NYC Airbnb Price Predictions

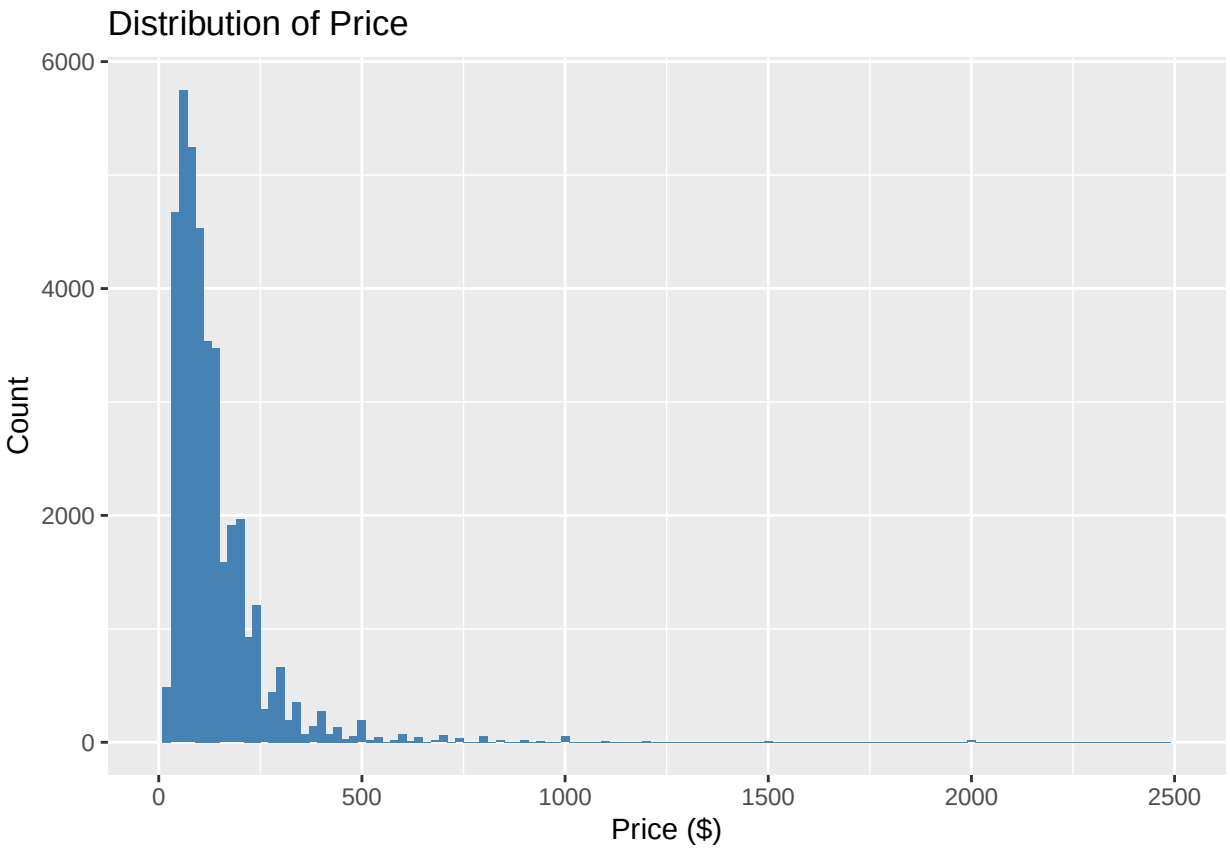
Blues Clues: Alex Shen, Daniel Zhou

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

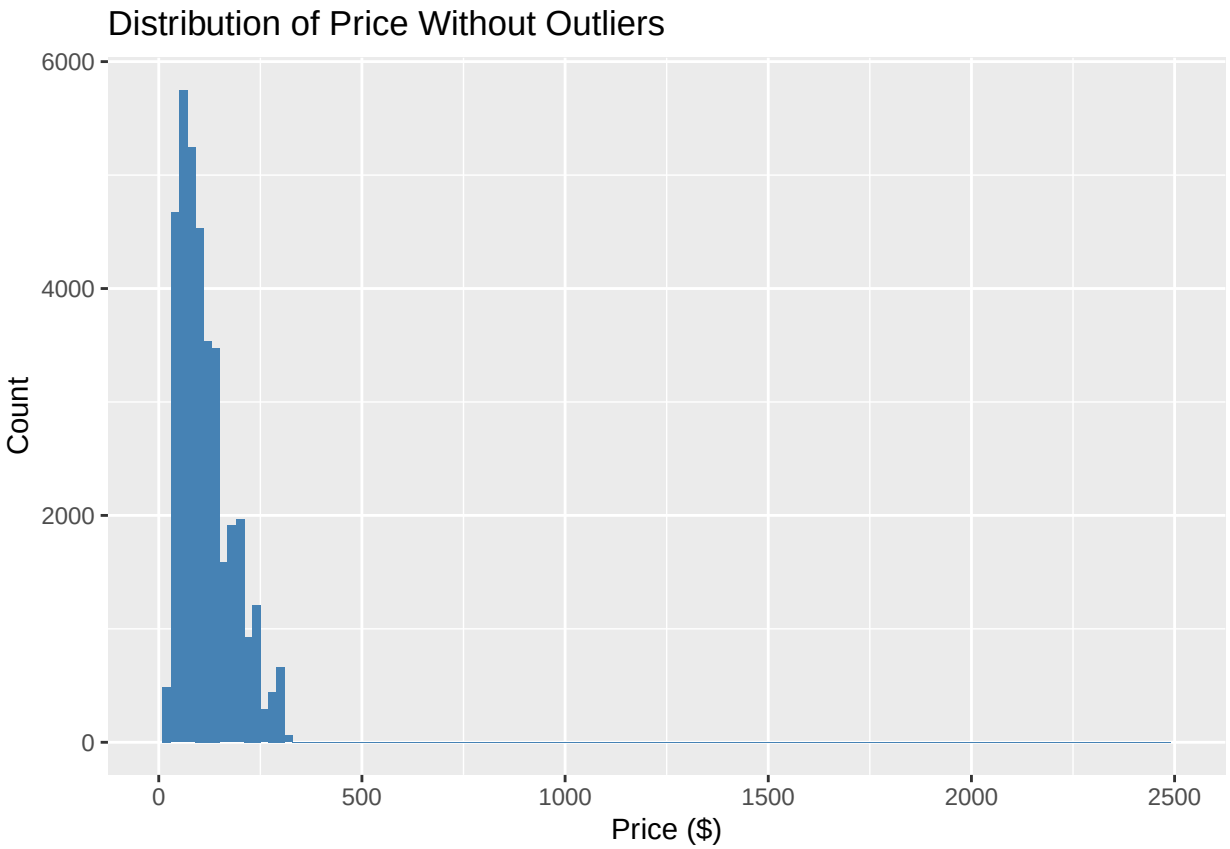
Reading and cleaning data

Our data set was obtained from Kaggle, and contains information about Airbnb listings for NYC in 2019. We want to see what factors can best predict the price of an Airbnb listing in NYC by creating a linear regression model with price as the response variable. The variables that could be used to create a linear regression at first glance are price (presumably per day/night in dollars), neighborhood_group (borough), neighborhood, latitude, longitude, room_type (shared, private, or entire house/apartment), minimum_nights (minimum nights required), number_of_reviews, last_review (date of last review), reviews_per_month, calculated_host_listings_count (number of listings from the same host), and availability_365 (number of days available).

After reading in the data, we do some cleaning necessary to conduct our linear regression. We select the variables that would be useful in conducting regression, make the necessary variables factor variables, and filter for prices greater than 0, since having prices equal to 0 does not make much sense, and would break the code later on.



Here we can see the distribution of price. As we can see, the vast majority of prices are between 0 and 300 dollars, and the data is skewed right with what looks like many outliers visually on the right tail. This makes sense for price to be concentrated at such prices (what average people can afford) and have a few outliers of super luxurious Airbnb's.

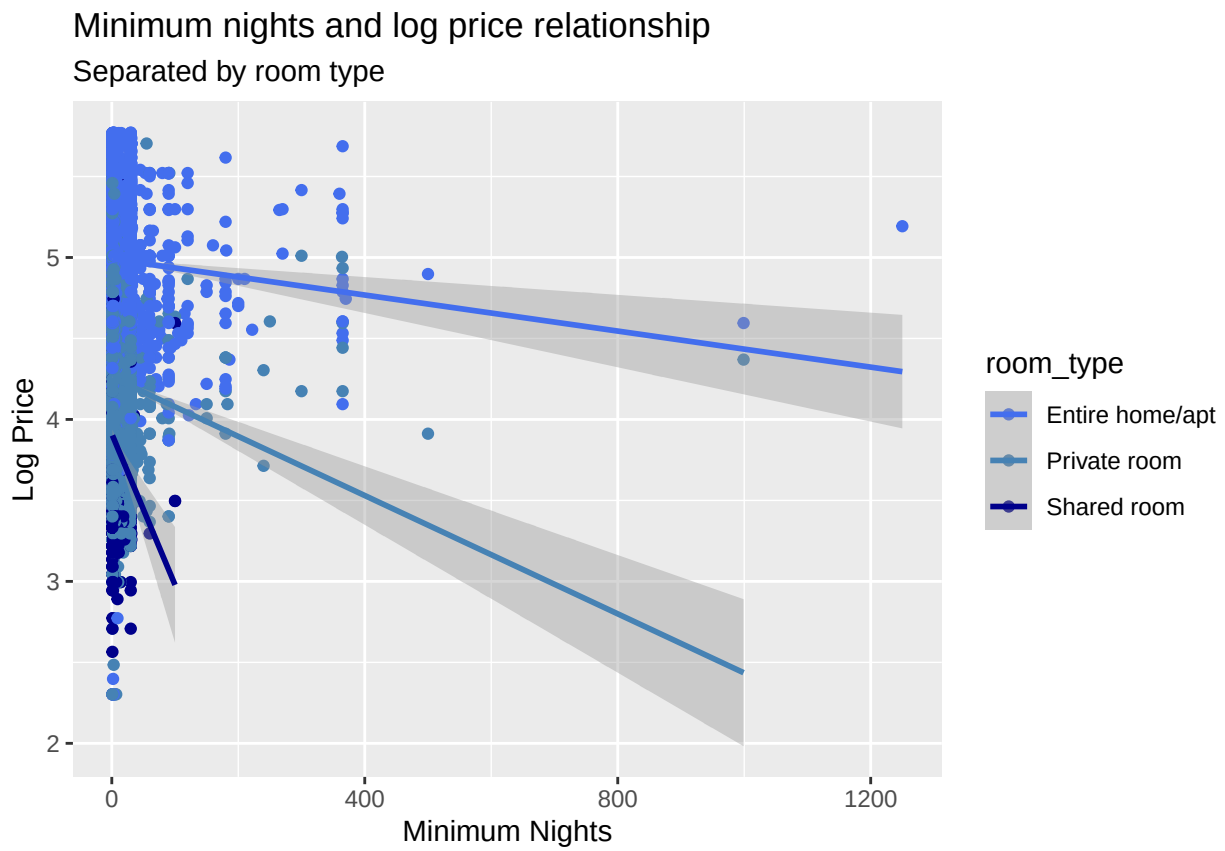


As a result, we eliminate price outliers from the data set. We also log transformed price to prevent negative price predictions from our model (and to help out the constant variance condition, discussed later). If we did not filter for price is greater than 0, this would not be possible.

Creating our Model

First, we wanted to think about what interaction effects we should include in our full model before being backwards selected. We thought that of all the variables, there could be an interaction between minimum nights and room type, so we decided to investigate.

```
## `geom_smooth()` using formula 'y ~ x'
```



Looking at the graph, we can tell that based on the level of room type, the relationship (slope) between minimum nights and log price changes. As a result there is an interaction between room type and minimum nights, so we will add this interaction effect to the full model.

term	estimate	std.error	statistic	p.value
(Intercept)	4.59395	0.01370	335.32928	0.00000
neighborhood_groupBrooklyn	0.25922	0.01346	19.25208	0.00000
neighborhood_groupManhattan	0.51824	0.01351	38.34819	0.00000
neighborhood_groupQueens	0.13641	0.01425	9.57024	0.00000
neighborhood_groupStaten Island	0.00725	0.02535	0.28586	0.77498
minimum_nights	-0.00160	0.00014	-11.43204	0.00000
room_typePrivate room	-0.69678	0.00432	-161.41861	0.00000
room_typeShared room	-1.07940	0.01497	-72.08490	0.00000
reviews_per_month	-0.00258	0.00123	-2.09947	0.03578
calculated_host_listings_count	0.00035	0.00008	4.12515	0.00004
availability_365	0.00040	0.00002	24.27309	0.00000
minimum_nights:room_typePrivate room	-0.00074	0.00025	-2.92974	0.00339
minimum_nights:room_typeShared room	-0.00774	0.00142	-5.43322	0.00000

We chose to use a backwards selection by AIC rather than BIC, because AIC usually lends itself more towards prediction tasks, which is what we are trying to do. We create our full model of all possible variables that we deem useful, then remove each variable that minimizes AIC until it is no longer possible.

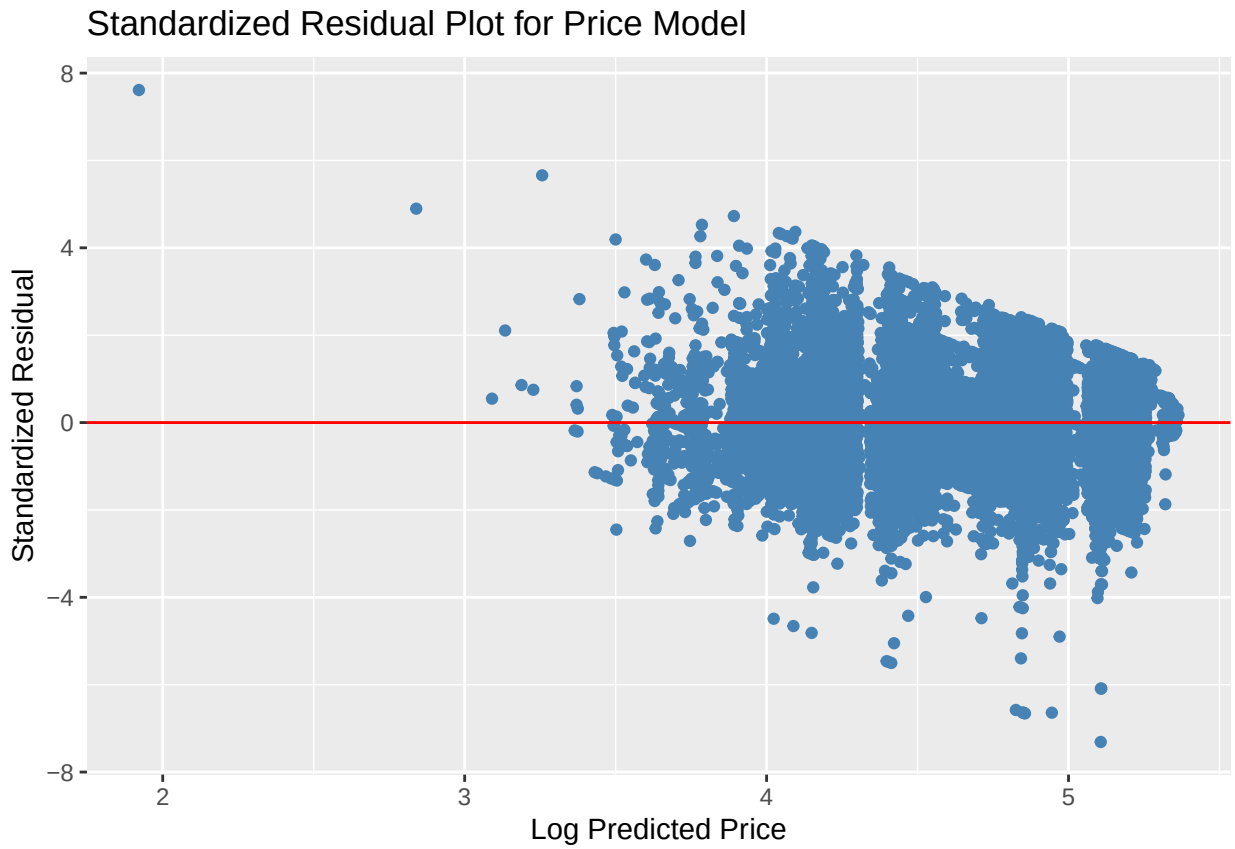
After looking at this model, we see that all of the coefficients have really low p values except for neighborhood_group Staten Island and number_of_reviews. Because neighborhood_group Staten Island is a level of

the neighborhood_group, it means that the price of Airbnb's in Staten Island is not significantly different than those in the base group (Bronx). Because the coefficient is so small and because it is a level of the larger neighborhood_group, we will keep it in the model. For number_of_reviews, the p-value is pretty much equal to our chosen significance cutoff of 0.05, so we decide to keep it in the model.

One thing to note is that our model has a log transformed response variable. So, an increase in one of the dependent variables (with a coefficient B) by 1 means that the predicted median price will be multiplied by a factor of e^B.

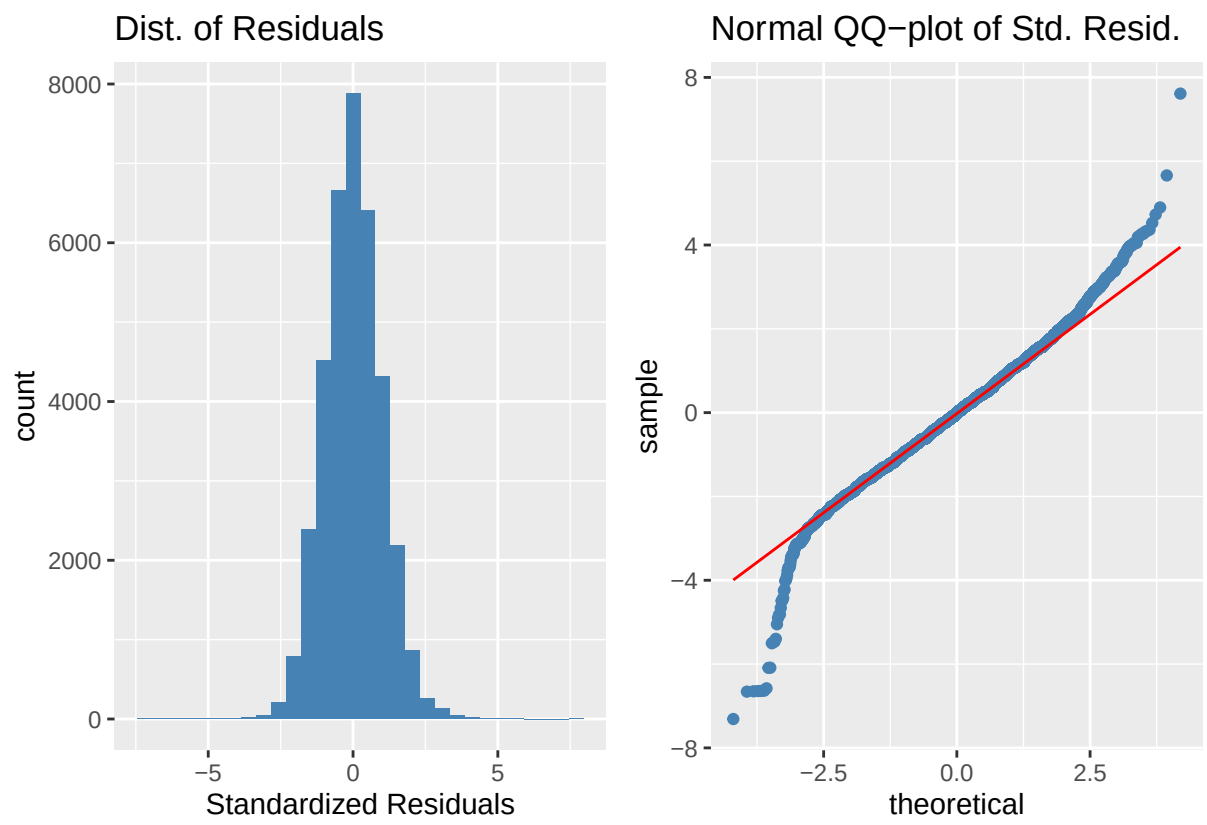
Next, we will check conditions.

Conditions



From our graph, we can see that there is a similar amount of positive and negative residuals, which satisfies the linearity condition. The magnitude of the residuals does seem to change a bit as price increases. However, this was much more pronounced before we did the log transformation. As a result, the constant variance condition is not completely satisfied, but we will proceed with caution.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The normality condition is satisfied because the distribution of the standardized residuals is approximately normal, seen in our histogram and in our normal QQ plot.

Result discussions

Going into the analysis, we did not have too much of an idea of which variables would be influential in predicting the prices of Airbnb's, except that we thought that neighborhood_group and room_type would matter (which was correct).

After looking at the results, most of the results were unsurprising. The base room_type of entire apartment/home usually costs more than a single room, which usually costs more than a double room. The number of listings a host had (calculated_host_listings_count) and the availability of the Airbnb (availability_365) both had small but positive coefficients, showing that as each of them increased generally the price would increase. We found that the the order of boroughs for median predicted price of Airbnb's from most expensive to least expensive is Manhattan, Brooklyn, Queens, Staten Island, and Bronx, with Staten Island and Bronx being very similar. For the minimum required nights, as minimum_nights increased price tended to decrease. This makes sense because the more nights you have to pay for, the lower the price can be (similar to buying things in bulk). Finally, it seemed that the number of reviews seemed to have a slight negative relationship with price. This did not make much sense, because it would seem like a lot of reviews for a host generally means a good Airbnb which can be priced higher. However, a lot of reviews for an Airbnb means that many people were able to stay, meaning the Airbnb might be in the more affordable range of prices rather than being one of the more expensive ones. As a result, after further thought, this result kind of makes sense as well.

We will use this model to create an RShiny app that allows users to enter in the information about an Airbnb and our app will predict the median price.