Model Selection to Predict the Residential Sale Price of Properties in Ames, Iowa

Philip Edwards, Manisha Pednekar, Mihir Parikh and David Stroud

MSDS 6372

Introduction

This project used the data from the Ames Housing data set which was compiled by Dean De Cock and has been used extensively in the field of data science education. The data set consists of 79 different explanatory variables, describing almost every aspect of residential homes in Ames Iowa.

Our team conducted Exploratory Data Analysis (EDA), conducted two different types of analysis and used three competing models in each of the analysis to predict the final price of each home.

Analysis Question 1

We will conduct an analysis to get an estimate of how the sale price of a house is related to the square footage of the living area of the house, overall room count of the house, and if it depends on which neighborhood the house is located in.

Analysis Question 1 EDA

We treated analysis question 1 as rigid application of the provided data without performing any extensive data cleansing/manipulation. The output is developed without trying to 'improve' the model other than using log transforms and removal of a few outliers.

Analysis Question 1 Results

After initial inspection of the Figures 1-3 below, we will proceed with log transformation on both SalePrice and GrLivArea to generate our model. It is worth noting that for a simple real estate model, we would discuss possible assumption violations with the requesting realtor to weigh ease of use (not log transforming and therefore not having to interpret sales price in log dollars) vs. accuracy of model tradeoffs.

Analysis Question 1 Model

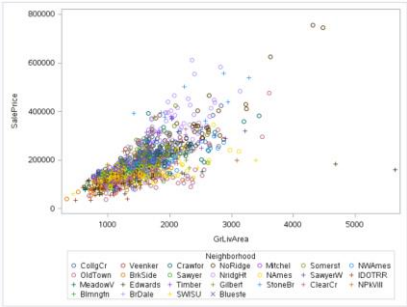

Figure 1 log(SalePrice) vs log(GrLivArea)

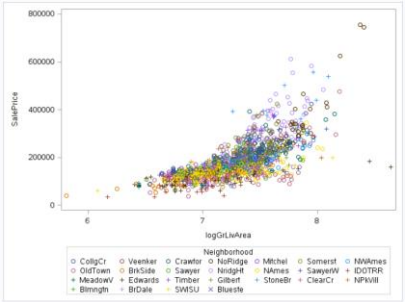

Figure 2 SalePrice vs log(GrLivArea)

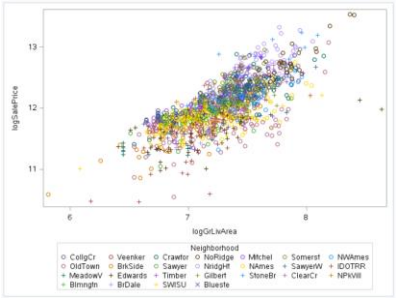

Figure 3 SalePrice vs GrLivArea

We also removed outliers by filtering out anything with GrLivArea > 4000. The scatterplots for our cleaned data are shown below for both GrLivArea and TotRmsAbvGrd.

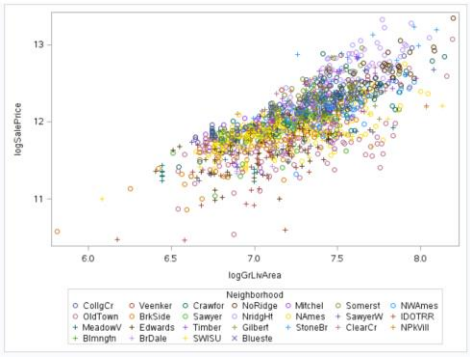

Figure 4 Final log(SalePrice) vs log(GrLivArea)

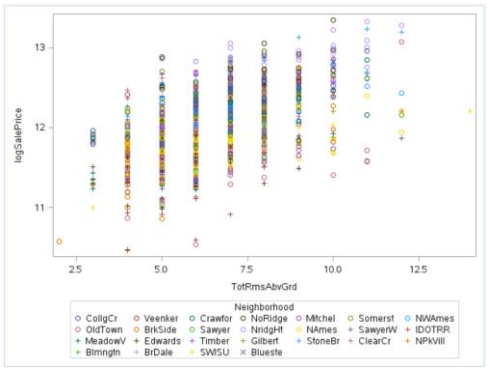

Figure 5 Final log(SalePrice) vs TotRmsAbvGrd

With this data, we generated the following model equation to fit using SAS. The model performed slightly better with interactions, but we did not include in the final model below and focused on ease of measurement and interpretability.

$$\mu\{\log(SalePrice) \mid \log(GrLivArea),\ Neighborhood,\ TotRmsAbvGrd\} = \beta_0 + \beta_1 \log(GrLivArea) + \beta_2 Neighborhood + \beta_3 TotRmsAbvGrd$$

Parameter	Estimate		Standard Error	t Value	Pr > |t|	95% Confidence Limits	
Intercept	7.168672282	B	0.20260431	35.38	<.0001	6.771238519	7.566106045
logGrLivArea	0.722889464		0.03006977	24.04	<.0001	0.663903846	0.781875081
Neighborhood Blmngtn	-0.121854885	B	0.07399429	-1.65	0.0998	-0.267003959	0.023294190
Neighborhood Blueste	-0.462694276	B	0.14679299	-3.15	0.0017	-0.750647138	-0.174741413
Neighborhood BrDale	-0.592610369	B	0.07517449	-7.88	<.0001	-0.740074553	-0.445146185
Neighborhood BrkSide	-0.465986932	B	0.06317076	-7.38	<.0001	-0.589904312	-0.342069552
Neighborhood ClearCr	-0.184175003	B	0.06801425	-2.71	0.0069	-0.317593483	-0.050756524
Neighborhood CollgCr	-0.126025427	B	0.05976513	-2.11	0.0351	-0.243262229	-0.008788625
Neighborhood Crawfor	-0.203162598	B	0.06362965	-3.19	0.0014	-0.327980132	-0.078345064
Neighborhood Edwards	-0.480237912	B	0.06107959	-7.86	<.0001	-0.600053183	-0.360422641
Neighborhood Gilbert	-0.214125468	B	0.06164052	-3.47	0.0005	-0.335041080	-0.093209856
Neighborhood IDOTRR	-0.674552814	B	0.06612353	-10.20	<.0001	-0.804262413	-0.544843215
Neighborhood MeadowV	-0.577602806	B	0.07441920	-7.76	<.0001	-0.723585409	-0.431620203
Neighborhood Mitchel	-0.269701843	B	0.06394964	-4.22	<.0001	-0.395147093	-0.144256592
Neighborhood NAmes	-0.339738153	B	0.05927574	-5.73	<.0001	-0.456014949	-0.223461356
Neighborhood NPkVill	-0.334042158	B	0.08598265	-3.88	0.0001	-0.502707920	-0.165376397
Neighborhood NWAmes	-0.265952631	B	0.06194918	-4.29	<.0001	-0.387473723	-0.144431540
Neighborhood NoRidge	0.003914099	B	0.06571831	0.06	0.9525	-0.125000619	0.132828817
Neighborhood NridgHt	0.154185329	B	0.06181881	2.49	0.0127	0.032919982	0.275450677
Neighborhood OldTown	-0.564420160	B	0.06055659	-9.32	<.0001	-0.683209518	-0.445630802
Neighborhood SWISU	-0.554029914	B	0.06933728	-7.99	<.0001	-0.690043694	-0.418016135
Neighborhood Sawyer	-0.350340111	B	0.06214458	-5.64	<.0001	-0.472244505	-0.228435716
Neighborhood SawyerW	-0.242145995	B	0.06281769	-3.85	0.0001	-0.365370779	-0.118921211
Neighborhood Somerst	-0.067099285	B	0.06116488	-1.10	0.2728	-0.187081880	0.052883310
Neighborhood StoneBr	0.131643018	B	0.06916747	1.90	0.0572	-0.004037656	0.267323691
Neighborhood Timber	-0.047469144	B	0.06550361	-0.72	0.4688	-0.175962699	0.081024411
Neighborhood Veenker	0.000000000	B	.	.	.	.	.
TotRmsAbvGrd	-0.019512203		0.00562523	-3.47	0.0005	-0.030546790	-0.008477616

Table 1 PROC GLM Output for Model

This can be interpreted for each neighborhood per the model output above. A sample walkthrough of a few neighborhoods is shown below, but the entire list of 25 neighborhoods will not be included for brevity:
For Blmngtn: $\log(SalePrice) = 7.1687 + 0.7229\log(GrLivArea) - .0195(TotRmsAbvGrd)$
For Edwards: $\log(SalePrice) = 7.0468 + 0.7229\log(GrLivArea) - .0195(TotRmsAbvGrd)$
For Veenker: $\log(SalePrice) = 6.6884 + 0.7229\log(GrLivArea) - .0195(TotRmsAbvGrd)$

Analysis Question 1 Assumption Analysis

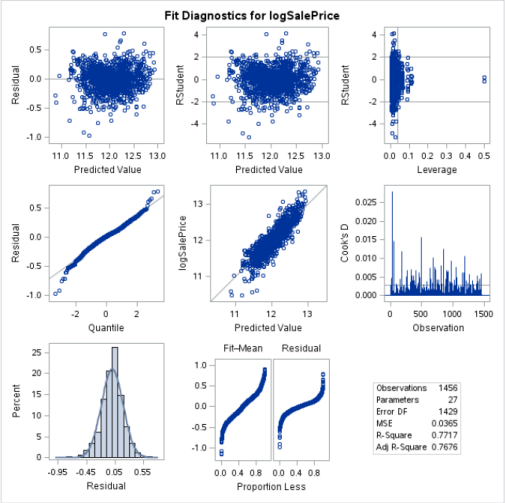

Figure 6 Fit Diagnostics from PROC GLM

We may assume that all assumptions are met are based on the plots in Figure 6.

- **Normality:** The residuals are distributed evenly, histogram of residuals are normally distributed, and QQ plot of residuals is linear.
- **Linear Trend:** From scatter plots (Figures 1-5), there is little evidence against linear trend.
- **Equal Standard Deviation:** There is little evidence from the scatterplot of heteroscedasticity.
- **Independence:** No evidence shows the observations are dependent. We'll assume no violation of independence.
- **Influential Observations:** Diagnostics plots show not enough evidence of any observation with outstandingly high leverage or cook's D.

Parameter Interpretation

One of the goals of this analysis was ease of interpretation. We did include log transforms, but the following parameter interpretations will convert this back to easier talking points. Again, using our 3 example neighborhoods instead of entire list for brevity.

- **Intercept:**
  - If living area is 0 feet and number of rooms are 0, the median sale price of houses in Bloomington will be $1298, the median sale price of houses in Edwards will be $1149; the median sale price of houses in Veenker will be $803. This is not practical, but a baseline for explanation of neighborhood sale price deltas.
- **Slopes:**
  - Each doubling of greater living area square footage while holding total rooms variable constant results in a 2^.723 = 1.65 multiplicative change in the median of the sale price of houses. That is a 65% increase in the sale price.
  - In addition, holding living area variable constant, for each 1 unit increase in the total rooms above ground associated with a multiplicative change of e^-0.01951 = .9807. That is a 1.93% decrease in the median estimated / predicted house price.
- **Multicollinearity:**
  - Using PROC CORR, we did see large positive correlation between TotRmsAbvGrd and logGrLivArea.

Pearson Correlation Coefficients Prob > |r| under H0: Rho=0 Number of Observations	logSalePrice	logGrLivArea	TotRmsAbvGrd	cNeighborhood
logSalePrice	1.00000 1456	0.73281 <.0001 1456	0.53345 <.0001 1456	0.16923 <.0001 1456
logGrLivArea	0.73281 <.0001 1456	1.00000 2915	0.80271 <.0001 2915	0.14925 <.0001 2915
TotRmsAbvGrd	0.53345 <.0001 1456	0.80271 <.0001 2915	1.00000 2915	0.09985 <.0001 2915
cNeighborhood	0.16923 <.0001 1456	0.14925 <.0001 2915	0.09985 <.0001 2915	1.00000 2915

```
proc corr data = house;
  var logSalePrice logGrLivArea TotRmsAbvGrd cNeighborhood;
run;
```

Figure 7 Correlation Table

- Multicollinearity can have an adverse effect on estimates of regression coefficients when several predictors in a multiple regression model that are highly correlated with each other, in this case we will leave them both in since it is only 2 variables. We did, however, run PROC REG using collinearity diagnostics to further investigate.

Collinearity Diagnostics						
			Proportion of Variation			
Number	Eigenvalue	Condition Index	Intercept	cNeighborhood	logGrLivArea	TotRmsAbvGrd
1	3.83728	1.00000	0.00005982	0.00998	0.00004363	0.00123
2	0.12873	5.45982	0.00043011	0.96360	0.00033030	0.01622
3	0.03362	10.68416	0.00715	0.01776	0.00201	0.34769
4	0.00037376	101.32488	0.99236	0.00866	0.99762	0.63487

```
proc reg data = house;
model logSalePrice = cNeighborhood logGrLivArea TotRmsAbvGrd / vif tol collin;
run;
```

Figure 8 Collinearity Diagnostics

Analysis 1 Conclusion:

Significant evidence shows that the increase of sale price of the houses is associated with square footage, location, and number of rooms. It is estimated that these factors explain 77% of the sale price (without accounting for interactions, stacking, etc.).

Analysis Question 2

Using the Ames Housing Data set, we analyzed three different models to determine which one would be the most predictive. Our models included Forward Selection, LASSO, and Stepwise Regression. We used the same testing and training datasets as in question one. The EDA in question two was conducted separate from question one.

Analysis Question 2 EDA

- Outlier removal of GrLivArea from homes with more than 4,000 square feet
- Upon review of the data description, it was determined that NA does not always mean that the values are missing. Data was coded to reflect a meaningful value.
- In cases where a very large portion of values are missing for a given feature, we may simply decide to drop that feature (column) altogether

Analysis Question 2 Assumption Analysis

- We may assume that all assumptions are met are based on the plots in Figure 9
- See notes from Figure 6 of this document for more details

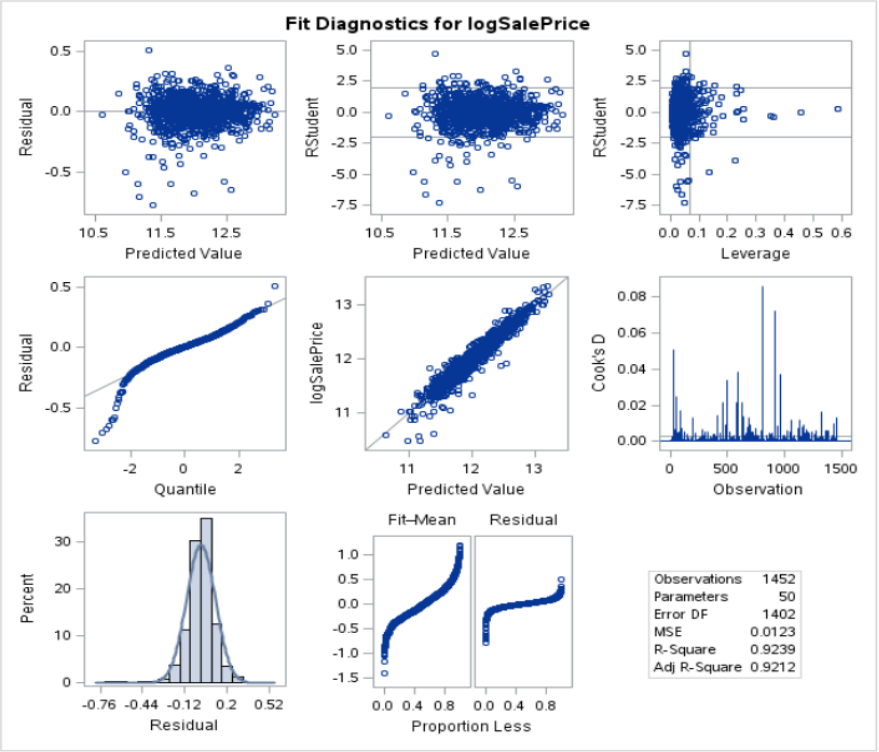

Figure 9 Fit Diagnostics from PROC GLM

Test Set Models

Forward selection Test data for Kaggle score: 1. Kaggle score: 0.12060	Analysis of Variance Source / DF / Sum of Squares / Mean Square / F Value Model / 33 / 207.38585 / 6.28442 / 457.25 Error / 1418 / 19.48909 / 0.01374 Corrected Total / 1451 / 226.87494 Root MSE 0.11724 Dependent Mean 12.02041 R-Square 0.9141 Adj R-Sq 0.9121 AIC -4737.34305 AICC -4735.56339 SBC -6011.79935 CV PRESS 20.92899
LASSO Test data for Kaggle score: 1. Kaggle score: 0.12077	Analysis of Variance Source / DF / Sum of Squares / Mean Square / F Value Model / 29 / 205.06560 / 7.07123 / 461.05 Error / 1422 / 21.80933 / 0.01534 Corrected Total / 1451 / 226.87494 Root MSE 0.12384 Dependent Mean 12.02041 R-Square 0.9039 Adj R-Sq 0.9019 AIC -4582.01752 AICC -4580.62033 SBC -5877.59660 CV PRESS 21.41060

<table>
<tr>
<td>Stepwise Regression

Test data for Kaggle score:
1. Kaggle score: 0.12085</td>
<td>
Analysis of Variance
<table>
<tr><th>Source</th><th>DF</th><th>Sum of Squares</th><th>Mean Square</th><th>F Value</th></tr>
<tr><td>Model</td><td>26</td><td>206.76214</td><td>7.95239</td><td>563.43</td></tr>
<tr><td>Error</td><td>1425</td><td>20.11279</td><td>0.01411</td><td></td></tr>
<tr><td>Corrected Total</td><td>1451</td><td>226.87494</td><td></td><td></td></tr>
</table>
<table>
<tr><td>Root MSE</td><td>0.11880</td></tr>
<tr><td>Dependent Mean</td><td>12.02041</td></tr>
<tr><td>R-Square</td><td>0.9113</td></tr>
<tr><td>Adj R-Sq</td><td>0.9097</td></tr>
<tr><td>AIC</td><td>-4705.60318</td></tr>
<tr><td>AICC</td><td>-4704.46192</td></tr>
<tr><td>SBC</td><td>-6017.02435</td></tr>
<tr><td>CV PRESS</td><td>21.47407</td></tr>
</table>
</td>
</tr>
<tr>
<td>Brute Force with pseudo-domain knowledge interaction creation.

Test data for Kaggle score:
1. Kaggle score: 0.11841</td>
<td>
The GLM Procedure
Dependent Variable: logSalePrice
<table>
<tr><th>Source</th><th>DF</th><th>Sum of Squares</th><th>Mean Square</th><th>F Value</th><th>Pr > F</th></tr>
<tr><td>Model</td><td>54</td><td>210.1852261</td><td>3.8923190</td><td>325.80</td><td><.0001</td></tr>
<tr><td>Error</td><td>1397</td><td>16.6897116</td><td>0.0119468</td><td></td><td></td></tr>
<tr><td>Corrected Total</td><td>1451</td><td>226.8749378</td><td></td><td></td><td></td></tr>
</table>
<table>
<tr><th>R-Square</th><th>Coeff Var</th><th>Root MSE</th><th>logSalePrice Mean</th></tr>
<tr><td>0.926437</td><td>0.909300</td><td>0.109302</td><td>12.02041</td></tr>
</table>
</td>
</tr>
</table>

Figure 10 Test Set Models

Model Interpretation

We used the following models for a comparison of variable selection techniques for sales price prediction.

- **Forward Selection:**
  - The simplest data-driven model building approach is forward selection. In this approach, one adds variables to the model one at a time. At each step, each variable that is not already in the model is tested for inclusion into the model. The most significant of these variables is added to the model, so long as its p-value is below a pre-set level.
- **LASSO:**
  - Lasso is a shrinkage and selection method for linear regression. It minimizes the usual sum of squared errors, with a bound of the sum of the absolute values of the coefficient. The main problem with lasso regression is when we have correlated variables, it retains only one variable and sets other correlated variables to zero. That will possibly lead to some loss of information resulting in lower accuracy in our model.

- **Stepwise Regression:**
  - This is semi-automated process of building a model by successively adding or removing variables based solely on the t-statistics of their estimated coefficients.

Model Selection

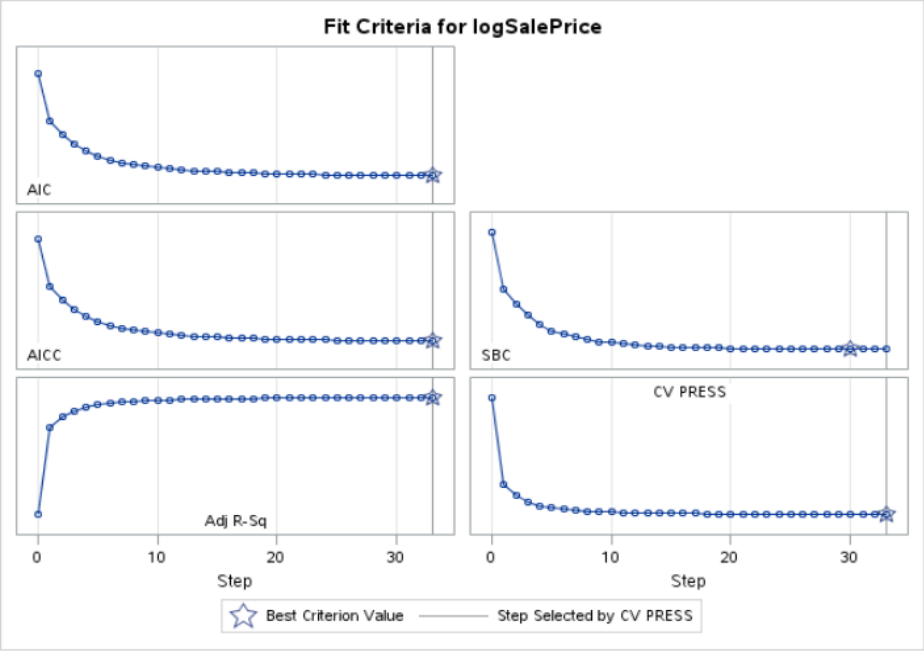

Figure 11 Fit Criteria

CV Press AIC AICC SBC, and Adjusted R2

- These techniques adjust the training error for the model size and can be used to select among a set of models with different numbers of variables.

Analysis 2 Conclusion:

Our analysis concludes that the most predictive model is the Forward Selection, but we were able to improve Kaggle scores by brute force feature engineering and retesting iteratively. The predictive models derived from three variable selection techniques produce similar efficiently in this analysis.

Conclusion

The first question in the analysis sought to determine each interpretation of parameters for helping real estate agents, contractores and prospective buyers gain insight into important factors that influence housing prices in Ames, Iowa. Our research shows that the increase of sale price of the houses is associated with square footage, location, and number of rooms.

In the second question, the model was aimed at determining the most predictive model. Our most predictive model was the Forward Selection. Further testing allowed our team to use domain knowledge and interaction creation to achieve a better predicting model, thus a higher Kaggle score.

Appendix A: Exploratory Data Analysis

The raw ingest begins with an extremely long SAS 'infile' statement defining each 'informat', 'format', and variable type as opposed to the much simpler 'PROC IMPORT datafile' due to SAS limitations encountered. In addition, we did have a minor edit to the .csv files headers (must begin with non-numeric character) before import. We also used R for data cleanup resulting in our best Kaggle score.

Three column headers are renamed in train.csv and test.csv as follows before data ingest:

- 3SsnPorch changed to X3SsnPorch
- 1stFlrSF changed to X1stFlrSF
- 2ndFlrSF changed to X2ndFlrSF

Data Cleansing

We did quite a bit of research in this area, and used several practices learned from this research. Reference links are provided in a later appendix.

```
/* Set categorical variables to numeric */
data house;
set house;
IF Alley = "Pave" THEN cAlley = 1;
IF Alley ^= "Pave" THEN cAlley = 0;
IF BldgType = "1Fam" OR "TwnhsE" THEN cBldgType = 1;
IF BldgType ^= "1Fam" OR "TwnhsE" THEN cBldgType = 0;
IF BsmtCond = "Ex" THEN cBsmtCond = 5;
IF BsmtCond = "Gd" THEN cBsmtCond = 4;
IF BsmtCond = "TA" THEN cBsmtCond = 3;
IF BsmtCond = "Fa" THEN cBsmtCond = 2;
IF BsmtCond = "NA" THEN cBsmtCond = 1;
IF BsmtCond = "Po" THEN cBsmtCond = 0;
IF BsmtCond ^="Ex" OR "Gd" OR "TA" OR "Fa" OR "NA" OR "Po" THEN cBsmtCond = 1;
IF BsmtExposure = "Gd" THEN cBsmtExposure = 3;
IF BsmtExposure = "Av" THEN cBsmtExposure = 2;
```

Figure A1 Example Data Cleansing

We converted all variables to numeric. As shown in Figure A1, we converted 'paved alley' to a 1 and 'not paved' to 0. In addition, we tried to preserve ordinality when possible as shown on the basement condition rows in Figure A1. Finally, there were some typos or missing values in the data that we addressed with mode imputation if we didn't know where they should fall, or assigning the appropriate value if our research gave us insight on where they should fall.

As shown in Figure A2, we also releveled some categorical variables that contained extremely rare classes. In this case, gas=1 and other=0 instead of using 5,4,3,2,1 in this case.

The FREQ Procedure

Heating	Frequency	Percent
Floor	1	0.07
GasA	1428	97.81
GasW	18	1.23
Grav	7	0.48
OthW	2	0.14
Wall	4	0.27

Figure A2 Heating Counts

We will not go through all variables here, but we basically stepped through each categorical variable in this manner until they were converted to numeric.

Missing Data

After taking care of these basic cleaning issues, we needed to address variables with missing values that we could fill in with supporting data.

The MEANS Procedure

Variable	N Miss	N
LotFrontage	480	2423
GarageYrBlt	159	2744

Figure A3 Missing Data Counts

Figure A3 shows that 'LotFrontage' and 'GarageYrBlt' both have quite a few missing values. We assigned any empty values for GarageYrBlt to equal YearBuilt. We were hoping to have the system create a polynomial to assign LotFrontage based on LotArea values (which we could do in R and Python), but settled for a percentage value in SAS instead.

Normalization

We normalized all continuous variables with log (+1 to remove possibility of log0) transformation. Figure A4 below shows the improved skewness from SalePrice to log(SalePrice). We will not include all continuous variable graphs here. Since YearSold value is 2006-2010, we normalized it by creating a value of 2010-YearSold.

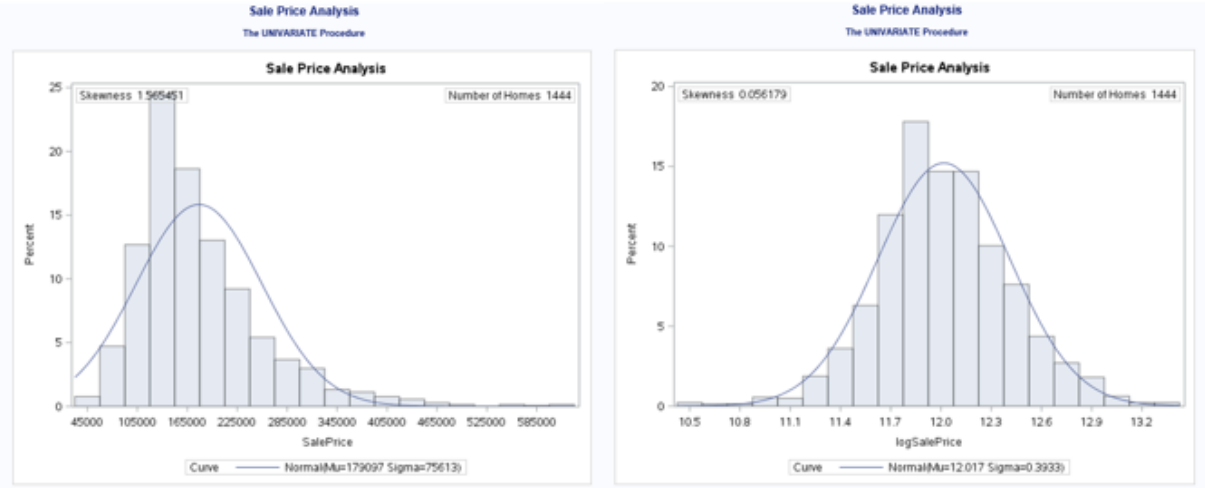

Figure A4 Improving Skewness with log+1 Transforms

Feature Engineering

Our feature engineering was developed from online research and from correlation plots in R (at the time of this writing we could not emulate exactly same way in SAS). We will define these below followed by a screenshot of the R code output (Figure A5) example correlation matrix.

- Bathrooms per room in the property. People looking for more bedrooms probably want more bathrooms.
  - BRBR = log(((FullBath + HalfBath)+1) / (1+BedroomAbvGr));
- Average room size. Larger rooms usually are desired.
  - GLARooms = log(1+(GrLivArea / TotRmsAbvGrd));
- Size of the garage affects the importance of its quality component.
  - GQGC = log(1+(cGarageQual * GarageCars));
- Size of living area in property vs. average living area size
  - GRGR=log(1+(GrLivArea/&n)); → &n = mean of all GrLivArea
- Size of the living area affects the importance of its quality component.

- OQGLA = log(1+(OverallQual * GrLivArea));

- The rest of these are based on correlation plots
  - OQTBSF = log(1+(OverallQual * TotalBsmtSF));
  - OQYB = log(1+(OverallQual * YearBuilt));
  - OQYR = log(1+(OverallQual * YearRemodAdd));
  - OQEC = log(1+(OverallQual * cExterCond));
  - OQFB = log(1+(OverallQual * FullBath));

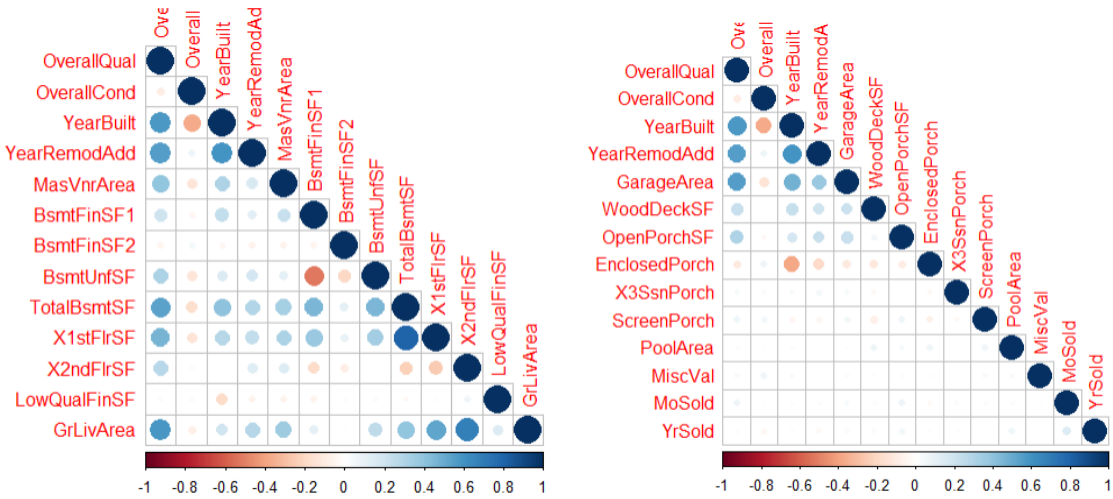

Figure A5 R Correlation Matrix Output

R code snippet for one of these matrixes shown below.

```
> library(corrplot)
> correlations <- cor(train[,c(5,6,7,8, 16:25)], use="everything")
> corrplot(correlations, method="circle", type="lower",  sig.level = 0.01, insig = "blank")
```

We are stacking some variables, but more of a plug and chug effort instead of any deeper dive analysis due to time. We do know what we expect to be highly important to the model from the correlations described above as well as general real estate domain knowledge.

Outlier Removal

We did remove outliers (from the training data set) as seen in Figure A6 below non-transformed. They correspond to property ids 1299, 524, 1183, and 692.

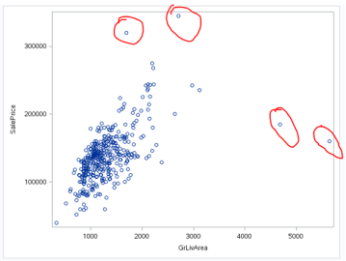

Figure A6 Sale Price vs GrLivArea Outliers

Appendix B: References

- https://www.fortunebuilders.com/what-are-the-biggest-factors-in-determining-property-value/
- http://ww2.amstat.org/publications/jse/v19n3/decock.pdf
- https://www.kaggle.com/skirmer/fun-with-real-estate-data
- https://nycdatascience.com/blog/student-works/kaggles-advanced-regression-competition-predicting-housing-prices-in-ames-iowa/
- http://support.sas.com/documentation/94/
- Also used EDA code and excerpts from MSDS 6371 Homework 13-14 submission.

Appendix C Code Repositories:

Analysis 1 Code Dump

```
/*************************************************************************************************************
  Please note that three column headers are renamed in .csv as follows before ingest
                        * 3SsnPorch changed to X3SsnPorch
                        * 1stFlrSF changed to X1stFlrSF
                        * 2ndFlrSF changed to X2ndFlrSF
  Other than the 3 changes above, we imported the untouched raw train and test files.
  PROC IMPORT datafile='data.csv' method created truncated columns, wrong data types, etc.
  We had to use the long winded code below based on the log file of the .csv import and manually edit
  the variable types to get the imported data to work.
*************************************************************************************************************/

/* Import training data, filter out unused columns for Analysis 1  */
data WORK.TRAIN;
%let _EFIERR_ = 0; /* set the ERROR detection macro variable */
infile '/home/pbedwards0/Phil/train.csv' delimiter = ',' MISSOVER DSD lrecl=32767 firstobs=2 ;  /* <---- YOUR PATH HERE */
informat Id best32. ;
informat MSSubClass best32. ;
informat MSZoning $20. ;
informat LotFrontage best32. ;
informat LotArea best32. ;
informat Street $20. ;
informat Alley $20. ;
informat LotShape $20. ;
informat LandContour $20. ;
informat Utilities $20. ;
informat LotConfig $20. ;
informat LandSlope $20. ;
informat Neighborhood $20. ;
informat Condition1 $$20. ;
informat Condition2 $20. ;
informat BldgType $20. ;
informat HouseStyle $20. ;
informat OverallQual best32. ;
informat OverallCond best32. ;
informat YearBuilt best32. ;
informat YearRemodAdd best32. ;
informat RoofStyle $20. ;
informat RoofMatl $20. ;
informat Exterior1st $20. ;
informat Exterior2nd $20. ;
informat MasVnrType $20. ;
informat MasVnrArea best32. ;
informat ExterQual $20. ;
informat ExterCond $$20. ;
informat Foundation $20. ;
informat BsmtQual $20. ;
informat BsmtCond $20. ;
informat BsmtExposure $20. ;
informat BsmtFinType1 $20. ;
informat BsmtFinSF1 best32. ;
informat BsmtFinType2 $20.;
informat BsmtFinSF2 best32. ;
informat BsmtUnfSF best32. ;
```

```
informat TotalBsmtSF best32. ;
informat Heating $20. ;
informat HeatingQC $20. ;
informat CentralAir $20. ;
informat Electrical $20. ;
informat X1stFlrSF best32. ;
informat X2ndFlrSF best32. ;
informat LowQualFinSF best32. ;
informat GrLivArea best32. ;
informat BsmtFullBath best32. ;
informat BsmtHalfBath best32. ;
informat FullBath best32. ;
informat HalfBath best32. ;
informat BedroomAbvGr best32. ;
informat KitchenAbvGr best32. ;
informat KitchenQual $20. ;
informat TotRmsAbvGrd best32. ;
informat Functional $20. ;
informat Fireplaces best32. ;
informat FireplaceQu $20. ;
informat GarageType $20. ;
informat GarageYrBlt best32. ;
informat GarageFinish $20. ;
informat GarageCars best32. ;
informat GarageArea best32. ;
informat GarageQual $20. ;
informat GarageCond $20. ;
informat PavedDrive $20. ;
informat WoodDeckSF best32. ;
informat OpenPorchSF best32. ;
informat EnclosedPorch best32. ;
informat X3SsnPorch best32. ;
informat ScreenPorch best32. ;
informat PoolArea best32. ;
informat PoolQC $20. ;
informat Fence $20. ;
informat MiscFeature $20. ;
informat MiscVal best32. ;
informat MoSold best32. ;
informat YrSold best32. ;
informat SaleType $20. ;
informat SaleCondition $20. ;
informat SalePrice best32. ;
format Id best32. ;
format MSSubClass best32. ;
format MSZoning $20. ;
format LotFrontage best32. ;
format LotArea best32. ;
format Street $20. ;
format Alley $20. ;
format LotShape $20. ;
format LandContour $20.;
format Utilities $20. ;
format LotConfig $20. ;
format LandSlope $20. ;
format Neighborhood $20. ;
format Condition1 $20. ;
format Condition2 $20. ;
format BldgType $20. ;
format HouseStyle $20. ;
format OverallQual best32. ;
format OverallCond best32. ;
format YearBuilt best32. ;
format YearRemodAdd best32. ;
format RoofStyle $20. ;
format RoofMatl $20. ;
format Exterior1st $20. ;
format Exterior2nd $20. ;
format MasVnrType $20. ;
format MasVnrArea best32. ;
format ExterQual $20. ;
format ExterCond $20. ;
format Foundation $20. ;
format BsmtQual $20. ;
format BsmtCond $20. ;
```

```
format BsmtExposure $20. ;
format BsmtFinType1 $20. ;
format BsmtFinSF1 best32. ;
format BsmtFinType2 $20. ;
format BsmtFinSF2 best32. ;
format BsmtUnfSF best32. ;
format TotalBsmtSF best32. ;
format Heating $20. ;
format HeatingQC $20. ;
format CentralAir $20. ;
format Electrical $20. ;
format X1stFlrSF best32. ;
format X2ndFlrSF best32. ;
format LowQualFinSF best32. ;
format GrLivArea best32. ;
format BsmtFullBath best32. ;
format BsmtHalfBath best32. ;
format FullBath best32. ;
format HalfBath best32. ;
format BedroomAbvGr best32. ;
format KitchenAbvGr best32. ;
format KitchenQual $20. ;
format TotRmsAbvGrd best32. ;
format Functional $20. ;
format Fireplaces best32. ;
format FireplaceQu $20. ;
format GarageType $20. ;
format GarageYrBlt best32. ;
format GarageFinish $20. ;
format GarageCars best32. ;
format GarageArea best32. ;
format GarageQual $20. ;
format GarageCond $20. ;
format PavedDrive $20. ;
format WoodDeckSF best32. ;
format OpenPorchSF best32. ;
format EnclosedPorch best32. ;
format X3SsnPorch best32. ;
format ScreenPorch best32. ;
format PoolArea best32. ;
format PoolQC $20. ;
format Fence $20. ;
format MiscFeature $20. ;
format MiscVal best32. ;
format MoSold best32. ;
format YrSold best32. ;
format SaleType $20. ;
format SaleCondition $20. ;
format SalePrice best32. ;
input
Id
MSSubClass
MSZoning $
LotFrontage
LotArea
Street $
Alley $
LotShape $
LandContour $
Utilities $
LotConfig $
LandSlope $
Neighborhood $
Condition1 $
Condition2 $
BldgType $
HouseStyle $
OverallQual
OverallCond
YearBuilt
YearRemodAdd
RoofStyle $
RoofMatl $
Exterior1st $
Exterior2nd $
```

```
MasVnrType $
MasVnrArea
ExterQual $
ExterCond $
Foundation $
BsmtQual $
BsmtCond $
BsmtExposure $
BsmtFinType1 $
BsmtFinSF1
BsmtFinType2 $
BsmtFinSF2
BsmtUnfSF
TotalBsmtSF
Heating $
HeatingQC $
CentralAir $
Electrical $
X1stFlrSF
X2ndFlrSF
LowQualFinSF
GrLivArea
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
KitchenQual $
TotRmsAbvGrd
Functional $
Fireplaces
FireplaceQu $
GarageType $
GarageYrBlt
GarageFinish $
GarageCars
GarageArea
GarageQual $
GarageCond $
PavedDrive $
WoodDeckSF
OpenPorchSF
EnclosedPorch
X3SsnPorch
ScreenPorch
PoolArea
PoolQC $
Fence $
MiscFeature $
MiscVal
MoSold
YrSold
SaleType $
SaleCondition $
SalePrice
;
if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection macro variable */
run;


/* Import test data */
data WORK.TEST;
%let _EFIERR_ = 0; /* set the ERROR detection macro variable */
infile '/home/pbedwards0/Phil/test.csv' delimiter = ',' MISSOVER DSD lrecl=32767 firstobs=2 ;  /* <---- YOUR PATH HERE */
informat Id best32. ;
informat MSSubClass best32. ;
informat MSZoning $20. ;
informat LotFrontage best32. ;
informat LotArea best32. ;
informat Street $20. ;
informat Alley $20. ;
informat LotShape $20. ;
informat LandContour $20. ;
informat Utilities $20. ;
```

```
informat LotConfig $20. ;
informat LandSlope $20. ;
informat Neighborhood $20. ;
informat Condition1 $$20. ;
informat Condition2 $20. ;
informat BldgType $20. ;
informat HouseStyle $20. ;
informat OverallQual best32. ;
informat OverallCond best32. ;
informat YearBuilt best32. ;
informat YearRemodAdd best32. ;
informat RoofStyle $20. ;
informat RoofMatl $20. ;
informat Exterior1st $20. ;
informat Exterior2nd $20. ;
informat MasVnrType $20. ;
informat MasVnrArea best32. ;
informat ExterQual $20. ;
informat ExterCond $$20. ;
informat Foundation $20. ;
informat BsmtQual $20. ;
informat BsmtCond $20. ;
informat BsmtExposure $20. ;
informat BsmtFinType1 $20. ;
informat BsmtFinSF1 best32. ;
informat BsmtFinType2 $20.;
informat BsmtFinSF2 best32. ;
informat BsmtUnfSF best32. ;
informat TotalBsmtSF best32. ;
informat Heating $20. ;
informat HeatingQC $20. ;
informat CentralAir $20. ;
informat Electrical $20. ;
informat X1stFlrSF best32. ;
informat X2ndFlrSF best32. ;
informat LowQualFinSF best32. ;
informat GrLivArea best32. ;
informat BsmtFullBath best32. ;
informat BsmtHalfBath best32. ;
informat FullBath best32. ;
informat HalfBath best32. ;
informat BedroomAbvGr best32. ;
informat KitchenAbvGr best32. ;
informat KitchenQual $20. ;
informat TotRmsAbvGrd best32. ;
informat Functional $20. ;
informat Fireplaces best32. ;
informat FireplaceQu $20. ;
informat GarageType $20. ;
informat GarageYrBlt best32. ;
informat GarageFinish $20. ;
informat GarageCars best32. ;
informat GarageArea best32. ;
informat GarageQual $20. ;
informat GarageCond $20. ;
informat PavedDrive $20. ;
informat WoodDeckSF best32. ;
informat OpenPorchSF best32. ;
informat EnclosedPorch best32. ;
informat X3SsnPorch best32. ;
informat ScreenPorch best32. ;
informat PoolArea best32. ;
informat PoolQC $20. ;
informat Fence $20. ;
informat MiscFeature $20. ;
informat MiscVal best32. ;
informat MoSold best32. ;
informat YrSold best32. ;
informat SaleType $20. ;
informat SaleCondition $20. ;
format Id best32. ;
format MSSubClass best32. ;
format MSZoning $20. ;
format LotFrontage best32. ;
format LotArea best32. ;
```

```
format Street $20. ;
format Alley $20. ;
format LotShape $20. ;
format LandContour $20.;
format Utilities $20. ;
format LotConfig $20. ;
format LandSlope $20. ;
format Neighborhood $20. ;
format Condition1 $20. ;
format Condition2 $20. ;
format BldgType $20. ;
format HouseStyle $20. ;
format OverallQual best32. ;
format OverallCond best32. ;
format YearBuilt best32. ;
format YearRemodAdd best32. ;
format RoofStyle $20. ;
format RoofMatl $20. ;
format Exterior1st $20. ;
format Exterior2nd $20. ;
format MasVnrType $20. ;
format MasVnrArea best32. ;
format ExterQual $20. ;
format ExterCond $20. ;
format Foundation $20. ;
format BsmtQual $20. ;
format BsmtCond $20. ;
format BsmtExposure $20. ;
format BsmtFinType1 $20. ;
format BsmtFinSF1 best32. ;
format BsmtFinType2 $20. ;
format BsmtFinSF2 best32. ;
format BsmtUnfSF best32. ;
format TotalBsmtSF best32. ;
format Heating $20. ;
format HeatingQC $20. ;
format CentralAir $20. ;
format Electrical $20. ;
format X1stFlrSF best32. ;
format X2ndFlrSF best32. ;
format LowQualFinSF best32. ;
format GrLivArea best32. ;
format BsmtFullBath best32. ;
format BsmtHalfBath best32. ;
format FullBath best32. ;
format HalfBath best32. ;
format BedroomAbvGr best32. ;
format KitchenAbvGr best32. ;
format KitchenQual $20. ;
format TotRmsAbvGrd best32. ;
format Functional $20. ;
format Fireplaces best32. ;
format FireplaceQu $20. ;
format GarageType $20. ;
format GarageYrBlt best32. ;
format GarageFinish $20. ;
format GarageCars best32. ;
format GarageArea best32. ;
format GarageQual $20. ;
format GarageCond $20. ;
format PavedDrive $20. ;
format WoodDeckSF best32. ;
format OpenPorchSF best32. ;
format EnclosedPorch best32. ;
format X3SsnPorch best32. ;
format ScreenPorch best32. ;
format PoolArea best32. ;
format PoolQC $20. ;
format Fence $20. ;
format MiscFeature $20. ;
format MiscVal best32. ;
format MoSold best32. ;
format YrSold best32. ;
format SaleType $20. ;
format SaleCondition $20. ;
```

input
Id
MSSubClass
MSZoning $
LotFrontage
LotArea
Street $
Alley $
LotShape $
LandContour $
Utilities $
LotConfig $
LandSlope $
Neighborhood $
Condition1 $
Condition2 $
BldgType $
HouseStyle $
OverallQual
OverallCond
YearBuilt
YearRemodAdd
RoofStyle $
RoofMatl $
Exterior1st $
Exterior2nd $
MasVnrType $
MasVnrArea
ExterQual $
ExterCond $
Foundation $
BsmtQual $
BsmtCond $
BsmtExposure $
BsmtFinType1 $
BsmtFinSF1
BsmtFinType2 $
BsmtFinSF2
BsmtUnfSF
TotalBsmtSF
Heating $
HeatingQC $
CentralAir $
Electrical $
X1stFlrSF
X2ndFlrSF
LowQualFinSF
GrLivArea
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
KitchenQual $
TotRmsAbvGrd
Functional $
Fireplaces
FireplaceQu $
GarageType $
GarageYrBlt
GarageFinish $
GarageCars
GarageArea
GarageQual $
GarageCond $
PavedDrive $
WoodDeckSF
OpenPorchSF
EnclosedPorch
X3SsnPorch
ScreenPorch
PoolArea
PoolQC $
Fence $

```
MiscFeature $
MiscVal
MoSold
YrSold
SaleType $
SaleCondition $
;
if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection macro variable */
run;

data train;
set train;
logGrLivArea = log(GrLivArea);
logSalePrice = log(SalePrice);
run;


data test;
set test;
logGrLivArea = log(GrLivArea);
logSalePrice = log(SalePrice);
run;

proc sgplot data=train;
scatter x=GrLivArea y=SalePrice / group=Neighborhood;
run;

proc sgplot data=train;
scatter x=logGrLivArea y=SalePrice / group=Neighborhood;
run;

proc sgplot data=train;
scatter x=logGrLivArea y=logSalePrice / group=Neighborhood;
run;

/* Outlier Removal */
data train;
set train;
    IF GrLivArea  > 4000 THEN DELETE;
  RUN;

proc sgplot data=train;
scatter x=logGrLivArea y=logSalePrice / group=Neighborhood;
run;
proc sgplot data=train;
scatter x=TotRmsAbvGrd y=logSalePrice / group=Neighborhood;
run;

/* Combine Data After Outlier Removal */
data house;
  set train test;
run;

data house;
set house;
IF Neighborhood="Blmngtn" THEN cNeighborhood = 1;
IF Neighborhood="Blueste" THEN cNeighborhood = 2;
IF Neighborhood="BrDale" THEN cNeighborhood = 3;
IF Neighborhood="BrkSide" THEN cNeighborhood = 4;
IF Neighborhood="ClearCr" THEN cNeighborhood = 5;
IF Neighborhood="CollgCr" THEN cNeighborhood = 6;
IF Neighborhood="Crawfor" THEN cNeighborhood = 7;
IF Neighborhood="Edwards" THEN cNeighborhood = 8;
IF Neighborhood="Gilbert" THEN cNeighborhood = 9;
IF Neighborhood="IDOTRR" THEN cNeighborhood = 10;
IF Neighborhood="MeadowV" THEN cNeighborhood = 11;
IF Neighborhood="Mitchel" THEN cNeighborhood = 12;
IF Neighborhood="NAmes" THEN cNeighborhood = 13;
IF Neighborhood="NoRidge" THEN cNeighborhood = 14;
IF Neighborhood="NPkVill" THEN cNeighborhood = 15;
IF Neighborhood="NridgHt" THEN cNeighborhood = 16;
IF Neighborhood="NWAmes" THEN cNeighborhood = 17;
IF Neighborhood="OldTown" THEN cNeighborhood = 18;
IF Neighborhood="Sawyer" THEN cNeighborhood = 19;
IF Neighborhood="SawyerW" THEN cNeighborhood = 20;
```

```
IF Neighborhood="Somerst" THEN cNeighborhood = 21;
IF Neighborhood="StoneBr" THEN cNeighborhood = 22;
IF Neighborhood="SWISU" THEN cNeighborhood = 23;
IF Neighborhood="Timber" THEN cNeighborhood = 24;
IF Neighborhood="Veenker" THEN cNeighborhood = 25;
run;

proc corr data = house;
  var logSalePrice logGrLivArea TotRmsAbvGrd cNeighborhood;
run;


proc reg data = house;
model logSalePrice = cNeighborhood logGrLivArea TotRmsAbvGrd / vif tol collin;
run;

/* Remove unused columns for Analysis 1 From house    */
proc sql noprint;
ALTER TABLE house DROP COLUMN MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour,
Utilities, LotConfig, LandSlope, Condition1, Condition2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt,
YearRemodAdd, RoofStyle, RoofMatl, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, ExterCond, Foundation,
BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinSF1, BsmtFinType2, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, Heating,
HeatingQC, CentralAir, Electrical, X1stFlrSF, X2ndFlrSF, LowQualFinSF, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath,
BedroomAbvGr, KitchenAbvGr, KitchenQual, Functional, Fireplaces, FireplaceQu, GarageType, GarageYrBlt,
GarageFinish, GarageCars, GarageArea, GarageQual, GarageCond, PavedDrive, WoodDeckSF, OpenPorchSF, EnclosedPorch,
X3SsnPorch, ScreenPorch, PoolArea, PoolQC, Fence, MiscFeature, MiscVal, MoSold, YrSold, SaleType, SaleCondition;
quit;


/* Model output code block. Cut paste variables from your testing output into the model below  */
proc glm data=house plots=all;
class Neighborhood;
model logSalePrice = Neighborhood logGrLivArea TotRmsAbvGrd
/solution cli clparm;
output out=results predicted=logpredict;
run;

ods graphics on;
proc glmselect data=train testdata = test
        seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL) ;
class Neighborhood;
model logSalePrice = Neighborhood logGrLivArea TotRmsAbvGrd / selection=LASSO( choose=CV stop=CV) CVdetails stb showpvalues;
/* modelaverage tables = (EffectSelectPct(all) ParmEst(All)) alpha = .05; */

run;
quit;
ods graphics off;
```

Analysis 2 Code Dump

```
data WORK.TRAIN;
%let _EFIERR_ = 0; /* set the ERROR detection macro variable */
infile '/home/jdstroud0/Stats2/AmesHousing/Version1/train.csv' delimiter = ',' MISSOVER DSD lrecl=32767 firstobs=2 ;  /* <---- YOUR PATH HERE */
informat Id best32. ;
informat MSSubClass best32. ;
informat MSZoning $20. ;
informat LotFrontage best32. ;
informat LotArea best32. ;
informat Street $20. ;
informat Alley $20. ;
informat LotShape $20. ;
informat LandContour $20. ;
informat Utilities $20. ;
informat LotConfig $20. ;
informat LandSlope $20. ;
informat Neighborhood $20. ;
informat Condition1 $20. ;
informat Condition2 $20. ;
informat BldgType $20. ;
informat HouseStyle $20. ;
informat OverallQual best32. ;
informat OverallCond best32. ;
informat YearBuilt best32. ;
informat YearRemodAdd best32. ;
```

```
informat RoofStyle $20. ;
informat RoofMatl $20. ;
informat Exterior1st $20. ;
informat Exterior2nd $20. ;
informat MasVnrType $20. ;
informat MasVnrArea best32. ;
informat ExterQual $20. ;
informat ExterCond $20. ;
informat Foundation $20. ;
informat BsmtQual $20. ;
informat BsmtCond $20. ;
informat BsmtExposure $20. ;
informat BsmtFinType1 $20. ;
informat BsmtFinSF1 best32. ;
informat BsmtFinType2 $20.;
informat BsmtFinSF2 best32. ;
informat BsmtUnfSF best32. ;
informat TotalBsmtSF best32. ;
informat Heating $20. ;
informat HeatingQC $20. ;
informat CentralAir $20. ;
informat Electrical $20. ;
informat X1stFlrSF best32. ;
informat X2ndFlrSF best32. ;
informat LowQualFinSF best32. ;
informat GrLivArea best32. ;
informat BsmtFullBath best32. ;
informat BsmtHalfBath best32. ;
informat FullBath best32. ;
informat HalfBath best32. ;
informat BedroomAbvGr best32. ;
informat KitchenAbvGr best32. ;
informat KitchenQual $20. ;
informat TotRmsAbvGrd best32. ;
informat Functional $20. ;
informat Fireplaces best32. ;
informat FireplaceQu $20. ;
informat GarageType $20. ;
informat GarageYrBlt best32. ;
informat GarageFinish $20. ;
informat GarageCars best32. ;
informat GarageArea best32. ;
informat GarageQual $20. ;
informat GarageCond $20. ;
informat PavedDrive $20. ;
informat WoodDeckSF best32. ;
informat OpenPorchSF best32. ;
informat EnclosedPorch best32. ;
informat X3SsnPorch best32. ;
informat ScreenPorch best32. ;
informat PoolArea best32. ;
informat PoolQC $20. ;
informat Fence $20. ;
informat MiscFeature $20. ;
informat MiscVal best32. ;
informat MoSold best32. ;
informat YrSold best32. ;
informat SaleType $20. ;
informat SaleCondition $20. ;
informat SalePrice best32. ;
format Id best32. ;
format MSSubClass best32. ;
format MSZoning $20. ;
format LotFrontage best32. ;
format LotArea best32. ;
format Street $20. ;
format Alley $20. ;
format LotShape $20. ;
format LandContour $20.;
format Utilities $20. ;
format LotConfig $20. ;
format LandSlope $20. ;
format Neighborhood $20. ;
format Condition1 $20. ;
format Condition2 $20. ;
```

```
format BldgType $20. ;
format HouseStyle $20. ;
format OverallQual best32. ;
format OverallCond best32. ;
format YearBuilt best32. ;
format YearRemodAdd best32. ;
format RoofStyle $20. ;
format RoofMatl $20. ;
format Exterior1st $20. ;
format Exterior2nd $20. ;
format MasVnrType $20. ;
format MasVnrArea best32. ;
format ExterQual $20. ;
format ExterCond $20. ;
format Foundation $20. ;
format BsmtQual $20. ;
format BsmtCond $20. ;
format BsmtExposure $20. ;
format BsmtFinType1 $20. ;
format BsmtFinSF1 best32. ;
format BsmtFinType2 $20. ;
format BsmtFinSF2 best32. ;
format BsmtUnfSF best32. ;
format TotalBsmtSF best32. ;
format Heating $20. ;
format HeatingQC $20. ;
format CentralAir $20. ;
format Electrical $20. ;
format X1stFlrSF best32. ;
format X2ndFlrSF best32. ;
format LowQualFinSF best32. ;
format GrLivArea best32. ;
format BsmtFullBath best32. ;
format BsmtHalfBath best32. ;
format FullBath best32. ;
format HalfBath best32. ;
format BedroomAbvGr best32. ;
format KitchenAbvGr best32. ;
format KitchenQual $20. ;
format TotRmsAbvGrd best32. ;
format Functional $20. ;
format Fireplaces best32. ;
format FireplaceQu $20. ;
format GarageType $20. ;
format GarageYrBlt best32. ;
format GarageFinish $20. ;
format GarageCars best32. ;
format GarageArea best32. ;
format GarageQual $20. ;
format GarageCond $20. ;
format PavedDrive $20. ;
format WoodDeckSF best32. ;
format OpenPorchSF best32. ;
format EnclosedPorch best32. ;
format X3SsnPorch best32. ;
format ScreenPorch best32. ;
format PoolArea best32. ;
format PoolQC $20. ;
format Fence $20. ;
format MiscFeature $20. ;
format MiscVal best32. ;
format MoSold best32. ;
format YrSold best32. ;
format SaleType $20. ;
format SaleCondition $20. ;
format SalePrice best32. ;
input
Id
MSSubClass
MSZoning $
LotFrontage
LotArea
Street $
Alley $
LotShape $
```

```
LandContour $
Utilities $
LotConfig $
LandSlope $
Neighborhood $
Condition1 $
Condition2 $
BldgType $
HouseStyle $
OverallQual
OverallCond
YearBuilt
YearRemodAdd
RoofStyle $
RoofMatl $
Exterior1st $
Exterior2nd $
MasVnrType $
MasVnrArea
ExterQual $
ExterCond $
Foundation $
BsmtQual $
BsmtCond $
BsmtExposure $
BsmtFinType1 $
BsmtFinSF1
BsmtFinType2 $
BsmtFinSF2
BsmtUnfSF
TotalBsmtSF
Heating $
HeatingQC $
CentralAir $
Electrical $
X1stFlrSF
X2ndFlrSF
LowQualFinSF
GrLivArea
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
KitchenQual $
TotRmsAbvGrd
Functional $
Fireplaces
FireplaceQu $
GarageType $
GarageYrBlt
GarageFinish $
GarageCars
GarageArea
GarageQual $
GarageCond $
PavedDrive $
WoodDeckSF
OpenPorchSF
EnclosedPorch
X3SsnPorch
ScreenPorch
PoolArea
PoolQC $
Fence $
MiscFeature $
MiscVal
MoSold
YrSold
SaleType $
SaleCondition $
SalePrice
;
if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection macro variable */
```

```
run;


/* Import test data */
data WORK.TEST;
%let _EFIERR_ = 0; /* set the ERROR detection macro variable */
infile '/home/jdstroud0/Stats2/AmesHousing/Version1/test.csv' delimiter = ',' MISSOVER DSD lrecl=32767 firstobs=2 ;  /* <---- YOUR PATH HERE */
informat Id best32. ;
informat MSSubClass best32. ;
informat MSZoning $20. ;
informat LotFrontage best32. ;
informat LotArea best32. ;
informat Street $20. ;
informat Alley $20. ;
informat LotShape $20. ;
informat LandContour $20. ;
informat Utilities $20. ;
informat LotConfig $20. ;
informat LandSlope $20. ;
informat Neighborhood $20. ;
informat Condition1 $$20. ;
informat Condition2 $20. ;
informat BldgType $20. ;
informat HouseStyle $20. ;
informat OverallQual best32. ;
informat OverallCond best32. ;
informat YearBuilt best32. ;
informat YearRemodAdd best32. ;
informat RoofStyle $20. ;
informat RoofMatl $20. ;
informat Exterior1st $20. ;
informat Exterior2nd $20. ;
informat MasVnrType $20. ;
informat MasVnrArea best32. ;
informat ExterQual $20. ;
informat ExterCond $$20. ;
informat Foundation $20. ;
informat BsmtQual $20. ;
informat BsmtCond $20. ;
informat BsmtExposure $20. ;
informat BsmtFinType1 $20. ;
informat BsmtFinSF1 best32. ;
informat BsmtFinType2 $20.;
informat BsmtFinSF2 best32. ;
informat BsmtUnfSF best32. ;
informat TotalBsmtSF best32. ;
informat Heating $20. ;
informat HeatingQC $20. ;
informat CentralAir $20. ;
informat Electrical $20. ;
informat X1stFlrSF best32. ;
informat X2ndFlrSF best32. ;
informat LowQualFinSF best32. ;
informat GrLivArea best32. ;
informat BsmtFullBath best32. ;
informat BsmtHalfBath best32. ;
informat FullBath best32. ;
informat HalfBath best32. ;
informat BedroomAbvGr best32. ;
informat KitchenAbvGr best32. ;
informat KitchenQual $20. ;
informat TotRmsAbvGrd best32. ;
informat Functional $20. ;
informat Fireplaces best32. ;
informat FireplaceQu $20. ;
informat GarageType $20. ;
informat GarageYrBlt best32. ;
informat GarageFinish $20. ;
informat GarageCars best32. ;
informat GarageArea best32. ;
informat GarageQual $20. ;
informat GarageCond $20. ;
informat PavedDrive $20. ;
informat WoodDeckSF best32. ;
informat OpenPorchSF best32. ;
```

```
informat EnclosedPorch best32. ;
informat X3SsnPorch best32. ;
informat ScreenPorch best32. ;
informat PoolArea best32. ;
informat PoolQC $20. ;
informat Fence $20. ;
informat MiscFeature $20. ;
informat MiscVal best32. ;
informat MoSold best32. ;
informat YrSold best32. ;
informat SaleType $20. ;
informat SaleCondition $20. ;
format Id best32. ;
format MSSubClass best32. ;
format MSZoning $20. ;
format LotFrontage best32. ;
format LotArea best32. ;
format Street $20. ;
format Alley $20. ;
format LotShape $20. ;
format LandContour $20.;
format Utilities $20. ;
format LotConfig $20. ;
format LandSlope $20. ;
format Neighborhood $20. ;
format Condition1 $20. ;
format Condition2 $20. ;
format BldgType $20. ;
format HouseStyle $20. ;
format OverallQual best32. ;
format OverallCond best32. ;
format YearBuilt best32. ;
format YearRemodAdd best32. ;
format RoofStyle $20. ;
format RoofMatl $20. ;
format Exterior1st $20. ;
format Exterior2nd $20. ;
format MasVnrType $20. ;
format MasVnrArea best32. ;
format ExterQual $20. ;
format ExterCond $20. ;
format Foundation $20. ;
format BsmtQual $20. ;
format BsmtCond $20. ;
format BsmtExposure $20. ;
format BsmtFinType1 $20. ;
format BsmtFinSF1 best32. ;
format BsmtFinType2 $20. ;
format BsmtFinSF2 best32. ;
format BsmtUnfSF best32. ;
format TotalBsmtSF best32. ;
format Heating $20. ;
format HeatingQC $20. ;
format CentralAir $20. ;
format Electrical $20. ;
format X1stFlrSF best32. ;
format X2ndFlrSF best32. ;
format LowQualFinSF best32. ;
format GrLivArea best32. ;
format BsmtFullBath best32. ;
format BsmtHalfBath best32. ;
format FullBath best32. ;
format HalfBath best32. ;
format BedroomAbvGr best32. ;
format KitchenAbvGr best32. ;
format KitchenQual $20. ;
format TotRmsAbvGrd best32. ;
format Functional $20. ;
format Fireplaces best32. ;
format FireplaceQu $20. ;
format GarageType $20. ;
format GarageYrBlt best32. ;
format GarageFinish $20. ;
format GarageCars best32. ;
format GarageArea best32. ;
```

```
format GarageQual $20. ;
format GarageCond $20. ;
format PavedDrive $20. ;
format WoodDeckSF best32. ;
format OpenPorchSF best32. ;
format EnclosedPorch best32. ;
format X3SsnPorch best32. ;
format ScreenPorch best32. ;
format PoolArea best32. ;
format PoolQC $20. ;
format Fence $20. ;
format MiscFeature $20. ;
format MiscVal best32. ;
format MoSold best32. ;
format YrSold best32. ;
format SaleType $20. ;
format SaleCondition $20. ;
input
Id
MSSubClass
MSZoning $
LotFrontage
LotArea
Street $
Alley $
LotShape $
LandContour $
Utilities $
LotConfig $
LandSlope $
Neighborhood $
Condition1 $
Condition2 $
BldgType $
HouseStyle $
OverallQual
OverallCond
YearBuilt
YearRemodAdd
RoofStyle $
RoofMatl $
Exterior1st $
Exterior2nd $
MasVnrType $
MasVnrArea
ExterQual $
ExterCond $
Foundation $
BsmtQual $
BsmtCond $
BsmtExposure $
BsmtFinType1 $
BsmtFinSF1
BsmtFinType2 $
BsmtFinSF2
BsmtUnfSF
TotalBsmtSF
Heating $
HeatingQC $
CentralAir $
Electrical $
X1stFlrSF
X2ndFlrSF
LowQualFinSF
GrLivArea
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
KitchenQual $
TotRmsAbvGrd
Functional $
Fireplaces
```

```
FireplaceQu $
GarageType $
GarageYrBlt
GarageFinish $
GarageCars
GarageArea
GarageQual $
GarageCond $
PavedDrive $
WoodDeckSF
OpenPorchSF
EnclosedPorch
X3SsnPorch
ScreenPorch
PoolArea
PoolQC $
Fence $
MiscFeature $
MiscVal
MoSold
YrSold
SaleType $
SaleCondition $
;
if _ERROR_ then call symputx('_EFIERR_',1);  /* set ERROR detection macro variable */
run;

/* Outlier Removal  */
data train;
set train;
    IF GrLivArea  > 4000 THEN DELETE;
  RUN;

  /* Combine Data After Outlier Removal */
data house;
   set train test;
run;
/* Nice little check sum to see that our data has values */
/* proc means data=house NMISS N; run; */

/* Set categorical variables to numeric */

data house;
set house;
IF Alley = "Pave" THEN cAlley = 1;
IF Alley ^= "Pave" THEN cAlley = 0;
IF BldgType = "1Fam" OR "TwnhsE" THEN cBldgType = 1;
IF BldgType ^= "1Fam" OR "TwnhsE" THEN cBldgType = 1;
IF BsmtCond = "Ex" THEN cBsmtCond = 5;
IF BsmtCond = "Gd" THEN cBsmtCond = 4;
IF BsmtCond = "TA" THEN cBsmtCond = 3;
IF BsmtCond = "Fa" THEN cBsmtCond = 2;
IF BsmtCond = "NA" THEN cBsmtCond = 1;
IF BsmtCond = "Po" THEN cBsmtCond = 0;
IF BsmtCond ^="Ex" OR "Gd" OR "TA" OR "Fa" OR "NA" OR "Po" THEN cBsmtCond = 1;
IF BsmtExposure = "Gd" THEN cBsmtExposure = 3;
IF BsmtExposure = "Av" THEN cBsmtExposure = 2;
IF BsmtExposure = "Mn" THEN cBsmtExposure = 1;
IF BsmtExposure = "No" THEN cBsmtExposure = 0;
IF BsmtExposure = "NA" THEN cBsmtExposure = 0;
IF BsmtExposure ^="Gd" OR "Av" OR "Mn" OR "No" OR "NA" THEN cBsmtExposure = 0;
IF BsmtFinType1 = "GLQ" THEN cBsmtFinType1 = 5;
IF BsmtFinType1 = "Unf" THEN cBsmtFinType1 = 4;
IF BsmtFinType1 = "ALQ" THEN cBsmtFinType1 = 3;
IF BsmtFinType1 = "BLQ" THEN cBsmtFinType1 = 2;
IF BsmtFinType1 = "Rec" THEN cBsmtFinType1 = 2;
IF BsmtFinType1 = "LwQ" THEN cBsmtFinType1 = 2;
IF BsmtFinType1 = "NA" THEN cBsmtFinType1 = 1;
IF BsmtFinType1 ^="GLQ" OR "Unf" OR "ALQ" OR "BLQ" OR "Rec" OR "LwQ" OR "NA" THEN cBsmtFinType1 = 1;
IF BsmtFinType2 = "GLQ" THEN cBsmtFinType2 = 6;
IF BsmtFinType2 = "Unf" THEN cBsmtFinType2 = 5;
IF BsmtFinType2 = "ALQ" THEN cBsmtFinType2 = 4;
IF BsmtFinType2 = "Rec" THEN cBsmtFinType2 = 3;
IF BsmtFinType2 = "LwQ" THEN cBsmtFinType2 = 3;
IF BsmtFinType2 = "BLQ" THEN cBsmtFinType2 = 2;
```

```
IF BsmtFinType2 = "NA" THEN cBsmtFinType2 = 1;
IF BsmtFinType2 ^="GLQ" OR "Unf" OR "ALQ" OR "BLQ" OR "Rec" OR "LwQ" OR "NA" THEN cBsmtFinType2 = 1;
IF BsmtQual = "Ex" THEN cBsmtQual = 5;
IF BsmtQual = "Gd" THEN cBsmtQual = 4;
IF BsmtQual = "TA" THEN cBsmtQual = 3;
IF BsmtQual = "Fa" THEN cBsmtQual = 2;
IF BsmtQual = "Po" THEN cBsmtQual = 1;
IF BsmtQual = "NA" THEN cBsmtQual = 0;
IF BsmtQual ^="Ex" OR "Gd" OR "TA" OR "Fa" OR "Po" OR "NA" THEN cBsmtQual = 0;
IF CentralAir = "Y" THEN cCentralAir = 1;
IF CentralAir ^= "Y" THEN cCentralAir = 0;
IF Condition1 = "PosN" OR "PosA" THEN cCondition1 = 1;
IF Condition1 ^= "PosN" OR "PosA" THEN cCondition1 = 0;
IF Condition2 = "PosN" OR "PosA" THEN cCondition2 = 1;
IF Condition2 ^= "PosN" OR "PosA" THEN cCondition2 = 0;
IF Electrical = "SBrkr" THEN cElectrical = 1;
IF Electrical ^= "SBrkr" THEN cElectrical = 0;
IF ExterCond = "Ex" THEN cExterCond = 5;
IF ExterCond = "Gd" THEN cExterCond = 4;
IF ExterCond = "TA" THEN cExterCond = 3;
IF ExterCond = "Fa" THEN cExterCond = 2;
IF ExterCond = "Po" THEN cExterCond = 1;
IF ExterQual = "Ex" THEN cExterQual = 5;
IF ExterQual = "Gd" THEN cExterQual = 4;
IF ExterQual = "TA" THEN cExterQual = 3;
IF ExterQual = "Fa" THEN cExterQual = 2;
IF ExterQual = "Po" THEN cExterQual = 1;
IF Fence = "GdPrv" THEN cFence = 2;
IF Fence = "MnPrv" THEN cFence = 1;
IF Fence = "GdWo" THEN cFence = 2;
IF Fence = "MnWw" THEN cFence = 1;
IF Fence = "NA" THEN cFence = 0;
IF FireplaceQu = "Ex" THEN cFireplaceQu = 5;
IF FireplaceQu = "Gd" THEN cFireplaceQu = 4;
IF FireplaceQu = "TA" THEN cFireplaceQu = 3;
IF FireplaceQu = "Fa" THEN cFireplaceQu = 2;
IF FireplaceQu = "Po" THEN cFireplaceQu = 1;
IF FireplaceQu = "NA" THEN cFireplaceQu = 0;
IF Foundation = "PConc" THEN cFoundation = 1;
IF Foundation = "BrkTil" THEN cFoundation = 0;
IF Foundation = "CBlock" THEN cFoundation = 0;
IF Foundation = "Slab" THEN cFoundation = 0;
IF Foundation = "Stone" THEN cFoundation = 0;
IF Foundation = "Wood" THEN cFoundation = 0;
IF Functional = "Typ" THEN cFunctional = 0;
IF Functional = "Mod" THEN cFunctional = 2;
IF Functional = "Min1" THEN cFunctional = 1;
IF Functional = "Min2" THEN cFunctional = 1;
IF Functional = "NA" THEN cFunctional = 2;
IF Functional = "Maj1" THEN cFunctional = 3;
IF Functional = "Maj2" THEN cFunctional = 4;
IF Functional = "Sev" THEN cFunctional = 5;
IF Functional = "Sal" THEN cFunctional = 6;
IF GarageCond = "Ex" THEN cGarageCond = 5;
IF GarageCond = "Gd" THEN cGarageCond = 4;
IF GarageCond = "TA" THEN cGarageCond = 3;
IF GarageCond = "Fa" THEN cGarageCond = 2;
IF GarageCond = "Po" THEN cGarageCond = 1;
IF GarageCond = "NA" THEN cGarageCond = 0;
IF GarageCond ^= "Ex" OR "Gd" OR "TA" OR "Fa" OR "Po" OR "NA" THEN cGarageCond = 0;
IF GarageFinish = "Fin" OR "RFn" THEN cGarageFinish = 1;
IF GarageFinish ^= "Fin" OR "RFn" THEN cGarageFinish = 0;
IF GarageQual = "Ex" THEN cGarageQual = 5;
IF GarageQual = "Gd" THEN cGarageQual = 4;
IF GarageQual = "TA" THEN cGarageQual = 3;
IF GarageQual = "Fa" THEN cGarageQual = 2;
IF GarageQual = "Po" THEN cGarageQual = 1;
IF GarageQual = "NA" THEN cGarageQual = 0;
IF GarageQual ^= "Ex" OR "Gd" OR "TA" OR "Fa" OR "Po" OR "NA" THEN cGarageQual = 0;
IF GarageType = "Attchd" THEN cGarageType = 1;
IF GarageType = "BuiltIn" THEN cGarageType = 1;
IF GarageType = "2Types" THEN cGarageType = 0;
IF GarageType = "Basment" THEN cGarageType = 0;
IF GarageType = "CarPort" THEN cGarageType = 0;
IF GarageType = "Detchd" THEN cGarageType = 0;
```

```
IF GarageType = "NA" THEN cGarageType = 0;
IF Heating = "GasA" THEN cHeating = 1;
IF Heating = "GasW" THEN cHeating = 1;
IF Heating = "Floor" THEN cHeating = 0;
IF Heating = "Grav" THEN cHeating = 0;
IF Heating = "OthW" THEN cHeating = 0;
IF Heating = "Wall" THEN cHeating = 0;
IF HeatingQC = "Ex" THEN cHeatingQC = 5;
IF HeatingQC = "Gd" THEN cHeatingQC = 4;
IF HeatingQC = "TA" THEN cHeatingQC = 3;
IF HeatingQC = "Fa" THEN cHeatingQC = 2;
IF HeatingQC = "Po" THEN cHeatingQC = 1;
IF KitchenQual = "Ex" THEN cKitchenQual = 5;
IF KitchenQual = "Gd" THEN cKitchenQual = 4;
IF KitchenQual = "TA" THEN cKitchenQual = 3;
IF KitchenQual = "Fa" THEN cKitchenQual = 2;
IF KitchenQual = "Po" THEN cKitchenQual = 1;
IF KitchenQual = "NA" THEN cKitchenQual = 3;
IF LandContour = "Lvl" THEN cLandContour = 1;
IF LandContour = "Bnk" THEN cLandContour = 0;
IF LandContour = "HLS" THEN cLandContour = 0;
IF LandContour = "Low" THEN cLandContour = 0;
IF LandSlope = "Gtl" THEN cLandSlope = 1;
IF LandSlope = "Mod" THEN cLandSlope = 0;
IF LandSlope = "Sev" THEN cLandSlope = 0;
IF LotConfig = "CulDSac" THEN cLotConfig = 1;
IF LotConfig = "FR3" THEN cLotConfig = 1;
IF LotConfig = "Inside" THEN cLotConfig = 0;
IF LotConfig = "Corner" THEN cLotConfig = 0;
IF LotConfig = "FR2" THEN cLotConfig = 0;
IF LotShape = "Reg" THEN cLotShape = 1;
IF LotShape = "IR1" THEN cLotShape = 0;
IF LotShape = "IR2" THEN cLotShape = 0;
IF LotShape = "IR3" THEN cLotShape = 0;
IF MasVnrType = "BrkFace" OR "Stone" OR "BrkCmn" OR "CBlock" THEN cMasVnrType = 1;
IF MasVnrType ^= "BrkFace" OR "Stone" OR "BrkCmn" OR "CBlock" THEN cMasVnrType = 0;
IF MiscFeature = "TenC" THEN cMiscFeature = 1;
IF MiscFeature = "Elev" THEN cMiscFeature = 0;
IF MiscFeature = "Gar2" THEN cMiscFeature = 0;
IF MiscFeature = "Othr" THEN cMiscFeature = 0;
IF MiscFeature = "Shed" THEN cMiscFeature = 0;
IF MiscFeature = "NA" THEN cMiscFeature = 0;
IF MSSubClass = 20 THEN cMSSubClass = 1;
IF MSSubClass = 30 THEN cMSSubClass = 0;
IF MSSubClass = 40 THEN cMSSubClass = 0;
IF MSSubClass = 45 THEN cMSSubClass = 0;
IF MSSubClass = 50 THEN cMSSubClass = 0;
IF MSSubClass = 60 THEN cMSSubClass = 1;
IF MSSubClass = 70 THEN cMSSubClass = 0;
IF MSSubClass = 75 THEN cMSSubClass = 0;
IF MSSubClass = 80 THEN cMSSubClass = 0;
IF MSSubClass = 85 THEN cMSSubClass = 0;
IF MSSubClass = 90 THEN cMSSubClass = 0;
IF MSSubClass = 120 THEN cMSSubClass = 1;
IF MSSubClass = 150 THEN cMSSubClass = 0;
IF MSSubClass = 160 THEN cMSSubClass = 0;
IF MSSubClass = 180 THEN cMSSubClass = 0;
IF MSSubClass = 190 THEN cMSSubClass = 0;
IF MSZoning = "FV" THEN cMSZoning = 4;
IF MSZoning = "RL" THEN cMSZoning = 3;
IF MSZoning = "RP" THEN cMSZoning = 3;
IF MSZoning = "RH" THEN cMSZoning = 2;
IF MSZoning = "RM" THEN cMSZoning = 2;
IF MSZoning = "NA" THEN cMSZoning = 2;
IF MSZoning = "A" THEN cMSZoning = 1;
IF MSZoning = "C (all)" THEN cMSZoning = 1;
IF MSZoning = "I" THEN cMSZoning = 1;
IF MSZoning ^= "FV" OR "RL" OR "RP" OR "RH" OR "RM" OR "NA" OR "A" OR "C (all)" OR "I" THEN cMSZoning = 1;
IF PavedDrive = "Y" THEN cPavedDrive = 1;
IF PavedDrive ^= "Y" THEN cPavedDrive = 0;
IF PoolQC = "Ex" THEN cPoolQC = 1;
IF PoolQC = "Gd" THEN cPoolQC = 0;
IF PoolQC = "TA" THEN cPoolQC = 0;
IF PoolQC = "Fa" THEN cPoolQC = 0;
IF PoolQC = "NA" THEN cPoolQC = 0;
```

```
IF RoofMatl = "Membran" THEN cRoofMatl = 1;
IF RoofMatl = "WdShake" THEN cRoofMatl = 1;
IF RoofMatl = "WdShngl" THEN cRoofMatl = 1;
IF RoofMatl = "ClyTile" THEN cRoofMatl = 0;
IF RoofMatl = "CompShg" THEN cRoofMatl = 0;
IF RoofMatl = "Metal" THEN cRoofMatl = 0;
IF RoofMatl = "Roll" THEN cRoofMatl = 0;
IF RoofMatl = "Tar&Grv" THEN cRoofMatl = 0;
IF RoofStyle = "Hip" THEN cRoofStyle = 1;
IF RoofStyle = "Shed" THEN cRoofStyle = 1;
IF RoofStyle = "Flat" THEN cRoofStyle = 0;
IF RoofStyle = "Gable" THEN cRoofStyle = 0;
IF RoofStyle = "Gambrel" THEN cRoofStyle = 0;
IF RoofStyle = "Mansard" THEN cRoofStyle = 0;
IF SaleCondition = "Partial" THEN cSaleCondition = 4;
IF SaleCondition = "Normal" THEN cSaleCondition = 3;
IF SaleCondition = "Alloca" THEN cSaleCondition = 3;
IF SaleCondition = "Abnorml" THEN cSaleCondition = 2;
IF SaleCondition = "Family" THEN cSaleCondition = 2;
IF SaleCondition = "AdjLand" THEN cSaleCondition = 1;
IF SaleType = "New" THEN cSaleType = 5;
IF SaleType = "Con" THEN cSaleType = 5;
IF SaleType = "CWD" THEN cSaleType = 4;
IF SaleType = "ConLI" THEN cSaleType = 4;
IF SaleType = "WD" THEN cSaleType = 3;
IF SaleType = "NA" THEN cSaleType = 3;
IF SaleType = "VWD" THEN cSaleType = 3;
IF SaleType = "COD" THEN cSaleType = 2;
IF SaleType = "ConLw" THEN cSaleType = 2;
IF SaleType = "ConLD" THEN cSaleType = 2;
IF SaleType = "Oth" THEN cSaleType = 1;
IF Street = "Pave" THEN cStreet = 1;
IF Street = "Grvl" THEN cStreet = 0;
IF Utilities = "AllPub" THEN cUtilities = 1;
IF Utilities = "NoSewr" THEN cUtilities = 0;
IF Utilities = "NoSeWa" THEN cUtilities = 0;
IF Utilities = "ELO" THEN cUtilities = 0;
IF Utilities = "NA" THEN cUtilities = 0;
RUN;


/* Clean up Lot Frontage missing values */
data house;
  set house;
  array change _numeric_;
      do over change;
         if LotFrontage=. then LotFrontage=LotArea*0.0078;
      end;
 run ;

 /* Clean up missing values  */
/* Replace Garage Year Built missing values with House Year Built  */
data house;
  set house;
  array change _numeric_;
      do over change;
         if GarageYrBlt=. then GarageYrBlt=YearBuilt;
      end;
 run ;

/* Replace missing masonry veneer area with zero  */
data house;
  set house;
  array change _numeric_;
      do over change;
         if MasVnrArea=. then MasVnrArea=0;
      end;
 run ;

 /* Replace missing values  */
data house;
  set house;
  array change _numeric_;
      do over change;
         if BsmtFinSF1=. then BsmtFinSF1=0;
```

```
        if BsmtFinSF2=. then BsmtFinSF2=0;
        if BsmtUnfSF=. then BsmtUnfSF=0;
        if TotalBsmtSF=. then TotalBsmtSF=0;
        if BsmtFullBath=. then BsmtFullBath=0;
        if BsmtHalfBath=. then BsmtHalfBath=0;
        if GarageCars=. then GarageCars=0;
        if GarageArea=. then GarageArea=0;
      end;
run ;

/* We were trying to do some special things with these fields such as divide by averages, but had to revert to rank values  */
data house;
set house;
IF Neighborhood="NoRidge" THEN cNeighborhood = 1;
IF Neighborhood="NridgHt" THEN cNeighborhood = 1;
IF Neighborhood="StoneBr" THEN cNeighborhood = 1;
IF Neighborhood^="StoneBr" OR "NridgHt" OR "NoRidge" THEN cNeighborhood = 0;

IF Exterior1st="ImStucc" THEN cExterior1st = 2;
IF Exterior1st="Stone" THEN cExterior1st = 2;
IF Exterior1st="CemntBd" THEN cExterior1st = 2;
IF Exterior1st="VinylSd" THEN cExterior1st = 2;
IF Exterior1st="BrkFace" THEN cExterior1st = 2;
IF Exterior1st="Plywood" THEN cExterior1st = 1;
IF Exterior1st="HdBoard" THEN cExterior1st = 1;
IF Exterior1st="Stucco" THEN cExterior1st = 1;
IF Exterior1st="NA" THEN cExterior1st = 0;
IF Exterior1st="WdShing" THEN cExterior1st = 1;
IF Exterior1st="Wd Sdng" THEN cExterior1st = 1;
IF Exterior1st="MetalSd" THEN cExterior1st = 1;
IF Exterior1st="AsbShng" THEN cExterior1st = 0;
IF Exterior1st="CBlock" THEN cExterior1st = 0;
IF Exterior1st="AsphShn" THEN cExterior1st = 0;
IF Exterior1st="BrkComm" THEN cExterior1st = 0;

IF Exterior2nd="Other" THEN cExterior2nd = 2;
IF Exterior2nd="ImStucc" THEN cExterior2nd = 2;
IF Exterior2nd="Stone" THEN cExterior2nd = 1;
IF Exterior2nd="CmentBd" THEN cExterior2nd = 2;
IF Exterior2nd="VinylSd" THEN cExterior2nd = 2;
IF Exterior2nd="BrkFace" THEN cExterior2nd = 2;
IF Exterior2nd="Plywood" THEN cExterior2nd = 1;
IF Exterior2nd="HdBoard" THEN cExterior2nd = 1;
IF Exterior2nd="Stucco" THEN cExterior2nd = 1;
IF Exterior2nd="WdShing" THEN cExterior2nd = 1;
IF Exterior2nd="Wd Shng" THEN cExterior2nd = 1;
IF Exterior2nd="Wd Sdng" THEN cExterior2nd = 1;
IF Exterior2nd="MetalSd" THEN cExterior2nd = 1;
IF Exterior2nd="Wd Sdng" THEN cExterior2nd = 1;
IF Exterior2nd="NA" THEN cExterior2nd = 1;
IF Exterior2nd="AsphShn" THEN cExterior2nd = 1;
IF Exterior2nd="AsbShng" THEN cExterior2nd = 1;
IF Exterior2nd="CBlock" THEN cExterior2nd = 1;
IF Exterior2nd="AsphShn" THEN cExterior2nd = 1;
IF Exterior2nd="Brk Cmn" THEN cExterior2nd = 1;

IF HouseStyle="2.5Fin" THEN cHouseStyle = 220000/100000;
IF HouseStyle="2Story" THEN cHouseStyle = 210051/100000;
IF HouseStyle="1Story" THEN cHouseStyle = 175985/100000;
IF HouseStyle="SLvl" THEN cHouseStyle = 166703/100000;
IF HouseStyle="2.5Unf" THEN cHouseStyle = 157354/100000;
IF HouseStyle="1.5Fin" THEN cHouseStyle = 143116/100000;
IF HouseStyle="SFoyer" THEN cHouseStyle = 135074/100000;
IF HouseStyle="1.5Unf" THEN cHouseStyle = 110150/100000;
RUN;


/* Nice little check sum to see that our data has values */
/* proc means data=house NMISS N; run;  */

/* Create new correlated variables */
data house;
set house;
BRBR = log(((FullBath + HalfBath)+1) / (1+BedroomAbvGr));
GLARooms = log(1+(GrLivArea / TotRmsAbvGrd));
```

```
GQGC = log(1+(cGarageQual * GarageCars));
/*  GRGR = GrLivArea Placeholder for code below instead of using ALTER TABLE ADD */
OQEC = log(1+(OverallQual * cExterCond));
OQFB  = log(1+(OverallQual * FullBath));
OQGLA = log(1+(OverallQual * GrLivArea));
OQTBSF = log(1+(OverallQual * TotalBsmtSF));
OQYB = log(1+(OverallQual * YearBuilt));
OQYR = log(1+(OverallQual * YearRemodAdd));
RUN;
/* GRGR Correlated variable based on macro variable (GrLivArea / mean(GrLivArea))  */
PROC SQL noprint;
ALTER TABLE house ADD GRGR num informat=best32. format=best32.;
SELECT mean(GrLivArea) INTO :n FROM house;
UPDATE house SET GRGR=log(1+(GrLivArea/&n));
QUIT;

/* Clear out the unused variables   */
proc sql noprint;
ALTER TABLE house DROP COLUMN Alley, BldgType, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2, BsmtQual, CentralAir,
Condition1, Condition2, Electrical, ExterCond,ExterQual,Fence,FireplaceQu,Foundation,Functional,GarageCond,GarageFinish,
GarageQual,GarageType,Heating,HeatingQC,KitchenQual,LandContour,LandSlope,LotConfig, LotShape,MasVnrType,MiscFeature,
MSSubClass, MSZoning,PavedDrive,PoolQC, RoofMatl, RoofStyle, SaleCondition, SaleType, Street, Utilities, Exterior1st, Exterior2nd,
HouseStyle, Neighborhood;
quit;

/* Take the logs of the continuous variables   */
data house;
set house;
logSalePrice = log(1+SalePrice);
logX1stFlrSF = log(1+X1stFlrSF);
logX2ndFlrSF = log(1+X2ndFlrSF);
logX3SsnPorch = log(1+X3SsnPorch);
logBsmtFinSF1 = log(1+BsmtFinSF1);
logBsmtFinSF2 = log(1+BsmtFinSF2);
logBsmtUnfSF = log(1+BsmtUnfSF);
logEnclosedPorch = log(1+EnclosedPorch);
logGarageArea = log(1+GarageArea);
logGrLivArea = log(1+GrLivArea);
logLotArea = log(1+LotArea);
logLotFrontage = log(1+LotFrontage);
logLowQualFinSF = log(1+LowQualFinSF);
logMasVnrArea = log(1+MasVnrArea);
logMiscVal = log(1+MiscVal);
logOpenPorchSF = log(1+OpenPorchSF);
logPoolArea = log(1+PoolArea);
logScreenPorch = log(1+ScreenPorch);
logTotalBsmtSF = log(1+ScreenPorch);
logWoodDeckSF = log(1+WoodDeckSF);
logYear=log(YearBuilt);
logYrRemod = log(YearRemodAdd);
run;

/* Normalize years  */
data house;
set house;
YrSold = 2010-YrSold;
run;

/* proc means data=house NMISS N; */


/* We ended up addressing a few more last minute outliers and also stacking terms here  */
data house_out;
set house;
if GrLivArea>4000 AND logSalePrice>1 then delete;
if GrLivArea=5642 then delete;
if GrLivArea=4676 then delete;
if LotArea=159000 then delete;
if LotArea=215245 then delete;
if LotArea=164660 then delete;
if LotArea=115149 then delete;
if TotalBsmtSF=6110 then delete;
sqOverallCond=OverallCond**2;
sqGarageCars=GarageCars**2;
sqLotArea=logLotArea**2;
```

```
stot=(logGrLivArea+cFunctional)**2;
ctot=(logGrLivArea+cFunctional)**3;
run;

/*My method
ods graphics on;
proc glmselect data=house_out
        seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot CRITERIONPANEL);
model logSalePrice = OverallQual logYear logYrRemod MasVnrArea BsmtFinSF1 BsmtFinSF1 BsmtFinSF2 BsmtFullBath KitchenAbvGr Fireplaces
WoodDeckSF EnclosedPorch YrSold cBsmtExposure cBsmtQual cCentralAir cExterQual cFoundation cFunctional|cCentralAir
cHeatingQC cKitchenQual cMSZoning cPavedDrive cSaleCondition cStreet cNeighborhood cExterior1st cExterior2nd logBsmtFinSF1
logBsmtFinSF2 X1stFlrSF stot logGarageArea logLotArea logLowQualFinSF logScreenPorch BRBR GQGC OQTBSF GRGR sqGarageCars
cExterior1st|MasVnrArea/
 selection=Forward(choose=CV stop=CV) CVdetails;
run;
quit;
ods graphics off;
*/


/* Model output code block. Cut paste variables from your testing output into the model below  */
/*proc glm data=house_out plots=all;
model logSalePrice =
OverallQual OQYB|OverallCond OQYR|OverallCond YearBuilt YearRemodAdd MasVnrArea OQYB|BsmtFinSF1 OQYR|BsmtFinSF1 BsmtFinSF2
BsmtFullBath KitchenAbvGr Fireplaces
WoodDeckSF EnclosedPorch YrSold cBsmtExposure cBsmtQual cCentralAir OQYB|cExterCond OQYR|cExterCond cExterQual cFoundation
cFunctional|cCentralAir
cHeatingQC cKitchenQual cMSZoning cPavedDrive cSaleCondition cStreet cNeighborhood cExterior1st cExterior2nd logBsmtFinSF1
logBsmtFinSF2 logBsmtUnfSF X1stFlrSF stot ctot logGarageArea logLotArea logLowQualFinSF logScreenPorch BRBR GQGC OQTBSF GRGR
sqGarageCars
cExterior1st|MasVnrArea
/solution cli clparm;
output out=results predicted=logpredict;
run;*/
data kaggle_out (KEEP=id SalePrice);
  set results6;
  array change _numeric_;
      do over change;
         if SalePrice=. then SalePrice=exp(logpredict);
      end;
 run ;

/* Clean up for Kaggle formatted output   */
proc sql noprint;
DELETE FROM kaggle_out WHERE ID<1461;
quit;

proc print data=kaggle_out;run;

/* Export to file */
proc export data=kaggle_out
outfile='/home/jdstroud0/Stats2/AmesHousing/Primary/final.csv'    /* <---- YOUR PATH HERE */
dbms=csv replace;
delimiter=',';
run;
```

Kaggle Score .11840 Code Dump

.11840 R Data Cleanup Code:

```
train <- read.csv("C:/Users/flip/Desktop/SMU/MSDS6371/group project/train.csv", stringsAsFactors=FALSE)
test <- read.csv("C:/Users/flip/Desktop/SMU/MSDS6371/group project/test.csv", stringsAsFactors=FALSE)

str(train)
str(test)


# #Outlier removal
# train <- train [!(house$GrLivArea > 4000),]

test$SalePrice <- 0
house <- rbind(train,test)
str(house)
names(house)

# library(DT)
```

```
# datatable(head(house, n=20),options = list(scrollX = TRUE))

# library(VIM)
# aggr(house, prop = F, numbers = T)
apply(is.na(house),2,sum)

library(dplyr)
summarize(group_by(house, Alley), mean(SalePrice, na.rm=T))
house$cAlley[house$Alley %in% c("Grvl")] <- 1
house$cAlley[!house$Alley %in% c("Grvl")] <- 0
summarize(group_by(house, cAlley), mean(SalePrice, na.rm=T))

table(house$BldgType)
house$cBldgType[house$BldgType %in% c("1Fam", "TwnhsE")] <- 1
house$cBldgType[!house$BldgType %in% c("1Fam", "TwnhsE")] <- 0
table(house$cBldgType)

summarize(group_by(house, BsmtCond), mean(SalePrice, na.rm=T))
house$cBsmtCond[house$BsmtCond == "Ex"] <- 5
house$cBsmtCond[house$BsmtCond == "Gd"] <- 4
house$cBsmtCond[house$BsmtCond == "TA"] <- 3
house$cBsmtCond[house$BsmtCond == "Fa"] <- 2
house$cBsmtCond[is.na(house$BsmtCond)] <- 1
house$cBsmtCond[house$BsmtCond == "Po"] <- 0
summarize(group_by(house, cBsmtCond), mean(SalePrice, na.rm=T))

summarize(group_by(house, BsmtExposure), mean(SalePrice, na.rm=T))
house$cBsmtExposure[house$BsmtExposure == "Gd"] <- 3
house$cBsmtExposure[house$BsmtExposure == "Av"] <- 2
house$cBsmtExposure[house$BsmtExposure == "Mn"] <- 1
house$cBsmtExposure[house$BsmtExposure == "No"] <- 0
house$cBsmtExposure[is.na(house$BsmtExposure)] <- 0
summarize(group_by(house, cBsmtExposure), mean(SalePrice, na.rm=T))

summarize(group_by(house, BsmtFinType1), mean(SalePrice, na.rm=T))
house$cBsmtFinType1[house$BsmtFinType1 == "GLQ"] <- 5
house$cBsmtFinType1[house$BsmtFinType1 == "Unf"] <- 4
house$cBsmtFinType1[house$BsmtFinType1 == "ALQ"] <- 3
house$cBsmtFinType1[house$BsmtFinType1 %in% c("BLQ", "Rec", "LwQ")] <- 2
house$cBsmtFinType1[is.na(house$BsmtFinType1)] <- 1
summarize(group_by(house, cBsmtFinType1), mean(SalePrice, na.rm=T))

summarize(group_by(house, BsmtFinType2), mean(SalePrice, na.rm=T))
house$cBsmtFinType2[house$BsmtFinType2 == "GLQ"] <- 6
house$cBsmtFinType2[house$BsmtFinType2 == "Unf"] <- 5
house$cBsmtFinType2[house$BsmtFinType2 == "ALQ"] <- 4
house$cBsmtFinType2[house$BsmtFinType2 %in% c("Rec", "LwQ")] <- 3
house$cBsmtFinType2[house$BsmtFinType2 == "BLQ"] <- 2
house$cBsmtFinType2[is.na(house$BsmtFinType2)] <- 1
summarize(group_by(house, cBsmtFinType2), mean(SalePrice, na.rm=T))

summarize(group_by(house, BsmtQual), mean(SalePrice, na.rm=T))
house$cBsmtQual[house$BsmtQual == "Ex"] <- 5
house$cBsmtQual[house$BsmtQual == "Gd"] <- 4
house$cBsmtQual[house$BsmtQual == "TA"] <- 3
house$cBsmtQual[house$BsmtQual == "Fa"] <- 2
house$cBsmtQual[house$BsmtQual == "Po"] <- 1
house$cBsmtQual[is.na(house$BsmtQual)] <- 0
summarize(group_by(house, cBsmtQual), mean(SalePrice, na.rm=T))

summarize(group_by(house, CentralAir), mean(SalePrice, na.rm=T))
house$cCentralAir[house$CentralAir == "Y"] <- 1
house$cCentralAir[house$CentralAir == "N"] <- 0
summarize(group_by(house, cCentralAir), mean(SalePrice, na.rm=T))

table(house$Condition1)
house$Condition1[house$Condition1 %in% c("PosA", "PosN")] <- 1
house$Condition1[!house$Condition1 %in% c("PosA", "PosN")] <- 0
table(house$cCondition1)

table(house$Condition2)
house$cCondition2 [house$Condition2 %in% c("PosA", "PosN")] <- 1
house$cCondition2 [!house$Condition2 %in% c("PosA", "PosN")] <- 0
table(house$cCondition2)
```

```
summarize(group_by(house, Electrical), mean(SalePrice, na.rm=T))
house$cElectrical[house$Electrical == "SBrkr" | is.na(house$Electrical)] <- 1
house$cElectrical[!house$Electrical == "SBrkr" & !is.na(house$Electrical)] <- 0
summarize(group_by(house, cElectrical), mean(SalePrice, na.rm=T))

summarize(group_by(house, ExterCond), mean(SalePrice, na.rm=T))
house$cExterCond[house$ExterCond == "Ex"] <- 5
house$cExterCond[house$ExterCond == "Gd"] <- 4
house$cExterCond[house$ExterCond == "TA"] <- 3
house$cExterCond[house$ExterCond == "Fa"] <- 2
house$cExterCond[house$ExterCond == "Po"] <- 1
summarize(group_by(house, cExterCond), mean(SalePrice, na.rm=T))

summarize(group_by(house, ExterQual), mean(SalePrice, na.rm=T))
house$cExterQual[house$ExterQual == "Ex"] <- 5
house$cExterQual[house$ExterQual == "Gd"] <- 4
house$cExterQual[house$ExterQual == "TA"] <- 3
house$cExterQual[house$ExterQual == "Fa"] <- 2
house$cExterQual[house$ExterQual == "Po"] <- 1
summarize(group_by(house, cExterQual), mean(SalePrice, na.rm=T))

summarize(group_by(house, Fence), mean(SalePrice, na.rm=T))
house$cFence[house$Fence %in% c("GdPrv", "GdWo")] <- 2
house$cFence[house$Fence %in% c("MnPrv", "MnWw")] <- 1
house$cFence[is.na(house$Fence)] <- 0
summarize(group_by(house, cFence), mean(SalePrice, na.rm=T))

summarize(group_by(house, FireplaceQu), mean(SalePrice, na.rm=T))
house$cFireplaceQu[house$FireplaceQu == "Ex"] <- 5
house$cFireplaceQu[house$FireplaceQu == "Gd"] <- 4
house$cFireplaceQu[house$FireplaceQu == "TA"] <- 3
house$cFireplaceQu[house$FireplaceQu == "Fa"] <- 2
house$cFireplaceQu[house$FireplaceQu == "Po"] <- 1
house$cFireplaceQu[is.na(house$FireplaceQu)] <- 0
summarize(group_by(house, cFireplaceQu), mean(SalePrice, na.rm=T))

summarize(group_by(house, Foundation), mean(SalePrice, na.rm=T))
house$cFoundation[house$Foundation == "PConc"] <- 1
house$cFoundation[house$Foundation != "PConc"] <- 0
summarize(group_by(house, cFoundation), mean(SalePrice, na.rm=T))

summarize(group_by(house, Functional), mean(SalePrice, na.rm=T))
house$cFunctional[house$Functional %in% c("Sal")] <- 6
house$cFunctional[house$Functional %in% c("Sev")] <- 5
house$cFunctional[house$Functional %in% c("Maj2")] <- 4
house$cFunctional[house$Functional %in% c("Maj1")] <- 3
house$cFunctional[house$Functional %in% c("Mod") | is.na(house$Functional)] <- 2
house$cFunctional[house$Functional %in% c("Min1", "Min2")] <- 1
house$cFunctional[house$Functional %in% c("Typ")] <- 0
summarize(group_by(house, cFunctional), mean(SalePrice, na.rm=T))

summarize(group_by(house, GarageCond), mean(SalePrice, na.rm=T))
house$cGarageCond[house$GarageCond == "Ex"] <- 5
house$cGarageCond[house$GarageCond == "Gd"] <- 4
house$cGarageCond[house$GarageCond == "TA"] <- 3
house$cGarageCond[house$GarageCond == "Fa"] <- 2
house$cGarageCond[house$GarageCond == "Po"] <- 1
house$cGarageCond[is.na(house$GarageCond)] <- 0
summarize(group_by(house, cGarageCond), mean(SalePrice, na.rm=T))

summarize(group_by(house, GarageFinish), mean(SalePrice, na.rm=T))
house$cGarageFinish[house$GarageFinish %in% c("Fin", "RFn")] <- 1
house$cGarageFinish[!house$GarageFinish %in% c("Fin", "RFn")] <- 0
summarize(group_by(house, cGarageFinish), mean(SalePrice, na.rm=T))

summarize(group_by(house, GarageQual), mean(SalePrice, na.rm=T))
house$cGarageQual[house$GarageQual == "Ex"] <- 5
house$cGarageQual[house$GarageQual == "Gd"] <- 4
house$cGarageQual[house$GarageQual == "TA"] <- 3
house$cGarageQual[house$GarageQual == "Fa"] <- 2
house$cGarageQual[house$GarageQual == "Po"] <- 1
house$cGarageQual[is.na(house$GarageQual)] <- 0
summarize(group_by(house, cGarageQual), mean(SalePrice, na.rm=T))

summarize(group_by(house, GarageType), mean(SalePrice, na.rm=T))
```

```
house$cGarageType[house$GarageType %in% c("Attchd", "BuiltIn")] <- 1
house$cGarageType[!house$GarageType %in% c("Attchd", "BuiltIn")] <- 0
summarize(group_by(house, cGarageType), mean(SalePrice, na.rm=T))

summarize(group_by(house, Heating), mean(SalePrice, na.rm=T))
house$cHeating[house$Heating %in% c("GasA", "GasW")] <- 1
house$cHeating[!house$Heating %in% c("GasA", "GasW")] <- 0
summarize(group_by(house, cHeating), mean(SalePrice, na.rm=T))

summarize(group_by(house, HeatingQC), mean(SalePrice, na.rm=T))
house$cHeatingQC[house$HeatingQC == "Ex"] <- 5
house$cHeatingQC[house$HeatingQC == "Gd"] <- 4
house$cHeatingQC[house$HeatingQC == "TA"] <- 3
house$cHeatingQC[house$HeatingQC == "Fa"] <- 2
house$cHeatingQC[house$HeatingQC == "Po"] <- 1
summarize(group_by(house, cHeatingQC), mean(SalePrice, na.rm=T))

summarize(group_by(house, KitchenQual), mean(SalePrice, na.rm=T))
house$cKitchenQual[house$KitchenQual == "Ex"] <- 5
house$cKitchenQual[house$KitchenQual == "Gd"] <- 4
house$cKitchenQual[house$KitchenQual == "TA"] <- 3
house$cKitchenQual[house$KitchenQual == "Fa"] <- 2
house$cKitchenQual[house$KitchenQual == "Po"] <- 1
house$cKitchenQual[is.na(house$KitchenQual)] <- 3
summarize(group_by(house, cKitchenQual), mean(SalePrice, na.rm=T))

table(house$LandContour)
house$cLandContour[house$LandContour == "Lvl"] <- 1
house$cLandContour[house$LandContour != "Lvl"] <- 0
table(house$cLandContour)

table(house$LandSlope)
house$cLandSlope[house$LandSlope == "Gtl"] <- 1
house$cLandSlope[house$LandSlope != "Gtl"] <- 0
table(house$cLandSlope)

summarize(group_by(house, LotConfig), mean(SalePrice, na.rm=T))
house$cLotConfig[house$LotConfig %in% c("CulDSac", "FR3")] <- 1
house$cLotConfig[!house$LotConfig %in% c("CulDSac", "FR3")] <- 0
summarize(group_by(house, cLotConfig), mean(SalePrice, na.rm=T))

table(house$LotShape)
house$cLotShape[house$LotShape == "Reg"] <- 1
house$cLotShape[house$LotShape != "Reg"] <- 0
table(house$cLotShape)

summarize(group_by(house, MasVnrType), mean(SalePrice, na.rm=T))
house$cMasVnrType[house$MasVnrType %in% c("Stone", "BrkFace") | is.na(house$MasVnrType)] <- 1
house$cMasVnrType[!house$MasVnrType %in% c("Stone", "BrkFace") & !is.na(house$MasVnrType)] <- 0
summarize(group_by(house, cMasVnrType), mean(SalePrice, na.rm=T))

summarize(group_by(house, MiscFeature), mean(SalePrice, na.rm=T))
house$cMiscFeature[house$MiscFeature %in% c("TenC")] <- 1
house$cMiscFeature[!house$MiscFeature %in% c("TenC")] <- 0
summarize(group_by(house, cMiscFeature), mean(SalePrice, na.rm=T))

summarize(group_by(house, MSSubClass), mean(SalePrice, na.rm=T))
house$cMSSubClass[house$MSSubClass %in% c("20", "60", "120")] <- 1
house$cMSSubClass[!house$MSSubClass %in% c("20", "60", "120")] <- 0
summarize(group_by(house, cMSSubClass), mean(SalePrice, na.rm=T))

summarize(group_by(house, MSZoning), mean(SalePrice, na.rm=T))
house$cMSZoning[house$MSZoning %in% c("FV")] <- 4
house$cMSZoning[house$MSZoning %in% c("RL","RP")] <- 3
house$cMSZoning[house$MSZoning %in% c("RH","RM")] <- 2
house$cMSZoning[house$MSZoning %in% c("C (all)","I")] <- 1
house$cMSZoning[is.na(house$MSZoning)] <- 2
summarize(group_by(house, cMSZoning), mean(SalePrice, na.rm=T))

summarize(group_by(house, PavedDrive), mean(SalePrice, na.rm=T))
house$cPavedDrive[house$PavedDrive == "Y"] <- 1
house$cPavedDrive[house$PavedDrive != "Y"] <- 0
summarize(group_by(house, cPavedDrive), mean(SalePrice, na.rm=T))

summarize(group_by(house, PoolQC), mean(SalePrice, na.rm=T))
```

```
house$cPoolQC[house$PoolQC %in% c("Ex")] <- 1
house$cPoolQC[!house$PoolQC %in% c("Ex")] <- 0
summarize(group_by(house, cPoolQC), mean(SalePrice, na.rm=T))

summarize(group_by(house, RoofMatl), mean(SalePrice, na.rm=T))
house$cRoofMatl[house$RoofMatl %in% c("Membran", "WdShake", "WdShngl")] <- 1
house$cRoofMatl[!house$RoofMatl %in% c("Membran", "WdShake", "WdShngl")] <- 0
summarize(group_by(house, cRoofMatl), mean(SalePrice, na.rm=T))

summarize(group_by(house, RoofStyle),mean(SalePrice, na.rm=T))
house$cRoofStyle[house$RoofStyle %in% c("Hip", "Shed")] <- 1
house$cRoofStyle[!house$RoofStyle %in% c("Hip", "Shed")] <- 0
summarize(group_by(house, cRoofStyle),mean(SalePrice, na.rm=T))

summarize(group_by(house, SaleCondition), mean(SalePrice, na.rm=T))
house$cSaleCondition[house$SaleCondition %in% c("Partial")] <- 4
house$cSaleCondition[house$SaleCondition %in% c("Normal", "Alloca")] <- 3
house$cSaleCondition[house$SaleCondition %in% c("Family","Abnorml")] <- 2
house$cSaleCondition[house$SaleCondition %in% c("AdjLand")] <- 1
summarize(group_by(house, cSaleCondition), mean(SalePrice, na.rm=T))

summarize(group_by(house, SaleType), mean(SalePrice, na.rm=T))
house$cSaleType[house$SaleType %in% c("New", "Con")] <- 5
house$cSaleType[house$SaleType %in% c("CWD", "ConLI")] <- 4
house$cSaleType[house$SaleType %in% c("WD", "VWD", "NA")] <- 3
house$cSaleType[house$SaleType %in% c("COD", "ConLw", "ConLD")] <- 2
house$cSaleType[house$SaleType %in% c("Oth")] <- 1
house$cSaleType[is.na(house$SaleType)] <- 0
summarize(group_by(house, cSaleType), mean(SalePrice, na.rm=T))

table(house$Street)
house$cStreet[house$Street == "Pave"] <- 1
house$cStreet[house$Street != "Pave"] <- 0
table(house$cStreet)

table(house$Utilities)
house$cUtilities[house$Utilities == "AllPub"] <- 1
house$cUtilities[house$Utilities != "AllPub"] <- 0
house$cUtilities[is.na(house$Utilities)] <- 0
table(house$cUtilities)

summarize(group_by(house, Neighborhood), mean(SalePrice, na.rm=T))
house$cNeighborhood[house$Neighborhood %in% c("NoRidge", "NridgHt", "StoneBr")] <- 1
house$cNeighborhood[!house$Neighborhood %in% c("NoRidge", "NridgHt", "StoneBr")] <- 0
summarize(group_by(house, cNeighborhood), mean(SalePrice, na.rm=T))

summarize(group_by(house, Exterior1st), mean(SalePrice, na.rm=T))
house$cExterior1st[house$Exterior1st %in% c("ImStucc", "Stone", "CemntBd", "VinylSd", "BrkFace")] <- 2
house$cExterior1st[house$Exterior1st %in% c("Plywood", "HdBoard", "Stucco", "WdShing", "Wd Sdng", "MetalSd")] <- 1
house$cExterior1st[house$Exterior1st %in% c("AsbShng", "CBlock", "AsphShn", "BrkComm") | is.na(house$Exterior1st)] <- 0
summarize(group_by(house, cExterior1st), mean(SalePrice, na.rm=T))

summarize(group_by(house, Exterior2nd), mean(SalePrice, na.rm=T))
house$cExterior2nd[house$Exterior2nd %in% c("Other", "ImStucc", "CmentBd", "VinylSd", "BrkFace")] <- 2
house$cExterior2nd[!house$Exterior2nd %in% c("Other", "ImStucc", "CmentBd", "VinylSd", "BrkFace")] <- 1
summarize(group_by(house, cExterior2nd), mean(SalePrice, na.rm=T))

summarize(group_by(house, HouseStyle), mean(SalePrice, na.rm=T))
house$cHouseStyle[house$HouseStyle %in% c("2.5Fin")] <- 220000/100000
house$cHouseStyle[house$HouseStyle %in% c("2Story")] <- 210051/100000
house$cHouseStyle[house$HouseStyle %in% c("1Story")] <- 175985/100000
house$cHouseStyle[house$HouseStyle %in% c("SLvl")] <- 166703/100000
house$cHouseStyle[house$HouseStyle %in% c("2.5Unf")] <- 157354/100000
house$cHouseStyle[house$HouseStyle %in% c("1.5Fin")] <- 143116/100000
house$cHouseStyle[house$HouseStyle %in% c("SFoyer")] <- 135074/100000
house$cHouseStyle[house$HouseStyle %in% c("1.5Unf")] <- 110150/100000
summarize(group_by(house, cHouseStyle), mean(SalePrice, na.rm=T))

# Normalize year sold
house$YrSold <- (2010 - house$YrSold)

# Clean up missing values
house$LotFrontage[is.na(house$LotFrontage)] <- (house$LotArea * 0.0078)
house$GarageYrBlt[is.na(house$GarageYrBlt)] <- house$YearBuilt
house$MasVnrArea[is.na(house$MasVnrArea)] <- 0
```

```
house$BsmtFinSF1[is.na(house$BsmtFinSF1)] <- 0
house$BsmtFinSF2[is.na(house$BsmtFinSF2)] <- 0
house$BsmtUnfSF[is.na(house$BsmtUnfSF)] <- 0
house$TotalBsmtSF[is.na(house$TotalBsmtSF)] <- 0
house$BsmtFullBath[is.na(house$BsmtFullBath)] <- 0
house$BsmtHalfBath[is.na(house$BsmtHalfBath)] <- 0
house$GarageCars[is.na(house$GarageCars)] <- 0
house$GarageArea[is.na(house$GarageArea)] <- 0

# Create new correlated variables
house$BRBR <- log(((house$FullBath + house$HalfBath)+1) / (1+house$BedroomAbvGr))
house$GLARooms <- log(1+(house$GrLivArea / house$TotRmsAbvGrd))
house$GQGC <- log(1+(house$cGarageQual * house$GarageCars))
house$GRGR <- log(1+(house$GrLivArea / mean(house$GrLivArea, na.rm=TRUE)))
house$OQEC <- log(1+(house$OverallQual * house$cExterCond))
house$OQFB <- log(1+(house$OverallQual * house$FullBath))
house$OQGLA <- log(1+(house$OverallQual * house$GrLivArea))
house$OQTBSF <-     log(1+(house$OverallQual * house$TotalBsmtSF))
house$OQYB <- log(1+(house$OverallQual * house$YearBuilt))
house$OQYR <- log(1+(house$OverallQual * house$YearRemodAdd))

# Take logs of continuous variables
house$logSalePrice <- log(1+house$SalePrice)
house$logX1stFlrSF <- log(1+house$X1stFlrSF)
house$logX2ndFlrSF <- log(1+house$X2ndFlrSF)
house$logX3SsnPorch <- log(1+house$X3SsnPorch)
house$logBsmtFinSF1 <- log(1+house$BsmtFinSF1)
house$logBsmtFinSF2 <- log(1+house$BsmtFinSF2)
house$logBsmtUnfSF <- log(1+house$BsmtUnfSF)
house$logEnclosedPorch <- log(1+house$EnclosedPorch)
house$logGarageArea <- log(1+house$GarageArea)
house$logGrLivArea <- log(1+house$GrLivArea)
house$logLotArea <- log(1+house$LotArea)
house$logLotFrontage <- log(1+house$LotFrontage)
house$logLowQualFinSF <- log(1+house$LowQualFinSF)
house$logMasVnrArea <- log(1+house$MasVnrArea)
house$logMiscVal <- log(1+house$MiscVal)
house$logOpenPorchSF <- log(1+house$OpenPorchSF)
house$logPoolArea <- log(1+house$PoolArea)
house$logScreenPorch <- log(1+house$ScreenPorch)
house$logTotalBsmtSF <- log(1+house$ScreenPorch)
house$logWoodDeckSF <- log(1+house$WoodDeckSF)

# Delete variables that are not needed
house$Street <- NULL
house$LotShape <- NULL
house$LandContour <- NULL
house$Utilities <- NULL
house$LotConfig <- NULL
house$LandSlope <- NULL
house$Neighborhood <- NULL
house$Condition1 <- NULL
house$Condition2 <- NULL
house$BldgType <- NULL
house$HouseStyle <- NULL
house$RoofStyle <- NULL
house$RoofMatl <- NULL
house$Exterior1st <- NULL
house$Exterior2nd <- NULL
house$MasVnrType <- NULL
house$ExterQual <- NULL
house$ExterCond <- NULL
house$Foundation <- NULL
house$BsmtQual <- NULL
house$BsmtCond <- NULL
house$BsmtExposure <- NULL
house$BsmtFinType1 <- NULL
house$BsmtFinType2 <- NULL
house$Heating <- NULL
house$HeatingQC <- NULL
house$CentralAir <- NULL
house$Electrical <- NULL
house$KitchenQual <- NULL
house$FireplaceQu <- NULL
house$GarageType <- NULL
```

```
house$GarageFinish <- NULL
house$GarageQual <- NULL
house$GarageCond <- NULL
house$PavedDrive <- NULL
house$Functional <- NULL
house$PoolQC <- NULL
house$Fence <- NULL
house$MiscFeature <- NULL
house$SaleType <- NULL
house$SaleCondition <- NULL
house$MSZoning <- NULL
house$Alley <- NULL

names(house)
datatable(head(house, n=20),options = list(scrollX = TRUE))


write.csv(house, file="C:/Users/flip/Desktop/SMU/MSDS6372/house1.csv", row.names=F)
```

.11840 SAS Code using house.csv file generated by R code above (with SalePrice and logSalePrices of zero deleted).

```
proc import datafile='/home/pbedwards0/Phil/house1.csv'
out=house dbms=csv replace;
getnames=yes;
run;


/* We ended up addressing a few more last minute outliers and also stacking terms here  */
data house_out;
set house;
if GrLivArea>4000 AND logSalePrice>1 then delete;
if GrLivArea=5642 then delete;
if GrLivArea=4676 then delete;
if LotArea=159000 then delete;
if LotArea=215245 then delete;
if LotArea=164660 then delete;
if LotArea=115149 then delete;
if TotalBsmtSF=6110 then delete;
sqOverallCond=OverallCond**2;
sqGarageCars=GarageCars**2;
sqLotArea=logLotArea**2;
stot=(logGrLivArea+cFunctional)**2;
ctot=(logGrLivArea+cFunctional)**3;
run;

/* Model output code block. Cut paste variables from your testing output into the model below  */
proc glm data=house_out plots=all;
model logSalePrice =
OverallQual|sqLotArea OQYB|sqOverallCond OQYR|sqOverallCond OverallCond YearBuilt YearRemodAdd MasVnrArea OQYB|BsmtFinSF1 OQYR|BsmtFinSF1
BsmtFinSF2 BsmtFullBath KitchenAbvGr Fireplaces
WoodDeckSF EnclosedPorch YrSold cBsmtExposure cBsmtQual cCentralAir OQYB|cExterCond cExterQual cFoundation cFunctional|cCentralAir
cHeatingQC cKitchenQual cMSZoning cPavedDrive cStreet cNeighborhood cExterior1st cExterior2nd logBsmtFinSF1
logBsmtFinSF2 logBsmtUnfSF X1stFlrSF stot ctot logGarageArea logLotArea logLowQualFinSF logScreenPorch BRBR GQGC OQTBSF GRGR sqGarageCars
cExterior1st|MasVnrArea sqLotArea|cSaleCondition
/solution cli clparm;
output out=results predicted=logpredict;
run;

/* !!!!!!! One problem we've noted is logpredict appends logpredict1 logpredict2 etc.. to the data table each time this is run  */
/* !!!!!!! This is an issues because in order for the automation below to work, it looks for logpredict only, so will only catch first run  */
/* !!!!!!! We caught this too late to clean the code, but the main workaround is anytime you want to run a new test, have to run ALL of the code */
/* !!!!!!! Basically have to push the run all button each time iwth nothing selected if you are trying to automate to final.csv output */


/* This preps the data for output by filling in the SalePrice with the predicted values from the proc glm command*/
/**** !!!!! YOU MUST RUN THE ENTIRE CODE OR THIS STEP WILL RECYCLE PREVIOUS PROC GLM OUTPUT  See note above !!!! ***/
data kaggle_out (KEEP=id SalePrice);
  set results;
  array change _numeric_;
      do over change;
         if SalePrice=. then SalePrice=exp(logpredict);
      end;
 run ;
```

```
/* Clean up for Kaggle formatted output   */
proc sql noprint;
DELETE FROM kaggle_out WHERE ID<1461;
quit;

proc print data=kaggle_out;run;

/* Export to file */
proc export data=kaggle_out
outfile='/home/pbedwards0/Phil/final.csv'            /* <---- YOUR PATH HERE */
dbms=csv replace;
delimiter=',';
run;
```