

MSDS 6372 PROJECT 1

Introduction

“Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. but this Kaggle competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence” (Source: [Kaggle competition website](#)). This project, created for MSDS 6372 investigates obvious as well as non-obvious factors that influence sale prices of houses in Ames, Iowa. We will conduct two analyses for this project: one to facilitate easy interpretation for realtors, contractors, and home buyers and the other to provide for the most accurate sale price predictions. To measure the predictiveness of our model, we will be utilizing [Kaggle](#), a platform for data science competitions, that currently has a competition to create an optimal model to predict the sale prices of a set of homes in Ames, Iowa.

Data Description

The dataset was compiled by Dr. Dean De Cock, professor of statistics & director of assessment at Truman State University in Kirksville, Missouri ([link](#)). The data were gathered directly from the Ames Assessor's Office and represents the sale of individual properties in Ames, Iowa, from 2006 to 2010 and can be obtained from the Kaggle competition “[House Prices: Advanced Regression Techniques](#).” It consists of 79 explanatory variables that describe elements of homes in Ames, Iowa. The training data set consists of 1460 homes with the test data set providing 1459 more observations, a total of 2919 homes.

Exploratory Analysis

Several variables that measured square footage were continuous; the remaining variables were categorical. In order to use the data set in R, the research team translated the categorical variables into numeric values. Additionally, the team addressed missing data and “not applicable” values using the impute function with either mode or median. After analyzing plots, the team decided to take the log of the values for the sale price (SalePrice), basement type 1 finish square footage (BSMTFinSF1), total basement square footage (TotalBSMTSF), and the above ground living area square footage (GrLivArea). Some extreme values were removed from the data, specifically any basement type 1 finish square footage (BSMTFinSF1) greater than 5600 square feet, any above ground living area square footage (GrLivArea) greater than 4000 square feet, and any dollar value of any miscellaneous feature (MiscVal) greater than \$8000, as these values did not seem to explain the features of the given population of homes. The team also added a few variables that seemed intuitive: the above ground living area square footage (GrLivArea) and the total basement square footage (TotalBSMTSF) were added to create a total square footage value and the total number of baths was created by summing the basement baths, above ground baths, and $0.5 \times$ the half baths in both the basement and above ground. All of the code for this exploratory data analysis can be found in the appendix.

ANALYSIS QUESTION 1

Problem Statement

Century 21 in Ames, Iowa, has commissioned this research team to present a model that provides an estimate for the sale prices of homes that is formed to facilitate the easy interpretation of parameters for use in helping real estate agents, contractors, and prospective buyers gain insight into the important factors that influence housing prices in Ames, Iowa.

Model Selection

In order to create a model for realtors, contractors, and prospective home buyers, the research team used a model averaging technique with a forward selection, least absolute shrinkage and selection operator (LASSO), and a least angle regression (LAR) technique. After analyzing the effects that were selected the highest percentage in all three of these models, the team created a custom model. This created the selected model, with an adjusted R^2 of 0.8648:

$$\log(\text{SalePrice}) = \beta_0 + \beta_1 \text{Total_Sq_Footage} + \beta_2 \text{OverallQual} + \beta_3 \text{YearBuilt} + \beta_4 \text{YearRemodAdd} + \beta_5 \text{GarageCars} + \beta_6 \text{CentralAir}$$

Assumptions

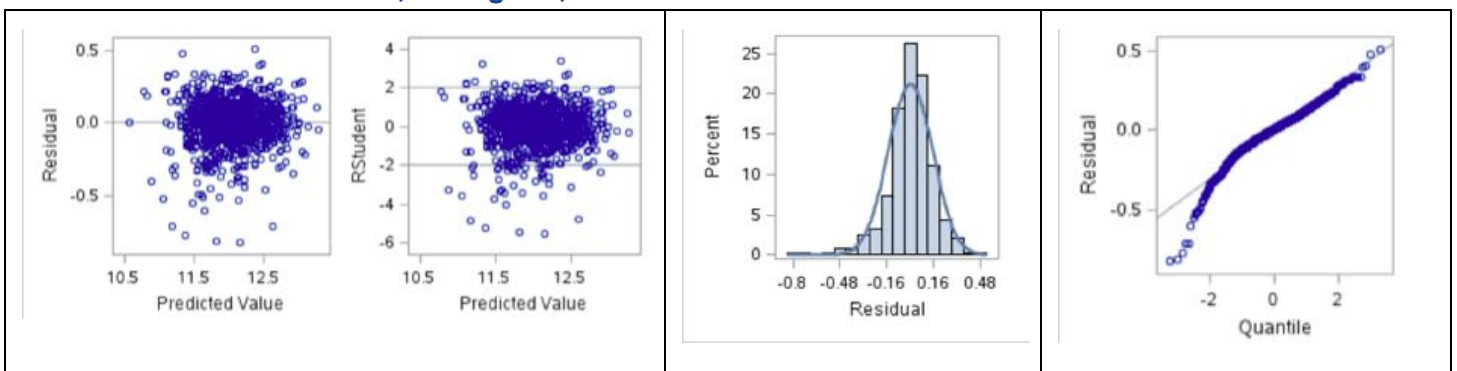
Normality: Visual analysis of the residual scatter plot, residual QQ-plot, and the generally normal histogram of residuals supports normality (see plots below). The histogram appears slightly left-skewed, but not enough to be concerned with the normality assumption.

Linearity: By looking at the residual plot, the residuals appear in a random cluster, so the data appear to be linear. Therefore, linearity will be assumed.

Equality of Variance: From the residual plot, the standard deviations appear to be relatively the same. Therefore, equality of variance can be assumed for the transformed data.

Independence: The research team assumed the observations about each house were independent of one another.

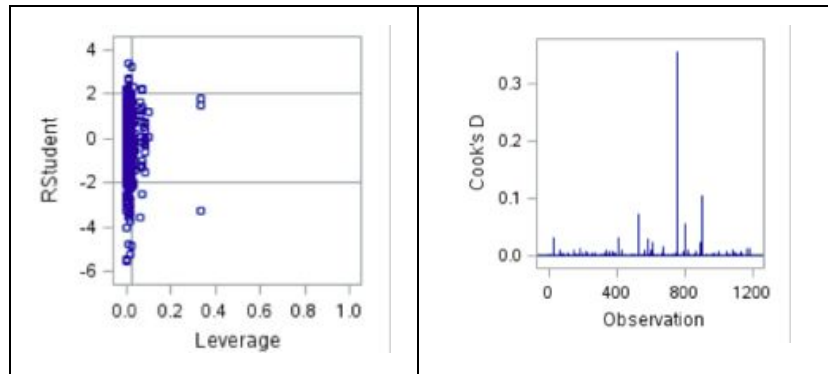
Residual Plots, Histogram, & QQ-Plot



Influential Points: According to the leverage plot, there are no points that are extremely concerning. According to the Cook's D plot, the highest value is about 0.4 and there are not any

single influential points (see plots below). The team feels comfortable that the assumptions are met here, so the analysis will be continued.

Leverage Plot & Cook's D Plot



Parameter Interpretation

This model provides real estate agents, contractors, and prospective home buyers with an analysis that allows for an easy interpretation of factors that influence the home prices in Ames. Specifically, the effects chosen were the total square footage, the overall material and finish of the home (on a scale from 1-10), the original construction date, the remodel date, the size of the garage in car capacity, and whether or not the home has central air conditioning. Of these effects, the overall quality and central air are categorical variables, while the others are continuous. The estimate for the total square footage is 0.0002, which provides us with an estimate of $e^{0.0002} = 1.0002$. In other words, there is a 0.02% increase in the predicted median sale price of homes with each one-foot increase in the total size of the home while holding all other factors constant. A 95% confidence interval for this increase is between $e^{0.000226} = 0.023\%$ and $e^{0.000259} = 0.026\%$. The estimate for the year that the home was built is 0.0013, suggesting that for every one year increase in the original construction date, the median sale price of the homes increases by 0.13% when holding all other factors constant. A 95% confidence interval for this increase is between 0.09% and 0.17%. The estimate for the year that the home was remodeled is 0.0023, suggesting that for every one year increase in the remodel date, the median sale price of the homes increases by 0.23% while holding all other factors constant. A 95% confidence interval for this increase is between 0.176% and 0.28%. The estimate for the size of the garage according to its car capacity is 0.0657, suggesting that for every one-unit increase in garage car capacity, the median sale price of the homes increases by 6.79% while holding all other factors constant. A 95% confidence interval for this increase is between 5.13% and 8.44%. The two categorical effects that were chosen for this model are overall quality and central air. Overall quality is rated on a scale of 1-10, with 1 being the lowest score at “very poor” and 10 being the highest score at “very excellent.” With the reference variable as a rating of 10, it is evident that scores lower than 10 decrease the median value of home prices, as expected. Changing from a rating of 10 to a rating of 1 decreases the predicted median sale price by $e^{-0.89} = 0.41$, or a decrease of about 59% while controlling for all other effects. A 95% confidence interval for this decrease is between $e^{-1.21} = 0.298$ and $e^{-0.57} = 0.566$, or between a 70% and 43% decrease. The multiplicative changes for the other quality ratings can be found using similar logic using the estimates from the parameter estimate table below. For central air, a change from having a central air system to not having a central air system is expected to

decrease the predicted median home price by $e^{-0.143} = 0.867$, or a 13.3% decrease. It is interesting to note that the factors that have the largest effects on the predicted median sale price are garage car capacity, overall quality, and central air. This information can be useful for real estate agents, contractors, and home buyers when looking at important features in the homes.

Parameter Estimates

Parameter	Estimate		Standard Error	t Value	Pr > t	95% Confidence Limits	
Intercept	4.514612187	B	0.57450360	7.86	<.0001	3.387443766	5.641780609
total_sq_footage	0.000242121		0.00000844	28.70	<.0001	0.000225570	0.000258673
OverallQual 1	-0.890773961	B	0.16266222	-5.48	<.0001	-1.209915081	-0.571632840
OverallQual 2	-0.815086276	B	0.10396742	-7.84	<.0001	-1.019068971	-0.611103581
OverallQual 3	-0.660053245	B	0.06254154	-10.55	<.0001	-0.782758909	-0.537347581
OverallQual 4	-0.527263414	B	0.05204769	-10.13	<.0001	-0.629380298	-0.425146529
OverallQual 5	-0.434303438	B	0.04893016	-8.88	<.0001	-0.530303767	-0.338303110
OverallQual 6	-0.375353108	B	0.04792915	-7.83	<.0001	-0.469389464	-0.281316753
OverallQual 7	-0.294316904	B	0.04647620	-6.33	<.0001	-0.385502595	-0.203131213
OverallQual 8	-0.193461904	B	0.04624243	-4.18	<.0001	-0.284188928	-0.102734881
OverallQual 9	-0.050135695	B	0.04984002	-1.01	0.3147	-0.147921162	0.047649771
OverallQual 10	0.000000000	B
YearBuilt	0.001304636		0.00020342	6.41	<.0001	0.000905534	0.001703737
YearRemodAdd	0.002296720		0.00027634	8.31	<.0001	0.001754544	0.002838896
GarageCars	0.065736470		0.00780280	8.42	<.0001	0.050427474	0.081045467
CentralAir N	-0.143127126	B	0.01913209	-7.48	<.0001	-0.180664023	-0.105590229
CentralAir Y	0.000000000	B

ANALYSIS QUESTION 2

Problem Statement

Century 21 in Ames, Iowa, also asked the research team to create models aimed at being the most predictive to determine the sale prices of homes, using the training data set and test data set noted in the Data Description above. Predictiveness of models are evaluated by the Adjusted R-squared value, the Akaike information criterion (AIC), and the average square error (ASE), as well as through the Kaggle competition site.







Model Selection

Model 1

The custom model was built based on the backward elimination model (model 2 below) and then manually adding and removing variables based on the significance level. The resulting backward elimination model included 29 variables, and an adjusted R^2 value of 0.9246. This model gave us the best kaggle score of 0.12059:

$$\log(\text{SalePrice}) = \beta_0 + \beta_1 \text{MSZoning} + \beta_2 \text{LotConfig} + \beta_3 \text{LandSlope} + \beta_4 \text{Neighborhood} + \beta_5 \text{Condition1} + \beta_6 \text{Condition2} + \beta_7 \text{BldgType} + \beta_8 \text{HouseStyle} + \beta_9 \text{OverallQual} + \beta_{10} \text{OverallCond} + \beta_{11} \text{YearBuilt} + \beta_{12} \text{YearRemodAdd} + \beta_{13} \text{ExterQual} + \beta_{14} \text{ExterCond} + \beta_{15} \text{Foundation} + \beta_{16} \text{BsmtQual} + \beta_{17} \text{BsmtUnfSF} + \beta_{18} \text{Heating} + \beta_{19} \text{HeatingQC} + \beta_{20} \text{CentralAir} + \beta_{21} \text{Fireplaces} + \beta_{22} \text{GarageArea} + \beta_{23} \text{GarageCond} + \beta_{24} \text{WoodDeckSF} + \beta_{25} \text{ScreenPorch} + \beta_{26} \text{SaleCondition} + \beta_{27} \text{total_sq_footage} + \beta_{28} \text{total_baths} + \beta_{29} \log \text{GrLivArea}$$

**See Model 2 for discussion on model assumptions*

Kaggle Score						
793	▼ 60	himanshudce		0.12058	6	11d
794	new	manisha pednekar		0.12058	4	7d
795	▼ 61	Emilyqxt		0.12059	3	1mo
796	▲ 582	Thejas Prasad		0.12059	34	now
Your Best Entry ↑ You advanced 268 places on the leaderboard! Your submission scored 0.12059, which is an improvement of your previous score of 0.12322. Great job! Tweet this!						
797	▲ 1281	David Stroud		0.12060	17	2h
798	▼ 63	shabuqiao		0.12061	20	1mo

Model 2

The backward elimination model began with all variables included. Again, using the Akaike Information Criteria, R then sequentially removed variables from the model until there are no variables left to remove based on the criteria. The resulting backward elimination model included 40 variables and an adjusted R^2 value of 0.9288:

$$\log(\text{SalePrice}) = \beta_0 + \beta_1 \text{MSZoning} + \beta_2 \text{Street} + \beta_3 \text{LotConfig} + \beta_4 \text{LandSlope} + \beta_5 \text{Neighborhood} + \beta_6 \text{Condition1} + \beta_7 \text{Condition2} + \beta_8 \text{BldgType} + \beta_9 \text{HouseStyle} + \beta_{10} \text{OverallQual} + \beta_{11} \text{OverallCond} + \beta_{12} \text{YearBuilt} + \beta_{13} \text{YearRemodAdd} + \beta_{14} \text{MasVnrArea} + \beta_{15} \text{ExterQual} + \beta_{16} \text{ExterCond} + \beta_{17} \text{Foundation} + \beta_{18} \text{BsmtQual} + \beta_{19} \text{BsmtFinType2} + \beta_{20} \text{BsmtFinSF2} + \beta_{21} \text{BsmtUnfSF} + \beta_{22} \text{Heating} + \beta_{23} \text{HeatingQC} + \beta_{24} \text{CentralAir} + \beta_{25} \text{X1stFlrSF} + \beta_{26} \text{X2ndFlrSF} + \beta_{27} \text{BedroomAbvGr} + \beta_{28} \text{TotRmsAbvGrd} + \beta_{29} \text{Fireplaces} + \beta_{30} \text{GarageArea} + \beta_{31} \text{GarageCond} + \beta_{32} \text{WoodDeckSF} + \beta_{33} \text{ScreenPorch} + \beta_{34} \text{Fence} + \beta_{35} \text{SaleCondition} + \beta_{36} \text{total_sq_footage} + \beta_{37} \text{total_baths} + \beta_{38} \log \text{BsmtFinSF1} + \beta_{39} \log \text{TotalBsmtSF} + \beta_{40} \log \text{GrLivArea}$$

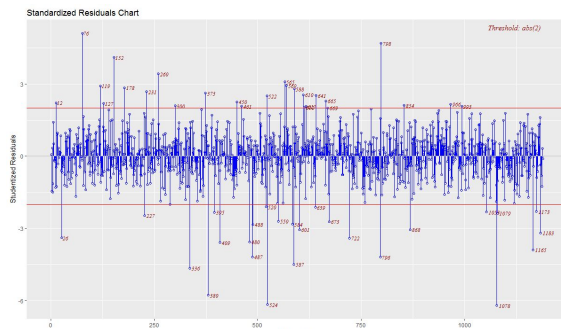
Assumptions: Models 2 and 3

Normality, linearity, and Equality of Variance: The residual plots (below) for each model showed even spreads and good distributions. There was no evidence against the assumptions for any of these models.

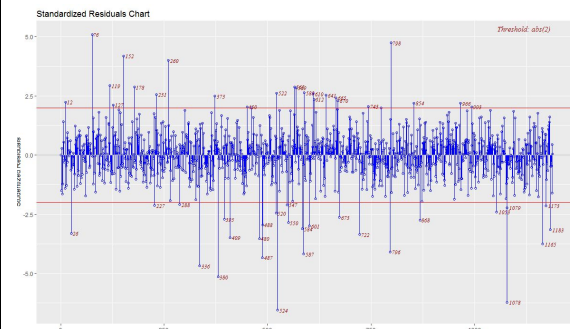
Independence: Independence is assumed.

Residual Plots

Custom Backward (Model 1)



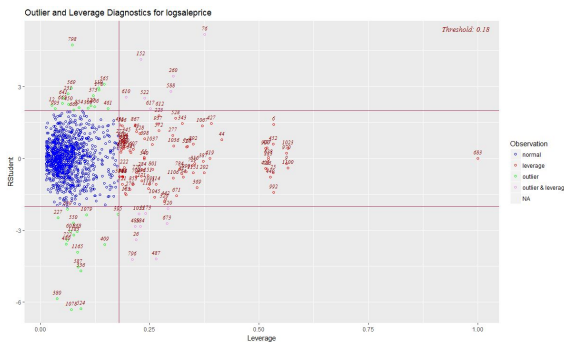
Backward (Model 2)



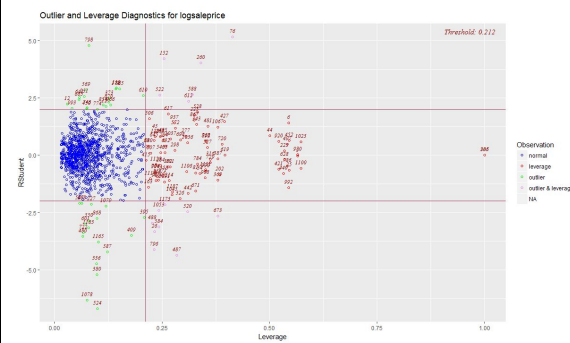
Influential Points: There were a few influential points that had both high leverage and large Cook's D values. The research team reviewed these points and concluded that these are valid observations from the same population and proceed with keeping them in the models. See leverage plots and Cook's D plots below.

Leverage Plots

Custom Backward (Model 1)



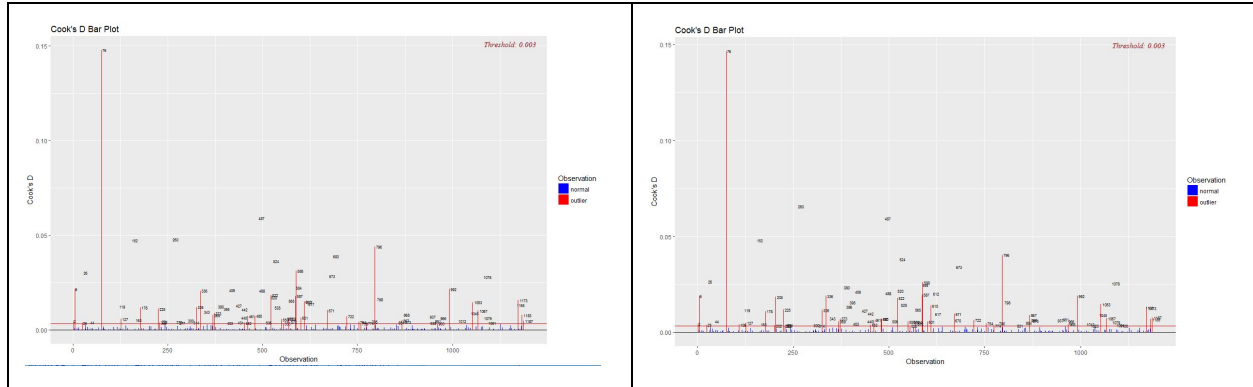
Backward (Model 2)



Cook's D Plots

Custom Backward (Model 1)

Backward (Model 2)



Model 3

For this model, a LASSO selection technique was used with an external cross validation technique to create the model:

$$\log(\text{SalePrice}) = \beta_0 + \beta_1 \text{total_sq_footage} + \beta_2 \text{GarageCars} + \beta_3 \text{YearBuilt} + \beta_4 \text{YearRemodAdd} + \beta_5 \text{GarageArea} + \beta_6 \text{FireplaceQu} + \beta_7 \text{BsmtFinSF1} + \beta_8 \text{KitchenQual} + \beta_9 \text{BsmtQual} + \beta_{10} \text{HeatingQC} + \beta_{11} \text{CentralAir}$$

This model has an adjusted R^2 of 0.8261, a BIC of -2075.79, an AIC of -1461.444, a CVPress of 13.83856, and a Kaggle Score of 0.16234.

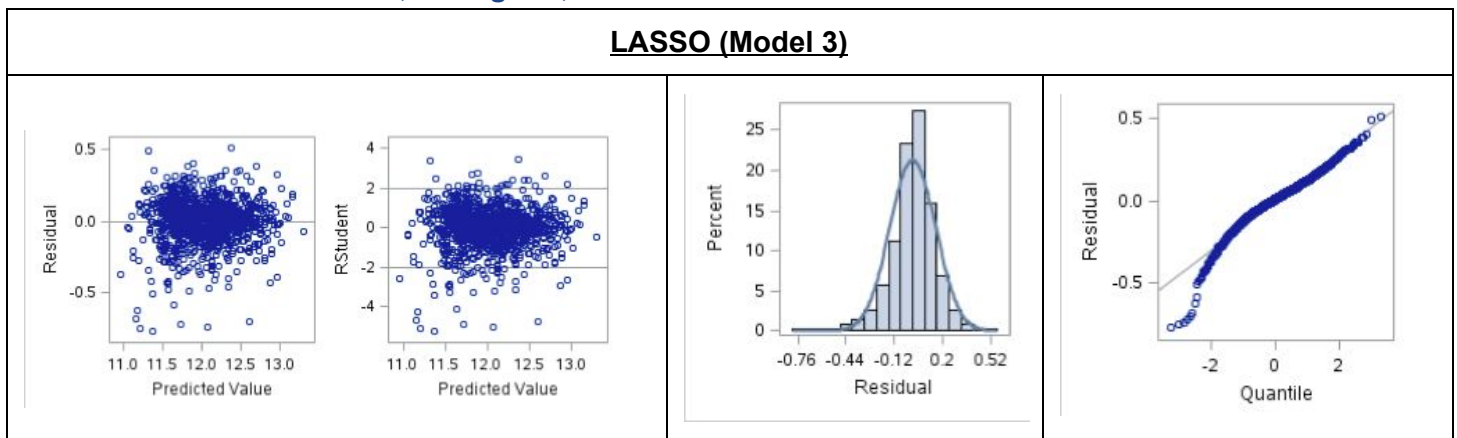
Assumptions: Model 3

Normality: Visual analysis of the residual scatter plot, residual QQ-plot, and the generally normal histogram of residuals supports normality (see plots below).

Linearity: By looking at the residual plot, the residuals appear in a relatively random cluster, so the data appear to be fairly linear.

Equality of Variance: From the residual plot, the standard deviations appear to be relatively the same. Therefore, equality of variance will be assumed for the data.

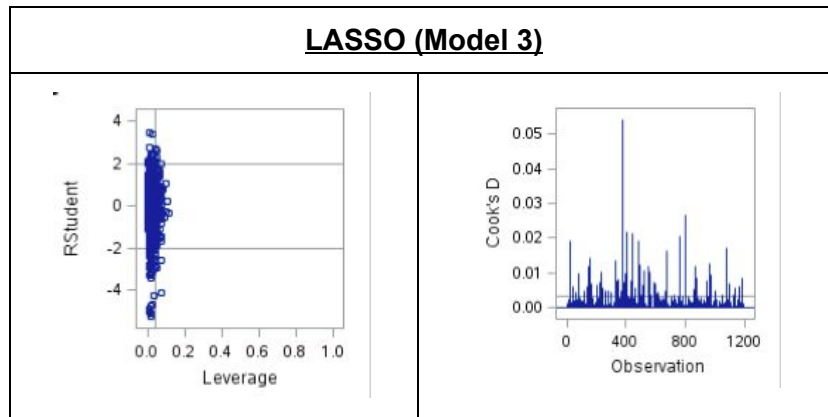
Residual Plots, Histogram, & QQ-Plot



Independence: Independence is assumed.

Influential Points: According to our Cook's D plot, there may be one point that has a slightly higher value, but since this value only has a Cook's D of 0.05, we are not highly concerned. The leverage plot also supports the conclusion that there are not any highly concerning influential points (see plots below).

Leverage Plot & Cook's D Plot



Comparing Competing Models

Test Set Models	Adj. R^2	AIC	ASE (test)	Kaggle Score	Kaggle Scoring Models	Adj. R^2	AIC	ASE (test)	Kaggle Score
Model 1: Custom	0.9246	-4887.98	0.01324789	0.12059	Model 1: Custom	0.9246	-4887.98	0.01324789	0.12059
Model 2: Backward	0.9249	-3408.73	0.0148149	0.12149	Model 2: Backward	0.9249	-3408.73	0.0148149	0.12149
Model 3: LASSO	0.8261	-1461.44	0.02818	0.16234	Model 3: LASSO	0.8261	-1461.444	0.02818	0.16234

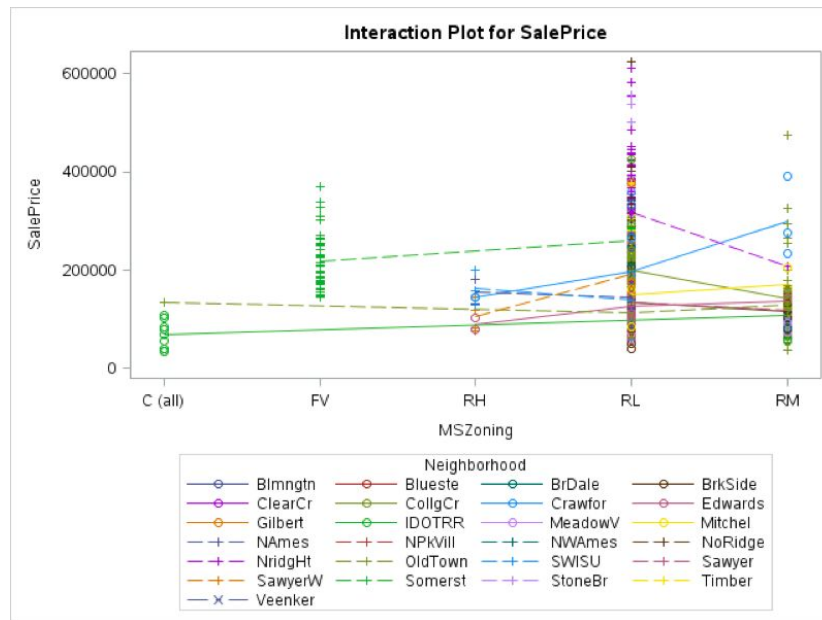
Conclusion

When ranking our models by both the test average square error as well as the Kaggle score, Model 1 (the custom model built off a backwards elimination) provides us with the lowest ASE (test) and Kaggle score. For this reason, the research team will recommend this model to be used for home price predictions in Ames. It is interesting to note that Model 2 (backward elimination) has the highest adjusted R^2 , but is not quite as predictive as Model 1 according to the other model details. Since we always need to keep in mind overfitting of the training data when we look at the adjusted R^2 value, we will proceed with our custom model as our final model. The model included obvious features such as overall quality and condition, the year the

home was built and remodeled, the total square footage, and the total number of baths. However, it also included less obvious factors, such as heating and air, the square footage of the unfinished basement, the screen porch area, and the wood deck area. Further research can be done on this project using more advanced regression techniques to come up with an even more predictive model.

2-Way ANOVA

In order to determine whether there is an interaction between the variables MSZoning and Neighborhood, we will run a 2-way analysis of variance. A visual inspection of the interaction plot between these two variables (below) provides us with evidence that an interaction term between MSZoning and Neighborhood may be appropriate.



An overall F-test reveals that there is a difference between groups in the non-additive model (p -value < 0.0001), suggesting that this model is appropriate (see table below).

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	40	4.5183687E12	112959217320	42.06	<.0001
Error	1148	3.0833196E12	2685818437.1		
Corrected Total	1188	7.6016883E12			

Further F-tests reveal that there is evidence that MSZoning is related to Neighborhood (p -value = 0.0012, see table below).

Source	DF	Type III SS	Mean Square	F Value	Pr > F
MSZoning	4	46123938417	11530984604	4.29	0.0019
Neighborhood	24	1.4920372E12	62168215245	23.15	<.0001
MSZoning*Neighborhood	12	87697322536	7308110211.3	2.72	0.0012

When analyzing the Bonferroni adjusted p -values, we can determine which zoning classifications are significantly related to which neighborhoods. Since there is such a large

number of significant differences, we will analyze one significant interaction and note that there are many others. There is strong evidence that the neighborhood Somerst with a zone classification of FV (Floating Village Residential) will have a higher mean estimated sale price than homes in the BrkSide neighborhood with a RL (Residential Low Density) zone classification (bonferroni adjusted p-value <0.0001). We are 95% confident that FV/Somerst homes will have a higher mean sale price between \$30,522 and \$136,302 than those in RL/BrkSide. In conclusion, there is evidence that the predicted mean sale prices of homes in various neighborhoods differs based on the zoning classifications (p-value = 0.0012). This information can be useful for real estate agents when selling homes and prospective home buyers to keep in mind when looking at different homes to purchase.

APPENDIX

Analysis Question 1: SAS Code

/*Data was cleaned in R. This cleaned data was imported into SAS for this analysis (see Analysis 2 for data cleaning code)*/

```
/*The following code was repeated for selection = forward, lasso, and lar to see which effects
were chosen the highest percentage of times with a model averaging technique*/
proc glmselect data=trainpro plots=all;
class MSZoning Street LotShape LandContour Utilities LotConfig LandSlope Neighborhood
Condition1 Condition2 BldgType HouseStyle RoofStyle RoofMatl Exterior1st Exterior2nd
MasVnrType ExterQual ExterCond Foundation BsmtQual BsmtCond BsmtExposure
BsmtFinType1 BsmtFinType2 Heating HeatingQC CentralAir Electrical KitchenQual Functional
FireplaceQu GarageType GarageFinish GarageQual GarageCond PavedDrive PoolArea
MiscVal SaleType SaleCondition;
model logSalePrice = Total_Sq_Footage MSSubClass LotFrontage LotArea OverallQual
OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF
TotalBsmtSF X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath
FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt
GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch X3SsnPorch
ScreenPorch PoolArea MiscVal MoSold YrSold MSZoning Street LotShape LandContour
Utilities LotConfig LandSlope Neighborhood Condition1 Condition2 BldgType HouseStyle
RoofStyle RoofMatl Exterior1st Exterior2nd MasVnrType ExterQual ExterCond Foundation
BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2 Heating HeatingQC
CentralAir Electrical KitchenQual Functional FireplaceQu GarageType GarageFinish
GarageQual GarageCond PavedDrive PoolArea MiscVal SaleType SaleCondition/
selection=forward(choose=cv stop=cv) cvmethod=random(10) stats=adjrsq;
modelaverage nsamples=1000 samplingmethod=URS(percent=100);
output out = results2 p=predict;
run;
```

/*Created custom model based on selection from above*/

```
proc glm data=trainpro plots=all;  
class centralair overallqual;  
model logsaleprice = Total_sq_footage overallqual yearbuilt yearremodadd garagecars  
centralair / solution clparm;  
run;
```

Analysis Question 2: R & SAS Code

Models 1 & 2: Backward Elimination and Custom Model (R Code)

```
#Multiple Linear Regression  
#install.packages("mice")  
#install.packages("lattice")  
#library(lattice)  
#library(mice)  
#install.packages("mlr") # Impute Package  
library(mlr) # Impute Library  
# To Increase the no of lines to print in console  
#options(max.print=1000000)  
  
#----Handelling NA-----  
# Handelling NAs, Use Impute Function  
  
train_d = read.csv("train.csv", stringsAsFactors = FALSE)  
test_d = read.csv("testCleaned.csv", stringsAsFactors = FALSE)  
  
train_imp <- impute(train_d, cols = list(LotFrontage = imputeMean()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)  
  
train_imp <- impute(train, cols = list(MSZoning = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)  
  
#?as.data.frame  
train_imp <- impute(train, cols = list(MasVnrType = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)  
  
train_imp <- impute(train, cols = list(MasVnrArea = imputeMean()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)  
  
train_imp <- impute(train, cols = list(BsmtQual = imputeMode()))
```

```
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
train_imp <- impute(train, cols = list(BsmtCond = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
train_imp <- impute(train, cols = list(BsmtExposure = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
train_imp <- impute(train, cols = list(BsmtFinType1 = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
train_imp <- impute(train, cols = list(BsmtFinType2 = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
train_imp <- impute(train, cols = list(Electrical = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
train_imp <- impute(train, cols = list(FireplaceQu = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
train_imp <- impute(train, cols = list(GarageType = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
train_imp <- impute(train, cols = list(GarageYrBlt = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
train_imp <- impute(train, cols = list(GarageFinish = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
train_imp <- impute(train, cols = list(GarageQual = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
train_imp <- impute(train, cols = list(GarageCond = imputeMode()))  
train <- as.data.frame(train_imp[1])  
colnames(train) <- colnames(train_d)
```

```
str(train)  
head(train, 10)
```

```
#----test----
```

```
test_imp <- impute(test_d, cols = list(LotFrontage = imputeMean()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(MSZoning = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
##?as.data.frame  
test_imp <- impute(test, cols = list(MasVnrType = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(MasVnrArea = imputeMean()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(BsmtQual = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(BsmtCond = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(BsmtExposure = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(BsmtFinType1 = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(BsmtFinType2 = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(Electrical = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(FireplaceQu = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(GarageType = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(GarageYrBlt = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(GarageFinish = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(GarageQual = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
test_imp <- impute(test, cols = list(GarageCond = imputeMode()))  
test <- as.data.frame(test_imp[1])  
colnames(test) <- colnames(test_d)
```

```
str(test)  
head(test, 10)  
#-----test impute end-----
```

```
#-----Handelling NA-----
```

```
#-----DATA PREP-----START-----
```

```
# Exploratory Data Analysis
```



```
#train = read.csv("train.csv")  
#test = read.csv("testCleaned.csv")
```

```
dim(train)  
dim(test)
```

```
str(train)  
#As we can see the training data contains 1460 records with 81 variables including an ID and  
the sale price. The test data has 1 fewer variable because  
#it should not contain the prediction target SalePrice:
```

```
str(test)  
#Now the factor levels should be identical across both data sets. Let's continue our exploration  
by looking at a summary of the training data.
```

```
summary(train)
```

```
#Inspecting the output above reveals that our data is not entirely clean. First, certain variables  
contain NA values which could  
#cause problems when we make predictive models later on. Second, the levels of some of the  
factor variables are not the same across the training  
#set and test set. For instance:
```

```
levels(train$MiscFeature)  
levels(test$MiscFeature)
```

```
#Differing factor levels could cause problems with predictive modeling later on so we need to  
resolve these issues before going further.  
#We can make sure the train and test sets have the same factor levels by loading each data set  
again without converting strings to factors,  
#combining them into one large data set, converting strings to factors for the combined data set  
and then separating them. Let's change any  
#NA values we find in the character data to a new level called "missing" while we're at it:
```

```
#train = read.csv("train.csv", stringsAsFactors = FALSE)  
#test = read.csv("testCleaned.csv", stringsAsFactors = FALSE)
```

```
# Remove the target variable not found in test set  
SalePrice = train$SalePrice  
train$SalePrice = NULL
```

```
# Remove extreme values from training set-----
```

```
# Combine data sets
full_data = rbind(train,test)

# Convert character columns to factor, filling NA values with "NoData"
for (col in colnames(full_data)){
  if (typeof(full_data[,col]) == "character"){
    new_col = full_data[,col]
    new_col[is.na(new_col)] = "NoData"
    full_data[col] = as.factor(new_col)
  }
}

str(full_data)

# Separate out our train and test sets
train = full_data[1:nrow(train),]
train$SalePrice = SalePrice
test = full_data[(nrow(train)+1):nrow(full_data),]

# Delete extreme values-----
train <- subset(train, GrLivArea < 4000)
train <- subset(train, BsmtFinSF1 < 5600)
train <- subset(train, MiscVal < 8000)

train <- subset(train, LotFrontage < 200)
#train <- subset(train, LotArea < 7000)
train <- subset(train, MasVnrArea < 1500)

# Fill remaining numeric columns with NA values with -1
train[is.na(train)] = -1
test[is.na(test)] = -1

# Add variable that combines above grade living area with basement sq footage
train$total_sq_footage = train$GrLivArea + train$TotalBsmtSF
test$total_sq_footage = test$GrLivArea + test$TotalBsmtSF

# Add variable that combines above ground and basement full and half baths
train$total_baths = train$BsmtFullBath + train$FullBath + (0.5 * (train$BsmtHalfBath +
train$HalfBath))
```

```
test$total_baths = test$BsmtFullBath + test$FullBath + (0.5 * (test$BsmtHalfBath +  
test$HalfBath))
```

```
# Add variable that combines Total SF  
#train$TotalSF = train$TotalBsmtSF + train$X1stFlrSF + train$X2ndFlrSF  
#test$TotalSF = test$TotalBsmtSF + test$X1stFlrSF + test$X2ndFlrSF
```

```
# Remove Id since it should have no value in prediction  
train$Id = NULL  
test$Id = NULL
```

```
#Handle NoData in Test  
test$KitchenQual[test$KitchenQual %in% c("NoData")] <- "TA"  
test$MSZoning[test$MSZoning %in% c("NoData")] <- "RL"  
test$Functional[test$Functional %in% c("NoData")] <- "Typ"
```

```
test_temp = read.csv("testCleaned.csv")
```

```
#We want to log certain variables based on assumptions-----
```

```
train$logsaleprice <- log(train$SalePrice+1)  
train$logBSMTFinSF1 <- log(train$BsmtFinSF1+1)  
train$logTotalBSMTSF <- log(train$TotalBsmtSF+1)  
train$logGrLivArea <- log(train$GrLivArea)
```

```
test$logBSMTFinSF1 <- log(test$BsmtFinSF1+1)  
test$logTotalBSMTSF <- log(test$TotalBsmtSF+1)  
test$logGrLivArea <- log(test$GrLivArea)
```

```
#write.csv(train , file = "train-pro.csv", row.names = FALSE)  
#write.csv(test , file = "test-pro.csv", row.names = FALSE)
```

```
# Splitting training data to train and test for internal rmse validation-----  
x <-sample(1:1189,200)  
train_test <- train[x,]  
train_train <- train[-x,]  
train_test$logprice <- train_test$logsaleprice  
train_test$logsaleprice = NULL  
train_test$SalePrice = NULL
```

```
#-----DATA PREP-----END-----
```

```
#-----ASE Test-----

# -----Final Models-----
# Remove Functional, MiscFeature doe test predictions , kaggle 0.12502, Condition2
model_b <- lm(logsaleprice ~ MSZoning + Street + LotConfig + LandSlope + Neighborhood +
Condition1 + BldgType + HouseStyle + OverallQual + OverallCond + YearBuilt +
YearRemodAdd + MasVnrArea + ExterQual + ExterCond + Foundation + BsmtQual +
BsmtFinType2 + BsmtFinSF2 + BsmtUnfSF + Heating + HeatingQC + CentralAir + X1stFlrSF +
X2ndFlrSF + BedroomAbvGr + TotRmsAbvGrd + Fireplaces + GarageArea + GarageCond +
WoodDeckSF + ScreenPorch + Fence + SaleCondition + total_sq_footage + total_baths +
logBSMTFinSF1 + logTotalBSMTSF + logGrLivArea , data = train_train)

# -----Best Model-----
# Custom from Backward Elim Remove Street, MasVnrArea, X1stFlrSF, X2ndFlrSF, Fence,
BsmtFinSF2, logBSMTFinSF1 , logTotalBSMTSF, BsmtFinType2 , BedroomAbvGr ,
TotRmsAbvGrd
# Best kaggle -> 0.12322 Custom from Backward Elim, removed Functional,Condition2
model_b <- lm(logsaleprice ~ MSZoning + LotConfig + LandSlope + Neighborhood + Condition1
+ BldgType + HouseStyle + OverallQual + OverallCond + YearBuilt + YearRemodAdd +
ExterQual + ExterCond + Foundation + BsmtQual + BsmtUnfSF + Heating + HeatingQC +
CentralAir + Fireplaces + GarageArea + GarageCond + WoodDeckSF + ScreenPorch +
SaleCondition + total_sq_footage + total_baths + logGrLivArea , data = train_train)

summary(model_b)

predictions <- predict(model_b, newdata = train_test)
#?rmse

rmse(train_testlogprice, predictions)
rmse_cv <- sqrt(sum(((train_testlogprice - predictions)^2)/200))
rmse_cv
#install.packages(Metrics)
#library(ModelMetrics)

ASE_Test <- sum(((train_testlogprice - predictions)^2)/200)
ASE_Test
exp(ASE_Test)

#-----ASE
Test-----

#-----olsrr Package-----Linear Model-----#
```

```
#install.packages("olsrr")
library(olsrr)

#Full Model

model <- lm(SalePrice ~ . , data = train)

#Residual vs Fitted Values Plot

ols_rvsp_plot(model)

#Residual Fit Spread Plot

ols_rfs_plot(model)

# Collinearity Diagnostics

ols_coll_diag(model)

#Stepwise AIC Backward Elimination Model -----Regression-----
b <- ols_stepaic_backward(model, details = TRUE)
b
# ?ols_stepaic_backward

# -----Final Models-----
# Remove Functional, MiscFeature doe test predictions , kaggle 0.12502
model_b <- lm(logsaleprice ~ MSZoning + Street + LotConfig + LandSlope + Neighborhood +
Condition1 + Condition2 + BldgType + HouseStyle + OverallQual + OverallCond + YearBuilt +
YearRemodAdd + MasVnrArea + ExterQual + ExterCond + Foundation + BsmtQual +
BsmtFinType2 + BsmtFinSF2 + BsmtUnfSF + Heating + HeatingQC + CentralAir + X1stFlrSF +
X2ndFlrSF + BedroomAbvGr + TotRmsAbvGrd + Fireplaces + GarageArea + GarageCond +
WoodDeckSF + ScreenPorch + Fence + SaleCondition + total_sq_footage + total_baths +
logBSMTFinSF1 + logTotalBSMTSF + logGrLivArea , data = train)

# -----Best Model-----
# Custom from Backward Elim Remove Street, MasVnrArea, X1stFlrSF, X2ndFlrSF, Fence,
BsmtFinSF2, logBSMTFinSF1 , logTotalBSMTSF, BsmtFinType2 , BedroomAbvGr ,
TotRmsAbvGrd
# Best kaggle -> 0.12322 Custom from Backward Elim, removed Functional
model_b <- lm(logsaleprice ~ MSZoning + LotConfig + LandSlope + Neighborhood + Condition1
+ Condition2 + BldgType + HouseStyle + OverallQual + OverallCond + YearBuilt +
YearRemodAdd + ExterQual + ExterCond + Foundation + BsmtQual + BsmtUnfSF + Heating +
```

```
HeatingQC + CentralAir + Fireplaces + GarageArea + GarageCond + WoodDeckSF +  
ScreenPorch + SaleCondition + total_sq_footage + total_baths + logGrLivArea , data = train)
```

```
summary(model_b)  
plot(model_b)  
ols_cooksd_barplot(model_b)  
#ols_cooksd_chart(model_b)  
ols_srsd_chart(model_b) # Residual  
ols_rsdlev_plot(model_b) # Leverage
```

```
predictions_b <- predict(model_b, newdata = test)  
#Exponentiate the predictions  
predictions_b <-exp(predictions_b)
```

```
predictions_b<- cbind(test_temp$Id, predictions_b)  
colnames(predictions_b) <- c("Id", "SalePrice")  
predictions_b <-as.data.frame(predictions_b)
```

```
predictions_b$SalePrice[predictions_b$SalePrice <= 0] <- 150000  
# Write CSV in R  
write.csv(predictions_b , file = "Result_b_25.csv", row.names = FALSE)
```

```
#-----End-----
```

```
#-----
```

Model 3: LASSO (SAS Code)

```
/*Data was cleaned in R. This cleaned data was imported into SAS for this analysis (see  
Analysis 2 for data cleaning code)*/
```

```
/*add saleprice column to test set*/  
data testpro;  
set testpro;  
SalePrice = .;  
logSalePrice = .;  
run;
```

```
/*generate random sample from training set to split into train and test set*/  
proc surveyselect data=trainpro samprate=0.50 out=Sample outall  
method=srs;  
run;
```



```
/*set training set & test set using random samples*/
```

```
data train;  
set Sample;  
where selected=1;  
run;
```

```
data test;  
set Sample;  
where selected=0;  
run;
```

```
data train;  
set train testpro;  
run;
```

```
/*lasso selection using external cross validation*/
```

```
proc glmselect data=train testdata=test plots(stepaxis=number)=(criterionpanel ASEPlot);  
class MSZoning Street LotShape LandContour Utilities LotConfig OverallQual LandSlope  
Neighborhood Condition1 Condition2 BldgType HouseStyle RoofStyle RoofMatl Exterior1st  
Exterior2nd MasVnrType ExterQual ExterCond Foundation BsmtQual BsmtCond BsmtExposure  
BsmtFinType1 BsmtFinType2 Heating HeatingQC CentralAir Electrical KitchenQual Functional  
FireplaceQu GarageType GarageFinish GarageQual GarageCond PavedDrive PoolArea  
MiscVal SaleType SaleCondition;  
model logSalePrice = Total_Sq_Footage MSSubClass LotFrontage LotArea OverallQual  
OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF  
TotalBsmtSF X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath  
FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt  
GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch X3SsnPorch  
ScreenPorch PoolArea MiscVal MoSold YrSold MSZoning Street LotShape LandContour  
Utilities LotConfig LandSlope Neighborhood Condition1 Condition2 BldgType HouseStyle  
RoofStyle RoofMatl Exterior1st Exterior2nd MasVnrType ExterQual ExterCond Foundation  
BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2 Heating HeatingQC  
CentralAir Electrical KitchenQual Functional FireplaceQu GarageType GarageFinish  
GarageQual GarageCond PavedDrive PoolArea MiscVal SaleType SaleCondition/  
selection=lasso(choose=cv stop=cv) / stats=bic;  
output out = results2 p=predict;  
run;
```

```
/*export data and un-log results*/
```

```
data results3;  
set results2;  
if predict > 0 then SalePrice = exp(Predict);  
if predict < 0 then SalePrice = 100000;
```

```
keep id SalePrice;  
where id > 1460;  
Run;
```

2-Way ANOVA: SAS Code

```
proc glm data=trainpro PLOTS=(DIAGNOSTICS RESIDUALS);  
class MSZoning Neighborhood;  
model SalePrice = MSZoning | Neighborhood / solution clparm;  
run;
```

```
proc mixed data=trainpro;  
class MSZoning Neighborhood;  
model SalePrice = MSZoning | Neighborhood;  
lsmeans MSZoning*Neighborhood / cl adjust=bon;  
run;
```