Applied Regression Final Project

Elda Piedade, Thomas Oswald

May 4th, 2020

Introduction

The objective of this project is to find the best linear regression model to predict the median home value for the houses in the Houston neighborhoods. Data was gathered from 96 zip codes in Houston by utilizing python web scrapping resources to collect data from the Texas Hometown Locator website (owned by HTL, Inc.).With the dataset extracted and cleaned, exploratory data analysis and statistical analysis were performed to understand the relationship between the median home value and other variables, such as diversity index, per capita income, and average household size. Based on the analysis, data was modeled with linear regression.

Importance

1. With a good model for prediction and analysis, individuals in Houston will be able to understand how to price their homes for sale.
2. Understanding how the demographic factors relate to median home value is valuable social knowledge.

Data

Response variable : Median Home Value Predictors:

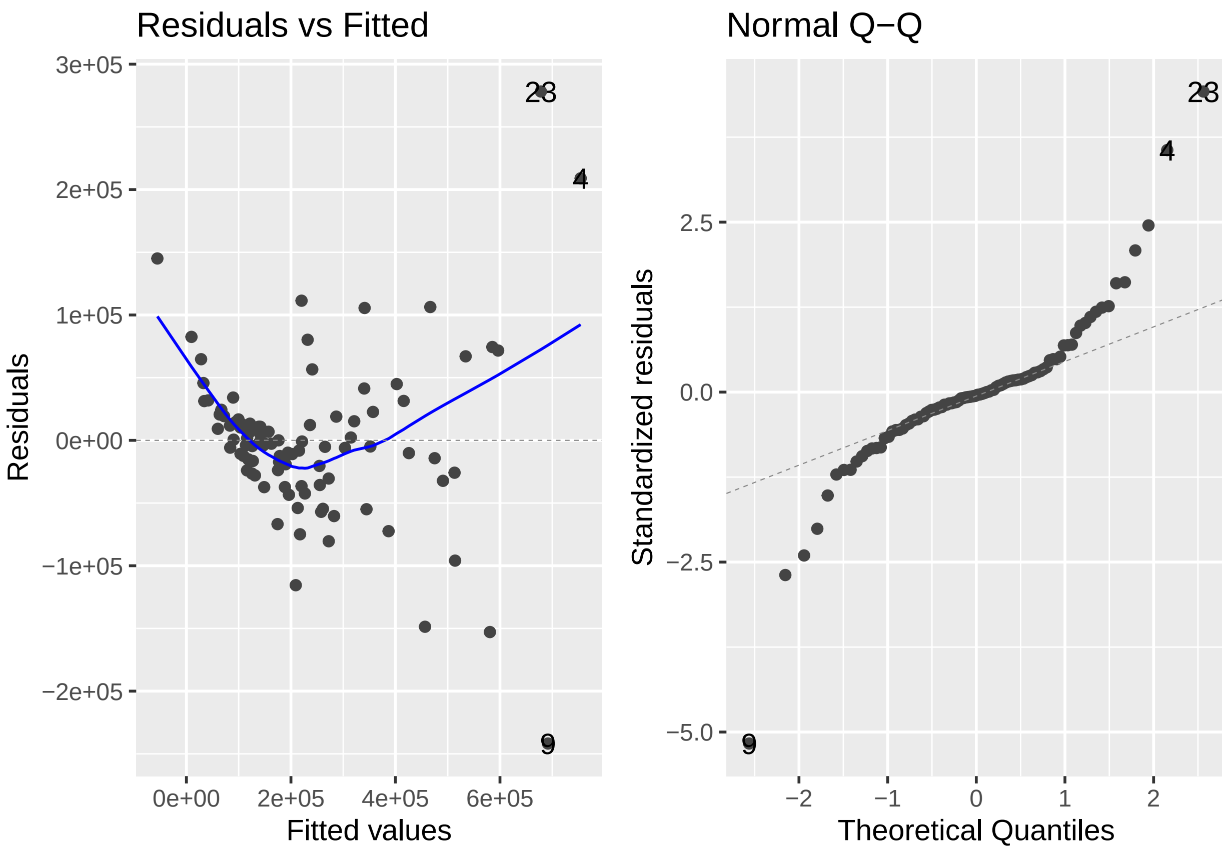
- x_1 -Total Population
- x_2 - Diversity Index
- Median Household Income
- Per Capita Income
- Total Housing Units
- Average Household Size
- Housing affordability Index

Data Loading & Checking full Model Accuracy:

After loeading the data and creating a full Linear Regression (LR) model, we found that the model is not adequate. The residuals have a tunel and bowl shape; the data is heavily-tailed distribution with three possible influencial points. To address this problem we will inspect the dat and perform a few transformations.

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7, data = reduce_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -241788  -25978   -3236   19032  278207
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  1.905e+05  1.112e+05   1.713 0.090259 .
## x1          -3.782e+00  1.724e+00  -2.193 0.030914 *
## x2          -2.845e+03  8.135e+02  -3.497 0.000740 ***
## x3           2.401e+00  7.015e-01   3.423 0.000943 ***
## x4           2.666e+00  1.152e+00   2.314 0.022996 *
## x5           1.099e+01  4.318e+00   2.546 0.012636 *
## x6           7.181e+04  2.827e+04   2.541 0.012821 *
## x7          -1.348e+03  2.093e+02  -6.439 6.14e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 68120 on 88 degrees of freedom
## Multiple R-squared:  0.8633, Adjusted R-squared:  0.8524
## F-statistic: 79.37 on 7 and 88 DF,  p-value: < 2.2e-16
```



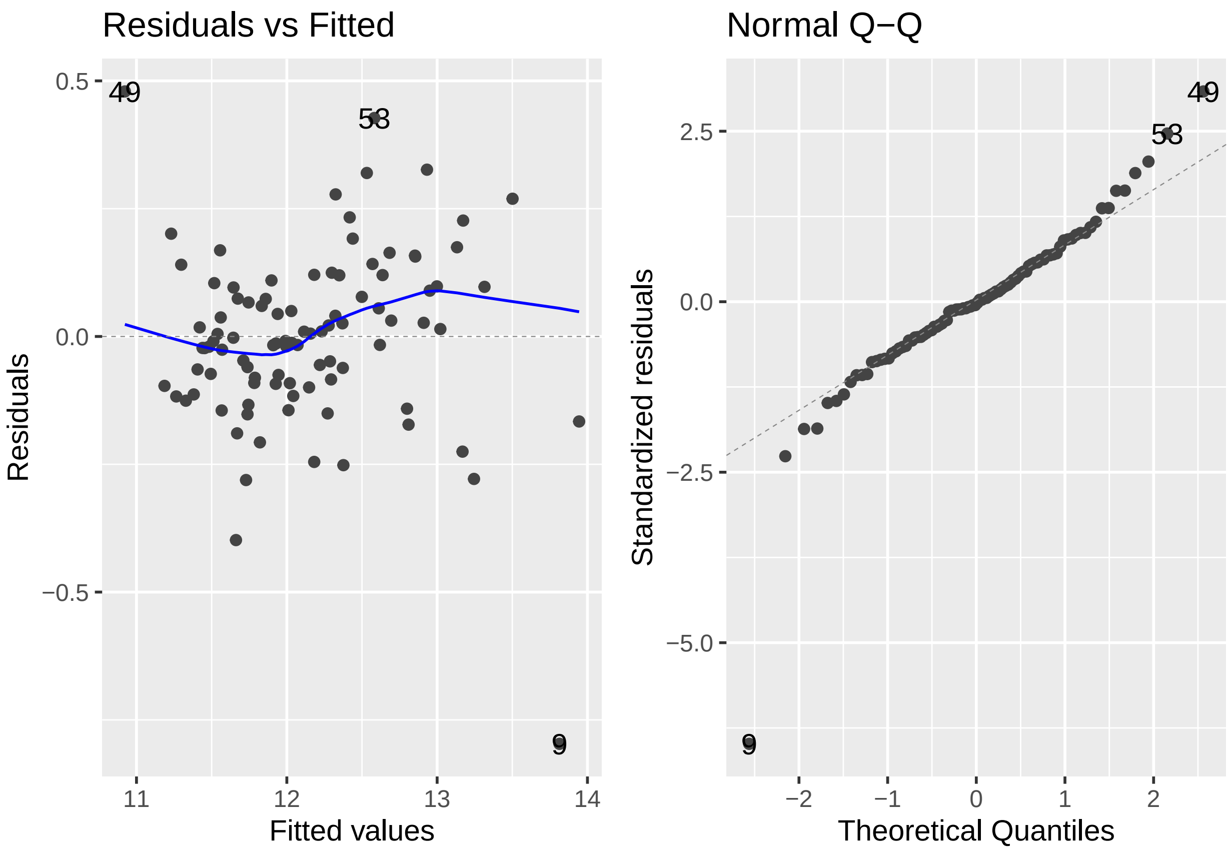
Data Transformation

Constant variance of erros assumptions can often be solve with response variable transformations. The log transformations performed the best compared to other transformations. It has the most appropriate properties for the normality of residuals and constant variance. The residual plot for the other transformation (reciprocal, square root, reciprocal square root and inverse) did not show much improvement from the original model and were very influenced by possible influencial points.

In conclusion, our best transformation is the " log" transformation. The residual plot does not appear to have any alarming shape, and the the residuals are normally distributed, except for the problematic observations, 9, 49 and 58. Our new transformed model indicate that a linear model provides a decent fit to the data.

```
y_recs <- y^(-1/2)
y_rec <- y^(-1)
```

```r
fit1 <- lm(sqrt(y)~x1+x2+x3+x4+x5+x6+x7,data=reduce_dat)
fit2 <- lm(log(y)~x1+x2+x3+x4+x5+x6+x7,data=reduce_dat)
fit3 <- lm(y_recs ~x1+x2+x3+x4+x5+x6+x7,data=reduce_dat)
fit4 <- lm(y_rec~x1+x2+x3+x4+x5+x6+x7,data=reduce_dat)
autoplot(fit2)[1:2]
```



Full regression model

```r
summary(lm(log(Median.Home.Value) ~., data = reduce_dat))
```

```
##
## Call:
## lm(formula = log(Median.Home.Value) ~ ., data = reduce_dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79685 -0.09169 -0.00563  0.09730  0.47930
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1.234e+01  2.922e-01  42.240  < 2e-16 ***
## Total.Population            -1.073e-05  4.531e-06  -2.369  0.02003 *
## Diversity.Index1            -2.442e-04  2.137e-03  -0.114  0.90931
## Median.Household.Income      1.481e-05  1.843e-06   8.034 3.95e-12 ***
## Per.Capita.Income           -2.328e-06  3.027e-06  -0.769  0.44398
## Total.Housing.Units          3.206e-05  1.135e-05   2.826  0.00584 **
## Average.Household.Size      -7.059e-02  7.427e-02  -0.950  0.34446
## Housing.Affordability.Index2 -6.052e-03  5.499e-04 -11.005  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.179 on 88 degrees of freedom
## Multiple R-squared:  0.927,  Adjusted R-squared:  0.9212
## F-statistic: 159.6 on 7 and 88 DF,  p-value: < 2.2e-16
```

Our linear regression model is significant given that the p-value for the F-test is smaller than our level of significance 0.05. Looking at the regressor individually, we found that the intercept, total population, median household income, total housing units and housing affordability index are signifacnt for predicting log(median home value). In model detail the F-test is perfomed to understand if at least one regressor is not equal to zero. The conclusion is supported by the t-test performed for each regressor. This is a first good step in our analysis and important to keep in mind.

Evaluating all possible subset regression models

For Variable selection it is a good practice to exclude problematic observations?

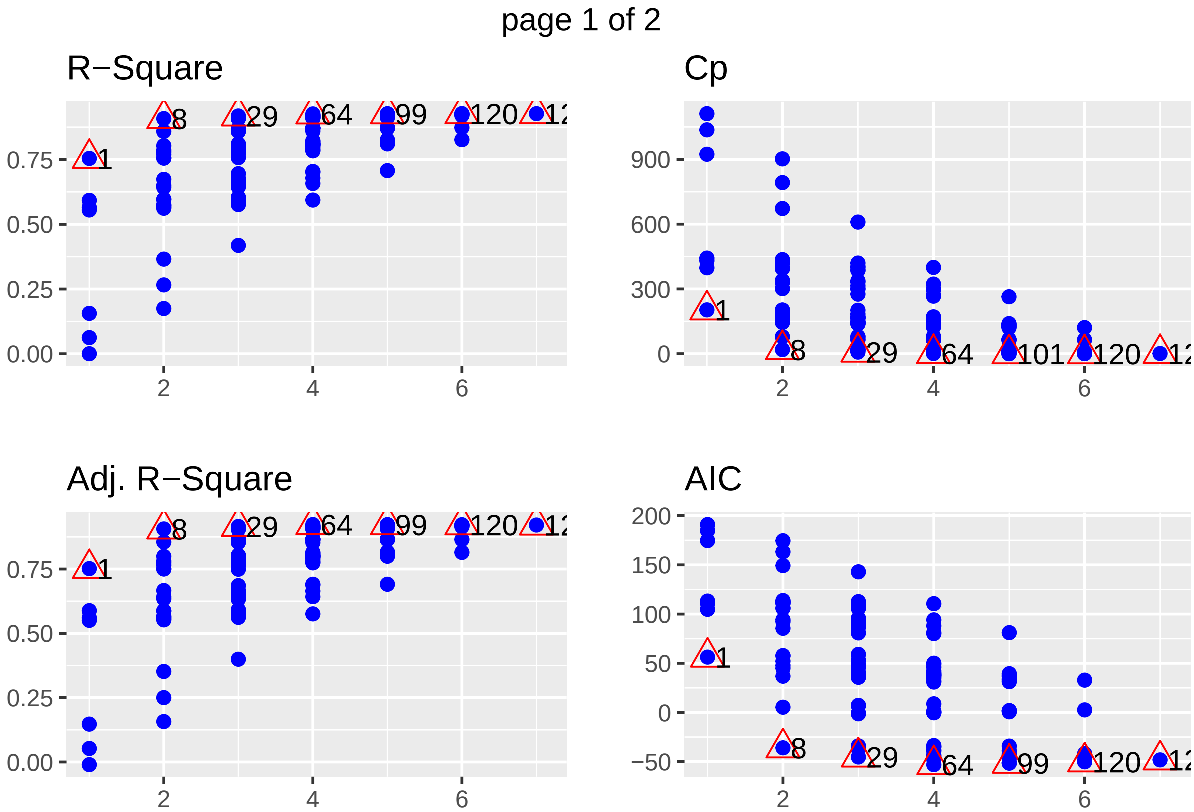
```
model <- lm(log(y)~x1+x2+x3+x4+x5+x6+x7,data=reduce_dat)
ols_step_all_possible(model)
```

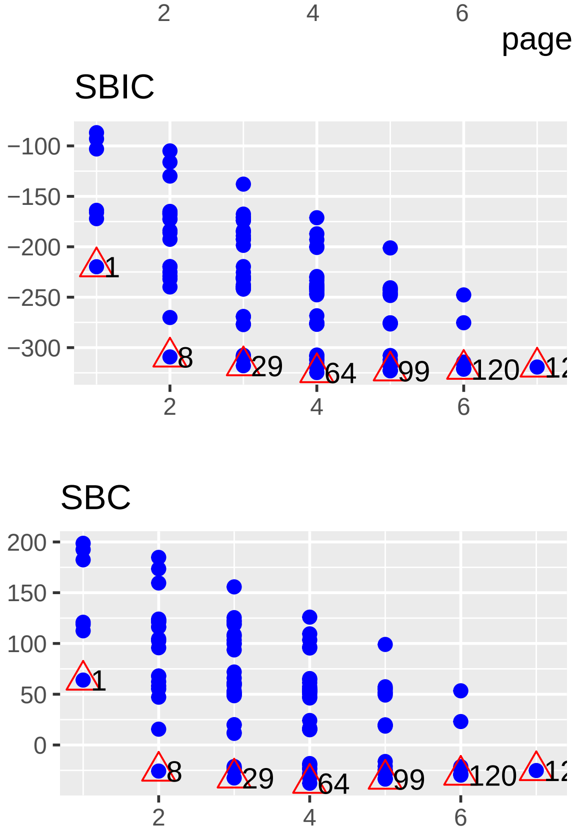
```
##      Index N           Predictors     R-Square Adj. R-Square Mallow's Cp
## 4        1 1                   x4 0.7542364063    0.75162190  204.108781
## 3        2 1                   x3 0.5924701930    0.58813477  399.013143
## 7        3 1                   x7 0.5650782657    0.56045144  432.016364
## 6        4 1                   x6 0.5552726240    0.55054148  443.830712
## 2        5 1                   x2 0.1561722533    0.14719536  924.687631
## 5        6 1                   x5 0.0624648472    0.05249107 1037.591196
## 1        7 1                   x1 0.0002556444   -0.01037993 1112.544084
## 22       8 2                x3 x7 0.9078934272    0.90591264   20.974797
## 25       9 2                x4 x7 0.8584101665    0.85536522   80.594808
## 23      10 2                x4 x5 0.8033616473    0.79913287  146.920132
## 10      11 2                x1 x4 0.7859433158    0.78133995  167.906645
## 21      12 2                x3 x6 0.7794151434    0.77467138  175.772127
## 24      13 2                x4 x6 0.7694915510    0.76453438  187.728588
## 19      14 2                x3 x4 0.7557829145    0.75053093  204.245468
## 15      15 2                x2 x4 0.7555720225    0.75031551  204.499562
## 28      16 2                x6 x7 0.6734735734    0.66645150  303.416050
## 20      17 2                x3 x5 0.6498043423    0.64227325  331.933973
## 18      18 2                x2 x7 0.6422929256    0.63460030  340.984119
## 9       19 2                x1 x3 0.5963018148    0.58762013  396.396605
## 14      20 2                x2 x3 0.5960531789    0.58736615  396.696175
## 12      21 2                x1 x6 0.5755959607    0.56646899  421.344097
## 26      22 2                x5 x6 0.5729201029    0.56373559  424.568110
## 13      23 2                x1 x7 0.5694229776    0.56016326  428.781629
## 27      24 2                x5 x7 0.5651292483    0.55577719  433.954938
## 17      25 2                x2 x6 0.5619026869    0.55248124  437.842467
## 11      26 2                x1 x5 0.3655416569    0.35189739  674.428466
## 16      27 2                x2 x5 0.2659316708    0.25014526  794.443766
## 8       28 2                x1 x2 0.1748089473    0.15706290  904.233171
## 59      29 3             x3 x6 x7 0.9181381613    0.91546875   10.631408
## 58      30 3             x3 x5 x7 0.9110100996    0.90810826   19.219668
## 56      31 3             x3 x4 x7 0.9099710977    0.90703537   20.471512
## 47      32 3             x2 x3 x7 0.9090908685    0.90612644   21.532058
## 37      33 3             x1 x3 x7 0.9079392531    0.90493727   22.919584
```

```
## 61     34 3          x4 x5 x7 0.8706665772    0.86644918   67.827646
## 40     35 3          x1 x4 x7 0.8696744040    0.86542466   69.023068
## 62     36 3          x4 x6 x7 0.8588963275    0.85429512   82.009056
## 50     37 3          x2 x4 x7 0.8586068748    0.85399623   82.357803
## 60     38 3          x4 x5 x6 0.8095801678    0.80337083  141.427734
## 38     39 3          x1 x4 x5 0.8090875353    0.80286213  142.021284
## 57     40 3          x3 x5 x6 0.8072723719    0.80098778  144.208287
## 54     41 3          x3 x4 x5 0.8054556396    0.79911180  146.397181
## 39     42 3          x1 x4 x6 0.8042949629    0.79791328  147.795625
## 48     43 3          x2 x4 x5 0.8036354994    0.79723231  148.590181
## 36     44 3          x1 x3 x6 0.8000652476    0.79354564  152.891806
## 55     45 3          x3 x4 x6 0.7882630617    0.78135860  167.111694
## 30     46 3          x1 x2 x4 0.7862567663    0.77928688  169.528984
## 34     47 3          x1 x3 x4 0.7860020001    0.77902380  169.835939
## 46     48 3          x2 x3 x6 0.7835354376    0.77647681  172.807782
## 35     49 3          x1 x3 x5 0.7828688819    0.77578852  173.610883
## 49     50 3          x2 x4 x6 0.7718814590    0.76444281  186.849103
## 44     51 3          x2 x3 x4 0.7572497802    0.74933401  204.478112
## 53     52 3          x2 x6 x7 0.6953948349    0.68546206  279.004172
## 43     53 3          x1 x6 x7 0.6758083190    0.66523685  302.603027
## 63     54 3          x5 x6 x7 0.6745596497    0.66394746  304.107489
## 45     55 3          x2 x3 x5 0.6630752064    0.65208853  317.944544
## 52     56 3          x2 x5 x7 0.6490395990    0.63759524  334.855375
## 33     57 3          x1 x2 x7 0.6441010751    0.63249568  340.805566
## 42     58 3          x1 x5 x7 0.6039923126    0.59107902  389.130693
## 29     59 3          x1 x2 x3 0.6039877214    0.59107428  389.136225
## 32     60 3          x1 x2 x6 0.5929897568    0.57971768  402.387145
## 51     61 3          x2 x5 x6 0.5886866362    0.57527424  407.571769
## 41     62 3          x1 x5 x6 0.5759467495    0.56211893  422.921449
## 31     63 3          x1 x2 x5 0.4184894128    0.39952711  612.634251
## 78     64 4       x1 x3 x5 x7 0.9260720441    0.92282246    3.072253
## 97     65 4       x3 x5 x6 x7 0.9218415588    0.91840602    8.169362
## 79     66 4       x1 x3 x6 x7 0.9198215845    0.91629726   10.603132
## 96     67 4       x3 x4 x6 x7 0.9190251992    0.91546587   11.562659
## 89     68 4       x2 x3 x6 x7 0.9181412352    0.91454305   12.627705
## 95     69 4       x3 x4 x5 x7 0.9147923395    0.91104695   16.662629
## 88     70 4       x2 x3 x5 x7 0.9135167739    0.90971531   18.199496
## 76     71 4       x1 x3 x4 x7 0.9108480443    0.90692928   21.414921
## 86     72 4       x2 x3 x4 x7 0.9104888316    0.90655427   21.847719
## 67     73 4       x1 x2 x3 x7 0.9095210145    0.90554392   23.013795
## 91     74 4       x2 x4 x5 x7 0.8719934267    0.86636676   68.228988
## 70     75 4       x1 x2 x4 x7 0.8716905012    0.86605052   68.593969
## 98     76 4       x4 x5 x6 x7 0.8714629191    0.86581294   68.868172
## 81     77 4       x1 x4 x5 x7 0.8707282546    0.86504598   69.753334
## 82     78 4       x1 x4 x6 x7 0.8696768774    0.86394839   71.020088
## 92     79 4       x2 x4 x6 x7 0.8592396013    0.85305233   83.595462
## 94     80 4       x3 x4 x5 x6 0.8226531351    0.81485767  127.676742
## 74     81 4       x1 x3 x4 x5 0.8202838442    0.81238423  130.531387
## 75     82 4       x1 x3 x4 x6 0.8155998124    0.80749431  136.174953
## 77     83 4       x1 x3 x5 x6 0.8123772614    0.80413011  140.057650
## 80     84 4       x1 x4 x5 x6 0.8103187239    0.80198109  142.537883
## 90     85 4       x2 x4 x5 x6 0.8095989894    0.80122971  143.405057
## 68     86 4       x1 x2 x4 x5 0.8090875844    0.80069583  144.021224
## 87     87 4       x2 x3 x5 x6 0.8075804835    0.79912248  145.837058
```

```
## 84     88 4          x2 x3 x4 x5 0.8056754284    0.79713369  148.132368
## 69     89 4          x1 x2 x4 x6 0.8043382098    0.79573769  149.743518
## 66     90 4          x1 x2 x3 x6 0.8004177327    0.79164489  154.467113
## 85     91 4          x2 x3 x4 x6 0.7926831234    0.78357029  163.786174
## 64     92 4          x1 x2 x3 x4 0.7862910820    0.77689728  171.487638
## 65     93 4          x1 x2 x3 x5 0.7835231737    0.77400771  174.822558
## 73     94 4          x1 x2 x6 x7 0.7038731299    0.69085656  270.789081
## 93     95 4          x2 x5 x6 x7 0.7006611610    0.68750341  274.659028
## 83     96 4          x1 x5 x6 x7 0.6783404416    0.66420156  301.552194
## 72     97 4          x1 x2 x5 x7 0.6575767439    0.64252517  326.569379
## 71     98 4          x1 x2 x5 x6 0.5934939269    0.57562553  403.779695
## 112    99 5       x1 x3 x5 x6 x7 0.9264587258    0.92237310    4.606359
## 110   100 5       x1 x3 x4 x5 x7 0.9261828611    0.92208191    4.938735
## 103   101 5       x1 x2 x3 x5 x7 0.9260982341    0.92199258    5.040698
## 117   102 5       x2 x3 x5 x6 x7 0.9221405315    0.91781501    9.809144
## 119   103 5       x3 x4 x5 x6 x7 0.9219981150    0.91766468    9.980735
## 111   104 5       x1 x3 x4 x6 x7 0.9201339585    0.91569696   12.226768
## 104   105 5       x1 x2 x3 x6 x7 0.9200243215    0.91558123   12.358864
## 116   106 5       x2 x3 x4 x6 x7 0.9190446044    0.91454708   13.539278
## 115   107 5       x2 x3 x4 x5 x7 0.9161461481    0.91148760   17.031489
## 101   108 5       x1 x2 x3 x4 x7 0.9118904826    0.90699551   22.158937
## 118   109 5       x2 x4 x5 x6 x7 0.8733200809    0.86628231   68.630566
## 106   110 5       x1 x2 x4 x5 x7 0.8723159360    0.86522238   69.840412
## 107   111 5       x1 x2 x4 x6 x7 0.8717565887    0.86463195   70.514343
## 113   112 5       x1 x4 x5 x6 x7 0.8717060840    0.86457864   70.575194
## 109   113 5       x1 x3 x4 x5 x6 0.8257438106    0.81606291  125.952935
## 114   114 5       x2 x3 x4 x5 x6 0.8228618050    0.81302079  129.425326
## 99    115 5       x1 x2 x3 x4 x5 0.8207556230    0.81079760  131.962964
## 100   116 5       x1 x2 x3 x4 x6 0.8157829531    0.80554867  137.954295
## 102   117 5       x1 x2 x3 x5 x6 0.8134404428    0.80307602  140.776674
## 105   118 5       x1 x2 x4 x5 x6 0.8103187239    0.79978088  144.537883
## 108   119 5       x1 x2 x5 x6 x7 0.7071187898    0.69084761  268.878541
## 125   120 6    x1 x3 x4 x5 x6 x7 0.9269511582    0.92202652    6.013050
## 123   121 6    x1 x2 x3 x5 x6 x7 0.9264712411    0.92151425    6.591279
## 121   122 6    x1 x2 x3 x4 x5 x7 0.9262121491    0.92123769    6.903447
## 126   123 6    x2 x3 x4 x5 x6 x7 0.9223048264    0.91706695   11.611193
## 122   124 6    x1 x2 x3 x4 x6 x7 0.9203348109    0.91496412   13.984770
## 124   125 6    x1 x2 x4 x5 x6 x7 0.8733910206    0.86485558   70.545094
## 120   126 6    x1 x2 x3 x4 x5 x6 0.8264459254    0.81474565  127.106991
## 127   127 7 x1 x2 x3 x4 x5 x6 x7 0.9269619897    0.92115215    8.000000
```

```
plot(ols_step_all_possible(model))
```





In looking for the "best" model, certain criteria must be met inorder for proper variable selection of the regressor equation. These criteria help us to be able to explain the data in the simpliest way with redundant predictors removed inorder minimize cost and to avoid multi-collinearity in our regression model.

The criteria for our variable selection include: 1) Large R^2 value 2) Maximum Adjusted R^2 value 3) Minimum MSres 4) Minimum Mallow's Cp Statistic value

Based on the above criteria, the "best" candidate models are:

1) Model 1: y ~ x4
2) Model 8: y ~ x3 + x7
3) Model 29: y ~ x3 + x6 + x7
4) Model 64: y ~ x1 + x3 + x5 + x7
5) Model 99: y ~ x1 + x3 + x5 + x6 + x7
6) Model 120:y ~ x1 + x3 + x4 + x5 + x6 + x7
7) Model 127:y ~.

Once we identified the "best"" candidate models, we compare its predicted residual error sum of squares (PRESS) statistic with other candidate models and selected the model with the smallest value. We also compare candidate models by performing a variance inflation factor (VIF) in order to quantify the severity of multicollinearity in the model.

```
fit1 <- lm(log(y)~x4,data=reduce_dat)
fit8 <- lm(log(y)~x3+x7,data=reduce_dat)
fit29 <- lm(log(y)~x3+x6+x7,data=reduce_dat)
fit64 <- lm(log(y)~x1+x3+x5+x7,data=reduce_dat)
fit99 <- lm(log(y)~x1+x3+x5+x6+x7,data=reduce_dat)
fit120 <- lm(log(y)~x1+x3+x4+x5+x6+x7,data=reduce_dat)
fit127 <- lm(log(y)~x1+x2+x3+x4+x5+x6+x7,data=reduce_dat)

PRESS(fit1)
```

```
## [1] 10.73345
```

```
PRESS(fit8)
```

```
## [1] 4.235565
```

```
PRESS(fit29)
```

```
## [1] 3.813863
```

```
PRESS(fit64)
```

```
## [1] 3.760315
```

```
PRESS(fit99)
```

```
## [1] 3.97833
```

```
PRESS(fit120)
```

```
## [1] 4.969797
```

```
PRESS(fit127)
```

```
## [1] 5.792131
```

```
vif(fit8)
```

```
##       x3       x7
## 1.081888 1.081888
```

```
vif(fit29)
```

```
##       x3       x6       x7
## 1.293148 2.136140 1.793435
```

```r
vif(fit64)
```

```
##       x1       x3       x5       x7 
## 7.539310 1.107578 8.311564 1.636704 
```

```r
vif(fit99)
```

```
##        x1        x3        x5        x6        x7 
## 16.681309  1.388036 16.575790  4.737732  1.960617 
```

```r
vif(fit120)
```

```
##        x1        x3        x4        x5        x6        x7 
## 17.011191  7.448650 14.010788 16.626136  6.039030  2.339524 
```

```r
vif(fit127)
```

```
##        x1        x2        x3        x4        x5        x6        x7 
## 17.742753  1.550252  7.646411 14.012521 16.921586  6.074919  2.360197 
```

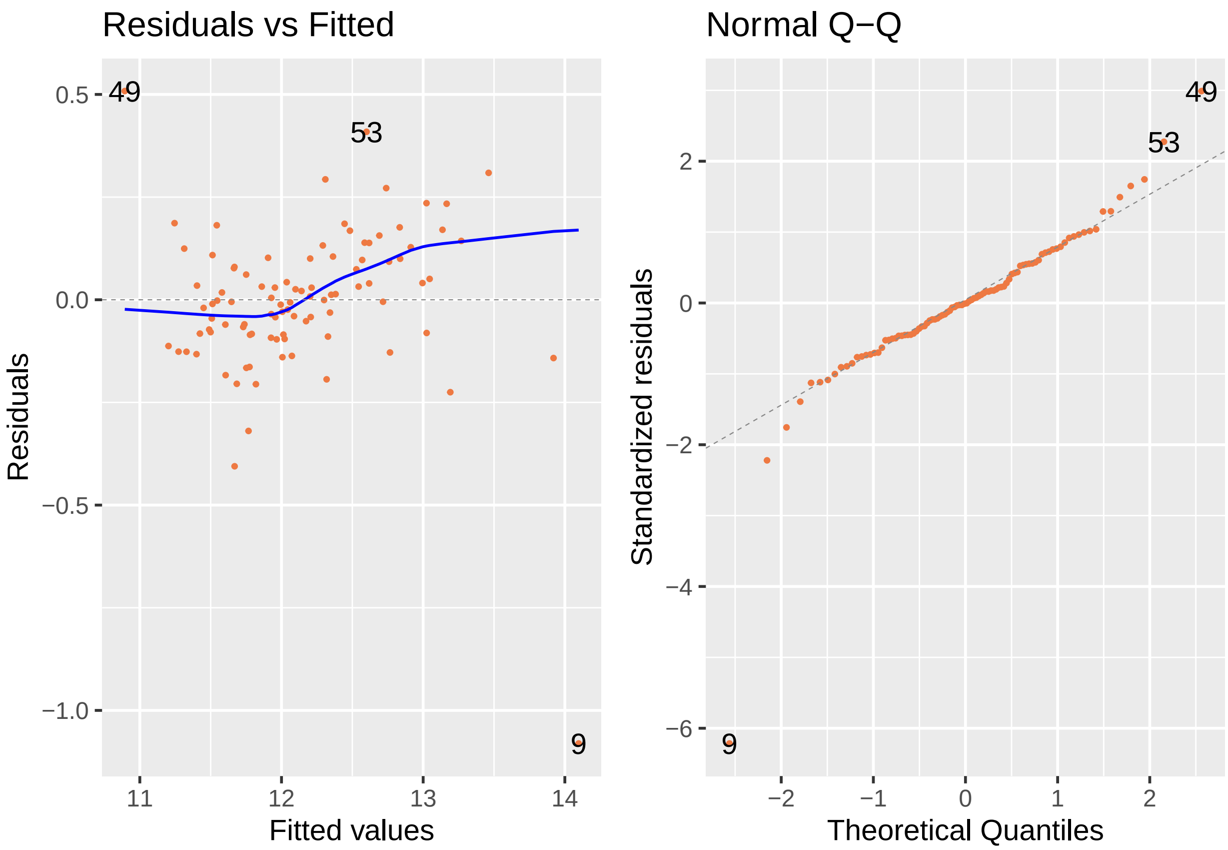
Interpretation of PRESS and Vif of candidate models:

The model with the lowest PRESS value is model 64 [log(y) ~ x2 + x3 + x6 + x7] however there is evidence of multicolliniearity.

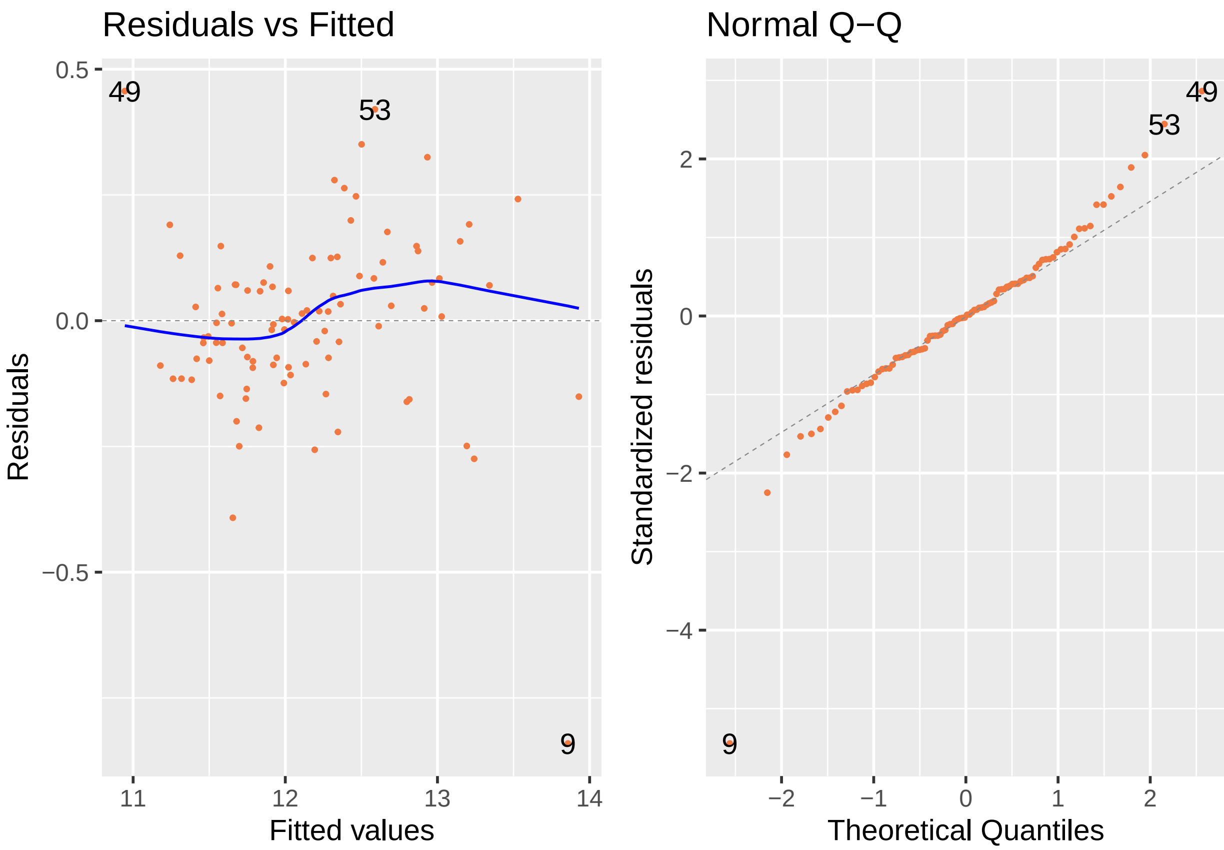
The model with the second lowest PRESS value is model 29 [log(y) ~ x3 + x6 + x7] and the same model doesnt show any evidence of multicolinearity in the variance inflation factor test of each regressor.

Plot of model:

```r
autoplot(fit29,size = 0.5, colour = 'sienna2')[1:2]
```




```r
autoplot(fit64,size = 0.5, colour = 'sienna2')[1:2]
```



Residuals vs Fitted
0.5
0.0
−0.5
Residuals
49
53
9
11
12
13
14
Fitted values
Normal Q−Q
2
0
−2
−4
Standardized residuals
49
53
9
−2
−1
0
1
2
Theoretical Quantiles