Tweets analysis

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
library(ggplot2)
library(dplyr)
library(stringr)
library(tidytext)
library(janeaustenr)
library(ggplot2)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:igraph':
##
##     crossing
```

```
library(igraph)
library(ggraph)
```

```
a = read.csv('ZelenskyyUa_tweets.csv')
a$year = substr(a$Datetime, 1, 4)
a = a[ which(a$Language=='en' & a$year == 2022),]
```

```
string = c()
for (i in range(1,dim(a)[1])){
  string = c(string, a$Text)
}

w = c()
for (j in string){
  b = unlist(strsplit(j, ' '))
  d = c()
  for (i in b){
    if ((substr(i, 1, 1)) != '@'){
    d = c(d, i)}}
  d = str_c(d, collapse = " ")
w = c(w, d)
}

df1 = data.frame()
for (k in w){
  df1 <- rbind(df1, k)
}


df1$year = 2022
colnames(df1) <- c("word", "year")

#removing punctuations
df1$word = gsub('[[:punct:] ]+',' ',df1$word)

write.csv(df1, 'zen_words.csv')
```

```
df11 = read.csv('zen_words.csv')

df11 = df11 %>%
  unnest_tokens(word, word)

df11 = df11 %>%
  unnest_tokens(word, word)%>%
  group_by(word)%>%
  summarise(count = n())%>%
  arrange(desc(count))

a <- df11 %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
head(a, 10)
```

```
## # A tibble: 10 x 2
##    word         count
##    <chr>        <int>
##  1 support        120
##  2 ukraine         98
##  3 security        88
##  4 grateful        68
##  5 amp             62
##  6 discussed       58
##  7 conversation    54
##  8 president       44
##  9 assistance      40
## 10 peace           40
```

```
a$total_sum = sum(a$count)
a = a %>%
  mutate(rank = row_number(), `term frequency` = count/total_sum)
```

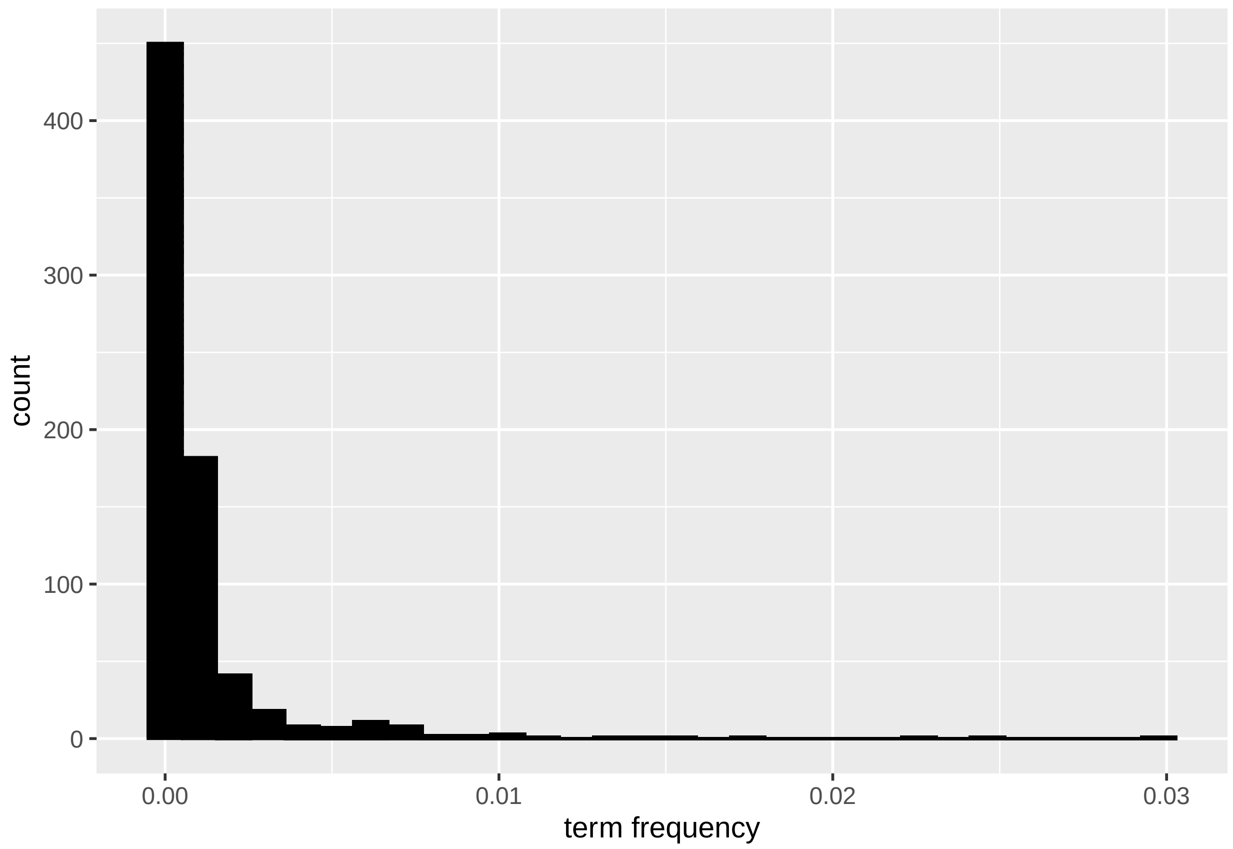
#Plot histogram of word frequencies

```
a$term_frequency = unlist(a$term_frequency)
```

```
## Warning: Unknown or uninitialised column: `term_frequency`.
```

```
ggplot(a, aes(`term frequency`, fill = word)) +
  geom_histogram(color = 'black', show.legend = FALSE)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



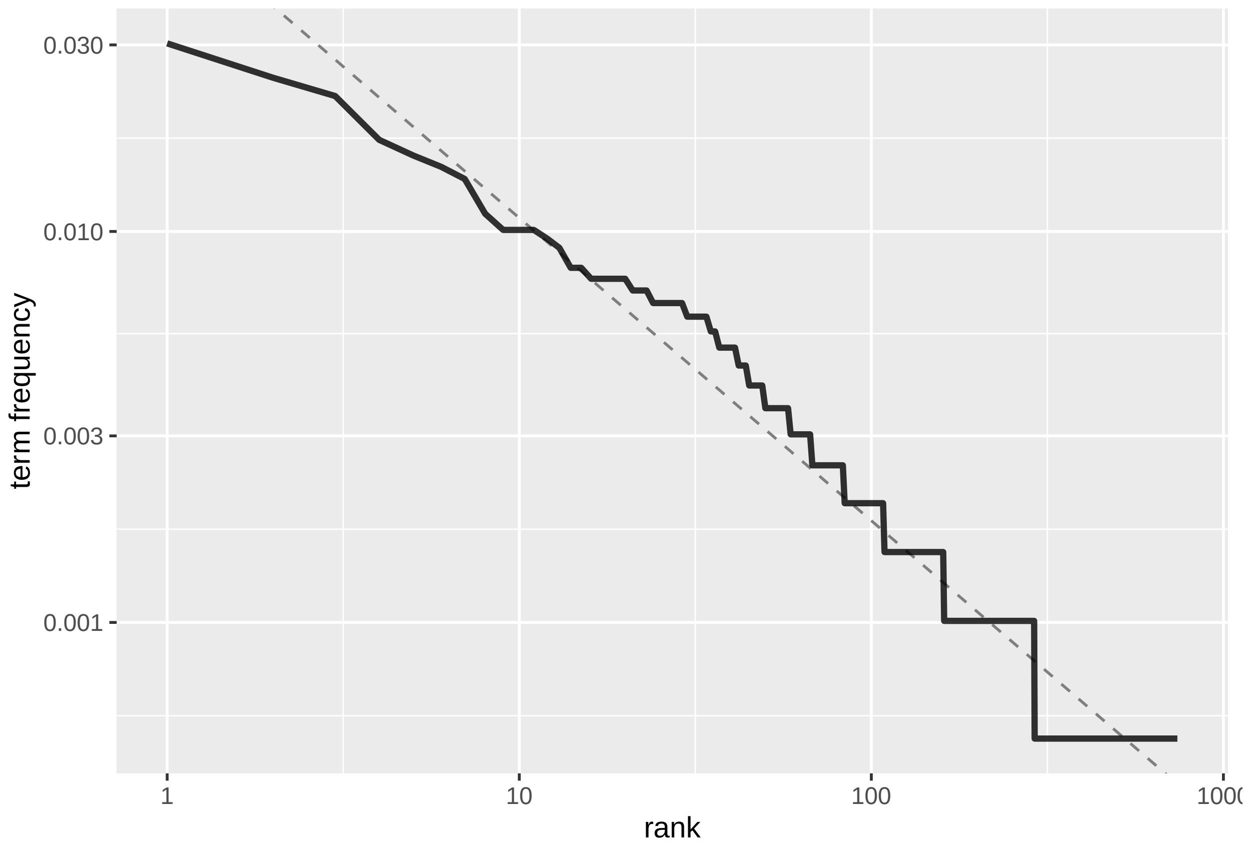
#Use Zipf's law and plot log-log plots of word frequencies and rank

```
lm(log10(`term frequency`) ~ log10(rank), data = a)
```

```
##
## Call:
## lm(formula = log10(`term frequency`) ~ log10(rank), data = a)
##
## Coefficients:
## (Intercept)  log10(rank)
##     -1.1911      -0.7743
```

```
log_plot = a %>%
  ggplot(aes(rank, `term frequency`)) +
  geom_abline(intercept = -1.1911 , slope = -0.7743,
              color = "gray50", linetype = 2) +
  geom_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +
  scale_x_log10() +
  scale_y_log10()

log_plot
```



Create bigram network graphs for each year

#forming bi-grams with two words, dividing the words and removing the stop words for the given words.

```
df11 = read.csv('zen_words.csv')
df11_bigrams <- df11 %>%
  unnest_tokens(bigram, word, token = "ngrams", n = 2)%>%
  count(bigram, sort = T)%>%
  separate(bigram, c('word1', 'word2'), sep = ' ') %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
```

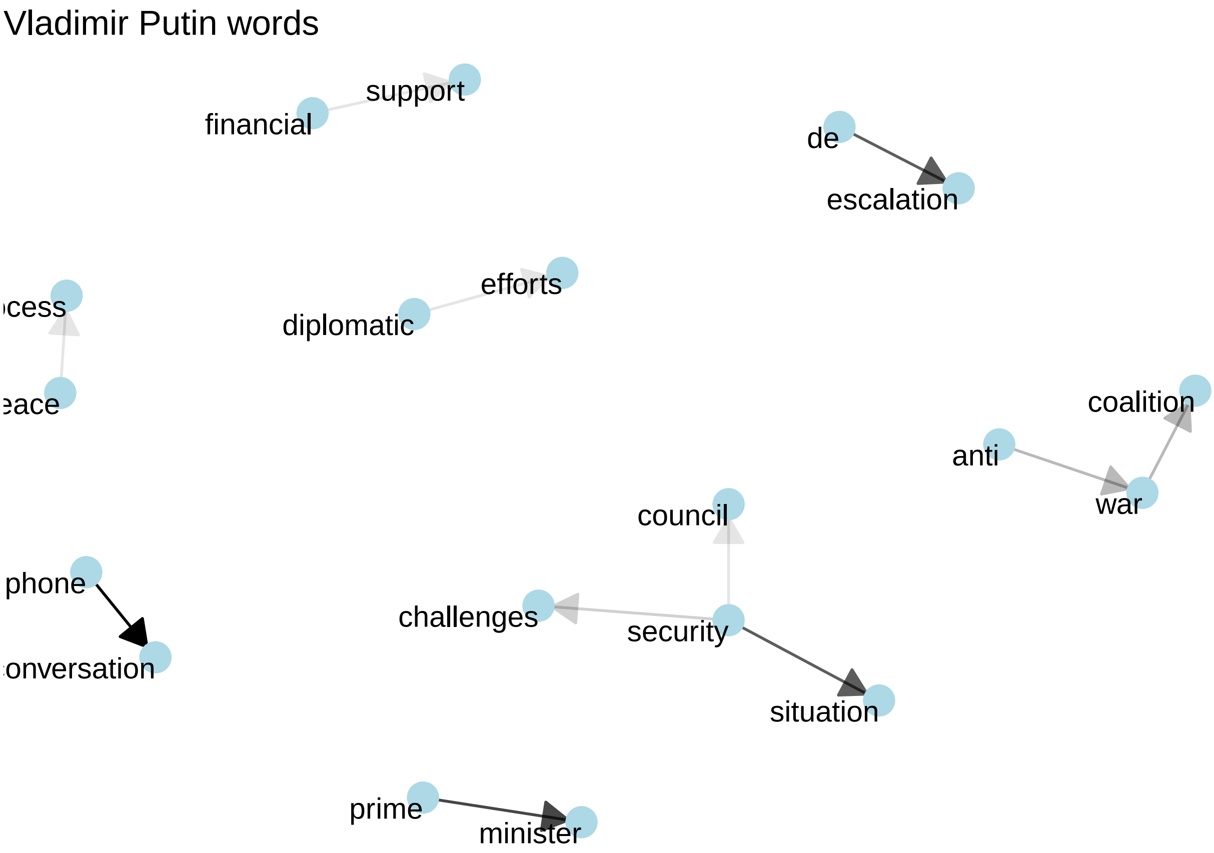
#at least 10 connections

```
connections <- df11_bigrams %>%
  filter(n > 10) %>%
  graph_from_data_frame()
```

```
set.seed(2020)

a <- grid::arrow(type = "closed", length = unit(.15, "inches"))

ggraph(connections, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "lightblue", size = 5) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_void() + ggtitle("Vladimir Putin words")
```



```
a = read.csv('KremlinRussia_E_tweets.csv')
a$year = substr(a$Datetime, 1, 4)
a = a[ which(a$Languages=='en' & a$year == 2022),]

string = c()
for (i in range(1,dim(a)[1])){
  string = c(string, a$Text)
}

w = c()
for (j in string){
  b = unlist(strsplit(j, ' '))
  d = c()
  for (i in b){
    if ((substr(i, 1, 1)) != '@'){
    d = c(d, i)}}
  d = str_c(d, collapse = " ")
w = c(w, d)
}
```

```r
df1 = data.frame()
for (k in w){
  df1 <- rbind(df1, k)
}


df1$year = 2022
colnames(df1) <- c("word", "year")

#removing punctuations
df1$word = gsub('[[:punct:] ]+',' ',df1$word)

write.csv(df1, 'kren_words.csv')
```

```r
df11 = read.csv('kren_words.csv')

df11 = df11 %>%
  unnest_tokens(word, word)

df11 = df11 %>%
  unnest_tokens(word, word)%>%
  group_by(word)%>%
  summarise(count = n())%>%
  arrange(desc(count))

a <- df11 %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```r
head(a, 10)
```

```
## # A tibble: 10 x 2
##    word         count
##    <chr>        <int>
##  1 https          328
##  2 president      102
##  3 putin           72
##  4 vladimir        72
##  5 telephone       64
##  6 conversation    60
##  7 meeting         56
##  8 minister        40
##  9 prime           32
## 10 talks           32
```

#Show top 10 words by the highest value of word frequency

```r
a$total_sum = sum(a$count)
a = a %>%
  mutate(rank = row_number(), `term frequency` = count/total_sum)
```

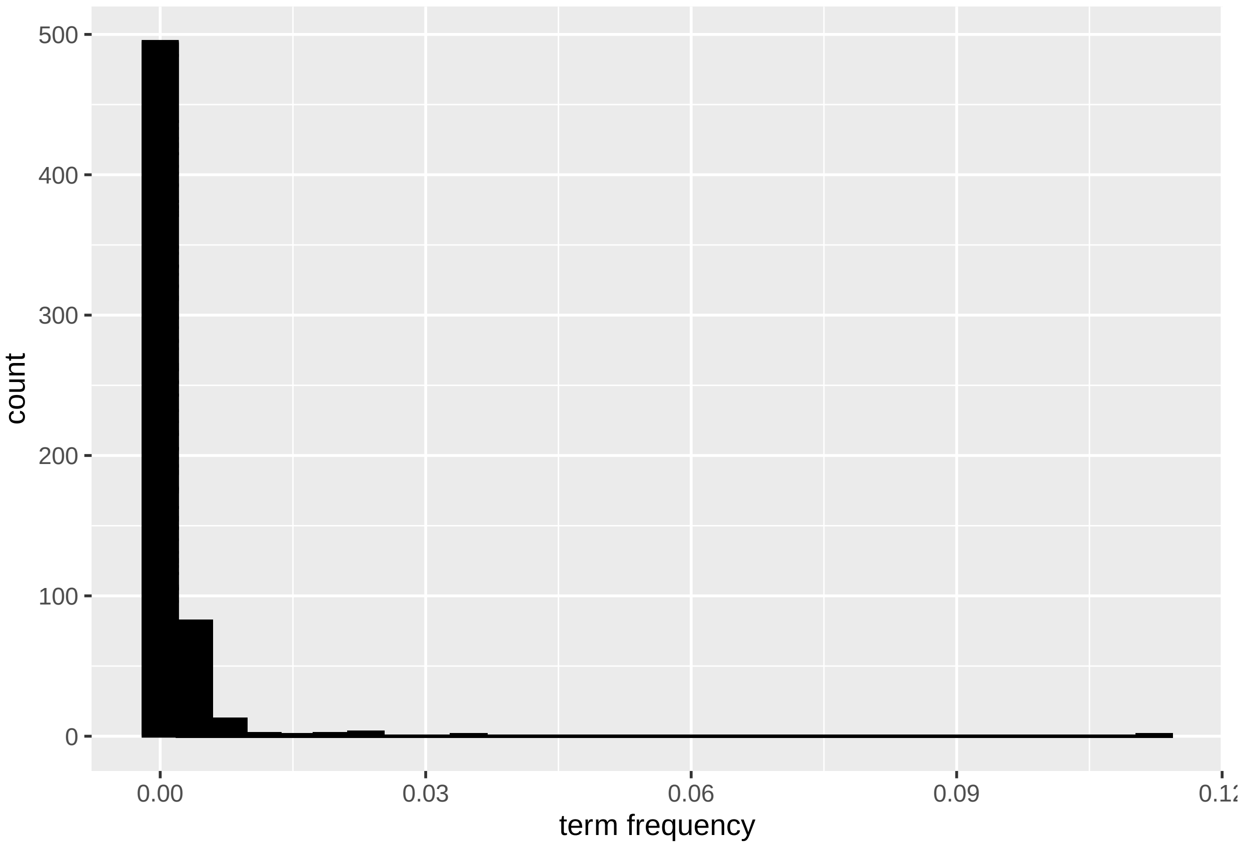
#Plot histogram of word frequencies

```
a$term_frequency = unlist(a$term_frequency)
```

```
## Warning: Unknown or uninitialised column: `term_frequency`.
```

```
ggplot(a, aes(`term frequency`, fill = word)) +
  geom_histogram(color = 'black', show.legend = FALSE)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



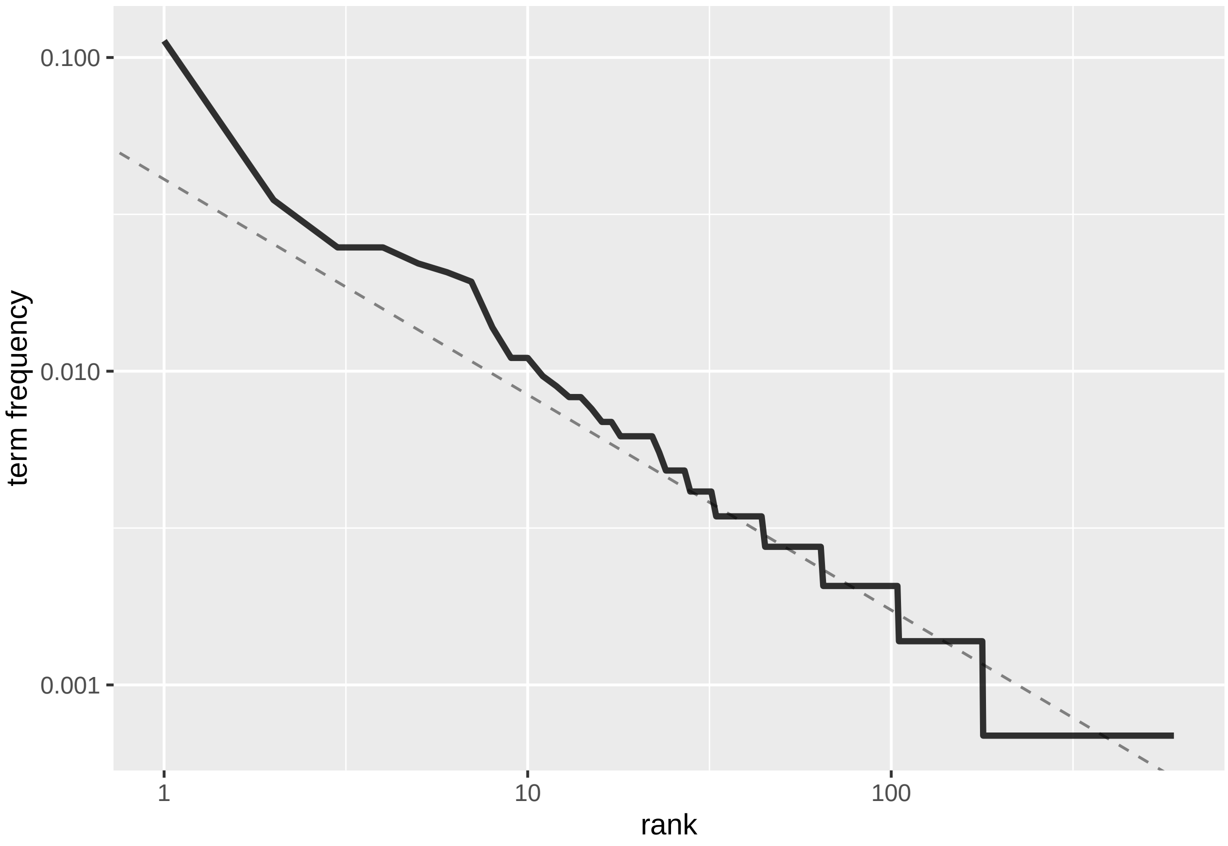
#Use Zipf's law and plot log-log plots of word frequencies and rank

```
lm(log10(`term frequency`) ~ log10(rank), data = a)
```

```
##
## Call:
## lm(formula = log10(`term frequency`) ~ log10(rank), data = a)
##
## Coefficients:
## (Intercept)  log10(rank)
##     -1.3881      -0.6867
```

```
log_plot = a %>%
  ggplot(aes(rank, `term frequency`)) +
  geom_abline(intercept = -1.3881 , slope = -0.6867,
              color = "gray50", linetype = 2) +
  geom_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +
  scale_x_log10() +
  scale_y_log10()

log_plot
```



Create bigram network graphs

#forming bi-grams with two words, dividing the words and removing the stop words for the given words.

```
df11 = read.csv('zen_words.csv')
df11_bigrams <- df11 %>%
  unnest_tokens(bigram, word, token = "ngrams", n = 2)%>%
  count(bigram, sort = T)%>%
  separate(bigram, c('word1', 'word2'), sep = ' ') %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

head(df11_bigrams, 10)
```

```
##          word1        word2  n
## 1        phone conversation 32
## 2        prime     minister 26
## 3           de   escalation 24
## 4     security    situation 24
## 5         anti          war 16
## 6          war    coalition 16
## 7     security   challenges 14
## 8   diplomatic      efforts 12
## 9    financial      support 12
## 10       peace      process 12
```

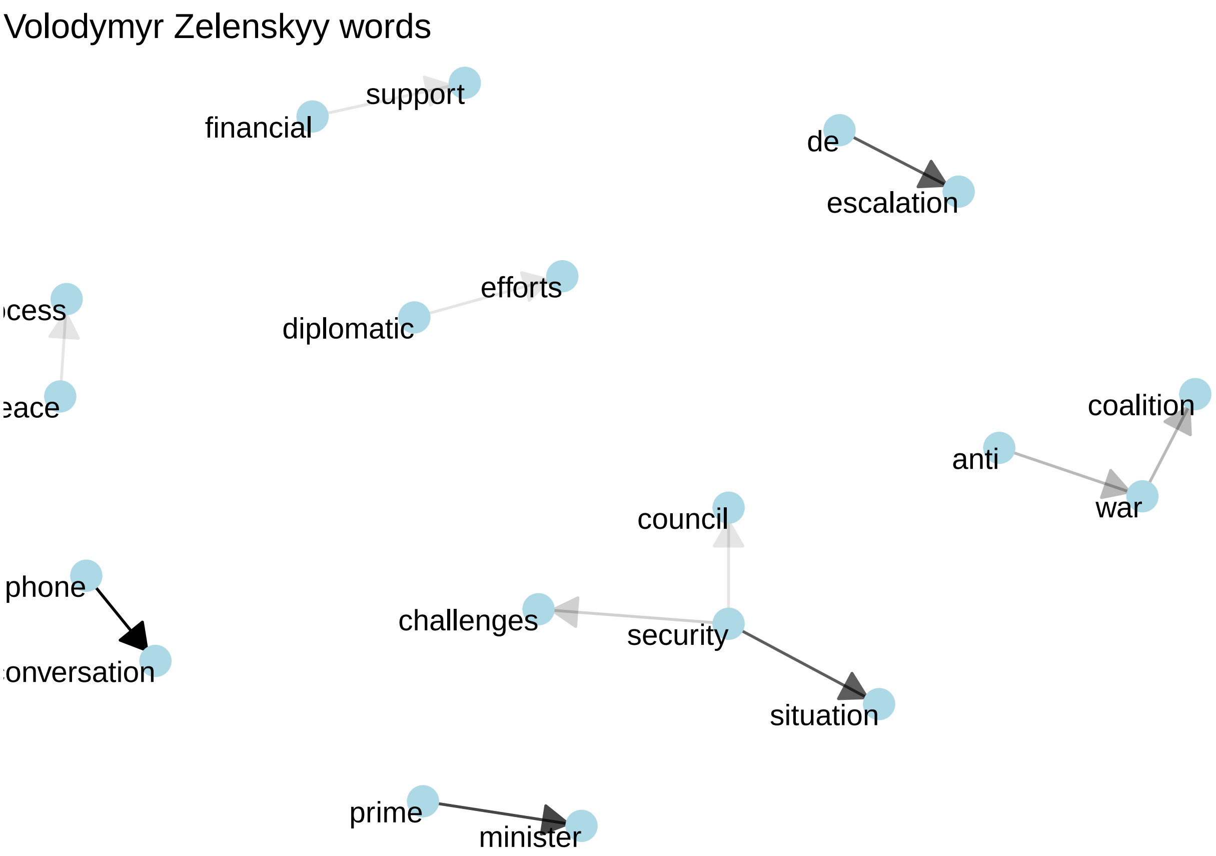
#at least 10 connections

```
connections <- df11_bigrams %>%
  filter(n > 10) %>%
  graph_from_data_frame()
```

```
set.seed(2020)

a <- grid::arrow(type = "closed", length = unit(.15, "inches"))

ggraph(connections, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color = "lightblue", size = 5) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
  theme_void() + ggtitle("Volodymyr Zelenskyy words")
```



Volodymyr Zelenskyy words