```r
# Librerie necessarie
library(readr)
library(dplyr)
library(stringr)
library(ggplot2)

# 1. Carica il file CSV (usa solo X1 e X2)
df_raw <- read_delim("lion king.csv", delim = ";", col_names = FALSE, trim_ws = TRUE)
df_raw <- df_raw[, 1:2]
colnames(df_raw) <- c("personaggio", "dialogo")

# 2. Pulizia dei dati
df_dialoghi <- df_raw %>%
  filter(!is.na(personaggio), !is.na(dialogo)) %>%
  filter(!str_detect(personaggio, "^\\{"), !str_detect(dialogo, "^\\{")) %>%
  mutate(
    personaggio = str_squish(personaggio),
    dialogo = str_squish(dialogo),
    personaggio = str_to_title(personaggio),
    personaggio = case_when(
      str_detect(personaggio, "Simba") ~ "Simba",
      str_detect(personaggio, "Nala|Nale") ~ "Nala",
      str_detect(personaggio, "Zazu") ~ "Zazu",
      str_detect(personaggio, "Mufasa") ~ "Mufasa",
      str_detect(personaggio, "Scar") ~ "Scar",
      str_detect(personaggio, "Rafiki") ~ "Rafiki",
      str_detect(personaggio, "Timon") ~ "Timon",
      str_detect(personaggio, "Pumbaa|Pumba") ~ "Pumbaa",
      str_detect(personaggio, "Sarabi") ~ "Sarabi",
      str_detect(personaggio, "Hyena|Shenzi|Banzai|Ed") ~ "Iene",
      TRUE ~ personaggio
    )
  )

# 3. Grafico dei personaggi con più battute
df_dialoghi %>%
  count(personaggio) %>%
  slice_max(n, n = 20) %>%
  ggplot(aes(x = reorder(personaggio, n), y = n, fill = personaggio)) +
  geom_col(show.legend = FALSE) +
  coord_flip() +
  labs(
    title = "Personaggi con più battute",
    x = "Personaggio",
    y = "Numero di battute"
  ) +
  theme_minimal()


# Librerie necessarie
library(tidytext)
library(tm)
library(wordcloud)
library(RColorBrewer)

# Estrai le parole dai dialoghi
parole <- df_dialoghi %>%
  unnest_tokens(word, dialogo) %>%
  anti_join(stop_words, by = "word") %>%  # rimuove parole comuni tipo "the", "and"
  count(word, sort = TRUE)

# 4.Wordcloud delle parole più frequenti
set.seed(123)
wordcloud(
  words = parole$word,
  freq = parole$n,
  min.freq = 5,
  max.words = 100,
  random.order = FALSE,
```

```
  colors = brewer.pal(8, "Dark2")
)



# Filtra solo i dialoghi di Simba
simba_dialoghi <- df_dialoghi %>%
  filter(personaggio == "Simba")

# Tokenizza e conta parole (rimuovendo stopwords)
simba_parole <- simba_dialoghi %>%
  unnest_tokens(word, dialogo) %>%
  anti_join(stop_words, by = "word") %>%
  count(word, sort = TRUE)

# 5. Grafico parole più usate da Simba
simba_parole %>%
  slice_max(n, n = 15) %>%
  ggplot(aes(x = reorder(word, n), y = n)) +
  geom_col(fill = "#F4C430") +
  coord_flip() +
  labs(
    title = "Le parole più usate da Simba",
    x = "Parola",
    y = "Frequenza"
  ) +
  theme_minimal()



# Usa dizionario Bing (positivo/negativo)
sentimenti <- df_dialoghi %>%
  unnest_tokens(word, dialogo) %>%
  inner_join(get_sentiments("bing")) %>%
  count(sentiment, sort = TRUE)

# 6. Grafico sentiment complessivo
ggplot(sentimenti, aes(x = sentiment, y = n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  scale_fill_manual(values = c("positive" = "#66C2A5", "negative" = "#FC8D62")) +
  labs(
    title = "Sentiment complessivo nei dialoghi",
    x = "Sentiment",
    y = "Numero di parole"
  ) +
  theme_minimal()




# Librerie necessarie
library(tidytext)
library(ggplot2)
library(dplyr)
library(stringr)

# Filtra solo Simba, Scar e Timon
personaggi_focus <- c("Simba", "Scar", "Timon")
dialoghi_focus <- df_dialoghi %>%
  filter(personaggio %in% personaggi_focus)

# Tokenizza, unisci con dizionario di sentimenti
sentiment_personaggi <- dialoghi_focus %>%
  unnest_tokens(word, dialogo) %>%
  inner_join(get_sentiments("bing")) %>%
  count(personaggio, sentiment, sort = TRUE)

# 7. Grafico a barre
ggplot(sentiment_personaggi, aes(x = personaggio, y = n, fill = sentiment)) +
```

```
geom_col(position = "dodge") +
scale_fill_manual(values = c("positive" = "#66C2A5", "negative" = "#FC8D62")) +
labs(
  title = "Sentiment dei dialoghi: Simba vs Scar vs Timon",
  x = "Personaggio",
  y = "Numero di parole con sentiment",
  fill = "Sentiment"
) +
theme_minimal()
```