Predicting House Prices Using Penalized Forms of Regression

Ali Alsous[1*]

February 9, 2021

Abstract: Penalization techniques originally featured in the literature in pursuit of improving upon OLS' unsatisfactory predictive ability and interpretability, specifically in cases where the number of predictors is high. One way to ameliorate this is to leverage L1 regularization in the form of a "least absolute shrinkage and selection operator" or *lasso.* Lasso is utilized to perform feature selection on the 80+ predictors present in the Ames, Iowa Housing Dataset. Features describing home characteristics included in the dataset relate to size, age, and quality of a home's many amenities. A multiple linear regression is then constructed using the features selected by lasso and compared with a baseline regression, with the response variable being sale price. Thus, a hedonic regression is constructed that estimates the demand preferences home buyers hold for certain characteristics. Results indicate that while the lasso-selected multiple linear regression possessed greater in-sample explanatory power, the simple OLS model maintained lower error when tested on data out-of-sample.

Keywords: Ames Iowa Dataset; Lasso; House Prices; Hedonic Regression

JEL Codes: R21; R22; R31; R32

[1*] University of North Carolina Wilmington, asa2265@uncw.edu (Note: Special thanks to Lucy D. for supplying me with ample cups of coffee for the past 3 months, they certainly kept me going.)

Introduction

Explaining and understanding the demand preferences of home buyers is crucial for suppliers to be able to match most efficiently said demand. With the help of modern statistical inference, one can massage large datasets such as the Ames, Iowa housing dataset to make informed projections and match the two—supply and demand—as closely as possible. Thanks to the advent of penalization techniques, throwing "everything but the kitchen sink" into a regression is not only possible, but perhaps optimal for producing interpretable findings out of large unwieldy sets of data. The Ames, Iowa housing dataset presents an excellent opportunity to test this premise.

The data contained within the Ames dataset were first collected by the Ames City Assessor's Office and later acquired, cleaned, and published by De Cock (2011) to provide students and instructors an alternative to the Boston housing dataset. The dataset is rich, containing about 2920 observations and 79 explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 19 continuous) related to assessing a home's value and one response variable corresponding to the sale price of homes sold in Ames, Iowa from 2006 to 2010. Examples of home characteristics found in the dataset describe the square footage of the home, number of rooms, quality, age, and condition of certain amenities, as well as the type of material used to construct certain features of a home.

Two variables, TotalSF (denoting a home's total habitable square footage) and OverallQual (denoting a home's overall level of quality) are highly correlated with the response variable of log-SalePrice. Together, the two features will be used to construct a simple OLS model to be compared with a multiple linear regression (least squares) that will have its features selected by lasso. Since much of the variation in log-SalePrice can likely be explained by the two aforementioned features alone, this parsimonious model serves as a simple yet non-trivial benchmark for a regularized regression to beat.

The lasso-selected multiple regression possessed greater explanatory power in-sample with an adjusted R^2 of 0.86 compared to an adjusted R^2 of 0.81 for the baseline model. However, it was the baseline model that yielded lower error when tested on data out-of-sample with an RMSE of 0.188 versus 0.229 for the lasso-selected model, a roughly 22% difference. The lasso selected model also maintains a lower level of residual standard error at 0.152 vs 0.176 for the baseline model

Literature Review

Penalization techniques originally featured in the literature in pursuit of improving upon OLS' unsatisfactory predictive ability and interpretability, especially in cases where the number of predictors is high. Ridge regression put forth by Hoerl and Kennard (1970) tackles the first issue by manipulating the bias-variance tradeoff in favor of more bias in order to decrease the variance in predictions made on future data. This is done by penalizing the model based on the sum of all squared β weights, also known as L2 regularization. The effect is that ridge regressions will shrink less important coefficients towards 0. This does not, however, address the second issue of interpretability as the model will still retain all predictors. For this reason, ridge regression is often a better option when intuition or theory suggests that the proportion of useful predictors is high. Another technique, subset selection, utilizes a discrete process, either dropping or retaining regressors altogether, which yields higher interpretability but often less stable results as the model becomes more sensitive to changes in the learning set. While the discrete nature of subset selection produces inherently variable predictions, Brieman (1996) addresses this issue by proposing the “bagging” of predictors; a process whereby multiple versions of a predictor are generated to attain an aggregate predictor. Brieman shows bagging to improve accuracy in models where predictor construction is highly sensitive to changes in the learning data.

A technique proposed by Tibshirani (1996) called the *lasso*, for “least absolute shrinkage and selection operator” seeks to inherit the best of both ridge regression and subset selection. It accomplishes this by both shrinking some coefficients towards 0 and setting others directly at 0. The model is instead penalized by the sum of the absolute value of all β weights, performing L1 regularization. The resulting models often provide interpretable findings while also exhibiting the stability of a ridge regression. Since then, many important variations that build upon the lasso have been proposed. The *elastic net* proposed by Zou and Hastie (2005) improves upon lasso in two key situations. First, in the $p > n$ case lasso can select up to n variables before saturating thus making it a weak candidate for modelling this situation. In contrast, the elastic net does not suffer from this fault. The elastic net also tends to perform better when there exist pairs of highly correlated predictors in the learning set as lasso indiscriminately only includes one of the correlated variables while elastic net either includes both or none. Zou and Hastie in their 2005 paper describe the elastic net as akin to “a stretchable fishing net that retains ‘all the big fish’.” This makes elastic net another candidate model for the case of predicting house prices, making full use of the many highly correlated explanatory variables present in the Ames, Iowa housing dataset. The *adaptive lasso* introduced by Zou (2006) uses initial estimates of OLS to weight the variable penalization adaptively, often adopting what is known as the “oracle property” in the process. That is to say, “it performs as well as if the true underlying model were given in advance" (Zou 2006). This makes the adaptive lasso another potential candidate for modeling and predicting house prices from the Ames, Iowa housing dataset.

Within the literature, predicting house prices using a hedonic regression of some kind has featured commonly throughout the years. Taking a quantile regression approach, Zietz et al. (2008) show that:

> "purchasers of higher-priced homes value certain housing characteristics such as square footage and the number of bathrooms differently from buyers of lower-priced homes. Other variables such as age are also shown to vary across the distribution of house prices."

This reflects the notion that some of the observed variation in the estimated prices of housing characteristics may not be priced the same across a given distribution of house prices. Dubin (1998) uses data containing multiple listings of homes to incorporate correlations of price existing between neighboring homes into a regression. Hallic et al. (2015) introduce the *network lasso* based on the Alternating Direction Method of Multipliers (ADMM) algorithm, that they develop and show, allows for guaranteed global convergence even on large graphs. Compared with typical approaches, they show the network lasso is "both a fast and accurate method of solving large optimization problems" such as estimating a hedonic regression to predict house prices with. Manasa et al. (2020) apply techniques such as ridge regression, support vector regression, and boosting algorithms to predict housing prices in India's third most populous city, Bangalore. Singh et al. (2020) make use of the Ames, Iowa housing dataset to construct models to estimate final sale price employing techniques such as random forest and gradient boosting. I will differ from them in my approach by instead using penalized forms of regression such as lasso to predict the sale price of a given home.

Data

This analysis owes itself to the Ames, Iowa housing dataset, for which all the data required come from. The data were first collected by the Ames City Assessor’s Office and later acquired, cleaned, and published by De Cock (2011) in order to provide students and instructors an alternative to the Boston housing dataset. The dataset is rich, containing about 2920 observations and 79 explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 19 continuous) related to assessing a home’s value and one response variable corresponding to the sale price of homes sold in Ames, Iowa from 2006 to 2010. The data are split evenly into a test and training set so that out-of-sample sale price predictions can be made. This explains why in the data tables below the sample size for SalePrice is 1460 for instance.

Examples of home characteristics found in the dataset relate to the square footage of the home and its amenities, number of rooms, quality and condition of certain amenities, the age of certain amenities, and in some cases the type of material used to construct a certain feature of a home. Table 1 below contains a full list of all the variables in the dataset and their brief descriptions. Economically speaking, this paper seeks to construct a hedonic regression that estimates the demand preferences home buyers hold with regards to a home’s characteristics. The Ames, Iowa dataset, containing its many descriptors of home characteristics, provides a unique opportunity to “throw everything but the kitchen sink” into a penalized regression to see what sticks and crucially, ascertain the marginal contributory value of those remaining characteristics to that of a home’s sale price. The goal for lasso’s L1 regularization is to produce a sparse and interpretable model so that insight can be gleaned regarding what features are most important to homebuyers.

One of the first things that should be addressed about the data is the distribution of the SalePrice variable. The data for this variable are skewed heavily to the right since relatively few people can afford homes on the expensive side. The skew of the variable is 1.879 and plotting a histogram of the sale price corroborates this visually. Thus, it is sensible to take the natural logarithm of this variable to normalize it and use log-prices when estimating any regressions. Figures 3.1.1 and 3.1.2 display the distribution of sale price both before and after taking the natural log of the variables. Likewise, figures 3.2.1 and 3.2.2 display a Q-Q plot of sale price both before and after taking the natural log. From looking at these figures it is clear that taking the natural log of the variable serves to normalize the distribution. Indeed, the skew of the log-SalePrice distribution has decreased to 0.1211. In addition, taking the natural logarithm of the response variable endows a log-linear relationship into the model making it possible to interpret coefficients found in terms of percent change for a unit change in any independent variables.

The data still requires a bit of cleaning before modelling. In total, 34 variables are missing at least one value. The approach taken to clean the data borrows in part from Bruin (2018). In the case of many variables that are categorical, the missing observations are the result of there being no category associated with variable non presence. Therefore, imputing a 'None' or 'Not Present' type of value will remedy the issue in these cases. For integer variables containing missing observations, the value 0 will most often be imputed. In descending order, starting with the variables containing the most missing observations, they are as follows:

1. PoolQC - 2909 NAs
2. MiscFeature - 2814 NAs
3. Alley - 2721 NAs

4. Fence - 2348 NAs
5. FireplaceQu - 1420 NAs
6. LotFrontage - 486 NAs
7. GarageYrBlt - 159 NAs
8. GarageFinish - 159 NAs
9. GarageQual - 159 NAs
10. GarageCond - 159 NAs
11. GarageType - 157 NAs
12. BsmtCond - 82 NAs
13. BsmtExposure - 82 NAs
14. BsmtQual - 81 NAs
15. BsmtFinType2 - 80 NAs
16. BsmtFinType1 - 79 NAs
17. MasVnrType - 24 NAs
18. MasVnrArea - 23 NAs
19. MSZoning- 4 NAs
20. Utilities - 2 NAs
21. BsmtFullBath - 2 NAs
22. BsmtHalfBath - 2 NAs
23. Functional - 2 NAs
24. Exterior1st - 1 NA
25. Exterior2nd - 1 NA
26. BsmtFinSF1 - 1 NA
27. BsmtFinSF2 - 1 NA
28. BsmtUnfSF - 1 NA
29. TotalBsmtSF - 1 NA
30. Electrical - 1 NA
31. KitchenQual - 1 NA
32. GarageCars - 1 NA
33. GarageArea - 1 NA
34. SaleType - 1 NA

The high number of missing observations among the PoolQC variable has to do with there being few homes that have a pool in the first place. Pool quality follows a simple ordinal ranking scale:

Ex Excellent

Gd Good

TA Average/Typical

Fa Fair

Po Poor

NA No Pool

Variables such as this which ordinally rank the quality of a given feature, of which there are many, can be numerically reencoded using a simple vector that takes the following form:

```
('None' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
```

This way, homes without a pool are no longer missing any observations and rather receive a score of 0 for PoolQC. This vector will be reused several times to numerically recode variables which exhibit ordinality (and that can be represented on a scale of 0-5).

The high number of missing observations among the MiscFeature variable similar to PoolQC is a result of there being no quality associated with “none”. The categories for MiscFeature are:

Elev	Elevator
Gar2	2nd Garage (if not described in garage section)
Othr	Other
Shed	Shed (over 100 SF)
TenC	Tennis Court
NA	None

Once again, missing observations will be reencoded as ‘None’, however the variable will be factorized since there exists no ordinality.

The 2721 missing observations associated with the Alley variable are remedied by encoding a “None” for those observations. The variable is also factorized for usage later. The same goes for the Fence variable. The FireplaceQu variable associated with fireplace quality can be reencoded numerically using the aforementioned quality ranking vector.

The LotFrontage variable provides a continuous measure of the linear feet of street connected to the property. For the 486 missing observations, the median lot frontage of a given home's neighborhood is imputed.

For the garage related variables, 158 of the 159 missing observations are due to those homes lacking any kind of garage. For the one home that did have a garage but contained missing values for garage condition, quality, and finish, the mode of those variables was imputed to replace the missing observations. Likewise, aside from GarageType these variables can be reencoded numerically since they are ordinal in nature.

Of the missing observations for basement related variables, 79 are the result of those homes lacking a basement to begin with. The remaining homes that possess a basement but are missing observations for one of the basement related categories are imputed using the modes of those variables. BsmtQual, BsmtCond, BsmtFinType1, BsmtFinType2, and BsmtExposure can all be reencoded numerically since they are ordinal.

For the two masonry related variables there exists one home that has a value entered for MasVnrArea but no value for MasVnrType. The mode is imputed here, assigning the home 'BrkFace' as its masonry veneer type. For the remaining 23 missing observations corresponding to homes without any masonry, a 'None' value is imputed. The variable is reencoded numerically since it arguably follows an ordinal structure where certain materials are more expensive and or desirable to use in a home. The vector used to recode the variable takes the following form:

```
('None' = 0, 'BrkCmn' = 0, 'BrkFace' = 1, 'Stone' = 2).
```

The MSZoning variable's 4 missing observations can be imputed using the mode of the variable after which the variable is factorized. The KitchenQual variable's lone missing observation is imputed using the mode and is then reencoded numerically using the aforementioned quality ranking vector. The Functional variable relates to home functionality and is ultimately ordinal in nature. The mode is imputed for the missing observation and the variable is reencoded numerically on a scale of 0-7 corresponding to the originally ordinal categories listed below.

Typ	Typical Functionality
Min1	Minor Deductions 1
Min2	Minor Deductions 2
Mod	Moderate Deductions
Maj1	Major Deductions 1
Maj2	Major Deductions 2
Sev	Severely Damaged
Sal	Salvage only

The Exterior1st and Exterior2nd variables both have lone missing observations that are imputed using the modes of these categorical variables. The Electrical variable relating to a home's electrical system has only one missing observation that is imputed using the mode of this categorical variable. The SaleType variable is also categorical, and its lone missing observation is imputed using the mode once again. Finally, the Utilities variable which ordinally ranks a home's access to public utilities will be removed from the investigation altogether as there is only one home in the entire dataset that does not have access to all public utilities making the variable virtually useless for the purposes of making predictions on unseen data. With that, the dataset is now free of any missing values.

The variable most correlated with SalePrice is OverallQual at 0.791. Figure 3.3 provides a visual representation of the relationship while also highlighting the heteroskedastic nature of the untransformed price variable, further reinforcing the need to take the natural logarithm of the variable for model estimation purposes.

Prior to modelling, using the already existing data I will create a few new variables that may provide additional predictive insight when modelling. First, an “Age” variable can be created by simply subtracting YearRemodAdd (the year that any remodeling occurred) from YrSold. Note that the value for YearRemodAdd defaults to the value of YearBuilt in the case that no remodeling has been done. With that, a dummy variable that conveys whether a home has been remodeled can be derived by checking if the year built matches the year any remodeling was done. One would expect older homes to be valued lower on average and indeed this proves to be the case with Age and SalePrice sharing a negative correlation of roughly -0.509. This serves as a form of penalty for the model to weigh against homes that are newer due to remodeling versus homes that are truly new (i.e. built from scratch). Furthermore, an “IsNew” dummy variable will be created by checking if the year sold of a home matches the year built. Altogether, 116 of the homes sold were new and the remaining 2803 were not. As shown in figure 3.4, the average price of new homes sold is substantially higher than those sold that were not new.

A variable, which I will name TotalSF, that combines above ground living area with basement square footage can also be created to represent the properties total living space in square footage. As one might intuit, the correlation between SalePrice and TotalSF is high at 0.779. Figure 3.5 depicts this clearly and also provides a valuable look at what may potentially be two outliers in the training data. The two far-off data points pictured correspond to observations 524 and 1299 in the training set. Both homes maintain the highest attainable score of 10 for OverallQual and are also among the largest properties in terms of total square footage, and yet remain under $200,000 in sale price. For this reason, these two observations will be removed from the analysis. Upon removing the two outlier observations, the correlation between SalePrice and TotalSF increased to 0.829, a rather stark improvement of nearly 6.42%

Observation #	**SalePrice**	**TotalSF**	**OverallQual**
524	$184,750	7814	10
1299	$160,000	11752	10

Similarly, the 6 variables which relate to a porch square footage will also be concatenated into a single variable named TotalPorchSF. The correlation between SalePrice and TotalPorchSF is weak but positive at 0.196. Lastly, the 4 variables relating to the number of bathrooms will be concatenated into one by adding them up to create a variable named TotalBathrooms. The two “half bathroom” variables will be divided by 2 when summing them with the “full bathroom” variables. The correlation between this consolidated bathroom variable and SalePrice is moderately strong and positive at 0.599.

Finally, the numeric YrSold variable will be converted into a factor as the data span 2006-2010 which contains a housing bubble and economic crisis. This is to control for the fact that homes sold in 2006/07 at the peak of the housing bubble likely went for more than an equivalent home would in 2009/10. Likewise, the numeric MoSold (month sold) variable will be factorized as well to account for any seasonality that takes place within a given year. Now that the data have been sufficiently cleaned, Table 2 below provides descriptive statistics for each of the variables contained in the dataset.

Methodology

The variables TotalSF and OverallQual are the two variables most correlated with log-SalePrice with correlations of roughly 0.82. Intuitively this makes sense as properties with greater square footage will, ceteris paribus, generally sell for more. The same can be said for properties of higher quality. Together the two features will be used to construct a simple baseline OLS model that takes the following form.

$$\widehat{y_i} = \alpha + \beta_1 \times TotSF_{i\,1} + \beta_2 \times Qual_{i\,2} + \varepsilon_i$$

where $\hat{y}$is log-SalePrice, α is the intercept, ε is the error term, and β_1 and β_2 are the slopes corresponding to the two explanatory variables. Since much of the variation in log-SalePrice can likely be explained by these two features alone, this parsimonious model serves as a simple yet non-trivial benchmark for a biased regression to beat.

Recall that lasso regression performs L1 regularization thanks to the addition of a penalty term that is equal to the absolute value of the magnitude of the coefficients. The resulting shrinkage of certain coefficients to 0 should produce a sparse model that is easy to interpret, thus taking care of feature selection for us in the process. The general solution path for lasso is found by minimizing the following loss function

$$L_{lasso}(\hat{\beta}) = \sum_{i=1}^{n} (y_i - \sum_{j} x_{ij}\hat{\beta})^2 + \lambda \sum_{j=1}^{p} |\widehat{\beta_j}|$$

where λ determines the stiffness of the L1 penalty. As $\lambda \to 0$ the model's estimates will approach that of OLS. As $\lambda \to \infty$ the model's estimate will all approach 0 as coefficients continually shrink. This minimizes the sum of squares as OLS does but subject to a constraint of

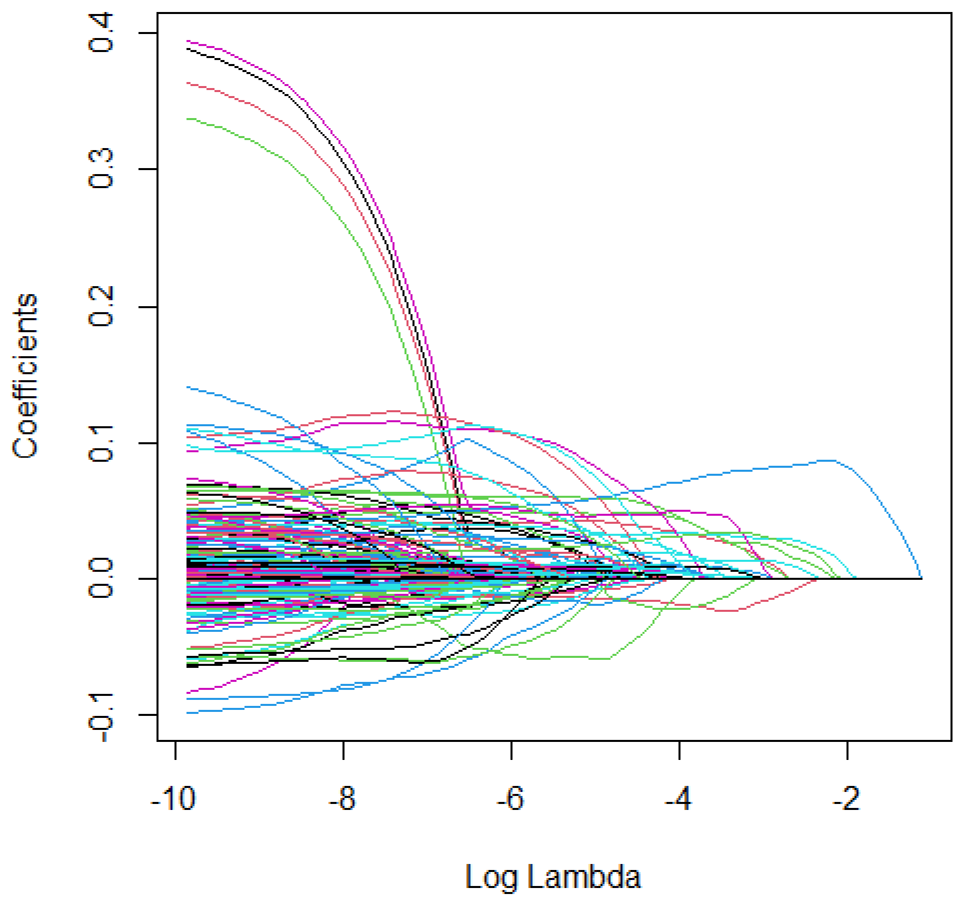
$$\sum_{j} |\beta_j| \leq t$$

Here $t \geq 0$is the tuning parameter (Tibshirani 1996). Note that when λ is exactly 0 the observed estimate will be equal to the estimate found by OLS as no shrinkage will take place. As λ increases, the model's bias will increase, and its variance decrease as more coefficients are eliminated from the final output. This raises the important question of what λ should be set to. One method is to estimate the model over many possible λ values and choose the λthat minimizes a given Information Criterion such as Akaike's or Bayes. Another is to perform cross-validation along a grid of possible λ values and to choose the value that minimizes MSE. With the *glmnet* package in R this can be done straightforwardly and will be the method employed for this analysis. The key point to emphasize is that variable selection will be taken care of automatically when estimating our regression with a penalty term. Variables that explain little to none of the variation in the response variable should be eliminated from final output. Using the variables selected by lasso, a new OLS model can be constructed to be compared with the baseline OLS model.

Results

Of the 95 variables presented to the model initially, the lasso model selects 8, shrinking the remaining 87 down to zero, thus eliminating them from the final model output. Lasso does this by k-fold cross-validating over a grid of possible penalty terms, choosing the term that minimizes error. It selected a value of 0.01 for the lambda penalty term. Below is the variable trace plot for the Lasso model displaying the variable shrinkage that occurred as a result of lasso's L1 regularization.

Figure 4.1 Lasso Variable Trace Plot depicting the level of shrinkage



The variables kept by lasso include the two variables used to estimate the baseline OLS model, TotalSF and OverallQual, as well as KitchenQual, Age, GarageCars, GarageFinish, GrLivArea, and IsNew. Using these variables selected by lasso, a new OLS model is constructed to be compared with the baseline OLS model. Below is a correlation matrix for the Lasso-Selected OLS effects.

Correlation Matrix of Lasso-Selected OLS Effects

	log-SalePrice	TotalSF	Overall Qual	Kitchen Qual	Age	GarageCars	GarageFinish	GrLivArea
log-SalePrice	1							
TotalSF	0.821	1	0.667	0.509	-0.368	0.559	0.423	0.867
OverallQual	0.821	0.667	1	0.675	-0.572	0.601	0.551	0.573
KitchenQual	0.67	0.509	0.675	1	-0.614	0.488	0.454	0.428
Age	-0.569	-0.368	-0.572	-0.614	1	-0.426	-0.448	-0.319
GarageCars	0.681	0.559	0.601	0.488	-0.426	1	0.577	0.494
GarageFinish	0.606	0.423	0.551	0.454	-0.448	0.577	1	0.357
GrLivArea	0.725	0.867	0.573	0.428	-0.319	0.494	0.357	1

Baseline OLS model (1)
vs Lasso Selected OLS model (2)

	Dependent variable:	
	Log-SalePrice	
	(1)	(2)
TotalSF	0.0002522***	0.000199***
	(0.00001)	(0.00001)
OverallQual	0.14277***	0.07850***
	(0.005)	(0.005)
KitchenQual		0.048592***
		(0.009)
Age		-0.0022***
		(0.0003)
GarageCars		0.06469***
		(0.007)
GarageFinish		0.04637***
		(0.006)
GrLivArea		0.0001***
		(0.00002)
IsNew		0.04264**
		(0.021)
Constant	10.508***	10.641***
	(0.021)	(0.033)
Observations	1,458	1,458
R^2	0.806	0.856
Adjusted R^2	0.805	0.855
Residual Std. Error	0.176 (df = 1455)	0.152 (df = 1449)
F Statistic	3,015*** (df = 2; 1455)	1,074*** (df = 8; 1449)
RMSE on Test Set	0.22867	0.18780

Note: *p<0.1; **p<0.05; ***p<0.01

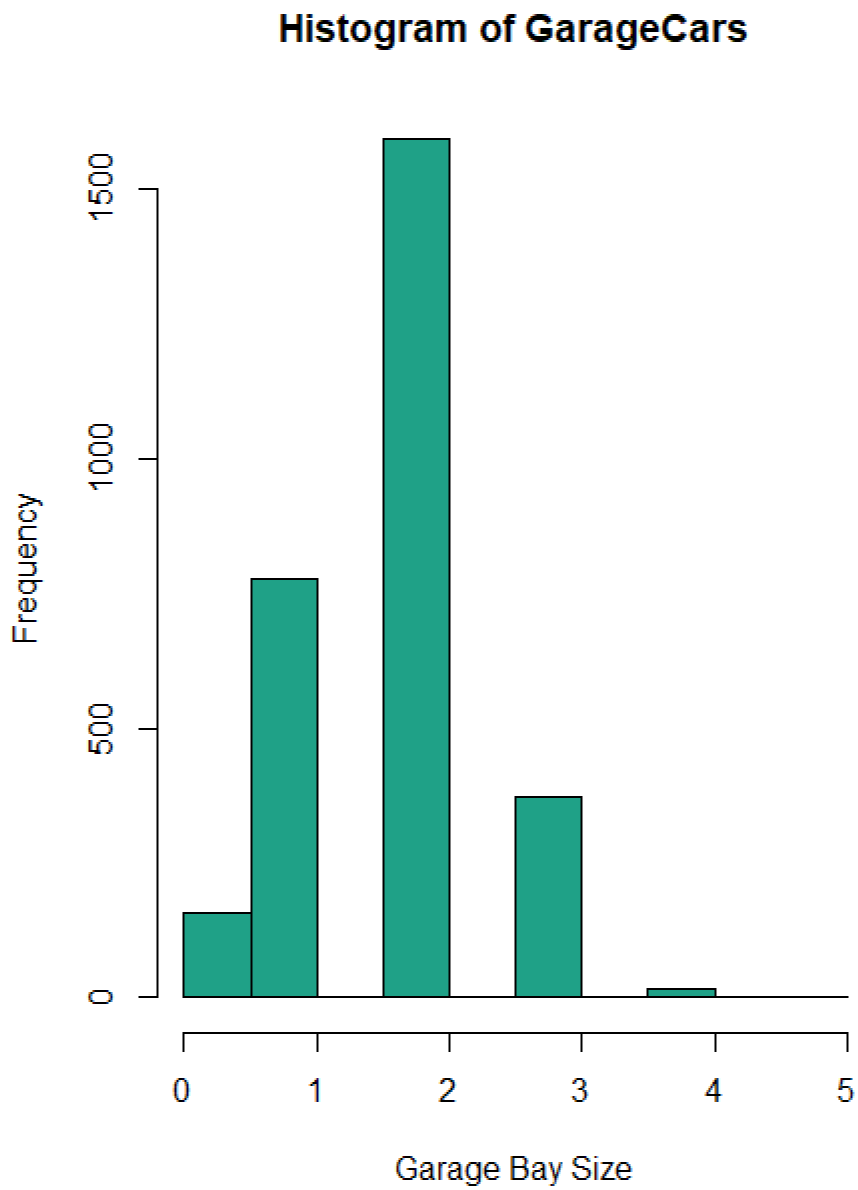
Looking at the overall significance of the models as proxied by F-tests, both models score highly in this regard. Both models have large F-statistics that are significant at the 1% level. This indicates that the coefficients jointly possess some predictive power for both models and that we can confidently reject the null hypothesis of $R^2 = 0$. The F-stat for the baseline model is higher than its lasso counterpart at 3,015 vs 1,074, however this may be due to so-called omitted variable bias lurking within the model. The adjusted R-squared, or coefficient of determination, for the baseline model is .805 vs .855 for the lasso selected model. The lasso selected model can therefore explain 5% more of the variation in the data. The lasso selected model also maintains a lower level of residual standard error at 0.152 vs 0.176 for the baseline model, demonstrating a lower level of uncertainty is present it the lasso-selected model's estimates. Figures 4.4 and 4.5 contain a scatterplot of the two model's residuals.

Both parameter estimates attained by the baseline OLS model are statistically significant at the 1% level. For the lasso selected model, all parameter estimates are significant at the 1% level except for the coefficient corresponding to the Age variable only being significant at the 5% level. For TotalSF, the parameter estimate in the baseline model is 0.0002522. Seeing as this model takes a log-linear form where the dependent variable is transformed using a natural logarithm, the economic interpretation is such that for every 1000 feet increase in total habitable square footage, we would expect a 25.22% increase in sale price. As one might expect, the lasso selected OLS model, which controls for more effects, estimates the effect of TotalSF to be less pronounced at 0.000199. Thus, the marginal sensitivity of TotalSF under this model is such that for every 1000 unit increase in habitable square-footage, we would expect a 19.9% increase in sale price ceteris paribus. The explanatory variable with the largest raw effect under both models belongs to OverallQual, denoting the general level of quality for a given home. The parameter estimate under the baseline model is 0.14277 indicating that for every one-unit increase in overall quality we would expect a concomitant increase of 14.28% in sale price holding all else equal. The lasso-selected model estimates the quality effect to be about half as strong at 0.0785. Economically speaking, we would therefore expect a less pronounced 7.85% increase in sale price for every one unit increase in OverallQual under this model.

The remaining 6 parameters of interest are only included in the lasso selected OLS model. The effect associated with the KitchenQual variable is moderately strong, corresponding to a 4.86% increase in sale price for every one unit increase in kitchen quality, which follows an ordinal scale of 1-5. The parameter estimate associated with the Age variable is the only negative term under either model, which intuitively makes sense as older homes will sell for less generally. Under this model, for every 1-year increase in a home's age we would expect a 0.22% reduction in a home’s sale price. Marginally speaking, this seems insignificant, but over a period of say 30 years, the standard length of a home mortgage, we would expect a 6.6% reduction in a home’s value to result from the increase in the home’s age, holding all else, such as quality, equal.

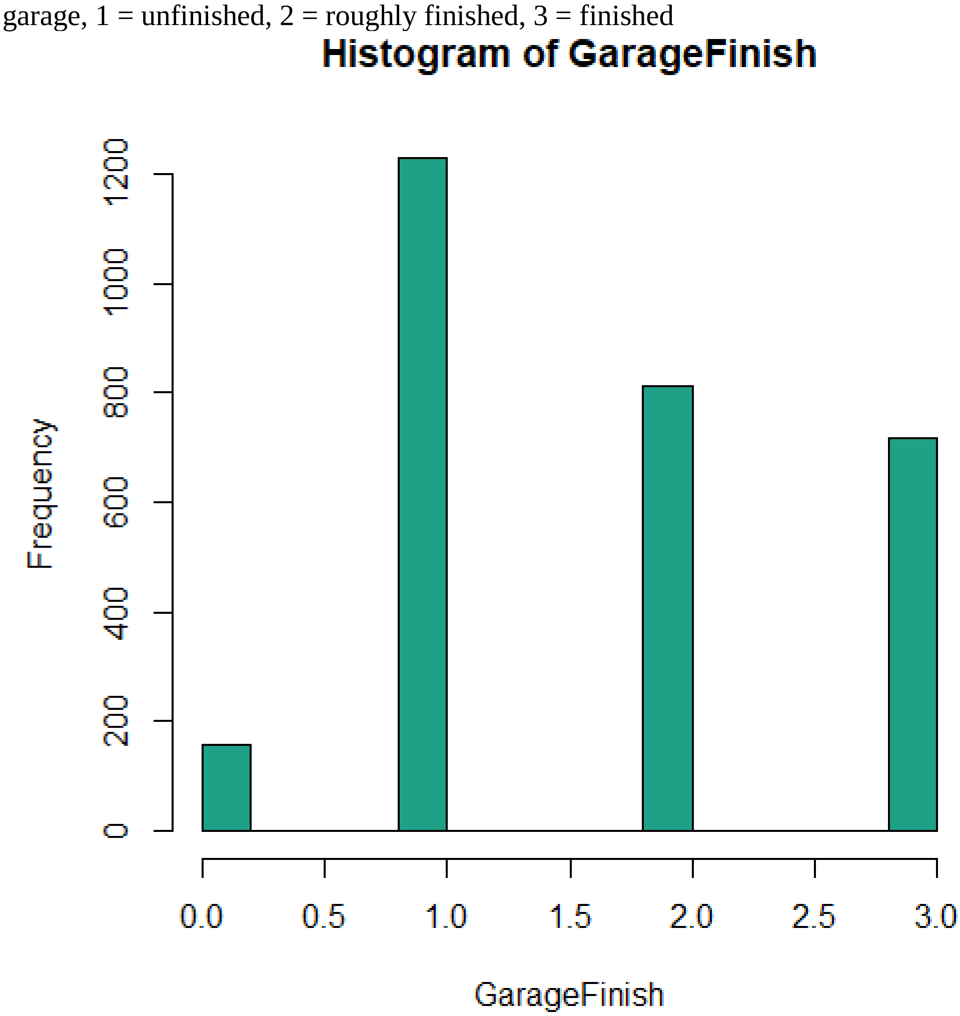
The next two parameters of interest, GarageCars and GarageFinish provide valuable insight regarding the specific preferences of homebuyers in Ames, Iowa. Being the only city in Iowa to boast a population greater than 50,000, Ames has a certain degree of suburban sprawl commonly seen in the midwestern United States. This necessitates, as in many parts of the U.S, the need to own a car to commute to and from work. As such, homebuyers place a premium on the presence of a car garage built into a home as well as a premium on the size and finish of a home’s garage section. Recall that the GarageCars variable elucidates the number of cars a home's garage bay can store. A score of 0 indicates there is no garage bay whereas 5 is the maximum number of cars any home found in the dataset could store in its garage bay. Below is a histogram of GarageCars showing that a 2-car garage bay is what is most typical in our sample.

Figure 4.2



The parameter estimate associated with GarageCars is 0.06469 indicating that for each additional car a home's garage bay can store, we would expect a home's sale price to increase by 6.47%, a noticeable increase. The variable GarageFinish also follows an ordinal scale with 0 equating to no garage, 1 with an unfinished garage, 2 with a "roughly finished" garage and 3 with a finished garage. The parameter estimate corresponding to GarageFinish is 0.04637 indicating that for every one-unit increase in GarageFinish we would expect a home to sell for 4.64% more in dollar terms.

Figure 4.3
0 = no garage, 1 = unfinished, 2 = roughly finished, 3 = finished



The next parameter to interpret is associated with the GrLivArea variable which denotes a home's above ground habitable square footage. The parameter estimate is 0.0001 indicating that for every 1000 square foot increase we would expect only a 0.1% increase in a home's sale price. This seems oddly low and is likely due to the effect already being accounted for in the model by the TotalSF variable causing it to be dominated in the model. Lastly of the interpretable coefficients, the IsNew variable is a simple dummy variable indicating whether a home sold was brand new or not. With a parameter estimate of 0.04264 we would expect a new home to sell for 4.26% more than a non-new home holding all else equal. Of course, holding "all else equal" is difficult since newly built homes cannot physically be the same age as a non-new home, for example. This raises the important question of how to interpret the intercept term. Economically speaking it is difficult if not impossible to interpret it since doing so requires holding all the coefficients at 0. However, a value of 0 for each characteristic of a home such as its square footage makes little to no sense in the real world.

Furthermore, estimating both models using robust HC2 standard errors had little effect on the two model's parameter estimates and ultimately the economic conclusions are the same.

Baseline OLS model (1)
vs Lasso Selected OLS model (2)
Using Robust HC2 Standard Errors

	Dependent variable:	
	Log-SalePrice	
	(1)	(2)
TotalSF	0.0003***	0.0002***
	(0.00001)	(0.00001)
OverallQual	0.143***	0.079***
	(0.005)	(0.005)
KitchenQual		0.049***

		(0.009)
Age		-0.002***
		(0.0003)
GarageCars		0.065***
		(0.008)
GarageFinish		0.046***
		(0.006)
GrLivArea		0.0001***
		(0.00002)
IsNew		0.043**
		(0.018)
Constant	10.508***	10.641***
	(0.023)	(0.032)
Observations	1,458	1,458
R^2	0.806	0.856
Adjusted R^2	0.805	0.855
Residual Std. Error	0.176 (df = 1455)	0.152 (df = 1449)
F Statistic	3,015*** (df = 2; 1455)	1,074*** (df = 8; 1449)
Note:		*p$^{<0.1;\ >**}$p$^{<0.05;\ >***}$p<0.01

Interestingly, when testing the models on the test set the baseline OLS model performs better than the lasso-selected OLS mode with 0.18780 RMSE versus 0.22867 for the lasso-selected model, a 21.8% improvement. Note that prior to calculating loss on the test set, predictions of log-SalePrice made by the model in log-terms are converted back into untransformed dollar-terms. It seems that despite possibly possessing more in-sample explanatory power, the more parsimonious baseline model appears to have greater predictive power when tested on data out-of-sample.

Figure 4.4: Base OLS model residual plot

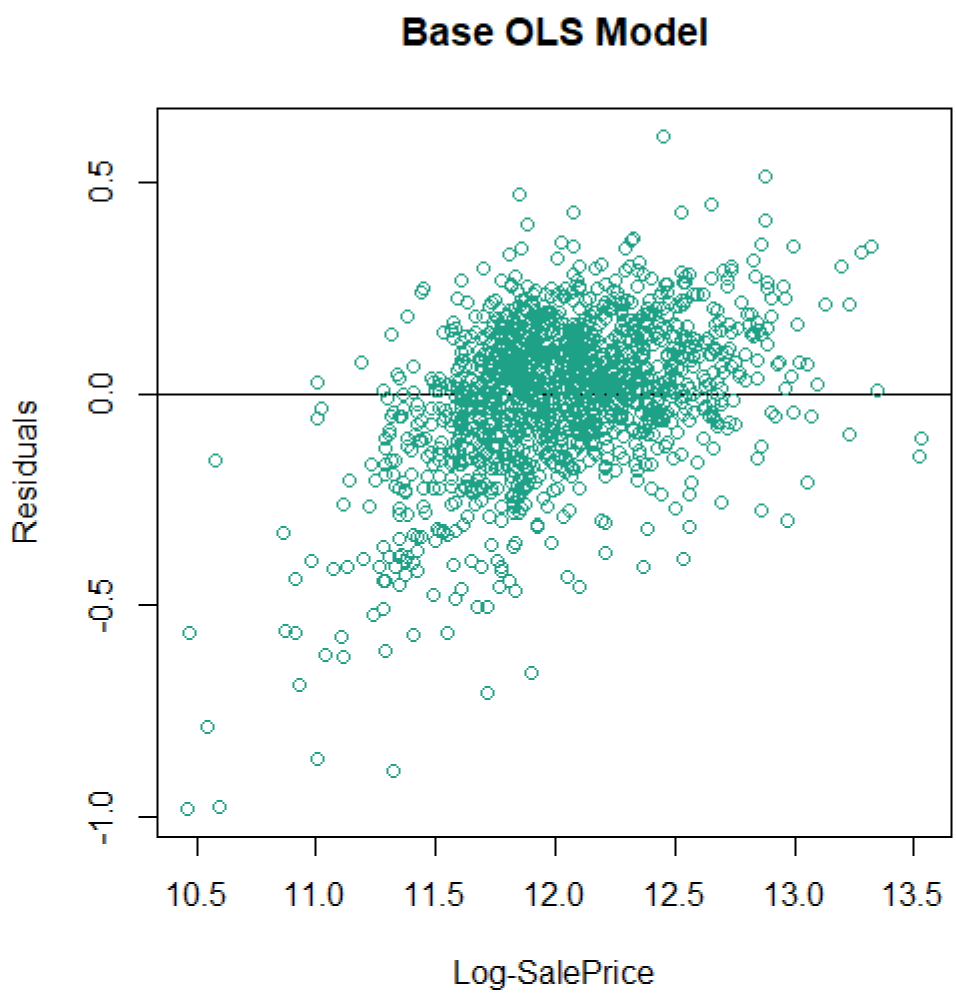
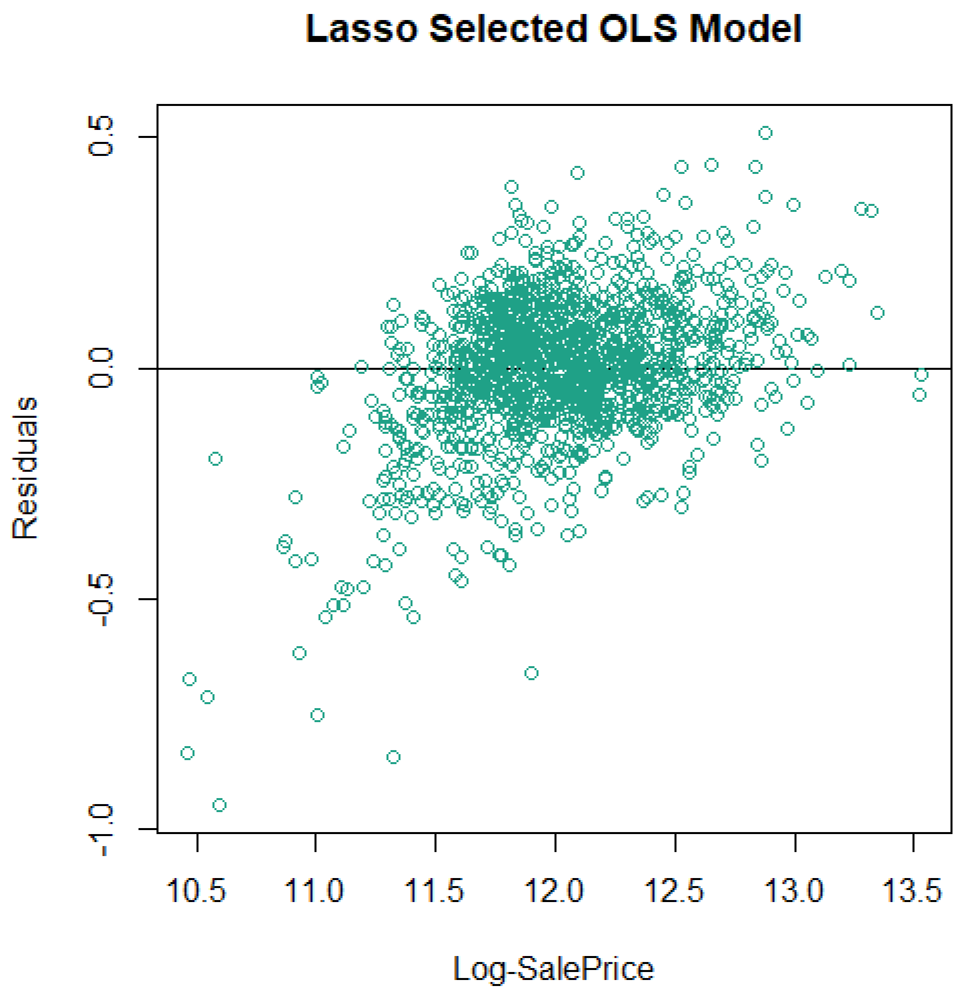


Figure 4.5: Lasso Selected OLS model residual plot



Conclusion

Understanding the demand preferences of home buyers is key for suppliers to be able to efficiently meet the demand. The Ames, Iowa housing dataset is rich and diverse in its descriptions of a home's characteristics making it a great candidate for L1 regularization in order to gleam the most important features for determining sale price. Doing so not only reveals which characteristics are important to home buyers but also allows us to ascertain the marginal degree to which home buyers value those characteristics of a home.

In the case of the home buyers of Ames, Iowa, the lasso-selected multiple regression revealed some of the specific preferences held such as the premium placed on the inclusion of a garage into a home as well as the preference for a certain level of finish regarding a home's garage. This fits in line with the observed results of the two models. Indeed, the model containing features selected by lasso had greater explanatory power in-sample with an adjusted R^2 of 0.86 compared to an adjusted R^2 of 0.81 for the baseline model. The lasso selected model also maintains a lower level of residual standard error at 0.152 vs 0.176 for the baseline model demonstrating a higher level of certainty in its estimates. However, it was the baseline model that yielded lower error when tested on data out-of-sample with an RMSE of 0.188 versus 0.229 for the lasso-selected model, a roughly 22% difference.

Of course, this analysis is by no means exhaustive. Further research could investigate and compare results between different penalization techniques such as L2 ridge regression and elastic-net (L1+L2) regression. Moreover, while this analysis made use of the features selected by lasso to construct a least squares multiple linear regression, the more complete approach perhaps would have been to also include and compare results of the regression estimated by lasso using its parameter estimates as opposed solely estimating with OLS using lasso's chosen features.

References

Hoerl, A.E. and Kennard, R.W., 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), pp.55-67.

Breiman, L., 1996. Bagging predictors. *Machine learning*, *24*(2), pp.123-140.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), pp.267-288.

Zou, H. and Hastie, T., 2005. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, *67*(2), pp.301-320.

Zou, H., 2006. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, *101*(476), pp.1418-1429.

Zietz, J., Zietz, E.N. and Sirmans, G.S., 2008. Determinants of house prices: a quantile regression approach. *The Journal of Real Estate Finance and Economics*, *37*(4), pp.317-333.

Dubin, R.A., 1998. Predicting house prices using multiple listings data. *The Journal of Real Estate Finance and Economics*, *17*(1), pp.35-59.

Hallac, D., Leskovec, J. and Boyd, S., 2015, August. Network lasso: Clustering and optimization in large graphs. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 387-396).

Manasa, J., Gupta, R. and Narahari, N.S., 2020, March. Machine learning based predicting house prices using regression techniques. In *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)* (pp. 624-630). IEEE.

Singh, A., Sharma, A. and Dubey, G., 2020. Big data analytics predicting real estate prices. *International Journal of System Assurance Engineering and Management*, pp.1-12.

De Cock, D., 2011. Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project. Journal of Statistics Education, 19(3).

Bruin, E., 2018. *House prices: Lasso, XGBoost, and a detailed EDA*. Kaggle.com. Available at: <https://www.kaggle.com/erikbruin/house-prices-lasso-xgboost-and-a-detailed-eda/report?select=test.csv>.

Fonti, V. and Belitser, E., 2017. Feature selection using lasso. VU Amsterdam Research Paper in Business Analytics, 30, pp.1-25.

Tables and Figures

Table 1: Variable Descriptions

Variable	Description
SalePrice	The property's sale price in dollars
MSSubClass	The building class
MSZoning	The general zoning classification
LotFrontage	Linear feet of street connected to property
LotArea	Lot size in square feet
Street	Type of road access
Alley	Type of alley access
LotShape	General shape of property
LandContour	Flatness of the property
Utilities	Type of utilities available
LotConfig	Lot configuration
LandSlope	Slope of property
Neighborhood	Physical locations within Ames city limits
Condition1	Proximity to main road or railroad
Condition2	Proximity to main road or railroad (if a second is present)
BldgType	Type of dwelling
HouseStyle	Style of dwelling
OverallQual	Overall material and finish quality
OverallCond	Overall condition rating
YearBuilt	Original construction date
YearRemodAdd	Remodel date
RoofStyle	Type of roof
RoofMatl	Roof material

Exterior1st	Exterior covering on house
Exterior2nd	Exterior covering on house (if more than one material)
MasVnrType	Masonry veneer type
MasVnrArea	Masonry veneer area in square feet
ExterQual	Exterior material quality
ExterCond	Present condition of the material on the exterior
Foundation	Type of foundation
BsmtQual	Height of the basement
BsmtCond	General condition of the basement
BsmtExposure	Walkout or garden level basement walls
BsmtFinType1	Quality of basement finished area
BsmtFinSF1	Type 1 finished square feet
BsmtFinType2	Quality of second finished area (if present)
BsmtFinSF2	Type 2 finished square feet
BsmtUnfSF	Unfinished square feet of basement area
TotalBsmtSF	Total square feet of basement area
Heating	Type of heating
HeatingQC	Heating quality and condition
CentralAir	Central air conditioning
Electrical	Electrical system
1stFlrSF	First Floor square feet
2ndFlrSF	Second floor square feet
LowQualFinSF	Low quality finished square feet (all floors)
GrLivArea	Above grade (ground) living area square feet
BsmtFullBath	Basement full bathrooms
BsmtHalfBath	Basement half bathrooms
FullBath	Full bathrooms above grade

HalfBath	Half baths above grade
Bedroom	Number of bedrooms above basement level
Kitchen	Number of kitchens
KitchenQual	Kitchen quality
TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)
Functional	Home functionality rating
Fireplaces	Number of fireplaces
FireplaceQu	Fireplace quality
GarageType	Garage location
GarageYrBlt	Year garage was built
GarageFinish	Interior finish of the garage
GarageCars	Size of garage in car capacity
GarageArea	Size of garage in square feet
GarageQual	Garage quality
GarageCond	Garage condition
PavedDrive	Paved driveway
WoodDeckSF	Wood deck area in square feet
OpenPorchSF	Open porch area in square feet
EnclosedPorch	Enclosed porch area in square feet
3SsnPorch	Three season porch area in square feet
ScreenPorch	Screen porch area in square feet
PoolArea	Pool area in square feet
PoolQC	Pool quality
Fence	Fence quality
MiscFeature	Miscellaneous feature not covered in other categories
MiscVal	$Value of miscellaneous feature
MoSold	Month Sold

YrSold	Year Sold
SaleType	Type of sale
SaleCondition	Condition of sale

Table 2: Summary Statistics

Note that factorized categorical variables are denoted by an adjacent asterisk (*) while numeric ordinal variables are denoted by an adjacent red triangle (▲). The remaining variables are numeric continuous, typically measured in square feet.

Variable	**n**	**mean**	**sd**	**median**	**min**	**max**	**skew**	**kurtosis**
MSSubClass*	2917	5.266	4.346	5.000	1.000	16.000	0.738	-0.477
MSZoning*	2917	4.028	0.659	4.000	1.000	5.000	-1.750	5.913
LotFrontage	2917	69.431	21.205	70.000	21.000	313.000	1.104	8.509
LotArea	2917	10139.439	7807.037	9452	1300	215245	13.103	274.975
Street ▲	2917	0.996	0.064	1.000	0.000	1.000	-15.487	237.922
Alley*	2917	1.986	0.260	2.000	1.000	3.000	-0.651	11.650
LotShape ▲	2917	2.601	0.568	3.000	0.000	3.000	-1.247	1.469
LandContour*	2917	3.779	0.701	4.000	1.000	4.000	-3.130	8.491
LotConfig*	2917	4.057	1.604	5.000	1.000	5.000	-1.197	-0.438
LandSlope ▲	2917	1.946	0.249	2.000	0.000	2.000	-4.971	26.486
Neighborhood*	2917	13.324	5.823	13.000	1.000	25.000	-0.011	-1.028
Condition1*	2917	3.040	0.873	3.000	1.000	9.000	2.988	15.731

Condition2*	2917	3.001	0.206	3.000	1.000	8.000	12.335	325.055
BldgType*	2917	1.506	1.207	1.000	1.000	5.000	2.190	3.181
HouseStyle*	2917	4.025	1.913	3.000	1.000	8.000	0.319	-0.953
OverallQual	2917	6.086	1.407	6.000	1.000	10.000	0.189	0.054
OverallCond	2917	5.565	1.113	5.000	1.000	9.000	0.569	1.469
YearBuilt	2917	1971.288	30.287	1973	1872	2010	-0.599	-0.514
YearRemodAdd	2917	1984.248	20.892	1993	1950	2010	-0.450	-1.348
RoofStyle*	2917	2.395	0.820	2.000	1.000	6.000	1.557	0.885
RoofMatl*	2917	2.063	0.539	2.000	2.000	8.000	8.718	76.801
Exterior1st*	2917	10.625	3.199	13.000	1.000	15.000	-0.732	-0.309
Exterior2nd*	2917	11.337	3.551	14.000	1.000	16.000	-0.681	-0.558
MasVnrType	2917	0.471	0.647	0.000	0.000	2.000	1.045	-0.053
MasVnrArea	2917	100.931	178.032	0.000	0.000	1600	2.620	9.430
ExterQual	2917	3.396	0.579	3.000	2.000	5.000	0.783	0.061
ExterCond	2917	3.086	0.372	3.000	1.000	5.000	1.314	6.262
Foundation*	2917	2.393	0.727	2.000	1.000	6.000	0.009	0.751
BsmtQual	2917	3.479	0.900	4.000	0.000	5.000	-1.258	4.051
BsmtCond	2917	2.921	0.567	3.000	0.000	4.000	-3.625	17.045

BsmtExposure	2917	1.623	1.067	1.000	0.000	4.000	1.122	-0.073
BsmtFinType1	2917	3.540	2.114	4.000	0.000	6.000	-0.148	-1.597
BsmtFinSF1	2917	438.865	444.181	368.000	0.000	4010	0.980	1.420
BsmtFinType2	2917	1.274	0.955	1.000	0.000	6.000	3.152	10.140
BsmtFinSF2	2917	49.599	169.232	0.000	0.000	1526	4.142	18.779
BsmtUnfSF	2917	560.504	439.699	467.000	0.000	2336	0.919	0.398
TotalBsmtSF	2917	1048.968	429.472	988.000	0.000	5095	0.671	3.699
Heating*	2917	2.025	0.246	2.000	1.000	6.000	12.068	167.684
HeatingQC	2917	4.151	0.958	5.000	1.000	5.000	-0.549	-1.150
CentralAir	2917	0.933	0.250	1.000	0.000	1.000	-3.456	9.946
Electrical*	2917	4.685	1.048	5.000	1.000	5.000	-3.078	7.624
X1stFlrSF	2917	1157.692	385.264	1082	334.000	5095	1.257	5.059
X2ndFlrSF	2917	335.862	428.120	0.000	0.000	2065	0.861	-0.427
LowQualFinSF	2917	4.698	46.413	0.000	0.000	1064	12.078	174.387
GrLivArea	2917	1498.252	496.909	1444	334.000	5095	1.068	2.447
BsmtFullBath	2917	0.429	0.524	0.000	0.000	3.000	0.622	-0.747
BsmtHalfBath	2917	0.061	0.246	0.000	0.000	2.000	3.928	14.808
FullBath	2917	1.567	0.552	2.000	0.000	4.000	0.165	-0.545

HalfBath	2917	0.380	0.503	0.000	0.000	2.000	0.696	-1.032
BedroomAbvGr	2917	2.860	0.823	3.000	0.000	8.000	0.326	1.930
KitchenAbvGr	2917	1.045	0.215	1.000	0.000	3.000	4.298	19.710
KitchenQual ▲	2917	3.510	0.661	3.000	2.000	5.000	0.437	-0.252
TotRmsAbvGrd	2917	6.448	1.564	6.000	2.000	15.000	0.749	1.147
Functional ▲	2917	6.848	0.640	7.000	1.000	7.000	-4.959	26.933
Fireplaces ▲	2917	0.596	0.645	1.000	0.000	4.000	0.725	0.038
FireplaceQu ▲	2917	1.767	1.806	1.000	0.000	5.000	0.174	-1.764
GarageType*	2917	3.484	1.934	2.000	1.000	7.000	0.632	-1.416
GarageYrBlt ▲	2917	1976.164	26.703	1978	1872	2010	-0.690	-0.273
GarageFinish ▲	2917	1.715	0.897	2.000	0.000	3.000	0.138	-1.064
GarageCars	2917	1.766	0.762	2.000	0.000	5.000	-0.219	0.233
GarageArea	2917	472.248	214.762	480.000	0.000	1488.000	0.217	0.859
GarageQual ▲	2917	2.802	0.714	3.000	0.000	5.000	-3.270	10.128
GarageCond ▲	2917	2.810	0.711	3.000	0.000	5.000	-3.390	10.605
PavedDrive ▲	2917	1.831	0.537	2.000	0.000	2.000	-2.976	7.097
WoodDeckSF	2917	93.629	126.533	0.000	0.000	1424.000	1.844	6.730
OpenPorchSF	2917	47.280	67.119	26.000	0.000	742.000	2.528	10.991

EnclosedPorch	2917	23.114	64.263	0.000	0.000	1012.000	4.000	28.286
X3SsnPorch	2917	2.604	25.197	0.000	0.000	508.000	11.366	148.943
ScreenPorch	2917	16.073	56.202	0.000	0.000	576.000	3.943	17.715
PoolArea	2917	2.089	34.561	0.000	0.000	800.000	17.680	326.240
PoolQC	2917	0.015	0.243	0.000	0.000	5.000	17.733	327.543
Fence*	2917	4.493	1.092	5.000	1.000	5.000	-1.992	2.719
MiscFeature*	2917	2.066	0.364	2.000	1.000	5.000	5.060	24.741
MiscVal	2917	50.861	567.595	0.000	0.000	17000.000	21.928	562.332
MoSold*	2917	6.214	2.713	6.000	1.000	12.000	0.197	-0.455
YrSold*	2917	2.793	1.315	3.000	1.000	5.000	0.132	-1.157
SaleType*	2917	8.492	1.594	9.000	1.000	9.000	-3.727	13.621
SaleCondition*	2917	4.778	1.078	5.000	1.000	6.000	-2.788	7.209
SalePrice	1458	180932.919	79495.060	163000	34900	755000	1.877	6.484
TotalBathrooms	2917	1.757	0.642	2.000	0.000	5.000	0.303	-0.146
Remod (dummy)	2917	0.466	0.499	0.000	0.000	1.000	0.138	-1.982
Age	2917	23.545	20.890	15.000	-2.000	60.000	0.449	-1.340
IsNew (dummy)	2917	0.039	0.194	0.000	0.000	1.000	4.754	20.612
TotalSF	2917	2547.219	782.028	2452	334.000	10190	1.011	4.113

TotalPorchSF	2917	89.072	107.715	50.000	0.000	1207.000	2.243	9.999

Figure 3.1.1: Distribution of the SalePrice variable before log-transformation.

Skew = 1.879


Distribution of Sale Price
Density
0.000000
0.000002
0.000004
0.000006
0.000008
0
200000
400000
600000
Sale Price

Figure 3.1.2: Distribution of SalePrice upon taking the natural logarithm. Skew = .1211

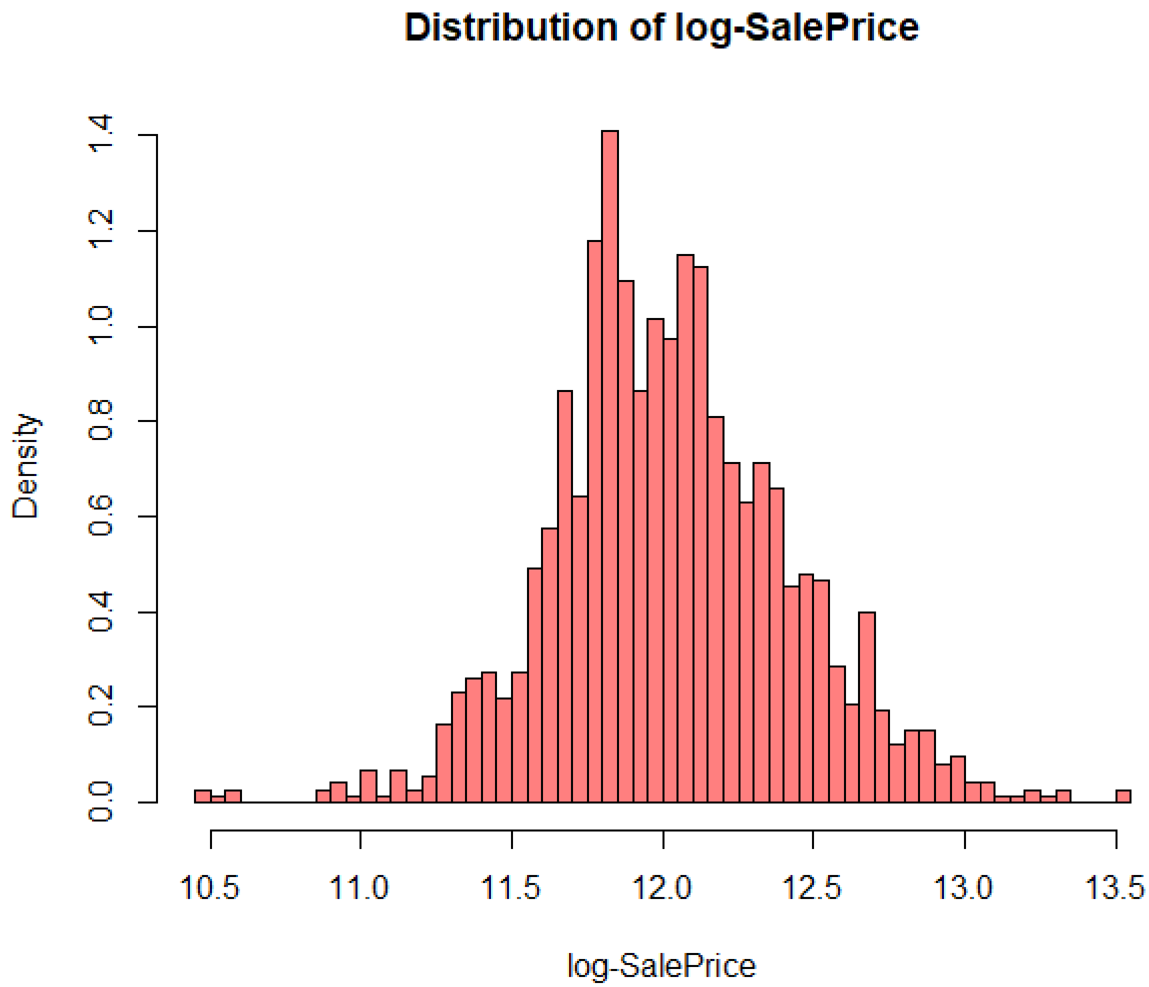


Figure 3.2.1: Q-Q plot of SalePrice before taking the natural logarithm.

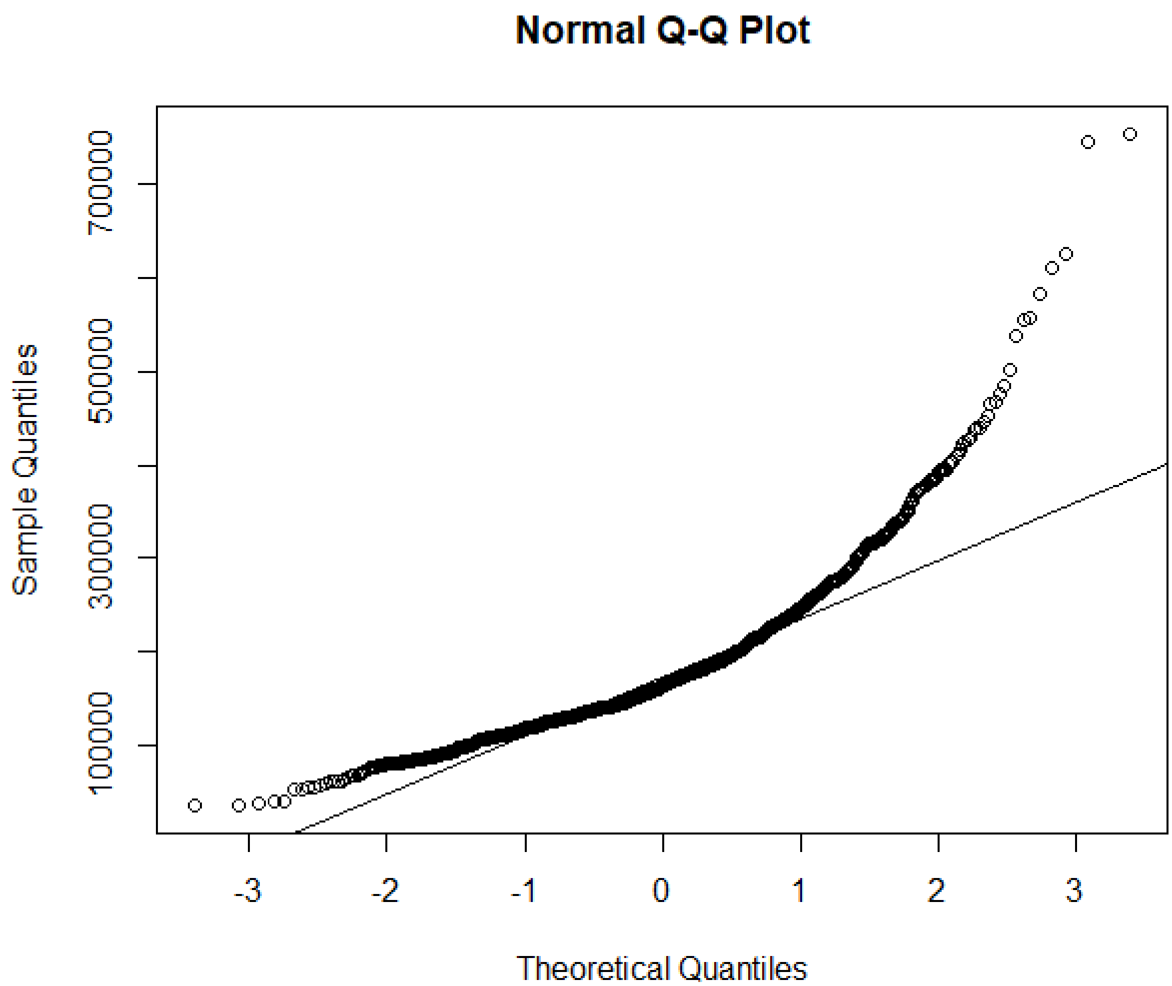


Figure 3.2.2: Q-Q plot of SalePrice upon taking the natural logarithm.

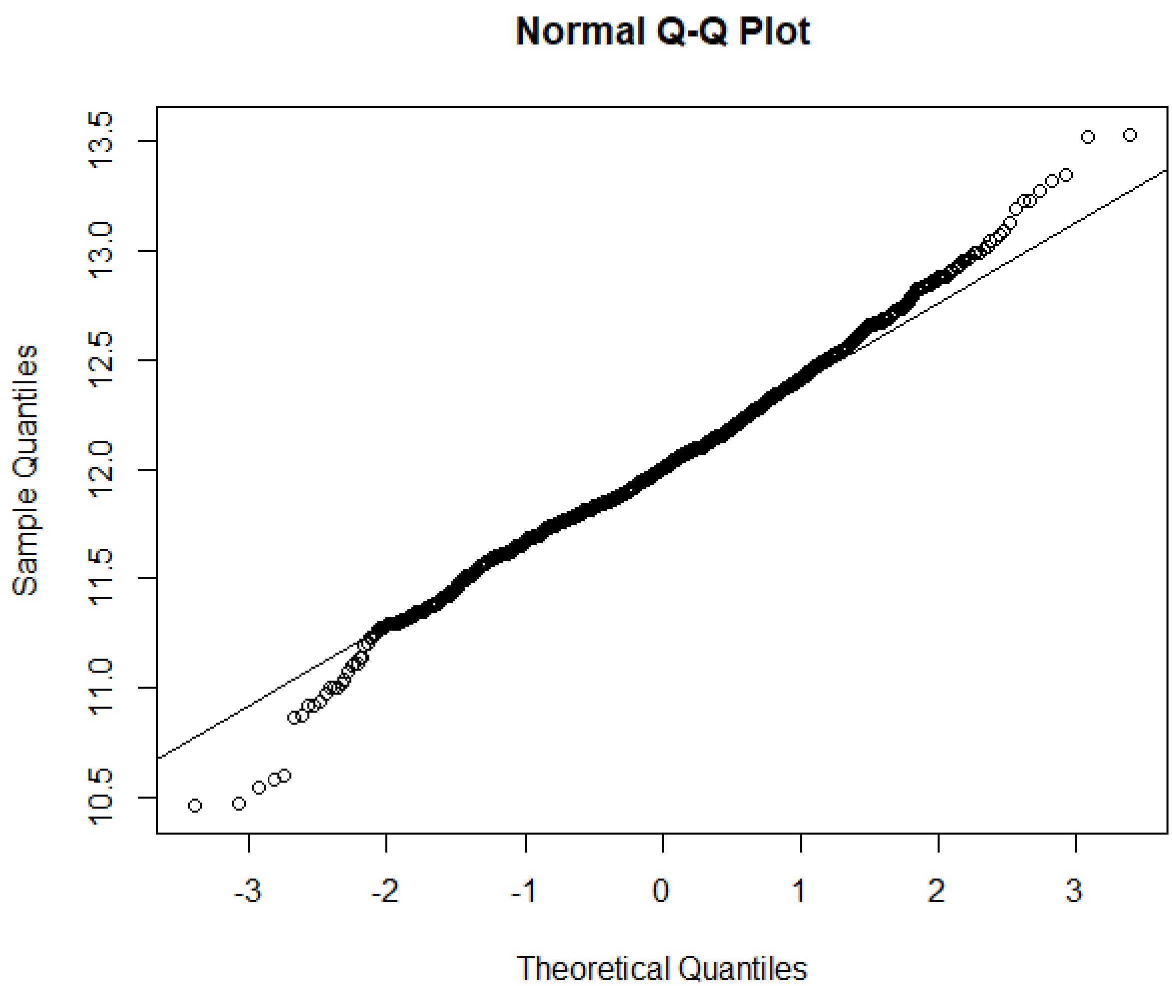


Figure 3.3: Relationship between SalePrice and OverallQual, highlighting the heteroskedastic and skewed nature of the untransformed price variable.

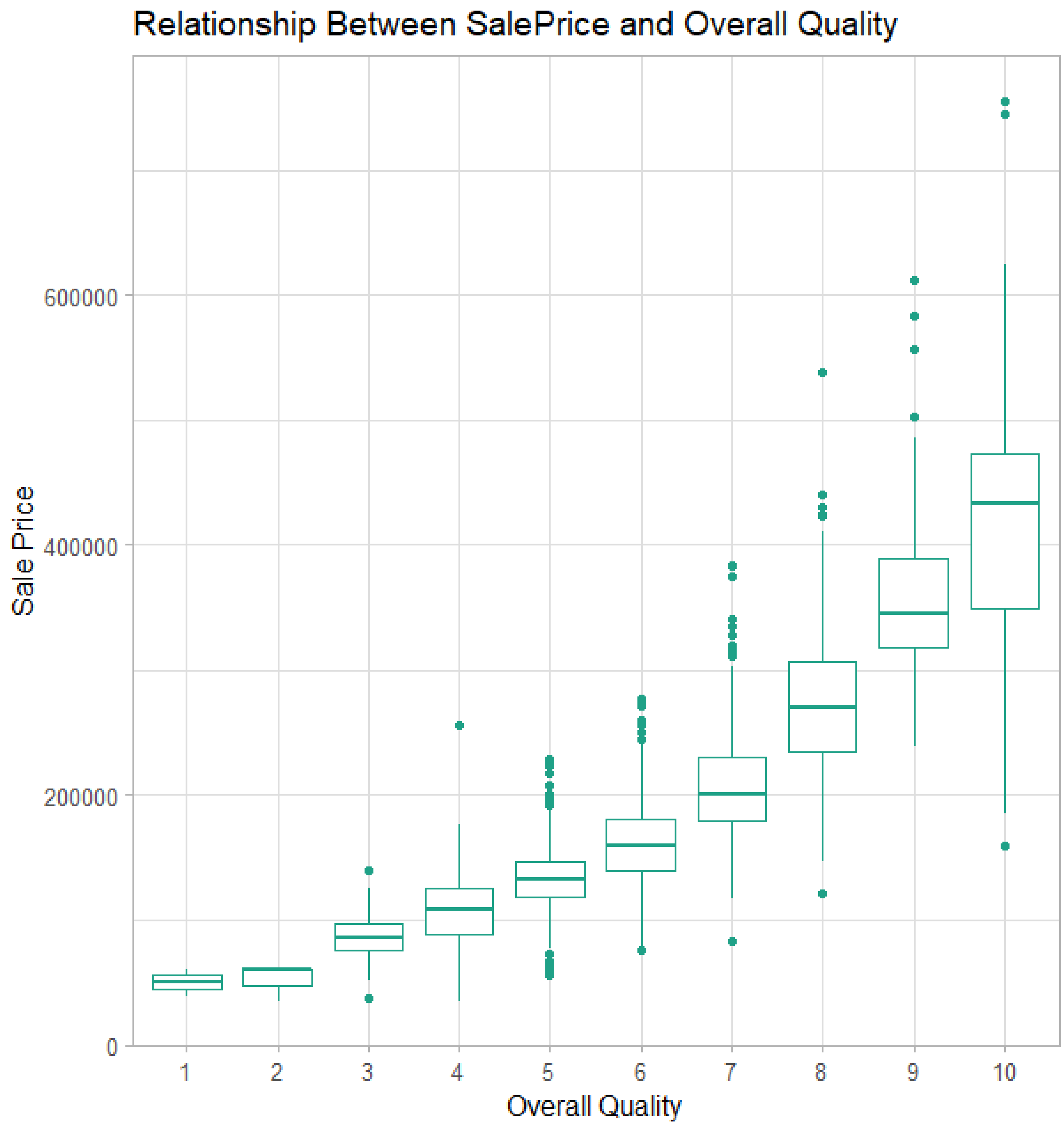


Figure 3.4: Histogram showing the differences in average home price between new and already built homes. The dashed line represents the median sale price across all homes. Note that N = 1460 here since that is the size of the training

set.

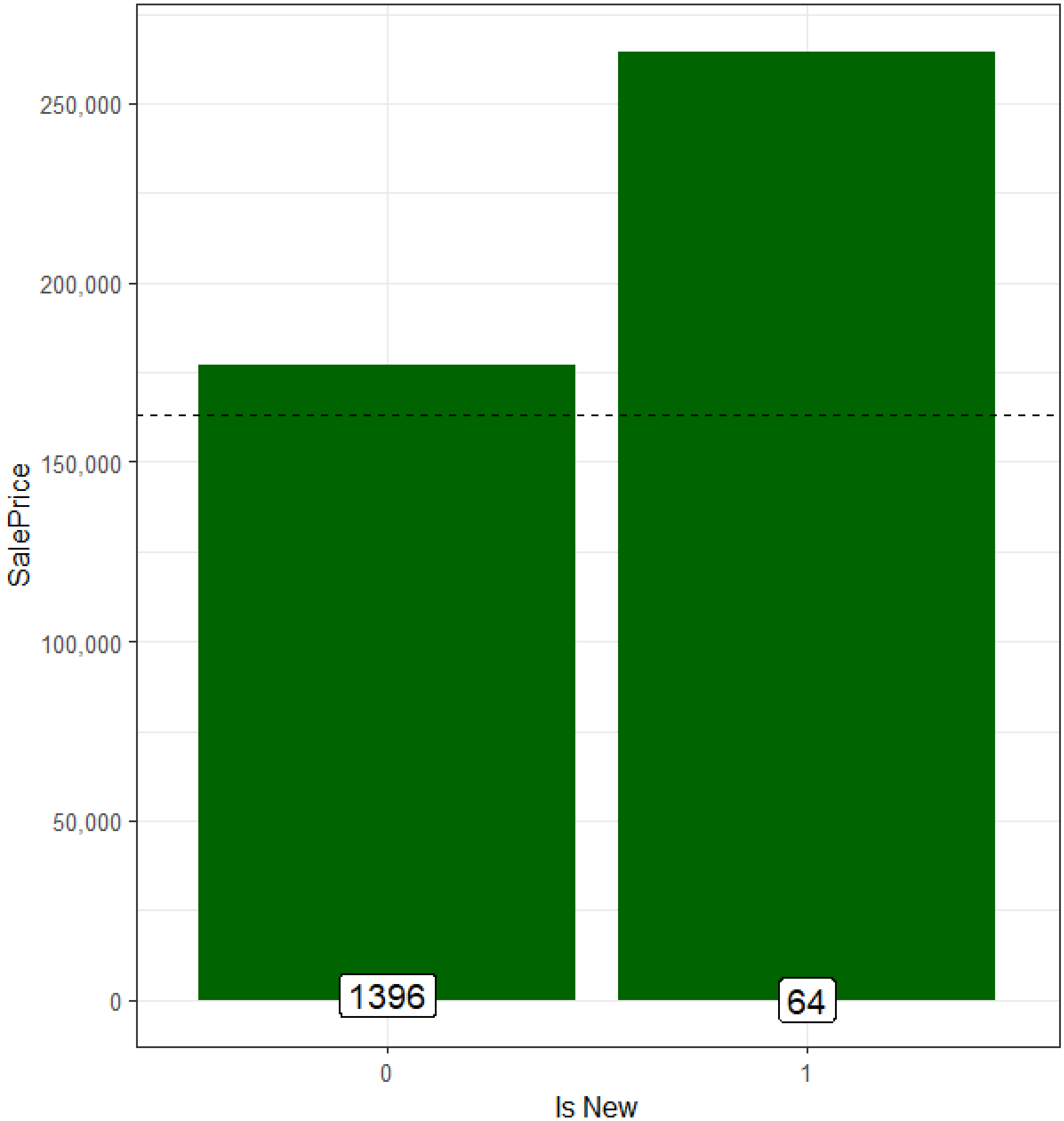
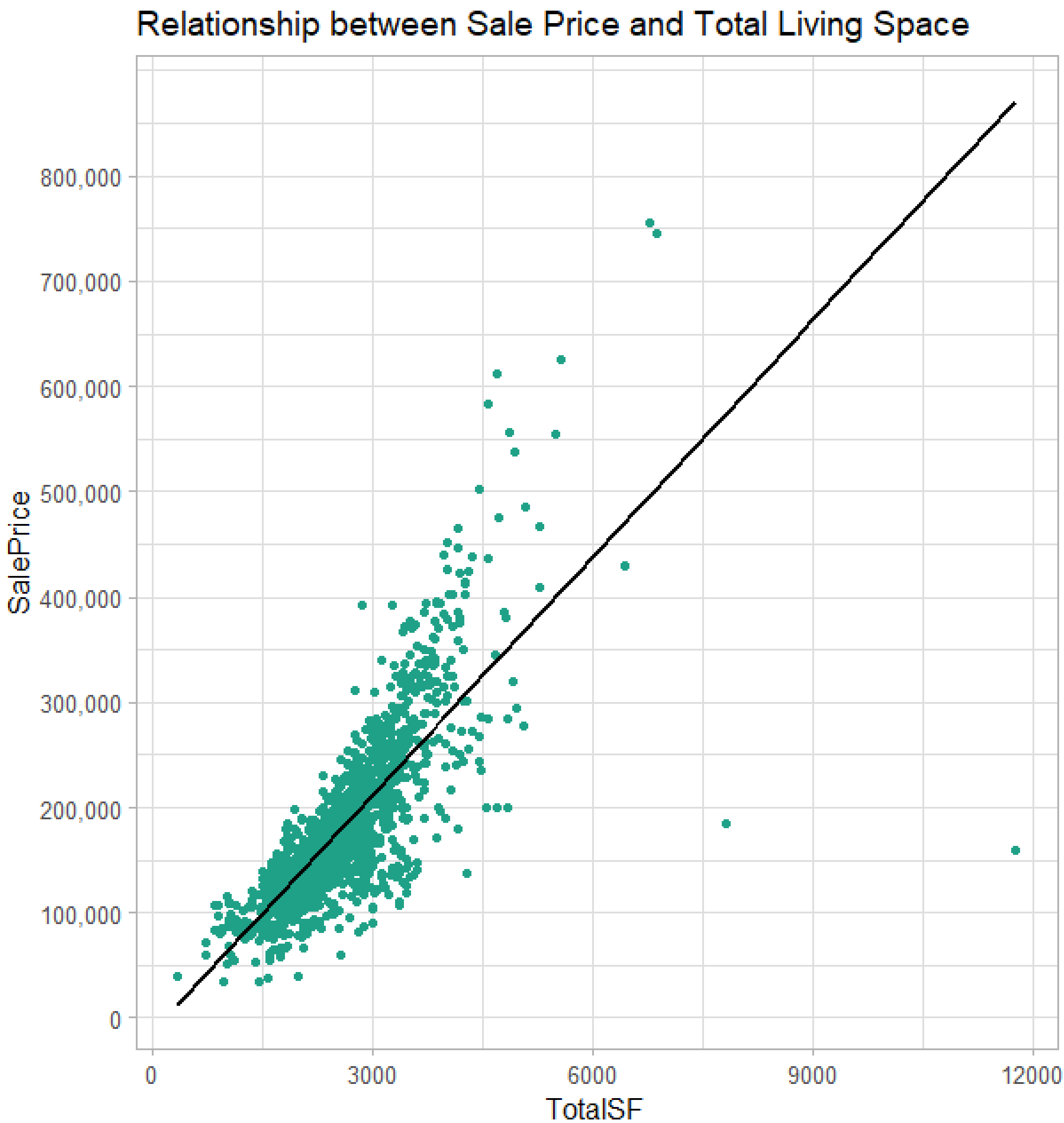


Figure 3.5: Relationship between SalePrice and Total living space. Two potential outliers to be removed are shown in the bottom right quadrant corresponding to observations 524 and 1299 in the training data. Before removal, r = 0.779. After removal, r = 0.829.



Appendix: R Code

```
# ECN 477 Research Project
library(psych)
library(dplyr)
library(plyr)
library(corrplot)
library(knitr)
library(ggplot2)
library(scales)
library(Rmisc)
library(ggrepel)
library(caret)
library(gridExtra)
train <- read.csv("train.csv", stringsAsFactors = F)
test <- read.csv("test.csv", stringsAsFactors = F)
dim(train)
str(train[,c(1:10, 81)])
test$Id <- NULL
train$Id <- NULL
test$SalePrice <- NA
all <- rbind(train, test)
dim(all)
# Sale price is right skewed
ggplot(data=all[!is.na(all$SalePrice),], aes(x=SalePrice)) +
  geom_histogram(fill="red", binwidth = 7500) +
  scale_x_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
summary(all$SalePrice)
# Correlation Matrix
# 37 numeric variables
# 10 numeric variables with corr > .50
numericVars <- which(sapply(all, is.numeric)) #index vector numeric variables
numericVarNames <- names(numericVars) #saving names vector for use later on
cat('There are', length(numericVars), 'numeric variables')
all_numVar <- all[, numericVars]
```

```
cor_numVar <- cor(all_numVar, use="pairwise.complete.obs") #correlations of all numeric variables
#sort on decreasing correlations with SalePrice
cor_sorted <- as.matrix(sort(cor_numVar[,'SalePrice'], decreasing = TRUE))
#select only high corelations
CorHigh <- names(which(apply(cor_sorted, 1, function(x) abs(x)>0.5)))
cor_numVar <- cor_numVar[CorHigh, CorHigh]
corrplot.mixed(cor_numVar, tl.col="black", tl.pos = "lt")
# Overall Quality
ggplot(data=all[!is.na(all$SalePrice),], aes(x=factor(OverallQual), y=SalePrice))+
  geom_boxplot(col='#1FA187') + labs(title ='Relationship Between SalePrice and Overall Quality',
                                x='Overall Quality', y ='Sale Price') +
  theme_light(base_size=12)
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
# Above Ground living area
ggplot(data=all[!is.na(all$SalePrice),], aes(x=GrLivArea, y=SalePrice))+
  geom_point(col='#1FA187') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +
  labs(x='Above Ground Living Area (sqft)', y='Sale Price', title='Relationship
      Between SalePrice and Above Ground Living Area') + theme_light(base_size=12)
  geom_text_repel(aes(label = ifelse(all$GrLivArea[!is.na(all$SalePrice)]>4500, rownames(all), '')))
# ID 524 and 1299 likely outliers as Qual is 10 and salePrice very low
all[c(524, 1299), c('SalePrice', 'GrLivArea', 'OverallQual')]
cor(all$SalePrice, all$GrLivArea, use= "pairwise.complete.obs")
cor(all$SalePrice, all$OverallQual, use= "pairwise.complete.obs")
# Correlation after taking out outliers
cor(all$SalePrice[-c(524, 1299)], all$GrLivArea[-c(524, 1299)], use= "pairwise.complete.obs")
#______________________________________________________________________________
# Cleaning and Imputing Missing Data
NAcol <- which(colSums(is.na(all)) > 0)
sort(colSums(sapply(all[NAcol], is.na)), decreasing = TRUE)
cat('There are', length(NAcol), 'columns with missing values')
# Pool Quality and PoolArea variables
# Creating generic  quality vector to represent multiple ordinal variables
```

```
# that follow the same quality leveling
### Ex   Excellent
### Gd   Good
### TA   Average
### Fa   Fair
### Po   Poor
### NA   None
Qualities <- c('None' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
# Pool Quality
all$PoolQC[is.na(all$PoolQC)] <- 'None'
all$PoolQC <-as.integer(revalue(all$PoolQC, Qualities))
table(all$PoolQC)
# Pool Area
all[all$PoolArea>0 & all$PoolQC==0, c('PoolArea', 'PoolQC', 'OverallQual')]
# Imputing Home quality as pool quality for 3 missing observations
all$PoolQC[2421] <- 2
all$PoolQC[2504] <- 3
all$PoolQC[2600] <- 2
# Misc. features:
##   Elev : Elevator
##   Gar2 : 2nd Garage (if not described in garage section)
##   Othr : Other
##   Shed : Shed (over 100 SF)
##   TenC : Tennis Court
##   NA   : None
## 2814 missing values (NAs)
## Converting MiscFeature into a factor
all$MiscFeature[is.na(all$MiscFeature)] <- 'None'
all$MiscFeature <- as.factor(all$MiscFeature)
ggplot(all[!is.na(all$SalePrice),], aes(x=MiscFeature, y=SalePrice)) +
  geom_bar(stat='summary', fun.y = "median", fill='red') +
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +
  geom_label(stat = "count", aes(label = ..count.., y = ..count..))
```

```
table(all$MiscFeature)
# Alley Variable:
##   Grvl : Gravel
##   Pave : Paved
##   NA   : No alley access
## 2721 NAs
## Converting Alley into a factor
all$Alley[is.na(all$Alley)] <- 'None'
all$Alley <- as.factor(all$Alley)
ggplot(all[!is.na(all$SalePrice),], aes(x=Alley, y=SalePrice)) +
  geom_bar(stat='summary', fun.y = "median", fill='red')+
  scale_y_continuous(breaks= seq(0, 200000, by=50000), labels = comma)
table(all$Alley)
# Fence Variable:
##   GdPrv : Gravel
##   MnPrv: Paved
##   GdWo : Good Wood
##   MnWw : Minimum Wood/Wire
##   NA   : No Fence
## 2348 NAs
## Converting Fence into a factor
all$Fence[is.na(all$Fence)] <- 'None'
table(all$Fence)
all[!is.na(all$SalePrice),] %>% group_by(Fence) %>%
  dplyr::summarise(median = median(SalePrice), counts=n())
all$Fence <- as.factor(all$Fence)
## Fireplace quality
## Gd   Good - Masonry Fireplace in main level
## TA   Average - Prefabricated Fireplace in main living area or Masonry Fireplace in basement
## Fa   Fair - Prefabricated Fireplace in basement
## Po   Poor - Ben Franklin Stove
## NA   No Fireplace
## No Missing Values
```

```
## Only obs missing correspond with homes without a fireplace
all$FireplaceQu[is.na(all$FireplaceQu)] <- 'None'
all$FireplaceQu<-as.integer(revalue(all$FireplaceQu, Qualities))
table(all$FireplaceQu)
table(all$Fireplaces)
sum(table(all$Fireplaces))
# LotFrontage variable: Linear feet of street connected to property
ggplot(all[!is.na(all$LotFrontage),], aes(x=as.factor(Neighborhood), y=LotFrontage)) +
  geom_bar(stat='summary', fun.y = "median", fill='red') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
for (i in 1:nrow(all)){
  if(is.na(all$LotFrontage[i])){
    all$LotFrontage[i] <- as.integer(median(all$LotFrontage[all$Neighborhood==all$Neighborhood[i]],
na.rm=TRUE))
  }
}
# LotShape: General shape of property
# Recoding as an ordinal variable
# Reg  Regular
# IR1  Slightly Irregular
# IR2  Moderately Irregular
# IR3  Irreuglar
all$LotShape<-as.integer(revalue(all$LotShape, c('IR3'=0, 'IR2'=1, 'IR1'=2, 'Reg'=3)))
table(all$LotShape)
sum(table(all$LotShape))
# LotConfig: Lot configuration
# Inside   Inside lot
# Corner   Corner lot
# CulDSac  Cul-de-sac
# FR2  Frontage on 2 sides of property
# FR3  Frontage on 3 sides of property
# Converting to a factor
ggplot(all[!is.na(all$SalePrice),], aes(x=as.factor(LotConfig), y=SalePrice)) +
```

```
  geom_bar(stat='summary', fun.y = "median", fill='red')+
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +
  geom_label(stat = "count", aes(label = ..count.., y = ..count..))
all$LotConfig <- as.factor(all$LotConfig)
table(all$LotConfig)
sum(table(all$LotConfig))
### Garage Variables (7 in Total)
# Imputing NAs for GarageYrBuilt using YearBuilt
all$GarageYrBlt[is.na(all$GarageYrBlt)] <- all$YearBuilt[is.na(all$GarageYrBlt)]
# Checking if all 157 NAs (GarageType) are the same observations
# among the variables with 157/159 NAs
length(which(is.na(all$GarageType) & is.na(all$GarageFinish)
             & is.na(all$GarageCond) & is.na(all$GarageQual)))
# Displaying 2 additional NAs
kable(all[!is.na(all$GarageType)
          & is.na(all$GarageFinish),
          c('GarageCars', 'GarageArea', 'GarageType',
            'GarageCond', 'GarageQual', 'GarageFinish')])
# Imputing mode.
all$GarageCond[2127] <- names(sort(-table(all$GarageCond)))[1]
all$GarageQual[2127] <- names(sort(-table(all$GarageQual)))[1]
all$GarageFinish[2127] <- names(sort(-table(all$GarageFinish)))[1]
# Displaying "fixed" house
kable(all[2127, c('GarageYrBlt', 'GarageCars', 'GarageArea',
                  'GarageType', 'GarageCond', 'GarageQual', 'GarageFinish')])
# Fixing 3 values for house 2577
all$GarageCars[2577] <- 0
all$GarageArea[2577] <- 0
all$GarageType[2577] <- NA
# Check if NAs of the character variables are now all 158
length(which(is.na(all$GarageType) &
               is.na(all$GarageFinish) &
               is.na(all$GarageCond) &
```

```
        is.na(all$GarageQual)))
# GarageType: Garage location
# 2Types   More than one type of garage
# Attchd   Attached to home
# Basment  Basement Garage
# BuiltIn  Built-In (Garage part of house - typically has room above garage)
# CarPort  Car Port
# Detchd   Detached from home
# NA       No Garage
# Converting to factor
all$GarageType[is.na(all$GarageType)] <- 'No Garage'
all$GarageType <- as.factor(all$GarageType)
table(all$GarageType)
# GarageFinish: Interior finish of the garage
# Fin  Finished
# RFn  Rough Finished
# Unf  Unfinished
# NA   No Garage
# Recoding as an ordinal variable
all$GarageFinish[is.na(all$GarageFinish)] <- 'None'
Finish <- c('None'=0, 'Unf'=1, 'RFn'=2, 'Fin'=3)
all$GarageFinish<-as.integer(revalue(all$GarageFinish, Finish))
table(all$GarageFinish)
# GarageQual: Garage quality
# Using the qualities vector to recode as an ordinal variable
all$GarageQual[is.na(all$GarageQual)] <- 'None'
all$GarageQual<-as.integer(revalue(all$GarageQual, Qualities))
table(all$GarageQual)
# GarageCond: Garage condition
# Same as above
all$GarageCond[is.na(all$GarageCond)] <- 'None'
all$GarageCond<-as.integer(revalue(all$GarageCond, Qualities))
table(all$GarageCond)
```

```
### Basement Variables (11 in Total)

### Five vars missing roughly 80 obs, other six only missing a couple

# Checking if all 79 NAs are the same observations among the variables with 80+ NAs

length(which(is.na(all$BsmtQual)

        & is.na(all$BsmtCond)

        & is.na(all$BsmtExposure)

        & is.na(all$BsmtFinType1)

        & is.na(all$BsmtFinType2)))

# BsmtFinType1 is the variable with 79 NAs

all[!is.na(all$BsmtFinType1) &
(is.na(all$BsmtCond)|is.na(all$BsmtQual)|is.na(all$BsmtExposure)|is.na(all$BsmtFinType2)), c('BsmtQual',
'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2')]

# Imputing modes

all$BsmtFinType2[333] <- names(sort(-table(all$BsmtFinType2)))[1]

all$BsmtExposure[c(949, 1488, 2349)] <- names(sort(-table(all$BsmtExposure)))[1]

all$BsmtCond[c(2041, 2186, 2525)] <- names(sort(-table(all$BsmtCond)))[1]

all$BsmtQual[c(2218, 2219)] <- names(sort(-table(all$BsmtQual)))[1]

# BasmtQual: Evaluates the height of the basement

# Using qualities vector to recode as ordinal

# Ex    Excellent (100+ inches)

# Gd    Good (90-99 inches)

# TA    Typical (80-89 inches)

# Fa    Fair (70-79 inches)

# Po    Poor (<70 inches

# NA    No Basement

all$BsmtQual[is.na(all$BsmtQual)] <- 'None'

all$BsmtQual<-as.integer(revalue(all$BsmtQual, Qualities))

table(all$BsmtQual)

# BsmtCond: Evaluates the general condition of the basement

# Ordinal

all$BsmtCond[is.na(all$BsmtCond)] <- 'None'

all$BsmtCond<-as.integer(revalue(all$BsmtCond, Qualities))

table(all$BsmtCond)

# BsmtExposure: Refers to walkout or garden level walls
```

```
# Ordinal

all$BsmtExposure[is.na(all$BsmtExposure)] <- 'None'

Exposure <- c('None'=0, 'No'=1, 'Mn'=2, 'Av'=3, 'Gd'=4)

all$BsmtExposure<-as.integer(revalue(all$BsmtExposure, Exposure))

table(all$BsmtExposure)

# BsmtFinType1: Rating of basement finished area

# Ordinal

all$BsmtFinType1[is.na(all$BsmtFinType1)] <- 'None'

FinType <- c('None'=0, 'Unf'=1, 'LwQ'=2, 'Rec'=3, 'BLQ'=4, 'ALQ'=5, 'GLQ'=6)

all$BsmtFinType1<-as.integer(revalue(all$BsmtFinType1, FinType))

table(all$BsmtFinType1)

# BsmtFinType2: Rating of basement finished area (if multiple types)

# Ordinal

all$BsmtFinType2[is.na(all$BsmtFinType2)] <- 'None'

FinType <- c('None'=0, 'Unf'=1, 'LwQ'=2, 'Rec'=3, 'BLQ'=4, 'ALQ'=5, 'GLQ'=6)

all$BsmtFinType2<-as.integer(revalue(all$BsmtFinType2, FinType))

table(all$BsmtFinType2)

# Remaining Variables

# Displaying remaining NAs. Using BsmtQual as a reference

# for the 79 houses without basement agreed upon earlier

all[(is.na(all$BsmtFullBath)|is.na(all$BsmtHalfBath)|is.na(all$BsmtFinSF1)|is.na(all$BsmtFinSF2)|is.na(all$BsmtU
nfSF)|is.na(all$TotalBsmtSF)),

    c('BsmtQual', 'BsmtFullBath', 'BsmtHalfBath', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF')]

# BsmtFullBath: Basement full bathrooms

# An integer variable

all$BsmtFullBath[is.na(all$BsmtFullBath)] <-0

table(all$BsmtFullBath)

# BsmtHalfBath: Basement half bathrooms

# An integer variable

all$BsmtHalfBath[is.na(all$BsmtHalfBath)] <-0

table(all$BsmtHalfBath)

# BsmtFinSF1: Type 1 finished square feet

# An integer variable
```

```
all$BsmtFinSF1[is.na(all$BsmtFinSF1)] <-0
# BsmtFinSF2: Type 2 finished square feet
# An integer variable
all$BsmtFinSF2[is.na(all$BsmtFinSF2)] <-0
# BsmtUnfSF: Unfinished square feet of basement area
# An integer variable
all$BsmtUnfSF[is.na(all$BsmtUnfSF)] <-0
# TotalBsmtSF: Total square feet of basement area
# An integer variable
all$TotalBsmtSF[is.na(all$TotalBsmtSF)] <-0
## Masonry veneer type and Masonry Area
# Checking if the 23 houses with veneer area NA are also NA in the veneer type
length(which(is.na(all$MasVnrType) & is.na(all$MasVnrArea)))
#find the one that should have a MasVnrType
all[is.na(all$MasVnrType) & !is.na(all$MasVnrArea), c('MasVnrType', 'MasVnrArea')]
#fix this veneer type by imputing the mode
all$MasVnrType[2611] <- names(sort(-table(all$MasVnrType)))[2] #taking the 2nd value as the 1st is 'none'
all[2611, c('MasVnrType', 'MasVnrArea')]
## Masonry Veneer type
# BrkCmn   Brick Common
# BrkFace  Brick Face
# CBlock   Cinder Block
# None     None
# Stone    Stone
all$MasVnrType[is.na(all$MasVnrType)] <- 'None'
all[!is.na(all$SalePrice),] %>% group_by(MasVnrType) %>%
          dplyr::summarise(median = median(SalePrice),
          counts=n()) %>% arrange(median)
# Assigning ordinality
Masonry <- c('None'=0, 'BrkCmn'=0, 'BrkFace'=1, 'Stone'=2)
all$MasVnrType<-as.integer(revalue(all$MasVnrType, Masonry))
table(all$MasVnrType)
# MasVnrArea: Masonry veneer area in square feet
```

```
all$MasVnrArea[is.na(all$MasVnrArea)] <-0
## MSZoning: Identifies the general zoning classification of the sale
# 4 Missing values
# Imputing the mode
all$MSZoning[is.na(all$MSZoning)] <- names(sort(-table(all$MSZoning)))[1]
all$MSZoning <- as.factor(all$MSZoning)
table(all$MSZoning)
sum(table(all$MSZoning))
## Kitchen quality
# 1 missing obs
# Converting to ordinal using qualities vector
all$KitchenQual[is.na(all$KitchenQual)] <- 'TA' #replace with most common value
all$KitchenQual<-as.integer(revalue(all$KitchenQual, Qualities))
table(all$KitchenQual)
sum(table(all$KitchenQual))
# KitchenAbvGr: Number of Kitchens above grade (ground)
# no missing obs
table(all$KitchenAbvGr)
sum(table(all$KitchenAbvGr))
## Utilities
# 2 missing obs
# AllPub   All public Utilities (E,G,W,& S)
# NoSewr   Electricity, Gas, and Water (Septic Tank)
# NoSeWa   Electricity and Gas Only
# ELO  Electricity only
table(all$Utilities)
# Only 1 home without access to all public utilities
# Not enough variation to make useful prediciton
# Removing data from analysis
all$Utilities <- NULL
## Functional: Home functionality
# 1 missing obs
# Typ  Typical Functionality
```

```
# Min1 Minor Deductions 1
# Min2 Minor Deductions 2
# Mod  Moderate Deductions
# Maj1 Major Deductions 1
# Maj2 Major Deductions 2
# Sev  Severely Damaged
# Sal  Salvage only
# Ordinal variable
# Imputing mode for the 1 missing obs
all$Functional[is.na(all$Functional)] <- names(sort(-table(all$Functional)))[1]
all$Functional <- as.integer(revalue(all$Functional, c('Sal'=0, 'Sev'=1, 'Maj2'=2, 'Maj1'=3, 'Mod'=4, 'Min2'=5,
'Min1'=6, 'Typ'=7)))
table(all$Functional)
sum(table(all$Functional))
## Exterior Variables (4 in Total)
# Exterior1st: Exterior covering on house
# 1 missing obs
# Categorical
# Imputing Mode
all$Exterior1st[is.na(all$Exterior1st)] <- names(sort(-table(all$Exterior1st)))[1]
all$Exterior1st <- as.factor(all$Exterior1st)
table(all$Exterior1st)
sum(table(all$Exterior1st))
# Exterior2nd: Exterior covering on house (if more than one material)
# 1 missing obs
# Categorical
# Imputing Mode
all$Exterior2nd[is.na(all$Exterior2nd)] <- names(sort(-table(all$Exterior2nd)))[1]
all$Exterior2nd <- as.factor(all$Exterior2nd)
table(all$Exterior2nd)
sum(table(all$Exterior2nd))
# ExterQual: Evaluates the quality of the material on the exterior
# no missing obs
```

```
# Converting to ordinal using qualities vector
all$ExterQual<-as.integer(revalue(all$ExterQual, Qualities))
table(all$ExterQual)
sum(table(all$ExterQual))
# ExterCond: Evaluates the present condition of the material on the exterior
# no missing obs
# Converting to ordinal using qualities vector
all$ExterCond<-as.integer(revalue(all$ExterCond, Qualities))
table(all$ExterCond)
sum(table(all$ExterCond))
# Electrical: Electrical system
# 1 missing obs
# SBrkr    Standard Circuit Breakers & Romex
# FuseA    Fuse Box over 60 AMP and all Romex wiring (Average)
# FuseF    60 AMP Fuse Box and mostly Romex wiring (Fair)
# FuseP    60 AMP Fuse Box and mostly knob & tube wiring (poor)
# Mix      Mixed
# Categorical // Converting to a factor
# Imputing mode for missing obs
all$Electrical[is.na(all$Electrical)] <- names(sort(-table(all$Electrical)))[1]
all$Electrical <- as.factor(all$Electrical)
table(all$Electrical)
# SaleType: Type of sale
# 1 missing obs
# Categorical
# Imputing mode
all$SaleType[is.na(all$SaleType)] <- names(sort(-table(all$SaleType)))[1]
all$SaleType <- as.factor(all$SaleType)
table(all$SaleType)
sum(table(all$SaleType))
# SaleCondition: Condition of sale
# No missing obs
# Categorical
```

```
all$SaleCondition <- as.factor(all$SaleCondition)
table(all$SaleCondition)
sum(table(all$SaleCondition))
##################################################
### Remaining Character Variables (No missing obs)
## 15 remaining
Charcol <- names(all[,sapply(all, is.character)])
Charcol
cat('There are', length(Charcol), 'remaining columns with character values')
# Foundation: Type of foundation
# Categorical
all$Foundation <- as.factor(all$Foundation)
table(all$Foundation)
sum(table(all$Foundation))
# Heating: Type of heating
# Categorical
all$Heating <- as.factor(all$Heating)
table(all$Heating)
sum(table(all$Heating))
# HeatingQC: Heating quality and condition
# Converting obs to ordinal values using the Qualities vector
all$HeatingQC<-as.integer(revalue(all$HeatingQC, Qualities))
table(all$HeatingQC)
sum(table(all$HeatingQC))
# CentralAir: Central air conditioning
# Binary variable
all$CentralAir<-as.integer(revalue(all$CentralAir, c('N'=0, 'Y'=1)))
table(all$CentralAir)
sum(table(all$CentralAir))
# RoofStyle: Type of Roof
# Categorical
all$RoofStyle <- as.factor(all$RoofStyle)
table(all$RoofStyle)
```

```
sum(table(all$RoofStyle))
# RoofMatl: Roof material
# Categorical
all$RoofMatl <- as.factor(all$RoofMatl)
table(all$RoofMatl)
sum(table(all$RoofMatl))
# LandContour: Flatness of the property
# Categorical
all$LandContour <- as.factor(all$LandContour)
table(all$LandContour)
sum(table(all$LandContour))
# LandSlope: Slope of property
# Gtl  Gentle slope
# Mod  Moderate Slope
# Sev  Severe Slope
# Ordinal
all$LandSlope<-as.integer(revalue(all$LandSlope, c('Sev'=0, 'Mod'=1, 'Gtl'=2)))
table(all$LandSlope)
sum(table(all$LandSlope))
# BldgType: Type of dwelling
# Categorical
all$BldgType <- as.factor(all$BldgType)
table(all$BldgType)
sum(table(all$BldgType))
# HouseStyle: Style of dwelling
# Categorical
all$HouseStyle <- as.factor(all$HouseStyle)
table(all$HouseStyle)
sum(table(all$HouseStyle))
# Neighborhood: Physical locations within Ames city limits
# Categorical
all$Neighborhood <- as.factor(all$Neighborhood)
table(all$Neighborhood)
```

```
sum(table(all$Neighborhood))
# Condition1: Proximity to various conditions
# Categorical
all$Condition1 <- as.factor(all$Condition1)
table(all$Condition1)
sum(table(all$Condition1))
# Condition2: Proximity to various conditions (if more than one is present)
# Categorical
all$Condition2 <- as.factor(all$Condition2)
table(all$Condition2)
sum(table(all$Condition2))
# Street: Type of road access to property
# Grvl Gravel
# Pave Paved
# Ordinal
all$Street<-as.integer(revalue(all$Street, c('Grvl'=0, 'Pave'=1)))
table(all$Street)
sum(table(all$Street))
# PavedDrive: Paved driveway
# Y    Paved
# P    Partial Pavement
# N    Dirt/Gravel
# Ordinal
all$PavedDrive<-as.integer(revalue(all$PavedDrive, c('N'=0, 'P'=1, 'Y'=2)))
table(all$PavedDrive)
sum(table(all$PavedDrive))
#______________________________________________________________________________
str(all$YrSold)
str(all$MoSold)
# Converting month sold into a factor
all$MoSold <- as.factor(all$MoSold)
# Plotting price both across and within years observed
ySold <- ggplot(all[!is.na(all$SalePrice),], aes(x=as.factor(YrSold), y=SalePrice)) +
```

```
  geom_bar(stat='summary', fun.y = "median", fill='red')+ xlab(' Year Sold') +
  scale_y_continuous(breaks= seq(0, 800000, by=25000), labels = comma) +
  geom_label(stat = "count", aes(label = ..count.., y = ..count..)) +
  coord_cartesian(ylim = c(0, 200000)) +
  geom_hline(yintercept=163000, linetype="dashed", color = "blue") #dashed line is median SalePrice
mSold <- ggplot(all[!is.na(all$SalePrice),], aes(x=MoSold, y=SalePrice)) +
  geom_bar(stat='summary', fun.y = "median", fill='red')+
  scale_y_continuous(breaks= seq(0, 800000, by=25000), labels = comma) +
  geom_label(stat = "count", aes(label = ..count.., y = ..count..)) + xlab("Month Sold")+
  coord_cartesian(ylim = c(0, 200000)) +
  geom_hline(yintercept=163000, linetype="dashed", color = "blue") #dashed line is median SalePrice
grid.arrange(ySold, mSold, widths=c(1,2))
# MSSubClass: Identifies the type of dwelling involved in the sale.
# Categorical
str(all$MSSubClass)
all$MSSubClass <- as.factor(all$MSSubClass)
all$MSSubClass<-revalue(all$MSSubClass, c('20'='1 story 1946+',
                                          '30'='1 story 1945-',
                                          '40'='1 story unf attic',
                                          '45'='1,5 story unf',
                                          '50'='1,5 story fin',
                                          '60'='2 story 1946+',
                                          '70'='2 story 1945-',
                                          '75'='2,5 story all ages',
                                          '80'='split/multi level',
                                          '85'='split foyer',
                                          '90'='duplex all style/age',
                                          '120'='1 story PUD 1946+',
                                          '150'='1,5 story PUD all',
                                          '160'='2 story PUD 1946+',
                                          '180'='PUD multilevel',
                                          '190'='2 family conversion'))
str(all$MSSubClass)
```

```
# Data cleaned: 56 numeric vars, 23 categorical
numericVars <- which(sapply(all, is.numeric)) #index vector numeric variables
factorVars <- which(sapply(all, is.factor)) #index vector factor variables
cat('There are', length(numericVars), 'numeric variables, and',
    length(factorVars), 'categorical variables')
#describe(all)
#write.csv(describe(all),"C:/Users/alial/Documents/ECN477/CleanedSumStats.csv")
# Replotting correlation matrix with new imputed data
# Now 16 variables with corr > .50
all_numVar <- all[, numericVars]
cor_numVar <- cor(all_numVar, use="pairwise.complete.obs") #correlations of all numeric variables
# Sorting by decreasing correlations with SalePrice
cor_sorted <- as.matrix(sort(cor_numVar[,'SalePrice'], decreasing = TRUE))
#select only high corelations
CorHigh <- names(which(apply(cor_sorted, 1, function(x) abs(x)>0.5)))
cor_numVar <- cor_numVar[CorHigh, CorHigh]
corrplot.mixed(cor_numVar, tl.col="black", tl.pos = "lt", tl.cex = 0.7,cl.cex = .7, number.cex=.7)
# Typo found: GarageYrBlt is entered as 2207 with YearRemod=2007, Correcting Error
all$GarageYrBlt[2593] <- 2007
# Flattening bathroom variables into one
all$TotBathrooms <- all$FullBath + (all$HalfBath*0.5)
+ all$BsmtFullBath + (all$BsmtHalfBath*0.5)
tb1 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=as.factor(TotBathrooms), y=SalePrice))+
  geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
tb2 <- ggplot(data=all, aes(x=as.factor(TotBathrooms))) +
  geom_histogram(stat='count')
grid.arrange(tb1, tb2)
# House age and remodel
all$Remod <- ifelse(all$YearBuilt==all$YearRemodAdd, 0, 1) #0=No Remodeling, 1=Remodeling
all$Age <- as.numeric(all$YrSold)-all$YearRemodAdd
ggplot(data=all[!is.na(all$SalePrice),], aes(x=Age, y=SalePrice))+
  geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
```

```
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
# Negative correlation between age and price
cor(all$SalePrice[!is.na(all$SalePrice)], all$Age[!is.na(all$SalePrice)])
ggplot(all[!is.na(all$SalePrice),], aes(x=as.factor(Remod), y=SalePrice)) +
  geom_bar(stat='summary', fun.y = "median", fill='blue') +
  geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=6) +
  scale_y_continuous(breaks= seq(0, 800000, by=50000), labels = comma) +
  theme_grey(base_size = 18) +
  geom_hline(yintercept=163000, linetype="dashed") #dashed line is median SalePrice
# Creating a dummy variable to represnt new homes
all$IsNew <- ifelse(all$YrSold==all$YearBuilt, 1, 0)
table(all$IsNew)
ggplot(all[!is.na(all$SalePrice),], aes(x=as.factor(IsNew), y=SalePrice)) +
  geom_bar(stat='summary', fun.y = "median", fill='darkgreen') +
  geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=5) +
  scale_y_continuous(breaks= seq(0, 800000, by=50000), labels = comma) +
  theme_bw(base_size = 12) +
  xlab("Is New") +
  geom_hline(yintercept=163000, linetype="dashed") #dashed line is median SalePrice
# Creating a Total sq ft variable
all$TotalSF <- all$GrLivArea + all$TotalBsmtSF
ggplot(data=all[!is.na(all$SalePrice),], aes(x=TotalSF, y=SalePrice))+
  geom_point(col='#1FA187') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +
  labs(x = 'TotalSF', title='Relationship between Sale Price and Total Living Space')+
  theme_light(base_size = 12)
  geom_text_repel(aes(label = ifelse(all$GrLivArea[!is.na(all$SalePrice)]>4500, rownames(all), '')))
# Corr before taking out outliers
cor(all$SalePrice, all$TotalSF, use= "pairwise.complete.obs")
# Corr after
cor(all$SalePrice[-c(524, 1299)], all$TotalSF[-c(524, 1299)], use= "pairwise.complete.obs")
all[c(524, 1299), c('SalePrice', 'TotalSF', 'OverallQual')]
all$TotalPorchSF <- all$OpenPorchSF + all$EnclosedPorch + all$X3SsnPorch + all$ScreenPorch
```

```
cor(all$SalePrice, all$TotalPorchSF, use= "pairwise.complete.obs")

all$YrSold <- as.factor(all$YrSold) # Converting year sold to a factor

# dropping two outliers

all <- all[-c(524, 1299),]

cor(all$SalePrice, all$TotBathrooms, use= "pairwise.complete.obs")

# Drop highly correlated variables

dropVars <- c('YearRemodAdd', 'GarageYrBlt', 'GarageArea', 'GarageCond', 'TotalBsmtSF', 'TotalRmsAbvGrd',
'BsmtFinSF1')

all <- all[,!(names(all) %in% dropVars)]

#write.csv(describe(all),"C:/Users/alial/Documents/ECN477/CleanedSumStatsDataSection.csv")

#______________________________________________________________________________

# price transformation here

# Distribution of sale price

hist(all$SalePrice, col=rgb(1,0,0, 0.5),
    main = "Distribution of Sale Price",
    breaks = 50,
    freq = F,
    xlab = "Sale Price")

# QQplot

qqnorm(all$SalePrice)

qqline(all$SalePrice)

# Converting to log-prices

all$SalePrice <- log(all$SalePrice)

qqnorm(all$SalePrice)

qqline(all$SalePrice)

# Skew is much lower now at .12

skew(all$LogSalePrice)

hist(all$SalePrice, col=rgb(1,0,0, 0.5),
    breaks = 50,
    freq = F,
    main = "Distribution of log-SalePrice",
    xlab = "log-SalePrice")

#______________________________________________________________________________
```

```
numericVarNames <- numericVarNames[!(numericVarNames %in% c('MSSubClass', 'MoSold', 'YrSold',
'SalePrice', 'OverallQual', 'OverallCond'))] #numericVarNames was created before having done anything

numericVarNames <- append(numericVarNames, c('Age', 'TotalPorchSF', 'TotBathrooms', 'TotalSqFeet'))

DFnumeric <- all[, names(all) %in% numericVarNames]

DFfactors <- all[, !(names(all) %in% numericVarNames)]

DFfactors <- DFfactors[, names(DFfactors) != 'SalePrice']

# Num of numeric vs factor variables

cat('There are', length(DFnumeric), 'numeric variables, and', length(DFfactors), 'factor variables')

# Normalizing predictors

for(i in 1:ncol(DFnumeric)){

  if (abs(skew(DFnumeric[,i]))>0.8){

    DFnumeric[,i] <- log(DFnumeric[,i] +1)

  }

}

PreNum <- preProcess(DFnumeric, method=c("center", "scale"))

print(PreNum)

#_______________________________________#####IGNORING FOR NOW

DFnorm <- predict(PreNum, DFnumeric)

dim(DFnorm) #2917 obs // 29 numeric predictors

# one-hot encoding

DFdummies <- as.data.frame(model.matrix(~.-1, DFfactors))

dim(DFdummies)

#check if some values are absent in the test set

ZerocolTest <- which(colSums(DFdummies[(nrow(all[!is.na(all$SalePrice),])+1):nrow(all),])==0)

colnames(DFdummies[ZerocolTest])

DFdummies <- DFdummies[,-ZerocolTest] #removing predictors

#check if some values are absent in the train set

ZerocolTrain <- which(colSums(DFdummies[1:nrow(all[!is.na(all$SalePrice),]),])==0)

colnames(DFdummies[ZerocolTrain])

DFdummies <- DFdummies[,-ZerocolTrain] #removing predictor

# Taking out variables with less than 10 obs

fewOnes <- which(colSums(DFdummies[1:nrow(all[!is.na(all$SalePrice),]),])<10)

colnames(DFdummies[fewOnes])
```

```
DFdummies <- DFdummies[,-fewOnes] #removing predictors

dim(DFdummies)

#______________________________________________________________

#combining all (now numeric) predictors into one dataframe

combined <- cbind(DFnorm, DFdummies)

#########

# Price var already transformed

# Building train and test sets

train1 <- combined[!is.na(all$SalePrice),]

test1 <- combined[is.na(all$SalePrice),]

##########################################################################################

library(glmnet)

tmp_coeffs <- coef(cvfit, s = "lambda.min")

cvfit <- glmnet::cv.glmnet(x=train1, y=all$SalePrice[!is.na(all$SalePrice)])

coef(cvfit, s = "lambda.1se")

###############################

set.seed(6300)

my_control <-trainControl(method="cv", number=5)

lassoGrid <- expand.grid(alpha = 1, lambda = seq(0.001,0.1,by = 0.0005))

lasso_mod <- train(x=train1, y=all$SalePrice[!is.na(all$SalePrice)], method='glmnet', trControl= my_control,
tuneGrid=lassoGrid)

lasso_mod$bestTune

fit_test <- predict(lasso_mod, newdata = test1, s= lasso_mod$lamda.min)

fit_testDF <- as.data.frame(fit_test)

fit_testDF <- exp(fit_testDF)

# Writing fit_test results to xlsx sheet

# Score of .12748

library("writexl")

write.csv(fit_testDF,"C:/Users/alial/Documents/ECN477/HousePriceProj/fit_test.csv")

write_xlsx(all$SalePrice[!is.na(all$SalePrice)],"C:/Users/alial/Documents/ECN477/HousePriceProj/y_values.xlsx")

###########################################################

# Variable importance

lassoVarImp <- varImp(lasso_mod,scale=F)
```

```
lassoImportance <- lassoVarImp$importance
varsSelected <- length(which(lassoImportance$Overall!=0))
varsNotSelected <- length(which(lassoImportance$Overall==0))
cat('Lasso uses', varsSelected, 'variables in its model, and did not select', varsNotSelected, 'variables.')
################################################################################
fin_coefs1 <- predict(lasso_mod$finalModel, type="coef")
fin_coefs <- as.data.frame(as.matrix(predict(lasso_mod$finalModel, type="coef")))
write_xlsx(fin_coefs,"C:/Users/alial/Documents/ECN477/HousePriceProj/lasso_results.xlsx")
#######################################################################################
y_var <- all$SalePrice[!is.na(all$SalePrice)]
min(lasso_mod$results$RMSE)
fit = glmnet(as.matrix(train1), model.matrix(all$SalePrice),
        lambda=cv.glmnet(as.matrix(train1), model.matrix(all$SalePrice)["lambda.1se"]))
fit = glmnet(as.matrix(train1), y_var)
plot(fit, xvar='lambda')
plot(fit, xvar='norm',abline(v=.01))
coef_list <- coef(fit)
coef_list <- as.data.frame(as.matrix(coef_list))
write_xlsx(coef_list,"C:/Users/alial/Documents/ECN477/HousePriceProj/coef_list.xlsx")
# baseline model
base_mod <- lm(SalePrice ~ TotalSF + OverallQual, data=all)
base_fit_test <- predict(base_mod, newdata = test1)
base_fit_testDF <- as.data.frame(base_fit_test)
base_fit_testDF <- exp(base_fit_testDF)
write.csv(base_fit_testDF,"C:/Users/alial/Documents/ECN477/HousePriceProj/BASE_fit_test.csv")
# Score of .18780
hist(all$GarageCars,col='#1FA187',
   xlab="Garage Bay Size",
   main="Histogram of GarageCars")
hist(all$GarageFinish,col='#1FA187',
   xlab="GarageFinish",
   main="Histogram of GarageFinish")
library(stargazer)
```

```
library(estimatr)
stargazer(base_mod,
    type = "html",
    title = "Baseline OLS Model",
    out = "C:/Users/alial/Documents/ECN477/HousePriceProj/BaselineMod.html")
base_mod_robust <-lm_robust(SalePrice ~ TotalSF + OverallQual, data=all)
summary(base_mod_robust)
# Generating OLS model using Lasso variable coefficients
lasso_OLS <- lm(SalePrice ~ TotalSF + OverallQual + KitchenQual +
        Age + GarageCars + GarageFinish + GrLivArea + IsNew, data=all)
summary(lasso_OLS)
stargazer(lasso_OLS,
    type = "html",
    title = "OLS Model Using Lasso's Selected Variables",
    out = "C:/Users/alial/Documents/ECN477/HousePriceProj/LASSOMod.html")
coef(lasso_OLS)
lasso_fit_test <- predict(lasso_OLS, newdata = test1)
lasso_fit_testDF <- as.data.frame(lasso_fit_test)
lasso_fit_testDF <- exp(lasso_fit_testDF)
write.csv(lasso_fit_testDF,"C:/Users/alial/Documents/ECN477/HousePriceProj/LASSO_fit_test.csv")
# Score of 0.22867
s5_lasso_mod <- lm(SalePrice ~ TotalSF + OverallQual, data=all)
summary(s5_lasso_mod)
# mod comparison
stargazer(base_mod, lasso_OLS,
    type="html",
    title="Baseline OLS model vs Lasso Selected OLS model",
    out = "C:/Users/alial/Documents/ECN477/HousePriceProj/MOD-Comparison.html")
plot(y_var, resid(base_mod),
  ylab="Residuals",
  xlab="Log-SalePrice",
  main="Base OLS Model",
  abline(0,0),
```

```
    col='#1FA187')
plot(y_var, resid(lasso_OLS),
    ylab="Residuals",
    xlab="Log-SalePrice",
    main="Lasso Selected OLS Model",
    abline(0,0),
    col='#1FA187')
stargazer(base_mod, lasso_OLS,
        se = starprep(base_mod, lasso_OLS),
        type="html",
        title="Baseline OLS model vs Lasso Selected OLS model",
        out = "C:/Users/alial/Documents/ECN477/HousePriceProj/RobustMOD-Comparison.html")
min(all$SalePrice)
correlate <- all %>% dplyr::select(SalePrice, TotalSF, OverallQual,
                        KitchenQual, Age, GarageCars,
                        GarageFinish, GrLivArea)
stargazer(cor(correlate), type = "html",
        title="Correlation Matrix of Lasso-Selected OLS Effects",
        out="C:/Users/alial/Documents/ECN477/HousePriceProj/lasso-ols-corrmat.html")
cor(all$SalePrice, all$TotalSF)
cor(all$SalePrice[0:1457], all$GrLivArea[0:1457])
```