**U.S. Office of Disease Prevention and Health Promotion (ODPHP) Archive:**

**Sticking Points and Evolution of Attitudes and Discussion Around Public Health**

Katie Eiswerth

2024-06-11

Introduction

The U.S. Office of Disease Prevention and Health Promotion (ODPHP) is a group that works within the Office of Health and Human Services with the aim of improving health literacy and equitability through education, science-backed health guidelines, and promoting public-health priorities. The work of the ODPHP is segmented into "National Health Initiatives" and "Nutrition and Physical Activity". The national health initiatives include equitable long-term recovery and resilience, healthy people, healthy aging, and health literacy. Nutrition and physical activity resources include dietary guidelines, physical activity guidelines, Move Your Way community resources, national youth sports strategies, President's Council, Food is Medicine, and the White House Conference on Hunger, Nutrition and Health. One responsibility of the ODPHP is to manage the health.gov website which, in part, maintains a news article archive. This archive was the basis of this research project, as it is the culmination of hundreds of health and public-health related articles from 2017 to present. The news archive can provide insight into the sentiment of the articles, the most important words in the articles, the most discussed topics by the ODPHP, and how this may have changed over the years. More specifically, the news archive will aid in answering the following questions:

Sentiment Related Questions:

1. How does sentiment change over the years? How does sentiment change by month, and can we observe any seasonality? How does sentiment change over time by month and year?

Word Importance Related Questions:

2. What are the most important words each year? Each month? Can we observe any seasonality or patterns?

Topic Related Questions:

3. What topics comprise our data and how can we characterize them? What is the word-topic probabilities and what are the document-topic probabilities?

Through this analysis, we hope to gain insights into what the conversation around health and public health consists of, what the overarching problems are, what the sentiment tends to be around these problems, and how this may have changed in the past 6 years. These insights are crucial in understanding our current public health landscape and what challenges cause the greatest barrier to achieving quality health. In doing so, we can be better equipped to tackle both current and future public health challenges

Scraping the health.gov Archive Data

The first step in this project is to get the data, which must be scraped from the news article archive. The health.gov news archive is organized by month and year. For example, one could navigate to the “September 2019” section to find articles published in September 2019. The archive houses articles published between December 2017 and June 2024, so there are 76 month-year sections total. Within these month-year sections, there are, on average, 3-10 articles. Therefore, accessing the article text is a multi-step process. First, we must scrape the urls for each of the 76 sections. Then, we can build a loop that, for each section, will scrape all of the article urls. Additionally, some of the sections have a “load more” button at the bottom of the page that the user must click to view all of the articles in that section. This only occurred when the section had about 5 or more articles, so not every section has a “load more” button. To ensure that, if a section had a “load more” button, R would go into the second page to scrape the remaining article urls, the “Rselenium” package was used and an “if statement” was built into the loop. The final step of this process was to scrape the body of text of each article and compile the text in a tibble. To aid in future analysis, the tibble includes a column indicating the section, month, and year for each body of text. Figure 1 displays the code used to retrieve the data.

Figure 1

## SECTION 1: SCRAPING THE DATA   
#1A: GET WEBPAGE URL AND SCRAPE SECTION URLS   
## URL TO NEWS ARCHIVE PAGE  
#create object that holds this url  
healthgov\_url <- "https://health.gov/news/archive?page=0"

## URLS FOR EVERY ARCHIVE SECTION  
section\_urls <- healthgov\_url %>%   
 read\_html() %>%   
 html\_elements("#block-views-block-archive-all-news-block-1 a") %>%   
 html\_attrs() %>%   
 tibble() %>%   
 rename(web = ".") %>%   
 unnest(cols = web) %>%   
 mutate(url = paste0("https://health.gov", web)) %>%  
 pull(url)

##GET THE NAMES OF THE SECTIONS  
section\_names <- healthgov\_url %>%   
 read\_html() %>%   
 html\_elements("#block-views-block-archive-all-news-block-1 a") %>%   
 html\_text()

#if section names are needed in tibble format later on:  
section\_names\_tib<- section\_names %>%   
 tibble() %>%   
 rename("section"=".") %>%   
 mutate(id = 1:76)  
  
sections\_df <- tibble(url = section\_urls, section = section\_names)  
sections\_df

## # A tibble: 76 × 2  
## url section   
## <chr> <chr>   
## 1 https://health.gov/news/archive/202406 June 2024   
## 2 https://health.gov/news/archive/202405 May 2024   
## 3 https://health.gov/news/archive/202404 April 2024   
## # ℹ 66 more rows

##1B: GET TEXT OF ARTICLES ON FIRST PAGE  
#GET ARTICLE URLS  
article\_urls <- list()  
for(i in 1:length(section\_urls)){  
 article\_urls[[i]] <- section\_urls[i] %>%  
 read\_html() %>%   
 html\_elements(".c-teaser\_\_link") %>%   
 html\_attrs() %>%   
 tibble() %>%   
 rename(web = ".") %>%   
 unnest(cols = web) %>%  
 filter(str\_detect(web,"/")) %>%   
 mutate(url = paste0("https://health.gov", web)) %>%   
 pull(url)   
}  
  
article\_urls\_sections <- article\_urls %>% tibble() %>% bind\_cols(sections\_df) %>% select(-url) %>% unnest(cols=".")

##LIST OF ARTICLE URLS ON THE FIRST PAGE  
View(article\_urls\_sections)

##GET TEXT FROM ARTICLES  
firstpage\_text <- list()  
  
for (i in 1:nrow(article\_urls\_sections)) {  
   
 firstpage\_url<-article\_urls\_sections[i, "."]  
   
 firstpage\_article\_text<-firstpage\_url %>%   
 as.character() %>%   
 read\_html() %>%   
 html\_elements("#block-healthgov-content .field\_\_item p") %>%   
 html\_text() %>%   
 tibble() %>%   
 mutate(section = article\_urls\_sections$section[i])  
   
 firstpage\_text[[i]]<-firstpage\_article\_text  
}  
  
firstpage\_text

scraped\_firstpage\_text <- bind\_rows(firstpage\_text) %>% rename(text=".")  
View(scraped\_firstpage\_text)

##1C: GET TEXT OF ARTICLES AFTER LOAD MORE  
  
# if the section has a "load more" button  
has\_load\_more <- function(healthgov\_url) {  
 page <- read\_html(healthgov\_url)  
 return(".c-btn--next" %in% html\_attr(page, "class"))  
}

# Loop through each section URL  
loadmore\_article\_urls <- list()  
for (i in 1:nrow(sections\_df)) {  
 section\_url <- sections\_df$url[i]  
 section\_name <- sections\_df$section[i]  
   
 # Check if the section has a "load more" button  
 if (has\_load\_more(section\_url)) {  
 # If it has a "load more" button, simulate clicking and extract additional article URLs  
 additional\_urls <- get\_additional\_article\_urls(section\_url)  
 if (!is.null(additional\_urls)) {  
 loadmore\_article\_urls[[length(loadmore\_article\_urls) + 1]] <- data.frame(url = additional\_urls, section = section\_name)  
 }  
 }  
   
 # Extract initial article URLs  
 initial\_urls <- section\_url %>%  
 read\_html() %>%  
 html\_elements(".c-teaser\_\_link") %>%  
 html\_attrs() %>%  
 tibble() %>%  
 rename(web = ".") %>%  
 unnest(cols = web) %>%  
 filter(str\_detect(web, "/")) %>%  
 mutate(url = paste0("https://health.gov", web),  
 section = section\_name)  
   
 loadmore\_article\_urls[[length(loadmore\_article\_urls) + 1]] <- initial\_urls  
}

# Combine all article URLs into a single data frame  
all\_loadmore\_article\_urls <- do.call(rbind, loadmore\_article\_urls) %>% select(url,section)   
View(all\_loadmore\_article\_urls)

# scrape text from article URL  
scrape\_article\_text <- function(url) {  
  
 page <- read\_html(url)  
 article\_text <- page %>%  
 html\_elements("#block-healthgov-content .field\_\_item p") %>%  
 html\_text() %>%  
 paste(collapse = "\n") # Combine multiple paragraphs into a single string  
 return(article\_text)  
}

# Add column to store article text  
all\_loadmore\_article\_urls <- all\_loadmore\_article\_urls %>%  
 mutate(article\_text = "")

# Loop through each row of the tibble containing article URLs and section names  
for (i in 1:nrow(all\_loadmore\_article\_urls)) {  
 # Scrape text from the article URL  
 article\_url <- all\_loadmore\_article\_urls$url[i]  
 article\_text <- scrape\_article\_text(article\_url)  
 # Update the tibble with the scraped article text  
 all\_loadmore\_article\_urls$article\_text[i] <- article\_text  
}

# View the updated tibble with article text  
View(all\_loadmore\_article\_urls)  
View(scraped\_firstpage\_text)

scraped\_loadmore\_text <- all\_loadmore\_article\_urls %>% select(-url) %>% rename("text"="article\_text")  
scraped\_loadmore\_text <- scraped\_loadmore\_text[, c(2, 1)]

all\_text<- scraped\_firstpage\_text %>% bind\_rows(scraped\_loadmore\_text)

Preparing the Data

After retrieving the data, the next step is to tidy and prepare the data for analysis. This includes using R’s unnest\_tokens() function to tidy the data and remove stop words, numbers, punctuation, and whitespaces. The unit of analysis for this project is one word, or unigram. The removal of stopwords included both R’s package of common stopwords as well as some stopwords that are unique to this corpus. The stopwords that are unique to this corpus included words like “health”, “public” and “community”, as well as names of people and places that were adding noise to the results of the analyses and diluting the meaningful insights. The choice to remove the names of people and places was specific to this use case and the goal of the analysis. Although some analyses would be interested to know who are the major players and spokespeople and where in the United States discussions and initiatives around public health are occurring, this is not pertinent to the research questions investigated in this project. Instead, the abundance of names and locations in the data was cluttering the results and making it difficult to identifier other, more relevant, patterns and trends. Figure 2 provides the code used to tidy and prepare the data, including commentary on why various stopwords were removed.

Figure 2

## SECTION 2: TIDY THE TEXT DATA AND REMOVE STOP WORDS

## # A tibble: 3,138 × 2  
## text section  
## <chr> <chr>   
## 1 ICYMI: A few weeks ago, the 2025 Dietary Guidelines Advisory Committ… June 2…  
## 2 The meeting’s first day featured engaging discussions of the Committ… June 2…  
## 3 The meeting’s second day began with an update from federal staff on … June 2…  
June 2…  
## # ℹ 3,128 more rows

##Tidy the Text  
tidy\_text<- all\_text %>%   
 unnest\_tokens(input=text,output=word)

## Remove stop words that are included in R's stopword list  
tidy\_text <- tidy\_text %>%   
 anti\_join(stop\_words)

## Joining with `by = join\_by(word)`

## Remove numbers  
tidy\_text$word <- gsub('[0-9]+', '', tidy\_text$word)

## Remove punctuation  
tidy\_text$word <- gsub('[[:punct:]]+', '', tidy\_text$word, perl = TRUE)

## Remove any empty strings  
tidy\_text <- tidy\_text %>%  
 filter(word != "")

##View the most common words and determine if there are unique stop words to remove  
top\_50\_words <- tidy\_text %>%   
 count(word) %>%   
 arrange(desc(n)) %>%   
 filter(n>200)  
View(top\_50\_words)

## Remove additional stop words that are unique to this use case.  
health\_archive\_stop\_words <- tibble(word = c("office", #public health "health",  
 "healthy", #public health  
 "disease", #public health  
 "prevention", #public health  
 "promotion", #public health  
 "secretary", #public health  
 "website", #public health  
 "people", #public health  
 "odphp", #public health  
 "human", #public health  
 "information", #public health  
 "care", #public health  
 "improve", #public health  
 "federal", #public health  
 "services", #public health  
 "activity", #public health  
 "physical", #public health  
 "community", #public health  
 "public", #public health  
 "policy", #public health  
 "communities", #public health  
 "resources", #public health  
 "grigsby", #name  
 "fuentes", #name  
 "huds", #name  
 "mariana",#location  
 "maloney", #name  
 "detroit",#location  
 "minnesota",#location  
 "minneapolis",#location  
 "catawba",#location  
 "fcpa",#location  
 "snhd",#location  
 "swv",#location  
 "sacramento",#location  
 "buckhead",#location  
 "tukwila",#location  
 "mallya", #name  
 "boulder",#location  
 "cpd",#location  
 "mdh",#location  
 "seiler", #name  
 "jernigan" #name),   
lexicon = c("katie"))  
health\_archive\_stop\_words

## # A tibble: 44 × 2  
## word lexicon  
## <chr> <chr>   
## 1 office katie   
## 2 health katie   
## 3 healthy katie   
## # ℹ 34 more rows

stop\_words <- stop\_words %>%   
 bind\_rows(health\_archive\_stop\_words)

#remove the stop words from the tidy text  
tidy\_text <- tidy\_text %>% anti\_join(stop\_words)

## Joining with `by = join\_by(word)`

# add a year column and month column to cleaned text  
section\_year <- str\_sub(string=tidy\_text$section, start=-4)  
section\_month <- str\_sub(string=tidy\_text$section, end=-6)  
  
tidy\_text<-tidy\_text %>%   
 mutate(year = section\_year) %>%   
 mutate(month = section\_month)  
  
tidy\_text

## # A tibble: 107,924 × 4  
## section word year month  
## <chr> <chr> <chr> <chr>  
## 1 June 2024 icymi 2024 June   
## 2 June 2024 weeks 2024 June   
## 3 June 2024 ago 2024 June   
## 4 June 2024 dietary 2024 June   
## 5 June 2024 guidelines 2024 June   
## 6 June 2024 advisory 2024 June   
## 7 June 2024 committee 2024 June   
## 8 June 2024 held 2024 June   
## 9 June 2024 meeting 2024 June   
## 10 June 2024 meeting 2024 June   
## # ℹ 107,914 more rows

Exploratory Data Analysis

Following the tidying and preparation of the data is the exploratory data analysis. This includes viewing the total number of documents, the earliest and most recent time periods represented, the most commonly used words in all of the archive articles, and total sentiment over time. There are 76 total documents, and they are printed in date form in Figure 3 after the “unique\_sections” command is called. It is important to note that, for the basis of this project, month-year sections were used as the document. However, there are certain times where the data is broken down either by year or month, instead of section. The reasoning for breaking the data down by year is to more clearly see changes over time, and the reasoning for breaking the data down by month is to identify any seasonality. The data represents 7 years and 12 months. The earliest time period can be found by calling the minimum of unique sections, which is December 2017, and a similar process using “max()” can find the latest time period, which is June, 2024. Figure 4 displays the top used words in the archive. They include national, guidelines, sports, social, objectives, adult, and move. Finally, Figure 5 shows the sentiment over time. This was found using R’s “get\_sentiments()” function that associates a positive or negative sentiment with every word in the corpus. We can then calculate the total sentiment by subtracting negative sentiment from positive sentiment. This total sentiment variable is then plotted across all 76 month-year sections. We see that, for the most part, overall sentiment is positive. There are particular peaks in positive sentiment in Fall 2018 and Fall 2020. The reasoning behind these peaks is discussed and investigated in the word importance portion of the project. There are several times where the sentient is negative, with a large dip in April 2022. To investigate why sentiment seems to dip so significantly in April of 2022, we looked only at text data from this time period to see the most commonly used words. Figure 3 shows the most common words to be elder, abuse, equity, questions, outcomes, justice, and conditions. It is likely that the topics of elder abuse and justice are driving the negative sentiment. A final observation from figure 5 is that there seems to be a gap in the data around August, 2019. Further investigation on this found that the health.gov archive does not include sections for August and September 2019. The reason for this is unknown, but it appears that there is a gap in archived articles published during this time period.

Figure 3

A graph with blue squares

Description automatically generated

## SECTION 3: EDA  
#Number of sections:  
tidy\_text\_chronological <- tidy\_text %>%   
 mutate(section = as.Date(paste0("01", section), format = "%d %B %Y"))  
tidy\_text\_chronological <- tidy\_text\_chronological %>% arrange(section)  
tidy\_text\_chronological <- tidy\_text %>%   
 mutate(section = as.Date(paste0("01", section), format = "%d %B %Y"))  
unique\_sections <- tidy\_text\_chronological$section %>% sort() %>% unique()  
unique\_sections

## [1] "2017-12-01" "2018-01-01" "2018-02-01" "2018-03-01" "2018-04-01"  
## [6] "2018-05-01" "2018-06-01" "2018-07-01" "2018-08-01" "2018-09-01"  
## [11] "2018-10-01" "2018-11-01" "2018-12-01" "2019-01-01" "2019-02-01"  
## [16] "2019-03-01" "2019-04-01" "2019-05-01" "2019-06-01" "2019-07-01"  
## [21] "2019-11-01" "2019-12-01" "2020-01-01" "2020-02-01" "2020-03-01"  
## [26] "2020-04-01" "2020-05-01" "2020-06-01" "2020-07-01" "2020-08-01"  
## [31] "2020-09-01" "2020-10-01" "2020-11-01" "2020-12-01" "2021-01-01"  
## [36] "2021-02-01" "2021-03-01" "2021-04-01" "2021-05-01" "2021-06-01"  
## [41] "2021-07-01" "2021-08-01" "2021-09-01" "2021-10-01" "2021-11-01"  
## [46] "2021-12-01" "2022-01-01" "2022-02-01" "2022-03-01" "2022-04-01"  
## [51] "2022-05-01" "2022-06-01" "2022-07-01" "2022-08-01" "2022-09-01"  
## [56] "2022-10-01" "2022-11-01" "2022-12-01" "2023-01-01" "2023-02-01"  
## [61] "2023-03-01" "2023-04-01" "2023-05-01" "2023-06-01" "2023-07-01"  
## [66] "2023-08-01" "2023-09-01" "2023-10-01" "2023-11-01" "2023-12-01"  
## [71] "2024-01-01" "2024-02-01" "2024-03-01" "2024-04-01" "2024-05-01"  
## [76] "2024-06-01"

#earlest date  
unique\_sections %>% min()

## [1] "2017-12-01"

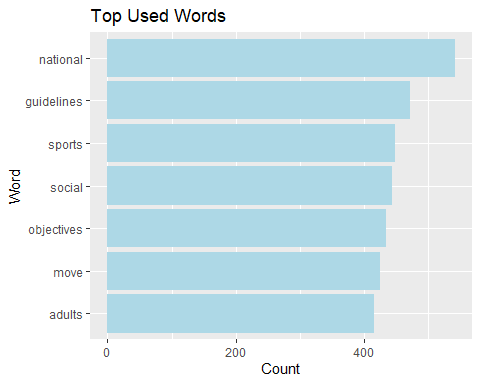
#most recent date  
unique\_sections %>% max()

## [1] "2024-06-01"

##View the most common words again now that stop words have been added   
top\_words <- tidy\_text %>%   
 count(word) %>%   
 arrange(desc(n)) %>%   
 filter(n>100)  
View(top\_words)

top\_words\_plot <- tidy\_text %>%   
 count(word) %>%   
 arrange(desc(n)) %>%   
 filter(n>400) %>%   
 ggplot() +  
 geom\_col(aes(x=n, y=reorder(word,n)),fill="lightblue") +  
 ggtitle("Top Used Words") +   
 xlab("Count") +  
 ylab("Word")  
top\_words\_plot

Figure 4



#Graph of Total Sentiment Over Time  
tidy\_text\_chronological <- tidy\_text %>%   
 mutate(section = as.Date(paste0("01", section), format = "%d %B %Y"))  
  
tidy\_text\_chronological <- tidy\_text\_chronological %>% arrange(section)

# Check the minimum value of the section variable  
min(tidy\_text\_chronological$section)

## [1] "2017-12-01"

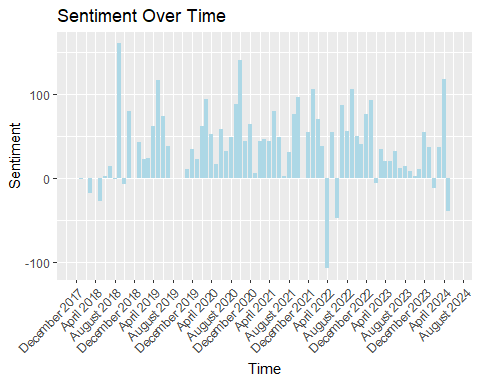
# Print the first few rows of the data frame  
head(tidy\_text\_chronological)

## # A tibble: 6 × 4  
## section word year month   
## <date> <chr> <chr> <chr>   
## 1 2017-12-01 steering 2017 December  
## 2 2017-12-01 committee 2017 December  
## 3 2017-12-01 infections 2017 December  
## 4 2017-12-01 hais 2017 December  
## 5 2017-12-01 written 2017 December  
## 6 2017-12-01 phase 2017 December

# Specify the limits of the x-axis because it was cutting off the first 8 months   
limits <- c(as.Date("2017-12-01"), max(tidy\_text\_chronological$section))  
  
sentiment\_over\_time\_plot <- tidy\_text\_chronological %>%  
 inner\_join(get\_sentiments()) %>%  
 count(section, sentiment) %>%  
 pivot\_wider(names\_from = sentiment, values\_from = n) %>%  
 mutate(sent = positive - negative) %>%  
 ggplot() +  
 geom\_col(aes(x = section, y = sent), fill = "lightblue") +   
 ggtitle("Sentiment Over Time") +   
 xlab("Time") +  
 ylab("Sentiment") +  
 scale\_x\_date(date\_breaks = "4 months", date\_labels = "%B %Y", limits=limits) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

sentiment\_over\_time\_plot

Figure 5



Sentiment Analysis

**The first research asks the following**:

How does sentiment change over the years? How does sentiment change by month, and can we observe any seasonality? How does sentiment change over time by month-year?

To answer this, we will implement sentiment analysis using the AFINN lexicon. This lexicon uses a numeric scale of -5 to +5 to indicate the sentiment of a word. The more negative the value, the more negative the sentiment. The more positive the value, the more positive the sentiment. A couple of exploratory tasks for the sentiment analysis were conducted before these analyses. These included viewing the top 20 words and their associated sentiment in the data to get an understanding of how AFINN sentiment assigns scores, and the second was looking at overall sentiment by year. However, it would be more fruitful to break overall sentiment down into months, so this was the next step. Looking at total sentiment by month, we see sentiment is highest in May and September and lowest in February and August. Looking at average sentiment by month provides a slightly different insight, showing the highest average sentiment occurring in September and October and the lowest occurring in February and April. We see two commonalities when looking at total and average sentiment over the months. First, September tends to have high sentiment and, second, February tends to have low sentiment. The high sentiment seen in September resembles the high peaks of positive sentiment in Fall of 2018 and 2020 that we observed in the exploratory data analysis. It seems that there may be something driving increased positive sentiment around this time of year, that was especially impacted in 2018 and 2020. This leads us to wonder what may be driving these patterns, and this question will be addressed in the discussion on word importance.

Figure 6

##SECTION 4: SENTIMENT ANALYSIS  
get\_sentiments()

## # A tibble: 6,786 × 2  
## word sentiment  
## <chr> <chr>   
## 1 2-faces negative   
## 2 abnormal negative   
## 3 abolish negative   
## 4 abominable negative   
## 5 abominably negative   
## 6 abominate negative   
## 7 abomination negative   
## 8 abort negative   
## 9 aborted negative   
## 10 aborts negative   
## # ℹ 6,776 more rows

#view the afinn sentiment scores of the first 20 words just to see how the rating system works  
afinn\_lexicon <- tidy\_text\_chronological %>%   
 inner\_join(get\_sentiments("afinn"),  
 relationship="many-to-many") %>% head(20)

## Joining with `by = join\_by(word)`

afinn\_lexicon

## # A tibble: 20 × 5  
## section word year month value  
## <date> <chr> <chr> <chr> <dbl>  
## 1 2017-12-01 prevent 2017 December -1  
## 2 2017-12-01 share 2017 December 1  
## 3 2017-12-01 importance 2017 December 2  
## 4 2017-12-01 prevent 2017 December -1  
## 5 2017-12-01 invite 2017 December 1  
## 6 2017-12-01 cancer 2017 December -1  
## 7 2017-12-01 cancer 2017 December -1  
## 8 2017-12-01 united 2017 December 1  
## 9 2017-12-01 save 2017 December 2  
## 10 2017-12-01 cancer 2017 December -1  
## 11 2017-12-01 recommends 2017 December 2  
## 12 2017-12-01 cancer 2017 December -1  
## 13 2017-12-01 cancer 2017 December -1  
## 14 2017-12-01 shared 2017 December 1  
## 15 2017-12-01 cancer 2017 December -1  
## 16 2017-12-01 cancer 2017 December -1  
## 17 2017-12-01 easy 2017 December 1  
## 18 2017-12-01 share 2017 December 1  
## 19 2017-12-01 cancer 2017 December -1  
## 20 2017-12-01 encourages 2017 December 2

#total sentiment by year  
afinn\_sentiment\_year <- tidy\_text\_chronological %>%   
 inner\_join(get\_sentiments("afinn"),  
 relationship="many-to-many") %>%   
 group\_by(year) %>%   
 summarize(totalval = sum(value))

## Joining with `by = join\_by(word)`

afinn\_sentiment\_year

## # A tibble: 8 × 2  
## year totalval  
## <chr> <dbl>  
## 1 2017 10  
## 2 2018 898  
## 3 2019 1214  
## 4 2020 1924  
## 5 2021 1600  
## 6 2022 1018  
## 7 2023 1034  
## 8 2024 742

#total sentiment by month  
months\_ordered <- c("January", "February", "March", "April", "May", "June",   
 "July", "August", "September", "October", "November", "December")  
tidy\_text\_chronological$month <- factor(tidy\_text\_chronological$month, levels = months\_ordered, ordered = TRUE)  
  
afinn\_total\_sentiment\_month <- tidy\_text\_chronological %>%   
 inner\_join(get\_sentiments("afinn"),  
 relationship="many-to-many") %>%   
 group\_by(month) %>%   
 summarize(totalval = sum(value))

## Joining with `by = join\_by(word)`

afinn\_total\_sentiment\_month

## # A tibble: 12 × 2  
## month totalval  
## <ord> <dbl>  
## 1 January 768  
## 2 February 490  
## 3 March 742  
## 4 April 668  
## 5 May 1042  
## 6 June 610  
## 7 July 594  
## 8 August 368  
## 9 September 1100  
## 10 October 676  
## 11 November 616  
## 12 December 766

#total sentiment by month plot  
afinn\_sentiment\_month\_plot <- afinn\_total\_sentiment\_month %>%   
 ggplot() +  
 geom\_col(aes(x=month,  
 y=totalval), fill = "lightblue")+  
 ggtitle("Total Sentiment by Month") +   
 xlab("Month") +  
 ylab("Total Sentiment") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))   
afinn\_sentiment\_month\_plot

Figure 7

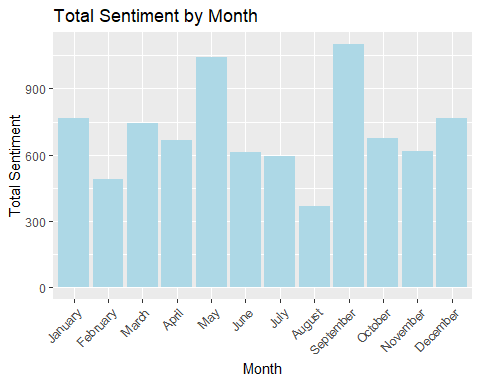


Figure 8

#ave sentiment by month  
months\_ordered <- c("January", "February", "March", "April", "May", "June",   
 "July", "August", "September", "October", "November", "December")  
tidy\_text\_chronological$month <- factor(tidy\_text\_chronological$month, levels = months\_ordered, ordered = TRUE)  
  
afinn\_ave\_sentiment\_month <- tidy\_text\_chronological %>%   
 inner\_join(get\_sentiments("afinn"),  
 relationship="many-to-many") %>%   
 group\_by(month) %>%   
 summarize(aveval = mean(value))

## Joining with `by = join\_by(word)`

afinn\_ave\_sentiment\_month

## # A tibble: 12 × 2  
## month aveval  
## <ord> <dbl>  
## 1 January 0.859  
## 2 February 0.528  
## 3 March 0.784  
## 4 April 0.482  
## 5 May 0.812  
## 6 June 0.792  
## 7 July 0.809  
## 8 August 0.742  
## 9 September 1.09   
## 10 October 0.949  
## 11 November 0.723  
## 12 December 0.731

#ave monthly sentiment plot  
ave\_sentiment\_month\_plot <- afinn\_ave\_sentiment\_month %>%  
 ggplot() +  
 geom\_col(aes(x=month,   
 y=aveval),fill = "lightblue")+  
 ggtitle("Average Sentiment by Month") +   
 xlab("Month") +  
 ylab("Average Sentiment") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))   
ave\_sentiment\_month\_plot

Figure 9

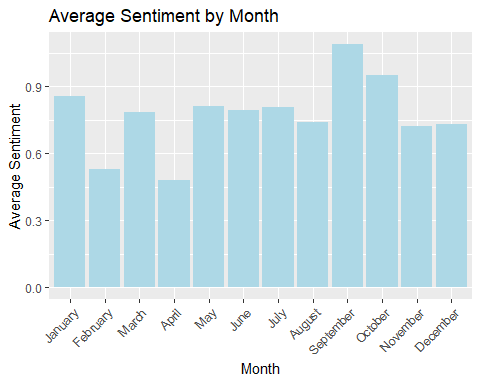


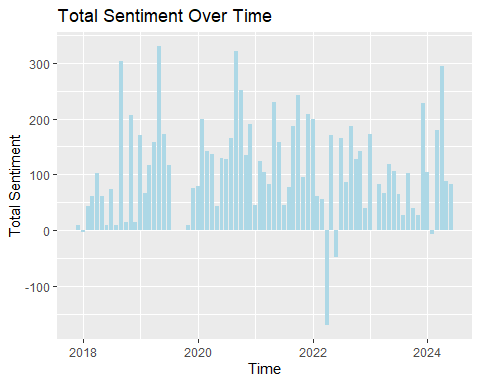
Figure 10

#total afinn sentiment by section plot  
total\_afinn\_sentiment\_plot <- tidy\_text\_chronological %>%   
 inner\_join(get\_sentiments("afinn"),  
 relationship="many-to-many") %>%   
 group\_by(section) %>%   
 summarize(totalval = sum(value)) %>%   
 ggplot() +  
 geom\_col(aes(x=section,   
 y=totalval),fill = "lightblue")+  
 ggtitle("Total Sentiment Over Time") +   
 xlab("Time") +  
 ylab("Total Sentiment")

## Joining with `by = join\_by(word)`

total\_afinn\_sentiment\_plot

Figure 11 Figure 5

A graph with blue and white lines

Description automatically generated

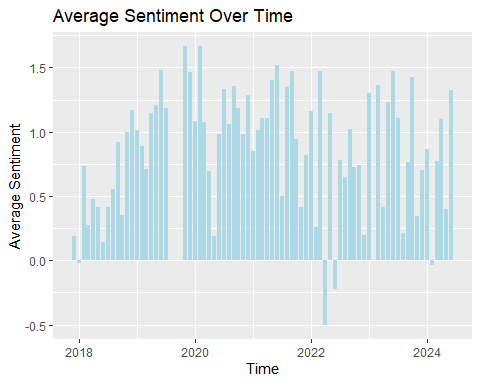
Figure 12

#average afinn sentiment by section plot  
ave\_afinn\_sentiment <- tidy\_text\_chronological %>%   
 inner\_join(get\_sentiments("afinn"),  
 relationship="many-to-many") %>%   
 group\_by(section) %>%   
 summarize(aveval = mean(value)) %>%   
 ggplot() +  
 geom\_col(aes(x=section,   
 y=aveval),fill = "lightblue")+  
 ggtitle("Average Sentiment Over Time") +   
 xlab("Time") +  
 ylab("Average Sentiment")

## Joining with `by = join\_by(word)`

ave\_afinn\_sentiment

Figure 13



The last step in the sentiment analysis is to look at total and average sentiment over time, using all 76 sections. By doing this, we can compare R’s default Bing sentiment breakdown, shown in figure 5, to what AFINN sentiment analysis can reveal about our data. Additionally, we may be able to discover patterns or insights that were not obvious when looking at the afinn sentiment by month. Figure 5 is provided, again, next to figure 11 to ease comparison. The comparison of total sentiment over time using the Bing lexicon and the afinn lexicon reveals that the afinn sentiment analysis finds more negative sentiment. There are more bars below the “0” line using the afinn lexicon. Specifically, the afinn lexicon shows negative sentiment in spring and summer of 2018 and 2024, where the Bing lexicon does not.

Word Importance

**To address the second research question, we can conduct TF-IDF analysis. The second question asks:**

2. What are the most important words each year? Each month? Can we observe any seasonality or patterns?

To discover the most important words for each year, we can use year as the document when using the “bind\_tf\_idf” function. Figure 14 demonstrates this code, and Figure 15 displays the top 5 words by tf\_idf score for each year.

**Below is a discussion on the most important words in each year.**

2017: Words center around “hais”, which stand for healthcare associated infections, and antibiotics.

2018: Words center around opioid use, preeclampsia (form of high blood pressure), and patient safety awareness week (psaw).

2019: Words center around educational efforts in recreation centers, college campuses, and classrooms, tobacco use reduction, and well as food vendors.

2020: Words center around efforts for people with disabilities, various regulations and laws, news outlets, and breastfeeding.

2021: Educational efforts for youth and children, Center for Advancing Research in Transportation Emissions, Energy and Health (CARTEEH), and National youth sports strategy (NYSS).

2022: The impact of firearms on public health, geriatric health, educational efforts, resilience, and social determinants of health (SDOH)

2023: Sickle cell disease (SCD), gut health, resilience, NHOS

2024: Loneliness, asthma, and NHOS

There are a couple elements of particular interest. First, it is interesting, and also logical, that opioid use seems to have been heavily discussed in 2018, as this was a time where opioid use and opioid related deaths were at a high. Second, the importance of the word “firearms” in 2022 is very interesting because it is a quite different topic than many of the other importance words, which represent common diseases or public health organizations. Lastly, the importance of the word “loneliness” in 2024 is also quite interesting because, again, it differs from the diseases and programs that are heavily discussed in the public health realm and, instead, represents a negative feeling experienced at the individual level. Additionally, we see that resilience is an important word in both 2022 and 2023, and the National Health Observances (nhos) appears to be relevant in 2023 and 2024. This indicates that these topics were particularly persistent in public health conversations.

Figure 14

## SECTION 5: WORD IMPORTANCE

#word importance by year  
TFIDF\_year\_plot <- tidy\_text %>%   
 count(year,word) %>%   
 bind\_tf\_idf(term = word,  
 document = year,  
 n = n) %>%   
 group\_by(year) %>%   
 top\_n(n = 5,  
 wt = tf\_idf) %>%   
 ggplot() +   
 geom\_col(aes(y = reorder(word,tf\_idf),  
 x = tf\_idf,  
 fill = year))+  
 facet\_wrap(~year,  
 scales = "free")+  
 ggtitle("Word Importance by Year") +   
 xlab("TF-IDF Score") +  
 ylab("Top Words") +  
 guides(fill = FALSE)

TFIDF\_year\_plot

Figure 15

A screenshot of a graph

Description automatically generated

To address the last piece of the research question, which asks about word importance by month and whether there is any seasonality or patterns to be seen, TF-IDF was conducted using month as the document. Figure 16 displays the code used to achieve this and Figure 17 displays the results.

**Below is a discussion of the most important words by month.**

January: Words center around pre-eclampsia and opioid use, which seemingly resemble the important words in 2018.

February: Words reflect the Center for Advancing Research in Transportation Emissions, Energy, and Health (CARTEEH) as well as hypertension. The words “girls” and “smart” may reflect education efforts for girls

March: Similar to January, we see high importance of words relating to opioid use. PSAW appears to be important, just as it was in 2018. This may be an indication that Patient Safety Awareness Week takes place in March, and that it received a particular large amount of attention in 2018.

April: Geriatric care and tobacco use.

May: Crime and media outlets

June: Anticoagulant medication and alcohol use.

July: Educational efforts at recreation centers and campuses and dietary guidelines.

August: Motherhood and vaccinations.

September: Food and dietary guidelines. This could be related to the start of the school year often occurring in September, causing greater momentum around health eating and habits for youth.

October: Educational efforts and guidelines. Similar to September, this may reflect an increased effort to improve health-related habits in schools at the beginning of the school year.

November: Support for disabled individuals, movement, access to care.

December: Firearm impacts on health, gut health, education. It is likely that the prevalence of discussion around firearms is driven by its popularity in 2022 and that, perhaps, it is not truly the month of December that sees this topic more commonly. It is possible that in December of 2022, there was an overwhelming about of conversation around this topic, and this impacted the overall word importances in December.

Investigating the word importance helps us address part of the first research question, where we saw a pattern of high sentiment in September and low sentiment in February. Looking at the word importances, we see that February primarily represents work and discussion around CARTEEH, hypertension and education for young girls. Although one would expect the topic of education for youth to be a positive one, it is possible that the discussion around hypertension and CARTEEH are the driving the sentiment. CARTEEH is an organization working on emissions, energy, and health. As concerns around pollution and climate change have grown, it is logical that this topic would be associated with a negative sentiment. Similarly, hypertension is estimated to be experienced by about 70% of Americans during their lifetimes and can result in negative health outcomes such as kidney disease, heart attack, or stroke. Therefore, it is also possible that concerns around hypertension drives this negative sentiment. September’s most important words center around food and dietary guidelines. Although obesity, childhood obesity, and malnutrition are very prevalent and heavily discussed topics in the realm of public health, it is possible that they may be lighter topics than others, such as opioid use, alcohol and tobacco use, firearms, and hypertension. Therefore, the dominances of food and diet related discussion in the month of September may raise the overall positive sentiment, compared to other months.

Figure 16

#word importance by month  
months\_ordered <- c("January", "February", "March", "April", "May", "June",   
 "July", "August", "September", "October", "November", "December")  
tidy\_text$month <- factor(tidy\_text$month, levels = months\_ordered, ordered = TRUE)  
  
TDIDF\_month\_plot <- tidy\_text %>%   
 count(month,word) %>%   
 bind\_tf\_idf(term = word,  
 document = month,  
 n = n) %>%   
 group\_by(month) %>%   
 top\_n(n = 5,  
 wt = tf\_idf) %>%   
 ggplot() +   
 geom\_col(aes(y = reorder(word,tf\_idf),  
 x = tf\_idf,  
 fill=month))+  
 facet\_wrap(~month,  
 scales = "free") +  
 scale\_fill\_hue() +  
 ggtitle("Word Importance by Month") +   
 xlab("TF-IDF Score") +  
 ylab("Top Words") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))+  
 guides(fill = FALSE)   
TDIDF\_month\_plot

Figure 17

A group of colorful bars

Description automatically generated

Topic Modeling

**The third research question is:**

What topics comprise our data and how can we characterize them? What are the word-topic probabilities and what are the document-topic probabilities?

To address this question, we can conduct the unsupervised learning method of topic modeling. This method allows us to see what topics comprise our text. First, we attempted several different topic models, each with a different number of topics, to identify the most appropriate number of topics to model our data. Figure 18 displays the code to build and run topic models with 5, 8, 10, and 12 topics. Figure 18 also includes the resulting plots of each of the topic models that show the top 5 words in each topic.

The most appropriate topic model for the data is that with 10 topics. This model is visualized in figure 20.

**Below is a discussion on the characteristics of each topic from the 10 topic model.**

1: Patient awareness, training, and education

2: Disease control, prevention, and awareness.

3: Structured educational programs

4: Alcohol consumption guidelines and impacts, sports initiatives.  
5: Adult health guidelines and education.

6: Children’s health guidelines and education through adult efforts and collaboration.

7: Movement, exercise, and sports for youth.

8: Movement, exercise, and sports for all ages.

9: Appropriate prescription medication uses and care delivery.

10: Quality of health measures, national health concerns.

Figure 18

## SECTION 6: TOPIC MODELING

#if document is year:   
year\_word\_counts <- tidy\_text %>%  
 count(year,word,sort=T)  
  
dtm\_year <- year\_word\_counts %>%  
 cast\_dtm(document = year,  
 term = word,  
 value = n)  
dtm\_year

## <<DocumentTermMatrix (documents: 8, terms: 6802)>>  
## Non-/sparse entries: 16191/38225  
## Sparsity : 70%  
## Maximal term length: 57  
## Weighting : term frequency (tf)

#if document is month:   
month\_word\_counts <- tidy\_text %>%  
 count(month,word,sort=T)  
  
dtm\_month <- month\_word\_counts %>%  
 cast\_dtm(document = month,  
 term = word,  
 value = n)  
dtm\_month

## <<DocumentTermMatrix (documents: 12, terms: 6802)>>  
## Non-/sparse entries: 20709/60915  
## Sparsity : 75%  
## Maximal term length: 57  
## Weighting : term frequency (tf)

# Topic Modeling: 5 topics  
lda\_5 <- LDA(dtm\_year, k = 5,   
 control = list(seed = 1234))  
  
# Grab the topic-word probabilities  
word\_topic\_prob\_5 <- tidy(lda\_5,  
 matrix = "beta")  
word\_topic\_prob\_5

## # A tibble: 34,010 × 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 webinar 0.00229   
## 2 2 webinar 0.00112   
## 3 3 webinar 0.000113  
## 4 4 webinar 0.000183  
## 5 5 webinar 0.00765   
## 6 1 alcohol 0.000191  
## 7 2 alcohol 0.00436   
## 8 3 alcohol 0.000679  
## 9 4 alcohol 0.00118   
## 10 5 alcohol 0.000658  
## # ℹ 34,000 more rows

top\_terms\_5 <- word\_topic\_prob\_5 %>%  
 group\_by(topic) %>%  
 slice\_max(beta, n = 5) %>%   
 ungroup() %>%  
 arrange(topic, -beta)  
top\_terms\_5

## # A tibble: 25 × 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 social 0.00668  
## 2 1 objectives 0.00573  
## 3 1 preventive 0.00496  
## 4 1 national 0.00468  
## 5 1 guidelines 0.00458  
## 6 2 sports 0.00760  
## 7 2 youth 0.00633  
## 8 2 active 0.00584  
## 9 2 move 0.00542  
## 10 2 objectives 0.00519  
## # ℹ 15 more rows

# Make a plot  
top\_terms\_plot\_5 <- top\_terms\_5 %>%  
 mutate(term = reorder\_within(term, beta, topic)) %>%  
 ggplot(aes(x = beta, y = term, fill = factor(topic))) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~factor(topic), scales = "free" ) +  
 scale\_y\_reordered()+  
 ggtitle("Beta Probabilities with 5 Topics") +   
 xlab("Beta Probabilities") +  
 ylab("Top Words")  
top\_terms\_plot\_5

A group of colorful bars

Description automatically generated

# Topic Modeling: 8 topics  
lda\_8 <- LDA(dtm\_year, k = 8,   
 control = list(seed = 1234))  
  
# Grab the topic-word probabilities  
word\_topic\_prob\_8 <- tidy(lda\_8,  
 matrix = "beta")  
word\_topic\_prob\_8

## # A tibble: 54,416 × 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 webinar 0.00229   
## 2 2 webinar 0.00302   
## 3 3 webinar 0.000115   
## 4 4 webinar 0.000187   
## 5 5 webinar 0.0104   
## 6 6 webinar 0.00174   
## 7 7 webinar 0.000286   
## 8 8 webinar 0.00166   
## 9 1 alcohol 0.000191   
## 10 2 alcohol 0.00000318  
## # ℹ 54,406 more rows

top\_terms\_8 <- word\_topic\_prob\_8 %>%  
 group\_by(topic) %>%  
 slice\_max(beta, n = 5) %>%   
 ungroup() %>%  
 arrange(topic, -beta)  
top\_terms\_8

## # A tibble: 41 × 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 social 0.00668  
## 2 1 objectives 0.00573  
## 3 1 preventive 0.00497  
## 4 1 national 0.00468  
## 5 1 guidelines 0.00458  
## 6 2 library 0.00854  
## 7 2 adults 0.00845  
## 8 2 social 0.00717  
## 9 2 programs 0.00691  
## 10 2 children 0.00577  
## # ℹ 31 more rows

# Make a plot  
top\_terms\_plot\_8 <- top\_terms\_8 %>%  
 mutate(term = reorder\_within(term, beta, topic)) %>%  
 ggplot(aes(x = beta, y = term, fill = factor(topic))) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~factor(topic), scales = "free" ) +  
 scale\_y\_reordered()+  
 ggtitle("Beta Probabilities with 8 Topics") +   
 xlab("Beta Probabilities") +  
 ylab("Top Words")  
top\_terms\_plot\_8

A group of colorful bars

Description automatically generated

# Topic Modeling: 12 topics  
lda\_12 <- LDA(dtm\_year, k = 12,   
 control = list(seed = 1234))

# Grab the topic-word probabilities  
word\_topic\_prob\_12 <- tidy(lda\_12,  
 matrix = "beta")  
word\_topic\_prob\_12

## # A tibble: 81,624 × 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 webinar 0.00288   
## 2 2 webinar 0.00393   
## 3 3 webinar 0.000116   
## 4 4 webinar 0.000187   
## 5 5 webinar 0.0105   
## 6 6 webinar 0.00147   
## 7 7 webinar 0.000130   
## 8 8 webinar 0.000605   
## 9 9 webinar 0.00000621  
## 10 10 webinar 0.00395   
## # ℹ 81,614 more rows

top\_terms\_12 <- word\_topic\_prob\_12 %>%  
 group\_by(topic) %>%  
 slice\_max(beta, n = 5) %>%   
 ungroup() %>%  
 arrange(topic, -beta)  
top\_terms\_12

## # A tibble: 61 × 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 social 0.00805  
## 2 1 national 0.00644  
## 3 1 objectives 0.00639  
## 4 1 preventive 0.00625  
## 5 1 related 0.00525  
## 6 2 adults 0.00806  
## 7 2 library 0.00767  
## 8 2 social 0.00688  
## 9 2 programs 0.00629  
## 10 2 education 0.00531  
## # ℹ 51 more rows

# Make a plot  
top\_terms\_plot\_12 <- top\_terms\_12 %>%  
 mutate(term = reorder\_within(term, beta, topic)) %>%  
 ggplot(aes(x = beta, y = term, fill = factor(topic))) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~factor(topic), scales = "free" ) +  
 scale\_y\_reordered()+  
 ggtitle("Beta Probabilities with 12 Topics") +   
 xlab("Beta Probabilities") +  
 ylab("Top Words") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
top\_terms\_plot\_12

A group of colorful bars

Description automatically generated

Figure 19

# Topic Modeling: 10 topics  
lda\_10 <- LDA(dtm\_year, k = 10,   
 control = list(seed = 1234))  
  
# Grab the topic-word probabilities  
word\_topic\_prob\_10 <- tidy(lda\_10,  
 matrix = "beta")  
word\_topic\_prob\_10

## # A tibble: 68,020 × 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 webinar 2.29e- 3  
## 2 2 webinar 2.99e- 3  
## 3 3 webinar 5.84e- 5  
## 4 4 webinar 1.87e- 4  
## 5 5 webinar 1.04e- 2  
## 6 6 webinar 1.18e- 3  
## 7 7 webinar 1.92e- 4  
## 8 8 webinar 2.48e- 4  
## 9 9 webinar 3.49e-180  
## 10 10 webinar 1.66e- 3  
## # ℹ 68,010 more rows

top\_terms\_10 <- word\_topic\_prob\_10 %>%  
 group\_by(topic) %>%  
 slice\_max(beta, n = 5) %>%   
 ungroup() %>%  
 arrange(topic, -beta)  
top\_terms\_10

## # A tibble: 51 × 3  
## topic term beta  
## <int> <chr> <dbl>  
## 1 1 social 0.00668  
## 2 1 objectives 0.00573  
## 3 1 preventive 0.00497  
## 4 1 national 0.00468  
## 5 1 guidelines 0.00458  
## 6 2 library 0.00857  
## 7 2 adults 0.00846  
## 8 2 social 0.00718  
## 9 2 programs 0.00693  
## 10 2 children 0.00579  
## # ℹ 41 more rows

# Make a plot  
top\_terms\_plot\_10 <- top\_terms\_10 %>%  
 mutate(term = reorder\_within(term, beta, topic)) %>%  
 ggplot(aes(x = beta, y = term, fill = factor(topic))) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~factor(topic), scales = "free" ) +  
 scale\_y\_reordered()+  
 ggtitle("Beta Probabilities with 10 Topics") +   
 xlab("Beta Probabilities") +  
 ylab("Top Words") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
top\_terms\_plot\_10

Figure 20

A screenshot of a graph

Description automatically generated

The final step to answer research question 3 is to look at the document-topic probabilities, or gamma probabilities. This will provide insight into what topics each document is composed of. Again, the document used for the topic modeling is year, so there are 7 documents total. Figure 21 presents the gamma probabilities.

**Below we discuss the document-topic makeup for each year.**

2017: Almost entirely composed of topic 9, which is characterized as appropriate prescription medication use and care delivery.

2018: Composed of topics 3 and 8, with a greater percentage of topic 3. Topic 3 represents education programs and topic 8 represents movement, exercise, and sports for all ages.

2019: Almost entirely composed of topic 4, which is characterized as alcohol consumption guidelines and impacts, sports initiatives

2020: Split between topics 5 and 6. Topic 5 is characterized as adult health guidelines and education and 6 as children’s health guidelines and education through adult efforts and collaboration. It seems 2020, as a whole, can be characterized and general health and lifestyle guidelines and education.

2021: Mostly composed of topic 10, which is characterized as quality of health measures, national health concerns

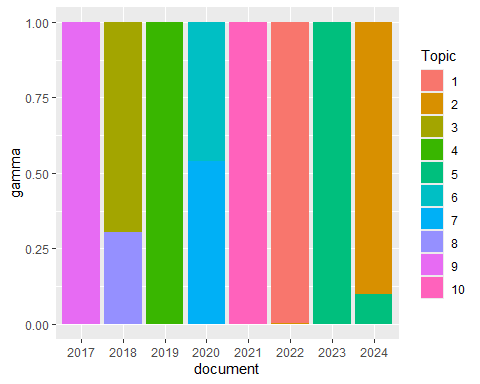
2022: Mostly composed of topic 1, which is characterized as patient awareness, training, and education

2023: Almost entirely composed of topic 5, which is characterized as adult health guidelines and education.

2024: Mostly topic 2, and some of topic 5. Topic 2 represents disease control, prevention, and awareness and topic 5 adult health guidelines and education.

Figure 21

# gamma probability for 10 topics  
  
tidy(lda\_10, "gamma") %>%   
 ggplot() +  
 geom\_col(aes(document,gamma, fill=factor(topic))) +  
 scale\_fill\_discrete(name="Topic")



Conclusion:

This analysis shed light on three research questions related to public health news. First, sentiment analysis showed that a pattern can be seen among high sentiment in the fall months, particularly in the fall of 2018 and 2020. A potential reason for this is because the beginning of the school year sparks conversations around healthy eating, healthy habits, and movement for youth. These conversations tend to be inspiring and more uplifting that many of the other topics discussed in public health. On the other hand, April of 2022 saw a drastic increase in negative sentiment due to attention around elder abuse issues. Lastly, the sentiment analysis revealed that an advantage of using the AFINN lexicon is that it provides more insight into where sentiment trends are negative. Examining word importances through TF-IDF provided insight on how the most heavily discussed topics change throughout the year, from dietary and exercise guidelines for children in the fall, to tobacco use in the spring, and alcohol consumption in the summer. Additionally, the TF-IDF scores indicated topics that were very important in certain years and less important in others, such as opioid use in 2018 and firearm safety in 2022. Finally, topic modeling revealed that there are 12 distinct topics that comprise the health.gov news and most of the documented years are dominated by one or two of these topics.

Citations

News archive. (n.d.). https://health.gov/news/archive?page=1