Ejercicios Sesión 2

Alberto Armas Alvarado

1. Leer archivo de datos y tokenización

setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

getwd()

library(tokenizers)

noticia <- readLines("noticias.txt",encoding="UTF-8")

tokenize\_sentences(noticia)

tokenize\_words(noticia[[8]])

1. Limpieza de datos

library(tm)

noticia <- readLines("noticias.txt",encoding="UTF-8")

corpus <- VCorpus(VectorSource(noticia))

d <- tm\_map(corpus, content\_transformer(tolower))

d <- tm\_map(d, stripWhitespace)

d <- tm\_map(d, removePunctuation)

d <- tm\_map(d, removeNumbers)

d[["8"]][["content"]]

noticia[[8]]

1. StopWords

d <- tm\_map(d, removeWords, stopwords("spanish"))

d[["1"]][["content"]]

##### Lista personalizada 1 ###############

sw <- readLines("stopwordses.txt", encoding="UTF-8")

sw <- iconv(sw, to="ASCII//TRANSLIT")

d <- tm\_map(d, removeWords, sw)

d[["1"]][["content"]]

##### Lista personalizada 2 ###############

noticia\_tidy <- noticia %>% select(-noticia) %>% unnest(cols = c(texto\_tokenizado))

head(noticia\_tidy)

##### Se filtran las stopwords ###############

noticia\_tidy <- noticia\_tidy %>% filter(!(texto\_tokenizado %in% sw))

head(noticia\_tidy)

1. Lematización

library(SnowballC)

library(corpus)

tabes=read.delim("lemmatization-es.txt",header = FALSE, sep = "",

stringsAsFactors = FALSE)

names(tabes) <- c("stem", "term")

head(tabes,10)

stem\_liste <- function(term) {

i <- match(term, tabes$term)

if (is.na(i)) {

stem <- term

} else {

stem <- tabes$stem[[i]]

}

stem

}

names(noticia\_tidy)[1] <- "term"

noticia\_tidy=noticia\_tidy %>%left\_join(tabes, by = "term")

noticia\_tidy$lemma=ifelse(is.na(noticia\_tidy$stem ), noticia\_tidy$term, noticia\_tidy$stem)

1. Tagging

library(udpipe)

ud\_model <- udpipe\_download\_model(language = "spanish")

ud\_model <- udpipe\_load\_model(ud\_model$file\_model)

noticia\_tag <- udpipe\_annotate(ud\_model, x = noticia$noticia, doc\_id = noticia$ID)

noticia\_tag <- as.data.frame(noticia\_tag)

library(lattice)

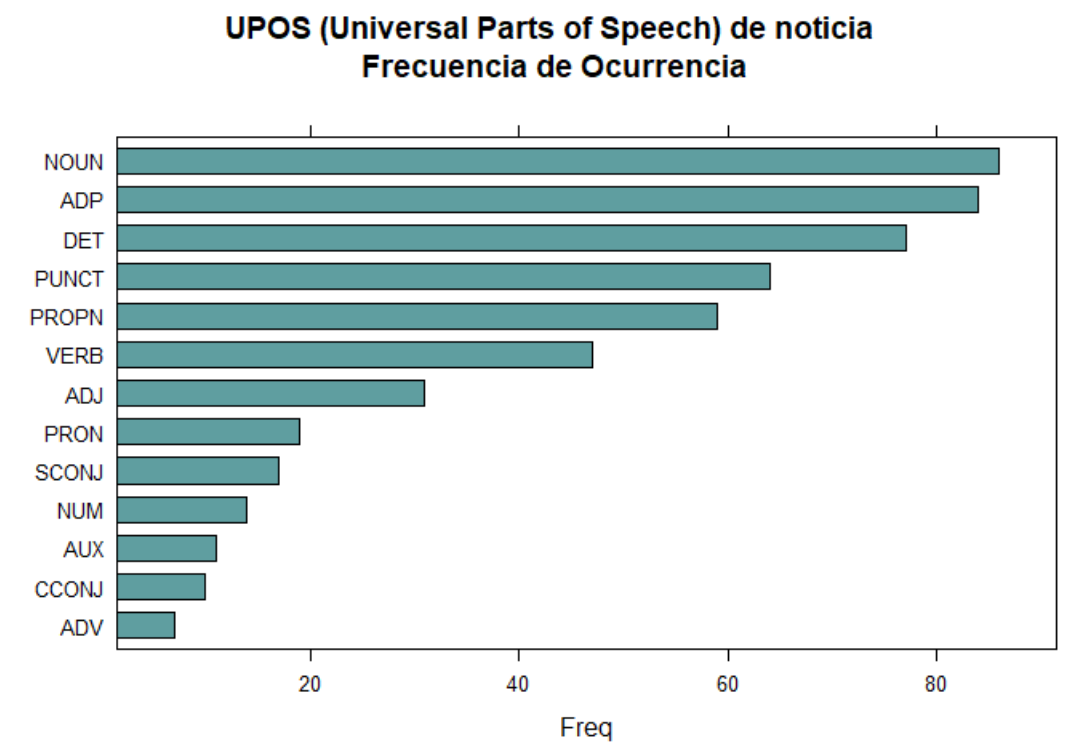
stats <- txt\_freq(noticia\_tag$upos)

stats$key <- factor(stats$key, levels = rev(stats$key))

barchart(key ~ freq, data = stats, col = "cadetblue",

main = "UPOS (Universal Parts of Speech) de noticia\n Frecuencia de Ocurrencia",

xlab = "Freq")



1. Bag of Words

noticia <- readLines("noticias.txt",encoding="UTF-8")

noticia=as.data.frame(noticia)

noticia$ID <- seq.int(nrow(noticia))

noticia <- noticia %>% mutate(texto\_tokenizado = map(.x = noticia,

.f = limpiar\_tokenizar))

noticia\_tidy <- noticia %>% select(-noticia) %>% unnest(cols = c(texto\_tokenizado))

head(noticia\_tidy)

####fecuencia#######

noticia\_tidy %>% group\_by(texto\_tokenizado) %>% count(texto\_tokenizado)%>%

arrange(desc(n))

####frecuencia sin stop words##########

noticia\_tidy %>% group\_by(texto\_tokenizado) %>% filter(!(texto\_tokenizado %in% sw))%>%

count(texto\_tokenizado)%>% arrange(desc(n))

####graficos1####

noticia\_tidy %>% group\_by(texto\_tokenizado) %>% count(texto\_tokenizado)%>%

filter(n >= 10) %>%arrange(desc(n))%>%

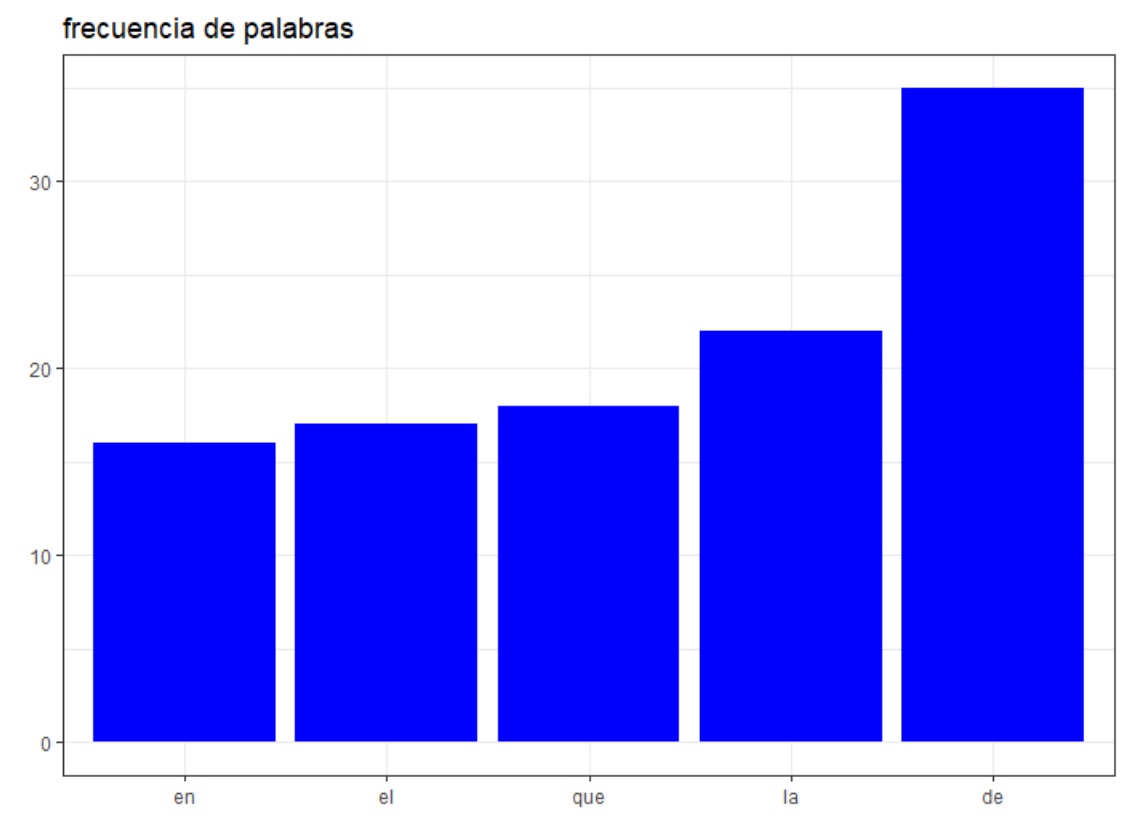
ggplot(aes(x = reorder(texto\_tokenizado,n), y = n)) +

ggtitle("frecuencia de palabras")+

geom\_col(fill="blue") +

theme\_bw() +

labs(y = "", x = "")



####graficos2####

noticia\_tidy %>% group\_by(texto\_tokenizado) %>% filter(!(texto\_tokenizado %in% sw))%>%

count(texto\_tokenizado)%>% filter(n >= 5) %>%arrange(desc(n))%>%

ggplot(aes(x = reorder(texto\_tokenizado,n), y = n)) +

ggtitle("frecuencia de palabras sin stop words")+

geom\_col(fill="red") +

theme\_bw() +

labs(y = "", x = "")



####TDM##############

d\_tdm <- TermDocumentMatrix(d)

m <- as.matrix(d\_tdm)

dim(m)

v <- sort(rowSums(m),decreasing=TRUE)

df <- data.frame(word = names(v),freq=v)

####DTM###########

d\_dtm <- DocumentTermMatrix(d)

m2 <- as.matrix(d\_dtm)

d\_dtm

