Analytical Methods I (ANLY 502)
Course Project: A statistical Analysis on Airbnb Listings for NYC-2019
Summer 2021

Ajay Acharya Kanagala, Jinee Hasmukhbh Patel, Rumana Myageri, Quyen Tuan Ngo

Introduction: Our dataset consists of Airbnb listings for NYC for the year 2019, taken from Kaggle. It has 48895 observations of 16 variables, of which 11 are numerical and 5 are categorical. We will be exploring and analyzing the data to establish any relationships between pricing and other factors/ variables like availability, geographic location, and reviews. Our goal is to develop a data-driven pricing strategy for a host.

Numerical	Categorical
id	name
host_id	host_name
latitude	neighbourhood_group
longitude	neighbourhood
price	room_type
minimum_nights	
number_of_reviews	
last_review	
reviews_per_month	
calculated_host_listings_count	
availability_365	

Exploratory Data Analysis

We begin by importing the data and checking for missing values

```
airbnb_data<-read.csv("C:\\Users\\myage\\OneDrive\\Documents\\Data
Analytics\\ANLY-502\\Project\\AB_NYC.csv")
```

```
head(airbnb_data)
```

```
   id                                          name host_id   host_name
neighbourhood_group
1 2539              Clean & quiet apt home by the park     2787          John
Brooklyn
```

```
2 2595                          Skylit Midtown Castle    2845    Jennifer
Manhattan
3 3647              THE VILLAGE OF HARLEM....NEW YORK !    4632   Elisabeth
Manhattan
4 3831                  Cozy Entire Floor of Brownstone    4869 LisaRoxanne
Brooklyn
5 5022 Entire Apt: Spacious Studio/Loft by central park    7192       Laura
Manhattan
6 5099         Large Cozy 1 BR Apartment In Midtown East    7322       Chris
Manhattan
  neighbourhood latitude longitude       room_type price minimum_nights
number_of_reviews last_review
1    Kensington 40.64749 -73.97237    Private room   149              1
9  2018-10-19
2       Midtown 40.75362 -73.98377 Entire home/apt   225              1
45  2019-05-21
3        Harlem 40.80902 -73.94190    Private room   150              3
0
4  Clinton Hill 40.68514 -73.95976 Entire home/apt    89              1
270  2019-07-05
5   East Harlem 40.79851 -73.94399 Entire home/apt    80             10
9  2018-11-19
6   Murray Hill 40.74767 -73.97500 Entire home/apt   200              3
74  2019-06-22
  reviews_per_month calculated_host_listings_count availability_365
1              0.21                              6              365
2              0.38                              2              355
3                NA                              1              365
4              4.64                              1              194
5              0.10                              1                0
6              0.59                              1              129
>
```

```
summary(airbnb_data)
```

```
       id                name              host_id             host_name
neighbourhood_group
 Min.    :    2539   Length:48895       Min.    :      2438   Length:48895
Length:48895
 1st Qu.: 9471945   Class :character   1st Qu.:  7822033   Class :character
Class :character
 Median :19677284   Mode  :character   Median : 30793816   Mode  :character
Mode  :character
 Mean   :19017143                      Mean   : 67620011
 3rd Qu.:29152178                      3rd Qu.:107434423
 Max.   :36487245                      Max.   :274321313

 neighbourhood         latitude       longitude       room_type
price
 Length:48895       Min.    :40.50   Min.    :-74.24   Length:48895       Min.
:    0.0
 Class :character   1st Qu.:40.69   1st Qu.:-73.98   Class :character   1st
Qu.:   69.0
 Mode  :character   Median :40.72   Median :-73.96   Mode  :character
Median :  106.0
```

```
                    Mean   :40.73   Mean   :-73.95                    Mean
:  152.7
                    3rd Qu.:40.76   3rd Qu.:-73.94                    3rd
Qu.:  175.0
                    Max.   :40.91   Max.   :-73.71                    Max.
:10000.0

 minimum_nights    number_of_reviews last_review        reviews_per_month
calculated_host_listings_count
 Min.   :   1.00   Min.   :  0.00    Length:48895       Min.   : 0.010
Min.   :  1.000
 1st Qu.:   1.00   1st Qu.:  1.00    Class :character   1st Qu.: 0.190    1st
Qu.:  1.000
 Median :   3.00   Median :  5.00    Mode  :character   Median : 0.720
Median :  1.000
 Mean   :   7.03   Mean   : 23.27                       Mean   : 1.373
Mean   :  7.144
 3rd Qu.:   5.00   3rd Qu.: 24.00                       3rd Qu.: 2.020    3rd
Qu.:  2.000
 Max.   :1250.00   Max.   :629.00                       Max.   :58.500
Max.   :327.000
                                                        NA's   :10052
 availability_365
 Min.   :  0.0
 1st Qu.:  0.0
 Median : 45.0
 Mean   :112.8
 3rd Qu.:227.0
 Max.   :365.0
```

```
length(airbnb_data)
[1] 16
```

We will now check for missing values in our dataset:

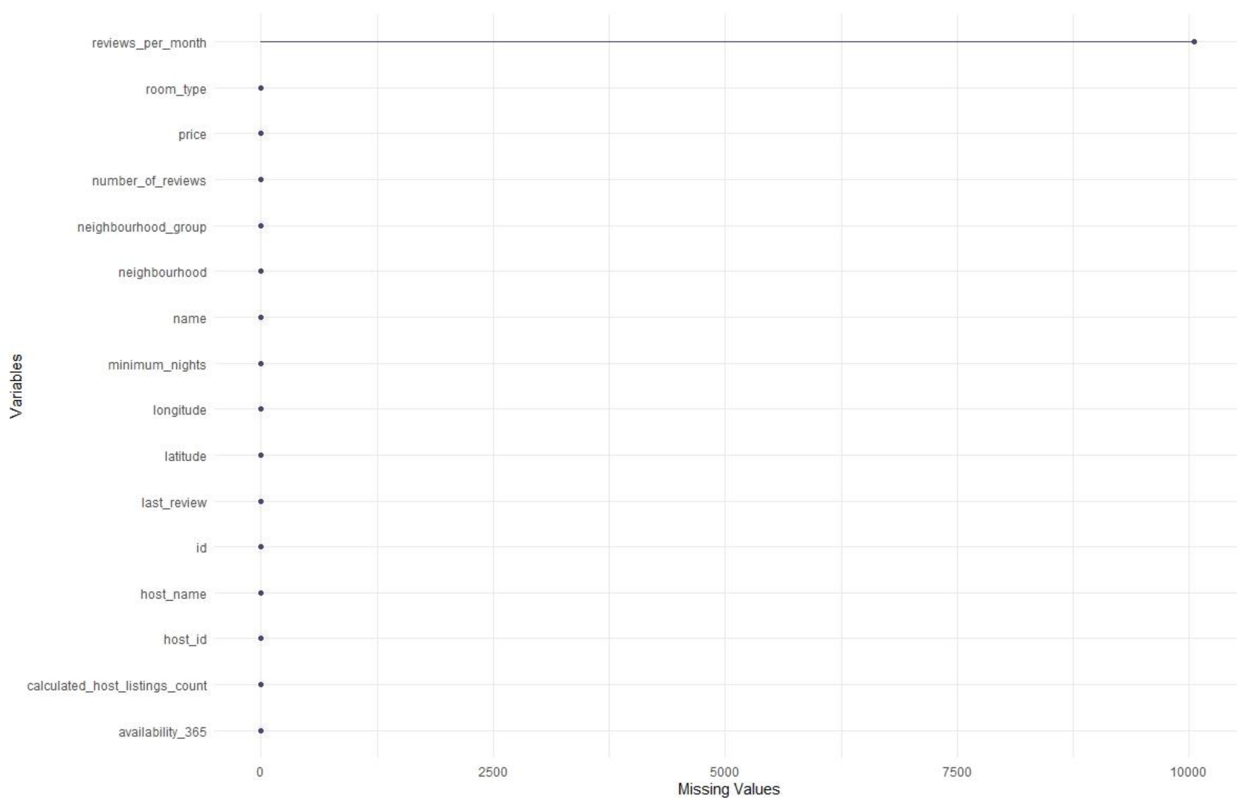
```
summary(is.na(airbnb_data))
```

```
     id               name           host_id         host_name
neighbourhood_group neighbourhood
 Mode :logical    Mode :logical    Mode :logical    Mode :logical    Mode
:logical       Mode :logical
 FALSE:48895      FALSE:48895      FALSE:48895      FALSE:48895      FALSE:48895
FALSE:48895

  latitude        longitude        room_type          price
minimum_nights  number_of_reviews
 Mode :logical    Mode :logical    Mode :logical    Mode :logical    Mode
:logical   Mode :logical
 FALSE:48895      FALSE:48895      FALSE:48895      FALSE:48895      FALSE:48895
FALSE:48895

 last_review      reviews_per_month calculated_host_listings_count
availability_365
```

```
 Mode  :logical   Mode  :logical     Mode  :logical                          Mode  
:logical  
 FALSE:48895      FALSE:38843        FALSE:48895                            FALSE:48895     
                  TRUE :10052                                                               
> 
```

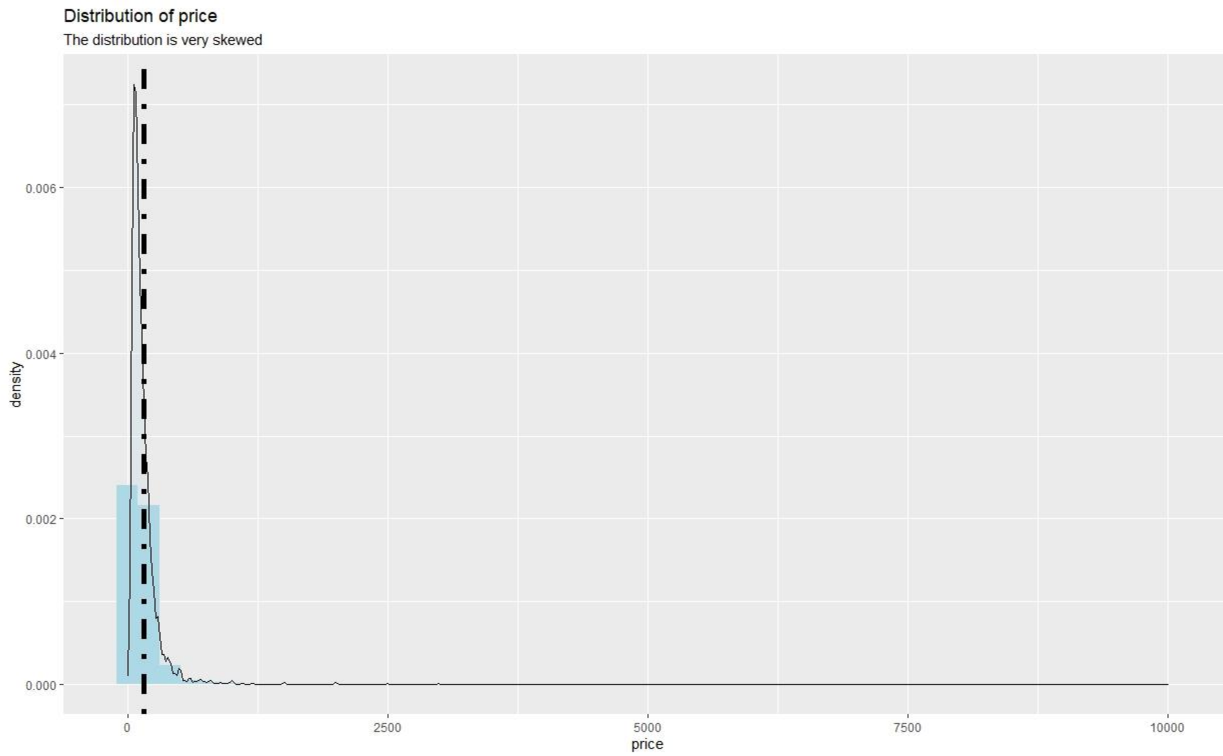
"Reviews Per month" has 10052 missing values.

```
install.packages("naniar")
library(naniar)

naniar::gg_miss_var(airbnb_data) +
  theme_minimal()+
  labs(y = " Missing Values")
```



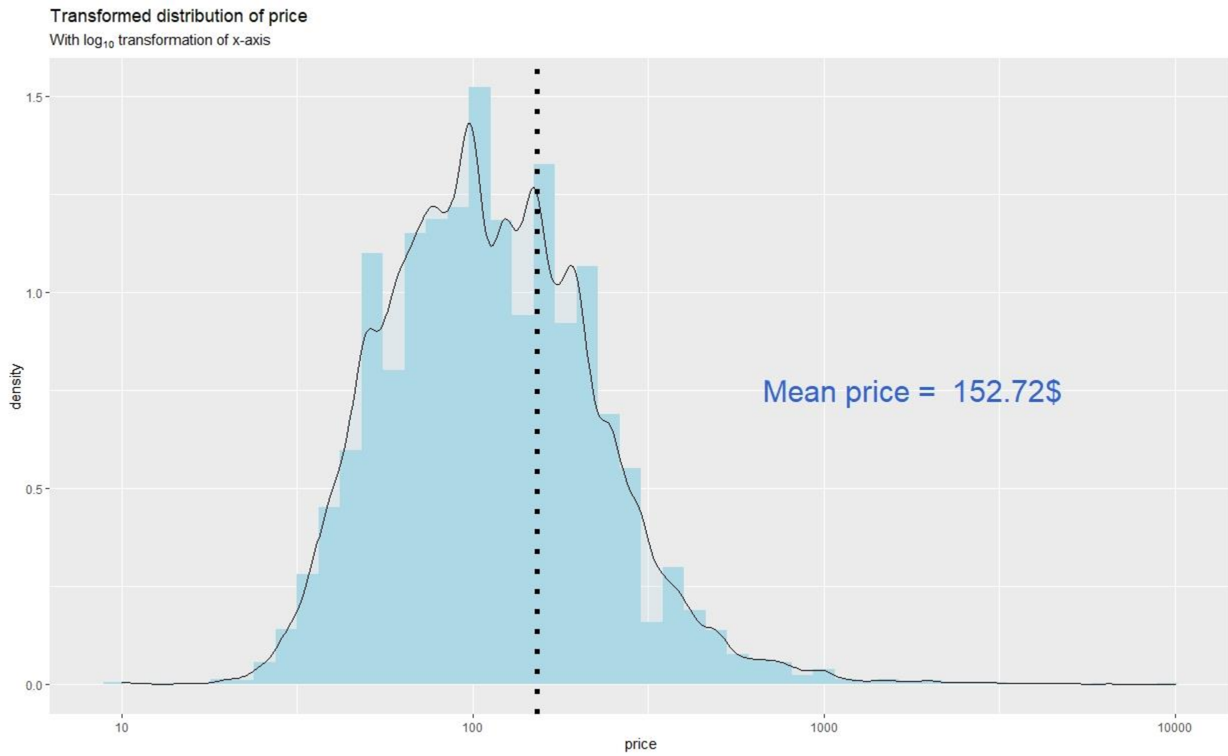
Visualizing price distribution using ggplot.

```
ggplot(airbnb_data, aes(price)) +
  geom_histogram(bins = 50, aes(y = ..density..), fill = "light blue") +
  geom_density(alpha = 0.2, fill = "light blue") +
  ggtitle("Distribution of price",
          subtitle = "The distribution is very skewed") +
  theme(axis.title = element_text(), axis.title.x = element_text()) +
  geom_vline(xintercept = round(mean(airbnb_data$price), 2), size = 2,
linetype = 4)
```



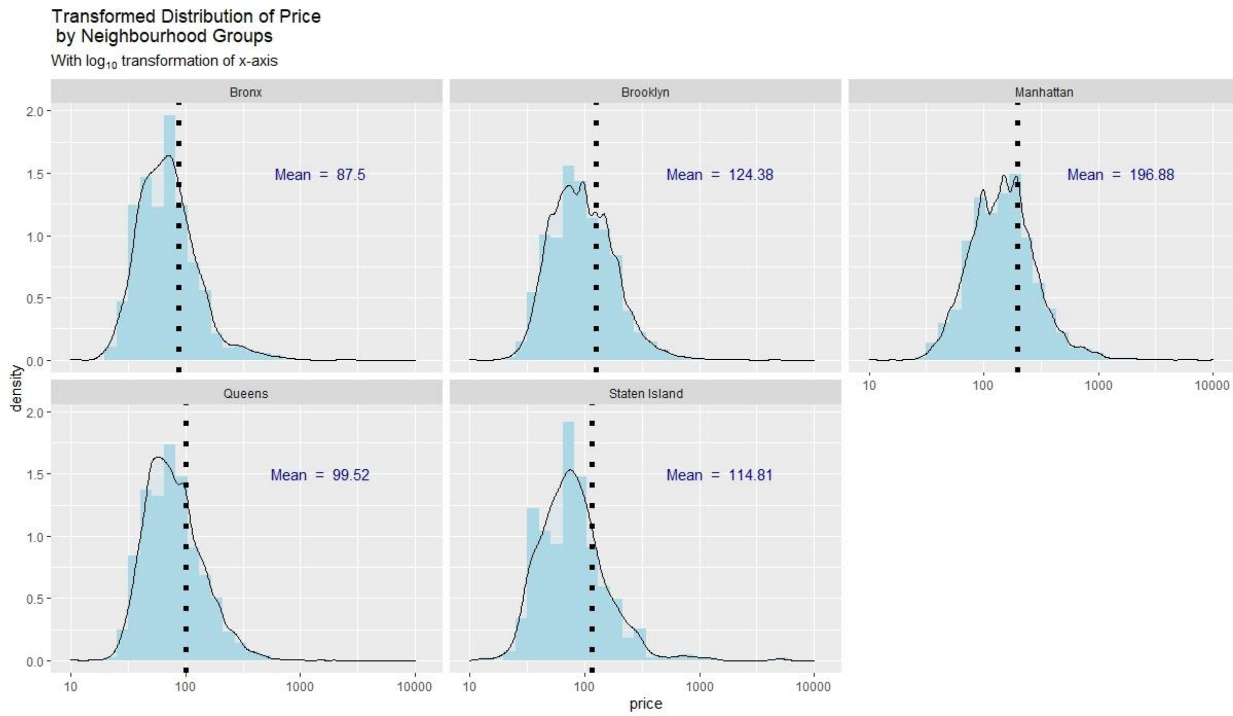
We can see that the price distribution is extremely skewed. We will be using log transformation to better visualize this data.

```
ggplot(airbnb_data, aes(price)) +
  geom_histogram(bins = 50, aes(y = ..density..), fill = "light blue") +
  geom_density(alpha = 0.2, fill = "light blue") +

  ggtitle("Transformed distribution of price",
          subtitle = expression("With" ~'log'[10] ~ "transformation of x-
axis")) +
  #theme(axis.title = element_text(), axis.title.x = element_text()) +
  geom_vline(xintercept = round(mean(airbnb_data$price), 2), size = 2,
linetype = 3) +
  scale_x_log10() +
  annotate("text", x = 1800, y = 0.75,label = paste("Mean price = ",
paste0(round(mean(airbnb_data$price), 2), "$")),
           color =  "#3263CD", size = 8)
```



Transforming price distribution for every neighborhood in NYC for visualization.

```
airbnb_nh <- airbnb_data %>%
  group_by(neighbourhood_group) %>%
  summarise(price = round(mean(price), 2))


ggplot(airbnb_data, aes(price)) +
  geom_histogram(bins = 30, aes(y = ..density..), fill = "light blue") +
  geom_density(alpha = 0.2, fill = "light blue") +
  ggtitle("Transformed Distribution of Price\n by Neighbourhood Groups",
          subtitle = expression("With" ~'log'[10] ~ "transformation of x-
axis")) +
  geom_vline(data = airbnb_nh, aes(xintercept = price), size = 2, linetype =
3) +
  geom_text(data = airbnb_nh,y = 1.5, aes(x = price + 1400, label =
paste("Mean  = ",price)), color = "dark blue", size = 4) +
  facet_wrap(~neighbourhood_group) +
  scale_x_log10()
```



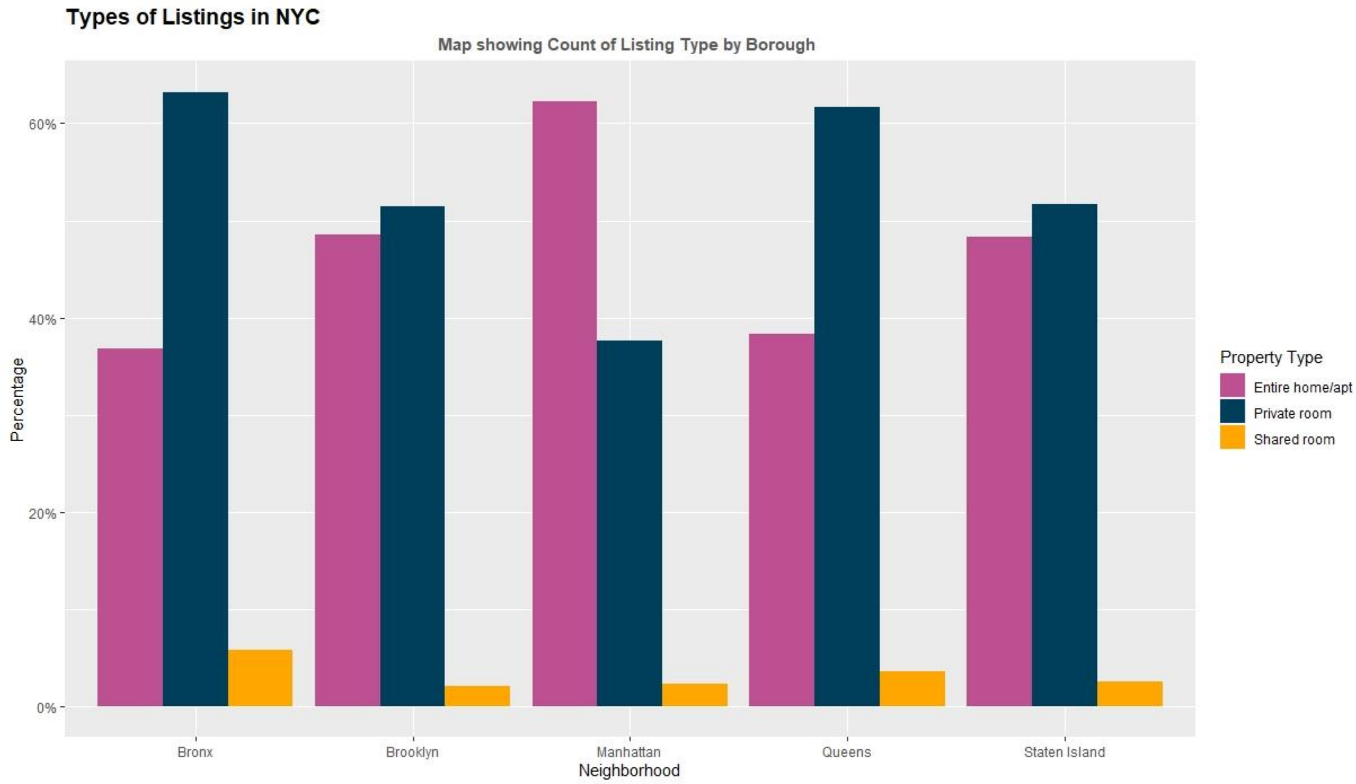
Let's take another look at the mean prices for each neighbourhood in NYC.

```
library(dplyr)


airbnb_data %>%
  filter(!(is.na(neighbourhood_group))) %>%
  filter(!(neighbourhood_group == "Unknown")) %>%
  group_by(neighbourhood_group) %>%
  summarise(mean_price = mean(price, na.rm = TRUE)) %>%
  ggplot(aes(x = reorder(neighbourhood_group, mean_price), y = mean_price,
fill = neighbourhood_group)) +
  geom_col(stat ="identity", color = "black", fill="#357b8a") +
  coord_flip() +
  theme_gray() +
  labs(x = "Neighbourhood Group", y = "Price") +
  geom_text(aes(label = round(mean_price,digit = 2)), hjust = 2.0, color =
"white", size = 3.5) +
  ggtitle("Mean Price comparison for each Neighbourhood Group", subtitle =
"Price vs Neighbourhood Group") +
  xlab("Neighbourhood Group") +
  ylab("Mean Price") +
  theme(legend.position = "none",
        plot.title = element_text(color = "black", size = 14, face = "bold",
hjust = 0.5),
        plot.subtitle = element_text(color = "darkblue", hjust = 0.5),
        axis.title.y = element_text(),
        axis.title.x = element_text(),
        axis.ticks = element_blank())
```

Exploring the types of listings in NYC by grouping them together for each neighbourhood.

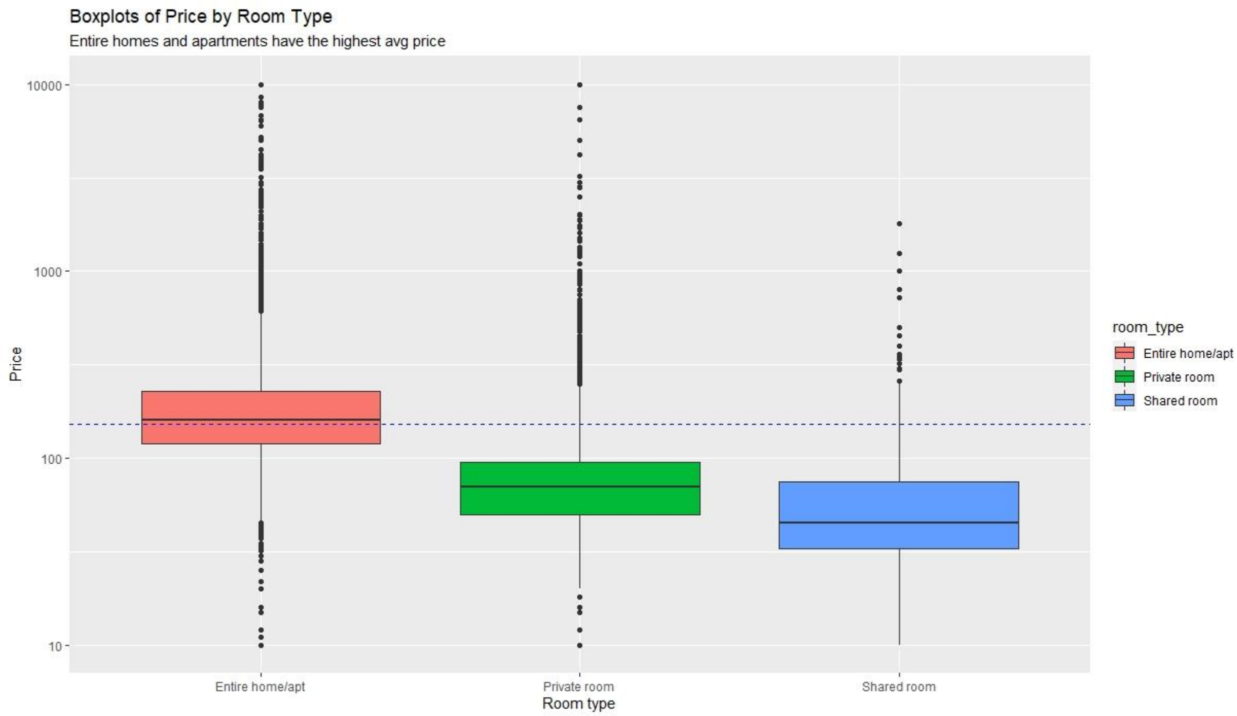
```
property_df <-  airbnb_data %>%
  group_by(neighbourhood_group, room_type) %>%
  summarize(Freq = n())

# propertydf <- propertydf %>%
#   filter(property_type %in%
c("Apartment","House","Condominium","Townhouse", "Loft"))

total_property <-  airbnb_data %>%
  filter(room_type %in% c("Private room","Entire home/apt","Entire
home/apt")) %>%
  group_by(neighbourhood_group) %>%
  summarize(sum = n())

property_ratio <- merge (property_df, total_property,
by="neighbourhood_group")

property_ratio <- property_ratio %>%
  mutate(ratio = Freq/sum)

ggplot(property_ratio, aes(x=neighbourhood_group, y = ratio, fill =
room_type)) +
  geom_bar(position = "dodge", stat="identity") +
  xlab("Borough") + ylab ("Count") +
  scale_fill_discrete(name = "Property Type") +
```

```
  scale_y_continuous(labels = scales::percent) +
  ggtitle("Types of Listings in NYC",
          subtitle = "Map showing Count of Listing Type by Borough ") +
  theme(plot.title = element_text(face = "bold", size = 14) ) +
  theme(plot.subtitle = element_text(face = "bold", color = "grey35", hjust =
0.5)) +
  theme(plot.caption = element_text(color =
"grey68"))+scale_color_gradient(low="#d3cbcb", high="#852eaa")+
  scale_fill_manual("Property Type", values=c("#bc5090","#003f5c", "#ffa600",
"#c299e5", "#056990")) +
  xlab("Neighborhood") + ylab("Percentage")
```



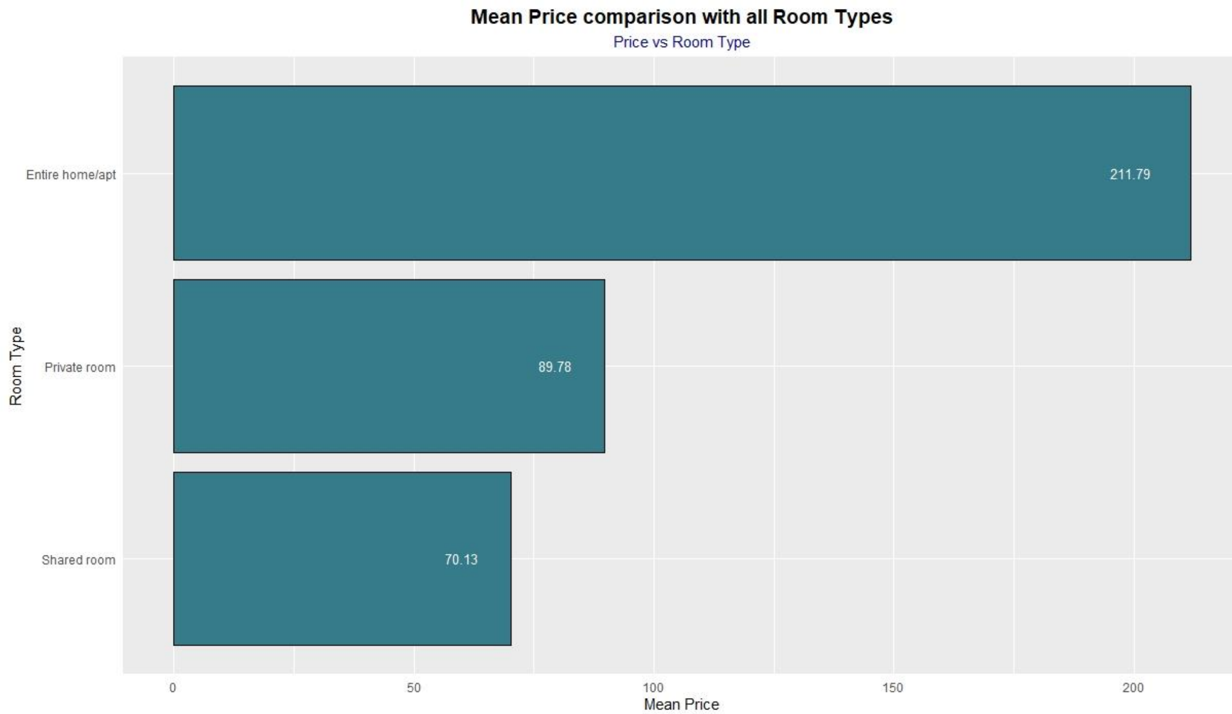
Boxplot of the prices for types of listings

```
ggplot(airbnb_data, aes(x = room_type, y = price)) +
  geom_boxplot(aes(fill = room_type)) + scale_y_log10() +
  xlab("Room type") +
  ylab("Price") +
  ggtitle("Boxplots of Price by Room Type",
          subtitle = "Entire homes and apartments have the highest avg
price") +
  geom_hline(yintercept = mean(airbnb_data$price), color = "blue", linetype =
2)
```



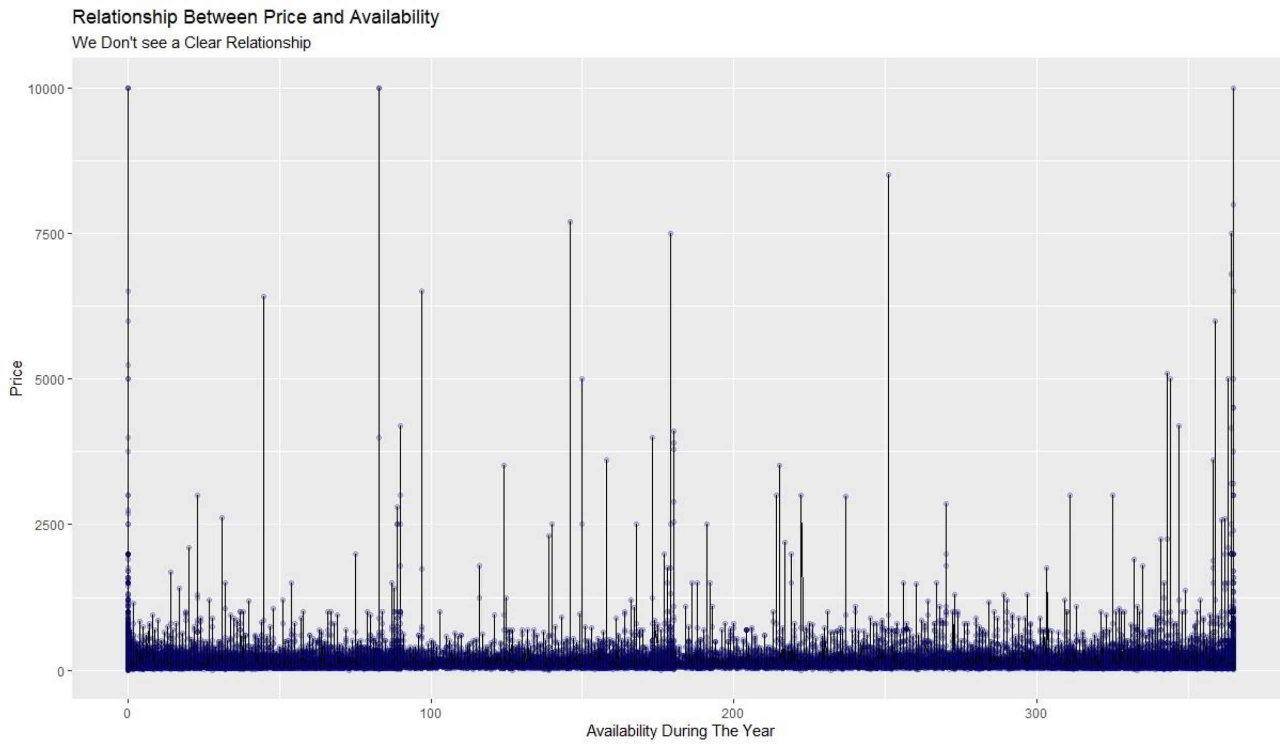
Bargraph to visualize the price for each type of listing.

```
airbnb_data %>%
  filter(!(is.na(room_type))) %>%
  filter(!(room_type == "Unknown")) %>%
  group_by(room_type) %>%
  summarise(mean_price = mean(price, na.rm = TRUE)) %>%
  ggplot(aes(x = reorder(room_type, mean_price), y = mean_price, fill =
room_type)) +
  geom_col(stat ="identity", color = "black", fill="#357b8a") +
  coord_flip() +
  theme_gray() +
  labs(x = "Room Type", y = "Price") +
  geom_text(aes(label = round(mean_price,digit = 2)), hjust = 2.0, color =
"white", size = 3.5) +
  ggtitle("Mean Price comparison with all Room Types", subtitle = "Price vs
Room Type") +
  xlab("Room Type") +
  ylab("Mean Price") +
  theme(legend.position = "none",
        plot.title = element_text(color = "black", size = 14, face = "bold",
hjust = 0.5),
        plot.subtitle = element_text(color = "darkblue", hjust = 0.5),
        axis.title.y = element_text(),
        axis.title.x = element_text(),
        axis.ticks = element_blank())
```



Analyzing the relationship between price and availability

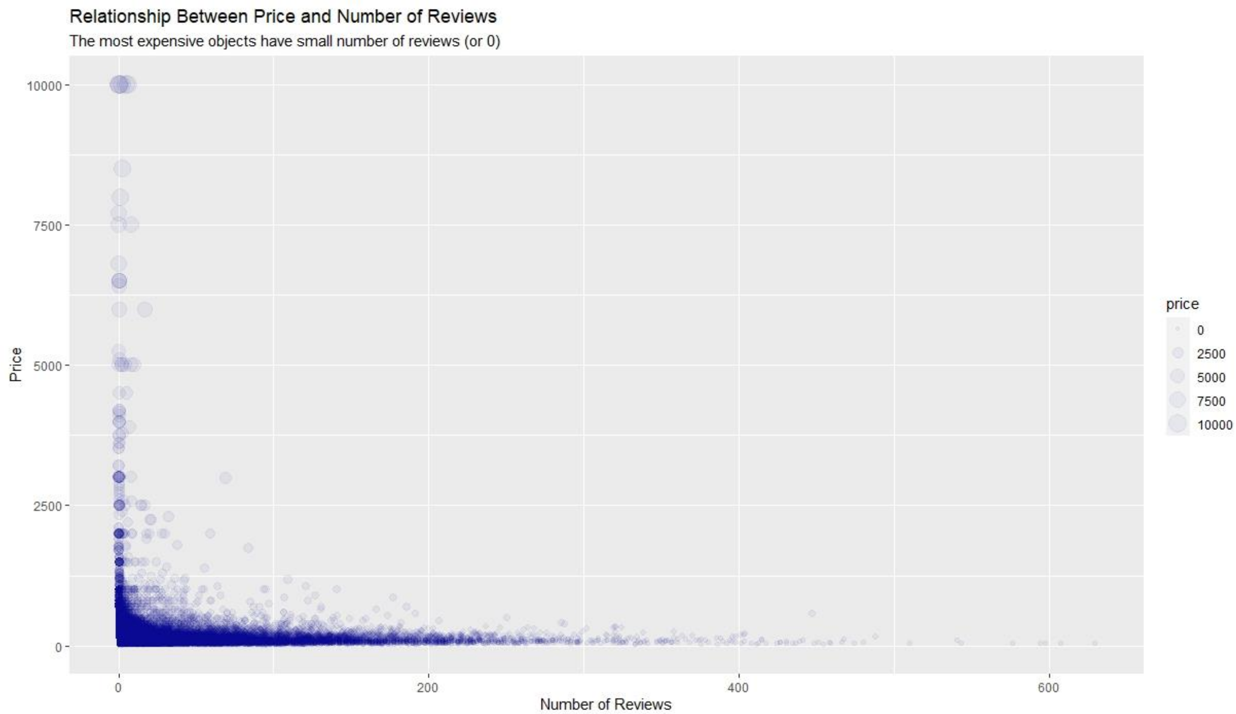
```
ggplot(airbnb_data, aes(availability_365, price)) +
  geom_point(alpha = 0.2, color = "darkblue") +
  geom_density(stat = "identity", alpha = 0.2) +
  xlab("Availability During The Year") +
  ylab("Price") +
  ggtitle("Relationship Between Price and Availability",
          subtitle = "We Don't see a Clear Relationship")
```



We don’t see a clear relationship between the two.

Analyzing the relationship between price and Reviews.

```
ggplot(airbnb_data, aes(number_of_reviews, price)) +
  theme(axis.title = element_text(), axis.title.x = element_text()) +
  geom_point(aes(size = price), alpha = 0.05, color = "darkblue") +
  xlab("Number of Reviews") +
  ylab("Price") +
  ggtitle("Relationship Between Price and Number of Reviews",
          subtitle = "The most expensive objects have small number of reviews
(or 0)")
```

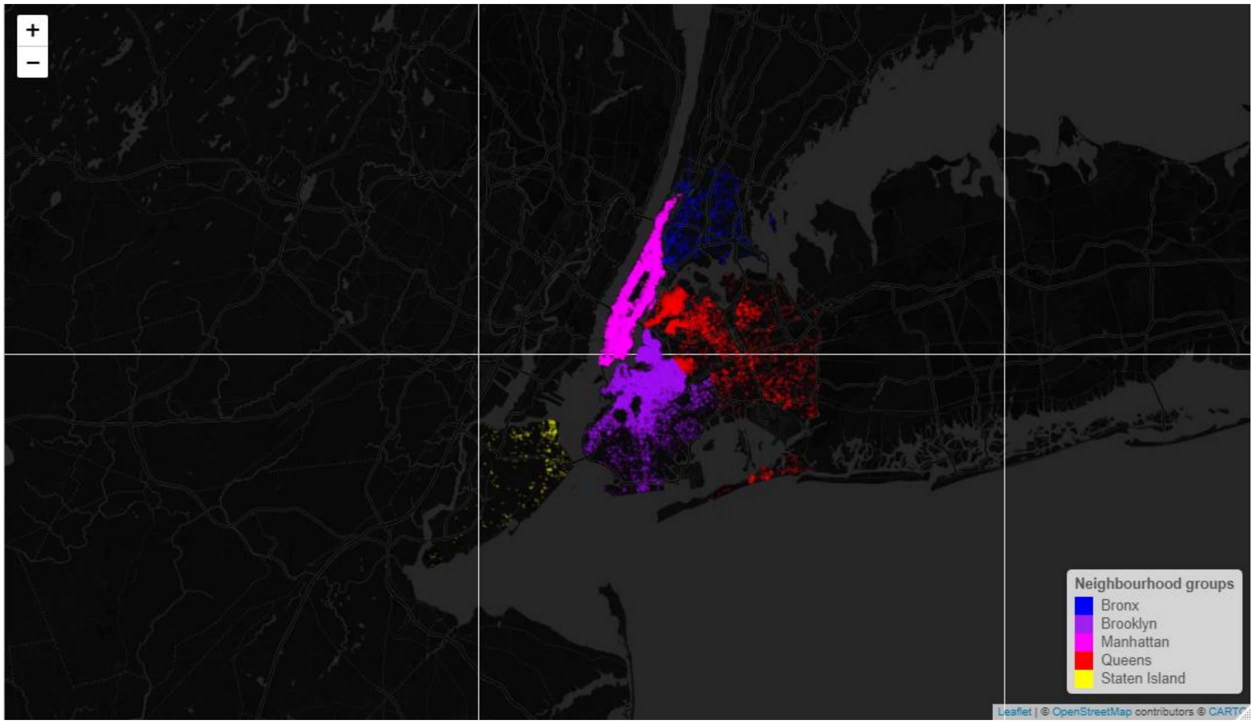


We see that the most expensive listings have the least number of reviews.

Let's look at a map of the NYC to show us where the most airbnbs are located

```
install.packages("ggthemes")
library(ggthemes)

install.packages("plotly")
library(plotly)

install.packages("GGally")
library(GGally)

install.packages("ggExtra")
library(ggExtra)


install.packages("RColorBrewer")
library(RColorBrewer)

install.packages("leaflet")
library(leaflet)

pal <- colorFactor(palette = c("blue", "purple", "magenta", "red", "yellow"),
domain = airbnb_data$neighbourhood_group)

leaflet(data = airbnb_data) %>%
addProviderTiles(providers$CartoDB.DarkMatterNoLabels) %>%
addCircleMarkers(~longitude, ~latitude, color = ~pal(neighbourhood_group),
weight = 1, radius=1, fillOpacity = 0.1, opacity = 0.1,

label = paste("Name:", airbnb_data$name)) %>%
```

```
addLegend("bottomright", pal = pal, values = ~neighbourhood_group,
          title = "Neighbourhood groups",
          opacity = 1
)
```



It is evident by exploring our data that Manhattan has the highest number of listings while it's the smallest neighbourhood group by area. It also has the most expensive listings which are mostly of the type "Entire Home or Apartment."

We can also conclude that the type "Entire Home or Apartment" is the most expensive type of listing in NYC.

<u>Inference</u>

From the previous exploratory discussion, there seems to be a price difference when it comes to room types. However, from the first sight alone is not enough to draw such conclusion. We can set up research hypothesis to test this relationship statistically.

Testing for Relationship between price and Listing Type

<u>Null Hypothesis: There is no statistical difference between price for Entire home/apt and shared room</u>

<u>Alt Hypothesis: There is statistical difference between price for Entire home/apt and shared room</u>

Firstly, we subset the data into two categories, Entire home/apt and shared room. Then we can use function inference to see what the mean prices of Airbnb and 95% confidence intervals for these subsets are. From the below data , we can see that not only two subsets have a significantly different mean, but they also have intervals at 90% totally not overlap. It's safe to say that we can reject the null hypothesis and come to conclusion that prices are different based on room type.

```
population <- read.csv("AB_NYC_2019.csv")

Entirehome <- subset(population,room_type =="Entire home/apt")

Sharedroom <- subset(population,room_type =="Shared room")

inference(Entirehome$price, est = "mean", type = "ci", method =
"theoretical")

 inference(Sharedroom$price, est = "mean", type = "ci", method =
"theoretical")
```

Entired home/apt

```
Summary statistics: mean = 211.7942 ;  sd = 284.0416 ;  n = 25409

Standard error = 1.7819

95 % Confidence interval = ( 208.3017 , 215.2867 )
```

Shared room

```
Summary statistics: mean = 70.1276 ;  sd = 101.7253 ;  n = 1160

Standard error = 2.9868

95 % Confidence interval = ( 64.2737 , 75.9815 )
```

Testing for Relationship between price and Neighbourhood group

We can also explore other variables within the data set to see whether we have another factor that can explain the prices. We can test out whether the location has impacts on pricing, specifically the neighborhood group. We can do a quick inference test whether means of prices across the neighborhoods are the same.

<u>Null: Price are indifferent statistically across neighborhoods.</u>

<u>Alt: There is a difference in prices for different neighborhoods.</u>

We can see that in the below summary that not only their mean prices vary significantly but the test result provides an astonishing small P value, which leads us to reject the null hypothesis.

```
inference(population$price, population$neighbourhood_group, est = "mean",
type = "ht", method = "theoretical", alternative = "greater")
```

Summary statistics:

```
n_Bronx = 1091, mean_Bronx = 87.4968, sd_Bronx = 106.7093

n_Brooklyn = 20104, mean_Brooklyn = 124.3832, sd_Brooklyn = 186.8735

n_Manhattan = 21661, mean_Manhattan = 196.8758, sd_Manhattan = 291.3832

n_Queens = 5666, mean_Queens = 99.5176, sd_Queens = 167.1022

n_Staten Island = 373, mean_Staten Island = 114.8123, sd_Staten Island =
277.6204
```

H_0: All means are equal.

H_A: At least one mean is different.

Analysis of Variance Table

Response: y

```
             Df     Sum Sq  Mean Sq F value    Pr(>F)

x             4   79590956 19897739  354.99 < 2.2e-16

Residuals 48890 2740322834    56051
```

Pairwise tests: t tests with pooled SD

```
               Bronx Brooklyn Manhattan Queens

Brooklyn      0.0000       NA        NA     NA

Manhattan     0.0000   0.0000        NA     NA

Queens        0.1246   0.0000         0     NA

Staten Island 0.0544   0.4392         0 0.2268
```

Modeling

Preceding displaying, we will divide the information into Training set and Testing set with the goal that we can utilize the testing set to approve our model. As it is a decent practice, we are parting the dataset into parts in the proportion of 70:30. Training set will be 70% percent of the first information. We will utilize the test dataset in the future for testing and expectation purposes. Articles with value equivalent to 0 will be omitted since cost can't be 0 (broken records). To eliminate the anomalies, we are separating the airbnb information by eliminating the outrageous upsides of cost from the two sides (10% from both the end). They would make prescient models fundamentally more fragile.

```
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

library(ggthemes)

## Warning: package 'ggthemes' was built under R version 4.0.5

library(GGally)

## Warning: package 'GGally' was built under R version 4.0.5

library(caret)

## Warning: package 'caret' was built under R version 4.0.5

library(glmnet)

## Warning: package 'glmnet' was built under R version 4.0.5

library(corrplot)
library(leaflet)

## Warning: package 'leaflet' was built under R version 4.0.5

library(kableExtra)

## Warning: package 'kableExtra' was built under R version 4.0.5

library(RColorBrewer)
library(plotly)
```

```
## Warning: package 'plotly' was built under R version 4.0.5
```

```
set.seed(252)
air_bnb <- read.csv("AB_NYC_2019.csv", encoding="UTF-8", stringsAsFactors = F
, na.strings = c(""))

air_bnb <- air_bnb %>% mutate(id = row_number())

air_bnb_train <- air_bnb %>% sample_frac(.7) %>% filter(price > 0)

air_bnb_test  <- anti_join(air_bnb, air_bnb_train, by = 'id') %>% filter(pric
e > 0)
nrow(air_bnb_train) + nrow(air_bnb_test) == nrow(air_bnb %>% filter(price > 0
))
```

```
## [1] TRUE
```

Perceptions:

The subsequent training dataset has 34,221 perceptions and testing dataset has 14.663 perceptions.

Second look just in case affirms that that in the wake of eliminating the perceptions with value 0 and parting the dataset, the number of perceptions in test and train dataset is equivalent to the all-out number of perceptions in the first dataset.

We attempt to foresee the cost of the airbnbs utilizing the excess covariates:

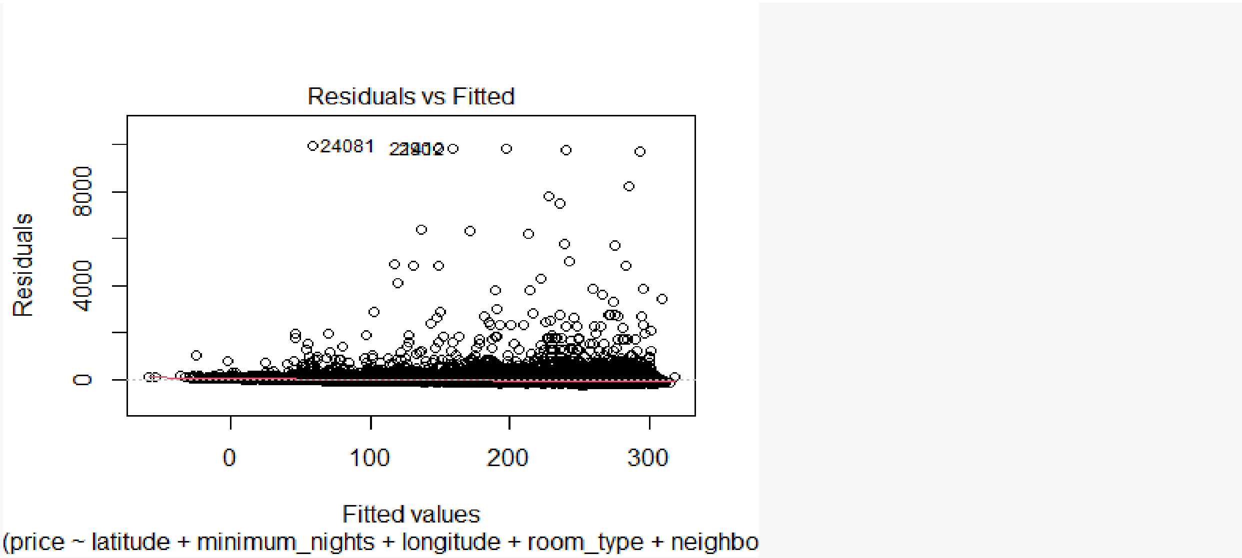
latitude longitude neighbourhood_group room_type number_of_reviews minimum_nights calculated_host_listings_count reviews_per_month availability_365

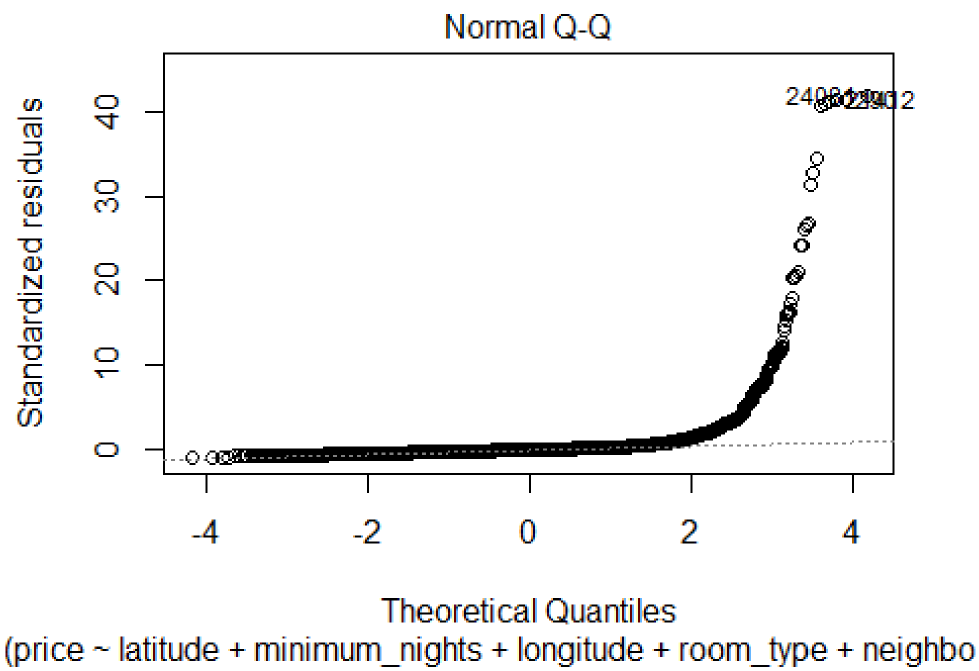
Plot of the First Linear Regression Model:

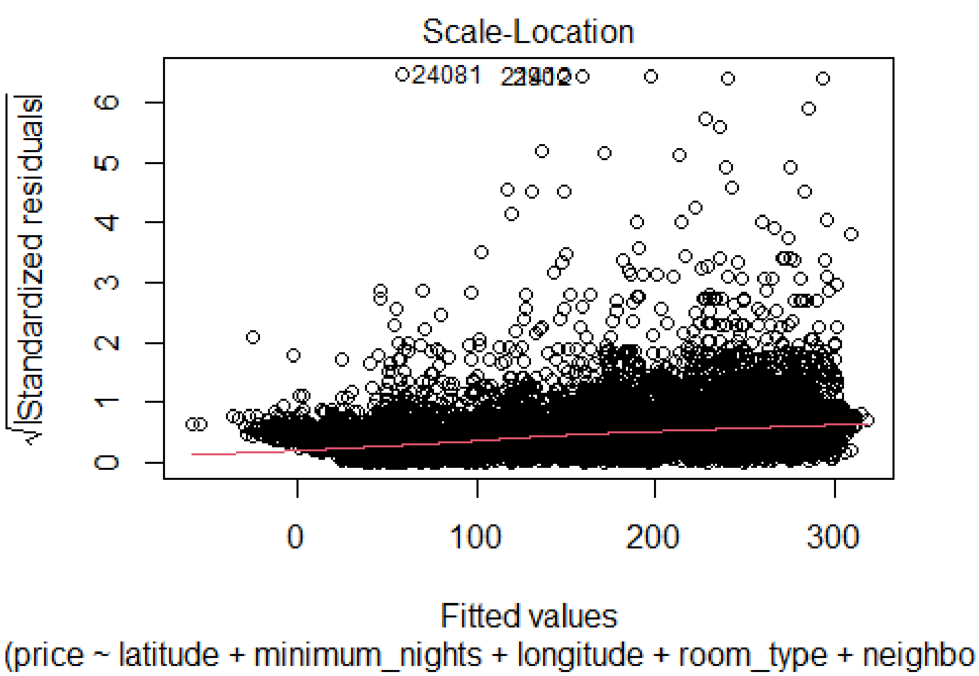
```
fst_model <- lm(price ~ latitude + minimum_nights+ longitude + room_type +  n
eighbourhood_group + availability_365 , data = air_bnb_train, )

summary(fst_model)
```

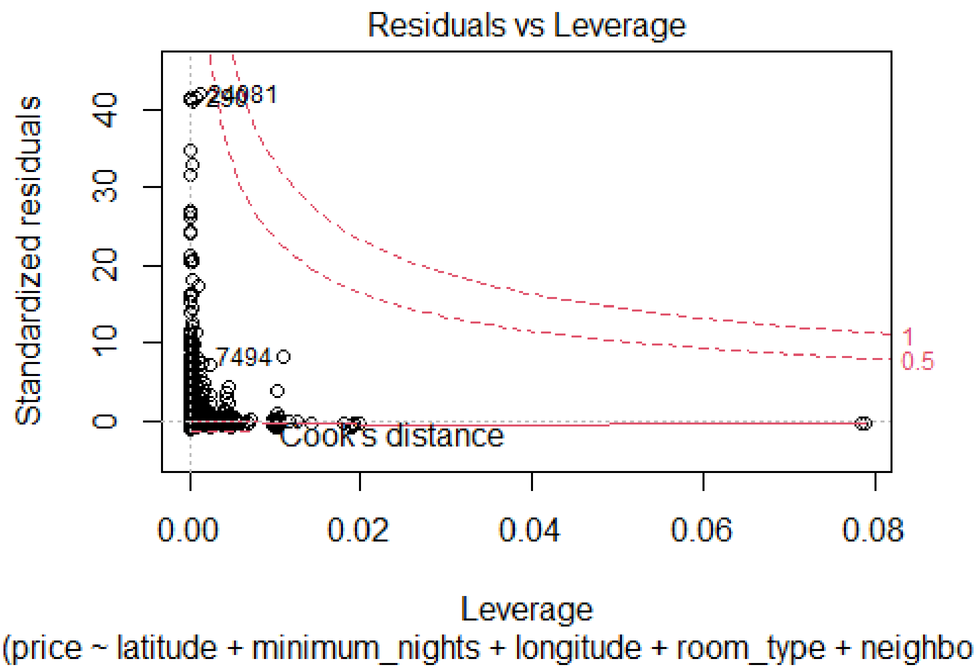
```
##
## Call:
## lm(formula = price ~ latitude + minimum_nights + longitude +
##     room_type + neighbourhood_group + availability_365, data = air_bnb_tra
in)
##
## Residuals:
##    Min      1Q  Median      3Q     Max
## -244.1  -62.4   -25.4   13.7 9941.1
##
```

```
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)

## (Intercept)                     -2.855e+04  3.962e+03  -7.204 5.95e-13 **
*
## latitude                        -2.089e+02  3.864e+01  -5.406 6.50e-08 **
*
## minimum_nights                   4.005e-02  6.705e-02   0.597  0.55030

## longitude                       -5.036e+02  4.445e+01 -11.329  < 2e-16 **
*
## room_typePrivate room           -1.048e+02  2.682e+00 -39.076  < 2e-16 **
*
## room_typeShared room            -1.349e+02  8.460e+00 -15.946  < 2e-16 **
*
## neighbourhood_groupBrooklyn     -3.330e+01  1.097e+01  -3.035  0.00241 **

## neighbourhood_groupManhattan     2.778e+01  9.994e+00   2.779  0.00545 **

## neighbourhood_groupQueens       -4.049e+00  1.058e+01  -0.383  0.70196

## neighbourhood_groupStaten Island -1.628e+02  2.117e+01  -7.693 1.48e-14 **
*
## availability_365                 1.525e-01  1.002e-02  15.221  < 2e-16 **
*
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 237.3 on 34208 degrees of freedom
## Multiple R-squared:  0.0877, Adjusted R-squared:  0.08743
## F-statistic: 328.8 on 10 and 34208 DF,  p-value: < 2.2e-16

plot(fst_model)
```


Residuals vs Fitted
Residuals
8000
4000
0
24081
0
100
200
300
Fitted values
(price ~ latitude + minimum_nights + longitude + room_type + neighbo


Normal Q-Q
Standardized residuals
Theoretical Quantiles
(price ~ latitude + minimum_nights + longitude + room_type + neighbo


Scale-Location
√|Standardized residuals|
24081
Fitted values
(price ~ latitude + minimum_nights + longitude + room_type + neighbo

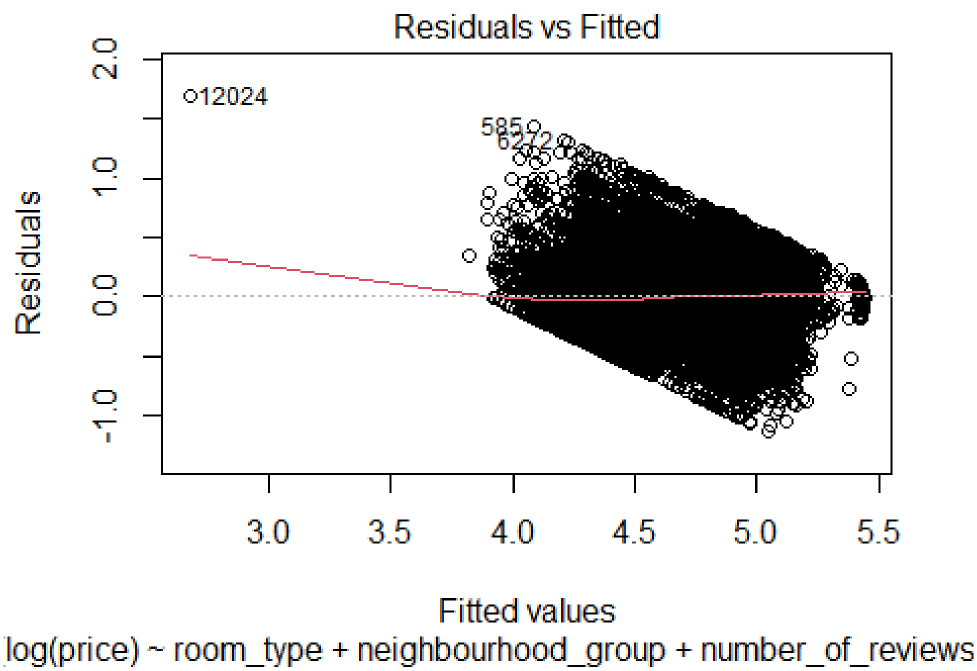


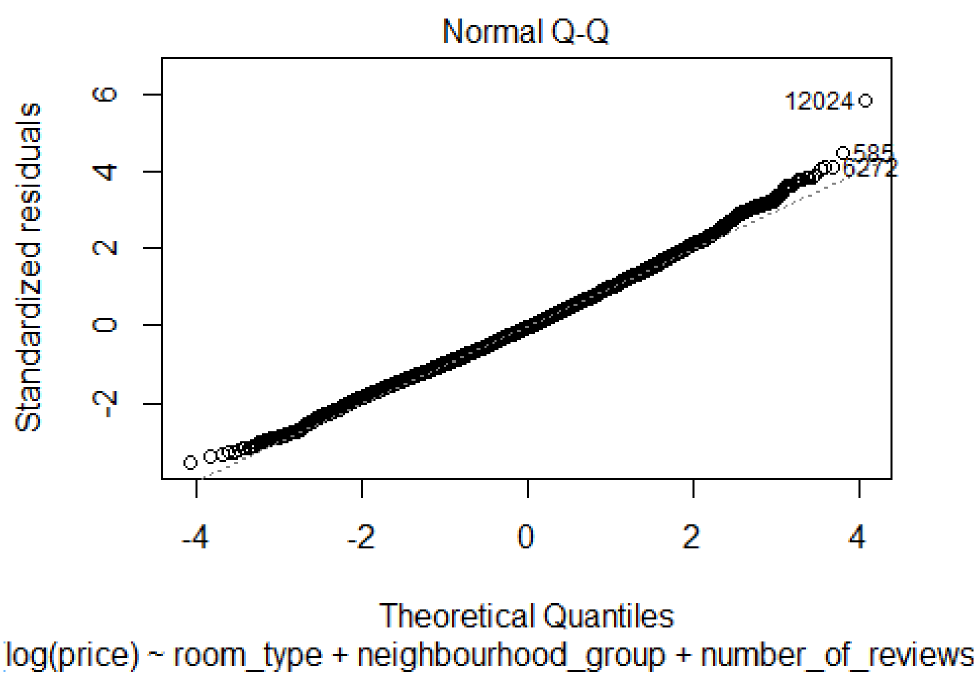
Plot of the Second Linear Regression Model:

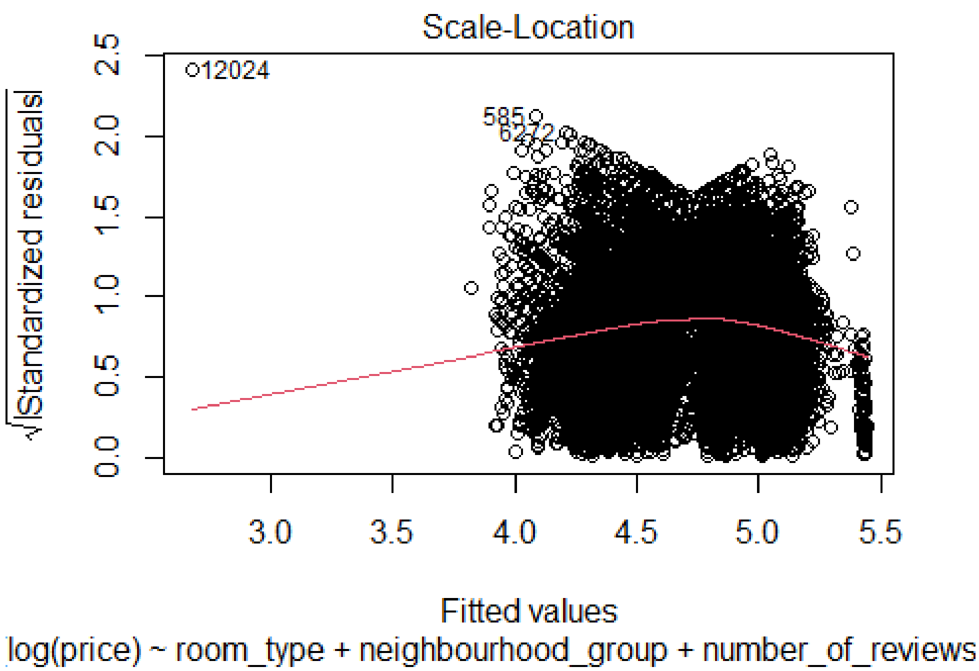
```
learn <- air_bnb_train %>% filter(price < quantile(air_bnb_train$price, 0.9)
& price > quantile(air_bnb_train$price, 0.1)) %>% tidyr::drop_na()

scnd_model <- lm(log(price) ~ room_type + neighbourhood_group + number_of_rev
iews +latitude + longitude + calculated_host_listings_count + availability_36
5 + reviews_per_month + minimum_nights, data = learn)

summary(scnd_model)

##
## Call:
## lm(formula = log(price) ~ room_type + neighbourhood_group + number_of_revi
ews +
##     latitude + longitude + calculated_host_listings_count + availability_3
65 +
##     reviews_per_month + minimum_nights, data = learn)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.13208 -0.22460 -0.01546  0.20825  1.69326
##
## Coefficients:
```

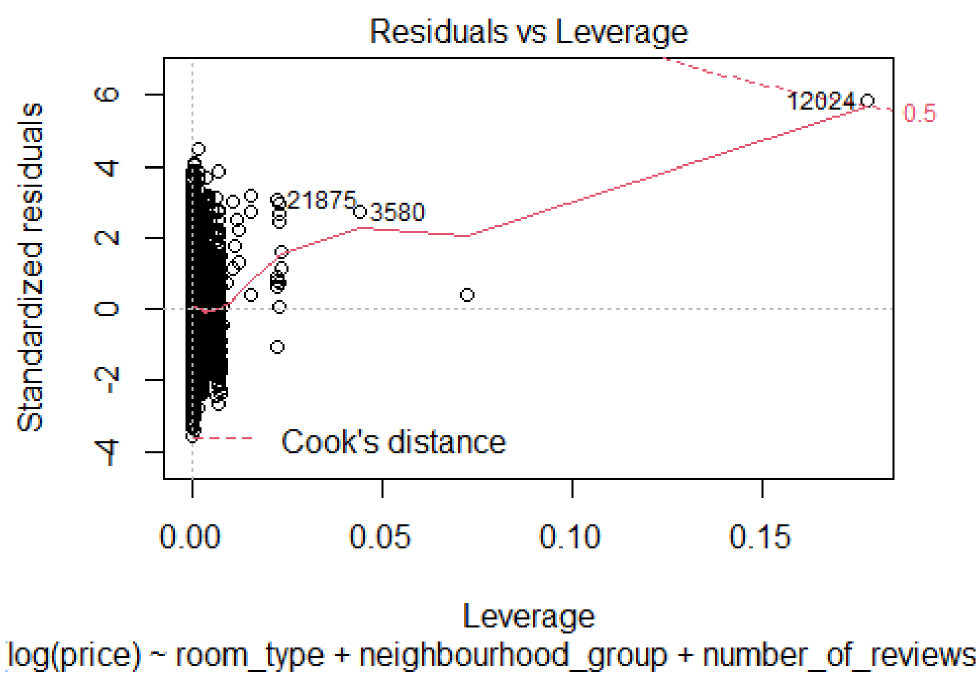
```
##                                    Estimate Std. Error  t value Pr(>|t|)

## (Intercept)                       -1.227e+02  6.686e+00  -18.347  < 2e-16 *
**
## room_typePrivate room             -5.429e-01  4.435e-03 -122.415  < 2e-16 *
**
## room_typeShared room              -6.533e-01  2.038e-02  -32.060  < 2e-16 *
**
## neighbourhood_groupBrooklyn       -2.925e-02  1.960e-02   -1.492   0.1357

## neighbourhood_groupManhattan       1.625e-01  1.806e-02    8.999  < 2e-16 *
**
## neighbourhood_groupQueens          4.678e-02  1.900e-02    2.462   0.0138 *

## neighbourhood_groupStaten Island -5.847e-01  3.686e-02  -15.862  < 2e-16 *
**
## number_of_reviews                 -6.683e-05  5.393e-05   -1.239   0.2153

## latitude                          -5.249e-01  6.509e-02   -8.064 7.75e-16 *
**
## longitude                         -2.013e+00  7.518e-02  -26.780  < 2e-16 *
**
## calculated_host_listings_count     5.605e-04  8.470e-05    6.617 3.74e-11 *
**
## availability_365                   3.211e-04  1.814e-05   17.704  < 2e-16 *
**
## reviews_per_month                 -2.378e-03  1.566e-03   -1.519   0.1288

## minimum_nights                    -1.690e-03  1.355e-04  -12.471  < 2e-16 *
**
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3195 on 22046 degrees of freedom
## Multiple R-squared:  0.4935, Adjusted R-squared:  0.4932
## F-statistic:  1653 on 13 and 22046 DF,  p-value: < 2.2e-16
```

```
plot(scnd_model)
```


Residuals vs Fitted
Residuals
Fitted values
log(price) ~ room_type + neighbourhood_group + number_of_reviews
12024
585
6272
2.0
1.0
0.0
-1.0
3.0
3.5
4.0
4.5
5.0
5.5


Normal Q-Q
Standardized residuals
Theoretical Quantiles
log(price) ~ room_type + neighbourhood_group + number_of_reviews
12024
585
6272
6
4
2
0
-2
-4
-2
0
2
4


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
12024
585
6272
3.0
3.5
4.0
4.5
5.0
5.5
Fitted values
log(price) ~ room_type + neighbourhood_group + number_of_reviews


Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
6
12024
0.5
21875
3580
Cook's distance
0.00
0.05
0.10
0.15
Leverage
log(price) ~ room_type + neighbourhood_group + number_of_reviews

Perceptions: Residuals versus fitted qualities shows that the specks are not equally disseminated around nothing and don't show a consistent difference around X. This implies that linearity and equivalent fluctuation suppositions are not satisfied.

QQ plot shows a 45-degree line implying that Normality presumptions are met.

Regression Formula for First Model:

Y= **`-28550 + (-208.9 * latitude) +(0.04005 * minimum_nights) + (-503.6 * longitude) +(-104.8 * room_typePrivate)+ (-134.9 * room_typeShared)+ (-33.30 * neighbourhood_groupBrooklyn)+ (27.78 * neighbourhood_groupManhattan) + (-4.049 * neighbourhood_groupQueens) + (-162.8 * neighbourhood_groupStaten Island) + (0.1525* availability_365)`**

The regression Formula is created from the Coefficients and Intercepts created from First Linear Regression Model

```
## Coefficients:
##                                     Estimate
## (Intercept)                       -2.855e+04
## latitude                          -2.089e+02
## minimum_nights                     4.005e-02
## longitude                         -5.036e+02
## room_typePrivate room             -1.048e+02
## room_typeShared room              -1.349e+02
## neighbourhood_groupBrooklyn       -3.330e+01
## neighbourhood_groupManhattan       2.778e+01
## neighbourhood_groupQueens         -4.049e+00
## neighbourhood_groupStaten Island -1.628e+02
## availability_365                   1.525e-01
```

Prediction:

We applied Multi- Linear Regression model on the data, room type Private and room type Shared is assigned numerical values respectively similarly for neighborhood Brooklyn, Manhattan, Queens, and Staten Island are assigned numerical values.

We Calculated the Price estimate manually based on the ID and compared with original price and found them to be very much different. Hence it proves that model does not fit. Hence it is not possible to predict the Price estimate accurately based on the above factors. Also, comparing the R Square model as R square is 8 %, which is a very low match.

ID		Price_Estimate		Original_Price
2539		336.690579		149
2595		2819.53549		225
3647		2816.430121		150
3831		2817.301644		89
5022		2811.216713		80
5099		2815.96387		200
5121		2814.291647		60

Conclusion:

In conclusion, we chose the multiple linear models to fit the above dataset, 10 variables were used in the model. We were able to find that the model would not be able to predict the price properly as Price estimate and original price does not match.

Thus, we may not be able to predict the price precisely based on these factors.