Knightdale Public Comment 04-20-2023

Tabitha Hagen

4/21/2023

# Setup

# Load the required libraries

# Read in the data set then look at the rows/observations/individual entries and the columns/variables/observations

# We can use the function select() to choose and rename the columns we wish to keep:

collective\_comments <-comments\_original %>%   
 select (date = Date\_Submitted, name=Name, address=Address, subject=Public\_Comment\_Subject,  
 position = Please\_indicate\_if\_you\_are\_in\_favor\_in\_opposition\_or\_do\_not\_have\_a\_stated\_position\_and\_have\_a\_concern\_or\_neutral\_statement,   
 indiv\_comment = If\_commenting\_on\_a\_Public\_Hearing\_item\_please\_list\_specific\_reasons\_why\_you\_are\_in\_favor\_or\_opposed\_to\_the\_item)  
  
collective\_comments # view the data

## # A tibble: 104 × 6  
## date name address subject posit…¹ indiv…²  
## <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 4/20/2023 10:35 Stacey Richard 4014 Twin Spires Dr… Genera… In opp… "SAVE …  
## 2 4/19/2023 22:01 Josh Fisher 1601 Mallard Trace … Genera… In opp… "I opp…  
## 3 4/19/2023 21:45 Brittani Kern 2305 Marks Creek Rd… Genera… In opp… "Too m…  
## 4 4/19/2023 21:21 Amy Zielenski 1221 River Chase Dr… Genera… In opp… "If we…  
## 5 4/19/2023 20:28 Jill Harned 225 Hidden Shadow L… Genera… In opp… "Again…  
## 6 4/19/2023 20:00 Robert Marshburn 1536 Pleasants Road… Genera… In opp… "I am …  
## 7 4/19/2023 19:54 Debra Marshburn 1536 Pleasants Road… Genera… In opp… "We li…  
## 8 4/19/2023 18:36 Hunter Haynes 1217 Buttercup Lane… ZMA-15… In opp… "This …  
## 9 4/19/2023 18:03 Kelly Marshburn 450 Chaplin Street,… ZMA-15… In opp… "I am …  
## 10 4/19/2023 16:42 Mary Phillips 6601 LAKE MYRA RD, … ZMA-15… In opp… "This …  
## # … with 94 more rows, and abbreviated variable names ¹​position, ²​indiv\_comment

# Tokenize the text and transform it to a tidy data structure.

# Initial count of the words as a starting point  
  
tidy\_comments<- collective\_comments %>%  
 unnest\_tokens("word", indiv\_comment)%>% #separate into 1 word per doc per row  
 count(word, sort=TRUE) %>% # Count the words  
 arrange(desc(n)) #arrange in descending order of the count  
  
dim(tidy\_comments) # view the data (number of rows/variables, number of columns/observations)

## [1] 1107 2

# Preprocess the text to take out:

# - unnecessary words

# Remove common words such as “the”,“for”,“to” etc.  
  
 data("stop\_words") # uses previously defined stop words  
 tidy\_comments<-tidy\_comments %>%  
 anti\_join(stop\_words) # extracts pre-defined stopping words from dataframe   
   
dim(tidy\_comments) # view the data (number of rows/variables, number of columns/observations)

## [1] 834 2

# Preprocess the text to take out:

# - custom list of unnecessary words

# remove custom list of undesirable words  
undesirable\_words <- c("knightdale", "smithfield", "poole", "wendell","county", "add", "please", "already", "wake", "county", "would", "rd", "project", "development", "also", "need", "increase", "marks", "near", "area", "additional", "much", "high", "mark", "mark's","myra", "requirement", "voice", "current", "goal", "apartment")  
 tidy\_comments<-tidy\_comments %>%  
 filter (!word %in% undesirable\_words)  
   
dim(tidy\_comments) # view the data (number of rows/variables, number of columns/observations)

## [1] 818 2

# Preprocess the text to take out:

# - Stemming a word refers to replacing it with its most basic conjugate form.

# Stemming is common practice because we don’t want the words “type” and “typing” to convey different meanings to algorithms.  
  
tidy\_comments<-tidy\_comments %>%  
 mutate\_at("word", list(~wordStem((.), language="en"))) # using SnowballC pkg  
  
dim(tidy\_comments) # view the data (number of rows/variables, number of columns/observations)

## [1] 818 2

# Preprocess the text to take out:

# - any punctuation

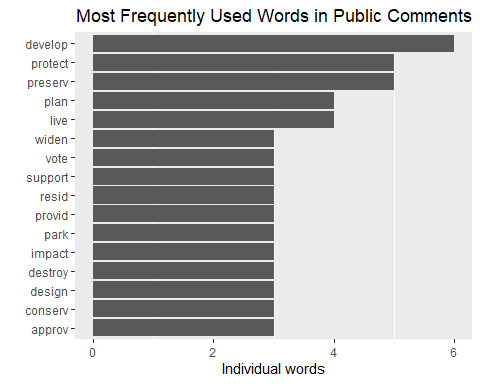
# - any non-alpha characters

# Continued pre-processing of number of characters, punctuation, and non-alpha characters  
tidy\_comments<- tidy\_comments %>%  
 filter(!str\_detect(word, "^\\b\\d+\\b"), # keep only words  
 !str\_detect(word, "\\s+"), # take out punctuation  
 !str\_detect(word, "[^a-zA-Z]")) # keep only alpha characters  
  
tidy\_comments # view the data

## # A tibble: 774 × 2  
## word n  
## <chr> <int>  
## 1 traffic 62  
## 2 creek 56  
## 3 land 49  
## 4 lake 46  
## 5 road 46  
## 6 town 36  
## 7 watersh 33  
## 8 rural 28  
## 9 natur 26  
## 10 peopl 22  
## # … with 764 more rows

# Count the most popular words in the text.

tidy\_comments %>%  
 count(word, sort=TRUE) %>% # Count the words to determine the most commonly used words.  
 top\_n(10) %>%  
 ungroup() %>% #Because count() performs a group\_by() on the word column  
 mutate(word = reorder(word, n)) %>%  
 ggplot() +  
 geom\_col(aes(word, n)) +  
 theme(legend.position = "none",   
 plot.title = element\_text(hjust = 0.5),  
 panel.grid.major = element\_blank()) +  
 xlab("") +   
 ylab("Individual words") +  
 ggtitle("Most Frequently Used Words in Public Comments") +  
 coord\_flip()



# Create a tidytext and tokenize by bigrams.

#Create a tidytext dataframe  
  
bigrams <- collective\_comments %>%  
 unnest\_tokens(bigram, indiv\_comment, token = "ngrams", n = 2) #create bigrams of 2 words  
  
dim(bigrams) # view the data (number of rows/variables, number of columns/observations)

## [1] 7280 6

head(bigrams) #view an extract of the tidytext dataframe

## # A tibble: 6 × 6  
## date name address subject posit…¹ bigram  
## <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 4/20/2023 10:35 Stacey Richard 4014 Twin Spires Drive,… Genera… In opp… save …  
## 2 4/20/2023 10:35 Stacey Richard 4014 Twin Spires Drive,… Genera… In opp… marks…  
## 3 4/20/2023 10:35 Stacey Richard 4014 Twin Spires Drive,… Genera… In opp… creek…  
## 4 4/20/2023 10:35 Stacey Richard 4014 Twin Spires Drive,… Genera… In opp… save …  
## 5 4/20/2023 10:35 Stacey Richard 4014 Twin Spires Drive,… Genera… In opp… lake …  
## 6 4/20/2023 10:35 Stacey Richard 4014 Twin Spires Drive,… Genera… In opp… myra …  
## # … with abbreviated variable name ¹​position

# Separate the bigram into Columns Word 1 and Word 2

#Separate the bigrams  
  
bigrams\_separated <- bigrams %>%  
 # separates n-gram into n columns, "word1", "word2", .., "wordn"  
 separate(bigram, c("word1", "word2"), sep = " ")   
  
# view the data (number of rows/variables, number of columns/observations)  
dim (bigrams\_separated)

## [1] 7280 7

head(bigrams\_separated) #view an extract of the tidytext dataframe

## # A tibble: 6 × 7  
## date name address subject posit…¹ word1 word2  
## <chr> <chr> <chr> <chr> <chr> <chr> <chr>  
## 1 4/20/2023 10:35 Stacey Richard 4014 Twin Spires D… Genera… In opp… save marks  
## 2 4/20/2023 10:35 Stacey Richard 4014 Twin Spires D… Genera… In opp… marks creek  
## 3 4/20/2023 10:35 Stacey Richard 4014 Twin Spires D… Genera… In opp… creek save   
## 4 4/20/2023 10:35 Stacey Richard 4014 Twin Spires D… Genera… In opp… save lake   
## 5 4/20/2023 10:35 Stacey Richard 4014 Twin Spires D… Genera… In opp… lake myra   
## 6 4/20/2023 10:35 Stacey Richard 4014 Twin Spires D… Genera… In opp… myra if   
## # … with abbreviated variable name ¹​position

#Pre-Process Text by Removing numbers, whitespaces, undesirable words, and stop words, words with less than 3 characters, etc.

# Pre-process tidytext dataframe  
  
# custom list of undesirable words  
undesirable\_words <- c("knightdale", "smithfield", "poole", "wendell","county", "add", "please", "already", "wake", "county", "would", "rd", "project", "development", "also", "need", "increase", "marks", "near", "area", "additional", "much", "high", "mark", "mark's","myra", "requirement", "voice", "current", "goal", "apartment")  
  
bigrams\_separated$word1 <- gsub("[^a-zA-Z]","", bigrams\_separated$word1) # only use alpha characters  
bigrams\_separated$word2 <- gsub("[^a-zA-Z]","", bigrams\_separated$word2) # only use alpha characters  
bigrams\_separated$word1 <- gsub("\\s+","", bigrams\_separated$word1) # get rid of whitespace  
bigrams\_separated$word2 <- gsub("\\s+","", bigrams\_separated$word2) # get rid of whitespace  
  
bigrams\_filtered <- bigrams\_separated %>%  
 # remove undesirable\_words  
 filter(!word1 %in% undesirable\_words) %>%  
 filter(!word2 %in% undesirable\_words) %>%  
 # remove stop\_words  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
# view the data (number of rows/variables, number of columns/observations)  
dim (bigrams\_filtered)

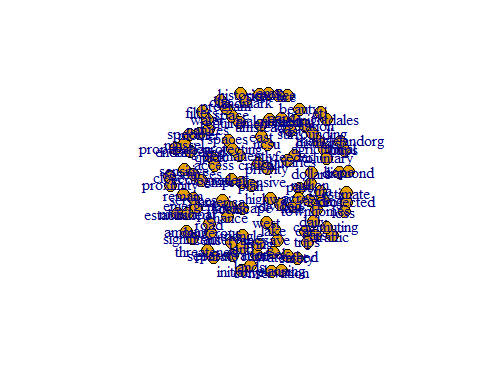
## [1] 927 7

# Count the most common bigrams.

# new bigram counts  
bigram\_counts <- bigrams\_filtered %>%   
 count(word1, word2, sort = TRUE)

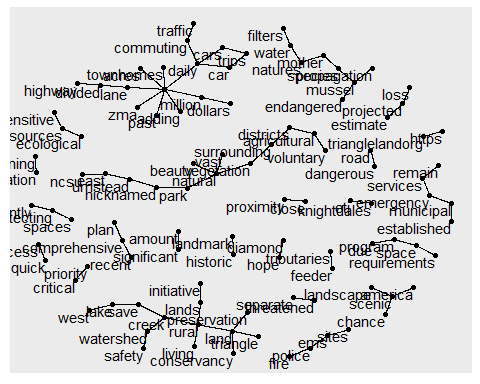
# Build a network of common bigrams

# filter for only relatively common combinations  
bigram\_graph\_common <- bigram\_counts %>%  
 filter(n > 2) %>% #include only repeated words  
 graph\_from\_data\_frame((directed = FALSE))  
  
plot(bigram\_graph\_common) # view simple graph



#create a better network graph  
library(ggraph)  
set.seed(2016)  
  
ggraph(bigram\_graph\_common, layout = "fr") +  
 #add edge\_alpha to make links transparent based on how common or rare the bigram is  
 geom\_edge\_link() +  
 geom\_node\_point() +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1)

## Warning: Using the `size` aesthetic in this geom was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` in the `default\_aes` field and elsewhere instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



# Visualize the graph using the Fruchterman-Reingold to visualize the nodes and ties (“fr”). Applying some polishing operations to make a better looking graph.

#plot the graph of bigrams  
  
set.seed(2017)  
  
ggraph(bigram\_graph\_common, layout = "fr") +  
 geom\_edge\_link(aes(edge\_alpha = n, edge\_width = n), show.legend = FALSE,edge\_colour = "cyan4") +  
 geom\_node\_point(size = 1) +  
 geom\_node\_text(aes(label = name), vjust = 1, hjust = 1) +  
 theme\_void()

