TextMining - Discursos Presidenciales

alpoza

6 de septiembre de 2017

## Objetivo

Existen artículos donde se realizan analíticas de los discursos presidenciales americanos, pero no encontré nada sobre los discursos de España. Así que el objetivo de este artículo es hacer una introducción a Text Mining sobre los **discursos inagurales de los presidentes** en todas las legislaturas de España.

## Librerías necesarias:

library(tm) #  
library(wordcloud)  
library(plyr)  
library(tidyverse)  
library(tidytext)  
library(SnowballC)  
library(ggrepel)  
library(stringr)

## Lectura de los documentos

La mejor forma que he encontrado de cargar todos los documentos es utilizando la función VCorpus del paquete tm. Los ficheros se encuentran en la carpeta data y los descargue manualmente desde <http://www.lamoncloa.gob.es/presidente/presidentes/Paginas/index.aspx>. TODO: automatizar el paso de descarga.

folderdir="data/"  
corpus\_raw = VCorpus(DirSource(folderdir, encoding = "UTF-8"), readerControl=list(reader=readPlain,language="es"))

Limpiamos:

add.stops=c("vez", "sino", "cada", "ello", "así", "sólo", "que", "que,", "señorías", "señorías,","gobierno","política","españa","españoles","país","ser","hacia","años","debe","cualquier","año","manera","todas","mayor","parte","presidenta","ustedes","vista","señora","hecho","sus")  
#add.stops=c("kk")  
corpus\_clean = corpus\_raw  
  
corpus\_clean <- tm\_map(corpus\_clean, content\_transformer(gsub),  
 pattern = "SS.SS.", replacement = "Sus Señorías")  
  
toSpace = content\_transformer(function(x, pattern) { return (gsub(pattern, " ", x))})  
corpus\_clean <- tm\_map(corpus\_clean, toSpace, "-")  
corpus\_clean <- tm\_map(corpus\_clean, toSpace, "’")  
corpus\_clean <- tm\_map(corpus\_clean, toSpace, "‘")  
corpus\_clean <- tm\_map(corpus\_clean, toSpace, "”")  
corpus\_clean <- tm\_map(corpus\_clean, toSpace, "“")  
  
corpus\_clean = tm\_map (corpus\_clean, content\_transformer(tolower))  
corpus\_clean = tm\_map (corpus\_clean, removePunctuation)  
corpus\_clean = tm\_map (corpus\_clean, stripWhitespace)  
corpus\_clean = tm\_map (corpus\_clean, removeWords, stopwords("spanish"))  
corpus\_clean = tm\_map (corpus\_clean, removeWords, add.stops)  
#corpus\_clean = tm\_map (corpus\_clean, stemDocument, language="spanish")  
  
corpus.control=list(stopwords=c(stopwords("spanish"),add.stops), removeNumbers=FALSE, removePunctuation=FALSE) #, tokenize=my\_tokenizer  
corpus\_clean.dtm<-DocumentTermMatrix(corpus\_clean, control=corpus.control)  
notSparse = removeSparseTerms(corpus\_clean.dtm, 0.99)  
corpus\_clean\_tidy = tidy(corpus\_clean.dtm)  
  
#creamos un docid para poder filtrar  
names = names(corpus\_clean\_tidy)  
corpus\_clean\_tidy = corpus\_clean\_tidy %>%   
 mutate(docid = str\_extract(document, "^..")) %>%   
 select(docid, names)  
  
findFreqTerms(corpus\_clean.dtm, lowfreq=100)

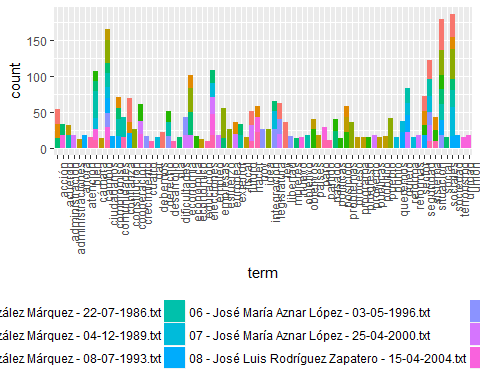
## [1] "acción" "acuerdo" "administración" "ahora"   
## [5] "cámara" "ciudadanos" "comunidades" "confianza"   
## [9] "constitución" "cooperación" "crecimiento" "desarrollo"   
## [13] "economía" "económica" "económico" "empleo"   
## [17] "esfuerzo" "española" "europea" "exterior"   
## [21] "futuro" "hacer" "hoy" "legislatura"   
## [25] "ley" "libertad" "medidas" "objetivo"   
## [29] "países" "políticas" "problemas" "proceso"   
## [33] "programa" "público" "puede" "quiero"   
## [37] "reforma" "seguridad" "sistema" "situación"   
## [41] "social" "sociales" "sociedad" "tiempo"

Mostramos el 10 terminos más frecuentes por cada discurso.

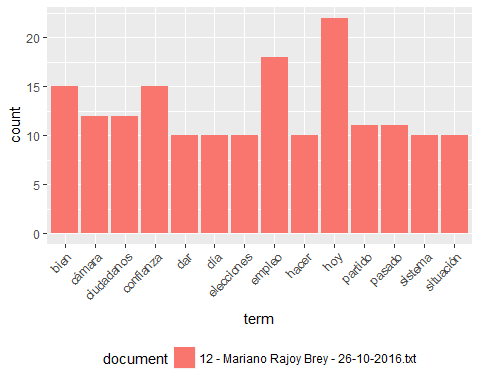
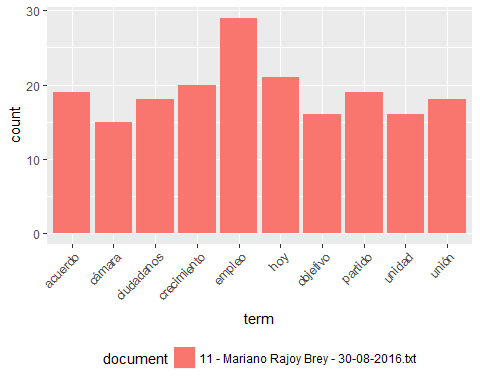
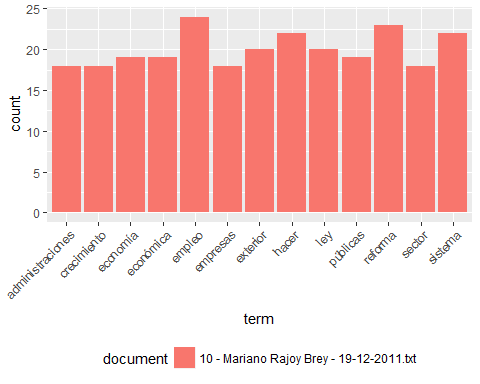
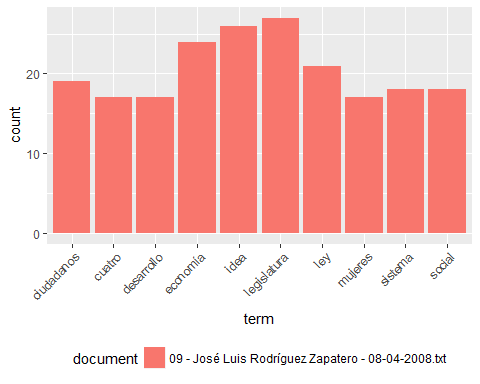
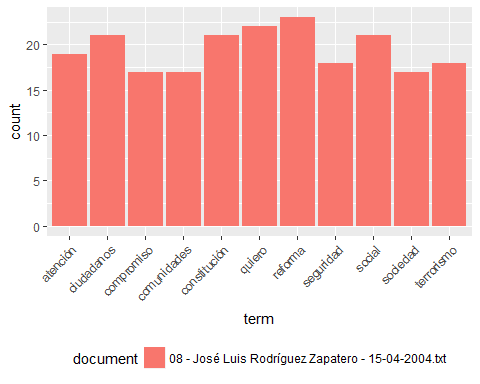
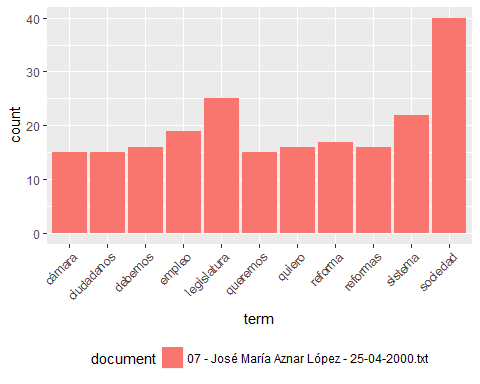
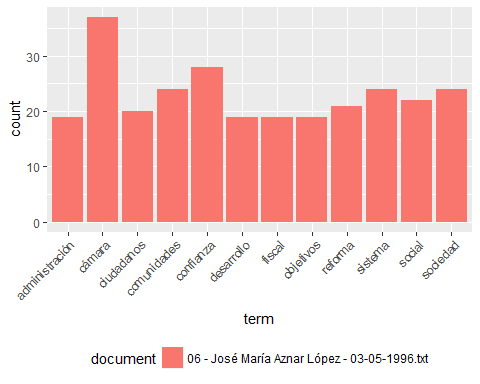
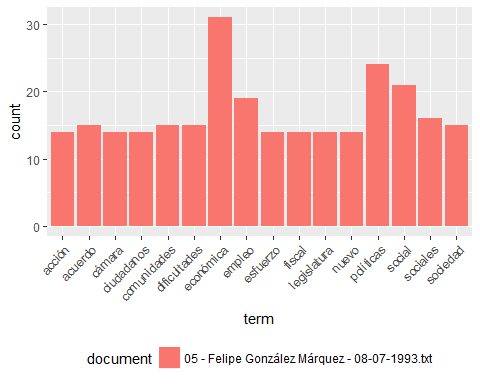
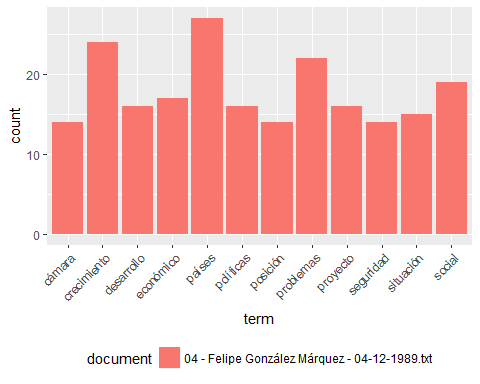
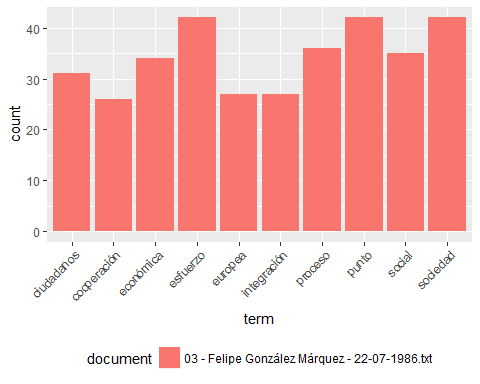
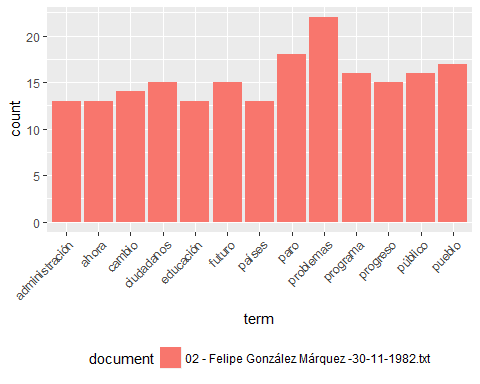
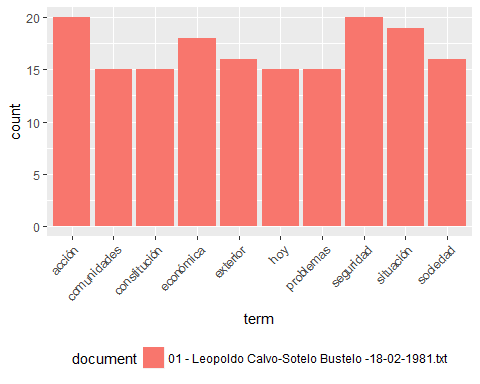
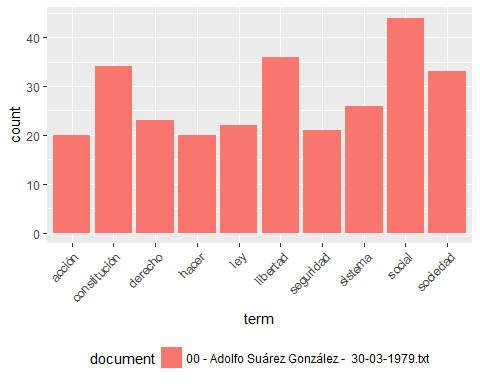
corpus\_clean\_tidy\_top = corpus\_clean\_tidy %>%   
 group\_by(document) %>%   
 top\_n(10, count) %>%   
 ungroup()   
   
corpus\_clean\_tidy\_top %>%   
 ggplot(aes(term,count)) +   
 geom\_bar(stat="identity", aes(fill=document, alpha=1/2)) +   
 facet\_wrap(~document, scales = "free", ncol = 1) +   
 theme(legend.position="none")



corpus\_clean\_tidy\_top %>%   
 ggplot(aes(term,count)) +   
 geom\_bar(stat="identity", aes(fill=document)) +   
 theme(axis.text.x = element\_text(angle = 90, hjust = 1)) +  
 theme(legend.position="bottom")

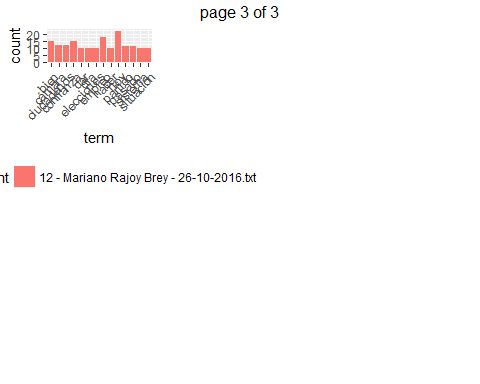
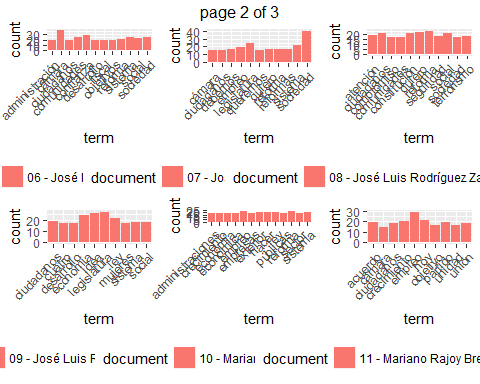


documents = corpus\_clean\_tidy\_top %>% select(document) %>% unique()  
plots <- corpus\_clean\_tidy\_top %>%  
 split(.$document) %>%  
 map(~ ggplot(.x, aes(term,count)) +   
 geom\_bar(stat="identity", aes(fill=document)) +   
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 theme(legend.position="bottom")  
 )  
  
walk(plots, print)



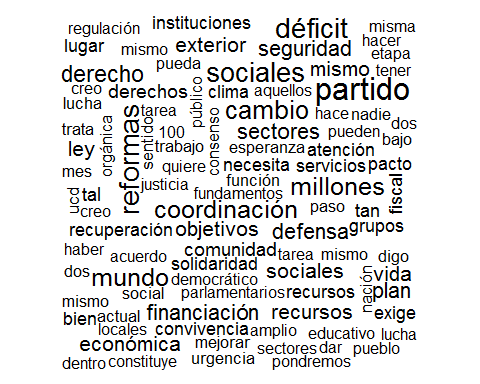
si lo queremos más compacto

library(gridExtra)  
marrangeGrob(plots, nrow=2, ncol=3)



Como nube de etiquetas:

wordcloud(corpus\_clean\_tidy$term,corpus\_clean\_tidy$count,min.freq=6)



TF-IDF:

# create tf-idf matrix