Exploring Airbnb in New York City before COVID-19

Hyunjoon Rhee

Contents

Introduction **1**

What is this topic? . . . 1

Where did it come from? . . . 2

What are the variables? . . . 2

Why is it interesting to us? . . . 2

Why are we creating a model for this data? What is the goal of this model? . . . 2

Methods **3**

Data Cleaning . . . 3

Considering Variables . . . 4

Correlation matrix . . . 4

Model Comparison . . . 5

Result **18**

testing . . . 18

Discussion **23**

Appendix **26**

Introduction

What is this topic?

This dataset include information of Airbnb in New York city. This dataset can be divided mainly into five groups.

It includes host information, including the host id and its name. It has shown the location of the accommodation based on cities and specific areas with latitude and longitude.

We can predict customers' preferences based on space type, price, and amount of days that they are willing to stay. With the number of reviews and recent review date, we can also predict their experience at the accommodations.

Lastly, the availability of accommodation will be listed with the amount of listing per host and the number of days when it is available for booking.

Where did it come from?

To make accurate predictions, we included observations from many reliable sources with credible values. This dataset is found on the website, Kaggle. Since this dataset includes variables from all kinds of perspectives, it indicates how active Airbnb market is in New York city. The original source of this public dataset can be found on the Airbnb Website.

What are the variables?

Variable	Type	Description
id	Integer	ID
name	Character	Name of the listing
host_id	Integer	Host ID
host_name	Character	Name of the host
neighbourhood_group	Character	Location
neighbourhood	Character	Area
latitude	Double	Latitude coordinates
longitude	Double	Longitude coordinates
room_type	Character	Listing space type
price	Integer	Price in dollars
minimum_nights	Integer	Amount of nights minimum
number_of_reviews	Integer	Number of reviews
last_review	Character	Latest review
reviews_per_month	Double	Number of reviews per month
calculated_host_listings_count	Integer	Amount of listing per host
availability_365	Integer	Number of days when listing is available for booking

Why is it interesting to us?

The year 2020 has been different from any other years due to the impact of COVID-19 pandemic. Due to the pandemic, there has been a huge crisis on travel, hotel and airbnb industry. This crisis caused fear to majority of the people. People are scared of traveling or even stepping one step outside of house. Moreover, the government regulation of traveling, by closing majority of the stores and preventing international travel significantly affected the usage of airbnb or hotels.

Many businesses in different locations got negative impact, including the Airbnb - an American vacation rental business. As other cities, one of the most populated cities, New York, wasn't exception. New York has became a much more quiet city than before.

As this situation is still going on and unpredictable, our interest came from our desire to travel around. Plus, just like us, there are lots of people who are seeking freedom of traveling due to such special circumstances. Wishing the situation gets better, our team has broke down the data to analyze the relationship between the price and other factors.

Why are we creating a model for this data? What is the goal of this model?

Since the dataset has more than 2500 observations, it is hard to read the result of data without a model. The regression model allow us to use the relationships between price and other variables to make predictions. Among the models, we will choose the best look model to show which variable has the most interactive with the response, price.

Methods

Data Cleaning

We have started with takeoff NA from the dataset.

```
airbnbdata = read.csv("AB_NYC_2019.csv")
airbnbdata[airbnbdata==0] <- NA
airbnbdata <- na.omit(airbnbdata)
head(airbnbdata)
```

```
##      id                                      name host_id   host_name
## 1  2539        Clean & quiet apt home by the park    2787        John
## 2  2595                     Skylit Midtown Castle    2845    Jennifer
## 4  3831           Cozy Entire Floor of Brownstone    4869 LisaRoxanne
## 6  5099 Large Cozy 1 BR Apartment In Midtown East    7322       Chris
## 8  5178          Large Furnished Room Near B'way     8967    Shunichi
## 10 5238        Cute & Cozy Lower East Side 1 bdrm    7549         Ben
##    neighbourhood_group  neighbourhood latitude longitude       room_type price
## 1             Brooklyn     Kensington 40.64749 -73.97237    Private room   149
## 2            Manhattan        Midtown 40.75362 -73.98377 Entire home/apt   225
## 4             Brooklyn   Clinton Hill 40.68514 -73.95976 Entire home/apt    89
## 6            Manhattan    Murray Hill 40.74767 -73.97500 Entire home/apt   200
## 8            Manhattan Hell's Kitchen 40.76489 -73.98493    Private room    79
## 10           Manhattan      Chinatown 40.71344 -73.99037 Entire home/apt   150
##    minimum_nights number_of_reviews last_review reviews_per_month
## 1               1                 9  2018-10-19              0.21
## 2               1                45  2019-05-21              0.38
## 4               1               270  2019-07-05              4.64
## 6               3                74  2019-06-22              0.59
## 8               2               430  2019-06-24              3.47
## 10              1               160  2019-06-09              1.33
##    calculated_host_listings_count availability_365
## 1                               6              365
## 2                               2              355
## 4                               1              194
## 6                               1              129
## 8                               1              220
## 10                              4              188
```

Split data into Train & Test set

We have split the data into the training set and testing set. We used a training set to find an optimal model. We have predicted price by using the optimal model we have chosen from the training set. We will compare the actual price and predicted price.

```
bnb_trn_idx  = sample(nrow(airbnbdata), size = trunc(0.80 * nrow(airbnbdata)))
bnb_trn_data = airbnbdata[bnb_trn_idx, ]
bnb_tst_data = airbnbdata[-bnb_trn_idx, ]
bnb_trn_data <- na.omit(bnb_trn_data)
bnb_tst_data <- na.omit(bnb_tst_data)
```

Remove Outlier

To get accurate statistical results, we have removed the outlier. We used the formula $Q_1 - 1.5 * IQR$ for the lower outlier and $Q_3 + 1.5 * IQR$ for the upper outlier.

```
iqr = IQR(bnb_trn_data$price,na.rm = TRUE)
quart = quantile(bnb_trn_data$price, c(0.25, 0.5, 0.75), type = 1)
upperoutlier = quart[3] + 1.5*iqr[1]
loweroutlier = quart[1]-1.5*iqr[1]
c(upperoutlier,loweroutlier)
```

```
## 75% 25%
## 335 -89
```

```
rowtoremove1 = c(which(bnb_trn_data$price > upperoutlier))
rowtoremove2 = c(which(bnb_trn_data$price < loweroutlier))
bnb_trn_data_new <- bnb_trn_data[-rowtoremove1,]
```

Considering Variables

To make a decision for which variable to consider, we have plotted the correlation matrix. And, we have chosen the highest five correlations which are close to 1 or -1 (absolute value close to 1).

```
bnb_trn_data_new$neighbourhood_group = as.factor(bnb_trn_data_new$neighbourhood_group)
bnb_trn_data_new$neighbourhood_group = as.numeric(bnb_trn_data_new$neighbourhood_group)
bnb_trn_data_new$room_type = as.factor(bnb_trn_data_new$room_type)
bnb_trn_data_new$room_type = as.numeric(bnb_trn_data_new$room_type)
```

```
bnb_trn_data_new <- subset(bnb_trn_data_new, select = -c(name,host_name,neighbourhood,last_review))
str(bnb_trn_data_new)
```

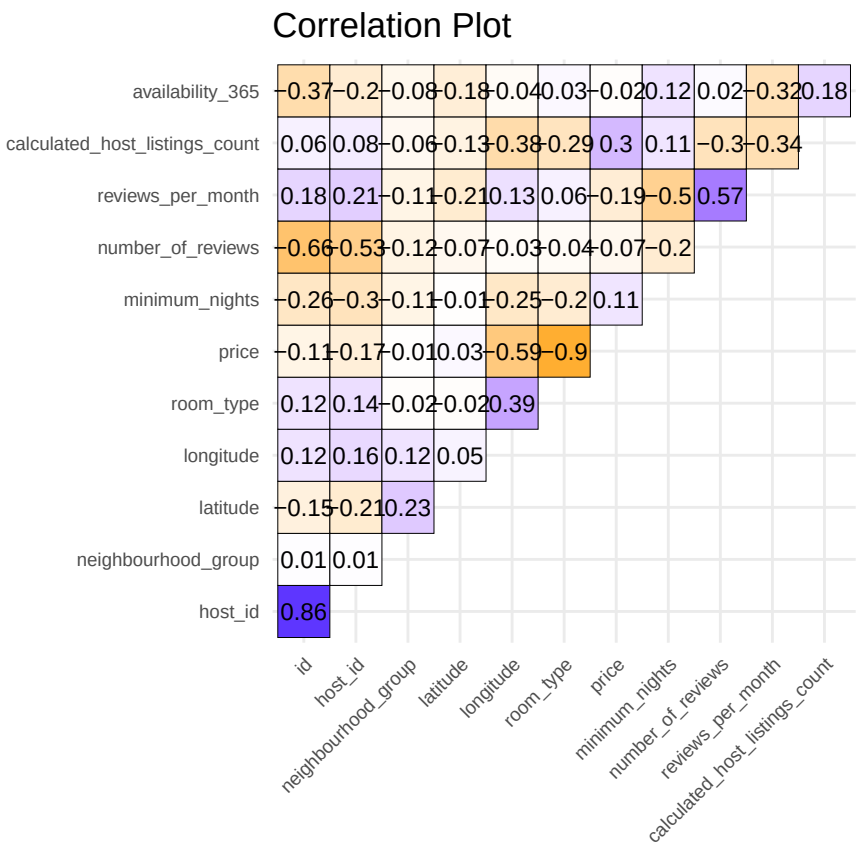
```
## 'data.frame':    19662 obs. of  12 variables:
##  $ id                            : int  2087524 32475687 9435931 34493353 15087285 1326514 19767452 3
##  $ host_id                       : int  10656683 230205171 5037211 28371926 77778146 7240751 2830766
##  $ neighbourhood_group           : num  2 1 2 2 2 2 2 2 3 1 ...
##  $ latitude                      : num  40.7 40.8 40.7 40.7 40.6 ...
##  $ longitude                     : num  -74 -73.8 -73.9 -73.9 -73.9 ...
##  $ room_type                     : num  1 1 1 2 2 2 1 1 1 2 ...
##  $ price                         : int  99 100 275 60 85 65 170 165 129 23 ...
##  $ minimum_nights                : int  4 2 5 2 1 3 2 8 1 2 ...
##  $ number_of_reviews             : int  74 22 77 9 2 72 12 23 178 18 ...
##  $ reviews_per_month             : num  1.93 4.93 1.89 4.82 0.09 1.22 0.53 0.38 3.43 1.25 ...
##  $ calculated_host_listings_count: int  1 1 2 1 8 1 1 1 1 1 ...
##  $ availability_365              : int  261 3 6 87 189 157 3 297 272 36 ...
```

Correlation matrix

```
library(ggcorrplot)
```

```
## Loading required package: ggplot2
```

```
correlation <- cor(bnb_trn_data_new)
correlation2 <- round(cor(correlation,use="complete.obs"), 2)
options(repr.plot.width=12, repr.plot.height=12)
ggcorrplot(correlation2, lab = TRUE, colors = c("orange", "white", "blue"), show.legend = F, outline.co
```



According to correlation matrix, correlation between price and room_type, longitude, calculated_host_listings_count, reviews_per_month, and minimum_nights are -0.9, -0.59, 0.31, -0.2, 0.12 respectively.

Model Comparison

3 Variables

```
model1 = lm(price ~ room_type, data = bnb_trn_data_new)
model2 = lm(price ~ room_type + calculated_host_listings_count, data = bnb_trn_data_new)
anova(model1, model2)
```

```
## Analysis of Variance Table
##
## Model 1: price ~ room_type
## Model 2: price ~ room_type + calculated_host_listings_count
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
```

```
## 1  19660 56770924
## 2  19659 56068504  1    702420 246.29 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- H_0: β_2 (calculated_host_listings_count) = 0
- H_1: β_2 (calculated_host_listings_count) != 0
- Test statistic: 261.63
- p-value: 2.2e-16
- Decision on $\alpha = 0.10$: Reject H0, select model2
- We started with simple regresson model that has room_type as a predictor of price and did ANOVA testing to compare this model with additive model that has room_type and calculated_host_listings_count as predictors. As a result, we reject H_0 and select model2 among those two models.

```
model3 = lm(price ~ room_type + longitude, data = bnb_trn_data_new)
anova(model1, model3)
```

```
## Analysis of Variance Table
##
## Model 1: price ~ room_type
## Model 2: price ~ room_type + longitude
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1  19660 56770924
## 2  19659 52656378  1   4114545 1536.1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- H_0: β_2 (longtitude) = 0
- H_1: β_2 (longitude) != 0
- Test statistic: 1543.8
- p-value: 2.2e-16
- Decision on $\alpha = 0.1$: Reject H0, select model3
- Similarily, we compared the simple regression model (model1) with the additive model that has room_type and longitude as predictors. As a result, again we reject H_0 and select model3 among model1 and model3.

```
modelvar3 = lm(price ~ room_type + calculated_host_listings_count+ longitude, data = bnb_trn_data_new)
anova(model2, modelvar3)
```

```
## Analysis of Variance Table
##
## Model 1: price ~ room_type + calculated_host_listings_count
## Model 2: price ~ room_type + calculated_host_listings_count + longitude
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1  19659 56068504
## 2  19658 52243606  1   3824898 1439.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- H_0: β_3 (longitude) = 0
- H_1: β_3 (longitude) != 0

- Test statistic: 1443.7
- p-value: 2.2e-16
- Decision on $\alpha = 0.1$: Reject H0, select modelvar3
- Lastly, we compared the model that has predictors of room_type and calculated_host_listings_count and the model with 3 predictors, room_type, calculated_host_listings_count, AND longitude. After performing ANOVA, we reject the null hypothesis and chose the model with three predictors.

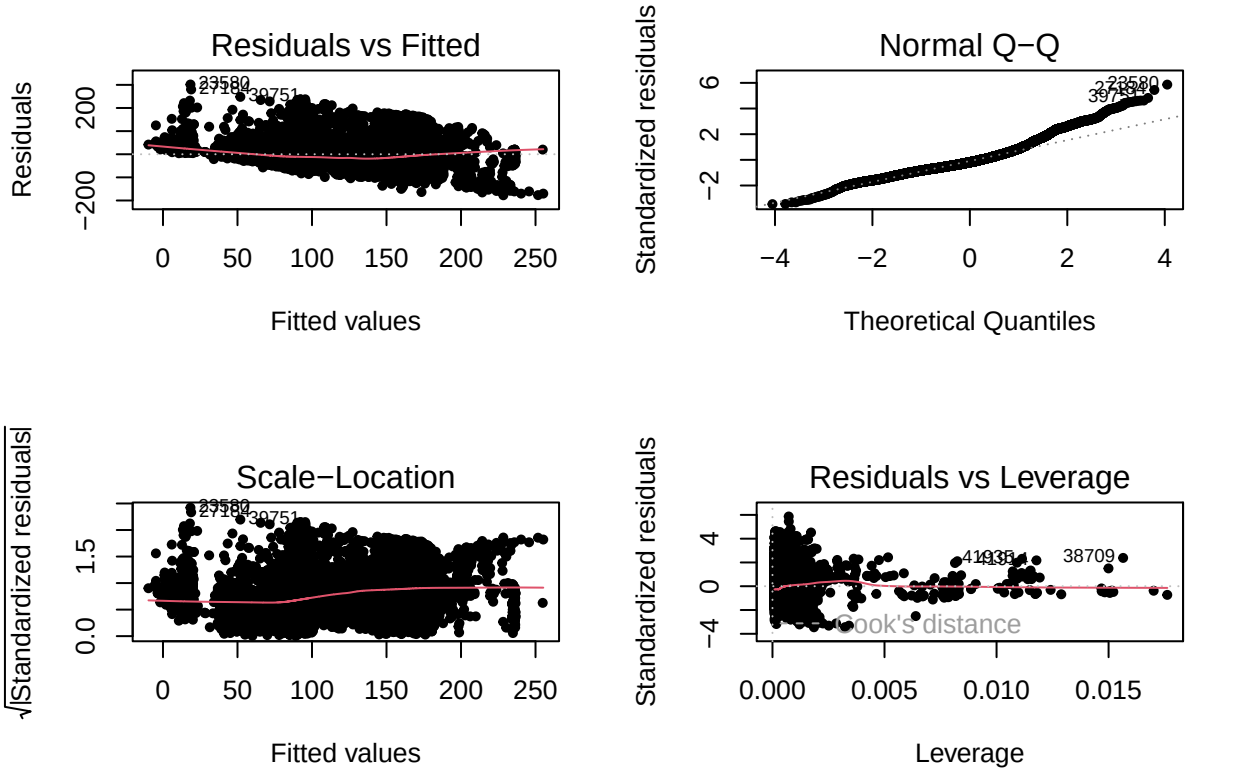
AIC and Log Transformation

```
var3original = lm(price ~ 1, data = bnb_trn_data_new)
var3_aic = step(var3original, scope = price ~ room_type * calculated_host_listings_count * longitude, di
```

```
## Start:  AIC=166057.4
## price ~ 1
##
##                                   Df Sum of Sq      RSS    AIC
## + room_type                        1  34737653 56770924 156673
## + longitude                        1   8456209 83052368 164153
## + calculated_host_listings_count   1   1855424 89653153 165657
## <none>                                         91508577 166057
##
## Step:  AIC=156672.6
## price ~ room_type
##
##                                   Df Sum of Sq      RSS    AIC
## + longitude                        1   4114545 52656378 155195
## + calculated_host_listings_count   1    702420 56068504 156430
## <none>                                         56770924 156673
## - room_type                        1  34737653 91508577 166057
##
## Step:  AIC=155195.3
## price ~ room_type + longitude
##
##                                   Df Sum of Sq      RSS    AIC
## + calculated_host_listings_count   1    412772 52243606 155043
## + room_type:longitude              1    244263 52412115 155106
## <none>                                         52656378 155195
## - longitude                        1   4114545 56770924 156673
## - room_type                        1  30395990 83052368 164153
##
## Step:  AIC=155042.6
## price ~ room_type + longitude + calculated_host_listings_count
##
##                                             Df Sum of Sq      RSS    AIC
## + room_type:longitude                        1    186507 52057099 154974
## + calculated_host_listings_count:longitude   1     35568 52208039 155031
## + room_type:calculated_host_listings_count   1      6478 52237128 155042
## <none>                                                   52243606 155043
## - calculated_host_listings_count             1    412772 52656378 155195
## - longitude                                  1   3824898 56068504 156430
## - room_type                                  1  29709961 81953567 163893
##
```

```
## Step:  AIC=154974.3
## price ~ room_type + longitude + calculated_host_listings_count +
##     room_type:longitude
##
##                                            Df Sum of Sq      RSS    AIC
## + calculated_host_listings_count:longitude  1     44474 52012626 154959
## <none>                                                    52057099 154974
## + room_type:calculated_host_listings_count  1      4715 52052384 154974
## - room_type:longitude                       1    186507 52243606 155043
## - calculated_host_listings_count            1    355016 52412115 155106
##
## Step:  AIC=154959.5
## price ~ room_type + longitude + calculated_host_listings_count +
##     room_type:longitude + longitude:calculated_host_listings_count
##
##                                            Df Sum of Sq      RSS    AIC
## <none>                                                    52012626 154959
## + room_type:calculated_host_listings_count  1       145 52012480 154961
## - longitude:calculated_host_listings_count  1     44474 52057099 154974
## - room_type:longitude                       1    195413 52208039 155031
```

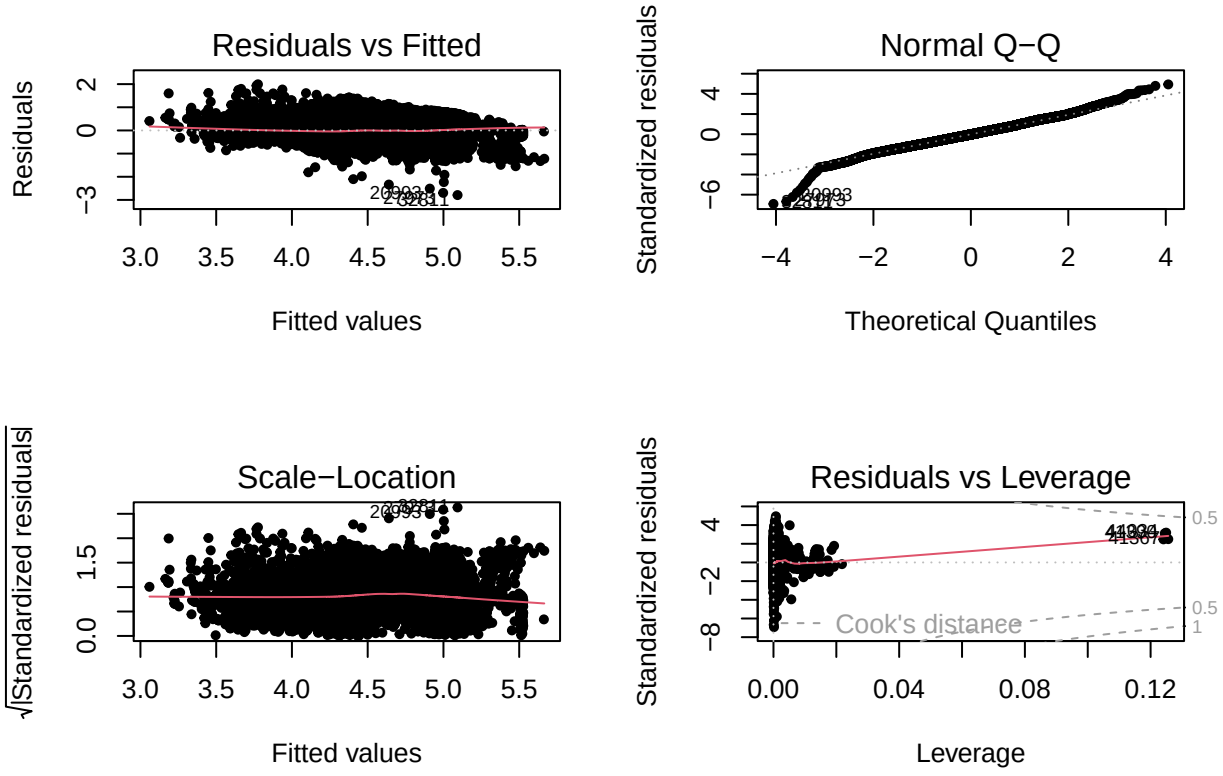
```
par(mfrow=c(2,2))
plot(var3_aic, pch = 20)
```



- In order to check if interaction between three variables (room_type, calculated_host_listings_count,

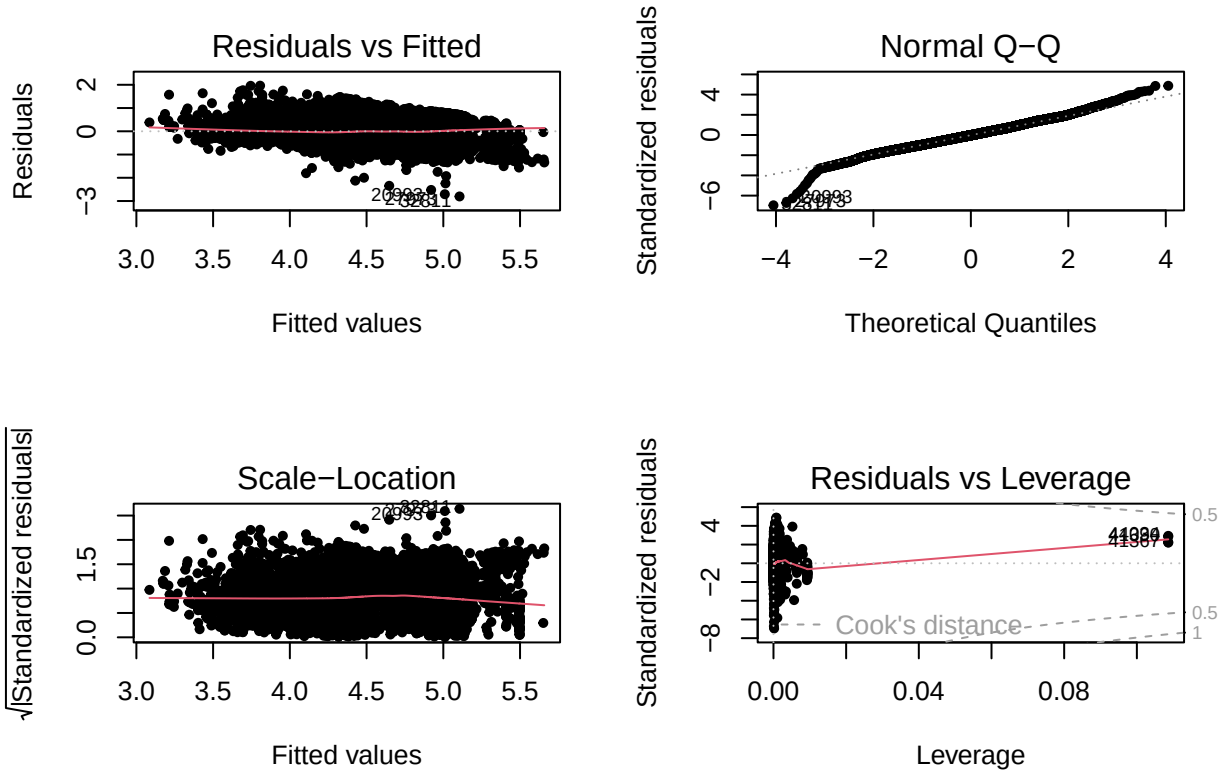
and longitude) makes a better model for price prediction, we peformed AIC search with both direction and scoped it to three-way interaction.

```
var3_aic_log = lm(log(price) ~ room_type + longitude + calculated_host_listings_count +
    room_type:longitude + room_type:calculated_host_listings_count +
    longitude:calculated_host_listings_count, data = bnb_trn_data_new)
par(mfrow=c(2,2))
plot(var3_aic_log,pch=20)
```



- Then we performed a log transformation on the response variable (price) to make the data more linear and normal. However, we observed that it would be better to consider another model by performing another log transformation on one of the predictor variable: calculated_host_listings_count.

```
var3_aic_log2 = lm(log(price) ~ room_type + longitude + log(calculated_host_listings_count) +
    room_type:longitude + room_type:calculated_host_listings_count +
    longitude:calculated_host_listings_count, data = bnb_trn_data_new)
par(mfrow=c(2,2))
plot(var3_aic_log2, pch = 20)
```



- We have concluded that transformation of log(calculated_host_listings_count) does not bring much difference to the original data.

```
var3predict = data.frame(room_type = 2, calculated_host_listings_count = 1, longitude = -73.95725)
exp(predict(var3_aic_log,newdata=var3predict))
```

```
##        1
## 77.72749
```

- We randomly selected one data point from the dataset, the one with room_type = 2(Private room), calculated_host_listings_count = 1, and longitude = -73.95725, and used the first log transformation model to predict the price. As a result, the prediction was a bit far from the actual price ($60).

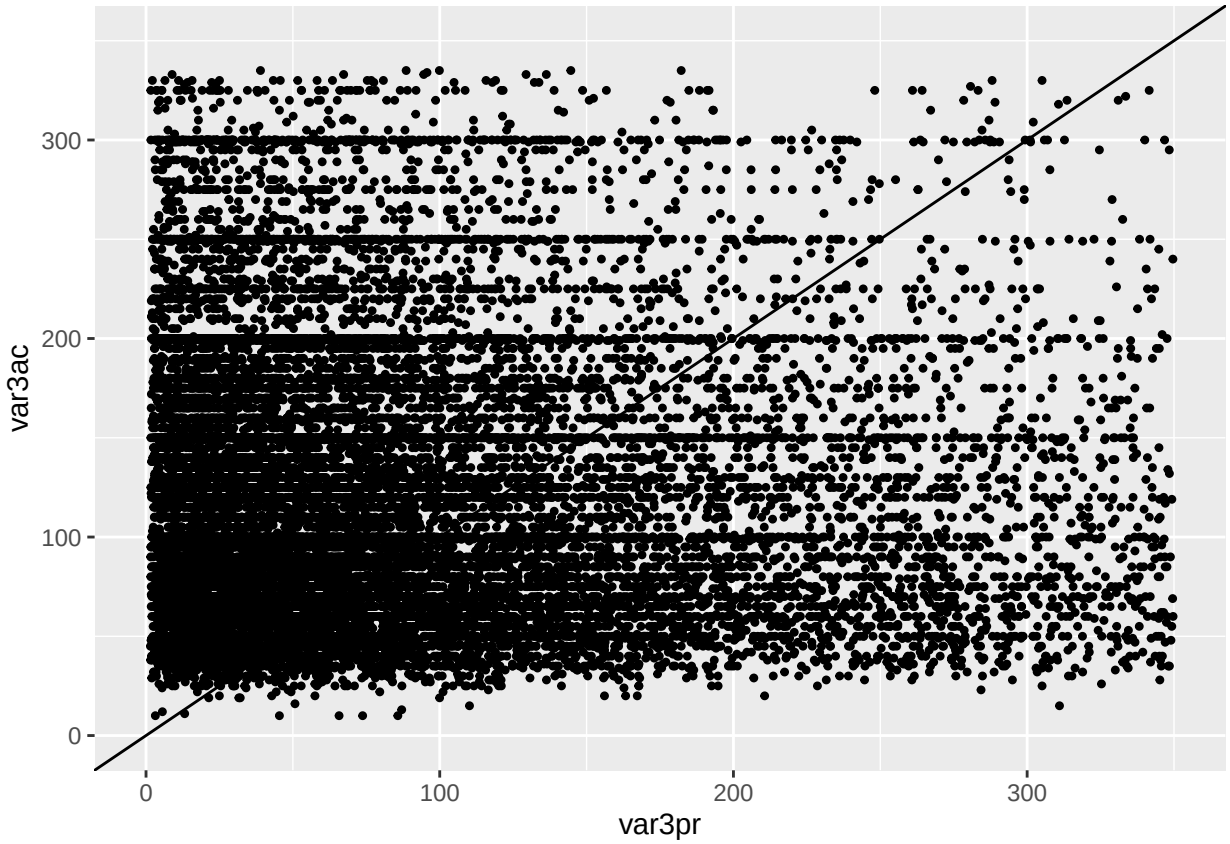
Simulation (3 Variables)

```
num_samples = nrow(bnb_trn_data_new)
var3predictionresult = rep(0, num_samples)

for (i in 1:num_samples) {
  room_type_val = sample(1:3,1)
  calculated_host_listings_count_value = sample(1:max(bnb_trn_data_new$calculated_host_listings_count),
  longitude_value = runif(1,min(bnb_trn_data_new$longitude), max(bnb_trn_data_new$longitude))
```

```
  var3prediction = data.frame(room_type = room_type_val, calculated_host_listings_count = calculated_hos
  var3predictionresult[i] = exp(predict(var3_aic_log,newdata=var3prediction))
}
#var3predictionresult
```

```
var3predictvsactual = data.frame(var3pr = var3predictionresult, var3ac = bnb_trn_data_new$price)
ggplot(var3predictvsactual) + geom_point(aes(x = var3pr, y = var3ac), pch=20) + geom_abline(intercept =
```

```
## Warning: Removed 1690 rows containing missing values (`geom_point()`).
```



- Also, we performed simulation of predicting the price of actual data with the log transformation model with three variables, and figured out that there is almost no correlation between our selected model and the actual data. In other words, the selected model was not optimal to predict the price. Therefore, we decided to add two more variables and do the same process again to find a better model.

5 Variables

```
model5 = lm(price ~ room_type + calculated_host_listings_count+ longitude + minimum_nights, data = bnb_
anova(modelvar3,model5)
```

```
## Analysis of Variance Table
```

```
##
## Model 1: price ~ room_type + calculated_host_listings_count + longitude
## Model 2: price ~ room_type + calculated_host_listings_count + longitude +
##     minimum_nights
##   Res.Df      RSS Df Sum of Sq     F    Pr(>F)
## 1  19658 52243606
## 2  19657 51997467  1    246140 93.05 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- H_0: β_4 (minimum_nights) = 0
- H_1: β_4 (minimum_nights) != 0
- Test statistic: 130.77
- p-value: 2.2e-16
- Decision on $\alpha = 0.1$: Reject H0, select model5
- We performed the ANOVA test to test whether adding predictor minimum_nights would make a better model to predict the price. The result shows significantly small p-value, so that we reject H_0 and select model5.

```
modelvar5 = lm(price ~ room_type + calculated_host_listings_count+ longitude + minimum_nights + reviews
anova(model5,modelvar5)
```

```
## Analysis of Variance Table
##
## Model 1: price ~ room_type + calculated_host_listings_count + longitude +
##     minimum_nights
## Model 2: price ~ room_type + calculated_host_listings_count + longitude +
##     minimum_nights + reviews_per_month
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1  19657 51997467
## 2  19656 51945202  1     52264 19.777 8.751e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- H_0: β_5 (reviews_per_month) = 0
- H_1: β_5 (reviews_per_month) != 0
- Test statistic: 25.122
- p-value: 5.428e-07
- Decision on $\alpha = 0.1$: Reject H0, select modelvar5
- Again, we performed the ANOVA test to test whether adding predictor reviews_per_month would make a better model to predict the price. The result shows significantly small p-value, so that we reject H_0 and select model with 5 predicting variables.

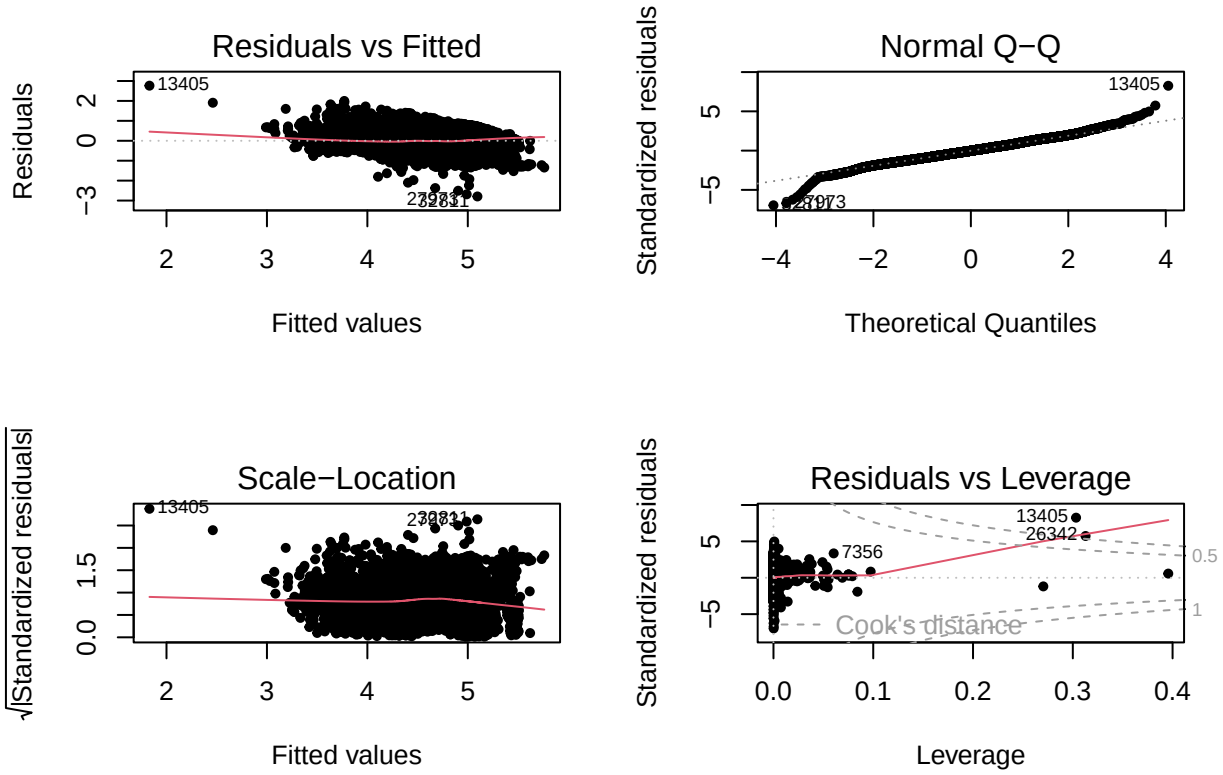
AIC and Log Transformation

```
var5original = lm(price ~ 1, data = bnb_trn_data_new)
var5_aic = step(var5original, scope = price ~ room_type * calculated_host_listings_count * longitude * r
coef(var5_aic)
```

```
##                                 (Intercept)
```

```
##                                  -2.830022e+04
##                                      room_type
##                                   7.463980e+03
##                                      longitude
##                                  -3.858640e+02
##                calculated_host_listings_count
##                                   1.866118e-01
##                                minimum_nights
##                                  -3.310810e+02
##                             reviews_per_month
##                                  -6.837651e+02
##                           room_type:longitude
##                                   1.019157e+02
##                      longitude:minimum_nights
##                                  -4.473254e+00
##                    longitude:reviews_per_month
##                                  -9.233046e+00
## calculated_host_listings_count:minimum_nights
##                                  -3.484929e-03
##              minimum_nights:reviews_per_month
##                                  -1.205303e+02
##    longitude:minimum_nights:reviews_per_month
##                                  -1.631016e+00
```

```
var5_aic_log = lm(log(price) ~ room_type + longitude + calculated_host_listings_count + minimum_nights +
+ calculated_host_listings_count:minimum_nights:reviews_per_month
, data = bnb_trn_data_new)
par(mfrow=c(2,2))
plot(var5_aic_log, pch = 20)
```



- In order to improve our model with 5 variables, we peformed AIC search with both direction and scoped it to the interaction between 5 predictors. Then, similar to what we did on 3 variables model, we performed a log transformation on the response variable (price). As a result, we got a new model that would predict the price better than the additive model with 5 predictors.

```
var5predict = data.frame(room_type = 2, calculated_host_listings_count = 1, longitude = -73.8765,minimu
exp(predict(var5_aic_log,newdata=var5predict))
```

```
##        1
## 65.43214
```

- Once again, we randomly selected one data point from the dataset, the one with room_type = 2(Private room), calculated_host_listings_count = 1, longitude = -73.95725, minimum_nights = 1, and reviews_per_month = 0.57 (the same data point we used for prediction with 3 variable aic model) and used the log transformation aic model of 5 variables to predict the price. As a result, the prediction (65.34578) is closer to the actual price($60) than the prediction with aic model of 3 variables.

```
num_samples = nrow(bnb_trn_data_new)
var5predictionresult = rep(0, num_samples)

for (i in 1:num_samples) {
  room_type_val = sample(1:3,1)
```

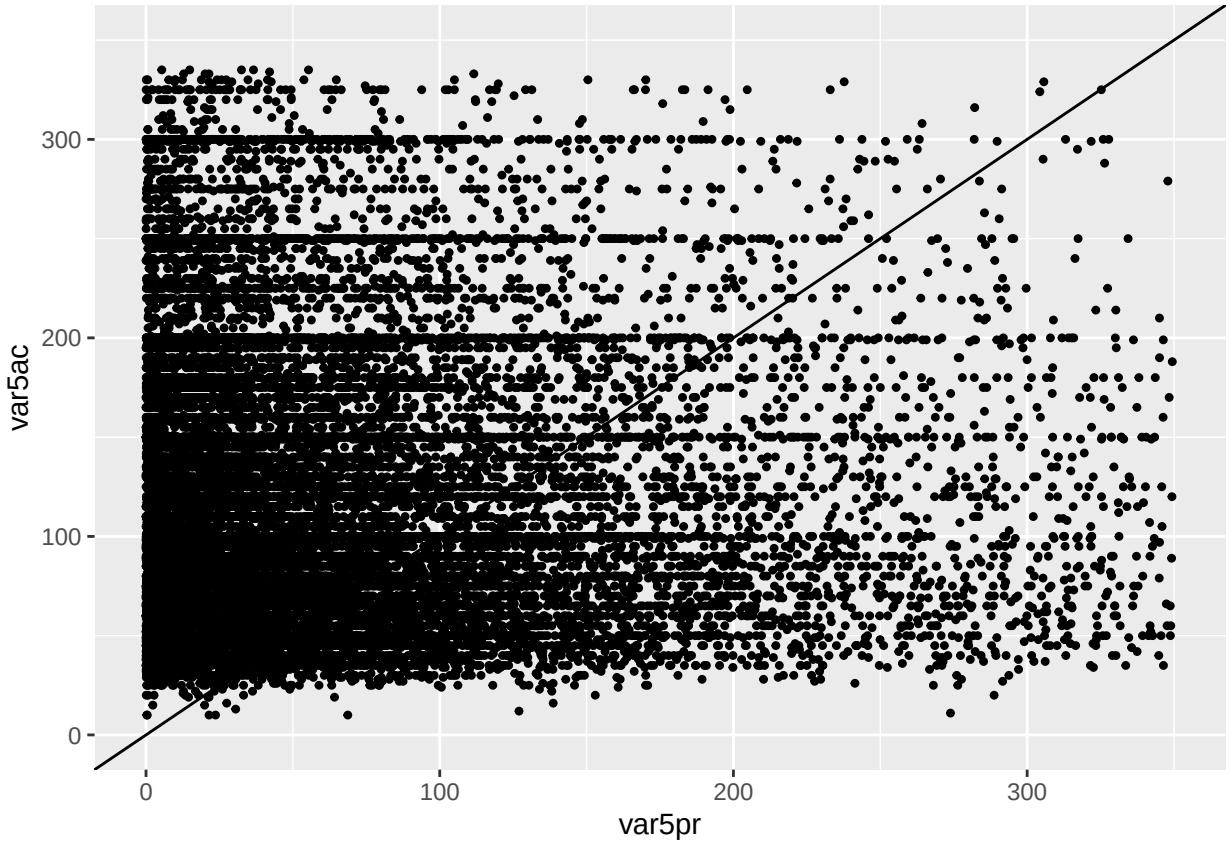
```
  calculated_host_listings_count_value = sample(1:max(bnb_trn_data_new$calculated_host_listings_count),
  longitude_value = round(runif(1,min(bnb_trn_data_new$longitude), max(bnb_trn_data_new$longitude)),5)
  minimum_nights_value = sample(1:40,1)
  reviews_per_month_value = round(runif(1,min(bnb_trn_data_new$reviews_per_month),20),2)

  prediction_train = data.frame(room_type = room_type_val, calculated_host_listings_count = calculated_
  var5predictionresult[i] = exp(predict(var5_aic_log,newdata=prediction_train))
}
#var5predictionresult
```

```
var5predictvsactual = data.frame(var5pr = var5predictionresult, var5ac = bnb_trn_data_new$price)
```

```
ggplot(var5predictvsactual) + geom_point(aes(x = var5pr, y = var5ac), pch=20) + geom_abline(intercept =
```

Simulation (5 Variables)

```
## Warning: Removed 165 rows containing missing values (`geom_point()`).
```



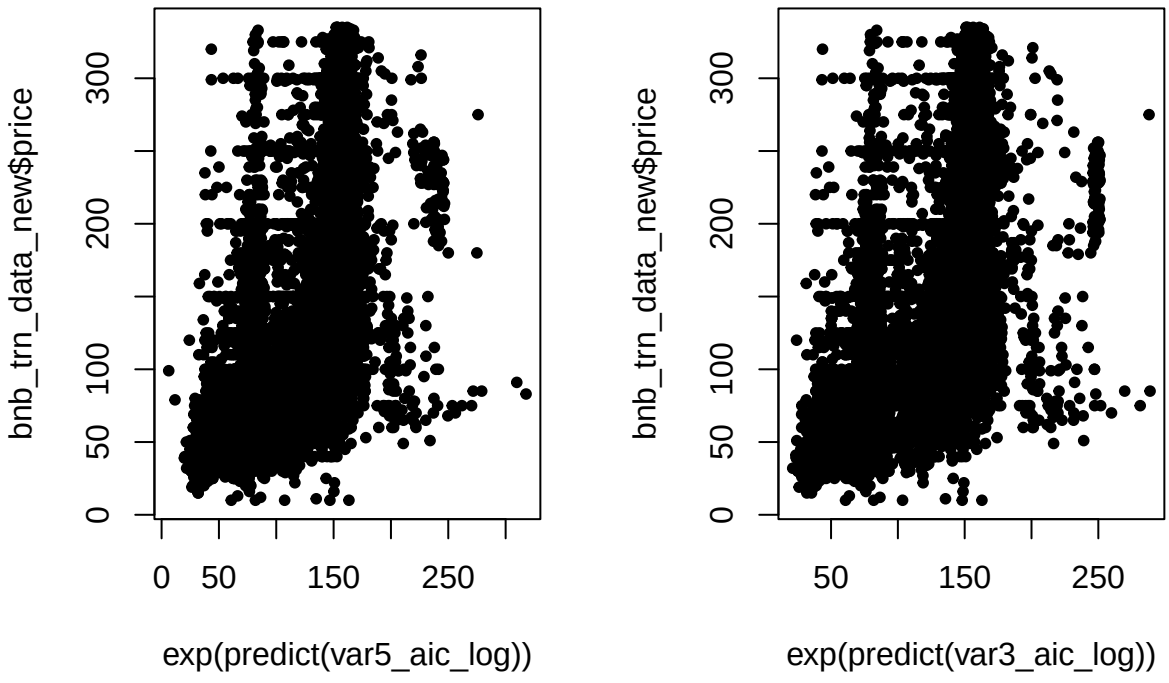
- Again, we performed simulation of predicting the price of actual data with the log transformation model with five variables, and figured out that there is a slightly higher correlation between our selected model and the actual data. In other words, the new selected model seems to better than the selected model with 3 variables.

```
anova(var3_aic_log,var5_aic_log)
```

```
## Analysis of Variance Table
##
## Model 1: log(price) ~ room_type + longitude + calculated_host_listings_count +
##     room_type:longitude + room_type:calculated_host_listings_count +
##     longitude:calculated_host_listings_count
## Model 2: log(price) ~ room_type + longitude + calculated_host_listings_count +
##     minimum_nights + reviews_per_month + room_type:longitude +
##     longitude:minimum_nights + calculated_host_listings_count:minimum_nights +
##     longitude:reviews_per_month + calculated_host_listings_count:reviews_per_month +
##     minimum_nights:reviews_per_month + calculated_host_listings_count:minimum_nights:reviews_per_mon
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1  19655 3197.9
## 2  19649 3162.2  6    35.747 37.021 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

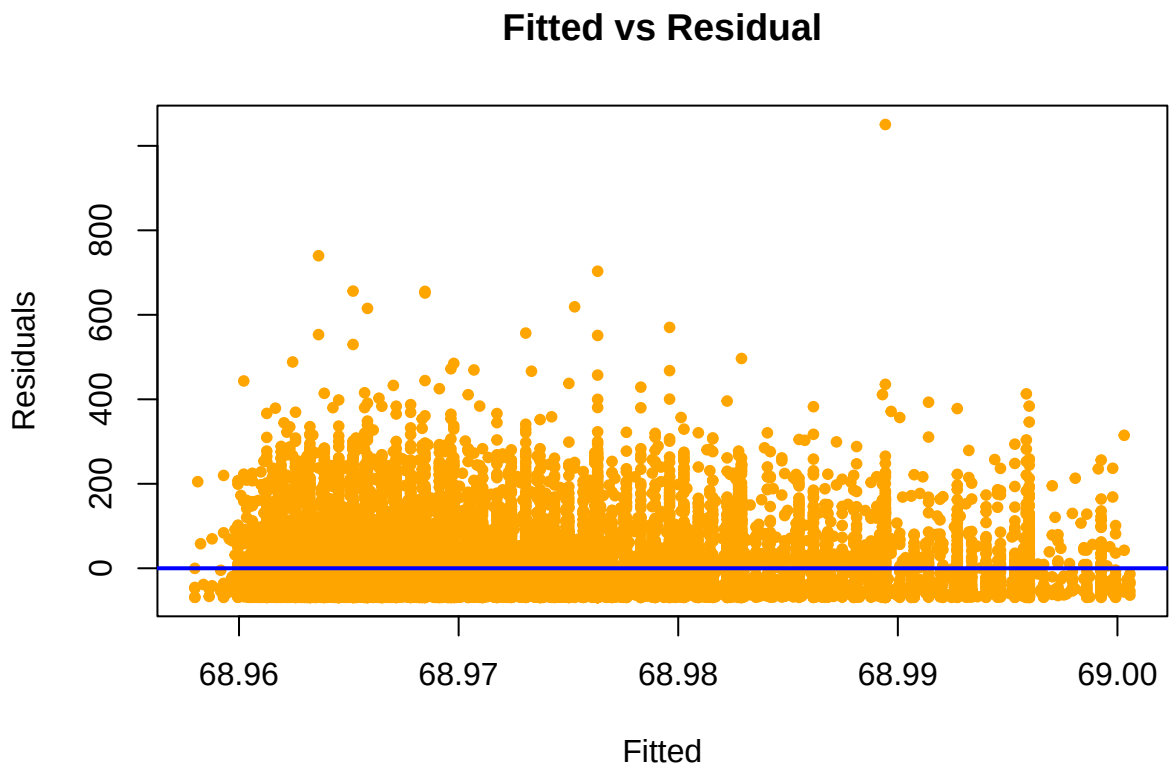
Predicted vs Actual

```
par(mfrow=c(1,2))
plot(exp(predict(var5_aic_log)), bnb_trn_data_new$price, pch = 20)
plot(exp(predict(var3_aic_log)), bnb_trn_data_new$price, pch = 20)
```



- In order to decide which model is actually better to predict the price between log transformed aic model with 3 variables and log transformed aic model with 5 variables, we performed ANOVA test again and plotted each model with the actual data. Since the p-value is signifcantly small and log aic model with 5 variable has the smaller RSS (3146.1) than another one (RSS: 3180.4), we choose the log aic model with 5 variables is better than the one with 3 variables.

```
var5_fit_model = lm(var5pr~var5ac,data=var5predictvsactual)
plot(fitted(var5_fit_model), resid(var5_fit_model), col = "orange", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted vs Residual")
abline(h = 0, col = "blue", lwd = 2)
```



- This fitted vs residuals plot of log aic model with 5 variables suggests that the model has constant variance and linearity.

```
rmse = function(actual, predicted) {
  sqrt(mean((actual - predicted) ^ 2))
}
```

```
rmse(exp(predict(var5_aic_log)), bnb_trn_data_new$price)
```

```
## [1] 51.69078
```

- The RMSE between the actual data and our selected model is not large.

Result

testing

After finding the best fitting model for the training data, we want to verify if our model correctly predicts the value from the testing data as well. The reason behind this check up is to see the difference in value of our prediction and the actual value. The level of difference tells us how close our model was, in terms of finding out the relationship among variables and response.

First step, we had to modify our data same as we did for the training dataset. We have removed the outliers for the testing data.

```
iqr_tst = IQR(bnb_tst_data$price,na.rm = TRUE)
iqr_tst
```

```
## [1] 105
```

```
quart_tst = quantile(bnb_tst_data$price, c(0.25, 0.5, 0.75), type = 1)
upperoutlier_tst = quart[3] + 1.5*iqr[1]
loweroutlier_tst = quart[1]-1.5*iqr[1]
c(upperoutlier_tst,loweroutlier_tst)
```

```
## 75% 25%
## 335 -89
```

```
rowtoremove1_tst = c(which(bnb_tst_data$price > upperoutlier))
rowtoremove2_tst = c(which(bnb_tst_data$price < loweroutlier))
bnb_tst_data_new <- bnb_tst_data[-rowtoremove1_tst,]
nrow(bnb_tst_data_new)
```

```
## [1] 4892
```

Next, same as before, we have first turned the neighbourhood_group and room_type characters into factors then into numeric values so that we could use them as numbers to calculate in the future. Plus, setup a model for testing equal to the model from the training.

```
bnb_tst_data_new$neighbourhood_group = as.factor(bnb_tst_data_new$neighbourhood_group)
bnb_tst_data_new$neighbourhood_group = as.numeric(bnb_tst_data_new$neighbourhood_group)
bnb_tst_data_new$room_type = as.factor(bnb_tst_data_new$room_type)
bnb_tst_data_new$room_type = as.numeric(bnb_tst_data_new$room_type)
```

5 variable model

```
var5_aic_log_tst = lm(log(price) ~ room_type + longitude + calculated_host_listings_count + minimum_nigh
+ calculated_host_listings_count:minimum_nights:reviews_per_month
, data = bnb_tst_data_new)
```

Simulation

With a model above, simulate and predict the values and store the values.

```
num_samples_tst = nrow(bnb_tst_data_new)
var5predictionresult_tst = rep(0, num_samples_tst)

for (i in 1:num_samples_tst) {
  room_type_val_tst = sample(1:3,1)
  calculated_host_listings_count_value_tst = sample(1:max(bnb_tst_data_new$calculated_host_listings_cou
  longitude_value_tst = round(runif(1,min(bnb_tst_data_new$longitude), max(bnb_tst_data_new$longitude))
  minimum_nights_value_tst = sample(1:40,1)
  reviews_per_month_value_tst = round(runif(1,min(bnb_tst_data_new$reviews_per_month),20),2)

  prediction_tst = data.frame(room_type = room_type_val_tst, calculated_host_listings_count = calculate
  var5predictionresult_tst[i] = exp(predict(var5_aic_log_tst,newdata=prediction_tst))
}
#var5predictionresult_tst
```

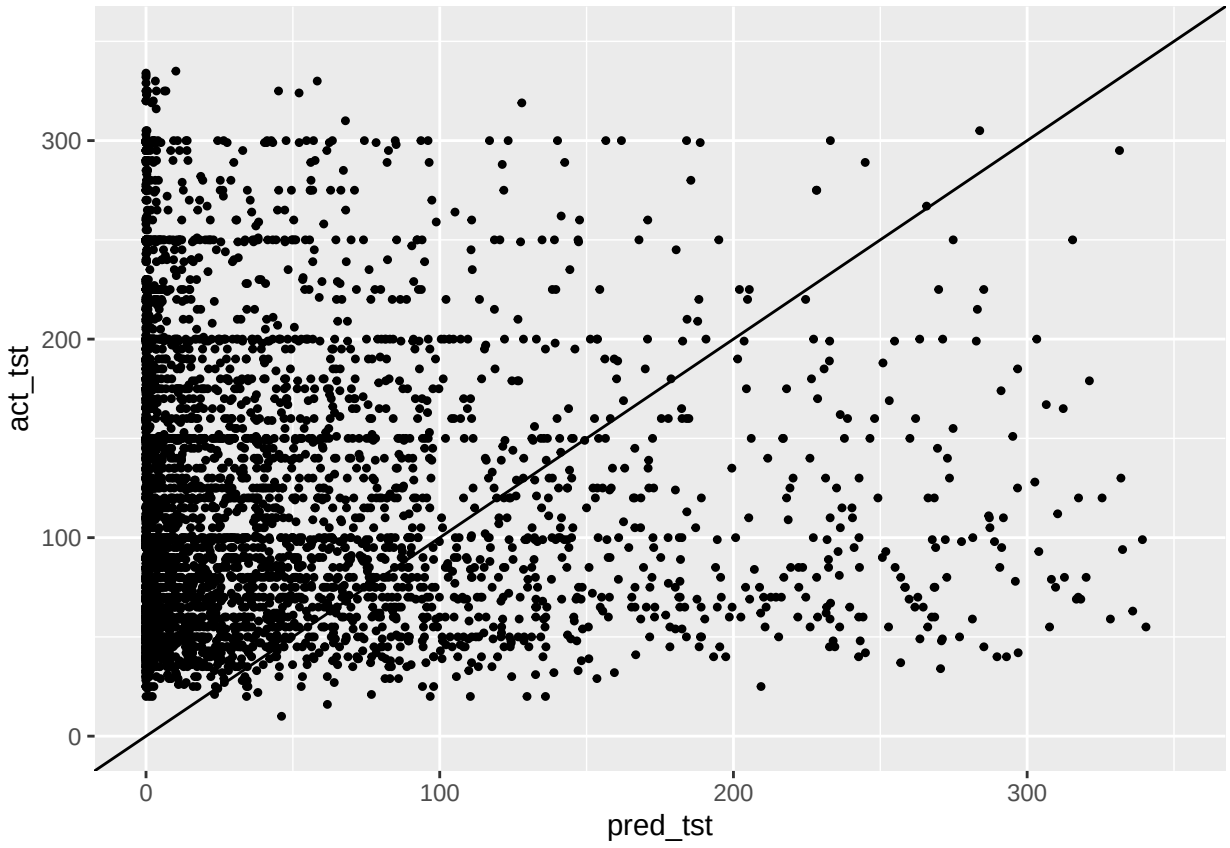
After predicting the values, we used these values to compare with the actual values of the price. We wanted to see the linear relationship of predicted and actual when we plotted. Therefore, we have additionally plotted 'Fitted vs Residual' graph to check the linearity of predicted and actual.

Prediction vs Actual

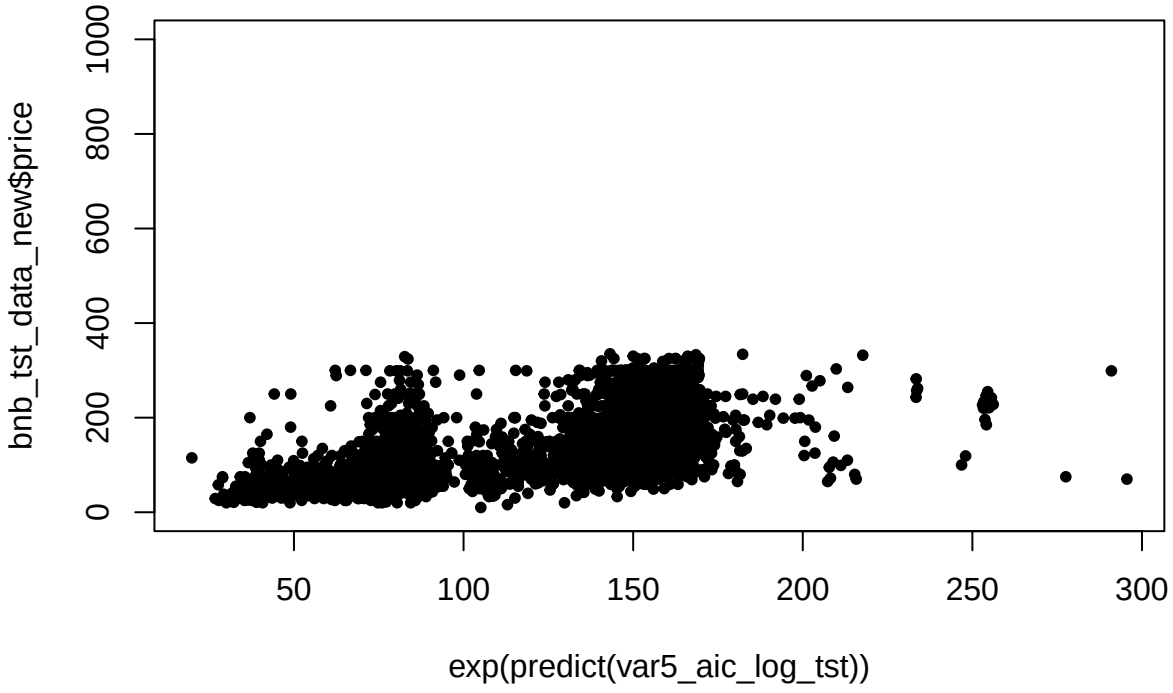
```
var5predictvsactual_tst = data.frame(pred_tst = var5predictionresult_tst, act_tst = bnb_tst_data_new$pri
```

```
ggplot(var5predictvsactual_tst) + geom_point(aes(x = pred_tst, y = act_tst), pch=20) + geom_abline(inte
```

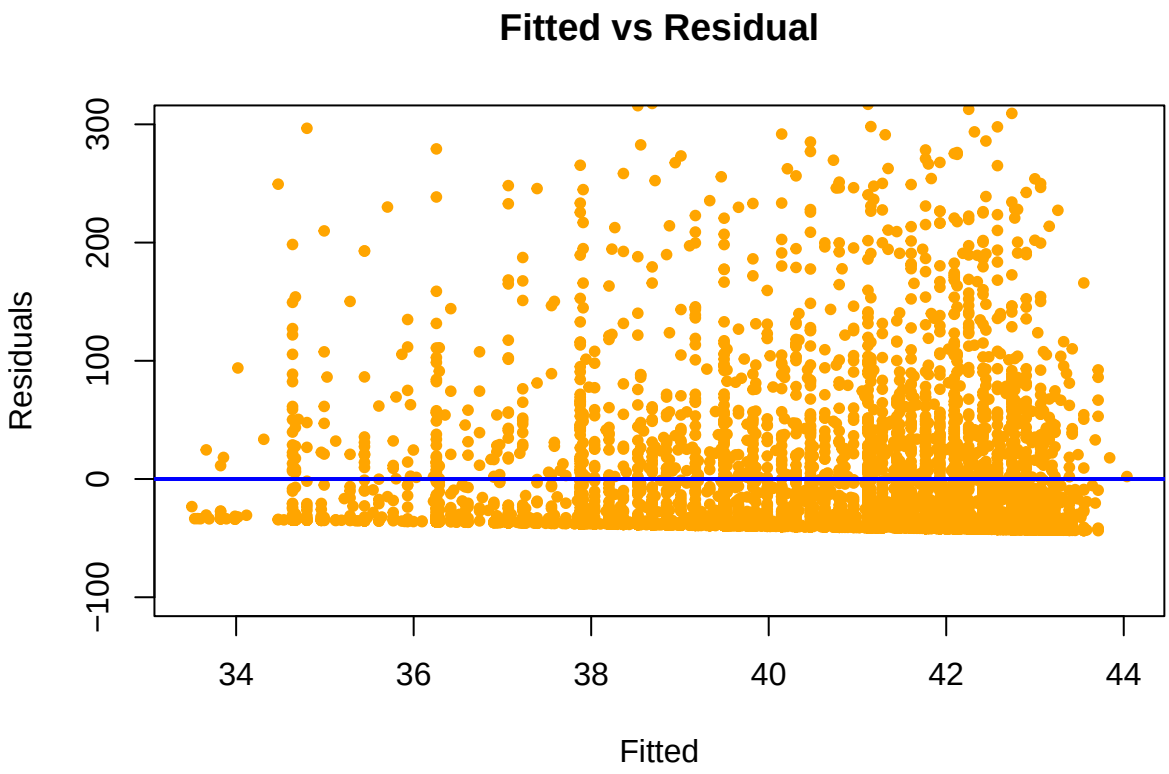
```
## Warning: Removed 31 rows containing missing values (`geom_point()`).
```




```
plot(exp(predict(var5_aic_log_tst)), bnb_tst_data_new$price, pch = 20,ylim=c(0,1000))
```




```
var5_fit_model_tst = lm(pred_tst~act_tst,data=var5predictvsactual_tst)
plot(fitted(var5_fit_model_tst), resid(var5_fit_model_tst), col = "orange", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted vs Residual",ylim=c(-100,300))
abline(h = 0, col = "blue", lwd = 2)
```



Considering two graphs - predicted value vs actual value and fitted vs residual - our team has though there might be a chance of these graphs showing linearity with a slope close to one in predicted vs actual value graph. Since it was hard to find the relationship just by looking at the first graph, our team has decided to examine the fitted vs residual plot.

The fitted vs residual graph shows little noise in the graph. However, it is hard to say that the graph is showing certain trend other than a straight line with $y = 0$. We could say that this graph shows that there might be a linear relationship between the predicted and actual value. These two graphs shows that our model, that we have found out through numerous model diagnostic methods, might be accurate enough to make prediction of the price based on the data.

Our team has successfully tested out, which model fits the best, considering numerous variables. The model that we have used was $log(price) = roomtype + longitude + hostlistings + minimumnights + reviewspermonth + roomtype : longitude + longitude : minimumnights + hostlistings : minimumnights + longitude : reviewspermonth + hostlistings : reviewspermonth + minimumnights : reviewspermonth + hostlistings : minimumnights : reviewspermonth.$

This model was able to predict the price points with some noise of error in the value. The usage of anova, log transformation, linearity testing led our team to approach this model, like mentioned in method.

However, the process of ending up with this model was not easy at the first place. Our team had an initial thought of thinking 3 variables would be enough to explain the complexity of the relationship of variables and price. But, after plotting the model with 3 variables, we have decided 3 variables were not enough. We had to test out other variables again, and reduce the error between value of predicted and actual price point of airbnb.

We believe that someday, this COVID-19 pandemic will disappear and airbnb will be back in business like it was before this moment. We were able to figure out that room type, longitude, host listings number, minimum nights, reviews per month influenced the price point of airbnb. Considering the geographical

location of neighborhoods, size of the house, the correlation among the variables and price, might come as a easy connected dots. The relationship of price and such variables will be discussed below.

Discussion

```
mean(airbnbdata$price[airbnbdata$room_type == "Entire home/apt"])
```

```
## [1] 208.1485
```

```
mean(airbnbdata$price[airbnbdata$room_type == "Private room"])
```

```
## [1] 87.08874
```

```
mean(airbnbdata$price[airbnbdata$room_type == "Shared room"])
```

```
## [1] 62.42836
```

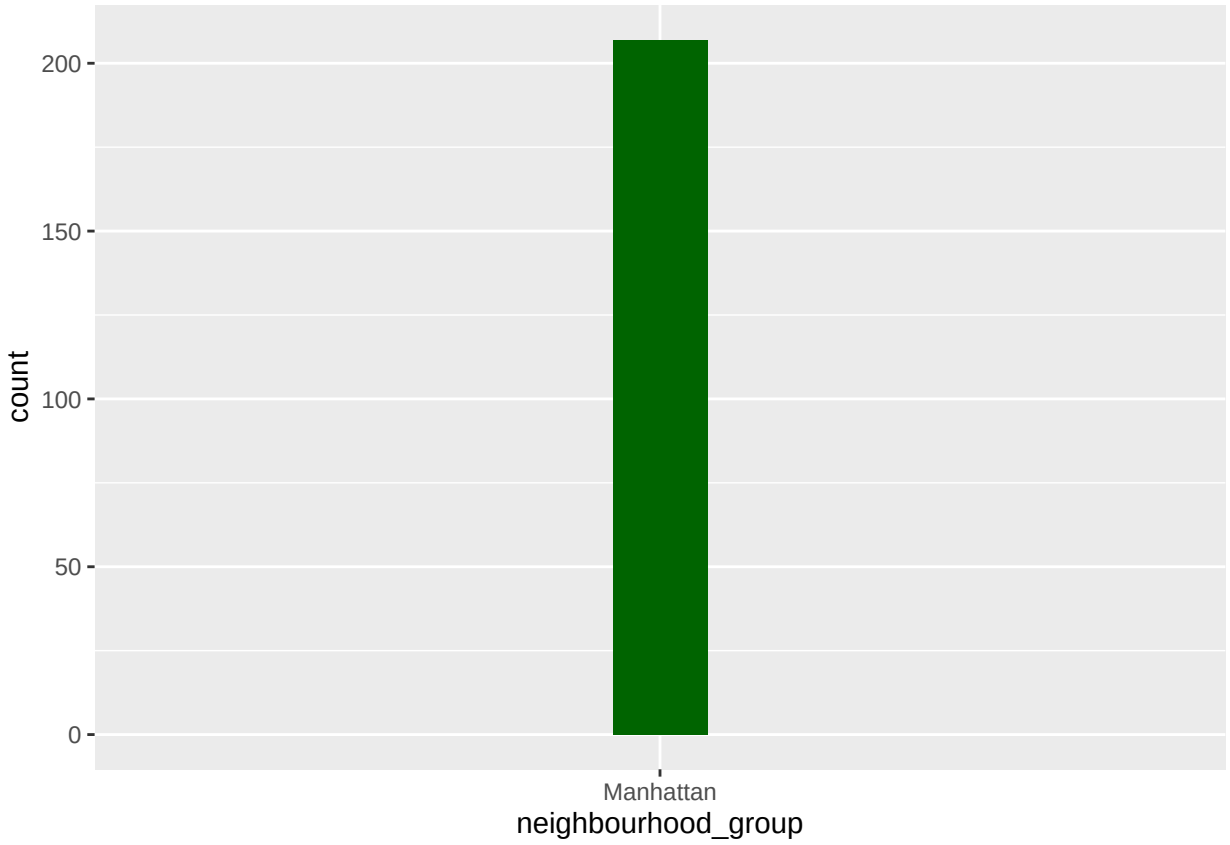
The correlation between room_type and price is -0.9. The number above shows that the average price of Entire home/apt is the greatest, Private Room, being the next, and Shared Room for the last. The correlation shows that the price is negatively related to Room type. We have made the room type into factors, then turned it into numerics. Entire home/apt= 1, Private Room= 2, Shared Room= 3.

```
t = data.frame(name = c("Bronx","Brooklyn","Manhattan","Queens","Staten Island"), min = c(min(airbnbdata
min(airbnbdata$longitude[airbnbdata$neighbourhood_group == "Brooklyn"]),
min(airbnbdata$longitude[airbnbdata$neighbourhood_group == "Manhattan"]),
min(airbnbdata$longitude[airbnbdata$neighbourhood_group == "Queens"]),
min(airbnbdata$longitude[airbnbdata$neighbourhood_group == "Staten Island"])), max =
c(max(airbnbdata$longitude[airbnbdata$neighbourhood_group == "Bronx"]),
max(airbnbdata$longitude[airbnbdata$neighbourhood_group == "Brooklyn"]),
max(airbnbdata$longitude[airbnbdata$neighbourhood_group == "Manhattan"]),
max(airbnbdata$longitude[airbnbdata$neighbourhood_group == "Queens"]),
max(airbnbdata$longitude[airbnbdata$neighbourhood_group == "Staten Island"])),
price = c(mean(airbnbdata$price[airbnbdata$neighbourhood_group == "Bronx"]),
mean(airbnbdata$price[airbnbdata$neighbourhood_group == "Brooklyn"]),
mean(airbnbdata$price[airbnbdata$neighbourhood_group == "Manhattan"]),
mean(airbnbdata$price[airbnbdata$neighbourhood_group == "Queens"]),
mean(airbnbdata$price[airbnbdata$neighbourhood_group == "Staten Island"])))
t
```

```
##            name       min       max     price
## 1         Bronx -73.93190 -73.78158  80.73775
## 2      Brooklyn -74.03942 -73.85676 128.41325
## 3     Manhattan -74.01801 -73.90855 195.92768
## 4        Queens -73.95927 -73.71299  95.37497
## 5 Staten Island -74.24442 -74.06356  88.97561
```

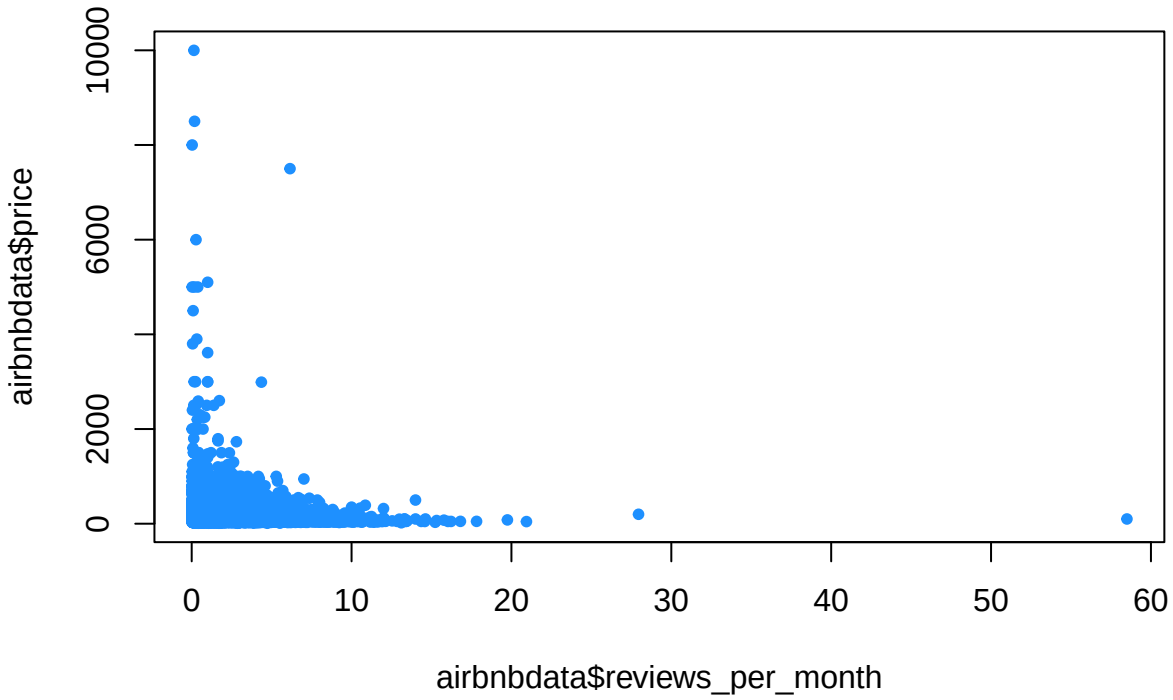
The correlation between longitude and price is -0.59. As the values of longitude increase, it means that far away from downtown New York City. Therefore, the convenience and the number of tour spots decrease. It results in a negative value of the correlation between price and longitude.

```
library(ggplot2)

onewith372 = airbnbdata[airbnbdata$calculated_host_listings_count == 327,]
ggplot(data = onewith372) + geom_bar(aes(x = neighbourhood_group),position = 'dodge', width = 0.1, fill
```



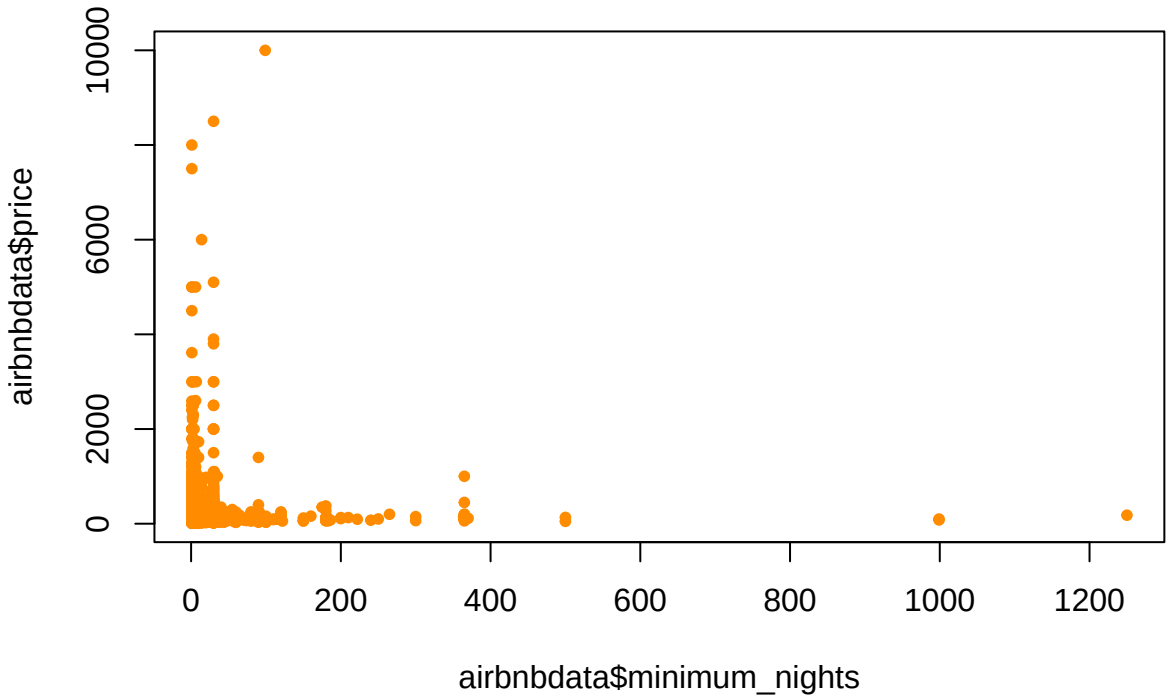
The correlation between calculated_host_listings_count and price is 0.31. As hosts with more numbers of the room have a high possibility of earning higher incomes, the rooms they own have higher possibility of locating near or at the center of the city. Therefore, the value of the correlation between the amount of listing per host and price is shown positive.

```
plot(airbnbdata$price ~ airbnbdata$reviews_per_month, pch = 20, col = "dodgerblue")
```



The correlation between reviews_per_month and price is -0.2. To have more number of reviews per month, the demand for rooms should be proportional. Since people tend to stay cheaper houses, the demand for the high price of Airbnb is lower than the low price of Airbnb. It results in the negative value of the correlation between price and reviews per month.

```
plot(airbnbdata$price ~ airbnbdata$minimum_nights, pch = 20, col = "darkorange")
```



The correlation between minimum_nights and price is 0.12. The plot suggests that there is almost no correlation between price and the minimum nights. However, since the correlation coefficient between price and the minimum nights is 0.12, we can predict that increasing minimum nights may imply a few people tend to stay in more expensive rooms.

Appendix

We used the following statistical methods:

- ANOVA Test

```
anova(var3_aic_log,var5_aic_log)
```

This test is to find out p value by comparing two different models, to see the significance of the additional variable.

- Log Transformation

```
var5_aic_log = lm(log(price) ~ room_type + longitude + calculated_host_listings_count + minimum_nights +
+ calculated_host_listings_count:minimum_nights:reviews_per_month
, data = bnb_trn_data_new)
```

Log Transformation allows the model to express normality of the data.

- AIC Selection

```
var5original = lm(price ~ 1, data = bnb_trn_data_new)
var5_aic = step(var5original, scope = price ~ room_type * calculated_host_listings_count * longitude * r
coef(var5_aic)
```

AIC selection finds the optimal model to improve linearity of the data.

- Multiple Linear Regression

```
modelvar3 = lm(price ~ room_type + calculated_host_listings_count+ longitude, data = bnb_trn_data_new)
```

Multiple Linear Regression is used to not only see the relationship of response and a single variable, but also see the dynamic changes of the response due to multiple variables.

- Fitted versus Residuals plot

```
plot(fitted(var5_fit_model), resid(var5_fit_model), col = "orange", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted vs Residual")
```

Fitted versus residuals plot shows the linearity and the check the constant variance of the model.

- Outlier removal

```
iqr = IQR(bnb_trn_data$price,na.rm = TRUE)
quart = quantile(bnb_trn_data$price, c(0.25, 0.5, 0.75), type = 1)
upperoutlier = quart[3] + 1.5*iqr[1]
loweroutlier = quart[1]-1.5*iqr[1]
c(upperoutlier,loweroutlier)
```

Outlier removal is crucial, because outlier may distort the overall trend of the data.

- Prediction

```
exp(predict(var5_aic_log,newdata=var5predict))
```

Prediction method was used to predict the price point, given certain condition.

- Simulation

```
for (i in 1:num_samples) {
  room_type_val = sample(1:3,1)
  calculated_host_listings_count_value = sample(1:max(bnb_trn_data_new$calculated_host_listings_count),1
  longitude_value = round(runif(1,min(bnb_trn_data_new$longitude), max(bnb_trn_data_new$longitude)),5)
  minimum_nights_value = sample(1:40,1)
  reviews_per_month_value = round(runif(1,min(bnb_trn_data_new$reviews_per_month),20),2)

  prediction_train = data.frame(room_type = room_type_val, calculated_host_listings_count = calculated_l
  var5predictionresult[i] = exp(predict(var5_aic_log,newdata=prediction_train))
}
```

Simulation method was used to randomly generate conditions, in order to predict the price point and compare them in the future. By randomly generating conditions, it allows us to check if the prediction are well distributed.

- RMSE

```
rmse = function(actual, predicted) {
  sqrt(mean((actual - predicted) ^ 2))
}
```

RMSE shows the level of difference between actual and predicted value.

- Train, Test Data Split

```
bnb_trn_idx  = sample(nrow(airbnbdata), size = trunc(0.80 * nrow(airbnbdata)))
bnb_trn_data = airbnbdata[bnb_trn_idx, ]
bnb_tst_data = airbnbdata[-bnb_trn_idx, ]
bnb_trn_data <- na.omit(bnb_trn_data)
bnb_tst_data <- na.omit(bnb_tst_data)
```

Train data is used to find the optimal model before predicting the outcome with the test model.

- Correlation Matrix

```
library(ggcorrplot)
correlation <- cor(bnb_trn_data_new)
correlation2 <- round(cor(correlation,use="complete.obs"), 2)
options(repr.plot.width=12, repr.plot.height=12)
ggcorrplot(correlation2, lab = TRUE, colors = c("orange", "white", "blue"), show.legend = F, outline.co
```

Correlation Matrix is used to see the relationship among the variables of the data. The level of relationship could be ranged from -1 to 1.