

---

---

# ANÁLISIS SENTIMIENTOS

#Fradejas + [https://rpubs.com/jboscomendoza/coheed\\_and\\_cambria](https://rpubs.com/jboscomendoza/coheed_and_cambria)

---

---

# PASO\_1: Cargamos librerías básicas

```
library(rvest)
library(httr)
library(xml2)
library(jsonlite)
library(tidyverse)
library(tidytext)
library(lubridate)
library(scales)
```

# PASO\_2: CARGAMOS cargar el diccionario que vamos a usar. Se encuentra en un #repositorio externo y lo tienes que cargar con la función read\_tsv(), que permite leer # una tabla cuyas columnas están separadas por tabuladores (tab separated values).

```
sentimientos <-
read_tsv("~/Desktop/TEDEX_2/HISTORIAS_AMOR/diccionarios/sentimientos.txt",
        col_types = "cccn",
        locale = default_locale())
```

sentimientos

```
#####
# en la columna lexicon y puede ser nrc, afinn y bing). La columna valor es siempre NA
# para nrc y bing, y un valor numérico entre -5 y +5 para afinn. En cambio, afinn
# tiene siempre un valor NA en la columna sentimiento, mientras que en bing puede
#ser negativo o positivo y en nrc uno de estos diez valores: positivo, negativo, ira,
# miedo, tristeza, disgusto, asombro, confianza, alegría, premonición.
#Algunas palabras pueden tener en nrc varios valores, pero siempre relacionados.
# Fíjate en el cuadro anterior, “abandonar” está marcado como miedo, negativo y
# tristeza.
#####
```

# PASO\_3: A continuación cargamos una pequeña modificación de una de las #funciones de tidytext.

```
source("~/Desktop/TEDEX_2/HISTORIAS_AMOR/codigo/get_sentiments.R")
```

```
# PASO_4: Cargas los textos
chica <- read_lines("~/Desktop/TEDEX_2/HISTORIAS_AMOR/datos/Chica_limpio.txt",
locale = default_locale())
chico <- read_lines("~/Desktop/TEDEX_2/HISTORIAS_AMOR/datos/Chico_limpio.txt",
locale = default_locale())
```

```
# PASO_5: Creamos un vector de caracteres con el título de cada entrega, y lo llamamos 'titulos'
```

```
titulos <- c("Chica",
"Chico")
```

```
#PASO_6: Creamos una lista, llamada 'libros'
```

```
libros <- list(chica,
chico)
```

```
# PASO_7: Creamos una gran tabla, que llamaremos 'serie', con todos los textos, con #los títulos, los números de capítulo de cada una de ellos y con los textos divididos # en palabras-token
```

```
serie <- tibble()
for(i in seq_along(titulos)) {
  limpio <- tibble(capitulo = seq_along(libros[[i]]),
  texto = libros[[i]]) %>%
  unnest_tokens(palabra, texto) %>%
  mutate(libro = titulos[i]) %>%
  select(libro, everything())
  serie <- rbind(serie, limpio)
}
```

```
#####
# PASO_8: El último paso previo es convertir la columna de los títulos –libro– en un #factor (<fct>). La tabla, como acabas de ver, tiene varios cientos de miles de líneas, # y necesitamos reducir esos varios cientos de miles de líneas a tan solo diez valores. #Hay varias maneras de hacerlo. La más sencilla es indicándole a R qué columna de la tabla serie quieras convertir en factores. Usaremos el signo $ para referirnos a una # columna determinada de una tabla, de modo que serie$libro solo tendrá en cuenta # la columna libro de la tabla serie.
```

```
#####
```

```
serie$libro
serie$libro <- factor(serie$libro, levels = rev(titulos))
```

```
#####
#ANÁLISIS
#####
```

```
# PASO_9: Lo primero que vas a ver es cuántas palabras positivas y cuántas negativas
#hay en los textos analizados. Utilizaremos el diccionario 'nrc' o 'bing'.
```

```
serie %>%
  right_join(get_sentiments("nrc")) %>%
  filter(!is.na(sentimiento)) %>%
  count(sentimiento, sort = TRUE)%>%
  top_n(15)
```

```
# Si cambias el argumento de get_sentiments() de nrc a bing
```

```
serie %>%
  right_join(get_sentiments("bing")) %>%
  filter(!is.na(sentimiento)) %>%
  count(sentimiento, sort = TRUE)
```

```
#PASO_10: Visualización de resultados (El valor del eje horizontal indica el número de
#la página)
```

```
serie %>%
  group_by(libro) %>%
  mutate(recuento_palabras = 1:n(),
         indice = recuento_palabras %% 500 + 1) %>%
  inner_join(get_sentiments("nrc")) %>%
  count(libro, indice = indice , sentimiento) %>%
  ungroup() %>%
  spread(sentimiento, n, fill = 0) %>%
  mutate(sentimiento = positivo - negativo, libro = factor(libro, levels = titulos)) %>%
  ggplot(aes(indice, sentimiento, fill = libro)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ libro, ncol = 2, scales = "free_x")
```

```
# PASO_11: Hacemos el cuento de palabras con bing
```

```
recuenta_palabras_bing <- serie %>%
  inner_join(get_sentiments("bing")) %>%
  count(palabra, sentimiento, sort = TRUE)
```

```
# PASO_12: Vemos los resultados en gráfico
```

```
recuenta_palabras_bing %>%
  group_by(sentimiento) %>%
  top_n(25) %>%
```

```

ggplot(aes(reordered(palabra, n), n, fill = sentimiento)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~sentimiento, scales = "free_y") +
  labs(y = "Contribución al sentimiento", x = NULL) +
  coord_flip()

```

#PASOS\_13: Comparamos nubes de POSITIVO y NEGATIVO

```

library(reshape2)
library(wordcloud)
serie %>%
  inner_join(get_sentiments("uva")) %>%
  count(palabra, sentimiento, sort = TRUE) %>%
  acast(palabra ~ sentimiento, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("gray20", "gray80"),
                    max.words = 50)

```

#PASO\_14: Hacemos gráficos por porcentajes

```

serie %>%
  group_by(libro) %>%
  inner_join(get_sentiments("nrc")) %>%
  count(sentimiento) %>%
  mutate(prop = n / sum(n)) %>%
  ggplot() +
  aes(libro, prop, fill = sentimiento) +
  geom_col(position = "stack", color = "black") +
  coord_flip() +
  scale_y_continuous(expand = c(0,0)) +
  theme_minimal()

```

# PASO\_15: Repetimos el mismo gráfico eliminando ahora los sentimientos "positivo" #y "negativo"

```

serie %>%
  group_by(libro) %>%
  inner_join(get_sentiments("nrc")) %>%
  count(sentimiento) %>%
  filter(!sentimiento %in% c("positivo", "negativo")) %>%
  mutate(prop = n / sum(n)) %>%
  ggplot() +
  aes(libro, prop, fill = sentimiento) +
  geom_col(position = "stack", color = "black") +
  coord_flip() +
  scale_y_continuous(expand = c(0,0)) +
  theme_minimal()

```

```
# PASO_16: Hacemos el recuento de palabras por sentimiento
```

```
recuenta_palabras<- serie %>%
  inner_join(get_sentiments("nrc")) %>%
  count(palabra, sentimiento, sort = TRUE)
```

```
# PASO_17: Vemos los resultados en gráfico
```

```
recuenta_palabras %>%
  group_by(sentimiento) %>%
  top_n(25) %>%
  ggplot(aes(reorder(palabra, n), n, fill = sentimiento)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~sentimiento, scales = "free_y") +
  labs(y = "Contribución al sentimiento", x = NULL) +
  coord_flip()
```

#PASO\_18: Quitaremos las palabras que no definen nadad de nuestros datos para mejorar la interpretabilidad de nuestros resultados.Eliminamos palabras que no definen: esta función y volvemos a Graficar

```
recuenta_palabras <-
recuenta_palabras %>%
filter(!palabra %in% c("señor", "madre", "dios", "niña", "boca", "sol"))
```

```
# PASO_19: Podemos verlo en gráfico
```

```
recuenta_palabras %>%
  group_by(sentimiento) %>%
  top_n(25) %>%
  ggplot(aes(reorder(palabra, n), n, fill = sentimiento)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~sentimiento, scales = "free_y") +
  labs(y = "Contribución al sentimiento", x = NULL) +
  coord_flip()
```

```
# PASO_20: Comprobamos el sentimiento dominante por libro
```

```
serie %>%
  inner_join(get_sentiments("nrc")) %>%
  group_by(libro) %>%
  count(sentimiento) %>%
  mutate(prop = n / sum(n)) %>%
  top_n(1, wt = prop) %>%
```

```

ggplot() +
  aes(libro, prop, fill = sentimiento) +
  geom_col(position = "stack", color = "black") +
  coord_flip() +
  scale_y_continuous(expand = c(0,0)) +
  theme_minimal()

# PASO_21: Comprobamos el sentimiento menos dominante por libro

serie %>%
  inner_join(get_sentiments("nrc")) %>%
  group_by(libro) %>%
  count(sentimiento) %>%
  mutate(prop = n / sum(n)) %>%
  top_n(-1, wt = prop) %>%
  ggplot() +
  aes(libro, prop, fill = sentimiento) +
  geom_col(position = "stack", color = "black") +
  coord_flip() +
  scale_y_continuous(expand = c(0,0)) +
  theme_minimal()

#####
#LIBRERÍA syuzhet
#####

#####
#PROVISIONAL
#####

# Análisis de cómo funciona un sentimiento en cada uno de los textos
#sentimiento en particular, acompañadas del disco al que pertenecen.

```

```

serie %>%
  inner_join(get_sentiments("nrc")) %>%
  group_by(libro) %>%
  count(sentimiento) %>%
  mutate(prop = n / sum(n)) %>%
  group_by(sentimiento) %>%
  top_n(5) %>%
  ggplot() +
  aes(sentimiento, prop, color = sentimiento) +
  geom_point() +
  geom_text(aes(label = paste0(libro, "\n")),
            vjust = -.3, size = 3) +
  scale_y_continuous(limits = c(0.15, 0.6)) +

```

```

theme_minimal() +
theme(legend.position = "none")

#####
# Cuando se trata de libros con varios poemas: hay que revisar la tabla "serie" con
# los columnas que vamos a usar aquí
#####
graficar_poema <- function(sentimiento, cantidad = 7) {
  serie %>%
    inner_join(get_sentiments("nrc")) %>%
    group_by(libro, poema) %>%
    count(sentimiento) %>%
    mutate(prop = n / sum(n)) %>%
    group_by(sentimiento) %>%
    top_n(cantidad) %>%
    filter(sentimiento == sentimiento) %>%
    mutate(poema = paste0(poema, "\n(", libro, ")"),
          poema = reorder(poema, prop)) %>%
    ggplot() +
    aes(cancion, prop) +
    geom_col(position = "dodge", fill = "#bb88ff") +
    theme_minimal() +
    theme(legend.position = "none", text = element_text(family = "serif")) +
    coord_flip() +
    labs(title = paste0("Libro\nPoema con más ", sentimiento),
         x = "Poema (Libro)", y = "Proporcion") +
    scale_y_continuous(limits = c(0, .6), expand = c(0, 0), label = percent_format())
}

#Probamos que funciona con la tristeza.
graficar_cancion("tristeza", 5)

```