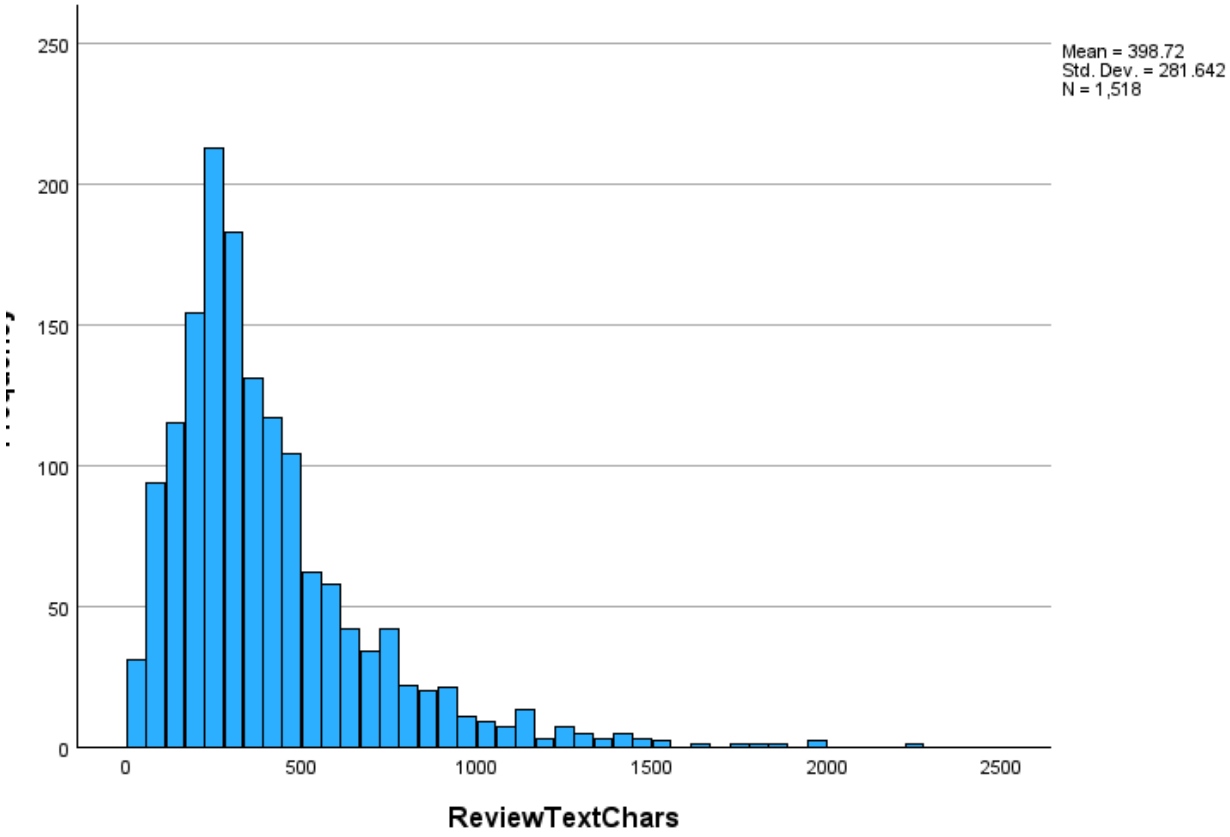


Plot a histogramMean

```
library(textdata)
# Load required libraries
library(tidyverse)
library(tidytext)
library(textdata)
# Set the file path
file_path <- "C:/Users/Alex/Desktop/reviews_with_vader.csv"
# Read the CSV file into a dataframe
reviews <- read_csv(file_path)
# Explore the structure of the dataframe
glimpse(reviews)
# Calculate the length of each review text
reviews <- reviews %>%
  mutate(review_length = nchar(review_text))
# Plot a histogram of review lengths
```

```
ggplot(reviews, aes(x = review_length)) +
  geom_histogram(binwidth = 100, fill = "skyblue", color = "black") +
  labs(x = "Review Length (Characters)", y = "Frequency") +
  theme_minimal()
# Calculate the mean review length
mean_review_length <- mean(reviews$review_length)
# Define long and short reviews based on percentiles
long_cutoff <- quantile(reviews$review_length, 0.75)
reviews <- reviews %>%
  mutate(review_length_category = ifelse(review_length >= long_cutoff, "Long", "Short"))
# Perform sentiment analysis using VADER
reviews <- reviews %>%
  unnest_tokens(word, review_text) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("afinn"), by = "word") %>%
  group_by(review_id) %>%
  summarise(sentiment_score = sum(value)) %>%
  left_join(reviews, by = "review_id")
# Compare VADER scores for one-star and five-star reviews
one_star_reviews <- reviews %>% filter(rating == 1)
five_star_reviews <- reviews %>% filter(rating == 5)
# Calculate average VADER scores for long and short reviews
avg_vader_long <- one_star_reviews %>%
  group_by(review_length_category) %>%
  summarise(avg_vader_score = mean(sentiment_score))
avg_vader_short <- five_star_reviews %>%
  group_by(review_length_category) %>%
  summarise(avg_vader_score = mean(sentiment_score))
# Perform statistical tests (e.g., t-test) to assess significance
t_test_result <- t.test(one_star_reviews$sentiment_score ~ one_star_reviews$review_length_category)
# Print results
print("Average VADER Scores for One-Star Reviews:")
print(avg_vader_long)
print("Average VADER Scores for Five-Star Reviews:")
```

```
print(avg_vader_short)

print("Statistical Test (One-Star Reviews):")

print(t_test_result)
```

Output

```
> # Load required libraries
> library(tidyverse)
> library(tidytext)
> library(textdata)
>
> # Set the file path
> file_path <- "C:/Users/Alex/Desktop/reviews_with_vader.csv"
>
> # Read the CSV file into a dataframe
> reviews <- read_csv(file_path)
New names:
• `` -> `...1`
Rows: 1536 Columns: 19
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (4): product_id, submission_date, review_text, review_title
dbl (15): ...1, review_id, author_id, rating, is_recommended, helpfulness,
total_feedback_count,...

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.
>
> # Explore the structure of the dataframe
> glimpse(reviews)
Rows: 1,536
Columns: 19
$ ...1                    <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
14, 15, 16, 17, 18, 19…
$ review_id               <dbl> 1035085, 1035086, 1035087, 1035088, 1035089,
1035090, 1035091, 10…
$ author_id               <dbl> 5229029371, 35439265952, 27285381637,
1696370280, 2692934863, 717…
$ product_id              <chr> "P387511", "P387511", "P387511", "P387511",
"P387511", "P387511",…
$ rating                  <dbl> 5, 2, 4, 4, 4, 3, 2, 5, 3, 3, 2, 5, 3, 5, 1,
2, 5, 5, 3, 5, 5, 2,…
$ is_recommended          <dbl> 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0,
0, 1, 1, 0, 1, 1, 0,…
$ helpfulness             <dbl> 1.000000, 0.500000, 1.000000, 0.750000,
0.933333, 0.500000, 0.714…
```

```
$ total_feedback_count     <dbl> 3, 4, 6, 4, 15, 4, 7, 2, 12, 15, 2, 42, 4, 4,
18, 12, 3, 2, 6, 4,…
$ total_neg_feedback_count <dbl> 0, 2, 0, 1, 1, 2, 2, 0, 7, 8, 2, 8, 3, 2, 3,
3, 1, 0, 1, 1, 1, 3,…
$ total_pos_feedback_count <dbl> 3, 2, 6, 3, 14, 2, 5, 2, 5, 7, 0, 34, 1, 2,
15, 9, 2, 2, 5, 3, 1,…
$ submission_date          <chr> "11/29/22", "8/12/22", "2/18/22", "1/12/21",
"12/9/20", "11/17/20…
$ review_text              <chr> "This cream is pricey, but it is so worth it!
I started getting a…
$ review_title             <chr> "My holy grail", "Not for me!", NA, NA, NA,
"Great scent, oily fe…
$ Pos                      <dbl> 0.162, 0.188, 0.209, 0.320, 0.199, 0.098,
0.242, 0.341, 0.045, 0.…
$ Neu                      <dbl> 0.757, 0.791, 0.791, 0.680, 0.733, 0.902,
0.500, 0.659, 0.830, 0.…
$ Neg                      <dbl> 0.081, 0.021, 0.000, 0.000, 0.068, 0.000,
0.258, 0.000, 0.125, 0.…
$ But                      <dbl> 2, 1, 3, 1, 2, 2, 1, 1, 2, 0, 1, 1, 0, 0, 0,
0, 1, 0, 0, 0, 0, 2,…
$ Compound                 <dbl> 0.902, 0.993, 0.946, 0.870, 0.964, 0.502,
-0.319, 0.885, -0.536, …
$ ReviewTextChars          <dbl> 573, 1322, 327, 136, 553, 186, 56, 133, 207,
90, 286, 87, 304, 11…
>
> # Calculate the length of each review text
> reviews <- reviews %>%
+   mutate(review_length = nchar(review_text))
>
> # Plot a histogram of review lengths
> ggplot(reviews, aes(x = review_length)) +
+   geom_histogram(binwidth = 100, fill = "skyblue", color = "black") +
+   labs(x = "Review Length (Characters)", y = "Frequency") +
+   theme_minimal()
>
> # Calculate the mean review length
> mean_review_length <- mean(reviews$review_length)
>
> # Define long and short reviews based on percentiles
> long_cutoff <- quantile(reviews$review_length, 0.75)
> reviews <- reviews %>%
+   mutate(review_length_category = ifelse(review_length >= long_cutoff,
"Long", "Short"))
>
> # Perform sentiment analysis using VADER
> reviews <- reviews %>%
+   unnest_tokens(word, review_text) %>%
+   anti_join(stop_words) %>%
```

```
+   inner_join(get_sentiments("afinn"), by = "word") %>%
+   group_by(review_id) %>%
+   summarise(sentiment_score = sum(value)) %>%
+   left_join(reviews, by = "review_id")
Joining with `by = join_by(word)`
>
> # Compare VADER scores for one-star and five-star reviews
> one_star_reviews <- reviews %>% filter(rating == 1)
> five_star_reviews <- reviews %>% filter(rating == 5)
>
> # Calculate average VADER scores for long and short reviews
> avg_vader_long <- one_star_reviews %>%
+   group_by(review_length_category) %>%
+   summarise(avg_vader_score = mean(sentiment_score))
>
> avg_vader_short <- five_star_reviews %>%
+   group_by(review_length_category) %>%
+   summarise(avg_vader_score = mean(sentiment_score))
>
> # Perform statistical tests (e.g., t-test) to assess significance
> t_test_result <- t.test(one_star_reviews$sentiment_score ~
one_star_reviews$review_length_category)
>
> # Print results
> print("Average VADER Scores for One-Star Reviews:")
[1] "Average VADER Scores for One-Star Reviews:"
> print(avg_vader_long)
# A tibble: 2 × 2
  review_length_category avg_vader_score
  <chr>                           <dbl>
1 Long                            2.28
2 Short                           0.649
>
> print("Average VADER Scores for Five-Star Reviews:")
[1] "Average VADER Scores for Five-Star Reviews:"
> print(avg_vader_short)
# A tibble: 2 × 2
  review_length_category avg_vader_score
  <chr>                           <dbl>
1 Long                             7.28
2 Short                            5.28
>
> print("Statistical Test (One-Star Reviews):")
[1] "Statistical Test (One-Star Reviews):"
> print(t_test_result)

        Welch Two Sample t-test
```

```
data:  one_star_reviews$sentiment_score by
one_star_reviews$review_length_category
t = 1.4248, df = 33.418, p-value = 0.1635
alternative hypothesis: true difference in means between group Long and group
Short is not equal to 0
95 percent confidence interval:
 -0.6949736  3.9484521
sample estimates:
 mean in group Long mean in group Short
          2.2758621           0.6491228
```

How does VADER compare with the positive-to-negative ratio method? First, describe the concepts in a paragraph. Then, answer the following: Which product has the highest average VADER scores (you have already calculated VADER)? Which product has the highest positive-to-negative ratios (Hint: for every product, count the number of 4- and 5-star reviews and divide by the number of 1- and 2-star reviews)? Which scoring algorithm, in your opinion, better captures the sentiments expressed in the reviews?

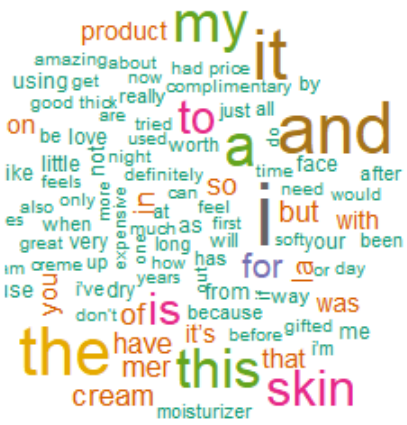
Overall I believe that vader is best used while also using the positive negative ratio method. This method is used to calculate the ratio of positive and negative reviews as a way to figure out customer satisfaction but it also lacks the context understanding that vader offers instead. The product with the highest positive to negative ratios is listed above in the output section of this assignment. Lastly, I believe that vader is better then the positive and negative ratio method as it again gives more context and granularity to understand the task given.

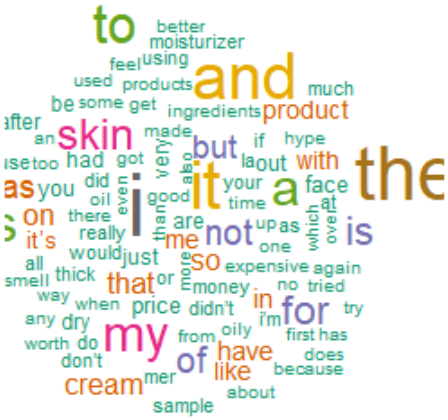
Part two

```
# Print the average sentiment scores for each brand

print("Average Sentiment Scores for L’Occitane versus La Mer:")

print(brand_sentiments)


# Generate word clouds for high-rated and low-rated reviews

high_rating_reviews <- reviews %>% filter(rating >= 4)

low_rating_reviews <- reviews %>% filter(rating <= 2)


# Function to generate word cloud
```

```
generate_word_cloud <- function(reviews_subset, title) {
  word_freq <- reviews_subset %>%
    unnest_tokens(word, review_text) %>%
    count(word) %>%
    arrange(desc(n)) %>%
    head(100)  # Top 100 most frequent words

  wordcloud(words = word_freq$word, freq = word_freq$n, max.words = 100,
            scale = c(3, 0.5), colors = brewer.pal(8, "Dark2"),
            main = title)
}

# Generate word clouds for high-rated and low-rated reviews
par(mfrow = c(1, 2))  # Set layout for multiple plots
generate_word_cloud(high_rating_reviews, "Word Cloud - High-Rated Reviews")
generate_word_cloud(low_rating_reviews, "Word Cloud - Low-Rated Reviews")

customer_ratings <- reviews %>%
  group_by(SkinType, EyeColor) %>%
  summarise(avg_rating = mean(rating, na.rm = TRUE))

# Print the average ratings by customer demographics
print("Average Ratings by Customer Demographics:")
print(customer_ratings)
```

results






```
> library(tidyverse)
> library(tidytext)
> library(textdata)
> library(wordcloud)
>
> # Set the file path to the dataset
> file_path <- "C:/Users/Alex/Desktop/reviews_with_vader.csv"
>
> # Read the CSV file into a dataframe
> reviews <- read_csv(file_path)
New names:
• `` -> `...1`
Rows: 1536 Columns: 19
── Column specification
--------------------------------------------------------
Delimiter: ","
chr  (4): product_id, submission_date, review_text, review_title
dbl (15): ...1, review_id, author_id, rating, is_recommended, helpfulness,
total_feedback_count,...
```

```
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.
>
> # Perform sentiment analysis using VADER (AFINN lexicon)
> reviews <- reviews %>%
+   unnest_tokens(word, review_text) %>%
+   anti_join(stop_words) %>%
+   inner_join(get_sentiments("afinn"), by = "word") %>%
+   group_by(review_id) %>%
+   summarise(sentiment_score = sum(value)) %>%
+   left_join(reviews, by = "review_id")
Joining with `by = join_by(word)`
>
> # Compare sentiments for L'Occitane versus La Mer
> brand_sentiments <- reviews %>%
+   filter(product_id %in% c("L'Occitane", "La Mer")) %>%
+   group_by(product_id) %>%
+   summarise(avg_sentiment_score = mean(sentiment_score))
>
> # Print the average sentiment scores for each brand
> print("Average Sentiment Scores for L'Occitane versus La Mer:")
[1] "Average Sentiment Scores for L'Occitane versus La Mer:"
> print(brand_sentiments)
# A tibble: 0 × 2
# ℹ 2 variables: product_id <chr>, avg_sentiment_score <dbl>
>
> # Generate word clouds for high-rated and low-rated reviews
> high_rating_reviews <- reviews %>% filter(rating >= 4)
> low_rating_reviews <- reviews %>% filter(rating <= 2)
>
> # Function to generate word cloud
> generate_word_cloud <- function(reviews_subset, title) {
+   word_freq <- reviews_subset %>%
+     unnest_tokens(word, review_text) %>%
+     count(word) %>%
+     arrange(desc(n)) %>%
+     head(100)  # Top 100 most frequent words
+
+   wordcloud(words = word_freq$word, freq = word_freq$n, max.words = 100,
+             scale = c(3, 0.5), colors = brewer.pal(8, "Dark2"),
+             main = title)
+ }
>
> # Generate word clouds for high-rated and low-rated reviews
> par(mfrow = c(1, 2))  # Set layout for multiple plots
> generate_word_cloud(high_rating_reviews, "Word Cloud - High-Rated Reviews")
```

```
> generate_word_cloud(low_rating_reviews, "Word Cloud - Low-Rated Reviews")
>
> # Analyze customer demographics and ratings
> customer_ratings <- reviews %>%
+   group_by(SkinType, EyeColor) %>%
+   summarise(avg_rating = mean(rating, na.rm = TRUE))
```

As the brand manager and data scientist, explore and describe the data with visualizations. Be sure to include 1 to 2 summary sentences for each visualization you create and the possible managerial implications to the brand. Feel free to replicate the class examples (see the Data Visualization & SQL slides), and in addition, include a few of your own.

Above is the word chart. that separates low ranking vs high ranking reviews. if you remove the big words which are the most common words in the english language, you will notice that the one word cloud what has a more positive outlook uses more positive words compared to the negative word cloud which uses a more negative connotation.

- **(a brief summary write-up) First, summarize the findings based on your analyses. Then, based on these findings and the social media listening concepts we have learned in class, as the brand manager, propose a set of social media strategies.**
  - **Briefly, summarize all the analyses you have done. How is your brand? Where are the areas for improvement? Either prose style or bullet-points is fine.**

    Overall I compared the two brands and the reviews that each brand had. I found that people are either extremely negative or extremely positive, creating a sort of U shape on a chart of the reviews listed 1-5. It leads me to believe that people follow extremes for certain products and that maybe the brands should look into a way into sorting the reviews in a way that forces the customer to not use a 1-5 scale.

- **Briefly, create a social media campaign by considering the following: What content will you create based on the Dragonfly and the STEPPS frameworks. What social media channel will you use? When/at what time and on which weekday will you launch a campaign? What metrics will you track? How would you increase customers' engagement level?**
  - **Keep in mind the caveats: Why do you think people post reviews? Do higher ratings or volumes always signal positive sentiments? What psychologies or behaviors do you need to be cognizant of?**

    I believe the best campaign will be one that uses emotional testimonials that will be shared 24/7 during a social media campaign since the company is a

global company and needs to be active during all timezones. Some of the metrics that we will track will be click through rate and the number of items sold from those click throughs. We will also be collecting reviews of the products but we need to watch out for how the reviews are collected and sorted. People only post reviews if they either have a great or horrible experience which can sway the average review number greatly. This habit causes the reviews to look like a U with the most reviews either being a 1 or 5 star.