Homework 4: Word Relationships

Paloma Cartwright

2022-05-01

Import EPA EJ Data

```
files <- list.files(path = here::here("dat"),
                    pattern = "EPA", full.names = T)
ej_reports <- lapply(files, pdf_text)
ej_pdf <- readtext(file = files,
                   docvarsfrom = "filenames",
                   docvarnames = c("type", "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 year
##   EPAEJ_2015.pdf  2136   8944       263 EPAEJ 2015
##   EPAEJ_2016.pdf  1599   7965       176 EPAEJ 2016
##   EPAEJ_2017.pdf  2774  16658       447 EPAEJ 2017
##   EPAEJ_2018.pdf  3973  30564       653 EPAEJ 2018
##   EPAEJ_2019.pdf  3773  22648       672 EPAEJ 2019
##   EPAEJ_2020.pdf  4493  30523       987 EPAEJ 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", "fy2017")
add_stops<- tibble(word = c(stop_words$word, more_stops))
stop_vec <- as_vector(add_stops)
```

Tidy the Data

```
#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")
```

Pull out info with 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)
```

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?

```
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")# two word pair
freq_words2 <- data.frame(freq_words2) %>% select(feature, frequency, token)

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") # two word pair

freq_words3 <- data.frame(freq_words3) %>% select(feature, frequency, token)

freq <- cbind(freq_words3, freq_words2) %>%
  kable()

freq
```

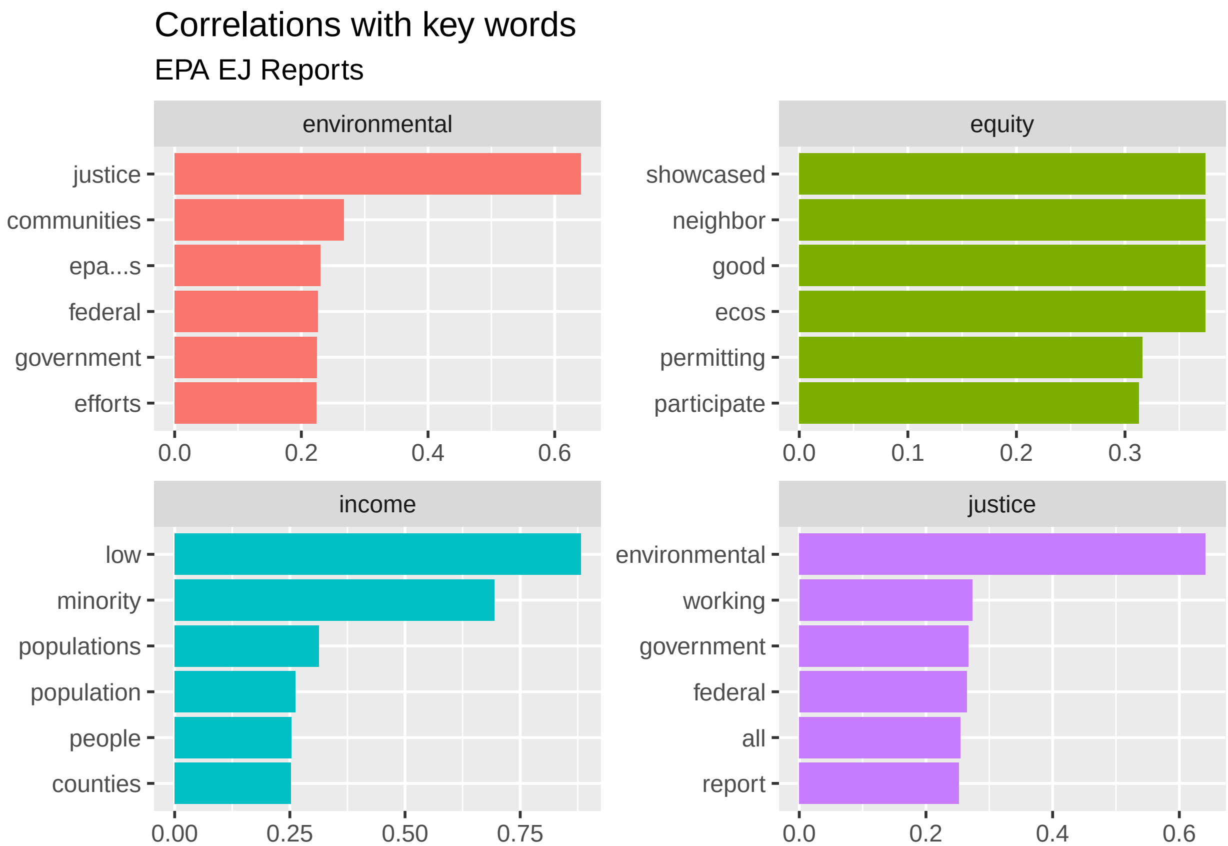
feature	frequency	token	feature	frequency	token
justice_progress_report	81	trigram	environmental_justice	556	bigram
environmental_justice_progress	80	trigram	technical_assistance	139	bigram
environmental_public_health	50	trigram	drinking_water	133	bigram
national_environmental_justice	37	trigram	public_health	123	bigram
office_environmental_justice	32	trigram	progress_report	108	bigram
epa's_environmental_justice	32	trigram	justice_progress	81	bigram
environmental_justice_concerns	30	trigram	air_quality	73	bigram
drinking_water_systems	29	trigram	water_systems	66	bigram
annual_environmental_justice	27	trigram	vulnerable_communities	65	bigram
environmental_justice_advisory	27	trigram	epa_region	62	bigram
fiscal_annual_environmental	25	trigram	environmental_public	57	bigram
justice_advisory_council	24	trigram	federal_agencies	56	bigram
environmental_justice_grants	22	trigram	national_environmental	51	bigram
technical_assistance_communities	20	trigram	superfund_sites	48	bigram
communities_environmental_justice	20	trigram	indigenous_peoples	46	bigram
safe_drinking_water	19	trigram	civil_rights	46	bigram
technical_assistance_services	19	trigram	local_governments	45	bigram
progress_report_2015-2016	18	trigram	urban_waters	44	bigram
interagency_environmental_justice	16	trigram	overburdened_communities	43	bigram
chemical_safety_pollution	16	trigram	action_plan	42	bigram

The bigrams seem more informative in this context because they are typical phrases that you see together like federal agencies and public health. The trigrams are not adding any extra value to most of the phrases with the added word.

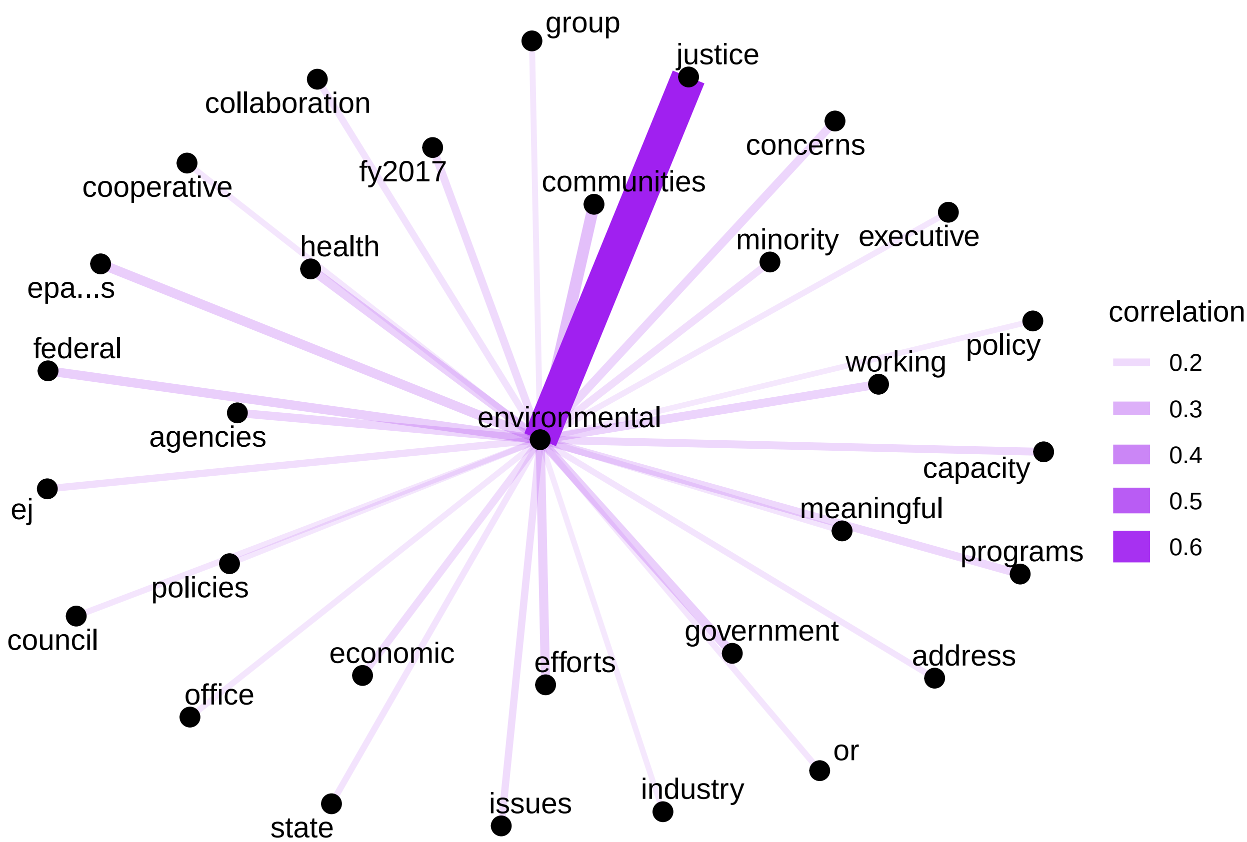
2. Choose a new focal term to replace "justice" and recreate the correlation table and network (see corr_paragraphs and corr_network chunks). Explore some of the plotting parameters in the cor_network chunk to see if you can improve the clarity or amount of information your plot conveys. Make sure to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

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

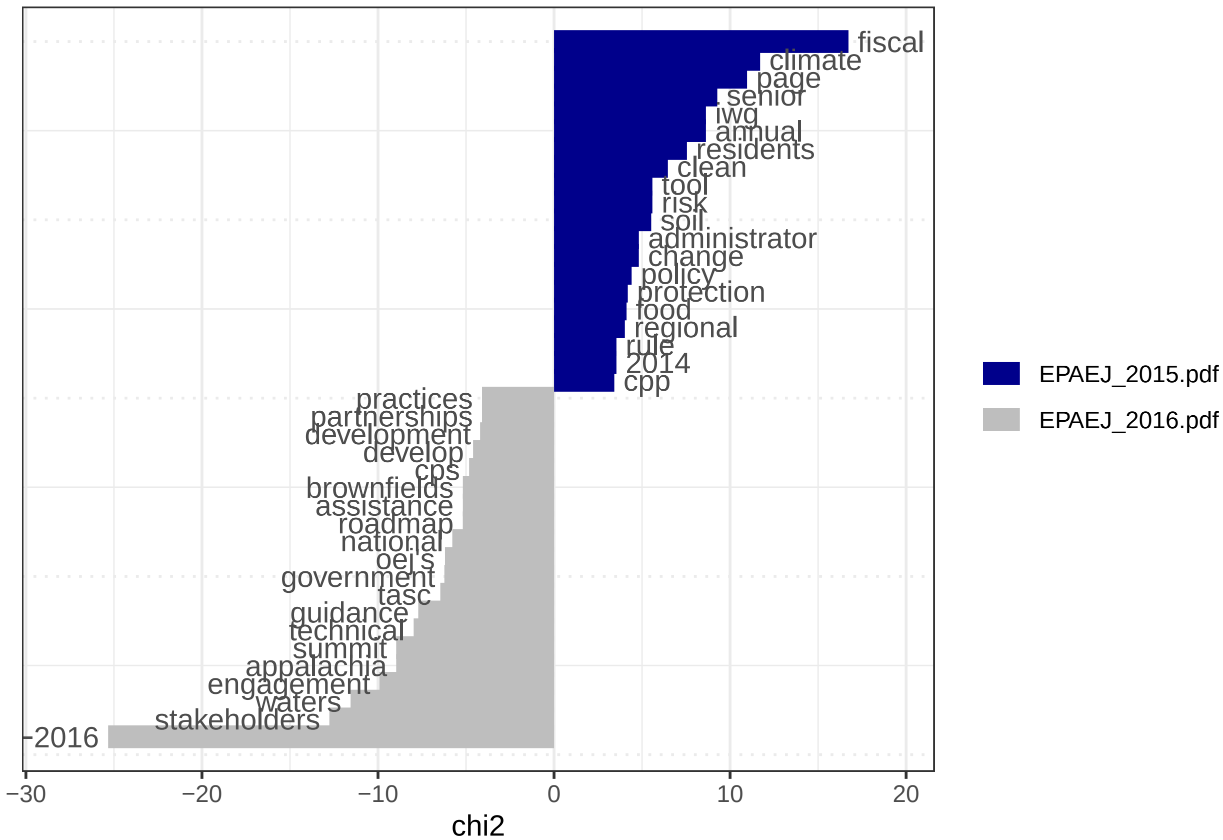


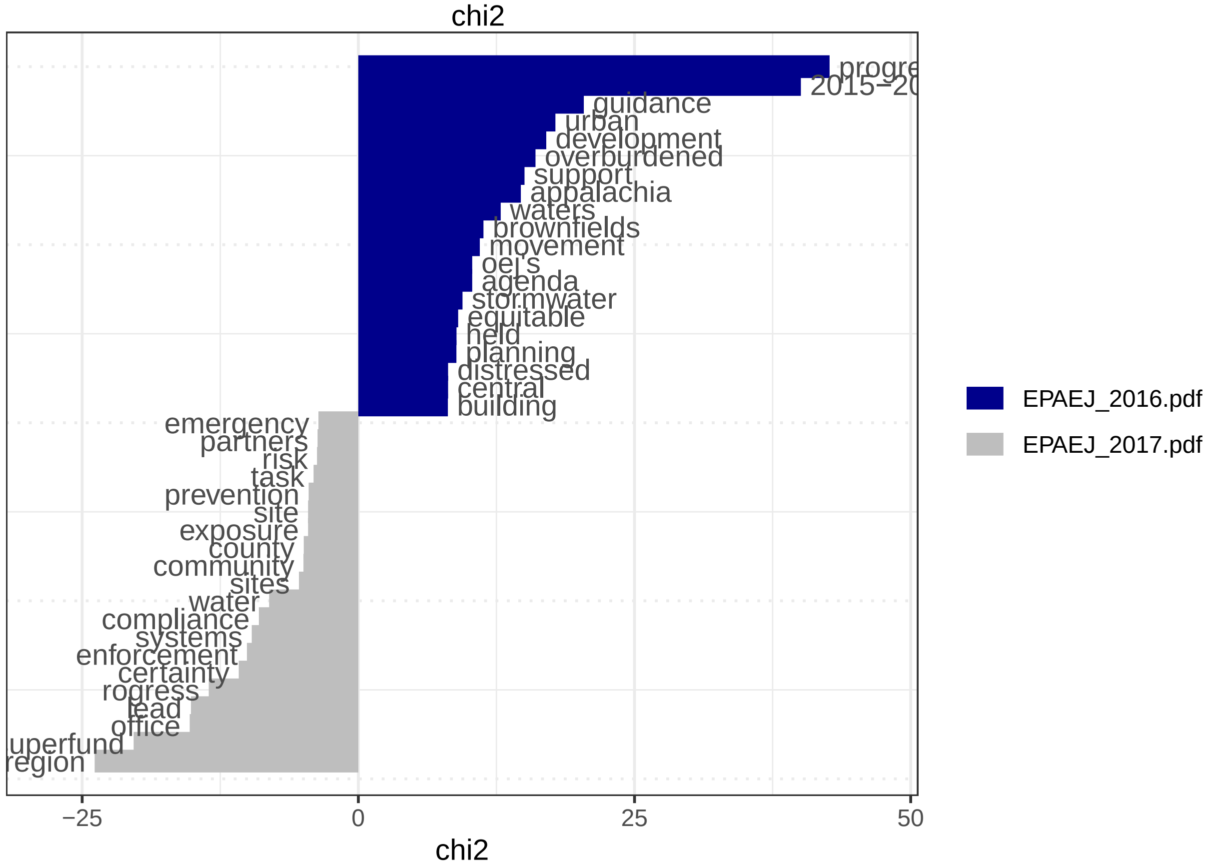

```
env_cors <- word_cors %>%
  filter(item1 == "environmental") %>%
  mutate(n = 1:n())

env_cors %>%
  filter(n <= 30) %>%
  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()
```

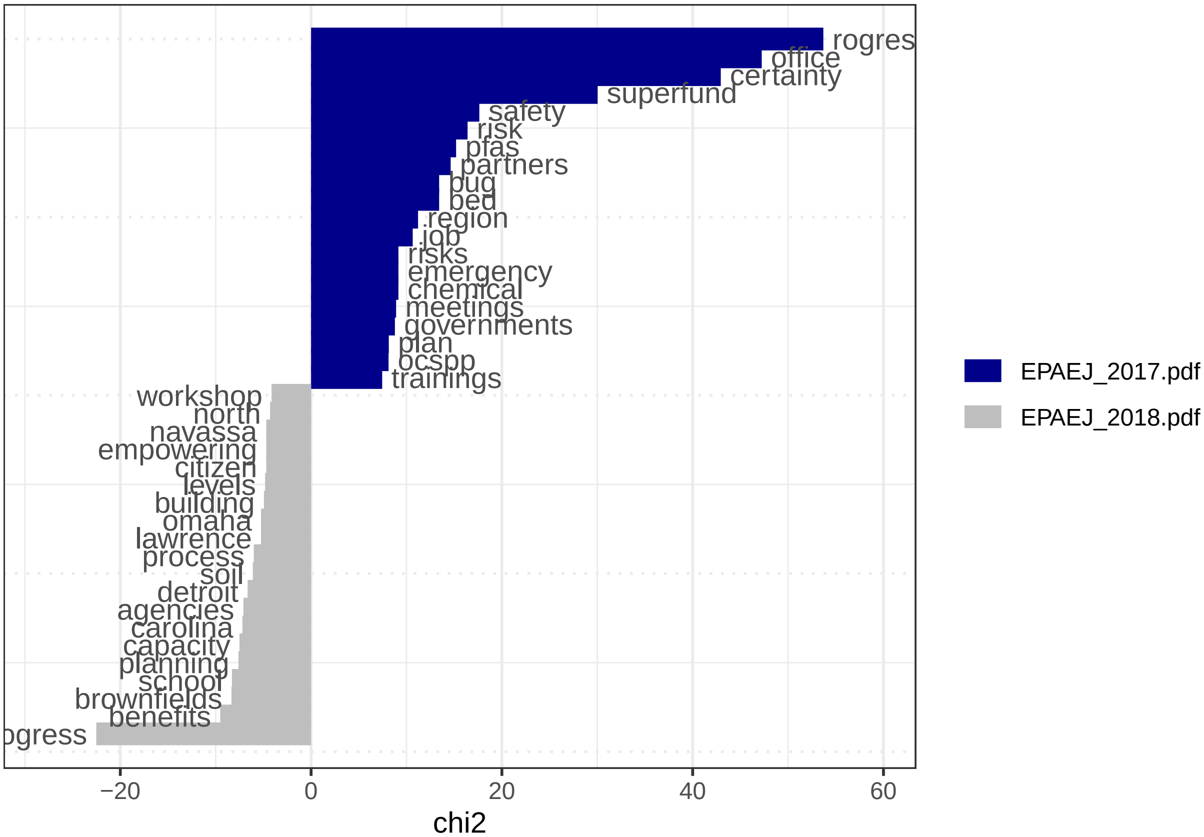


3. Write a function that allows you to conduct a keyness analysis to compare two individual EPA reports (hint: that means target and reference need to both be individual reports). Run the function on 3 pairs of reports, generating 3 keyness plots.

```
keyness_fn <- function(){
  for (i in 1:3) {
    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()
```


fiscal
climate
page
senior
iwg
annual
residents
clean
tool
risk
soil
administrator
change
policy
protection
food
regional
rule
2014
cpp
practices
partnerships
development
develop
cps
brownfields
assistance
roadmap
national
oej's
government
tasc
guidance
technical
summit
appalachia
engagement
waters
stakeholders
-2016
−30
−20
−10
0
10
20
chi2
EPAEJ_2015.pdf
EPAEJ_2016.pdf


guidance
urban
development
overburdened
support
appalachia
waters
brownfields
movement
oej's
agenda
stormwater
equitable
held
planning
distressed
central
building
emergency
partners
risk
task
prevention
site
exposure
county
community
sites
water
compliance
systems
enforcement
certainty
rogress
lead
office
uperfund
region
−25
0
25
50
chi2
EPAEJ_2016.pdf
EPAEJ_2017.pdf



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

```
woi <- c("minority")

in_window <- tokens_keep(toks1,
                         pattern = woi,
                         window = 10) %>%
  tokens_remove(pattern = woi) %>%
  tokens_tolower() %>%
  tokens_remove(pattern = (stop_vec))

in_dfm <- dfm(in_window)

out_window <- tokens_remove(toks1,
                            pattern = woi,
                            window = 10) %>%
  tokens_tolower() %>%
  tokens_remove(patter = (stop_vec))

out_dfm <- dfm(out_window)

dfms <- rbind(in_dfm, out_dfm)

in_keyness <- textstat_keyness(dfms,
```

```
                                        target = seq_len(ndoc(in_dfm)))
in_keyness[1:10] %>%
  kable()
```

feature	chi2	p	n_target	n_reference
low-income	1124.93375	0	57	31
populations	421.04183	0	35	48
income	173.67506	0	14	17
indigenous	147.95410	0	25	80
low-	144.76159	0	8	4
tribal	76.78515	0	31	216
communities	66.93060	0	71	869
low	62.78066	0	6	8
1994	53.84014	0	5	6
historically	53.84014	0	5	6