Ames Housing

Andy Heroy, Kito Patterson, Ryan Quincy Paul

February 13, 2019

Introduction:

For our project, we will estimate how the sale prices of homes in Ames Iowa are affected by various real estate attributes. First, we will utilize multiple linear regression to predict the sale price on many explanatory variables. Then a two-way ANOVA will be run on sale price and two categorical variables to analyze group variability.

Data Description:

As stated above, we will use the Ames housing training dataset for our analysis. This dataset was created by Dean De Cock as a modern alternative to the outdated Boston housing dataset (detail can be found here at the Kaggle website). The dataset contains 1460 observations and 79 explanatory variables. For our analysis we will be using some of those 79 for analysis as well as a few of our own created features in the prediction.

Data Preparation

First off, we need to go through and do a little house cleaning with the data. After combining [11] the test and train datasets, we decided to create a smaller subset in hopes of targeting homes that represent typical home sale transactions in Ames, IA. As a result, we removed all commercial or severely damaged properties while keeping homes sold under "normal" conditions only.

We further reduced the dataset [12] by excluding rows with missing values rather than imputing the missing values. Lot Frontage NA values were converted to 0 and the GarageYrBlt values were converted to inherit the YearBuilt value if the property did indeed have a garage. Lastly, all missing numeric values were transformed to 0 and all missing character values were changed to "None".

The final step in our data scrubbing process was to convert all character columns to factor columns [12]. Now that our final dataset is completely clean and uniform, we can confidently move on to the exploratory analysis process.

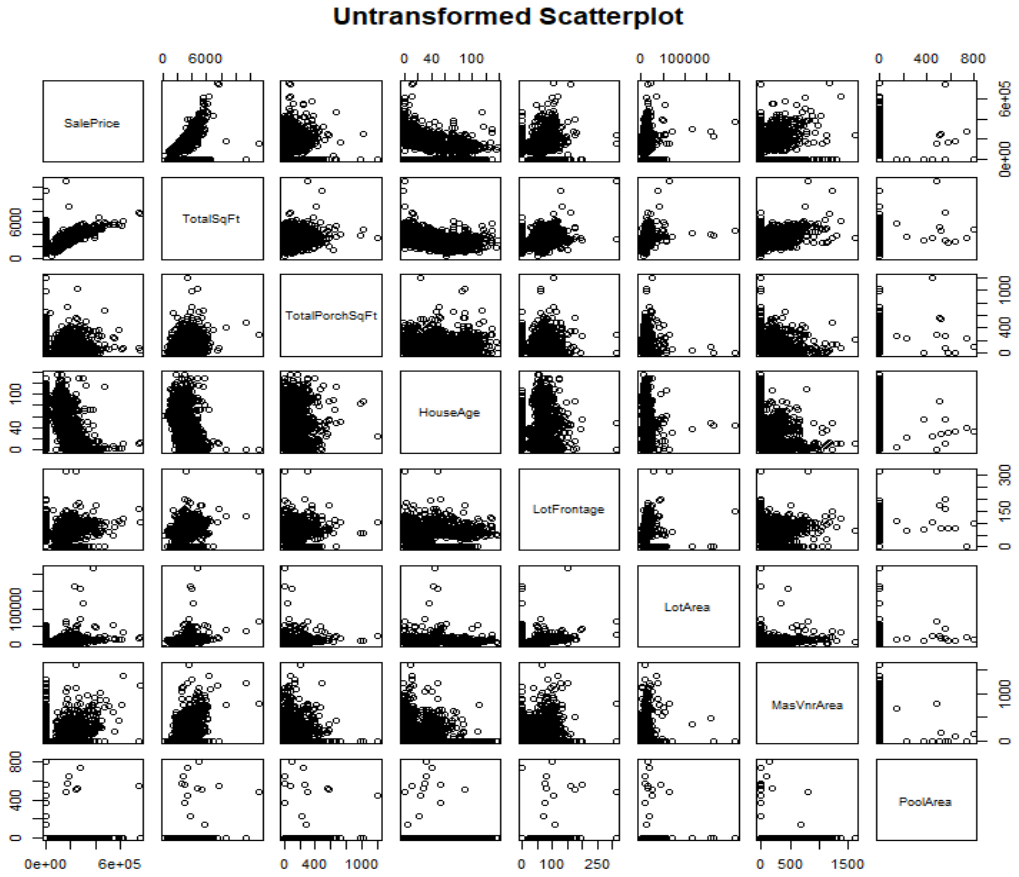
Exploratory Data Analysis

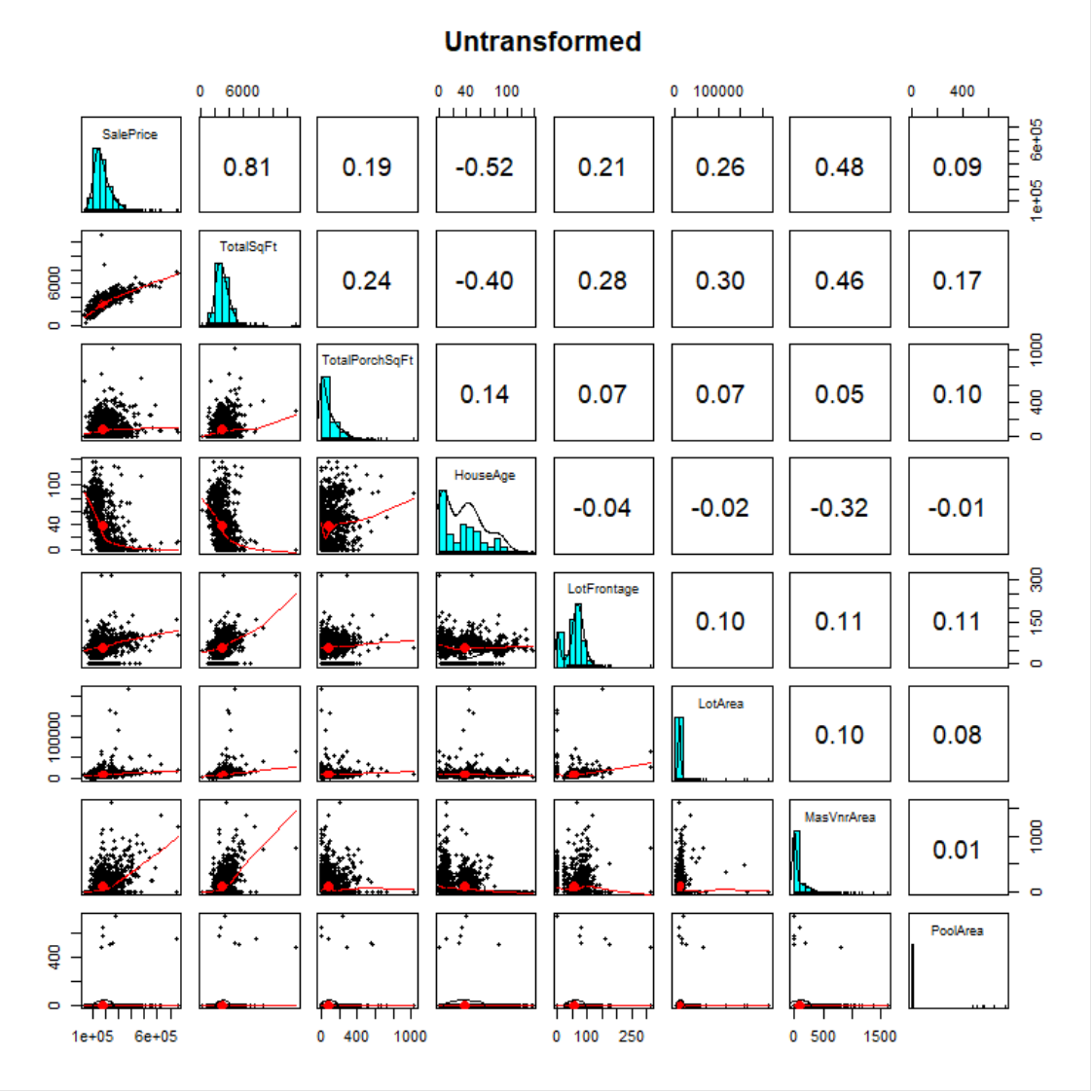
This portion of the analysis, we decided to aggregate some of the variables in order to better represent some of the housing attributes we thought should be examined. We chose:

- TotalSqFt to represent total indoor area of the house
- TotalPorchSqFt to represent total porch space.
- TotalBaths to represent the total bathroom count
- HouseAge to represent house age.

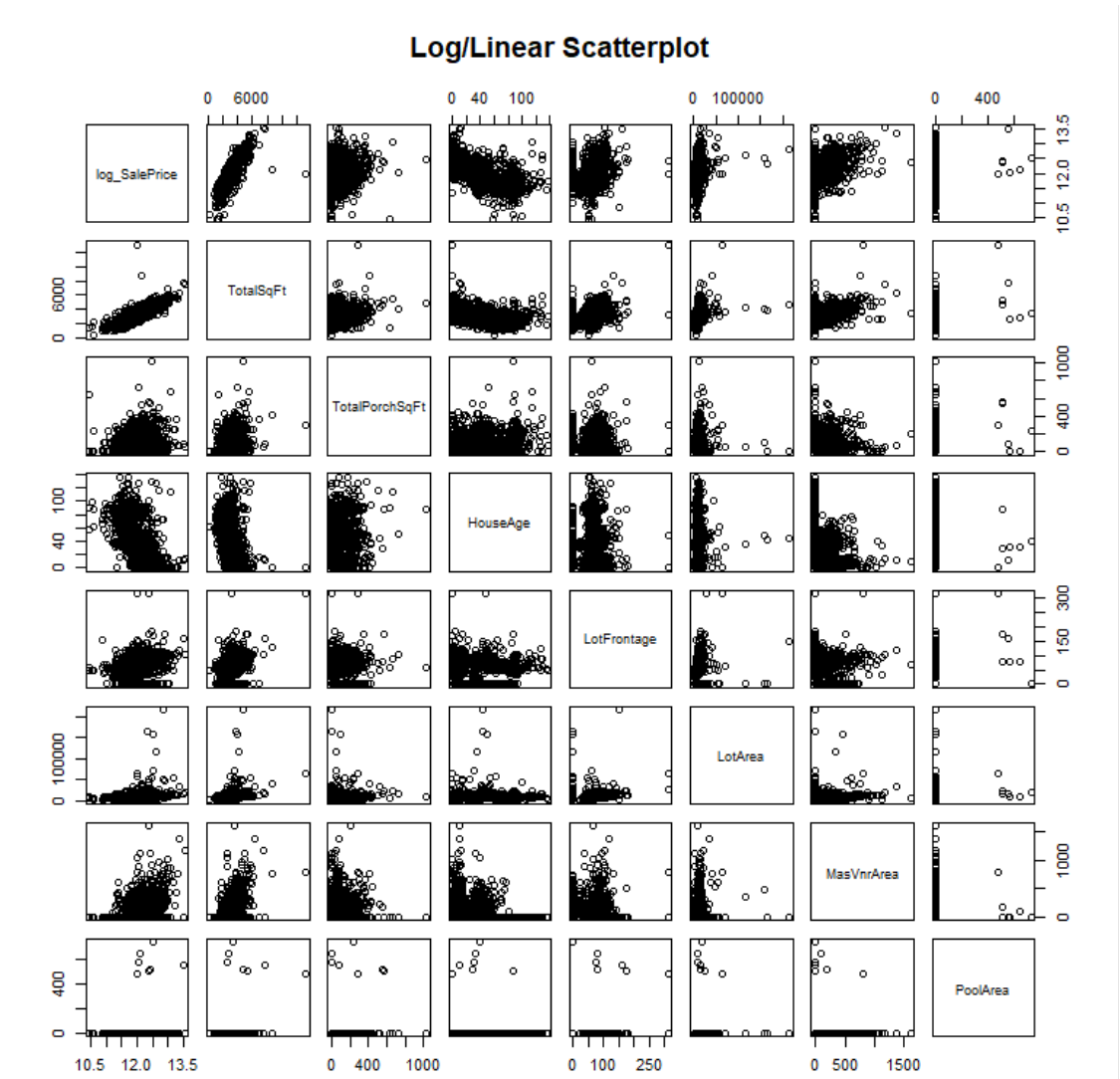
```r
#Added Features for Analysis
df3$TotalSqFt <- (df3$GrLivArea + df3$TotalBsmtSF + df3$GarageArea)
df3$TotalPorchSqFt <- (df3$OpenPorchSF+df3$EnclosedPorch+df3$ScreenPorch)
df3$TotalBaths <-
df3$BsmtFullBath+(df3$BsmtHalfBath*0.5)+df3$FullBath+(df3$HalfBath*0.5)
df3$HouseAge <- as.numeric(df3$YrSold) - as.numeric(df3$YearBuilt)
```

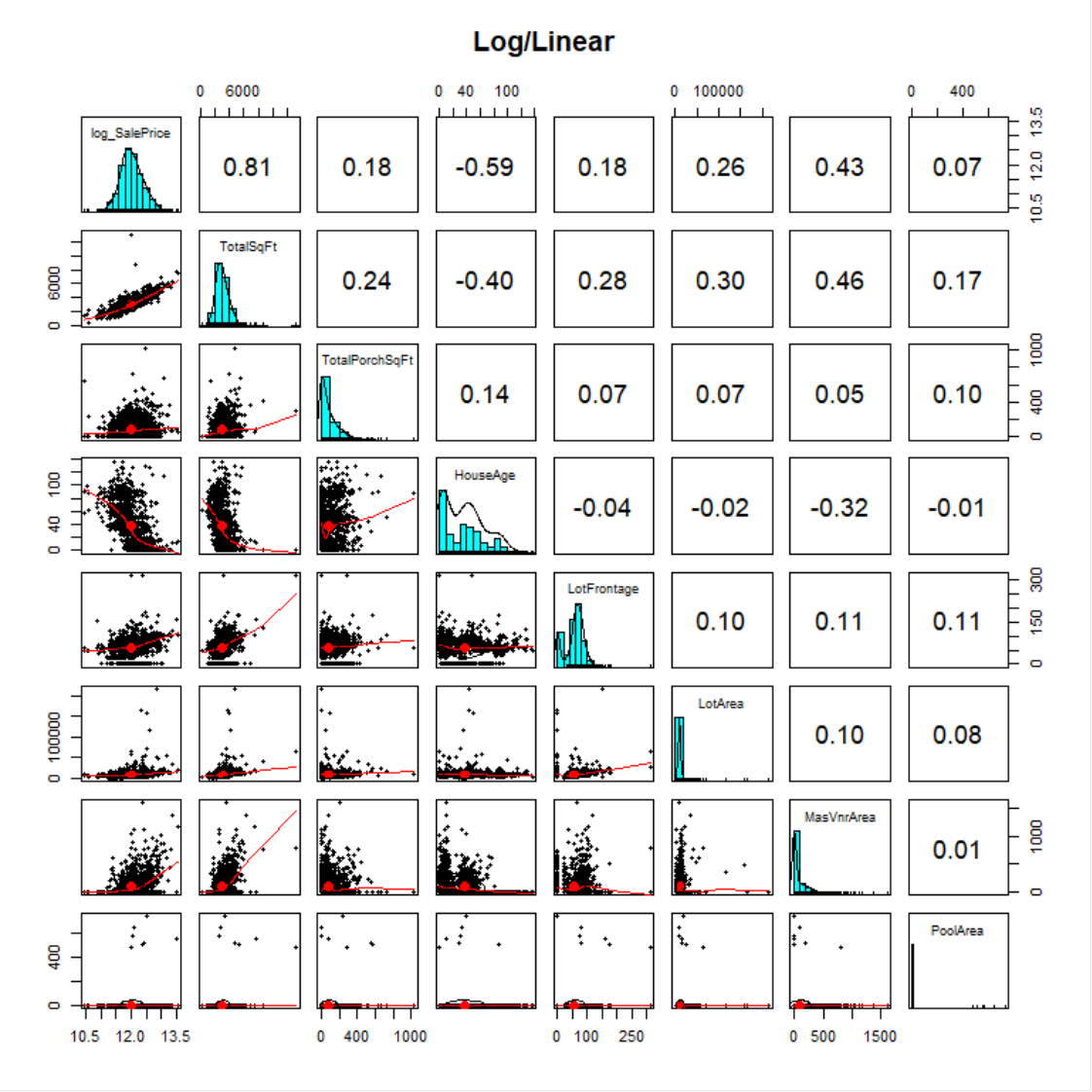
Now we'll generate an initial scatterplot matrix to view the untransformed numerical data and see if we find any correlation. We picked these variables in order to look at our feature creation and seeing how square footage, in all its forms, plays into the Sale Price.





While there does appear to be some correlation between the area and age-related variables, the data doesn't seem to be normally distributed. As such, we're going to try a log transformation to see if normality can be improved. First, we'll try a log-linear approach and only log the sale price.


Log/Linear Scatterplot
log_SalePrice
TotalSqFt
TotalPorchSqFt
HouseAge
LotFrontage
LotArea
MasVnrArea
PoolArea



While we see an improvement in the Log_SalePrice distribution, we still see evidence of non-linearity and skew in some of the other features. To address that, we will try a log-log transformation on the remaining variables and test for improvement. See the scatterplot [13] and residuals [14] chart in the appendix. While some distributions were improved with the transformation, many of the scatterplots still exhibited clustering and non-linear trends. As such, we will stay with a Log-linear transformation.

Model Selection

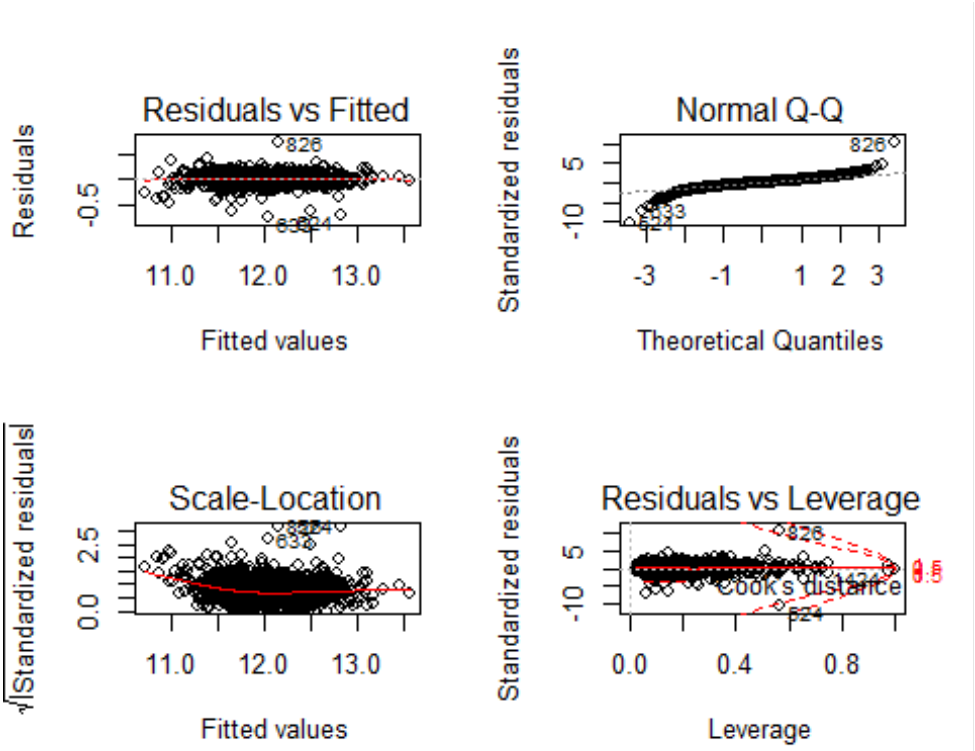
Now that we've explored a bit more of the data, we're going to look at a basic random tree analysis to see what variables *should* show up in our feature selection when determining a linear model. According to the results, [17] we would expect to see total indoor square footage and neighborhood indicators in our model. We additionally ran a correlation heatmap to get a better understanding of multicollinearity. There does seem to be some redundancy between explanatory variables; as a result, we noted them as possible candidates for removal in subsequent modeling efforts. See appendix [15].

Objective 1 - Multiple Linear Regression

NOTE: Due to differences in randomization results when splitting the training and test sets on different machines, despite keeping the seed constant, the analysis below will not match the results of the code. Code can be found in Appendix [18].

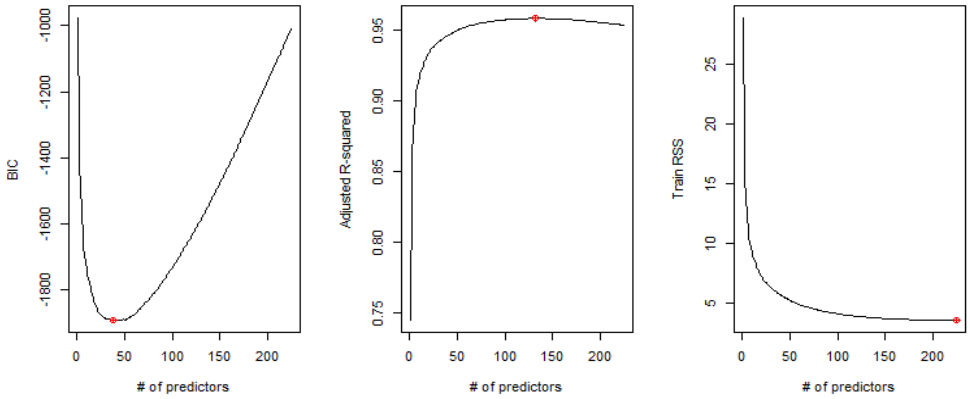
We will use forward selection to determine the most effective explanatory variables that can be used to predict the sale price of homes in Ames, Iowa. First, we will split the `training.csv` given in the Kaggle data set into another training and test set to determine the effectiveness of our models as the `test.csv` in the Kaggle set does not provide sale prices. We will also be dropping factors that have only one level, logged variables that we decided above not to use, and Id as it is a known arbitrary field.

A quick check of the assumptions by plotting the full model yields some mixed results. There are some outliers according to the Residuals versus Fitted and Cook's D plots. We believe the analysis can continue with them as they are legitimate houses with high prices. The residuals are relatively normal shaped according to the QQ plot and the samples are assumed to be independent.

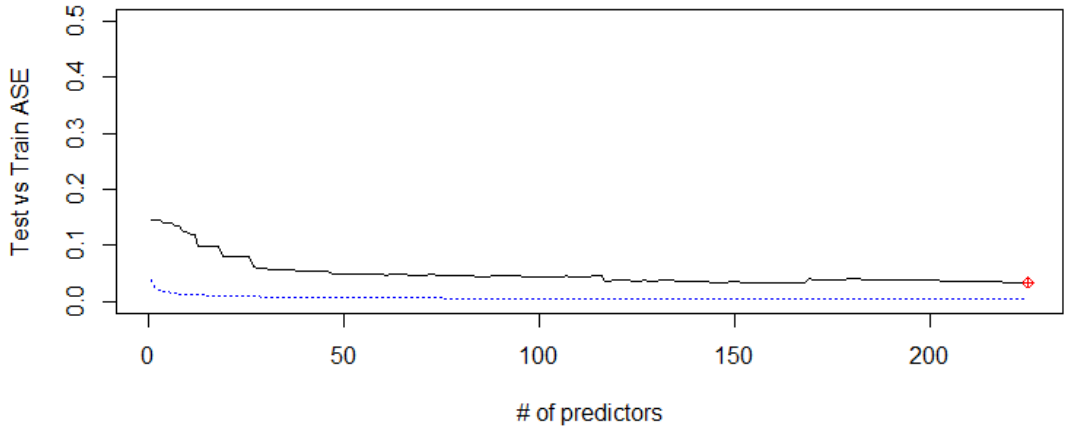


Below you will see a few measures used to determine the effectiveness of the model at each step of the forward selection. First is the Bayesian Information Criterion (BIC). A lower value means the predicted values are closer to the actual results. As can be seen at about 40 predictors, the BIC also has a penalty that goes up as the number of predictors goes up. Next is the adjusted R-squared, a measure of how much of the variability of the response is explained by the model. This peaks at about 100 predictors. Finally, we have the residual sum of squares (RSS), the sum of the squared differences of the predictions from the actual results without a penalty like the BIC. It's lowest when

all the predictors are added, but it levels out at around 50 predictors.



We will also look at the average squared error (ASE) of the predictions versus the actual values of each step in the feature selection. As can be seen below, and like the RSS above, it continues to decrease until all the variables are added. However, there is a final significant dip at about 60 predictors.



NOTE: Reminder, this analysis does not match the plot due to randomization differences across machines despite keeping the seed constant.)

After more drilling, we've determined that the last significant drop in ASE is found at step 58 where the ASE drops from .04 to .0226. Looking at the variables included at this step, we see some categorical factors included, but not all the levels of those factors. From here we will create a linear model with every continuous variable or categorical variable that is represented by at least 1 level. The summary of the fit is too long to output in this paper. Below are some selected regression coefficients and their interpretation. As described earlier, we use a log-linear transformation which means we logged the sale price but none of the explanatory variables. Besides the intercept, in the interpretations for the slope coefficients assume all other variables are held constant.

Variable	Coefficient	p-value	Interpretation
Intercept	11.04	< 2e-16	With all variables at 0, the sale price of the house would

			be $e^{11.04}$ = $62317.65
Total Indoor Area (sqft)	2.167e-4	< 2e-16	A 100 sqft increase would yield a .22% increase in sale price
Unfinished Garage	-.0697	5.16e-4	An unfinished garage decreases the sale price by -6.9%
Lot Area (sqft)	1.75e-6	.0163	Every acre (43,560 sqft) yields a 7.62% increase in sale price
Crawford Neighborhood	.1713	.00604	The sale price of a home in the Crawford neighborhood is increased 17.13%
MeadowV Neighborhood	-.32	1.06e-4	The sale price of a home in the MeadowV neighborhood is decreased 32%
Stucco Exterior	.258	8.94e-4	A stucco exterior increases the sale price 25.8%
Asphalt Shingle Exterior	.2263	.138	An asphalt shingle exterior covering increases the sale price 22.6%

The starting price with all variables at zero is highly significant as well as the increase of price per square foot of indoor space. We also have examples of highly significant and high practical impact categorical levels (houses found in the Crawford and MeadowV neighborhoods) as well as a non-significant but high impact level (asphalt shingle exterior covering). There are also many variables thrown into the model that are non-significant (p-value > .05). Below I will fit one more "hybrid" model with all non-significant continuous variables and categorical variables without a significant level removed from the forward selection model. It seems that the total indoor square footage was the last highly significant variable added at step 58 and many non-significant factors were added on the way. It's hard to determine the order, as the `regsubset` method will shuffle the variables if it detects high collinearity.

```
hybridModel <-
lm(log(SalePrice)~LotArea+LotConfig+Neighborhood+Condition1+RoofMatl+Exterior
1st+ExterCond+BsmtExposure+BsmtFinType1+BsmtFinType2+Heating+CentralAir+Funct
ional+GarageType+GarageFinish+SaleType+TotalSqFt, data = mlrHousesTrain)
```

The ASE of the new model (compared with .0226 above):

```
## [1] 0.02982036
```

This is better than the ASE from the forward model selection (**NOTE:** as already mentioned, the output does not match the analysis due to randomization differences) and significantly reduces the number of variables in the fit, thereby reducing the risk of overfitting. Below are the first five predictions along with the confidence intervals. Since we logged the sale prices, we transform them back to meaningful numbers by raising them as a power of e.

```
head(exp(hybridModelPredictions))

##         fit      lwr      upr
## 1  191241.1 184809.7 197896.4
## 2  186438.9 166509.0 208754.4
## 7  268893.1 254549.3 284045.2
## 8  211972.8 185070.0 242786.3
```

```
## 10 131239.5 118969.6 144775.0
## 11 128030.7 120516.3 136013.6
```

Conclusion

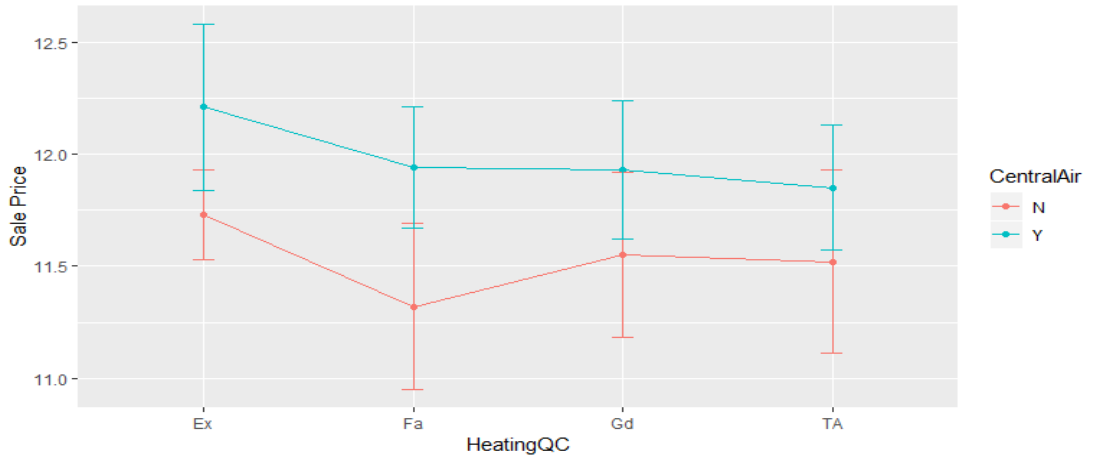
We used forward stepwise model selection with some manual adjustments to come up with a linear model to predict sale prices of homes in Ames, Iowa. As far as new insights, we have learned about the simplicity of feature selection. Non-significant factors are not pruned on the way to the most accurate model. Manual pruning needs to be done after the feature selection has completed to reduce needless complexity. We also learned that set.seed() does not promise that randomizations will be held constant across machines. From a reporting perspective, next time we can write our findings in a way that reacts to the output of the code instead of assuming the output of a seed will remain the same.

Objective 2 - A Two-way ANOVA

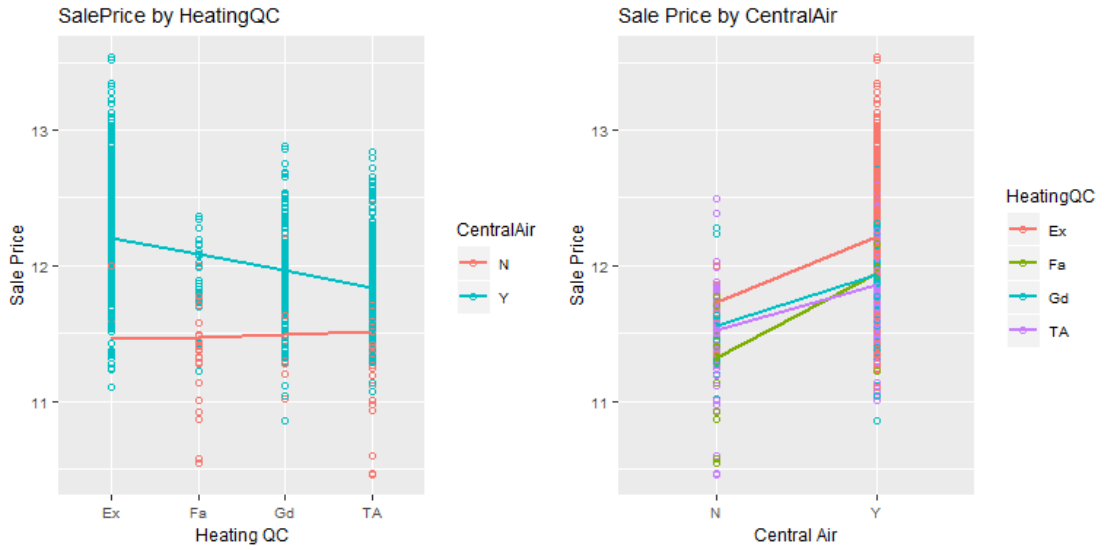
Next, we will utilize a two-way ANOVA on two categorical variables as they relate to the sale price of a home. When looking at various categorical variables, trying to find an even blend of saturation across the group means is our goal. Our investigation led us to HeatingQC and CentralAir. If you've ever been to Iowa, the summers can be blisteringly hot, while the winters can be extremely cold. As such, we hypothesize that both factors will play a vital role in the sale price of a home as comfort is more of a necessity in such a wide-ranging climate.

Initially, we looked at summary stats to see what the means look like across the rating groups. The EDA showed that the CentralAir = No and HeatingQC = Poor are categories with very low counts. We decided to subset that out as that will cause problems for with ANOVA. See Appendix [21] for updated table. We're starting the analysis with the logged sale price response variable, as our earlier research indicated improvements with a transform which was confirmed.

```
##   HeatingQC CentralAir   N   Min   Max  IQR  Mean   SD   SE
## 1        Ex          N   8 11.34 12.00 0.20 11.73 0.20 0.07
## 2        Fa          N  24 10.54 12.00 0.41 11.32 0.37 0.08
## 3        Gd          N  13 11.02 12.28 0.33 11.55 0.37 0.10
## 4        Po          N   1 11.37 11.37 0.00 11.37   NA
## 5        TA          N  49 10.46 12.49 0.42 11.52 0.41 0.06
## 6        Ex          Y 726 11.10 13.53 0.49 12.21 0.37 0.01
## 7        Fa          Y  25 11.23 12.37 0.35 11.94 0.27 0.05
## 8        Gd          Y 227 10.86 12.89 0.35 11.93 0.31 0.02
## 9        TA          Y 378 11.00 12.83 0.31 11.85 0.28 0.01
```



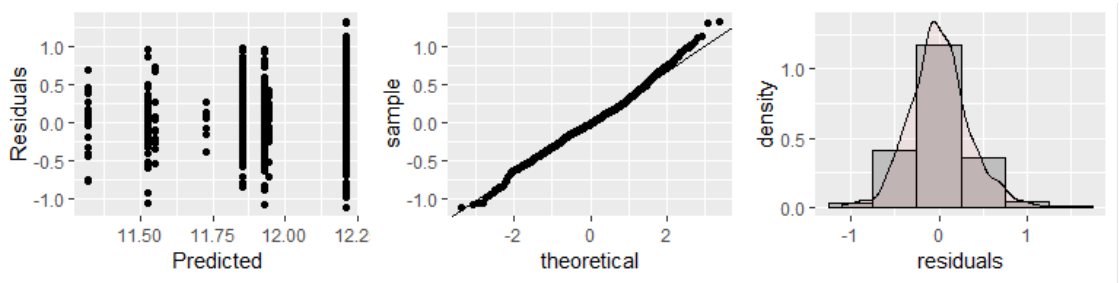
The standard deviations (the whiskers) in the means plot above give us insight to evaluate the equal variance assumption. Visually, the variances are relatively equal, but our sumstats table shows us that we have unequal sample sizes across the different HeatingQC groups. Below are the scatter plots of each specific categorical variable showing relatively equal variance, but with a significant difference in sample size. It is important to note that the mean sale price in some levels of heatingQC increase while the mean sale price in the two levels of CentralAir decrease. This indicates a possible interaction within the model.



Now we'll look at the residual diagnostics of our non-additive two-way ANOVA to get an understanding of the residual plots to further check our assumptions. To start, we'll fit the full model with an interaction term. Code in Appendix [22].

```
## Anova Table (Type III tests)
##
## Response: log_SalePrice
##                        Sum Sq   Df   F value    Pr(>F)
## (Intercept)          1100.56    1 9649.2372 < 2.2e-16 ***
## HeatingQC               1.26    3    3.6861   0.01162 *
## CentralAir              1.84    1   16.1573 6.131e-05 ***
## HeatingQC:CentralAir    0.89    3    2.5866   0.05166 .
## Residuals             164.47 1442
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



- Normality - Histogram and QQ Plots
  - QQ plot show some curvature in the Q3 section. But it doesn't look bad enough to violate the normality assumption.
  - The histogram of the residuals looks to be normally distributed.
- Constant Variance - Looking at the residuals plot, visually we see relatively even distributions between the groups. We may have some uneven sample sizes that are causing some clustering but for now we shall proceed with caution.
- Independence of variables - We have independence of variables
- Check for outliers - There does not appear to be any obvious outliers from the residuals.

Keeping unequal sample sizes in mind, the assumptions are met for our two-way ANOVA. The Type III sum of squares test shows us that the interaction term is right on the cusp of being significant (F-stat: 2.587, p = 0.0517). As such, we've decided to keep it in the model. It is evident that there is a difference in sale price among groups of CentralAir vs HeatingQC (F-stat: 9649.24, p < .0001). This is a very large F-stat, but this could be explained by the differences in sample sizes between groups or other co-variates that we failed to control for.

Conclusion

Given the existence of two categorical variables with a continuous response variable, a two-way ANOVA can be run to determine differences of the continuous response among groups of two categorical variables. The analysis was run on pairings of central air conditioning and the quality rating of the heating system. We determined that there is a significant difference in the sale price of houses between different pairings of those two groups, with some concerns over sample sizes between the groups.

Appendix

1. DataImport above

```
train <- read.csv("train.csv", stringsAsFactors = FALSE)
test <- read.csv("test.csv", stringsAsFactors = FALSE)

train$split <- "train" #Add column to delineate train
test$split <- "test" #Add column to delineate test
test$SalePrice <- NA #Dummy value for empty SalePrice
df <- rbind(train, test) #Append train and test to make data cleanup
easier
```

2. Data Cleaning above

```
#Subsets the data to a Sale Condition of "Normal"" and removal of
commercial/severly damaged properties.
#This was done to represent a typical homesale.
df2 <- subset(df, SaleCondition=="Normal" | MSZoning !="C (all)" |
Functional != "Sev")

#Remove rows with NA values per column
df3 <- df2[!is.na(df2$Utilities),]
df3 <- df3[!is.na(df3$Exterior1st),]
df3 <- df3[!is.na(df3$Exterior2nd),]
df3 <- df3[!is.na(df3$MasVnrType),] #Removes same rows for MsVnrArea
df3 <- df3[!is.na(df3$Electrical),]
df3 <- df3[!is.na(df3$BsmtFullBath),]
df3 <- df3[!is.na(df3$BsmtHalfBath),]
df3 <- df3[!is.na(df3$SaleType),]

#Replace NA with 0 LotFrontage
df3$LotFrontage[is.na(df3$LotFrontage)] <- 0

#Replace NA with YearBuilt for GarageYrBlt
df3$GarageYrBlt <- ifelse(is.na(df3$GarageYrBlt) &
!is.na(df3$GarageType), df3$YearBuilt, df3$GarageYrBlt)


#Transform remaining NA values to "NA"
#Replacing all numeric "NA" values with 0 #
df3_num <- names(df3[,sapply(df3,function(x) {is.numeric(x)})])
df3[,df3_num] <- sapply(df3[,df3_num],function(x){
ifelse(is.na(x),0,x)})


#Replacing all character "NA" values with "None"
df3_char <- names(df3[,sapply(df3,function(x){is.character(x)})])
df3[,df3_char] <-
sapply(df3[,df3_char],function(x){ifelse(is.na(x),"None",x)})

#Turn all character columns to factors
df3[sapply(df3, is.character)] <- lapply(df3[sapply(df3,
is.character)], as.factor)
#Check NA's by column
#colSums(is.na(df3))
```

3. Feature Addition

```
#Added Features for Analysis
df3$TotalSqFt <- (df3$GrLivArea + df3$TotalBsmtSF + df3$GarageArea)
df3$TotalPorchSqFt <-
```

```
(df3$OpenPorchSF+df3$EnclosedPorch+df3$ScreenPorch)
df3$TotalBaths <-
df3$BsmtFullBath+(df3$BsmtHalfBath*0.5)+df3$FullBath+(df3$HalfBath*0.5)
df3$HouseAge <- as.numeric(df3$YrSold) - as.numeric(df3$YearBuilt)
```

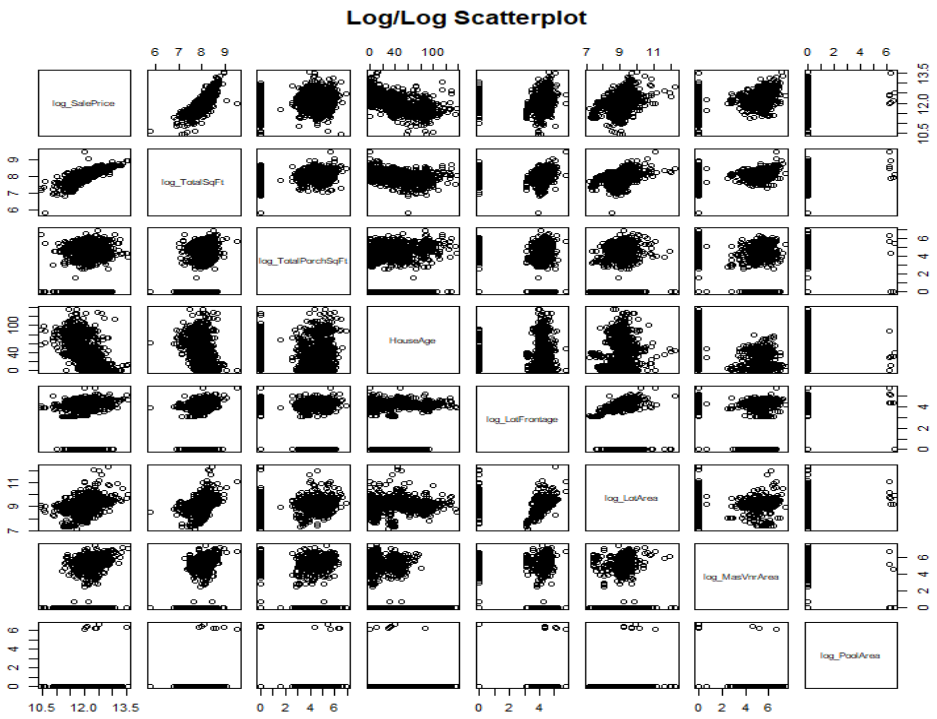
4. EDA Graphs untransformed – Code

```
#Also to check for multi-collinelarity
#Should we include calculated columns to divide all SF metrics by 100?
pairs(~SalePrice + TotalSqFt + TotalPorchSqFt + HouseAge +
        LotFrontage + LotArea + MasVnrArea + PoolArea, data=df3,
main="Untransformed Scatterplot")

#Different Scatterplot view of the training data.
require(psych)
library(psych)
pairs.panels(df3[df3$split == "train",c("SalePrice", "TotalSqFt",
"TotalPorchSqFt","HouseAge","LotFrontage",
                          "LotArea", "MasVnrArea", "PoolArea")],
             main="Untransformed",
             method = "pearson",
             density = TRUE)
```

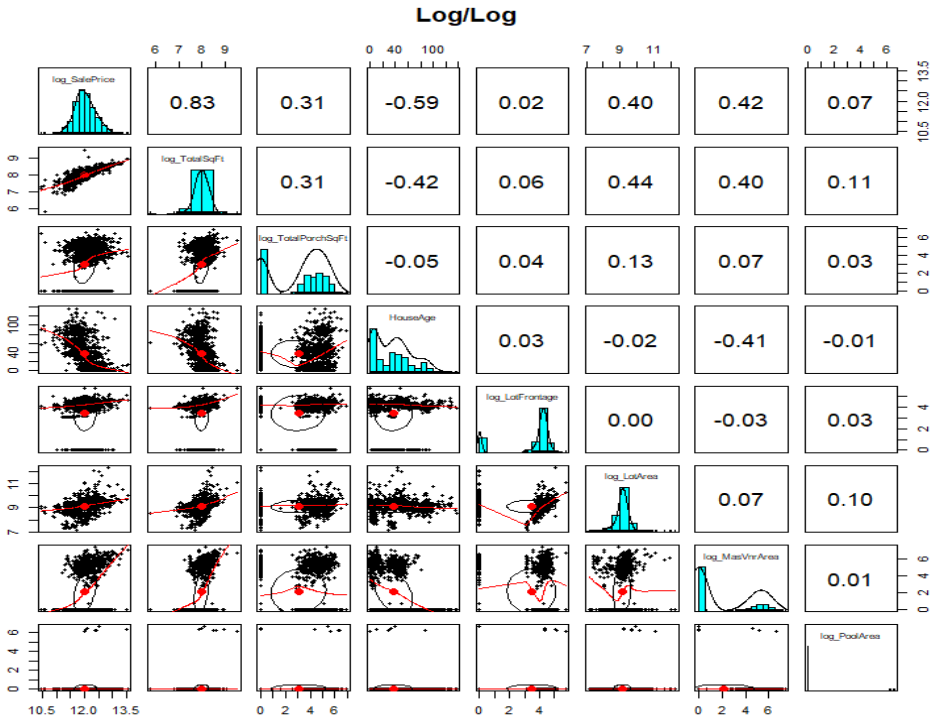
5. Log Transforms

```
# Logged variables for regression
df3$log_SalePrice <-
as.numeric(ifelse(df3$split=="train",log(df3$SalePrice)," "))
df3$log_TotalSqFt <- log(df3$TotalSqFt+1)
df3$log_TotalPorchSqFt <- log(df3$TotalPorchSqFt+1)
df3$log_HouseAge <- log(df3$HouseAge+1)
df3$log_LotFrontage <- log(df3$LotFrontage+1)
df3$log_LotArea <- log(df3$LotArea+1)
df3$log_MasVnrArea <- log(df3$MasVnrArea+1)
df3$log_PoolArea <- log(df3$PoolArea+1)
```

6. EDA Graphs – Log/Log Transformation – Scatter above



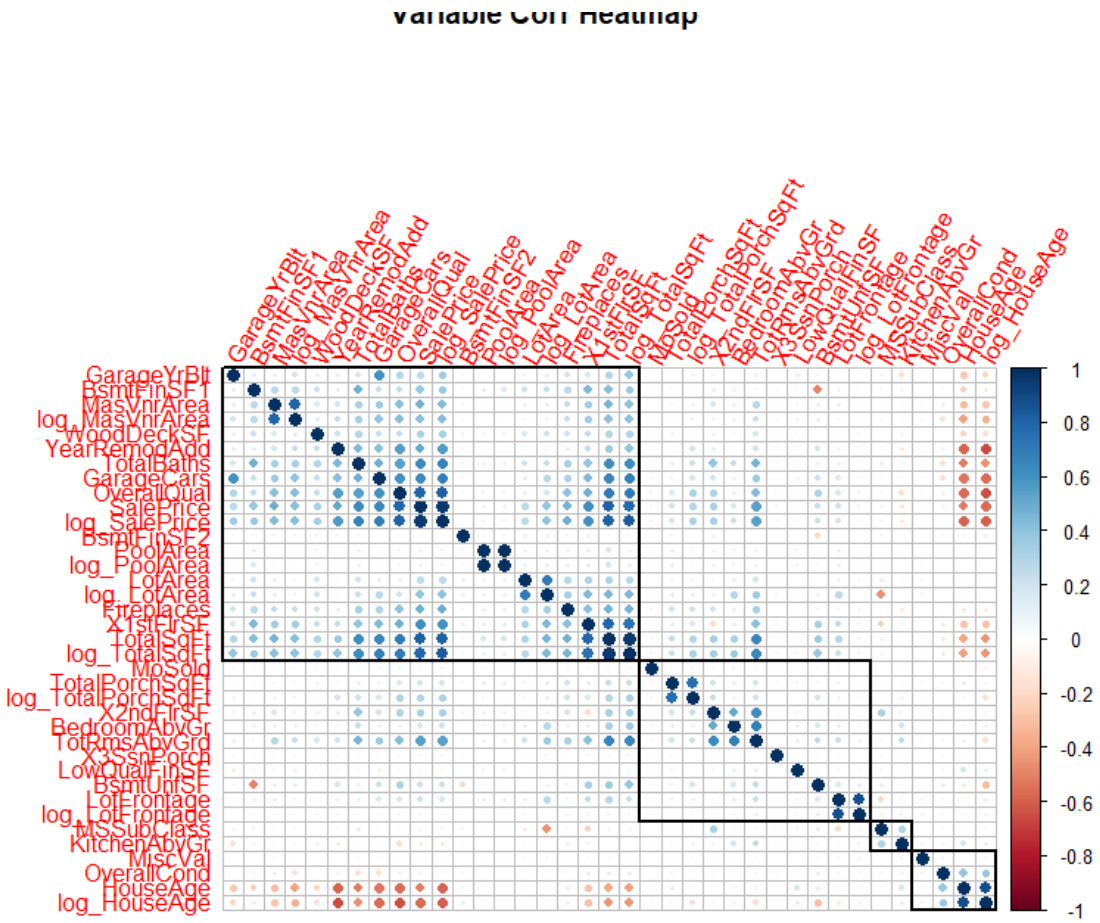
7. EDA Graphs – Log/Log Transformation – Residuals above



8. EDA Graphs – Log/Log – Code

```
#Remove columns used to calculate Features below
#As some of the aggregated variables are related, we remove them to
stifle collinearity

df3 <- subset(df3, select = -c(Id, GrLivArea, TotalBsmtSF, GarageArea,
OpenPorchSF, EnclosedPorch, ScreenPorch, BsmtFullBath, BsmtHalfBath,
FullBath, HalfBath, YrSold, YearBuilt))

#Split df3 back to training set to use the logged Sale Price and begin
considering our model.
split_df <- split(df3, df3$split)
df_train <- split_df[[2]]
df_test <- split_df[[1]]


#Scatterplot matrix for log/linear relationship
pairs(~log_SalePrice + TotalSqFt + TotalPorchSqFt + HouseAge +
        LotFrontage + LotArea + MasVnrArea + PoolArea, data=df_train,
main="Log/Linear Scatterplot")

#Different Scatterplot view
pairs.panels(df_train[,c("log_SalePrice", "TotalSqFt",
"TotalPorchSqFt","HouseAge","LotFrontage",
                         "LotArea", "MasVnrArea", "PoolArea")],
             main="Log/Linear",
             method = "pearson",
             density = TRUE)

#Scatterplot matrix for log/log relationship
pairs(~log_SalePrice + log_TotalSqFt + log_TotalPorchSqFt + HouseAge +
        log_LotFrontage + log_LotArea + log_MasVnrArea + log_PoolArea,
data=df_train, main="Log/Log Scatterplot")

#Different Scatterplot view
#Returns -inf NAN values when logging explanatory variables
pairs.panels(df_train[,c("log_SalePrice", "log_TotalSqFt",
"log_TotalPorchSqFt","HouseAge","log_LotFrontage",
                         "log_LotArea", "log_MasVnrArea",
"log_PoolArea")],
             main="Log/Log",
             method = "pearson",
             density = TRUE)
```

9. COR Heatmap – Chart above

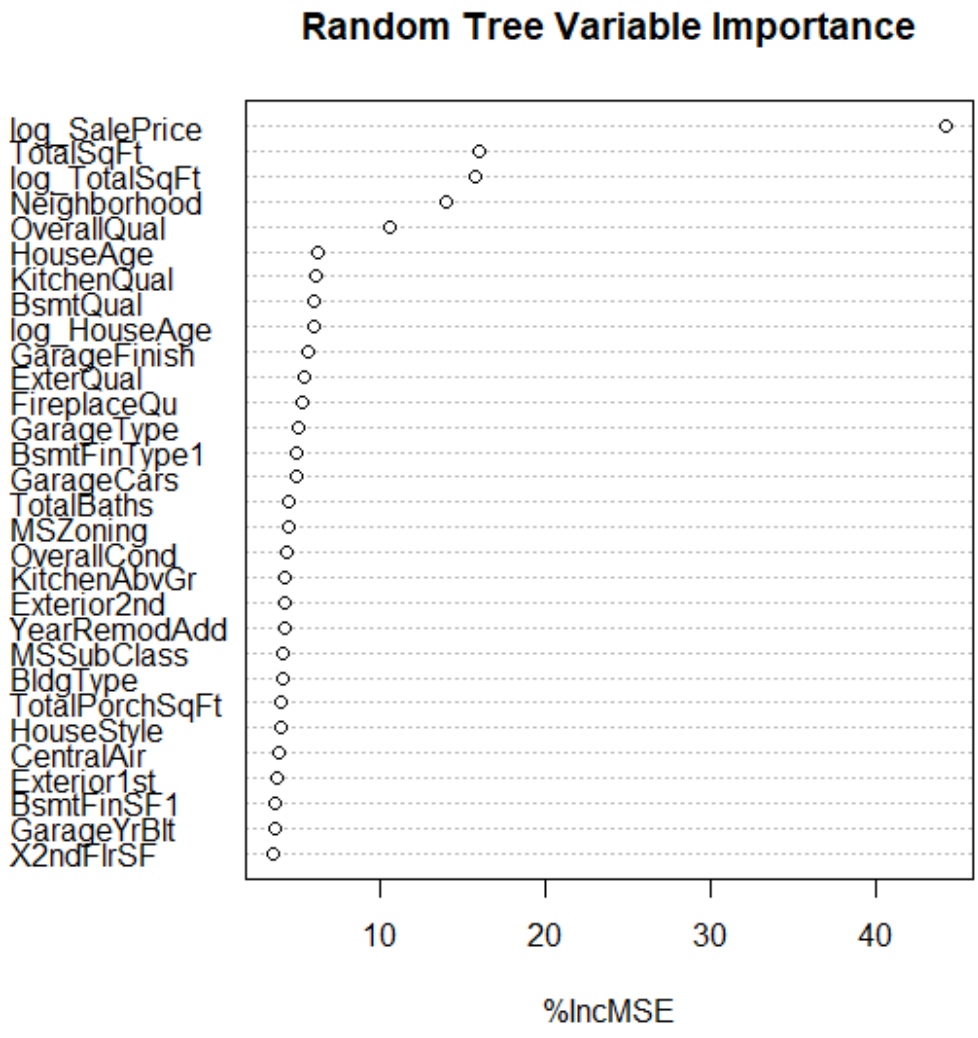


10. COR Heatmap – Code above

```
#Check for multi-collinelarity
library(corrplot)
#Return numeric values only
df_train_numeric <- df_train[, sapply(df_train, is.numeric)]
#df_train_numeric <- df_train_numeric[,-c(30:36)] #Remove log columns with
NaN
#Correlation Plot
df_corr <- round(cor(df_train_numeric),2)
corrplot(df_corr, method="circle", order="hclust", addrect=4, win.asp=.7,
title="Variable Corr Heatmap",tl.srt=60)

#Possible Correlations
#TotalSqFt vs TotalBaths <br/>
#TotalSqFt vs OverallQual
```

```
#TotalSqFt vs X1stFlrSF
#TotalSqFt vs GarageCars
#TotalSqFt vs TotRmsAbvGrd
#TotRmsAbvGrd vs X2ndFlrSF
#TotRmsAbvGrd vs BedroomAbvGrd
#TotRmsAbvGrd vs GarageYrBlt
```

11. Random Forrest -Chart above



12. Random Forrest - Code

```
library(randomForest)
rf_model <- randomForest(SalePrice~., data=df_train, importance = TRUE)
#importance(rf_model)
#Variable importance for placement order (Forward, Backward, Stepwise)
varImpPlot(rf_model,type=1, main='Random Tree Variable Importance')
set.seed(25) #Used this to sanity check the ASE results
```

13. MLR – Split Training Set and Residuals above

```
trainingSet <- df3[df3$split == "train",] # Getting back just the records
found in `training.csv`
index<-sample(1:dim(trainingSet)[1], dim(trainingSet)[1] / 2, replace = F)
#Used to divide the set into 2 sets

#Dropping unused columns
columnsToDrop <- c("Utilities",
"SaleCondition","log_TotalBaths","log_TotalSqFt_100","Id","split",
"log_TotalSqFt","log_TotalPorchSqFt","log_HouseAge",
"log_MasVnrArea","log_LotFrontage","log_PoolArea",
"log_SalePrice","log_LotArea")
trainingSet <- trainingSet[, !names(trainingSet) %in% columnsToDrop]

par(mfrow=c(2,2))
plot(lm(log(SalePrice)~.,trainingSet))

#Splitting the training and test set
mlrHousesTrain<-trainingSet[index,]
mlrHousesTest<-trainingSet[-index,]
```

14. Forward Selection Model

```
library(leaps)

#There are 239 variables as it splits categoricals into multiple
variables. Setting that as the max. really.big must be true on sets with
more than 50 variables
reg.fwd=regsubsets(log(SalePrice)~., data = mlrHousesTrain, method =
"forward", really.big=T, nvmax=239)

#Getting total runs for the x-axis of the plots below by counting how many
bic values are returned
totalRuns <- length(summary(reg.fwd)$bic)

#Using code from HW2 to plot BIC, ADJR2, and RSS at each model selection
step
par(mfrow=c(1,3))
bics<-summary(reg.fwd)$bic
plot(1:totalRuns,bics,type="l",ylab="BIC",xlab="# of predictors")
index<-which(bics==min(bics))
points(index,bics[index],col="red",pch=10)

adjr2<-summary(reg.fwd)$adjr2
plot(1:totalRuns,adjr2,type="l",ylab="Adjusted R-squared",xlab="# of
predictors")
index<-which(adjr2==max(adjr2))
points(index,adjr2[index],col="red",pch=10)
```

```
rss<-summary(reg.fwd)$rss
plot(1:totalRuns,rss,type="l",ylab="Train RSS",xlab="# of predictors")
index<-which(rss==min(rss))
points(index,rss[index],col="red",pch=10)
```

15. ASE Calculation

```
#Function provided by Dr. Turner in Unit 2 to give predictions at a
specific step of feature selection
predict.regsubsets =function (object , newdata ,id ,...){
  form=as.formula (object$call [[2]])
  mat=model.matrix(form ,newdata )
  coefi=coef(object ,id=id)
  xvars=names(coefi)
  mat[,xvars]%*%coefi
}

#Get predictions at each step and calculate ASE
testASE<-c()
for (i in 1:totalRuns){
  predictions<-
predict.regsubsets(object=reg.fwd,newdata=mlrHousesTest,id=i)
  testASE[i]<-mean((log(mlrHousesTest$SalePrice)-predictions)^2)
}
par(mfrow=c(1,1))
plot(1:totalRuns,testASE,type="l",xlab="# of predictors",ylab="Test vs
Train ASE",ylim=c(0,.5))

#Get the model step with the lowest ASE
lowestASEModelIndex<-which(testASE==min(testASE))
# in case multiple models have the same ASE
if (length(lowestASEModelIndex) > 1) {
  lowestASEModelIndex = lowestASEModelIndex[1]
}

#Plot the ASE at each step
points(index,testASE[lowestASEModelIndex],col="red",pch=10)
rss<-summary(reg.fwd)$rss

testSampleSize <- dim(mlrHousesTest)[1]
lines(1:totalRuns,rss/testSampleSize,lty=3,col="blue")
```

16. Linear Model

```
#Model with all factors selected by forward model section at step 58
forwardSelectedModel <-
lm(log(SalePrice)~MSZoning+LotFrontage+LotArea+LotConfig+Neighborhood+Cond
ition1+HouseStyle+RoofMatl+Exterior1st+ExterCond+Foundation+BsmtExposure+B
smtFinType1+BsmtFinSF1+BsmtFinType2+Heating+CentralAir+Electrical+LowQualF
```

```
inSF+TotRmsAbvGrd+Functional+Fireplaces+FireplaceQu+GarageType+GarageFinis
h+Fence+MoSold+SaleType+TotalSqFt+Condition2+GarageCond+MiscFeature, data
= mlrHousesTrain)
```

17. ASE of hybrid model

```
hybridModel <-
lm(log(SalePrice)~LotArea+LotConfig+Neighborhood+Condition1+RoofMatl+Exter
ior1st+ExterCond+BsmtExposure+BsmtFinType1+BsmtFinType2+Heating+CentralAir
+Functional+GarageType+GarageFinish+SaleType+TotalSqFt, data =
mlrHousesTrain)
#Removing test set rows levels that were not present in the training set.
I should've checked for level representation in both sets with the seed
specified. Too late to fix the right way now and change my analysis above.
These account for 16 rows and prompts a warning because of prediction on a
set with less rows than the one used to make the fit.
mlrHousesTest2 <- mlrHousesTest[!mlrHousesTest$RoofMatl %in%
c("ClyTile","Roll","Membran") & mlrHousesTest$Heating != "OthW",]
hybridModelPredictions <- predict(hybridModel, newdata=mlrHousesTest2,
interval="confidence")

#Calculating new ASE for the hybrid model
hybridModelTestASE <- mean((log(mlrHousesTest2$SalePrice) -
hybridModelPredictions[,1])^2) #Selecting the fit column of predictions
hybridModelTestASE
head(exp(hybridModelPredictions))

library(ggplot2)
library(dplyr)
```

18. Two-way ANOVA – Summary Code

```
#split the cleaned data back into train
Clean.train <- subset(df3, split=="train")

# Summary function from the HW3
mysummary<-function(x){
  result<-
round(c(length(x),min(x),max(x),IQR(x),mean(x),sd(x),sd(x)/sqrt(length(x))
),2)
  names(result)<-c("N","Min","Max","IQR","Mean","SD","SE")
  return(result)
}
#Summary stats
sumstats<-
aggregate(log(SalePrice)~HeatingQC*CentralAir,data=Clean.train,mysummary)
sumstats<-cbind(sumstats[,1:2],sumstats[,-(1:2)])
```

```
# Subset out the HeatingQC = "Po"
#with(Clean.train, table(HeatingQC, CentralAir))
Clean.train <- Clean.train[!(Clean.train$HeatingQC == "Po"),]

#Summary stats w/o Po
sumstats<-
aggregate(log(SalePrice)~HeatingQC*CentralAir,data=Clean.train,mysummary)
sumstats<-cbind(sumstats[,1:2],sumstats[,-(1:2)])
sumstats

 # Means Plot from the HW3

ggplot(sumstats,aes(x=HeatingQC,y=Mean,group=CentralAir,colour=CentralAir)
)+
  ylab("Sale Price")+
  geom_line()+
  geom_point()+
  geom_errorbar(aes(ymin=Mean-SD,ymax=Mean+SD),width=.1)
```

19. SumStats with reduced categories above

```
##   HeatingQC CentralAir   N   Min   Max  IQR  Mean   SD   SE
## 1        Ex          N   8 11.34 12.00 0.20 11.73 0.20 0.07
## 2        Fa          N  24 10.54 12.00 0.41 11.32 0.37 0.08
## 3        Gd          N  13 11.02 12.28 0.33 11.55 0.37 0.10
## 4        TA          N  49 10.46 12.49 0.42 11.52 0.41 0.06
## 5        Ex          Y 726 11.10 13.53 0.49 12.21 0.37 0.01
## 6        Fa          Y  25 11.23 12.37 0.35 11.94 0.27 0.05
## 7        Gd          Y 227 10.86 12.89 0.35 11.93 0.31 0.02
## 8        TA          Y 378 11.00 12.83 0.31 11.85 0.28 0.01
```

20. Two-way ANOVA – Scatter

```
library("gridExtra")
p1 <- ggplot(Clean.train, aes(x = HeatingQC, y = log_SalePrice)) +
        geom_point(shape=1) +
        geom_smooth(method=lm, se=FALSE) +
        xlab("Heating QC") +
        ylab("Sale Price") +
        theme(text = element_text(size=9)) +
        ggtitle("SalePrice by HeatingQC")

p2 <- ggplot(Clean.train, aes(x = CentralAir, y = log_SalePrice)) +
        geom_point(shape=1) +
        geom_smooth(method=lm, se=FALSE) +
        xlab("Central Air") +
```

```
        ylab("Sale Price") +
        theme(text = element_text(size=9)) +
        ggtitle("Sale Price by CentralAir")
grid.arrange(p1, p2, ncol=2)
```

21. Two-way ANOVA – Residuals above

```
library(car)
require(gridExtra)
library(gridExtra)
library(grid)
library(ggplot2)

ano.fit <-
aov(log_SalePrice~HeatingQC+CentralAir+HeatingQC:CentralAir,data=Clean.tra
in)
Anova(ano.fit,type=3)

Anovadata<-
data.frame(fitted.values=ano.fit$fitted.values,residuals=ano.fit$residuals
)
#ANOVA and residual code borrowed from HW3
#Residual vs Fitted
plot1<-
ggplot(Anovadata,aes(x=fitted.values,y=residuals))+ylab("Residuals")+
  xlab("Predicted")+geom_point()

#QQ plot of residuals  #Note the diagonal abline is only good for qqplots
of normal data.
plot2<-ggplot(Anovadata,aes(sample=residuals))+
  stat_qq()+geom_abline(intercept=mean(Anovadata$residuals), slope =
sd(Anovadata$residuals))

#Histogram of residuals
plot3<-ggplot(Anovadata, aes(x=residuals)) +
  geom_histogram(aes(y=..density..),binwidth=.5,color="black",
fill="gray")+
  geom_density(alpha=.05, fill="red")

grid.arrange(plot1, plot2,plot3, ncol=3)
```