EDS 231 Final Project: Climate Gentrification Text & Sentiment Analysis

Alex Clippinger, Halina Do-Linh, Desik Somasundaram, Alex Vand

2022-05-25

Background and Research question

Language allows us to articulate our thoughts and emotions. In our Ethics and Bias class, we listened to Valencia Gunder, a prominent climate activist, explain how the coining of the term "climate gentrification" finally gave rise to awareness about a problem and emotion that has been felt by low-income communities in Miami for several years now. The Keenan et al. 2018 paper talks about Miami-Dade County, Florida as a case study for market mechanisms enabling this phenomenon.

Our initial research question was to understand the emergence of the term "climate gentrification" both spatially and temporally. The idea was this analysis would give insights into how language spreads and evolves, highlighting the emotions of those who have been historically left out of the conversation around climate action. Valencia Gunder mentioned how city officials and stakeholders often neglected the concerns of the people in the city of Miami before the term "climate gentrification" was coined and supported by academic literature.

Due to data acquisition limitations, we had to pivot our research question. We are now comparing how the two data sources, Nexis Uni and Twitter, discuss the term "climate gentrification". We were also not able to move forward with our spatial and temporal analysis. Instead, we ran sentiment, word relationship, and topic modeling analysis on both datasets and compared them. We hope that research on this topic will encourage more engagement between researchers and people that are living through the problems researchers are attempting to help solve.

Data collection plan

Data will be accessed via Twitter (tweets) and Nexis Uni (news publications). Tweets referring to the keywords "climate gentrification" along with the hashtag "#climategentrification" will be queried for the time period of 2019-present. The query will be conducted on Brandwatch's Consumer Research, which will be accessed through the UCSB Collaboratory license. The project team has completed a consultation with UCSB Collaboratory to ensure necessary access. Our query resulted in approximately 10,276 tweets. The entire Nexis Uni database was queried for "climate gentrification", which resulted in 498 unique news articles, law reviews and journals, legal news, legal briefs, statutes and legislation, and directories from 2014-present. The Nexis Uni database access is available through the UCSB library as well.

Analysis plan

Our analysis focused on the sentiment, word relationships, and topics of discussion surrounding climate gentrification. This first required cleaning the Twitter and Nexis data by removing domain-specific stop words, stemming key terms, and removing unnecessary terms, phrases, URLs, etc. Next, the team conducted sentiment analysis on the data to identify key emotions surrounding the topic. Then, we used word relationship analysis to dive deeper into the words that were common in the key emotions. Lastly, we used topic modeling to elucidate the primary themes of the discussion.

Setup Data

Setup stop words and Bing/NRC sentiments

```
#read in stop words data
data(stop_words)
```

```
## Read in Nexis Uni Data

# read in nexis uni data
my_files <- list.files(pattern = ".docx", path = here("data"),
                      full.names = TRUE, recursive = TRUE, ignore.case = TRUE)

cg_nex_data <- lnt_read(my_files) # object of class 'LNT output'
```

```
## Warning in lnt_asDate(date.v, ...): More than one language was detected. The
## most likely one was chosen (English 87.05%)
```

```
cg_nex_meta_df <- cg_nex_data@meta
cg_nex_articles_df <- cg_nex_data@articles
cg_nex_paragraphs_df <- cg_nex_data@paragraphs
```

Setup stop words and Bing/NRC sentiments

```
#read in stop words data
data(stop_words)
```

Cleaning Nexis Uni Data

```
cg_nex_dat2<- data_frame(element_id = seq(1:length(cg_nex_meta_df$Headline)),
                     Date = cg_nex_meta_df$Date,
                     Headline = cg_nex_meta_df$Headline)


cg_nex_paragraphs_dat <- data_frame(element_id = cg_nex_paragraphs_df$Art_ID,
                               Text  = cg_nex_paragraphs_df$Paragraph)

cg_nex_dat3 <- inner_join(cg_nex_dat2, cg_nex_paragraphs_dat, by = "element_id") %>%
  janitor::clean_names()

cg_nex_dat3 <- subset(cg_nex_dat3, text != " " )
```

```r
cg_nex_dat3 <- cg_nex_dat3[!grepl("POSTED", cg_nex_dat3$text,ignore.case = TRUE),]
cg_nex_dat3 <- cg_nex_dat3[!grepl("GRAPHIC", cg_nex_dat3$text,ignore.case = TRUE),]
cg_nex_dat3 <- cg_nex_dat3[!grepl(":", cg_nex_dat3$text),]
cg_nex_dat3 <- cg_nex_dat3[!grepl("LINK TO", cg_nex_dat3$text,ignore.case = TRUE),]
cg_nex_dat3 <- cg_nex_dat3[grepl("[a-zA-Z]", cg_nex_dat3$text),]

# clean the corpus
cg_nex_corp <- corpus(x = cg_nex_articles_df, text_field = "Article")
cg_nex_corp.stats <- summary(cg_nex_corp)
head(cg_nex_corp.stats, n = 25)
```

```
##      Text Types Tokens Sentences ID
## 1   text1   235    413        14  1
## 2   text2   429    957        24  2
## 3   text3   429    957        24  3
## 4   text4   430    963        24  4
## 5   text5   430    963        24  5
## 6   text6   430    963        24  6
## 7   text7   551   1344        43  7
## 8   text8   562   1110        50  8
## 9   text9   562   1110        50  9
## 10 text10   562   1110        50 10
## 11 text11   319    608        23 11
## 12 text12   871   2187        94 12
## 13 text13   296    608        27 13
## 14 text14   622   1541        59 14
## 15 text15   587   1654        67 15
## 16 text16  1009   2308        65 16
## 17 text17  1022   2345        65 17
## 18 text18   294    544        27 18
## 19 text19   296    546        27 19
## 20 text20   361    776        31 20
## 21 text21   294    544        27 21
## 22 text22   672   1680        67 22
## 23 text23   736   1871        82 23
## 24 text24  1146   2684        87 24
## 25 text25  1364   3831       152 25
```

```r
toks <- tokens(cg_nex_corp, remove_punct = TRUE, remove_numbers = TRUE)
# added some project-specific stop words here
more_stops <- c(stopwords("en"), "like", "just", "say", "year")
add_stops<- tibble(word = c(stop_words$word, more_stops))
stop_vec <- as_vector(add_stops)
toks1 <- tokens_select(toks, pattern = stop_vec, selection = "remove")

# unnest to word-level tokens, remove stop words, and join sentiment words
cg_nex_text_words <- cg_nex_dat3  %>%
  unnest_tokens(output = word, input = text, token = 'words') %>%
  drop_na()
```

Convert Nexis Uni to document-feature matrix

```
dfm_comm<- dfm(toks1, tolower = TRUE)
dfm <- dfm_wordstem(dfm_comm)
dfm <- dfm_trim(dfm, min_docfreq = 2) #remove terms only appearing in one doc (min_termfreq = 10)

print(head(dfm))
```

```
## Document-feature matrix of: 6 documents, 12,866 features (98.17% sparse) and 1 docvar.
##        features
## docs    york kansa citi miami denver mantra locat relev consider real
##   text1    2     1    8     2      2      1     3     1        1    5
##   text2    0     0    9     9      0      0     0     0        1    2
##   text3    0     0    9     9      0      0     0     0        1    2
##   text4    0     0    9     9      0      0     0     0        1    2
##   text5    0     0    9     9      0      0     0     0        1    2
##   text6    0     0    9     9      0      0     0     0        1    2
## [ reached max_nfeat ... 12,856 more features ]
```

```
#remove rows (docs) with all zeros
sel_idx <- slam::row_sums(dfm) > 0
dfm <- dfm[sel_idx, ]
```

Initial exploration of Nexis Uni data

```
cg_nex_words_by_date <- cg_nex_text_words %>%
    anti_join(stop_words) %>%
    group_by(date) %>%
    count(date, word)
```

Compare top ten most common words per day

```
## Joining, by = "word"
```

```
cg_nex_top_words_by_date <- cg_nex_words_by_date %>% group_by(date) %>% top_n(n = 10, wt = n)
cg_nex_top_words_by_date[order(cg_nex_top_words_by_date$n, decreasing = TRUE),]
```

```
## # A tibble: 3,460 x 3
## # Groups:   date [228]
##    date       word        n
##    <date>     <chr>   <int>
##  1 2019-04-02 housing   369
##  2 2019-04-02 fair      224
##  3 2021-07-20 climate   193
##  4 2021-11-28 housing   175
##  5 2021-06-30 climate   161
##  6 2021-11-28 flood     134
##  7 2016-10-31 housing   122
```

```
##  8 2020-01-01 id        121
##  9 2021-02-26 housing   112
## 10 2020-06-29 flood     110
## # ... with 3,450 more rows
```

```
cg_nex_text_words %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("red", "green"),
                   max.words = 100)
```

Positive-Negative Wordcloud of Nexis Uni

b) Twitter Data Preparation

```
raw_tweets <- readxl::read_excel(here("data","twitter_data_agg.xlsx"),sheet = 1, col_names = TRUE, col_

dat <- raw_tweets[,c(4,6)] # Extract Date and Title fields

tweets <- tibble(text = dat$Title,
                 id = seq(1:length(dat$Title)),
                 date = as.Date(as.numeric(dat$Date), origin = "1899-12-30"))
```

Cleaning Twitter Data

```
cg_t_corpus <- corpus(dat$Title) # enter quanteda
#summary(corpus)

cg_t_tokens <- tokens(cg_t_corpus) # tokenize the text so each doc (page, in this case) is a list of to

# clean it up
cg_t_tokens <- tokens(cg_t_tokens, remove_punct = TRUE,
                 remove_numbers = TRUE)

cg_t_tokens <- tokens_select(cg_t_tokens, stopwords('english'), selection='remove') # stopwords lexicon

# tokens <- tokens_wordstem(tokens) #stem words down to their base form for comparisons across tense an

cg_t_tokens <- tokens_tolower(cg_t_tokens)


theString <- unlist(strsplit(tweets$text, " "))
regex <- "(^|[^@\\w])@(\\w{1,15})\\b"
tweets$text <- gsub(regex, "", tweets$text)
# let's clean up the URLs from the tweets
tweets$text <- gsub("http[^[:space:]]*", "",tweets$text)
tweets$text <- str_to_lower(tweets$text)
tokenized_tweets <- tweets %>%
  unnest_tokens(word, text)

# tokenize tweets to individual words
words <- tweets %>%
  select(id, date, text) %>%
  unnest_tokens(output = word,
                input = text,
                token = "words") # %>%
```

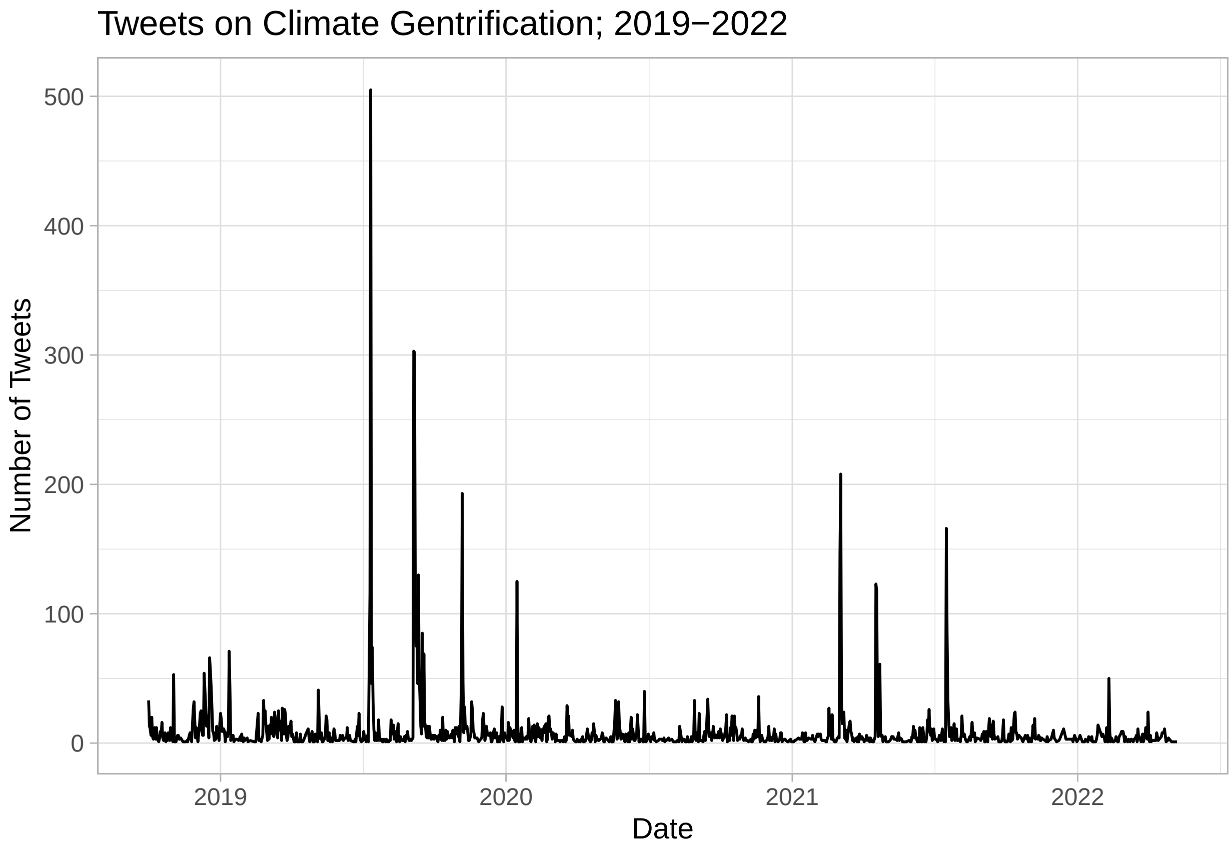
Initial exploration of twitter data

```


# Simple plot of tweets per day
daily_tweets <- tweets %>%
  count(date)

daily_tweets_plot <- ggplot(daily_tweets, aes(x = date, y = n)) +
  geom_line() +
  theme_light() +
  labs(y = "Number of Tweets",
       x = "Date",
       title = "Tweets on Climate Gentrification; 2019-2022")

#ggsave("plots/daily_tweets.png", daily_tweets_plot)
daily_tweets_plot
```



Time-Series

The date with the highest number of tweets about climate gentrification is March 4, 2021. On March 3rd, CNN released a story titled High ground, high prices, which reported on climate gentrification. Specific problems discussed in the article include Miami's Little Haiti neighborhood and New Orleans, particularly after displacement caused by Hurricane Katrina.

Another notable date is July 17th, 2021, which corresponds to the Washington Post's article covering climate gentrification following the tragic Surfside condo collapse on June 24th, 2021.

Keywords-in-context We can use the kwic function (keywords-in-context) to briefly examine the context in which certain words or patterns appear.

```
head(kwic(cg_t_tokens, pattern = phrase("climate gentrification"), window = 5))
```

```
## Keyword-in-context with 6 matches.
##  [text1, 26:27] inequalities calling closer attention green |
##  [text2, 12:13]      course help provide historical context |
##  [text3, 10:11]         came minutes writing workshop event |
##    [text6, 4:5]                  rt@spacecrone presentation |
##    [text7, 4:5]                  rt@spacecrone presentation |
##    [text8, 2:3]                                presentation |
##
##  climate gentrification |
##  climate gentrification |
##  climate gentrification |
##  climate gentrification |
##  climate gentrification |
```

```
##  climate gentrification |
##
##  https://t.co/vhzquxv9pc
##  norfolk virginia areas nhttps://t.co/rdmelyrkyc
##  #miami#littlehaiti#sihowsundays#sihowthedoctor#gentrification
##  tonight efforts put money climate
##  tonight efforts put money climate
##  tonight efforts put money climate
```

```
hash_tweets <- tokens(cg_t_corpus, remove_punct = TRUE) %>%
  tokens_keep(pattern = "#*")

dfm_hash <- dfm(hash_tweets)

tstat_freq <- textstat_frequency(dfm_hash, n = 100)
head(tstat_freq, 10)
```

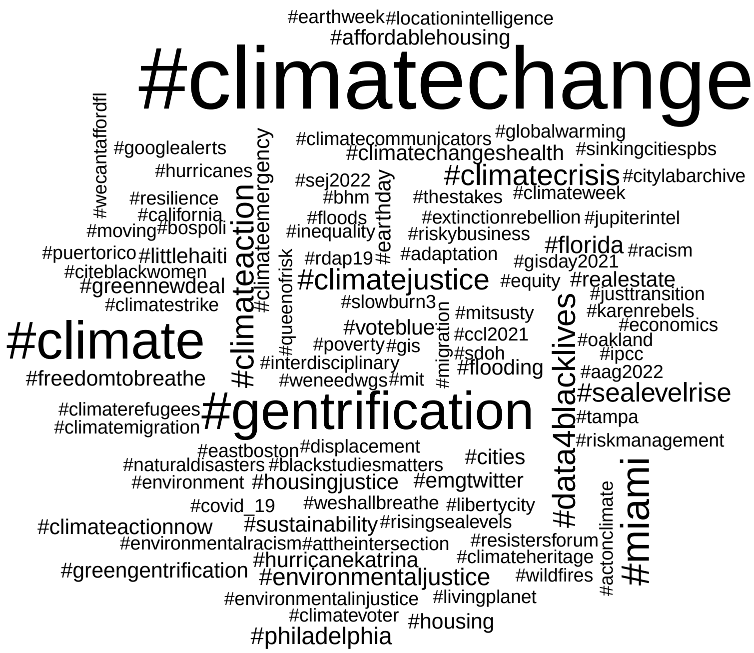
Wordcloud of hashtags

```
##                   feature frequency rank docfreq group
## 1  #climategentrification       733    1     733   all
## 2          #climatechange       469    2     469   all
## 3                #climate       252    3     252   all
## 4         #gentrification       251    4     251   all
## 5                  #miami       152    5     151   all
## 6          #climateaction       102    6     102   all
## 7        #data4blacklives        96    7      96   all
## 8         #climatejustice        84    8      84   all
## 9          #climatecrisis        81    9      81   all
## 10          #sealevelrise        65   10      65   all
```

```
# tidytext gives us tools to convert to tidy from non-tidy formats
hash_tib <- tidy(dfm_hash)

hash_tib %>%
  count(term) %>%
  with(wordcloud(term, n, max.words = 100))
```

```
## Warning in wordcloud(term, n, max.words = 100): #climategentrification could not
## be fit on page. It will not be plotted.
```




```r
cg_t_dfm <- dfm(cg_t_tokens)

#topfeatures(dfm, 12)

cg_t_dfm.sentiment <- dfm_lookup(cg_t_dfm, dictionary = data_dictionary_LSD2015)

#head(textstat_polarity(tokens, data_dictionary_LSD2015, fun = sent_logit))
```

Convert to document feature matrix using `quanteda` `textstat_polarity()`

```r
words_by_date <- tokenized_tweets %>%
    anti_join(stop_words) %>%
    group_by(date) %>%
    count(date, word)
```

Compare top ten most common tweets per day

```
## Joining, by = "word"
```

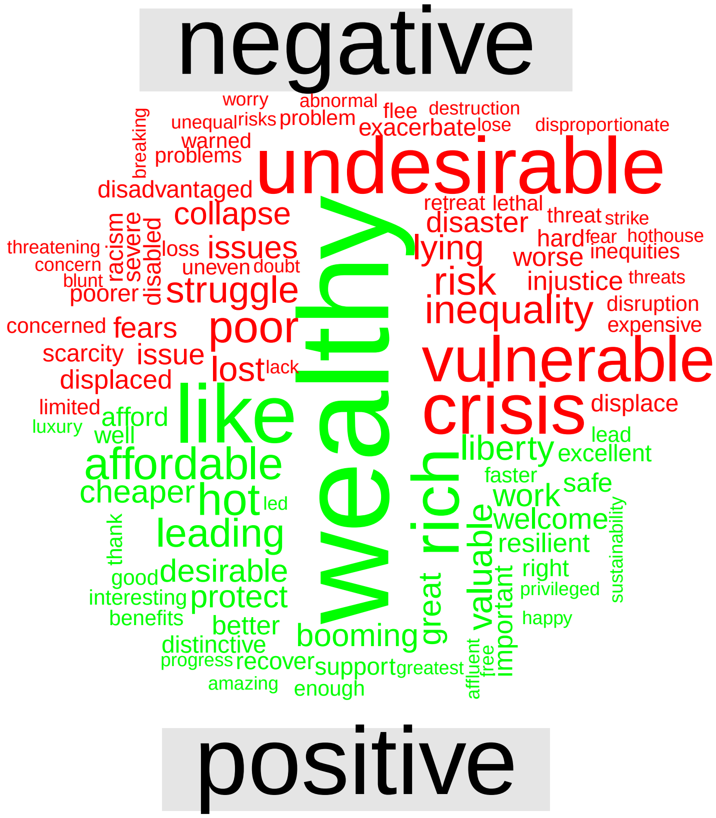
```r
top_words_by_date <- words_by_date %>% group_by(date) %>% top_n(n = 10, wt = n)
top_words_by_date[order(top_words_by_date$n, decreasing = TRUE),]
```

```
## # A tibble: 20,742 x 3
## # Groups:   date [1,100]
##    date       word          n
##    <date>     <chr>     <int>
##  1 2019-07-12 elevation   826
##  2 2019-07-12 location    765
##  3 2019-07-12 day         437
##  4 2019-07-12 rt          424
##  5 2019-07-12 miami       419
##  6 2019-07-12 seas        312
##  7 2019-09-06 climate     293
##  8 2019-07-12 rising      291
##  9 2019-09-05 climate     287
## 10 2019-07-12 estate      260
## # ... with 20,732 more rows
```

```r
words %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("red", "green"),
                   max.words = 100)
```

Positive-Negative Wordcloud of Tweets

```
## Joining, by = "word"
```




```r
at_tweets <- tokens(cg_t_corpus, remove_punct = TRUE) %>%
               tokens_keep(pattern = "@*")

dfm_at<- dfm(at_tweets)

tstat_freq <- textstat_frequency(dfm_at, n = 10)

tstat_freq
```

Most tagged accounts on Twitter

```
##             feature frequency rank docfreq group
## 1      @motherjones       866    1     866   all
## 2              @cnn       542    2     542   all
## 3             @nrdc       186    3     157   all
## 4      @nadegegreen       181    4     179   all
## 5       @kai_wright       181    4     164   all
## 6     @ianguelovski       162    6     162   all
## 7             @cnbc       156    7     156   all
## 8             @cnni       147    8     147   all
## 9  @action__johnson       130    9     118   all
## 10            @wlrn       129   10     129   all
```

Analysis

Sentiment Analysis

Get Bing and NRC sentiments

```r
bing_sent <- get_sentiments('bing') # grab the bing sentiment lexicon from tidytext
# head(bing_sent, n = 20)
nrc_sent <- get_sentiments('nrc') %>%
            filter(!sentiment %in% c("positive","negative")) # requires downloading a large dataset via
```

Nexis Uni Sentiment

```r
cg_nex_sent_words <- cg_nex_text_words %>% # break text into individual words
  anti_join(stop_words, by = 'word') %>% # returns only the rows without stop words
  inner_join(bing_sent, by = 'word') # joins and retains only sentiment words
```

Add Bing sentiments

```r
cg_nex_word_counts <- cg_nex_text_words %>%
  inner_join(nrc_sent) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

Add NRC sentiment word count

Results The following figure displays trends in Nexis Uni data sentiment over time

```r
cg_nex_sent_counts <- cg_nex_text_words %>%
        inner_join(nrc_sent) %>%
        group_by(date) %>%
        count(sentiment, sort = TRUE) %>%
        mutate(sentwords_per_day = sum(n)) %>%
        mutate(pct_contribution = ((n/sentwords_per_day)*100)) %>%
        filter(date >= "2018-01-01")

cg_nex_sent_timeplot<-cg_nex_sent_counts %>%
  group_by(date) %>%
  ggplot(aes(date, pct_contribution, group=sentiment, color=sentiment))  +
  geom_smooth(span = 0.7)  +
  labs(x = "Date",
      y = "Contribution to sentiment(%)",
      title = "NEXIS UNI")+
  theme(legend.position = "none" )
```

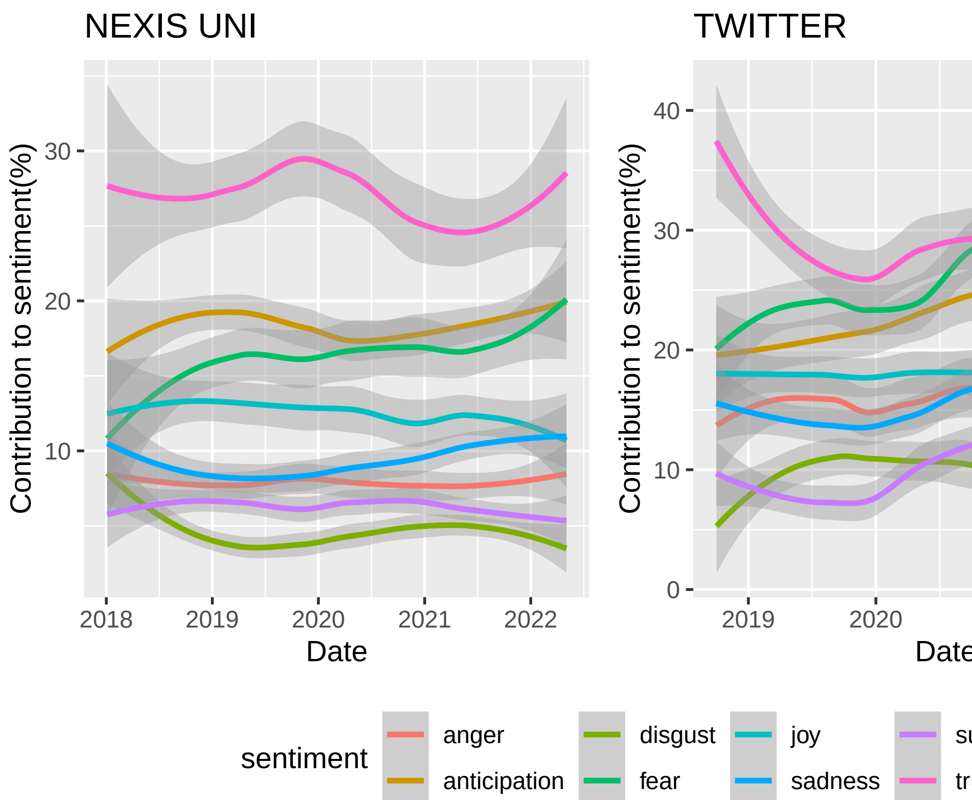
Twitter Sentiment

```
cg_t_word_counts <- words %>%
  inner_join(nrc_sent) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
cg_t_sent_counts <- words %>%
        inner_join(nrc_sent) %>%
        group_by(date) %>%
        count(sentiment, sort = TRUE) %>%
        mutate(sentwords_per_day = sum(n)) %>%
        mutate(pct_contribution = ((n/sentwords_per_day)*100))

cg_t_sent_timeplot<-cg_t_sent_counts %>%
  group_by(date) %>%
  ggplot(aes(date, pct_contribution, group=sentiment, color=sentiment)) +
  geom_smooth(span = 0.7) +
  labs(x = "Date",
       y = "Contribution to sentiment(%)",
       title = "TWITTER") +
  theme(legend.position = "bottom")
```

```
(cg_nex_sent_timeplot+cg_t_sent_timeplot) + plot_layout(guides = "collect") & theme(legend.position = 'b
```



Add NRC sentiment word count

Figure X1 shows the percent contribution to overall sentiment from the Nexis Uni data subset to 2018-2022 to better align with the Twitter data and the percent contribution to overall sentiment from the Twitter data. Both figures indicate that trust, anticipation and fear are the top 3 emotions in both the published sentiment from Nexis Uni data and people's sentiment from the Twitter data. Another interesting observation was the percentage contribution of anger and sadness is higher in Twitter data when compared to the Nexis Uni data. This aligns with our expectations as the emotions are similar yet more muted in the Nexis Uni data.

```
#tokenize tweets to individual words
words_forsent <- tweets %>%
  select(id, date, text) %>%
  unnest_tokens(output = word, input = text, token = "words") %>%
  anti_join(stop_words, by = "word") %>%
  left_join(bing_sent, by = "word") %>%
  left_join(
    tribble(
      ~sentiment, ~sent_score,
      "positive", 1,
      "negative", -1),
    by = "sentiment")

#take average sentiment score by tweet
tweets_sent <- tweets %>%
  left_join(
    words_forsent %>%
      group_by(id) %>%
      summarize(
```

```
        sent_score = mean(sent_score, na.rm = T)),
    by = "id")

neutral <- length(which(tweets_sent$sent_score == 0))
positive <- length(which(tweets_sent$sent_score > 0))
negative <- length(which(tweets_sent$sent_score < 0))

Sentiment <- c("Positive","Neutral","Negative")
Count <- c(positive,neutral,negative)
output <- data.frame(Sentiment,Count)
output$Sentiment<-factor(output$Sentiment,levels=Sentiment)
cg_sentplot_t <- ggplot(output, aes(x=Sentiment,y=Count))+
  geom_bar(stat = "identity", aes(fill = Sentiment))+
  scale_fill_manual("legend", values = c("Positive" = "#5ab4ac", "Neutral" = "lightgray", "Negative" =
  ggtitle("TWITTER")
```

```
#tokenize tweets to individual words
words_forsent_nex <-cg_nex_dat3  %>%
  unnest_tokens(output = word, input = text, token = 'words') %>%
  anti_join(stop_words, by = "word") %>%
  left_join(bing_sent, by = "word") %>%
  left_join(
    tribble(
      ~sentiment, ~sent_score,
      "positive", 1,
      "negative", -1),
    by = "sentiment")

#take average sentiment score by tweet
nex_sent <- cg_nex_dat3 %>%
  left_join(
    words_forsent_nex %>%
      group_by(element_id) %>%
      summarize(
        sent_score = mean(sent_score, na.rm = T)),
    by = "element_id") %>%
  group_by(element_id) %>%
  summarize(
        mean_sent_score = mean(sent_score, na.rm = T))

neutral <- length(which(nex_sent$mean_sent_score == 0))
positive <- length(which(nex_sent$mean_sent_score > 0))
negative <- length(which(nex_sent$mean_sent_score < 0))

Sentiment <- c("Positive","Neutral","Negative")
Count <- c(positive,neutral,negative)
nexoutput <- data.frame(Sentiment,Count)
nexoutput$Sentiment<-factor(nexoutput$Sentiment,levels=Sentiment)

cg_sentplot_nex <-ggplot(nexoutput, aes(x=Sentiment,y=Count))+
  geom_bar(stat = "identity", aes(fill = Sentiment))+
  scale_fill_manual("legend", values = c("Positive" = "#5ab4ac", "Neutral" = "lightgray", "Negative" =
  ggtitle("NEXIS UNI")
```

```r
(cg_sentplot_nex+cg_sentplot_t) + plot_layout(guides = "collect") & theme(legend.position = 'bottom')
```

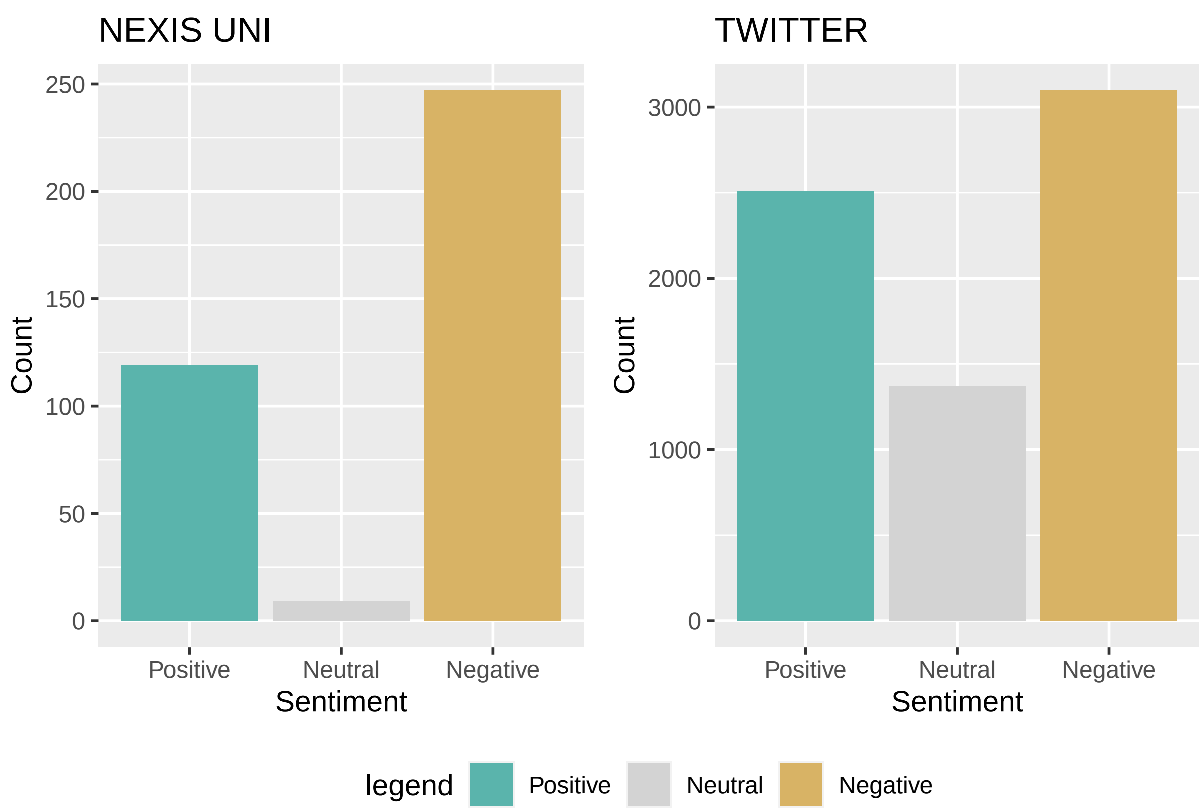


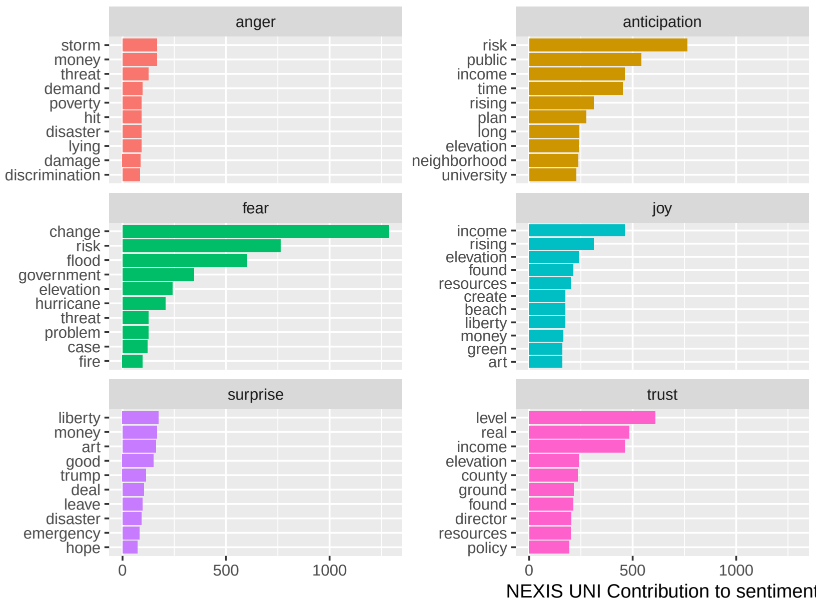
Fig XX shows the overall sentiment score classification by publication for Nexis Uni and by tweet for the Twitter data. With the Nexis Uni publications being longer in length, we suspect there are less neutral classification given that neutral is discrete score of 0. Otherwise, the positive and negative sentiment distributions are similar when comparing the two data sources.

```r
cg_nex_word_nplot <- cg_nex_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "NEXIS UNI Contribution to sentiment",
       y = NULL)
```

```r
cg_t_word_nplot <-cg_t_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
```

```r
  labs(x = "TWITTER Contribution to sentiment",
       y = NULL)
```

```r
cg_nex_word_nplot/cg_t_word_nplot
```



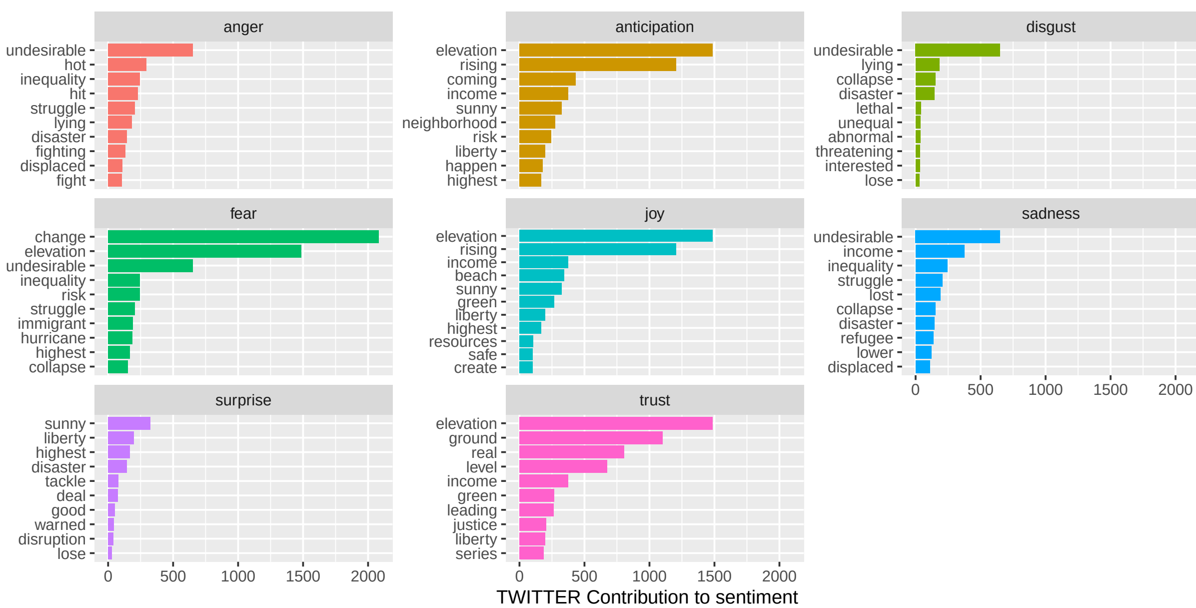


Fig XX show the top 10 words for emotion by data source. The only top word that is common from both datasets is for the fear emotion, where the top word is "change". In joy and sadness, we see the word "income" is in the top three. Otherwise, we see very little similarities between the two datasets. This leads to think that the two sources may not be talking about the same topics within these emotions or they might be using different words to talk about the same topics. We will explore this further using topic modeling analysis.

We also noticed that words such as "undesirable" and "income" are common in many of the emotions from the Twitter data. "Income" also comes up in multiple emotions in the Nexis Uni data. To further analyze this, we are going to use word relationship analysis to gain more context regarding the use of “undesirable” and “income”.

Word relationships / Correlations of words in Nexis Uni

```
# create objects to do finds correlations
# convert to tidy format and apply my stop words
raw_text <- tidy(cg_nex_corp)

# distribution of most frequent words across documents
raw_words <- raw_text %>%
  unnest_tokens(word, text) %>%
  anti_join(add_stops, by = 'word') %>%
  count(word, sort = TRUE)

report_words <- raw_words

par_tokens <- unnest_tokens(raw_text, output = paragraphs, input = text, token = "paragraphs")

par_tokens <- par_tokens %>%
 mutate(par_id = 1:n())

par_words <- unnest_tokens(par_tokens, output = word, input = paragraphs, token = "words")
```

```
# find words that occur close together in the nexis uni docs
word_pairs <- par_words %>%
  pairwise_count(word, par_id, sort = TRUE, upper = FALSE) %>%
  anti_join(add_stops, by = c("item1" = "word")) %>%
  anti_join(add_stops, by = c("item2" = "word"))
```

```
# plot correlations
word_pairs_nex_plot <- word_pairs %>%
  filter(n >= 200) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = n, edge_width = n), edge_colour = "dodgerblue") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()

ggsave("word_pairs_nex_plot.png",
       plot = word_pairs_nex_plot,
       path = "plots")
```

```
# plot correlations by paragraph
word_cors <- par_words %>%
  anti_join(add_stops, by = c("word" = "word")) %>%
  add_count(par_id) %>%
  filter(n >= 200) %>%
  select(-n) %>%
  pairwise_cor(word, par_id, sort = TRUE)
```

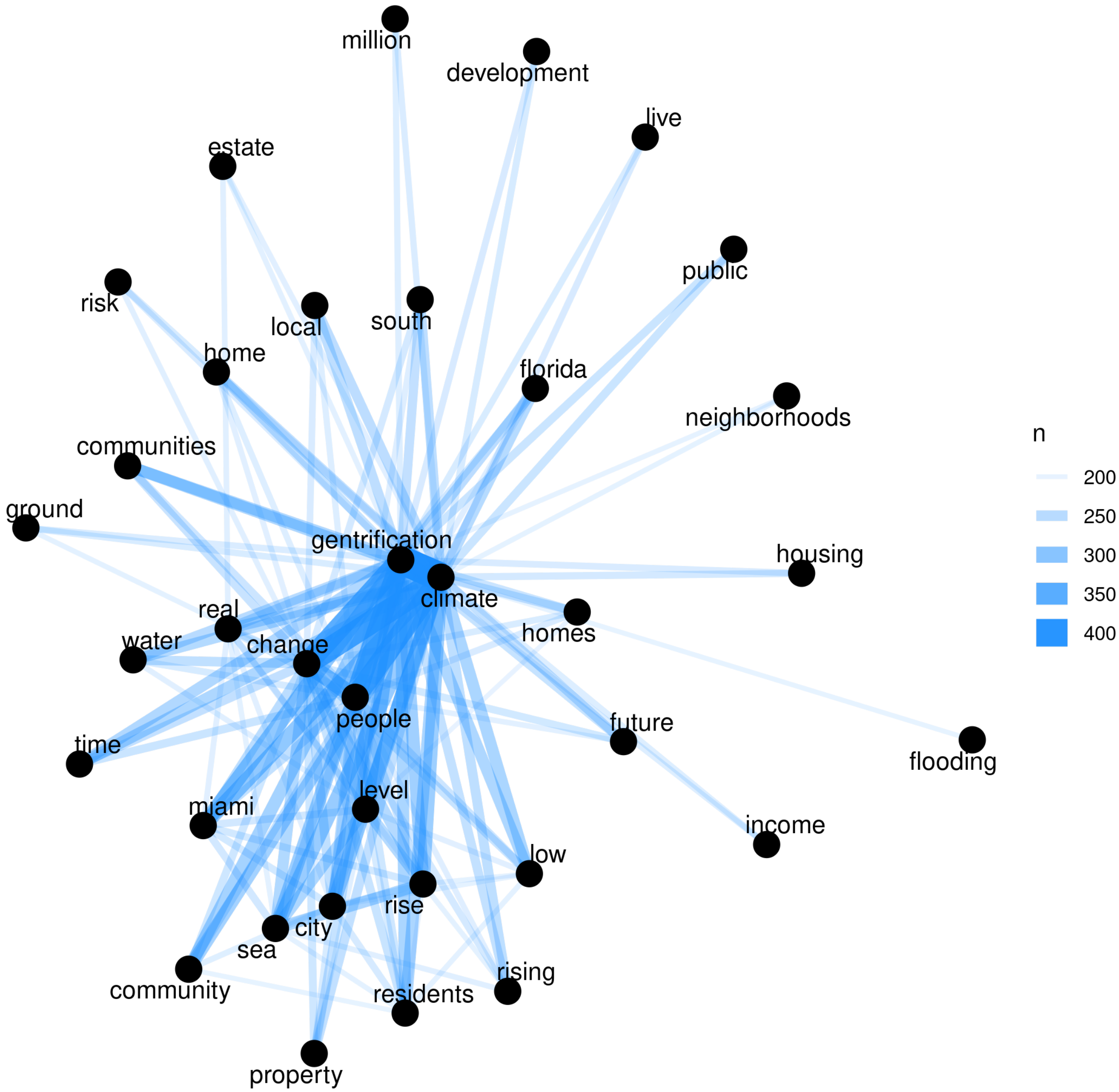


Figure 1: Word Pairs Plot

```r
key_word_corr_nex <- word_cors %>%
  filter(item1 %in% c("income")) %>%
  group_by(item1) %>%
  top_n(6) %>%
  ungroup() %>%
  mutate(item1 = as.factor(item1),
         name = reorder_within(item2, correlation, item1)) %>%
  ggplot(aes(y = name, x = correlation, fill = item1)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ item1, ncol = 2, scales = "free") +
  scale_y_reordered() +
  labs(
    y = NULL,
    x = NULL,
    title = "Correlations with key words",
    subtitle = "Climate gentrification NEXIS UNI"
  )
```

```r
# let's zoom in on just one of our key terms
undesirable_cors <- word_cors %>%
  filter(item1 == "undesirable") %>%
  mutate(n = 1:n())

ggsave("key_word_corr_nex_plot.png",
       plot = key_word_corr_nex,
       path = "plots")
```

```r
# let's zoom in on income key term
income_cors <- word_cors %>%
  filter(item1 == "income") %>%
  mutate(n = 1:n())

# correlation network
income_corr_nex_plot <- income_cors  %>%
  filter(n <= 50) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation),
                 edge_colour = "cyan4") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name),
                 repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()
```

```r
income_cg <- c("income", "climate gentrification")
income_toks_inside <- tokens_keep(toks1, pattern = income_cg, window = 20)
income_toks_inside <- tokens_remove(income_toks_inside, pattern = income_cg) # remove the keywords
income_toks_outside <- tokens_remove(toks1, pattern = income_cg, window = 20)
```

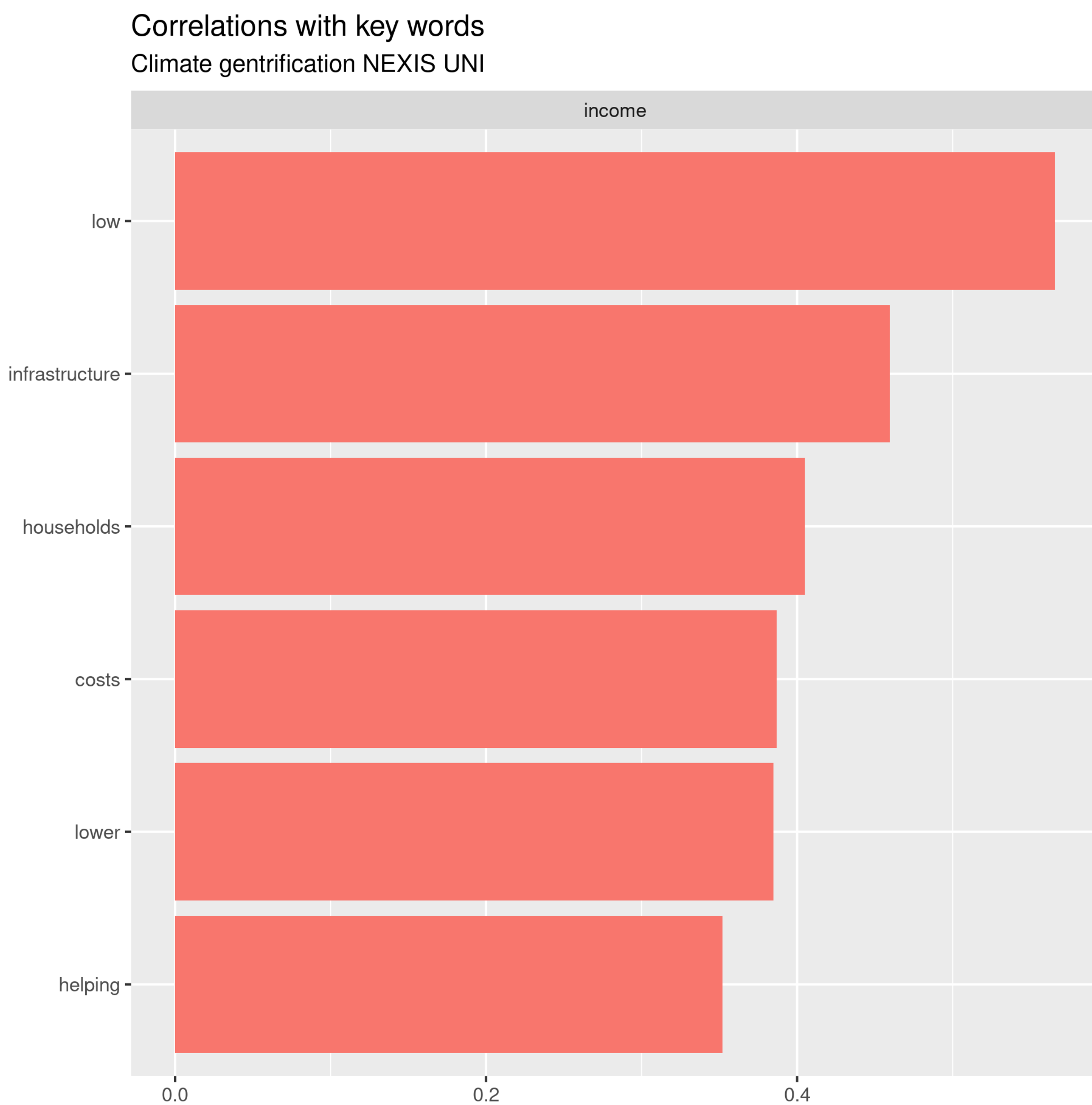


Figure 2: Correlation with Key Words

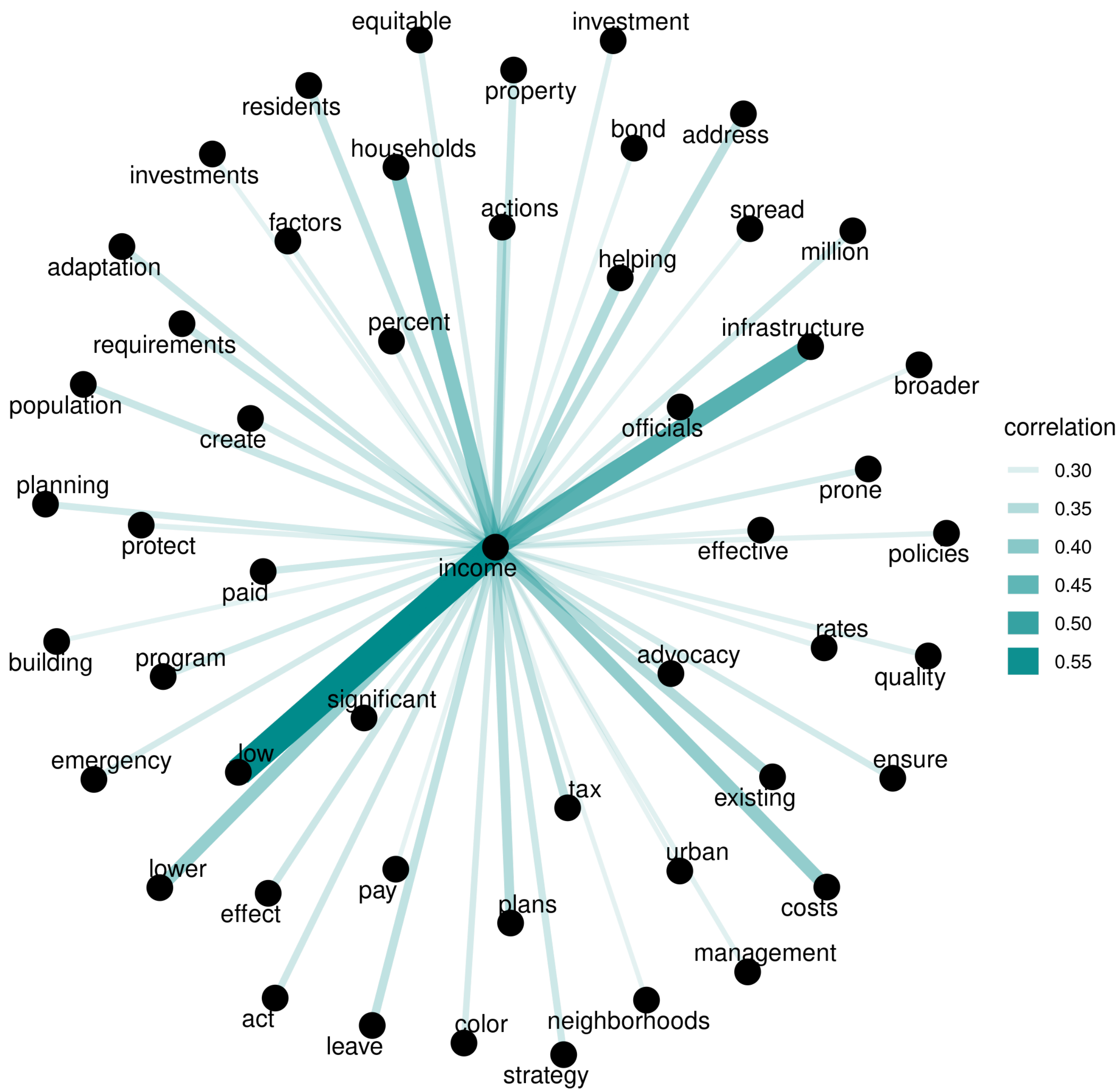


Figure 3: Income Correlation Plot

```r
income_dfmat_inside <- dfm(income_toks_inside)
income_dfmat_outside <- dfm(income_toks_outside)

income_tstat_key_inside <- textstat_keyness(rbind(income_dfmat_inside, income_dfmat_outside),
                                    target = seq_len(ndoc(income_dfmat_inside)))
head(income_tstat_key_inside, 10)
```

“income” and climate gentrification as multi-word term of interest in Nexis Uni

```
##       feature     chi2 p n_target n_reference
## 1         low 606.0426 0       57         219
## 2     housing 572.2374 0      192        2292
## 3  subsidized 415.5128 0       24          50
## 4    airlines 406.0781 0       18          25
## 5      median 391.0187 0       22          44
## 6   household 378.2444 0       17          24
## 7  apartments 342.5736 0       23          58
## 8         tax 313.5740 0       70         613
## 9       units 306.7513 0       46         286
## 10   bookings 301.0392 0        6           0
```

Twitter Word Relationships/Correlations

```r
# create objects to do finds correlations
# convert to tidy format and apply my stop words
cg_t_raw_text <- tidy(cg_t_corpus)

# distribution of most frequent words across documents
cg_t_raw_words <- cg_t_raw_text %>%
  unnest_tokens(word, text) %>%
  anti_join(add_stops, by = 'word') %>%
  count(word, sort = TRUE)

cg_t_report_words <- cg_t_raw_words

cg_t_par_tokens <- unnest_tokens(cg_t_raw_text, output = paragraphs, input = text, token = "paragraphs"

cg_t_par_tokens <- par_tokens %>%
 mutate(par_id = 1:n())

cg_t_par_words <- unnest_tokens(cg_t_par_tokens, output = word, input = paragraphs, token = "words")
```

```r
# find words that occur close together in the tweets
cg_t_word_pairs <- cg_t_par_words %>%
  pairwise_count(word, par_id, sort = TRUE, upper = FALSE) %>%
  anti_join(add_stops, by = c("item1" = "word")) %>%
  anti_join(add_stops, by = c("item2" = "word"))
```

```r
# plot correlations
word_pairs_t_plot <- cg_t_word_pairs %>%
  filter(n >= 200) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = n, edge_width = n), edge_colour = "dodgerblue") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name),
                 repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()

ggsave("word_pairs_t_plot.png",
       plot = word_pairs_t_plot,
       path = "plots")
```

```r
# plot correlations by paragraph
cg_t_word_cors <- cg_t_par_words %>%
  anti_join(add_stops, by = c("word" = "word")) %>%
  add_count(par_id) %>%
  filter(n >= 200) %>%
  select(-n) %>%
  pairwise_cor(word, par_id, sort = TRUE)
```

```r
key_word_corr_t_plot <- cg_t_word_cors %>%
  filter(item1 %in% c("undesirable", "income")) %>%
  group_by(item1) %>%
  top_n(6) %>%
  ungroup() %>%
  mutate(item1 = as.factor(item1),
         name = reorder_within(item2, correlation, item1)) %>%
  ggplot(aes(y = name, x = correlation, fill = item1)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ item1, ncol = 2, scales = "free") +
  scale_y_reordered() +
  labs(
    y = NULL,
    x = NULL,
    title = "Correlations with key words",
    subtitle = "Climate gentrification TWITTER"
  )

ggsave("key_word_corr_t_plot.png",
       plot = key_word_corr_t_plot,
       path = "plots")
```

```r
# let's zoom in on just one of our key terms
cg_t_undesirable_cors <- cg_t_word_cors %>%
  filter(item1 == "undesirable") %>%
  mutate(n = 1:n())

# correlation network
```

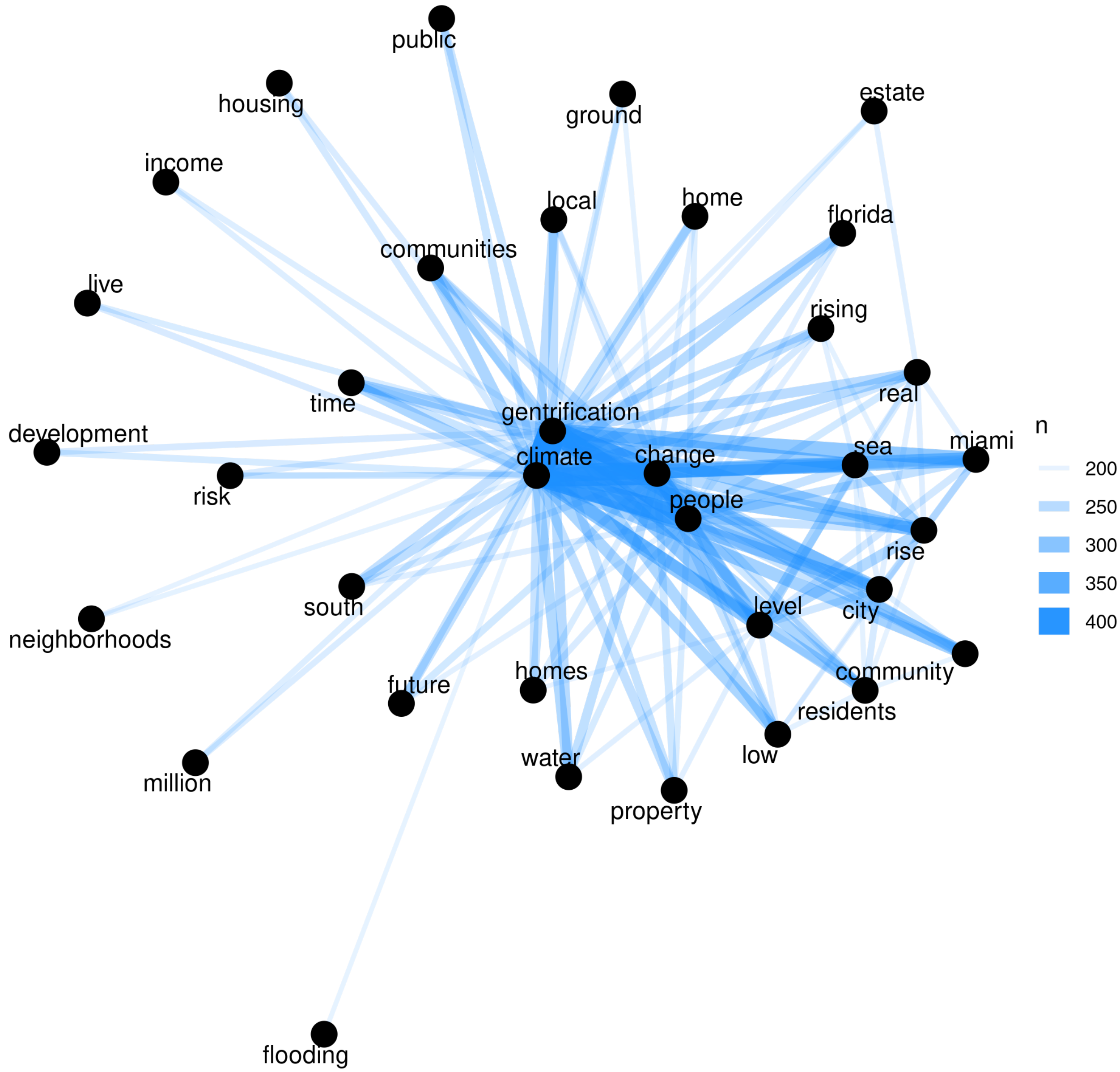


Figure 4: Twitter Word Pairs

```
undesirable_corr_t_plot <- cg_t_undesirable_cors  %>%
  filter(n <= 50) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation), edge_colour = "cyan4") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()

ggsave("undesirable_corr_t_plot.png",
       plot = undesirable_corr_t_plot,
       path = "plots")
```

```
# let's zoom in on just one of our key terms
cg_t_income_cors <- cg_t_word_cors %>%
  filter(item1 == "income") %>%
  mutate(n = 1:n())

# correlation network
income_corr_t_plot <- cg_t_income_cors  %>%
  filter(n <= 50) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation), edge_colour = "cyan4") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()

ggsave("income_corr_t_plot.png",
       plot = income_corr_t_plot,
       path = "plots")
```

```
cg_t_undesirable_cg <- c("undesirable", "climate gentrification")
cg_t_undesirable_toks_inside <- tokens_keep(cg_t_tokens, pattern = cg_t_undesirable_cg, window = 20)
cg_t_undesirable_toks_inside <- tokens_remove(cg_t_undesirable_toks_inside, pattern = cg_t_undesirable_c
cg_t_undesirable_toks_outside <- tokens_remove(cg_t_tokens, pattern = cg_t_undesirable_cg, window = 20)
```

```
cg_t_undesirable_dfmat_inside <- dfm(cg_t_undesirable_toks_inside)
cg_t_undesirable_dfmat_outside <- dfm(cg_t_undesirable_toks_outside)

cg_t_undesirable_tstat_key_inside <- textstat_keyness(rbind(cg_t_undesirable_dfmat_inside, cg_t_undesir
                                     target = seq_len(ndoc(cg_t_undesirable_dfmat_inside)))
head(cg_t_undesirable_tstat_key_inside, 10)
```

"undesirable" and climate gentrification as multi-word term of interest in Nexis Uni

```
##                      feature      chi2 p n_target n_reference
## 1                 considered 11717.380 0      642           6
## 2                     refuge 11390.924 0      635          16
## 3                    seeking 11390.924 0      635          16
## 4                   starting  9571.360 0      615         101
## 5                    effects  9294.890 0      642         155
## 6              @motherjones  7361.639 0      602         264
## 7                       move  7318.739 0      638         332
## 8  https://t.co/cumife4viv  6499.446 0      353           0
## 9                    wealthy  5956.419 0      642         526
## 10                    people  4426.292 0      644         851
```

```
cg_t_income_cg <- c("income", "climate gentrification")
cg_t_income_toks_inside <- tokens_keep(cg_t_tokens, pattern = cg_t_income_cg, window = 20)
cg_t_income_toks_inside <- tokens_remove(cg_t_income_toks_inside, pattern = cg_t_income_cg) # remove th
cg_t_income_toks_outside <- tokens_remove(cg_t_tokens, pattern = cg_t_income_cg, window = 20)
```

```
cg_t_income_dfmat_inside <- dfm(cg_t_income_toks_inside)
cg_t_income_dfmat_outside <- dfm(cg_t_income_toks_outside)

cg_t_income_tstat_key_inside <- textstat_keyness(rbind(cg_t_income_dfmat_inside, cg_t_income_dfmat_outs
                                      target = seq_len(ndoc(cg_t_income_dfmat_inside)))
head(cg_t_income_tstat_key_inside, 10)
```

"income" and climate gentrification as multi-word term of interest in Nexis Uni

```
##                      feature      chi2 p n_target n_reference
## 1                      lower 4407.6862 0       48          34
## 2                       aims 3811.2906 0       28           4
## 3           @climatelawnews 2748.5774 0       26          12
## 4                    protect  879.6801 0       31         131
## 5  https://t.co/kahmlxptoi  813.6973 0        6           0
## 6                     cooler  678.1314 0        7           3
## 7               @bsaclimate  653.4807 0        5           0
## 8                    climbed  607.5339 0        6           2
## 9                    eastern  607.5339 0        6           2
## 10                       low  504.8567 0       12          29
```

#N-gram comparison between Nexis Uni and Twitter data

```
toks2 <- tokens_ngrams(toks1, n=3)
dfm2 <- dfm(toks2)
dfm2 <- dfm_remove(dfm2, pattern = c(stop_vec))
freq_words2 <- textstat_frequency(dfm2, n=20)
freq_words2$token <- rep("trigram", 20)
freq_words2
```

```
##                                feature frequency rank docfreq group   token
```

```
## 1                    sea_level_rise       429    1      135    all trigram
## 2            adjustment_failure_costs       273    2        1    all trigram
## 3            greenhouse_gas_emissions       212    3       27    all trigram
## 4              impacts_climate_change       195    4       60    all trigram
## 5     recommendation_congress_direct       177    5        1    all trigram
## 6                   clean_future_act       147    6        1    all trigram
## 7        jurisdiction_energy_commerce       146    7        1    all trigram
## 8                  rising_sea_levels       145    8       97    all trigram
## 9           green_blue_infrastructure       127    9        1    all trigram
## 10                  fair_housing_act       115   10        9    all trigram
## 11             effects_climate_change       112   11       45    all trigram
## 12                 moving_forward_act       106   12        1    all trigram
## 13           science_space_technology       106   12        1    all trigram
## 14 environmental_justice_communities       102   14        5    all trigram
## 15           energy_commerce_building       102   14        1    all trigram
## 16            commerce_building_block       102   14        1    all trigram
## 17   environmental_protection_agency        94   17       17    all trigram
## 18             climate_change_impacts        92   18       30    all trigram
## 19      committee_jurisdiction_energy        92   18        1    all trigram
## 20            nightly_business_report        90   20        6    all trigram
```

```r
#tokens1 <- tokens_select(tokens1,pattern = stopwords("en"), selection = "remove")
```

```r
cg_t_toks2 <- tokens_ngrams(cg_t_tokens, n=3)
cg_t_dfm2 <- dfm(cg_t_toks2)
cg_t_dfm2 <- dfm_remove(cg_t_dfm2, pattern = c(stop_vec))
cg_t_freq_words2 <- textstat_frequency(cg_t_dfm2, n=20)
cg_t_freq_words2$token <- rep("trigram", 20)
cg_t_freq_words2
```

```
##                                          feature frequency rank docfreq group
## 1                         effects_climate_change       672    1     672   all
## 2            neighborhoods_considered_undesirable       642    2     642   all
## 3                  move_neighborhoods_considered       638    3     638   all
## 4                         wealthy_people_seeking       635    4     635   all
## 5                          people_seeking_refuge       635    4     635   all
## 6                         seeking_refuge_effects       635    4     635   all
## 7                         refuge_effects_climate       632    7     632   all
## 8                    starting_move_neighborhoods       615    8     615   all
## 9                          change_starting_move       614    9     614   all
## 10                      climate_change_starting       613   10     613   all
## 11                   @motherjones_wealthy_people       595   11     595   all
## 12                       rt_@motherjones_wealthy       594   12     594   all
## 13                                sea_level_rise       497   13     496   all
## 14 considered_undesirable_https://t.co/cumife4viv       353   14     353   all
## 15                   called_climate_gentrification       343   15     343   all
## 16                              like_little_haiti       341   16     341   all
## 17                            miami's_little_haiti       341   16     339   all
## 18                         target_developers_seas       340   18     338   all
## 19                        developers_seas_started       340   18     338   all
## 20                              seas_started_rise       339   20     337   all
##      token
## 1  trigram
```

```
## 2  trigram
## 3  trigram
## 4  trigram
## 5  trigram
## 6  trigram
## 7  trigram
## 8  trigram
## 9  trigram
## 10 trigram
## 11 trigram
## 12 trigram
## 13 trigram
## 14 trigram
## 15 trigram
## 16 trigram
## 17 trigram
## 18 trigram
## 19 trigram
## 20 trigram
```

```
#tokens1 <- tokens_select(tokens1,pattern = stopwords("en"), selection = "remove")
```

Topic Modeling Analysis for Nexis Uni

Optimization for k

```
result <- FindTopicsNumber(
  dfm,
  topics = seq(from = 2, to = 20, by = 1),
  metrics = c("CaoJuan2009",  "Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 77),
  verbose = TRUE
)

FindTopicsNumber_plot(result)
```

FindTopicsNumber: 4, 7, 12 k=5: 75%/30% k=7: 55%/50% k=12: 90%/25%

Topic models for k=5, k=7 and k=12

```
k <- 5

topicModel_k5 <- LDA(dfm, k, method="Gibbs", control=list(iter = 500, verbose = 25))
#nTerms(dfm_comm)

tmResult_5 <- posterior(topicModel_k5)
attributes(tmResult_5)
#nTerms(dfm_comm)
```

```r
beta_5 <- tmResult_5$terms   # get beta from results
dim(beta_5)                 # K distributions over nTerms(DTM) terms# lengthOfVocab
terms(topicModel_k5, 10)
```

```r
k <- 7

topicModel_k7 <- LDA(dfm, k, method="Gibbs", control=list(iter = 500, verbose = 25))
#nTerms(dfm_comm)

tmResult_7 <- posterior(topicModel_k7)
attributes(tmResult_7)
#nTerms(dfm_comm)
beta_7 <- tmResult_7$terms   # get beta from results
dim(beta_7)                 # K distributions over nTerms(DTM) terms# lengthOfVocab
terms(topicModel_k7, 10)
```

```r
k <- 12

topicModel_k12 <- LDA(dfm, 12, method="Gibbs", control=list(iter = 500, verbose = 25))

tmResult_12 <- posterior(topicModel_k12)
terms(topicModel_k12, 10)
theta_12 <- tmResult_12$topics
beta_12 <- tmResult_12$terms
vocab <- (colnames(beta_12))
```

Top words per topic

```r
comment_topics_5 <- tidy(topicModel_k5, matrix = "beta")

comment_topics_7 <- tidy(topicModel_k7, matrix = "beta")

comment_topics_12 <- tidy(topicModel_k12, matrix = "beta")

top_terms_5 <- comment_topics_5 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms_7 <- comment_topics_7 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms_12 <- comment_topics_12 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
```

Plots of top terms per topics

```
top_terms_5_plot <- top_terms_5 %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip() +
  labs(title="Top Terms for 5-Topic Model")

top_terms_7_plot <- top_terms_7 %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip() +
  labs(title="Top Terms for 7-Topic Model")


top_terms_12_plot <- top_terms_12 %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()+
  labs(title="Top Terms for 12-Topic Model")


top_terms_5_plot / top_terms_7_plot / top_terms_12_plot
```

Top 5 terms per topic

```
top5termsPerTopic_5 <- terms(topicModel_k5, 5)
topicNames_5 <- apply(top5termsPerTopic_5, 2, paste, collapse=" ")
topicNames_5

top5termsPerTopic_7 <- terms(topicModel_k7, 5)
topicNames_7 <- apply(top5termsPerTopic_7, 2, paste, collapse=" ")
topicNames_7


top5termsPerTopic_12 <- terms(topicModel_k12, 5)
topicNames_12 <- apply(top5termsPerTopic_12, 2, paste, collapse=" ")
topicNames_12
```

Topic Modeling Intertopic Distance Maps

```
# k=5
library(LDAvis)
```

```r
library("tsne")
svd_tsne <- function(x) tsne(svd(x)$u)
json <- createJSON(
  phi = tmResult_5$terms,
  theta = tmResult_5$topics,
  doc.length = rowSums(dfm),
  vocab = colnames(dfm),
  term.frequency = colSums(dfm),
  mds.method = svd_tsne,
  plot.opts = list(xlab="", ylab="")
)
serVis(json)
```

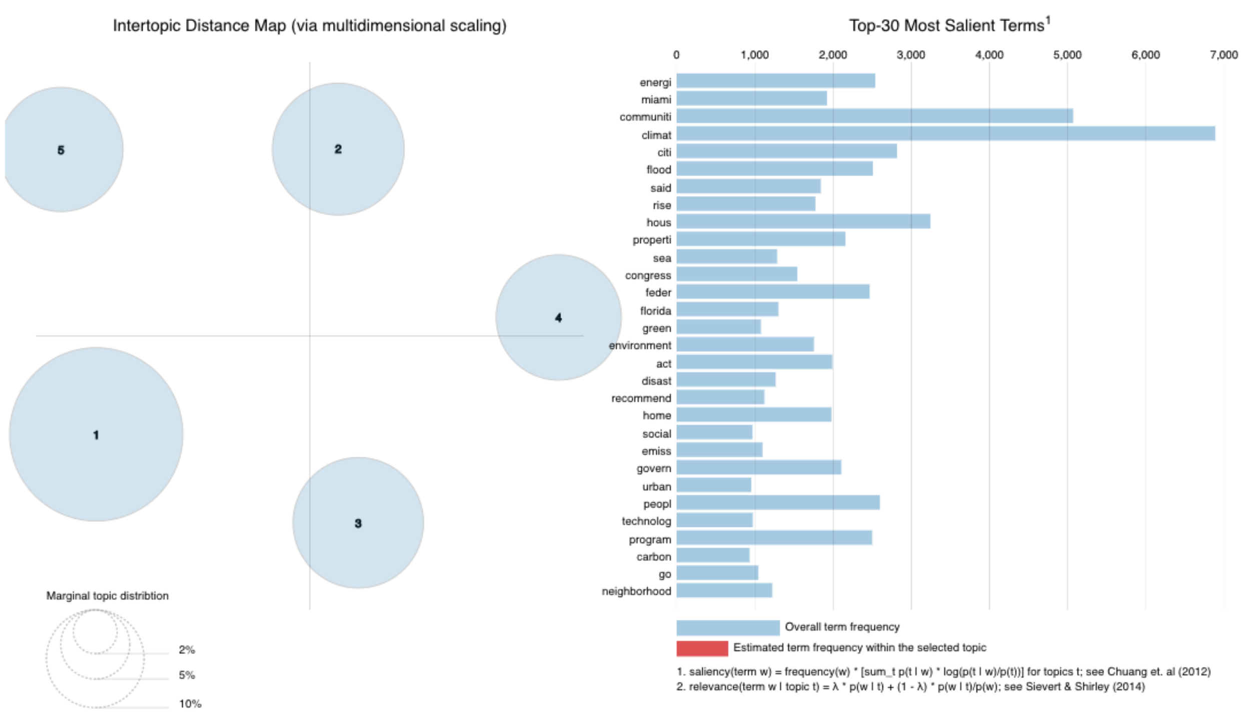


Figure 5: Topic Modeling Intertopic Distance Map for k=5

```r
# k=7
library(LDAvis)
library("tsne")
svd_tsne <- function(x) tsne(svd(x)$u)
json <- createJSON(
  phi = tmResult_7$terms,
  theta = tmResult_7$topics,
  doc.length = rowSums(dfm),
  vocab = colnames(dfm),
  term.frequency = colSums(dfm),
  mds.method = svd_tsne,
  plot.opts = list(xlab="", ylab="")
)
serVis(json)
```

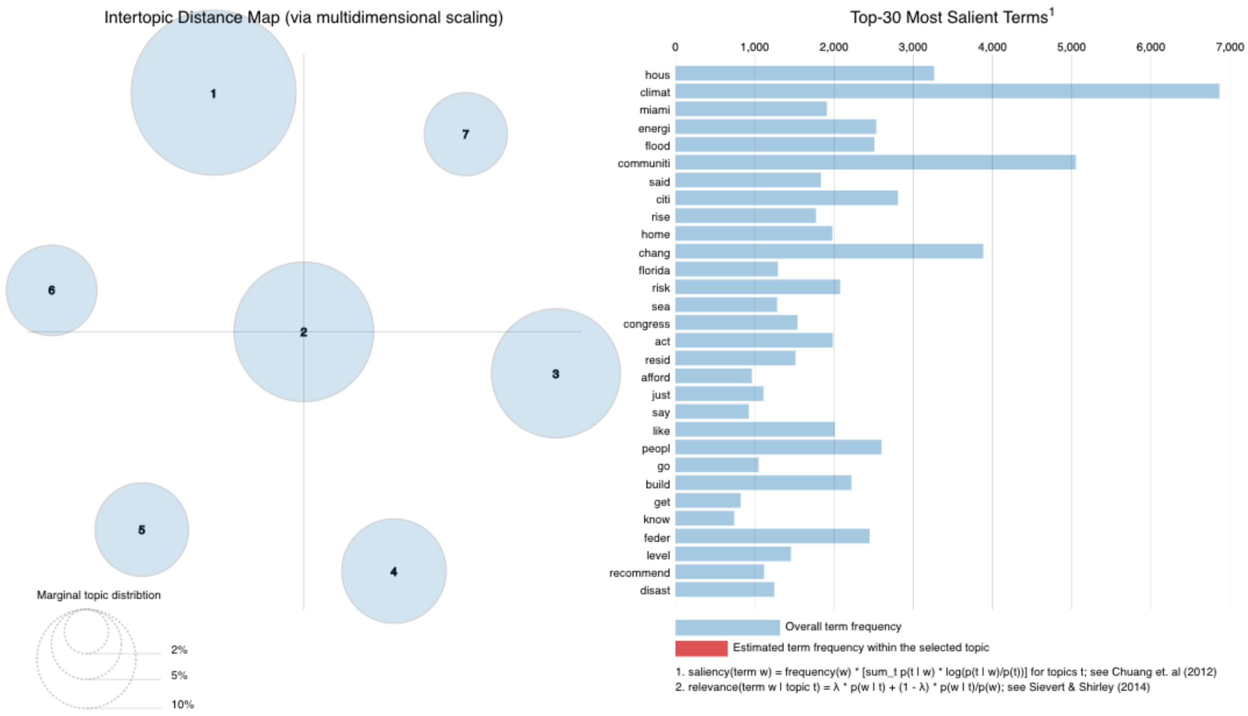


Figure 6: Topic Modeling Intertopic Distance Map for k=7

```r
# k=12
library(LDAvis)
library("tsne")
svd_tsne <- function(x) tsne(svd(x)$u)
json <- createJSON(
  phi = tmResult_12$terms,
  theta = tmResult_12$topics,
  doc.length = rowSums(dfm),
  vocab = colnames(dfm),
  term.frequency = colSums(dfm),
  mds.method = svd_tsne,
  plot.opts = list(xlab="", ylab="")
)
serVis(json)
```

Topic Modeling Analysis for Twitter Data

```r
# Simple plot of tweets per day
daily_tweets <- tweets %>%
  count(date)

daily_tweets_plot <- ggplot(daily_tweets, aes(x = date, y = n)) +
  geom_line() +
  theme_light() +
```

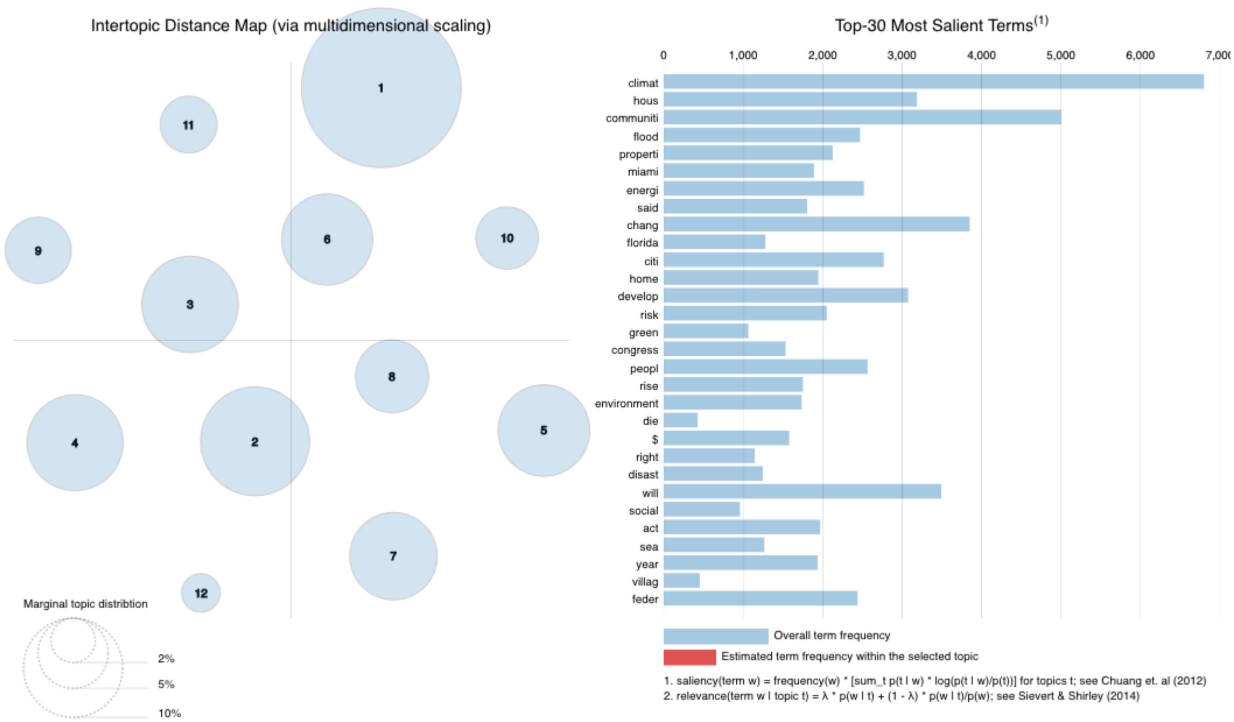


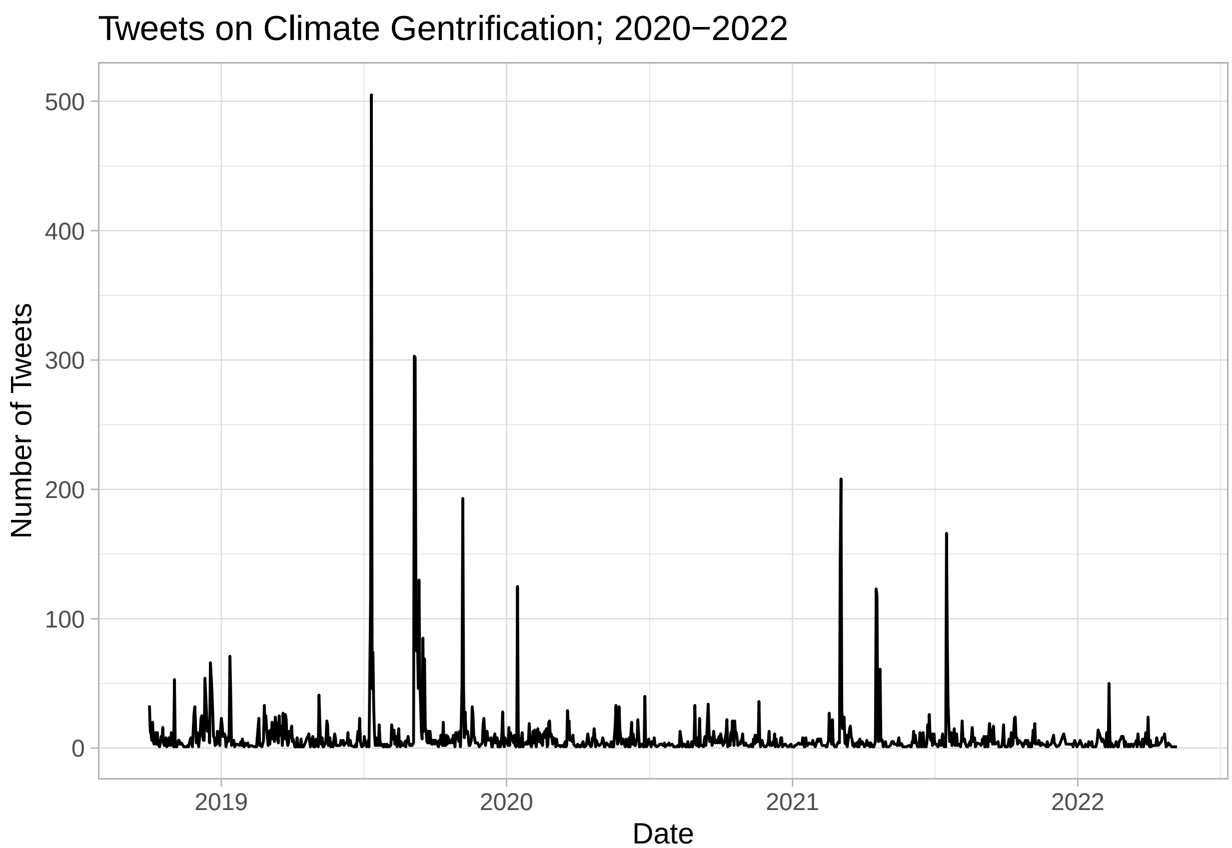
Figure 7: Topic Modeling Intertopic Distance Map for k=12

```
  labs(y = "Number of Tweets",
       x = "Date",
       title = "Tweets on Climate Gentrification; 2020-2022")

ggsave("plots/daily_tweets.png", daily_tweets_plot)
```

```
## Saving 6.5 x 4.5 in image
```

```
daily_tweets_plot
```



The date with the highest number of tweets about climate gentrification is March 4, 2021. On March 3rd, CNN released a story titled High ground, high prices, which reported on climate gentrification. Specific problems discussed in the article include Miami's Little Haiti neighborhood and New Orleans, particularly after displacement caused by Hurricane Katrina.

Another notable date is July 17th, 2021, which corresponds to the Washington Post's article covering climate gentrification following the tragic Surfside condo collapse on June 24th, 2021.

Corpus

```
cg_t_tm_corp <- corpus(x = tweets, text_field = "text")
cg_t_tm_corp.stats <- summary(cg_t_tm_corp)
cg_t_toks <- tokens(cg_t_tm_corp, remove_punct = TRUE, remove_numbers = TRUE)

# Added some project-specific stop words here
cg_t_add_stops <- c(stopwords("en"), "rt", "n", "climate", "gentrification")
cg_t_toks1 <- tokens_select(toks, pattern = cg_t_add_stops, selection = "remove")
```

```
cg_t_dfm_comm<- dfm(cg_t_toks1, tolower = TRUE)
cg_t_dfm <- dfm_wordstem(cg_t_dfm_comm)

#remove rows (docs) with all zeros
sel_idx <- slam::row_sums(cg_t_dfm) > 0
cg_t_dfm <- cg_t_dfm[sel_idx, ]
```

```r
#
cg_t_result <- FindTopicsNumber(
  cg_t_dfm,
  topics = seq(from = 2, to = 20, by = 1),
  metrics = c("CaoJuan2009",  "Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 77),
  verbose = TRUE
)

FindTopicsNumber_plot(cg_t_result)
```

```r
k <- 3

cg_t_topicModel_k3 <- LDA(cg_t_dfm, k, method="Gibbs", control=list(iter = 500, verbose = 100))

cg_t_tmResult <- posterior(cg_t_topicModel_k3)
terms(cg_t_topicModel_k3, 10)
theta <- cg_t_tmResult$topics
beta <- cg_t_tmResult$terms
vocab <- (colnames(beta))
```

```r
cg_t_comment_topics <- tidy(cg_t_topicModel_k3, matrix = "beta")

cg_t_top_terms <- cg_t_comment_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

cg_t_top_terms
```

```r
cg_t_terms_plot <- cg_t_top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()

ggsave("plots/twitter_topic_terms.png", terms_plot)
terms_plot
```

```r
cg_t_top5termsPerTopic <- terms(cg_t_topicModel_k3, 5)
cg_t_topicNames <- apply(cg_t_top5termsPerTopic, 2, paste, collapse=" ")
```

```r
exampleIds <- c(1, 2, 3, 4, 5, 6)
N <- length(exampleIds)

#lapply(epa_corp[exampleIds], as.character) #uncomment to view example text
# get topic proportions form example documents
topicProportionExamples <- theta[exampleIds,]
```

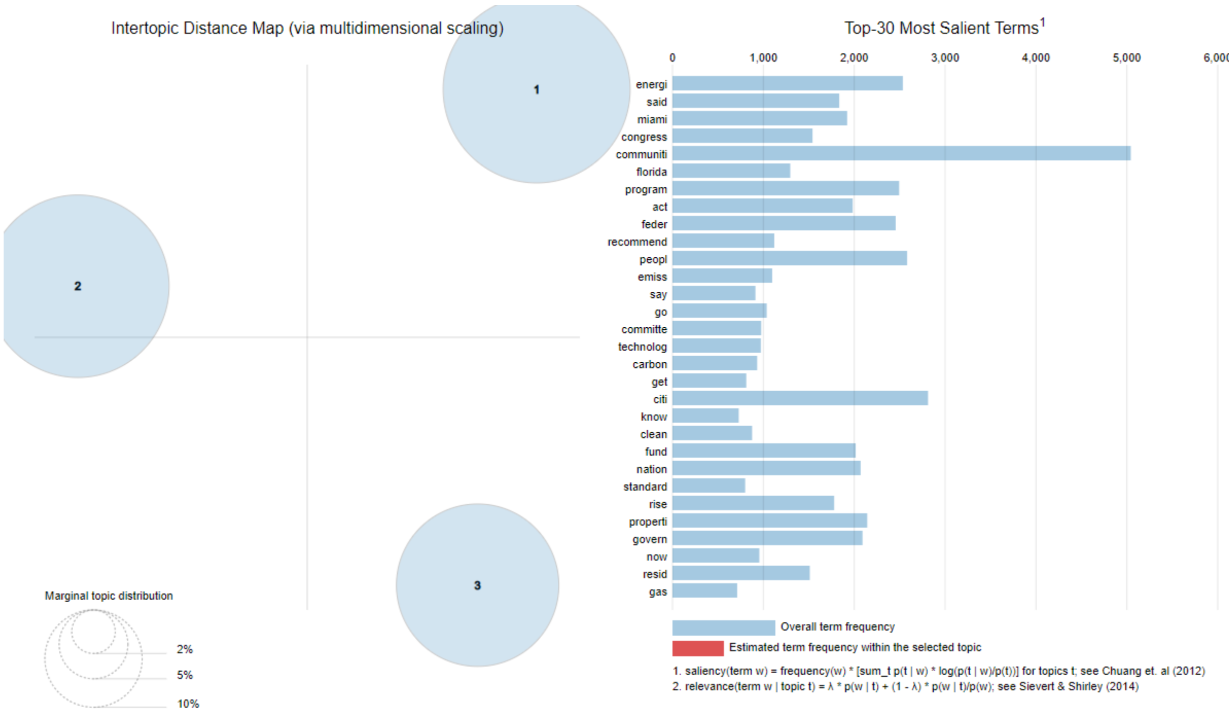


Figure 8: Twitter Topic Modeling Intertopic Distance Map for k=3

```
colnames(topicProportionExamples) <- topicNames
vizDataFrame <- reshape2::melt(cbind(data.frame(topicProportionExamples),
                          document=factor(1:N)),
                     variable.name = "topic",
                     id.vars = "document")

ggplot(data = vizDataFrame, aes(topic, value, fill = document), ylab = "proportion") +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  coord_flip() +
  facet_wrap(~ document, ncol = N)
```

Topic modeling for short form text data, such as tweets, has important limitations. For instance, given the character limit imposed on users, there is pervasive use of slang, short-hand words, and other text that will not be parsed by a topic model such as LDA.

Given the nature of Twitter data in the context of topic modeling, our analysis focused on only 3 topics. This was supported by the CaoJuan2009 and Deveaud2014 metrics. The three topics identified are, broadly…

1. Miami (Top Words: Miami, rise, neighborhood, sea, resid(ence, ents))

- It is intuitive that Miami is the primary focus of one of our topics. As stated previously, Miami neighborhood's such as Little Haiti have gained much national attention due to the pervasive climate gentrification in the area. Additionally, this can likely be attributed to active advocates for local communities in Miami, such as Valencia Gunder.

2. Housing Crisis (Top Words: Communities, hous(e, ing), will, crisis, people)

- The second topic focuses on the housing crisis and impact on individual's living situations as a result of climate gentrification.

3. Change (Top Words: Change, new, move, now, impact)

- The final topic addresses action being taken to create impactful change. The top 5 words are encouraging public participation in the issue and relate to the advocacy surrounding climate gentrification.

Avenues for Further Research

Understanding how these emotions and subjects surrounding climate gentrification vary spatially and temporally is crucial to this study, as climate gentrification captures the growing awareness of the problem in low-income communities. Twitter data contains the location of the Twitter profile of the message while Nexis data contains a geographic classification of each news article. Analyzing text from both Twitter and Nexis Uni will allow the team to compare sentiment between different data sources.

Many studies have conducted sentiment analysis and topic modeling on climate change issues using geotagged Tweets and other data sources. For instance, the Dahal et al. 2019 paper successfully used Twitter data to evaluate public opinion on climate change over space and time. The study used Latent Dirichlet Allocation for topic modeling and Valence Aware Dictionary and Sentiment Reasoner for sentiment analysis. However, no thorough research has been completed specifically focusing on spreading awareness of climate gentrification.

Citations

Dahal, B., Kumar, S.A.P. & Li, Z. Topic modeling and sentiment analysis of global climate change tweets. Soc. Netw. Anal. Min. 9, 24 (2019). https://doi.org/10.1007/s13278-019-0568-8

Keenan, Jesse & Hill, Thomas & Gumber, Anurag. (2018). Climate gentrification: From theory to empiricism in Miami-Dade County, Florida. Environmental Research Letters. 13. 14. 10.1088/1748-9326/aabb32.