Airbnb Market Analysis Boston

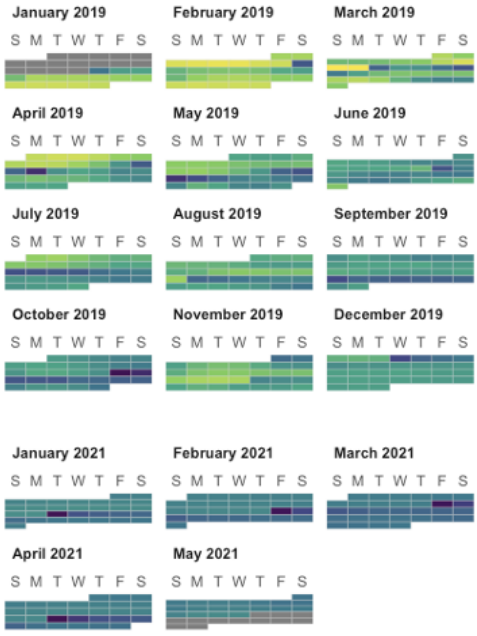
1. Comprehensive list of matrices

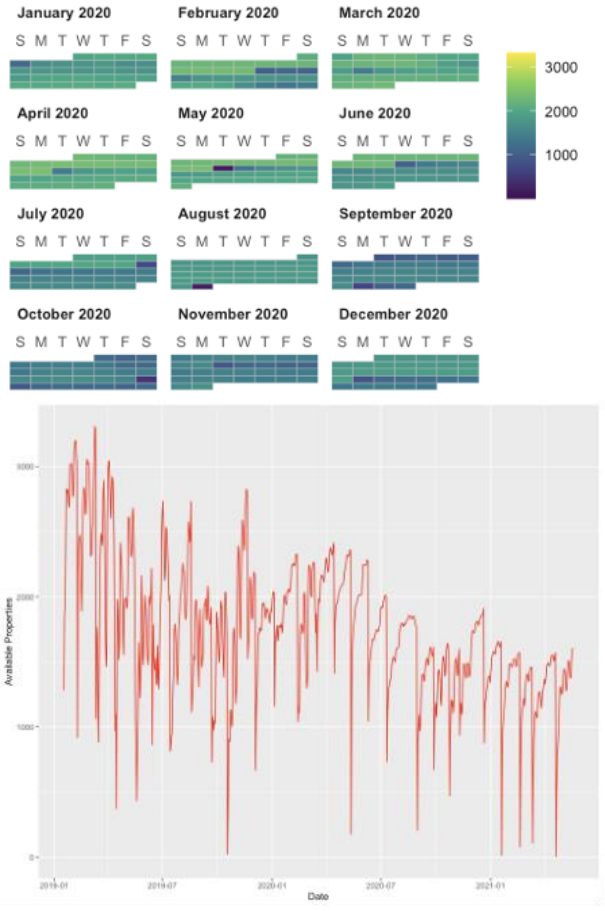
1.1. Market supply: the number of active listings

The following graphs show the amount of available and unavailable properties given from the Airbnb dataset. The numbers are gathered as the total number of available and unavailable properties for all properties in the Boston area. The two together can be used to interpret the market supply during the selected time periods. Since we are assessing the effect of the pandemic, 2019 is used as a baseline for normal market conditions and 2020 represents the market during the pandemic. Data from January 2019 to April 2021 and most of the focus is for the time from January 2020 to June 2020.

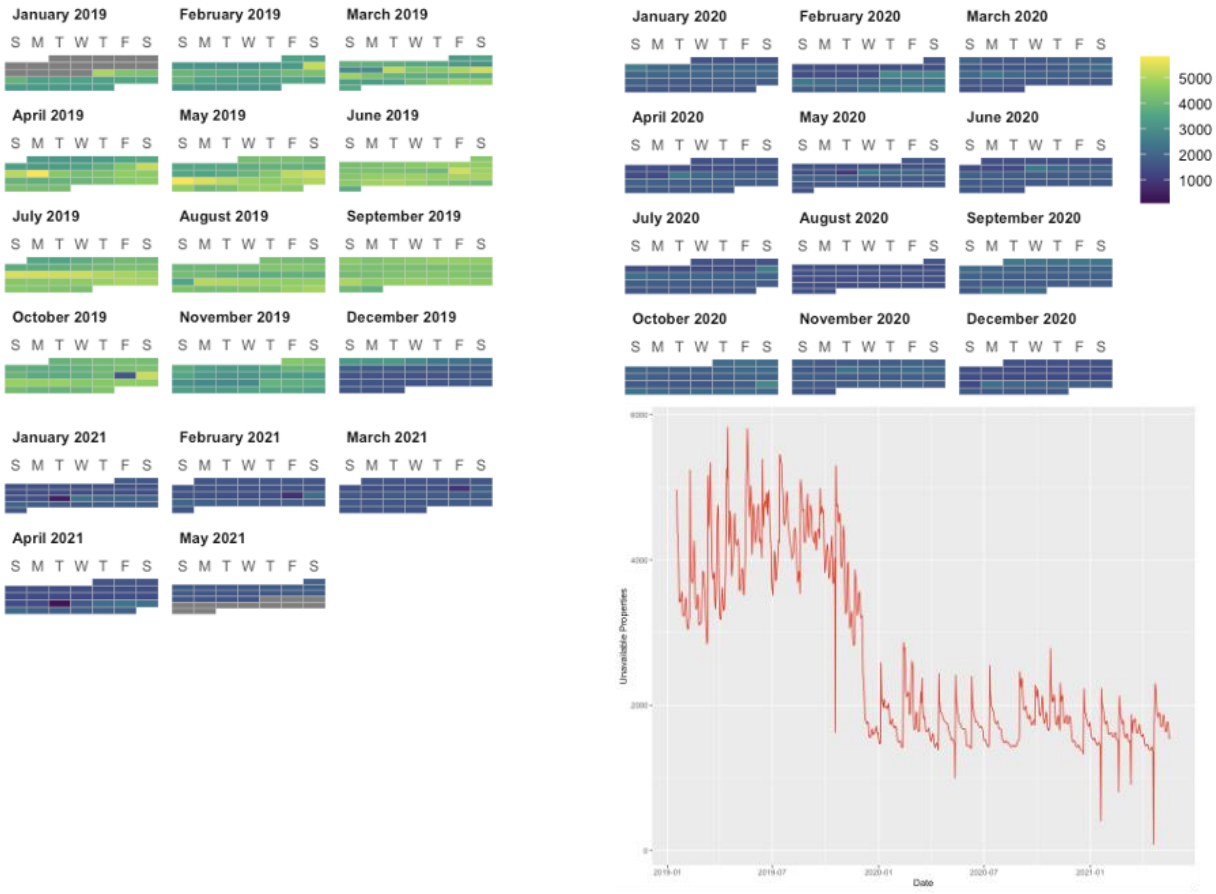
2019-Current

Availability of Properties





Property Unavailability

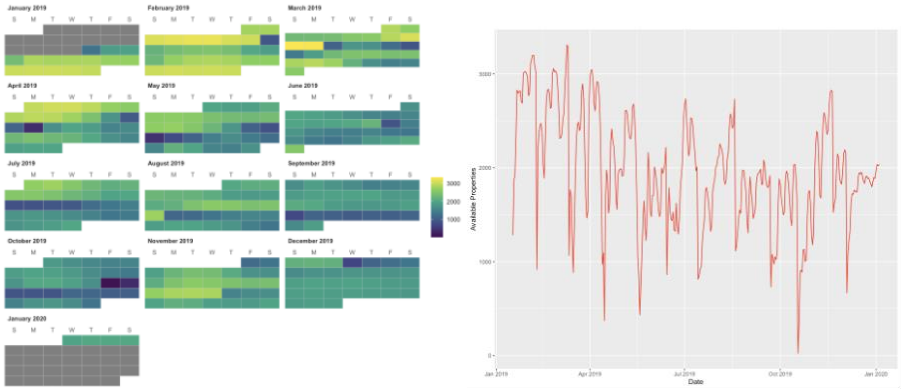


Remarks

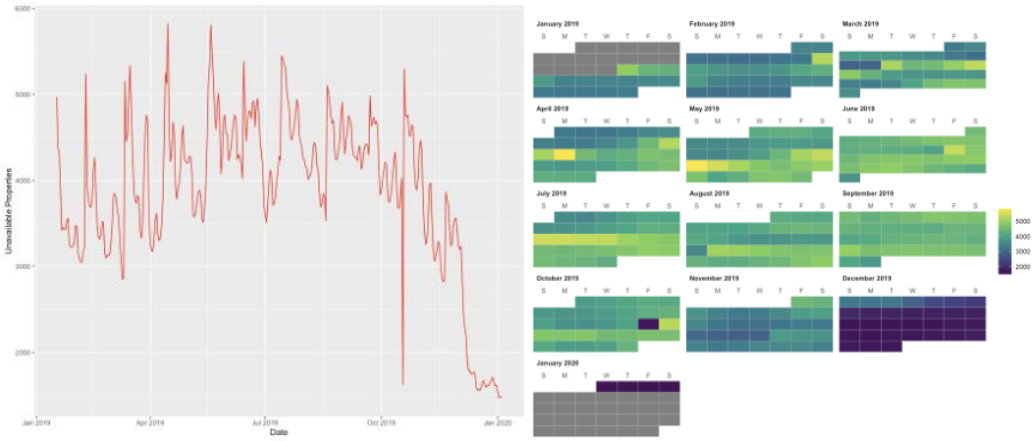
The amount of unavailable properties drops off significantly starting in December of 2019 and seems to mostly stabilize for the entirety of 2020. This can be interpreted as a drop in supply in the market.

2019

Property Availability



Property Unavailability

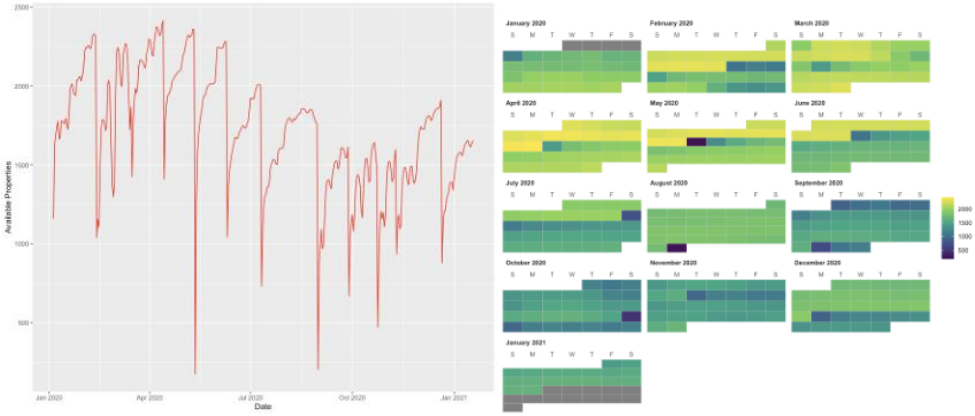


Remarks

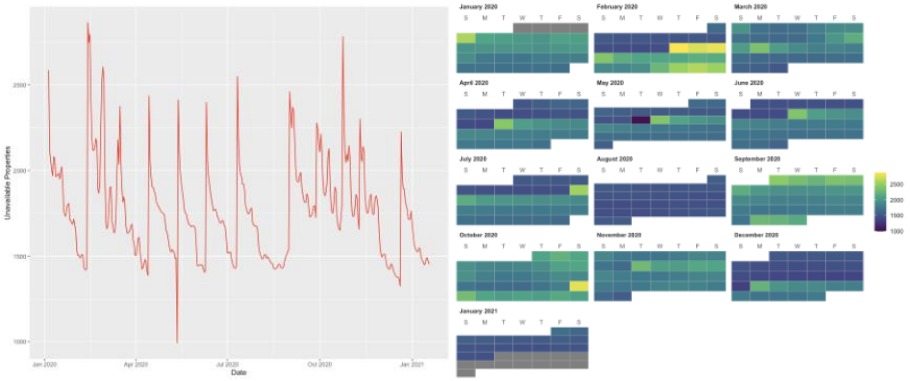
When looking at 2019 alone, the supply decrease in December is clearer. Property availability is stable for the month of December. These two factors together point to a drop in market supply that starts in December.

2020

Property Availability



Property Unavailability

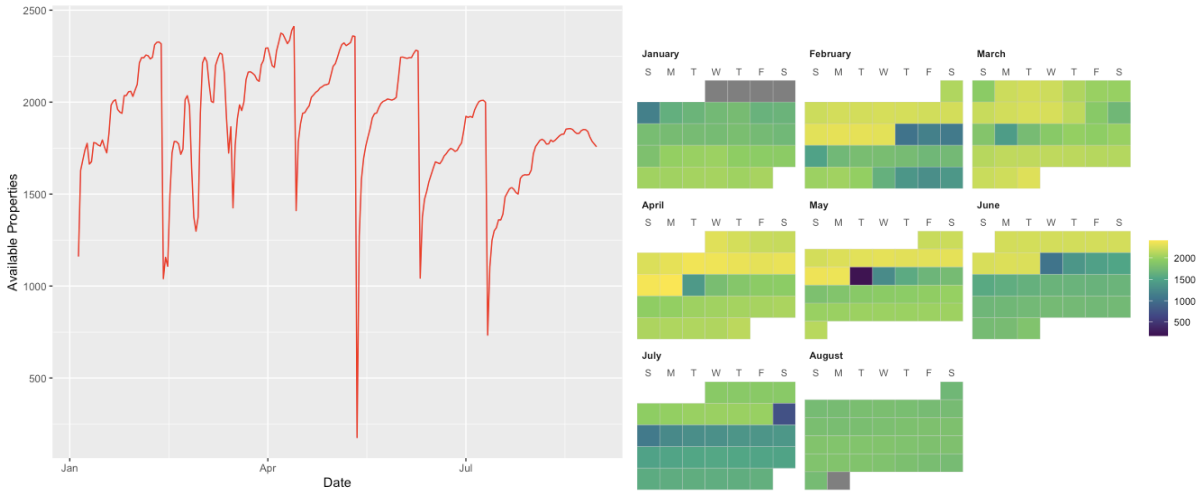


Remarks

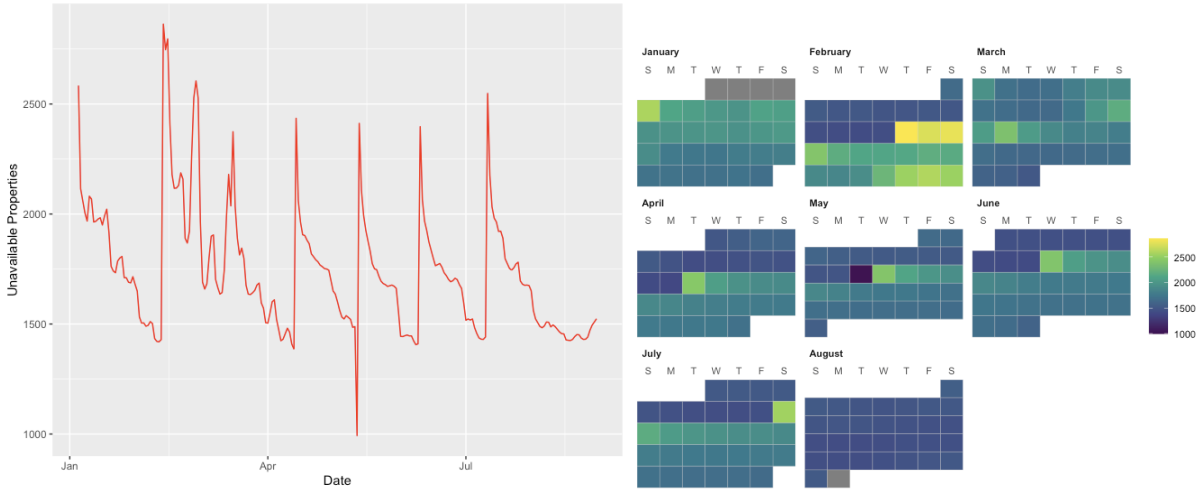
For all of 2020, the shift in December is followed for availability and unavailability of properties. There is an increase in availability in February and a further decrease from March to May.

Pandemic January 2020 to July 2020

Property Availability



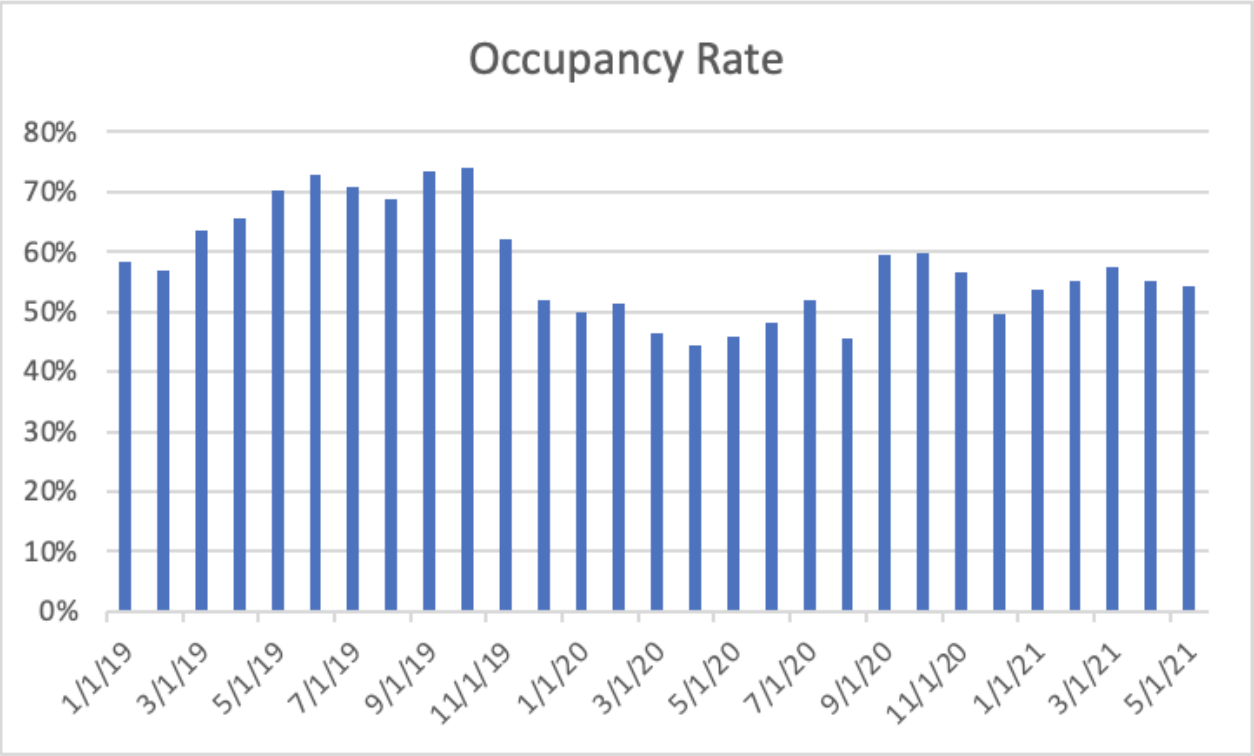
Property Unavailability



Remarks

There is an increase in availability in February and a further decrease in properties that are unavailable from March to May, the decrease is larger and therefore points to a further decrease in overall market supply. There is a clear drop in supply in May observed as both Available and Unavailable property numbers drop significantly together when they are expected to have opposite movements. Overall supply is lower during these months likely due to owners deciding to either sell or delist their Airbnb listings based on pandemic difficulties and an increase in home value.

Occupancy Rate



Remarks:

The graph above shows the occupancy rate gathered from the Airbnb calendar dataset. The rate is calculated through taking the mean of a dummy variable, grouped by month, that is equal to 1 if a property is unavailable and 0 if it is listed as available. The occupancy rate points to a drop in market demand that starts in December of 2019 when the rate drops below 50%. The rate does not go above 50% until August of 2020 but this rate is still lower than the prior year since supply drops around that time as well. If everything was held equal, a decrease in supply should have caused the occupancy rate to increase rather than decrease.

1.2. Market demand: occupancy rate, flight data, traffic data

Market Demand (Flight Data)

Flight Statistics (1/19-12/20)

- Clear drop in flights to BOS in March of 2020 (start of pandemic in USA)
- Lowest month is April (Massive Drop)
- ~39K domestic passengers to 1.5M from the same time the year prior
- ~1K international passengers to 343K
- ~98% decrease in flight travel to Boston during this time period
- As of December 2020, the number of domestic flights to BOS is still well below the average in 2019
- The overall drop in flight passengers to Boston can be interpreted as a drop in Airbnb demand since overall there are less people traveling to Boston who do not live in or near the city which means there are less people to who might require Airbnb

Source: https://www.transtats.bts.gov/Data_Elements.aspx?Data=1

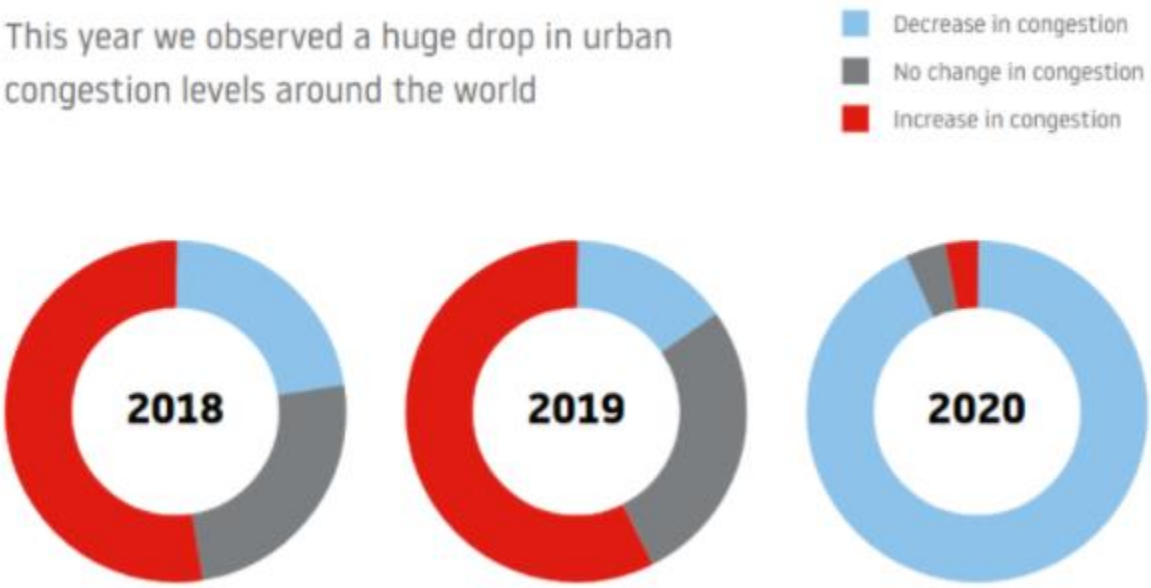
This is overall passenger travel data for BOS (Logan International Airport in Boston) as the destination airport for air travel divided by month.

Year	Month	Domestic	International	Total
2019	1	1,085,964	254,885	1,340,849
2019	2	1,088,080	208,274	1,296,354
2019	3	1,423,435	300,030	1,723,465
2019	4	1,456,224	343,259	1,799,483
2019	5	1,554,028	357,299	1,911,327
2019	6	1,540,859	385,762	1,926,621
2019	7	1,566,401	440,828	2,007,229
2019	8	1,584,508	437,829	2,022,337
2019	9	1,386,290	348,132	1,734,422
2019	10	1,491,495	331,494	1,822,989
2019	11	1,321,081	252,876	1,573,957
2019	12	1,322,640	269,517	1,592,157
2019	**TOTAL**	**16,821,005**	**3,930,185**	**20,751,190**
2020	1	1,171,313	282,258	1,453,571
2020	2	1,170,085	222,901	1,392,986
2020	3	702,732	134,801	837,533
2020	4	39,642	718	40,360
2020	5	92,195	1,221	93,416
2020	6	210,383	2,756	213,139
2020	7	338,969	16,795	355,764
2020	8	304,510	29,555	334,065
2020	9	274,127	25,863	299,990
2020	10	337,445	24,798	362,243
2020	11	325,345		
2020	12	311,793		
2020	**TOTAL**	**5,278,539**	**741,666**	**5,383,067**

Market Demand (Traffic Data)

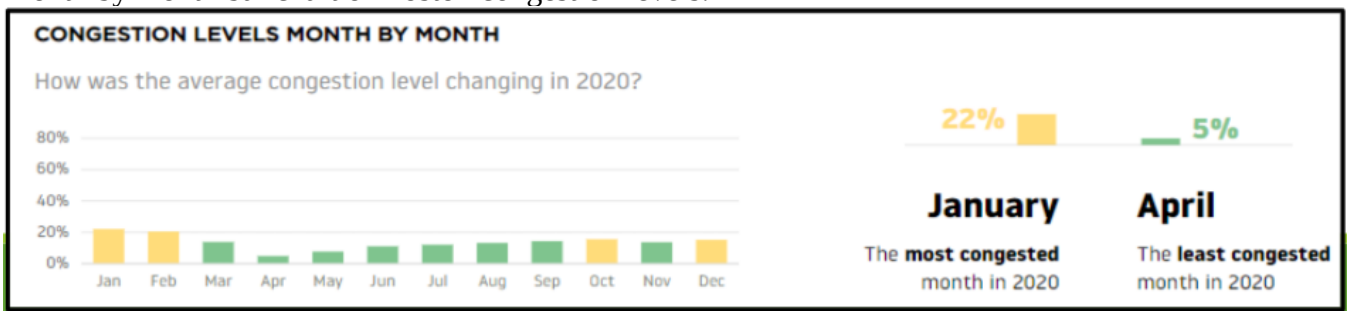
Source: https://www.tomtom.com/en_gb/traffic-index/boston-traffic/
This information is the overall traffic congestion from 416 cities in 57 different countries (Boston USA included)
List of cities: https://www.tomtom.com/en_gb/traffic-index/ranking/

Overall Global congestion graph:



The global traffic congestion growth was stable from 2018 to 2019 with most cities being monitored experiencing an increase in congestion during this timeframe. However, from 2019 to 2020 nearly all the cities monitored experience a decrease in traffic congestion levels and Boston was no exception to this trend.
The traffic congestion in Boston from 2018 to 2019 was stable, experiencing virtually no significant change in traffic congestion around the city (less than 1% change). The congestion level in 2020 was a mere 15% which is a staggering 42% decrease in traffic congestion from the 2 years prior.

Month by month bar chart of Boston congestion levels:



With less people traveling, the demand for Airbnb is bound to decrease compared to previous years.

1.3. Customer comments: topics, sentiments, etc.

Review.csv dataset was used to complete this task. This dataset contains all the reviews from the past 10 years or so. For the purpose of our analysis, reviews from January 2020 to June 2020 were filtered out.

Each month was filtered out individually and IDs like ‘Jan’, ‘Feb’ and so forth were added to each dataset so that it will be easier to do analysis and compare the monthly results. This was followed by merging all the six-month datasets to one individual dataset. Sample sets for the review can be found below.

	listing_id	id	date	reviewer_id	comments	month	line
1	22195	586265909	2020-01-01	75308875	Clean, comfy, and convenient as advertised. All guests were ...	Jan	1
2	22195	586304419	2020-01-01	161890616	After booking my stay here I was surprised to find myself in ...	Jan	2
3	22195	590079879	2020-01-07	321732761	The location is perfect which is near the city and by local fav...	Jan	3
4	22195	592146717	2020-01-12	62004285	Was very disappointed with the experience. Extremely noisy ...	Jan	4
5	22195	595054621	2020-01-19	261884581	Instead of the pretty well lit room in the picture we were in ...	Jan	5
6	22195	597392754	2020-01-25	65656566	Place was very poorly run. The pictures are not what youâ€...	Jan	6
1654	22195	603022802	2020-02-08	240692234	The unit we stayed in is not the same as the picture and was...	Feb	1654
4819	40601	625238755	2020-05-18	118217777	Excellent host, newly renovated private bathroom, and place...	May	4819
4820	190170	621580465	2020-04-02	51090585	Fanny was a great host and we were welcomed with great h...	May	4820
3227	190170	613260191	2020-03-01	1936152	Stayed here for 2 months and at the end I was wondering w...	Mar	3227
4629	190170	621580465	2020-04-02	51090585	Fanny was a great host and we were welcomed with great h...	Apr	4629

Fig1 : Sample set for review dataset

Below are the results of analysis:

1. Here is a word cloud based on a six months review. It gives us an overview of what kind of reviews were left by the customers. From the word cloud we can tell most of the positive reviews reflected good hygiene and ambience. As per negative reviews, it was a concern of a noisy neighborhood.

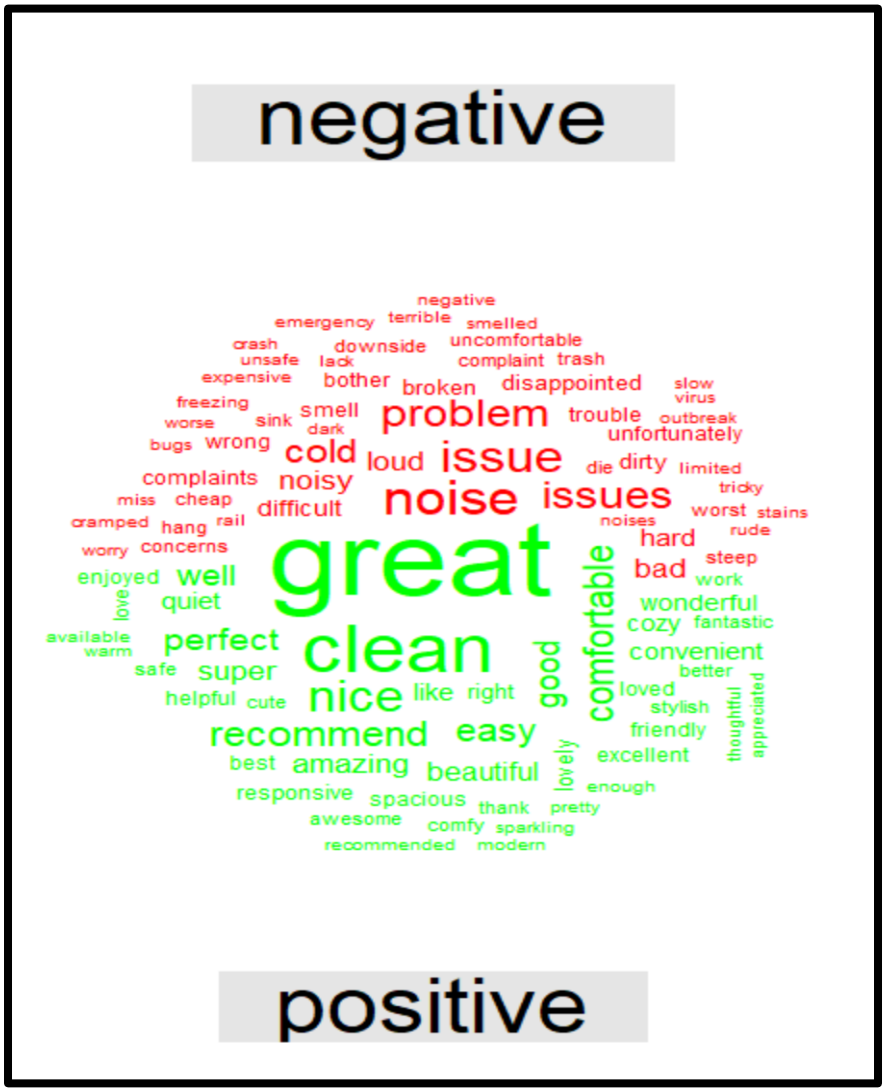


Fig2 : Word Cloud for the overall 6months reviews

2. Next picture shows the sentiment variation. We wanted to see if thoughts and opinions of the people changed after COVID19 or during the lockdown period. Looking at the picture, we don't really see significant differences. The proportion of positive sentiments VS negative remains the same.

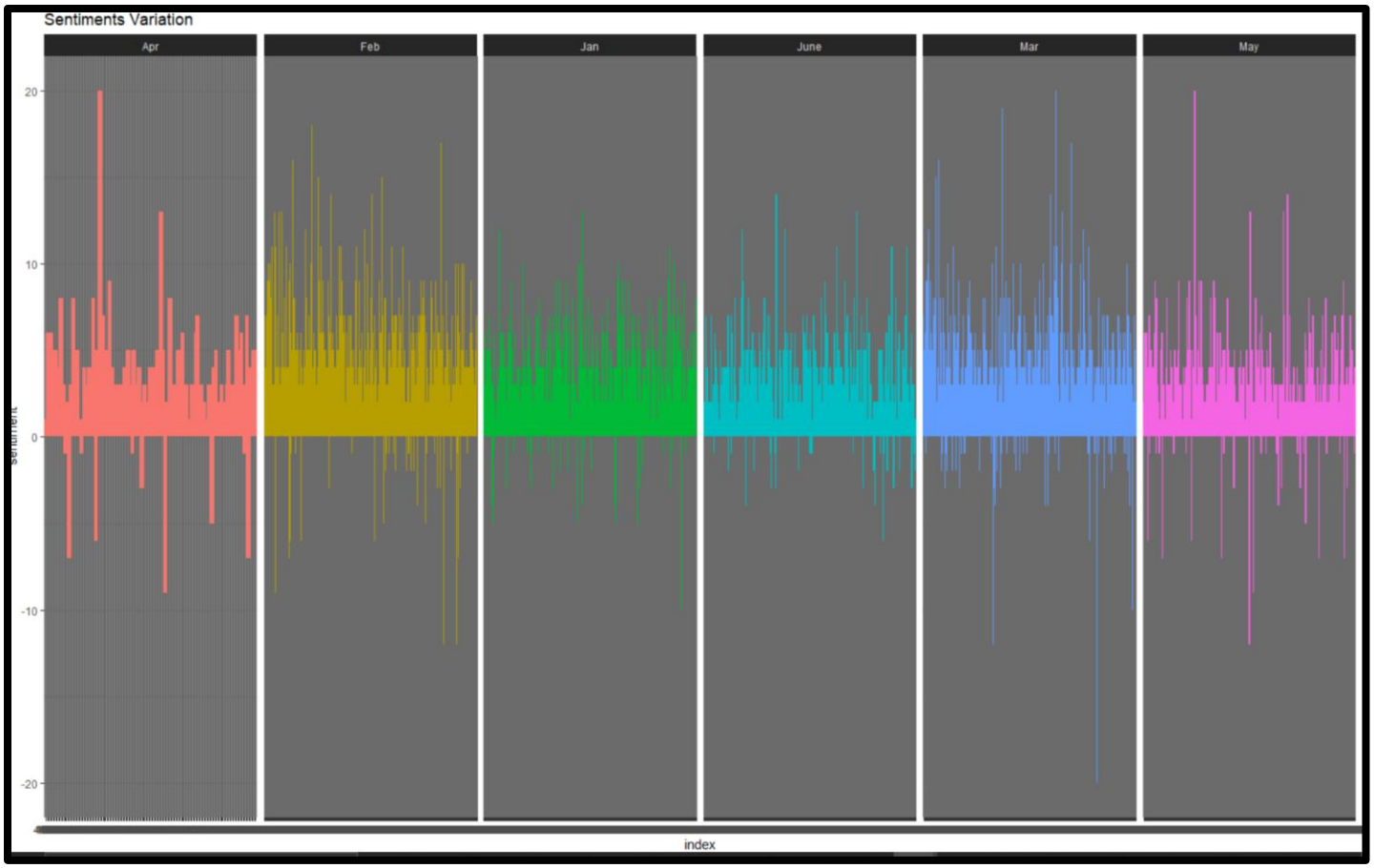


Fig3 : Monthly sentiment variation

3. Below shows a plot of comparison of positive and negative words used by reviewers. We noticed most of the reviews were based on the same topics. Positive reviews left by most of the customers were about cleanliness and how easy, comfortable, and convenient the place was, whereas negative comments were mostly about issues like noise and cold. One interesting thing we noticed was during the month of April, the number of reviews were significantly less compared to earlier months. We think this has to do with Covid19. During the month of April, there was a surge on Covid19 cases, and most parts of the country were under lockdown. Since a lot of people were not traveling, this implies that not many people were using Airbnb.

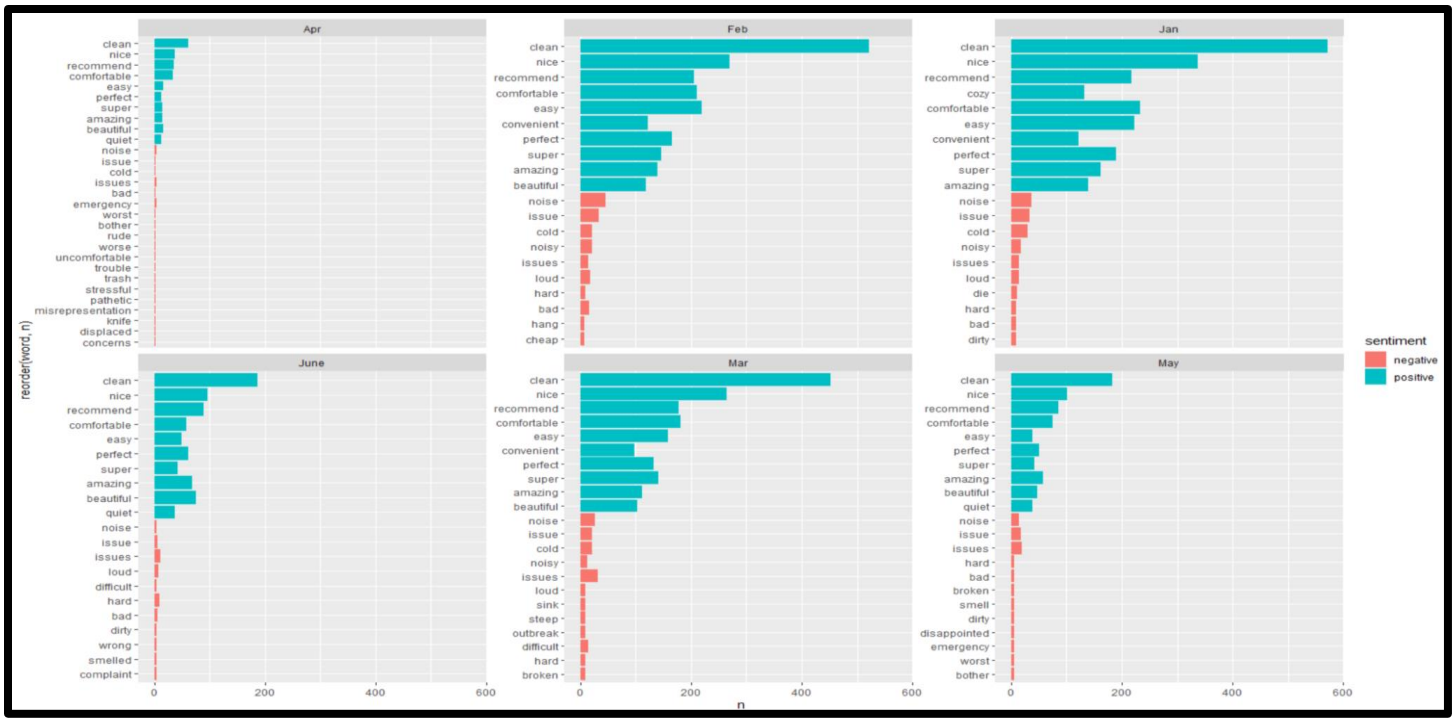


Fig4: Sentiment analysis based on Topics

4. This analysis was carried out using the get_nrc_sentiment function via Syuzhet packages for analysis of emotion words expressed in text. This function lets you analyze your text based on several words for emotional expression of anger, fear, anticipation, trust, surprise, sadness, joy, and disgust.

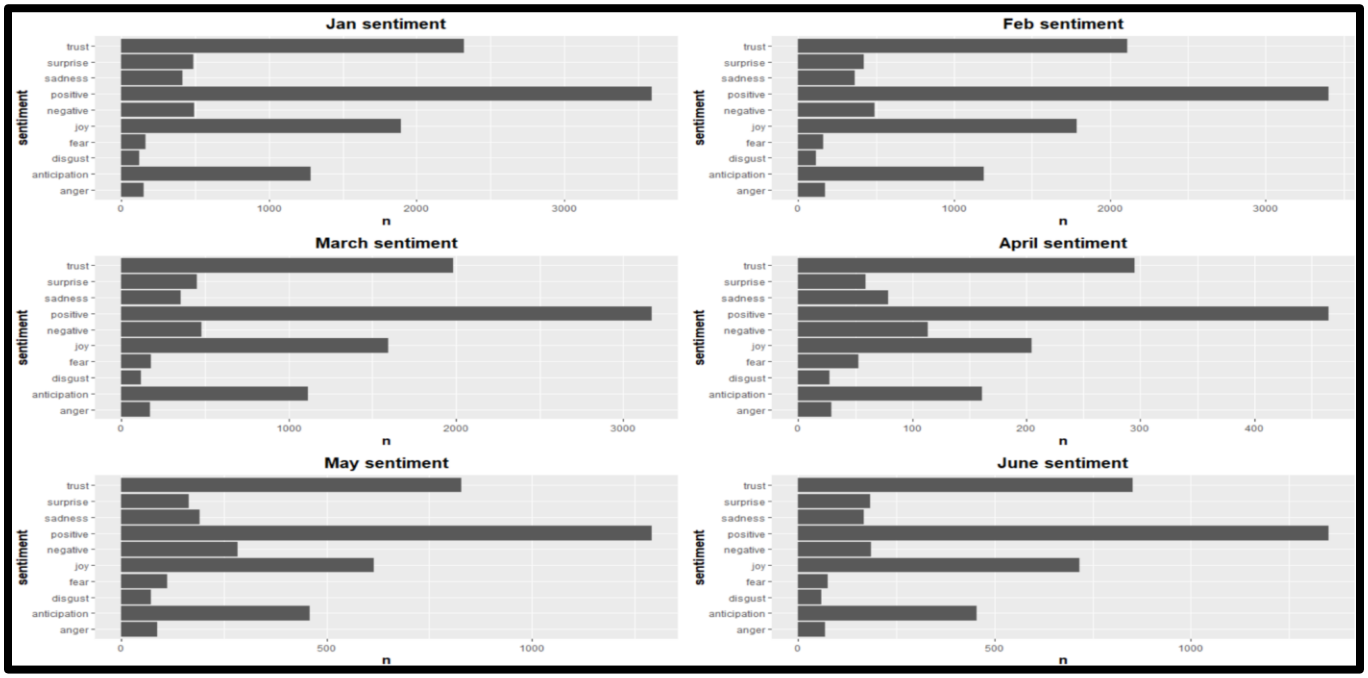


Fig5: Sentiment analysis based on emotional expression

2. What factors have affected Airbnb hosts' market exit decisions?
   2.1. Please specify how "exit" listings are defined and identified in your dataset.

   listings.csv dataset was used to complete this task. This dataset contains all the listings from the individual months. For the purpose of our analysis, listings from January 2020 to June 2020 were filtered out. All month's data was then merged into new dataset and IDs like 'Jan', 'Feb' and so forth were added to each dataset so that it will be easier to do analysis and compare the monthly results. This was followed by merging all the six-month datasets to one individual dataset.

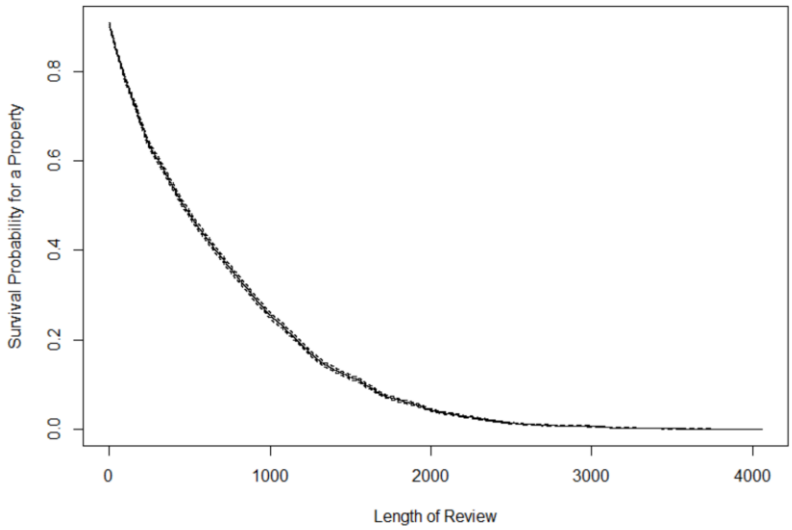
   We evaluated below factors for the exit criteria and used the Survival Analysis Model to find out if the length of review affects property's survival on the market.

   - Availability_30,availability_90,availability_365
   - Number_of_reviews,number_of_reviews_ltm
   - First_review
   - Last_review
   - Review_scores_rating
   - Review_scores_accuracy
   - Review_scores_cleanliness
   - Review_scores_checkin
   - Review_scores_communication
   - Review_scores_location
   - Review_scores_value
   - Instant_bookable

   Results

   ❑ The survival of the property to stay on the market is more, when the time difference between the reviews is less thus as the property gets more recent reviews , the probability of a property staying on market is more.

   ❑ If the property does not get reviews, then the probability of the property exiting will be more



- ❑ The analysis also indicated that the first review and last review were statistically significant factors in the market exit decision making process

```
> summary(coxph)
Call:
coxph(formula = Surv(lenreview, availability) ~ X, data = new_data,
    method = "breslow")

  n= 17627, number of events= 17627
   (4794 observations deleted due to missingness)

                                coef  exp(coef)   se(coef)         z Pr(>|z|)
Xavailability_30             1.643e-04  1.000e+00  6.202e-04     0.265    0.791
Xavailability_90             3.452e-05  1.000e+00  2.247e-04     0.154    0.878
Xavailability_365            1.116e-05  1.000e+00  6.016e-05     0.185    0.853
Xnumber_of_reviews           1.176e-04  1.000e+00  1.396e-04     0.843    0.400
Xnumber_of_reviews_ltm       4.159e-04  1.000e+00  3.849e-04     1.081    0.280
Xfirst_review                2.190e-01  1.245e+00  2.584e-05  8474.072   <2e-16
Xlast_review                -2.190e-01  8.033e-01  2.584e-05 -8474.313   <2e-16
Xreview_scores_rating       -1.416e-04  9.999e-01  8.740e-04    -0.162    0.871
Xreview_scores_accuracy      4.020e-03  1.004e+00  8.642e-03     0.465    0.642
Xreview_scores_cleanliness   4.151e-03  1.004e+00  8.884e-03     0.467    0.640
Xreview_scores_checkin      -1.560e-03  9.984e-01  1.024e-02    -0.152    0.879
Xreview_scores_communication -3.414e-03  9.966e-01  8.931e-03    -0.382    0.702
Xreview_scores_location      1.322e-04  1.000e+00  1.024e-02     0.013    0.990
Xreview_scores_value        -2.298e-03  9.977e-01  7.976e-03    -0.288    0.773
Xinstant_bookable            2.415e-03  1.002e+00  1.585e-02     0.152    0.879

Xavailability_30
Xavailability_90
Xavailability_365
Xnumber_of_reviews
Xnumber_of_reviews_ltm
Xfirst_review                ***
Xlast_review                 ***
Xreview_scores_rating
Xreview_scores_accuracy
Xreview_scores_cleanliness
Xreview_scores_checkin
Xreview_scores_communication
Xreview_scores_location
Xreview_scores_value
Xinstant_bookable
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

3. What types of properties have been affected the most?(Niraj Bista)
   3.1. Please propose a reasonable indicator(s) to quantify individual listings’ performance.

Unique neighborhoods:

```
 [1] "East Boston"      "Roxbury"          "Beacon Hill"      "Back Bay"         "North End"        "Dorchester"       "South End"
 [8] "Charlestown"      "Jamaica Plain"    "Allston-Brighton" "South Boston"     "West Roxbury"     "Mattapan"         "Roslindale"
[15] "Downtown"         "Government Center" "Mission Hill"    "Fenway/Kenmore"   "West End"         "Chinatown"        "Hyde Park"
[22] "Cambridge"        "Theater District" "Leather District" "Downtown Crossing" "Financial District" "Chestnut Hill"  "Somerville"
[29] "Harvard Square"   "Newton"           "Brookline"        "Revere"           "Everett"          "Winthrop"         NA
[36] "Chelsea"
```

Unique room types:

```
$room_type
[1] "Entire home/apt" "Private room"    "Shared room"     "Hotel room"
```

Minimum night's stay:

```
  [1]   28    3   91   29    1    2    6   32    5    7   30   31    4   10   60  180    9  100   33   20   25 1000    8   45   14  150   92   90  110  600  153   11
 [33]   40  400   59   21   50  271  365  120   55  240   15  182   35   23   56
```

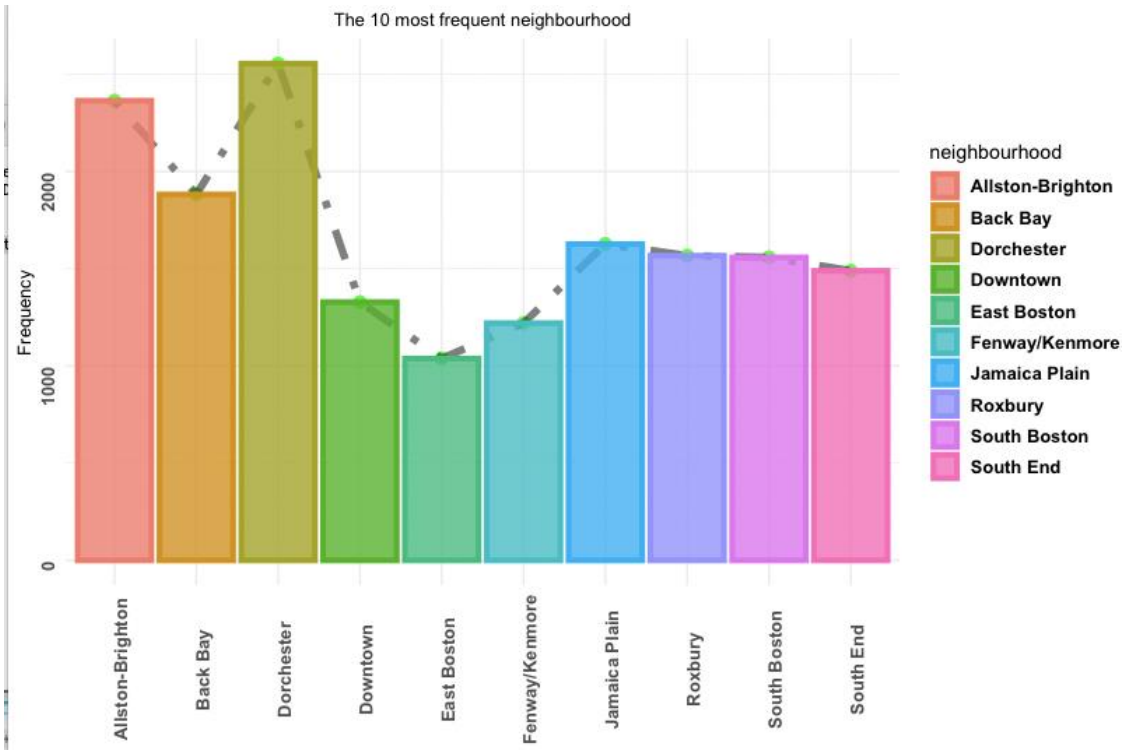
The number of rental Airbnb's in each neighborhood and the percentage distribution between them.
Most in Dorchester 11.45% and least in Chelsea 0.004%

	Frequency	Percent
Chelsea	1	0.004485311
Newton	1	0.004485311
Winthrop	1	0.004485311
Everett	2	0.008970621
Harvard Square	2	0.008970621
Chestnut Hill	6	0.026911864
Revere	6	0.026911864
Cambridge	7	0.031397174
Leather District	9	0.040367795
Government Center	25	0.112132765
Somerville	25	0.112132765
Brookline	28	0.125588697
Financial District	105	0.470957614
Theater District	204	0.915003364
West Roxbury	232	1.040592061
Hyde Park	233	1.045077372
Downtown Crossing	251	1.125812963
Mattapan	308	1.381475667
Roslindale	422	1.892801076
Chinatown	425	1.906257008
Charlestown	464	2.081184122
North End	550	2.466920834
West End	626	2.807804440
Mission Hill	860	3.857367123
Beacon Hill	874	3.920161471
East Boston	1038	4.655752411
Fenway/Kenmore	1220	5.472078941
Downtown	1328	5.956492487
South End	1489	6.678627495
South Boston	1558	6.988113927
Roxbury	1567	7.028481722
Jamaica Plain	1627	7.297600359
Back Bay	1882	8.441354564
Allston-Brighton	2364	10.603274277
Dorchester	2555	11.459968603

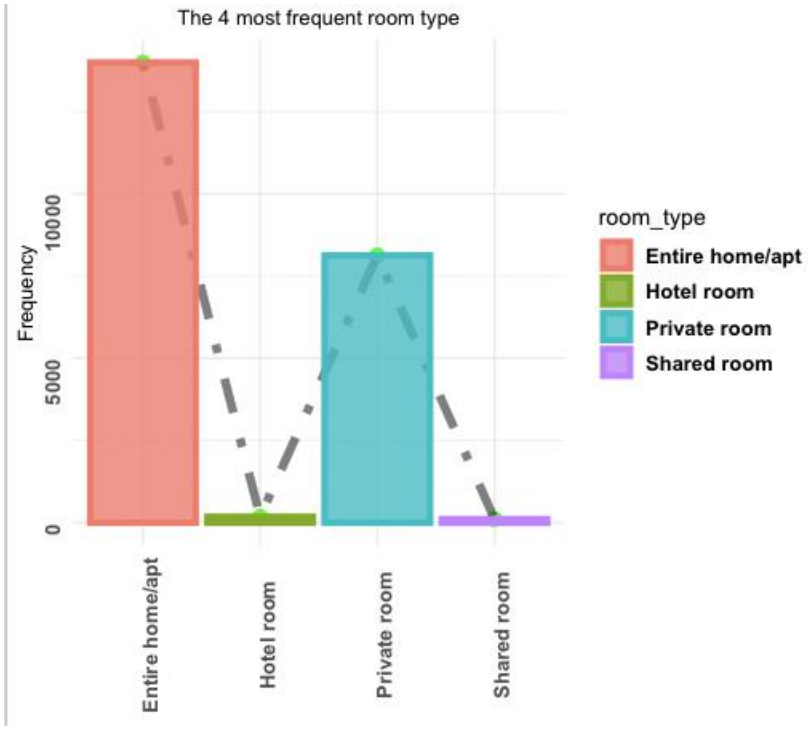
Most of the Airbnb rentals are entire home/apartment with a 62.41% and private room with a 36.25%

	Frequency	Percent
Shared room	113	0.5039918
Hotel room	185	0.8251193
Private room	8129	36.2561884
Entire home/apt	13994	62.4147005

The 10 most frequent neighborhoods.



The 4 most frequent room types.



Reference Material

https://www.airdna.co/airdna-glossary-of-metric-definitions

code for Market Supply and Demand BEGINS

Listing Merge.R

```
#install.packages('lubridate')

#install.packages("dplyr")

library(lubridate)

library(dplyr)

path <- "~/Dropbox/MBAD 6211/rStudio/GroupProject/Data"

dfJan <- read.csv(file.path(path, "listings_jan.csv"), header=TRUE)

dfFeb <- read.csv(file.path(path, "listings_feb.csv"), header=TRUE)

dfMar <- read.csv(file.path(path, "listings_mar.csv"), header=TRUE)

dfApr <- read.csv(file.path(path, "listings_apr.csv"), header=TRUE)

dfMay <- read.csv(file.path(path, "listings_may.csv"), header=TRUE)

dfJun <- read.csv(file.path(path, "listings_jun.csv"), header=TRUE)

dfJul <- read.csv(file.path(path, "listingsJul.csv"), header=TRUE)

dfAug <- read.csv(file.path(path, "listings_aug.csv"), header=TRUE)

dfSep <- read.csv(file.path(path, "listings_sep.csv"), header=TRUE)

dfOct <- read.csv(file.path(path, "listings_oct2.csv"), header=TRUE)

dfNov <- read.csv(file.path(path, "listings_nov.csv"), header=TRUE)

dfDec <- read.csv(file.path(path, "listings_dec.csv"), header=TRUE)

#summary(df1)

#str(df1)

#df1 <- merge(dfJan, dfFeb, dfMar, dfApr, dfMay,

#               dfJun, dfJul,all=TRUE)

df1 <- merge(dfJan, dfFeb,all=TRUE)

df2 <- merge(dfMar, dfApr,all=TRUE)

df3 <- merge(dfJun, dfJul,all=TRUE)

df6 <- merge(df1, df2,all=TRUE)
```

```
df7 <- merge(df2,dfMay,all=TRUE)

df7<- merge(df6, df7, all=TRUE)

dfTot<-merge(df7,df3,all=TRUE)

#, dfAug, dfSep, dfOct, dfNov, dfDec)

#dfTot <- dfTot[order(dfTot$listing_id),]

write.csv(dfTot,file.path(path, "listings_test_1.csv"), row.names = TRUE)
```

CalMerge.R

```
#install.packages('lubridate')
#install.packages("dplyr")


library(lubridate)
library(dplyr)


path <- "~/Dropbox/MBAD 6211/rStudio/GroupProject/Data"


df1 <- read.csv(file.path(path, "calendarJan.csv"), na.strings=c("NA",""), header=TRUE)
df2 <- read.csv(file.path(path, "calendarFeb.csv"), na.strings=c("NA",""), header=TRUE)
df3 <- read.csv(file.path(path, "calendarMar.csv"), na.strings=c("NA",""), header=TRUE)
df4 <- read.csv(file.path(path, "calendarApr.csv"), na.strings=c("NA",""), header=TRUE)
df5 <- read.csv(file.path(path, "calendarMay.csv"), na.strings=c("NA",""), header=TRUE)
df6 <- read.csv(file.path(path, "calendarJun.csv"), na.strings=c("NA",""), header=TRUE)
df7 <- read.csv(file.path(path, "calendarJul.csv"), na.strings=c("NA",""), header=TRUE)
df8 <- read.csv(file.path(path, "calendarAug.csv"), na.strings=c("NA",""), header=TRUE)
df9 <- read.csv(file.path(path, "calendarSep.csv"), na.strings=c("NA",""), header=TRUE)
df10 <- read.csv(file.path(path, "calendarOct.csv"), na.strings=c("NA",""), header=TRUE)
df11 <- read.csv(file.path(path, "calendarNov.csv"), na.strings=c("NA",""), header=TRUE)
df12 <- read.csv(file.path(path, "calendarDec.csv"), na.strings=c("NA",""), header=TRUE)



dfJan <- df1[df1$date >= "2020-01-01" & df1$date < "2020-02-13",]
dfFeb <- df2[df2$date >= "2020-02-13" & df2$date < "2020-03-16",]
dfMar <- df3[df3$date >= "2020-03-16" & df3$date < "2020-04-14",]
dfApr <- df4[df4$date >= "2020-04-14" & df4$date < "2020-05-12",]
dfMay <- df5[df5$date >= "2020-05-12" & df5$date < "2020-06-10",]
dfJun <- df6[df6$date >= "2020-06-10" & df6$date < "2020-07-11",]
```

```
dfJul <- df7[df7$date >= "2020-07-11" & df7$date < "2020-08-31",]
dfAug <- df8[df8$date >= "2020-08-31" & df8$date < "2020-09-28",]
dfSep <- df9[df9$date >= "2020-09-28" & df9$date < "2020-10-24",]
dfOct <- df10[df10$date >= "2020-10-24" & df10$date < "2020-11-10",]
dfNov <- df11[df11$date >= "2020-11-10" & df11$date < "2020-12-21",]
dfDec <- df12[df12$date >= "2020-12-21" & df12$date < "2021-01-19",]

summary(df1)
str(df1)

dfTot <- rbind(dfJan, dfFeb, dfMar, dfApr, dfMay,
               dfJun, dfJul)
#, dfAug, dfSep, dfOct, dfNov, dfDec)

dfTot <- dfTot[order(dfTot$listing_id),]

write.csv(dfTot,file.path(path, "calendarTot3.csv"), row.names = TRUE)
```

Calendar Plots Final.R

```
#install.packages('flexdashboard')
#install.packages('tidyverse')
#install.packages('ggplot2')
#install.packages('lubridate')
#install.packages('ggthemes')
#devtools::install_github("jayjacobs/ggcal") #install.packages('ggcal')
#install.packages('plotly')
#install.packages('sf')
#install.packages('tmap')
#install.packages('DT')
#install.packages('readr')
library(tidyverse)
library(ggplot2)
library(lubridate)
library(ggthemes)
#install.packages('ggcal')
library(ggcal)
library(plotly)
library(sf)
```

```
library(tmap)
library(DT)
library(readr)
library(stringr)

path <- "~/Dropbox/MBAD 6211/rStudio/GroupProject/Data"

read_csv(file.path(path, "calendarTot3.csv")) %>%
 mutate(price = as.numeric(str_extract(price,'[0-9.]+'))) ->
 calendar
read_csv(file.path(path, "listings_test_1.csv")) %>%
 st_as_sf(coords=c('longitude','latitude'),crs=4326) %>%
 mutate(FullyBooked=availability_365==0) ->
 listings
calendar$available <- as.character(calendar$available)
calendar$available [calendar$available == "TRUE"] <- '1'
calendar$available [calendar$available == "FALSE"] <- '0'
calendar$available <- as.numeric(calendar$available)
avail_length <- function(lseq) {
 rl <- rle(lseq)
 rl <- rl$lengths[rl$values]
 rl <- mean(rl)
 ifelse(is.nan(rl),0,rl)
}

calendar %>% group_by(listing_id) %>%
 summarise(al = sum(available)) -> avail

listings %>%
 left_join(avail,by=c("id"="listing_id")) ->
 listings



calendar %>% count(date,wt=available) %>%
 rename(`Available Properties`=n,Date=date) %>%
 mutate(`Week Day`=wday(Date,label=TRUE),Week=week(Date))  -> res
ggcal(res$Date,res$`Available Properties`) + scale_fill_viridis_c()
```

```
res %>%
 filter(Week < 53) %>%
 ggplot(aes(x=Week,y=`Available Properties`,fill=`Week Day`)) +
 geom_col(position='fill') + scale_fill_hue()

res %>% ggplot(aes(x=Date,y=`Available Properties`)) + geom_line(col='red')

calendar %>% count(date,wt=!available) %>%
 rename(`Unavailable Properties`=n,Date=date) %>%
 mutate(`Week Day`=wday(Date,label=TRUE),Week=week(Date))  -> res
ggcal(res$Date,res$`Unavailable Properties`) + scale_fill_viridis_c()

res %>%
 filter(Week < 53) %>%
 ggplot(aes(x=Week,y=`Unavailable Properties`,fill=`Week Day`)) +
 geom_col(position='fill') + scale_fill_hue()

res %>% ggplot(aes(x=Date,y=`Unavailable Properties`)) + geom_line(col='red')
```

- **# Code Market Supply and Demand ENDS**

❑ Code for Most affected properties analysis begins

```
# Installs required pacman packages if needed.

if (!require("pacman")) install.packages("pacman")


# Use pacman to load add-on packages as desired

pacman::p_load(pacman, tidyr, purrr, dplyr, readr, plyr, ggplot2)


setwd("~/Desktop/R/proj/data/")

listings_files_dir <- "listings"

listings_files <- list.files(path = listings_files_dir, pattern = "*.csv.gz", full.names = TRUE)

listings_files


df_listings <- ldply(listings_files, read_csv)


df_listings <- select(df_listings, neighbourhood, latitude, longitude, room_type, price, minimum_nights,
reviews_per_month, host_id, id, last_review)

head(df_listings, 10)


c(unique(df_listings["neighbourhood"]))

c(unique(df_listings["room_type"]))

c(unique(df_listings["minimum_nights"]))


# The number of rental airbnbs in each neighbourhood and the percentage distribution between them.

# Most in Dorchester  11.45% and least in Chelsea 0.004%

freq_location <- data.frame(cbind(Frequency = table(df_listings$neighbourhood), Percent =
prop.table(table(df_listings$neighbourhood)) * 100))

freq_location <- freq_location[order(freq_location$Frequency), ]

freq_location


freq_area <- data.frame(cbind(Frequency = table(df_listings$neighbourhood), Percent =
prop.table(table(df_listings$neighbourhood)) * 100))

freq_area <- freq_area[order(freq_area$Frequency), ]

freq_area


# Most of the airbnb rentals are entire home/apartment with a 62.41% and private room with a 36.25%

freq_type <- data.frame(cbind(Frequency = table(df_listings$room_type), Percent =
prop.table(table(df_listings$room_type)) * 100))

freq_type <- freq_type[order(freq_type$Frequency), ]

freq_type
```

```
tema <- theme(
   plot.title = element_text(size = 10, hjust = .5),
   axis.text.x = element_text(size = 10, angle = 90, face = "bold"),
   axis.text.y = element_text(size = 10, angle = 90, face = "bold"),
   axis.title.x = element_text(size = 10),
   axis.title.y = element_text(size = 10),
   legend.text = element_text(size = 10, face = "bold")
)

df_listings <- data.frame(neighbourhood = row.names(tail(freq_area, 10)), Frequency = tail(freq_area,
10)$Frequency)

options(repr.plot.width = 8, repr.plot.height = 2)
ggplot(data = df_listings, mapping = aes(x = neighbourhood, y = Frequency)) +
   theme_minimal() +
   geom_point(size = 3, color = "green") +
   ggtitle("The 10 most frequent neighbourhood") +
   xlab("") +
   geom_line(color = "black", size = 2, linetype = 16, group = 1, alpha = .5) +
   geom_bar(stat = "identity", mapping = aes(fill = neighbourhood, color = neighbourhood), alpha = .8,
size = 1.5) +
   tema


df_listings <- data.frame(room_type = row.names(tail(freq_type, 4)), Frequency = tail(freq_type,
4)$Frequency)

options(repr.plot.width = 8, repr.plot.height = 2)
ggplot(data = df_listings, mapping = aes(x = room_type, y = Frequency)) +
   theme_minimal() +
   geom_point(size = 3, color = "green") +
   ggtitle("The 4 most frequent room type") +
   xlab("") +
   geom_line(color = "black", size = 2, linetype = 16, group = 1, alpha = .5) +
   geom_bar(stat = "identity", mapping = aes(fill = room_type, color = room_type), alpha = .8, size =
1.5) +
   tema
```

- ❑ **Code for Most affected properties analysis ends**

- **Code for Customer comments such as topics, sentiments, etc. begins**

```
library(dbplyr)
library(dplyr)
library(tidyverse)
library(magrittr)
library(sqldf)
library(tidytext)
library(wordcloud)
library(reshape2)
library(syuzhet)
library(textdata)
library(ggplot2)
library(gridExtra)
df <- read.csv("reviews.csv")
# df<-df%>%mutate(date=as.Date(date,format="%Y-%m-%d"))
# claimsData<-claimsData %>% mutate(IncidentDate=as.Date(IncidentDate, format="%m/%d/%Y"))%>%
summary(df)
# subset(df,format(as.Date(date),"%Y")==2020)
# filter.date(df$date,date.start="2020-01-01",date.end="2020-06-30")
new_df <- sqldf("select listing_id, id,date,reviewer_id,comments
             from df
             where date>='2020-01-01' AND date<='2020-06-30'
             ")
jan <- sqldf("select listing_id, id,date,reviewer_id,comments
             from new_df
             where date>='2020-01-01' AND date<='2020-01-31'
             ")
feb <- sqldf("select listing_id, id,date,reviewer_id,comments
             from new_df
             where date>='2020-02-01' AND date<='2020-02-28'
             ")
mar <- sqldf("select listing_id, id,date,reviewer_id,comments
             from new_df
             where date>='2020-03-01' AND date<='2020-03-31'
             ")

apr <- sqldf("select listing_id, id,date,reviewer_id,comments
             from new_df
```

```
              where date>='2020-04-01' AND date<='2020-04-30'
              ")

may <- sqldf("select listing_id, id,date,reviewer_id,comments
              from new_df
              where date>='2020-04-01' AND date<='2020-05-31'
              ")

june <- sqldf("select listing_id, id,date,reviewer_id,comments
              from new_df
              where date>='2020-06-01' AND date<='2020-06-30'
              ")
jan$month <- "Jan"
feb$month <- "Feb"
mar$month <- "Mar"
apr$month <- "Apr"
may$month <- "May"
june$month <- "June"

new_data <- rbind(jan, feb, mar, apr, may, june)
new_data$line <- 1:nrow(new_data)
new_data$line <- as.character(new_data$line)

jan %>%
 unnest_tokens(word, comments) %>%
 anti_join(stop_words) %>% # remove stop words
 count(word, sort = T) %>%
 mutate(word = str_extract(word, "[a-z]+")) %>%
 na.omit() %>%
 # unnest_tokens(word,comments,token="ngrams",n=2)
 top_n(30) %>%
 # filter(n > 5) %>% #Extract words with frequencies > 20
 ggplot(., aes(reorder(word, n), n)) +
 geom_bar(stat = "identity", fill = "light blue") +
 coord_flip() +
 ylab("Score") +
 xlab("Words") +
 ggtitle("Word Frequency") +
```

```
 theme(
   plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
   axis.title.x = element_text(face = "bold", size = 13),
   axis.title.y = element_text(face = "bold", size = 13)
 )

# Get the sentiments Variation
new_data %>%
 unnest_tokens(word, comments) %>%
 anti_join(stop_words) %>%
 inner_join(get_sentiments("bing")) %>%
 count(month, index = line, sentiment) %>%
 spread(sentiment, n, fill = 0) %>%
 mutate(sentiment = positive - negative) %>%
 ggplot(., aes(index, sentiment, fill = month)) +
 geom_col(show.legend = FALSE, width = 3) +
 facet_wrap(~month, ncol = 18, scales = "free_x") +
 ggtitle("Sentiments Variation") +
 theme(
   plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
   axis.title.x = element_text(face = "bold", size = 13),
   axis.title.y = element_text(face = "bold", size = 13)
 ) +
 theme_dark()

# plot a comparison of positive and negative words used by reviewers
new_data %>%
 unnest_tokens(word, comments) %>%
 anti_join(stop_words) %>%
 inner_join(get_sentiments("bing")) %>%
 group_by(sentiment, month) %>%
 count(word) %>%
 top_n(10) %>%
 ggplot(., aes(reorder(word, n), n, fill = sentiment)) +
 geom_col(show.legend = T) +
 coord_flip() +
 facet_wrap(~sentiment, scales = "free_y") +
 facet_wrap(~month, scales = "free_y")
```

```
xlab("Words") +
 ylab("frequency") +
 ggtitle("Word Usage") +
 theme(
   plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
   axis.title.x = element_text(face = "bold", size = 13),
   axis.title.y = element_text(face = "bold", size = 13)
 ) + theme_dark()


new_data %>%
 unnest_tokens(word, comments) %>%
 mutate(word = gsub("problems", "problem", word)) %>%
 inner_join(get_sentiments("bing")) %>%
 count(word, sentiment) %>%
 acast(word ~ sentiment, value.var = "n", fill = 0) %>%
 comparison.cloud(
   color = c("red", "green"),
   max.words = 100
 ) + theme_dark()


# %>%ggplot()+facet_wrap(~month, scales = "free_y")
# Note the acast function is from the reshape2 package
# Functions such as comparison.cloud() require you to turn the data frame into a matrix with reshape2's
acast()


# nrc sentiment
h6 <- june %>%
 # group_by(month='June')%>%
 unnest_tokens(word, comments) %>%
 right_join(get_sentiments("nrc")) %>%
 anti_join(stop_words) %>%
 mutate(word = str_extract(word, "[a-z]+")) %>%
 na.omit() %>%
 # group_by(month) %>%
 count(sentiment, sort = TRUE) %>%
 ggplot(., aes(n, sentiment)) +
 geom_col() +
 guides(fill = FALSE) +
```

```
 # facet_wrap(~month, ncol = 18, scales = "free_x") +
 ggtitle("June sentiment") +
 theme(
   plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
   axis.title.x = element_text(face = "bold", size = 13),
   axis.title.y = element_text(face = "bold", size = 13),
   # panel.background = element_blank(fill='black'),
   legend.background = element_rect(fill = "black", color = NA)
 ) +
 theme_classic()

h5 <- may %>%
 # group_by(month='June')%>%
 unnest_tokens(word, comments) %>%
 right_join(get_sentiments("nrc")) %>%
 anti_join(stop_words) %>%
 mutate(word = str_extract(word, "[a-z]+")) %>%
 na.omit() %>%
 # group_by(month='June') %>%
 count(sentiment, sort = TRUE) %>%
 ggplot(., aes(n, sentiment)) +
 geom_col() +
 guides(fill = FALSE) +
 # facet_wrap(~month, ncol = 18, scales = "free_x") +
 ggtitle("May sentiment") +
 theme(
   plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
   axis.title.x = element_text(face = "bold", size = 13),
   axis.title.y = element_text(face = "bold", size = 13)
 )

h4 <- apr %>%
 # group_by(month='June')%>%
 unnest_tokens(word, comments) %>%
 right_join(get_sentiments("nrc")) %>%
 anti_join(stop_words) %>%
 mutate(word = str_extract(word, "[a-z]+")) %>%
 na.omit() %>%
```

```
 # group_by(month='June') %>%
 count(sentiment, sort = TRUE) %>%
 ggplot(., aes(n, sentiment)) +
 geom_col() +
 guides(fill = FALSE) +
 # facet_wrap(~month, ncol = 18, scales = "free_x") +
 ggtitle("April sentiment") +
 theme(
   plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
   axis.title.x = element_text(face = "bold", size = 13),
   axis.title.y = element_text(face = "bold", size = 13)
 )

h3 <- mar %>%
 # group_by(month='June')%>%
 unnest_tokens(word, comments) %>%
 right_join(get_sentiments("nrc")) %>%
 anti_join(stop_words) %>%
 mutate(word = str_extract(word, "[a-z]+")) %>%
 na.omit() %>%
 # group_by(month='June') %>%
 count(sentiment, sort = TRUE) %>%
 ggplot(., aes(n, sentiment)) +
 geom_col() +
 guides(fill = FALSE) +
 # facet_wrap(~month, ncol = 18, scales = "free_x") +
 ggtitle("March sentiment") +
 theme(
   plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
   axis.title.x = element_text(face = "bold", size = 13),
   axis.title.y = element_text(face = "bold", size = 13)
 )

h2 <- feb %>%
 # group_by(month='June')%>%
 unnest_tokens(word, comments) %>%
 right_join(get_sentiments("nrc")) %>%
 anti_join(stop_words) %>%
```

```
 mutate(word = str_extract(word, "[a-z]+")) %>%
 na.omit() %>%
 # group_by(month='June') %>%
 count(sentiment, sort = TRUE) %>%
 ggplot(., aes(n, sentiment)) +
 geom_col() +
 guides(fill = FALSE) +
 # facet_wrap(~month, ncol = 18, scales = "free_x") +
 ggtitle("Feb sentiment") +
 theme(
   plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
   axis.title.x = element_text(face = "bold", size = 13),
   axis.title.y = element_text(face = "bold", size = 13)
 )
h1 <- jan %>%
 # group_by(month='June')%>%
 unnest_tokens(word, comments) %>%
 right_join(get_sentiments("nrc")) %>%
 anti_join(stop_words) %>%
 mutate(word = str_extract(word, "[a-z]+")) %>%
 na.omit() %>%
 # group_by(month='June') %>%
 count(sentiment, sort = TRUE) %>%
 ggplot(., aes(n, sentiment)) +
 geom_col() +
 guides(fill = FALSE) +
 # facet_wrap(~month, ncol = 18, scales = "free_x") +
 ggtitle("Jan sentiment") +
 theme(
   plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
   axis.title.x = element_text(face = "bold", size = 13),
   axis.title.y = element_text(face = "bold", size = 13)
 )
# multiplot(h1, h2, h3, h4,h5,h6, cols=2)
grid.arrange(h1, h2, h3, h4, h5, h6, ncol = 2)
```

❑ **Code for Customer comments such as topics, sentiments, etc. ends**

❑ Code for Factors affecting market “exit” decisions begins

```
library(dbplyr)
library(dplyr)
library(tidyverse)
library(magrittr)
# install.packages('sqldf')
library(sqldf)
library(tidytext)
# install.packages('wordcloud')
library(wordcloud)
library(reshape2)
# install.packages('syuzhet')
library(syuzhet)
# install.packages('textdata')
library(textdata)
library(ggplot2)
library(gridExtra)

getwd()
setwd("~/Mangala_Local/DSBA_6211/Project/Consolidated")

df_jan <- read.csv("jan_listings.csv")
df_feb <- read.csv("feb_listings.csv")
df_mar <- read.csv("mar_listings.csv")
df_apr <- read.csv("apr_listings.csv")
df_may <- read.csv("may_listings.csv")
df_june <- read.csv("june_listings.csv")

df_jan$last_review <- as.Date(df_jan$last_review)
df_jan$first_review <- as.Date(df_jan$first_review)
df_feb$last_review <- as.Date(df_feb$last_review)
df_feb$first_review <- as.Date(df_feb$first_review)
df_mar$last_review <- as.Date(df_mar$last_review)
df_mar$first_review <- as.Date(df_mar$first_review)
df_apr$last_review <- as.Date(df_apr$last_review)
df_apr$first_review <- as.Date(df_apr$first_review)
```

```
df_may$last_review <- as.Date(df_may$last_review)
df_may$first_review <- as.Date(df_may$first_review)
df_june$last_review <- as.Date(df_june$last_review)
df_june$first_review <- as.Date(df_june$first_review)
str(df_jan)
summary(df_jan)

new_df_jan <- sqldf("select id,
               case
                 when has_availability='t' then 1
              when has_availability='f' then 0
             end as availability,
             availability_30,availability_60,availability_90,availability_365,
             number_of_reviews,number_of_reviews_ltm,
             first_review,last_review,
             (last_review-first_review) as lenreview,
             review_scores_rating,review_scores_accuracy,review_scores_cleanliness,
             review_scores_checkin,review_scores_communication,review_scores_location,
             review_scores_value,instant_bookable
           from df_jan
             ")

new_df_jan

new_df_feb <- sqldf("select id,
                case
                when has_availability='t' then 1
                when has_availability='f' then 0
             end as availability,
             availability_30,availability_60,availability_90,availability_365,
             number_of_reviews,number_of_reviews_ltm,
          first_review,last_review,
             (last_review-first_review) as lenreview,
             review_scores_rating,review_scores_accuracy,review_scores_cleanliness,
             review_scores_checkin,review_scores_communication,review_scores_location,
             review_scores_value,instant_bookable
           from df_feb")
```

```
new_df_mar <- sqldf("select id,
               case
                 when has_availability='t' then 1
                 when has_availability='f' then 0
             end as availability,
              availability_30,availability_60,availability_90,availability_365,
             number_of_reviews,number_of_reviews_ltm,
             first_review,last_review,
             (last_review-first_review) as lenreview,
             review_scores_rating,review_scores_accuracy,review_scores_cleanliness,
             review_scores_checkin,review_scores_communication,review_scores_location,
             review_scores_value,instant_bookable
           from df_mar
             ")

new_df_apr <- sqldf("select id,
                case
                  when has_availability='t' then 1
                  when has_availability='f' then 0
             end as availability,
             availability_30,availability_60,availability_90,availability_365,
             number_of_reviews,number_of_reviews_ltm,
             first_review,last_review,
             (last_review-first_review) as lenreview,
             review_scores_rating,review_scores_accuracy,review_scores_cleanliness,
             review_scores_checkin,review_scores_communication,review_scores_location,
             review_scores_value,instant_bookable
           from df_apr
             ")

new_df_may <- sqldf("select id,
                  case
                      when has_availability='t' then 1
                      when has_availability='f' then 0
                  end as availability,
               availability_30,availability_60,availability_90,availability_365,
               number_of_reviews,number_of_reviews_ltm,
               first_review,last_review,
```

```
               (last_review-first_review) as lenreview,
             review_scores_rating,review_scores_accuracy,review_scores_cleanliness,
               review_scores_checkin,review_scores_communication,review_scores_location,
               review_scores_value,instant_bookable
               from df_may
             ")

new_df_june <- sqldf("select id,
           case
                when has_availability='t' then 1
                when has_availability='f' then 0
             end as availability,
             availability_30,availability_60,availability_90,availability_365,
             number_of_reviews,number_of_reviews_ltm,
             first_review,last_review,
             (last_review-first_review) as lenreview,
             review_scores_rating,review_scores_accuracy,review_scores_cleanliness,
           review_scores_checkin,review_scores_communication,review_scores_location,
             review_scores_value,instant_bookable
           from df_june
             ")



new_df_jan$month <- "Jan"
new_df_feb$month <- "Feb"
new_df_mar$month <- "Mar"
new_df_apr$month <- "Apr"
new_df_may$month <- "May"
new_df_june$month <- "June"


new_data <- rbind(
 new_df_jan, new_df_feb, new_df_mar,
 new_df_apr, new_df_may, new_df_june
)

new_data$line <- 1:nrow(new_data)
```

```
new_data$line <- as.character(new_data$line)


library(ggplot2)
library(survival)



km1 <- survfit(Surv(lenreview, availability) ~ 1, data = new_data)
summary(km1)
plot(km1, xlab = "Length of Review", ylab = "Survival Probability for a Property")



# A special visualization package from github
# install.packages('devtools', dependencies = TRUE)

library(devtools)
# devtools::install_github('sachsmc/ggkm')
library(ggkm)



# To test all independent variables

attach(new_data)
X <- cbind(
 availability_30, availability_90, availability_365,
 number_of_reviews, number_of_reviews_ltm,
 first_review, last_review,
 review_scores_rating, review_scores_accuracy,
 review_scores_cleanliness, review_scores_checkin,
 review_scores_communication, review_scores_location,
 review_scores_value, instant_bookable
)
coxph <- coxph(Surv(lenreview, availability) ~ X, method = "breslow", data = new_data)
summary(coxph)
```

- ❑ **Code for Factors affecting market "exit" decisions ends**