multiple case 9.31

```r
setwd('/Users/quebec/Playground/ALSM/case')
pacman::p_boot()
pacman::p_load(MASS,MPV,leaps,corrplot,data.table,car,lmtest,alr3,ramify)
.dt<-fread('./data/APPENC07.txt')
rownames(.dt)<-sapply(.dt[,1],as.character)
names(.dt)<-c('id','price','feet','bedrooms','bathrooms','air','garage','pool','year','quality','s
.dt$id<-NULL
```

EDA

```r
colSums(is.na(.dt))
```

```
##     price      feet  bedrooms bathrooms       air    garage      pool      year 
##         0         0         0         0         0         0         0         0 
##   quality     style       lot   highway 
##         0         0         0         0 
```

没有缺失值.

```r
summary(.dt)
```

```
##      price             feet         bedrooms       bathrooms    
##  Min.   : 84000   Min.   : 980   Min.   :0.000   Min.   :0.000  
##  1st Qu.:180000   1st Qu.:1701   1st Qu.:3.000   1st Qu.:2.000  
##  Median :229900   Median :2061   Median :3.000   Median :3.000  
##  Mean   :277894   Mean   :2261   Mean   :3.471   Mean   :2.642  
##  3rd Qu.:335000   3rd Qu.:2636   3rd Qu.:4.000   3rd Qu.:3.000  
##  Max.   :920000   Max.   :5032   Max.   :7.000   Max.   :7.000  
##       air             garage         pool              year     
##  Min.   :0.0000   Min.   :0.0   Min.   :0.00000   Min.   :1885  
##  1st Qu.:1.0000   1st Qu.:2.0   1st Qu.:0.00000   1st Qu.:1956  
##  Median :1.0000   Median :2.0   Median :0.00000   Median :1966  
##  Mean   :0.8314   Mean   :2.1   Mean   :0.06897   Mean   :1967  
##  3rd Qu.:1.0000   3rd Qu.:2.0   3rd Qu.:0.00000   3rd Qu.:1981  
##  Max.   :1.0000   Max.   :7.0   Max.   :1.00000   Max.   :1998  
##      quality          style             lot           highway
##  Min.   :1.000   Min.   : 1.000   Min.   : 4560   Min.   :0.00000
##  1st Qu.:2.000   1st Qu.: 1.000   1st Qu.:17205   1st Qu.:0.00000
##  Median :2.000   Median : 2.000   Median :22200   Median :0.00000
##  Mean   :2.184   Mean   : 3.345   Mean   :24370   Mean   :0.02107
##  3rd Qu.:3.000   3rd Qu.: 7.000   3rd Qu.:26787   3rd Qu.:0.00000
##  Max.   :3.000   Max.   :11.000   Max.   :86830   Max.   :1.00000
```

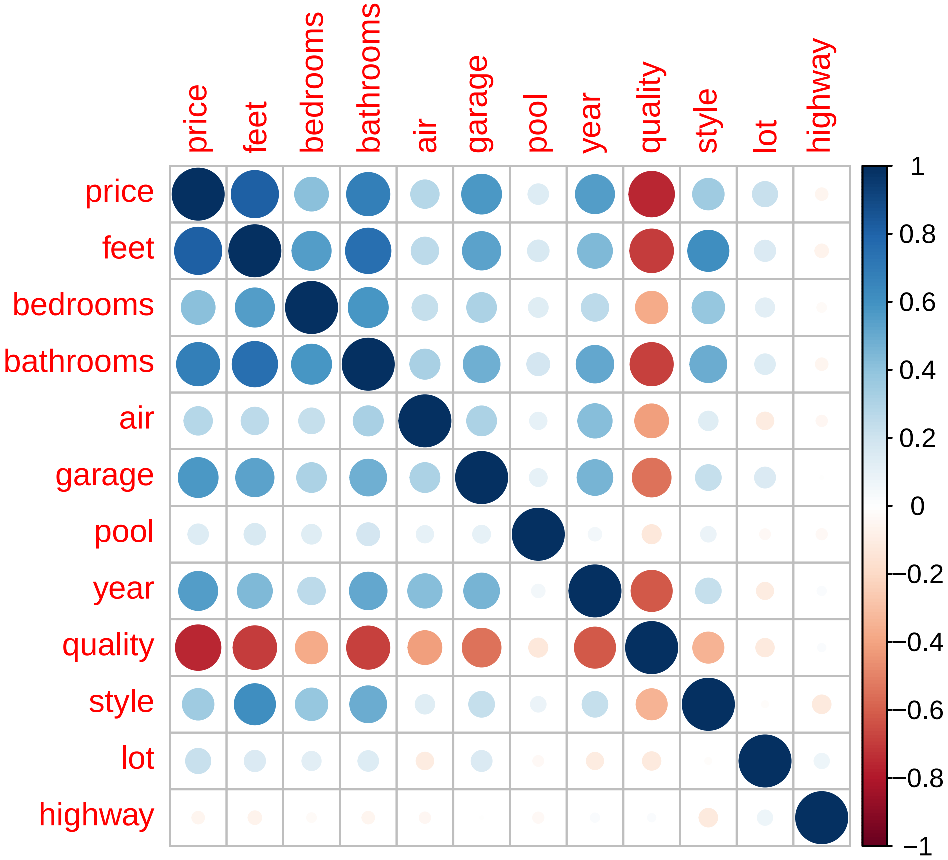
```
table(pool=.dt$pool,highway=.dt$highway)
```

```
##     highway
## pool   0   1
##     0 475  11
##     1  36   0
```

highway 和 pool 都算是少有的，特别是有 pool 的房子，只有 11 个.

相关系数图

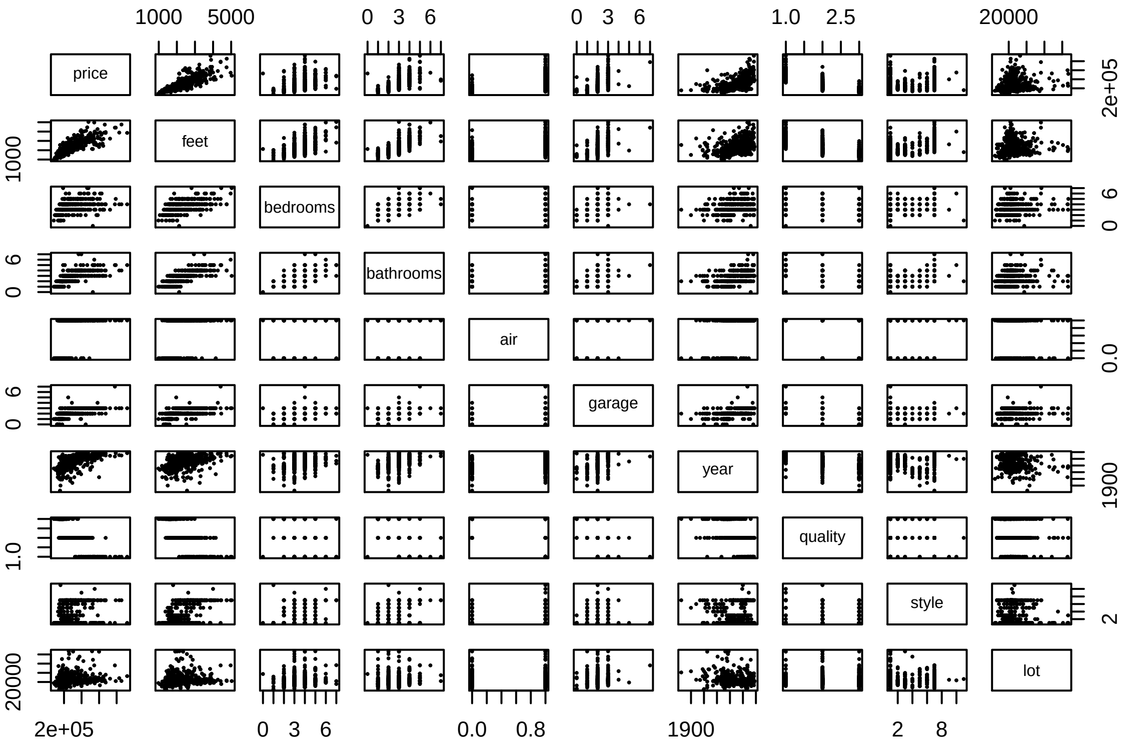
```
corrplot(cor(.dt),method="circle")
```



从这幅图来看，price 与 highway,pool 的线性关系很小，与 feet,bedrooms,quality,bathrooms 有很强的相关性. 多重共线性也是存在的，feet 与多个变量都有明显的线性关系，从直观上也容易理解.

对相关性明显的做一下 scatterplot，否则因为数据量太大，什么也看不清

```
.dt<-as.data.frame(.dt)
droplist<-c('pool','highway')
pairs(.dt[,!colnames(.dt) %in% droplist],cex=0.2)
```



模型构建

```
.dt$pool<-factor(.dt$pool)
.dt$highway<-factor(.dt$highway)
train_size=300
set.seed(17)
.dt<-.dt[sample(nrow(.dt)),] # 可能没有必要，但是还是置乱一下
.dt$class<-gl(2,k=train_size,l=nrow(.dt),labels=c('training','validation'))
```

full model

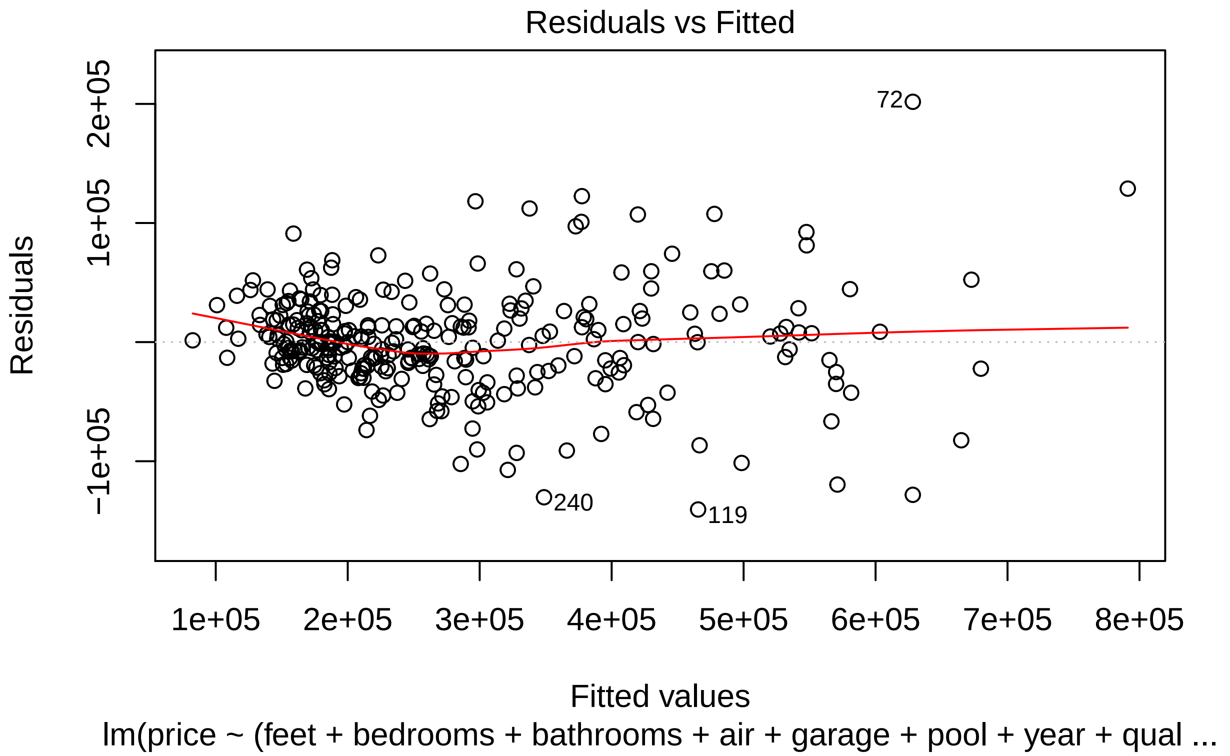
先看看 full model 的情况

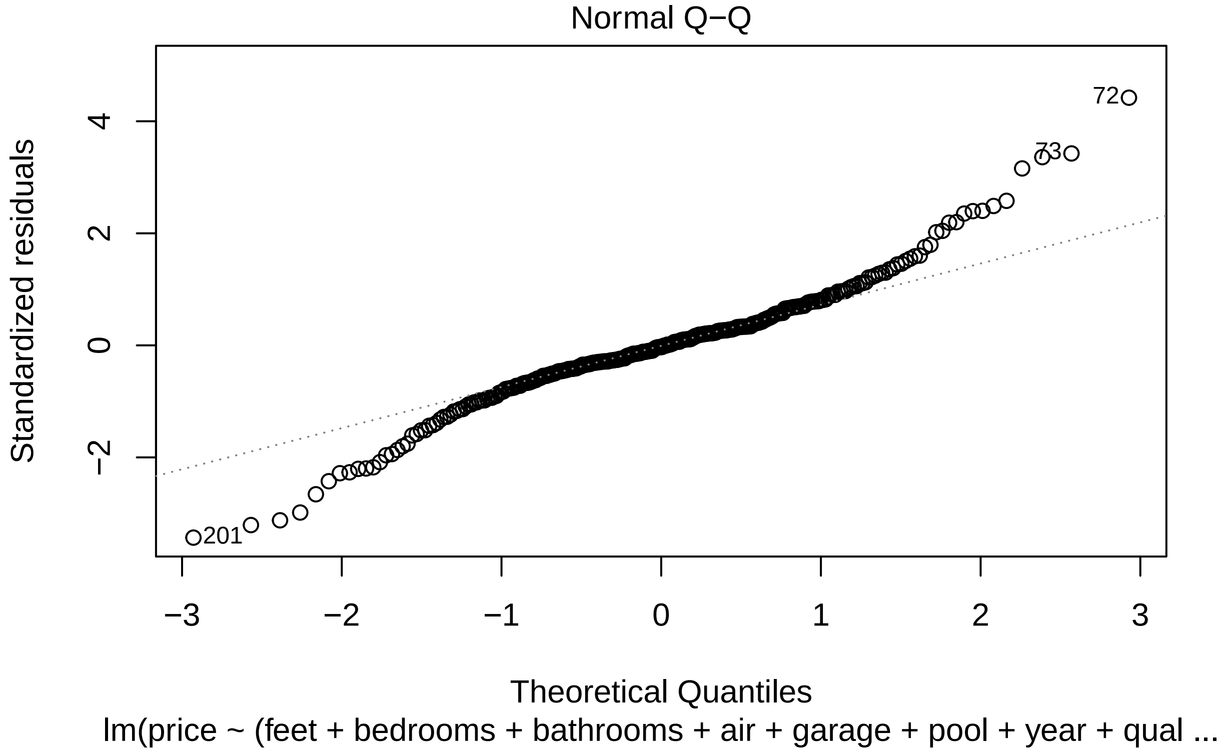
```
fit.full<-lm(price~(feet+bedrooms+bathrooms+air+garage+pool+year+quality+style+lot+highway)^2,.dt,
#summary(fit.full)$coe[,4]
summary(fit.full)$r.squared
```

```
## [1] 0.9012007
```

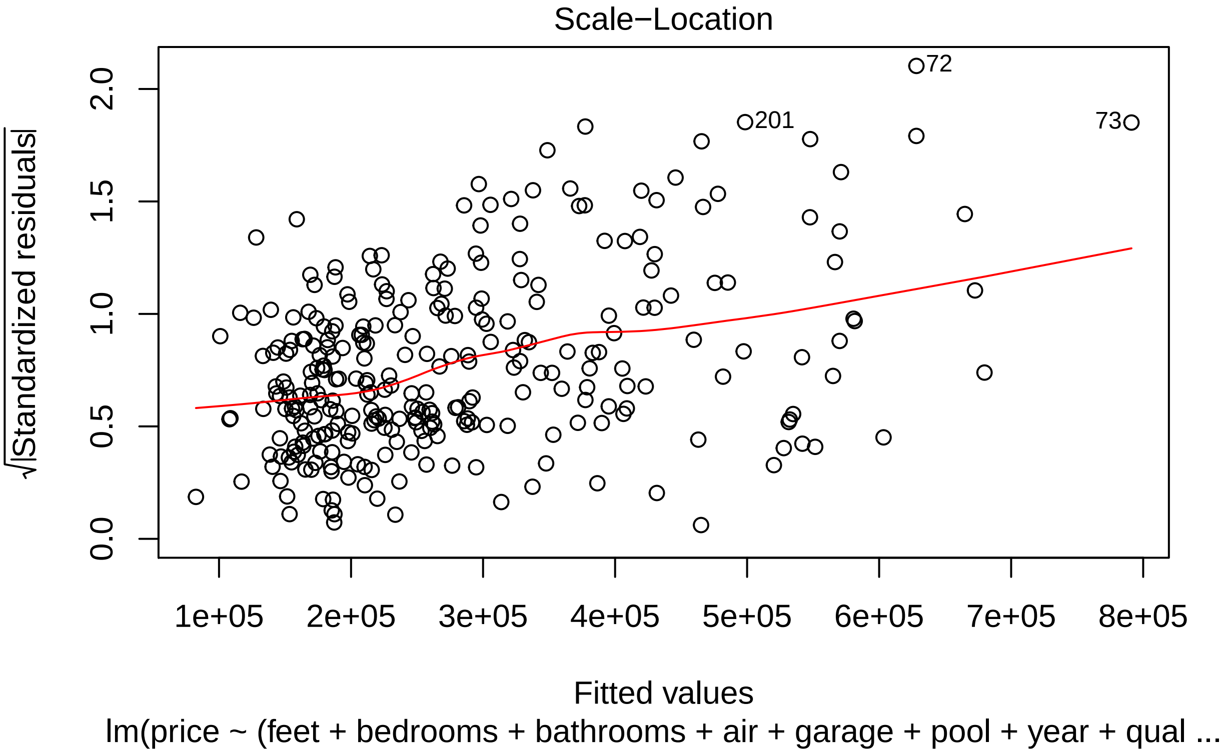
```r
plot(fit.full)
```

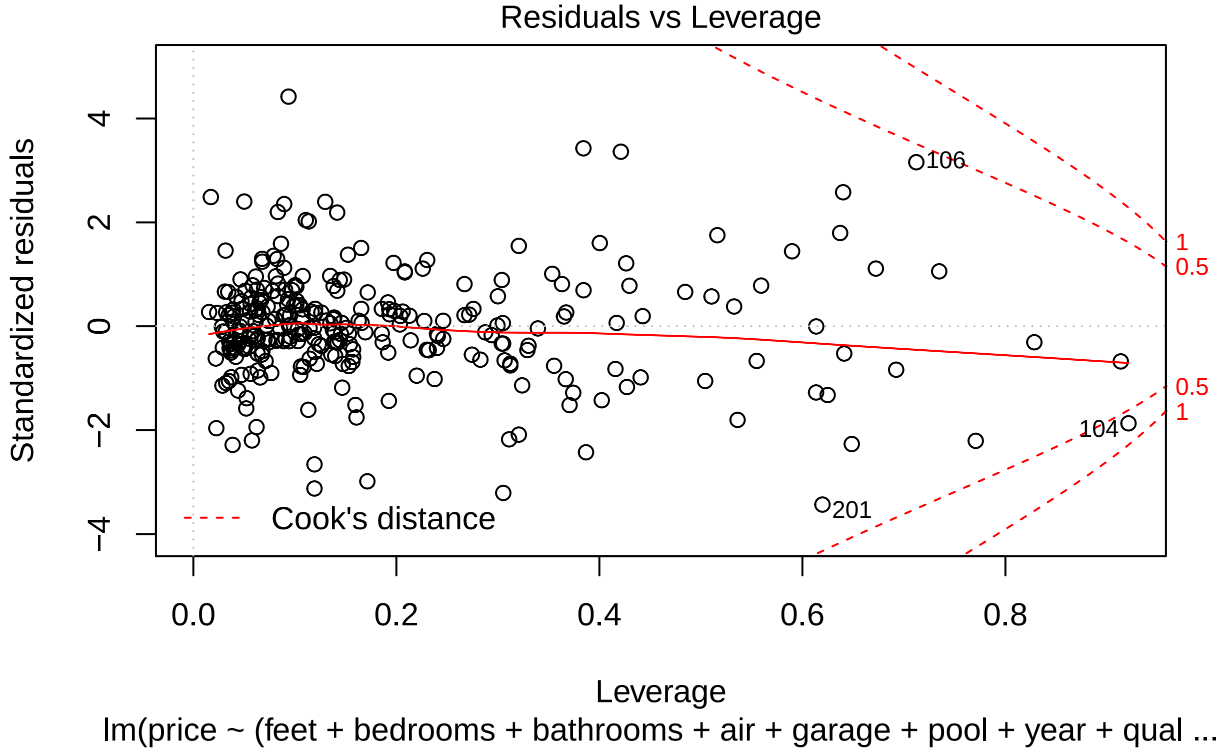
```
## Warning: not plotting observations with leverage one:
##   32, 49, 56, 164, 211, 284
```





```
## Warning: not plotting observations with leverage one:
##   32, 49, 56, 164, 211, 284
```





等方差不太成立，但是上下还是均匀的，也就是说模型还是不错的，正态性偏离也严重

```r
bptest(fit.full)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit.full
## BP = 85.094, df = 60, p-value = 0.01826
```

```r
shapiro.test(resid(fit.full))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(fit.full)
## W = 0.96018, p-value = 2.594e-07
```

```r
durbinWatsonTest(fit.full)
```

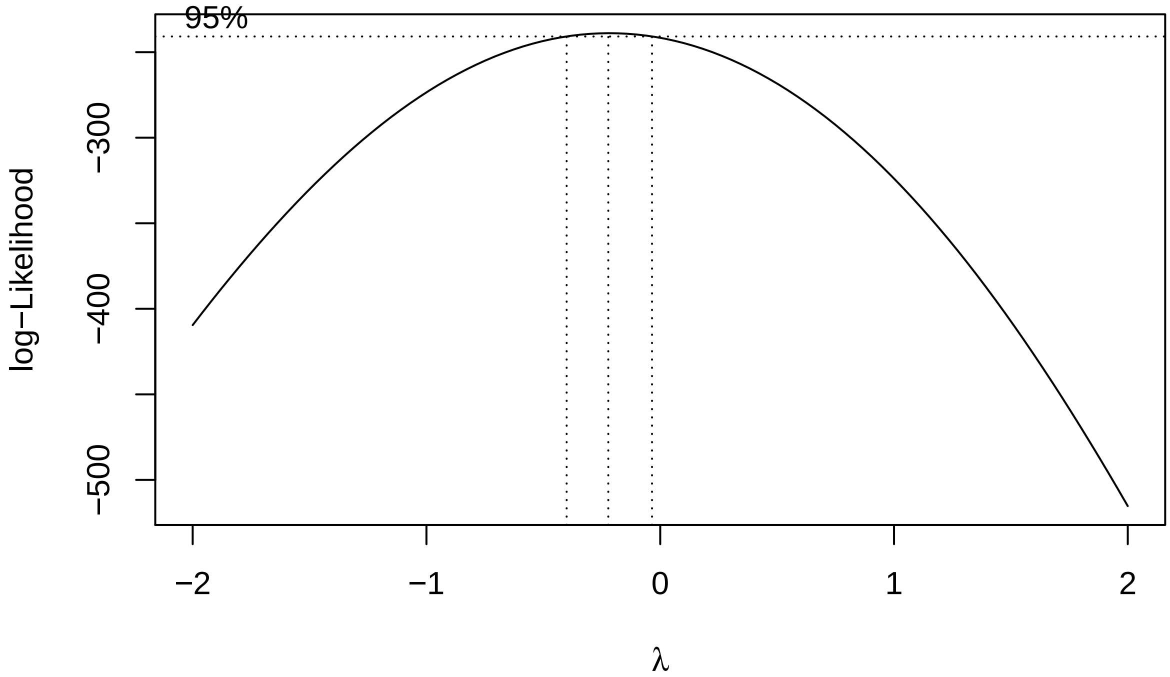
```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.06711234      1.846963   0.186
##  Alternative hypothesis: rho != 0
```

等方差的假设有点问题但不算太严重，方差不相关的假设还是没被拒绝，不过误差正态性不满足

幂变换

尝试做变换

```r
boxcox(fit.full)
```



但鉴于做简单线性回归的教训，虽然幂变换能改善不等方差和正态性，但可能 SSE 非常大，用于预测效果很差

```
findTransform<-function(lambda) {
.dt$y.tran.0<-with(.dt,ifelse(lambda==0,log(price),price^(lambda)))
#.dt$y.tran.0<-log(.dt$price)
fit.full.0<-lm(y.tran.0~feet+bedrooms+bathrooms+air+garage+pool+year+quality+style+lot+highway,.dt
sum((exp(predict(fit.full.0))-.dt[.dt$class=="training",]$price)^2)
}
tmp<-seq(-2,2,by=0.1)[sapply(seq(-2,1,by=0.1),findTransform)==min(sapply(seq(-2,2,by=0.1),findTran
findTransform(tmp)/deviance(fit.full)
```

```
## [1] 44.34568
```

对 Y 做任何变换偏差很大，预测很不准确 (最好的对数变换 SSE 增大了近 10 倍)，这是不可接受，所以尽管原来的模型不满足误差正态性假设，但由于要求的是预测，而不是推断，所以我们决定不对 Y(price) 做变换.

模型选择

由于变量个数很多，所以不再用手动 drop1,add1 的做法 ### step #### backward

```
#step(fit.full, direction="backward")
fit.backward<-lm(price ~ feet + bedrooms + bathrooms + air + garage +
    pool + year + quality + style + lot + feet:bedrooms + feet:garage +
    feet:year + feet:lot + bedrooms:bathrooms + bedrooms:air +
    bedrooms:pool + bedrooms:year + bathrooms:year + air:year +
    air:lot + garage:pool + garage:quality + pool:style + year:style +
    year:lot + quality:lot + style:lot,.dt,class=='training')
```

由于 step 的输出结果太长，因此注释了，实际上是会用到的，结果在第二行得到的 formula 是：formula = price ~ feet + bedrooms + bathrooms + air + garage + pool + year + quality + style + lot + feet:bedrooms + feet:garage + feet:year + feet:lot + bedrooms:bathrooms + bedrooms:air + bedrooms:pool + bedrooms:year + bathrooms:year + air:year + air:lot + garage:pool + garage:quality + pool:style + year:style + year:lot + quality:lot + style:lot #### forward 如果是 forward

```
#step(fit.null, scope = list(upper=fit.full),direction="forward")
fit.forward<-lm(formula = price ~ feet + quality + style + garage + lot +
    year + bathrooms + pool + quality:garage + feet:garage +
    feet:year + feet:style + quality:lot + style:year + lot:year +
    feet:lot + style:lot + garage:lot + feet:bathrooms + garage:year +
    lot:pool + garage:pool + quality:pool + quality:year, data = .dt,
    subset = class == "training")
```

lm(formula = price ~ feet + quality + style + garage + lot + year + bathrooms + pool + quality:garage + feet:garage + feet:year + feet:style + quality:lot + style:year + lot:year + feet:lot + style:lot + garage:lot + feet:bathrooms + garage:year + lot:pool + garage:pool + quality:pool + quality:year, data = .dt, subset = class == "training")，这个结果比起 backward 要少

BIC 结果相同，不过这是用 AIC 得到的，试一试保留变量数更少的 BIC，这次我们用 both(实际上还是 Forward)

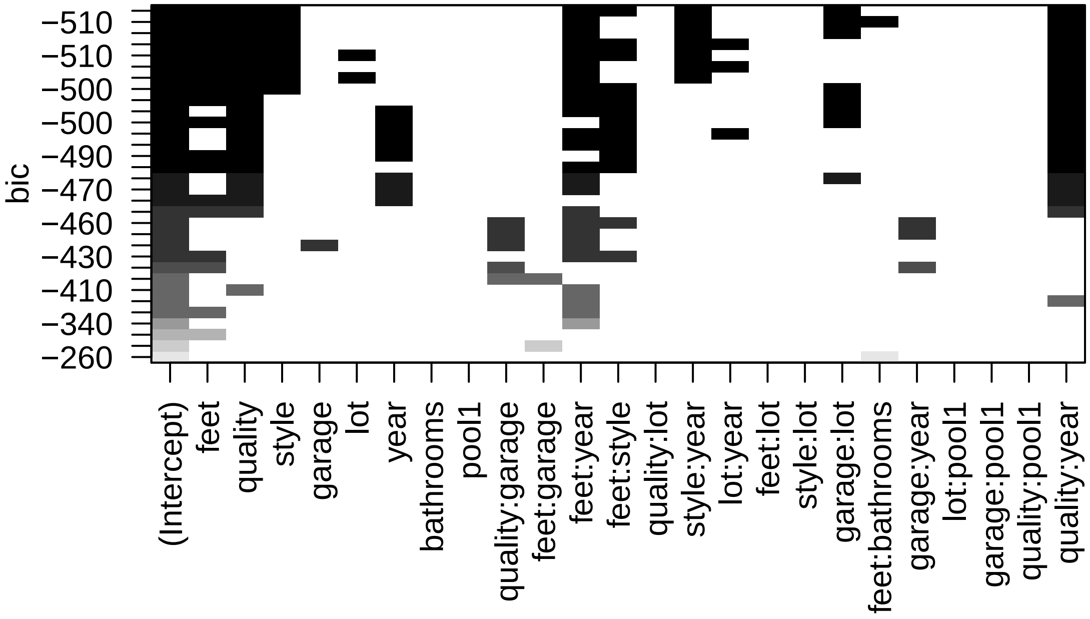
```
#step(fit.null,scope=list(upper=fit.full),dir='both',k=log(train_size))
fit.bic<-lm(formula = price ~ feet + quality + style + garage + lot +
    year + bathrooms + quality:garage + feet:garage + feet:year +
    feet:style + quality:lot + style:year + lot:year, data = .dt,
    subset = class == "training")
```

lm(formula = price ~ feet + quality + style + garage + lot + year + bathrooms + quality:garage + feet:garage + feet:year + feet:style + quality:lot + style:year + lot:year, data = .dt, subset = class == "training") 变量数比起 backward 少得更多

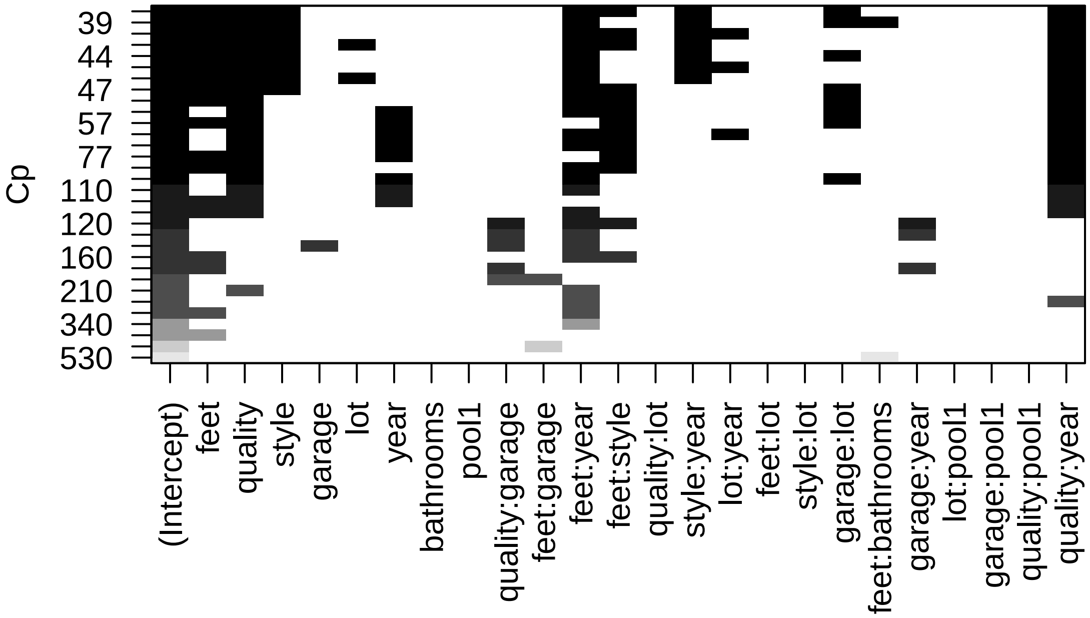
自动选择子集 (包括作图)

考虑变量个数很多，运行 regsubsets 要很多时间，因此我们用保留变量数最多的 backward 的结果

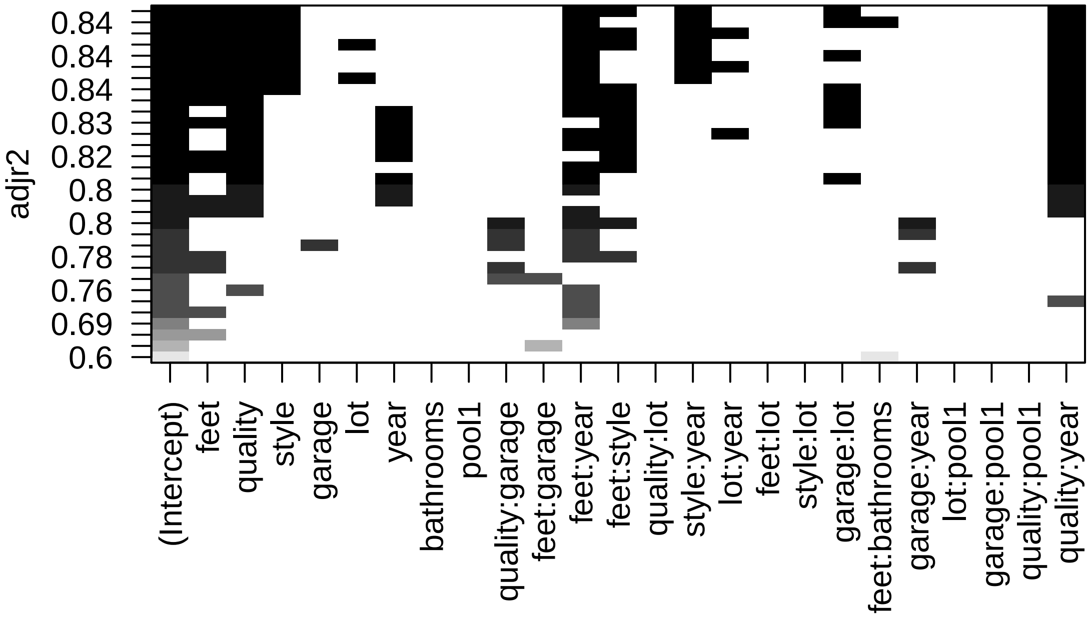
```
best <- function(model, ...)
{
  subsets <- regsubsets(formula(model), model.frame(model), ...)
  subsets <- with(summary(subsets),
                  cbind(p = as.numeric(rownames(which)), which, rss, rsq, adjr2, cp, bic))

  return(subsets)
}

subsets<-regsubsets(formula(fit.forward), model.frame(fit.forward),nbest=4,really.big = TRUE)
plot(subsets, scale="bic")
```



```
plot(subsets, scale = "Cp")
```



```
plot(subsets, scale = "adjr2")
```

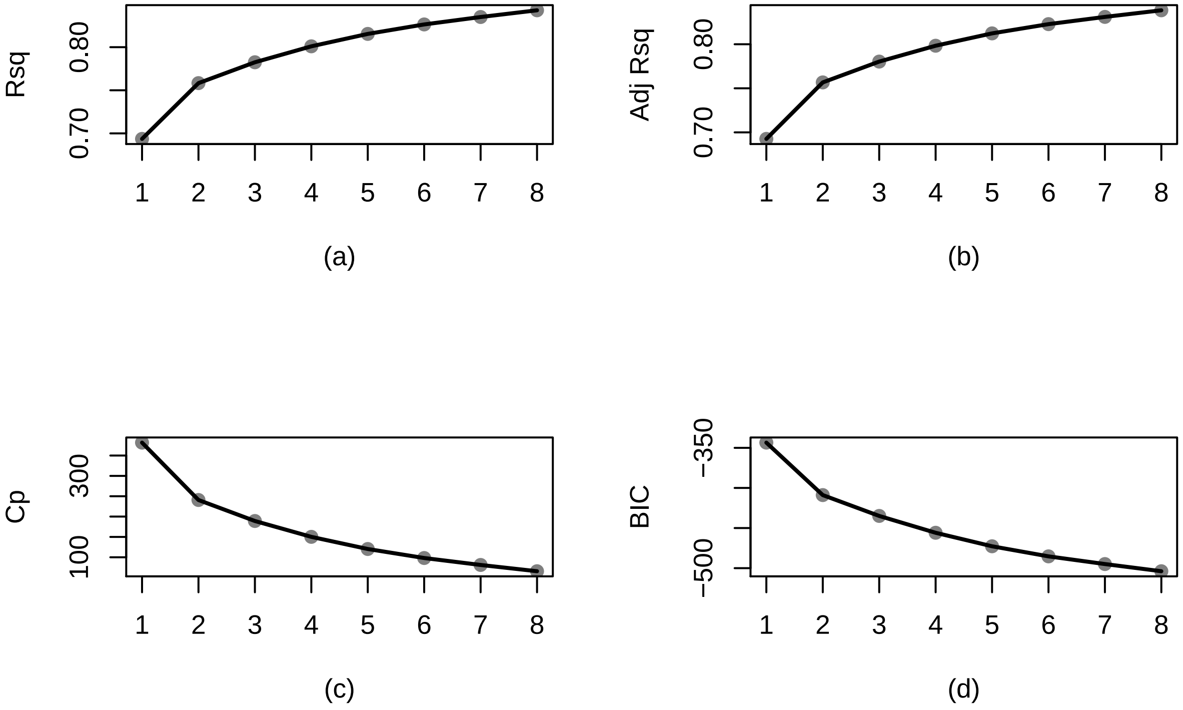


可以发现，这 3 幅图，上面的模型几乎没有变过，各种评价指标得到的最佳模型都一样

```
idx<-1:8
(x<-round(best(fit.backward),4))
```

```
##    p (Intercept) feet bedrooms bathrooms air garage pool1 year quality style lot
## 1 1           1    0        0         0   0      0     0    0       0     0   0
## 2 2           1    0        0         0   0      0     0    0       0     0   0
## 3 3           1    0        0         0   0      1     0    0       0     0   0
## 4 4           1    1        0         0   0      0     0    0       0     1   0
## 5 5           1    1        0         0   0      0     0    0       0     1   1
## 6 6           1    1        0         0   0      0     0    0       0     1   1
## 7 7           1    1        0         0   0      0     0    1       1     1   1
## 8 8           1    1        0         0   0      0     0    1       1     1   0
##   feet:bedrooms feet:garage feet:year feet:lot bedrooms:bathrooms bedrooms:air
## 1             0           0         1        0                  0            0
## 2             0           1         0        0                  0            0
## 3             0           0         1        0                  0            0
## 4             0           0         1        0                  0            0
## 5             0           0         1        0                  0            0
## 6             1           0         1        0                  0            0
## 7             0           0         1        0                  0            0
## 8             0           0         1        0                  0            1
##   bedrooms:pool1 bedrooms:year bathrooms:year air:year air:lot garage:pool1
## 1              0             0              0        0       0            0
## 2              0             0              0        0       0            0
## 3              0             0              0        0       0            0
## 4              0             0              0        0       0            0
## 5              0             0              0        0       0            0
## 6              0             0              0        0       0            0
## 7              0             0              0        0       0            0
## 8              0             0              0        0       1            0
##   garage:quality pool1:style year:style year:lot quality:lot style:lot
## 1              0           0          0        0           0         0
## 2              1           0          0        0           0         0
## 3              1           0          0        0           0         0
## 4              0           0          1        0           0         0
## 5              0           0          1        0           0         0
## 6              0           0          1        0           0         0
## 7              0           0          1        0           0         0
## 8              0           0          1        0           0         0
##             rss    rsq  adjr2       cp       bic
## 1 1.703245e+12 0.6936 0.6926 381.7568 -343.4982
## 2 1.343869e+12 0.7583 0.7567 240.7534 -408.8893
## 3 1.209077e+12 0.7825 0.7803 189.1169 -434.8941
## 4 1.106147e+12 0.8010 0.7983 150.1590 -455.8826
## 5 1.026131e+12 0.8154 0.8123 120.3189 -472.7051
## 6 9.654425e+11 0.8264 0.8228  98.1697 -485.2905
## 7 9.174701e+11 0.8350 0.8310  81.0805 -494.8767
## 8 8.737990e+11 0.8428 0.8385  65.7028 -503.8038
```

```
par(mfrow = c(2, 2), pch = 19)
plot(rsq ~ p, x, xlab = "(a)", ylab = "Rsq", col = "gray50")
lines(idx, tapply(x[, "rsq"], x[, "p"], max), lwd = 2)

plot(adjr2 ~ p, x, xlab = "(b)", ylab = "Adj Rsq", col = "gray50")
lines(idx, tapply(x[, "adjr2"], x[, "p"], max), lwd = 2)

plot(cp ~ p, x, xlab = "(c)", ylab = "Cp", col = "gray50")
lines(idx, tapply(x[, "cp"], x[, "p"], min), lwd = 2)

plot(bic ~ p, x, xlab = "(d)", ylab = "BIC", col = "gray50")
lines(idx, tapply(x[, "bic"], x[, "p"], min), lwd = 2)
```



从这里可以看出，6-8 个变量是合适的，而 8 个就非常好了

根据 x，模型的选择情况是 | 变量个数 | formula | |———-|——————————————————————
——| | 6 | price~feet+style+lot+feet:year+year:style+quality:lot | | 7 | price~feet+style+lot+feet:year+year:style+year:
| | 8 | price~feet+bedrooms+style+lot+feet:year+bedrooms:year+year:style+quality:lot | ## 模型的预测能力的判断

```
newsummary <- function(formula)
{
    training.model<-lm(formula,.dt,class=="training")
    validation.model<-lm(formula,.dt,class=="validation")
    list('coefs'    = cbind(training=round(summary(training.model)$coef[, 1:2], 4),validation=cbin
         'criteria' = cbind(training=c(
                            'PRESS' = PRESS(training.model),
                            'MSE'   = anova(training.model)["Residuals", "Mean Sq"],
                            'Rsq'   = summary(training.model)$adj.r.squared),validation=c(
                            'PRESS' = PRESS(validation.model),
                            'MSE'   = anova(validation.model)["Residuals", "Mean Sq"],
                            'Rsq'   = summary(validation.model)$adj.r.squared)))

}
```

```
print('6 个变量')
```

```
## [1] "6个变量"
```

```
newsummary(price~feet+style+lot+feet:year+year:style+quality:lot)$criteria
```

```
##           training   validation
## PRESS 1.060704e+12 9.552962e+11
## MSE   3.318283e+09 3.825118e+09
## Rsq   8.215450e-01 8.057092e-01
```

```
print('7 个变量')
```

```
## [1] "7个变量"
```

```
newsummary(price~feet+style+lot+feet:year+year:style+year:lot+quality:lot)$criteria
```

```
##           training   validation
## PRESS 1.035712e+12 9.378043e+11
## MSE   3.203903e+09 3.720234e+09
## Rsq   8.276963e-01 8.110366e-01
```

```
print('8 个变量')
```

```
## [1] "8个变量"
```

```
newsummary(price~feet+bedrooms+style+lot+feet:year+bedrooms:year+year:style+quality:lot)$criteria
```

```
##           training   validation
## PRESS 9.837273e+11 9.645773e+11
## MSE   3.054731e+09 3.810232e+09
## Rsq   8.357186e-01 8.064653e-01
```

验证集竟然比训练集的效果还要好，原因就是，训练集包含了更多的极端情况. 通过对比 training 的

PRESS,我们选出具有 8 个变量的模型,也就是 price~feet+bedrooms+style+lot+feet:year+bedrooms:year+year:style+
同样也是在验证集上表现最好的.

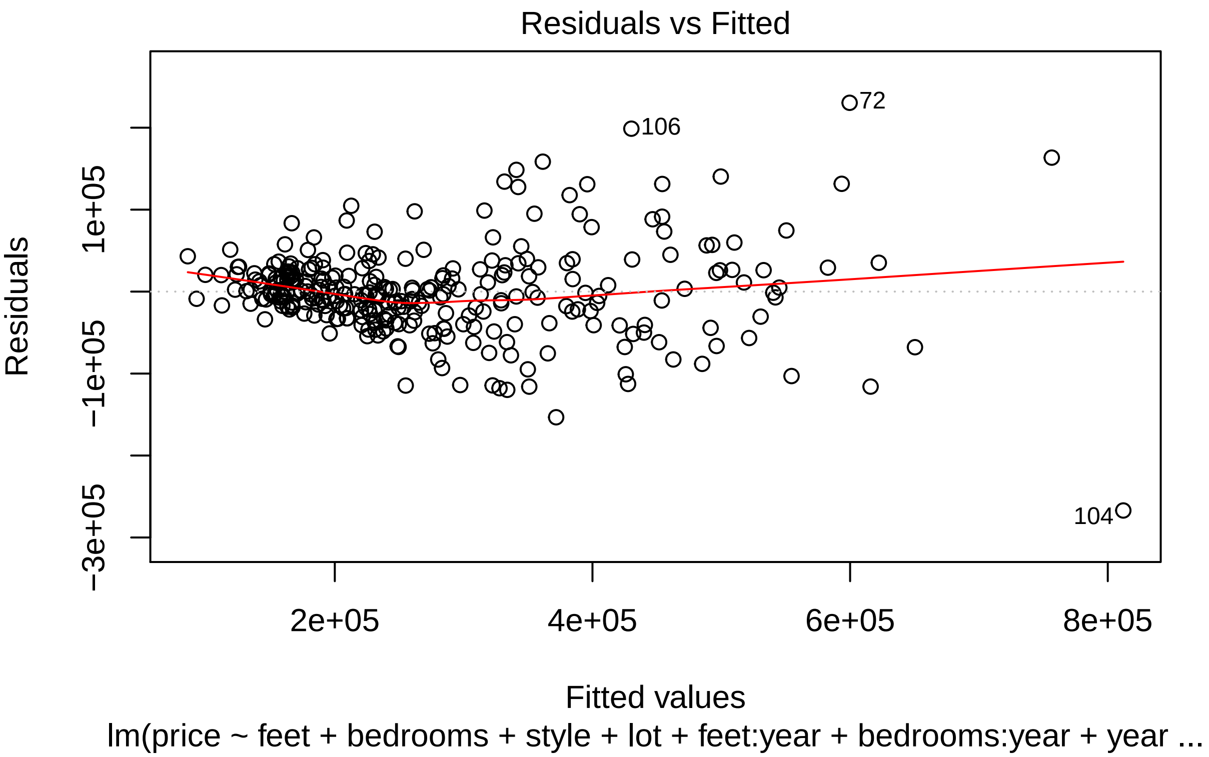
总结

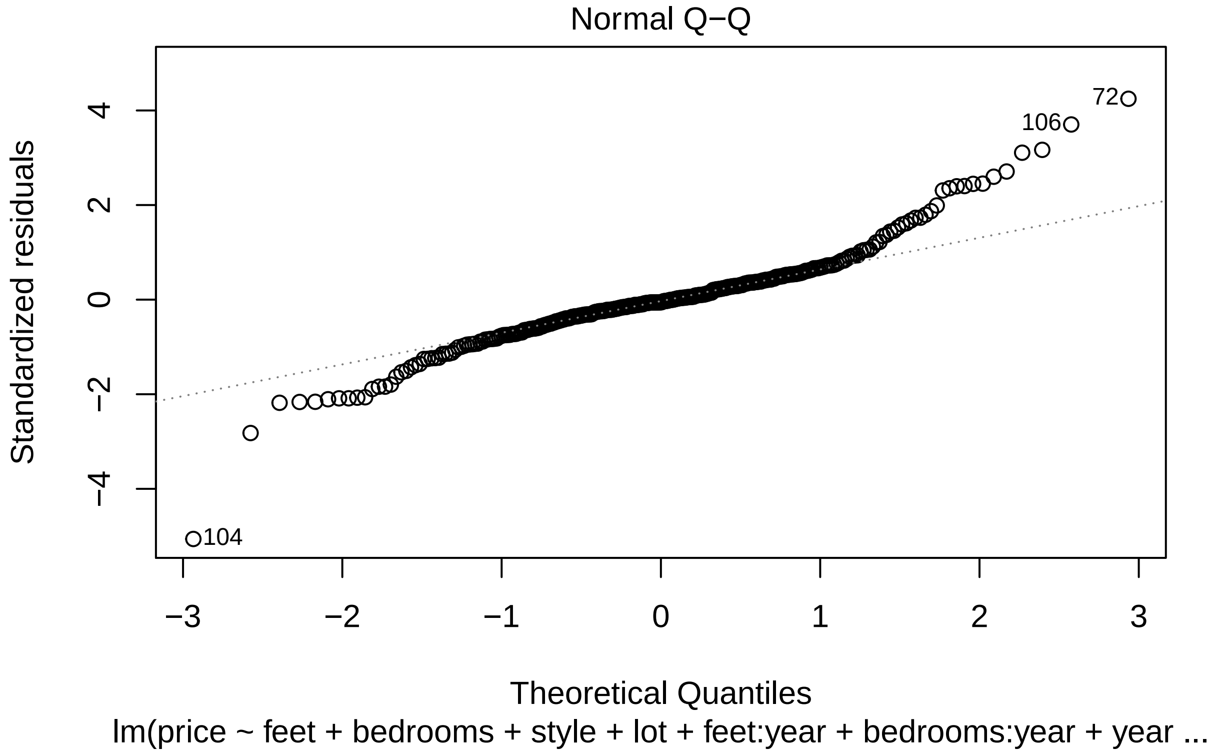
选择了有 8 个变量的模型,formula 是 `price~feet+bedrooms+style+lot+feet:year+bedrooms:year+year:style+qu`

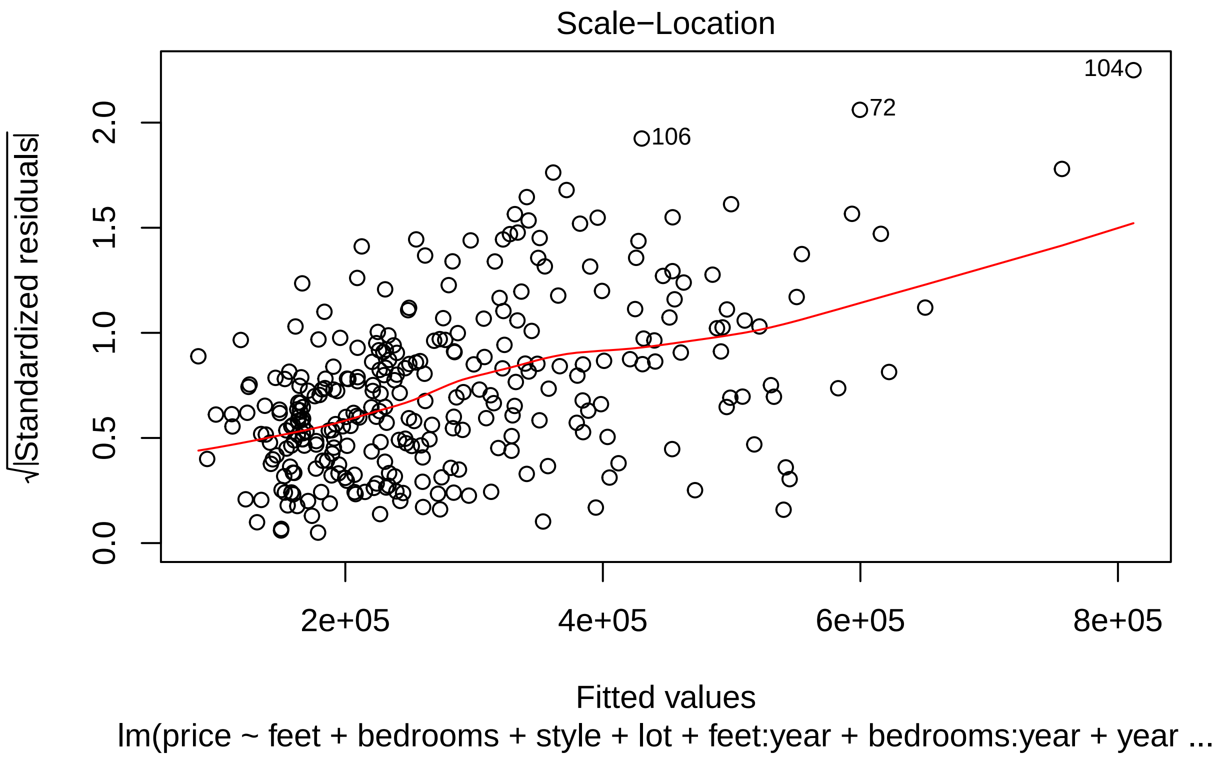
```
fit.final<-lm(price~feet+bedrooms+style+lot+feet:year+bedrooms:year+year:style+quality:lot,.dt,cla
summary(fit.final)
```

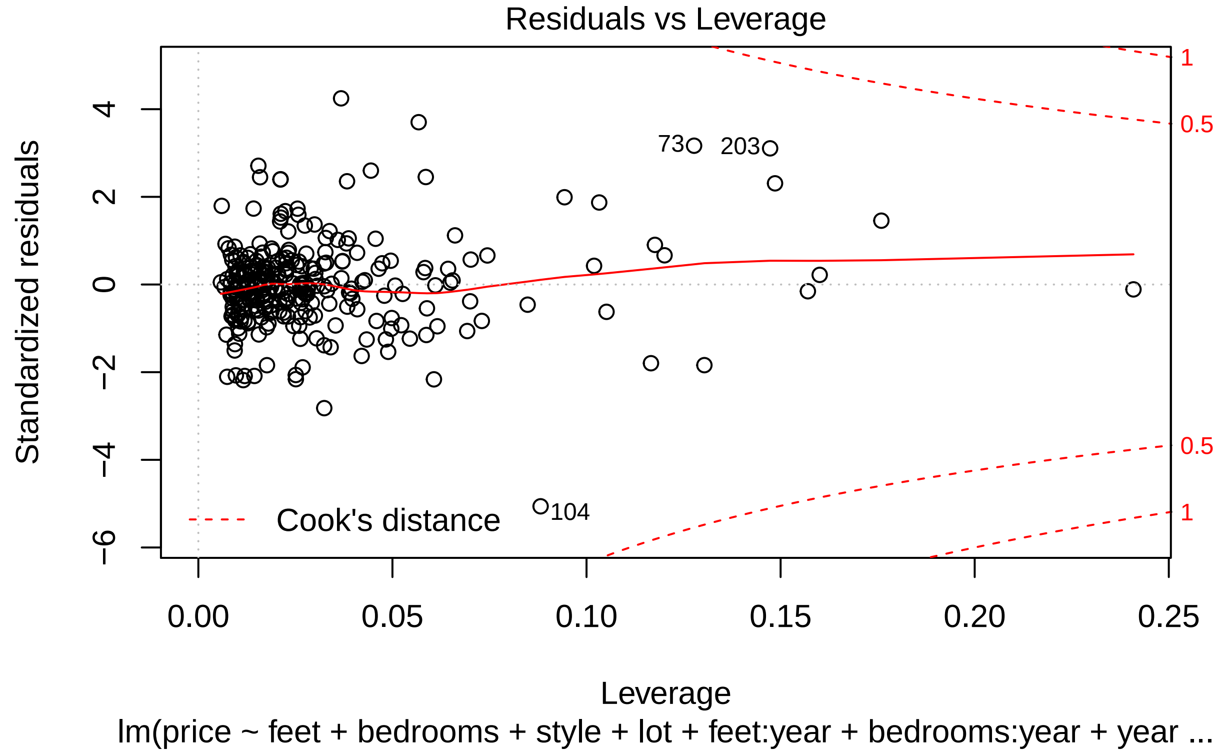
```
##
## Call:
## lm(formula = price ~ feet + bedrooms + style + lot + feet:year +
##     bedrooms:year + year:style + quality:lot, data = .dt, subset = class ==
##     "training")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -267056  -26423   -3069   22893  230384
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.268e+03  1.578e+04  -0.207    0.836
## feet          -5.681e+03  6.843e+02  -8.302 3.90e-15 ***
## bedrooms       1.793e+06  4.185e+05   4.285 2.48e-05 ***
## style          9.492e+05  1.734e+05   5.473 9.57e-08 ***
## lot            4.240e+00  7.059e-01   6.006 5.66e-09 ***
## feet:year      2.952e+00  3.469e-01   8.510 9.37e-16 ***
## bedrooms:year -9.163e+02  2.125e+02  -4.312 2.21e-05 ***
## style:year    -4.857e+02  8.813e+01  -5.511 7.88e-08 ***
## lot:quality   -1.192e+00  2.792e-01  -4.271 2.64e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55270 on 291 degrees of freedom
## Multiple R-squared:  0.8401, Adjusted R-squared:  0.8357
## F-statistic: 191.1 on 8 and 291 DF,  p-value: < 2.2e-16
```

```
plot(fit.final)
```


Residuals vs Fitted
Residuals
1e+05
−1e+05
−3e+05
2e+05
4e+05
6e+05
8e+05
106
72
104
Fitted values
lm(price ~ feet + bedrooms + style + lot + feet:year + bedrooms:year + year ...


Normal Q−Q
Standardized residuals
4
2
0
−2
−4
−3
−2
−1
0
1
2
3
72
106
104
Theoretical Quantiles
lm(price ~ feet + bedrooms + style + lot + feet:year + bedrooms:year + year ...





```
shapiro.test(fit.final$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  fit.final$residuals
## W = 0.93891, p-value = 8.534e-10
```

```
durbinWatsonTest(fit.final)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.04014526      1.910823   0.412
##  Alternative hypothesis: rho != 0
```

```
bptest(fit.final)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit.final
## BP = 77.685, df = 8, p-value = 1.427e-13
```

误差等方差不成立，误差正态不成立 (这是因为没有对 y 做对数变换，但是如前所说，为了不牺牲预测效果，放弃了对数变换)，误差无关成立. 可能有条件不错但价格高得离谱的房子，也会有条件很好但便宜卖的房子.