Introduction

In this report, we are continuing our work with the Ames Housing data set. We are still investigating the “SalePrice” response variable for homes in Ames. In order to restrict our calculations to single family homes, and no other type of property, the data will be filtered and a subset will be used to build models. We are producing both Simple Linear Regression models, and Multiple Regression models. New to this report, are Multiple Regression models with interaction terms.

In order to settle on predictor variables, I did create additional models, testing other continuous variables (GarageSF and FirstFlrSF), to see if they offered an improvement over the variables I have been working with. Those models were not improvements, so I did not bother to include them in this report. I am continuing my work using GrLivArea and TotBsmtSF as my primary predictors.

To evaluate our models, we will look at QQ plots of the residuals, scatterplots of the residuals, data with fitted regression lines, as well as fitted models (summary statistical values) for the models. We will also transform the response variable and look at the impact that transformation has on the Multiple Regressions models. We will see that transformation can have a beneficial effect on the outcome of regression, and that viable transformations are not limited to taking the log of the response variable.

Sample Definition

The problem statement specifies we are building models for the sale prices of homes in Ames. In order to restrict the data to the appropriate observations, it has been filtered. First, all properties which are not zoned as residential are dropped; 168 observations were dropped as a result of this condition. Dropping non-residential properties eliminates any variation from commercial properties with attached living spaces.

Next, all properties which are not single-family homes are dropped from the data set. This reduced the data set by 440 observations. While Condos, Duplexes, and other multi-tenant options are viable housing choices, they potentially have different sale characteristics than stand-alone homes, and we do not want to mix the housing types in this model.

Finally, all observations which have a sale condition that is not “Normal” are dropped from the data set. This reduced the number of observations by an additional 379. Any non-normal sale, such as a foreclosure or sale between family members may not be representative of the true (or fair market) value of the property. There is a risk non-normal sales may be a lower price than what the house would sell for in a traditional sale, and so are excluded for the purpose of this model. The final working data set is 1943 observations of 82 variables. A visual summary of the data drops is shown in Figure 1.

FIGURE 1 - WATERFALL OF DATA DROPPED

Simple Linear Regression Models

Simple Model 1: Using Above Grade Living Area (GrLivArea) as Predictor Variable

The first model I fit was for the predictor variable GrLivArea. The Python 'ols' module from the Statsmodel package produces the following results:

TABLE 1 - FITTED RESULTS FOR SIMPLE LINEAR REGRESSION MODEL 1

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.600
Model:                            OLS   Adj. R-squared:                  0.600
Method:                 Least Squares   F-statistic:                     2914.
Date:                Fri, 30 Jun 2017   Prob (F-statistic):               0.00
Time:                        14:03:27   Log-Likelihood:                -23628.
No. Observations:                1943   AIC:                         4.726e+04
Df Residuals:                    1941   BIC:                         4.727e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   8837.5707   3312.706      2.668      0.008    2340.736    1.53e+04
GrLivArea    113.7602      2.107     53.985      0.000     109.627     117.893
==============================================================================
Omnibus:                      504.065   Durbin-Watson:                   1.218
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2911.911
Skew:                           1.091   Prob(JB):                         0.00
Kurtosis:                       8.586   Cond. No.                     4.96e+03
==============================================================================
```

I interpret the R^2 result as showing that the above grade living area variable accounts for 60% of the variability of sale price for homes in the data set. The test statistic, F, shows we would reject a null hypothesis of all the ß values being equal to 0. The value for the t-test is significant. A visualization of the data with the regression line is shown in Figure 2, followed by a plot of the residuals in Figure 3 and the QQ plot for the residuals in Figure 4.

FIGURE 2 – DATA AND FITTED REGRESSION LINE FOR LIVING AREA

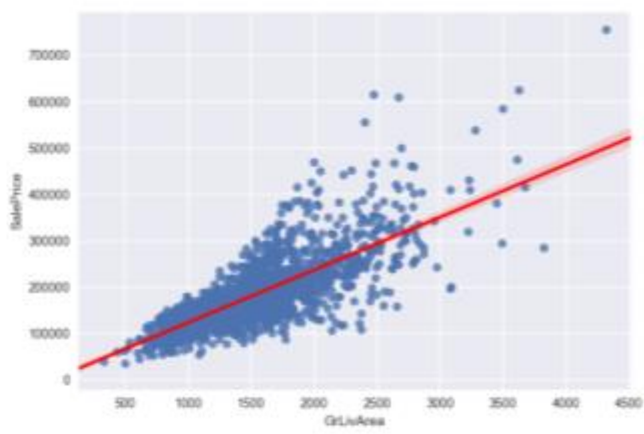
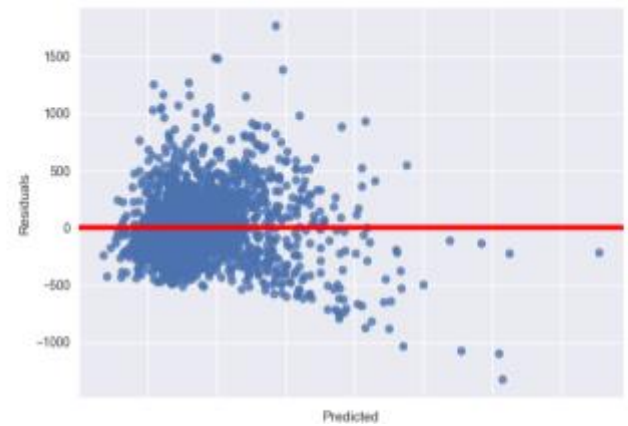
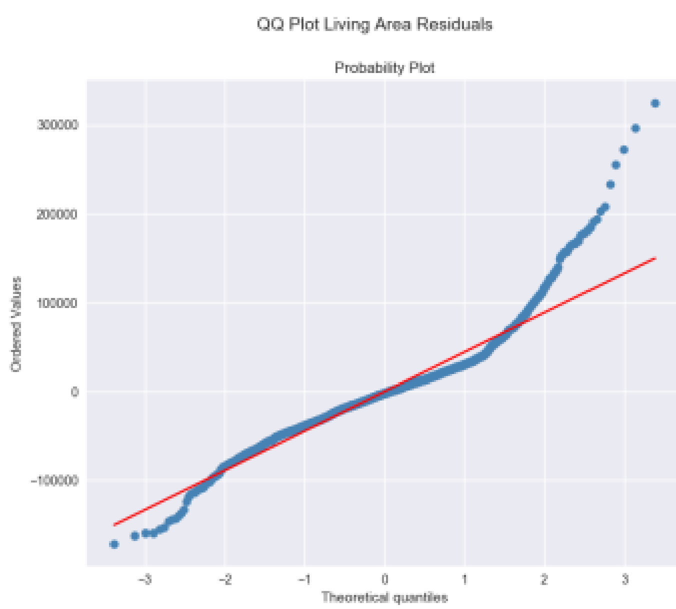


FIGURE 3 - RESIDUALS VERSUS FITTED VALUES



The plot of the Sale and Area data above indicates heteroscedasticity, which in turn indicates that the t-statistic may be unreliable. The plot of Residuals versus Predicted values fails to show a pleasing, random distribution, another indicator of heteroscedasticity.

FIGURE 4 – QQ PLOT OF MODEL RESIDUALS



The QQ pot of the residuals shows a non-normal distribution for the residuals. This is an indicator that the F- and t-statistics may not be trustworthy for this model.

Simple Model 2: Using Total Basement Area (TotBsmtArea) as Predictor Variable

The second model I fit was for the predictor variable TotalBasmtSF. The Python 'ols' module from the Statsmodel package produces the following results:

TABLE 2 - FITTED RESULTS FOR SIMPLE LINEAR REGRESSION MODEL 2

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.427
Model:                            OLS   Adj. R-squared:                  0.427
Method:                 Least Squares   F-statistic:                     1447.
Date:                Fri, 30 Jun 2017   Prob (F-statistic):          4.03e-237
Time:                        14:46:27   Log-Likelihood:                -23977.
No. Observations:                1943   AIC:                         4.796e+04
Df Residuals:                    1941   BIC:                         4.797e+04
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept    5.55e+04   3467.676     16.004      0.000    4.87e+04    6.23e+04
TotalBsmtSF  119.1081      3.131     38.045      0.000     112.968     125.248
==============================================================================
Omnibus:                      531.409   Durbin-Watson:                   1.184
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1892.889
Skew:                           1.320   Prob(JB):                         0.00
Kurtosis:                       7.052   Cond. No.                     3.06e+03
==============================================================================
```

As with the first model, the F-statistic and the t-test values show significance, but given the QQ plot of residuals, this may not be reliable. The R^2 value indicates ~43% of price variability is related to basement size. The data and regression line are shown in Figure 5, the plot of the residuals is shown in Figure 6, and Figure 7 shows the QQ plot for the residuals.

FIGURE 5 – DATA AND FITTED REGRESSION LINE FOR SIZE OF BASEMENT

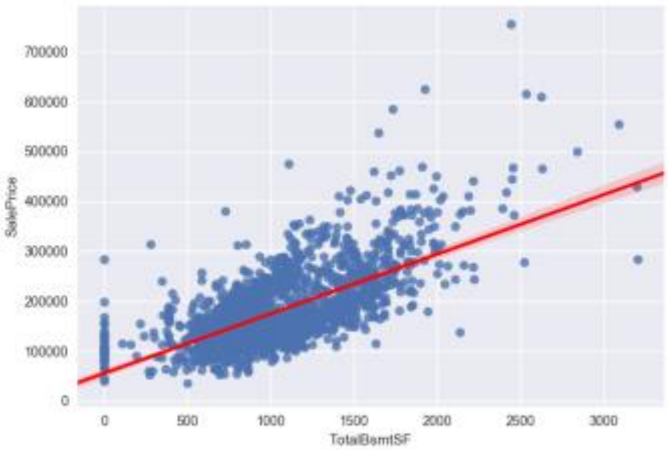
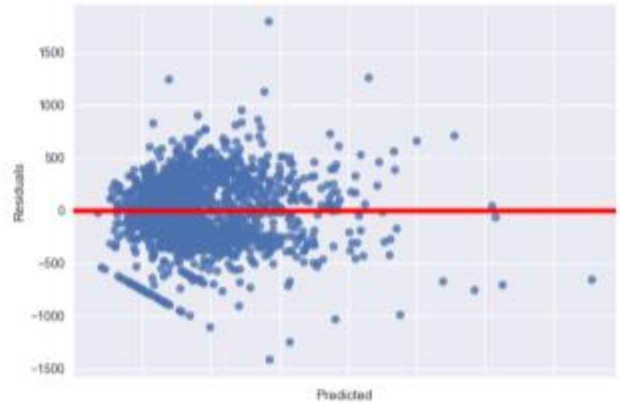
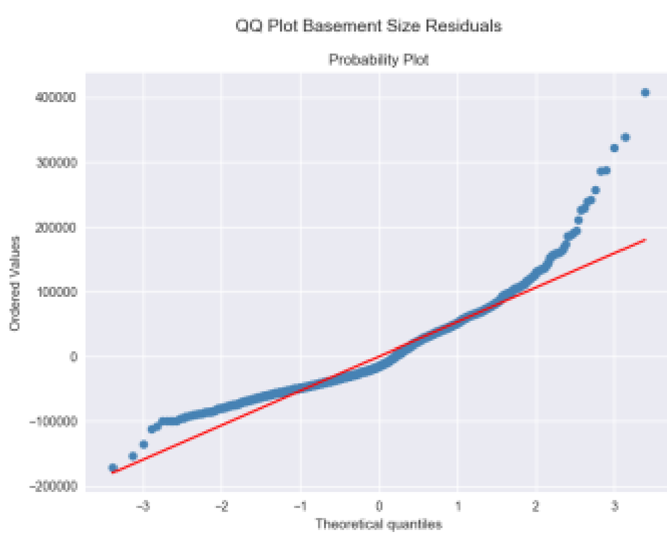


FIGURE 6 – RESIDUALS VERSUS FITTED VALUES



As with the first model, the data plus fitted line, and the residuals versus predicted graphs show heteroscedastic, non-normal data.

FIGURE 7 - QQ PLOT OF RESIDUALS



This model's QQ line starts above the theoretical line, whereas in Model 1 it started below. But like Model 1, this QQ plot veers sharply upward toward the right side of the chart. These residuals are also non-normal.

Multiple Linear Regression Model (MR1)

Creating a multiple regression model using both the GrLivArea and TotalBsmtSF predictor variables, yields the following result:

TABLE 3 - FITTED RESULTS FOR MODEL MR1

OLS Regression Results

Dep. Variable:	SalePrice	R-squared:	0.740
Model:	OLS	Adj. R-squared:	0.740
Method:	Least Squares	F-statistic:	2767.
Date:	Fri, 30 Jun 2017	Prob (F-statistic):	0.00
Time:	14:54:23	Log-Likelihood:	-23208.
No. Observations:	1943	AIC:	4.642e+04
Df Residuals:	1940	BIC:	4.644e+04
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-3.22e+04	2955.449	-10.895	0.000	-3.8e+04	-2.64e+04
TotalBsmtSF	74.4874	2.301	32.376	0.000	69.975	78.999
GrLivArea	89.7072	1.854	48.393	0.000	86.072	93.343

Omnibus:	246.364	Durbin-Watson:	1.390
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1407.130
Skew:	0.446	Prob(JB):	2.79e-306
Kurtosis:	7.072	Cond. No.	6.62e+03

The combination of the two variables together now explain 74% of the variation in price, per the R^2 value. The t-test values and the F-statistic remain significant, but the QQ and Residual distribution plots indicate caution in relying on the t-test and F-statistic.

The plot of the residuals is shown in Figure 8. Adding the two variables together gives a better result than either of the variables on their own. More predictor variables may not always improve the results; if the predictors are strongly collinear adding the extras will not improve the model.

FIGURE 8 - RESIDUALS VERSUS FITTED VALUES

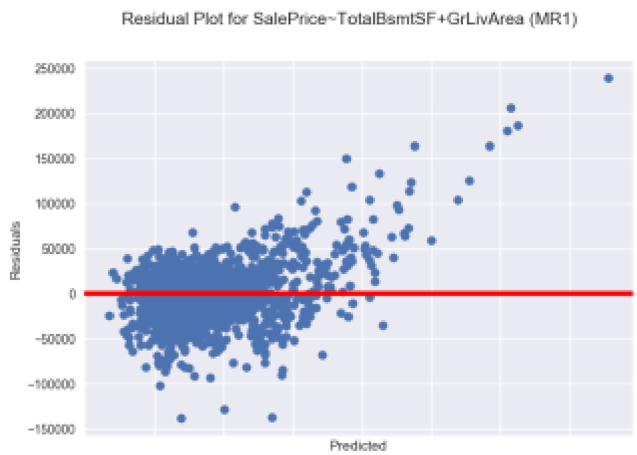
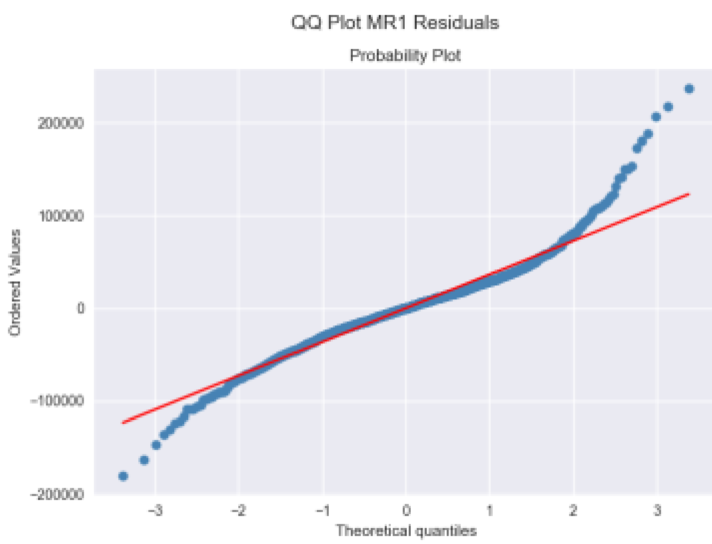


FIGURE 9 - QQ PLOT OF MULTIPLE REGRESSION RESIDUALS



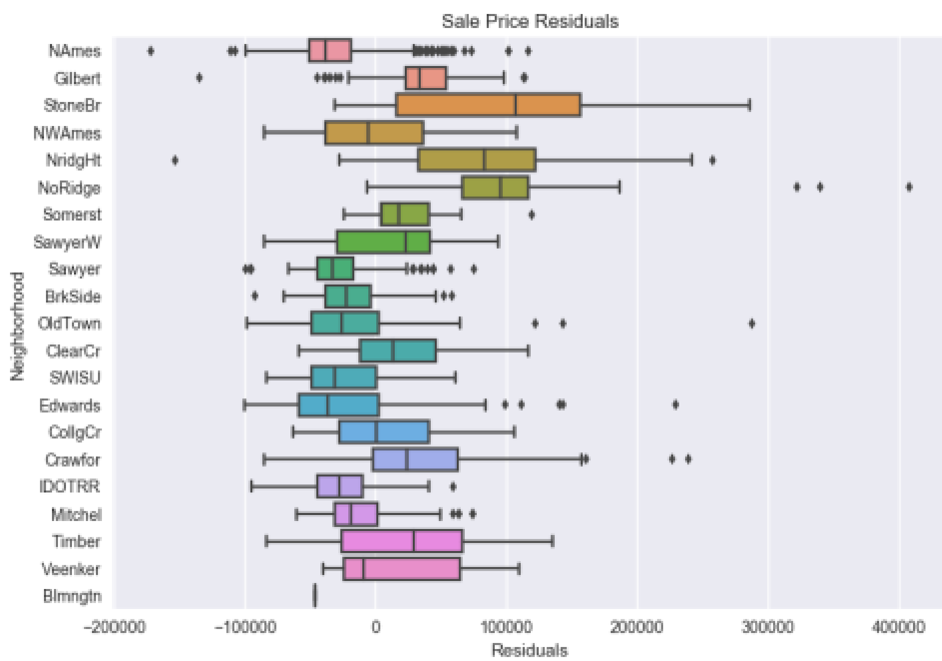
The residual results for the Multiple Regression still show issues with heteroscedasticity and non-normality. The QQ plot shows a "heavy tail" shape.

Multiple Regression #1, response variable = SalePrice, calculation for MAE = 26657.0235441

Neighborhood Accuracy

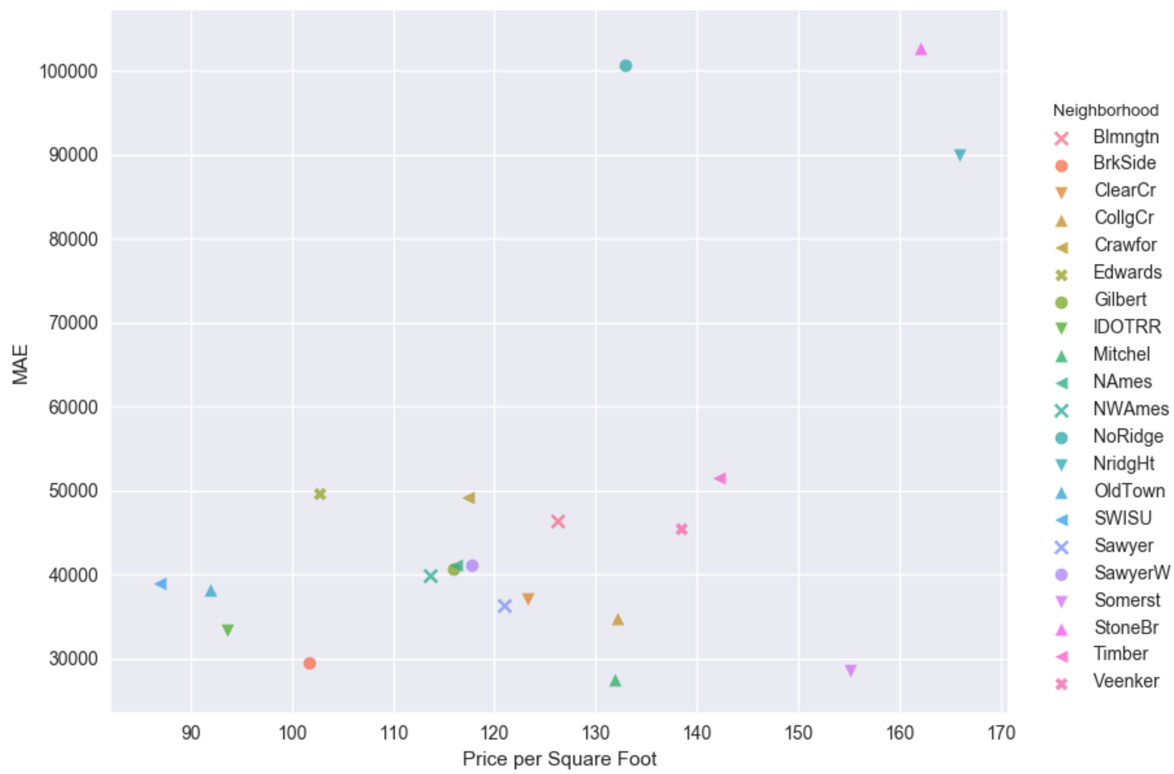
To assess model accuracy by neighborhood, we will first look at the boxplot of residuals, grouped by neighborhood.

FIGURE 10 – RESIDUALS BY NEIGHBORHOOD



We can see from the boxplot, there are neighborhoods which have predictions that tend to be routinely too high or too low. For example, North Ames, Sawyer, Brook Side, Edwards, IDOTRR and Mitchel seem to have chronical high predictions; their residuals' IQR boxes are below zero. Having an IRQ box below 0 would mean the interquartile range for those residuals is negative; the predicted value being higher than the actual value. Conversely, Stone Brook, North Ridge Heights, and Gilbert all have IRQ boxes above 0, indicating that their predicted prices are routinely lower than their actual prices. College Creek, Sawyer West and Northwest Ames center roughly around 0, indicating fairly accurate predictions for those neighborhoods.

FIGURE 11 – MEAN AVERAGE ERROR V. PRICE PER SQUARE FOOT, BY NEIGHBORHOOD



Looking at the MAE versus price per square foot for each neighborhood, Stone Brook has the overall highest mean absolute error, and one of the highest prices per square foot of house. While Mitchel has the lowest MAE and a middle-value price per square foot. I interpret the high MAE for Stone Brook as meaning the discrepancy between predicted and actual for homes in Mitchel is generally large. It is worth noting that North Ridge has a very high MAE and that it also has the 3 most extreme outliers of all the residuals. I believe the high MAE is related to the outliers.

New Multiple Regression model, using grouped indicator variables (MR2)

We were instructed to create and use grouped neighborhood bins. I used these to create my second model multiple regression model (MR2), whose results are shown in Table 4:

TABLE 4 - FITTED RESULTS FOR MODEL MR2

OLS Regression Results

Dep. Variable:	SalePrice	R-squared:	0.850
Model:	OLS	Adj. R-squared:	0.849
Method:	Least Squares	F-statistic:	997.2
Date:	Fri, 07 Jul 2017	Prob (F-statistic):	0.00
Time:	13:01:12	Log-Likelihood:	-22673.
No. Observations:	1943	AIC:	4.537e+04
Df Residuals:	1931	BIC:	4.544e+04
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.822e+04	6.67e+04	-0.273	0.785	-1.49e+05	1.13e+05
TotalBsmtSF	-11.5910	5.201	-2.229	0.026	-21.791	-1.391
n_bins	-4084.6074	2.07e+04	-0.198	0.843	-4.46e+04	3.65e+04
TotalBsmtSF:n_bins	22.1723	1.897	11.688	0.000	18.452	25.893
GrLivArea	56.3801	3.875	14.548	0.000	48.780	63.981
GrLivArea:n_bins	10.2406	1.605	6.382	0.000	7.094	13.388
ppsf	595.6632	712.419	0.836	0.403	-801.527	1992.854
ppsf:n_bins	-157.1382	106.147	-1.480	0.139	-365.312	51.036
mae	-2.3496	1.631	-1.441	0.150	-5.548	0.849
ppsf:mae	0.0321	0.017	1.911	0.056	-0.001	0.065
n_bins:mae	-0.2336	0.486	-0.481	0.631	-1.186	0.719
ppsf:n_bins:mae	-0.0019	0.002	-0.819	0.413	-0.006	0.003

Omnibus:	366.216	Durbin-Watson:	1.715
Prob(Omnibus):	0.000	Jarque-Bera (JB):	4127.409
Skew:	0.539	Prob(JB):	0.00
Kurtosis:	10.058	Cond. No.	2.19e+09

I built my second Multiple Regression model using the formula:

*SalePrice~TotalBsmtSF*n_bins+GrLivArea*n_bins+ppsf*n_bins*mae*

Computing the MAE for this new model yielded:

Multiple Regression #2, response variable = SalePrice, MAE = 19660.7109417.

This is an improvement over my first multiple regression model which has an MAE of 26657.0235441. Based solely on the MAE, the second model is a better fit. This is consistent with the fitted results; model 1 has an R^2 of .740, while model 2 has an R^2 of .850

Comparison of Models using Y versus those using log(Y)

Model 3 (MR3) – Multiple Regression, with Y = SalePrice

For this portion of the exercise we are allowed to expand our predictor variable selections. We must have at least four continuous variables which will be used to predict first SalePrice then log(SalePrice). I added 3 predictor variables to MR2, reduced the interaction terms, and settled on the following formula:

SalePrice~ TotalBsmtSF+GrLivArea+n_bins+mae+GarageArea+BsmtFinSF1+OverallQual

which gave me the results shown in Table 5.

TABLE 5 - FITTED RESULTS FOR MODEL MR3

OLS Regression Results

Dep. Variable:	SalePrice	R-squared:	0.882
Model:	OLS	Adj. R-squared:	0.882
Method:	Least Squares	F-statistic:	2070.
Date:	Fri, 07 Jul 2017	Prob (F-statistic):	0.00
Time:	13:14:14	Log-Likelihood:	-22441.
No. Observations:	1943	AIC:	4.490e+04
Df Residuals:	1935	BIC:	4.494e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-9.763e+04	2701.622	-36.136	0.000	-1.03e+05	-9.23e+04
TotalBsmtSF	21.1093	1.990	10.605	0.000	17.206	25.013
GrLivArea	57.5635	1.584	36.338	0.000	54.457	60.670
n_bins	1.043e+04	772.776	13.499	0.000	8915.884	1.19e+04
mae	0.5269	0.043	12.158	0.000	0.442	0.612
GarageArea	28.6512	3.734	7.673	0.000	21.328	35.974
BsmtFinSF1	25.7569	1.650	15.609	0.000	22.521	28.993
OverallQual	1.617e+04	647.955	24.950	0.000	1.49e+04	1.74e+04

Omnibus:	682.330	Durbin-Watson:	1.712
Prob(Omnibus):	0.000	Jarque-Bera (JB):	8794.860
Skew:	1.278	Prob(JB):	0.00
Kurtosis:	13.105	Cond. No.	2.22e+05

The plot of residuals for MR3 looks a lot like the one for MR1, with the exception of having a smaller lower bound on the residuals.

FIGURE 12

The QQ plot looks roughly the same at the one for MR1. It still looks strongly clustered to the left with a log tail to the right.

FIGURE 13

Multiple Regression #3, response variable = SalePrice, has MAE = 17466.2979491.

Model 4 – Multiple Regression, with Y = log(SalePrice) (MR4)

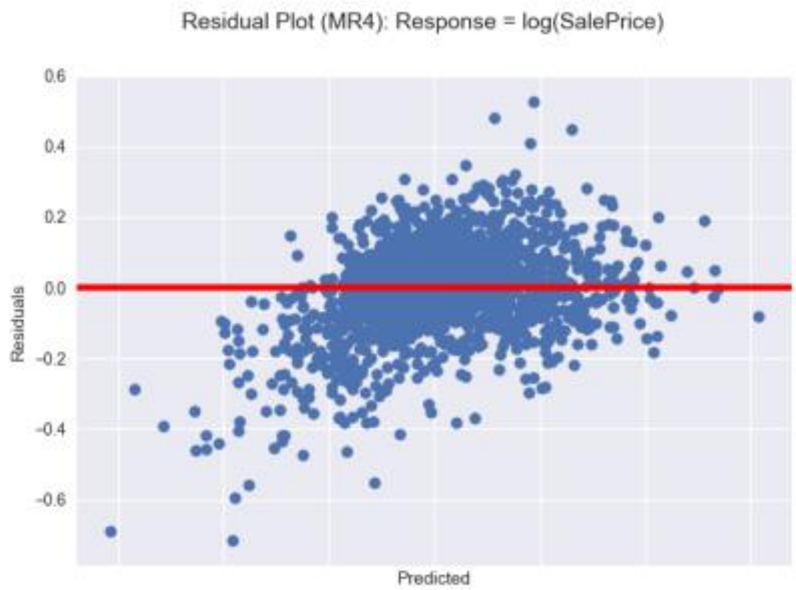
Using the same basic formula as MR3, changing the Response variable to be the log(SalePrice) produced the following fit values:

TABLE 6 - FITTED RESULTS FOR LOG(SALEPRICE) MODEL

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                logSale   R-squared:                       0.887
Model:                            OLS   Adj. R-squared:                  0.886
Method:                 Least Squares   F-statistic:                     2159.
Date:                Fri, 07 Jul 2017   Prob (F-statistic):               0.00
Time:                        13:19:44   Log-Likelihood:                 1289.6
No. Observations:                1943   AIC:                            -2563.
Df Residuals:                    1935   BIC:                            -2519.
Df Model:                           7
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept      10.6172      0.013    791.664      0.000      10.591      10.644
TotalBsmtSF   8.93e-05   9.88e-06      9.037      0.000    6.99e-05       0.000
GrLivArea       0.0003   7.86e-06     38.537      0.000       0.000       0.000
n_bins          0.0601      0.004     15.675      0.000       0.053       0.068
mae         -3.959e-07   2.15e-07     -1.840      0.066   -8.18e-07    2.61e-08
GarageArea      0.0002   1.85e-05     10.569      0.000       0.000       0.000
BsmtFinSF1      0.0001   8.19e-06     14.927      0.000       0.000       0.000
OverallQual     0.0991      0.003     30.817      0.000       0.093       0.105
==============================================================================
Omnibus:                      261.707   Durbin-Watson:                   1.602
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              723.417
Skew:                          -0.718   Prob(JB):                    8.17e-158
Kurtosis:                       5.621   Cond. No.                     2.22e+05
==============================================================================
```

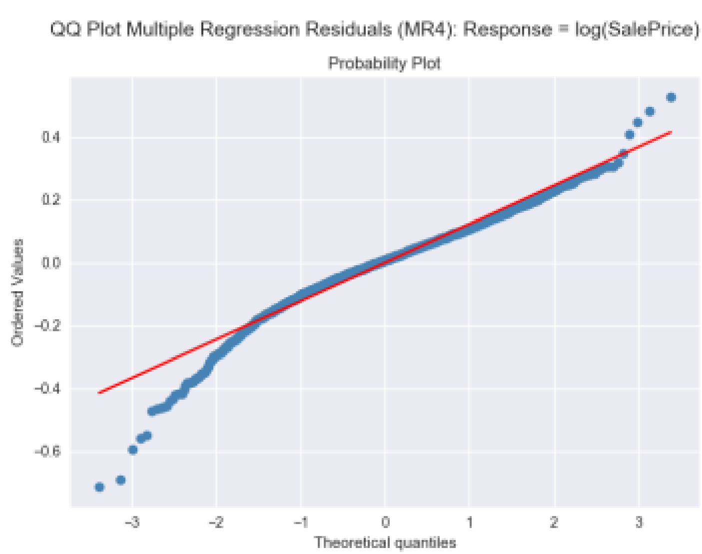
In Figure 14 we can see that the residual plot is much more normally distributed.

FIGURE 14



The QQ plot of residuals shows better fit for larger values, but more deviation for the smaller values. It also fits the normal line for a greater distance than any of the previous models.

FIGURE 15



Multiple Regression #4, response variable = log(SalePrice), MAE = 0.0921212173334

Comparison of MR3 (Y=SalePrice) and MR4 (Y=log(SalePrice))

Looking at the R2 values for MR3 and MR4, there is only a small difference between .882 and .887, although the MR4 value of .887 is better. The MAE values are distinctly different with MR3 having 17466.30 and MR4 having .092. The graphical representation of the residuals for MR4 are more normally distributed and indicate the superior model. Finally, the QQ plot for MR4 tracks the normal line for a greater range than does the plot for MR3, which I interpret as meaning it can be considered normal for the values away from the ends. Its overall shape appears a better fit to normal than the one for MR3.

It looks like a log transformation could be beneficial when you are working with a model whose residuals have a long right tail, and you want to normalize the data to some degree. Another potential transformation might be taking the square root of the response variable. I suspect there are many different transformations that are helpful, depending on the data and your goal. I am certain we will learn more about this in the coming weeks.

Model 5 – Multiple Regression, with Y = sqrt(SalePrice) (MR5)

Per the instructions, I fit a model using the sqrt(SalePrice) transformation. I used the same formula as in the MR4 model, only changing the response variable. It yielded the following results:

TABLE 7 – FITTED RESULTS FOR SQRT(SALEPRICE) MODEL

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               sqrtSale   R-squared:                       0.903
Model:                            OLS   Adj. R-squared:                  0.902
Method:                 Least Squares   F-statistic:                     2564.
Date:                Sat, 08 Jul 2017   Prob (F-statistic):               0.00
Time:                        13:52:28   Log-Likelihood:                -8998.2
No. Observations:                1943   AIC:                         1.801e+04
Df Residuals:                    1935   BIC:                         1.806e+04
Df Model:                           7
Covariance Type:            nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept    109.8659      2.673     41.104      0.000     104.624     115.108
TotalBsmtSF    0.0211      0.002     10.707      0.000       0.017       0.025
GrLivArea      0.0649      0.002     41.390      0.000       0.062       0.068
n_bins        12.4729      0.765     16.314      0.000      10.973      13.972
mae            0.0002   4.29e-05      5.807      0.000       0.000       0.000
GarageArea     0.0373      0.004     10.108      0.000       0.030       0.045
BsmtFinSF1     0.0273      0.002     16.692      0.000       0.024       0.030
OverallQual   19.6316      0.641     30.624      0.000      18.374      20.889
==============================================================================
Omnibus:                       88.865   Durbin-Watson:                   1.654
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              256.647
Skew:                          -0.165   Prob(JB):                     1.86e-56
Kurtosis:                       4.750   Cond. No.                     2.22e+05
==============================================================================
```

FIGURE 16

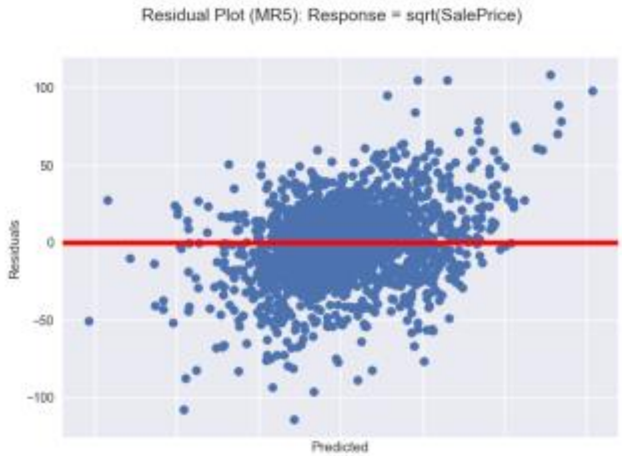
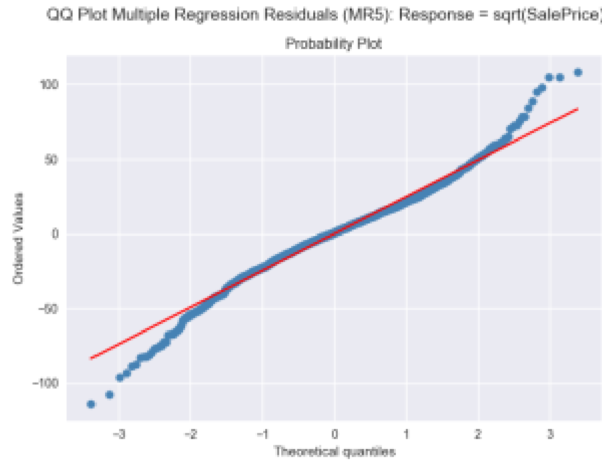


FIGURE 17



Taking the square root of the response variable does look like it can be beneficial in some cases. Both the QQ plot of residuals and the residuals versus predicted are better than for the untransformed case. Both plots look very similar to those for the log transformation. The R^2 value is better than for the log transformation, but the MAE value is higher.

Multiple Regression #5, response variable = sqrt(SalePrice), MAE = 18.5188194156

Conclusions

Working with this data we have demonstrated how adding variables to a model helps increase the R^2 value. The difference between MR1's R^2 value of .740 and MR2's R2 value of .831 is a considerable improvement. Adding interaction terms also turned out to be beneficial, helping me reach an R^2 of .882 for MR3.

By transforming the response variable, we were able to improve the R^2 value, but more importantly, we greatly reduced the Mean Absolute Error for those models. Additionally, the transformation had the effect of improving the normalcy of the residuals. Taking the log of the response variable is one viable transformation, taking the square root appears to be another. Depending on the data (right skew, left skew) there may be different standard techniques for doing transformation.

In the case of the Ames data, transformation was helpful in so far as it normalized the data, and lowered the Mean Absolute Error. Of the models fit, I feel MR4, the log(SalePrice) was the best, and the one I would recommend moving forward with. While not all the variation is explained by that model, it does explain roughly 89% which seems like a good starting point.

Appendix

Code used in generating this report

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Tues Ju1 04 11:01:38 2017

@author: tamtwill
"""

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib.sankey import Sankey
import numpy as np
from sklearn.metrics import mean_absolute_error
from statsmodels.formula.api import ols
import scipy.stats as stats

df = pd.read_csv('/Users/tamtwill/NorthwesternU_MSPA/410 -
Regression/Week1_LR/ames_housing_data.csv', sep = ",")
obs0 = len(df)
sankey0 = "Starting number = "+ str(obs0)

# drop the non-residential properties first
resid_only = df[((df['Zoning'] == 'RL') |(df['Zoning'] == 'RM') |
(df['Zoning'] == 'RH')|(df['Zoning'] == 'RP'))]
obs1 = len(resid_only)
sankey1 = "Drop all properties NOT zoned residential, # remaining = " + str(obs1)
drop1 = obs0 - obs1
sandrop1 = "Dropped" + str(drop1)

# keep only single family detatched
family1 = resid_only[(resid_only['BldgType'] == '1Fam')]
obs2 = len(family1)
sankey2 = "Drop all properties NOT Single Family, # remaining = " + str(obs2)
drop2 = obs1 - obs2
```

```
# keep normal sales, getting rid of all the weird sale types
norm_only = family1[(family1['SaleCondition'] == 'Normal')]
obs3 = len(norm_only)
sankey3 = "Drop all sales where condidtions are NOT normal, # remaining = " + str(obs3)
drop3 = obs2 - obs3

# make a sankey chart showing the criteria and remaining number of observations
# for the data waterfall
fig = plt.figure(figsize=(8, 12))
ax = fig.add_subplot(1, 1, 1, xticks=[], yticks=[],
title="Waterfall of dropped observations")
obs = [obs0, obs1, obs2, obs3]
labels = ["Drop all properties NOT zoned residential", "Drop all properties NOT Single Family",
"Drop all sales where condidtions are NOT normal", "Remaining Observations"]
colors = ["#25EE46", "#2ADCB1", "#2ADCDC", "#20A6EE"]

sankey = Sankey(ax=ax, scale=0.0015, offset=0.3)
for input_obs, output_obs, label, prior, color in zip(obs[:-1], obs[1:],
labels, [None, 0, 1, 2, 3], colors):
if prior != 1:
sankey.add(flows=[input_obs, -output_obs, output_obs - input_obs],
orientations=[0, 0, 1],
patchlabel=label,
labels=['', None, 'dropped'],
prior=prior,
connect=(1, 0),
pathlengths=[0, 0, 2],
trunklength=10.,
rotation=-90,
facecolor=color)
else:
sankey.add(flows=[input_obs, -output_obs, output_obs - input_obs],
orientations=[0, 0, 1],
patchlabel=label,
labels=['', labels[-1], 'dropped'],
prior=prior,
connect=(1, 0),
pathlengths=[0, 0, 2],
trunklength=10.,
rotation=-90,
facecolor=color)
diagrams = sankey.finish()
for diagram in diagrams:
diagram.text.set_fontweight('bold')
diagram.text.set_fontsize('10')
```

```
for text in diagram.texts:
text.set_fontsize('10')
ylim = plt.ylim()
plt.ylim(ylim[0]*1.05, ylim[1])
plt.show()


df_houses = norm_only

# fit the regression line for above grade living area v saleprice and plot
# ---------------------------------------------------------------------------
x=df_houses['GrLivArea']
y=df_houses['SalePrice']

my_lm = ols('SalePrice~ GrLivArea',data = df_houses)
results = my_lm.fit()
print results.summary()
print ''

fig = plt.figure()
fig.suptitle('Data and fitted regression line - SalePrice v Living Area', fontsize=14)
fig= sns.regplot(x,y, line_kws = {'color':'red'})
plt.show()


fig = plt.figure()
ax = fig.add_subplot(111)
fig = sns.residplot(x='SalePrice',y = 'GrLivArea', data = df_houses, )
fig.set(xticklabels=[])
ax.set_ylabel('Residuals')
ax.set_xlabel('Predicted')
plt.axhline(linewidth=4, color='r')
plt.show()

my_predicts = results.fittedvalues
my_res = results.resid
my_res.describe()

fig = plt.figure()
fig.suptitle('QQ Plot Living Area Residuals', fontsize=14)
ax = fig.add_subplot(111)
qqp = stats.probplot(my_res, dist="norm", plot=plt, );
ax.get_lines()[0].set_markerfacecolor('steelblue')
plt.show()

# fit the regression line for Basement v saleprice and plot
```

```
# ---------------------------------------------------------------------------
x=df_houses['TotalBsmtSF']
y=df_houses['SalePrice']
my_lm = ols('SalePrice~ TotalBsmtSF',data = df_houses).fit()
print my_lm.summary()
print ''

fig = plt.figure()
fig.suptitle('Data and fitted regression line - SalePrice v Bassement Area', fontsize=14)
fig= sns.regplot(x,y, line_kws = {'color':'red'})
plt.show()

my_predicts = my_lm.fittedvalues
my_res = my_lm.resid
my_res.describe()

fig = plt.figure()
ax = fig.add_subplot(111)
fig.suptitle('QQ Plot Basement Size Residuals', fontsize=14)
stats.probplot(my_res, dist="norm", plot=plt, )
ax.get_lines()[0].set_markerfacecolor('steelblue')
plt.show()

fig = plt.figure()
ax = fig.add_subplot(111)
fig = sns.residplot(x='SalePrice',y = 'TotalBsmtSF', data = df_houses)
fig.set(xticklabels=[])
ax.set_ylabel('Residuals')
ax.set_xlabel('Predicted')
plt.axhline(linewidth=4, color='r')
plt.show()



# multiple regression model
# ---------------------------------------------------------------------------
my_lm1 = ols(formula = 'SalePrice~ TotalBsmtSF+GrLivArea',data = df_houses).fit()
print my_lm1.summary()

#fig = plt.figure()
fig, ax = plt.subplots()
plt.scatter(df_houses['SalePrice'],my_lm1.resid)
plt.ylabel('Residuals')
plt.xlabel('Predicted')
fig.suptitle('Residual Plot for SalePrice~TotalBsmtSF+GrLivArea (MR1)', fontsize=14)
plt.axhline(linewidth=4, color='r')
```

```
ax.tick_params(labelbottom='off')
plt.show()


my_predicts1 = my_lm1.fittedvalues
my_res1 = my_lm1.resid
my_res1.describe()
mult_mae1 = mean_absolute_error(df_houses['SalePrice'], my_predicts1)
print ''
print "Multiple Regression #1, response variable = SalePrice, MAE = ", mult_mae1
print ''

fig = plt.figure()
fig.suptitle('QQ Plot MR1 Residuals', fontsize=14)
ax = fig.add_subplot(111)
stats.probplot(my_res1, dist="norm", plot=plt, )
ax.get_lines()[0].set_markerfacecolor('steelblue')
plt.show()

# Neighborhood accuracy section
# ----------------------------------------------------------------------------

# Create dataframe with residuals and neighborhood, by appending residuals to
# df_houses, adding the residuals and finding the difference between actual and
# predicted SalePrice

df_res = df_houses.copy(deep=True)
df_res = df_res[['SalePrice', 'Neighborhood', 'GrLivArea']]
df_res['Residuals'] = my_res
df_res['Abs_res'] = df_res['Residuals'].abs()
#df_res['Diff'] = df_res['SalePrice'] - df_res['Predicted']


fig = plt.figure()
fig = sns.boxplot(x="Residuals", y="Neighborhood", orient = 'h', data=df_res)
fig.set_title('Sale Price Residuals')
plt.show()

grouped_df = df_res.groupby(['Neighborhood'])
grouped_df.describe()

# Sum all the SF for each Neighborhood
tot_SF = df_res['GrLivArea'].groupby([df_res['Neighborhood']]).sum()

# Sum all the prices paid in each neighborhood
tot_pr = df_res['SalePrice'].groupby([df_res['Neighborhood']]).sum()
```

```
# Find price per SF by dividing sum of all prices by sum of all area by neighborhood
# Note to self - access neighborhood name as pr_per-SF.loc['name']
pr_per_SF = tot_pr.div(tot_SF)

# count the number of houses in each neighborhood
houses = df_res['SalePrice'].groupby([df_res['Neighborhood']]).count()

# get the total of the absolute value of the residuals for each neighborhood
tot_abs_res = df_res['Abs_res'].groupby([df_res['Neighborhood']]).sum()


# Compute the MAE as the sum of abs(y minus y-hat) over n, where y - y_hat is the
# same as the residual. So, for each neighborhood, get
# sum of abs(residuals))/count of houses.

i = 0
mae_list = []
neighborhood_list = []
sf_list = []
for neighborhood, group in df_res.groupby('Neighborhood'):
n=neighborhood
neighborhood_list.append(n)
mae = round(tot_abs_res[i]/houses[i], 2)
mae_list.append(mae)
sf_list.append(round(pr_per_SF[i], 2))
i+=1
# make a dataframe out of the SF and MAE values
df_mae = pd.DataFrame(neighborhood_list)
df_mae['ppsf'] = pd.Series(sf_list, index = df_mae.index)
df_mae['mae'] = pd.Series(mae_list, index = df_mae.index)
df_mae.rename(columns = {0:'Neighborhood'}, inplace = True)
# per item 4 on the assignment, plot df_mae
sort_mae = df_mae.sort(['Neighborhood'])
fig = plt.figure()
my_marks = ['x','o','v','^','<', 'X','o','v','^','<', 'x','o','v','^','<', 'x','o','v','^','<','X']
fig = sns.lmplot(x='ppsf', y="mae", data=sort_mae, hue='Neighborhood', markers=my_marks,
fit_reg=False)
# some code from stackoverflow to get the legend out of the way
for ax in fig.axes.flat:
box = ax.get_position()
ax.set_position([box.x0,box.y0,box.width*0.85,box.height])
ax.set_ylabel('MAE')
ax.set_xlabel('Price per Square Foot')
sns.plt.show()
```

```
# create groups based on price per sq ft
# --------------------------------------------------------------------------
df_mae_sorted = df_mae.sort(columns='ppsf')
i=0
bin_list = []
for neighborhood in df_mae['Neighborhood']:
if df_mae.loc[i]['ppsf'] <= 102.00 :
bin_list.append(1)
elif 102.00 < df_mae.loc[i]['ppsf'] <= 118.00 :
bin_list.append(2)
elif 118.00 < df_mae.loc[i]['ppsf'] <= 134.00 :
bin_list.append(3)
elif 134.00 < df_mae.loc[i]['ppsf'] <= 150.00 :
bin_list.append(4)
else:
bin_list.append(5)
i+=1
df_mae['n_bins'] = pd.Series(bin_list, index = df_mae.index)

# new multiple regression model
# --------------------------------------------------------------------------
# add the data just calculated back into the main dataframe by leveraging merge
# on Neighborhood
target_cols = ['Neighborhood', 'ppsf', 'mae', 'n_bins']
df_merge = df_houses.merge(df_mae[target_cols], on='Neighborhood', how='left')

my_lm1 = ols(formula = 'SalePrice~ TotalBsmtSF+GrLivArea+ppsf+mae+n_bins',data =
df_merge).fit()
print my_lm1.summary()
print ''
print ''


my_lm2 = ols(formula = 'SalePrice~
TotalBsmtSF*n_bins+GrLivArea*n_bins+ppsf*n_bins*mae',data = df_merge).fit()
print my_lm2.summary()

my_predicts2 = my_lm2.fittedvalues
my_res2 = my_lm2.resid
my_res2.describe()
mult_mae2 = mean_absolute_error(df_houses['SalePrice'], my_predicts2)
print ''
print "Multiple Regression #2, response variable = SalePrice, MAE = ", mult_mae2
print ''
```

```
# new multiple regression model comparing Y versus log(Y)
# --------------------------------------------------------------------------

# create log(SalePrice) column
df_log = df_merge[['TotalBsmtSF','GrLivArea','SalePrice', 'GarageArea','BsmtFinSF1',
'OverallQual','Neighborhood', 'ppsf', 'mae', 'n_bins']]
df_log['logSale'] = np.log(df_log.SalePrice)

my_lm3 = ols(formula = 'SalePrice~
TotalBsmtSF+GrLivArea+n_bins+mae+GarageArea+BsmtFinSF1+OverallQual',data =
df_merge).fit()
print my_lm3.summary()


#fig = plt.figure()
fig, ax = plt.subplots()
plt.scatter(df_houses['SalePrice'],my_lm3.resid)
plt.ylabel('Residuals')
plt.xlabel('Predicted')
fig.suptitle('Residual Plot (MR3): Response = SalePrice', fontsize=14)
ax.tick_params(labelbottom='off')
plt.axhline(linewidth=4, color='r')
plt.show()

my_predicts3 = my_lm3.fittedvalues
my_res3 = my_lm3.resid

mult_mae3 = mean_absolute_error(df_houses['SalePrice'], my_predicts3)
print ''
print "Multiple Regression #3, response variable = SalePrice, MAE = ", mult_mae3
print ''

fig = plt.figure()
ax=fig.add_subplot(111)
fig.suptitle('QQ Plot Multiple Regression Residuals (MR3): Response = SalePrice', fontsize=14)
stats.probplot(my_res3, dist="norm", plot=plt)
ax.get_lines()[0].set_markerfacecolor('steelblue')
plt.show()

# now, with log(SalePrice) as the response variable
my_lm4 = ols(formula = 'logSale~
TotalBsmtSF+GrLivArea+n_bins+mae+GarageArea+BsmtFinSF1+OverallQual',data =
df_log).fit()
print my_lm4.summary()
```

```
#fig = plt.figure()
fig, ax = plt.subplots()
plt.scatter(df_log['logSale'],my_lm4.resid)
plt.ylabel('Residuals')
plt.xlabel('Predicted')
fig.suptitle('Residual Plot (MR4): Response = log(SalePrice)', fontsize=14)
plt.axhline(linewidth=4, color='r')
ax.tick_params(labelbottom='off')
plt.show()

my_predicts4 = my_lm4.fittedvalues
my_res4 = my_lm4.resid
my_res4.describe()
mult_mae4 = mean_absolute_error(df_log['logSale'], my_predicts4)
print ''
print "Multiple Regression #4, response variable = log(SalePrice), MAE = ", mult_mae4
print ''


fig = plt.figure()
ax = fig.add_subplot(111)
fig.suptitle('QQ Plot Multiple Regression Residuals (MR4): Response = log(SalePrice)',
fontsize=14)
stats.probplot(my_res4, dist="norm", plot=plt, )
ax.get_lines()[0].set_markerfacecolor('steelblue')
plt.show()

# new multiple regression model trying sqrt(Y)
# --------------------------------------------------------------------------

# create sqrt(SalePrice) column
df_sr = df_merge[['TotalBsmtSF','GrLivArea','SalePrice', 'GarageArea','BsmtFinSF1',
'OverallQual','Neighborhood', 'ppsf', 'mae', 'n_bins']]
df_sr['sqrtSale'] = np.sqrt(df_sr.SalePrice)

my_lm5 = ols(formula = 'sqrtSale~
TotalBsmtSF+GrLivArea+n_bins+mae+GarageArea+BsmtFinSF1+OverallQual',data =
df_sr).fit()
print ' '
print my_lm5.summary()
print ' '

#fig = plt.figure()
fig, ax = plt.subplots()
plt.scatter(df_log['logSale'],my_lm5.resid)
```

```
plt.ylabel('Residuals')
plt.xlabel('Predicted')
fig.suptitle('Residual Plot (MR5): Response = sqrt(SalePrice)', fontsize=14)
plt.axhline(linewidth=4, color='r')
ax.tick_params(labelbottom='off')
plt.show()

my_predicts5 = my_lm5.fittedvalues
my_res5 = my_lm5.resid
my_res5.describe()
mult_mae5 = mean_absolute_error(df_sr['sqrtSale'], my_predicts5)
print ''
print "Multiple Regression #5, response variable = sqrt(SalePrice), MAE = ", mult_mae5
print ''


fig = plt.figure()
ax = fig.add_subplot(111)
fig.suptitle('QQ Plot Multiple Regression Residuals (MR5): Response = sqrt(SalePrice)',
fontsize=14)
stats.probplot(my_res5, dist="norm", plot=plt, )
ax.get_lines()[0].set_markerfacecolor('steelblue')
plt.show()
```