Predicting The Price of A House

Ornery Outliers: Brittney Miller, Samantha Rogers, Daniel Geib, Yang Tsai

April 18, 2020

1 Introduction

In 2019 alone, 5.34 million homes were purchased in the United States. (Rudden, 2020) Every year, buyers and sellers are asking themselves, "What is the best price for this house?". Buyers want to make sure they are getting the best value and price on a house while sellers generally want as much money as possible for their previous home. How can both buyers and sellers tell what is a "good deal" on the price of a house for both parties? We have decided to use the data set, "House Price Prediction", from Kaggle.com to attempt to create a model that accurately predicts the value of a house. Buyers and sellers could then use that prediction to tell whether the price they are paying or being paid is fair.

2 Data

In our data set we have several qualitative and quantitative variables that will be analyzed. To elaborate on the variables, the variables and meaning are as follows:

Variable Name	Description	Variable Range
date	date house was sold	
price	dollar amount the house was sold at	$0 to $26,590,000
bedrooms	Number of bedrooms in the house	0 to 9
bathrooms	Number of bathrooms in the house	0 to 8
sqft_living	square feet measurement of living space	370 to 13,540
sqft_lot	square feet measurement of land plot	638 to 1,074,218
floors	the number of floors in the house	1 to 3.5
waterfront	Is it a waterfront home	No=0 , Yes=1
view	the score of the view from the house	0 to 4
condition	the condition of the house	1 to 5
sqft_above	square feet measurement of upper floors	0 to 9410
sqft_basement	square feet measurement of basement	0 to 4820
yr_built	the year that the house was built	1900 to 2014
yr_renovated	year the house was renovated (0 = no renovation)	N/A
street	the street that the home is on	N/A
city	the city that the home is in	N/A
statezip	the state and zip code of the house	N/A
country	the country the house is in	USA

Table 1: Variable Description

We will be using a multiple linear regression analysis to create the model of house price, however, in order to use these variables in the final model, we must verify that model assumptions of normality, identically distributed data, and constant variance are satisfied.

Our preliminary investigation identifies that this data set is not normally distributed, as seen from the Q-Q plot in Figure:1. There is a long-tailed distribution which means that the data contains more extreme values than expected if the distribution of our data is normal. Additionally there exists non constant variance as shown in Figure:2. Our residual plot shows an unbalance about the Y-axis. In general, the house prices are fairly constant, but we do have some extreme house prices (Figure: 3) that are distorting our data set. There is also some evidence of heteroscedasticity which shows that as the predicted value increases, the variance also increases. There is evidence of outliers and influential points as shown in (Figure: 4).

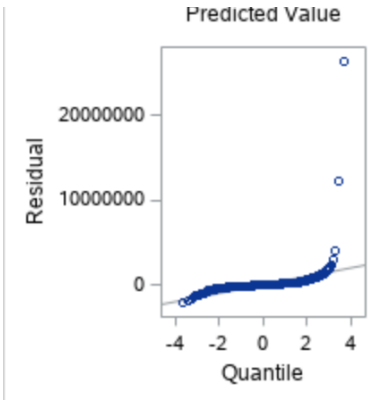

Figure 1: Q-Q Plot

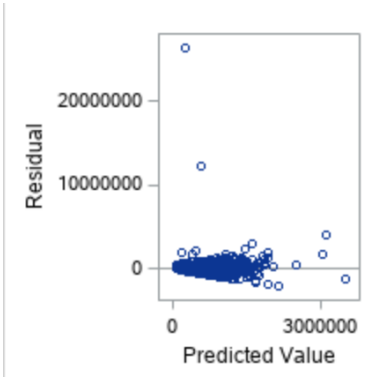

Figure 2: Predicted value against Residuals

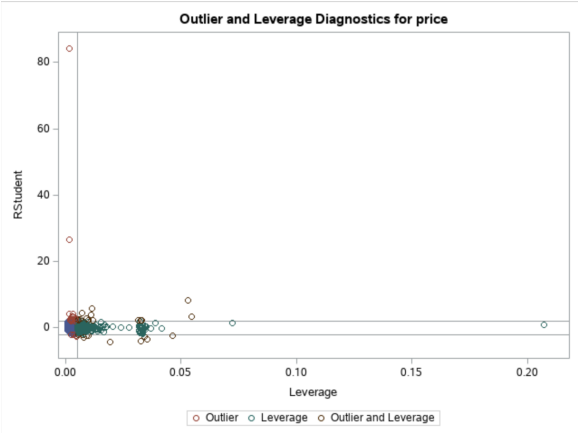

Figure 3: Outlier and Leverage Diagnostics for price

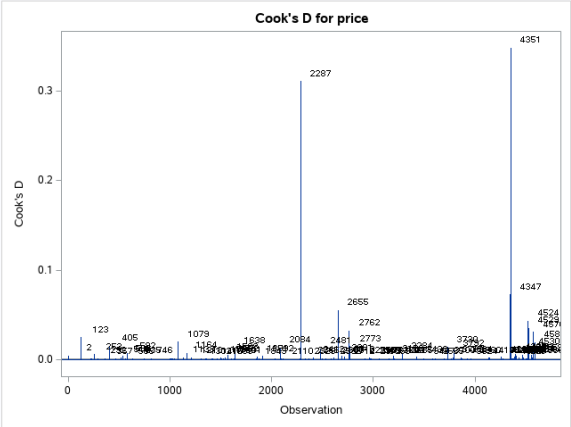

Figure 4: Cook's D Influential Points

3 Remedial Measures

We first start by randomly withholding 20% of the observations for our test set. We then designate the remaining 80% of the data as the training set. We will then use the 20% with the final model to test the accuracy of our predictions.

3.1 Initial Variable Elimination

Our data set contains 17 explanatory variables for analysis. We have a variable with the date sold, 13 quantitative variables, and four qualitative variables. The problem with all these different variables is that there are many unique categorical inputs for date, street, city, and statezip. Between the aforementioned variables, we would have to create 4,718 dummy variables in order to observe the effects they have on house price. The complicated problem arises now that we have more predictor variables than we have observations (n = 4,600). There are ways we can trim down the amount of dummy variables in our data such as translating date to numerical values or by using a penalized regression approach that would allow us to proceed with variable selection even when the number of predictor variables are greater than the number of observations. However, that process is incredibly tedious and would be counter-productive to our goal of being able to predict the more complicated variable of total price using simple variables. After considering our goal and the scale of this project, we have decided to remove the variables date, street, city, and statezip from our initial model before performing additional analysis. Because all observations are in the United States, we have removed country as a variable to avoid redundancy.

3.2 Transformation of House Price Followed by Removal of Outliers

As shown in Figure:3, there exist extreme outliers with respect to both the X and Y axis. The range house prices without the removal of outliers is \$0 to \$26,590,000. If we were to fit a prediction model to our current data set, the few extremely expensive houses would cause our model to overestimate the house price of new data. This would make the model unreliable. There are 49 houses of varying sizes that are sold with no money transaction. Fitting a model to the data which included these extreme values, would be unreliable.

To fix this, we remove the 49 houses sold for 0\$. We also decided to remove the three houses which were sold for a value above \$5,000,000. This forceful approach is necessary because these houses are sold at an abnormal value. The response variable price takes on values from 0\$ to \$26,590,000. Removing these abnormal sales will restrict the value of price to be between \$7,800 and \$5,000,000.

Upon the removal of these sales from our data, The fit diagnostics are greatly improved. See Figure:5. However, the critical assumptions of Normality and constant variance, are still not met. Therefore the data needs more manipulation. Applying a log transformation, we again improve our fit as shown below in Figure:6.

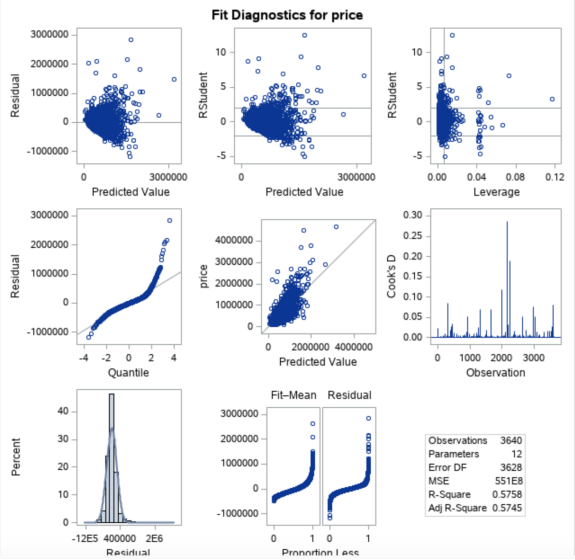

Figure 5: Fit Diagnostics after removed outliers

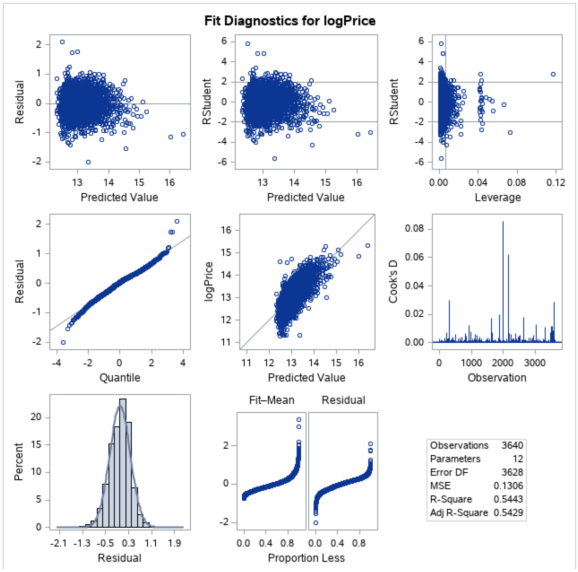

Figure 6: Log transformation

3.3 Influential Points

Influential Points are outliers in the x-axis which affect the accuracy of the parameter estimates but they do not affect the accuracy of the predictive power of our model. As shown in Figure:3, there are multiple influential points in the data set. To address the possible root of our problem, we analyze the yr_renovated variable. This variable is a major factor in the problems of the model. Its input is based on the year that a house is renovated, however, if a house has never been renovated, the input is zero. Suddenly the majority of the houses are at zero and other houses are around 1900 to 2000. That is a huge gap between the two group of points which pushes some of the observations to the edge of the plot. To fix this, we have replaced the yr_renovated variable with a categorical variable called renovation. Doing so ensures that the two group of points are not 2000 units apart and pull the influential points back in. We have also log transformed three additional variables (sqft_lot, sqft_above, and sqft_basement) which eliminated the biggest influential point. Even after all of the transformations, there are still influential points remaining. It would not be appropriate, probable, or valid to delete or fix the many influential points that exist so we are unable to completely fix the influential points in the x-axis. When we obtain the coefficient estimates from the final model, the interpretations of the coefficient will be unreliable because the coefficients will be unstable and change with new data.

3.4 Multicollinearity

Collinearity Diagnostics										
			Proportion of Variation							
Number	Eigenvalue	Condition Index	Intercept	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view
1	8.66592	1.00000	0.00000146	0.00049522	0.00044515	2.25475E-14	0.00234	0.00069883	0.00021069	0.00139
2	1.30944	2.57256	7.310833E-7	0.00014770	0.00003166	3.87688E-16	0.00077864	0.00046108	0.32454	0.21658
3	0.80892	3.27307	6.877794E-7	0.00015567	0.00001972	1.74975E-15	0.82208	0.00006448	0.00534	0.00094541
4	0.72154	3.46559	0.00000109	0.00000462	8.027873E-7	9.04951E-15	0.00060707	0.00230	0.43847	0.12057
5	0.64228	3.67321	3.48769E-10	0.00021298	0.00153	5.45535E-14	0.07040	0.00290	0.00008897	0.00278
6	0.50364	4.14808	9.423067E-8	0.00011210	0.00004682	1.13267E-14	0.02235	0.00202	0.22526	0.59886
7	0.19926	6.59475	0.00006243	0.00027575	0.00764	1.67128E-12	0.02551	0.00002569	0.00072218	0.03093
8	0.06605	11.45397	0.00000280	0.03099	0.02640	1.43605E-12	0.03822	0.55431	0.00000539	0.00046989
9	0.03743	15.21652	0.00002151	0.51440	0.21555	3.2696E-12	0.01330	0.06577	0.00194	0.01748
10	0.02931	17.19439	0.00000911	0.41836	0.64529	2.17935E-13	0.00072113	0.20537	0.00153	0.00063862
11	0.01616	23.15515	0.00172	0.02921	0.01510	8.43419E-14	0.00222	0.16055	0.00072258	0.00410
12	0.00005839	385.23845	0.99818	0.00563	0.08795	1.34569E-16	0.00146	0.00553	0.00117	0.00525
13	1E-12	2943793	1.73636E-16	0	0	1.00000	0	0	0	0

Collinearity Diagnostics					
	Proportion of Variation				
Number	condition	sqft_above	sqft_basement	yr_built	yr_renovated
1	0.00031491	2.88633E-14	1.73831E-13	0.00000151	0.00245
2	0.00012259	3.45933E-15	1.2956E-12	7.742259E-7	0.00075937
3	0.00015722	1.9011E-14	1.19647E-12	6.555388E-7	0.02976
4	0.00005563	4.11636E-14	1.94772E-11	0.00000118	0.00703
5	0.00002365	1.3406E-13	5.08094E-13	1.545419E-8	0.57319
6	0.00001736	4.61385E-14	2.27971E-11	1.069697E-7	0.02643
7	0.03336	2.96662E-12	1.42209E-14	0.00005993	0.09804
8	0.03942	3.42905E-12	1.08853E-11	0.00000113	0.01112
9	0.04685	4.78602E-12	9.76622E-12	0.00001692	0.00025490
10	0.02801	2.20126E-13	4.34601E-12	0.00001346	0.01291
11	0.67128	1.0953E-13	6.00175E-13	0.00197	0.11964
12	0.18039	1.08747E-14	1.4062E-12	0.99794	0.11841
13	0	1.00000	1.00000	1.76697E-16	0

Figure 7: Multicollinearity

Multicollinearity occurs when two explanatory variables have a linear relationship. When multicollinearity is present in the model, it causes problems with accurate fitting and interpretation. Furthermore, variable selection methods may throw out variables that shouldn't be thrown out or keep variables that should not be kept. All of which prevents us from finishing an accurate regression analysis. In order to diagnose multicollinearity we need to look at the variance inflation factor and the condition index. If the VIF value is much higher than 10 it indicates that there is a multicollinearity problem. If the conditional index is much greater than ten and the proportion of variance is greater than 50% in two or more variables, then the same problem of multicollinearity arises. Figure:7 shows that there are multiple VIF values far greater than 10 which shows that there is indeed a multicollinearity problem. The way to fix multicollinearity is to find which explanatory variables shares a linear relationship with each other then remove one or more variables. Figure 8 shows us that log-sqft_living and log-sqft_above share a linear relationship. Since the square foot measurement of the living space is arguably more useful than the square foot measurement of above, we have opted to remove logAbove from our model and thereby fix the problem with multicollinearity.

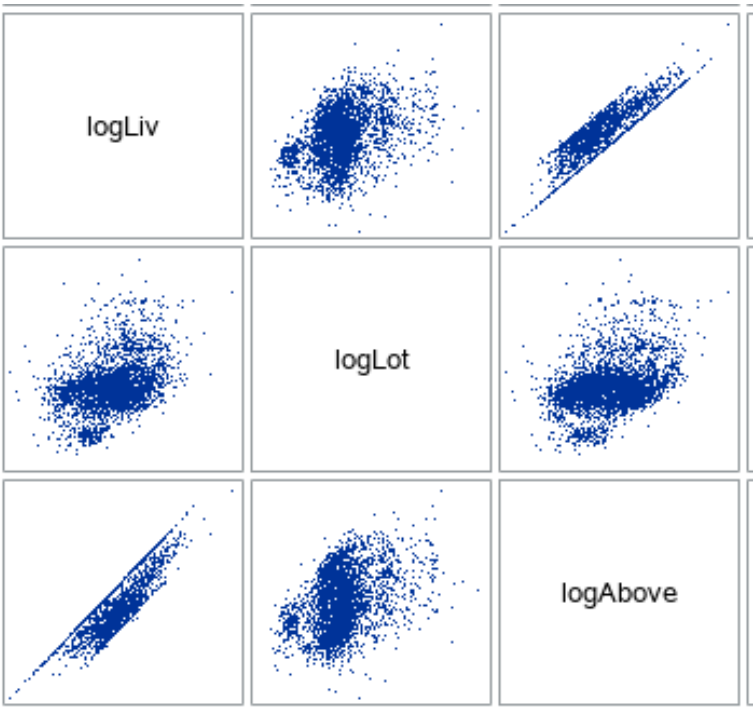

Figure 8: Scatter plot of logLiv and logAbove

3.5 Variable Selection

We attempted 3 different variable selection methods, Criterion Selection, Stepwise Selection, and LASSO.The results of Criterion Selection, as shown in figure:9, compares all possible combination of variables and finds the best combination determined by adjusted R-square, C(p), AIC, and SBC. We want a model with the C(p) to be about equal to the number of variables in the model plus one with a small AIC and small SBC, and with a large R-squared. We find the model which best satisfies this criteria includes the following 11 variables: bedrooms, bathrooms, log_sqftliving, log_sqft_lot, floors, waterfront, view, condition, log_sqft_basement, yr_built, and renovation. The problem with Criterion Selection is that there are too many combinations to consider all at once, so therefore we are more inclined to use this selection as a comparison to other selective methods.

Number in Model	Adjusted R-Square	R-Square	C(p)	AIC	SBC	Variables in Model
11	0.5544	0.5557	12.0000	-7460.1943	-7385.80402	bedrooms bathrooms logLiv logLot floors waterfront view condition logBase yr_built renovation
10	0.5541	0.5553	13.5042	-7458.6802	-7390.48911	bedrooms bathrooms logLiv logLot floors waterfront view condition yr_built renovation
10	0.5539	0.5551	15.2000	-7456.9809	-7388.78978	bedrooms bathrooms logLiv logLot floors waterfront view condition logBase yr_built
9	0.5536	0.5547	16.6094	-7455.5667	-7393.57477	bedrooms bathrooms logLiv logLot floors waterfront view condition yr_built
10	0.5535	0.5548	17.8921	-7454.2847	-7386.09362	bedrooms bathrooms logLiv logLot floors view condition logBase yr_built renovation
9	0.5532	0.5543	19.7056	-7452.4689	-7390.47697	bedrooms bathrooms logLiv logLot floors view condition yr_built renovation
9	0.5530	0.5541	21.1294	-7451.0452	-7389.05331	bedrooms bathrooms logLiv logLot floors view condition logBase yr_built
8	0.5527	0.5537	22.8444	-7449.3333	-7393.54057	bedrooms bathrooms logLiv logLot floors view condition yr_built
9	0.5513	0.5524	35.0766	-7437.1290	-7375.13712	bedrooms bathrooms logLiv logLot floors waterfront view logBase yr_built
10	0.5512	0.5524	36.9229	-7435.2821	-7367.09104	bedrooms bathrooms logLiv logLot floors waterfront view logBase yr_built renovation
8	0.5511	0.5521	35.6920	-7436.5253	-7380.73262	bedrooms bathrooms logLiv logLot floors waterfront view yr_built
9	0.5510	0.5521	37.5282	-7434.6884	-7372.69646	bedrooms bathrooms logLiv logLot floors waterfront view yr_built renovation
8	0.5505	0.5515	40.3525	-7431.8903	-7376.09764	bedrooms bathrooms logLiv logLot floors view logBase yr_built
9	0.5504	0.5515	42.1720	-7430.0697	-7368.07782	bedrooms bathrooms logLiv logLot floors view logBase yr_built renovation
7	0.5503	0.5512	41.2368	-7431.0248	-7381.43129	bedrooms bathrooms logLiv logLot floors view yr_built
8	0.5502	0.5512	43.0443	-7429.2160	-7373.42325	bedrooms bathrooms logLiv logLot floors view yr_built renovation

Figure 9: Criterion Selection

The results of Step-wise Variable selection are displayed in figure:10. The Step-wise method added 10 explanatory variables to the model. In a step-wise selection, variables are entered into the model one at a time based on the lowest p-value. Then the p-value of the variable is evaluated again to determine if any variables needs to be removed. We have chosen the entrance and exit thresholds to be p-value of 0.05. The step-wise selection is convenient in comparing the criterion selection method; however, the resulting model from a step-wise selection process may not be reliable to predict new data set purely because some variables are significant by coincidence and other variables that should be statistically significant may instead be discarded.

Summary of Stepwise Selection								
Step	Variable Entered	Variable Removed	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	logLiv		1	0.4562	0.4562	804.653	3049.69	<.0001
2	yr_built		2	0.0283	0.4845	575.586	199.62	<.0001
3	floors		3	0.0239	0.5083	382.741	176.45	<.0001
4	view		4	0.0189	0.5272	230.459	145.27	<.0001
5	bedrooms		5	0.0079	0.5351	168.362	61.35	<.0001
6	bathrooms		6	0.0104	0.5455	85.2287	83.34	<.0001
7	logLot		7	0.0056	0.5512	41.2368	45.57	<.0001
8	condition		8	0.0025	0.5537	22.8444	20.31	<.0001
9	waterfront		9	0.0010	0.5547	16.6094	8.22	0.0042
10	renovation		10	0.0006	0.5553	13.5042	5.10	0.0240

Figure 10: Stepwise Variable Selection

The LASSO selection technique results are displayed in figure:11. LASSO is a penalized regression approach which adds bias to our model and decreases variance. Penalized

regression methods are useful because, though our predictions from the resulting model would be an underestimate or an overestimate, our model will do much better in predicting with a new data set due to the decrease in variance. We chose LASSO because LASSO forces the coefficients of unnecessary variables to zero which results in a more simple model. In this case, we have nine predictor variables remaining, which is less than the first two methods. These variables are logLiv, yr_built, view, floors, bathrooms, bedrooms, condition, and waterfront. For our final linear regression model, we will be using the model with the variables recommended from the LASSO regression.

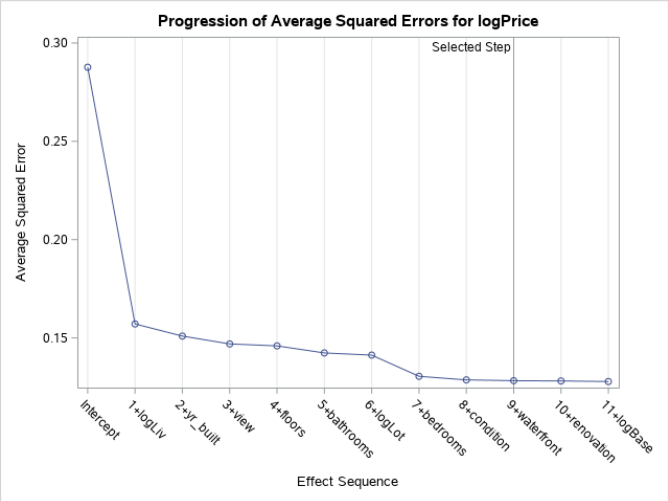

Figure 11: LASSO Variable Selection

3.6 Interaction terms

In a linear model, an interpretation of the coefficient is the unit of increase dependent on the associated explanatory variable while holding all the other variables constant. However an interaction occurs when there are variables that are influenced by other variables such that a variable held constant may increase or decrease due to change in other variables. For our model we consider three second-order interaction terms. The first variable bedFloor is the interaction between bedrooms and floors. The reasoning is that bedrooms would increase if there are more floors. The second variable is bathFloor, the interaction term between bathrooms and floors. It follows the reasoning of the first, more floor equates to more bathrooms. The last interaction term we considered is yrCon. This is the interaction term between yr_built and condition of the house. The logic behind this choice is that the condition of the house naturally worsens overtime depending on the age of the house; therefore, we would expect older houses to be in a worse condition. So we added these three terms to our model and ran a new analysis and found that no interaction terms are significant enough to keep. As seen in Figure: 12 the most significant interaction term is yrCon with a p-value between 0.05 and 0.06. But we are using the common significance value of 0.05, so there is not enough evidence to keep any of the interaction terms.

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	9.98247	2.43074	4.11	<.0001
bedrooms	1	-0.09824	0.02396	-4.10	<.0001
bathrooms	1	0.11825	0.03379	3.50	0.0005
logLiv	1	0.37024	0.07276	5.09	<.0001
logLot	1	-0.06284	0.00824	-7.63	<.0001
floors	1	0.05893	0.04626	1.27	0.2028
waterfront	1	0.24558	0.07800	3.15	0.0017
view	1	0.07880	0.00863	9.13	<.0001
condition	1	1.36419	0.69112	1.97	0.0485
logAbove	1	0.52002	0.07105	7.32	<.0001
logBase	1	0.02710	0.00484	5.59	<.0001
yr_built	1	-0.00169	0.00124	-1.36	0.1744
renovation	1	0.03407	0.01376	2.48	0.0133
bedFloor	1	0.01187	0.01482	0.80	0.4233
bathFloor	1	-0.01200	0.01900	-0.63	0.5277
yrCon	1	-0.00066700	0.00035362	-1.89	0.0594

Figure 12: Interaction Term

3.7 Interpretation of coefficients

Parameter Estimates						
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variance Inflation
Intercept	1	15.12743	0.52451	28.84	<.0001	0
logLiv	1	0.85302	0.02563	33.28	<.0001	3.44964
yr_built	1	-0.00424	0.00025714	-16.47	<.0001	1.66172
view	1	0.08013	0.00860	9.32	<.0001	1.25945
floors	1	0.11984	0.01432	8.37	<.0001	1.68854
bathrooms	1	0.10799	0.01342	8.05	<.0001	3.09201
logLot	1	-0.05243	0.00763	-6.87	<.0001	1.36485
bedrooms	1	-0.08586	0.00877	-9.79	<.0001	1.81145
condition	1	0.04445	0.00969	4.59	<.0001	1.23116
waterfront	1	0.22510	0.07851	2.87	0.0042	1.14396

Figure 13: Parameter Estimates for Final Model

In the model there consists of influential points that cannot be removed. This makes it difficult to interpret the coefficients. The influential points make the coefficients of the model less accurate but does not take away the predictive power of the model. It is not

as useful to interpret the coefficients because when there is new data the coefficients will change. The model given here states that all houses will have a base price of 3 million and change based off of what variables the house has. This is a very large starting point and does not seem reasonable; however, the intercept of the model is not a concern because we are not interested in a house where all the variables in our model equal zero. If we were to interpret the coefficients of the variables we would see that not all of them would be accurate or make sense. The two most important variables are logLiv and yr_built because those two are the first variables entered into the model by LASSO selection. The coefficient of logLiv is 0.853. Since our model is on a log scale, we will exponentiate the coefficient to relate it to the actual house price.

$$e^{0.853} = \$2.35$$

The interpretation of logliv(the log transformation of sqft_living) is that for every square-foot of the house the price will increase on average by $2.35 while holding all other variables constant. Take the average living space size of 2130.83 square feet. The houses are worth $5,000.37 just for the size of the living space.

The next important variable is the year when house is built. The coefficient of yr_built is interestingly -0.00242.

$$e^{-0.00242} = -\$0.96$$

When looking at newer houses, we will predict that for each year a house is newer the price will be decreased by a whole dollar. This means that the more recent the house is, the lower the price is by year. The concept is counter-intuitive at first, but consider that older houses that are still around implies that those houses are well-built and have endured a long time which could be why older houses are more expensive according to our model.

4 Final Model Assumptions

For the model to be valid it has to pass the assumptions of normality, identically distributed data, and constant variance. As seen in the QQ plot(Figure:14) the data is normally distributed. We can see that there are a few outliers but not enough to say our model is not valid. From the plot of predicted values against residuals (Figure:15) we can see that there is constant variance within our data. This is a huge improvement from the raw data. As seen in the normal quantille plot(Figure:16) the data passes the assumption of normality. In the original data there was multicolinearity, but was fixed for the final model as seen in Figure: 17. The final model passes the model assumptions of normality, constant variance and identically distributed data, but because of the influential points, causing the model to be nonsensical, we will want to consider an alternative method.

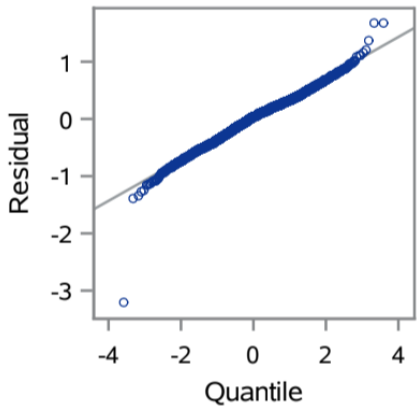

Figure 14: QQ plot for the final model

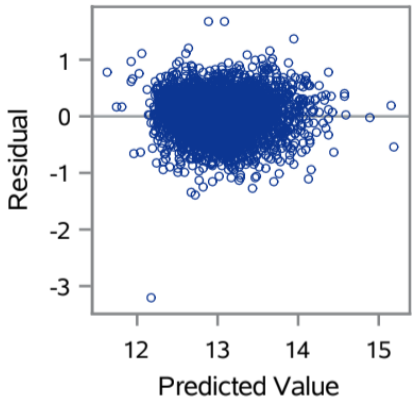

Figure 15: Residual against predicted values

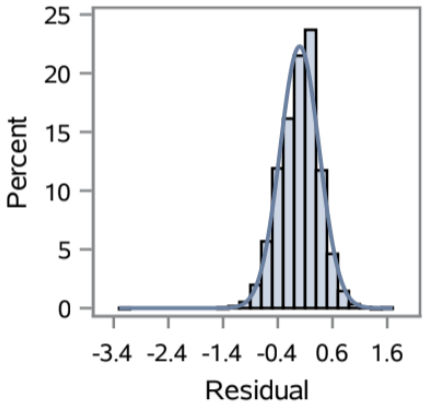

Figure 16: Normal Quantile plot

Collinearity Diagnostics										
	Proportion of Variation									
Number	Intercept	logLiv	yr_built	view	floors	bathrooms	logLot	bedrooms	condition	waterfront
1	0.00000205	0.00001503	0.00000220	0.00152	0.00098839	0.00056757	0.00011764	0.00057680	0.00047026	0.00018874
2	2.763724E-7	0.00000136	3.062017E-7	0.23581	0.00010441	4.669821E-7	0.00001061	0.00004048	0.00005322	0.39340
3	3.454088E-7	0.00000122	3.856617E-7	0.66727	0.00013966	0.00001337	0.00001483	0.00000101	0.00003598	0.59965
4	0.00003973	0.00009477	0.00003319	0.00659	0.18231	0.07992	0.00309	0.00169	0.04369	0.00002417
5	0.00003610	0.00002089	0.00003835	0.02508	0.38911	0.12737	0.00001332	0.15784	0.00105	0.00080609
6	0.00000185	0.00001250	0.00000378	0.00731	0.09192	0.41767	0.00025088	0.63941	0.06472	0.00159
7	0.00027835	0.00166	0.00035077	0.00229	0.10419	0.00024812	0.04675	0.03981	0.72050	0.00165
8	0.00338	0.00648	0.00359	0.00319	0.18151	0.00425	0.85100	0.00244	0.04533	0.00154
9	0.01514	0.97345	0.02352	0.03617	0.00678	0.23385	0.09325	0.15457	0.00031461	0.00021513
10	0.98112	0.01827	0.97247	0.01478	0.04294	0.13611	0.00550	0.00362	0.12383	0.00093681

Figure 17: Multicollinarity for the Final Model

5 Alternative Methods

5.1 Regression Tree Analysis

We have chosen Regression tree Analysis as an alternative method. Regression Tree analysis is a type of nonlinear predictive method. The first benefit of this analysis is it makes for a fast and easy prediction. Regression trees have a fast interpretation because there isn't an equation to compute, the information is all given by the tree. To make a prediction we follow the branches until we reach the end node (Figure: 18) which gives us the average pricing of all the houses in the selection. The second benefit is the ease of determining variable importance based on the timing and frequency of the split. The earlier and more splits on a variable indicates that the variable is important in determining the accuracy of our prediction. According to Figure: 19, the square-foot measurement of living space is the most important (by far) variable which we can confirm with Figure: 19. logLiv is the first variable to split and splits 13 times. Its value of importance is much greater than all other variables. The next set of important variables are yr_built and logLot.

The interpretation of the most influential variables is important in knowing how to interpret the regression tree. The most important variable in the model is logLiv (the sqft of the house) From the first split of the tree we can predict that when a house is under 2,000 sqft we will predict the average house price to be $150,000. From the right side of the first split, if a house has over 2,000 sqft we will predict an average house price of $450,000. The further down the more specific the nodes will get. In this case, we cannot see the decimal places and therefore we can not say how much more specific the nodes get. We can see in the lowest right node that a house with over 4,225 sqft will be predicted to have an average house price of over $1 million.

Variable Importance			
	Training		
Variable	Relative	Importance	Count
logLiv	1.0000	22.6276	13
yr_built	0.3083	6.9768	10
logLot	0.2355	5.3281	10
logAbove	0.1521	3.4415	7
condition	0.1290	2.9200	3
view	0.1160	2.6246	3
bedrooms	0.1066	2.4116	4
bathrooms	0.1016	2.2986	3
logBase	0.0682	1.5431	2

Figure 19: Regression Tree Analysis

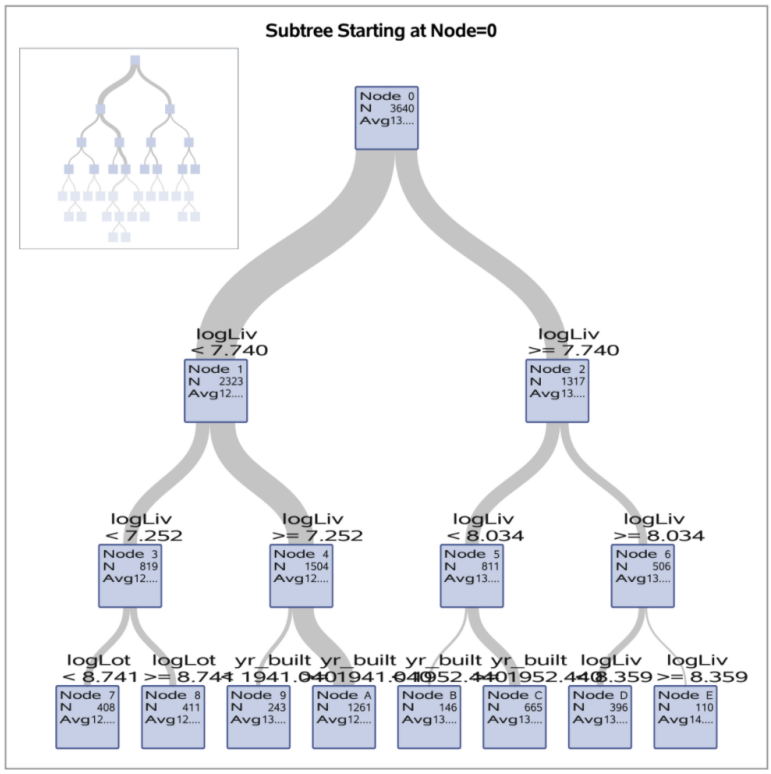

Figure 18: Regression tree

Random Forest Frid

The HPFOREST Procedure

Loss Reduction Variable Importance					
Variable	Number of Rules	MSE	OOB MSE	Absolute Error	OOB Absolute Error
logLiv	30382	0.073488	0.04005	0.082914	0.033318
logAbove	28643	0.055319	0.02088	0.065576	0.017193
bathrooms	10336	0.029720	0.01591	0.030255	0.011586
view	3998	0.008775	0.00390	0.009259	0.002635
logBase	18503	0.021551	0.00271	0.030015	0.002495
floors	7739	0.006630	0.00053	0.010368	0.001033
waterfront	399	0.001044	0.00019	0.001023	0.000107
bedrooms	6120	0.006427	-0.00188	0.009935	-0.001370
renovation	11668	0.003448	-0.00374	0.009136	-0.003191
condition	10901	0.008240	-0.00387	0.012206	-0.002313
yr_built	37535	0.029443	-0.00935	0.055362	-0.003431
logLot	42351	0.030704	-0.01374	0.058495	-0.007870

Figure 20: Random Forest Analysis Table

5.2 Random Forest

In the most general sense, random forest is essentially the combination of multiple regression trees fitted to randomly chosen samples of our data set. The random forest takes the averages not over a single node but over the entire group of trees which will make our model more accurate. The result of random forest (Figure: 20) shows that living space is the most important variable (same as regression tree analysis) so we can be sure that the square-foot measurement of the living space is the best variable to predict the house price.

6 Accuracy of Our Regression Analysis

We have stated that the interpretation of our linear model is not reliable due to the presence of influential points that make our parameter estimates unstable; however, our predictive ability is not affected by the influential points. We can test the accuracy of our model by determining the average square error of our prediction.

In the beginning, we stated that we have withheld 20% of our data for the purpose of testing. We now use the test set to calculate the average square error of our model. The average square error is how closely our model fits our test data set. The smaller the ASE,

the better our model predicts the data set. Using 20% of our data set, we can validate our model's ability to predict on future data. This determines how reliable and practical our model is.

The average square error is 0.142 for the linear model, 0.162 for the regression tree, and 0.026 for the random forest. Surprisingly, our linear model is more accurate than the single tree; however, the random forest prediction is by far more accurate than the other methods.

7 Conclusion

Having a model that can accurately predict house prices can help people selling the homes as well prospective home buyers to be able to accurately predict the actual cost of a house. This could help sellers to accurately price their houses as well as gives data as to why they are asking for a specific price. This could also help potential buyers in knowing whether the price the home is listed as is actually a reasonable price. Banks or insurance companies can also use this model to evaluate the value of a home they are providing loans for or insurance for. Overall, there are many different uses that this model can apply to for both consumers and companies.

Moving forward what will we consider for future research? One point of interest to consider is; Generating a model from the following year's data 2021, to see if this model is consistent with 2020's model. For a second research method it would be beneficial to look into remodeled houses. Remodeling a house will increase the worth but it would be interesting to consider how much the price increase will be. When remodeling an older house we predict that it will increase the house price more than if a newer house was remodeled. It would be interesting to look into our prediction and see if it is valid. This could help buyer's and sellers in their decision on weather it is beneficial to remodel the house before selling or buying.

8 Sources

Rudden, J. (2020, March 02). Existing home sales in the U.S. 2005-2021. Retrieved April 14, 2020, from https://www.statista.com/statistics/226144/us-existing-home-sales/

9 APPENDIX

9.1 Data set Import

```
/* This first line of code will need to be changed */
FILENAME REFFILE '/home/u42023602/Ornery Outliers/data.csv';
PROC IMPORT DATAFILE=REFFILE replace
DBMS=CSV
OUT=house;
GETNAMES=YES;
RUN;
```

9.2 Useful Information about Our Data Set

```
Proc Means Data = house;
Var price bedrooms bathrooms sqft_living sqft_lot floors waterfront
 view condition sqft_above sqft_basement yr_built yr_renovated;
Run;
```

9.3 Visuals of Crude Data

```
/* Box Plot */
proc sgplot data=house;
  vbox price;
title1 'House Price Data';
run;

/* Histogram */
proc univariate data=house;
 var price;
 histogram price;
run;

/* Scatterplot Matrix (Reduced) */
proc sgscatter data=house;
matrix price bedrooms bathrooms floors / markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
proc sgscatter data=house;
matrix price sqft_living sqft_lot sqft_above sqft_basement
/ markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
proc sgscatter data=house;
matrix price waterfront view / markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
proc sgscatter data=house;
```

```
matrix price condition/ markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
proc sgscatter data=house;
matrix price yr_built yr_renovated / markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
```

9.4 Data Analysis using Full Initial Model with Multicollinearity Test

```
proc reg data=house plots=(CooksD RStudentByLeverage DFFITS DFBETAS);
model price = bedrooms bathrooms sqft_living sqft_lot floors waterfront
 view condition sqft_above sqft_basement yr_built yr_renovated / vif collin;
output out=houseOut1 r=resid p=pred;
title1 'Check Initial Model';
run;
proc reg data=house plots(label)=(CooksD RStudentByLeverage DFFITS DFBETAS);
model price = bedrooms bathrooms sqft_living sqft_lot floors waterfront
 view condition sqft_above sqft_basement yr_built yr_renovated;
run;

%resid_num_diag(dataset=houseOut1, datavar=resid, label='residual',
predvar=pred, predlabel='predicted');
run;
```

9.5 Transform Data Set

```
data tran; set house;
if price > 5000000 then delete;
if price < 1000 then delete;
renovation = yr_renovated;
if yr_renovated > 0 then renovation = 1;
logPrice = log(price);
logLiv = log(sqft_living);
logLot = log(sqft_lot);
logAbove = log(sqft_above);
logBase = log(sqft_basement + 1);
run;
```

9.6 Separation into Training and Test Sets

```
/* Separate Into Training and Test Sets.
Only Fit Models to the Training Set. The variable
"Selected" separates training (0) from test (1) */
proc surveyselect data=tran seed=12345 out=houseSelect
   rate=0.2 outall; /* Withold 20% for validation */
```

```
run;

data train; set houseSelect;
if Selected = 0;
run;

data test; set houseSelect;
if Selected = 1;
run;
```

9.7 Useful Information of Transformed Data

```
Proc Means Data = train;
Var price bedrooms bathrooms sqft_living sqft_lot floors waterfront
 view condition sqft_above sqft_basement yr_built yr_renovated;
Run;
```

9.8 Visuals of Transformed Data Set

```
/* Box Plot */
proc sgplot data=tran;
  vbox logPrice;
  title1 'Transformed House Price Data';
run;

/* Histogram */
proc univariate data=tran;
 var logPrice;
 histogram logPrice;
run;

/* Scatterplot Matrix (Reduced) */
proc sgscatter data=tran;
matrix logPrice bedrooms bathrooms floors / markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
proc sgscatter data=tran;
matrix logPrice logLiv logLot logAbove logBase
/ markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
proc sgscatter data=tran;
matrix logPrice waterfront view/ markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
proc sgscatter data=tran;
matrix logPrice condition/ markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
```

```
proc sgscatter data=tran;
matrix logPrice yr_built renovation / markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
```

9.9 Useful Information of Training Set

```
Proc Means Data = train;
Var price bedrooms bathrooms sqft_living sqft_lot floors waterfront
 view condition sqft_above sqft_basement yr_built yr_renovated;
Run;
```

9.10 Visual of Training Data

```
  /* Box Plot */
proc sgplot data=train;
  vbox logPrice;
  title1 'Training House Price Data';
run;

/* Histogram */
proc univariate data=train;
 var logPrice;
 histogram logPrice;
run;

/* Scatterplot Matrix (Reduced) */
proc sgscatter data=train;
matrix logPrice bedrooms bathrooms floors / markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
proc sgscatter data=train;
matrix logPrice logLiv logLot logAbove logBase
/ markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
proc sgscatter data=train;
matrix logPrice waterfront view/ markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
proc sgscatter data=train;
matrix logPrice condition/ markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
proc sgscatter data=train;
matrix logPrice yr_built renovation / markerattrs=(symbol=CIRCLEFILLED size=2pt);
run;
```

9.11 Check Initial Training Model

```
  proc reg data=train plots=(CooksD RStudentByLeverage DFFITS DFBETAS);
model logPrice = bedrooms bathrooms logLiv logLot floors waterfront
 view condition logAbove logBase yr_built renovation / vif collin;
output out=houseOut2 r=resid p=pred;
title1 'Check Traning Model';
run;
proc reg data=train plots(label)=(CooksD RStudentByLeverage DFFITS DFBETAS);
model logPrice = bedrooms bathrooms logLiv logLot floors waterfront
 view condition logAbove logBase yr_built renovation;
run;

 %resid_num_diag(dataset=houseOut2, datavar=resid,
 label='residual', predvar=pred, predlabel='predicted');
 run;
```

9.12 Variable Selection

```
proc reg data=train;
model logPrice = bedrooms bathrooms logLiv logLot floors waterfront
 view condition logBase yr_built renovation
 / selection=AdjRSq Cp AIC SBC;
title1 'Compare Selection Criteria';
run;

proc reg data=train;
model logPrice = bedrooms bathrooms logLiv logLot floors waterfront
 view condition logBase yr_built renovation
/ selection=stepwise slentry=.05 slstay=.05;
title1 'Stepwise Selection';
run;

proc glmselect data=train plots=(criterion ase);
model logPrice = bedrooms bathrooms logLiv logLot floors waterfront
 view condition logBase yr_built renovation
/ selection=lasso(adaptive choose=sbc stop=none);
title1 'LASSO';
run;
```

9.13 Interaction Terms

```
data houseAct; set train;
bedFloor = bedrooms*floors;
bathFloor = bathrooms*floors;
```

```
yrCon = condition*yr_built;
run;
proc reg data=houseAct;
model logPrice = logLiv yr_built view floors bathrooms logLot bedrooms condition waterfr
 bedFloor bathFloor yrCon;
title1 'Check Interaction Terms';
run;
```

9.14 Final Model

```
    proc reg data=train plots=(CooksD RStudentByLeverage DFFITS DFBETAS);
model logPrice = logLiv yr_built view floors bathrooms logLot bedrooms condition waterfr
 / vif collin;
output out=houseOut2 r=resid p=pred;
title1 'Check Final Model';
run;
proc reg data=train plots(label)=(CooksD RStudentByLeverage DFFITS DFBETAS);
model logPrice = logLiv yr_built view floors bathrooms logLot bedrooms condition waterfr
;
run;

%resid_num_diag(dataset=houseOut2, datavar=resid, label='residual',
predvar=pred, predlabel='predicted');
run;
```

9.15 Accuracy Test

```
proc reg data=train noprint;
model logPrice = logLiv yr_built view floors bathrooms logLot bedrooms condition waterfr
store regModel;
title1 'Accuracy Test';
run;

/* Calculate MSPR */
proc plm restore=regModel;
 score data=test out=newTest predicted;
 run;
data newTest; set newTest;
MSE = (logPrice - Predicted)**2;
run;
proc means data = newTest;
var MSE;
run;
proc univariate data=newTest;
 var MSE;
```

```
 histogram MSE;
run;
proc print data=newTest;
where MSE > 4;
run;
```

9.16 Regression Tree

```
proc hpsplit data=train seed=123 maxdepth=8 maxbranch=2;
model logPrice = bedrooms bathrooms logLiv logLot floors waterfront
 view condition logAbove logBase yr_built renovation;
code file='/home/u42023602/Ornery Outliers/tree.sas';
/* This saves the tree to a file (need to change the path) */
output out=treeHouse;
title1 'Regression Tree Analysis';
run;

proc sgplot data=treeHouse;
scatter x=logPrice y=p_logPrice /
markerattrs=(symbol=circlefilled size=6pt);
run;

/* Call the test data and include the tree,
this will make predictions on the tree */
data scored;
set test;
%include '/home/u42023602/Ornery Outliers/tree.sas';
run;

/* Now calculate the MSPR as we did in OLS */
data testTree;
set scored;
ASE = (logPrice - p_logPrice)**2;
run;
proc means data = testTree;
var ASE;
run;
```

9.17 Random Forest

```
proc hpforest data=tran seed=134 scoreprole=oob;
input bedrooms bathrooms logLiv logLot floors waterfront
 view condition logAbove logBase yr_built renovation;
score out=treeOut;
target logPrice;
```

```
ods output FitStatistics=fitstats
VariableImportance=varimp;
title1 'Random Forest';
run;

/* Call the test data and include the tree,
this will make predictions on the tree */
data scored2;
set test;
set treeOut;
run;

/* Now calculate the MSPR as we did in OLS */
data testTree;
set scored2;
ASE = (logPrice - p_logPrice)**2;
run;
proc means data = testTree;
var ASE;
run;
```

9.18 Macro for Pearson Correlation Test and More

```
  /* Macro for Pearson Correlation Test and More */
%macro resid_num_diag(dataset,datavar,label='requested variable',
predvar=' ',predlabel='predicted variable'); title;
data shortfourplotdataset; set &dataset; label &datavar = &label;
if &datavar ne .; run; proc means data=shortfourplotdataset noprint;
var &datavar; output out=shortfourplotoutset N=nval mean=meanval;
data shortfourplotoutset; set shortfourplotoutset; xn=nval;
CALL SYMPUT('nval',xn);  xmean=meanval; CALL SYMPUT('meanval',xmean);
%global nvalue; %let nvalue=&nval; %global meanvalue;
%let meanvalue=&meanval; run; %if &predvar ne ' ' %then %do;
data shortfourplotdataset; set shortfourplotdataset;
label &predvar = &predlabel;
proc sort data=shortfourplotdataset out=shortfourplottemp;
by descending &predvar;
data shortfourplottemp; set shortfourplottemp;
shortfourplotorder = _n_;
shortfourplotgroup = 1-(shortfourplotorder < ceil(&nvalue/2));
proc means data=shortfourplottemp median noprint;
by shortfourplotgroup;    var &datavar;
output out=shortfourplotouttemp median=medresid;
run;
data shortfourplottempnew; merge shortfourplottemp shortfourplotouttemp;
```

```
by shortfourplotgroup;    d = abs(&datavar-medresid);
run;  run;   proc ttest data=shortfourplottempnew plots=none;
class shortfourplotgroup;    var d;
ods output TTests=shortfourplotBFtemp;
title1 '(Ignore this nuisance output)';  run;  run;
data shortfourplotBFtemp2; set shortfourplotBFtemp;
if method = 'Pooled';    t_BF = abs(tValue);
BF_pvalue = probt;    keep t_BF BF_pvalue;
proc print data=shortfourplotBFTemp2;
title1 'P-value for Brown-Forsythe test of constant variance';
title2 'in ' &label ' vs. ' &predlabel;
run;
%end;
proc sort data=shortfourplotdataset out=shortfourplottemp;
by &datavar; data shortfourplottemp; set shortfourplottemp;
n=&nvalue;  expectNorm = probit((_n_-.375)/(n+.25));
proc corr data=shortfourplottemp;  var &datavar expectNorm;
title1 'Output for correlation test of normality of ' &label;
title2 '(Check text Table B.6 for threshold)'; run; title; quit;
%mend resid_num_diag;
```