Minimizing Zestimate® Error

Springboard Career Track - Capstone Project 1 Final Report

By: Ken Wallace

18 October 2017

Problem statement:

Minimize the error between estimated home values and the actual sales prices for houses that sold. The problem as stated in the Kaggle competition is to attempt to improve upon Zillow's estimated home values, Zestimates, by evaluating features of homes in three southern California counties and calculating the error between the Zestimate and the sale price of that property ($logerror = log(Zestimate) - log(SalePrice)$).

Revised problem statement: For this project, I am only creating a model(s) that will provide the best MAE (Mean Absolute Error) for the prediction for logerror. This does not incorporate the 6 different time points required by the Kaggle competition, as the purpose of this project was to gain a strong understanding of how to analyze, visualize, gain statistical inferences, and model data, not specifically as a Kaggle exercise.

Data source and description:

The data was provided by Zillow for a Kaggle competition. There were two datasets provided, requiring data wrangling and merging, with the following information:

1. Training data (for 2016), containing 90275 rows and 3 columns:
    a. Parcel ID
    b. Log error
    c. Transaction date
2. Properties data (for 2016), containing 2985217 rows and 58 columns, including:
    a. Parcel ID, which will be used to merge on with the training data
    b. Columns containing various information for each property that can (hopefully) be used to predict the log error.

Data wrangling steps:

Training data:

After looking at the data, a scatterplot of logerror across the dataset showed that 98% of the values (excluding the top and bottom 1%) ranged in logerror from approximately -0.34 to +0.46. Then, looking at number of sales by month shows a distinct uptick after March, peaking in June, but staying high through September. Per Kaggle, we have incomplete data for Oct, Nov, and Dec that will be provided on 2 OCT 2017. After incorporating the data provided on 2 October 2017, the data sales data set for the three counties is now complete.

Properties data:

The first thing that is obvious in the properties data is the number of missing values. Only Parcel ID exists for all rows, with the rest having missing values ranging from 11437 (0.38%) to 2983593 (99.95%).

In order to explore the data, the first thing was to create a copy of the data frame and drop rows with missing values in the latitude and longitude columns (this was done by dropping latitude, as the same set was missing from longitude). This took a number of other missing values away as well. I then looked at the value counts for a number of the object variables (heating, zoning, aircon, etc.) just to consider imputing missing rows with most common values.

Update: The rows that were eliminated during the dropping of missing latitude and longitude values may return after incorporation of data provided on 2 October 2017 by Kaggle/Zillow. The cleaning operations that were executed during the original creation of the data wrangling notebook will be rerun after including (and merging) the 2017 data with the 2016 data.

Merged data:
The first step after merging the two datasets was to drop any columns with more than 25% missing values. This action dropped 28 columns. For the remaining columns, I looked at variables that could logically be grouped and subset them. For example, in the living area columns, there were a number duplicate values in calculated living area and finished living area. I used this logic to eliminate columns that did not provide useful information and to complete the columns with missing values through various methods. Any rows within the living area subset that contained no values were dropped. After cleaning this subset, the original data frame was updated to align with the subset.

The same logic was applied to the assessed taxes columns (where some NaNs in the tax_assessed_structure column could be completed as the difference between tax_assessed_parcel and tax_assessed_land).

At this point there was only one missing value for zoning_county code, which was replaced with the most common value for the value in that row's zoning column. For latitude and longitude values, I sorted the data by those columns and did a forward fill of NA values, assuming the nearest known location was a close enough approximation to use.

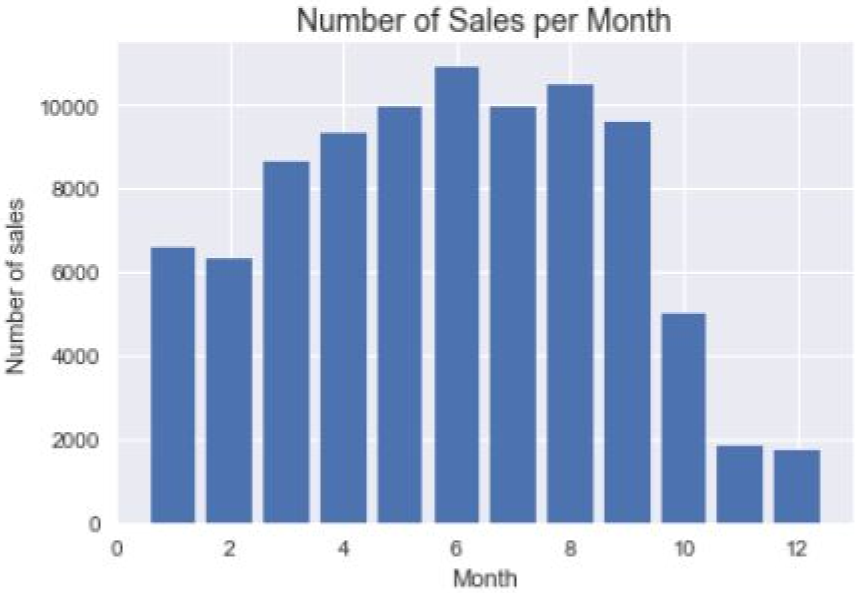
For bathrooms, the num_bath column had no missing values, so the num_bath_full and num_bath_calc columns were dropped.

Not knowing yet what to do with the remaining columns with missing values (build_year, censustractandblock, and lot area), the df was forked and these columns were dropped. They can easily be added back in.

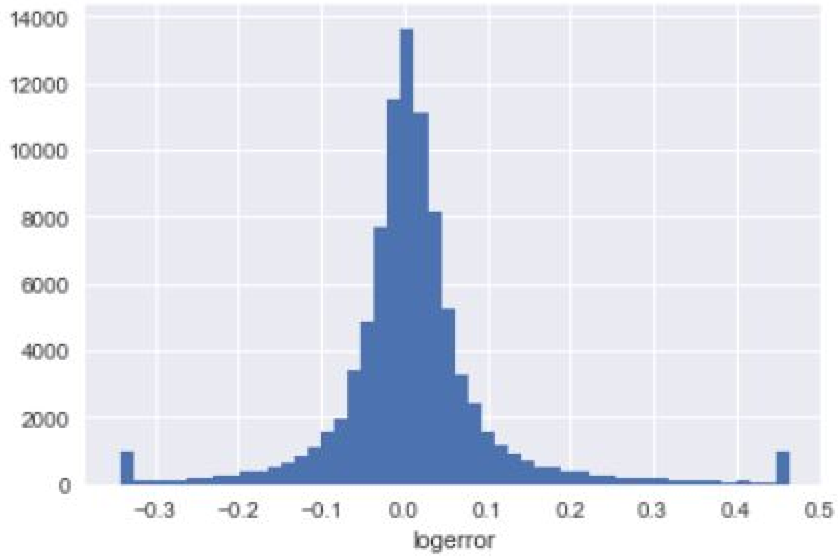
Findings -- Data Wrangling through Inferential Statistics:
As of this milestone report, the data wrangling, data story, and inferential statistics sections have been completed. The statistics section should be reviewed for completeness and correctness.

Observations from the data so far (from visualizations) include:

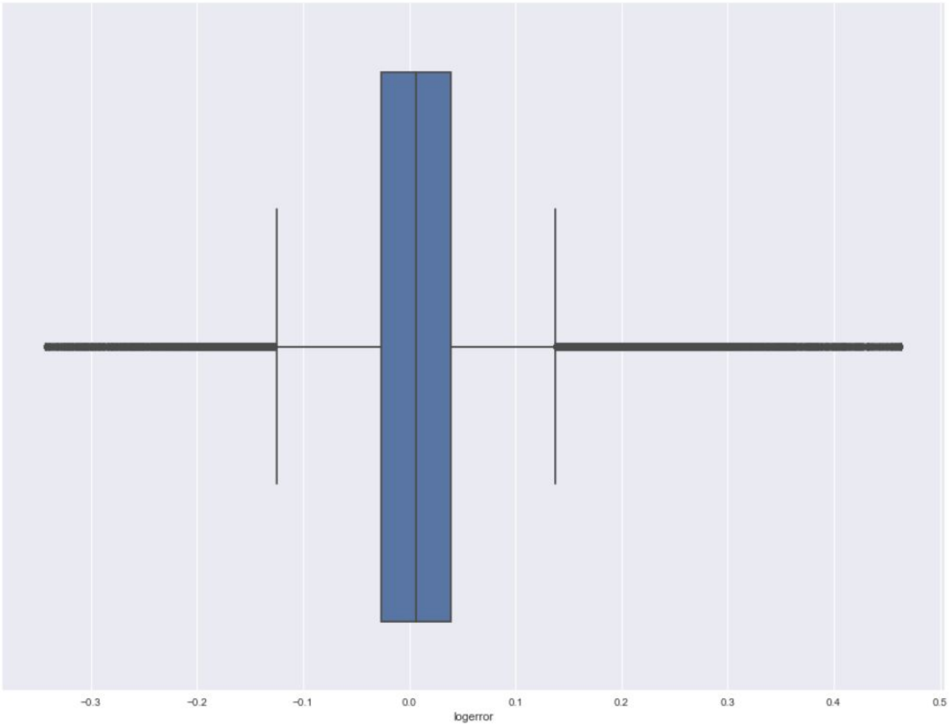
- Sales by month numbers are significantly higher starting with month 3 (March) compared with January and February.



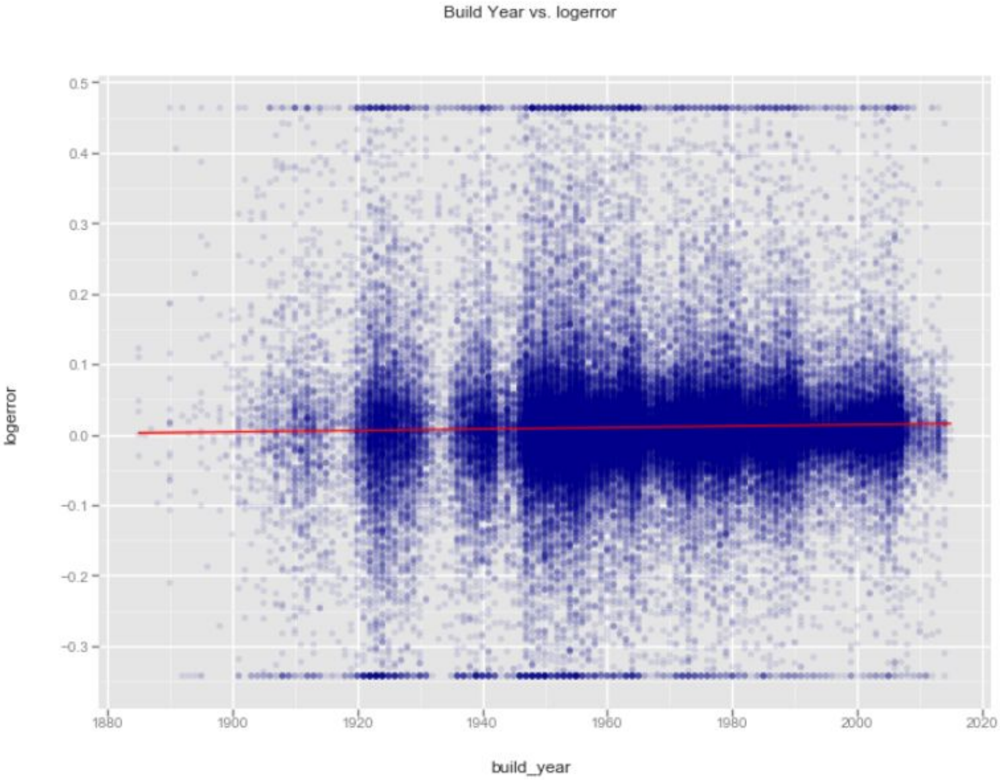
- Logerror values are slightly skewed to the right, with the mean value greater than zero (0.0618) and a longer tail toward the right (outliers collapsed to the 1st and 99th percentile values).



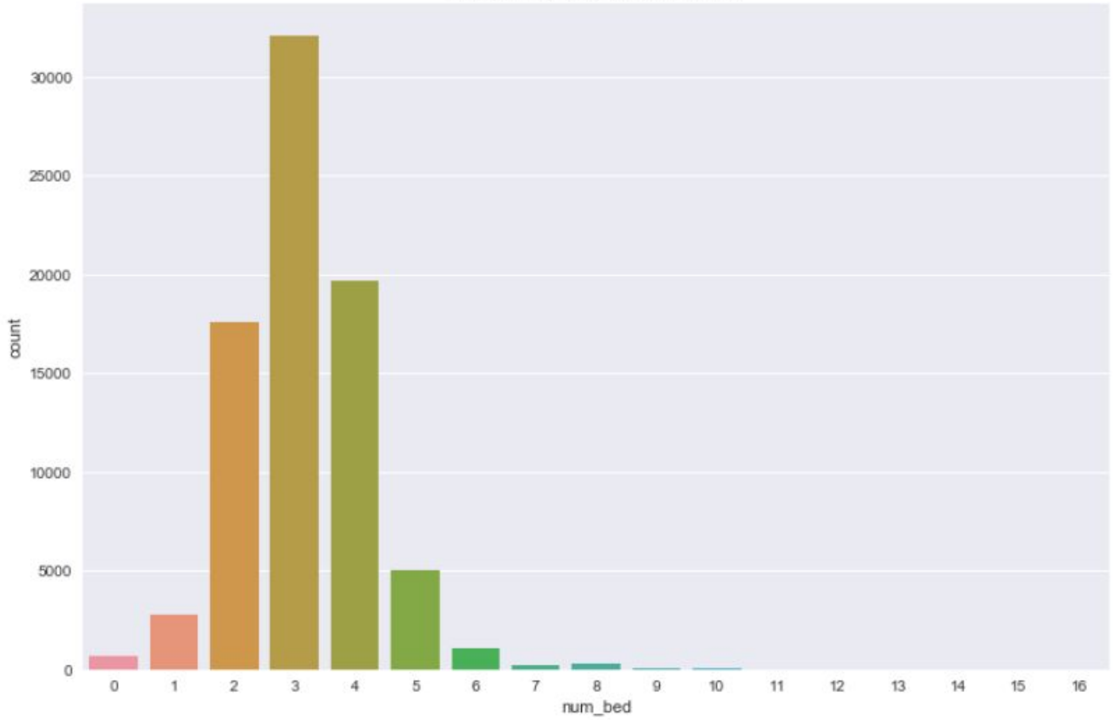
- The median value is 0.0334, with 25% and 75% values of 0.0141 and 0.0714, respectively.

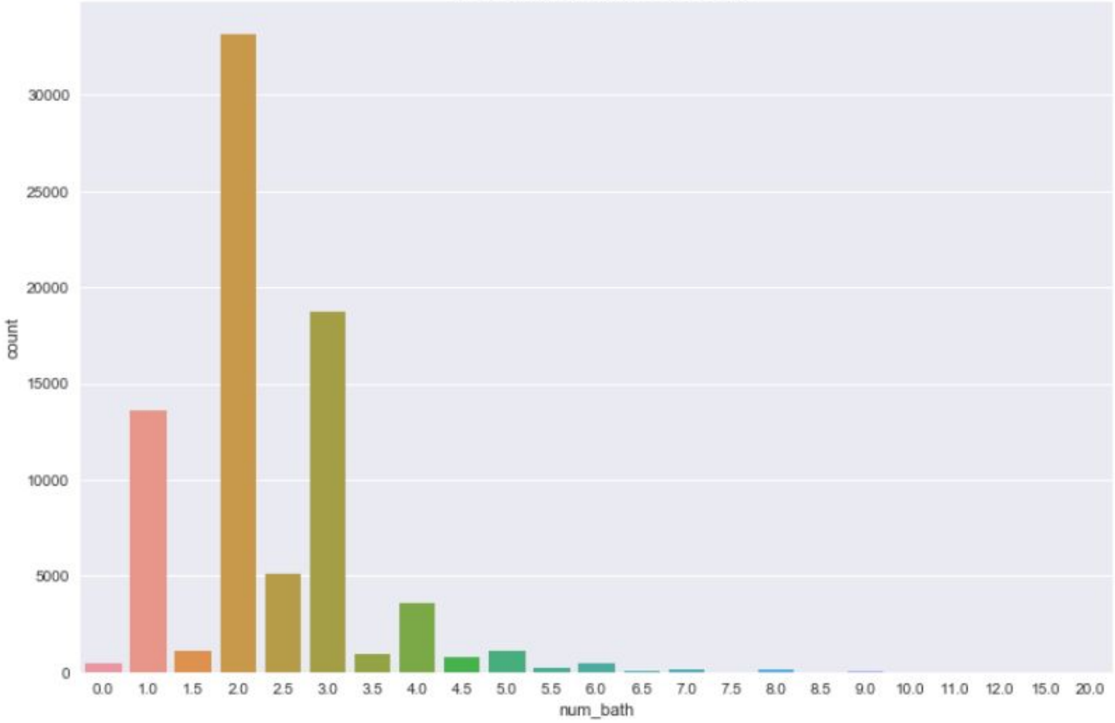


- The logerror increases as build_year increases, using a linear model

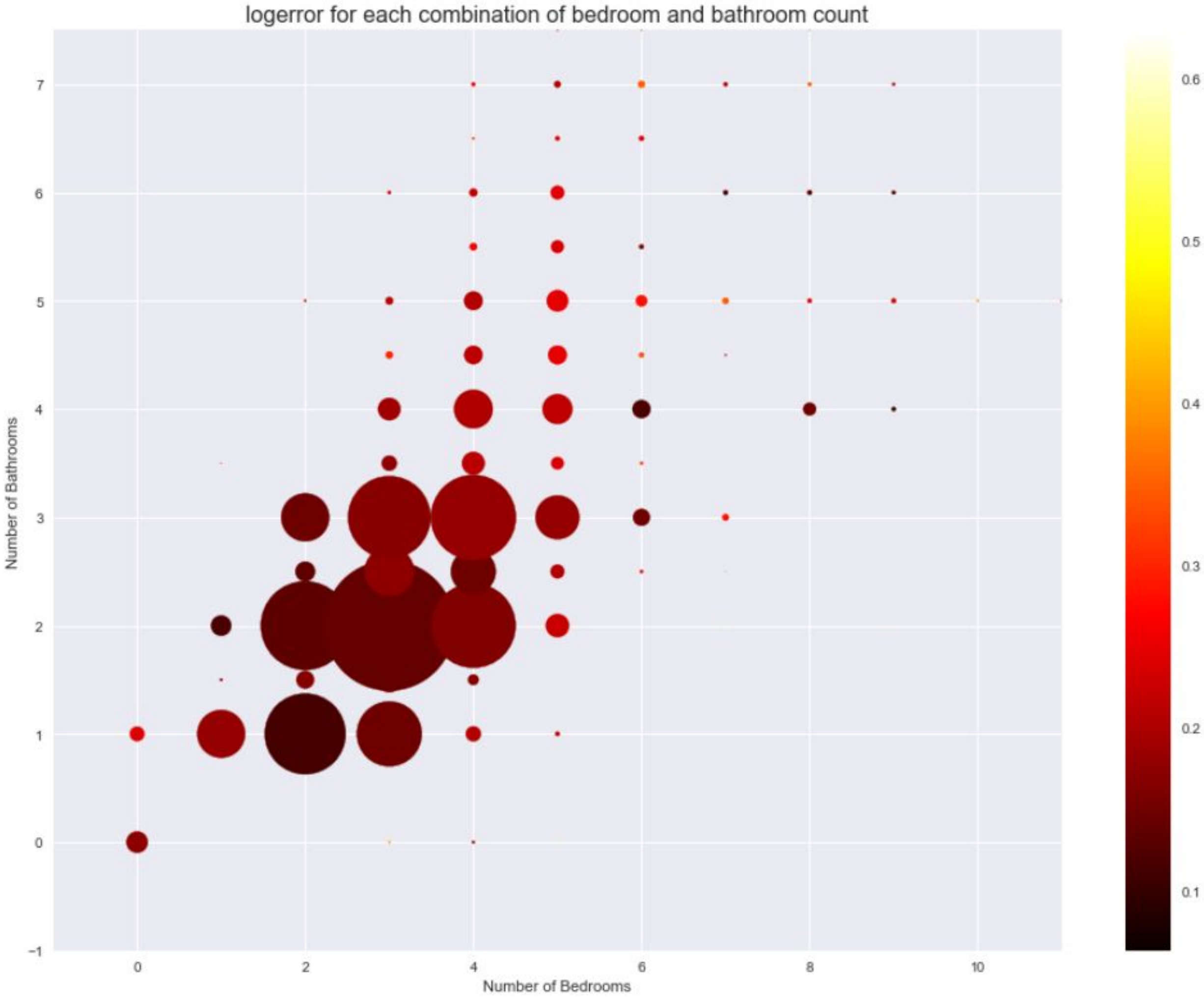


- 3 BR is the most common number and 2 BA is the most common number. This combination is also the most common.

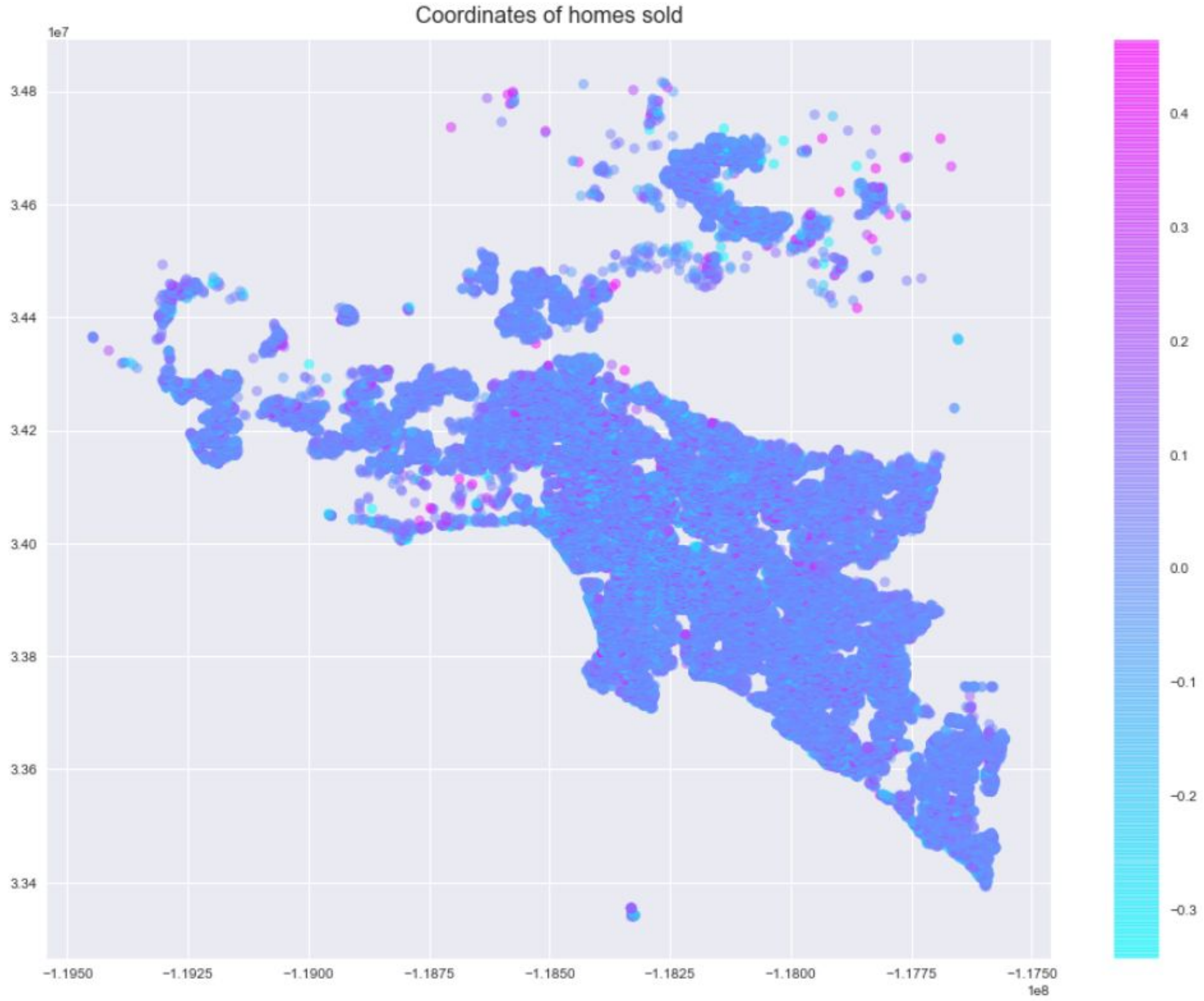

Frequency of bedroom count


Frequency of Bathroom count

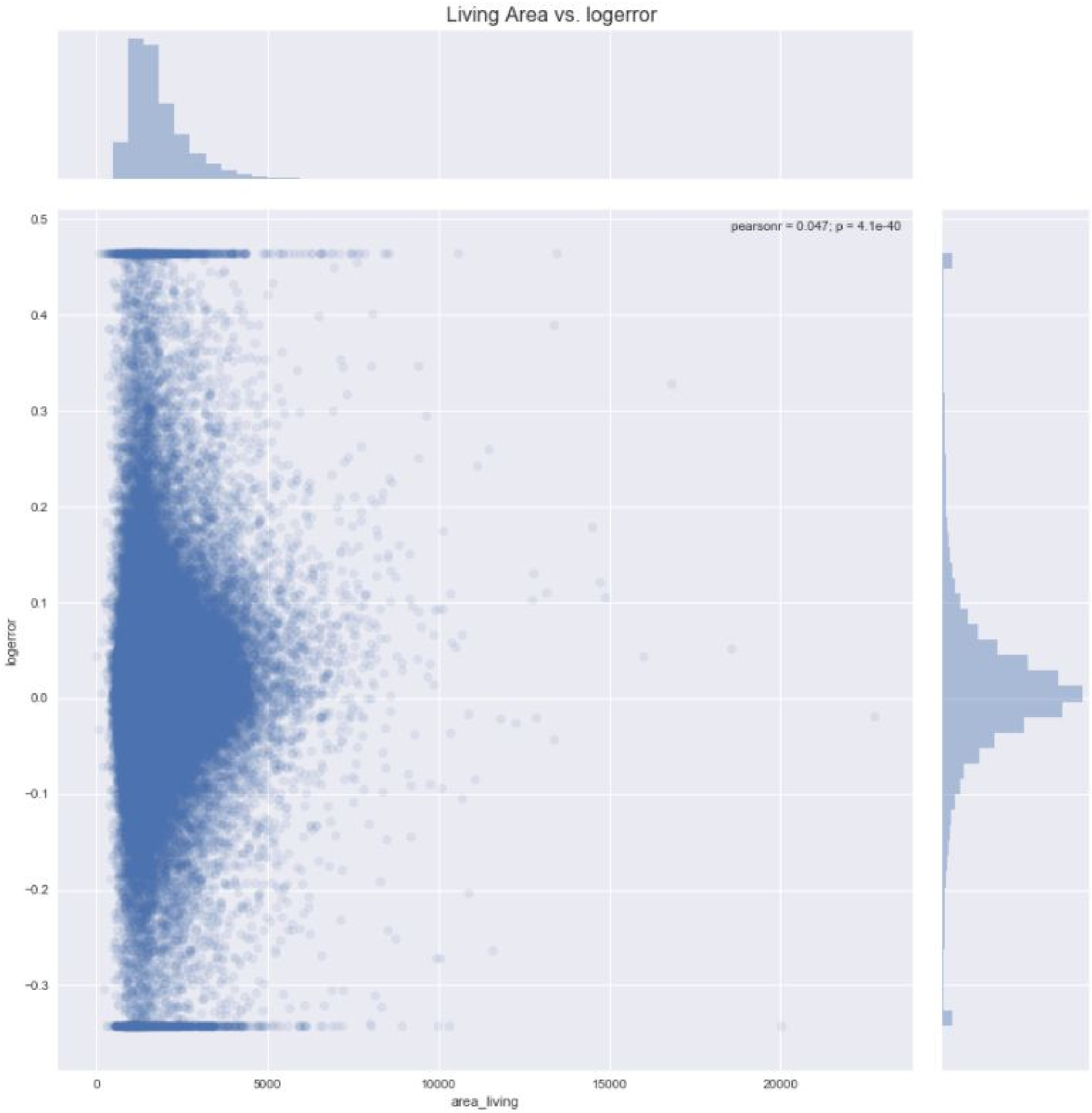
- Generally speaking, the higher the frequency for each combination of bedroom and bathroom count, the lower the absolute value of logerror. The lowest error for houses with fewer than 4 bedrooms and 3 bathrooms appears to be seen for either 2 bedrooms and 1 bath or 1 bedroom and 2 bath (though 2/1 is much more common than 1/2). The logerror for 3/2 homes is on the low side.



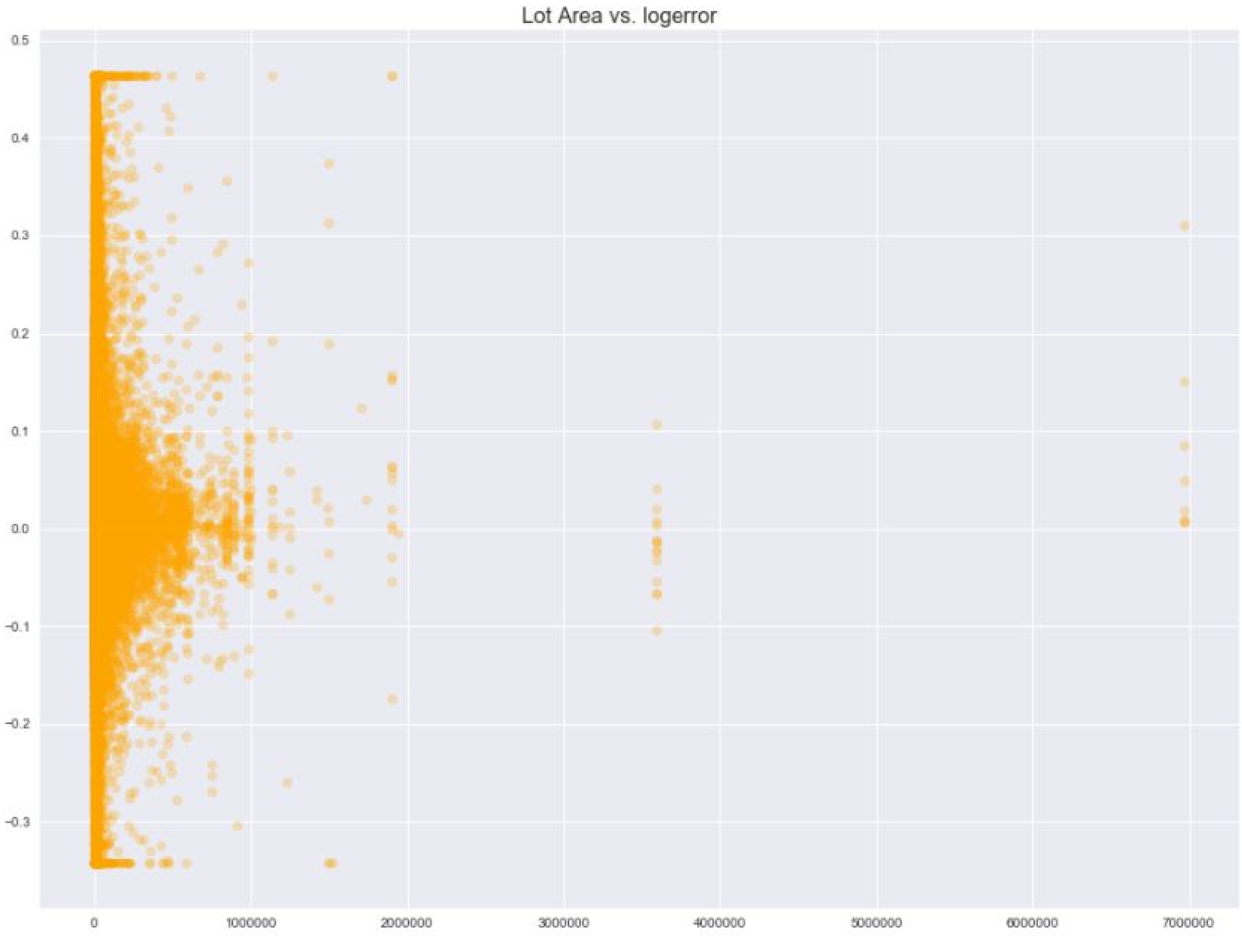
- There do not appear to be any specific areas with higher or lower error rates, they look well distributed.



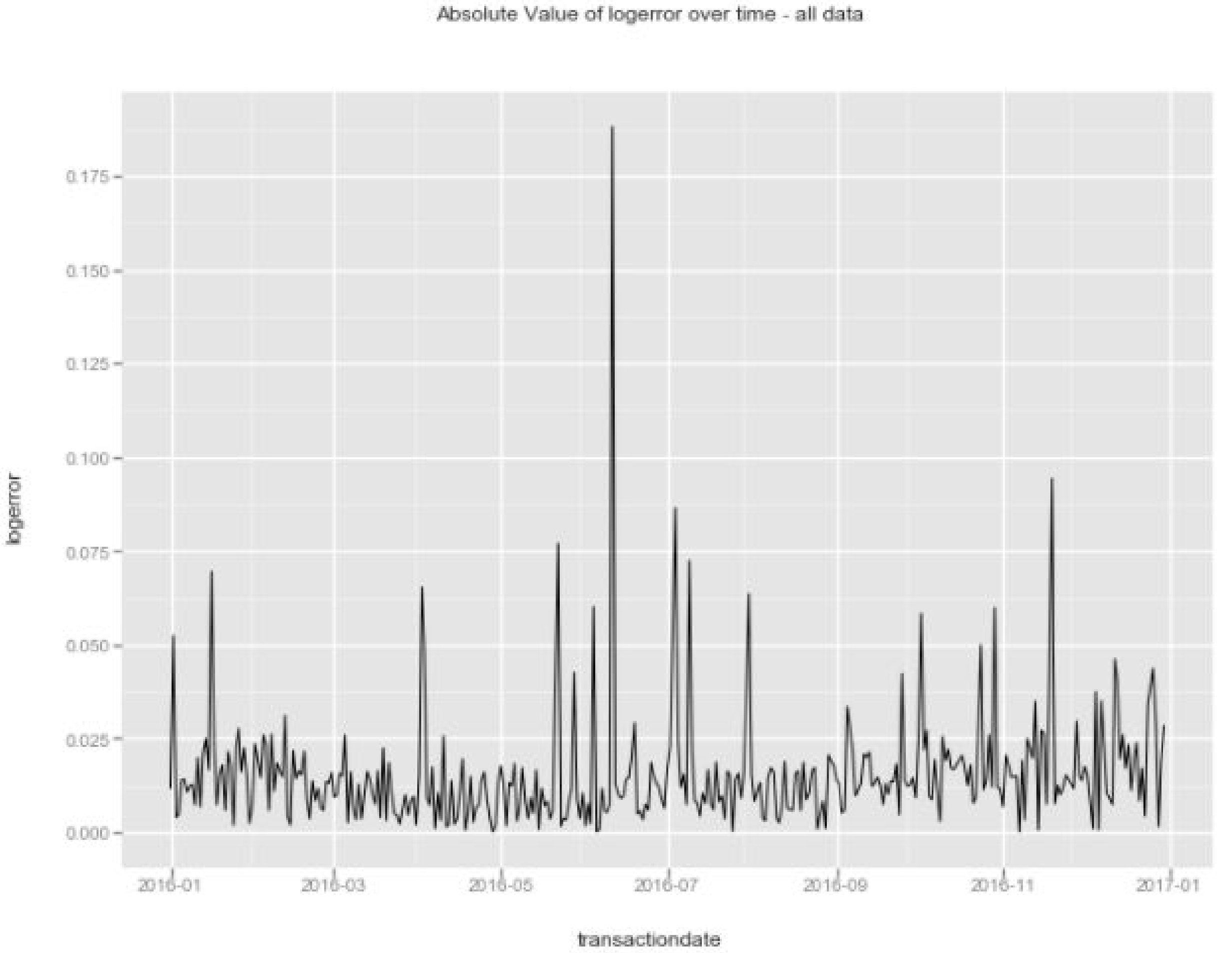
- The range of logerror values becomes smaller as the living area increases. The average living area is 1822 ft^2, with a standard deviation of 952 ft^2.



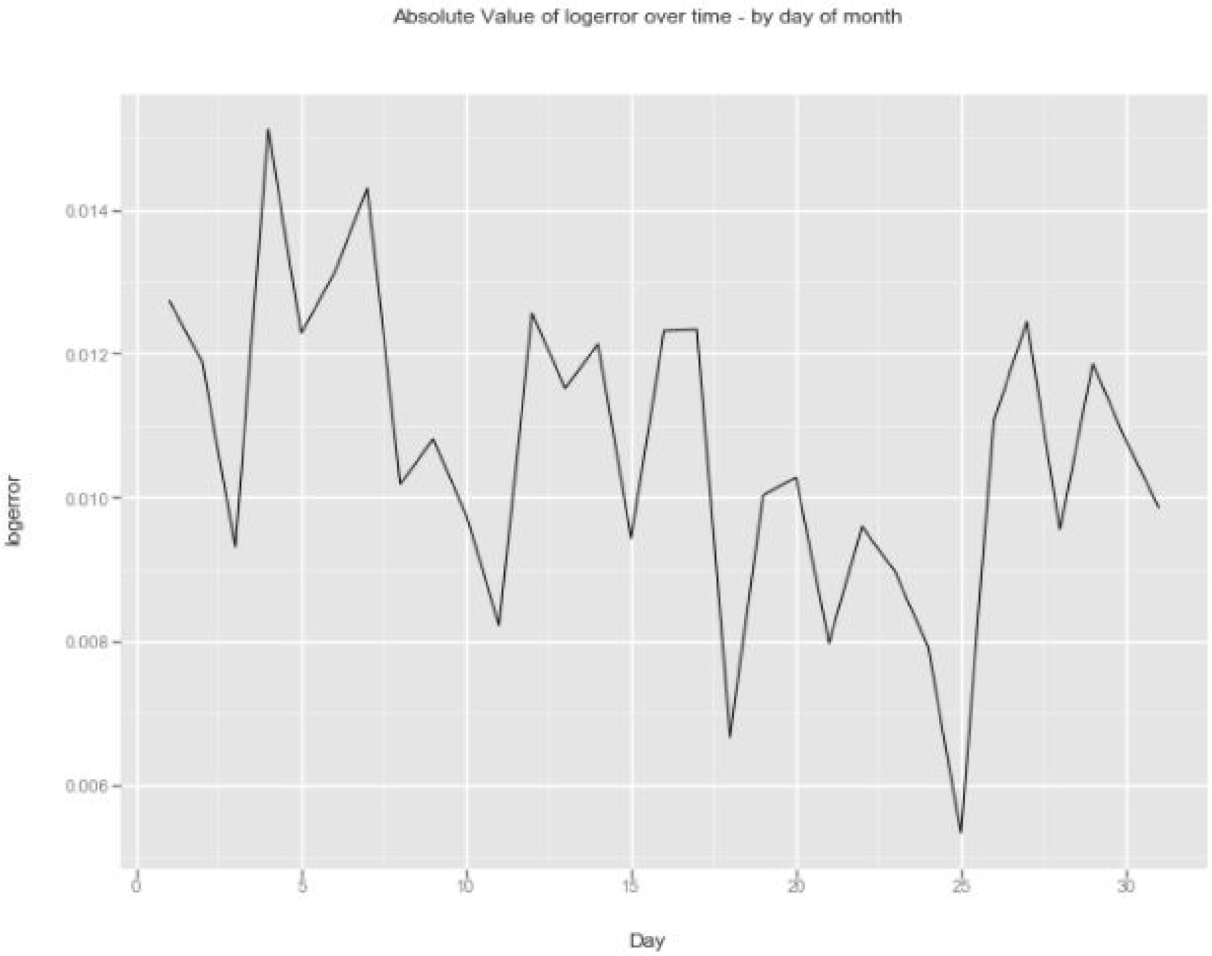
- Likewise, the range of logerror values becomes smaller as the lot area increases.



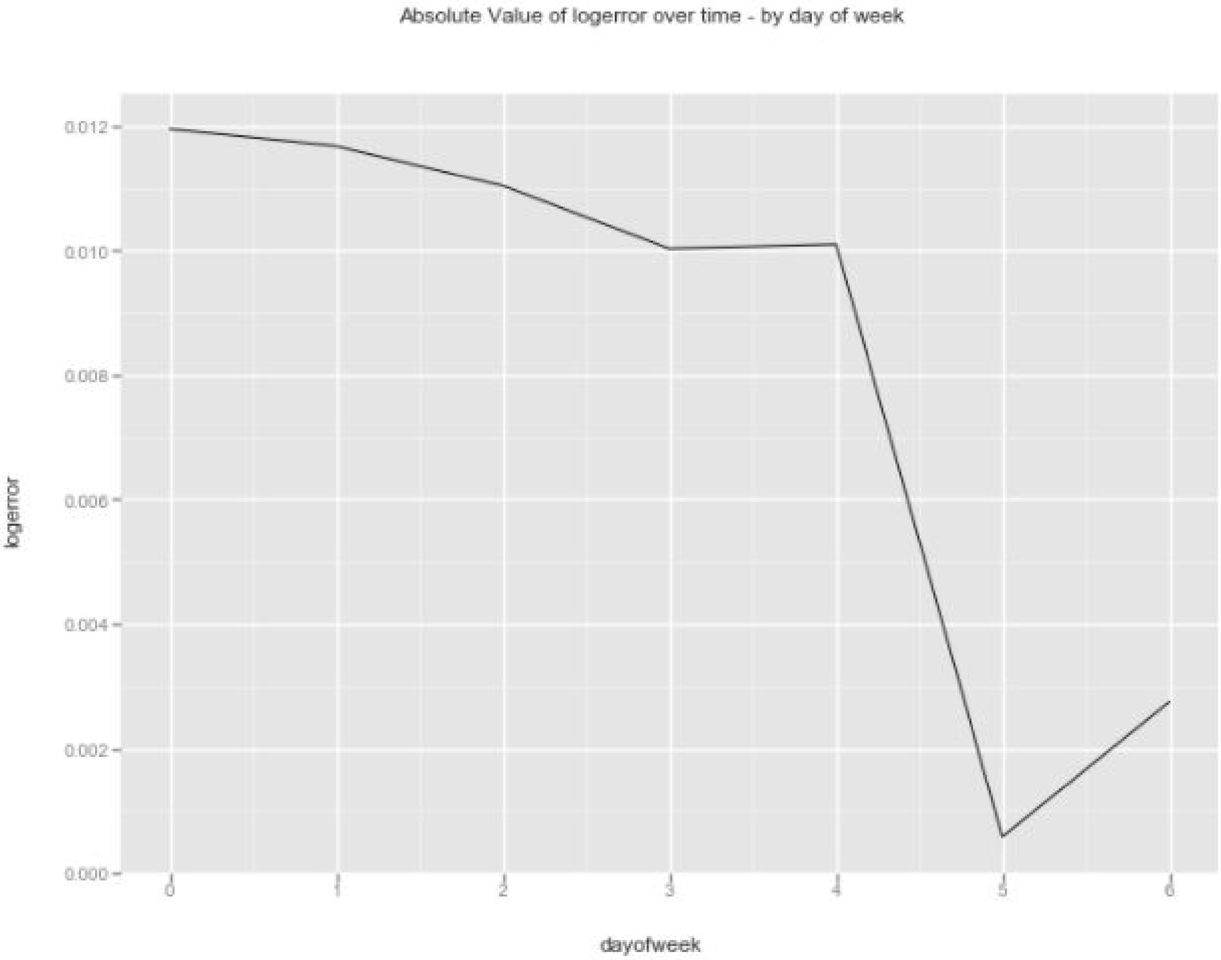
- There are a few spikes in logerror throughout the year, with the largest of those in June 2016 (approx 2x the error of the next largest spike).



- The average logerror for day of month appears to drop to its lowest point on the 25th of each month.



- The average logerror for day of the week appears to drop to its lowest point on the Saturday of each week, with Sunday being the second lowest. The third lowest, Friday, is 5x the average weekend error.



- There are no strong correlations between variables that aren't expected (e.g. assessed taxes for parcel and structure are highly correlated, as expected, as are living area and number of bedrooms and bathrooms).

Inferential Statistics:

Correlations

The absolute value of the Pearson correlations from logerror to each of the other variables are:

build_year	0.161333
county	0.081492
num_rooms	0.063035
tax_total	0.055983
area_living	0.048698
parcelid	0.048442
tax_assessed_land	0.046881
tax_assessed_parcel	0.04189
longitude	0.041444
tax_assessed_structure	0.021335
latitude	0.020818
month	0.018686
index_orig	0.018486
dayofweek	0.009764
Day	0.009252
area_lot	0.005284
num_bed	0.004231
city	0.003472
zip	0.002153
bbsum	0.002064
num_bath	0.000773

The highest value is just over 16%, which is not particularly strong.

For Kendall correlations, the top two correlated columns are the same, but the values are lower (build_year is about 10%).

For Spearman correlations, now county is higher than build_year, but it’s still only just over 10%.

Independent t-test for means

- logerror to number of bedrooms
- logerror to number of bathrooms
- logerror to sum of bedrooms and bathrooms
- logerror to living area
- logerror to lot area

For each of these t-tests, which will be performed as Welch's t-tests due to differing population variances, all null hypotheses are that the means are equal. P-values less than .05 will reject the null hypothesis in favor of the alternate, which is that the means are not equal.

Because all p-values are 0.0, it appears that this test is inadequate. Will proceed with machine learning operations to see if there is any connection between the dependent variable (logerror) and the independent variables.

Predictive modeling:

Models and parameters used:

The following regression models were created, with some slight variations of the features and hyperparameters in a couple of them. Regression models were appropriate here because the error rate is a continuous, rather than discrete, value.

There were three categories of regressions performed; Linear, Decision Tree, and Ensemble methods (specifically Random Forest).

Model	Features	Hyperparameters
Linear regression	Array with 9 variables ('logerror','zip','num_bed', 'num_bath','area_living', 'area_lot', 'month','city','county', 'tax_total')	NA
Decision Tree		Max depth = 2
Decision Tree		Max depth = 5
Decision Trees with Adaboost		NA
Random Forest		NA
Random Forest	Array with 3 variables ('area_living','area_lot','tax_total')	NA

Results:

Again, the goal for this project is to come up with a model with the *lowest* possible error. In this case, error calculated is the Mean Absolute Error between the difference of the predicted value and the actual sale prices for the homes in question.

In order from lowest to highest Mean Absolute Error, the errors and methods are:

- .0618 - Linear Regression
- .0619 - Decision Trees (Max depth = 2)
- .0622 - Decision Trees (Max depth = 5)
- .0642 - Random Forest (9 variables)
- .0672 - Random Forest (3 variables)
- .0738 - Decision Trees with Adaboost

Conclusions and recommendations:

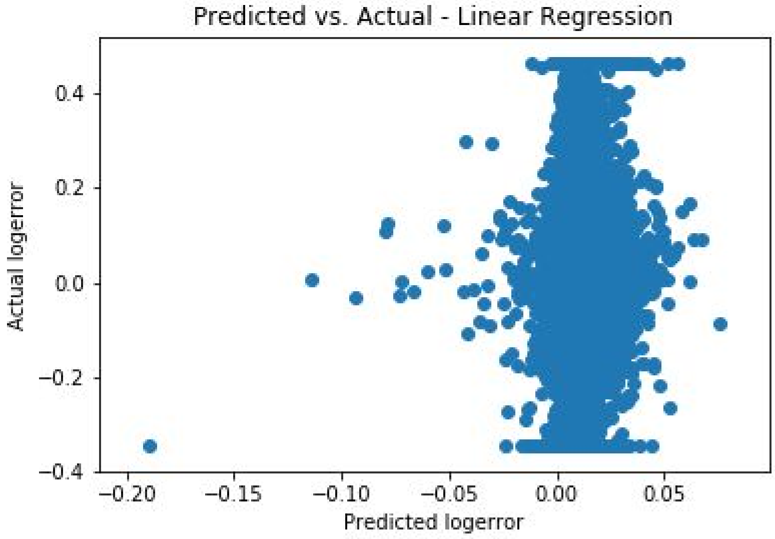
The results here are actually quite counter-intuitive Looking at the scatterplots (see Appendix A), it would have appeared that the Random Forest regressions would have had the lowest error and the Linear Model would have had the highest. Scatterplots were generated for each of the above models and the fit line (RF models only) looked much better for the RF (specifically the one with 9 features) than was expected for LM. One caveat, though, the RF models were fit with criterion = ‘mse’ because ‘mae’ would not run on my computer.

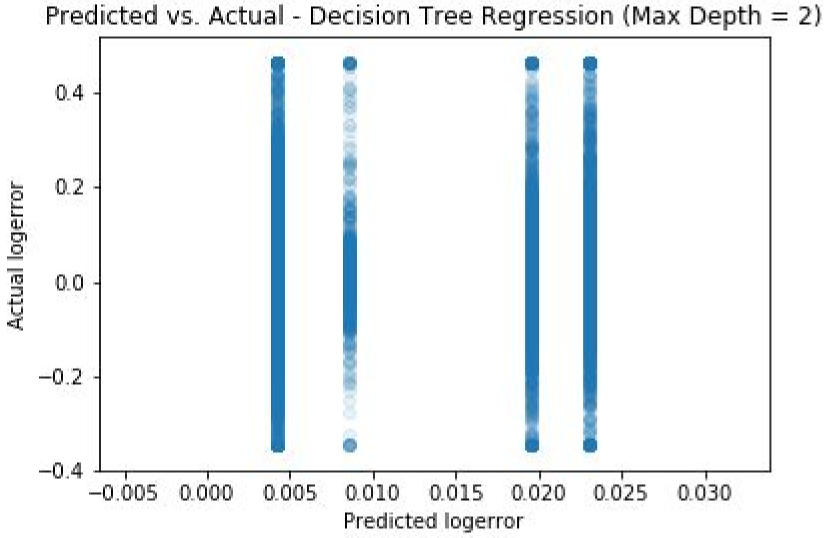
In addition, the LM and Decision Tree models appear to show near zero values for predicted logerror, compared with a range of actual logerror values between -0.36 and +0.47. This indicated that those models would be weaker, but the MAE values said otherwise.

All of this could be because using the logerror may also be counter-intuitive with regards to the way these plots should look. And if not, then the explanation isn’t clear to me. The data provided by Zillow did not allow this, but it might make more sense if we were predicting a more direct response variable, such as sale price, based on the features provided. By attempting to predict the error (and logerror at that) between actual sale prices and estimated values, we have lost the direct link between the features and their impact on the estimate.

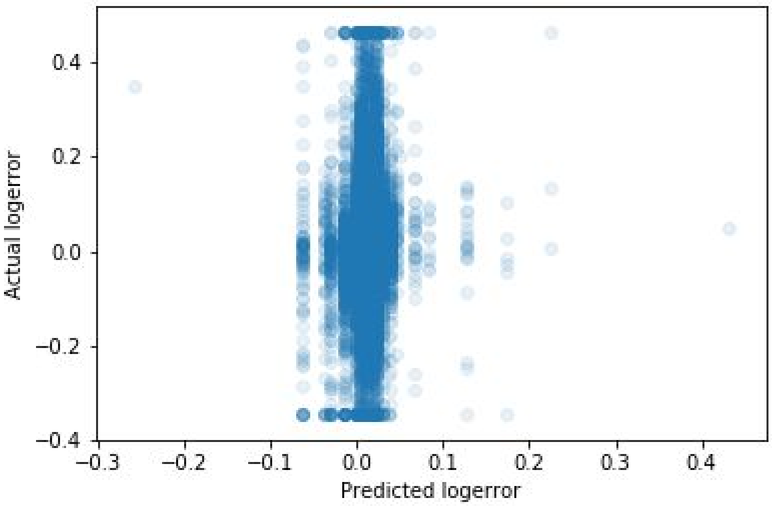
If one were to continue to improve the models for this prediction, I can’t help but recommend introducing additional data sources, such as stock market data, regional employment data, and recent neighborhood comparables, for example. As for model improvements, Additional regression models such as polynomial, lasso and ridge might also be worthwhile to try.

Appendix A

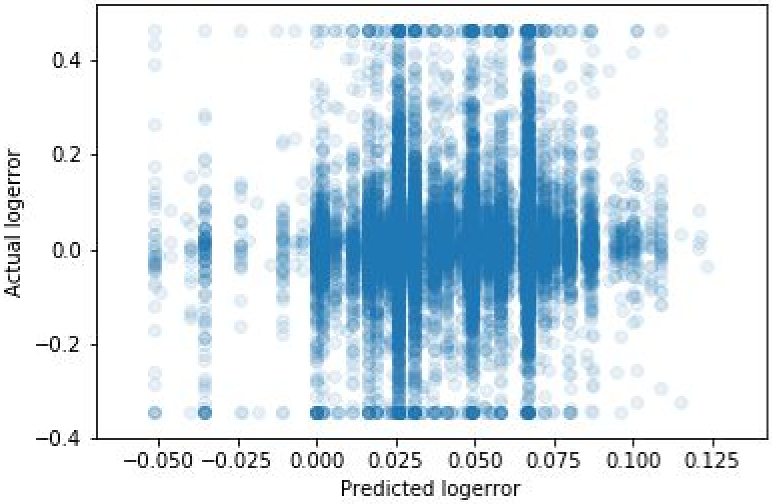

Predicted vs. Actual - Linear Regression
Actual logerror
0.4
0.2
0.0
−0.2
−0.4
−0.20
−0.15
−0.10
−0.05
0.00
0.05
Predicted logerror

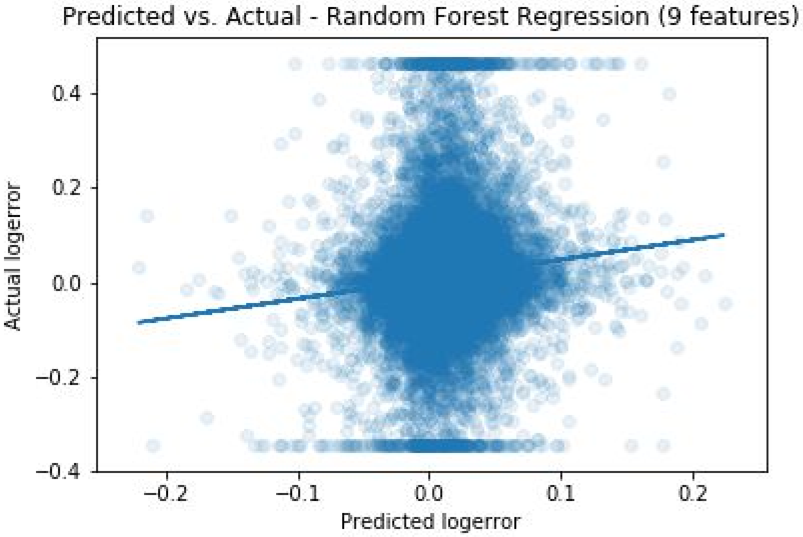

Predicted vs. Actual - Decision Tree Regression (Max Depth = 2)
Actual logerror
0.4
0.2
0.0
−0.2
−0.4
−0.005
0.000
0.005
0.010
0.015
0.020
0.025
0.030
Predicted logerror

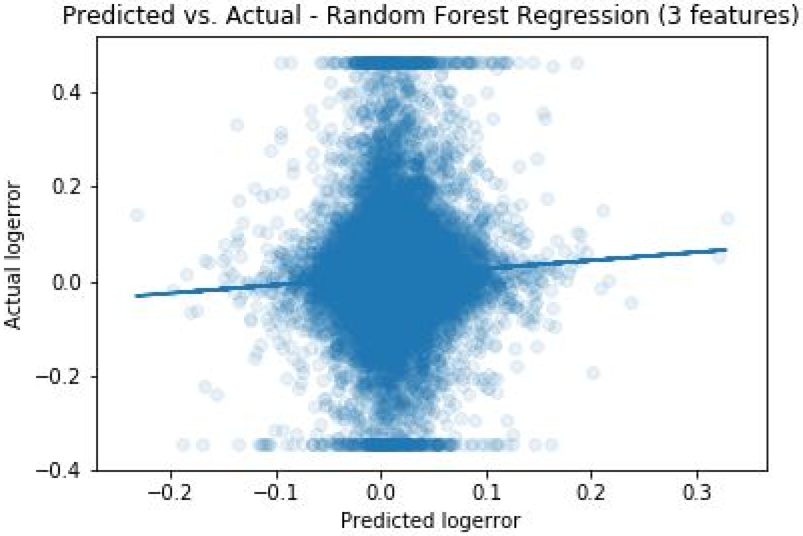
Predicted vs. Actual - Decision Tree Regression (Max Depth = 5)



Predicted vs. Actual - Decision Tree Regression w/ AdaBoost




Predicted vs. Actual - Random Forest Regression (9 features)
Actual logerror
Predicted logerror
0.4
0.2
0.0
−0.2
−0.4
−0.2
−0.1
0.0
0.1
0.2


Predicted vs. Actual - Random Forest Regression (3 features)
Actual logerror
Predicted logerror
0.4
0.2
0.0
−0.2
−0.4
−0.2
−0.1
0.0
0.1
0.2
0.3