Sesión 02

Miguel Angel Espinoza Carranza

library(tokenizers)

options(max.print = 25)

###############tokenizacion por oraciones#######################

base\_analisis <- readLines("noticia2.txt", encoding = "UTF-8")

tokenize\_sentences(base\_analisis)

##############tokenizacion por palarbas#################

tokenize\_words(base\_analisis)

tokenize\_words(base\_analisis[[2]])

##################limpieza##########

####forma 1#######

## install.packages("tm")

library(tm)

corpus <- VCorpus(VectorSource(base\_analisis))

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

d <- tm\_map(d, stripWhitespace)

d <- tm\_map(d, removePunctuation)

d <- tm\_map(d,removeNumbers)

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

base\_analisis[[1]]

######forma 2###########

library(tidyverse)

library(dplyr)

limpiar\_tokenizar <- function(texto){

# El orden de la limpieza no es arbitrario

# Se convierte todo el texto a minúsculas

nuevo\_texto <- tolower(texto)

# Eliminación de páginas web (palabras que empiezan por "http." seguidas

# de cualquier cosa que no sea un espacio)

nuevo\_texto <- str\_replace\_all(nuevo\_texto,"http\\S\*", "")

# Eliminación de signos de puntuación

nuevo\_texto <- str\_replace\_all(nuevo\_texto,"[[:punct:]]", " ")

# Eliminación de números

nuevo\_texto <- str\_replace\_all(nuevo\_texto,"[[:digit:]]", " ")

# Eliminación de espacios en blanco múltiples

nuevo\_texto <- str\_replace\_all(nuevo\_texto,"[\\s]+", " ")

# Eliminacion de tildes

nuevo\_texto <- chartr('áéíóúñ','aeioun',nuevo\_texto)

# Tokenización por palabras individuales

nuevo\_texto <- str\_split(nuevo\_texto, " ")[[1]]

# Eliminación de tokens con una longitud < 2

nuevo\_texto <- keep(.x = nuevo\_texto, .p = function(x){str\_length(x) > 1})

return(nuevo\_texto)

}

#test = "El presidente de Perú, Martin VIZCARRA nació en 1990 en cañete a las 12:00 del mediodia

# según e'l portal l'limpieza de6 TEXTO https://t.co/rnHPgyhx4Z usuario tRodrigo #text"

test = base\_analisis[[1]]

limpiar\_tokenizar(texto = test)

base\_analisis=as.data.frame(base\_analisis)

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

.f = limpiar\_tokenizar))

base\_analisis %>% select(texto\_tokenizado) %>% head()

base\_analisis %>% slice(1) %>% select(texto\_tokenizado) %>% pull()

base\_analisis$ID <- seq.int(nrow(base\_analisis))

####################Stop Words#######################

##############ingles#######################

library(stopwords)

tokenize\_words(base\_analisis[[1]])

tokenize\_words(base\_analisis[[1]], stopwords = stopwords::stopwords("en"))

# length(stopwords(source = "smart"))

# length(stopwords(source = "snowball"))

# length(stopwords(source = "stopwords-iso"))

# stopwords(language = "en", source = "smart")

###########español#####################

####palabras por defecto de la libreria###

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############

base\_analisis\_tidy <- base\_analisis %>% dplyr::select(-base\_analisis) %>% unnest()

head(base\_analisis\_tidy)

# Se filtran las stopwords

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

head(base\_analisis\_tidy)

####lemtizacion\_español########

#############stemming y lemmantizacion########

library(SnowballC)

## install.packages("corpus")

library(corpus)

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

stringsAsFactors = FALSE ,encoding="UTF-8")

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

}

##text <- "los presidentes de las regiones tuvieron que aumentar los dias de cuarentena debido al incremento de nuevos casos"

text = (base\_analisis[[1]])

text\_tokens(text, stemmer = stem\_liste)

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

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

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

base\_analisis\_tidy

######tagging#################

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

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

### TAGGING ejemplo Noticias ###

base\_analisis\_02 <- udpipe\_annotate(ud\_model, x = base\_analisis$base\_analisis, doc\_id = base\_analisis$ID)

base\_analisis\_02 <- as.data.frame(base\_analisis\_02)

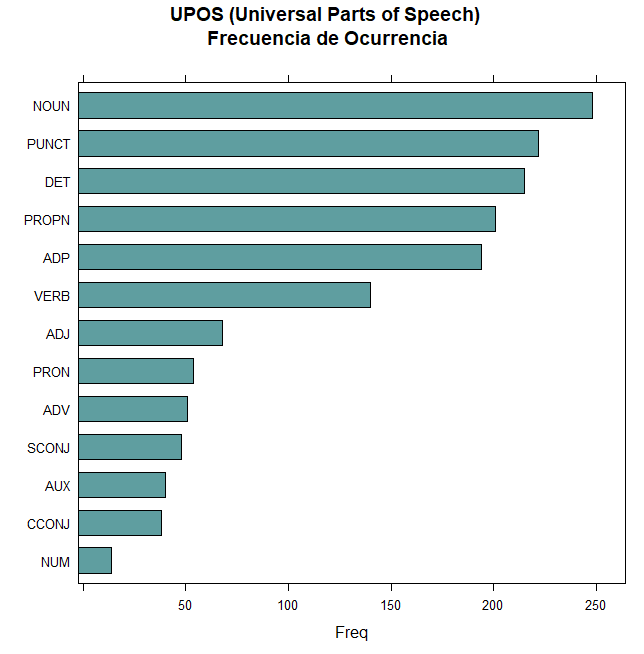
stats <- txt\_freq(base\_analisis\_02$upos)

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

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

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

xlab = "Freq")



#######Bag of words##############

###nuevo documento#####

base\_analisis\_03 <- readLines("noticia2.txt", encoding = "UTF-8")

base\_analisis\_03=as.data.frame(base\_analisis\_03)

base\_analisis\_03$ID <- seq.int(nrow(base\_analisis\_03))

base\_analisis\_03 <- base\_analisis\_03 %>% mutate(texto\_tokenizado = map(.x = base\_analisis\_03,

.f = limpiar\_tokenizar))

base\_analisis\_03\_tidy <- base\_analisis\_03 %>% select(-base\_analisis\_03) %>% unnest()

head(base\_analisis\_03\_tidy)

#b <- udpipe\_annotate(ud\_model, x = base\_analisis\_03\_tidy$texto\_tokenizado)

#b <- as.data.frame(b)

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

base\_analisis\_03\_tidy %>% group\_by(texto\_tokenizado) %>% count(texto\_tokenizado)%>%

arrange(desc(n))

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

base\_analisis\_03\_tidy %>% group\_by(texto\_tokenizado) %>% filter(!(texto\_tokenizado %in% sw))%>%

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

####graficos1####

base\_analisis\_03\_tidy %>% group\_by(texto\_tokenizado) %>% count(texto\_tokenizado)%>%

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

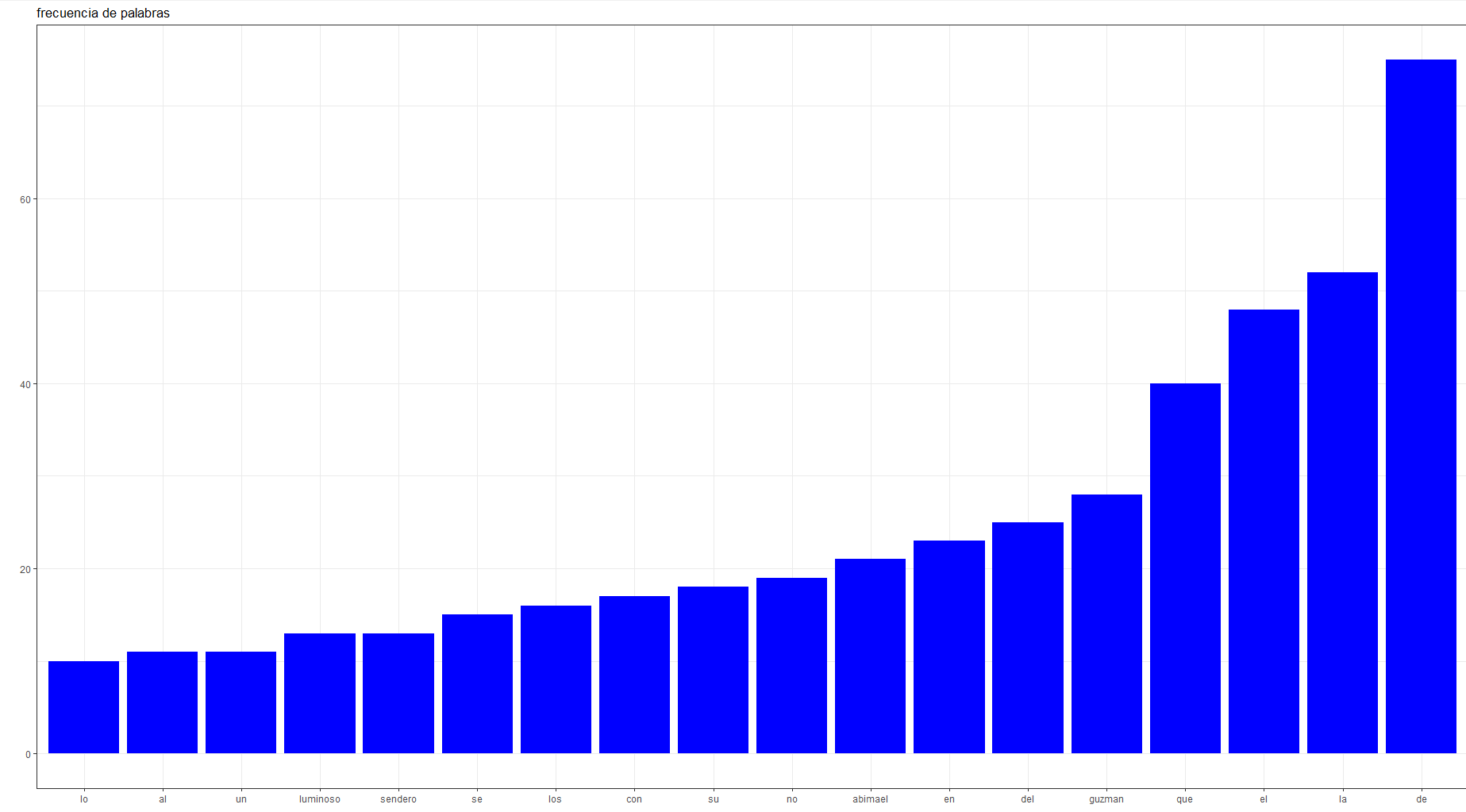
ggplot(aes(x = reorder(texto\_tokenizado,n), y = n)) +

ggtitle("frecuencia de palabras")+

geom\_col(fill="blue") +

theme\_bw() +

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



####graficos2####

base\_analisis\_03\_tidy %>% group\_by(texto\_tokenizado) %>% filter(!(texto\_tokenizado %in% sw))%>%

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

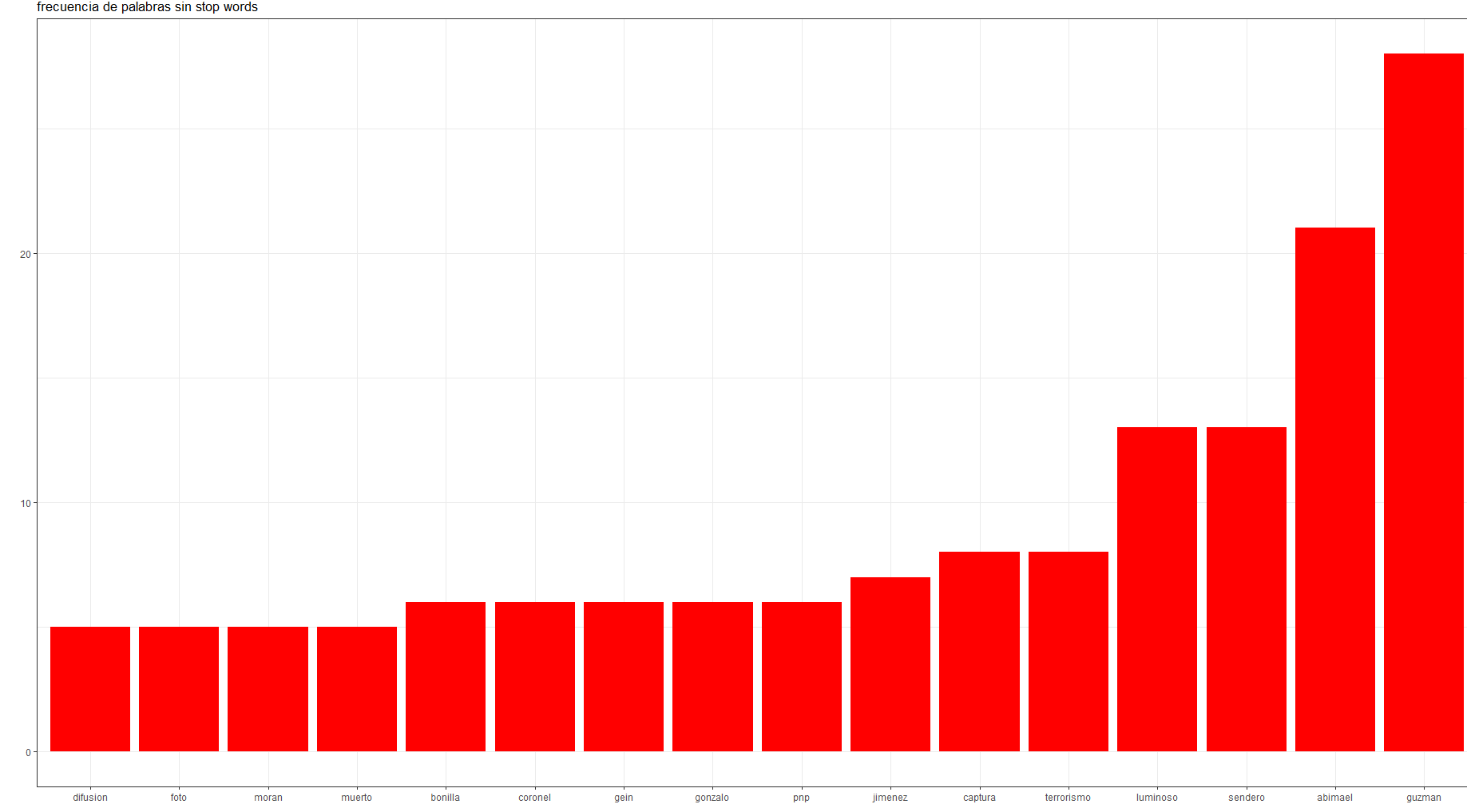
ggplot(aes(x = reorder(texto\_tokenizado,n), y = n)) +

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)