navyblue'),
       legend.position = "none"
  )

# Zipfs Law  Code

users <- reviews %>% group_by(UserId) %>% summarise(count = n()) %>% arrange(desc(count))
users <- users %>% mutate(rank = rank(-count))
subUsers <-  sample_n(users, 10000)
userPlot <- ggplot(subUsers, aes(x=rank, y=count)) +
  geom_point() +
  geom_smooth() +
  scale_x_log10() +
  scale_y_log10() +
  labs(title="Zipfs in Reviewers", x="Rank of user that has reviewed", y="Number of Reviews
written") +
  theme(
    axis.title = element_text(size=12, color='navyblue'),
    plot.title = element_text(size=16, color='navyblue'),
    axis.text = element_text(color='dodgerblue4')
  )


products <- reviews %>% group_by(ProductId) %>% summarise(count = n()) %>% arrange(desc(count)) %>%
mutate(rank = rank(-count))

sampleProds <-  sample_n(products, 1000)
productZipf <- ggplot(sampleProds, aes(x=rank, y=count)) +
  geom_point() +
  geom_smooth() +
  scale_x_log10() +
  scale_y_log10() +
  labs(title="Zipfs in Products", x="Rank of product and reviews", y="Number of Reviews the
Product") +
  theme(
    axis.title = element_text(size=12, color='navyblue'),
    plot.title = element_text(size=16, color='navyblue'),
    axis.text = element_text(color='dodgerblue4')
  )
```