

STOR 455 Homework #5

40 points - Due Friday 3/11 at 5:00pm

Directions: For parts 6 and 9 you may work together, but they should be **submitted individually** by each group member. For parts 7 and 8, you should have only **one submission per group**. There will be separate places on Gradescope to submit the individual vs group work.

Situation: Can we predict the selling price of a house in Ames, Iowa based on recorded features of the house? That is your task for this assignment. Each team will get a dataset with information on forty potential predictors and the selling price (in \$1,000's) for a sample of homes. The data sets for your group are AmesTrain??.csv and AmesTest??.csv (where ?? corresponds to your group number) A separate file identifies the variables in the Ames Housing data and explains some of the coding.

```
library(readr)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

AmesTrain2 <- read.csv("AmesTrain2.csv")
AmesTest2 <- read.csv("AmesTest2.csv")
```

Part 6. Cross-validation:

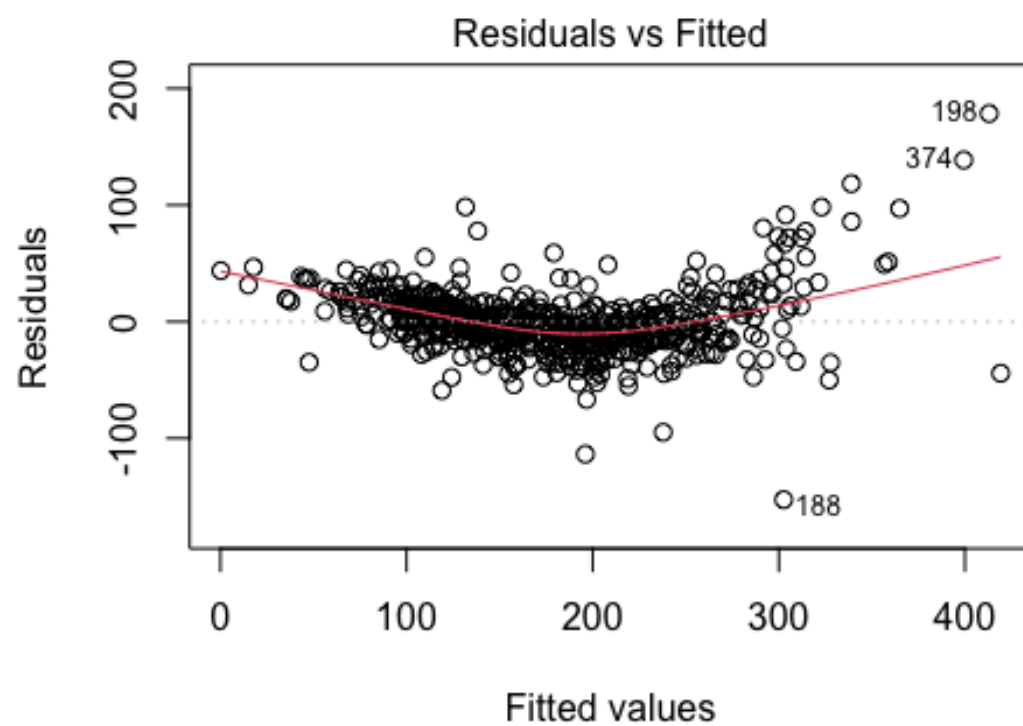
In some situations, a model might fit the peculiarities of a specific sample of data well, but not reflect structure that is really present in the population. A good test for how your model might work on “real” house prices can be simulated by seeing how well your fitted model does at predicting prices that were NOT in your original sample. This is why we reserved an additional 200 cases as a holdout sample in AmesTest??.csv. Import your holdout test data and

- Compute the predicted Price for each of the cases in the holdout test sample, using your model resulting from the initial fit and residual analysis in parts 1 and 2 of Homework #3. This should be done with the same AmesTrain??.csv dataset that you used for homework #3, with your homework #3 group number, and AmesTest?? also using your homework #3 group number.

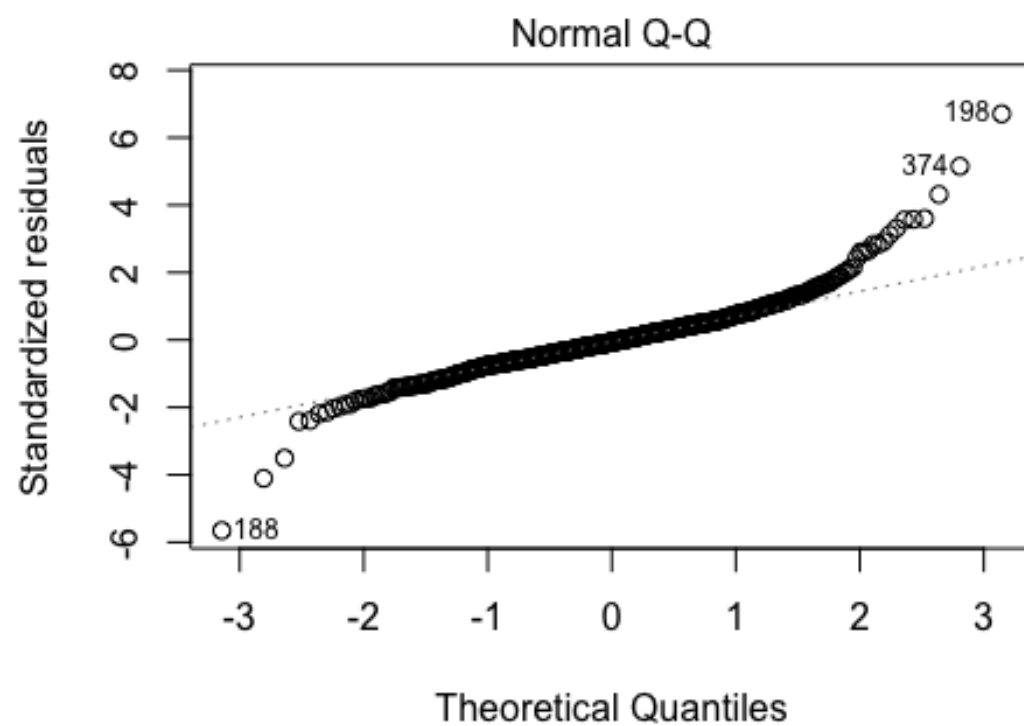
I removed Order from the dataset since it is just a counter. I also removed variables that are components of a total variable in the dataset like BasementFinSF & BasementUnFinSF and FirstSF & SecondSF.

```
Adjusted_TrainData = select(AmesTrain2, !Order & !BasementFinSF &
!BasementUnFinSF & !FirstSF & !SecondSF)
Adjusted_TestData = select(AmesTest2, !Order & !BasementFinSF &
!BasementUnFinSF & !FirstSF & !SecondSF)

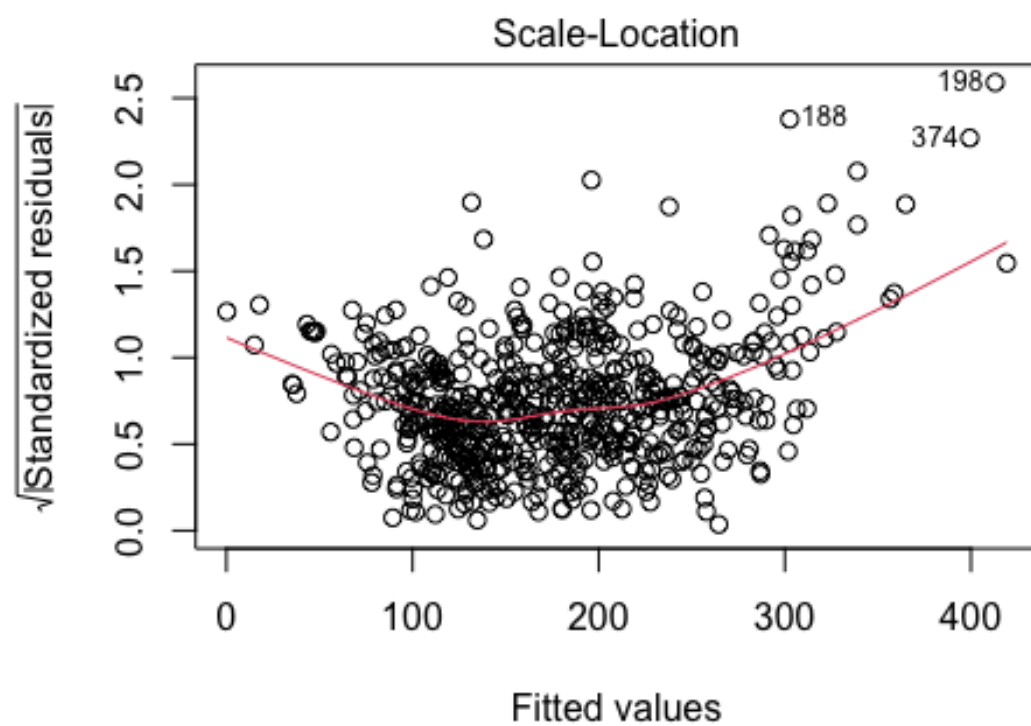
House = lm(Price ~ Quality + GroundSF + BasementSF + YearBuilt +
  GarageSF + LotArea + Condition + BasementFBath + YearRemodel +
  Fireplaces + ScreenPorchSF + LotFrontage + Bedroom + EnclosedPorchSF,
  data = Adjusted_TrainData)
plot(House)
```



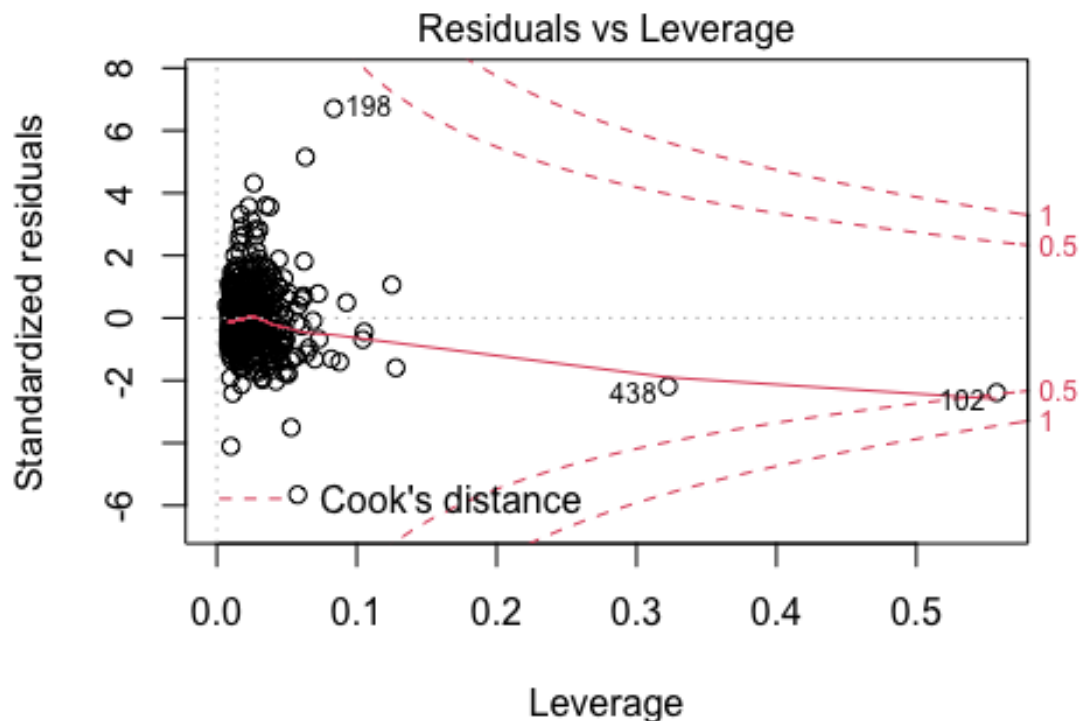
ice ~ Quality + GroundSF + BasementSF + YearBuilt + GarageSF +



ice ~ Quality + GroundSF + BasementSF + YearBuilt + GarageSF +



ice ~ Quality + GroundSF + BasementSF + YearBuilt + GarageSF +



ice ~ Quality + GroundSF + BasementSF + YearBuilt + GarageSF +

```
NewHouse = predict(House, newdata=Adjusted_TestData)
```

- Compute the residuals for the 200 holdout cases.

```
residuals = AmesTest2$Price - NewHouse
```

- Compute the mean and standard deviation of these residuals. Are they close to what you expect from the training model?

The mean of the residuals is 3 which interprets to \$3000 which is low since the house prices are in thousands of dollars. The standard deviation is almost the same to the standard deviation of the training model.

```
mean(residuals)
```

```
## [1] 3.007899
```

```
summary(House)$sigma
```

```
## [1] 27.81253
```

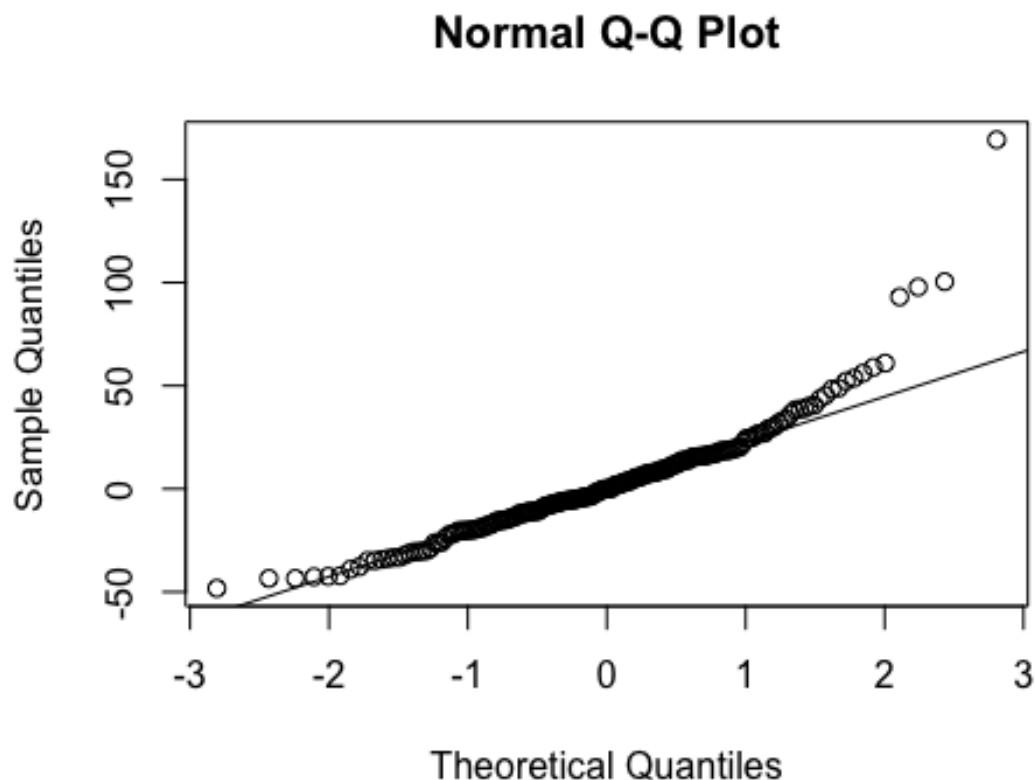
```
sd(residuals)
```

```
## [1] 27.57541
```

- Construct a plot of the residuals to determine if they are normally distributed. Is this plot what you expect to see considering the training model?

Most the data points fit the normally distributed line like the training model and also the right tail skew is similar. There is no longer a left tail skew on the test model in comparison to the training model which has it. Overall, it is a plot I expect since it is not too different.

```
qqnorm(residuals)
qqline(residuals)
```



- Are any holdout cases especially poorly predicted by the training model? If so, identify by the row number(s) in the holdout data.

Since there isn't a "maximum" residual that is considered poorly predicted, I listed the top 10 highest residuals.

```
head(sort(abs(residuals), decreasing=TRUE), 10)
```

```
##          8        137        167        136         17         52        186
99
## 169.23611 100.46801  97.80726  93.05555  60.95624  58.93988  56.11684
53.68590
##        126        153
##  52.19540  48.83043
```

- Compute the correlation between the predicted values and actual prices for the holdout sample. This is known as the cross-validation correlation. We don't expect the training model to do better at predicting values different from those that were used to build it (as reflected in the original R^2), but an effective model shouldn't do a lot worse at predicting the holdout values. Square the cross-validation correlation to get an R^2 value and subtract it from the original multiple R^2 of the training sample. This is known as the shrinkage. We won't have specific rules about how little the shrinkage should be, but give an opinion on whether the shrinkage looks OK to you or too large in your situation. The shrinkage is OK since it is very close to the original because the training model's r^2 is only about .017 less.

```
cor(AmesTest2$Price, NewHouse)^2

## [1] 0.876163

summary(House)$r.squared

## [1] 0.8587356

HouseShrinkage = summary(House)$r.squared - cor(AmesTest2$Price, NewHouse)^2
HouseShrinkage

## [1] -0.01742733
```

Part 7. Find a "fancy model":

Use AmesTrain?.csv, where ?? corresponds to your *new* homework #5 group number. In addition to the quantitative predictors, you may now consider models with

- Categorical variables - Just put these in the model and let R take care of making the indicator predictors (and picking one category to leave out). Use factor() to treat a numeric variable as categorical. You'll see the coefficients for each indicator when you look at the summary() and they will be grouped together in the ANOVA. Be careful, since adding a single categorical variable with a lot of categories might actually be adding a lot of new indicator terms.

Our group decided to start with the original model that was formed based on what I did in Homework #3 where I used stepwise selection. Then, we converted the variables Quality, Condition, and Bedroom into categorical variables because they are categories with measurements. We also added the qualitative variable HouseStyle because we expect it to impact the price. Now that some of the variables are categorical, we used the stepwise selection process to determine which predictors are useful.

```
Cate_House = lm(Price ~ factor(Quality) + GroundSF + BasementSF + YearBuilt +
  GarageSF + LotArea + factor(Condition) + BasementFBath + YearRemodel +
  Fireplaces + ScreenPorchSF + LotFrontage + factor(Bedroom) +
  EnclosedPorchSF + factor(HouseStyle),
  data = Adjusted_TrainData)
MSE = (summary(Cate_House)$sigma)^2
nothing = lm(Price~1, data=Adjusted_TrainData)
step(nothing, scope = list(upper=Cate_House), scale = MSE, trace=FALSE)
```



```
##
## Call:
## lm(formula = Price ~ factor(Quality) + GroundSF + YearBuilt +
##     BasementSF + factor(Condition) + LotArea + BasementFBath +
##     GarageSF + Fireplaces + YearRemodel + ScreenPorchSF + EnclosedPorchSF
##     LotFrontage, data = Adjusted_TrainData)
##
## Coefficients:
##      (Intercept)      factor(Quality)3      factor(Quality)4
factor(Quality)5      -1.597e+03      2.108e+01      1.337e+01
1.279e+01
##    factor(Quality)6      factor(Quality)7      factor(Quality)8
factor(Quality)9      1.328e+01      2.770e+01      5.847e+01
1.111e+02
##    factor(Quality)10      GroundSF      YearBuilt
BasementSF      1.376e+02      5.193e-02      4.993e-01
2.340e-02
##    factor(Condition)2      factor(Condition)3      factor(Condition)4
factor(Condition)5      -3.463e+01      -4.780e+00      1.642e+01
3.094e+01
##    factor(Condition)6      factor(Condition)7      factor(Condition)8
factor(Condition)9      3.465e+01      3.891e+01      4.239e+01
3.894e+01
##      LotArea      BasementFBath      GarageSF
Fireplaces      7.119e-04      1.154e+01      3.036e-02
9.115e+00
##      YearRemodel      ScreenPorchSF      EnclosedPorchSF
LotFrontage      2.996e-01      6.349e-02      4.413e-02
6.395e-02
```

- Transformations of predictors - You can include functions of quantitative predictors. Probably best to use the I() notation so you don't need to create new columns when you run the predictions for the test data.
- Transformations of the response - You might address curvature or skewness in residual plots by transforming the response prices with a function like log(Price), sqrt(Price), Price^2, etc.. These should generally not need the I() notation to make these adjustments. IMPORTANT: If you transform Price, be sure to reverse the transformation when making final predictions!

We created a linear model with the new amount of predictors and saw that there's linearity and constant variance, but it is skewed at the left and right tails. Therefore, we tried

different transformations on the response and square rooting it was the best as the R^2 increased and the right tail skew improved. After transforming the response, we looked at the plots of the predictors in relation to Price. We decided to square the GroundSF and BasementSF because their plots show a curve. This led to a higher R^2 , the linearity and constant variance were met, and the right tail skew continued to improve. We tried other predictor transformations as well but they did not make huge differences, so we decided not to transform them.

```
BestCate_House = lm(Price ~ factor(Quality) + GroundSF + YearBuilt +
  BasementSF +
    factor(Condition) + LotArea + BasementFBath + GarageSF +
    Fireplaces + YearRemodel + ScreenPorchSF + EnclosedPorchSF + LotFrontage,
  data = Adjusted_TrainData)
summary(BestCate_House)
```

```
##
## Call:
## lm(formula = Price ~ factor(Quality) + GroundSF + YearBuilt +
##     BasementSF + factor(Condition) + LotArea + BasementFBath +
##     GarageSF + Fireplaces + YearRemodel + ScreenPorchSF + EnclosedPorchSF
## +
##     LotFrontage, data = Adjusted_TrainData)
##
## Residuals:
```

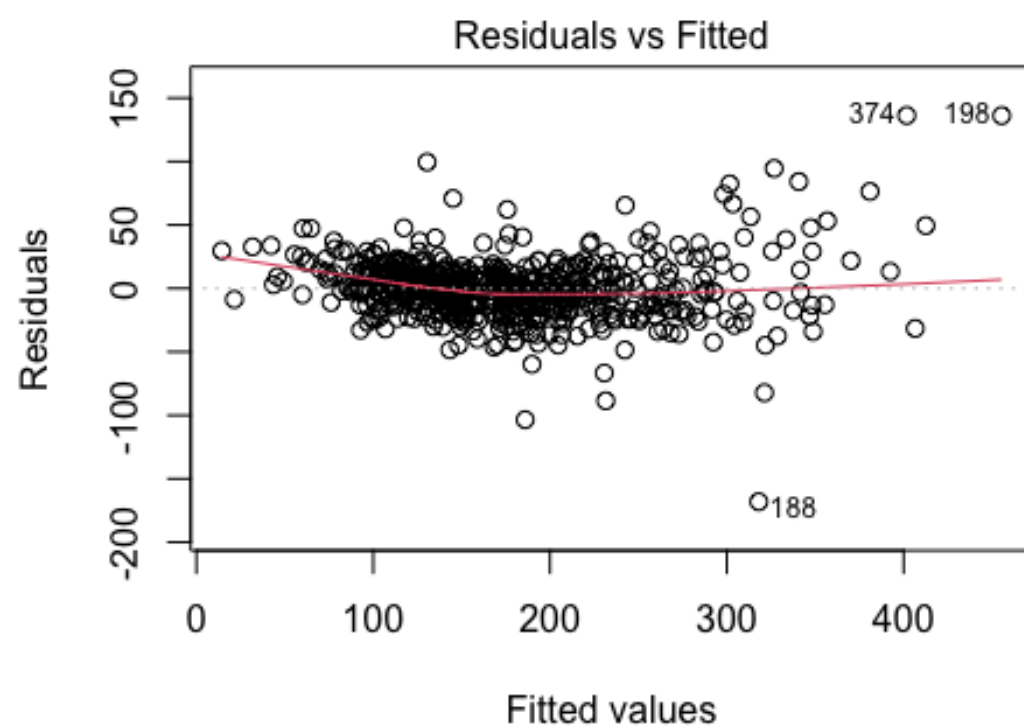
	Min	1Q	Median	3Q	Max
	-168.06	-13.13	-0.58	11.10	136.25

```
##
## Coefficients:
```

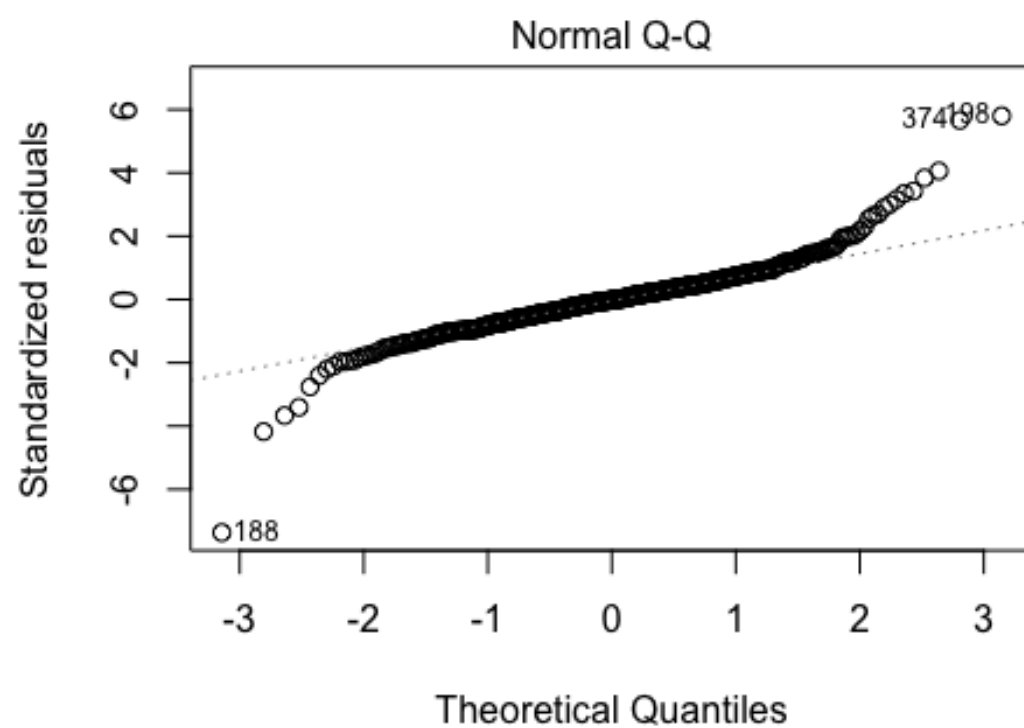
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.597e+03	1.482e+02	-10.780	< 2e-16 ***
factor(Quality)3	2.108e+01	1.993e+01	1.058	0.290601
factor(Quality)4	1.337e+01	1.844e+01	0.725	0.468693
factor(Quality)5	1.279e+01	1.822e+01	0.702	0.482966
factor(Quality)6	1.328e+01	1.843e+01	0.721	0.471447
factor(Quality)7	2.770e+01	1.873e+01	1.478	0.139826
factor(Quality)8	5.847e+01	1.907e+01	3.067	0.002268 **
factor(Quality)9	1.111e+02	2.008e+01	5.532	4.83e-08 ***
factor(Quality)10	1.376e+02	2.649e+01	5.194	2.86e-07 ***
GroundSF	5.193e-02	3.099e-03	16.755	< 2e-16 ***
YearBuilt	4.993e-01	6.456e-02	7.734	4.73e-14 ***
BasementSF	2.340e-02	3.581e-03	6.534	1.42e-10 ***
factor(Condition)2	-3.463e+01	2.538e+01	-1.365	0.172933
factor(Condition)3	-4.780e+00	2.152e+01	-0.222	0.824267
factor(Condition)4	1.642e+01	2.088e+01	0.786	0.431980
factor(Condition)5	3.094e+01	2.035e+01	1.520	0.129023
factor(Condition)6	3.465e+01	2.039e+01	1.700	0.089751 .
factor(Condition)7	3.891e+01	2.038e+01	1.909	0.056794 .
factor(Condition)8	4.239e+01	2.097e+01	2.022	0.043674 *
factor(Condition)9	3.894e+01	2.236e+01	1.742	0.082094 .
LotArea	7.119e-04	9.578e-05	7.433	3.91e-13 ***

```
## BasementFBath      1.154e+01  2.110e+00  5.472 6.67e-08 ***
## GarageSF           3.036e-02  6.366e-03  4.769 2.35e-06 ***
## Fireplaces         9.115e+00  1.979e+00  4.605 5.09e-06 ***
## YearRemodel        2.996e-01  7.410e-02  4.044 5.99e-05 ***
## ScreenPorchSF      6.349e-02  1.646e-02  3.858 0.000127 ***
## EnclosedPorchSF    4.413e-02  2.058e-02  2.144 0.032451 *
## LotFrontage        6.395e-02  3.400e-02  1.881 0.060488 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25 on 572 degrees of freedom
## Multiple R-squared:  0.8884, Adjusted R-squared:  0.8831
## F-statistic: 168.7 on 27 and 572 DF,  p-value: < 2.2e-16

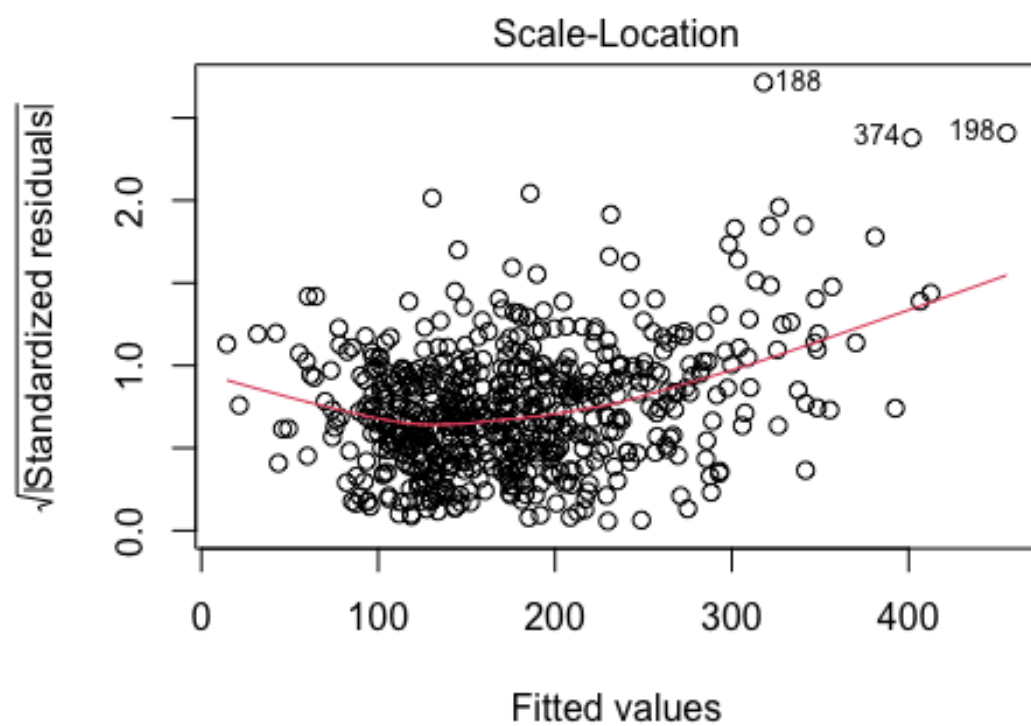
plot(BestCate_House)
```



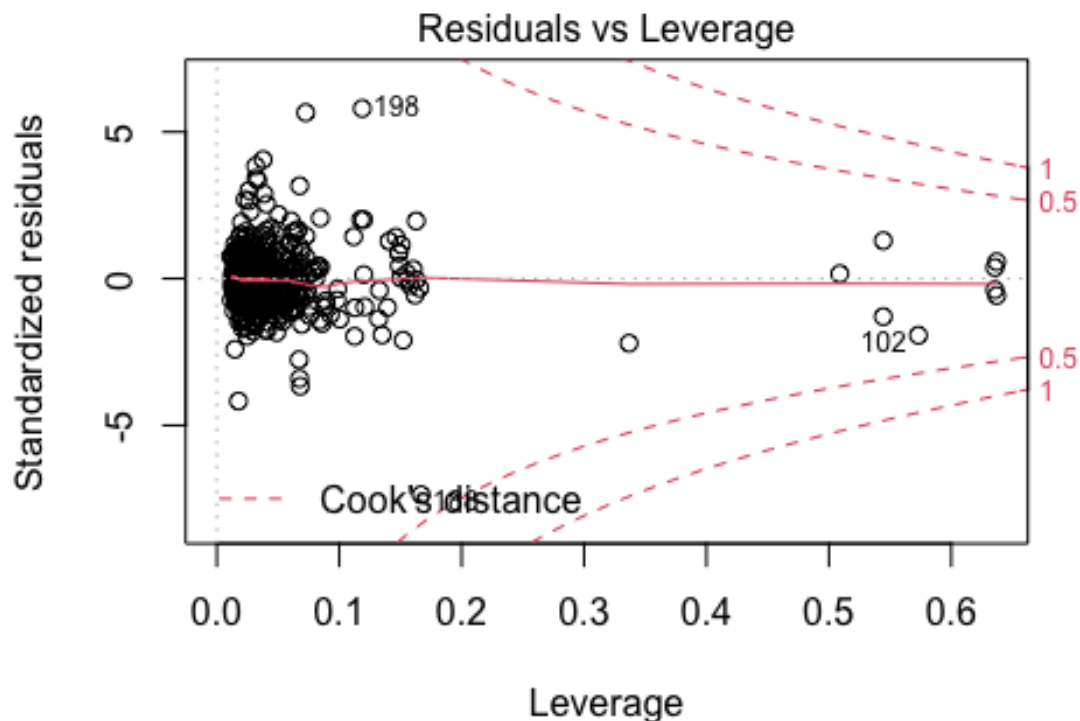
$\text{price} \sim \text{factor}(\text{Quality}) + \text{GroundSF} + \text{YearBuilt} + \text{BasementSF} + \text{fact}$



$\text{price} \sim \text{factor}(\text{Quality}) + \text{GroundSF} + \text{YearBuilt} + \text{BasementSF} + \text{fact}$



$\text{price} \sim \text{factor}(\text{Quality}) + \text{GroundSF} + \text{YearBuilt} + \text{BasementSF} + \text{fact}$



$\sqrt{\text{price}} \sim \text{factor}(\text{Quality}) + \text{GroundSF} + \text{YearBuilt} + \text{BasementSF} + \text{fact}$

```
Transformed_Model = lm(sqrt(Price) ~ factor(Quality) + GroundSF + YearBuilt +
  BasementSF +
    factor(Condition) + LotArea + BasementFBath + GarageSF +
    Fireplaces + YearRemodel + ScreenPorchSF + EnclosedPorchSF + LotFrontage,
  data = Adjusted_TrainData)
summary(Transformed_Model)

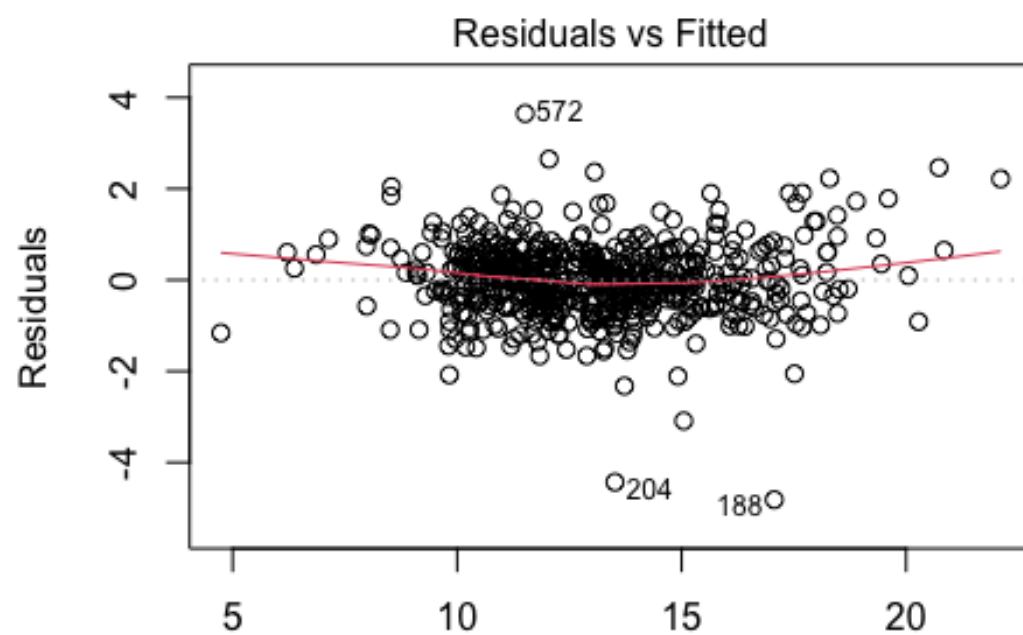
##
## Call:
## lm(formula = sqrt(Price) ~ factor(Quality) + GroundSF + YearBuilt +
##   BasementSF + factor(Condition) + LotArea + BasementFBath +
##   GarageSF + Fireplaces + YearRemodel + ScreenPorchSF + EnclosedPorchSF
## +
##   LotFrontage, data = Adjusted_TrainData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.8163 -0.4490  0.0032  0.4548  3.6499
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.944e+01  4.805e+00 -12.369  < 2e-16 ***
## factor(Quality)3    2.329e+00  6.462e-01   3.604  0.000341 ***
```

```

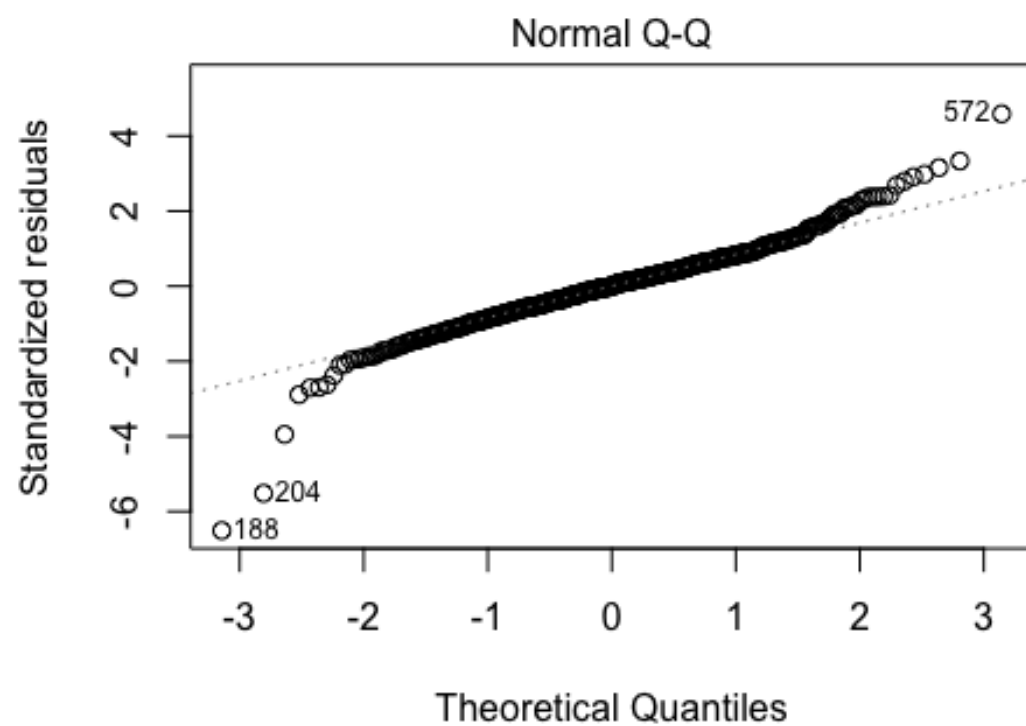
## factor(Quality)4      2.279e+00  5.979e-01   3.812 0.000153 ***
## factor(Quality)5      2.493e+00  5.909e-01   4.219 2.85e-05 ***
## factor(Quality)6      2.533e+00  5.978e-01   4.238 2.63e-05 ***
## factor(Quality)7      3.065e+00  6.075e-01   5.045 6.10e-07 ***
## factor(Quality)8      3.886e+00  6.183e-01   6.285 6.51e-10 ***
## factor(Quality)9      5.127e+00  6.512e-01   7.873 1.75e-14 ***
## factor(Quality)10     6.222e+00  8.589e-01   7.244 1.41e-12 ***
## GroundSF              1.826e-03  1.005e-04  18.167 < 2e-16 ***
## YearBuilt              2.030e-02  2.093e-03   9.698 < 2e-16 ***
## BasementSF            8.364e-04  1.161e-04   7.203 1.86e-12 ***
## factor(Condition)2    -2.515e+00  8.230e-01  -3.056 0.002351 **
## factor(Condition)3    -8.954e-01  6.977e-01  -1.283 0.199856
## factor(Condition)4     3.985e-02  6.769e-01   0.059 0.953079
## factor(Condition)5     5.733e-01  6.599e-01   0.869 0.385322
## factor(Condition)6     7.756e-01  6.611e-01   1.173 0.241204
## factor(Condition)7     9.382e-01  6.610e-01   1.419 0.156324
## factor(Condition)8     1.027e+00  6.798e-01   1.510 0.131483
## factor(Condition)9     9.186e-01  7.250e-01   1.267 0.205668
## LotArea                2.037e-05  3.106e-06   6.558 1.22e-10 ***
## BasementFBath          3.819e-01  6.842e-02   5.582 3.67e-08 ***
## GarageSF               1.226e-03  2.064e-04   5.940 4.96e-09 ***
## Fireplaces             3.696e-01  6.418e-02   5.759 1.39e-08 ***
## YearRemodel            1.212e-02  2.403e-03   5.046 6.07e-07 ***
## ScreenPorchSF          1.960e-03  5.336e-04   3.674 0.000261 ***
## EnclosedPorchSF        1.587e-03  6.675e-04   2.377 0.017761 *
## LotFrontage            2.252e-03  1.102e-03   2.042 0.041578 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8106 on 572 degrees of freedom
## Multiple R-squared:  0.9074, Adjusted R-squared:  0.9031
## F-statistic: 207.7 on 27 and 572 DF,  p-value: < 2.2e-16

plot(Transformed_Model)

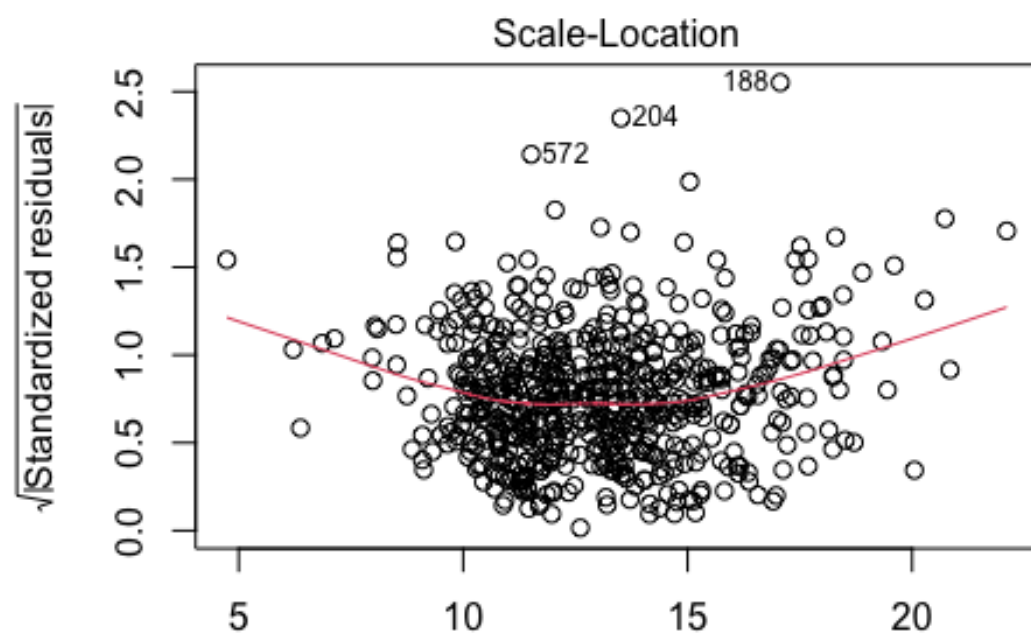
```

$\text{sqrt}(\text{Price}) \sim \text{factor}(\text{Quality}) + \text{GroundSF} + \text{YearBuilt} + \text{BasementSF}$

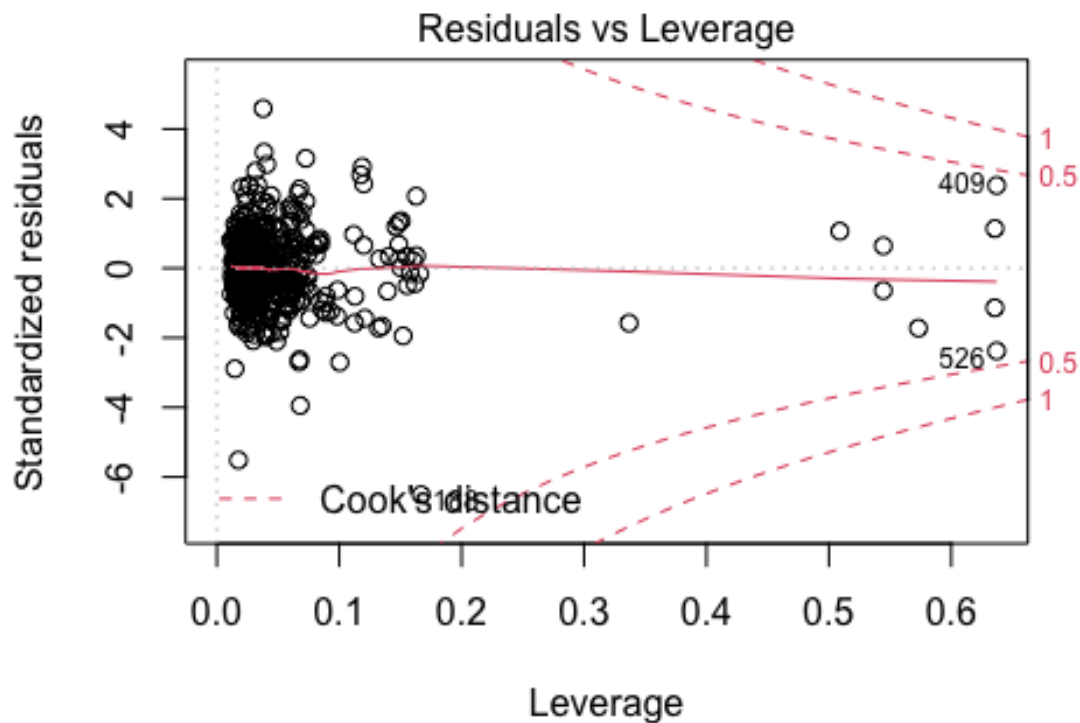


$\text{sqrt}(\text{Price}) \sim \text{factor}(\text{Quality}) + \text{GroundSF} + \text{YearBuilt} + \text{BasementSF}$



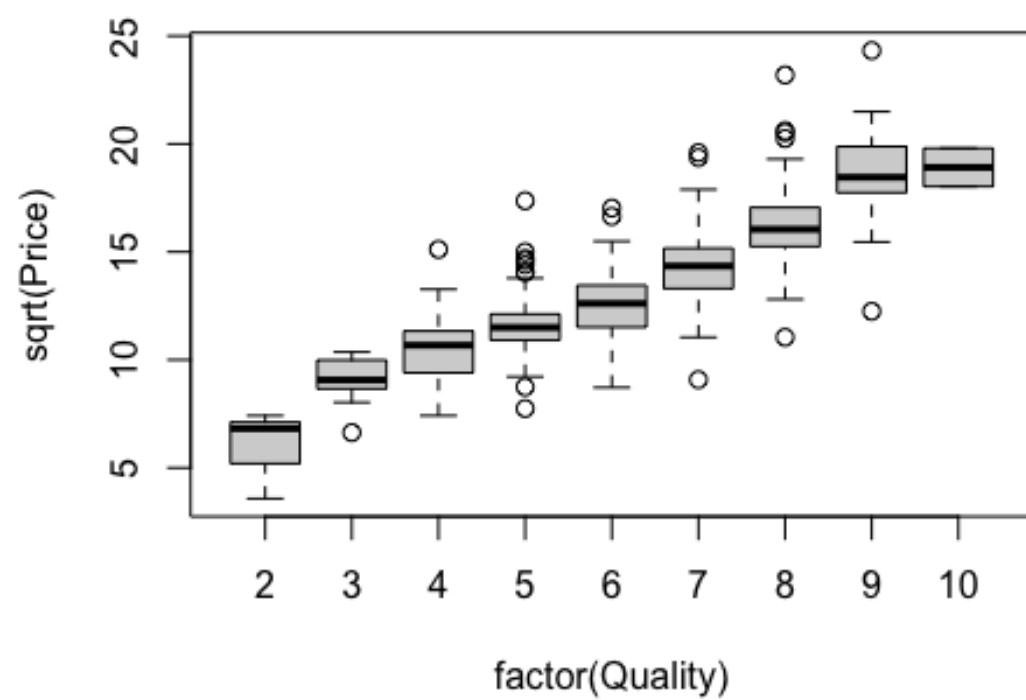
Fitted values

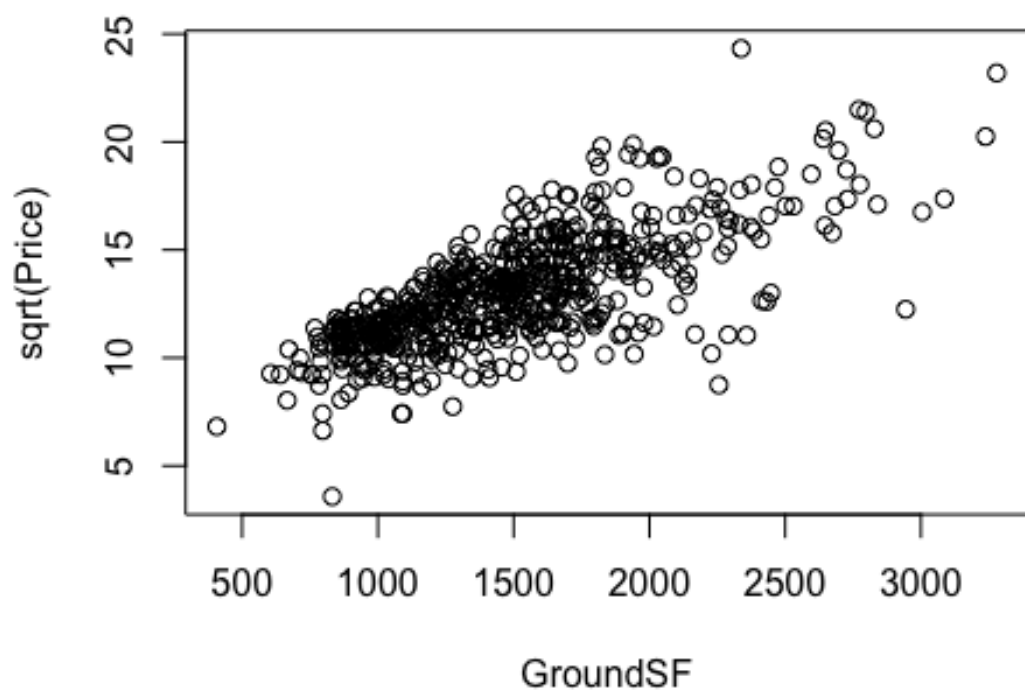
$\text{sqrt}(\text{Price}) \sim \text{factor}(\text{Quality}) + \text{GroundSF} + \text{YearBuilt} + \text{BasementSF}$

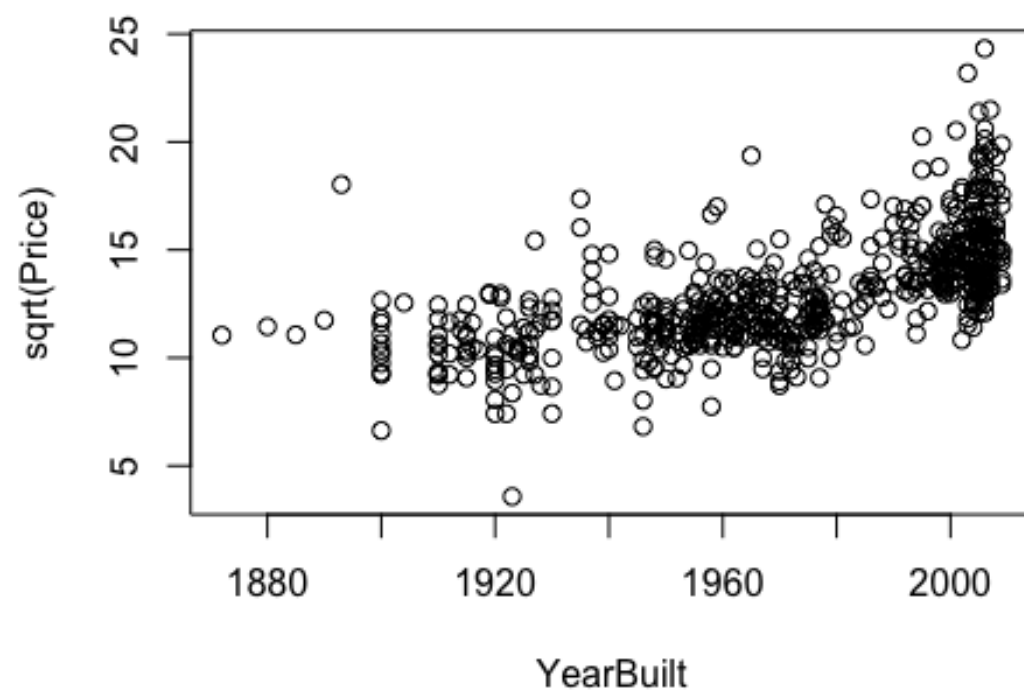


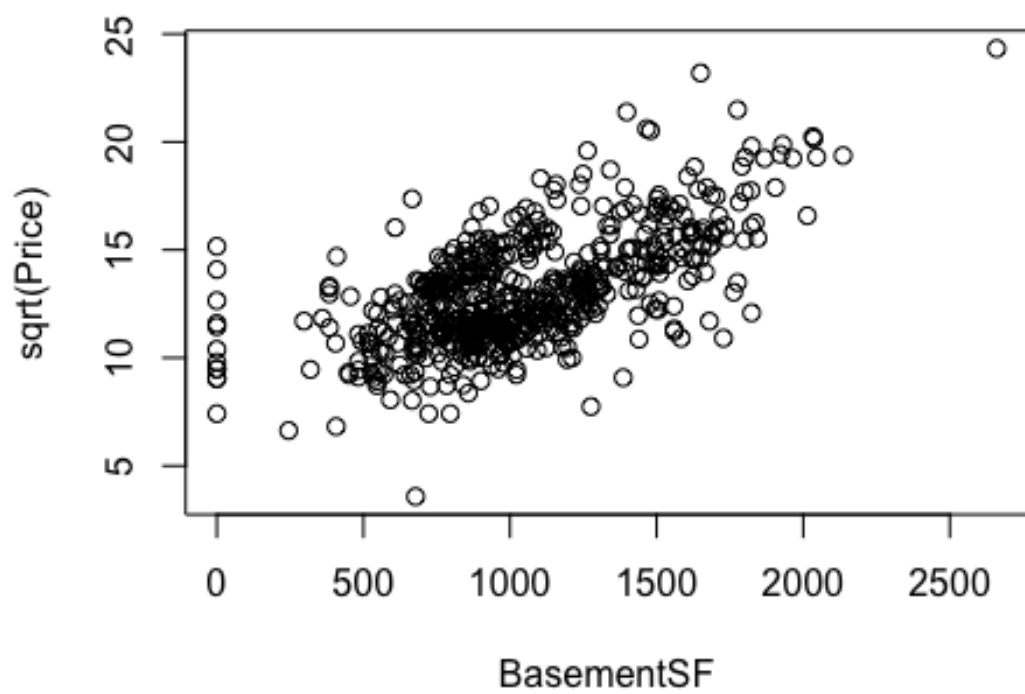
`sqrt(Price) ~ factor(Quality) + GroundSF + YearBuilt + BasementSF`

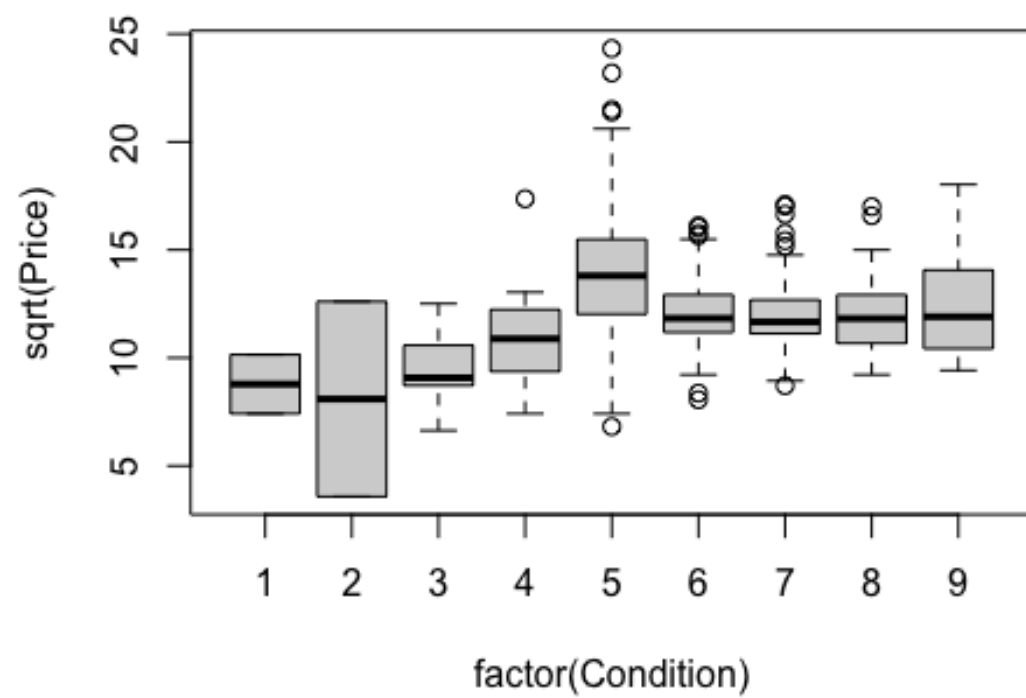
```
plot(sqrt(Price) ~ factor(Quality) + GroundSF + YearBuilt + BasementSF +
     factor(Condition) + LotArea + BasementFBath + GarageSF +
     Fireplaces + YearRemodel + ScreenPorchSF + EnclosedPorchSF + LotFrontage,
     data = Adjusted_TrainData)
```

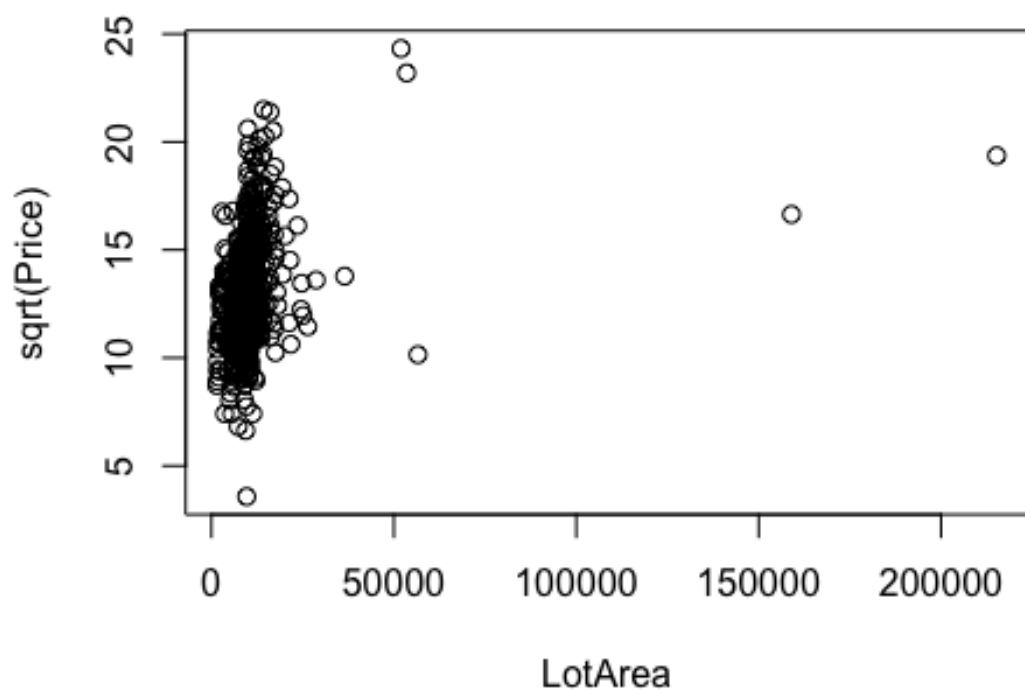


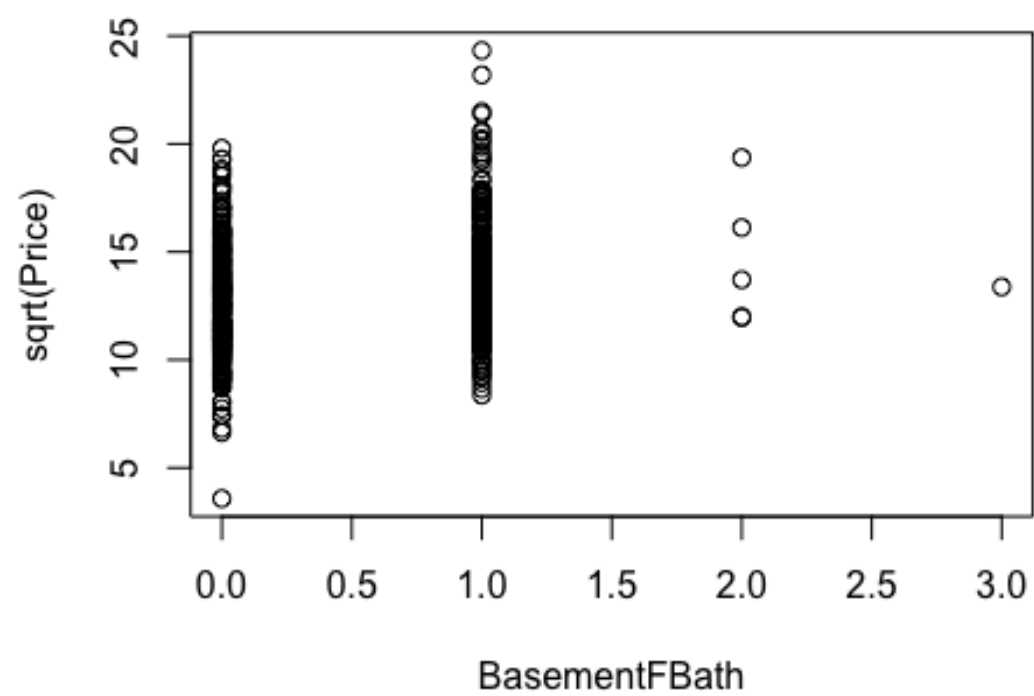


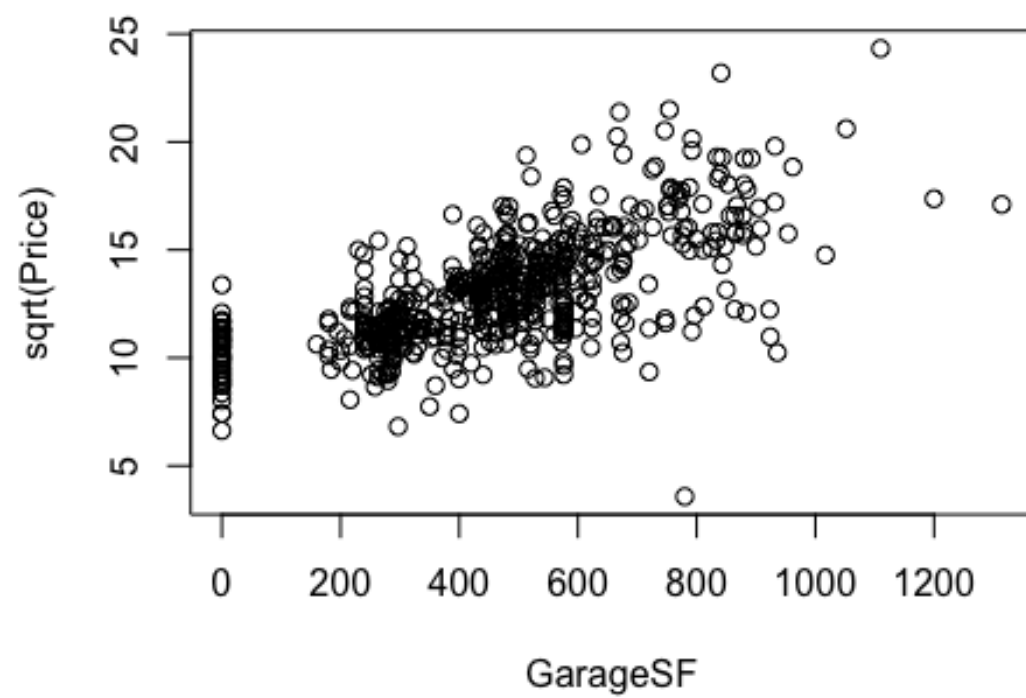


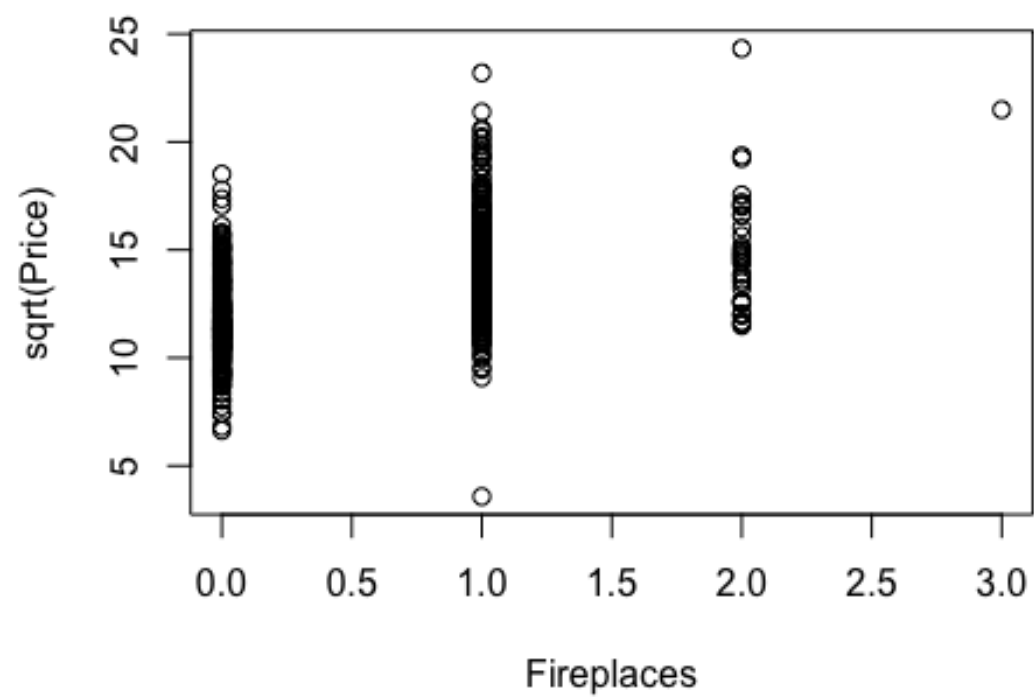


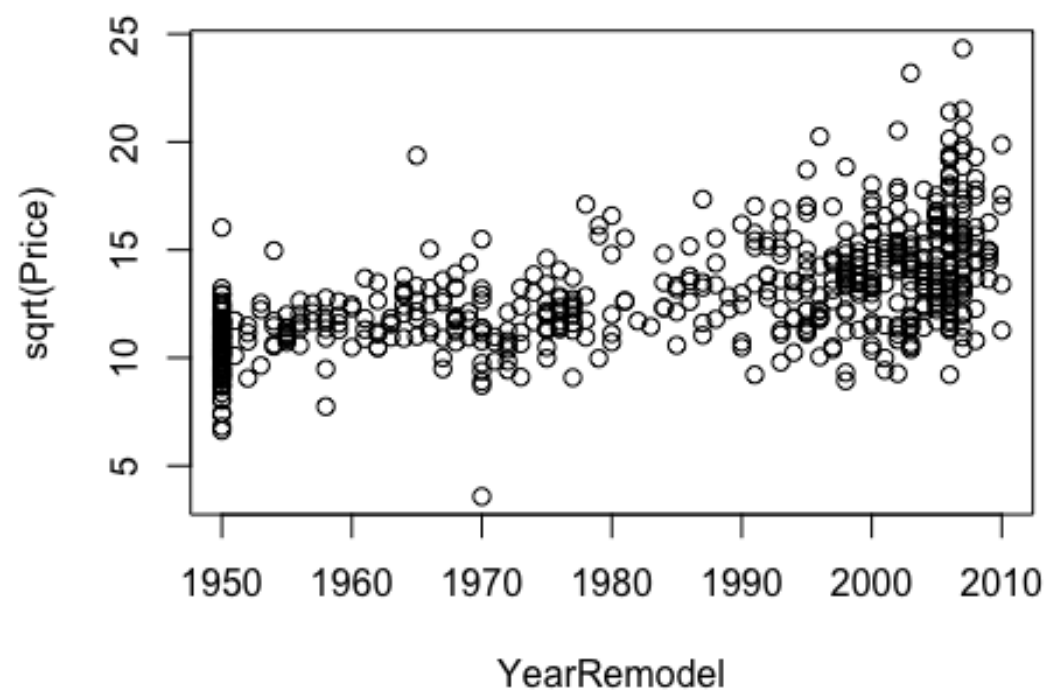


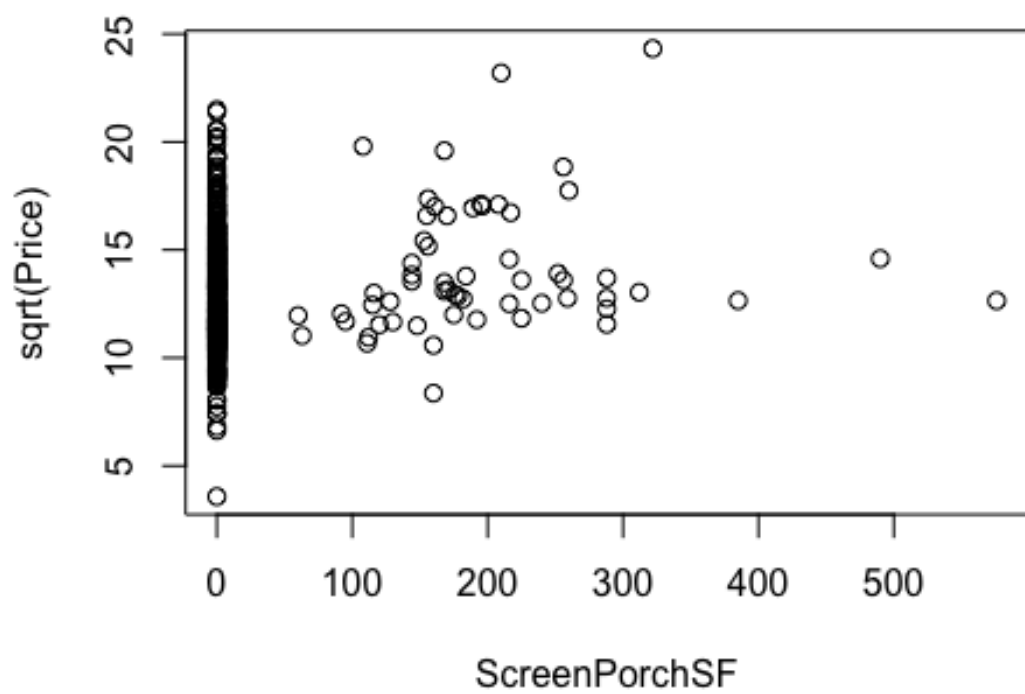


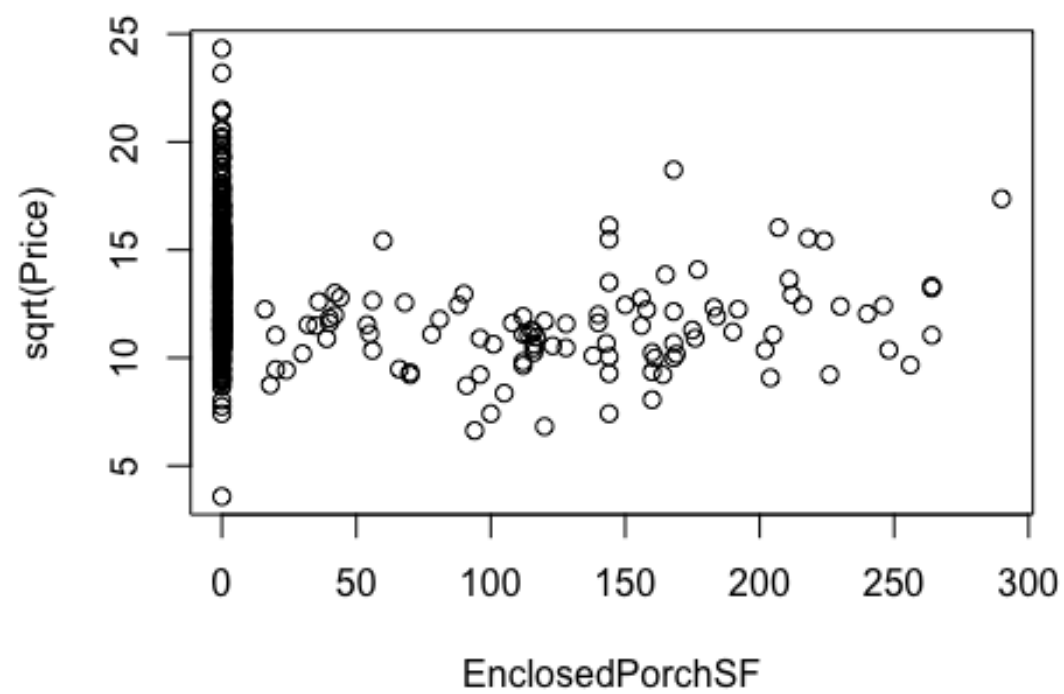


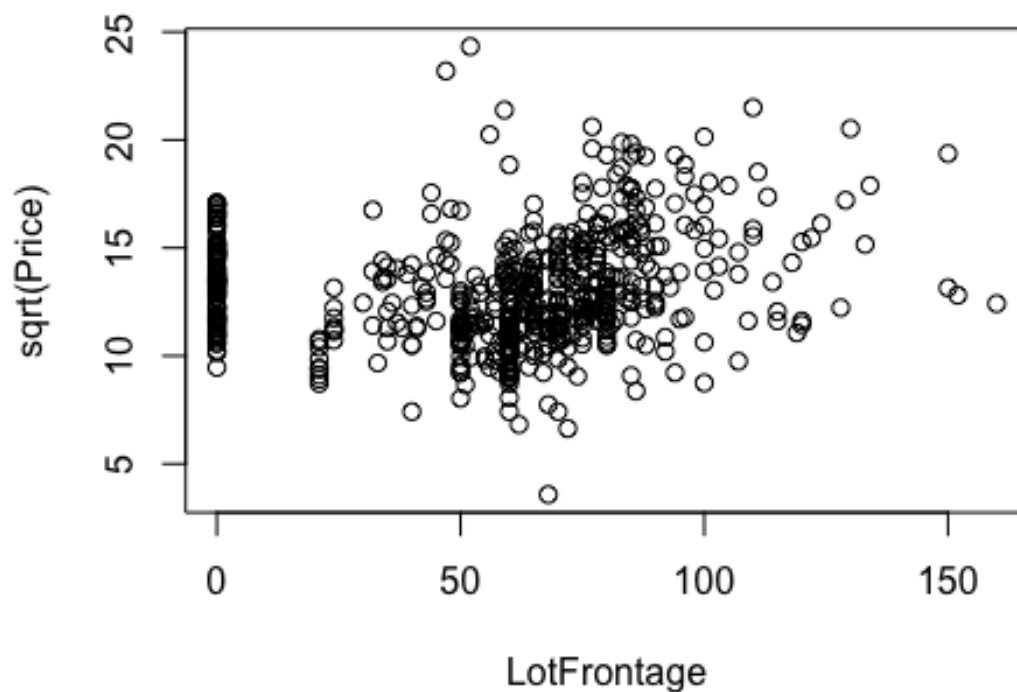












```
Transformed_Model2 = lm(sqrt(Price) ~ factor(Quality) + I(GroundSF^2) +
YearBuilt + I(BasementSF^2) +
  factor(Condition) + LotArea + BasementFBath + GarageSF +
  Fireplaces + YearRemodel + ScreenPorchSF + EnclosedPorchSF + LotFrontage,
  data = Adjusted_TrainData)
summary(Transformed_Model2)
```

```
##
## Call:
## lm(formula = sqrt(Price) ~ factor(Quality) + I(GroundSF^2) +
##     YearBuilt + I(BasementSF^2) + factor(Condition) + LotArea +
##     BasementFBath + GarageSF + Fireplaces + YearRemodel + ScreenPorchSF +
##     EnclosedPorchSF + LotFrontage, data = Adjusted_TrainData)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
##	-4.9746	-0.4505	0.0161	0.4697	3.3059

```
##
## Coefficients:
```

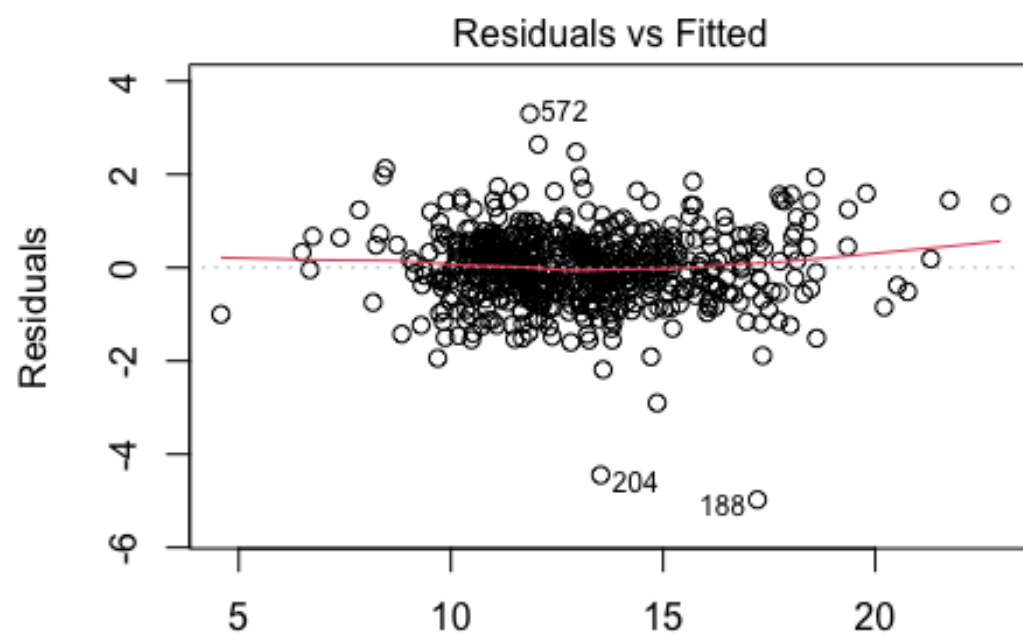
	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	-5.793e+01	4.730e+00	-12.248	< 2e-16 ***
## factor(Quality)3	2.766e+00	6.331e-01	4.368	1.49e-05 ***
## factor(Quality)4	2.740e+00	5.850e-01	4.684	3.52e-06 ***

```

## factor(Quality)5      3.055e+00  5.766e-01  5.298 1.68e-07 ***
## factor(Quality)6      3.156e+00  5.826e-01  5.418 8.91e-08 ***
## factor(Quality)7      3.697e+00  5.905e-01  6.260 7.55e-10 ***
## factor(Quality)8      4.419e+00  6.013e-01  7.349 6.93e-13 ***
## factor(Quality)9      5.440e+00  6.360e-01  8.553 < 2e-16 ***
## factor(Quality)10     6.578e+00  8.409e-01  7.823 2.51e-14 ***
## I(GroundSF^2)         5.228e-07  2.773e-08 18.850 < 2e-16 ***
## YearBuilt             1.968e-02  2.048e-03  9.608 < 2e-16 ***
## I(BasementSF^2)       4.434e-07  5.245e-08  8.454 2.36e-16 ***
## factor(Condition)2    -2.831e+00  8.106e-01 -3.492 0.000516 ***
## factor(Condition)3    -1.400e+00  6.857e-01 -2.042 0.041576 *
## factor(Condition)4    -4.097e-01  6.655e-01 -0.616 0.538417
## factor(Condition)5     1.423e-01  6.486e-01  0.219 0.826400
## factor(Condition)6     3.458e-01  6.493e-01  0.532 0.594595
## factor(Condition)7     4.894e-01  6.491e-01  0.754 0.451157
## factor(Condition)8     4.704e-01  6.665e-01  0.706 0.480561
## factor(Condition)9     3.626e-01  7.113e-01  0.510 0.610385
## LotArea               1.723e-05  3.106e-06  5.547 4.44e-08 ***
## BasementFBath         3.714e-01  6.689e-02  5.552 4.33e-08 ***
## GarageSF              1.172e-03  2.036e-04  5.756 1.41e-08 ***
## Fireplaces            4.102e-01  6.246e-02  6.567 1.15e-10 ***
## YearRemodel           1.283e-02  2.359e-03  5.440 7.91e-08 ***
## ScreenPorchSF         1.767e-03  5.266e-04  3.355 0.000847 ***
## EnclosedPorchSF       1.493e-03  6.572e-04  2.272 0.023437 *
## LotFrontage           2.361e-03  1.087e-03  2.172 0.030243 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.798 on 572 degrees of freedom
## Multiple R-squared:  0.9103, Adjusted R-squared:  0.9061
## F-statistic: 215 on 27 and 572 DF, p-value: < 2.2e-16

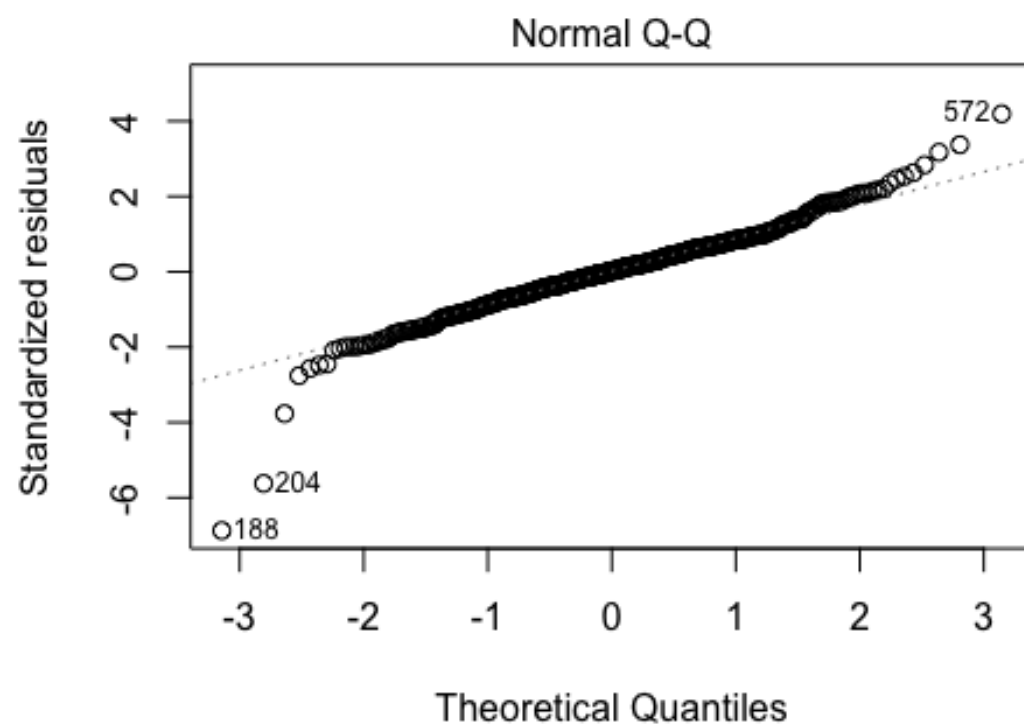
plot(Transformed_Model2)

```

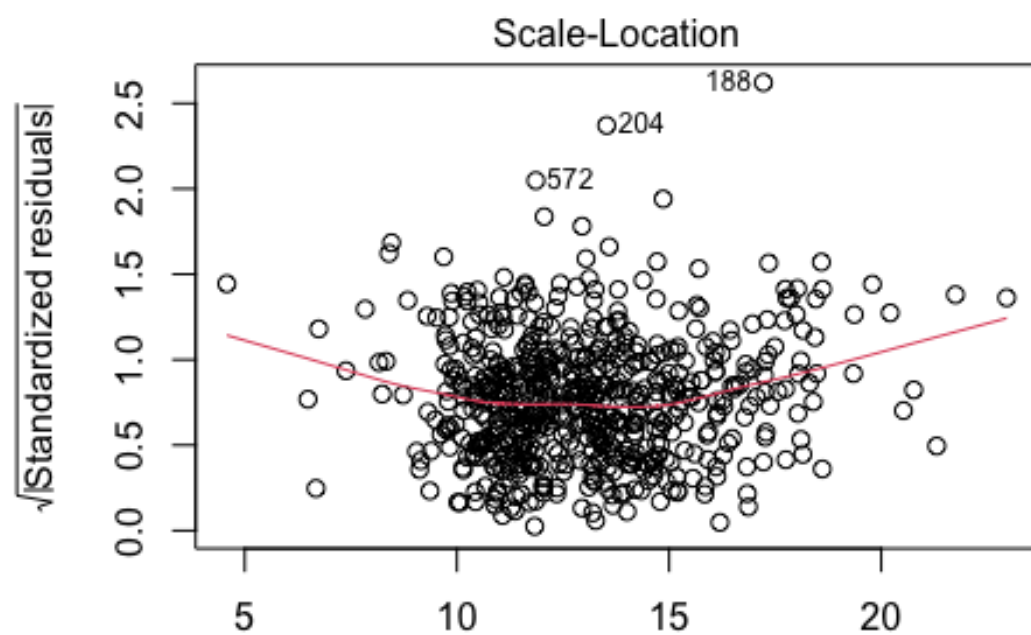


Fitted values

$\sqrt{\text{Price}} \sim \text{factor}(\text{Quality}) + \text{I}(\text{GroundSF}^2) + \text{YearBuilt} + \text{I}(\text{Basem})$

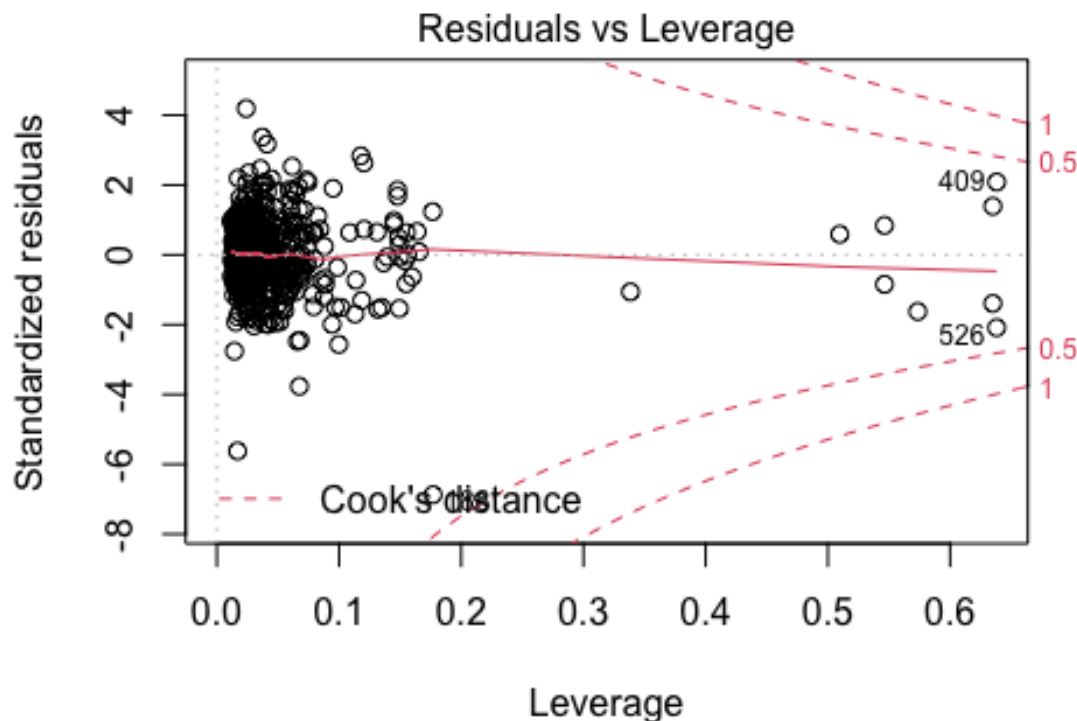


$\sqrt{\text{Price}} \sim \text{factor}(\text{Quality}) + \text{l}(\text{GroundSF}^2) + \text{YearBuilt} + \text{l}(\text{Basem})$



Fitted values

$\sqrt{\text{Price}} \sim \text{factor}(\text{Quality}) + \text{I}(\text{GroundSF}^2) + \text{YearBuilt} + \text{I}(\text{Basem})$



`sqrt(Price) ~ factor(Quality) + I(GroundSF^2) + YearBuilt + I(BasementSF^2)`

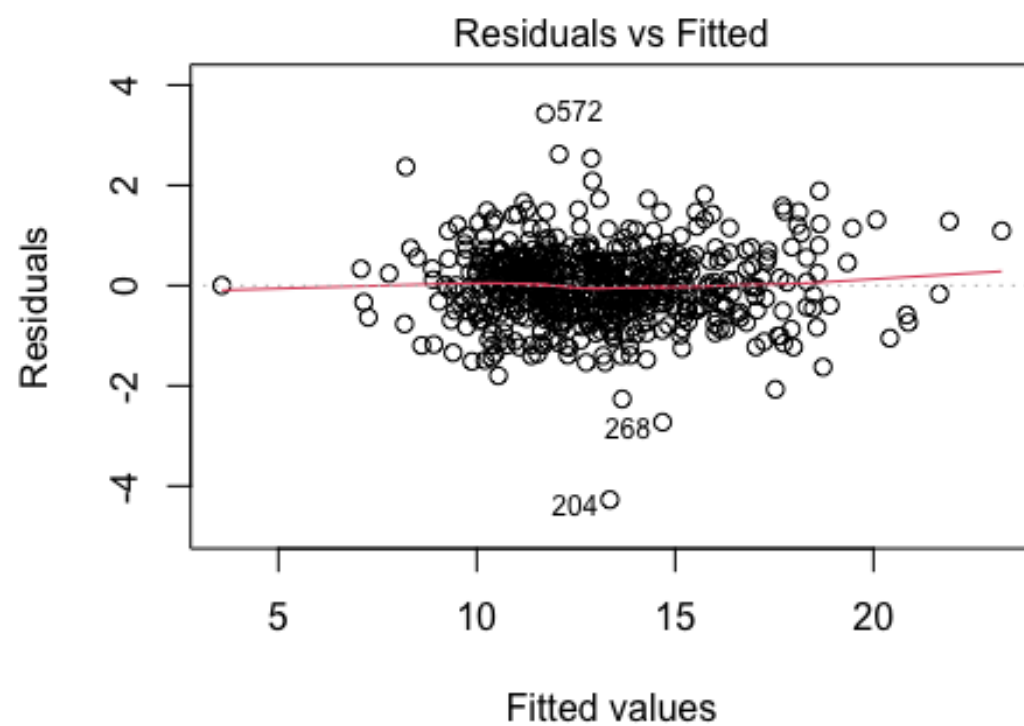
- Combinations of variables - This might include interactions or other combinations. You do not need the `I()` notation when making an interaction using a categorical predictor (e.g. `GroundSF*CentralAir`).

We added two interaction term to the model, `factor(Condition)*YearRemodel` and `GroundSF*BasementSF`. The term `factor(Condition)*YearRemodel` was added because the two variables have a relation since the `YearRemodel` would impact the condition of the house, so we expected the term would improve our model. We added `GroundSF*BasementSF` to the model because the term could represent the square feet of each house, which would affect Price. We did try other interaction terms but they either decreased R^2 or increased it by a small amount. Throughout all of the transformations, there were data points such as the 188th house with a high residual and the 409th and 526th house with high leverages, but they were not high enough to be considered influential points. With the final transformed model, the only problematic house is the 188th house which is an influential point. However, the linear conditions are met and there's a high R^2 of .92 so this is a good model to predict the house prices.

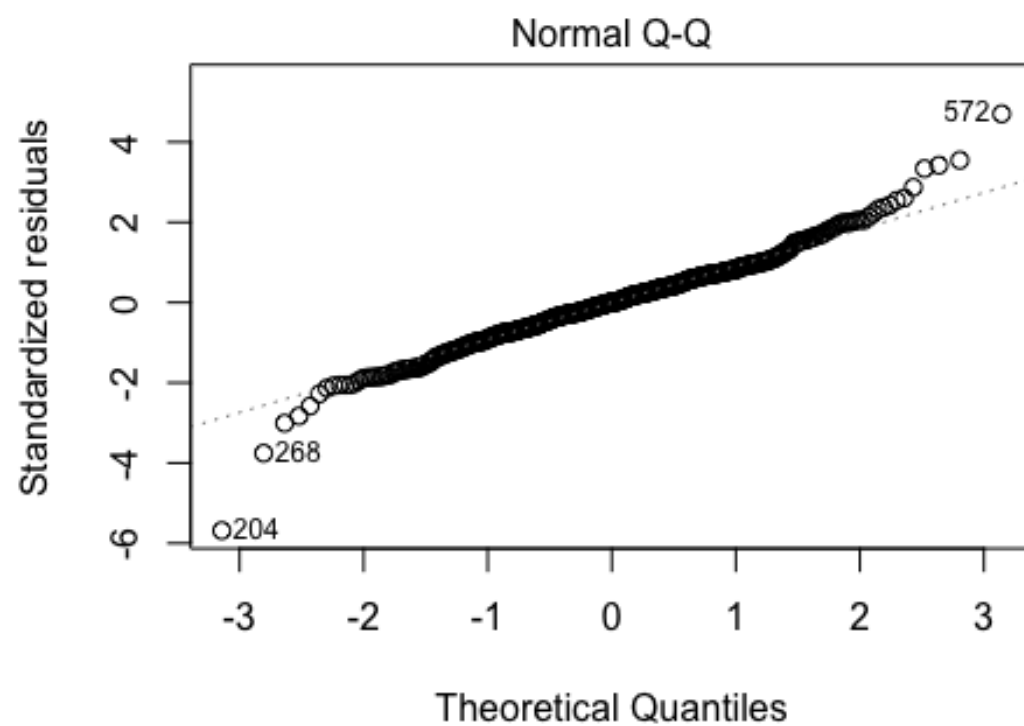
```
Transformed_Model3 = lm(sqrt(Price) ~ factor(Quality) + I(GroundSF^2) +
  YearBuilt + I(BasementSF^2) +
  factor(Condition) + LotArea + BasementFBath + GarageSF +
  Fireplaces + YearRemodel + ScreenPorchSF + EnclosedPorchSF + LotFrontage
+ factor(Condition)*YearRemodel + GroundSF*BasementSF, data =
```

```
Adjusted_TrainData)  
plot(Transformed_Model3)
```

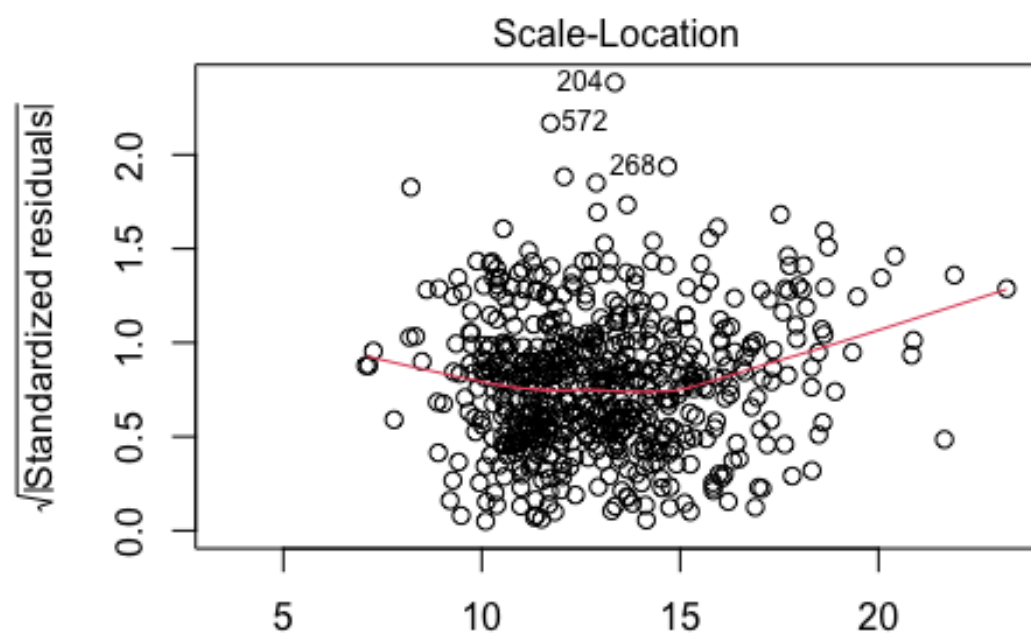
```
## Warning: not plotting observations with leverage one:  
##    409, 526
```



$\sqrt{\text{Price}} \sim \text{factor}(\text{Quality}) + \text{I}(\text{GroundSF}^2) + \text{YearBuilt} + \text{I}(\text{Basem})$

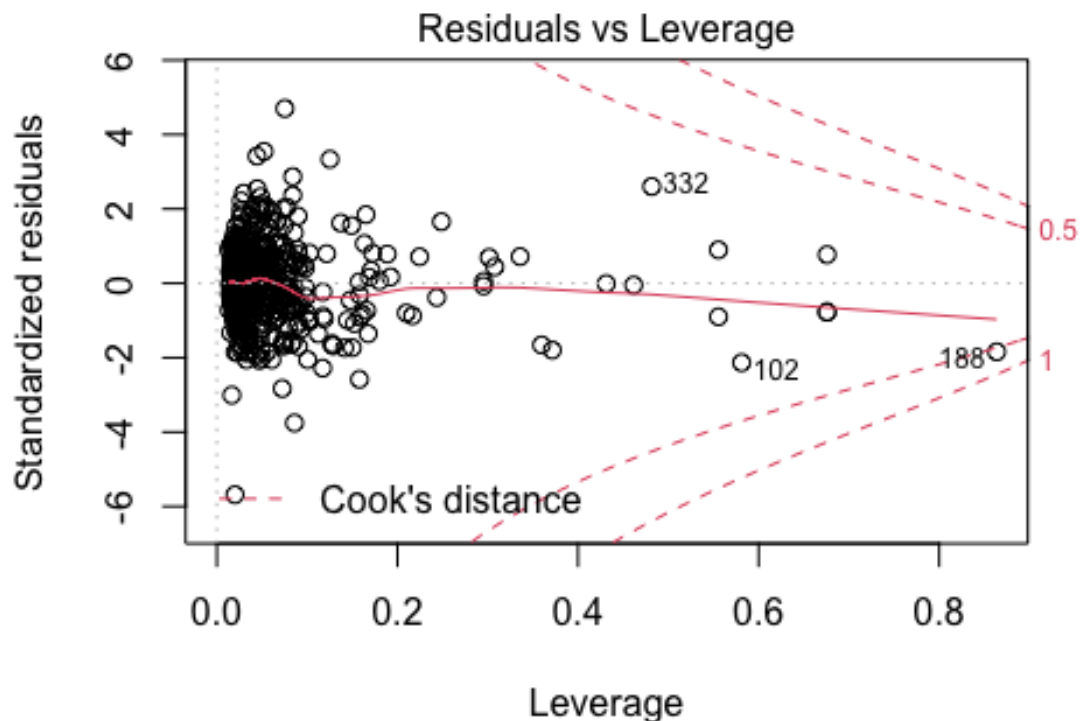


$\sqrt{\text{Price}} \sim \text{factor}(\text{Quality}) + \text{l}(\text{GroundSF}^2) + \text{YearBuilt} + \text{l}(\text{Basem})$



Fitted values

$\sqrt{\text{Price}} \sim \text{factor}(\text{Quality}) + \text{I}(\text{GroundSF}^2) + \text{YearBuilt} + \text{I}(\text{Basem})$



```
sqrt(Price) ~ factor(Quality) + I(GroundSF^2) + YearBuilt + I(BasementSF^2) + factor(Condition) + LotArea +
```

```
summary(Transformed_Model13)
```

```
##
## Call:
## lm(formula = sqrt(Price) ~ factor(Quality) + I(GroundSF^2) +
##     YearBuilt + I(BasementSF^2) + factor(Condition) + LotArea +
##     BasementFBath + GarageSF + Fireplaces + YearRemodel + ScreenPorchSF +
##     EnclosedPorchSF + LotFrontage + factor(Condition) * YearRemodel +
##     GroundSF * BasementSF, data = Adjusted_TrainData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2720 -0.4579  0.0023  0.4448  3.4327
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -8.131e+01  1.486e+02  -0.547  0.584497
## factor(Quality)3    1.491e+00  6.928e-01   2.152  0.031832 *
## factor(Quality)4    1.814e+00  6.509e-01   2.786  0.005513 **
## factor(Quality)5    2.156e+00  6.489e-01   3.322  0.000951 ***
## factor(Quality)6    2.197e+00  6.548e-01   3.356  0.000845 ***
## factor(Quality)7    2.678e+00  6.626e-01   4.041  6.07e-05 ***
## factor(Quality)8    3.357e+00  6.694e-01   5.015  7.12e-07 ***
```

```

## factor(Quality)9          4.624e+00  6.941e-01  6.662 6.46e-11 ***
## factor(Quality)10         5.489e+00  8.729e-01  6.287 6.49e-10 ***
## I(GroundSF^2)             3.008e-07  1.460e-07  2.060 0.039814 *
## YearBuilt                 1.687e-02  2.146e-03  7.860 1.97e-14 ***
## I(BasementSF^2)           1.409e-07  1.875e-07  0.751 0.452681
## factor(Condition)2        -3.578e+02  2.153e+02  -1.662 0.097114 .
## factor(Condition)3         4.063e+02  1.608e+02  2.528 0.011760 *
## factor(Condition)4         3.335e+01  1.503e+02  0.222 0.824454
## factor(Condition)5         1.326e+01  1.484e+02  0.089 0.928865
## factor(Condition)6         3.380e+01  1.489e+02  0.227 0.820548
## factor(Condition)7         4.433e+01  1.492e+02  0.297 0.766484
## factor(Condition)8         5.928e+01  1.502e+02  0.395 0.693300
## factor(Condition)9         2.635e-01  3.973e+00  0.066 0.947138
## LotArea                   2.012e-05  3.041e-06  6.617 8.55e-11 ***
## BasementFBath              3.661e-01  6.456e-02  5.671 2.27e-08 ***
## GarageSF                   1.366e-03  1.964e-04  6.956 9.75e-12 ***
## Fireplaces                 4.073e-01  6.133e-02  6.641 7.38e-11 ***
## YearRemodel                2.774e-02  7.637e-02  0.363 0.716507
## ScreenPorchSF              1.966e-03  5.074e-04  3.876 0.000119 ***
## EnclosedPorchSF            1.069e-03  6.426e-04  1.664 0.096614 .
## LotFrontage                1.749e-03  1.045e-03  1.673 0.094870 .
## GroundSF                   2.188e-04  3.796e-04  0.576 0.564561
## BasementSF                 -3.042e-04  4.141e-04  -0.735 0.462948
## factor(Condition)2:YearRemodel 1.793e-01  1.096e-01  1.636 0.102375
## factor(Condition)3:YearRemodel -2.083e-01  8.241e-02  -2.528 0.011752 *
## factor(Condition)4:YearRemodel -1.692e-02  7.705e-02  -0.220 0.826238
## factor(Condition)5:YearRemodel -6.483e-03  7.611e-02  -0.085 0.932152
## factor(Condition)6:YearRemodel -1.677e-02  7.636e-02  -0.220 0.826265
## factor(Condition)7:YearRemodel -2.202e-02  7.650e-02  -0.288 0.773598
## factor(Condition)8:YearRemodel -2.950e-02  7.701e-02  -0.383 0.701861
## factor(Condition)9:YearRemodel      NA      NA      NA      NA
## GroundSF:BasementSF         5.368e-07  3.080e-07  1.743 0.081942 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7592 on 562 degrees of freedom
## Multiple R-squared:  0.9202, Adjusted R-squared:  0.915
## F-statistic: 175.2 on 37 and 562 DF,  p-value: < 2.2e-16

```

Keep general track of the approaches you try and explain what guides your decisions as you select a new set of predictors (but again you don't need to give full details of every model you consider). Along the way you should consider some residual analysis.

Notes/Tips:

- WARNING: When using a categorical predictor with multiple categories in `regsubsets()`, R will create indicators and treat them as separate predictors when deciding which to put into a model. So you might get a model with quantitative predictors like `LotArea` and `GroundSF` along with specific indicators like `GarageQTA` and `HouseStyle1Story`. This may not be very useful, since we should generally use all

indicators for a categorical predictor if we include one in the model. On the other hand, when using the `step()` function, R will generally keep the multiple indicators for different categories of the same variable together as a unit.

- In some cases the indicators created for different categorical variables will have identical values. For example, if you include both `GarageC` and `GarageQ` in a model, R will produce values for each of the indicators. The indicators for `GarageQNone` and `GarageCNone` (equal to one only for houses that don't have a garage) will be identical. This may be handled differently in R depending on the procedure. `regsubsets()` may give a "warning" about variables being linearly dependent. You can still use the results, just be aware that some variables are completely dependent. `lm()` might give output with coefficients (and tests) of some predictors listed as NA. This is not a problem, R is just automatically deleting one of the redundant variables. If you are predicting for a house with no garage you might have a coefficient to use for `GarageQNone` but then you don't need to worry about having one for `GarageCNone`.
- If your residual analysis from homework #3 or an early model here suggest you might want to do a transformation for the response variable (Price), do so *before* fitting a lot more models. No sense fine tuning a set of predictors for Price, then deciding you should be predicting $\log(\text{Price})$ or Price^2 . So make that decision fairly early, but don't get too picky and expect to get perfect plot of residuals versus fits or an exact normal quantile plot.
- Similarly, if you decide that some data cases should be dropped from the training set, don't wait until late in the process to do so. For example, if you spot a *very* large residual you should look at the characteristics for that house to see if it should be deleted. Don't forget about the value of simple plots (like a scatterplot of Price vs. LotArea) for helping to see what is going on and recognize extreme cases. Be sure to document any adjustments you make in the final report.
- Comparing C_p from different predictor pools - While Mallows's C_p is a useful tool for comparing models from the same pool of predictors. You should not use it to compare models based on different predictor pools. For example, if you add a bunch of categorical variables to all the quantitative predictors from homework #3 to make a new "full" model, then find C_p from a model that you fit in homework #3, it will be worse than it was before. If you look at the formula for calculating C_p , you will see that all that has changed is MSE for the full model after adding the new batch of predictors.
- I should be able to follow the steps you use when selecting a model. I certainly don't need to see every bit of output, but it might help to include more of the R commands you use. For example, saying you used backward elimination is not very helpful when I don't know what you start with for the full model or pool of predictors (e.g. did you include Condition and Quality as numeric predictors? or did you decide to eliminate one of GroundSF, FirstSF, or SecondSF due to redundancy?). The easiest way to convey this in many cases is to show the R command you used. It is fine to

abbreviate the output (for example, delete many steps in a stepwise procedure using `trace=FALSE`), but it would be helpful if you identified the parts you do include. For example, a sentence like “After 12 steps of the stepwise procedure, we have the output below for the fitted model.” Similarly, I don’t need to see 600 residuals, using `head` and `sort` can show the important ones.

- Once you have settled on a response, made adjustments to the data (if needed), and chosen a set of predictors, be sure to include the `summary()` for your “fancy” model at this stage.

Part 8: Cross-validation for your “fancy” model

Redo the cross-validation analysis with your test data for your new fancy model. Use `AmesTest??.csv`, where ?? corresponds to your new group number from homework #5. Discuss how the various measures (mean of residuals, std. dev of residuals, shape of the distributions of residuals, cross-validation correlation, and shrinkage) compare to the results you had for your basic model. Don’t worry about looking for poorly predicted cases this time. If you transformed the response variable, consider how to take this into account for your residual analysis. In order to compare residuals they should have the same units!

The mean of the transformed model’s residuals is almost to 0 which is much smaller than the basic model’s mean of 3. The standard deviation of the residuals is .9658 which is close to the transformed model’s residual standard error of .759. It is very small in comparison to the basic model’s standard deviation which was very large at 27. The residuals are normally distributed although there are a few data points that cause a skew on the left and right tail, but it is less than the training model. The cross-validation correlation is high at .876 and the transformed model’s R^2 is .92. The shrinkage is close to 0.04 which is great, given that we had to shift all the quality 1 houses to quality 2. These statistics show that this model is a good model to predict house prices in Ames.

```
#This code was used to search for the categorical indicator that was not in  
our training model so that we could change the houses' values for that  
categorical indicator to its nearest categorical indicator.  
TestCategories = lm(sqrt(Price) ~ factor(Quality) + I(GroundSF^2) + YearBuilt  
+ I(BasementSF^2) +  
  factor(Condition) + LotArea + BasementFBath + GarageSF +  
  Fireplaces + YearRemodel + ScreenPorchSF + EnclosedPorchSF + LotFrontage  
+ factor(Condition)*YearRemodel + GroundSF*BasementSF, data =  
Adjusted_TestData)  
summary(TestCategories)  
  
##  
## Call:  
## lm(formula = sqrt(Price) ~ factor(Quality) + I(GroundSF^2) +  
##   YearBuilt + I(BasementSF^2) + factor(Condition) + LotArea +  
##   BasementFBath + GarageSF + Fireplaces + YearRemodel + ScreenPorchSF +  
##   EnclosedPorchSF + LotFrontage + factor(Condition) * YearRemodel +  
##   GroundSF * BasementSF, data = Adjusted_TestData)  
##
```

```

## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.35541 -0.39367  0.00001  0.35656  1.80715
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -7.343e+02  5.440e+02  -1.350  0.17897
## factor(Quality)2    5.190e+00  9.808e-01   5.291 3.87e-07 ***
## factor(Quality)3    6.673e+00  1.032e+00   6.467 1.11e-09 ***
## factor(Quality)4    5.755e+00  7.695e-01   7.478 4.40e-12 ***
## factor(Quality)5    6.204e+00  7.801e-01   7.953 2.87e-13 ***
## factor(Quality)6    6.550e+00  7.944e-01   8.245 5.21e-14 ***
## factor(Quality)7    6.805e+00  8.251e-01   8.248 5.12e-14 ***
## factor(Quality)8    8.062e+00  8.377e-01   9.625 < 2e-16 ***
## factor(Quality)9    8.981e+00  8.667e-01  10.363 < 2e-16 ***
## factor(Quality)10   1.048e+01  1.003e+00  10.449 < 2e-16 ***
## I(GroundSF^2)      1.531e-07  2.266e-07   0.676 0.50013
## YearBuilt          1.091e-02  4.279e-03   2.550 0.01171 *
## I(BasementSF^2)    1.205e-07  2.372e-07   0.508 0.61209
## factor(Condition)3  4.905e+02  5.507e+02   0.891 0.37438
## factor(Condition)4  7.819e+02  5.483e+02   1.426 0.15577
## factor(Condition)5  6.778e+02  5.453e+02   1.243 0.21572
## factor(Condition)6  7.009e+02  5.456e+02   1.285 0.20076
## factor(Condition)7  7.168e+02  5.468e+02   1.311 0.19175
## factor(Condition)8  7.095e+02  5.446e+02   1.303 0.19444
## factor(Condition)9 -5.524e-02  1.579e+00  -0.035 0.97214
## LotArea           8.140e-05  1.770e-05   4.598 8.50e-06 ***
## BasementFBath      1.826e-01  1.166e-01   1.566 0.11923
## GarageSF           1.001e-03  3.217e-04   3.111 0.00220 **
## Fireplaces         2.720e-01  9.767e-02   2.785 0.00599 **
## YearRemodel        3.563e-01  2.717e-01   1.312 0.19152
## ScreenPorchSF      4.356e-04  1.026e-03   0.425 0.67169
## EnclosedPorchSF    -3.630e-04  9.933e-04  -0.365 0.71524
## LotFrontage        2.283e-03  1.770e-03   1.290 0.19891
## GroundSF           1.076e-03  6.439e-04   1.671 0.09667 .
## BasementSF         6.992e-04  6.374e-04   1.097 0.27430
## factor(Condition)3:YearRemodel -2.430e-01  2.741e-01  -0.887 0.37656
## factor(Condition)4:YearRemodel -3.911e-01  2.728e-01  -1.433 0.15365
## factor(Condition)5:YearRemodel -3.378e-01  2.713e-01  -1.245 0.21490
## factor(Condition)6:YearRemodel -3.494e-01  2.715e-01  -1.287 0.19984
## factor(Condition)7:YearRemodel -3.572e-01  2.721e-01  -1.313 0.19112
## factor(Condition)8:YearRemodel -3.538e-01  2.710e-01  -1.306 0.19354
## factor(Condition)9:YearRemodel      NA         NA         NA         NA
## GroundSF:BasementSF 1.115e-07  3.865e-07   0.289 0.77327
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6777 on 163 degrees of freedom
## Multiple R-squared:  0.9493, Adjusted R-squared:  0.9381
## F-statistic: 84.7 on 36 and 163 DF, p-value: < 2.2e-16

```

```
Adjusted_TestData$Quality[Adjusted_TestData$Quality == 1] <- 2
NewHouseCV = predict(Transformed_Model3, newdata=Adjusted_TestData)

## Warning in predict.lm(Transformed_Model3, newdata = Adjusted_TestData):
## prediction from a rank-deficient fit may be misleading

residuals2 = sqrt(AmesTest2$Price) - NewHouseCV
mean(residuals2)

## [1] 0.05376515

summary(Transformed_Model3)$sigma

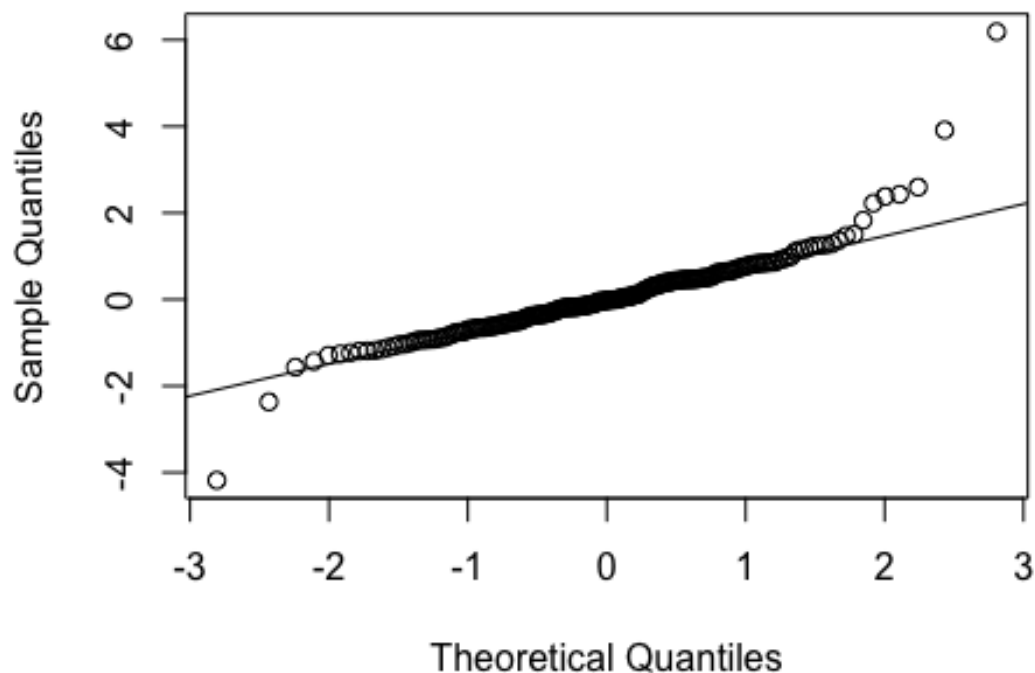
## [1] 0.7591932

sd(residuals2)

## [1] 0.9657668

qqnorm(residuals2)
qqline(residuals2)
```

Normal Q-Q Plot



```
cor(sqrt(AmesTest2$Price), NewHouseCV)^2

## [1] 0.8760979
```



```
summary(Transformed_Model13)$r.squared

## [1] 0.9202156

shrinkage = summary(Transformed_Model13)$r.squared -
cor(sqrt(AmesTest2$Price), NewHouseCV)^2
shrinkage

## [1] 0.04411776
```

Note on missing categories:

When creating the predictions using `predict(yourmodel,AmesTest)` you may see an error like:

```
Error in model.frame.default(Terms, newdata, na.action = na.action, xlev =
object$xlevels) : factor HouseStyle has new levels 1.5Unf, 2.5Fin, 2.5Unf
```

This occurs because the holdout sample has a value for the categorical variable that was not present in your training sample, so there is no indicator in your model to handle that case. To get a prediction for that house, you'll need to switch the category to one that is in your training data. In the example above you might choose to replace the "2.5Fin" house style with "2Story". If you are not sure what category to use, try whatever R uses as the "left out" reference category. Be sure to record any changes like this that you make.

Part 9. Final Model

Again, you may choose to make some additional adjustments to your model after considering the final residual analysis. If you do so, please explain what (and why) you did and provide the `summary()` for your new final model.

Suppose that you are interested in a house in Ames, Iowa that has characteristics listed below and want to find a 95% prediction interval for the price of this house.

A 2 story 11 room home, built in 1983 and remodeled in 1999 on a 21540 sq. ft. lot with 400 feet of road frontage. Overall quality is good (7) and condition is average (5). The quality and condition of the exterior are both good (Gd) and it has a poured concrete foundation. There is an 757 sq. foot basement that has excellent height, but is completely unfinished and has no bath facilities. Heating comes from a gas air furnace that is in excellent condition and there is central air conditioning. The house has 2432 sq. ft. of living space above ground, 1485 on the first floor and 947 on the second, with 4 bedrooms, 2 full and one half baths, and 1 fireplace. The 2 car, built-in garage has 588 sq. ft. of space and is average (TA) for both quality and construction. The only porches or decks is a 384 sq. ft. open porch in the front.

We are 95% confident that a house in Ames, Iowa with these conditions is priced within \$207,372 to \$314,862.

```
conditions=data.frame(YearBuilt=1983, YearRemodel=1999, LotArea=21540,
LotFrontage=400, Quality=7, Condition=5, BasementSF=757, BasementFBath=0,
```

```
GroundSF=2432, Fireplaces=1, GarageSF=588, ScreenPorchSF=0,
EnclosedPorchSF=0)
predict.lm(Transformed_Model3, conditions, interval="prediction")

## Warning in predict.lm(Transformed_Model3, conditions, interval =
"prediction"):
## prediction from a rank-deficient fit may be misleading

##          fit          lwr          upr
## 1 16.07238 14.40042 17.74434

14.40042^2

## [1] 207.3721

17.74434^2

## [1] 314.8616
```