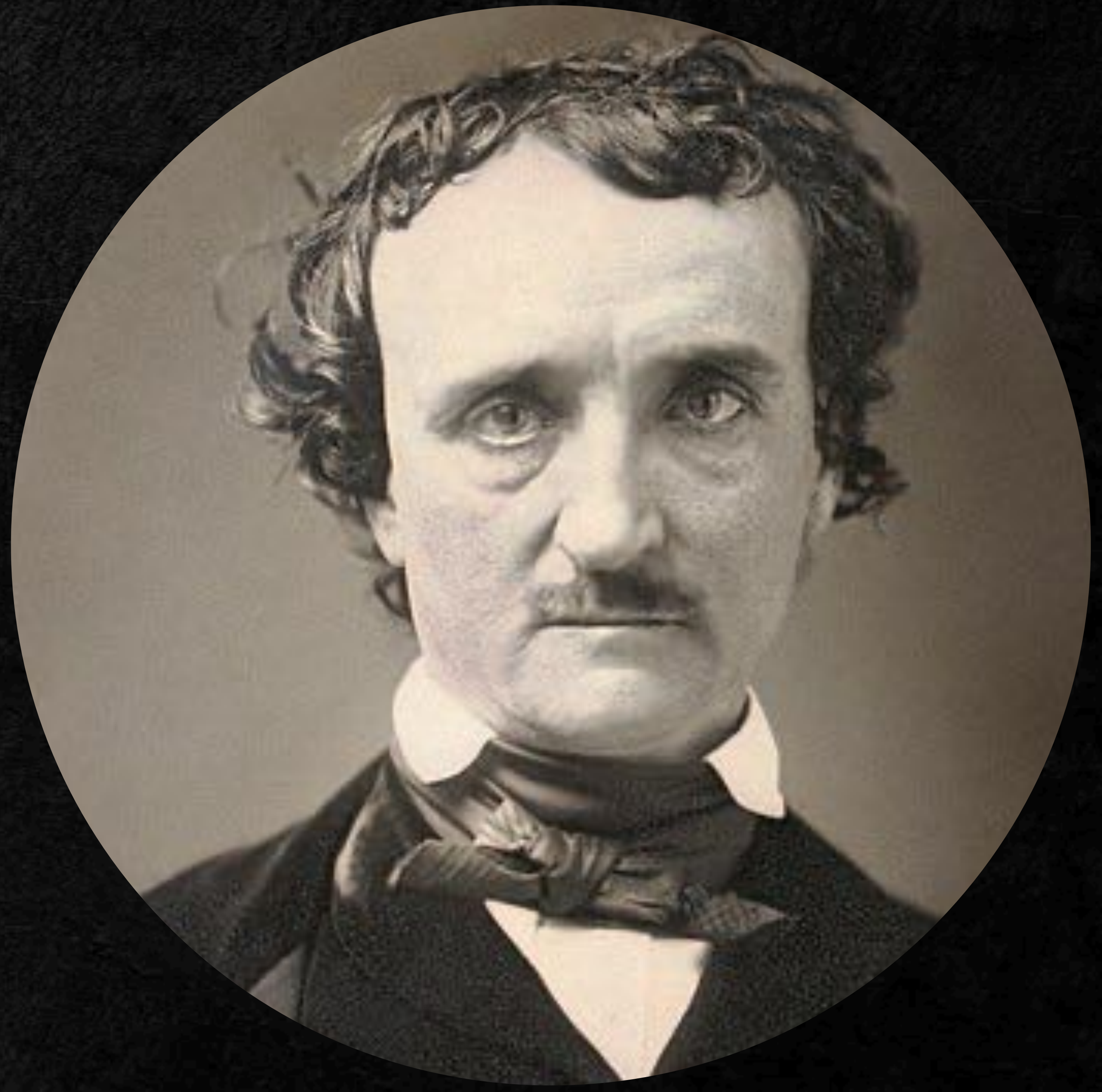
# PROGRAMANDO IA COM R

Treemap House of Horror: Spooky EDA/LDA/Features

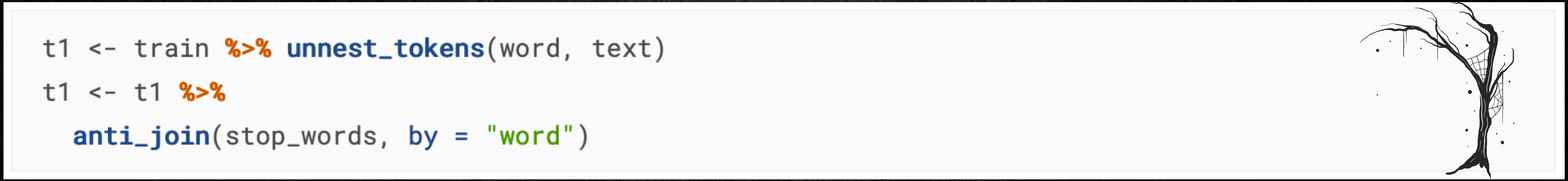
# ANALISE DE AUTORES DE TERROR



# COMO O **R** FOI USADO

# CÓDIGOS

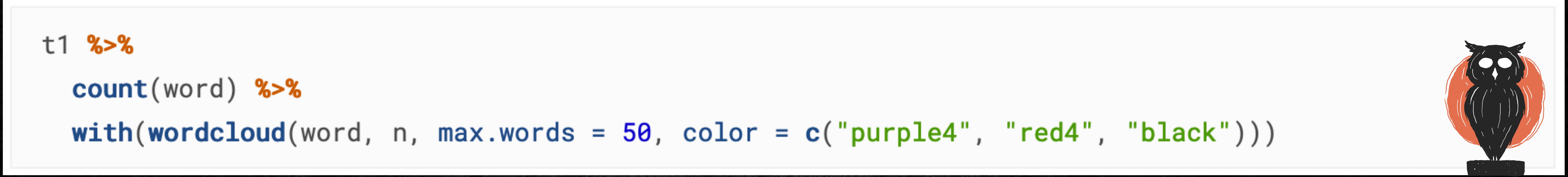
**Tratamento:** Eliminar pontos / Deixar todas as letras em minúsculo / Identificar palavras de parada (e/próximo)



```r
t1 <- train %>% unnest_tokens(word, text)
t1 <- t1 %>%
  anti_join(stop_words, by = "word")
```

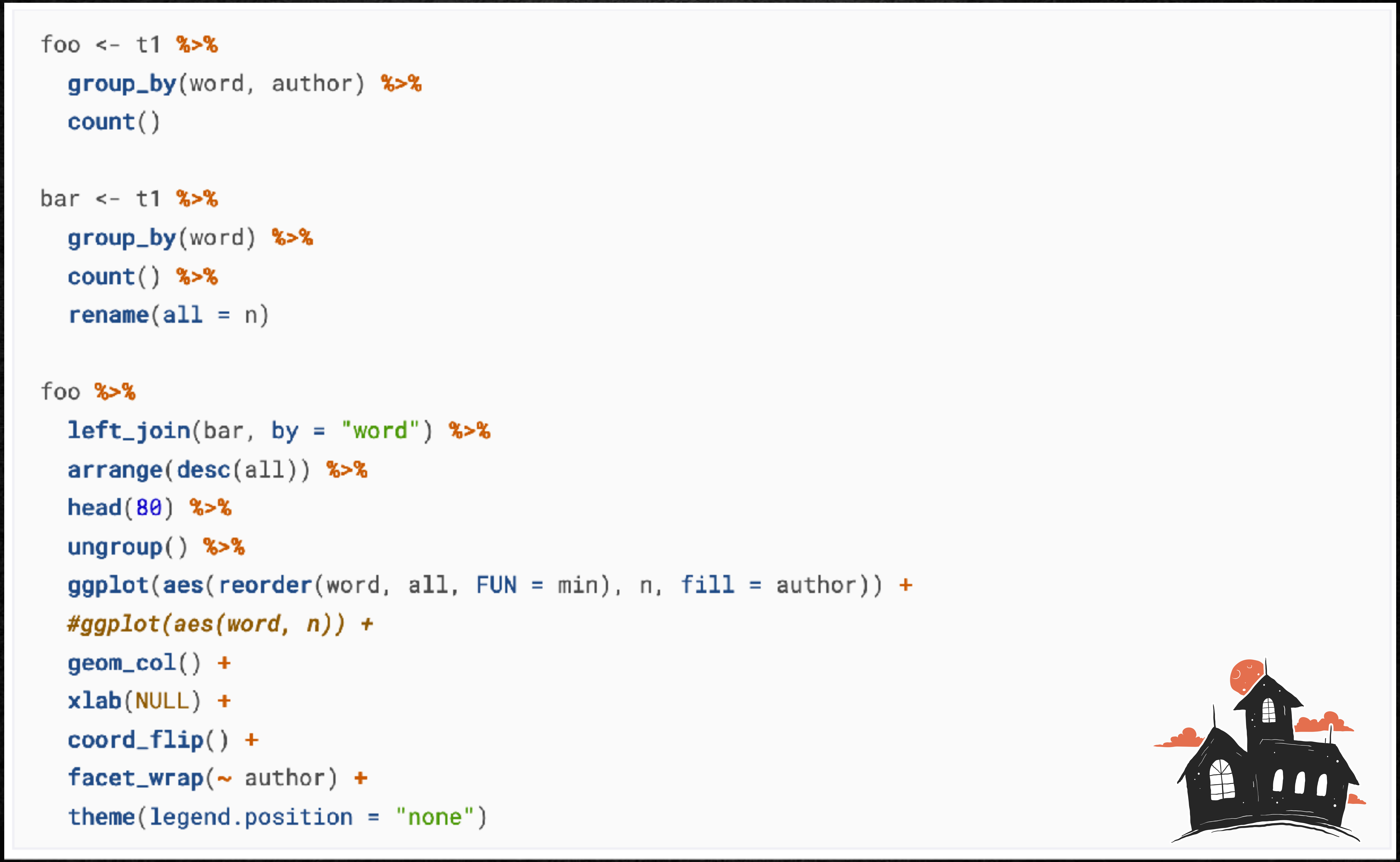
# CÓDIGOS

**Palavras:** Separar palavras / Organizar em nuvem de palavras / A nuvem mostra as palavras mais usadas de todos os autores



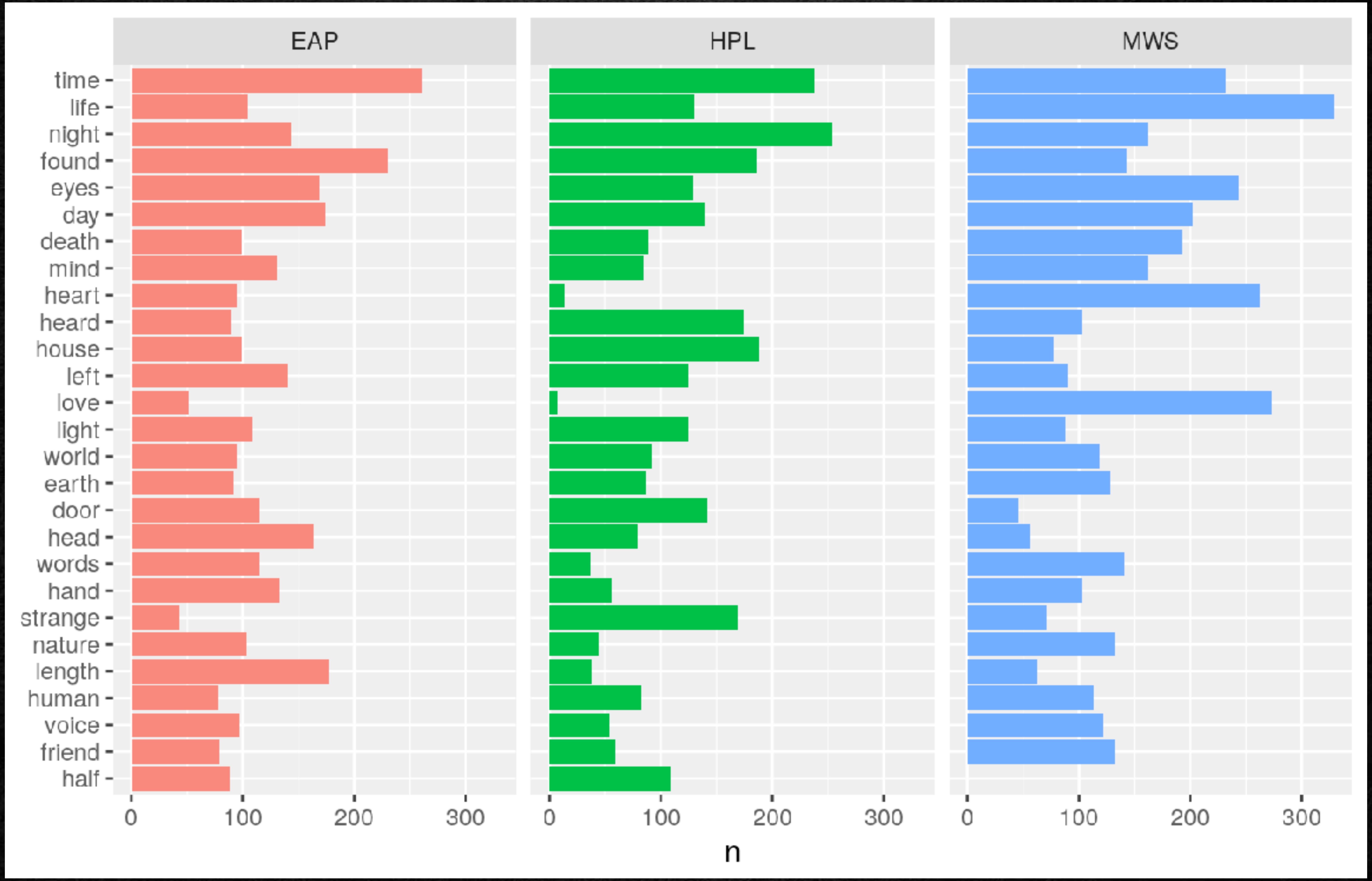
```r
t1 %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 50, color = c("purple4", "red4", "black")))
```

# PALAVRAS MAIS USADAS

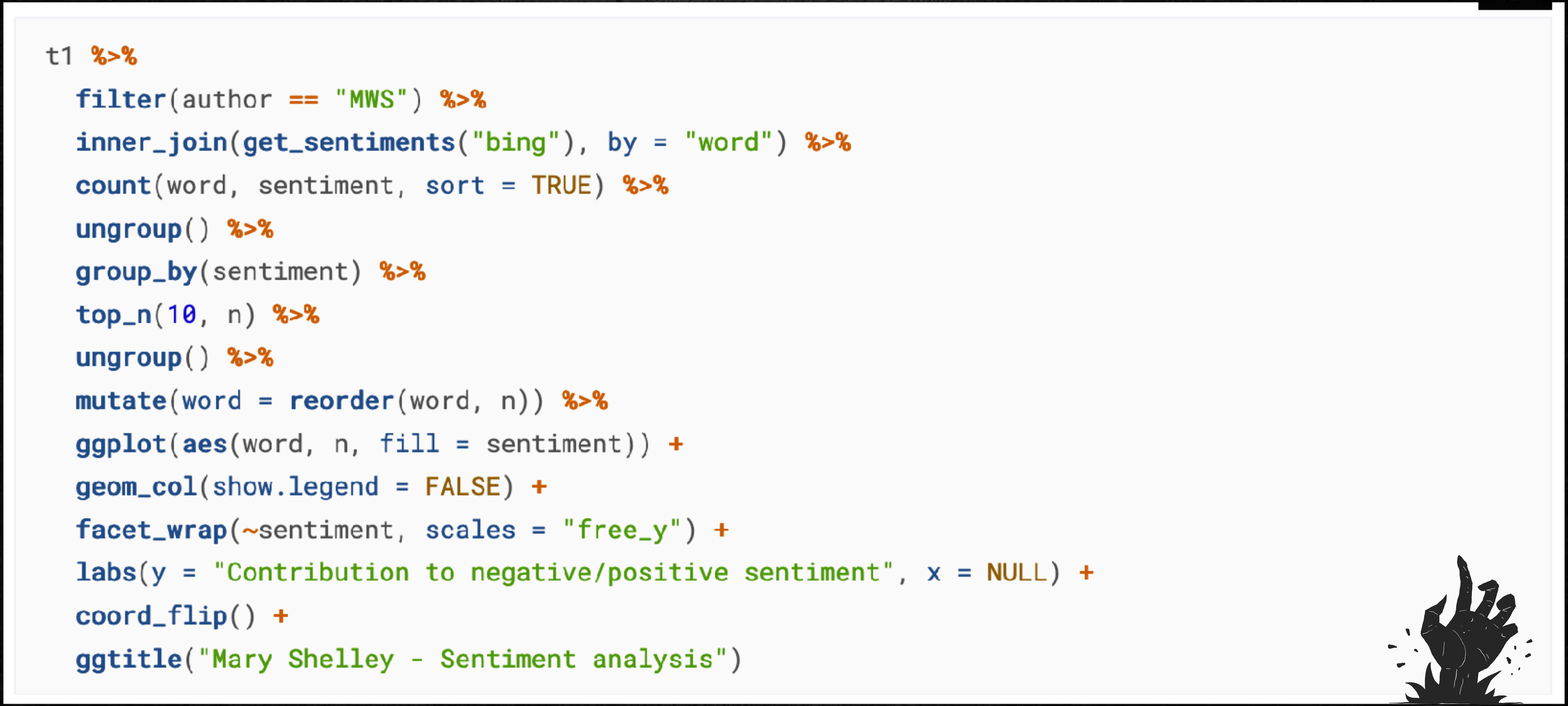


```r
foo <- t1 %>%
  group_by(word, author) %>%
  count()

bar <- t1 %>%
  group_by(word) %>%
  count() %>%
  rename(all = n)

foo %>%
  left_join(bar, by = "word") %>%
  arrange(desc(all)) %>%
  head(80) %>%
  ungroup() %>%
  ggplot(aes(reorder(word, all, FUN = min), n, fill = author)) +
  #ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip() +
  facet_wrap(~ author) +
  theme(legend.position = "none")
```

# PALAVRAS MAIS USADAS

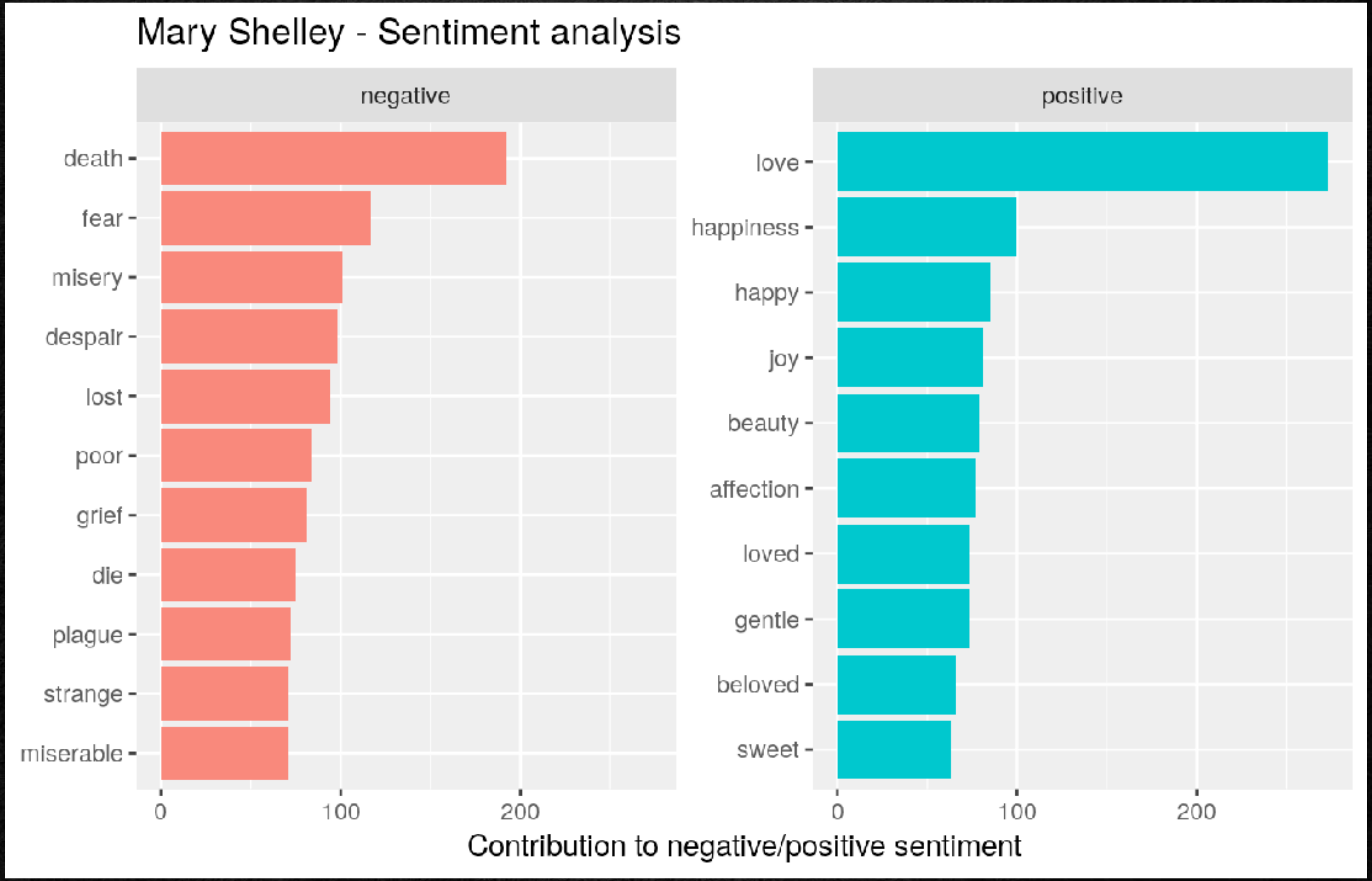


# MARY SHELLEY

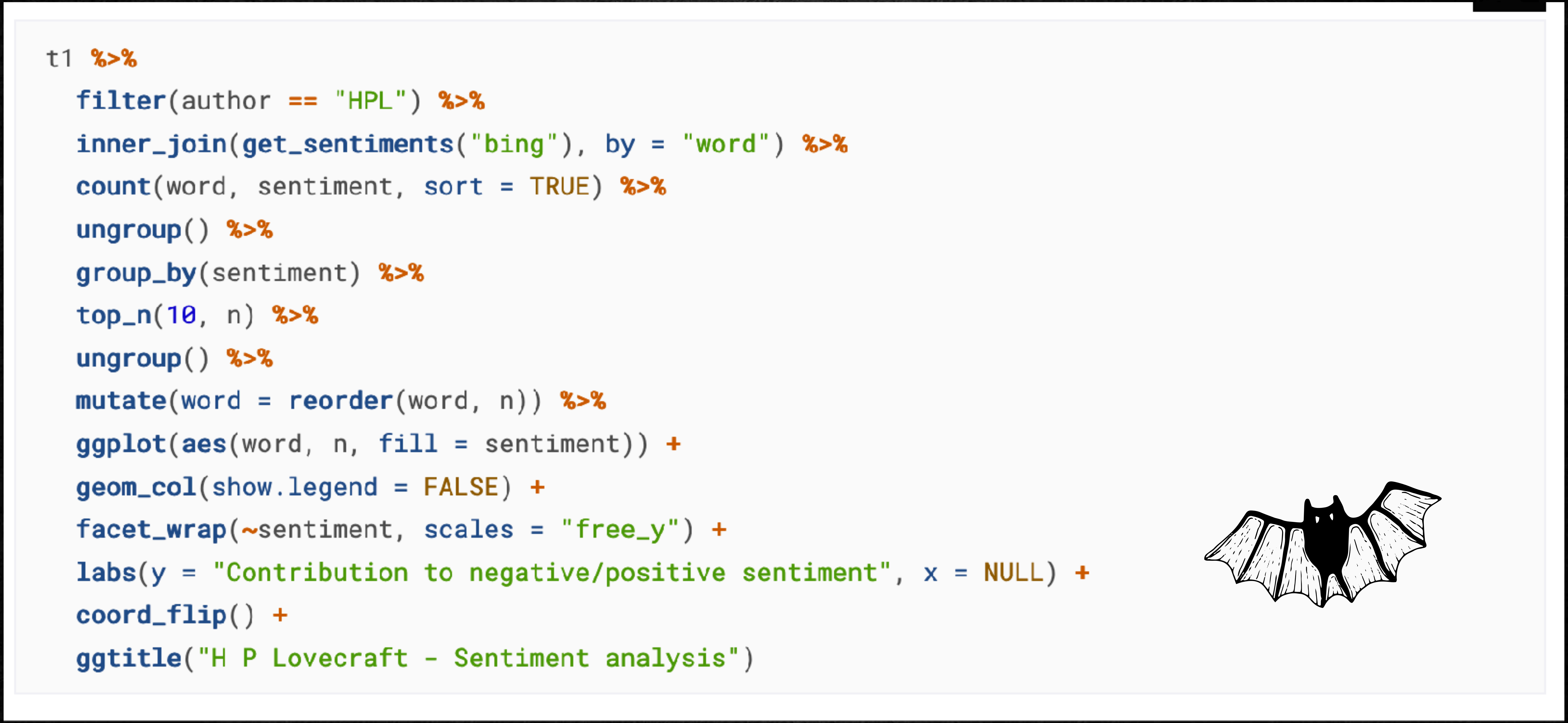


```
t1 %>%
  filter(author == "MWS") %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup() %>%
  group_by(sentiment) %>%
  top_n(10, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to negative/positive sentiment", x = NULL) +
  coord_flip() +
  ggtitle("Mary Shelley - Sentiment analysis")
```

# MARY POSITIVA

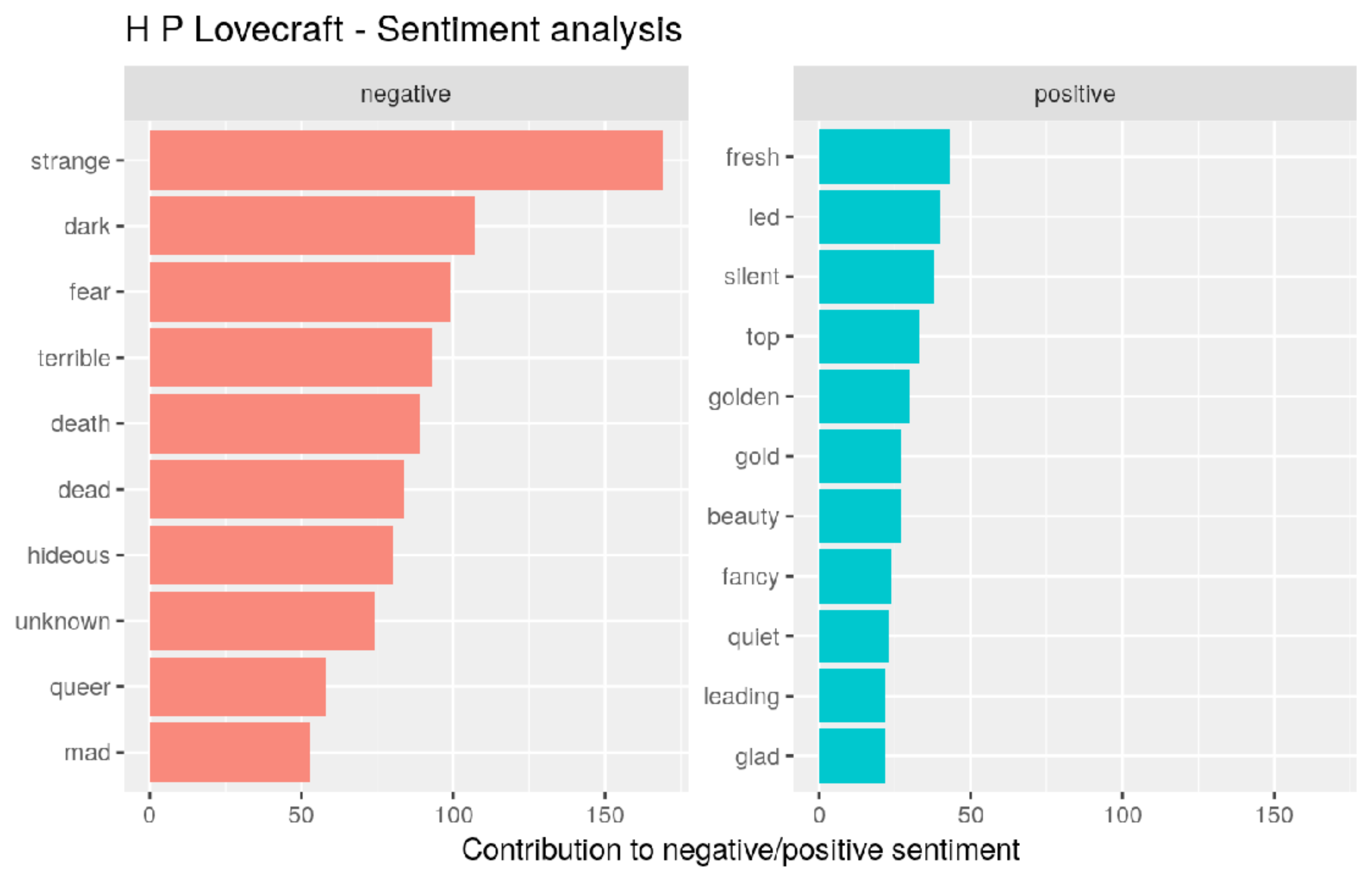


# H.P. LOVECRAFT

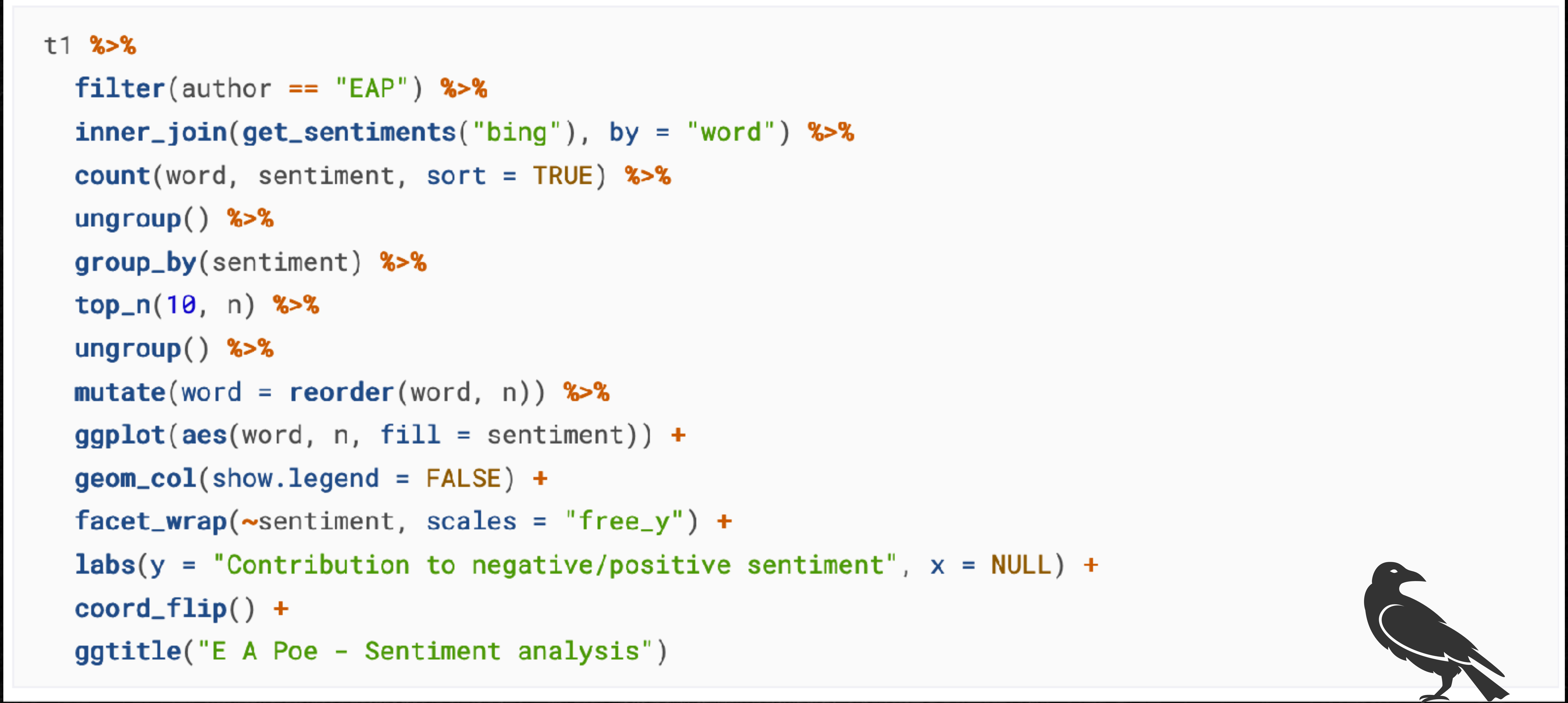


```r
t1 %>%
  filter(author == "HPL") %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup() %>%
  group_by(sentiment) %>%
  top_n(10, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to negative/positive sentiment", x = NULL) +
  coord_flip() +
  ggtitle("H P Lovecraft - Sentiment analysis")
```

# LOVECRAFT NEGATIVO

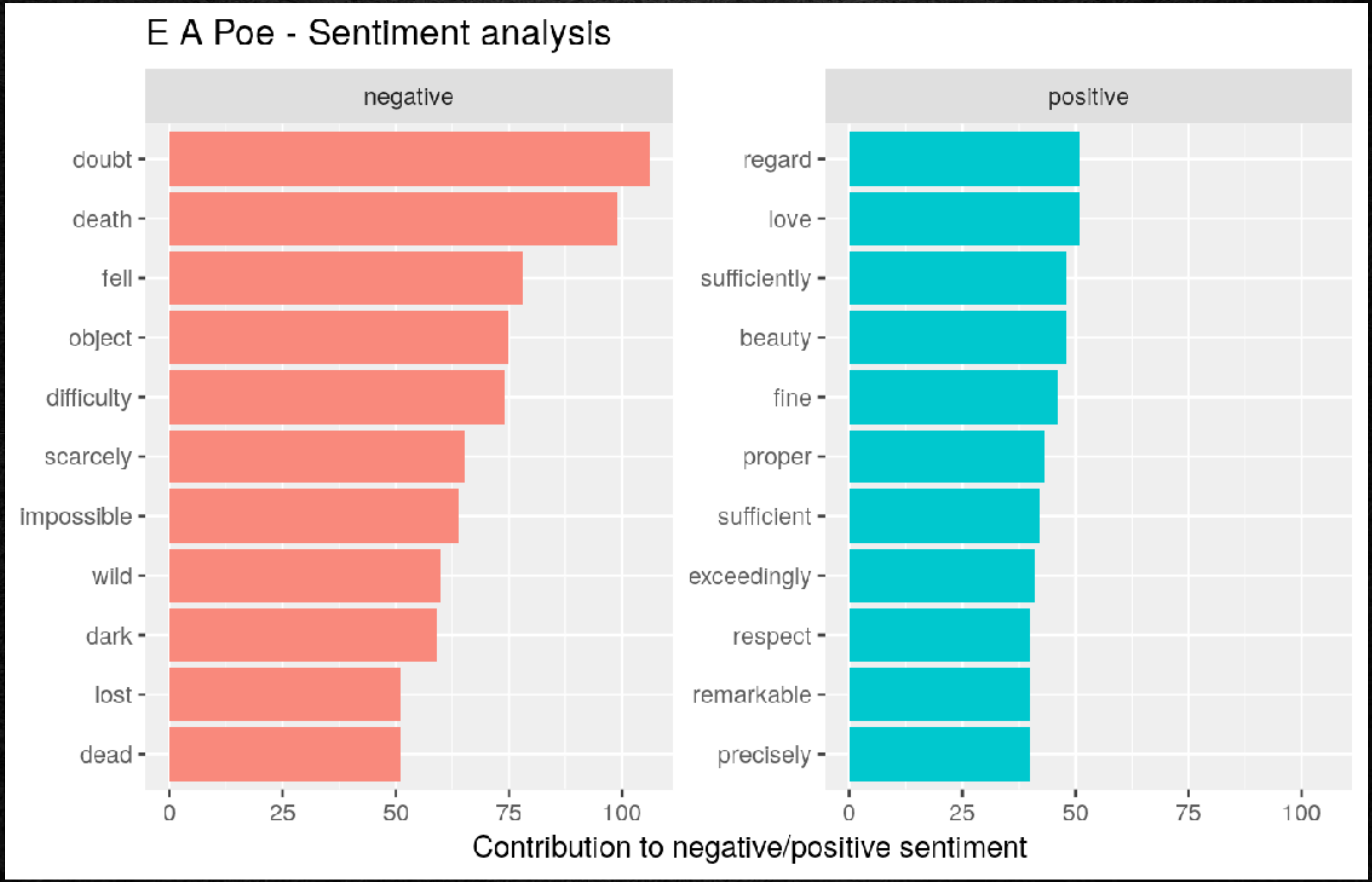


# EDGAR ALLAN POE

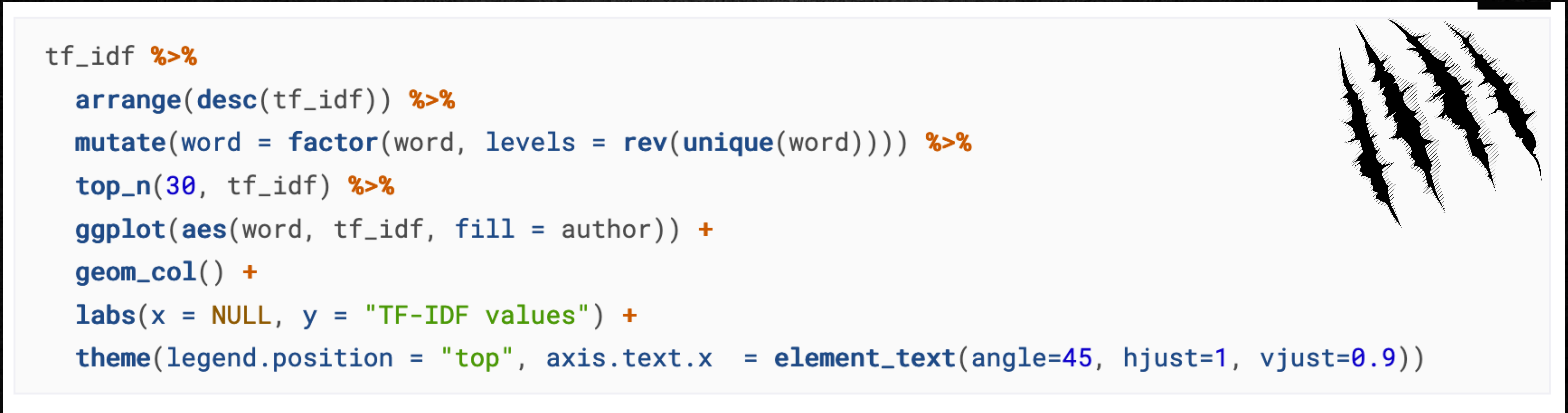


```r
t1 %>%
  filter(author == "EAP") %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup() %>%
  group_by(sentiment) %>%
  top_n(10, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to negative/positive sentiment", x = NULL) +
  coord_flip() +
  ggtitle("E A Poe - Sentiment analysis")
```

# POE EQUILIBRADO

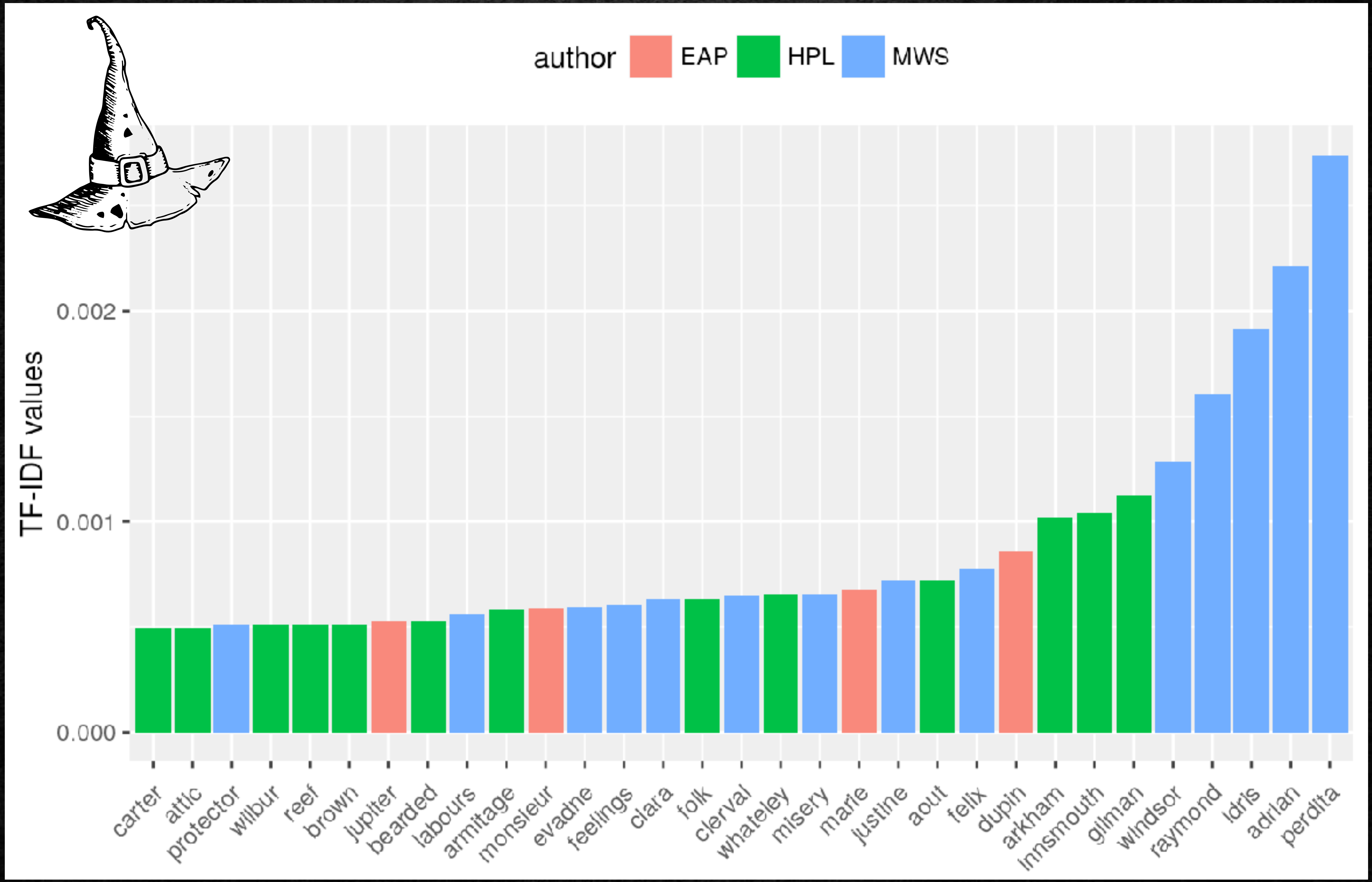


# PALAVRAS POR AUTOR



```r
tf_idf %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  top_n(30, tf_idf) %>%
  ggplot(aes(word, tf_idf, fill = author)) +
  geom_col() +
  labs(x = NULL, y = "TF-IDF values") +
  theme(legend.position = "top", axis.text.x  = element_text(angle=45, hjust=1, vjust=0.9))
```

# PALAVRAS POR AUTOR



# R X PYTHON

Davidson Mizael
Eduardo Nascimento
Nilo Andrade
Pedro Albuquerque