```
##############################################################
#ANÁLISIS TRAMA NARRATIVA BASADA EN DICCIONARIOS DE EMOCIONES#
##############################################################

#PASO_1: Establecer directorio de trabajo

# PASO_2: Cargamos las tres librerías que vamos a usar

library(tidyverse)
library(tidytext)
library(syuzhet)


# PASO_3: Cargar el diccionario  de sentimientos y la función get_sentiments.R:

sentimientos <- read_tsv("~/Desktop/sentimientos/diccionarios/sentimientos.txt",
                    col_types = "cccn",
                    locale = default_locale())
source("~/Desktop/sentimientos/codigo/get_sentiments.R")


# PASO_4: Cargamos el texto que se va a analizar y comprobamos que se ha
incorporado bien abriendo el comienzo y el final del texto

texto_entrada <-
read_lines("~/Desktop/sentimientos/datos/manipulacion/maria_limpio.txt",
                      locale = default_locale())

head(texto_entrada)
tail(texto_entrada)

# PASO_5: Convertimos  el textoen tibble para poder dividirlo en palabras-token
# y hacer los primeros cálculos.

texto_analizar <- tibble(texto = texto_entrada)

# PASO_6: Dividimos el texto en palabras-tokens y le ordenamos
# a) que lo divida también en páginas (de 400 palabras);
# b) que estableca qué palabras son positivas, cuáles negativas;
# c) que recuente los sentimientos de cada página y genere una tabla interna con la
# valencia para cada página.
# d) que por último convierta los números de la columna  negativo en número
#negativos.
#El código para hacer todo esto es

texto_analizar <- texto_analizar %>%
  unnest_tokens(palabra, texto) %>%
```

```
  mutate(pagina = (1:n()) %/% 400 + 1) %>%
  inner_join(get_sentiments("nrc")) %>%
  count(sentimiento, pagina = pagina) %>%
  spread(sentimiento, n, fill = 0) %>%
  mutate(negativo = negativo*-1)

texto_analizar

#PASO_7: Para representar gráficamente el resultado, hemos de calcular la puntuación
#para cada página.

puntuacion <- texto_analizar %>%
  mutate(sentimiento = positivo+negativo) %>%
  select(pagina, sentimiento)

# PASO_8: Ahora podemos ya representar gráficamente el resultado.
ggplot(data = puntuacion, aes(x = pagina, y = sentimiento)) +
  geom_bar(stat = "identity", color = "midnightblue") +
  theme_minimal() +
  ylab("Sentimiento") +
  xlab("Tiempo narrativo") +
  ggtitle(expression(paste("Sentimiento en ",
                           italic("Los cuatro jinetes del Apocalipsis")))) +
  theme(legend.justification=c(0.91,0), legend.position=c(1, 0))

#PASO_9: Quitamos el ruido, para ver con nitidez la curva de la línea por la que se
#transita la narración. Aquí es donde entra la magia del paquete syuzhet

texto_trans <- get_dct_transform(puntuacion$sentimiento,
                                 low_pass_size = 10,
                                 x_reverse_len = nrow(puntuacion),
                                 scale_range = TRUE)

#PASO_10: Precisamos a qué porción de texto corresponden los datos observados
# porción del texto corresponde,
texto_trans

# PASO_11: Creamos una tabla que reúna la información para los gráficos

texto_trans <- tibble(pagina = seq_along(texto_trans),
                      ft = texto_trans)

texto_trans


# PASO_12: Ahora estás en disposición de dibujar la gráfica.
```

```
ggplot(texto_trans, aes(x = pagina, y = ft)) +
  geom_bar(stat = "identity", alpha = 0.8,
           color = "aquamarine3", fill = "aquamarine3") +
  theme_minimal() +
  labs(x = "Tiempo narrativo",
       y = "Transformación Valorada del Sentimiento") +
  ggtitle(expression(paste("Forma de la historia de ",
                           italic("Los cuatro jinetes del Apocalipsis"))))

# Otra forma de gráfica sencilla  de líneas que se obtiene con la función plot()
plot(texto_trans,
     type = "l",
     yaxt = 'n',
     ylab = "",
     xlab = "Tiempo narrativo",
     main = "La forma de la historia:\n
Los cuatro jinetes del Apocalipsis")
abline(h = 0.0, col = "red")
```