MIS 506 Final Project Part 1

1.Load the packages tidytext and tidyverse

library(tidyverse)  
library(tidytext)  
library(widyr)

1. Using the link above, load the review dataset into your workspace. We will need the columns containing which product the review is for as well as the review text, indicate which columns they are and create a new cleaned dataframe with only these two columns. Print the first few rows of data.

I added another column for my additional analysis in problem 9.

amazon\_original <-read\_csv ("1429\_1.csv")  
amazon <-amazon\_original %>%   
 select (name, text=reviews.text, reviewer=reviews.username)  
amazon

## # A tibble: 34,660 x 3  
## name text reviewer  
## <chr> <chr> <chr>   
## 1 All-New Fire HD 8 Tablet, 8 HD Di~ This product so far has not disa~ Adapter   
## 2 All-New Fire HD 8 Tablet, 8 HD Di~ great for beginner or experience~ truman   
## 3 All-New Fire HD 8 Tablet, 8 HD Di~ Inexpensive tablet for him to us~ DaveZ   
## 4 All-New Fire HD 8 Tablet, 8 HD Di~ I've had my Fire HD 8 two weeks ~ Shacks   
## 5 All-New Fire HD 8 Tablet, 8 HD Di~ I bought this for my grand daugh~ explore~  
## 6 All-New Fire HD 8 Tablet, 8 HD Di~ This amazon fire 8 inch tablet i~ tklit   
## 7 All-New Fire HD 8 Tablet, 8 HD Di~ Great for e-reading on the go, n~ Droi   
## 8 All-New Fire HD 8 Tablet, 8 HD Di~ I gave this as a Christmas gift ~ Kacy   
## 9 All-New Fire HD 8 Tablet, 8 HD Di~ Great as a device to read books.~ Weebee   
## 10 All-New Fire HD 8 Tablet, 8 HD Di~ I love ordering books and readin~ RoboBob   
## # ... with 34,650 more rows

1. Summarize the data by outputting the products reviewed in this dataset as well as the number of reviews per product (group\_by the column “name”).

There are 49 products that were reviewed. The number of reviews range from 1 to 10966.

product\_count <- amazon %>%   
 group\_by(name) %>%  
 count(name, sort = TRUE)  
  
product\_count

## # A tibble: 49 x 2  
## # Groups: name [49]  
## name n  
## <chr> <int>  
## 1 "Fire Tablet, 7 Display, Wi-Fi, 8 GB - Includes Special Offers, Magent~ 10966  
## 2 <NA> 6760  
## 3 "Echo (White),,,\r\nEcho (White),,," 3309  
## 4 "Amazon Kindle Paperwhite - eBook reader - 4 GB - 6 monochrome Paperwh~ 3176  
## 5 "All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi, 16 GB - Includes Speci~ 2814  
## 6 "Amazon Fire Tv,,,\r\nAmazon Fire Tv,,," 2527  
## 7 "Fire Kids Edition Tablet, 7 Display, Wi-Fi, 16 GB, Green Kid-Proof Ca~ 1685  
## 8 "Brand New Amazon Kindle Fire 16gb 7 Ips Display Tablet Wifi 16 Gb Blu~ 1038  
## 9 "Kindle Voyage E-reader, 6 High-Resolution Display (300 ppi) with Adap~ 580  
## 10 "Fire Tablet, 7 Display, Wi-Fi, 8 GB - Includes Special Offers, Black" 372  
## # ... with 39 more rows

1. From our data, select two products that are reasonable to compare and create new dataframes for each product containing only the reviews for that product. Identify which product you chose and print the first few rows of data for each product’s dataframe.

My first product is “All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi, 16 GB” with 2814 observations. The shortened name is “HD8Tablet”. My second product is “Fire Kids Edition Tablet, 7 Display, Wi-Fi, 16 GB” with 1685 observations. The shortened name is “KidsTablet”.

name <- amazon$name  
product\_name <-split (amazon, name)  
HD8Tablet <- product\_name [[1]] # All-New Fire HD 8 Tablet, 8 HD Display, Wi-Fi, 16 GB  
KidsTablet <- product\_name [[37]] # Fire Kids Edition Tablet, 7 Display, Wi-Fi, 16 GB,   
  
products <- bind\_rows(HD8Tablet %>%   
 mutate(name = "HD8Tablet"),  
 KidsTablet %>%   
 mutate(name = "KidsTablet"))

1. Create a tidy text object using the “products” dataframe created in step 4 (unigrams).

tidy\_products <- products %>%  
 unnest\_tokens("word", text)

1. Clean the text of the dataset’s review column as you see fit and find the top 20 frequent words for the products’ reviews. Visualize the results. Which words are unique to each product’s top 20 list? Which words appear in both?

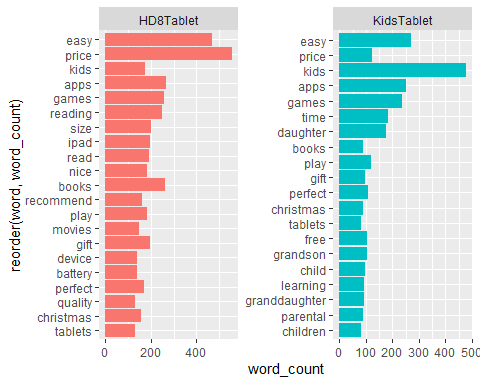
undesirable\_words <- c("amazon", "tablet", "kindle", "fire", "display", "screen", "product", "bought", "love", "loves", "purchased")  
tidy\_products\_filtered <- tidy\_products %>%  
 anti\_join(stop\_words)%>% #remove the stop words using the lexicon  
 filter (!word %in% undesirable\_words) %>% #remove the undesirable words  
 filter(nchar(word) > 3) %>% #remove words with less than 3 characters  
 filter(   
 !str\_detect(word, "^\\b\\d+\\b"), #remove numbers  
 !str\_detect(word, "\\s+"), #remove white spaces  
 !str\_detect(word, "[^a-zA-Z]")) #remove special characters

## Joining, by = "word"

tidy\_products\_filtered %>%  
   
 group\_by(name, word)%>%  
 summarise(word\_count = n()) %>%  
 arrange(desc(word\_count)) %>%  
 top\_n(20) %>%  
 ggplot(aes(x=reorder(word, word\_count), y=word\_count, fill=name)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~name, ncol =5, scales = "free") +   
 coord\_flip()

## `summarise()` has grouped output by 'name'. You can override using the `.groups` argument.

## Selecting by word\_count



Words common to both are “easy”, “price”, “kids”, “games”, “apps”, “play”, “gift”, “perfect”, “christmas”, and “tablets”. The words “kids”, “time”, “daughter”, “granddaughter”, “children”, and “paternal” are more likely to appear in the reviews for the Kids Tablet, which makes sense since the reviews are probably written by the ones who bought it, i.e. the parents or grandparents. The words “reading”, “size”, “lpad”, “nice”, “recommend”, “movies”, “battery”, and “device” are more prevalent in the HD8 Tablet, indicating a more mature user who is more interested in more serious computing power.

1. Perform a sentiment analysis and calculate polarity of reviews for each product name using the bing sentiment dictionary. Based on your results, which product would you choose to purchase?

bing<-get\_sentiments("bing")  
  
joined\_tidy\_products\_filtered <- tidy\_products\_filtered %>%  
 inner\_join(bing) %>%  
 count(name,sentiment,sort = TRUE) %>%   
 # Group by sentiment  
 spread(sentiment, n, fill = 0) %>%  
 mutate(polarity = positive - negative) %>%  
 mutate(percent\_positive = positive / (positive + negative) \* 100)

## Joining, by = "word"

joined\_tidy\_products\_filtered

## # A tibble: 2 x 5  
## name negative positive polarity percent\_positive  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 HD8Tablet 870 3320 2450 79.2  
## 2 KidsTablet 691 1788 1097 72.1

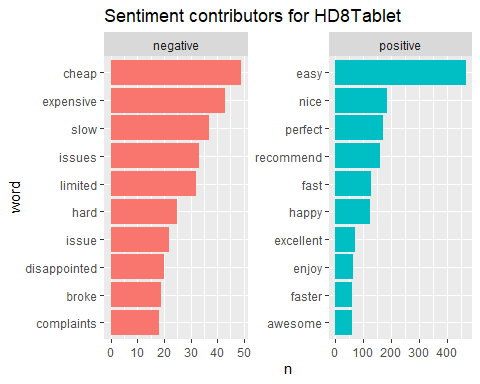
The HD8 Tablet has a positive polarity of 79%. The Kids Tablet has a positive polarity of 72%. The reviews are more positive for the HD8 Tablet, so that should be the one to buy.

1. It’s important to understand which words specifically are driving sentiment scores. Using the bing sentiment dictionary, count by word and sentiment for each product to find “positive” words used in each product’s reviews. Visualize the top “positive” words used in each product’s reviews. Do the same for “negative” words.

tidy\_products\_filtered %>%  
 filter(name == "HD8Tablet") %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(word,sentiment,sort = TRUE) %>%   
 group\_by(sentiment)%>%  
 top\_n(10)%>%  
 ungroup() %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(word,n, fill=sentiment)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~ sentiment, ncol =5, scales = "free")+  
 ggtitle("Sentiment contributors for HD8Tablet") +  
 coord\_flip()

## Joining, by = "word"

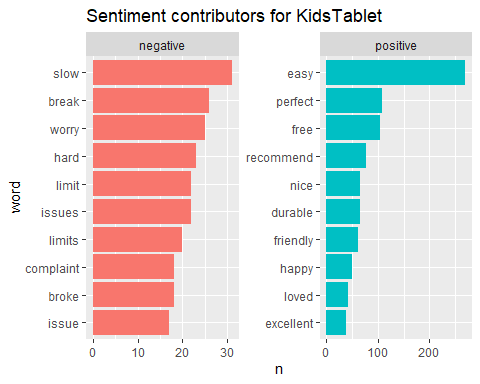
## Selecting by n



tidy\_products\_filtered %>%  
 filter(name == "KidsTablet") %>%  
 inner\_join(get\_sentiments("bing")) %>%  
 count(word,sentiment,sort = TRUE) %>%   
 group\_by(sentiment)%>%  
 top\_n(10)%>%  
 ungroup() %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(word,n, fill=sentiment)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~ sentiment, ncol =5, scales = "free")+  
 ggtitle("Sentiment contributors for KidsTablet") +  
 coord\_flip()

## Joining, by = "word"

## Selecting by n



Positive terms that are duplicates are “easy”, “perfect”, “recommend”, “nice”, “excellent”, and “happy”. Positive terms unique to the HD Tablet are “fast”, “faster”, “enjoy”, and “awesome”, indicating a preference for performance and excitement. Positive terms unique to the kids tablet are “Free”, “durable”, and “happy”, which are all terms that one would expect from an adult reviewing a kids electronic device.

Negative terms that are duplicates are “issue/issues”, “limits/limit”, “complaint”, “hard”, “slow”, and “broke”. Negative terms unique to the HD Tablet are “cheap”, “expensive” (both of them!), and “disappointed”. Negative terms unique to the Kids tablet are “complaint”, “worry”, and “break”.

1. Calculate the correlation among words (unigrams) in the reviews field for the two products’ review data using pairwise\_cor() from the widyr package (correlation among words even if they don’t appear next to each other in the text) (filter for at least relatively common words n()>=2000). Plot networks of correlations among words using igraph and ggraph libraries. Explain the results

I changed n to >= 200 to get some results instead of an error.

We’re pairing every word in every review for a given product, and seeing if the same combination occurs in the other product. The correlation shows up as NaN (Not a Number). So it looks like the results are the following words all paired up with each other: movies, nice, ipad, daughter, price, tablets, perfect, size, reading, christmas, gift, easy, read, books, time, play, games, apps, recommend, and kids. Not very exciting. And because the correlation is NaN, graphing it won’t do any good.

product\_word\_cors <- tidy\_products\_filtered %>%  
 group\_by(word) %>%  
 filter(n() >= 200) %>%  
 pairwise\_cor(word, name, sort = TRUE)

## Warning: `tbl\_df()` was deprecated in dplyr 1.0.0.  
## Please use `tibble::as\_tibble()` instead.

product\_word\_cors

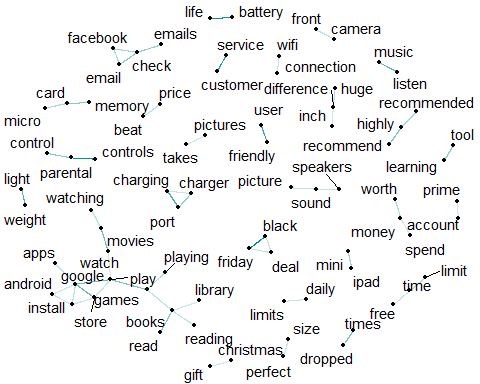
## # A tibble: 380 x 3  
## item1 item2 correlation  
## <chr> <chr> <dbl>  
## 1 movies gift NaN  
## 2 nice gift NaN  
## 3 ipad gift NaN  
## 4 daughter gift NaN  
## 5 price gift NaN  
## 6 tablets gift NaN  
## 7 perfect gift NaN  
## 8 size gift NaN  
## 9 reading gift NaN  
## 10 christmas gift NaN  
## # ... with 370 more rows

“pairwise\_cor” does not yield a lot of information when there are only two things to compare. Our data doesn’t have a “review\_id”, but I took the reviewer and made that the unique key to compare the reviews with. And I make n >= 20.

product\_word\_cors <- tidy\_products\_filtered %>%  
 group\_by(word) %>%  
 filter(n() >= 20) %>%  
 pairwise\_cor(word, reviewer, sort = TRUE)  
  
product\_word\_cors

## # A tibble: 112,560 x 3  
## item1 item2 correlation  
## <chr> <chr> <dbl>  
## 1 friday black 0.895  
## 2 black friday 0.895  
## 3 controls parental 0.795  
## 4 parental controls 0.795  
## 5 life battery 0.723  
## 6 battery life 0.723  
## 7 service customer 0.593  
## 8 customer service 0.593  
## 9 friendly user 0.557  
## 10 user friendly 0.557  
## # ... with 112,550 more rows

library (igraph)  
library (ggraph)  
  
product\_word\_cors %>%  
 filter(correlation > .20) %>%  
 graph\_from\_data\_frame() %>%  
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = correlation), edge\_colour = "cyan4", show.legend = FALSE) +  
 geom\_node\_point( size = 1) +  
 geom\_node\_text(aes(label = name), repel = TRUE) +  
 theme\_void()



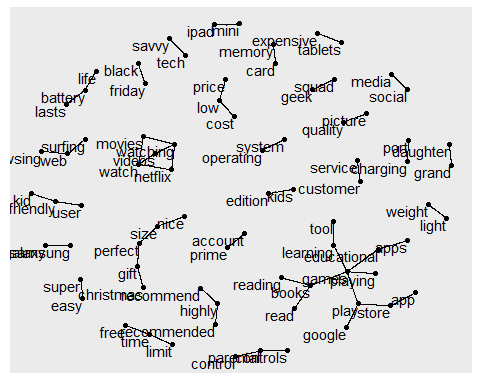
We have one larger network related to what you can do on it, which includes apps, install, movies, books, games, play, read, and library. There’s a small network of “picture/sound/speakers” which seem related to hardware. There are other small networks that seem easily recognizable, including “highly/recommend/recommended”, “facebook/check/emails/email”, and “free/time/limit”.

1. Create a tidy text object using the “products” dataframe created in step 4 (bigrams). Clean the bigrams dataset by removing stop words and words with less than 3 characters. Count and visualize the most common bigrams using igraph and ggraph libraries. Explain the results.

products\_bigrams <- products %>%  
 unnest\_tokens(bigram, text, token = "ngrams", n = 2)  
  
bigrams\_separated <- products\_bigrams %>%  
 separate(bigram, c("word1", "word2"), sep = " ")  
  
bigrams\_filtered <- products\_bigrams %>%  
 separate(bigram, c("word1", "word2"), sep = " ") %>%  
 filter(!word1 %in% stop\_words$word,  
 !word2 %in% stop\_words$word )%>%   
 filter(!word1 %in% undesirable\_words,  
 !word2 %in% undesirable\_words)%>%   
 filter(!nchar(word1) < 3,  
 !nchar(word2) < 3)  
  
bigram\_counts <- bigrams\_filtered %>%   
 count(word1, word2, sort = TRUE)  
bigram\_counts

## # A tibble: 5,815 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 battery life 113  
## 2 black friday 83  
## 3 play games 82  
## 4 parental controls 80  
## 5 google play 74  
## 6 user friendly 58  
## 7 app store 47  
## 8 christmas gift 47  
## 9 highly recommend 47  
## 10 watch movies 47  
## # ... with 5,805 more rows

set.seed(1234)  
bigram\_graph <- bigram\_counts %>%  
 filter(n >= 10) %>%  
 graph\_from\_data\_frame(directed = FALSE)  
  
set.seed(2017)  
  
ggraph(bigram\_graph, layout = "fr") +  
 geom\_edge\_link() +  
 geom\_node\_point() +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1)



From looking at the counts, “battery/life” is the most popular bigram, followed by “black/friday”, “play/games”, and “parental/controls”. When graphing, there’s an interesting clump of “movies/watch/watching/videos/netflix”. There is also a network of “read/reading/books/games/store/app/apps”. Other noticeable networks are “geek/squad”, “prime/account”, and “super/easy”.