Assignment 4: Word Relationships

Marie Rivers

4/27/2022

```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(here)
```

```r
files <- list.files(path = here("data/EJ"),
                    pattern = "pdf$", full.names = TRUE)

ej_reports <- lapply(files, pdf_text)

ej_pdf <- readtext(file = here("data/EJ", "*.pdf"),
                   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) %>%
  knitr::kable(caption = "Summary of EPA Reprot Corpus")
```

Table 1: Summary of EPA Reprot Corpus

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

Text	Types	Tokens	Sentences	type	subj	year
EPA_EJ_2019.pdf	3773	22648	672	EPA	EJ	2019
EPA_EJ_2020.pdf	4493	30523	987	EPA	EJ	2020

```
# Add 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)
```

Create different data objects that will be used 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)
```

```
## 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")
```

```
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, 15) %>%
  knitr::kable(caption = "Subset of Top 5 Words")
```

Table 2: Subset of Top 5 Words

feature	frequency	rank	docfreq	group
environmental	127	1	1	2015
communities	99	2	1	2015
epa	92	3	1	2015
justice	84	4	1	2015
community	47	5	1	2015
environmental	109	1	1	2016
communities	85	2	1	2016
justice	71	3	1	2016
epa	48	4	1	2016
federal	31	5	1	2016
environmental	178	1	1	2017
communities	135	2	1	2017
epa	130	3	1	2017
justice	109	4	1	2017
community	105	5	1	2017

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?

```r
# bigrams
toks2 <- tokens_ngrams(toks1, n=2)
dfm2 <- dfm(toks2) # document feature matrix
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")

bigrams <- freq_words2 %>%
  knitr::kable(caption = "Bigrams")
bigrams
```

Table 3: Bigrams

feature	frequency	rank	docfreq	group	token
environmental_justice	556	1	6	all	bigram
technical_assistance	139	2	6	all	bigram
drinking_water	133	3	6	all	bigram
public_health	123	4	6	all	bigram
progress_report	108	5	6	all	bigram
air_quality	73	6	6	all	bigram
water_systems	66	7	6	all	bigram
vulnerable_communities	65	8	6	all	bigram
epa_region	62	9	5	all	bigram
environmental_public	57	10	6	all	bigram

feature	frequency	rank	docfreq	group	token
federal_agencies	56	11	6	all	bigram
national_environmental	51	12	6	all	bigram
justice_fy2017	51	12	1	all	bigram
fy2017_progress	51	12	1	all	bigram
superfund_sites	48	15	4	all	bigram
indigenous_peoples	46	16	6	all	bigram
civil_rights	46	16	5	all	bigram
local_governments	45	18	6	all	bigram
urban_waters	44	19	6	all	bigram
overburdened_communities	43	20	6	all	bigram

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

trigrams <- freq_words3 %>%
  knitr::kable(caption = "Trigrams")
trigrams
```

Table 4: Trigrams

feature	frequency	rank	docfreq	group	token
justice_fy2017_progress	51	1	1	all	trigram
fy2017_progress_report	51	1	1	all	trigram
environmental_public_health	50	3	6	all	trigram
environmental_justice_fy2017	50	3	1	all	trigram
national_environmental_justice	37	5	6	all	trigram
office_environmental_justice	32	6	6	all	trigram
epa's_environmental_justice	32	6	6	all	trigram
environmental_justice_progress	30	8	4	all	trigram
justice_progress_report	30	8	4	all	trigram
environmental_justice_concerns	30	8	5	all	trigram
drinking_water_systems	29	11	5	all	trigram
annual_environmental_justice	27	12	5	all	trigram
environmental_justice_advisory	27	12	6	all	trigram
fiscal_annual_environmental	25	14	3	all	trigram
justice_advisory_council	24	15	6	all	trigram
environmental_justice_grants	22	16	5	all	trigram
technical_assistance_communities	20	17	6	all	trigram
communities_environmental_justice	20	17	5	all	trigram
safe_drinking_water	19	19	5	all	trigram
technical_assistance_services	19	19	5	all	trigram

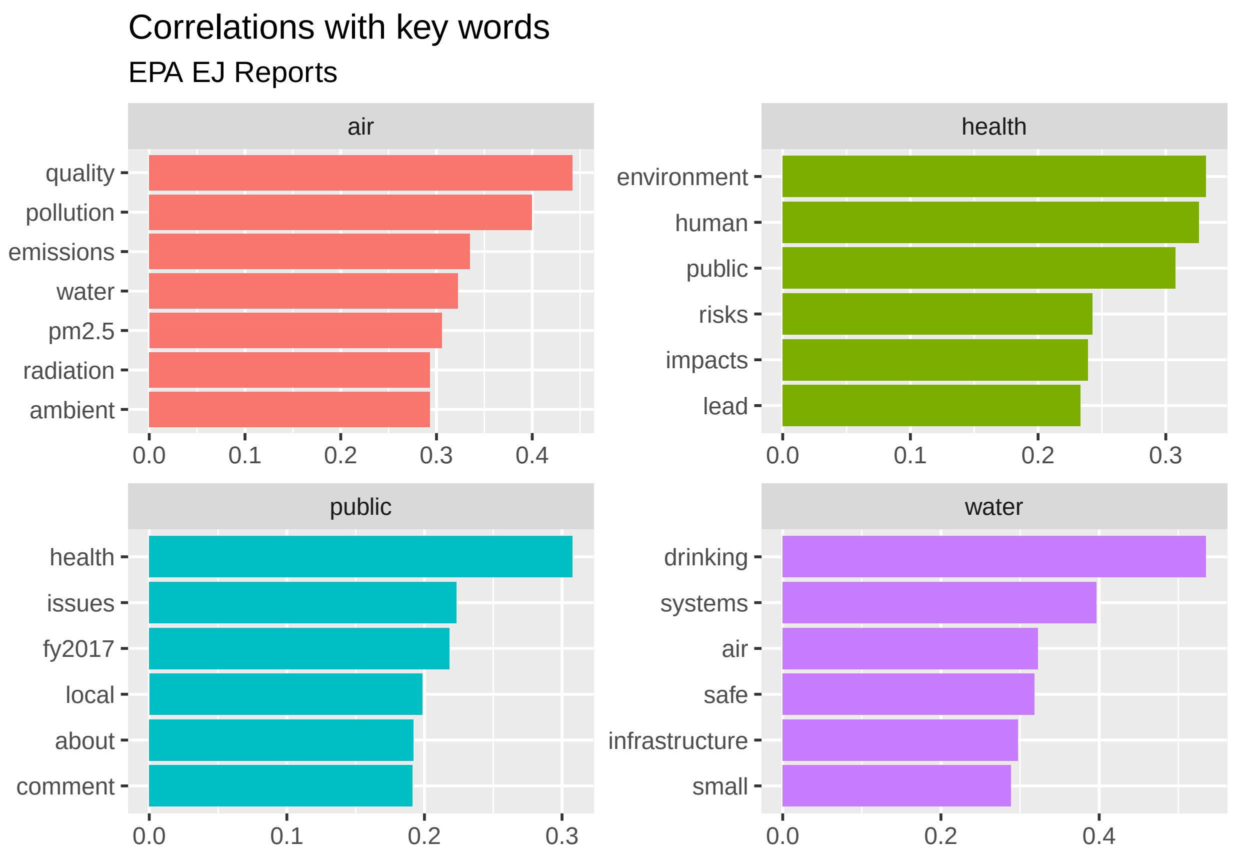
The three most frequent trigrams are justice_fy2017_progress, fy2017_progress_report, and environmental_public_health. The three most frequent bigrams are environmental_justice, technical_assistance, and drinking_water. The words 'environmental' and 'justice' appear several times in both the top bigrams and top trigrams. The bigrams seem more informative than the trigrams because there is more variety in the terms. Most of the trigrams are variations of 'environmental justice'. Also, the top trigrams are likely all part of the same phrase 'environmental justice fy2017 progress report'.

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!

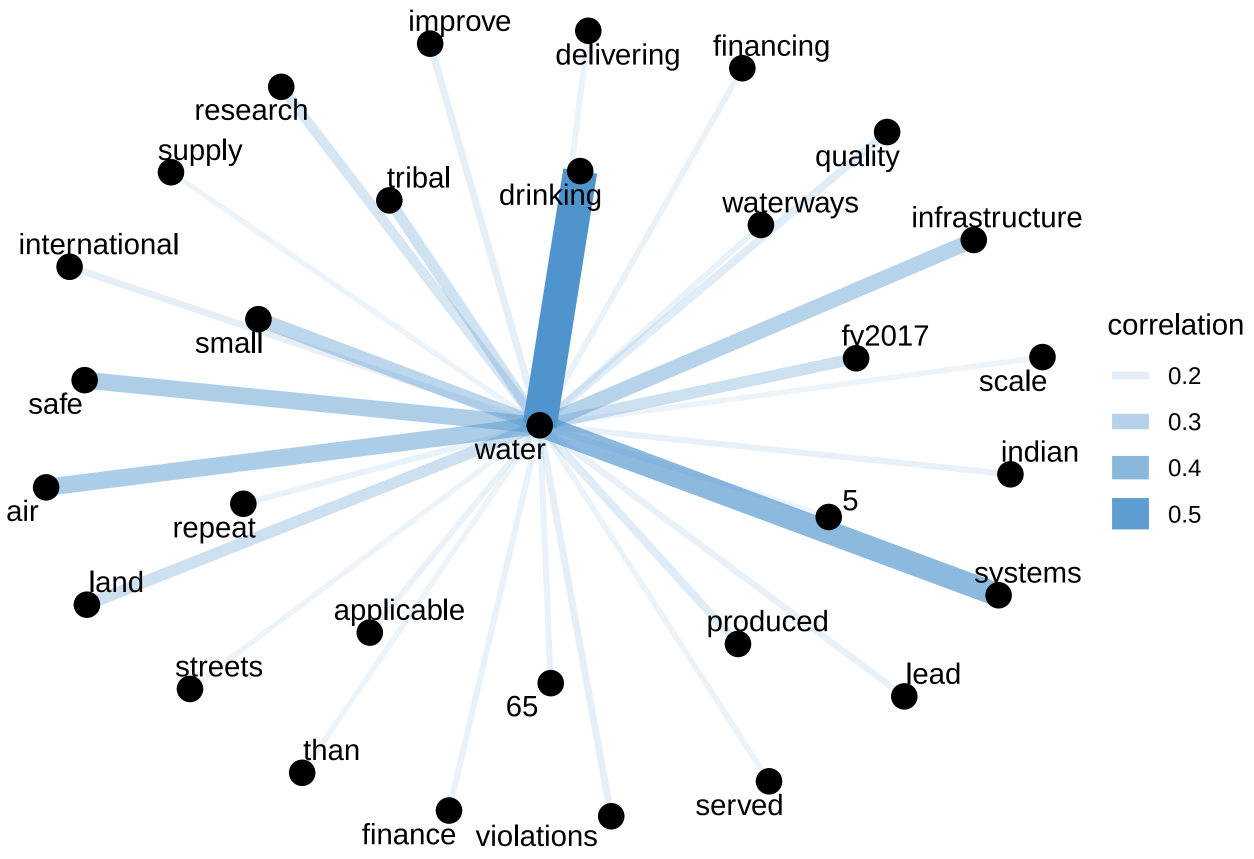
```
word_cors <- par_words %>%
  add_count(par_id) %>%
  filter(n >= 50) %>%
  select(-n) %>%
  pairwise_cor(word, par_id, sort = TRUE)
```

```
water_cors <- word_cors %>%
  filter(item1 == "water")

  word_cors %>%
  filter(item1 %in% c("water", "air", "health", "public"))%>%
  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
```



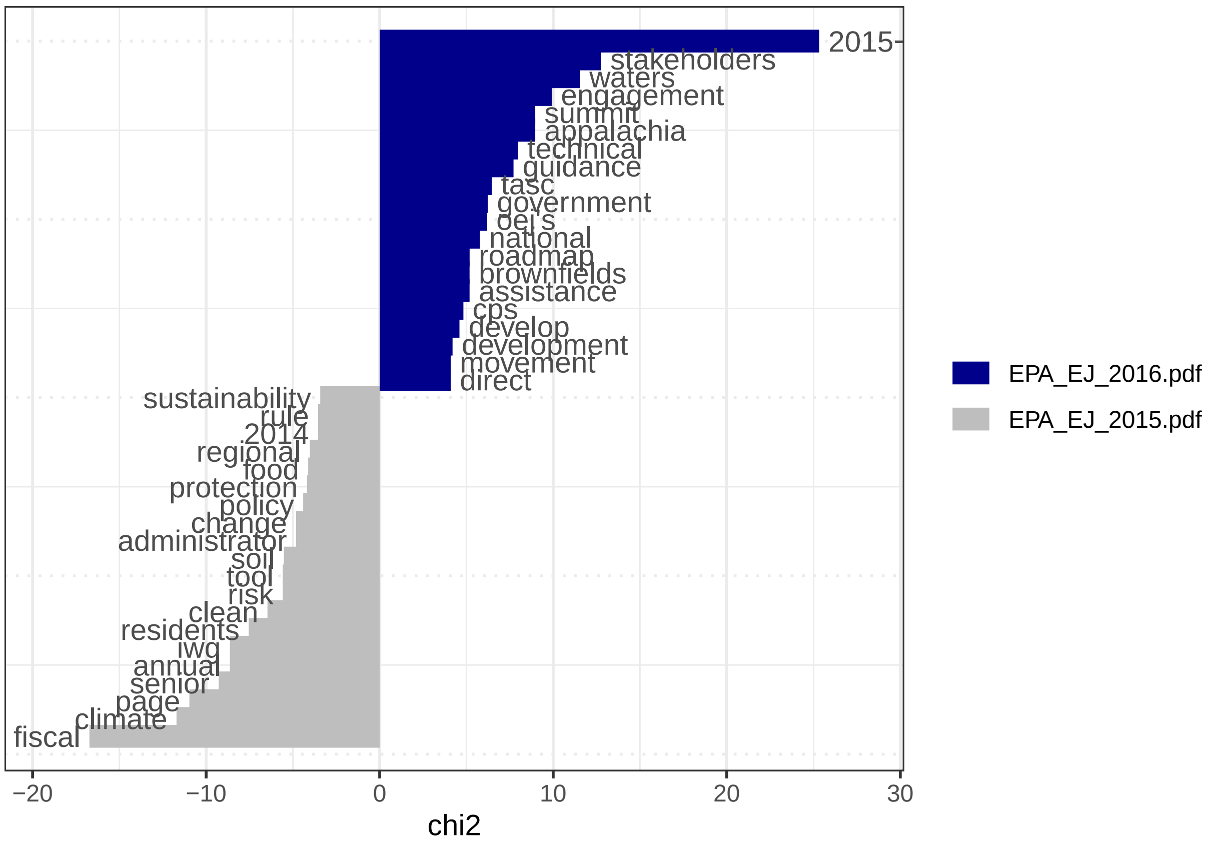
```r
#let's zoom in on just one of our key terms
water_cors <- word_cors %>%
  filter(item1 == "water") %>%
  mutate(n = 1:n())

water_cors  %>%
  filter(n <= 30) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation), edge_colour = "steelblue3") +
  geom_node_point(size = 4) +
  geom_node_text(aes(label = name), repel = TRUE,
               point.padding = unit(0.2, "lines")) +
  theme_void()
```



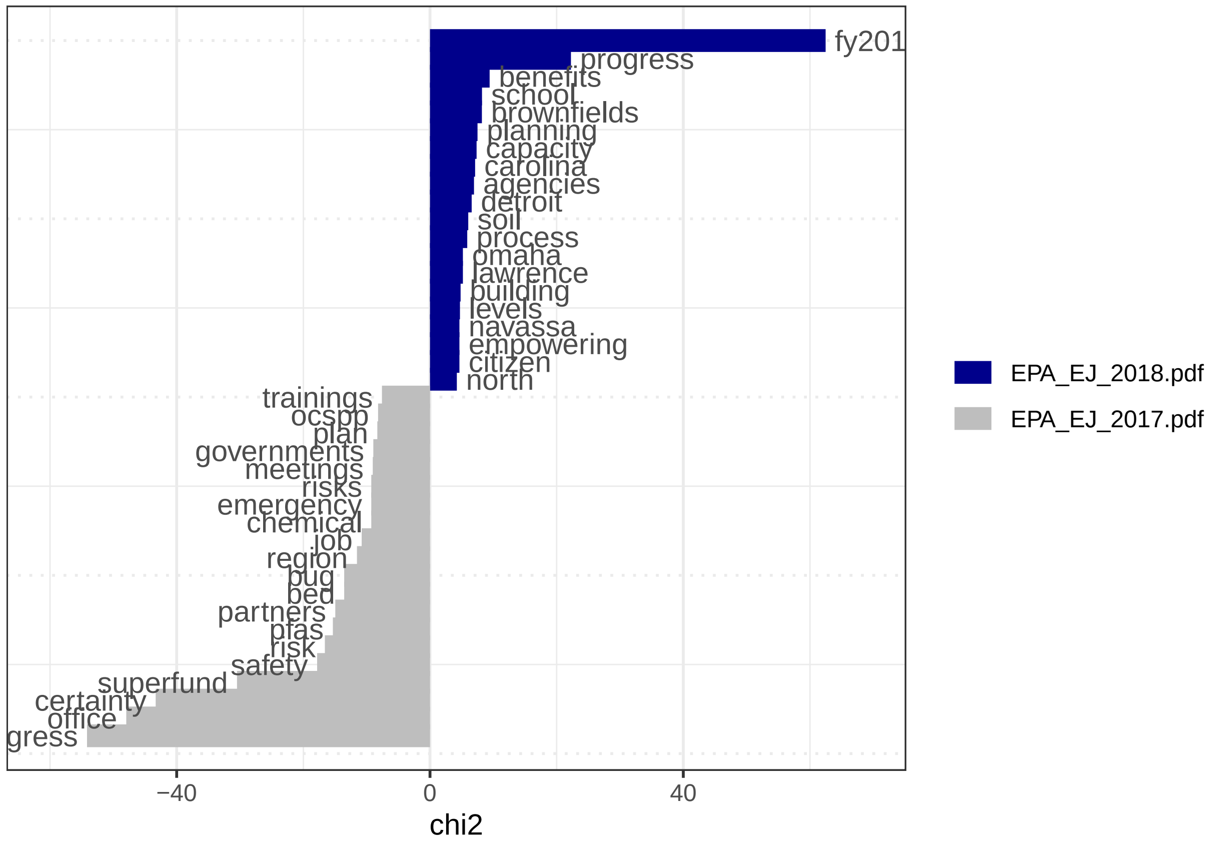
3. Write a function that allows you to conduct a keyness analysis to compare two individual EPA reports (hint: that means target and reference need to both be individual reports). Run the function on 3 pairs of reports, generating 3 keyness plots.

```r
keyness_function <- function(reference_report_year, target_report_year) {
  files <- list.files(path = here("data/EJ"),
                   pattern = "pdf$", full.names = TRUE)
  ej_reports <- lapply(files, pdf_text)
  ej_pdf <- readtext(file = here("data/EJ", "*.pdf"),
                  docvarsfrom = "filenames",
                  docvarnames = c("type", "subj", "year"),
                  sep = "_")
  epa_corp <- corpus(x = ej_pdf, text_field = "text" )
  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_function_plot <- dfm %>%
    dfm_subset(year %in% c(reference_report_year, target_report_year)) %>%
    textstat_keyness(target = paste0("EPA_EJ_", target_report_year, ".pdf")) %>%
    textplot_keyness()
  keyness_function_plot
}
```

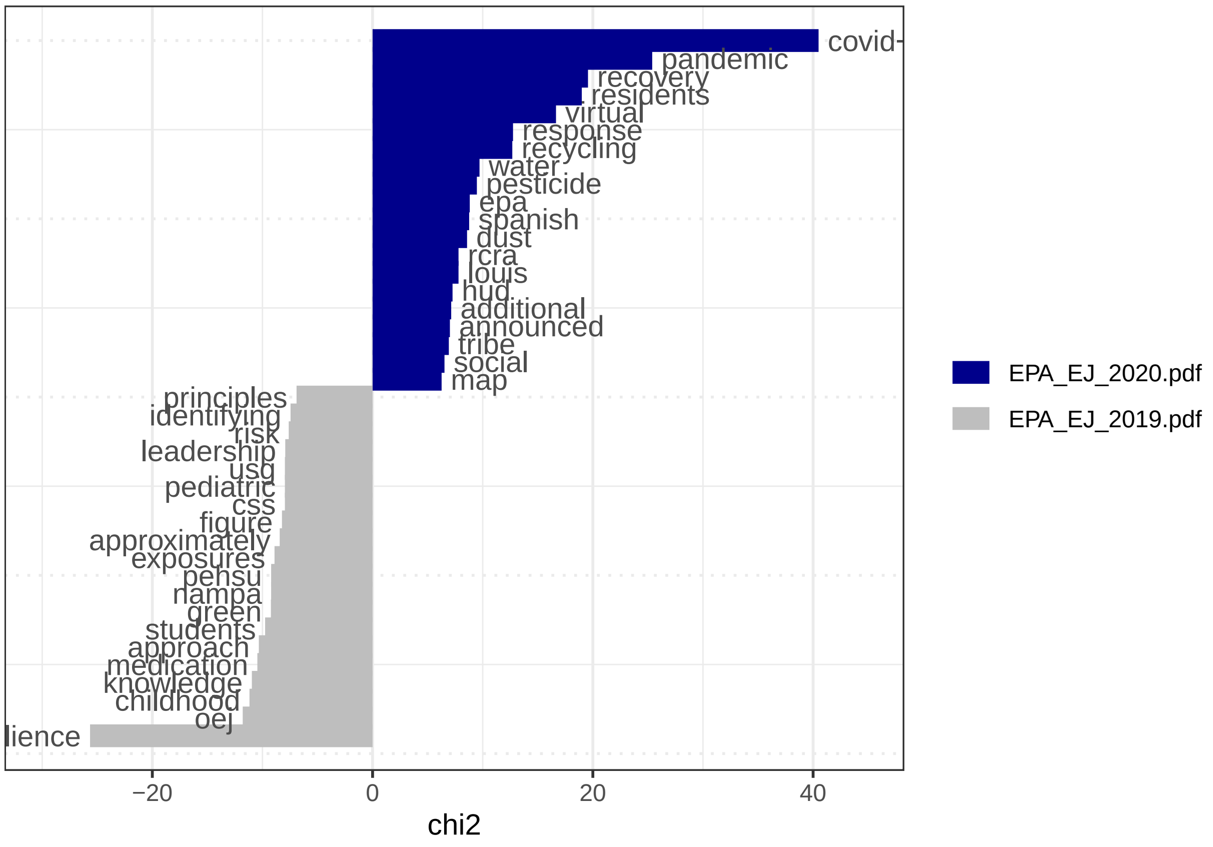
```
# 2015 vs. 2016
keyness_function(reference_report_year = 2015, target_report_year = 2016)
```



```
# 2017 vs. 2018
keyness_function(reference_report_year = 2017, target_report_year = 2018)
```



```
# 2019 vs. 2020
keyness_function(reference_report_year = 2019, target_report_year = 2020)
```



```
# hey covid-2019 pandemic
```

Note: EPA_EJ_2015.pdf is report for FY 2015 (published Sept. 2016)

EPA_EJ_2016.pdf is report for FY 2015-2016

EPA_EJ_2017.pdf is report for FY 2018

EPA_EJ_2018.pdf is report for FY 2017

EPA_EJ_2019.pdf is report for FY 2019

EPA_EJ_2020.pdf is report for FY 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