airbnb

DSBA 6211 Advanced Business Analytics

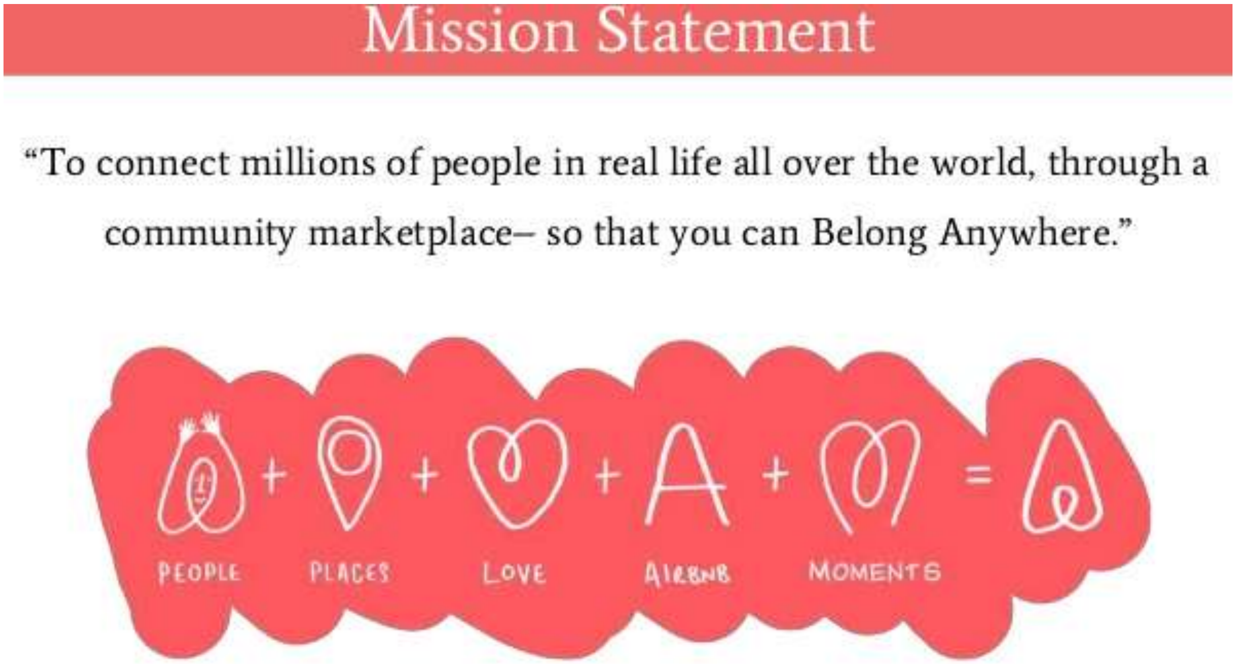
Team Members:
Jonathan Peters
Minglan Ye
Alvis Chung
Lilly Xiong
Dipin Kasana
Robert Weil

Aug 5th, 2020

Table of contents

Business Overview

Airbnb is a trusted community marketplace for people to list, discover, and book unique accommodations around the world. Through its platform, people connect to unique travel experiences, at any price point, in 191 countries and over 65,000 cities worldwide. It's mission is to help create a world where you can belong anywhere and where people can live in a place, instead of just traveling to it.[1]



Mission Statement

"To connect millions of people in real life all over the world, through a community marketplace– so that you can Belong Anywhere."

Figure 1: Airbnb's mission statement

The Airbnb platform is the center of the business. It allows users to search through listings and filter down to the based on various attributes getting as specific as the type of home they are looking to rent out. The founders have recently expanded their business to include event planning services, Airbnb plus which includes better accommodations, and experiences that can be booked as part of the vacation rental. Despite these innovations, the majority of the business still relies on hosts renting out properties to guests.

Airbnb has a unique business model; they operate in the lodging marketplace but do not own any properties. They work as an intermediary between real estate owners that want to rent space or rooms in their homes and consumers who are looking for an alternative to traditional

[1] Airbnb company value https://www.comparably.com/companies/airbnb/mission#:~:text=Airbnb%20Mission%20Statement,of%20just%20traveling%20to%20it.

lodging such as hotels. Whether you are a homeowner or a renter, creating an account on Airbnb is free for all users. Once an account is created, a user will have access to all Airbnb listings, both domestic and international, and a host can easily list and manage their rentable space on the site. Airbnb's platform makes the booking process extremely user friendly for both the renter and the host, and this is one of the many reasons attributing to their success.

The Airbnb host is responsible for pricing their rental space on the platform, and Airbnb profits from an additional fee added to the listing. The consumer pricing of rentals varies based on customer preferences, such as location and amenities. Many customers find Airbnb prices to be relatively lower than traditional hotels, while offering a more exclusive lodging experience. Airbnb is now a household name used by millions of consumers around the world.

Purpose

Airbnb, like most businesses, has seen a serious decline in business since the outbreak of COVID-19 and many hosts are struggling to find renters. New York City has been hit especially hard by the outbreak and serves as a microcosm of the larger Airbnb market because of the large population, variety of listing types and prices. Our goal is to use the New York City data to provide insights into the extent of the downturn and the types of the listings most seriously affected.

Objective

The data encapsulates the first six months of 2020 in the New York City market focusing on listings, calendars, and reviews. The analysis will answer the following questions:

How does the COVID-19 pandemic affect NYC's Airbnb market? We will design and consider a comprehensive list of matrices, such as:

a. Market supply: the number of active listings
b. Market demand: occupancy rate, average monthly reviews, etc.
c. Customer comments: topics, sentiments, etc.

What factors have affected Airbnb hosts' market exit decisions?
What types of properties have been affected the most?

d. Propose a reasonable indicator(s) to quantify individual listings' performance

Research Problems

The data is a 6 months sampling of Airbnb New York City listing data. We will be using 4 datasets in our analysis that include listing, calendar, and review data for Airbnb in New York City and another for COVID-19 data in the city. Each dataset contains information that will enable us to thoroughly examine the Airbnb strategy as a whole. We will be using R studio for our project as well as Tableau for data visualization and python for data cleaning.

The source of our data is insideairbnb.com which sources all its data from the information made publicly available by Airbnb. However, Inside Airbnb is an organization that is clearly aligned against Airbnb's current policies and is working with this data to provide analysis in an attempt to further the cause of enhanced regulation on short term housing. The data itself, aside from the presentation and source, has potential challenges as well. The location of the listings is made inaccurate in an attempt to keep listings anonymous until they are booked. This means actual locations can be up to 450 feet from the listed area. This is not a huge discrepancy but in an area like Manhattan where there are many listings in the same building and blocks are tightly packed. With each listing being individually anonymized, this can skew the data in each neighborhood.

The actual number of bookings is not provided by Airbnb, so we are forced to use the assumption made by the San Francisco model to best estimate this figure by using a review rate assumption. The Budget and Legislative Analyst Office and the CEO of Airbnb use a review rate of 72% to convert review to bookings, but our data source, Inside Airbnb, found a rate of 30.5% to be more accurate, but did not account for reviews from deleted listings, so the less conservative rate of 50% was determined to be most accurate.

Data Description

Data Tables	Records
listingfinal.csv	303,408
calendar.csv	18,747,307
reviews2020.csv	1,048,576
coviddata.csv	123

listingfinal.csv		
Variables	**Data Types**	**Definition**
Id	Factor	Unique identifier for each Airbnb host
Last_scraped	Date	Latest extraction date of data
Name	Factor	Unique name for each Airbnb listing
Host_id	Factor	Unique identifier for each host
Host_since	Date	Date host listed the Airbnb on the website
Host_response_time	Factor	Timeframe the host responds
Host_response_rate	Factor	Rate when host responds
Host_acceptance_rate	Factor	Rate when host accepts
Host_is_superhost = binary (T/F)	Factor	1 indicates Airbnb is superhost, 0 is not
Host_neighbourhood	Factor	Neighborhood to each Airbnb host
Host_listings_count	num	Count of host listings
Host_total_listings_count	num	Count of host total listings
Host_verifications	Factor	Type of verification
Neighbourhood_group_cleansed	Factor	Neighborhood to Airbnb location
Property_type	Factor	Type of property
Room_type	Factor	Type of room
Accommodates	num	Number of guests to be accommodate
bathrooms	num	Number of bathrooms
bedrooms	num	Number of bedrooms
beds	num	Number of beds
bed_type	Factor	Type of bed
amenities	Factor	Type of features of a Airbnb
Price	num	Price for rent (in $)
Security_deposit	num	Security deposit (in $)
Cleaning_fee	num	Cleaning fees (in $)
Guests_included	int	Number of guests can be including
Extra_people	num	Extra charge on additional person (in $)
Minimum_nights	int	Shortest nights can stay
Maximum_nights	int	Longest nights can stay
Availbility_365	int	Number of availabilities in 365 days
Number_of_reviews	int	Number of reviews
Number_of_reviews_ltm	int	Number of reviews within the last twelve months
first_review	Date	Date of first review received on this Airbnb

last_review	Date	Date of last review received on this Airbnb
review_scores_rating	num	Number of review score rating (scale 1-10)
review_score_accuracy	num	Number of review score accuracy (scale 1-10)
review_score_cleanliness	num	Number of review score cleanliness (scale 1-10)
review_score_checkin	num	Number of review score checking (scale 1-10)
review_score_communication	num	Number of review score communication (scale 1-10)
review_score_location	num	Number of review score location (scale 1-10)
review_score_value	num	Number of review score value (scale 1-10)
calculate_host_listings_count	int	Number of total host listing counts
calculate_host_listings_count_entire_homes	int	Number of total host listing counts (entire)
calculate_host_listings_count-private_rooms	int	Number of total host listing counts (private)
calculate_host_listings_count_shared_rooms	int	Number of total host listing counts (shared)
reviews_per_month	num	Number of reviews per month (in %)
Month	int	Month extracted from last_scraped date
Day	int	Date extracted from last_scraped date
Latitude	num	Latitude
Longitude	num	Longitude

calendar.csv		
Variables	**Data Types**	**Definition**
Listing_Id	Factor	Unique identifier for each listing
Date	Date	Latest extraction date of data
Available	Factor	Total reserves (for any open claims)
Price	num	Price for Rent ($ per day)
Adjusted_Price	num	Price for Rent subject to surge
Minimum_Nights	int	shortest nights can stay
Maximum_Nights	int	longest nights can stay

reviews2020.csv		
Variables	**Data Types**	**Definition**
listing_id	Factor	Unique identifier for each Airbnb place
Id	Factor	Unique identifier for each Airbnb host
Date	Date	Date comment was posted
Reviewer_id	Factor	Unique identifier of each Airbnb guest
Reviewer_name	Factor	Name of Airbnb guest
comments	Factor	Comments/reviews of the place made by the guest

coviddata.csv		
Variables	**Data Types**	**Definition**
DATE_OF_INTEREST	Date	Date of the data record
CASE COUNT	int	Number of cases count
HOSPITALIZED_COUNT	int	Number of hospitalized counts
DEATH_COUNT	int	Number of death count

Analysis Methods and Results

On January 21st, the CDC confirmed the first case of coronavirus in the US.[2] The case was a Washington resident who had visited Wuhan and returned to the United States after contracting the virus. The virus eventually would flourish and quickly spread to New York. By March 1st, New York had its first case and the state began contract tracing. Within the month, Governor Cuomo had shut down all nonessential business and suspended all non-elective surgeries to free up hospital space.[3] This action essentially shut down the state and decimated the tourist and lodging industry. Our analysis began by focusing on the change in inventory and available listings in the city.

[2] https://www.ajmc.com/focus-of-the-week/a-timeline-of-covid19-developments-in-2020///?p=1

[3] https://abcnews.go.com/US/News/timeline-100-days-york-gov-andrew-cuomos-covid/story?id=71292880

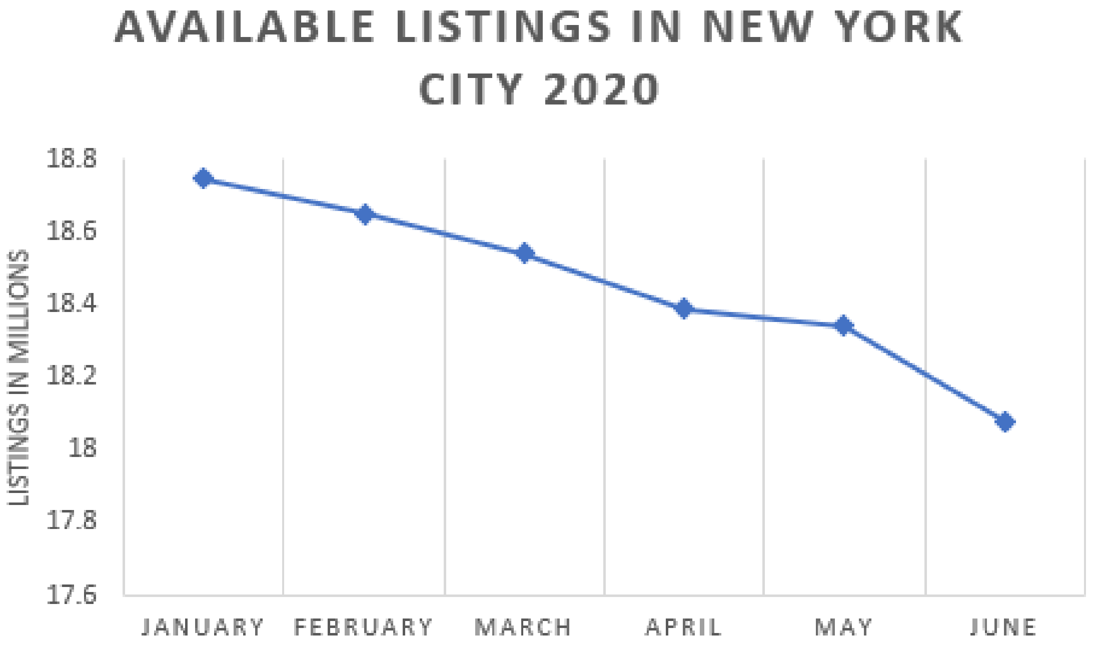


Figure 2: Active listings in New York City since January 2020.

The above graph shows the number of active listings in the New York City area for the rest of the year at the start of each month. This decrease is in all likelihood a result of listings being removed, an indication that hosts are exiting the market or finding a new platform to use for their properties. However, this is not necessarily the case, because the data does not differentiate between bookings and removed listings. Therefore, to really get a look at a business and the potential impact of COVID-19, we need a better indicator of bookings numbers. For that, we used reviews as a way to estimate the bookings numbers.

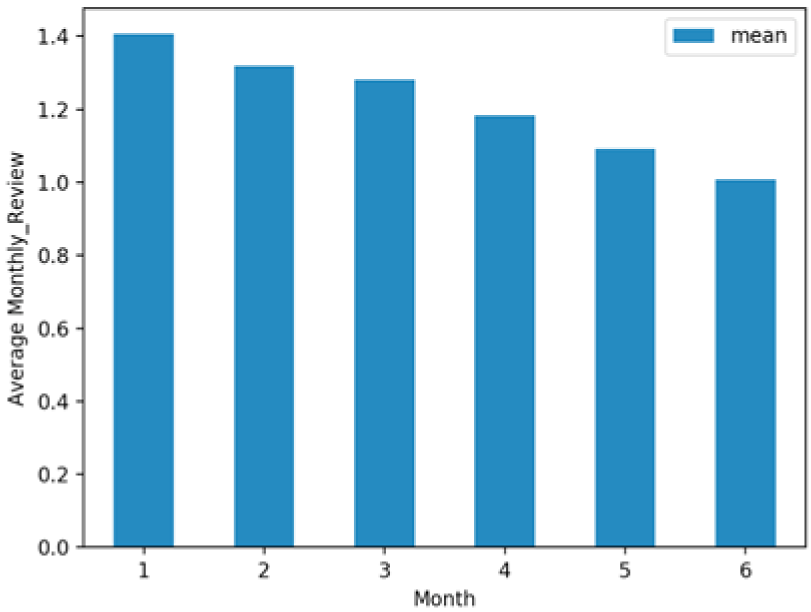


Figure 3: Average number of monthly reviews per listing for the past six months.

In the above chart, we can observe the average monthly review from January to June and see there is a significant drop of reviews every month. By using the assumption made by Inside Airbnb, we can develop a clear picture of the bookings decrease. The most likely reason is that customers are concerned about the COVID-19 pandemic, many of them have cancelled their bookings due to this concern and have received full refunds. This fear does not only affect customers but also extends to hosts. Despite the fact that many hosts no longer live in the properties they own, they still fear inviting guests into their properties because of the potential spread of the virus.

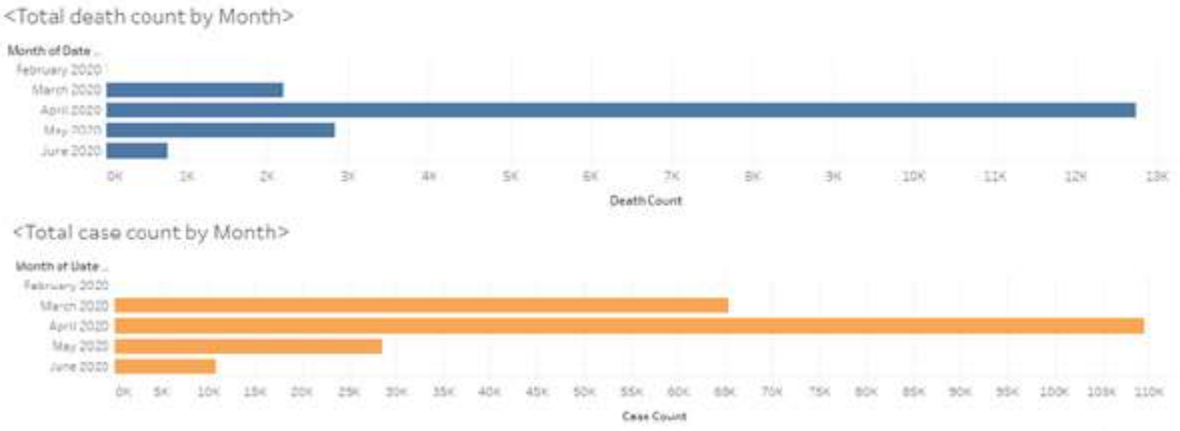


Figure 4: Distribution of the number of COVID-19 cases and deaths for NYC in 2020.

What we see here is the COVID-19 cases and deaths across New York City since the first case was seen at the end of February until the end of June. There is almost a perfect sync of dates between the outbreak and the decrease in listings and reviews of Airbnb. In March, we observed a total of 65,263 cases and 2,194 deaths within 1 month of the first case. Apparently, we can understand that there is a tremendous drop of reviews between January to March and April to June.[4] Based on the data, some hosts even considered exiting due to high operating costs with the properties especially with enhanced cleaning procedures. Some may turn the properties into long-term rentals instead where the income is stable and there is less risk of outbreak.

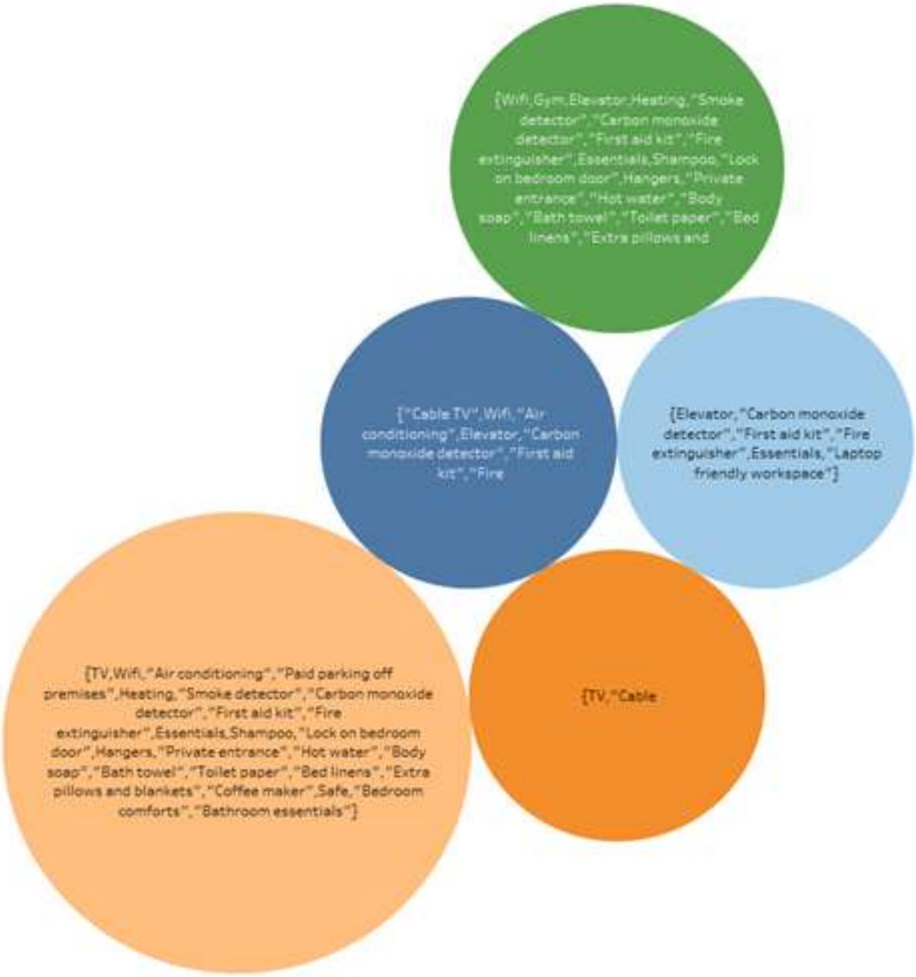


Figure 5: Top five amenities and features for listings

We then observed the top 5 amenities groups and features based on the average pricing per listing. We can see that the most popular amenities were TV/cable, WIFI, heating, smoke detector, carbon monoxide detector, and fire extinguishers. These show that customers are more willing to spend on listings with these features based on this pattern. As a result, hosts

[4] https://www1.nyc.gov/site/doh/covid/covid-19-data.page

should continue to focus on adding or keeping these features to attract more customers to their listings. These amenities also give us insight into the decision-making process of prospective guests. Based on the features such as smoke and carbon monoxide detectors and fire extinguishers, it is clear safety has been a high priority to guests even before the pandemic. We can use this information to extrapolate that post-COVID safety will still be an important feature, but the amenities that focus on safety will just shift. We can imagine enhanced cleaning and sanitation moving to the top of these lists. These are areas forward thinking hosts should focus on if they hope for their business to recover.

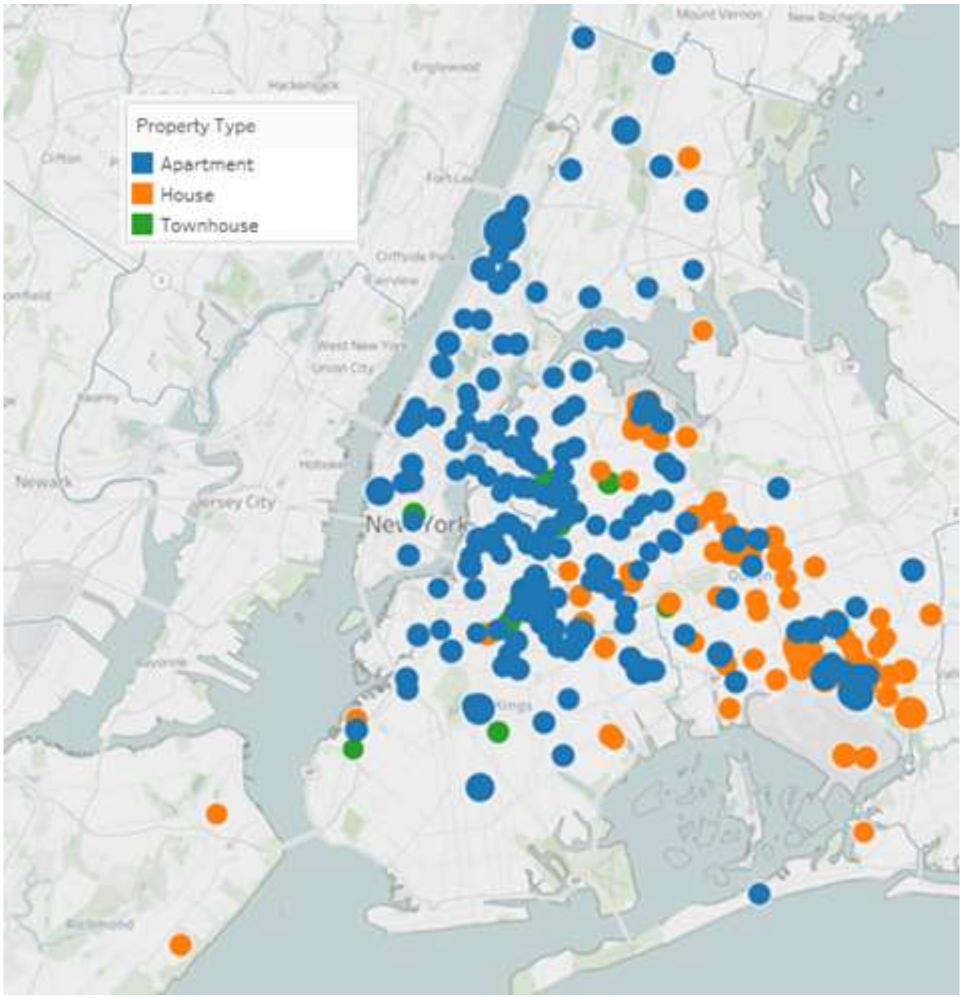


Figure 6: Distribution of different property types in NYC

The figure above shows almost uniformly across the city that “Apartment” is the most commonly booked type of listing, with houses being a close second farther out in the more suburban areas of Queens. These numbers are unsurprising because apartments were far and away the most common type of listing in the city. However, the discrepancy between house and townhouse is significant because the total number of houses and townhouses are not very different.

Below is a snapshot from an interactive map with listing counts over the New York City map and most listings are in the city center. This is similar to the above map in showing distribution, but it puts a number to it. You can utilize the zoom function in the map to find out the individual location of the listings as they are anonymized by Airbnb.

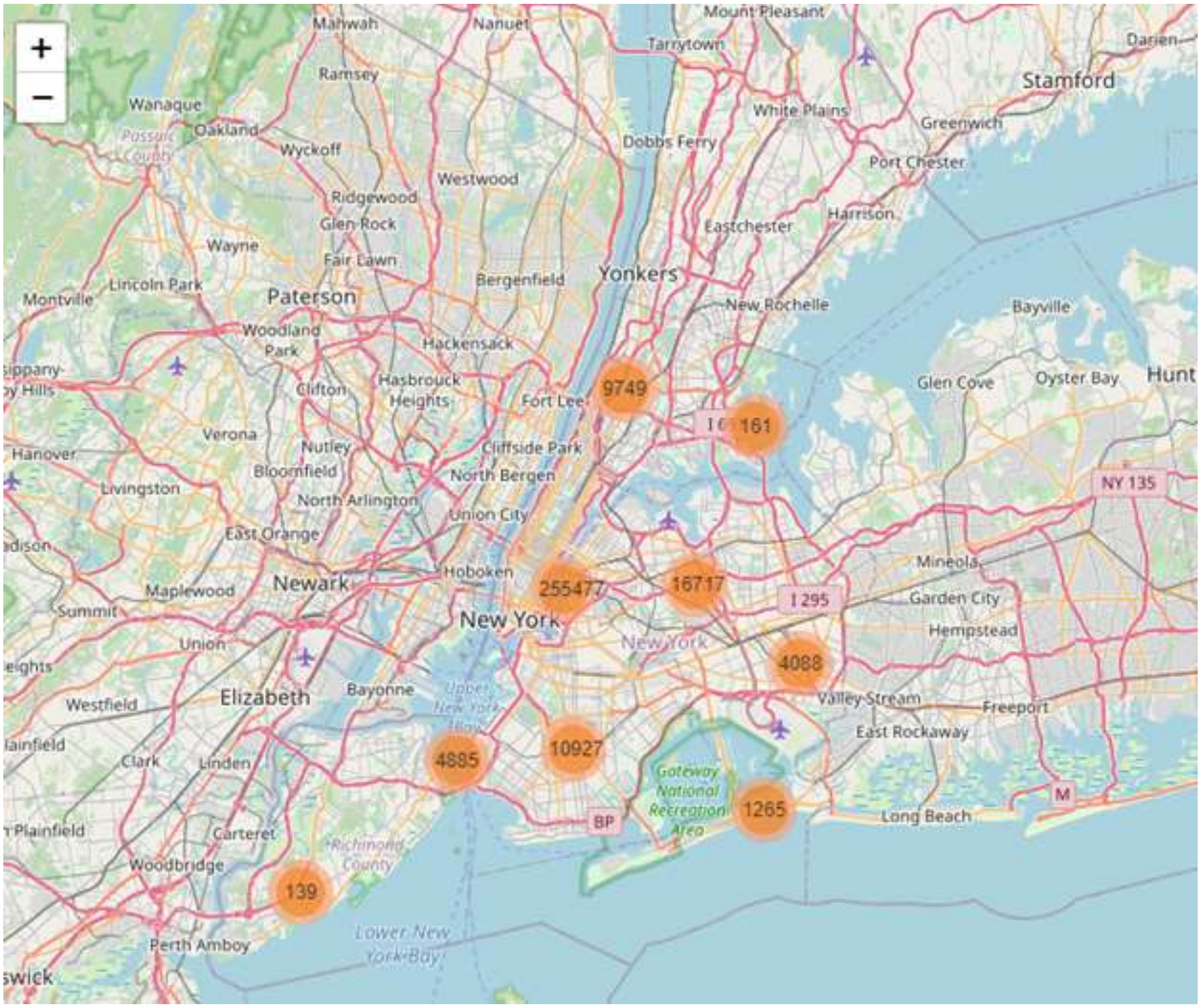


Figure 7: Numbers of Airbnb listing across different locations in NYC.

We then took a look at the listing numbers across the five boroughs of New York City and graphed them below. The neighborhood “Manhattan” has the highest number of listings among all other boroughs. Manhattan is the most densely populated area in New York City and is home to the majority of sightseeing and tourist destinations such as Wall Street, Broadway, and Central Park. As we can see from the below graph, Manhattan has over 134,000 unique listings on the Airbnb platform. The density of listings along with the sheer number of listings in Manhattan led us to hypothesize this would be the area most seriously impacted by COVID-19.

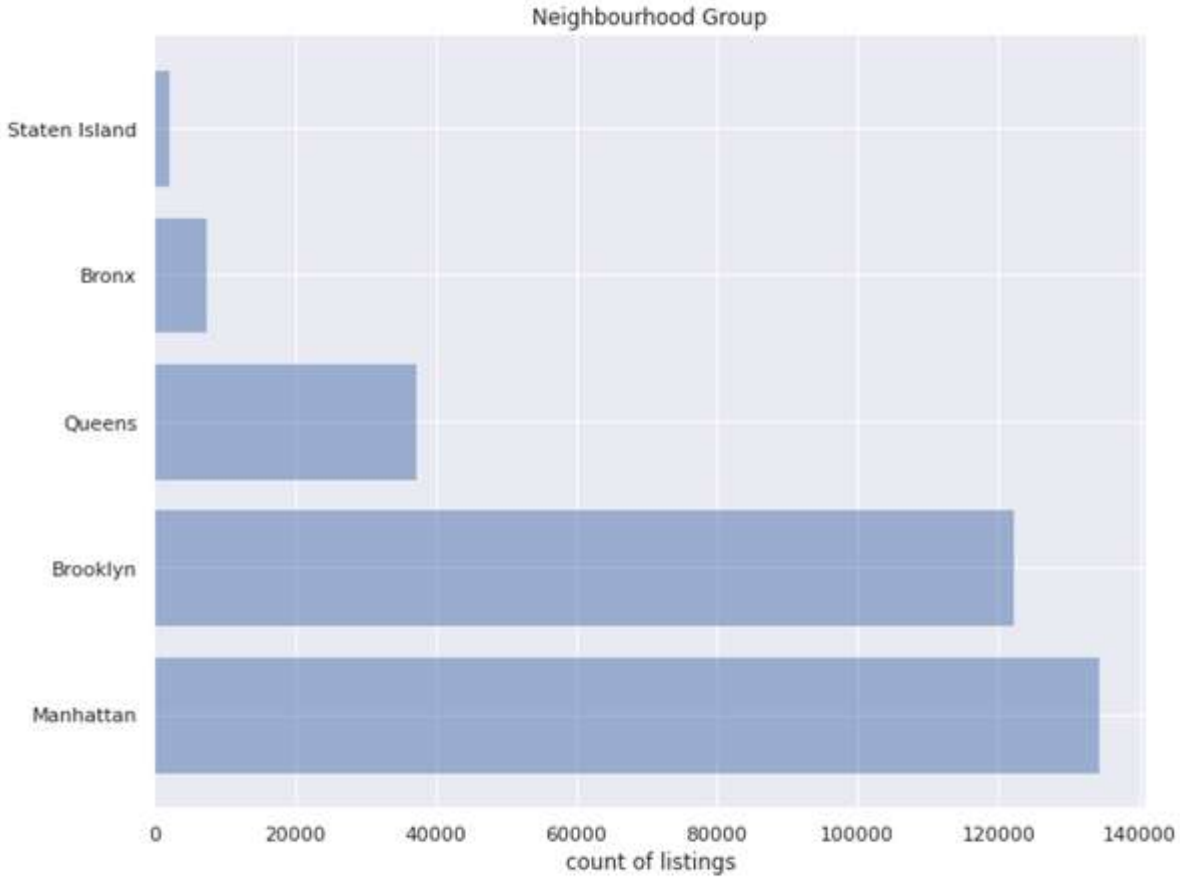


Figure 8: Number of listings in the five major neighborhoods in NYC

Our analysis then took us to look at the hosts who were listing these properties. We simply generated a top host graph to see which host has the highest listing. The below chart shows that host ID: 30283594 has over 500 listings which is almost 5 times more than the 20th of the host ID. It is our belief that these types of hosts, the ones with the largest number of listings, will be the ones least affected by COVID-19. With so many listings to insulate their business and such large holdings, they will probably be able to stay afloat, while smaller owners or owners who rent out their homes part time will likely remove listings. Hosts with his many listings have turned, more or less, gig work into what is essentially a hotel and full-time business.

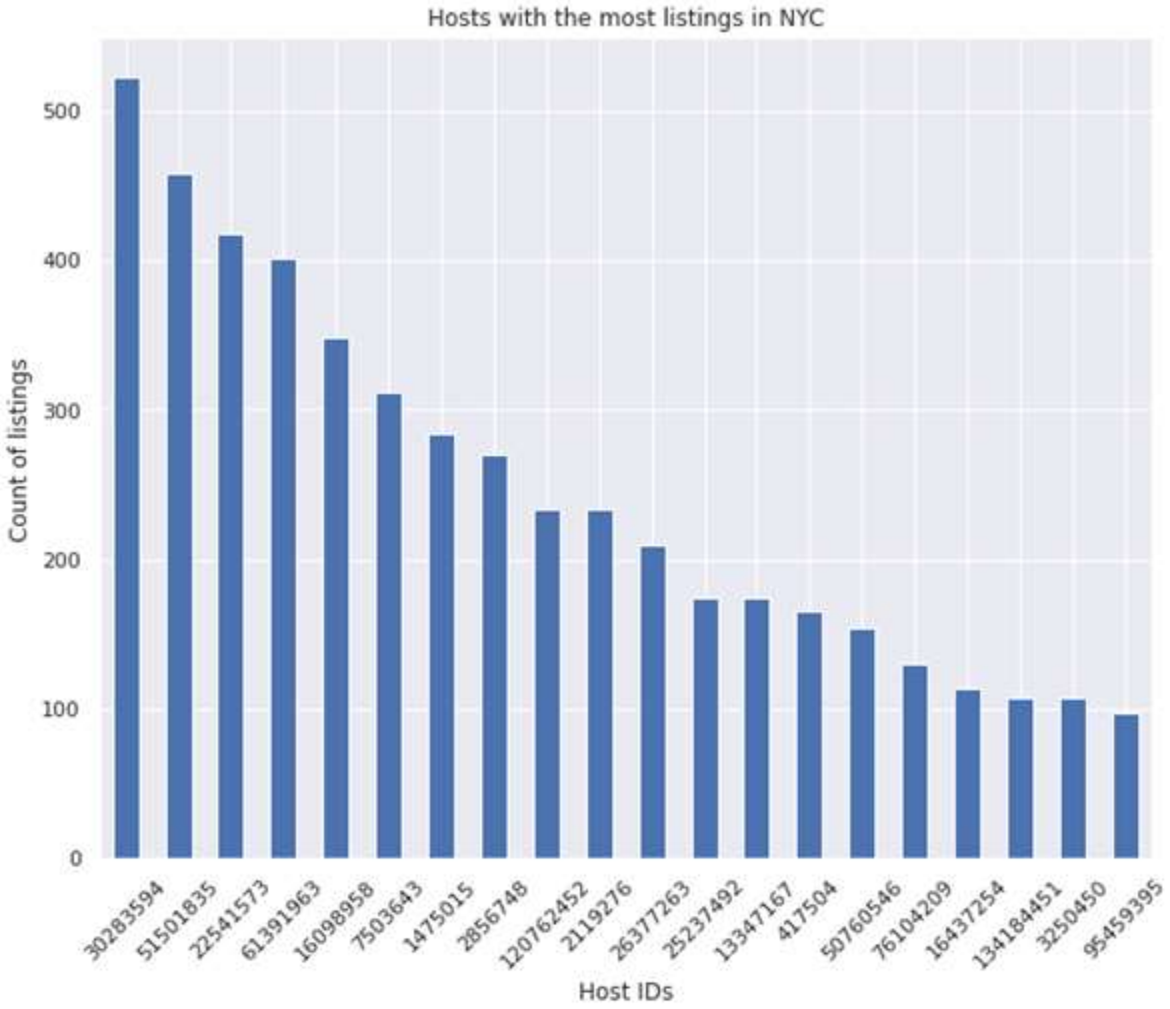


Figure 9: Distribution of hosts with the highest number of listings in NYC

We looked up one of the host properties and below are the pictures of the Airbnb. It is located in the center of Manhattan, which provides easy access for tourists or those on business in the city and as you can see below it is clearly decorated and appointed as if it is a hotel room.

[5]

Figure 10: Screenshot of listing from one of the hosts with the greatest number of listings in NYC

[5] https://www.airbnb.com/rooms/20845031?source_impression_id=p3_1596287348_sQExDhDH6qF5lO41

We then decided to look at price to see the impact of COVID-19 on the NYC Airbnb market. We began by plotting the price distribution graph and average price graph to see if the pandemic has any negative impact on the listing price. Surprisingly, the price for each listing maintained at a similar level for the past 6 months. From the graph below, we can see that hosts haven't adjusted any prices in response to the pandemic. This, however, does not mean there are still bookings at this price.

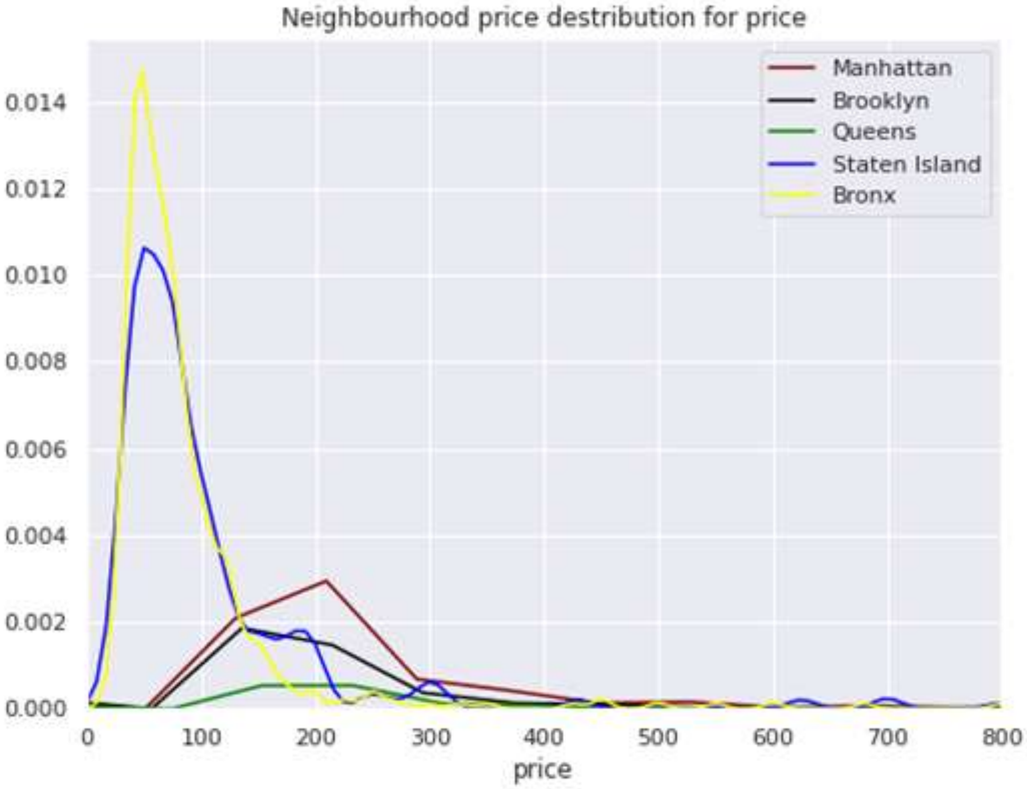


Figure 11: Variation in the listing price among the five major neighborhoods in NYC.

From the figure above, we saw the distribution of the price for each neighborhood as highly skewed. We decided to use 500 as our cut off point for the price and remove listings higher than this number to try and normalize the data and remove the skew of these listings. In the line plot below, we can see that even this cut off is not enough to affect the highly skewed boroughs of Bronx and Staten Island where the majority of prices are well below those of Manhattan, Brooklyn, and Queens.

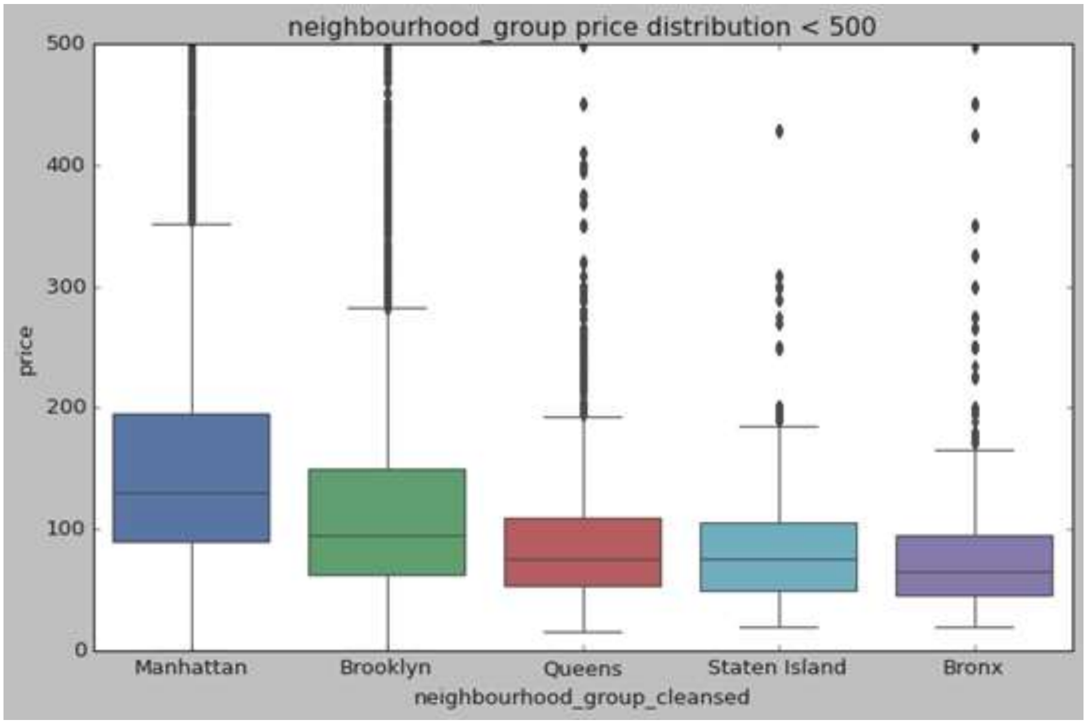


Figure 12: Box plot for variation in listing price among the five major neighborhoods in NYC

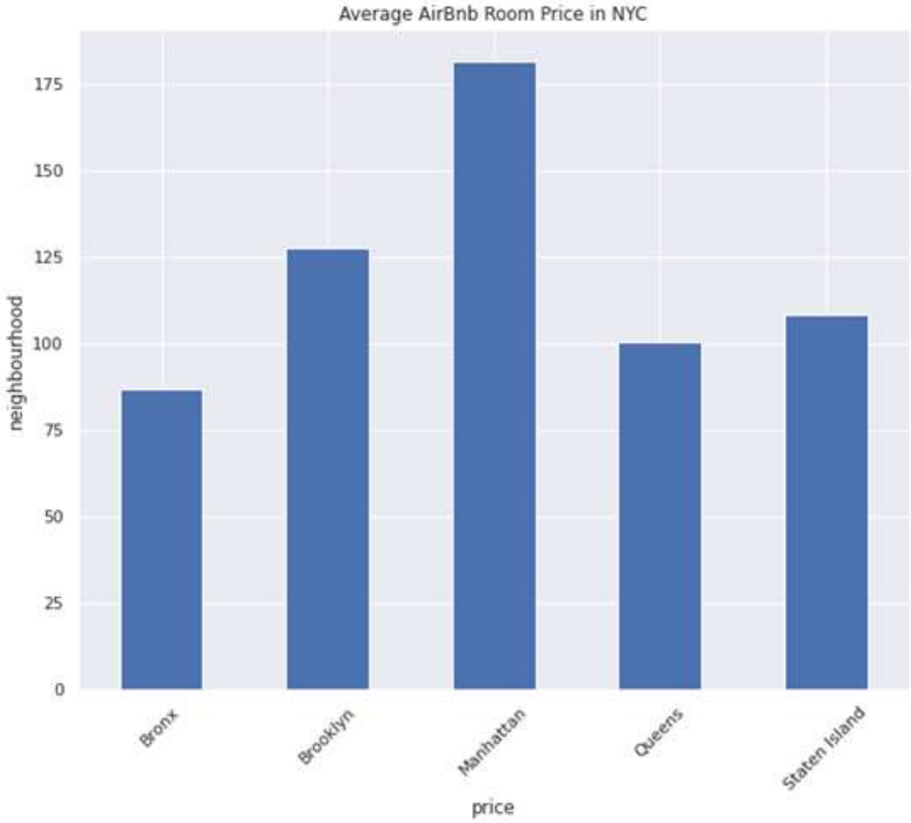


Figure 13: Average listing price for the five major neighborhoods in NYC

We then looked at average price per borough comparatively and it shows that Manhattan has the highest average night rate compared to other neighborhoods.

After exploring the average price, we decided to plot a graph to see if COVID-19 has had any impact on the cleaning fee. We assumed it would due to the increased risk to individuals cleaning the home being exposed to the virus, and the increased need for enhanced due diligence in this service. Interestingly, we found that the price of cleaning fees has maintained the same level across the boroughs, but we can see overall, the cleaning fee has been on an upward trend since January.

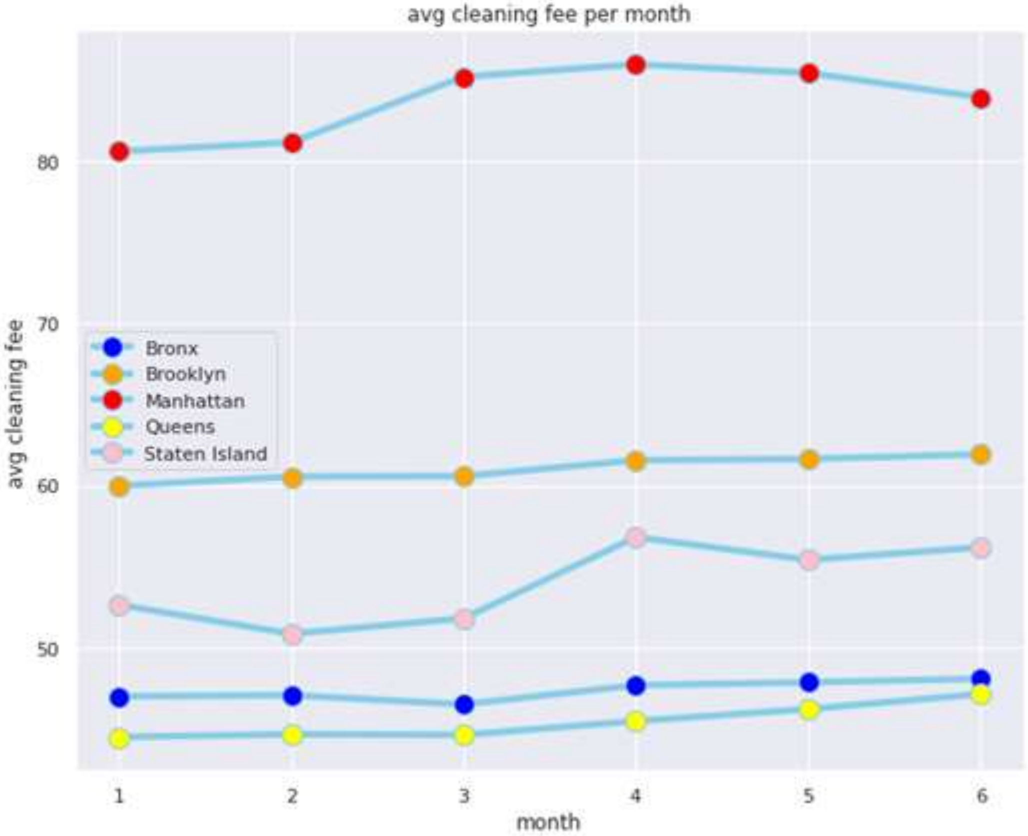


Figure 14: Variation in average monthly cleaning fee among the five neighborhoods in NYC

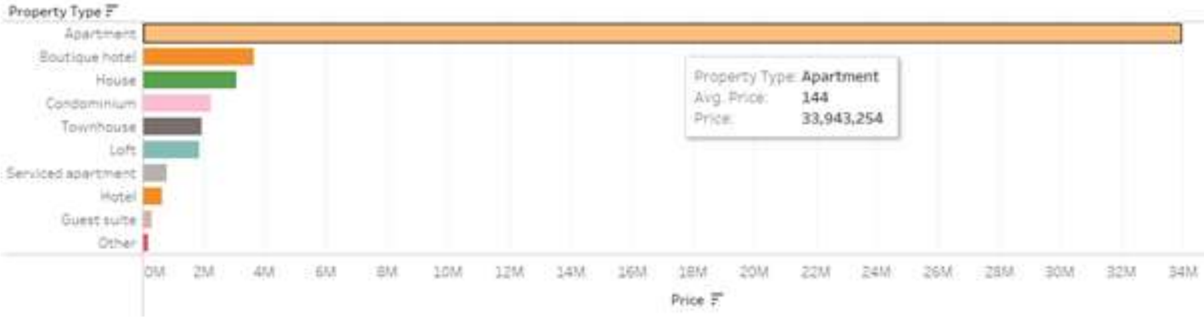


Figure 15: Distribution of different property types in NYC

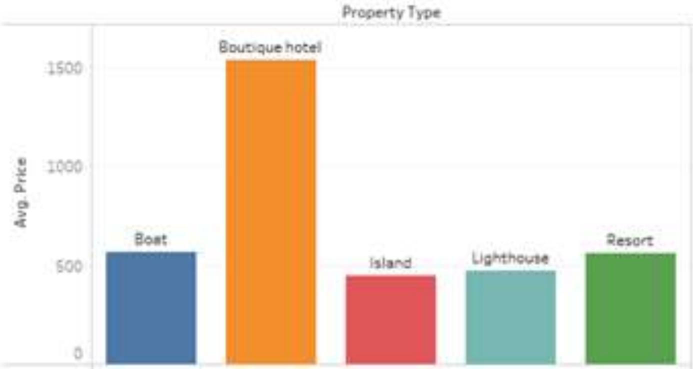


Figure 16: Average price for other unique property types offered by Airbnb in NYC

As we look at the property types, we see that the top property type of listing is apartment type as it exceeds over 70% of total sales compared to the other types offered. During this pandemic period, Airbnb should also market more listings into apartment types and boutique hotels based on popularity and pricing to reduce loss in the following months.

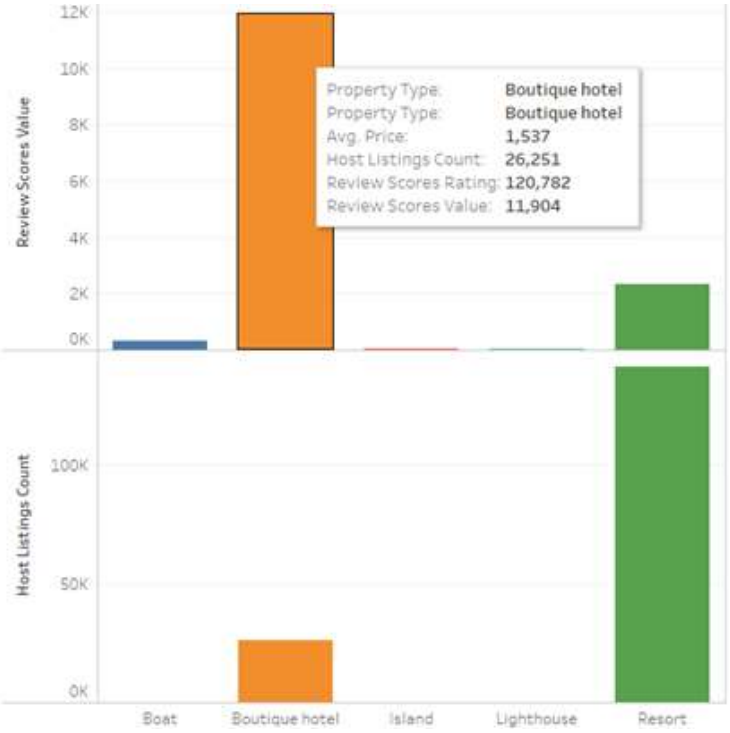


Figure 17: Number of listings vs total review score for unique properties

It is very interesting when we look further ahead, it seems like that boutique hotel has the highest score rating but it turns out that the listings for boutique is only 20% the number relative to listings classified as resorts. We then pulled the number of reviews per property type and came up with the map below.

Figure 18 helps to examine the impact of COVID-19 on the performance of the Airbnb market through analyzing the number of last reviews recorded for listings within the last five months. Based on the number of last reviews submitted, a significant decline in reservations was observed from mid-March, weeks after the first case of COVID-19 was detected in New York City.

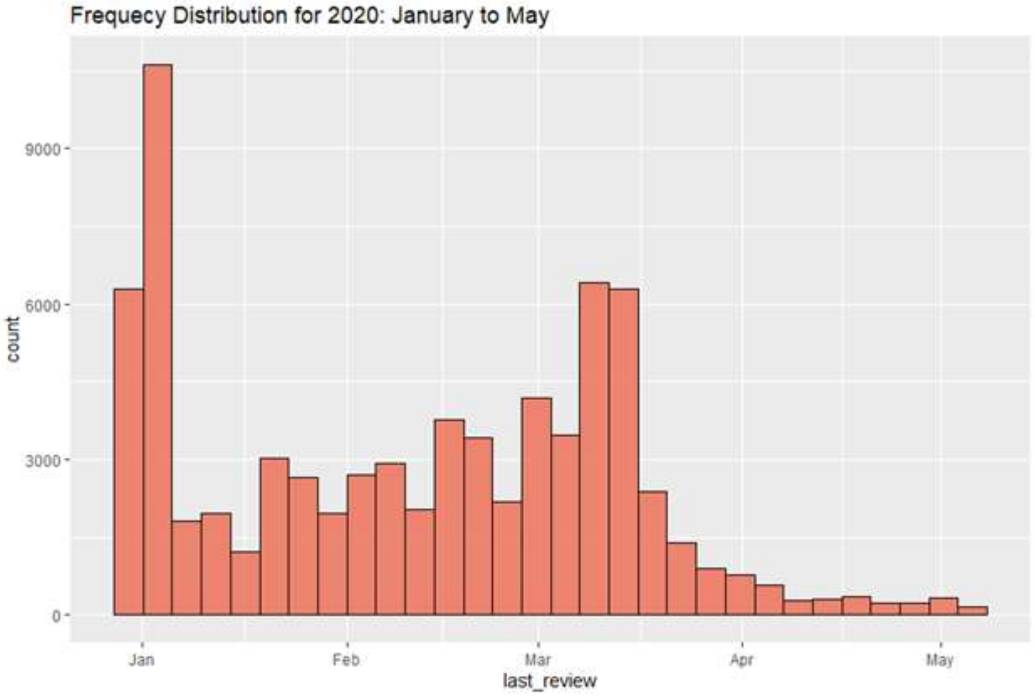


Figure 18: Distribution of the number of last reviews recorded in the past six months.

Figure 19 replicates the findings from Figure 18, however, the results are broken down for each of the five major neighborhoods in NYC. The results show a similar distribution in Brooklyn, Manhattan, and Queens, where significant data was available for visualization.

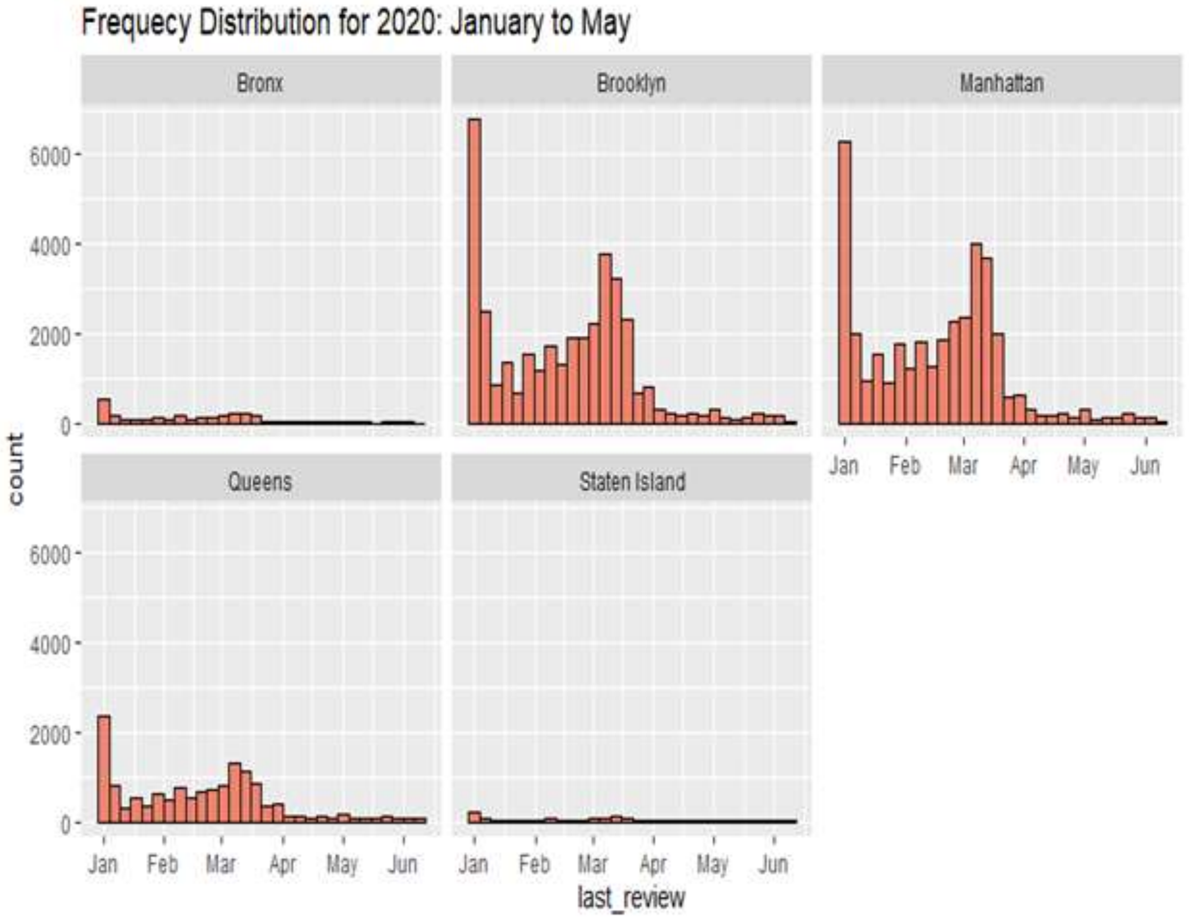


Figure 19: Distribution of the number of last reviews in the past six months for five major NYC neighborhood

Time Series Analysis:

The outbreak of COVID-19 did not spare most of the businesses in New York City (NYC), which was hit hardest among cities in the United States. The time series analysis of the Airbnb market in NYC was aimed to explore the historic performance of the market prior to the pandemic (officially announced on March 11th, 2020), and predict the market performance for the near future through various forecasting models using RStudio. While the dataset did not include details on occupancy rate or booking rate, the team decided to use the number of reviews parameter as an indicator for booking rate. As per the Airbnb website, the number of bookings is approximately twice the number of reviews which made time series analysis a suitable predictor.

Figure 20 investigates the seasonality, trend, and random noise generated over the last five years, since January 2015, where the number of weekly reviews were analyzed for the time series. The first panel in Figure 20 provides the distribution of the actual observed reviews per week over the last five years. The second panel (trend) indicates the trend of bookings, where a peak was recorded during the summer of 2018. As expected, seasonality can be seen in the time series with bookings soaring during the summer seasons. Finally, the random noise due to unknown events was recorded in the final panel. The fluctuation in booking rate (number of reviews) can be compared across y-axis, which indicates the estimated increase or decrease in bookings over time due to trend, seasonality, and random events.

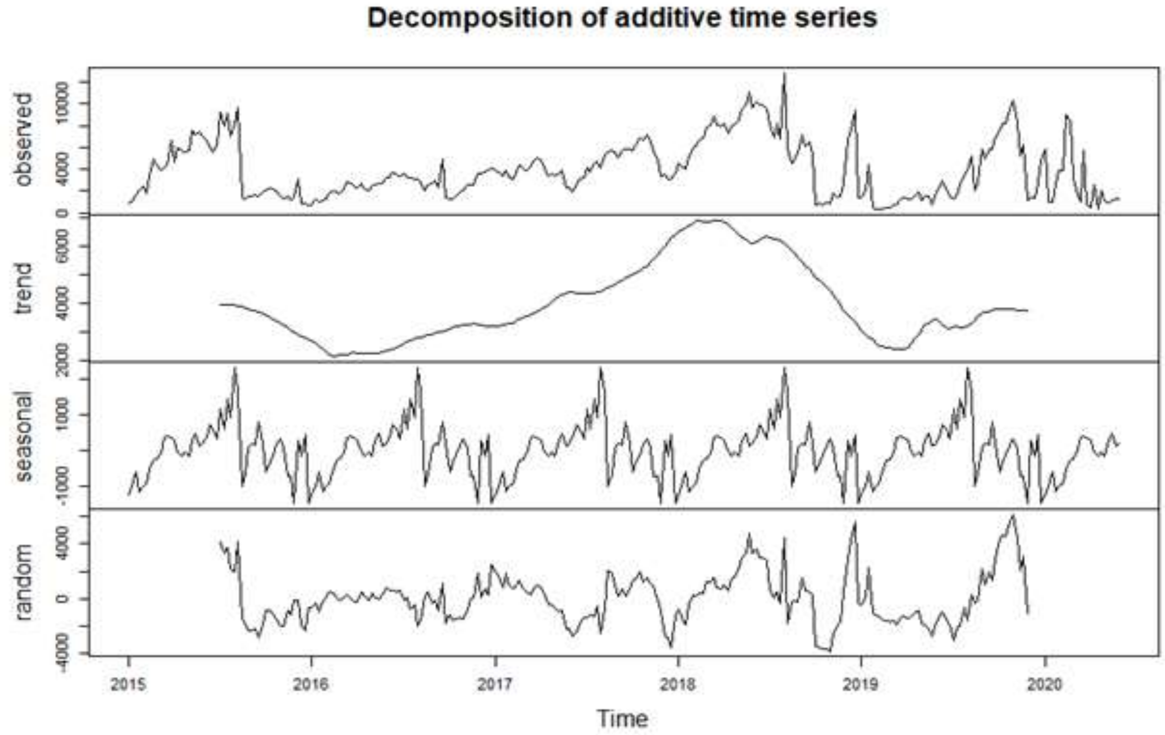


Figure 20: Exploring time series to visualize trend, seasonality, and random noise in the data.

Since the forecasting models were aimed to study the impact of COVID-19 on the Airbnb market performance in NYC, the team developed a robust but shorter time series to record the daily booking information (total daily reviews) between January 2019 and May 2020. The WHO characterized COVID-19 as pandemic on March 11th, 2020, hence, the cut-off for partitioning the time series data to test the forecasting model and validate the predicted results were considered after the announcement date. The team developed various linear forecasting models, using different combinations of seasonal and trend components, and ARIMA (Auto Regressive Integrated Moving Average) forecasting model. Among all the combinations developed for the linear forecasting model, the quadratic linear model without seasonality component recorded the best performance with Mean Absolute Percentage Error (MAPE) of 157.1203. Figure 21 shows the forecasting results for the quadratic linear model, where the dotted blue line indicates the forecasted bookings for the months of April and May 2020 against the actual observed bookings in black.

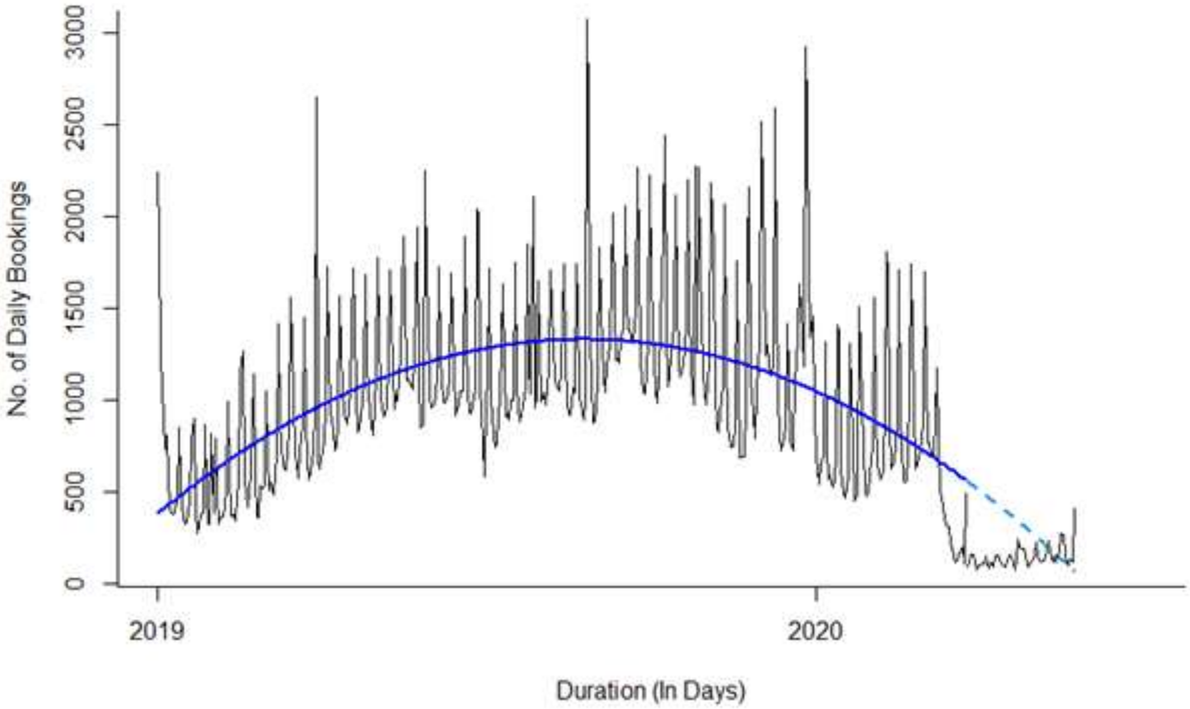
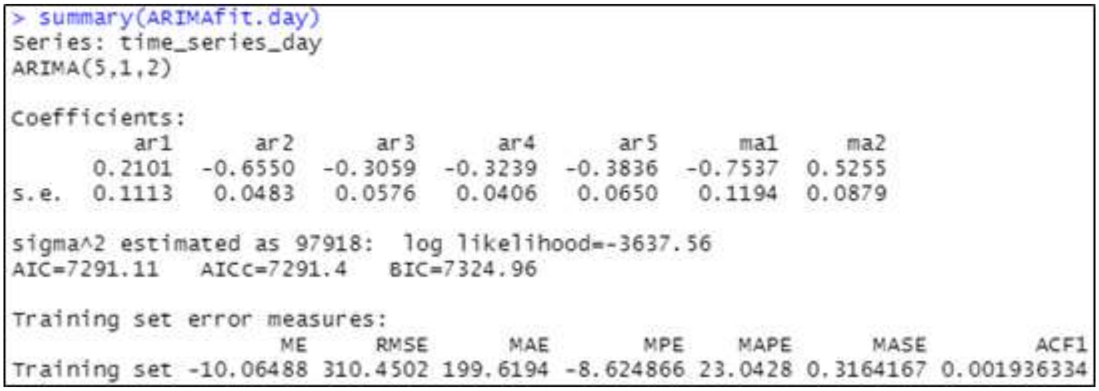


Figure 21: Quadratic Linear Model forecasting for April and May 2020.

The ARIMA forecasting model was developed on the same time series used to develop the quadratic linear model, but the predictive accuracy of the ARIMA model was far superior than its counterpart. As shown in Figure 22, the ARIMA model recorded MAPE value of 23.0428, much lower than the quadratic linear model. The coefficients of the ARIMA model are also summarized in Figure 22, where ARIMA (5,1,2) was considered as the best forecasting model. Based on the identified optimal model, the number of bookings can be predicted by using autoregressive terms of order five (p), single order of integration (d), and second order of moving average (q), which is generally represented as ARIMA (p,q,d). Figure 23 shows the predicted results for the number of daily bookings in NYC for April and May 2020 (denoted in blue), using the ARIMA (5,1,2) model. The orange line indicates the total daily bookings at an upper confidence level of 95%.



```
> summary(ARIMAfit.day)
Series: time_series_day
ARIMA(5,1,2)

Coefficients:
         ar1      ar2      ar3      ar4      ar5      ma1     ma2
      0.2101  -0.6550  -0.3059  -0.3239  -0.3836  -0.7537  0.5255
s.e.  0.1113   0.0483   0.0576   0.0406   0.0650   0.1194  0.0879

sigma^2 estimated as 97918:  log likelihood=-3637.56
AIC=7291.11   AICc=7291.4   BIC=7324.96

Training set error measures:
                    ME     RMSE      MAE       MPE    MAPE      MASE        ACF1
Training set -10.06488 310.4502 199.6194 -8.624866 23.0428 0.3164167 0.001936334
```

Figure 22: Summary of ARIMA model

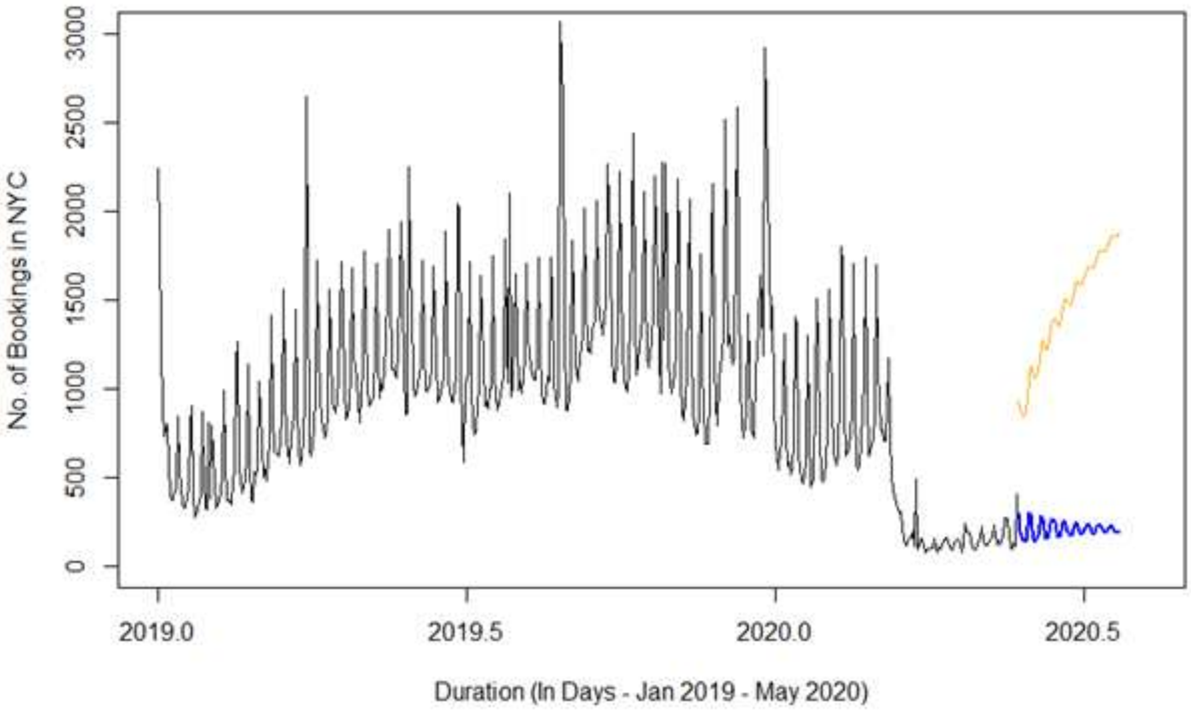
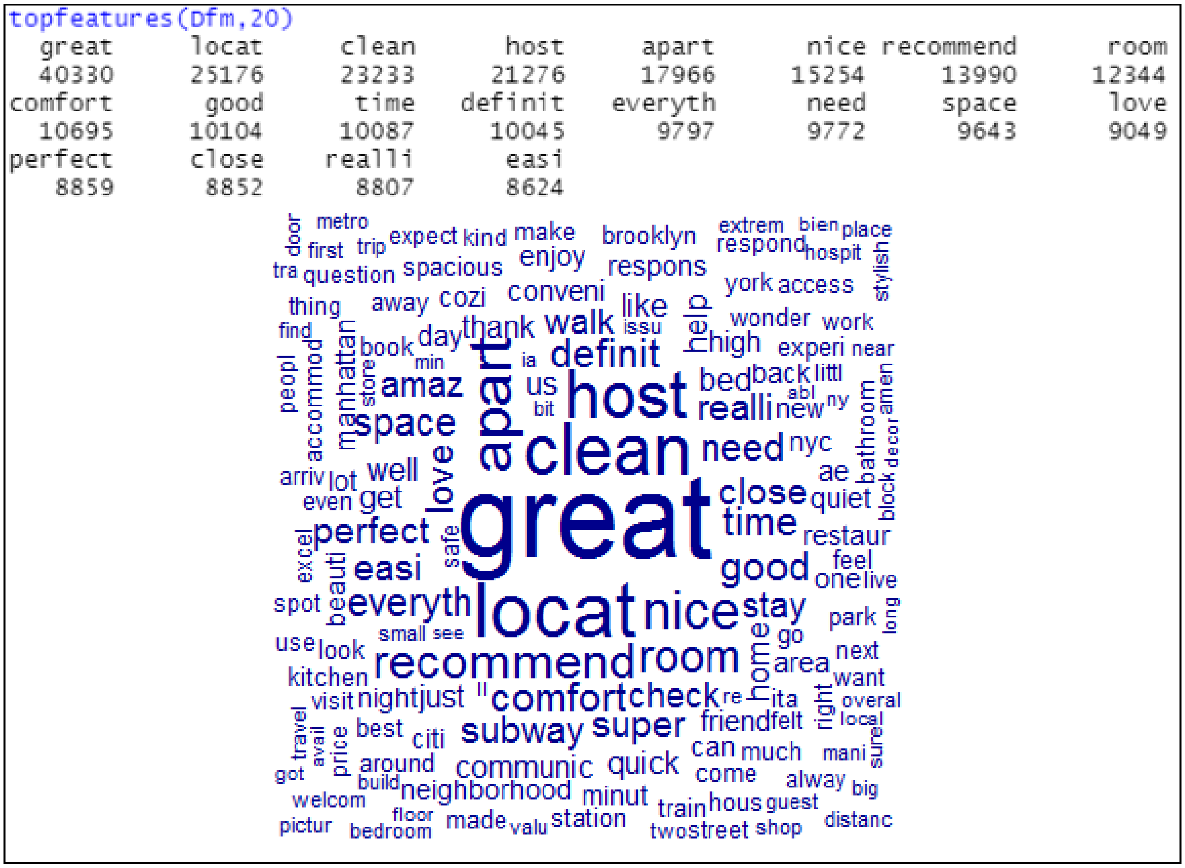


Figure 23: ARIMA model forecasting results for April and May 2020.

Topic Analysis

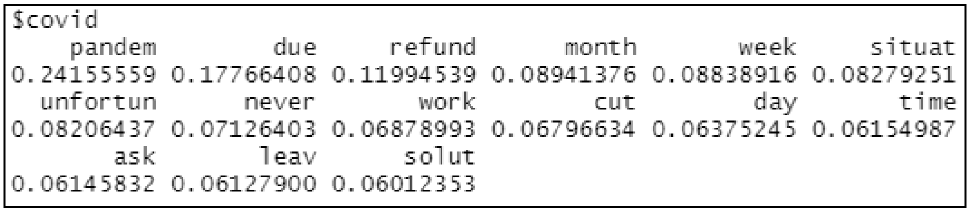
We performed topic modeling to see the changes in reviewer comments over the last few months since January. We wanted to see how guests' reviews are changing as the pandemic occurs. Below we generated the top 20 words and a word cloud from the overall dataset. Overall, guests seem to care about how great the place is, which is identified by the cleanliness, the location, room and space. Nowadays, we can see that the relationship that people care more about cleanliness because health and safety are the priority concern during the pandemic.



```
topfeatures(Dfm,20)
   great      locat      clean       host      apart       nice recommend       room
   40330      25176      23233      21276      17966      15254     13990      12344
comfort       good       time     definit    everyth       need     space       love
   10695      10104      10087      10045       9797       9772      9643       9049
perfect      close     realli       easi
    8859       8852       8807       8624
```

Figure 24: Topic modeling results for reviewer comments

Next, we check to see the relevance of covid in the comments. As shown below, the words such as pandemic, due, refund, month, week, situation, unfortunate, never, work, cut, day, time, ask, leave, and solution are related to the word covid. Moreover, it was interesting to see that the word refund was mentioned in the correlation table. This could be said there might be situations between customers and hosts who are experiencing a refund process during covid-19 pandemic.



```
$covid
    pandem        due     refund      month       week     situat
0.24155559 0.17766408 0.11994539 0.08941376 0.08838916 0.08279251
  unfortun      never       work        cut        day       time
0.08206437 0.07126403 0.06878993 0.06796634 0.06375245 0.06154987
       ask       leav      solut
0.06145832 0.06127900 0.06012353
```

Figure 25: Top terms identified most similar to the covid term

The topic modeling for each month shows that the customers reviews remain the same as the pandemic proceeds. Below confirms that the customers continue to care primarily about the cleanliness, the space and room and location of the place. Based on these topics, we can observe that cleanness doesn't affect much based on the comments from January to June. In other words, this could be said before pandemic cleanness was also used frequently as one of the factors to determine if the Airbnb listing is the preeminent property.

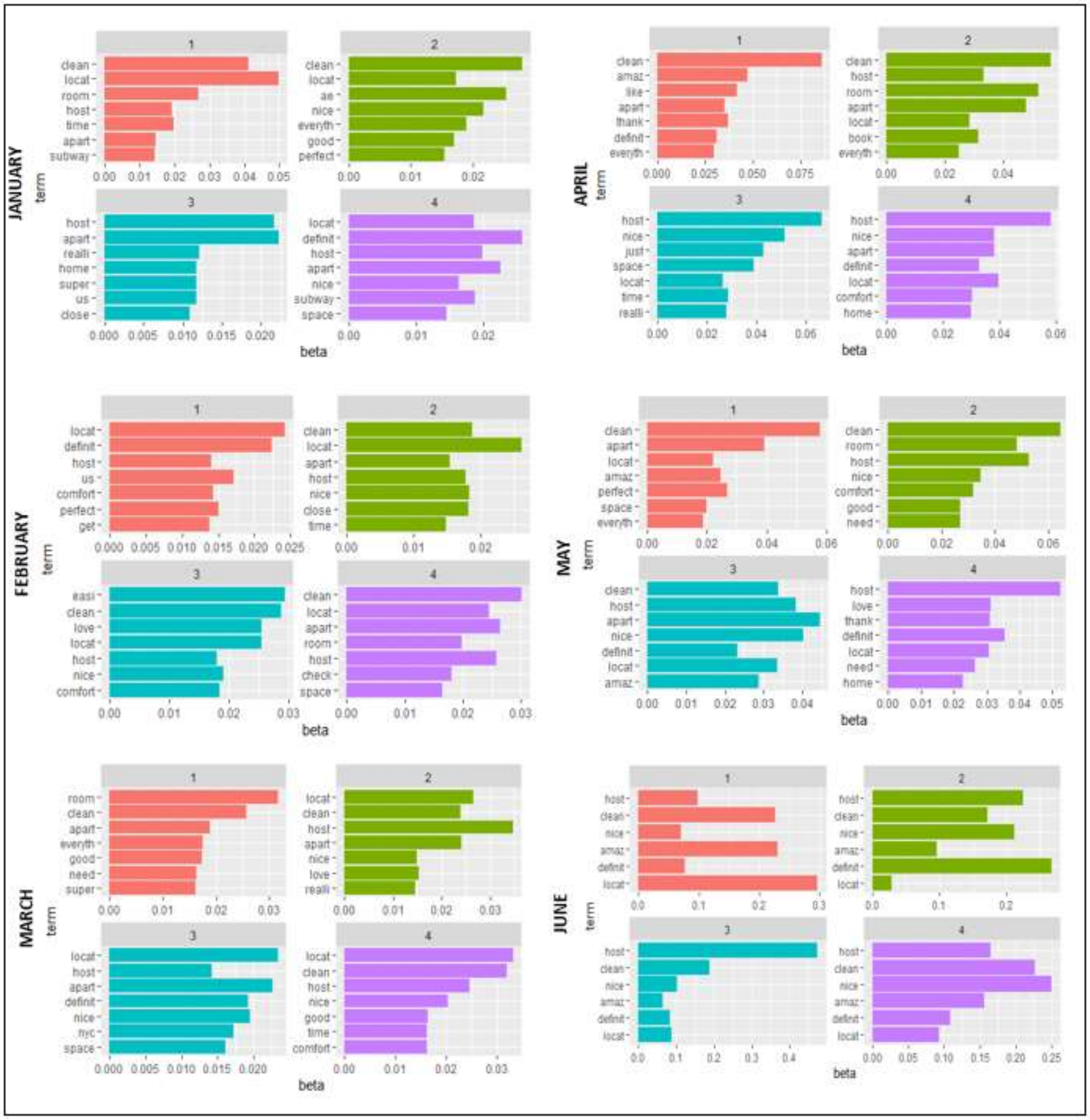


Figure 26: Topic modeling results broken down for the past six months.

Conclusion

In conclusion, we can see from the models on page 10 the Covid-19 pandemic has caused a drastic decline in the number of listings in New York from January 2020 to June 2020. Using the amount of reviews (pg. 11) as a guide, we can also see the demand is down in this area as well. We can see from our topics modeling (pg.28 and 30) of customer comments that cleanliness has become a priority for customers due to the risk of Covid-19. It also appears, based on the correlation model (pg. 29), many customers have been requesting refunds. This is more than likely due to the government shutdowns causing the cancellation of vacations.

When it comes to the question, “what factors have affected Airbnb hosts’ market exit decisions?” We believe the two main factors are lack of demand due to vacation cancellations, and increased operating cost due to extra cleaning procedures to combat the Covid-19 virus. This is reflected in the graph on page 10 and the word cloud on page 28 that shows an increase of the word “clean” from customer comments.

References

-https://www.digitaltrends.com/home/what-is-airbnb/

- **<u>Combined File in R</u>**

```
df <- read.csv("listings5.csv",na.strings=c('NA',''))
df1 <- read.csv("listings4.csv",na.strings=c('NA',''))
df2 <- read.csv("listings3.csv",na.strings=c('NA',''))
df3 <- read.csv("listings2.csv",na.strings=c('NA',''))
df4 <- read.csv("listings1.csv",na.strings=c('NA',''))
df5 <- read.csv("listings0.csv",na.strings=c('NA',''))
sum1 <- merge(df,df1, all=TRUE)
sum2 <- merge(sum1,df2, all=TRUE)
sum3 <- merge(sum2,df3, all=TRUE)
sum4 <- merge(sum3,df4, all=TRUE)
sum <- merge(sum4,df5, all=TRUE)
sum
summary(sum)
str(sum)
df <- read.csv("reviews5.csv",na.strings=c('NA',''))
df1 <- read.csv("reviews4.csv",na.strings=c('NA',''))
df2 <- read.csv("reviews3.csv",na.strings=c('NA',''))
df3 <- read.csv("reviews2.csv",na.strings=c('NA',''))
df4 <- read.csv("reviews1.csv",na.strings=c('NA',''))
df5 <- read.csv("reviews0.csv",na.strings=c('NA',''))
sum1 <- merge(df,df1, all=TRUE)
```

```
sum2 <- merge(sum1,df2, all=TRUE)
sum3 <- merge(sum2,df3, all=TRUE)
sum4 <- merge(sum3,df4, all=TRUE)
sum <- merge(sum4,df5, all=TRUE)
sum
summary(sum)
str(sum)
write.csv(sum,"~/DSBA 6211 project\\reviewsdata2020.csv", row.names = FALSE)
df1 <- read.csv("listingdata.csv",na.strings=c('NA',''))
df1 = subset(df1,select = -c(6,18:20,36:41))

## date
df1$last_scraped <- as.Date(df1$last_scrape,format="%Y-%m-%d")
df1$host_since <- as.Date(df1$host_since,format="%Y-%m-%d")
df1$first_review <- as.Date(df1$first_review,format="%Y-%m-%d")
df1$last_review <- as.Date(df1$last_review,format="%Y-%m-%d")

## Categorical
df1$id <- factor(df1$id)
df1$name <- factor(df1$name)
df1$host_id <- factor(df1$host_id)
df1$host_response_time <- factor(df1$host_response_time)
df1$host_verifications <- factor(df1$host_verifications)
df1$neighbourhood <- factor(df1$neighbourhood)
df1$city <- factor(df1$city)
df1$property_type <- factor(df1$property_type)
df1$room_type <- factor(df1$room_type)
df1$bed_type <- factor(df1$bed_type)
df1$amennities <- factor(df1$amenities)

## Integer
df1$security_deposit <- as.numeric(df1$security_deposit)
df1$guests_included <- as.numeric(df1$guests_included)

str(df1)
## Sent for ipynb for further data cleansing
## further drop unimportant
df2 <- read.csv("listing456.csv",na.strings=c('NA',''))
df2 = subset(df1,select = -c(15,16))
write.csv(df2,"~/DSBA 6211 project\\listing789.csv", row.names = FALSE)
df3 <- read.csv("listing789.csv",na.strings=c('NA',''))
df3 = subset(df3,select = -c(49,50))
write.csv(df3,"~/DSBA 6211 project\\listingfinal.csv", row.names = FALSE)
df4 <- read.csv("listingfinal.csv",na.strings=c('NA',''))
```

```
str(df4)

dfJan <- read.csv('Jancalendar.csv.gz')
summary(dfJan)
str(dfJan)
dfJan$available <- factor(dfJan$available)
dfJan$date <- ymd(dfJan$date)
count(dfJan,avaiable=2)
dfFeb <- read.csv('Febcalendar.csv.gz')
count(dfFeb,available=2)
dfMar <- read.csv('Marchcalendar.csv.gz')
count(dfMar, available=2)
dfApril <- read.csv('Aprilcalendar.csv.gz')
count(dfApril,available=2)
dfJune <- read.csv('Junecalendar.csv.gz')
count(dfJune,available=2)
dfMay <- read.csv('Maycalendar.csv.gz')
count(dfMay,available=2)
dfavail <- read.csv('availability.csv')
dfavail$Month <- month(dfavail$Month)
barchart(dfavail,
     main='Number of Available Listings Per Month',
     xlab='Month')
bc <- ggplot(data=dfavail)+
  geom_bar(mapping=aes(x=Month, y=Available.Listings),stat='identity')
```

- **Reviews Dataset Cleaning in R**

```
library(quanteda)
library(tidyverse)

#downloadind and reading the data
reviews <- read.csv('reviews.csv',stringsAsFactors = F)

#order by date
reviews <- reviews[order(reviews$date),]
summary(reviews)

#change data type for date variable to date
reviews$date <- as.Date(reviews$date)
str(reviews)

#Keeping all observations from 2020 to now
Reviews2020 <- reviews[(reviews$date> "2020-01-01" & reviews$date < "2020-06-30"),]
```

```
#cleaning the comments
Reviews2020$comments <- gsub(pattern = "\\W", replace =" ", Reviews2020$comments)
Reviews2020$comments <- gsub(pattern = "\\d", replace =" ", Reviews2020$comments)
Reviews2020$comments <- gsub(pattern = "\\W", replace =" ", Reviews2020$comments)
Reviews2020$comments <- gsub(pattern = "\\b[A-z]\\b{1}", replace =" ",
Reviews2020$comments)
stripWhitespace(Reviews2020$comments)
Reviews2020$comments

# remove non-english
library(stringi)
Reviews2020$comments <- stringi::stri_trans_general(Reviews2020$comments, "latin-ascii")

# creating subset
Reviews2020<-subset(Reviews2020, select = -c(reviewer_id, reviewer_name, id,listing_id))

#Exporting dataset
write.csv(Reviews2020,"C:/Users/Xou/Documents\\Reviews2020.csv", row.names = FALSE)
```

- **Python Notebook Data Cleaning**

https://colab.research.google.com/drive/11ZfwXs8exz7U9B3NBviK0E62xecgtHVM?authuser=1#scrollTo=pu29iT0v9EuR

- **Python**

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
df = pd.read_csv('/Users/alvis/Desktop/Project/listing789.csv', sep = ',' )
# df = df.groupby('Month').agg(
#     min=pd.NamedAgg(column='reviews_per_month', aggfunc=min),
#     mean=pd.NamedAgg(column='reviews_per_month', aggfunc=np.mean),
#     max=pd.NamedAgg(column='reviews_per_month', aggfunc=max),
# )
# df.plot.bar(rot=0)
# plt.ylabel('Average Monthly Reviews')
# plt.show()

## only take mean for considerations
df = df.groupby('Month').agg(
    mean=pd.NamedAgg(column='reviews_per_month', aggfunc=np.mean),
)
df.plot.bar(rot=0)
plt.ylabel('Average Monthly_Review')
plt.show()
```

```
df = df.groupby('Month').agg(
    max=pd.NamedAgg(column='reviews_per_month', aggfunc=max),
)
df.plot.bar(rot=0)
plt.ylabel('Maximum Monthly_Review')
plt.show()

df = df.groupby('Month').agg(
    min=pd.NamedAgg(column='reviews_per_month', aggfunc=min),
)
df.plot.bar(rot=0)
plt.ylabel('Minimum Monthly_Review')
plt.show()
```

- **Time Series Analysis in R**

```
library(tidyverse)
library(ggplot2)
library(lubridate)
library(forecast)
library(zoo)
ts <- read.csv("reviews.csv")
summary(ts)

# Time Series Visualization --------------------------------


ds_15 <- ts %>%
  filter(date >= ("2010-01-01") & date <= ("2020-06-01"))

ds_15$date <- as.Date(ds_15$date, format = "%Y-%m-%d")

ds_15 <- ds_15 %>%
  mutate(week = (year(date) - year(min(date)))*52 +
           (week(date)-week(min(date))))

ds_15 <- ds_15 %>%
  mutate(month = (year(date) - year(min(date)))*12 +
           (month(date)-month(min(date))))

ds_15 <- ds_15 %>%
  mutate(day = (year(date) - year(min(date)))*365 +
           (month(date)-month(min(date)))*30+
           (day(date)-day(min(date))))
```

```
ds_15 <- ds_15 %>%
  mutate(year = year(date))

ds_15 <- ds_15 %>%
  group_by(week) %>%
  mutate(count_week = n())

ds_15 <- ds_15 %>%
  group_by(day) %>%
  mutate(count_day = n())

names(ds_15)[1] <- "id"


ts_15 <- ds_15[c(3,7)]
ts_15 <- ts_15 %>%
  distinct(week, .keep_all = TRUE)

ts_15_day <- ds_15[c(5,8)]
ts_15_day <- ts_15_day %>%
  distinct(day, .keep_all = TRUE)

ts_2015<- ts(ts_15$count_week,
             start=c(2010,1),
             frequency = 52)

ts_2015_day<- ts(ts_15_day$day,
                 start=c(2010,1),
                 frequency = 365)

components.ts = decompose(ts_2015)
plot(components.ts)

components.ts = decompose(ts_2015_day)
plot(components.ts)

# ANALYSIS of TIme Series --------------------------


# Working with 2019-20 data for time series modeling and other analysis

ds_19 <- ts %>%
  filter(date >= ("2019-01-01") & date <= ("2020-06-01"))
```

```
ts$date <- as.Date(ts$date, format = "%Y-%m-%d")

ds_19 <- ds_19 %>%
  mutate(week = (year(date) - year(min(date)))*52 +
           (week(date)-week(min(date))))

ds_19 <- ds_19 %>%
  mutate(month = (year(date) - year(min(date)))*12 +
           (month(date)-month(min(date))))

ds_19 <- ds_19 %>%
  mutate(day = (year(date) - year(min(date)))*365 +
           (month(date)-month(min(date)))*30+
           (day(date)-day(min(date))))

ds_19 <- ds_19 %>%
  mutate(year = year(date))

ds_19 <- ds_19 %>%
  group_by(week) %>%
  mutate(count_week = n())

ds_19 <- ds_19 %>%
  group_by(day) %>%
  mutate(count_day = n())

names(ds_19)[1] <- "id"
str(ds_19)

# Time Series ------------------ TIME Series prediction for n

#Daily Data Time Series
ts_daily <- ds_19
ts_daily <- ts_daily[c(5,8)]

ts_daily <- ts_daily %>%
  distinct(day, .keep_all = TRUE)

ts_daily <- ts_daily %>%
  summarize(day = day + 1, count_day)


plot(ts_daily)
```

```
time_series_day <- ts(ts_daily$count_day,
              start=c(2019,1),
              frequency = 365)


nValid_day <- 60
nTrain_day <- length(time_series_day) - nValid_day

train.tsDK.day <- window(time_series_day, start=c(2019,1),end=c(2019,nTrain_day))
valid.tsDK.day <- window(time_series_day, start=c(2019,nTrain_day+1),
end=c(2019,nTrain_day+nValid_day))

plot(time_series_day)

'Linear Trend model'
train.lm.day <- tslm(train.tsDK.day ~ trend)
summary(train.lm.day)
train.lm.pred.day <- forecast(train.lm.day,h=nValid_day,level = 0)
accuracy(train.lm.pred.day, valid.tsDK.day)

'Quadratic Model with Trend'
train.lm.quad.day <- tslm(train.tsDK.day ~ trend+I(trend^2))
summary(train.lm.quad.day)
train.lm.quad.pred.day <- forecast(train.lm.quad.day,h=nValid_day,level = 0)
accuracy(train.lm.quad.pred.day, valid.tsDK.day)

'Quadratic Model with Trend & Season'
train.lm.quad.s.day <- tslm(train.tsDK.day ~ trend+I(trend^2)+season)
summary(train.lm.quad.s.day)
train.lm.quad.s.pred.day <- forecast(train.lm.quad.s.day,h=nValid_day,level = 0)
accuracy(train.lm.quad.s.pred.day, valid.tsDK.day)

'ARIMA model'
ARIMAfit.day <- auto.arima(time_series_day, approximation=FALSE,trace=TRUE)

summary(ARIMAfit.day)

pred_days <- predict(ARIMAfit.day,n.ahead=60)
pred_days

par(mfrow = c(1,1))
plot(time_series_day,type='l',ylim=c(1,3000),xlim=c(2019,2020.6),xlab = 'Duration (In Days - Jan
2019 - May 2020)',ylab = 'No. of Bookings in NYC')
```

```
lines(pred_days$pred,col='blue',lwd = 2)
lines(pred_days$pred+2*pred_days$se,col='orange')
lines(pred_days$pred-2*pred_days$se,col='orange')



# Visualize the linear trend model
par(mfrow = c(1, 1))
plot(train.lm.quad.pred.day, ylim = c(100, 3000),xlim=c(2019.0,2020.5),  ylab = "No. of Daily
Bookings in NYC", xlab = "(In Days - Jan 2019 - May 2020)",
     bty = "l", xaxt = "n",main = "", flty = 2)
axis(1, at = seq(2019.0, 2020.5, 1), labels = format(seq(2019.0, 2020.5, 1)))
lines(train.lm.quad.pred.day$fitted, lwd = 2, col = "blue")
lines(valid.tsDK.day)


par(mfrow = c(1, 1))
plot(train.lm.pred, ylim = c(1300, 2600),  ylab = "Ridership", xlab = "Time",
     bty = "l", xaxt = "n", xlim = c(1991,2006),main = "", flty = 2)
axis(1, at = seq(1991, 2006, 1), labels = format(seq(1991, 2006, 1)))
lines(train.lm.pred$fitted, lwd = 2, col = "blue")
lines(valid.ts)



# -------------- weekly Data Time Series -------------
ts_19 <- ds_19
ts_19 <- ts_19[c(3,7)]
ts_19 <- ts_19 %>%
  distinct(week, .keep_all = TRUE)

ts_19 <- ts_19 %>%
  summarize(week = week + 1, count_week)


plot(ts_19)

ts_20<- ts(ts_19$count_week,
           start=c(2019,1),
           frequency = 52)
plot(ts_20)

# Data Partition: Considering last 3 months to validate to compare effect of covid
```

```
nValid <- 12
nTrain <- length(ts_20) - nValid

train.tsDK <- window(ts_20, start=c(2019,1),end=c(2019,nTrain))
valid.tsDK <- window(ts_20, start=c(2019,nTrain+1), end=c(2019,nTrain+nValid))


# Time series modeling

'Linear Trend model'
train.lm <- tslm(train.tsDK ~ trend)
summary(train.lm)
train.lm.pred <- forecast(train.lm,h=nValid,level = 0)
accuracy(train.lm.pred, valid.tsDK)

# Visualize the linear trend model
par(mfrow = c(1, 1))
plot(train.lm.pred, ylim = c(100, 15000),xlim=c(2019,2020.8),  ylab = "No. of Reviews", xlab =
"Time in Weeks",
     bty = "l", xaxt = "n",main = "", flty = 2)
axis(1, at = seq(2019.0, 2020, 1), labels = format(seq(2019.0, 2020, 1)))
lines(train.lm.pred$fitted, lwd = 2, col = "blue")
lines(valid.tsDK)

'Quadratic Trend Model'
train.lm.poly <- tslm(train.tsDK ~ trend + I(trend^2))
summary(train.lm.poly)
train.lm.poly.pred <- forecast(train.lm.poly,h=nValid,level = 0)
accuracy(train.lm.poly.pred, valid.tsDK)

'linear trend with season'
train.lmDK.season <- tslm(train.tsDK ~ season)
summary(train.lmDK.season)
train.lmDK.season.pred <- forecast(train.lmDK.season, h=nValid, level=0 )
accuracy(train.lmDK.season.pred, valid.tsDK)

'linear trend with season'
train.lmDK.trend.season <- tslm(train.tsDK ~ trend+season)
summary(train.lmDK.trend.season )
train.lmDK.trend.season.pred <- forecast(train.lmDK.trend.season, h=nValid, level=0 )
accuracy(train.lmDK.trend.season.pred, valid.tsDK)

'Polynomial trend with season'
train.lmDK.poly.trend.season <- tslm(train.tsDK ~ trend+I(trend^2)+season)
```

```
summary(train.lmDK.poly.trend.season )
train.lmDK.poly.trend.season.pred <- forecast(train.lmDK.trend.season, h=nValid, level=0 )
accuracy(train.lmDK.poly.trend.season.pred, valid.tsDK)

# ARIMA MODEL on weekly time series-----------------

ARIMAfit <- auto.arima(ts_20, approximation=FALSE,trace=TRUE)

summary(ARIMAfit)

pred <- predict(ARIMAfit,n.ahead=12)
pred

par(mfrow = c(1,1))
plot(ts_20,type='l',ylim=c(1,16000),xlim=c(2019,2020.8),xlab = 'Time (Jan 2019 - May
2020)',ylab = 'No. of Booking in NYC')
lines(pred$pred,col='blue')
lines(pred$pred+2*pred$se,col='orange')
lines(pred$pred-2*pred$se,col='orange')
```

- **Topic Modeling on Reviews Dataset in R**

```
Reviews2020 <- read.csv('Reviews2020.csv',stringsAsFactors = F)

Jan <- Reviews2020[(Reviews2020$date> "2020-01-01" & Reviews2020$date < "2020-01-31"),]
Feb <- Reviews2020[(Reviews2020$date> "2020-02-01" & Reviews2020$date < "2020-02-
28"),]
Mar <- Reviews2020[(Reviews2020$date> "2020-03-01" & Reviews2020$date < "2020-03-
31"),]
Apr <- Reviews2020[(Reviews2020$date> "2020-04-01" & Reviews2020$date < "2020-04-30"),]
May <- Reviews2020[(Reviews2020$date> "2020-05-01" & Reviews2020$date < "2020-05-
31"),]
Jun <- Reviews2020[(Reviews2020$date> "2020-06-01" & Reviews2020$date < "2020-06-30"),]

### GENERAL OVERVIEW OF DATASET ###
Corpus <- corpus(Reviews2020$comments)
summary(Corpus)

library(stopwords)
Dfm <- dfm(Corpus,
        remove_punc = T,
        remove = c(stopwords("english"),stopwords("spanish"), stopwords("french"),
                stopwords('german'),"great" ,"place", "stay", "airbnb"),
        stem = T)
dim(Dfm)
```

```
topfeatures(Dfm,20)

textplot_wordcloud(Dfm,min_size=1, max_size=7, max_words=200)

Dfm<- dfm_trim(Dfm,min_termfreq=100, min_docfreq=5)
dim(Dfm)
summary(Dfm)
Dfm <- na.omit(Dfm)

#Finding terms related to covid
term_sim <- textstat_simil(Dfm,
                selection="covid",
                margin="feature",
                method="correlation")
as.list(term_sim,n=15)

############# GENERATING MONTHLY WORDCLOUDS ######################
### JANUARY ###
Corpus1 <- corpus(Jan$comments)
summary(Corpus1)

library(stopwords)
Dfm1 <- dfm(Corpus1,
        remove_punc = T,
        remove = c(stopwords("english"),stopwords("spanish"), stopwords("french"),
                stopwords('german')),
        stem = T)
dim(Dfm1)
topfeatures(Dfm1,20)

Dfm1 <- dfm_remove(Dfm1, c("airbnb","place","stay","great", "recommend"))

textplot_wordcloud(Dfm1,min_size=1, max_size=7, max_words=200)

### FEBRUARY ###
Corpus2 <- corpus(Feb$comments)
summary(Corpus2)

library(stopwords)
Dfm2 <- dfm(Corpus2,
         remove_punc = T,
         remove = c(stopwords("english"),stopwords("spanish"), stopwords("french"),
                 stopwords('german')),
         stem = T)
```

```
dim(Dfm2)
topfeatures(Dfm2,20)

Dfm2 <- dfm_remove(Dfm2, c("airbnb","place","stay","great", "recommend"))

textplot_wordcloud(Dfm2,min_size=1, max_size=7, max_words=200)

### MARCH ###
Corpus3 <- corpus(Mar$comments)
summary(Corpus3)

library(stopwords)
Dfm3 <- dfm(Corpus3,
          remove_punc = T,
          remove = c(stopwords("english"),stopwords("spanish"), stopwords("french"),
                    stopwords('german')),
          stem = T)
dim(Dfm3)
topfeatures(Dfm3,20)

Dfm3 <- dfm_remove(Dfm3, c("airbnb","place","stay","great", "recommend"))

textplot_wordcloud(Dfm3,min_size=1, max_size=7, max_words=200)

### ARPIL ###
Corpus4 <- corpus(Apr$comments)
summary(Corpus4)

library(stopwords)
Dfm4 <- dfm(Corpus4,
          remove_punc = T,
          remove = c(stopwords("english"),stopwords("spanish"), stopwords("french"),
                    stopwords('german')),
          stem = T)
dim(Dfm4)
topfeatures(Dfm4,20)

Dfm4 <- dfm_remove(Dfm4, c("airbnb","place","stay","great", "recommend"))

textplot_wordcloud(Dfm4,min_size=1, max_size=7, max_words=200)

### MAY ###
Corpus5 <- corpus(May$comments)
summary(Corpus5)
```

```
library(stopwords)
Dfm5 <- dfm(Corpus5,
        remove_punc = T,
        remove = c(stopwords("english"),stopwords("spanish"), stopwords("french"),
                stopwords('german'), "place", "stay", "airbnb"),
        stem = T)
dim(Dfm5)
topfeatures(Dfm5,20)

Dfm5 <- dfm_remove(Dfm5, c("airbnb","place","stay","great", "recommend"))

textplot_wordcloud(Dfm5,min_size=1, max_size=7, max_words=200)

### JUNE ###
Corpus6 <- corpus(Jun$comments)
summary(Corpus6)

library(stopwords)
Dfm6 <- dfm(Corpus6,
        remove_punc = T,
        remove = c(stopwords("english"),stopwords("spanish"), stopwords("french"),
                stopwords('german')),
        stem = T)
dim(Dfm6)
topfeatures(Dfm6, 20)

Dfm6 <- dfm_remove(Dfm6, c("airbnb","place","stay","great", "recommend"))

textplot_wordcloud(Dfm6,min_size=1, max_size=7, max_words=200)


###################### GENTERATING PLOTS ################################
### JANUARY ###
library(topicmodels)
library(tidytext)

Dfm1<- dfm_trim(Dfm1,min_termfreq=100, min_docfreq=5)
dim(Dfm1)

Dfm1 <- as.matrix(Dfm1)
Dfm1 <-Dfm1[which(rowSums(Dfm1)>0),]
Dfm1 <- as.dfm(Dfm1)
Dfm1
```

```
Lda1 <- LDA(Dfm1,k=4,control=list(seed=101))
Lda1

Lda1_td <- tidy(Lda1)
Lda1_td

library(ggplot2)
library(dplyr)

top_terms1 <- Lda1_td %>%
  group_by(topic) %>%
  top_n(7, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms1 %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()

### FEBRUARY ###
Dfm2<- dfm_trim(Dfm2,min_termfreq=100, min_docfreq=5)
dim(Dfm2)

Dfm2 <- as.matrix(Dfm2)
Dfm2 <-Dfm2[which(rowSums(Dfm2)>0),]
Dfm2 <- as.dfm(Dfm2)
Dfm2

Lda2 <- LDA(Dfm2,k=4,control=list(seed=101))
Lda2

Lda2_td <- tidy(Lda2)
Lda2_td

top_terms2 <- Lda2_td %>%
  group_by(topic) %>%
  top_n(7, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
```

```
top_terms2 %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()

### MARCH ###
Dfm3<- dfm_trim(Dfm3,min_termfreq=100, min_docfreq=5)
dim(Dfm3)

Dfm3 <- as.matrix(Dfm3)
Dfm3 <-Dfm3[which(rowSums(Dfm3)>0),]
Dfm3 <- as.dfm(Dfm3)
Dfm3

Lda3 <- LDA(Dfm3,k=4,control=list(seed=101))
Lda3

Lda3_td <- tidy(Lda3)
Lda3_td

top_terms3 <- Lda3_td %>%
  group_by(topic) %>%
  top_n(7, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms3 %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()

### APRIL ###
Dfm4<- dfm_trim(Dfm4,min_termfreq=100, min_docfreq=5)
dim(Dfm4)

Dfm4 <- as.matrix(Dfm4)
Dfm4 <-Dfm4[which(rowSums(Dfm4)>0),]
Dfm4 <- as.dfm(Dfm4)
Dfm4
```

```
Lda4 <- LDA(Dfm4,k=4,control=list(seed=101))
Lda4

Lda4_td <- tidy(Lda4)
Lda4_td

top_terms4 <- Lda4_td %>%
  group_by(topic) %>%
  top_n(7, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms4 %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()

### MAY ###
Dfm5<- dfm_trim(Dfm5,min_termfreq=100, min_docfreq=5)
dim(Dfm5)

Dfm5 <- as.matrix(Dfm5)
Dfm5 <-Dfm5[which(rowSums(Dfm5)>0),]
Dfm5 <- as.dfm(Dfm5)
Dfm5

Lda5 <- LDA(Dfm5,k=4,control=list(seed=101))
Lda5

Lda5_td <- tidy(Lda5)
Lda5_td

top_terms5 <- Lda5_td %>%
  group_by(topic) %>%
  top_n(7, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms5 %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()

### JUNE ###
Dfm6<- dfm_trim(Dfm6,min_termfreq=100, min_docfreq=5)
dim(Dfm6)

Dfm6 <- as.matrix(Dfm6)
Dfm6 <-Dfm6[which(rowSums(Dfm6)>0),]
Dfm6 <- as.dfm(Dfm6)
Dfm6

Lda6 <- LDA(Dfm6,k=4,control=list(seed=101))
Lda6

Lda6_td <- tidy(Lda6)
Lda6_td

top_terms6 <- Lda6_td %>%
  group_by(topic) %>%
  top_n(7, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms6 %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```

- **Tableau**

Two files - Line Graph and Exploration