Topic 5: Word Relationships

Steven Cognac

2022-04-27

```
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(here)
library(gt)
```

Import data

```
# I've downloaded all the PDFs to a local drive for this
files <- list.files(path = here("05_EPA_txt_analysis"), pattern = "EPA", full.names = T)

ej_reports <- lapply(files, pdf_text)

ej_pdf <- readtext(file = files,
                   docvarsfrom = "filenames",
                   docvarnames = c("type", "year"),
                   sep = "_")
```

```
## Warning in get_docvars_filenames(files, dvsep, docvarnames, docvarsfrom == :
## Fewer docnames supplied than existing docvars - last 1 docvar given generic
## names.
```

```
# 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 year docvar3
##  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
```

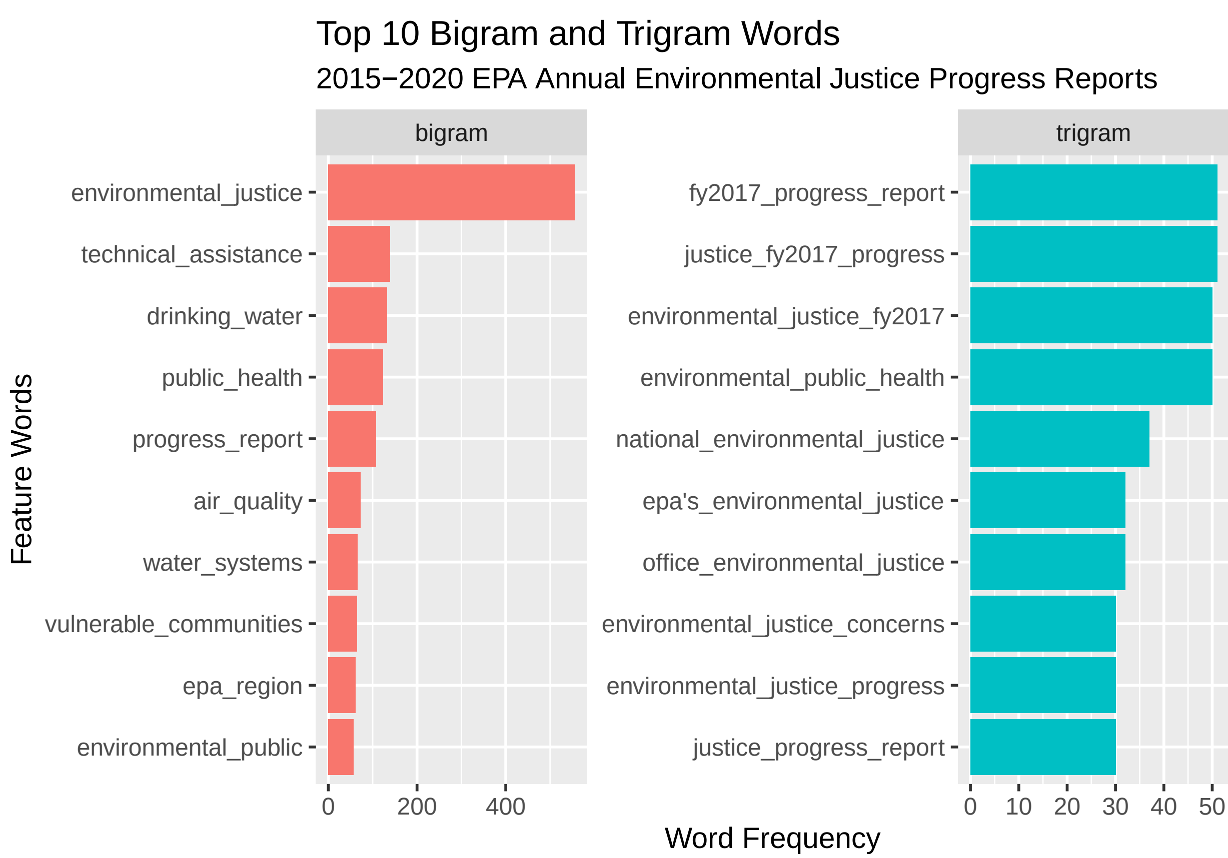
```
# Let's adding some additional, context-specific stop words to stop word lexicon
more_stops <-c("2015",
               "2016",
               "2017",
               "2018",
               "2019",
               "2020",
               "www.epa.gov",
               "https")

add_stops<- tibble(word = c(stop_words$word, more_stops))
stop_vec <- as_vector(add_stops)
```

Part 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?

```
# output is a list of vectors
tokens <- tokens(epa_corp, remove_punct = TRUE)

# some token cleaning
toks1 <- tokens_select(tokens, min_nchar = 3)
toks1 <- tokens_tolower(toks1)
toks1 <- tokens_remove(toks1, pattern = (stop_vec))

# two words is a "bi-gram" "two-word pairs" as a fundamental unit of analysis
toks2 <- tokens_ngrams(toks1, n=2)
toks3 <- tokens_ngrams(toks1, n=3)

# create document feature matrix from tokens
# rows refer to number of occurrences within entire corpus in entire document
dfm2 <- dfm(toks2)
dfm2 <- dfm_remove(dfm2, pattern = c(stop_vec))

dfm3 <- dfm(toks3)
dfm3 <- dfm_remove(dfm3, pattern = c(stop_vec))

# calculate basic statistics
freq_words2 <- textstat_frequency(dfm2, n=20, groups = year)
freq_words2$token <- rep("bigram", 20)
```

```r
freq_words3 <- textstat_frequency(dfm3, n=20, groups = year)
freq_words3$token <- rep("trigram", 20)
```

```r
freq_compare <- rbind(freq_words3 %>% slice_max(frequency, n = 10),
                      freq_words2 %>% slice_max(frequency, n = 10)) %>%
  mutate(order = reorder_within(feature, frequency, token))

freq_compare %>%
  ggplot(aes(y = reorder(feature, desc(frequency)), x = frequency)) +
  geom_col(aes(fill = token)) +
  facet_wrap(~token, scales = "free") +
  labs(y = "Feature Words",
       x = "Word Frequency",
       title = "Top 10 Bigram and Trigram Words",
       subtitle = "2015-2020 EPA Annual Environmental Justice Progress Reports") +
  scale_y_discrete(limits=rev) +
  theme(legend.position="none")
```



Part 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!

```
#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)
```

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

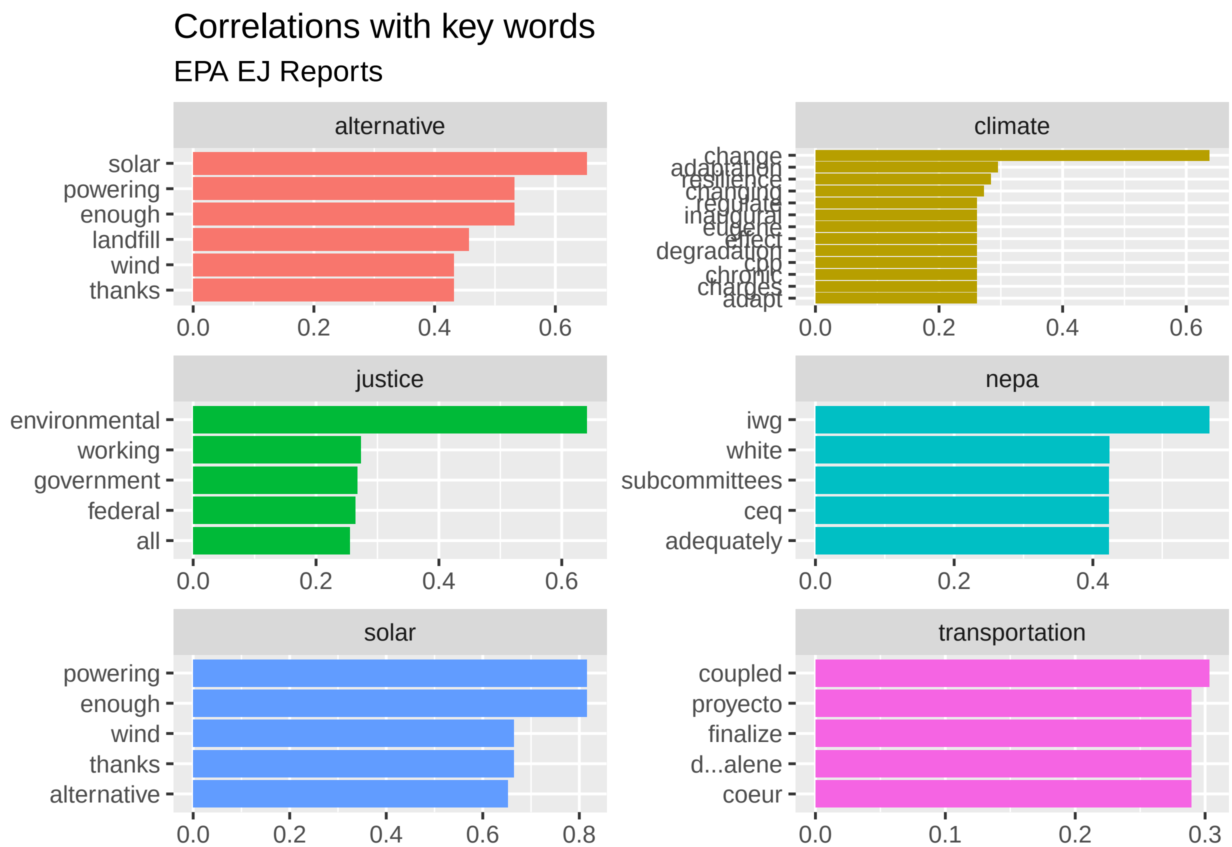
```
# tokenizing to the "paragraph" level
# helps extract meaning from words by attaching a paragraph ID
par_tokens <- unnest_tokens(raw_text, output = paragraphs, input = text, token = "paragraphs")

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

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

```
# create correlations between words
word_cors <- par_words %>%
  add_count(par_id) %>%
  filter(n >= 50) %>%
  select(-n) %>%
  pairwise_cor(word, par_id, sort = TRUE)
```

Word of interest: "energy"

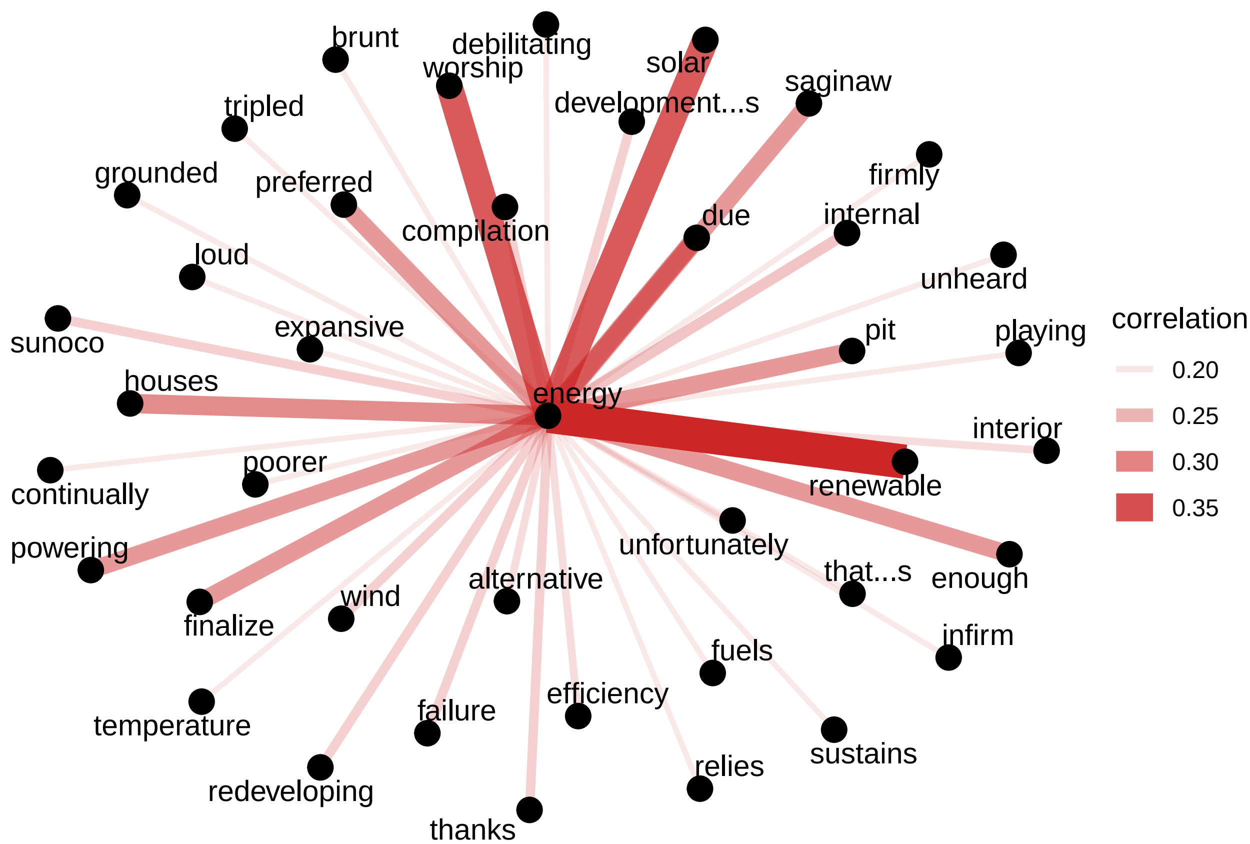
```
# select correlations that contain "energy"
energy_cors <- word_cors %>%
  filter(item1 == "energy")

# filtering words in the context of `item1`
```

```r
word_cors_df <- word_cors %>%
  filter(item1 %in% c("climate",
                      "transportation",
                      "nepa",
                      "alternative",
                      "solar",
                      "justice")) %>%
  group_by(item1) %>%
  slice_max(correlation, n=5) %>%
  ungroup() %>%
  mutate(item1 = as.factor(item1),
  name = reorder_within(item2, correlation, item1))

ggplot(word_cors_df, 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
energy_cors <- word_cors %>%
  filter(item1 == "energy") %>%
```

```r
  mutate(n = 1:n())

energy_cors  %>%
  filter(n <= 40) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation), edge_colour = "firebrick3") +
  geom_node_point(size = 4) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()
```

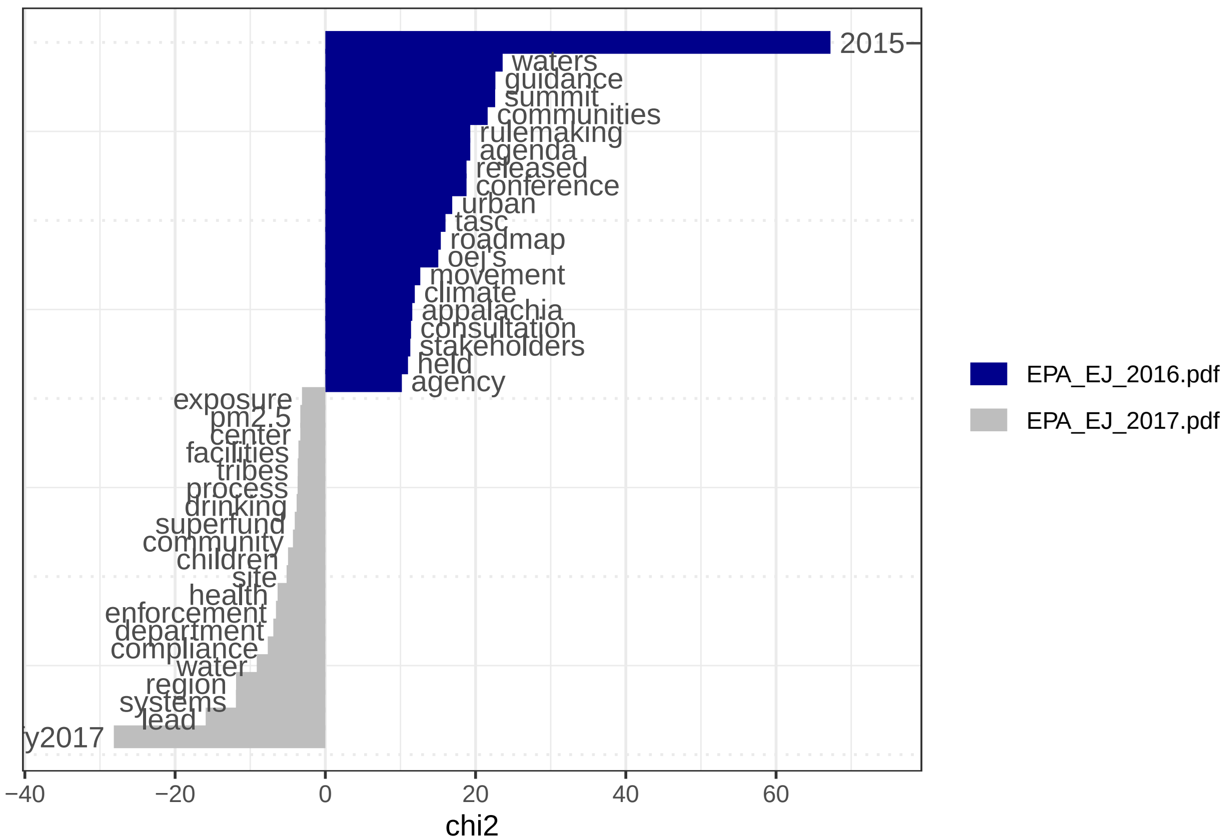


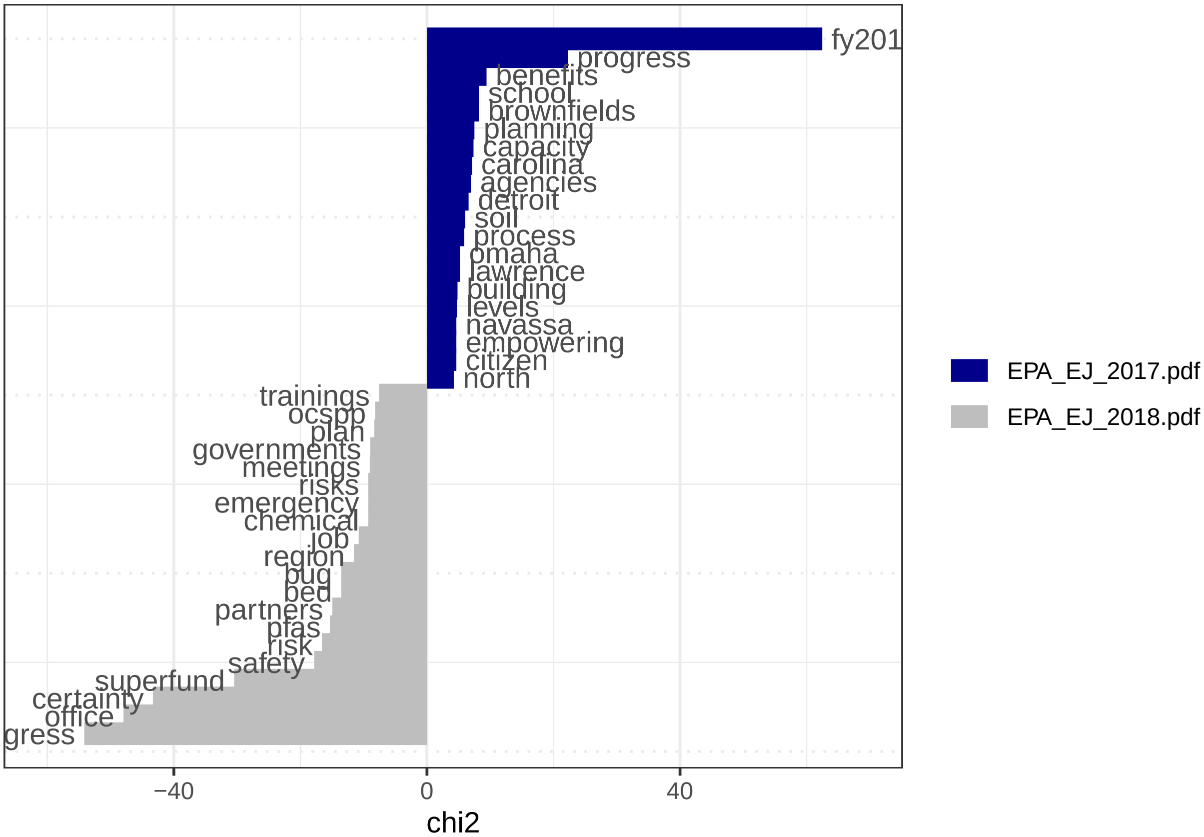
Not surprisingly, we see "renewable energy" and "solar energy" as our most highly correlated word pairs. One surprising correlation is "worship" and energy.

Part 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.

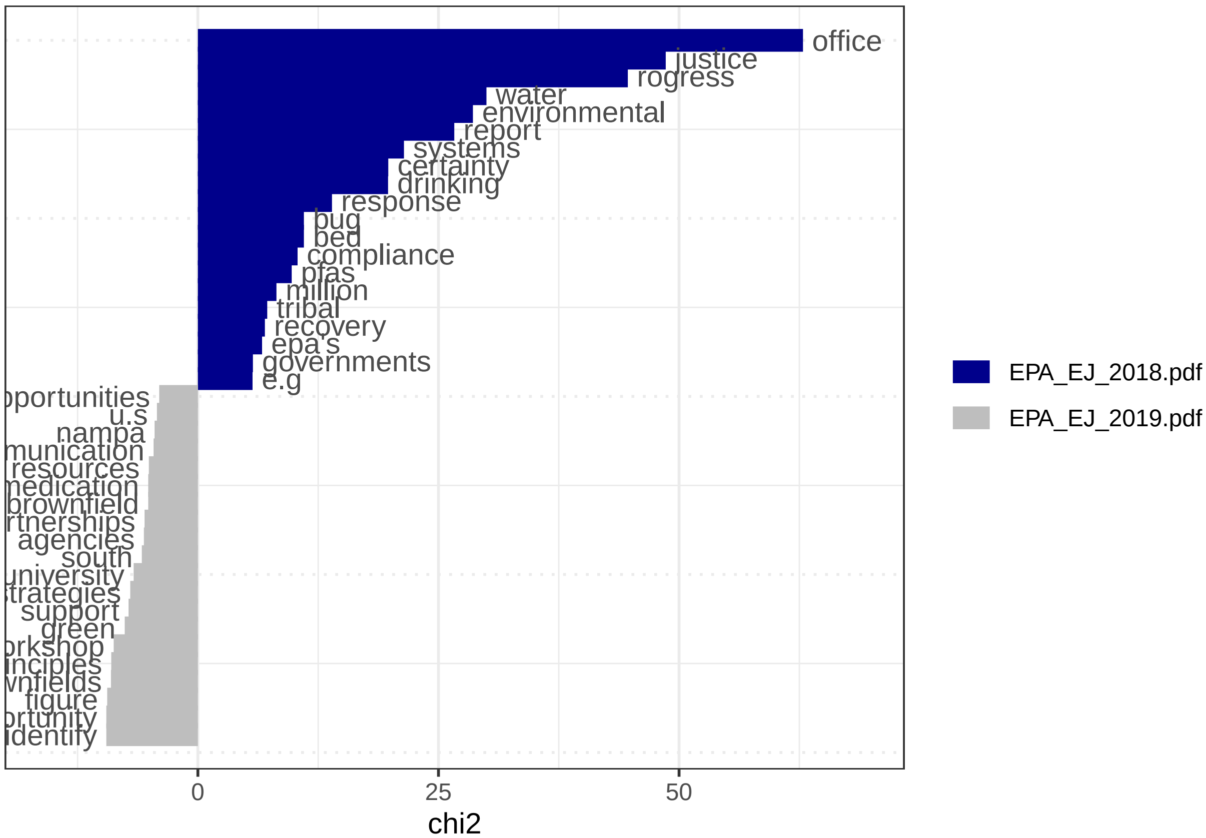
```r
keyness_fn <- function(){
  for (i in 2:4) {
```

```
    reports <- epa_corp[i:(i+1)]
    reps_tok <- tokens(reports, remove_punct = TRUE)
    reps <- tokens_select(reps_tok, min_nchar = 3) %>%
      tokens_tolower() %>%
      tokens_remove(pattern = (stop_vec))
    dfm <- dfm(reps)
    keyness <- textstat_keyness(dfm, target = 1)
    print(textplot_keyness(keyness))
  }
}
keyness_fn()
```





fy201
progress
benefits
school
brownfields
planning
capacity
carolina
agencies
detroit
soil
process
omaha
lawrence
building
levels
navassa
empowering
citizen
north
trainings
ocspp
plan
governments
meetings
risks
emergency
chemical
job
region
bug
bed
partners
pfas
risk
safety
superfund
certainty
office
gress
−40
0
40
chi2
EPA_EJ_2017.pdf
EPA_EJ_2018.pdf



Part 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 to 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

```
term <- c("water")

# words within a 10-word window of "water"
toks_inside <- tokens_keep(tokens, pattern = term, window = 10) %>%
  tokens_remove(pattern = term) %>% # remove the keywords
  tokens_tolower() %>%
  tokens_remove(pattern = c(stop_vec))

dfm_inside <- dfm(toks_inside)

# words outside a 10-word window of "water"
toks_outside <- tokens_remove(tokens, pattern = term, window = 10) %>%
```

```
  tokens_tolower() %>%
  tokens_remove(pattern = c(stop_vec))

dfm_outside <- dfm(toks_outside)

# combine datasets
dfm_inside_outside <- rbind(dfm_inside, dfm_outside)

# compute associations with keywords
keyness2 <- textstat_keyness(dfm_inside_outside, target = seq_along(ndoc(dfm_inside)))

gt_keyness2 <- gt(keyness2[1:20])
gt_keyness2 %>%
  tab_header(title = "Top 20 Word Associations with 'water'")
```

Top 20 Word Associations with 'water'

feature	chi2	p	n_target	n_reference
130	99.42090	0.000000e+00	1	0
600,000	99.42090	0.000000e+00	1	0
lay-	99.42090	0.000000e+00	1	0
sole	99.42090	0.000000e+00	1	0
stable	99.42090	0.000000e+00	1	0
treaties	99.42090	0.000000e+00	1	0
source	53.53376	2.542411e-13	2	14
flooding	50.21352	1.378897e-12	2	15
improved	50.21352	1.378897e-12	2	15
happen	49.21383	2.295275e-12	1	1
merrimac	49.21383	2.295275e-12	1	1
organizational	49.21383	2.295275e-12	1	1
sanitation	49.21383	2.295275e-12	1	1
clean	49.19614	2.316036e-12	4	84
farmworker	32.47980	1.204378e-08	1	2
community-led	24.11403	9.079653e-07	1	3
urban	23.14454	1.502692e-06	3	87
management	20.37933	6.351223e-06	3	97
company	19.09556	1.243341e-05	1	4
heart	19.09556	1.243341e-05	1	4