KOGOD SCHOOL *of* BUSINESS
AMERICAN UNIVERSITY • WASHINGTON, DC

ITEC 621 Predictive Analytics Project

New York Airbnb Pricing in the COVID-19 Age

Thursday Section

Team 5

Last Updated: 29 April 2021

Deliverable: 4

1. Business Case: Throughout the COVID-19 pandemic, the lodging industry has struggled mightily (Glusac). But amid cancelations and vacancies everywhere is a telling trend: short-term rentals have fared better than their large hotel competitors (Oliver). According to an analysis from data firm STR, hotel occupancy had dropped 77.3 percent year-over-year in the last week of March 2020, compared to a less than 50 percent downturn for short-term rentals in the same period (Mooney). An extended disruption to the hotel industry presents a window of opportunity for Airbnb and other short-term rental companies, which many Americans perceive to be safer options (Medine). By conducting an analysis of the factors that determine a New York Airbnb's nightly price, including COVID-19 cases, hospitalizations, and deaths in the city on particular dates, and by developing a more accurate, finely-tuned predictive pricing model, Airbnb can achieve greater success in America's most highly populated city. This includes attracting new renters and customers to the Airbnb platform, providing a reliable extra source of income to more renters during a period of economic uncertainty, and capitalizing on a down period for hotels to emerge from the pandemic with a larger share of the lodging industry.

2. Business Question: In the interest of properly meeting customer demand and maximizing revenue for renters and Airbnb, what are the main drivers of effective per-night pricing for Airbnb listings in the New York boroughs of Manhattan, Queens, Brooklyn, the Bronx, and Staten Island, including bedrooms, bathrooms, room type, and days available per year? And as the COVID-19 pandemic continues, how might the present severity of the pandemic in New York impact how nightly prices are calibrated?

3. Analytics Question: What is the effect of an Airbnb listing's number of bedrooms, New York borough, room type (i.e. "entire home/apt" or "private room"), days available per year, and amenities (i.e. "air conditioning" and "WiFi") on its price per night? And to what extent does the number of NYC COVID-19 cases, deaths, and hospitalizations on the day of the listing's most recent review have an effect on its price per night?

The outcome variable of interest (3.1) is price (USD), which is quantitative and truncated at 10. The analytics question we examine (3.2) is quantitative, intended to explore which factors are most important in determining the price. And key predictors (3.3) include "bedrooms" (numeric), "borough" (categorical), "WiFi" (binary), "CableTV" (binary), "availability_365" (binary), and COVID-19 "Case Count" (numeric). Our analysis explores quantitative parametric (WLS and ridge regression) and non-parametric (random forest) modeling methods. While we are interested in predictive accuracy, our primary analytics goal is interpretation, to adequately explain model insights to Airbnb executives and property renters.

4. Data Set Description: "Airbnb US dataset" from Kaggle.com contains 158,250 Airbnb listings from eight US states and the District of Columbia, mostly from the years 2018, 2019, and 2020 (Karun). Acknowledging the differences between cities, we are focusing on listings only from New York, or 36,900 of those observations, of which just 24,029 are complete. Each listing's most recent review date, no more than 14 days after the most recent rental, per Airbnb policy (*How*), was joined with a data set from the NYC Department of Health recording the city's daily COVID-19 statistics (New York). The combined data set includes 37 quantitative, binary, and categorical predictors. Predictors of interest for price, in addition to the small group listed above, include binary variables for amenities like "AC," "patio or balcony," and "Parking"; numeric variables like "min_nights" (the smallest rental term possible, in nights), the daily NYC COVID-19 hospitalization, and daily NYC COVID-19 death count; and a categorical variable for room type, with levels "Entire home/apt," "Hotel room," "Private room," and "Shared room."

5. Descriptive Analytics

5.1-5.2 Visual and Quantitative Analytics: Again, our outcome variable for this New York Airbnb listing analysis is price, in US dollars, per night. The distribution of the price variable is right skewed, demonstrated both in histograms and right-side departure from the QQ line (explained further in OLS testing below). We believe this reflects high-priced outliers.

Initial descriptive analysis focused on important predictors highlighted above in "Analytics Question" and "Data Set Description." This includes "borough," a categorical variable that serves as the primary marker of listing location, as well as "room type." Approximately 82.7% of listings are located in Manhattan and Brooklyn (almost equally divided), and entire houses and private rooms constitute 97% of all listings (almost equally divided). Our preliminary analysis also delved into the most commonly understood size differentiators in Airbnb listings. Around 83% of listings have only one bathroom, 78% only have one bedroom, and 61% only have one bed. An examination of relationships between predictors reveals only a few strong correlations overall, and they are positive. The strongest are between "Beds" and "accommodates" (0.737), "Death Count" and "Hospitalized Count" (0.854), and "Case Count" and "Hospitalized Count" (0.866), all of which are logical and intuitive. "Beds" (0.3961), "bedrooms" (0.4483), and "accommodates" (0.5345) have relatively stronger (positive) correlation with price than the rest of the predictors.

There are several categorical and binary variables for which the level impacts Airbnb listing price. A visual inspection of boxplots and significant ANOVA tests indicates that price varies by borough (Manhattan has the highest average); by room type (Hotel room has the highest average); and by the presence of some amenities like AC, pool (higher average price when present) and free parking (surprisingly, lower average price when present). Other amenities, like waterfront location, Wifi, and bed linens produce insignificant ANOVA tests.

Finally, an initial observation that hotel rooms seem to be more concentrated in Manhattan led to a Chi square test between room type and borough. A significant Chi square test result confirms dependence between borough and room type.

5.3 Data Pre-Processing and Transformations*:* The 158,250 US Airbnb listings were filtered down to 36,900, narrowing the focus to just New York. Next, we removed several nuisance variables describing the host, including ID (one per listing), host_ID (one per host), name (one per listing), and instant_bookable (binary). Other variables excluded were neighborhood, of which there were hundreds overall and multiple for each listing, causing confusion; latitude and longitude, which did not lend themselves to interpretation; and property_type, which had dozens of factors. Using str_detect() and ifelse(), text strings contained in a single amenities variable became 15 indicator variables for amenities. After all date values in last_review were transformed to a YYYY-MM-DD format, three variables highlighting daily New York COVID statistics (cases, hospitalizations, and deaths) were added to the initial data set with left_join(). This linked COVID-19 statistics to the Airbnb listing's last review date. A right-skewed distribution of the response variable required logging for predictive modeling, including ordinary least squares (testing below). Finally, we removed listings with prices of more than $400 per night (more than four standard deviations above the mean) to limit the effect of outliers.

6. Modeling Methods and Model Specifications

6.0 Initial Set of Predictors: To analyze the effects of some more obvious differences between Airbnb listings, the initial set of predictors included the categorical room type and the numeric number of bathrooms, bedrooms, maximum guests accommodated. Also included were indicators for 11 amenities—heating, elevator, patio or balcony, AC, CableTV, WiFi, free parking, pool, private entrance, coffee maker, and cooking equipment—we believed, if present, might impact price, and transformed these variables into binary predictors. We utilized the categorical borough to observe how different areas of New York affect the listing price. To understand how booking requirements set by the host impact price, we included the numeric minimum nights and availability (out of 365 days). And finally, to observe COVID-19's impact on price, we included case and death counts for the day of the listing's last review (0 before March 11, 2020).

6.1 Initial OLS Modeling: An OLS model fit using these 21 predictors produced a significant F test, indicating that the model is better at predicting price than the null model. Significant predictors at the .01 significance level include accommodates, CableTV, Wifi, free parking, pool, elevator, patio or balcony, cooking equipment, borough (Manhattan), room type (Hotel room), room type (private room), bathrooms, bedrooms, and availability.

6.2 OLS Assumptions Tested: Airbnb sets its minimum price to rent at $10, meaning the response (price) is not technically a continuous variable (YC). In addition, the histogram of price shows a right-skewed distribution, so, again, we will need to log the response variable in order to use OLS. The histogram of residuals appears normal, and the qq-plot of residuals shows normality in the middle quantiles with obvious tail-wagging only in the upper quantiles (EN). There is a slight concern about independence of the predictors. A condition index of 43.71 is slightly high, but below the threshold of 50, suggesting that our model's multicollinearity is tolerable. In addition, no predictor's GVIF value (or VIF for one-factor variables) approaches 10—the highest is 2.71—which suggests that this assumption is met (XI). While not present for most predictors, we detected slight curvature in accommodates, bathrooms, bedrooms, and min_nights plotted against price (LI). However, subsequent exploration with polynomial terms (see "Goodies") found little to no improvement in MSE. For this reason, the small amount of curvature, and our emphasis on interpretability, we opted not to include polynomial terms in our initial specification. All of the observations and errors (OI & EI) are believed to be independent with regard to price. There is likely serial correlation between date and COVID case numbers, but we are focused on the study of case numbers and price, not date. The mean of the residuals is extremely small (-9.90548e-16), confirming the assumption that the error average is 0 (EA). Error variance is not constant, as a significant Breusch Pagan test suggested the presence of heteroskedasticity (EV). We will need to fit a weighted least squares model to correct for that.

6.3 Model Specifications Evaluated (and Variable Selection): The first model specification used in this exercise was the initial set of 21 predictors selected using a business perspective. The second model specification for all three models was selected using stepwise variable selection and a conservative p-value threshold of .01 to include only the most significant predictors. The full model for this variable selection exercise was all predictors except last_review, which includes dates and would have yielded an error. The lower bound was the null model plus "Case Count," a variable whose inclusion was essential for answering part of our business question, regardless of its significance. Stepwise variable selection pared our second specification down to 19 predictors: the numeric maximum guests accommodated, bathrooms, bedrooms, rev_per_month (review per month of an Airbnb listing), host_listings (the total number listings operated by a particular listing's host), availability, and case count; binary variables indicating whether the host is a "superhost," whether the listing is "instant bookable," and the presence of cable TV, pool, free parking, elevator, cooking equipment, AC, and patio or balcony; and the categorical New York borough and room type. All are significant at the .01 significance level except the Brooklyn (p-value .048), Queens (.018), and Staten Island (.238) levels of borough; the shared room level of room type (.02); and case count, which was not close to significant. Because variable selection resulted in a more compact subset of predictors, we will refer to this specification as the "small" or "stepwise" subset of predictors or specification.'

6.3a OLS Assumptions (Second Set of Predictors): Again, Airbnb sets its minimum price at $10, meaning the response variable is truncated at 10 and not technically continuous (YC). The QQ plot reveals tail-wagging in the upper quantiles, and right skewness is confirmed by a histogram of price. The response variable will need to be logged using this specification as well (EN). With a smaller set of significant predictors, the condition index has dropped to a decent 29.33, and the highest VIF or GVIF value is the 2.44 attached to "accommodates," meaning multicollinearity levels are fine for this specification (XI). We again detected slight curvature (LI) for predictors bathrooms, bedrooms, host listings, and accommodates plotted against price. But given that our separate exploration of polynomial terms (see "Goodies") later found little to no improvement in MSE, the small amount of curvature, and our focus on interpretability, we chose not to include polynomial terms in our small specification. All observations and errors are believed to be independent (OI & EI), the mean of the residuals is 0 (EA), and error variance is not constant, as confirmed by a significant Breusch Pagan test (EV). We will also need to fit a weighted least squares model using these predictors on price.

6.4 Methods Evaluated: First, we examined a weighted least squares model, as suggested above, for both specifications/sets of predictors. Next, we evaluated both specifications using ridge regression to reduce dimensionality and associated model variance with shrinkage, but, for analysis purposes, to avoid risking completely dropping variables like COVID-19 case count. And finally, we examined both sets of predictors using a random forest model. This was to analyze how much it improved predictive accuracy and to extract a clearer understanding of which predictors are most "important" in predicting price. We logged the response variable, price, as we constructed all models. This corrected for right skewness as required prior to using OLS/WLS. In the cases of ridge regression and random forest, we logged the response to allow a like for like comparison of MSE between WLS and these models.

6.5 Cross-Validation Testing: In all six combinations of model and specification, we tested using the 10-fold cross validation for its proven results as a robust evaluation method. To address severe heteroskedasticity that was not corrected with the first iteration of weighting, we weighted a second time before fitting our final WLS models. Using the caret package and 10-fold cross validation, we obtained a mean squared error of 0.1643 for the initial WLS specification and 0.1582 for the small specification. Again logging our Y vector for ridge regression, we used cv.glmnet(), and its default of 10-fold cross validation, to obtain MSEs of 0.1652 for the unweighted initial specification, 0.1592 for the unweighted small specification, 0.1651 for the weighted initial specification, and 0.1592 again for the weighted small specification. Finally, using 10-fold cross validation on random forest yielded MSEs of 0.1524 for the initial specification and 0.1413 for the small specification.

6.6 Final Method/Specification Selected: While the random forest model and small specification were best for predictive accuracy, we honored our primary goal of interpretability to best serve our audience of Airbnb executives and renters. Consequently, we selected the WLS model with the small specification, our lowest-MSE parametric model (0.1582). We then fit this model on the entire data set with lm(), logging the response and adding its corresponding weight vector. We are keeping all predictors from the small specification. This includes "bathrooms," which was significant in the initial OLS fitting and peculiarly became insignificant in the process of weighting the model. Its presence allows for comparison with other predictors, and its absence would not improve the MSE.

7. Analysis of Results: The model explains around 58% of the variability in NY Airbnb prices (with the majority of outliers excluded), as demonstrated by the R-squared value. The first summary plot indicates that minor heteroskedasticity is still present, but this model was ultimately selected because it yielded our lowest 10-fold CV test MSE value (step 6.5). The QQ plot shows that the residuals are relatively normally distributed, with minor tail wagging on the right. The output displays all predictors as significant, with the exception of "bathrooms," as explained above and shown in the "coefplot" (on the zero line).

Quantitative predictors: all interpretations are "on average and holding everything else constant":
The number of bedrooms has the highest effect on the response, with a 10% price increase for each additional bedroom. Second highest is capacity (accommodates); as the capacity increases by 1, the listing price increases by around 8%. Next is the listing's number of reviews per month, but in the opposite direction. As reviews per month increase by 1, the price interestingly drops by around 2.8%. The effects on price of other quantitative variables like host listings, availability, and COVID case count, while significant, are minimal, close to 0.

Binary predictors: all interpretations are "on average and holding everything else constant":
Among the "amenities" binary predictors, the availability of a swimming pool has the highest effect, with a 17% price increase. Next highest are the availability of a balcony, elevator, AC, and Cable TV, leading to price increases of 13.5%, 10.8%, 10.4%, and 9.6%, respectively, when present. Interestingly, the availability of free parking, cooking equipment, and bed linens decreases listing price by around 4.48% , 4.15%, and 2.46%, respectively. Per our descriptive statistics, we are confident that this decrease is not related to a less expensive borough or room type. Additionally, if the owner of the Airbnb listing is a superhost, the listing price increases by around 6.7%. That seems logical, since superhosts are regarded as experienced hosts who provide extraordinary experiences for their guests, as defined by Airbnb. And, interestingly enough, if the listing happens to be instant bookable, the price drops by 1.3%.

Categorical predictors: all interpretations are "on average and holding everything else constant":
For the borough variable, we chose the Bronx as our reference level, since it has the cheapest average and median price. Our model output shows that the borough in which the property is located significantly increases the price—by 41.22% , 19.14%, and 9.92% if the property is located in Manhattan, Brooklyn, and Queens, respectively, compared to the Bronx. For Staten Island, compared to Bronx, the small price increase of 0.19% that the output shows is not significantly significant. That means that when comparing a Bronx property to a Staten Island property, the borough doesn't play a significant role in determining listing price. For the room type variable, we chose entire homes as our reference level, since it is the variety most sought after type by Airbnb users. The output demonstrates that room type significantly decreases the price—by 86.19% and 54.50% if the property is a shared room or private room, respectively, compared to an entire

home. For the hotel room type, compared to entire homes, the effect of 4.45% displayed in the output is not statistically significant. That means that, when comparing a hotel room and an entire home, the room type itself doesn't play a role.

8. Conclusions and Lessons Learned

8.1 Conclusions from the Analysis

The most influential predictors of an Airbnb's price in this analysis were features that might be expected intuitively: location (borough, in this case), number of bedrooms, the inclusion of luxurious amenities like a balcony or pool, and the privacy of the location. Some aspects that one might expect would be important, such as the presence of heating or access to a backyard, proved insignificant in our model's determination of pricing. Similarly, we were surprised by the negative impact of features such as free parking and the number of reviews per month on listing price.

We can conclude now, based on our model, that these features matter more in the determination of New York Airbnb pricing than the presence and severity of a global pandemic. COVID cases, hospitalizations, and deaths all proved to have a near-zero impact on price in this analysis. Specifically, while prices or rental rates as a whole might have dropped due to the pandemic, our analysis found that the price of an Airbnb in New York City does not seem to fluctuate as COVID cases rise and fall. We would want to conduct further time series analysis to confirm this result, but unfortunately, we do not have access to the historical and specific, by-day Airbnb pricing that could help us dive deeper into this aspect of the project.

Still, our model would prove useful to New York City Airbnb owners, both for its individual predictive ability and for the information uncovered as a result of this analysis, such as the factors that we have determined should (like bedrooms, balconies, and borough) and should not (like bathrooms and daily COVID cases) play a role in decision-making when deciding on a price. This model was built to help the individual renter effectively price their unit, but it does not necessarily answer the question of what price point would optimize profit based on demand. Again, internal data from Airbnb would be needed for us to determine those factors.

8.2 Project Issues, Challenges, and Lessons Learned: One main challenge the team faced was the lack of a narrow Airbnb booking date. The variable "Last_Review" provided a base estimate of when the booking occurred and, to our benefit, Airbnb policy only allows 14 days after the booking ends to leave a review, mitigating this variable's bias. Additionally, our research did not discover any data sets that already simultaneously possessed both COVID and Airbnb listing data, hence requiring us to join two separate data sets. While we were able to successfully combine the two, direct links between COVID and Airbnb proved difficult with many other dimensions affecting price. Prior to cleaning the data set, all 15 amenities were piled into one column, and utilizing R to transform them into separate binary variables proved challenging and time-consuming. Additionally, we discovered that outlier prices, upwards of $1,000, ran the risk of significantly impacting models, requiring us to filter for bookings of less than $400 late in the project cycle. Keeping the priciest listings would have made the price distribution even more skewed, and price already almost always results in right skewness. Lastly, the computational power needed to test our random forest models was far greater than we expected, requiring several hours to complete each iteration.

An important lesson learned was that regardless of coefficients and relationships extracted, our results contain value. Although our model did not detect a compelling link between COVID-19 cases and Airbnb pricing, the insight that measurements of listing location, size, and quality impact pricing most, even during a pandemic, would be helpful to multiple Airbnb stakeholders. With more time and resources, the team would have further investigated the connection between events during the COVID pandemic and Airbnb pricing, particularly the effect of legislative policies affecting lockdowns. As previously stated, the analysis would have gained additional insight from having specific booking dates and stay lengths. Possible questions worthy of investigation in future projects include whether amenities impact prices differently by city; how climates and seasons impact pricing; and how varying COVID-19 lockdown policies by city influenced listing prices. Lastly, a comparison of hotel and Airbnb metrics is worthy of analysis. Hotel availability and pricing are more stagnant than that of Airbnb, providing an interesting entry point for a comparative look at how COVID affected both industries.

Appendix Contents

Contents

Data Set/Basic Overview of Data Cleaning:

Our primary data set:

```
Airbnb <- read.table("airbnb_dataset_v1 Only NY.csv",
  header = T, sep = ",", stringsAsFactors = T
)
```

Code to organize the dates in the data set:

```
a <- as.Date(Airbnb$last_review, format = "%m/%d/%Y") # Produces NA when format
is not "%m/%d/%Y"
b <- as.Date(Airbnb$last_review, format = "%d-%m-%Y") # Produces NA when format
is not "%d-%m-%Y"

a[is.na(a)] <- b[!is.na(b)] # Combine both while keeping their ranks
Airbnb$last_review <- a # Put it back in the dataframe
```

A brief sample of binary variable transformations:

```
Airbnb <- Airbnb %>%
```

```
  mutate(host_is_superhost = ifelse(host_is_superhost == "t", 1, 0)) %>%
  mutate(host_identity_verified = ifelse(host_identity_verified == "t", 1, 0))
```

A brief sample of string detection and binary transformations within the "amenities" column:

```
%>%
  mutate(instant_bookable = ifelse(instant_bookable == "t", 1, 0)) %>%
  mutate(
    AC = str_detect(amenities, "Air conditioning"),
      %>%
  mutate(
    AC = ifelse(AC == "TRUE", 1, 0),
```

How nyccovid was joined to the Airbnb data set by date:

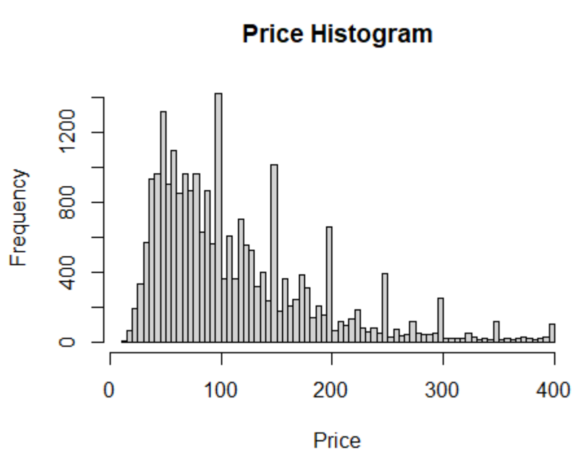
```
nyccovid <- read_csv("nyccovid.csv", col_types = cols(date_of_interest = col_da
te(format = "%m/%d/%Y")))

Airbnb_covid <- left_join(Airbnb, nyccovid, by = c("last_review" = "date_of_int
erest"))

Airbnb <- Airbnb_covid %>%
  mutate(
    CASE_COUNT = replace_na(CASE_COUNT, 0),
    HOSPITALIZED_COUNT = replace_na(HOSPITALIZED_COUNT, 0),
    DEATH_COUNT = replace_na(DEATH_COUNT, 0)
  )
```

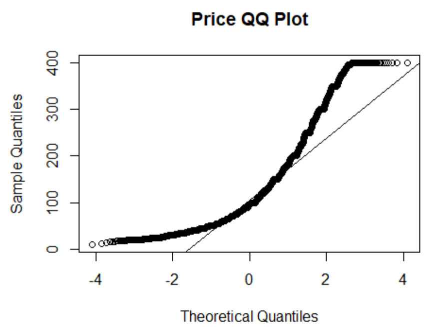
Removing outliers from the response variable:

```
Airbnb <- Airbnb %>%
  filter(price <= 400)
```

Descriptive Statistics

A histogram and QQ plot of our response variable (price, shown below) demonstrate right skewness:



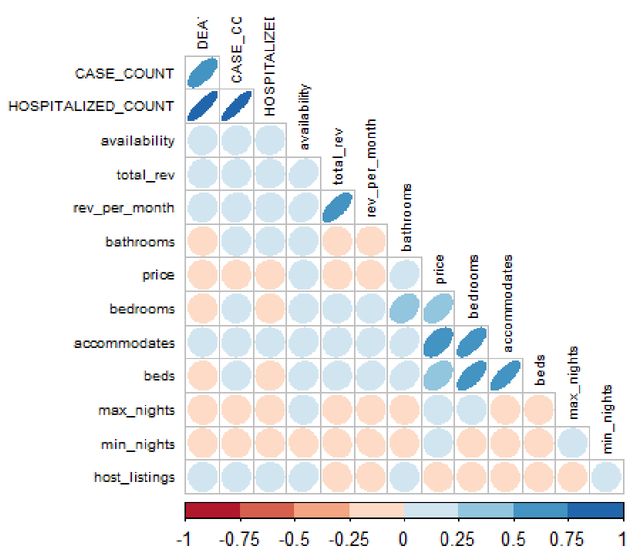


A quick overview of listing counts by room type and borough:

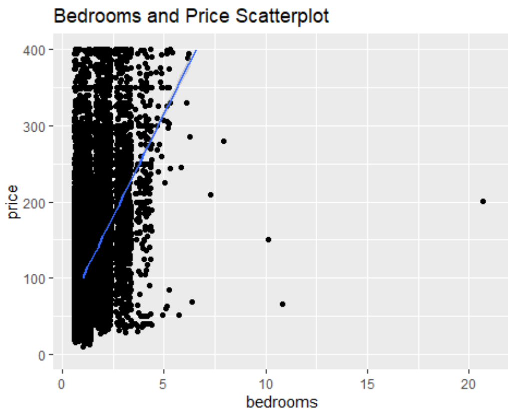
```
##    room_type       borough       count
##    <fct>           <fct>         <int>
##  1 Entire home/apt Bronx           233
##  2 Entire home/apt Brooklyn       5150
##  3 Entire home/apt Manhattan      5169
##  4 Entire home/apt Queens         1150
##  5 Entire home/apt Staten Island   110
##  6 Hotel room      Brooklyn         11
##  7 Hotel room      Manhattan       140
##  8 Hotel room      Queens            9
##  9 Private room    Bronx           431
## 10 Private room    Brooklyn       5008
## 11 Private room    Manhattan      4047
## 12 Private room    Queens         1998
## 13 Private room    Staten Island   103
## 14 Shared room     Bronx            20
## 15 Shared room     Brooklyn        153
## 16 Shared room     Manhattan       205
## 17 Shared room     Queens           90
## 18 Shared room     Staten Island     2
```

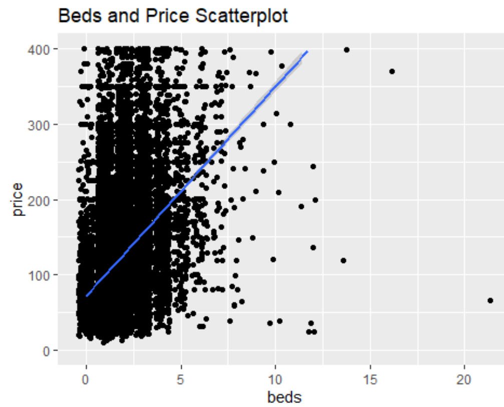
Correlations

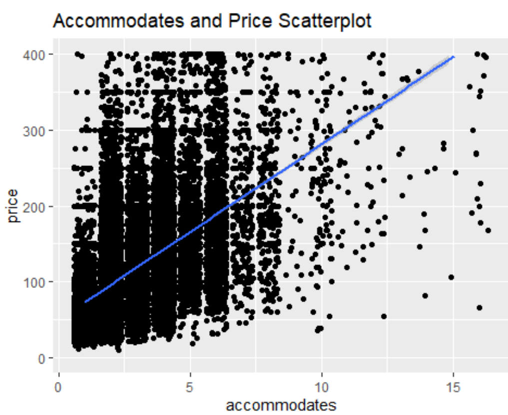
Below, a corrplot illustrates the correlations in our initial data set:



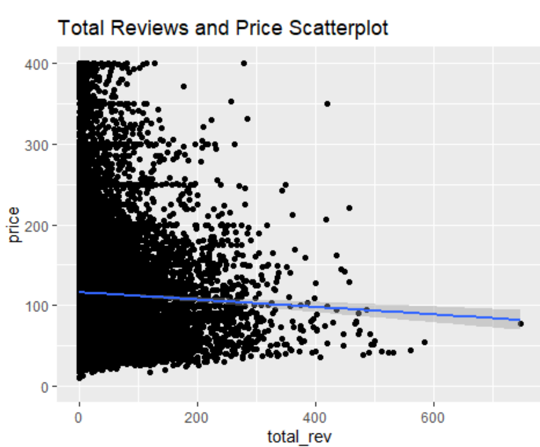
From 5.1-5.2: Bedrooms, beds, accommodates vs. price (strongest positive correlations with price)

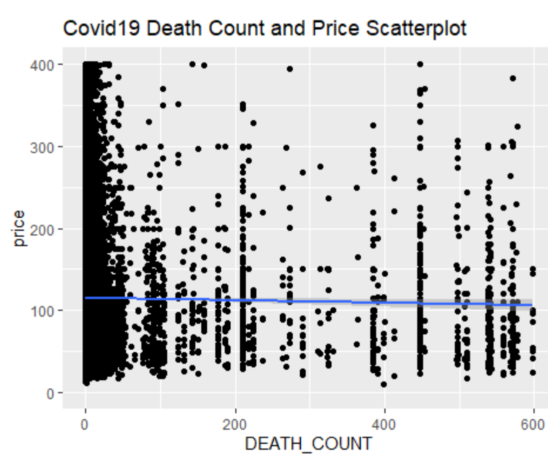






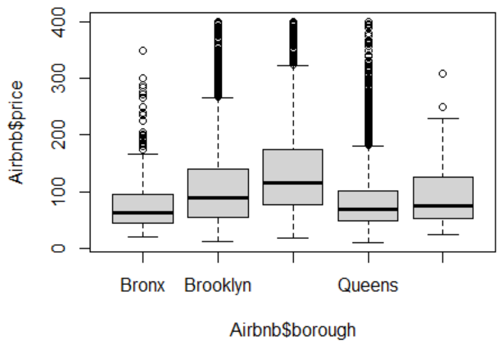
Other Plots Mentioned in 5.1-5.2: Total Reviews vs. Price, COVID Deaths vs. Price:





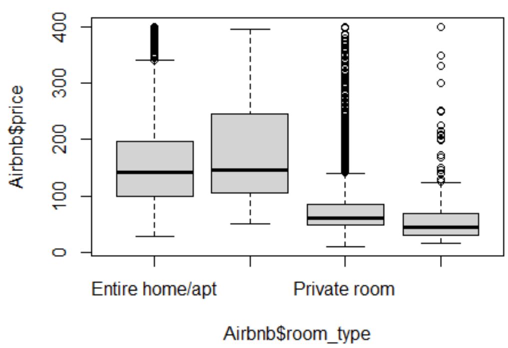
ANOVA

Price by Borough (Mentioned in 5.1-5.2)



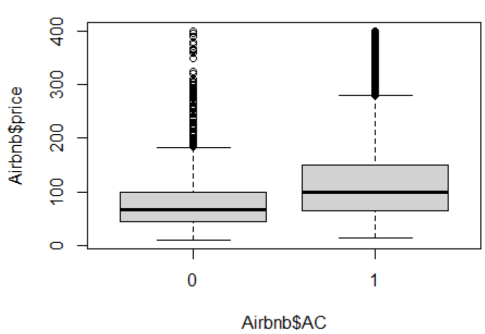
p-value: *<2e-16* ***

Price by Room Type (Mentioned in 5.1-5.2)



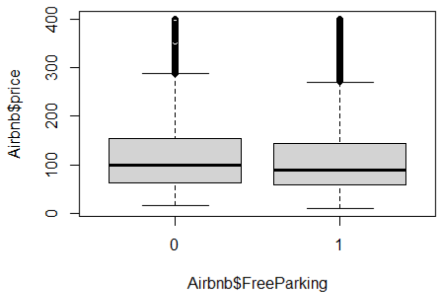
p-value: <2e-16***

Price by presence of AC (Mentioned in 5.1-5.2)



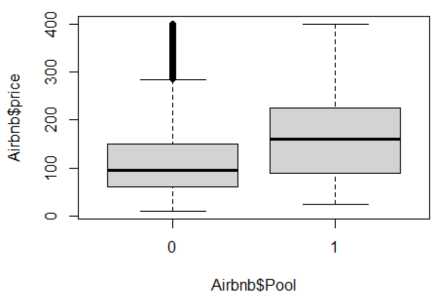
p-value: <2e-16 ***

Price by presence of Free Parking (Mentioned in 5.1-5.2)



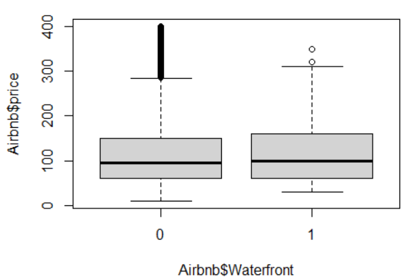
p-value: <2e-16***

Price by presence of Pool (Mentioned in 5.1-5.2)



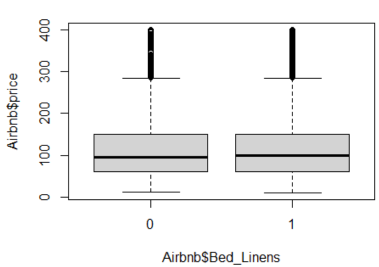
p-value: <2e-16***

Price by presence of Waterfront (Mentioned in 5.1-5.2)



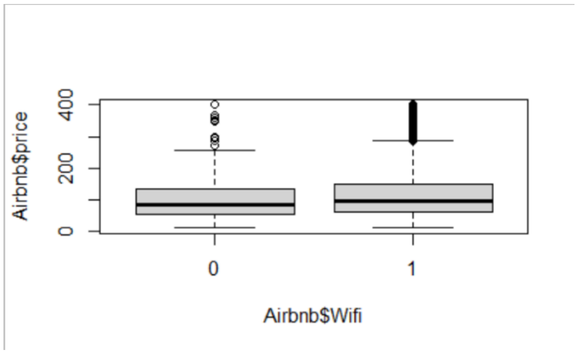
p-value: 0.479

Price by presence of Linens (Mentioned in 5.1-5.2)



p-value: 0.126

Price by presence of Wifi (Mentioned in 5.1 – 5.2)



p-value: 0.00173

Chi Square Test: Borough vs. Room Type

Observed Table: Borough vs. Room Type:

```
##                  room_type
## borough           Entire home/apt Hotel room Private room Shared room
##   Bronx                       233          0          431          20
##   Brooklyn                   5150         11         5008         153
##   Manhattan                  5169        140         4047         205
##   Queens                     1150          9         1998          90
##   Staten Island               110          0          103           2
```

Expected Table: Borough vs. Room Type:

```
##                  room_type
## borough           Entire home/apt Hotel room Private room Shared room
##   Bronx                  336.2357   4.554497     329.8310   13.378834
##   Brooklyn              5074.0132  68.730284    4977.3613  201.895210
##   Manhattan             4699.9264  63.663074    4610.4002  187.010279
##   Queens                1596.1365  21.620542    1565.7326   63.510342
##   Staten Island          105.6881   1.431603     103.6749    4.205335
```

The results of the Chi-squared test, clearly demonstrating dependence (Mentioned in 5.1-5.2):

```
##  Pearson's Chi-squared test
##
## data:  cross.table
## X-squared = 606.37, df = 12, p-value < 2.2e-16
```

Fitting the OLS Regression model (initial set of predictors)

Initial predictors selected using business rationale:

```
lm.fit.airbnb <- lm(price ~ accommodates + AC + CableTV + Wifi + FreeParking +
Pool + Garden_Backyard + Heating + Elevator + Patio_Balcony + Pvt_Entrance + Co
oking_Equip + Coffee_Machine + borough + room_type + bathrooms + bedrooms + min
_nights + availability + CASE_COUNT + DEATH_COUNT, data = Airbnb)
```

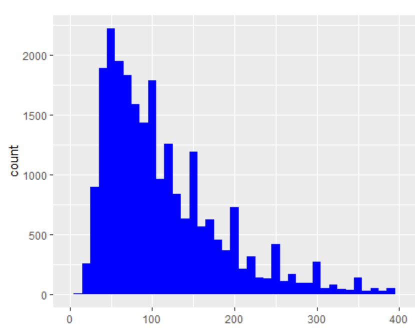
Coefficients for model based on initial set of predictors:

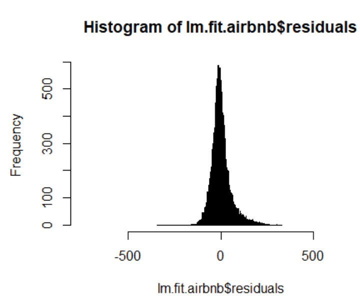
```
##
## Coefficients:
```

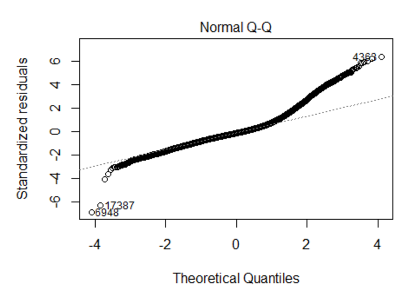
```
##                          Estimate  Std. Error t value Pr(>|t|)
## (Intercept)             43.7646810  3.7426851  11.693  < 2e-16 ***
## accommodates             9.2208018  0.3076959  29.967  < 2e-16 ***
## AC                       7.8639696  1.0457226   7.520 5.66e-14 ***
## CableTV                 10.7564247  0.8216070  13.092  < 2e-16 ***
## Wifi                     0.4867926  2.7305803   0.178 0.858509
## FreeParking             -6.6025793  0.8163576  -8.088 6.36e-16 ***
## Pool                    26.1250577  3.4219685   7.635 2.35e-14 ***
## Garden_Backyard          0.9703631  1.2486302   0.777 0.437083
## Heating                  1.1725162  1.6290250   0.720 0.471676
## Elevator                13.8939426  0.8613262  16.131  < 2e-16 ***
## Patio_Balcony           17.0710258  1.2087043  14.123  < 2e-16 ***
## Pvt_Entrance            -0.2433831  0.8846441  -0.275 0.783227
## Cooking_Equip           -5.7368215  0.9167041  -6.258 3.96e-10 ***
## Coffee_Machine          -0.0355508  0.9244034  -0.038 0.969323
## boroughBrooklyn         17.7916372  2.0772211   8.565  < 2e-16 ***
## boroughManhattan        41.8904047  2.1065359  19.886  < 2e-16 ***
## boroughQueens            7.7838370  2.1996742   3.539 0.000403 ***
## boroughStaten Island    -5.5788880  4.0953155  -1.362 0.173128
## room_typeHotel room      5.3836400  4.2555428   1.265 0.205852
## room_typePrivate room  -53.4920611  0.8486413 -63.033  < 2e-16 ***
## room_typeShared room   -69.4037057  2.5370959 -27.356  < 2e-16 ***
## bathrooms               11.0957268  0.9166434  12.105  < 2e-16 ***
## bedrooms                19.0199825  0.7847963  24.236  < 2e-16 ***
## min_nights              -0.0075906  0.0136336  -0.557 0.577700
## availability             0.0042640  0.0024492   1.741 0.081698 .
## CASE_COUNT              -0.0001637  0.0004724  -0.347 0.728925
## DEATH_COUNT             -0.0075688  0.0061036  -1.240 0.214972
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.15 on 24002 degrees of freedom
## Multiple R-squared:  0.497,  Adjusted R-squared:  0.4965
## F-statistic: 912.2 on 26 and 24002 DF,  p-value: < 2.2e-16
```

OLS Assumption Tests (initial set of predictors)

1. YC: Price is not a discrete or count data, but because it is truncated as zero, it falls slightly short of the true definition of continuous. Thus, we must log the response variable in order to be able to use OLS in our model analysis.







2. EN:

```
See above:
```

3. XI (Predictors are Independent):

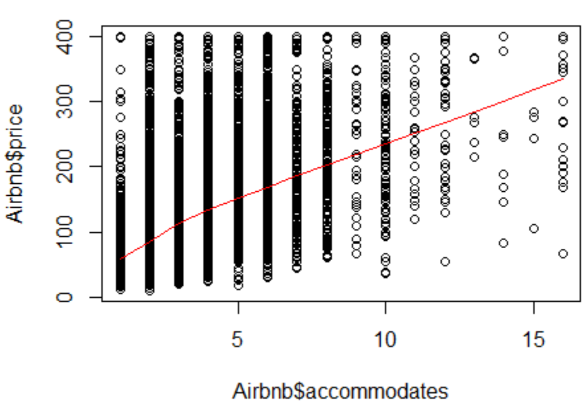
Condition Index:

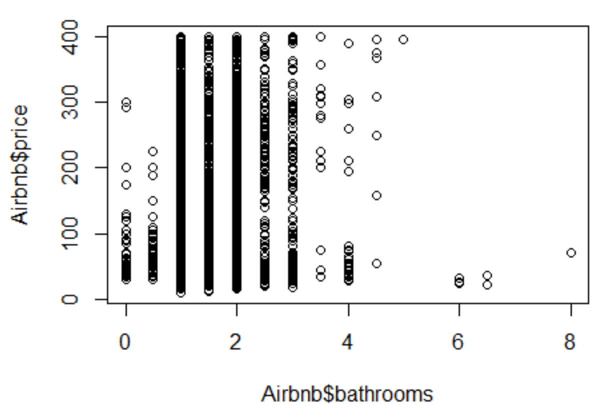
```
##  [1]  1.000000  2.656201  2.839832  3.096939  3.354670  3.399502  3.416577
##  [8]  3.481278  3.607357  3.755567  4.051759  4.201556  4.375926  4.639074
## [15]  4.763325  4.990666  5.191031  6.373712  6.690868  7.629467  8.423555
## [22] 11.061063 12.082130 13.763073 17.420428 24.314025 44.031163
```

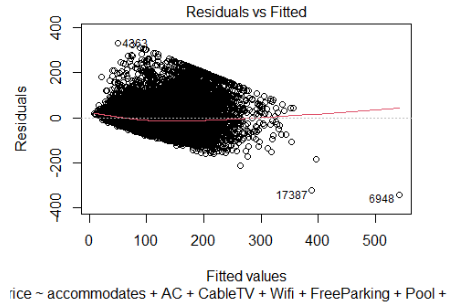
VIF and GVIF Values:

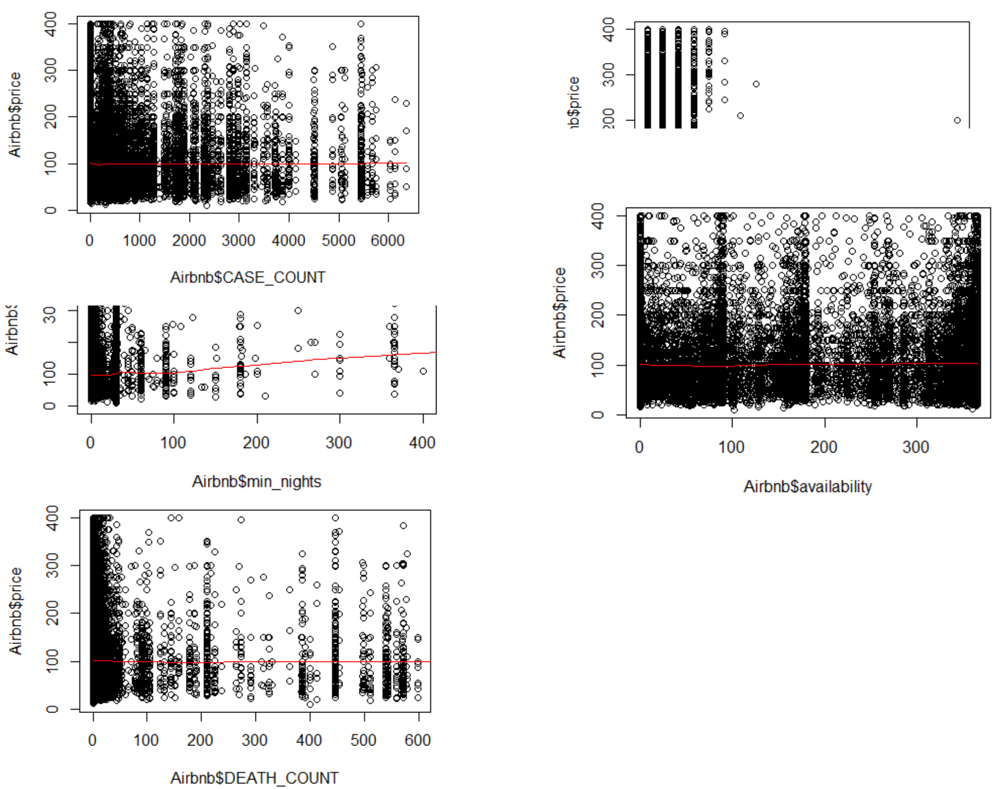
```
##                     GVIF Df GVIF^(1/(2*Df))
## accommodates    2.429789  1        1.558778
## AC              1.102285  1        1.049898
## CableTV         1.091190  1        1.044600
## Wifi            1.041211  1        1.020397
## FreeParking     1.471999  1        1.213260
## Pool            1.023006  1        1.011438
## Garden_Backyard 1.265819  1        1.125086
## Heating         1.074563  1        1.036611
## Elevator        1.134138  1        1.064959
## Patio_Balcony   1.222735  1        1.105774
## Pvt_Entrance    1.213480  1        1.101581
## Cooking_Equip   1.850142  1        1.360199
## Coffee_Machine  1.830735  1        1.353047
## borough         1.282503  4        1.031590
## room_type       1.658078  3        1.087930
## bathrooms       1.157699  1        1.075964
## bedrooms        2.046145  1        1.430435
## min_nights      1.040405  1        1.020002
## availability    1.120351  1        1.058467
## CASE_COUNT      1.950323  1        1.396540
## DEATH_COUNT     1.938775  1        1.392399
```

4. LI (Y and Xs have a linear relationship):







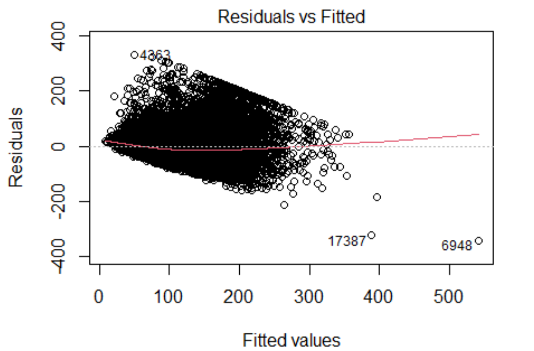


5 and 6: Observations are independent (OI) and errors are independent (EI): Explained in 6.2

7. EA: Error averages equal 0:

```
mean(lm.fit.airbnb$residuals)

## [1] -9.874772e-16
```

8. EV: the error variance is constant:




```
library(lmtest)
bptest(lm.fit.airbnb, data = Airbnb)

##
##  studentized Breusch-Pagan test
##
## data:  lm.fit.airbnb
## BP = 2243.4, df = 26, p-value < 2.2e-16
```

Variable Selection for Second Specification:

Brief snapshot of the setup for stepwise variable selection:

```
CASE_COUNT is included to preserve analysis of COVID's impact
airbnb.low <- lm(price ~ CASE_COUNT, data = Airbnb)
airbnb.high <- lm(price ~ . -last_review, data = Airbnb)

#We will be very strict and include a p-value threshold of 0.01
qchisq(0.01, 1, lower.tail=F)

## [1] 6.634897

kval <- qchisq(0.01, 1, lower.tail=F)

#To run stepwise variable selection
Airbnb.step.backward <- step(airbnb.high, scope = list(lower = airbnb.low, uppe
r = airbnb.high), direction = "both", test = "F", k = kval)

summary(Airbnb.step.backward)


## Coefficients:
##                        Estimate  Std. Error t value Pr(>|t|)
## (Intercept)          46.7220187   2.5557527  18.281  < 2e-16 ***
## superhost             7.5341592   0.8569971   8.791  < 2e-16 ***
## instant_bookable     -2.6521949   0.7578475  -3.500 0.000467 ***
## accommodates          9.6812777   0.3056385  31.676  < 2e-16 ***
## boroughBrooklyn      17.0235777   2.0564587   8.278  < 2e-16 ***
## boroughManhattan     42.6448898   2.0817963  20.485  < 2e-16 ***
```

```
## boroughQueens            9.3340361   2.1800883   4.281 1.86e-05 ***
## boroughStaten Island   -6.7083550   4.0527836  -1.655 0.097888 .
## room_typeHotel room    17.9028854   4.2673236   4.195 2.73e-05 ***
## room_typePrivate room -52.4139005   0.8337303 -62.867  < 2e-16 ***
## room_typeShared room  -68.6244581   2.5010412 -27.438  < 2e-16 ***
## bathrooms              11.2025188   0.9078179  12.340  < 2e-16 ***
## bedrooms               18.2088168   0.7791874  23.369  < 2e-16 ***
## rev_per_month          -3.8529351   0.2835138 -13.590  < 2e-16 ***
## host_listings          -0.2461823   0.0192379 -12.797  < 2e-16 ***
## availability            0.0118872   0.0024598   4.833 1.36e-06 ***
## AC                      8.0395814   1.0200206   7.882 3.36e-15 ***
## CableTV                11.2440984   0.8177829  13.749  < 2e-16 ***
## FreeParking            -5.7881694   0.8107209  -7.140 9.63e-13 ***
## Pool                   27.5617144   3.3912569   8.127 4.60e-16 ***
## Elevator               12.9265267   0.8526834  15.160  < 2e-16 ***
## Patio_Balcony          17.4918809   1.1500883  15.209  < 2e-16 ***
## Cooking_Equip          -3.8733809   0.8455036  -4.581 4.65e-06 ***
## Bed_Linens             -3.8827108   0.8208765  -4.730 2.26e-06 ***
## CASE_COUNT             -0.0002977   0.0003376  -0.882 0.377830
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.66 on 24004 degrees of freedom
## Multiple R-squared:  0.5063, Adjusted R-squared:  0.5059
## F-statistic:  1026 on 24 and 24004 DF,  p-value: < 2.2e-16
```

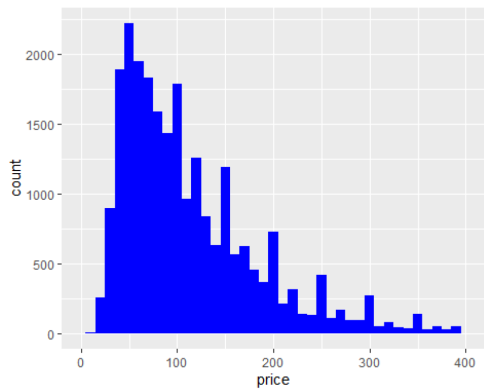
OLS Testing for smaller set of predictors (second specification):

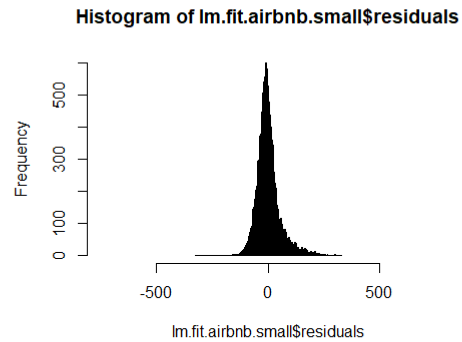
Fitting the model for the smaller specification:

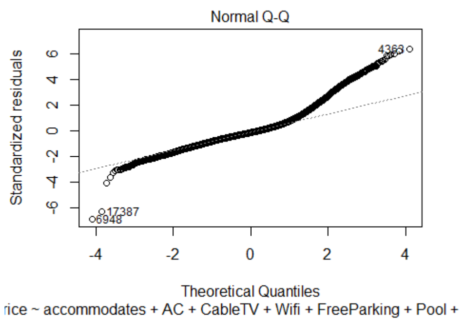
```
#To fit the OLS regression model using this smaller set of predictors:
lm.fit.airbnb.small <- lm(price ~ superhost + instant_bookable + accommodates +
borough + room_type + bathrooms + bedrooms + rev_per_month + host_listings + av
ailability + AC + CableTV + FreeParking + Pool + Elevator + Patio_Balcony + Coo
king_Equip + Bed_Linens + CASE_COUNT, data = Airbnb)

#To display the summary for the smaller lm.fit.airbnb.small model:
```

1) YC: Explained in 6.3a

2) EN:







3) XI Predictors are Independent:

Condition Index:

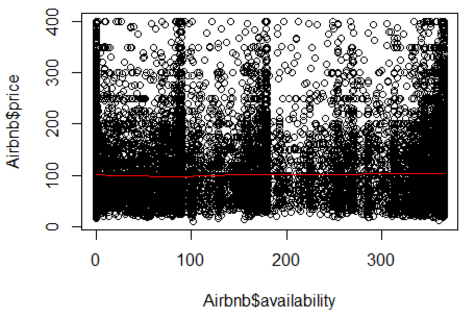
```
##  [1]  1.000000  2.616069  2.962790  3.005820  3.102178  3.161887  3.224827
##  [8]  3.285720  3.372562  3.540591  3.576536  3.673065  3.941477  4.100351
## [15]  4.308931  4.405333  4.589329  4.815291  5.573261  6.401076  8.070352
## [22] 10.389931 12.127205 12.768923 29.339260
```

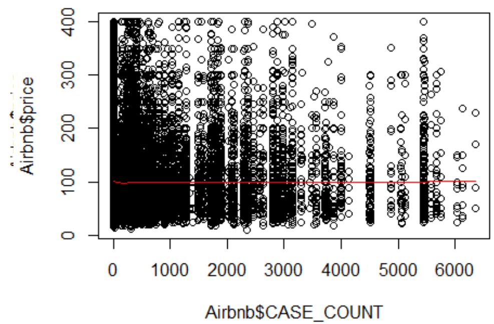
VIFs and GVIFs:

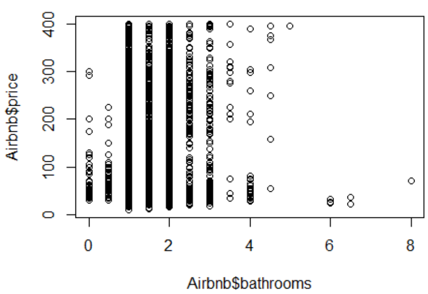
```
##                     GVIF Df GVIF^(1/(2*Df))
## superhost       1.209346  1        1.099703
## instant_bookable 1.052240  1        1.025787
## accommodates    2.442949  1        1.562994
## borough         1.272787  4        1.030610
## room_type       1.660790  3        1.088226
## bathrooms       1.157086  1        1.075680
## bedrooms        2.055322  1        1.433639
## rev_per_month   1.256795  1        1.121069
## host_listings   1.071202  1        1.034989
## availability    1.151533  1        1.073095
## AC              1.068691  1        1.033775
## CableTV         1.101594  1        1.049568
## FreeParking     1.479322  1        1.216274
## Pool            1.023814  1        1.011837
## Elevator        1.132607  1        1.064240
```

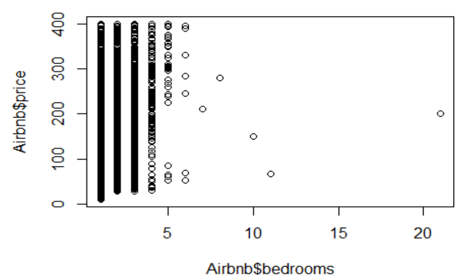
```
## Patio_Balcony     1.128049  1         1.062097
## Cooking_Equip     1.603803  1         1.266413
## Bed_Linens        1.441812  1         1.200755
## CASE_COUNT        1.015016  1         1.007480
```

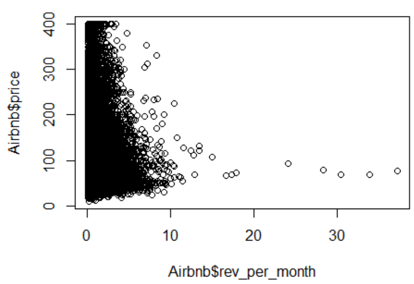
4. LI (Y and Xs have a linear relationship):

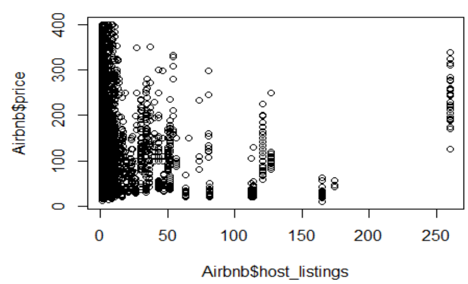












5 and 6: Observations are independent (OI) and errors are independent (EI): Explained in 6.3a

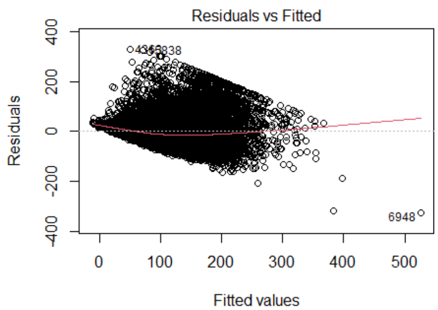
7. EA: Error averages equal 0:

```r
mean(lm.fit.airbnb.small$residuals)
```

```
## [1] 1.394899e-15
```

8. EV: the error variance is constant:

```
#To examine the first residual plot:
plot(lm.fit.airbnb.small, which = 1)
```




```
library(lmtest)
#To run the Breusch Pagan test
bptest(lm.fit.airbnb.small, data = Airbnb)

##
##  studentized Breusch-Pagan test
##
## data:  lm.fit.airbnb.small
## BP = 2186.4, df = 24, p-value < 2.2e-16
```

Weighted Least Squares (Weighting and Reweighting)

Quick demonstration of weighting and reweighting:

```
abs.res.airbnb <- abs(residuals(lm.fit.airbnb))
fitted.airbnb <- fitted(lm.fit.airbnb)
lm.abs.res.airbnb <- lm(abs.res.airbnb ~ fitted.airbnb)
wts.airbnb <- 1/fitted(lm.abs.res.airbnb)^2

lm.wls.airbnb <- lm(log(price) ~ accommodates + AC + CableTV + Wifi + FreeParki
ng + Pool + Garden_Backyard + Heating + Elevator + Patio_Balcony + Pvt_Entrance
+ Cooking_Equip + Coffee_Machine + borough + room_type + bathrooms + bedrooms +
min_nights + availability + CASE_COUNT + DEATH_COUNT, weights = wts.airbnb, dat
a = Airbnb)

bptest(lm.wls.airbnb)

##
##  studentized Breusch-Pagan test
##
## data:  lm.wls.airbnb
## BP = 599.82, df = 26, p-value < 2.2e-16
```

Reweighting:

```
abs.res.wls <- abs(residuals(lm.wls.airbnb))
fitted.airbnb.wls <- fitted(lm.wls.airbnb)
wls.abs.res.airbnb <- lm(abs.res.wls ~ fitted.airbnb.wls)
wts.airbnb.wls <- 1/fitted(wls.abs.res.airbnb)^2
```

We then repeated the same step for the small set of predictors:

```
lm.fit.airbnb.small <- lm(log(price) ~ superhost + accommodates + borough + roo
m_type + bathrooms + bedrooms + rev_per_month + host_listings + availability +
CableTV + FreeParking + Pool + Elevator + Patio_Balcony + Bed_Linens + CASE_COU
NT + AC + instant_bookable + Cooking_Equip, data = Airbnb)

abs.res.airbnb.small <- abs(residuals(lm.fit.airbnb.small))
fitted.airbnb.small <- fitted(lm.fit.airbnb.small)
lm.abs.res.airbnb.small <- lm(abs.res.airbnb.small ~ fitted.airbnb.small)
wts.airbnb.small <- 1/fitted(lm.abs.res.airbnb.small)^2

lm.wls.airbnb.small <- lm(log(price) ~ superhost + accommodates + borough + roo
m_type + bathrooms + bedrooms + rev_per_month + host_listings + availability +
CableTV + FreeParking + Pool + Elevator + Patio_Balcony + Bed_Linens + CASE_COU
NT + AC + instant_bookable + Cooking_Equip, weights = wts.airbnb.small, data =
Airbnb)
```

Reweighting:

```
abs.res.wls.small <- abs(residuals(lm.wls.airbnb.small))
fitted.airbnb.wls.small <- fitted(lm.wls.airbnb.small)
wls.abs.res.airbnb.small <- lm(abs.res.wls.small ~ fitted.airbnb.wls.small)
wts.airbnb.wls.small <- 1/fitted(wls.abs.res.airbnb.small)^2
```

WLS Models, 10FCV Testing

Output and Results, Initial WLS Model:

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat, weights = wts)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9775 -0.8401 -0.0305  0.8154  6.2804
##
## Coefficients:
##                            Estimate   Std. Error t value Pr(>|t|)
## (Intercept)             4.135829679  0.028983677 142.695  < 2e-16 ***
## accommodates            0.073479024  0.002409564  30.495  < 2e-16 ***
## AC                      0.108900119  0.008078072  13.481  < 2e-16 ***
## CableTV                 0.091026474  0.006398143  14.227  < 2e-16 ***
## Wifi                   -0.007798099  0.021162504  -0.368  0.71251
## FreeParking            -0.051210866  0.006337522  -8.081 6.75e-16 ***
## Pool                    0.153915060  0.026777169   5.748 9.14e-09 ***
## Garden_Backyard         0.016850011  0.009707647   1.736  0.08262 .
## Heating                -0.000143614  0.012609079  -0.011  0.99091
## Elevator                0.116594554  0.006703645  17.393  < 2e-16 ***
## Patio_Balcony           0.128144620  0.009411179  13.616  < 2e-16 ***
## Pvt_Entrance            0.020728281  0.006884353   3.011  0.00261 **
## Cooking_Equip          -0.064447490  0.007111307  -9.063  < 2e-16 ***
## Coffee_Machine          0.019724895  0.007173472   2.750  0.00597 **
## boroughBrooklyn         0.198533995  0.016018791  12.394  < 2e-16 ***
## boroughManhattan        0.402802020  0.016251568  24.785  < 2e-16 ***
## boroughQueens           0.077499521  0.016961091   4.569 4.92e-06 ***
## `boroughStaten Island`  0.007227810  0.031659936   0.228  0.81942
## `room_typeHotel room`  -0.049161713  0.033364840  -1.473  0.14064
```

```
## `room_typePrivate room` -0.548470125  0.006599331 -83.110  < 2e-16 ***
## `room_typeShared room`  -0.857700581  0.019461565 -44.072  < 2e-16 ***
## bathrooms               -0.003336335  0.007115459  -0.469  0.63916
## bedrooms                 0.109988336  0.006164560  17.842  < 2e-16 ***
## min_nights              -0.000202040  0.000106119  -1.904  0.05693 .
## availability             0.000035670  0.000019020   1.875  0.06075 .
## CASE_COUNT              -0.000002846  0.000003668  -0.776  0.43789
## DEATH_COUNT             -0.000100864  0.000047347  -2.130  0.03316 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.274 on 24002 degrees of freedom
## Multiple R-squared:  0.5638, Adjusted R-squared:  0.5634
## F-statistic:  1193 on 26 and 24002 DF,  p-value: < 2.2e-16
```

MSE:

```
## [1] 0.1643234
```

Output and Testing, Small WLS Model:

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat, weights = wts)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9613 -0.8474 -0.0450  0.8039  6.5210
##
## Coefficients:
##                             Estimate   Std. Error t value Pr(>|t|)
## (Intercept)              4.143797479  0.019550453 211.954  < 2e-16 ***
## superhost                0.067161046  0.006593973  10.185  < 2e-16 ***
## accommodates             0.077240422  0.002381841  32.429  < 2e-16 ***
## boroughBrooklyn          0.191412908  0.015648713  12.232  < 2e-16 ***
## boroughManhattan         0.412285661  0.015852122  26.008  < 2e-16 ***
## boroughQueens            0.099211684  0.016588069   5.981 2.25e-09 ***
## `boroughStaten Island`   0.001977316  0.030958036   0.064   0.9491
## `room_typeHotel room`    0.044493654  0.033313197   1.336   0.1817
## `room_typePrivate room` -0.545371459  0.006427665 -84.848  < 2e-16 ***
## `room_typeShared room`  -0.861912684  0.018871704 -45.672  < 2e-16 ***
## bathrooms               -0.001417206  0.006977609  -0.203   0.8391
## bedrooms                 0.104384067  0.006099265  17.114  < 2e-16 ***
## rev_per_month           -0.027628257  0.002170159 -12.731  < 2e-16 ***
## host_listings           -0.003799437  0.000145495 -26.114  < 2e-16 ***
## availability             0.000132719  0.000018921   7.014 2.37e-12 ***
## CableTV                  0.096476866  0.006318174  15.270  < 2e-16 ***
## FreeParking             -0.044805546  0.006234183  -7.187 6.81e-13 ***
## Pool                     0.177359293  0.026398119   6.719 1.88e-11 ***
## Elevator                 0.108041792  0.006582429  16.414  < 2e-16 ***
## Patio_Balcony            0.135236661  0.008883782  15.223  < 2e-16 ***
## Bed_Linens              -0.024616680  0.006313801  -3.899 9.69e-05 ***
## CASE_COUNT              -0.000005348  0.000002596  -2.060   0.0394 *
## AC                       0.104218876  0.007784406  13.388  < 2e-16 ***
```

```
## instant_bookable        -0.012945393  0.005822276  -2.223   0.0262 *
## Cooking_Equip           -0.041473391  0.006498378  -6.382 1.78e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.28 on 24004 degrees of freedom
## Multiple R-squared:  0.5801, Adjusted R-squared:  0.5797
## F-statistic:  1382 on 24 and 24004 DF,  p-value: < 2.2e-16
```

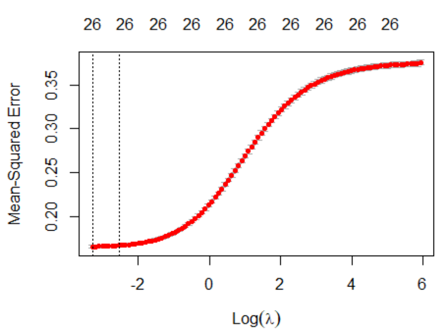
MSE:

```
## [1] 0.1582183
```

Ridge Regression Tuning and 10FCV Results:

Testing: Initial specification, unweighted ridge model:

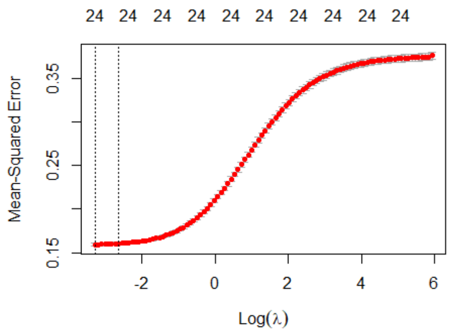
```
library(glmnet)
x <- model.matrix(price ~ accommodates + AC + CableTV + Wifi + FreeParking + Po
ol + Garden_Backyard + Heating + Elevator + Patio_Balcony + Pvt_Entrance + Cook
ing_Equip + Coffee_Machine + borough + room_type + bathrooms + bedrooms + min_n
ights + availability + CASE_COUNT + DEATH_COUNT, weights = wts.airbnb, data = A
irbnb)[,-1]
y <- log(Airbnb$price)


##      Best Lambda Best 10FCV
## [1,]  0.03752173  0.1651945

plot(ridge.b.n)
```



Testing: Small specification, unweighted ridge model

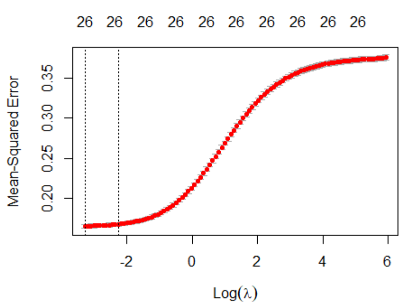
```
x2 <- model.matrix(price ~ superhost + accommodates + borough + room_type + bat
hrooms + bedrooms + rev_per_month + host_listings + availability + CableTV + Fr
eeParking + Pool + Elevator + Patio_Balcony + Bed_Linens + CASE_COUNT + AC + in
stant_bookable + Cooking_Equip, data = Airbnb)[,-1]
y2 <- log(Airbnb$price)

cbind("Best Lambda"= lamda.s.n, "Best 10FCV" = mincv.s.n)
```

```
##       Best Lambda Best 10FCV
## [1,]  0.03752173  0.1591904

plot(ridge.s.n)
```



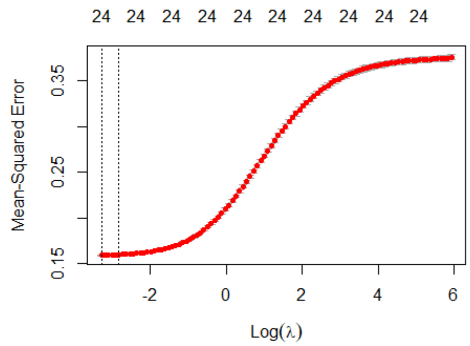
Testing: Initial specification, weighted ridge model

```
x3 <- model.matrix(price ~ accommodates + AC + CableTV + Wifi + FreeParking + P
ool + Garden_Backyard + Heating + Elevator + Patio_Balcony + Pvt_Entrance + Coo
king_Equip + Coffee_Machine + borough + room_type + bathrooms + bedrooms + min_
nights + availability + CASE_COUNT + DEATH_COUNT, weights = wts.airbnb, weights
= wts.airbnb.wls ,data = Airbnb)[,-1]
y3 <- log(Airbnb$price)

cbind("Best Lambda"= lamda.b.y , "Best 10FCV" = mincv.b.y )

##       Best Lambda Best 10FCV
## [1,]  0.03752173  0.1651503
```

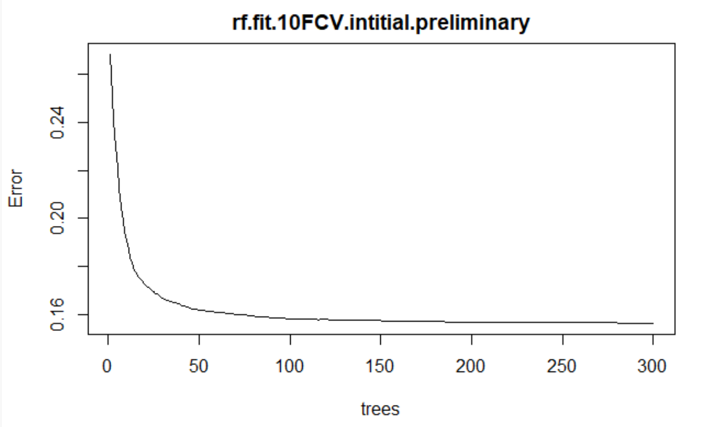


Testing: Small specification, weighted ridge model:

```
x4 <- model.matrix(price ~ superhost + accommodates + borough + room_type + bat
hrooms + bedrooms + rev_per_month + host_listings + availability + CableTV + Fr
eeParking + Pool + Elevator + Patio_Balcony + Bed_Linens + CASE_COUNT + AC + in
stant_bookable + Cooking_Equip,  weights= wts.airbnb.wls.small,  data = Airbnb)
[,-1]
y4 <- log(Airbnb$price)

cbind("Best Lambda"= lamda.s.y, "Best 10FCV" = mincv.s.y)

##       Best Lambda Best 10FCV
## [1,]  0.03752173  0.1591717
```



Random Forest 10FCV Testing



Demonstration that MSE flattens around 100 trees:

Note: due to intense computing requirements, we have not knitted our Random Forest results and are instead including a printout below:

Random Forest Initial Specification:

24029 samples
21 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 21626, 21627, 21626, 21626, 21627, 21626, ...
Resampling results across tuning parameters:

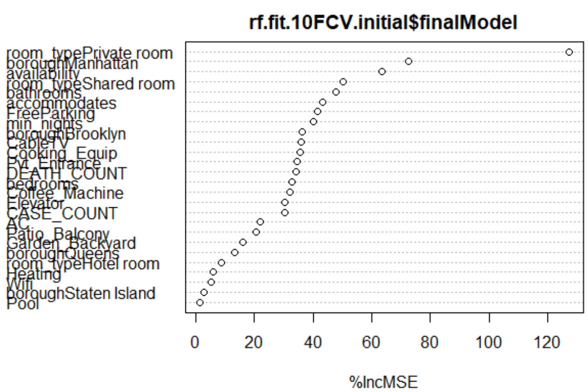
mtry	RMSE	Rsquared	MAE
2	0.4092091	0.5889211	0.3250095
14	0.3903765	0.5953336	0.3048054
26	0.3949214	0.5868556	0.3080530

RMSE was used to select the optimal model using the smallest value.

The final value used for the model was mtry = 14.

	%IncMSE	IncNodePurity
accommodates	43.404093	1102.795124
AC	22.115288	95.032959

CableTV	35.776367	126.565519
Wifi	5.185664	34.663574
FreeParking	41.577509	126.648367
Pool	1.431771	21.258479
Garden_Backyard	16.022644	61.407878
Heating	5.906738	52.196474
Elevator	30.479005	128.963909
Patio_Balcony	20.529940	76.185597
Pvt_Entrance	34.652319	109.238623
Cooking_Equip	35.408503	116.219448
Coffee_Machine	32.240097	110.087294
boroughBrooklyn	36.342894	77.729516
boroughManhattan	72.531369	406.932044
boroughQueens	13.085133	54.655083
boroughStaten Island	2.621566	8.434982
room_typeHotel room	8.554801	8.324571
room_typePrivate room	127.316122	2611.811123
room_typeShared room	50.206192	389.896715
bathrooms	47.762396	287.636432
bedrooms	32.759833	466.182032
min_nights	40.010425	323.242824
availability	63.592215	684.896767
CASE_COUNT	30.478050	350.446747
DEATH_COUNT	34.077705	282.192429



% Var explained: 59.16, MSE = 0.1524

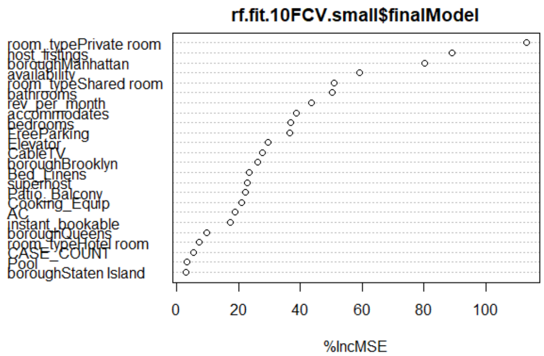
Random Forest Small Specification:

24029 samples
19 predictors

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 21627, 21625, 21626, 21626, 21627, 21626, ...
Resampling results across tuning parameters:

mtry	RMSE	Rsquared	MAE
2	0.3976157	0.6132250	0.3158090
13	0.3759029	0.6242913	0.2912379
24	0.3807247	0.6152005	0.2946226

The final value used for the model was mtry = 13.



	%IncMSE	IncNodePurity
superhost	22.982706	83.312245
accommodates	38.604441	1167.476373
boroughBrooklyn	26.290767	65.033970
boroughManhattan	80.427783	417.518998
boroughQueens	9.861623	44.404602
boroughStaten Island	3.080405	7.689080
room_typeHotel room	7.239547	6.166065
room_typePrivate room	113.190093	2455.364063
room_typeShared room	50.819138	361.951527
bathrooms	50.341477	259.319456
bedrooms	36.865479	555.512649
rev_per_month	43.496074	857.717358
host_listings	89.185989	499.885642
availability	59.115226	526.673386
CableTV	27.713127	109.361700
FreeParking	36.733678	108.061942
Pool	3.191615	20.106787
Elevator	29.639503	116.069390
Patio_Balcony	22.160824	72.244284
Bed_Linens	23.468774	93.047222
CASE_COUNT	5.543201	357.588783
AC	18.819879	84.566780
instant_bookable	17.511152	108.405018
Cooking_Equip	20.955839	94.042562

Mean of squared residuals: 0.1413
% Var explained: 62.14

We chose random forest as our only non-parametric modeling method applying it to the two model specifications we decided on, namely the "initial set" and the "small set." We first used random splitting and the randomForest function to inspect how the MSE behaves as the number of trees fitted increase. We realized that the MSE clearly flattens once it reaches 100 trees. So, we decided to use 100 trees as our value for the number of trees.

We then used the "caret" package and first fitted a random forest model with the initial set of predictors using 10-fold CV and the results showed that 14 variables is the optimal number of predictors to be used for each tree fitted. The final model yielded a test MSE of 0.1535 and RMSE of 0.3903 while explaining 59.16 % of the variability in price.

For the small set of predictors, the results showed that 13 variables is the optimal number of predictors to be used for each tree fitted. The final model yielded a test MSE of 0.1423 and RMSE of 0.3759 while explaining 62.14% of the variability in price.

Fitting the Final Model Choice:

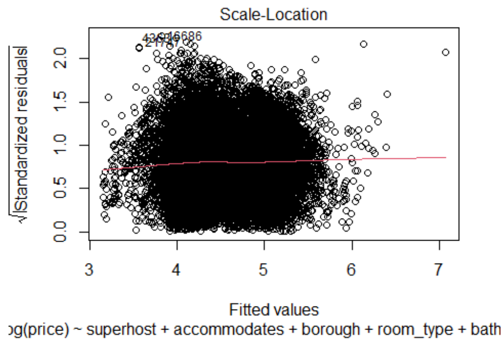
Snapshot of fitting the final (small WLS) model on the entire data set:

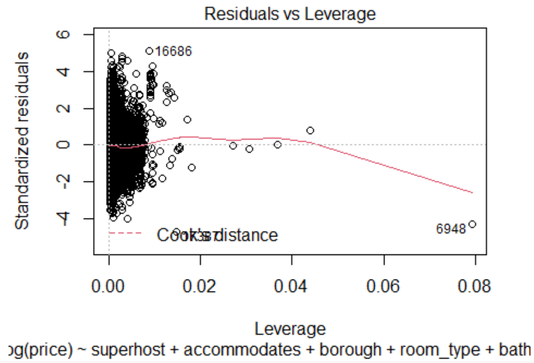
```
lm.wls.airbnb.final <- lm(log(price) ~ superhost + accommodates + borough + roo
m_type + bathrooms + bedrooms + rev_per_month + host_listings + availability +
```

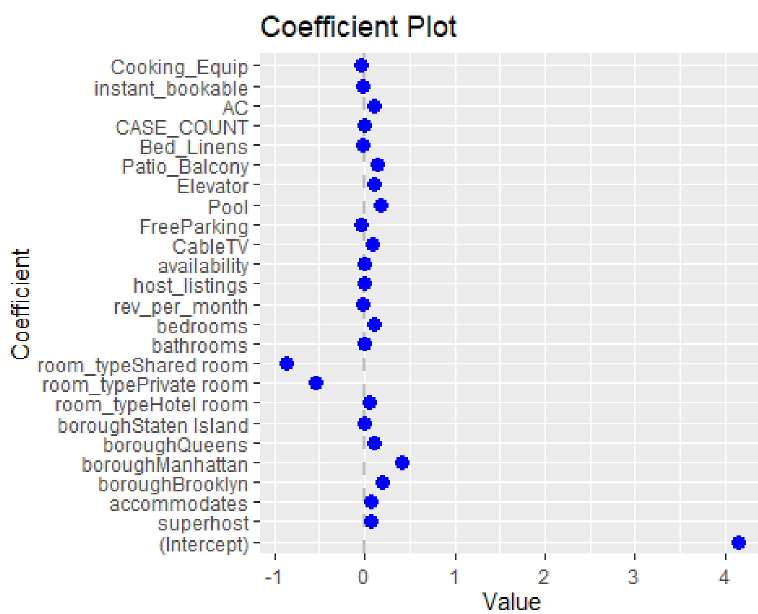
```
CableTV + FreeParking + Pool + Elevator + Patio_Balcony + Bed_Linens + CASE_COU
NT + AC + instant_bookable + Cooking_Equip, weights = wts.airbnb.wls.small, dat
a = Airbnb)

##
## Coefficients:
##                         Estimate   Std. Error t value Pr(>|t|)
## (Intercept)          4.143797479  0.019550453 211.954  < 2e-16 ***
## superhost            0.067161046  0.006593973  10.185  < 2e-16 ***
## accommodates         0.077240422  0.002381841  32.429  < 2e-16 ***
## boroughBrooklyn      0.191412908  0.015648713  12.232  < 2e-16 ***
## boroughManhattan     0.412285661  0.015852122  26.008  < 2e-16 ***
## boroughQueens        0.099211684  0.016588069   5.981 2.25e-09 ***
## boroughStaten Island 0.001977316  0.030958036   0.064   0.9491
## room_typeHotel room  0.044493654  0.033313197   1.336   0.1817
## room_typePrivate room -0.545371459 0.006427665 -84.848 < 2e-16 ***
## room_typeShared room -0.861912684  0.018871704 -45.672  < 2e-16 ***
## bathrooms           -0.001417206  0.006977609  -0.203   0.8391
## bedrooms             0.104384067  0.006099265  17.114  < 2e-16 ***
## rev_per_month       -0.027628257  0.002170159 -12.731  < 2e-16 ***
## host_listings       -0.003799437  0.000145495 -26.114  < 2e-16 ***
## availability         0.000132719  0.000018921   7.014 2.37e-12 ***
## CableTV              0.096476866  0.006318174  15.270  < 2e-16 ***
## FreeParking         -0.044805546  0.006234183  -7.187 6.81e-13 ***
## Pool                 0.177359293  0.026398119   6.719 1.88e-11 ***
## Elevator             0.108041792  0.006582429  16.414  < 2e-16 ***
## Patio_Balcony        0.135236661  0.008883782  15.223  < 2e-16 ***
## Bed_Linens          -0.024616680  0.006313801  -3.899 9.69e-05 ***
## CASE_COUNT          -0.000005348  0.000002596  -2.060   0.0394 *
## AC                   0.104218876  0.007784406  13.388  < 2e-16 ***
## instant_bookable    -0.012945393  0.005822276  -2.223   0.0262 *
## Cooking_Equip       -0.041473391  0.006498378  -6.382 1.78e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.28 on 24004 degrees of freedom
## Multiple R-squared:  0.5801, Adjusted R-squared:  0.5797
## F-statistic:  1382 on 24 and 24004 DF,  p-value: < 2.2e-16
```

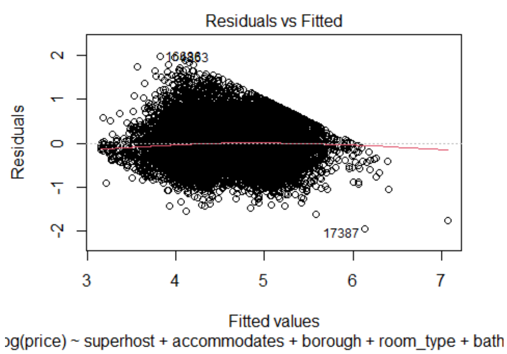
Plots for this model appear on the next page:

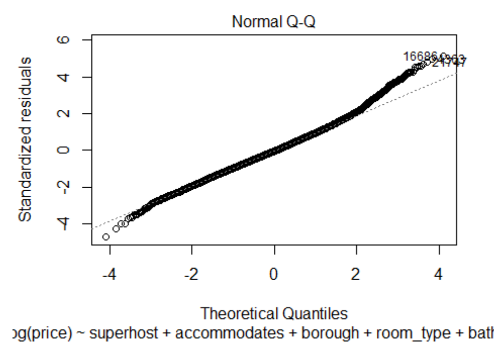
1. Fitted values versus the square root of standardized residuals
2. Leverage versus standardized residuals
3. Coefficient plot for all 19 predictors in the model
4. Fitted values versus residuals
5. QQ plot of normality











Goodies/Just for Fun:

```
# We noticed minor non-linearity between some predictors and price, so we tested all possible combinations and we are attaching here some, not all, of them. We are not including a polynomial term, but if we had, this is a snapshot of what the results on that might have looked like. We did not include any in our final model because the improvement in MSE (if applicable) is so minimal that it is not worth complicating the model and increasing the variance.

## Adding a polynomial term to Accommodates:
```

```r
lm.fit.airbnb.10FCV.small.poly <- train(log(price) ~ superhost + poly(accommoda
tes, 2) + borough + room_type + bathrooms + bedrooms + rev_per_month + host_lis
tings + availability + CableTV + FreeParking + Pool + Elevator + Patio_Balcony
+ Bed_Linens + CASE_COUNT + AC + instant_bookable + Cooking_Equip, weights = wt
s.airbnb.small, data = Airbnb, method = "lm", trControl = trainControl(method="
cv", number=10))

lm.fit.airbnb.10FCV.small.poly$results$RMSE^2
```

```
## [1] 0.1570489
```

```r
## Adding a polynomial term to bathrooms:

lm.fit.airbnb.small.bathrooms <- lm(price ~ superhost + instant_bookable + acco
mmodates + borough + room_type + poly(bathrooms,2,raw = TRUE) + bedrooms + rev_
per_month + host_listings + availability + AC + CableTV + FreeParking + Pool +
Elevator + Patio_Balcony + Cooking_Equip + Bed_Linens + CASE_COUNT, data = Airb
nb)
summary(lm.fit.airbnb.small.bathrooms,digits=4)
```

```
##
## Coefficients:
##                                 Estimate  Std. Error t value Pr(>|t|)
## (Intercept)                   36.7008947   2.9179294  12.578  < 2e-16 ***
## superhost                      7.5272133   0.8561187   8.792  < 2e-16 ***
## instant_bookable              -2.5634257   0.7571736  -3.386 0.000712 ***
## accommodates                   9.6203571   0.3054457  31.496  < 2e-16 ***
## boroughBrooklyn               16.9553748   2.0543719   8.253  < 2e-16 ***
## boroughManhattan              42.7750983   2.0797420  20.568  < 2e-16 ***
## boroughQueens                  9.4207470   2.1778865   4.326 1.53e-05 ***
## boroughStaten Island          -6.6285807   4.0486424  -1.637 0.101594
## room_typeHotel room           17.6689288   4.2630744   4.145 3.42e-05 ***
## room_typePrivate room        -52.6744800   0.8336850 -63.183  < 2e-16 ***
## room_typeShared room         -68.6066309   2.4984772 -27.459  < 2e-16 ***
## poly(bathrooms, 2, raw = TRUE)1  25.1537917   2.1658097  11.614  < 2e-16 ***
## poly(bathrooms, 2, raw = TRUE)2  -3.4717797   0.4894381  -7.093 1.34e-12 ***
## bedrooms                      17.8335604   0.7801839  22.858  < 2e-16 ***
## rev_per_month                 -3.8447178   0.2832254 -13.575  < 2e-16 ***
## host_listings                 -0.2472145   0.0192187 -12.863  < 2e-16 ***
## availability                   0.0121958   0.0024576   4.962 7.01e-07 ***
## AC                             8.0197062   1.0189783   7.870 3.69e-15 ***
## CableTV                       11.1716949   0.8170079  13.674  < 2e-16 ***
## FreeParking                   -5.7768748   0.8098909  -7.133 1.01e-12 ***
## Pool                          27.2925297   3.3879912   8.056 8.27e-16 ***
## Elevator                      12.8154371   0.8519527  15.042  < 2e-16 ***
## Patio_Balcony                 17.2049377   1.1496206  14.966  < 2e-16 ***
## Cooking_Equip                 -4.0515785   0.8450099  -4.795 1.64e-06 ***
## Bed_Linens                    -3.7772988   0.8201692  -4.606 4.14e-06 ***
## CASE_COUNT                    -0.0002711   0.0003372  -0.804 0.421539
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 51.61 on 24003 degrees of freedom
## Multiple R-squared:  0.5074, Adjusted R-squared:  0.5069
## F-statistic: 988.9 on 25 and 24003 DF,  p-value: < 2.2e-16
```

BP Test for weighted model (with polynomial bathroom term):

```
##
##  studentized Breusch-Pagan test
##
## data:  lm.wls.bathroom
## BP = 655.4, df = 25, p-value < 2.2e-16
```

```
lm.fit.bathroom.10FCV <- train(log(price) ~ superhost + instant_bookable + acco
mmodates + borough + room_type + poly(bathrooms,2) + bedrooms + rev_per_month +
host_listings + availability + AC + CableTV + FreeParking + Pool + Elevator + P
atio_Balcony + Cooking_Equip + Bed_Linens + CASE_COUNT, weights=wts.bathroom,da
ta = Airbnb,method="lm",trControl = trainControl(method="cv", number=10))
```

10FCV results for weighted model (with a polynomial bathroom term):

```
lm.fit.bathroom.10FCV$results$RMSE
```

```
## [1] 0.3976856
```

```
lm.fit.bathroom.10FCV$results$RMSE^2
```

```
## [1] 0.1581538
```

```
## Adding a polynomial term to bedrooms:

lm.fit.airbnb.small.bedrooms <- lm(price ~ superhost + instant_bookable + accom
modates + borough + room_type + poly(bedrooms,2,raw = TRUE) + bathrooms + rev_p
er_month + host_listings + availability + AC + CableTV + FreeParking + Pool + E
levator + Patio_Balcony + Cooking_Equip + Bed_Linens + CASE_COUNT, data = Airbn
b)

summary(lm.fit.airbnb.small.bedrooms,digits=4)
##
## Coefficients:
##                                 Estimate  Std. Error t value Pr(>|t|)
## (Intercept)                   41.2026637   2.6242149  15.701  < 2e-16 ***
## superhost                      7.5435465   0.8555741   8.817  < 2e-16 ***
## instant_bookable              -2.6017719   0.7566093  -3.439 0.000585 ***
## accommodates                   9.1345423   0.3111227  29.360  < 2e-16 ***
## boroughBrooklyn               16.7985007   2.0531948   8.182 2.94e-16 ***
## boroughManhattan              42.5785510   2.0783510  20.487  < 2e-16 ***
## boroughQueens                  9.1914668   2.1765243   4.223 2.42e-05 ***
## boroughStaten Island          -6.9334763   4.0461284  -1.714 0.086614 .
## room_typeHotel room           18.3677252   4.2605479   4.311 1.63e-05 ***
## room_typePrivate room        -52.0518905   0.8333172 -62.463  < 2e-16 ***
## room_typeShared room         -68.2289835   2.4972732 -27.321  < 2e-16 ***
## poly(bedrooms, 2, raw = TRUE)1 25.8197416   1.1492109  22.467  < 2e-16 ***
## poly(bedrooms, 2, raw = TRUE)2 -1.2734875   0.1415411  -8.997  < 2e-16 ***
## bathrooms                     11.0108685   0.9065601  12.146  < 2e-16 ***
## rev_per_month                 -3.8115621   0.2830802 -13.465  < 2e-16 ***
```

```
## host_listings                     -0.2468461   0.0192061 -12.853  < 2e-16 ***
## availability                       0.0118512   0.0024557   4.826 1.40e-06 ***
## AC                                 8.0750838   1.0183338   7.930 2.29e-15 ***
## CableTV                           11.3500828   0.8165093  13.901  < 2e-16 ***
## FreeParking                       -5.8864357   0.8094478  -7.272 3.65e-13 ***
## Pool                              27.6072221   3.3856271   8.154 3.68e-16 ***
## Elevator                          13.0165996   0.8513257  15.290  < 2e-16 ***
## Patio_Balcony                     17.3752666   1.1482509  15.132  < 2e-16 ***
## Cooking_Equip                     -3.8420574   0.8441062  -4.552 5.35e-06 ***
## Bed_Linens                        -3.8266946   0.8195365  -4.669 3.04e-06 ***
## CASE_COUNT                        -0.0002941   0.0003370  -0.873 0.382848
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.58 on 24003 degrees of freedom
## Multiple R-squared:  0.508,  Adjusted R-squared:  0.5075
## F-statistic: 991.4 on 25 and 24003 DF,  p-value: < 2.2e-16
```

After weighting, a BP test for a weighted model with a polynomial term for bedrooms:

```
bptest(lm.wls.bedrooms)

##
##  studentized Breusch-Pagan test
##
## data:  lm.wls.bedrooms
## BP = 625.45, df = 25, p-value < 2.2e-16

lm.fit.bedrooms.10FCV <- train(log(price) ~ superhost + instant_bookable + acco
mmodates + borough + room_type + poly(bedrooms,2) + bathrooms + rev_per_month +
host_listings + availability + AC + CableTV + FreeParking + Pool + Elevator + P
atio_Balcony + Cooking_Equip + Bed_Linens + CASE_COUNT, weights=wts.bedrooms,da
ta = Airbnb,method="lm",trControl = trainControl(method="cv", number=10))
```

10FCV Results for a weighted model (with a polynomial bedroom term):

```
lm.fit.bedrooms.10FCV$results$RMSE

## [1] 0.3992493

#To obtain the MSE
lm.fit.bedrooms.10FCV$results$RMSE^2

## [1] 0.1594

## A model with a polynomial term for host_listings, followed by all the above
plus host squared:

lm.fit.airbnb.small.host <- lm(price ~ superhost + instant_bookable + accommoda
tes + borough + room_type + poly(host_listings,2,raw = TRUE) + bathrooms + rev_
per_month + bedrooms + availability + AC + CableTV + FreeParking + Pool + Eleva
tor + Patio_Balcony + Cooking_Equip + Bed_Linens + CASE_COUNT, data = Airbnb)

lm.fit.airbnb.small.quad <- lm(price ~ superhost + instant_bookable + accommoda
tes + borough + room_type + poly(host_listings,2,raw = TRUE) + poly(bathrooms,2
,raw=TRUE) + rev_per_month + poly(bedrooms,2,raw=TRUE) + availability + AC + Ca
```

```
bleTV + FreeParking + Pool + Elevator + Patio_Balcony + Cooking_Equip + Bed_Lin
ens + CASE_COUNT, data = Airbnb)

##
## Coefficients:
##                                 Estimate  Std. Error t value Pr(>|t|)
## (Intercept)                   46.8595262   2.5384940  18.460  < 2e-16
## superhost                      7.1736885   0.8514383   8.425  < 2e-16
## instant_bookable              -2.5045306   0.7527705  -3.327 0.000879
## accommodates                   9.9467913   0.3039262  32.728  < 2e-16
## boroughBrooklyn               16.9855336   2.0425636   8.316  < 2e-16
## boroughManhattan              43.5242445   2.0682977  21.044  < 2e-16
## boroughQueens                  9.7953426   2.1655062   4.523 6.12e-06
## boroughStaten Island          -6.9629305   4.0254222  -1.730 0.083690
## room_typeHotel room           24.3432365   4.2533470   5.723 1.06e-08
## room_typePrivate room        -51.6811327   0.8290822 -62.335  < 2e-16
## room_typeShared room         -66.9762433   2.4858037 -26.943  < 2e-16
## poly(host_listings, 2, raw = TRUE)1  -1.0242231   0.0469747 -21.804  < 2e-16
## poly(host_listings, 2, raw = TRUE)2   0.0049043   0.0002705  18.131  < 2e-16
## bathrooms                     11.9503512   0.9026264  13.240  < 2e-16
## rev_per_month                 -3.9925159   0.2817032 -14.173  < 2e-16
## bedrooms                      17.7591195   0.7743196  22.935  < 2e-16
## availability                   0.0165430   0.0024566   6.734 1.69e-11
## AC                             7.8598143   1.0131766   7.758 9.00e-15
## CableTV                       11.2032934   0.8122600  13.793  < 2e-16
## FreeParking                   -6.1209283   0.8054517  -7.599 3.08e-14
## Pool                          25.4532349   3.3703481   7.552 4.44e-14
## Elevator                      12.2036387   0.8478595  14.393  < 2e-16
## Patio_Balcony                 17.2352774   1.1424045  15.087  < 2e-16
## Cooking_Equip                 -3.5709341   0.8399559  -4.251 2.13e-05
## Bed_Linens                    -3.7025194   0.8153902  -4.541 5.63e-06
## CASE_COUNT                    -0.0003527   0.0003353  -1.052 0.292813
##
##
## Residual standard error: 51.31 on 24003 degrees of freedom
## Multiple R-squared:  0.513,  Adjusted R-squared:  0.5125
## F-statistic:  1011 on 25 and 24003 DF,  p-value: < 2.2e-16

summary(lm.fit.airbnb.small.quad,digits=4)
##
## Coefficients:
##                                 Estimate  Std. Error t value Pr(>|t|)
## (Intercept)                   31.7707513   2.9494193  10.772  < 2e-16
## superhost                      7.1763863   0.8491866   8.451  < 2e-16
## instant_bookable              -2.3694854   0.7508994  -3.156   0.0016
## accommodates                   9.3540955   0.3091433  30.258  < 2e-16
## boroughBrooklyn               16.6998899   2.0373294   8.197 2.59e-16
## boroughManhattan              43.5849403   2.0629205  21.128  < 2e-16
## boroughQueens                  9.7398400   2.1598702   4.509 6.53e-06
## boroughStaten Island          -7.1051834   4.0148646  -1.770   0.0768
## room_typeHotel room           24.5656176   4.2425294   5.790 7.11e-09
## room_typePrivate room        -51.5807106   0.8286963 -62.243  < 2e-16
## room_typeShared room         -66.5742712   2.4796071 -26.849  < 2e-16
```

```
## poly(host_listings, 2, raw = TRUE)1  -1.0253031   0.0468507 -21.884  < 2e-16
## poly(host_listings, 2, raw = TRUE)2   0.0049007   0.0002698  18.166  < 2e-16
## poly(bathrooms, 2, raw = TRUE)1      25.2708438   2.1493491  11.757  < 2e-16
## poly(bathrooms, 2, raw = TRUE)2      -3.3614847   0.4855154  -6.924 4.52e-12
## rev_per_month                        -3.9440851   0.2809970 -14.036  < 2e-16
## poly(bedrooms, 2, raw = TRUE)1       24.8230257   1.1430805  21.716  < 2e-16
## poly(bedrooms, 2, raw = TRUE)2       -1.2426980   0.1404964  -8.845  < 2e-16
## availability                          0.0168033   0.0024505   6.857 7.20e-12
## AC                                    7.8753456   1.0105074   7.793 6.78e-15
## CableTV                              11.2366417   0.8102623  13.868  < 2e-16
## FreeParking                          -6.2056403   0.8033952  -7.724 1.17e-14
## Pool                                 25.2385468   3.3616470   7.508 6.22e-14
## Elevator                             12.1845007   0.8458233  14.405  < 2e-16
## Patio_Balcony                        16.8438424   1.1401485  14.773  < 2e-16
## Cooking_Equip                        -3.7131249   0.8381134  -4.430 9.45e-06
## Bed_Linens                           -3.5459257   0.8133867  -4.359 1.31e-05
## CASE_COUNT                           -0.0003234   0.0003344  -0.967   0.3336
##
## Residual standard error: 51.18 on 24001 degrees of freedom
## Multiple R-squared:  0.5156, Adjusted R-squared:  0.5151
## F-statistic: 946.3 on 27 and 24001 DF,  p-value: < 2.2e-16
```

```
anova(lm.fit.airbnb.small,lm.fit.airbnb.small.quad)
```

```
## Analysis of Variance Table
##
## Response: log(price)
##                     Df Sum Sq Mean Sq    F value    Pr(>F)
## superhost            1    6.2    6.19    39.1997 3.891e-10 ***
## accommodates         1 2552.9 2552.87 16163.7311 < 2.2e-16 ***
## borough              4  813.6  203.41  1287.9055 < 2.2e-16 ***
## room_type            3 1465.1  488.37  3092.1335 < 2.2e-16 ***
## bathrooms            1    6.1    6.11    38.7115 4.994e-10 ***
## bedrooms             1   46.4   46.36   293.5229 < 2.2e-16 ***
## rev_per_month        1   35.5   35.54   225.0453 < 2.2e-16 ***
## host_listings        1  104.1  104.12   659.2401 < 2.2e-16 ***
## availability         1    8.6    8.58    54.3224 1.756e-13 ***
## CableTV              1   47.0   47.02   297.7077 < 2.2e-16 ***
## FreeParking          1   13.7   13.74    87.0046 < 2.2e-16 ***
## Pool                 1   14.9   14.89    94.2804 < 2.2e-16 ***
## Elevator             1   54.0   53.98   341.7991 < 2.2e-16 ***
## Patio_Balcony        1   31.7   31.70   200.7340 < 2.2e-16 ***
## Bed_Linens           1    6.2    6.15    38.9467 4.428e-10 ***
## CASE_COUNT           1    0.6    0.65     4.1154   0.04251 *
## AC                   1   26.9   26.86   170.0956 < 2.2e-16 ***
## instant_bookable     1    0.8    0.82     5.2204   0.02233 *
## Cooking_Equip        1    6.2    6.18    39.1468 3.997e-10 ***
## Residuals        24004 3791.1    0.16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

After running BP test on both the host and quad models, we weight and then obtain their 10FCV MSE results:

```
lm.fit.host.10FCV$results$RMSE

## [1] 0.3947661

lm.fit.host.10FCV$results$RMSE^2

## [1] 0.1558403

lm.fit.quad.10FCV$results$RMSE

## [1] 0.393127

lm.fit.quad.10FCV$results$RMSE^2

## [1] 0.1545488
```

References

Glusac, E. (2020, May 14). *Hotels vs. Airbnb: Has Covid-19 Disrupted the Disrupter?* The New York Times. https://www.nytimes.com/2020/05/14/travel/hotels-versus-airbnb-pandemic.html.

How do reviews work for stays? - Airbnb Help Center. Airbnb. (2021). https://www.airbnb.com/help/article/13/how-do-reviews-work-for-stays.

Oliver, D. (2020, August 27). *Travelers are flocking to Airbnb, Vrbo more than hotels during COVID-19 pandemic. But why?* USA Today. https://www.usatoday.com/story/travel/hotels/2020/08/26/airbnb-vrbo-more-popular-than-hotels-during-covid-19-pandemic/5607312002/.

Karun, K. (2021, February 7). *Airbnb US dataset*. Kaggle. https://www.kaggle.com/kavithakaruna/airbnb-us-dataset.

Mooney, J. (2020, July 28). *Short-term rentals weathered COVID-19 better than hotels, data firms say*. S&P Global Market Intelligence. https://www.spglobal.com/marketintelligence/en/news-insights/latest-news-headlines/short-term-rentals-weathered-covid-19-better-than-hotels-data-firms-say-59606578.

Medine, T. (2020, October 28). 54% of Americans Have Stayed in Hotels and Airbnbs During the Coronavirus Pandemic. ValuePenguin. https://www.valuepenguin.com/news/americans-stay-in-hotels-and-airbnbs-during-coronavirus.

Molla, R. (2019, March 25). *American consumers spent more on Airbnb than on Hilton last year*. Vox. https://www.vox.com/2019/3/25/18276296/airbnb-hotels-hilton-marriott-us-spending.

New York City Department of Health. (2021). *COVID-19: NYC Health Data*. COVID-19: Latest Data - NYC Health. https://www1.nyc.gov/site/doh/covid/covid-19-data.page#epicurve.