Tarea extra MH BAIN 25/05/2022

Hector Asorey de Pablos

25/05/2022

### Por favor asegúrate de tener **tu** ruta al espacio de trabajo R en el chunk siguiente

### Es buena práctica cargar al inicio las librerías necesarias para ejecutar el resto del markdown

library(data.table)  
library(stringr)  
library(quanteda)  
library(quanteda.textplots)  
library(quanteda.textstats)  
library(quanteda.textmodels)  
library(igraph)  
library(ggplot2)  
library(dplyr)

# Objetivos de este documento RMarkdown y de la tarea

El objetivo de este documento es concretar un trabajo muy especial, como es el de decidir la MH de la asignatura BAIN entre un selecto grupo de candidatos. Esta tarea espero esté bien diseñada para *no debe ser especialmente trabajosa en tiempo*, sí en originalidad y rapidez con los conceptos y metodologías Text Mining y SNA.

Es inevitable extenderse sobre el tema del que trata la tarea. Y es sobre los “hashtag games” del conjunto de trolls que se han analizado en la parte de Text Mining. De todas las categorías contenidas en este dataset esta es la más peculiar.

El dataset fue recopilado y clasificado por dos profesores de la Clemson University: Darren Linvill and Patrick Warren. Estos dos profesores publicaron un paper (el enlace muestra el paper provisional, desconozco si ha habido alguna edición o publicación adicional):

<https://pwarren.people.clemson.edu/Linvill_Warren_TrollFactory.pdf>

Hay que señalar la diferencia que establecen entre “account category” y “handle”. Este último concepto es equivalente al de “usuario twitter”. La primera pregunta es ¿un mismo handle pertenece a varias “account category”?

Después de una descripción de los métodos de clasificación (sin ninguna referencia a modelos de clasificación tipo “Machine Learning”) estos autores concluyen que:

“With the exception of the Fearmonger category, handles were consist and did not switch between categories.”

La categoría “Hashtag Gamer” está descrita del siguiente modo:

Hashtag Gamer (110 handles, 216,895 tweets, M = 1955.31, SD = 3176.10). These handles are dedicated almost entirely to playing hashtag games, a popular word game played on Twitter. Users add a hashtag to a tweet (e.g., #ThingsILearnedFromCartoons) and then answer the implied question (Haskell, 2015). These handles also posted tweets that seemed organizational regarding these games, e.g. @AmandaVGreen’s quote tweet, August 31, 2016, “15 minutes till we play @TheHashtagGame with @HashtagRoundup & @HashtagZoo! Who’s ready to #hashtag!”.

Many of these tweets were mundane, including @DonnieLMiller, April 12, 2017, “#OffendEveryoneIn4Words fart in your face.” Others, however, often using the same hashtag, were socially divisive, including @DonnieLMiller, April 12, 2017: “#OffendEveryoneIn4Words undocumented immigrants are ILLEGALS.” Many tweets from Hashtag Gamers were overtly political, e.g. @LoraGreeen, July 11, 2015, “#WasteAMillionIn3Words Donate to #Hillary”.

While many tweets shared themes seen in the Right Troll category, Left Troll themes also appeared, e.g., @LoraGreeen, January 25, 2016, “#ItsSoWhiteOutsideThat Donald Trump thought it was a meeting of his followers.”

Alguna referencia adicional sobre los hashtag games: <https://www.thundertech.com/blog-news/march-2015/hashtag-games-the-hunger-games-of-social-media>

Este tema de hashtag games es relativamente poco conocido (desde luego para mi como usuario absolutamente remoto de las redes sociales) pero resulta interesante. En resumen: se trata de encontrar un tag “simpático” o “resultón” que se propague con facilidad, complementándolo con contenido que va desde lo “graciosillo” a lo directamente insultante.

Estaría genial poder hacer esa clasificación por contenido, pero desde luego se me antoja extraordinariamente difícil y desde luego costoso en tiempo. Pero sí que me parece abordable extraer los “hashtag\_games” del conjunto de trolls, prepararlo de forma rápida y crear un grafo con el que poder abordar una serie de preguntas (entre otras muchas que estoy seguro que se os ocurrirán en el proceso de análisis).

Las preguntas para guiar el análisis (la extracción de conocimiento que es de lo que trata este trabajo) al final, después de cargar y dejar el dataset “compacto” para análisis.

POR FAVOR sed conscientes que no se ha hecho ninguna limpieza de caracteres especiales para dejar los hashtags tal cual aparecen (p.ej. #2016In4Words)

# 1. Carga del conjunto base de trolls

Debemos cargar el conjunto de trolls con el texto sin filtrar para poder conservar los tags (todo lo que comienza por #)

load('todos\_los\_trolls.rda')  
dim(trollstot)

## [1] 2925313 16

summary(trollstot)

## external\_author\_id author content region   
## Length:2925313 Length:2925313 Length:2925313 Length:2925313   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## language publish\_date harvested\_date following   
## Length:2925313 Length:2925313 Length:2925313 Min. : -1   
## Class :character Class :character Class :character 1st Qu.: 337   
## Mode :character Mode :character Mode :character Median : 1524   
## Mean : 3468   
## 3rd Qu.: 4769   
## Max. :76210   
##   
## followers updates post\_type account\_type   
## Min. : -1 Min. : -1 :1645622 Right :705987   
## 1st Qu.: 327 1st Qu.: 1817 QUOTE\_TWEET: 39938 Russian :695625   
## Median : 1297 Median : 4404 RETWEET :1239753 local :459220   
## Mean : 7098 Mean : 10572 Left :421597   
## 3rd Qu.: 11162 3rd Qu.: 12440 Hashtager:241786   
## Max. :251276 Max. :166113 news :139006   
## (Other) :262092   
## retweet account\_category new\_june\_2018 alt\_external\_id   
## 0:1645622 NonEnglish :811299 0:2312723 Length:2925313   
## 1:1279691 RightTroll :705987 1: 612590 Class :character   
## NewsFeed :598226 Mode :character   
## LeftTroll :421597   
## HashtagGamer:241786   
## Commercial :121904   
## (Other) : 24514

Extraemos los hashtag\_games

games <- trollstot[trollstot$account\_category == "HashtagGamer", ]  
  
dim(games)

## [1] 241786 16

length(unique(games$author))

## [1] 112

table(games$account\_category, games$language)

##   
## Albanian Arabic Catalan Croatian Czech Danish Dutch English  
## Commercial 0 0 0 0 0 0 0 0  
## Fearmonger 0 0 0 0 0 0 0 0  
## HashtagGamer 133 235 121 130 42 31 247 236092  
## LeftTroll 0 0 0 0 0 0 0 0  
## NewsFeed 0 0 0 0 0 0 0 0  
## NonEnglish 0 0 0 0 0 0 0 0  
## RightTroll 0 0 0 0 0 0 0 0  
## Unknown 0 0 0 0 0 0 0 0  
##   
## Estonian Farsi (Persian) Finnish French German Hindi Hungarian  
## Commercial 0 0 0 0 0 0 0  
## Fearmonger 0 0 0 0 0 0 0  
## HashtagGamer 224 460 114 185 241 1 17  
## LeftTroll 0 0 0 0 0 0 0  
## NewsFeed 0 0 0 0 0 0 0  
## NonEnglish 0 0 0 0 0 0 0  
## RightTroll 0 0 0 0 0 0 0  
## Unknown 0 0 0 0 0 0 0  
##   
## Icelandic Indonesian Italian Japanese Korean Kurdish  
## Commercial 0 0 0 0 0 0  
## Fearmonger 0 0 0 0 0 0  
## HashtagGamer 143 54 850 8 3 29  
## LeftTroll 0 0 0 0 0 0  
## NewsFeed 0 0 0 0 0 0  
## NonEnglish 0 0 0 0 0 0  
## RightTroll 0 0 0 0 0 0  
## Unknown 0 0 0 0 0 0  
##   
## LANGUAGE UNDEFINED Latvian Lithuanian Malay Norwegian Polish  
## Commercial 0 0 0 0 0 0  
## Fearmonger 0 0 0 0 0 0  
## HashtagGamer 15 88 113 46 531 87  
## LeftTroll 0 0 0 0 0 0  
## NewsFeed 0 0 0 0 0 0  
## NonEnglish 0 0 0 0 0 0  
## RightTroll 0 0 0 0 0 0  
## Unknown 0 0 0 0 0 0  
##   
## Portuguese Pushto Romanian Russian Serbian Slovak Slovenian  
## Commercial 0 0 0 0 0 0 0  
## Fearmonger 0 0 0 0 0 0 0  
## HashtagGamer 20 97 429 21 2 51 4  
## LeftTroll 0 0 0 0 0 0 0  
## NewsFeed 0 0 0 0 0 0 0  
## NonEnglish 0 0 0 0 0 0 0  
## RightTroll 0 0 0 0 0 0 0  
## Unknown 0 0 0 0 0 0 0  
##   
## Somali Spanish Swedish Tagalog (Filipino) Thai  
## Commercial 0 0 0 0 0  
## Fearmonger 0 0 0 0 0  
## HashtagGamer 76 346 80 109 4  
## LeftTroll 0 0 0 0 0  
## NewsFeed 0 0 0 0 0  
## NonEnglish 0 0 0 0 0  
## RightTroll 0 0 0 0 0  
## Unknown 0 0 0 0 0  
##   
## Traditional Chinese Turkish Ukrainian Urdu Uzbek Vietnamese  
## Commercial 0 0 0 0 0 0  
## Fearmonger 0 0 0 0 0 0  
## HashtagGamer 2 43 1 12 120 129  
## LeftTroll 0 0 0 0 0 0  
## NewsFeed 0 0 0 0 0 0  
## NonEnglish 0 0 0 0 0 0  
## RightTroll 0 0 0 0 0 0  
## Unknown 0 0 0 0 0 0

Nos quedamos solo con los de inglés:

games <- games[games$language == "English", ]  
dim(games)

## [1] 236092 16

save(games,   
 file = "hashtag\_games.rda")

# Scope del trabajo

1. Tienes nodos (“handles” o “author”) como accounts (¿una o varias personas? Nunca lo sabremos), o hastags.
2. Tienes conexiones o enlaces (retweets, hashtags compartidos).

Por ejemplo, este hashtag es compartido por muchos handles: #BritainInOut #GoodbyeUK #RemainINEU #EUref

MUUUUCHO CUIDADO CON MAYUSCULAS Y MINUSCULAS EN HASHTAGS. Por ejemplo: #GrowingUpWithObama y #growingupwithobama

(este es tambien un ejemplo de hashtag compartido por handles)

1. Puedes elegir entre varias posibilidades de fuerza del enlace:
   * número de apariciones del hashtag
   * número de retweets del hashtag

PREGUNTAS

* ¿Es posible explorar una configuración de la difusión de los hashtags en un grafo -sea de lo que sea (hashtags o handles)?
* Si fuera así, ¿se puede encontrar la cadena más larga de conexiones o usos de hashtags? ¿Cuáles fueron?
* ¿Hay algún nodo del grafo con especial relevancia? (generador? retweeteador? un hashtag concreto?)

Cargamos el objeto generado previamente

load("hashtag\_games.rda")

myDf <- as.data.table(games)

Limpiamos hipervínculos

filtroUrl <- "http[s]?://(?:[a-zA-Z]|[0-9]|[$-\_@.&+]|[!\*\\(\\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+"  
filtroNuevo <- "http[s]?:"  
filtroNuevo2 <- "http[s]?"  
filtroNuevo3 <- " ?(f|ht)tp(s?)://(.\*)[.][a-z]+"  
  
myDf$content <- str\_replace\_all(myDf$content, pattern = filtroUrl, "")  
myDf$content <- str\_replace\_all(myDf$content, pattern = filtroNuevo, "")  
myDf$content <- str\_replace\_all(myDf$content, pattern = filtroNuevo2, "")  
myDf$content <- str\_replace\_all(myDf$content, pattern = filtroNuevo3, "")

Limpiamos de emoticonos

myDf$content <- gsub("[^\x01-\x7F]", "", myDf$content)

Convertimos a fecha el campo “publish date”

myDf$publish\_date <- as.Date(myDf$publish\_date, "%m/%d/%Y")

Eliminamos algunos campos del dataframe

myDf2 <- (myDf[, c(2:14)])

myDf2 <- myDf2[, c(1:5, 10:12)]

firstCorpus <- corpus(myDf2$content)  
docvars(firstCorpus, "date") <- myDf2$publish\_date  
docvars(firstCorpus, "author") <- myDf2$author  
firstTokens <- tokens(firstCorpus, remove\_punct = TRUE)

Vamos a ver los 100 primeros términos más repetidos.

firstDfm <- dfm(firstTokens)  
topfeatures(firstDfm, 100)

## the to a   
## 73377 52067 50407   
## i you and   
## 47918 33304 32386   
## of my in   
## 30960 27471 27080   
## is for it   
## 23274 19821 16960   
## your that on   
## 16263 16226 15945   
## me with be   
## 14178 13702 12941   
## have not are   
## 11163 10950 9988   
## i'm all this   
## 9692 9367 9102   
## @midnight at like   
## 8584 8492 8246   
## don't when but   
## 8168 7913 7352   
## get just up   
## 7304 7177 7066   
## people so they   
## 6958 6713 6687   
## if was do   
## 6675 6536 6378   
## out can no   
## 6366 6351 6250   
## what it's we   
## 5987 5951 5918   
## about from one   
## 5823 5807 5792   
## will as how   
## 5627 5397 5394   
## who an by   
## 5333 5329 5321   
## or would make   
## 5088 4897 4803   
## more love trump   
## 4638 4503 4219   
## go time he   
## 4112 4032 4018   
## them know because   
## 3856 3757 3722   
## day never want   
## 3679 3533 3529   
## #todolistbeforechristmas good now   
## 3454 3396 3388   
## than their new   
## 3365 3348 3308   
## you're only life   
## 3293 3193 3151   
## too there his   
## 3146 3118 3099   
## think some can't   
## 3084 3001 2992   
## #thingsyoucantignore our her   
## 2965 2889 2866   
## she really back   
## 2856 2854 2796   
## best being take   
## 2767 2676 2639   
## got has #2016in4words   
## 2636 2628 2626   
## why #mustbebanned #igetdepressedwhen   
## 2620 2611 2578   
## need   
## 2575

Salen muchas preposiciones y términos que no aportan significado alguno. Por tanto, vamos a aplicar stopwords del diccionario nltk, a ver si nos filtra todo, evitando así tener que crearnos nuestras stopwords.

secondDfm <- dfm\_remove(firstDfm, stopwords("english", source = "nltk"))  
topfeatures(secondDfm, 100)

## i'm @midnight   
## 9692 8584   
## like get   
## 8246 7304   
## people one   
## 6958 5792   
## would make   
## 4897 4803   
## love trump   
## 4503 4219   
## go time   
## 4112 4032   
## know day   
## 3757 3679   
## never want   
## 3533 3529   
## #todolistbeforechristmas good   
## 3454 3396   
## new life   
## 3308 3151   
## think can't   
## 3084 2992   
## #thingsyoucantignore really   
## 2965 2854   
## back best   
## 2796 2767   
## take got   
## 2639 2636   
## #2016in4words #mustbebanned   
## 2626 2611   
## #igetdepressedwhen need   
## 2578 2575   
## see still   
## 2544 2537   
## #ihatepokemongobecause every   
## 2533 2515   
## i'd 2   
## 2491 2466   
## us hashtag   
## 2419 2392   
## say game   
## 2355 2350   
## work right   
## 2262 2262   
## always going   
## 2261 2240   
## much #alternativeacronyminterpretations   
## 2227 2182   
## #istartcryingwhen look   
## 2158 2151   
## #giftideasforpoliticians world   
## 2130 2107   
## 1 #toavoidworki   
## 2106 2073   
## way first   
## 2065 2040   
## someone even   
## 2027 2024   
## twitter #rejecteddebatetopics   
## 2022 2017   
## better tell   
## 2013 2008   
## could hate   
## 1992 1992   
## man house   
## 1991 1968   
## play #secondhandgifts   
## 1954 1948   
## president white   
## 1916 1872   
## #donttellanyonebut #tofeelbetteri   
## 1872 1865   
## #christmasaftermath last   
## 1841 1832   
## stop something   
## 1818 1804   
## things #myolympicsportwouldbe   
## 1797 1795   
## give #obamaswishlist   
## 1790 1790   
## find great   
## 1776 1751   
## everyone let   
## 1750 1726   
## christmas eat   
## 1723 1702   
## #betteralternativetodebates year   
## 1675 1668   
## hillary #childrenthinkthat   
## 1655 1655   
## come that's   
## 1642 1621   
## without #howtoloseyourjob   
## 1621 1610   
## oh little   
## 1607 1598   
## real put   
## 1592 1588   
## #valentinesdayin3words money   
## 1582 1577

Podemos ver que a pesar de ser tweets “de juego”, sigue estando el tema político de fondo, estando los términos “trump” y “hillary” muy presentes, además de hashtagas evidentemente políticos, como “obamaswishlist”.

También vemos que salen número o contracciones en estos términos, así que, para obtener términos más representativos, vamos a hacer nuestras propias stopwords.

myStopwords <- c(stopwords(language = "en", source = "nltk"),  
 "put",  
 "i'm",  
 "can",  
 "we're",  
 "that's",  
 "even",  
 "i'd",  
 "can't",  
 "still",  
 "get",  
 "would",  
 "oh",  
 "ever",  
 "1",  
 "2",  
 "3",  
 "4",  
 "5",  
 "6",  
 "7",  
 "8",  
 "9")

Vamos a filtrar más mediante el uso de unas stopword propias, a ver si obtenemos términos más significativos.

thirdDfm <- dfm\_remove(firstDfm, myStopwords)  
topfeatures(thirdDfm, 100)

## @midnight like   
## 8584 8246   
## people one   
## 6958 5792   
## make love   
## 4803 4503   
## trump go   
## 4219 4112   
## time know   
## 4032 3757   
## day never   
## 3679 3533   
## want #todolistbeforechristmas   
## 3529 3454   
## good new   
## 3396 3308   
## life think   
## 3151 3084   
## #thingsyoucantignore really   
## 2965 2854   
## back best   
## 2796 2767   
## take got   
## 2639 2636   
## #2016in4words #mustbebanned   
## 2626 2611   
## #igetdepressedwhen need   
## 2578 2575   
## see #ihatepokemongobecause   
## 2544 2533   
## every us   
## 2515 2419   
## hashtag say   
## 2392 2355   
## game work   
## 2350 2262   
## right always   
## 2262 2261   
## going much   
## 2240 2227   
## #alternativeacronyminterpretations #istartcryingwhen   
## 2182 2158   
## look #giftideasforpoliticians   
## 2151 2130   
## world #toavoidworki   
## 2107 2073   
## way first   
## 2065 2040   
## someone twitter   
## 2027 2022   
## #rejecteddebatetopics better   
## 2017 2013   
## tell could   
## 2008 1992   
## hate man   
## 1992 1991   
## house play   
## 1968 1954   
## #secondhandgifts president   
## 1948 1916   
## white #donttellanyonebut   
## 1872 1872   
## #tofeelbetteri #christmasaftermath   
## 1865 1841   
## last stop   
## 1832 1818   
## something things   
## 1804 1797   
## #myolympicsportwouldbe give   
## 1795 1790   
## #obamaswishlist find   
## 1790 1776   
## great everyone   
## 1751 1750   
## let christmas   
## 1726 1723   
## eat #betteralternativetodebates   
## 1702 1675   
## year hillary   
## 1668 1655   
## #childrenthinkthat come   
## 1655 1642   
## without #howtoloseyourjob   
## 1621 1610   
## little real   
## 1598 1592   
## #valentinesdayin3words money   
## 1582 1577   
## #probabletrumpstweets friends   
## 1577 1573   
## use happy   
## 1571 1553   
## #wheniwasyoung show   
## 1540 1520   
## #thingsthatshouldbecensored #ineedalawyerbecause   
## 1520 1520   
## keep #obamanextjob   
## 1518 1507   
## watch said   
## 1504 1502

100 términos más repetidos, tanto siendo hashtags como si no.

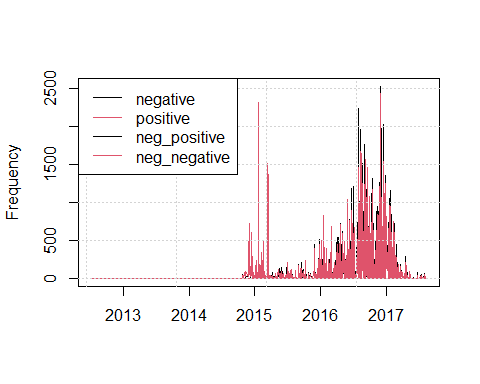
Ahora haremos un análisis de sentimientos.

tokens2 <- tokens\_select(firstTokens, pattern = myStopwords, selection = "remove")

toks\_dictionary <- tokens\_lookup(tokens2, data\_dictionary\_LSD2015)

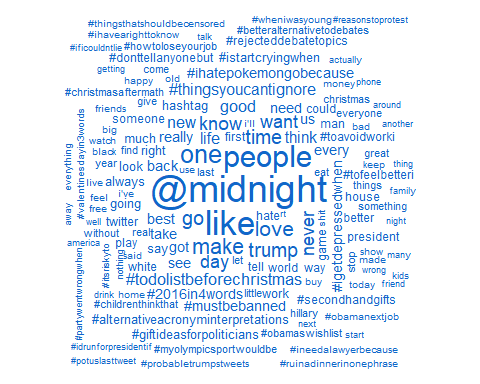
dfmat\_twitter <- dfm(toks\_dictionary) %>%  
 dfm\_group(groups = date)

matplot(dfmat\_twitter$date, dfmat\_twitter, type = "l", lty = 1, col = 1:2,  
 ylab = "Frequency", xlab = "")  
grid()  
legend("topleft", col = 1:2, legend = colnames(dfmat\_twitter), lty = 1, bg = "white")



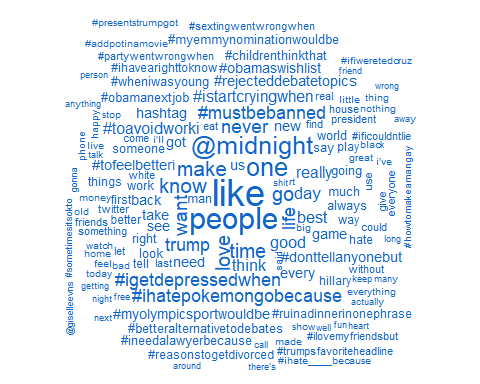
Este gráfico nos muestra que en 2016 hubo un aumento siginificativo de la negatividad en los tweets, que duraría hasta 2017, coincidiendo con el periodo electoral estadounidense.

textplot\_wordcloud(thirdDfm, max\_words = 150, color = "#1067c9")

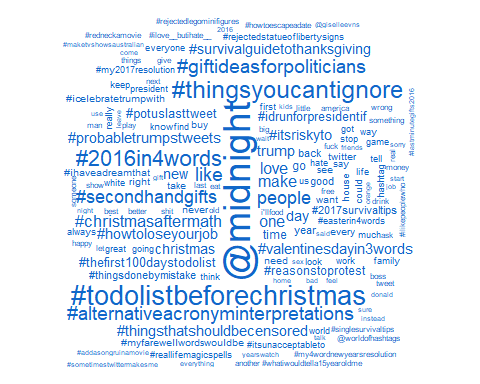


Términos a destacar que aparecen en esta nube… Lo primero, @midnight, que es el término que más se repite… @midnight es un “late night show” cómico, en ComedyCentral. Seguramente, fuera el que empezará esto de poner tweets con un hashtag y seguir la broma. Lo segundo, los hashtags políticos… Temas como “2016in4words”, “probabletrumptweets”, “giftideasforpoliticians”. “betteralternativetodebates”… son temas en los cuales puedes expresar cualquier pensamiento político, incluso temas muy radicales, bajo el amparo de “es una broma”. Además, que salgan estos temas tanto, denota la importancia de estas elecciones de 2016, las cuales se caracterizaron por una gran separación dentro de la población de EEUU. Lo tercero, términos que aparecen sin ser hashtags. Aparece “hillary”, “america”, “president”, “black”… Lo destacable de que salgan estos términos es, que pueden haber aparecido tanto en tweets donde había un hashtag político como aquellos donde no.

corpus\_subset(firstCorpus,   
 date < '2016-11-08') %>%  
 tokens(remove\_punct = TRUE) %>%  
 tokens\_remove(myStopwords) %>%  
 dfm() %>%  
 dfm\_group(groups = date) %>%  
 dfm\_trim(verbose = FALSE) %>%  
 textplot\_wordcloud(color = "#1067c9", max\_words = 150)



corpus\_subset(firstCorpus,   
 date >= '2016-11-08') %>%  
 tokens(remove\_punct = TRUE) %>%  
 tokens\_remove(myStopwords) %>%  
 dfm() %>%  
 dfm\_group(groups = date) %>%  
 dfm\_trim(verbose = FALSE) %>%  
 textplot\_wordcloud(color = "#1067c9", max\_words = 150)



Podemos ver que tanto antes de las elecciones como después, los hashtags políticos siguen estando muy presentes.

png(file="ClusterDendogramOfAuthors.png",  
 width=5000,  
 height=5200,  
 res = 300,  
 bg = "white")  
  
dfmat\_profiles <- dfm\_group(thirdDfm, groups = author)  
  
tstat\_dist <- as.dist(textstat\_dist(dfmat\_profiles))  
profiles\_clust <- hclust(tstat\_dist)  
plot(profiles\_clust)  
  
dev.off()

## png   
## 2

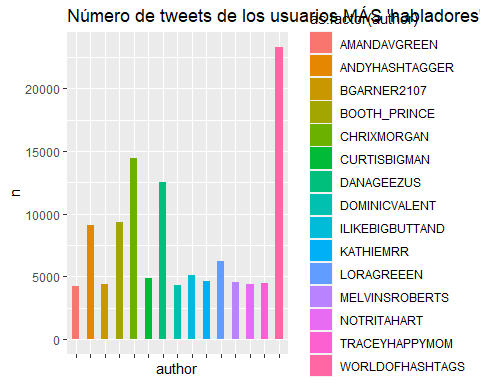
Este diagrama nos muestra la similitud entre autores, y podemos ver que hay uno claramente aislado del resto, “WorldOfHashtags”. También destacar a “chrixmorgan”, “andyhashtagger”, “booth\_prince” y “danageezus”. Esto parece indicar que pueden ser bots. Más adelante confirmaremos esto.

dataToRepresent <- myDf2 %>% group\_by(author) %>% tally()

dataToRepresent <- dataToRepresent[order(dataToRepresent$n, decreasing = TRUE), c(1,2)]

dataToRepresent2 <- dataToRepresent[c(1:15),]

ggplot(data = dataToRepresent2, aes(x = author, y = n, fill=as.factor(author))) +  
 ggtitle("Número de tweets de los usuarios MÁS 'habladores'") +   
 geom\_bar(stat = "identity", width = 0.5) +   
 theme(axis.text.x = element\_blank())



Aquí vemos a los usuarios que más twittearon de la categoría “Hashtag Games”. Vamos a hacer algunos cálculos… Un año tiene 365 x 24 = 8760 horas. Esto quiere decir, que en caso del usuario “WorldOfHashtags”, escribe unos 3 tweets cada hora de cada día del año. Esto hace sospechar que usuarios como estos sean bots. Y es así. “WorldOfHashtags” es una cuenta creada por “Internet Research Agency”, una compañia rusa que actúa en representación del gobierno ruso. Curiosamente, esta cuenta solo estuvo activa desde el 22 de Abril de 2015 al 17 de Abril de 2017, coincidiendo de lleno con las elecciones estadounidenses del 2016. Así pues, no solo los bots eran como vimos en clase “RightTrolls” o “LeftTrolls”, sino que también atacaban desde este “Hashtag Games”. Fuente: <https://russiatweets.com/author/WORLDOFHASHTAGS>

Destacar también unos usuarios que comentamos previamente, “chrixmorgan”, “andyhashtagger”, “booth\_prince” y “danageezus”. Si nos fijamos en estos nombres, son aquellos de los que podíamos sospechar que eran bots, y que según esta gráfica, se aumenta la probabilidad de que lo sean. Si son bots, y detrás de ellos está la misma empresa que tras “WorldOfHashtags”, la “Internet Research Company”.

Por lo tanto, tenemos que estos cinco usuarios que más twittean son bots. Fuentes: <https://russiatweets.com/author/ANDYHASHTAGGER>, <https://russiatweets.com/author/CHRIXMORGAN>, <https://russiatweets.com/author/DANAGEEZUS>, <https://russiatweets.com/author/BOOTH_PRINCE>

Pero, ¿y el resto? También son trolls todos ellos, y detrás de ellos la misma empresa, y por tanto el mismo gobierno, el ruso. Fuentes: <https://russiatweets.com/author/TRACEYHAPPYMOM>, <https://russiatweets.com/author/BGARNER2107/tweets?page=124>, <https://www.russiatweets.com/author/CURTISBIGMAN/tweets?page=20>, <https://russiatweets.com/author/DOMINICVALENT>, <https://www.russiatweets.com/author/ILIKEBIGBUTTAND/tweets?page=10>, <https://www.russiatweets.com/author/KATHIEMRR/tweets>, <https://russiatweets.com/author/LORAGREEEN>, <https://www.russiatweets.com/author/MELVINSROBERTS/tweets?page=71>

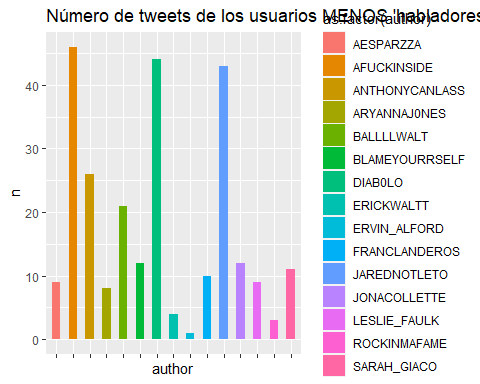
También recomiendo este enlace, donde se pueden ver muchas de estas cuentas creadas por la “Internet Research Agency”. <https://russiatweets.com/author?page=2>

Los enlaces muestran no solo cuando se creo y cerró la cuenta falsa, sino la frecuencia de tweets en función del tiempo y la región donde se llevo a cabo las publicaciones de los tweets.

dataToRepresent <- dataToRepresent[order(dataToRepresent$n), c(1,2)]

dataToRepresent3 <- dataToRepresent[c(1:15),]

ggplot(data = dataToRepresent3, aes(x = author, y = n, fill=as.factor(author))) +  
 ggtitle("Número de tweets de los usuarios MENOS 'habladores'") +   
 geom\_bar(stat = "identity", width = 0.5) +   
 theme(axis.text.x = element\_blank())



Vemos aquí los usuarios menos activos de “Hashtag Games”. Estos números si se pueden corresponder a personas normales que no fuesen bots.

A partir de aquí, vamos a crear un nuevo dataframe desde 0, para solo quedarnos con los hashtags, y eliminar el resto de contenidos. Con este dataframe crearemos los grafos.

dfHashtags <- as.data.table(games)

Nos quedamos con las menciones. En twitter, si quieres mencionar a otro usuario, se hace mediante la formula @ + nombreDeUsuario.

dfHashtags$to <- str\_extract(dfHashtags$content, '@\\w+')

Quitamos la @ para solo quedarnos con el nombre del usuario.

dfHashtags$to <- str\_replace\_all(dfHashtags$to, "@", "")

Convertimos todo a mayúsculas en la columna de destinatario. Esto es para evitar que, por ejemplo, @user, @User y @USER sean distintos, puesto que lo más seguro es que sean el mismo usuario pero mal escrito.

dfHashtags$to <- toupper(dfHashtags$to)

Lo mismo en la columna de autores

dfHashtags$author <- toupper(dfHashtags$author)

Extraemos los hashtags, eliminando el resto del contenido.

dfHashtags$content <- str\_extract(dfHashtags$content, '#\\w+')

Nos quedamos solo con tres columnas, la de autor del tweet, mencionado en el tweet, y hashtag del contenido del mensaje.

dfHashtags2 <- dfHashtags[, c(2,17,3)]

vectorMasHabladores <- dataToRepresent2$author

vectorMenosHabladores <- dataToRepresent3$author

DfMasHabladores <- subset(dfHashtags2, author %in% vectorMasHabladores)

DfMasHabladoresHashtags <- unique(DfMasHabladores[,c(3)])

DfMenosHabladores <- subset(dfHashtags2, author %in% vectorMenosHabladores)

DfMenosHabladoresHashtags <- unique(DfMenosHabladores[,c(3)])

Vamos a ver qué temas tienen en común los 15 que más hablan con los 15 que menos.

DfComun <- inner\_join(DfMasHabladoresHashtags, DfMenosHabladoresHashtags)

Vemos aquí los hashtags o temas comúnes entre los 15 usuarios más activos y los 15 menos activos. Podemos ver que hay varios temas políticos relacionados con las elecciones (“TrumpCampaignSlogans”, “UnitedStatesIn3Words”, “MakeAMovieHillary”…)

DfComun[c(2:46),]

## content  
## 1: #IGotNobelPrizeFor  
## 2: #BadPlacesToPutAPenis  
## 3: #StickHairInAMovie  
## 4: #MyPreWorkOutRoutine  
## 5: #InKindergartenILearned  
## 6: #AFunFactAboutVaginas  
## 7: #RedneckIn3Words  
## 8: #ReasonMyMomIsBetter  
## 9: #MakeAMovieHillary  
## 10: #AlcoholicMovies  
## 11: #ThingsITryToForgetButCant  
## 12: #SuperheroPresidentialSlogans  
## 13: #TrumpCampaignSlogans  
## 14: #UnitedStatesIn3Words  
## 15: #TrumpenASong  
## 16: #MyDoomsdayPlans  
## 17: #ReasonIAmNotPresident  
## 18: #AllWentWrongWhen  
## 19: #FallDepressionReasons  
## 20: #DumbGenieWishes  
## 21: #GameOfThrones  
## 22: #GiveASongAnEgo  
## 23: #AnnoyYourPartnerIn5Words  
## 24: #WasteAMillionIn3Words  
## 25: #LiesMenTell  
## 26: #RejectedWeekends  
## 27: #WeirdThingsMyBossSaid  
## 28: #OppositeDay  
## 29: #America  
## 30: #iHQ  
## 31: #AndThenIWasAskedToLeave  
## 32: #IMightNeedALawyerBecause  
## 33: #EmojiAWeekend  
## 34: #MichaelJackson  
## 35: #reasoniamnotpresident  
## 36: #BUT  
## 37: #WhyITweetIn5Words  
## 38: #WarningLabelsForPeople  
## 39: #5WordMarriageSecret  
## 40: #BlackPickupLines  
## 41: #DeviousMaids  
## 42: #LGBT  
## 43: #MyRashDecision  
## 44: #PatrioticSchmovies  
## 45: #MyJawDroppedWhen  
## content

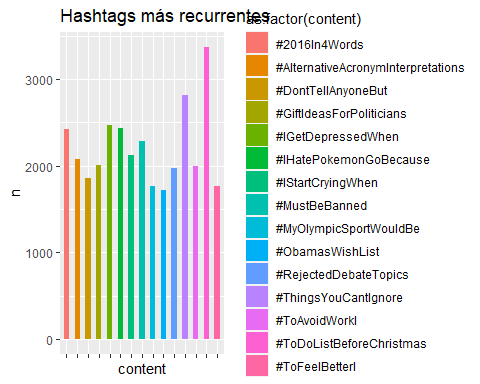
dfOnlyHashtags <- dfHashtags2[, c(3)]

dataToRepresentHashtags <- dfOnlyHashtags %>% group\_by(content) %>% tally()

dataToRepresentHashtags <- dataToRepresentHashtags[order(dataToRepresentHashtags$n, decreasing = TRUE), c(1,2)]

dataToRepresentHashtags2 <- dataToRepresentHashtags[c(2:16),]

ggplot(data = dataToRepresentHashtags2, aes(x = content, y = n, fill=as.factor(content))) +  
 ggtitle("Hashtags más recurrentes") +   
 geom\_bar(stat = "identity", width = 0.5) +   
 theme(axis.text.x = element\_blank())



Podemos ver con este gráfico los hashtags o temas que más se repiten. Teniendo en cuenta que participan según nuestros datos 112 personas, que haya temas que ocurren más de 2000 o 3000 veces, supone que eran usuarios muy activos o cuentas falsas (bots), como hemos visto previamente.

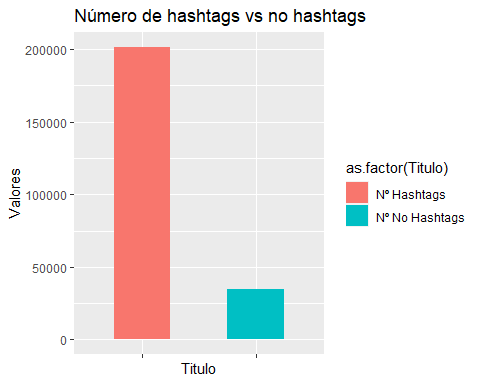
Sorprende que temas más políticos no estén presentes en esta lista, aunque tiene sentido. De esta manera, los bots pueden meter mensajes políticos dentro de temas no políticos, para intentar aparecer neutrales a pesar de tener una agenda claramente definida.

Quiero ver ahora, de todos los tweets que hay, en cuántos no hay ningún hashtag.

hashtagsValidos <- sum(dataToRepresentHashtags$n[c(2:nrow(dataToRepresentHashtags))])  
noHashtags <- dataToRepresentHashtags$n[1]

Titulo <- c("Nº Hashtags", "Nº No Hashtags")  
Valores <- c(hashtagsValidos, noHashtags)  
  
dataframeARepresentar <- data.frame(Titulo, Valores)

ggplot(data = dataframeARepresentar, aes(x = Titulo, y = Valores, fill=as.factor(Titulo))) +  
 ggtitle("Número de hashtags vs no hashtags") +   
 geom\_bar(stat = "identity", width = 0.5) +   
 theme(axis.text.x = element\_blank())



Mediante este gráfico, podemos ver que entre 30000-40000 personas catalogadas como “Hashtag Gamer”, no usaron ningún hashtag para participar en el juego. Esto podría ser porque no eran participantes “directos”, sino comentarios a las bromas (en esto consistía el juego) de otros usuarios.

corpusHashtags <- corpus(dfHashtags$content)  
tokensHashtags <- tokens(corpusHashtags)  
dfmHashtags <- dfm(tokensHashtags)  
topfeatures(dfmHashtags, 100)

## #todolistbeforechristmas #thingsyoucantignore   
## 3431 2950   
## #2016in4words #mustbebanned   
## 2621 2572   
## #igetdepressedwhen #ihatepokemongobecause   
## 2571 2529   
## #istartcryingwhen #alternativeacronyminterpretations   
## 2151 2135   
## #giftideasforpoliticians #toavoidworki   
## 2104 2064   
## #rejecteddebatetopics #secondhandgifts   
## 1993 1944   
## #donttellanyonebut #tofeelbetteri   
## 1872 1850   
## #christmasaftermath #myolympicsportwouldbe   
## 1776 1775   
## #obamaswishlist #childrenthinkthat   
## 1761 1647   
## #betteralternativetodebates #howtoloseyourjob   
## 1639 1608   
## #valentinesdayin3words #probabletrumpstweets   
## 1582 1561   
## #wheniwasyoung #ineedalawyerbecause   
## 1532 1516   
## #thingsthatshouldbecensored #myemmynominationwouldbe   
## 1501 1495   
## #obamanextjob #itsriskyto   
## 1490 1481   
## #ruinadinnerinonephrase #ihavearighttoknow   
## 1412 1382   
## #reasonstogetdivorced #survivalguidetothanksgiving   
## 1346 1317   
## #sextingwentwrongwhen #idrunforpresidentif   
## 1257 1238   
## #partywentwrongwhen #potuslasttweet   
## 1232 1216   
## #thefirst100daystodolist #ificouldntlie   
## 1189 1184   
## #reasonstoprotest #ilovemyfriendsbut   
## 1177 1140   
## #trumpsfavoriteheadline #ihate\_\_\_\_because   
## 1121 1117   
## #makemehateyouinonephrase #presentstrumpgot   
## 1114 1066   
## #howtomakeamangay #addpotinamovie   
## 1047 1026   
## #ifiweretedcruz #ifihadabodydouble   
## 1021 1017   
## #thingseveryboywantstohear #iamnotthepresidentbecause   
## 1017 1010   
## #myuselesssuperpower #thingsmoretrustedthanhillary   
## 978 943   
## #armywomenproblems #2017survivaltips   
## 929 894   
## #thingsdonebymistake #iwouldprefertoforget   
## 866 863   
## #sometimesitsokto #icelebratetrumpwith   
## 852 841   
## #growingupwithobama #thingsnottaughtatschool   
## 830 806   
## #hillarypickuplines #ihateitwhen   
## 791 789   
## #myfarewellwordswouldbe #tosavemoneyi   
## 788 786   
## #thingsiwontbelieve #ifiwereyourmom   
## 786 764   
## #ihaveadreamthat #itturnedawkwardwhen   
## 756 754   
## #reasonstoleaveeu #makeamovieblack   
## 732 732   
## #rejectedstatueoflibertysigns #easterin4words   
## 712 709   
## #alternativeoscarnominations #thingsnottodoatthezoo   
## 688 665   
## #isisamovie #itsunacceptableto   
## 658 653   
## #my2017resolution #mymondaydemotivation   
## 647 644   
## #2016electionin3words #signsyouareamerican   
## 642 624   
## #reallifemagicspells #lostin3words   
## 599 588   
## #howtoescapeadate #onewordoffmoviequotes   
## 579 574   
## #rejectedlegominifigures #mymondaystartswith   
## 554 554   
## #allwentwrongwhen #thingsthatdisturbme   
## 548 543   
## #maketvsexy #imtoooldfor   
## 539 534   
## #thingsiwonttellmydad #ilove\_\_butihate\_\_   
## 534 527   
## #itstimetopanicwhen #ididmybestbut   
## 524 522   
## #love #thingsinventedwhilehigh   
## 506 504   
## #ilikepeoplewho #meangirlsrules   
## 501 486   
## #whatiwouldtella15yearoldme #thingsyoushouldntask   
## 483 482

Ahora podemos ver los 100 hashtags más relevantes.

Para evitar hacer un grafo a partir de un dataframe de 236092 filas, voy a reducirlo un poco, seleccionando temas políticos.

temas\_de\_interes <- c("#todolistbeforechristmas", "#thingsyoucantignore", "#2016in4words", "#mustbebanned", "#igetdepressedwhen", "#giftideasforpoliticians",  
 "#rejecteddebatetopics", "#obamaswishlist", "#betteralternativetodebates", "#probabletrumpstweets", "#thingsthatshouldbecensored",  
 "#obamanextjob", "#idrunforpresidentif", "#trumpsfavoriteheadline", "#ihate\_\_\_\_because", "#thingsmoretrustedthanhillary",  
 "#hillarypickuplines")

Pero antes de reducir la búsqueda, quiero ver cómo sería el grafo completo…

He elegido que el grafo sea dirigido, puesto que un usuario menciona a alguien en el tweet, y veo clara una relación entre mencionador y mencionado.

grafo2 <- graph\_from\_data\_frame(dfHashtags2, directed = TRUE, vertices = NULL)

Reducimos un poco el grafo anterior.

grafo3 <- simplify(grafo2)

Guardamos ese grafo para poderlo tratar con Gephi. Será un grafo enorme, muy difícil de tratar. (3638 nodos y 12942 enlaces)

write\_graph(grafo3,  
 file = "grafoIntermedio.gml",  
 format = "gml")

Ahora sí, aplicamos lo anterior, reducimos los temas a algunos que he considerado de interés.

network\_final <- subgraph.edges(grafo2,  
 which(E(grafo2)$content %in% temas\_de\_interes),  
 delete.vertices = TRUE)

Borramos aquellos usuarios NA. Es decir, borramos aquello donde no se mecionase a nadie, pues quiero ver relaciones entre autores de tweets y mencionados.

network\_final <- delete\_vertices(network\_final, "NA")

Guardamos ese grafo para poder tratarlo con Gephi. De esta manera, tendremos un grafo muy sencillo de tratar (63 nodos y 65 relaciones)

write\_graph(network\_final,  
 file = "grafoFinal1.gml",  
 format = "gml")

Este grafo nos sitúa como centro al famoso “WorldOfHashtags”, siendo satélites cuentas falsas rusas, a la vez que estas también están conectadas con otras cuentas rusas más en la periferia. Pero lo más curioso, es una de las cuentas que aparecen en la periferia…, “REALDONALDTRUMP”, la cuenta de twitter de Donald Trump.

En mi opinión, no da muy buena imagen que dentro de una red en la cual predominan las cuentas falsas rusas, el que saldría elegido presidente esté dentro de esa red.

Vamos a calcular algunas métricas sobre este grafo.

degreeBots <- degree(network\_final)  
degreeBots

## ABIGAILSSILK ADRIENNE\_GG AIDEN7757 AMANDAVGREEN ANDYDREWDANIELS   
## 0 0 1 1 1   
## ANDYHASHTAGGER ANNIEPOSHES PUREDAVIE QUEENOFTHEWO TESS\_SNIDER   
## 2 1 0 0 2   
## TONEPORTER TRACEYHAPPYMOM WORLDOFHASHTAGS BABYJANNYMEOW BGARNER2107   
## 2 3 43 0 1   
## BOOTH\_PRINCE BRIGHTANDGLORY BRUCIEDUBLIN CALEBPAAR CASCASEYP   
## 0 0 1 2 0   
## CASSIEWELTCH CATELINEWATKINS CHARLESJHARPER CHRIXCHASE CHRIXMORGAN   
## 1 0 1 2 4   
## CLAUDIA42KERN CURTISBIGMAN DAILEYJADON DANAGEEZUS DANNYTHEHAPPIES   
## 2 1 1 2 2   
## DICKYIRWIN DOMINICVALENT DONNIELMILLER ELIANLINDSEY ERICARUTTER   
## 2 2 1 0 1   
## FAMEONYOUBITCH GAB1ALDANA HELLIEEDWARDS HEYHEYHAILEY HIPPPO\_   
## 2 1 2 0 0   
## HOLYCRAPCHRIX ILIKEBIGBUTTAND JARRARDNORMAN JASPER\_FLY JEANETTEDBOLDEN   
## 1 2 0 1 2   
## JMSCOXXX KATHIEMRR KENCANNONE LORAGREEEN LUGHHALLEY   
## 2 1 1 0 2   
## LYNDAWKING LYNSRODRIGUEZ MELVINSROBERTS MR\_CLAMPIN NOTRITAHART   
## 0 1 1 0 0   
## GISELLEEVNS REALDONALDTRUMP CNN SARAHNITYSEEKER STEPHENASMITH   
## 9 1 1 14 1   
## TIME SS MSBHAVEN81   
## 1 1 1

De 63 nodos, 43 están directamente relacionados con “WorldOfHashtags”

EigenvectorCentrality <- evcent(network\_final)$vector  
EigenvectorCentrality

## ABIGAILSSILK ADRIENNE\_GG AIDEN7757 AMANDAVGREEN ANDYDREWDANIELS   
## 0.000000000 0.000000000 0.016837867 0.016837867 0.096018471   
## ANDYHASHTAGGER ANNIEPOSHES PUREDAVIE QUEENOFTHEWO TESS\_SNIDER   
## 0.031611011 0.016837867 0.000000000 0.000000000 0.192036942   
## TONEPORTER TRACEYHAPPYMOM WORLDOFHASHTAGS BABYJANNYMEOW BGARNER2107   
## 0.112856338 0.208874809 1.000000000 0.000000000 0.096018471   
## BOOTH\_PRINCE BRIGHTANDGLORY BRUCIEDUBLIN CALEBPAAR CASCASEYP   
## 0.000000000 0.000000000 0.031319572 0.192036942 0.000000000   
## CASSIEWELTCH CATELINEWATKINS CHARLESJHARPER CHRIXCHASE CHRIXMORGAN   
## 0.016837867 0.000000000 0.096018471 0.192036942 0.195644448   
## CLAUDIA42KERN CURTISBIGMAN DAILEYJADON DANAGEEZUS DANNYTHEHAPPIES   
## 0.192036942 0.000000000 0.016837867 0.096911955 0.192036942   
## DICKYIRWIN DOMINICVALENT DONNIELMILLER ELIANLINDSEY ERICARUTTER   
## 0.127338042 0.048157439 0.096018471 0.000000000 0.096018471   
## FAMEONYOUBITCH GAB1ALDANA HELLIEEDWARDS HEYHEYHAILEY HIPPPO\_   
## 0.192036942 0.016837867 0.112856338 0.000000000 0.000000000   
## HOLYCRAPCHRIX ILIKEBIGBUTTAND JARRARDNORMAN JASPER\_FLY JEANETTEDBOLDEN   
## 0.016837867 0.112856338 0.000000000 0.096018471 0.112856338   
## JMSCOXXX KATHIEMRR KENCANNONE LORAGREEEN LUGHHALLEY   
## 0.127338042 0.031319572 0.096018471 0.000000000 0.192036942   
## LYNDAWKING LYNSRODRIGUEZ MELVINSROBERTS MR\_CLAMPIN NOTRITAHART   
## 0.000000000 0.096018471 0.096018471 0.000000000 0.000000000   
## GISELLEEVNS REALDONALDTRUMP CNN SARAHNITYSEEKER STEPHENASMITH   
## 0.326182780 0.018785481 0.003035241 0.175360710 0.009305338   
## TIME SS MSBHAVEN81   
## 0.018785481 0.096018471 0.000000000

La “Eigenvector Centrality” nos da una medida sobre la relevancia de un nodo en la red. Podemos ver que “WorldOfHashtags” tiene una importancia clave dentro de esta red.

Lo siguiente a calcular es la “closeness”, que nos da una medida del número de pasos necesarios desde un nodo para acceder al resto de ellos.

ClosenessBots <- closeness(network\_final)  
ClosenessBots

## ABIGAILSSILK ADRIENNE\_GG AIDEN7757 AMANDAVGREEN ANDYDREWDANIELS   
## NaN NaN 1.0000000 1.0000000 0.1428571   
## ANDYHASHTAGGER ANNIEPOSHES PUREDAVIE QUEENOFTHEWO TESS\_SNIDER   
## 0.5000000 1.0000000 NaN NaN 0.1428571   
## TONEPORTER TRACEYHAPPYMOM WORLDOFHASHTAGS BABYJANNYMEOW BGARNER2107   
## 0.1666667 0.1666667 0.3333333 NaN 0.1428571   
## BOOTH\_PRINCE BRIGHTANDGLORY BRUCIEDUBLIN CALEBPAAR CASCASEYP   
## NaN NaN 1.0000000 0.1428571 NaN   
## CASSIEWELTCH CATELINEWATKINS CHARLESJHARPER CHRIXCHASE CHRIXMORGAN   
## 1.0000000 NaN 0.1428571 0.1428571 0.1111111   
## CLAUDIA42KERN CURTISBIGMAN DAILEYJADON DANAGEEZUS DANNYTHEHAPPIES   
## 0.1428571 1.0000000 1.0000000 0.1250000 0.1428571   
## DICKYIRWIN DOMINICVALENT DONNIELMILLER ELIANLINDSEY ERICARUTTER   
## 0.1666667 0.5000000 0.1428571 NaN 0.1428571   
## FAMEONYOUBITCH GAB1ALDANA HELLIEEDWARDS HEYHEYHAILEY HIPPPO\_   
## 0.1428571 1.0000000 0.1666667 NaN NaN   
## HOLYCRAPCHRIX ILIKEBIGBUTTAND JARRARDNORMAN JASPER\_FLY JEANETTEDBOLDEN   
## 1.0000000 0.1666667 NaN 0.1428571 0.1666667   
## JMSCOXXX KATHIEMRR KENCANNONE LORAGREEEN LUGHHALLEY   
## 0.1666667 1.0000000 0.1428571 NaN 0.1428571   
## LYNDAWKING LYNSRODRIGUEZ MELVINSROBERTS MR\_CLAMPIN NOTRITAHART   
## NaN 0.1428571 0.1428571 NaN NaN   
## GISELLEEVNS REALDONALDTRUMP CNN SARAHNITYSEEKER STEPHENASMITH   
## NaN NaN NaN NaN NaN   
## TIME SS MSBHAVEN81   
## NaN NaN NaN

Ahora calculamos la authority\_score y la hub\_score (hacen referencia al valor del contenido del nodo y al valor que tienen los nodos referenciados por otro nodo, respectivamente).

authority\_score(network\_final, scale = TRUE, weights = NULL, options = arpack\_defaults)

## $vector  
## ABIGAILSSILK ADRIENNE\_GG AIDEN7757 AMANDAVGREEN ANDYDREWDANIELS   
## 3.611216e-18 3.611216e-18 3.611216e-18 3.611216e-18 3.611216e-18   
## ANDYHASHTAGGER ANNIEPOSHES PUREDAVIE QUEENOFTHEWO TESS\_SNIDER   
## 7.222433e-18 3.611216e-18 3.611216e-18 3.611216e-18 7.222433e-18   
## TONEPORTER TRACEYHAPPYMOM WORLDOFHASHTAGS BABYJANNYMEOW BGARNER2107   
## 7.222433e-18 1.444487e-17 1.000000e+00 3.611216e-18 3.611216e-18   
## BOOTH\_PRINCE BRIGHTANDGLORY BRUCIEDUBLIN CALEBPAAR CASCASEYP   
## 3.611216e-18 3.611216e-18 3.611216e-18 7.222433e-18 3.611216e-18   
## CASSIEWELTCH CATELINEWATKINS CHARLESJHARPER CHRIXCHASE CHRIXMORGAN   
## 3.611216e-18 3.611216e-18 3.611216e-18 7.222433e-18 1.444487e-17   
## CLAUDIA42KERN CURTISBIGMAN DAILEYJADON DANAGEEZUS DANNYTHEHAPPIES   
## 7.222433e-18 3.611216e-18 3.611216e-18 7.222433e-18 7.222433e-18   
## DICKYIRWIN DOMINICVALENT DONNIELMILLER ELIANLINDSEY ERICARUTTER   
## 7.222433e-18 7.222433e-18 3.611216e-18 3.611216e-18 3.611216e-18   
## FAMEONYOUBITCH GAB1ALDANA HELLIEEDWARDS HEYHEYHAILEY HIPPPO\_   
## 7.222433e-18 3.611216e-18 7.222433e-18 3.611216e-18 3.611216e-18   
## HOLYCRAPCHRIX ILIKEBIGBUTTAND JARRARDNORMAN JASPER\_FLY JEANETTEDBOLDEN   
## 3.611216e-18 7.222433e-18 3.611216e-18 3.611216e-18 7.222433e-18   
## JMSCOXXX KATHIEMRR KENCANNONE LORAGREEEN LUGHHALLEY   
## 7.222433e-18 3.611216e-18 3.611216e-18 3.611216e-18 7.222433e-18   
## LYNDAWKING LYNSRODRIGUEZ MELVINSROBERTS MR\_CLAMPIN NOTRITAHART   
## 3.611216e-18 3.611216e-18 3.611216e-18 3.611216e-18 3.611216e-18   
## GISELLEEVNS REALDONALDTRUMP CNN SARAHNITYSEEKER STEPHENASMITH   
## 2.009650e-01 3.483163e-02 3.440058e-03 1.948906e-01 1.711770e-02   
## TIME SS MSBHAVEN81   
## 3.483163e-02 4.789165e-02 7.222433e-18   
##   
## $value  
## [1] 59.41907  
##   
## $options  
## $options$bmat  
## [1] "I"  
##   
## $options$n  
## [1] 63  
##   
## $options$which  
## [1] "LM"  
##   
## $options$nev  
## [1] 1  
##   
## $options$tol  
## [1] 0  
##   
## $options$ncv  
## [1] 0  
##   
## $options$ldv  
## [1] 0  
##   
## $options$ishift  
## [1] 1  
##   
## $options$maxiter  
## [1] 1000  
##   
## $options$nb  
## [1] 1  
##   
## $options$mode  
## [1] 1  
##   
## $options$start  
## [1] 1  
##   
## $options$sigma  
## [1] 0  
##   
## $options$sigmai  
## [1] 0  
##   
## $options$info  
## [1] 0  
##   
## $options$iter  
## [1] 1  
##   
## $options$nconv  
## [1] 1  
##   
## $options$numop  
## [1] 20  
##   
## $options$numopb  
## [1] 0  
##   
## $options$numreo  
## [1] 20

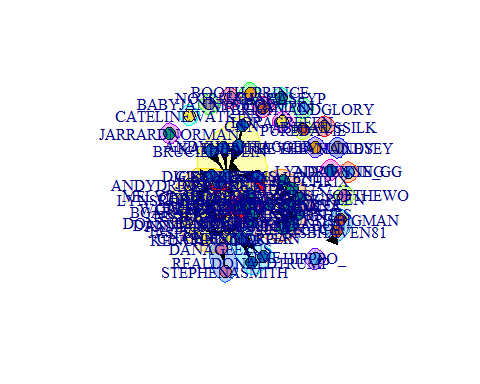
hub\_score(network\_final, scale = TRUE, weights = NULL, options = arpack\_defaults)

## $vector  
## ABIGAILSSILK ADRIENNE\_GG AIDEN7757 AMANDAVGREEN ANDYDREWDANIELS   
## 5.869243e-17 5.869243e-17 6.848656e-02 6.848656e-02 3.514102e-01   
## ANDYHASHTAGGER ANNIEPOSHES PUREDAVIE QUEENOFTHEWO TESS\_SNIDER   
## 7.183001e-02 6.848656e-02 5.869243e-17 5.869243e-17 7.028204e-01   
## TONEPORTER TRACEYHAPPYMOM WORLDOFHASHTAGS BABYJANNYMEOW BGARNER2107   
## 4.198968e-01 7.713070e-01 1.000000e+00 5.869243e-17 3.514102e-01   
## BOOTH\_PRINCE BRIGHTANDGLORY BRUCIEDUBLIN CALEBPAAR CASCASEYP   
## 5.869243e-17 5.869243e-17 7.062114e-02 7.028204e-01 5.869243e-17   
## CASSIEWELTCH CATELINEWATKINS CHARLESJHARPER CHRIXCHASE CHRIXMORGAN   
## 6.848656e-02 5.869243e-17 3.514102e-01 7.028204e-01 7.273008e-01   
## CLAUDIA42KERN CURTISBIGMAN DAILEYJADON DANAGEEZUS DANNYTHEHAPPIES   
## 7.028204e-01 5.869243e-17 6.848656e-02 3.574255e-01 7.028204e-01   
## DICKYIRWIN DOMINICVALENT DONNIELMILLER ELIANLINDSEY ERICARUTTER   
## 4.220313e-01 1.391077e-01 3.514102e-01 5.869243e-17 3.514102e-01   
## FAMEONYOUBITCH GAB1ALDANA HELLIEEDWARDS HEYHEYHAILEY HIPPPO\_   
## 7.028204e-01 6.848656e-02 4.198968e-01 5.869243e-17 5.869243e-17   
## HOLYCRAPCHRIX ILIKEBIGBUTTAND JARRARDNORMAN JASPER\_FLY JEANETTEDBOLDEN   
## 6.848656e-02 4.198968e-01 5.869243e-17 3.514102e-01 4.198968e-01   
## JMSCOXXX KATHIEMRR KENCANNONE LORAGREEEN LUGHHALLEY   
## 4.220313e-01 7.062114e-02 3.514102e-01 5.869243e-17 7.028204e-01   
## LYNDAWKING LYNSRODRIGUEZ MELVINSROBERTS MR\_CLAMPIN NOTRITAHART   
## 5.869243e-17 3.514102e-01 3.514102e-01 5.869243e-17 5.869243e-17   
## GISELLEEVNS REALDONALDTRUMP CNN SARAHNITYSEEKER STEPHENASMITH   
## 4.695394e-16 5.869243e-17 5.869243e-17 6.260525e-16 5.869243e-17   
## TIME SS MSBHAVEN81   
## 5.869243e-17 5.869243e-17 5.869243e-17   
##   
## $value  
## [1] 59.41907  
##   
## $options  
## $options$bmat  
## [1] "I"  
##   
## $options$n  
## [1] 63  
##   
## $options$which  
## [1] "LM"  
##   
## $options$nev  
## [1] 1  
##   
## $options$tol  
## [1] 0  
##   
## $options$ncv  
## [1] 0  
##   
## $options$ldv  
## [1] 0  
##   
## $options$ishift  
## [1] 1  
##   
## $options$maxiter  
## [1] 1000  
##   
## $options$nb  
## [1] 1  
##   
## $options$mode  
## [1] 1  
##   
## $options$start  
## [1] 1  
##   
## $options$sigma  
## [1] 0  
##   
## $options$sigmai  
## [1] 0  
##   
## $options$info  
## [1] 0  
##   
## $options$iter  
## [1] 1  
##   
## $options$nconv  
## [1] 1  
##   
## $options$numop  
## [1] 20  
##   
## $options$numopb  
## [1] 0  
##   
## $options$numreo  
## [1] 19

Corroboramos mediante estas medidas esta relevancia clave de “WorldOfHahstags” dentro de la red.

Cálculo de comunidades mediante el algoritmo Louvain. Hago todo esto con igraph, pero me parece mucho más sencillo con Gephi.

networkBotsUndirected <- as.undirected(network\_final, mode = "collapse")  
enronLouvain <- cluster\_louvain(networkBotsUndirected)  
  
plot(enronLouvain, network\_final)



Ahora haré algo parecido a lo que me enviaste por mail… Quiero ver la relación de usuarios y temas.

dfHashtags <- as.data.table(games)  
  
dfHashtags <- dfHashtags[ , c(2:3, 6, 8:10, 11 )]  
dfHashtags$content <- tolower(dfHashtags$content)  
dfHashtags$content <- str\_extract(dfHashtags$content, '#\\w+')  
  
dfHashtags <- dfHashtags[which(!is.na(dfHashtags$content)), ]

hashtags <- dfHashtags %>%  
 group\_by(content)  
  
hashtags\_m <- hashtags[ , c(1,2)]  
View(hashtags\_m)  
length(unique(hashtags\_m$content))

## [1] 4539

Voy a seleccionar únicamente a los cinco usuarios más “habladores”, que eran los bots rusos…

dfReducido <- subset(hashtags\_m, author %in% dataToRepresent2$author[c(1:5)])

Elimino filas duplicadas, para simplificar el grafo. Quiero ver los temas como nodos únicos, no que aparezca un mismo tema como múltiples nodos.

dfReducidoFinal <- distinct(dfReducido)

Guardo el grafo en local para tratarlo con Gephi.

grafo2 <- graph\_from\_data\_frame(dfReducidoFinal, directed = TRUE, vertices = NULL)  
  
write\_graph(grafo2,  
 file = "grafoFinal2.gml",  
 format = "gml")

En este grafo podemos ver como las cinco cuentas falsas rusas más activas escogían una gran diversidad de temas.

En conclusión, tenemos que los bots rusos influyeron de todas las maneras posibles, no solo haciendo tweets de tendencia política marcada, sino aprovechando la organización de un juego de un programa americano para también atacar desde ese frente, para cumplir con su agenda. Al final, todo esto hizo que Donald Trump fuese escogido como el cuadragésimo quinto presidente de los Estados Unidos, más afín al presidente ruso, Vladimir Putin. Los rusos hicieron todo aquello que estaba en su mano para ayudar a la elección de Trump, atacando desde todo frente posible.