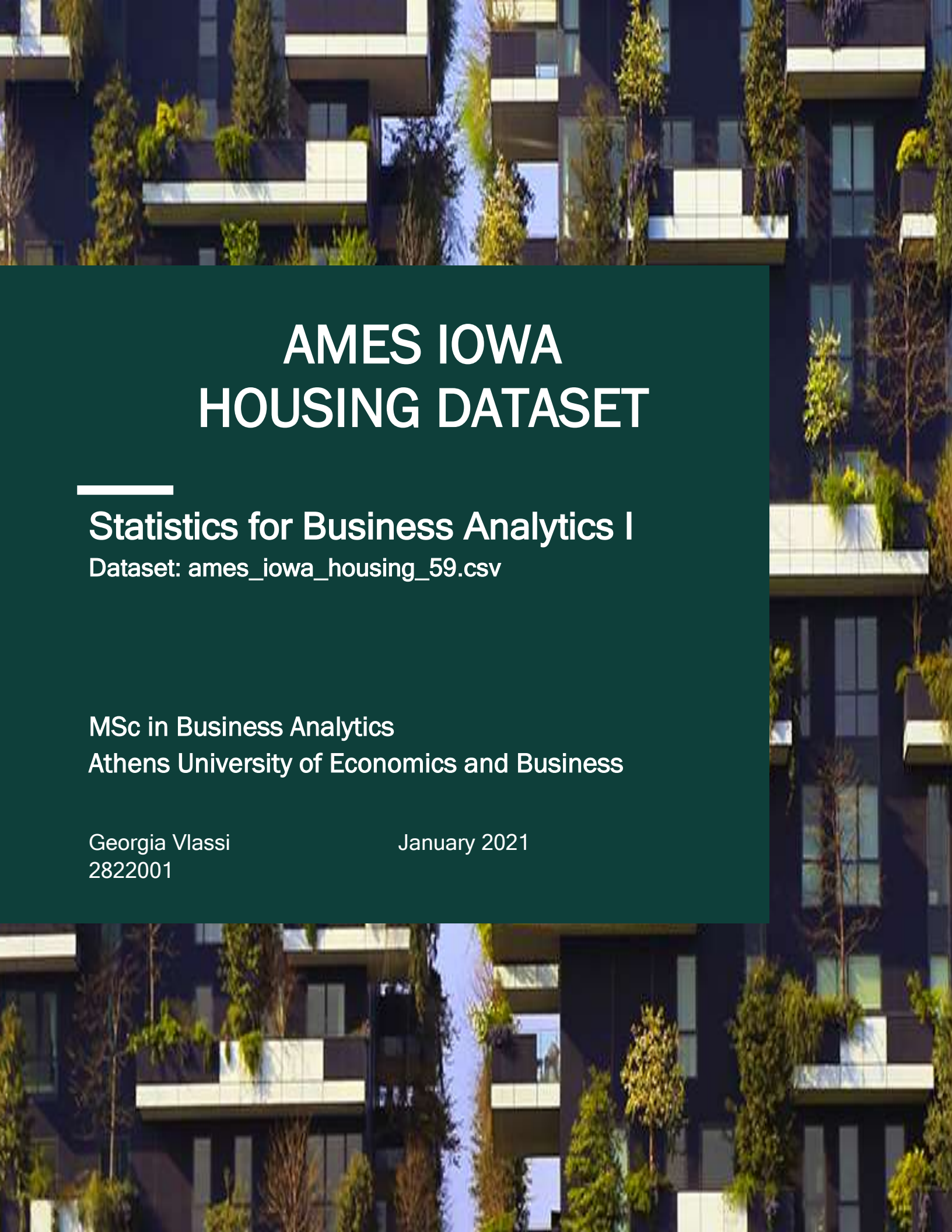

AMES IOWA HOUSING DATASET

Statistics for Business Analytics I

Dataset: ames_iowa_housing_59.csv

MSc in Business Analytics
Athens University of Economics and Business

Georgia Vlassi
2822001

January 2021

Table of Contents

Abstract

The data of this report refer to 1500 residential property sales that had occurred in Ames, Iowa between 2006 and 2012. The dataset contains 82 both ordinal, nominal, continuous and discrete variables, which focus on the quality and quantity of many physical attributes of the property. In order to reduce the number of variables that describe the sale price of a property precisely, Lasso and Stepwise procedure were used. The main scope of this report is to identify the best model for predicting sales price of the properties.

1. Introduction – Description of the Problem

The main scope of this report is to identify the best regression model, including the most important variables that precisely describe the predicting sales price of Ames, Iowa properties between 2006 and 2012. The dataset contains 82 both ordinal, nominal, discrete and continuous attributes, which focus on the quality and quantity of many physical attributes of the property. There are 23 Ordinal variables that appraise the quality and condition of house parts. The 23 Nominal variables describe the types and styles of houses, materials and locations. The 14 Discrete variables quantify the number of bedrooms, fireplaces, basement baths, garage spots and also year built and sold. Finally, there are 22 Continuous variables, which describe the area dimensions in square feet of basement, dwelling etc.

Prior to the analysis of our model, there are a few transformations that should be implemented. As we import the dataset, we observe that there are variables that are not defined correctly. There are also null values that should be handled. After analysis and data modifications Correlations, Lasso and Stepwise procedure were performed, to find the 11 most accurate and highly correlated variables that predict the sales price of properties.

After applying multiple linear regression models, which will be interpret further in our report ,we end up with 11 attributes that will be used as input to our predicting model. These attributes describe the overall quality of the dwelling, the ground living area, the square feet of lot, the year the house built and quality of basement and kitchen.

2. Descriptive analysis and exploratory data analysis

3.1 Data Cleansing

In order to manipulate our data, the R language was used. After importing the dataset, there were only integer and character classes of variables. We observed that variables ‘PID’, ‘MS.Subclass’, ‘Overall.Qual’ and ‘Overall.Cond’ were not defined correctly. These were converted to characters, as described to the specifications.

During Data Cleansing procedure, all null (NA) values must be replaced. At Table 1 there are listed all null values of every variable.

X	Order	PID	MS.SubClass	MS.Zoning	Lot.Frontage	Lot.Area	Street	Alley
0	0	0	0	0	253	0	0	1408
Lot.Shape	Land.Contour	Utilities	Lot.Config	Land.Slope	Neighborhood	Condition.1	Condition.2	Bldg.Type
0	0	0	0	0	0	0	0	0
House.Style	Overall.Qual	Overall.Cond	Year.Built	Year.Remod.Add	Roof.Style	Roof.Matl	Exterior.1st	Exterior.2nd
0	0	0	0	0	0	0	0	0
Mas.Vnr.Type	Mas.Vnr.Area	Exter.Qual	Exter.Cond	Foundation	Bsmt.Qual	Bsmt.Cond	Bsmt.Exposure	BsmtFin.Type.1
13	13	0	0	0	51	51	54	51
BsmtFin.SF.1	BsmtFin.Type.2	BsmtFin.SF.2	Bsmt.Unf.SF	Total.Bsmt.SF	Heating	Heating.QC	Central.Air	Electrical
1	51	1	1	1	0	0	0	0
X1st.Flr.SF	X2nd.Flr.SF	Low.Qual.Fin.SF	Gr.Liv.Area	Bsmt.Full.Bath	Bsmt.Half.Bath	Full.Bath	Half.Bath	Bedroom.AbvGr
0	0	0	0	1	1	0	0	0
Kitchen.AbvGr	Kitchen.Qual	TotRms.AbvGrd	Functional	Fireplaces	Fireplace.Qu	Garage.Type	Garage.Yr.Blt	Garage.Finish
0	0	0	0	0	717	78	78	78
Garage.Cars	Garage.Area	Garage.Qual	Garage.Cond	Paved.Drive	Wood.Deck.SF	Open.Porch.SF	Enclosed.Porch	X3Ssn.Porch
0	0	78	78	0	0	0	0	0
Screen.Porch	Pool.Area	Pool.QC	Fence	Misc.Feature	Misc.Val	Mo.Sold	Yr.Sold	Sale.Type
0	0	1494	1185	1451	0	0	0	0
Sale.Condition	SalePrice							
0	0							

Table 1 Number of null values per attribute

A new data frame was created, without including variables ‘X’, ‘Order’, ‘PID’, ‘Alley’, ‘Pool.QC’, ‘Fence’ and ‘Misc.Feature’, because either they were not necessary to our analysis or contained more than 70% of null values.

3.2 Data Transformation

Attributes with null values were separated into different categories and handled, respectively. The null values of numeric variables were updated to 0, and further the character null values were updated to ‘No Basement’, ‘No Garage’ or ‘NA’.

During Data Transformation procedure, all integers were converted to numeric variables and all characters to factors. Concerning our main scope of analysis, to find the attributes highly correlated to ‘SalePrice’, which is numeric, all ordinal factors were converted to numeric and all nominal factors to dummy variables. As a result, binary attribute was created for every unique category of the categorical

values. Succeeding implementing all the above transformation, a new data frame was produced, including only numeric variables (see Commands 1 in R source code for details).

3.3 Select Attributes

Following the conversion of some nominal variables to dummy, we ended up with a new table including 230 numeric attributes. We constructed a Corrplot to find the highly correlated attributes to our target SalePrice. At Figure 1 are presented the attributes that have correlation smaller than -0.25 or greater than 0.25 with 'SalePrice'. All the colorful squares show the correlation between two variables, if is blue they are positive correlated and if it is orange, they are negative correlation.

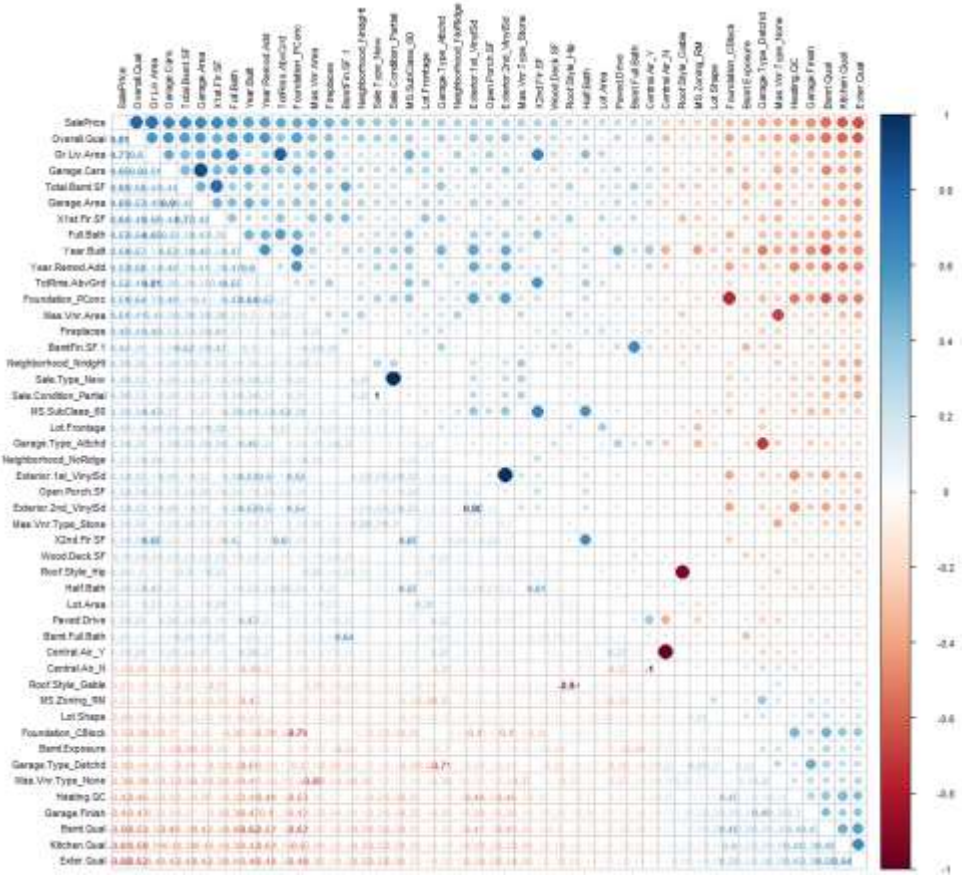

Figure 1 Attributes highly correlated to 'SalePrice'

After Correlation plot the 45 highly correlated attributes, that are shown in Table 2, were used as input for Lasso. The summary of model is shown is Table 3. The LASSO method was used, because it is feature selection technique that can remove the appropriate variables, without much loss of information. It, also, performs better on large datasets. The covariates with p-value > 0.05 should be removed later from our model. In summary of a model, the p-value indicates if we should reject or accept the H0 (null) hypothesis, that the specific attribute could be equal to 0. Further analysis and

interpretation of coefficients will be later on our final model. In the beginning, we implement Lasso with the coefficients against the log-lambda value and with each curve labeled, as shown in Figure 2. We used cross validation to find a reasonable value for lambda. To find explicitly the selected optimal values of λ check Figure 3, where the λ at which the minimal MSE was achieved is 513.2092 and the most regularized model whose mean squared error was within one standard error of the minimal is 4786.206. From Lasso we chose the coefficients produced from lambda.1se, as it has fewer variables while it does not differ significantly from the model derived from lambda.min in terms of Mean Square Error. At Table 4 are shown the attributes produced from Lasso and would be used as input to Stepwise procedure.

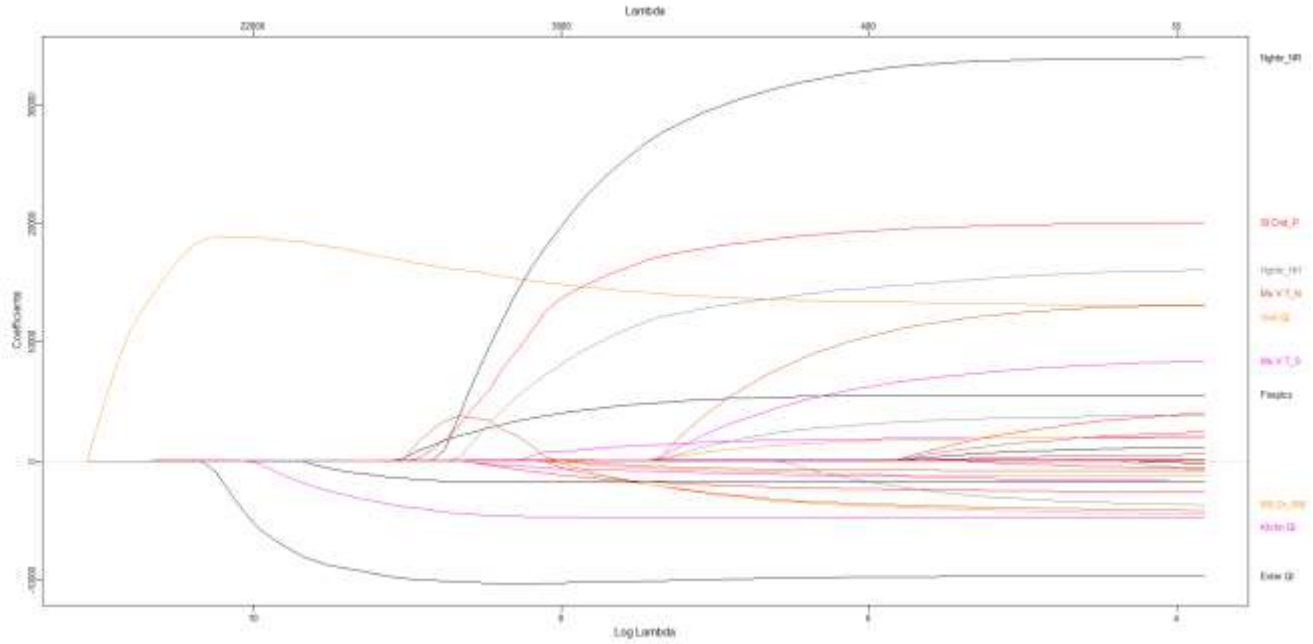

Figure 2 Coefficients against the log-lambda

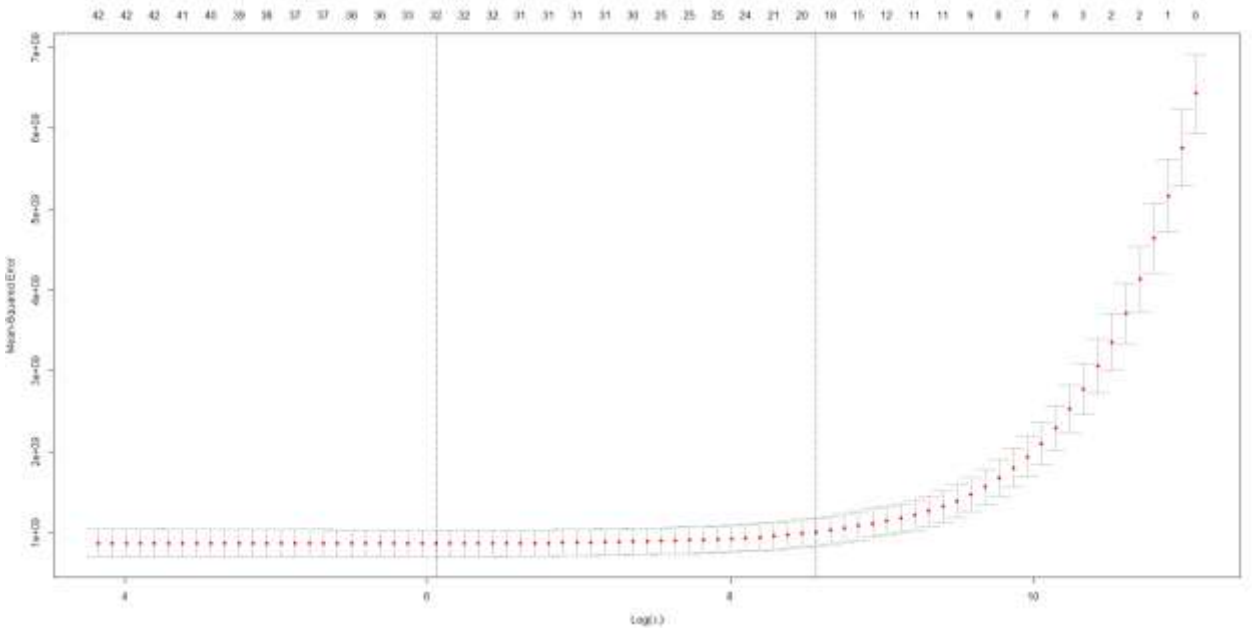

Figure 3 Log-lambda against Mean Square Error

Succeeding the Lasso, Stepwise procedure was conducted with 20 variables as input. The aforementioned summary is shown in Table 5. Stepwise procedure was selected, as it double checks the variables by adding or removing step by step covariates and it uses AIC , which is preferable for prediction. We ended up with the model with 18 covariates, where the AIC has its minimum value, as shown in Table 6.

VIF was used to detect whether multicollinearity exists in our regression model. It measures how much the variance (or standard error) of the estimated regression coefficient is inflated due to collinearity. If VIF (see Table 7) is lower than 10 we accept the coefficient, as it does not disrupt collinearity (see Commands 2 in R source code for details).

3.4 Data Analysis

During to our explanatory analysis, we had to draw some histogram graphs, regarding the 18 attributes occurred from Correlation plot, Lasso and Stepwise, to explain a dwelling, as shown at Figure 4. A typical house has “Above average” overall quality and “Average” exterior quality. It contains minimum 1 fireplace, a ‘highly’ qualified kitchen, a ‘medium’ garage and basement and has been constructed mostly after 2000. At Table 8 the normality of the above attributes is shown and for all attributes we must reject the normality (Shapiro test p-value < 2.2e-16 < 0.05 & Kolmogorov-Smirnov test p-value < 2.2e-16 < 0.05). As our dataset is greater than 50 observations, QQ plots are shown in

Figure 5 and we observe that these attributes are not normally distributed (see Commands 3 in R source code for details).

After constructed the above histograms and QQ plots, we observed that attributes ‘Neighborhood_NridgHt’ and ‘Neighborhood_NoRidge’ should be removed, as all values are in one category and as a result, they did not have a great impact in ‘SalePrice’.

From all the above steps we ended up with a model, which contains 15 attributes: ‘SalePrice’, ‘Overall.Qual’, ‘Gr.Liv.Area’, ‘Total.Bsmt.SF’, ‘Garage.Area’, ‘Exter.Qual’, ‘Year.Built’, ‘Mas.Vnr.Area’, ‘Fireplaces’, ‘ BsmtFin.SF.1’, ‘Lot.Frontage’, ‘Lot.Area’, ‘Bsmt.Exposure’, ‘Bsmt.Qual’, ‘Kitchen.Qual’.

3. Pairwise Comparisons

A new data frame was created, including only the 15 attributes that mentioned above. A new correlation plot (see Figure 6) was designed to give more emphasis to the relationships among attributes. If the ellipse leans towards the right, it is positive correlation and if it leans to the left, it is negative correlation.

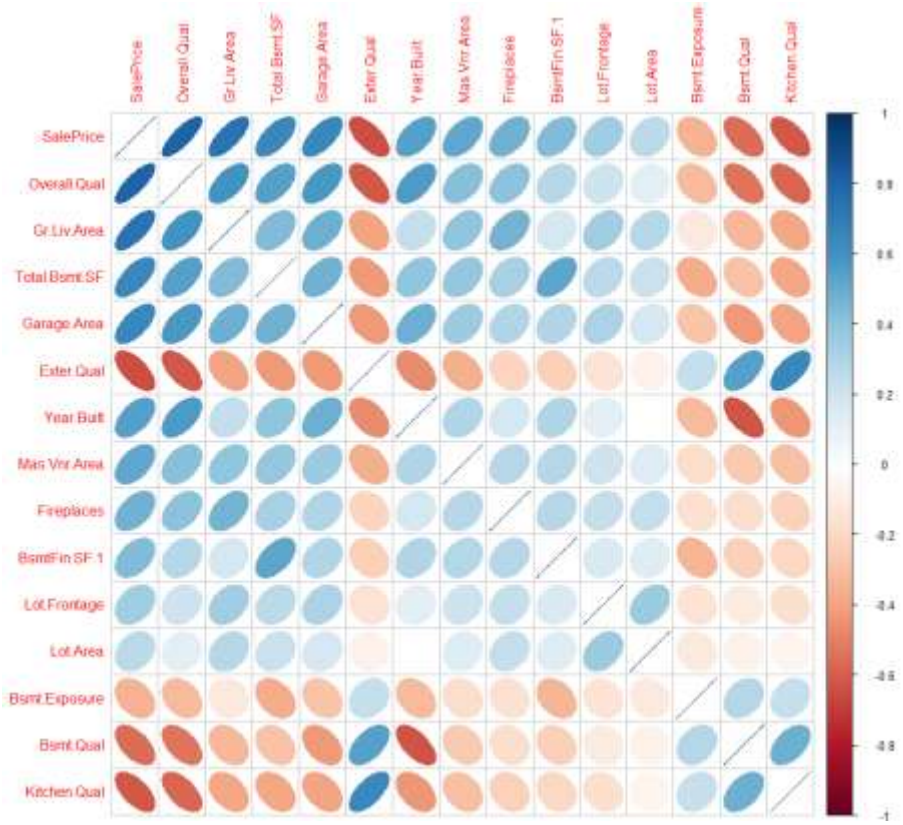

Figure 6 Correlation plot of final attributes after Lasso and Stepwise

It is important to be constructed pairwise comparisons between attributes and target 'SalePrice'. For all categorical variables, boxplots at Figure 7 were constructed. Regarding 'Overall Quality', 'External Quality', 'Kitchen Quality' and 'Basement Quality ' we observe that they are relative to price. 'Very Excellent' or 'Excellent' quality have the houses with prices nearly 700.000$. There is no great variance regarding 'Basement Exposure' and 'SalePrice', as the mean is very close to median at each level. Also, at 'Kitchen Quality' and 'Basement Quality' there are levels without values.

Observing the scatterplots at Figure 8 of numeric variables with 'SalePrice', we can conclude that none of the attributes is normally distributed to 'SalePrice', as every p-value < 2.2e-16 < 0.05, which against null hypothesis that every numeric attribute is normally distributed to 'SalePrice' (see Commands 4 for details).

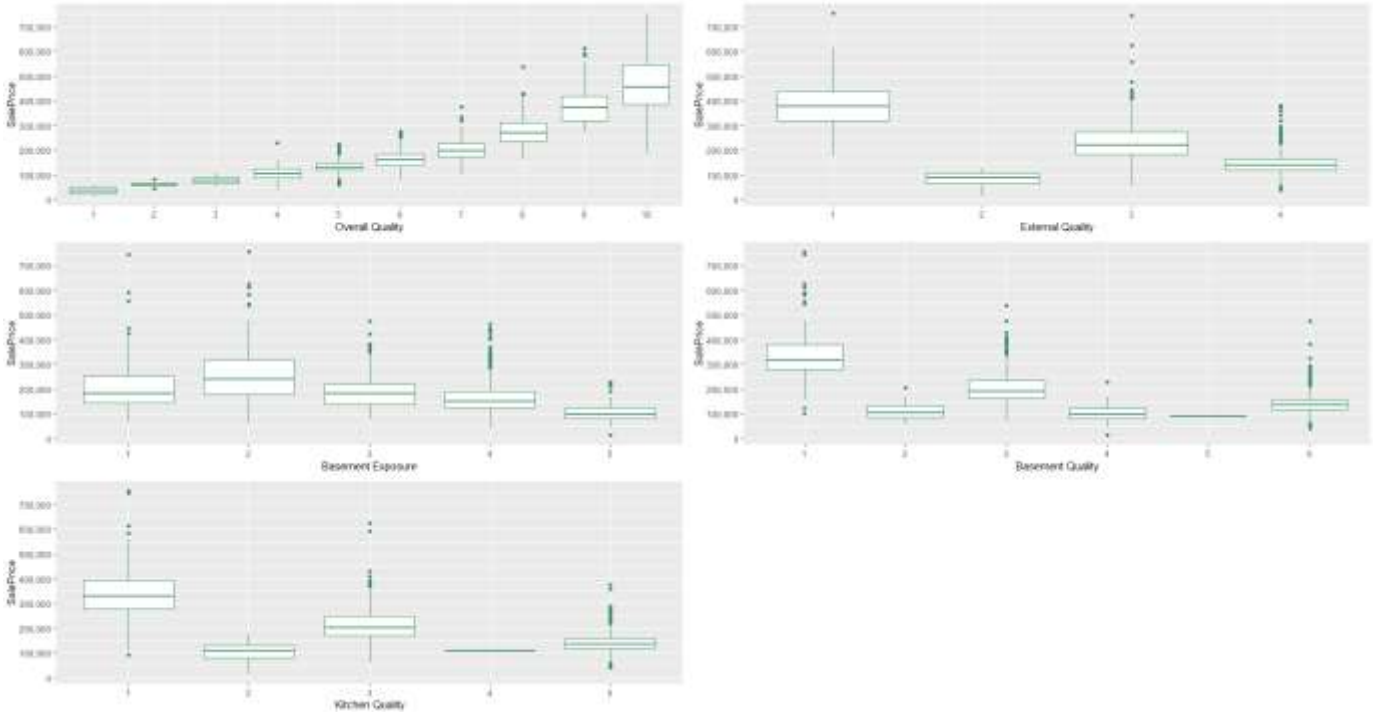

Figure 7 Boxplots among ‘SalePrice’ and categorical attributes

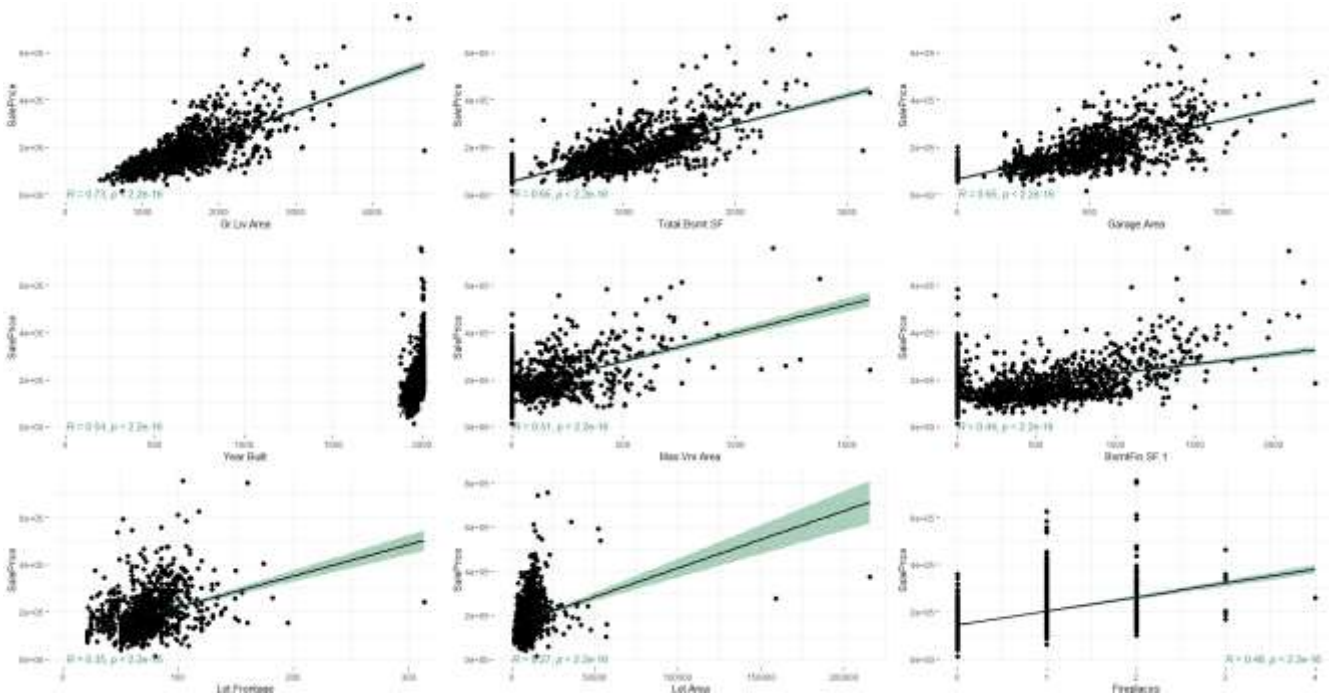

Figure 8 Scatterplots among ‘SalePrice’ and numeric attributes

4. Predictive or Descriptive models

5.1 Summary Analysis of model and assumptions

Since we have finished with the variable analysis, we can train some models and see the results. The first model that we are going to train is a full model based on all variables that LASSO and Stepwise procedure indicated to us. At Table 10 we can see the results from the summary of the model.

The mathematical formulation of first model is shown below:

SalePrice = -2.309e+05 + Overall.Qual*1.438e+04 + Gr.Liv.Area*4.672e+01 + Total.Bsmt.SF*1.576e+01 + Garage.Area*3.348e+01 - Exter.Qual*1.268e+04 Year.Built*1.407e+02 + Mas.Vnr.Area*2.866e+01 + Fireplaces*5.908e+03 + - BsmtFin.SF.1*2.160e+01 + Lot.Frontage*1.401e+02 + Lot.Area*5.620e-01 - Bsmt.Exposure*2.305e+03 - Bsmt.Qual*2.599e+03 - Kitchen.Qual*6.047e+03 +ε
where ε ~ N(0, 29800)

At Table 11, we observe that when we exclude Intercept from our model, the updated statistic Adjusted R-squared has been increased from 0,8625 to 0,9771. Although, this value is closer to 1, which means that 98% of the variability is explained by this model, this value is iconic. The real Adjusted R-squared value of our final model without Intercept is 0.8628, which means that it has not a great impact. Respectively, we reach the outcome that we should not exclude Intercept.

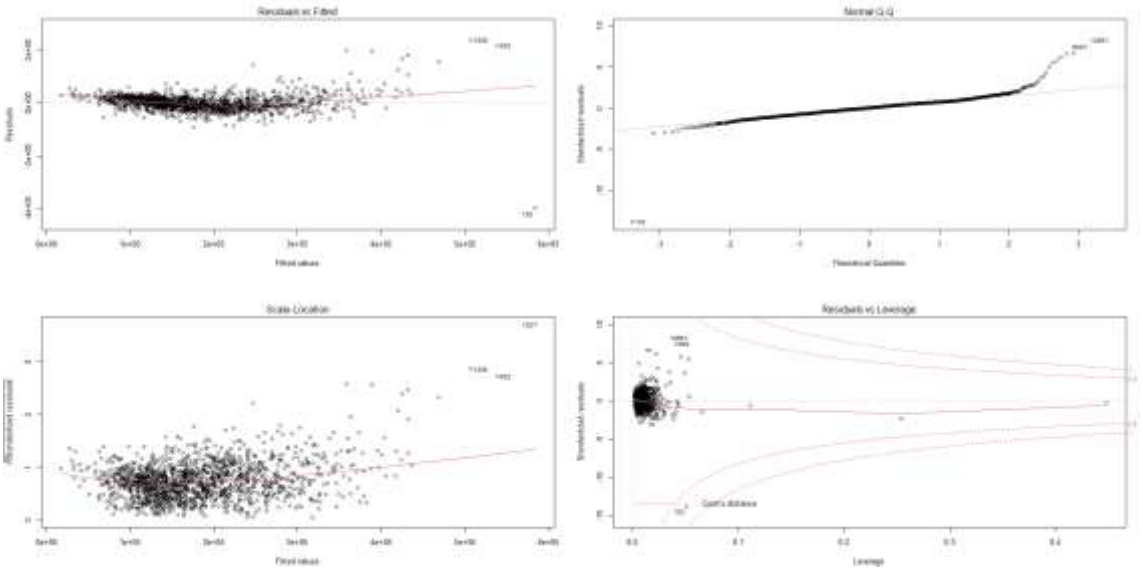

Figure 9 Plot of the residuals

Having Figure 9 as reference, we should check the below assumptions for our first model:

I. Normality of residuals

In the beginning we check for the normality of residuals. The assumptions of normality (Lilliefors KS $p < 2.22e\text{-}16 < 0.05$, Shapiro-Wilk $p < 2.22e\text{-}16 < 0.05$) are violated, which indicates that the residuals do not come from a normally distributed population. As our observations are greater than 50, we also implemented a QQ plot, which evaluates the normality of errors assumption. If the residuals were normally distributed, they would lie exactly on this line. As shown at Figure 10, there is deviation near the ends.

II. Homoscedasticity of residuals variance

As shown in the third of Figure 9, the square root of the standardized residuals, indicates the residuals that have a mean of zero and a variance of one. In this case the residuals seem to increase as the fitted Y values increase which means existence of heteroscedasticity. We will also test this assumption with the No-constant variance score test which is used for testing homoscedasticity. As shown in Table 12 (Non-constant Variance Score Test $P < 2.22e\text{-}16 < 0.05$), we must reject the null hypothesis that the variance of the residuals is constant and infer that we have heteroscedasticity.

III. Linearity of the data.

The p-values shown in Figure 12, indicates the null hypothesis that the variable (y) is assumed to be linear to predictor (x). For the p-values less than 0.05 we must reject the null hypothesis of linearity. In case all variables where linear to predictor 'SalePrice' the blue line would lie exactly on this line at Figure 13.

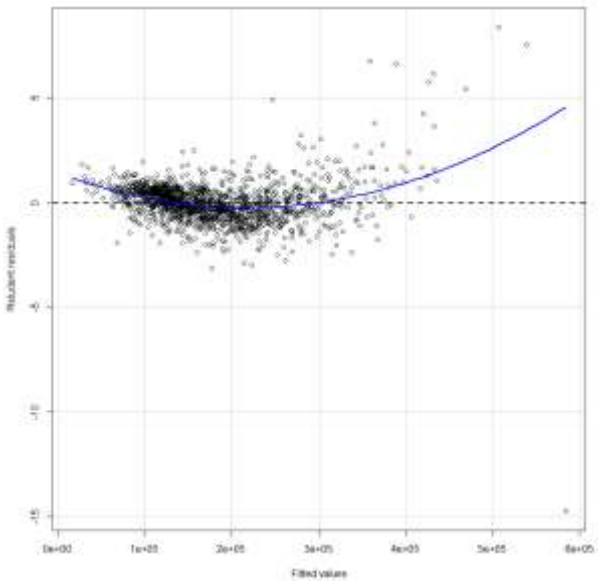

Figure 13 Residuals Plot for the full model

IV. Independence of residuals error terms.

Autocorrelation is a characteristic of data that violates the assumption of instance independence. We implemented (Durbin Watson test p-value 0.714 >0.05) and therefore we do not reject the null hypothesis that there is no correlation among the residuals(see Figure 14 for details). As a result, the residuals may have a zero autocorrelation.

We conclude that the first three assumptions of the linear model are violated. We will give a brief analysis of possible fixed to the violations with updated models.

5.2 Summary Analysis of model with logarithmic SalePrice and assumptions

In our first attempt to fix the above violations of assumptions we created a new model, including logarithmic 'SalePrice'. From the updated model no significant attributes like 'Exter.Qual', 'Mas.Vnr.Area' and 'Bsmt.Qual' have been removed. At Table 13 we can see the results from the summary of the updated model.

The mathematical formulation of second model is shown below:

log(SalePrice) = 6.542e+00 + Overall.Qual*1.060e-01 + Gr.Liv.Area*2.116e-04 + Total.Bsmt.SF*8.000e-05 + Garage.Area*2.046e-04 + Year.Built*2.183e-03 + Fireplaces*4.712e-02 + BsmtFin.SF.1*8.443e-05 + Lot.Frontage*7.836e-04 + Lot.Area*2.293e-06 - Bsmt.Exposure*9.746e-03 - Kitchen.Qual*1.974e-02 + ε, where $\varepsilon \sim N(0, 0.1423)$

Considering Figure 15, Figure 16, Figure 17, the assumptions of normality (Lilliefors KS $p < 2.22e\text{-}16 < 0.05$, Shapiro-Wilk $p < 2.22e\text{-}16 < 0.05$), homoscedasticity (Non-constant Variance Score Test $P = 0.47811 > 0.05$) and autocorrelation (Durbin Watson test $p = 0.982 > 0.05$), we conclude that only homoscedasticity fixes and the violations of normality and linearity still remain.

5.3 Summary Analysis of model with polynomials and assumptions

In our second attempt to fix the above violations of assumptions we created a new model, including polynomials. From the updated model no significant attributes like 'I(Gr.Liv.Area^8)', 'Exter.Qual' and 'Bsmt.Exposure' have been removed. At Table 14 we can see the results from the summary of the updated model.

The mathematical formulation of third model is shown below:

SalePrice = -2.546e+05 + Overall.Qual^5*2.347e+00 + Total.Bsmt.SF*.203e+01 + Garage.Area*5.043e+01 + Year.Built*1.781e+02 +Mas.Vnr.Area*2.997e+01 + Fireplaces*1.576e+04 + BsmtFin.SF.1*1.371e+01 + Lot.Frontage*2.253e+02 + Lot.Area*8.375e-01 -Bsmt.Qual*2.940e+03 - Kitchen.Qual*6.226e+03 + ε, where ε ~ N(0, 32460)

Considering Figure 18, Figure 19, the assumptions of normality (Lilliefors KS p < 2.22e-16 < 0.05, Shapiro-Wilk p < 2.22e-16 < 0.05), homoscedasticity (Non-constant Variance Score Test p < 2.22e-16 < 0.05) and autocorrelation (Durbin Watson test p = 0.226 > 0.05), we conclude that only linearity fixes and the violations of normality and heteroscedasticity still remain (see Commands 5 for details).

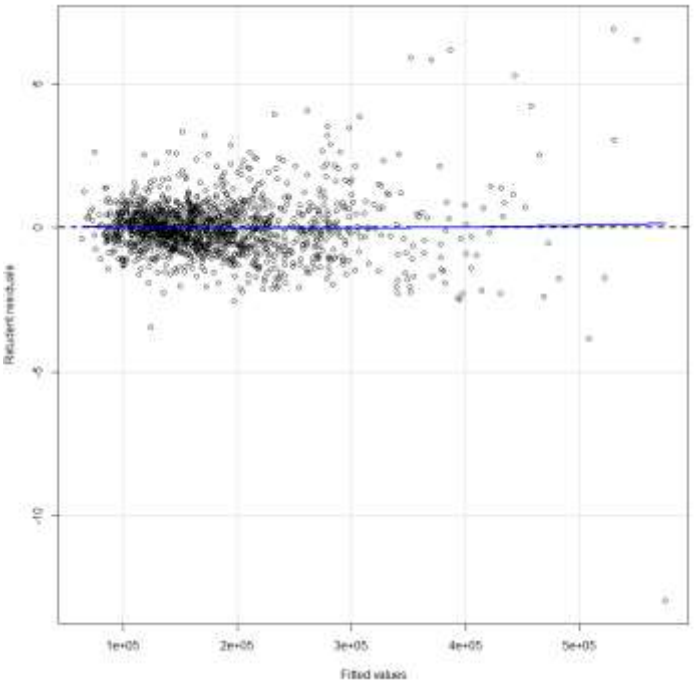

Figure 19 Residuals Plot for the model with polynomials

5.4 Predictions

As shown in Commands 6, in order to assess the out of sample predictive ability of the model we used leave-one-out and 10-fold cross-validation on our train dataset. These procedures were implemented on the three models mentioned above in order to find the minimum RMSE, which indicates the minimum deviation. The 10-fold cross validation is a procedure on which we are using a part of our dataset as test and we train our model on the rest 9 parts of dataset. In our case we are going to use 10-fold cross validation in the model with the log(SalePrice). The results are shown at Table 15.

RMSE =0.13 and MAE = 0.097 indicate the range of the error between the predicted values and the actual values. As these two are less than 1, it is an acceptable error. Rsquared = 0.87 indicates that our model can explain about 87% of our observed data.

In order to simulate the predictive models described above, we used the test dataset 'ames_iowa_housing_test.csv' given in order to see how the model performs. As shown in Commands 7, transformations were applied on test dataset, so as to have the same format as the training data .

We calculated the RMSE's of the same models, the first one with 15 attributes , the second one with the log(SalePrice) and the third one with the polynomials. After comparing the results, we ended up that the model with the log(SalePrice) has the minimum RMSE as shown in Table 16. We can observe that we have similar results (slightly higher value) on MAE and RMSE with the ones that occurred from cross-validation at training dataset, so the fact that we have little difference here means that our model has a good fit on our data 74%.

Concluding we have a model which can make predictions about the sale price of properties based on 11 variables and can fit well on both test and train data. Reading Table 17 we can have a brief interpretation of final model from test dataset.

5.5 Interpretation of the final model

Taking into consideration what mentioned above, we have minimum RMSE and furthermore less deviation with the second model of the log(SalePrice) At Table 13 we can see the results from the summary of the model.

The positive intercept indicates that, when all variables are zero, we will earn a fixed amount of $e^{6.54}$ = 692,28$. The model can predict a relationship identified on data over a limited range. Zero values on all variables are not included in this limited range and as a result the model does not perform well in this case.

Regarding the estimate of the coefficients, if we increase by 1 unit one of them, without updating the others, we will have an update at SalePrice by β. For example, if we increase the 'Overall Quality' by 1, this will increase by $e^{0.106}$ = 1.112$ in the sale price of the house.

Residual standard error is a measure, which shows how well the model fits the data. It represents the average distance that the observed and the predicted values fall. In this case, where it has a small value indicates that the model prediction will be very accurate.

The p-value is the probability that the variable is not relevant. More specific, it tests the null hypothesis where the coefficient is equal to zero. The p-values that are smaller than 0.05, indicate that we must reject the null hypothesis. If a predictor has a low p-value is an important addition to our model because it changes the value related to our target variable.

The 87,51% of the variability is explained by this model, as indicates the Adjusted R-squared, which is the percent of the standard deviation. As the R squared is greater than 70%, we have a good, fitted model. We take into concern the Adjusted R-squared , as we have one predictor variable and 11 target variables, and not Multiple R-squared which tests one predictor with one target variable.

5. Further analysis

A further analysis to the predictive model that was created, would be to examine the expected 'SalePrice' when all variables are centered to the mean. More specific, we aim to find the expected sales price for a typical/ordinary house profile.

In order to achieve the above assumption, we re-scaled the 11 variables of our final model to their mean, as shown in Commands 8. New model was created due to the updated attributes. As our target is variable 'SalePrice', we excluded it from the update.

```
Call:
lm(formula = log(SalePrice) ~ Overall.Qual + Gr.Liv.Area + Total.Bsmt.SF +
    Garage.Area + Year.Built + Fireplaces + BsmtFin.SF.1 + Lot.Frontage +
    Lot.Area + Bsmt.Exposure + Kitchen.Qual, data = test_df_new)

Residuals:
     Min       1Q   Median       3Q      Max
-1.46797 -0.08668  0.02245  0.10920  0.47634

Coefficients:
                Estimate Std. Error  t value Pr(>|t|)
(Intercept)    1.201e+01  8.750e-03 1372.337  < 2e-16 ***
Overall.Qual   9.808e-02  1.044e-02    9.394  < 2e-16 ***
Gr.Liv.Area    1.727e-04  2.489e-05    6.938 1.27e-11 ***
Total.Bsmt.SF  1.578e-06  3.047e-05    0.052  0.95872
Garage.Area    2.774e-04  5.699e-05    4.867 1.53e-06 ***
Year.Built     2.736e-03  3.933e-04    6.956 1.13e-11 ***
Fireplaces     7.908e-02  1.594e-02    4.961 9.71e-07 ***
BsmtFin.SF.1  -3.282e-05  2.422e-05   -1.355  0.17599
Lot.Frontage  -9.504e-04  5.485e-04   -1.733  0.08373 .
Lot.Area       1.195e-06  2.674e-06    0.447  0.65522
Bsmt.Exposure -1.869e-02  8.619e-03   -2.169  0.03056 *
Kitchen.Qual  -4.104e-02  1.289e-02   -3.185  0.00154 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1957 on 488 degrees of freedom
Multiple R-squared:  0.7607,    Adjusted R-squared:  0.7553
F-statistic:   141 on 11 and 488 DF,  p-value: < 2.2e-16
```

Table 18 Summary of test model with rescaled values

Comparing the results of the summary at Table 18 with the result of our final model at Table 13, we observe that there is reduction at values of Intercept and Rsquared, which is reasoning, as the extreme values have been replaced with the mean.

6. Conclusions and Discussion

The main scope of this analysis was to predict sales price of the properties in Ames, Iowa based on data from 2006 to 2012. This goal achieved with the use of a linear regression model, which indicates that the sales price depends on 11 attributes, among them the overall quality of the house, the year that the house built, the existence of garage area, fireplace and basement, as also the lot area.

As we discussed, it was not possible to fix all the linear models' assumptions, more specific the normality and linearity of the residuals. This may occur due to the methods we chose to clean and transform missing values from our dataset. Converting the categorical variables to dummies lead to miscalculation of linear model. Although, the variability of the model is greater than 70% and can be explained, a different manipulation of categorical variables, as well more recent data from Ames, Iowa would be more accurate and probably could lead us to better results.

Appendix A: Tables and Figures

```
 [1] "SalePrice"              "Overall.Qual"           "Gr.Liv.Area"            "Garage.Cars"            "Total.Bsmt.SF"
 [6] "Garage.Area"            "X1st.Flr.SF"            "Full.Bath"              "Year.Built"             "Year.Remod.Add"
[11] "TotRms.AbvGrd"          "Foundation_PConc"       "Mas.Vnr.Area"           "Fireplaces"             "BsmtFin.SF.1"
[16] "Neighborhood_NridgHt"   "Sale.Type_New"          "Sale.Condition_Partial" "MS.SubClass_60"         "Lot.Frontage"
[21] "Garage.Type_Attchd"     "Neighborhood_NoRidge"   "Exterior.1st_VinylSd"   "Open.Porch.SF"          "Exterior.2nd_VinylSd"
[26] "Mas.Vnr.Type_Stone"     "X2nd.Flr.SF"            "Wood.Deck.SF"           "Roof.Style_Hip"         "Half.Bath"
[31] "Lot.Area"               "Paved.Drive"            "Bsmt.Full.Bath"         "Central.Air_Y"          "Central.Air_N"
[36] "Roof.Style_Gable"       "MS.Zoning_RM"           "Lot.Shape"              "Foundation_CBlock"      "Bsmt.Exposure"
[41] "Garage.Type_Detchd"     "Mas.Vnr.Type_None"      "Heating.QC"             "Garage.Finish"          "Bsmt.Qual"
[46] "Kitchen.Qual"           "Exter.Qual"
```

Table 2 Highly correlated attributes to 'SalePrice' produced from Correlation plot

```
Coefficients: (1 not defined because of singularities)
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)          -3.685e+05  1.435e+05  -2.569 0.010311 *
Overall.Qual          1.294e+04  9.954e+02  13.000  < 2e-16 ***
Gr.Liv.Area           2.692e+01  1.466e+01   1.835 0.066642 .
Garage.Cars          -1.477e+03  2.446e+03  -0.604 0.546055
Total.Bsmt.SF         8.165e+00  3.235e+00   2.524 0.011708 *
Garage.Area           3.456e+01  8.545e+00   4.045 5.51e-05 ***
X1st.Flr.SF           2.710e+01  1.494e+01   1.814 0.069893 .
Full.Bath            -8.875e+01  2.198e+03  -0.040 0.967795
Year.Built            7.955e+01  5.767e+01   1.380 0.167943
Year.Remod.Add        1.261e+02  5.292e+01   2.383 0.017278 *
TotRms.AbvGrd        -7.210e+02  8.730e+02  -0.826 0.409029
Foundation_PConc      6.537e+02  3.415e+03   0.191 0.848204
Mas.Vnr.Area          3.902e+01  6.665e+00   5.854 5.92e-09 ***
Fireplaces            5.637e+03  1.404e+03   4.015 6.24e-05 ***
BsmtFin.SF.1          2.195e+01  2.517e+00   8.720  < 2e-16 ***
Neighborhood_NridgHt  1.623e+04  4.031e+03   4.027 5.94e-05 ***
Sale.Type_New        -1.604e+04  2.877e+04  -0.558 0.577099
Sale.Condition_Partial 3.593e+04  2.858e+04   1.257 0.208807
MS.SubClass_60        1.200e+03  3.317e+03   0.362 0.717646
Lot.Frontage          7.698e+01  4.295e+01   1.792 0.073318 .
Garage.Type_Attchd    2.791e+03  2.450e+03   1.139 0.254778
Neighborhood_NoRidge  3.399e+04  5.436e+03   6.253 5.28e-10 ***
Exterior.1st_VinylSd  4.488e+03  8.732e+03   0.514 0.607363
Open.Porch.SF        -4.916e+00  1.278e+01  -0.385 0.700545
Exterior.2nd_VinylSd -8.358e+03  8.721e+03  -0.958 0.338012
Mas.Vnr.Type_Stone    8.708e+03  3.200e+03   2.721 0.006588 **
X2nd.Flr.SF           1.877e+01  1.467e+01   1.279 0.201142
Wood.Deck.SF          4.152e+00  6.378e+00   0.651 0.515147
Roof.Style_Hip        1.822e+03  5.973e+03   0.305 0.760415
Half.Bath             1.639e+02  2.173e+03   0.075 0.939905
Lot.Area              5.564e-01  9.940e-02   5.598 2.59e-08 ***
Paved.Drive           2.170e+03  1.628e+03   1.333 0.182761
Bsmt.Full.Bath       -5.424e+02  1.887e+03  -0.287 0.773788
Central.Air_Y         4.026e+03  3.449e+03   1.167 0.243240
Central.Air_N                NA         NA      NA       NA
Roof.Style_Gable     -4.398e+03  5.699e+03  -0.772 0.440395
MS.Zoning_RM         -4.535e+03  2.544e+03  -1.783 0.074853 .
Lot.Shape            -8.577e+02  5.598e+02  -1.532 0.125695
Foundation_CBlock    -1.025e+02  2.972e+03  -0.034 0.972489
Bsmt.Exposure        -2.661e+03  7.540e+02  -3.529 0.000431 ***
Garage.Type_Detchd    4.775e+03  2.950e+03   1.619 0.105718
Mas.Vnr.Type_None     1.344e+04  2.307e+03   5.825 7.04e-09 ***
Heating.QC           -1.318e+03  5.396e+02  -2.442 0.014710 *
Garage.Finish        -1.796e+03  8.014e+02  -2.241 0.025171 *
Bsmt.Qual            -1.692e+03  6.687e+02  -2.530 0.011496 *
Kitchen.Qual         -4.876e+03  8.470e+02  -5.757 1.04e-08 ***
Exter.Qual           -9.706e+03  1.657e+03  -5.859 5.74e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 28310 on 1454 degrees of freedom
Multiple R-squared:  0.8795,    Adjusted R-squared:  0.8758
F-statistic: 235.9 on 45 and 1454 DF,  p-value: < 2.2e-16
```

Table 2 Summary of model after Correlation plot

```
 [1] "(Intercept)"            "Overall.Qual"           "Gr.Liv.Area"            "Total.Bsmt.SF"          "Garage.Area"
 [6] "X1st.Flr.SF"            "Year.Built"             "Mas.Vnr.Area"           "Fireplaces"             "BsmtFin.SF.1"
[11] "Neighborhood_NridgHt"   "Sale.Type_New"          "Sale.Condition_Partial" "Lot.Frontage"           "Neighborhood_NoRidge"
[16] "Lot.Area"               "Bsmt.Exposure"          "Garage.Finish"          "Bsmt.Qual"              "Kitchen.Qual"
[21] "Exter.Qual"
```

Table 3 Attributes produced from Lasso

```
Call:
lm(formula = SalePrice ~ Overall.Qual + Gr.Liv.Area + Total.Bsmt.SF +
    Garage.Area + X1st.Flr.SF + Year.Built + Mas.Vnr.Area + Fireplaces +
    BsmtFin.SF.1 + Neighborhood_NridgHt + Sale.Type_New + Sale.Condition_Partial +
    Lot.Frontage + Neighborhood_NoRidge + Lot.Area + Bsmt.Exposure +
    Garage.Finish + Bsmt.Qual + Kitchen.Qual + Exter.Qual, data = houses59_num)

Residuals:
    Min      1Q  Median      3Q     Max
-392317  -14864     -75   14055  220006

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           -1.638e+05  7.425e+04  -2.206 0.027551 *
Overall.Qual           1.393e+04  9.490e+02  14.675  < 2e-16 ***
Gr.Liv.Area            4.331e+01  2.278e+00  19.014  < 2e-16 ***
Total.Bsmt.SF          9.163e+00  3.192e+00   2.871 0.004153 **
Garage.Area            3.257e+01  4.960e+00   6.567 7.09e-11 ***
X1st.Flr.SF            7.947e+00  3.530e+00   2.251 0.024514 *
Year.Built             1.056e+02  3.720e+01   2.838 0.004609 **
Mas.Vnr.Area           1.805e+01  5.255e+00   3.435 0.000608 ***
Fireplaces             5.894e+03  1.369e+03   4.305 1.78e-05 ***
BsmtFin.SF.1           2.318e+01  2.111e+00  10.979  < 2e-16 ***
Neighborhood_NridgHt   1.673e+04  3.957e+03   4.227 2.51e-05 ***
Sale.Type_New         -1.210e+04  2.906e+04  -0.416 0.677276
Sale.Condition_Partial 3.348e+04  2.888e+04   1.160 0.246425
Lot.Frontage           8.572e+01  4.098e+01   2.091 0.036659 *
Neighborhood_NoRidge   3.684e+04  5.383e+03   6.843 1.13e-11 ***
Lot.Area               6.088e-01  9.951e-02   6.118 1.21e-09 ***
Bsmt.Exposure         -2.486e+03  7.411e+02  -3.354 0.000815 ***
Garage.Finish         -1.461e+03  7.586e+02  -1.926 0.054348 .
Bsmt.Qual             -1.782e+03  6.282e+02  -2.837 0.004618 **
Kitchen.Qual          -5.409e+03  8.234e+02  -6.569 6.98e-11 ***
Exter.Qual            -1.065e+04  1.647e+03  -6.470 1.33e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 28830 on 1479 degrees of freedom
Multiple R-squared:  0.8729,    Adjusted R-squared:  0.8712
F-statistic:   508 on 20 and 1479 DF,  p-value: < 2.2e-16
```

Table 4 Summary of model after Lasso

```
Call:
lm(formula = SalePrice ~ Overall.Qual + Gr.Liv.Area + Total.Bsmt.SF +
    Garage.Area + X1st.Flr.SF + Year.Built + Mas.Vnr.Area + Fireplaces +
    BsmtFin.SF.1 + Neighborhood_NridgHt + Lot.Frontage + Neighborhood_NoRidge +
    Lot.Area + Bsmt.Exposure + Bsmt.Qual + Kitchen.Qual + Exter.Qual,
    data = houses59_num)

Residuals:
    Min      1Q  Median      3Q     Max
-379135  -14998    -603   13916  219009

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           -2.512e+05  7.206e+04  -3.486 0.000504 ***
Overall.Qual           1.424e+04  9.604e+02  14.832  < 2e-16 ***
Gr.Liv.Area            4.373e+01  2.301e+00  19.008  < 2e-16 ***
Total.Bsmt.SF          1.040e+01  3.234e+00   3.217 0.001323 **
Garage.Area            3.136e+01  4.859e+00   6.453 1.48e-10 ***
X1st.Flr.SF            7.663e+00  3.586e+00   2.137 0.032754 *
Year.Built             1.516e+02  3.627e+01   4.180 3.08e-05 ***
Mas.Vnr.Area           1.785e+01  5.339e+00   3.343 0.000849 ***
Fireplaces             5.739e+03  1.387e+03   4.139 3.69e-05 ***
BsmtFin.SF.1           2.122e+01  2.114e+00  10.035  < 2e-16 ***
Neighborhood_NridgHt   1.984e+04  3.991e+03   4.972 7.39e-07 ***
Lot.Frontage           1.025e+02  4.151e+01   2.470 0.013636 *
Neighborhood_NoRidge   3.274e+04  5.432e+03   6.028 2.09e-09 ***
Lot.Area               6.096e-01  1.011e-01   6.030 2.07e-09 ***
Bsmt.Exposure         -2.758e+03  7.517e+02  -3.669 0.000252 ***
Bsmt.Qual             -2.108e+03  6.335e+02  -3.328 0.000895 ***
Kitchen.Qual          -5.790e+03  8.344e+02  -6.939 5.90e-12 ***
Exter.Qual            -1.214e+04  1.658e+03  -7.321 4.02e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 29290 on 1482 degrees of freedom
Multiple R-squared:  0.8686,    Adjusted R-squared:  0.8671
F-statistic: 576.1 on 17 and 1482 DF,  p-value: < 2.2e-16
```

Table 5 Summary of model after Stepwise procedure

Overall.Qual	Gr.Liv.Area	Total.Bsmt.SF	Garage.Area	X1st.Flr.SF	Year.Built	Mas.Vnr.Area
3.10	2.40	3.39	1.84	3.27	2.14	1.47
Fireplaces	BsmtFin.SF.1	Neighborhood_NridgHt	Lot.Frontage	Neighborhood_NoRidge	Lot.Area	Bsmt.Exposure
1.44	1.52	1.32	1.35	1.24	1.24	1.30
Bsmt.Qual	Kitchen.Qual	Exter.Qual				
2.07	1.94	2.23				

Table 6 VIF to check multicollinearity

[1] "SalePrice"	"4.0385622384619e-34"
[1] "Overall.Qual"	"5.03502898097296e-23"
[1] "Gr.Liv.Area"	"6.26484471107668e-25"
[1] "Total.Bsmt.SF"	"1.11179352832166e-17"
[1] "Garage.Area"	"2.87710392509189e-13"
[1] "X1st.Flr.SF"	"2.26331887158489e-24"
[1] "Exter.Qual"	"1.33632294874834e-48"
[1] "Year.Built"	"2.08410223673854e-26"
[1] "Mas.Vnr.Area"	"1.02826536359726e-49"
[1] "Fireplaces"	"1.26191731017475e-42"
[1] "BsmtFin.SF.1"	"3.638866441706e-32"
[1] "Lot.Frontage"	"1.59613481175636e-31"
[1] "Neighborhood_NridgHt"	"1.4170542353249e-61"
[1] "Neighborhood_NoRidge"	"1.78835693088115e-63"
[1] "Lot.Area"	"3.3707357040062e-57"
[1] "Bsmt.Exposure"	"3.37204946407452e-45"
[1] "Bsmt.Qual"	"3.41566955124203e-41"
[1] "Kitchen.Qual"	"7.1186499397217e-43"

Table 7 Check Normality with Shapiro Test

```
Call:
lm(formula = SalePrice ~ Overall.Qual + Gr.Liv.Area + Total.Bsmt.SF +
    Garage.Area + Exter.Qual + Year.Built + Mas.Vnr.Area + Fireplaces +
    BsmtFin.SF.1 + Lot.Frontage + Lot.Area + Bsmt.Exposure +
    Bsmt.Qual + Kitchen.Qual, data = final_df)

Residuals:
    Min      1Q  Median      3Q     Max
-399436  -15339   -1039   14884  237863

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -2.309e+05  7.295e+04  -3.165 0.001582 **
Overall.Qual   1.438e+04  9.716e+02  14.804  < 2e-16 ***
Gr.Liv.Area    4.672e+01  2.210e+00  21.139  < 2e-16 ***
Total.Bsmt.SF  1.576e+01  2.578e+00   6.114 1.24e-09 ***
Garage.Area    3.348e+01  4.919e+00   6.806 1.45e-11 ***
Exter.Qual    -1.268e+04  1.679e+03  -7.554 7.32e-14 ***
Year.Built     1.407e+02  3.676e+01   3.827 0.000135 ***
Mas.Vnr.Area   2.866e+01  5.221e+00   5.489 4.74e-08 ***
Fireplaces     5.908e+03  1.398e+03   4.226 2.52e-05 ***
BsmtFin.SF.1   2.160e+01  2.145e+00  10.068  < 2e-16 ***
Lot.Frontage   1.401e+02  4.136e+01   3.387 0.000725 ***
Lot.Area       5.620e-01  1.025e-01   5.485 4.85e-08 ***
Bsmt.Exposure -2.305e+03  7.616e+02  -3.026 0.002522 **
Bsmt.Qual     -2.599e+03  6.394e+02  -4.065 5.06e-05 ***
Kitchen.Qual  -6.047e+03  8.457e+02  -7.150 1.36e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 29800 on 1485 degrees of freedom
Multiple R-squared:  0.8637,    Adjusted R-squared:  0.8625
F-statistic: 672.4 on 14 and 1485 DF,  p-value: < 2.2e-16
```

Table 10 Summary of model occurred after Lasso and Stepwise

```
Call:
lm(formula = SalePrice ~ Overall.Qual + Gr.Liv.Area + Total.Bsmt.SF +
    Garage.Area + Exter.Qual + Year.Built + Mas.Vnr.Area + Fireplaces +
    BsmtFin.SF.1 + Lot.Frontage + Lot.Area + Bsmt.Exposure +
    Bsmt.Qual + Kitchen.Qual + Exter.Qual - 1, data = final_df)

Residuals:
    Min      1Q  Median      3Q     Max
-399516  -15471    -501   14687  237813

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
Overall.Qual   1.487e+04  9.623e+02  15.453  < 2e-16 ***
Gr.Liv.Area    4.525e+01  2.167e+00  20.878  < 2e-16 ***
Total.Bsmt.SF  1.626e+01  2.581e+00   6.301 3.88e-10 ***
Garage.Area    3.629e+01  4.853e+00   7.478 1.29e-13 ***
Exter.Qual    -1.304e+04  1.680e+03  -7.760 1.57e-14 ***
Year.Built     2.543e+01  5.030e+00   5.055 4.83e-07 ***
Mas.Vnr.Area   2.989e+01  5.222e+00   5.723 1.27e-08 ***
Fireplaces     5.877e+03  1.402e+03   4.191 2.94e-05 ***
BsmtFin.SF.1   2.190e+01  2.150e+00  10.188  < 2e-16 ***
Lot.Frontage   1.373e+02  4.148e+01   3.309 0.000958 ***
Lot.Area       5.389e-01  1.025e-01   5.257 1.68e-07 ***
Bsmt.Exposure -2.529e+03  7.606e+02  -3.326 0.000904 ***
Bsmt.Qual     -3.443e+03  5.828e+02  -5.908 4.27e-09 ***
Kitchen.Qual  -6.202e+03  8.468e+02  -7.324 3.92e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 29890 on 1486 degrees of freedom
Multiple R-squared:  0.9773,    Adjusted R-squared:  0.9771
F-statistic:  4578 on 14 and 1486 DF,  p-value: < 2.2e-16
```

Table 11 Summary of model without Intercept

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 2222.96, Df = 1, p = < 2.22e-16
```

Table 12 Non-constant Variance Test

```
Call:
lm(formula = log(SalePrice) ~ Overall.Qual + Gr.Liv.Area + Total.Bsmt.SF +
    Garage.Area + Year.Built + Fireplaces + BsmtFin.SF.1 + Lot.Frontage +
    Lot.Area + Bsmt.Exposure + Kitchen.Qual, data = final_df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.69093 -0.06620  0.00520  0.07994  0.45528

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.542e+00  3.103e-01  21.085  < 2e-16 ***
Overall.Qual   1.060e-01  4.501e-03  23.544  < 2e-16 ***
Gr.Liv.Area    2.116e-04  1.039e-05  20.375  < 2e-16 ***
Total.Bsmt.SF  8.000e-05  1.207e-05   6.627 4.78e-11 ***
Garage.Area    2.046e-04  2.343e-05   8.732  < 2e-16 ***
Year.Built     2.183e-03  1.594e-04  13.688  < 2e-16 ***
Fireplaces     4.712e-02  6.649e-03   7.087 2.11e-12 ***
BsmtFin.SF.1   8.443e-05  1.016e-05   8.306  < 2e-16 ***
Lot.Frontage   7.836e-04  1.972e-04   3.975 7.39e-05 ***
Lot.Area       2.293e-06  4.889e-07   4.689 2.99e-06 ***
Bsmt.Exposure -9.746e-03  3.618e-03  -2.694  0.00715 **
Kitchen.Qual  -1.974e-02  3.642e-03  -5.420 6.94e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1423 on 1488 degrees of freedom
Multiple R-squared:  0.8761,    Adjusted R-squared:  0.8751
F-statistic: 956.1 on 11 and 1488 DF,  p-value: < 2.2e-16
```

Table 13 Summary of model with log(SalePrice)

```
Call:
lm(formula = SalePrice ~ I(Overall.Qual^5) + Total.Bsmt.SF +
    Garage.Area + Year.Built + Mas.Vnr.Area + Fireplaces + BsmtFin.SF.1 +
    Lot.Frontage + Lot.Area + Bsmt.Qual + Kitchen.Qual, data = final_df)

Residuals:
    Min      1Q  Median      3Q     Max
-389843  -16464   -1271   14306  215359

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -2.546e+05  7.635e+04  -3.335 0.000875 ***
I(Overall.Qual^5)  2.347e+00  8.805e-02  26.655  < 2e-16 ***
Total.Bsmt.SF      2.203e+01  2.744e+00   8.028 1.99e-15 ***
Garage.Area        5.043e+01  5.252e+00   9.602  < 2e-16 ***
Year.Built         1.781e+02  3.819e+01   4.664 3.38e-06 ***
Mas.Vnr.Area       2.997e+01  5.700e+00   5.259 1.66e-07 ***
Fireplaces         1.576e+04  1.438e+03  10.958  < 2e-16 ***
BsmtFin.SF.1       1.371e+01  2.280e+00   6.016 2.24e-09 ***
Lot.Frontage       2.253e+02  4.446e+01   5.068 4.54e-07 ***
Lot.Area           8.375e-01  1.108e-01   7.559 7.08e-14 ***
Bsmt.Qual         -2.940e+03  6.860e+02  -4.286 1.93e-05 ***
Kitchen.Qual      -6.226e+03  8.766e+02  -7.102 1.89e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32460 on 1488 degrees of freedom
Multiple R-squared:  0.838,     Adjusted R-squared:  0.8368
F-statistic: 699.6 on 11 and 1488 DF,  p-value: < 2.2e-16
```

Table 14 Summary of model with polynomials

```
Linear Regression

1500 samples
  11 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1351, 1351, 1349, 1349, 1350, 1351, ...
Resampling results:

  RMSE     Rsquared   MAE
  0.13928  0.8790556  0.0974508

Tuning parameter 'intercept' was held constant at a value of TRUE
```

Table 15 10-fold cross-validation for model with log(SalePrice)

```
Linear Regression

500 samples
 11 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 451, 449, 449, 450, 450, 450, ...
Resampling results:

  RMSE       Rsquared  MAE
  0.2073462  0.745958  0.1404634

Tuning parameter 'intercept' was held constant at a value of TRUE
```

Table 16 10-fold cross-validation for model with log(SalePrice), test data frame

```
Call:
lm(formula = log(SalePrice) ~ Overall.Qual + Gr.Liv.Area + Total.Bsmt.SF +
    Garage.Area + Year.Built + Fireplaces + BsmtFin.SF.1 + Lot.Frontage +
    Lot.Area + Bsmt.Exposure + Kitchen.Qual, data = test_df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.46797 -0.08668  0.02245  0.10920  0.47634

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.847e+00  7.583e-01   7.712 7.04e-14 ***
Overall.Qual   9.808e-02  1.044e-02   9.394  < 2e-16 ***
Gr.Liv.Area    1.727e-04  2.489e-05   6.938 1.27e-11 ***
Total.Bsmt.SF  1.578e-06  3.047e-05   0.052  0.95872
Garage.Area    2.774e-04  5.699e-05   4.867 1.53e-06 ***
Year.Built     2.736e-03  3.933e-04   6.956 1.13e-11 ***
Fireplaces     7.908e-02  1.594e-02   4.961 9.71e-07 ***
BsmtFin.SF.1  -3.282e-05  2.422e-05  -1.355  0.17599
Lot.Frontage  -9.504e-04  5.485e-04  -1.733  0.08373 .
Lot.Area       1.195e-06  2.674e-06   0.447  0.65522
Bsmt.Exposure -1.869e-02  8.619e-03  -2.169  0.03056 *
Kitchen.Qual  -4.104e-02  1.289e-02  -3.185  0.00154 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1957 on 488 degrees of freedom
Multiple R-squared:  0.7607,    Adjusted R-squared:  0.7553
F-statistic:   141 on 11 and 488 DF,  p-value: < 2.2e-16
```

Table 17 Summary of final test model

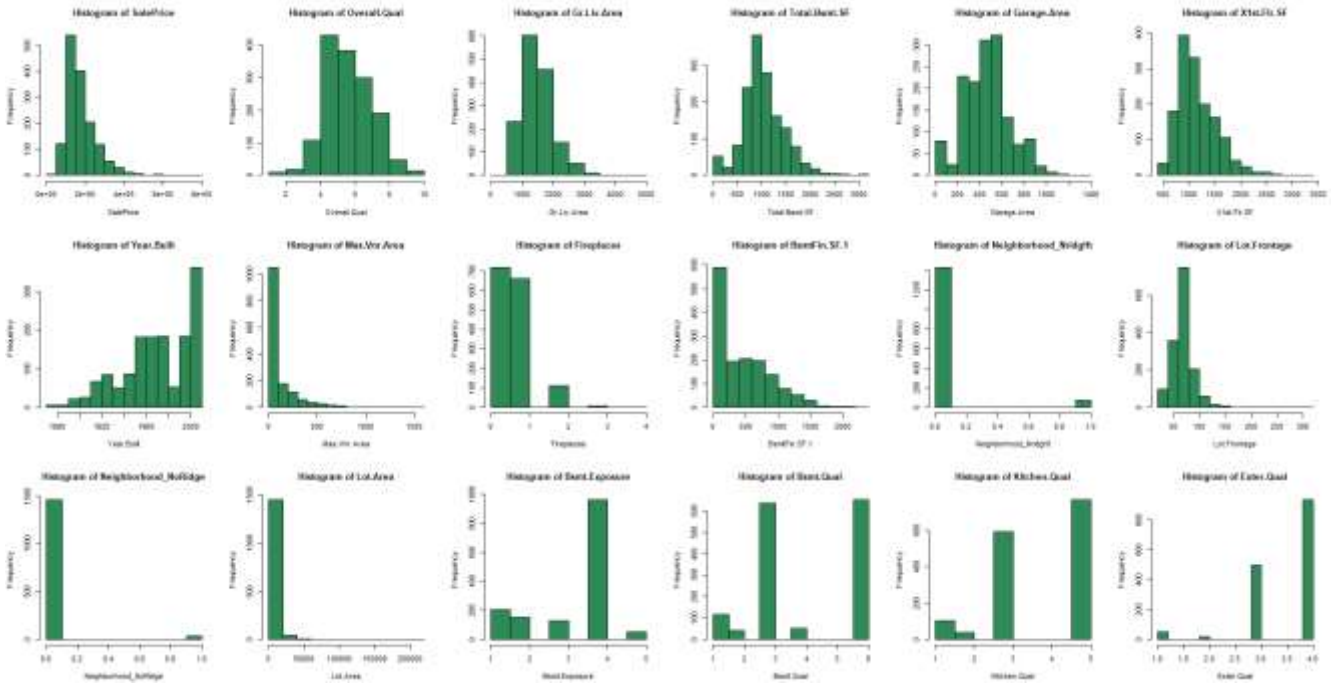

Figure 4 Histograms of most important attributes

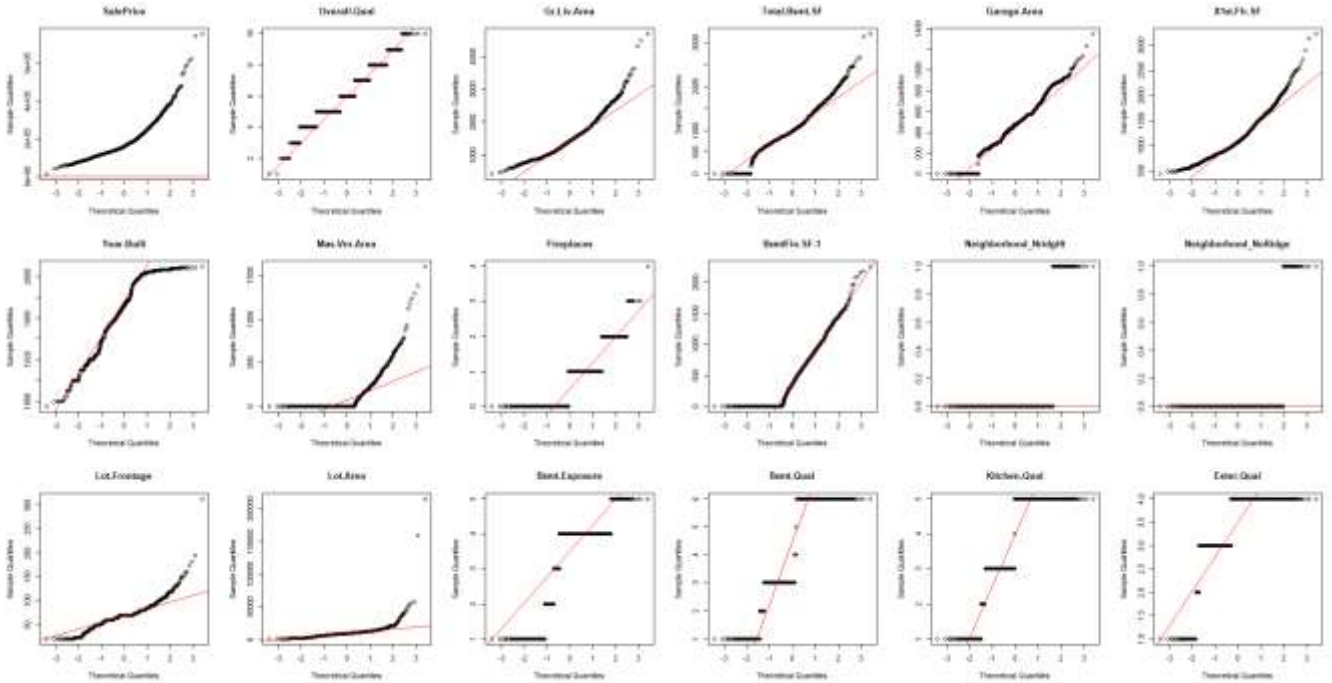

Figure 5 QQ plots of most important attributes

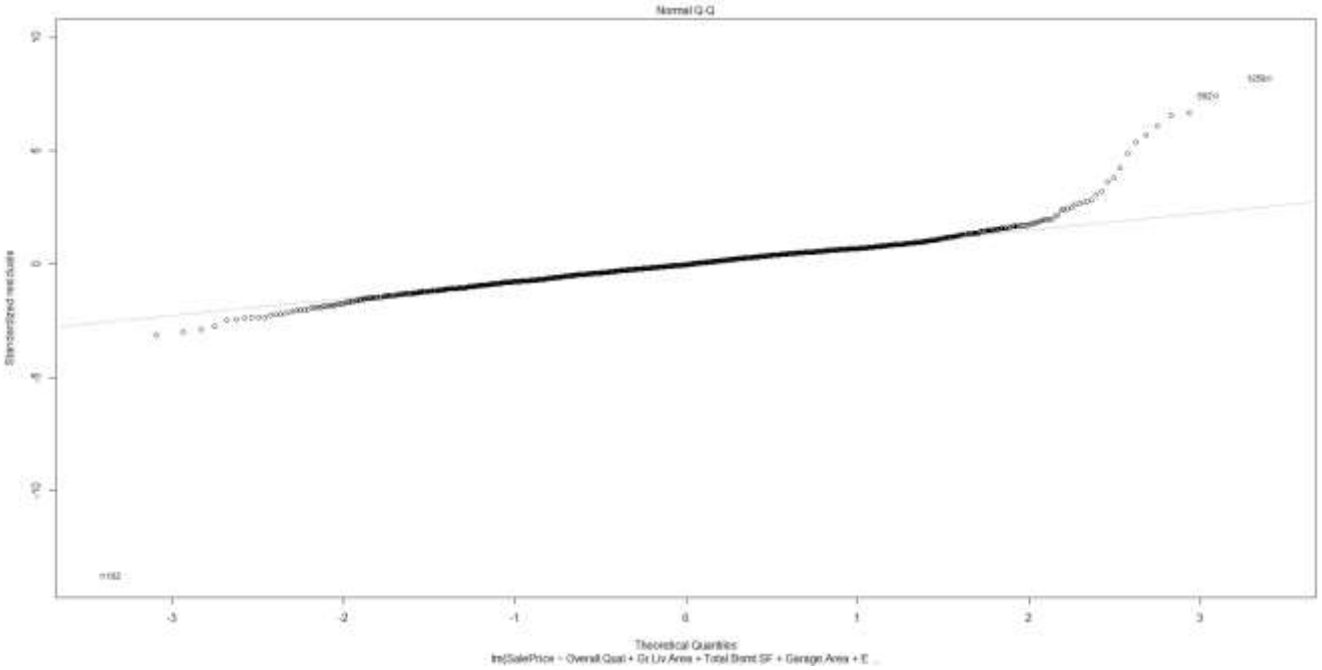

Figure 10 QQ plot for the residuals of null model

	Test stat	Pr(>|Test stat|)	
Overall.Qual	11.0209	< 2.2e-16	***
Gr.Liv.Area	8.3079	< 2.2e-16	***
Total.Bsmt.SF	4.2967	1.847e-05	***
Garage.Area	5.4126	7.233e-08	***
Exter.Qual	6.5102	1.024e-10	***
Year.Built	-1.1389	0.2549112	
Mas.Vnr.Area	4.7979	1.765e-06	***
Fireplaces	0.2508	0.8020057	
BsmtFin.SF.1	3.5132	0.0004561	***
Lot.Frontage	-1.2520	0.2107830	
Lot.Area	-2.1348	0.0329392	*
Bsmt.Exposure	1.1602	0.2461351	
Bsmt.Qual	8.9145	< 2.2e-16	***
Kitchen.Qual	8.1560	7.301e-16	***
Tukey test	15.0467	< 2.2e-16	***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Figure 12 Residuals Plot =F for the full model

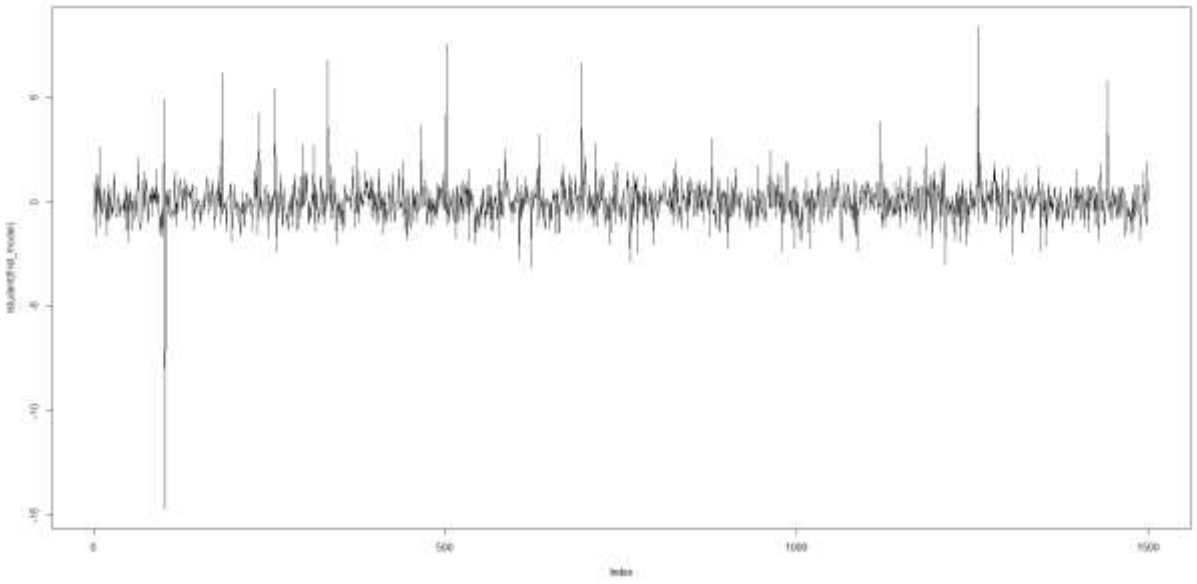

Figure 14 Independence of errors

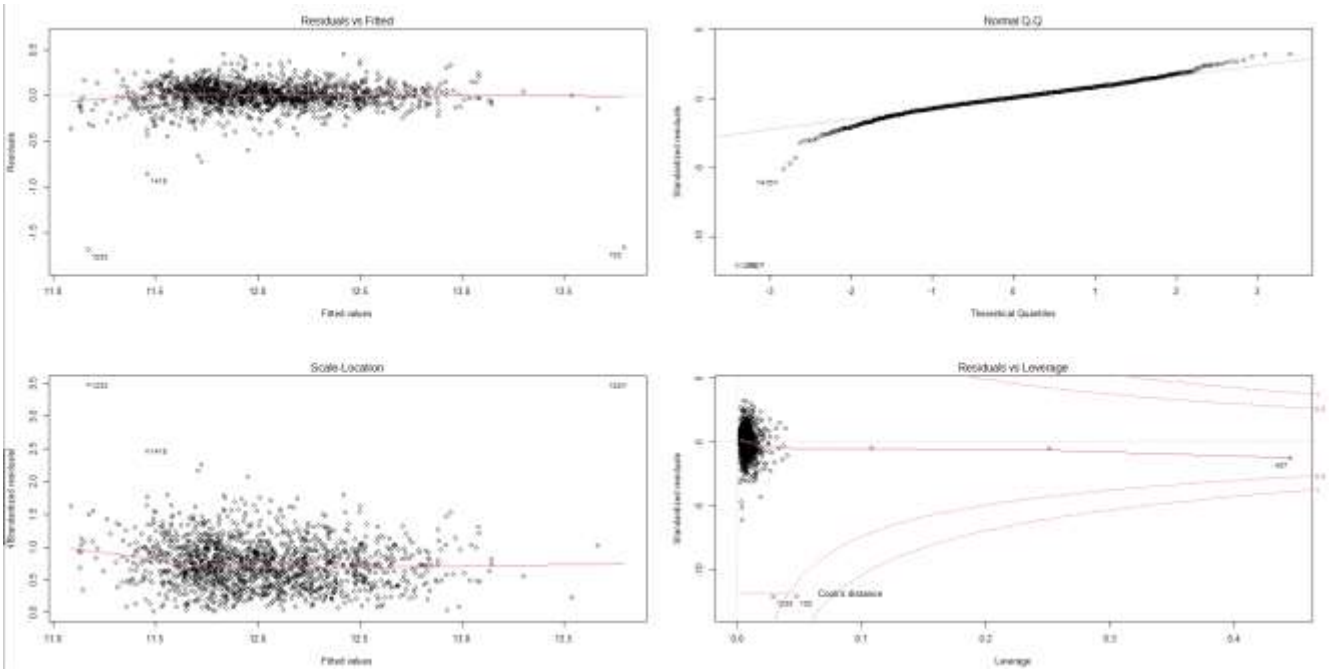

Figure 15 Plot of the residuals with log(SalePrice)

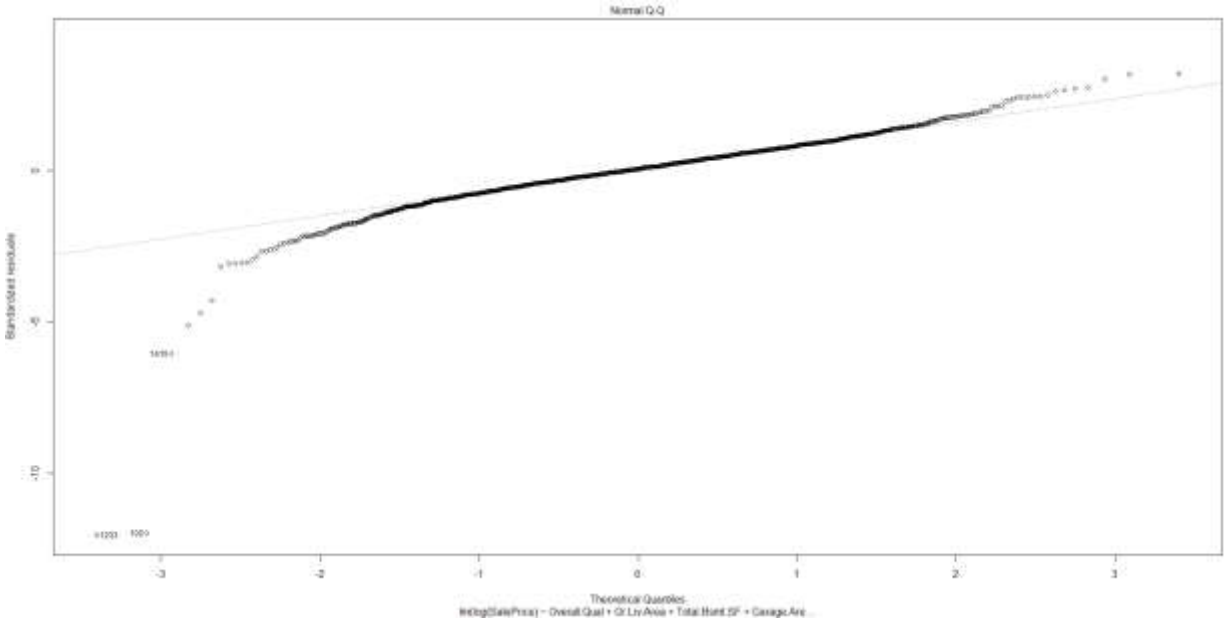

Figure 16 QQ plot for the residuals of model with log(SalePrice)

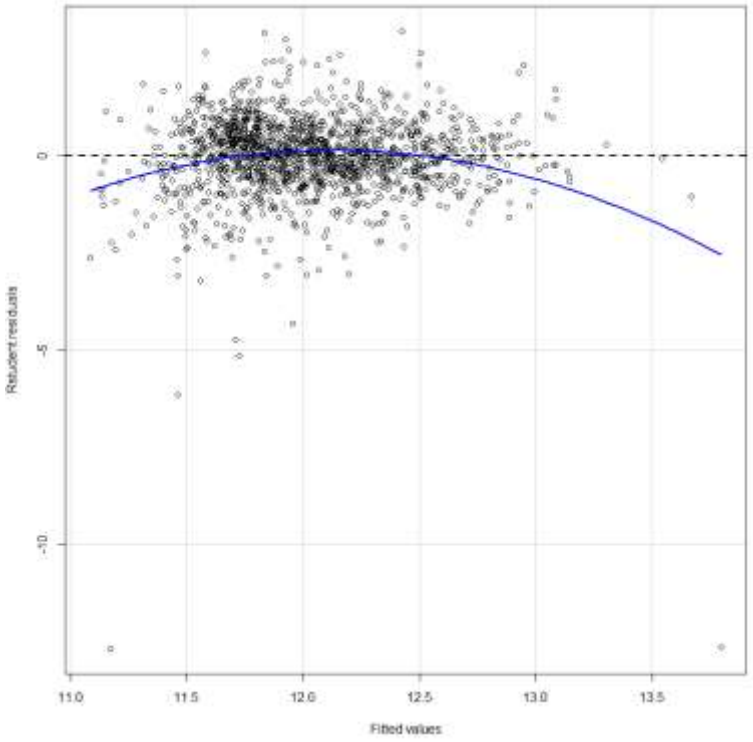

Figure 17 Residuals Plot for the model with log(SalePrice)

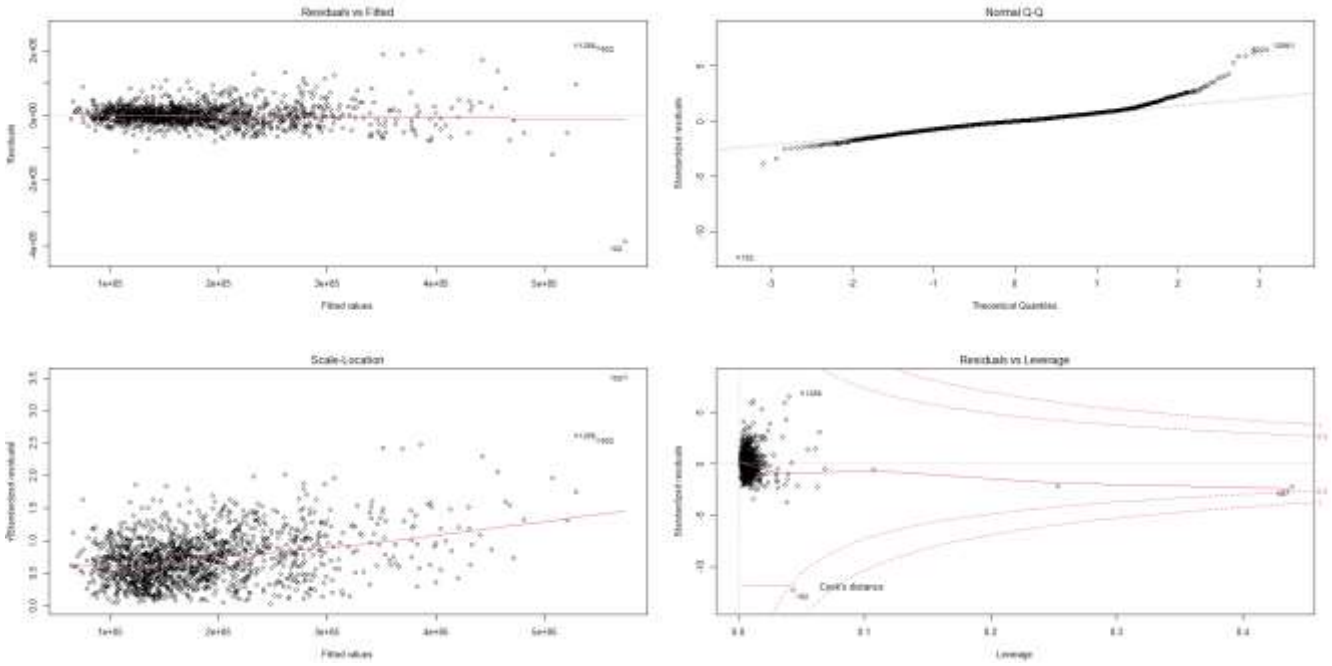

Figure 18 Plot of the residuals for model with polynomials

Appendix B: Useful links

Plot Two Continuous Variables: Scatter Graph and Alternatives
http://www.sthda.com/english/articles/32-r-graphics-essentials/131-plot-two-continuous-variables-scatter-graph-and-alternatives/

Ggplot2 scatter plots
http://www.sthda.com/english/wiki/ggplot2-scatter-plots-quick-start-guide-r-software-and-data-visualization

Quick Tutorial On LASSO Regression
https://rstatisticsblog.com/data-science-in-action/machine-learning/lasso-regression/

Cross-Validation in R programming
https://www.geeksforgeeks.org/cross-validation-in-r-programming/

Linear Regression Assumptions and Diagnostics in R: Essentials
http://www.sthda.com/english/articles/39-regression-model-diagnostics/161-linear-regression-assumptions-and-diagnostics-in-r-essentials/

Evaluating Logistic Regression Models
https://www.r-bloggers.com/2015/08/evaluating-logistic-regression-models/

Model Training and Tuning
https://topepo.github.io/caret/model-training-and-tuning.html

Predict in R: Model Predictions and Confidence Intervals
http://www.sthda.com/english/articles/40-regression-analysis/166-predict-in-r-model-predictions-and-confidence-intervals/

Appendix C: R source code

COMMANDS 1

```
#Read dataset
houses59 <- read.csv('ames_iowa_housing_59.csv', header = TRUE, sep = ';')

#Describe dataset
str(houses59)
dim(houses59)
names(houses59)
View(houses59)

#PID, MS.Subclass, Overall.Qual and Overall.Cond not defined correctly
#Convert PID and MS.SubClass to factor as it is nominal
houses59$PID <- as.character(houses59$PID)
houses59$MS.SubClass <- as.character(houses59$MS.SubClass)

#Data Cleaning - Null replacement
#Count NAs for each variable using apply
sapply(houses59, function(x) sum(is.na(x)))

#New DF without 'X', 'PID', Alley', 'Pool.QC', 'Fence', 'Misc.Feature', as more than 80% values were
missing
houses59 <- subset(houses59, select = -c(X,Order,PID,Alley,Pool.QC,Fence,Misc.Feature))

#DISCRETE: replace Null values
houses59$Bsmt.Full.Bath[is.na(houses59$Bsmt.Full.Bath)] <- 0
houses59$Bsmt.Half.Bath[is.na(houses59$Bsmt.Half.Bath)] <- 0
houses59$Garage.Yr.Blt[is.na(houses59$Garage.Yr.Blt)] <- 2099

#CONTINUOUS: replace Null values with 0 and mean
houses59$Lot.Frontage <- ifelse(is.na(houses59$Lot.Frontage), round(mean(houses59$Lot.Frontage,
na.rm=TRUE)), houses59$Lot.Frontage)
houses59$Mas.Vnr.Area[is.na(houses59$Mas.Vnr.Area)] <- 0
houses59$BsmtFin.SF.1[is.na(houses59$BsmtFin.SF.1)] <- 0
houses59$BsmtFin.SF.2[is.na(houses59$BsmtFin.SF.2)] <- 0
houses59$Bsmt.Unf.SF[is.na(houses59$Bsmt.Unf.SF)] <- 0
houses59$Total.Bsmt.SF[is.na(houses59$Total.Bsmt.SF)] <- 0

#Convert Integer to numeric
intIndex <- sapply(houses59, class) == "integer"
houses59[,intIndex] <- lapply(houses59[,intIndex], as.numeric)

#NOMINAL: replace Null values
houses59$Mas.Vnr.Type[is.na(houses59$Mas.Vnr.Type)] <- "NA"
houses59$Garage.Type[is.na(houses59$Garage.Type)] <- "No Garage"

#ORDINAL: replace Null values
houses59$Bsmt.Qual[is.na(houses59$Bsmt.Qual)] <- "No Basement"
```

```
houses59$Bsmt.Cond[is.na(houses59$Bsmt.Cond)] <- "No Basement"
houses59$Bsmt.Exposure[is.na(houses59$Bsmt.Exposure)] <- "No Basement"
houses59$BsmtFin.Type.1[is.na(houses59$BsmtFin.Type.1)] <- "No Basement"
houses59$BsmtFin.Type.2[is.na(houses59$BsmtFin.Type.2)] <- "No Basement"
houses59$Fireplace.Qu[is.na(houses59$Fireplace.Qu)] <- "No Fireplace"
houses59$Garage.Finish[is.na(houses59$Garage.Finish)] <- "No Garage"
houses59$Garage.Qual[is.na(houses59$Garage.Qual)] <- "No Garage"
houses59$Garage.Cond[is.na(houses59$Garage.Cond)] <- "No Garage"

#Convert character to factor
chrIndex <- sapply(houses59, class) == "character"
houses59[,chrIndex] <- lapply(houses59[,chrIndex], as.factor)

names(Filter(is.factor, houses59))
names(Filter(is.numeric, houses59))

#Convert Ordinals Factors to numeric
houses59$Lot.Shape <- as.numeric(houses59$Lot.Shape)
houses59$Utilities <- as.numeric(houses59$Utilities)
houses59$Land.Slope <- as.numeric(houses59$Land.Slope)
houses59$Exter.Qual <- as.numeric(houses59$Exter.Qual)
houses59$Exter.Cond <- as.numeric(houses59$Exter.Cond)
houses59$Bsmt.Qual <- as.numeric(houses59$Bsmt.Qual)
houses59$Bsmt.Cond <- as.numeric(houses59$Bsmt.Cond)
houses59$Bsmt.Exposure <- as.numeric(houses59$Bsmt.Exposure)
houses59$BsmtFin.Type.1 <- as.numeric(houses59$BsmtFin.Type.1)
houses59$BsmtFin.Type.2 <- as.numeric(houses59$BsmtFin.Type.2)
houses59$Heating.QC <- as.numeric(houses59$Heating.QC)
houses59$Electrical <- as.numeric(houses59$Electrical)
houses59$Kitchen.Qual<- as.numeric(houses59$Kitchen.Qual)
houses59$Functional <- as.numeric(houses59$Functional)
houses59$Fireplace.Qu <- as.numeric(houses59$Fireplace.Qu)
houses59$Garage.Finish <- as.numeric(houses59$Garage.Finish)
houses59$Garage.Qual <- as.numeric(houses59$Garage.Qual)
houses59$Garage.Cond <- as.numeric(houses59$Garage.Cond)
houses59$Paved.Drive <- as.numeric(houses59$Paved.Drive)

#Convert Nominal Factors to Dummies
for (level in unique(houses59$MS.SubClass)){houses59[paste("MS.SubClass",level,sep = "_")] <-
ifelse(houses59$MS.SubClass == level,1,0)}
for (level in unique(houses59$MS.Zoning)){houses59[paste("MS.Zoning",level,sep = "_")] <-
ifelse(houses59$MS.Zoning == level,1,0)}
for (level in unique(houses59$Street)){houses59[paste("Street",level,sep = "_")] <-
ifelse(houses59$Street == level,1,0)}
for (level in unique(houses59$Land.Contour)){houses59[paste("Land.Contour",level,sep = "_")] <-
ifelse(houses59$Land.Contour == level,1,0)}
for (level in unique(houses59$Lot.Config)){houses59[paste("Lot.Config",level,sep = "_")] <-
ifelse(houses59$Lot.Config == level,1,0)}
for (level in unique(houses59$Neighborhood)){houses59[paste("Neighborhood",level,sep = "_")] <-
ifelse(houses59$Neighborhood == level,1,0)}
```

```
for (level in unique(houses59$Condition.1)){houses59[paste("Condition.1",level,sep = "_")] <-
ifelse(houses59$Condition.1 == level,1,0)}
for (level in unique(houses59$Condition.2)){houses59[paste("Condition.2",level,sep = "_")] <-
ifelse(houses59$Condition.2 == level,1,0)}
for (level in unique(houses59$Bldg.Type)){houses59[paste("Bldg.Type",level,sep = "_")] <-
ifelse(houses59$Bldg.Type == level,1,0)}
for (level in unique(houses59$House.Style)){houses59[paste("House.Style",level,sep = "_")] <-
ifelse(houses59$House.Style == level,1,0)}
for (level in unique(houses59$Roof.Style)){houses59[paste("Roof.Style",level,sep = "_")] <-
ifelse(houses59$Roof.Style == level,1,0)}
for (level in unique(houses59$Roof.Matl)){houses59[paste("Roof.Matl",level,sep = "_")] <-
ifelse(houses59$Roof.Matl == level,1,0)}
for (level in unique(houses59$Exterior.1st)){houses59[paste("Exterior.1st",level,sep = "_")] <-
ifelse(houses59$Exterior.1st == level,1,0)}
for (level in unique(houses59$Exterior.2nd)){houses59[paste("Exterior.2nd",level,sep = "_")] <-
ifelse(houses59$Exterior.2nd == level,1,0)}
for (level in unique(houses59$Mas.Vnr.Type)){houses59[paste("Mas.Vnr.Type",level,sep = "_")] <-
ifelse(houses59$Mas.Vnr.Type == level,1,0)}
for (level in unique(houses59$Foundation)){houses59[paste("Foundation",level,sep = "_")] <-
ifelse(houses59$Foundation == level,1,0)}
for (level in unique(houses59$Heating)){houses59[paste("Heating",level,sep = "_")] <-
ifelse(houses59$Heating == level,1,0)}
for (level in unique(houses59$Central.Air)){houses59[paste("Central.Air",level,sep = "_")] <-
ifelse(houses59$Central.Air == level,1,0)}
for (level in unique(houses59$Garage.Type)){houses59[paste("Garage.Type",level,sep = "_")] <-
ifelse(houses59$Garage.Type == level,1,0)}
for (level in unique(houses59$Misc.Feature)){houses59[paste("Misc.Feature",level,sep = "_")] <-
ifelse(houses59$Misc.Feature == level,1,0)}
for (level in unique(houses59$Sale.Type)){houses59[paste("Sale.Type",level,sep = "_")] <-
ifelse(houses59$Sale.Type == level,1,0)}
for (level in unique(houses59$Sale.Condition)){houses59[paste("Sale.Condition",level,sep = "_")] <-
ifelse(houses59$Sale.Condition == level,1,0)}

#Create new DF only with numeric variables
numeric <- unlist(lapply(houses59, is.numeric))
houses59_num <- houses59[,numeric]
dim(houses59_num)
str(houses59_num)
```

COMMANDS 2

```
#Correlations - LASSO - Stepwise
#Find correlations
library(corrplot)
correlations <- cor(houses59_num, use="pairwise.complete.obs")

#sort on decreasing correlations with SalePrice variable
cor_sorted <- as.matrix(sort(correlations['SalePrice'], decreasing = TRUE))

#select only high correlations
CorHigh <- names(which(apply(cor_sorted, 1, function(x) (abs(x) < -0.25 || abs(x)> 0.25))))
```

```
correlations <- correlations[CorHigh, CorHigh]
corrplot.mixed(correlations, tl.col="black", tl.pos = "lt", tl.cex = 0.7,cl.cex = .7, number.cex=.7)
CorHigh
#High correlated vars with
SalePrice:Overall.Qual,Gr.Liv.Area,Garage.Cars,Total.Bsmt.SF,Garage.Area,X1st.Flr.SF,Full.Bath,Year.Built,
#Year.Remod.Add,TotRms.AbvGrd,Foundation_PConc, Mas.Vnr.Area

#Conduct Lasso
#Mfull 46 variables
require(glmnet)
library(glmnet)
mfull <- lm(SalePrice ~
Overall.Qual+Gr.Liv.Area+Garage.Cars+Total.Bsmt.SF+Garage.Area+X1st.Flr.SF+Full.Bath+Year.Built+Year.Remod.Add+

TotRms.AbvGrd+Foundation_PConc+Mas.Vnr.Area+Fireplaces+BsmtFin.SF.1+Neighborhood_NridgHt+Sale.Type_New+Sale.Condition_Partial+MS.SubClass_60+Lot.Frontage+Garage.Type_Attchd+Neighborhood_NoRidge+

Exterior.1st_VinylSd+Open.Porch.SF+Exterior.2nd_VinylSd+Mas.Vnr.Type_Stone+X2nd.Flr.SF+Wood.Deck.SF+Roof.Style_Hip+Half.Bath+

Lot.Area+Paved.Drive+Bsmt.Full.Bath+Central.Air_Y+Central.Air_N+Roof.Style_Gable+MS.Zoning_RM+Lot.Shape+Foundation_CBlock+Bsmt.Exposure+

Garage.Type_Detchd+Mas.Vnr.Type_None+Heating.QC+Garage.Finish+Bsmt.Qual+Kitchen.Qual+Exter.Qual, data = houses59_num)

summary(mfull)

X <- model.matrix(mfull)[,-1]
Y <- houses59_num$SalePrice
lasso <- glmnet(X, Y, standardize = TRUE,alpha = 1)

library(plotmo)
plot_glmnet(lasso)

lasso <- cv.glmnet(X,Y, alpha = 1)
plot(lasso)

lasso$lambda.1se#4786.206
lasso$lambda.min#513.2092

lassoModel <- coef(lasso, s = lasso$lambda.1se)
lassoModel
#Return coefficients with values
rownames(coef(lasso, s = 'lambda.1se'))[coef(lasso, s = 'lambda.1se')[,1]!= 0]

#Stepwise with the above coefficients produced by 'lambda.1se'
```

```
#Mstep 20 variables
mstep <- lm(SalePrice ~ Overall.Qual+
Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+X1st.Flr.SF+Year.Built+

Mas.Vnr.Area+Fireplaces+BsmtFin.SF.1+Neighborhood_NridgHt+Sale.Type_New+Sale.Condition_P
artial+

Lot.Frontage+Neighborhood_NoRidge+Lot.Area+Bsmt.Exposure+Garage.Finish+Bsmt.Qual+Kitchen
.Qual+Exter.Qual, data = houses59_num)

summary(mstep)
step(mstep, direction='both')
#Univariate Description of step model
summary(mstep)
anova(mstep)

mstep_after <- lm(SalePrice ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+X1st.Flr.SF+Year.Built+Mas.Vnr.Area+Fire
places+

BsmtFin.SF.1+Neighborhood_NridgHt+Lot.Frontage+Neighborhood_NoRidge+Lot.Area+Bsmt.Expo
sure+Bsmt.Qual+
                Kitchen.Qual+Exter.Qual, data = houses59_num)

summary(mstep_after)

#VIF
require(car)
#Everything below 10
round(vif(mstep_after),2)

COMMANDS 3
#Descriptive Analysis

#Histograms
par(mfrow=c(3,6))
hist(houses59_num$SalePrice, main = "Histogram of SalePrice", xlab = "SalePrice",col="seagreen")
hist(houses59_num$Overall.Qual, main = "Histogram of Overall.Qual", xlab =
"Overall.Qual",col="seagreen")
hist(houses59_num$Gr.Liv.Area, main = "Histogram of Gr.Liv.Area", xlab =
"Gr.Liv.Area",col="seagreen")
hist(houses59_num$Total.Bsmt.SF, main = "Histogram of Total.Bsmt.SF", xlab =
"Total.Bsmt.SF",col="seagreen")
hist(houses59_num$Garage.Area, main = "Histogram of Garage.Area", xlab =
"Garage.Area",col="seagreen")
hist(houses59_num$X1st.Flr.SF, main = "Histogram of X1st.Flr.SF", xlab =
"X1st.Flr.SF",col="seagreen")
hist(houses59_num$Year.Built, main = "Histogram of Year.Built", xlab =
"Year.Built",col="seagreen")
```

```
hist(houses59_num$Mas.Vnr.Area, main = "Histogram of Mas.Vnr.Area", xlab =
"Mas.Vnr.Area",col="seagreen")
hist(houses59_num$Fireplaces, main = "Histogram of Fireplaces", xlab = "Fireplaces",col="seagreen")
hist(houses59_num$BsmtFin.SF.1, main = "Histogram of BsmtFin.SF.1", xlab =
"BsmtFin.SF.1",col="seagreen")
hist(houses59_num$Neighborhood_NridgHt, main = "Histogram of Neighborhood_NridgHt", xlab =
"Neighborhood_NridgHt",col="seagreen")
hist(houses59_num$Lot.Frontage, main = "Histogram of Lot.Frontage", xlab =
"Lot.Frontage",col="seagreen")
hist(houses59_num$Neighborhood_NoRidge, main = "Histogram of Neighborhood_NoRidge", xlab =
"Neighborhood_NoRidge",col="seagreen")
hist(houses59_num$Lot.Area, main = "Histogram of Lot.Area", xlab = "Lot.Area",col="seagreen")
hist(houses59_num$Bsmt.Exposure, main = "Histogram of Bsmt.Exposure", xlab =
"Bsmt.Exposure",col="seagreen")
hist(houses59_num$Bsmt.Qual, main = "Histogram of Bsmt.Qual", xlab =
"Bsmt.Qual",col="seagreen")
hist(houses59_num$Kitchen.Qual, main = "Histogram of Kitchen.Qual", xlab =
"Kitchen.Qual",col="seagreen")
hist(houses59_num$Exter.Qual, main = "Histogram of Exter.Qual", xlab =
"Exter.Qual",col="seagreen")

df <- houses59_num
%>%select(SalePrice,Overall.Qual,Gr.Liv.Area,Total.Bsmt.SF,Garage.Area,X1st.Flr.SF,Exter.Qual,Y
ear.Built,Mas.Vnr.Area,Fireplaces,

BsmtFin.SF.1,Lot.Frontage,Neighborhood_NridgHt,Neighborhood_NoRidge,Lot.Area,Bsmt.Exposure
,Bsmt.Qual,Kitchen.Qual)

#P-Values of Shapiro Test for Normality
for (i in 1:18) {
  print(c(names(df)[i],shapiro.test(df[,i])$p.value))}

#P-Values of Kolmogorov-Smirnov test for Normality
library(nortest)
ks.test(houses59_num$Overall.Qual, 'pnorm')
ks.test(houses59_num$Gr.Liv.Area, 'pnorm')
ks.test(houses59_num$Total.Bsmt.SF, 'pnorm')
ks.test(houses59_num$Garage.Area, 'pnorm')
ks.test(houses59_num$Garage.Area, 'pnorm')
ks.test(houses59_num$Year.Built, 'pnorm')
ks.test(houses59_num$Mas.Vnr.Area, 'pnorm')
ks.test(houses59_num$Fireplaces, 'pnorm')
ks.test(houses59_num$BsmtFin.SF.1, 'pnorm')
ks.test(houses59_num$Lot.Frontage, 'pnorm')
ks.test(houses59_num$Lot.Area, 'pnorm')
ks.test(houses59_num$Bsmt.Exposure, 'pnorm')
ks.test(houses59_num$Bsmt.Qual, 'pnorm')
ks.test(houses59_num$Kitchen.Qual, 'pnorm')
ks.test(houses59_num$Exter.Qual, 'pnorm')
```

```
#QQ Plots
par(mfrow=c(3,6))
qqnorm(houses59_num$SalePrice, main = "SalePrice");qqline(houses59_num$Overall.Qual,col =
'red')
qqnorm(houses59_num$Overall.Qual, main = "Overall.Qual");qqline(houses59_num$Overall.Qual,col
= 'red')
qqnorm(houses59_num$Gr.Liv.Area, main = "Gr.Liv.Area");qqline(houses59_num$Gr.Liv.Area,col =
'red')
qqnorm(houses59_num$Total.Bsmt.SF, main =
"Total.Bsmt.SF");qqline(houses59_num$Total.Bsmt.SF,col = 'red')
qqnorm(houses59_num$Garage.Area, main = "Garage.Area");qqline(houses59_num$Garage.Area,col
= 'red')
qqnorm(houses59_num$X1st.Flr.SF, main = "X1st.Flr.SF");qqline(houses59_num$X1st.Flr.SF,col =
'red')
qqnorm(houses59_num$Year.Built, main = "Year.Built");qqline(houses59_num$Year.Built,col = 'red')
qqnorm(houses59_num$Mas.Vnr.Area, main =
"Mas.Vnr.Area");qqline(houses59_num$Mas.Vnr.Area,col = 'red')
qqnorm(houses59_num$Fireplaces, main = "Fireplaces");qqline(houses59_num$Fireplaces,col = 'red')
qqnorm(houses59_num$BsmtFin.SF.1, main =
"BsmtFin.SF.1");qqline(houses59_num$BsmtFin.SF.1,col = 'red')
qqnorm(houses59_num$Neighborhood_NridgHt, main =
"Neighborhood_NridgHt");qqline(houses59_num$Neighborhood_NridgHt,col = 'red')
qqnorm(houses59_num$Neighborhood_NoRidge, main =
"Neighborhood_NoRidge");qqline(houses59_num$Neighborhood_NoRidge,col = 'red')
qqnorm(houses59_num$Lot.Frontage, main = "Lot.Frontage");qqline(houses59_num$Lot.Frontage,col
= 'red')
qqnorm(houses59_num$Lot.Area, main = "Lot.Area");qqline(houses59_num$Lot.Area,col = 'red')
qqnorm(houses59_num$Bsmt.Exposure, main =
"Bsmt.Exposure");qqline(houses59_num$Bsmt.Exposure,col = 'red')
qqnorm(houses59_num$Bsmt.Qual, main = "Bsmt.Qual");qqline(houses59_num$Bsmt.Qual,col =
'red')
qqnorm(houses59_num$Kitchen.Qual, main =
"Kitchen.Qual");qqline(houses59_num$Kitchen.Qual,col = 'red')
qqnorm(houses59_num$Exter.Qual, main = "Exter.Qual");qqline(houses59_num$Exter.Qual,col =
'red')
```

COMMANDS 4

```
#Pairwise comparisons
#Final Model
library(dplyr)
final_df <- houses59_num
%>%select(SalePrice,Overall.Qual,Gr.Liv.Area,Total.Bsmt.SF,Garage.Area,Exter.Qual,Year.Built,Ma
s.Vnr.Area,Fireplaces,
                        BsmtFin.SF.1,Lot.Frontage,Lot.Area,Bsmt.Exposure,Bsmt.Qual,Kitchen.Qual)

View(final_df)
str(final_df)

library(corrplot)
round(cor(final_df), 2)
```

```
#Diagonal is always one since every variables is perfectly correlated with itself!
corrplot(cor(final_df), method = "ellipse")

#FACTORS
require(gridExtra)
box_oq <- ggplot(data=final_df, aes(x=factor(Overall.Qual),
y=SalePrice))+geom_boxplot(col='seagreen4') + labs(x='Overall Quality')+
  scale_y_continuous(breaks=seq(0, 800000,by=100000), labels=scales::comma)
box_eq <- ggplot(data=final_df, aes(x=factor(Exter.Qual),
y=SalePrice))+geom_boxplot(col='seagreen4') + labs(x='External Quality')+
  scale_y_continuous(breaks=seq(0, 800000,by=100000), labels=scales::comma)
box_be <- ggplot(data=final_df, aes(x=factor(Bsmt.Exposure),
y=SalePrice))+geom_boxplot(col='seagreen4') + labs(x='Basement Exposure')+
  scale_y_continuous(breaks=seq(0, 800000,by=100000), labels=scales::comma)
box_bq <- ggplot(data=final_df, aes(x=factor(Bsmt.Qual),
y=SalePrice))+geom_boxplot(col='seagreen4') + labs(x='Basement Quality')+
  scale_y_continuous(breaks=seq(0, 800000,by=100000), labels=scales::comma)
box_kq <- ggplot(data=final_df, aes(x=factor(Kitchen.Qual),
y=SalePrice))+geom_boxplot(col='seagreen4') + labs(x='Kitchen Quality')+
  scale_y_continuous(breaks=seq(0, 800000,by=100000), labels=scales::comma)

grid.arrange(box_oq, box_eq, box_be, box_bq, box_kq, ncol=2)


library(ggplot2)
library(ggpubr)
#NUMERIC
scat_gra <- ggscatter(final_df, x = "Gr.Liv.Area", y = "SalePrice",add = "reg.line", conf.int = TRUE,
add.params = list(fill = "seagreen4"), ggtheme = theme_minimal())+
          stat_cor(method = "pearson", label.x = 3, label.y = 30,col = 'seagreen4')
scat_tbs <- ggscatter(final_df, x = "Total.Bsmt.SF", y = "SalePrice",add = "reg.line", conf.int = TRUE,
add.params = list(fill = "seagreen4"), ggtheme = theme_minimal())+
          stat_cor(method = "pearson", label.x = 3, label.y = 30,col = 'seagreen4')
scat_ga <- ggscatter(final_df, x = "Garage.Area", y = "SalePrice",add = "reg.line", conf.int = TRUE,
add.params = list(fill = "seagreen4"), ggtheme = theme_minimal())+
         stat_cor(method = "pearson", label.x = 3, label.y = 30,col = 'seagreen4')
scat_yb <- ggscatter(final_df, x = "Year.Built", y = "SalePrice",add = "reg.line", conf.int = TRUE,
add.params = list(fill = "seagreen4"), ggtheme = theme_minimal())+
         stat_cor(method = "pearson", label.x = 3, label.y = 30,col = 'seagreen4')
scat_mva <-ggscatter(final_df, x = "Mas.Vnr.Area", y = "SalePrice",add = "reg.line", conf.int = TRUE,
add.params = list(fill = "seagreen4"), ggtheme = theme_minimal())+
         stat_cor(method = "pearson", label.x = 3, label.y = 30,col = 'seagreen4')
scat_bsf <-ggscatter(final_df, x = "BsmtFin.SF.1", y = "SalePrice",add = "reg.line", conf.int = TRUE,
add.params = list(fill = "seagreen4"), ggtheme = theme_minimal())+
         stat_cor(method = "pearson", label.x = 3, label.y = 30,col = 'seagreen4')
scat_lf <- ggscatter(final_df, x = "Lot.Frontage", y = "SalePrice",add = "reg.line", conf.int = TRUE,
add.params = list(fill = "seagreen4"), ggtheme = theme_minimal())+
         stat_cor(method = "pearson", label.x = 3, label.y = 30,col = 'seagreen4')
scat_la <- ggscatter(final_df, x = "Lot.Area", y = "SalePrice",add = "reg.line", conf.int = TRUE,
add.params = list(fill = "seagreen4"), ggtheme = theme_minimal())+
```

```
        stat_cor(method = "pearson", label.x = 3, label.y = 30,col = 'seagreen4')
scat_f <-  ggscatter(final_df, x = "Fireplaces", y = "SalePrice",add = "reg.line", conf.int = TRUE,
add.params = list(fill = "seagreen4"), ggtheme = theme_minimal())+
        stat_cor(method = "pearson", label.x = 3, label.y = 30,col = 'seagreen4')


grid.arrange(scat_gra, scat_tbs,scat_ga, scat_yb, scat_mva, scat_bsf, scat_lf, scat_la, scat_f, ncol=3)
```

COMMANDS 5

```
#Model without Intercept
no_Intercept <- lm(SalePrice ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Exter.Qual+Year.Built+Mas.Vnr.Area+Firep
laces+

BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Bsmt.Qual+Kitchen.Qual+Exter.Qual -
1,data=final_df)

n <- nrow(houses59_num)
summary(no_Intercept)#0.9771
true.r2 <- 1-sum(no_Intercept$res^2)/((n-1)*var(houses59_num$SalePrice))
true.r2 #0.8628203

#Checking Assumptions
#Final model
first_model <- lm(SalePrice ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Exter.Qual+Year.Built+Mas.Vnr.Area+Firep
laces+
                BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Bsmt.Qual+Kitchen.Qual, data =
final_df)

summary(first_model)

par(mfrow=c(2,2))
plot(first_model)

#Step 1: Normality of the residuals
#QQ Plot - Checking for the normality of errors
plot(first_model, which = 2)
#Find residuals
Model.residuals <- rstudent(first_model)
Model.residuals
#Shapiro Wilk for residuals sample > 50
shapiro.test(Model.residuals)
lillie.test(Model.residuals)

#Step 2: Check Constant variance outliers
yhat <- fitted(first_model)
par(mfrow=c(1,2))
plot(yhat, Model.residuals)
```

```
abline(h=c(-2,2), col=2, lty=2)
plot(yhat, Model.residuals^2)
abline(h=4, col=2, lty=2)
library(car)
ncvTest(first_model)

#Step 3: Check for non linearity
library(car)
residualPlot(first_model, type='rstudent')
residualPlots(first_model, plot=F, type = "rstudent")

#Step 4: Check Independence of errors accept
plot(rstudent(first_model), type='l')
library(car);
durbinWatsonTest(first_model)

#Fix assumptions Log Price
library(boot)

second_model <- lm(log(SalePrice) ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Exter.Qual+Year.Built+Mas.Vnr.Area+Firep
laces+
                BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Bsmt.Qual+Kitchen.Qual, data =
final_df)


second_model_n <- lm(log(SalePrice) ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Year.Built+Fireplaces+
                  BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Kitchen.Qual, data = final_df)
#We should exclude Exter.Qual, Mas.Vnr.Area and Bsmt.Qual

summary(second_model)
summary(second_model_n)

par(mfrow=c(2,2))
plot(second_model_n)

#Step 1: Normality of the residuals
#QQ Plot - Checking for the normality of errors
plot(second_model, which = 2)
#Find residuals
Model.residuals <- rstudent(second_model_n)
Model.residuals
#Shapiro Wilkcoxon for residuals sample > 50
shapiro.test(Model.residuals)
lillie.test(Model.residuals)

#Step 2: Check Constant variance outliers
yhat <- fitted(second_model_n)
par(mfrow=c(1,2))
```

```
plot(yhat, Model.residuals)
abline(h=c(-2,2), col=2, lty=2)
plot(yhat, Model.residuals^2)
abline(h=4, col=2, lty=2)
ncvTest(second_model_n)

#Step 3: Check for non linearity
library(car)
residualPlot(second_model_n, type='rstudent')
residualPlots(second_model_n, plot=F, type = "rstudent")
#Step 4: Check Independence of errors accept
plot(rstudent(second_model_n), type='l')
library(car);
durbinWatsonTest(second_model_n)

#Fix assumptions Polynomials
#We should exclude Exter.Qual, Mas.Vnr.Area and Bsmt.Qual
third_model <- lm(SalePrice ~
I(Overall.Qual^5)+I(Gr.Liv.Area^8)+Total.Bsmt.SF+Garage.Area+Exter.Qual+Year.Built+Mas.Vnr.A
rea+Fireplaces+
                BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Bsmt.Qual+Kitchen.Qual, data =
final_df)

#We should exclude I(Gr.Liv.Area^8), Exter.Qual,Bsmt.Exposure
third_model_n <- lm(SalePrice ~
I(Overall.Qual^5)+Total.Bsmt.SF+Garage.Area+Year.Built+Mas.Vnr.Area+Fireplaces+
                 BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Qual+Kitchen.Qual, data = final_df)

summary(third_model)
summary(third_model_n)

par(mfrow=c(2,2))
plot(third_model_n)

#Step 1: Normality of the residuals
#QQ Plot - Checking for the normality of errors
plot(third_model_n, which = 2)
#Find residuals
Model.residuals <- rstudent(third_model_n)
Model.residuals
#Shapiro Wilkcoxon for residuals sample > 50
shapiro.test(Model.residuals)
lillie.test(Model.residuals)

#Step 2: Check Constant variance outliers
yhat <- fitted(third_model_n)
par(mfrow=c(1,2))
plot(yhat, Model.residuals)
abline(h=c(-2,2), col=2, lty=2)
plot(yhat, Model.residuals^2)
```

```
abline(h=4, col=2, lty=2)
ncvTest(third_model_n)

#Step 3: Check for non linearity
library(car)
residualPlot(third_model_n, type='rstudent')
residualPlots(third_model_n, plot=F, type = "rstudent")

#Step 4: Check Independence of errors accept
plot(rstudent(third_model_n), type='l')
library(car);
durbinWatsonTest(third_model_n)
```

COMMANDS 6

```
#use k - fold cross validation to evaluate model Choose min RMSE
#First Model
library(caret)
set.seed(1)
train_control_CV_1 <- trainControl(method = "CV", number = 10)
model_1 <- train(SalePrice ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Exter.Qual+Year.Built+Mas.Vnr.Area+Firep
laces+
                 BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Bsmt.Qual+Kitchen.Qual,
data=final_df, trControl=train_control_CV_1,
               method="lm")

print(model_1)
# Leave one out cross validation
train_control_LOOCV <- trainControl(method = "LOOCV")
model_LOOCV_1 <- train(SalePrice ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Exter.Qual+Year.Built+Mas.Vnr.Area+Firep
laces+
                       BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Bsmt.Qual+Kitchen.Qual,
data=final_df, trControl=train_control_LOOCV, method="lm")
print(model_LOOCV_1)


#Second Model
#We should exclude Exter.Qual, Mas.Vnr.Area and Bsmt.Qual
set.seed(1)
train_control_CV_2 <- trainControl(method = "CV", number = 10)
model_2 <- train(log(SalePrice) ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Year.Built+Fireplaces+
                 BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Kitchen.Qual, data=final_df,
trControl=train_control_CV_2,
               method="lm")

print(model_2)

# Leave one out cross validation
```

```
train_control_LOOCV_2 <- trainControl(method = "LOOCV")
model_LOOCV_2 <- train(log(SalePrice) ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Year.Built+Fireplaces+
                BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Kitchen.Qual, data=final_df,
trControl=train_control_LOOCV_2, method="lm")
print(model_LOOCV_2)


#Third Model
#We should exclude I(Gr.Liv.Area^8), Exter.Qual,Bsmt.Exposure
set.seed(1)
train_control_CV_3 <- trainControl(method = "CV", number = 10)
model_3 <- train(SalePrice ~
I(Overall.Qual^5)+Total.Bsmt.SF+Garage.Area+Year.Built+Mas.Vnr.Area+Fireplaces+
                 BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Qual+Kitchen.Qual, data=final_df,
trControl=train_control_CV_3,
               method="lm")

print(model_3)

# Leave one out cross validation
train_control_LOOCV_3 <- trainControl(method = "LOOCV")
model_LOOCV_3 <- train(SalePrice ~
I(Overall.Qual^5)+Total.Bsmt.SF+Garage.Area+Year.Built+Mas.Vnr.Area+Fireplaces+
                BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Qual+Kitchen.Qual, data=final_df,
trControl=train_control_LOOCV_3, method="lm")
print(model_LOOCV_3)
```

COMMANDS 7

```
test <- read.csv('ames_iowa_housing_test.csv', header = TRUE, sep = ';')

#New data frame only with features
#We should exclude Exter.Qual, Mas.Vnr.Area and Bsmt.Qual
test_df <-
test%>%select(SalePrice,Overall.Qual,Gr.Liv.Area,Total.Bsmt.SF,Garage.Area,Exter.Qual,Year.Built,
Mas.Vnr.Area,Fireplaces,
                BsmtFin.SF.1,Lot.Frontage,Lot.Area,Bsmt.Exposure,Bsmt.Qual,Kitchen.Qual)

str(test_df)

#Data Cleaning - Null replacement
#Count NAs for each variable using apply
sapply(test_df, function(x) sum(is.na(x)))

#CONTINUOUS: replace Null values with 0 and mean
test_df$Lot.Frontage <- ifelse(is.na(test_df$Lot.Frontage), round(mean(test_df$Lot.Frontage,
na.rm=TRUE)), test$Lot.Frontage)
test_df$Garage.Area[is.na(test_df$Garage.Area)] <- 0
test_df$Mas.Vnr.Area[is.na(test_df$Mas.Vnr.Area)] <- 0
```

```
#Convert Integer to numeric
intIndex_test <- sapply(test_df, class) == "integer"
test_df[,intIndex_test] <- lapply(test_df[,intIndex_test], as.numeric)

#ORDINAL: replace Null values
test_df$Bsmt.Exposure[is.na(test_df$Bsmt.Exposure)] <- "No Basement"
test_df$Bsmt.Qual[is.na(test_df$Bsmt.Qual)] <- "No Basement"

#Convert character to factor
chrIndex_test <- sapply(test_df, class) == "character"
test_df[,chrIndex_test] <- lapply(test_df[,chrIndex_test], as.factor)

#Convert Ordinals Factors to numeric
test_df$Bsmt.Exposure <- as.numeric(test_df$Bsmt.Exposure)
test_df$Kitchen.Qual<- as.numeric(test_df$Kitchen.Qual)
test_df$Bsmt.Qual <- as.numeric(test_df$Bsmt.Qual)
test_df$Exter.Qual <- as.numeric(test_df$Exter.Qual)

#First Model
library(caret)
set.seed(1)
test_CV_1 <- trainControl(method = "CV", number = 10)
test_model_1 <- train(SalePrice ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Exter.Qual+Year.Built+Mas.Vnr.Area+Firep
laces+
                      BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Bsmt.Qual+Kitchen.Qual,
data=test_df, trControl=test_CV_1,
                    method="lm")

print(test_model_1)
# Leave one out cross validation
train_control_LOOCV_1 <- trainControl(method = "LOOCV")
test_LOOCV_1 <- train(SalePrice ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Exter.Qual+Year.Built+Mas.Vnr.Area+Firep
laces+
                      BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Bsmt.Qual+Kitchen.Qual,
data=test_df, trControl=train_control_LOOCV_1, method="lm")
print(test_LOOCV_1)

#Second Model
#We should exclude Exter.Qual, Mas.Vnr.Area and Bsmt.Qual
set.seed(1)
test_CV_2 <- trainControl(method = "CV", number = 10)
test_model_2 <- train(log(SalePrice) ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Year.Built+Fireplaces+
                      BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Kitchen.Qual, data=test_df,
trControl=test_CV_2,
                    method="lm")
```

```
print(test_model_2)

#Predictions
final_test <- lm(log(SalePrice) ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Year.Built+Fireplaces+
            BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Kitchen.Qual,data=test_df)

summary(final_test)

# Leave one out cross validation
train_control_LOOCV_2 <- trainControl(method = "LOOCV")
test_LOOCV_2 <- train(log(SalePrice) ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Year.Built+Fireplaces+
              BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Kitchen.Qual, data=test_df,
trControl=train_control_LOOCV_2, method="lm")
print(test_LOOCV_2)


#Third Model
#We should exclude I(Gr.Liv.Area^8), Exter.Qual,Bsmt.Exposure
set.seed(1)
test_CV_3 <- trainControl(method = "CV", number = 10)
test_model_3 <- train(SalePrice ~
I(Overall.Qual^5)+Total.Bsmt.SF+Garage.Area+Year.Built+Mas.Vnr.Area+Fireplaces+
              BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Qual+Kitchen.Qual, data=test_df,
trControl=test_CV_3,
             method="lm")

print(test_model_3)

# Leave one out cross validation
train_control_LOOCV_3 <- trainControl(method = "LOOCV")
test_LOOCV_3 <- train(SalePrice ~
I(Overall.Qual^5)+Total.Bsmt.SF+Garage.Area+Year.Built+Mas.Vnr.Area+Fireplaces+
              BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Qual+Kitchen.Qual, data=test_df,
trControl=train_control_LOOCV_3, method="lm")
print(test_LOOCV_3)

COMMANDS 8
# model with centered covariates
test_df_new <- as.data.frame(scale(test_df, center = TRUE, scale = F))
str(test_df_new)
test_df_new$SalePrice<- test_df$SalePrice

tmodel <- lm(log(SalePrice) ~
Overall.Qual+Gr.Liv.Area+Total.Bsmt.SF+Garage.Area+Year.Built+Fireplaces+
          BsmtFin.SF.1+Lot.Frontage+Lot.Area+Bsmt.Exposure+Kitchen.Qual,data=test_df_new)

summary(tmodel)
```