Assignment_3_Belete

Nathan Belete

1/23/2018

Contents

Introduction

The purpose of this report is to analysis data collected from the Ames Assessor's office on properties sold in Ames, IA from 2006 to 2010. In this paper I will perform an explanatory data analysis on this data set and work towards building a linear regression model to predict the value of residential single family dwelling in Ames, IA.

Sample Population

The Ames housing data set has 2930 rows of data and 82 variables. For this paper, I'm interested in building a regression model to model the sale price of single family properties. For this reason, I've filtered the housing data set to reflect the following characteristics:

N	Drop Condition
01	Residential Zoning Classification
02	Single Family Dwellings
03	Normal Sale Condition

Since we are only interested in modeling the sale price of residential properties, it doesn't make sense to include properties that are zoned for agricultural, commercial, industrial, or floating village use. As a result, properties that are zoned for residential high, medium, and low densities as well as residential low density park were included in the sample data. In addition, the sample data set is limited to only single-family dwellings. Finally, I didn't want an abnormal sale conditions to influence the regression models so only properties that were sold under normal sale condition were included in the sample data set.

```
##                      Count
## 01: Not Residential    168
## 02: Not SFR            440
## 03: Not Normal Sale    379
## 99: Eligible Sample   1943
```

Given the drop conditions mentioned above, our sample data set contains 1943 rows of data and 22 variables which consist of 20 explanatory variables, 1 dependent variable, and 1 drop condition variable. A total of 987 rows of data were not included in this analysis; 168 properties were not zoned for residential use, 440 properties were not detached single-family dwellings, and the sale of 379 properties were not under normal sale conditions.

Next, I'll pick twenty variables that I found to be interesting. Here are the twenty variables plus the sale price variable:

N	Variables of Interest
01	LotArea
02	Neighborhood
03	HouseStyle
04	OverallQual
05	OverallCond
06	KitchenQual
07	YearBuilt
08	YearRemodel
09	Foundation
10	Electrical
11	KitchenAbvGr
12	FullBath
13	BedroomAbvGr
14	GrLivArea
15	RoofStyle
16	SaleType
17	MoSold
18	TotalBsmtSF
19	PoolArea
20	MiscVal
21	SalePrice

Train/Test Split

To assess the Goodnes-Of-Fit and the Predictive Accuracy of our model, our sample data set was split into two groups; training and testing data. Note that the spilt was done randomly using a random uniform distribution. The number of observations in each table is shown below.

Data	Count
Sample	1,943
Training	1,388
Testing	555

In this paper, we will use our sample data to conduct our explanatory data analysis on our response and explanatory variables. We will then use our training data set to train our models and try to find a model that fits our training data best while abiding by the assumptions of the model. In a future paper, we will use the testing data to assess the predictive accuracy of our model using out-of-sample data.

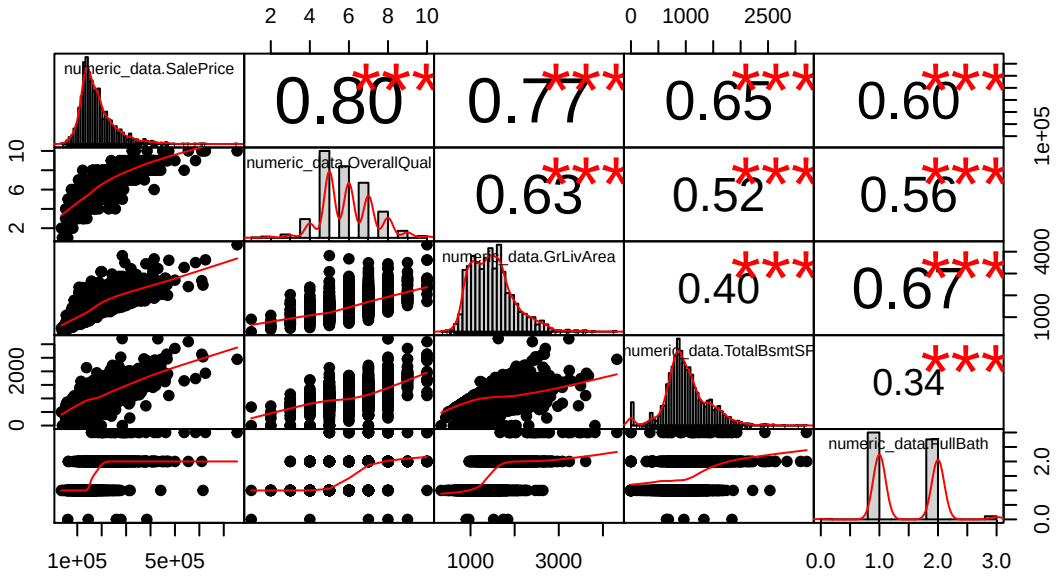
Simple Linear Regression Models

Before building our regression models, we need to first identify exploratory variables that explain our response variable best. Visually inspecting each explanatory variable against the Sale Price variable is extremely time consuming. With this in mind, my approach to identify the most important variables was to create a matrix of adjusted R squared value for each explanatory variable using a simple linear regression model. The variables listed below are the variables that had the highest adjusted R squared value.

Explanatory Variable	Adjusted R Squared Value
OverallQual	0.6403674982
GrLivArea	0.6000305534
TotalBsmtSF	0.4268763035
FullBath	0.3630033250

Next, I wanted to see how these variables were related to each other. To do this, I first wanted to see how correlated these variables were with each other. As you can see below, the response variable is highly correlated with all four explanatory variables. Since the Overall Quality variable had an adjusted R squared of 0.64 and since it was the most correlated variable against the Sale Price response variable (correlation value of 0.80), I elected to make this one of my two variables.

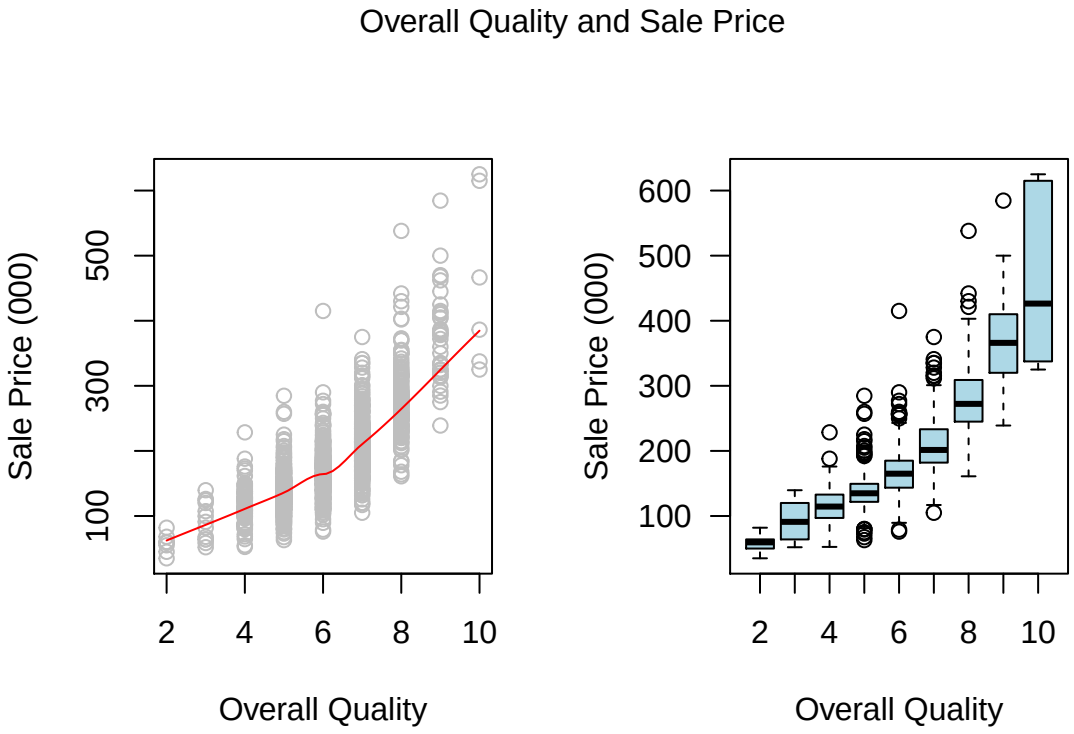
The logical next step would have been to use the General Living Area as my second variable since it had the second highest adjusted R suared value. However, since Dr. Bhatti used a similar variable in his R code example, I wanted to try something else. For this reason, I chose the runner up which was the Total Basement Square Footage variable which has a strong correlation to our response variable (0.65) and an adjusted R squared value of around 0.43.



Now that we have identified our explanatory variables, lets see how these variables perform again the response variable.

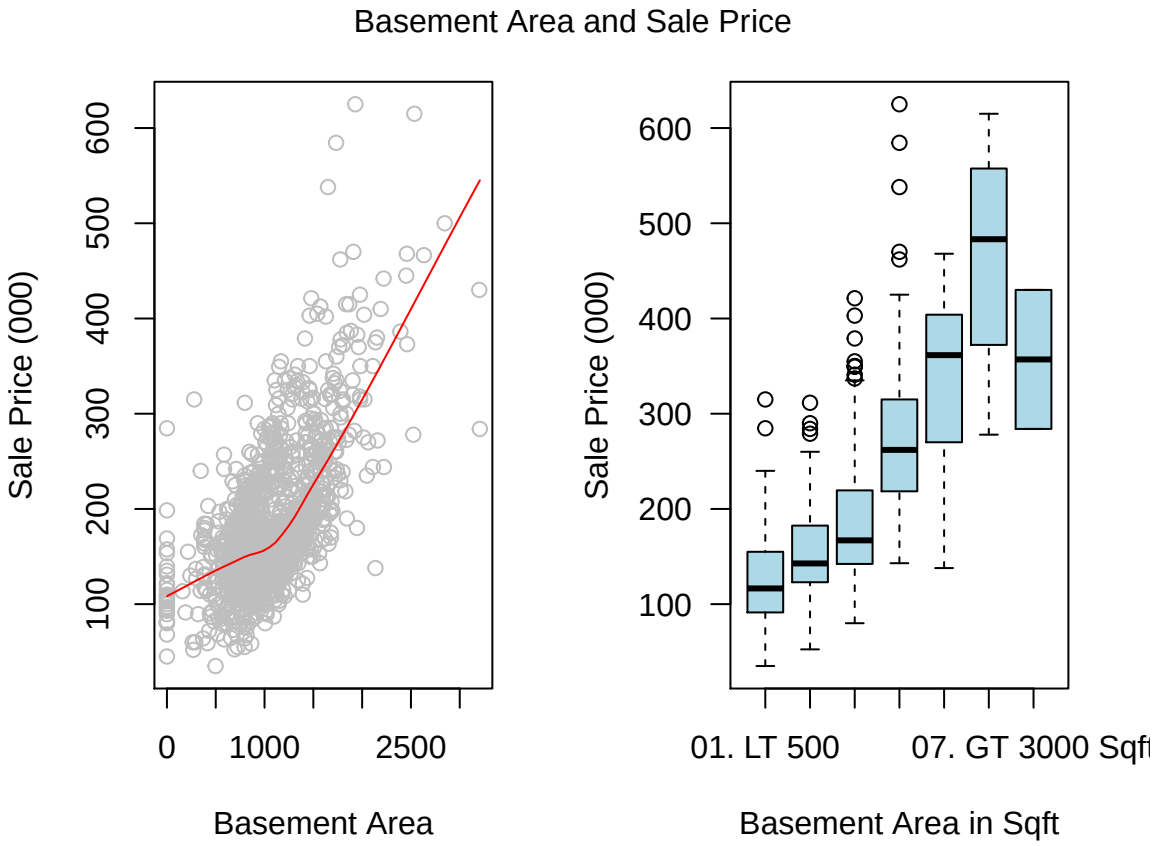
Overall Quality vs Sale Price

Below, we have a scatterplot and a boxplot of overall quality and sale price represented in thousands. The figure below seems to indicate that there is a positive relationship between these two variables. As you can see, as the overall quality of material and finish of the house increases, the sale price increases as well. Further, the boxplot shows that the median sale price of properties who have a high quality rating is larger than properties with lower quality ratings.



Total Basement Square Footage vs Sale Price

Below, we have a scatterplot and a boxplot of total basement area in sqft and sale price represented in thousands. The figure below seems to indicate that there is a positive relationship between these two variables. As you can see, as the total basement area increases, the sale price increases as well. Further, the boxplot shows that the median sale price of properties that have larger basement areas tend to be higher than properties that have less total basement space.



Linear Regression Model Assumptions

Before we start building our regression models, I want to discuss some of the Ordinary Least Squares assumptions. First, we are building a regression model so our response variable must be continuous; the explanatory variables can be either discrete or continuous. Second, the linear models are linear in the parameters and not the predictor variables. Third, the residuals of our regression model is assumed to be an independent identically distributed normal variable with mean equal to zero and variance sigma squared (fixed variance i.e. homoscedastic). When building our models, we'll want to test the residuals to make sure that our model abides by these assumptions.

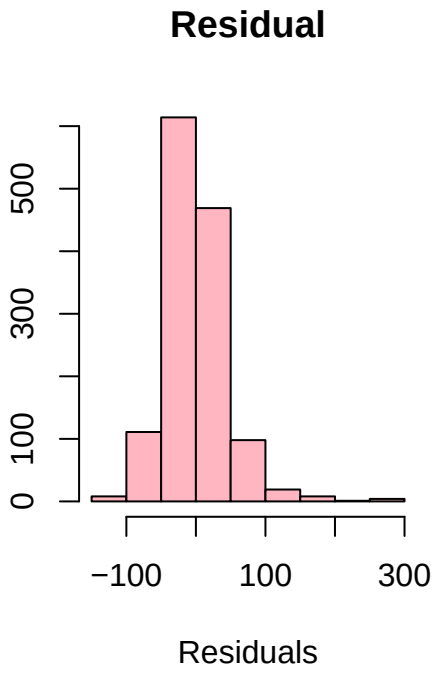
Model 1: Sale Price ~ Overall Quality

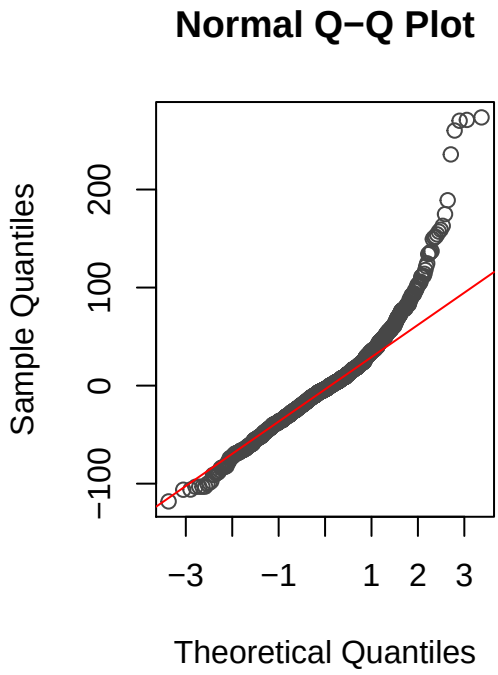
Using Overall Quality as an explanatory variable, the simple linear regression has an adjusted r-squared of around 0.65. Further, the overall quality variable as well as the y intercept are both statistically significant at the 0.001 significance level. Its worth noting that for an increase in overall quality rating, the sale price of a residential property is expected to increase by $45,109. Interestingly, the y intercept is equal to -$91,162 implying that a property that has a 0 rating would have a sale price below 0. While the Y intercept is statistically significant, I don't think the business context is valid.

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual, data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

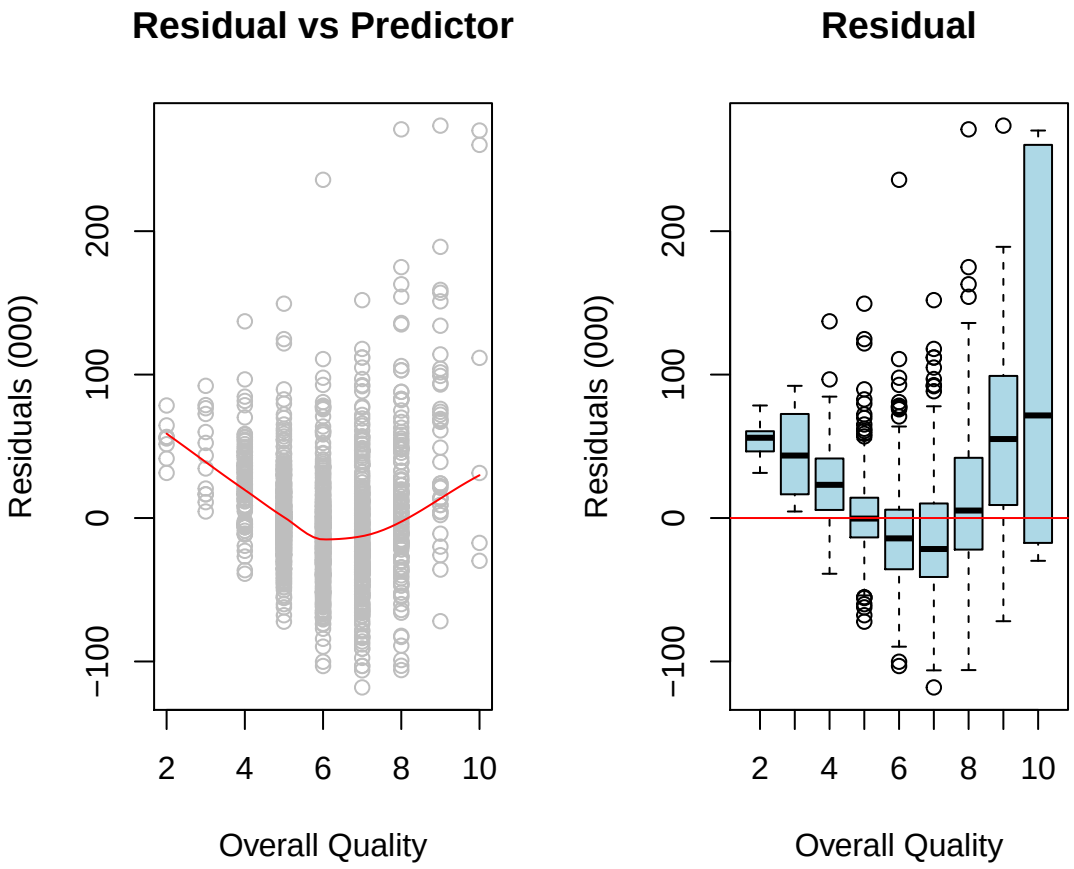
```
## -118103  -25974   -3732   18368  273569
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -84294.3     5578.3  -15.11   <2e-16 ***
## OverallQual  43913.9      908.9   48.31   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43370 on 1330 degrees of freedom
## Multiple R-squared:  0.637,  Adjusted R-squared:  0.6368
## F-statistic:  2334 on 1 and 1330 DF,  p-value: < 2.2e-16
```

A closer look at the residual plots below seems to indicate that the residual variable is not normally distributed since residuals in the second and third quantile deviate from the normal line. The histogram confirms this assumption as the the residuals are skewed right with noticeable extreme values.





Taking a look at the residual vs predictor plot, there seems to be a downward trend between properties with low to average overall quality ratings and then a positive trend for properties with higher than average overall quality ratings. This assumption is confirmed by the box plot which shows that the median error amount for properties with higher overall quality is greater that for properties with lower overall quality ratings. The pattern seen in these plots indicate that the variance of the residuals across values of the predictor variable are not fixed violating the homoscedastic assumption of our linear model.



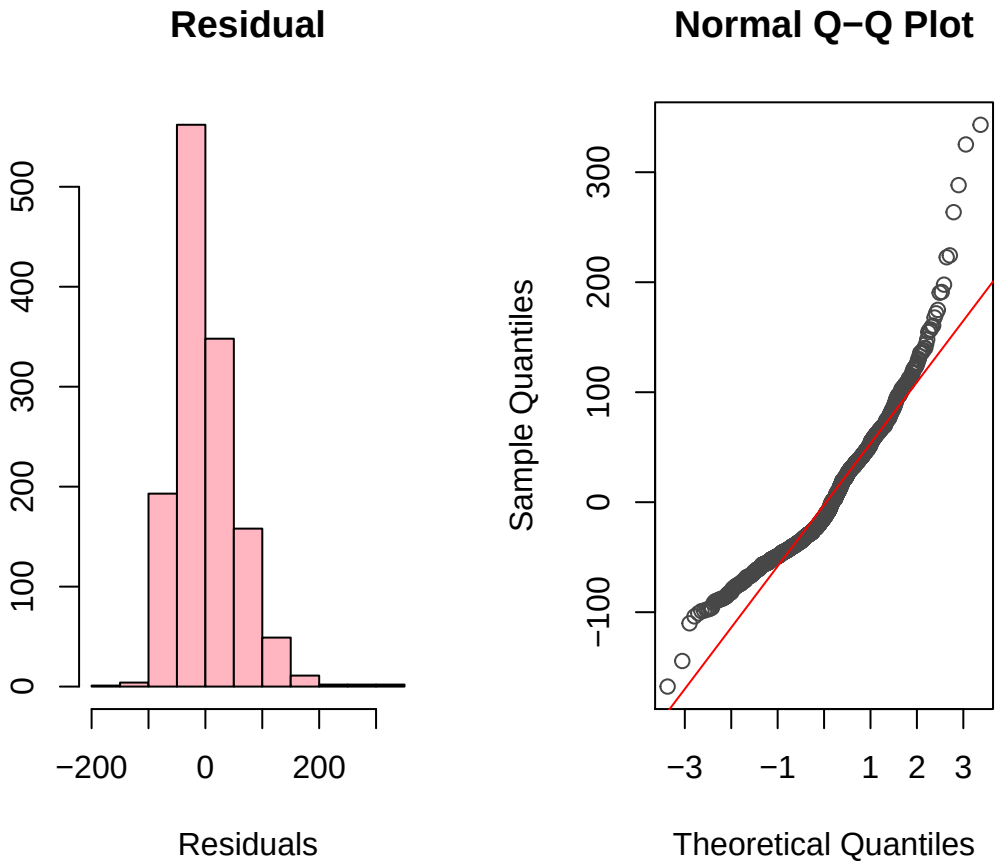
Since this model violates both the homoscedastic and normality assumptions, this model is not reliable unless these issues are addressed.

Model 2: Sale Price ~ Total Basement Sqft

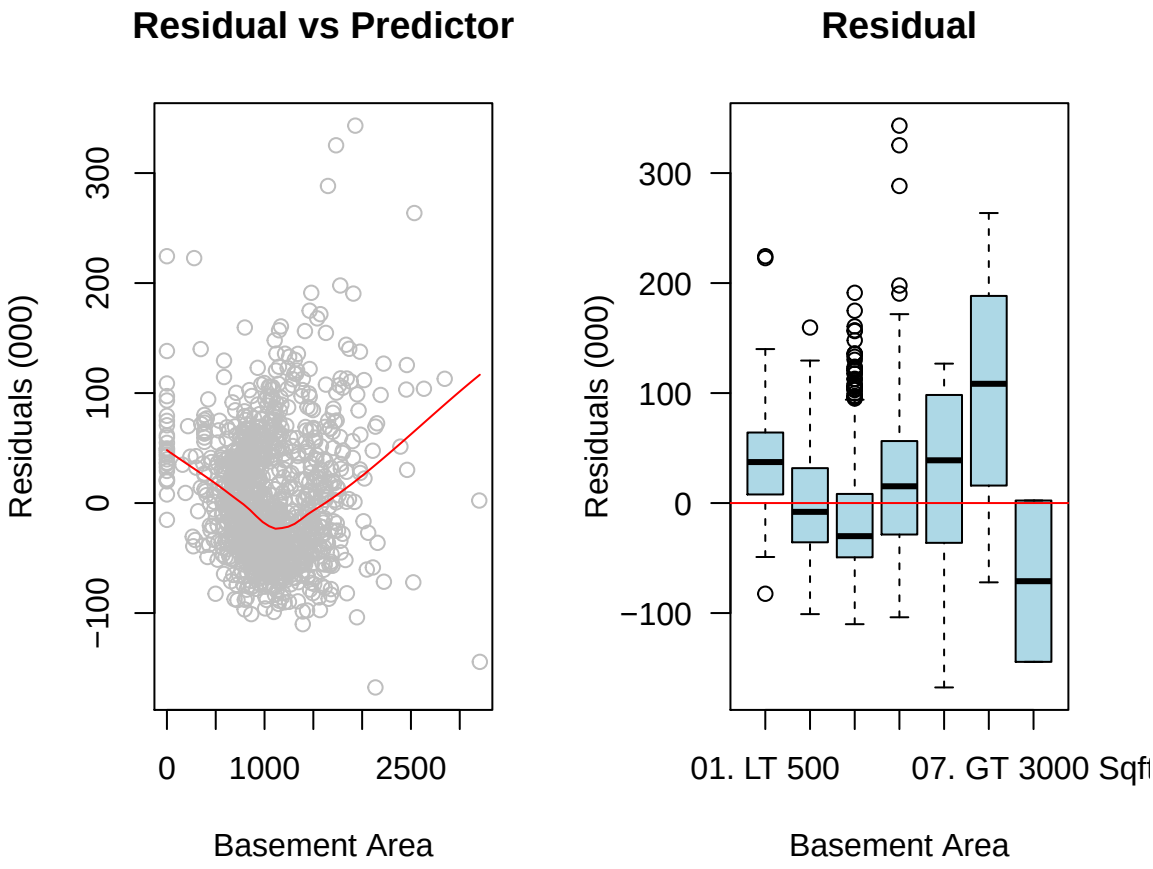
Using the Total Basement Square footage as an explanatory variable, the simple linear regression model has an adjusted r-squared of around 0.42. Further, the Total Basement Square footage variable as well as the y intercept are both statistically significant at the 0.001 significance level. Its worth noting that for a single unit increase in the total basement square footage, the sale price of a residential property is expected to increase by $120. Interestingly, the y intercept is equal to $55,018 implying that a property with no basement area has a sale price estimate of $55,018.

```
##
## Call:
## lm(formula = SalePrice ~ TotalBsmtSF, data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -167630  -40066  -13867   35202  343122
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 60284.180   4190.416   14.39   <2e-16 ***
## TotalBsmtSF   114.816      3.778   30.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55300 on 1330 degrees of freedom
## Multiple R-squared:  0.4099, Adjusted R-squared:  0.4094
## F-statistic: 923.7 on 1 and 1330 DF,  p-value: < 2.2e-16
```

A closer look at the residual plots below seems to indicate that the residual variable is not normally distributed since residuals in the second and third quantile deviate from the normal line. The histogram confirms this assumption as the the residuals are skewed right with noticeable extreme values.



Taking a look at the residual vs predictor plot, there seems to be a downward trend between Total Basement Square footage vales between 0 and 1000 and then a positive trend after. This is confirmed by the box plot which shows the median of the residual trend downward for properties with total basement area less than 1,500 sqft and the mean of residual trending upwards for properties with total basement area greater than 1,500 sqft. The pattern seen in these plots indicate that the variance of the residuals across values of the predictor variable are not fixed violating the homoscedastic assumption of our linear model.



Since this model violates both the homoscedastic and normality assumptions, this model is not reliable unless these issues are addressed.

Multiple Regression Model

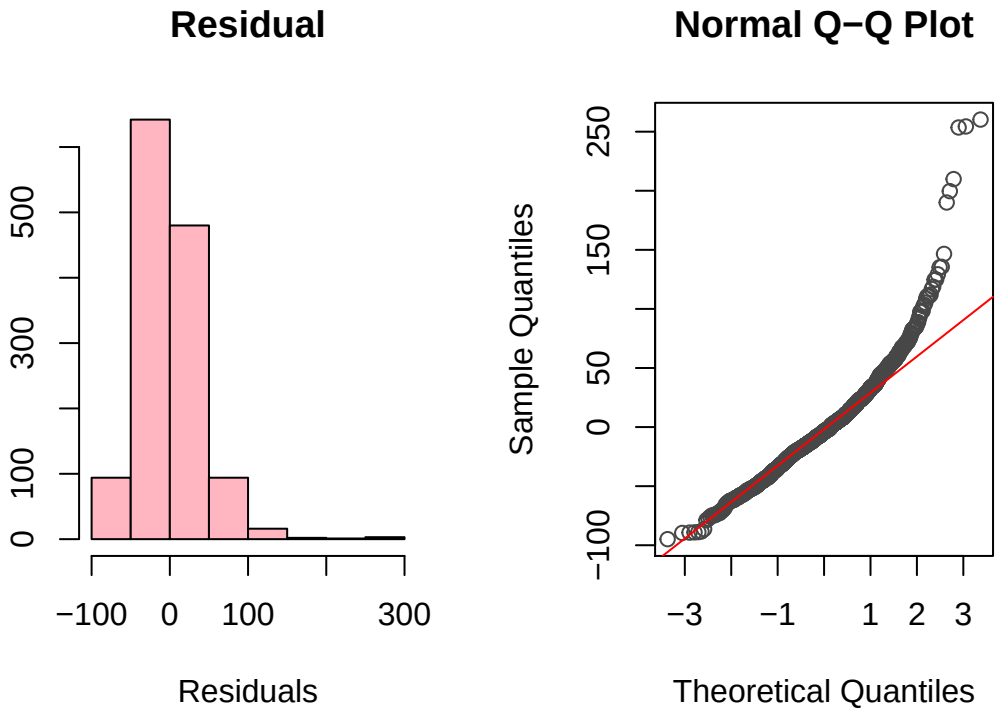
Model 3: Sale Price ~ Overall Quality + Total Basement Sqft

Using Overall Quality and Total Basement Sqft as explanatory variables, the multiple regression model has an adjusted r-squared of around 0.72. Further, Overall Quality, Total Basement Square footage, and the y intercept are both statistically significant at the 0.001 significance level.

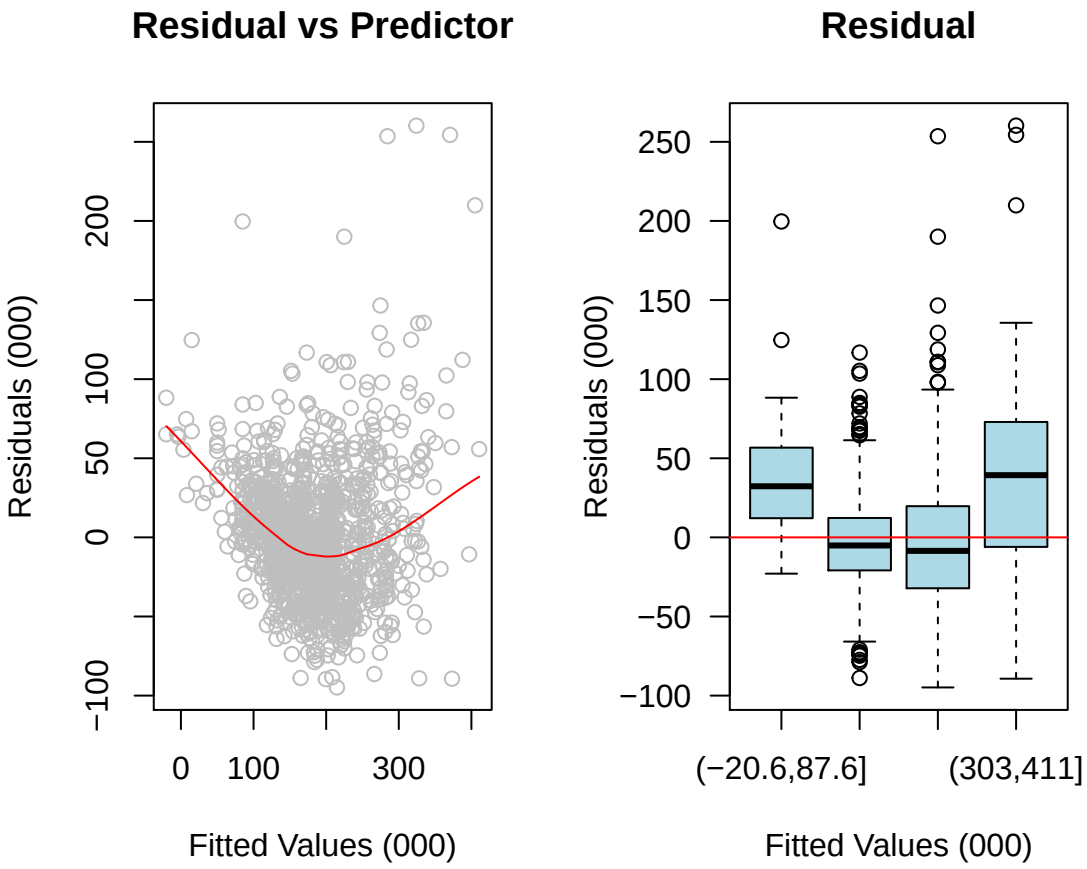
Its worth noting that for a single unit increase in the total basement square footage, the sale price of a residential property is expected to increase by $58. And for a single unit increase in overall quality rating, the sale price of a residential property is expected to increase by $35,798. Interestingly, the y intercept is equal to -$95,018 implying that a property with no basement area and an overall rating of 0 will have a sale price estimate of -$95,018. While the Y intercept is statistically significant, I don't think the business context is valid.

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual + TotalBsmtSF, data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -94843 -22655  -4132  18905 260233
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -90311.232   4976.360  -18.15   <2e-16 ***
## OverallQual  35069.010    937.420   37.41   <2e-16 ***
## TotalBsmtSF     57.101      3.056   18.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38610 on 1329 degrees of freedom
## Multiple R-squared:  0.7126, Adjusted R-squared:  0.7121
## F-statistic:  1647 on 2 and 1329 DF,  p-value: < 2.2e-16
```

A closer look at the residual plots below seems to indicate that the residual variable is not normally distributed since residuals in the second and third quantile deviate from the normal line. The histogram confirms this assumption as the the residuals are skewed right with noticeable extreme values.



Taking a look at the residual vs predictor plot, there seems to be a downward trend for fitted values between 0 and 300 and then a positive trend after. This is confirmed by the box plot which shows the median of the residual trend downward for properties with fitted values less than 320 and the mean of residual trending upwards for properties with fitted values greater than 320. The pattern seen in these plots indicate that the variance of the residuals across values of the predictor variable are not fixed violating the homoscedastic assumption of our linear model.



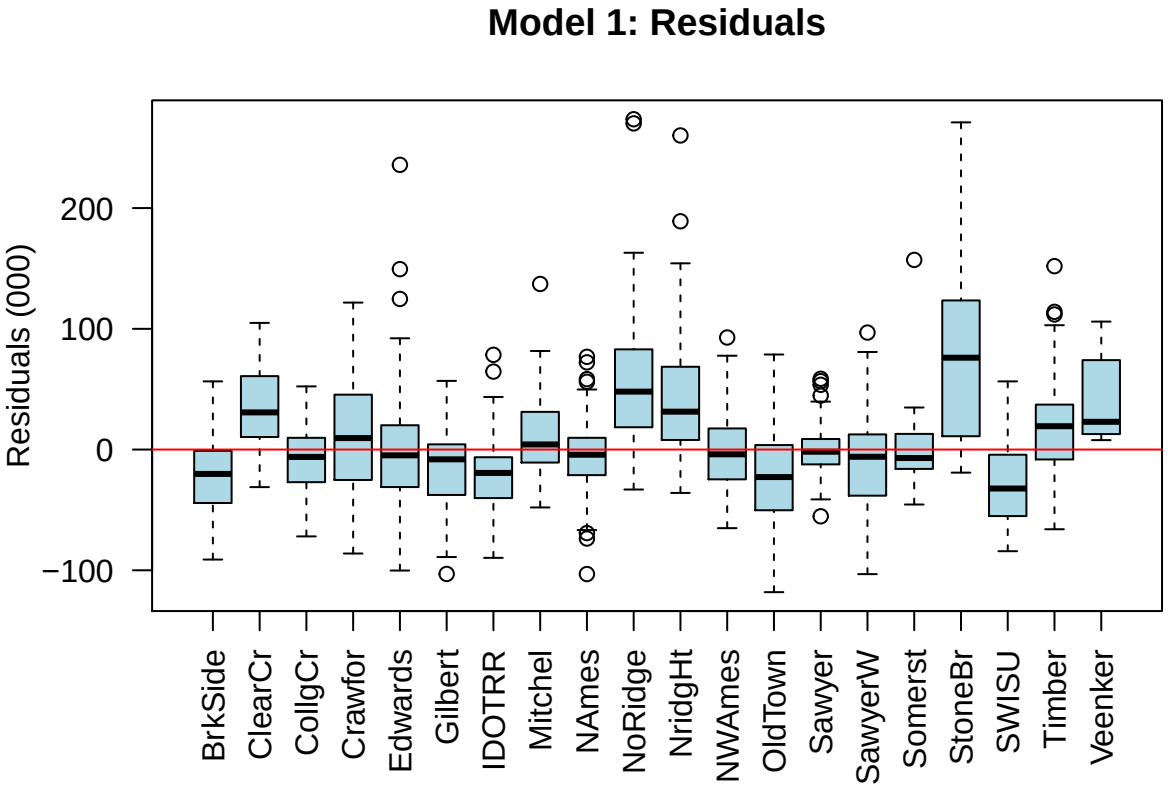
In comparison to the simple linear regression models above, this multiple linear regression does a much better job at explaining the variation in data showing an adjusted R squared of around 0.70 compared to 0.62 for model 1 and 0.39 for model 2. Since adding additional explanatory variables will always increase model the R squared value of any regression model, it should come to no surprise that the adjusted R squared value for model 3 is higher than the adjusted R squared values for the simple regression models. Finally, the pattern seen in these plots indicate that the variance of the residuals across values of the fitted variables are not fixed violating the homoscedastic assumption of our linear model.

Neighborhood Accuracy

In this section, I will show how the three models from above (two simple linear regression models and one multiple regression model) across neighborhoods. I want to see if a particular model fits the data best across neighborhoods or if certain neighborhoods are a best fit by a particular model. Further, I'll explore if there are neighborhoods that are consistently over-predicted or under-predicted.

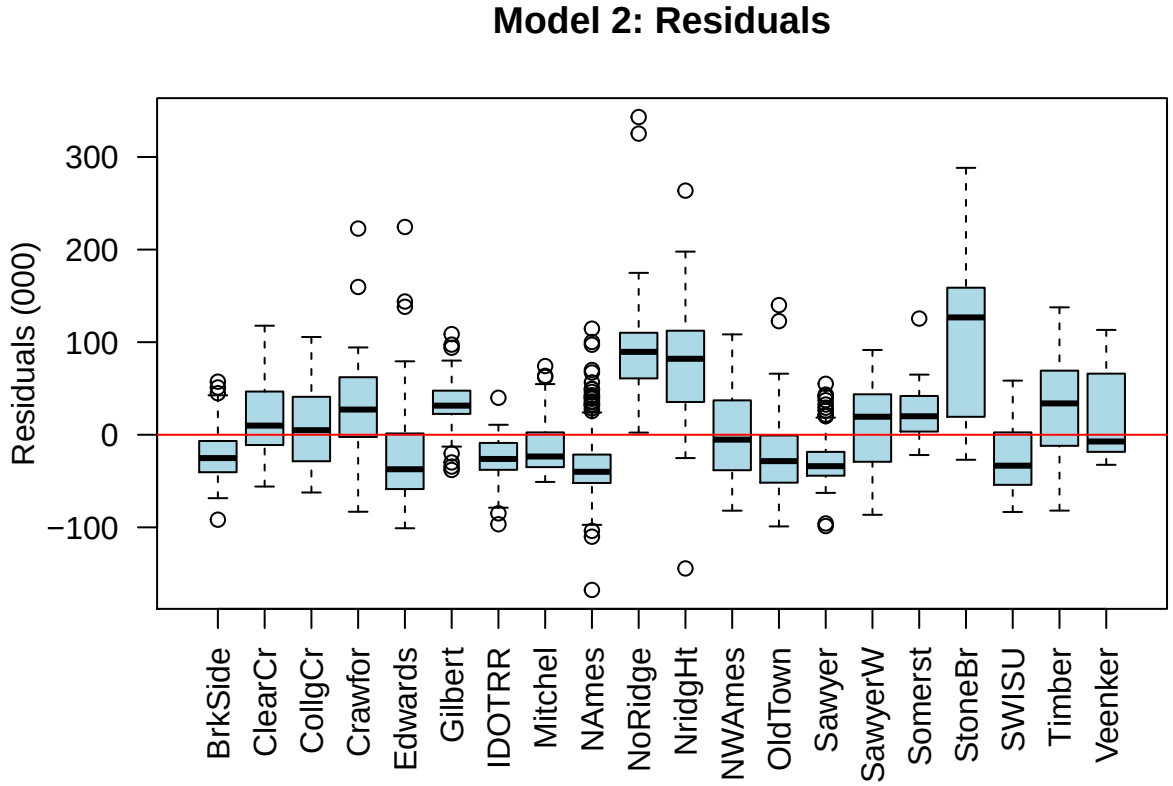
Model 1: Residuals by Neighborhood

Below we have a boxplot of model 1 distributed by neighborhood. For the most part, the residuals across neighborhoods center around zero with the exception of NoRidge, NridgHt, StondBr, SWISU, and IDOTRR. The residuals by neighborhood plot suggests that model 1 tends to over-predict in NoRidge, NridgHt, and StondBr and under-predict in SWISU, and IDOTRR. Note that Blmngtn was not identified as one of the neighborhoods that was under-predicted since there was only one property in our training data set.



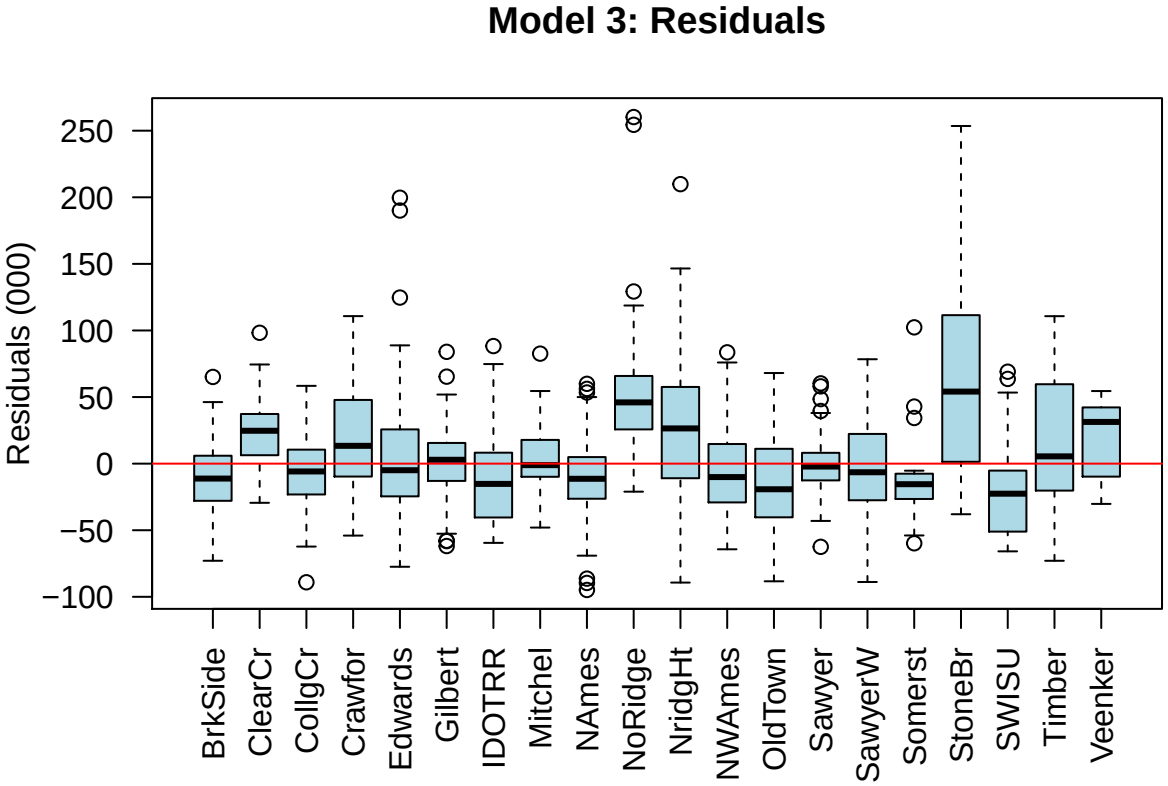
Model 2: Residuals by Neighborhood

Next, we have a boxplot of model 2 distributed by neighborhood. For the most part, the residuals across neighborhoods center around zero with the exception of NoRidge, NridgHt, StondBr, SWISU, and Edwards. The residuals by neighborhood plot suggests that model 2 tends to over-predict in NoRidge, NridgHt, and StondBr and under-predict in SWISU, and Edwards. Its worth nothing that Model 1 & 2 both over-predicted in NoRidge, NridgHt, and StondBr and under-predicted in SWISU. Note that Blmngtn was not identified as one of the neighborhoods that was under-predicted since there was only one property in our training data set.



Model 3: Residuals by Neighborhood

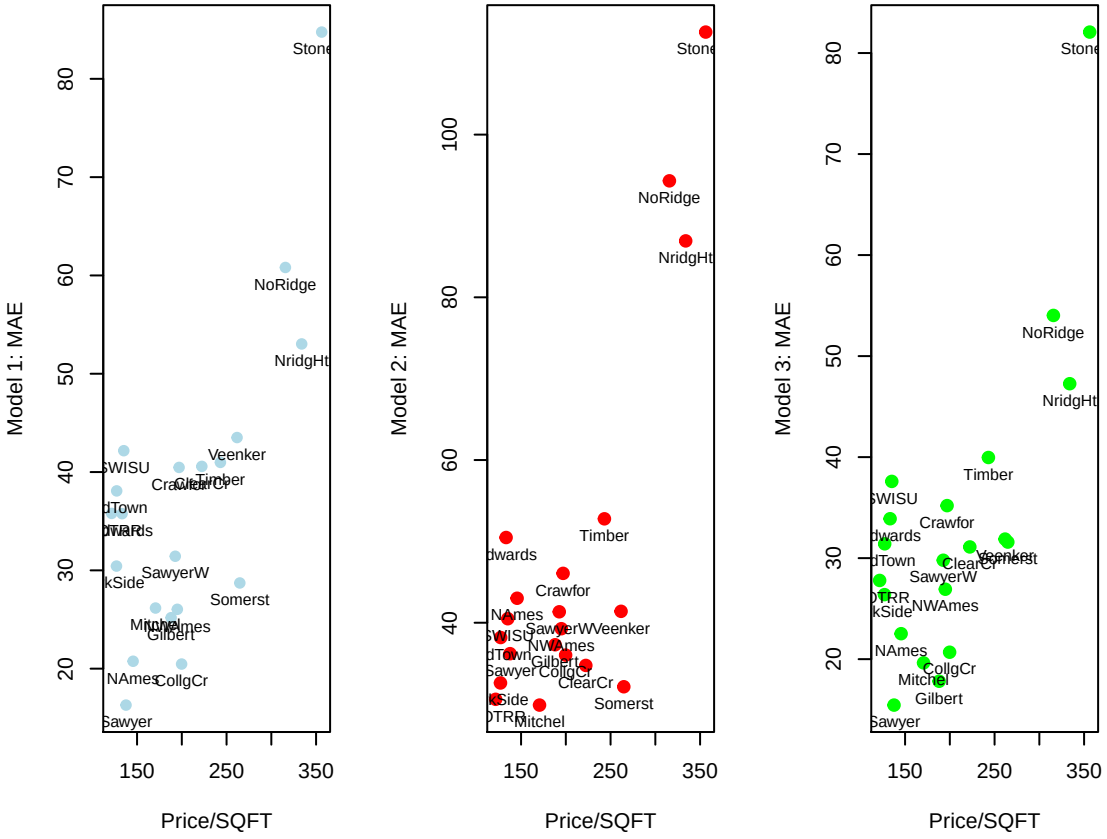
Finally, we have a boxplot of model 3 distributed by neighborhood. For the most part, the residuals across neighborhoods center around zero with the exception of NoRidge, NridgHt, StondBr, and SWISU.The residuals by neighborhood plot suggests that model 3 tends to over-predict in NoRidge, NridgHt, and StondBr and under-predict in SWISU. Its worth nothing that Model 1, 2 & 3 both over-predicted in NoRidge, NridgHt, and StondBr and under-predicted in SWISU. Note that Blmngtn was not identified as one of the neighborhoods that was under-predicted since there was only one property in our training data set.



After evaluating our residual by neighborhood plots, its clear that NoRidge, NridgHt, and StondBr are neighborhoods where our model consistently over-predicts and SWISU where it under-predicts. With this in mind, it maybe worthwhile to create dummary variables that take into account the neighborhoods that our model seems to over/under predict.

Mean Absolute Error

Below is a plot of the mean price per square foot and the mean Mean Absolute Error by neighborhood for each of our regression models. The variables in the top right quadrant of the plots are neighborhoods with high mean price per square foot and with high mean Mean Absolute Error. Properties in the bottom left quadrant are neighborhoods with low mean price per square foot and with low mean Mean Absolute Error. Its surprising to see that neighborhoods with high mean price per square foot tend to be over-predicted by the regression models. When looking at medium priced neighborhoods, our models seem to predict these areas well, with all three models having fairly close mean MAE.



Grouped Neighborhoods

To take into account the variation we are seeing across price per square foot, we are going to group neighborhoods by price per square foot. We'll create 4 groups as shown below. Note that our least expensive neighborhoods will be in group 1, our most expensive neighborhoods in group 4, and everything else in between will be in group 2 & 3.

```
##
##           Group 1 Group 2 Group 3 Group 4
##   BrkSide       1       0       0       0
##   ClearCr       0       1       0       0
##   CollgCr       0       1       0       0
##   Crawfor       0       1       0       0
##   Edwards       1       0       0       0
##   Gilbert       0       1       0       0
##   IDOTRR        1       0       0       0
##   Mitchel       1       0       0       0
##   NAmes         1       0       0       0
##   NoRidge       0       0       0       1
##   NridgHt       0       0       0       1
##   NWAmes        0       1       0       0
##   OldTown       1       0       0       0
##   Sawyer        1       0       0       0
##   SawyerW       0       1       0       0
##   Somerst       0       0       1       0
##   StoneBr       0       0       0       1
##   SWISU         1       0       0       0
##   Timber        0       0       1       0
##   Veenker       0       0       1       0
```

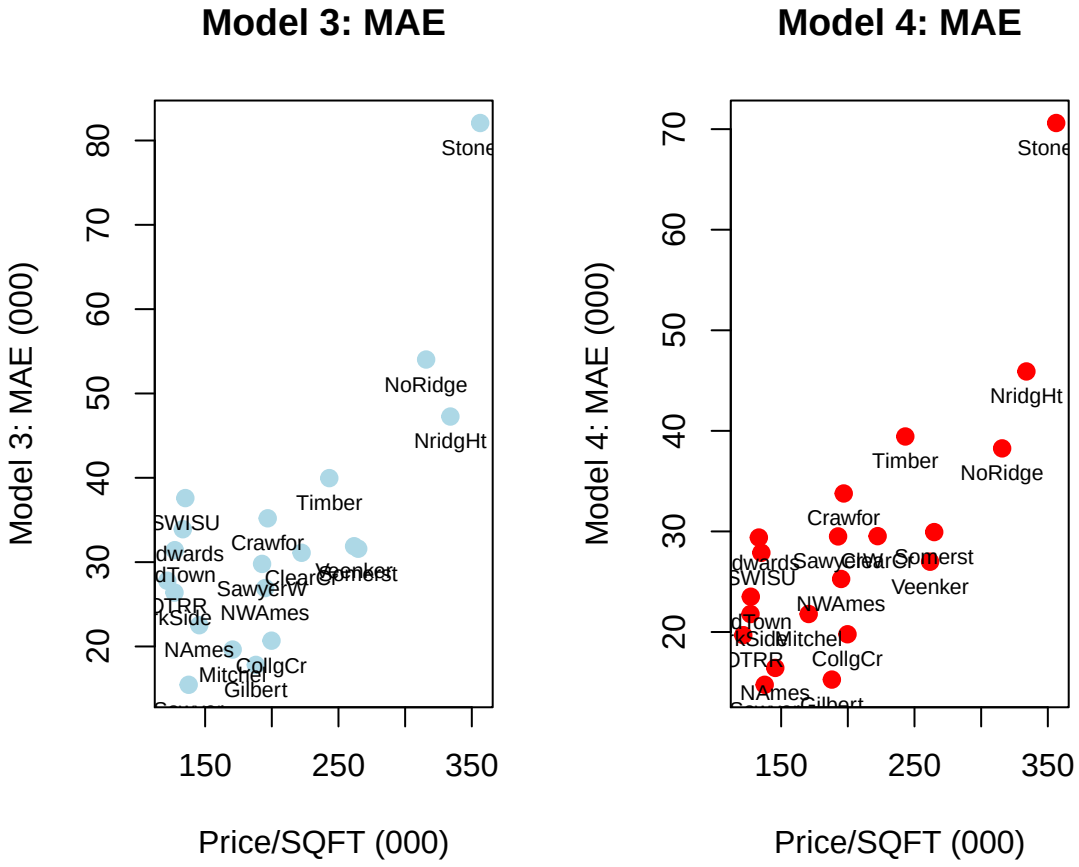
With these summary variables at hand, we want to include them in our multiple regression model from above (model 3) and see if these dummy variables have any effect on our response variable. Note that the inclusion of these dummary variables will increase our adjusted R squared; as a result, we'll compare the MAE values of both models instead.

With that said, we still want to understand the summary statistics of our new model. As you can see below, the addition of the dummary variables has increased the adjusted R squared to around 0.78. Looking at the coefficients, it looks like the group 4 indicator is not statistically significant. This is partially due to the fact that we only had 3 neighborhoods with limited data. Further; it looks like for a unit increase in overall quality and total basement sqft, we can expect an increase of $25,514 and $46 in our estimated Sale Price, respectively.

Looking at the group indictors, we see that group 1, group 2, and group 3 are -$93,712, -$74,163, and -$58,396 respectively. Since we know that the neighborhoods in group 1 have a lower average sale price than the neighborhoods in group 2 and 3 (in order), it should be no surprise that the coefficient for group 3 is greater than group 2 which is greater than group 1.

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual + TotalBsmtSF + Group.1.ID +
##     Group.2.ID + Group.3.ID + Group.4.ID, data = train.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -115824  -19569   -3136   15605  235254
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  59284.389   8820.882   6.721 2.67e-11 ***
## OverallQual  24200.912   1029.873  23.499  < 2e-16 ***
## TotalBsmtSF     45.830      2.782  16.474  < 2e-16 ***
## Group.1.ID  -89580.523   4599.298 -19.477  < 2e-16 ***
## Group.2.ID  -67709.403   4075.036 -16.616  < 2e-16 ***
## Group.3.ID  -46494.664   5614.815  -8.281 2.96e-16 ***
## Group.4.ID          NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34040 on 1326 degrees of freedom
## Multiple R-squared:  0.7771, Adjusted R-squared:  0.7763
## F-statistic: 924.6 on 5 and 1326 DF,  p-value: < 2.2e-16
```

Comparing the MAE of model 3 vs model 4, you can see that model 4 has a lower MAE than model 3. For example, the neighborhood StonBr has a MAE in model 3 of around 65,000 while in model 4 it is 60,000. Based on the MAE's of these two models, model 4 fits better than model 3.



SalePrice versus Log SalePrice as the Response

In this section we will fit two models using the explanatory variables below. The response variable will be Sale Price and log(Sale Price).

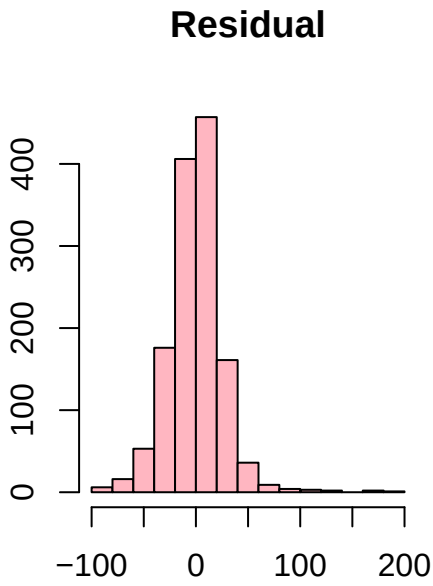
Continuous Variables	Discrete Variables
TotalBsmtSF	Group.1.ID
GrLivArea	Group.2.ID
LotArea	Group.3.ID
BedroomAbvGr	Group.4.ID
OverallQual	

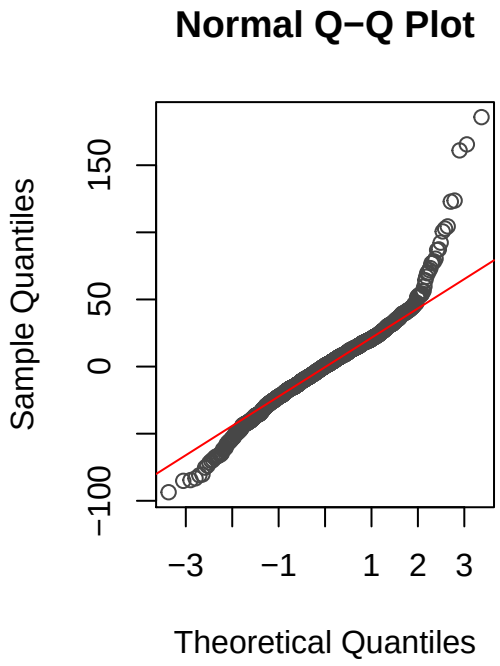
SalePrice Model

Our first model is a multiple regression model with the explanatory variables listed above. The adjusted R squared is around 0.87 and every explanatory variable is statistically significant at the 0.001 significance level with the exception of Group 4 indicator. Group 4 has only 3 neighborhoods and the amount of data in our sample is limited. Further, you can see that the cheaper neighborhood groups (Group 1) have coefficients that are less than more expensive neighborhoods (Group 3). Further, its not clear why the number of bedrooms above ground has a negative coefficient since one would assume the Sale Price of a property is a function of the number of bedrooms; this maybe due to multicollinearity between variable (GrLiving Area?). Otherwise, the coefficient of the other variables are positive as one would expect.

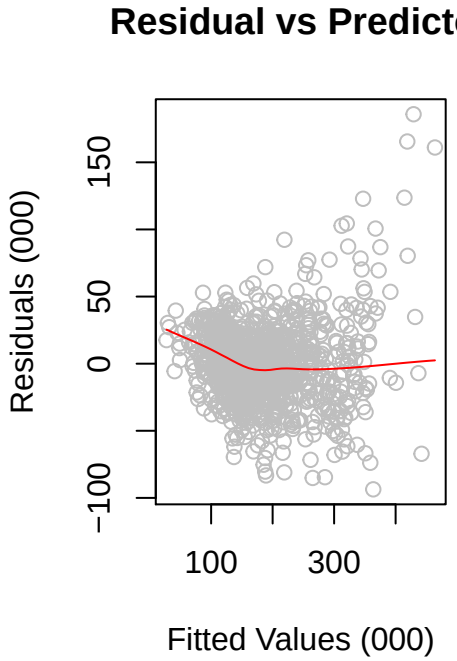
```
##
## Call:
## lm(formula = SalePrice ~ TotalBsmtSF + GrLivArea + LotArea +
##     BedroomAbvGr + OverallQual + Group.1.ID + Group.2.ID + Group.3.ID +
##     Group.4.ID, data = train.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -93630 -15182    606  14322 185836
##
## Coefficients: (1 not defined because of singularities)
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.872e+04  7.373e+03   3.895 0.000103 ***
## TotalBsmtSF   3.666e+01  2.186e+00  16.767  < 2e-16 ***
## GrLivArea     5.921e+01  2.324e+00  25.478  < 2e-16 ***
## LotArea       7.614e-01  9.043e-02   8.420  < 2e-16 ***
## BedroomAbvGr -9.695e+03  1.258e+03  -7.710 2.46e-14 ***
## OverallQual   1.613e+04  8.574e+02  18.808  < 2e-16 ***
## Group.1.ID   -6.458e+04  3.659e+03 -17.650  < 2e-16 ***
## Group.2.ID   -5.022e+04  3.218e+03 -15.608  < 2e-16 ***
## Group.3.ID   -3.315e+04  4.415e+03  -7.507 1.11e-13 ***
## Group.4.ID           NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26320 on 1323 degrees of freedom
## Multiple R-squared:  0.867,  Adjusted R-squared:  0.8662
## F-statistic:  1078 on 8 and 1323 DF,  p-value: < 2.2e-16
```

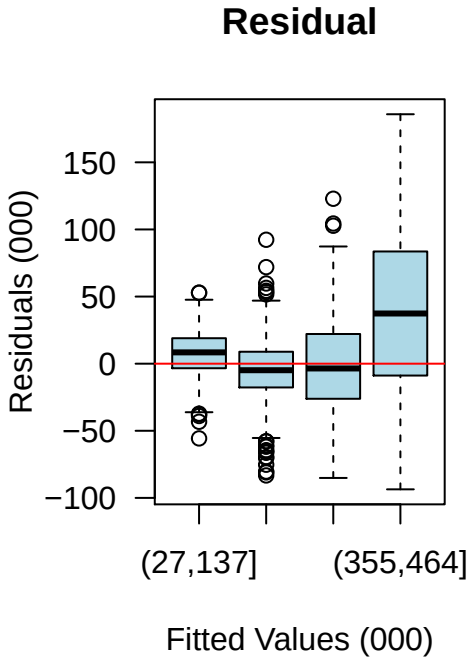
A closer look at the residual plots below seems to indicate that the residual variable is not normally distributed since residuals in the second and third quantile deviate from the normal line. The histogram confirms this assumption as the the residuals are skewed right with noticeable extreme values.





Taking a look at the residual vs predictor plot, we see that the model residuals are higher for the large fitted values and close to zero for everything else. This assumption is confirmed by the box plot which shows that the median error amount for properties with higher fitted values greater that for properties with lower fitted values. The pattern seen in these plots indicate that the variance of the residuals across values of the predictor variable are not fixed violating the homoscedastic assumption of our linear model.





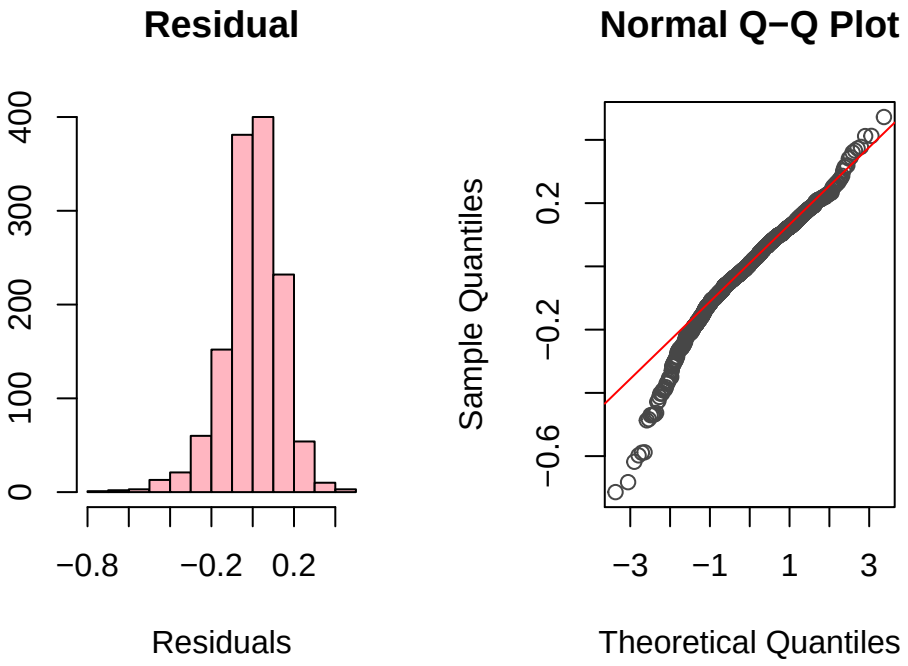
Since this model violates both the homoscedastic and normality assumptions, this model is not reliable unless these issues are addressed.

Log SalePrice Model

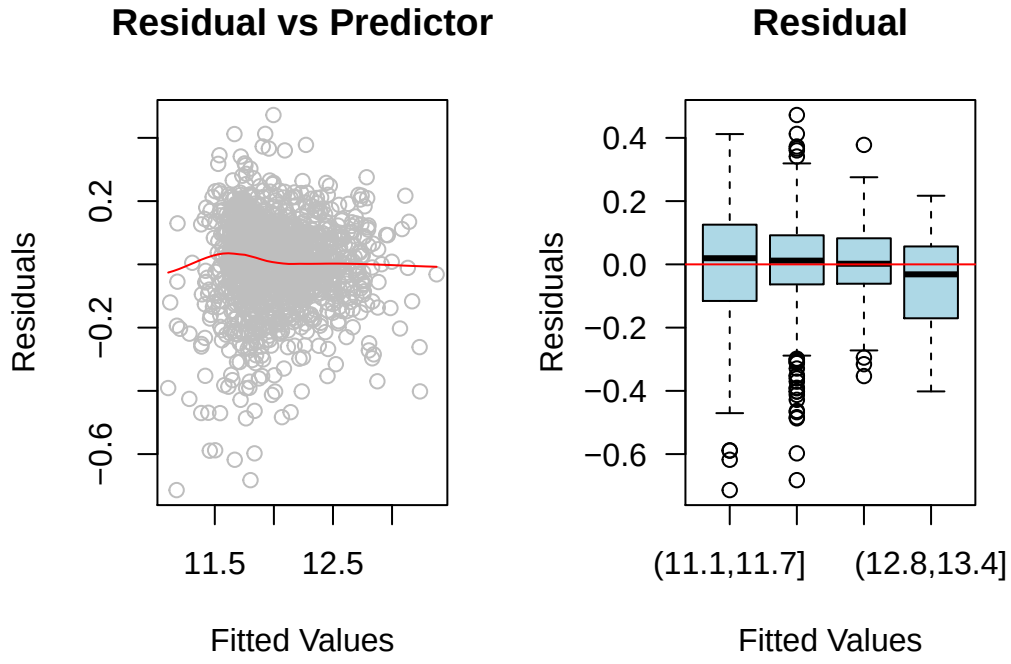
Our second model is a multiple regression model log sale price as the response variable and with the explanatory variables listed above. The adjusted R squared is around 0.86 and every explanatory variable is statistically significant at the 0.001 significance level with the exception of Group 3 & 4 indicators. Further, you can see that the cheaper neighborhood groups (Group 1) have coefficients that are less than more expensive neighborhoods (Group 2). The coefficients for the other variables are positive as one would expect. Note that we are taking log(sale price) as the response variable. As a result, the cofficients imply a percenate increase/decrease in sale price for a unit increase in our explanatory variable while holding all other explanatory variables fixed.

```
##
## Call:
## lm(formula = log(SalePrice) ~ TotalBsmtSF + GrLivArea + LotArea +
##     OverallQual + Group.1.ID + Group.2.ID + Group.3.ID + Group.4.ID,
##     data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.71382 -0.07200  0.00840  0.09268  0.47228
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.091e+01  3.731e-02 292.495  < 2e-16 ***
## TotalBsmtSF  1.898e-04  1.157e-05  16.408  < 2e-16 ***
## GrLivArea    2.638e-04  1.025e-05  25.725  < 2e-16 ***
## LotArea      3.226e-06  4.801e-07   6.720 2.70e-11 ***
## OverallQual  1.026e-01  4.533e-03  22.639  < 2e-16 ***
## Group.1.ID  -1.928e-01  1.937e-02  -9.954  < 2e-16 ***
## Group.2.ID  -7.681e-02  1.702e-02  -4.513 6.97e-06 ***
## Group.3.ID  -3.129e-02  2.346e-02  -1.334    0.183
## Group.4.ID          NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1399 on 1324 degrees of freedom
## Multiple R-squared:  0.8566, Adjusted R-squared:  0.8558
## F-statistic:  1130 on 7 and 1324 DF,  p-value: < 2.2e-16
```

A closer look at the residual histogram plot seems to indicate that the residuals follow a normal distribution with negative skew. The residuals also seem to be centered around 0 with a very small variance. The Normal Q-Q plot also indicates that the residuals are have a slight negatively skew. The plots seem to indicate that the residual variable is slightly normally distributed.



Taking a look at the residual vs predictor plot, we see that the model residuals are lower than zero for the small fitted values and close to zero for everything else. This assumption is confirmed by the box plot which shows that the median error amount for properties with low fitted values is less than that of for properties with higher fitted values. The pattern seen in these plots indicate that the variance of the residuals across values of the predictor variable are not fixed violating the homoscedastic assumption of our linear model. With that said, this model is very close to having equal varience throughout!



This model looks to be both normally distributed and seems to have almost equal variance throughout. This model is not perfect and would suggest further enhancing the model to address the minor normality and homoscedasticity issues mentioned above.

Comparison and Discussion of Model Fits

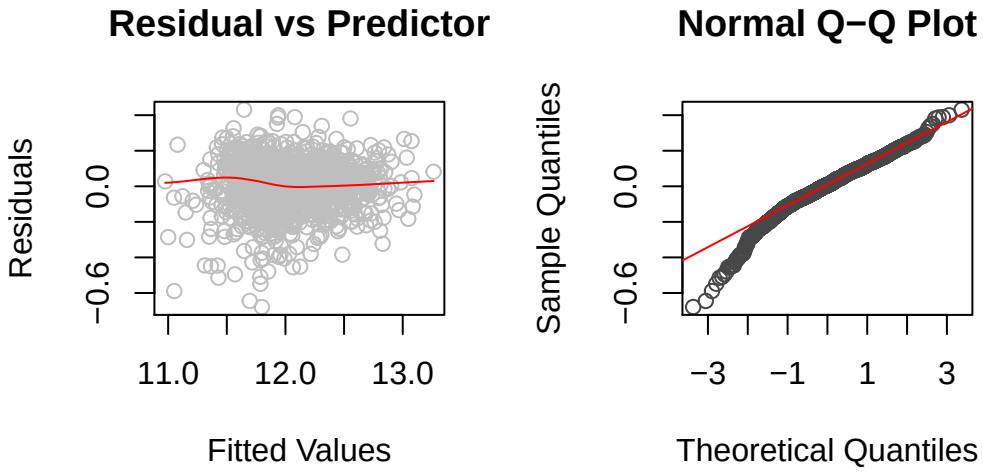
Comparing the two models above, we can see that the Sale Price model has the highest adjusted R squared value at 0.8697 compared to the Log Sale Price model with an adjusted R squared value of 0.8626. With that said, the analysis of residuals in both models showed that the Log Sale Price model seemed to be the model with residuals that were close to being normally distributed with equal variance across fitted values. Since linear regression models assume that the residual is normal and homoscedastic, and since the Log Sale Price seemed to have residuals with a close to normal distribution with equal variance throughout, the Log Sale Price model best fits our data best. This is confirmed by examining the MAE of both models below.

Model	MAE
Sale Price Model	19,325
Log Sale Price Model	17,672

As you can see above, the Log Sale Price model has the lowest MAE among the two of them. Since these models have identical explanatory variables and since the only thing different is the response variable; its clear that the transformation of the response variable has helped the residuals become 'more' normal with equal variance. Its worth noting that since we have picked a model with Log Sale Price as the response variable, the interpretation of the model is different than your typical regression model. For example, in our Log Sale Price model the coefficient for OverallQual is 0.1038. This implies that for a unit increase in OverallQual, we can expect the Sale Price variable to increase by 10.38% holding all other explanatory variables constant.

In general, the transformation of a variable will help improve the model when the response variable and explanatory variable are not linear. When we have non-linearity, taking the log of the response or explanatory variable will help improve the fit of the model. In the example below, I will take the log of LotArea and GrLivArea and show how it will improve the fit of the model.

Below is a plot of residuals after fitting a modeling using the same explanatory variables from our prior two models after taking the log of LotArea and GrLivArea as our explanatory variables. As you see below, the residuals are random with mean of around zero and the quantiles almost follow a normal distribution. This residual plot is not perfect! However, its a slight improvement from prior models. The MAE also decreased to 17,190; confirming that taking the transformation of a variable will help improve the model.



Appendix

```
---
title: "Assignment_3_Belete"
author: "Nathan Belete"
date: "1/23/2018"
output: pdf_document
toc: yes
---
\newpage

# Introduction

The purpose of this report is to analysis data collected from the Ames Assessor's office on properties s

## Sample Population

The Ames housing data set has 2930 rows of data and 82 variables. For this paper, I'm interested in buil

\setlength{\leftskip}{1cm}

N   | Drop Condition
-- | ---------
01 | Residential Zoning Classification
02 | Single Family Dwellings
03 | Normal Sale Condition

\setlength{\leftskip}{0pt}

Since we are only interested in modeling the sale price of residential properties, it doesn't make sens


suppressWarnings(library("PerformanceAnalytics")) # correlation plot



# Show the distribution  of our dropCondition
# Read in csv file for Ames housing data;
path.name <- '//Users//nathanbelete//Desktop//Predictive Analytics//Pred 410//Week 1//'
file.name <- paste(path.name,'ames_housing_data.csv',sep='')

# Read in the csv file into an R data frame;
ames.df <- read.csv(file.name,header=TRUE,stringsAsFactors=FALSE)

##############################################################################################
# Create a waterfall of drop conditions
##############################################################################################

ames.df$dropCondition <- ifelse(ames.df$Zoning %in% c('A (agr)','C (all)',
                                                      'FV','I (all)'),    '01: Not Residential',
                          ifelse(ames.df$BldgType !='1Fam',               '02: Not SFR',
                          ifelse(ames.df$SaleCondition  !='Normal',       '03: Not Normal Sale',
```

```
                                                                    '99: Eligible Sample'
                          )))

# Save the table
waterfall <- table(ames.df$dropCondition)

# Show the distribution  of our dropCondition
waterfall <- as.matrix(waterfall,4,1)
colnames(waterfall)[1] <- "Count"
waterfall

# Eliminate all observations that are not part of the eligible sample population
eligible.population <- subset(ames.df,dropCondition=='99: Eligible Sample')



Given the drop conditions mentioned above, our sample data set contains 1943 rows of data and 22 variabl




Next, I'll pick twenty variables that I found to be interesting. Here are the twenty variables plus the

\newpage
\setlength{\leftskip}{1cm}

N   | Variables of Interest
--- | ---------
01 | LotArea
02 | Neighborhood
03 | HouseStyle
04 | OverallQual
05 | OverallCond
06 | KitchenQual
07 | YearBuilt
08 | YearRemodel
09 | Foundation
10 | Electrical
11 | KitchenAbvGr
12 | FullBath
13 | BedroomAbvGr
14 | GrLivArea
15 | RoofStyle
16 | SaleType
17 | MoSold
18 | TotalBsmtSF
19 | PoolArea
20 | MiscVal
21 | SalePrice

\setlength{\leftskip}{0pt}

#############################################################################################
```

```r
# Create a list of interesting predictor variables
#################################################################################################

# Create list of important variables
skinny.df <- subset(eligible.population,
                            select=c( "LotArea","Neighborhood",
                                      "HouseStyle","OverallQual",'OverallCond',
                                      "KitchenQual","YearBuilt", "YearRemodel",
                                      "Foundation", "Electrical","KitchenAbvGr",
                                      "TotalBsmtSF","FullBath",'BedroomAbvGr',
                                      "GrLivArea", "RoofStyle", "SaleType",
                                      "MoSold","SalePrice",
                                      "PoolArea","MiscVal","dropCondition"))

# View the contents of the data frame
#str(skinny.df)

#################################################################################################
# Delete observations with missing values
#################################################################################################

# Count the number of missing values
#colSums(sapply(skinny.df , is.na))

# Delete any missing values
sample.df <- na.omit(skinny.df)

#################################################################################################
# Define some discrete variables and indicator variables
#################################################################################################

# Split numeric and character data
numeric_data <- sample.df[sapply(sample.df ,is.numeric)]
categorical_data <- sample.df[sapply(sample.df ,is.character)]

# Pool Indicator
sample.df$PoolInd <- ifelse(sample.df$PoolArea>0,1,0)

# Foundation Indicator
sample.df$BrkInd <- ifelse(sample.df$Foundation == 'BrkTil',1,0)
sample.df$CBlockInd <- ifelse(sample.df$Foundation == 'CBlock',1,0)
sample.df$PConcInd <- ifelse(sample.df$Foundation == 'PConc',1,0)

# House Style Indicator
sample.df$OneStoryInd <- ifelse(sample.df$HouseStyle == '1Story',1,0)
sample.df$TwoStoryInd <- ifelse(sample.df$HouseStyle == '2Story',1,0)

# Kitchen Quality Indicator
sample.df$KitchenTypicalInd <- ifelse(sample.df$KitchenQual == 'TA',1,0)
sample.df$KitchenGoodInd <- ifelse(sample.df$KitchenQual == 'Gd',1,0)

# Electrical Indicator
sample.df$ElectricalSbrkrInd <- ifelse(sample.df$Electrical == 'SBrkr',1,0)
```

```
sample.df$ElectricalFuseAInd <- ifelse(sample.df$Electrical == 'FuseA',1,0)

# Roof Indicator
sample.df$GableInd <- ifelse(sample.df$RoofStyle == 'Gable',1,0)
sample.df$HipInd <- ifelse(sample.df$RoofStyle == 'Hip',1,0)

# Sale Type Indicator
sample.df$SaleTypeWdInd <- ifelse(sample.df$SaleType == 'WD ',1,0)

# Year Built Ind
sample.df$BuiltPre50sInd <- ifelse(sample.df$YearBuilt < 1950,1,0)
sample.df$Built50sInd <- ifelse(sample.df$YearBuilt >= 1950 &
                                sample.df$YearBuilt < 1960,   1,0)
sample.df$Built60sInd <- ifelse(sample.df$YearBuilt >= 1960 &
                                  sample.df$YearBuilt < 1970,   1,0)
sample.df$Built70sInd <- ifelse(sample.df$YearBuilt >= 1970 &
                                  sample.df$YearBuilt < 1980,   1,0)
sample.df$Built80sInd <- ifelse(sample.df$YearBuilt >= 1980 &
                                  sample.df$YearBuilt < 1990,   1,0)
sample.df$Built90sInd <- ifelse(sample.df$YearBuilt >= 1990 &
                                  sample.df$YearBuilt < 2000,   1,0)

# Year Remodel Ind
sample.df$RemodelPre50sInd <- ifelse(sample.df$YearRemodel == 1950,1,0)
sample.df$Remodel50sInd <- ifelse(sample.df$YearRemodel > 1950 &
                                  sample.df$YearRemodel < 1960,   1,0)
sample.df$Remodel60sInd <- ifelse(sample.df$YearRemodel >= 1960 &
                                  sample.df$YearRemodel < 1970,   1,0)
sample.df$Remodel70sInd <- ifelse(sample.df$YearRemodel >= 1970 &
                                  sample.df$YearRemodel < 1980,   1,0)
sample.df$Remodel80sInd <- ifelse(sample.df$YearRemodel >= 1980 &
                                  sample.df$YearRemodel < 1990,   1,0)
sample.df$Remodel90sInd <- ifelse(sample.df$YearRemodel >= 1990 &
                                  sample.df$YearRemodel < 2000,   1,0)

# Neighborhood Indicator
sample.df$BrkSideInd <- ifelse(sample.df$Neighborhood == 'BrkSide',1,0)
sample.df$CollgCrInd <- ifelse(sample.df$Neighborhood == 'CollgCr',1,0)
sample.df$EdwardsInd <- ifelse(sample.df$Neighborhood == 'Edwards',1,0)
sample.df$GilbertInd <- ifelse(sample.df$Neighborhood == 'Gilbert',1,0)
sample.df$NAmesInd <- ifelse(sample.df$Neighborhood == 'NAmes',1,0)
sample.df$NWAmesInd <- ifelse(sample.df$Neighborhood == 'NWAmes',1,0)
sample.df$OldTownInd <- ifelse(sample.df$Neighborhood == 'OldTown',1,0)
sample.df$SawyerInd <- ifelse(sample.df$Neighborhood == 'Sawyer',1,0)



# List out sample.df
#str(sample.df)


##############################################################################################
# Add a train/test flag to split the sample
```

```
#############################################################################################
sample.df$u <- runif(n=dim(sample.df)[1],min=0,max=1)
sample.df$train <- ifelse(sample.df$u<0.70,1,0)

#############################################################################################
# Prepare data set for EDA
#############################################################################################

eda.data <- sample.df[,c( "LotArea" ,"OverallQual" , "OverallCond","TotalBsmtSF",
                          "FullBath","BedroomAbvGr","GrLivArea", "MoSold","SalePrice",
                          "PoolInd","BrkInd", "ElectricalFuseAInd",
                          "CBlockInd","PConcInd","OneStoryInd","TwoStoryInd" ,
                          "KitchenTypicalInd", "KitchenGoodInd" , "ElectricalSbrkrInd",
                          "GableInd","HipInd","SaleTypeWdInd","BuiltPre50sInd" ,
                          "Built50sInd","Built60sInd","Built70sInd","Built80sInd"  ,
                          "Built90sInd","RemodelPre50sInd","Remodel50sInd",
                          "Remodel60sInd","Remodel70sInd","Remodel80sInd","Remodel90sInd" ,
                          "Neighborhood",
                          "BrkSideInd", "CollgCrInd", "EdwardsInd",
                          "GilbertInd","NAmesInd","NWAmesInd","OldTownInd","SawyerInd",
                          "train")]
# training data set
train.df <- subset(eda.data,train==1);

# testing data set
test.df <- subset(eda.data,train==0);



## Train/Test Split

To assess the Goodnes-Of-Fit and the Predictive Accuracy of our model, our sample data set was split in

\setlength{\leftskip}{1cm}

Data   | Count
-- | ---------
Sample | 1,943
Training | 1,388
Testing | 555

\setlength{\leftskip}{0pt}

In this paper, we will use our sample data to conduct our explanatory data analysis on our response and


\newpage

# Simple Linear Regression Models

Before building our regression models, we need to first identify exploratory variables that explain our
```

```
\setlength{\leftskip}{1cm}

Explanatory Variable   | Adjusted R Squared Value
-- | ---------
OverallQual | 0.6403674982
GrLivArea | 0.6000305534
TotalBsmtSF | 0.4268763035
FullBath | 0.3630033250
\setlength{\leftskip}{0pt}



Next, I wanted to see how these variables were related to each other. To do this, I first wanted to see

The logical next step would have been to use the General Living Area as my second variable since it had

################################################################################
# corr plot
################################################################################

numeric_data <- sample.df[sapply(sample.df ,is.numeric)]

#2,10,7,8 variables with high adj.r.squared
corr.data <- data.frame(numeric_data$SalePrice,numeric_data$OverallQual,
                        numeric_data$GrLivArea,numeric_data$TotalBsmtSF,
                        numeric_data$FullBath)
#cor(corr.data)
suppressWarnings(chart.Correlation(corr.data, histogram=TRUE, pch=19))




Now that we have identified our explanatory variables, lets see how these variables perform again the r

\newpage
## Overall Quality vs Sale Price

Below, we have a scatterplot and a boxplot of overall quality and sale price represented in thousands. T



################################################################################
# Scatterplot
################################################################################

par(mfrow = c(1,2))
plot(SalePrice/1000 ~ OverallQual, data = train.df, col = "gray",
     xlab='Overall Quality',ylab='Sale Price (000)')
suppressWarnings(
  with(train.df, lines(loess.smooth(OverallQual, SalePrice/1000), col = "red"))
```

```
)

# OverallQual
boxplot((train.df$SalePrice)/1000 ~ (train.df$OverallQual), las = 1,
        xlab='Overall Quality',ylab='Sale Price (000)',

        col = "light blue")
mtext('Overall Quality and Sale Price', side = 3, line = -1, outer = TRUE)

par(mfrow=c(1,1))



## Total Basement Square Footage vs Sale Price

Below, we have a scatterplot and a boxplot of total basement area in sqft and sale price represented in




###############################################################################################
# Scatterplot
###############################################################################################
# TotalBsmtSF

par(mfrow = c(1,2))
plot(SalePrice/1000 ~ TotalBsmtSF, data = train.df, col = "gray",
     xlab='Basement Area',ylab='Sale Price (000)'
     )
with(train.df, lines(loess.smooth(TotalBsmtSF, SalePrice/1000), col = "red"))

train.df$TotalBsmtSF_cuts <-
  ifelse(train.df$TotalBsmtSF < 500,"01. LT 500",
         ifelse(train.df$TotalBsmtSF >= 500 &
                  train.df$TotalBsmtSF < 1000, "02. Btwn 500 & 1000",
                ifelse(train.df$TotalBsmtSF >= 1000 &
                  train.df$TotalBsmtSF < 1500, "03. Btwn 1000 & 1500",
                  ifelse(train.df$TotalBsmtSF >= 1500 &
                  train.df$TotalBsmtSF < 2000, "04. Btwn 1500 & 2000",
                  ifelse(train.df$TotalBsmtSF >= 2000 &
                  train.df$TotalBsmtSF < 2500, "05. Btwn 2000 & 2500",
                  ifelse(train.df$TotalBsmtSF >= 2500 &
                  train.df$TotalBsmtSF < 3000, "06. Btwn 2500 & 3000",
                  "07. GT 3000 Sqft"))))))

# TotalBsmtSF
boxplot((train.df$SalePrice)/1000 ~ (train.df$TotalBsmtSF_cuts), las = 1,
        xlab='Basement Area in Sqft',ylab='Sale Price (000)',
        col = "light blue")
mtext('Basement Area and Sale Price', side = 3, line = -3, outer = TRUE)

par(mfrow = c(1,1))
```

```
## Linear Regression Model Assumptions

Before we start building our regression models, I want to discuss some of the Ordinary Least Squares ass

## Model 1: Sale Price ~ Overall Quality

Using Overall Quality as an explanatory variable, the simple linear regression has an adjusted r-squared
```

```
# Fit a linear regression model
model.1 <- lm(SalePrice ~ OverallQual, data=train.df)

# Display model summary
model.1.summary <- summary(model.1)
#model.1.summary$adj.r.squared
model.1.summary
```

```
A closer look at the residual plots below seems to indicate that the residual variable is not normally d
```

```
par(mfrow=c(1,2))
# Make a histogram
hist(model.1$residuals/1000, col = "lightpink" ,
     xlab = "Residuals", ylab = "", main  = "Residual")
# Make a scatterplot
qqnorm(model.1$residuals/1000, col = "gray28")
qqline(model.1$residuals/1000, col = "red")
par(mfrow=c(1,1))
```

```
\newpage

Taking a look at the residual vs predictor plot, there seems to be a downward trend between properties w
```

```
# Assess the normality of the residuals
par(mfrow = c(1, 2))
# Make a scatterplot
plot(train.df$OverallQual,model.1$residuals/1000, col = "gray" ,
     xlab = "Overall Quality", ylab = "Residuals (000)")
suppressWarnings(
  lines(loess.smooth(train.df$OverallQual, model.1$residuals/1000), col = "red")
)
title('Residual vs Predictor')
# boxplot
boxplot(model.1$residuals/1000 ~ train.df$OverallQual, col = "light blue",
        xlab = "Overall Quality",  ylab = "Residuals (000)", main = "Residual")
abline(h=0, col = "red")
par(mfrow = c(1, 1))
```

```
Since this model violates both the homoscedastic and normality assumptions, this model is not reliable u

## Model 2: Sale Price ~ Total Basement Sqft

Using the Total Basement Square footage as an explanatory variable, the simple linear regression model l

\newpage

# Fit a linear regression model
model.2 <- lm(SalePrice ~ TotalBsmtSF, data=train.df)

# Display model summary
model.2.summary <- summary(model.2)
model.2.summary


A closer look at the residual plots below seems to indicate that the residual variable is not normally d


par(mfrow=c(1,2))
# Make a histogram
hist(model.2$residuals/1000, col = "lightpink" ,
     xlab = "Residuals", ylab = "", main  = "Residual")
# Make a scatterplot
qqnorm(model.2$residuals/1000, col = "gray28")
qqline(model.2$residuals/1000, col = "red")
par(mfrow=c(1,1))

\newpage

Taking a look at the residual vs predictor plot, there seems to be a downward trend between Total Baseme


# Assess the normality of the residuals
par(mfrow = c(1, 2))
# Make a scatterplot
plot(train.df$TotalBsmtSF,model.2$residuals/1000, col = "gray" ,
     xlab = "Basement Area", ylab = "Residuals (000)")
suppressWarnings(
  lines(loess.smooth(train.df$TotalBsmtSF, model.2$residuals/1000), col = "red")
)
title('Residual vs Predictor')
# boxplot
boxplot(model.2$residuals/1000 ~ train.df$TotalBsmtSF_cuts, las = 1, col = "light blue",
        xlab = "Basement Area",  ylab = "Residuals (000)", main = "Residual")
abline(h=0, col = "red")
par(mfrow = c(1, 1))


Since this model violates both the homoscedastic and normality assumptions, this model is not reliable u
```

```
# Multiple Regression Model

## Model 3: Sale Price ~ Overall Quality + Total Basement Sqft

Using Overall Quality and Total Basement Sqft as explanatory variables, the multiple regression model ha

Its worth noting that for a single unit increase in the total basement square footage, the sale price o

\newpage

################################################################################
# Model 3: Multiple Linear Regression
################################################################################

# Fit a multiple linear regression model
model.3 <- lm(SalePrice ~ OverallQual + TotalBsmtSF, data=train.df)

# Display model summary
model.3.summary <- summary(model.3)
model.3.summary

A closer look at the residual plots below seems to indicate that the residual variable is not normally d

par(mfrow=c(1,2))
# Make a histogram
hist(model.3$residuals/1000, col = "lightpink" ,
     xlab = "Residuals", ylab = "", main  = "Residual")
# Make a scatterplot
qqnorm(model.3$residuals/1000, col = "gray28")
qqline(model.3$residuals/1000, col = "red")
par(mfrow=c(1,1))

\newpage

Taking a look at the residual vs predictor plot, there seems to be a downward trend for fitted values be



# Assess the normality of the residuals
par(mfrow = c(1, 2))
# Make a scatterplot
plot(model.3$fitted.values/1000,model.3$residuals/1000, col = "gray" ,
     xlab = "Fitted Values (000)", ylab = "Residuals (000)")
suppressWarnings(
  lines(loess.smooth(model.3$fitted.values/1000, model.3$residuals/1000), col = "red")
)
title('Residual vs Predictor')
# boxplot
boxplot(model.3$residuals/1000 ~ cut(model.3$fitted.values/1000,breaks = 4), las = 1, col = "light blue
        xlab = "Fitted Values (000)",  ylab = "Residuals (000)", main = "Residual")
abline(h=0, col = "red")
par(mfrow = c(1, 1))
```

```
In comparison to the simple linear regression models above, this multiple linear regression does a much

\newpage

# Neighborhood Accuracy

In this section, I will show how the three models from above (two simple linear regression models and o

## Model 1: Residuals by Neighborhood

Below we have a boxplot of model 1 distributed by neighborhood. For the most part, the residuals across


# residuals by Neighborhood
par(mfrow = c(1, 1))
# boxplot
boxplot(model.1$residuals/1000 ~ train.df$Neighborhood, las = 2, col = "light blue",
          ylab = "Residuals (000)", main = "Model 1: Residuals")
abline(h=0, col = "red")
par(mfrow = c(1, 1))
\newpage

## Model 2: Residuals by Neighborhood

Next, we have a boxplot of model 2 distributed by neighborhood. For the most part, the residuals across


# residuals by Neighborhood
par(mfrow = c(1, 1))
# boxplot
boxplot(model.2$residuals/1000 ~ train.df$Neighborhood, las = 2, col = "light blue",
          ylab = "Residuals (000)", main = "Model 2: Residuals")
abline(h=0, col = "red")
par(mfrow = c(1, 1))
\newpage
## Model 3: Residuals by Neighborhood

Finally, we have a boxplot of model 3 distributed by neighborhood. For the most part, the residuals acr



# residuals by Neighborhood
par(mfrow = c(1, 1))
# boxplot
boxplot(model.3$residuals/1000 ~ train.df$Neighborhood, las = 2, col = "light blue",
          ylab = "Residuals (000)", main = "Model 3: Residuals")
abline(h=0, col = "red")
par(mfrow = c(1, 1))

After evaluating our residual by neighborhood plots, its clear that NoRidge, NridgHt, and StondBr are n
\newpage
```

```
## Mean Absolute Error

Below is a plot of the mean price per square foot and the mean Mean Absolute Error by neighborhood for c


# model 1
mae.1 <- aggregate(abs(model.1$residuals), by=list(Neighborhood=train.df$Neighborhood), FUN=mean)
colnames(mae.1) <- c('Neighborhood','Mean Absolute Error')

# model 2
mae.2 <- aggregate(abs(model.2$residuals), by=list(Neighborhood=train.df$Neighborhood), FUN=mean)
colnames(mae.2) <- c('Neighborhood','Mean Absolute Error')

# model 3
mae.3 <- aggregate(abs(model.3$residuals), by=list(Neighborhood=train.df$Neighborhood), FUN=mean)
colnames(mae.3) <- c('Neighborhood','Mean Absolute Error')

# price per square feet
avgSalePrice <- aggregate(train.df$SalePrice, by=list(Neighborhood=train.df$Neighborhood), FUN=mean)
colnames(avgSalePrice) <- c('Neighborhood','AvgSalePrice')

mae <- avgSalePrice
mae$m1 <- mae.1$`Mean Absolute Error`
mae$m2 <- mae.2$`Mean Absolute Error`
mae$m3 <- mae.3$`Mean Absolute Error`

par(mfrow=c(1,3))
plot(mae$AvgSalePrice/1000, mae$m1/1000, xlab = "Price/SQFT", ylab = "Model 1: MAE",
     col= "lightblue", pch = 19, cex = 1, lty = "solid", lwd = 1)
text(mae$AvgSalePrice/1000, mae$m1/1000, labels=mae$Neighborhood, cex= 0.7, pos = 1)

plot(mae$AvgSalePrice/1000, mae$m2/1000, xlab = "Price/SQFT", ylab = "Model 2: MAE",
    col= "red", pch = 19, cex = 1, lty = "solid", lwd = 2)
text(mae$AvgSalePrice/1000, mae$m2/1000, labels=mae$Neighborhood, cex= 0.7, pos = 1)

plot(mae$AvgSalePrice/1000, mae$m3/1000, xlab = "Price/SQFT", ylab = "Model 3: MAE",
      col= "green", pch = 19, cex = 1, lty = "solid", lwd = 2)
text(mae$AvgSalePrice/1000, mae$m3/1000, labels=mae$Neighborhood, cex= 0.7, pos = 1)
par(mfrow=c(1,1))

\newpage

## Grouped Neighborhoods

To take into account the variation we are seeing across price per square foot, we are going to group nei


# sort our data
avgSalePrice.ordered <- avgSalePrice[order(avgSalePrice$AvgSalePrice),]
# create 4 groups
avgSalePrice.ordered$group <- cut(avgSalePrice.ordered$AvgSalePrice, breaks = 4)
# id the groups
unique.groups <- unique(avgSalePrice.ordered$group)
```

```
# create groups ID's
avgSalePrice.ordered$GroupId <- ifelse(avgSalePrice.ordered$group == unique.groups[1],"Group 1",
                                ifelse(avgSalePrice.ordered$group == unique.groups[2],"Group 2",
                                ifelse(avgSalePrice.ordered$group ==unique.groups[3],"Group 3",
                                ifelse(avgSalePrice.ordered$group == unique.groups[4],"Group 4",
                                       ""))))
group1.df <- avgSalePrice.ordered[avgSalePrice.ordered$GroupId == "Group 1",]
group2.df <- avgSalePrice.ordered[avgSalePrice.ordered$GroupId == "Group 2",]
group3.df <- avgSalePrice.ordered[avgSalePrice.ordered$GroupId == "Group 3",]
group4.df <- avgSalePrice.ordered[avgSalePrice.ordered$GroupId == "Group 4",]
# Group Indicator
train.df$Group.1.ID <- ifelse(train.df$Neighborhood %in% group1.df$Neighborhood,1,0)
train.df$Group.2.ID <- ifelse(train.df$Neighborhood %in% group2.df$Neighborhood,1,0)
train.df$Group.3.ID <- ifelse(train.df$Neighborhood %in% group3.df$Neighborhood,1,0)
train.df$Group.4.ID <- ifelse(train.df$Neighborhood %in% group4.df$Neighborhood,1,0)

as.matrix(table(avgSalePrice.ordered$Neighborhood,avgSalePrice.ordered$GroupId))
```

With these summary variables at hand, we want to include them in our multiple regression model from abo

With that said, we still want to understand the summary statistics of our new model. As you can see bel

\newpage

Looking at the group indictors, we see that group 1, group 2, and group 3 are -$93,712, -$74,163, and -

```
################################################################################
# Model 4: Multiple Linear Regression
################################################################################

# Fit a multiple linear regression model
model.4 <- lm(SalePrice ~ OverallQual + TotalBsmtSF +
                Group.1.ID + Group.2.ID + Group.3.ID + Group.4.ID, data=train.df)

# Display model summary
model.4.summary <- summary(model.4)
model.4.summary
```

\newpage
Comparing the MAE of model 3 vs model 4, you can see that model 4 has a lower MAE than model 3. For ex

```
# model 4
mae.4 <- aggregate(abs(model.4$residuals), by=list(Neighborhood=train.df$Neighborhood), FUN=mean)
colnames(mae.4) <- c('Neighborhood','Mean Absolute Error')

mae$m4 <- mae.4$`Mean Absolute Error`
```

```
par(mfrow=c(1,2))
plot(mae$AvgSalePrice/1000, mae$m3/1000, xlab = "Price/SQFT (000)", ylab = "Model 3: MAE (000)",
      main = "Model 3: MAE", col= "lightblue", pch = 19, cex = 1, lty = "solid", lwd = 2)
text(mae$AvgSalePrice/1000, mae$m3/1000, labels=mae$Neighborhood, cex= 0.7, pos = 1)

plot(mae$AvgSalePrice/1000, mae$m4/1000, xlab = "Price/SQFT (000)", ylab = "Model 4: MAE (000)",
       main = "Model 4: MAE",col= "red", pch = 19, cex = 1, lty = "solid", lwd = 2)
text(mae$AvgSalePrice/1000, mae$m4/1000, labels=mae$Neighborhood, cex= 0.7, pos = 1)

par(mfrow=c(1,1))




# SalePrice versus Log SalePrice as the Response

In this section we will fit two models using the explanatory variables below. The response variable wil

\setlength{\leftskip}{1cm}

Continuous Variables   | Discrete Variables
-- | ---------
TotalBsmtSF | Group.1.ID
GrLivArea | Group.2.ID
LotArea | Group.3.ID
BedroomAbvGr | Group.4.ID
OverallQual |
\setlength{\leftskip}{0pt}

\newpage

## SalePrice Model

Our first model is a multiple regression model with the explanatory variables listed above. The adjuste


#############################################################################################
# Model 5: Multiple Linear Regression
#############################################################################################

# Fit a multiple linear regression model
model.5 <- lm(SalePrice ~ TotalBsmtSF + GrLivArea + LotArea + BedroomAbvGr + OverallQual +
                Group.1.ID + Group.2.ID + Group.3.ID + Group.4.ID, data=train.df)

# Display model summary
model.5.summary <- summary(model.5)
model.5.summary

\newpage
A closer look at the residual plots below seems to indicate that the residual variable is not normally


Par(mfrow=c(1,2))
```

```r
# Make a histogram
hist(model.5$residuals/1000, col = "lightpink" ,
     xlab = "Residuals", ylab = "", main  = "Residual")
# Make a scatterplot
qqnorm(model.5$residuals/1000, col = "gray28")
qqline(model.5$residuals/1000, col = "red")
par(mfrow=c(1,1))
```

Taking a look at the residual vs predictor plot, we see that the model residuals are higher for the larg

```r
# Assess the normality of the residuals
par(mfrow = c(1, 2))
# Make a scatterplot
plot(model.5$fitted.values/1000,model.5$residuals/1000, col = "gray" ,
     xlab = "Fitted Values (000)", ylab = "Residuals (000)")
suppressWarnings(
  lines(loess.smooth(model.5$fitted.values/1000, model.5$residuals/1000), col = "red")
)
title('Residual vs Predictor')
# boxplot
boxplot(model.5$residuals/1000 ~ cut(model.5$fitted.values/1000,breaks = 4), las = 1, col = "light blue
        xlab = "Fitted Values (000)",  ylab = "Residuals (000)", main = "Residual")
abline(h=0, col = "red")
par(mfrow = c(1, 1))
```

Since this model violates both the homoscedastic and normality assumptions, this model is not reliable u

\newpage

Log SalePrice Model

Our second model is a multiple regression model log sale price as the response variable and with the exp

```r
##############################################################################
# Model 6: Log SalePrice Response Model
##############################################################################

# Fit a multiple linear regression model
model.6 <- lm(log(SalePrice) ~ TotalBsmtSF + GrLivArea + LotArea  + OverallQual +
                Group.1.ID + Group.2.ID + Group.3.ID + Group.4.ID, data=train.df)

# Display model summary
model.6.summary <- summary(model.6)
model.6.summary
```

\newpage
A closer look at the residual histogram plot seems to indicate that the residuals follow a normal distri

```r
par(mfrow=c(1,2))
# Make a histogram
hist(model.6$residuals, col = "lightpink" ,
     xlab = "Residuals", ylab = "", main  = "Residual")
# Make a scatterplot
qqnorm(model.6$residuals, col = "gray28")
qqline(model.6$residuals, col = "red")
par(mfrow=c(1,1))
```

Taking a look at the residual vs predictor plot, we see that the model residuals are lower than zero fo

```r
# Assess the normality of the residuals
par(mfrow = c(1, 2))
# Make a scatterplot
plot(model.6$fitted.values,model.6$residuals, col = "gray" ,
     xlab = "Fitted Values", ylab = "Residuals")
suppressWarnings(
  lines(loess.smooth(model.6$fitted.values, model.6$residuals), col = "red")
)
title('Residual vs Predictor')
# boxplot
boxplot(model.6$residuals ~ cut(model.6$fitted.values,breaks = 4), las = 1, col = "light blue",
        xlab = "Fitted Values",  ylab = "Residuals", main = "Residual")
abline(h=0, col = "red")
```

This model looks to be both normally distributed and seems to have almost equal variance throughout. Th

Comparison and Discussion of Model Fits

```r
mae.5 <- mean(abs(model.5$residuals))
mae.6 <- mean(abs(train.df$SalePrice-exp(model.6$fitted.values)))
```

Comparing the two models above, we can see that the Sale Price model has the highest adjusted R squared

\setlength{\leftskip}{1cm}

Model	MAE
Sale Price Model	19,325

```
Log Sale Price Model | 17,672

\setlength{\leftskip}{0pt}


As you can see above, the Log Sale Price model has the lowest MAE among the two of them. Since these mod

In general, the transformation of a variable will help improve the model when the response variable and



################################################################################
# Model 6: Log SalePrice Response Model
################################################################################

# Fit a multiple linear regression model
model.7 <- lm(log(SalePrice) ~ TotalBsmtSF + log(GrLivArea) + log(LotArea)  + OverallQual +
                Group.1.ID + Group.2.ID + Group.3.ID + Group.4.ID, data=train.df)

# Display model summary
model.7.summary <- summary(model.7)

mae.7 <- mean(abs(train.df$SalePrice-exp(model.7$fitted.values)))




Below is a plot of residuals after fitting a modeling using the same explanatory variables from our pric


par(mfrow=c(1,2))
# Make a scatterplot
plot(model.7$fitted.values,model.7$residuals, col = "gray" ,
     xlab = "Fitted Values", ylab = "Residuals")
suppressWarnings(
  lines(loess.smooth(model.7$fitted.values, model.7$residuals), col = "red")
)
title('Residual vs Predictor')
# Make a scatterplot
qqnorm(model.7$residuals, col = "gray28")
qqline(model.7$residuals, col = "red")
par(mfrow=c(1,1))
```