Apple Worldwide Developers Conference 1997 - 2010

Andrés Rincón - Santiago Quiroga

2024-12-03

```
##### importar datos
# Carga de librerías necesarias
suppressMessages(suppressWarnings(library(readr)))  # Lectura de archivos CSV
suppressMessages(suppressWarnings(library(tidyverse)))  # Conjunto de paquetes para manipulación de dat
# warnings debido a caracteres no UTF-8 o vacios ("")
# UTF-8 (8-bit Unicode Transformation Format) es un formato de codificación de caracteres
# capaz de codificar todos los code points validos en Unicode
# Importar los textos de las conferencias
text_1997 <- read_csv("AppleWWDC1997_es.txt", col_names = FALSE, show_col_types = FALSE)
```

Importacion de textos.

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2001 <- read_csv("AppleWWDC2001_es.txt", col_names = FALSE, show_col_types = FALSE)
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2005 <- read_csv("AppleWWDC2005_es.txt", col_names = FALSE, show_col_types = FALSE)
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2008 <- read_csv("AppleWWDC2008_es.txt", col_names = FALSE, show_col_types = FALSE)
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2010 <- read_csv("AppleWWDC2010_es.txt", col_names = FALSE, show_col_types = FALSE)
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
class(text_1997)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

```
text_1997 <- c(text_1997)
class(text_1997)
```

```
## [1] "list"
```

```
text_1997 <- unlist(text_1997)
class(text_1997)
```

```
## [1] "character"
```

```
names(text_1997) <- NULL  # importante!
head(text_1997, n = 3)
```

```
## [1] "Buenos días"
## [2] "Ambos llevaban corbata toda la semana"
## [3] "Noticias"
```

```
text_2001 <- unlist(c(read_csv("AppleWWDC2001_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2005 <- unlist(c(read_csv("AppleWWDC2005_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2008 <- unlist(c(read_csv("AppleWWDC2008_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2010 <- unlist(c(read_csv("AppleWWDC2010_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
names(text_2001) <- NULL
```

```
names(text_2005) <- NULL
```

```
names(text_2008) <- NULL
```

```
names(text_2010) <- NULL
```

```
##### data frame formato tidy

text_1997 <- tibble(line = 1:length(text_1997), text = text_1997)  # tibble en lugar de data_frame
class(text_1997)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
dim(text_1997)
```

```
## [1] 1322    2
```

```
head(text_1997, n = 3)
```

```
## # A tibble: 3 x 2
##    line text
##   <int> <chr>
## 1     1 Buenos días
## 2     2 Ambos llevaban corbata toda la semana
## 3     3 Noticias
```

```
# texto no normalizado
# no tiene "estructura" para analizar

text_2001 <- tibble(line = 1:length(text_2001), text = text_2001)
```

```
text_2005 <- tibble(line = 1:length(text_2005), text = text_2005)
```

```
text_2008 <- tibble(line = 1:length(text_2008), text = text_2008)
```

```
text_2010 <- tibble(line = 1:length(text_2010), text = text_2010)
```

#Tokenizacion

```
suppressMessages(suppressWarnings(library(tidytext)))
suppressMessages(suppressWarnings(library(magrittr)))
##### tokenizacion formato tidy

text_1997 %<>%
  unnest_tokens(input = text, output = word) %>%
  filter(!is.na(word))  # importante!
class(text_1997)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
dim(text_1997)
```

```
## [1] 11356     2
```

```
head(text_1997, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 buenos
##  2     1 días
##  3     2 ambos
##  4     2 llevaban
##  5     2 corbata
##  6     2 toda
##  7     2 la
##  8     2 semana
##  9     3 noticias
## 10     4 tú
```

```
text_2001 %<>%
  unnest_tokens(input = text, output = word) %>%
  filter(!is.na(word))
dim(text_2001)
```

```
## [1] 15099     2
```

```
text_2005 %<>%
  unnest_tokens(input = text, output = word) %>%
  filter(!is.na(word))
dim(text_2005)
```

```
## [1] 8102    2
```

```
text_2008 %<>%
  unnest_tokens(input = text, output = word) %>%
  filter(!is.na(word))
dim(text_2008)
```

```
## [1] 14515     2
```

```
text_2010 %<>%
  unnest_tokens(input = text, output = word) %>%
  filter(!is.na(word))
dim(text_2010)
```

```
## [1] 6412    2
```

```
head(text_2001, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 buenos
##  2     1 días
##  3     2 estamos
##  4     2 muy
##  5     2 contentos
##  6     2 de
##  7     2 estar
##  8     2 aquí
##  9     2 en
## 10     2 nueva
```

```
head(text_2005, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 bienvenidos
##  2     1 a
##  3     1 nuestra
##  4     1 conferencia
##  5     1 mundial
##  6     1 de
##  7     1 desarrolladores
##  8     1 2005
##  9     1 hoy
```

```
## 10     1 es
```

```
head(text_2008, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 estoy
##  2     1 muy
##  3     1 contento
##  4     1 de
##  5     1 estar
##  6     1 aquí
##  7     1 esta
##  8     1 vez
##  9     2 buenos
## 10     2 días
```

```
head(text_2010, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 así
##  2     1 que
##  3     1 volvamos
##  4     1 al
##  5     1 iphone
##  6     2 en
##  7     2 2007
##  8     2 el
##  9     2 iphone
## 10     2 reinventó
```

#Nomrmalizacion de texto

```
##### texto con numeros?

text_1997 %>%
  filter(grepl(pattern = '[0-9]', x = word)) %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 35 x 2
##    word      n
##    <chr> <int>
##  1 10       13
##  2 18        6
##  3 20        4
##  4 100       3
##  5 30        3
##  6 5         3
##  7 500       3
##  8 12        2
##  9 14        2
## 10 15        2
## # i 25 more rows
```

```
text_2001 %>%
  filter(grepl(pattern = '[0-9]', x = word)) %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 95 x 2
##    word      n
##    <chr> <int>
##  1 10      102
##  2 g4       33
##  3 4        14
##  4 1        13
##  5 3        13
##  6 867      13
##  7 os10     13
##  8 3d       12
##  9 7        12
## 10 99       10
## # i 85 more rows
```

```
text_2005 %>%
  filter(grepl(pattern = '[0-9]', x = word)) %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 58 x 2
##    word      n
##    <chr> <int>
##  1 10       28
##  2 20        8
##  3 2.1       6
##  4 2         5
##  5 2006      4
##  6 400       4
##  7 264       3
##  8 500       3
##  9 7         3
## 10 9         3
## # i 48 more rows
```

```
text_2008 %>%
  filter(grepl(pattern = '[0-9]', x = word)) %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 72 x 2
##    word      n
##    <chr> <int>
##  1 3g       33
##  2 2.0      21
##  3 10       20
##  4 70        5
##  5 100       4
##  6 11        4
##  7 199       4
##  8 20        4
##  9 3d        4
## 10 5         4
```

```
## # i 62 more rows
```

```
text_2010 %>%
  filter(grepl(pattern = '[0-9]', x = word)) %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 43 x 2
##    word      n
##    <chr> <int>
##  1 4        42
##  2 3gs      11
##  3 a4        9
##  4 3g        8
##  5 199       5
##  6 2010      5
##  7 2007      4
##  8 24        4
##  9 30        4
## 10 720p      4
## # i 33 more rows
```

```
##### remover texto con numeros

text_1997 %<>%
  filter(!grepl(pattern = '[0-9]', x = word))
dim(text_1997)
```

```
## [1] 11285     2
```

```
text_2001 %<>%
  filter(!grepl(pattern = '[0-9]', x = word))
dim(text_2001)
```

```
## [1] 14670     2
```

```
text_2005 %<>%
  filter(!grepl(pattern = '[0-9]', x = word))
dim(text_2005)
```

```
## [1] 7977    2
```

```
text_2008 %<>%
  filter(!grepl(pattern = '[0-9]', x = word))
dim(text_2008)
```

```
## [1] 14323     2
```

```
text_2010 %<>%
  filter(!grepl(pattern = '[0-9]', x = word))
dim(text_2010)
```

```
## [1] 6262    2
```

```
dim(stop_words)
```

```
## [1] 1149    2
```

```
head(stop_words, n = 10)
```

```
## # A tibble: 10 x 2
```

```
##    word        lexicon
##    <chr>       <chr>
##  1 a           SMART
##  2 a's         SMART
##  3 able        SMART
##  4 about       SMART
##  5 above       SMART
##  6 according   SMART
##  7 accordingly SMART
##  8 across      SMART
##  9 actually    SMART
## 10 after       SMART
```

```
table(stop_words$lexicon)
```

```
##
##     onix    SMART snowball
##      404      571      174
```

```
###### stop words
# no hay diccionarios en español disponibles en tidytext
# diccionario COUNTWORDSFREE en español (con acentos)
# http://countwordsfree.com/stopwords/spanish
# otras alternativas:
#   https://github.com/stopwords-iso/stopwords-es
#   de tm::stopwords("spanish")
# se conserva el mismo formato de los diccionarios en tidytext
stop_words_es <- tibble(word = unlist(c(read.table("stop_words_spanish.txt", quote="\"", comment.char="
dim(stop_words_es)
```

```
## [1] 444   2
```

```
head(stop_words_es, n = 10)
```

```
## # A tibble: 10 x 2
##    word      lexicon
##    <chr>     <chr>
##  1 algún     custom
##  2 alguna    custom
##  3 algunas   custom
##  4 alguno    custom
##  5 algunos   custom
##  6 ambos     custom
##  7 ampleamos custom
##  8 ante      custom
##  9 antes     custom
## 10 aquel     custom
```

```
##### remover stop words

text_1997 %<>%
  anti_join(x = ., y = stop_words_es)
```

```
## Joining with `by = join_by(word)`
```

```
dim(text_1997)
```

```
## [1] 4009    2
```

```
head(text_1997, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 días
##  2     2 llevaban
##  3     2 corbata
##  4     2 semana
##  5     3 noticias
##  6     4 tú
##  7     5 corto
##  8     5 haré
##  9     5 uh
## 10     5 sexto
```

```
text_2001 %<>%
  anti_join(x = ., y = stop_words_es)
```

```
## Joining with `by = join_by(word)`
```

```
dim(text_2001)
```

```
## [1] 5957    2
```

```
head(text_2001, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 días
##  2     2 contentos
##  3     2 york
##  4     2 geniales
##  5     2 compartir
##  6     3 ustedes
##  7     3 mañana
##  8     4 tiendas
##  9     5 tiendas
## 10     5 tyson's
```

```
text_2005 %<>%
  anti_join(x = ., y = stop_words_es)
```

```
## Joining with `by = join_by(word)`
```

```
dim(text_2005)
```

```
## [1] 3080    2
```

```
head(text_2005, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 bienvenidos
##  2     1 conferencia
##  3     1 mundial
```

```
##  4     1 desarrolladores
##  5     1 día
##  6     1 importante
##  7     2 geniales
##  8     2 ti
##  9     2 quiero
## 10     2 comenzar
```

```
text_2008 %<>%
  anti_join(x = ., y = stop_words_es)
```

```
## Joining with `by = join_by(word)`
```

```
dim(text_2008)
```

```
## [1] 5820    2
```

```
head(text_2008, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 contento
##  2     2 días
##  3     2 trabajando
##  4     2 duro
##  5     2 geniales
##  6     2 ansiosos
##  7     2 compartir
##  8     2 ustedes
##  9     2 gracias
## 10     2 venir
```

```
text_2010 %<>%
  anti_join(x = ., y = stop_words_es)
```

```
## Joining with `by = join_by(word)`
```

```
dim(text_2010)
```

```
## [1] 2421    2
```

```
head(text_2010, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 volvamos
##  2     1 iphone
##  3     2 iphone
##  4     2 reinventó
##  5     2 consideramos
##  6     2 teléfono
##  7     3 difícil
##  8     3 recordar
##  9     3 operadores
## 10     3 iphone
```

```
##### remover acentos
replacement_list <- list('á' = 'a', 'é' = 'e', 'í' = 'i', 'ó' = 'o', 'ú' = 'u')

text_1997 %<>%
  mutate(word = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                       new = replacement_list %>% str_c(collapse = ''),
                       x = word))
dim(text_1997)
```

```
## [1] 4009    2
```

```
head(text_1997, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 dias
##  2     2 llevaban
##  3     2 corbata
##  4     2 semana
##  5     3 noticias
##  6     4 tu
##  7     5 corto
##  8     5 hare
##  9     5 uh
## 10     5 sexto
```

```
text_2001 %<>%
  mutate(word = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                       new = replacement_list %>% str_c(collapse = ''),
                       x = word))
dim(text_2001)
```

```
## [1] 5957    2
```

```
head(text_2001, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 dias
##  2     2 contentos
##  3     2 york
##  4     2 geniales
##  5     2 compartir
##  6     3 ustedes
##  7     3 mañana
##  8     4 tiendas
##  9     5 tiendas
## 10     5 tyson's
```

```
text_2005 %<>%
  mutate(word = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                       new = replacement_list %>% str_c(collapse = ''),
                       x = word))
dim(text_2005)
```

```
## [1] 3080    2
```

```
head(text_2005, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 bienvenidos
##  2     1 conferencia
##  3     1 mundial
##  4     1 desarrolladores
##  5     1 dia
##  6     1 importante
##  7     2 geniales
##  8     2 ti
##  9     2 quiero
## 10     2 comenzar
```

```
text_2008 %<>%
  mutate(word = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                       new = replacement_list %>% str_c(collapse = ''),
                       x = word))
dim(text_2008)
```

```
## [1] 5820    2
```

```
head(text_2008, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 contento
##  2     2 dias
##  3     2 trabajando
##  4     2 duro
##  5     2 geniales
##  6     2 ansiosos
##  7     2 compartir
##  8     2 ustedes
##  9     2 gracias
## 10     2 venir
```

```
text_2010 %<>%
  mutate(word = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                       new = replacement_list %>% str_c(collapse = ''),
                       x = word))
dim(text_2010)
```

```
## [1] 2421    2
```

```
head(text_2010, n = 10)
```

```
## # A tibble: 10 x 2
##     line word
##    <int> <chr>
##  1     1 volvamos
##  2     1 iphone
##  3     2 iphone
```

```
##  4     2 reinvento
##  5     2 consideramos
##  6     2 telefono
##  7     3 dificil
##  8     3 recordar
##  9     3 operadores
## 10     3 iphone
```

#Tokens mas frecuentes

```
##### top 10 de tokens mas frecuentes

text_1997 %>%
  count(word, sort = TRUE) %>%
  head(n = 10)
```

```
## # A tibble: 10 x 2
##    word          n
##    <chr>     <int>
##  1 apple       107
##  2 realmente    59
##  3 um           34
##  4 eh           32
##  5 personas     31
##  6 software     31
##  7 hardware     29
##  8 quiero       28
##  9 gente        26
## 10 mundo        23
```

```
text_2001 %>%
  count(word, sort = TRUE)  %>%
  head(n = 10)
```

```
## # A tibble: 10 x 2
##    word             n
##    <chr>        <int>
##  1 mac             91
##  2 os              81
##  3 puedes          67
##  4 gracias         46
##  5 rapido          37
##  6 aplicaciones    34
##  7 realmente       30
##  8 apple           28
##  9 sistema         27
## 10 te              24
```

```
text_2005 %>%
  count(word, sort = TRUE)  %>%
  head(n = 10)
```

```
## # A tibble: 10 x 2
##    word             n
##    <chr>        <int>
##  1 apple           43
##  2 intel           33
```

```
##  3 mac             29
##  4 aplicaciones    26
##  5 años            24
##  6 powerpc         24
##  7 xcode           24
##  8 os              23
##  9 transicion      22
## 10 procesadores    18
```

```
text_2008 %>%
  count(word, sort = TRUE)  %>%
  head(n = 10)
```

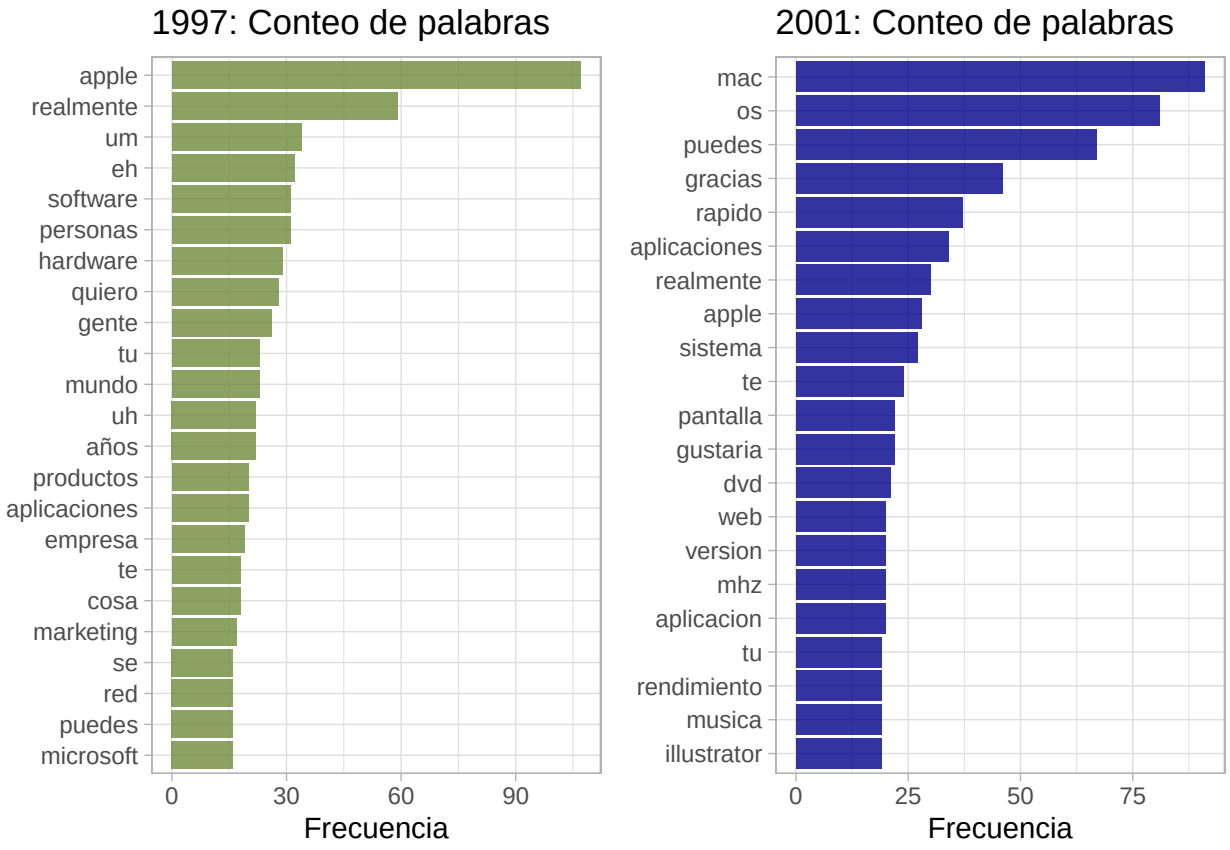
```
## # A tibble: 10 x 2
##    word             n
##    <chr>        <int>
##  1 iphone         166
##  2 aplicacion      71
##  3 aplicaciones    55
##  4 realmente       45
##  5 telefono        36
##  6 correo          34
##  7 puedes          34
##  8 sdk             34
##  9 directamente    32
## 10 juego           32
```

```
text_2010 %>%
  count(word, sort = TRUE)  %>%
  head(n = 10)
```

```
## # A tibble: 10 x 2
##    word             n
##    <chr>        <int>
##  1 iphone          70
##  2 pantalla        29
##  3 telefono        26
##  4 realmente       25
##  5 pixeles         24
##  6 puedes          23
##  7 video           21
##  8 camara          20
##  9 tu              18
## 10 aplicaciones    16
```

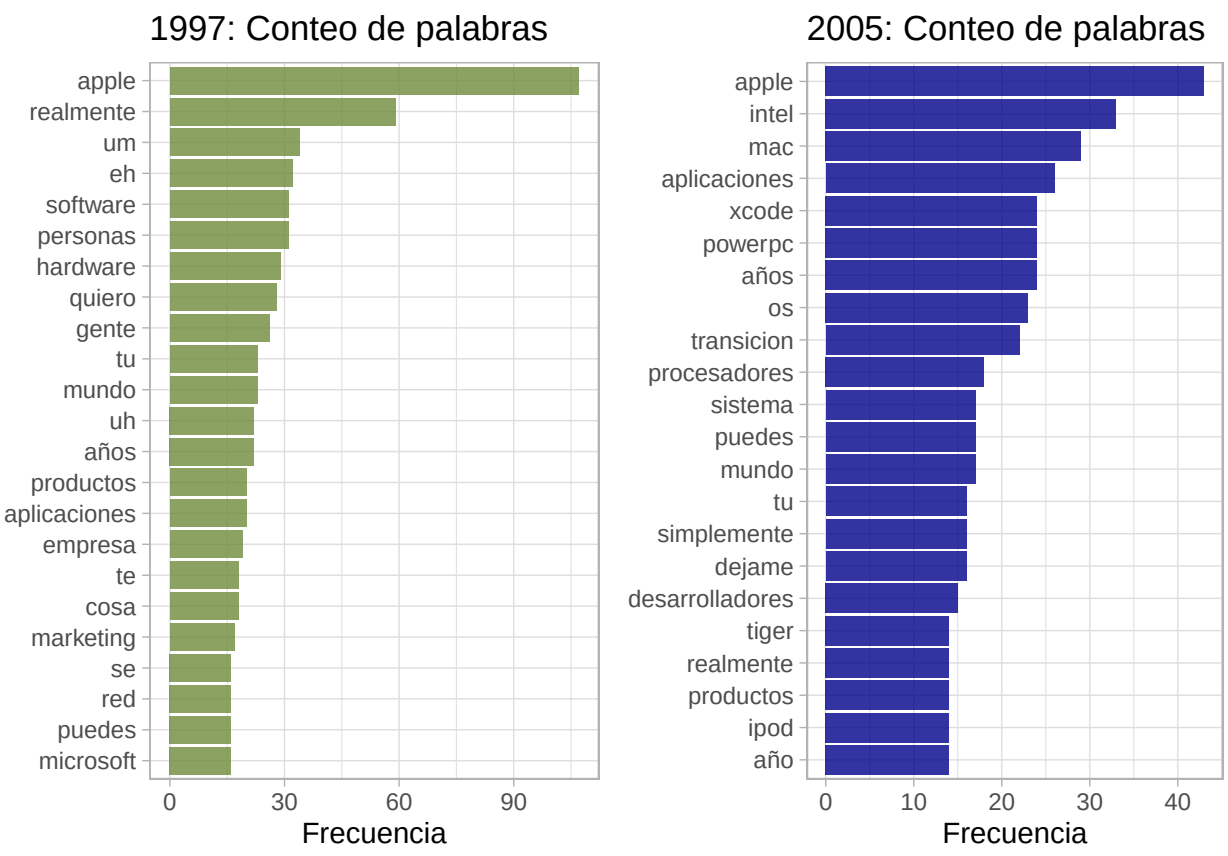
```
##### viz
suppressMessages(suppressWarnings(library(gridExtra)))


p1 <- text_1997 %>%
  count(word, sort = TRUE) %>%
  slice_max(order_by = n, n = 20) %>%  # Mostrar solo las 10 palabras más frecuentes
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n)) +
    theme_light() +
    geom_col(fill = 'darkolivegreen4', alpha = 0.8) +
```

```
    xlab(NULL) +
    ylab("Frecuencia") +
    coord_flip() +
    ggtitle(label = '1997: Conteo de palabras')


p2 <- text_2001 %>%
  count(word, sort = TRUE) %>%
  slice_max(order_by = n, n = 20) %>%  # Mostrar solo las 10 palabras más frecuentes
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n)) +
    theme_light() +
    geom_col(fill = 'blue4', alpha = 0.8) +
    xlab(NULL) +
    ylab("Frecuencia") +
    coord_flip() +
    ggtitle(label = '2001: Conteo de palabras')

# Desplegar gráfico
grid.arrange(p1, p2, ncol = 2)
```

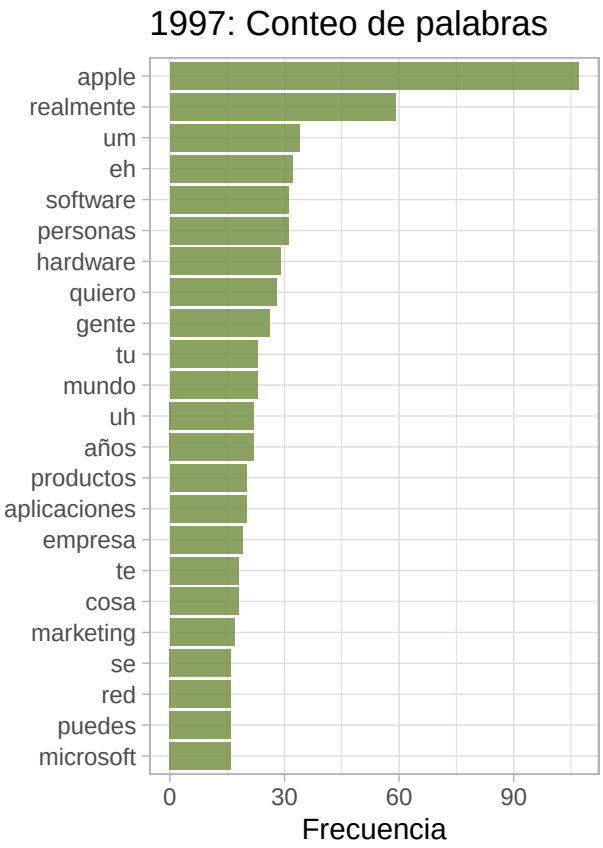


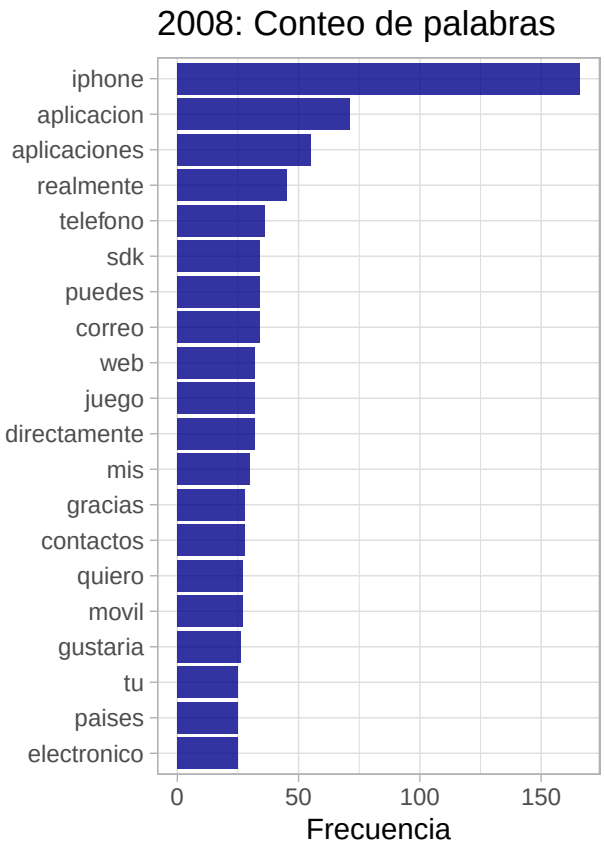

```
##### viz
suppressMessages(suppressWarnings(library(gridExtra)))


p1 <- text_1997 %>%
  count(word, sort = TRUE) %>%
```

```r
  slice_max(order_by = n, n = 20) %>%  # Mostrar solo las 10 palabras más frecuentes
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n)) +
    theme_light() +
    geom_col(fill = 'darkolivegreen4', alpha = 0.8) +
    xlab(NULL) +
    ylab("Frecuencia") +
    coord_flip() +
    ggtitle(label = '1997: Conteo de palabras')


p2 <- text_2005 %>%
  count(word, sort = TRUE) %>%
  slice_max(order_by = n, n = 20) %>%  # Mostrar solo las 10 palabras más frecuentes
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n)) +
    theme_light() +
    geom_col(fill = 'blue4', alpha = 0.8) +
    xlab(NULL) +
    ylab("Frecuencia") +
    coord_flip() +
    ggtitle(label = '2005: Conteo de palabras')

# Desplegar gráficos
grid.arrange(p1, p2, ncol = 2)
```



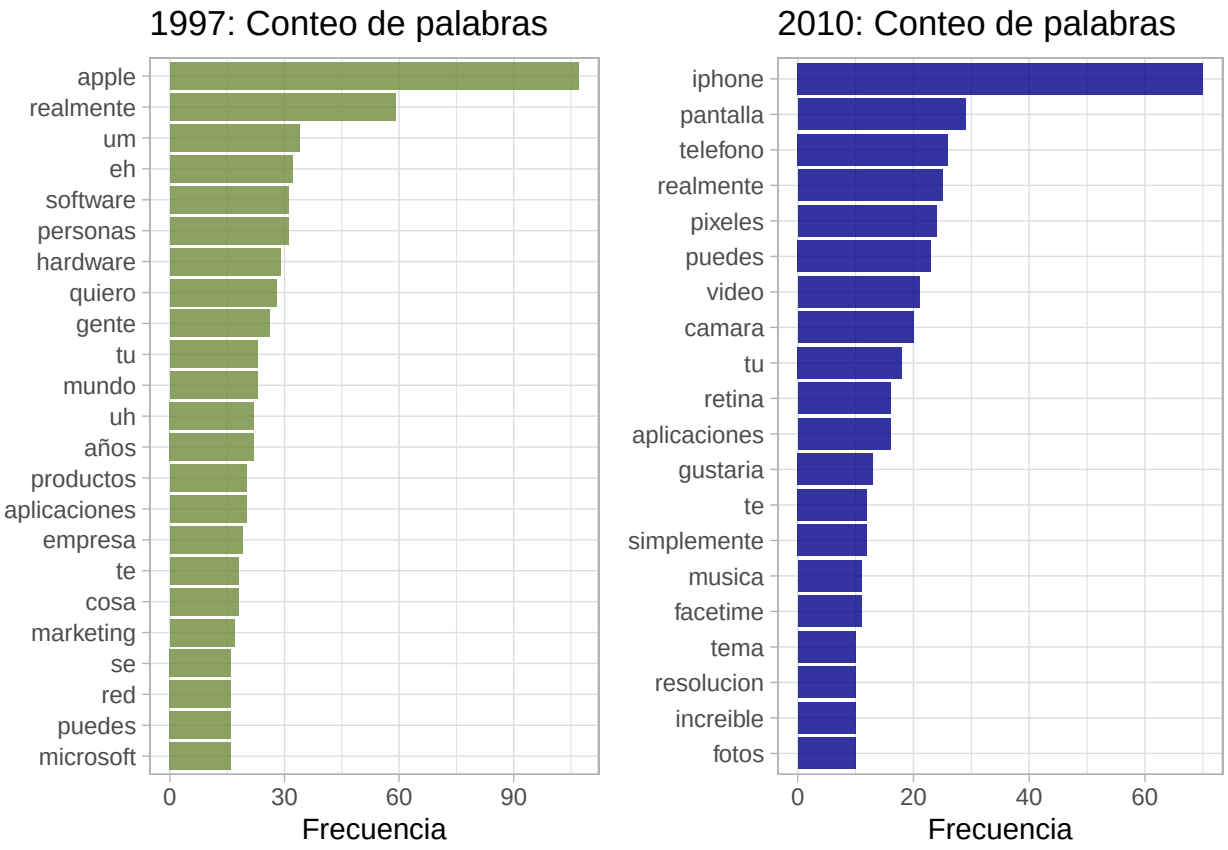
```r
##### viz
suppressMessages(suppressWarnings(library(gridExtra)))


p1 <- text_1997 %>%
  count(word, sort = TRUE) %>%
  slice_max(order_by = n, n = 20) %>%  # Mostrar solo las 20 palabras más frecuentes
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n)) +
    theme_light() +
    geom_col(fill = 'darkolivegreen4', alpha = 0.8) +
    xlab(NULL) +
    ylab("Frecuencia") +
    coord_flip() +
    ggtitle(label = '1997: Conteo de palabras')


p2 <- text_2008 %>%
  count(word, sort = TRUE) %>%
  slice_max(order_by = n, n = 20) %>%  # Mostrar solo las 20 palabras más frecuentes
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n)) +
    theme_light() +
    geom_col(fill = 'blue4', alpha = 0.8) +
    xlab(NULL) +
    ylab("Frecuencia") +
    coord_flip() +
    ggtitle(label = '2008: Conteo de palabras')

# Desplegar gráficos
grid.arrange(p1, p2, ncol = 2)
```






```
##### viz
suppressMessages(suppressWarnings(library(gridExtra)))


p1 <- text_1997 %>%
  count(word, sort = TRUE) %>%
  slice_max(order_by = n, n = 20) %>%  # Mostrar solo las 20 palabras más frecuentes
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n)) +
    theme_light() +
    geom_col(fill = 'darkolivegreen4', alpha = 0.8) +
    xlab(NULL) +
    ylab("Frecuencia") +
    coord_flip() +
    ggtitle(label = '1997: Conteo de palabras')


p2 <- text_2010 %>%
  count(word, sort = TRUE) %>%
  slice_max(order_by = n, n = 20) %>%  # Mostrar solo las 20 palabras más frecuentes
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x = word, y = n)) +
    theme_light() +
    geom_col(fill = 'blue4', alpha = 0.8) +
    xlab(NULL) +
    ylab("Frecuencia") +
    coord_flip() +
```

```
    ggtitle(label = '2010: Conteo de palabras')

# Desplegar gráficos
grid.arrange(p1, p2, ncol = 2)
```




```
suppressMessages(suppressWarnings(library(wordcloud)))
###### viz
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))

set.seed(123)
text_1997 %>%
  count(word, sort = TRUE) %>%
  with(wordcloud(words = word, freq = n, max.words = 20, colors = 'darkolivegreen4'))
title(main = "1997")

set.seed(123)
text_2001 %>%
  count(word, sort = TRUE) %>%
  with(wordcloud(words = word, freq = n, max.words = 20, colors = 'blue4'))
title(main = "2001")
```

1997 **2001**


```
suppressMessages(suppressWarnings(library(wordcloud)))
###### viz
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))

set.seed(124)
text_1997 %>%
  count(word, sort = TRUE) %>%
  with(wordcloud(words = word, freq = n, max.words = 20, colors = 'darkolivegreen4'))
title(main = "1997")

set.seed(124)
text_2005 %>%
  count(word, sort = TRUE) %>%
  with(wordcloud(words = word, freq = n, max.words = 20, colors = 'blue4'))
```

```
## Warning in wordcloud(words = word, freq = n, max.words = 20, colors = "blue4"):
## apple could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = word, freq = n, max.words = 20, colors = "blue4"):
## aplicaciones could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = word, freq = n, max.words = 20, colors = "blue4"):
## procesadores could not be fit on page. It will not be plotted.
```

```
title(main = "2005")
```


```
suppressMessages(suppressWarnings(library(wordcloud)))
###### viz
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))

set.seed(123)
text_1997 %>%
  count(word, sort = TRUE) %>%
  with(wordcloud(words = word, freq = n, max.words = 20, colors = 'darkolivegreen4'))
title(main = "1997")

set.seed(123)
text_2008 %>%
  count(word, sort = TRUE) %>%
  with(wordcloud(words = word, freq = n, max.words = 20, colors = 'blue4'))
title(main = "2008")
```

1997 **2008**


```
suppressMessages(suppressWarnings(library(wordcloud)))
###### viz
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))

set.seed(123)
text_1997 %>%
  count(word, sort = TRUE) %>%
  with(wordcloud(words = word, freq = n, max.words = 20, colors = 'darkolivegreen4'))
title(main = "1997")

set.seed(123)
text_2010 %>%
  count(word, sort = TRUE) %>%
  with(wordcloud(words = word, freq = n, max.words = 20, colors = 'blue4'))
```

```
## Warning in wordcloud(words = word, freq = n, max.words = 20, colors = "blue4"):
## iphone could not be fit on page. It will not be plotted.
```

```
title(main = "2010")
```

1997 **2010**


```
##### frecuencias relativas de la palabras
bind_rows(mutate(.data = text_1997, author = "1997"),
                      mutate(.data = text_2001, author = "2001")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n/sum(n)) %>%
  select(-n) %>%
  spread(author, proportion, fill = 0) -> frec  # importante!
frec %<>%
  select(word, "1997", "2001")
dim(frec)
```

```
## [1] 3479    3
```

```
head(frec, n = 10)
```

```
## # A tibble: 10 x 3
##    word          `1997`   `2001`
##    <chr>          <dbl>    <dbl>
##  1 abajo       0.000499 0.000336
##  2 abarca      0.000249 0
##  3 abdominales 0.000249 0
##  4 abiertas    0.000249 0.000168
##  5 abierto     0.000998 0.000168
##  6 abiertos    0        0.000168
##  7 abogando    0.000249 0
##  8 aborda      0.000249 0
##  9 abordar     0.000249 0
## 10 abramos     0        0.000168
```

```
##### top 10 palabras en comun
# orden anidado respecto a petro y duque
frec %>%
```

```
  filter(1997 !=0, 2001 != 0) %>%
  arrange(desc(1997), desc(2001)) -> frec_comun
dim(frec_comun)
```

```
## [1] 3479    3
```

```
head(frec_comun, n = 10)
```

```
## # A tibble: 10 x 3
##    word          `1997`   `2001`
##    <chr>          <dbl>    <dbl>
##  1 abajo       0.000499 0.000336
##  2 abarca      0.000249 0
##  3 abdominales 0.000249 0
##  4 abiertas    0.000249 0.000168
##  5 abierto     0.000998 0.000168
##  6 abiertos    0        0.000168
##  7 abogando    0.000249 0
##  8 aborda      0.000249 0
##  9 abordar     0.000249 0
## 10 abramos     0        0.000168
```

```
###### proporcion palabras en comun
dim(frec_comun)[1]/dim(frec)[1]
```

```
## [1] 1
```

```
##### frecuencias relativas de la palabras
bind_rows(mutate(.data = text_1997, author = "1997"),
                    mutate(.data = text_2005, author = "2005")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n/sum(n)) %>%
  select(-n) %>%
  spread(author, proportion, fill = 0) -> frec  # importante!
frec %<>%
  select(word, "1997", "2005")
dim(frec)
```

```
## [1] 2705    3
```

```
head(frec, n = 10)
```

```
## # A tibble: 10 x 3
##    word          `1997`   `2005`
##    <chr>          <dbl>    <dbl>
##  1 abajo       0.000499 0
##  2 abarca      0.000249 0
##  3 abarrotada  0        0.000325
##  4 abdominales 0.000249 0
##  5 abiertas    0.000249 0
##  6 abierto     0.000998 0.000974
##  7 abiertos    0        0.000649
##  8 abogando    0.000249 0
##  9 aborda      0.000249 0
## 10 abordar     0.000249 0
```

```
##### top 10 palabras en comun
# orden anidado respecto a petro y duque
frec %>%
  filter(1997 !=0, 2005 != 0) %>%
  arrange(desc(1997), desc(2005)) -> frec_comun1
dim(frec_comun1)
```

```
## [1] 2705    3
```

```
head(frec_comun1, n = 10)
```

```
## # A tibble: 10 x 3
##    word          `1997`   `2005`
##    <chr>          <dbl>    <dbl>
##  1 abajo       0.000499 0
##  2 abarca      0.000249 0
##  3 abarrotada  0        0.000325
##  4 abdominales 0.000249 0
##  5 abiertas    0.000249 0
##  6 abierto     0.000998 0.000974
##  7 abiertos    0        0.000649
##  8 abogando    0.000249 0
##  9 aborda      0.000249 0
## 10 abordar     0.000249 0
```

```
###### proporcion palabras en comun
dim(frec_comun1)[1]/dim(frec)[1]
```

```
## [1] 1
```

```
##### frecuencias relativas de la palabras
bind_rows(mutate(.data = text_1997, author = "1997"),
                      mutate(.data = text_2008, author = "2008")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n/sum(n)) %>%
  select(-n) %>%
  spread(author, proportion, fill = 0) -> frec2  # importante!
frec2 %<>%
  select(word, "1997", "2008")
dim(frec2)
```

```
## [1] 3395    3
```

```
head(frec2, n = 10)
```

```
## # A tibble: 10 x 3
##    word          `1997`   `2008`
##    <chr>          <dbl>    <dbl>
##  1 aaron       0        0.000172
##  2 abajo       0.000499 0.000687
##  3 abandona    0        0.000172
##  4 abandonado  0        0.000172
##  5 abarca      0.000249 0
##  6 abdominales 0.000249 0
##  7 abiertas    0.000249 0
##  8 abierto     0.000998 0.000172
```

```
##  9 abiertos    0        0.000172
## 10 abogados    0        0.000172
```

```
##### top 10 palabras en comun

frec2 %>%
  filter(1997 !=0, 2008 != 0) %>%
  arrange(desc(1997), desc(2008)) -> frec_comun2
dim(frec_comun2)
```

```
## [1] 3395    3
```

```
head(frec_comun2, n = 10)
```

```
## # A tibble: 10 x 3
##    word          `1997`   `2008`
##    <chr>          <dbl>    <dbl>
##  1 aaron       0        0.000172
##  2 abajo       0.000499 0.000687
##  3 abandona    0        0.000172
##  4 abandonado  0        0.000172
##  5 abarca      0.000249 0
##  6 abdominales 0.000249 0
##  7 abiertas    0.000249 0
##  8 abierto     0.000998 0.000172
##  9 abiertos    0        0.000172
## 10 abogados    0        0.000172
```

```
###### proporcion palabras en comun
dim(frec_comun2)[1]/dim(frec2)[1]
```

```
## [1] 1
```

```
##### frecuencias relativas de la palabras
bind_rows(mutate(.data = text_1997, author = "1997"),
                      mutate(.data = text_2010, author = "2010")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n/sum(n)) %>%
  select(-n) %>%
  spread(author, proportion, fill = 0) -> frec3  # importante!
frec3 %<>%
  select(word, "1997", "2010")
dim(frec3)
```

```
## [1] 2542    3
```

```
head(frec3, n = 10)
```

```
## # A tibble: 10 x 3
##    word          `1997`   `2010`
##    <chr>          <dbl>    <dbl>
##  1 aac         0        0.000413
##  2 abajo       0.000499 0.00165
##  3 abarca      0.000249 0
##  4 abdominales 0.000249 0
##  5 abiertas    0.000249 0.000413
##  6 abierto     0.000998 0.000413
```

```
##  7 abiertos    0        0.000413
##  8 abogando    0.000249 0
##  9 aborda      0.000249 0
## 10 abordar     0.000249 0
```

```
##### top 10 palabras en comun

frec3 %>%
  filter(1997 !=0, 2010 != 0) %>%
  arrange(desc(1997), desc(2010)) -> frec_comun3
dim(frec_comun3)
```

```
## [1] 2542    3
```

```
head(frec_comun3, n = 10)
```

```
## # A tibble: 10 x 3
##    word          `1997`   `2010`
##    <chr>          <dbl>    <dbl>
##  1 aac         0        0.000413
##  2 abajo       0.000499 0.00165
##  3 abarca      0.000249 0
##  4 abdominales 0.000249 0
##  5 abiertas    0.000249 0.000413
##  6 abierto     0.000998 0.000413
##  7 abiertos    0        0.000413
##  8 abogando    0.000249 0
##  9 aborda      0.000249 0
## 10 abordar     0.000249 0
```

```
###### proporcion palabras en comun
dim(frec_comun3)[1]/dim(frec3)[1]
```

```
## [1] 1
```

```
##### Asignar frecuencias relativas a las variables correspondientes
frec <- bind_rows(
  mutate(text_1997, author = "freq_1997"),
  mutate(text_2001, author = "freq_2001"),
  mutate(text_2005, author = "freq_2005"),
  mutate(text_2008, author = "freq_2008"),
  mutate(text_2010, author = "freq_2010")
) %>%
  # Calcular frecuencias relativas
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n / sum(n)) %>%
  select(-n) %>%
  spread(author, proportion, fill = 0)  # Crear columnas separadas por autor

# Mostrar las dimensiones para verificar
dim(frec)
```

```
## [1] 5644    6
```

```
# Verificar las primeras filas
head(frec)
```

```
## # A tibble: 6 x 6
##   word       freq_1997 freq_2001 freq_2005 freq_2008 freq_2010
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
## 1 aac         0         0                 0  0         0.000413
## 2 aaron       0         0                 0  0.000172  0
## 3 abajo       0.000499  0.000336          0  0.000687  0.00165
## 4 abandona    0         0                 0  0.000172  0
## 5 abandonado  0         0                 0  0.000172  0
## 6 abarca      0.000249  0                 0  0         0
```

```
cor.test(x = frec$freq_1997, y = frec$freq_2001)
```

```
##
##  Pearson's product-moment correlation
##
## data:  frec$freq_1997 and frec$freq_2001
## t = 29.867, df = 5642, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3467456 0.3918056
## sample estimates:
##       cor
## 0.3694928
```

```
cor.test(x = frec$freq_1997, y = frec$freq_2005)
```

```
##
##  Pearson's product-moment correlation
##
## data:  frec$freq_1997 and frec$freq_2005
## t = 41.369, df = 5642, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4621531 0.5021951
## sample estimates:
##       cor
## 0.4824261
```

```
cor.test(x = frec$freq_1997, y = frec$freq_2008)
```

```
##
##  Pearson's product-moment correlation
##
## data:  frec$freq_1997 and frec$freq_2008
## t = 20.806, df = 5642, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2425473 0.2910109
## sample estimates:
##       cor
## 0.2669479
```

```
cor.test(x = frec$freq_1997, y = frec$freq_2010)
```

```
##
##  Pearson's product-moment correlation
##
```

```
## data:  frec$freq_1997 and frec$freq_2010
## t = 16.712, df = 5642, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1921831 0.2419032
## sample estimates:
##      cor
## 0.217184
```

```
# Verifica los nombres de las columnas
names(frec_comun)
```

```
## [1] "word" "1997" "2001"
```

```
# Si las columnas son diferentes, renómbralas
colnames(frec_comun) <- c("word", "freq_1997", "freq_2001", "freq_2005", "freq_2008", "freq_2010")

# Filtra las filas con valores no NA
frec_comun <- frec_comun %>%
  filter(!is.na(freq_1997) & !is.na(freq_2001))

# Realiza el análisis de correlación
cor.test(x = frec_comun$freq_1997, y = frec_comun$freq_2001)
```

```
##
##  Pearson's product-moment correlation
##
## data:  frec_comun$freq_1997 and frec_comun$freq_2001
## t = 20.589, df = 3477, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2997067 0.3589539
## sample estimates:
##       cor
## 0.3296548
```

#7 Analisis de sentimiento

```
# diccionarios
# no hay diccionarios en español disponibles en tidytext
# https://www.kaggle.com/datasets/rtatman/sentiment-lexicons-for-81-languages
positive_words <- read_csv("positive_words_es.txt", col_names = "word", show_col_types = FALSE) %>%
  mutate(sentiment = "Positivo")
negative_words <- read_csv("negative_words_es.txt", col_names = "word", show_col_types = FALSE) %>%
  mutate(sentiment = "Negativo")
sentiment_words <- bind_rows(positive_words, negative_words)
# comparacion de diccionarios
get_sentiments("bing") %>%
  count(sentiment)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   4781
## 2 positive   2005
```

```r
sentiment_words %>%
  count(sentiment)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 Negativo   2720
## 2 Positivo   1555
```

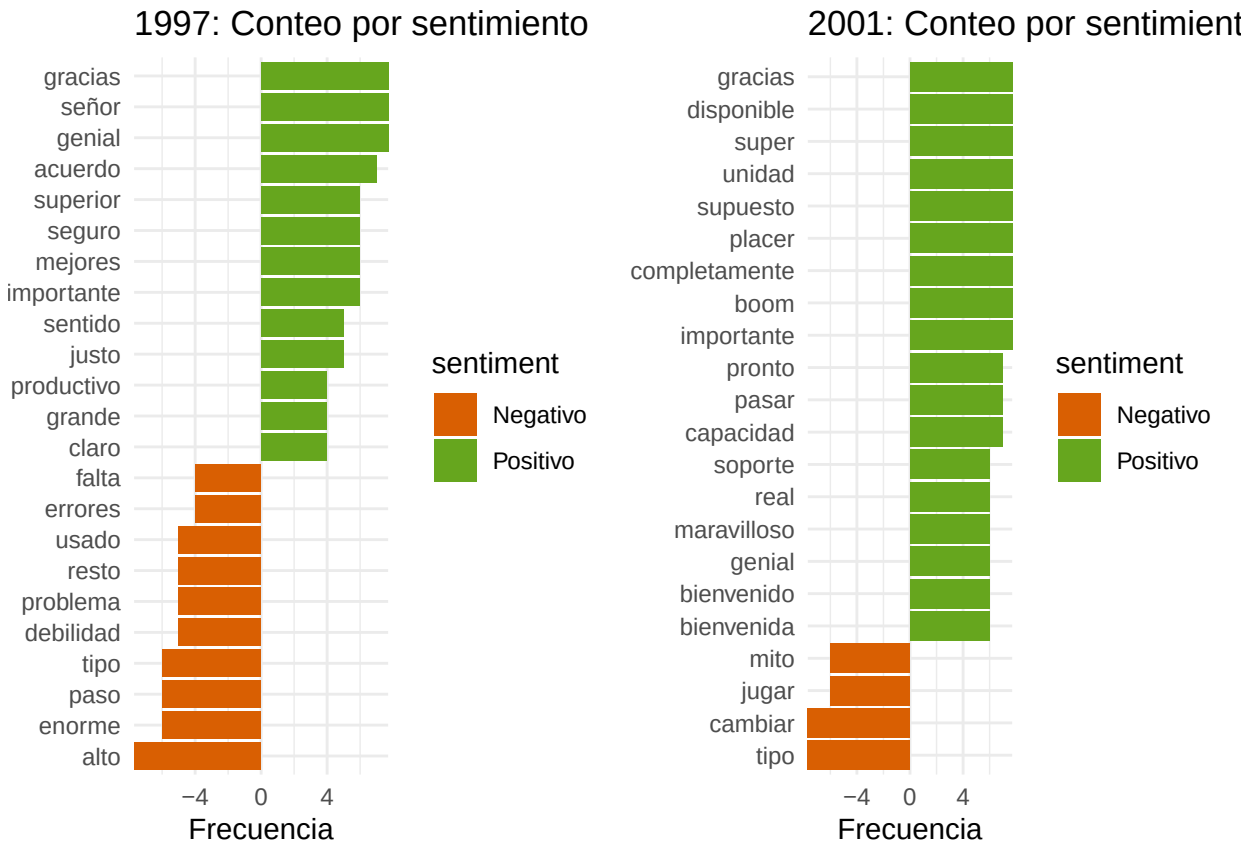
```r
###### viz
suppressMessages(suppressWarnings(library(RColorBrewer)))

# ---------- petro ----------
p1 <- text_1997 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  slice_max(order_by = abs(n), n = 20) %>%  # Mostrar las 50 palabras más frecuentes
  mutate(n = ifelse(sentiment == "Negativo", -n, n)) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
    geom_col() +
    scale_fill_manual(values = brewer.pal(8, 'Dark2')[c(2, 5)]) +
    coord_flip(ylim = c(-7, 7)) +  # Ajusta los límites si es necesario
    labs(y = "Frecuencia",
         x = NULL,
         title = "1997: Conteo por sentimiento") +
    theme_minimal()
```

```
## Joining with `by = join_by(word)`
```

```r
##### viz
library(RColorBrewer) # Asegúrate de cargar RColorBrewer si no está cargado

p2 <- text_2001 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  slice_max(order_by = abs(n), n = 20) %>%  # Seleccionar las 50 palabras más frecuentes
  mutate(n = ifelse(sentiment == "Negativo", -n, n)) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
    geom_col() +
    scale_fill_manual(values = brewer.pal(8,'Dark2')[c(2,5)]) +
    coord_flip(ylim = c(-7,7)) +  # Ajustar los límites si es necesario
    labs(y = "Frecuencia",
         x = NULL,
         title = "2001: Conteo por sentimiento") +
    theme_minimal()
```

```
## Joining with `by = join_by(word)`
```

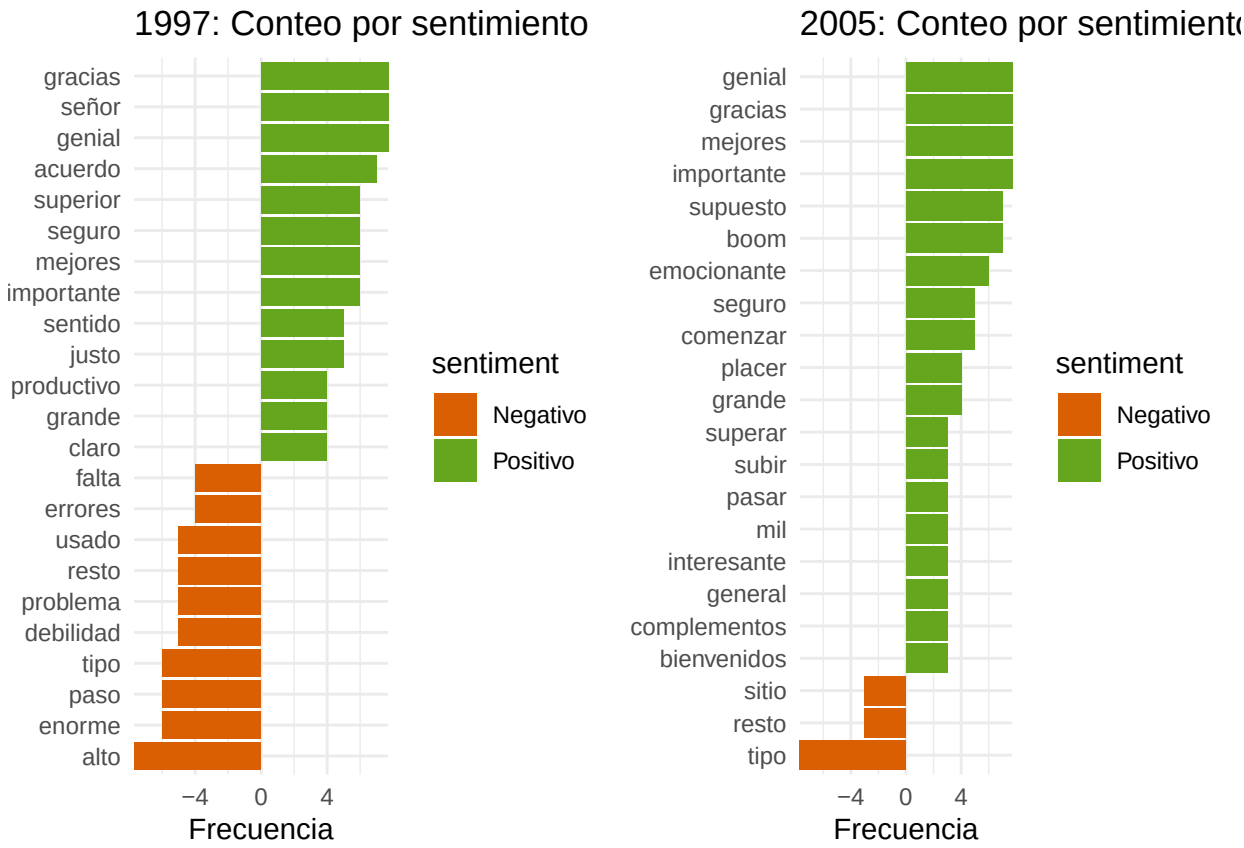
```r
grid.arrange(p1, p2, ncol = 2)
```




```
text_2005 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  slice_max(order_by = abs(n), n = 20) %>%  # Mostrar las 50 palabras más frecuentes
  mutate(n = ifelse(sentiment == "Negativo", -n, n)) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
    geom_col() +
    scale_fill_manual(values = brewer.pal(8, 'Dark2')[c(2, 5)]) +
    coord_flip(ylim = c(-7, 7)) +  # Ajusta los límites si es necesario
    labs(y = "Frecuencia",
         x = NULL,
         title = "2005: Conteo por sentimiento") +
    theme_minimal() -> p2
```

```
## Joining with `by = join_by(word)`
```

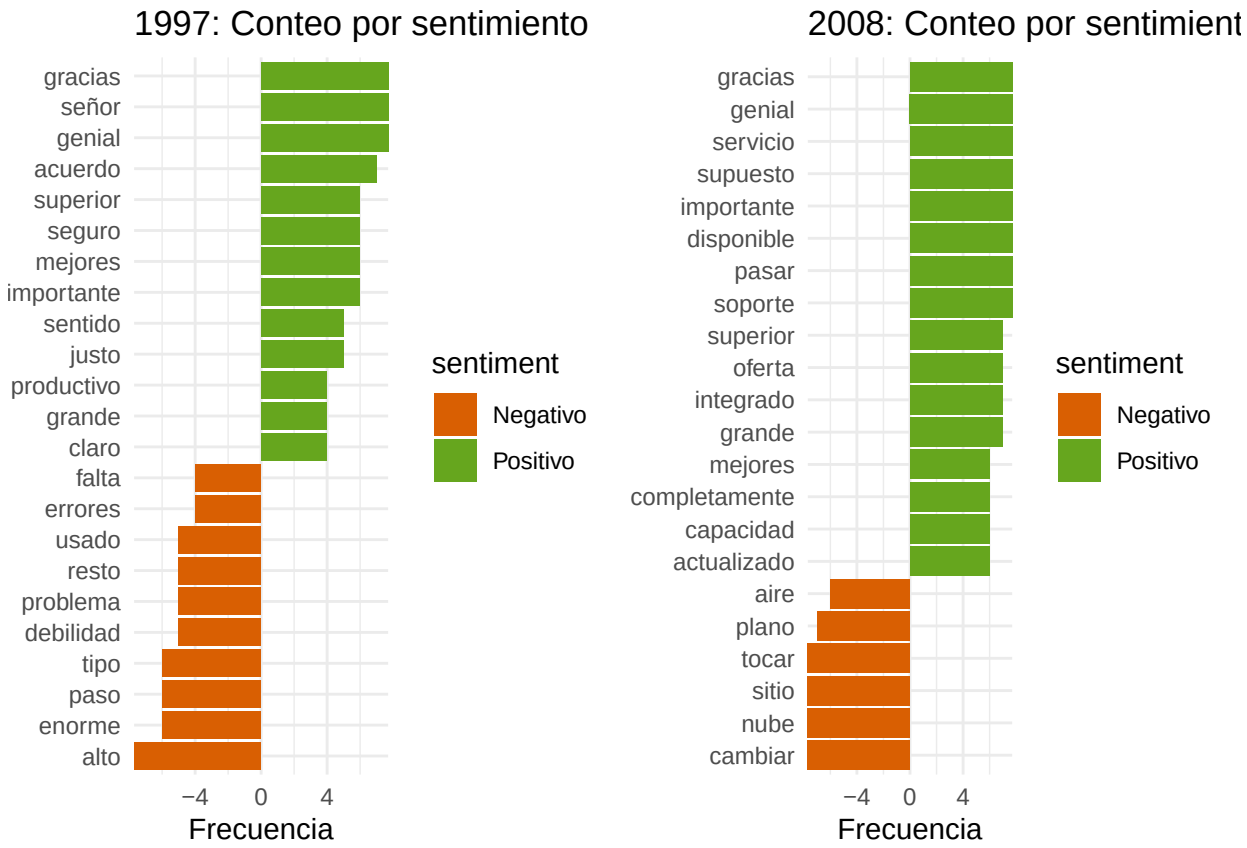
```
grid.arrange(p1, p2, ncol = 2)
```




```
text_2008 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  slice_max(order_by = abs(n), n = 20) %>%  # Mostrar las 50 palabras más frecuentes
  mutate(n = ifelse(sentiment == "Negativo", -n, n)) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
    geom_col() +
    scale_fill_manual(values = brewer.pal(8, 'Dark2')[c(2, 5)]) +
    coord_flip(ylim = c(-7, 7)) +  # Ajusta los límites si es necesario
    labs(y = "Frecuencia",
         x = NULL,
         title = "2008: Conteo por sentimiento") +
    theme_minimal() -> p2
```

```
## Joining with `by = join_by(word)`
```

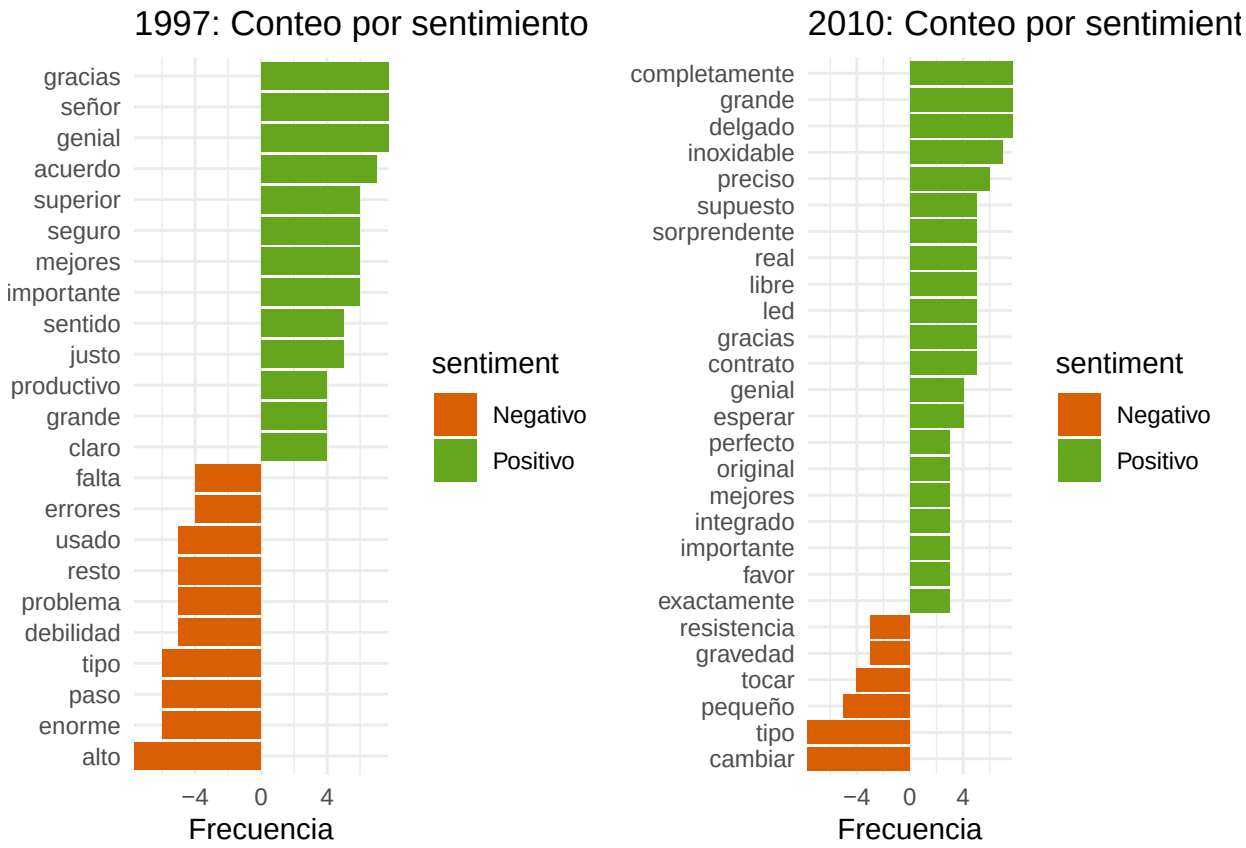
```
grid.arrange(p1, p2, ncol = 2)
```




```
text_2010 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  slice_max(order_by = abs(n), n = 20) %>%  # Mostrar las 50 palabras más frecuentes
  mutate(n = ifelse(sentiment == "Negativo", -n, n)) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
    geom_col() +
    scale_fill_manual(values = brewer.pal(8, 'Dark2')[c(2, 5)]) +
    coord_flip(ylim = c(-7, 7)) +  # Ajusta los límites si es necesario
    labs(y = "Frecuencia",
         x = NULL,
         title = "2010: Conteo por sentimiento") +
    theme_minimal() -> p2
```

```
## Joining with `by = join_by(word)`
```

```
grid.arrange(p1, p2, ncol = 2)
```




```
suppressMessages(suppressWarnings(library(reshape2)))  # acast
##### viz
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))

set.seed(123)
text_1997 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = brewer.pal(8,'Dark2')[c(2,5)],
                   max.words = 50, title.size = 1.5)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## debilidad could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problema could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## resto could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## usado could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## acuerdo could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
```

```
## errores could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## importante could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## mejores could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## superior could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## eliminar could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## imaginar could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## pequeño could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## grande could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## correcto could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## firmemente could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## fortaleza could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## hambre could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## horrible could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## intereses could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## licenciar could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## muerto could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## perdido could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## piedra could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## preocupado could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## probado could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problemas could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## triste could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## ventaja could not be fit on page. It will not be plotted.
```

```
title(main = "1997")

set.seed(123)
text_2001 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = brewer.pal(8,'Dark2')[c(2,5)],
                   max.words = 50, title.size = 1.5)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## manipular could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problema could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## quemar could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## interrupciones could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## pequeño could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## mejores could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## superior could not be fit on page. It will not be plotted.
```

```
title(main = "2001")
```

1997 **2001**


```
suppressMessages(suppressWarnings(library(reshape2)))  # acast
##### viz
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))

set.seed(123)
text_1997 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = brewer.pal(8,'Dark2')[c(2,5)],
                   max.words = 50, title.size = 1.5)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## debilidad could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problema could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## resto could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## usado could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## acuerdo could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## errores could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## importante could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
```

```
## mejores could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## superior could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## eliminar could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## imaginar could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## pequeño could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## grande could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## correcto could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## firmemente could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## fortaleza could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## hambre could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## horrible could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## intereses could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## licenciar could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## muerto could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## perdido could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## piedra could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## preocupado could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## probado could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problemas could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## triste could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## ventaja could not be fit on page. It will not be plotted.
title(main = "1997")
# ----------  ----------
```

```
set.seed(123)
text_2005 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = brewer.pal(8,'Dark2')[c(2,5)],
                   max.words = 50, title.size = 1.5)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## particularmente could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problema could not be fit on page. It will not be plotted.
```

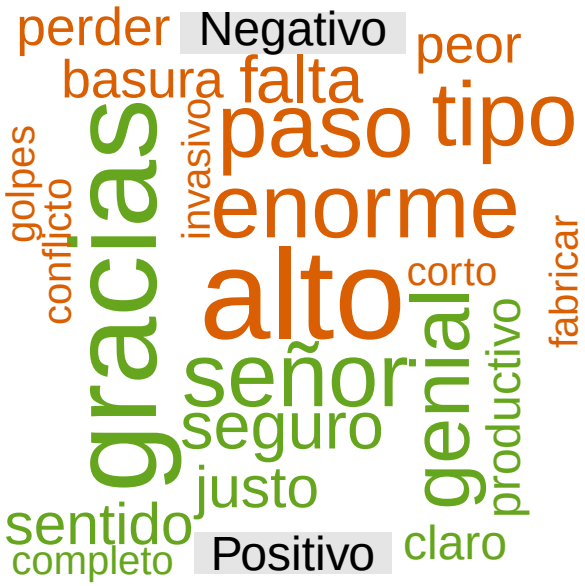
```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## propietario could not be fit on page. It will not be plotted.
```

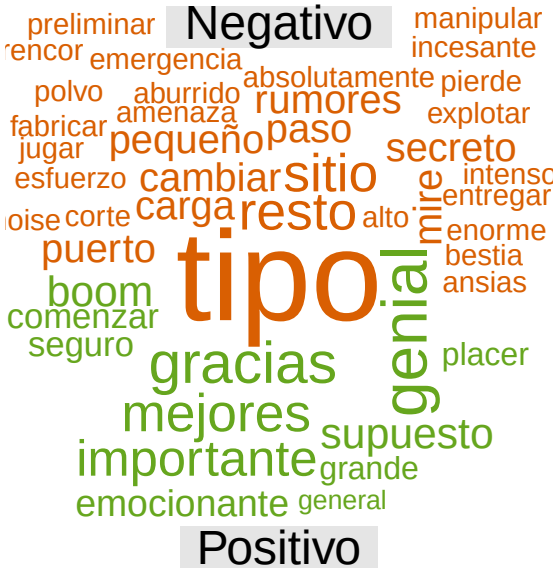
```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## separar could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## usado could not be fit on page. It will not be plotted.
```

```
title(main = "2005")
```

1997 **2005**






```
suppressMessages(suppressWarnings(library(reshape2)))  # acast
##### viz
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))
# ----------  ----------
set.seed(123)
text_1997 %>%
  inner_join(sentiment_words) %>%
```

```
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = brewer.pal(8,'Dark2')[c(2,5)],
                   max.words = 50, title.size = 1.5)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## debilidad could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problema could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## resto could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## usado could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## acuerdo could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## errores could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## importante could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## mejores could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## superior could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## eliminar could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## imaginar could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## pequeño could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## grande could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## correcto could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## firmemente could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## fortaleza could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## hambre could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## horrible could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## intereses could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## licenciar could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## muerto could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## perdido could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## piedra could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## preocupado could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## probado could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problemas could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## triste could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## ventaja could not be fit on page. It will not be plotted.
```

```
title(main = "1997")
# ----------  ----------
set.seed(123)
text_2008 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = brewer.pal(8,'Dark2')[c(2,5)],
                   max.words = 50, title.size = 1.5)
```

```
## Joining with `by = join_by(word)`

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], : nube
## could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## gracias could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## supuesto could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## absolutamente could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## disponible could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## importante could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## estudiante could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problemas could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## resto could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## actualizado could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## completamente could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## mejores could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## atrapado could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## mezclar could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## pequeño could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## persistente could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problema could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## recibir could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## usado could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## excelente could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## fenomenal could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## inmediatamente could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## interesante could not be fit on page. It will not be plotted.
title(main = "2008")
```

1997 **2008**


```
suppressMessages(suppressWarnings(library(reshape2)))  # acast
##### viz
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))
# ----------  ----------
set.seed(123)
text_1997 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = brewer.pal(8,'Dark2')[c(2,5)],
                   max.words = 50, title.size = 1.5)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## debilidad could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problema could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## resto could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## usado could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## acuerdo could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## errores could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## importante could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
```

```
## mejores could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## superior could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## eliminar could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## imaginar could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## pequeño could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## grande could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## correcto could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## firmemente could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## fortaleza could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## hambre could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## horrible could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## intereses could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## licenciar could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## muerto could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## perdido could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## piedra could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## preocupado could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## probado could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problemas could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## triste could not be fit on page. It will not be plotted.

## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## ventaja could not be fit on page. It will not be plotted.
title(main = "1997")
# ----------  ----------
```

```r
set.seed(123)
text_2010 %>%
  inner_join(sentiment_words) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = brewer.pal(8,'Dark2')[c(2,5)],
                   max.words = 50, title.size = 1.5)
```

```
## Joining with `by = join_by(word)`
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## gravedad could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## resistencia could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## problema could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## sorprendente could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## esperar could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## exactamente could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## importante could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## integrado could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## explotar could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## imaginar could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## inaudito could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## negro could not be fit on page. It will not be plotted.
```

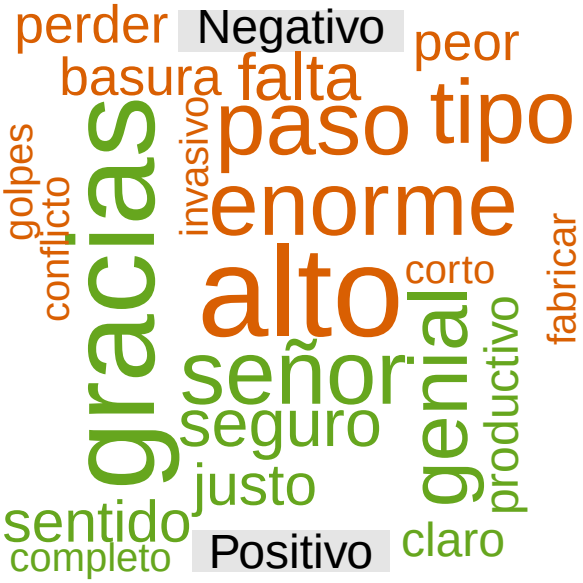
```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## opuesto could not be fit on page. It will not be plotted.
```

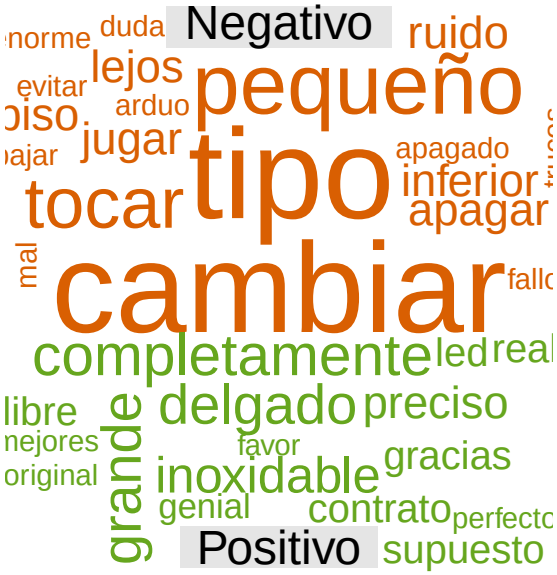
```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## papelera could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = brewer.pal(8, "Dark2")[c(2, 5)], :
## parecer could not be fit on page. It will not be plotted.
```

```r
title(main = "2010")
```

1997 **2010**






```
text_1997 <- unlist(c(read_csv("AppleWWDC1997_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

8

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2001 <- unlist(c(read_csv("AppleWWDC2001_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2005 <- unlist(c(read_csv("AppleWWDC2005_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2008 <- unlist(c(read_csv("AppleWWDC2008_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2010 <- unlist(c(read_csv("AppleWWDC2010_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
```

```
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
names(text_1997) <- NULL
text_1997 <- tibble(line = 1:length(text_1997), text = text_1997)
```

```
text_1997 %>%
  unnest_tokens(tbl = ., input = text, output = bigram, token = "ngrams", n = 2) %>%
  filter(!is.na(bigram)) -> text_1997_bi  # importante!
dim(text_1997_bi)
```

```
## [1] 10362     2
```

```
names(text_2001) <- NULL
text_2001 <- tibble(line = 1:length(text_2001), text = text_2001)
```

```
text_2001 %>%
  unnest_tokens(tbl = ., input = text, output = bigram, token = "ngrams", n = 2) %>%
  filter(!is.na(bigram)) -> text_2001_bi  # importante!
dim(text_2001_bi)
```

```
## [1] 13537     2
```

```
names(text_2005) <- NULL
text_2005 <- tibble(line = 1:length(text_2005), text = text_2005)
```

```
text_2005 %>%
  unnest_tokens(tbl = ., input = text, output = bigram, token = "ngrams", n = 2) %>%
  filter(!is.na(bigram)) -> text_2005_bi  # importante!
dim(text_2005_bi)
```

```
## [1] 7594    2
```

```
names(text_2008) <- NULL
text_2008 <- tibble(line = 1:length(text_2008), text = text_2008)
```

```
text_2008 %>%
  unnest_tokens(tbl = ., input = text, output = bigram, token = "ngrams", n = 2) %>%
  filter(!is.na(bigram)) -> text_2008_bi  # importante!
dim(text_2008_bi)
```

```
## [1] 13615     2
```

```
names(text_2010) <- NULL
text_2010 <- tibble(line = 1:length(text_2010), text = text_2010)
```

```
text_2010 %>%
  unnest_tokens(tbl = ., input = text, output = bigram, token = "ngrams", n = 2) %>%
  filter(!is.na(bigram)) -> text_2010_bi  # importante!
dim(text_2010_bi)
```

```
## [1] 5955    2
```

```
head(text_1997_bi, n = 10)
```

```
## # A tibble: 10 x 2
##     line bigram
##    <int> <chr>
##  1     1 buenos días
```

```
##  2     2 ambos llevaban
##  3     2 llevaban corbata
##  4     2 corbata toda
##  5     2 toda la
##  6     2 la semana
##  7     5 corto así
##  8     5 así que
##  9     5 que lo
## 10     5 lo haré
```

```
head(text_2005_bi, n = 10)
```

```
## # A tibble: 10 x 2
##     line bigram
##    <int> <chr>
##  1     1 bienvenidos a
##  2     1 a nuestra
##  3     1 nuestra conferencia
##  4     1 conferencia mundial
##  5     1 mundial de
##  6     1 de desarrolladores
##  7     1 desarrolladores 2005
##  8     1 2005 hoy
##  9     1 hoy es
## 10     1 es un
```

```
head(text_2008_bi, n = 10)
```

```
## # A tibble: 10 x 2
##     line bigram
##    <int> <chr>
##  1     1 estoy muy
##  2     1 muy contento
##  3     1 contento de
##  4     1 de estar
##  5     1 estar aquí
##  6     1 aquí esta
##  7     1 esta vez
##  8     2 buenos días
##  9     2 días hemos
## 10     2 hemos estado
```

```
head(text_2010_bi, n = 10)
```

```
## # A tibble: 10 x 2
##     line bigram
##    <int> <chr>
##  1     1 así que
##  2     1 que volvamos
##  3     1 volvamos al
##  4     1 al iphone
##  5     2 en 2007
##  6     2 2007 el
##  7     2 el iphone
##  8     2 iphone reinventó
##  9     2 reinventó lo
```

```
## 10     2 lo que
```

```
###### top 10 de bigramas mas frecuentes
# hay bigramas que no son interesantes (e.g., "de la")
# esto motiva el uso de stop words nuevamente
text_1997_bi %>%
  count(bigram, sort = TRUE) %>%
  head(n = 10)
```

```
## # A tibble: 10 x 2
##    bigram        n
##    <chr>     <int>
##  1 creo que     94
##  2 lo que       70
##  3 así que      35
##  4 que apple    34
##  5 y creo       33
##  6 en el        32
##  7 que no       32
##  8 de las       30
##  9 ya sabes     30
## 10 en la        28
```

```
###### top 10 de bigramas mas frecuentes
# hay bigramas que no son interesantes (e.g., "de la")
# esto motiva el uso de stop words nuevamente
text_2001_bi %>%
  count(bigram, sort = TRUE) %>%
  head(n = 10)
```

```
## # A tibble: 10 x 2
##    bigram       n
##    <chr>    <int>
##  1 lo que      85
##  2 así que     81
##  3 os 10       69
##  4 en el       66
##  5 de la       53
##  6 mac os      52
##  7 en la       39
##  8 es un       37
##  9 para que    31
## 10 es una      27
```

```
###### top 10 de bigramas mas frecuentes
# hay bigramas que no son interesantes (e.g., "de la")
# esto motiva el uso de stop words nuevamente
text_2005_bi %>%
  count(bigram, sort = TRUE) %>%
  head(n = 10)
```

```
## # A tibble: 10 x 2
##    bigram       n
##    <chr>    <int>
##  1 lo que      44
##  2 así que     41
##  3 en el       33
```

```
##  4 vamos a     23
##  5 ya sabes    21
##  6 más de      20
##  7 os 10       20
##  8 que es      19
##  9 de apple    18
## 10 de la       18
```

```
###### top 10 de bigramas mas frecuentes
# hay bigramas que no son interesantes (e.g., "de la")
# esto motiva el uso de stop words nuevamente
text_2008_bi %>%
  count(bigram, sort = TRUE) %>%
  head(n = 10)
```

```
## # A tibble: 10 x 2
##    bigram            n
##    <chr>         <int>
##  1 en el            89
##  2 así que          76
##  3 el iphone        71
##  4 lo que           69
##  5 en la            61
##  6 de la            43
##  7 para que         39
##  8 con el           29
##  9 voy a            28
## 10 la aplicación    26
```

```
###### top 10 de bigramas mas frecuentes
# hay bigramas que no son interesantes (e.g., "de la")
# esto motiva el uso de stop words nuevamente
text_2010_bi %>%
  count(bigram, sort = TRUE) %>%
  head(n = 10)
```

```
## # A tibble: 10 x 2
##    bigram         n
##    <chr>      <int>
##  1 lo que        40
##  2 iphone 4      38
##  3 el iphone     36
##  4 así que       33
##  5 en el         22
##  6 en la         21
##  7 de la         19
##  8 adelante y    17
##  9 para que      17
## 10 es el         15
```

```
text_1997_bi %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!grepl(pattern = '[0-9]', x = word1)) %>%
  filter(!grepl(pattern = '[0-9]', x = word2)) %>%
  filter(!word1 %in% stop_words_es$word) %>%
  filter(!word2 %in% stop_words_es$word) %>%
```

```
  mutate(word1 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word1)) %>%
  mutate(word2 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word2)) %>%
  filter(!is.na(word1)) %>%
  filter(!is.na(word2)) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n) -> text_1997_bi_counts  # importante para la conformacion de la red!
dim(text_1997_bi_counts)
```

```
## [1] 743   3
```

```
text_2001_bi %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!grepl(pattern = '[0-9]', x = word1)) %>%
  filter(!grepl(pattern = '[0-9]', x = word2)) %>%
  filter(!word1 %in% stop_words_es$word) %>%
  filter(!word2 %in% stop_words_es$word) %>%
  mutate(word1 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word1)) %>%
  mutate(word2 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word2)) %>%
  filter(!is.na(word1)) %>%
  filter(!is.na(word2)) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n) -> text_2001_bi_counts  # importante para la conformacion de la red!
dim(text_2001_bi_counts)
```

```
## [1] 1171    3
```

```
text_2005_bi %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!grepl(pattern = '[0-9]', x = word1)) %>%
  filter(!grepl(pattern = '[0-9]', x = word2)) %>%
  filter(!word1 %in% stop_words_es$word) %>%
  filter(!word2 %in% stop_words_es$word) %>%
  mutate(word1 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word1)) %>%
  mutate(word2 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word2)) %>%
  filter(!is.na(word1)) %>%
  filter(!is.na(word2)) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n) -> text_2005_bi_counts  # importante para la conformacion de la red!
dim(text_2005_bi_counts)
```

```
## [1] 588   3
```

```
text_2008_bi %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
```

```
  filter(!grepl(pattern = '[0-9]', x = word1)) %>%
  filter(!grepl(pattern = '[0-9]', x = word2)) %>%
  filter(!word1 %in% stop_words_es$word) %>%
  filter(!word2 %in% stop_words_es$word) %>%
  mutate(word1 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word1)) %>%
  mutate(word2 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word2)) %>%
  filter(!is.na(word1)) %>%
  filter(!is.na(word2)) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n) -> text_2008_bi_counts  # importante para la conformacion de la red!
dim(text_2008_bi_counts)
```

```
## [1] 1188    3
```

```
text_2010_bi %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!grepl(pattern = '[0-9]', x = word1)) %>%
  filter(!grepl(pattern = '[0-9]', x = word2)) %>%
  filter(!word1 %in% stop_words_es$word) %>%
  filter(!word2 %in% stop_words_es$word) %>%
  mutate(word1 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word1)) %>%
  mutate(word2 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word2)) %>%
  filter(!is.na(word1)) %>%
  filter(!is.na(word2)) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n) -> text_2010_bi_counts  # importante para la conformacion de la red!
dim(text_2010_bi_counts)
```

```
## [1] 508   3
```

```
head(text_1997_bi_counts, n = 10)
```

```
## # A tibble: 10 x 3
##    word1     word2       weight
##    <chr>     <chr>        <int>
##  1 rap       city             8
##  2 correo    electronico      7
##  3 street    journal          6
##  4 wall      street           6
##  5 productos realmente        5
##  6 apple     deberia          4
##  7 disco     duro             4
##  8 pequeña   cosa             4
##  9 and       play             3
## 10 apple     necesita         3
```

```
head(text_2001_bi_counts, n = 10)
```

```
## # A tibble: 10 x 3
##    word1   word2     weight
##    <chr>   <chr>      <int>
##  1 mac     os            52
##  2 sistema operativo      9
##  3 super   drive           9
##  4 power   mac             8
##  5 gracias steve           7
##  6 disco   duro            6
##  7 power   max             6
##  8 centro  comercial       5
##  9 grabar  dvd             5
## 10 libro   mundial         5
```

```
head(text_2005_bi_counts, n = 10)
```

```
## # A tibble: 10 x 3
##    word1        word2       weight
##    <chr>        <chr>        <int>
##  1 sistema      operativo       12
##  2 mac          os              10
##  3 binarios     universales      7
##  4 has          visto            7
##  5 codigo       fuente           6
##  6 procesadores intel            6
##  7 dejame       abrir            5
##  8 excelentes   productos        5
##  9 dejame       mostrarte        4
## 10 wolfram      research         4
```

```
head(text_2008_bi_counts, n = 10)
```

```
## # A tibble: 10 x 3
##    word1       word2        weight
##    <chr>       <chr>         <int>
##  1 correo      electronico      25
##  2 gustaria    invitar          12
##  3 app         store            11
##  4 mis         contactos         8
##  5 sitio       web               6
##  6 correos     electronicos      5
##  7 dispositivo movil             5
##  8 interfaz    web               5
##  9 mac         os                5
## 10 tu          iphone            5
```

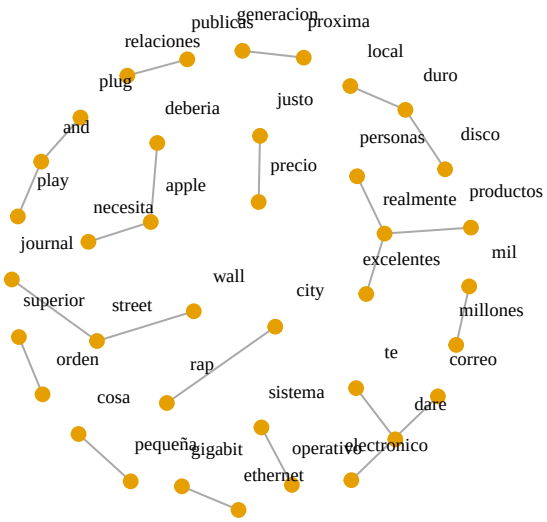
```
head(text_2010_bi_counts, n = 10)
```

```
## # A tibble: 10 x 3
##    word1        word2      weight
##    <chr>        <chr>       <int>
##  1 pantalla     retina         14
##  2 acero        inoxidable      7
##  3 wi           fi              7
##  4 estaciones   base            5
##  5 tu           telefono        5
```

```
##  6 alta         definicion      4
##  7 flash        led             4
##  8 computadoras portatiles      3
##  9 has          visto           3
## 10 iphone       original        3
```

```
##### definir una red a partir de la frecuencia (weight) de los bigramas
# binaria, no dirigida, ponderada, simple
# se recomienda variar el umbral del filtro y construir bigramas no consecutivos para obtener redes con
suppressMessages(suppressWarnings(library(igraph)))

# Crear la red
g <- text_1997_bi_counts %>%
  filter(weight > 2) %>%
  graph_from_data_frame(directed = FALSE)

# Ajuste visual
set.seed(123)
plot(g,
     layout = layout_with_kk,  # Layout con más dispersión
     vertex.color = 1,
     vertex.frame.color = 1,
     vertex.size = 6,  # Tamaño de los vértices mayor para mayor separación
     vertex.label.color = 'black',
     vertex.label.cex = 0.6,  # Reducir un poco el tamaño de la fuente
     vertex.label.dist = 3,  # Aumenté la distancia entre las etiquetas y los vértices
     main = "1997 - Umbral = 2")
```

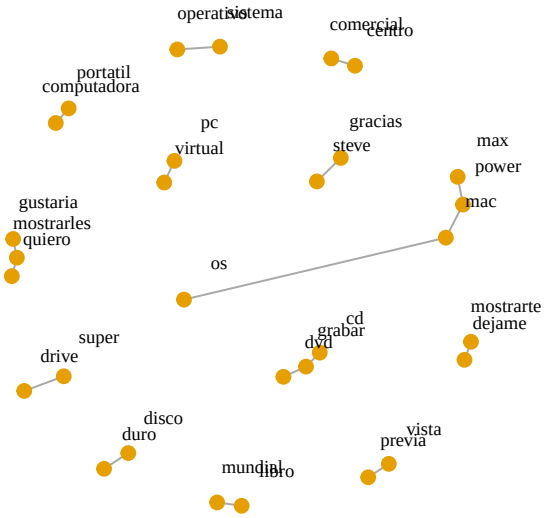
1997 − Umbral = 2




```
##### definir una red a partir de la frecuencia (weight) de los bigramas
# binaria, no dirigida, ponderada, simple
# se recomienda variar el umbral del filtro y construir bigramas no consecutivos para obtener redes con
suppressMessages(suppressWarnings(library(igraph)))

# Crear la red
g <- text_2001_bi_counts %>%
  filter(weight > 3) %>%
```

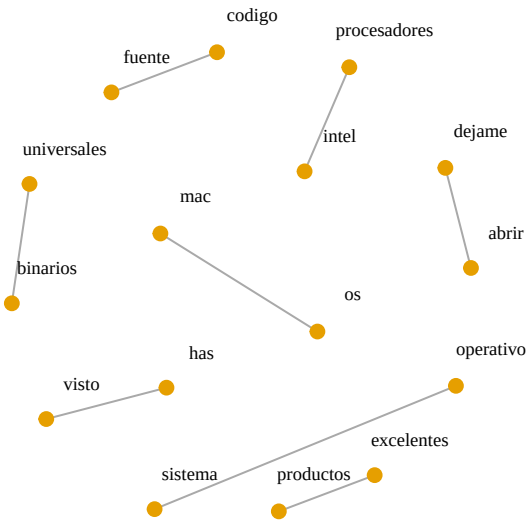
```
  graph_from_data_frame(directed = FALSE)

# Ajuste visual
set.seed(123)
plot(g,
     layout = layout_with_kk,  # Layout con más dispersión
     vertex.color = 1,
     vertex.frame.color = 1,
     vertex.size = 6,  # Tamaño de los vértices mayor para mayor separación
     vertex.label.color = 'black',
     vertex.label.cex = 0.6,  # Reducir un poco el tamaño de la fuente
     vertex.label.dist = 3,  # Aumenté la distancia entre las etiquetas y los vértices
     main = "2001 - Umbral = 3")
```

2001 – Umbral = 3



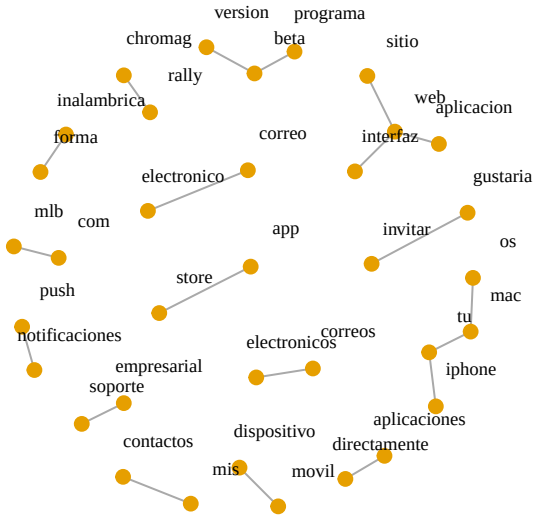

```
##### definir una red a partir de la frecuencia (weight) de los bigramas
# binaria, no dirigida, ponderada, simple
# se recomienda variar el umbral del filtro y construir bigramas no consecutivos para obtener redes con
suppressMessages(suppressWarnings(library(igraph)))

# Crear la red
g <- text_2005_bi_counts %>%
  filter(weight > 4) %>%
  graph_from_data_frame(directed = FALSE)

# Ajuste visual
set.seed(123)
plot(g,
     layout = layout_with_kk,  # Layout con más dispersión
     vertex.color = 1,
     vertex.frame.color = 1,
     vertex.size = 6,  # Tamaño de los vértices mayor para mayor separación
     vertex.label.color = 'black',
     vertex.label.cex = 0.6,  # Reducir un poco el tamaño de la fuente
     vertex.label.dist = 3,  # Aumenté la distancia entre las etiquetas y los vértices
     main = "2005 - Umbral = 4")
```

2005 − Umbral = 4



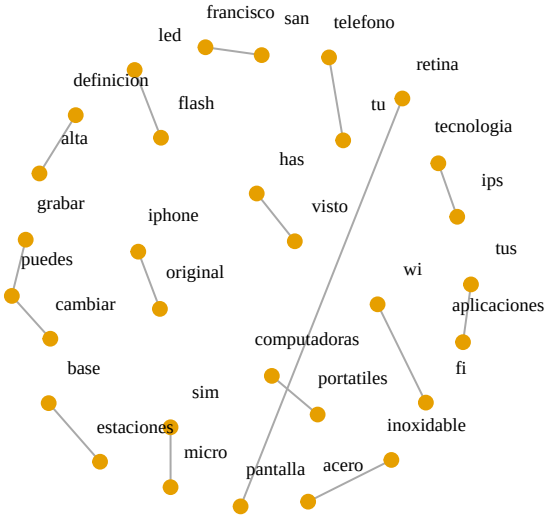

```
##### definir una red a partir de la frecuencia (weight) de los bigramas
# binaria, no dirigida, ponderada, simple
# se recomienda variar el umbral del filtro y construir bigramas no consecutivos para obtener redes con
suppressMessages(suppressWarnings(library(igraph)))

# Crear la red
g <- text_2008_bi_counts %>%
  filter(weight > 3) %>%
  graph_from_data_frame(directed = FALSE)

# Ajuste visual
set.seed(123)
plot(g,
     layout = layout_with_kk,  # Layout con más dispersión
     vertex.color = 1,
     vertex.frame.color = 1,
     vertex.size = 6,  # Tamaño de los vértices mayor para mayor separación
     vertex.label.color = 'black',
     vertex.label.cex = 0.6,  # Reducir un poco el tamaño de la fuente
     vertex.label.dist = 3,  # Aumenté la distancia entre las etiquetas y los vértices
     main = "2008 - Umbral = 3")
```

2008 − Umbral = 3



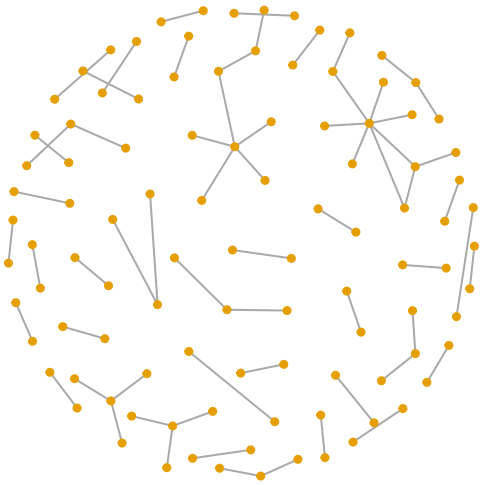

```
##### definir una red a partir de la frecuencia (weight) de los bigramas
# binaria, no dirigida, ponderada, simple
# se recomienda variar el umbral del filtro y construir bigramas no consecutivos para obtener redes con
suppressMessages(suppressWarnings(library(igraph)))

# Crear la red
g <- text_2010_bi_counts %>%
  filter(weight > 2) %>%
  graph_from_data_frame(directed = FALSE)

# Ajuste visual
set.seed(123)
plot(g,
     layout = layout_with_kk,  # Layout con más dispersión
     vertex.color = 1,
     vertex.frame.color = 1,
     vertex.size = 6,  # Tamaño de los vértices mayor para mayor separación
     vertex.label.color = 'black',
     vertex.label.cex = 0.6,  # Reducir un poco el tamaño de la fuente
     vertex.label.dist = 3,  # Aumenté la distancia entre las etiquetas y los vértices
     main = "2010 - Umbral = 2")
```

2010 − Umbral = 2



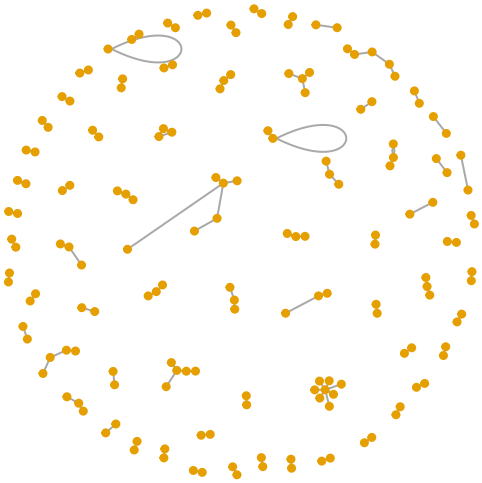

```
##### red con un umbral diferente
g <- text_1997_bi_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
# viz
set.seed(123)
plot(g, layout = layout_with_kk, vertex.color = 1, vertex.frame.color = 1, vertex.size = 3, vertex.label
```

1997 − Umbral = 1



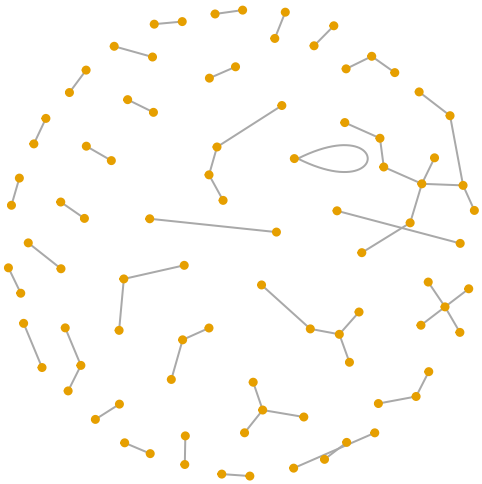
```
##### red con un umbral diferente
g <- text_2001_bi_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
# viz
set.seed(123)
plot(g, layout = layout_with_kk, vertex.color = 1, vertex.frame.color = 1, vertex.size = 3, vertex.label
```

2001 − Umbral = 1



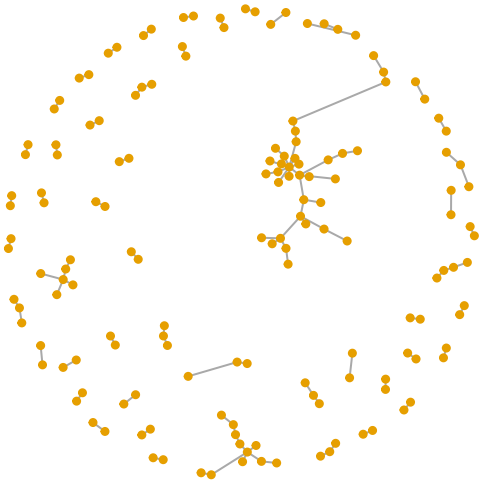
```
##### red con un umbral diferente
g <- text_2005_bi_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
# viz
set.seed(123)
plot(g, layout = layout_with_kk, vertex.color = 1, vertex.frame.color = 1, vertex.size = 3, vertex.label
```

2005 − Umbral = 1



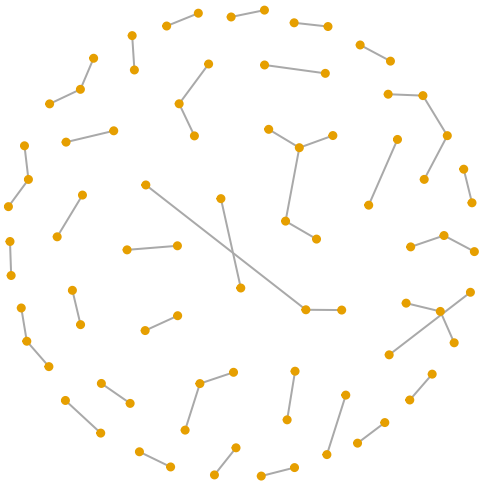
```
##### red con un umbral diferente
g <- text_2008_bi_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
# viz
set.seed(123)
plot(g, layout = layout_with_kk, vertex.color = 1, vertex.frame.color = 1, vertex.size = 3, vertex.label
```

2008 − Umbral = 1



```
##### red con un umbral diferente
g <- text_2010_bi_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
# viz
set.seed(123)
plot(g, layout = layout_with_kk, vertex.color = 1, vertex.frame.color = 1, vertex.size = 3, vertex.label
```

2010 − Umbral = 1

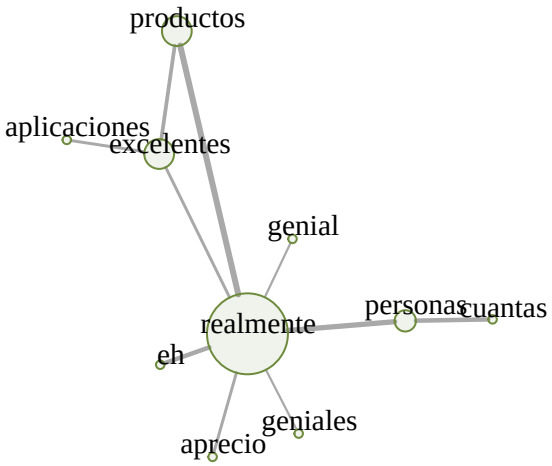


```
##### componente conexa mas grande de la red
g <- text_1997_bi_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
# grafo inducido por la componente conexa
V(g)$cluster <- clusters(graph = g)$membership
```

```
## Warning: `clusters()` was deprecated in igraph 2.0.0.
```

```
## i Please use `components()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

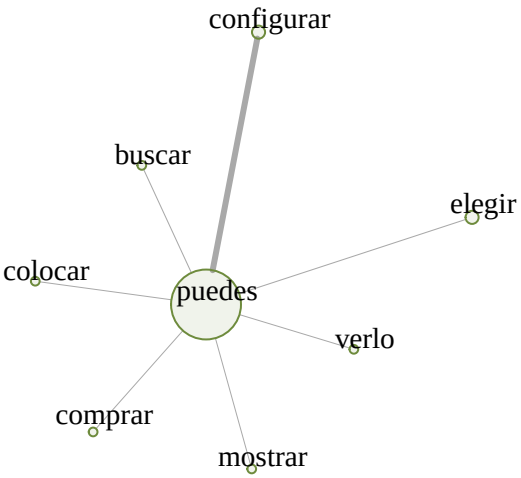
```
gcc <- induced_subgraph(graph = g, vids = which(V(g)$cluster == which.max(clusters(graph = g)$csize)))
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))

set.seed(123)
plot(gcc, layout = layout_with_kk, vertex.color = adjustcolor('darkolivegreen4', 0.1), vertex.frame.colo
title(main = "Componente conexa", outer = T, line = -1)
```




```
##### componente conexa mas grande de la red
g <- text_2001_bi_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
# grafo inducido por la componente conexa
V(g)$cluster <- clusters(graph = g)$membership
```

```
gcc <- induced_subgraph(graph = g, vids = which(V(g)$cluster == which.max(clusters(graph = g)$csize)))
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))

# viz 2
set.seed(123)
plot(gcc, layout = layout_with_kk, vertex.color = adjustcolor('darkolivegreen4', 0.1), vertex.frame.colo
title(main = "Componente conexa", outer = T, line = -1)
```

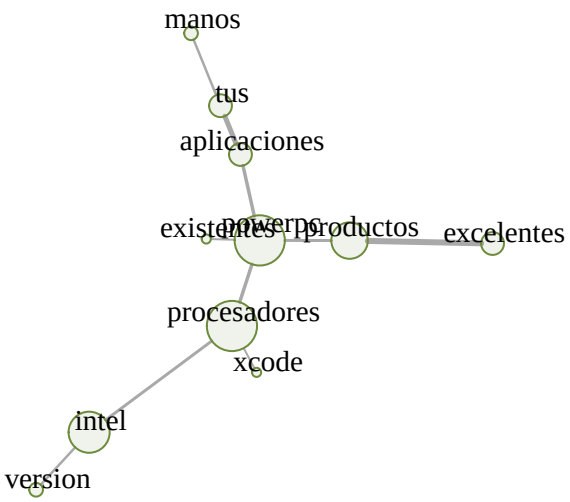
Componente conexa




```
##### componente conexa mas grande de la red
g <- text_2005_bi_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
# grafo inducido por la componente conexa
V(g)$cluster <- clusters(graph = g)$membership
```

```
gcc <- induced_subgraph(graph = g, vids = which(V(g)$cluster == which.max(clusters(graph = g)$csize)))
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))

# viz 2
set.seed(123)
plot(gcc, layout = layout_with_kk, vertex.color = adjustcolor('darkolivegreen4', 0.1), vertex.frame.col
title(main = "Componente conexa", outer = T, line = -1)
```

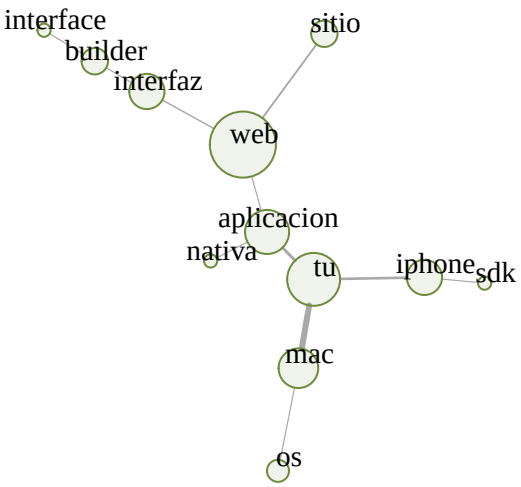
Componente conexa




```
##### componente conexa mas grande de la red
g <- text_2008_bi_counts %>%
  filter(weight > 2) %>%
  graph_from_data_frame(directed = FALSE)
# grafo inducido por la componente conexa
V(g)$cluster <- clusters(graph = g)$membership
```

```
gcc <- induced_subgraph(graph = g, vids = which(V(g)$cluster == which.max(clusters(graph = g)$csize)))
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))

# viz 2
set.seed(123)
plot(gcc, layout = layout_with_kk, vertex.color = adjustcolor('darkolivegreen4', 0.1), vertex.frame.col
title(main = "Componente conexa", outer = T, line = -1)
```

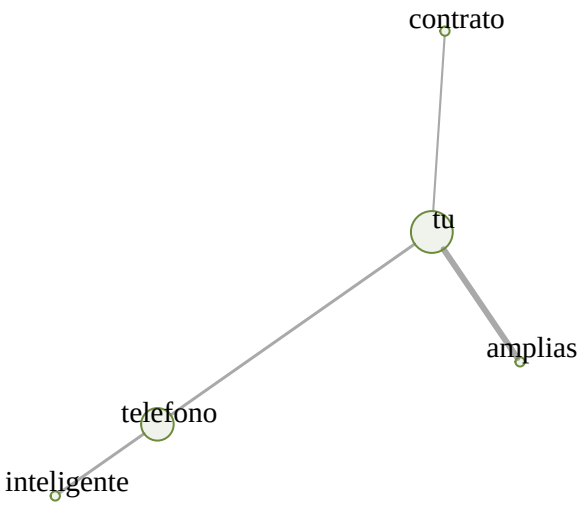
Componente conexa




```
##### componente conexa mas grande de la red
g <- text_2010_bi_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
# grafo inducido por la componente conexa
V(g)$cluster <- clusters(graph = g)$membership
```

```
gcc <- induced_subgraph(graph = g, vids = which(V(g)$cluster == which.max(clusters(graph = g)$csize)))
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))

# viz 2
set.seed(123)
plot(gcc, layout = layout_with_kk, vertex.color = adjustcolor('darkolivegreen4', 0.1), vertex.frame.col
title(main = "Componente conexa", outer = T, line = -1)
```

Componente conexa




```
##### importar datos
text_1997 <- unlist(c(read_csv("AppleWWDC1997_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

9

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2001 <- unlist(c(read_csv("AppleWWDC2001_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2005 <- unlist(c(read_csv("AppleWWDC2005_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2008 <- unlist(c(read_csv("AppleWWDC2008_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
text_2010 <- unlist(c(read_csv("AppleWWDC2010_es.txt", col_names = FALSE, show_col_types = FALSE)))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
```

```
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
names(text_1997) <- NULL
text_1997 <- tibble(line = 1:length(text_1997), text = text_1997)
```

```
names(text_2001) <- NULL
text_2001 <- tibble(line = 1:length(text_2001), text = text_2001)
```

```
names(text_2005) <- NULL
text_2005 <- tibble(line = 1:length(text_2005), text = text_2005)
```

```
names(text_2008) <- NULL
text_2008 <- tibble(line = 1:length(text_2008), text = text_2008)
```

```
names(text_2010) <- NULL
text_2010 <- tibble(line = 1:length(text_2010), text = text_2010)
```

```
##### tokenizar en skip-gram
# en este caso cada token es un unigrama o un bigrama regular o un bigrama con espaciamiento
text_1997 %>%
  unnest_tokens(tbl = ., input = text, output = skipgram, token = "skip_ngrams", n = 2) %>%
  filter(!is.na(skipgram)) -> text_1997_skip
dim(text_1997_skip)
```

```
## [1] 31144     2
```

```
text_2001 %>%
  unnest_tokens(tbl = ., input = text, output = skipgram, token = "skip_ngrams", n = 2) %>%
  filter(!is.na(skipgram)) -> text_2001_skip
dim(text_2001_skip)
```

```
## [1] 40728     2
```

```
text_2005 %>%
  unnest_tokens(tbl = ., input = text, output = skipgram, token = "skip_ngrams", n = 2) %>%
  filter(!is.na(skipgram)) -> text_2005_skip
dim(text_2005_skip)
```

```
## [1] 22792     2
```

```
text_2008 %>%
  unnest_tokens(tbl = ., input = text, output = skipgram, token = "skip_ngrams", n = 2) %>%
  filter(!is.na(skipgram)) -> text_2008_skip
dim(text_2008_skip)
```

```
## [1] 40861     2
```

```
text_2010 %>%
  unnest_tokens(tbl = ., input = text, output = skipgram, token = "skip_ngrams", n = 2) %>%
  filter(!is.na(skipgram)) -> text_2010_skip
dim(text_2010_skip)
```

```
## [1] 17871     2
```

```
head(text_1997_skip, n = 10)
```

```
## # A tibble: 10 x 2
##     line skipgram
```

```
##    <int> <chr>
##  1     1 buenos
##  2     1 buenos días
##  3     1 días
##  4     2 ambos
##  5     2 ambos llevaban
##  6     2 ambos corbata
##  7     2 llevaban
##  8     2 llevaban corbata
##  9     2 llevaban toda
## 10     2 corbata
```

```
head(text_2001_skip, n = 10)
```

```
## # A tibble: 10 x 2
##     line skipgram
##    <int> <chr>
##  1     1 buenos
##  2     1 buenos días
##  3     1 días
##  4     2 estamos
##  5     2 estamos muy
##  6     2 estamos contentos
##  7     2 muy
##  8     2 muy contentos
##  9     2 muy de
## 10     2 contentos
```

```
head(text_2005_skip, n = 10)
```

```
## # A tibble: 10 x 2
##     line skipgram
##    <int> <chr>
##  1     1 bienvenidos
##  2     1 bienvenidos a
##  3     1 bienvenidos nuestra
##  4     1 a
##  5     1 a nuestra
##  6     1 a conferencia
##  7     1 nuestra
##  8     1 nuestra conferencia
##  9     1 nuestra mundial
## 10     1 conferencia
```

```
head(text_2008_skip, n = 10)
```

```
## # A tibble: 10 x 2
##     line skipgram
##    <int> <chr>
##  1     1 estoy
##  2     1 estoy muy
##  3     1 estoy contento
##  4     1 muy
##  5     1 muy contento
##  6     1 muy de
##  7     1 contento
```

```
##  8     1 contento de
##  9     1 contento estar
## 10     1 de
```

```
head(text_2010_skip, n = 10)
```

```
## # A tibble: 10 x 2
##     line skipgram
##    <int> <chr>
##  1     1 así
##  2     1 así que
##  3     1 así volvamos
##  4     1 que
##  5     1 que volvamos
##  6     1 que al
##  7     1 volvamos
##  8     1 volvamos al
##  9     1 volvamos iphone
## 10     1 al
```

```
suppressMessages(suppressWarnings(library(ngram)))
# contar palabras en cada skip-gram
text_1997_skip$num_words <- text_1997_skip$skipgram %>%
  map_int(.f = ~ wordcount(.x))
head(text_1997_skip, n = 10)
```

```
## # A tibble: 10 x 3
##     line skipgram         num_words
##    <int> <chr>                <int>
##  1     1 buenos                   1
##  2     1 buenos días              2
##  3     1 días                     1
##  4     2 ambos                    1
##  5     2 ambos llevaban           2
##  6     2 ambos corbata            2
##  7     2 llevaban                 1
##  8     2 llevaban corbata         2
##  9     2 llevaban toda            2
## 10     2 corbata                  1
```

```
# remover unigramas
text_1997_skip %<>%
  filter(num_words == 2) %>%
  select(-num_words)
dim(text_1997_skip)
```

```
## [1] 19788     2
```

```
suppressMessages(suppressWarnings(library(ngram)))
# contar palabras en cada skip-gram
text_2001_skip$num_words1 <- text_2001_skip$skipgram %>%
  map_int(.f = ~ wordcount(.x))
head(text_2001_skip, n = 10)
```

```
## # A tibble: 10 x 3
##     line skipgram          num_words1
##    <int> <chr>                  <int>
```

```
##  1     1 buenos                    1
##  2     1 buenos días               2
##  3     1 días                      1
##  4     2 estamos                   1
##  5     2 estamos muy               2
##  6     2 estamos contentos         2
##  7     2 muy                       1
##  8     2 muy contentos             2
##  9     2 muy de                    2
## 10     2 contentos                 1
```

```
text_2001_skip %<>%
  filter(num_words1 == 2) %>%
  select(-num_words1)
dim(text_2001_skip)
```

```
## [1] 25629     2
```

```
suppressMessages(suppressWarnings(library(ngram)))
# contar palabras en cada skip-gram
text_2005_skip$num_words2 <- text_2005_skip$skipgram %>%
  map_int(.f = ~ wordcount(.x))
head(text_2005_skip, n = 10)
```

```
## # A tibble: 10 x 3
##     line skipgram            num_words2
##    <int> <chr>                    <int>
##  1     1 bienvenidos                  1
##  2     1 bienvenidos a                2
##  3     1 bienvenidos nuestra          2
##  4     1 a                            1
##  5     1 a nuestra                    2
##  6     1 a conferencia                2
##  7     1 nuestra                      1
##  8     1 nuestra conferencia          2
##  9     1 nuestra mundial              2
## 10     1 conferencia                  1
```

```
text_2005_skip %<>%
  filter(num_words2 == 2) %>%
  select(-num_words2)
dim(text_2005_skip)
```

```
## [1] 14690     2
```

```
suppressMessages(suppressWarnings(library(ngram)))
# contar palabras en cada skip-gram
text_2008_skip$num_words3 <- text_2008_skip$skipgram %>%
  map_int(.f = ~ wordcount(.x))
head(text_2008_skip, n = 10)
```

```
## # A tibble: 10 x 3
##     line skipgram       num_words3
##    <int> <chr>               <int>
##  1     1 estoy                   1
##  2     1 estoy muy               2
##  3     1 estoy contento          2
```

```
##  4     1 muy                    1
##  5     1 muy contento           2
##  6     1 muy de                 2
##  7     1 contento               1
##  8     1 contento de            2
##  9     1 contento estar         2
## 10     1 de                     1
```

```
text_2008_skip %<>%
  filter(num_words3 == 2) %>%
  select(-num_words3)
dim(text_2008_skip)
```

```
## [1] 26346     2
```

```
suppressMessages(suppressWarnings(library(ngram)))
# contar palabras en cada skip-gram
text_2010_skip$num_words4 <- text_2010_skip$skipgram %>%
  map_int(.f = ~ wordcount(.x))
head(text_2010_skip, n = 10)
```

```
## # A tibble: 10 x 3
##     line skipgram        num_words4
##    <int> <chr>                <int>
##  1     1 así                      1
##  2     1 así que                  2
##  3     1 así volvamos             2
##  4     1 que                      1
##  5     1 que volvamos             2
##  6     1 que al                   2
##  7     1 volvamos                 1
##  8     1 volvamos al              2
##  9     1 volvamos iphone          2
## 10     1 al                       1
```

```
text_2010_skip %<>%
  filter(num_words4 == 2) %>%
  select(-num_words4)
dim(text_2010_skip)
```

```
## [1] 11459     2
```

```
head(text_1997_skip, n = 10)
```

```
## # A tibble: 10 x 2
##     line skipgram
##    <int> <chr>
##  1     1 buenos días
##  2     2 ambos llevaban
##  3     2 ambos corbata
##  4     2 llevaban corbata
##  5     2 llevaban toda
##  6     2 corbata toda
##  7     2 corbata la
##  8     2 toda la
##  9     2 toda semana
## 10     2 la semana
```

```r
head(text_2001_skip, n = 10)
```

```
## # A tibble: 10 x 2
##     line skipgram
##    <int> <chr>
##  1     1 buenos días
##  2     2 estamos muy
##  3     2 estamos contentos
##  4     2 muy contentos
##  5     2 muy de
##  6     2 contentos de
##  7     2 contentos estar
##  8     2 de estar
##  9     2 de aquí
## 10     2 estar aquí
```

```r
head(text_2005_skip, n = 10)
```

```
## # A tibble: 10 x 2
##     line skipgram
##    <int> <chr>
##  1     1 bienvenidos a
##  2     1 bienvenidos nuestra
##  3     1 a nuestra
##  4     1 a conferencia
##  5     1 nuestra conferencia
##  6     1 nuestra mundial
##  7     1 conferencia mundial
##  8     1 conferencia de
##  9     1 mundial de
## 10     1 mundial desarrolladores
```

```r
head(text_2008_skip, n = 10)
```

```
## # A tibble: 10 x 2
##     line skipgram
##    <int> <chr>
##  1     1 estoy muy
##  2     1 estoy contento
##  3     1 muy contento
##  4     1 muy de
##  5     1 contento de
##  6     1 contento estar
##  7     1 de estar
##  8     1 de aquí
##  9     1 estar aquí
## 10     1 estar esta
```

```r
head(text_2010_skip, n = 10)
```

```
## # A tibble: 10 x 2
##     line skipgram
##    <int> <chr>
##  1     1 así que
##  2     1 así volvamos
##  3     1 que volvamos
```

```
##  4     1 que al
##  5     1 volvamos al
##  6     1 volvamos iphone
##  7     1 al iphone
##  8     2 en 2007
##  9     2 en el
## 10     2 2007 el
```

```
##### omitir stop words
text_1997_skip %>%
  separate(skipgram, c("word1", "word2"), sep = " ") %>%
  filter(!grepl(pattern = '[0-9]', x = word1)) %>%
  filter(!grepl(pattern = '[0-9]', x = word2)) %>%
  filter(!word1 %in% stop_words_es$word) %>%
  filter(!word2 %in% stop_words_es$word) %>%
  mutate(word1 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word1)) %>%
  mutate(word2 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word2)) %>%
  filter(!is.na(word1)) %>%
  filter(!is.na(word2)) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n) -> text_1997_skip_counts
dim(text_1997_skip_counts)
```

```
## [1] 1852    3
```

```
text_2001_skip %>%
  separate(skipgram, c("word1", "word2"), sep = " ") %>%
  filter(!grepl(pattern = '[0-9]', x = word1)) %>%
  filter(!grepl(pattern = '[0-9]', x = word2)) %>%
  filter(!word1 %in% stop_words_es$word) %>%
  filter(!word2 %in% stop_words_es$word) %>%
  mutate(word1 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word1)) %>%
  mutate(word2 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                        new = replacement_list %>% str_c(collapse = ''),
                        x = word2)) %>%
  filter(!is.na(word1)) %>%
  filter(!is.na(word2)) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n) -> text_2001_skip_counts
dim(text_2001_skip_counts)
```

```
## [1] 2901    3
```

```
text_2005_skip %>%
  separate(skipgram, c("word1", "word2"), sep = " ") %>%
  filter(!grepl(pattern = '[0-9]', x = word1)) %>%
  filter(!grepl(pattern = '[0-9]', x = word2)) %>%
  filter(!word1 %in% stop_words_es$word) %>%
  filter(!word2 %in% stop_words_es$word) %>%
  mutate(word1 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
```

```
                      new = replacement_list %>% str_c(collapse = ''),
                      x = word1)) %>%
  mutate(word2 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                      new = replacement_list %>% str_c(collapse = ''),
                      x = word2)) %>%
  filter(!is.na(word1)) %>%
  filter(!is.na(word2)) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n) -> text_2005_skip_counts
dim(text_2005_skip_counts)
```

```
## [1] 1580    3
```

```
text_2008_skip %>%
  separate(skipgram, c("word1", "word2"), sep = " ") %>%
  filter(!grepl(pattern = '[0-9]', x = word1)) %>%
  filter(!grepl(pattern = '[0-9]', x = word2)) %>%
  filter(!word1 %in% stop_words_es$word) %>%
  filter(!word2 %in% stop_words_es$word) %>%
  mutate(word1 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                      new = replacement_list %>% str_c(collapse = ''),
                      x = word1)) %>%
  mutate(word2 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                      new = replacement_list %>% str_c(collapse = ''),
                      x = word2)) %>%
  filter(!is.na(word1)) %>%
  filter(!is.na(word2)) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n) -> text_2008_skip_counts
dim(text_2008_skip_counts)
```

```
## [1] 3065    3
```

```
text_2010_skip %>%
  separate(skipgram, c("word1", "word2"), sep = " ") %>%
  filter(!grepl(pattern = '[0-9]', x = word1)) %>%
  filter(!grepl(pattern = '[0-9]', x = word2)) %>%
  filter(!word1 %in% stop_words_es$word) %>%
  filter(!word2 %in% stop_words_es$word) %>%
  mutate(word1 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                      new = replacement_list %>% str_c(collapse = ''),
                      x = word1)) %>%
  mutate(word2 = chartr(old = names(replacement_list) %>% str_c(collapse = ''),
                      new = replacement_list %>% str_c(collapse = ''),
                      x = word2)) %>%
  filter(!is.na(word1)) %>%
  filter(!is.na(word2)) %>%
  count(word1, word2, sort = TRUE) %>%
  rename(weight = n) -> text_2010_skip_counts
dim(text_2010_skip_counts)
```

```
## [1] 1219    3
```

```
head(text_1997_skip_counts, n = 10)
```

```
## # A tibble: 10 x 3
```

```
##    word1     word2       weight
##    <chr>     <chr>        <int>
##  1 rap       city             8
##  2 correo    electronico      7
##  3 punto     vista            7
##  4 street    journal          6
##  5 wall      journal          6
##  6 wall      street           6
##  7 apple     deberia          5
##  8 creadores clones           5
##  9 hardware  apple            5
## 10 productos realmente        5
```

```
head(text_2001_skip_counts, n = 10)
```

```
## # A tibble: 10 x 3
##    word1      word2     weight
##    <chr>      <chr>      <int>
##  1 mac        os            52
##  2 sistema    operativo      9
##  3 super      drive          9
##  4 power      mac            8
##  5 gracias    steve          7
##  6 disco      duro           6
##  7 disponible mac            6
##  8 etapas     pipeline       6
##  9 power      max            6
## 10 centro     comercial      5
```

```
head(text_2005_skip_counts, n = 10)
```

```
## # A tibble: 10 x 3
##    word1        word2       weight
##    <chr>        <chr>        <int>
##  1 sistema      operativo       12
##  2 mac          os              10
##  3 binarios     universales      7
##  4 has          visto            7
##  5 codigo       fuente           6
##  6 procesadores intel            6
##  7 año          viene            5
##  8 dejame       abrir            5
##  9 excelentes   productos        5
## 10 powerpc      intel            5
```

```
head(text_2008_skip_counts, n = 10)
```

```
## # A tibble: 10 x 3
##    word1        word2        weight
##    <chr>        <chr>         <int>
##  1 correo       electronico      25
##  2 gustaria     invitar          12
##  3 software     iphone           12
##  4 app          store            11
##  5 mis          contactos         8
##  6 barra        herramientas      6
```

```
##  7 directamente tu                6
##  8 interfaz     usuario           6
##  9 sdk          iphone            6
## 10 sitio        web               6
```

```
head(text_2010_skip_counts, n = 10)
```

```
## # A tibble: 10 x 3
##    word1        word2      weight
##    <chr>        <chr>       <int>
##  1 pantalla     retina         14
##  2 acero        inoxidable      7
##  3 wi           fi              7
##  4 estaciones   base            5
##  5 pixeles      pulgada         5
##  6 tu           telefono        5
##  7 alta         definicion      4
##  8 directamente tu              4
##  9 flash        led             4
## 10 imovie       iphone          4
```

```
##### definir una red a partir de la frecuencia (weight) de los bigramas
g <- text_1997_skip_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
g <- igraph::simplify(g)  # importante!
# grafo inducido por la componente conexa
V(g)$cluster <- clusters(graph = g)$membership
gcc <- induced_subgraph(graph = g, vids = which(V(g)$cluster == which.max(clusters(graph = g)$csize)))
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))
# viz 1
set.seed(123)
plot(gcc, layout = layout_with_fr, vertex.color = 1, vertex.frame.color = 1, vertex.size = 3, vertex.lab
# viz 2
set.seed(123)
plot(gcc, layout = layout_with_fr, vertex.color = adjustcolor('darkolivegreen4', 0.1), vertex.frame.colo
title(main = "Componente conexa", outer = T, line = -1)
```

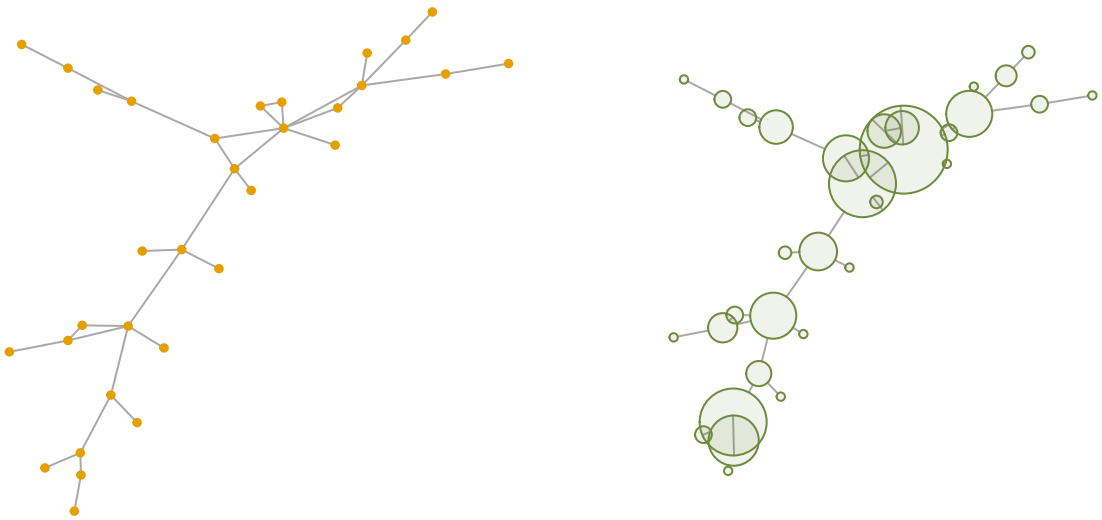
Componente conexa

```
##### definir una red a partir de la frecuencia (weight) de los bigramas
g <- text_2001_skip_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
g <- igraph::simplify(g)  # importante!
# grafo inducido por la componente conexa
V(g)$cluster <- clusters(graph = g)$membership
gcc <- induced_subgraph(graph = g, vids = which(V(g)$cluster == which.max(clusters(graph = g)$csize)))
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))
# viz 1
set.seed(123)
plot(gcc, layout = layout_with_fr, vertex.color = 1, vertex.frame.color = 1, vertex.size = 3, vertex.la
# viz 2
set.seed(123)
plot(gcc, layout = layout_with_fr, vertex.color = adjustcolor('darkolivegreen4', 0.1), vertex.frame.col
title(main = "Componente conexa", outer = T, line = -1)
```

Componente conexa

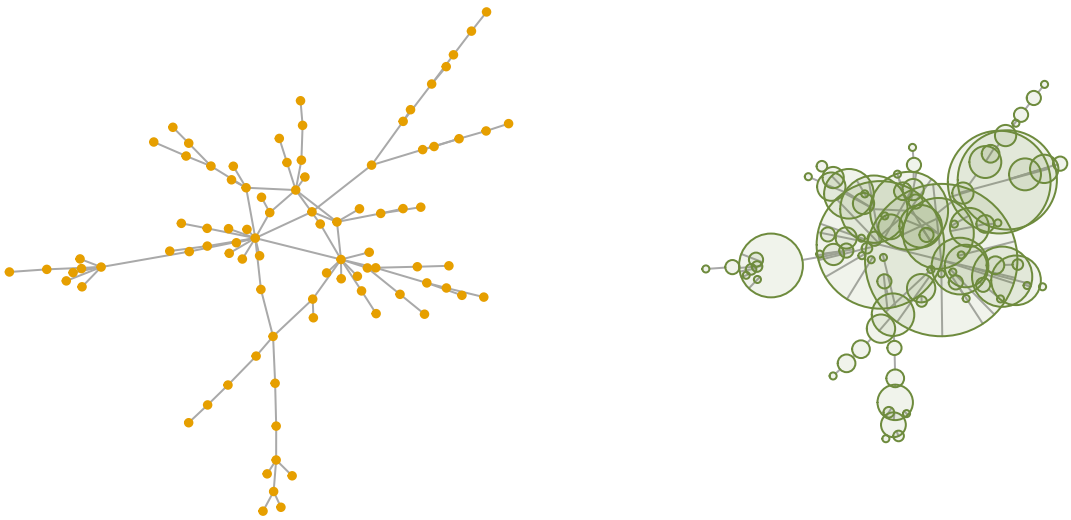
```
##### definir una red a partir de la frecuencia (weight) de los bigramas
g <- text_2005_skip_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
g <- igraph::simplify(g)  # importante!
# grafo inducido por la componente conexa
V(g)$cluster <- clusters(graph = g)$membership
gcc <- induced_subgraph(graph = g, vids = which(V(g)$cluster == which.max(clusters(graph = g)$csize)))
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))
# viz 1
set.seed(123)
plot(gcc, layout = layout_with_fr, vertex.color = 1, vertex.frame.color = 1, vertex.size = 3, vertex.la
# viz 2
set.seed(123)
plot(gcc, layout = layout_with_fr, vertex.color = adjustcolor('darkolivegreen4', 0.1), vertex.frame.col
title(main = "Componente conexa", outer = T, line = -1)
```

Componente conexa



```
##### definir una red a partir de la frecuencia (weight) de los bigramas
g <- text_2008_skip_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
g <- igraph::simplify(g)  # importante!
# grafo inducido por la componente conexa
V(g)$cluster <- clusters(graph = g)$membership
gcc <- induced_subgraph(graph = g, vids = which(V(g)$cluster == which.max(clusters(graph = g)$csize)))
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))
# viz 1
set.seed(123)
plot(gcc, layout = layout_with_fr, vertex.color = 1, vertex.frame.color = 1, vertex.size = 3, vertex.lab
# viz 2
set.seed(123)
plot(gcc, layout = layout_with_fr, vertex.color = adjustcolor('darkolivegreen4', 0.1), vertex.frame.colo
title(main = "Componente conexa", outer = T, line = -1)
```

Componente conexa



```
##### definir una red a partir de la frecuencia (weight) de los bigramas
g <- text_2010_skip_counts %>%
  filter(weight > 1) %>%
  graph_from_data_frame(directed = FALSE)
g <- igraph::simplify(g)  # importante!
# grafo inducido por la componente conexa
V(g)$cluster <- clusters(graph = g)$membership
gcc <- induced_subgraph(graph = g, vids = which(V(g)$cluster == which.max(clusters(graph = g)$csize)))
par(mfrow = c(1,2), mar = c(1,1,2,1), mgp = c(1,1,1))
# viz 1
set.seed(123)
plot(gcc, layout = layout_with_fr, vertex.color = 1, vertex.frame.color = 1, vertex.size = 3, vertex.la
# viz 2
set.seed(123)
plot(gcc, layout = layout_with_fr, vertex.color = adjustcolor('darkolivegreen4', 0.1), vertex.frame.col
title(main = "Componente conexa", outer = T, line = -1)
```

Componente conexa