Topic 5: Word Relationships

import EPA EJ Data

```
#setwd(here('data/'))

files <- list.files(path = here("data_EPA/"),
                    pattern = "EPA*", full.names = TRUE)

ej_reports <- lapply(files, pdf_text)


ej_pdf <- readtext(file = here("data_EPA", "EPA*"),
                   docvarsfrom = "filenames",
                   docvarnames = c("type", "subj", "year"),
                   sep = "_")

#creating an initial corpus containing our data
epa_corp <- corpus(x = ej_pdf, text_field = "text")
summary(epa_corp)
```

```
## Corpus consisting of 6 documents, showing 6 documents:
##
##             Text Types Tokens Sentences type subj year
##  EPA_EJ_2015.pdf  2136   8944       263  EPA   EJ 2015
##  EPA_EJ_2016.pdf  1599   7965       176  EPA   EJ 2016
##  EPA_EJ_2017.pdf  3973  30564       653  EPA   EJ 2017
##  EPA_EJ_2018.pdf  2774  16658       447  EPA   EJ 2018
##  EPA_EJ_2019.pdf  3773  22648       672  EPA   EJ 2019
##  EPA_EJ_2020.pdf  4493  30523       987  EPA   EJ 2020
```

```
# adding additional, context-specific stop words to stop word lexicon
more_stops <-c("2015","2016", "2017", "2018", "2019", "2020", "www.epa.gov", "https")
add_stops<- tibble(word = c(stop_words$word, more_stops))
stop_vec <- as_vector(add_stops)
```

create different data objects to set up for subsequent analyses

```
#convert to tidy format and apply stop words
raw_text <- tidy(epa_corp)

#Distribution of most frequent words across documents
raw_words <- raw_text %>%
  mutate(year = as.factor(year)) %>%
```

```
  unnest_tokens(word, text) %>%
  anti_join(add_stops, by = 'word') %>%
  count(year, word, sort = TRUE)

#number of total words by document
total_words <- raw_words %>%
  group_by(year) %>%
  summarize(total = sum(n))

report_words <- left_join(raw_words, total_words)
```

```
## Joining, by = "year"
```

```
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")
```

quanteda

```
tokens <- tokens(epa_corp, remove_punct = TRUE)
toks1<- tokens_select(tokens, min_nchar = 3)
toks1 <- tokens_tolower(toks1)
toks1 <- tokens_remove(toks1, pattern = (stop_vec))
dfm <- dfm(toks1)

#first the basic frequency stat
tstat_freq <- textstat_frequency(dfm, n = 5, groups = year)
head(tstat_freq, 10)
```

```
##          feature frequency rank docfreq group
## 1  environmental       127    1       1  2015
## 2    communities        99    2       1  2015
## 3            epa        92    3       1  2015
## 4        justice        84    4       1  2015
## 5      community        47    5       1  2015
## 6  environmental       109    1       1  2016
## 7    communities        85    2       1  2016
## 8        justice        71    3       1  2016
## 9            epa        48    4       1  2016
## 10       federal        31    5       1  2016
```

bigrams

```
# bigrams
```

```
toks2 <- tokens_ngrams(toks1, n=2)
dfm2 <- dfm(toks2)
dfm2 <- dfm_remove(dfm2, pattern = c(stop_vec))
freq_words2 <- textstat_frequency(dfm2, n=20)
freq_words2$token <- rep("bigram", 20)
#tokens1 <- tokens_select(tokens1,pattern = stopwords("en"), selection = "remove")
```

Assignment

1. What are the most frequent trigrams in the dataset? How does this compare to the most frequent bigrams? Which n-gram seems more informative here, and why?

```
toks3 <- tokens_ngrams(toks1, n=3)
dfm3 <- dfm(toks3)
dfm3 <- dfm_remove(dfm3, pattern = c(stop_vec))
freq_words3 <- textstat_frequency(dfm3, n=20)
freq_words3$token <- rep("trigram", 20)
```

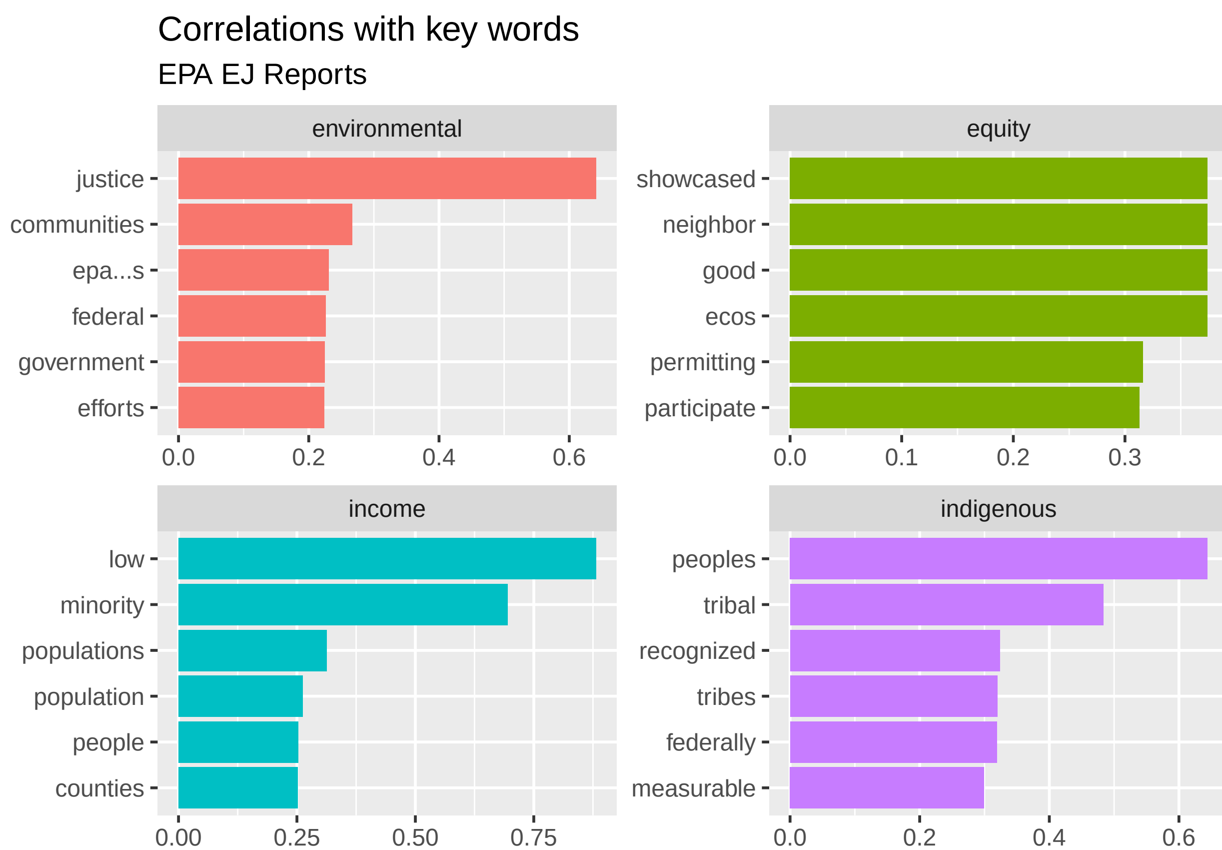
The most frequent trigrams in the dataset are justice_fy2017_progress, fy2017_progress_report, environmental_public_health, environmental_justice_fy2017, and national_environmental_justice, followed by more trigrams including more variations of environmental justice. All of the trigrams have a frequency of 51 or less. In comparison, the most frequent bigram is environmental_justice with a frequency of 556, followed by technical_assistance, drinnking_water, public_health, and progress_report, which are all between 108-139, far behind environmental_justice. Because environmental_justice is used so frequently as seen in the bigram dataframe, I would say the trigrams are more informative because the various contexts of how environmental justice is used is more clearly highlighted when adding the extra word (i.e., environmental justice progress, environmental justice grants, communities environmental justice, etc.).

2. Choose a new focal term to replace "justice" and recreate the correlation table and network (see corr_paragraphs and corr_network chunks). Explore some of the plotting parameters in the cor_network chunk to see if you can improve the clarity or amount of information your plot conveys. Make sure to use a different color for the ties!

```
word_cors_2 <- par_words %>%
  add_count(par_id) %>%
  filter(n >= 50) %>%
  select(-n) %>%
  pairwise_cor(word, par_id, sort = TRUE)

indig_cors <- word_cors_2 %>%
  filter(item1 == "indigenous")

  word_cors_2 %>%
  filter(item1 %in% c("environmental", "indigenous", "equity", "income"))%>%
  group_by(item1) %>%
  top_n(6) %>%
  ungroup() %>%
  mutate(item1 = as.factor(item1),
  name = reorder_within(item2, correlation, item1)) %>%
```

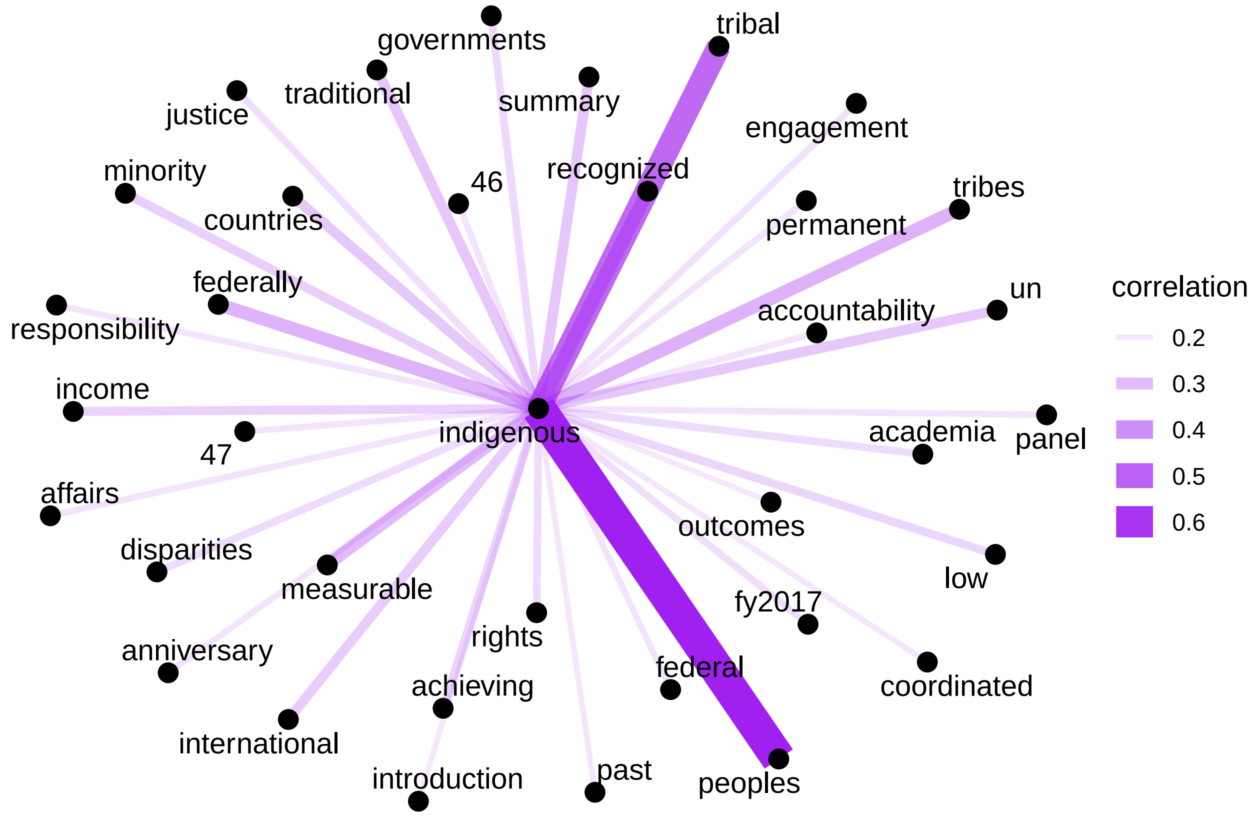
```
  ggplot(aes(y = name, x = correlation, fill = item1)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~item1, ncol = 2, scales = "free")+
  scale_y_reordered() +
  labs(y = NULL,
         x = NULL,
         title = "Correlations with key words",
         subtitle = "EPA EJ Reports")
```

```
## Selecting by correlation
```




```
  #let's zoom in on just one of our key terms
   indigenous_cors <- word_cors_2 %>%
  filter(item1 == "indigenous") %>%
   mutate(n = 1:n())
```

```
indigenous_cors  %>%
  filter(n <= 35) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation), edge_colour = "purple") +
  geom_node_point(size = 3) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()
```



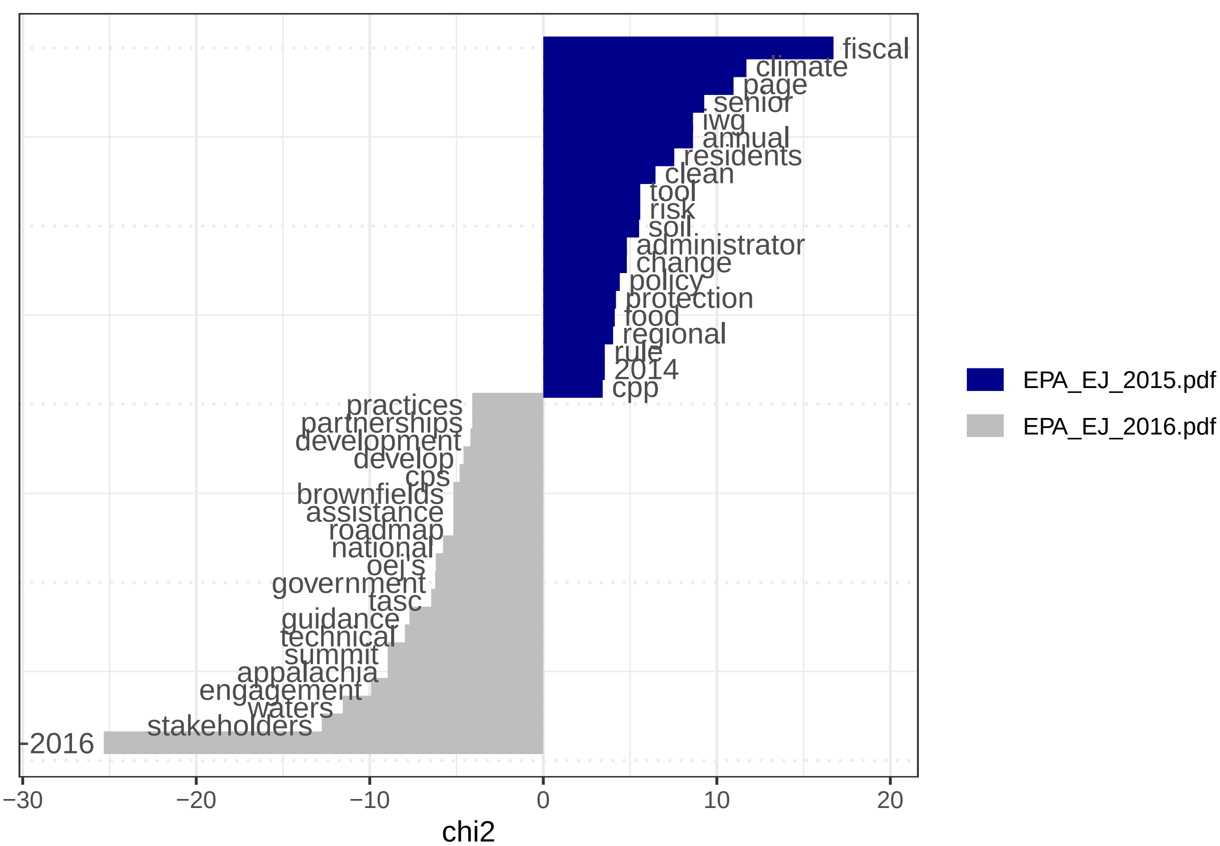
I chose the word "indigenous" for this exercise. When creating the network I decided to modify the color parameter, decreased the node size, and also filtered for only the top 35 words in order to reduce some of the crowdedness of the plot.

3. Write a function that allows you to conduct a keyness analysis to compare two individual EPA reports (hint: that means target and reference need to both be individual reports). Run the function on 3 pairs of reports, generating 3 keyness plots.

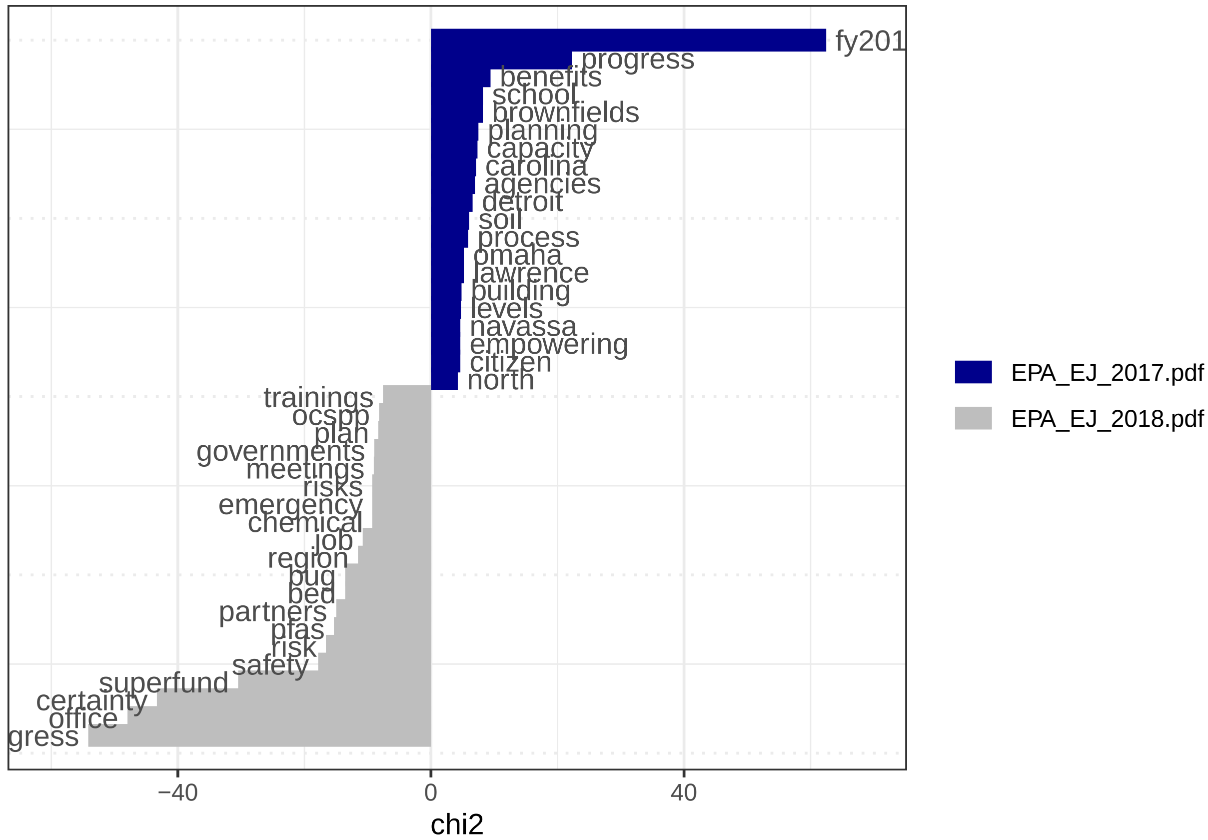
```
# make new dfm with just two reports so that the target is one doc and the reference is another single

# two different years will be the two parameters
# before making corpus, filter for only the two relevant years
# use one year as the target_year and the other will automatically be the reference_year

keyness_function <- function(target_year, reference_year){
  files <- list.files(path = here("data_EPA/"),
                    pattern = "EPA*", full.names = TRUE)

  ej_reports <- lapply(files, pdf_text)
  ej_pdf <- readtext(file = here("data_EPA", "EPA*"),
                   docvarsfrom = "filenames",
                   docvarnames = c("type", "subj", "year"),
                   sep = "_")

  ej_pdf_filtered <- ej_pdf %>%
    filter(year == target_year | year == reference_year)
```

```r
# creating an initial corpus containing the data
epa_corp <- corpus(x = ej_pdf_filtered, text_field = "text")

# adding some additional, context-specific stop words to stop word lexicon
more_stops <-c("2015","2016", "2017", "2018", "2019", "2020", "www.epa.gov", "https")
add_stops<- tibble(word = c(stop_words$word, more_stops))
stop_vec <- as_vector(add_stops)

tokens <- tokens(epa_corp, remove_punct = TRUE)
toks1<- tokens_select(tokens, min_nchar = 3)
toks1 <- tokens_tolower(toks1)
toks1 <- tokens_remove(toks1, pattern = (stop_vec))
dfm <- dfm(toks1)

keyness <- textstat_keyness(dfm, target = 1) # target = 1 is the first report, dfm2 = bigrams
return(textplot_keyness(keyness))

}
```

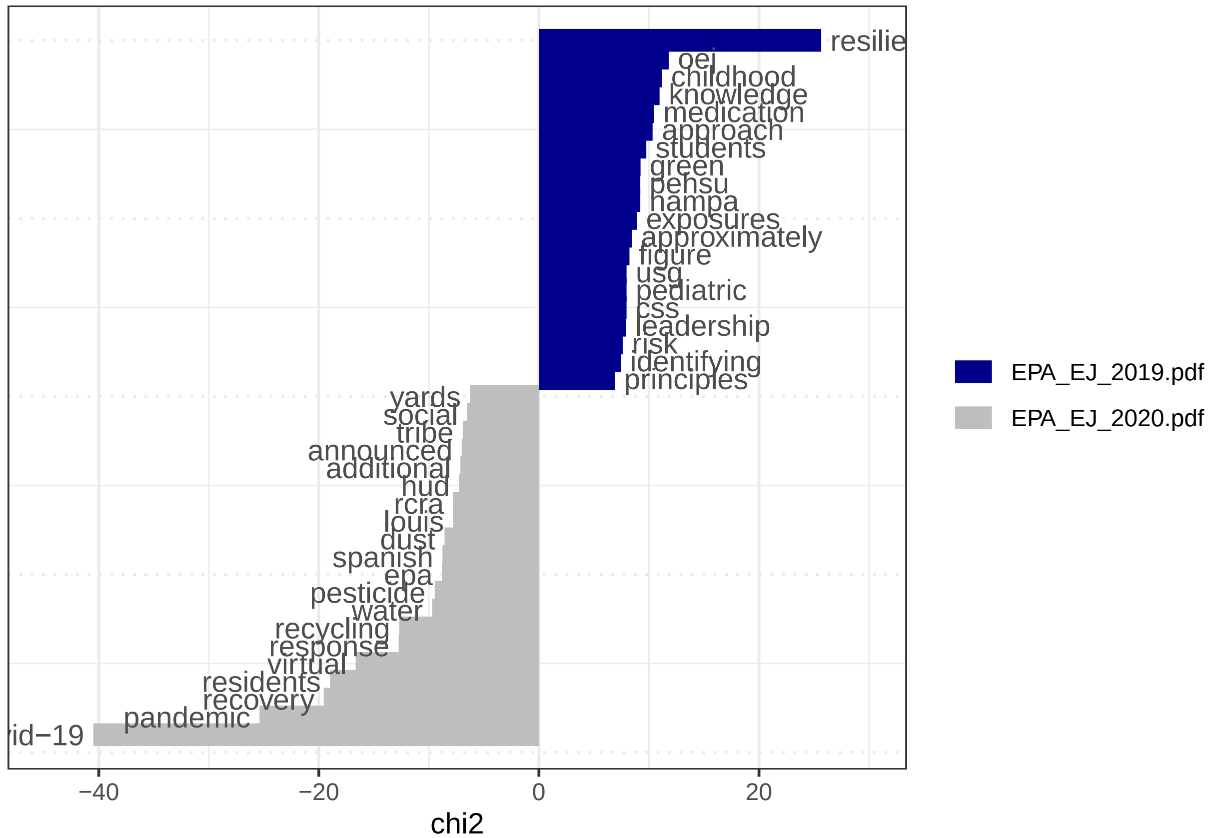
```r
# produce 3 keyness plots

keyness_function(2015, 2016)
```



```
keyness_function(2017, 2018)
```




```
keyness_function(2019, 2020)
```



4. Select a word or multi-word term of interest and identify words related to it using windowing and keyness comparison. To do this you will create two objects: one containing all words occurring within a 10-word window of your term of interest, and the second object containing all other words. Then run a keyness comparison on these objects. Which one is the target, and which the reference? Hint

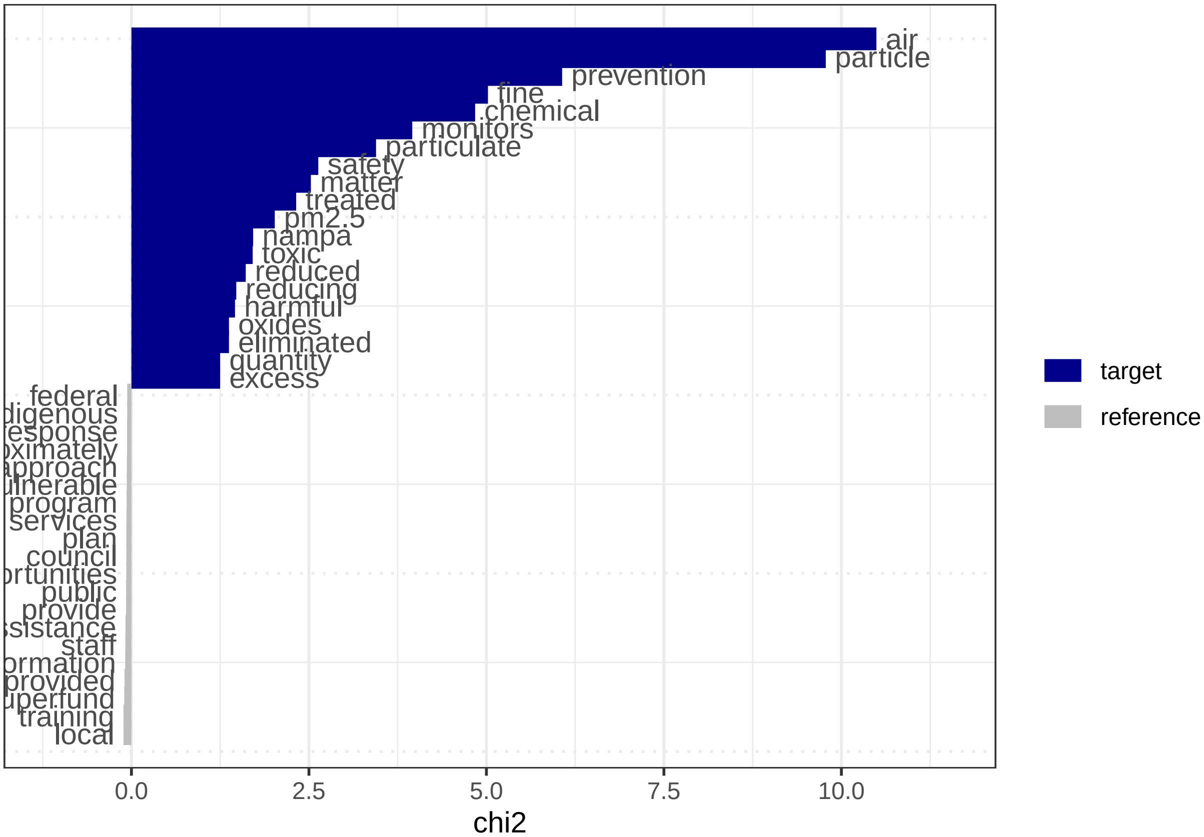
```
# use toks1

pollution <- c("pollution", "pollut*")

toks_inside <- tokens_keep(toks1, pattern = pollution, window = 10)

toks_inside <- tokens_remove(toks1, pattern = pollution) # remove the keywords

toks_outside <- tokens_remove(toks1, pattern = pollution, window = 10)
```

```
# compute words' association with the keywords using textstat_keyness
dfmat_inside <- dfm(toks_inside)
dfmat_outside <- dfm(toks_outside)

tstat_key_inside <- textstat_keyness(rbind(dfmat_inside, dfmat_outside),
                                     target = seq_len(ndoc(dfmat_inside)))
head(tstat_key_inside, 50)
```

```
##          feature       chi2           p n_target n_reference
## 1            air 10.4932311 0.001198127      246         171
```

```
## 2        particle  9.7801057 0.001764102       13          1
## 3      prevention  6.0664533 0.013777334       39         19
## 4            fine  5.0215466 0.025033800       12          3
## 5        chemical  4.8411840 0.027787810       38         20
## 6        monitors  3.9556951 0.046713021        6          0
## 7     particulate  3.4454616 0.063425701       21         10
## 8          safety  2.6319623 0.104732200       52         35
## 9          matter  2.5273475 0.111888156       22         12
## 10        treated  2.3203391 0.127692189       10          4
## 11          pm2.5  2.0192685 0.155313950       30         19
## 12          nampa  1.7143082 0.190427352        9          4
## 13          toxic  1.7080223 0.191242168       24         15
## 14        reduced  1.6101894 0.204465174        7          2
## 15       reducing  1.4767277 0.224287209       37         26
## 16        harmful  1.4587067 0.227136011       10          5
## 17     eliminated  1.3740114 0.241123867        5          1
## 18         oxides  1.3740114 0.241123867        5          1
## 19         excess  1.2490132 0.263741031        3          0
## 20       quantity  1.2490132 0.263741031        3          0
## 21     stationary  1.2490132 0.263741031        3          0
## 22           vocs  1.2490132 0.263741031        3          0
## 23         ocspp  1.1989888 0.273523871       20         13
## 24        engines  1.1669314 0.280032661        8          4
## 25 nonattainment  1.1669314 0.280032661        8          4
## 26      hazardous  1.1618987 0.281072108       37         27
## 27         pounds  1.1385378 0.285961516       16         10
## 28           13.7  0.9577534 0.327753565        1          0
## 29           139f  0.9577534 0.327753565        1          0
## 30           14.2  0.9577534 0.327753565        1          0
## 31           19.3  0.9577534 0.327753565        1          0
## 32           1990  0.9577534 0.327753565        1          0
## 33          2,500  0.9577534 0.327753565        1          0
## 34         2-year  0.9577534 0.327753565        1          0
## 35           2003  0.9577534 0.327753565        1          0
## 36          2006-  0.9577534 0.327753565        1          0
## 37            208  0.9577534 0.327753565        1          0
## 38            3.9  0.9577534 0.327753565        1          0
## 39         35,000  0.9577534 0.327753565        1          0
## 40           37.3  0.9577534 0.327753565        1          0
## 41           43.1  0.9577534 0.327753565        1          0
## 42          5,200  0.9577534 0.327753565        1          0
## 43            562  0.9577534 0.327753565        1          0
## 44         7-part  0.9577534 0.327753565        1          0
## 45         82,000  0.9577534 0.327753565        1          0
## 46   all-electric  0.9577534 0.327753565        1          0
## 47        amended  0.9577534 0.327753565        1          0
## 48    anonymously  0.9577534 0.327753565        1          0
## 49       athletic  0.9577534 0.327753565        1          0
## 50     behavioral  0.9577534 0.327753565        1          0
```

```
textplot_keyness(tstat_key_inside)
```



I chose the word pollution as my word of interest. When running the keyness comparison, the object including the words occurring within a 10-word window is the target object (dfmat_inside), and the object containing all words outside of the window is the reference object (dfmat_outside).