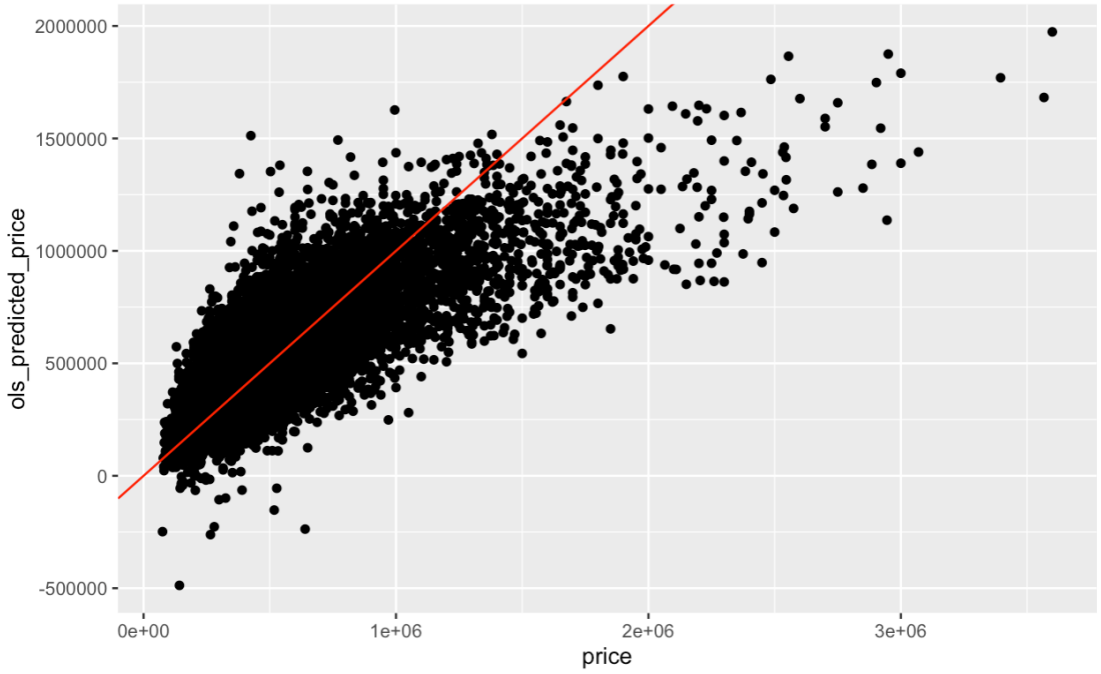
Andrew Goldblatt
ECON 213 - PSET3

#1
The OLS regression was based on the lm regression function with price on the left side and:

Price+beds+baths+living_area_sqft+lot_sqft+floors+has_waterfront_view+rating_of_view+unit_condition+construction_and_design_grade+floors_above_ground_level_sqft+basement_sqft+construction_year+most_recent_renovation+avg_living_area_sqft_of_nearby_neighborhoods+avg_lot_sqft_of_nearby_neighborhoods

On the right side. Data columns including X1, ID, date, and order were left out of the regression as those variables do not describe the value value of the homes and are used for identification and not qualitative descriptors of the house. The data for this was the training set. The regression was then used to predict home values in the training set, resulting in this distribution of ols predicted prices vs actual price:

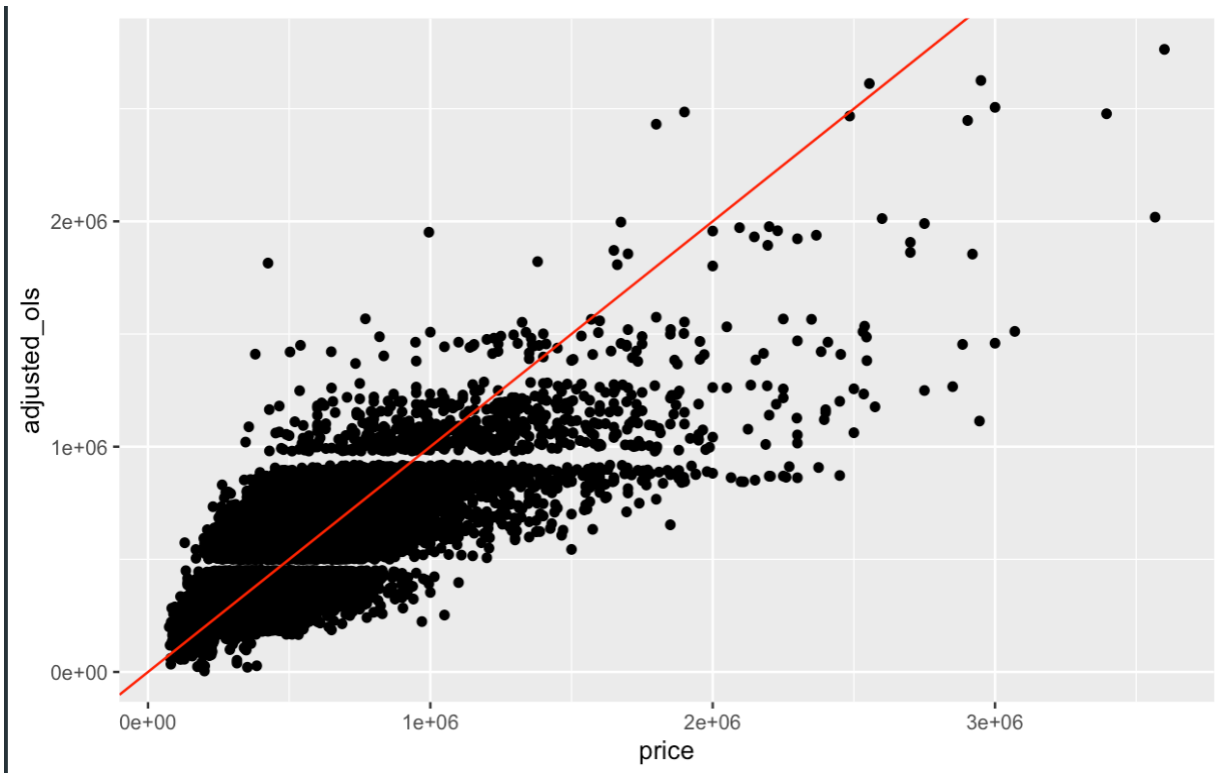


The ols model undervalued many homes over $1 million, over valued homes between roughly between $200k and $1m.

The following ols adjustment was applied to the predicted prices so the estimates were more closely aligned with the actual price for the test set.

```
ols_adj_training_set_results <- training_set_results %>% mutate(adjusted_ols = case_when(
    `ols_predicted_price` > 1700000 ~ ols_predicted_price*1.4,
    `ols_predicted_price` > 1500000 ~ ols_predicted_price*1.2,
    `ols_predicted_price` > 1300000 ~ ols_predicted_price*1.05,
    `ols_predicted_price` > 1150000 ~ ols_predicted_price*.99,
    `ols_predicted_price` > 1000000 ~ ols_predicted_price*.98,
    `ols_predicted_price` > 900000 ~ ols_predicted_price*.92,
    `ols_predicted_price` < 0 ~ 200000,
    `ols_predicted_price` < 200000 ~ ols_predicted_price*1.5,
    `ols_predicted_price` < 500000 ~ ols_predicted_price*.90,
    `ols_predicted_price` < 900000 ~ ols_predicted_price))
ols_adj_training_set_results
```

Note: sometimes the OLS predicted a negative home value which isn't a reasonable estimate so those homes were adjusted to the low end of the actual home price spectrum in the training set.



The results of those adjustments yielded the distribution in the figure above. There is high variability in the OLS model, however the adjusted model had a more evenly distribution of points on either side of the line for the whole range. This adjustment yielded prediction results that were 1.256% more accurate than the non adjustment ols prediction. This accuracy metric was determined by the mean of the absolute value of % price difference of the estimate from the actual price.

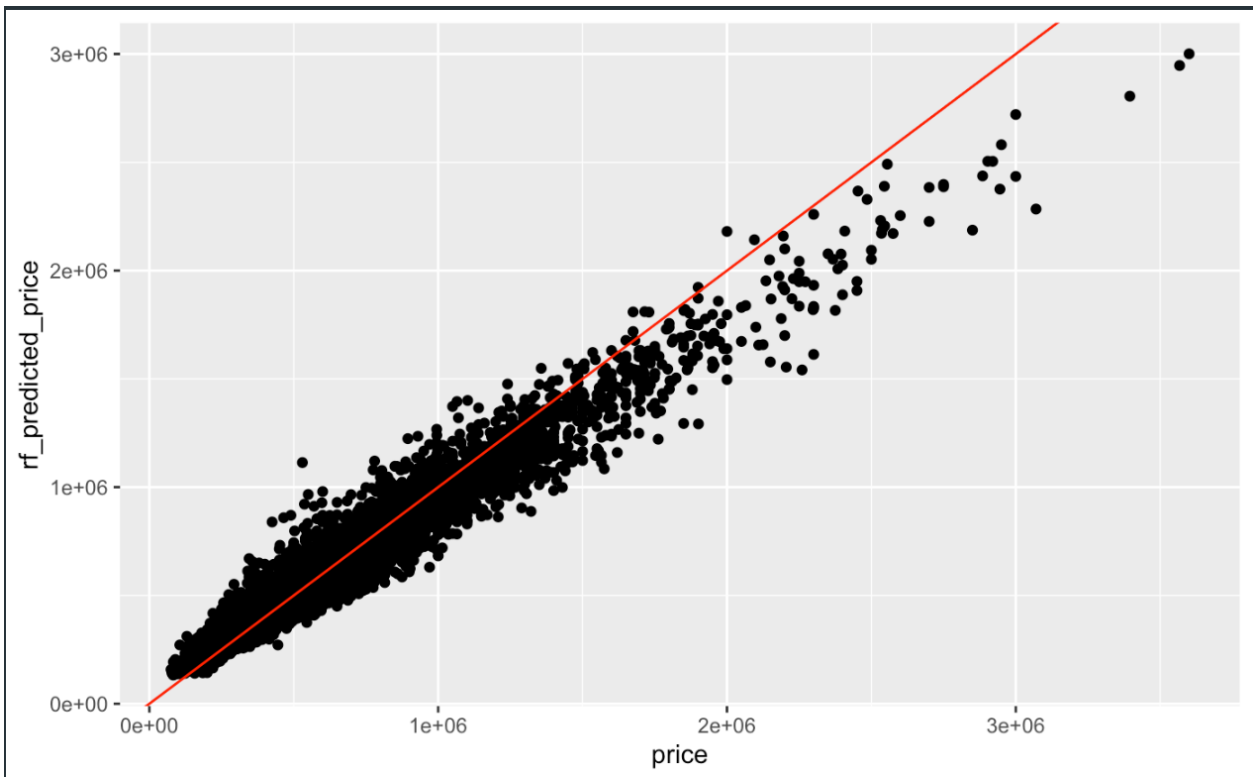
The regression was then applied to test set 1 and then the ols prediction mutation/adjustment was then applied so the predictions were less likely to be undervalued/overvalued based on the

#2

The machine learning prediction followed a very similar process to adjusting the OLS prediction for test set 1. Firstly the randomForest regression had price on the left and

baths + lot_sqft + has_waterfront_view + unit_condition + floors_above_ground_level_sqft + construction_year + avg_living_area_sqft_of_nearby_neighborhoods + beds + living_area_sqft + living_area_sqft + floors + rating_of_view + construction_and_design_grade + basement_sqft +most_recent_renovation +avg_lot_sqft_of_nearby_neighborhoods

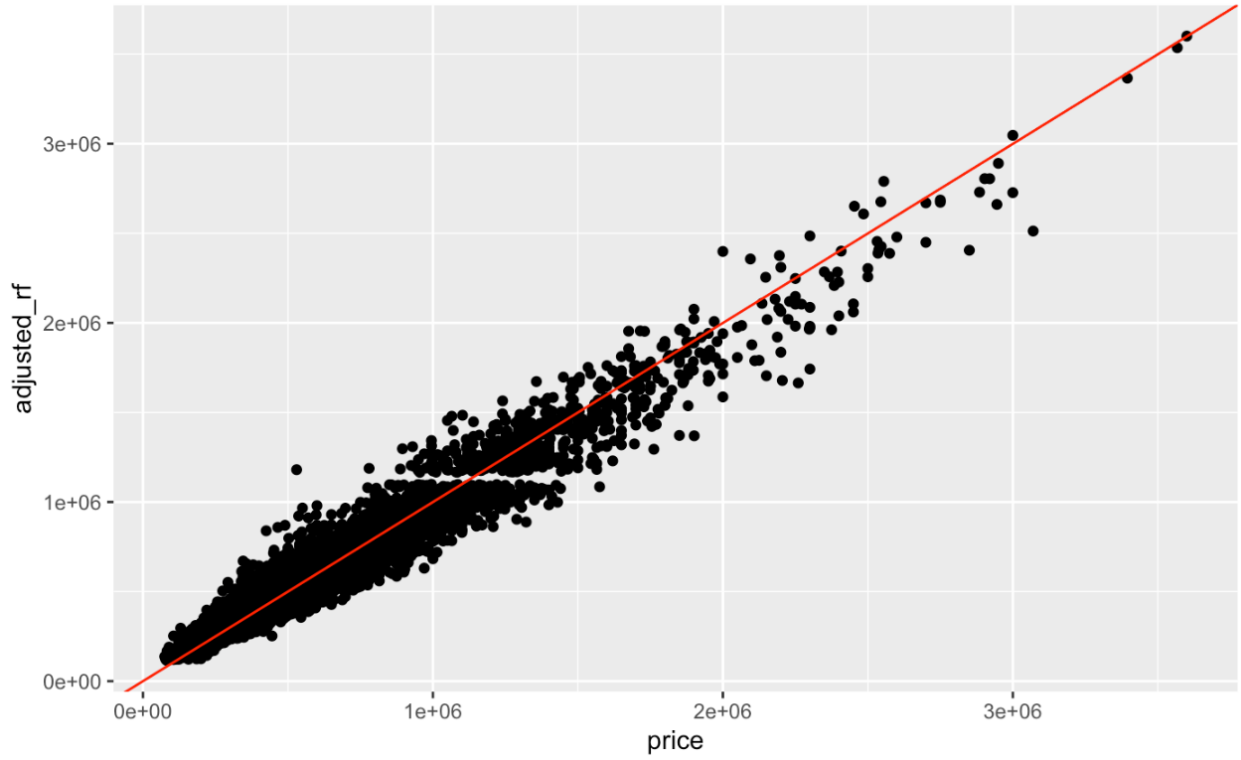
On the right. The same non-descriptive variables were left out of the machine learning prediction as they would not benefit the regression.



Similar patterns emerged where the prediction undervalued the higher priced houses and undervalued the lower priced houses. An adjustment was made so the randomForest machine learning predicted house prices were more closely priced to the true price using the following adjustments:

```
rf_adj_training_set_results <- training_set_results %>% mutate(adjusted_rf = case_when(
    `rf_predicted_price` > 2750000 ~ rf_predicted_price*1.2,
    `rf_predicted_price` > 2300000 ~ rf_predicted_price*1.12,
    `rf_predicted_price` > 2000000 ~ rf_predicted_price*1.1,
    `rf_predicted_price` > 1500000 ~ rf_predicted_price*1.08,
    `rf_predicted_price` > 1300000 ~ rf_predicted_price*1.06,
    `rf_predicted_price` > 1100000 ~ rf_predicted_price*1.06,
    `rf_predicted_price` > 9000000 ~ rf_predicted_price*1,
    `rf_predicted_price` < 200000 ~ rf_predicted_price*.87,
    `rf_predicted_price` < 300000 ~ rf_predicted_price*.93,
    `rf_predicted_price` < 500000 ~ rf_predicted_price*.95,
    `rf_predicted_price` < 9000000 ~ rf_predicted_price))
rf_adj_training_set_results
```

Those adjustments yielded a prediction that was closer to an even distribution of over predicting and under predicting for any given price point. The adjusted prediction was 1.029% closer to the actual price than the unadjusted prediction.



The rf prediction was then applied to the data and subsequent adjustments to create the ML predicted price.

#3

In data exploration, I determined that test set 2 contained much bigger, higher end homes than the training set. From the adjustments made in part 2, I determined that the randomForest under predicted values for higher cost homes. To make more accurate predictions, using machine learning as the predicted values were much closer to actual values for the training set, I created

a subsection of the training set that was more representative of the homes found in test set 2. In making the randomForest machine learning model, I was able to determine which variables have the most impact on the mean square error of the predicted price.

	%IncMSE	IncNodePurity
baths	31.59565	7.703757e+13
lot_sqft	59.11016	7.423446e+13
has_waterfront_view	23.03641	2.763997e+13
unit_condition	30.55361	2.616854e+13
floors_above_ground_level_sqft	27.25617	1.369842e+14
construction_year	102.40547	1.674322e+14
avg_living_area_sqft_of_nearby_neighborhoods	72.14709	1.994166e+14
beds	29.21420	2.644773e+13
living_area_sqft	39.62522	2.779035e+14
floors	20.93355	2.275389e+13
rating_of_view	30.26820	6.430951e+13
construction_and_design_grade	58.26308	3.163487e+14
basement_sqft	35.57104	5.750061e+13
most_recent_renovation	17.81178	1.762744e+13
avg_lot_sqft_of_nearby_neighborhoods	81.24757	8.599292e+13

I identified 4 variables that were intrinsic to the quantification of a house price based. Those were living area (39.6%IncMSE), lot_sqft (59.1%IncMSE), construction and design grade(58.3%IncMSE), and construction year (102.4%IncMSE). Since these variables carried the highest importance in determining the predicted price of the home, I filtered the training set to be more like the test set in these variables, as there will be scaling in the other variables as we filter these variables.

```
filter(living_area_sqft > 3600,
       lot_sqft>8200,
       construction_and_design_grade >9,
       construction_year > 1950)
```

These variables were filtered so the training set more closely resembled the test set 2. This filtering identified the higher end homes. A perfect sample training set of the test set wasn't practical nor possible given the data available in the training set; however a training set that more closely resembled test set 2 would more closely predict the accurate values of the homes through the range of the test set 2 homes. From the initial accuracy of the prediction tests in part 2, the randomForest was good at predicting values in the mid range but wasn't as accurate on the low end and high end distribution of home prices. The goal of the filtering was to shift the midrange up to higher valued homes (which typically have larger living areas, lots, higher

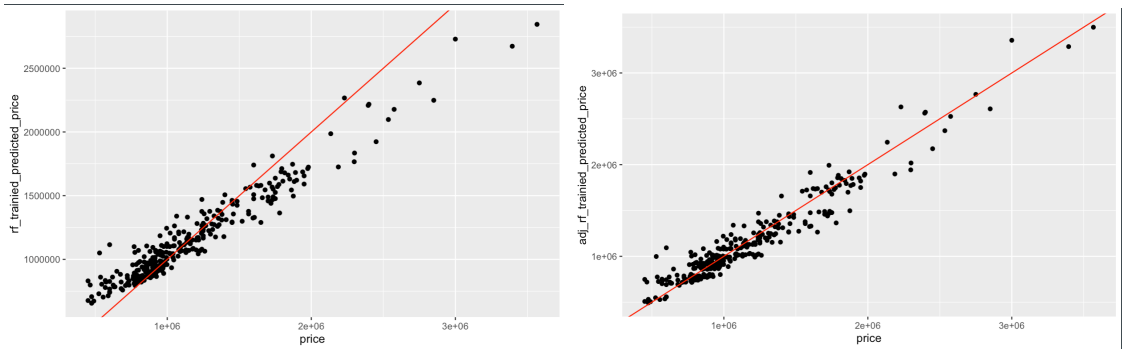
construction and design grades, and newer construction years) so it could more accurately predict the midrange homes of test set 2. “Matching” the training set to the test set included looking at the range, IQR, median, and mean. I wanted to retain at least 300 data points for the randomForest to use for determining the prediction so I had to balance the removal of data through filtering.

This shift, along with the coefficient adjustment Reduced the mean square error of the predicted prices from the actual prices of test set 2. The RF overvalued homes on the low end and undervalued homes on the high end so the adjustment was necessary to get more accurate predictions through the range.

The following coefficients were placed on the predicted prices

```
TS2$pred_not_adjusted <-predict(TS2_rf, TS2)
TS2 <- TS2 %>% mutate(pred = case_when(
    `pred_not_adjusted` > 2600000 ~ pred_not_adjusted*1.23,
    `pred_not_adjusted` > 2100000 ~ pred_not_adjusted*1.16,
    `pred_not_adjusted` > 1900000 ~ pred_not_adjusted*1.13,
    `pred_not_adjusted` > 1500000 ~ pred_not_adjusted*1.1,
    `pred_not_adjusted` > 1300000 ~ pred_not_adjusted*1,
    `pred_not_adjusted` > 1100000 ~ pred_not_adjusted*.98,
    `pred_not_adjusted` > 900000 ~ pred_not_adjusted*.95,
    `pred_not_adjusted` > 750000 ~ pred_not_adjusted*.90,
    `pred_not_adjusted` < 750000 ~ pred_not_adjusted*.75))
TS2
```

The original prediction is on the left and the adjusted prediction is on the right.



This adjustment yielded predictions for the training set that were 2.73% more accurate than the not adjusted predictions. The new random forest prediction, derived from the training set that

was specially filtered to more closely match homes in test set 2, were applied to test set two and the adjustments were applied. This yielded the predicted prices for homes in test set 2.