Martin-Final.R

serio

Wed Jul 31 16:38:10 2019

#Martin-Final  
#Final Project  
#Ethan Martin  
  
#Before I got started I had to register as a Twitter Ap developer to get the Consumer API keys I needed to access the data  
#Set WD  
#Load Packages------------  
require(twitteR)

## Loading required package: twitteR

## Warning: package 'twitteR' was built under R version 3.5.3

require(RCurl)

## Loading required package: RCurl

## Warning: package 'RCurl' was built under R version 3.5.1

## Loading required package: bitops

## Warning: package 'bitops' was built under R version 3.5.2

require(wordcloud)

## Loading required package: wordcloud

## Warning: package 'wordcloud' was built under R version 3.5.3

## Loading required package: RColorBrewer

## Warning: package 'RColorBrewer' was built under R version 3.5.2

require(tm)

## Loading required package: tm

## Warning: package 'tm' was built under R version 3.5.3

## Loading required package: NLP

## Warning: package 'NLP' was built under R version 3.5.2

require(tidyverse)

## Loading required package: tidyverse

## Warning: package 'tidyverse' was built under R version 3.5.3

## -- Attaching packages ----------------------------------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 2.1.3 v dplyr 0.8.3  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.5.2

## Warning: package 'tibble' was built under R version 3.5.3

## Warning: package 'tidyr' was built under R version 3.5.2

## Warning: package 'readr' was built under R version 3.5.1

## Warning: package 'dplyr' was built under R version 3.5.3

## Warning: package 'stringr' was built under R version 3.5.2

## -- Conflicts -------------------------------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x ggplot2::annotate() masks NLP::annotate()  
## x tidyr::complete() masks RCurl::complete()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::id() masks twitteR::id()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::location() masks twitteR::location()

require(tidytext)

## Loading required package: tidytext

## Warning: package 'tidytext' was built under R version 3.5.3

require(tibble)  
require(topicmodels)

## Loading required package: topicmodels

## Warning: package 'topicmodels' was built under R version 3.5.3

#Storing the Keys------------  
api\_key <- "zmI9oGmAkUH9fVehqMg8Vwlso"  
api\_secret <- "AjVI8GE7n5qZC1wOrelYqgNG7bAVgyOnPTeHIe9kfvufRLKSr6"   
access\_token <- "1155137965561135105-o5L4ACZiXUZ9xfpr34HHpsrFPd2Vsk"  
access\_token\_secret <- "T39PIAw0WraB6UXC536axLdCmsSY3WjBvv5orcCOCCcUF"   
#Accessing the data  
setup\_twitter\_oauth(api\_key, api\_secret, access\_token, access\_token\_secret)

## [1] "Using direct authentication"

1

## [1] 1

#Time to download some twitter data--------------  
dunedin\_tweets <- searchTwitter("dunedin", n = 100, lang = "en")  
dunedin\_tweets\_text <- sapply(dunedin\_tweets, function(x) x$getText())  
head(dunedin\_tweets\_text, 10)

## [1] "Are any twitter whanau going to the AGM in Dunedin this weekend? I'm a delegate, been instructed to sit with a bunc… https://t.co/nN07dDLtvl"   
## [2] "Cal Stevenson hurts in this. He was jumped from Bluefield to Dunedin (skipping two levels altogether) and started p… https://t.co/mFOGdBMKGK"   
## [3] "@DrJamesCMorgan @bournbrookmag I suggest Robin Murray’s work, + the Dunedin study,pointing to cannabis’ effects on… https://t.co/Nni8chVw04"   
## [4] "Wow end of an era. Nicely told yarn Elena. https://t.co/wQl4EngUa8"   
## [5] "RT @Disc\_NewZealand: Dunedin. New Zealand https://t.co/qzfvIDZRuH"   
## [6] "Dunedin History Museum hosts 'The Picture Gallery: Ink on Skin' event https://t.co/xMZTayEduL"   
## [7] "@Alisonmau $250 for kilts in Dunedin for a number of schools, $250 blazers... costs about $800 for year 9 and then… https://t.co/kqvKEiB3SQ"   
## [8] "RT @CrAaronHawkins: This isn't unexpected, given they've already gone in to liquidation, but it would appear to be the final nail in the co…"   
## [9] "@ChickenMan3010 As a side note, the Philles left Jack Russell years ago and the Cardinals left Al Lang years ago, b… https://t.co/DU8znkyXLn"   
## [10] "Kai Inu Totora. Eat, Drink, Be Well in kiwi. Seriously my life's motto in another language <U+0001F44F>\n•\n•\n<U+0001F4F8>: @dennyver\n•\n•… https://t.co/cIhaCY1OTu"

#create corpus  
dunedin\_tweets\_text\_corpus <- Corpus(VectorSource(dunedin\_tweets\_text))  
#clean up  
dunedin\_tweets\_text\_corpus <- tm\_map(dunedin\_tweets\_text\_corpus, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(dunedin\_tweets\_text\_corpus,  
## content\_transformer(tolower)): transformation drops documents

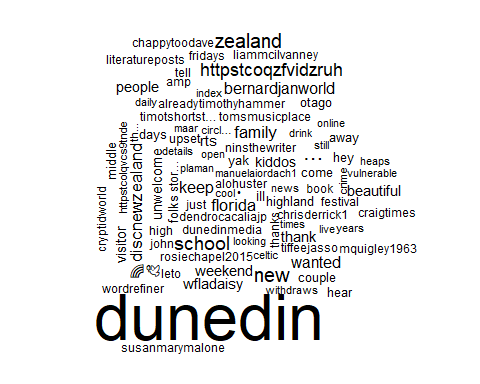
dunedin\_tweets\_text\_corpus <- tm\_map(dunedin\_tweets\_text\_corpus, removePunctuation)

## Warning in tm\_map.SimpleCorpus(dunedin\_tweets\_text\_corpus,  
## removePunctuation): transformation drops documents

dunedin\_tweets\_text\_corpus <- tm\_map(dunedin\_tweets\_text\_corpus, function(x)removeWords(x,stopwords()))

## Warning in tm\_map.SimpleCorpus(dunedin\_tweets\_text\_corpus, function(x)  
## removeWords(x, : transformation drops documents

wordcloud(dunedin\_tweets\_text\_corpus)



#I dont think this is the data I'm trying to aquire exactly to many tweets from scotland and new zealand (other Dunedins)  
dunedinFL\_tweets <- searchTwitter("dunedinFL", n = 100, lang = "en")

## Warning in doRppAPICall("search/tweets", n, params = params,  
## retryOnRateLimit = retryOnRateLimit, : 100 tweets were requested but the  
## API can only return 56

dunedinFL\_tweets\_text <- sapply(dunedinFL\_tweets, function(x) x$getText())  
head(dunedinFL\_tweets\_text, 10)

## [1] "The Original Sandbar Grill\n2602 Bayshore Blvd\nDunedin, Florida 34698\n(727) 734-1962\nhttps://t.co/5pWcvsPdoe… https://t.co/HNOWYTxE0o"   
## [2] "overcast clouds -&gt; mist\ntemperature down 85°F -&gt; 80°F\nhumidity up 70% -&gt; 88%\nwind 6mph -&gt; 8mph"   
## [3] "current weather in Dunedin: overcast clouds, 85°F\n70% humidity, wind 6mph, pressure 1016mb"   
## [4] "Join us at Sandbar Grill in Dunedin for Happy Hour 4 pm - 7 pm daily!\n\n #DunedinFL #Dunedin #EatLocal https://t.co/MtZ0GRyLfu"   
## [5] "current weather in Dunedin: broken clouds, 90°F\n66% humidity, wind 5mph, pressure 1018mb"   
## [6] "RT @roosites: <U+274C> 7 Overrated Business Tips You Shouldn't Follow\n\n<U+0001F310> https://t.co/MgOnoSTpjH\n\n<U+0001F198> For help with your website, contact us: http…"  
## [7] "<U+274C> 7 Overrated Business Tips You Shouldn't Follow\n\n<U+0001F310> https://t.co/MgOnoSTpjH\n\n<U+0001F198> For help with your website, contac… https://t.co/Ln1p47zW06"  
## [8] "current weather in Dunedin: thunderstorm, 84°F\n69% humidity, wind 9mph, pressure 1016mb"   
## [9] "RT @KipAuthor: Dolphin! From today’s paddling workout. Great way to start a #Monday! #dolphins #amwritingscifi #paddlingwriter #writingcomm…"   
## [10] "current weather in Dunedin: thunderstorm, 89°F\n66% humidity, wind 3mph, pressure 1017mb"

#This is mostly weather tweets, some of this is exactly what i"m looking for, but not enough of it. Also just not enough overall.  
#I am not satisfied with this data set. Think I am going to try a broader category.  
beer\_tweets <- searchTwitter("craftbeer", n=1000, lang = "en")  
beer\_tweets\_text <- sapply(beer\_tweets, function(x) x$getText())  
head(beer\_tweets\_text, 10)

## [1] "RT @acajunbowl: Come have a beer and attend Cajun Night at Millennial! #cajunfood #foodtruck #swflfoodies #swfl #craftbeer #brewery #gumbo.…"   
## [2] "RT @SilversmithBrew: You know what they say about when life gives you melons? \nYou might be dyslexic.<U+0001F605> This joke is brought to you by Silve…"  
## [3] "Did your favorite pub make the list? @craftbeerdotcom has compiled the 2019 Great American Beer Bars list based upo… https://t.co/IrHSH0sydW"   
## [4] "Take A Sneak Peek Of Newtown Brewing Co., Opening Soon Locally- #BucksCounty @BrewingNewtown \n#CraftBeer <U+0001F37B>… https://t.co/jyw9NBMj8Q"  
## [5] "RT @mikezoller: The latest from @porchdrinkingco #craftbeer Denver Beer Beat | Colorado Beer Events for Week of July 31, 2019 https://t.co/…"   
## [6] "Congratulations to Team ‘Los Gatos Locos’ for winning 1st place at Margaritas!\n.\n.\n#trivianight #triviawinners… https://t.co/C2P6RlN8Yr"   
## [7] "Congratulations to Team ‘Patel’ for winning 2nd place at Margaritas!\n.\n.\n#trivianight #triviawinners… https://t.co/UytZu2281s"   
## [8] "The latest from @porchdrinkingco #craftbeer Denver Beer Beat | Colorado Beer Events for Week of July 31, 2019 https://t.co/1HpxQ3kzWH"   
## [9] "Maybe a new #craftbeer flavor? https://t.co/GqUhDWEeal"   
## [10] "New Beer Friday 2/3/17 https://t.co/67JJyb5J0K"

#I would like to narrow this down more to florida craft beers, which I have a very good knowledge of that will hopefully  
#help me interpret the data. skimming through the tweets I noticed this---"Enjoying a @jdubsbrewing Poolside by the ... poolside.  
#\n\nI'm such a hack. \n\n#florida #craftbeer #drinkfloridacraft https://t.co/tH5HwqAmcO" I'm very familiar with this brewery/beer  
#and they used the hashtag #drinkfloridacraft. I am going to try this hashtag and see how that goes.  
FLbeer\_tweets <- searchTwitter("drinkfloridacraft", n=1000, lang = "en")

## Warning in doRppAPICall("search/tweets", n, params = params,  
## retryOnRateLimit = retryOnRateLimit, : 1000 tweets were requested but the  
## API can only return 11

FLbeer\_tweets\_text <- sapply(FLbeer\_tweets, function(x) x$getText())  
head(FLbeer\_tweets\_text, 10)

## [1] "RT @FloridaBeerBlog: Have you read about two of the most fantastic beers that ever came out of Tampa? Head to https://t.co/IIkaFpXS32 righ…"   
## [2] "Have you read about two of the most fantastic beers that ever came out of Tampa? Head to https://t.co/IIkaFpXS32 r… https://t.co/koX8vqpcaX"   
## [3] "RT @FloridaBeerBlog: Very excited to rep my new @playalindabrewco gear. Thank you so much! #florida #craftbeer #drinkfloridacraft https://t…"   
## [4] "Very excited to rep my new @playalindabrewco gear. Thank you so much! #florida #craftbeer #drinkfloridacraft https://t.co/y4N2st478H"   
## [5] "RT @FloridaBeerBlog: I figure tonight calls for some Mexican Lager courtesy of @coppertailbrewing and @herculesbrewingusa #drinkfloridacraf…"   
## [6] "I figure tonight calls for some Mexican Lager courtesy of @coppertailbrewing and @herculesbrewingusa… https://t.co/4Yogwauji4"   
## [7] "Enjoying a @jdubsbrewing Poolside by the ... poolside. \n\nI'm such a hack. \n\n#florida #craftbeer #drinkfloridacraft https://t.co/tH5HwqAmcO"  
## [8] "So I've been to Half Barrel Beer Project before, and Episode 20 of the Florida Beer Podcast has our interview with… https://t.co/PsMXvsHAOS"   
## [9] "Hanging out with some friends in Deerfield Beach at the recent Ocean Blues and Brews festival. Want to read more ab… https://t.co/e76s0m6aNS"   
## [10] "Finally got around to writing up my review of the Ocean Brews and Blues Festival in Delray! Take a look now at… https://t.co/Hbs17hmrk9"

#This only returned 11 tweets, not a popular hashtag. I'm going to go on the internet and look up popular florida craft brewing hashtags and  
#see if that helps.  
#Ok. Optimizing hashtags is apparently a much bigger deal than I realized. Not that it doesn't make sense, I just never thought about it.  
#I think there is no single hashtag that is going to do the trick. I am going to see if I can figure out how to search for multiple  
#hash tags at once. If possible that should give me the data I want.....I think.....  
hashtags <- 'craftbeer + florida'  
FLbeer\_tweets <- searchTwitter(hashtags, n = 1000, lang = 'en')

## Warning in doRppAPICall("search/tweets", n, params = params,  
## retryOnRateLimit = retryOnRateLimit, : 1000 tweets were requested but the  
## API can only return 41

#I only got 41, but thanks to the magic of copy and paste im going to do a quick visualization to see how it looks.  
FLbeer\_tweets\_text <- sapply(FLbeer\_tweets, function(x) x$getText())  
#create corpus  
FLbeer\_tweets\_text\_corpus <- Corpus(VectorSource(FLbeer\_tweets\_text))  
#clean up  
FLbeer\_tweets\_text\_corpus <- tm\_map(FLbeer\_tweets\_text\_corpus, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(FLbeer\_tweets\_text\_corpus,  
## content\_transformer(tolower)): transformation drops documents

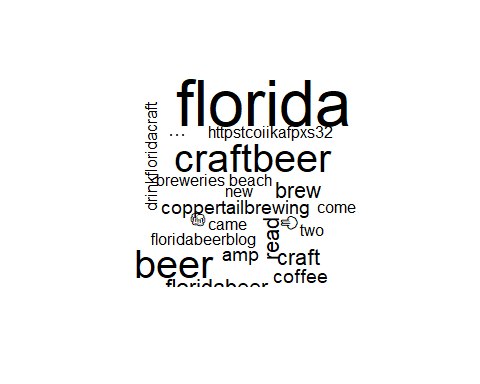
FLbeer\_tweets\_text\_corpus <- tm\_map(FLbeer\_tweets\_text\_corpus, removePunctuation)

## Warning in tm\_map.SimpleCorpus(FLbeer\_tweets\_text\_corpus,  
## removePunctuation): transformation drops documents

FLbeer\_tweets\_text\_corpus <- tm\_map(FLbeer\_tweets\_text\_corpus, function(x)removeWords(x,stopwords()))

## Warning in tm\_map.SimpleCorpus(FLbeer\_tweets\_text\_corpus, function(x)  
## removeWords(x, : transformation drops documents

wordcloud(FLbeer\_tweets\_text\_corpus, max.words = 100)



#I learned two things. I don't think this is a big enough sample. Also, emoji's show up on wordclouds. In this case  
#it was the two beer mugs clinking together. So that was cool...back to the data mines we go.  
#I may have to give up on making these florida specific for now. going to add drinklocal hashtag to craft beer. It's my hope  
#this will eliminate the craft beers made by the conglomerates and will get more small brewery information.  
hashtags <- 'craftbeer + drinklocal'  
FLbeer\_tweets <- searchTwitter(hashtags, n = 1000, lang = 'en')   
#strip retweets out of the data  
FLbeer\_tweets <-strip\_retweets(FLbeer\_tweets)  
  
#Copy/paste...  
FLbeer\_tweets\_text <- sapply(FLbeer\_tweets, function(x) x$getText())  
#create corpus  
FLbeer\_tweets\_text\_corpus <- Corpus(VectorSource(FLbeer\_tweets\_text))  
#clean up  
FLbeer\_tweets\_text\_corpus <- tm\_map(FLbeer\_tweets\_text\_corpus, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(FLbeer\_tweets\_text\_corpus,  
## content\_transformer(tolower)): transformation drops documents

FLbeer\_tweets\_text\_corpus <- tm\_map(FLbeer\_tweets\_text\_corpus, removePunctuation)

## Warning in tm\_map.SimpleCorpus(FLbeer\_tweets\_text\_corpus,  
## removePunctuation): transformation drops documents

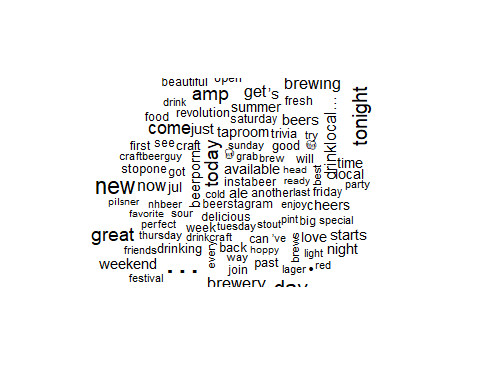
FLbeer\_tweets\_text\_corpus <- tm\_map(FLbeer\_tweets\_text\_corpus, removeNumbers)

## Warning in tm\_map.SimpleCorpus(FLbeer\_tweets\_text\_corpus, removeNumbers):  
## transformation drops documents

FLbeer\_tweets\_text\_corpus <- tm\_map(FLbeer\_tweets\_text\_corpus, removeWords, c(stopwords("english"), "craftbeer"))

## Warning in tm\_map.SimpleCorpus(FLbeer\_tweets\_text\_corpus, removeWords,  
## c(stopwords("english"), : transformation drops documents

wordcloud(FLbeer\_tweets\_text\_corpus, max.words = 100)



#A lot of the words wouldn't fit at first so I added a max.words=100 to it. I think we are headed in the right direction. its a nice large set of the sort of data I was looking for.  
#I think I like the data set, but I am going to switch to the tidy method we learned in datacamp for the midterm to further explore  
#Convert to a format tidytext can use--------------------  
FLbeerTweets <- twListToDF(FLbeer\_tweets)  
  
tidy\_beer <- FLbeerTweets %>%   
 # Tokenize the twitter data  
 unnest\_tokens(word, text) %>%  
 # Remove stop words  
 anti\_join(stop\_words)

## Joining, by = "word"

tidy\_beer %>%   
 # Compute word counts  
 count(word) %>%   
 # Arrange the counts in descending order  
 arrange(desc(n))

## # A tibble: 3,870 x 2  
## word n  
## <chr> <int>  
## 1 https 779  
## 2 t.co 779  
## 3 craftbeer 272  
## 4 beer 256  
## 5 drinklocal 201  
## 6 ipa 73  
## 7 day 56  
## 8 amp 51  
## 9 tonight 51  
## 10 brewery 45  
## # ... with 3,860 more rows

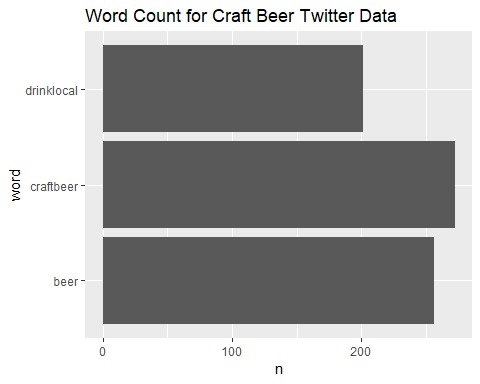
#Looks like I need to add some custom stop words  
custom\_stop\_words <- tribble(  
 # Column names should match stop\_words  
 ~word, ~lexicon,  
 # Add http, win, and t.co as custom stop words  
 "http", "CUSTOM",  
 "https", "CUSTOM",  
 "t.co", "CUSTOM",   
 "amp" , "CUSTOM"  
)  
  
# Bind the custom stop words to stop\_words  
stop\_words2 <- stop\_words %>%   
 bind\_rows(custom\_stop\_words)  
#lets try this again  
tidy\_beer <- FLbeerTweets %>%   
 # Tokenize the twitter data  
 unnest\_tokens(word, text) %>%  
 # Remove stop words  
 anti\_join(stop\_words2)

## Joining, by = "word"

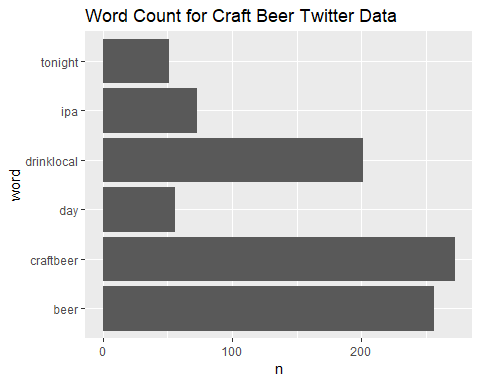
tidy\_beer %>%   
 # Compute word counts  
 count(word) %>%   
 # Arrange the counts in descending order  
 arrange(desc(n))

## # A tibble: 3,867 x 2  
## word n  
## <chr> <int>  
## 1 craftbeer 272  
## 2 beer 256  
## 3 drinklocal 201  
## 4 ipa 73  
## 5 day 56  
## 6 tonight 51  
## 7 brewery 45  
## 8 tap 43  
## 9 beerporn 39  
## 10 brewing 37  
## # ... with 3,857 more rows

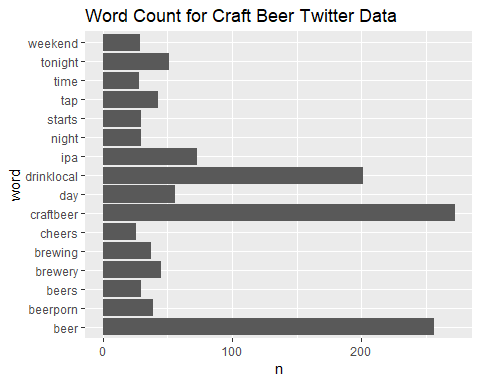
#Everything seems to be working, lets try to visualize---------------------  
word\_counts <- tidy\_beer %>%   
 count(word) %>%   
 filter(n > 100)  
  
# Create a bar plot using the new word\_counts  
ggplot(word\_counts, aes(x = word, y = n)) +  
 geom\_col() +  
 coord\_flip() +  
 ggtitle("Word Count for Craft Beer Twitter Data")



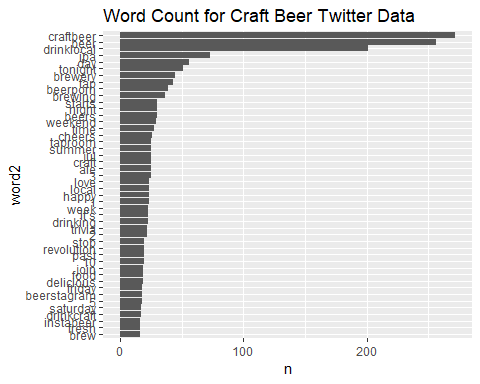
# I need to set a lower bar, I'm only returning 3 words  
word\_counts <- tidy\_beer %>%   
 count(word) %>%   
 filter(n > 50)  
ggplot(word\_counts, aes(x = word, y = n)) +  
 geom\_col() +  
 coord\_flip() +  
 ggtitle("Word Count for Craft Beer Twitter Data")



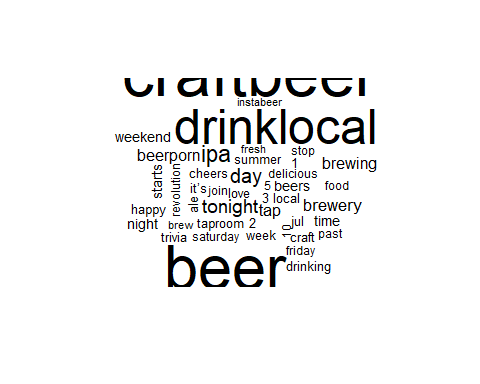
#There are now 5 terms, lowering the bar....  
word\_counts <- tidy\_beer %>%   
 count(word) %>%   
 filter(n > 25)  
ggplot(word\_counts, aes(x = word, y = n)) +  
 geom\_col() +  
 coord\_flip() +  
 ggtitle("Word Count for Craft Beer Twitter Data")



#That looks better, up to 16 words, but it needs to be put in order and a few more words would be good  
word\_counts <- tidy\_beer %>%   
 count(word) %>%   
 filter(n > 15) %>%  
 mutate(word2 = fct\_reorder(word, n))  
ggplot(word\_counts, aes(x = word2, y = n)) +  
 geom\_col() +  
 coord\_flip() +  
 ggtitle("Word Count for Craft Beer Twitter Data")



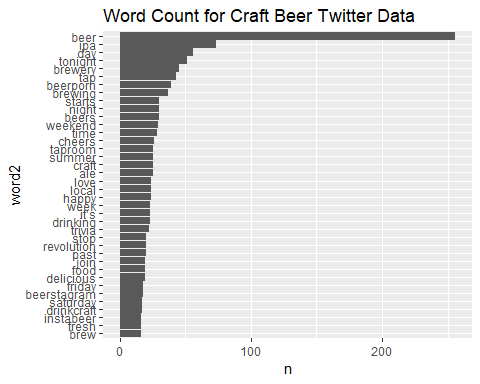
#There we go, that looks better  
wordcloud(  
 # Assign the word column to words  
 words = word\_counts$word,   
 # Assign the count column to freq  
 freq = word\_counts$n,  
 max.words = 50  
)



#Some how some numbers and jul snuck in there. Also going to get rid of the two hashtags I searched for in the first place.  
custom\_stop\_words2 <- tribble(  
 # Column names should match stop\_words  
 ~word, ~lexicon,  
 # Add http, win, and t.co as custom stop words  
 "http", "CUSTOM",  
 "https", "CUSTOM",  
 "t.co", "CUSTOM",   
 "amp", "CUSTOM",  
 "craftbeer", "CUSTOM",  
 "drinklocal", "CUSTOM",  
 "2", "CUSTOM",  
 "3", "CUSTOM",  
 "1", "CUSTOM",  
 "4", "CUSTOM",  
 "5", "CUSTOM",  
 "6", "CUSTOM",  
 "7", "CUSTOM",  
 "8", "CUSTOM",  
 "9", "CUSTOM",  
 "10", "CUSTOM",  
 "it's", "CUSTOM",  
 "jul", "CUSTOM"  
)  
stop\_words3 <- stop\_words %>%   
 bind\_rows(custom\_stop\_words2)  
#lets try this once again  
tidy\_beer <- FLbeerTweets %>%   
 # Tokenize the twitter data  
 unnest\_tokens(word, text) %>%  
 # Remove stop words  
 anti\_join(stop\_words3)

## Joining, by = "word"

word\_counts <- tidy\_beer %>%   
 count(word) %>%   
 filter(n > 15) %>%  
 mutate(word2 = fct\_reorder(word, n))  
ggplot(word\_counts, aes(x = word2, y = n)) +  
 geom\_col() +  
 coord\_flip() +  
 ggtitle("Word Count for Craft Beer Twitter Data")



#There we go, that looks better  
wordcloud(  
 # Assign the word column to words  
 words = word\_counts$word,   
 # Assign the count column to freq  
 freq = word\_counts$n,  
 max.words = 40,  
 color = "blue")



#I see ipa.. That makes sense as it is probably the most popular type of craft beer. There are a few hashtags in there it looks like as well.  
#Things like "beerporn" (probably pictures of beer looking delicious). Also "instabeer" and "beerstagram" are instagram specific hashtags.  
#There are a lot of date/time type words. This is probably due to tweets related to beer festivals and releases. In the wordmap shown in fact,  
#the words festival, tonight, starts, and time all appeared.   
#The words tap, taproom, and brewery suggest that the best place to get great craft beer is still on tap at your local brewery.  
  
#Lets explore some sentiments-----------------------------  
# Join tidy\_beer and the bing sentiment dictionary  
bing\_sentiment\_beer <- tidy\_beer %>%   
 inner\_join(get\_sentiments("bing"))

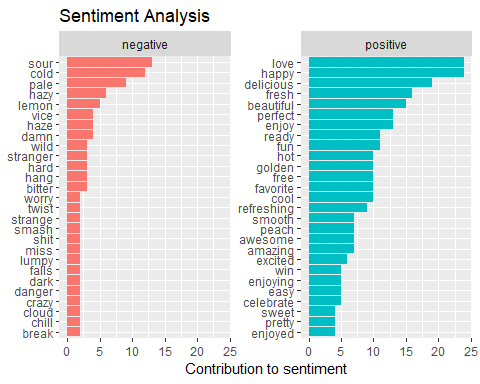
## Joining, by = "word"

#I would like to see more rows so I changing the tibble options  
options(tibble.print\_max = 30, tibble.print\_min = 25)  
  
  
bing\_word\_counts <- bing\_sentiment\_beer %>%  
 count(word, sentiment, sort = TRUE) %>%  
 ungroup()  
bing\_word\_counts

## # A tibble: 237 x 3  
## word sentiment n  
## <chr> <chr> <int>  
## 1 happy positive 24  
## 2 love positive 24  
## 3 delicious positive 19  
## 4 fresh positive 16  
## 5 beautiful positive 15  
## 6 enjoy positive 13  
## 7 perfect positive 13  
## 8 sour negative 13  
## 9 cold negative 12  
## 10 fun positive 11  
## 11 ready positive 11  
## 12 cool positive 10  
## 13 favorite positive 10  
## 14 free positive 10  
## 15 golden positive 10  
## 16 hot positive 10  
## 17 pale negative 9  
## 18 refreshing positive 9  
## 19 amazing positive 7  
## 20 awesome positive 7  
## 21 peach positive 7  
## 22 smooth positive 7  
## 23 excited positive 6  
## 24 hazy negative 6  
## 25 celebrate positive 5  
## # ... with 212 more rows

#As expected it was pretty positive. Also upon closer inspection even the negative words arent so negative in beer culture. Sour is not a bad thing,  
#it is in fact a popular style of beer suited well to summertime. The next negative word is cold, which is obviously a good thing for beer to be  
#in the summer. Pale is the next negative word and I'm sure that is just the beer style "pale ale" showing up and not likely a comlplexion issue.   
#Hazy is not good if its weather, but if its beer then it probably refers to hazy new england style IPA which is a popular new beer trend.  
#Lets look closer  
bing\_word\_counts %>%  
 group\_by(sentiment) %>%  
 top\_n(25) %>%  
 ungroup() %>%  
 mutate(word = reorder(word, n)) %>%  
 ggplot(aes(word, n, fill = sentiment)) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~sentiment, scales = "free\_y") +  
 labs(  
 title = "Sentiment Analysis",  
 y = "Contribution to sentiment",  
 x = NULL) +  
 coord\_flip()

## Selecting by n



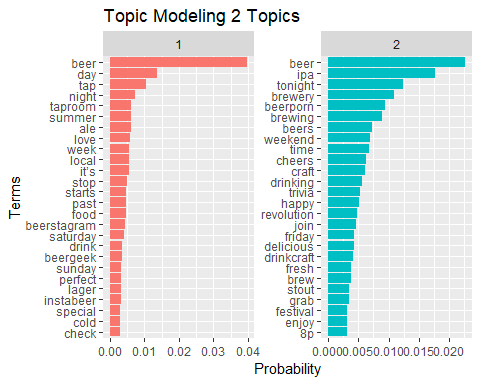
#I'd say at least 7 out of the top ten negative words are actually positive. This data set is pretty overwhelmingly positive, as I suspected.  
#People like to brag about their craft beer, and even if they do have something they dislike they generally only post about the good ones.  
#I personally believe this is because people want to be seen "living their best lives" on social media. Also, craft beer is delicious.  
  
#Lets try LDA modeling, first we need the data to be a DTM------------------------------------  
  
# Assign the DTM to dtm\_beer  
dtm\_beer <- tidy\_beer %>%   
 count(word, id) %>%   
 # Cast the word counts by tweet into a DTM  
 cast\_dtm(id, word, n)  
  
# Coerce dtm\_beer into a matrix called matrix\_beer (do this if you want to look at specific parts of the data)  
#matrix\_beer <- as.matrix(dtm\_beer)  
  
#Create a topic model  
lda\_out <- LDA(  
 dtm\_beer,  
 k = 2,  
 method = "Gibbs",  
 control = list(seed = 22)  
)  
#Take a look at the topic model  
glimpse(lda\_out)

## Formal class 'LDA\_Gibbs' [package "topicmodels"] with 16 slots  
## ..@ seedwords : NULL  
## ..@ z : int [1:7454] 1 1 2 1 2 1 1 1 1 1 ...  
## ..@ alpha : num 25  
## ..@ call : language LDA(x = dtm\_beer, k = 2, method = "Gibbs", control = list(seed = 22))  
## ..@ Dim : int [1:2] 750 3854  
## ..@ control :Formal class 'LDA\_Gibbscontrol' [package "topicmodels"] with 14 slots  
## ..@ k : int 2  
## ..@ terms : chr [1:3854] "00" "03.08" "07" "072opl90pi" ...  
## ..@ documents : chr [1:750] "1155991619767693313" "1156554321795911681" "1154148024786051073" "1154528834357727233" ...  
## ..@ beta : num [1:2, 1:3854] -8.22 -10.63 -10.61 -8.24 -8.22 ...  
## ..@ gamma : num [1:750, 1:2] 0.548 0.508 0.478 0.567 0.459 ...  
## ..@ wordassignments:List of 5  
## .. ..$ i : int [1:7279] 1 1 1 1 1 1 1 1 1 1 ...  
## .. ..$ j : int [1:7279] 1 93 360 2398 2647 2712 2715 2762 2833 3161 ...  
## .. ..$ v : num [1:7279] 1 1 2 1 1 1 1 1 1 1 ...  
## .. ..$ nrow: int 750  
## .. ..$ ncol: int 3854  
## .. ..- attr(\*, "class")= chr "simple\_triplet\_matrix"  
## ..@ loglikelihood : num -60277  
## ..@ iter : int 2000  
## ..@ logLiks : num(0)   
## ..@ n : int 7454

# Tidy the matrix of word probabilities  
lda\_topics <- lda\_out %>%   
 tidy(matrix="beta")  
  
# Arrange the topics by word probabilities in descending order  
lda\_topics %>%   
 arrange(desc(beta))

## # A tibble: 7,708 x 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 beer 0.0398   
## 2 2 beer 0.0227   
## 3 2 ipa 0.0176   
## 4 1 day 0.0138   
## 5 2 tonight 0.0123   
## 6 2 brewery 0.0109   
## 7 1 tap 0.0106   
## 8 2 beerporn 0.00941  
## 9 2 brewing 0.00893  
## 10 2 beers 0.00725  
## 11 1 night 0.00715  
## 12 2 weekend 0.00701  
## 13 2 time 0.00677  
## 14 2 cheers 0.00628  
## 15 1 ale 0.00616  
## 16 1 summer 0.00616  
## 17 1 taproom 0.00616  
## 18 2 craft 0.00604  
## 19 1 love 0.00592  
## 20 1 it’s 0.00567  
## 21 1 local 0.00567  
## 22 1 week 0.00567  
## 23 2 drinking 0.00556  
## 24 2 trivia 0.00532  
## 25 2 happy 0.00508  
## # ... with 7,683 more rows

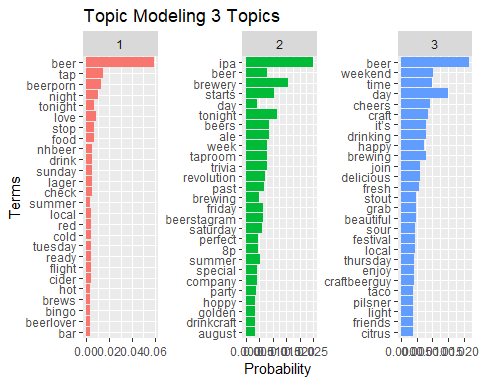
# Select the top 25 terms by topic and reorder term  
word\_probs <- lda\_topics %>%   
 group\_by(topic) %>%   
 top\_n(25, beta) %>%   
 ungroup() %>%  
 mutate(term2 = fct\_reorder(term, beta))  
  
# Plot word\_probs2, color and facet based on topic  
ggplot(  
 word\_probs,   
 aes(term2, beta, fill=as.factor(topic))  
) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~topic, scales = "free") +  
 coord\_flip() +  
 labs(title = "Topic Modeling 2 Topics",  
 y = "Probability",  
 x ="Terms")



#The second topic seems to be beer related words and the first appears to be when and where to get beer.  
#We need more topics  
#Create a model with more topics------------------------  
lda\_out2 <- LDA(  
 dtm\_beer,  
 k = 3,  
 method = "Gibbs",  
 control = list(seed = 22)  
)  
# Tidy the matrix of word probabilities again  
lda\_topics2 <- lda\_out2 %>%   
 tidy(matrix="beta")  
  
# Arrange the topics by word probabilities in descending order again  
lda\_topics2 %>%   
 arrange(desc(beta))

## # A tibble: 11,562 x 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 beer 0.0595   
## 2 2 ipa 0.0252   
## 3 3 beer 0.0216   
## 4 2 brewery 0.0158   
## 5 3 day 0.0150   
## 6 1 tap 0.0147   
## 7 1 beerporn 0.0129   
## 8 2 tonight 0.0116   
## 9 2 starts 0.0105   
## 10 1 night 0.0101   
## 11 3 weekend 0.0101   
## 12 3 time 0.00978  
## 13 3 cheers 0.00908  
## 14 2 ale 0.00877  
## 15 2 beers 0.00877  
## 16 3 craft 0.00874  
## 17 1 love 0.00839  
## 18 2 beer 0.00807  
## 19 2 taproom 0.00807  
## 20 2 week 0.00807  
## 21 3 brewing 0.00804  
## 22 3 drinking 0.00804  
## 23 3 it’s 0.00804  
## 24 2 trivia 0.00772  
## 25 3 happy 0.00734  
## # ... with 1.154e+04 more rows

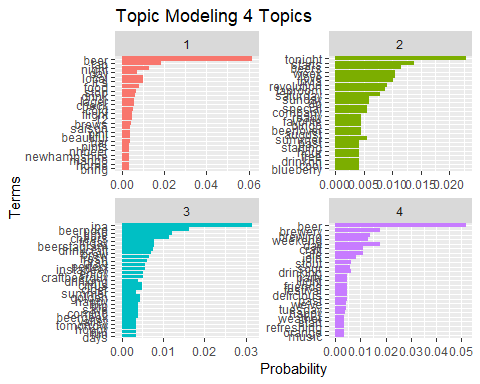
# Select the top 25 terms by topic and reorder term  
word\_probs2 <- lda\_topics2 %>%   
 group\_by(topic) %>%   
 top\_n(25, beta) %>%   
 ungroup() %>%  
 mutate(term2 = fct\_reorder(term, beta))  
  
# Plot word\_probs2, color and facet based on topic  
ggplot(  
 word\_probs2,   
 aes(term2, beta, fill=as.factor(topic))  
) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~topic, scales = "free") +  
 coord\_flip() +  
 labs(title = "Topic Modeling 3 Topics",  
 y = "Probability",  
 x ="Terms")



#The second topic seems to be when people are taking pictures of beer, the first topic is what people are doing while they drink beer,  
#and the third topic is summer beer words maybe?  
#Lets try 4 topics now-------------  
lda\_out3 <- LDA(  
 dtm\_beer,  
 k = 4,  
 method = "Gibbs",  
 control = list(seed = 22)  
)  
# Tidy the matrix of word probabilities again  
lda\_topics3 <- lda\_out3 %>%   
 tidy(matrix="beta")  
  
# Arrange the topics by word probabilities in descending order again  
lda\_topics3 %>%   
 arrange(desc(beta))

## # A tibble: 15,416 x 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 beer 0.0617   
## 2 4 beer 0.0517   
## 3 3 ipa 0.0316   
## 4 2 tonight 0.0228   
## 5 1 tap 0.0187   
## 6 4 brewery 0.0177   
## 7 4 day 0.0177   
## 8 3 beerporn 0.0163   
## 9 2 starts 0.0137   
## 10 4 brewing 0.0137   
## 11 1 night 0.0129   
## 12 4 weekend 0.0128   
## 13 3 time 0.0123   
## 14 2 beers 0.0114   
## 15 3 cheers 0.0114   
## 16 4 ale 0.0111   
## 17 4 craft 0.0111   
## 18 2 love 0.0105   
## 19 2 week 0.0105   
## 20 1 it’s 0.0102   
## 21 1 local 0.0102   
## 22 2 trivia 0.0101   
## 23 2 revolution 0.00916  
## 24 2 taproom 0.00871  
## 25 1 food 0.00847  
## # ... with 1.539e+04 more rows

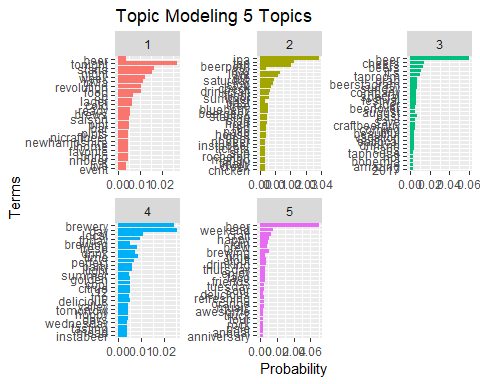
# Select the top 25 terms by topic and reorder term  
word\_probs3 <- lda\_topics3 %>%   
 group\_by(topic) %>%   
 top\_n(25, beta) %>%   
 ungroup() %>%  
 mutate(term2 = fct\_reorder(term, beta))  
  
# Plot word\_probs3, color and facet based on topic  
bestLDAplot <- ggplot(  
 word\_probs3,   
 aes(term2, beta, fill=as.factor(topic))  
) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~topic, scales = "free") +  
 coord\_flip() +  
 labs(title = "Topic Modeling 4 Topics",  
 y = "Probability",  
 x ="Terms")  
bestLDAplot



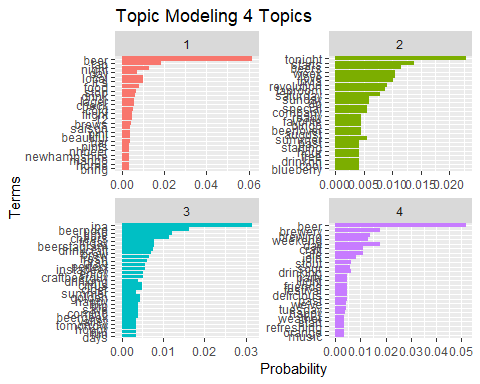
#The first topic appears to be related to ticketed events where people are drinking. The second topic appears to be related to advertising new beers  
#live music and trivia. these are more common types of beer news as opposed to the special events in the first topic. The third topic seems to  
#relate to posting pictures of beer. The fourth topic seems to be how much fun people are having drinking beer.  
#Let's look at 5 topics now.------------------  
lda\_out4 <- LDA(  
 dtm\_beer,  
 k = 5,  
 method = "Gibbs",  
 control = list(seed = 22)  
)  
# Tidy the matrix of word probabilities again  
lda\_topics4 <- lda\_out4 %>%   
 tidy(matrix="beta")  
  
# Arrange the topics by word probabilities in descending order again  
lda\_topics4 %>%   
 arrange(desc(beta))

## # A tibble: 19,270 x 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 5 beer 0.0706  
## 2 3 beer 0.0600  
## 3 2 ipa 0.0389  
## 4 1 tonight 0.0269  
## 5 4 day 0.0256  
## 6 4 brewery 0.0240  
## 7 2 tap 0.0225  
## 8 2 beerporn 0.0203  
## 9 1 starts 0.0162  
## 10 1 night 0.0156  
## 11 5 weekend 0.0151  
## 12 3 cheers 0.0139  
## 13 3 beers 0.0133  
## 14 2 love 0.0132  
## 15 5 craft 0.0130  
## 16 1 week 0.0124  
## 17 2 ale 0.0121  
## 18 1 trivia 0.0113  
## 19 3 it’s 0.0112  
## 20 5 brewing 0.0109  
## 21 5 happy 0.0109  
## 22 4 local 0.0107  
## 23 1 food 0.0103  
## 24 1 revolution 0.0103  
## 25 3 taproom 0.0101  
## # ... with 1.924e+04 more rows

# Select the top 25 terms by topic and reorder term  
word\_probs4 <- lda\_topics4 %>%   
 group\_by(topic) %>%   
 top\_n(25, beta) %>%   
 ungroup() %>%  
 mutate(term2 = fct\_reorder(term, beta))  
# Plot word\_probs3, color and facet based on topic  
ggplot(  
 word\_probs4,   
 aes(term2, beta, fill=as.factor(topic))  
) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~topic, scales = "free") +  
 coord\_flip() +  
 labs(title = "Topic Modeling 5 Topics",  
 y = "Probability",  
 x ="Terms")



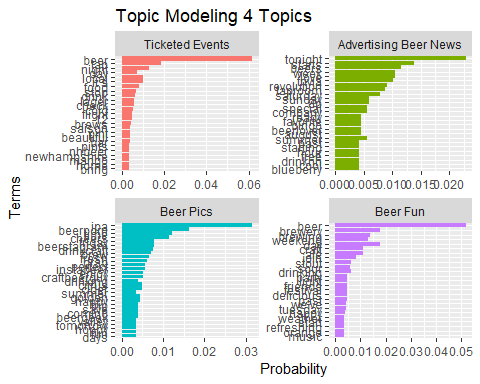
#I think the topics got a little muddled here. 1 and 4 are similar and the topics aren't as distinct.   
#The best model was with 4 topics.------------------------  
bestLDAplot



#I would like to name the facets. After some research this should do the trick.  
#Make a list of the names I want the plots to have tied to the current names of the plots.  
plot\_names <- list(  
 "1" = "Ticketed Events",  
 "2" = "Advertising Beer News",  
 "3" = "Beer Pics",   
 "4" = "Beer Fun")  
  
#Create a labeller function  
beer\_labeller <- function(variable,value){  
 return(plot\_names[value])  
}  
#Pass that function into the facet\_wrap  
bestLDAplot <- ggplot(  
 word\_probs3,   
 aes(term2, beta, fill=as.factor(topic))  
) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~topic, scales = "free", labeller=beer\_labeller) +  
 coord\_flip() +  
 labs(title = "Topic Modeling 4 Topics",  
 y = "Probability",  
 x ="Terms")

## Warning: The labeller API has been updated. Labellers taking `variable`and  
## `value` arguments are now deprecated. See labellers documentation.

bestLDAplot



#There it is! Success!  
#Now that I have the topics labeled, lets see if any more information can be gleaned from them.  
  
#Topic 1, Ticketed Events-----------------  
  
#Words like tonight, food, starts, tickets, week, special, and fun lead me to believe this column features special events that are less frequent  
#and more "special." To judge by the 4th word, love, people tend to enjoy these types of things.  
  
#I would say that festivals, are a great way to get people to your brewery/bar. Also when you get them there provide food. This is supported  
#by words like food and eatlocal. The typical way to accomplish this would be food trucks. So have a festival and get some foodtrucks there.  
#This will also help keep people from getting too blasted at your event in my experience. These are designed to bring in huge crowds on already  
#good days.  
  
#Topic 2, Advertising Beer News-----------------  
  
# Words like beer, news, brewed, fresh, citrus indicate this topic is about new beer releases. The presence of trivia, live, and rock also  
#support that this is a topic related to not ticketed special events. the presence of Wednesday supports that these are more low key weekly  
#events.  
  
  
#It is always important to announce your new beer releases. This is a great chance to get your regulars to come in on a day they might not   
#normally to try a new beer. Also you might gain a new customer if it is a particular flavour or style they enjoy. Also trivia and live  
#music are a great way to have that same effect. These types of things should be planned more durring the week to lure in customers on slower  
#days, unlike topic 1.  
  
#Topic 3 Beer Pics-----------------------------------------------  
  
# Hashtags donminate this list. beerporn, like its close relative foodporn, are hashtags that often accompany the sharing of artfully staged  
#photos of beer/food. Beerstagram, drinkcraft. beergeek, and craftbeerguy also make an appearence. People often take a picture of their beer  
#and then post it with a string of hashtags.   
  
#I would say that propper hashtag useage will help spread your social media efforts wider. I might encourage a bar/brewery owner to consider  
#posting popular beer-related hashtags somewhere to help increase their social media presence. Maybe even a special little table somewhere   
#with a backdrop featuring your logo, maybe some props (frames, some hops, funny glasses, that sort of thing) as a sort of social media   
#photo-booth/beerporn presentation station.  
  
#Topic 4, Beer Fun----------------------------------------  
  
#This topic seems to be about how much fun people are having drinking beer! I also have fun drinking beer so I can see why this is a topic.  
#Words like delicious, cold, enjoy, beautiful, and tasty seem to support this. Also, when you are having fun you often use social media as a   
#way to let all your friends know how much fun you are having and to invite them to join you for a beer, maybe even a chicken taco.  
  
#Again, social media is a great way to draw in customers. If people are having fun they want to show everyone that they are  
#"living their best lives." As was mentioned for the previous topic, Encourage social media use any way that you can. Have someone in charge of  
#posting and responding to others. If you don't have someone talented running your social media you should consider hiring someone to do that  
#specifically, or more likely, look for that quality in people you hire for other things and tack that responsibility onto their others.  
  
#Thank you for a great class!-------------------------------------------------