# Topic 5: Word Relationships

Peter Menzies

## Read in data

```
setwd('~/MEDS/Spring_Quarter/EDS_231_Text/dat/')
files <- list.files(pattern = "pdf$")

ej_reports <- lapply(files, pdf_text)

ej_pdf <- readtext("*.pdf", docvarsfrom = "filenames",
                   docvarnames = c("type", "year"),
                   sep = "_")

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
##  EPAEJ_2015.pdf  2136   8944       263 EPAEJ 2015
##  EPAEJ_2016.pdf  1599   7965       176 EPAEJ 2016
##  EPAEJ_2017.pdf  3974  30562       653 EPAEJ 2017
##  EPAEJ_2018.pdf  2788  16724       447 EPAEJ 2018
##  EPAEJ_2019.pdf  3775  22644       672 EPAEJ 2019
##  EPAEJ_2020.pdf  4493  30523       987 EPAEJ 2020
```

```
# Additional stop words
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)
```

### Cleaning and tokenizing

```
raw_text <- tidy(epa_corp)

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")
```

# 1. Bigrams vs. Trigrams

```
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)
```

### 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)

head(freq_words2, 10)
```

```
##                   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
## 7           water_systems        66    7       6   all bigram
## 8  vulnerable_communities        65    8       6   all bigram
## 9              epa_region        62    9       5   all bigram
## 10   environmental_public        57   10       6   all bigram
```

### Trigrams

```
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)

head(freq_words3, 10)
```

```
##                           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
## 7     epa's_environmental_justice        31    7       6   all trigram
```

```
## 8  environmental_justice_progress        30    8       4   all trigram
## 9          justice_progress_report        30    8       4   all trigram
## 10 environmental_justice_concerns        30    8       5   all trigram
```

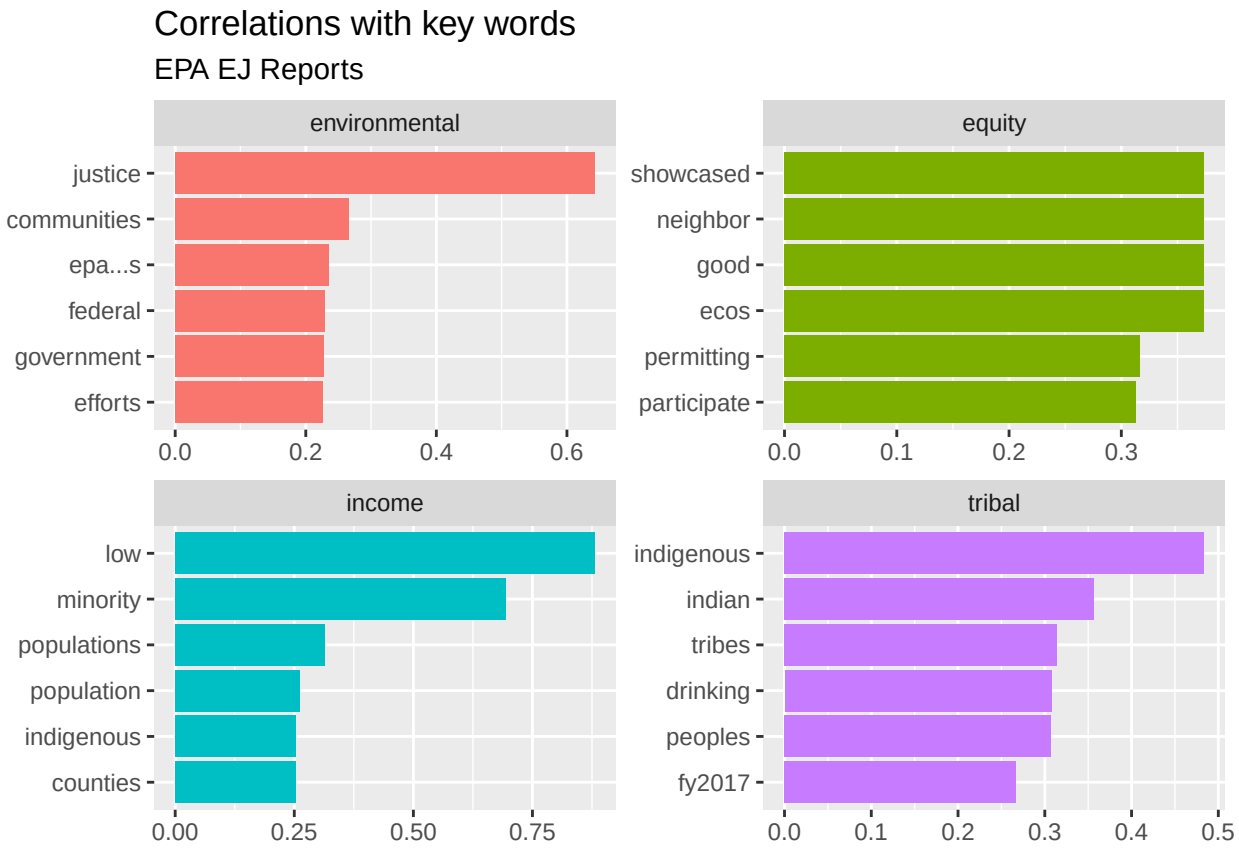
**Which is more informative?**

In this instance, bigrams seem to be more useful. The trigrams have a tendency to pick up on different variations of the same thing or concept. The bigrams appear to capture a wider range of important phrases.

## 2. Correlation table and network for "tribal"

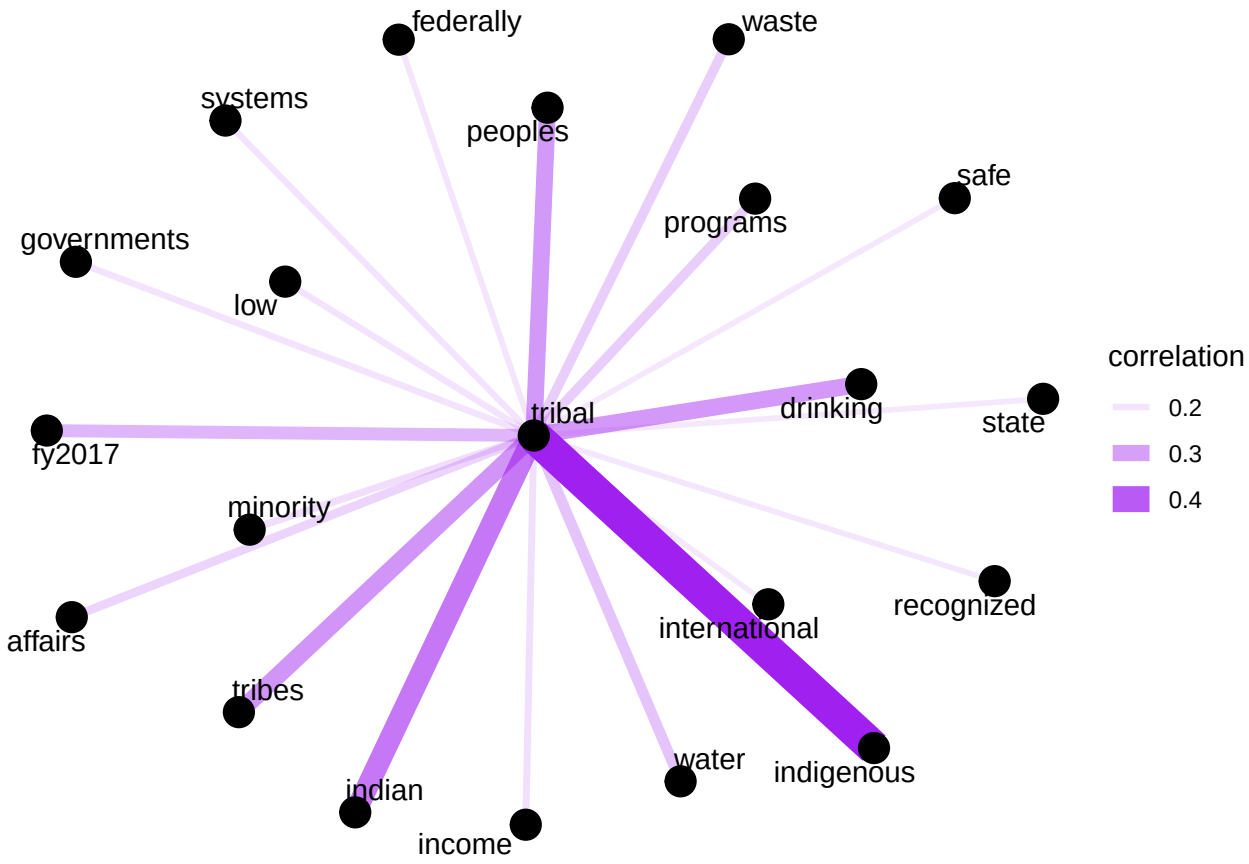
**Table**

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


word_cors %>%
filter(item1 %in% c("environmental", "tribal", "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")
```




**Network**

```
tribal_cors <- word_cors %>%
  filter(item1 == "tribal") %>%
  mutate(n = 1:n())

tribal_cors  %>%
  filter(n <= 20) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation), edge_colour = "purple") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()
```




## 3. Keyness function

```
plot_keyness <- function(doc1, doc2) {

  logical_vec <- dfm@docvars[["docname_"]] %in% c(doc1, doc2)
  dfm_sub <- dfm_subset(dfm, logical_vec)

  keyness <- textstat_keyness(dfm_sub, target = doc1)
  plot <- textplot_keyness(keyness)

  return(plot)
}
```

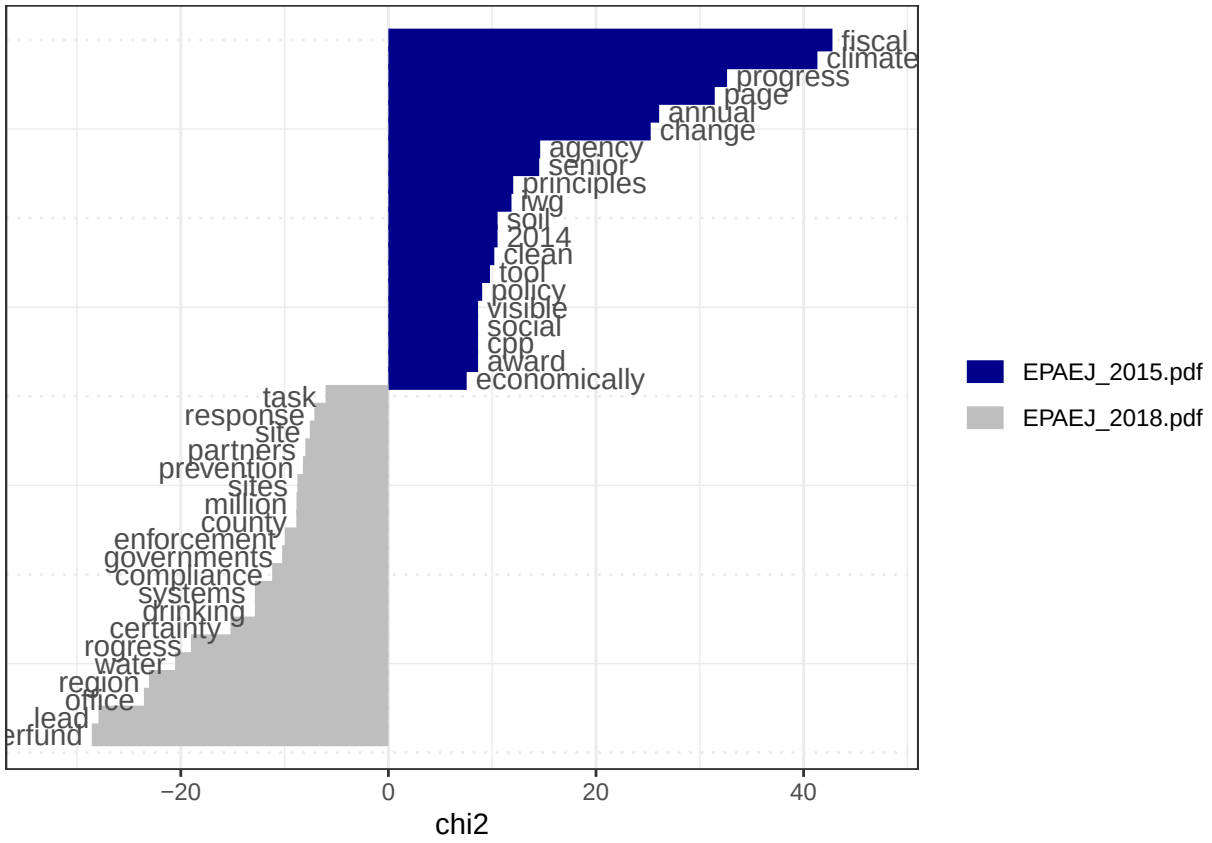
**Bonus function with option for ngram**

```
plot_keyness_ngram <- function(doc1, doc2, ngram = 1) {

  docs <- ej_pdf %>%
    filter(doc_id %in% c(doc1, doc2))

  corpus <- corpus(docs, text_field = "text")
```

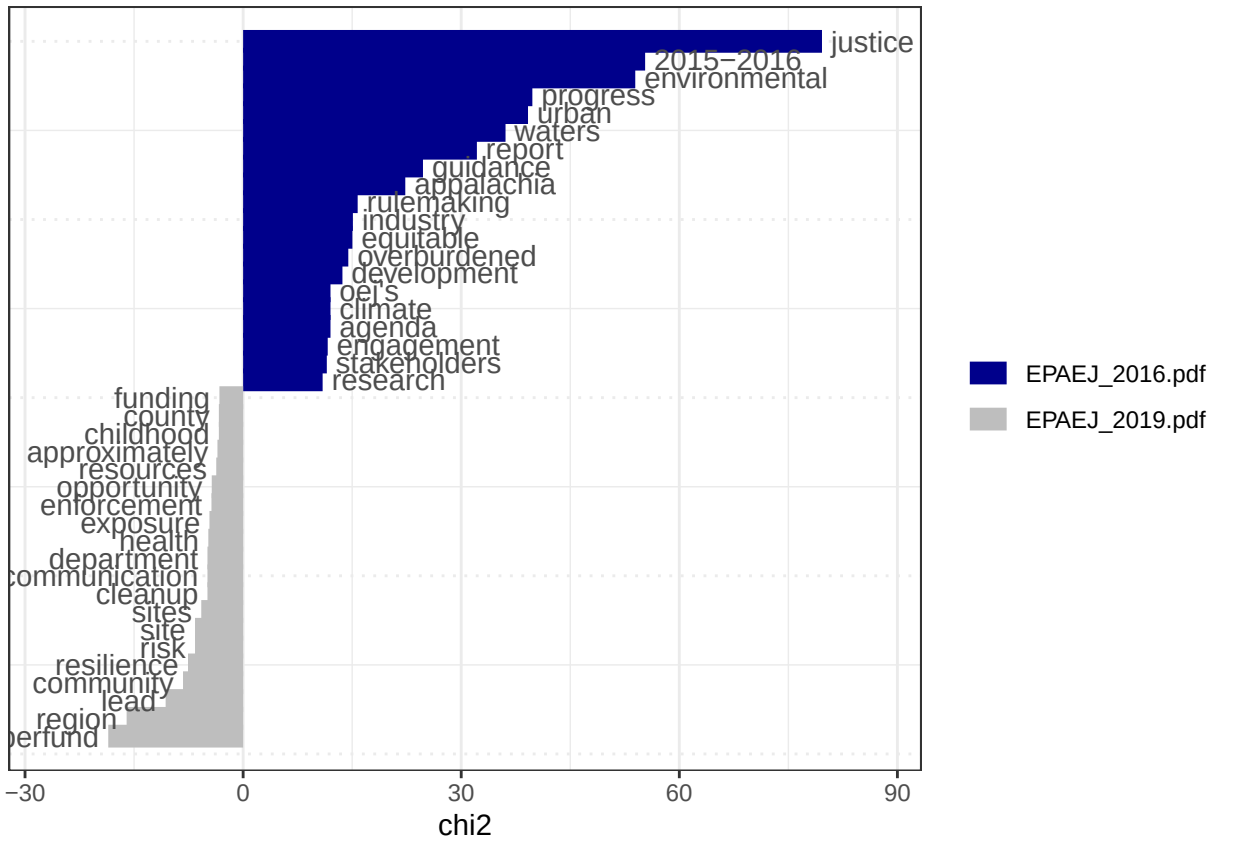
```
  tokens <- tokens(corpus, remove_punct = TRUE)
  tokens <- tokens_select(tokens, min_nchar = 3)
  tokens <- tokens_ngrams(tokens, n = ngram)
  tokens <- tokens_tolower(tokens)
  tokens <- tokens_remove(tokens, pattern = (stop_vec))

  dfm <- dfm(tokens)

  keyness <- textstat_keyness(dfm, target = doc1)
  plot <- textplot_keyness(keyness)

  return(plot)
}
```

**Functions in action**

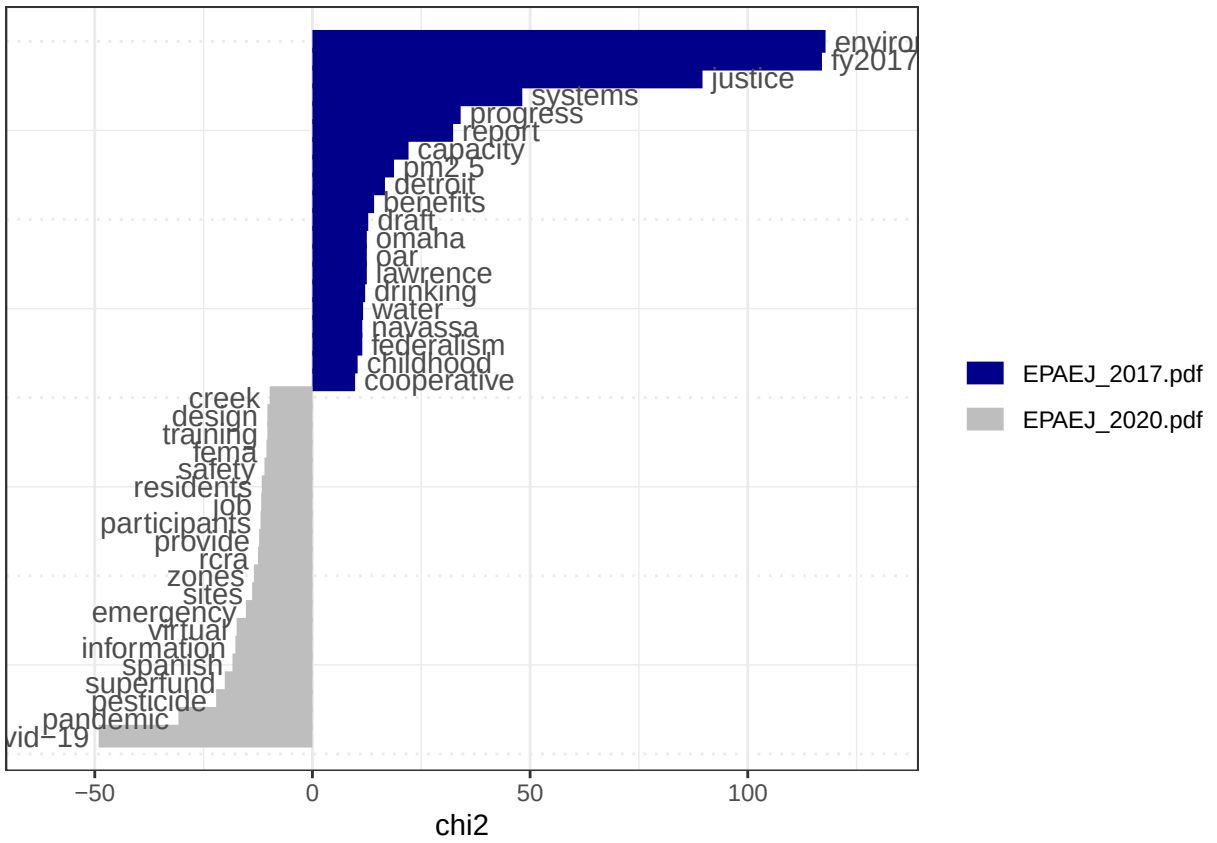
```
plot_keyness("EPAEJ_2015.pdf", "EPAEJ_2018.pdf")
```



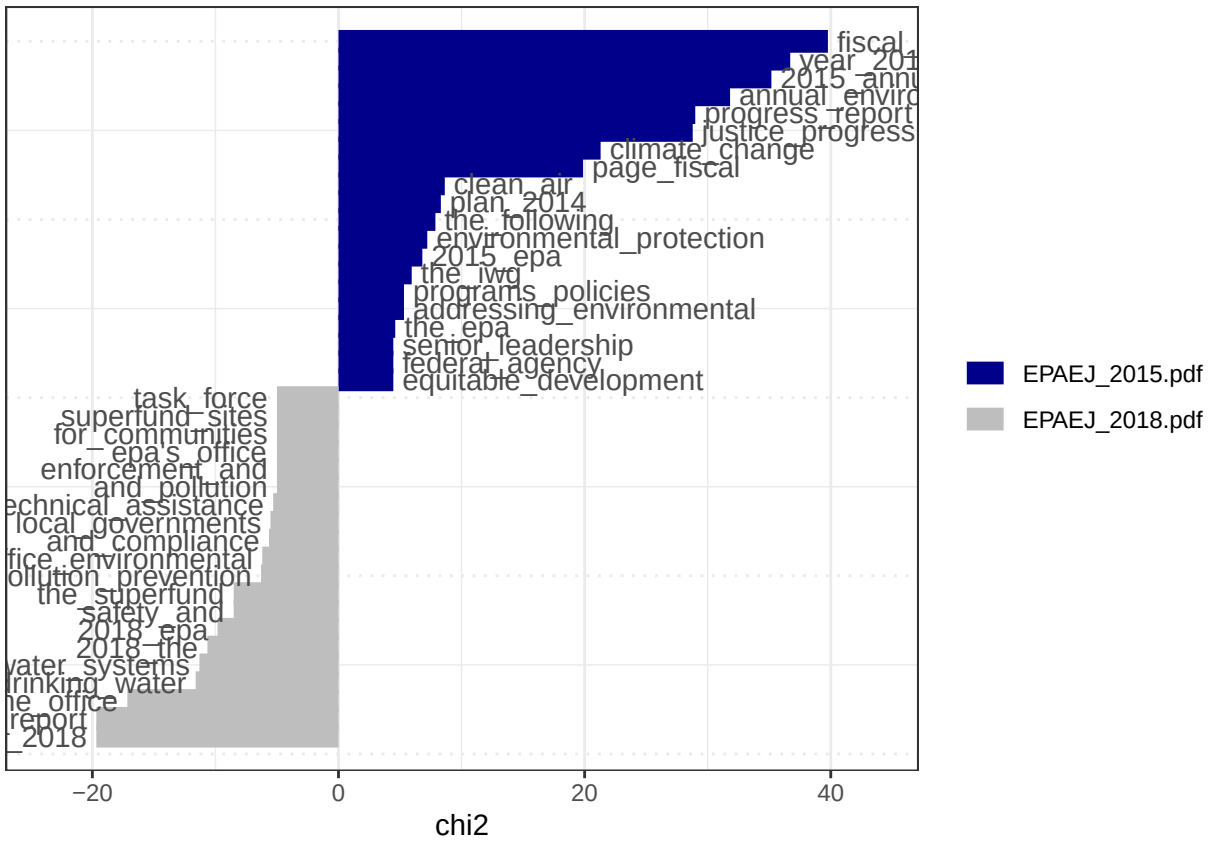

```
plot_keyness("EPAEJ_2016.pdf", "EPAEJ_2019.pdf")
```




```
plot_keyness("EPAEJ_2017.pdf", "EPAEJ_2020.pdf")
```




```
plot_keyness_ngram("EPAEJ_2015.pdf", "EPAEJ_2018.pdf", ngram = 2)
```



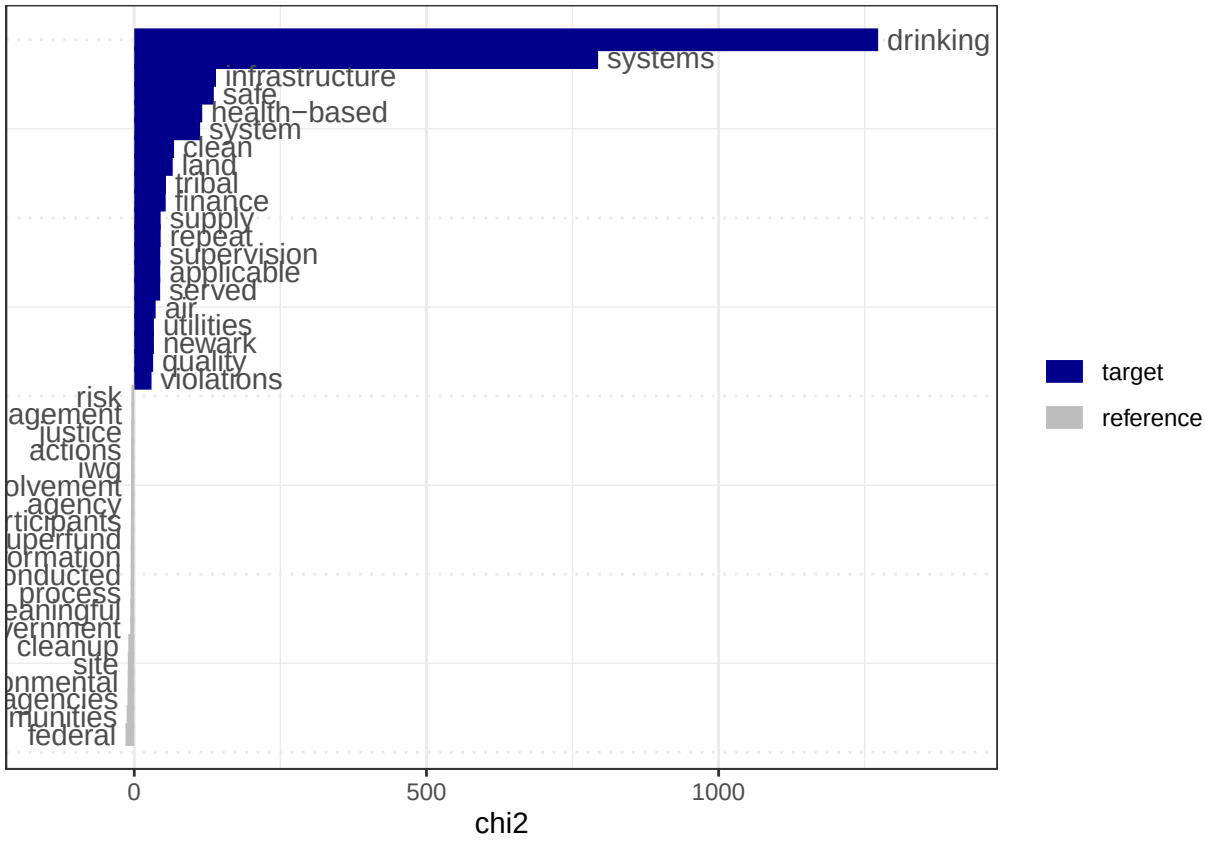

# 4. Word of interest

### Tokenize and subset

```
word <- "water"

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

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

### Convert to dfm and calculate keyness

```
dfm_inside <- dfm(toks_inside)
dfm_outside <- dfm(toks_outside)

doc_indices <- seq_len(ndoc(dfm_inside))

keyness_inside <- textstat_keyness(rbind(dfm_inside, dfm_outside),
                                   target = doc_indices)

textplot_keyness(keyness_inside)
```




**Target and reference**

The target here is all the documents contained in `dfm_inside`, i.e. the dfm containing only the words related to "water", and not water itself. The reference is all the documents contained in `dfm_outside`—the dfm containing all the words outside of the "water" windows.