DATA 605: Final Project

Andrew Carson

December 27, 2017

Introduction

Your final is due by the end of day on 12/27/2017. You should:

- post your solutions to your GitHub account.

You are also expected to:

- make a short presentation during our last meeting (3-5 minutes) or post a recording to the board.

This project will show off your ability to understand the elements of the class.

You are to register for Kaggle.com (free) and compete in the House Prices: Advanced Regression Techniques competition. https://www.kaggle.com/c/house-prices-advanced-regression-techniques . I want you to do the following.

Solution

Pick one of the quantitative independent variables from the training data set (train.csv) , and define that variable as X. Pick SalePrice as the dependent variable, and define it as Y for the next analysis.

```
#download the files and then load them from storage
train <- read.csv("C:/Users/Andy/Desktop/Personal/Learning/CUNY/DATA605/HW/FinalProject/train.csv",
                  stringsAsFactors = FALSE)

#define X and Y.
#X - GrLivArea: Above grade (ground) living area square feet
X <- train$GrLivArea
Y <-train$SalePrice
```

Probability.

Calculate as a minimum the below probabilities a through c. Assume the small letter "x" is estimated as the 1st quartile of the X variable, and the small letter "y" is estimated as the 2d quartile of the Y variable. Interpret the meaning of all probabilities.

```
# get quartiles
summary(X)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334    1130    1464    1515    1777    5642
```

```
x <- summary(X)[2]
x
```

```
## 1st Qu.
##    1130
```

```
summary(Y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   34900  130000  163000  180900  214000  755000
```

```
y <- summary(Y)[3]
y
```

```
## Median
## 163000
```

```
#set up for probability questions
df<-data.frame(cbind(X,Y), stringsAsFactors = FALSE)
```

a. P(X > x | Y > y) - the probability that X (Above grade (ground) living area square feet) is greater than x (1130 sq ft) given that Y (SalePrice) is greater than y ($163,000) is 0.989011.

```
#given that Y > y
data_a <- df[df$Y > y,]

#P(X > x | Y > y)
nrow(data_a[data_a$X > x,]) / nrow(data_a)
```

```
## [1] 0.989011
```

b. P(X > x, Y > y) - the probability that X (Above grade (ground) living area square feet) is greater than x (1130 sq ft) and that Y (SalePrice) is greater than y ($163,000) is 0.4931507.

```
#P(X > x, Y > y)
nrow(df[df$X > x & df$Y > y,]) / nrow(df)
```

```
## [1] 0.4931507
```

c. P(X < x | Y > y) - the probability that X (Above grade (ground) living area square feet) is less than x (1130 sq ft) given that Y (SalePrice) is greater than y ($163,000) is 0.01098901.

```
#given that Y > y
data_c <- df[df$Y > y,]

#P(X < x | Y > y)
nrow(data_c[data_a$X < x,]) / nrow(data_c)
```

```
## [1] 0.01098901
```

Does splitting the training data in this fashion make them independent? In other words, does P(X|Y)=P(X)P(Y))?

- Answer: no, they are not independent. The probabilities change significantly depending on what values of X we are looking at with respect to Y, and these should all be the same if X and Y are independent.

Check mathematically, and then evaluate by running a Chi Square test for association. You might have to research this.

```
#check mathematically
#does #P(X > x, Y > y) = P(X > x)*P(Y > y)? no
nrow(df[df$X > x & df$Y > y,]) / nrow(df) ==
  nrow(df[df$X > x,]) / nrow(df) * nrow(df[df$Y > y,]) / nrow(df)
```

```
## [1] FALSE
```

```
#chi square test
chisq.test(X,Y)
```

```
##
```

```
##  Pearson's Chi-squared test
##
## data:  X and Y
## X-squared = 589730, df = 569320, p-value < 2.2e-16
```

- Answer: X-squared is very high and the p-value is practically 0. Therefore, there is a very strong association between X and Y, and as such, they are NOT independent.

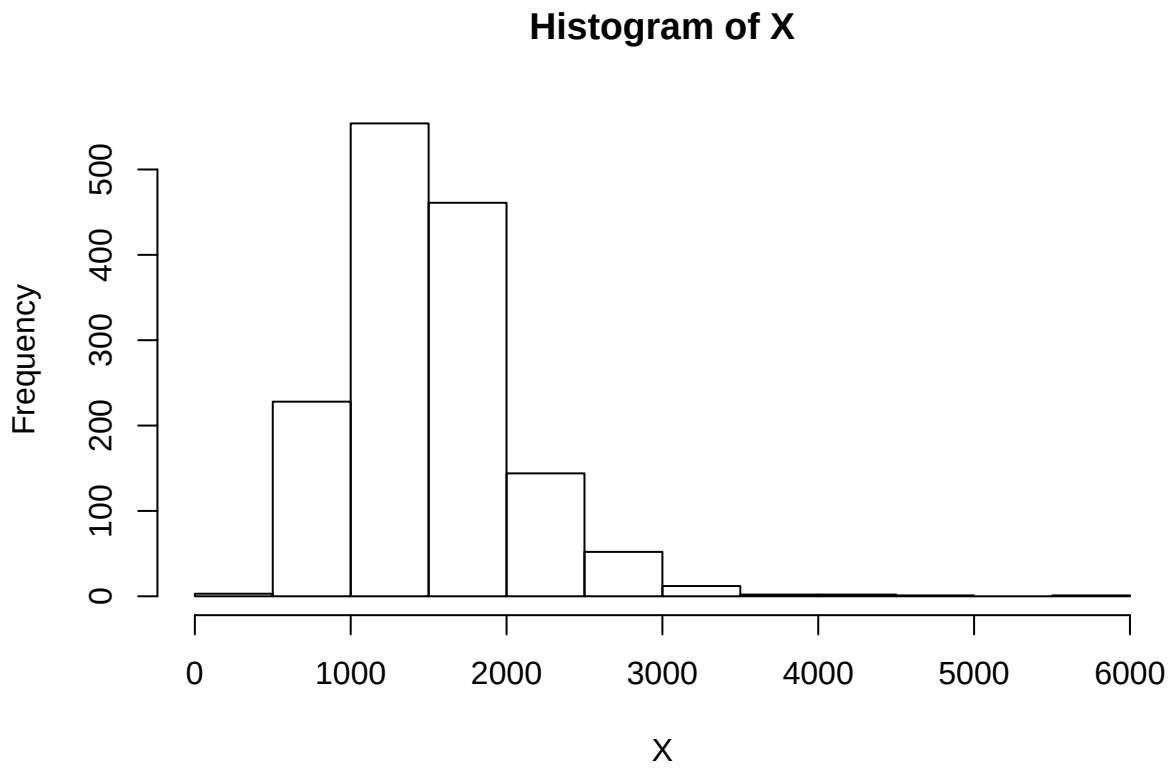
Descriptive and Inferential Statistics.

Provide univariate descriptive statistics and appropriate plots for both variables. Provide a scatterplot of X and Y.

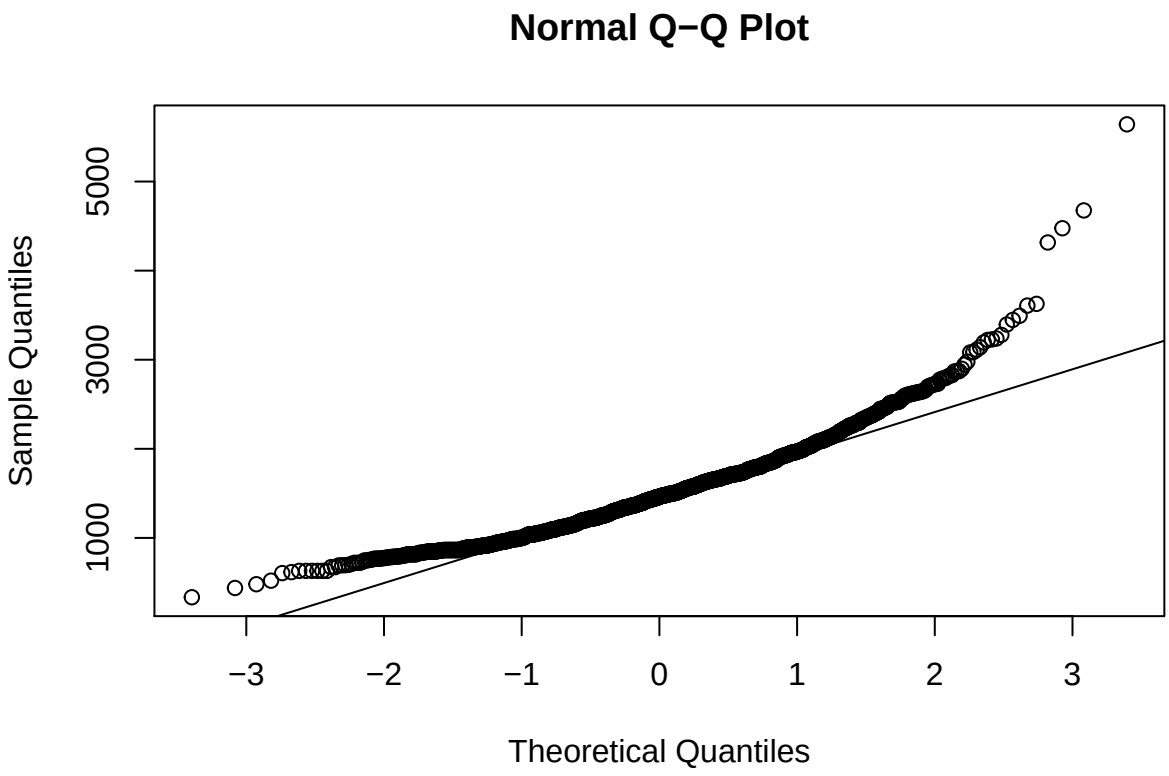
```
#X
summary(X)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334    1130    1464    1515    1777    5642
```

```
hist(X)
```



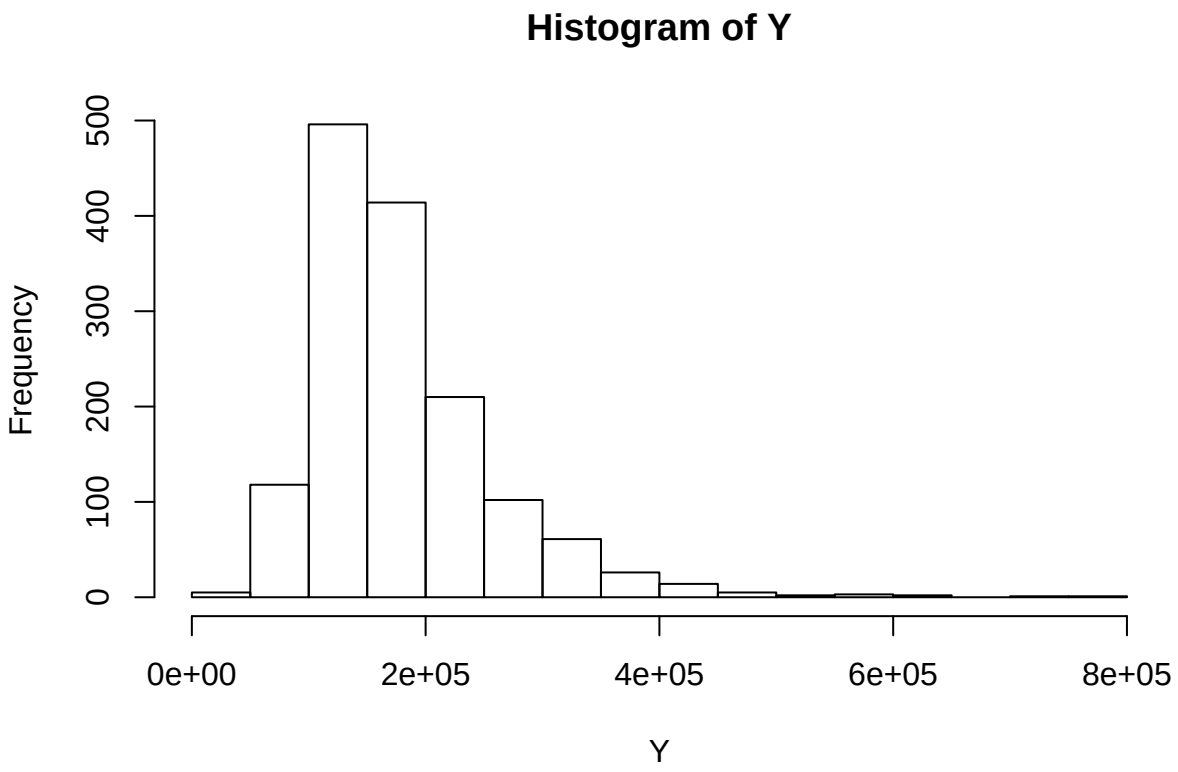

```
qqnorm(X)
qqline(X)
```



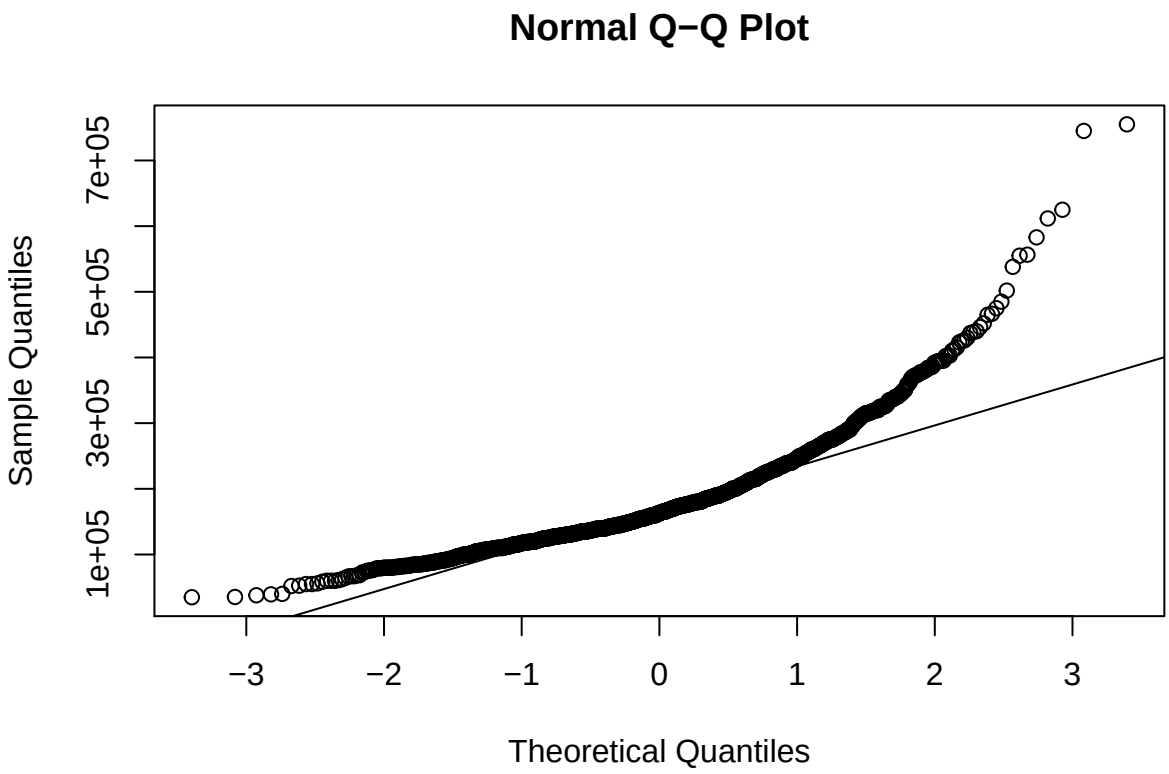

```
#Y
summary(Y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   34900  130000  163000  180900  214000  755000
```

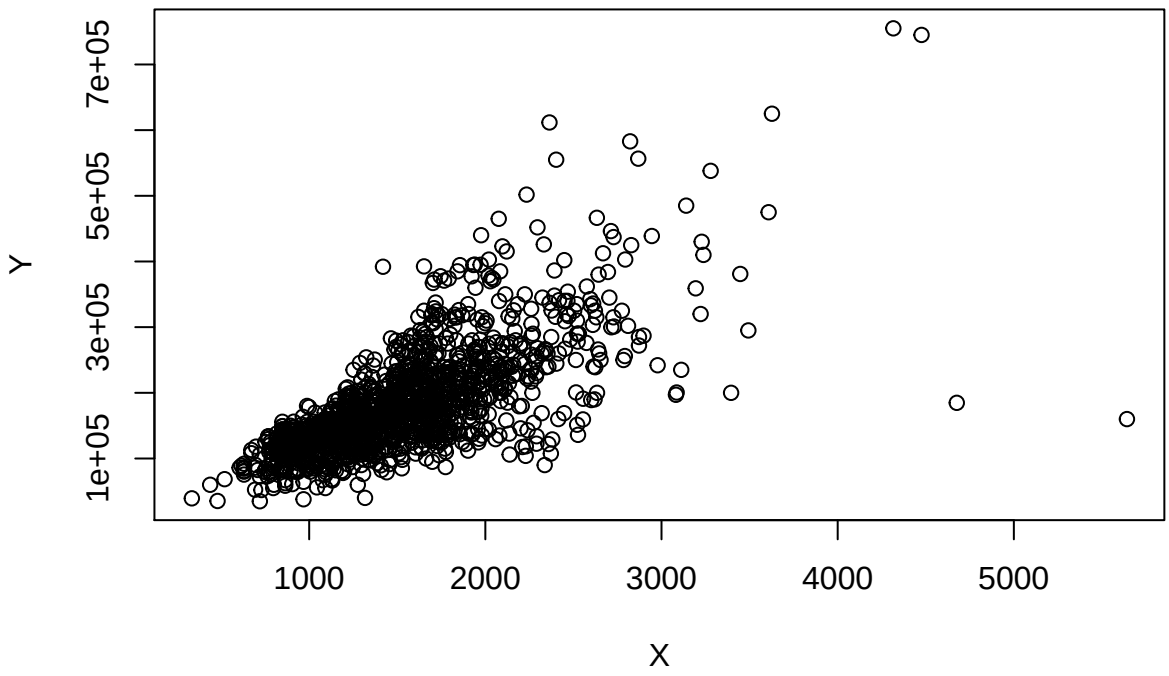
```
hist(Y)
```




```
qqnorm(Y)
qqline(Y)
```


Normal Q−Q Plot
Sample Quantiles
1e+05
3e+05
5e+05
7e+05
−3
−2
−1
0
1
2
3
Theoretical Quantiles

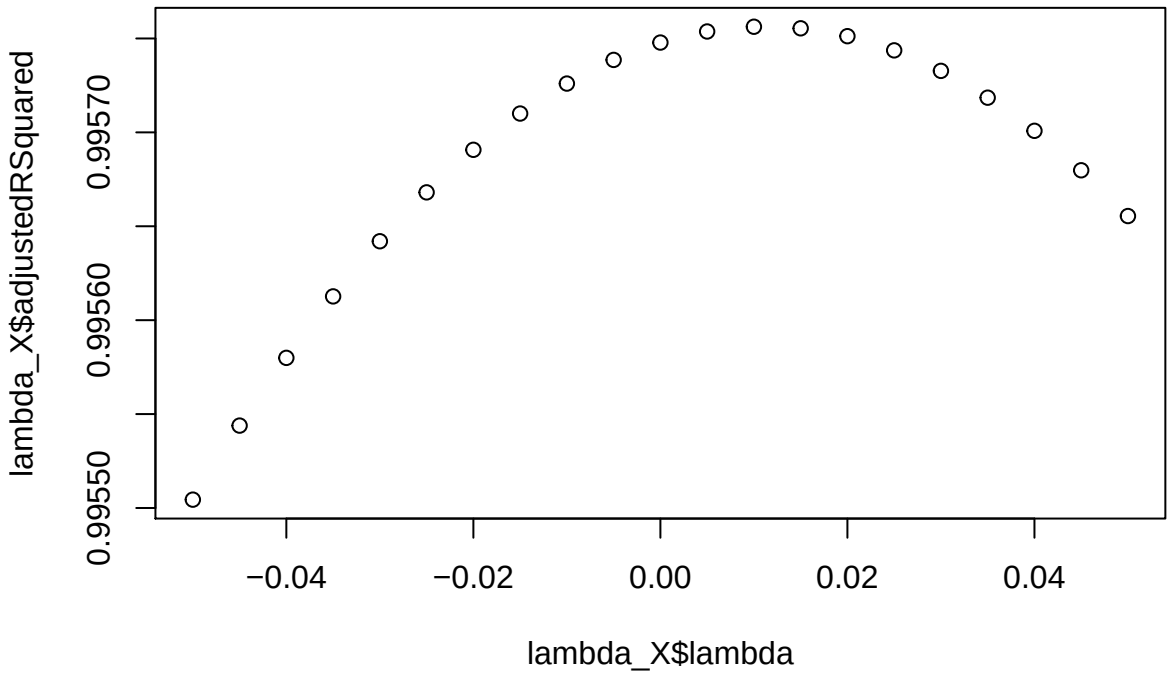
```
#X vs. Y
plot(X,Y)
```



Transform both variables simultaneously using Box-Cox transformations. You might have to research this.

```
### Box-Cox transformations
# boxCox function
boxCox <- function(a, lambda){
  if(lambda == 0){
    return(log(a))
  }else{
    return((a^lambda - 1)/lambda)
  }
}

#check normality function
bcNormality <-function(a,lambda){
    temp<-boxCox(a,lambda)
    temp2<-data.frame(qqnorm(temp,  plot.it = FALSE), stringsAsFactors = FALSE)
    temp3<-summary(lm(temp2$y ~ temp2$x))$adj.r.squared
    return(temp3)
}

#find best lambda for X
lambda_X<-c()
for(i in seq(-.05,.05,.005)){
  lambda_X<-rbind(lambda_X,cbind(i,bcNormality(X,i)))
}

lambda_X <- data.frame(lambda_X, stringsAsFactors = FALSE)
```

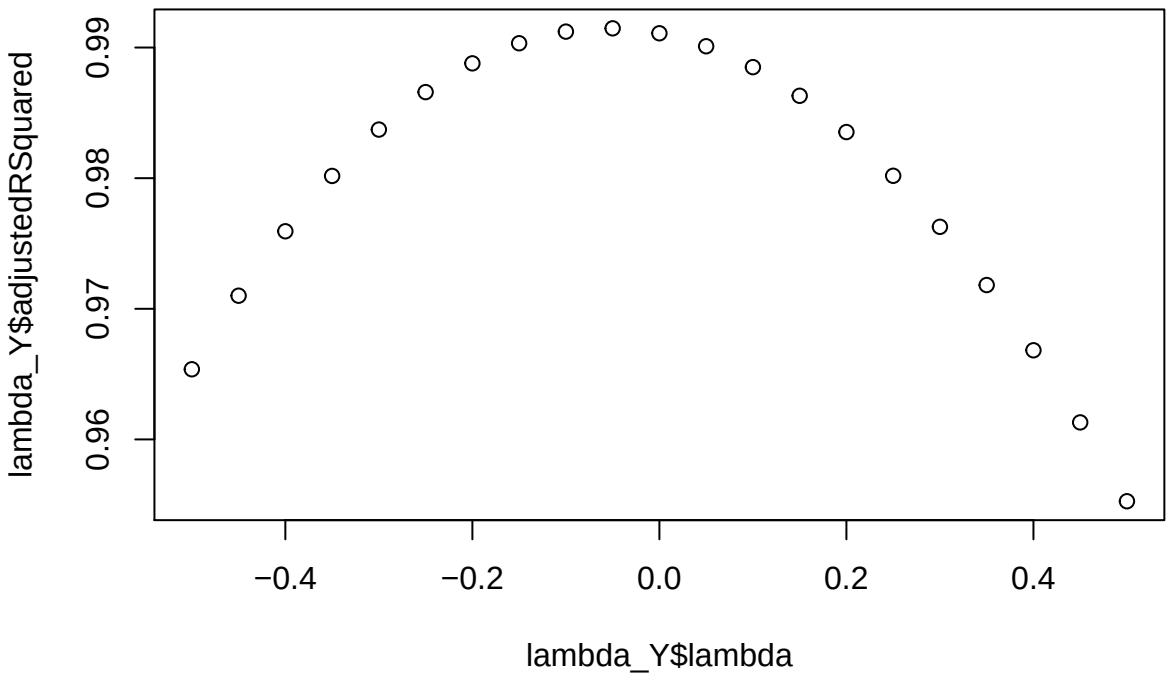
```
names(lambda_X) <-c("lambda","adjustedRSquared")
plot(lambda_X$lambda,lambda_X$adjustedRSquared)
```




```
bestLambda_X <- lambda_X$lambda[which(lambda_X$adjustedRSquared == max(lambda_X$adjustedRSquared))]
bestLambda_X
```

```
## [1] 0.01
```

```
bc_X <-boxCox(X, bestLambda_X)

#find best lambda for Y
lambda_Y<-c()
for(i in seq(-.5,.5,.05)){
  lambda_Y<-rbind(lambda_Y,cbind(i,bcNormality(Y,i)))
}

lambda_Y <- data.frame(lambda_Y, stringsAsFactors = FALSE)
names(lambda_Y) <-c("lambda","adjustedRSquared")
plot(lambda_Y$lambda,lambda_Y$adjustedRSquared)
```



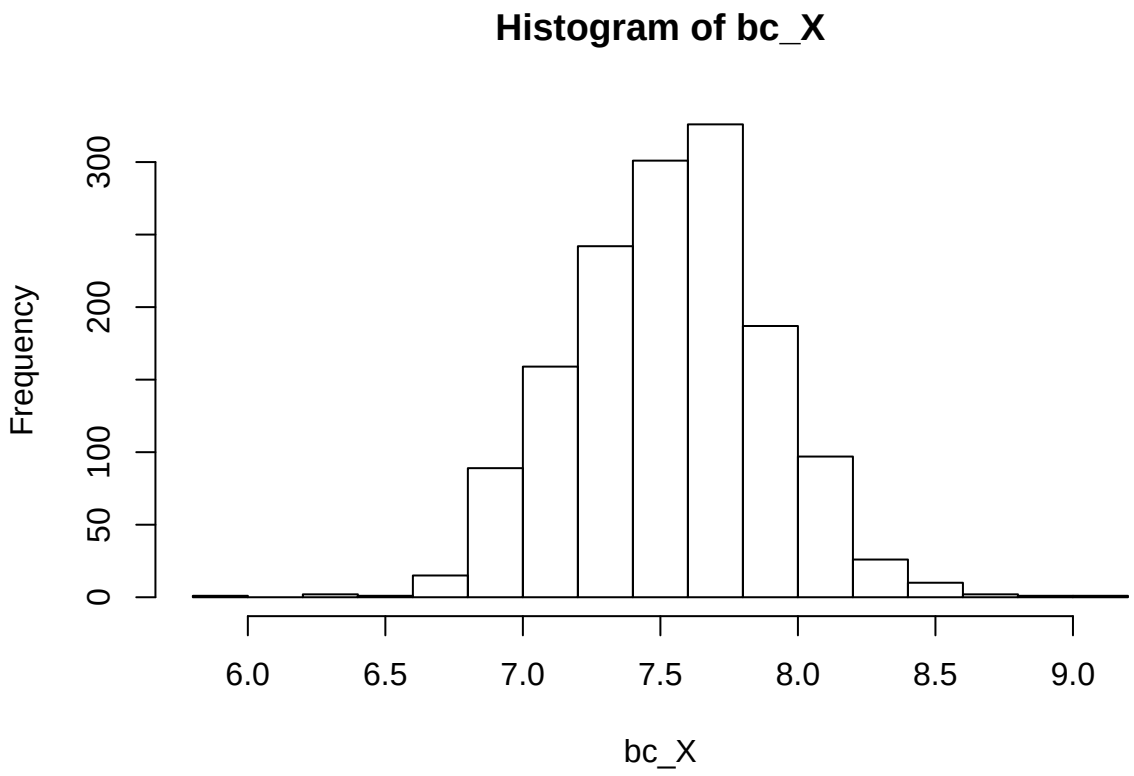

```
bestLambda_Y <- lambda_Y$lambda[which(lambda_Y$adjustedRSquared == max(lambda_Y$adjustedRSquared))]
bestLambda_Y
```

```
## [1] -0.05
```

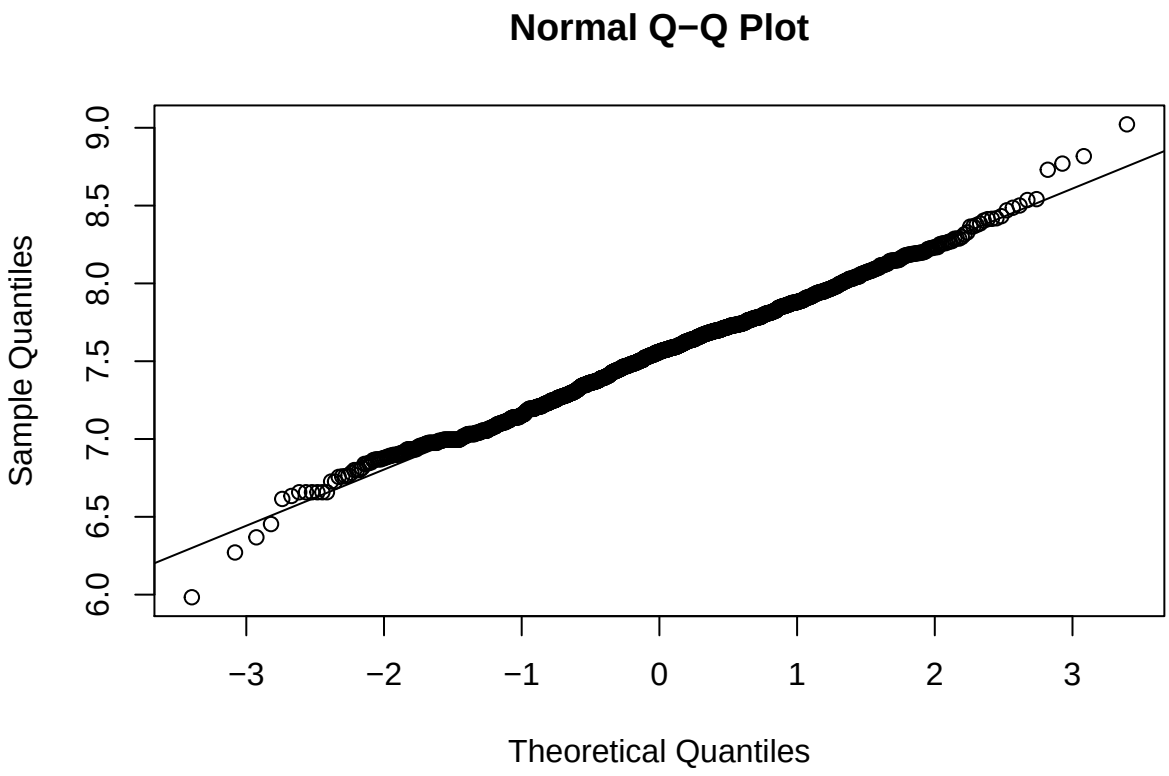
```
bc_Y <-boxCox(Y, bestLambda_Y)

### new summary statistics
#bc_X
summary(bc_X)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.983   7.282   7.561   7.539   7.770   9.022
```

```
hist(bc_X)
```



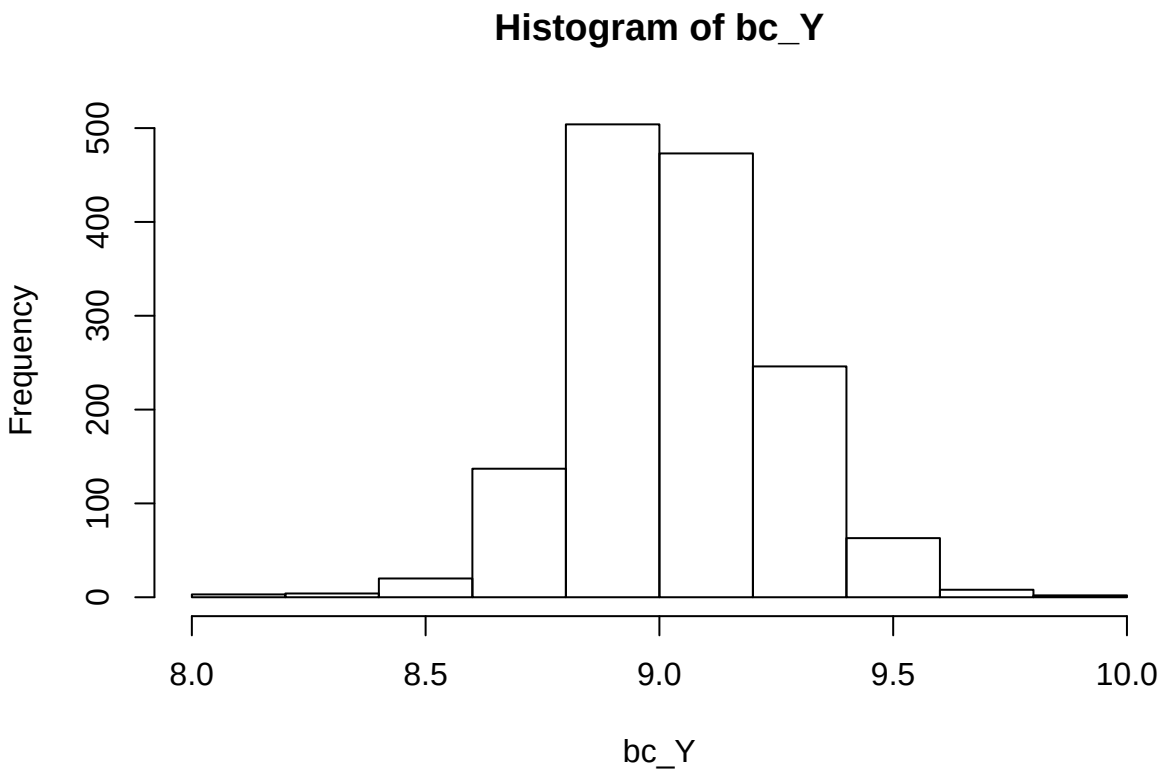

```
qqnorm(bc_X)
qqline(bc_X)
```



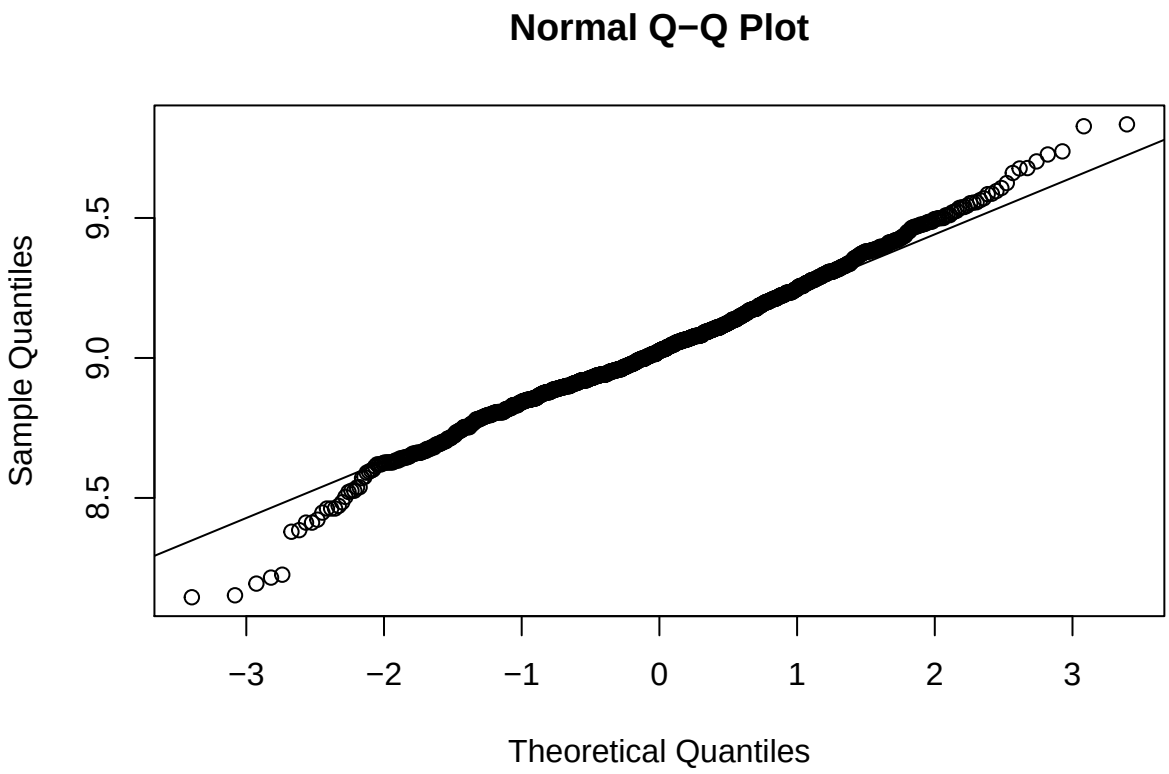

```
#bc_Y
summary(bc_Y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   8.145   8.900   9.025   9.035   9.173   9.834
```

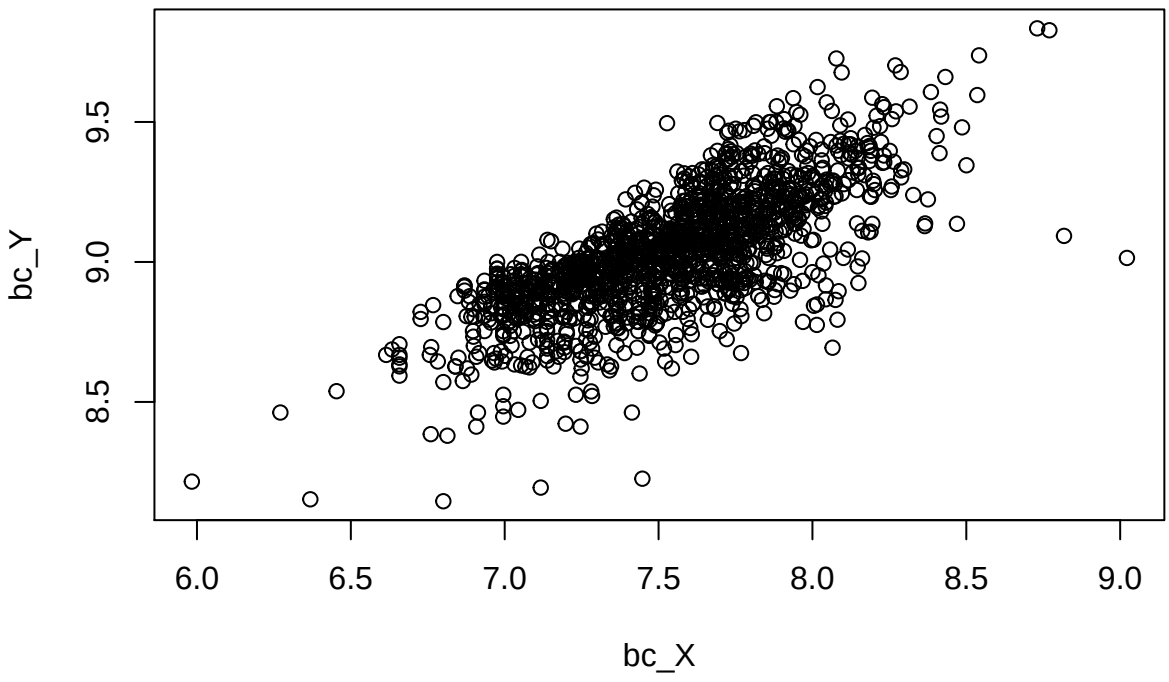
```
hist(bc_Y)
```




```
qqnorm(bc_Y)
qqline(bc_Y)
```




```
#plot transformed X and transformed Y
plot(bc_X,bc_Y)
```



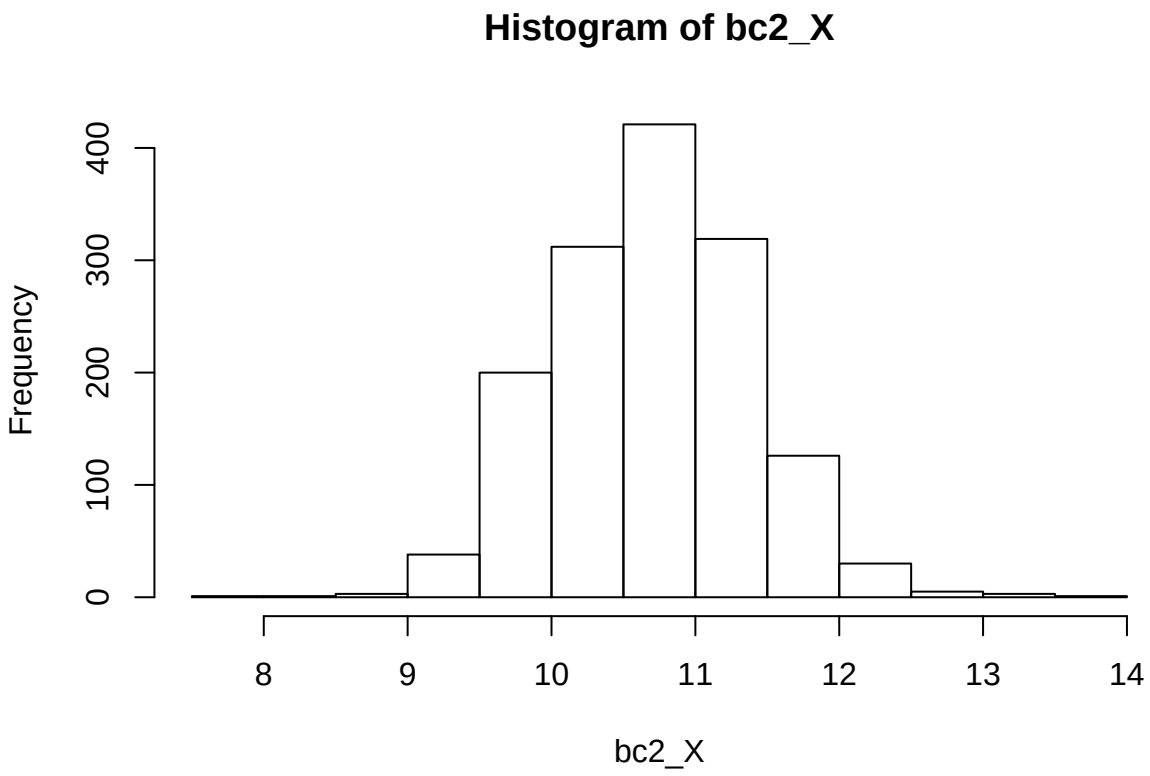

```
#how close is our manual approach? pretty close
library(MASS)
bc <- boxcox(Y ~ X, plotit = FALSE)
bc$x[which.max(bc$y)]
```

```
## [1] 0.1
```

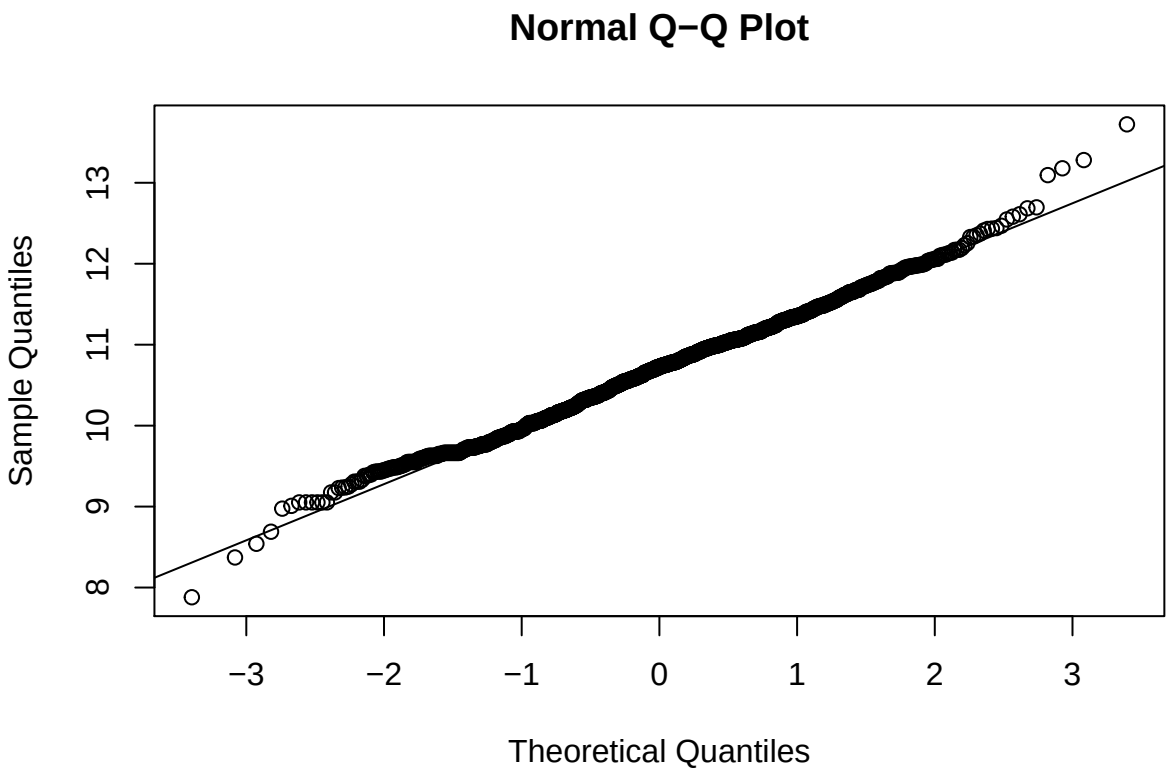
```
bc2_X <-boxCox(X,bc$x[which.max(bc$y)])
summary(bc2_X)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.88   10.20   10.73   10.70   11.13   13.72
```

```
hist(bc2_X)
```




```
qqnorm(bc2_X)
qqline(bc2_X)
```



Linear Algebra and Correlation.

Using at least three untransformed variables, build a correlation matrix.

```
df2 <- data.frame(train$LotArea, train$X1stFlrSF, train$X2ndFlrSF, stringsAsFactors = FALSE)
names(df2) <-c("LotArea","X1stFlrSF","X2ndFlrSF")
correlationMatrix <- cor(df2)
correlationMatrix
```

```
##              LotArea  X1stFlrSF   X2ndFlrSF
## LotArea   1.00000000  0.2994746  0.05098595
## X1stFlrSF 0.29947458  1.0000000 -0.20264618
## X2ndFlrSF 0.05098595 -0.2026462  1.00000000
```

Invert your correlation matrix. (This is known as the precision matrix and contains variance inflation factors on the diagonal.)

```
inverse_correlationMatrix <-MASS::ginv(correlationMatrix)
inverse_correlationMatrix
```

```
##            [,1]       [,2]       [,3]
## [1,]  1.1144421 -0.3600471 -0.1297831
## [2,] -0.3600471  1.1591459  0.2532538
## [3,] -0.1297831  0.2532538  1.0579380
```

Multiply the correlation matrix by the precision matrix, and then multiply the precision matrix by the correlation matrix.

```
#as expected, these both produce the identity matrix
round(correlationMatrix %*% inverse_correlationMatrix)
```

```
##           [,1] [,2] [,3]
## LotArea      1    0    0
## X1stFlrSF    0    1    0
## X2ndFlrSF    0    0    1
```

```
round(inverse_correlationMatrix %*%  correlationMatrix)
```

```
##      LotArea X1stFlrSF X2ndFlrSF
## [1,]       1         0         0
## [2,]       0         1         0
## [3,]       0         0         1
```

Calculus-Based Probability & Statistics.

Many times, it makes sense to fit a closed form distribution to data. For your non-transformed independent variable, location shift (if necessary) it so that the minimum value is above zero.

```
#no need to shift since minimum value is above zero
min(df$X)
```

```
## [1] 334
```

Then load the MASS package and run fitdistr to fit a density function of your choice. (See https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/fitdistr.html).

```
fittedDistribution <- fitdistr(df$X, "normal")
fittedDistribution
```

```
##       mean          sd
##   1515.46370    525.30039
##  (  13.74774) (   9.72112)
```

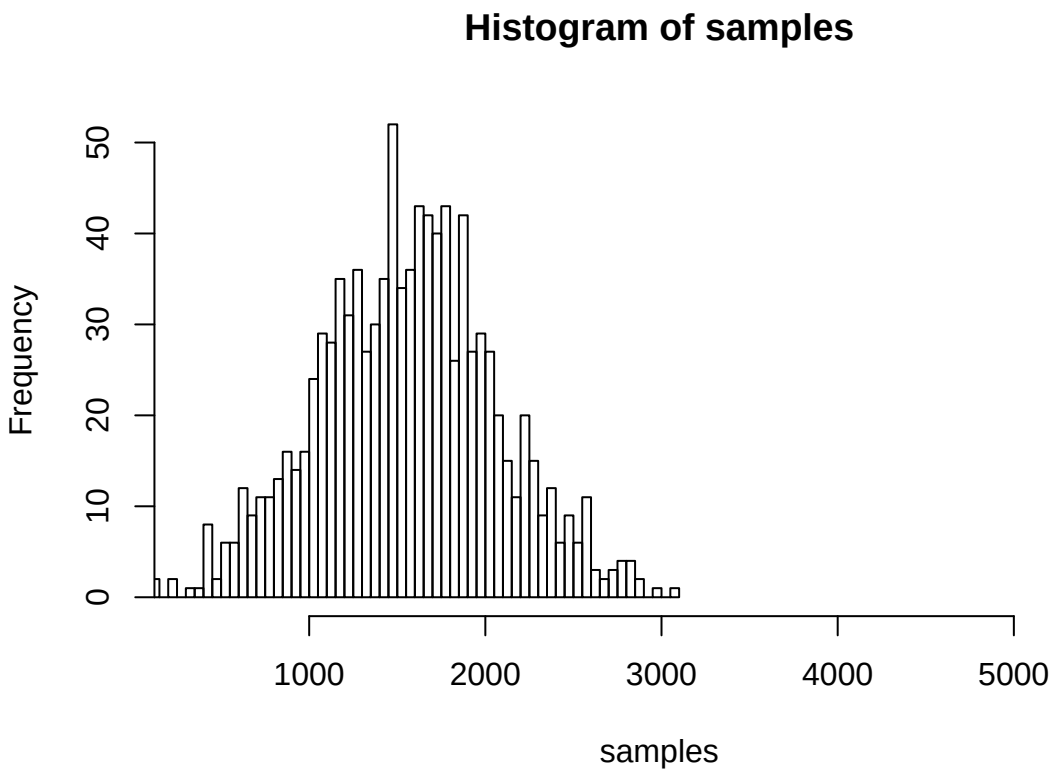
Find the optimal value of the parameters for this distribution, and then take 1000 samples from this distribution (e.g., rexp(1000,λ) for an exponential).

```
samples<-rnorm(1000,mean = fittedDistribution$estimate[1], sd = fittedDistribution$estimate[2])
```

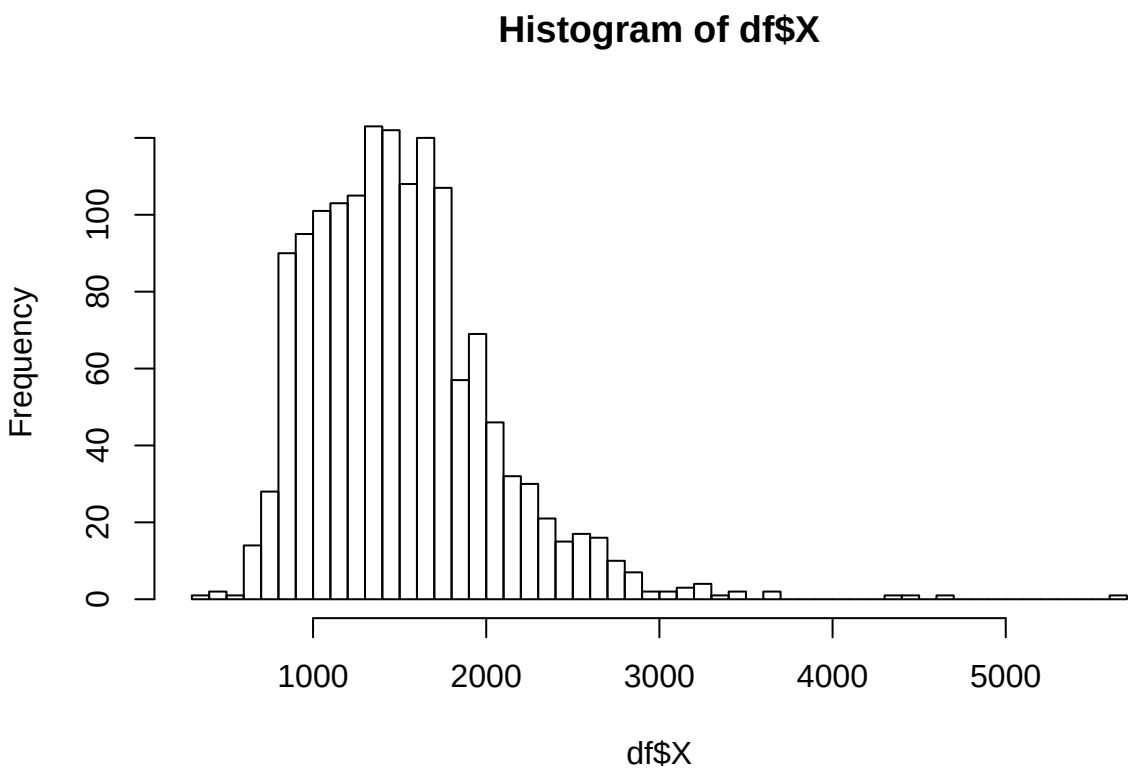
Plot a histogram and compare it with a histogram of your non-transformed original variable.

- Answer: the sample histogram looks normal, which makes sense since it comes from the normal distribution. However, it differs from the non-transformed original variable histogram, which is not normally distributed. This distribution has a right skew and a much lower median value, even though the mean and sd are roughly the same. So it would not be appropriate to use the normal distribution to model the non-transformed original variable. The box-cox transformed variable is much closer to being normally distributed and thus could be approximated using the fitted distribution.

```
hist(samples, xlim=c(min(df$X),max(df$X)), breaks = 50)
```




```r
hist(df$X, breaks = 50)
```




```
#medians
median(samples)
```

```
## [1] 1558.946
```

```
median(df$X)
```

```
## [1] 1464
```

Modeling.

Build some type of regression model and submit your model to the competition board. Provide your complete model summary and results with analysis.

I began by reading in the data and doing some initial exploration. One of the first things I noticed was that there were lots of missing values in what I judged to be important columns. In order to use these in a linear model effectively, I needed to "fill in" this missing values in some way. While I could have treated each column with missing values individually, for the sake of time, I wrote code that looped through each of the columns in *train* and *test*, filling in the *NA* value with the median of the column if it was numeric or with the word "Missing" if it was categorical. I am sure this is not the best treatment for all columns, but it does a pretty good job and is time effective.

```
library(dplyr)
library(ggplot2)
library(stringr)

#get test data
test<- read.csv("C:/Users/Andy/Desktop/Personal/Learning/CUNY/DATA605/HW/FinalProject/test.csv",
```

```
                stringsAsFactors = FALSE)

### fill in missing values for both test and train
#train
for(i in 2:(ncol(train)-1)){
  if (is.character(train[,i])){
    temp<-NA
    temp <-as.character(count(train,train[,i], sort = TRUE)[1,1])
    if(is.na(temp)){
      temp <- "Missing"
    }
    train[which(is.na(train[,i])),i] <- temp

  } else if(is.numeric(train[,i])){
      train[which(is.na(train[,i])),i] <-median(train[,i], na.rm = TRUE)
  }

} #for

#test
for(i in 2:(ncol(test))){
  if (is.character(test[,i])){
    temp<-NA
    temp <-as.character(count(test,test[,i], sort = TRUE)[1,1])
    if(is.na(temp)){
      temp <- "Missing"
    }
    test[which(is.na(test[,i])),i] <- temp

  } else if(is.numeric(test[,i])){
      test[which(is.na(test[,i])),i] <-median(test[,i], na.rm = TRUE)
  }

} #for
```

Next, based on previous data exploration, I attempted to normalize *SalePrice* using the boxCox transformation. This would ensure that my model residuals would be more normally distributed, and hence, I would be meeting the assumptions necessary for using a linear model.

I also added various new features based on exploring the existing variables and modifying them in some way to be (I hoped) more effective in predicting the *SalePrice* in my model. I also did some cleanup along the way, overwriting some outliers and changing the data type in a few places to prevent these data irregularities from throwing off the model.

```
####add features  for both test and train
#box cox transformation of SalePrice  for train
train$bc_SalePrice<-boxCox(train$SalePrice, bestLambda_Y)

#yearsold
train$YearsOld <- max(train$YearBuilt) - train$YearBuilt
test$YearsOld <- max(train$YearBuilt) - test$YearBuilt
train$YearsOld_Exp <- (train$YearsOld)^(.5)
test$YearsOld_Exp <- (test$YearsOld)^(.5)

#YearsRemod
```

```
train$YearsRemod <- max(train$YearBuilt) - train$YearRemodAdd
test$YearsRemod <- max(train$YearBuilt) - test$YearRemodAdd

#GarageYrsOld
train$GarageYrsOld <- max(train$GarageYrBlt) - train$GarageYrBlt
test$GarageYrBlt[which(test$GarageYrBlt == 2207)] <- "2007" #replace error - 2207
test$GarageYrBlt <- as.numeric(test$GarageYrBlt)
test$GarageYrsOld <- max(test$GarageYrBlt) - test$GarageYrBlt
train$GarageYrsOld_Exp <- (train$GarageYrsOld)^(.3)
test$GarageYrsOld_Exp <- (test$GarageYrsOld)^(.3)

#Month Median
monthMedian <-group_by(train,MoSold) %>% summarise(median = median(bc_SalePrice))
train$MoSold_Med <-NA
test$MoSold_Med <- NA
for (i in 1:nrow(monthMedian)){
  train$MoSold_Med[which(train$MoSold == monthMedian$MoSold[i])] <- monthMedian$median[i]
  test$MoSold_Med[which(test$MoSold == monthMedian$MoSold[i])] <- monthMedian$median[i]
}

#YrSold Median
yearMedian <-group_by(train,YrSold) %>% summarise(median = median(bc_SalePrice))
train$YrSold_Med <-NA
test$YrSold_Med <- NA
for (i in 1:nrow(yearMedian)){
  train$YrSold_Med[which(train$YrSold == yearMedian$YrSold[i])] <- yearMedian$median[i]
  test$YrSold_Med[which(test$YrSold == yearMedian$YrSold[i])] <- yearMedian$median[i]
}

#LotArea_Outlier
train$LotArea_Outlier <- train$LotArea
train$LotArea_Outlier[which(train$LotArea > 50000)] <- 50000 #set outliers to max reasonable
train$LotArea_Log <- log(train$LotArea)
test$LotArea_Outlier <- test$LotArea
test$LotArea_Outlier[which(test$LotArea > 50000)] <- 50000 #set outliers to max reasonable
test$LotArea_Log <- log(test$LotArea)

#BsmtFinSF1_Outlier
train$BsmtFinSF1_Outlier <- train$BsmtFinSF1
train$BsmtFinSF1_Outlier[which(train$BsmtFinSF1_Outlier > 3000)] <-
  median(train$BsmtFinSF1_Outlier) #set outliers to median
train$BsmtFinSF1_Outlier <- (train$BsmtFinSF1_Outlier)^(1.5)
test$BsmtFinSF1_Outlier <- test$BsmtFinSF1
test$BsmtFinSF1_Outlier[which(test$BsmtFinSF1_Outlier > 3000)] <-
  median(test$BsmtFinSF1_Outlier) #set outliers to median
test$BsmtFinSF1_Outlier <- (test$BsmtFinSF1_Outlier)^(1.5)

#BsmtUnfSF_Exp
train$BsmtUnfSF_Exp<- train$BsmtUnfSF
train$BsmtUnfSF_Exp <- (train$BsmtUnfSF_Exp)^(2)
test$BsmtUnfSF_Exp<- test$BsmtUnfSF
test$BsmtUnfSF_Exp <- (test$BsmtUnfSF_Exp)^(2)
```

```
#TotalBsmtSF
train$TotalBsmtSF_Outlier <- train$TotalBsmtSF
train$TotalBsmtSF_Outlier[which(train$TotalBsmtSF_Outlier > 4000)] <-
  median(train$TotalBsmtSF_Outlier) #set outliers to median
train$TotalBsmtSF_Outlier <- (train$TotalBsmtSF_Outlier)^(1.5)
test$TotalBsmtSF_Outlier <- test$TotalBsmtSF
test$TotalBsmtSF_Outlier[which(test$TotalBsmtSF_Outlier > 4000)] <-
  median(test$TotalBsmtSF_Outlier) #set outliers to median
test$TotalBsmtSF_Outlier <- (test$TotalBsmtSF_Outlier)^(1.5)

#X2ndFlrSF
train$X2ndFlrSF_NoZero <- train$X2ndFlrSF
train$X2ndFlrSF_NoZero[which(train$X2ndFlrSF_NoZero == 0)] <- mean(train$X2ndFlrSF_NoZero[which(train$X
test$X2ndFlrSF_NoZero <- test$X2ndFlrSF
test$X2ndFlrSF_NoZero[which(test$X2ndFlrSF_NoZero == 0)] <-
  mean(test$X2ndFlrSF_NoZero[which(test$X2ndFlrSF_NoZero != 0)]) #set zeros to mean

#OpenPorchSF
train$OpenPorchSF_NoZero <- train$OpenPorchSF
train$OpenPorchSF_NoZero[which(train$OpenPorchSF_NoZero == 0)] <- median(train$OpenPorchSF_NoZero[which
test$OpenPorchSF_NoZero <- test$OpenPorchSF
test$OpenPorchSF_NoZero[which(test$OpenPorchSF_NoZero == 0)] <- median(test$OpenPorchSF_NoZero[which(te


#EnclosedPorch
train$EnclosedPorch_NoZero <- train$EnclosedPorch
train$EnclosedPorch_NoZero[which(train$EnclosedPorch_NoZero == 0)] <- median(train$EnclosedPorch_NoZero
test$EnclosedPorch_NoZero <- test$EnclosedPorch
test$EnclosedPorch_NoZero[which(test$EnclosedPorch_NoZero == 0)] <- median(test$EnclosedPorch_NoZero[wh


#ScreenPorch
train$ScreenPorch_NoZero <- train$ScreenPorch
train$ScreenPorch_NoZero[which(train$ScreenPorch_NoZero == 0)] <- median(train$ScreenPorch_NoZero[which
test$ScreenPorch_NoZero <- test$ScreenPorch
test$ScreenPorch_NoZero[which(test$ScreenPorch_NoZero == 0)] <- median(test$ScreenPorch_NoZero[which(te


#PoolExists
train$PoolExists<-0
train$PoolExists[which(train$PoolArea > 0)]<-1
test$PoolExists<-0
test$PoolExists[which(test$PoolArea > 0)]<-1

#MSSubClass
train$MSSubClass_Char <- as.character(train$MSSubClass)
test$MSSubClass_Char <- as.character(test$MSSubClass)
```

Next, I made the categorical variables into binary columns. For example, if a single column had 5 distinct categorical values in it, I made 5 new columns, one for each of the distinct values, with 1s and 0s in it. The 1s indicated that in the original column and row, the value there corresponded to the given distinct value being considered, while the 0s indicated a different value.

While lm() will automatically do this for you, and I had originally NOT done this, I added this step in order

to better automate feature selection below, as you will see.

```
###binarize variables and add into training set
#train
for(i in 2:length(train)){
  #i<-3
  if(is.character(train[,i])){
    #get distinct values
    distinct<-unique(train[,i])
    for(j in 1:length(distinct)){
      #j<-1
      train$temp <- train[,i]
      index<-which(train$temp == distinct[j])
      notIndex <-which(train$temp != distinct[j])
      train$temp[index] <-1
      train$temp[notIndex] <-0
      train$temp<-as.numeric(train$temp)
      names(train)[length(train)] <- paste0(names(train[i]),"_",distinct[j])
      #View(cbind(train[i],train[length(train)]))
    }#for
  }#if
}#for

#test
for(i in 2:length(test)){
  #i<-3
  if(is.character(test[,i])){
    #get distinct values
    distinct<-unique(test[,i])
    for(j in 1:length(distinct)){
      #j<-1
      test$temp <- test[,i]
      index<-which(test$temp == distinct[j])
      notIndex <-which(test$temp != distinct[j])
      test$temp[index] <-1
      test$temp[notIndex] <-0
      test$temp<-as.numeric(test$temp)
      names(test)[length(test)] <- paste0(names(test[i]),"_",distinct[j])
      #View(cbind(test[i],test[length(test)]))
    }#for
  }#if
}#for
```

Finally, for modeling purposes, I created two subset data sets *train_subset* and *test_subset* that only containted numeric variables, including the binarized variables I had created above to stand in for the categorical variables.

```
## create subset of train and test for modeling.  Only include numeric variables.
#train
train_subset<-c()
train_subset_names<-c()
for(i in 1:length(train)){
  if(is.numeric(train[,i])){
    train_subset<-cbind(train_subset,train[,i])
    train_subset_names<-c(train_subset_names,names(train)[i])
```

```
  }#if
}#for
train_subset<-data.frame(train_subset,stringsAsFactors = FALSE)
names(train_subset)<-train_subset_names

#test
test_subset<-c()
test_subset_names<-c()
for(i in 1:length(test)){
  if(is.numeric(test[,i])){
    test_subset<-cbind(test_subset,test[,i])
    test_subset_names<-c(test_subset_names,names(test)[i])
  }#if
}#for
test_subset<-data.frame(test_subset,stringsAsFactors = FALSE)
names(test_subset)<-test_subset_names
```

Now I could create the initial model. I included all variables from the data set excluding the *Id*, *SalePrice*, and any other variables that were linear combinations of other variables, and hence, redundant. I set the target variable to be the boxCox transformed variable *bc_SalePrice*.

The initial model had an adjusted R^2 of 0.9345, which is pretty good. But as one can see, there are lots of variables that appear to have no real significance (i.e., a high p-value).

```
#create initial model
model <- lm(bc_SalePrice ~ .
            - Id
            - SalePrice
            - TotalBsmtSF #singularities
            - GrLivArea #singularities
            - YearBuilt #singularities
            - YearRemodAdd #singularities
            - GarageYrBlt #singularities

            , data = train_subset)
summary <-summary(model)

#Adjusted R-squared:  0.9346
summary$adj.r.squared
```

```
## [1] 0.9345589
```

```
#summary
summary
```

```
##
## Call:
## lm(formula = bc_SalePrice ~ . - Id - SalePrice - TotalBsmtSF -
##     GrLivArea - YearBuilt - YearRemodAdd - GarageYrBlt, data = train_subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38401 -0.02354  0.00092  0.02617  0.32545
##
## Coefficients: (48 not defined because of singularities)
##                         Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)            1.486e+01  5.411e+00   2.747 0.006105 **
## MSSubClass             4.958e-04  4.829e-04   1.027 0.304707
## LotFrontage            1.320e-04  1.117e-04   1.181 0.237730
## LotArea                9.351e-07  3.809e-07   2.455 0.014226 *
## OverallQual            2.030e-02  2.602e-03   7.800 1.35e-14 ***
## OverallCond            2.062e-02  2.234e-03   9.229  < 2e-16 ***
## MasVnrArea             1.371e-05  1.462e-05   0.937 0.348791
## BsmtFinSF1             2.017e-04  4.393e-05   4.591 4.89e-06 ***
## BsmtFinSF2             2.035e-04  3.876e-05   5.252 1.78e-07 ***
## BsmtUnfSF              1.468e-04  3.102e-05   4.732 2.49e-06 ***
## X1stFlrSF              1.306e-04  1.394e-05   9.366  < 2e-16 ***
## X2ndFlrSF              1.214e-04  2.300e-05   5.279 1.54e-07 ***
## LowQualFinSF           1.017e-04  5.307e-05   1.917 0.055539 .
## BsmtFullBath           1.025e-02  5.027e-03   2.038 0.041738 *
## BsmtHalfBath           9.258e-04  7.598e-03   0.122 0.903043
## FullBath               1.257e-02  5.591e-03   2.248 0.024755 *
## HalfBath               1.409e-02  5.266e-03   2.676 0.007554 **
## BedroomAbvGr           2.069e-03  3.532e-03   0.586 0.558097
## KitchenAbvGr          -2.296e-02  1.540e-02  -1.491 0.136274
## TotRmsAbvGrd           7.959e-04  2.415e-03   0.330 0.741820
## Fireplaces             9.056e-03  6.424e-03   1.410 0.158918
## GarageCars             1.018e-02  5.511e-03   1.847 0.065052 .
## GarageArea             6.172e-05  1.946e-05   3.172 0.001552 **
## WoodDeckSF             4.621e-05  1.479e-05   3.125 0.001822 **
## OpenPorchSF            1.080e-04  6.560e-05   1.646 0.100104
## EnclosedPorch          1.045e-04  3.946e-05   2.648 0.008216 **
## X3SsnPorch             7.960e-05  5.564e-05   1.431 0.152822
## ScreenPorch            1.455e-04  3.493e-05   4.165 3.33e-05 ***
## PoolArea               7.000e-04  5.756e-04   1.216 0.224181
## MiscVal               -1.367e-05  1.582e-05  -0.865 0.387443
## MoSold                 2.561e-04  7.401e-04   0.346 0.729325
## YrSold                -3.607e-03  1.897e-03  -1.902 0.057475 .
## YearsOld              -9.428e-04  6.914e-04  -1.364 0.172948
## YearsOld_Exp          -2.472e-04  9.474e-03  -0.026 0.979187
## YearsRemod            -4.387e-04  1.420e-04  -3.090 0.002046 **
## GarageYrsOld           1.014e-03  4.069e-04   2.492 0.012844 *
## GarageYrsOld_Exp      -3.689e-02  1.713e-02  -2.154 0.031472 *
## MoSold_Med            -5.875e-02  6.102e-02  -0.963 0.335882
## YrSold_Med            -2.344e-01  2.128e-01  -1.101 0.270950
## LotArea_Outlier       -2.347e-06  1.104e-06  -2.126 0.033689 *
## LotArea_Log            6.142e-02  1.477e-02   4.158 3.45e-05 ***
## BsmtFinSF1_Outlier     1.895e-07  8.802e-07   0.215 0.829590
## BsmtUnfSF_Exp          1.595e-08  7.823e-09   2.039 0.041651 *
## TotalBsmtSF_Outlier   -2.675e-06  6.409e-07  -4.174 3.22e-05 ***
## X2ndFlrSF_NoZero      -5.129e-06  2.384e-05  -0.215 0.829691
## OpenPorchSF_NoZero    -1.104e-04  7.917e-05  -1.394 0.163513
## EnclosedPorch_NoZero  -1.089e-04  7.752e-05  -1.405 0.160339
## ScreenPorch_NoZero     5.763e-05  9.155e-05   0.629 0.529180
## PoolExists            -3.183e-01  3.721e-01  -0.855 0.392557
## MSZoning_RL           -4.626e-03  1.687e-02  -0.274 0.783989
## MSZoning_RM           -2.140e-02  1.920e-02  -1.114 0.265291
## `MSZoning_C (all)`    -2.639e-01  3.019e-02  -8.742  < 2e-16 ***
## MSZoning_FV            1.942e-02  2.373e-02   0.818 0.413363
## MSZoning_RH                   NA         NA      NA       NA
```

```
## Street_Pave            5.250e-02  3.117e-02   1.685 0.092344 .
## Street_Grvl                   NA         NA      NA       NA
## Alley_Missing         -2.267e-02  1.233e-02  -1.839 0.066170 .
## Alley_Grvl            -2.119e-02  1.556e-02  -1.362 0.173528
## Alley_Pave                    NA         NA      NA       NA
## LotShape_Reg          -8.056e-03  2.217e-02  -0.363 0.716395
## LotShape_IR1          -1.475e-02  2.205e-02  -0.669 0.503563
## LotShape_IR2          -4.407e-03  2.345e-02  -0.188 0.850937
## LotShape_IR3                  NA         NA      NA       NA
## LandContour_Lvl       -1.280e-03  1.022e-02  -0.125 0.900319
## LandContour_Bnk       -1.255e-02  1.293e-02  -0.970 0.332216
## LandContour_Low       -2.904e-02  1.561e-02  -1.861 0.063011 .
## LandContour_HLS               NA         NA      NA       NA
## Utilities_AllPub       1.538e-01  7.096e-02   2.167 0.030447 *
## Utilities_NoSeWa              NA         NA      NA       NA
## LotConfig_Inside       3.986e-02  3.120e-02   1.278 0.201666
## LotConfig_FR2          2.648e-02  3.214e-02   0.824 0.410018
## LotConfig_Corner       4.608e-02  3.141e-02   1.467 0.142636
## LotConfig_CulDSac      6.166e-02  3.207e-02   1.923 0.054738 .
## LotConfig_FR3                 NA         NA      NA       NA
## LandSlope_Gtl          8.360e-02  2.954e-02   2.830 0.004736 **
## LandSlope_Mod          9.945e-02  2.956e-02   3.364 0.000791 ***
## LandSlope_Sev                 NA         NA      NA       NA
## Neighborhood_CollgCr  -7.077e-02  4.764e-02  -1.486 0.137669
## Neighborhood_Veenker  -4.314e-02  5.068e-02  -0.851 0.394828
## Neighborhood_Crawfor  -7.305e-03  4.814e-02  -0.152 0.879428
## Neighborhood_NoRidge  -3.549e-02  4.898e-02  -0.724 0.468919
## Neighborhood_Mitchel  -9.257e-02  4.784e-02  -1.935 0.053241 .
## Neighborhood_Somerst  -5.476e-02  4.974e-02  -1.101 0.271217
## Neighborhood_NWAmes   -8.181e-02  4.727e-02  -1.731 0.083744 .
## Neighborhood_OldTown  -8.422e-02  4.751e-02  -1.773 0.076506 .
## Neighborhood_BrkSide  -5.454e-02  4.812e-02  -1.133 0.257244
## Neighborhood_Sawyer   -7.983e-02  4.767e-02  -1.675 0.094267 .
## Neighborhood_NridgHt  -2.576e-02  4.844e-02  -0.532 0.595054
## Neighborhood_NAmes    -8.443e-02  4.733e-02  -1.784 0.074702 .
## Neighborhood_SawyerW  -6.375e-02  4.774e-02  -1.335 0.182048
## Neighborhood_IDOTRR   -7.467e-02  4.888e-02  -1.528 0.126849
## Neighborhood_MeadowV  -1.148e-01  4.796e-02  -2.393 0.016863 *
## Neighborhood_Edwards  -1.083e-01  4.737e-02  -2.287 0.022381 *
## Neighborhood_Timber   -6.090e-02  4.849e-02  -1.256 0.209354
## Neighborhood_Gilbert  -6.569e-02  4.783e-02  -1.373 0.169871
## Neighborhood_StoneBr   1.558e-02  4.899e-02   0.318 0.750511
## Neighborhood_ClearCr  -3.901e-02  4.901e-02  -0.796 0.426252
## Neighborhood_NPkVill  -5.022e-02  5.542e-02  -0.906 0.365039
## Neighborhood_Blmngtn  -4.359e-02  4.989e-02  -0.874 0.382437
## Neighborhood_BrDale   -4.331e-02  4.689e-02  -0.924 0.355804
## Neighborhood_SWISU    -6.011e-02  4.978e-02  -1.208 0.227463
## Neighborhood_Blueste          NA         NA      NA       NA
## Condition1_Norm        2.982e-02  4.211e-02   0.708 0.479029
## Condition1_Feedr       4.768e-03  4.277e-02   0.111 0.911255
## Condition1_PosN        4.159e-02  4.459e-02   0.933 0.351071
## Condition1_Artery     -1.817e-02  4.349e-02  -0.418 0.676143
## Condition1_RRAe       -5.119e-02  4.608e-02  -1.111 0.266788
## Condition1_RRNn        2.785e-02  5.161e-02   0.540 0.589618
```

```
## Condition1_RRAn        1.024e-02  4.435e-02   0.231 0.817394
## Condition1_PosA        6.650e-03  4.784e-02   0.139 0.889462
## Condition1_RRNe               NA         NA      NA       NA
## Condition2_Norm        4.228e-01  1.719e-01   2.460 0.014051 *
## Condition2_Artery      3.304e-01  1.813e-01   1.823 0.068587 .
## Condition2_RRNn        4.059e-01  1.770e-01   2.293 0.022014 *
## Condition2_Feedr       4.499e-01  1.783e-01   2.523 0.011751 *
## Condition2_PosN        3.139e-03  1.783e-01   0.018 0.985954
## Condition2_PosA        5.490e-01  1.907e-01   2.879 0.004056 **
## Condition2_RRAn        3.697e-01  1.820e-01   2.031 0.042434 *
## Condition2_RRAe               NA         NA      NA       NA
## BldgType_1Fam          1.887e-02  3.914e-02   0.482 0.629844
## BldgType_2fmCon       -2.187e-02  7.855e-02  -0.278 0.780751
## BldgType_Duplex        2.981e-02  4.364e-02   0.683 0.494675
## BldgType_TwnhsE       -1.015e-03  1.333e-02  -0.076 0.939347
## BldgType_Twnhs                NA         NA      NA       NA
## HouseStyle_2Story      2.539e-02  4.226e-02   0.601 0.548121
## HouseStyle_1Story      9.039e-03  5.090e-02   0.178 0.859083
## HouseStyle_1.5Fin      2.075e-02  4.501e-02   0.461 0.644845
## HouseStyle_1.5Unf      1.356e-01  7.843e-02   1.729 0.083997 .
## HouseStyle_SFoyer     -3.934e-04  5.549e-02  -0.007 0.994343
## HouseStyle_SLvl        3.443e-02  5.605e-02   0.614 0.539168
## HouseStyle_2.5Unf      7.698e-02  3.848e-02   2.000 0.045685 *
## HouseStyle_2.5Fin             NA         NA      NA       NA
## RoofStyle_Gable       -2.336e-01  8.723e-02  -2.678 0.007514 **
## RoofStyle_Hip         -2.328e-01  8.730e-02  -2.667 0.007766 **
## RoofStyle_Gambrel     -2.539e-01  8.944e-02  -2.839 0.004606 **
## RoofStyle_Mansard     -2.101e-01  8.680e-02  -2.421 0.015636 *
## RoofStyle_Flat        -2.465e-01  9.631e-02  -2.560 0.010605 *
## RoofStyle_Shed                NA         NA      NA       NA
## RoofMatl_CompShg       2.039e+00  2.677e-01   7.618 5.26e-14 ***
## RoofMatl_WdShngl       2.092e+00  2.700e-01   7.748 1.99e-14 ***
## RoofMatl_Metal         2.142e+00  2.827e-01   7.578 7.04e-14 ***
## RoofMatl_WdShake       2.009e+00  2.703e-01   7.432 2.04e-13 ***
## RoofMatl_Membran       2.257e+00  2.812e-01   8.026 2.41e-15 ***
## `RoofMatl_Tar&Grv`     2.058e+00  2.726e-01   7.552 8.53e-14 ***
## RoofMatl_Roll          2.034e+00  2.764e-01   7.361 3.41e-13 ***
## RoofMatl_ClyTile              NA         NA      NA       NA
## Exterior1st_VinylSd   -3.018e-03  8.968e-02  -0.034 0.973158
## Exterior1st_MetalSd    1.238e-02  9.419e-02   0.131 0.895457
## `Exterior1st_Wd Sdng` -3.955e-02  9.200e-02  -0.430 0.667366
## Exterior1st_HdBoard   -1.509e-02  9.260e-02  -0.163 0.870544
## Exterior1st_BrkFace    3.366e-02  9.228e-02   0.365 0.715332
## Exterior1st_WdShing   -7.426e-03  9.269e-02  -0.080 0.936156
## Exterior1st_CemntBd   -5.850e-02  1.002e-01  -0.584 0.559343
## Exterior1st_Plywood   -1.302e-02  9.254e-02  -0.141 0.888122
## Exterior1st_AsbShng    9.269e-03  9.753e-02   0.095 0.924304
## Exterior1st_Stucco    -2.727e-03  9.436e-02  -0.029 0.976954
## Exterior1st_BrkComm   -1.786e-01  1.124e-01  -1.588 0.112576
## Exterior1st_AsphShn   -1.237e-02  1.206e-01  -0.103 0.918268
## Exterior1st_Stone     -1.328e-02  1.042e-01  -0.127 0.898579
## Exterior1st_ImStucc   -2.456e-02  1.113e-01  -0.221 0.825361
## Exterior1st_CBlock            NA         NA      NA       NA
## Exterior2nd_VinylSd    5.621e-02  5.960e-02   0.943 0.345734
```

```
## Exterior2nd_MetalSd    4.983e-02  6.609e-02   0.754 0.451082
## `Exterior2nd_Wd Shng`  6.173e-02  6.326e-02   0.976 0.329324
## Exterior2nd_HdBoard    6.176e-02  6.345e-02   0.973 0.330557
## Exterior2nd_Plywood    6.114e-02  6.315e-02   0.968 0.333191
## `Exterior2nd_Wd Sdng`  8.588e-02  6.309e-02   1.361 0.173701
## Exterior2nd_CmentBd    1.220e-01  7.368e-02   1.656 0.097929 .
## Exterior2nd_BrkFace    3.822e-02  6.506e-02   0.588 0.556975
## Exterior2nd_Stucco     6.534e-02  6.543e-02   0.999 0.318213
## Exterior2nd_AsbShng    2.696e-02  6.807e-02   0.396 0.692064
## `Exterior2nd_Brk Cmn`  1.237e-01  7.625e-02   1.623 0.104947
## Exterior2nd_ImStucc    6.902e-02  6.593e-02   1.047 0.295345
## Exterior2nd_AsphShn    9.253e-02  7.846e-02   1.179 0.238481
## Exterior2nd_Stone      4.961e-02  6.965e-02   0.712 0.476444
## Exterior2nd_Other             NA         NA      NA       NA
## Exterior2nd_CBlock            NA         NA      NA       NA
## MasVnrType_BrkFace     1.934e-02  1.705e-02   1.134 0.257020
## MasVnrType_None        1.403e-02  1.723e-02   0.815 0.415487
## MasVnrType_Stone       2.246e-02  1.814e-02   1.238 0.215786
## MasVnrType_BrkCmn             NA         NA      NA       NA
## ExterQual_Gd          -1.642e-02  2.792e-02  -0.588 0.556663
## ExterQual_TA          -1.516e-02  2.729e-02  -0.556 0.578636
## ExterQual_Ex          -2.080e-02  3.046e-02  -0.683 0.494935
## ExterQual_Fa                  NA         NA      NA       NA
## ExterCond_TA          -1.639e-02  4.281e-02  -0.383 0.701926
## ExterCond_Gd          -2.616e-02  4.288e-02  -0.610 0.541853
## ExterCond_Fa          -3.894e-02  4.495e-02  -0.866 0.386607
## ExterCond_Po           2.368e-02  7.935e-02   0.298 0.765453
## ExterCond_Ex                  NA         NA      NA       NA
## Foundation_PConc      -3.616e-02  2.889e-02  -1.252 0.210958
## Foundation_CBlock     -4.728e-02  2.895e-02  -1.633 0.102675
## Foundation_BrkTil     -5.912e-02  2.877e-02  -2.055 0.040073 *
## Foundation_Wood       -1.234e-01  4.588e-02  -2.690 0.007256 **
## Foundation_Slab       -3.764e-02  3.444e-02  -1.093 0.274581
## Foundation_Stone              NA         NA      NA       NA
## BsmtQual_Gd           -9.089e-03  1.384e-02  -0.657 0.511612
## BsmtQual_TA           -5.917e-03  1.250e-02  -0.473 0.636147
## BsmtQual_Ex            6.428e-03  1.611e-02   0.399 0.690016
## BsmtQual_Fa                   NA         NA      NA       NA
## BsmtCond_TA           -1.097e-01  7.581e-02  -1.446 0.148323
## BsmtCond_Gd           -1.112e-01  7.624e-02  -1.458 0.145016
## BsmtCond_Fa           -1.254e-01  7.533e-02  -1.665 0.096134 .
## BsmtCond_Po                   NA         NA      NA       NA
## BsmtExposure_No       -2.716e-03  5.509e-03  -0.493 0.622052
## BsmtExposure_Gd        1.805e-02  7.602e-03   2.374 0.017732 *
## BsmtExposure_Mn       -2.769e-03  7.574e-03  -0.366 0.714719
## BsmtExposure_Av               NA         NA      NA       NA
## BsmtFinType1_GLQ       2.326e-02  9.538e-03   2.439 0.014882 *
## BsmtFinType1_ALQ       1.790e-02  9.392e-03   1.905 0.056958 .
## BsmtFinType1_Unf       1.001e-02  1.096e-02   0.913 0.361266
## BsmtFinType1_Rec       1.119e-02  9.530e-03   1.174 0.240683
## BsmtFinType1_BLQ       1.466e-02  9.521e-03   1.539 0.123953
## BsmtFinType1_LwQ              NA         NA      NA       NA
## BsmtFinType2_Unf      -5.909e-03  2.209e-02  -0.267 0.789151
## BsmtFinType2_BLQ      -3.386e-02  2.199e-02  -1.540 0.123866
```

```
## BsmtFinType2_ALQ       1.195e-02  2.337e-02   0.511 0.609272
## BsmtFinType2_Rec      -1.323e-02  2.119e-02  -0.624 0.532617
## BsmtFinType2_LwQ      -1.441e-02  2.193e-02  -0.657 0.511233
## BsmtFinType2_GLQ              NA         NA      NA       NA
## Heating_GasA           3.109e-02  6.193e-02   0.502 0.615754
## Heating_GasW           6.133e-02  6.408e-02   0.957 0.338768
## Heating_Grav          -6.855e-02  6.835e-02  -1.003 0.316125
## Heating_Wall           6.518e-02  7.219e-02   0.903 0.366740
## Heating_OthW           3.591e-02  7.746e-02   0.464 0.643011
## Heating_Floor                 NA         NA      NA       NA
## HeatingQC_Ex           1.661e-02  6.662e-02   0.249 0.803146
## HeatingQC_Gd           5.649e-03  6.670e-02   0.085 0.932516
## HeatingQC_TA          -8.245e-04  6.658e-02  -0.012 0.990121
## HeatingQC_Fa           1.318e-03  6.742e-02   0.020 0.984411
## HeatingQC_Po                  NA         NA      NA       NA
## CentralAir_Y           3.520e-02  9.763e-03   3.605 0.000325 ***
## CentralAir_N                  NA         NA      NA       NA
## Electrical_SBrkr       8.001e-02  1.117e-01   0.716 0.474056
## Electrical_FuseF       8.146e-02  1.130e-01   0.721 0.471077
## Electrical_FuseA       9.228e-02  1.115e-01   0.828 0.408036
## Electrical_FuseP       4.909e-02  1.138e-01   0.431 0.666262
## Electrical_Mix                NA         NA      NA       NA
## KitchenQual_Gd        -4.540e-03  1.340e-02  -0.339 0.734824
## KitchenQual_TA        -4.227e-03  1.235e-02  -0.342 0.732172
## KitchenQual_Ex         2.938e-02  1.554e-02   1.891 0.058827 .
## KitchenQual_Fa                NA         NA      NA       NA
## Functional_Typ         1.878e-01  7.208e-02   2.606 0.009286 **
## Functional_Min1        1.783e-01  7.231e-02   2.465 0.013829 *
## Functional_Maj1        1.653e-01  7.414e-02   2.229 0.025978 *
## Functional_Min2        1.642e-01  7.321e-02   2.242 0.025129 *
## Functional_Mod         1.210e-01  7.453e-02   1.624 0.104631
## Functional_Maj2        2.953e-02  7.852e-02   0.376 0.706952
## Functional_Sev                NA         NA      NA       NA
## FireplaceQu_Missing   -1.179e-02  1.596e-02  -0.739 0.460120
## FireplaceQu_TA        -4.237e-03  1.513e-02  -0.280 0.779557
## FireplaceQu_Gd        -5.417e-03  1.506e-02  -0.360 0.719173
## FireplaceQu_Fa        -1.636e-02  1.801e-02  -0.908 0.364011
## FireplaceQu_Ex        -1.444e-02  1.976e-02  -0.731 0.465075
## FireplaceQu_Po                NA         NA      NA       NA
## GarageType_Attchd      6.295e-02  2.758e-02   2.283 0.022618 *
## GarageType_Detchd      6.843e-02  2.745e-02   2.493 0.012815 *
## GarageType_BuiltIn     6.004e-02  2.874e-02   2.089 0.036906 *
## GarageType_CarPort     7.530e-02  3.704e-02   2.033 0.042315 *
## GarageType_Basment     6.188e-02  3.212e-02   1.927 0.054279 .
## GarageType_2Types             NA         NA      NA       NA
## GarageFinish_RFn      -7.814e-04  4.895e-03  -0.160 0.873188
## GarageFinish_Unf      -7.943e-03  6.051e-03  -1.313 0.189561
## GarageFinish_Fin              NA         NA      NA       NA
## GarageQual_TA          2.873e-02  6.509e-02   0.441 0.658997
## GarageQual_Fa          5.494e-03  6.391e-02   0.086 0.931501
## GarageQual_Gd          2.920e-02  6.881e-02   0.424 0.671355
## GarageQual_Ex          1.713e-01  1.003e-01   1.707 0.088047 .
## GarageQual_Po                 NA         NA      NA       NA
## GarageCond_TA          1.268e-01  8.815e-02   1.438 0.150719
```

```
## GarageCond_Fa          1.091e-01  8.882e-02   1.229 0.219441
## GarageCond_Gd          1.352e-01  9.214e-02   1.467 0.142627
## GarageCond_Po          1.624e-01  9.670e-02   1.679 0.093387 .
## GarageCond_Ex                 NA         NA      NA       NA
## PavedDrive_Y           1.404e-02  1.221e-02   1.150 0.250455
## PavedDrive_N           5.752e-03  1.378e-02   0.417 0.676521
## PavedDrive_P                  NA         NA      NA       NA
## PoolQC_Missing                NA         NA      NA       NA
## PoolQC_Ex             -1.371e-02  9.324e-02  -0.147 0.883121
## PoolQC_Fa             -1.023e-01  7.691e-02  -1.330 0.183819
## PoolQC_Gd                     NA         NA      NA       NA
## Fence_Missing          9.936e-03  1.864e-02   0.533 0.594172
## Fence_MnPrv            6.188e-03  1.906e-02   0.325 0.745500
## Fence_GdWo            -1.346e-02  2.013e-02  -0.669 0.503916
## Fence_GdPrv            2.994e-03  2.050e-02   0.146 0.883903
## Fence_MnWw                    NA         NA      NA       NA
## MiscFeature_Missing    7.426e-03  1.197e-01   0.062 0.950547
## MiscFeature_Shed       1.519e-02  1.178e-01   0.129 0.897453
## MiscFeature_Gar2       2.378e-01  2.485e-01   0.957 0.338737
## MiscFeature_Othr       7.701e-03  1.283e-01   0.060 0.952143
## MiscFeature_TenC              NA         NA      NA       NA
## SaleType_WD           -3.346e-02  3.519e-02  -0.951 0.341882
## SaleType_New           2.123e-02  5.117e-02   0.415 0.678269
## SaleType_COD          -2.372e-02  3.609e-02  -0.657 0.511149
## SaleType_ConLD         4.952e-02  4.151e-02   1.193 0.233116
## SaleType_ConLI        -4.562e-02  4.371e-02  -1.044 0.296892
## SaleType_CWD           5.655e-03  4.614e-02   0.123 0.902481
## SaleType_ConLw        -2.082e-02  4.541e-02  -0.459 0.646600
## SaleType_Con           1.231e-02  5.499e-02   0.224 0.822904
## SaleType_Oth                  NA         NA      NA       NA
## SaleCondition_Normal   3.232e-02  1.382e-02   2.339 0.019485 *
## SaleCondition_Abnorml -2.238e-03  1.518e-02  -0.147 0.882781
## SaleCondition_Partial -1.188e-02  3.914e-02  -0.304 0.761477
## SaleCondition_AdjLand  7.084e-02  3.851e-02   1.839 0.066131 .
## SaleCondition_Alloca   2.796e-02  2.540e-02   1.100 0.271369
## SaleCondition_Family          NA         NA      NA       NA
## MSSubClass_Char_60     8.871e-03  4.393e-02   0.202 0.840018
## MSSubClass_Char_20     5.684e-02  5.175e-02   1.098 0.272256
## MSSubClass_Char_70     3.007e-02  4.082e-02   0.737 0.461458
## MSSubClass_Char_50     3.593e-02  4.595e-02   0.782 0.434367
## MSSubClass_Char_190   -7.200e-03  8.780e-02  -0.082 0.934657
## MSSubClass_Char_45    -8.116e-02  7.593e-02  -1.069 0.285334
## MSSubClass_Char_90            NA         NA      NA       NA
## MSSubClass_Char_120    1.777e-02  3.129e-02   0.568 0.570114
## MSSubClass_Char_30     1.765e-02  4.842e-02   0.365 0.715522
## MSSubClass_Char_85     2.933e-02  4.052e-02   0.724 0.469234
## MSSubClass_Char_80     2.282e-03  4.455e-02   0.051 0.959162
## MSSubClass_Char_160   -4.660e-02  3.295e-02  -1.414 0.157652
## MSSubClass_Char_75    -1.685e-02  5.201e-02  -0.324 0.746022
## MSSubClass_Char_180           NA         NA      NA       NA
## MSSubClass_Char_40            NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.05597 on 1187 degrees of freedom
## Multiple R-squared:  0.9468, Adjusted R-squared:  0.9346
## F-statistic:  77.6 on 272 and 1187 DF,  p-value: < 2.2e-16
```

In order to eliminate the high p-values, I set up a loop to automatically remove the variables that had the highest p-values. I started with the variable that had the highest p-value (least significant), and worked my way down to the lowest p-value until all variables were under my desired threshold. I could not do this if I was using categorical variables as originally given. Some values in a categorical variable were significant while others were not, and to remove those that were not would have required removing the whole categorical variable. By splitting them up into separate fields, I could remove the insignificant values for a categorical variable while keeping those that were significant.

Interestingly, the adjusted R^2 value did not change very much, becoming 0.9352 with a p-value threshold of under 0.05. However, one can see that the the number of variables is greatly reduced.

```
##loop through variables to eliminate high p-values
model2<-model
summary2 <-summary

while(sort(summary2$coefficients[,4], decreasing = TRUE)[1]>0.05) {
  #update model by removing highest p-value until threshold reached
  name <-names(sort(summary2$coefficients[,4], decreasing = TRUE)[1])
  model2<- update(model2, as.formula(paste0(". ~ . -",name)))
  summary2<-summary(model2)
}

#p-value limit: 0.10 - Adjusted R-squared:  0.9372
#p-value limit: 0.05 - Adjusted R-squared:  0.9352
summary2$adj.r.squared
```

```
## [1] 0.9352582
```

```
#summary2
summary2
```

```
##
## Call:
## lm(formula = bc_SalePrice ~ LotArea + OverallQual + OverallCond +
##     BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF +
##     LowQualFinSF + BsmtFullBath + FullBath + HalfBath + Fireplaces +
##     GarageCars + GarageArea + WoodDeckSF + EnclosedPorch + ScreenPorch +
##     PoolArea + YearsOld + YearsRemod + GarageYrsOld + GarageYrsOld_Exp +
##     LotArea_Outlier + LotArea_Log + BsmtUnfSF_Exp + TotalBsmtSF_Outlier +
##     PoolExists + MSZoning_RM + `MSZoning_C (all)` + LotConfig_CulDSac +
##     LandSlope_Gtl + LandSlope_Mod + LandSlope_Sev + Neighborhood_CollgCr +
##     Neighborhood_Mitchel + Neighborhood_NWAmes + Neighborhood_OldTown +
##     Neighborhood_BrkSide + Neighborhood_Sawyer + Neighborhood_NAmes +
##     Neighborhood_SawyerW + Neighborhood_IDOTRR + Neighborhood_MeadowV +
##     Neighborhood_Edwards + Neighborhood_Timber + Neighborhood_Gilbert +
##     Neighborhood_SWISU + Condition1_Norm + Condition1_PosN +
##     Condition1_Artery + Condition1_RRAe + Condition2_Norm + Condition2_Artery +
##     Condition2_RRNn + Condition2_Feedr + Condition2_PosA + Condition2_RRAn +
##     HouseStyle_1.5Unf + RoofStyle_Gable + RoofStyle_Hip + RoofStyle_Gambrel +
##     RoofStyle_Mansard + RoofStyle_Flat + RoofStyle_Shed + RoofMatl_CompShg +
##     RoofMatl_WdShngl + RoofMatl_Metal + RoofMatl_WdShake + RoofMatl_Membran +
##     `RoofMatl_Tar&Grv` + RoofMatl_Roll + RoofMatl_ClyTile + Exterior1st_MetalSd +
##     `Exterior1st_Wd Sdng` + Exterior1st_BrkFace + Exterior1st_BrkComm +
```

```
##     `Exterior2nd_Wd Sdng` + Foundation_CBlock + Foundation_BrkTil +
##     Foundation_Wood + BsmtQual_Gd + BsmtQual_TA + BsmtCond_Fa +
##     BsmtExposure_Gd + BsmtFinType2_BLQ + Heating_GasW + Heating_Grav +
##     HeatingQC_Ex + CentralAir_Y + CentralAir_N + KitchenQual_Ex +
##     Functional_Typ + Functional_Min1 + Functional_Maj1 + Functional_Min2 +
##     Functional_Mod + GarageType_2Types + GarageFinish_Unf + GarageCond_TA +
##     PoolQC_Missing + PoolQC_Fa + SaleType_WD + SaleType_COD +
##     SaleType_ConLI + SaleCondition_Normal + SaleCondition_AdjLand +
##     MSSubClass_Char_20 + MSSubClass_Char_70 + MSSubClass_Char_50 +
##     MSSubClass_Char_45, data = train_subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39122 -0.02562  0.00169  0.02836  0.30049
##
## Coefficients: (5 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             5.344e+00  1.742e-01  30.671  < 2e-16 ***
## LotArea                 6.853e-07  3.278e-07   2.091 0.036724 *
## OverallQual             2.347e-02  2.216e-03  10.588  < 2e-16 ***
## OverallCond             2.117e-02  1.886e-03  11.229  < 2e-16 ***
## BsmtFinSF1              2.272e-04  2.390e-05   9.505  < 2e-16 ***
## BsmtFinSF2              2.135e-04  2.527e-05   8.450  < 2e-16 ***
## BsmtUnfSF               1.647e-04  2.247e-05   7.330 3.94e-13 ***
## X1stFlrSF               1.476e-04  1.004e-05  14.696  < 2e-16 ***
## X2ndFlrSF               1.289e-04  7.365e-06  17.504  < 2e-16 ***
## LowQualFinSF            1.195e-04  3.427e-05   3.486 0.000506 ***
## BsmtFullBath            1.388e-02  4.154e-03   3.342 0.000856 ***
## FullBath                1.025e-02  4.830e-03   2.122 0.034017 *
## HalfBath                9.918e-03  4.554e-03   2.178 0.029567 *
## Fireplaces              1.372e-02  3.038e-03   4.516 6.84e-06 ***
## GarageCars              1.422e-02  4.975e-03   2.859 0.004320 **
## GarageArea              5.588e-05  1.739e-05   3.213 0.001343 **
## WoodDeckSF              3.224e-05  1.353e-05   2.384 0.017272 *
## EnclosedPorch           9.233e-05  2.781e-05   3.320 0.000924 ***
## ScreenPorch             1.476e-04  2.854e-05   5.173 2.65e-07 ***
## PoolArea                1.102e-03  3.138e-04   3.513 0.000457 ***
## YearsOld               -1.037e-03  1.632e-04  -6.355 2.85e-10 ***
## YearsRemod             -4.421e-04  1.236e-04  -3.578 0.000358 ***
## GarageYrsOld            1.164e-03  2.730e-04   4.263 2.16e-05 ***
## GarageYrsOld_Exp       -4.356e-02  1.032e-02  -4.222 2.58e-05 ***
## LotArea_Outlier        -3.329e-06  8.435e-07  -3.947 8.34e-05 ***
## LotArea_Log             7.243e-02  8.867e-03   8.169 7.07e-16 ***
## BsmtUnfSF_Exp           1.393e-08  7.002e-09   1.989 0.046936 *
## TotalBsmtSF_Outlier    -3.142e-06  4.997e-07  -6.288 4.33e-10 ***
## PoolExists             -5.619e-01  1.889e-01  -2.975 0.002981 **
## MSZoning_RM            -1.914e-02  7.682e-03  -2.491 0.012845 *
## `MSZoning_C (all)`     -2.453e-01  2.219e-02 -11.055  < 2e-16 ***
## LotConfig_CulDSac       1.800e-02  6.485e-03   2.775 0.005593 **
## LandSlope_Gtl           8.231e-02  2.399e-02   3.431 0.000620 ***
## LandSlope_Mod           9.651e-02  2.440e-02   3.955 8.05e-05 ***
## LandSlope_Sev                  NA         NA      NA       NA
## Neighborhood_CollgCr   -3.935e-02  6.369e-03  -6.179 8.50e-10 ***
## Neighborhood_Mitchel   -6.706e-02  9.798e-03  -6.844 1.16e-11 ***
```

```
## Neighborhood_NWAmes   -4.793e-02  8.799e-03  -5.447 6.07e-08 ***
## Neighborhood_OldTown  -5.806e-02  1.092e-02  -5.319 1.22e-07 ***
## Neighborhood_BrkSide  -2.674e-02  1.112e-02  -2.404 0.016349 *
## Neighborhood_Sawyer   -4.948e-02  9.045e-03  -5.470 5.36e-08 ***
## Neighborhood_NAmes    -4.951e-02  7.048e-03  -7.025 3.39e-12 ***
## Neighborhood_SawyerW  -3.693e-02  8.729e-03  -4.231 2.49e-05 ***
## Neighborhood_IDOTRR   -4.884e-02  1.482e-02  -3.295 0.001008 **
## Neighborhood_MeadowV  -6.507e-02  1.644e-02  -3.957 7.98e-05 ***
## Neighborhood_Edwards  -7.498e-02  8.213e-03  -9.129  < 2e-16 ***
## Neighborhood_Timber   -3.377e-02  1.032e-02  -3.271 0.001098 **
## Neighborhood_Gilbert  -3.256e-02  8.280e-03  -3.933 8.83e-05 ***
## Neighborhood_SWISU    -4.131e-02  1.387e-02  -2.979 0.002943 **
## Condition1_Norm        2.427e-02  5.833e-03   4.160 3.38e-05 ***
## Condition1_PosN        3.439e-02  1.519e-02   2.264 0.023748 *
## Condition1_Artery     -2.647e-02  1.074e-02  -2.465 0.013829 *
## Condition1_RRAe       -5.265e-02  1.873e-02  -2.811 0.005006 **
## Condition2_Norm        3.757e-01  4.060e-02   9.254  < 2e-16 ***
## Condition2_Artery      2.579e-01  6.332e-02   4.073 4.92e-05 ***
## Condition2_RRNn        3.352e-01  5.777e-02   5.802 8.16e-09 ***
## Condition2_Feedr       3.953e-01  4.750e-02   8.321  < 2e-16 ***
## Condition2_PosA        5.610e-01  7.106e-02   7.895 5.97e-15 ***
## Condition2_RRAn        3.149e-01  7.017e-02   4.488 7.80e-06 ***
## HouseStyle_1.5Unf      1.350e-01  4.953e-02   2.725 0.006505 **
## RoofStyle_Gable       -2.322e-01  4.993e-02  -4.650 3.65e-06 ***
## RoofStyle_Hip         -2.281e-01  5.001e-02  -4.560 5.57e-06 ***
## RoofStyle_Gambrel     -2.554e-01  5.302e-02  -4.818 1.61e-06 ***
## RoofStyle_Mansard     -1.960e-01  5.231e-02  -3.746 0.000187 ***
## RoofStyle_Flat        -2.495e-01  6.450e-02  -3.869 0.000115 ***
## RoofStyle_Shed                NA         NA      NA       NA
## RoofMatl_CompShg       2.116e+00  1.535e-01  13.790  < 2e-16 ***
## RoofMatl_WdShngl       2.179e+00  1.566e-01  13.918  < 2e-16 ***
## RoofMatl_Metal         2.215e+00  1.697e-01  13.050  < 2e-16 ***
## RoofMatl_WdShake       2.061e+00  1.562e-01  13.196  < 2e-16 ***
## RoofMatl_Membran       2.300e+00  1.706e-01  13.485  < 2e-16 ***
## `RoofMatl_Tar&Grv`     2.132e+00  1.580e-01  13.495  < 2e-16 ***
## RoofMatl_Roll          2.136e+00  1.639e-01  13.032  < 2e-16 ***
## RoofMatl_ClyTile              NA         NA      NA       NA
## Exterior1st_MetalSd    1.144e-02  4.783e-03   2.392 0.016889 *
## `Exterior1st_Wd Sdng` -2.728e-02  9.208e-03  -2.962 0.003106 **
## Exterior1st_BrkFace    2.963e-02  9.254e-03   3.202 0.001395 **
## Exterior1st_BrkComm   -9.073e-02  4.394e-02  -2.065 0.039119 *
## `Exterior2nd_Wd Sdng`  2.259e-02  9.099e-03   2.482 0.013172 *
## Foundation_CBlock     -1.230e-02  5.137e-03  -2.394 0.016801 *
## Foundation_BrkTil     -2.648e-02  7.676e-03  -3.449 0.000580 ***
## Foundation_Wood       -8.903e-02  3.340e-02  -2.666 0.007774 **
## BsmtQual_Gd           -1.592e-02  6.248e-03  -2.548 0.010947 *
## BsmtQual_TA           -1.765e-02  6.993e-03  -2.523 0.011734 *
## BsmtCond_Fa           -1.941e-02  9.520e-03  -2.039 0.041661 *
## BsmtExposure_Gd        2.198e-02  6.205e-03   3.543 0.000409 ***
## BsmtFinType2_BLQ      -2.586e-02  1.062e-02  -2.436 0.014972 *
## Heating_GasW           3.689e-02  1.519e-02   2.428 0.015321 *
## Heating_Grav          -9.577e-02  2.398e-02  -3.994 6.86e-05 ***
## HeatingQC_Ex           1.425e-02  4.006e-03   3.557 0.000389 ***
## CentralAir_Y           3.162e-02  7.893e-03   4.006 6.51e-05 ***
```

```
## CentralAir_N                   NA         NA      NA       NA
## KitchenQual_Ex          3.355e-02  7.403e-03   4.532 6.37e-06 ***
## Functional_Typ          1.695e-01  2.423e-02   6.997 4.11e-12 ***
## Functional_Min1         1.477e-01  2.609e-02   5.661 1.83e-08 ***
## Functional_Maj1         1.357e-01  2.880e-02   4.713 2.69e-06 ***
## Functional_Min2         1.435e-01  2.619e-02   5.480 5.08e-08 ***
## Functional_Mod          1.024e-01  2.905e-02   3.526 0.000437 ***
## GarageType_2Types      -6.159e-02  2.444e-02  -2.520 0.011864 *
## GarageFinish_Unf       -9.513e-03  4.270e-03  -2.228 0.026037 *
## GarageCond_TA           1.901e-02  9.066e-03   2.097 0.036151 *
## PoolQC_Missing                 NA         NA      NA       NA
## PoolQC_Fa              -1.195e-01  5.067e-02  -2.359 0.018475 *
## SaleType_WD            -4.507e-02  7.544e-03  -5.974 2.95e-09 ***
## SaleType_COD           -4.561e-02  1.113e-02  -4.098 4.42e-05 ***
## SaleType_ConLI         -6.025e-02  2.634e-02  -2.288 0.022317 *
## SaleCondition_Normal    3.381e-02  5.412e-03   6.248 5.55e-10 ***
## SaleCondition_AdjLand   6.363e-02  3.041e-02   2.092 0.036610 *
## MSSubClass_Char_20      1.045e-02  4.820e-03   2.167 0.030394 *
## MSSubClass_Char_70      3.696e-02  9.755e-03   3.789 0.000158 ***
## MSSubClass_Char_50      2.489e-02  6.535e-03   3.809 0.000146 ***
## MSSubClass_Char_45     -1.114e-01  5.121e-02  -2.176 0.029718 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05567 on 1353 degrees of freedom
## Multiple R-squared:   0.94,  Adjusted R-squared:  0.9353
## F-statistic: 199.8 on 106 and 1353 DF,  p-value: < 2.2e-16
```

When I submitted this reduced model, it did not score as well in Kaggle as previous submissions using all of the variables had done. So I decided to try one final approach. I would remove any variables whose removal increased the adjusted R^2 value of the model. I created another loop to do so, and had the loop continue while there were any increases in the adjusted R^2 value of the model. Once this stopped, the loop ended and I had my final model.

This approach barely increased the adjusted R^2 to 0.9354472, and it did not remove many variables. However, it did produce my highest Kaggle score.

```
### loop through variables to increase adj.r-squared
model3<-model
summary3 <-summary
previous_r2<-0
current_r2 <-.1
change<-"Yes"

while(change=="Yes") {
  change <- "No"
  names_check<-names(sort(summary3$coefficients[,4], decreasing = TRUE))
  for(i in 1:length(names_check)){
    #i<-1
    #update model by removing variable and see if adj.rsquared is higher
    model3_compare<- update(model3, as.formula(paste0(". ~ . -",names_check[i])))
    current_r2<-summary(model3_compare)$adj.r.squared

    #if adjrquared increases, update model
    if(current_r2 > previous_r2){
```

```
      model3<-model3_compare
      summary3<-summary(model3_compare)
      change<-"Yes"
      previous_r2 <- current_r2
    } else{
      #do nothing
    } #if

    #print(paste0(i,": adj-r.squared: ", previous_r2))
  }#for
} #while
```

```
#adjusted r squared - 0.9354472
summary3$adj.r.squared
```

```
## [1] 0.9354472
```

```
#summary3
summary3
```

```
##
## Call:
## lm(formula = bc_SalePrice ~ MSSubClass + LotFrontage + LotArea +
##     OverallQual + OverallCond + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF +
##     X1stFlrSF + X2ndFlrSF + LowQualFinSF + BsmtFullBath + FullBath +
##     HalfBath + KitchenAbvGr + Fireplaces + GarageCars + GarageArea +
##     WoodDeckSF + OpenPorchSF + EnclosedPorch + X3SsnPorch + ScreenPorch +
##     PoolArea + YrSold + YearsOld + YearsRemod + GarageYrsOld +
##     GarageYrsOld_Exp + YrSold_Med + LotArea_Outlier + LotArea_Log +
##     BsmtUnfSF_Exp + TotalBsmtSF_Outlier + OpenPorchSF_NoZero +
##     EnclosedPorch_NoZero + PoolExists + MSZoning_RL + MSZoning_RM +
##     `MSZoning_C (all)` + MSZoning_FV + MSZoning_RH + Street_Pave +
##     Street_Grvl + Alley_Missing + Alley_Grvl + Alley_Pave + LotShape_Reg +
##     LotShape_IR1 + LotShape_IR2 + LotShape_IR3 + LandContour_Lvl +
##     LandContour_Bnk + LandContour_Low + LandContour_HLS + Utilities_AllPub +
##     Utilities_NoSeWa + LotConfig_Inside + LotConfig_FR2 + LotConfig_Corner +
##     LotConfig_CulDSac + LotConfig_FR3 + LandSlope_Gtl + LandSlope_Mod +
##     LandSlope_Sev + Neighborhood_CollgCr + Neighborhood_Veenker +
##     Neighborhood_Crawfor + Neighborhood_NoRidge + Neighborhood_Mitchel +
##     Neighborhood_Somerst + Neighborhood_NWAmes + Neighborhood_OldTown +
##     Neighborhood_BrkSide + Neighborhood_Sawyer + Neighborhood_NridgHt +
##     Neighborhood_NAmes + Neighborhood_SawyerW + Neighborhood_IDOTRR +
##     Neighborhood_MeadowV + Neighborhood_Edwards + Neighborhood_Timber +
##     Neighborhood_Gilbert + Neighborhood_StoneBr + Neighborhood_ClearCr +
##     Neighborhood_NPkVill + Neighborhood_Blmngtn + Neighborhood_BrDale +
##     Neighborhood_SWISU + Neighborhood_Blueste + Condition1_Norm +
##     Condition1_Feedr + Condition1_PosN + Condition1_Artery +
##     Condition1_RRAe + Condition1_RRNn + Condition1_RRAn + Condition1_PosA +
##     Condition1_RRNe + Condition2_Norm + Condition2_Artery + Condition2_RRNn +
##     Condition2_Feedr + Condition2_PosN + Condition2_PosA + Condition2_RRAn +
##     Condition2_RRAe + BldgType_1Fam + BldgType_2fmCon + BldgType_Duplex +
##     BldgType_TwnhsE + BldgType_Twnhs + HouseStyle_1.5Unf + HouseStyle_2.5Unf +
##     RoofStyle_Gable + RoofStyle_Hip + RoofStyle_Gambrel + RoofStyle_Mansard +
##     RoofStyle_Flat + RoofStyle_Shed + RoofMatl_CompShg + RoofMatl_WdShngl +
##     RoofMatl_Metal + RoofMatl_WdShake + RoofMatl_Membran + `RoofMatl_Tar&Grv` +
```

```
##     RoofMatl_Roll + RoofMatl_ClyTile + Exterior1st_VinylSd +
##     Exterior1st_MetalSd + `Exterior1st_Wd Sdng` + Exterior1st_HdBoard +
##     Exterior1st_BrkFace + Exterior1st_WdShing + Exterior1st_CemntBd +
##     Exterior1st_Plywood + Exterior1st_AsbShng + Exterior1st_Stucco +
##     Exterior1st_AsphShn + Exterior1st_Stone + Exterior1st_ImStucc +
##     Exterior1st_CBlock + `Exterior2nd_Wd Sdng` + Exterior2nd_CmentBd +
##     Exterior2nd_BrkFace + Exterior2nd_AsbShng + `Exterior2nd_Brk Cmn` +
##     Exterior2nd_Other + Exterior2nd_CBlock + MasVnrType_BrkFace +
##     MasVnrType_None + MasVnrType_Stone + MasVnrType_BrkCmn +
##     ExterQual_Gd + ExterQual_TA + ExterQual_Ex + ExterQual_Fa +
##     ExterCond_TA + ExterCond_Gd + ExterCond_Fa + ExterCond_Po +
##     ExterCond_Ex + Foundation_PConc + Foundation_CBlock + Foundation_BrkTil +
##     Foundation_Wood + Foundation_Slab + Foundation_Stone + BsmtQual_Gd +
##     BsmtQual_TA + BsmtQual_Ex + BsmtQual_Fa + BsmtCond_TA + BsmtCond_Gd +
##     BsmtCond_Fa + BsmtCond_Po + BsmtExposure_No + BsmtExposure_Gd +
##     BsmtExposure_Mn + BsmtExposure_Av + BsmtFinType1_GLQ + BsmtFinType1_ALQ +
##     BsmtFinType1_Unf + BsmtFinType1_Rec + BsmtFinType1_BLQ +
##     BsmtFinType1_LwQ + BsmtFinType2_Unf + BsmtFinType2_BLQ +
##     BsmtFinType2_ALQ + BsmtFinType2_Rec + BsmtFinType2_LwQ +
##     BsmtFinType2_GLQ + Heating_GasA + Heating_GasW + Heating_Grav +
##     Heating_Wall + Heating_OthW + Heating_Floor + HeatingQC_Ex +
##     HeatingQC_Gd + HeatingQC_TA + HeatingQC_Fa + HeatingQC_Po +
##     CentralAir_Y + CentralAir_N + Electrical_SBrkr + Electrical_FuseF +
##     Electrical_FuseA + Electrical_FuseP + Electrical_Mix + KitchenQual_Gd +
##     KitchenQual_TA + KitchenQual_Ex + KitchenQual_Fa + Functional_Typ +
##     Functional_Min1 + Functional_Maj1 + Functional_Min2 + Functional_Mod +
##     Functional_Maj2 + Functional_Sev + FireplaceQu_Missing +
##     FireplaceQu_TA + FireplaceQu_Gd + FireplaceQu_Fa + FireplaceQu_Ex +
##     FireplaceQu_Po + GarageType_Attchd + GarageType_Detchd +
##     GarageType_BuiltIn + GarageType_CarPort + GarageType_Basment +
##     GarageType_2Types + GarageFinish_RFn + GarageFinish_Unf +
##     GarageFinish_Fin + GarageQual_TA + GarageQual_Fa + GarageQual_Gd +
##     GarageQual_Ex + GarageQual_Po + GarageCond_TA + GarageCond_Fa +
##     GarageCond_Gd + GarageCond_Po + GarageCond_Ex + PavedDrive_Y +
##     PavedDrive_N + PavedDrive_P + PoolQC_Missing + PoolQC_Ex +
##     PoolQC_Fa + PoolQC_Gd + Fence_Missing + Fence_MnPrv + Fence_GdWo +
##     Fence_GdPrv + Fence_MnWw + MiscFeature_Missing + MiscFeature_Shed +
##     MiscFeature_Gar2 + MiscFeature_Othr + MiscFeature_TenC +
##     SaleType_WD + SaleType_New + SaleType_COD + SaleType_ConLD +
##     SaleType_ConLI + SaleType_CWD + SaleType_ConLw + SaleType_Con +
##     SaleType_Oth + SaleCondition_Normal + SaleCondition_Abnorml +
##     SaleCondition_Partial + SaleCondition_AdjLand + SaleCondition_Alloca +
##     SaleCondition_Family + MSSubClass_Char_60 + MSSubClass_Char_20 +
##     MSSubClass_Char_70 + MSSubClass_Char_50 + MSSubClass_Char_190 +
##     MSSubClass_Char_45 + MSSubClass_Char_90 + MSSubClass_Char_120 +
##     MSSubClass_Char_30 + MSSubClass_Char_85 + MSSubClass_Char_80 +
##     MSSubClass_Char_160 + MSSubClass_Char_75 + MSSubClass_Char_180 +
##     MSSubClass_Char_40, data = train_subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38328 -0.02289  0.00118  0.02594  0.32817
##
## Coefficients: (45 not defined because of singularities)
```

```
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.469e+01  5.288e+00   2.778 0.005558 **
## MSSubClass              4.480e-04  4.396e-04   1.019 0.308314
## LotFrontage             1.321e-04  1.100e-04   1.200 0.230236
## LotArea                 9.824e-07  3.738e-07   2.628 0.008699 **
## OverallQual             2.013e-02  2.524e-03   7.974 3.53e-15 ***
## OverallCond             2.067e-02  2.178e-03   9.491  < 2e-16 ***
## BsmtFinSF1              2.053e-04  3.169e-05   6.480 1.33e-10 ***
## BsmtFinSF2              1.995e-04  3.716e-05   5.368 9.54e-08 ***
## BsmtUnfSF               1.448e-04  2.945e-05   4.917 1.00e-06 ***
## X1stFlrSF               1.341e-04  1.219e-05  10.996  < 2e-16 ***
## X2ndFlrSF               1.280e-04  1.142e-05  11.207  < 2e-16 ***
## LowQualFinSF            9.911e-05  4.175e-05   2.374 0.017753 *
## BsmtFullBath            9.382e-03  4.661e-03   2.013 0.044353 *
## FullBath                1.291e-02  5.367e-03   2.406 0.016270 *
## HalfBath                1.474e-02  5.141e-03   2.868 0.004208 **
## KitchenAbvGr           -2.014e-02  1.415e-02  -1.423 0.155035
## Fireplaces              7.726e-03  6.308e-03   1.225 0.220927
## GarageCars              1.108e-02  5.412e-03   2.047 0.040880 *
## GarageArea              6.021e-05  1.903e-05   3.165 0.001591 **
## WoodDeckSF              4.647e-05  1.456e-05   3.192 0.001449 **
## OpenPorchSF             1.152e-04  6.387e-05   1.803 0.071591 .
## EnclosedPorch           1.124e-04  3.842e-05   2.924 0.003516 **
## X3SsnPorch              7.992e-05  5.477e-05   1.459 0.144763
## ScreenPorch             1.532e-04  3.070e-05   4.990 6.92e-07 ***
## PoolArea                6.797e-04  5.594e-04   1.215 0.224612
## YrSold                 -3.670e-03  1.869e-03  -1.964 0.049784 *
## YearsOld               -9.547e-04  2.067e-04  -4.619 4.27e-06 ***
## YearsRemod             -4.269e-04  1.379e-04  -3.095 0.002015 **
## GarageYrsOld            9.980e-04  3.337e-04   2.991 0.002838 **
## GarageYrsOld_Exp       -3.609e-02  1.270e-02  -2.841 0.004576 **
## YrSold_Med             -2.630e-01  2.051e-01  -1.282 0.200015
## LotArea_Outlier        -2.455e-06  1.076e-06  -2.282 0.022660 *
## LotArea_Log             6.165e-02  1.448e-02   4.257 2.24e-05 ***
## BsmtUnfSF_Exp           1.555e-08  7.585e-09   2.051 0.040523 *
## TotalBsmtSF_Outlier    -2.610e-06  6.059e-07  -4.307 1.79e-05 ***
## OpenPorchSF_NoZero     -1.185e-04  7.724e-05  -1.535 0.125154
## EnclosedPorch_NoZero   -1.138e-04  7.543e-05  -1.508 0.131798
## PoolExists             -3.041e-01  3.624e-01  -0.839 0.401577
## MSZoning_RL            -5.772e-03  1.641e-02  -0.352 0.725131
## MSZoning_RM            -2.168e-02  1.869e-02  -1.160 0.246086
## `MSZoning_C (all)`     -2.653e-01  2.947e-02  -9.002  < 2e-16 ***
## MSZoning_FV             1.763e-02  2.319e-02   0.760 0.447187
## MSZoning_RH                    NA         NA      NA       NA
## Street_Pave             4.840e-02  3.034e-02   1.595 0.110957
## Street_Grvl                    NA         NA      NA       NA
## Alley_Missing          -2.281e-02  1.207e-02  -1.889 0.059089 .
## Alley_Grvl             -1.949e-02  1.513e-02  -1.288 0.198077
## Alley_Pave                     NA         NA      NA       NA
## LotShape_Reg           -6.765e-03  2.189e-02  -0.309 0.757291
## LotShape_IR1           -1.366e-02  2.179e-02  -0.627 0.530668
## LotShape_IR2           -2.750e-03  2.318e-02  -0.119 0.905576
## LotShape_IR3                   NA         NA      NA       NA
## LandContour_Lvl        -1.898e-03  1.001e-02  -0.190 0.849672
```

```
## LandContour_Bnk       -1.453e-02  1.253e-02  -1.160 0.246450
## LandContour_Low       -2.939e-02  1.535e-02  -1.914 0.055850 .
## LandContour_HLS              NA         NA      NA       NA
## Utilities_AllPub       1.211e-01  6.247e-02   1.938 0.052797 .
## Utilities_NoSeWa             NA         NA      NA       NA
## LotConfig_Inside       4.392e-02  3.077e-02   1.427 0.153788
## LotConfig_FR2          3.129e-02  3.175e-02   0.985 0.324668
## LotConfig_Corner       5.025e-02  3.097e-02   1.623 0.104917
## LotConfig_CulDSac      6.572e-02  3.164e-02   2.077 0.038015 *
## LotConfig_FR3                NA         NA      NA       NA
## LandSlope_Gtl          8.504e-02  2.910e-02   2.922 0.003541 **
## LandSlope_Mod          1.013e-01  2.912e-02   3.479 0.000520 ***
## LandSlope_Sev                NA         NA      NA       NA
## Neighborhood_CollgCr  -6.801e-02  4.704e-02  -1.446 0.148470
## Neighborhood_Veenker  -4.135e-02  4.999e-02  -0.827 0.408265
## Neighborhood_Crawfor  -6.601e-03  4.745e-02  -0.139 0.889385
## Neighborhood_NoRidge  -3.129e-02  4.819e-02  -0.649 0.516277
## Neighborhood_Mitchel  -8.975e-02  4.722e-02  -1.901 0.057570 .
## Neighborhood_Somerst  -5.111e-02  4.909e-02  -1.041 0.298031
## Neighborhood_NWAmes   -7.895e-02  4.669e-02  -1.691 0.091142 .
## Neighborhood_OldTown  -8.349e-02  4.681e-02  -1.784 0.074746 .
## Neighborhood_BrkSide  -5.224e-02  4.736e-02  -1.103 0.270216
## Neighborhood_Sawyer   -7.736e-02  4.707e-02  -1.644 0.100537
## Neighborhood_NridgHt  -2.130e-02  4.771e-02  -0.446 0.655426
## Neighborhood_NAmes    -8.170e-02  4.674e-02  -1.748 0.080750 .
## Neighborhood_SawyerW  -6.075e-02  4.719e-02  -1.287 0.198207
## Neighborhood_IDOTRR   -7.172e-02  4.810e-02  -1.491 0.136216
## Neighborhood_MeadowV  -1.144e-01  4.726e-02  -2.420 0.015668 *
## Neighborhood_Edwards  -1.051e-01  4.678e-02  -2.246 0.024887 *
## Neighborhood_Timber   -5.775e-02  4.786e-02  -1.207 0.227815
## Neighborhood_Gilbert  -6.259e-02  4.725e-02  -1.325 0.185523
## Neighborhood_StoneBr   1.846e-02  4.836e-02   0.382 0.702771
## Neighborhood_ClearCr  -3.684e-02  4.842e-02  -0.761 0.446948
## Neighborhood_NPkVill  -4.923e-02  5.486e-02  -0.897 0.369721
## Neighborhood_Blmngtn  -4.169e-02  4.927e-02  -0.846 0.397607
## Neighborhood_BrDale   -3.823e-02  4.623e-02  -0.827 0.408439
## Neighborhood_SWISU    -5.765e-02  4.911e-02  -1.174 0.240643
## Neighborhood_Blueste         NA         NA      NA       NA
## Condition1_Norm        3.268e-02  4.166e-02   0.784 0.432940
## Condition1_Feedr       8.472e-03  4.227e-02   0.200 0.841181
## Condition1_PosN        4.485e-02  4.412e-02   1.017 0.309538
## Condition1_Artery     -1.529e-02  4.301e-02  -0.355 0.722321
## Condition1_RRAe       -4.549e-02  4.551e-02  -1.000 0.317739
## Condition1_RRNn        3.725e-02  5.066e-02   0.735 0.462213
## Condition1_RRAn        1.338e-02  4.390e-02   0.305 0.760612
## Condition1_PosA        8.562e-03  4.731e-02   0.181 0.856421
## Condition1_RRNe              NA         NA      NA       NA
## Condition2_Norm        3.254e-01  1.196e-01   2.721 0.006593 **
## Condition2_Artery      2.422e-01  1.345e-01   1.800 0.072093 .
## Condition2_RRNn        3.061e-01  1.265e-01   2.419 0.015723 *
## Condition2_Feedr       3.539e-01  1.256e-01   2.817 0.004931 **
## Condition2_PosN       -9.081e-02  1.275e-01  -0.712 0.476592
## Condition2_PosA        4.515e-01  1.441e-01   3.133 0.001769 **
## Condition2_RRAn        2.694e-01  1.331e-01   2.024 0.043193 *
```

```
## Condition2_RRAe               NA         NA      NA       NA
## BldgType_1Fam          1.815e-02  3.862e-02   0.470 0.638494
## BldgType_2fmCon       -2.427e-02  7.716e-02  -0.315 0.753176
## BldgType_Duplex        2.829e-02  4.158e-02   0.680 0.496344
## BldgType_TwnhsE        5.537e-04  1.314e-02   0.042 0.966394
## BldgType_Twnhs                NA         NA      NA       NA
## HouseStyle_1.5Unf      1.162e-01  5.855e-02   1.985 0.047368 *
## HouseStyle_2.5Unf      6.695e-02  3.106e-02   2.156 0.031297 *
## RoofStyle_Gable       -2.326e-01  8.477e-02  -2.744 0.006161 **
## RoofStyle_Hip         -2.311e-01  8.483e-02  -2.724 0.006540 **
## RoofStyle_Gambrel     -2.494e-01  8.708e-02  -2.864 0.004256 **
## RoofStyle_Mansard     -1.995e-01  8.438e-02  -2.365 0.018193 *
## RoofStyle_Flat        -2.464e-01  9.395e-02  -2.623 0.008831 **
## RoofStyle_Shed                NA         NA      NA       NA
## RoofMatl_CompShg       2.057e+00  2.086e-01   9.865  < 2e-16 ***
## RoofMatl_WdShngl       2.107e+00  2.119e-01   9.945  < 2e-16 ***
## RoofMatl_Metal         2.161e+00  2.242e-01   9.637  < 2e-16 ***
## RoofMatl_WdShake       2.024e+00  2.115e-01   9.573  < 2e-16 ***
## RoofMatl_Membran       2.286e+00  2.247e-01  10.173  < 2e-16 ***
## `RoofMatl_Tar&Grv`     2.077e+00  2.139e-01   9.711  < 2e-16 ***
## RoofMatl_Roll          2.068e+00  2.182e-01   9.475  < 2e-16 ***
## RoofMatl_ClyTile              NA         NA      NA       NA
## Exterior1st_VinylSd    1.700e-01  6.230e-02   2.728 0.006455 **
## Exterior1st_MetalSd    1.793e-01  6.261e-02   2.864 0.004254 **
## `Exterior1st_Wd Sdng`  1.373e-01  6.318e-02   2.174 0.029904 *
## Exterior1st_HdBoard    1.624e-01  6.251e-02   2.599 0.009477 **
## Exterior1st_BrkFace    2.094e-01  6.334e-02   3.306 0.000976 ***
## Exterior1st_WdShing    1.721e-01  6.348e-02   2.711 0.006798 **
## Exterior1st_CemntBd    1.198e-01  7.190e-02   1.666 0.096005 .
## Exterior1st_Plywood    1.645e-01  6.253e-02   2.631 0.008620 **
## Exterior1st_AsbShng    1.843e-01  6.905e-02   2.670 0.007698 **
## Exterior1st_Stucco     1.771e-01  6.361e-02   2.784 0.005448 **
## Exterior1st_AsphShn    1.940e-01  8.697e-02   2.231 0.025881 *
## Exterior1st_Stone      1.580e-01  7.850e-02   2.013 0.044349 *
## Exterior1st_ImStucc    1.568e-01  8.582e-02   1.827 0.067871 .
## Exterior1st_CBlock     1.227e-01  9.136e-02   1.343 0.179590
## `Exterior2nd_Wd Sdng`  2.572e-02  1.029e-02   2.500 0.012562 *
## Exterior2nd_CmentBd    5.953e-02  3.695e-02   1.611 0.107429
## Exterior2nd_BrkFace   -2.184e-02  1.774e-02  -1.231 0.218445
## Exterior2nd_AsbShng   -3.156e-02  2.862e-02  -1.103 0.270403
## `Exterior2nd_Brk Cmn`  6.316e-02  4.130e-02   1.529 0.126458
## Exterior2nd_Other     -6.050e-02  5.858e-02  -1.033 0.301963
## Exterior2nd_CBlock            NA         NA      NA       NA
## MasVnrType_BrkFace     1.843e-02  1.672e-02   1.102 0.270565
## MasVnrType_None        1.053e-02  1.650e-02   0.638 0.523357
## MasVnrType_Stone       2.186e-02  1.775e-02   1.232 0.218221
## MasVnrType_BrkCmn             NA         NA      NA       NA
## ExterQual_Gd          -9.535e-03  2.649e-02  -0.360 0.718917
## ExterQual_TA          -8.394e-03  2.584e-02  -0.325 0.745347
## ExterQual_Ex          -1.213e-02  2.897e-02  -0.419 0.675656
## ExterQual_Fa                  NA         NA      NA       NA
## ExterCond_TA          -1.493e-02  4.209e-02  -0.355 0.722833
## ExterCond_Gd          -2.489e-02  4.218e-02  -0.590 0.555327
## ExterCond_Fa          -3.499e-02  4.418e-02  -0.792 0.428584
```

```
## ExterCond_Po          3.459e-02  7.779e-02   0.445 0.656606
## ExterCond_Ex                 NA         NA      NA       NA
## Foundation_PConc     -2.910e-02  2.776e-02  -1.049 0.294608
## Foundation_CBlock    -4.067e-02  2.760e-02  -1.473 0.140885
## Foundation_BrkTil    -5.210e-02  2.761e-02  -1.887 0.059381 .
## Foundation_Wood      -1.171e-01  4.486e-02  -2.611 0.009140 **
## Foundation_Slab      -3.064e-02  3.333e-02  -0.919 0.358165
## Foundation_Stone             NA         NA      NA       NA
## BsmtQual_Gd          -1.128e-02  1.355e-02  -0.833 0.405125
## BsmtQual_TA          -7.837e-03  1.222e-02  -0.641 0.521386
## BsmtQual_Ex           3.891e-03  1.576e-02   0.247 0.805033
## BsmtQual_Fa                  NA         NA      NA       NA
## BsmtCond_TA          -1.085e-01  7.471e-02  -1.452 0.146699
## BsmtCond_Gd          -1.107e-01  7.515e-02  -1.473 0.141108
## BsmtCond_Fa          -1.255e-01  7.427e-02  -1.690 0.091367 .
## BsmtCond_Po                  NA         NA      NA       NA
## BsmtExposure_No      -2.728e-03  5.342e-03  -0.511 0.609728
## BsmtExposure_Gd       1.954e-02  7.384e-03   2.646 0.008251 **
## BsmtExposure_Mn      -2.859e-03  7.397e-03  -0.387 0.699177
## BsmtExposure_Av              NA         NA      NA       NA
## BsmtFinType1_GLQ      2.315e-02  9.348e-03   2.476 0.013414 *
## BsmtFinType1_ALQ      1.869e-02  9.189e-03   2.034 0.042192 *
## BsmtFinType1_Unf      1.030e-02  9.306e-03   1.107 0.268681
## BsmtFinType1_Rec      1.235e-02  9.348e-03   1.321 0.186677
## BsmtFinType1_BLQ      1.468e-02  9.350e-03   1.570 0.116614
## BsmtFinType1_LwQ             NA         NA      NA       NA
## BsmtFinType2_Unf     -5.700e-03  2.172e-02  -0.262 0.793029
## BsmtFinType2_BLQ     -3.307e-02  2.165e-02  -1.528 0.126867
## BsmtFinType2_ALQ      1.272e-02  2.307e-02   0.551 0.581599
## BsmtFinType2_Rec     -1.136e-02  2.076e-02  -0.547 0.584362
## BsmtFinType2_LwQ     -1.476e-02  2.143e-02  -0.689 0.491247
## BsmtFinType2_GLQ             NA         NA      NA       NA
## Heating_GasA          3.349e-02  6.127e-02   0.546 0.584836
## Heating_GasW          6.132e-02  6.330e-02   0.969 0.332860
## Heating_Grav         -6.614e-02  6.749e-02  -0.980 0.327291
## Heating_Wall          6.316e-02  7.081e-02   0.892 0.372537
## Heating_OthW          3.604e-02  7.605e-02   0.474 0.635628
## Heating_Floor                NA         NA      NA       NA
## HeatingQC_Ex          2.186e-02  6.557e-02   0.333 0.738851
## HeatingQC_Gd          1.146e-02  6.565e-02   0.174 0.861517
## HeatingQC_TA          5.479e-03  6.552e-02   0.084 0.933368
## HeatingQC_Fa          7.376e-03  6.640e-02   0.111 0.911560
## HeatingQC_Po                 NA         NA      NA       NA
## CentralAir_Y          3.490e-02  9.479e-03   3.682 0.000241 ***
## CentralAir_N                 NA         NA      NA       NA
## Electrical_SBrkr      9.362e-02  1.102e-01   0.849 0.395798
## Electrical_FuseF      9.459e-02  1.115e-01   0.848 0.396565
## Electrical_FuseA      1.060e-01  1.100e-01   0.963 0.335611
## Electrical_FuseP      6.216e-02  1.121e-01   0.554 0.579358
## Electrical_Mix               NA         NA      NA       NA
## KitchenQual_Gd       -4.263e-03  1.312e-02  -0.325 0.745228
## KitchenQual_TA       -4.097e-03  1.206e-02  -0.340 0.734147
## KitchenQual_Ex        2.983e-02  1.525e-02   1.956 0.050682 .
## KitchenQual_Fa               NA         NA      NA       NA
```

```
## Functional_Typ         1.850e-01  7.093e-02   2.609 0.009204 **
## Functional_Min1        1.728e-01  7.101e-02   2.434 0.015095 *
## Functional_Maj1        1.608e-01  7.278e-02   2.210 0.027299 *
## Functional_Min2        1.619e-01  7.198e-02   2.249 0.024691 *
## Functional_Mod         1.159e-01  7.303e-02   1.587 0.112705
## Functional_Maj2        3.043e-02  7.710e-02   0.395 0.693104
## Functional_Sev                NA         NA      NA       NA
## FireplaceQu_Missing   -1.264e-02  1.575e-02  -0.803 0.422295
## FireplaceQu_TA        -4.445e-03  1.493e-02  -0.298 0.765939
## FireplaceQu_Gd        -5.084e-03  1.487e-02  -0.342 0.732471
## FireplaceQu_Fa        -1.539e-02  1.780e-02  -0.864 0.387651
## FireplaceQu_Ex        -1.429e-02  1.948e-02  -0.734 0.463152
## FireplaceQu_Po                NA         NA      NA       NA
## GarageType_Attchd      6.301e-02  2.694e-02   2.339 0.019523 *
## GarageType_Detchd      6.745e-02  2.685e-02   2.512 0.012141 *
## GarageType_BuiltIn     5.935e-02  2.808e-02   2.114 0.034726 *
## GarageType_CarPort     6.591e-02  3.604e-02   1.829 0.067694 .
## GarageType_Basment     6.332e-02  3.144e-02   2.014 0.044225 *
## GarageType_2Types             NA         NA      NA       NA
## GarageFinish_RFn      -1.036e-03  4.821e-03  -0.215 0.829892
## GarageFinish_Unf      -7.183e-03  5.903e-03  -1.217 0.223875
## GarageFinish_Fin              NA         NA      NA       NA
## GarageQual_TA          2.072e-02  6.399e-02   0.324 0.746099
## GarageQual_Fa         -8.146e-04  6.287e-02  -0.013 0.989665
## GarageQual_Gd          1.858e-02  6.736e-02   0.276 0.782757
## GarageQual_Ex          1.695e-01  9.816e-02   1.727 0.084409 .
## GarageQual_Po                 NA         NA      NA       NA
## GarageCond_TA          1.331e-01  8.556e-02   1.556 0.120066
## GarageCond_Fa          1.142e-01  8.634e-02   1.322 0.186299
## GarageCond_Gd          1.421e-01  8.975e-02   1.583 0.113725
## GarageCond_Po          1.666e-01  9.398e-02   1.772 0.076616 .
## GarageCond_Ex                 NA         NA      NA       NA
## PavedDrive_Y           1.363e-02  1.207e-02   1.130 0.258866
## PavedDrive_N           6.480e-03  1.353e-02   0.479 0.632054
## PavedDrive_P                  NA         NA      NA       NA
## PoolQC_Missing                NA         NA      NA       NA
## PoolQC_Ex             -9.218e-03  8.869e-02  -0.104 0.917238
## PoolQC_Fa             -1.038e-01  7.585e-02  -1.369 0.171298
## PoolQC_Gd                     NA         NA      NA       NA
## Fence_Missing          1.130e-02  1.831e-02   0.617 0.537155
## Fence_MnPrv            7.194e-03  1.871e-02   0.385 0.700598
## Fence_GdWo            -1.041e-02  1.971e-02  -0.528 0.597380
## Fence_GdPrv            4.110e-03  2.011e-02   0.204 0.838078
## Fence_MnWw                    NA         NA      NA       NA
## MiscFeature_Missing    2.347e-02  1.120e-01   0.210 0.834030
## MiscFeature_Shed       2.157e-02  1.126e-01   0.191 0.848171
## MiscFeature_Gar2       4.303e-02  1.263e-01   0.341 0.733337
## MiscFeature_Othr       2.584e-03  1.240e-01   0.021 0.983378
## MiscFeature_TenC              NA         NA      NA       NA
## SaleType_WD           -3.598e-02  3.446e-02  -1.044 0.296676
## SaleType_New           2.092e-02  5.006e-02   0.418 0.676136
## SaleType_COD          -2.555e-02  3.529e-02  -0.724 0.469120
## SaleType_ConLD         4.895e-02  4.053e-02   1.208 0.227407
## SaleType_ConLI        -4.705e-02  4.308e-02  -1.092 0.275031
```

```
## SaleType_CWD           4.807e-03  4.537e-02   0.106 0.915638
## SaleType_ConLw        -2.033e-02  4.457e-02  -0.456 0.648335
## SaleType_Con           1.221e-02  5.415e-02   0.225 0.821697
## SaleType_Oth                  NA         NA      NA       NA
## SaleCondition_Normal   3.153e-02  1.358e-02   2.321 0.020430 *
## SaleCondition_Abnorml -3.881e-03  1.496e-02  -0.259 0.795327
## SaleCondition_Partial -1.501e-02  3.844e-02  -0.390 0.696322
## SaleCondition_AdjLand  6.758e-02  3.758e-02   1.798 0.072348 .
## SaleCondition_Alloca   2.403e-02  2.424e-02   0.991 0.321667
## SaleCondition_Family          NA         NA      NA       NA
## MSSubClass_Char_60     1.951e-02  3.722e-02   0.524 0.600154
## MSSubClass_Char_20     5.155e-02  4.787e-02   1.077 0.281687
## MSSubClass_Char_70     4.053e-02  3.498e-02   1.159 0.246888
## MSSubClass_Char_50     4.281e-02  3.817e-02   1.122 0.262238
## MSSubClass_Char_190    3.869e-03  8.414e-02   0.046 0.963329
## MSSubClass_Char_45    -7.712e-02  7.276e-02  -1.060 0.289389
## MSSubClass_Char_90            NA         NA      NA       NA
## MSSubClass_Char_120    1.298e-02  2.813e-02   0.461 0.644691
## MSSubClass_Char_30     1.081e-02  4.476e-02   0.241 0.809224
## MSSubClass_Char_85     1.872e-02  3.455e-02   0.542 0.588010
## MSSubClass_Char_80     2.663e-02  3.353e-02   0.794 0.427135
## MSSubClass_Char_160   -3.141e-02  2.395e-02  -1.311 0.189939
## MSSubClass_Char_75    -2.396e-02  4.164e-02  -0.575 0.565124
## MSSubClass_Char_180           NA         NA      NA       NA
## MSSubClass_Char_40            NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05559 on 1211 degrees of freedom
## Multiple R-squared:  0.9464, Adjusted R-squared:  0.9354
## F-statistic: 86.25 on 248 and 1211 DF,  p-value: < 2.2e-16
```

I made one last attempt to increase my Kaggle score. I added interaction variables to see if I could get the adjusted R^2 to increase even more. To limit the number of variables to check and to only use those most significant, I used the model2 as a starting point and checked each variable's interaction against every other variable using the ":" operator. As you can imagine, with 107 variables to check, this took a long time (41.28717 mins).

I immediately got increased adjusted R^2 values, and it ended up being 0.9772524 after my code completed. However, my Kaggle submission was much much worse, suggesting that I was overfitting the data I had and that this model was not generalizing. This is not surprising as the variables increased to 766 in count.

```
### loop through variables to increase adj.r-squared
model4<-model2
summary4 <-summary2
previous_r2<-0
current_r2 <-.1
startTime<-Sys.time()

  names_check<-names(sort(summary4$coefficients[,4], decreasing = FALSE))
  for(i in 2:(length(names_check)-1)){
    for(j in (i+1):length(names_check)){
      #only check ones that have not yet been checked
      #i<-2
      #j<-3
```

```
      #update model by adding interaction variable and see if adj.rsquared is higher
      model4_compare<- update(model4, as.formula(paste0(". ~ . +",names_check[i],":",names_check[j])))
      current_r2<-summary(model4_compare)$adj.r.squared

      #if adjrquared increases, update model
      if(current_r2 > previous_r2){
        model4<-model4_compare
        summary4<-summary(model4_compare)
        previous_r2 <- current_r2
      } else{
        #do nothing
      } #if

      print(paste0(i," ", j,": adj-r.squared: ", previous_r2))
    }# for j
  }#for i

#time elapsed
endTime<-Sys.time()
timeElapsed <- endTime - startTime


#adjusted r squared - 0.9772524
summary4$adj.r.squared

#summary4
#summary4

#number of variables - 766
length(summary4$coefficients[,4])
```

So now what to do? Again, many of these new variables had an extremely high p-value. So I decided to remove any that had high p-values, as I had done previously. I also wanted to reduce the number of variables being used since I still had way too many and was overfitting. The adj.r.squared went up even more to a point (above 0.9807), and then came back down again. I tried different p-values to reduce the number of variables, but none of these produced a better score on Kaggle.

When looking at the model, I could see that many variables were being reused in the interactions (e.g., 1st floor square footage) and the original variables that were NOT interactive were still the most significant. So I wasn't gaining any generalizability by adding these interactive variables as they did not contain new information, even though the R^2 value was increasing due to overfitting. Consequently I decided to be satisfied with my score as it was and to move on.

```
###model 5 - remove all interaction variables that do NOT lower the adj.r.squared value
model5 <- model4
summary5 <-summary4

while(sort(summary5$coefficients[,4], decreasing = TRUE)[1]>0.001) {
  #update model by removing highest p-value until threshold reached
  name <-names(sort(summary5$coefficients[,4], decreasing = TRUE)[1])
  model5<- update(model5, as.formula(paste0(". ~ . -",name)))
  summary5<-summary(model5)
  print(summary5$adj.r.squared)
}
```

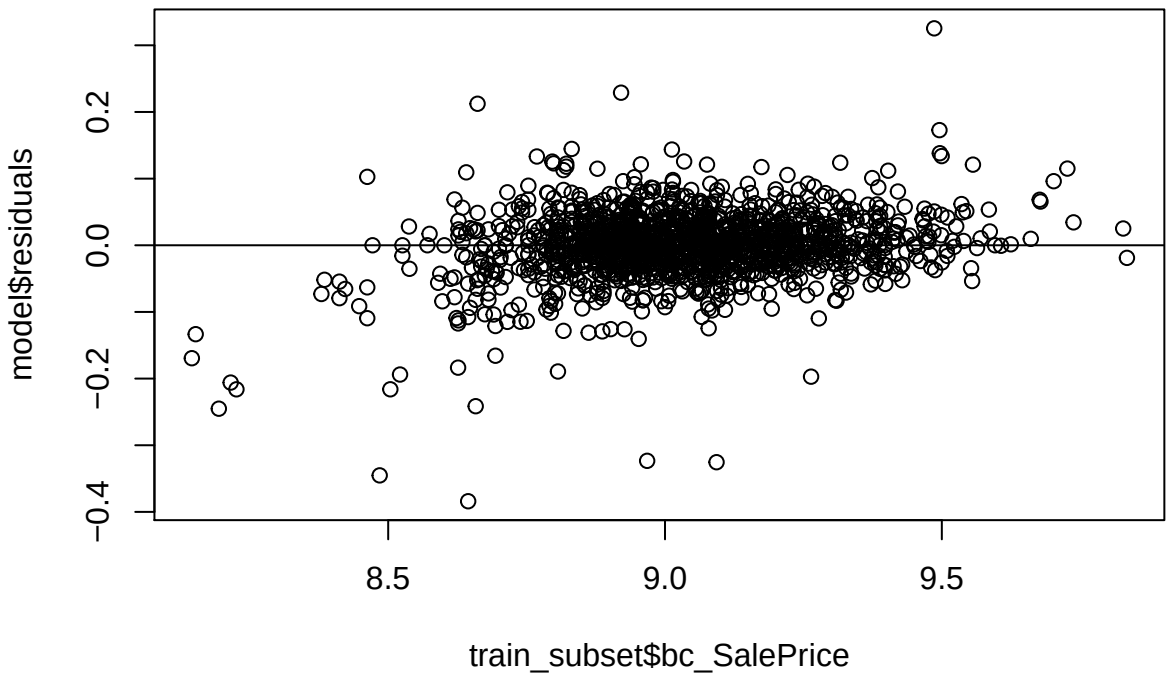
```
#p-value limit: 0.10 - Adjusted R-squared: 0.978595
#p-value limit: 0.05 - Adjusted R-squared: 0.9768811
#p-value limit: 0.01 - Adjusted R-squared: 0.9604068
#p-value limit: 0.005 - Adjusted R-squared: 0.9538743
#p-value limit: 0.001 - Adjusted R-squared: 0.9506303
summary5$adj.r.squared

#summary2
summary5

#number of variables - 396 (0.05), 175 (0.01), 130 (0.005), 119 (0.001)
length(summary5$coefficients[,4])
```

Did my first three linear models meet the necessary assumptions? Looking at the residual plots for each, we can see that each is fairly similar. While not perfect, each of the models does:

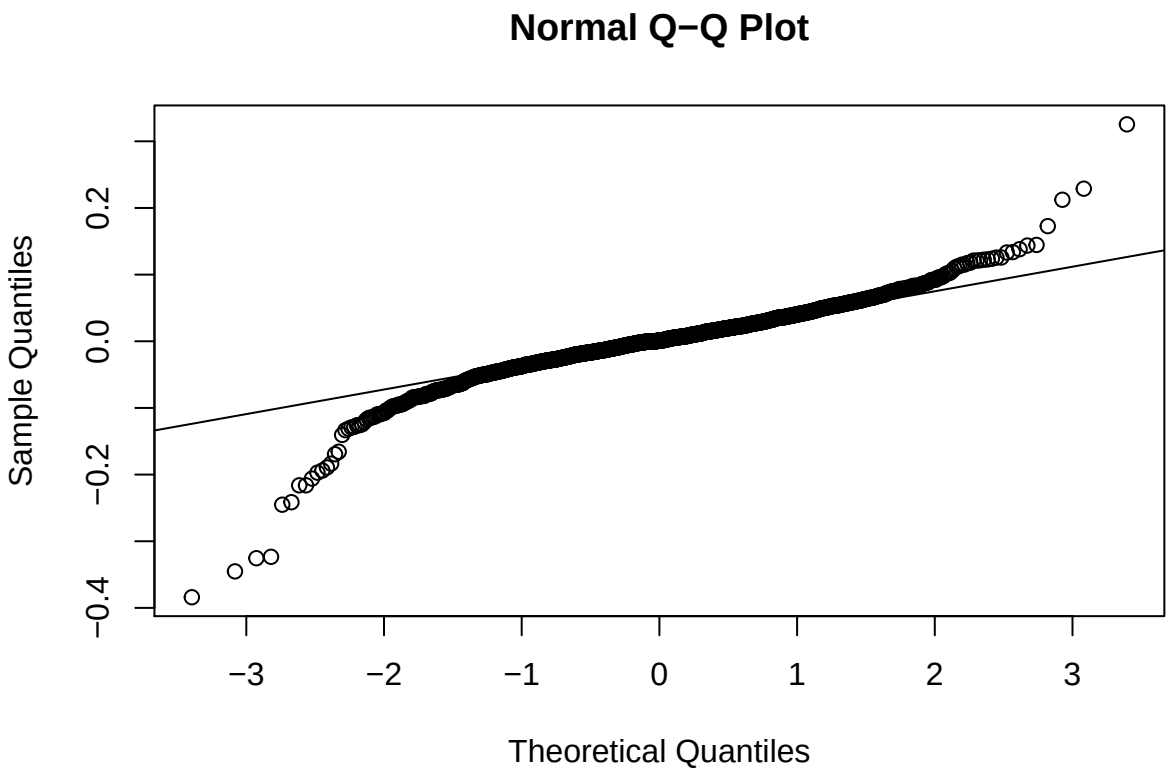
- have roughly constant variation of the residuals across the range of fitted values
- have roughly equal distribution of residuals across the X axis
- have no obvious pattern in the residuals that was missed by the regression
- have residuals that do follow a normal distribution for the most part, excepting the tails (which happens frequently)

In short, while not perfect, each of the models seems to work pretty well and meets the conditions for a linear model. With more time, I would work on trying to understand what is happening on the tails and see if this can be corrected in some way.

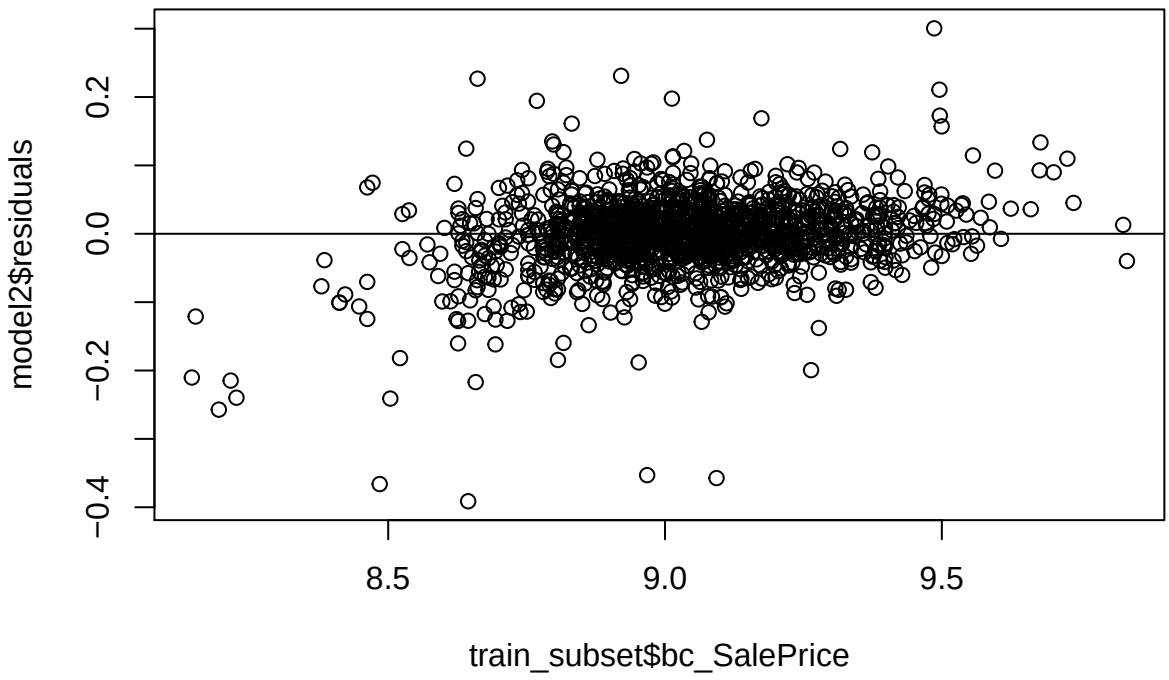
```
#model
plot(train_subset$bc_SalePrice,model$residuals)
abline(h=0)
```



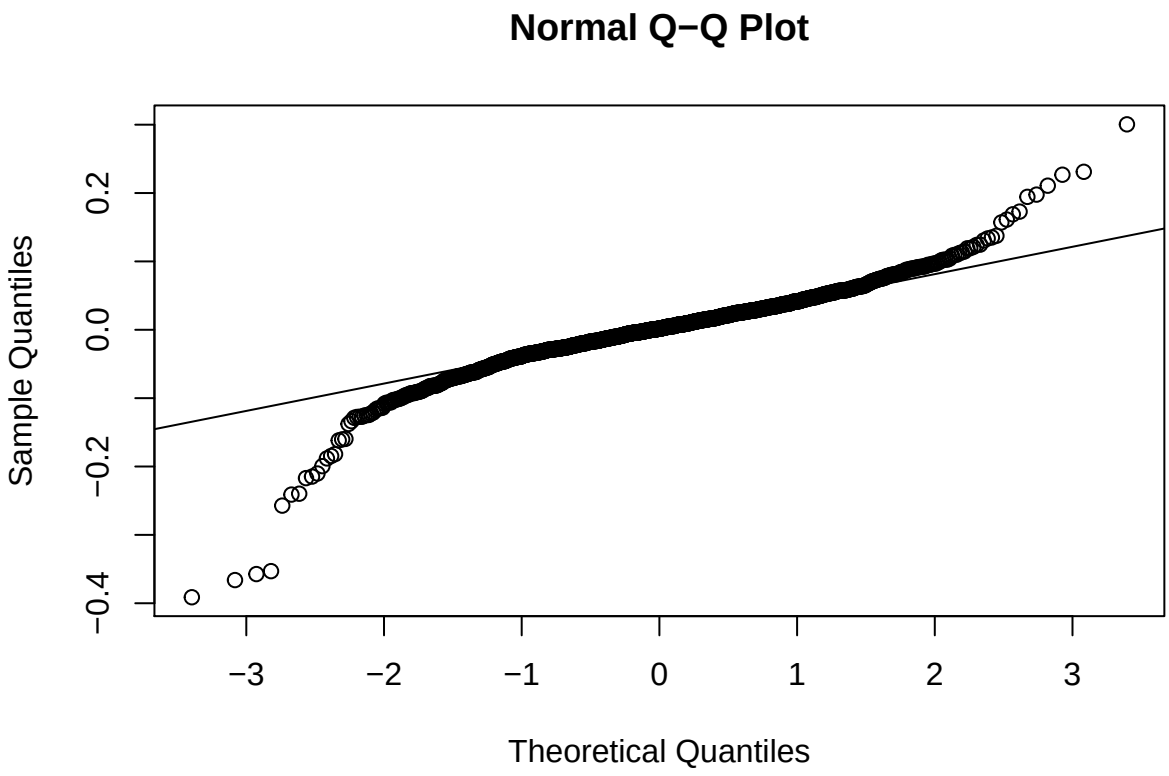

```
qqnorm(model$residuals)
qqline(model$residuals)
```



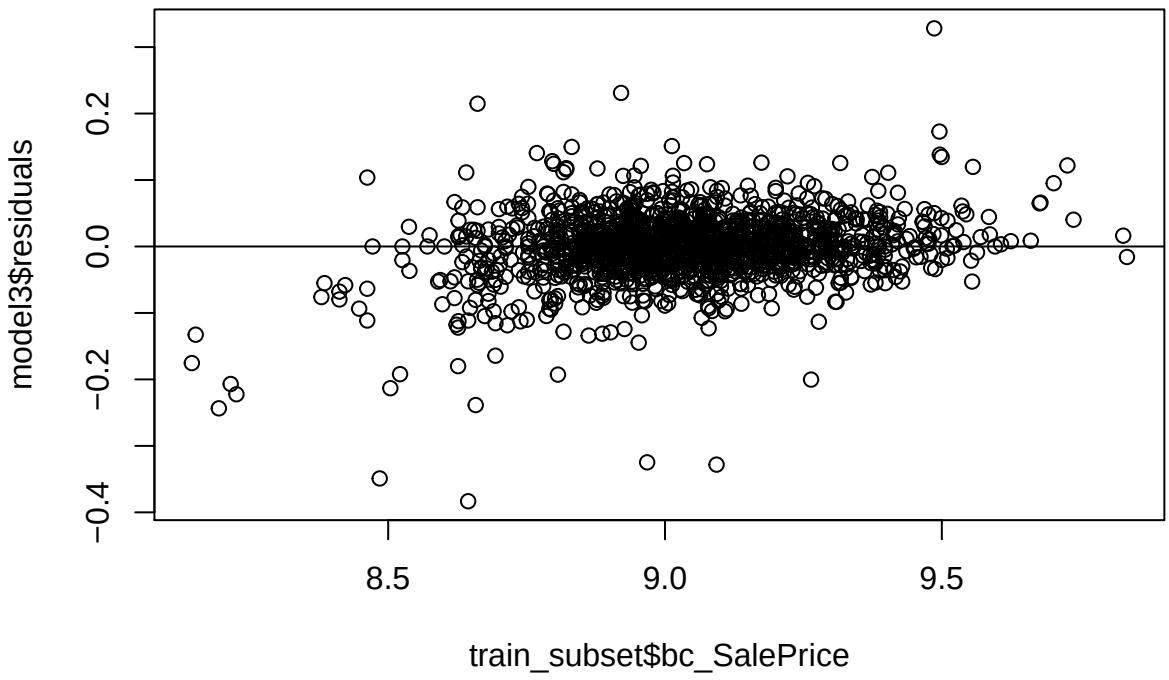

```
#model2
plot(train_subset$bc_SalePrice,model2$residuals)
abline(h=0)
```



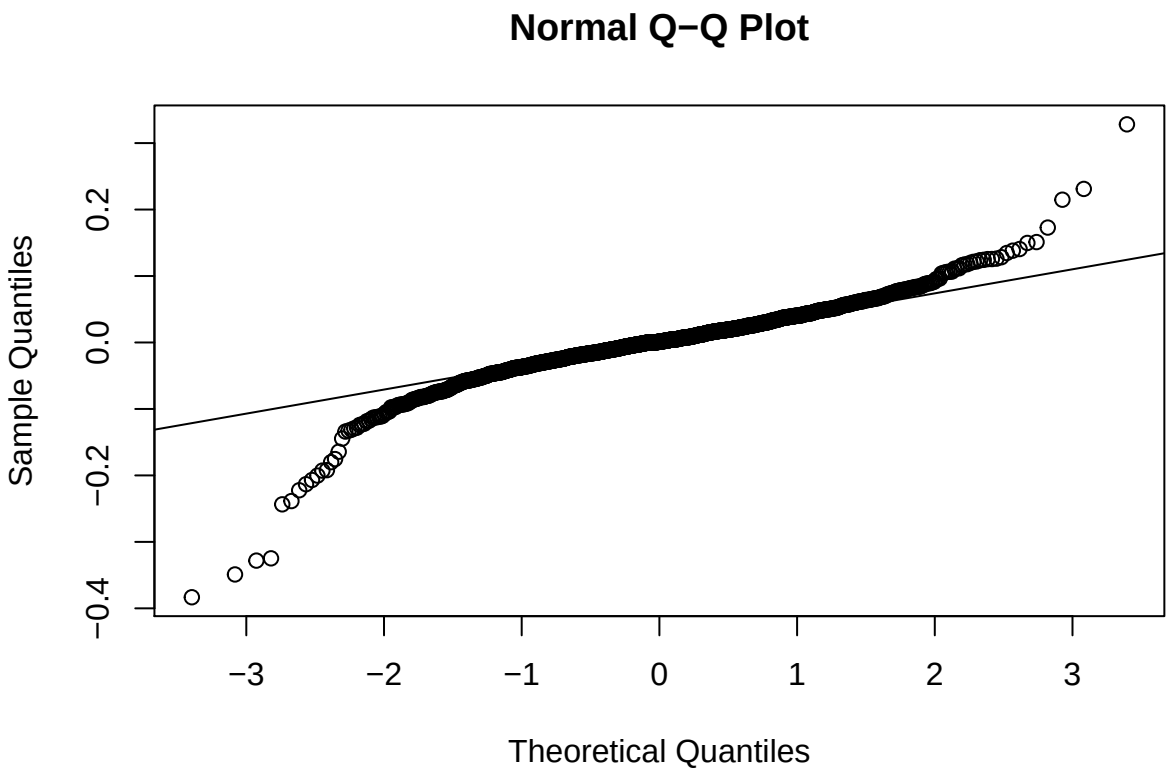

```
qqnorm(model2$residuals)
qqline(model2$residuals)
```




```
#model3
plot(train_subset$bc_SalePrice,model3$residuals)
abline(h=0)
```




```
qqnorm(model3$residuals)
qqline(model3$residuals)
```



Next I predicted on the *test_subset* and submitted to Kaggle for evaluation. I added some of the missing columns that *test_subset* did not have as a result of not having the same categorical values as *train_subset* did. Then I predicted the *bc_SalePrice* using the *test_subset* and my best model3. I transformed *bc_SalePrice* back into *SalePrice*. After checking for missing values, I wrote my output and submitted to Kaggle.

```
#add dummy columns for model to work
test_subset$SalePrice <-0
test_subset$Utilities_NoSeWa <- 0
test_subset$Condition2_RRNn <- 0
test_subset$Condition2_RRAn <- 0
test_subset$Condition2_RRAe <- 0
test_subset$HouseStyle_2.5Fin <- 0
test_subset$RoofMatl_Metal <- 0
test_subset$RoofMatl_Membran <- 0
test_subset$RoofMatl_Roll <- 0
test_subset$RoofMatl_ClyTile <- 0
test_subset$Exterior1st_Stone <- 0
test_subset$Exterior1st_ImStucc <- 0
test_subset$Exterior2nd_Other<- 0
test_subset$Heating_OthW<- 0
test_subset$Heating_Floor<- 0
test_subset$Electrical_Mix<- 0
test_subset$GarageQual_Ex<- 0
test_subset$PoolQC_Fa<- 0
test_subset$MiscFeature_TenC<- 0

#predict test
```

```
predictions<-predict(model3,test_subset)
prediction_df<-data.frame(cbind(test$Id,predictions))
names(prediction_df) <- c("Id", "SalePrice")

#transform bc_SalePrice back to SalePrice
prediction_df$SalePrice <- (prediction_df$SalePrice * bestLambda_Y + 1)^(1/bestLambda_Y)


########## final checks
#check how many rows missing
nrow(prediction_df[is.na(prediction_df$SalePrice),])
which(is.na(prediction_df$SalePrice))

#check if any values less than 0.  Assign substitute value
prediction_df$SalePrice[prediction_df$SalePrice < 0] <- median(train$SalePrice)

#output
write.csv(prediction_df,
          paste0("C:/Users/Andy/Desktop/Personal/Learning/CUNY/DATA605/HW/FinalProject/submission_",
                 str_replace_all(Sys.time(),"[: ]","_"),".csv"),
          row.names = FALSE)

#Submit to Kaggle
#https://www.kaggle.com/c/house-prices-advanced-regression-techniques/
```

Report your Kaggle.com user name and score

I submitted various outputs from several model trials, hoping to break into the top 50% by rank, which I eventually did. At the time of this writing, I had the rank, score, and percentile below:

- User Name: Andrew Carson
- Best Score: #1350, 0.13597
- Percentile: top 50.4% (1-1350/2723 teams)
- 


```
print(paste0("Percentile: ",1-1350/2723))
```

```
## [1] "Percentile: 0.504223283143592"
```

Conclusion

So what matters in determining a house price besides location, location, location? Without normalizing the variables it is a little difficult to say in terms of impact or contribution to the overall price. However, we can say, based on p-value, which variables are most highly correlated with the *SalePrice*. The top 10, based on using model2 (and ignoring the intercept), are:

```
mostImportant<-data.frame(sort(summary2$coefficients[,4], decreasing = FALSE)[2:11])
names(mostImportant)<- c("P-Value")
mostImportant
```

```
##                         P-Value
## X2ndFlrSF          5.397998e-62
## X1stFlrSF          1.793886e-45
## RoofMatl_WdShngl   2.974779e-41
## RoofMatl_CompShg   1.406629e-40
## `RoofMatl_Tar&Grv` 4.936624e-39
```

```
## RoofMatl_Membran   5.618631e-39
## RoofMatl_WdShake   1.709298e-37
## RoofMatl_Metal     9.461889e-37
## RoofMatl_Roll      1.176066e-36
## OverallCond        4.878451e-28
```

In short, how much square footage does the house have? More is better. What kind of roof does it have? Certain kinds (mebrane, metal) are better than others (Composite Shingle, Woodshake). What is the Overall condition of the house? A higher ranking is better. Other high ranking variables not displayed are zoning (commercial zoning lowers the price), overall quality (higher is better), basement square footage (more is better), and lot area (more is better). None of these is surprising and each makes sense with our own experiences of what most people value in a house.