Predict Home Sales Price

March 18, 2018

PSTAT126

Professor Xiyue Liao

Zian He

2.Introduction

There are 12 variables. We choose the “Sale Price” as the response and the other 11 variables to be potential predictors to predict the home sale price. “Air conditioning”, “Pool”, “Quality” and “Adjacent to highway” are four categorical variables. There are 521 observations in this given data. Our research question is to determine which variables greatly contribute to home sales prices and are there any interactions between these variables? We want to predict a new home sale prices with certain useful characteristics.

3.Question of interest

Predict residential home sales prices in a midwestern city as a function of various characteristics of the home and surrounding property.

4.Regression methods

Select variables: best subset, adjusted R square criteria

Check “LINE” conditions of a model: “Residual vs Fit” plot, normal Q-Q plot

Find interactions: Add1(), F test between full model and reduced

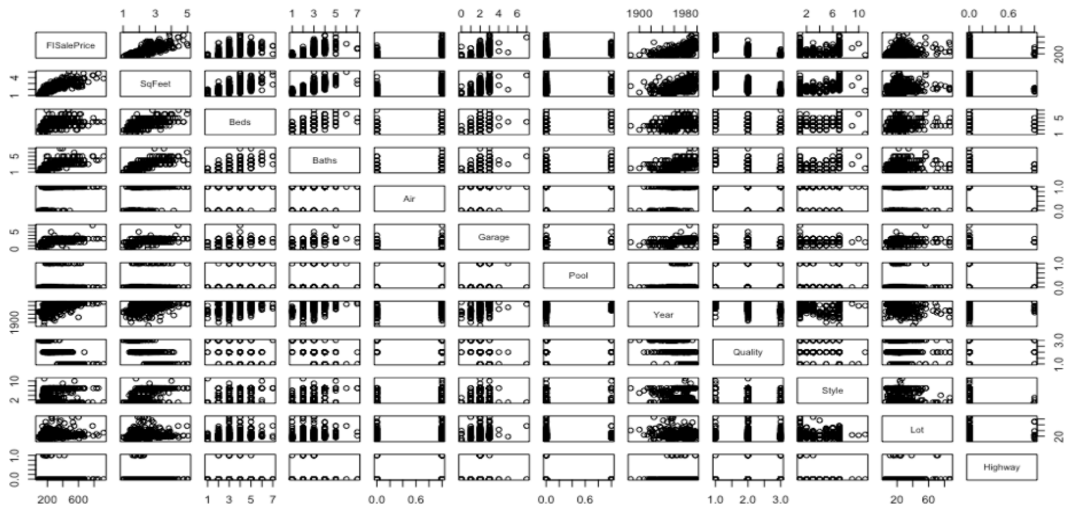
Transform response: Boxcox()

Detect influential data: Cook.distance

5.Regression Analysis, Results and Interpretation

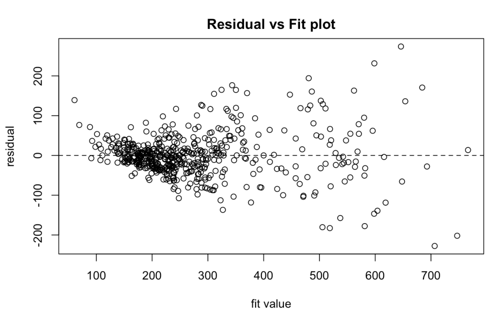
Firstly, to get an overview of how each variable relates to home sale price we use the pairs () function to draw a scatterplot matrix.

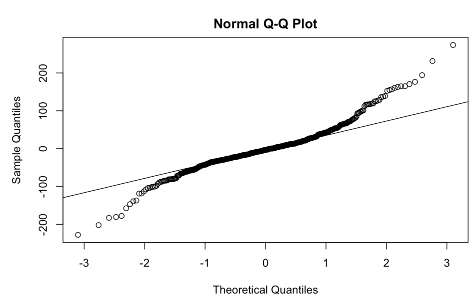
From the chart below, we find that the variable “lot” and “style” have no strong linear relationship with home sale price, so we need to pay attention to whether keep these two variables or not in the next step.



Then we use the best subset method to select useful predictor variables. First we add all the 11 variables into the model. It offers eight possible best models. Based on the adjusted R square criteria, we choose the best model with the largest adjusted R^2 value which is 0.7872. The best model indicates that “SqFeet, quality, year, highway, style, beds, garage and lot” are the most useful variables to predict the home sale prices.

After selecting useful variables, we want to check how well our basic model satisfies the “Line” condition. So we draw the “Residual vs Fit” plot, Normal Q-Q plot. We find that the model doesn’t meet the “Line” condition well.

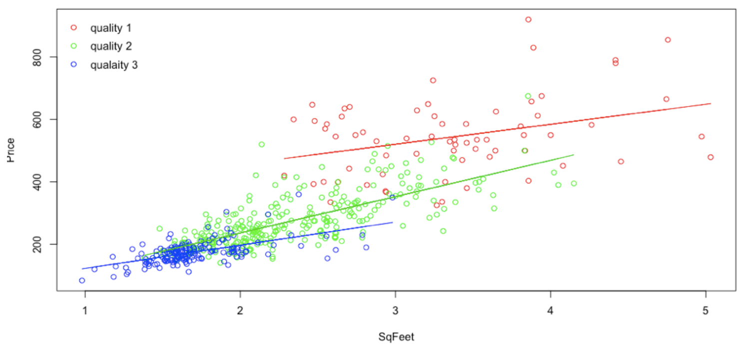




So we need to consider adding interaction terms into the model. First we consider only interactions between continuous variables. There are 6 continuous variables. Rather than testing all the possible interactions between any two of them, we decide to optimize the procedure that we first only consider the interactions between the most significant variable and other variables. Summarizing our original model with 8 selected variables, we find that “SqFeet” has the smallest p value, which indicates that it is the most significant continuous variable in predicting the home sale price. Via the add1 () function, we add one possible interaction term between “SqFeet” and the left 5 continuous variables at a time into the model and check the p value. We choose “SqFeet*beds” term because it has lowest p value. Then we make a summary table of the model with “SqFeet*beds” added. We find that the variable “garage” is no longer significant in the model. Therefore, we remove the variable “garage” out of my model. Here, we take advantage of the idea from the stepwise method to delete no more useful variable.

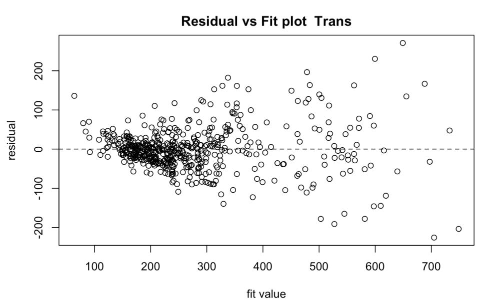
Repeating the same procedure, we choose the second significant variable “bed” and add its interaction with the left 4 continuous variables into the model. Finally we also add “year*beds” interaction term to the model. To avoid a model to be overly complicated, here the most two significant interactions are enough.

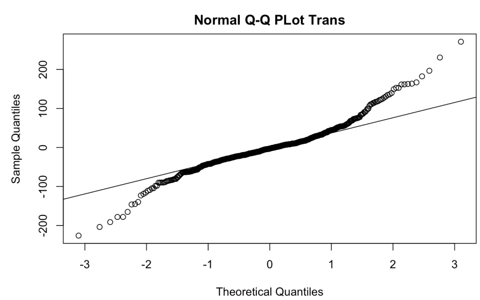
After considered the interactions between the continuous variables, we also need to consider the interactions between continuous variables and categorical variables. There are two selected categorical variables in the original model, “quality” and “highway”. We find that most of the “highway” values are 0, so highway might not be a significant variable to affect other variables in predicting the home sale price. We draw the plot of relations between “Price” and “Sqfeet” under different qualities, ranging from 1 to 3. That is we want to analysis whether the quality will have additional effect on “Sqfeet” in predicting the price.



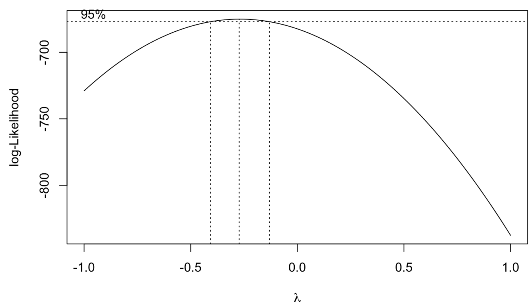
From the plot we can see there are interactions between each different quality. Then we build a full model adding the interactions “SqFeet*quality”. Then we conduct F test to compare the reduced model without “SqFeet*quality” and the full model using the anova table. We find the p value 0.0001075 is very small, so we can reject the reduced model that has no interactions. So we decide to add the interaction “SqFeet*quality”.

After selected all the interactions, we draw the “Residual vs Fit” plot and Normal Q-Q plot again to check whether our model satisfies the “LINE” conditions.

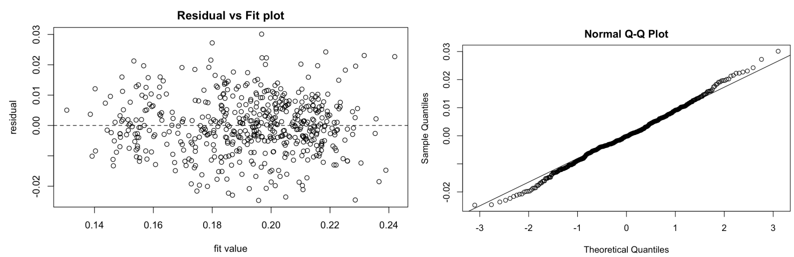




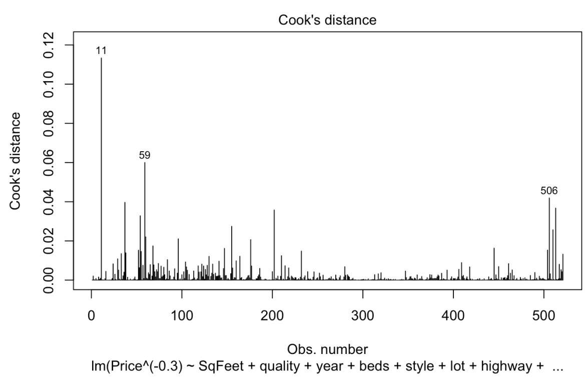
From the plots, we find that Residual vs Fit plot has “fanning” effect, which means this model has “non-constant error variances”. Because of the “fanning” effect, we have to transform the response home sale price. Using the boxcox method, we choose -0.3 as lambda’s value in the model from the log-likelihood plot.



Then we draw Residual vs Fit plot and Normal Q-Q plot again to check the line conditions. This time the model satisfies the “LINE” conditions very well.



Lastly, we also need to determine whether there are influential points in our data. From the plot, the largest Cook.Distance value is smaller than 0.12. Since none of the data points has a Cook's Distance value greater than 0.5, we can conclude that there is no data point in our model to be potentially influential.



Go back to our Research Question: To predict the sale price of a home in a Midwestern city that is built in 1980, with size of 3000 square feet, 4 bedrooms and 40 square feet size lot, in medium quality and the style 7, and not adjacent to highway.

To better predict a new individual home sale price, we need to build a prediction interval. We use the predict function to get a point estimate and a 95% prediction interval.

```
new = data.frame(year = 1980, SqFeet = 3, beds = 4, quality = factor(2), style = 7, lot = 40, highway = 0)
pi = predict(model.cox, new, interval = 'prediction', level = 0.95)
pi
```
```

        fit       lwr       upr
 1 0.1686517 0.1502424 0.1870611
```

To explain this interval, we need to transform these prediction limits back into the original units. 95% P.I. is (267.12941956228303, 554.6575875914154)

So we can be 95% confident that the residential home sales prices in a Midwestern city for a new house with characteristics that built in 1980, 3000 square feet, 4 bedrooms, medium quality, style = 7, 40 square feet size lot, absence of adjacency to highway will between 267.13 and 554.66.

6. Conclusion

To build an effective model to predict the home sales prices, we select the most useful predictor variables, and then find interactions between those variables. After that, to satisfy the model with the "LINE" conditions, we need to transform the response. We get a fine model that satisfies the "LINE" conditions well. We also need to detect whether there are influential points in the data set. Finally we use the model to solve the research question to predict a new home sale price and get a prediction interval. To conclude, "SqFeet, quality, year, highway, style, beds, and lot" variables are useful predictors of home sale price. "SqFeet*beds", "year*beds" and "SqFeet*quality" are useful interactions terms in this model.

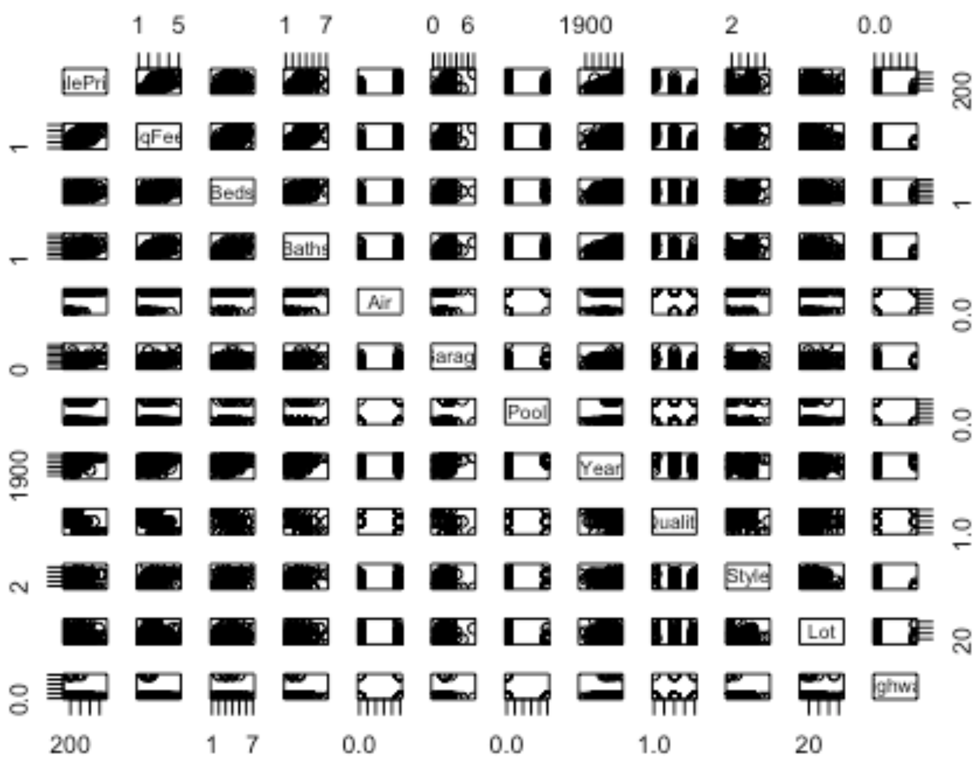
Our finalized model is:

Price^(0.3)~SqFeet+quality+year+beds+style+lot+highway+SqFeet*beds+year*beds+SqFeet*quality

7.Appendix

Final Project

```
dat=read.table('/Users/xuzhenyi/Desktop/PSTAT\ 126\ Regression\ Analysis/Proj
ect\ Data\ Sets-20180308/realestate.txt',header = T)
Price=dat$SalePrice
SqFeet=dat$SqFeet
beds=dat$Beds
baths=dat$Baths
air=dat$Air
garage=dat$Garage
pool=dat$Pool
year=dat$Year
quality=factor(dat$Quality)
style=dat$Style
lot=dat$Lot
highway=dat$Highway
pairs(dat)
```



from the pairs plot, we can find lot and style are not related to home sale price

CHOOSE PREDICTORS

subset regression procedure

```
library(leaps)
mod.subset=regsubsets(cbind(SqFeet,quality,year,garage,highway,air,pool,beds,
baths,style,lot),Price)
summary.mod=summary(mod.subset)
summary.mod$which
```

```
##   (Intercept) SqFeet quality  year garage highway   air  pool  beds baths
## 1        TRUE   TRUE   FALSE FALSE  FALSE   FALSE FALSE FALSE FALSE FALSE
## 2        TRUE   TRUE    TRUE FALSE  FALSE   FALSE FALSE FALSE FALSE FALSE
## 3        TRUE   TRUE    TRUE FALSE  FALSE   FALSE FALSE FALSE FALSE FALSE
## 4        TRUE   TRUE    TRUE  TRUE  FALSE   FALSE FALSE FALSE FALSE FALSE
## 5        TRUE   TRUE    TRUE  TRUE  FALSE   FALSE FALSE FALSE FALSE FALSE
## 6        TRUE   TRUE    TRUE  TRUE   TRUE   FALSE FALSE FALSE FALSE FALSE
## 7        TRUE   TRUE    TRUE  TRUE   TRUE   FALSE FALSE FALSE  TRUE FALSE
## 8        TRUE   TRUE    TRUE  TRUE   TRUE    TRUE FALSE FALSE  TRUE FALSE
##   style   lot
## 1 FALSE FALSE
## 2 FALSE FALSE
## 3  TRUE FALSE
## 4  TRUE FALSE
## 5  TRUE  TRUE
## 6  TRUE  TRUE
## 7  TRUE  TRUE
## 8  TRUE  TRUE
```

```
summary.mod$adjr2
```

```
## [1] 0.6763211 0.7404796 0.7629433 0.7735508 0.7826106 0.7844199 0.7861583
## [8] 0.7872412
```

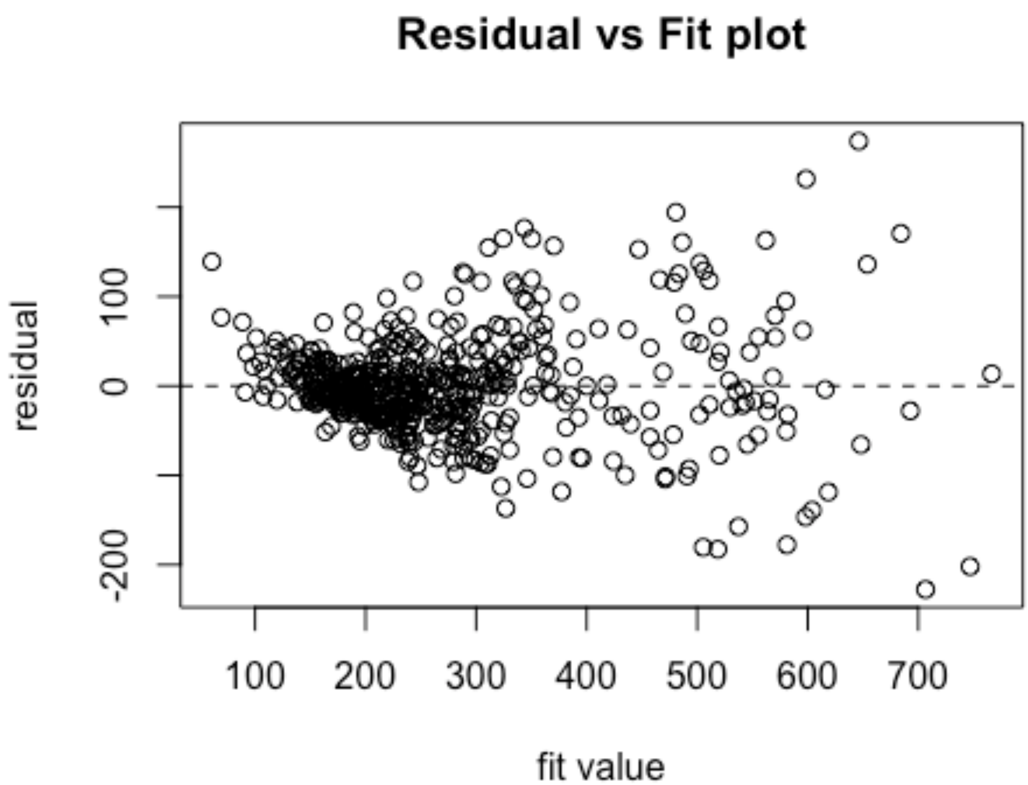
Based on adjusted R^2 critiera

Therefore the best model is price~SqFeet+quality+year+garage+highway+beds+style+lot

LINE Conditions test (without transformation and interaction)

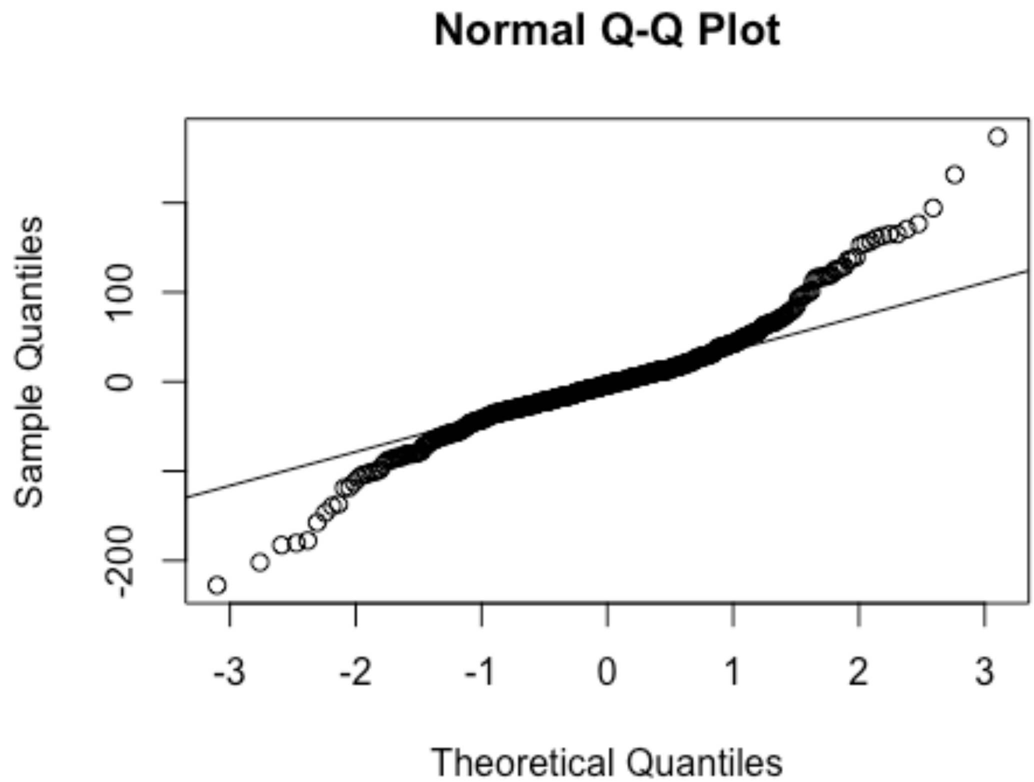
from the residual vs Fit plot, we can find out that it has “fanning effect”,which means this model has non-constant error varriances

```
model=lm(Price ~SqFeet+quality+year+garage+highway+beds+style+lot,data=dat)
price=fitted(model)
error=Price-price
plot(price,error,xlab='fit value',ylab='residual',main='Residual vs Fit plot')
abline(h=0,lty=2)
```



Normal Q-Q Plot (without transformation and interaction)

```
qqnorm(error)
qqline(error)
```



consider the interaction term (continous+continous varibles)

```
mod0=lm(Price~SqFeet+quality+year+garage+highway+beds+style+lot)
add1(mod0,~.+SqFeet*year+SqFeet*garage+SqFeet*beds+SqFeet*style+SqFeet*lot,te
st='F')
```

```
## Single term additions
##
## Model:
## Price ~ SqFeet + quality + year + garage + highway + beds + style +
##     lot
##                Df Sum of Sq     RSS    AIC F value    Pr(>F)
## <none>                      1703742 4236.2
## SqFeet:year    1     31105 1672637 4228.6  9.4843 0.0021841 **
## SqFeet:garage  1       527 1703215 4238.1  0.1578 0.6913487
## SqFeet:beds    1     66286 1637456 4217.6 20.6453 6.908e-06 ***
## SqFeet:style   1     39382 1664361 4226.1 12.0675 0.0005568 ***
## SqFeet:lot     1       293 1703450 4238.1  0.0876 0.7673484
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

choose SqFeet*beds

```
model.1=update(mod0,~.+SqFeet*beds)
summary(model.1)
```

```
##
## Call:
## lm(formula = Price ~ SqFeet + quality + year + garage + highway +
##     beds + style + lot + SqFeet:beds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -205.952  -27.855   -3.374   23.067  265.581
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2585.3792   374.8166  -6.898 1.57e-11 ***
## SqFeet        168.7848    14.5601  11.592  < 2e-16 ***
## quality2     -134.3541    10.2084 -13.161  < 2e-16 ***
## quality3     -137.1633    13.5255 -10.141  < 2e-16 ***
## year            1.3135     0.1892   6.943 1.17e-11 ***
## garage          6.7438     4.8989   1.377 0.169244
## highway       -37.9596    17.4969  -2.170 0.030505 *
## beds           30.4206     7.7983   3.901 0.000109 ***
## style          -6.4448     1.3049  -4.939 1.07e-06 ***
## lot             1.3352     0.2272   5.878 7.52e-09 ***
## SqFeet:beds   -13.8395     3.0459  -4.544 6.91e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56.66 on 510 degrees of freedom
## Multiple R-squared:  0.8337, Adjusted R-squared:  0.8305
## F-statistic: 255.7 on 10 and 510 DF,  p-value: < 2.2e-16
```

based on summary table, when beds*SqFeet was added in the model, garage variable is no longer significant. We remove the garage variable from model

```
model.1=update(model.1,~.-garage)

add1(model.1,~.+year*beds+year*style+year*lot,test='F')
```

```
## Single term additions
##
## Model:
## Price ~ SqFeet + quality + year + highway + beds + style + lot +
##     SqFeet:beds
##            Df Sum of Sq     RSS    AIC F value  Pr(>F)
## <none>                  1643541 4217.5
## year:beds   1   16531.5 1627009 4214.2  5.1819 0.02324 *
```

```
## year:style  1   13073.4 1630467 4215.3  4.0893 0.04368 *
## year:lot    1    1444.6 1642096 4219.0  0.4487 0.50327
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We add year*beds interaction term into model based on p value

```
model.2=lm(Price~SqFeet+quality+year+beds+style+lot+highway+SqFeet*beds+year*
beds)
summary(model.2)

##
## Call:
## lm(formula = Price ~ SqFeet + quality + year + beds + style +
##     lot + highway + SqFeet * beds + year * beds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -211.173  -28.780   -2.693   24.178  258.667
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -171.80482 1163.75763  -0.148   0.8827
## SqFeet       187.98757   15.62214  12.033  < 2e-16 ***
## quality2    -133.77931   10.13186 -13.204  < 2e-16 ***
## quality3    -137.60989   13.37720 -10.287  < 2e-16 ***
## year           0.07027    0.59903   0.117   0.9067
## beds        -733.75784  336.45392  -2.181   0.0297 *
## style         -6.78953    1.29893  -5.227 2.52e-07 ***
## lot            1.36024    0.22515   6.041 2.95e-09 ***
## highway      -35.02249   17.49188  -2.002   0.0458 *
## SqFeet:beds  -18.33460    3.46583  -5.290 1.82e-07 ***
## year:beds      0.39407    0.17311   2.276   0.0232 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56.48 on 510 degrees of freedom
## Multiple R-squared:  0.8348, Adjusted R-squared:  0.8315
## F-statistic: 257.7 on 10 and 510 DF,  p-value: < 2.2e-16
```

check summary table again, it works fine!

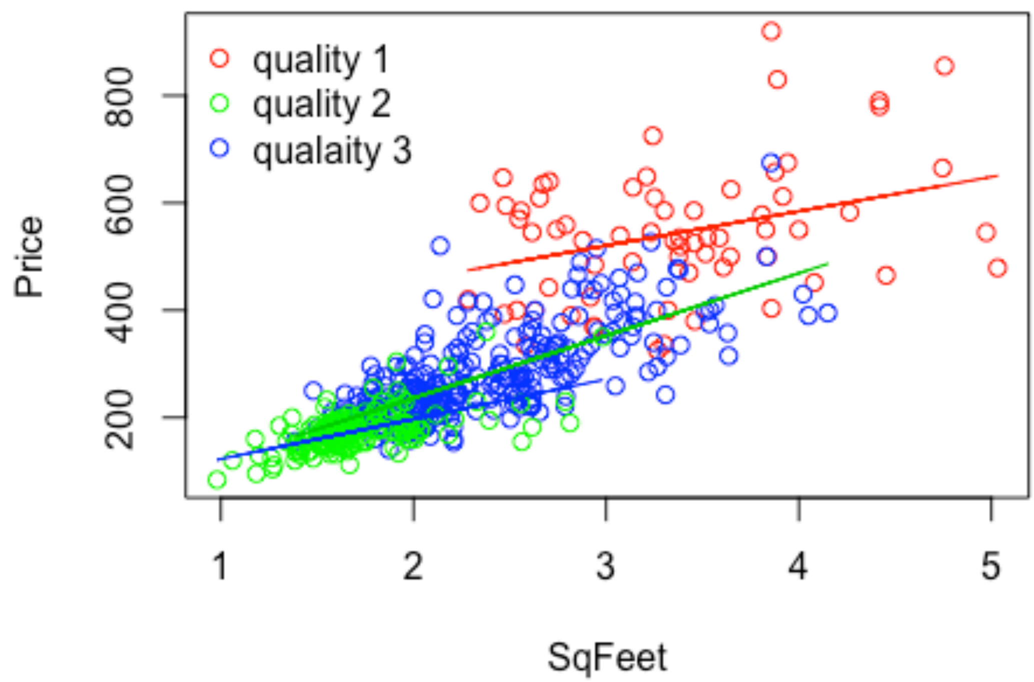
consider interaction term (categorical+continous)

quality VS SqFeet

```
yhat=fitted(model)
y_1=yhat[quality=='1']
plot(SqFeet,Price,type='n')
fitt = lm(Price ~ SqFeet+quality)
summary(fitt)

##
## Call:
## lm(formula = Price ~ SqFeet + quality)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -230.59  -31.00   -7.57   24.31  327.25
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  209.212     20.315   10.30   <2e-16 ***
## SqFeet        99.440      5.579   17.82   <2e-16 ***
## quality2    -166.260     10.396  -15.99   <2e-16 ***
## quality3    -203.563     13.070  -15.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.53 on 517 degrees of freedom
## Multiple R-squared:  0.7881, Adjusted R-squared:  0.7869
## F-statistic:   641 on 3 and 517 DF,  p-value: < 2.2e-16

points(SqFeet[quality=='1'],Price[quality=='1'],col='red')
points(SqFeet[quality=='2'],Price[quality=='2'],col='blue')
points(SqFeet[quality=='3'],Price[quality=='3'],col='green')
legend('topleft',bty='n',col = c('red','green', 'blue'), c('quality 1', 'qual
ity 2', 'qualaity 3'), pch = c(1, 1, 1))
for(i in 1:3) {
 data.quality=dat[quality==i, ]
 x.q=data.quality$SqFeet
 y.q=data.quality$SalePrice
 y.fit=lm(y.q ~ x.q)
 response.q=fitted(y.fit)
 lines(x.q,response.q,col=i+1)
}
```



anova

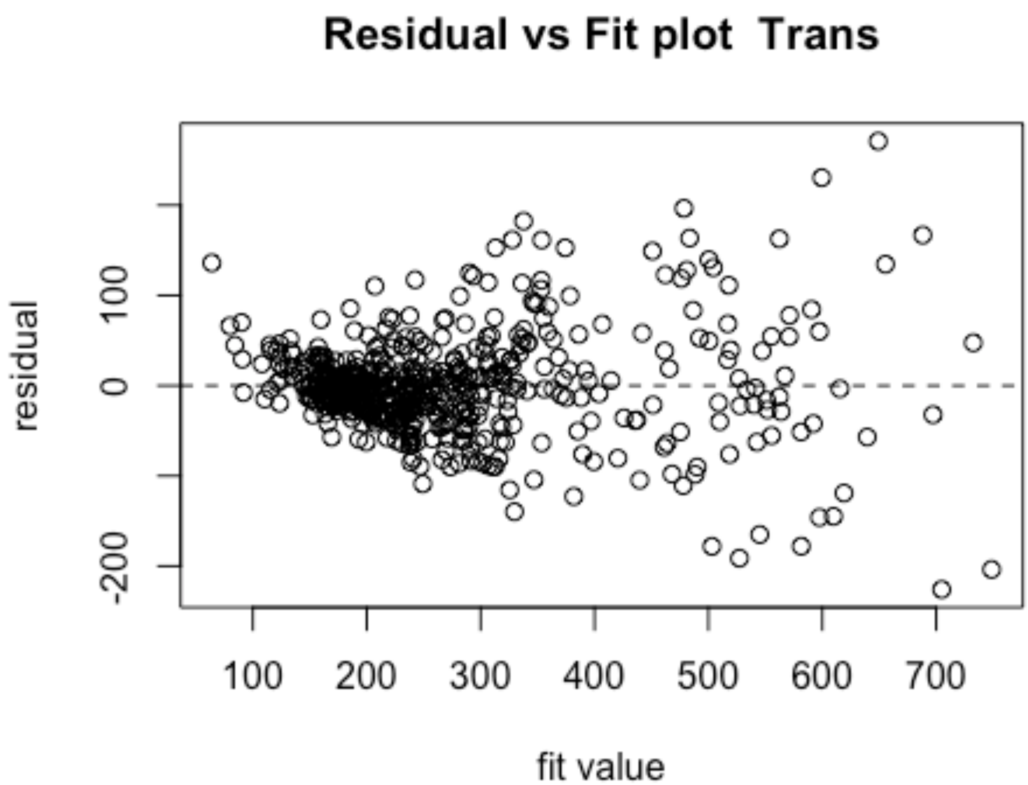
```
reduced.mod=model.2
full.mod=lm(Price~SqFeet+quality+year+beds+style+lot+highway+SqFeet*beds+year
*beds+SqFeet*quality)
anova(reduced.mod,full.mod)

## Analysis of Variance Table
##
## Model 1: Price ~ SqFeet + quality + year + beds + style + lot + highway +
##     SqFeet * beds + year * beds
## Model 2: Price ~ SqFeet + quality + year + beds + style + lot + highway +
##     SqFeet * beds + year * beds + SqFeet * quality
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    510 1627009
## 2    508 1569517  2     57492 9.3041 0.0001075 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

LINE Conditions test (With interaction terms)

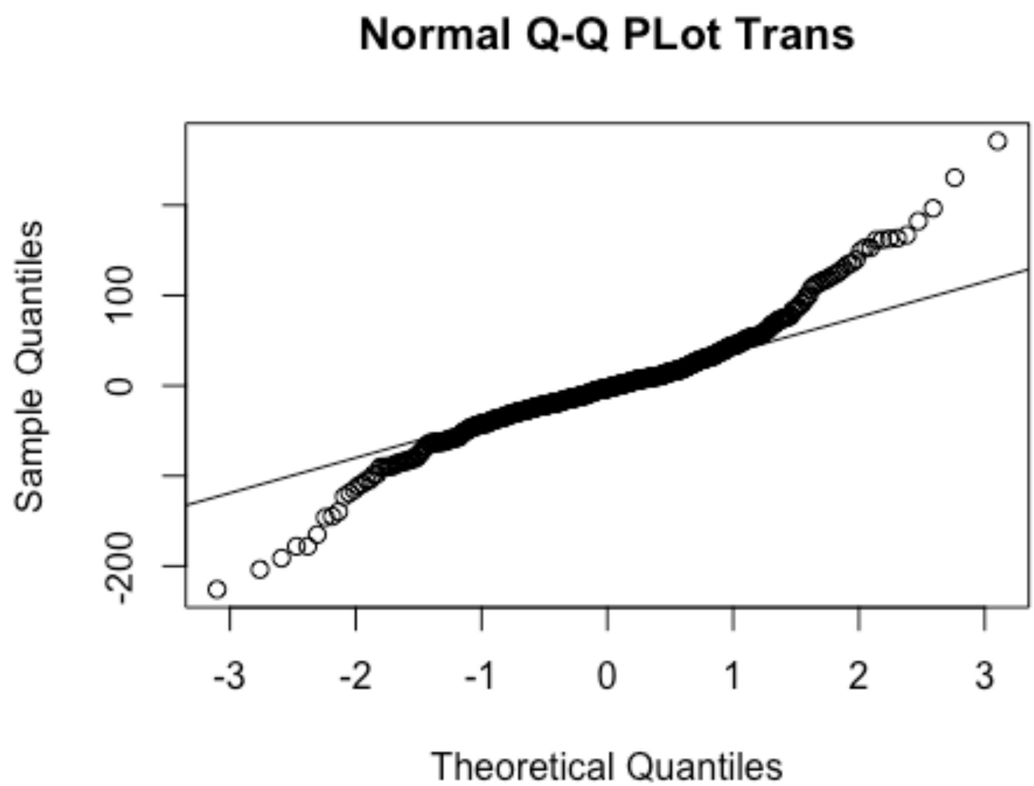
Residual vs Fit Plot

```
model.subset=lm(price~SqFeet+quality+year+beds+style+lot+highway+SqFeet*beds+
year*beds+SqFeet*quality)
price.subset=fitted(model.subset)
error.subset=Price-price.subset
plot(price.subset,error.subset,xlab='fit value',ylab='residual',main='Residua
l vs Fit plot  Trans')
abline(h=0,lty=2)
```



Normal Q-Q Plot

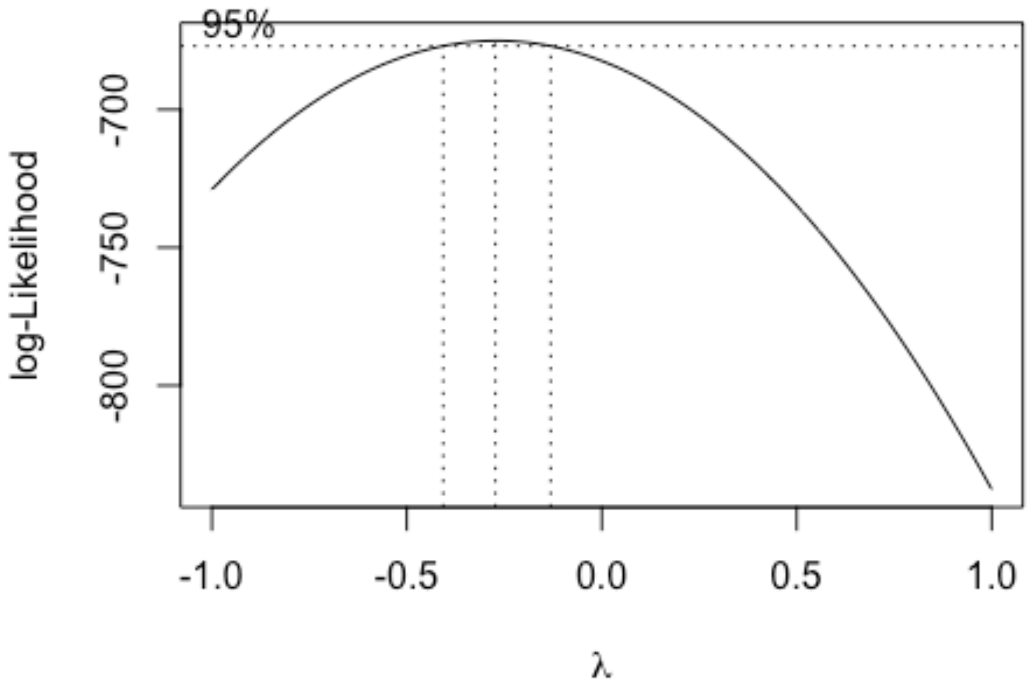
```
e=error.subset
qqnorm(e,main='Normal Q-Q PLot Trans')
qqline(e)
```



based on Q-Q plot and Residual vs fit charts, the model still doesn't fullfill LINE conditions. Since the residual vs fit plot has fanning effect, we decide to transform response (home sale price) using Box-cox transform

Transformation using Box-cox transformation

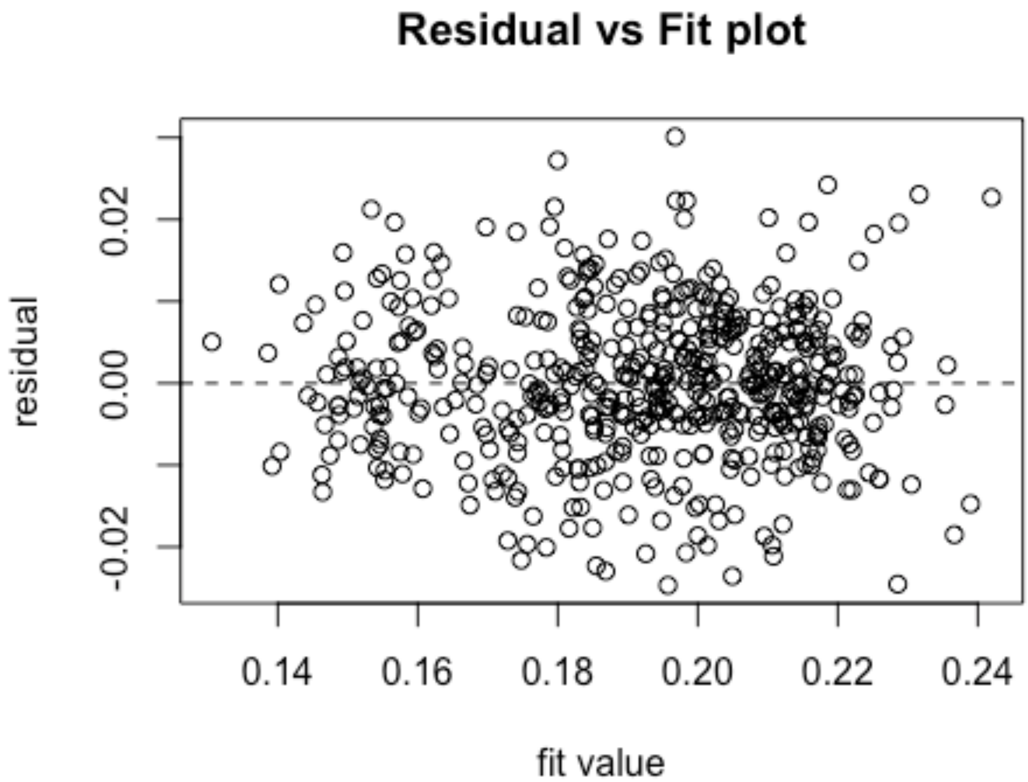
```
library(MASS)
boxcox.trans=boxcox(Price~SqFeet+quality+year+beds+style+lot+highway+SqFeet*b
eds+year*beds+SqFeet*quality,lambda=seq(-1,1))
```



##
based on the log-likelihood plot, we find that -0.3 is the best value of lambda for this data set.

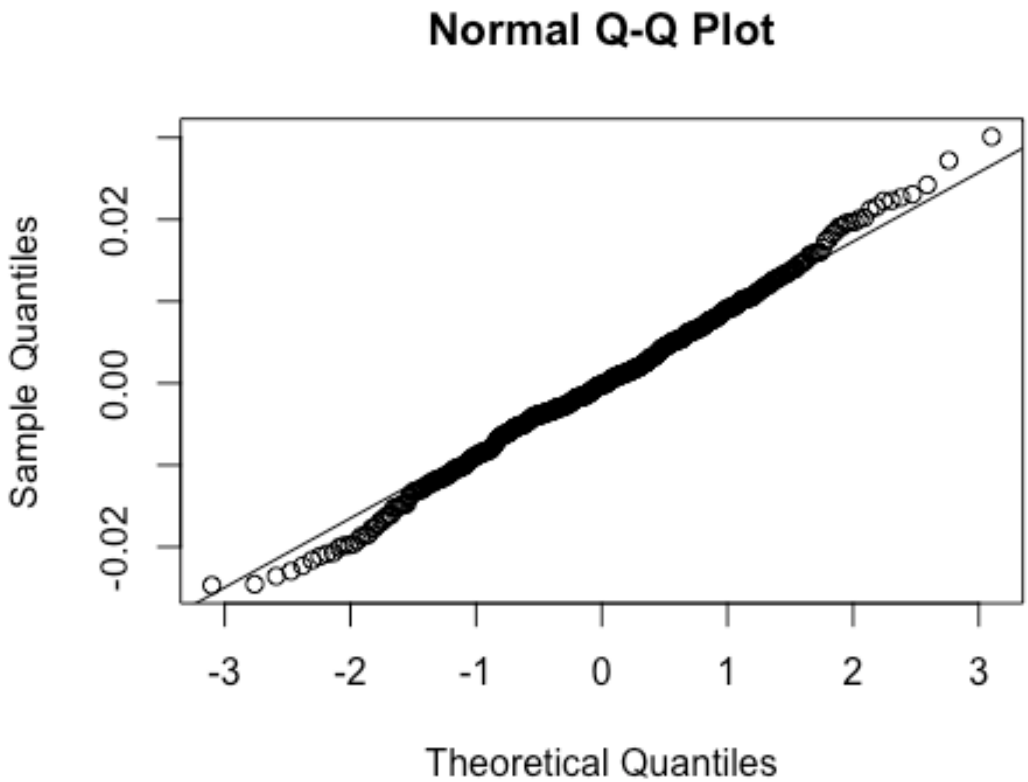
Residual vs Fit Plot (with boxcox)

```
model.cox=lm(Price^(-0.3)~SqFeet+quality+year+beds+style+lot+highway+SqFeet*b
eds+year*beds+SqFeet*quality,data=dat)
price.cox=fitted(model.cox)
error.cox=Price^(-0.3)-price.cox
plot(price.cox,error.cox,xlab='fit value',ylab='residual',main='Residual vs F
it plot')
abline(h=0,lty=2)
```



Q-Q Plot

```
error.cox=Price^(-0.3)-price.cox
qqnorm(error.cox)
qqline(error.cox)
```




```
new = data.frame(year = 1980, SqFeet = 3, beds = 4, quality = factor(2), style
 = 7, lot = 40, highway = 0)
pi = predict(model.cox, new, interval = 'prediction', level = 0.95)
pi
```

```
##         fit       lwr       upr
## 1 0.1686517 0.1502424 0.1870611
```

```
model.cox=lm(Price^(-0.3)~SqFeet+quality+year+beds+style+lot+highway+SqFeet*b
eds+year*beds+SqFeet*quality,data=dat)
cooks.distance(model.cox)
```

```
##            1            2            3            4            5
## 3.231531e-05 2.053885e-03 2.560864e-04 3.916777e-05 7.983176e-04
##            6            7            8            9           10
## 7.040696e-05 4.519917e-08 1.562266e-03 3.477008e-04 9.315738e-04
##           11           12           13           14           15
## 1.133353e-01 7.520999e-05 1.571254e-04 7.075329e-06 7.132778e-04
##           16           17           18           19           20
## 4.455091e-03 5.958288e-07 3.890250e-05 8.512148e-06 5.916722e-05
##           21           22           23           24           25
## 1.113543e-05 1.036516e-04 8.555638e-04 8.272125e-03 8.220237e-06
##           26           27           28           29           30
## 2.983976e-03 3.182245e-04 4.100489e-07 1.081275e-02 5.111704e-03
```

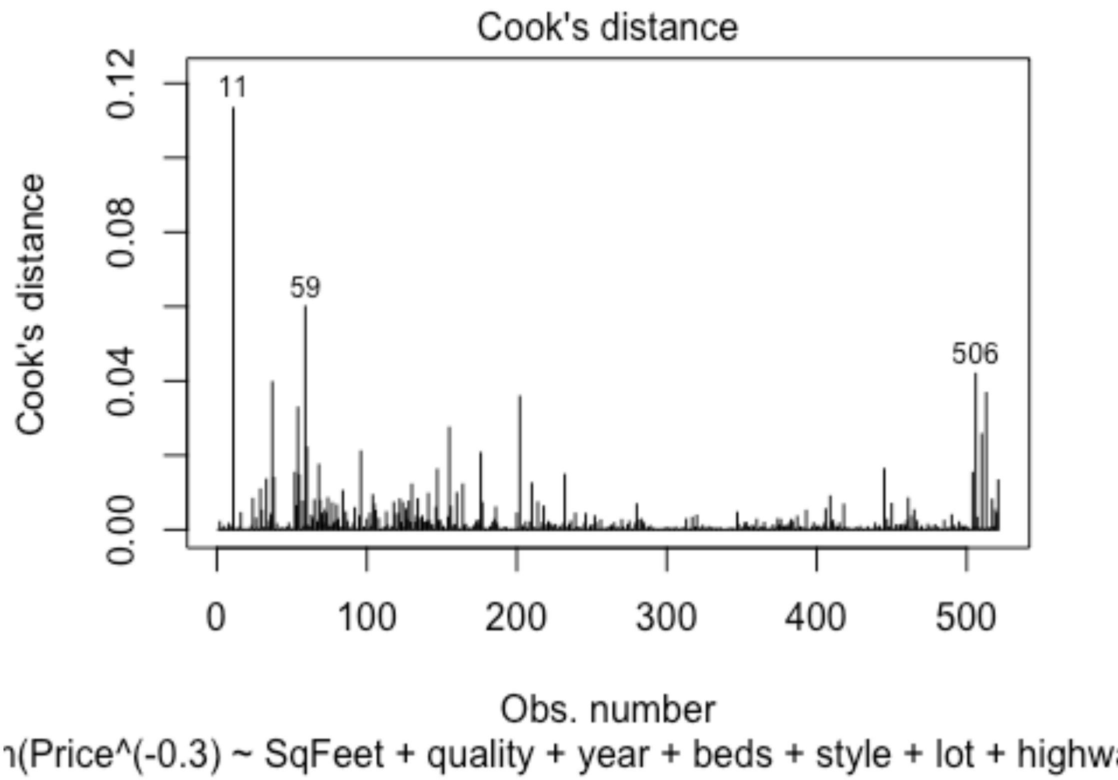
```
##            31           32           33           34           35 
## 4.811122e-04 1.538870e-04 1.348542e-02 3.919459e-04 2.188409e-03 
##            36           37           38           39           40 
## 4.064461e-03 3.965489e-02 1.391211e-02 2.700384e-05 1.464390e-03 
##            41           42           43           44           45 
## 2.499888e-07 1.093901e-05 5.401465e-04 2.208814e-05 1.673190e-04 
##            46           47           48           49           50 
## 9.152733e-05 5.713220e-04 1.590606e-03 2.482621e-04 2.243395e-05 
##            51           52           53           54           55 
## 1.889769e-04 1.526980e-02 6.321778e-03 3.281730e-02 1.462382e-02 
##            56           57           58           59           60 
## 6.688803e-05 7.566207e-03 7.618995e-04 5.998827e-02 2.205989e-02 
##            61           62           63           64           65 
## 9.891500e-04 6.329464e-04 3.754294e-03 2.935365e-03 7.903444e-03 
##            66           67           68           69           70 
## 1.206303e-04 2.083828e-03 1.744747e-02 7.777011e-03 4.290500e-03 
##            71           72           73           74           75 
## 1.611859e-03 5.290044e-03 4.287962e-03 8.470593e-03 8.119146e-04 
##            76           77           78           79           80 
## 7.487801e-04 7.121817e-03 1.618529e-03 2.177908e-03 6.560745e-03 
##            81           82           83           84           85 
## 2.725896e-03 1.396928e-04 6.307420e-04 1.041872e-02 3.815365e-04 
##            86           87           88           89           90 
## 4.649742e-03 2.132903e-03 6.619213e-05 3.197772e-04 1.769346e-04 
##            91           92           93           94           95 
## 1.549799e-03 5.787420e-03 1.579900e-04 5.038269e-07 3.561795e-03 
##            96           97           98           99          100 
## 2.105574e-02 5.000565e-05 6.065302e-04 8.356203e-10 2.715175e-03 
##          101          102          103          104          105 
## 2.057474e-04 4.261544e-03 8.490337e-04 9.250518e-03 6.728130e-03 
##          106          107          108          109          110 
## 5.002467e-03 8.187913e-06 2.879840e-03 4.216036e-06 6.570106e-05 
##          111          112          113          114          115 
## 2.065191e-04 4.623359e-04 4.671880e-03 1.085318e-03 8.334784e-05 
##          116          117          118          119          120 
## 1.490503e-04 7.049167e-04 7.237107e-03 4.070142e-03 1.288268e-04 
##          121          122          123          124          125 
## 4.206049e-03 8.132971e-03 1.780848e-03 7.197091e-03 4.664603e-04 
##          126          127          128          129          130 
## 5.449444e-03 2.663495e-03 7.531359e-03 1.878185e-03 1.211727e-02 
##          131          132          133          134          135 
## 8.484962e-05 1.859107e-03 3.515913e-03 8.165969e-03 2.901974e-03 
##          136          137          138          139          140 
## 8.065847e-07 3.536287e-03 1.892818e-03 1.897498e-03 2.189786e-03 
##          141          142          143          144          145 
## 9.660011e-03 2.253555e-03 1.082318e-03 1.137908e-03 2.674421e-05 
##          146          147          148          149          150 
## 5.890006e-03 1.615649e-02 2.300822e-03 2.319371e-03 2.328493e-04 
##          151          152          153          154          155 
## 8.132519e-04 2.347177e-04 7.765307e-08 3.301424e-03 2.742826e-02
```

```
##          156          157          158          159          160 
## 6.149696e-03 3.745239e-04 1.273482e-03 1.145204e-03 9.831009e-03 
##          161          162          163          164          165 
## 9.462227e-05 6.518510e-05 1.554109e-04 1.214595e-02 1.019497e-03 
##          166          167          168          169          170 
## 1.330405e-03 1.066488e-05 2.886718e-04 1.117293e-04 1.333061e-04 
##          171          172          173          174          175 
## 6.410417e-04 1.256947e-03 2.128298e-03 2.395559e-03 1.107082e-03 
##          176          177          178          179          180 
## 2.064147e-02 7.190868e-03 2.131715e-04 7.758929e-05 9.559123e-05 
##          181          182          183          184          185 
## 2.894971e-05 6.777066e-04 7.168855e-04 1.825175e-03 2.682627e-03 
##          186          187          188          189          190 
## 5.966334e-03 1.547905e-03 2.284427e-04 1.030129e-05 6.643229e-05 
##          191          192          193          194          195 
## 1.630309e-04 5.198375e-04 3.668530e-04 3.106403e-05 1.589339e-05 
##          196          197          198          199          200 
## 2.952404e-05 6.611919e-05 6.002566e-08 4.718922e-05 4.324118e-03 
##          201          202          203          204          205 
## 2.434453e-04 3.578431e-02 4.341141e-05 9.514771e-04 4.519627e-04 
##          206          207          208          209          210 
## 2.008250e-03 1.991267e-04 1.301132e-04 1.950369e-03 1.244278e-02 
##          211          212          213          214          215 
## 1.756988e-05 5.070984e-04 4.015839e-05 7.312726e-03 9.833038e-05 
##          216          217          218          219          220 
## 3.307991e-04 1.725376e-03 6.216069e-03 1.110775e-03 8.156200e-06 
##          221          222          223          224          225 
## 2.041988e-03 1.570908e-03 9.598939e-04 4.559081e-04 1.294835e-03 
##          226          227          228          229          230 
## 1.181002e-03 1.102386e-04 2.012199e-04 7.519636e-04 1.308836e-04 
##          231          232          233          234          235 
## 1.520863e-03 1.479253e-02 6.646500e-04 3.486023e-05 1.318947e-03 
##          236          237          238          239          240 
## 2.206115e-03 2.135560e-06 3.623850e-05 4.298943e-03 5.784942e-04 
##          241          242          243          244          245 
## 2.407706e-04 5.982949e-04 7.340749e-05 6.237429e-05 1.534438e-03 
##          246          247          248          249          250 
## 4.206845e-03 1.210914e-06 1.962371e-04 8.544940e-04 1.192570e-03 
##          251          252          253          254          255 
## 5.856534e-05 3.583189e-03 1.697388e-03 6.947033e-06 9.740691e-05 
##          256          257          258          259          260 
## 2.519763e-03 1.564867e-04 1.333668e-05 2.447893e-04 1.747598e-04 
##          261          262          263          264          265 
## 2.572967e-04 7.072896e-04 1.006750e-03 5.624477e-04 1.710763e-03 
##          266          267          268          269          270 
## 7.374298e-05 6.381571e-04 1.870948e-04 7.000039e-05 2.576532e-03 
##          271          272          273          274          275 
## 6.791343e-04 1.696814e-05 3.913623e-05 1.214245e-03 2.186466e-03 
##          276          277          278          279          280 
## 7.806297e-06 4.478255e-05 1.500088e-04 1.587565e-03 6.815447e-03
```

```
##          281          282          283          284          285
## 2.472431e-03 5.492687e-06 2.740320e-03 2.004714e-03 1.515914e-03
##          286          287          288          289          290
## 1.635775e-04 1.640798e-04 5.024809e-04 9.311229e-04 1.186745e-04
##          291          292          293          294          295
## 1.700175e-04 1.804645e-04 2.587616e-06 1.876264e-06 8.723351e-05
##          296          297          298          299          300
## 3.193166e-05 1.574720e-05 1.676049e-04 2.014975e-04 2.794890e-04
##          301          302          303          304          305
## 4.146207e-04 2.233603e-05 2.763427e-04 2.694462e-05 8.126416e-05
##          306          307          308          309          310
## 6.293507e-04 1.646119e-06 8.340590e-06 1.184149e-04 1.590820e-04
##          311          312          313          314          315
## 3.092741e-04 1.991641e-04 2.769439e-03 4.630840e-05 2.263286e-05
##          316          317          318          319          320
## 1.292906e-04 3.194979e-03 1.433923e-04 6.138315e-04 3.755950e-03
##          321          322          323          324          325
## 3.719227e-04 1.107944e-04 3.217320e-04 1.014263e-03 3.301316e-05
##          326          327          328          329          330
## 4.543084e-04 1.597551e-05 7.037074e-04 8.523864e-05 3.249216e-04
##          331          332          333          334          335
## 3.198603e-04 2.500035e-05 1.438980e-04 3.212155e-04 2.661813e-05
##          336          337          338          339          340
## 3.305081e-04 3.816989e-06 7.507871e-06 3.171022e-05 1.038902e-06
##          341          342          343          344          345
## 1.428454e-04 6.393727e-05 3.595719e-04 7.010896e-05 2.150834e-04
##          346          347          348          349          350
## 5.445739e-07 4.588167e-03 1.284302e-03 3.800585e-05 1.703235e-04
##          351          352          353          354          355
## 5.468239e-04 1.649278e-03 1.793774e-03 7.229331e-04 6.040933e-04
##          356          357          358          359          360
## 9.596217e-05 1.045095e-03 7.261978e-04 8.499673e-05 2.666066e-03
##          361          362          363          364          365
## 4.387471e-05 3.627591e-04 7.017339e-04 4.254424e-04 1.826238e-03
##          366          367          368          369          370
## 4.345593e-04 3.122491e-05 1.649757e-07 1.728209e-04 2.535604e-04
##          371          372          373          374          375
## 1.195170e-03 2.349068e-05 5.705717e-04 2.787967e-03 1.109888e-03
##          376          377          378          379          380
## 2.616752e-03 7.579284e-04 7.251797e-04 8.330571e-04 8.464385e-04
##          381          382          383          384          385
## 8.293947e-04 1.666105e-03 2.614035e-03 2.078656e-03 2.057863e-05
##          386          387          388          389          390
## 1.974120e-04 3.479008e-03 1.684618e-04 7.118142e-04 4.306966e-04
##          391          392          393          394          395
## 1.708538e-04 2.007416e-04 5.087979e-03 6.218240e-05 2.590338e-05
##          396          397          398          399          400
## 1.047820e-06 6.927536e-04 1.706929e-03 2.650151e-06 7.773635e-04
##          401          402          403          404          405
## 7.255993e-04 7.875122e-04 7.353309e-05 5.863490e-04 1.567227e-03
```

```
##          406          407          408          409          410 
## 5.474051e-03 3.507176e-04 3.377992e-06 8.932995e-03 1.873834e-03 
##          411          412          413          414          415 
## 2.388495e-03 8.395167e-04 1.039221e-06 6.558577e-04 1.646279e-03 
##          416          417          418          419          420 
## 4.020337e-04 3.961944e-05 6.737587e-03 5.592059e-04 1.385027e-05 
##          421          422          423          424          425 
## 4.308643e-04 2.936465e-05 1.153571e-05 1.027094e-04 6.214406e-05 
##          426          427          428          429          430 
## 2.124974e-04 5.133338e-04 7.041742e-05 7.722435e-04 1.585096e-05 
##          431          432          433          434          435 
## 4.385336e-08 1.715244e-04 4.890309e-06 6.557005e-04 2.550014e-05 
##          436          437          438          439          440 
## 5.037081e-05 6.876826e-05 1.458393e-04 1.641893e-03 2.379069e-04 
##          441          442          443          444          445 
## 4.649640e-04 8.619164e-04 8.762675e-05 2.896011e-05 1.629348e-02 
##          446          447          448          449          450 
## 3.404436e-04 2.683658e-03 6.896382e-04 2.764947e-04 6.914856e-03 
##          451          452          453          454          455 
## 1.533129e-04 6.516185e-07 1.179530e-03 1.927994e-05 1.019091e-03 
##          456          457          458          459          460 
## 1.074066e-03 1.356996e-03 2.150503e-04 1.027513e-03 2.267072e-03 
##          461          462          463          464          465 
## 8.386206e-03 1.992958e-04 3.510601e-03 3.065777e-05 5.180944e-03 
##          466          467          468          469          470 
## 3.970652e-04 2.283671e-03 8.056284e-08 2.329476e-04 8.470269e-04 
##          471          472          473          474          475 
## 4.150935e-05 1.562538e-06 1.846875e-05 1.366539e-03 4.655125e-04 
##          476          477          478          479          480 
## 2.143718e-05 7.997019e-04 1.227394e-06 1.179258e-03 2.811809e-04 
##          481          482          483          484          485 
## 4.558417e-04 1.256930e-04 2.787215e-05 1.698892e-05 2.344198e-03 
##          486          487          488          489          490 
## 1.762593e-05 2.371236e-04 9.741962e-05 3.294225e-04 3.874449e-03 
##          491          492          493          494          495 
## 3.059234e-04 1.013375e-03 2.067398e-04 1.644368e-05 1.969050e-03 
##          496          497          498          499          500 
## 8.791032e-04 3.751035e-04 6.673268e-04 8.952117e-04 4.763775e-04 
##          501          502          503          504          505 
## 2.586333e-04 4.376178e-05 2.059207e-05 1.528842e-02 6.260875e-04 
##          506          507          508          509          510 
## 4.189998e-02 3.197768e-03 4.278733e-04 8.972421e-05 2.567476e-02 
##          511          512          513          514          515 
## 2.767390e-04 1.885383e-05 3.672892e-02 3.008139e-04 1.502742e-04 
##          516          517          518          519          520 
## 5.637967e-04 8.059433e-03 7.162526e-04 5.296092e-03 4.275935e-03 
##          521 
## 1.322587e-02

plot(model.cox, which = 4)
```



Since none of the data points has a Cook's Distance value greater than 0.5, thus we could conclude that there is data point in our model to be flagged as potentially influential.