Determining a Fair Apartment Price

Stat 410 Final Project

Jack Dzialo, Devin Abraham, Tom Vincent

Purpose

When coming up with ideas for our final project, we wanted to work on something that related to a real problem that we had in our lives. Our whole group is looking for housing next year, both near Rice and out of Houston. Two of us are looking for larger houses to share with roommates, while one is looking for a one 1-2 bedroom apartment in a different city. Since we are all facing the same problem, but are looking at many different housing factors we wanted to come up with a model to ensure we are getting a fair price for the given housing option, or even find below-market opportunities allowing us to get a great deal. Since there are so many houses on the market all with different qualities, it is difficult to determine a fair price.

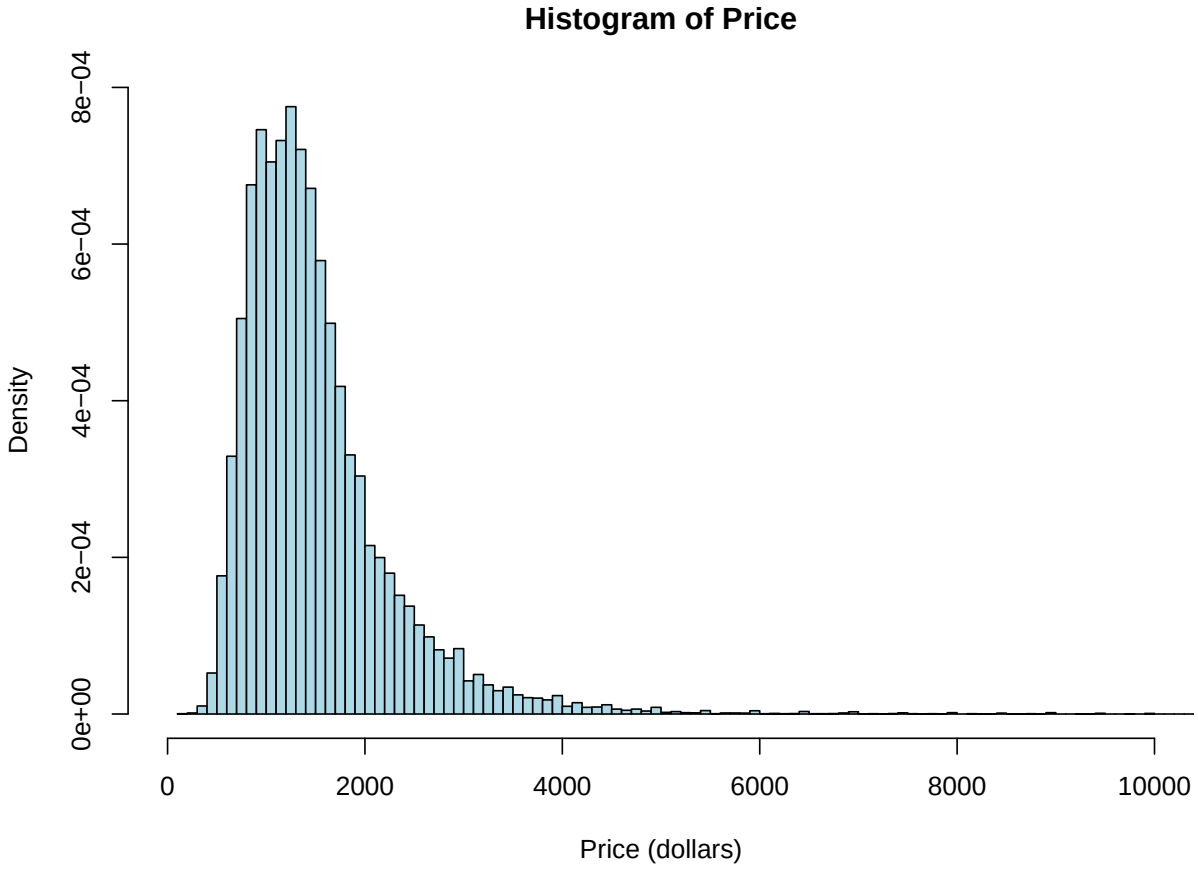
Therefore, for our final project, we aim to address the problem of accurately predicting apartment prices based on a variety of influencing factors. Our primary goal is to develop a reliable regression model that can assess the fairness of rental prices, providing a valuable tool for prospective renters to make informed decisions. The effects of variables such as location, size, amenities, and neighborhood characteristics on rental costs are of particular interest, as understanding these can aid in identifying key drivers of price variations. By achieving these objectives, we hope to contribute to a more transparent rental market.

Definitions

1. AIC: Measures trade off between model fit and complexity with a penalty for more complexity
2. $AICc$: Corrected version of AIC with a larger penalty for regressors
3. BIC: Penalizes model complexity heavily and favors simple model with large sample sizes
4. $PRESS$: Predicted Residual Error Sum of Squares; predicts how accurate a model is at predicting data it hasn't seen before

Data Collection

We got our data from the UC Irvine Machine Learning repository titled Apartment for Rent. In addition, we supplemented this dataset with a US cities dataset from Simple Maps so we could include the population and density of the city the apartment is located in. The Apartment for Rent dataset included 22 variables and 52,746 oberservations. The 22 variables included id, category, title, body, amenities, bathrooms, bedrooms, currency, fee, has_photo, pets_allowed, price, price_display, price_type, square_feet, address, cityname, state, latitude, longitude, source, time. After exploring the data, we found columns with just text such as Title, Body, and Address which we were not able to turn into quantitative data or factors so we decided to not include it in the model. In addition, we did not include the following columns since they were not directly related to the housing. We removed id, category, currency, price_type, state, latitude, longitude, and time. To include the population and density of the city, were able to merge the datasets including the population and density for the listed city in each row. There were a range of responses for pets_allowed, has_fee, and has_photo but we were able to convert them into binary indicator variables 1 for yes, 0 for no.



Data Synopsis for All Variables

	Statistic	Value
1	Mean Price	1522
2	Max Price	52500
3	Min Price	100
4	Median Price	1349

	Statistic	Value
1	Mean Square Feet	954
2	Max Square Feet	12000
3	Min Square Feet	107
4	Median Square Feet	898

	Statistic	Value
1	Mean Bedrooms	2
2	Max Bedrooms	9
3	Min Bedrooms	0
4	Median Bedrooms	2

	Statistic	Value
1	Mean Bathrooms	1.0
2	Max Bathrooms	8.5
3	Min Bathrooms	1.0
4	Median Bathrooms	1.0

	Statistic	Value
1	Mean Population	1220.000
2	Max Population	18908.608
3	Min Population	0.054
4	Median Population	148.879

	Statistic	Value
1	Mean Density	1588.0
2	Max Density	28653.9
3	Min Density	2.2
4	Median Density	1254.2

	Statistic	Value
1	Mean Density	1588.0
2	Max Density	28653.9
3	Min Density	2.2
4	Median Density	1254.2

	Statistic	Value
1	Proportion with Photos	0.911
2	Proportion with fees	0.003
3	Proportion that allow pets	0.024

Data Analysis - Model Building

Forward Stepwise Search

First Iteration

For the first iteration of our stepwise search, we start off with a linear model that only consists of an intercept, β_0.

$$Y = \beta_0$$

After fitting the basic model with every predictor, we find that the Square feet variable results in the lowest P value of the model. We created an R function to run forward stepwise search that inputs the potential variables to be included and the variables already in the model.

Statistic	Values
Predictor	square_feet
P-Value	0
Adjusted R Squared	0.1745
AIC	703090
AICc	703090
BIC	703108
PRESS	34891683588

Forward Stepwise Search

Second Iteration

For the second iteration of our stepwise search, we use the linear model from the previous iteration, being the model with an intercept β_0 and a single predictor square feet, denoted as X_1.

$$Y = \beta_0 + X_1\beta_1$$

After fitting the model with every predictor, we find that the Density variable results in the lowest P value of the model.

	Predictor	P-Value	Adjusted R Squared	AIC	AICc	BIC	PRESS
First Iteration	square_feet	0	0.1745	703090	703090	703108	34891683588
Second Iteration	density	0	0.3183	693058	693058	693084	28826984608
Difference	NA	NA	-0.144	10032	10032	10024	6064698980

Forward Stepwise Search

Second Iteration

The model improved when including density, while $AIC, AICc, BIC, PRESS$ got smaller.

$Model:$

$$Y = \beta_0 + X_1\beta_1 + X_2\beta_2$$

	Predictor	P-Value	Adjusted R Squared	AIC	AICc	BIC	PRESS
First Iteration	density	0	0.3183	693058	693058	693084	28826984608
Second Iteration	population	0	0.3384	691486	691486	691521	27977452423
Difference	NA	NA	-0.02	1572	1572	1563	849532185

Third Iteration

The model improved when including population, while $AIC, AICc, BIC, PRESS$ got smaller. We also decided to divide population by 1000, as the values of population are very high, leading to a very small β value. Dividing by 1000 leads to a higher β estimate, which is easier to understand, as it does not get truncated by functions that display the coefficients and their corresponding important information.

$Model:$

$$Y = \beta_0 + X_1\beta_1 + X_2\beta_2 + X_3\beta_3$$

	Predictor	P-Value	Adjusted R Squared	AIC	AICc	BIC	PRESS
First Iteration	population	0	0.3384	691486	691486	691521	27977452423
Second Iteration	bathrooms	0	0.3425	691161	691161	691205	27811253745
Difference	NA	NA	-0.004	325	325	316	166198678

Fourth Iteration

The model improved when including bathrooms, while $AIC, AICc, BIC, PRESS$ got smaller.

$Model:$

$$Y = \beta_0 + X_1\beta_1 + X_2\beta_2 + X_3\beta_3 + X_4\beta_4$$

	Predictor	P-Value	Adjusted R Squared	AIC	AICc	BIC	PRESS
First Iteration	bathrooms	0	0.3425	691161	691161	691205	27811253745
Second Iteration	bedrooms	0	0.3491	690629	690629	690683	27542867182
Difference	NA	NA	-0.007	532	532	522	268386563

Fifth Iteration

The model improved when including bedrooms, while $AIC, AICc, BIC, PRESS$ got smaller.

$Model:$

$$Y = \beta_0 + X_1\beta_1 + X_2\beta_2 + X_3\beta_3 + X_4\beta_4 + X_5\beta_5$$

	Predictor	P-Value	Adjusted R Squared	AIC	AICc	BIC	PRESS
First Iteration	bedrooms	0	0.3491	690629	690629	690683	27542867182
Second Iteration	has_photo	0	0.3497	690588	690588	690650	27521373719
Difference	NA	NA	-0.001	41	41	33	21493463

Sixth Iteration

The model improved when including has_photo, while $AIC, AICc, BIC, PRESS$ got smaller.

$Model:$

$$Y = \beta_0 + X_1\beta_1 + X_2\beta_2 + X_3\beta_3 + X_4\beta_4 + X_5\beta_5 + X_6\beta_6$$

	Predictor	P-Value	Adjusted R Squared	AIC	AICc	BIC	PRESS
Sixth Iteration	has_photo	0	0.3497	690588	690588	690650	27521373719
Seveneth Iteration	pets_allowed	2e-04	0.3498	690576	690576	690647	27514925508
Difference	NA	NA	0	12	12	3	6448211

Seventh Iteration

The model improved when including pets_allowed, while $AIC, AICc, BIC, PRESS$ got smaller.

$Model:$

$$Y = \beta_0 + X_1\beta_1 + X_2\beta_2 + X_3\beta_3 + X_4\beta_4 + X_5\beta_5 + X_6\beta_6 + X_7\beta_7$$

	Predictor	P-Value	Adjusted R Squared	AIC	AICc	BIC	PRESS
Seventh Iteration	pets_allowed	2e-04	0.3498	690576	690576	690647	27514925508
Eight Iteration	fee	0.0195	0.3499	690573	690573	690652	27513340266
Difference	NA	NA	0	3	3	-5	1585242

Forward Stepwise Search

Completed

Hence, we have found our linear model since the previous iteration did not improve the model, with the predictor variables being Square feet (X_1), density (X_2), population(X_3), bathrooms (X_4), bedrooms (X_5), has photo (X_6), and pets allowed (X_7).

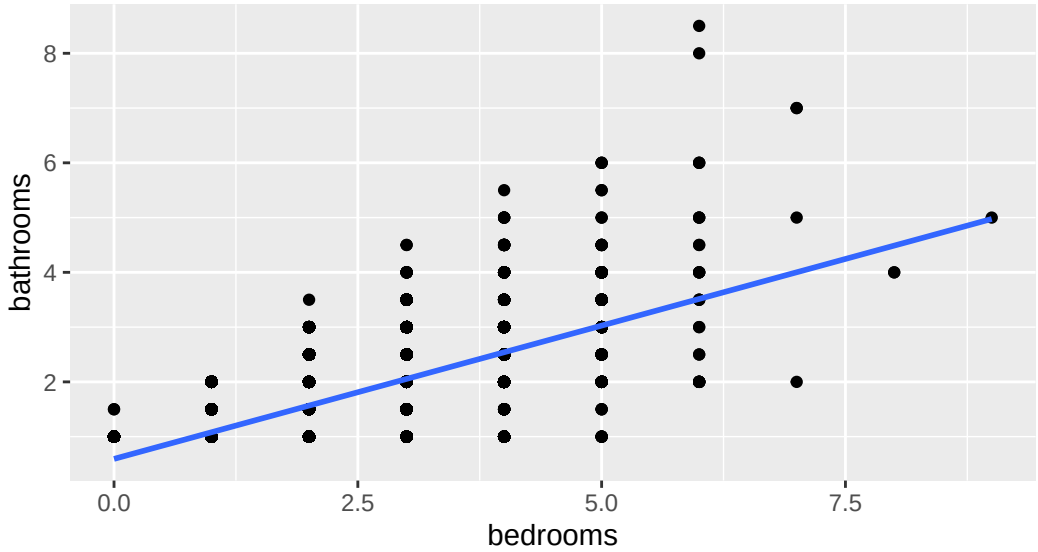
$$Y = \beta_0 + X_1\beta_1 + X_2\beta_2 + X_3\beta_3 + X_4\beta_4 + X_5\beta_5 + X_6\beta_6 + X_7\beta_7$$

After fitting the model with every predictor, we find that the Population variable results in the lowest P value of the model.

Table 9: Summary of Model

Statistic	Values	SE	P_Value	R_Squared
Intercept	174.6287	14.3582	0.0000000	0.3499
Square Feet	1.0232	0.0129	0.0000000	
Density	0.1899	0.0021	0.0000000	
Population	0.0528	0.0014	0.0000000	
Bathrooms	221.8106	8.8454	0.0000000	
Bedrooms	-141.8307	6.1611	0.0000000	
Has Photo	-72.6216	11.0829	0.0000000	
Pets Allowed	-76.1832	20.6061	0.0002183	

Note that the coefficient for bedrooms is negative, which we thought was suspicious as usually more bedrooms are more valuable when looking for houses. We thought that this might be due to multicollinearity with the bathrooms variable, so we decided to plot the two against each other.



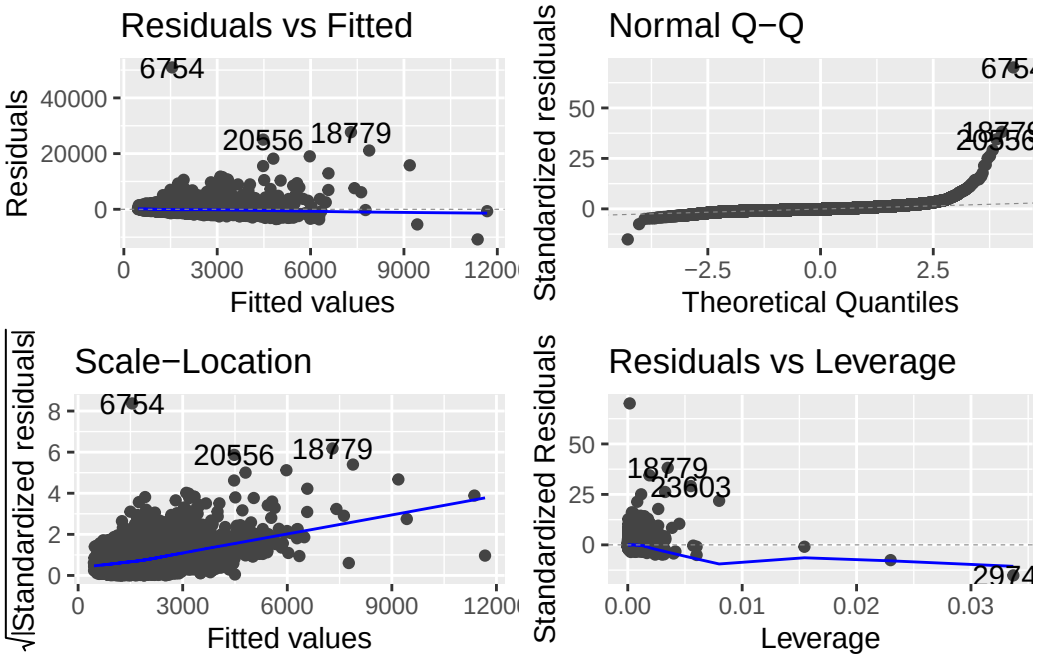
We noticed that there seems to be a trend, as when amount of bedrooms go up, amount of bathrooms go up. The R^2 value of the line of best fit was also 0.46, indicating a clear trend. Therefore, we decided to remove bedrooms from our model, due to its redundancy, given its

correlation with the bathrooms predictor. The R^2 of the new model decreases slightly, but is acceptable due to it being a reduction of overfitting.

Analysis of Residuals - Model Improvement

We will now conduct a thorough examination of the model's residuals to identify any potential issues.

Residuals vs. Fitted Values

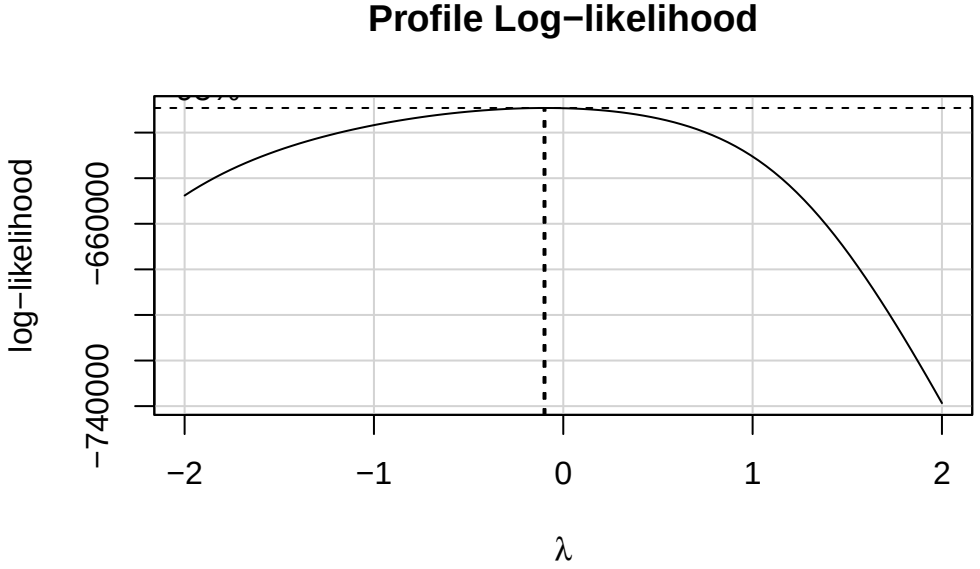


We can see slight heteroskedasticity in the Residuals vs. Fitted plot, with a slight downward trend. There are a few outliers points and the model decreases in accuracy for extreme fitted values. Since our model is not able to discern luxury apartments this may explain some of the large outliers. The Q-Q residual plot shows a normal distribution for a good amount of the data. The plot has deviates from the normal distribution around the extremes, meaning our model struggles to predict very large or small rents, but for the vast majority of the data it follows a normal distribution. Again this could be due to features of apartments that our data could not capture, like luxury accommodations or a bad location. The Scale Location plot is not ideal, as the blue line should be horizontal. The means homoscedasticity is violated, but this is likely to have higher variation in high-fitted values. As prices become more expensive, there are factors we can not account for, especially with luxury apartments. The Residuals vs. Leverage plot

shows high leverage points, highlighting points that warrant further analysis to determine if they are outliers and whether they accurately reflect the relationship in question.

Outliers and Data Transformation

A few points show up multiple times as outliers/high leverage points, being the apartments denoted by the indexes 6754, 18779, and 20556. After analyzing these points, we see that their price differs from the price in the description of the apartment, leading us to believe that these apartments were mistakenly inputted into the dataset. To address the non-normality and heteroscedasticity observed in some plots, we applied a Box-Cox transformation to the data. This transformation assists in normalizing the data and stabilizing variance.



The BoxCox formula is: $y(\lambda) = \frac{y^{\lambda}-1}{\lambda}$ if $\lambda \neq 0, \ln(y),$ if $\lambda = 0$ Since the optimal lambda value is close to 0, we can simply take a logarithm of the price, as defined by the BoxCox formula. Our adjusted model is then:

$$Y = \beta_0 + X_1\beta_1 + X_2\beta_2 + X_3\beta_3 + X_4\beta_4 + X_6\beta_6 + X_7\beta_7$$

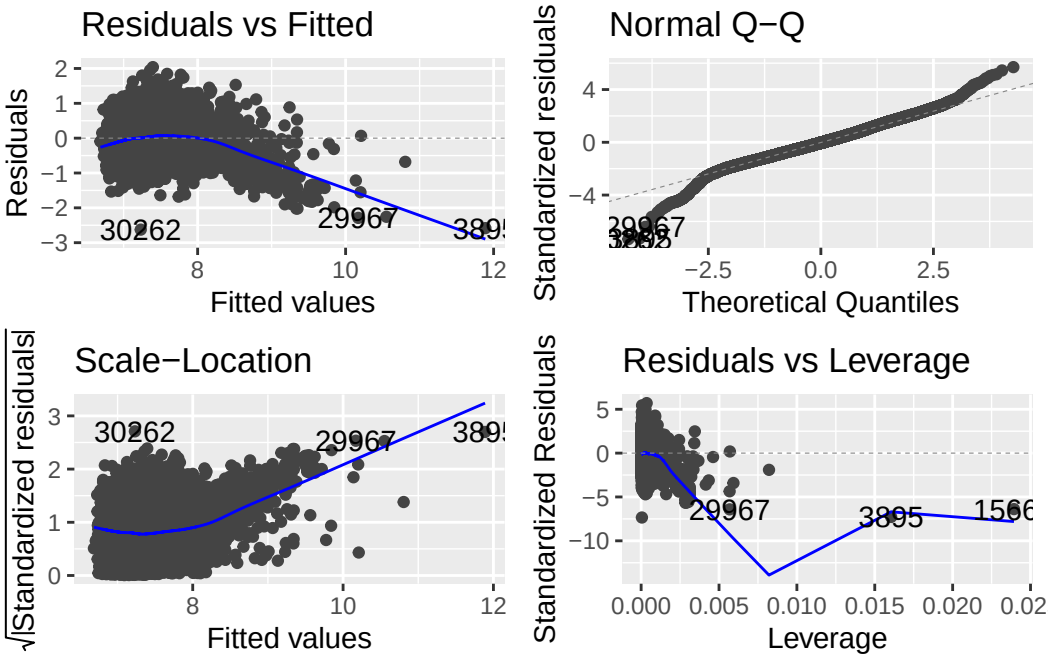
where the predictors are square feet (X_1), density (X_2), population(X_3), bathrooms (X_4), has photo (X_6), and pets allowed (X_7).

Table 10: Summary of Model

Statistic	Values	SE	P_Value	R_Squared
Intercept	6.5663	0.0081	0e+00	0.3455
Square Feet	0.0004	0.0000	0e+00	

Statistic	Values	SE	P_Value	R_Squared
Density	0.0001	0.0000	0e+00	
Population	-0.0037	0.0007	6e-07	
Bathrooms	0.1170	0.0041	0e+00	
Has Photo	-0.0435	0.0055	0e+00	
Pets Allowed	-0.0603	0.0102	0e+00	

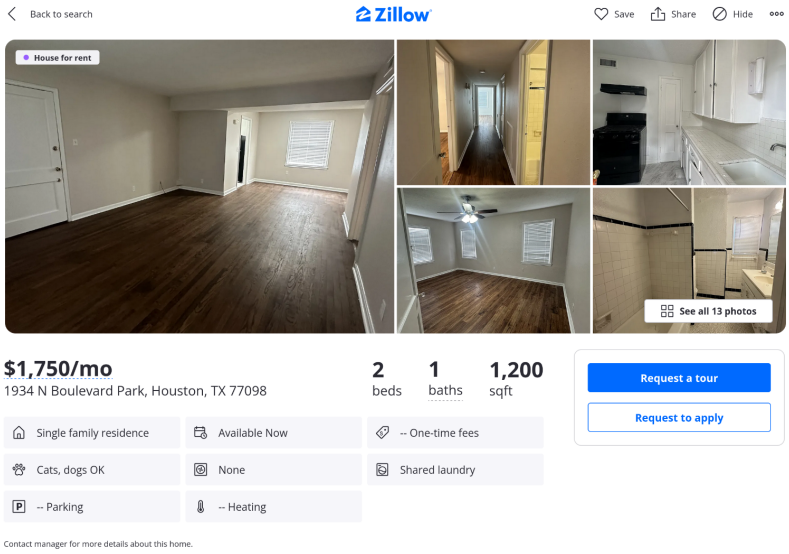
The R^2 improved from 0.3433 to 0.3455, which is a very slight but noticeable improvement. Let us now see the effects the transformation had on the residuals, which is more important.



In the Residuals vs Fitted plot, residuals for lower-priced houses are more evenly distributed around zero, although the model still struggles with higher-priced apartments, as indicated by the skewness of the red line. The QQ plot reflects enhanced accuracy, particularly in the upper tail, with the maximum of the standardized residuals decreasing from nearly 60 to about 5, showing the theoretical and predicted residuals are closer together. The Scale-Location plot suggests that while homoscedasticity issues persist for high fitted values, the data for lower fitted values is now more normally distributed. The Residuals vs Leverage plot shows similar information, but since we removed all high leverage points (which were outliers), there are no points which lie outside of the Cook's distance. Overall, the normality and homoscedasticity of the residuals improved, indicating that our transformation made the linear model more accurate.

Confidence Interval for a Real Apartment Listing

To test our model on relevant data, we found a real apartment listing around Rice. The density metric was taken from the dataset, multiplied by 2 to account for the higher density of the area around the Med Center.



The confidence interval generated by the model:

fit	lwr	upr
1772.582	1747.293	1798.238

We see that the predicted price falls within the confidence interval! Especially with the high variance of the data, this is a successful result.

Summary

In our analysis, we explored the data and identified eight usable regressors, comprising five continuous and three categorical variables. We employed a Forward Stepwise Search to construct our model, using 6 of the 8 regressors, then enhanced it by taking the logarithm of the price, which resulted in an improved normality and R^2. However, there are some limitations to our approach. Forward Stepwise Search is a greedy algorithm, meaning it makes the best choice at each step but does not guarantee the most optimal solution overall. In future work, we aim to expand our dataset to include more regressors. Our group hypothesizes that a regressor representing the luxuriousness of an apartment could have a significant impact on the model. Additionally, incorporating interaction terms might help account for nonlinear effects.

Reflection

Our group spent about 18 hours on this project. Our first problem was that we couldn't find good data. Initially, we started with the idea to predict the probability that a NCAA Baseball Team wins a game given a log of all NCAA college games from last year. But, we ran into the problem with the data that we had hoped to use. Luckily, a few of our group members brought up that they were looking for apartments next year and so we decided to find data that fit that idea. We were able to find clean data and started our forward selection process to find the most important variables. Our second problem was that we had a few outlier points. Instead of removing without any evidence as to if it was a high leverage point, initially we removed these points. After further thought, we decided to test if these points were of high leverage before removing and gave us more insight into our data.

The following were some items that we learned by doing this project: Forward Stepwise Search, and specifically how to decide whether to add a variable into the data or NOT. AIC and its variants, which helps measure trade off between model fit and complexity with a penalty for more complexity introduced into the model. How to look for GOOD data, i.e. data that is easy to clean and friendly in R's environment. How to read different plots, such as the Residuals vs. Fitted Values, the Quantile Quantile Plot, and the Residuals vs. Leverage. For instance, for the Quantile Quantile plot, we were able to determine if our data compares to the distribution of a normal distribution. How to adjust the model such as adding a scaling function like a natural logarithm function.

Advice that I would give students next time and that I wish I knew was how important it is to pick good data. This made our process a lot easier because the data was easily manipulated in R.

Sources

Apartment for Rent Classified [Dataset]. (2019). UCI Machine Learning Repository. https://doi.org/10.24432/C5X623.

Appendix

Histogram

```
data <- read.csv2("apartments.csv")
data$price <- as.numeric(data$price)
```

```
Warning: NAs introduced by coercion
```

Function to determine which variable has the lowest P-Value

```
cleaned <- data %>%
              filter(currency == "USD", price_type == "Monthly", nchar(state) == 2, bathrooms
              mutate(pets_allowed = ifelse(pets_allowed %in% c("Cats", "Dogs", "Cats, Dogs",
              mutate(has_photo = ifelse(has_photo %in% c("Thumbnail", "Yes"), 1, 0)) %>%
              mutate(fee = ifelse(fee == "Yes", 1, 0))
cities <- read.csv("uscities.csv")
cleaned <- cleaned %>%
  mutate(city = tolower(cityname), state = tolower(state))
cities <- cities %>%
  mutate(city = tolower(city), state = tolower(state_id))
cleaned <- cleaned %>%
  mutate(city = trimws(city), state = trimws(state))
cities <- cities %>%
  mutate(city = trimws(city), state = trimws(state))
data <- cleaned %>%
  left_join(cities %>% select(city, state, population, density),
            by = c("city", "state"))
```

```
Warning in left_join(., cities %>% select(city, state, population, density), : Detected an u
i Row 48924 of `x` matches multiple rows in `y`.
i Row 62 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#can do either median or mean.
pop_mean <- (mean(cities$population, na.rm=TRUE))
density_med <- mean(cities$density, na.rm=TRUE)
cleaned <- mutate(data, population = ifelse(is.na(population), pop_mean, population), densit
Y <- cleaned$price
cleaned$bathrooms <- as.numeric(cleaned$bathrooms)
cleaned$bedrooms <- as.numeric(cleaned$bedrooms)
cleaned$price <- as.numeric(cleaned$price)
cleaned$density <- as.numeric(cleaned$density)
cleaned$population <- as.numeric(cleaned$population)
cleaned$square_feet <- as.numeric(cleaned$square_feet)
regression <- function(regressors, add_col = NULL) {
  low_pval <- 10
  best_var <- NULL
  adj_r2 <- 0
```

```
  best_var_sum <- NULL
  high_t_val <- 0
  AIC <- NULL
  AICc <- NULL
  BIC <- NULL
  # Iterate through the regressors
  for (i in regressors) {
    # Dynamically create the formula
    if (is.null(add_col) || length(add_col) == 0) {
      formula <- reformulate(i, response = "price")
    } else {
      formula <- reformulate(c(add_col, i), response = "price")
    }
    # Fit the linear model
    model <- invisible(lm(formula, data = cleaned))
    sum <- summary(model)
    # Extract the p-value for the slope (the last row of coefficients)
    # Ensure the slope term exists
    if (nrow(sum$coefficients) > 1) {
      pvalue <- sum$coefficients[nrow(sum$coefficients), "Pr(>|t|)"]
      tvalue <- abs(sum$coefficients[nrow(sum$coefficients), "t value"])
      # Update the lowest p-value and corresponding variable
      if (tvalue > high_t_val) {
        low_pval <- pvalue
        best_var <- i
        adj_r2 <- sum$adj.r.squared
        best_var_sum <- sum
        high_t_val <- tvalue
        AIC <- extractAIC(model, k=2)[2]
        npar <- length(sum$coefficients) + 1;
        n <- length(sum$residuals)
        AICc <- AIC + 2*npar*(npar + 1) / (n - npar - 1)
        BIC <- extractAIC(model, k = log(n))[2]
  residuals <- residuals(model)
  leverage <- hatvalues(model)
  press_statistic <- sum((residuals / (1 - leverage))^2)
      }
    }
  }
  # Return the variable with the lowest p-value
  return(c(best_var, low_pval, adj_r2, AIC, AICc, BIC, press_statistic))
}
```