Exploratory Data Analysis

```
# Set CRAN mirror
options(repos = c(CRAN = "https://cloud.r-project.org"))

install.packages(c("tidyverse", "tidytext", "caret",  "tm", "textdata", "textclean", "wordcloud"))
```

```
df <- read_csv("C:/Users/Lenovo/Documents/Amazon Fine Food Reviews/Cleaned_Reviews.csv")
```

```
## Rows: 392206 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): ProductId, UserId, ProfileName, Date, Summary, Text
## dbl (4): Id, HelpfulnessNumerator, HelpfulnessDinominator, Score
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Let's View the structure of the dataset
str(df)
```

```
## spc_tbl_ [392,206 x 10] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                    : num [1:392206] 1 2 3 4 5 6 7 8 9 10 ...
##  $ ProductId             : chr [1:392206] "B001E4KFG0" "B00813GRG4" "B000LQOCH0" "B000UA0QIQ" ...
##  $ UserId                : chr [1:392206] "A3SGXH7AUHU8GW" "A1D87F6ZCVE5NK" "ABXLMWJIXXAIN" "A395BOR
##  $ ProfileName           : chr [1:392206] "delmartian" "dll pa" "Natalia Corres \"Natalia Corres\"" "
##  $ Date                  : chr [1:392206] "27-04-2011" "07-09-2012" "18-08-2008" "13-06-2011" ...
##  $ HelpfulnessNumerator  : num [1:392206] 1 0 1 3 0 0 0 0 1 0 ...
##  $ HelpfulnessDinominator: num [1:392206] 1 0 1 3 0 0 0 0 1 0 ...
##  $ Score                 : num [1:392206] 5 1 4 2 5 4 5 5 5 5 ...
##  $ Summary               : chr [1:392206] "Good Quality Dog Food" "Not as Advertised" "\"Delight\" sa
##  $ Text                  : chr [1:392206] "I have bought several of the Vitality canned dog food proc
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ProductId = col_character(),
##   ..   UserId = col_character(),
##   ..   ProfileName = col_character(),
##   ..   Date = col_character(),
##   ..   HelpfulnessNumerator = col_double(),
##   ..   HelpfulnessDinominator = col_double(),
##   ..   Score = col_double(),
##   ..   Summary = col_character(),
##   ..   Text = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
# Checking the first few rows of the dataset
head(df)
```

```
## # A tibble: 6 x 10
```

```
##      Id ProductId  UserId         ProfileName         Date  HelpfulnessNumerator
##   <dbl> <chr>      <chr>          <chr>               <chr>                 <dbl>
## 1     1 B001E4KFG0 A3SGXH7AUHU8GW "delmartian"        27-0~                     1
## 2     2 B00813GRG4 A1D87F6ZCVE5NK "dll pa"            07-0~                     0
## 3     3 B000LQOCH0 ABXLMWJIXXAIN  "Natalia Corres \"~ 18-0~                     1
## 4     4 B000UA0QIQ A395BORC6FGVXV "Karl"              13-0~                     3
## 5     5 B006K2ZZ7K A1UQRSCLF8GW1T "Michael D. Bigham~ 21-1~                     0
## 6     6 B006K2ZZ7K ADT0SRK1MGOEU  "Twoapennything"    12-0~                     0
## # i 4 more variables: HelpfulnessDinominator <dbl>, Score <dbl>, Summary <chr>,
## #   Text <chr>
```

```r
# Summary of the dataset
summary(df)
```

```
##        Id          ProductId            UserId          ProfileName
##  Min.   :     1   Length:392206      Length:392206      Length:392206
##  1st Qu.:113284   Class :character   Class :character   Class :character
##  Median :249205   Mode  :character   Mode  :character   Mode  :character
##  Mean   :261710
##  3rd Qu.:407322
##  Max.   :568454
##      Date           HelpfulnessNumerator HelpfulnessDinominator     Score
##  Length:392206      Min.   :  0.000      Min.   :  0.000        Min.   :1.000
##  Class :character   1st Qu.:  0.000      1st Qu.:  0.000        1st Qu.:4.000
##  Mode  :character   Median :  0.000      Median :  1.000        Median :5.000
##                     Mean   :  1.733      Mean   :  2.204        Mean   :4.179
##                     3rd Qu.:  2.000      3rd Qu.:  2.000        3rd Qu.:5.000
##                     Max.   :866.000      Max.   :923.000        Max.   :5.000
##    Summary             Text
##  Length:392206      Length:392206
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```r
# Checking for missing values
colSums(is.na(df))
```

```
##                      Id               ProductId                  UserId
##                       0                       0                       0
##             ProfileName                    Date   HelpfulnessNumerator
##                       3                       0                       0
## HelpfulnessDinominator                   Score                 Summary
##                       0                       0                       0
##                    Text
##                       0
```

```r
# Identying rows with missing values in ProfileName
missing_profile <- df[is.na(df$ProfileName), ]

# Let's View the rows with missing ProfileName
head(missing_profile)
```

```
## # A tibble: 3 x 10
##       Id ProductId  UserId         ProfileName Date       HelpfulnessNumerator
```

```
##    <dbl> <chr>      <chr>          <chr>       <chr>                      <dbl>
## 1 172463 B001FA1L9I AC9U70TRGPDGJ  <NA>        12-11-2010                     0
## 2 297276 B007OXJM6E A29D7XVSBCLFLD <NA>        30-12-2011                     0
## 3 515437 B004S04X4W A2H7605OSHVIQ5 <NA>        27-09-2012                     0
## # i 4 more variables: HelpfulnessDinominator <dbl>, Score <dbl>, Summary <chr>,
## #   Text <chr>
```

```
# Checking the rows where UserId is one of the three given UserIds
specific_users <- df %>%
  filter(UserId %in% c("AC9U70TRGPDGJ", "A29D7XVSBCLFLD", "A2H7605OSHVIQ5"))

# Viewing the specific rows with UserId and ProfileName
specific_users
```

```
## # A tibble: 3 x 10
##       Id ProductId  UserId         ProfileName Date       HelpfulnessNumerator
##    <dbl> <chr>      <chr>          <chr>       <chr>                      <dbl>
## 1 172463 B001FA1L9I AC9U70TRGPDGJ  <NA>        12-11-2010                     0
## 2 297276 B007OXJM6E A29D7XVSBCLFLD <NA>        30-12-2011                     0
## 3 515437 B004S04X4W A2H7605OSHVIQ5 <NA>        27-09-2012                     0
## # i 4 more variables: HelpfulnessDinominator <dbl>, Score <dbl>, Summary <chr>,
## #   Text <chr>
```

```
# Filling missing ProfileName for the specific UserIds with the most frequent ProfileName for each User
df <- df %>%
  group_by(UserId) %>%
  mutate(ProfileName = ifelse(is.na(ProfileName),
                              first(na.omit(ProfileName)),
                              ProfileName)) %>%
  ungroup()

# Those uer Id only comes once so we fill it with "Unknown"
df$ProfileName[is.na(df$ProfileName)] <- "Unknown"

# Checking if missing values are filled
colSums(is.na(df))
```

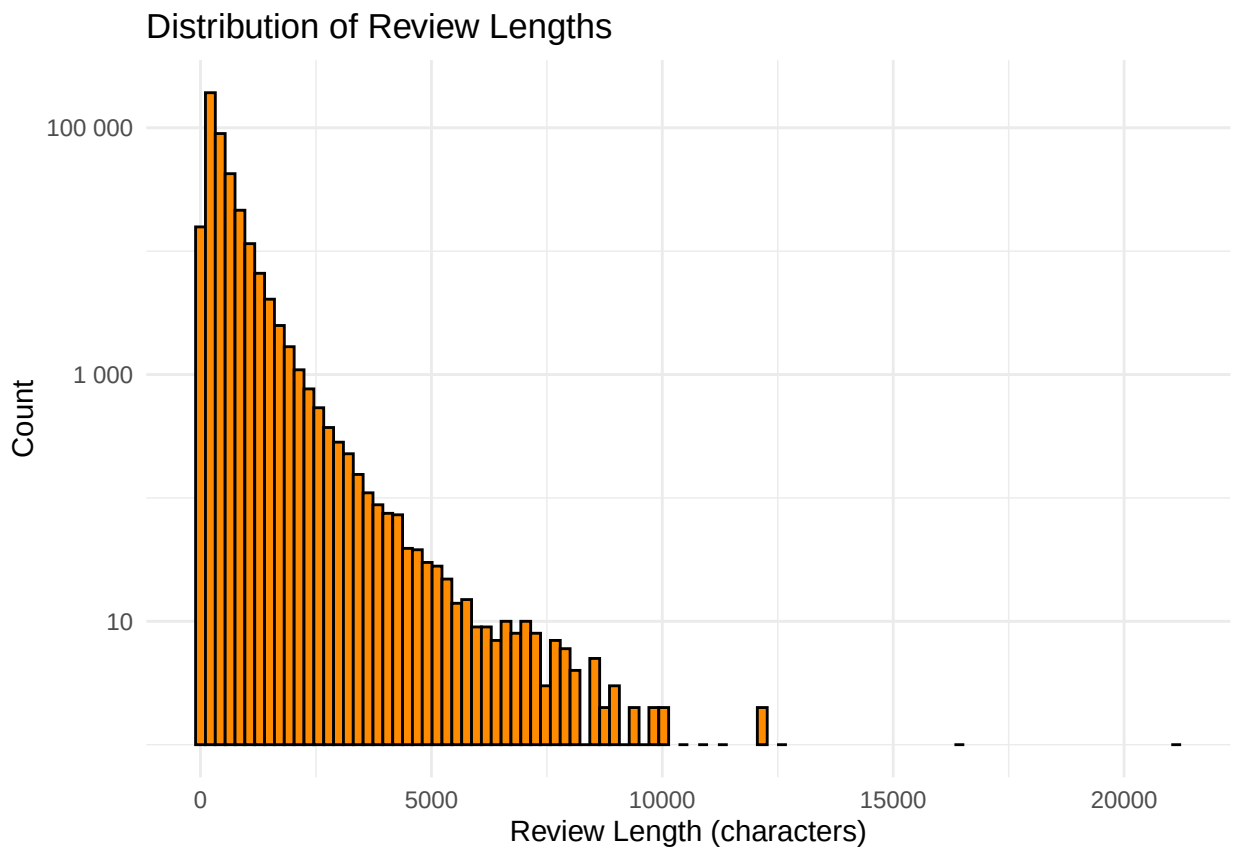
```
##                      Id               ProductId                  UserId
##                       0                       0                       0
##             ProfileName                    Date    HelpfulnessNumerator
##                       0                       0                       0
## HelpfulnessDinominator                   Score                 Summary
##                       0                       0                       0
##                    Text
##                       0
```

```
# Cleaning Text column by replacing invalid UTF-8 characters with byte codes
df$Text <- iconv(df$Text, from = "UTF-8", to = "UTF-8", sub = "byte")

# Let's Calculate the length of each review
df$review_length <- nchar(df$Text)

# Visualizing the length distribution
ggplot(df, aes(x = review_length)) +
  geom_histogram(bins = 100, fill = "darkorange", color = "black") +
  scale_y_log10(labels = label_number(scale = 1, accuracy = 1)) +
```

```
  theme_minimal() +
  labs(title = "Distribution of Review Lengths", x = "Review Length (characters)", y = "Count")
```



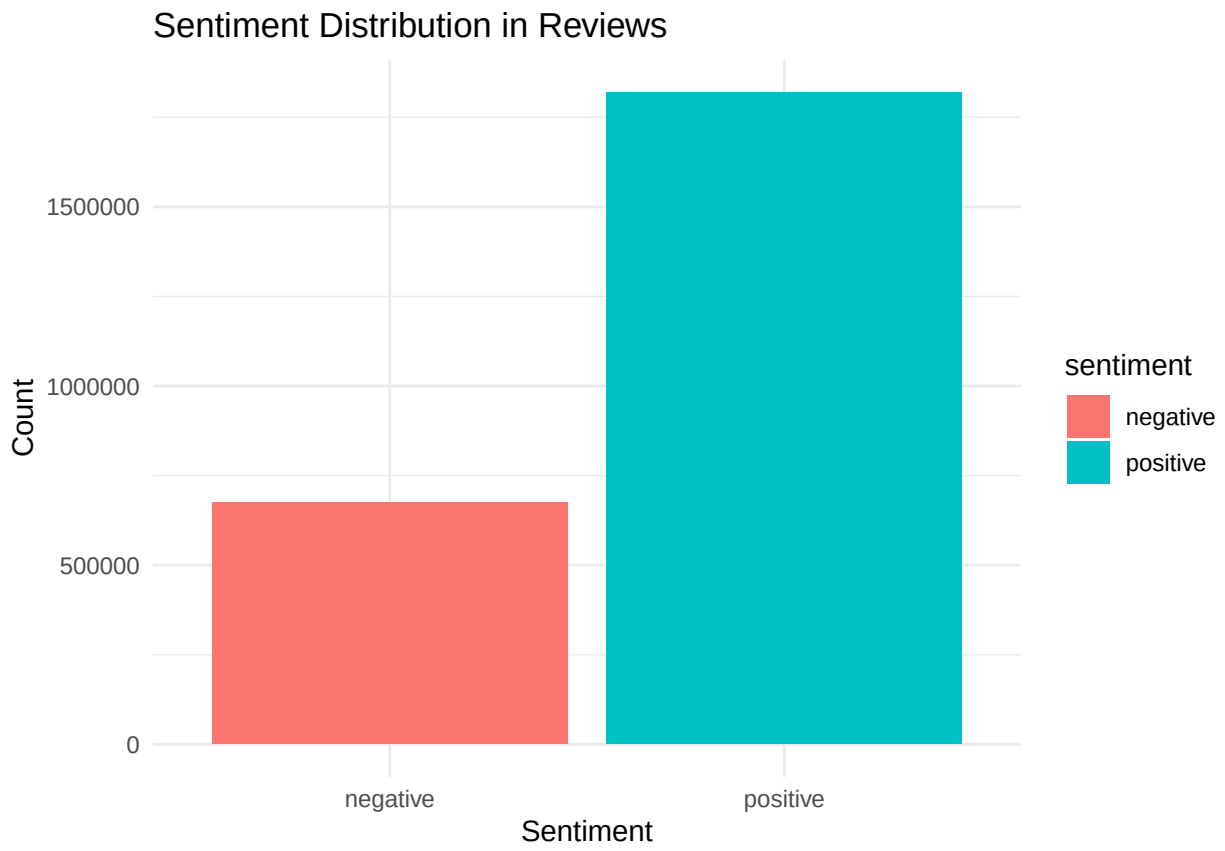

```
# Load prepossessed word frequencies
word_freq <- readRDS("word_freq.rds")

# Now let's visualize the wordcloud
wordcloud(names(word_freq), word_freq, min.freq = 100, max.words = 100, colors = brewer.pal(8, "Dark2"))
```

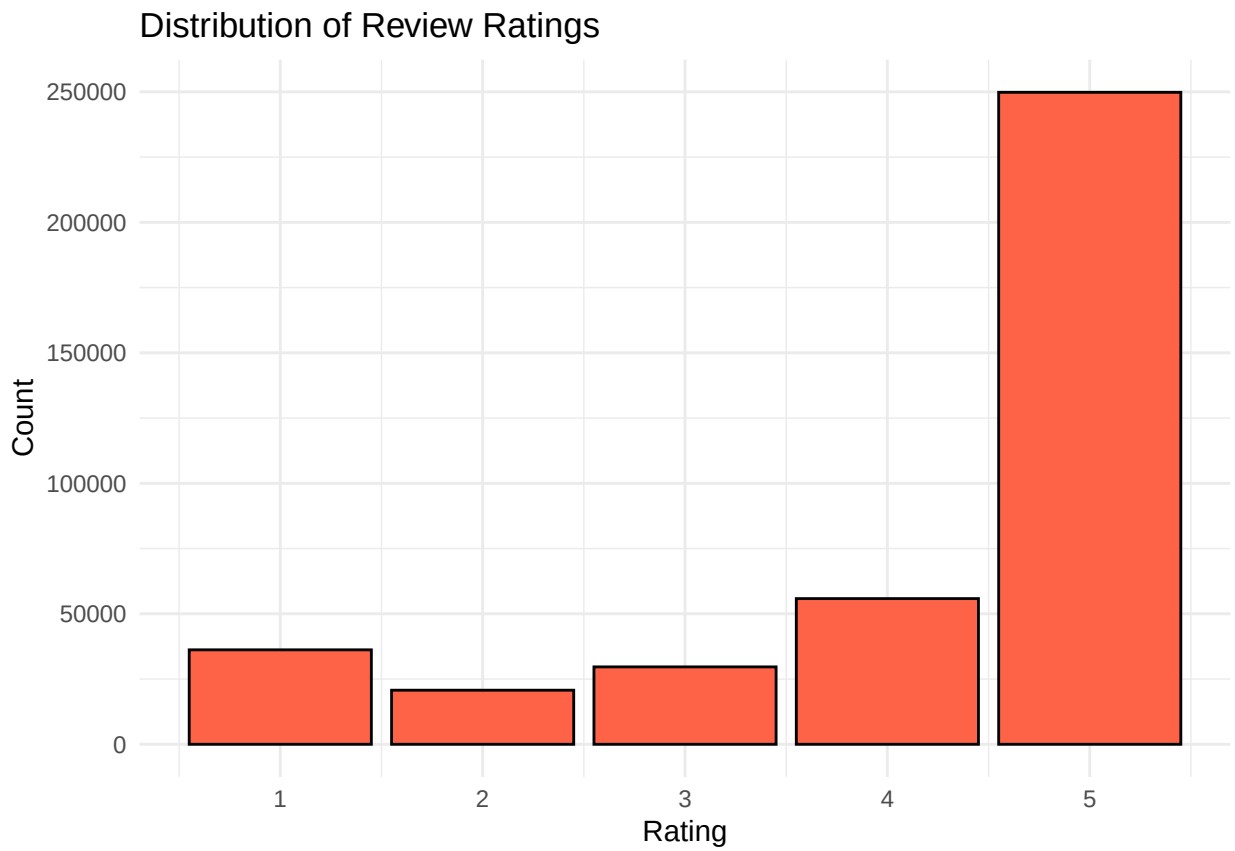

```
# Let's do Bing lexicon sentiment analysis
bing_sentiment <- df %>%
  unnest_tokens(word, Text) %>%
  inner_join(get_sentiments("bing")) %>%
  count(sentiment)
```

```
## Joining with `by = join_by(word)`
```

```
# Plotting sentiment distribution
ggplot(bing_sentiment, aes(x = sentiment, y = n, fill = sentiment)) +
  geom_bar(stat = "identity") +
  labs(title = "Sentiment Distribution in Reviews", x = "Sentiment", y = "Count") +
  theme_minimal()
```

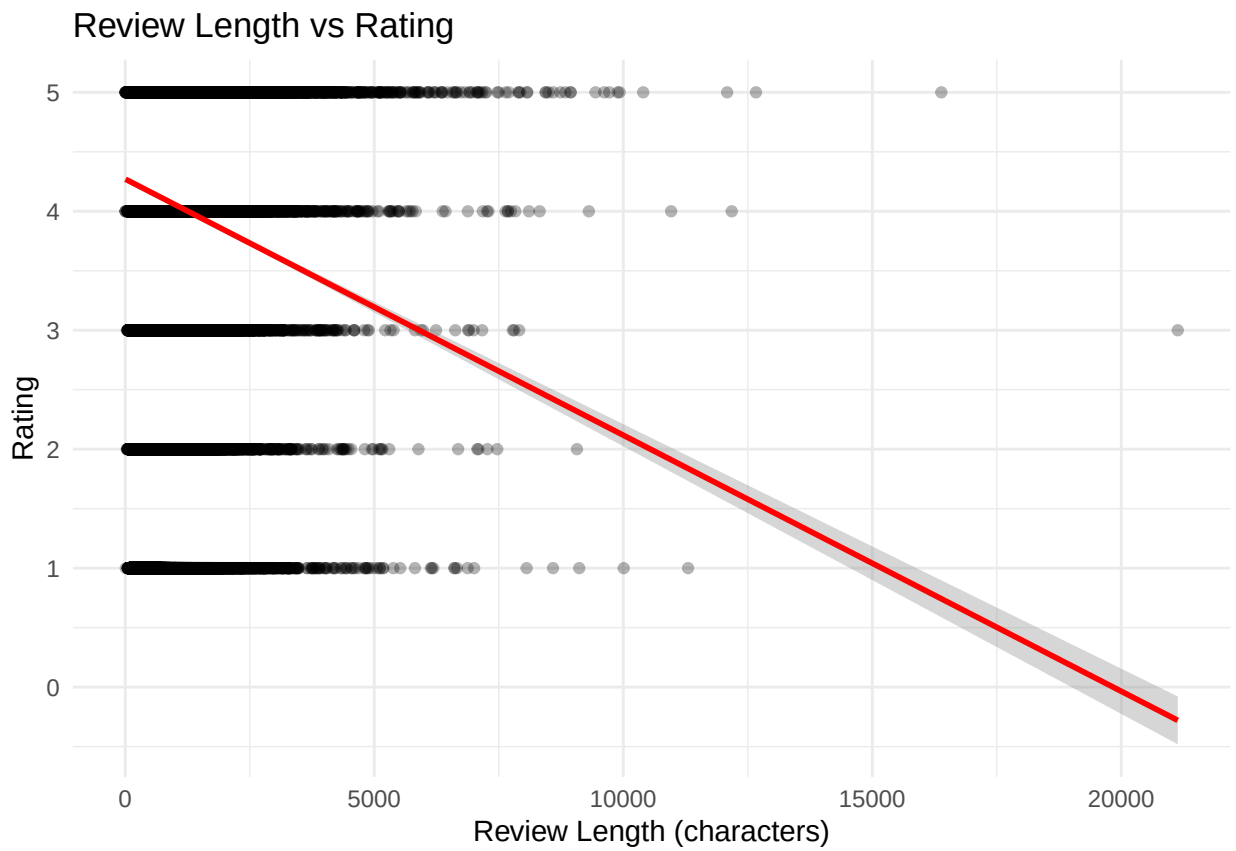



```r
# Let's visualise the Review Ratigns
ggplot(df, aes(x = Score)) +
  geom_bar(fill = "tomato", color = "black") +
  labs(title = "Distribution of Review Ratings", x = "Rating", y = "Count") +
  theme_minimal()
```



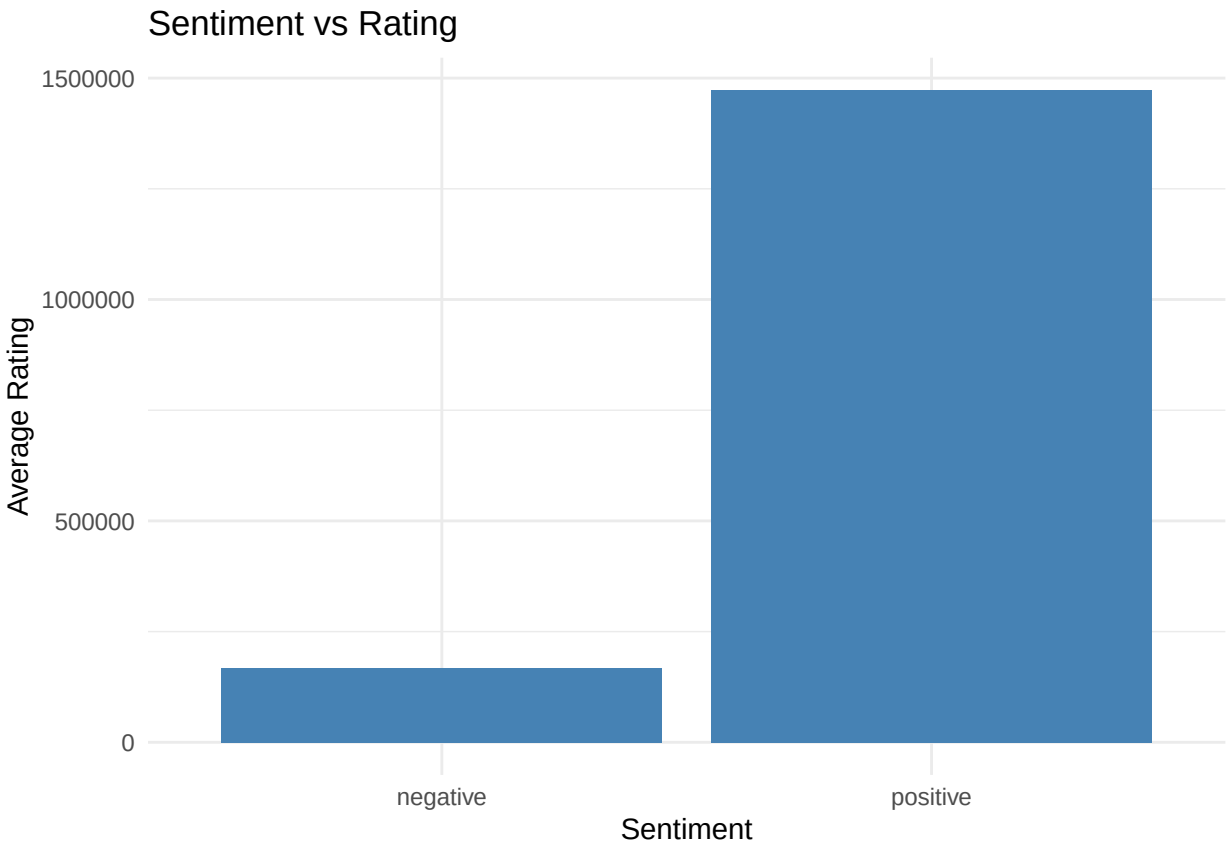

```
# Now we are going to see the ratigns with our column Score
ggplot(df, aes(x = review_length, y = Score)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Review Length vs Rating", x = "Review Length (characters)", y = "Rating") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Let's make sentiment polarity (positive/negative)
df$Sentiment <- ifelse(df$Score > 3, "positive", "negative")

ggplot(df, aes(x = Sentiment, y = Score)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Sentiment vs Rating", x = "Sentiment", y = "Average Rating") +
  theme_minimal()
```


Sentiment vs Rating
1500000
1000000
500000
0
Average Rating
negative
positive
Sentiment