Predictive Model of Selling Price For Homes in Ames, Iowa

Kelly Radimer

Background

For this project, I play the role of a statistical consultant working for a real estate investment firm, tasked with developing a model to predict the selling price of a given home in Ames, Iowa, so as to identify homes that would be good investments for the firm.

Training Data and relevant packages

The data were randomly divided into three separate pieces: a training data set, a testing data set, and a validation data set, to allow me to better assess the model produced. First, I created a model using the training data set.

```
load("ames_train.Rdata")
```

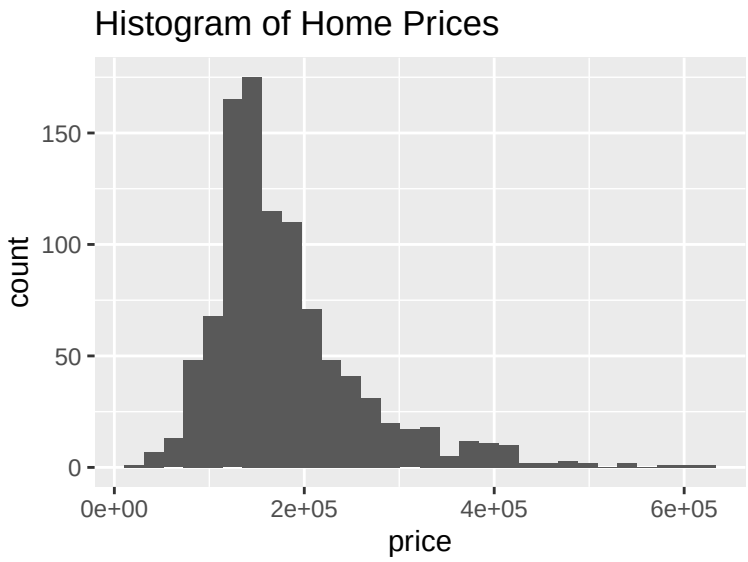
Load the necessary packages:

```
library(statsr)
library(dplyr)
library(BAS)
library(MASS)
library(devtools)
library(ggplot2)
```

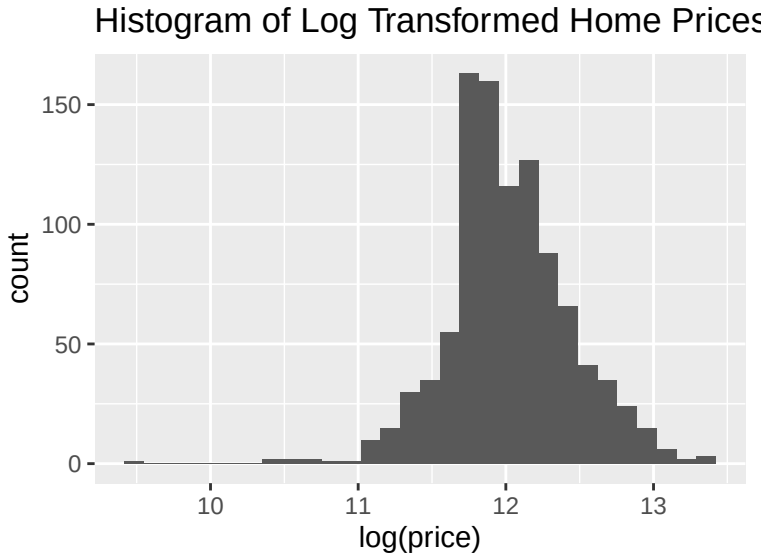
Part 1 - Exploratory Data Analysis (EDA)

Home prices and home areas tend to be right skewed, so our models will probably perform better if we take the log of those fields.

```
ggplot(data=ames_train, aes(x=price))+geom_histogram(bins = 30)+
    ggtitle("Histogram of Home Prices")
```

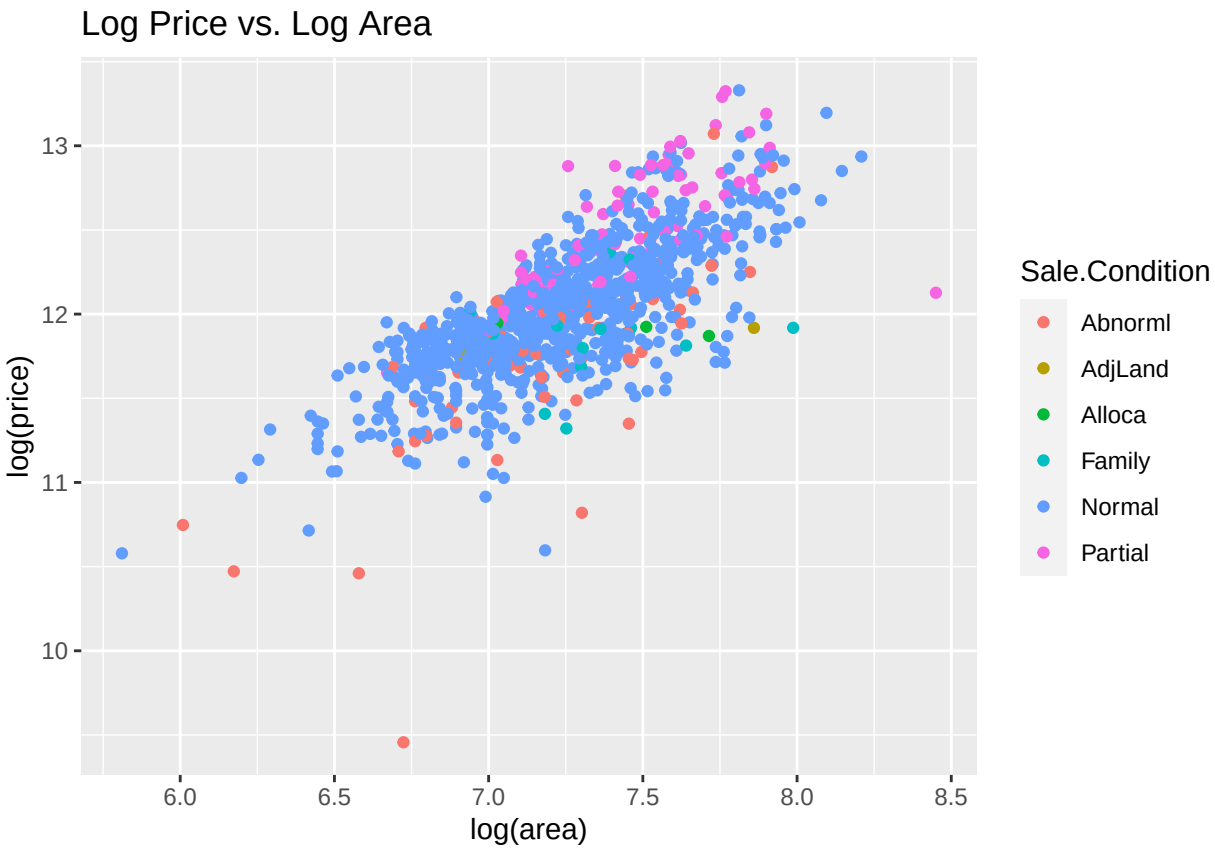



```
ggplot(data=ames_train, aes(x=log(price)))+geom_histogram(bins = 30)+
    ggtitle("Histogram of Log Transformed Home Prices")
```



Past analysis has suggested that houses sold under abnormal and partial conditions may not be useful in building our models. Let's have a look at how these homes compare to those sold under other conditions.

```
qplot(log(area), log(price), data = ames_train, colour=Sale.Condition)+
    ggtitle("Log Price vs. Log Area")
```




```
ames_train%>%filter(Sale.Condition=="Abnorml")%>%summarise(mean=mean(price), sd=sd(price), count=n())
```

```
## # A tibble: 1 x 3
##      mean     sd count
##     <dbl>  <dbl> <int>
## 1 143740. 76042.    61
```

```
ames_train%>%filter(Sale.Condition=="Partial")%>%summarise(mean=mean(price), sd=sd(price), count=n())
```

```
## # A tibble: 1 x 3
##      mean      sd count
##     <dbl>   <dbl> <int>
## 1 285172. 107858.    82
```

```
ames_train%>%filter(Sale.Condition!="Abnorml")%>%filter(Sale.Condition!="Partial")%>%summarise(mean=mean
```

```
## # A tibble: 1 x 3
##      mean     sd count
##     <dbl>  <dbl> <int>
## 1 173906. 71660.   857
```

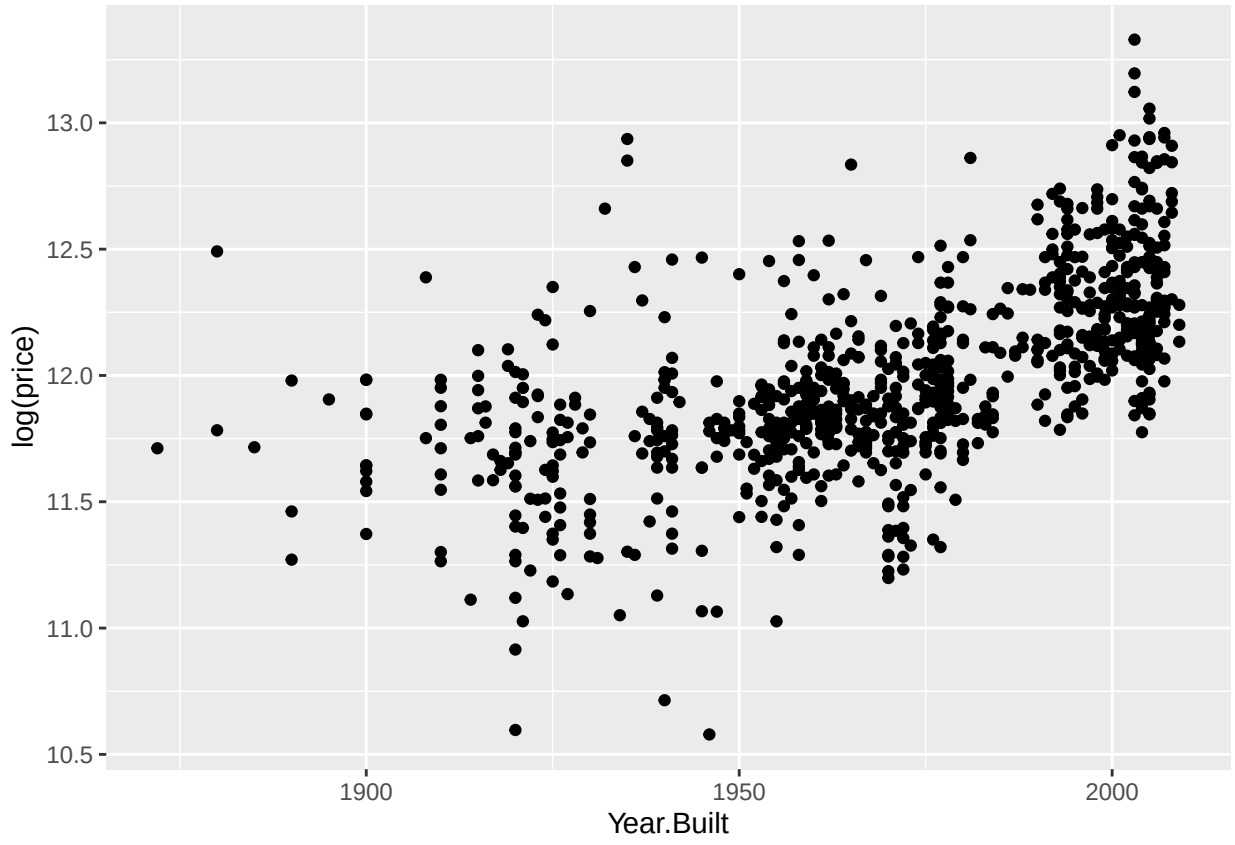
Homes with sale condition "Abnorml" tended to sell for less than homes of similar area with other sale conditions. Those with sale condition "Partial" tended to sell for more than homes with similar area sold

under Normal conditions. The code book explains that Partial means "Home was not completed when last assessed (associated with New Homes)." Abnormal sales include "trade, foreclosure, short sale." Though the investment firm might purchase foreclosures and short sales, what they're really interested in is the resale value of these homes, which would be better represented by homes with normal sale conditions, so we will remove the 61 Abnormal and 82 Partial sales from the data set.

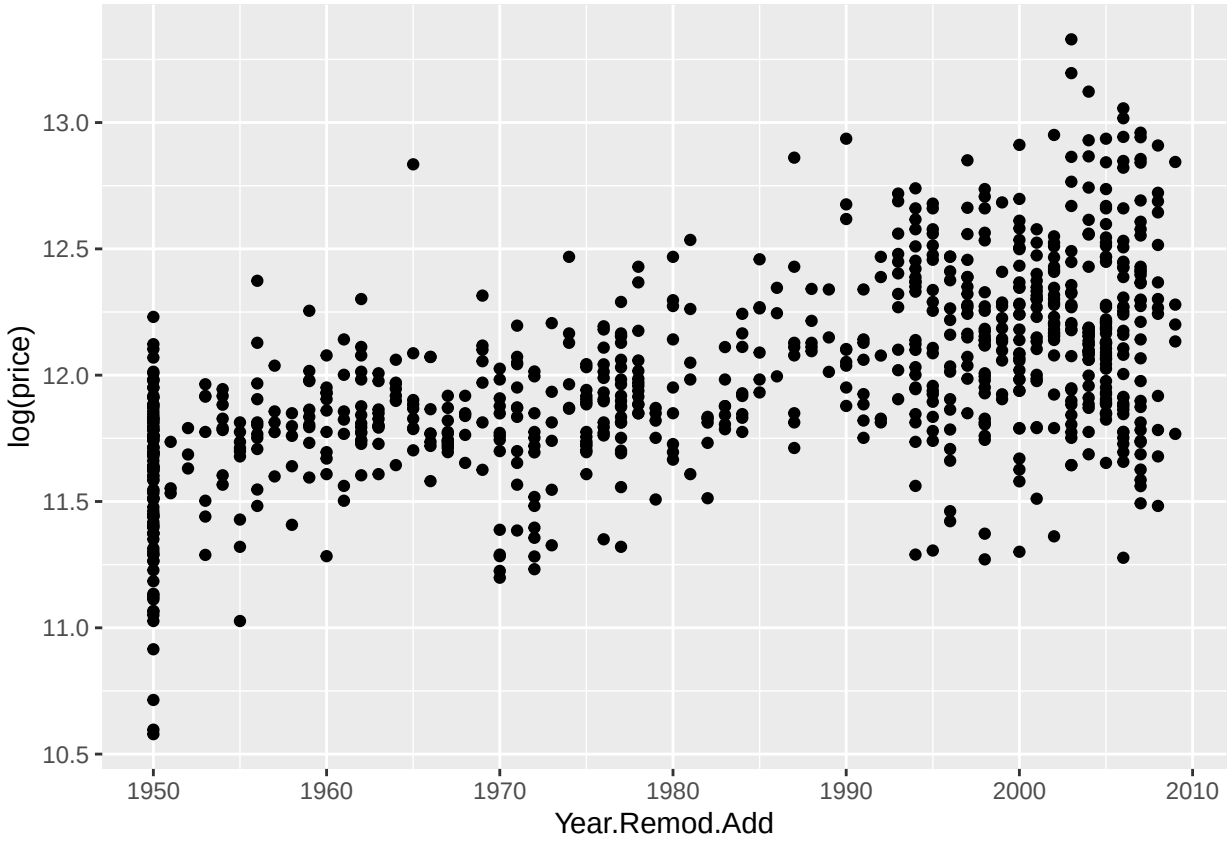
```
ames_train<-ames_train%>%filter(Sale.Condition!="Abnorml")%>%filter(Sale.Condition!="Partial")
```

Next we will look at scatterplots of variables that from past experience we believe are good single predictors of price: year built, year of remodel, bedrooms above ground, log(lot area), and overall quality. Our scatterplot above has already shown us that log(area) is also a good predictor.

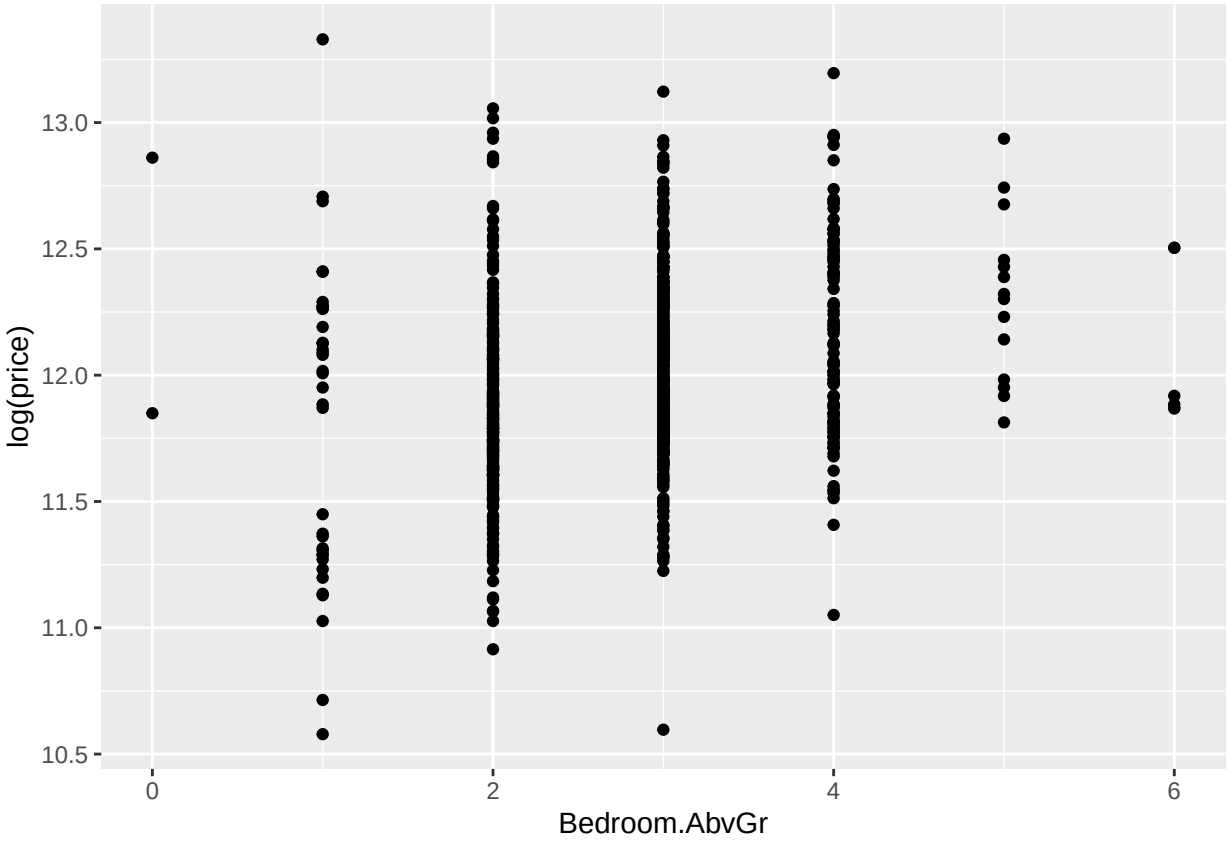
```
ggplot(data=ames_train, aes(x=Year.Built, y=log(price)))+geom_point()
```



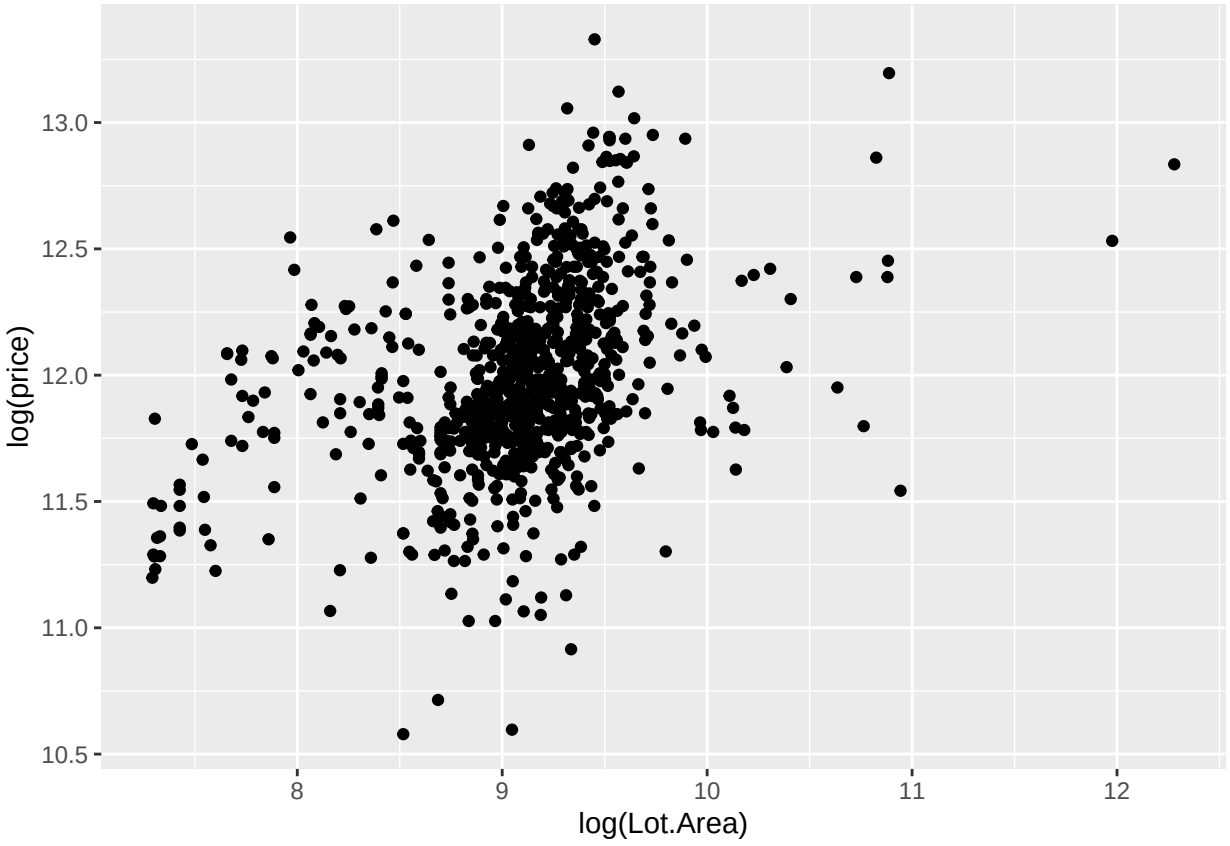

```
ggplot(data=ames_train, aes(x=Year.Remod.Add, y=log(price)))+geom_point()
```



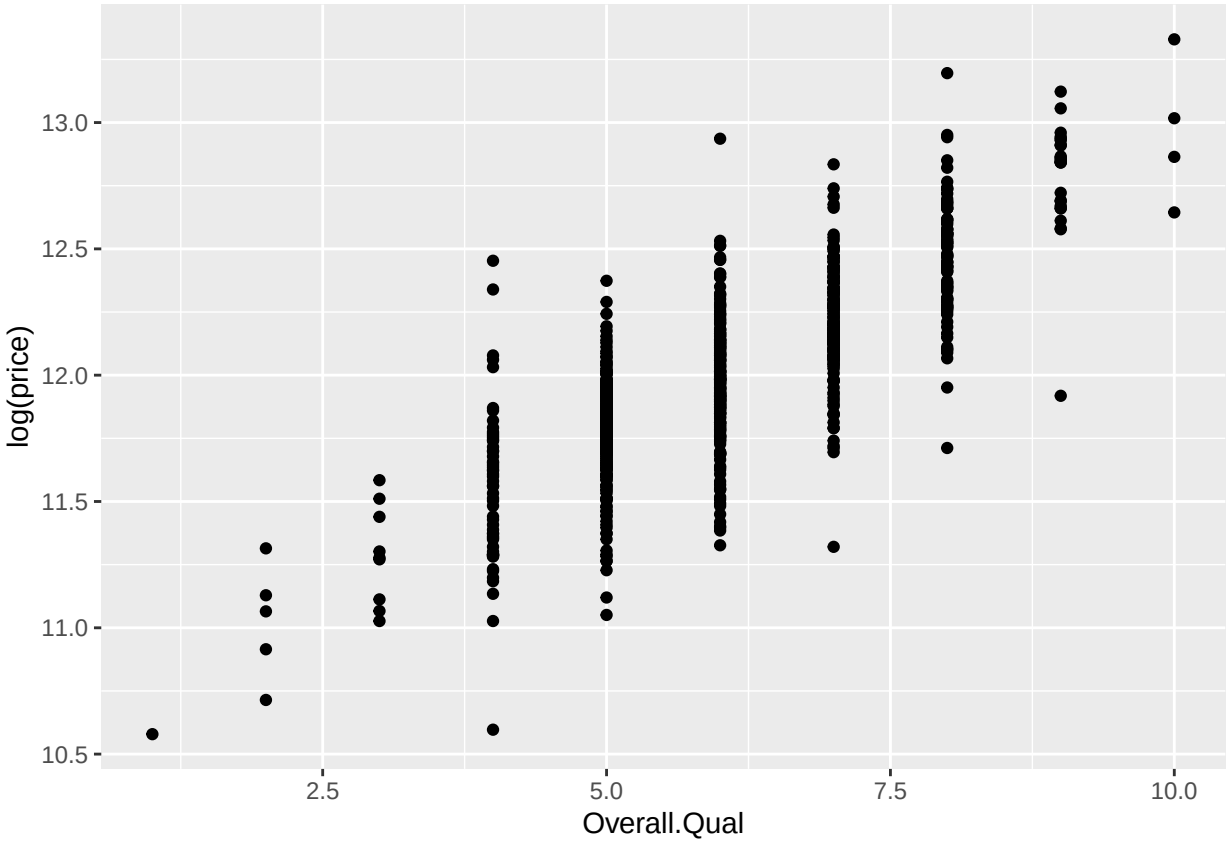

```
ggplot(data=ames_train, aes(x=Bedroom.AbvGr, y=log(price)))+geom_point()
```




```
ggplot(data=ames_train, aes(x=log(Lot.Area), y=log(price)))+geom_point()
```




```
ggplot(data=ames_train, aes(x=Overall.Qual, y=log(price)))+geom_point()
```



All of these variables do appear to have at least moderately strong correlation with price.

Part 2 - Development and assessment of an initial model, following a semi-guided process of analysis

Section 2.1 An Initial Model

For our initial model we will begin with the quantitative variables that we identified as being good single predictors in our EDA above: year built, year of remodel, bedrooms above ground, log(lot area), overall quality, and log(area). It is reasonable to believe that houses that were more recently built, more recently remodeled, having more bedrooms, having a bigger lot, having a larger area and being of higher quality will be worth more. Additionally, we will consider lot slope, exterior Quality, Central Cooling and kitchen quality, as people may be concerned with the slope of the land their house is on, the quality of the home's exterior, whether the house has central AC and how nice the kitchen is.

```
model_full=lm(log(price)~Year.Built+Year.Remod.Add+log(area)+Bedroom.AbvGr+log(Lot.Area)+Overall.Qual+K
summary(model_full)
```

```
##
## Call:
## lm(formula = log(price) ~ Year.Built + Year.Remod.Add + log(area) +
##     Bedroom.AbvGr + log(Lot.Area) + Overall.Qual + Kitchen.Qual +
##     Land.Slope + Exter.Qual + Central.Air, data = ames_train)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.85183 -0.07574  0.01036  0.08476  0.51653
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.1639089  0.7165926  -0.229 0.819132
## Year.Built      0.0024911  0.0002294  10.857  < 2e-16 ***
## Year.Remod.Add  0.0012154  0.0003311   3.671 0.000257 ***
## log(area)       0.4377137  0.0251463  17.407  < 2e-16 ***
## Bedroom.AbvGr  -0.0332322  0.0079404  -4.185 3.15e-05 ***
## log(Lot.Area)   0.1431066  0.0099187  14.428  < 2e-16 ***
## Overall.Qual    0.0858357  0.0062563  13.720  < 2e-16 ***
## Kitchen.QualFa -0.1686934  0.0464373  -3.633 0.000297 ***
## Kitchen.QualGd -0.0993812  0.0301948  -3.291 0.001039 **
## Kitchen.QualPo -0.0622966  0.1423824  -0.438 0.661839
## Kitchen.QualTA -0.1531060  0.0327123  -4.680 3.34e-06 ***
## Land.SlopeMod   0.0814814  0.0254315   3.204 0.001407 **
## Land.SlopeSev   0.0113951  0.0712308   0.160 0.872940
## Exter.QualFa   -0.1991913  0.0654422  -3.044 0.002409 **
## Exter.QualGd   -0.0985533  0.0419993  -2.347 0.019180 *
## Exter.QualTA   -0.1022949  0.0454213  -2.252 0.024571 *
## Central.AirY    0.1886575  0.0240346   7.849 1.27e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1367 on 840 degrees of freedom
## Multiple R-squared:  0.8726, Adjusted R-squared:  0.8701
## F-statistic: 359.5 on 16 and 840 DF,  p-value: < 2.2e-16
```

This model appears to be a good starting place. All predictors are statistically significant in at least one category.

Section 2.2 Model Selection

We will begin by trying out AIC model selection to see if there are any predictors that we can eliminate.

```
model_AIC=stepAIC(model_full, direction = "backward", k = 2, trace = TRUE)
```

```
## Start:  AIC=-3393.6
## log(price) ~ Year.Built + Year.Remod.Add + log(area) + Bedroom.AbvGr +
##     log(Lot.Area) + Overall.Qual + Kitchen.Qual + Land.Slope +
##     Exter.Qual + Central.Air
##
##                  Df Sum of Sq    RSS     AIC
## <none>                        15.704 -3393.6
## - Exter.Qual      3    0.1845 15.889 -3389.6
## - Land.Slope      2    0.1919 15.896 -3387.2
## - Year.Remod.Add  1    0.2519 15.956 -3382.0
## - Bedroom.AbvGr   1    0.3275 16.031 -3377.9
## - Kitchen.Qual    4    0.4934 16.197 -3375.1
```

```
## - Central.Air     1    1.1519 16.856 -3334.9
## - Year.Built      1    2.2036 17.907 -3283.1
## - Overall.Qual    1    3.5190 19.223 -3222.3
## - log(Lot.Area)   1    3.8917 19.596 -3205.9
## - log(area)       1    5.6645 21.368 -3131.6
```

```
summary(model_AIC)
```

```
##
## Call:
## lm(formula = log(price) ~ Year.Built + Year.Remod.Add + log(area) +
##     Bedroom.AbvGr + log(Lot.Area) + Overall.Qual + Kitchen.Qual +
##     Land.Slope + Exter.Qual + Central.Air, data = ames_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85183 -0.07574  0.01036  0.08476  0.51653
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.1639089  0.7165926  -0.229 0.819132
## Year.Built      0.0024911  0.0002294  10.857  < 2e-16 ***
## Year.Remod.Add  0.0012154  0.0003311   3.671 0.000257 ***
## log(area)       0.4377137  0.0251463  17.407  < 2e-16 ***
## Bedroom.AbvGr  -0.0332322  0.0079404  -4.185 3.15e-05 ***
## log(Lot.Area)   0.1431066  0.0099187  14.428  < 2e-16 ***
## Overall.Qual    0.0858357  0.0062563  13.720  < 2e-16 ***
## Kitchen.QualFa -0.1686934  0.0464373  -3.633 0.000297 ***
## Kitchen.QualGd -0.0993812  0.0301948  -3.291 0.001039 **
## Kitchen.QualPo -0.0622966  0.1423824  -0.438 0.661839
## Kitchen.QualTA -0.1531060  0.0327123  -4.680 3.34e-06 ***
## Land.SlopeMod   0.0814814  0.0254315   3.204 0.001407 **
## Land.SlopeSev   0.0113951  0.0712308   0.160 0.872940
## Exter.QualFa   -0.1991913  0.0654422  -3.044 0.002409 **
## Exter.QualGd   -0.0985533  0.0419993  -2.347 0.019180 *
## Exter.QualTA   -0.1022949  0.0454213  -2.252 0.024571 *
## Central.AirY    0.1886575  0.0240346   7.849 1.27e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1367 on 840 degrees of freedom
## Multiple R-squared:  0.8726, Adjusted R-squared:  0.8701
## F-statistic: 359.5 on 16 and 840 DF,  p-value: < 2.2e-16
```

Stepwise AIC did not eliminate any of our variables. Let's try BIC.

```
model_BIC=stepAIC(model_full, direction = "backward", k = log(857), trace = TRUE)
```

```
## Start:  AIC=-3312.79
## log(price) ~ Year.Built + Year.Remod.Add + log(area) + Bedroom.AbvGr +
##     log(Lot.Area) + Overall.Qual + Kitchen.Qual + Land.Slope +
##     Exter.Qual + Central.Air
```

```
##
##                   Df Sum of Sq    RSS     AIC
## - Exter.Qual       3    0.1845 15.889 -3323.0
## - Land.Slope       2    0.1919 15.896 -3315.9
## - Kitchen.Qual     4    0.4934 16.197 -3313.3
## <none>                         15.704 -3312.8
## - Year.Remod.Add   1    0.2519 15.956 -3305.9
## - Bedroom.AbvGr    1    0.3275 16.031 -3301.9
## - Central.Air      1    1.1519 16.856 -3258.9
## - Year.Built       1    2.2036 17.907 -3207.0
## - Overall.Qual     1    3.5190 19.223 -3146.3
## - log(Lot.Area)    1    3.8917 19.596 -3129.8
## - log(area)        1    5.6645 21.368 -3055.6
##
## Step:  AIC=-3323.03
## log(price) ~ Year.Built + Year.Remod.Add + log(area) + Bedroom.AbvGr +
##     log(Lot.Area) + Overall.Qual + Kitchen.Qual + Land.Slope +
##     Central.Air
##
##                   Df Sum of Sq    RSS     AIC
## - Land.Slope       2    0.2045 16.093 -3325.6
## <none>                         15.889 -3323.0
## - Year.Remod.Add   1    0.2904 16.179 -3314.3
## - Bedroom.AbvGr    1    0.3617 16.250 -3310.5
## - Kitchen.Qual     4    0.8954 16.784 -3303.1
## - Central.Air      1    1.3709 17.259 -3258.9
## - Year.Built       1    2.3297 18.218 -3212.5
## - log(Lot.Area)    1    4.0612 19.950 -3134.7
## - Overall.Qual     1    4.1875 20.076 -3129.3
## - log(area)        1    5.6733 21.562 -3068.1
##
## Step:  AIC=-3325.58
## log(price) ~ Year.Built + Year.Remod.Add + log(area) + Bedroom.AbvGr +
##     log(Lot.Area) + Overall.Qual + Kitchen.Qual + Central.Air
##
##                   Df Sum of Sq    RSS     AIC
## <none>                         16.093 -3325.6
## - Year.Remod.Add   1    0.2973 16.390 -3316.6
## - Bedroom.AbvGr    1    0.4387 16.532 -3309.3
## - Kitchen.Qual     4    0.9494 17.042 -3303.5
## - Central.Air      1    1.3761 17.469 -3262.0
## - Year.Built       1    2.3521 18.445 -3215.4
## - Overall.Qual     1    4.0437 20.137 -3140.2
## - log(Lot.Area)    1    4.5926 20.686 -3117.2
## - log(area)        1    5.9450 22.038 -3062.9
```

```
summary(model_BIC)
```

```
##
## Call:
## lm(formula = log(price) ~ Year.Built + Year.Remod.Add + log(area) +
##     Bedroom.AbvGr + log(Lot.Area) + Overall.Qual + Kitchen.Qual +
##     Central.Air, data = ames_train)
##
```

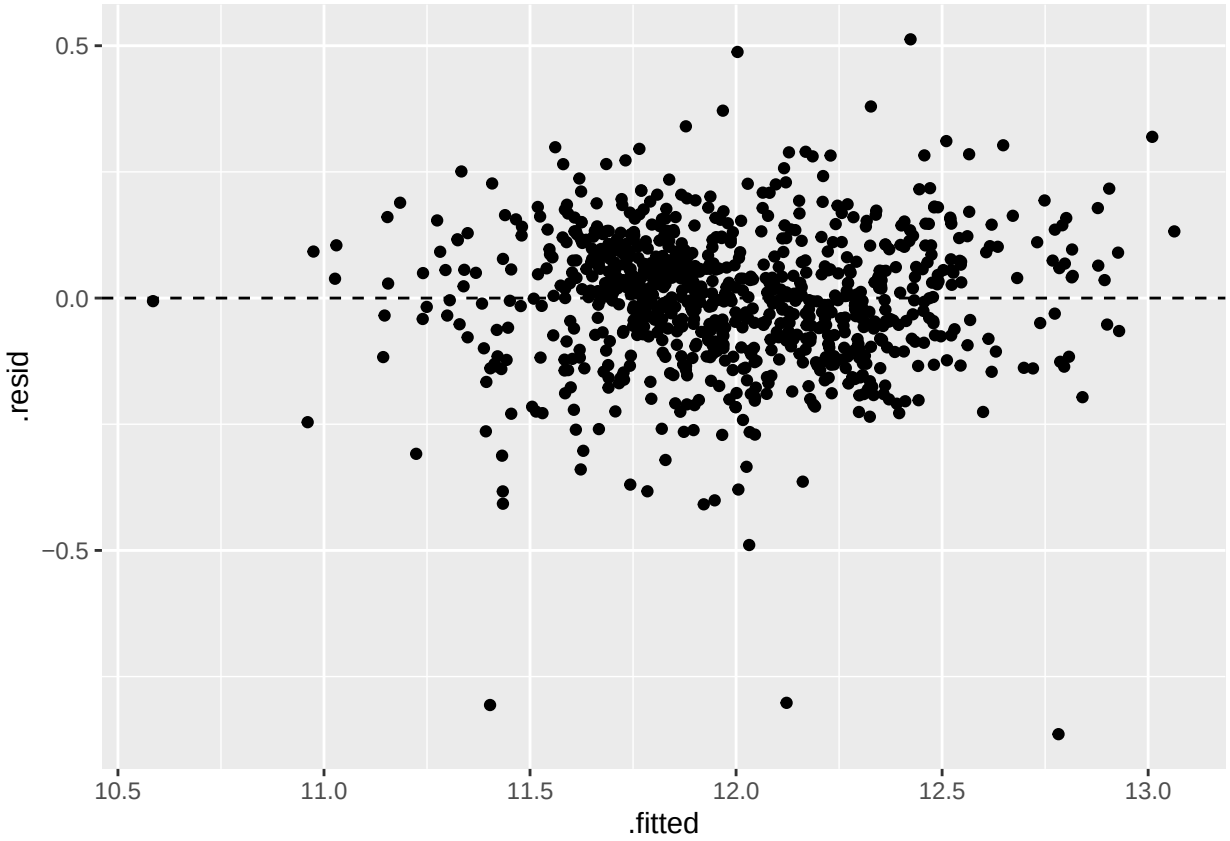
```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.86406 -0.07773  0.00930  0.08462  0.51264
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.5463646  0.6957010  -0.785    0.432
## Year.Built      0.0025089  0.0002258  11.113  < 2e-16 ***
## Year.Remod.Add  0.0013116  0.0003320   3.951 8.44e-05 ***
## log(area)       0.4427042  0.0250569  17.668  < 2e-16 ***
## Bedroom.AbvGr  -0.0373823  0.0077887  -4.800 1.88e-06 ***
## log(Lot.Area)   0.1494194  0.0096220  15.529  < 2e-16 ***
## Overall.Qual    0.0871765  0.0059828  14.571  < 2e-16 ***
## Kitchen.QualFa -0.2035973  0.0434594  -4.685 3.27e-06 ***
## Kitchen.QualGd -0.1422876  0.0245713  -5.791 9.88e-09 ***
## Kitchen.QualPo -0.0923101  0.1423457  -0.648    0.517
## Kitchen.QualTA -0.1939070  0.0277201  -6.995 5.39e-12 ***
## Central.AirY    0.1988140  0.0233890   8.500  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.138 on 845 degrees of freedom
## Multiple R-squared:  0.8694, Adjusted R-squared:  0.8677
## F-statistic: 511.4 on 11 and 845 DF,  p-value: < 2.2e-16
```

BIC model selection eliminated Exterior Quality and Land Slope. The results of the two model selection methods were not consistent. BIC placed more emphasis on obtaining a parsimonious model, whereas AIC placed greater value on having the best possible predictions. Our Adjusted R-squared was 0.8701 with the additional two predictors and 0.8677 without them. If we can simplify our model by 2 predictors and only lose 0.0024 in our Adjusted R-squared, that's probably worth it, so we will use the BIC model.

Section 2.3 Initial Model Residuals

To assess the performance of our BIC model, let's have a look at the residual plot.

```
ggplot(model_BIC, aes(x=.fitted, y=.resid))+geom_point()+geom_hline(yintercept = 0, linetype="dashed")
```



The model is, overall, a good fit for the data, as there is no clear form to the residual plot. The variance appears to be approximately equal across the spectrum for all fitted values, so we don't need to be worried about our predictions for especially high or low priced homes to be more or less accurate than other homes. There are three points with unusually low residuals, meaning that our model dramatically overestimated the price of these three homes. If we notice a similar phenomena in the test data, we may want to figure out what characteristics these homes have in common so that the investment firm can avoid overpaying for such homes.

Section 2.4 Initial Model RMSE

How far off do our predictions of home price tend to be? To answer this we can calculate the root mean square error.

```
predict_BIC<-exp(predict(model_BIC, ames_train))
resid_BIC<-ames_train$price - predict_BIC
rmse_BIC<-sqrt(mean(resid_BIC^2))
rmse_BIC
```

```
## [1] 25862.69
```

On average, our predictions of home prices are off by $25,862.69.

Section 2.5 Overfitting

To avoid using a model that is overly-tuned to specifically fit the training data, we will check the performance of our model on out-of-sample data.

```
load("ames_test.Rdata")
```

As we did with our training data, we will filter out any partial or abnormal sales from the test data.

```
ames_test<-ames_test%>%filter(Sale.Condition!="Abnorml")%>%filter(Sale.Condition!="Partial")
```

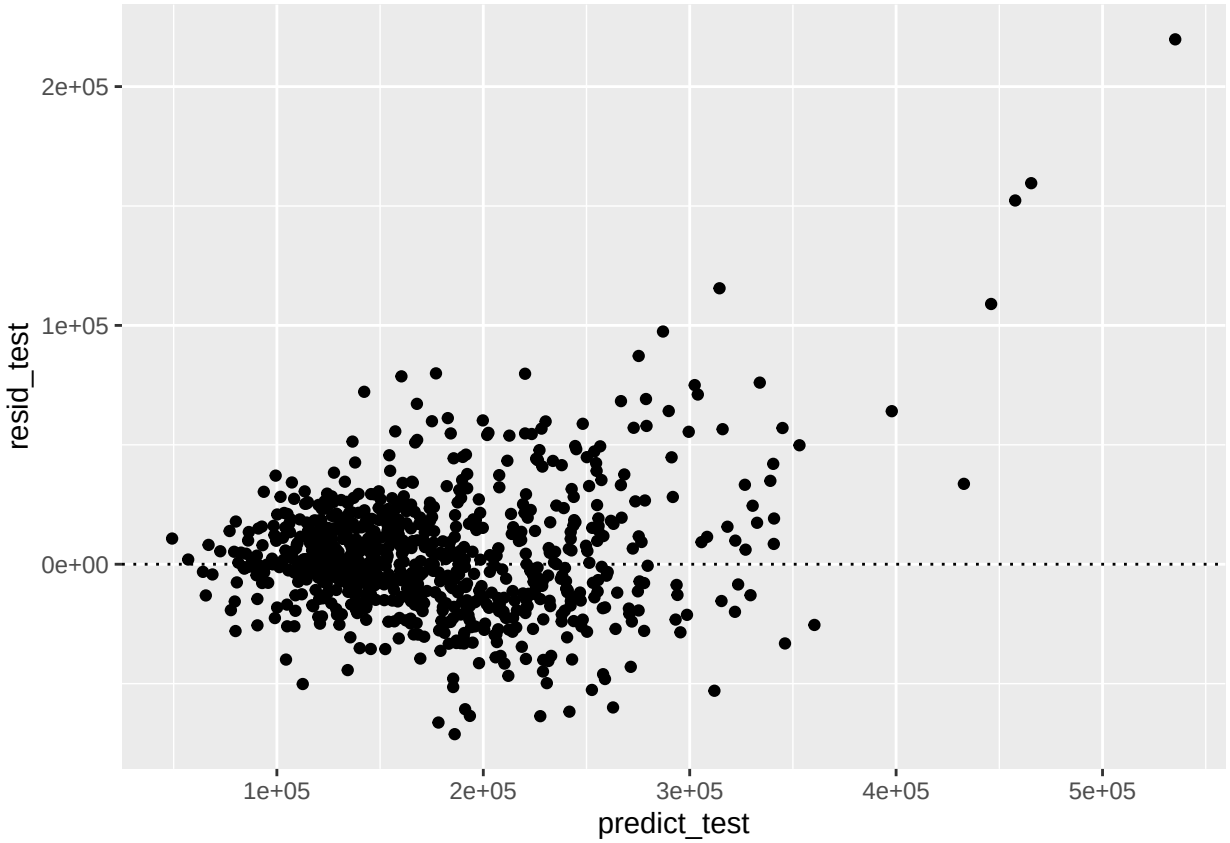
The number of observations in the set of test data didn't change when we filtered out abnormal and partial sales, so it would appear that there weren't any partial or abnormal sales in the test data. Now let's see how accurate our predictions are when applying our training model to the test data.

```
predict_test<-exp(predict(model_BIC, ames_test))
resid_test<-ames_test$price - predict_test
rmse_test<-sqrt(mean(resid_test^2))
rmse_test
```

```
## [1] 26757.99
```

For the test data, our predictions are off by an average of $26,757.99, which is a little more than our predictions were off for the training data, but not a huge amount. Let's check out a residual plot.

```
df=data.frame(predict_test, resid_test)
ggplot(df, aes(x=predict_test, y=resid_test))+geom_point()+geom_hline(yintercept = 0, linetype="dotted")
```



The model performs pretty well on homes where our predictions of price ranged from 0 to 450,000 dollars, but for the highest predicted prices, our model performs poorly, with increasingly large residuals for the three highest estimated priced homes. In all three of these cases our predictions were too low, in one case by over $200,000. These three homes were ID numbers 640, 326, and 8. Let's see if these homes have anything in particular in common to see if there is another variable that we should be taking into account in our final model.

```
test_outliers<-ames_test[c(8, 326, 640), 1:81]
```

All of these homes have three-car garages, at least one fireplace, an open porch and a wood deck, so we should see whether these would be significant variables to include in our model. Also, they all seem to have a lot of bathrooms, so let's include those as well.

Part 3 Development of a Final Model

Section 3.1 Final Model

In order to choose the best possible coefficients for our predictors, let's combine the test and training data together into a single dataset so that we are using all available data to create our model.

```
ames_combo<-rbind(ames_test, ames_train)
```

Let's add a new variable that will give the total number of bathrooms, multiplying the number of half baths by 0.5.

```
ames_combo <- ames_combo %>%
  mutate(Total.Baths=ames_combo$Full.Bath+.5*ames_combo$Half.Bath)
```

Additionally, there is one home in the data set with no garage, which is giving us a value of NA for Garage.Cars.

```
summary(ames_combo$Garage.Cars)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   1.000   2.000   1.745   2.000   5.000       1
```

Since there is no garage, this home has garage capacity for 0 cars, so I will code this as a 0 car garage.

```
ames_combo$Garage.Cars[which(is.na(ames_combo$Garage.Cars))]<-0
summary(ames_combo$Garage.Cars)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   2.000   1.744   2.000   5.000
```

As discussed below, I am also going to include an interaction variable between the number of bedrooms and the area of the home. We'll start by centering the variables.

```
ames_combo<-ames_combo%>%mutate(BedsC=ames_combo$Bedroom.AbvGr - mean(ames_combo$Bedroom.AbvGr), areaC=a
```

Next we'll multiply the centered variables for bedrooms and area.

```
ames_combo<-ames_combo%>%mutate(Beds.Area=ames_combo$BedsC * ames_combo$areaC)
```

Now we'll craft our final model. Discussion of the means by which we arrived at this model can be found in the sections below.

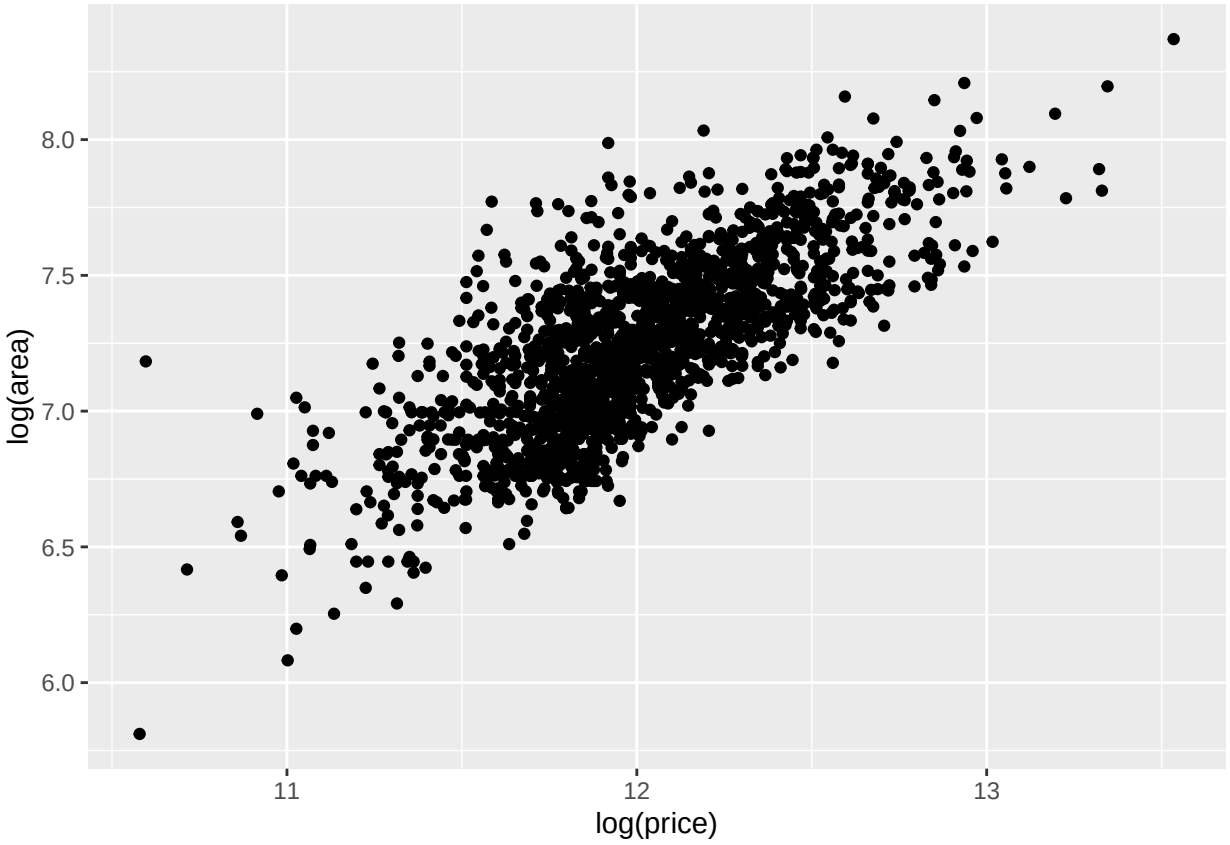
```
combo_interact=lm(log(price)~Year.Built+Year.Remod.Add+log(area)+Bedroom.AbvGr+log(Lot.Area)+Overall.Qua
summary(combo_interact)
```

```
##
## Call:
## lm(formula = log(price) ~ Year.Built + Year.Remod.Add + log(area) +
##     Bedroom.AbvGr + log(Lot.Area) + Overall.Qual + Kitchen.Qual +
##     Central.Air + Total.Baths + Fireplaces + Garage.Cars + Wood.Deck.SF +
##     Beds.Area, data = ames_combo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84729 -0.07381  0.00535  0.08162  0.43314
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.467e-01  5.245e-01   0.280 0.779701
## Year.Built      2.292e-03  1.674e-04  13.690  < 2e-16 ***
## Year.Remod.Add  1.308e-03  2.169e-04   6.030 2.02e-09 ***
## log(area)       4.351e-01  2.078e-02  20.941  < 2e-16 ***
## Bedroom.AbvGr  -2.382e-02  5.391e-03  -4.419 1.06e-05 ***
## log(Lot.Area)   1.268e-01  6.681e-03  18.986  < 2e-16 ***
## Overall.Qual    8.184e-02  3.997e-03  20.474  < 2e-16 ***
## Kitchen.QualFa -2.112e-01  2.919e-02  -7.234 7.13e-13 ***
## Kitchen.QualGd -1.348e-01  1.641e-02  -8.215 4.24e-16 ***
## Kitchen.QualPo -9.118e-02  1.299e-01  -0.702 0.482720
## Kitchen.QualTA -1.815e-01  1.830e-02  -9.920  < 2e-16 ***
## Central.AirY    1.434e-01  1.532e-02   9.358  < 2e-16 ***
## Total.Baths    -4.054e-02  8.835e-03  -4.589 4.79e-06 ***
## Fireplaces      5.068e-02  5.833e-03   8.690  < 2e-16 ***
## Garage.Cars     3.996e-02  5.947e-03   6.718 2.52e-11 ***
## Wood.Deck.SF    8.904e-05  2.560e-05   3.478 0.000519 ***
## Beds.Area       1.960e-05  6.549e-06   2.993 0.002800 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1277 on 1657 degrees of freedom
## Multiple R-squared:  0.8842, Adjusted R-squared:  0.8831
## F-statistic: 790.7 on 16 and 1657 DF,  p-value: < 2.2e-16
```

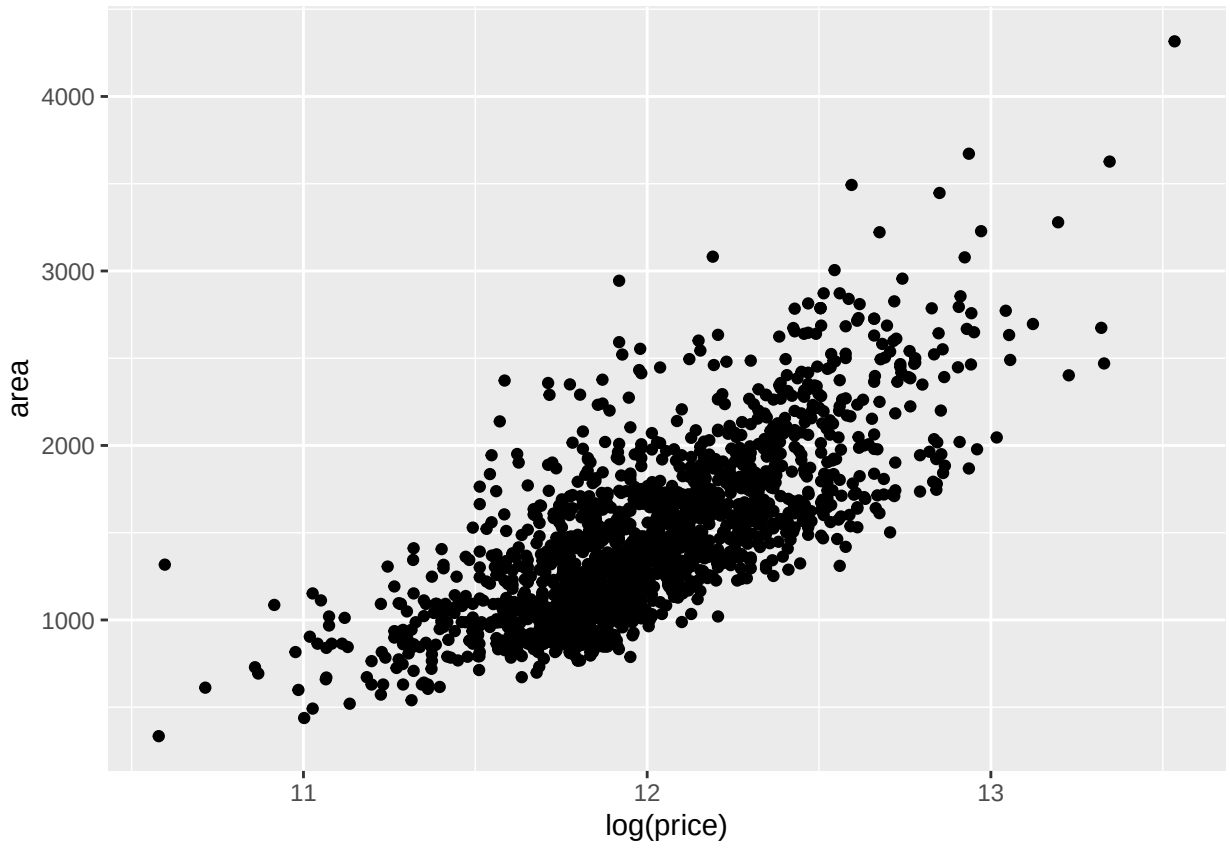
Section 3.2 Transformation

I transformed price, area and lot area, because all three of these distributions were skewed. Additionally, the variables had a stronger linear association when transformed.

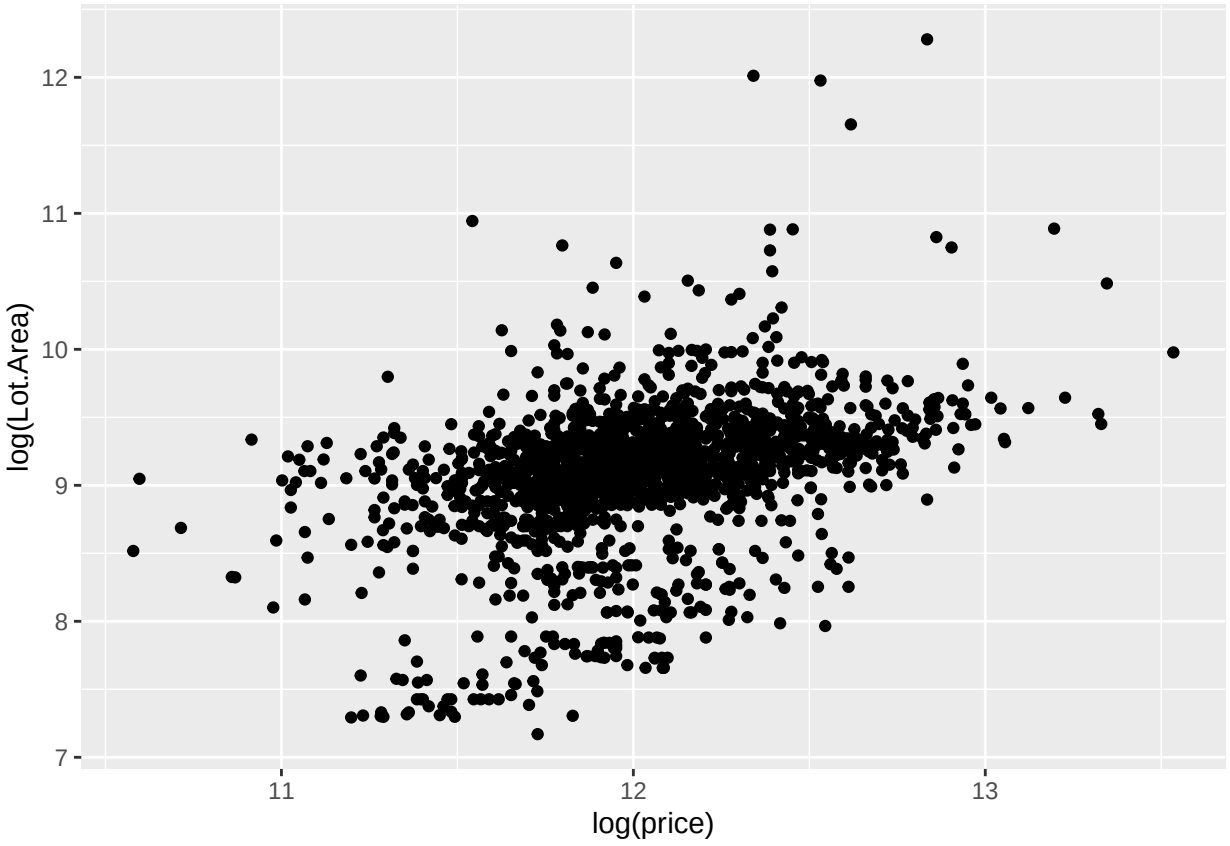
```
ggplot(ames_combo, aes(x=log(price), y=log(area)))+geom_point()
```



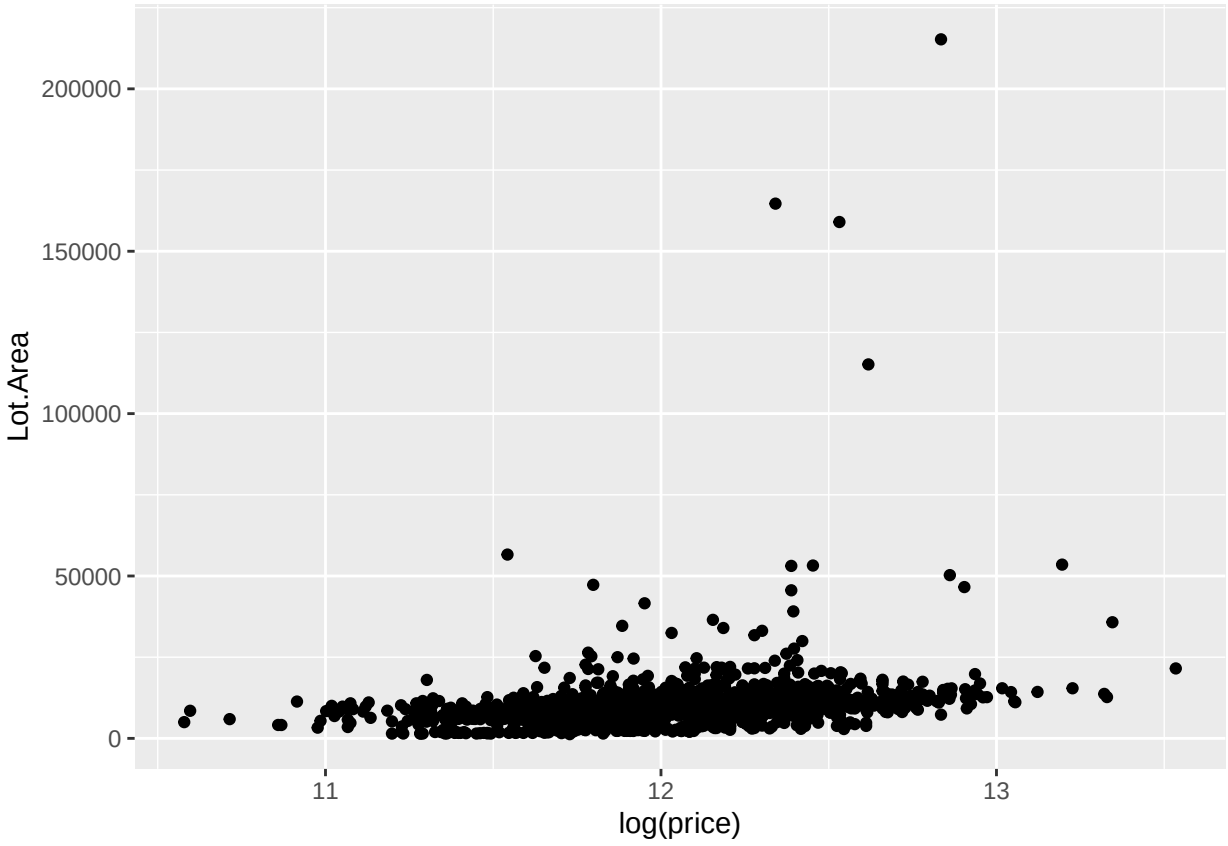

```
ggplot(ames_combo, aes(x=log(price), y=area))+geom_point()
```




```
ggplot(ames_combo, aes(x=log(price), y=log(Lot.Area)))+geom_point()
```




```
ggplot(ames_combo, aes(x=log(price), y=Lot.Area))+geom_point()
```



Section 3.3 Variable Interaction

One of the capstone quizzes suggested that number of bedrooms has a positive association with the price of the house, except that all other things being held equal, larger houses with fewer bedrooms sold for more than larger houses with more bedrooms, so this appeared to be an important variable interaction. I created my interaction variable as described above, and found that it was included by both BIC and AIC model selection.

Section 3.4 Variable Selection

I used backward elimination BIC to select my variables for the final method. I used BIC because I think it is good to create a parsimonious model to avoid overfitting.

I started with a full model that included all the variables from our training model, the variables that we identified as potentially significant during the test of our training model, and our interaction variable.

```
combo_full=lm(log(price)~Year.Built+Year.Remod.Add+log(area)+Bedroom.AbvGr+log(Lot.Area)+Overall.Qual+K
summary(combo_full)
```

```
##
## Call:
## lm(formula = log(price) ~ Year.Built + Year.Remod.Add + log(area) +
##     Bedroom.AbvGr + log(Lot.Area) + Overall.Qual + Kitchen.Qual +
##     Central.Air + Total.Baths + Fireplaces + Garage.Cars + Wood.Deck.SF +
```

```
##     Open.Porch.SF + Beds.Area, data = ames_combo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85012 -0.07458  0.00422  0.08115  0.44404
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.748e-01  5.261e-01   0.522 0.601521
## Year.Built      2.267e-03  1.675e-04  13.541  < 2e-16 ***
## Year.Remod.Add  1.290e-03  2.166e-04   5.956 3.15e-09 ***
## log(area)       4.293e-01  2.087e-02  20.567  < 2e-16 ***
## Bedroom.AbvGr  -2.306e-02  5.391e-03  -4.278 2.00e-05 ***
## log(Lot.Area)   1.256e-01  6.689e-03  18.782  < 2e-16 ***
## Overall.Qual    8.119e-02  3.999e-03  20.302  < 2e-16 ***
## Kitchen.QualFa -2.084e-01  2.916e-02  -7.146 1.33e-12 ***
## Kitchen.QualGd -1.345e-01  1.639e-02  -8.209 4.44e-16 ***
## Kitchen.QualPo -8.480e-02  1.297e-01  -0.654 0.513295
## Kitchen.QualTA -1.803e-01  1.828e-02  -9.864  < 2e-16 ***
## Central.AirY    1.452e-01  1.531e-02   9.482  < 2e-16 ***
## Total.Baths    -4.104e-02  8.824e-03  -4.651 3.56e-06 ***
## Fireplaces      5.077e-02  5.823e-03   8.718  < 2e-16 ***
## Garage.Cars     4.094e-02  5.951e-03   6.879 8.52e-12 ***
## Wood.Deck.SF    9.520e-05  2.568e-05   3.707 0.000217 ***
## Open.Porch.SF   1.351e-04  5.407e-05   2.499 0.012566 *
## Beds.Area       1.909e-05  6.542e-06   2.919 0.003563 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1275 on 1656 degrees of freedom
## Multiple R-squared:  0.8846, Adjusted R-squared:  0.8834
## F-statistic: 746.9 on 17 and 1656 DF,  p-value: < 2.2e-16
```

It would appear that the inclusion of the additional variables identified during the test phase improved our model, as the adjusted r squared is now 0.8834, and all included variables are statistically significant. Next I carried out AIC model selection.

```
combo_AIC=stepAIC(combo_full, direction = "backward", k = 2, trace = TRUE)
```

```
## Start:  AIC=-6877.59
## log(price) ~ Year.Built + Year.Remod.Add + log(area) + Bedroom.AbvGr +
##     log(Lot.Area) + Overall.Qual + Kitchen.Qual + Central.Air +
##     Total.Baths + Fireplaces + Garage.Cars + Wood.Deck.SF + Open.Porch.SF +
##     Beds.Area
##
##                  Df Sum of Sq    RSS     AIC
## <none>                        26.923 -6877.6
## - Open.Porch.SF   1    0.1015 27.025 -6873.3
## - Beds.Area       1    0.1385 27.062 -6871.0
## - Wood.Deck.SF    1    0.2234 27.147 -6865.8
## - Bedroom.AbvGr   1    0.2975 27.221 -6861.2
## - Total.Baths     1    0.3518 27.275 -6857.9
## - Year.Remod.Add  1    0.5767 27.500 -6844.1
```

```
## - Garage.Cars     1    0.7693 27.693 -6832.4
## - Fireplaces      1    1.2357 28.159 -6804.5
## - Central.Air     1    1.4616 28.385 -6791.1
## - Kitchen.Qual    4    1.6177 28.541 -6787.9
## - Year.Built      1    2.9809 29.904 -6703.8
## - log(Lot.Area)   1    5.7353 32.659 -6556.3
## - Overall.Qual    1    6.7013 33.625 -6507.5
## - log(area)       1    6.8771 33.800 -6498.8
```

```
summary(combo_AIC)
```

```
##
## Call:
## lm(formula = log(price) ~ Year.Built + Year.Remod.Add + log(area) +
##     Bedroom.AbvGr + log(Lot.Area) + Overall.Qual + Kitchen.Qual +
##     Central.Air + Total.Baths + Fireplaces + Garage.Cars + Wood.Deck.SF +
##     Open.Porch.SF + Beds.Area, data = ames_combo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85012 -0.07458  0.00422  0.08115  0.44404
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.748e-01  5.261e-01   0.522 0.601521
## Year.Built      2.267e-03  1.675e-04  13.541  < 2e-16 ***
## Year.Remod.Add  1.290e-03  2.166e-04   5.956 3.15e-09 ***
## log(area)       4.293e-01  2.087e-02  20.567  < 2e-16 ***
## Bedroom.AbvGr  -2.306e-02  5.391e-03  -4.278 2.00e-05 ***
## log(Lot.Area)   1.256e-01  6.689e-03  18.782  < 2e-16 ***
## Overall.Qual    8.119e-02  3.999e-03  20.302  < 2e-16 ***
## Kitchen.QualFa -2.084e-01  2.916e-02  -7.146 1.33e-12 ***
## Kitchen.QualGd -1.345e-01  1.639e-02  -8.209 4.44e-16 ***
## Kitchen.QualPo -8.480e-02  1.297e-01  -0.654 0.513295
## Kitchen.QualTA -1.803e-01  1.828e-02  -9.864  < 2e-16 ***
## Central.AirY    1.452e-01  1.531e-02   9.482  < 2e-16 ***
## Total.Baths    -4.104e-02  8.824e-03  -4.651 3.56e-06 ***
## Fireplaces      5.077e-02  5.823e-03   8.718  < 2e-16 ***
## Garage.Cars     4.094e-02  5.951e-03   6.879 8.52e-12 ***
## Wood.Deck.SF    9.520e-05  2.568e-05   3.707 0.000217 ***
## Open.Porch.SF   1.351e-04  5.407e-05   2.499 0.012566 *
## Beds.Area       1.909e-05  6.542e-06   2.919 0.003563 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1275 on 1656 degrees of freedom
## Multiple R-squared:  0.8846, Adjusted R-squared:  0.8834
## F-statistic: 746.9 on 17 and 1656 DF,  p-value: < 2.2e-16
```

AIC did not eliminate any predictors. Next I tried BIC.

```
combo_BIC=stepAIC(combo_full, direction = "backward", k = log(1674), trace = TRUE)
```

```
## Start:  AIC=-6779.98
## log(price) ~ Year.Built + Year.Remod.Add + log(area) + Bedroom.AbvGr +
##     log(Lot.Area) + Overall.Qual + Kitchen.Qual + Central.Air +
##     Total.Baths + Fireplaces + Garage.Cars + Wood.Deck.SF + Open.Porch.SF +
##     Beds.Area
##
##                   Df Sum of Sq    RSS     AIC
## - Open.Porch.SF    1    0.1015 27.025 -6781.1
## <none>                         26.923 -6780.0
## - Beds.Area        1    0.1385 27.062 -6778.8
## - Wood.Deck.SF     1    0.2234 27.147 -6773.6
## - Bedroom.AbvGr    1    0.2975 27.221 -6769.0
## - Total.Baths      1    0.3518 27.275 -6765.7
## - Year.Remod.Add   1    0.5767 27.500 -6751.9
## - Garage.Cars      1    0.7693 27.693 -6740.2
## - Fireplaces       1    1.2357 28.159 -6712.3
## - Kitchen.Qual     4    1.6177 28.541 -6712.0
## - Central.Air      1    1.4616 28.385 -6698.9
## - Year.Built       1    2.9809 29.904 -6611.6
## - log(Lot.Area)    1    5.7353 32.659 -6464.1
## - Overall.Qual     1    6.7013 33.625 -6415.3
## - log(area)        1    6.8771 33.800 -6406.6
##
## Step:  AIC=-6781.1
## log(price) ~ Year.Built + Year.Remod.Add + log(area) + Bedroom.AbvGr +
##     log(Lot.Area) + Overall.Qual + Kitchen.Qual + Central.Air +
##     Total.Baths + Fireplaces + Garage.Cars + Wood.Deck.SF + Beds.Area
##
##                   Df Sum of Sq    RSS     AIC
## <none>                         27.025 -6781.1
## - Beds.Area        1    0.1461 27.171 -6779.5
## - Wood.Deck.SF     1    0.1972 27.222 -6776.3
## - Bedroom.AbvGr    1    0.3184 27.343 -6768.9
## - Total.Baths      1    0.3434 27.368 -6767.4
## - Year.Remod.Add   1    0.5930 27.618 -6752.2
## - Garage.Cars      1    0.7361 27.761 -6743.5
## - Fireplaces       1    1.2315 28.256 -6713.9
## - Kitchen.Qual     4    1.6451 28.670 -6711.9
## - Central.Air      1    1.4283 28.453 -6702.3
## - Year.Built       1    3.0567 30.081 -6609.1
## - log(Lot.Area)    1    5.8788 32.904 -6459.0
## - Overall.Qual     1    6.8369 33.862 -6411.0
## - log(area)        1    7.1523 34.177 -6395.5
```

```
summary(combo_BIC)
```

```
##
## Call:
## lm(formula = log(price) ~ Year.Built + Year.Remod.Add + log(area) +
##     Bedroom.AbvGr + log(Lot.Area) + Overall.Qual + Kitchen.Qual +
```

```
##     Central.Air + Total.Baths + Fireplaces + Garage.Cars + Wood.Deck.SF +
##     Beds.Area, data = ames_combo)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84729 -0.07381  0.00535  0.08162  0.43314
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.467e-01  5.245e-01   0.280 0.779701
## Year.Built      2.292e-03  1.674e-04  13.690  < 2e-16 ***
## Year.Remod.Add  1.308e-03  2.169e-04   6.030 2.02e-09 ***
## log(area)       4.351e-01  2.078e-02  20.941  < 2e-16 ***
## Bedroom.AbvGr  -2.382e-02  5.391e-03  -4.419 1.06e-05 ***
## log(Lot.Area)   1.268e-01  6.681e-03  18.986  < 2e-16 ***
## Overall.Qual    8.184e-02  3.997e-03  20.474  < 2e-16 ***
## Kitchen.QualFa -2.112e-01  2.919e-02  -7.234 7.13e-13 ***
## Kitchen.QualGd -1.348e-01  1.641e-02  -8.215 4.24e-16 ***
## Kitchen.QualPo -9.118e-02  1.299e-01  -0.702 0.482720
## Kitchen.QualTA -1.815e-01  1.830e-02  -9.920  < 2e-16 ***
## Central.AirY    1.434e-01  1.532e-02   9.358  < 2e-16 ***
## Total.Baths    -4.054e-02  8.835e-03  -4.589 4.79e-06 ***
## Fireplaces      5.068e-02  5.833e-03   8.690  < 2e-16 ***
## Garage.Cars     3.996e-02  5.947e-03   6.718 2.52e-11 ***
## Wood.Deck.SF    8.904e-05  2.560e-05   3.478 0.000519 ***
## Beds.Area       1.960e-05  6.549e-06   2.993 0.002800 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1277 on 1657 degrees of freedom
## Multiple R-squared:  0.8842, Adjusted R-squared:  0.8831
## F-statistic: 790.7 on 16 and 1657 DF,  p-value: < 2.2e-16
```

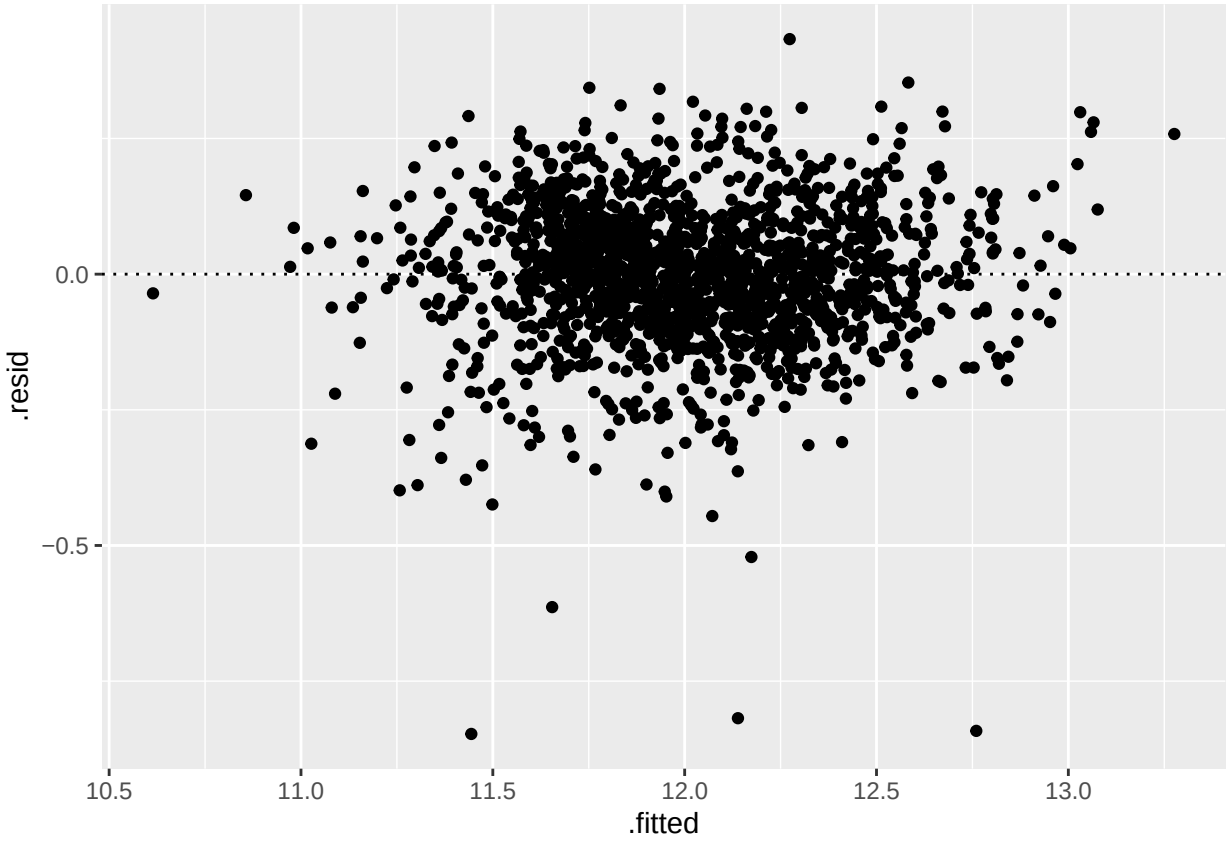
BIC eliminated porch square footage and left us with an adjusted r-squared of 0.8831, which is still very close to the adjusted r-squared of our full model, so I dropped porch square footage to help avoid overfitting. It turned out to be good that we added in the interaction variable, deck square footage, bathrooms, fireplaces and number of cars the garage accommodates.

Part 4 Final Model Assessment

Section 4.1 Final Model Residual

Here is a residual plot for the combo_interact model.

```
ggplot(combo_interact, aes(x=.fitted, y=.resid))+geom_point()+geom_hline(yintercept = 0, linetype= "dot
```



We see the same 3 low outliers as before, but there is no clear form to the residual plot so the model appears to be a good fit for the data. We see approximately equal variance for all values of predicted price.

Section 4.2 Final Model RMSE

Let's check out the RMSE for our final model.

```
predict_combo<-exp(predict(combo_interact, ames_combo))
resid_combo<-ames_combo$price - predict_combo
rmse_combo<-sqrt(mean(resid_combo^2))
rmse_combo
```

```
## [1] 24094.15
```

Our model is working great, with an average prediction error of $24,094.15.

Section 4.3 Final Model Evaluation

One strength of the model is that it appears to function just about as well for low priced as high priced homes. Also, it has a very high adjusted r squared, indicating that we were able to account for over 88% of the variability in price using the identified predictors. A weakness of the model is that it has a lot of

predictors, so it is fairly complicated and could potentially be overfitted to the data that we used. Also, there are still a few significant outliers in the residuals, so occasionally the model meaninfully overestimates the price of a home.

Section 4.4 Final Model Validation

We will now testing the final model on a separate, validation data set to determine how the model will perform in real-life practice.

```
load("ames_validation.Rdata")
```

In order to use our model on the validation data we will need to create a total baths variable in that data set and clean up any NA's in the garage cars variable.

```
ames_validation<-ames_validation%>%mutate(Total.Baths=ames_validation$Full.Bath+.5*ames_validation$Half
ames_validation$Garage.Cars[which(is.na(ames_validation$Garage.Cars))]<-0
```

We will also need to create our interaction variable.

```
ames_validation<-ames_validation%>%mutate(BedsC=ames_validation$Bedroom.AbvGr - mean(ames_validation$Be
```

Next we'll multiply the centered variables for bedrooms and area.

```
ames_validation<-ames_validation%>%mutate(Beds.Area=ames_validation$BedsC * ames_validation$areaC)
```

Now we will see how our final model does when applied to the out-of-sample validation data. We will assess this first using average prediction error.

```
predict_val<-exp(predict(combo_interact, ames_validation))
resid_val<-ames_validation$price - predict_val
rmse_val<-sqrt(mean(resid_val^2))
rmse_val
```

```
## [1] 21821.52
```

The average prediction error for the validation data is just $21,821.52, which is lower than the RMSE we achieved with the training, testing and combined data. Perhaps this data set doesn't have homes as unusual as our three outliers from the training data.

Next we'll check to see what percent of the 95% predictive confidence intervals contain the true price of the house in the validation data set.

```
predict.full <- exp(predict(combo_interact, ames_validation, interval = "prediction"))
coverage.prob.full <- mean(ames_validation$price > predict.full[,"lwr"] &
                           ames_validation$price < predict.full[,"upr"])
coverage.prob.full
```

```
## [1] 0.9567497
```

95.67% of the confidence intervals calculated using my final model capture the true price of the home in the validation data set, so my model is slightly out-performing the expressed uncertainty. It's better to be out-performing than under-performing, so this is good.

Part 5 Conclusion

Overall, we have developed a model that estimates the price of a house with an average error of about $24,000, and is able to account for about 88% of the variability in price. 95% Confidence intervals for home price generated by our model capture the actual price of out-of-sample homes just over 95% of the time. We therefore recommend this model to the investment firm for use in selecting properties to purchase. It might be off on individual properties from time to time, but in the long run it is highly reliable.