Venice Pirates

Fall 2017, 425 Final Project

Mauricio Campos, Joshua Loyal, Austin Jay Warner

12/19/2017

Introduction

Spending days on the open sea hunting for treasure can leave any pirate suffering from scurvy. Therefore, it is important for a pirate to have an easy way to find the perfect place on land to drop anchor and to rest up for the next big excursion. In this project we use data from Airbnb in Venice, Italy to design a data-driven application that will help these pirates find the perfect booking.

Of course, the concept of a 'perfect' booking can mean different things for different pirates. For some pirates the price of a booking could be the important factor. As such they may be interested in knowing the affect of various co-variates on the price of a booking. This information would allow them to determine whether they are getting a fair price. Other pirates may be concerned with where to dock their ships. As such they would like to discriminate the price of the bookings by their geographic location within Venice. While others pirates may be concerned with the attractions close to their booking. With these factors in mind, we narrowed down our analysis to answering the following three questions:

1. Is there a difference in pricing in Airbnb listings among the different neighborhoods of Venice?
2. What landmarks distinguish the Venetian neighborhoods?
3. What variables best predict prices of Airbnb listings in Venice?

In what follows, we answer the first question using an ANOVA with the Venetian neighborhoods as the independent variable and price as the dependent variable. The second question is tackled using a linear model to distinguish the neighborhoods based on the user's text reviews. Finally, another linear model is used to develop a model to predict price based on common co-variates in the Airbnb dataset.[1][2][3]

Data Background

The data was taken from the website insideairbnb.com, which is an independent entity from Airbnb that lets you explore data from certain cities around the world. The data that they compile in their website is publicly available information. The data set for Venice was released in partnership with RESET VENEZIA, a group of local activists that strive to use data for the benefit of the local communities in Venice. The data was compiled on May 9th 2017 and has 6027 listings and 16 variables.

Data Cleaning

Of the 16 variables, 4 represent ID variables and the rest have potential use in a linear model. These are: neighborhood group, neighborhood, latitude, longitude, room type, price, minimum nights, number of reviews, date of the last review, reviews per month, calculated host listings count and days available in a year. Date of the last review was not taken into account since not all listings had reviews and number or reviews or reviews per month could be a much more powerful predictive variable in a future model. Another variable that was discarded early on was the calculated host listings count which specified, for each host, the estimated number

[1] The accompanying github repo with the code and the data can be found at https://github.com/moecampos/425-project/tree/master.

[2] The video presentation of the UI can be found on youtube at https://www.youtube.com/watch?v=AC9EP9B6KNs.

[3] The UI is hosted on shinyapps at https://joshloyal.shinyapps.io/425-project/.

of listings that entity had. Since this variable describes best the host instead of the listing we decided to not use it in the model.

There were 784 NA's in the data set, however they were all in the variable reviews per month. By doing careful analysis of the data set it was discovered that all the NA's corresponded to listings with zero reviews so the NA's were changed into zeroes.

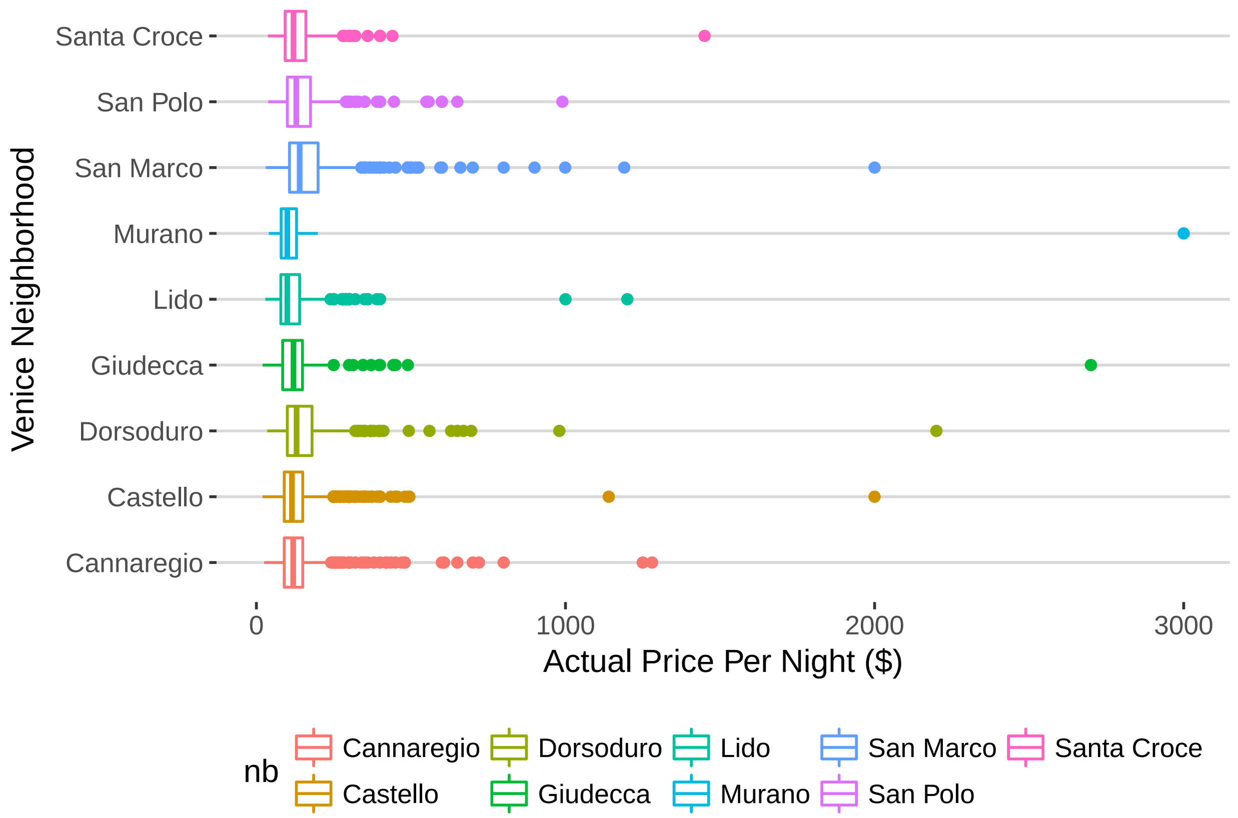
The location of a listing will have a large impact on the pricing in a predictive sense. Our data gives us several ways of measuring the location of a listing: the neighborhood of the listing, the neighborhood group of the listing, the latitude, and the longitude. These covariates are strongly correlated and so we decided to remove some of these predictors. The most interpretable of these covariates were neighborhood and neighborhood group. Neighborhood group classifies listings according to whether they are mainland or island and since a given neighborhood is either on the island or mainland, it is sufficient to consider only neighborhood.

Once we decided to use neighborhoods as our geographical predictor, we faced the challenge of scale. The neighborhood predictor, as a categorical variable, has 56 levels, which is simply too many levels to incorporate into a simple linear regression. In order to reduce the number of neighborhoods, we chose the nine most frequent neighborhoods in the dataset. This reduced set of neighborhoods included "San Marco", "Castello", "Cannaregio", "Dorsoduro", "Giudecca", "Lido", "Santa Croce", "Murano", and "San Polo." Indeed, these neighborhoods account for 81.5% of the listings. Furthermore, we think that these neighborhood are the most convenient for pirates, since each of these neighborhoods is accessible by water.

In the end, the dataset was reduced to 8 variables and 4910 observations. Of those 8 variables, 7 were used in the pricing model and the 8th one (listing id) was used to link the listings with the analysis of the reviews.

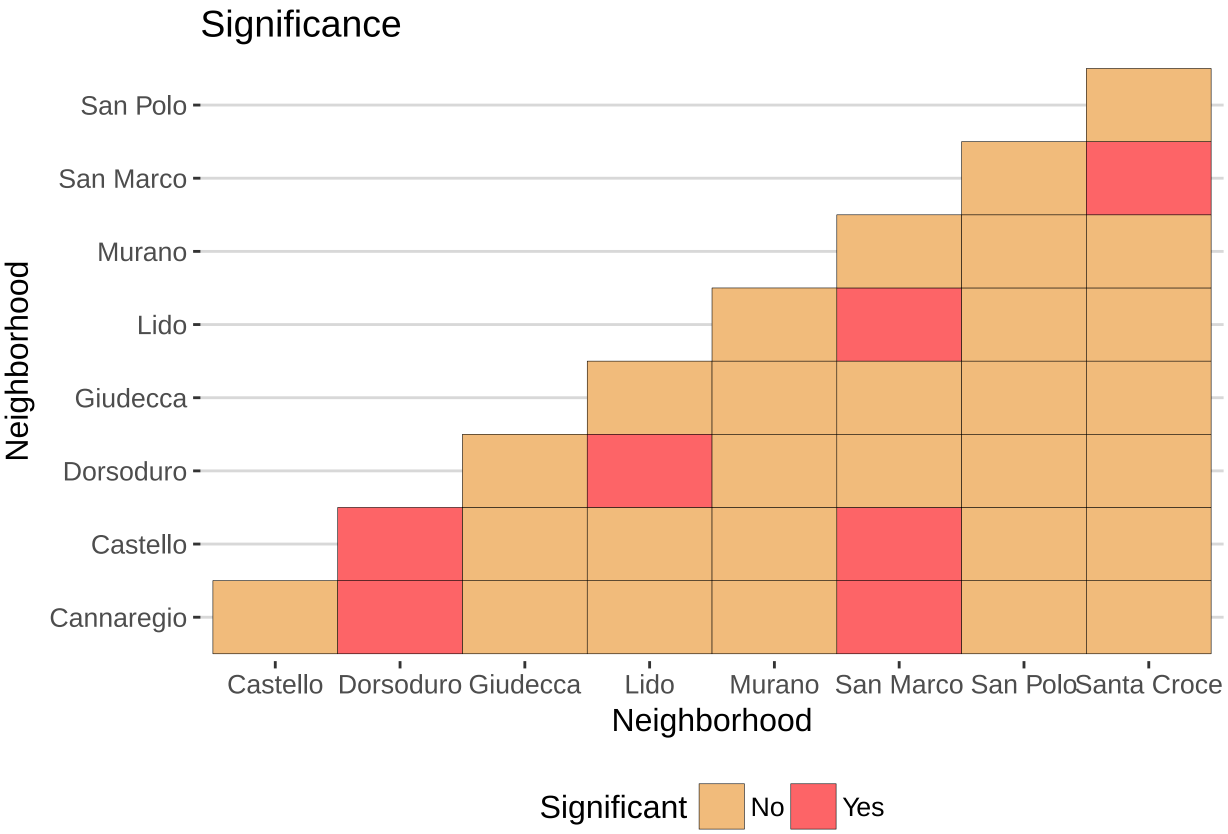
Neighborhood ANOVA

The goal of this section was to investigate the effect the Venetian neighborhoods had on the price of a listing. The following scatter-plot displays the price split across the various neighborhoods:



Although not rigorous, there seems to some variation in the prices among the neighborhoods. In particular, San Marco and Dorsoduro appear to have higher prices than the other neighborhoods.

In order to investigate the pricing differences in a statistically rigorous way, we fit a one-way ANOVA model with neighborhood as the independent variable and price as the response. Since we were interested in comparing all pairwise differences in the results, we used Tukey's HSD at an overall confidence level of 95% to measure the statistical significance of these contrasts. The result of this analysis is displayed in the following figure:



Each square represents the pairwise comparison between the corresponding column and row. An orange box indicates the Tukey's HSD concluded the difference in price was not significant at an $\alpha = 5\%$ level. A red box indicates that the test was in fact significant. According to this analysis, we found that there are significant differences in pricing between Dorsoduro and Castello, Dorsoduro and Cannaregio, Dorsoduro and Lido, as well as between San Marco and Cannaregio, San Marco and Castello, San Marco and Santa Croce, and San Marco and Lido. In fact, San Marco and Dorsoduro have the most red squares associated with them. This seems to match our observation that they are priced differently than the other neighborhoods. This makes sense because these neighborhoods are located in the heart of Venice.

Text Analysis

The question posed by this section is the following: What are the important terms within the user reviews that can distinguish the nine Venetian neighborhoods? When choosing a location to stay the price is not the only determining factor. One may also want the location to be close to a landmark they plan to visit or be in a district know for a particular interest. For example, a neighborhood may be know for its restaurants. Therefore, a visitor who plans to do a lot sampling of local cuisine may want to book a room in the 'foody' neighborhood. In summery, the coefficients of a multinomial lasso trained on tf-idf features were used to determine the most important terms in the user reviews.

Data Cleaning

The raw data used in this analysis comprised of 216,295 multi-lingual user reviews of listings in Venice taken from the Inside Airbnb project. It should be noted that each listing could have multiple reviews. As far as data quality, some reviews are clearly spam coming from bots posting on Airbnb's website. Nothing was done to remove these reviews. They seemed to comprise only a small fraction of the total reviews and the final model does not seem affected by them.

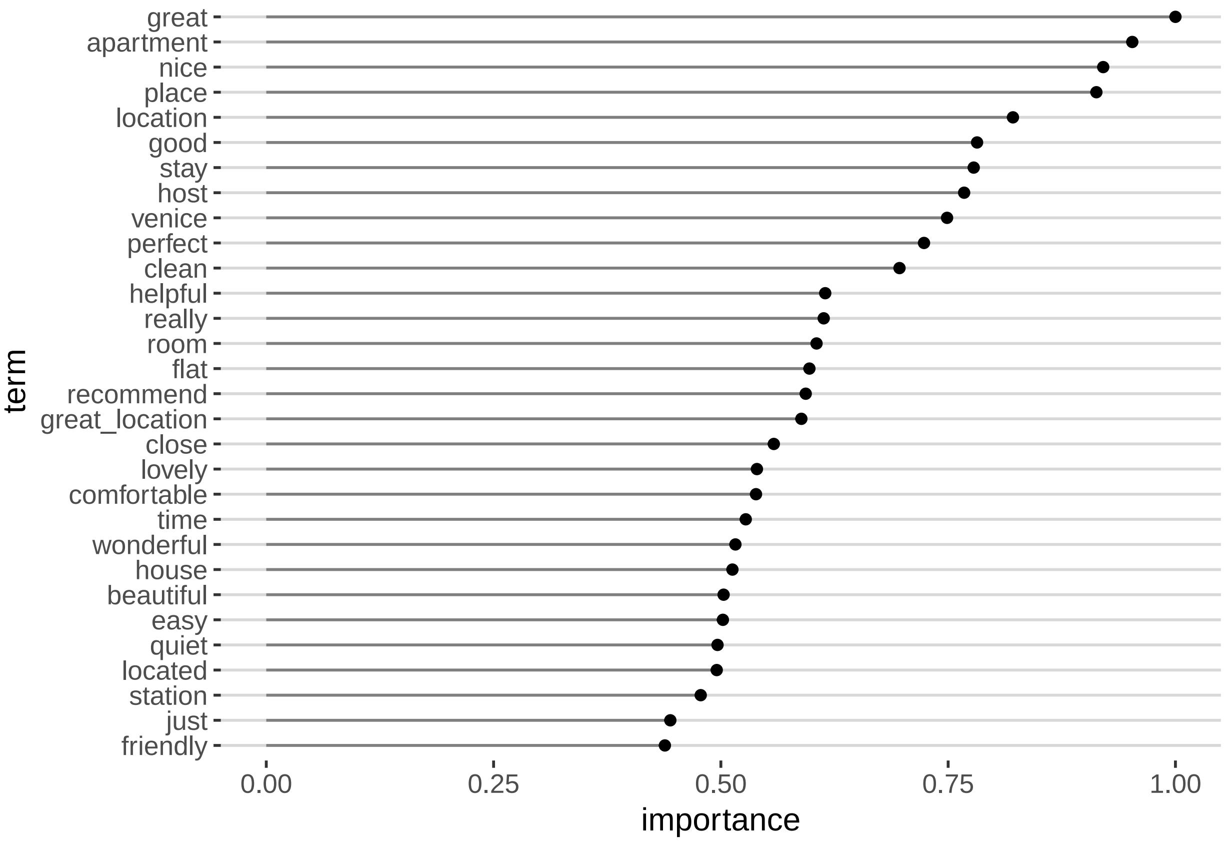
A few steps were taken before the feature extraction phase. The reviews were joined with the listing table (a inner join on `listing_id = id`) and filtered to lie in the nine neighborhoods considered in this report. In addition, only English reviews were used in the analysis. The `textcat` package was used to classify the language of the reviews. This package compares n-gram statistics between languages to make its decision. The aforementioned filtering allowed us to cut down the number of reviews to a total of 127,133 used in this analysis.

Feature Extraction

TF-IDF (term frequency - inverse document frequency) features were used in the final model. In order to construct these the corpus had to be cleaned and tokenized. In particular, we performed the following pre-processing on the corpus

- Tokens were defined by the regexp `\\b\\w\\w+\\b`. This expression defines a token as 2 or more alphanumeric characters (punctuation is completely ignored and always treated as a token separator).
- All numbers were removed with the regexp `\\d+`. It was assumed numbers would mostly correspond to prices, dates, and street numbers which were not of interest in this analysis.
- All letters were converted to lowercase.
- Tokens occurring in more than 80% of the document or less than 0.1% of the document were removed. The assumption is that they occur in too much or too little of the corpus to be discriminative.
- A list of common stop words (words like the, a, you, we) were removed from the corpus. In addition, the names of the neighborhoods as well as miss-spellings of the neighborhoods were removed from the corpus. This was done to avoid learning the name of the neighborhood as the most important feature.

Once the text was cleaned, uni-grams and bi-grams were construct and transformed into tf-idf weights using the `text2vec` package. The following bar chart contains the top 30 most important uni-grams and bi-grams in the corpus:



The importance is given by the sum of the terms weight across the whole corpus divided by the summation of all tf-idf weights. The terms look as you would expect from a corpus of house review data. Many terms describing the contents of the listings and the hosts (apartment, great, friendly, etc.).

Modeling

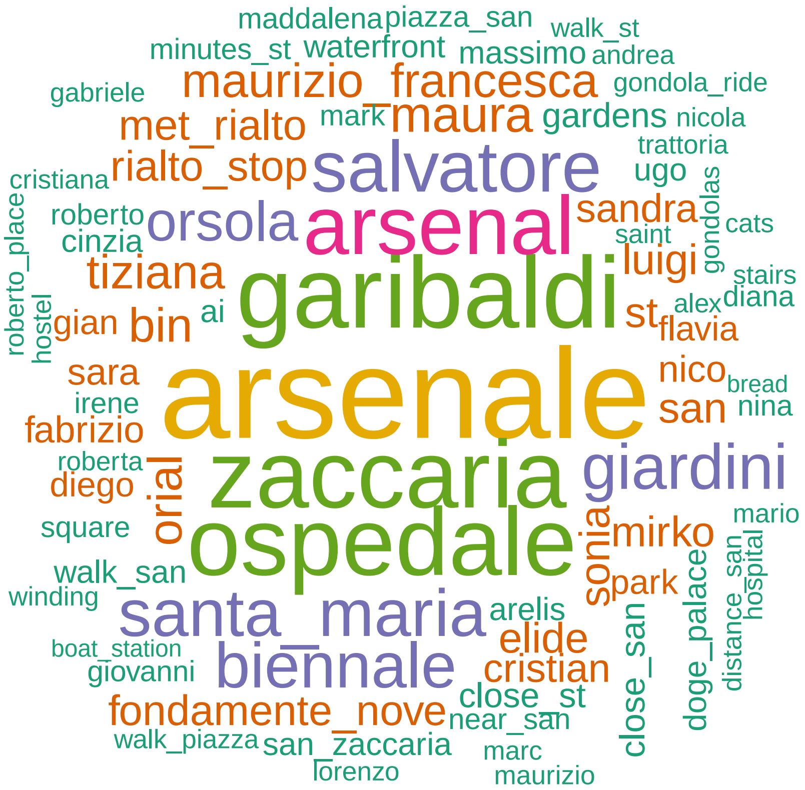
A multinomial lasso model was used to determine the effect of the various terms in classifying the nine neighborhoods of Venice. The model was fit using the `glmnet` package. The value of the regularization parameter λ was chosen using 5-fold cross-validation. The result is a model of the form:

$$Pr(\text{Neighborhood} = k|X = x) = \frac{e^{\beta_{0k}+\beta_k^T x}}{\sum_{l=1}^{K} e^{\beta_{0l}+\beta_l^T x}}$$

The non-zero coefficients should give us a way to measure the importance of a term in classifying a particular neighborhood.

Results

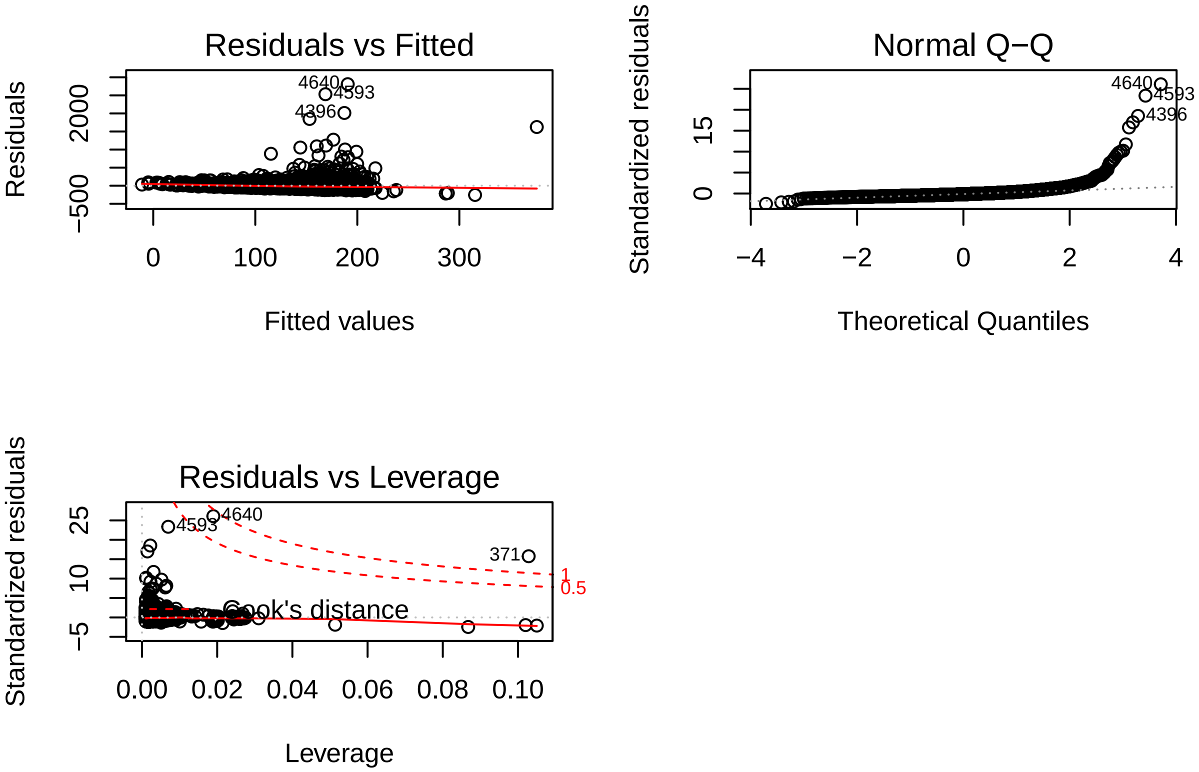
The results of this analysis are the coefficients of the multinomial lasso. Large coefficients in magnitude should indicate a stronger effect on the classification of a given neighborhood. In order to display this information to the user, the UI utilizes a word-cloud per neighborhood of the top 100 terms ranked by the magnitude of the coefficients of the model. Words that are larger in the word-cloud have larger coefficients in magnitude. An example for the neighborhood Castello is displayed below.



One can see that the most important terms include: `arsenale`, `garibaldi`, and `zaccaria`. This corresponds to the Venetian Arsenal, the Via Garibaldi, and the San Zacarria which are various attractions in the area. Of course the model is not perfect. Names of various landlords as well as sites located in other neighborhoods are located in the word cloud. However, they tend to have smaller coefficients compared to the actual landmarks.

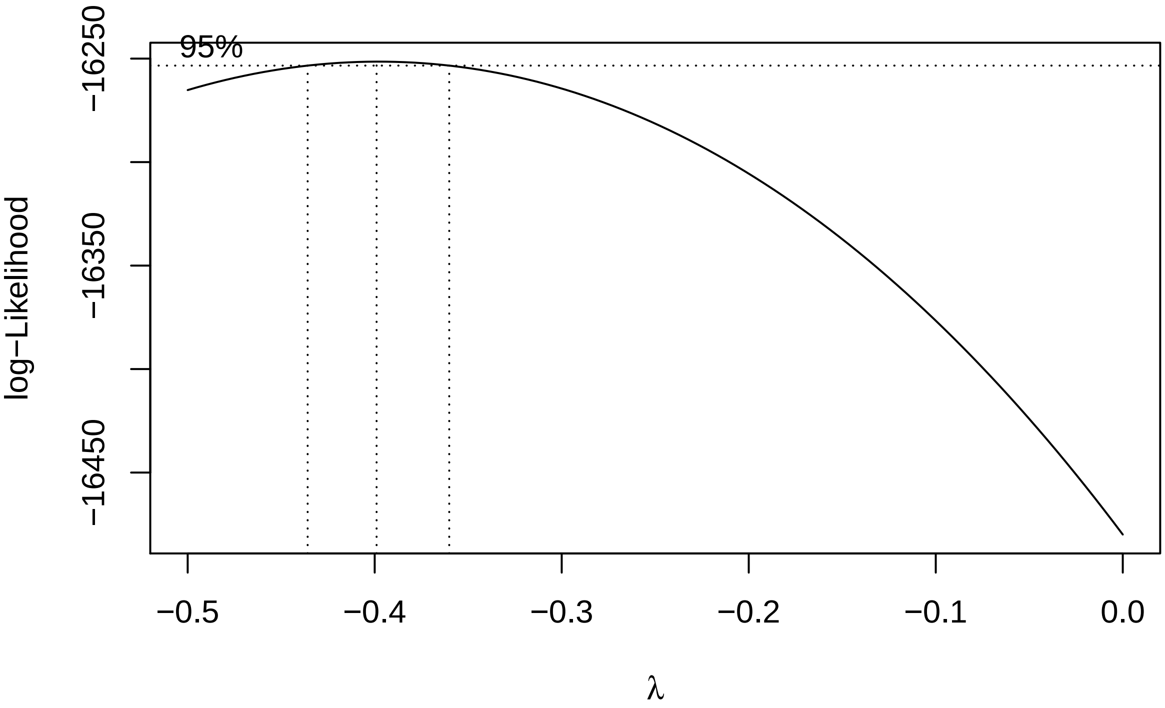
Pricing Model

Finally, we would like develop a model to predict the price of a listing as well as determine the effect of the co-variates on that price. The first model considered was: `Price ~ availability + neighborhood + room type + minimum nights + number of reviews + reviews per month`. This first model presented many problems, including a small R^2 value of 0.099. Also, observing the diagnostic plots we can see that this model also posed problems of normality, homogeneity of the variance and some really influential points.

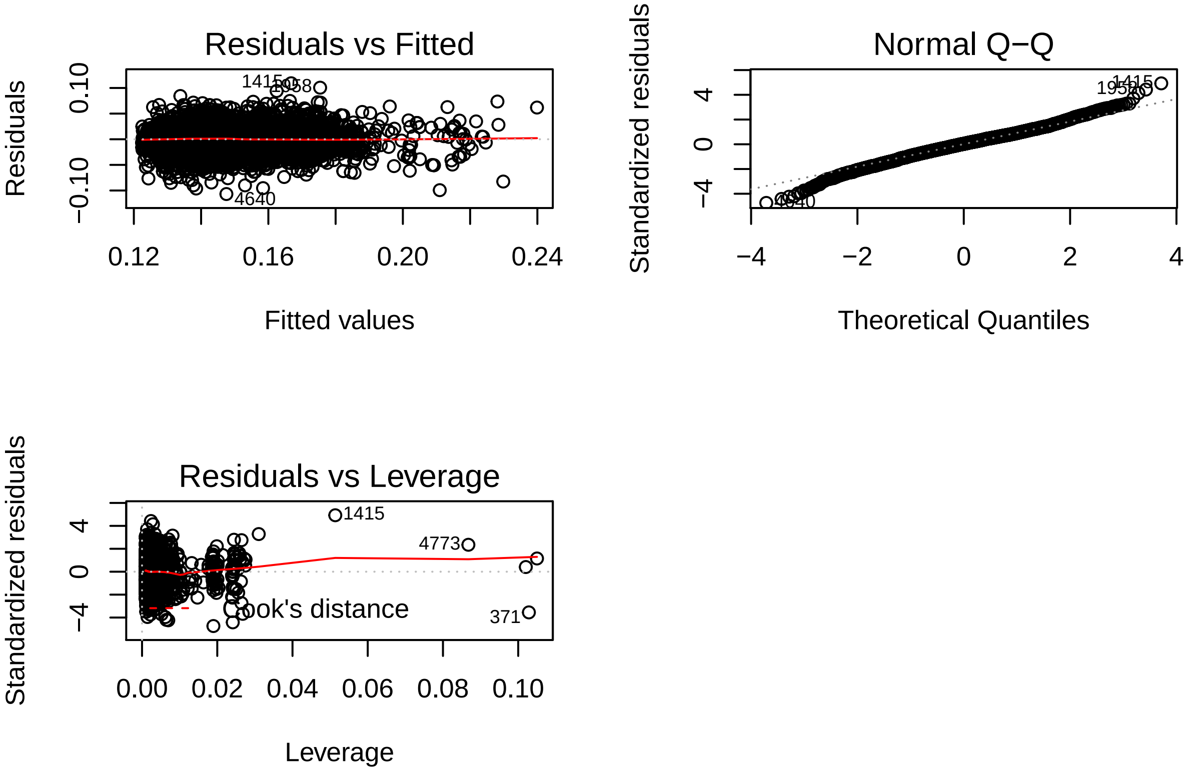


Seeing these plots we can also identify several listings that are clear outliers. These listings had prices above $1000. It is important to note that in the whole data set only 11 listings have prices above that price range. For exploration purposes, the three influential observations were removed from the model and then fitted again and the results were that minimum nights was no longer a significant variable in the model.

In addition, we considered doing a Box-Cox transformation in order to improve the goodness-of-fit of the model. The following displays the log-likelihood of the box-cox fit:



We can see from the confidence interval that we can use the value of $\lambda = -0.4$ to transform price. We then fitted the model again but with the transformed price. Observing the diagnostic plots we can see that the original problems haven been fixed. The R^2 was also increased to 0.3139.



However, in this model minimum nights was no longer a significant variable. We decided to use leaps to select the best model, observing both AIC and BIC. The result was that the best model (with the lowest AIC and BIC) was the following:

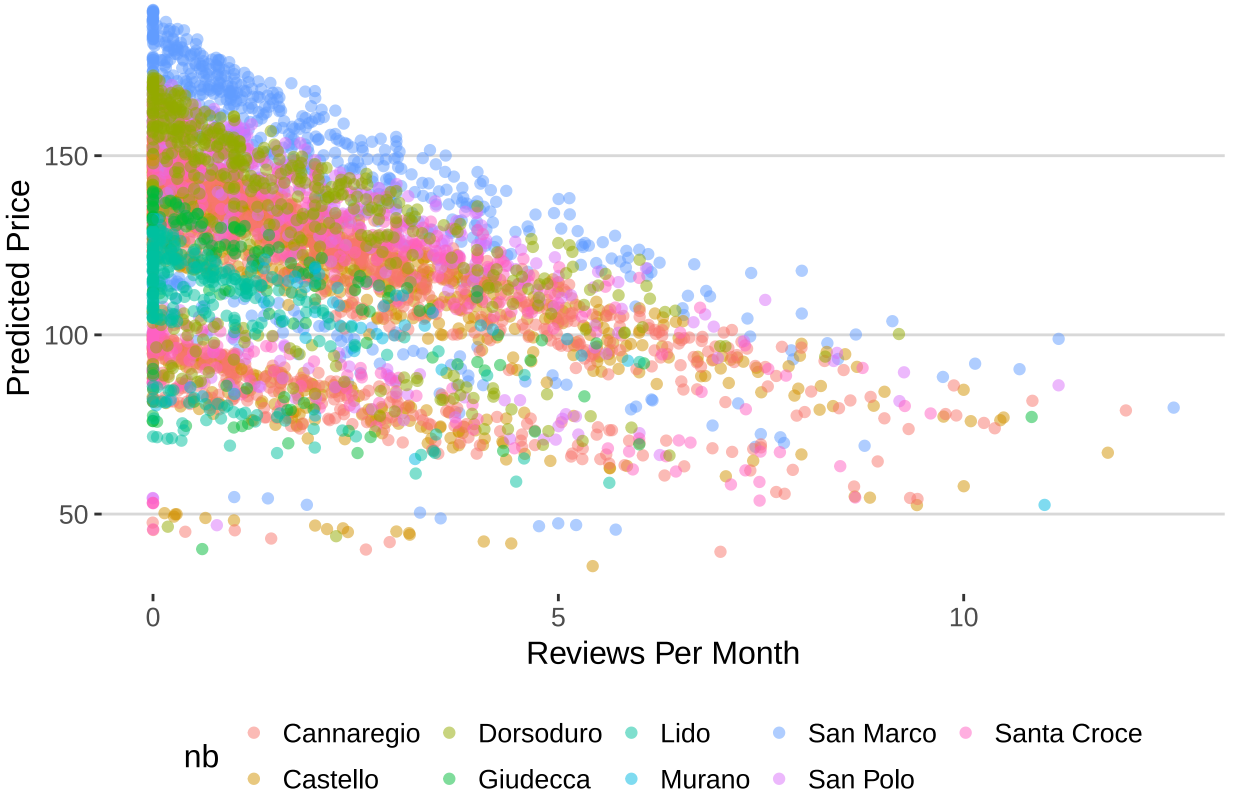
Price$^{-0.4}$ ~ availability + neighborhood + room type + number of reviews + reviews per month.

With the following summary statistics:

```
##
## Call:
## lm(formula = price.bc ~ avail + num_reviews + rpm + nb + room_type,
##     data = listings)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.106867 -0.013667  0.000831  0.014297  0.108977
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.457e-01  1.119e-03 130.164  < 2e-16 ***
## avail                 -3.519e-05  3.334e-06 -10.552  < 2e-16 ***
## num_reviews            3.246e-05  8.148e-06   3.984 6.87e-05 ***
## rpm                    3.019e-03  2.163e-04  13.961  < 2e-16 ***
## nbCastello             7.986e-04  9.214e-04   0.867  0.38612
## nbDorsoduro           -5.469e-03  1.213e-03  -4.510 6.63e-06 ***
## nbGiudecca             5.645e-03  1.900e-03   2.971  0.00298 **
## nbLido                 9.839e-03  1.583e-03   6.217 5.50e-10 ***
## nbMurano               8.427e-03  3.166e-03   2.662  0.00780 **
## nbSan Marco           -1.054e-02  1.076e-03  -9.802  < 2e-16 ***
## nbSan Polo            -5.509e-03  1.323e-03  -4.162 3.20e-05 ***
## nbSanta Croce         -3.458e-03  1.308e-03  -2.643  0.00823 **
## room_typePrivate room  2.556e-02  8.290e-04  30.832  < 2e-16 ***
## room_typeShared room   7.447e-02  3.497e-03  21.296  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02276 on 4896 degrees of freedom
## Multiple R-squared:  0.3139, Adjusted R-squared:  0.3121
## F-statistic: 172.3 on 13 and 4896 DF,  p-value: < 2.2e-16
```

Due to the transformation of price most of the interpretability of the coefficients on price is gone, but we can still interpret their effect on the transformed price. We can see, for example, that of all the Neighborhood dummy variables, San Marco has the most negative coefficient. This means that on average San Marco has the lowest transformed price, if the remaining variables remain constant. Since the transformation on price is the reciprocal of price, this result translates to a Neighborhood with higher prices on average. For example, Lido has the most positive coefficient, so we would expect it, on average, to have the lowest prices.

This observation is also backed up by the ANOVA presented in the previous section. Furthermore, we can observe this affect on neighborhood in the scatterplots of the predicted price vs. an other covariate. In particular, a scatterplot of Predicted Price vs. Reviews Per Month we can see a clear stratification of the prices by neighborhood:



San Marco in blue has a very high price independent of the co-variate, while Lido in green has a lower price.

It is also important to note that the highest price estimated by the model is of $190.66, which is lower than 15.21% of the original prices. In other words, the model is clearly underestimating the prices although not to a great extent.

Conclusion

Until now pirates in Venice, Italy had no data driven way to determine the optimal houses to pillage based on Airbnb listings. In order to help these pirates we identified differences in pricing among the Venetian neighborhoods, highlighted terms in the user reviews that could distinguish the neighborhoods, as well as built a model that could predict prices for new listings.

In particular, we demonstrated that San Marco and Dorsoduro tend to have higher prices than the rest of the neighborhood. Therefore pirates that are interested in expensive houses should plan to visit those neighborhoods. Conversely, pirates interested in cheap accommodations should travel elsewhere.

Furthermore, we were able to build a model to predict the neighborhoods in Venice based on the textual reviews. An important side-effect of this analysis is that it revealed important landmarks in the various neighborhoods. For example, pirates short on weapons may want to visit the Venetian Arsenal in Castello to re-stock their supply.

Finally, our pricing model will be indispensable to pirates who want to forecast their costs while staying in the area. They can now consider the effect on price a listings availability, neighborhood choice, room type, number of reviews, and number of reviews per month will have before booking a stay.

In summary, we believe our insights will be indispensable for pirates visiting Venice, and we hope that our analysis will provide insights to friendly pirates everywhere!