3.1.3 Caso práctico: El estado de ánimo de mi libro o la noticia del día

```
library(tidyverse)
library(tidytext)
library(textdata)

# Cargar el texto del libro
texto_libro <- readLines("libro.txt")

# Convertir el texto a un data frame
df_texto <- tibble(texto = texto_libro)

# Tokenización del texto
df_tokenizado <- df_texto %>%
  unnest_tokens(word, texto)

# Cargar el lexicon de sentimientos
data("bing")

# Unirse al lexicon de sentimientos
df_sentimientos <- df_tokenizado %>%
  inner_join(bing, by = "word")

# Contar los sentimientos positivos y negativos
sentimientos_contados <- df_sentimientos %>%
  count(sentiment) %>%
  spread(sentiment, n, fill = 0)

# Calcular el estado de ánimo general
sentimientos_contados$estado_animo <- ifelse(sentimientos_contados$positive > sentimientos_contados$negative, "Positivo",
                                             ifelse(sentimientos_contados$positive < sentimientos_contados$negative, "Negativo", "Neutral"))

# Imprimir el resultado
print(sentimientos_contados)
```

Este código cargará el texto del libro "seleccionado", analizará los sentimientos de cada palabra utilizando el lexicon de sentimientos bing, contará el número de palabras positivas y negativas, y determinará si el estado de ánimo general del texto es positivo, negativo o neutral.

Características de un corpus big data:

- Volumen masivo: Un corpus big data típicamente contiene una cantidad enorme de datos de texto. Esto puede ser desde millones hasta miles de millones de documentos o más.
- Variedad de fuentes: Los datos en un corpus big data pueden provenir de diversas fuentes, como redes sociales, sitios web, documentos gubernamentales, foros en línea, noticias, entre otros. Esta variedad de fuentes puede enriquecer el corpus con una amplia gama de lenguaje y estilos de escritura.
- Velocidad de adquisición: La recopilación de datos en un corpus big data puede ser continua y en tiempo real, lo que implica una alta velocidad de adquisición de datos para mantenerse al día con la información que se genera constantemente en línea.
- Complejidad de procesamiento: Dado el tamaño masivo del corpus, el procesamiento de datos y el análisis lingüístico pueden requerir técnicas y herramientas específicas de big data para manejar eficientemente la carga de trabajo.
- Diversidad lingüística: Debido a la variedad de fuentes de datos, un corpus big data puede contener textos en varios idiomas y dialectos, lo que lo hace útil para análisis multilingües y estudios comparativos.
- Desafíos de almacenamiento y procesamiento: El almacenamiento y procesamiento de un corpus big data pueden ser desafiantes debido a la necesidad de infraestructura de almacenamiento y computación escalable.

Ejemplo

```
# Cargar el paquete `tm` para el análisis de texto
library(tm)

# Crear un vector con las rutas a los archivos del corpus
archivos <- list.files("ruta/al/corpus", full.names = TRUE)

# Función para leer un archivo de texto y convertirlo a un corpus
leer_corpus <- function(archivo)
{ texto <- readLines(archivo)
corpus(VectorSource(texto)) }

# Crear un corpus a partir de los archivos
corpus <- lapply(archivos, leer_corpus)

# Combinar todos los documentos en un solo corpus
corpus <- tm_corpus(corpus)



# Convertir todo el texto a minúsculas
corpus <- tm_map(corpus, content_transformer(tolower))


# Eliminar puntuación y símbolos especiales
corpus <- tm_map(corpus, removePunctuation)


# Eliminar palabras vacías (stopwords)
corpus <- tm_map(corpus, removeWords, stopwords("es"))


# Lemmatizar las palabras (opcional)
#corpus <- tm_map(corpus, stemDocument)
```

```
# Obtener la frecuencia de términos
frecuencias <- tm_term_matrix(corpus)

# Visualizar las 10 palabras más frecuentes
top_palabras <- sort(colSums(frecuencias), decreasing = TRUE)[1:10]
print(top_palabras)

# Crear un gráfico de nube de palabras
wordcloud(corpus, size = sqrt(colSums(frecuencias)), min.freq = 5)

# Calcular la distancia entre documentos
distancia <- dist(t(frecuencias))

# Visualizar la distancia entre documentos mediante un MDS
mds <- cmdscale(distancia)
plot(mds, labels = names(corpus))



# Crear un modelo LDA (Análisis Discriminante Lineal)
modelo_lda <- LDA(corpus, k = 5)

# Visualizar los tópicos
print(topics(modelo_lda, 5))

# Asignar cada documento a un tópico
doc_topic <- classify(modelo_lda, corpus)

# Visualizar la distribución de tópicos por documento
barplot(table(doc_topic))
```

```
# Ejemplo en R para administrar Palabras de opinión negativa de grandes volúmenes de datos

library(tm)
library(SnowballC)
library(wordcloud)
library(tidyverse)

# Ejemplo de datos
corpus <- Corpus(VectorSource(c("Este producto es terrible", "El servicio es pésimo", "No lo recomiendo", "Muy mala
experiencia")))

corpus <- tm_map(corpus, content_transformer(tolower)) # Convertir a minúsculas
corpus <- tm_map(corpus, removePunctuation) # Eliminar puntuación
corpus <- tm_map(corpus, removeWords, stopwords("es")) # Eliminar palabras vacías
corpus <- tm_map(corpus, stemDocument) # Lematización


# Convertir a matriz de términos-documento
dtm <- DocumentTermMatrix(corpus)



# Frecuencia de palabras
freq <- sort(colSums(as.matrix(dtm)), decreasing = TRUE)

# Nube de palabras
wordcloud(names(freq)[1:20], freq[1:20], min.freq = 5)

# Diccionario de palabras negativas
negatives <- c("terrible", "pésimo", "malo", "deficiente", "insatisfactorio")

# Identificar palabras negativas en el corpus
corpus_neg <- tm_map(corpus, function(x) sum(grepl(negatives, x)) > 0)

# Proporción de documentos con palabras negativas
prop_neg <- mean(corpus_neg)

# Filtrar documentos con palabras negativas
corpus_negativo <- corpus[corpus_neg]
```

```
# Gráfico de barras con la frecuencia de palabras negativas
barplot(table(corpus_negativo))

# Exportar frecuencia de palabras
write.csv(freq, "frecuencia_palabras.csv")

# Exportar documentos con palabras negativas
write.csv(corpus_negativo, "documentos_negativos.csv")
```

E-books

- Bouso Freijo, J. (2018). El paquete estadístico R: (2 ed.). CIS - Centro de Investigaciones Sociológicas. https://elibro.net/es/lc/ucags/titulos/105698
- Royé, D. & Serrano Notivoli, R. (2019). Introducción a los SIG con R: (ed.). Prensas de la Universidad de Zaragoza. https://elibro.net/es/lc/ucags/titulos/122173
- Mas Elias, J. (2020). Análisis de datos con R en estudios internacionales: (ed.). Editorial UOC. https://elibro.net/es/lc/ucags/titulos/167261
- Alonso, J. C. & Largo, M. F. (2022). Empezando a visualizar datos con R y ggplot2: (1 ed.). Editorial Universidad Icesi. https://elibro.net/es/lc/ucags/titulos/225846
- Pujol Jover, M. & Pujol Jover, M. (2017). Análisis cuantitativo con R: matemáticas, estadística y econometría: (ed.). Editorial UOC. https://elibro.net/es/lc/ucags/titulos/58652
- Cabrero Ortega, M. Y. & García Pérez, A. (2022). Análisis estadístico de datos espaciales con QGIS y R: (1 ed.). UNED - Universidad Nacional de Educación a Distancia. https://elibro.net/es/lc/ucags/titulos/218566
- Gil Pascual, J. A. (2021). Minería de texto con R: aplicaciones y técnicas estadísticas de apoyo: (ed.). UNED - Universidad Nacional de Educación a Distancia. https://elibro.net/es/lc/ucags/titulos/188719
- Iryopogu, J. (2021). Análisis de datos con Power BI, R-RStudio y Knime: curso práctico: (1 ed.). RA-MA Editorial. https://elibro.net/es/lc/ucags/titulos/222665

Referencias

- Shalabh, Shalabh. (2023). The Big R-Book: From Data Science to Learning Machines and Big Data. Journal of the Royal Statistical Society Series A: Statistics in Society. 186. 896-897. 10.1093/jrsssa/qnad029.
- Balazka, Dominik & Rodighiero, Dario. (2020). Big Data and the Little Big Bang: An Epistemological (R)evolution. Frontiers in Big Data. 3. 1-13. 10.3389/fdata.2020.00031.
- Hodeghatta, U.R. & Nayak, U.. (2016). Business analytics using R-A practical approach. 10.1007/978-1-4842-2514-1.
- Tripathi, Subhashini. (2016). Learn Business Analytics in Six Steps Using SAS and R. 10.1007/978-1-4842-1001-7.
- Ohri, Ajay. (2013). R for Business Analytics. 10.1007/978-1-4614-4343-8.