**Project 1**

**By: Lizzy Sterling and Scott Herford**

Table of Contents

[Introduction 1](#_Toc506679124)

[Data Description 1](#_Toc506679125)

[Exploratory Analysis 2](#_Toc506679126)

[Analysis Question 1 3](#_Toc506679127)

[The Problem 3](#_Toc506679128)

[Forward Model 3](#_Toc506679129)

[Backward Model 3](#_Toc506679130)

[Stepwise Model 4](#_Toc506679131)

[Table: 4](#_Toc506679132)

[Analysis Question 2 5](#_Toc506679133)

[The Problem 5](#_Toc506679134)

[LASSO (CV-AIC) 5](#_Toc506679135)

[LASSO (AIC-CV) 5](#_Toc506679136)

[ElasticNet (CV-AIC) 6](#_Toc506679137)

[Comparing Competing Models 7](#_Toc506679138)

[Conclusion 7](#_Toc506679139)

[Appendix 8](#_Toc506679140)

[Exploratory Analysis 8](#_Toc506679141)

[Analysis 1 8](#_Toc506679142)

# 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”(De Cock, D., Kaggle 2011). This project, investigates factors that influence sale prices of houses in Ames, Iowa. We will conduct two analyses for this project to examine the influence that neighborhood and square footage have on sale prices of homes as well as other factors that may be significant on the sale price of homes. This project could benefit realtors and homebuyers in Ames that may want to know what explanatory variables influence home sale prices.

# Data Description

This data is from the Kaggle competition “House Prices: Advanced Regression Techniques” (see more at <https://www.kaggle.com/c/house-prices-advanced-regression-techniques>). The data 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, and observations consisting of 2919 homes.

***Note: All figures and graphs have been placed in the appendix for readability and space conservation purposes. Please refer to the*** [***appendix***](#_Appendix) ***to see figures referenced in this project.***

# Exploratory Analysis

To begin this analysis, we first needed to check all of the variables to make sure that all of the data was entered in correctly. We found that LotFrontage was saved in the data as a character datatype rather than numeric, so we altered the datatype in order to run a proper analysis.

Next we took into consideration three main assumptions: **Normality**, **Equality of Variance**, **linearity,** and **Independence**. We started to look at this by comparing our response variable, SalePrice, side-by-side with every other quantifiable variable. Looking at [figure 1](#_Exploratory_Analysis), the data appeared to be very skewed and non-linear, so we attempted a log transformation on SalePrice, the results of which can be found in [figure 2](#_Exploratory_Analysis_1). We are going to keep the log transformation of SalePrice for the remainder of our analysis because it improved the residuals contained in the diagnostic plot output; thus, the log transformation of SalePrice decreased chances of violating any of the above mentioned assumptions for regression analysis.

Once we decided to log our response variable, we decided to look at regressions of each explanatory variable on logSalePrice as well as the output of residuals. We observed a few variables that appeared to be skewed due to outliers along with residuals that contained outliers with high leverage points and some that appeared to violate the assumption of linearity, so we decided to look into transformations. We found that the variables GrLivArea, LowQualFinSF, TotalBsmtSF, PoolArea, and MiscVal all had major issues with equality of variances, so we decided to log them and rerun the individual analysis with logSalePrice. We found that GrLivArea performed a lot better when logged as well as PoolArea and MiscVal. With the latter two variables we kept them logged, but keep in mind that both of these have very few data points so they might not meet other assumptions. Although we saw these improvements, we are not going to dwell too much here since logging the explanatory variables will not have too large of an effect on our models due to the fact that we are fitting so many variables into a linear model.

Because each model is unique, we are going to conduct further analysis as we create new models.

In terms of interpretation, we are going to look at a few key parameters. We are going to use our forward model as an example:

logSalePrice = .8455 + .0788\*OverallQual + .4588\*log(GrLivArea) …

If we were using the untransformed version of our response variable, the intercept indicates that as all other parameters are 0, estimated SalePrice = .8455. But Bbcause our response variable is transformed, we are going to focus on the variables rather than the intercept. Looking at OverallQual, as OverallQual increases by one unit, estimated SalePrice is estimated to increase by (e^.0788 = 1.0820) 8.2%, holding all other variables constant. Looking at GrLivArea, because it is also transformed we need to look at this a little differently. As GrLivArea doubles in square feet, estimated sales price is estimated to increase by $.14. You can do the same thing for all of the remaining variables. When looking at the confidence intervals, all you have to do is take each value and raise it over e. For example, we will look at OverallQual. From [figure 6](#_Forward_Selection_12) we see that the 95% confidence interval for the estimated value of OverallQual is [.070,.088]. Therefore we are going to say that as OverallQual increases by one unit, we can say with 95% confidence that SalePrice will increase by [e^.070,e^.088], or [7.25%, 9.20%]. Interpretations of the rest of the models will be conducted in the same manner.

# Analysis Question 1

## The Problem

We want to create a model that will help buyers, real estate agents, and contractors gain insight into what variables influence housing prices in Ames, Iowa. By doing this, we are going to perform three basic methods of model selection: **Forward**, **Backward**, and **Stepwise** selection.

## Forward Model

We began our model by including all qualitative and quantitative variables into a glmselect procedure. After running a procglmselect procedure with forward model selction a few times, we ended up deciding that this model ([figure 3](#_Forward_Selection_7)) was the best fit. To ensure that we do not have collinearity, we checked the VIFs and found that although Yearbuilt, logGrLivArea and OverallQual have a higher VIF than the rest, they are still pretty low with values around 2, so we kept both of these parameters in our model. Now we are going to check the assumptions on this model in particular:

**Normality:** Looking at the histogram ([figure 4](#_Forward_Selection_9)) the data appears to be normal, although the QQ-plot showsa slight curvature pattern. We are going to assume normality due the large number of data points and even if the data is not normal, we can assume that the central limit theorem will kick in.

**Equality of Variance:** Looking at the residual plot ([figure 5](#_Forward_Selection_10)) , there still seems to be a few outliers, but looking at the cook’s D and leverage, these points don’t seem to be anything to worry about, so we are going to assume equality of variance.

**Independence:** For the sake of this analysis, we are going to assume independent data.

Since all of our assumptions are met, we are going to look into the statistics of the model, which can be found in the [table](#_Table:). For information on the estimates, look to [figure 6](#_Forward_Selection_11).

## Backward Model

We began our model by including all qualitative and quantitative variables into a glmselect procedure. After running the backward selection procedure a few times, we ended up deciding that this model ([figure 7](#_Backward_Selection_3)) was the best fit. To account for collinearity, we checked variance inflation and found that non of the variables are collinear. Now we are going to check the assumptions on this model in particular:

**Normality:** From the histogram ([figure 8](#_Backward_Selection_3)) we can say that data appear to be relatively normal. Even though the QQ-plot shows evidence of a slight pattern, the sample size is large enough that we would assume the central limit theorem to kick in. For these reasons we are going to assume normality.

**Equality of Variance:** the residual plot ([figure 9](#_Backward_Selection_4)) does not look great – there are a few points in the bottom right that seem to be skewing the data. This is most likely due to the MiscVal variables, which we noted earlier were very few data points, so we are going to try logging MiscVal, which makes the residual plot more of a random cluster ([figure 10](#_Backward_Selection_5)), but when we try to fit the model again with the logged variable, it becomes more trouble than it’s worth because of the huge number of zeros. Therefore we are going to keep MiscVal in the model even though it skews the variance. Because we are only looking at one particular outlier, we are going to assume equality of variance.

**Independence:** For the sake of analysis, we are going to assume the data to be independent.

Since all of our assumptions are met, we are going to look into the statistics of the model, which can be found in the [table](#_Table:). For information on the estimates, look to [figure 11](#_Backward_Selection_6).

## Stepwise Model

We began our model by including all qualitative and quantitative variables into a glmselect procedure. After running the forward selection procedure a few times, we ended up deciding that this model ([figure 12](#_Stepwise_Selection_1)) was the best fit. We checked the variance inflation factors and although OverallQual, logGrLivArea, and YearBuilt all had higher VIF, we are not going to worry about collinearity since they are relatively small (all < 5). Now we are going to check the assumptions on this model in particular:

**Normality:** from the histogram ([figure 13](#_Stepwise_Selection_2)) the data is obviously normal. Again, the QQ-plot appears to be slightly skewed, but the sample size is large enough that the central limit theorem will kick in, so we can assume normality.

**Equality of Variance:** Although the residuals do not look terribly skewed ([figure 14](#_Stepwise_Selection_3)), there does seem to be an outlier or two that are significantly altering the model. We looked into this and found that TotalBsmtSF has a value that is much higher than its next highest value (look at this individual home, the basement area is the same as the General Living Area, so it is likely that this is a large, one-story home that was entered in as basement). For this reason, we are going to leave in the outlier and just run the analysis assuming equality of variance since the variances are otherwise relatively equal.

**Independence:** for the sake of analysis, we are going to assume independence of the data.

Since all of our assumptions are met, we are going to look into the statistics of the model, which can be found in the [table](#_Table:). For information on the estimates, look to [figure 15](#_Stepwise_Selection_4).

## Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | Adjusted R2 | AIC | CVPRESS | Kaggle Score |
| Forward | .8631 | -3166.34605 | 26.67304 | .17533 |
| Backward | .8701 | -3210.84885 | 25.83838 | 1.24987 |
| Stepwise | .8631 | -3166.34605 | 26.05902 | .17533 |

# Analysis Question 2

## The Problem

We would like to build the most predictive model for sale prices of homes in all of Ames, Iowa. We will use four model selection techniques to analyze the influential factors on sale prices of homes in Ames.

## LASSO (CV-AIC)

We began our model by including all qualitative and quantitative variables into a glmselect procedure utilizing shrinkage methods; specifically, we decided to include the feature selection method called LASSO along with model selection techniques beginning with the CVPRESS statistic and stopping with the Akaike Information Criterion(AIC) statistic. In our first model for question 2, we decided to use the Ames training set to observe which explanatory variables would be considered the most influential on our response sales prices of homes in Ames. Also, we performed a log transformation on the response SalesPrice; however, no transformations were conducted on any of the included predictor variables in the model.

With regard to assumptions on the dataset, we did identify several outliers from the following variables: LotArea, LotFrontage, BsmtFinSF1, GrLivArea, LowQualFinSF, TotalBsmt, PoolArea, and MiscVal. Several of the above variables contained high leverage points including LotArea, LogSalePrice, BsmtFinSF1, GrLivArea, and TotalBsmtSF, and LotArea as well as TotalBsmtSF revealed high Cook’s D values. Lastly, the following predictors seemed to violate the assumption of constant variance: GrLivArea, LowQualFinSF, TotalBsmt, PoolArea, and MiscVal.

As stated above, we attempted to conduct transformations on a few of the concerning variables; however, when we ran the models, we realized that several contained zero values where would could have changed the existing zero values to successfully run the model with log transformed predictors, but we chose not to follow through in the interest of time. In observing the residuals of the model, we determined that the distribution was skewed, patterns were observed in the predicted v. residual plot, and independence was assumed moving forward, so we can say that the model did violate assumptions of normality and constant variance; however, it’s possible to have massaged the data to meet the assumptions through appropriate transformations of concerning variables within the dataset.

For this particular model, the CVPRESS statistic of 32.22436 was shown to be the best selection criterion ([Figure 17](#LASSO CV-AIC)) in producing an output of 13 predictor variables with parameter estimates ([Figure 20](#LASSO CV-AIC)) on log(SalesPrice). As we’ll see later, this was not the best model out of the three ran for question 2 in terms of producing the best prediction accuracy.

## LASSO (AIC-CV)

We began our model by including all qualitative and quantitative variables into a glmselect procedure utilizing shrinkage methods; specifically, we decided to include the feature selection method called LASSO along with model selection techniques beginning with the Akaike Information Criterion(AIC) statistic and stopping with the CVPRESS statistic. In our second model for question 2, we decided to use the Ames training set to observe which explanatory variables would be considered the most influential on our response sales prices of homes in Ames. Also, we performed a log transformation on the response SalesPrice; however, no transformations were conducted on any of the included predictor variables in the model.

With regard to assumptions on the dataset, we did identify several outliers from the following variables: LotArea, LotFrontage, BsmtFinSF1, GrLivArea, LowQualFinSF, TotalBsmt, PoolArea, and MiscVal. Several of the above variables contained high leverage points including LotArea, LogSalePrice, BsmtFinSF1, GrLivArea, and TotalBsmtSF, and LotArea as well as TotalBsmtSF revealed high Cook’s D values. Lastly, the following predictors seemed to violate the assumption of constant variance: GrLivArea, LowQualFinSF, TotalBsmt, PoolArea, and MiscVal.

As stated above, we attempted to conduct transformations on a few of the concerning variables; however, when we ran the models, we realized that several contained zero values where would could have changed the existing zero values to successfully run the model with log transformed predictors, but we chose not to follow through in the interest of time. In observing the residuals of the model, we determined that the distribution was skewed, patterns were observed in the predicted v. residual plot, and independence was assumed moving forward, so we can say that the model did violate assumptions of normality and constant variance; however, it’s possible to have massaged the data to meet the assumptions through appropriate transformations of concerning variables within the dataset.

For this particular model, the AIC statistic of -2286.48849 was shown to be the best selection criterion ([Figure 22](#LASSO AIC-CV)) in producing an output of 24 predictor variables with parameter estimates ([Figure 25](#LASSO AIC-CV)) on log(SalesPrice). As we’ll see later, this was not the best model out of the three ran for question 2 in terms of producing the best prediction accuracy.

## ElasticNet (CV-AIC)

We began our model by including all qualitative and quantitative variables into a glmselect procedure utilizing shrinkage methods; specifically, we decided to include the feature selection method called ElasticNet along with model selection techniques beginning with the CVPRESS statistic and stopping with the Akaike Information Criterion(AIC) statistic. In our third model for question 2, we decided to use the Ames training and Ames test set to observe which explanatory variables would be considered the most influential on our response sales prices of homes in Ames. We decided to partition the data sets by train (60%) and test (40%). Also, we performed a log transformation on the response SalesPrice; however, no transformations were conducted on any of the included predictor variables in the model. Here, we examined VIF scores in the dataset, and no predictors with VIF scores >=10 were included in the model.

With regard to assumptions on the dataset, we did identify several outliers from the following variables: LotArea, LotFrontage, BsmtFinSF1, GrLivArea, LowQualFinSF, TotalBsmt, PoolArea, and MiscVal. Several of the above variables contained high leverage points including LotArea, LogSalePrice, BsmtFinSF1, GrLivArea, and TotalBsmtSF, and LotArea as well as TotalBsmtSF revealed high Cook’s D values. Lastly, the following predictors seemed to violate the assumption of constant variance: GrLivArea, LowQualFinSF, TotalBsmt, PoolArea, and MiscVal.

As stated above, we attempted to conduct transformations on a few of the concerning variables; however, when we ran the models, we realized that several contained zero values where would could have changed the existing zero values to successfully run the model with log transformed predictors, but we chose not to follow through in the interest of time. In observing the residuals of the model, we determined that the distribution was skewed, patterns were observed in the predicted v. residual plot, and independence was assumed moving forward, so we can say that the model did violate assumptions of normality and constant variance; however, it’s possible to have massaged the data to meet the assumptions through appropriate transformations of concerning variables within the dataset.

For this particular model, the CVPRESS statistic of 14.43177 was shown to be the best selection criterion ([Figure 26](#ElasticNet CV-AIC)) in producing an output of 24 predictor variables with parameter estimates ([Figure 29](#ElasticNet CV-AIC)) on log(SalesPrice). The third model was determined to be the best model out of the three ran for question 2 in terms of producing the best prediction accuracy using the feature selection method of ElasticNet and partitioning the Ames training set and test sets producing a Kaggle score of .19781.

## Comparing Competing Models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kaggle Competition Models | Adjusted R2 | AIC | CVPRESS | ASE | Kaggle Score |
| LASSO | .8123 | -2811.9501 | 32.22436 |  | 1.14645 |
| LASSO | .6985 | -2286.48849 | 35.41294 |  | 1.14645 |
| ElasticNet | .8709 | -1873.22861 | 14.43177 | (Train) .01999  (Test) 0.02972 | .19781 |

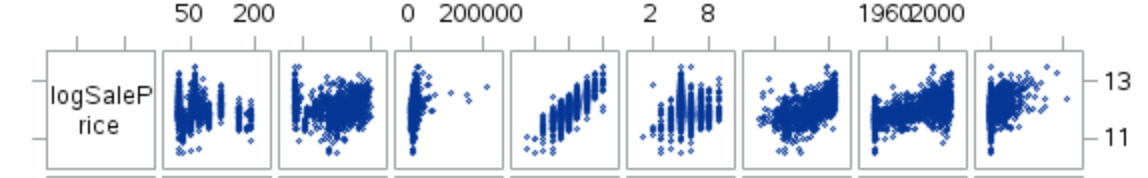
# Conclusion

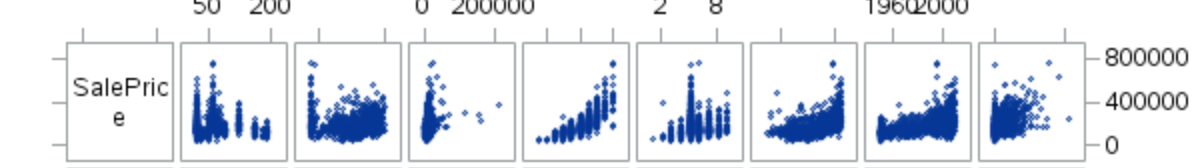
Given our relatively small knowledge of statistics and model fitting, we are proud of our ability to produce some not-terrible kaggle scores. That being said, there are a few things that we realized too late that we could have done to help our models. One of the biggest, and most time consuming, problems we came across dealt with transforming our explanatory variables. Two variables in particular, MiscVal and PoolArea, had a large number of zeros and only a few actual data points. This caused issues because the data was obviously skewed, but we could not do a transformation due to high number of zeros. We tried to fix this problem by changing all of the zeros to 1s to hopefully keep the same general idea of the data, but we decided that this was more trouble than would be worth it since the process of model fitting is so rigid. In the end, out of a total of six models ran for prediction accuracy on Ames home sale prices, the stepwise method was found to be the best fit for prediction and produced the best Kaggle score.

## 

## Appendix

### Exploratory Analysis

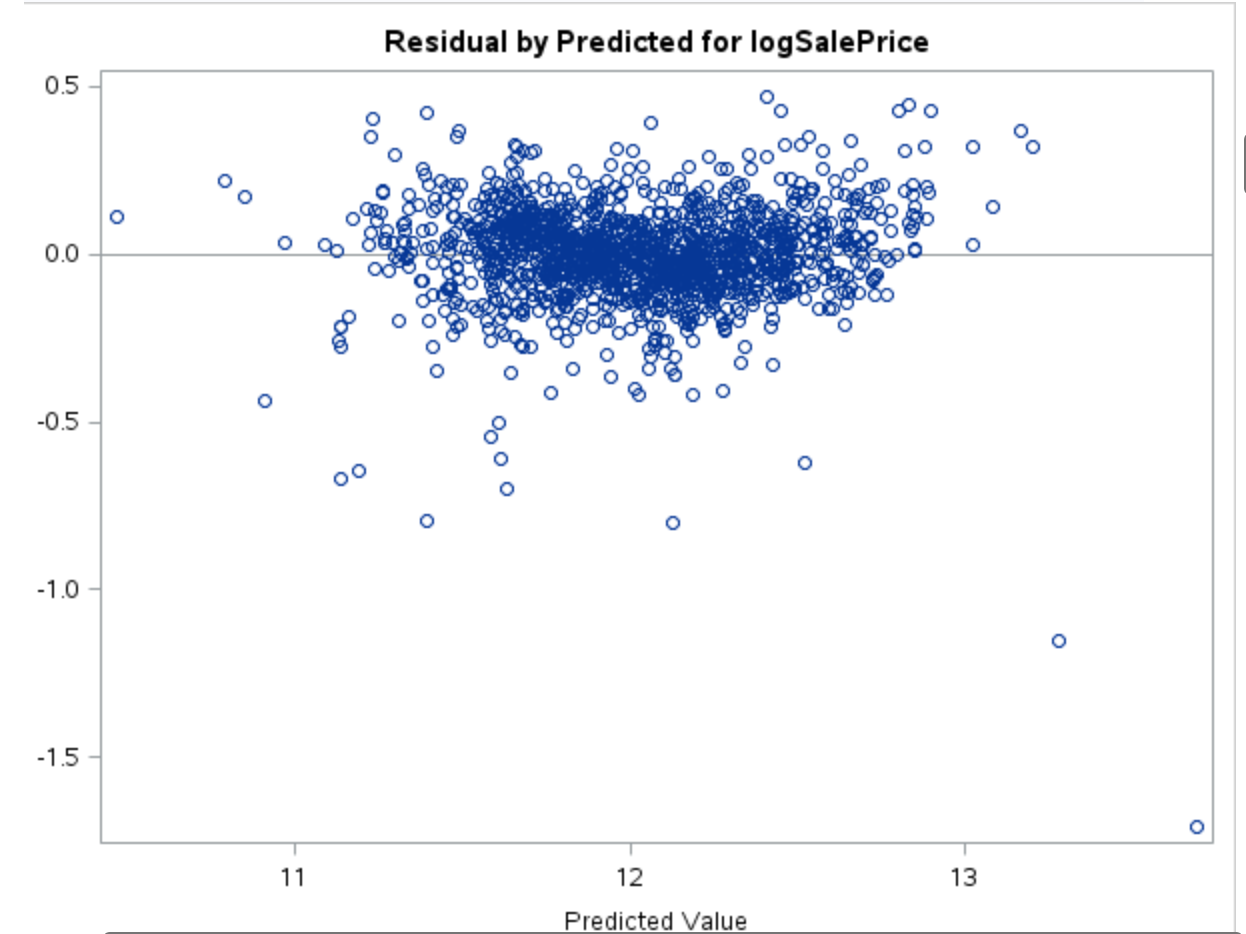
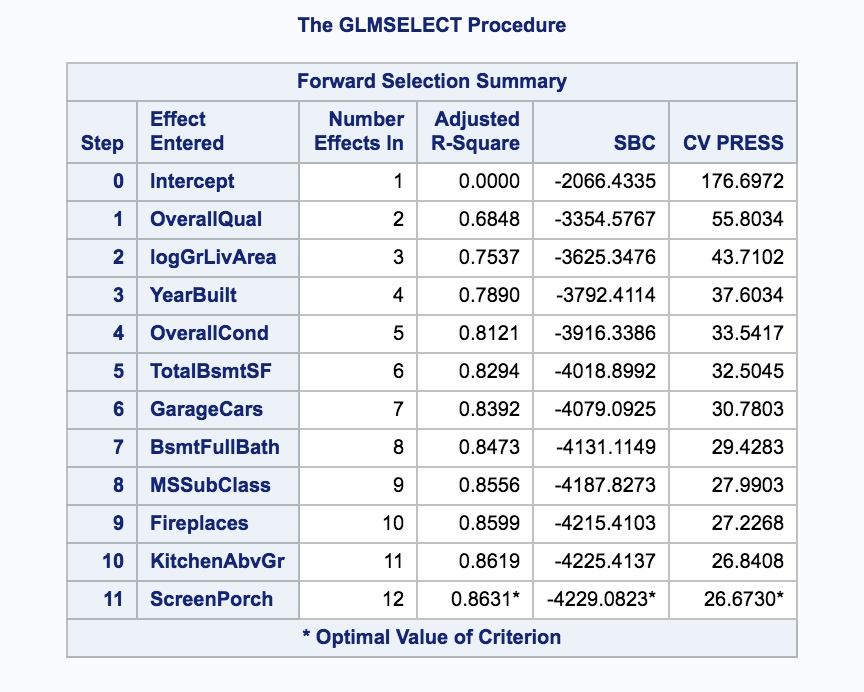
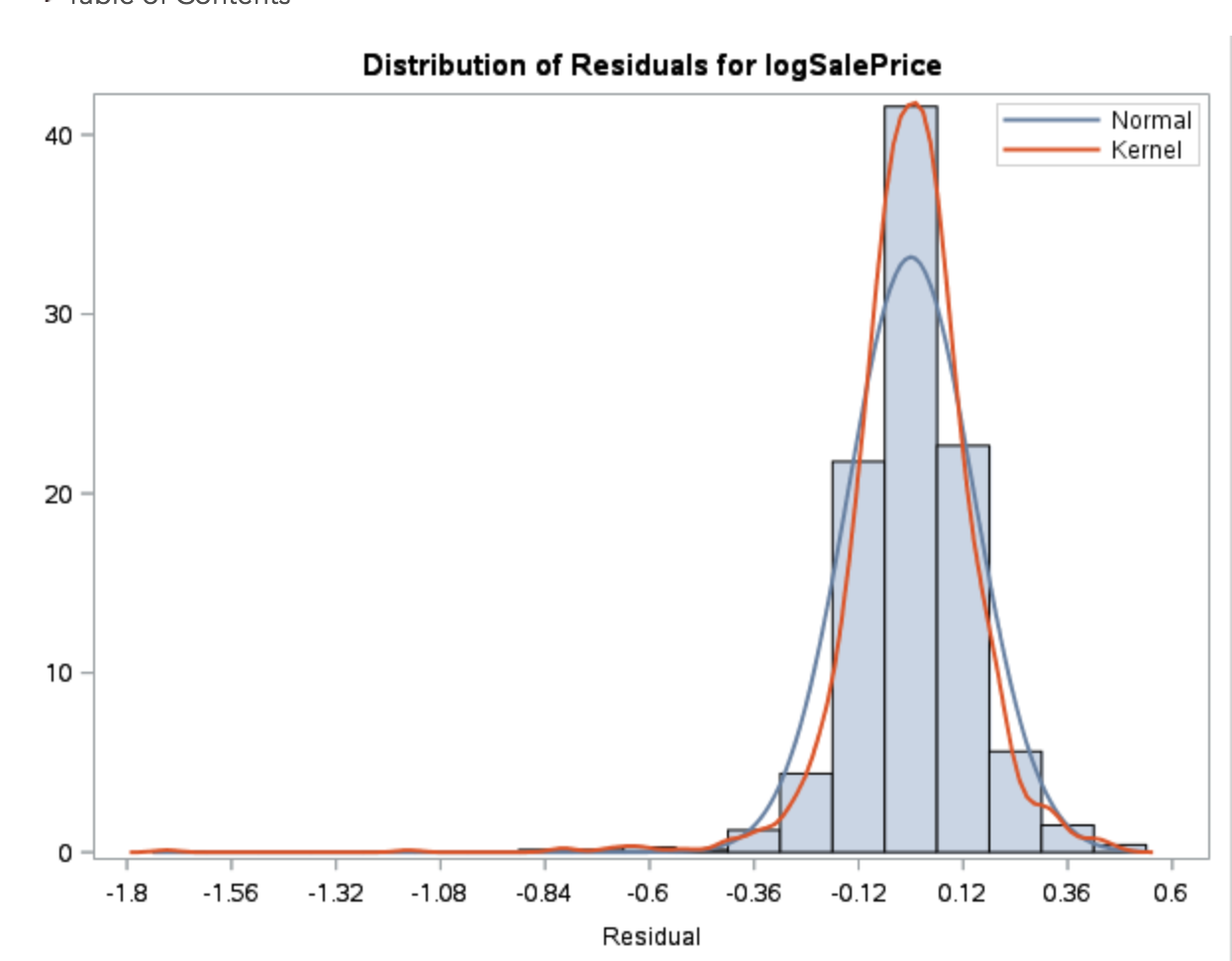
*Figure 1 Figure 2*

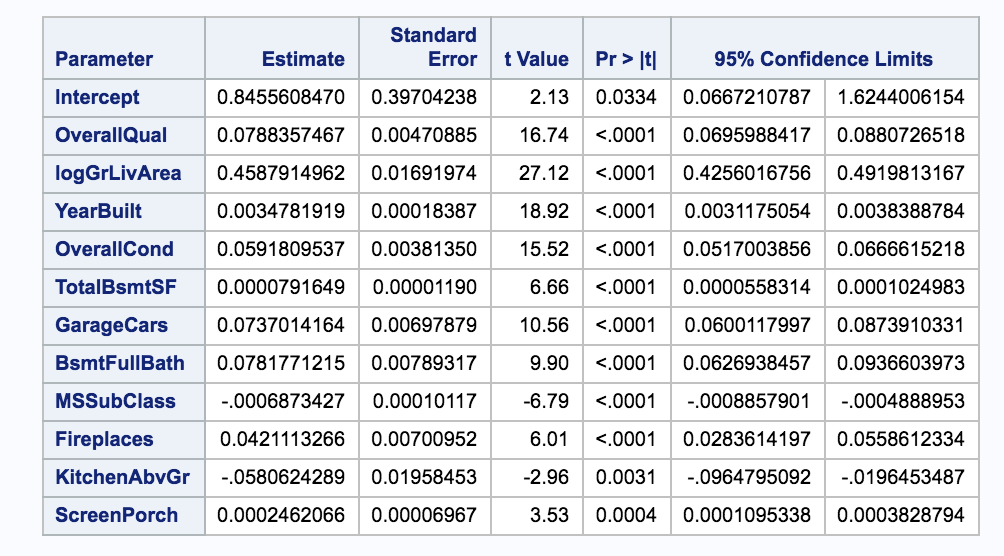


### Analysis 1

#### Forward Selection

*Figure 3 Figure 4 Figure 5*

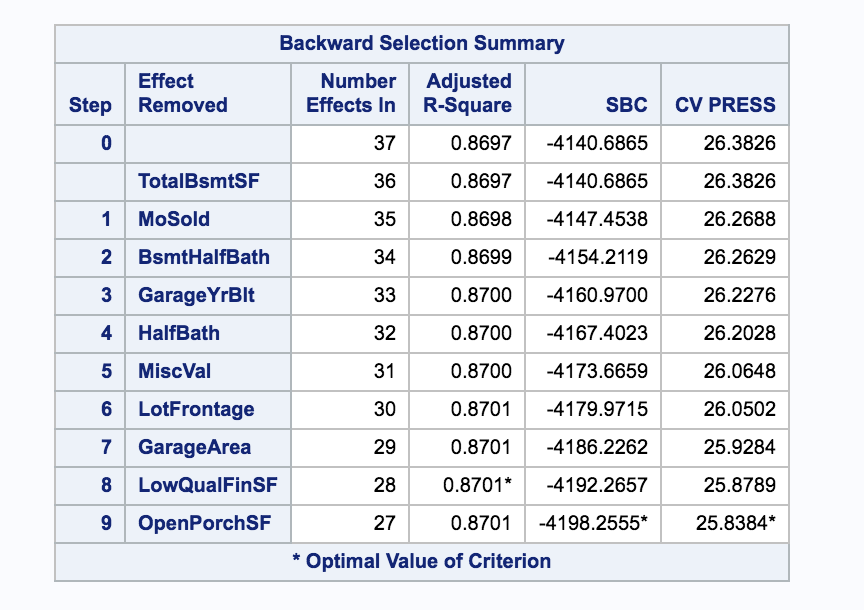
**

**

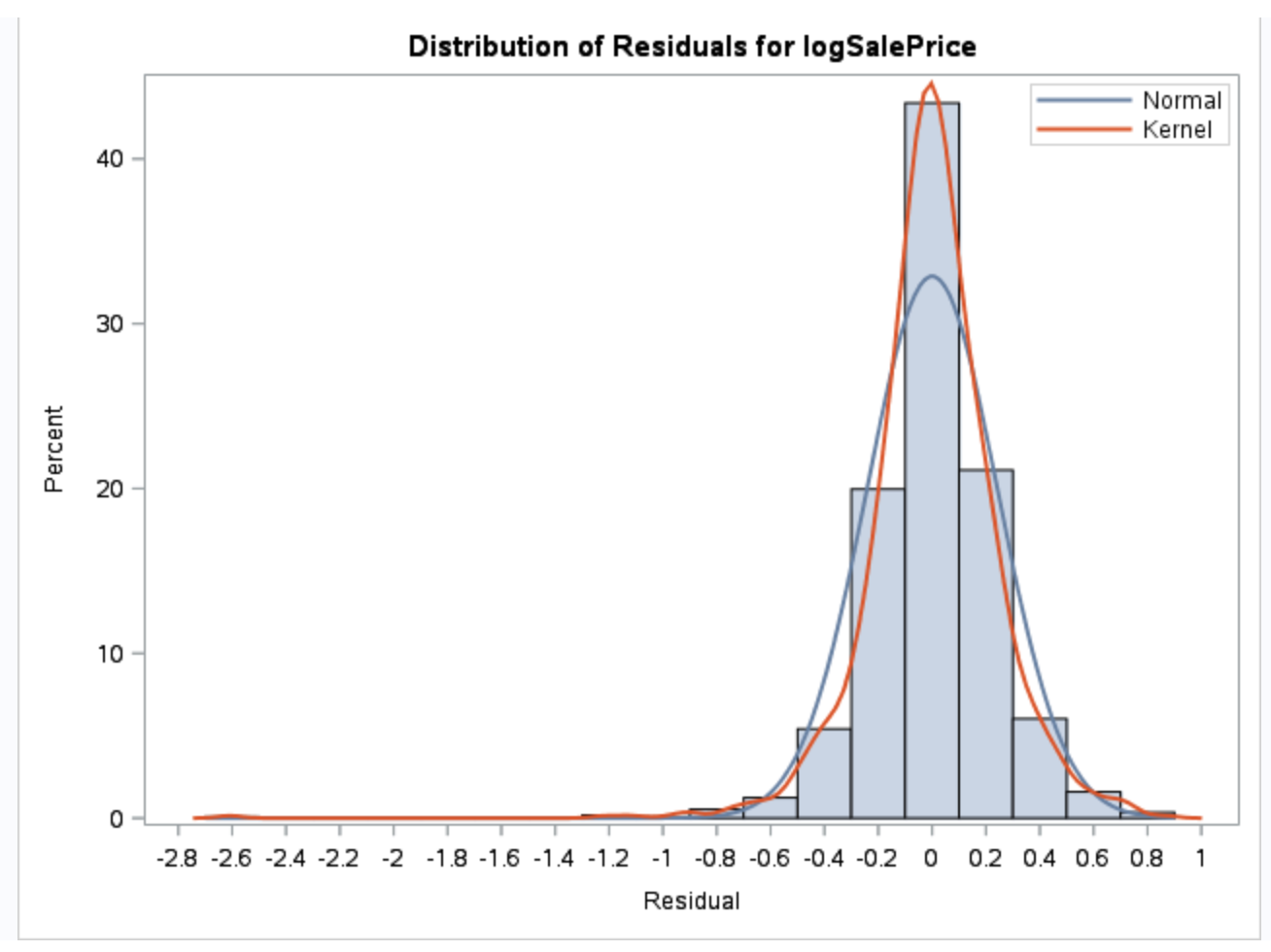
*Figure 6*

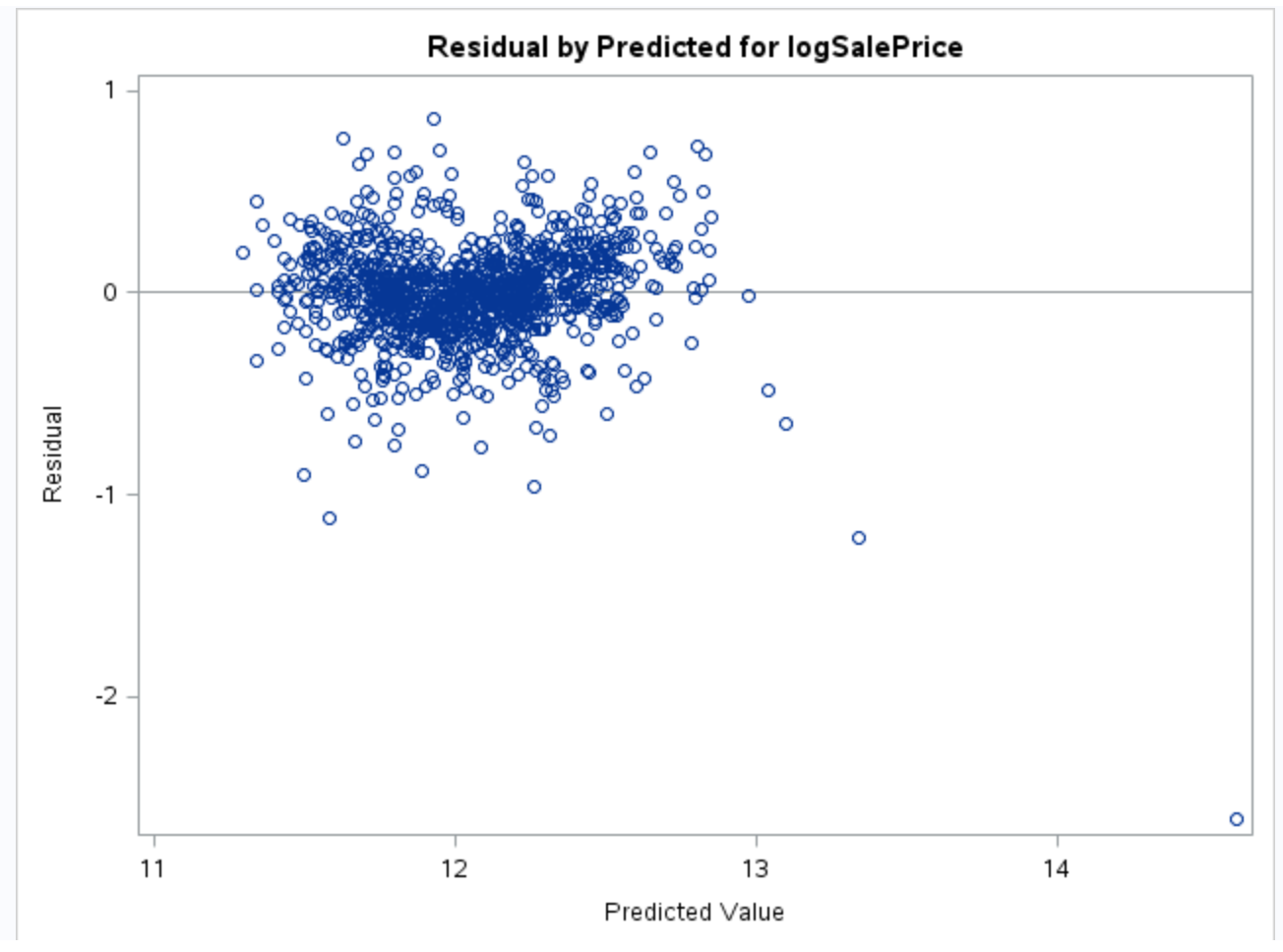
#### 

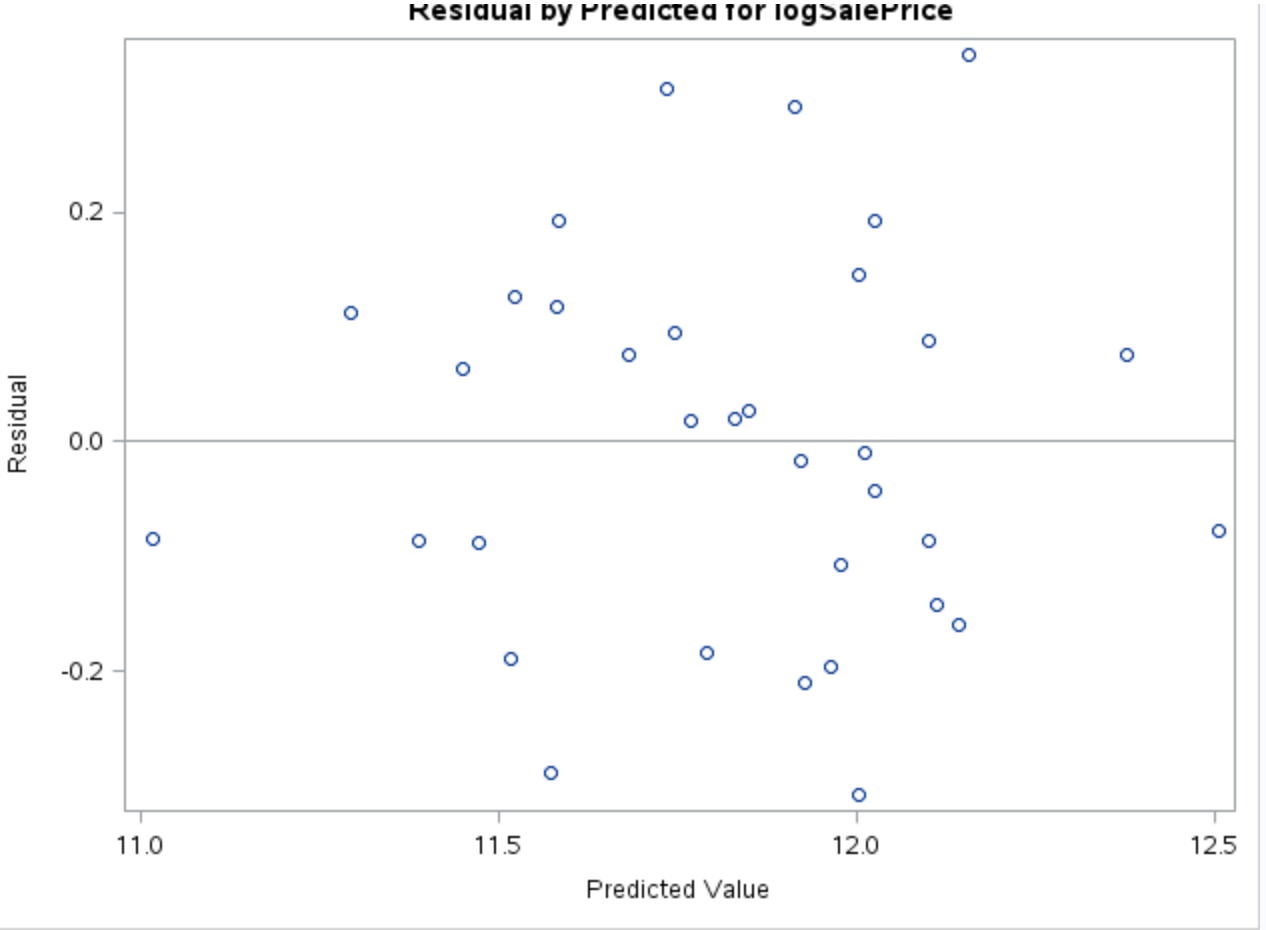
#### Backward Selection

*Figure 7*

*Figure 8 Figure 9*



**

*Figure 10 Figure 11*

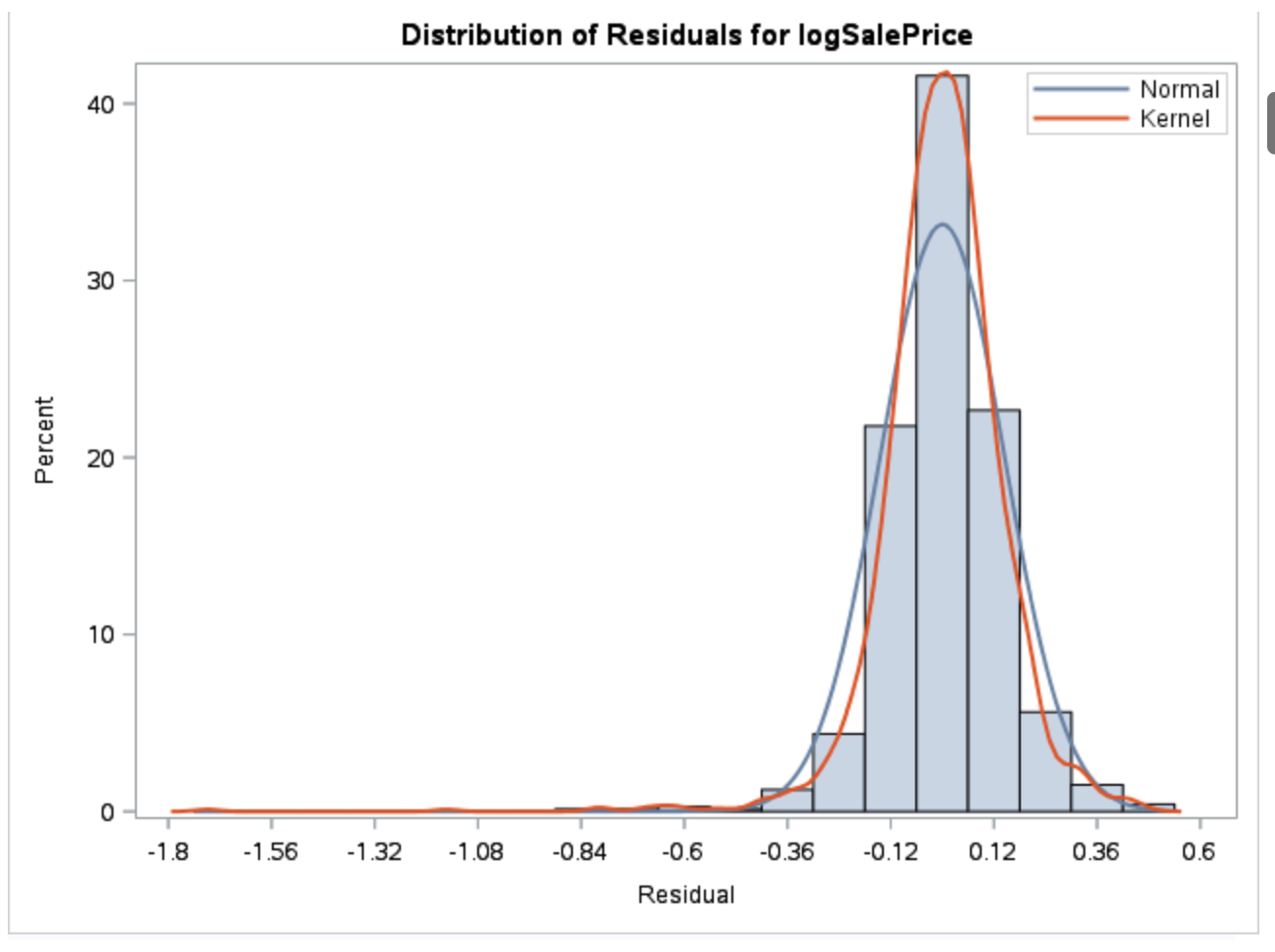
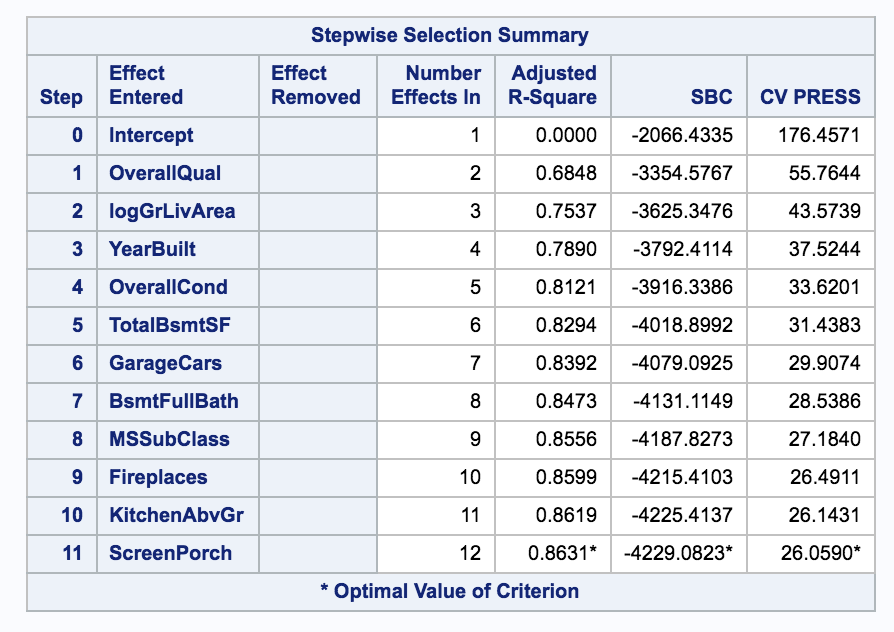
#### 

#### 

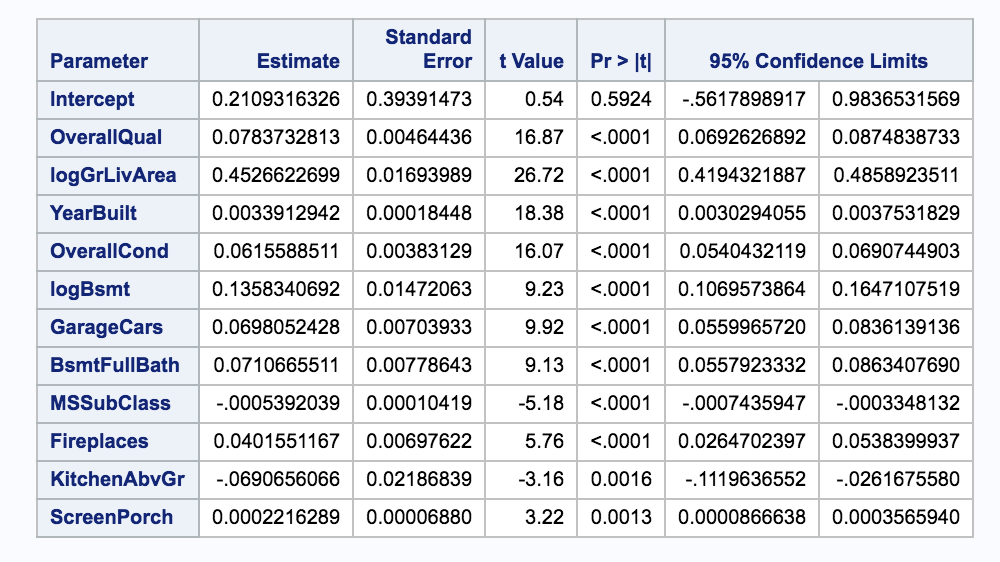
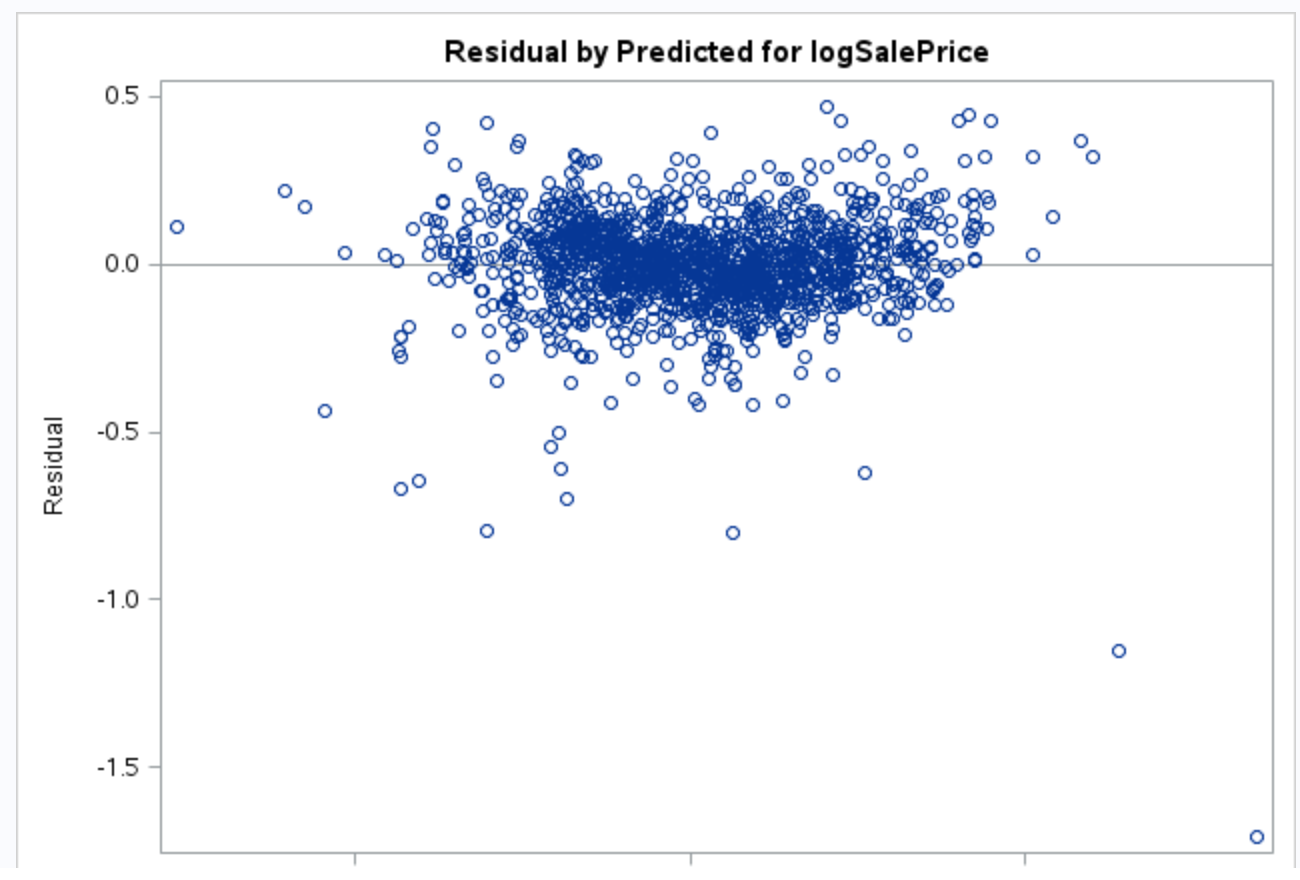
#### 

#### Stepwise Selection

*Figure 12 Figure 13*

**

*Figure 14 Figure 15*

**

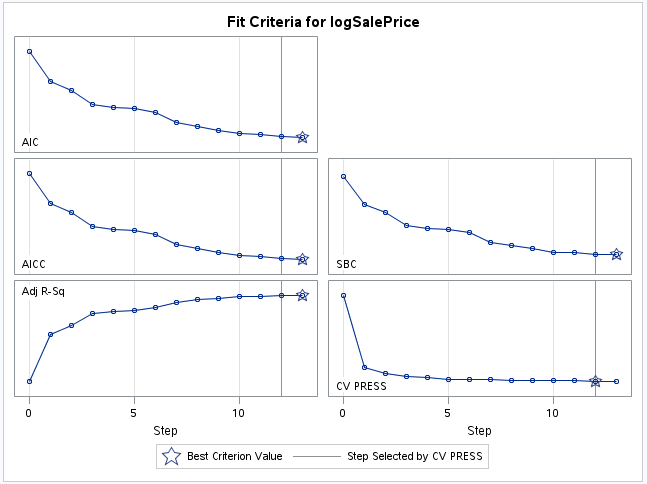
#### Analysis 2

#### LASSO CV-AIC

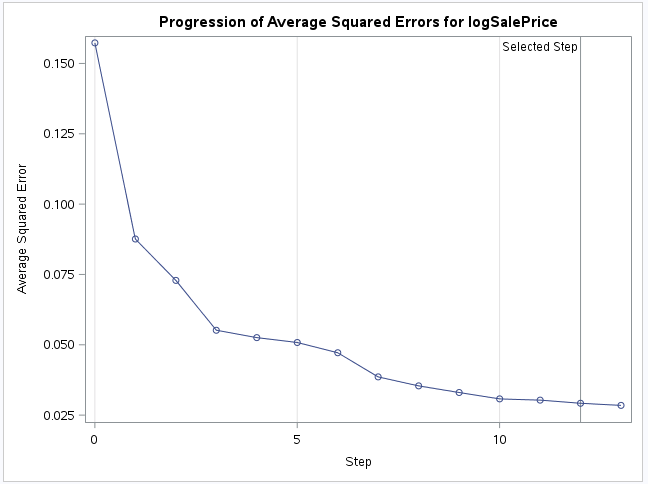
*Figure 16*



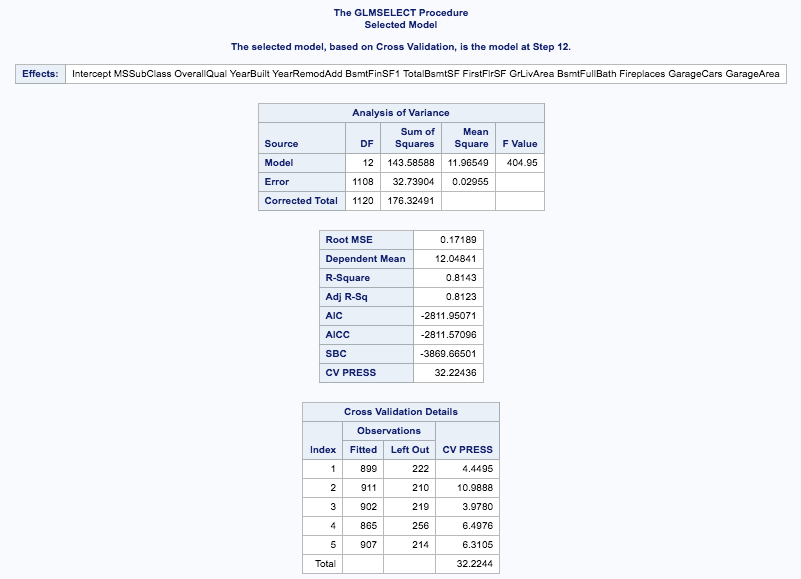
*Figure 17*



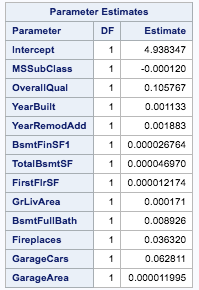
*Figure 18*



*Figure 19*

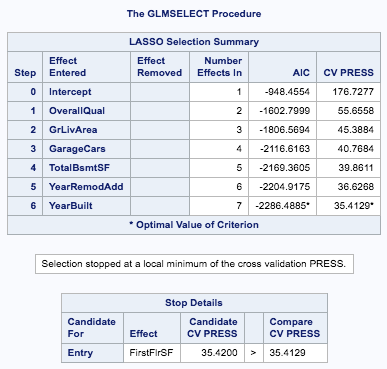


*Figure 20*

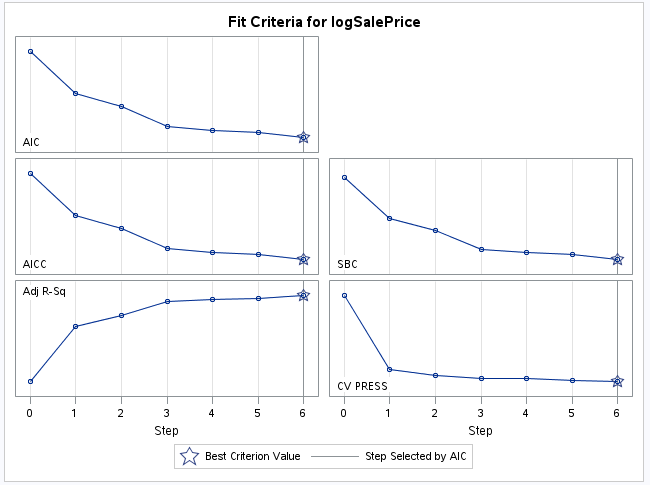


#### LASSO AIC-CV

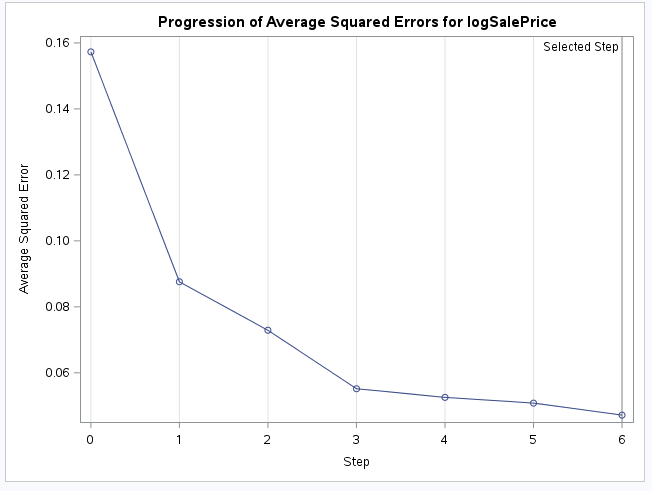
*Figure 21*



*Figure 22*



*Figure 23*



*Figure 24*

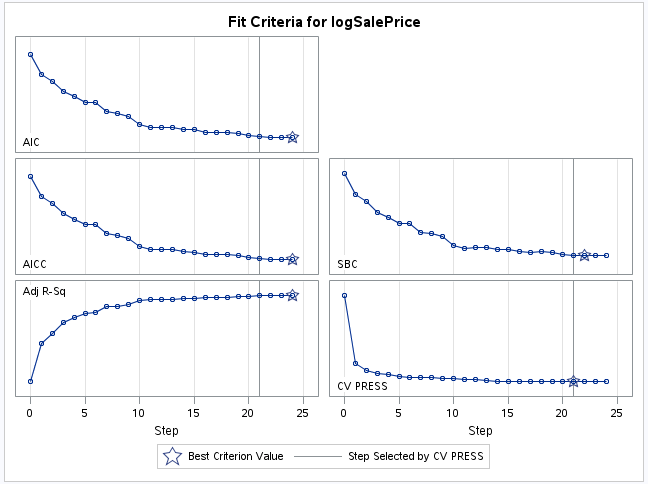


#### ElasticNet CV-AIC

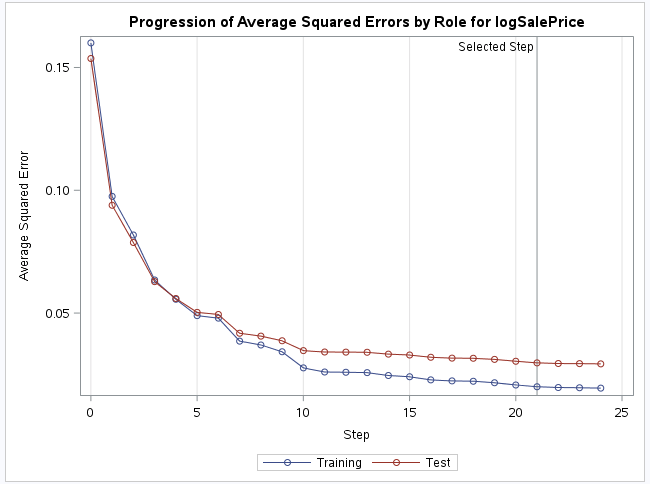
*Figure 25*



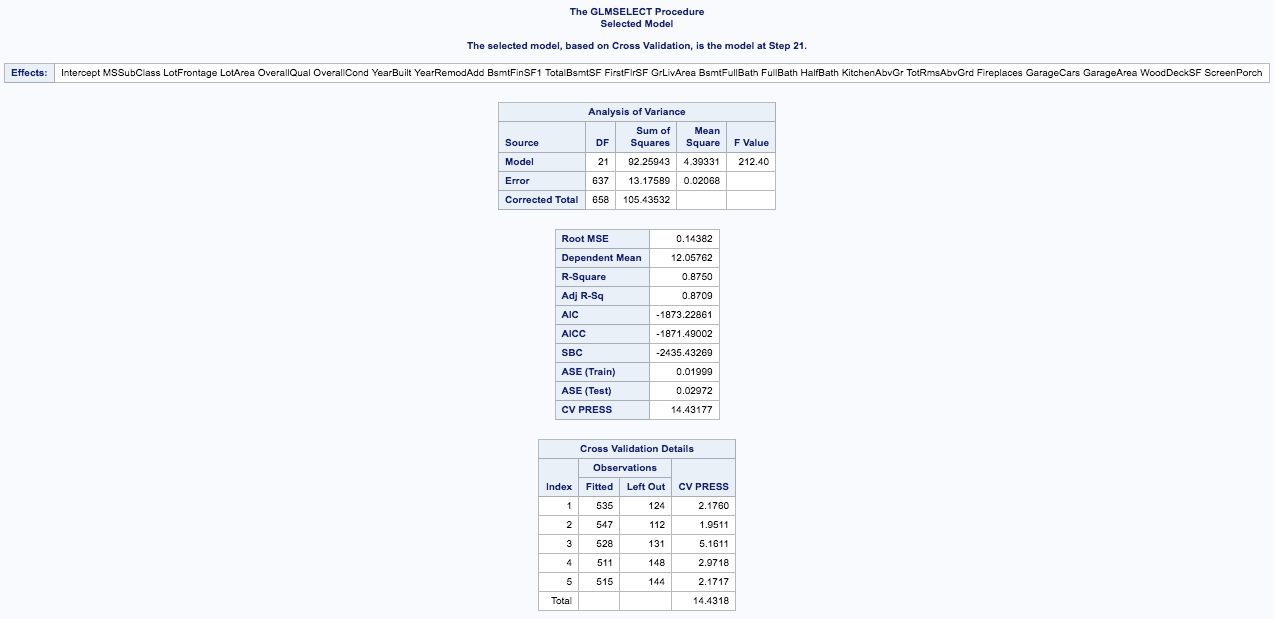
*Figure 26*



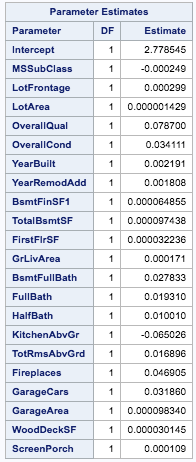
*Figure 27*



*Figure 28*



*Figure 29*



#### Analysis 1 Code:

|  |
| --- |
|  |
| /\*Test Data - altered column names\*/ |
|  | PROC IMPORT OUT = WORK.test |
|  | DATAFILE = '/home/lsterling0/Imports/test.csv' |
|  | DBMS=CSV REPLACE; |
|  | GETNAMES=YES; |
|  | RUN; |
|  |  |
|  | /\*Train Data - altered column names\*/ |
|  | PROC IMPORT OUT = WORK.train |
|  | DATAFILE = '/home/lsterling0/Imports/train.csv' |
|  | DBMS=CSV REPLACE; |
|  | GETNAMES=YES; |
|  | RUN; |
|  |  |
|  |  |
|  | /\*We need to add a SalePrice column to the test set\*/ |
|  | data test; |
|  | set test; |
|  | SalePrice = .; |
|  | run; |
|  |  |
|  | /\*We need to look into outliers of the data\*/ |
|  |  |
|  | /\*LotFrontage is saved as a character when it should be saved as numeric\*/ |
|  | \*We also need to combine test and train into one dataset - train2; |
|  | data Train2; |
|  | set Train test; \*combining the sets together; |
|  | LotFrontageNum = input(LotFrontage, 8.); \*Creating a new numeric column with values from LotFrontage; |
|  | drop LotFrontage; \*deleting the character column LotFrontage; |
|  | rename LotFrontageNum=LotFrontage; \*renaming the numeric column to match the original column name; |
|  | run; |
|  |  |
|  | /\*Let's go ahead and log SalePrice\*/ |
|  | data Train2; |
|  | set Train2; |
|  | logSalePrice = log(SalePrice); |
|  | logGrLivArea = log(GrLivArea); |
|  | logLowQualFinSF = log(LowQualFinSF); \*Does not help; |
|  | SRTotalBsmtSF = sqrt(TotalBsmtSF); \*we tried both log and sqrt - neither helped with the cluster; |
|  | logPoolArea = log(PoolArea); |
|  | logMiscVal = log(MiscVal); |
|  | run; |
|  | \*From here on out we will be using train2; |
|  |  |
|  | /\*EXPLORATORY ANALYSIS\*/ |
|  |  |
|  | /\*Here I separated all of the variables, along with SalePrice, into 4 different matrices because running them all together got really small\*/ |
|  |  |
|  | \*MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea; |
|  | proc sgscatter data=train2; |
|  | matrix logSalePrice MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea; |
|  | run; |
|  | \*LotArea - potential outliers |
|  | lotFrontage - data looks off; |
|  | proc glm data = train2 plots=all; |
|  | model logSalePrice = LotArea; |
|  | run; \*definite outlier(s); |
|  |  |
|  | proc glm data = train2 plots=all; |
|  | model logSalePrice = LotFrontage; |
|  | run; \*looks okay; |
|  |  |
|  | \*BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF GrLivArea; |
|  | proc sgscatter data=train2; |
|  | matrix logSalePrice BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF GrLivArea; |
|  | run; |
|  |  |
|  | proc glm data = train2 plots=all; |
|  | model logSalePrice = BsmtFinSF1; |
|  | run; |
|  |  |
|  | proc glm data = train2 plots=all; |
|  | model logSalePrice = logGrLivArea; |
|  | run; |
|  |  |
|  | proc glm data = train2 plots=all; |
|  | model logSalePrice = logLowQualFinSF; |
|  | run; |
|  |  |
|  | proc glm data = train2 plots=all; |
|  | model logSalePrice = SRTotalBsmtSF; |
|  | run; |
|  |  |
|  | proc glm data = train2 plots=all; |
|  | model logSalePrice = logPoolArea; |
|  | run; |
|  |  |
|  | proc glm data = train2 plots=all; |
|  | model logSalePrice = logMiscVal; |
|  | run; |
|  |  |
|  | \*BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces; |
|  | proc sgscatter data = train2; |
|  | matrix logSalePrice BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces; |
|  | run; |
|  |  |
|  | \*GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold; |
|  | proc sgscatter data=train2; |
|  | matrix logSalePrice GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold; |
|  | run; |
|  |  |
|  |  |
|  | /\*Now that we have logged values that seemed to be off, we can start looking for any outliers\*/ |
|  | \*I am not sure that proc glm is the best way to look at outliers when there are so many class variables; |
|  | proc glm data=Train2 plots = all; |
|  | class ; \*should we include all?; |
|  | \*model logSalePrice ; \*Use this if logSalePrice looks better; |
|  | model SalePrice = /solution; \*insert variable being considered; |
|  | run; |
|  |  |
|  | /\*FORWARD SELECTION\*/ |
|  | \*Not including any logged variables; |
|  | proc glmselect data=train2 plots=all; |
|  | class MSZoning Street Alley 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 PoolQC Fence MiscFeature SaleType SaleCondition; |
|  | model logSalePrice = MSSubClass logLotFrontage logLotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF logGrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch logPoolArea logMiscVal MoSold YrSold / selection=Forward(stop=CV) cvmethod=random(5) stats=adjrsq; |
|  | output out = results2 p=predict; |
|  | run; |
|  |  |
|  |  |
|  | /\*Check assumptions for given model\*/ |
|  | \*Scatterplot Matrix; |
|  | proc sgscatter data=train2; |
|  | matrix ; |
|  | run; |
|  |  |
|  | \*Assumption plots and VIF; |
|  | proc reg data=train2 plots=all; |
|  | model / VIF; |
|  | run; |
|  |  |
|  | /\*Get confidence intervals for final model \*/ |
|  | proc glm data = train2 plots = all; |
|  | class ; |
|  | model /solution clparm; |
|  | run; |
|  |  |
|  | /\*BACKWARD SELECTION\*/ |
|  | \*Not including any logged variables; |
|  | proc glmselect data=train2 plots=all; |
|  | class MSZoning Street Alley 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 PoolQC Fence MiscFeature SaleType SaleCondition; |
|  | model logSalePrice = MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold / selection=Backward(stop=CV) cvmethod=random(5) stats=adjrsq; |
|  | output out = results2 p=predict; |
|  | run; |
|  | /\*Check assumptions for given model\*/ |
|  | \*Scatterplot Matrix; |
|  | proc sgscatter data=train2; |
|  | matrix ; |
|  | run; |
|  |  |
|  | \*Assumption plots and VIF; |
|  | proc reg data=train2 plots=all; |
|  | model / VIF; |
|  | run; |
|  |  |
|  | /\*Get confidence intervals for final model\*/ |
|  | proc glm data = train2 plots = all; |
|  | class ; |
|  | model /solution clparm; |
|  | run; |
|  |  |
|  | /\*STEPWISE SELECTION\*/ |
|  | \*Not including any logged variables; |
|  | proc glmselect data=train2 plots=all; |
|  | class MSZoning Street Alley 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 PoolQC Fence MiscFeature SaleType SaleCondition; |
|  | model SalePrice = MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold / selection=Stepwise(stop=CV) cvmethod=random(5) stats=adjrsq; |
|  | output out = results2 p=predict; |
|  | run; |
|  |  |
|  | /\*Check assumptions for given model\*/ |
|  | \*Scatterplot Matrix; |
|  | proc sgscatter data=train2; |
|  | matrix ; |
|  | run; |
|  |  |
|  | \*Assumption plots and VIF; |
|  | proc reg data=train2 plots=all; |
|  | model / VIF; |
|  | run; |
|  |  |
|  | /\*Get confidence intervals for final model\*/ |
|  | proc glm data = train2 plots = all; |
|  | class ; |
|  | model /solution clparm; |
|  | run; |
|  |  |
|  | /\*Use the following code to output the proper file to submi in kaggle\*/ |
|  | data results3; |
|  | set results2; |
|  | if predict > 0 then SalePrice = exp(Predict); \*uncomment if SalePrice is logged; |
|  | \*if predict > 0 then SalePrice = Predict; |
|  | if predict < 0 then SalePrice = 10000; \*This gets rid of any negative values; |
|  | keep id SalePrice; \*we only want to keep ID and SalePrice; |
|  | where id > 1460; \*We only want to include SalePrice from the empty Test dataset; |
|  | run; |
|  |  |
|  | proc export data = results3 outfile = \_dataout dbms = csv replace; |
|  | run; |
|  |  |

#### Analysis 2 Code:

/\*Test Data - altered column names\*/

PROC IMPORT OUT = ames\_test

DATAFILE = '/home/jherford0/sasuser.v94/Ames\_Test\_Revised.csv'

DBMS=CSV REPLACE;

GETNAMES=YES;

RUN;

/\*Train Data - altered column names\*/

PROC IMPORT OUT = ames\_train

DATAFILE = '/home/jherford0/sasuser.v94/Ames\_Train\_Revised.csv'

DBMS=CSV REPLACE;

GETNAMES=YES;

RUN;

/\*We need to add a SalePrice column to the test set\*/

data ames\_test;

set ames\_test;

logSalePrice = .;

run;

/\*We need to look into outliers of the data\*/

/\*LotFrontage is saved as a character when it should be saved as numeric\*/

\*We also need to combine test and train into one dataset - train2;

data ames\_train2;

set ames\_train ames\_test; \*combining the sets together;

LotFrontageNum = input(LotFrontage, 8.); \*Creating a new numeric column with values from LotFrontage;

drop LotFrontage; \*deleting the character column LotFrontage;

rename LotFrontageNum=LotFrontage; \*renaming the numeric column to match the original column name;

run;

/\*LotFrontage is saved as a character when it should be saved as numeric\*/

\*We also need to combine test and train into one dataset - test2;

data ames\_test2;

set ames\_train ames\_test; \*combining the sets together;

LotFrontageNum = input(LotFrontage, 8.); \*Creating a new numeric column with values from LotFrontage;

drop LotFrontage; \*deleting the character column LotFrontage;

rename LotFrontageNum=LotFrontage; \*renaming the numeric column to match the original column name;

run;

/\*Let's go ahead and log SalePrice\*/

data ames\_train2;

set ames\_train2;

logSalePrice = log(SalePrice);

logGrLivArea = log(GrLivArea);

logPoolArea = log(PoolArea);

logMiscVal = log(MiscVal);

run;

\*From here on out we will be using train2;

data ames\_test2;

set ames\_test2;

logSalePrice = log(SalePrice);

logGrLivArea = log(GrLivArea);

logPoolArea = log(PoolArea);

logMiscVal = log(MiscVal);

run;

/\*EXPLORATORY ANALYSIS\*/

/\*Here I separated all of the variables, along with SalePrice, into 4 different matrices because running them all together got really small\*/

\*MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea;

proc sgscatter data=ames\_train2;

matrix logSalePrice MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea;

run;

\*LotArea - potential outliers

lotFrontage - data looks off;

proc glm data = ames\_train2 plots=all;

model logSalePrice = LotArea;

run; \*definite outlier(s);

proc glm data = ames\_train2 plots=all;

model logSalePrice = LotFrontage;

run; \*looks okay;

\*BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF GrLivArea;

proc sgscatter data=ames\_train2;

matrix logSalePrice BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF GrLivArea;

run;

proc glm data = ames\_train2 plots=all;

model logSalePrice = BsmtFinSF1;

run;

proc glm data = ames\_train2 plots=all;

model logSalePrice = logGrLivArea;

run;

proc glm data = ames\_train2 plots=all;

model logSalePrice = logLowQualFinSF;

run;

proc glm data = ames\_train2 plots=all;

model logSalePrice = SRTotalBsmtSF;

run;

proc glm data = ames\_train2 plots=all;

model logSalePrice = logPoolArea;

run;

proc glm data = ames\_train2 plots=all;

model logSalePrice = logMiscVal;

run;

\*BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces;

proc sgscatter data = ames\_train2;

matrix logSalePrice BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces;

run;

\*GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold;

proc sgscatter data=ames\_train2;

matrix logSalePrice GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold;

run;

/\*Now that we have logged values that seemed to be off, we can start looking for any outliers\*/

\*I am not sure that proc glm is the best way to look at outliers when there are so many class variables;

proc glm data=ames\_Train2 plots = all;

class ; \*should we include all?;

\*model logSalePrice ; \*Use this if logSalePrice looks better;

model SalePrice = /solution; \*insert variable being considered;

run;

\*Assumption plots and VIF;

\*No predictors in the model revealed VIF scores of >=10;

proc reg data=ames\_train2 plots=all;

model logSalePrice = MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold / VIF;

run;

\*Model 1; /\*No predictors contained log transformations on Ames training dataset\*/ /\*LASSO CV-AIC\*/

ods graphics on;

proc glmselect data=ames\_train2

seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

class MSZoning Street Alley 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 PoolQC Fence MiscFeature SaleType SaleCondition;

model logSalePrice = MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold / selection=LASSO(choose=CV stop=AIC) CVDETAILS ;

output out= results7 p=predict;

run;

quit;

ods graphics off;

\*Model 2; /\*No predictors contained log transformations on Ames training dataset\*/ /\*LASSO AIC-CV\*/

ods graphics on;

proc glmselect data=ames\_train2

seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

class MSZoning Street Alley 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 PoolQC Fence MiscFeature SaleType SaleCondition;

model logSalePrice = MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold / selection=LASSO(choose=AIC stop=CV) CVdetails ;

output out= results8 p=predict;

run;

quit;

ods graphics off;

\*Model 3; /\*Includes Ames training & test sets along with no log transformations conducted on predictors\*/ /\*LASSO CV-AIC\*/

/\*data is parititioned training 60% test 40%\*/

ods graphics on;

proc glmselect data=ames\_train2 seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

partition fraction(test=0.4);

class MSZoning Street Alley 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 PoolQC Fence MiscFeature SaleType SaleCondition;

model logSalePrice = MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold / selection=LASSO(choose=CV stop=AIC) CVdetails ;

output out=results9 p=predict;

run;

quit;

ods graphics off;

\*Model 4;

\*No variables with high VIF scores included in model;

\*Predictors with VIF scores >= 10 were not included;

\*Includes Ames training \* test sets along with no log transformations conducted on predictors

\*Partitioned training 60% test 40%, ElasticNet CV-AIC;

ods graphics on;

proc glmselect data=ames\_train2 seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);

partition fraction(test=0.4);

class MSZoning Street Alley 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 PoolQC Fence MiscFeature SaleType SaleCondition;

model logSalePrice = MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF FirstFlrSF SecondFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd Fireplaces GarageYrBlt GarageCars GarageArea WoodDeckSF OpenPorchSF EnclosedPorch ThreeSsnPorch ScreenPorch PoolArea MiscVal MoSold YrSold / selection=elasticnet(choose=CV stop=AIC) CVdetails ;

output out=results10 p=predict;

run;

quit;

ods graphics off;

/\*Use the following code to output the proper file to submi in kaggle\*/

data results10;

set results7;

if predict > 0 then logSalePrice = exp(Predict); \*uncomment if SalePrice is logged;

\*if predict > 0 then logSalePrice = Predict;

if predict < 0 then logSalePrice = 10000; \*This gets rid of any negative values;

keep id logSalePrice; \*we only want to keep ID and logSalePrice;

where id > 1460; \*We only want to include logSalePrice from the empty Test dataset;

run;

proc export data = results10 outfile = \_dataout dbms = csv replace;

run;