# Prediction

## Summary


- The final model is $price^{1/3} \sim saledate + gba + grade + ayb + tot\_bathrm + fireplaces + extwall + age + yr\_rmdl + eyb + rooms + stories + kitchens$ where $tot\_bathrm$ is sum of $bathrm$ and 0.5*$hf\_bathrm$ and $age$ is $saledate$ minus $eyb$, $yr\_rmdl$, or $ayb$, whichever is the latest.


### Preprocessing

I changed some categorical variables with no order, that is heat, ac, stories, style, extwall, from type chr to type factor. For saledate, I kept only the year and converted the data to int. Since grade is ordinal, I associated the levels with numbers, with larger numbers representing better grades.

#### Missing Data

- yr_rmdl: I replaced the missing data (NA) in yr_rmdl with eyb, which has no NA values.
- stories: I replaced missing data (NA) with the number 2, matching what is indicated in style.

#### Transformation

- price: response variate price raised to the power of 1/3

#### Added variables

- age: saledate minus eyb, yr_rmdl, or ayb, whichever is the latest
- tot_bathrm: bathrm + 0.5*hl_bathrm
- rmdl: binary variable, 1 is the house was remodelled(if yr_rmdl is not NA), 0 if it wasn't

### Model Building

Stepwise regression with a $AIC$-value of 29823.78.

Main function used: `step` function

## 1. Preprocessing

### 1.1 Loading data

```
load("final.Rdata")
```

## 1.2 Dealing with NA values

### yr_rmdl

```
NA1 <- which(colSums(is.na(dtrain)) > 0)
sort(colSums(sapply(dtrain[NA1], is.na)), decreasing = TRUE)
```

```
## yr_rmdl stories
##     578       2
```

```
NA2 = which(colSums(is.na(dtest)) > 0)
sort(colSums(sapply(dtest[NA2], is.na)), decreasing = TRUE)
```

```
## yr_rmdl stories
##     587       2
```

```
nrow(subset(dtrain, eyb>yr_rmdl))
```

```
## [1] 10
```

```
nrow(subset(dtrain, eyb<yr_rmdl))
```

```
## [1] 712
```

```
nrow(subset(dtrain, eyb==yr_rmdl))
```

```
## [1] 3
```

```
nrow(subset(dtrain, is.na(yr_rmdl)))
```

```
## [1] 578
```

```
dtrain$rmdl = ifelse(is.na(dtrain$yr_rmdl), 0, 1)

for (i in 1:nrow(dtrain)){
  if(is.na(dtrain$yr_rmdl[i])){
    dtrain$yr_rmdl[i] = as.integer(dtrain$eyb[i])
  }
}
```

We see that both dtrain and dtest have NA values in yr_rmdl and stories. I investigated more and found that a large majorityof observations have eyb values that are lower than yr_rmdl, specifically 712 out of 1303. The second most common values for yr_rmdl is NA, which I interpret as no remodelling done. Thus, I chose to set all NA values in yr_rmdl to the last time an improvement was done to the house, which is eyb. Based on the data, we see that most houses that were sold with high prices were remodelled, so I created a new variable rmdl to indicate whether the house was remodelled before replacing the NA values.

### stories

```
dtrain[which(is.na(dtrain$stories)),]
```

```
##    bathrm hf_bathrm       heat ac rooms bedrm  ayb yr_rmdl  eyb stories
## 6       3         1 Forced Air  Y     7     4 2014    2015 2015      NA
## 51      3         1 Forced Air  Y     9     4 2015    2016 2016      NA
##                saledate  price  gba   style         grade      extwall kitchens
## 6  2014-09-11 00:00:00 766900 2816 2 Story  Good Quality Common Brick        1
## 51 2017-01-03 00:00:00 706900 2182 2 Story Above Average Brick/Siding        1
##    fireplaces landarea rmdl
## 6           1     5565    0
## 51          0     3018    0
```

```
dtest[which(is.na(dtest$stories)),]
```

```
##      Id bathrm hf_bathrm       heat ac rooms bedrm  ayb yr_rmdl  eyb stories
## 7     7      2         1 Forced Air  Y     8     4 1940      NA 1940      NA
## 155 155      3         1 Forced Air  Y     9     4 2015      NA 2016      NA
##                saledate  gba   style         grade      extwall kitchens
## 7   2018-04-04 00:00:00 2124 2 Story   Low Quality Brick/Stucco        1
## 155 2016-12-08 00:00:00 2182 2 Story Above Average Brick/Siding        1
##     fireplaces landarea
## 7            0     1062
## 155          0     3025
```

```
dtrain[which(is.na(dtrain$stories)), "stories"] = as.numeric(2)
```

Observe that all house with NA values in stories in both datasets have the style “2 story”, so I replaced the NA values with the number 2.

### 1.3 Label encoding/Factoring categorical variables

**heat, ac, style, extwall, grade, saledate**

```
dtrain$heat = as.factor(dtrain$heat)

dtrain$ac = ifelse(dtrain$ac == "Y", 1, 0)

dtrain$style = as.factor(dtrain$style)

Qualities = c('Low Quality', 'Fair Quality', 'Average', 'Above Average', 'Good Quality', 'Very Good', 'S

dtrain$grade = as.factor(dtrain$grade)

dtrain$grade = as.numeric(factor(dtrain$grade,levels=Qualities)) - 1

dtrain$extwall = as.factor(dtrain$extwall)

dtrain$saledate = as.integer(format(as.Date(dtrain$sale,format="%Y-%m-%d"),"%Y"))
```

I factorized heat, style, and extwall, and changed "Y" in to 1 and "N" to 0 in ac. Since grade is ordinal, I assigned numerical values to the levels of grade with the highest grade being matched with the largest number. I also extracted the year in saledate and changed the type to number in preparation for the calculation of age later on.

## 1.4 New variables

**age, tot_bathrm**

```
age = c()

for (i in 1:nrow(dtrain)){
  if (dtrain$yr_rmdl[i]>=dtrain$eyb[i]){
    if(dtrain$saledate[i] >= dtrain$yr_rmdl[i]){
      age = c(age,dtrain$saledate[i]-dtrain$yr_rmdl[i])
    } else if (dtrain$saledate[i]>=dtrain$eyb[i]){
      age = c(age,dtrain$saledate[i]-dtrain$eyb[i])
    } else {
      age = c(age,dtrain$saledate[i]-dtrain$ayb[i])
    }
  } else {
    if(dtrain$saledate[i] >= dtrain$eyb[i]){
      age = c(age,dtrain$saledate[i]-dtrain$eyb[i])
    } else if (dtrain$saledate[i]>=dtrain$yr_rmdl[i]){
      age = c(age,dtrain$saledate[i]-dtrain$yr_rmdl[i])
    } else {
      age = c(age,dtrain$saledate[i]-dtrain$ayb[i])
    }
  }
}

dtrain$age = age
dtrain$tot_bathrm = dtrain$bathrm + (dtrain$hf_bathrm*0.5)
```

Since yr_rmdl, ayb, eyb, and saledate by themselves don't really mean much, I created a new variable age that calculates the age of the house by subtracting the largest value among yr_rmdl, ayb, and eyb from saledate. Similarly, I added another variable tot_bathrm, which combines bathrm and hf_bathrm, with hf_bathrm subjecting to a factor of 0.5 as it is not the same as a full bathrm.

## 1.5 Removing levels with few or no observations in train or test

**extwall**

```
tapply(dtrain$extwall,dtrain$extwall,length)
```

```
##          Adobe       Aluminum   Brick Veneer   Brick/Siding    Brick/Stone
##              1             72             14             89              5
##   Brick/Stucco   Common Brick       Concrete Concrete Block     Face Brick
##             10            474              4              1              4
```

```
##       Hardboard   Metal Siding        Shingle          Stone   Stone Veneer
##              11              3             70              6              5
##   Stone/Siding   Stone/Stucco         Stucco   Stucco Block   Vinyl Siding
##              16              2             77              2            352
##    Wood Siding
##              85
```

```
tapply(dtest$extwall,dtest$extwall,length)
```

```
##      Aluminum Brick Veneer Brick/Siding  Brick/Stone Brick/Stucco Common Brick
##           83            8           95            4            8          532
##     Concrete   Face Brick    Hardboard Metal Siding      Shingle        Stone
##            2            2            9            2           49            4
## Stone Veneer Stone/Siding Stone/Stucco       Stucco Stucco Block Vinyl Siding
##            4            7            1           65            1          305
##  Wood Siding
##           96
```

```
dtrain = dtrain[!(dtrain$extwall=="Adobe"),]
dtrain = dtrain[!(dtrain$extwall=="Stone/Stucco"),]
dtrain = dtrain[!(dtrain$extwall=="Stucco Block"),]
dtrain = dtrain[!(dtrain$extwall=="Concrete Block"),]
```

I removed levels of extwall with very few observations in both datasets since it will not help with the prediction.

### style

```
tapply(dtrain$style,dtrain$style,length)
```

```
##         1 Story   1.5 Story Fin 1.5 Story Unfin         2 Story   2.5 Story Fin
##             229             114               5             829              77
## 2.5 Story Unfin         3 Story         4 Story        Bi-Level         Default
##              21              13               1               1               1
##     Split Foyer     Split Level
##               3               3
```

```
tapply(dtest$style,dtest$style,length)
```

```
##         1 Story   1.5 Story Fin 1.5 Story Unfin         2 Story   2.5 Story Fin
##             242             120               5             784              69
## 2.5 Story Unfin         3 Story         Default     Split Foyer     Split Level
##              21              18               2               6              10
```

```
dtrain = dtrain[!(dtrain$style=="4 Story"),]
dtrain = dtrain[!(dtrain$style=="Bi-Level"),]
```

I removed levels of style with very few observations in both datasets since it will not help with the prediction.

**heat**

```
tapply(dtrain$heat,dtrain$heat,length)
```

```
##     Air Exchng  Elec Base Brd     Forced Air Gravity Furnac  Hot Water Rad
##              1              1            630              1            409
##        Ht Pump        No Data   Wall Furnace      Warm Cool Water Base Brd
##             26              1              1            224              1
```

```
tapply(dtest$heat,dtest$heat,length)
```

```
## Elec Base Brd    Forced Air Hot Water Rad       Ht Pump     Warm Cool
##             1           611           401            17           247
```

```
dtrain = dtrain[!(dtrain$heat=="Air Exchng"),]
dtrain = dtrain[!(dtrain$heat=="Gravity Furnac"),]
dtrain = dtrain[!(dtrain$heat=="Wall Furnace"),]
dtrain = dtrain[!(dtrain$heat=="Water Base Brd"),]
dtrain[dtrain$heat == "No Data",]
```

```
##     bathrm hf_bathrm    heat ac rooms bedrm  ayb yr_rmdl  eyb stories saledate
## 966      0         0 No Data  0     0     0 1941    1928 1928       1     2006
##      price gba   style grade      extwall kitchens fireplaces landarea rmdl age
## 966 150300 640 1 Story     0 Common Brick        0          0     3011    0  78
##     tot_bathrm
## 966          0
```

```
dtrain = dtrain[!(dtrain$heat=="No Data"),]
```

I removed levels of heat with very few observations in both datasets. I also noticed one observation with missing data in many columns, and thus I removed it.

# 2. Visualization of important variables

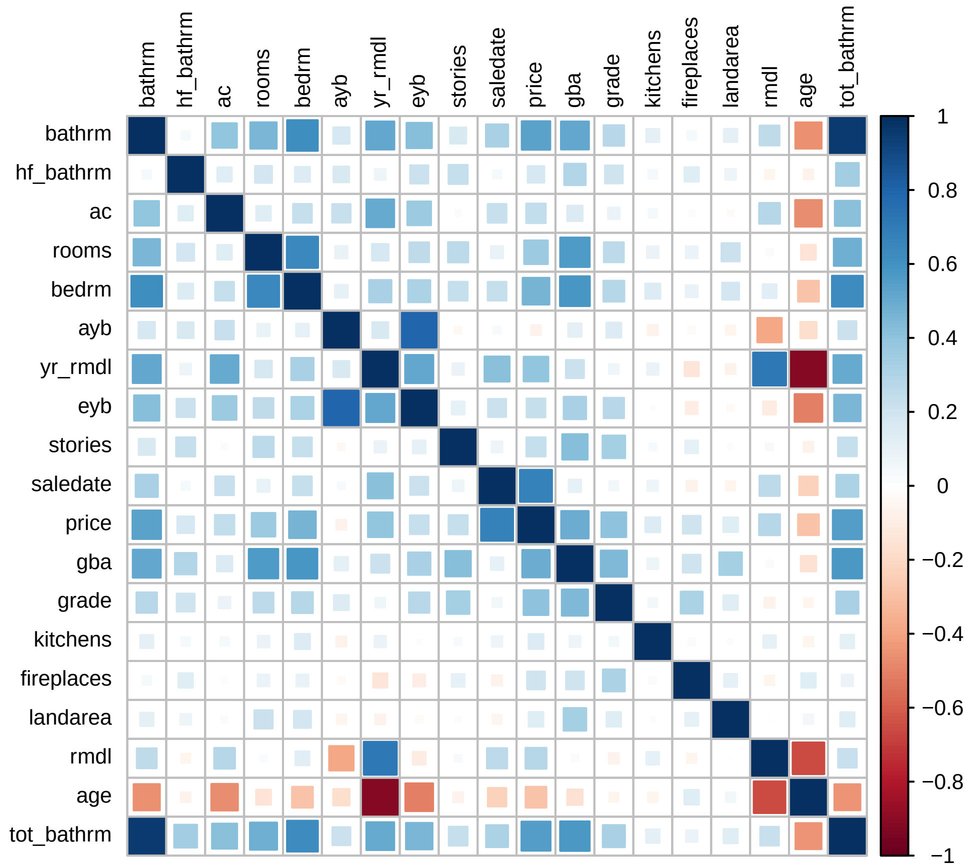
## Correlations

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.0.2
```

```
## corrplot 0.84 loaded
```

```
numeric_vars = which(sapply(dtrain, is.numeric))

dtrain_numvar = dtrain[, numeric_vars]
cor_numvar = cor(dtrain_numvar, use="pairwise.complete.obs")

corrplot(cor_numvar, method="square",tl.col="black", tl.pos = "lt",tl.cex = 0.7,cl.cex = .7)
```




As expected, there is strong negative correlation between age and price. bathrm, rooms, bedrooms, yr_rmdl, saledate, gba, grade, and tot_bathrm have strong positive correlations with price. We will use this knowledge to inspect our model later.

# 3. Model Specification

## 3.1 Automated method

```
null = lm(price~1, data=dtrain)
fullmodel = lm(price~., data=dtrain)

step(null,scope = list(upper=fullmodel),direction="both",trace=0)
```

```
##
## Call:
## lm(formula = price ~ saledate + gba + grade + ayb + tot_bathrm +
##     fireplaces + extwall + age + yr_rmdl + eyb + rooms + stories +
##     kitchens, data = dtrain)
##
## Coefficients:
##         (Intercept)             saledate                   gba
##          -3.321e+07            1.815e+04             8.515e+01
##               grade                  ayb            tot_bathrm
```

```
##           5.800e+04          -2.038e+03           3.160e+04
##          fireplaces  extwallBrick Veneer  extwallBrick/Siding
##           4.372e+04          -4.593e+03          -6.041e+04
##  extwallBrick/Stone  extwallBrick/Stucco  extwallCommon Brick
##          -7.921e+04          -6.413e+04          -1.570e+04
##     extwallConcrete    extwallFace Brick     extwallHardboard
##          -9.581e+04          -2.775e+03           1.003e+05
## extwallMetal Siding       extwallShingle         extwallStone
##           1.124e+05          -1.427e+04           1.183e+05
## extwallStone Veneer  extwallStone/Siding       extwallStucco
##          -6.027e+03          -3.313e+04          -1.904e+04
## extwallVinyl Siding   extwallWood Siding                 age
##          -4.131e+04           1.703e+04          -1.839e+03
##             yr_rmdl                  eyb               rooms
##          -1.390e+03           1.757e+03           7.827e+03
##             stories             kitchens
##          -1.365e+04           4.071e+04
```

```r
# Step:  AIC=29823.78
model = lm(price ~ saledate + gba + grade + ayb + tot_bathrm + fireplaces + extwall + age + yr_rmdl + ey

# step(fullmodel, scope = list(lower=null),direction="backward",trace=0)
# Step:  AIC=29825.44
# model = price ~ bathrm + hf_bathrm + rooms + ayb + yr_rmdl + eyb + stories + saledate + gba + grade +
```

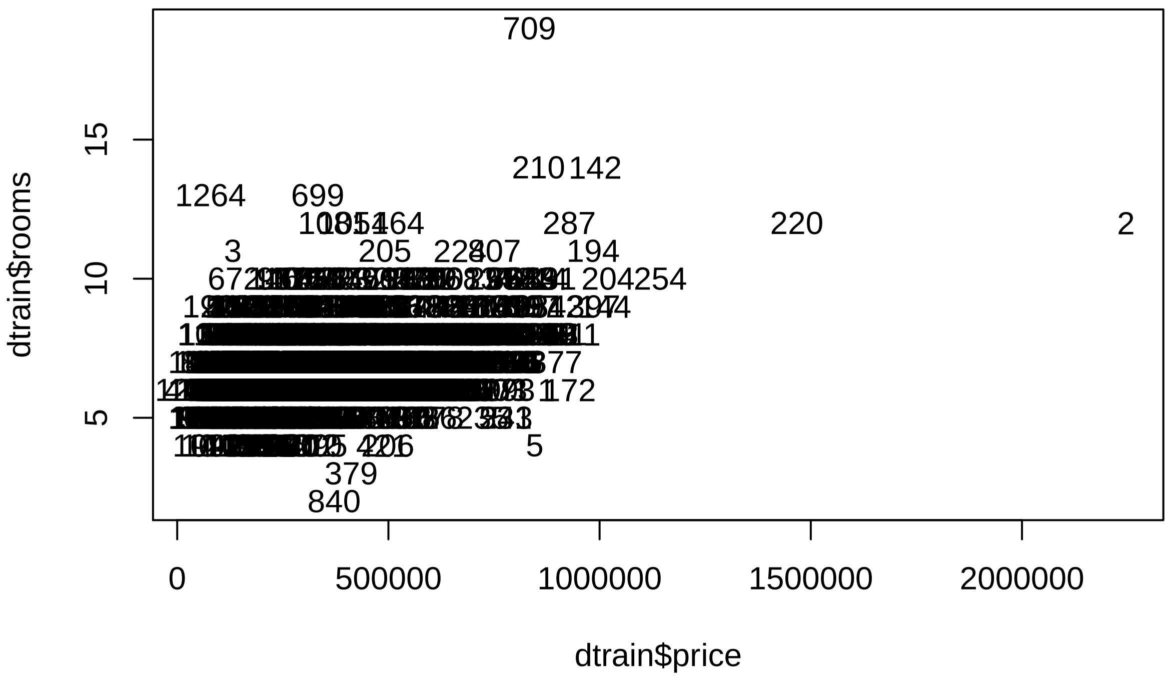
Stepwise regression and forward selection gave the same models and AIC values. However, backward selection gave a different model with a higher AIC, thus I will choose the first model.

The AIC for the exhaustive model generated using regsubsets function in the package leaps is much larger than the others, so I will keep the model obtained from stepwise regression.

# 4. Outliers

## 4.1 rooms

```r
plot(dtrain$price, dtrain$rooms, type="n")
text(dtrain$price, dtrain$rooms)
```




```r
tapply(dtrain$rooms,dtrain$rooms,length)
```

```
##   2   3   4   5   6   7   8   9  10  11  12  13  14  19
##   1   1  25 118 458 326 229  79  37   5   6   2   2   1
```

```r
tapply(dtest$rooms,dtest$rooms,length)
```

```
##   2   3   4   5   6   7   8   9  10  11  12  13
##   1   2  29  97 452 317 227  92  43   9   7   1
```

```r
dtrain = dtrain[!(dtrain$rooms=="19"),]
dtrain = dtrain[!(dtrain$rooms=="14"),]
```

We see that there are very few houses with 14 or 19 rooms, and the max number of rooms among the data in dtest is 13, so we will remove the 3 rows with the most rooms.

### 4.2 price

```r
sort(dtrain$price, decreasing=T)[1]
```

```
## [1] 2246100
```

```
sort(dtrain$price, decreasing=T)[2]
```

```
## [1] 1466800
```

```
sort(dtrain$price, decreasing=T)[3]
```

```
## [1] 1143800
```

```
sort(dtrain$price, decreasing=T)[4]
```

```
## [1] 1019800
```

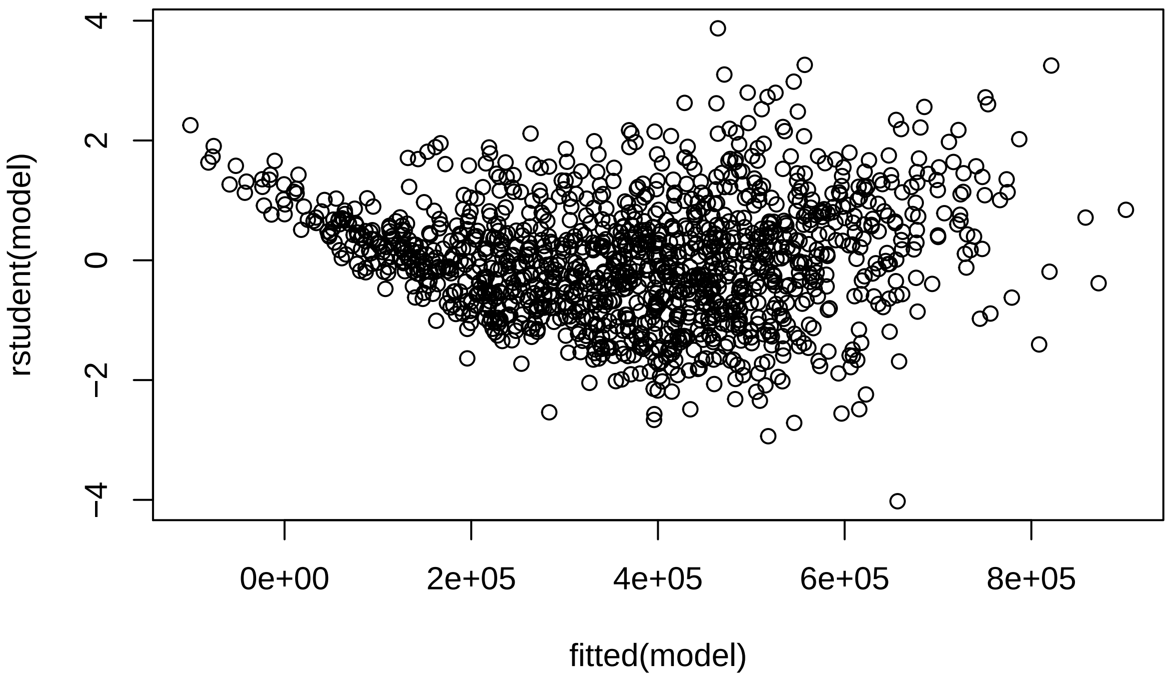
```
dtrain = dtrain[!(dtrain$price == "2246100"),]
dtrain = dtrain[!(dtrain$price == "1466800"),]

model = lm(price ~ saledate + gba + grade + ayb + tot_bathrm + fireplaces + extwall + age + yr_rmdl + ey
```

We see that the differences between 1st, 2nd largest price and the others are very big, so the observations with the 2 largest prices are removed.

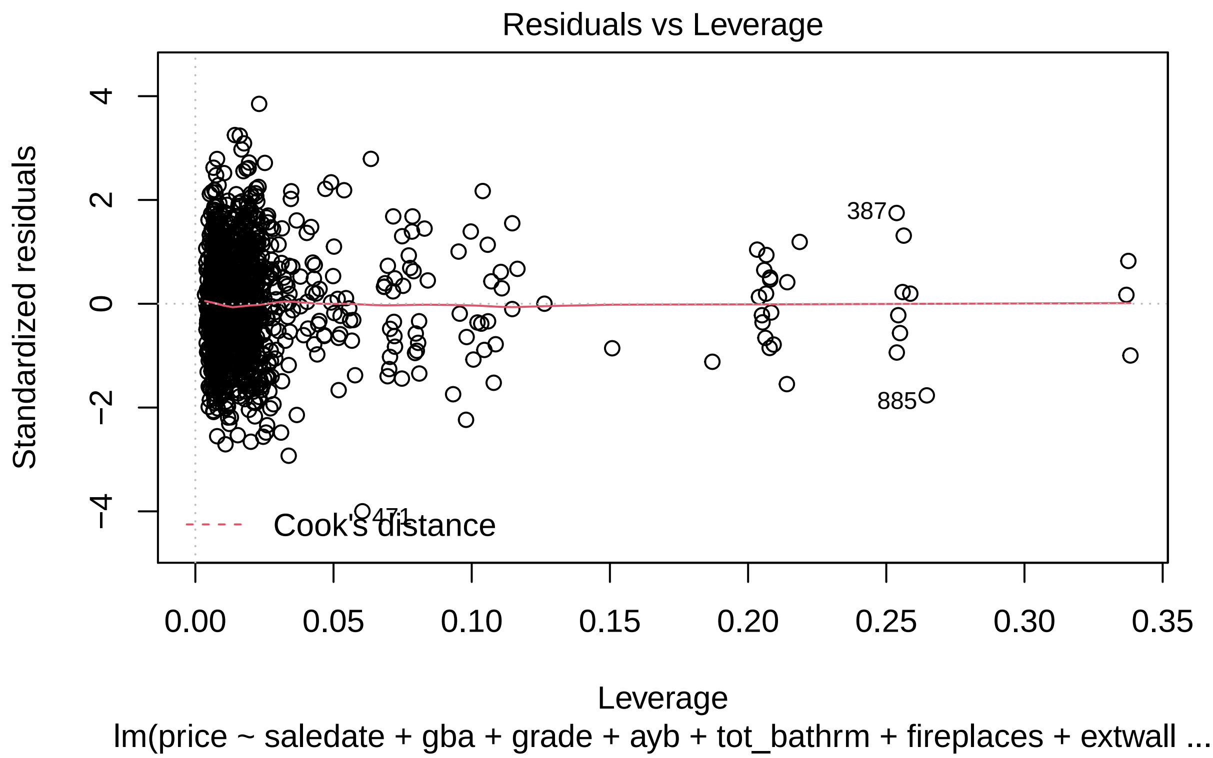
# 5. Assumptions for Linear Regression Model

## E(ei) = 0, Normality, constant variance

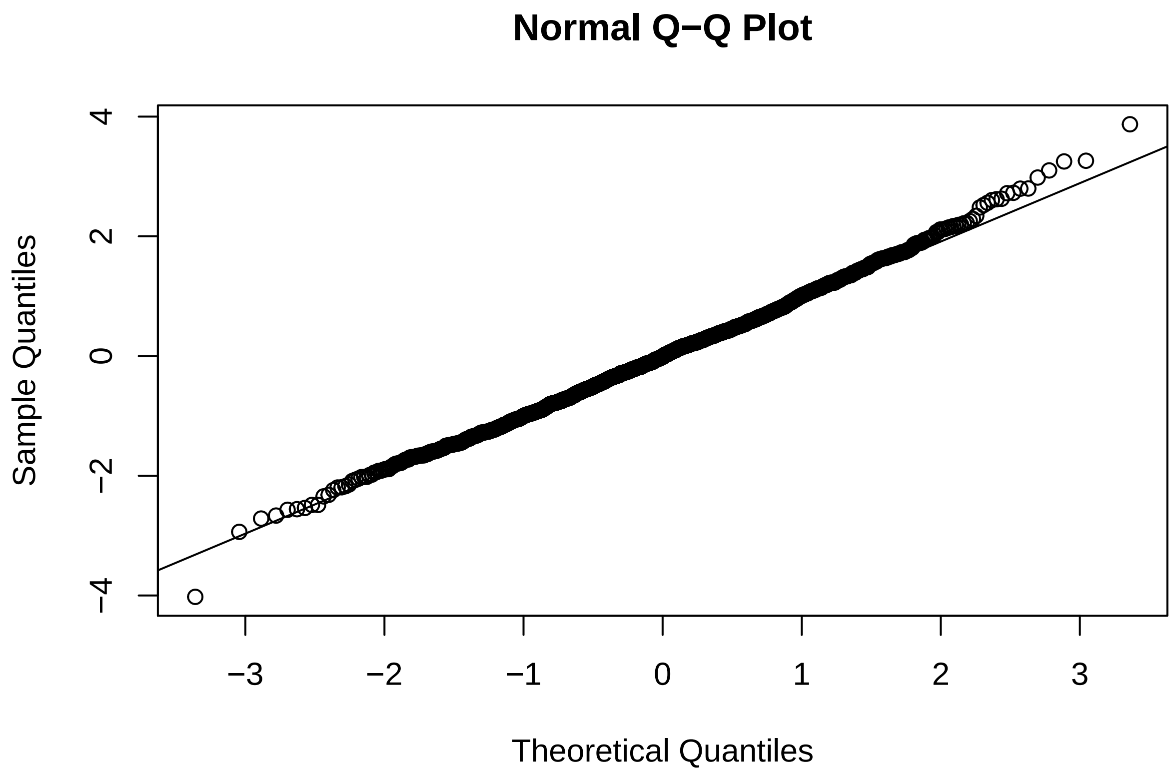
```
# residual vs. fitted
plot(fitted(model),rstudent(model))
```




```r
# Cook's distance
plot(model,which=5)
```




```
# QQ-plot
qqnorm(rstudent(model))
qqline(rstudent(model))
```




di vs $\hat{y}$ shows a pattern, so I will investigate later. All points have a cook's distance below 1. The residuals seem to follow standard normal.

## 6. Influential points

### Checking hii and |di|

```
hatm = hatvalues(model)
hatv = as.data.frame(hatm)
mean = 2*(19 + 1)/1303
hatv$warn = ifelse(hatv[,'hatm']>mean, '>', '-')
bighatv = subset(hatv, warn==">")

resm = rstudent(model)
resv = as.data.frame(resm)
cutoff = 2.5
resv$warn = ifelse(abs(resv[,'resm'])>cutoff, '>','-')
bigresv = subset(resv, warn==">")

bighatv
```

```
##              hatm warn
## 1    0.05382465    >
```

```
## 21   0.03470411     >
## 32   0.04246609     >
## 34   0.07976037     >
## 94   0.03102269     >
## 97   0.04290233     >
## 105  0.10866750     >
## 112  0.07160293     >
## 121  0.04411490     >
## 125  0.03132922     >
## 129  0.03420756     >
## 131  0.20835402     >
## 137  0.03380760     >
## 139  0.20794526     >
## 141  0.07473291     >
## 146  0.03234252     >
## 147  0.03376873     >
## 151  0.33688134     >
## 153  0.33834170     >
## 165  0.04190187     >
## 166  0.10795812     >
## 173  0.03509054     >
## 174  0.03119938     >
## 175  0.04375883     >
## 179  0.33758693     >
## 187  0.04299786     >
## 191  0.15083116     >
## 192  0.20779865     >
## 194  0.03138311     >
## 198  0.11466646     >
## 201  0.09966318     >
## 217  0.09525037     >
## 220  0.03258782     >
## 234  0.10578762     >
## 237  0.04910356     >
## 244  0.07858293     >
## 251  0.05150187     >
## 267  0.20650244     >
## 285  0.20662091     >
## 290  0.10398854     >
## 291  0.03255435     >
## 296  0.05184996     >
## 298  0.07896363     >
## 303  0.11051779     >
## 304  0.21867193     >
## 319  0.10457300     >
## 324  0.08061613     >
## 325  0.08102991     >
## 326  0.08014497     >
## 327  0.07950043     >
## 339  0.04445946     >
## 387  0.25373621     >
## 400  0.06965781     >
## 401  0.03456555     >
## 430  0.09816188     >
```

```
## 437  0.06825511     >
## 442  0.04031055     >
## 462  0.04248612     >
## 471  0.06045377     >
## 490  0.25868575     >
## 495  0.05241238     >
## 507  0.03405587     >
## 512  0.04911249     >
## 516  0.04703072     >
## 517  0.08095931     >
## 536  0.06864655     >
## 549  0.05451650     >
## 557  0.03671423     >
## 565  0.21402303     >
## 572  0.04985080     >
## 576  0.07838629     >
## 581  0.07204358     >
## 599  0.20501014     >
## 601  0.11654137     >
## 617  0.04321552     >
## 618  0.20523043     >
## 622  0.03804846     >
## 633  0.05177819     >
## 634  0.10060655     >
## 643  0.07152956     >
## 651  0.10345135     >
## 657  0.03666831     >
## 667  0.05255144     >
## 674  0.07224041     >
## 704  0.03329721     >
## 706  0.03342821     >
## 707  0.20584728     >
## 711  0.06832194     >
## 715  0.09566416     >
## 723  0.25433944     >
## 737  0.07482872     >
## 741  0.25590522     >
## 749  0.10591836     >
## 755  0.07185652     >
## 756  0.10710130     >
## 761  0.20327922     >
## 780  0.06352257     >
## 782  0.05015690     >
## 783  0.07722945     >
## 793  0.08293552     >
## 795  0.04487373     >
## 799  0.08411384     >
## 803  0.05569522     >
## 814  0.05599027     >
## 815  0.03396299     >
## 820  0.21416767     >
## 838  0.09794387     >
## 841  0.03803395     >
## 868  0.05714365     >
```

```
## 885  0.26465159    >
## 889  0.03917478    >
## 909  0.04494128    >
## 912  0.10208777    >
## 913  0.03378360    >
## 943  0.12632589    >
## 945  0.11088444    >
## 983  0.25380054    >
## 985  0.04047824    >
## 1010 0.05779389    >
## 1015 0.11467924    >
## 1028 0.20626034    >
## 1039 0.05028054    >
## 1041 0.04075562    >
## 1054 0.09327124    >
## 1084 0.20924268    >
## 1093 0.03198152    >
## 1100 0.07515247    >
## 1110 0.25631773    >
## 1132 0.25496931    >
## 1144 0.03528195    >
## 1150 0.20798719    >
## 1174 0.07774831    >
## 1204 0.07220143    >
## 1205 0.07049038    >
## 1210 0.20393209    >
## 1231 0.07013626    >
## 1232 0.07043543    >
## 1238 0.06956001    >
## 1261 0.05664717    >
## 1277 0.04661951    >
## 1287 0.18707249    >
## 1302 0.04669386    >
```

```
bigresv
```

```
##           resm warn
## 3    -2.665212    >
## 5     3.872097    >
## 49    2.627458    >
## 99    2.796438    >
## 107   2.982201    >
## 144   2.727288    >
## 145   2.605115    >
## 147  -2.937206    >
## 189   2.522285    >
## 200   3.262722    >
## 206   2.720553    >
## 214   2.560326    >
## 238   2.620566    >
## 246   3.101041    >
## 257   3.250990    >
## 471  -4.022780    >
## 780   2.798880    >
```

```
## 970  -2.714866     >
## 1014 -2.558257     >
## 1223 -2.568516     >
## 1274 -2.537927     >
```

```
dtrain = dtrain[-c(147,471,780),]
dtrain = dtrain[!(dtrain$age < 0),]
```

Observations 147, 471, 780 have large hii and rstudent values, thus they are influential points. There are also a few observations with negative age, so I will remove those as well.

# 7. Transformation

## Boxcox

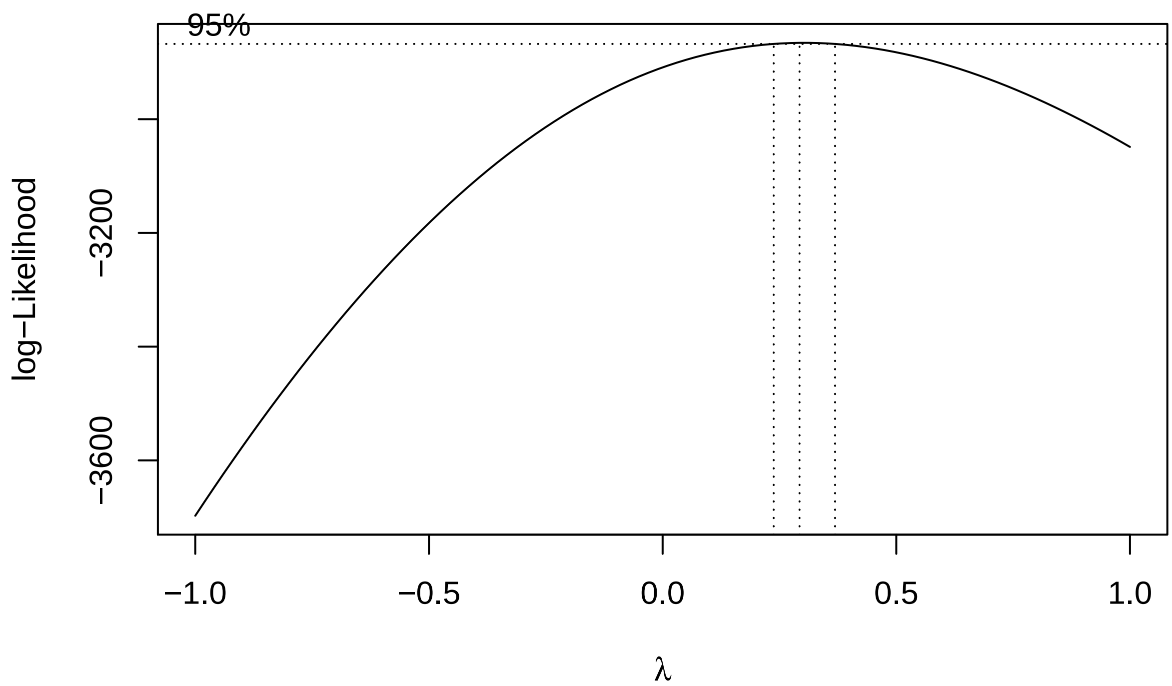
```
library(MASS)

model = lm(price ~ saledate + gba + grade + ayb + tot_bathrm + fireplaces + extwall + age + yr_rmdl + e
AIC(model)
```

```
## [1] 33080.57
```

```
summary(model)$adj.r.squared
```

```
## [1] 0.747082
```

```
boxcox(model,lambda = seq(-1,1,1/20))
```




```
model = lm((price^(1/3)) ~ saledate + gba + grade + ayb + tot_bathrm + fireplaces + extwall + age + yr_r
AIC(model)
```

```
## [1] 8281.993
```

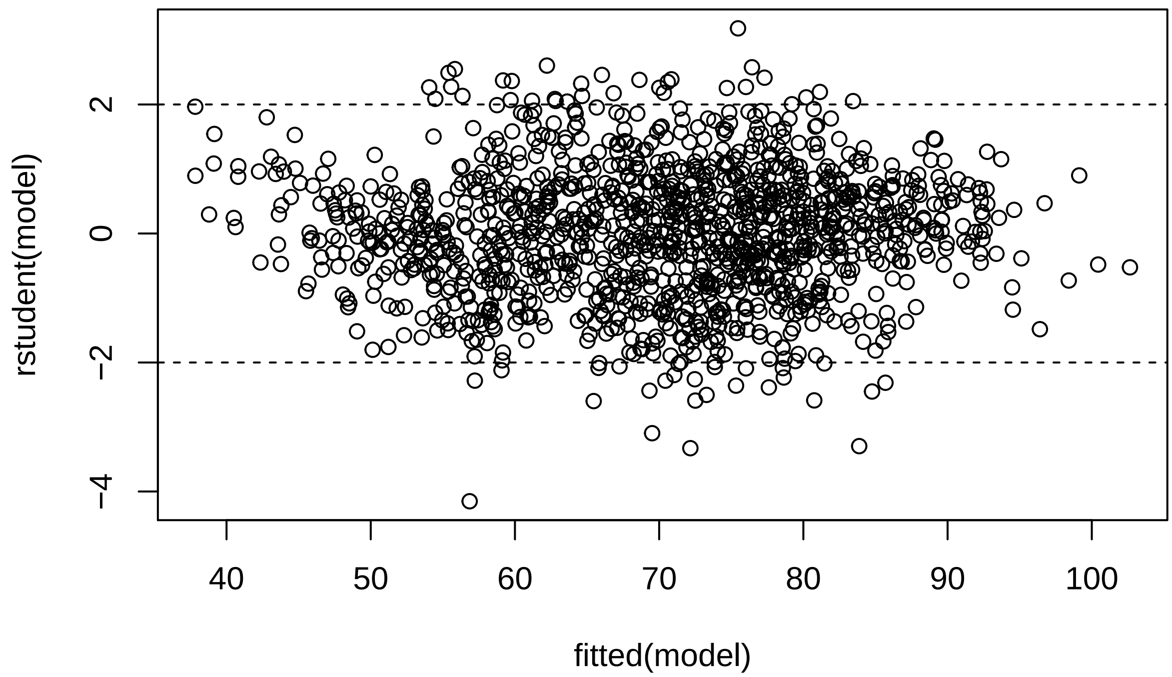
```
summary(model)$adj.r.squared
```

```
## [1] 0.7838498
```

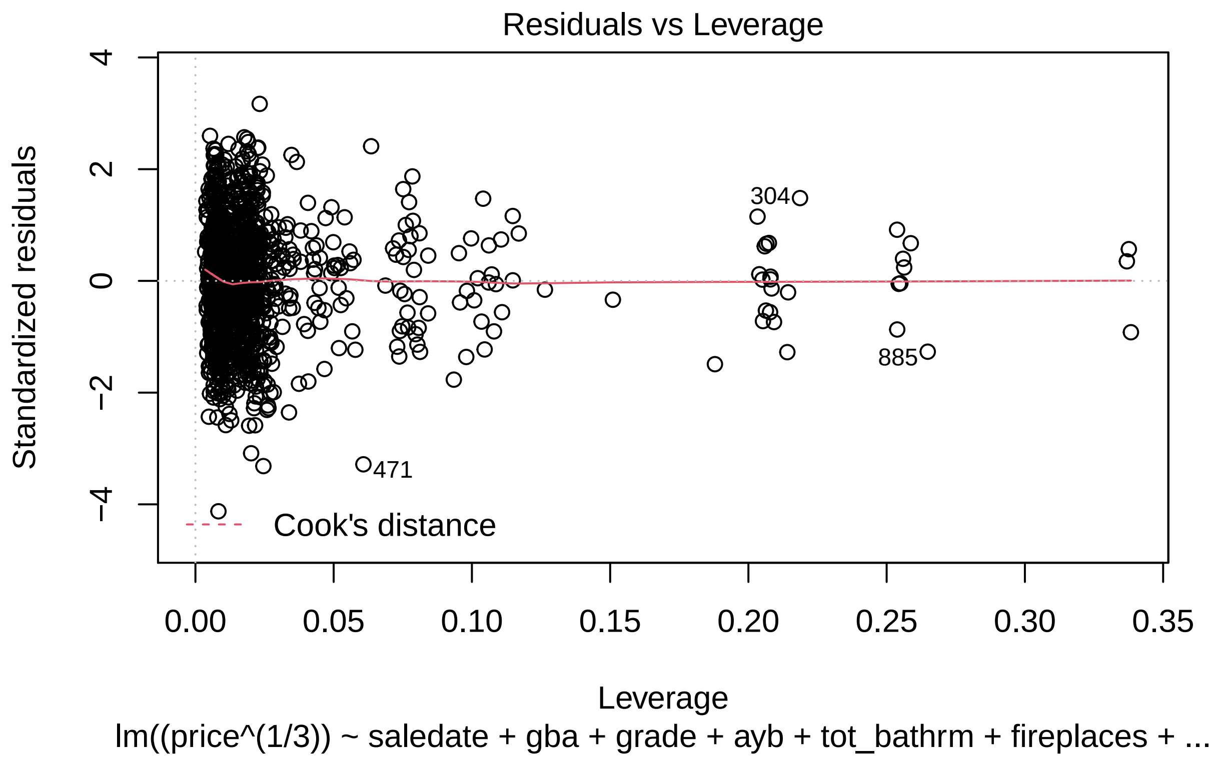
Since the 95% confidence interval of $\lambda$ does not contain "nice" or common values, I will use the MLE, which is 1/3. The new model gives a higher adjusted r-squared value.

### Back to checking the 3 assumptions

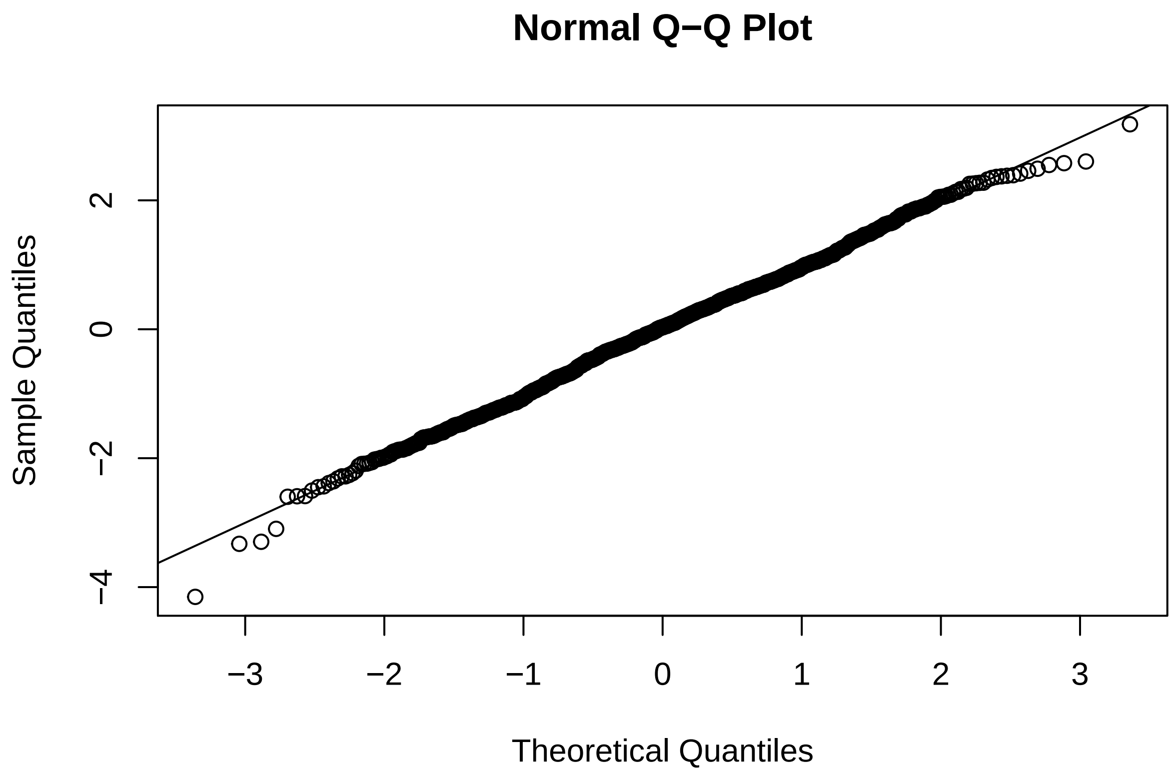
```
# residual vs. fitted
plot(fitted(model),rstudent(model))
abline(h=c(-2,2),lty=2)
```




```
# Cook's distance
plot(model,which=5)
```




```r
# QQ-plot
qqnorm(rstudent(model))
qqline(rstudent(model))
```




After the transformation, di vs $\hat{y}$ doesn't show an obvious pattern. All points have a cook's distance below 1. The residuals seem to follow standard normal. All assumptions are met.

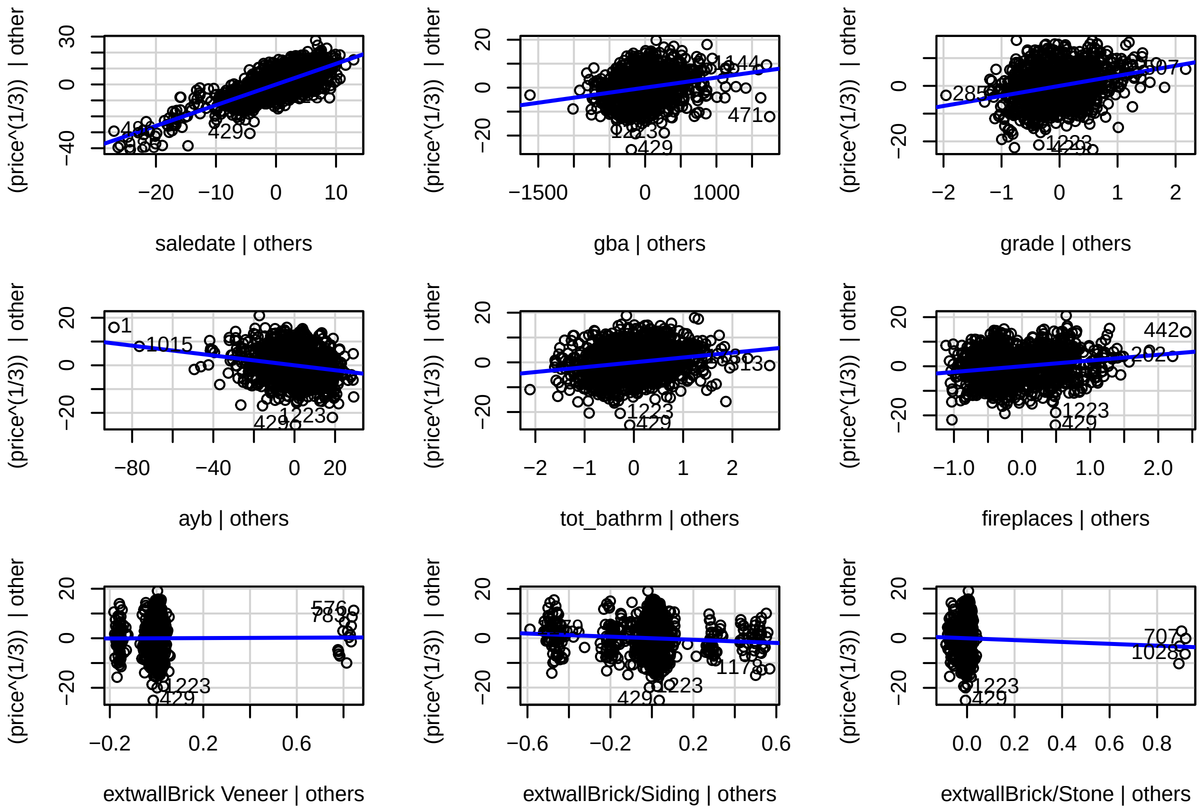
### avPlots

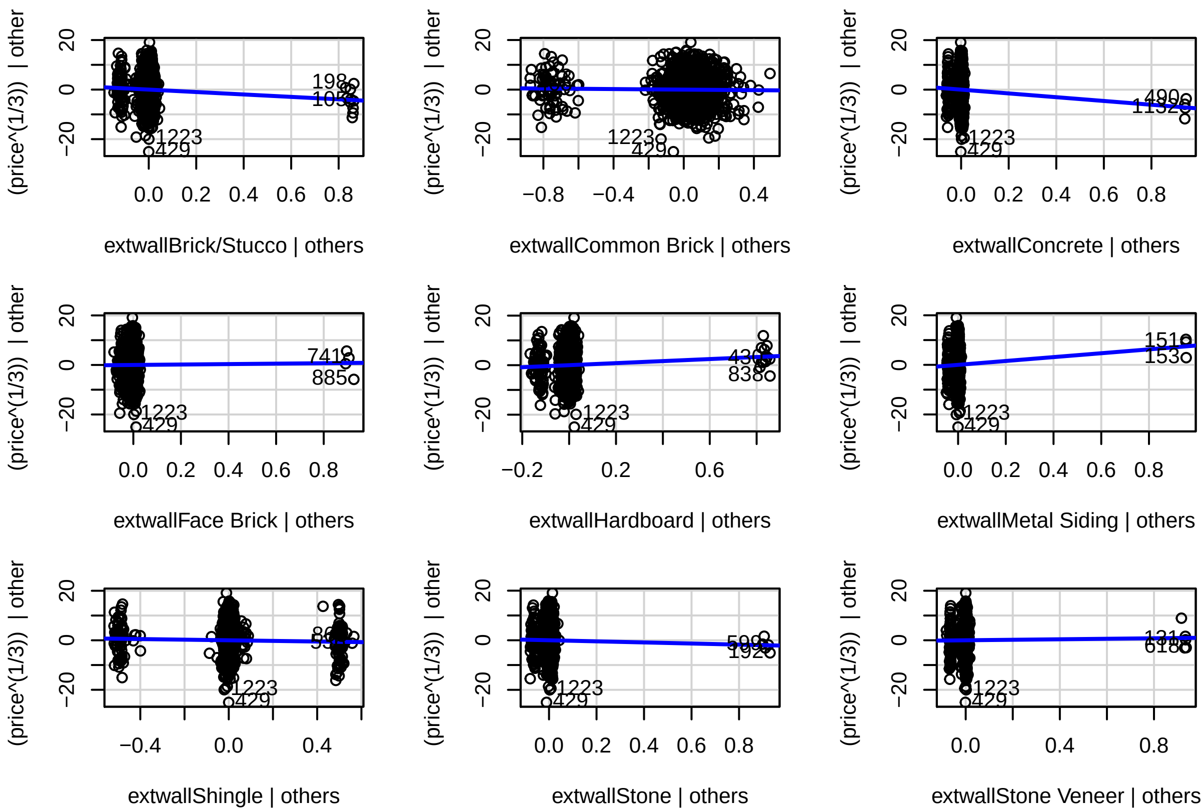
```
library(car)
```

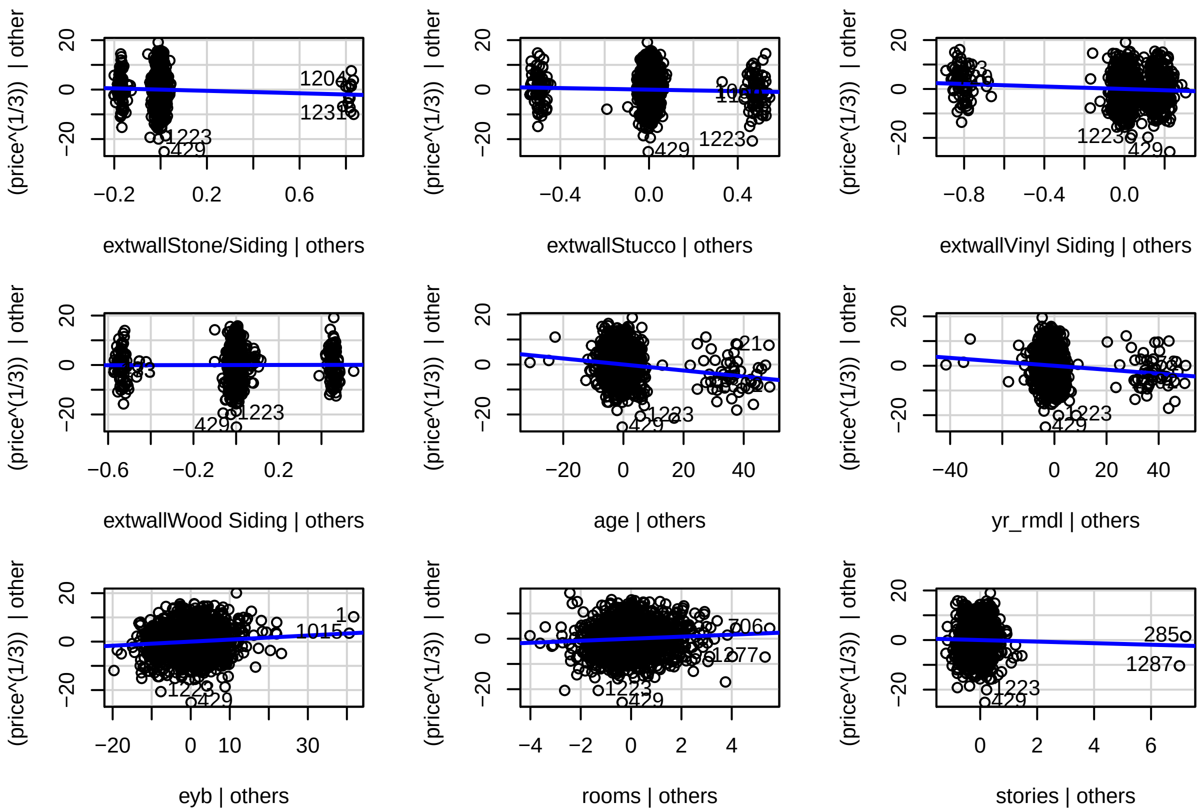
```
## Warning: package 'car' was built under R version 4.0.2
```

```
## Loading required package: carData
```

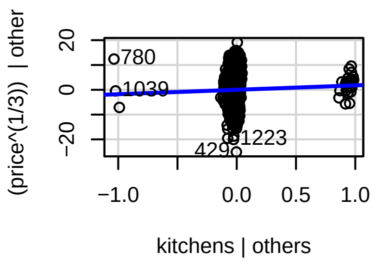
```
avPlots(model)
```


(price^(1/3)) | other
saledate | others
gba | others
grade | others
ayb | others
tot_bathrm | others
fireplaces | others
extwallBrick Veneer | others
extwallBrick/Siding | others
extwallBrick/Stone | others


(price^(1/3)) | other
extwallBrick/Stucco | others
extwallCommon Brick | others
extwallConcrete | others
extwallFace Brick | others
extwallHardboard | others
extwallMetal Siding | others
extwallShingle | others
extwallStone | others
extwallStone Veneer | others


extwallStone/Siding | others
extwallStucco | others
extwallVinyl Siding | others
extwallWood Siding | others
age | others
yr_rmdl | others
eyb | others
rooms | others
stories | others

## Added−Variable Plots




```r
model2 = lm((price^(1/3)) ~ saledate + gba + grade + ayb + tot_bathrm + fireplaces + extwall + age + yr_

summary(model)$adj.r.squared
```

```
## [1] 0.7838498
```

```r
summary(model2)$adj.r.squared
```

```
## [1] 0.7829209
```

```r
AIC(model)
```

```
## [1] 8281.993
```

```r
AIC(model2)
```

```
## [1] 8286.497
```

The plots suggested linear relationships in all plots, except rooms, which has a weak linear relationship with price, so let's try a model without rooms. That gives a higher AIC and lower adjusted r-squared, so I will retain the old model.

The final model is $price^{1/3} \sim saledate + gba + grade + ayb + tot\_bathrm + fireplaces + extwall + age + yr\_rmdl + eyb + rooms + stories + kitchens$.