# Price Prediction of Upper East Side Airbnb Listings

Armale Khan, Haeun Kim, Lakshmipriya Narayanan, Emily Painter

# 1 Introduction to the Data

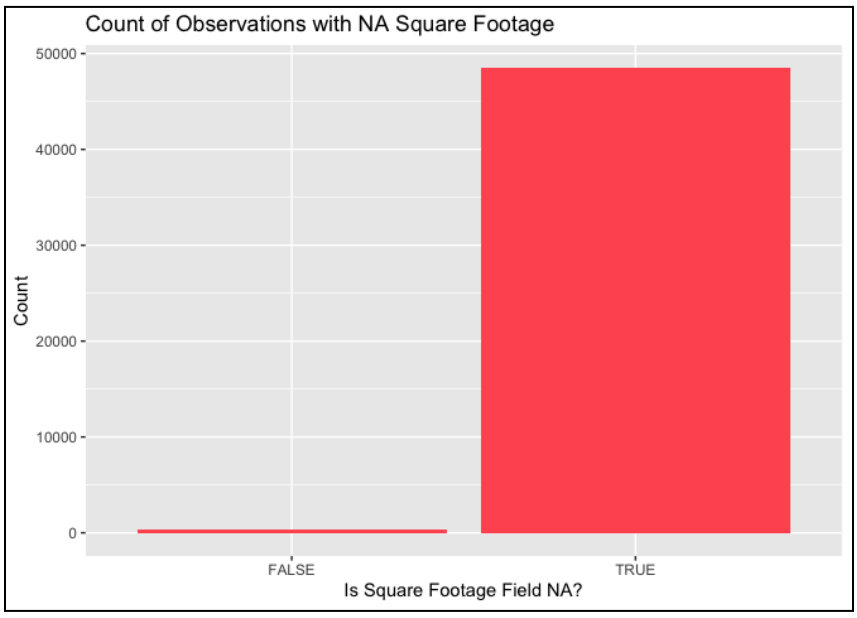
Our dataset, taken from Kaggle [1], records Airbnb listings in New York City from August of 2019. Airbnb stands for 'AIR Bed and Breakfast' and is an American company which allows hosts to rent out their property to travelers [2]. The original dataset contains 48,864 observations and 106 variables, with variables ranging from details of the property itself to metadata about the listing on the Airbnb website. Our analysis goal is to determine which features of an Airbnb listing are most significant in determining the price of a one night stay.

## 1.1 Eliminating Variables and Subsetting the Data

Given the sheer number of features and possible predictors in the raw data, it was unreasonable to create a linear regression model with every possible variable. Before beginning preliminary regression analysis, we needed to eliminate the majority of variables as contenders for regression.

Some variables could easily be removed from consideration as potentially worthwhile regressors. URLs, text descriptions, IDs, and detailed location information—to name just a few examples—could all be removed without hesitation. We also decided early on to consider only continuous regressors and to eliminate boolean and factor type variables. This first step, removing non-numeric variables, took our number of potential regressors from 106 down to 46. Removing uninteresting or unusable numeric variables brought the number of potential regressors from the raw data alone down to 21. Figure 1.1.1 shows the count of observations with NA square footage—just one variable we had to remove despite it seemingly having promise as a regressor.

**Figure 1.1.1 - Count of Listings with Square Footage**



Though at first we hoped our introduction of the `distance` variable (detailed in section 1.2) would allow us to accurately model all 48,864 observations regardless of neighborhood or borough, this was not the case. Our earliest models performed very poorly, with $R^2$ values around 0.2. We hoped limiting the data to a smaller region, rather than the entirety of New York City, would result in a better fitted model. In determining which subset of the data we should use, we created many regression models to see which neighborhoods and zip codes modeled best. We found that the Upper East Side region of Manhattan performed significantly better than all other neighborhoods, only performing worse than our subsets that contained only a single zip code. We chose to use our Upper East Side subset for our regression analysis. This reduced the size of our data from 48,864 observations to 1,685 observations.

Data cleaning after we had subsetted the data down to this more manageable size was relatively straightforward. We coerced variables into appropriate data types, rectified inconsistencies in formatting, replaced NA fields with 0 when appropriate, and removed erroneous entries. Erroneous entries included two observations that were falsely tagged as belonging to a zip code in Upper East Side Manhattan but whose geographic coordinates revealed they were not in Manhattan at all. Finally, we dropped all rows in the dataframe that had any NA fields, which left us with 854 viable observations with which to fit our model.

## 1.2 Feature Engineering

In addition to using many of the naturally occurring variables in the dataset in our regression analysis, we also manipulated some of the non-numeric variables in our dataset to create our own regressors. We also used feature engineering in crafting our response variable.

As briefly alluded to in the previous section, before we had even begun to work with the data, we knew we would need to develop a `distance` heuristic. Since our data spanned all of New York City, we needed a metric to account for the physical location of a listing. Average monthly rent in urban areas is nearly $600 greater than average monthly rent in suburban areas [3], so we anticipated that a rental property in the outskirts of New York City was going to be priced differently than a property located in the heart of Manhattan. To approximate the impact of location, we used the `latitude` and `longitude` fields from the original data to calculate the distance in miles from each listing to Times Square, the most visited tourist destination in the United States [4]. This calculated distance was stored in a new variable called `distance`. Though it did not completely eliminate the issue of location as we hoped, we continued to use the `distance` feature in our regression analysis.

Another potential regressor we engineered from unusable raw data was the `days_hosting` variable, which we derived from the `host_since` variable. After coercing the `host_since` field into a Date object, we created the `days_hosting` variable by subtracting the `host_since` field from 01/01/2022.

The last feature transformation we did to create a potential predictor variable was on the `amenities` field. In the raw data, this variable consisted of a comma separated list of all amenities offered at the given listing. Naturally, since this field was entirely text, it was not useful to us in its original form. We instead created a `num_amenities` field by summing the number of commas plus one for each listing, creating a metric with which we could measure how many amenities were offered for each listing.

We also performed feature engineering on our response variable, `price`. The original data had a price for one night, price for one week, and price for one month field for each listing. We established from the beginning that we wanted to model against the price for one night only, and initially we used that variable in its raw form. Eventually, we realized that minor feature transformation on the response variable would improve our model. We determined that adding the `cleaning_fee` for each listing to the price for one night would be a more accurate representation of the price of staying at the given listing for one night, since the cleaning fee is a flat fee that all guests must pay regardless of the length of their stay. Adding the cleaning fee to the price of a one night stay for the `price` variable was the last feature transformation we performed.

## 1.3 Exploratory Analysis

After completing our data cleaning and before moving to our regression analysis, we wanted to explore some of the trends we could see from the data by comparing each potential regressor against price.

**Figure 1.3.1 - Scatterplots of Price Against Potential Predictors**

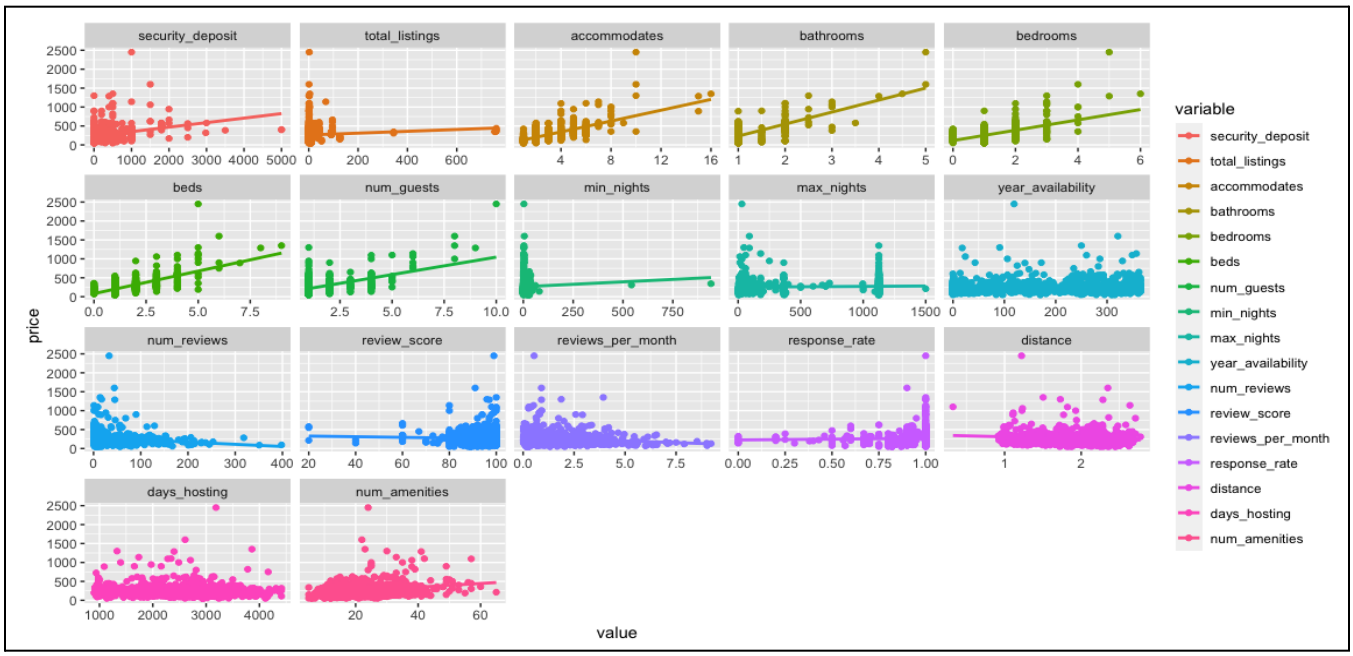
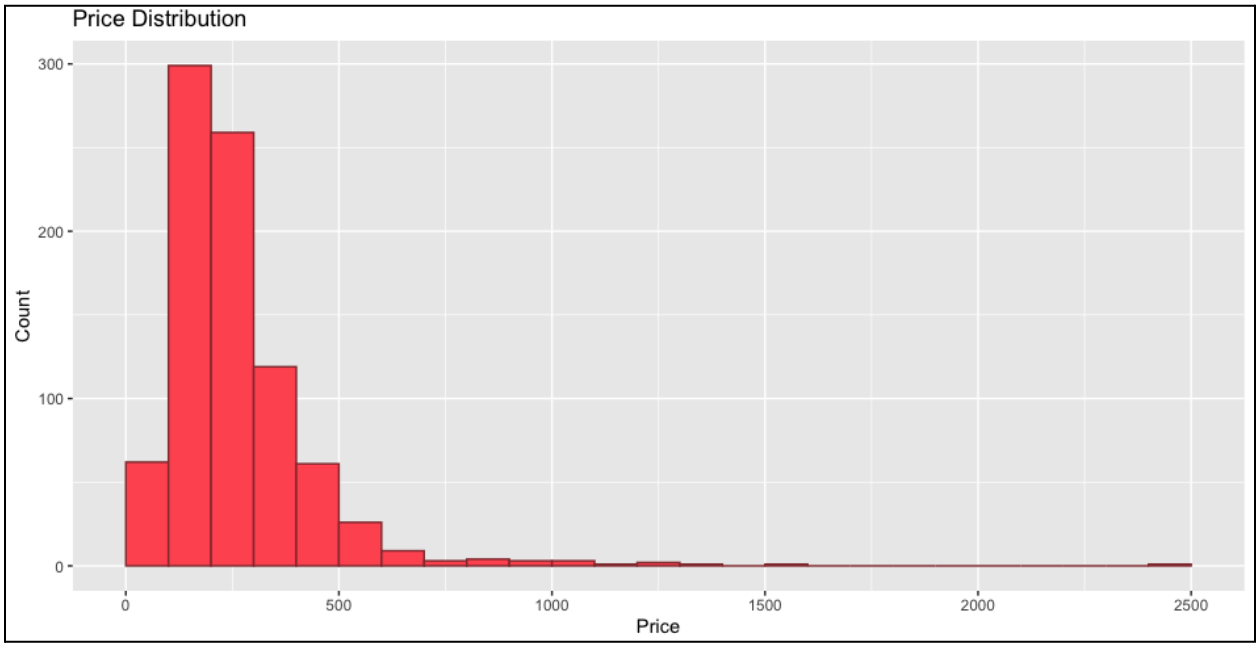


Figure 1.3.1 shows us the relationship between each predictor variable and the response variable `price`. We can see that some variables such as `security_deposit`, `accommodates`, `bathrooms`, `beds`, and `num_guests` show a strong linear relationship with price, while others show a relatively weak linear relationship. From this plot, we began to develop an understanding of the relationships within the data and began to predict which variables would most impact our model.

We also created a histogram of our response variable, `price`, to better visualize the distribution of prices within our dataset.

**Figure 1.3.2 - Histogram of Price for a One Night Stay**




We can see that the price is overall distributed between $0 to $2500, but the majority of the listings are concentrated in the $0 to $500 price range, with very few listings priced over $900. In fact, there are no prices between $1600 and $2400. This gap is very large, and we anticipated this gap could affect our further analysis.

In summary, we attempted to model `price` with the following regressors:

**Table 1.3.1 - Summary of Potential Regressors**

| Variable | Description |
|---|---|
| security_deposit | Security deposit should the guest break or damage something |
| total_listings | Total number of listings hosted by this host |
| accommodates | How many people can be accommodated (number of bed spaces) |
| bathrooms | Number of bathrooms |
| bedrooms | Number of bedrooms |
| beds | Number of beds |
| num_guests | Number of guests allowed |
| min_nights | Average minimum number of nights guest must stay for this booking |
| max_nights | Average maximum number of nights guests are allowed to stay for this booking |
| year_availability | Number of days listing was available in last 365 days |
| num_reviews | Total number of reviews for this listing |
| review_score | Aggregate review rating out of 100 |
| reviews_per_month | Average reviews per month |
| response_rate | Host response rate to inquiring guests |
| distance | Distance in miles from Times Square |
| days_hostings | Number of days each host has been a host on Airbnb as of New Year's Day 2022 |
| num_amenities | Total number of amenities offered at this listing |

# 2 Regression Analysis

After narrowing our data down to a more manageable level, both in number of instances and number of variables, we moved on to performing regression analysis in order to determine which features most influence the price of a one night stay at an Airbnb in Upper East Side Manhattan.

## 2.1 Variable Selection

We utilized the backwards selection approach in choosing which features were most influential in our multiple linear regression model. We first fit a regression model to all 17 potential regressors.

**Figure 2.1.1 - Full Model**

```
Call:
lm(formula = price ~ security_deposit + total_listings + accommodates +
    bathrooms + bedrooms + beds + num_guests + min_nights + max_nights +
    year_availability + num_reviews + review_score + reviews_per_month +
    response_rate + distance + days_hosting + num_amenities,
    data = airbnb)

Residuals:
    Min      1Q  Median      3Q     Max
-388.46  -56.95   -9.75   52.26  966.31

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -1.921e+02  5.168e+01  -3.718 0.000214 ***
security_deposit   7.481e-02  7.352e-03  10.176  < 2e-16 ***
total_listings     1.814e-01  5.659e-02   3.204 0.001404 **
accommodates       2.637e+01  3.644e+00   7.237 1.04e-12 ***
bathrooms          1.820e+02  1.150e+01  15.831  < 2e-16 ***
bedrooms           2.571e+00  6.815e+00   0.377 0.706129
beds               2.568e+01  5.907e+00   4.347 1.55e-05 ***
num_guests         1.743e+01  4.130e+00   4.220 2.71e-05 ***
min_nights        -1.792e-01  9.769e-02  -1.834 0.066966 .
max_nights        -5.599e-03  7.152e-03  -0.783 0.433924
year_availability  1.279e-01  2.898e-02   4.412 1.16e-05 ***
num_reviews       -1.817e-01  9.745e-02  -1.865 0.062600 .
review_score       1.117e+00  4.279e-01   2.609 0.009236 **
reviews_per_month -6.152e+00  3.040e+00  -2.024 0.043335 *
response_rate      1.661e+01  2.417e+01   0.687 0.492165
distance          -2.671e+01  8.235e+00  -3.243 0.001228 **
days_hosting      -5.418e-03  5.275e-03  -1.027 0.304663
num_amenities      1.198e+00  3.982e-01   3.008 0.002709 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 102.2 on 836 degrees of freedom
Multiple R-squared:  0.7034,    Adjusted R-squared:  0.6974
F-statistic: 116.6 on 17 and 836 DF,  p-value: < 2.2e-16
```

From this R output we see that some of our regressors are wholly insignificant. The `bedrooms`, `max_nights`, `response_rate`, and `days_hosting` variables all have p-values above 0.30, which is greater than any significance level $\alpha$ we may choose. We chose the most liberal conventional significance level $\alpha$ = 0.10 and re-fit our model for comparison in Figure 2.1.2.

**Figure 2.1.2 - Reduced Model**

```
Call:
lm(formula = price ~ security_deposit + total_listings + accommodates +
    bathrooms + beds + num_guests + min_nights + year_availability +
    num_reviews + review_score + reviews_per_month + distance +
    num_amenities, data = airbnb)

Residuals:
    Min      1Q  Median      3Q     Max
-388.84  -58.02   -9.11   51.76  967.08

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -1.924e+02  4.577e+01  -4.205 2.89e-05 ***
security_deposit   7.417e-02  7.299e-03  10.161  < 2e-16 ***
total_listings     1.831e-01  5.623e-02   3.256 0.001175 **
accommodates       2.669e+01  3.448e+00   7.740 2.85e-14 ***
bathrooms          1.834e+02  1.100e+01  16.681  < 2e-16 ***
beds               2.650e+01  5.740e+00   4.616 4.52e-06 ***
num_guests         1.754e+01  4.112e+00   4.265 2.23e-05 ***
min_nights        -1.771e-01  9.729e-02  -1.820 0.069137 .
year_availability  1.251e-01  2.844e-02   4.400 1.22e-05 ***
num_reviews       -2.226e-01  8.847e-02  -2.517 0.012035 *
review_score       1.097e+00  4.249e-01   2.581 0.010030 *
reviews_per_month -4.309e+00  2.682e+00  -1.607 0.108500
distance          -2.752e+01  8.158e+00  -3.374 0.000776 ***
num_amenities      1.223e+00  3.949e-01   3.098 0.002013 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 102.1 on 840 degrees of freedom
Multiple R-squared:  0.7026,    Adjusted R-squared:  0.698
F-statistic: 152.6 on 13 and 840 DF,  p-value: < 2.2e-16
```

**Figure 2.1.3 - Reduced Model without Monthly Reviews**

```
Call:
lm(formula = price ~ security_deposit + total_listings + accommodates +
    bathrooms + beds + num_guests + min_nights + year_availability +
    num_reviews + review_score + distance + num_amenities, data = airbnb)

Residuals:
    Min      1Q  Median      3Q     Max
-383.88  -57.99   -9.41   52.20  971.47

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -1.931e+02  4.581e+01  -4.216 2.76e-05 ***
security_deposit   7.631e-02  7.183e-03  10.623  < 2e-16 ***
total_listings     1.890e-01  5.616e-02   3.366 0.000798 ***
accommodates       2.631e+01  3.443e+00   7.640 5.92e-14 ***
bathrooms          1.832e+02  1.100e+01  16.645  < 2e-16 ***
beds               2.696e+01  5.738e+00   4.698 3.07e-06 ***
num_guests         1.764e+01  4.116e+00   4.287 2.02e-05 ***
min_nights        -1.691e-01  9.726e-02  -1.738 0.082525 .
year_availability  1.303e-01  2.829e-02   4.606 4.74e-06 ***
num_reviews       -2.877e-01  7.873e-02  -3.654 0.000274 ***
review_score       1.047e+00  4.242e-01   2.468 0.013793 *
distance          -2.749e+01  8.165e+00  -3.366 0.000797 ***
num_amenities      1.220e+00  3.953e-01   3.088 0.002085 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 102.2 on 841 degrees of freedom
Multiple R-squared:  0.7017,    Adjusted R-squared:  0.6974
F-statistic: 164.8 on 12 and 841 DF,  p-value: < 2.2e-16
```

We first noticed that our adjusted $R^2$ had increased from 0.6974 to 0.698—a small but noteworthy improvement. The second thing we noticed is that the `reviews_per_month` variable no longer registered as significant by our significance level $\alpha = 0.10$. However, upon refitting the model without `reviews_per_month`, our adjusted $R^2$ decreased, as seen in Figure 2.1.3. For this reason, we prefer the reduced model as in Figure 2.1.2, with `security_deposit`, `total_listings`, `accommodates`, `bathrooms`, `beds`, `num_guests`, `min_nights`, `year_availability`, `num_reviews`, `review_score`, `reviews_per_month`, `distance`, and `num_amenities` as our predictor variables for the response `price`. To confirm that none of our four dropped variables—`bedrooms`, `max_nights`, `response_rate`, and `days_hosting`—are significant, we performed an analysis of variance.

**Table 2.1.1 - ANOVA Comparison of Reduced versus Full Model**

| Model | Residual DF | Residual SS | DF | SS | $F_0$ | Pr(>F) |
|---|---|---|---|---|---|---|
| 1 | 840 | 8,755,084 | | | | |
| 2 | 836 | 8,730,831 | 4 | 24,253 | 0.5806 | 0.6768 |

*Model 1: Reduced Model as in Figure 2.1.2*
*Model 2: Full Model as in Figure 2.1.1*

From our ANOVA table we concluded that none of the dropped variables likely contributed significantly to the model, as the p value of 0.6768 is significantly greater than our significance level of $\alpha = 0.10$. We can conclude that our reduced model is the better fit for our data.

With assurances that our reduced model was the better fit for our data over the full model, we checked for any problems with multicollinearity before continuing with our analysis.

**Table 2.1.2 - Variance Inflation Factors**

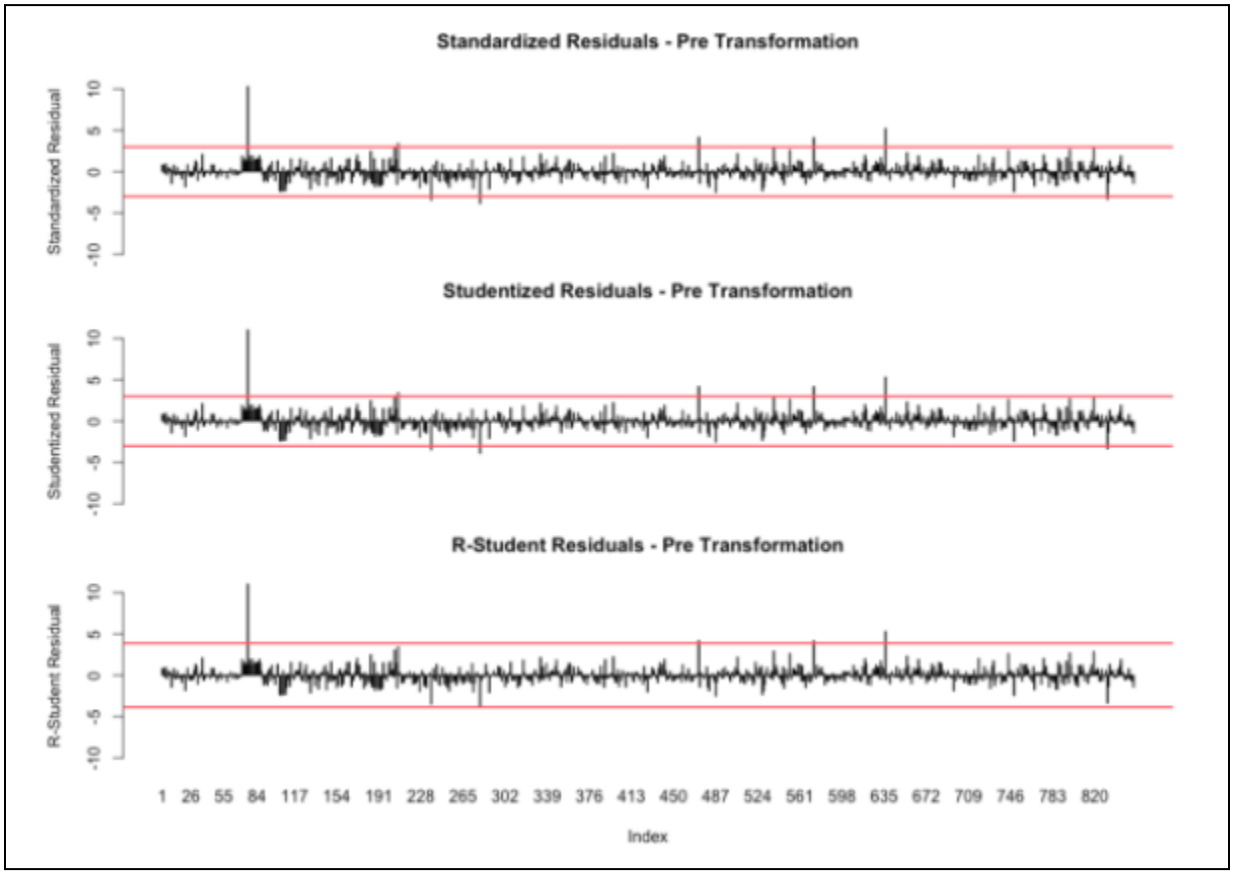
| Variable | VIF | Variable | VIF |
|---|---|---|---|
| security_deposit | 1.194 | year_availability | 1.160 |
| total_listings | 1.184 | num_reviews | 1.391 |
| accommodates | 2.888 | review_score | 1.086 |
| bathrooms | 1.539 | reviews_per_month | 1.427 |
| beds | 2.823 | distance | 1.042 |
| num_guests | 1.681 | num_amenities | 1.110 |
| min_nights | 1.159 | | |

All variance inflation factors are significantly less than ten, so there is no evidence to suggest that our model has a multicollinearity problem. Though we might have expected `beds` and `accommodates` to be collinear with one another, the variance inflation factors tell a different story. This independence makes enough sense once we consider that a bed may accommodate one or two people, so the relationship between `beds` and `accommodates` is not always consistent from house to house. We concluded through the above process that this reduced model is the best fit, and no further predictor variables should be dropped.

## 2.2 Residual Analysis

After concluding that our thirteen remaining regressors are the best variables to use in modeling the data, we conducted a thorough residual analysis. We first plotted the standardized, studentized, and R-student residuals in Figure 2.2.1 to determine if there are any extreme values in our data.

**Figure 2.2.1 - Residual Barplots Pre Transformation**




The three plots as shown in Figure 2.2.1 appear mostly standard with one notable exception—an extraordinary outlier at index 76. This observation has a standardized, studentized, and R-student residual all greater than 10, significantly greater than the cutoffs of 3, 3, and 3.87 respectively. Upon further investigation of the nature of this data point, we found that the listing at this index is over $800 greater in price than the second highest price listing in the data, making it exceptionally unusual in the y space. In addition to this one observation that is particularly noteworthy, we also remark that there are eight other observations that register as being potentially influential according to their standardized and studentized residuals, as well as four other observations that are influential according to their R-student residuals. These potentially influential points may warrant further investigation if they continue to show influence post-transformation, should a transformation prove necessary.

In order to determine if a transformation may be needed, we checked the normality and constant variance assumptions necessary for linear regression. We first plotted a histogram of the residuals in Figure 2.2.2 to see if there are any obvious problems with our normality assumption.

**Figure 2.2.2 - Histogram of Studentized Residuals Pre Transformation**

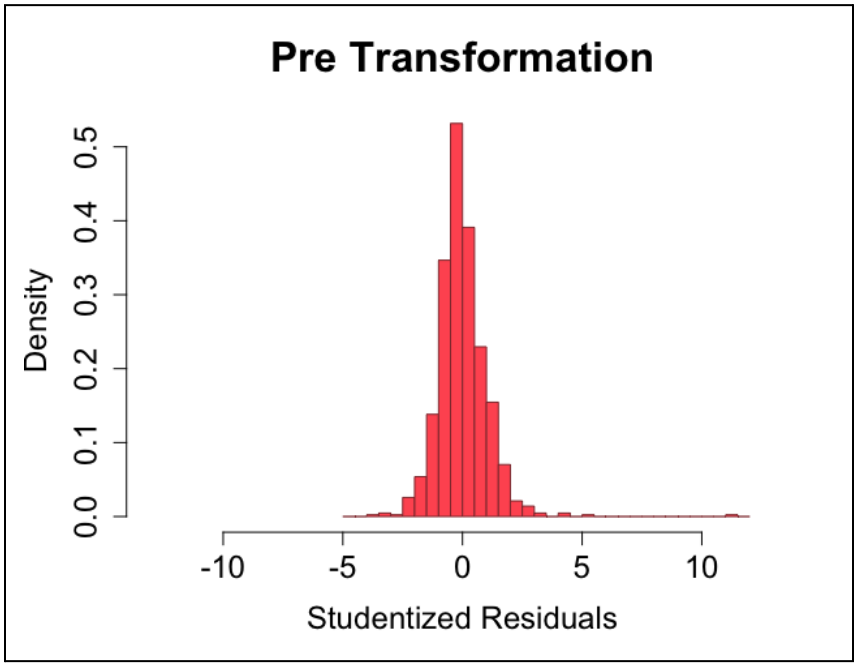
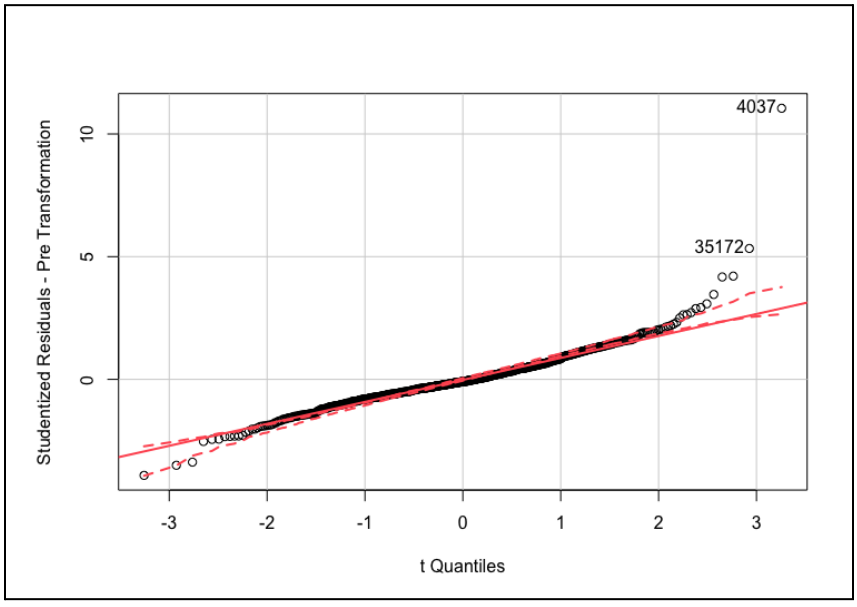



Figure 2.2.2 on its own is not particularly enlightening. We can tell there is a degree of right-skewness in our residuals as we had guessed from Figure 2.2.1, but otherwise, the normality assumption does not initially appear to be violated in any major way. The shape of the distribution is promising, though the spread is larger than we would desire from a perfect normal distribution. We decided to resort to other graphs to better visualize the normality, or lack thereof, of our residuals. We next created a QQ plot to test just this.

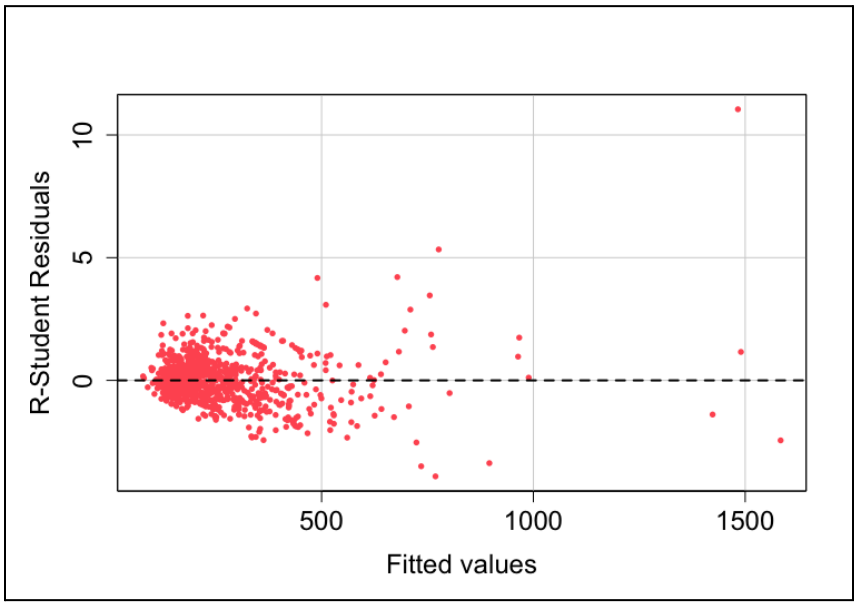
**Figure 2.2.3 - QQ Plot Pre Transformation**



As we can see from Figure 2.2.3, our residuals do not fall along the straight line as closely as they should, indicating our normality assumption is not satisfied. The two most outlying points we can see from the graph are observations 4,037 and 35,172. Observation 4037 is one we're actually already familiar with—it's the same as index 76 from Figure 2.2.1, representing the highest price listing in all of the data. Observation 35,172 is the fourth highest price listing in the data, made more unusual than the second and third highest price data points by its relatively few bathrooms (2.5), few allowed guests (1), and nonexistent security deposit, all of which tend to increase as price increases.

Since our QQ plot did not exhibit perfect linearity, in combination with the wide spread we see in Figure 2.2.2, we suspected a transformation on the response variable price may be necessary. We plotted the residuals against the fitted values in Figure 2.2.4 to see if its shape provided any further insight as to how we should transform our data.

**Figure 2.2.4 - Residuals Against Fitted Values Pre Transformation**




The residuals versus fitted values plot was created in order to analyze equality of variance in our data. From Figure 2.2.4, we clearly observe that the points on the plot do not form a horizontal band around the zero line; instead, they appear to form a funnel. This indicates non-constant variance and must be fixed. To rectify the unequal variance, we considered transformation. Since the variance is an increasing function of the response variable, the appropriate transformation is the square-root transformation, where price will be transformed to square-root of price.

The Box-cox power transformation method corroborated our choice to use the square root method of transformation, as it suggested that the best possible $\lambda$ for transformation is 0.303, which is

sufficiently close to 0.5. We plotted the residuals against the fitted values again post transformation and constructed a new QQ plot to determine the efficacy of our transformation.

**Figure 2.2.5 - Residuals vs Fitted Values Post Transformation**

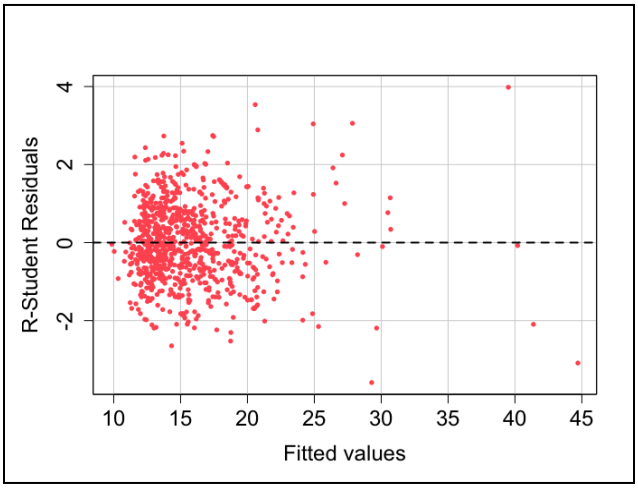
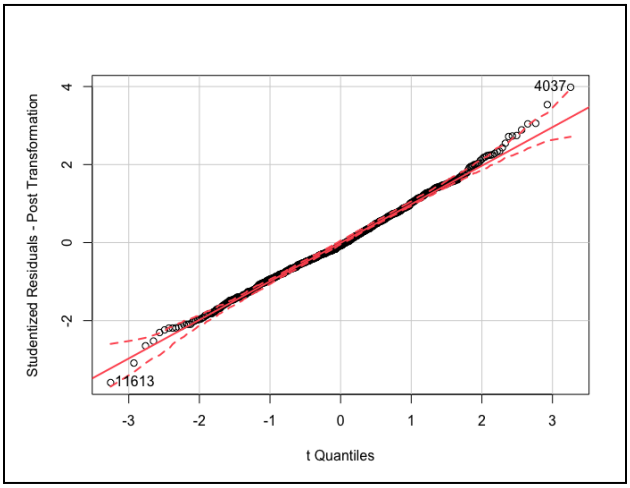
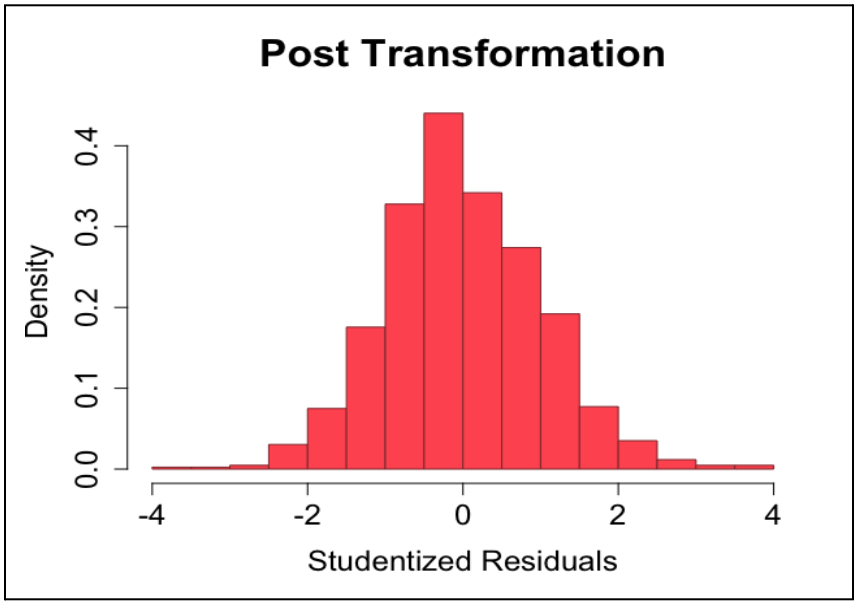



**Figure 2.2.6 - QQ Plot Post Transformation**



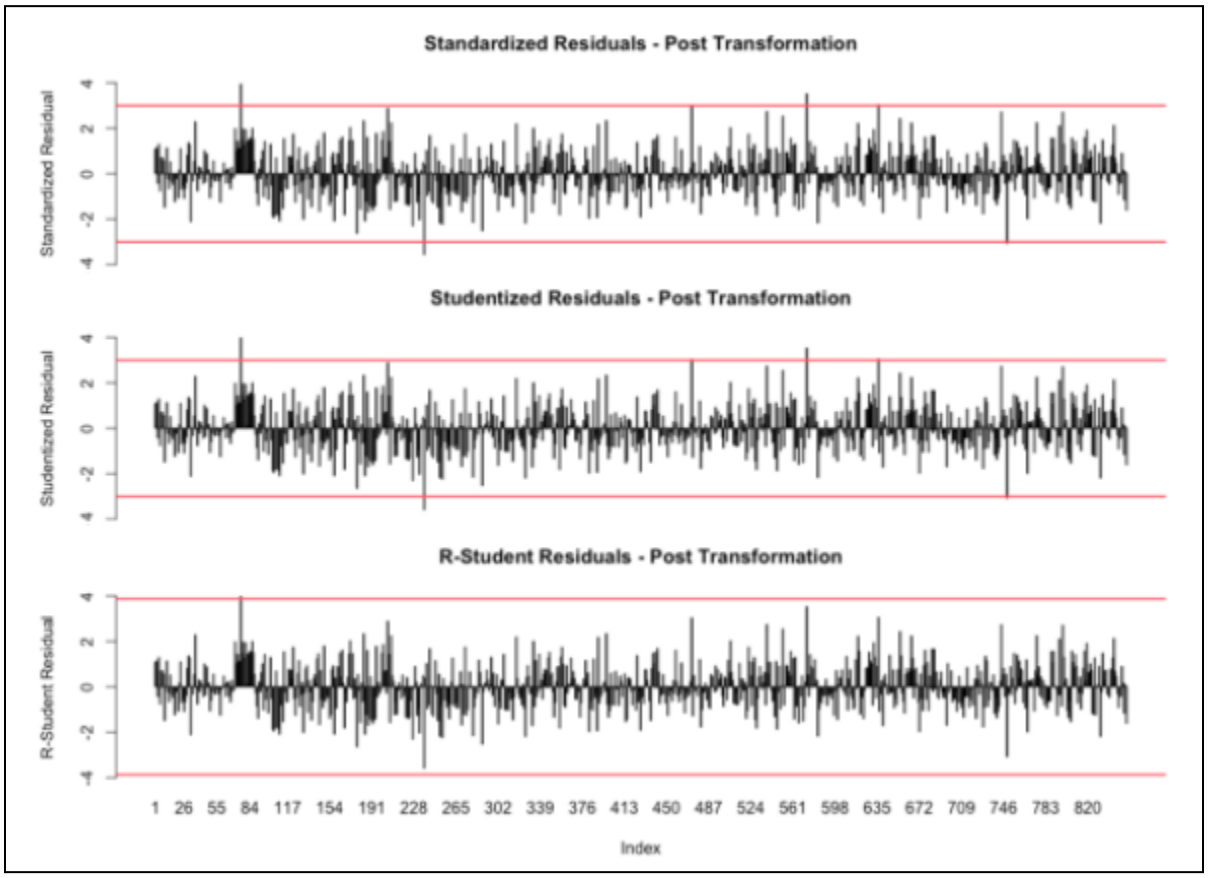

From these two plots we can immediately see a remarkable improvement compared to our residual distribution prior to transformation. In Figure 2.2.5 we see that the residuals now roughly form a horizontal band centered at zero and no longer exhibits the funneling effect we saw in Figure 2.2.4, meaning our constant variance assumption is no longer in violation. In Figure 2.2.6 we see that our residuals much more closely follow the straight line, so our normality assumption is no longer violated. Our histogram of residuals post transformation in Figure 2.2.7 should corroborate this conclusion.

**Figure 2.2.7 - Histogram of Studentized Residuals Post Transformation**



The distribution of residuals as in Figure 2.2.7 does appear more normal than the distribution we saw in Figure 2.2.2. The spread has significantly lessened and the shape remains approximately normal, so we conclude that our square root transformation positively impacted our model and that some of our key assumptions about the distribution of residuals, mainly the constant variance and normality assumption, are no longer violated. We then looked to see how many observations still registered as potential outliers post transformation in Figure 2.2.8.

**Figure 2.2.8 - Residual Barplots Post Transformation**




Index 76 is again marked as a potential outlier for all three residual scaling methods, but it is no longer nearly as far outlying as it was pre transformation. In general, the number of observations that are now considered potential outliers has reduced for all three scaling methods. Where the standardized and studentized residuals once showed nine potential outliers, each now registers only six potential outliers. The potentially influential points according to the R-student residuals dropped from five to only one, index 76.

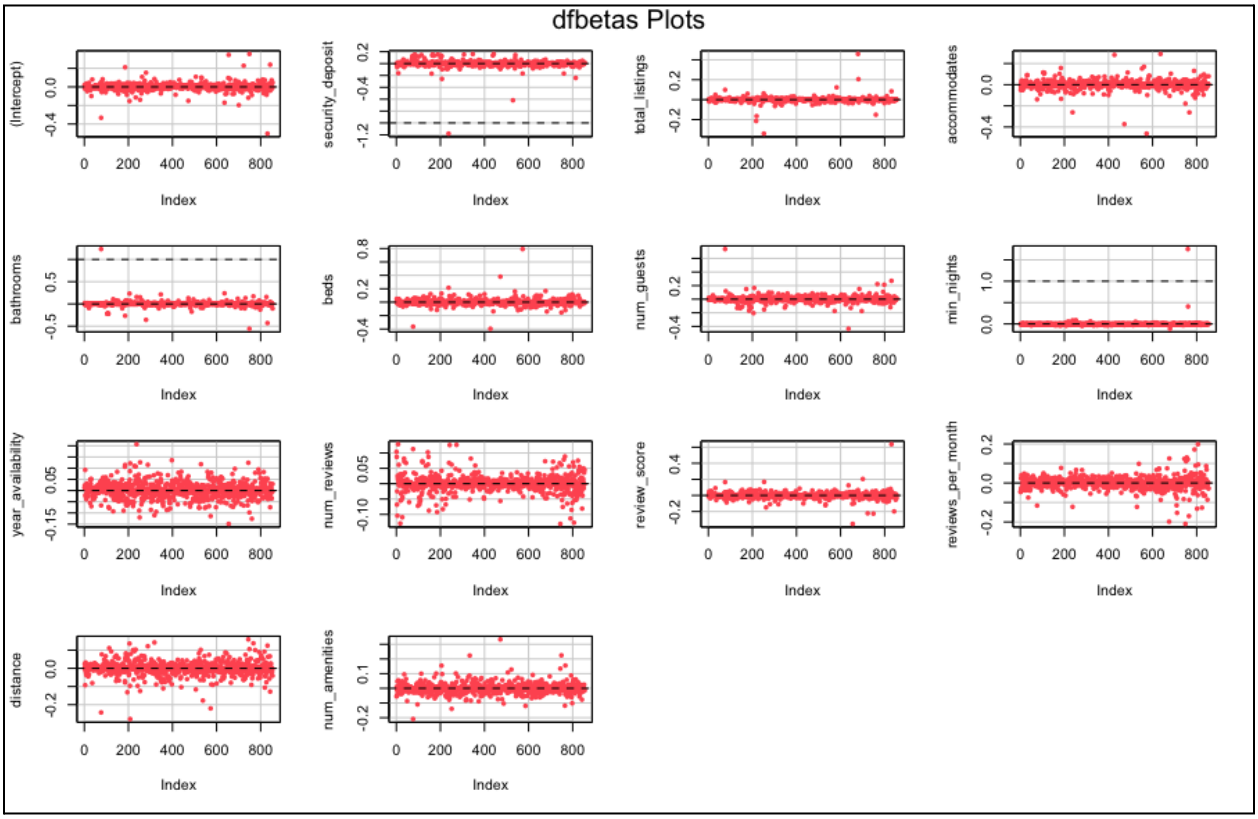
Interestingly enough, our transformed model's adjusted $R^2$ was worse than our regular reduced model—0.650 as opposed to the 0.698 from Figure 2.1.2. This does not suggest any fundamental

problem with our transformation, but it is worthy of noting that the fit of the transformed model was in some regards worse than the fit of the reduced model.

## 2.3 Influential Analysis

Having found our ideal model, we were ready to move to our influential analysis. We had already identified a few points that were possibly influential in the y space, but we had yet to examine any metrics that measure oddness in the X space. To begin with our influential analysis, we tested for any undue influence on each individual regressor using DFBETAS.

**Figure 2.3.1 - DFBETAS plots**




Of these fourteen graphs, only three are noteworthy. The `security_deposit`, `bathrooms`, and `min_nights` graphs each reveal one potentially influential point that is exerting disproportionate influence on the estimate for that regressor, respectively. Naturally, we investigated the nature of each of these three points to determine why these points in particular are so influential, and if they should be removed.

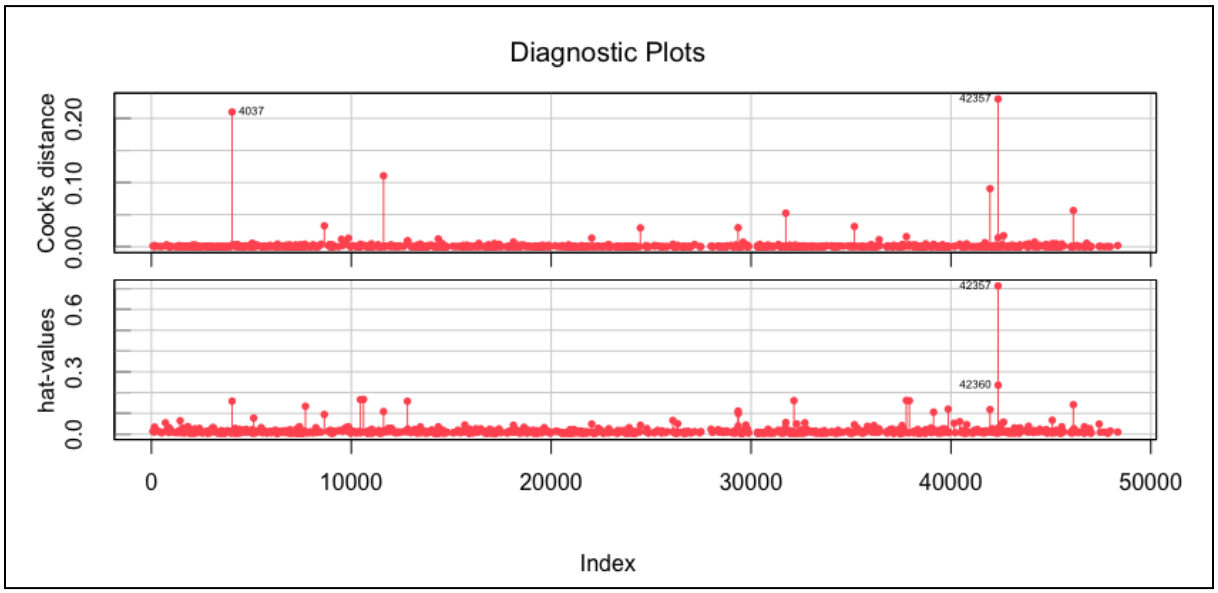
To start with `security_deposit`, we noticed Observation 11,613 exerts great influence. Upon closer inspection, we saw this listing was one of only two with a security deposit of $5,000, the

maximum security deposit in all our data. Naturally, then, this observation exerts great influence on the estimate of the impact of `security_deposit`.

The reasoning behind the influential point for `bathrooms` is similarly intuitive. This point, Observation 4,037, is one we should be overly familiar with by now—it's the same as the extreme outlier we saw in Figure 2.2.1, with a residual greater than 10 prior to transformation. This listing has five bathrooms, tied for highest with only one other listing, and thus exerts great influence not only in the y space, but also in the X space. It is almost expected that the most expensive listing in the data had an outsized influence on at least one of the predictor variables.

Finally for influential points according to DFBETAS, we see that Observation 42,357 exerts great influence over the estimate for the impact of the `min_nights` field. This is perhaps the most interesting influential point of the bunch, but its unusual influence will be more easily explained after we have first examined some of our other influential measures, namely the hat value plot.

**Figure 2.3.2 - Cook's Distance and Hat Values Plots**




A few points are particularly conspicuous amongst these two plots—Observation 4,037, Observation 42,357, and Observation 42,360. We have established repeatedly the abnormalities of Observation 4,037, but we have yet to discuss the unique features of Observation 42,357 (and likewise Observation 42,360, which is very similar to Observation 42,357).

Both observations 42,357 and 42,360 are unusual in the same way—they have extraordinarily high values in the `min_nights` field. To be more precise, Observation 42,357 has the highest average minimum required nights at 941 and Observation 42,360 has the second highest average minimum required nights at 542. For comparison, the median average minimum required nights is 3.

To try to determine why these average minimum nights were so much higher than the other instances, we returned to the original dataframe of all 106 variables to glean as much information as we could about these two listings, especially looking for commonalities between the two that could explain why both listings evidently require average minimum stays of over a year. We found that both of these listings are hosted by the same company—an apartment rental company called Sonder [5]. Rental properties owned by Sonder tend to require longer minimum stays than the typical Airbnb, which could be explained by the fact that Sonder properties are owned by a company rather than an individual, meaning that no individual is kept out of their home when a guest stays at a Sonder property for an extended period of time.

Still, the typical Sonder listing requires minimum stays often in the months—not years. These two observations still stand out as particularly long minimum stays. Looking at these two listings' minimum minimum nights field, we see much more normal values, in the months rather than the years. This suggests that our usage of the average minimum nights field may be responsible for this abnormality, and in the future if we were to perform regression analysis again, using the minimum minimum nights field may be a better choice over the average minimum nights field.

In addition to the three aforementioned points, we found there were 63 listings in total that are potentially exerting an outsized influence on the model.

**Table 2.3.1 - Summary of Influence Measures**

| | COVRATIO | Cook's D | DFFITS | Hat Value | DFBETAS |
|---|---|---|---|---|---|
| **No. of Potentially Influential Observations** | 57 | 0 | 18 | 31 | 3 |
| **Cutoff Value** | <0.95, >1.05 | 0.954 | 0.387 | 0.049 | 1 |

We concluded that none of our potentially influential observations appear to be erroneous, so we chose not to remove the outliers or leverage points. Having completed our residual and influential analysis, we are ready to return to our analysis goal—determining which features are the most impactful in determining the price of a one night stay at an Airbnb in Upper East Side Manhattan.

# 3 Conclusion

After all the required analyses and transformations were performed, we finalized our model from the reduced model

Price = -192.40 + 0.07(**security_deposit**) + 0.18(**total_listings**) + 26.70(**accommodates**) + 183.40(**bathrooms**) + 26.50(**beds**)+ 17.54(**num_guests**) - 0.18(**min_nights**) + 0.13(**year_availability**) - 0.22(**num_reviews**) + 1.10(**review_score**) - 4.31(**reviews_per_month**) - 27.52(**distance**) + 1.22(**num_amenities**)

to the transformed model

Sqrt(Price) = 6.05 + .0020(**security_deposit**) + 0.0051(**total_listings**) + 0.94(**accommodates**) + 2.76(**bathrooms**) + 0.63(**beds**)+ 0.26(**num_guests**) - 0.0044(**min_nights**) + 0.0038(**year_availability**) - 0.0081(**num_reviews**) + 0.0191(**review_score**) - 0.15(**reviews_per_month**) - 0.73(**distance**) + 0.04(**num_amenities**)

Table 3.1 summarizes the contribution of each regressor and its significance by p-value, from most significant to least significant (in the reduced model). Both models are included to give a sense of scale using the regular unit, $, as well as the transformed unit, sqrt($).

**Table 3.1 - Summary of Contribution of Each Regressor**

| Variable | Estimate ($) | P-value (Reduced) | Estimate (sqrt($)) | P-value (Transformed) |
|---|---|---|---|---|
| bathrooms | 183.40 | <2e-16 | 2.76 | <2e-16 |
| security_deposit | -0.07 | <2e-16 | 0.002 | 1.21e-06 |
| accommodates | 26.70 | 2.85e-14 | 0.94 | 0.00731 |
| beds | 26.50 | 4.52e-06 | 0.63 | <2e-16 |
| num_guests | 17.54 | 2.23e-05 | 0.26 | 4.37e-05 |
| year_availability | 0.13 | 1.22e-05 | 0.0038 | 7.95e-07 |
| distance | -27.52 | 0.0008 | -0.73 | 0.0010 |
| total_listings | 0.18 | 0.0012 | 0.0051 | <2e-16 |
| num_amenities | 1.22 | 0.0020 | 0.04 | 3.46e-06 |
| review_score | 1.10 | 0.0100 | 0.0191 | 0.0967 |
| num_reviews | 0.22 | 0.0120 | 0.0081 | 0.0007 |
| min_nights | -0.18 | 0.0691 | -0.0044 | 0.0189 |
| reviews_per_month | -4.31 | 0.1085 | -0.15 | 0.0350 |

From Table 3.1 we conclude that `bathrooms`, `security_deposit`, and `accommodates` are the most important factors in determining the price of a one night stay in an Airbnb in Upper East Side Manhattan. Overall, it seems the features of the property itself are on balance more influential than the metadata about the listing.

# 4 Reflection

Considering our data was not made for regression, we are pleased with our results and overall how our data modeled. We encountered traditional problems in our initial residual plots that were rectified via transformation. Our adjusted $R^2$ was closer to 1 than to 0. Overall, we believe our regression analysis was a success. Still, there were aspects of the project that could have gone smoother.

As previously mentioned, using the average minimum nights field likely influenced the variability in the X space and created more influential points than we would have otherwise had if we had used the minimum minimum nights field. Additionally, we had a major oversight early on in the project—ignoring the `cleaning_fee` and `security_deposit` field. With 106 variables, we had to pick and choose variables to consider sometimes somewhat arbitrarily, leading us to exclude these two fields for weeks. We completed almost all of our analysis before discovering the impact of `cleaning_fee` and `security_deposit`, and we had to redo all of the work we had done. It does not take a leap of logic to conclude that there could have been other features in the dataset that, had we considered them, would have improved our model.

In the future, we could conduct similar analysis on different cities and regions in the United States to compare and contrast which features are more significant in which regions. With more time, we could further investigate potentially influential points and consider the deletion influence of each observation. Further analysis could also consider more regressors, including categorical features.

# 5 Appendices

## 5.1 References

[1] “AB_NY_AUGUST_2019,” *www.kaggle.com*. https://www.kaggle.com/datasets/ybra1993/ab-ny-august-2019 (accessed May 12, 2022).

[2] Erika Rawes and Tyler Lacoma , ”What is Airbnb? What to know before becoming a guest or host”, *Digitaltrends,* 19-Dec-2021. [Online]. https://www.digitaltrends.com/home/what-is-airbnb/

[3] Axiometrics, “How much cheaper are the suburbs?,” *Forbes*, 30-Jun-2016. [Online]. Available: https://www.forbes.com/sites/axiometrics/2016/06/30/how-much-cheaper-are-the-suburbs/?sh=6862776d4761. [Accessed: 10-May-2022].

[4] J. Haverford and Joey Haverford (62 Articles Published) , “The 20 most visited landmarks in the USA (that are actually surprising),” *TheTravel*, 07-Oct-2018. [Online]. Available: https://www.thetravel.com/the-20-most-visited-landmarks-in-the-usa-that-are-actually-surprising/. [Accessed: 10-May-2022].

[5] B. Carson, “Sonder Raises $135 Million To Turn Airbnb-Style Apartments Into A Different Kind Of Hotel,” *Forbes*. https://www.forbes.com/sites/bizcarson/2018/08/23/sonder-raises-135-million-to-turn-airbnb-style-apartments-into-a-different-kind-of-hotel/?sh=74353af5455d (accessed May 12, 2022).

## 5.2 Division of Work

All group members worked together throughout the project. We met weekly and brought to meetings new graphs and new code we had been working on and everyone provided insight on changes that needed to be made in our model. The presentation and report were written collaboratively and we did not assign specific roles to any one group member.

## 5.3 Code

```
library(ggplot2)
library(dplyr)
library(geosphere)
library(reshape2)
library(stringr)
library(car)
library(MASS)
library(psych)
library(tidyr)

# DATA CLEANING

listings <- read.csv("listings.csv")

# Eliminate all non-numeric type variables except for the select few
# non-numeric variables that we intend to manipulate.
airbnb <- dplyr::select(listings, where(is.numeric) |
                        all_of(c('amenities', 'price', 'security_deposit',
                                 'cleaning_fee', 'zipcode', 'host_since',
                                 'host_response_rate')))

# NOTE: host_listings_count and host_total_listings_count are equal
#       for 48,846 observations, so we will only consider
#       host_total_listings_count.

# NOTE: minimum and maximum nights can vary based on condition
#       (higher minimum during busy times, lower maximum during specific
#        seasons, etc) and so there are both typical minimums and maximums
#        and maximum minimums, minimum minimums, maximum maximums, and
#        minimum maximums. Most listings just have one minimum and one
#        maximum, but for the sake of accuracy, we will use the
#        minimum and maximum averages.

# Eliminate all numeric variables that we're not interested in examining.
airbnb <- dplyr::select(airbnb,
                        -all_of(c('id', 'scrape_id', 'host_id',
                                  'host_listings_count', 'maximum_nights',
                                  'minimum_minimum_nights',
                                  'maximum_minimum_nights',
                                  'minimum_maximum_nights',
                                  'maximum_maximum_nights', 'minimum_nights',
                                  'availability_30', 'availability_60',
                                  'availability_90', 'number_of_reviews_ltm',
                                  'review_scores_accuracy',
```

```
                                  'review_scores_cleanliness',
                                  'review_scores_checkin',
                                  'review_scores_communication',
                                  'review_scores_location',
                                  'review_scores_value',
                                  'calculated_host_listings_count',
                                  'calculated_host_listings_count_entire_homes',
                                  'calculated_host_listings_count_private_rooms',
                                  'calculated_host_listings_count_shared_rooms')))

# Show square footage is largely NA and should be removed from the list of
# potential predictors, Figure 1.1.1
ggplot(data = data.frame(is.na(airbnb$square_feet))) +
  geom_bar(aes(x = is.na.airbnb.square_feet.)) +
  labs(title = "Count of Observations with NA Square Footage",
       y = "Count",
       x = "Is Square Footage Field NA?")

# Convert price, cleaning_fee, and security_deposit to numeric type.
airbnb$price <- gsub('\\$|,', '', airbnb$price)
airbnb$price <- as.numeric(airbnb$price)

airbnb$security_deposit <- gsub('\\$|,', '', airbnb$security_deposit)
airbnb$security_deposit <- as.numeric(airbnb$security_deposit)

airbnb$cleaning_fee <- gsub('\\$|,', '', airbnb$cleaning_fee)
airbnb$cleaning_fee <- as.numeric(airbnb$cleaning_fee)

# Standardize the zipcode field.
airbnb$zipcode = gsub('NY|-| ', '', airbnb$zipcode)
airbnb$zipcode = substring(airbnb$zipcode, first = 1, last = 5)
airbnb$zipcode <- as.factor(airbnb$zipcode)

# Create a distance variable that measures the distance in miles from the
# listing to Times Square.
for (i in 1:48864) {
  airbnb$distance[i] <- (distm(c(airbnb$longitude[i], airbnb$latitude[i]),
                               c(-73.985130, 40.758896),
                               distHaversine) / 1609.34)[1,1]
}

# Create a days_hosting variable that measures the number of days each host
# has used Airbnb as a host.
airbnb$host_since <- as.Date(airbnb$host_since)
airbnb$days_hosting <- as.Date('2022-01-01') - airbnb$host_since

# Convert response rate to numeric type. There are many NA values introduced
# during the coercion but all of the NAs are appropriately tagged "N/A" in
```

```
# the original dataset or are entirely blank (so also NA).
airbnb$host_response_rate = gsub('%| ', '', airbnb$host_response_rate)
airbnb$host_response_rate = as.numeric(airbnb$host_response_rate)
airbnb$host_response_rate = airbnb$host_response_rate / 100

# Create a variable to track the number of amenities offered for each listing
airbnb$num_amenities <- str_count(airbnb$amenities, ',') + 1

# Drop the non-numeric variables now that we have derived the information we
# wanted from them, as well as the square footage variable since it is
# largely NA.
airbnb <- dplyr::select(airbnb, -all_of(c('amenities', 'latitude',
                                          'longitude', 'host_since',
                                          'square_feet')))

# Subset the data down to only those listings in Upper East Side Manhattan.
airbnb <- airbnb[airbnb$zipcode == 10065 |
                 airbnb$zipcode == 10128 |
                 airbnb$zipcode == 10075 |
                 airbnb$zipcode == 10028 |
                 airbnb$zipcode == 10021, ]

# Remove two observations with erroneous zipcode tagging, as latitude and
# longitude reveal these two listings are not located in Manhattan.
airbnb <- airbnb[airbnb$distance < 6, ]

# Reassign NAs in the security_deposit and cleaning_fee field to zero.
airbnb$security_deposit <- airbnb$security_deposit %>% replace_na(0)
airbnb$cleaning_fee <- airbnb$cleaning_fee %>% replace_na(0)

# Redefine the price of a one-night stay in the given listing to include
# its cleaning fee.
airbnb$price <- airbnb$price + airbnb$cleaning_fee

# Drop the cleaning fee column.
airbnb <- dplyr::select(airbnb, -cleaning_fee)

# Reorder columns.
airbnb <- airbnb[, c(13, 14, 1, 2, 3, 4, 5, 6, 7, 8, 9,
                     10, 11, 12, 16, 17, 18, 19, 15)]

names(airbnb) <- c('price', 'security_deposit', 'total_listings',
                   'accommodates', 'bathrooms', 'bedrooms', 'beds',
                   'num_guests', 'min_nights', 'max_nights',
                   'year_availability', 'num_reviews', 'review_score',
                   'reviews_per_month', 'response_rate', 'distance',
                   'days_hosting', 'num_amenities', 'zipcode')
```

```
# Remove rows with NA values.
airbnb <- na.omit(airbnb)

# EXPLORATORY ANALYSIS

# Create scatterplots of price against each potential predictor variable
individual_plots <- melt(airbnb[, 1:18], id.vars = 'price', na.rm = TRUE)

# Figure 1.3.1
ggplot(data = individual_plots) +
  geom_jitter(mapping = aes(x = value, y = price, colour = variable)) +
  geom_smooth(mapping = aes(x = value, y = price, colour = variable),
              method = lm, se = FALSE) +
  facet_wrap(~variable, scales = "free_x")

# Plot a histogram of the price variable, Figure 1.3.2
ggplot(data = airbnb, mapping = aes(x = price)) +
  geom_histogram(breaks = seq(0, 2500, 100),
                 color = "#91383b", fill = '#ff5a60') +
  labs(title = 'Price Distribution', y = 'Count', x = 'Price')

# MODELING

# Create our first full linear regression model, Figure 2.1.1
full <- lm(price ~ security_deposit + total_listings + accommodates +
                   bathrooms + bedrooms + beds + num_guests + min_nights +
                   max_nights + year_availability + num_reviews + review_score +
                   reviews_per_month + response_rate + distance + days_hosting +
                   num_amenities, data = airbnb)

summary(full)

# Take only those variables that proved to be significant when tested against
# a significance level of 0.10—Figure 2.1.2
reduced <- lm(price ~ security_deposit + total_listings + accommodates +
                      bathrooms + beds + num_guests + min_nights +
                      year_availability + num_reviews + review_score +
                      reviews_per_month +distance + num_amenities,
              data = airbnb)

summary(reduced)

# Figure 2.1.3
summary(lm(price ~ security_deposit + total_listings + accommodates +
             bathrooms + beds + num_guests + min_nights + year_availability +
             num_reviews + review_score + distance + num_amenities,
           data = airbnb))
```

```
# Compare the two models. The p value is significantly greater than our
# significance level of 0.10, meaning that the additional predictor variables
# of the full model do not improve the precision of our model and that our
# reduced model is a better fit of the data.
anova(reduced, full)

# No evidence of multicollinearity in our model.
vif(reduced)

# RESIDUAL ANALYSIS PRE TRANSFORMATION

# Standardized
range(stdres(reduced))

# Figure 2.2.1
barplot(height = stdres(reduced), names.arg = 1:854,
        main = "Standardized Residuals - Pre Transformation", xlab = "Index",
        ylab = "Standardized Residual", ylim = c(-11, 11))

abline(h = 3, col = "#ff5a60", lwd = 2)
abline(h = -3, col = "#ff5a60", lwd = 2)

# Nine significant data points
sum(abs(stdres(reduced)) > 3)

# Studentized
range(studres(reduced))

# Figure 2.2.1
barplot(height = studres(reduced), names.arg = 1:854,
        main = "Studentized Residuals - Pre Transformation", xlab = "Index",
        ylab = "Studentized Residual", ylim = c(-11, 11))

abline(h = 3, col = "#ff5a60", lwd = 2)
abline(h = -3, col = "#ff5a60", lwd = 2)

# Nine significant data points
sum(abs(studres(reduced)) > 3)

# R-Student
range(rstudent(reduced))

# Figure 2.2.1
barplot(height = rstudent(reduced), names.arg = 1:854,
        main = "R-Student Residuals - Pre Transformation", xlab = "Index",
        ylab = "R-Student Residual", ylim=c(-11, 11))

abline(h = qt(0.10 / (2 * 854), 839, lower.tail = FALSE),
```

```
       col = "#ff5a60", lwd = 2)
abline(h = -qt(0.10 / (2 * 854), 839, lower.tail = FALSE),
       col = "#ff5a60", lwd = 2)

# Five significant data points
sum(abs(rstudent(reduced)) > qt(0.10 / (2 * 854), 839, lower.tail = FALSE))

# Normal Residual Plot pre, Figure 2.2.2
hist(studres(reduced), breaks = seq(-5, 12, 0.5), freq = FALSE, col = "#ff5a60",
     border = "#91383b", cex.axis = 1.5, cex.lab = 1.5, cex.main = 2,
     main = "Pre Transformation", xlab = 'Studentized Residuals', xlim = c(-13, 13))

# QQ Plot pre, Figure 2.2.3
qqPlot(reduced, col.lines = "#ff5a60", envelope = c(style = 'lines'),
       ylab = 'Studentized Residuals - Pre Transformation')

# Fitted against residuals pre, Figure 2.2.4
residualPlot(reduced, type = "rstudent", quadratic = FALSE, col = "#ff5a60",
             pch = 16, cex = 0.75, cex.axis = 1.5, cex.lab = 1.5,
             ylab = "R-Student Residuals")

# Residuals against regressors pre
residualPlots(reduced, type = "rstudent", fitted = FALSE, quadratic = FALSE,
              col = "#ff5a60", pch = 16, cex = 0.75, cex.axis = 1,
              cex.lab = 1.3, ylab = "R-Student Residuals")

# TRANSFORMATION

# Find ideal lambda value according to boxcox power transformation
boxCox(reduced)$x[which.max(boxCox(reduced)$y)]

# Our plot of the residuals against the fitted values showed nonconstant error
# variance, so we will transform the response, price, using the square root
# method.
transformed <- lm(sqrt(price) ~ security_deposit + total_listings +
                                accommodates + bathrooms + beds + num_guests +
                                min_nights + year_availability + num_reviews +
                                review_score + reviews_per_month + distance +
                                num_amenities, data = airbnb)

summary(transformed)

# RESIDUAL ANALYSIS POST TRANSFORMATION

# Standardized
range(stdres(transformed))

# Figure 2.2.8
```

```
barplot(height = stdres(transformed), names.arg = 1:854,
        main = "Standardized Residuals - Post Transformation", xlab = "Index",
        ylab = "Standardized Residual", ylim = c(-4, 4))

abline(h = 3, col = "#ff5a60", lwd = 2)
abline(h = -3, col = "#ff5a60", lwd = 2)

# Six significant data points
sum(abs(stdres(transformed)) > 3)

# Studentized
range(studres(transformed))

# Figure 2.2.8
barplot(height = studres(transformed), names.arg = 1:854,
        main = "Studentized Residuals - Post Transformation", xlab = "Index",
        ylab = "Studentized Residual", ylim = c(-4, 4))

abline(h = 3, col = "#ff5a60", lwd = 2)
abline(h = -3, col = "#ff5a60", lwd = 2)

# Six significant data points
sum(abs(studres(transformed)) > 3)

# R-Student
range(rstudent(transformed))

# Figure 2.2.8
barplot(height = rstudent(transformed), names.arg = 1:854,
        main = "R-Student Residuals - Post Transformation", xlab = "Index",
        ylab = "R-Student Residual", ylim=c(-4, 4))

abline(h = qt(0.10 / (2 * 854), 839, lower.tail = FALSE),
       col = "#ff5a60", lwd = 2)
abline(h = -qt(0.10 / (2 * 854), 839, lower.tail = FALSE),
       col = "#ff5a60", lwd = 2)

# One significant data point
sum(abs(rstudent(transformed)) > qt(0.10 / (2 * 854), 839, lower.tail = FALSE))

# Normal Residual Plot post, Figure 2.2.7
hist(studres(transformed), breaks = seq(-4, 4, 0.5), freq = FALSE, col = "#ff5a60",
     border = "#91383b", cex.axis = 1.5, cex.lab = 1.5, cex.main = 2,
     main = "Post Transformation", xlab = 'Studentized Residuals')

# QQ plot post, Figure 2.2.6
qqPlot(transformed, col.lines = "#ff5a60", envelope = c(style = 'lines'),
       ylab = 'Studentized Residuals - Post Transformation')
```

```
# Fitted against residuals post, Figure 2.2.5
residualPlot(transformed, type = "rstudent", quadratic = FALSE, col = "#ff5a60",
             pch = 16, cex = 0.75, cex.axis = 1.5, cex.lab = 1.5,
             ylab = "R-Student Residuals")

# Residuals against regressors post
residualPlots(transformed, type = "rstudent", fitted = FALSE, quadratic = FALSE,
              col = "#ff5a60", pch = 16, cex = 0.75, cex.axis = 1,
              cex.lab = 1.3, ylab = "R-Student Residuals")

# INFLUENTIAL POINTS ANALYSIS

inf_measures <- influence.measures(transformed)
inf_measures_df <- data.frame(summary(inf_measures))

# Plot dfbetas, Figure 2.3.1
dfbetasPlots(transformed, intercept = TRUE, col = "#ff5a60", pch = 16,
             cex = 0.75, layout = c(4, 4))

# Plot Cook's D and Hat Values, Figure 2.3.2
influenceIndexPlot(transformed, vars = c('Cook', 'hat'), pch = 16,
                   cex = 0.75, col = "#ff5a60", id = list(method = "y", n = 2,
                                                          cex = 0.50,
                                                          col = carPalette()[1],
                                                          location = "lr"))

# p = 14, n = 854, n - p = 840

# How many points have significant COVRATIO values? 57
sum(inf_measures_df$cov.r > (1 + (3 * 14 / 840)) | inf_measures_df$cov.r < (1 - (3 * 14
/ 840)))

# How many points have significant DFFITS values? 18
sum(abs(inf_measures_df$dffit) > 3 * sqrt(14 / 840))

# How many points have significant Cook's D values? 0
sum(inf_measures_df$cook.d > qf(0.5, 14, 840))

# How many points have significant hat values? 31
sum(inf_measures_df$hat > 3 * 14 / 854)
```