

What Best Predicts Airbnb Pricing in New Orleans?

J. Walker Blackston, MSPH

7/7/2019

est. reading time: 15 minutes. For the technically incurious, plots and a main take-aways will be provided at the bottom.

Having now lived in New Orleans for the past year, several things have become apparent:

- 1) The heat is as advertised.
- 2) The people are as advertised.
- 3) Be careful about the company in which you mention an “Airbnb” - *whether you are on the business end or renting*

Regardless, I love this place and it’s clear that many of our millions of yearly visitors do as well. I am not here to provide support for our current zoning or short-term rental policies. Like it or not, Airbnb is here to stay. **Plus, CNN Money rated us as one of the “worst cities for renters in the United States,” so we should be arming ourselves with as much data as possible here.** https://money.cnn.com/2015/03/16/real_estate/cities-highest-rent/index.html

For my purposes, this analysis will only evaluate pricing for New Orleans where a friend is considering renting out several units. **Finding the “ideal” pricing for these units, would be fundamental to getting and securing clients.**

The main purpose of this analysis is simple: *how can we model optimal pricing across different types of Airbnb rentals in Nola?*

Good ol’ Fashioned Linear Regression:

Despite the appeal of sexier approaches, the regression remains one of the most popular approaches to answering research questions (citation). With this in mind, I wanted to see what combination of factors most predicted price, a continuous outcome, for selected Airbnb rentals in New Orleans, Louisiana. Here goes nothin’!

Import all data and relevant packages:

note: I will be hiding excessive code (e.g. loading-in packages) for presentation wherever possible, but will provide a footnote .txt file of my complete code for the more technically curious

All data were obtained from “<http://insideairbnb.com/get-the-data.html>”

The data:

Looks like we have a lot of variables (features, to the ML folks), to deal with here - let’s see which we can just eliminate via good ol’ fashioned eyeball test.

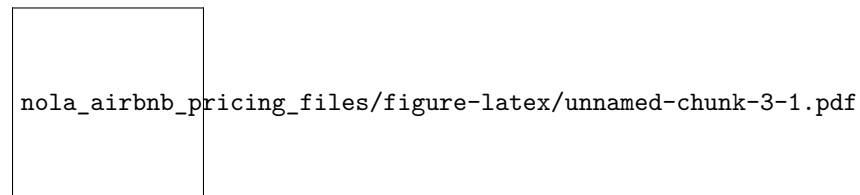
For our purposes, it is not going to be useful to keep any URL’s, ID’s, notes, names, streets, or further location information beyond neighborhood. This leaves us a few variables that could likely impact price (shown in the code below):

```
keep <- c("host_listings_count",
          "neighbourhood_cleansed", "room_type", "accommodates", "bathrooms", "bedrooms", "beds", "price",
          "cleaning_fee", "minimum_nights", "maximum_nights", "number_of_reviews", "review_scores_rating")
df_nola <- listings[keep]
```

Let's visually inspect some of them.

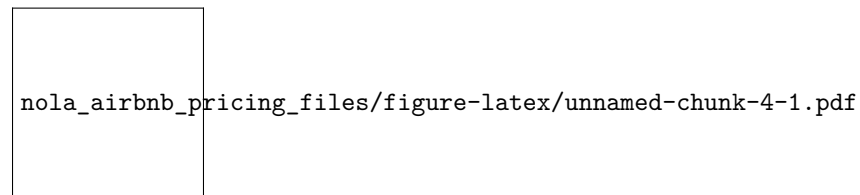
- Accommodations/Recommended Guest Count:

I noticed a few demo projects online predicting Airbnb pricing, and this 'accommodates' variable appears in all of them. It makes sense, logically, that the more people your unit could hold, the more you should charge.



notice that the y-axis is a little funky, as my girlfriend points out, lovingly. This is because we have funky signs in front of our data that confuses our ggplot function. Let's get rid of them in the next line of data management.

- Beds:



From this, we notice a few things. Prices for accommodations do not increase with linearity (or continuously as accommodations increase), but rather, cluster around what is likely the most common accommodation count. Also, our data looks a little funky when we look at the number of beds for some listings.

Yeah, let's clean this up. We need to remove listings with more than 10 beds... because that seems ridiculous and also, like, a *hotel*. We should also remove the \$ sign for price and cleaning fees to help with our analysis and presentation. Our audience can assume we are dealing in U.S. dollars.

```
df_cleaned <- subset(df_nola, df_nola$beds < 7)
df_cleaned$price = as.numeric(gsub("\\$", "", df_cleaned$price)) #converts price to numeric
df_cleaned$cleaning_fee = as.numeric(gsub("\\$", "", df_cleaned$cleaning_fee))
df_cleaned <- subset(df_cleaned, df_cleaned$price != "NA") #remove n=113 missing prices
```

This cleaned a total of 285 listings from our data set. No fear, we are still sufficiently powered with a few thousand listings remaining for further analysis. That said, what are the variables we should focus on that may or may not pass the eyeball test?

Before we jump right into our models, however, it would be useful to impute data for review scores. We can justify this move since its missingness represents less than 1% of the total sample size, but as a valuable predictor in a short list of predictors, we would like to be able to employ a complete case analysis.

Imputation:

A basic rundown of the imputation process: 1) Generate a random list of NA's that comprise 10% of our sample size, 2) Initialize a temporary data set for our imputation package, 'mice', to generate values and store, and 3) compile our final data set from one of the 5 generated/imputed data sets.

For more technical details on 'mice' or the mathematical support for imputation please see footnotes.

nola_airbnb_pricing_files/figure-latex/unnamed-chunk-8-1.

Check how this impacted our bed-price visualization:

Notice how this distribution differs from Figure 1. It's been normalized through a combination of imputation and trimming strange bed totals. Upon further reflection, however, we will note in constructing our models that beds and bedrooms are nearly perfectly correlated. This suggests that we use one or the other. Using both might lead to overfitting. Our final models will reflect this choice.

nola_airbnb_pricing_files/figure-latex/unnamed-chunk-9-

And finally, how our variables are correlated with price:

Ignoring the obvious correlations, we should notice a few interesting things happening. Review scores for the listing (averaged rating across all metrics rating the stay, then scaled to a number out of 100) *do not seem to correlate with price in any significant capacity*. Also, bathrooms and the number of other listings seem to moderately correlate with changes in price. Let's include all of these variables in our modeling efforts.

Linear Models:

The basic form of our first model's equation will be:

$$Price = \alpha + \beta_{Accommodates} + \beta_{Beds} + \beta_{Bedrooms} + \beta_{Bathrooms} + \beta_{HostListingsCount} + \beta_{CleaningFee} + \beta_{ReviewScores}$$

Let's fit this model on our data and interpret (or don't!) our beta estimates/global model:

```
##
## Call:
## lm(formula = compl_df$price ~ compl_df$accommodates + compl_df$beds +
##      compl_df$bedrooms + compl_df$bathrooms + compl_df$host_listings_count +
##      compl_df$cleaning_fee + compl_df$review_scores_rating, data = compl_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -755.61  -55.46  -15.51   32.27  882.69
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.804e+02  2.431e+01  -7.420 1.32e-13 ***
## compl_df$accommodates    1.399e+01  1.473e+00   9.496 < 2e-16 ***
## compl_df$beds           -3.188e+00  2.081e+00  -1.532  0.126
```

```
## compl_df$bedrooms      1.252e+01  3.064e+00  4.086 4.43e-05 ***
## compl_df$bathrooms     2.992e+01  2.842e+00 10.527 < 2e-16 ***
## compl_df$host_listings_count 1.738e-01  5.862e-03 29.648 < 2e-16 ***
## compl_df$cleaning_fee   7.136e-01  4.043e-02 17.649 < 2e-16 ***
## compl_df$review_scores_rating 1.808e+00  2.511e-01  7.197 6.83e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 126.1 on 6556 degrees of freedom
## Multiple R-squared:  0.3487, Adjusted R-squared:  0.348
## F-statistic: 501.5 on 7 and 6556 DF,  p-value: < 2.2e-16
```

Before interpreting anything, we need to see if our model is ‘globally significant’ at $p < .05$. In our F-statistic, it appears so ($p < .001$). Now, in the ‘Pr(>|t|)’ column, we find p-values assessing the t-value for each parameter of interest. The null hypothesis here is that the parameter’s distribution does not significantly differ from the standard t-distribution with mean of zero. Again, more technical details available below for the interested technical parties. In short, all predictors can be interpreted *except* for number of beds. Interesting. Now, we should caution that our model R-squared (or its ability to explain variation present in the data) is 0.35 or about 35%. This isn’t great, but not terrible. Some respected findings in various industries have been built on models with an R-square of 0.20 or lower.

Also, we should notice something: review ratings, why include them? As a first pass, this model is fine, but if we look again at our correlation plot, reviews (and cleaning fees) should not be included according to their low baseline correlation with our outcome, price. We will trim our model to include only variables with higher correlations: Accommodates ($r = 0.42$), number of concurrent listings for the host ($r = 0.29$), number of beds ($r = 0.35$), number of bedrooms ($r = 0.41$), and bathrooms ($r = 0.30$). We will remove beds because beds and bedrooms so strongly correlate and may result in overfitting.

This model would generalize to:

$$Price = \alpha + \beta_{Accommodates} + \beta_{Beds} + \beta_{Bedrooms} + \beta_{Bathrooms} + \beta_{CleaningFee} + \beta_{HostListingsCount}$$

When fit, this equation would look like this:

```
##
## Call:
## lm(formula = price ~ accommodates + bedrooms + bathrooms + cleaning_fee +
##     host_listings_count, data = compl_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -757.92  -55.93  -15.68   32.77  890.36
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -8.201781   3.979865  -2.061  0.0394 *
## accommodates  12.454946   1.309354   9.512 < 2e-16 ***
## bedrooms      12.387395   2.953408   4.194 2.77e-05 ***
## bathrooms     30.646412   2.850814  10.750 < 2e-16 ***
## cleaning_fee   0.703139   0.040411  17.400 < 2e-16 ***
## host_listings_count 0.170204  0.005857  29.058 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 126.6 on 6558 degrees of freedom
```

```
## Multiple R-squared:  0.3433, Adjusted R-squared:  0.3428
## F-statistic: 685.8 on 5 and 6558 DF,  p-value: < 2.2e-16
```

Aside from these model diagnostics, looks like our RMSE (root-mean squared error, details in the footnotes)
=

```
## [1] 126.5179
```

***Important caveat:* We cannot, in any way, derive any causality from these findings. This was merely hypothesis generating, and an exploration of the data, and training exercise.**

But let's make some simple predictions. You just spiffed up your place and want to rent it out. Maybe you will be spending the summer in the Mediterranean or somewhere offshore...

Our model for pricing would therefore be (including coefficients):

$$Y = 13.1X_i + 19.8X_j + 14.4X_k + 0.64X_l + 0.18X_m$$

letting i = accommodates, j = bedrooms, k = bathrooms, l = cleaning fee multiplier, and m = no. of listings for current host

###If you were renting out a 1 bedroom, 1 bath, which could accommodate a couple of 2, charge a flat \$100 cleaning fee, and had no prior hostings... you could reasonably charge about \$124.58 USD. This breaks with data available from insideairbnb.com, which estimates Nola's average price as \$181.00.

#II. Airbnb Pricing within the Milan Neighborhood:

Now, we want to specifically assess and test our model in a specific borough of Nola- Milan. Here's a Wikipedia link to give you some context: https://en.wikipedia.org/wiki/Milan,_New_Orleans. We can implement the same model specifications and treat the New Orleans overall data as a training set, with this as a test set.

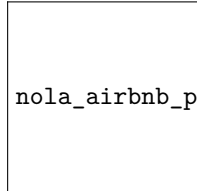
First, some brief data management:

```
##      host_listings_count neighbourhood_cleansed      room_type
##              "numeric"           "character"      "character"
##            accommodates           bathrooms           bedrooms
##              "numeric"           "numeric"           "numeric"
##              beds              price      cleaning_fee
##              "numeric"           "numeric"           "numeric"
##            minimum_nights      maximum_nights      number_of_reviews
##              "numeric"           "numeric"           "numeric"
##      review_scores_rating      instant_bookable
##              "numeric"           "logical"
```

I selected our baseline data set, 'df_cleaned', from before with only the neighborhood values equivalent to 'Milan.' However, we still need to conduct our imputation and compile this subset based on the same procedures and packages from before.

Visualization:

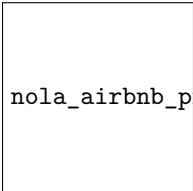
Let's inspect the same variables within our subset to ensure our model can be replicated or 'tested' on these



nola_airbnb_pricing_files/figure-latex/unnamed-chunk-15

data. First, let's produce an updated correlation matrix:

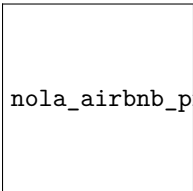
A lot of the same relationships hold: price moderately correlates with expected guest total, or accommodations, ($r = 0.50$), number of bedrooms ($r = 0.55$), bathrooms ($r = 0.62$), but not with host listings ($r = -0.04$) or cleaning fee ($r = 0.33$) as before.



nola_airbnb_pricing_files/figure-latex/unnamed-chunk-16-1.pdf

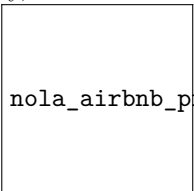
Some strange things happening in our distribution, but the data are what they are. It's likely due to the pure lack of rentals for 1 or 5 people in this specific neighborhood, so on second thought, no big deal.

Let's keep looking.



nola_airbnb_pricing_files/figure-latex/unnamed-chunk-17-1.pdf

And finally, let's look at any relationship with bathrooms, the strongest correlate with price so far in our



nola_airbnb_pricing_files/figure-latex/unnamed-chunk-18-1.pdf

analysis:

Lots of 1 bathroom rentals in Milan, with what seems like the highest prices.

Milan-specific Linear Models:

With significant variables from our second correlation matrix, we will begin to model our pricing data for Milan:

```
##  
## Call:  
## lm(formula = price ~ accommodates + bedrooms + bathrooms, data = milan)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -171.76  -44.24   -6.30   28.45  380.26
```

```
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.054     22.270   0.407  0.68519
## accommodates  -6.531      7.149  -0.913  0.36316
## bedrooms      38.274     13.711   2.792  0.00627 **
## bathrooms     81.034     15.663   5.174 1.17e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 85.13 on 101 degrees of freedom
## Multiple R-squared:  0.435, Adjusted R-squared:  0.4182
## F-statistic: 25.92 on 3 and 101 DF, p-value: 1.624e-12
```

with an RMSE =

```
## [1] 83.48868
```

Our model has improved from our full-city modelling efforts. The adjusted R-squared value improved by 7% and our root mean squared error went from 127.5 to 84.5 on fewer data.

That's just dandy. But what of the parameter coefficients to make pricing *predictions*?

Our final model will take the form:

$$Y = 34.8X_i + 79.9X_j$$

letting i = number of bedrooms, j = number of bathrooms

So, I might recommend my friend, who's considering renting out two 1-bedroom, 1-bathroom units, to charge about: $34.8(1) + 79.9(1) + 77.4$ (average cleaning fee for neighborhood - amenable to host desired fee) = **\$192.10**

Take-aways:

- 1) Within Milan of New Orleans, bathrooms - not number of bedrooms, expected guest total or even your *ratings*- most explained variation in price. For every 1 additional bathroom, hosts charge almost an extra \$80.00 dollars.
- 2) Our milan-specific model improved (in terms of R-squared and model explainability of variance) when given the same parameters, *but* the same variables were not significant/included.
- 3) We have a set of variables/priors to start out with when modeling at prices in other cities! (Future app idea...?)