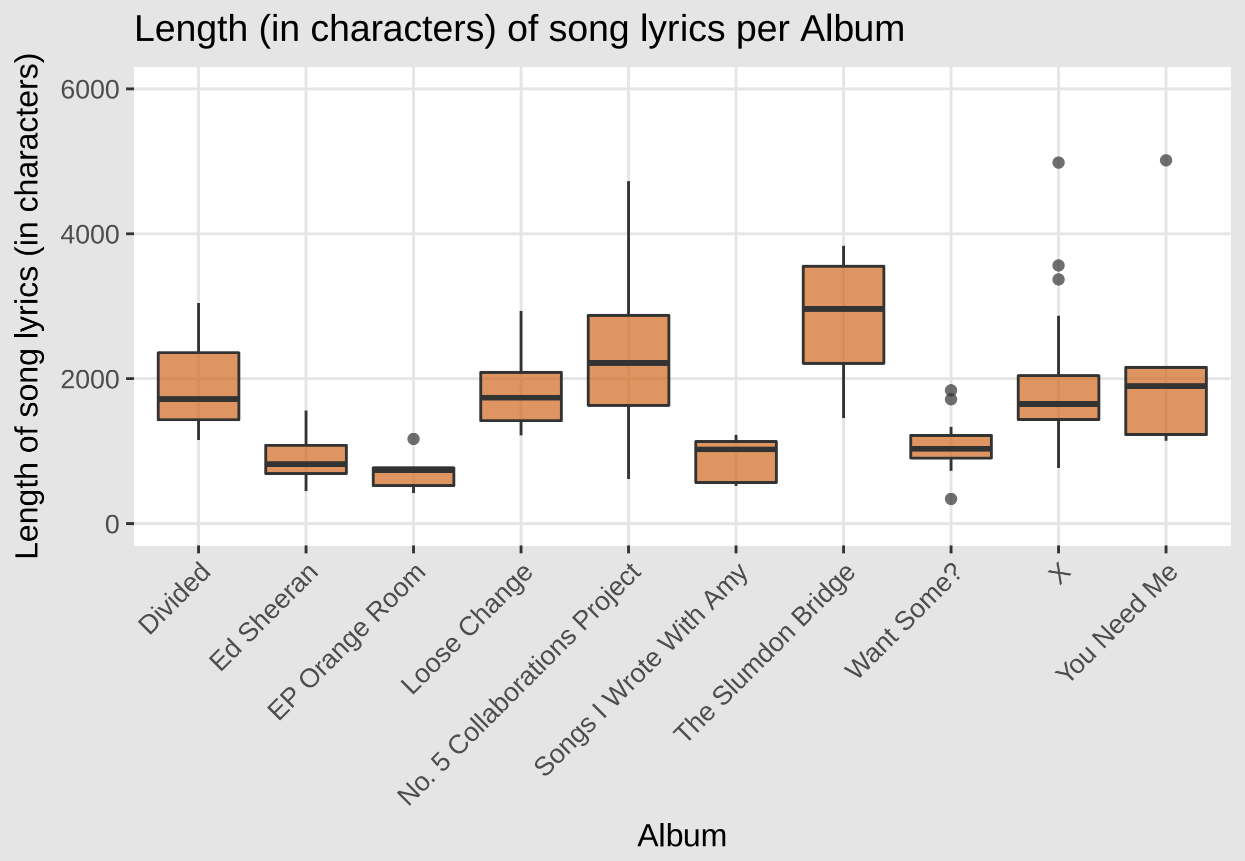
Text analysis of Ed Sheeran Songs

Chi Ting Low

```r
#ed sheeran lyrics
library(readxl)
library(tidyverse)
library(stringr)
library(tidytext)
library(wordcloud)
library(tm)
library(stopwords)
library(ggthemes)
library(cld3)
library(DT)
library(lattice)
library(udpipe)

#read data
songs <- read_xlsx('Lyrics.xlsx')

#lyrics words counts
songs$characters <- str_count(songs$Lyrics)
```

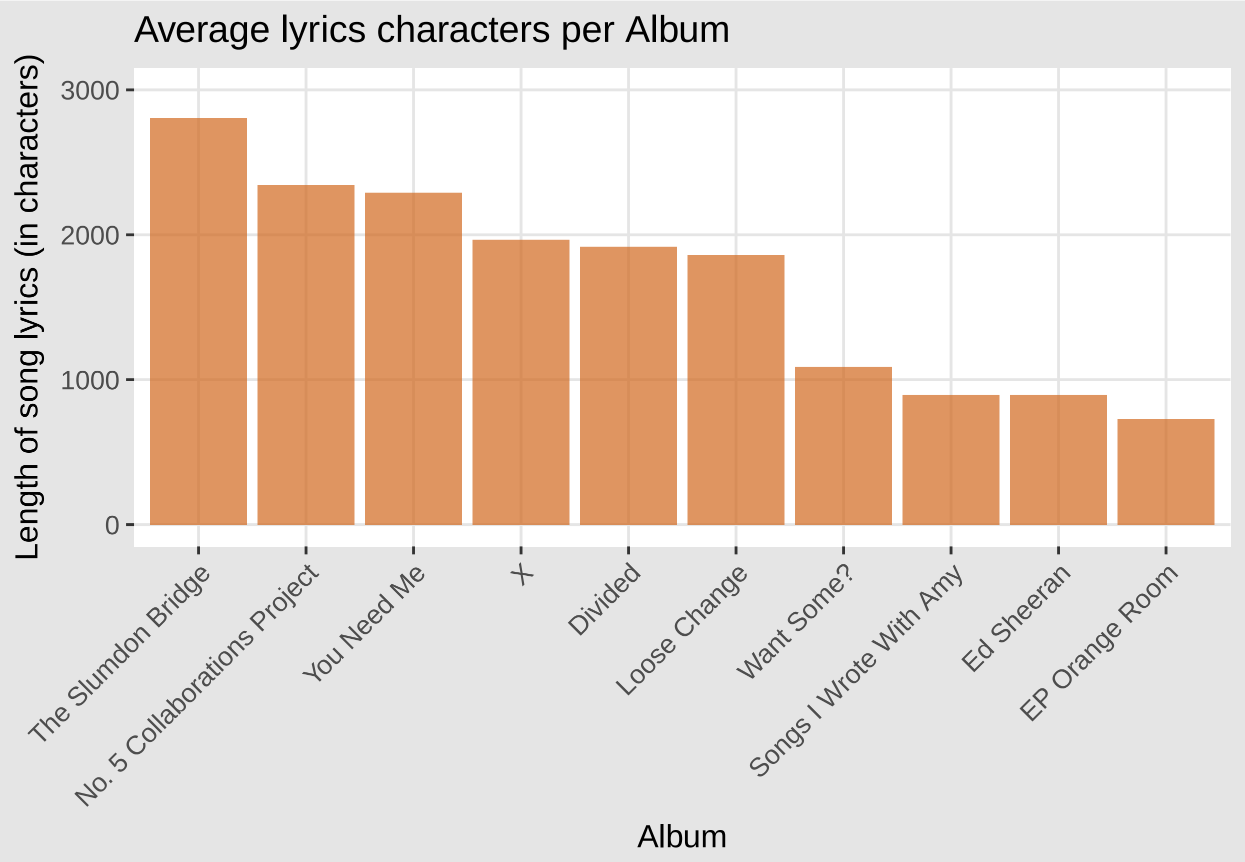
R “str_count”

```r
## number of characters per song
songs %>%
  ggplot() +
  geom_boxplot(aes(Album, characters), fill = "chocolate", alpha = 0.7) +
  labs(y = "Length of song lyrics (in characters)", x = "Album",
       title = "Length (in characters) of song lyrics per Album") +
  theme_igray() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ylim(0, 6000)
```



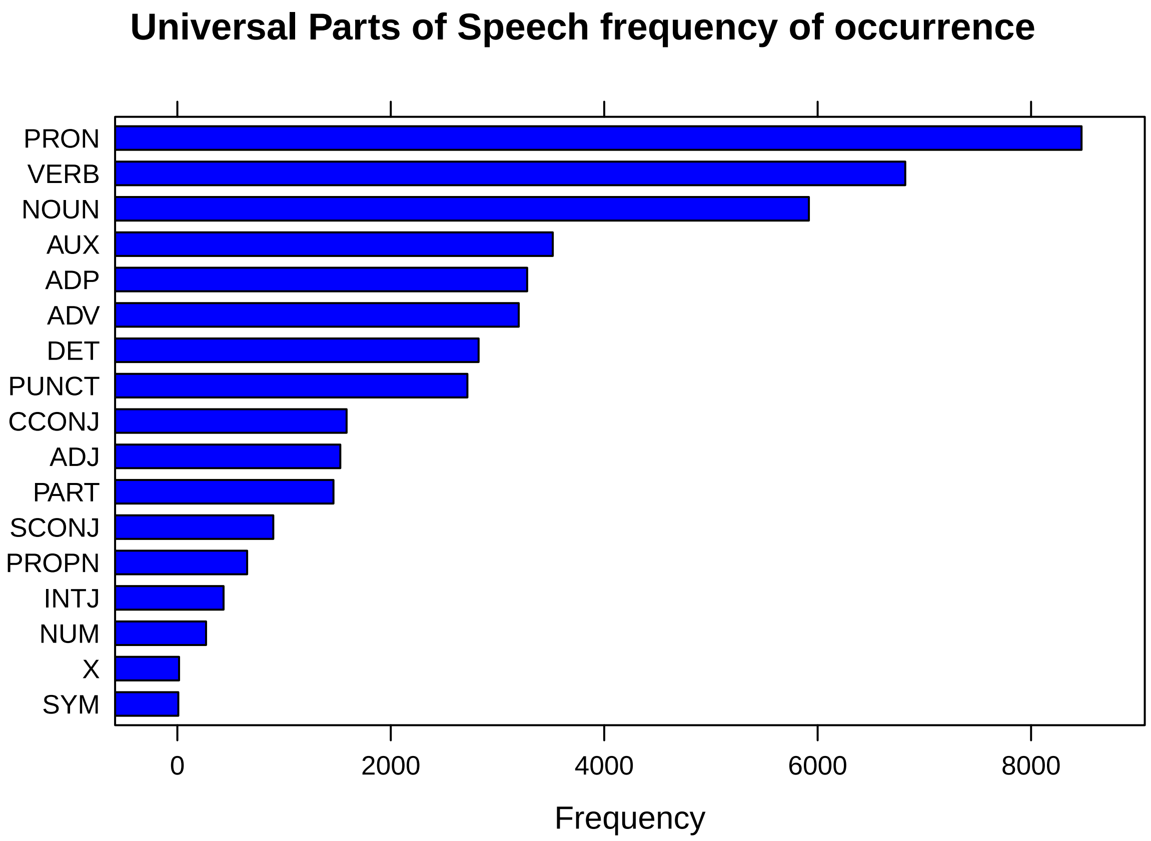
The Slumdon Bridge No. 5 Collaboration Project X Divided

```
songs %>%
  group_by(Album) %>%
  summarise(characters = round(mean(characters, na.rm = TRUE), 0)) %>%
  ggplot(aes(reorder(Album, -characters), characters)) +
  geom_col(fill = "chocolate", alpha = 0.7) +
  labs(y = "Length of song lyrics (in characters)", x = "Album",
       title = "Average lyrics characters per Album") +
  theme_igray() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ylim(0, 3000)
```



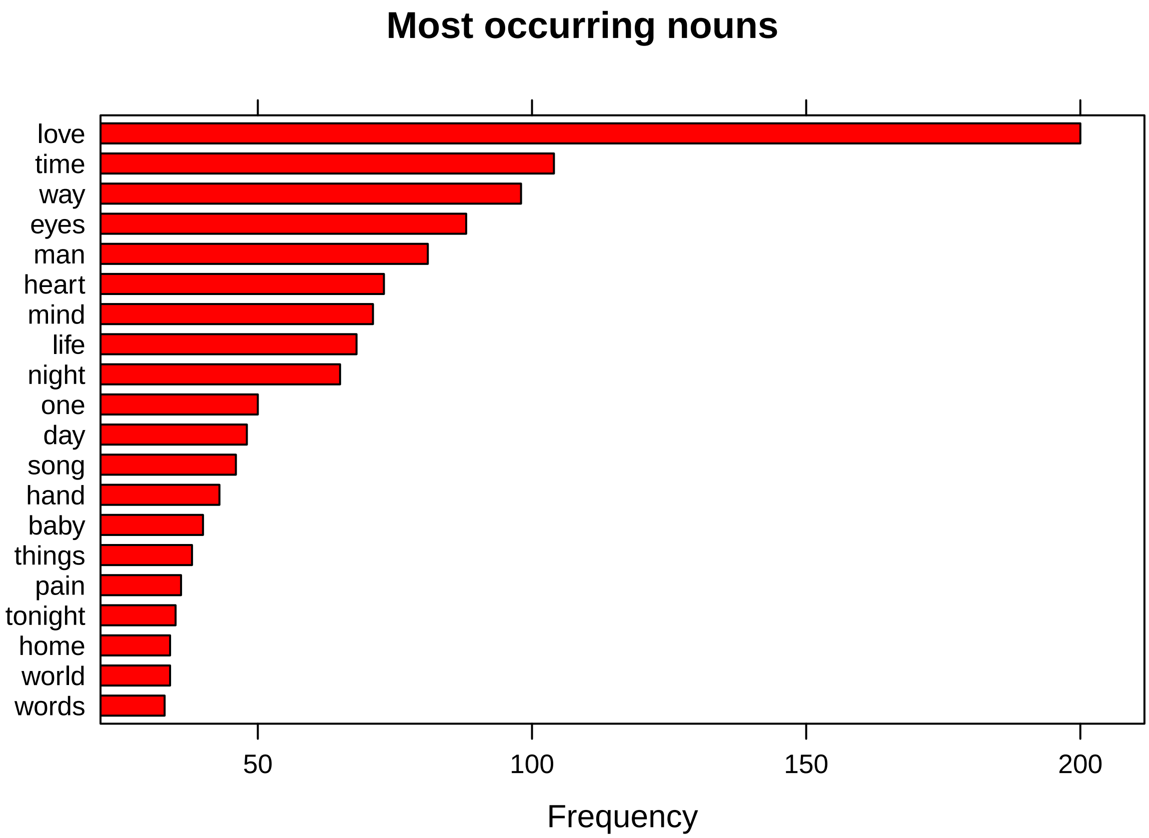
The Slumdon Bridge

```
#english
#model <- udpipe_download_model(language = "english")
udmodel <- udpipe_load_model(file = 'english-ud-2.0-170801.udpipe')

#tokenized the lyrics
token <- udpipe_annotate(udmodel, songs$Lyrics)
x <- data.frame(token)


#Universal POS
key <- txt_freq(x$upos)
key$key <- factor(key$key, levels = rev(key$key))
barchart(key ~ freq,
         data = key,
         col = "blue",
         main = "Universal Parts of Speech frequency of occurrence",
         xlab = "Frequency")
```



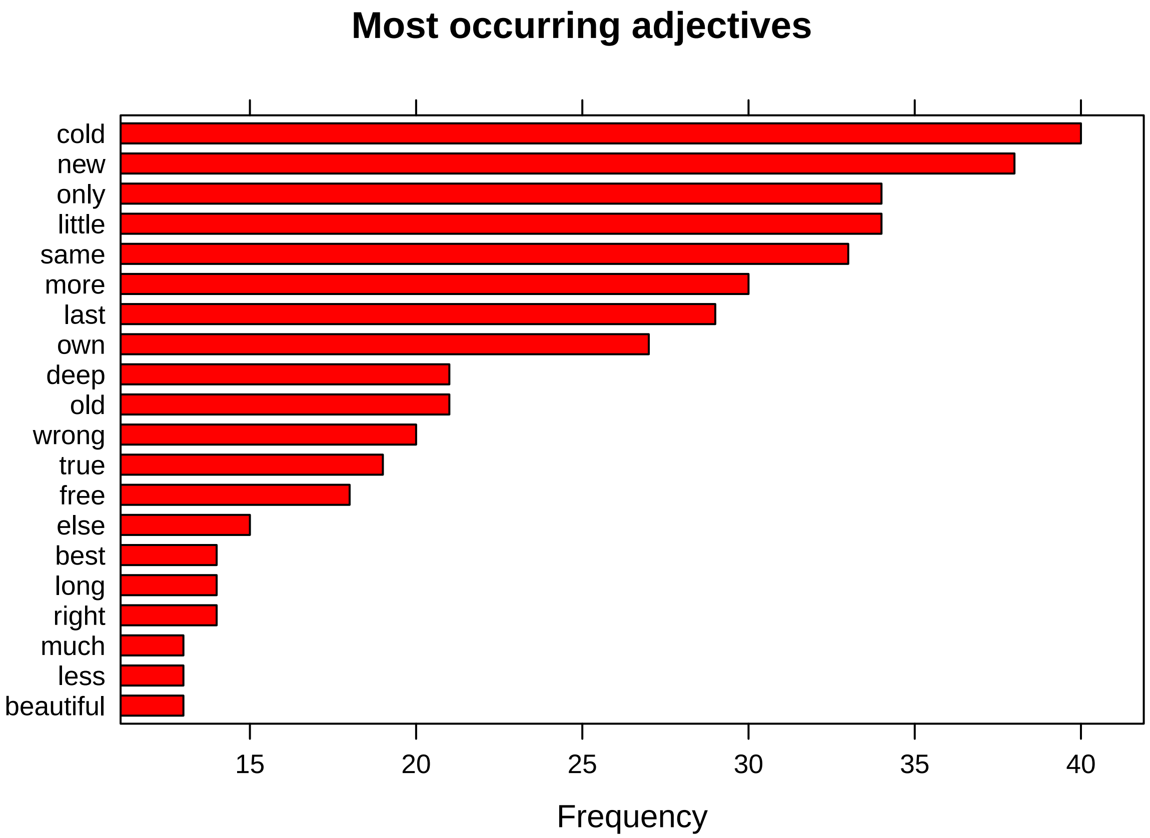
udpipe Ed Sheeran

```
#top nouns
nouns <- subset(x, upos %in% c("NOUN"))
nouns <- txt_freq(nouns$token)
nouns$key <- factor(nouns$key,
                    levels = rev(nouns$key))
barchart(key ~ freq,
         data = head(nouns, 20),
         col = "red",
         main = "Most occurring nouns",
         xlab = "Frequency")
```



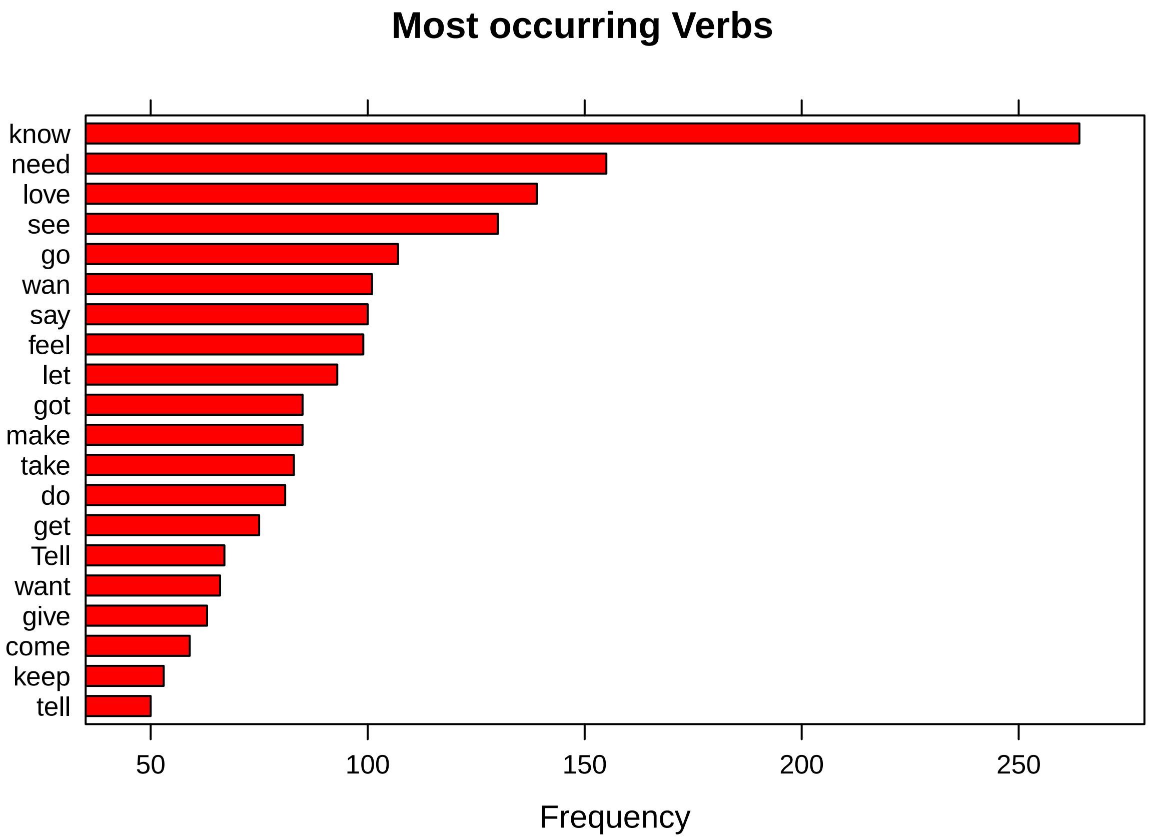
20 l

```
#top adjective
adj <- subset(x, upos %in% c("ADJ"))
adj <- txt_freq(adj$token)
adj$key <- factor(adj$key,
                    levels = rev(adj$key))
barchart(key ~ freq,
         data = head(adj, 20),
         col = "red",
         main = "Most occurring adjectives",
         xlab = "Frequency")
```



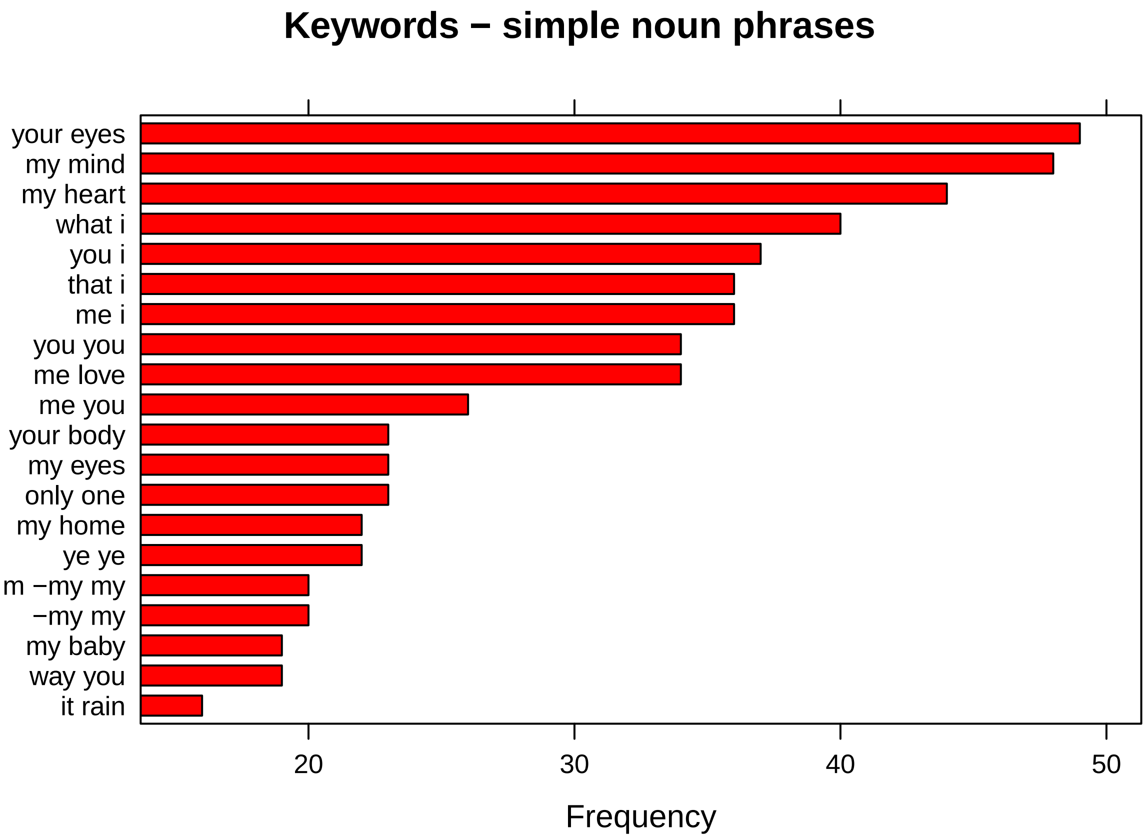
cold new only little

```
#top verb
verb <- subset(x, upos %in% c("VERB"))
verb <- txt_freq(verb$token)
verb$key <- factor(verb$key,
                    levels = rev(verb$key))
barchart(key ~ freq,
         data = head(verb, 20),
         col = "red",
         main = "Most occurring Verbs",
         xlab = "Frequency")
```



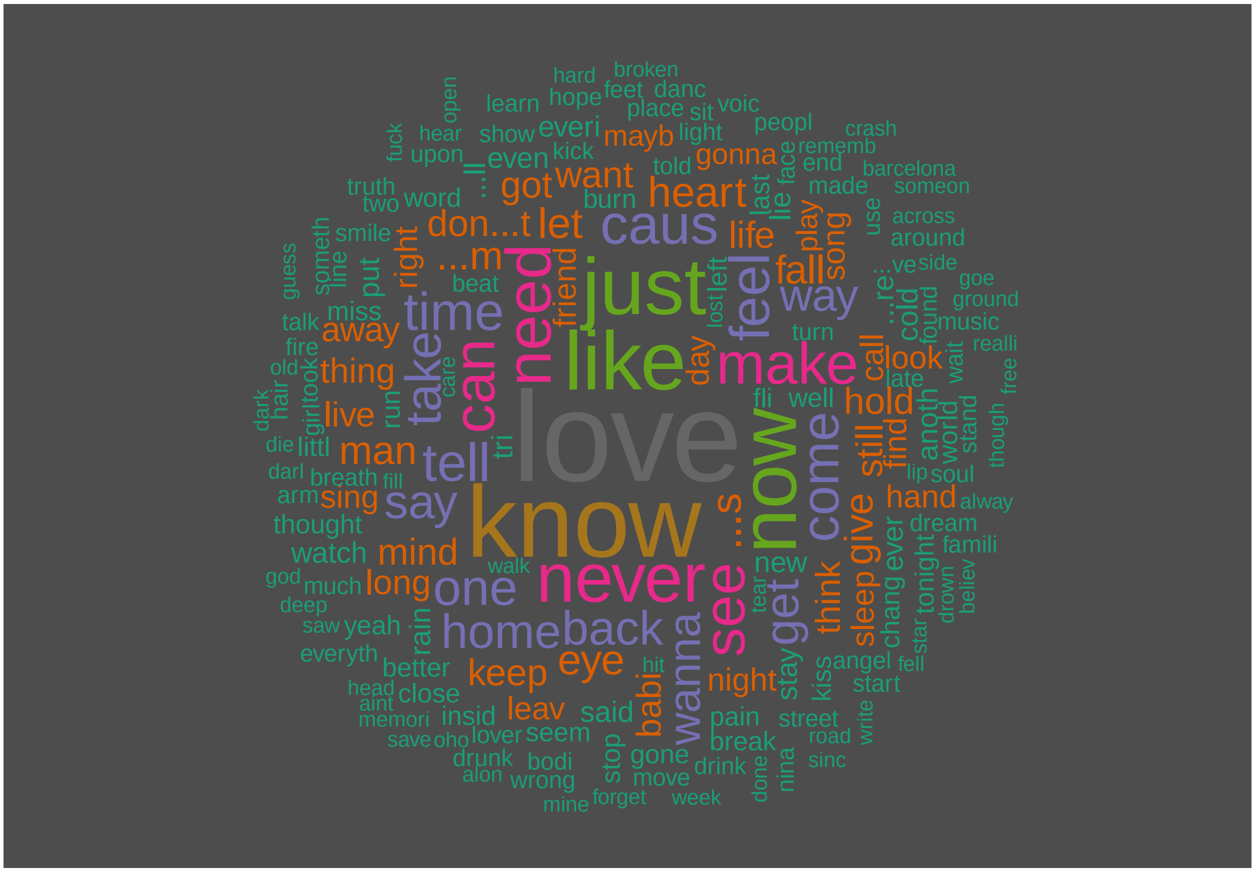
know need love

```
##combining the most frequent nouns and verb
x$phrase_tag <- as_phrasemachine(x$upos,
                                 type = "upos")
words <- keywords_phrases(x = x$phrase_tag,
                          term = tolower(x$token),
                          pattern = "(A|N)*N(P+D*(A|N)*N)*",
                          is_regex = TRUE,
                          detailed = FALSE)
words <- subset(words, ngram > 1 & freq > 3)
words$key <- factor(words$keyword,
                    levels = rev(words$keyword))
barchart(key ~ freq,
         data = head(words, 20),
         col = "red",
         main = "Keywords - simple noun phrases",
         xlab = "Frequency")
```



eyes my mind my heart Ed Sheeran

```
## _text cleaning
#convert into corpus
docs <- Corpus(VectorSource(songs$Lyrics))
# Convert the text to lower case
docs <- tm_map(docs, content_transformer(tolower))
# Remove numbers
docs <- tm_map(docs, removeNumbers)
# Remove english common stopwords
docs <- tm_map(docs, removeWords, stopwords("english"))
# Remove punctuations
docs <- tm_map(docs, removePunctuation)
# Eliminate extra white spaces
docs <- tm_map(docs, stripWhitespace)
# Text stemming
docs <- tm_map(docs, stemDocument)


dtm <- TermDocumentMatrix(docs)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing = TRUE)
d <- data.frame(word = names(v),freq = v)

# Create a word cloud
par(bg = "grey30")
set.seed(1234)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
          max.words = 200, random.order = FALSE, rot.per = 0.35,
          colors = brewer.pal(8, "Dark2"))
```



Ed Sheeran

https://cran.r-project.org/web/packages/udpipe/udpipe.pdf

https://github.com/bnosac/udpipe

https://cran.r-project.org/web/packages/udpipe/vignettes/udpipe-annotation.html

https://bnosac.github.io/udpipe/en/index.html

http://ufal.mff.cuni.cz/udpipe/users-manual

http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know