EDS 231 - Word Relationships

Juliet Cohen

2022-05-01

```r
library(tidyr) #text analysis in R
library(pdftools)
library(lubridate) #working with date data
library(tidyverse)
library(tidytext)
library(readr)
library(quanteda)
library(readtext) #quanteda subpackage for reading pdf
library(quanteda.textstats)
library(quanteda.textplots)
library(ggplot2)
library(forcats)
library(stringr)
library(quanteda.textplots)
library(widyr)# pairwise correlations
library(igraph) #network plots
library(ggraph)
library(gt)
```

Import EPA EJ Data

```r
setwd(".")
files <- list.files(pattern = "*pdf$")

files <- str_subset(files, pattern="EPA")

ej_reports <- lapply(files, pdf_text)

ej_pdf <- readtext(files, 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  2774  16658       447  EPA   EJ 2017
```

```
## EPA_EJ_2018.pdf  3973  30564       653  EPA   EJ 2018
## EPA_EJ_2019.pdf  3773  22648       672  EPA   EJ 2019
## EPA_EJ_2020.pdf  4493  30523       987  EPA   EJ 2020
```

```
#I'm 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)
```

Now we'll create some different data objects that will set us up for the subsequent analyses

```
#convert to tidy format and apply my 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)

# for the analysis that we want to do at the word level:
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")
```

```
tokens <- tokens(epa_corp, remove_punct = TRUE) # create token obj
toks1<- tokens_select(tokens, min_nchar = 3)
toks1 <- tokens_tolower(toks1)
toks1 <- tokens_remove(toks1, pattern = (stop_vec)) # remove stop words
dfm <- dfm(toks1) # has docs in 1 col, the rows refer to num of occurrences for each word in the corpus
# fundamental obj for text analysis in quanteda

#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
```

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?

Start by looking at bigrams:

```
toks2 <- tokens_ngrams(toks1, n=2) # bigram, tokenize, it goes thru the text with a 2 word window and c
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")

head(freq_words2)
```

```
##                  feature frequency rank docfreq group  token
## 1 environmental_justice       556    1       6   all bigram
## 2  technical_assistance       139    2       6   all bigram
## 3        drinking_water       133    3       6   all bigram
## 4         public_health       123    4       6   all bigram
## 5       progress_report       108    5       6   all bigram
## 6           air_quality        73    6       6   all bigram
```

The top 5 most frequent bigrams are:
1. environmental__justice
2. technical__assistance
3. drinking__water
4. public__health
5. progress__report

```
toks2 <- tokens_ngrams(toks1, n = 3) # trigram, tokenize, it goes thru the text with a 3 word window an
dfm2 <- dfm(toks2)
dfm2 <- dfm_remove(dfm2, pattern = c(stop_vec))
freq_words2 <- textstat_frequency(dfm2, n=20)
freq_words2$token <- rep("trigram", 20)

head(freq_words2)
```

```
##                            feature frequency rank docfreq group   token
## 1        justice_fy2017_progress        51    1       1   all trigram
## 2          fy2017_progress_report        51    1       1   all trigram
## 3     environmental_public_health        50    3       6   all trigram
## 4    environmental_justice_fy2017        50    3       1   all trigram
## 5 national_environmental_justice        37    5       6   all trigram
## 6   office_environmental_justice        32    6       6   all trigram
```

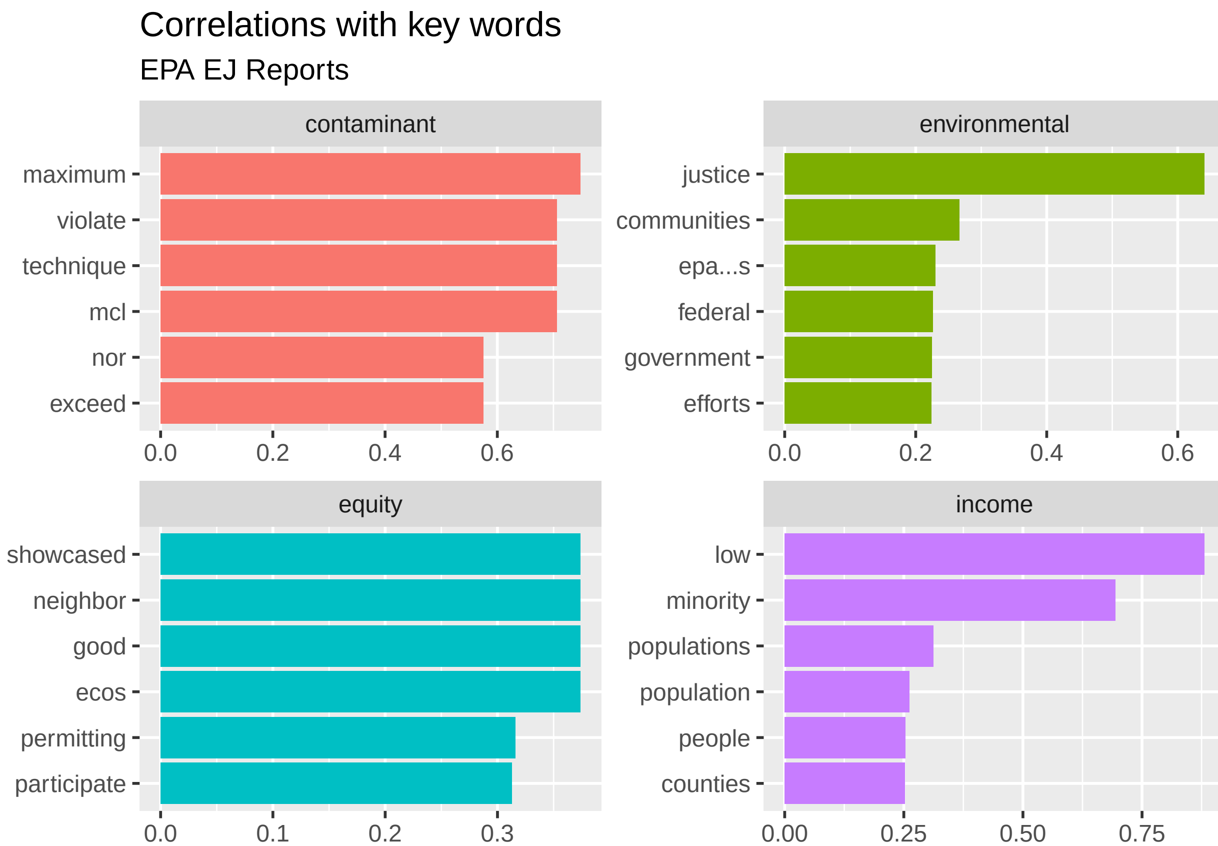
The top 5 most frequent trigrams are:
1. justice__fy2017__progress
2. fy2017__progress__report
3. environmental__public__health
4. environmental__justice__fy2017
5. national__environmental__justice

The trigrams show more repetitive words such as justice, progress, fy2017, and environmental, and appear to be words that do not form a sensical, stand-alone phrase when read together, while the bigrams are more diverse and the words make sense when read together in sequence. Therefore I think that bigrams are more informative here.

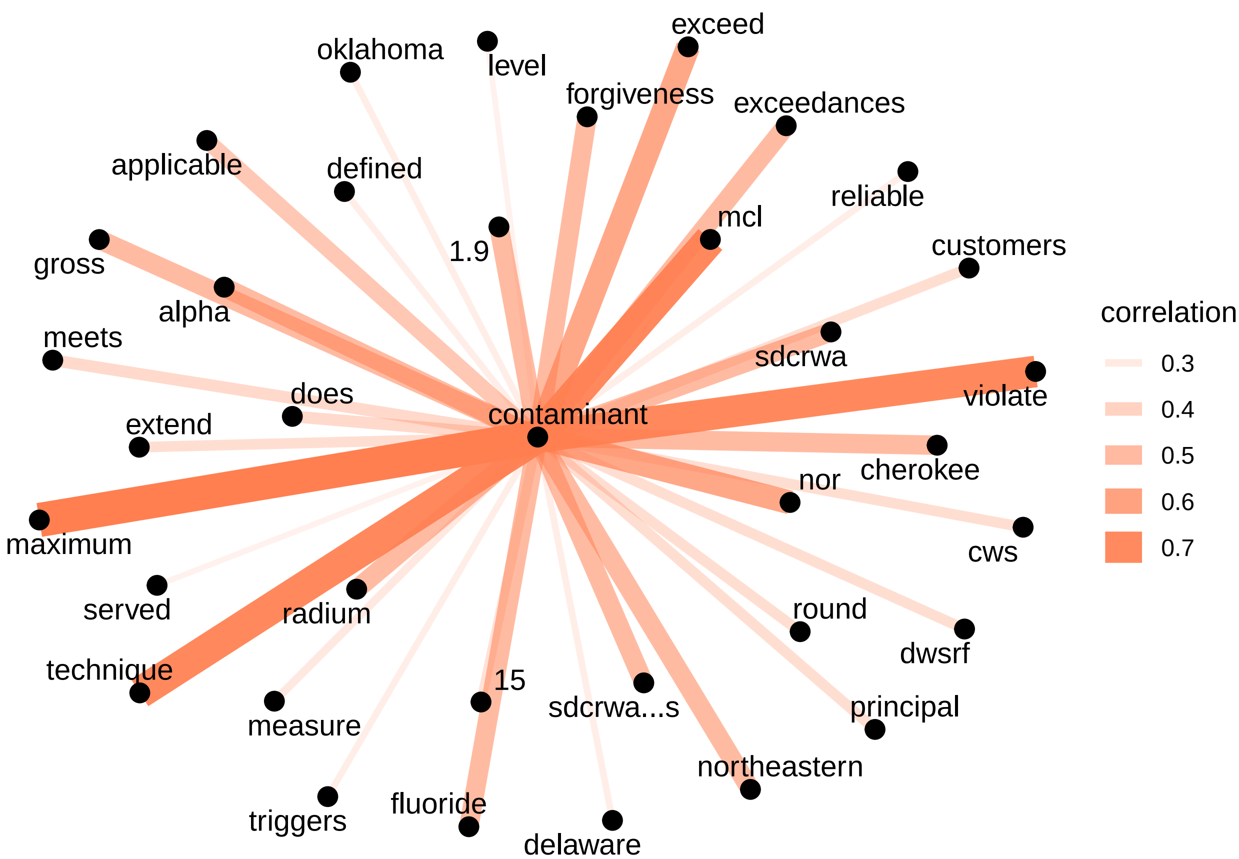
2. Choose a new focal term to replace "justice" and recreate the correlation table and network (see corr_paragraphs and corr_network chunks). Explore some of the plotting parameters in the cor_network chunk to see if you can improve the clarity or amount of information your plot conveys. Make sure to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

I replaces the term "justice" with the term "contaminant" and recreated the correlation table and network. I explored some of the plotting parameters to improve the clarity and amount of information conveyed by the plot. I used a different color for the ties.

```
word_cors <- par_words %>%
  add_count(par_id) %>%
  filter(n >= 50) %>%
  select(-n) %>%
  pairwise_cor(word, par_id, sort = TRUE) # generates correlation coefficients rather than just the num
# cols = item1 and item2 and correlation

contaminant_cors <- word_cors %>%
  filter(item1 == "contaminant")

word_cors %>%
  filter(item1 %in% c("environmental", "contaminant", "equity", "income")) %>%
  group_by(item1) %>%
  top_n(6) %>%
  #slice_max(item1, 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")
```




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

contaminant_cors  %>%
  filter(n <= 35) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation), edge_colour = "coral") +
  geom_node_point(size = 3) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.5, "lines")) +
  theme_void()
```



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.

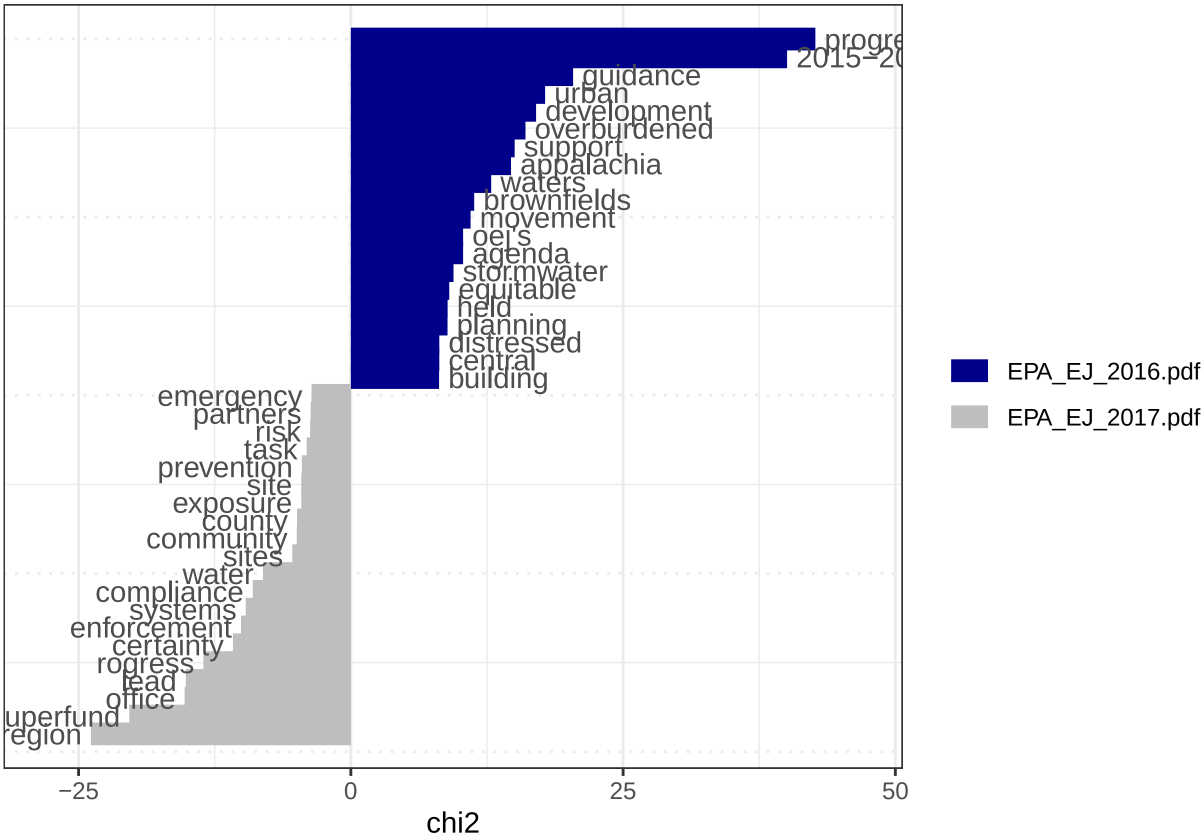
Use all of the frequencies for each word in each document and calculate a chi-square to see which words occur significantly more or less within a particular target document.

Function:

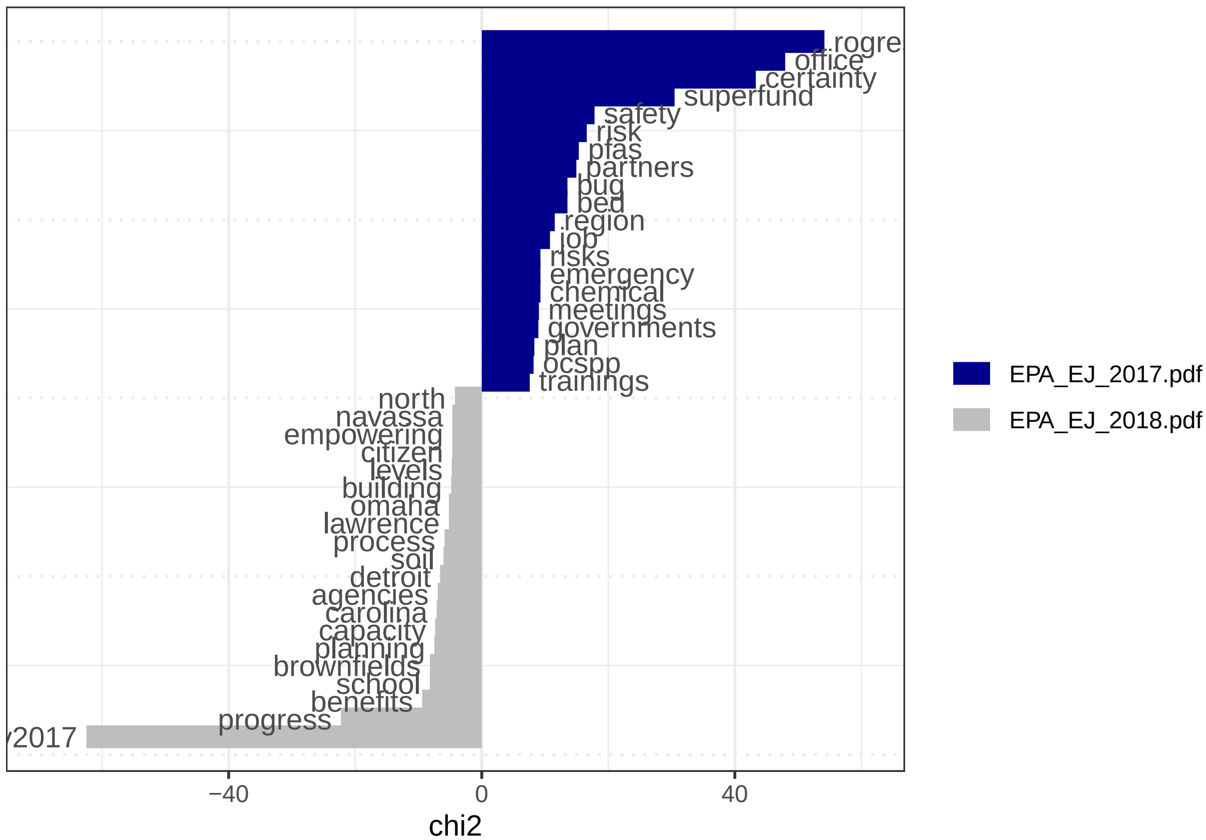
```
keyness_analysis <- function(index) {
    epa_corp_subset <- c(epa_corp[index], epa_corp[index + 1])
    tokens <- tokens(epa_corp_subset, remove_punct = TRUE) # create token obj
    toks1 <- tokens_select(tokens, min_nchar = 3)
    toks1 <- tokens_tolower(toks1)
    toks1 <- tokens_remove(toks1, pattern = (stop_vec)) # remove stop words
    dfm <- dfm(toks1)

    keyness <- textstat_keyness(dfm, target = 1)
    return(textplot_keyness(keyness))
}
```

Run function to demonstrate 3 EPA comparisons:

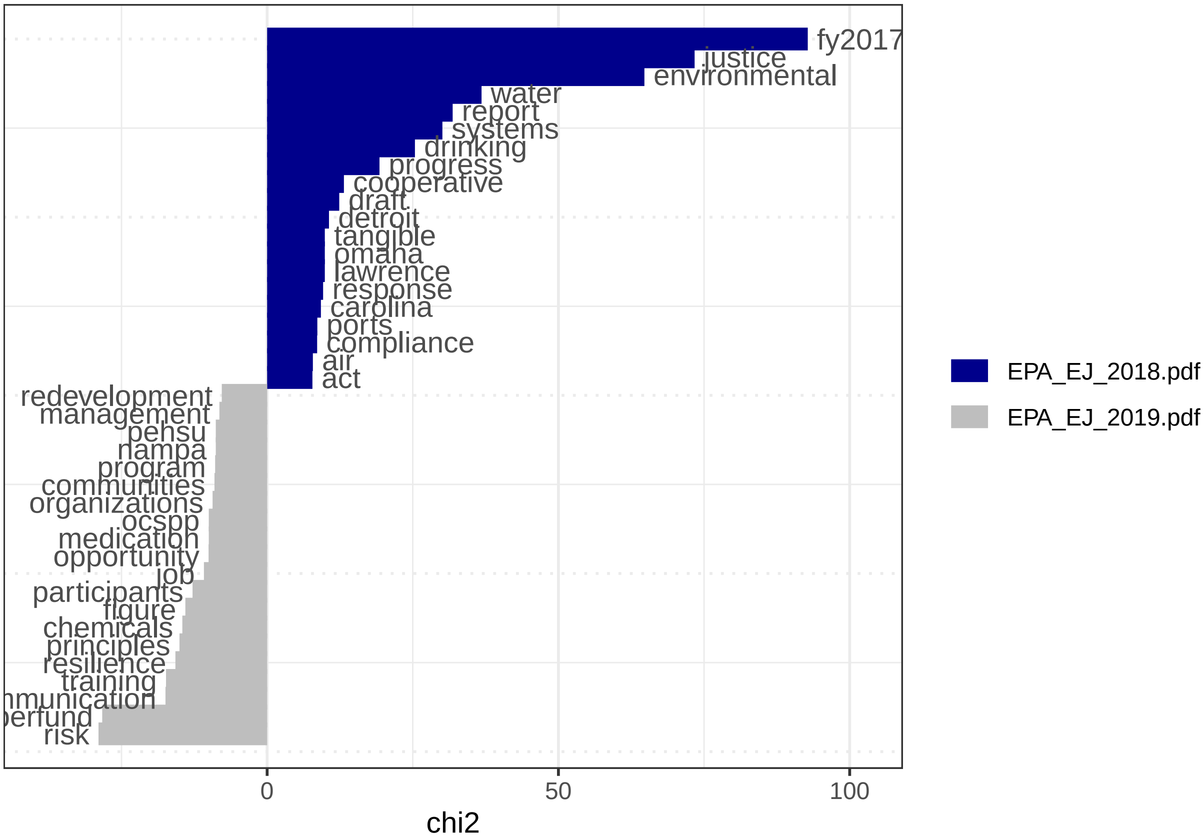
```r
keyness_analysis(index = 2)
```




```r
keyness_analysis(index = 3)
```




```
keyness_analysis(index = 4)
```



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

All words occurring within a 10-word window of my term of interest ("contaminant" and its variants) is represented by the object `toks_inside` which serves as the target, and all other words are represented by the object `toks_outside` which serves as the reference. In the dataframe I create called `tstat_key_inside`, the columns `n_target` and `n_reference` contribute to the statistical analysis done on the word associations.

```
# start with the obj toks1 because that is in the format we want
# create an object containing all words occurring within a 10-word window of my term of interest: conta
# We select two tokens objects for words inside and outside of the 10-word windows of the keywords
contam_words <- c("contaminant", "contamination", "contaminating", "contaminated", "contaminate", "conta
toks_inside <- tokens_keep(toks1, pattern = contam_words, window = 10)
toks_inside <- tokens_remove(toks_inside, pattern = contam_words) # remove the keywords
toks_outside <- tokens_remove(toks1, pattern = contam_words, window = 10)

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

# combine the objects
tstat_key_inside <- textstat_keyness(rbind(dfmat_inside, dfmat_outside),
```

```r
                                    target = seq_len(ndoc(dfmat_inside)))

# take a look at the top 10 words associated with my term of interest
head(tstat_key_inside, 20)
```

```
##          feature      chi2            p n_target n_reference
## 1          cubic 158.72215 0.000000e+00        6           0
## 2       sediment 145.06197 0.000000e+00        7           2
## 3           site 125.25917 0.000000e+00       29         124
## 4         eating 104.62644 0.000000e+00        5           1
## 5           fish  80.65981 0.000000e+00        9          16
## 6  investigating  75.48850 0.000000e+00        4           1
## 7          reuse  73.62987 0.000000e+00       10          23
## 8     properties  64.20064 1.110223e-15       11          33
## 9       decision  59.26840 1.376677e-14        5           5
## 10         yards  58.41687 2.120526e-14        7          13
## 11         canal  51.96727 5.643264e-13        4           3
## 12       maximum  47.61521 5.186407e-12        3           1
## 13          soil  44.61588 2.397393e-11        9          31
## 14       putting  37.11807 1.111886e-09        3           2
## 15   sustainably  37.11807 1.111886e-09        3           2
## 16           245  34.87283 3.519585e-09        2           0
## 17     consuming  34.87283 3.519585e-09        2           0
## 18           doe  34.87283 3.519585e-09        2           0
## 19         lead-  34.87283 3.519585e-09        2           0
## 20           mcl  34.87283 3.519585e-09        2           0
```

```r
# make formal table of the top 10 words associated with my term
keyness_table <- gt(tstat_key_inside[1:20]) %>%
  tab_header(title = "Table 1. Top 20 words associated with 'contaminant' and its variants")

keyness_table
```

Table 1. Top 20 words associated with 'contaminant' and its variants

feature	chi2	p	n_target	n_reference
cubic	158.72215	0.000000e+00	6	0
sediment	145.06197	0.000000e+00	7	2
site	125.25917	0.000000e+00	29	124
eating	104.62644	0.000000e+00	5	1
fish	80.65981	0.000000e+00	9	16
investigating	75.48850	0.000000e+00	4	1
reuse	73.62987	0.000000e+00	10	23
properties	64.20064	1.110223e-15	11	33
decision	59.26840	1.376677e-14	5	5
yards	58.41687	2.120526e-14	7	13
canal	51.96727	5.643264e-13	4	3
maximum	47.61521	5.186407e-12	3	1
soil	44.61588	2.397393e-11	9	31
putting	37.11807	1.111886e-09	3	2
sustainably	37.11807	1.111886e-09	3	2
245	34.87283	3.519585e-09	2	0
consuming	34.87283	3.519585e-09	2	0
doe	34.87283	3.519585e-09	2	0
lead-	34.87283	3.519585e-09	2	0

mcl	34.87283	3.519585e-09	2	0