Combined assessment :: Data Visualization and Text Mining

Airbnb Case Study

<u>Team 3</u>: Raquel Alvarenga, Matteo Meroni, Noor Hejeeali , Ece Aker, and Darshil Panchal.
Data Visualization and Text Analytics and Natural Language Processing (NLP)
MBAN1
12/14/2021
Hult International Business School

Executive Summary

Airbnb guest reviews in the USA, Brazil, and Spain are remarkably consistent, even though they have different geographies and different cultures. The top factors to a successful property review in these countries are the accuracy of the property description, location, cleanliness, comfort level, and interaction with the host. Guests expressed highly negative sentiment towards noisy areas and properties lacking a heating system. Furthermore, properties listed in the target countries are in coastal cities, with apartment units as the most popular rentals.

Introduction

This report analyzes Airbnb customers' reviews across three different markets: USA, Spain, and Brazil. The reviews were filtered to English only to standardize and avoid complexity. The main objective is to understand if customers' preferences when renting a property vary from country to country. The analysis is supported by various text mining techniques and visualizations to identify patterns and business insights in property reviews. The term *token* is commonly used in this report and refers to specific words extracted from the reviews.

Word Frequencies

Word frequencies in the USA, Brazil, and Spain reviews indicate that guests emphasize the location and cleanliness of units and the interaction with the host during their stay*(see Appendix I)*. The token *beach* has one of the highest frequencies in reviews, which is foreseen as the properties listed in our target countries are close to the ocean *(see Appendix II)*. In addition, it also seems that apartment units are the properties most in-demand in these locations.

Words Frequency Correlation

To understand the similarity and relationship of tokens in the property reviews, we have performed a correlogram along with a correlation test that compared tokens in the USA with

Brazil and Spain *(see Appendix III)*. USA and Brazil tokens have a correlation coefficient of 0.90, while USA and Spain correlate by 0.87. Both sets are strongly correlated, with words outside the line being more specific for each country. For example, the words “New York” and “condo” are more specific to the US, the names “Alan” and “Marcela” to Brazil, and “Barcelona” and “Julian” to Spain. The USA has much more condominiums listed on Airbnb than Brazil or Spain, explaining the exclusivity of the term “condo” *(see Appendix III and dashboard 1)*.

Sentiment Analysis

We performed a sentiment analysis to determine customers’ positive and negative associations with properties in their reviews. The most common words related to a specific sentiment were remarkably homogeneous in all countries *(see Appendix V)*. In brief, people are looking for clean, quiet, safe, and comfortable places to enjoy their stay, and if overall satisfied, there is a high chance they will end up recommending the property in the review. However, on the other hand, there is a negative sentiment for properties that reside in noisy areas and properties lacking a heating system.

N-Gram

So far, we understand the most common words and sentiments in our reviews. The next step is to analyze the semantic structure of our terms to understand their level of association and gain more context. The bigrams from the USA, Brazil, and Spain *(See Appendix VI)* point in the same direction; guests factor in the location of properties when booking on the platform. Furthermore, they refer to the amenities of the place and the host experience. Hosts categorized as Superhosts have a median review score rating of 97 on their properties, while non-super hosts have a 93 *(See Appendix IV & Dashboard 3)*.

TF-IDF

The Term Frequency and Inverse Document Frequency helped identify tokens with less frequency but with the highest importance, which provided more insights into our analysis. The most relevant words in our TF-IDFs refer to local, touristic cities and monuments in each destination. For example, we encounter New York, Manhattan, Brooklyn, and the Hawaiian Islands in the USA. In Spain, the city Barcelona, with mentions of characteristic places such as the Sagrada Familia and the Rambas. Lastly, Brazil stands out for the coastal cities Rio de Janeiro, Ipanema, and Copacabana *(see Appendix VII and dashboard 4)*.

Gini Decision Tree

We have designed a Gini Decision Tree model based on a sample of 10,000 reviews to predict business success or failure based on reviews score rating. The reviews score rating has a range that goes from 1 to 100: If the total review score per property is greater than 90, we classify it as business success and below 90 as a business failure *(see Appendix VIII)*. Each other category that we used to predict the outcome has a score range between 1-10. According to the model, there are high chances of business success if the review score accuracy of the listing is equal to 10: this confirms the importance of offering a description of the accommodation that is complete and honest. If less, there are other scores to factor, including cleanliness, checking, and communication with the host *(see dashboard 2)*.

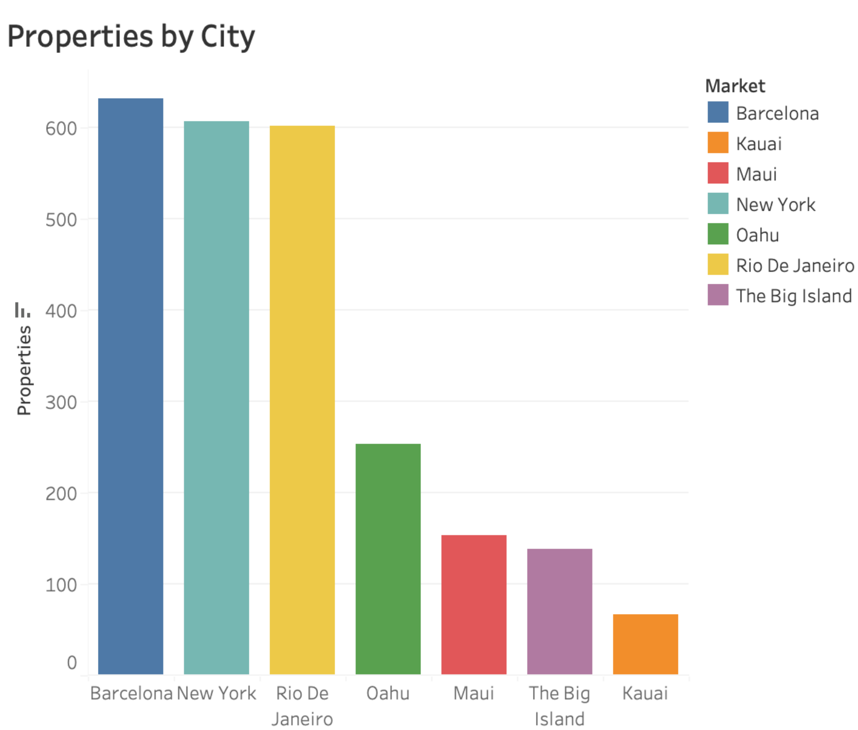
Conclusion

In conclusion, most reviews are positive in the three countries analyzed, and the topics are homogeneous. Customers are happy when the description of the accommodation is consistent with what they find and instead complain when they have to stay in dirty or noisy places.

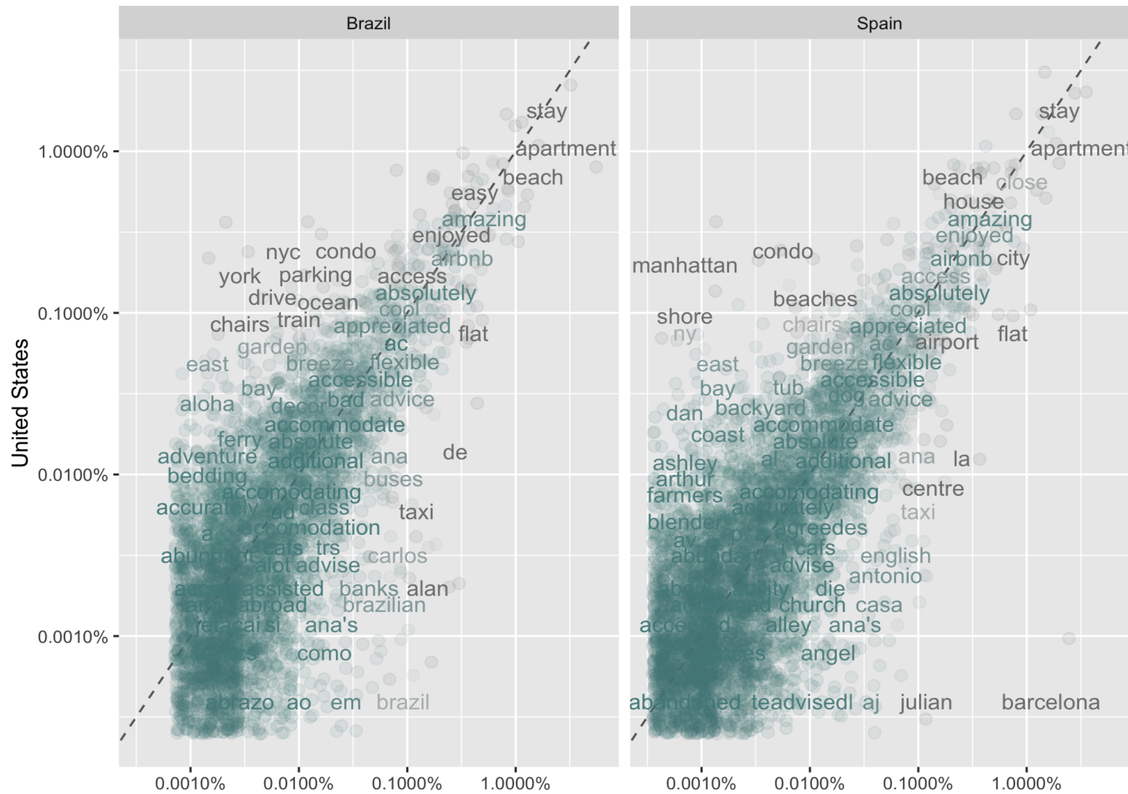
Appendix

Appendix I : Word Cloud of the 20 most frequent words in the reviews

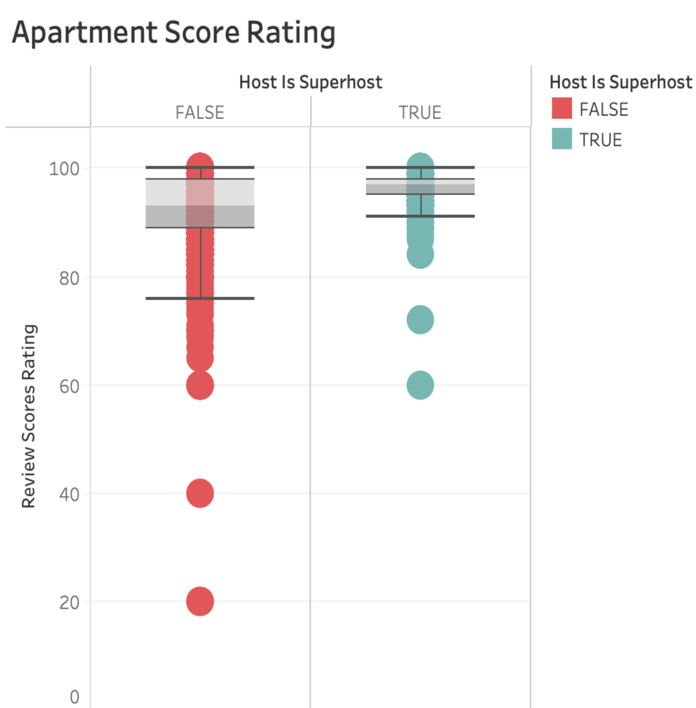
Appendix II : Number of Properties by City



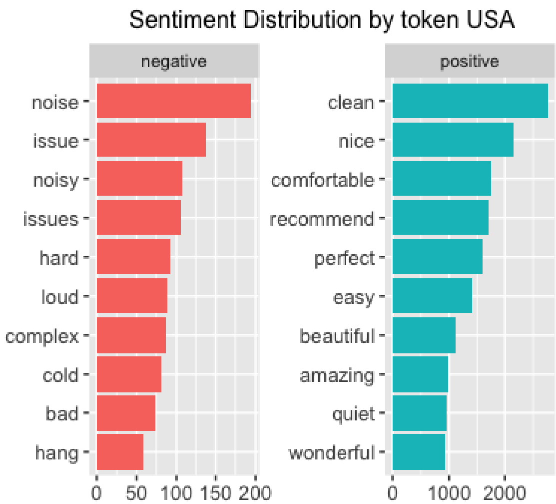
Appendix III: Correlogram of United States vs Brazil and Spain

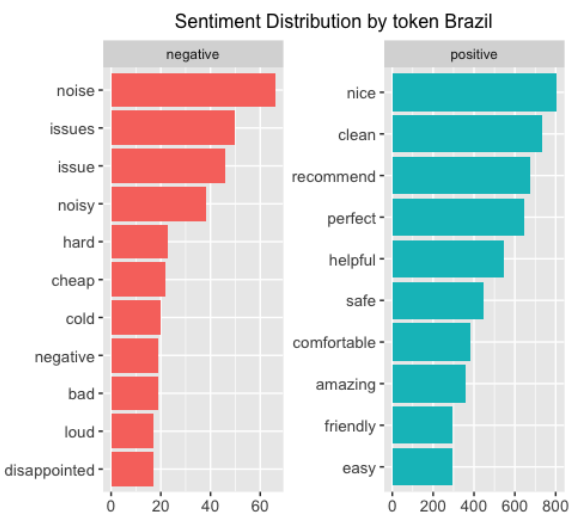


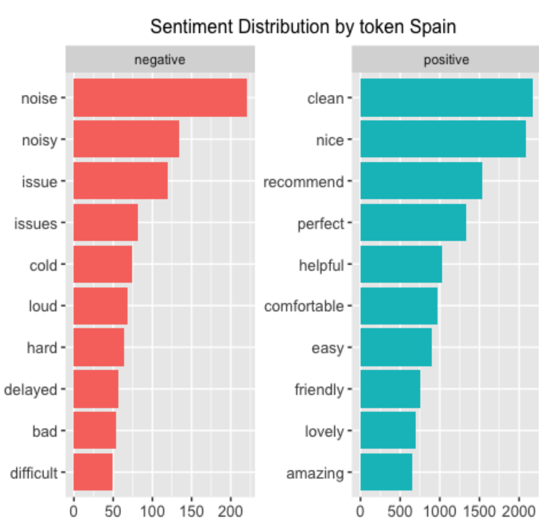
Appendix IV: Boxplot of the Apartment Score Rating By Type of Host



Appendix V: Sentiment Analysis in the USA, Brazil, and Spain.

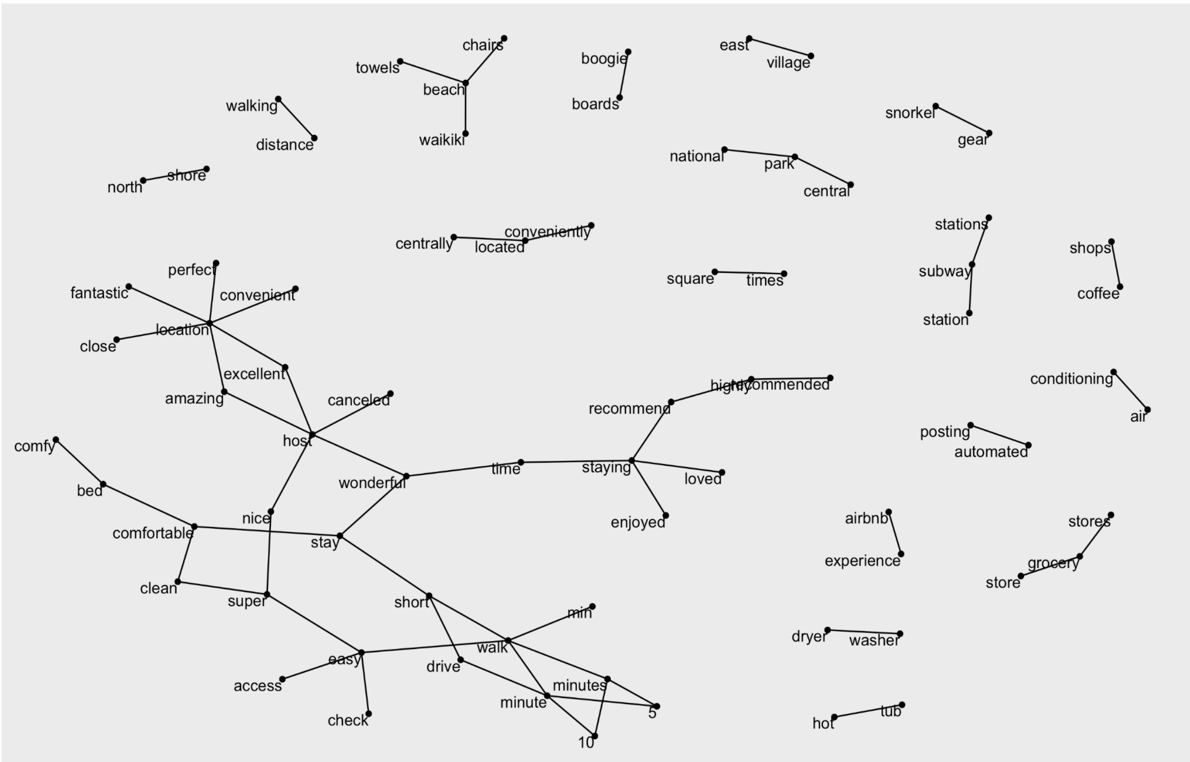




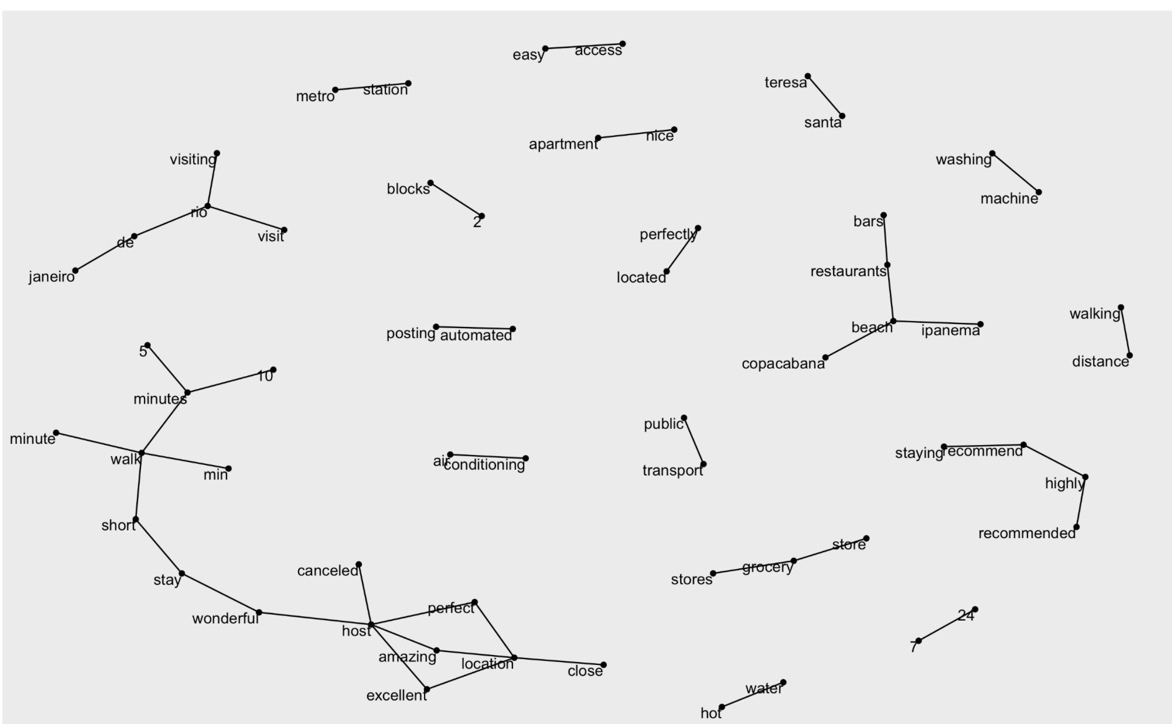


Appendix VI: Bigrams

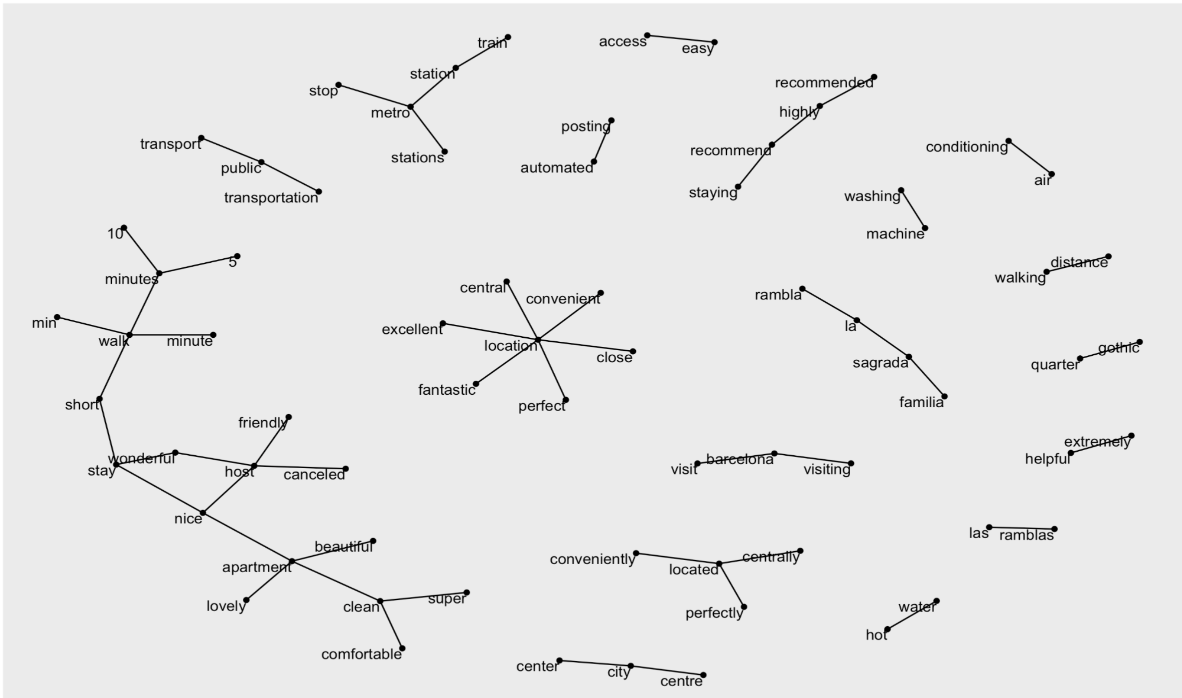
<u>Bi-gram analysis for Brazil's reviews</u>



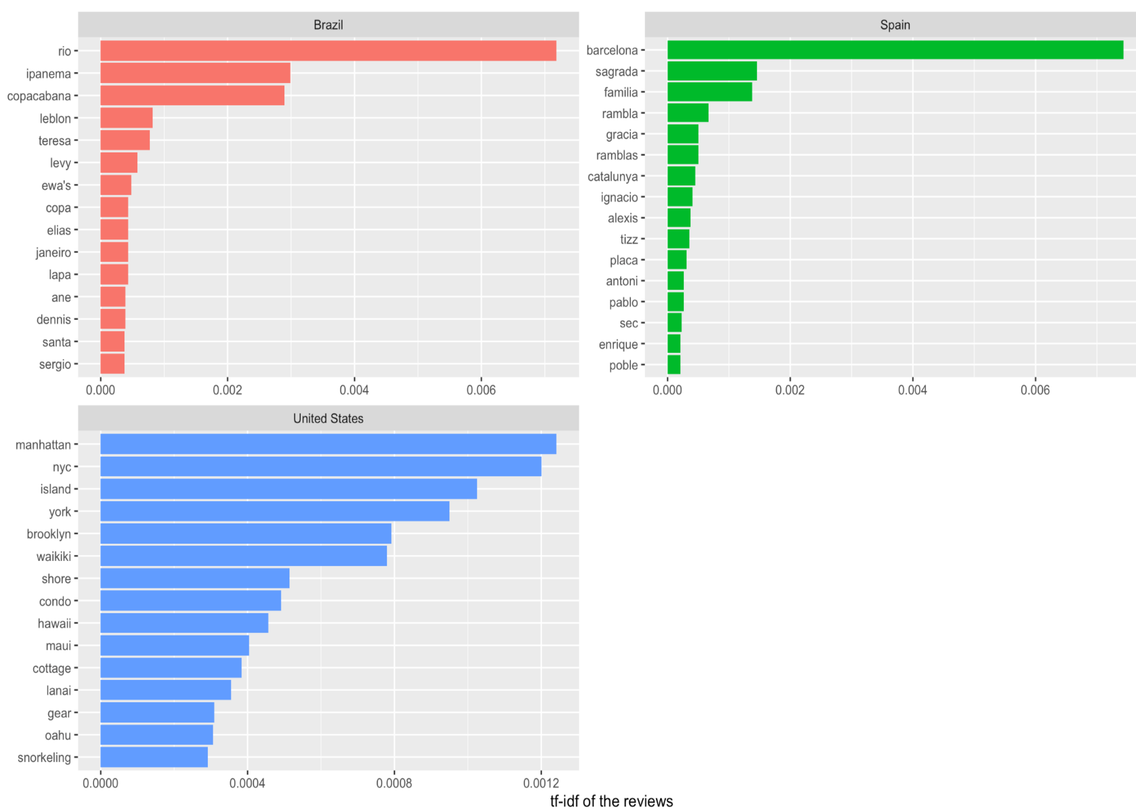
<u>Bi-gram analysis for Brazil's reviews</u>



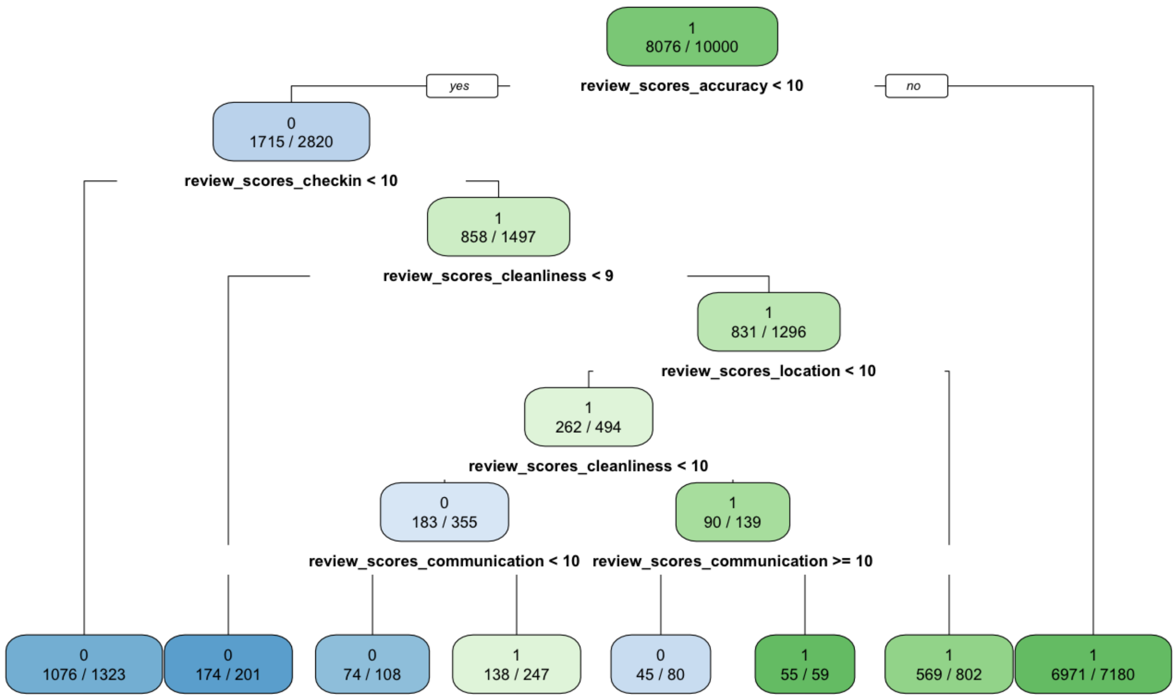
Bi-gram analysis for Spain's reviews



Appendix VII: TF-IDF for Reviews

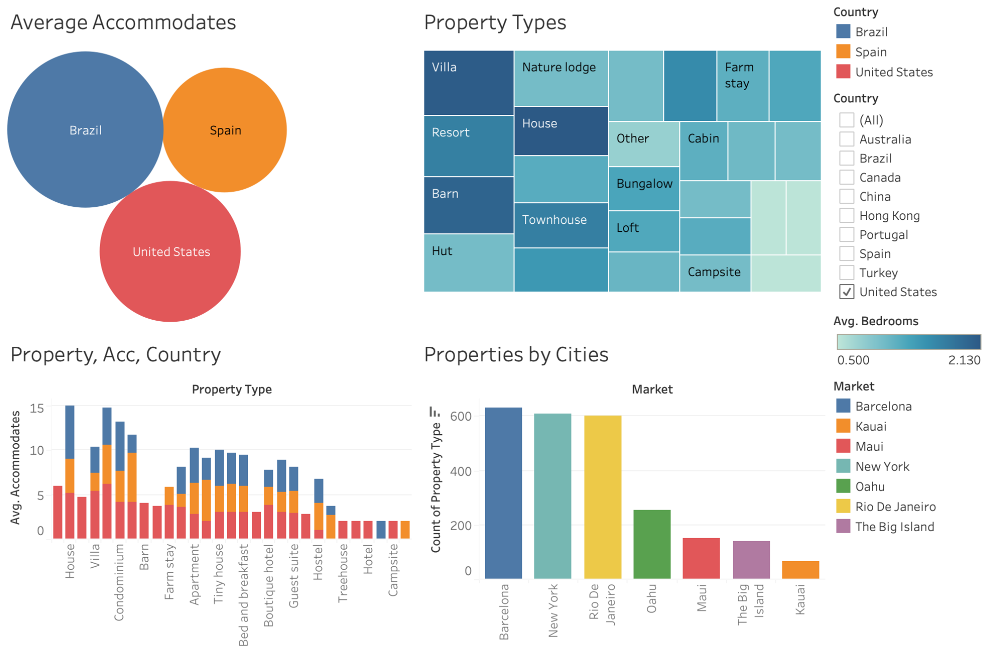


Appendix VIII: Gini Decision Tree Predictive Model for the Reviews Score



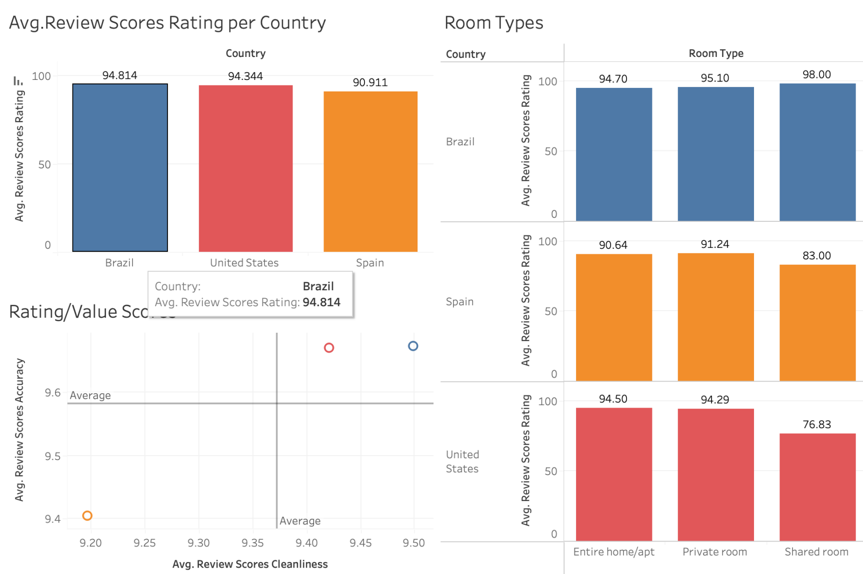
Dashboards

Dashboard 1:



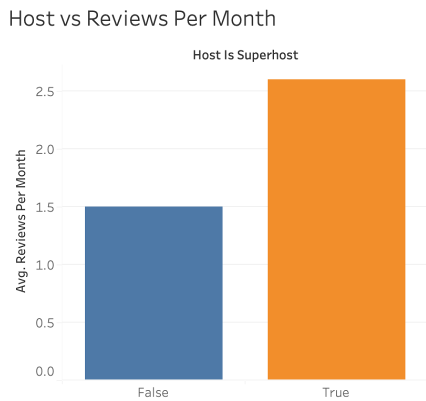
Either a resort, house, or apartment is the most popular choice in these 3 countries. A house is popular in Brazil while a townhouse is popular in Spain. A resort is a popular listing in the United States. Properties which are closer to the beach have a higher number of bookings compared to properties listed in the city. Brazil has the highest average amount of accommdates for a property compared to Spain and the United States. Barcelona and New York have the highest number of property listings.

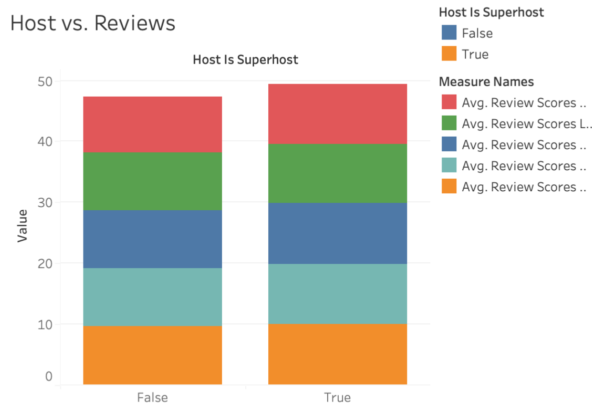
Dashboard 2:

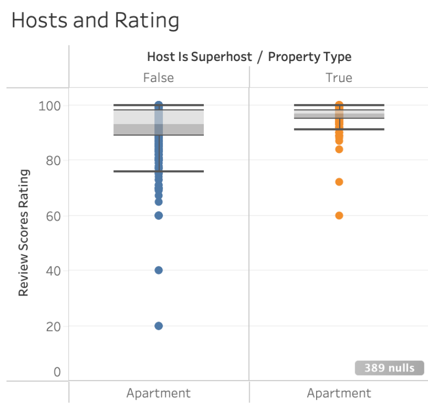


Brazil has the highest average score rating. United States has the second highest with the last one being Spain. The review score check-in and cleaniness are the 2 most important review scores factor. We saw this from the Gini-Tree and see it has the highest impact. United States and Brazil are above the average in these 2 specific review score ratings while Spain is below the average. The shared room in Brazil has the highest rating where private room has the highest room in Spain and United States.

Dashboard 3:





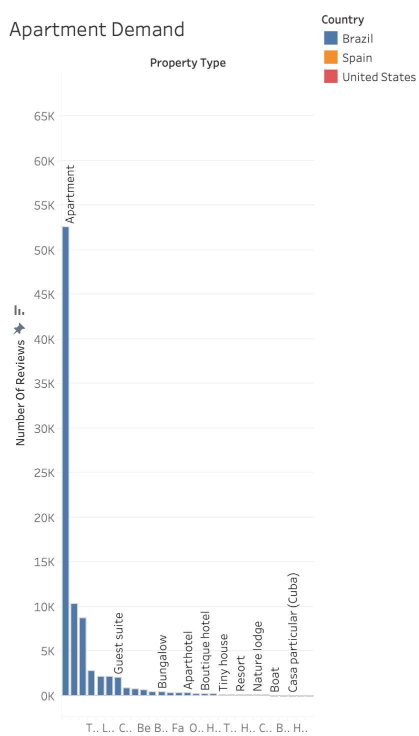


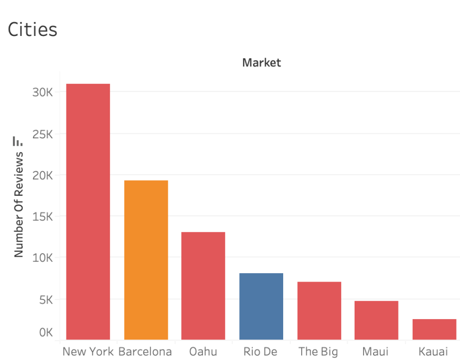
If a superhost is more likely to get a overall higher review score rating compared to the a host who is not a superhost. A superhost has fewer property listings but a higher amount of reviews per month.

Dashboard 4:

In our The Term Frequency and Inverse Document analysis, we prefered to focus on cities. As you can see from the Cities graph, cities that we encountered in our analysis also relevant and high proportion in the number of reviews.

Furthermore, in our word frequency analysis, we realized that the token "beach" is quite frequent and apartment units in our target countries which are close to the beach have high demands. So we want to ensure about this statement also in our visualization. According to Apartment Demand graph you can clearly see that the high demand for apartment is quite clear in number of reviews.





R Code

```
#Downloading Libraries
library(tidytext)
library(tidyverse)
library(textdata)
library(wordcloud)
library(RColorBrewer)
library(wordcloud2)
library(tm)
library(ggthemes)
library(igraph)
library(ggraph)
library(janeaustenr)
library(dplyr)
library(stringr)
library(tidyr)
library(tidytuesdayR)
library(mongolite)
library(splitstackshape)
library(textcat)
library(wordcloud2)
library(ggplot2)
library(scales)
library(rpart)
library(rpart.plot)
library(topicmodels)

#Connecting to MongoDB
connection_string <-
'mongodb+srv://matteomeroni:PqUXcQYiaqu7wWw@cluster0.u48ss.mongodb.net/sample_airbnb?retry
Writes=true&w=majority'
airbnb_collection <- mongo(collection="listingsAndReviews", db="sample_airbnb",
url=connection_string)

airbnb_all <- airbnb_collection$find()

glimpse(airbnb_all)

#creating the data frame that we are going to use for the analysis
my_airbnb <- airbnb_all %>%
  select(-images, -host, -address, -availability, -review_scores, -reviews)
my_airbnb <- cbind(my_airbnb, airbnb_all$review_scores, airbnb_all$availability, airbnb_all$address,
airbnb_all$host )
my_airbnb <- cSplit(my_airbnb, "listing_url", "/") %>%
```

```
  select(-listing_url_1, -listing_url_2, -listing_url_3, -listing_url_4)

#remove empty data frame reviews
reviews_text <- airbnb_all$reviews
reviews_text <- Filter(function(x) dim(x)[1] > 0, reviews_text)

#create data frame with all the reviews
reviews_text <- bind_rows(reviews_text) %>%
  select(listing_id, comments)
reviews_text$listing_id <- as.numeric(reviews_text$listing_id)

#inner_join my_airbnb with the reviews
my_airbnb_join <- my_airbnb%>%
  inner_join(reviews_text, by = c("listing_url_5" = "listing_id"))
my_airbnb_join$comments <- sapply(my_airbnb_join$comments, function(x) gsub("[^\x01-\x7F]", "",
x))
table(my_airbnb_join$country)

#export the file that we are goingto usefor the analysis on Tableau
tableau <- my_airbnb %>%
  select(market, country, property_type, room_type, room_type, bed_type, minimum_nights,
maximum_nights, cancellation_policy, number_of_reviews, bedrooms, bed_type, beds, accommodates,
price, security_deposit, bathrooms, cleaning_fee, guests_included, extra_people, reviews_per_month,
review_scores_accuracy, review_scores_cleanliness, review_scores_checkin,
review_scores_communication, review_scores_location, review_scores_location, review_scores_value,
review_scores_rating)

write.csv(tableau,"/Users/matteomeroni/Desktop/Text Analytics/tableau_file.csv")

########################################
######### WORDCLOUD #############
########################################

#wordcloud of the 50 most frquent words in the comments
token <- my_airbnb_join %>%
  filter(country %in% c("United States", "Brazil", "Spain")) %>%
  unnest_tokens(word,comments) %>%
  anti_join(stop_words) %>%
  count(word) %>%
  top_n(50,n)

wordcloud2(token, size= 0.4, color='random-dark')

#wordcloud of the 50 most frquent AMENITIES
```

```
amenities <- my_airbnb_join$amenities
amenities <- data.frame(unlist(amenities)) %>%
  count(unlist.amenities.)

amenities_50 <-  amenities%>%
  top_n(50)

wordcloud2(amenities_50, size= 0.2, color='random-dark')

###########################
##########SAMPLES#######
###########################

#crate samples of United States, Brazil and Spain filtering  for english
united_states <- my_airbnb_join %>%
  filter(country == "United States")
brazil <- my_airbnb_join %>%
  filter(country == "Brazil")
spain <- my_airbnb_join %>%
  filter(country == "Spain")

sample_usa <- united_states[sample(nrow(united_states), 10000), ]
sample_spain <- spain[sample(nrow(spain), 10000), ]

sample_usa_eng <- sample_usa %>%
  mutate(lang = textcat(comments)) %>%
  filter(lang == "english")

sample_brazil_eng <- brazil %>%
  mutate(lang = textcat(comments)) %>%
  filter(lang == "english")

sample_spain_eng <- sample_spain %>%
  mutate(lang = textcat(comments)) %>%
  filter(lang == "english")

##############################################################
########### Sentiment analysis with USA #######################
################################################################
usa_token <- sample_usa_eng %>%
  unnest_tokens(word, comments) %>%
  anti_join(stop_words)

###############################################################
```

```
######## Most common positive and negative words #############
##################################################################

bing_counts_usa <- usa_token %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
  ungroup()

bing_counts_usa %>%
    group_by(sentiment) %>%
    top_n(10) %>%
    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 sentiment United States", x=NULL)+
    coord_flip()

#NRC Sentiment
USA_nrc <-USA_token_sentiment %>%
  inner_join(get_sentiments("nrc")) %>%
  count(word,sentiment,sort=TRUE) %>%
  ungroup()

#My NRC Count
USA_nrc_count<- USA_nrc %>%
  group_by(sentiment) %>%
  count(sentiment,sort=TRUE)

#plotting my sentiments
USA_sentiment <- USA_nrc_count %>%
  ggplot(aes(x = reorder(sentiment, n),y = n, fill= sentiment)) +
  geom_col(show.legend = FALSE) +
  coord_flip() +
  labs(x = NA,
       y = "n",
       title = "Number of words by sentiment")+
  theme(axis.title.y = element_blank())+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(axis.text = element_text(size = 12))+
  theme(plot.title = element_text(size = 13))

USA_sentiment
```

```
#######################################################################
########### Sentiment analysis with Spain #######################
#######################################################################

spain_token <- sample_spain_eng %>%
   unnest_tokens(word, comments) %>%
   anti_join(stop_words)

######################################################################
######## Most common positive and negative words ##############
######################################################################

bing_counts_spain <- spain_token %>%
   inner_join(get_sentiments("bing")) %>%
   count(word, sentiment, sort=T) %>%
   ungroup()

bing_counts_spain %>%
   group_by(sentiment) %>%
   top_n(10) %>%
   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 sentiment Spain", x=NULL)+
   coord_flip()

#######################################################################
########### Sentiment analysis with Brazil ##########################
#######################################################################

brazil_token <- sample_brazil_eng %>%
  unnest_tokens(word, comments) %>%
  anti_join(stop_words)

######################################################################
######## Most common positive and negative words ##############
######################################################################

bing_counts_brazil <- brazil_token %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort=T) %>%
```

```
  ungroup()

bing_counts_brazil %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  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 sentiment Brazil", x=NULL)+
  coord_flip()



##############################################
###### N-grams and tokenizing USA ###########
##############################################

usa_bigrams <- sample_usa_eng %>%
   unnest_tokens(bigram, comments, token = "ngrams", n=2)

#to remove stop words from the bigrams data, we need to use the separate function:
usa_bigrams_separated <- usa_bigrams %>%
   separate(bigram, c("word1", "word2"), sep = " ")

usa_bigrams_filtered <- usa_bigrams_separated %>%
   filter(!word1 %in% stop_words$word) %>%
   filter(!word2 %in% stop_words$word)

#creating the new bigram, "no-stop-words":
usa_bigram_counts <- usa_bigrams_filtered %>%
   count(word1, word2, sort = TRUE)

#####################################################
####### VISUALIZING A BIGRAM NETWORK #################
#####################################################

usa_bigram_graph <- usa_bigram_counts %>%
   filter(n>50) %>%
   graph_from_data_frame()

ggraph(usa_bigram_graph, layout = "fr") +
```

```
  geom_edge_link()+
  geom_node_point()+
  geom_node_text(aes(label=name), vjust =1, hjust=1)

     #creating bigram for comfortable
     comfortable_bigram <- bigram_counts %>%
       filter(word1 == "comfortable")

     comfortable_graph <- comfortable_bigram %>%
       filter(n>5) %>%
       graph_from_data_frame()

     ggraph(comfortable_graph, layout = "fr") +
       geom_edge_link()+
       geom_node_point()+
       geom_node_text(aes(label=name), vjust =1, hjust=1)

     #creating bigram for location
     location_bigram <- bigram_counts %>%
       filter(word1 == "location")

     location_graph <- location_bigram %>%
       filter(n>5) %>%
       graph_from_data_frame()

     ggraph(location_graph, layout = "fr") +
       geom_edge_link()+
       geom_node_point()+
       geom_node_text(aes(label=name), vjust =1, hjust=1)

################################################
###### N-grams and tokenizing Brazil ###########
################################################
brazil_bigrams <- sample_brazil_eng %>%
  unnest_tokens(bigram, comments, token = "ngrams", n=2)

#to remove stop words from the bigrams data, we need to use the separate function:
brazil_bigrams_separated <- brazil_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

brazil_bigrams_filtered <- brazil_bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
```

```
#creating the new bigram, "no-stop-words":
brazil_bigram_counts <- brazil_bigrams_filtered %>%
   count(word1, word2, sort = TRUE)

###########################################################
####### VISUALIZING A BIGRAM NETWORK #################
###########################################################
brazil_bigram_graph <- brazil_bigram_counts %>%
   filter(n>25) %>%
   graph_from_data_frame()

ggraph(brazil_bigram_graph, layout = "fr") +
   geom_edge_link()+
   geom_node_point()+
   geom_node_text(aes(label=name), vjust =1, hjust=1)

###################################################
###### N-grams and tokenizing Spain ###########
###################################################
spain_bigrams <- sample_spain_eng %>%
  unnest_tokens(bigram, comments, token = "ngrams", n=2)

#to remove stop words from the bigrams data, we need to use the separate function:
spain_bigrams_separated <- spain_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

spain_bigrams_filtered <- spain_bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

#creating the new bigram, "no-stop-words":
spain_bigram_counts <- spain_bigrams_filtered %>%
  count(word1, word2, sort = TRUE)

###########################################################
####### VISUALIZING A BIGRAM NETWORK #################
###########################################################
spain_bigram_graph <- spain_bigram_counts %>%
  filter(n>50) %>%
  graph_from_data_frame()

ggraph(spain_bigram_graph, layout = "fr") +
  geom_edge_link()+
  geom_node_point()+
```

```
  geom_node_text(aes(label=name), vjust =1, hjust=1)



##########################################################
######## TF-IDF framework in Airbnb reviews ##########
##########################################################

airbnb_reviews_idf <- my_airbnb_join %>%
   select(country, comments) %>%
   filter(country %in% c("United States", "Brazil", "Spain"))

sample_airbnb_reviews_idf <- airbnb_idf[sample(nrow(airbnb_idf), 10000), ]

sample_airbnb_reviews_idf_eng <- sample_airbnb_reviews_idf %>%
    mutate(lang = textcat(comments)) %>%
    filter(lang == "english")

reviews_airbnb_token <- sample_airbnb_reviews_idf_eng %>%
    unnest_tokens(word, comments) %>%
    count(country, word, sort=TRUE) %>%
    ungroup()

reviews_total_words <- reviews_airbnb_token %>%
    group_by(country) %>%
    summarize(total=sum(n))

reviews_airbnb_words <- left_join(reviews_airbnb_token, reviews_total_words)

ggplot(reviews_airbnb_words, aes(n/total, fill = country))+
    geom_histogram(show.legend=FALSE)+
    xlim(NA, 0.001) +
    facet_wrap(~country, ncol=2, scales="free_y")

#######################################################
################# TF_IDF ##########################
#######################################################

reviews_country_words <- reviews_airbnb_words %>%
   bind_tf_idf(word, country, n)

reviews_country_words %>%
  arrange(desc(tf_idf)) %>%
```

```
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(country) %>%
  top_n(15) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=country))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf of the reviews")+
  facet_wrap(~country, ncol=2, scales="free")+
  coord_flip()

#####################################################
####### TF-IDF framework in Airbnb summary ###########
#####################################################

airbnb_summary_idf <- my_airbnb %>%
   select(country, summary) %>%
   filter(country %in% c("United States", "Brazil", "Spain"))

airbnb_summary_idf_eng <- airbnb_summary_idf %>%
    mutate(lang = textcat(summary)) %>%
    filter(lang == "english")

airbnb_summary_token <- airbnb_summary_idf_eng %>%
    unnest_tokens(word, summary) %>%
    count(country, word, sort=TRUE) %>%
    ungroup()

summary_total_words <- airbnb_summary_token %>%
    group_by(country) %>%
    summarize(total=sum(n))

airbnb_summary_words <- left_join(airbnb_summary_token, summary_total_words)

ggplot(airbnb_summary_words, aes(n/total, fill = country))+
    geom_histogram(show.legend=FALSE)+
    xlim(NA, 0.001) +
    facet_wrap(~country, ncol=2, scales="free_y")

##################################################
################# TF_IDF ##########################
##################################################

summary_country_words <- airbnb_summary_words %>%
    bind_tf_idf(word, country, n)
```

```
summary_country_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word=factor(word, levels=rev(unique(word)))) %>%
  group_by(country) %>%
  top_n(15) %>%
  ungroup %>%
  ggplot(aes(word, tf_idf, fill=country))+
  geom_col(show.legend=FALSE)+
  labs(x=NULL, y="tf-idf Summary")+
  facet_wrap(~country, ncol=2, scales="free")+
  coord_flip()

#####################################################
################# CORRELOGRAMS #################
#####################################################

frequency <- bind_rows(mutate(usa_token, author="United States"),
                       mutate(brazil_token, author= "Brazil"),
                       mutate(spain_token, author="Spain")
         )%>% #closing bind_rows
          mutate(word=str_extract(word, "[a-z']+")) %>%
          count(author, word) %>%
          group_by(author) %>%
          mutate(proportion = n/sum(n))%>%
          select(-n) %>%
          spread(author, proportion) %>%
          gather(author, proportion, `Brazil`, `Spain`)

#let's plot the correlograms:
ggplot(frequency, aes(x=proportion, y=`United States`,
          color = abs(`United States`- proportion)))+
          geom_abline(color="grey40", lty=2)+
          geom_jitter(alpha=.1, size=2.5, width=0.3, height=0.3)+
          geom_text(aes(label=word), check_overlap = TRUE, vjust=1.5) +
          scale_x_log10(labels = percent_format())+
          scale_y_log10(labels= percent_format())+
          scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+
          facet_wrap(~author, ncol=2)+
          theme(legend.position = "none")+
          labs(y= "United States", x=NULL)

##########################################
######### doing the cor.test() ###########
```

```
###########################################

cor.test(data=frequency[frequency$author == "Brazil",],
         ~proportion + `United States`)

cor.test(data=frequency[frequency$author == "Spain",],
         ~proportion + `United States`)

###########################
#########Gini Tree#########
###########################

review <- my_airbnb_join %>%
     select(review_scores_accuracy, review_scores_cleanliness, review_scores_checkin,
review_scores_communication, review_scores_location, review_scores_rating, review_scores_value)

clean_review <- na.omit(review)
sample_clean_review <- clean_review[sample(nrow(clean_review), 10000), ]

sample_clean_review$binary <- c()
for(i in 1:nrow(sample_clean_review)){
  if (sample_clean_review$review_scores_rating[i] > 90) {
    sample_clean_review$binary[i] <- 1
  } else {
  sample_clean_review$binary[i] <- 0
  }
}

my_tree <-
rpart(binary~review_scores_accuracy+review_scores_cleanliness+review_scores_checkin+review_scores
_communication+review_scores_location+review_scores_value, data=sample_clean_review, method =
"class",
             cp = 0.02)
rpart.plot(my_tree, type=1, extra=1)

my_tree <- rpart(binary ~
review_scores_accuracy+review_scores_cleanliness+review_scores_checkin+review_scores_communicat
ion+review_scores_location+review_scores_value,
             data=sample_clean_review, method="class",
             control = rpart.control(minsplit = 20,
                                     minbucket = 15,
                                     cp = 0.005))
rpart.plot(my_tree, type=2, extra=2)
```

```
#####################################################
######### Running LDA per reviews ###################
######################################################

sample_usa_dtm <- sample_usa_eng %>%
  unnest_tokens(word, comments) %>%
  anti_join(stop_words) %>%
  count(listing_url_5, word) %>%
  cast_dtm(listing_url_5, word, n)

usa_lda <- LDA(sample_usa_dtm, k=3, control = list(seed=123))

##########################################################
### Running LDA per token
##########################################################
usa_topics <- tidy(usa_lda, matrix="beta")

top_terms <- usa_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

#lets plot the term frequencies by topic
top_terms %>%
  mutate(term=reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend=FALSE) +
  facet_wrap(~topic, scales = "free") +
  coord_flip()
```