Unlocking Disney's World: Text Mining Insights

Navigating Consumer Sentiment Analysis, Tokenization and TD-IDF in Disneyland Reviews from Scratch

Photo by PAN XIAOZHEN on Unsplash

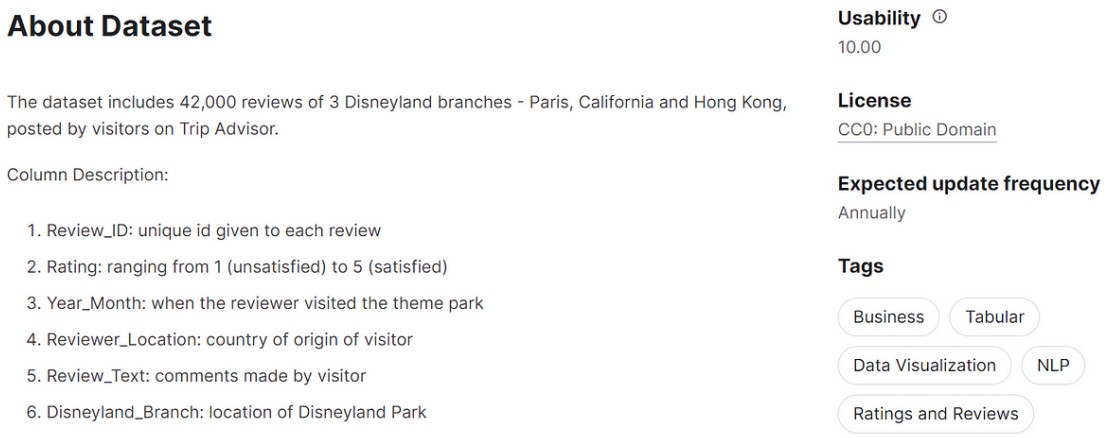
In today's digital age, public opinion intricately weaves through the vast web, offering a unique window into consumer experiences across various services and destinations. How can we analyze all this information and uncover the valuable insights that lie within?

We will explore a simple text mining project focused on the globally renowned Disney Parks. To achieve this, we'll harness the power of the **R language** within **R Studio**, mastering the creation of our very own text mining process and the analysis of unstructured categorical data.

The dataset we are going to work with it was obtained from Kaggle (url: https://www.kaggle.com/datasets/arushchillar/disneyland-reviews).

Acknowledging the dataset creator is essential.

Here is a screenshot from Kaggle with the different specifications of the dataset provided:



Screenshot from Kaggle | https://www.kaggle.com/datasets/arushchillar/disneyland-reviews

Let's get started!

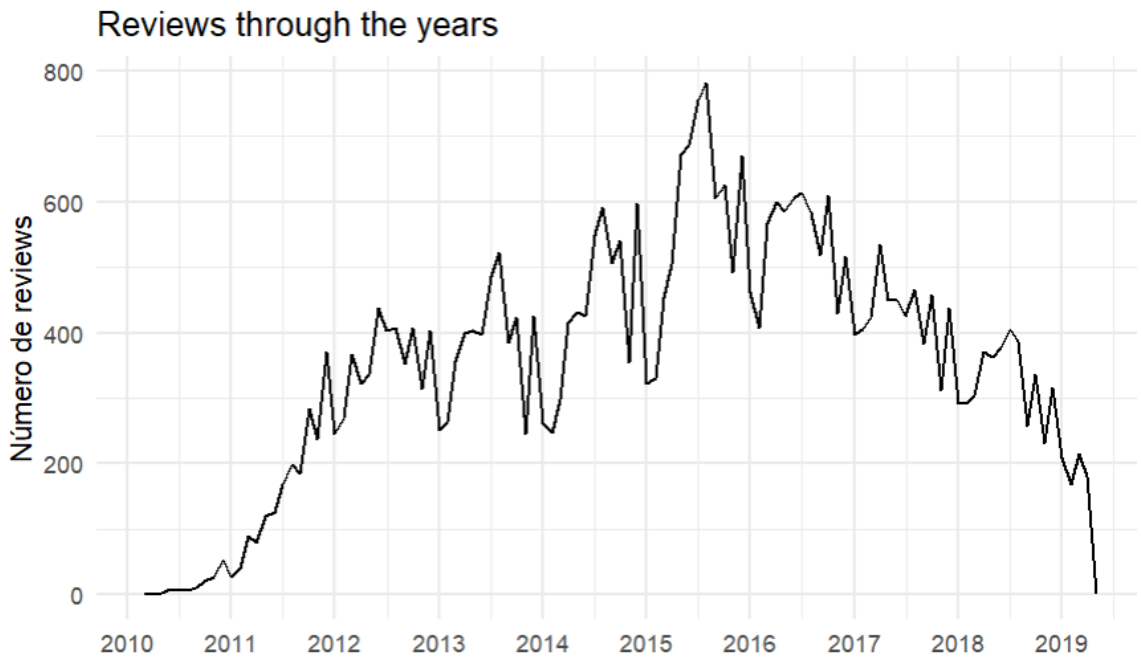
```
# DEPENDENCIES -----------------------------------------------------------
library(ggplot2)
library(tidyverse)
library(tidytext)
library(wordcloud)
library(dplyr)
library(textdata)


# DATA LOAD --------------------------------------------------------------

disneyland <- read_csv("disneyland.csv")
head(disneyland)
```

First of all we start by creating our DEPENCIES section with the libraries we are going to use. We're loading essential tools like ggplot2 for visualization and tidyverse for data manipulation.

Then, we import our **dataset**, named "disneyland.csv", and store it in a variable called "disneyland". Using the head() function, we quickly preview the first few rows of the dataset. This initial step sets the stage for further exploration and analysis of Disneyland-related data.

We can now start our project with an exploratory analysis. Our initial step involves plotting the available data.

```
disneyland %>%
  count(year_month) %>%
  ggplot(aes(x = year_month, y = n)) +
  geom_line() +
  scale_x_date(breaks = "1 year", date_labels = "%Y") +
  labs(
    title = "Reviews a lo largo del tiempo",
    x = "",
    y = "Número de reviews"
  ) +
  theme_minimal()rr
```



ggplot graph made in Rstudio | *Image by author*

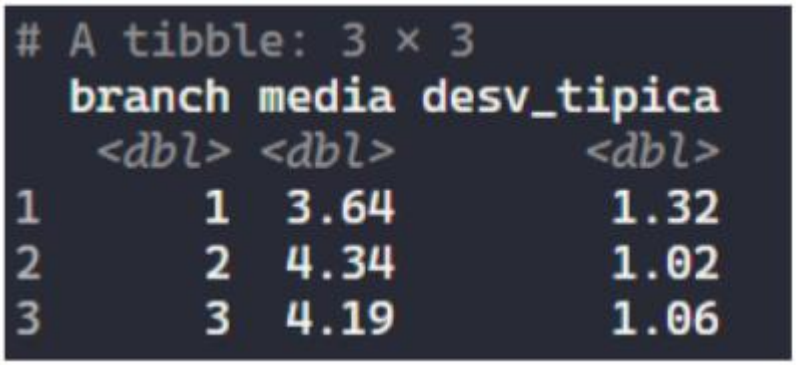
As we can see in the graph, the period with the **highest number of reviews** is towards the end of 2015, spanning between 2015 and 2016.

When analyzing the data on the graph, a relatively stable growth pattern is observed until 2015, with fluctuations in the number of reviews. However, in that year, a significant increase occurs, reaching its peak in mid-2015.

From then on, the number of reviews stabilizes until 2018, when a sharp decrease occurs, persisting until 2019, reaching its minimum.

This may indicate an **increase in the popularity** of Disneyland parks or a **rising trend of people leaving reviews** in general.

```
# MEAN:
mean(disneyland$rating)
#  4.234734

# STD
sd(disneyland$rating)
#  1.050746rr
```



```
# A tibble: 3 × 3
  branch media desv_tipica
   <dbl> <dbl>       <dbl>
1      1  3.64        1.32
2      2  4.34        1.02
3      3  4.19        1.06
```

Variables described for each branch (each disneyland park) | *Image by author*

Considering that the standard deviation is a measure of data dispersion or variability, analyzing it allows us to observe how the data deviates from the calculated mean.

These findings suggest that Disneyland California has the **highest average** score, followed by Disneyland Hong Kong, and finally, Disneyland Paris with the lowest average score. This clearly indicates which park performs worse compared to the rest, although there is uncertainty between California and Hong Kong.

Tokenization

Now we can start with our tokenization process, which consists in breaking down a text into smaller units, or tokens (words, phrases, symbols, or other meaningful elements).

In the context of natural language processing, tokenization typically involves splitting a sentence or document into individual words or phrases. This step is **crucial** for various **text analysis tasks**, as it allows for the **transformation of unstructured text data into a format that can be easily processed and analyzed by computers.**

```
data(stop_words)

# TOKENIZATION
word_true = unnest_tokens(disneyland, word, review_text) #
tokenization by words

# Counting the occurrences of each word in each park, grouped by the
park identifier
word_count <- word_true %>%
  group_by(branch) %>%
  count(word, sort = TRUE)
```

```
# Get the top 5 most frequent words for each park

# PARK 1 (branch = 1)
parque_1 <- word_count %>%
  filter(branch == 1) %>% # select data with branch = 1 to display
only the top 5 words for that park
  slice_max(n = 5, order_by = n) %>%
  arrange(branch, desc(n)) # to display the most frequent ones
# Convert to a dataframe
top_words_park1 <- as.data.frame(parque_1)
# View the table:
top_words_park1

# Repeat the process for the remaining parks, selecting the
appropriate branch for each one

# PARK 2 (branch = 2)
parque_2 <- word_count %>%
  filter(branch == 2) %>% # select data with branch = 2 to display
only the top 5 words for that park
  slice_max(n = 5, order_by = n) %>%
  arrange(branch, desc(n)) # to display the most frequent ones
# Convert to a dataframe
top_words_park2 <- as.data.frame(parque_2)
# View the table:
top_words_park2

# PARK 3 (branch = 3)
parque_3 <- word_count %>%
  filter(branch == 3) %>% # select data with branch = 3 to display
only the top 5 words for that park
  slice_max(n = 5, order_by = n) %>%
  arrange(branch, desc(n)) # to display the most frequent ones
# Convert to a dataframe
top_words_park3 <- as.data.frame(parque_3)
# View the table:
top_words_park3

# View the 3 tables together for comparison:
top_words_park1
top_words_park2
top_words_park3
```

	branch	word	n
1	1	the	52107
2	1	and	27956
3	1	to	26011
4	1	a	20141
5	1	is	13659

	branch	word	n
1	2	the	110450
2	2	and	65372
3	2	to	62492
4	2	a	46971
5	2	we	33710

	branch	word	n
1	3	the	115420
2	3	and	65671
3	3	to	59207
4	3	a	47472
5	3	of	35145

output from running top_words_park1, 2 and 3, variables | *Image by author*

As you can see, we obtain these three tables representing the top 5 most repeated words in the reviews for each park, with ‘branch’ serving as the park identifier.

As observed, all tables present results such as articles, prepositions, the copulative conjunction ‘and’, as well as other types of words used for sentence formation, but which do not provide significant information about the review content themselves.

It could be assumed that by eliminating stop words, this issue would dissipate. However, during tokenization, we merely divided the reviews, which are sentences, into words; therefore, we didn't consider all the words that form a sentence. When removing stop words, we eliminate from the tokenization result those words essential for forming meaningful sentences in the language, but which won't provide useful information for future analysis. Hence, we need to fix this in the next step, also part of the tokenization process.

```
# TOKENIZATION
word_true = unnest_tokens(disneyland, word, review_text) #
tokenization by words
disney_words = anti_join(word_true, stop_words, by = "word") # join
with stop words

# Counting the occurrences of each word in each park, grouped by the
park identifier
word_count <- disney_words %>%
  group_by(branch) %>%
  count(word, sort = TRUE)

# Get the top 5 most frequent words for each park

# PARK 1 (branch = 1)
parque_1 <- word_count %>%
  filter(branch == 1) %>% # select data with branch = 1 to display
only the top 5 words for that park
  slice_max(n = 5, order_by = n) %>%
  arrange(branch, desc(n)) # to display the most frequent ones
# Convert to a dataframe
top_words_park1 <- as.data.frame(parque_1)
# View the table:
top_words_park1

# Repeat the process for the remaining parks, selecting the
appropriate branch for each one

# PARK 2 (branch = 2)
parque_2 <- word_count %>%
  filter(branch == 2) %>% # select data with branch = 2 to display
only the top 5 words for that park
  slice_max(n = 5, order_by = n) %>%
  arrange(branch, desc(n)) # to display the most frequent ones
# Convert to a dataframe
top_words_park2 <- as.data.frame(parque_2)
# View the table:
top_words_park2

# PARK 3 (branch = 3)
parque_3 <- word_count %>%
  filter(branch == 3) %>% # select data with branch = 3 to display
only the top 5 words for that park
  slice_max(n = 5, order_by = n) %>%
  arrange(branch, desc(n)) # to display the most frequent ones
# Convert to a dataframe
top_words_park3 <- as.data.frame(parque_3)
# View the table:
top_words_park3

# View the 3 tables together for comparison:
top_words_park1
```

```
top_words_park2
top_words_park3
   branch       word    n    branch       word     n    branch   word     n
 1      1 disneyland 7121  1      2       park 17129  1      3   park 16666
 2      1       park 6721  2      2 disneyland 15150  2      3 disney 14122
 3      1     disney 6616  3      2      rides 12818  3      3  rides 12618
 4      1      rides 6238  4      2       time 12390  4      3   time 10101
 5      1        day 5984  5      2     disney 11835  5      3    day  8680
```

output from running the new top_words_park1, 2 and 3, variables | *Image by author*

Once the previous exercise is repeated, but **with the stop words removed,** we obtain these three corresponding tables again, representing the top 5 most repeated words in the reviews for each park.

Indeed, **by removing the stop words**, we are left with lists of words that are more representative of our data.

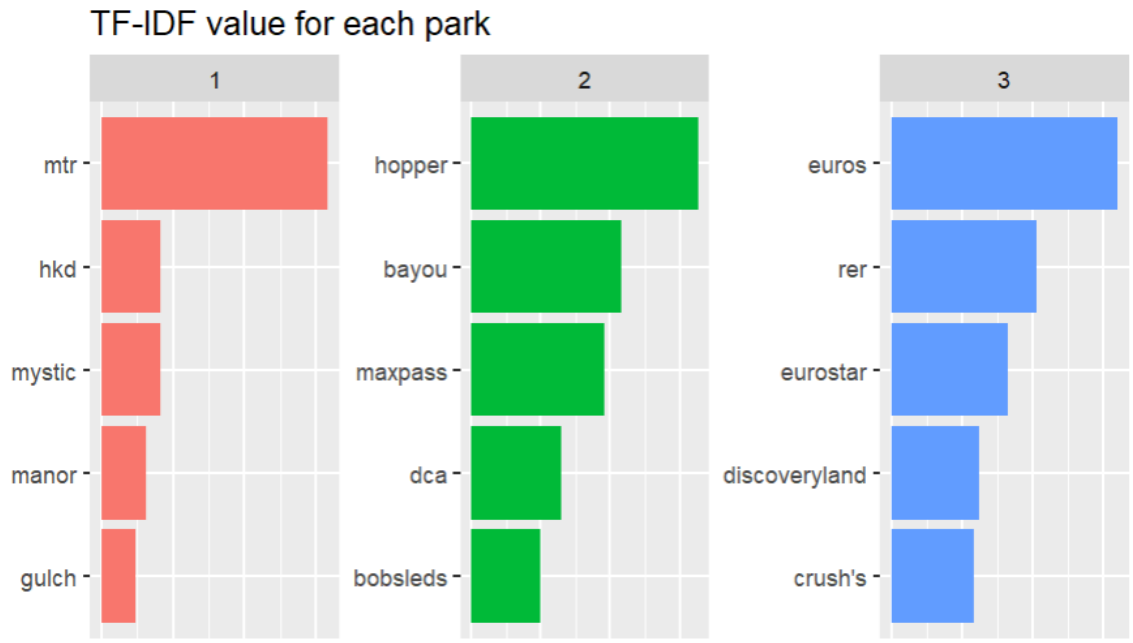
However, they still consist of **very general words** that could apply to any Disney park, and they do not truly reveal particular characteristics that would allow us to identify each branch.

TD-IDF (term frequency-inverse document frequency)

As an alternative to word frequency, we have the TF-IDF. This metric **evaluates the importance** of e**ach word** by measuring whether the word is useful for distinguishing one text from another.

For example, a word that appears frequently in one text but rarely or never in others will be considered more valuable for distinguishing between texts and will have a higher TF-IDF value than a word that is common in all texts.

```
# Applying TF-IDF and visualizing it graphically
disney_words %>%
  count(branch, word) %>%
  bind_tf_idf(word, branch, n) %>%
  arrange(desc(tf_idf)) %>%
  group_by(branch) %>%
  slice_max(tf_idf, n = 5) %>%
  ungroup() %>%
  ggplot(aes(tf_idf, fct_reorder(word, tf_idf), fill =
as.factor(branch))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(branch ~ ., ncol = 3, scales = "free") +
  labs(title = "TF-IDF Value for Each Park",
       x = "",
       y = NULL) +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank())
```



output from running the code provided | *Image by author*

When implementing TF-IDF, we encounter this graph, which represents the words with the highest TF-IDF. This allows us to easily identify each park as the words correspond to location-specific elements of the parks.

In the first graph, with **branch = 1** and coral color, we can assume that this branch corresponds to Disney Hong Kong, as the words refer to that location. For example, "Mtr" refers to the Disneyland Resort Line, which connects Sunny Bay with Hong Kong Disneyland Resort via railway.

Mtr Disneyland Resort Line | Source: https://disney.fandom.com/wiki/Disneyland_Resort_Line

In the second graph, with **branch = 2** and green color, we can infer that it represents Disneyland Orlando. Not only does it contain English words typical of the location, but it also mentions "**beyou**," a restaurant located in the United States park (https://touringplans.com/blog/disneyland-blue-bayou-review/).

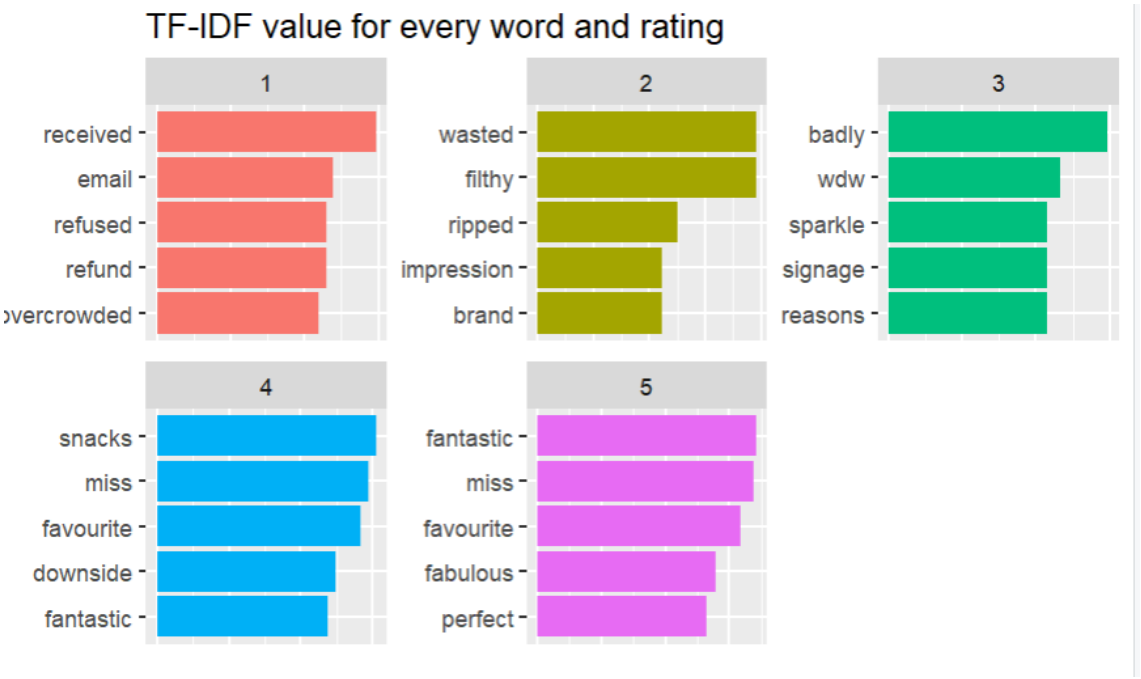
Moreover, the most repeated word is "**Hopper**," which is a type of ticket allowing entry to both Disneyland Orlando and Disney California Adventure.

Hopper Ticket | Source: Trips With Tykes

Therefore, the third graph corresponds to **branch = 3,** representing the park located in Paris, Disneyland Paris or Eurodisney. This notion is reinforced when examining its words with the highest TF-IDF, which include "**euros**" and "**eurostar**," indicating a European location.

```
# Applying TF-IDF while considering the need to view other words
disney_branch3 <- disney_words %>%
  filter(branch == 3)
disney_branch3 %>%
  count(rating, word) %>%
  filter(n > 20) %>%
  bind_tf_idf(word, rating, n) %>%
  arrange(desc(tf_idf)) %>%
  group_by(rating) %>%
  slice_max(tf_idf, n = 5) %>%
  ungroup() %>%
  ggplot(aes(tf_idf, reorder_within(word, tf_idf, rating), fill =
as.factor(rating))) +
  geom_col(show.legend = FALSE) +
  scale_y_reordered() +
  facet_wrap(rating ~ ., ncol = 3, scales = "free") +
  labs(title = "TF-IDF Value for Every Word and Rating",
       x = "",
```

```
      y = NULL) +
theme(axis.text.x = element_blank(),
      axis.ticks.x = element_blank())
```



Graph output from running the previous code | *Image by author*

Let's delve deeper into understanding people's opinions of the park with a branch value of 3. To do this, we have repeated the graph of the most relevant words in terms of TF-IDF, but this time discriminating by the rating value (note: we have retained words that appear at least 20 times).

So, in this graph we find a representation of the appearance of the most frequent words according to the rating given by users in their reviews. Thus, for each rating, we can observe the words that have been most repeated when writing the reviews.

We can discern different types of insights:

- For reviews that have given a **rating of 1**, it seems to be indicated by the words that there have been problems, which have attempted to be resolved via email to manage monetary refunds, and judging by the rating, we could say that they have not been satisfactorily resolved.
- The word "**overcrowded**" is also part of the list, which could refer to the park being too crowded to be comfortable for spending the day, or it could be related to some problem that caused someone to be left out and thus request a refund.
- In the **rating of 2**, we could conclude that it speaks of negative comments regarding the **appearance, cleanliness, and state of deterioration of the park in question,** with words such as "used," "disgusting," "torn," "impression," and "mark." It seems that the users' initial impression has not been very good…
- Regarding a **rating of 3,** the word "badly" stands out among the rest. Even though it's not such a low rating, we can see **some discontent** with what appears to be the signage in the Walt Disney World. Apparently, the signage may be insufficient to find what is desired within this specific park.

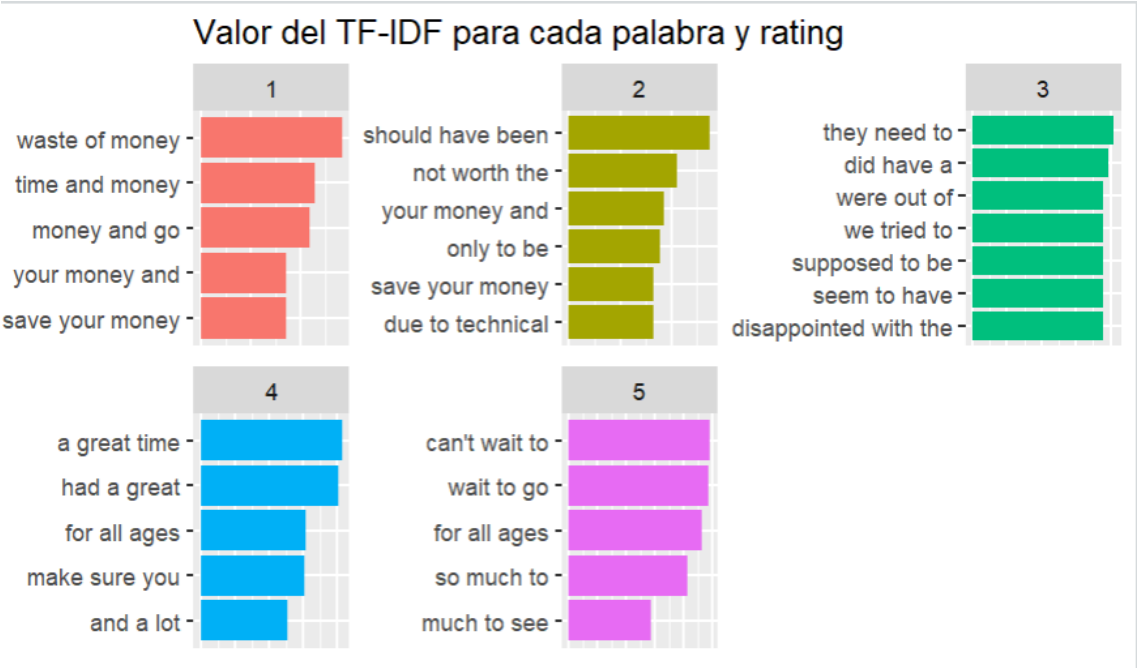
- For a **rating of 4,** one word stands out above the others: "**snacks**." It is very likely that these snacks found in the park are favorites for many visitors and are quite enjoyable. It also tells us about something that is their favorite within the venue and something they missed and expected to find within the park.
- Finally, **in rating of 5,** which is the **highest**. Overall, it details that the park is **fabulous, fantastic, and perfect.** Just like in the rating, they talk about their favorite part of the venue. They also mention things they may miss in the place, but still give the highest rating.

TF-IDF and N-Grams

```
disney_ngrams <- disneyland %>%
  unnest_tokens(word, review_text, token = "ngrams", n = 3)

disney_words <- anti_join(disney_ngrams, stop_words, by = "word") #
join with stop words

disney_branch3 <- disney_words %>%
  filter(branch == 3)
disney_branch3 %>%
  count(rating, word) %>%
  filter(n > 20) %>%
  bind_tf_idf(word, rating, n) %>%
  arrange(desc(tf_idf)) %>%
  group_by(rating) %>%
  slice_max(tf_idf, n = 5) %>%
  ungroup() %>%
  ggplot(aes(tf_idf, reorder_within(word, tf_idf, rating), fill =
as.factor(rating))) +
  geom_col(show.legend = FALSE) +
  scale_y_reordered() +
  facet_wrap(rating ~ ., ncol = 3, scales = "free") +
  labs(title = "TF-IDF Value for Each Word and Rating",
       x = "",
       y = NULL) +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank())
```

Sometimes using a single word as a token is too simple. We can utilize more than one word appearing consecutively in the text, known as **n-grams.** We repeated the previous graph, but this time using n-grams of length 3 (3 consecutive words). To do this, we tokenized the text again but changing the function to unnest_tokens().

By using n-grams of a single word, we can obtain **more detailed and precise information** about the topics mentioned by customers in their comments. This is because **n-grams capture combinations of words that are often used together and have a more specific and precise meaning.**



Grah output from the code | *Image by author*

In this graph, we analyze the **most common combinations of three consecutive words** in the reviews from all Disneyland branches, without discriminating by rating or branch. This allows us to have a more precise idea of the expressions that are most frequently used in the reviews, not just the most important isolated words. Consequently, we will have a more accurate orientation to establish a possible context that helps us better understand these reviews and the reasons for the ratings.

Sentiment Analysis (Opinion Mining)

Finally, we are going to perform something known as sentiment analysis, which aims to analyze the sentiment (whether positive or negative) of a text.

There are many ways to extract sentiment from a text. One of the most basic methods is to **associate** a **sentiment value with each word in a text;** the overall sentiment of the text will be the sum of the individual values.

There are usually databases with pre-calculated values for words. In R, we can do this by executing the following. (We do it keeping the 5 reviews with the highest sentiment value and the 5 with the lowest).

```
get_sentiments("afinn")
disney_sentiment <- disneyland %>%
 unnest_tokens(word, review_text) %>%
 inner_join(get_sentiments("afinn")) %>%
 group_by(review_id, rating) %>%
 summarise(sentiment = sum(value))
head(disney_sentiment)
```

	review_id	rating	sentiment
1	117296400	4	234
2	230962271	5	226
3	168242566	5	207
4	473516988	5	159
5	350318985	5	154

	review_id	rating	sentiment
1	110066682	4	-83
2	182917369	1	-54
3	452236934	1	-42
4	615064889	1	-37
5	184474692	2	-32

5 reviews with highest level of sentiment (left) and lowest (right) | *Image by author*

It can be said that they correspond to the rating given by the user. Although we have an exception in the table with the 5 reviews with the lowest value. But generally analyzing, we see that when the sentiment is low, a low rating is given, and if we look at the high sentiment, we observe a high rating.

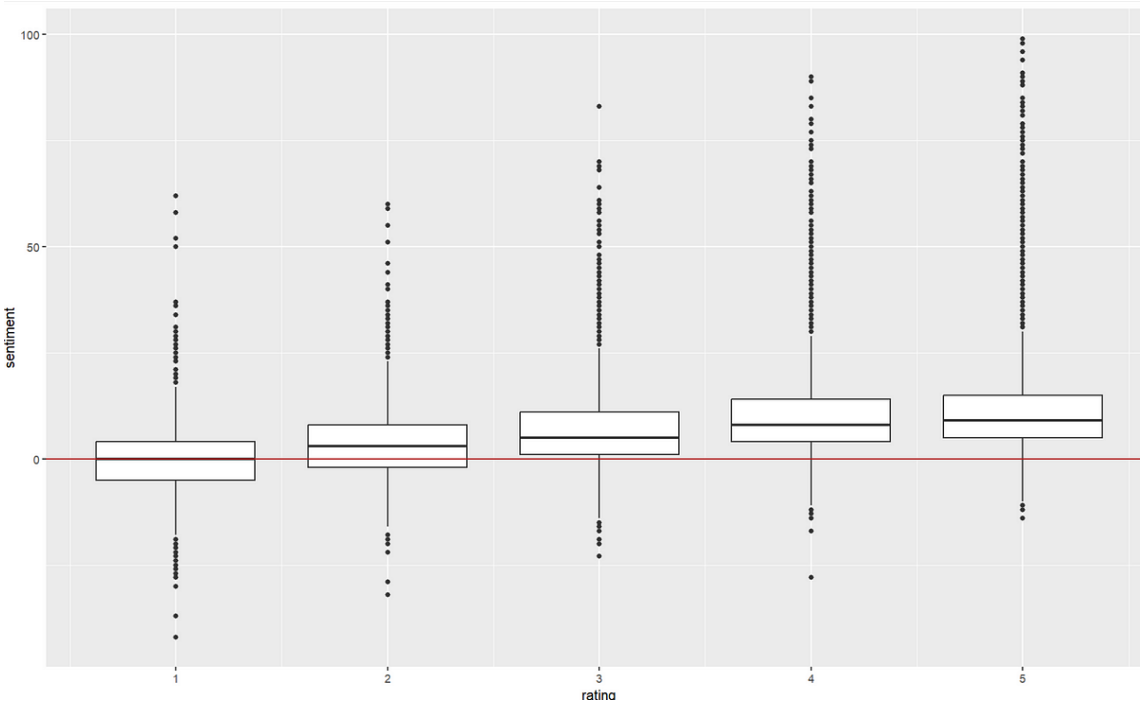
If the **relationship** between **rating and sentiment** is **positive**, it means that as the **rating increases**, the positive sentiment of the reviews also increases. However, if the relationship is negative, it means the opposite.

In general, a **positive correlation** between **rating and sentiment** is expected, as reviews with a high score are supposed to convey a positive sentiment and reviews with a low score are supposed to convey a negative sentiment. However, there may be exceptions or nuances depending on the circumstances and personal experiences of each user.

In any case, to verify if the relationship holds true in this specific dataset, it is necessary to analyze the obtained graph and calculate the correlation between rating and sentiment, which would be the following step.

This code generates a graph that shows the relationship between the rating and sentiment of Disneyland reviews:

```
disney_sentiment %>%
  filter(sentiment < 100,
         sentiment > -50) %>%
  ggplot(aes(x = rating, y = sentiment)) +
  geom_boxplot(aes(group = rating)) +
  geom_hline(yintercept = 0, color = "firebrick")
```



Graph obtained from running the previous code | *Image by author*

Here, we can observe that the majority of reviews with a high rating (4 and 5) have a positive sentiment, while reviews with a low rating (1 and 2) have a negative sentiment.

However, it is important to clarify that we are conducting a very basic sentiment analysis, and there may be cases where the rating and sentiment do not completely match.

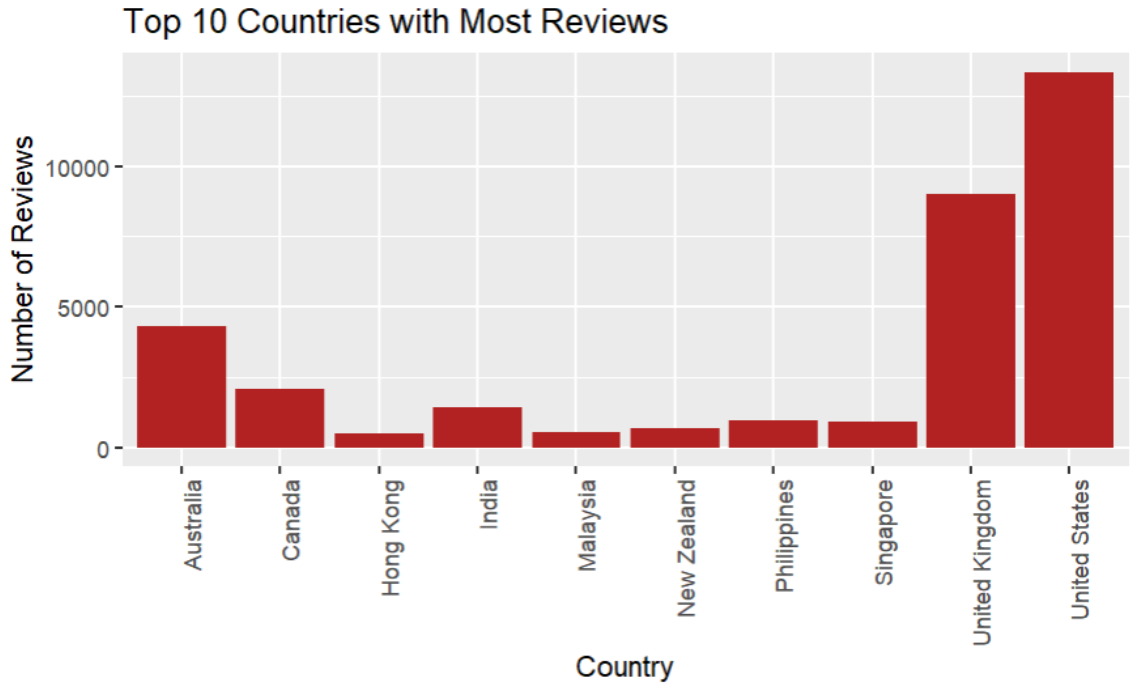
Additionally, this analysis does not take into account the context of the reviews and there could be false positives or negatives. Therefore, it is important to take a **critical approach** and consider other more advanced techniques for a more accurate sentiment analysis.

Extra Step for Further Analysis

The dataset contains the variable “reviewer_location” with the origin of the user who has made the review. This variable has not been used at all until now, that we will analyze this variable to obtain some relevant conclusion.

```
# We store the number of reviews sorted from highest to lowest
country_count <- disneyland %>%
  count(reviewer_location, sort = TRUE)

# Graph with the top 10 countries with the most reviews
# The x-axis represents the country and the y-axis represents the
number of reviews
ggplot(data = country_count[1:10, ], aes(x = reviewer_location, y =
n)) +
  geom_bar(stat = "identity", fill = "firebrick") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Country", y = "Number of Reviews", title = "Top 10
Countries with Most Reviews")

# The code part "axis.text.x = element_text(angle = 90, hjust = 1)"
# changes the orientation of the x-axis labels to a 90-degree
```

```
# counterclockwise angle. Additionally, hjust = 1 centers them.

# `geom_col()` is used to create bar charts where the height of each
bar is determined by a variable on the Y-axis.
# The `fill` option is used to set the fill color of each bar.
```



outcome graph | *Image by author*

Based on the analysis of the `reviewer_location` variable, we can draw three interesting conclusions:

1. The majority of users who have left reviews are English speakers, such as the United States, the United Kingdom, Canada, or Australia.
2. There is a significant number of users from Asian countries, such as Japan, China, or South Korea, indicating a high popularity of Disneyland parks in that region of the world.
3. Conversely, there are very few users from Latin American countries, suggesting that Disneyland parks are not a very popular tourist destination in that region.

Intrigued by our exploration of Disneyland's digital landscape?

Remember! If you've liked my article and want to keep up, you can find me in:

- LinkdIn: https://www.linkedin.com/in/sandra-galiano-bernardino-aaa677222/
- Medium: https://medium.com/@SanGaliber