Introduction

In this report, we are working with the Ames Housing data set, looking at the relationships between the various variables. Specifically, we are interested in the response variable SalePrice, which is what we are ultimately going to be building models to predict. The problem statement specifically notes our objective is "to be able to provide estimates of home values for 'typical' homes in Ames, Iowa.' To this end, some of the data will be dropped as not-relevant to the problem; details on this are given in the Data section of this document.

To explore the Ames Housing data, I used statistical summary data to look at standard statistical measures like mean, standard deviation, and quartiles. I also used graphical exploration techniques like scatterplot plot matrices, histograms and boxplots to visualize relationships. I also used ANOVA to look at the relationship between certain variables and sale price.

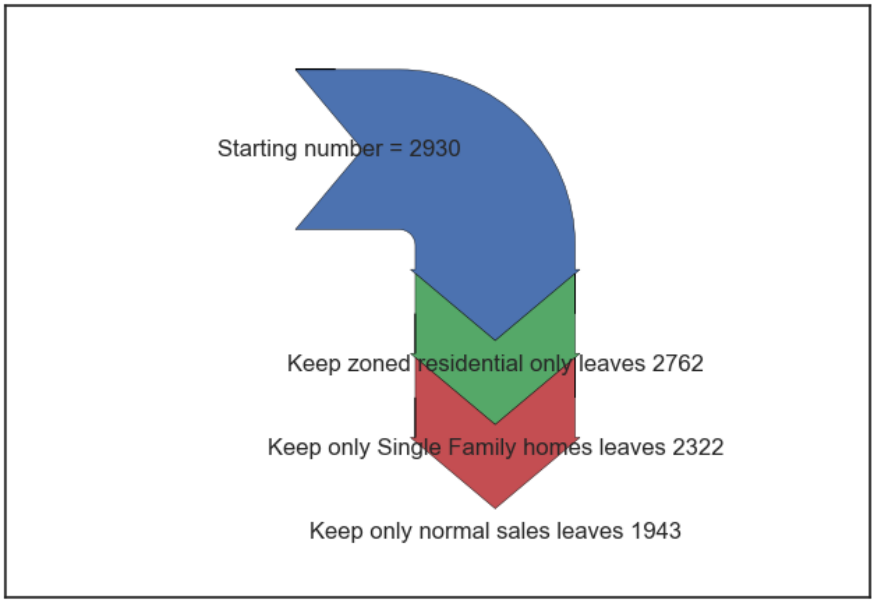
There is a Real Estate maxim: "location, location, location" however we will see that the Ames data do not have strong price differentiation between neighborhoods. Neighborhood turned out not to be a good predictor variable for SalePrice. I found it surprising, but the size of the basement did seem to be a good price indicator, along with above grade living area, and overall quality of the home.

Sample Definition

The data we are using in this report come from the Ames Iowa Assessor's Office. The information was used for computing the assessed value of properties which were sold during the period 2006 – 2010. The initial data set has 2930 observations of 82 variables. In order to fulfill the stated objective of predicting the value of a 'typical' home in Ames, the data have been reduced as shown in Figure 1.

I felt a home zoned commercial would not be typical. Single-family homes are inherently different from multi-family properties; so to have a consistent property type to work with, non-single family properties were also considered atypical. Finally, any property sold thru a non-"Normal" sale type is likely to have atypical sales prices due to the nature of the sale, an example being a foreclosure, so these properties were also considered atypical. Observation data for properties which are zoned residential, are single family homes, and were sold thru a normal transaction were retained. All other observations were dropped as non-relevant for the specific question at hand. The end result is 1943 observations of 82 variables.

Figure 1 - Waterfall of data restrictions



Data Quality Check

After reviewing the available variables, I selected the following twenty to work with, based on my best estimate of what is generally important in the real estate market. The target variable, SalePrice was also retained.

Table 1 – Selected 20 Variables

LotArea	**Heating**
Utilities	**FullBath**
Neighborhood	**HalfBath**
HouseStyle	**BedroomAbvGr**
BsmtFinSF2	**Fireplaces**
OverallQual	**GarageCars**
OverallCond	**PoolArea**
YearBuilt	**Fence**
YearRemodel	**GrLivArea**
TotalBsmtSF	**YrSold**

The “describe” function was used on the reduced, 20-variable data set to check for values which might constitute a red flag. Minimum and maximum values for the ordinal variables fall within a minimum of 0 and maximums that are plausible. It is worth noting that there is a SalePrice which appears to be an outlier, but further exploration shows it to be a property with very desirable traits in all categories; no obvious error in collection or transcription.

Tables 2 – Results of “describe” function

	LOTAREA	BSMTFINSF2	OVERALL QUAL	OVERALL COND	YEAR BUILT	YEAR REMODEL	TOTAL BSMTSF	FULLBATH
COUNT	1943	1943	1943	1943	1943	1943	1943	1943
MEAN	10848.20	59.24	5.98	5.76	1967.02	1982.65	1032.41	1.50
STD	7854.86	182.43	1.32	1.16	29.41	20.68	401.35	0.54
MIN	2500	0	1	1	1872	1950	0	0
25%	8166	0	5	5	1950	1962	801	1
50%	9759	0	6	5	1967	1991	973	1
75%	11851	0	7	7	1995	2002	1228	2
MAX	215245	1526	10	9	2010	2010	3206	3

Table 3 – Continuation of Results of “describe” function

	HALFBATH	BEDROOM ABVGR	FIREPLACES	GARAGE CARS	POOL AREA	GRLIV AREA	YRSOLD	SALEPRICE
COUNT	1943	1943	1943	1943	1943	1943	1943	1943
MEAN	0.38	2.92	0.64	1.73	2.21	1491.08	2007.87	178463.57
STD	0.49	0.71	0.66	0.72	34.95	498.12	1.29	73141.01
MIN	0	0	0	0	0	334	2006	35000
25%	0	3	0	1	0	1107	2007	130000
50%	0	3	1	2	0	1440	2008	160000
75%	1	3	1	2	0	1752.5	2009	208250
MAX	2	5	4	5	800	4316	2010	755000

The “info” function was used to check for missing values. Only the “Fence” column showed missing values. Since not all houses will have fenced yards, missing values in that variable do not constitute a quality issue.

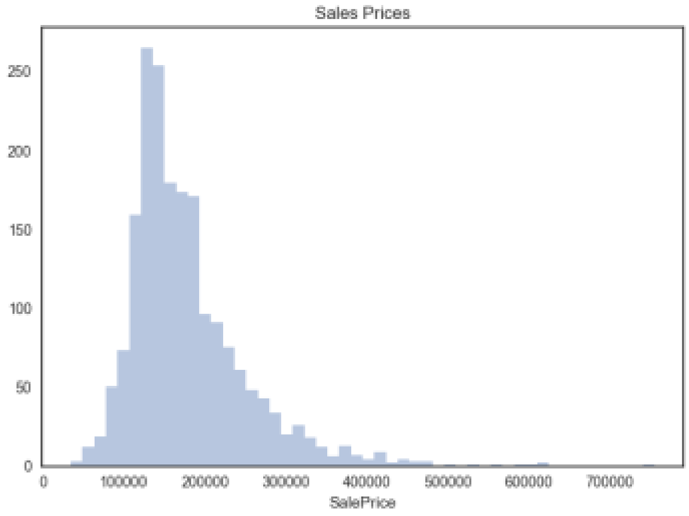
TABLE 4 – RESULTS OF FUNCTION INFO ON 20 SELECTED VARIABLES

LotArea	1943	non-null int64
Utilities	1943	non-null object
Neighborhood	1943	non-null object
HouseStyle	1943	non-null object
BsmtFinSF2	1943	non-null float64
OverallQual	1943	non-null int64
OverallCond	1943	non-null int64
YearBuilt	1943	non-null int64
YearRemodel	1943	non-null int64
TotalBsmtSF	1943	non-null float64
Heating	1943	non-null object
FullBath	1943	non-null int64
HalfBath	1943	non-null int64
BedroomAbvGr	1943	non-null int64
Fireplaces	1943	non-null int64
GarageCars	1943	non-null float64
PoolArea	1943	non-null int64
Fence	467	non-null object
GrLivArea	1943	non-null int64
YrSold	1943	non-null int64
SalePrice	1943	non-null int64

Initial Exploratory Data Analysis

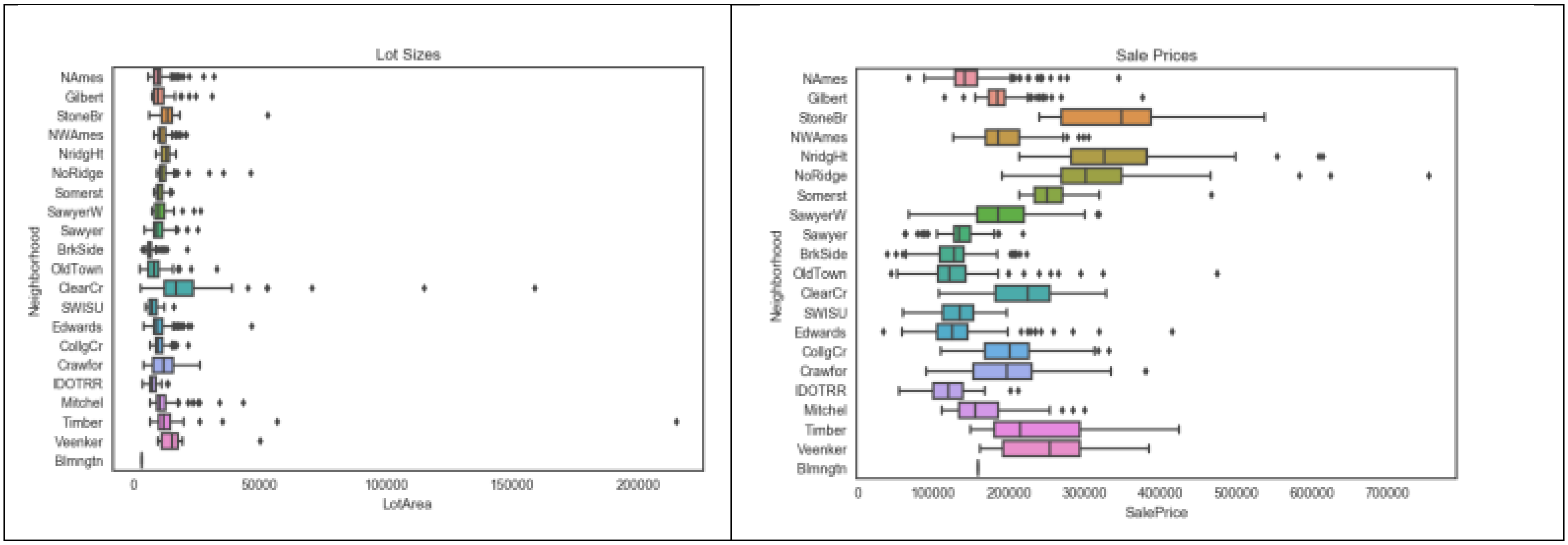
I began by looking at the target variable itself, SalePrice. As shown in Figure 2, SalePrice is skew (1.842) and has a long right tail. It is also leptokurtic with a kurtosis of 6.096.

Figure 2 – Distribution of Sales Prices



Real estate prices are often thought to be driven by neighborhood, and by the size of the house lot. A boxplot of those 2 variables against SalePrice, shown in Figure 3, revealed neither is going to be a great predictor. There is a lot of overlap between the size of lots in the different neighborhoods, and there is a great deal of commonality between prices in neighborhoods.

Figure 3 – Boxplots for Lot size and SalePrice by Neighborhood



Using scatterplot matrices, split into 2 groups of 10, I evaluated the 20 variables and reduced them to 10 based on the most promising looking results in the scatterplots. Variables with visible trends, that appeared to have possible correlations were selected. The large scatterplots are shown in Figures 10 and 11 in the Appendix.

Scatterplot matrices for the reduced 10 variables are shown in Figures 4 and 5.

FIGURE 4 – FIRST ½ OF 10 VARIABLE SCATTERPLOT MATRICES

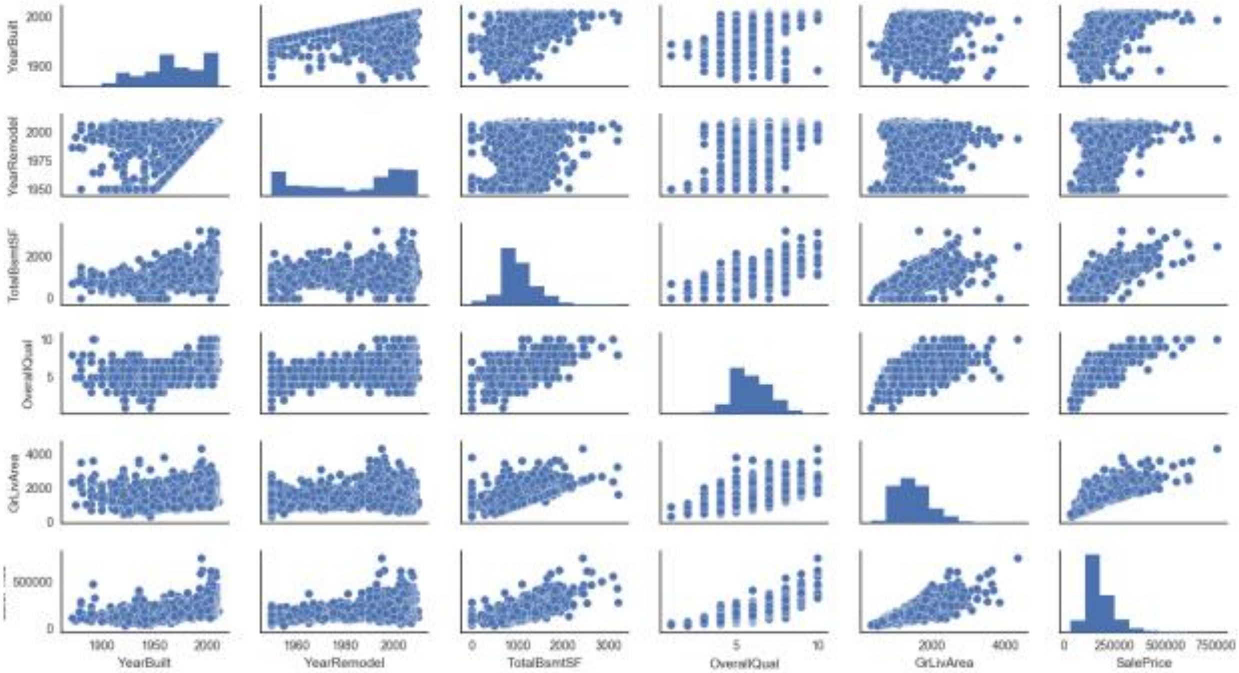
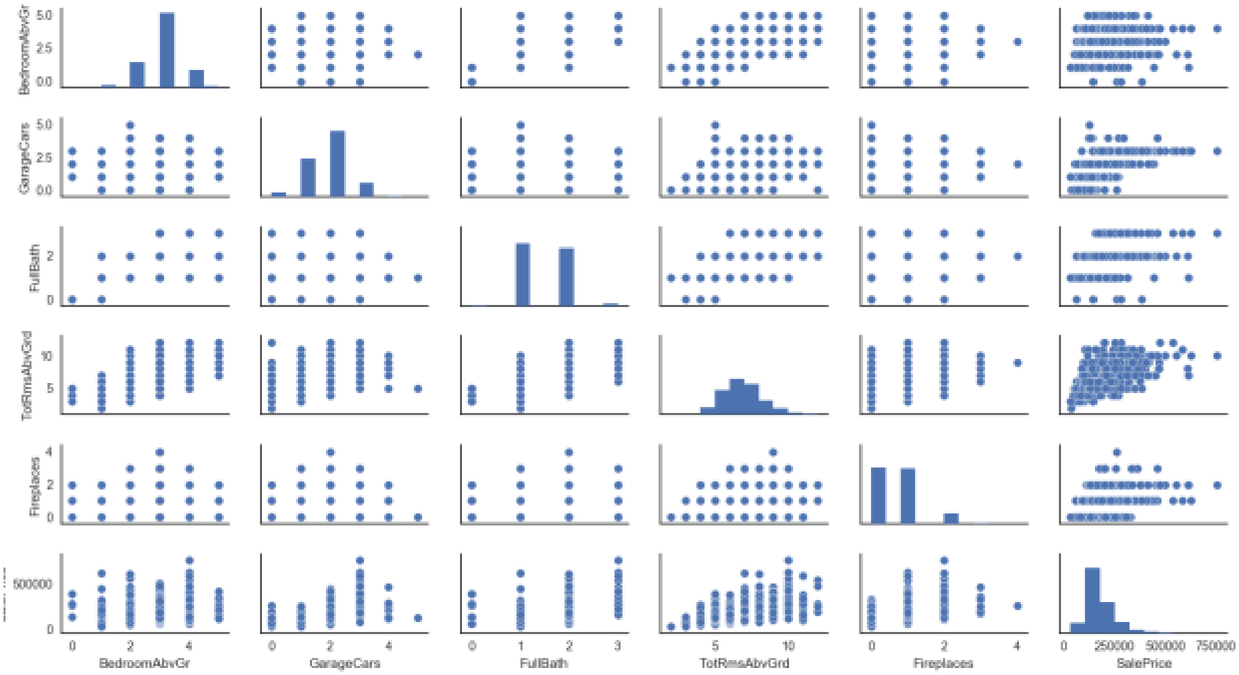


Figure 5 - Second ½ of 10 Variable scatterplot Matrices



An ANOVA computation shows all 10 variables are significant for SalePrice, shown in Table 5.

Table 5 – ANOVA of 10 selected variables

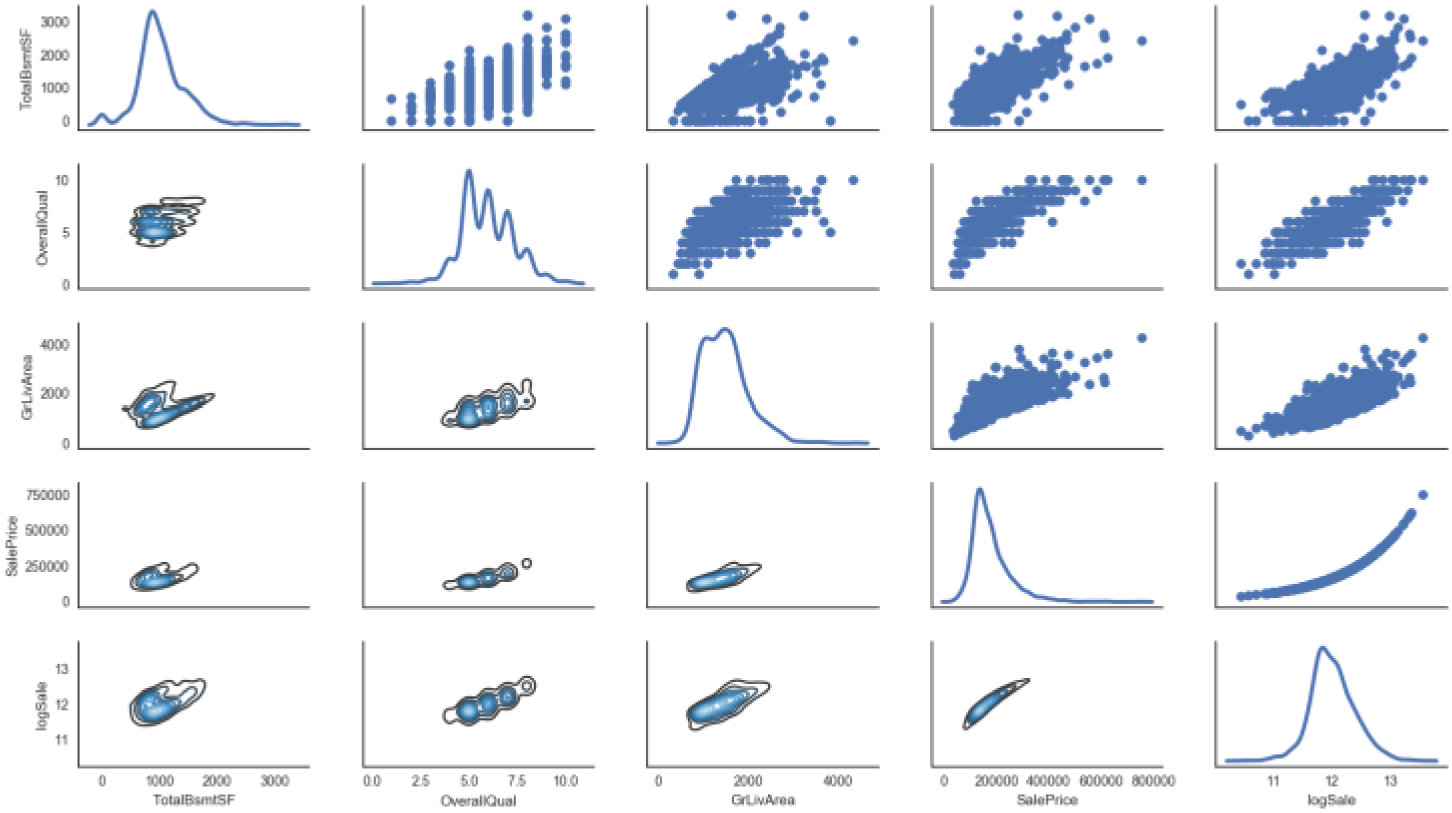
	sum_sq	df	F	PR(>F)
YearBuilt	8.61E+10	1	113.582884	8.25E-26
YearRemodel	3.89E+10	1	51.327365	1.11E-12
TotalBsmtSF	3.41E+11	1	450.328844	5.23E-90
OverallQual	2.44E+11	1	322.403117	8.59E-67
GrLivArea	5.29E+11	1	698.305811	1.27E-131
BedroomAbvGr	6.44E+10	1	84.939483	7.79E-20
GarageCars	4.94E+10	1	65.136562	1.22E-15
FullBath	1.56E+10	1	20.556077	6.15E-06
TotRmsAbvGrd	2.63E+08	1	0.346491	5.56E-01
Fireplaces	1.74E+10	1	22.956226	1.78E-06
Residual	1.46E+12	1932	NaN	NaN

From these results, I selected OverallQual, TotalBsmtSF, and GrLivArea as my final 3 variables.

Exploratory Data Analysis for Modeling

Figure 6 shows the PairGrid for the final 3 variables relative to SalePrice and log(SalePrice).

FIGURE 6 – PAIRGRID OF FINAL VARIABLES



A visual inspection of the pair grid reveals some promising looking clusters for log(SalePrice) against all three of the variables selected. The log(SalePrice) relationships seem to be more differentiated than SalePrice alone, so I opted to use the log values for fitting the regression lines. The graphical visualization of the fitted regression lines for these variables are shown in the figures below.

Figure 7 – Regression line for GrLivArea and log(SalePrice)

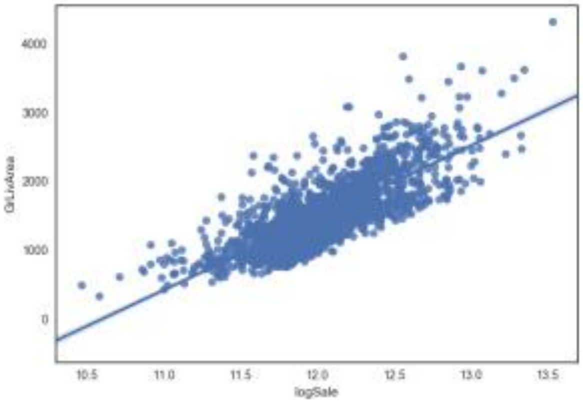


Figure 8 – Regression line for OverallQual and log(SalePrice)

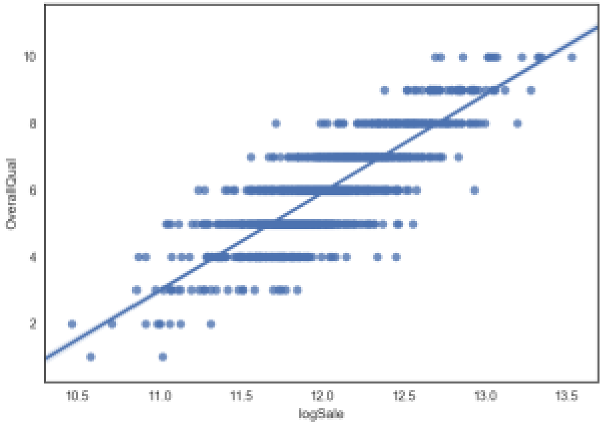
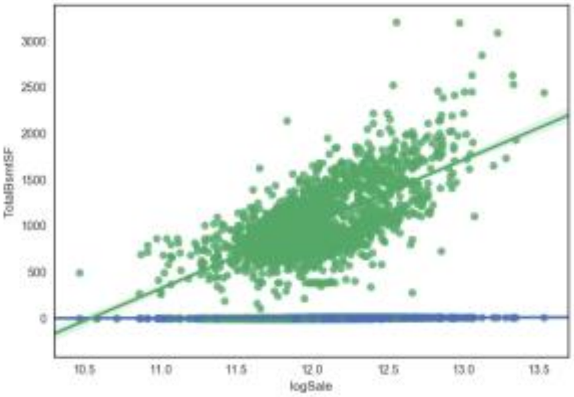


Figure 9 – Regression line for TotalBsmtSF and log(SalePrice)



Conclusion

The Ames real estate market does not have a lot of clear separation. There is no hard separation between prices in various neighborhoods for example. Lot sizes are fairly consistent across the market, age does not offer any sharp dividing lines for price. For all of the variables, there is overlap, which means model building will need to include multiple variables in order to be able to separate properties into distinct groups on which to predict price.

The at-grade living area variable looks like it has a slight heteroscedasticity, some transformation of that variable might be beneficial. For houses that have basements, there is a relationship between price and square feet, but there are a number of properties with no basement. I was not able to find another variable that was a better predictor than basement size; however, there will need to be work done on the final model to account for pricing homes without basements. Number of rooms at grade may offer value as a predictor if it were transformed, likewise year of remodel. Additional exploration, transforming those variables may yield good predictor results.

Appendix

FIGURE 10
FIRST ½ OF 20 VARIABLE SCATTERPLOT MATRICES

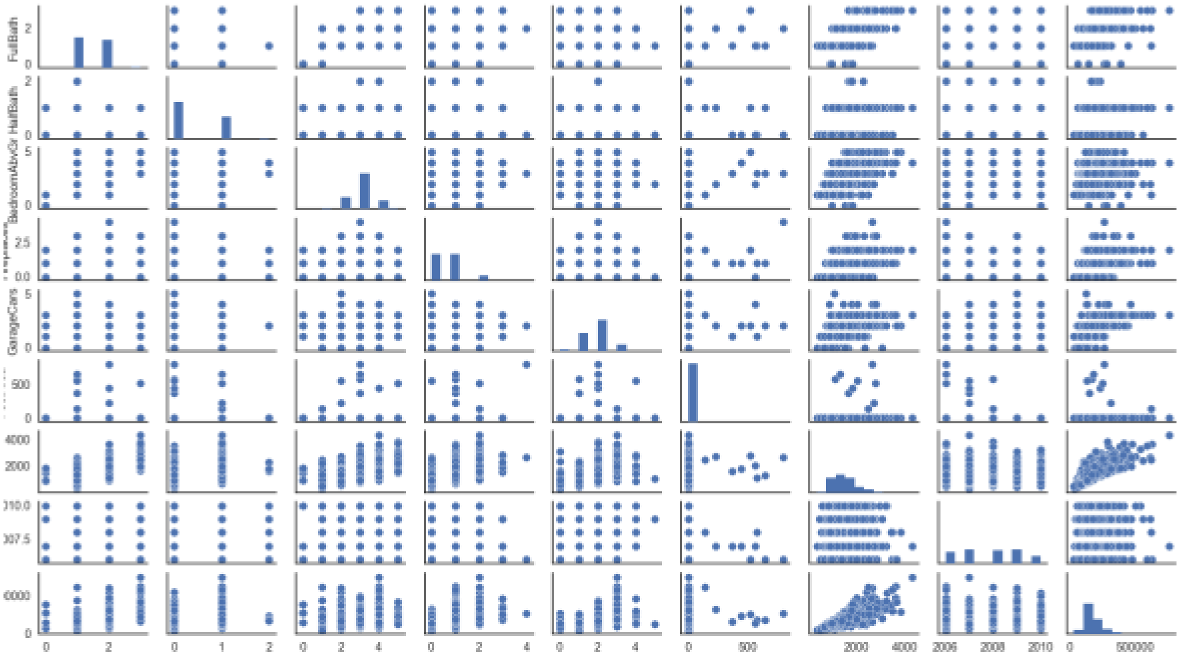
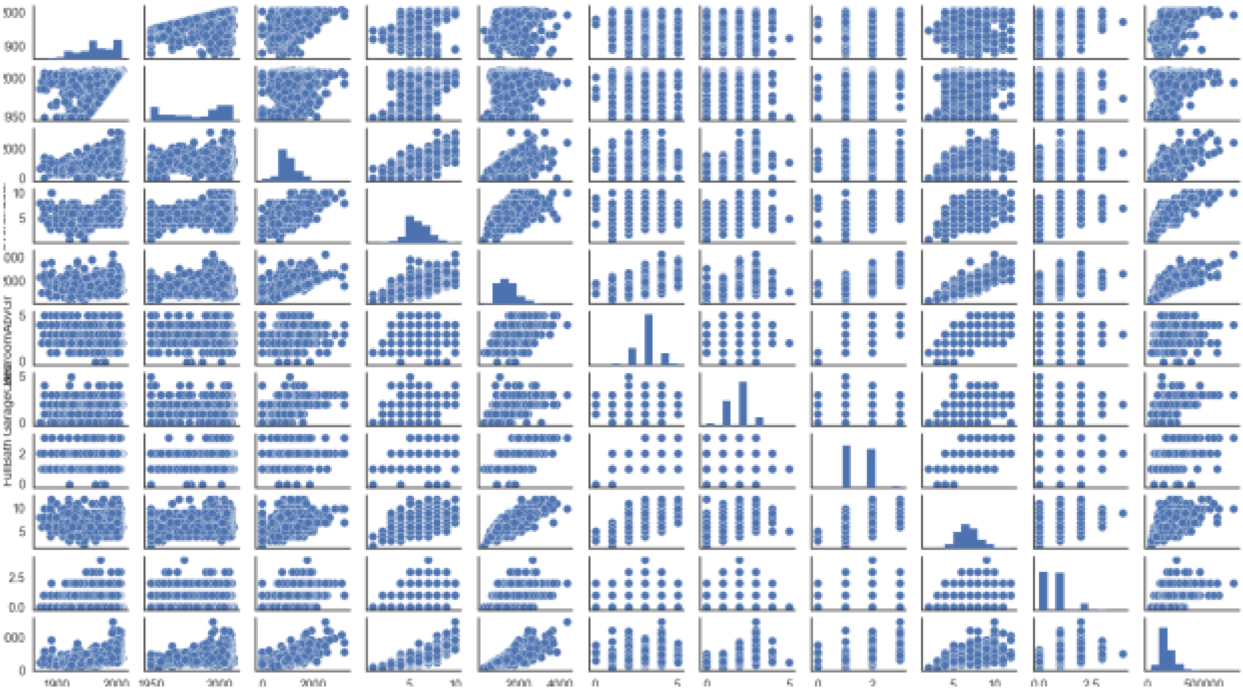


FIGURE 11
SECOND 1/2 OF 20 VARIABLE SCATTERPLOT MATRICES



Code used in generating this report

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Wed Jun 21 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
import statsmodels.api as sm
from statsmodels.formula.api import ols
import scipy.stats as sp

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 = "Keep zoned residential only leaves " + 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 = "Keep only Single Family homes leaves " + 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 = "Keep only normal sales leaves " + str(obs3)
drop3 = obs2 - obs3

# make a sankey chart showing the criteria and remaining number of observations
# for the data waterfall
sankey = Sankey(unit=None)
# first diagram, indexed by prior=0
sankey.add(flows=[1, -1],
orientations=[0,-1],
labels=[sankey0, sankey1])
# second diagram indexed by prior=1
sankey.add(flows=[1, -1],
orientations=[0,0],
labels=[' ', sankey2],
prior=0,
connect=(1, 0))
# second diagram indexed by prior=1
sankey.add(flows=[1, -1],
orientations=[0,0],
labels=[' ', sankey3],
prior=1,
connect=(1, 0))
sankey.finish()

new_df = norm_only
```

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


# pick 20 columns to Data Quality check
df_20 = new_df[['LotArea', 'Utilities','Neighborhood','HouseStyle','BsmtFinSF2',
'OverallQual','OverallCond','YearBuilt','YearRemodel','TotalBsmtSF',
'Heating','FullBath','HalfBath','BedroomAbvGr','Fireplaces','GarageCars',
'PoolArea', 'Fence', 'GrLivArea','YrSold','SalePrice']]
df_20.dtypes

# look at the one really big sale price
tmp = new_df[(new_df['SalePrice'] == 755000)]
print tmp

# Data quality check on the 20
# first, are the types as expected
df_20.dtypes

# let's look at the summary data for the 20
print df_20.describe()
# looking at the .info(), to see if there are missing values
print df_20.info()


# look at a boxplot of SalePrice
plt.figure()
ax = sns.boxplot(x="SalePrice", orient = 'v', data=new_df)
plt.show()

# and look at the distribution of SalePrice
plt.figure
ax=sns.distplot(new_df['SalePrice'], kde=False)
ax.set(title='Sales Prices')
plt.show()
print 'Skew = ', sp.skew(new_df['SalePrice'])
print 'Kurtosis = ', sp.kurtosis(new_df['SalePrice'])

# pick 10 features, try looking at a pairs plot for inspiration, split df into 2
# for readability of output
df_20_1 = new_df[['LotArea', 'Utilities','TotRmsAbvGrd','HouseStyle','BsmtFinSF2',
'OverallQual','OverallCond','YearBuilt','YearRemodel','TotalBsmtSF','SalePrice']]
#plt.figure()
```

```
ax = sns.pairplot(df_20_1)
plt.show()

df_20_2 = new_df[['Heating','FullBath','HalfBath','BedroomAbvGr','Fireplaces','GarageCars',
'PoolArea', 'Fence', 'GrLivArea','YrSold','SalePrice']]
#plt.figure()
ax = sns.pairplot(df_20_2)
plt.show()

# do some plotting for lot size and sales price, these value wide differences
# between min and max
plt.figure()
ax = sns.boxplot(x="LotArea", y="Neighborhood", orient = 'h', data=df_20)
ax.set_title('Lot Sizes')
plt.show()



plt.figure()
ax = sns.boxplot(x="SalePrice", y="Neighborhood", orient = 'h', data=df_20)
ax.set_title('Sale Prices')
plt.show()

# based on posible correlations in pairplot, pull out likely columns
df_10 = new_df[['YearBuilt','YearRemodel','TotalBsmtSF','OverallQual','GrLivArea',
'BedroomAbvGr','GarageCars','FullBath','TotRmsAbvGrd','Fireplaces', 'SalePrice']]
print df_10.describe()
print df_10.info()

df_10_1 = new_df[['YearBuilt','YearRemodel','TotalBsmtSF','OverallQual','GrLivArea',
'SalePrice']]
df_10_2 = new_df[['BedroomAbvGr','GarageCars','FullBath','TotRmsAbvGrd',
'Fireplaces', 'SalePrice']]
#plt.figure()
ax = sns.pairplot(df_10_1)
plt.show()

#plt.figure()
ax = sns.pairplot(df_10_2)
plt.show()

# OK, nenver used Python for statistics, so based on what I can find, this seems
# to be the closest anova function to what I'm used to in R
my_lm = ols('SalePrice~ YearBuilt+
YearRemodel+TotalBsmtSF+OverallQual+GrLivArea+BedroomAbvGr+GarageCars+FullBath+
TotRmsAbvGrd+Fireplaces',
data = new_df).fit()
```

```
print(sm.stats.anova_lm(my_lm, typ=2))


df_3 = new_df[['TotalBsmtSF','OverallQual','GrLivArea', 'SalePrice']]
#plt.figure()
ax = sns.pairplot(df_3)
plt.show()

# compute log od SalePrice and add to dataframe
df_3['logSale'] = np.log(df_3.SalePrice)

#plt.figure()
ax = sns.pairplot(df_3)
plt.show()

# chart fitted regression lines for the 3 variables selected
#plt.figure()
ax = sns.lmplot( x='SalePrice', y='TotalBsmtSF', data = df_3)
plt.show()
ax = sns.lmplot( x='SalePrice', y='OverallQual', data = df_3)
plt.show()
ax = sns.lmplot( x='SalePrice', y='GrLivArea', data = df_3)
plt.show()



# look at the Pair grid for SalePrice and log(SalePrice)
sns.set(style="white")
ax = sns.PairGrid(df_3, diag_sharey=False)
ax.map_lower(sns.kdeplot, cmap="Blues_d")
ax.map_upper(plt.scatter)
ax.map_diag(sns.kdeplot, lw=3)

ax=sns.regplot(x='logSale', y='OverallQual', data=df_3)
plt.show()
ax=sns.regplot(x='logSale', y='GrLivArea', data=df_3)
plt.show()
ax=sns.regplot(x='logSale', y='TotalBsmtSF', data=df_3)
plt.show()

plt.close()
```