

# **Construcción de Wordclouds en R**

**José David Sánchez Bravo**

**Profesor:  
Andrés Mauricio Arcila**

**Universidad Icesi  
Facultad de Ciencias administrativas y económicas  
Programa de Economía y negocios internacionales  
Santiago de Cali  
Marzo de 2013**

# Apuntes de Economía

**ISSN 1794-029X**

No. 37, Marzo de 2013

Editor

Julio César Alonso

[jcalonso@icesi.edu.co](mailto:jcalonso@icesi.edu.co)

Asistente editorial

Andrés Felipe Muñoz G.

[afmunoz@icesi.edu.co](mailto:afmunoz@icesi.edu.co)

Gestión Editorial

Departamento de Economía - Universidad Icesi

[www.icesi.edu.co](http://www.icesi.edu.co)

Tel: 5552334 ext: 8398. Fax: 5551441

Calle 18 # 122-135 Cali, Valle del Cauca, Colombia

## Contenido

1. Introducción	4
2. Introducción al paquete <i>wordclouds</i>	6
2.1 Estructura Básica	6
3. Construcción de una nube de texto	11
3.1 Poniéndole color a las nubes de texto	13
3.2 Cambiándole el tipo de la letra en las nubes de texto	15
4. Nubes de palabras para comprar textos	16

# Construcción de Wordclouds en R.

Julio César Alonso\*

Andrés Mauricio Arcila\*\*

José David Sánchez\*\*\*

Departamento de Economía  
Universidad Icesi  
Cali - Colombia

Marzo de 2013

## Resumen

Este documento, de carácter pedagógico, permite introducir a estudiante o analista en los pasos para la elaboración y presentación de textos en nubes de texto (también conocidos como *wordclouds*). Una nube de palabras es una representación gráfica de textos que permite identificar las palabras que aparecen con mayor frecuencia en este. Empleando el paquete de *wordcloud* del *software* estadístico R, este documento muestra cómo crear diferentes nubes de palabras en el que se contemplan varias alternativas para la presentación de textos. Además, en este documento se muestra como realizar este tipo de gráficos que permita comparar dos textos al mismo tiempo. La realización de análisis comparativos. Este documento supone un conocimiento previo en el manejo de R.

**Palabras claves:** Nube de palabras, visualización de textos, R.

## Abstract

This document has a pedagogical nature. We introduce student or analyst step by step in how to construct a word clouds using R. A word cloud is a text 's graphical representation to identify the words that appear most frequently. Using the package *wordcloud* for R , this paper develops two complete examples of word clouds and comparison word clouds. This document assumes a prior working knowledge of R.

**Keywords:** Word clouds, text visualization, R.

---

\*Profesor del departamento de Economía y director del CIENFI (Centro de investigaciones en Economía y Finanzas) jcalonso@icesi.edu.co

\*\*Profesor del departamento de Economía.

\*\*\*Estudiante de noveno semestre de Economía en la Universidad Icesi.

## Objetivos de Aprendizaje

Al finalizar la lectura de este documento se espera que el lector esté en capacidad de:

- Organizar los archivos y base de datos de textos necesarias para graficar información empleando nubes de palabras en R.
- Graficar textos en nubes de palabras empleando R usando el paquete *wordcloud*.

## 1. Introducción

Actualmente, en múltiples disciplinas ha aumentado el interés por presentar información tanto cuantitativa como cualitativa a través de herramientas visuales. Mediante las herramientas visuales se ha logrado presentar con mayor facilidad y de manera estilizada información que describe el comportamiento de las variables bajo estudio. Las representaciones gráficas permiten típicamente resumir la información de tal manera que una idea relativamente complicada puede ser explicada rápidamente. De la misma manera, las nubes de palabras (también conocidas como *wordclouds* por su nombre en inglés) son un modo especial de presentar un texto de manera visual (Ver por ejemplo Baralt, Melissa y Pennestri (2011) y McNaught y Lam (2010) ).

Las nubes de texto se caracterizan por enfatizar las palabras que con mayor frecuencia se presentan en el texto graficado. De esta manera, las nubes de texto se convierten en una herramienta visual de lectura intuitiva que sintetiza las principales ideas de un texto. Las nubes de texto o **wordclouds** son una herramienta que permiten visualizar textos facilitando el análisis preliminar de estos. Esta representación gráfica puede considerarse como un resumen gráfico del texto analizado que resalta las principales ideas del texto. La frecuencia de las palabras o caracteres en los textos son representadas en las nubes de texto por su tamaño, lastimosamente la construcción de este tipo de gráficos no es sencilla ni se puede realizar con software convencionalmente instalado en computadores personales. Este documento tiene como objetivo mostrar al lector, paso a paso, como emplear el paquete *wordcloud* en R para la construcción de este tipo de gráficos.

Antes de entrar en el detalle de la construcción de las nubes de texto, consideremos la Figura 1 se presenta un ejemplo de este tipo de gráficos que permiten observar la frecuencia de las palabras en el discurso de posesión del presidente de los Estados Unidos Barack Obama en su primer término.



acuerdo con las preferencias del usuario. En este documento se presenta una breve introducción de los pasos para la elaboración de wordclouds en R desarrollando ejemplos con base en la minuta del 21 de marzo de 2014 del Banco de la República de Colombia. También, se presenta un análisis de los discursos del presidente de Colombia Santos y el líder de las FARC (Timochenko) al anunciar el inicio de los diálogos de paz el 4 de septiembre de 2012.

El presente documento se divide de la siguiente manera: la primera sección es la presente introducción. La segunda parte, presenta una introducción al paquete *wordcloud* en R. La tercera parte del documento muestra como hacer una nube de texto para un texto empleando como ejemplo la minuta del Banco de la República. La cuarta parte presenta en detalle como construir una nube de texto que permita comparar dos discursos al mismo tiempo. Como ejemplo se emplean los discursos de las dos partes al iniciar los diálogos de paz.

## 2. Introducción al paquete *wordcloud*

*Wordcloud* es un paquete de R creado por Ian Fellows ( Fellows, 2013). Los paquetes de R corresponden a un conjunto de funciones de R que permiten desarrollar algún tipo de tarea específica. En este caso construir los gráficos de interés. El paquete se ejecuta con el software R. Para la ejecución del paquete es necesario instalar primero R . Después de tener instalado R, el paquete se puede instalar con la siguiente línea de código.

```
install.packages("wordcloud")
```

La instalación del paquete solo se requiere hacer una vez. Es decir, si el paquete ya está instalado, no es necesario realizar el paso anterior. Un vez instalado el paquete, es necesario cargarlo antes de poder emplear cualquiera de las funciones. Es importante aclarar, que los paquetes se deben cargar cada vez que se abra de nuevo R. El paquete se puede cargar empleando el siguiente código.

```
library(wordcloud)
```

Otro paquete que permitirá un manejo de los textos que emplearemos es el denominado *tm*. Con este paquete podremos, por ejemplo, transformar los textos en matrices e importar los textos. Este paquete se puede instalar y cargar con el siguiente código.

```
install.packages("tm")  
library(tm)
```

Recuerde que otra forma de instalar paquetes cuando se emplea RStudio es mediante la opción “package installer” ubicada en el menú “Packages Data”.

### 2.1. Estructura Básica

Para graficar usando el paquete *wordcloud* se emplea una serie de funciones que tienen una estructura estándar y que implican entre otros los siguientes argumentos:

- El texto que contiene las palabras o caracteres que se van a graficar en la nube de palabras. Este argumento se denomina en el paquete “words”. Este argumento es obligatorio para emplear la función.
- La frecuencia (número de veces que se repite cada palabra en el texto) de las palabras en el gráfico. Este argumento se denomina en el paquete “freq”. Este argumento es obligatorio para emplear la función.
- La escala de tamaño que se emplearán para dibujar las palabras dentro de la nube de texto. Este argumento se denomina en el paquete “scale”. Si este argumento no se especifica, entonces el paquete emplea como valor de defecto el tamaño de letra 4 para la palabra o letra con la mínima frecuencia y aumenta en 0.5 el tamaño de letra para caracteres o palabras a medida que aumenta la frecuencia.
- El mínimo de frecuencia de repetición de una palabra para ser incluida en la nube de texto. Este argumento se denomina en el paquete “min.freq”. Si este argumento no se especifica, entonces el paquete emplea como valor de defecto 3.
- El número máximo de palabras que se van a incluir en la nube de texto. Este argumento se denomina en el paquete “max.words”. Si este argumento no se especifica, entonces el paquete no pone restricciones al número de palabras o caracteres y las incluye todas las que tiene frecuencias mayores o iguales a “min.freq”.

La función básica del paquete *wordcloud* tiene el mismo nombre del paquete: “wordcloud”. Además, de los argumentos enumerados anteriormente, la función “wordcloud” permite añadir características que mejoran la apariencia del gráfico, dentro de estas características se encuentra el color, el tipo de letra y el tamaño de la nube, entre otros.

Antes de presentar los ejemplos, creamos una nube de texto muy sencilla. Partamos de los objetos “letters” y “LETTERS” que se encuentran en el paquete base de R.

```
letters
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n"
## [15] "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"

LETTERS
## [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N"
## [15] "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
```

Como se puede observar, estos objetos tienen todas las letras del alfabeto en inglés. Es decir, no incluye ni la “ñ” ni la “LL”. El primer objeto contiene todas las letras en minúsculas y el segundo en mayúsculas. Ahora emplearemos estos dos objetos como caracteres a graficar en la nube de texto, pero falta la frecuencia de repetición de cada una de las letras. Supongamos que la primera letra (“a”) se repite una vez y la última (“Z”) se repite 1000 veces. Y para generar las frecuencias de repetición de las otras letras empleemos el siguiente código.

```
seq(1, 1000, len = 52)
```

```
## [1] 1.00 20.59 40.18 59.76 79.35 98.94 118.53
## [8] 138.12 157.71 177.29 196.88 216.47 236.06 255.65
## [15] 275.24 294.82 314.41 334.00 353.59 373.18 392.76
## [22] 412.35 431.94 451.53 471.12 490.71 510.29 529.88
## [29] 549.47 569.06 588.65 608.24 627.82 647.41 667.00
## [36] 686.59 706.18 725.76 745.35 764.94 784.53 804.12
## [43] 823.71 843.29 862.88 882.47 902.06 921.65 941.24
## [50] 960.82 980.41 1000.00
```

Este código genera una secuencia de 52 números que van de 1 a 1000 aumentando en intervalos regulares. Ya tenemos un listado de caracteres y una frecuencia de repetición de estas letras, entonces podemos crear una nube de texto con estos elementos. Para tal fin emplee la siguiente línea de código.

```
wordcloud(words = c(letters, LETTERS), freq = seq(1, 1000, len = 52))
```

En este caso obtenemos la nube de texto que se presenta en la Figura 2. Noten que la letra más grande es la zeta mayúscula, pues es la letra con la frecuencia de 1000 en nuestra base de datos ficticia. Además, noten que ni las letras a, b y c minúsculas no se encuentran en la nube, pues estas letras tienen frecuencias inferiores a tres, que es el valor por omisión para incluir un carácter o palabra en la nube.

```
wordcloud(words = c(letters, LETTERS), freq = seq(1, 1000, len = 52))
```



Figura 2: Primer ejemplo de nube de texto

En la siguiente tabla se presentan todos los argumentos de la función “wordcloud”.

Argumento	Breve descripción	Valor por defecto
words	Las palabras que serán graficadas	
freq	La frecuencia de las palabras.	
scale	Indica el rango del tamaño de las palabras.	c(4,.5)
min.freq	Las palabras con frecuencia menor a min.freq no serán graficadas	3
max.words	El máximo de las palabras en el wordcloud.	Inf
random.order	Grafica las palabras aleatoriamente. Si es FALSE, entonces las palabras aparecen de manera descendente según la frecuencia.	TRUE
random.color	Elige los colores de manera aleatoria. Si es FALSE, los colores serán elegidos con base a la frecuencia.	FALSE
rot.per	90 grados de rotación a las palabras graficadas.	0.1
colors	El color de las palabras de menor a mayor frecuencia.	"black"
ordered.colors	Si es TRUE, entonces los colores de las palabras se asignaran en orden	FALSE
use.r.layout	Si es FALSE, entonces c++ código es utilizado para la detección de colisión.	FALSE
fixed.asp	Si es TRUE, el aspecto de la escala es arreglado.	TRUE

Cuadro 1: Argumentos de la función “wordcloud”.

Otra función importante del paquete *wordcloud* es la función “commonality.cloud”. Esta función también permite crear una nube de texto como lo hace la función “wordcloud” pero además permite crear nubes de texto que permitan comparar dos textos. Mas adelante se discutirá en detalle este tipo de nubes. En la siguiente tabla se presentan todos los argumentos de la función “commonality.cloud”.

Argumento	Breve descripción
term.matrix	Una matriz con los dos textos a comparar. La primera columna corresponde a las palabras que están presentes en los dos textos. La segunda columna corresponde a la frecuencia de la respectiva palabra en el texto 1. La tercera fila corresponde a la frecuencia de la respectiva palabra en el texto 2.
max.words	El máximo número de palabras que se graficarán en la nube de texto.
...	Además se pueden emplear los mismos argumentos de la función “wordcloud”

Cuadro 2: Argumentos de la función “commonality.cloud”.

En las siguientes secciones se discutirá de manera detallada los pasos necesarios para crear la nubes de texto.

### 3. Construcción de una nube de texto

En esta sección se presentan los pasos necesarios para crear una nube de texto para la minuta de la reunión mensual de la Junta Directiva del Banco de la República de Colombia realizada el 21 de marzo de 2014. El texto de dicha minuta se encuentra en formato “.txt” en el archivo “Minuta.txt” que se encuentra disponible en el sitio *web* de Apuntes de Economía. El primer paso será cargar dicho archivo. Esto se puede realizar empleando el siguiente código.

```
minuta1 <- read.table("Minuta.txt", header = TRUE, sep = "\n",  
  encoding = "UTF-8 ")
```

La función “read.table” leerá el documento y lo guardará en el objeto *minuta1* con una clase *data.frame*. Ahora debemos dar formato al texto importado para poderlo graficar con las funciones del paquete *wordcloud*. Empleando el paquete *tm* se convertirá el objeto *minuta1* en tipo *Corpus*. En este caso *Corpus* corresponderá a la principal estructura en el manejo de datos o colección de datos. Para hacerlo, el primer paso implicará emplear el siguiente código:

```
Texto1 <- VectorSource(minuta1)
```

Esta línea de código cambia la codificación del texto que se encuentra en el objeto *minuta1* a UTF-8. En otras palabras, cambia el texto que se encuentra en español, que incluye tildes, a un formato de inglés que no incluye tildes. Es importante anotar que R no podrá construir nubes de palabras con palabras que tengan tildes, por eso es necesario transformar el objeto *minuta1*. Para explotar todas las utilidades del paquete *tm*, es necesario trabajar con objetos de la clase *Corpus*. Esto se puede realizar empleando la siguiente línea de código.

```
Texto1 <- Corpus(Texto1)
```

Ahora podemos empezar a “limpiar” el texto de la minuta. Es decir, debemos quitar los signos de puntuación, los números, los espacios en blanco y quitar las mayúsculas. Eso se puede realizar (respectivamente) empleando las siguientes líneas de código.

```
Texto1 <- tm_map(Texto1, removePunctuation)  
Texto1 <- tm_map(Texto1, removeNumbers)  
Texto1 <- tm_map(Texto1, stripWhitespace)  
Texto1 <- tm_map(Texto1, tolower)
```

Ahora, será necesario excluir las preposiciones del texto. esto se puede realizar empleando el siguiente código.

```
Texto1 <- tm_map(Texto1, removeWords, stopwords("spanish"))
```

Ahora, que ya se han realizado todas las transformaciones del texto, es importante anotar que el objeto *Texto1* es de clase “*Corpus*”. Para poder emplear las funciones “wordcloud” y

“commonality.cloud” se debe transformar la clase del objeto a clase “data.frame” para la función “wordcloud” y la clase “matrix” para la función ‘commonality.cloud’. Esto se puede lograr con las dos siguientes líneas de código.

```

Texto2 <- TermDocumentMatrix(Texto1)
Texto2 <- as.matrix(Texto2)
v <- sort(rowSums(Texto2), decreasing = TRUE)
Texto3 <- data.frame(word = names(v), freq = v)

```

Con esto hemos construido un objeto **matrix** (objeto *Texto2*) cuyos nombres de filas son las palabras que se van a graficar y la columna corresponde a la frecuencia de cada palabra. También hemos construido un **data frame** (objeto *Texto3*) con dos variables. La primera variable corresponde a las palabras que queremos graficar (**word**) y la segunda corresponde a la frecuencia con que aparece la palabra en la minuta (**freq**). Ahora, podemos graficar una nube de texto empleando la función “wordcloud”. Una nube de texto con las 100 palabras más empleadas se pueden producir con el siguiente código.

```

wordcloud(words = Texto3$word, freq = Texto3$freq, max.words = 100,
          random.order = FALSE)

```



Figura 3: Nube de texto de la minuta de la Reunión de Junta del Banco de República del 21 de marzo de 2014 empleando la función “wordcloud”.







Figura 6: Nube de texto de la minuta de la Reunión de Junta del Banco de República del 21 de marzo de 2014 con diferentes escalas de colores.

### 3.2. Cambiándole el tipo de la letra en las nubes de texto

Otras opciones que permiten dar formato diferente a una nube de palabras es cambiar el tipo de letra. El tipo de letra en la función “wordcloud” se puede controlar con el argumento “vfont”. En la Figura 7 se presentan diferentes opciones de nubes de texto que se generaron con el siguiente código.

```
par(mfrow = c(2, 2))

wordcloud(words = Texto3$word, freq = Texto3$freq, max.words = 100,
  random.order = FALSE, colors = brewer.pal(9, "Blues"))
wordcloud(words = Texto3$word, freq = Texto3$freq, max.words = 100,
  random.order = FALSE, colors = brewer.pal(9, "BuGn"), vfont = c("script",
  "plain"))
wordcloud(words = Texto3$word, freq = Texto3$freq, max.words = 100,
  random.order = FALSE, vfont = c("serif", "plain"))
```

```
wordcloud(words = Texto3$word, freq = Texto3$freq, max.words = 100,
  random.order = FALSE, vfont = c("gothic english", "plain"))
```



Figura 7: Nube de texto de la minuta de la Reunión de Junta del Banco de República del 21 de marzo de 2014 con diferentes tipos de letra.

#### 4. Nubes de palabras para comparar textos

R permite en una misma nube graficar dos textos de manera simultánea. Esto permite realizar comparaciones de dos textos de manera gráfica. El paquete *wordcloud* permite realizar dos tipos de comparaciones de textos. La primera es una que distinga las diferencias más grandes entre los textos; esto se puede realizar mediante la función “*comparison.cloud*”. El segundo tipo de comparaciones es una que resalta las similitudes entre los textos; esto se puede realizar empleando la función “*commonality.cloud*”. Es decir, en el primer tipo de comparación se concentra la atención sobre aquellas palabras que son diferentes entre los textos, mientras que la segunda comparación destaca aquellas palabras que son comunes a los textos. A continuación, veremos como construir ambos tipos de nubes de palabras.

Para desarrollar un ejemplo en detalle, emplearemos los discursos realizados por el presidente Juan Manuel Santos de Colombia y el jefe de las FARC Rodrigo Londoño alias “Timochenko” al inicio de los diálogos de paz el 4 de septiembre de 2012.

Como se mencionó anteriormente, la función “comparison.cloud” del paquete *wordcloud* se puede emplear para construir nubes de palabras que encuentren las diferencias entre los textos. En el siguiente cuadro se presentan todos los argumentos de la función “comparison.cloud”.

Argumento	Breve descripción	Valor por defecto
term.matrix	Una matriz con los dos textos a comparar. La primera columna corresponde a las palabras que están presentes en los dos textos. La segunda columna corresponde a la frecuencia de la respectiva palabra en el texto 1. La tercera fila corresponde a la frecuencia de la respectiva palabra en el texto 2.	
scale	Indica el rango del tamaño de las palabras.	c(4,.5)
max.words	El máximo de las palabras en el wordcloud.	300
random.order	Grafica las palabras aleatoriamente. Si es FALSE, entonces las palabras aparecen de manera descendente según la frecuencia.	FALSE
rot.per	90 grados de rotación a las palabras graficadas.	0.1
colors	El color de las palabras de menor a mayor frecuencia.	rewer.pal (ncol(term.matrix),"Dark2")
ordered.colors	Si es TRUE, entonces los colores de las palabras se asignaran en orden	FALSE
use.r.layout	Si es FALSE, entonces c++ código es utilizado para la detección de colisión.	FALSE

Cuadro 3: Argumentos de la función “comparison.cloud”.

Además de los paquetes ya empleados hasta ahora, necesitaremos el paquete *stringr*. El paquete se puede instalar con la siguiente línea de código.

```
install.packages("stringr")
```

La instalación del paquete solo se requiere hacer una vez. Una vez instalado el paquete, es necesario cargarlo antes de poder emplear cualquiera de las funciones. El paquete se puede cargar empleando el siguiente código.

```
library(stringr)
```

Al igual que en la nube anterior, el primer paso para construir esta nube para comparar textos es cargar los datos de los dos discursos que se encuentran en el archivo “discursos.txt”.

Para entender la estructura del archivo puede abrir el archivo. El archivo se puede cargar con la siguiente línea de código (el archivo debe estar en la misma carpeta donde está el **working space**).

```
Texto4 <- paste(readLines("discursos.txt"), collapse = "\n")
```

Noten que en este caso se está empleando un código diferente al empleado en nuestro primer ejemplo. En este caso la función “readLines” lee el archivo en el que están los discursos línea por línea. Mas adelante será claro porque esto es útil.

Las siguientes líneas de código separa los dos discursos y crea la correspondiente base de datos

```
splitat <- str_locate_all(Texto4, "(Santos|Timochenko)\\:")[[1]]
speaker <- str_sub(Texto4, splitat[, 1], splitat[, 2])
content <- str_sub(Texto4, splitat[, 2] + 1, c(splitat[-1, 1] -
  1, nchar(Texto4)))
names(content) <- speaker
tmp <- list()
for (sp in c("Santos:", "Timochenko:")) {
  tmp[sp] <- paste(content[sp == speaker], collapse = "\n")
}
collected <- unlist(tmp)
```

Como lo realizamos con el ejercicio anterior, podemos empezar a “limpiar” el texto. Es decir, debemos quitar los signos de puntuación, los números, los espacios en blanco y quitar las mayúsculas. Eso se puede realizar (respectivamente) empleando las siguientes líneas de código.

```
Texto5 <- Corpus(VectorSource(collected))
Texto5 <- tm_map(Texto5, removePunctuation)
Texto5 <- tm_map(Texto5, removeNumbers)
Texto5 <- tm_map(Texto5, stripWhitespace)
Texto5 <- tm_map(Texto5, tolower)
Texto5 <- tm_map(Texto5, function(x) removeWords(x, stopwords("Spanish")))
Texto6 <- TermDocumentMatrix(Texto5)
Texto6 <- as.matrix(Texto6)
```

Finalmente, podemos construir la nube de palabras empleando la función “comparison.cloud”.

```
comparison.cloud(Texto6, max.words = 200, random.order = FALSE)
```



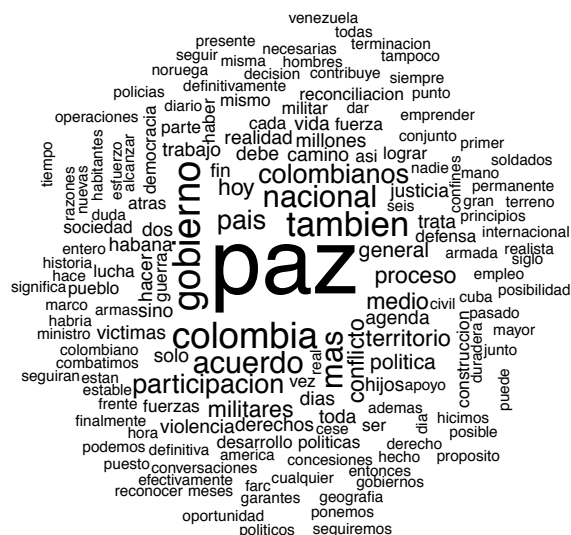


Figura 9: Nube de texto comparativo de las similitudes de los discursos de Santos y Timochenko.

La figura 9 corresponde a un gráfico comparativo en el que se destaca las palabras comunes en los discursos respectivos de Santos y Timochenko. La palabra más empleada en ambos discursos es “paz”.

Antes de terminar, es importante anotar que el paquete *wordcloud* tiene muchas más opciones que las discutidas en este documento. La intención de este documento es brindar una introducción a dicho paquete y en ningún momento se pretende proveer un manual completo del paquete.

## Referencias

Baralt, Melissa, S Pennestri y M Selvandin (2011). “ACTION RESEARCH USING WORDLES TO TEACH FOREIGN LANGUAGE WRITING”. En: *About Language Learning & Technology* 15.2, págs. 12-22. URL: <http://www.llt.msu.edu/issues/june2011/v15n2.pdf#page=17>.

- Fellows, Ian (2013). *wordcloud: Word Clouds*. R package version 2.4. URL: <http://CRAN.R-project.org/package=wordcloud>.
- McNaught, C y Paul Lam (2010). "Using Wordle as a Supplementary Research Tool." En: *Qualitative Report* 15.3, págs. 630-643. URL: <http://files.eric.ed.gov/fulltext/EJ887905.pdf>.
- R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: <http://www.r-project.org/>.