Travaux Pratiques - Modèles de Régression régularisée

MECH Bealy & CHHEANG Vinha

10/10/2021

IV. Real estate data

Load data:

```
dataHouse = read.csv("./housedata.csv")
names(dataHouse)
```

```
##  [1] "id"            "date"          "price"         "bedrooms"
##  [5] "bathrooms"     "sqft_living"   "sqft_lot"      "floors"
##  [9] "waterfront"    "view"          "condition"     "grade"
## [13] "sqft_above"    "sqft_basement" "yr_built"      "yr_renovated"
## [17] "zipcode"       "lat"           "long"          "sqft_living15"
## [21] "sqft_lot15"
```

```
head(dataHouse)
```

```
##           id            date   price bedrooms bathrooms sqft_living sqft_lot
## 1 7129300520 20141013T000000  221900        3      1.00        1180     5650
## 2 6414100192 20141209T000000  538000        3      2.25        2570     7242
## 3 5631500400 20150225T000000  180000        2      1.00         770    10000
## 4 2487200875 20141209T000000  604000        4      3.00        1960     5000
## 5 1954400510 20150218T000000  510000        3      2.00        1680     8080
## 6 7237550310 20140512T000000 1225000        4      4.50        5420   101930
##   floors waterfront view condition grade sqft_above sqft_basement yr_built
## 1      1          0    0         3     7       1180             0     1955
## 2      2          0    0         3     7       2170           400     1951
## 3      1          0    0         3     6        770             0     1933
## 4      1          0    0         5     7       1050           910     1965
## 5      1          0    0         3     8       1680             0     1987
## 6      1          0    0         3    11       3890          1530     2001
##   yr_renovated zipcode     lat     long sqft_living15 sqft_lot15
## 1            0   98178 47.5112 -122.257          1340       5650
## 2         1991   98125 47.7210 -122.319          1690       7639
## 3            0   98028 47.7379 -122.233          2720       8062
## 4            0   98136 47.5208 -122.393          1360       5000
## 5            0   98074 47.6168 -122.045          1800       7503
## 6            0   98053 47.6561 -122.005          4760     101930
```

```
str(dataHouse)
```

```
## 'data.frame':    21613 obs. of  21 variables:
##  $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : chr  "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...
##  $ price        : num  221900 538000 180000 604000 510000 ...
```

```
##  $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
##  $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
##  $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
##  $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
##  $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
##  $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
##  $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
```

The number of observation: 21613.

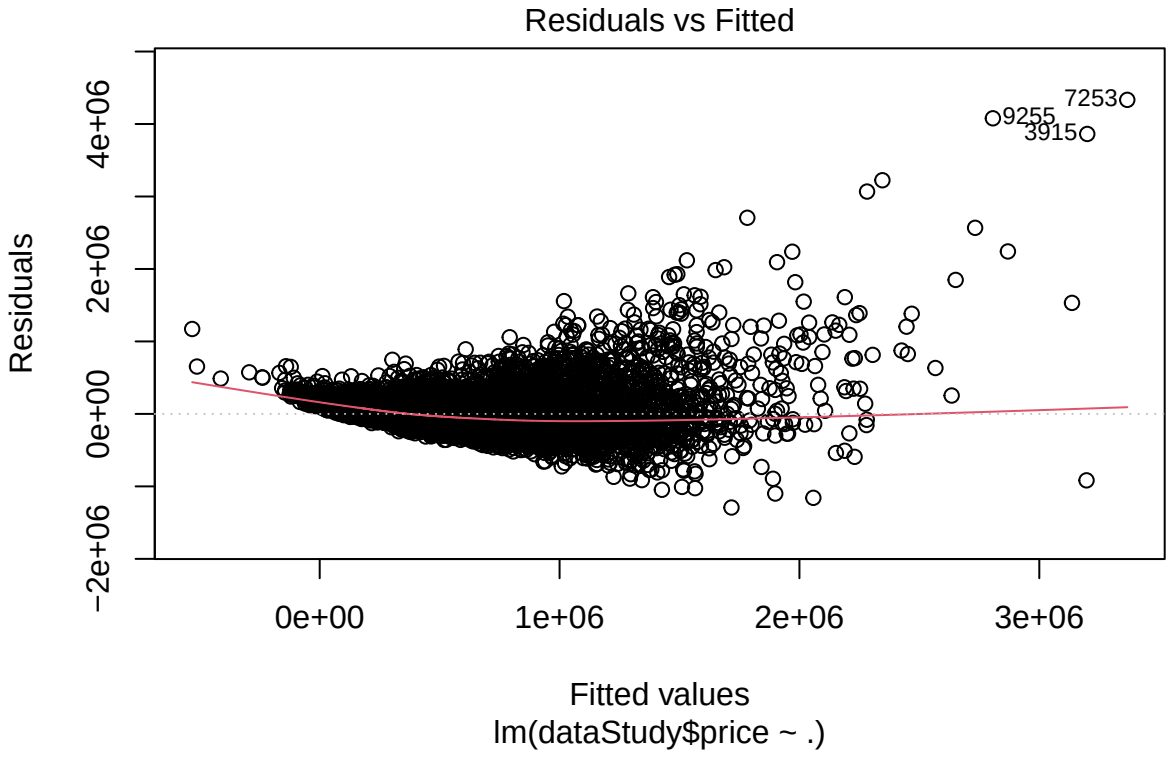
A. Study the ability of regular linear model with variable selection:

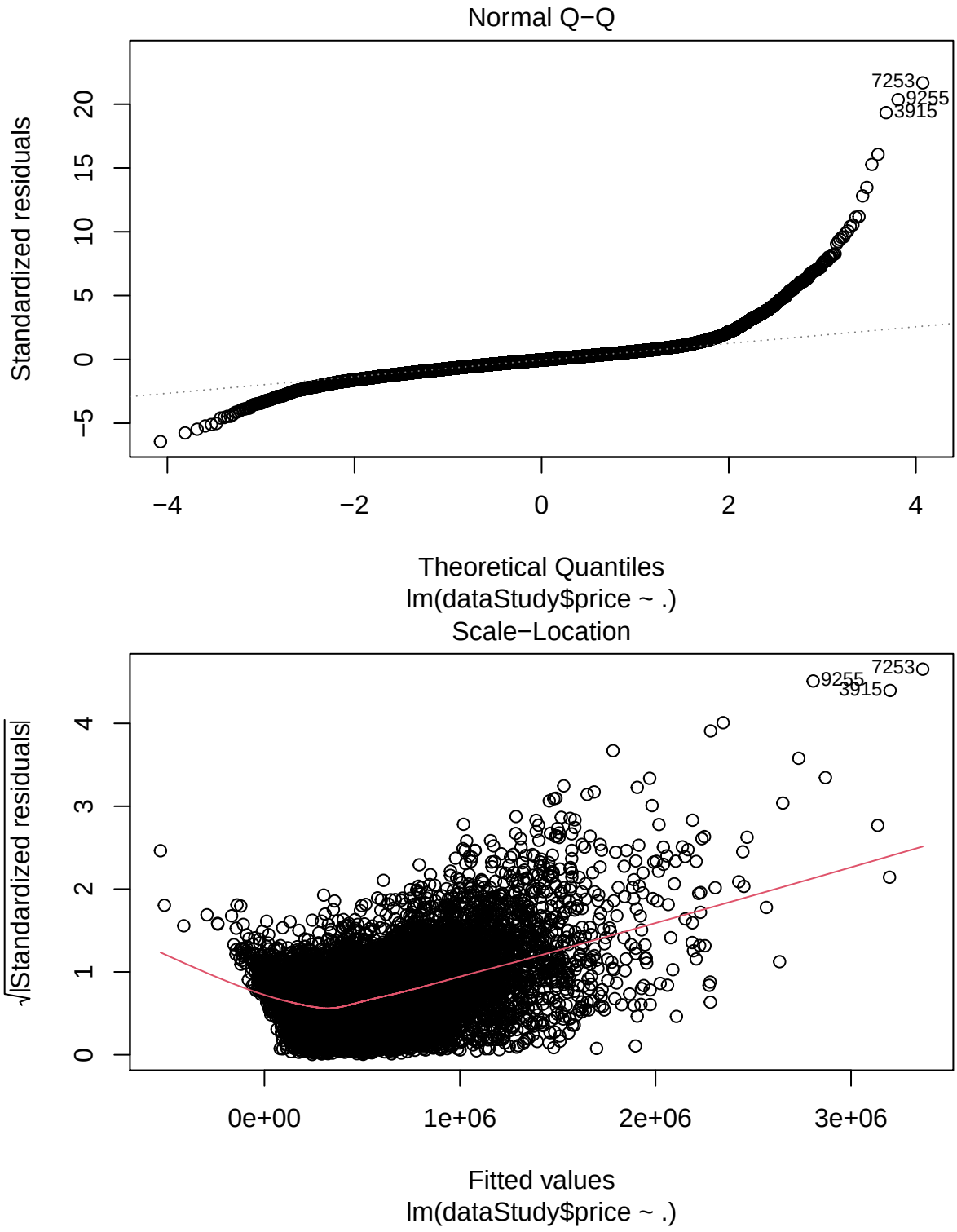
```
dataStudy = dataHouse[ , 3:21]
mol = lm(dataStudy$price~. , data = dataStudy)
summary(mol)
```

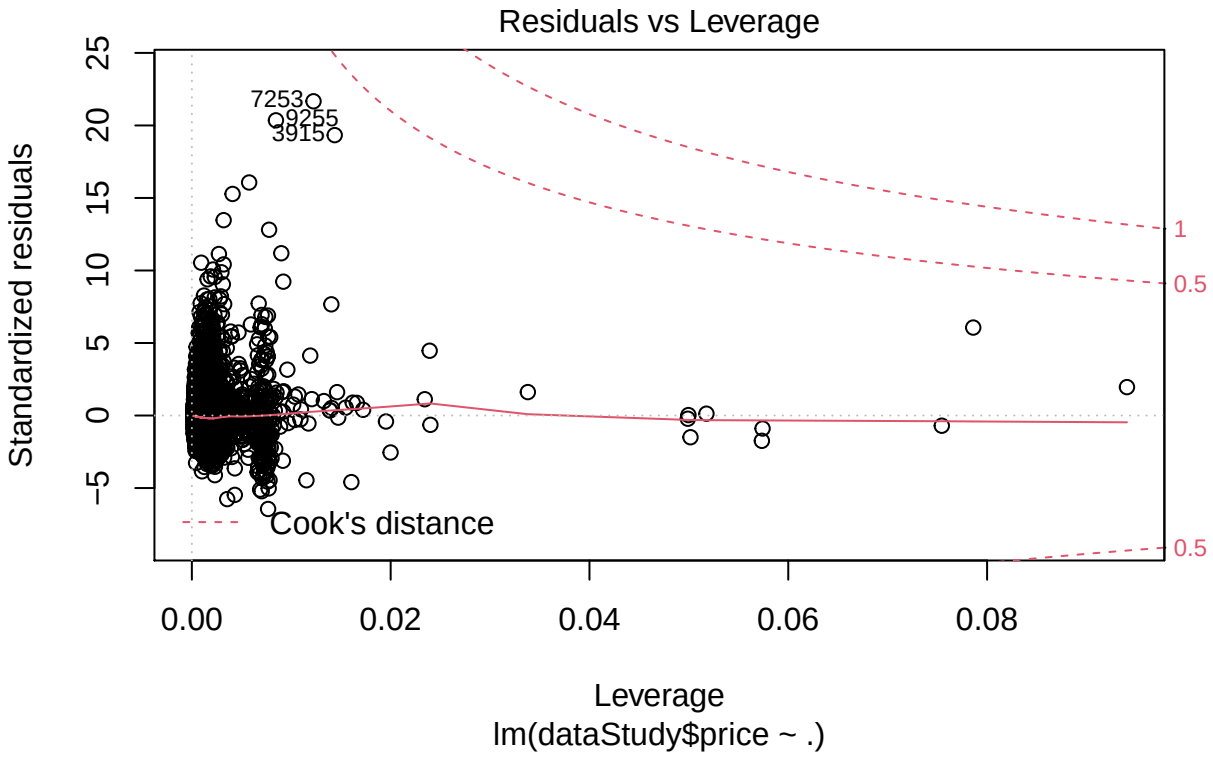
```
##
## Call:
## lm(formula = dataStudy$price ~ ., data = dataStudy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1291725   -99229    -9739    77583  4333222
##
## Coefficients: (1 not defined because of singularities)
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.690e+06  2.931e+06   2.282  0.02249 *
## bedrooms      -3.577e+04  1.892e+03 -18.906  < 2e-16 ***
## bathrooms      4.114e+04  3.254e+03  12.645  < 2e-16 ***
## sqft_living    1.501e+02  4.385e+00  34.227  < 2e-16 ***
## sqft_lot       1.286e-01  4.792e-02   2.683  0.00729 **
## floors         6.690e+03  3.596e+03   1.860  0.06285 .
## waterfront     5.830e+05  1.736e+04  33.580  < 2e-16 ***
## view           5.287e+04  2.140e+03  24.705  < 2e-16 ***
## condition      2.639e+04  2.351e+03  11.221  < 2e-16 ***
## grade          9.589e+04  2.153e+03  44.542  < 2e-16 ***
## sqft_above     3.113e+01  4.360e+00   7.139 9.71e-13 ***
## sqft_basement         NA         NA      NA       NA
## yr_built      -2.620e+03  7.266e+01 -36.062  < 2e-16 ***
## yr_renovated   1.981e+01  3.656e+00   5.420 6.03e-08 ***
## zipcode       -5.824e+02  3.299e+01 -17.657  < 2e-16 ***
## lat            6.027e+05  1.073e+04  56.149  < 2e-16 ***
## long          -2.147e+05  1.313e+04 -16.349  < 2e-16 ***
```

```
## sqft_living15  2.168e+01  3.448e+00   6.289 3.26e-10 ***
## sqft_lot15    -3.826e-01  7.327e-02  -5.222 1.78e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201200 on 21595 degrees of freedom
## Multiple R-squared:  0.6997, Adjusted R-squared:  0.6995
## F-statistic:  2960 on 17 and 21595 DF,  p-value: < 2.2e-16
```

```
plot(mol)
```




Normal Q−Q
Standardized residuals
7253
9255
3915
−5
0
5
10
15
20
−4
−2
0
2
4
Theoretical Quantiles
lm(dataStudy$price ~ .)
Scale−Location
√|Standardized residuals|
9255
7253
3915
0
1
2
3
4
0e+00
1e+06
2e+06
3e+06
Fitted values
lm(dataStudy$price ~ .)

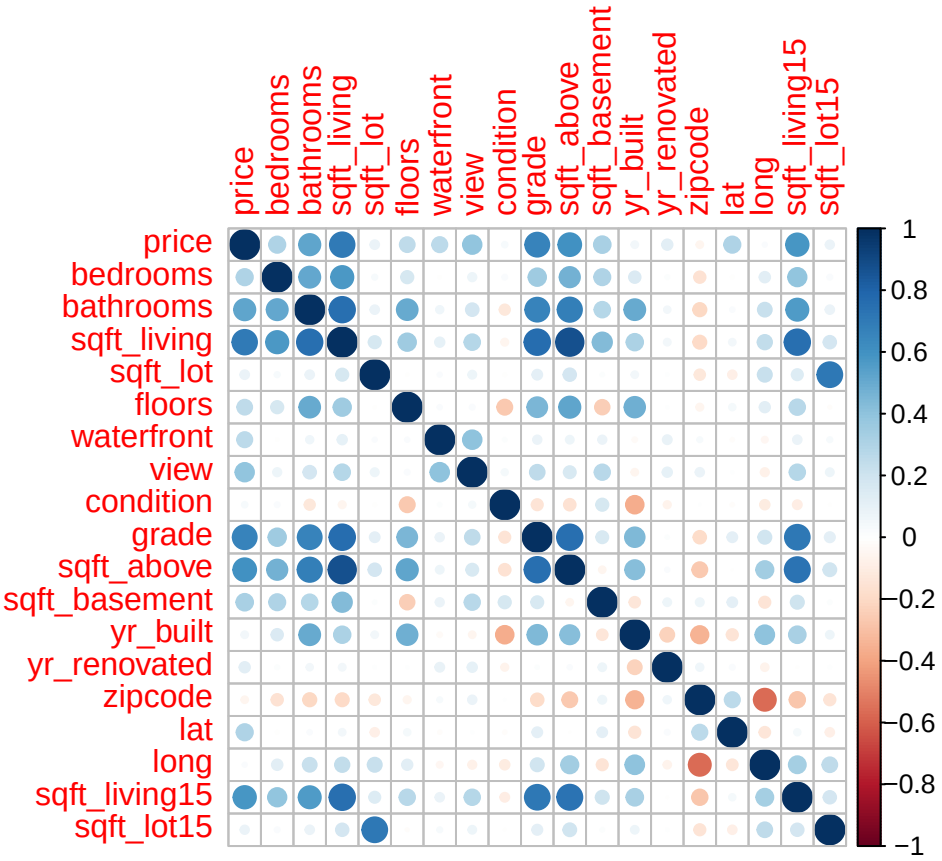


check corrolation of all variables:

```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
corrplot(cor(dataStudy))
```



Backward selection:

```
Regbackward=step(mol,direction='backward')
```

```
## Start:  AIC=527906.5
## dataStudy$price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_above +
##     sqft_basement + yr_built + yr_renovated + zipcode + lat +
##     long + sqft_living15 + sqft_lot15
##
##
## Step:  AIC=527906.5
## dataStudy$price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     floors + waterfront + view + condition + grade + sqft_above +
##     yr_built + yr_renovated + zipcode + lat + long + sqft_living15 +
##     sqft_lot15
##
##                 Df  Sum of Sq        RSS    AIC
## <none>                        8.7461e+14 527907
## - floors         1 1.4017e+11 8.7475e+14 527908
## - sqft_lot       1 2.9164e+11 8.7490e+14 527912
## - sqft_lot15     1 1.1046e+12 8.7572e+14 527932
## - yr_renovated   1 1.1897e+12 8.7580e+14 527934
## - sqft_living15  1 1.6017e+12 8.7621e+14 527944
## - sqft_above     1 2.0640e+12 8.7668e+14 527955
## - condition      1 5.0994e+12 8.7971e+14 528030
```

```
## - bathrooms      1 6.4764e+12 8.8109e+14 528064
## - long           1 1.0826e+13 8.8544e+14 528170
## - zipcode        1 1.2626e+13 8.8724e+14 528214
## - bedrooms       1 1.4476e+13 8.8909e+14 528259
## - view           1 2.4720e+13 8.9933e+14 528507
## - waterfront     1 4.5671e+13 9.2028e+14 529005
## - sqft_living    1 4.7447e+13 9.2206e+14 529046
## - yr_built       1 5.2669e+13 9.2728e+14 529168
## - grade          1 8.0354e+13 9.5497e+14 529804
## - lat            1 1.2769e+14 1.0023e+15 530850
```

```
summary(Regbackward)
```

```
##
## Call:
## lm(formula = dataStudy$price ~ bedrooms + bathrooms + sqft_living +
##     sqft_lot + floors + waterfront + view + condition + grade +
##     sqft_above + yr_built + yr_renovated + zipcode + lat + long +
##     sqft_living15 + sqft_lot15, data = dataStudy)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1291725   -99229    -9739    77583  4333222
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.690e+06  2.931e+06   2.282  0.02249 *
## bedrooms      -3.577e+04  1.892e+03 -18.906  < 2e-16 ***
## bathrooms      4.114e+04  3.254e+03  12.645  < 2e-16 ***
## sqft_living    1.501e+02  4.385e+00  34.227  < 2e-16 ***
## sqft_lot       1.286e-01  4.792e-02   2.683  0.00729 **
## floors         6.690e+03  3.596e+03   1.860  0.06285 .
## waterfront     5.830e+05  1.736e+04  33.580  < 2e-16 ***
## view           5.287e+04  2.140e+03  24.705  < 2e-16 ***
## condition      2.639e+04  2.351e+03  11.221  < 2e-16 ***
## grade          9.589e+04  2.153e+03  44.542  < 2e-16 ***
## sqft_above     3.113e+01  4.360e+00   7.139 9.71e-13 ***
## yr_built      -2.620e+03  7.266e+01 -36.062  < 2e-16 ***
## yr_renovated   1.981e+01  3.656e+00   5.420 6.03e-08 ***
## zipcode       -5.824e+02  3.299e+01 -17.657  < 2e-16 ***
## lat            6.027e+05  1.073e+04  56.149  < 2e-16 ***
## long          -2.147e+05  1.313e+04 -16.349  < 2e-16 ***
## sqft_living15  2.168e+01  3.448e+00   6.289 3.26e-10 ***
## sqft_lot15    -3.826e-01  7.327e-02  -5.222 1.78e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201200 on 21595 degrees of freedom
## Multiple R-squared:  0.6997, Adjusted R-squared:  0.6995
## F-statistic:  2960 on 17 and 21595 DF,  p-value: < 2.2e-16
```

After using backward selection, we can not eliminate variable in model. However, we will check it with LASSO model.

B. LASSO model

```
library(lars)
```

```
## Loaded lars 1.2
```

```
dataLasso = dataHouse[ ,4:21]
X = as.matrix(cbind(dataLasso[,-ncol(dataLasso)]))
Y = dataHouse[ ,3]
modelLasso = lars(X, Y, type = "lasso")
modelLasso$beta
```

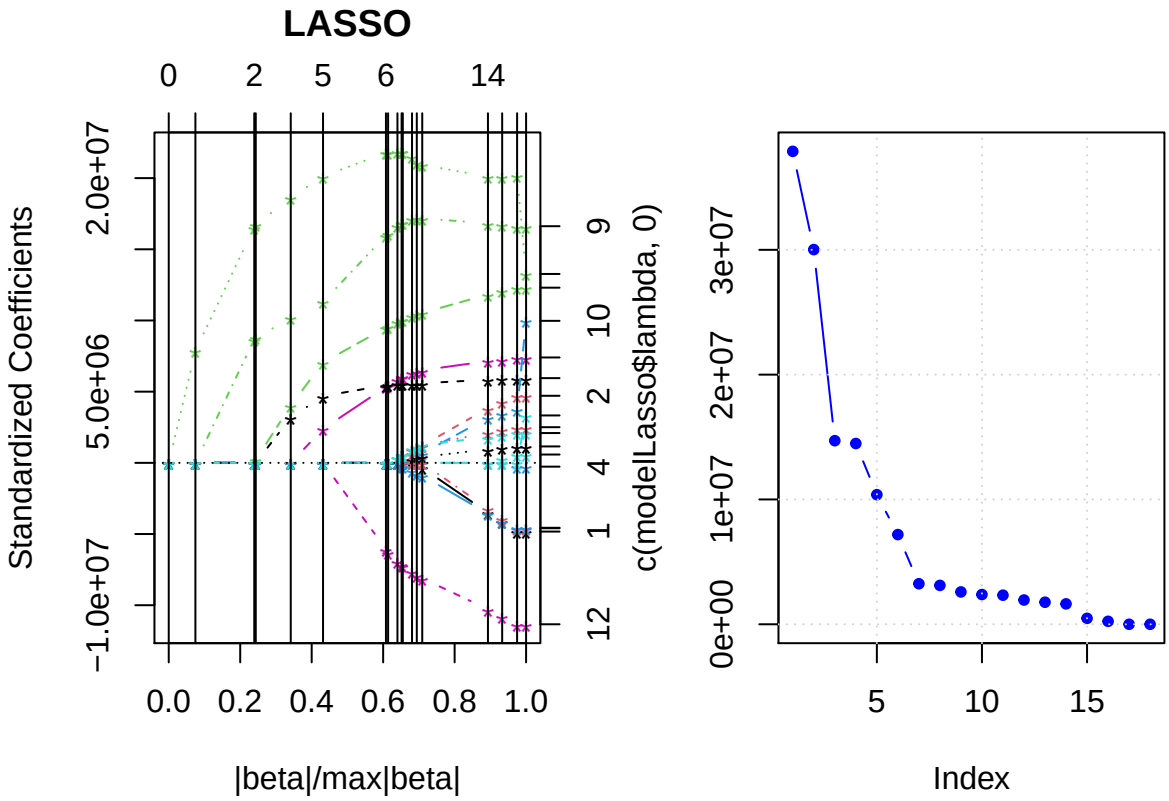
```
##      bedrooms  bathrooms sqft_living    sqft_lot   floors waterfront     view
## 0       0.000     0.0000     0.00000  0.00000000    0.000        0.0     0.00
## 1       0.000     0.0000    58.28569  0.00000000    0.000        0.0     0.00
## 2       0.000     0.0000   122.61879  0.00000000    0.000        0.0     0.00
## 3       0.000     0.0000   123.70180  0.00000000    0.000        0.0     0.00
## 4       0.000     0.0000   138.03791  0.00000000    0.000        0.0 28201.76
## 5       0.000     0.0000   149.10566  0.00000000    0.000   187388.7 41593.89
## 6       0.000     0.0000   161.30757  0.00000000    0.000   419103.5 48430.33
## 7       0.000     0.0000   161.47510  0.00000000    0.000   426987.1 48617.54
## 8       0.000     0.0000   161.82909  0.00000000    0.000   458236.3 49311.29
## 9       0.000     0.0000   162.04843  0.00000000    0.000   470725.4 49384.75
## 10      0.000   721.7134   161.70589  0.00000000    0.000   473822.2 49391.62
## 11      0.000  5642.7250   159.17011  0.00000000    0.000   495944.0 49372.21
## 12      0.000  8041.5222   155.96435  0.00000000    0.000   505516.6 49653.38
## 13  -2523.426 10683.5697   155.13484  0.00000000    0.000   511665.4 49622.48
## 14 -25708.950 33938.7901   148.81580  0.00000000    0.000   561792.7 51861.92
## 15 -30448.814 37819.8946   148.97564  0.00000000 3842.816   572059.9 52276.60
## 16 -35317.350 41598.2592   149.44204 -0.04375656 7347.797   581916.1 52823.11
## 17 -35317.350 41598.2592    98.25656 -0.04375656 7347.797   581916.1 52823.11
##    condition    grade sqft_above sqft_basement  yr_built yr_renovated   zipcode
## 0      0.000     0.00   0.000000       0.00000     0.000     0.000000    0.0000
## 1      0.000     0.00   0.000000       0.00000     0.000     0.000000    0.0000
## 2      0.000 50266.47   0.000000       0.00000     0.000     0.000000    0.0000
## 3      0.000 50798.65   0.000000       0.00000     0.000     0.000000    0.0000
## 4      0.000 58843.55   0.000000       0.00000     0.000     0.000000    0.0000
## 5      0.000 65417.13   0.000000       0.00000     0.000     0.000000    0.0000
## 6      0.000 92489.08   0.000000       0.00000 -1406.038     0.000000    0.0000
## 7      0.000 93282.50   0.000000       0.00000 -1453.834     0.000000    0.0000
## 8   4012.793 96477.89   0.000000       0.00000 -1608.119     0.000000    0.0000
## 9   5695.167 97501.53   0.000000       0.00000 -1653.934     0.000000    0.0000
## 10  6083.851 97712.17   0.000000       0.00000 -1671.232     0.000000    0.0000
## 11  9454.220 99177.87   0.000000       0.00000 -1777.020     3.990969    0.0000
## 12 11239.728 99488.33   3.196060       0.00000 -1830.953     5.778407    0.0000
## 13 12712.236 99347.14   5.444775       0.00000 -1877.539     6.953849    0.0000
## 14 22079.074 97392.29  26.056178       0.00000 -2381.103    16.193259 -409.7930
## 15 24153.737 96783.69  28.254336       0.00000 -2502.141    17.891270 -497.5846
## 16 26173.243 96189.57  30.747966       0.00000 -2625.842    19.541826 -583.8375
## 17 26173.243 96189.57  81.933448      51.18548 -2625.842    19.541826 -583.8375
##          lat        long sqft_living15
## 0        0.0       0.000     0.0000000
## 1        0.0       0.000     0.0000000
## 2        0.0       0.000     0.0000000
## 3    10269.1       0.000     0.0000000
```

```
## 4  197850.9       0.000     0.0000000
## 5  345736.8       0.000     0.0000000
## 6  466327.0       0.000     0.0000000
## 7  470403.9       0.000     0.6567418
## 8  487841.8       0.000     3.3472708
## 9  494153.3   -9161.997     5.1497416
## 10 495602.3  -11221.662     5.6564257
## 11 506941.7  -27317.488     9.6642439
## 12 512908.8  -36320.325    11.2204316
## 13 516749.2  -43118.955    12.3745852
## 14 580756.2 -172792.997    17.8965509
## 15 592942.0 -198134.642    19.4441228
## 16 604042.5 -220318.566    20.7848368
## 17 604042.5 -220318.566    20.7848368
## attr(,"scaled:scale")
##  [1] 1.367286e+02 1.132218e+02 1.350202e+05 6.089238e+06 7.938388e+01
##  [6] 1.271891e+01 1.126565e+02 9.566587e+01 1.728044e+02 1.217378e+05
## [11] 6.506305e+04 4.318191e+03 5.905095e+04 7.865785e+03 2.037028e+01
## [16] 2.070320e+01 1.007595e+05
```

```
modelLasso$lambda
```

```
##  [1] 3.788985e+07 3.002010e+07 1.470878e+07 1.448142e+07 1.038049e+07
##  [6] 7.184651e+06 3.245943e+06 3.113564e+06 2.591841e+06 2.382277e+06
## [11] 2.330805e+06 1.944587e+06 1.766870e+06 1.633747e+06 4.749052e+05
## [16] 2.359090e+05 1.702676e-06
```

```
par(mfrow=c(1,2))
plot(modelLasso)
plot(c(modelLasso$lambda,0),pch=20,type="b",col="blue")
grid()
```



We observed that all variables are useful for model.