Introduction

In this report, we are working with the Ames Housing data set. We are investigating the “SalePrice” response variable for homes in Ames. In order to have meaningful data to use in building our model, the data will be filtered and a subset will be used to build models. Specifically, we will build linear regression models for two separate variables and a third model which looks at the combination of those two variables. Finally, we will transform “SalePrice” and re-evaluate the relationship between the transformed response variable and the two predictor variables and their combined effect. The primary tools used in this report are scatterplots, fitted plots, residual plots, and summary data from linear models. We will see that in this particular case, transforming the response variable has a small effect, and that effect could be considered beneficial as it helps center the residual plots.

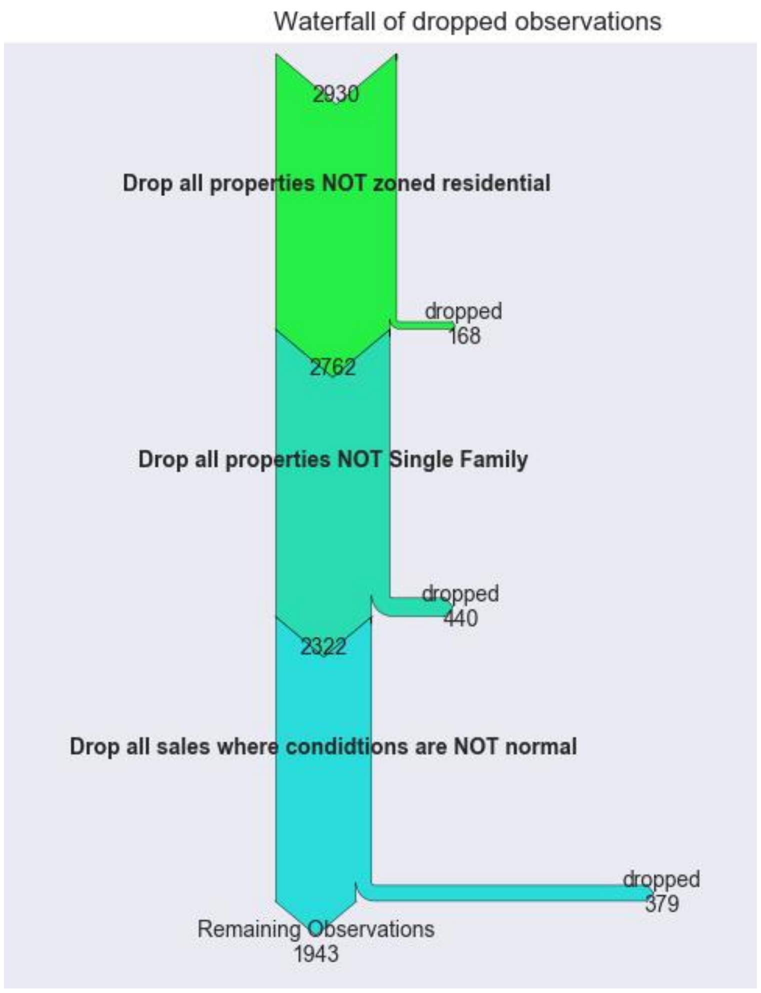
Sample Definition

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 an 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 DROPS



Exploratory Data Analysis

For building a Simple Linear Regression model, we will want to work with continuous, ordinal data. Of the 82 columns in the data frame, only 15 meet that criteria. I split the 15 columns into 2 separate data frames, retaining the SalePrice column in each frame. This allowed me to generate scatterplot matrices of the columns to evaluate possible candidate predictor variables. The scatterplots are shown in Figures 2 and 3.

FIGURE 2 - SCATTERPLOT MATRIX FOR FIRST HALF OF CONTINUOUS VARIABLES

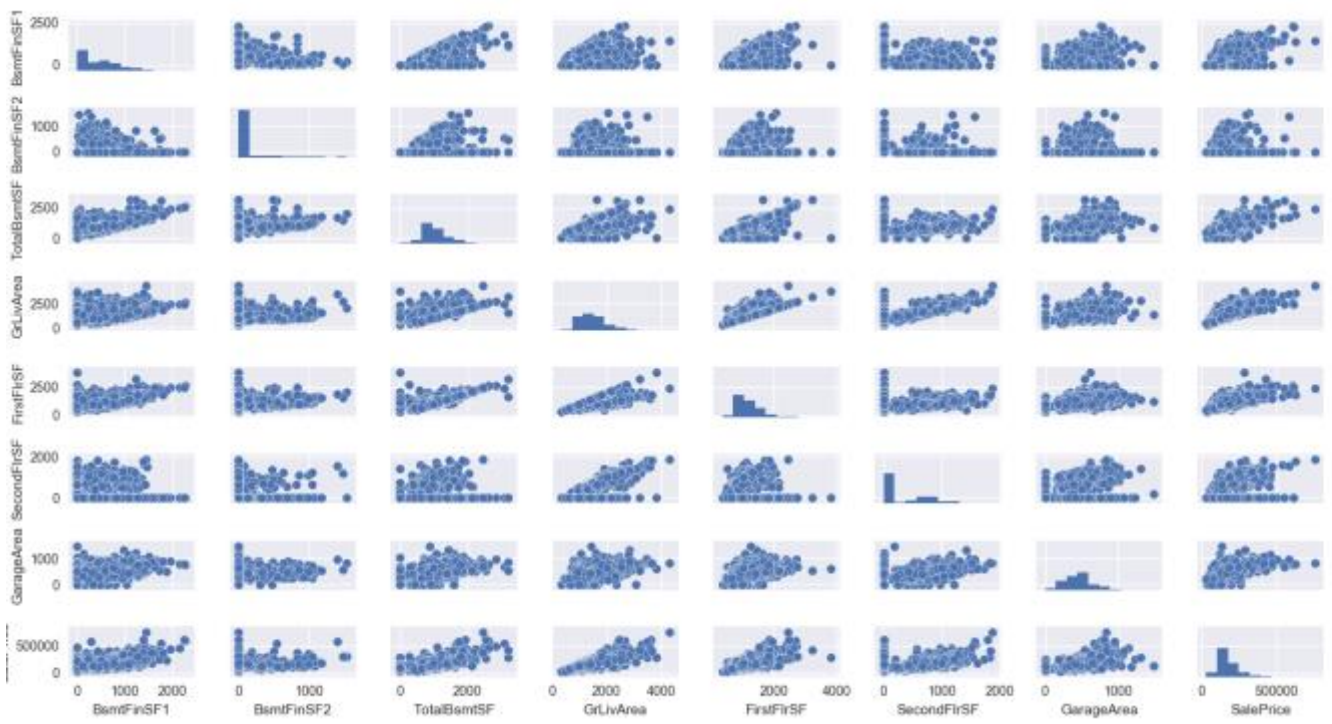
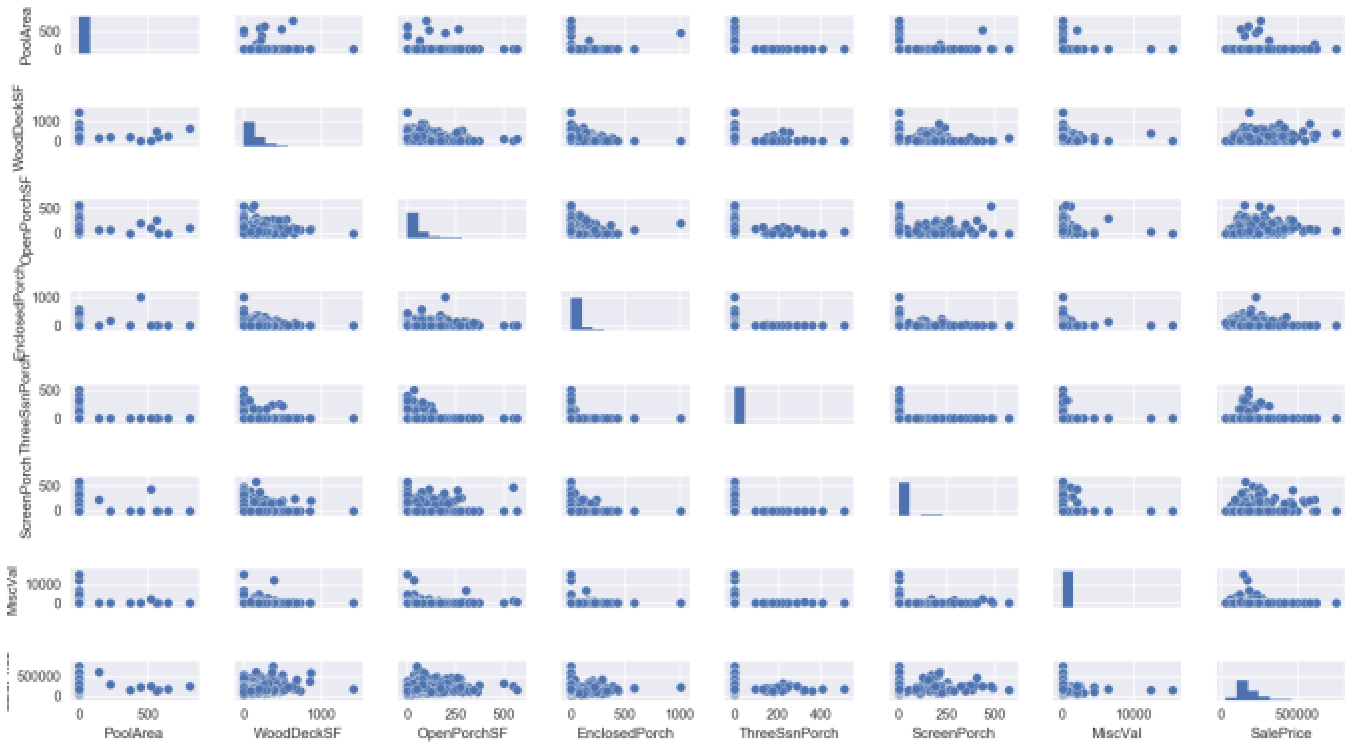


FIGURE 3 - SCATTERPLOT MATRIX FOR SECOND HALF OF CONTINUOUS VARIABLES



In selecting predictor variables, I sought scatterplots showing a mostly coherent cluster of points which fell along a line, indicating a linear relationship of some sort. The two most obvious variables to fit these criteria are GrLivArea and FirstFlrSF. However, the relationship between these two variables and the response variable seems virtually identical, so I opted to use GrLivArea, but not FirstFlrSF. I thought a better model would come from using variables that seem less collinear than GrLivArea and FirstFlrSF.

For the second variable, I thought the next best-looking scatterplot was TotalBsmtSF vs SalePrice. While not all homes have basements, the basement area seemed to provide the best visual distinction of all the remaining continuous variables scatterplots. The other variable plots are too dispersed, lack a linear clustering, or have significantly more 0 values than Basement Area.

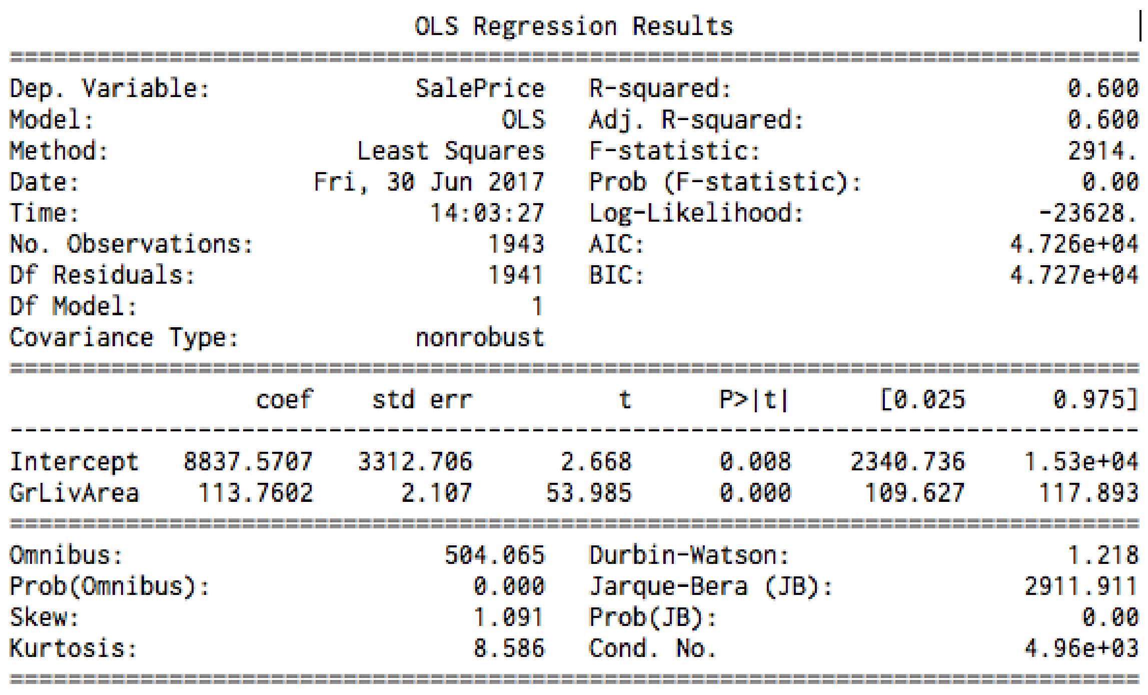
Simple Linear Regression Models

Model1: Above Grade Living Area Regression Model

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

Coefficient, Above Grade Living Area [0.00527633]
Intercept, Above Grade Living Area 549.450947909

TABLE 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 4, followed by a plot of the residuals in Figure 5.

FIGURE 4 – DATA AND FITTED REGRESSION LINE FOR LIVING AREA

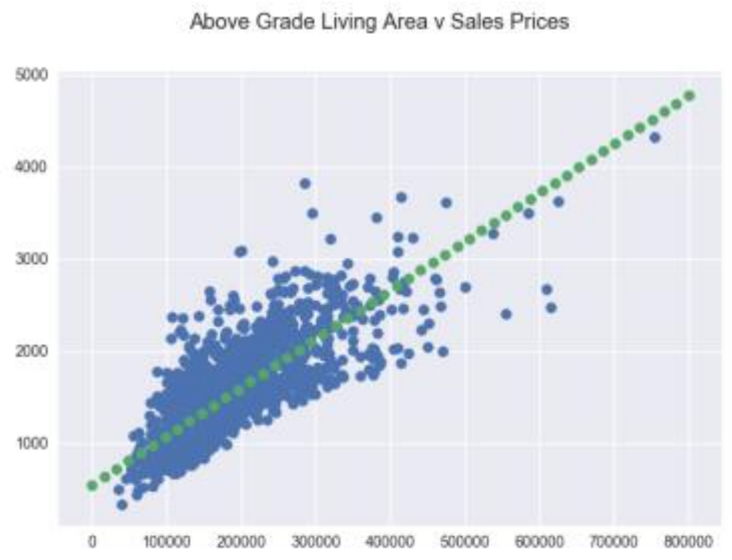
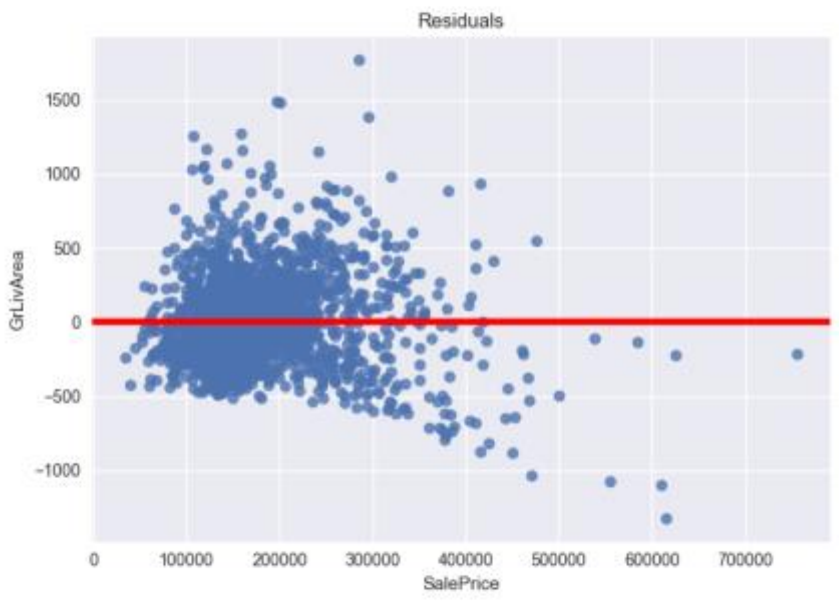


FIGURE 5 - RESIDUALS VERSUS FITTED VALUES



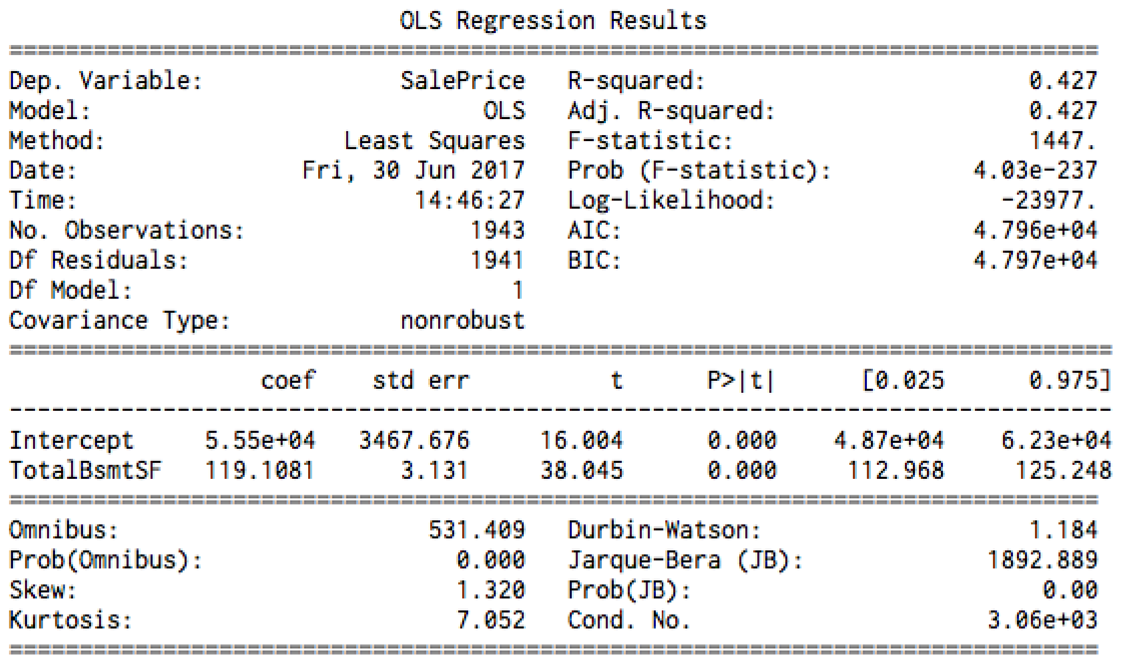
Both figures 4 and 5 show clustering on the left side, indicative of a long-tail skew to the right in home prices.

Model2: Total Basement Area Regression Model

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

Coefficient, Total Basement Area [0.00358642]
Intercept, Total Basement Area 392.364990034

TABLE 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. The R^2 value indicates ~43% of price variability is related to basement size. The data and regression line are shown in Figure 6 and the plot of the residuals is shown in Figure 7.

FIGURE 6 – DATA AND FITTED REGRESSION LINE FOR SIZE OF BASEMENT

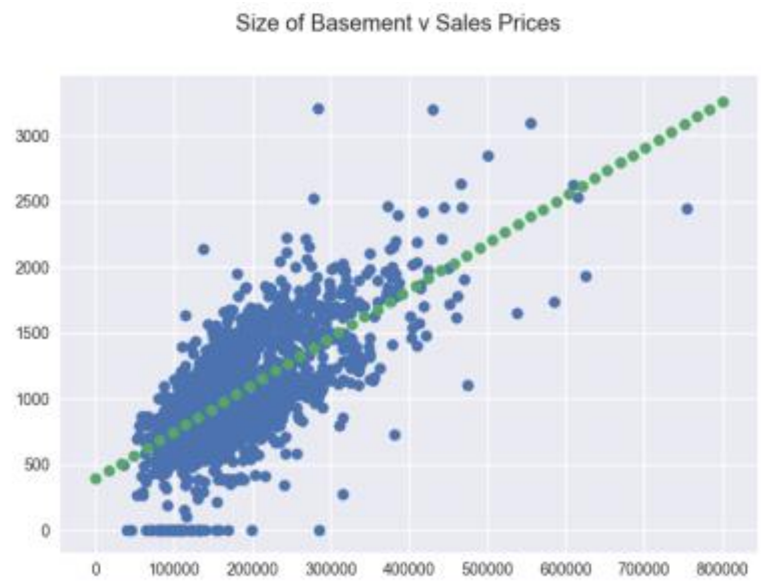
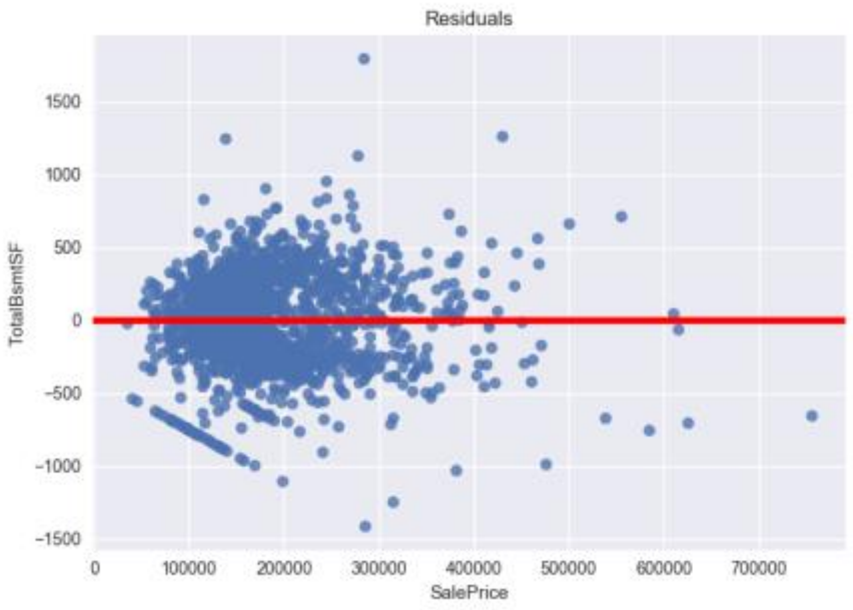


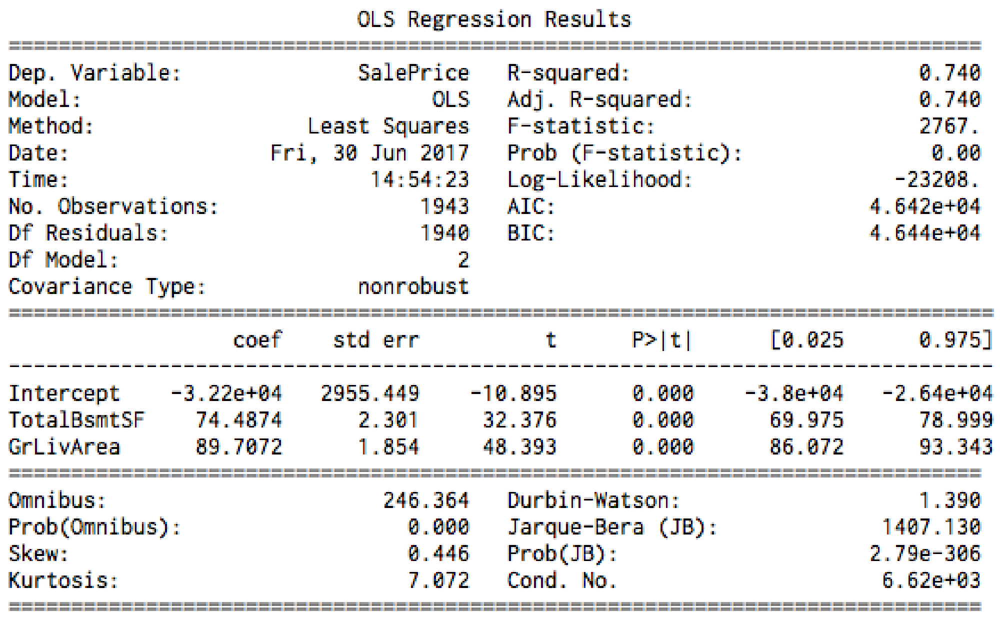
FIGURE 7 – RESIDUALS VERSUS FITTED VALUES



Multiple Linear Regression Model

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

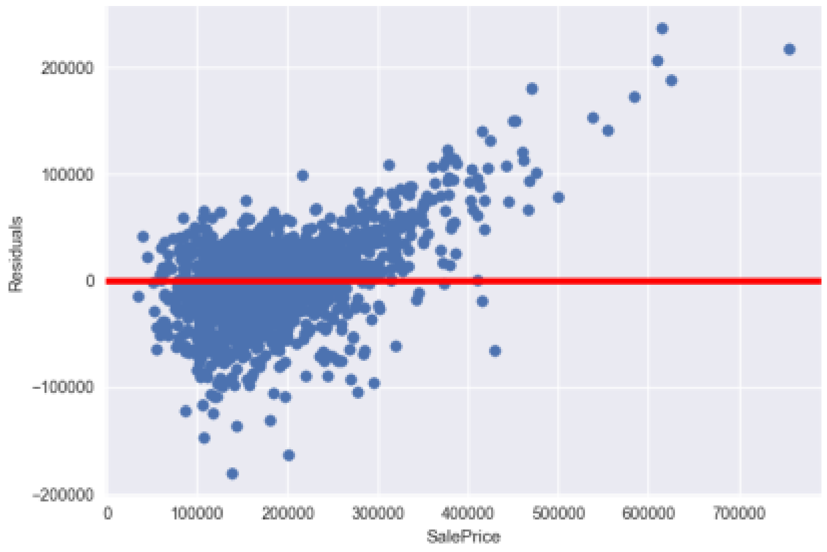
TABLE 3



```
                            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. T t-test values and the F-statistic remain significant. 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



The residuals here don't have the nicely dispersed characteristic that is desirable. They almost have a linear look to them.

log(SalePrice) Response Models

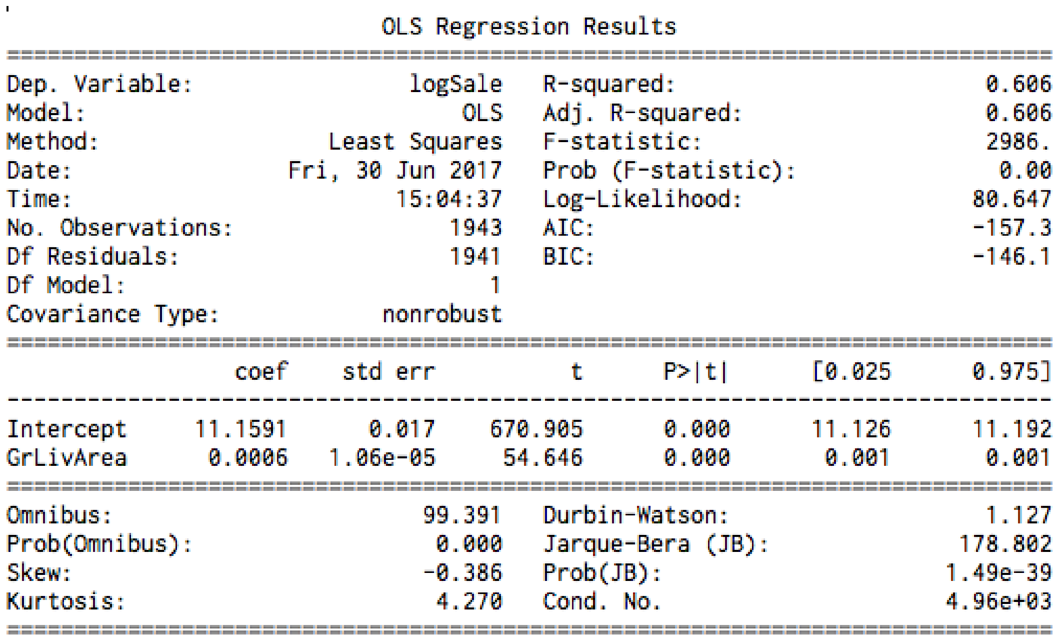
I transformed the response variable, SalePrice by taking the log of each observation. Using those log(SalePrice) values as the new response variable had the following effects.

Model1: Above Grade Living Area Regression Model

First looking at the simple regression for above grade living area:

Coefficient, Above Grade Living Area [1048.225784]
Intercept, Above Grade Living Area -11109.8850375

TABLE 4



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                logSale   R-squared:                       0.606
Model:                            OLS   Adj. R-squared:                  0.606
Method:                 Least Squares   F-statistic:                     2986.
Date:                Fri, 30 Jun 2017   Prob (F-statistic):               0.00
Time:                        15:04:37   Log-Likelihood:                 80.647
No. Observations:                1943   AIC:                            -157.3
Df Residuals:                    1941   BIC:                            -146.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     11.1591      0.017    670.905      0.000      11.126      11.192
GrLivArea      0.0006   1.06e-05     54.646      0.000       0.001       0.001
==============================================================================
Omnibus:                       99.391   Durbin-Watson:                   1.127
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              178.802
Skew:                          -0.386   Prob(JB):                     1.49e-39
Kurtosis:                       4.270   Cond. No.                     4.96e+03
==============================================================================
```

FIGURE 9 – DATA AND FITTED REGRESSION LINE FOR LIVING AREA V LOG(SALES PRICE)

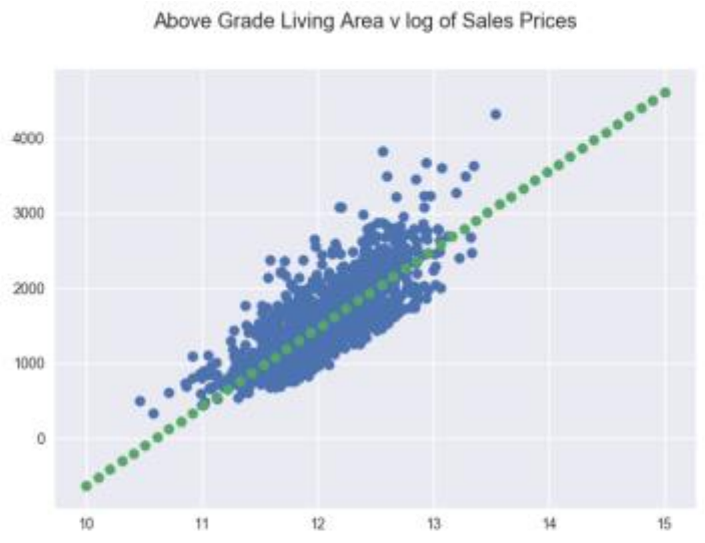
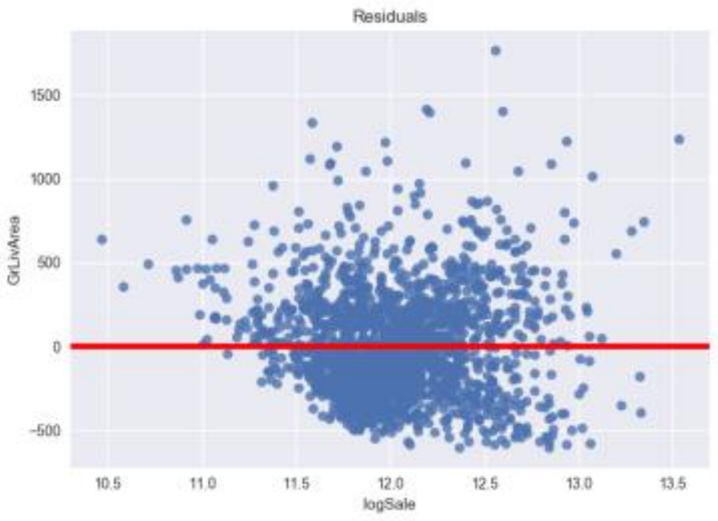


FIGURE 10 - RESIDUALS VERSUS FITTED VALUES



Model2: Total Basement Area Regression Model

Coefficient, Total Basement Area [696.47127192]
Intercept, Total Basement Area -7340.03493632

TABLE 5

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                logSale   R-squared:                       0.412
Model:                            OLS   Adj. R-squared:                  0.412
Method:                 Least Squares   F-statistic:                     1361.
Date:                Fri, 30 Jun 2017   Prob (F-statistic):          3.42e-226
Time:                        15:14:40   Log-Likelihood:                -308.26
No. Observations:                1943   AIC:                             620.5
Df Residuals:                    1941   BIC:                             631.7
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept     11.4103      0.018    642.179      0.000      11.375      11.445
TotalBsmtSF    0.0006    1.6e-05     36.889      0.000       0.001       0.001
==============================================================================
Omnibus:                       10.819   Durbin-Watson:                   1.199
Prob(Omnibus):                  0.004   Jarque-Bera (JB):               13.089
Skew:                           0.092   Prob(JB):                      0.00144
Kurtosis:                       3.358   Cond. No.                     3.06e+03
==============================================================================
```

FIGURE 11 – DATA AND FITTED REGRESSION LINE FOR SIZE OF BASEMENT V LOG(SALE PRICE)

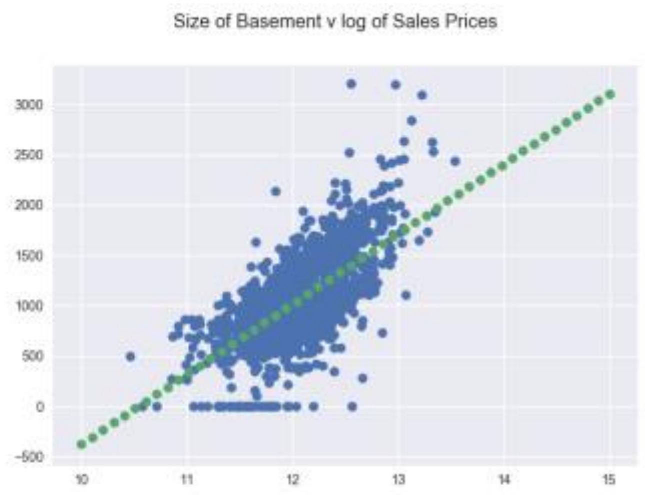
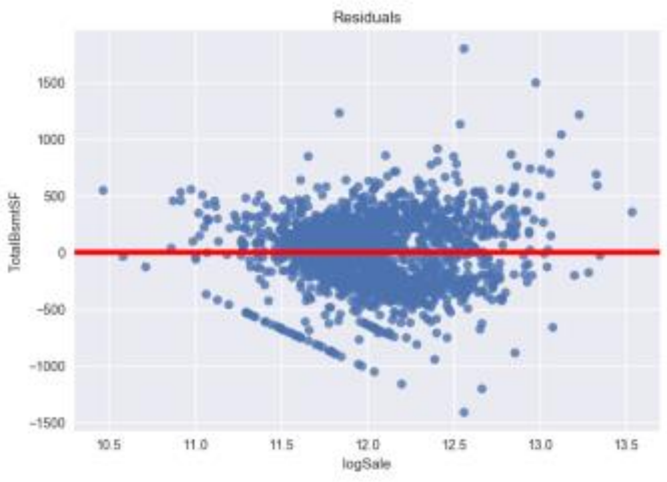
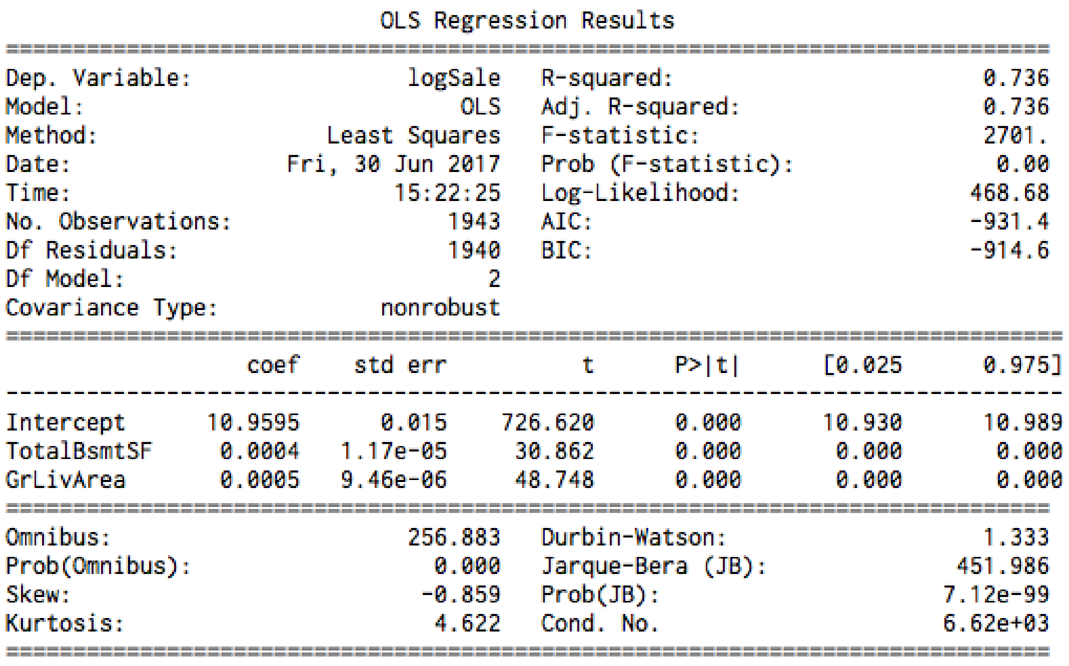


FIGURE 12 – RESIDUALS VERSUS FITTED VALUES



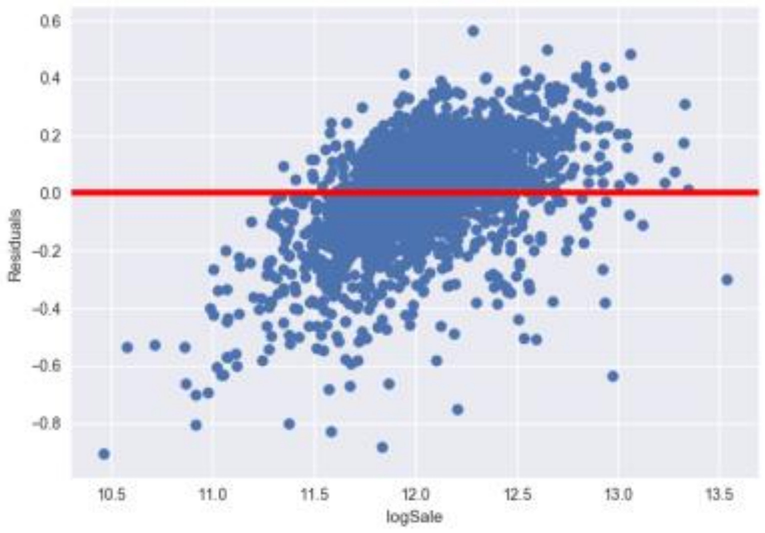
Model3: Multiple Linear Regression Model

TABLE 6



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                logSale   R-squared:                       0.736
Model:                            OLS   Adj. R-squared:                  0.736
Method:                 Least Squares   F-statistic:                     2701.
Date:                Fri, 30 Jun 2017   Prob (F-statistic):               0.00
Time:                        15:22:25   Log-Likelihood:                 468.68
No. Observations:                1943   AIC:                            -931.4
Df Residuals:                    1940   BIC:                            -914.6
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     10.9595      0.015    726.620      0.000      10.930      10.989
TotalBsmtSF    0.0004   1.17e-05     30.862      0.000       0.000       0.000
GrLivArea      0.0005   9.46e-06     48.748      0.000       0.000       0.000
==============================================================================
Omnibus:                      256.883   Durbin-Watson:                   1.333
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              451.986
Skew:                          -0.859   Prob(JB):                     7.12e-99
Kurtosis:                       4.622   Cond. No.                     6.62e+03
==============================================================================
```

FIGURE 13 - RESIDUALS VERSUS FITTED VALUES



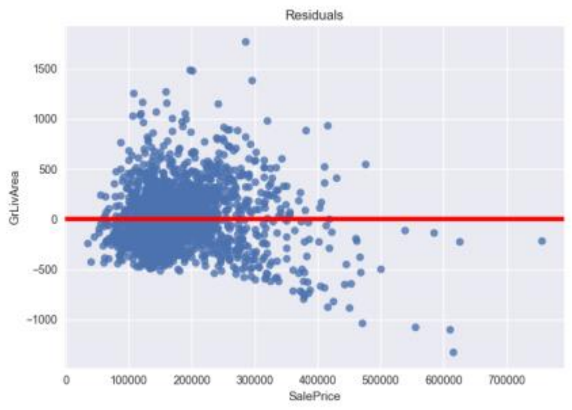
Comparing R^2 values for the models using both the regular and the transformed response variaible values, we can produce Table 7.

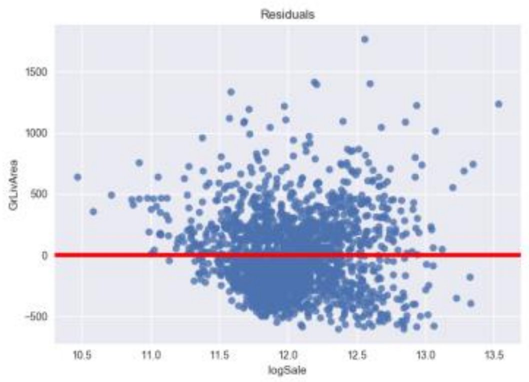
TABLE 7 – COMPARISON OF R^2 VALUES FOR THE DIFFERENT MODELS AND RESPONSE VARIABLES

	SalePrice	*Log(SalePrice)*
GrLivArea	.600	.606
TotalBsmtArea	.427	.412
Multiple Regression	.740	.736

There are small changes in the R2 values for the regression models when the transformed response variable is used, but none of the changes are particularly compelling. In this specific example, the transformation did not show a big impact, but does seem to help center the residuals. We can see this visually by pairwise comparisons of the residual plots for each model.

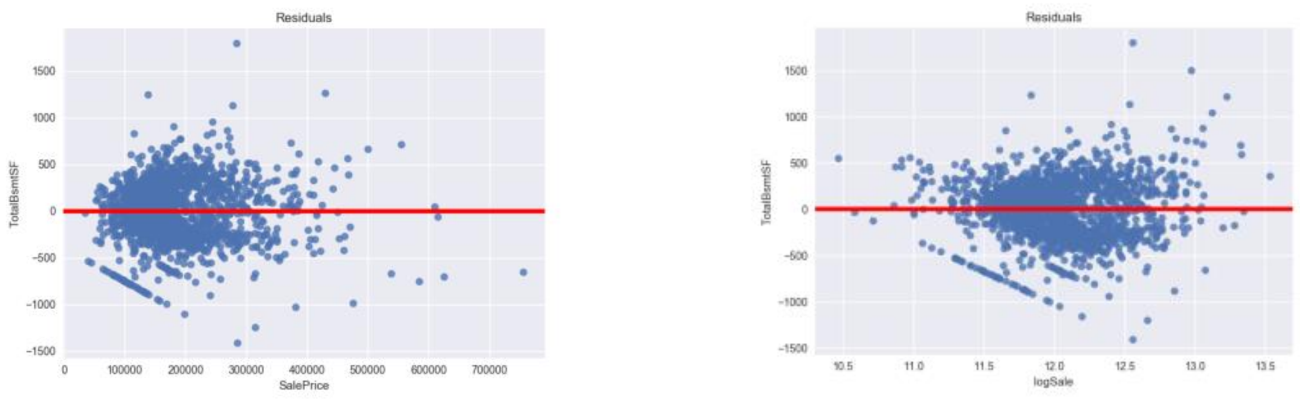
FIGURE 14 - COMPARISON OF RESIDUALS VERSUS FITTED VALUES FOR PREDICTOR LIVING AREA





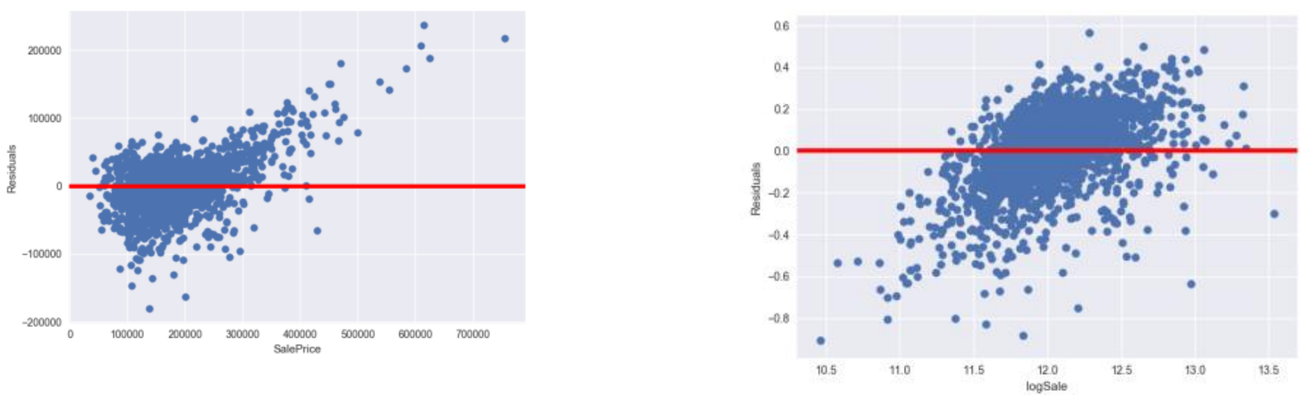
The residuals are better centered left-to-right for the transformed response variable. However, the top-to-bottom variation shows a lot more range on the top.

FIGURE 15 - COMPARISON OF RESIDUALS VERSUS FITTED VALUES PREDICTOR SIZE OF BASEMENT



Here also, the data are better centered for the transformed response variable.

FIGURE 16 - COMPARISON OF RESIDUALS VERSUS FITTED VALUES FOR MULTIPLE PREDICTOR VARIABLES



Finally, the data are better centered left-to-right, and seem to be better centered top-to-bottom as well.

Conclusions

The Ames housing data can produce linear models from the above-grade-living-area and the total-size-of-the-basement variables which account for roughly 73% of the variation see in the sale price for single family homes. If you were a Real Estate investor, or even trying to sell a home in Ames, you would probably want to add additional predictor variables to get a higher percentage of the variation accounted for. It may also be the case that my selection of Basement Size as a predictor variable was misguided, and a different variable may have been more effective.

The living-area variable is strong predictor on its own, accounting for 60% of variation in sales price. Both the predictor variable used showed significance, and combined to give a better model than either alone. Transforming the response variable by taking the log of the values had the effect of centering the residuals, even though it did not impact the R^2 values (as a test of goodness-of-fit) much. Since a well-fitted model should have centered residuals without a detectable pattern, I'd keep the transformation as part of model building for this data set. Since the SalePrice itself is not a normal distribution, this transformation appears useful in addressing that non-normality when building the models.

Appendix

Code used in generating this report

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Wed Jun 28 09:53:17 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.linear_model import LinearRegression
import statsmodels.api as sm
from statsmodels.formula.api import ols

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, 10],
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

# check for abnormal sale prices, like 0, or negative
print "Maximum Sales Price", max(df_houses.SalePrice)
print "Mimimum Sales Price", min(df_houses.SalePrice)
tmp = df_houses[(df_houses['SalePrice'] == 755000)]
print tmp

# Common sense suggests house prices will be most impacted by the size of the
# house, so let's restrict our view to continuous size related features
df_size = df_houses[['BsmtFinSF1', 'BsmtFinSF2','TotalBsmtSF','GrLivArea',
'FirstFlrSF','SecondFlrSF', 'GarageArea','PoolArea','WoodDeckSF',
'OpenPorchSF','EnclosedPorch','ThreeSsnPorch','ScreenPorch','MiscVal','SalePrice']]
df_size_1 = df_size[['BsmtFinSF1', 'BsmtFinSF2','TotalBsmtSF','GrLivArea',
'FirstFlrSF','SecondFlrSF', 'GarageArea','SalePrice']]
```

```
df_size_2 = df_size.iloc[:, 7:15]

ax = sns.pairplot(df_size_1)
plt.show()

ax = sns.pairplot(df_size_2)
plt.show()

df_final2 = df_houses[['TotalBsmtSF','GrLivArea','SalePrice']]
ax = sns.pairplot(df_final2)
plt.show()

# fit the regression line for above grade living area v saleprice and plot
# ---------------------------------------------------------------------------
x=df_final2['SalePrice']
y=df_final2['GrLivArea']
X = x[:, np.newaxis]
X.shape
model = LinearRegression(fit_intercept = True)
model.fit(X,y)
print "Coefficient, Above Grade Living Area", model.coef_
print "Intercept, Above Grade Living Area",model.intercept_
print ''

fig = plt.figure()
xfit=np.linspace(-1, 800000)
Xfit = xfit[:, np.newaxis]
yfit = model.predict(Xfit)
plt.scatter(x,y)
plt.scatter(xfit, yfit)
fig.suptitle('Above Grade Living Area v Sales Prices')
plt.show()

fig = plt.figure()
ax = sns.residplot(x='SalePrice',y = 'GrLivArea', data = df_final2)
ax.set(title='Residuals')
plt.axhline(linewidth=4, color='r')
plt.show()

my_lm = ols('SalePrice~ GrLivArea',data = df_final2)
results = my_lm.fit()
print results.summary()
print ''
```

```
# fit the regression line for Basement v saleprice and plot
# ----------------------------------------------------------------------------
x=df_final2['SalePrice']
y=df_final2['TotalBsmtSF']
X = x[:, np.newaxis]
X.shape
model = LinearRegression(fit_intercept = True)
model.fit(X,y)
print "Coefficient, Total Basement Area", model.coef_
print "Intercept, Total Basement Area", model.intercept_
print ''

fig = plt.figure()
xfit=np.linspace(-1, 800000)
Xfit = xfit[:, np.newaxis]
yfit = model.predict(Xfit)
plt.scatter(x,y)
plt.scatter(xfit, yfit)
fig.suptitle('Size of Basement v Sales Prices')
plt.show()

fig = plt.figure()
ax = sns.residplot(x='SalePrice',y = 'TotalBsmtSF', data = df_final2)
ax.set(title='Residuals')
plt.axhline(linewidth=4, color='r')
plt.show()

my_lm = ols('SalePrice~ TotalBsmtSF',data = df_final2).fit()
print my_lm.summary()
print ''



# multiple regression model
my_lm = ols(formula = 'SalePrice~ TotalBsmtSF+GrLivArea',data = df_final2).fit()
print my_lm.summary()

fig = plt.figure()
plt.scatter(df_final2['SalePrice'],my_lm.resid)
plt.xlabel('SalePrice')
plt.ylabel('Residuals')
plt.axhline(linewidth=4, color='r')
plt.show()

# ----------------------------------------------------------------------------
# Repeating above using log(SalePrice) in place of SalePrice
# good coding hygiene says this should probably be a function,
# since duplicating code is bad, but I'm still feeling wretched,
# so, copying and pasting and feeling badly about it
# --------------------------------------------------------------------------

# create log(SalePrice) column
df_log = df_houses[['TotalBsmtSF','GrLivArea','SalePrice']]
df_log['logSale'] = np.log(df_log.SalePrice)
df_log = df_log[['TotalBsmtSF','GrLivArea','logSale']]
ax = sns.pairplot(df_log)
plt.show()

# fit the regression line for above grade living area v logSale and plot
# --------------------------------------------------------------------------
x=df_log['logSale']
y=df_log['GrLivArea']
X = x[:, np.newaxis]
X.shape
model = LinearRegression(fit_intercept = True)
model.fit(X,y)
print "Coefficient, Above Grade Living Area", model.coef_
print "Intercept, Above Grade Living Area",model.intercept_
print ''

fig = plt.figure()
xfit=np.linspace(10,15)
Xfit = xfit[:, np.newaxis]
yfit = model.predict(Xfit)
plt.scatter(x,y)
plt.scatter(xfit, yfit)
fig.suptitle('Above Grade Living Area v log of Sales Prices')
plt.show()

fig = plt.figure()
ax = sns.residplot(x='logSale',y = 'GrLivArea', data = df_log)
ax.set(title='Residuals')
plt.axhline(linewidth=4, color='r')
plt.show()

my_lm = ols('logSale~ GrLivArea',data = df_log)
results = my_lm.fit()
print results.summary()
print ''

# fit the regression line for Basement v logSale and plot
# --------------------------------------------------------------------------
```

```
x=df_log['logSale']
y=df_log['TotalBsmtSF']
X = x[:, np.newaxis]
X.shape
model = LinearRegression(fit_intercept = True)
model.fit(X,y)
print "Coefficient, Total Basement Area", model.coef_
print "Intercept, Total Basement Area", model.intercept_
print ''

fig = plt.figure()
xfit=np.linspace(10, 15)
Xfit = xfit[:, np.newaxis]
yfit = model.predict(Xfit)
plt.scatter(x,y)
plt.scatter(xfit, yfit)
fig.suptitle('Size of Basement v log of Sales Prices')
plt.show()

fig = plt.figure()
ax = sns.residplot(x='logSale',y = 'TotalBsmtSF', data = df_log)
ax.set(title='Residuals')
plt.axhline(linewidth=4, color='r')
plt.show()

my_lm = ols('logSale~ TotalBsmtSF',data = df_log).fit()
print my_lm.summary()
print ''


# multiple regression model
my_lm = ols(formula = 'logSale~ TotalBsmtSF+GrLivArea',data = df_log).fit()
print my_lm.summary()

fig = plt.figure()
plt.scatter(df_log['logSale'],my_lm.resid)
plt.xlabel('logSale')
plt.ylabel('Residuals')
plt.axhline(linewidth=4, color='r')
plt.show()
```