Análisis de Datos de Salud y Cuidado Personal

STDT Ing. George Albadr

2024-10-21

Cargar los conjuntos de datos

```
# Librerías y carga de datos
suppressPackageStartupMessages({
  library(readr)
  library(dplyr)
  library(tm)
  library(wordcloud)
  library(RColorBrewer)
})

# Cargar los datos
Health_and_Personal_Care_metadata <- read_csv("Health_and_Personal_Care_metadata.csv")
Health_and_Personal_Care <- read_csv("Health_and_Personal_Care.csv")
```

Preguntas a responder:

1. ¿Cuántos productos contienen reviews con las palabras “love”, “recommend” y “enjoy”?

```
# Filtrar las reseñas que contienen todas las palabras clave: "love", "recommend" y "enjoy"
filtered_reviews <- Health_and_Personal_Care %>%
  filter(
    grepl("love", text, ignore.case = TRUE) &
    grepl("recommend", text, ignore.case = TRUE) &
    grepl("enjoy", text, ignore.case = TRUE)
  )

# Contar el número de productos únicos que cumplen con el criterio
unique_products <- filtered_reviews %>%
  distinct(product_id) %>%
  count() %>%
  pull(n)

# Mostrar el resultado
cat("Número de productos con reseñas que contienen 'love', 'recommend' y 'enjoy':", unique_products, "\n
```

```
## Número de productos con reseñas que contienen 'love', 'recommend' y 'enjoy': 110
```

2. De los reviews de la pregunta 1, encuentre el top 5 de las tiendas que los venden?

```
# Unir las reseñas filtradas con los metadatos para obtener información de la tienda
merged_data <- filtered_reviews %>%
  inner_join(Health_and_Personal_Care_metadata, by = c("product_id" = "parent_id"))
```

```r
# Contar el número de reseñas por tienda
store_counts <- merged_data %>%
  group_by(store) %>%
  summarise(review_count = n()) %>%
  arrange(desc(review_count))

# Obtener las 5 tiendas con más reseñas
top_5_stores <- store_counts %>%
  slice_max(order_by = review_count, n = 5)

# Mostrar el resultado
cat("Top 5 tiendas con más reseñas filtradas:\n")
```

```
## Top 5 tiendas con más reseñas filtradas:
```

```r
print(top_5_stores)
```

```
## # A tibble: 5 x 2
##   store               review_count
##   <chr>                      <int>
## 1 <NA>                           7
## 2 Bestrice                       2
## 3 Jitner                         2
## 4 Sweetsation Therapy            2
## 5 sequel 65                      2
```

3. Generar un wordcloud sin stopwords de los reviews de la pregunta 1.

```r
# Combinar todo el texto de las reseñas filtradas en un solo string
all_text <- paste(filtered_reviews$text, collapse = " ")

# Crear un corpus de texto para el procesamiento
corpus <- Corpus(VectorSource(all_text))

# Limpieza del texto:
# - Convertir a minúsculas
# - Eliminar puntuación
# - Eliminar números
# - Eliminar stopwords en inglés
corpus <- corpus %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(removePunctuation) %>%
  tm_map(removeNumbers) %>%
  tm_map(removeWords, stopwords("en"))
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(tolower)): transformation
## drops documents
```

```
## Warning in tm_map.SimpleCorpus(., removePunctuation): transformation drops
## documents
```

```
## Warning in tm_map.SimpleCorpus(., removeNumbers): transformation drops
## documents
```

```
## Warning in tm_map.SimpleCorpus(., removeWords, stopwords("en")): transformation
## drops documents
```

```r
# Generar la nube de palabras
wordcloud(
  words = corpus,
  max.words = 100,
  random.order = FALSE,
  colors = brewer.pal(8, "Dark2")
)
```

```
## Warning in wordcloud(words = corpus, max.words = 100, random.order = FALSE, :
## recommended could not be fit on page. It will not be plotted.
```

```
## Warning in strwidth(words[i], cex = size[i], ...): conversion failure on ''s'
## in 'mbcsToSbcs': dot substituted for <e2>
```

```
## Warning in strwidth(words[i], cex = size[i], ...): conversion failure on ''s'
## in 'mbcsToSbcs': dot substituted for <80>
```

```
## Warning in strwidth(words[i], cex = size[i], ...): conversion failure on ''s'
## in 'mbcsToSbcs': dot substituted for <99>
```

```
## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : conversion failure on ''s' in 'mbcsToSbcs': dot substituted for
## <e2>
```

```
## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : conversion failure on ''s' in 'mbcsToSbcs': dot substituted for
## <80>
```

```
## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : conversion failure on ''s' in 'mbcsToSbcs': dot substituted for
## <99>
```

```
## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : font metrics unknown for Unicode character U+2019
```

```
## Warning in wordcloud(words = corpus, max.words = 100, random.order = FALSE, :
## problem could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = corpus, max.words = 100, random.order = FALSE, :
## minutes could not be fit on page. It will not be plotted.
```

4. Generar un wordcloud de los reviews de las 5 tiendas encontradas en la pregunta 2. Deberá de incluir todos los reviews de esas 5 tiendas

```
# Filtrar las reseñas que pertenecen a las 5 tiendas principales
store_reviews <- merged_data %>%
  filter(store %in% top_5_stores$store)

# Combinar todo el texto de estas reseñas
all_store_text <- paste(store_reviews$text, collapse = " ")

# Crear un corpus de texto para el procesamiento
store_corpus <- Corpus(VectorSource(all_store_text))

# Limpieza del texto:
# - Convertir a minúsculas
# - Eliminar puntuación
# - Eliminar números
# - Eliminar stopwords en inglés
store_corpus <- store_corpus %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(removePunctuation) %>%
  tm_map(removeNumbers) %>%
  tm_map(removeWords, stopwords("en"))
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(tolower)): transformation
## drops documents
```

```
## Warning in tm_map.SimpleCorpus(., removePunctuation): transformation drops
## documents
```

```
## Warning in tm_map.SimpleCorpus(., removeNumbers): transformation drops
## documents
```

```
## Warning in tm_map.SimpleCorpus(., removeWords, stopwords("en")): transformation
## drops documents
```

```
# Generar la nube de palabras
wordcloud(
  words = store_corpus,
  max.words = 100,
  random.order = FALSE,
  colors = brewer.pal(8, "Dark2")
)
```

5. ¿Cuáles son las 25 palabras más frecuentes de los reviews?

```
# Crear una matriz de términos y documentos (TermDocumentMatrix)
tdm <- TermDocumentMatrix(corpus)

# Convertir la matriz a formato de matriz estándar de R
matrix <- as.matrix(tdm)

# Calcular la frecuencia de cada palabra sumando las ocurrencias en todos los documentos
word_freq <- sort(rowSums(matrix), decreasing = TRUE)

# Extraer las 25 palabras más frecuentes
top_25_words <- head(word_freq, 25)

# Mostrar las palabras más frecuentes
cat("Las 25 palabras más frecuentes en las reseñas filtradas son:\n")
```

```
## Las 25 palabras más frecuentes en las reseñas filtradas son:
print(top_25_words)
```

```
##      love      like recommend   product     enjoy       one    really      just
##       115       100        95        82        81        76        75        72
##       can      well       use      will      also       get     great      good
##        70        66        65        60        57        54        54        51
##     using      time    highly    little     first      much     water      skin
##        49        48        41        40        39        38        36        35
##      foam
##        34
```