STAT 5120 Final Project

Shaoran Sun

May 8, 2016

Contents

1 Introduction

This report is to build and select a model for the response y, using the generated data set "dat.cvs" given by the course instructor. Most of the tools studied in course STAT 5120 (Linear Regression) will be used to facilitate model selection, model estimation, model validation, and model diagnosis procedures. This report is written by Shaoran Sun individually.

2 Materials and methods

Data is given by the instructor. Data contains one response variable, y, and 10 predictors, $x_1, x_2, \ldots, x_{10}$. There are 100 intances of data, in total of 100 rows. See the data on Appendix I

The analysis method used are:

- Linear Transformation
- Detecting Outliers in Predictors with leverages
- Detecting Infuential Observations with DFFITS and Cook's Distance
- Automated Forward Selection Procedure
- Diagnosing Multicollinearity with Variance Inflation Factor (VIF)
- General Linear F Test

The criteria for statistical significance is $P < 0.05$.

3 Results

With the given data set "dat.cvs",

- y is log-transformed
- Entry 10 is reomoved due to large influence to the overall data
- Predictors x1, x6, x7, x8, x9, and x10 are *removed* due to insignificance.
- Predictors x2, x3, x4,and x5 are kept, and combined as a first order model.

The final model is:

$$\mathbf{y = 2.00 \cdot x_2 + 0.91 \cdot x_3 + 0.40 \cdot x_4 + 3.35 \cdot x_5 + 1.51}$$

```
Call:
lm(formula = y ~ x2 + x3 + x4 + x5)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.29129 -0.46534  0.00512  0.62212  1.87873 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  1.50163    0.09635  15.584  < 2e-16 ***
x2           1.99583    0.08829  22.605  < 2e-16 ***
x3           0.91006    0.10112   8.999 2.46e-14 ***
x4           0.40358    0.08986   4.491 2.01e-05 ***
x5           3.34805    0.09792  34.191  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9338 on 94 degrees of freedom
Multiple R-squared:  0.955,	Adjusted R-squared:  0.9531 
F-statistic: 498.6 on 4 and 94 DF,  p-value: < 2.2e-16
```

The intercept, x2, and x5 all have p-values of $(< 2e - 16)$, x3 has p-value of $2.46e - 14$, and x4 has p-value of $2.01e - 05$. The overall p-value is $(< 2.2e - 16)$. These p-values are all less than 0.05 and significant.

The R^2 is 0.955, which means the model represents 95.5% of the data.

4 Detailed Procedures

4.1 Read in data and fit the originial data to a model

I first plot in all 10 variables in responsing to y. All the p-values to the variables are greater than 0.05, which indicates that none of the variable is significant.

The 5 assumptions about linear regression are:

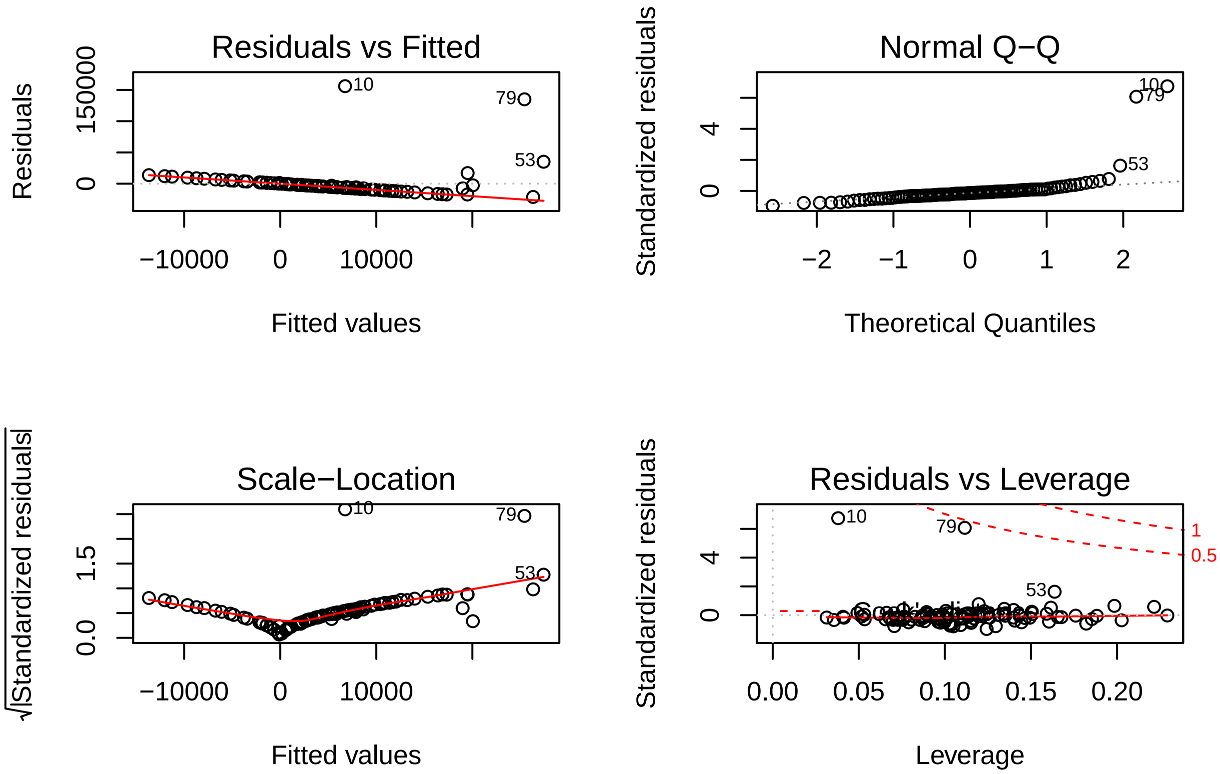
1. There exists a linear relation between the response and predictor variable(s).
2. The error terms have the constant variance
3. The error terms are independent, have mean 0.
4. Model fits all observations well (no outliers).
5. The errors follow a Normal distribution.

The Residuals vs Fitted graph suggested that residuals do not fall in a horizontal band around 0, and they have an apparent pattern. Assumption 1 and 3 are not met.

The residuals also do not have similar vertical variation across fits. Assumption 2 is not met.

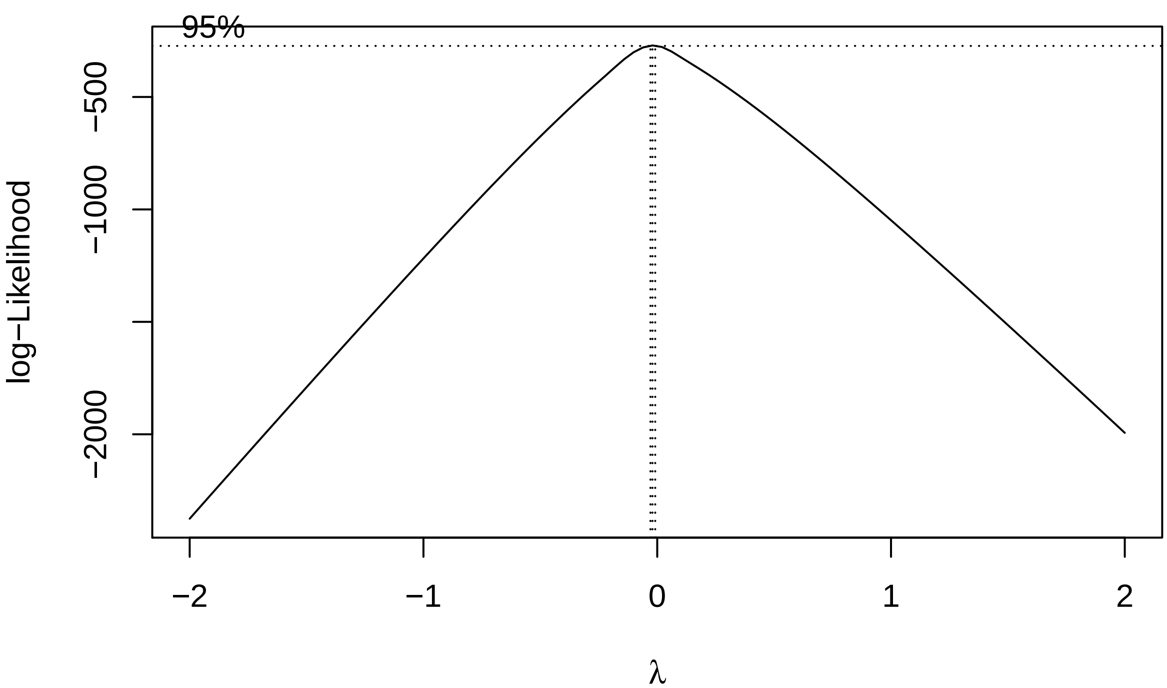
Hence, we consider using a boxcox transformation to test for log-likelihood.

```
##
## Call:
## lm(formula = dat$y ~ ., data = dat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -21213  -7359  -3119   1056 156003
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4856.9     2554.0   1.902   0.0604 .
## x1           18253.4    50397.7   0.362   0.7181
## x2          -16220.4    50336.8  -0.322   0.7480
## x3            -723.3     2565.5  -0.282   0.7786
## x4             520.9     2317.8   0.225   0.8227
## x5            4563.1     2562.1   1.781   0.0783 .
## x6           -2807.6     2457.2  -1.143   0.2563
## x7           -2535.7     2810.4  -0.902   0.3694
## x8            2839.7     2348.2   1.209   0.2297
## x9           -3654.9     2392.7  -1.527   0.1302
## x10           -577.3     2607.0  -0.221   0.8253
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23590 on 89 degrees of freedom
## Multiple R-squared:  0.1113, Adjusted R-squared:  0.01149
## F-statistic: 1.115 on 10 and 89 DF,  p-value: 0.36
```



4.2 Boxcox transformation

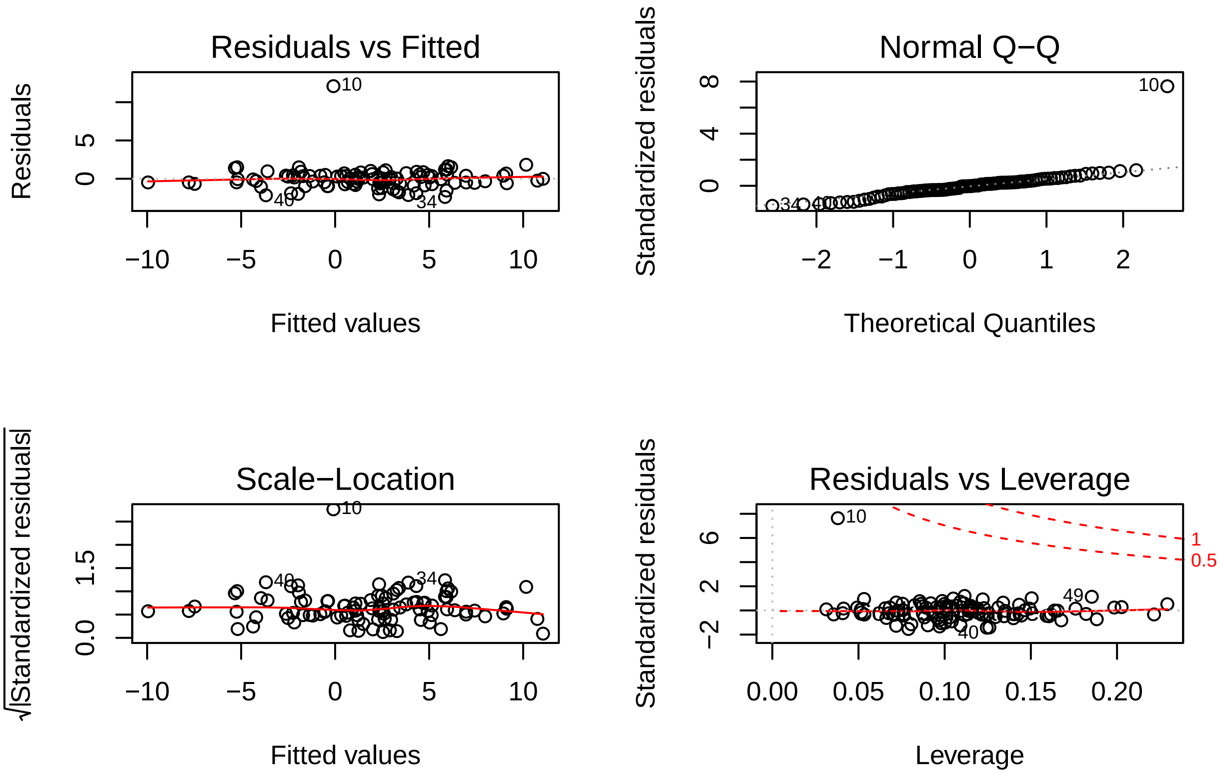
As we can see from the boxcox graph, λ is very close to 0. So we consider a log-transformation.



4.3 Fit the log-transformed y

After the transformation, some of the predictors are starting to become significant, namely, x3, x5, and intercept. The overall p-value also becomes very significant, comparing to before the transformaiton, where p-value $= 0.36$.

```
##
## Call:
## lm(formula = dat$y ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3787 -0.6130 -0.0369  0.4978 12.0928
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.610750   0.174805   9.215 1.35e-14 ***
## x1          -1.323510   3.449464  -0.384   0.7021
## x2           3.262825   3.445294   0.947   0.3462
## x3           0.945029   0.175598   5.382 5.92e-07 ***
## x4           0.349093   0.158641   2.201   0.0304 *
## x5           3.261169   0.175364  18.597  < 2e-16 ***
## x6          -0.104953   0.168182  -0.624   0.5342
## x7           0.001392   0.192358   0.007   0.9942
## x8           0.086216   0.160720   0.536   0.5930
## x9          -0.066157   0.163769  -0.404   0.6872
## x10          0.080254   0.178437   0.450   0.6540
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.614 on 89 degrees of freedom
## Multiple R-squared:  0.8795, Adjusted R-squared:  0.866
## F-statistic: 64.97 on 10 and 89 DF,  p-value: < 2.2e-16
```



4.4 Detect and remove outliers

Next, I will detect outliers.

From the above graphs, we can see that outliers definitely exist, possibly, data 10.

Obtain leverages and two measures that can be used to identify influential points, DFFITS (difference in fts) and Cook's Distance.

From the output of influence, DFFITS (difference in fts), and Cook's Distance, we can see that observation 10 is indeed an influential outlier. Hence we remove it.

After removing observation 10, the model fits the data better than before, as we can see in the following output table. Next, we will be choosing which predictors are actually significant.

```
##
## Call:
## lm(formula = no10$y ~ ., data = no10)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3706 -0.4279  0.0241  0.5760  1.8382
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.482703   0.103688  14.300  < 2e-16 ***
## x1          0.692642   2.042699   0.339    0.735
## x2          1.295552   2.039959   0.635    0.527
## x3          0.909189   0.103719   8.766 1.25e-13 ***
## x4          0.407380   0.093778   4.344 3.73e-05 ***
```

```
## x5           3.322817   0.103653  32.057  < 2e-16 ***
## x6          -0.007948   0.099586  -0.080    0.937    
## x7           0.130916   0.114019   1.148    0.254    
## x8           0.070550   0.094905   0.743    0.459    
## x9           0.027593   0.096969   0.285    0.777    
## x10          0.052156   0.105381   0.495    0.622    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.9533 on 88 degrees of freedom
## Multiple R-squared:  0.9561, Adjusted R-squared:  0.9511 
## F-statistic: 191.6 on 10 and 88 DF,  p-value: < 2.2e-16
```

4.5 Check for multicolinearity issue

```
##          y     x1     x2     x3     x4     x5     x6     x7     x8     x9
## y    1.000  0.554  0.555  0.324  0.042  0.808 -0.015  0.141  0.205 -0.096
## x1   0.554  1.000  0.999  0.060 -0.020  0.063  0.001 -0.003  0.077 -0.085
## x2   0.555  0.999  1.000  0.060 -0.020  0.064  0.003  0.008  0.079 -0.088
## x3   0.324  0.060  0.060  1.000 -0.003  0.126  0.028  0.063  0.017 -0.007
## x4   0.042 -0.020 -0.020 -0.003  1.000 -0.061  0.000  0.071 -0.144  0.085
## x5   0.808  0.063  0.064  0.126 -0.061  1.000 -0.030  0.125  0.211 -0.087
## x6  -0.015  0.001  0.003  0.028  0.000 -0.030  1.000  0.034  0.109 -0.166
## x7   0.141 -0.003  0.008  0.063  0.071  0.125  0.034  1.000  0.019 -0.068
## x8   0.205  0.077  0.079  0.017 -0.144  0.211  0.109  0.019  1.000  0.044
## x9  -0.096 -0.085 -0.088 -0.007  0.085 -0.087 -0.166 -0.068  0.044  1.000
## x10  0.033  0.158  0.160 -0.069 -0.112 -0.042  0.100 -0.051  0.078 -0.058
##        x10
## y    0.033
## x1   0.158
## x2   0.160
## x3  -0.069
## x4  -0.112
## x5  -0.042
## x6   0.100
## x7  -0.051
## x8   0.078
## x9  -0.058
## x10  1.000
```

```
##         x1         x2         x3         x4         x5         x6 
## 515.180302 515.882996   1.028513   1.049128   1.099961   1.060420 
##         x7         x8         x9        x10 
##   1.095493   1.101787   1.069866   1.068766
```

We can see that x1 and x2 are highly correlated with 0.999 correlation, and 515 VIFs, which is way greater than 10, the threshold. We will next apply automated search to search for significant predictors. If x1 and x2 are both in the result, we will remove one of them in the final predictors.

4.6 Use automated search

Now I use automated search, after removing data 10 from the data set.

```
##
## Call:
## lm(formula = y ~ x5 + x2 + x3 + x4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.29129 -0.46534  0.00512  0.62212  1.87873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.50163    0.09635  15.584  < 2e-16 ***
## x5           3.34805    0.09792  34.191  < 2e-16 ***
## x2           1.99583    0.08829  22.605  < 2e-16 ***
## x3           0.91006    0.10112   8.999 2.46e-14 ***
## x4           0.40358    0.08986   4.491 2.01e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9338 on 94 degrees of freedom
## Multiple R-squared:  0.955,  Adjusted R-squared:  0.9531
## F-statistic: 498.6 on 4 and 94 DF,  p-value: < 2.2e-16
```

Using automated forward search, we get x2, x3, x4, and x5 are chosen, among which all are significant with p-values way less than 0.05. The over all fitting of the 4 predictors results in a p-value of $<$ 2.2e-16. This proves that after log-transformation, removing data 10 and automated forward selection, the model fits better. x1 and x2 are not both in the result, so there is no multicolinearity issue.

4.7 Model Comparison and Selection

Next, we will check use techniques in Model Comparison and Selection, by comparing R2.adj, PRESS, AIC, BIC, and Cp. We will be using data after removing outlier and log-transformation.

After comparing all permutation's R2.adj, PRESS, AIC, BIC, and Cp. We have the following results:

Permutation Number	Criteria	Result Predictors	p-value	R-squared
489	R2.adj	x2+x3+x4+x5+x7	$<$ 2.2e-16	0.9556
481	PRESS, AIC, BIC, Cp	x2+x3+x4+x5	$<$ 2.2e-16	0.9550

4.8 General F-test

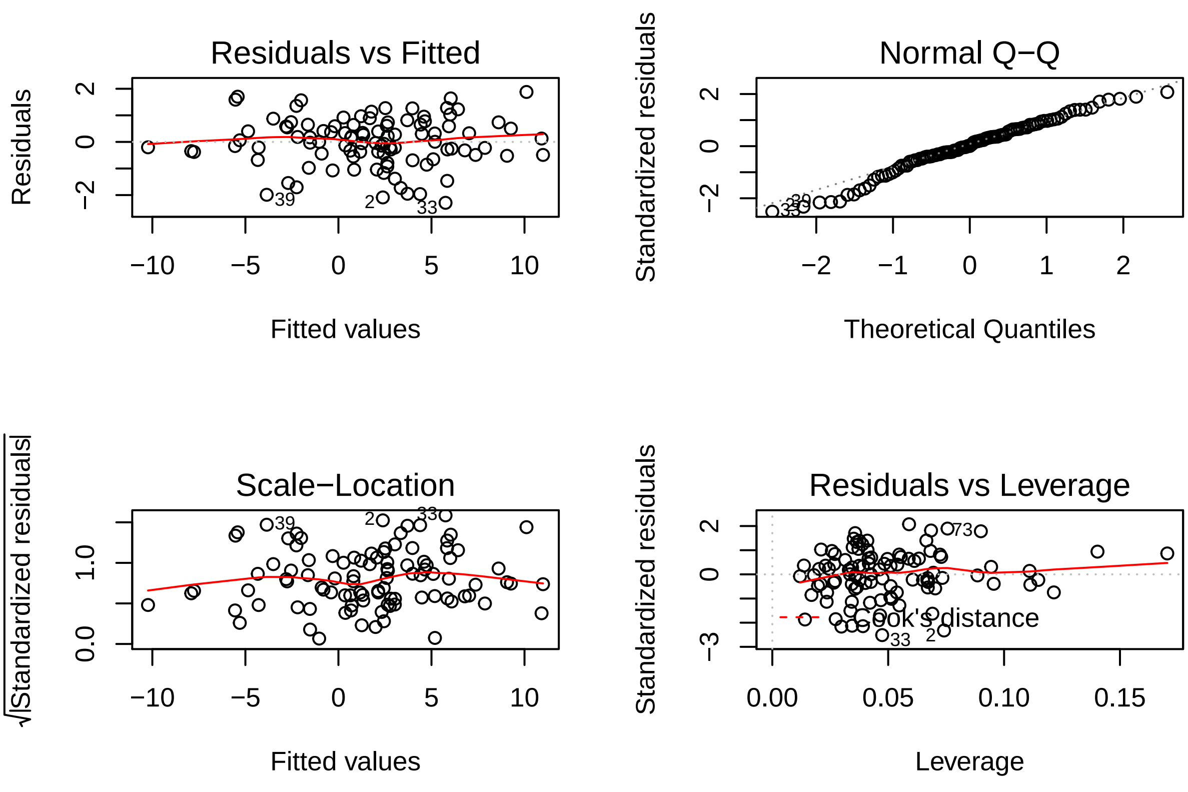
Model 481 is a reduced model of 489, with an extra x7. We apply a general F-test to test if x7 is significant.

```
## Analysis of Variance Table
##
## Model 1: y ~ x2 + x3 + x4 + x5
## Model 2: y ~ x2 + x3 + x4 + x5 + x7
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     94 81.960
## 2     93 80.929  1    1.0317 1.1856  0.279
```

p-value is greater than 0.05. Hence, x7 is not significant, we only need x2, x3, x4, and x5.

Finally, fit our final model again, and test for goodness of fit.

```
##
## Call:
## lm(formula = y ~ x2 + x3 + x4 + x5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.29129 -0.46534  0.00512  0.62212  1.87873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.50163    0.09635  15.584  < 2e-16 ***
## x2           1.99583    0.08829  22.605  < 2e-16 ***
## x3           0.91006    0.10112   8.999 2.46e-14 ***
## x4           0.40358    0.08986   4.491 2.01e-05 ***
## x5           3.34805    0.09792  34.191  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9338 on 94 degrees of freedom
## Multiple R-squared:  0.955,  Adjusted R-squared:  0.9531
## F-statistic: 498.6 on 4 and 94 DF,  p-value: < 2.2e-16
```



5 Conclusion

With the given data set "dat.cvs",

- y is log-transformed
- Entry 10 is reomoved due to large influence to the overall data
- Predictors x1, x6, x7, x8, x9, and x10 are *removed* due to insignificance.
- Predictors x2, x3, x4,and x5 are kept, and combined as a first order model.

The final model is:

$$\mathbf{y = 2.00 \cdot x_2 + 0.91 \cdot x_3 + 0.40 \cdot x_4 + 3.35 \cdot x_5 + 1.51}$$

```
##
## Call:
## lm(formula = y ~ x2 + x3 + x4 + x5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.29129 -0.46534  0.00512  0.62212  1.87873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.50163    0.09635  15.584  < 2e-16 ***
## x2           1.99583    0.08829  22.605  < 2e-16 ***
## x3           0.91006    0.10112   8.999 2.46e-14 ***
## x4           0.40358    0.08986   4.491 2.01e-05 ***
## x5           3.34805    0.09792  34.191  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9338 on 94 degrees of freedom
## Multiple R-squared:  0.955,  Adjusted R-squared:  0.9531
## F-statistic: 498.6 on 4 and 94 DF,  p-value: < 2.2e-16
```

Predictor	P-value
intercept	$(< 2e - 16)$
x2	$(< 2e - 16)$
x3	$2.46e - 14$
x4	$2.01e - 05$
x5	$(< 2e - 16)$
overall	$(< 2.2e - 16)$

The intercept, x2, and x5 all have p-values of $(< 2e - 16)$, x3 has p-value of $2.46e - 14$, and x4 has p-value of $2.01e - 05$. The overall p-value is $(< 2.2e - 16)$. These p-values are all less than 0.05 and significant.

The R^2 is 0.955, which means the model represents 95.5% of the data. The model fits the data very well.

6 Appendix

6.1 Data

```
##               y          x1           x2          x3          x4
## 1  2.436641e+02  1.08580359  1.114076637  0.45462375  1.69147892
## 2  1.348317e+00  1.30488896  1.311393972 -2.01623853 -0.14386029
## 3  2.785373e+00  1.03064839  1.032495758  1.02785284 -0.30580807
## 4  4.870431e+01  0.15783723  0.115275934 -0.12345028  1.71466970
## 5  5.107077e+03  1.32553541  1.270779866 -0.47953329 -1.26886600
## 6  2.461621e+01  0.17907553  0.169041211  1.10933527 -1.54807478
## 7  4.053033e-01 -1.15115545 -1.158052862 -0.26611181 -1.04691094
## 8  1.273015e+00  0.34749784  0.413766183 -1.14806766  2.07995485
## 9  1.750351e+04  1.09237146  1.068349232 -0.31490694  0.78322136
## 10 1.627548e+05 -0.12377383 -0.101591919  0.14495762 -0.80827415
## 11 1.142597e+04  3.62769114  3.608260193  0.08461995 -2.67794463
## 12 3.286586e+00  1.30989523  1.307003554  0.14383078 -0.93232006
## 13 1.317835e+01  0.75371880  0.861374750 -0.47295516  0.72538498
## 14 7.994837e+01  1.08365898  1.057540228  0.40165760 -2.39403767
## 15 1.254888e+01  0.36798425  0.355010226 -0.91801520 -1.46354593
## 16 6.475994e+02  0.56864183  0.640659644  0.54047899  0.18522984
## 17 1.883914e+02 -1.28581387 -1.212272920  1.48905504 -0.41666222
## 18 1.435408e-02 -1.26743305 -1.234753509  0.97972598 -0.19470807
## 19 2.152715e+03 -1.06712497 -1.084466308  0.53220005  0.22177933
## 20 2.748900e+01 -1.38924287 -1.443990181  2.09600223 -0.67593870
## 21 4.269482e+00  1.54605968  1.601931488 -0.23083356  0.04281459
## 22 9.142846e+01  1.40197875  1.357894118  2.57464218 -2.13216101
## 23 1.362112e+00 -1.32641841 -1.315556541 -0.46236821 -0.07518034
## 24 1.557144e+03  0.35885165  0.341494903  0.29181328  0.73709888
## 25 2.503421e-01 -1.13524440 -1.129708685 -0.07397919  1.20814059
## 26 5.816126e+00 -0.12596809 -0.143102696  1.09787376  0.84864383
## 27 6.741841e-01 -1.08959331 -1.148468804 -1.00854906  0.32894141
## 28 2.615835e-01 -0.80121686 -0.768502037 -0.26400286 -1.98235826
## 29 3.706270e-01  1.65612148  1.622377304 -0.43629692 -1.22394032
## 30 4.953901e+00  0.15058629  0.135064308 -1.33142033  0.03848109
## 31 2.597080e-04 -2.53107104 -2.505959283 -1.30320557  0.00049758
## 32 3.564594e+00 -0.18153431 -0.253450070  1.17657303 -1.85181932
## 33 1.081779e-01 -1.38348270 -1.408735905 -1.22537602 -0.27192656
## 34 3.198690e+01  1.88159981  1.874747994  0.43157478  0.91527663
## 35 1.989805e+00 -0.60581127 -0.592719513 -0.62328645  0.30325301
## 36 5.305765e-03  0.58361327  0.528917046 -0.18197329  0.28060109
## 37 7.083158e+00 -0.84461949 -0.883527854  1.91593170  1.52952326
## 38 9.732534e-01 -0.72132166 -0.721717588  0.65413664  1.66975546
## 39 8.182734e-01  0.25261865  0.283697303 -0.61464193  0.79174582
## 40 2.902513e-03 -1.03292033 -0.980865674 -0.28024369 -0.23159184
## 41 1.257483e+01 -0.67510456 -0.666402448  0.52190114 -0.77282839
## 42 1.649000e-01 -0.39816292 -0.410390828 -1.68731670 -1.04893744
## 43 7.227816e-02  0.93858479  0.886642035 -1.51878437  0.32664528
## 44 2.300922e+00  0.20579738  0.240324984  1.10446829 -0.85297338
## 45 1.840516e+01  0.23249175  0.238624787 -0.27242598  1.58819474
## 46 5.036520e+00  0.07543418 -0.009001722  0.23625602 -0.15096018
## 47 1.587850e+02 -1.11552973 -1.183864277  0.71806593 -0.10832554
## 48 2.266285e+02  0.12797868  0.197831749 -0.10367126  0.81746507
## 49 2.103047e+03  0.47872975  0.407202656 -0.43740217 -0.11192762
```

```
## 50  1.132980e+01  1.03329881  0.945368691 -0.99770994 -0.80953017
## 51  3.306079e-03 -0.75883165 -0.698295514 -0.26168555 -1.01764340
## 52  1.071461e+01  0.50535547  0.562841300 -0.15480350 -0.08928677
## 53  6.252071e+04  1.97540640  2.022707360 -1.06790809 -1.07014097
## 54  1.287958e+00 -0.07329602 -0.143874188 -0.55971338  1.35908249
## 55  6.659074e-03  0.19541958  0.211574619  0.04815683  0.59368838
## 56  2.690739e+01  0.12752347  0.208074066 -0.53552187  0.24887243
## 57  2.940000e-05 -0.41541674 -0.494214472 -0.23662847 -0.28910232
## 58  2.975394e+01  1.68323258  1.663675494 -0.99888433  1.54195691
## 59  2.920670e-04 -2.44654200 -2.488391942  0.14588764 -1.73328437
## 60  6.307318e+00  0.34100938  0.302777031 -0.08609828  0.66903108
## 61  1.193899e+02 -0.18924895 -0.288557906  1.43968231 -0.06564846
## 62  7.696165e-02 -0.76369536 -0.883269646 -0.83050063  1.30934590
## 63  1.672548e+01  0.01165956  0.096513307 -0.94760951 -2.03652077
## 64  3.648559e+04  0.82425511  0.774069393  0.74772505 -0.04836412
## 65  5.497997e+00 -0.14192249 -0.098859346  0.98504724 -1.92635500
## 66  5.130241e+00 -0.23270481 -0.242229449 -0.27862360 -0.13957151
## 67  6.441906e-01  0.30558641  0.290689350 -0.65171173  0.87539299
## 68  3.404922e+02  0.46611254  0.528902504  0.96794395 -0.40101247
## 69  2.566154e-01  0.62581862  0.558346997 -0.67825142  0.58769228
## 70  1.759111e+01  0.85003368  0.812606327  0.51304928  0.72763581
## 71  1.113728e-02 -0.94471441 -1.075045805 -0.92857612 -0.06104308
## 72  2.119511e-01 -0.08572455 -0.033742947 -0.47742476 -0.02729820
## 73  1.161045e-02 -1.30038218 -1.257658260 -0.56671344 -1.18622081
## 74  1.937389e-02 -1.07592262 -1.101434275 -1.86585366 -1.70593077
## 75  2.096970e+03 -0.07949225 -0.039044588  1.38451027 -1.17504176
## 76  1.140421e+03  0.52876397  0.588710603  0.97256422  0.88795657
## 77  6.825316e+02  1.84080141  1.823142945 -0.85451309 -1.18813479
## 78  8.503310e+01  2.73513491  2.701251153  1.55863973  1.07017669
## 79  1.604147e+05  1.14786183  1.076712312  0.16096038  0.56421133
## 80  8.874957e+00  2.01690019  2.011348457  0.85080296 -0.79997528
## 81  3.340411e+00 -0.91269148 -0.959555564  1.00999031 -0.46185769
## 82  2.224338e+00  0.86804623  0.857885282 -0.35598100 -0.70646613
## 83  5.213965e+00  0.54884570  0.573991163  1.13458700 -1.10844945
## 84  9.729204e+02  0.90238651  0.927947423  0.64271667 -0.98890188
## 85  2.568767e+02  0.28912132  0.245031745 -0.09302933 -0.57104074
## 86  1.798269e+02  0.85320283  0.893602153  1.30646113  0.26420252
## 87  8.867621e+00  0.35337766  0.300787443  0.80271550  1.18431669
## 88  3.570715e-01  0.75767741  0.749203259 -0.12250077 -1.52831193
## 89  2.444073e-02 -0.33085611 -0.360392140  0.46498725 -1.71305365
## 90  1.919601e-02 -0.53951992 -0.533460585 -0.16224967  0.74267012
## 91  1.099085e-01 -1.16200586 -1.215341928  0.01830939 -0.58786565
## 92  2.522732e+00  0.66898554  0.681065108 -0.63230742  0.13901909
## 93  1.487809e+00 -0.55842836 -0.544165424  1.72685772  0.71848450
## 94  1.344393e-01 -1.29742868 -1.223401507 -0.99316974  0.79462799
## 95  2.626604e+02  0.95731501  0.942356124  1.73279356  1.34003018
## 96  5.862148e+00  1.48626190  1.551085923 -0.45254324 -0.46271087
## 97  7.357203e+00  0.79175894  0.792401715 -0.46968650 -0.71305170
## 98  1.226574e+03  1.39698691  1.399955091  1.17417859  0.55071969
## 99  1.305502e+01  0.30679588  0.242993584 -1.68151109 -0.85418856
## 100 4.444400e+01  1.31951096  1.343514903  0.70519380  0.51040363
##              x5          x6          x7          x8           x9
## 1    0.106603041  1.90546080 -0.64313318  0.83133738  0.069397226
## 2    0.048113406  0.13304981  0.77112214 -1.00337110 -0.637381125
```

```
## 3   -0.689535891 -0.96139722 -0.42695282 -2.03898688 -1.327918885
## 4    0.726207244  0.13140390  0.34750769  1.62399961 -0.702380618
## 5    1.783569221 -0.38016806 -1.69596388  1.17447004 -0.444843952
## 6    0.111867633  0.72595169  1.42055461  0.27589284 -0.815191465
## 7   -0.234039291  0.24481605  0.97600319  1.83334807  0.185934914
## 8   -0.522305869  0.28911221 -0.90288983  0.16025309 -2.035447258
## 9    1.674133143  0.23688817 -0.47114442  0.20716043 -0.798898709
## 10  -0.469902806 -0.58441930 -0.68181262 -0.06050827 -0.629779662
## 11   0.271191966 -1.28401940 -0.55178324  0.14962121 -0.624638810
## 12  -1.072868526  1.44455897  1.95752485  0.74129585 -1.516699256
## 13  -0.417091415  0.28162045  0.71172742  0.45612887  0.713363388
## 14   0.846814894 -0.67379802  0.40452412 -0.22245711  0.940187027
## 15   0.606506960 -0.04757949 -0.34626245  0.68569564 -1.695373745
## 16   1.028743441  0.04754917 -0.16413534  0.21921100  0.364608977
## 17   1.107018328 -0.56406240  0.12190554 -0.36376932 -0.241138967
## 18  -0.760967134 -0.80492656 -0.86978750  0.39825746  1.158414868
## 19   1.829845819 -0.73342870 -0.94666949  0.19451624 -0.905648810
## 20   0.831477112  0.13696804 -0.05774661 -0.76973991 -0.788922227
## 21  -1.103237249 -0.43830930  2.71029359  0.20913206 -1.170123589
## 22  -0.595501385 -0.33002193 -1.93515195 -1.67960528  0.841716695
## 23   0.660742855 -0.62337290  0.79310289  0.22332240 -0.069931890
## 24   1.277794077 -0.19302307  0.36741309  0.54070993  1.578376651
## 25   0.006477450 -0.19545180  0.32556015 -1.09316541  1.106197464
## 26   0.344472105 -0.75029863  2.29815638 -0.67631217 -1.007589516
## 27   0.230299672 -0.07183219 -0.69532621 -1.35099094  0.665157761
## 28   0.051483369 -1.01026450 -1.20622548  0.69385552 -0.226950504
## 29  -1.639354604 -1.71285686 -0.68068843 -0.38012039  1.646191506
## 30   0.227475120  1.27095593  0.54184668 -0.78465842  0.006866338
## 31  -0.962824638 -0.82055062 -0.50390831 -3.71666465  0.147991891
## 32   0.334373806  0.50178516  0.09219205  1.98368334  1.917987357
## 33  -0.080795698 -0.81023288 -0.37316075  0.03634908 -0.233534655
## 34  -0.074325193  1.06945246  0.01082879  1.15854431 -0.835067690
## 35   0.143999844  0.66648608  0.19368505 -0.91332612  0.760951052
## 36  -2.331155078  0.94379764 -1.68707155 -0.65986776 -1.258608935
## 37  -0.030598786  0.81315420 -0.66453940  0.14655152  0.265973462
## 38  -0.515142922  0.40762951 -1.09066780 -0.55211951  1.108107710
## 39  -0.293450681  2.06573211  0.23969523 -0.49416974  1.784205557
## 40  -0.910760616 -0.79746786  0.68803614 -0.65368461 -0.029511134
## 41   0.538701964 -0.72237000  0.31719450 -0.59548843  0.204709211
## 42  -0.379373368 -1.33663099 -0.36466150  0.28464055 -0.719051993
## 43  -1.649353154 -0.90813227 -0.13618015 -2.42328543 -0.111159099
## 44  -0.587146856  1.35898135  0.68224837 -0.70483893  0.399725069
## 45  -0.178258312  0.02239690 -0.38183419  1.32645517  0.820034528
## 46   0.511254635 -0.16785673 -0.65089906 -0.89565732  0.000981539
## 47   1.394295955 -0.63117192  0.39522009  1.76262402  1.002257090
## 48   0.751390370 -1.36665842  1.27679956 -0.03452018 -1.264967063
## 49   1.360312084 -0.54764007  1.21249499 -1.46879553  0.343642876
## 50   0.668041103  0.65942119 -1.80600241  0.73713093  0.212311038
## 51  -1.497937838  0.92783553  0.33461328  1.64629052  1.270520162
## 52  -0.001121686 -0.36074558  0.28086720  0.64195012  1.304493929
## 53   2.025708921  0.18372060  0.32747127  0.85675322 -2.434304298
## 54  -0.135071269 -1.57990902  0.10195429  1.29633293 -0.747062182
## 55  -1.953302876 -3.32045561 -0.75547571 -1.08434120  1.079579708
## 56   0.733094038  0.66993500 -0.68709033  0.97455377  0.067599046
```

```
## 57  -3.110706679  0.21091427 -0.94814810  0.50479839  1.448104278
## 58  -0.560684284  0.72928295  0.01218412 -0.07001845  0.485269016
## 59  -1.113642189  0.90204810 -0.43588234 -0.29226662 -1.412875319
## 60   0.099917732  1.33957013 -1.00504540 -1.27988134  0.010248035
## 61   0.678790584 -1.37578479  1.58067168 -0.17721153  0.903785763
## 62  -0.328917315 -1.03580965 -0.19824551 -1.56607538 -0.083228044
## 63   0.902234190 -0.51993425  1.55323708  0.02340231 -1.725675371
## 64   2.176541490  1.25439732  0.01221935  0.53989788 -1.176444197
## 65   0.359155361 -0.38137468 -0.21197593  1.06951079 -1.437608440
## 66   0.175534351 -0.65950463  0.30733107 -0.41385941  0.241848292
## 67  -1.149128635  0.68920141  0.30618828  0.64407819  0.294383024
## 68   0.838351139 -0.18738236  0.24405404 -0.29996224  0.875283830
## 69  -1.125241257  1.59653610  1.63733360 -1.94479562  0.352309840
## 70  -0.368528506 -0.85826810  0.07802833  1.15353352  0.804810292
## 71  -0.828135290 -0.08516574 -0.69509037  0.09517886  0.985063420
## 72  -0.749955503 -0.22971663 -0.36249241  0.25308145 -0.001017382
## 73  -0.851768270 -1.23800605 -0.92535162  0.52885085  0.803418511
## 74  -0.731668258  1.35741196  0.38386335  0.80694433 -0.126094011
## 75   1.691575384  0.83540373  1.09031679  0.22570345  0.009331247
## 76   0.624028040  0.23273456  1.00896086 -1.03580403  0.613231664
## 77   0.614029002 -0.03046943  0.98165223  0.51575063  1.194503979
## 78  -1.089315254 -1.26292043 -0.38436811 -0.62779261 -0.363265560
## 79   1.816590555 -0.98607306 -0.18009314  1.68261597 -0.204480248
## 80  -1.088306167  0.94007138 -0.20274248  0.56399795 -0.724853921
## 81   0.279441416  2.95171219 -0.12877507 -0.30540863 -2.141559914
## 82  -0.428174906 -0.28351003 -0.76705812  0.17748279  1.411135532
## 83  -0.058817748  1.25912009  0.63157173  0.72074879 -1.970260344
## 84   1.144135369  0.10655278 -0.02424298  0.14306334  0.777691117
## 85   0.873171495 -0.07761400 -0.29553356  1.26574014  1.681854681
## 86   0.181052974  1.27242068 -0.51875229 -2.01387625 -0.079564773
## 87  -0.625515250 -1.64171035 -0.40230057 -0.66624567 -0.787110302
## 88  -0.986388177  0.69065410 -2.06836333  0.92654073 -0.690501123
## 89  -1.770361614  1.71259812 -0.18706843 -1.22297350 -1.986949883
## 90  -0.846689080  0.04733734  0.13436506 -0.81681853 -1.841003952
## 91  -0.481208263  0.92316435  0.25664424  1.72950423  0.830516979
## 92  -0.485354570  0.95476905 -0.16132746 -2.38182018 -0.093828781
## 93  -0.737795961 -0.82398117  1.81235077  0.15040187  0.920305706
## 94  -0.200039016  0.60943564  1.10171351 -0.51613173  0.642118668
## 95   0.105960702 -0.25820161  0.12448903 -1.02063243  0.860383689
## 96  -0.555747690 -0.69003614  0.38342397  0.50078504  2.059735896
## 97   0.020224070 -0.40333495 -0.74090945 -0.71827456 -0.115692554
## 98   0.074184141 -1.19916492  0.80995952 -1.23494372 -0.885696554
## 99   0.793337809 -0.35857612 -0.62057603 -0.53959174  1.010234700
## 100 -0.749197139  1.73137720  0.18763258  1.88859218 -0.606162042
##             x10
## 1    1.13955430
## 2    0.36182605
## 3    0.50456596
## 4   -0.53134563
## 5    0.47327357
## 6    2.17941996
## 7    1.06412371
## 8    1.41167025
## 9   -0.46691668
```

```
## 10   0.55953961
## 11   1.82517774
## 12   0.18046497
## 13   1.64755010
## 14   0.02070462
## 15   0.54561368
## 16   1.21556816
## 17   0.72169162
## 18  -0.30776948
## 19  -0.18771025
## 20   1.45771451
## 21   1.28044712
## 22   0.23462303
## 23   1.41575945
## 24   0.56858929
## 25  -0.81765305
## 26   0.40475753
## 27   1.85524370
## 28   0.65924981
## 29   1.02656510
## 30  -0.89032164
## 31   0.62618347
## 32  -1.11311773
## 33   0.77693533
## 34   0.98145266
## 35   0.65527363
## 36   0.02572670
## 37   0.11347937
## 38  -1.15593094
## 39  -0.53636143
## 40  -2.16853334
## 41  -0.44425598
## 42  -0.68087290
## 43  -0.65405770
## 44   0.85179036
## 45   0.42393059
## 46   1.69118235
## 47  -0.80354661
## 48  -0.30648334
## 49  -2.42375948
## 50   0.73293923
## 51  -0.63928612
## 52   0.41350240
## 53   0.24463690
## 54   0.08462778
## 55  -0.69888375
## 56   0.99111492
## 57   0.55902480
## 58   1.05353168
## 59  -0.90060787
## 60  -0.24721684
## 61   0.23984043
## 62   1.06366407
## 63   0.41491551
```

```
## 64  -0.89119739
## 65  -0.52471453
## 66   1.09703071
## 67   0.12833406
## 68   2.47980070
## 69   1.91084613
## 70   1.21482615
## 71   1.18321540
## 72   0.72596536
## 73   0.16631868
## 74   0.45776133
## 75   0.50387033
## 76  -1.24822048
## 77   0.96434421
## 78  -0.57090721
## 79   0.24812163
## 80   1.49124498
## 81   0.51753355
## 82  -0.80164123
## 83  -0.46179780
## 84   0.39187139
## 85   1.16138340
## 86   0.20073253
## 87  -0.79295204
## 88   2.43258380
## 89   0.01108954
## 90   0.65262878
## 91  -0.52023995
## 92   0.82471275
## 93   0.20835946
## 94  -0.58733443
## 95  -1.58502046
## 96   1.13572107
## 97   0.12638203
## 98  -0.33588327
## 99  -0.93919173
## 100  0.19932810
```

6.2 R Code

```
dat <- read.csv("~/Desktop/5120 Final/dat.csv")
attach(dat)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     y

## The following objects are masked from dat (pos = 3):
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y

## The following objects are masked from dat (pos = 5):
```

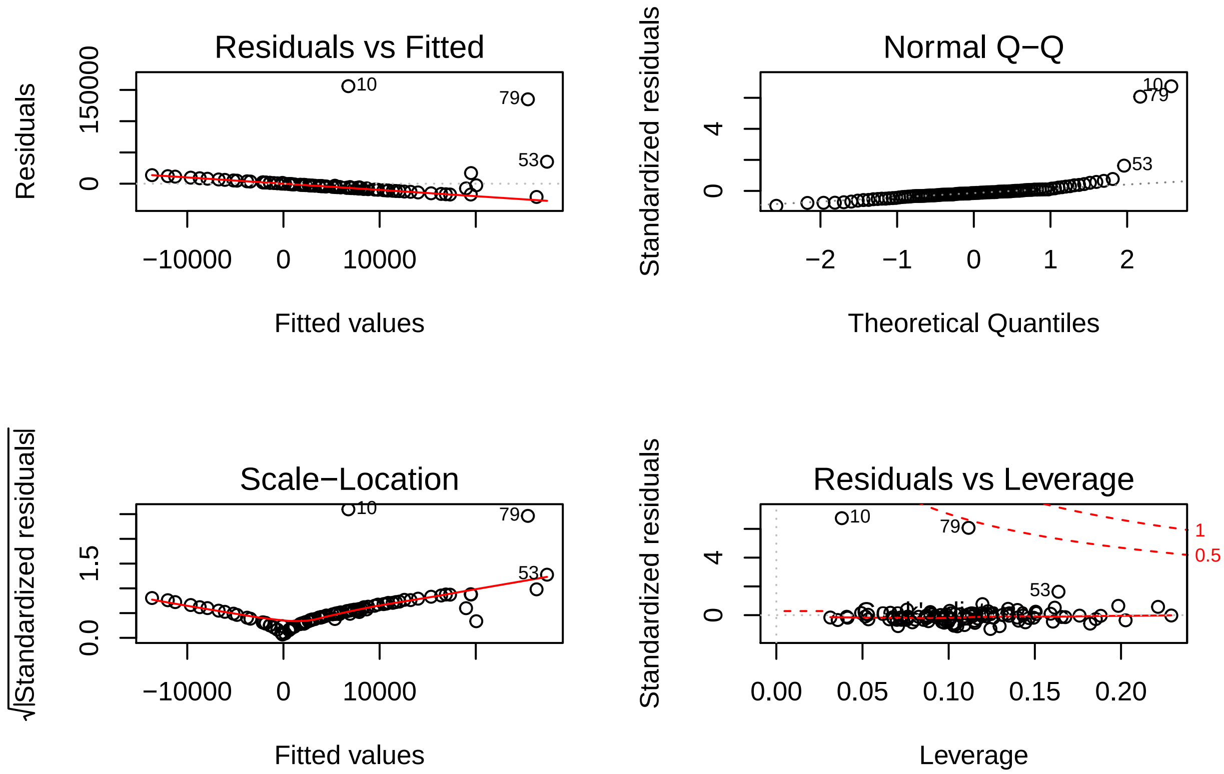
```
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y

## The following objects are masked from dat (pos = 7):
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y
```

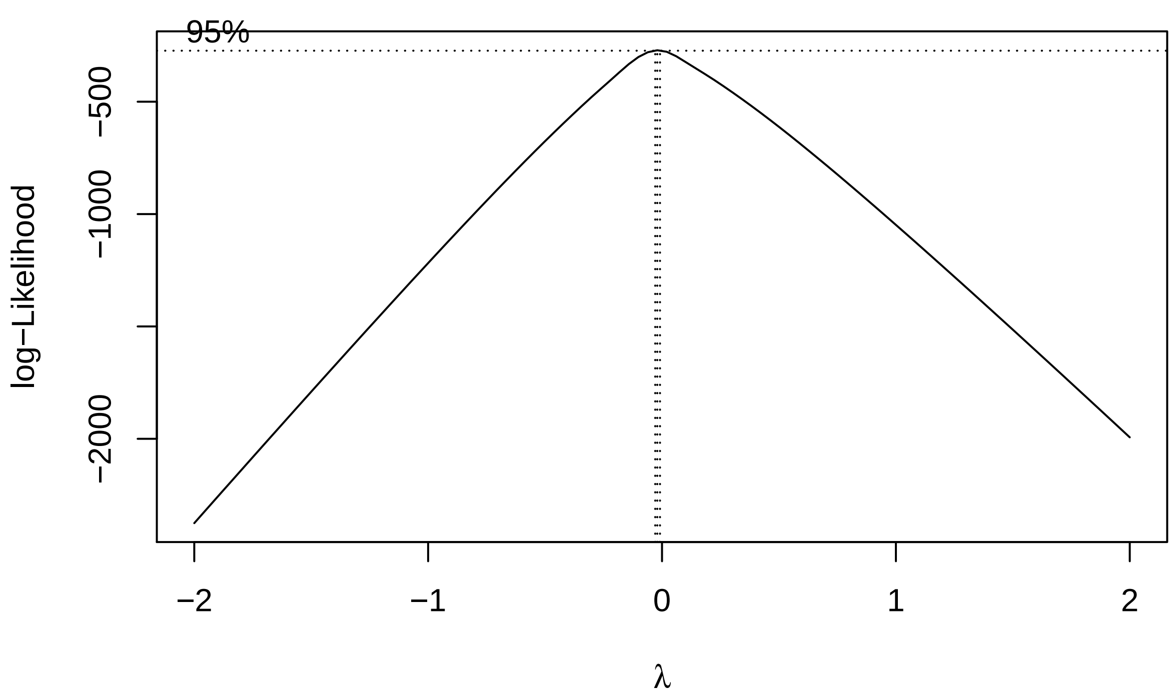
```r
dat.lm=lm(dat$y~.,data=dat)
summary(dat.lm)
```

```
##
## Call:
## lm(formula = dat$y ~ ., data = dat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -21213  -7359  -3119   1056 156003
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4856.9     2554.0   1.902   0.0604 .
## x1           18253.4    50397.7   0.362   0.7181
## x2          -16220.4    50336.8  -0.322   0.7480
## x3            -723.3     2565.5  -0.282   0.7786
## x4             520.9     2317.8   0.225   0.8227
## x5            4563.1     2562.1   1.781   0.0783 .
## x6           -2807.6     2457.2  -1.143   0.2563
## x7           -2535.7     2810.4  -0.902   0.3694
## x8            2839.7     2348.2   1.209   0.2297
## x9           -3654.9     2392.7  -1.527   0.1302
## x10           -577.3     2607.0  -0.221   0.8253
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23590 on 89 degrees of freedom
## Multiple R-squared:  0.1113, Adjusted R-squared:  0.01149
## F-statistic: 1.115 on 10 and 89 DF,  p-value: 0.36
```

```r
par(mfrow=c(2,2))
plot(dat.lm)
```




```
par(mfrow=c(1,1))
library(MASS)
boxcox(dat.lm, plotit = TRUE)
```

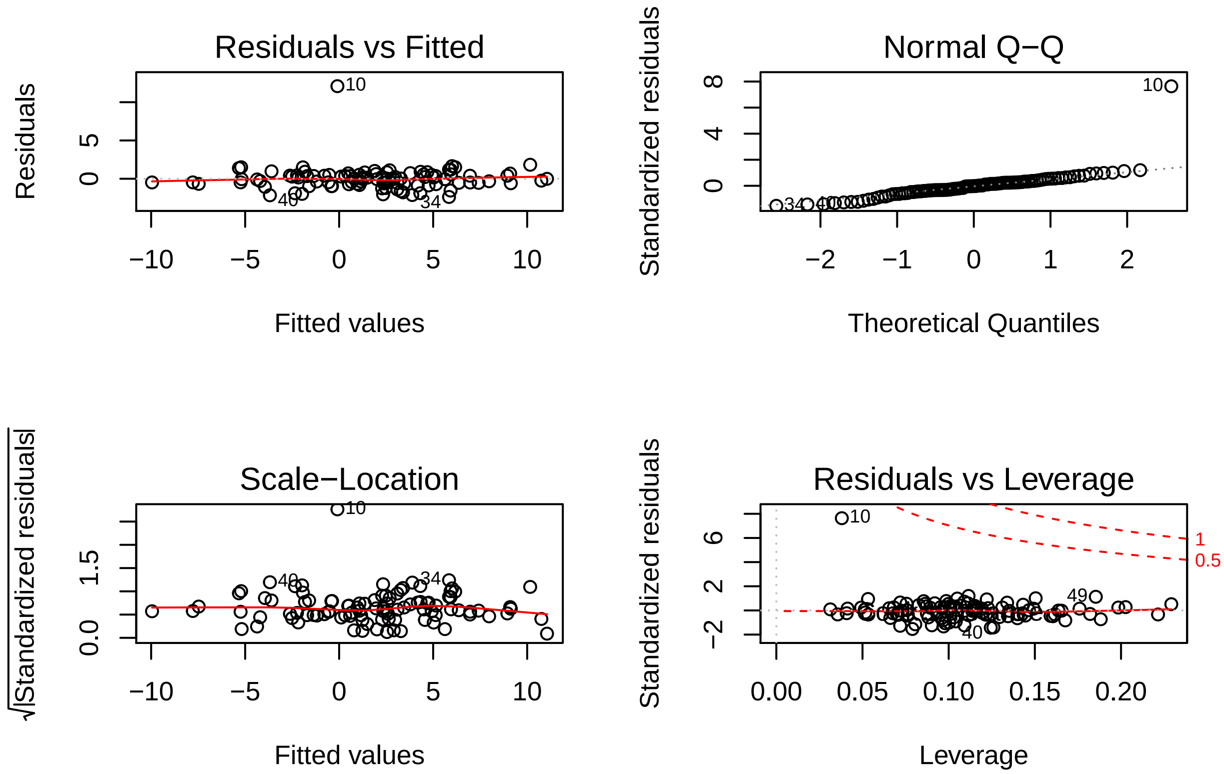



```
dat$y=log(dat$y)
dat.lm=lm(dat$y~.,data=dat)
```

```r
summary(dat.lm)
```

```
##
## Call:
## lm(formula = dat$y ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3787 -0.6130 -0.0369  0.4978 12.0928
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.610750   0.174805   9.215 1.35e-14 ***
## x1          -1.323510   3.449464  -0.384   0.7021
## x2           3.262825   3.445294   0.947   0.3462
## x3           0.945029   0.175598   5.382 5.92e-07 ***
## x4           0.349093   0.158641   2.201   0.0304 *
## x5           3.261169   0.175364  18.597  < 2e-16 ***
## x6          -0.104953   0.168182  -0.624   0.5342
## x7           0.001392   0.192358   0.007   0.9942
## x8           0.086216   0.160720   0.536   0.5930
## x9          -0.066157   0.163769  -0.404   0.6872
## x10          0.080254   0.178437   0.450   0.6540
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.614 on 89 degrees of freedom
## Multiple R-squared:  0.8795, Adjusted R-squared:  0.866
## F-statistic: 64.97 on 10 and 89 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(dat.lm)
```




```r
attach(dat)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     y

## The following objects are masked from dat (pos = 3):
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y

## The following objects are masked from dat (pos = 4):
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y

## The following objects are masked from dat (pos = 6):
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y

## The following objects are masked from dat (pos = 8):
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y
```

```r
dat.lm$residuals
```

```
##           1           2           3           4           5           6
##  0.36165149 -2.03679604 -1.25990826 -0.87294012 -0.59094408  0.22694916
```

```
##           7           8           9          10          11          12
##  0.91445506 -0.42747145  0.67615956 12.09277675  0.39684465  0.70673482
##          13          14          15          16          17          18
##  0.61301290 -1.54270869 -0.57145897 -0.49268510  0.90761831 -1.89567053
##          19          20          21          22          23          24
##  1.50472567  0.03107907  0.21320996  0.73273280 -0.66043868  0.39177543
##          25          26          27          28          29          30
## -0.97302267 -2.12985599  0.39241405 -0.96315529  0.36131924  0.52319298
##          31          32          33          34          35          36
## -0.47421161 -1.21115573  0.28165607 -2.37873048  0.36915979 -0.05378739
##          37          38          39          40          41          42
## -0.05074383  0.49917022 -0.72100531 -2.16000066  0.24585338  0.43959228
##          43          44          45          46          47          48
##  0.97031349  0.04012755  1.02409875 -1.77783334  0.56734973  0.49731647
##          49          50          51          52          53          54
##  1.64630937 -1.89003794 -0.47072953 -0.26611134 -0.01188162 -0.82421181
##          55          56          57          58          59          60
## -1.06336897 -0.88784144 -0.48258014  0.84599502 -0.66984518 -0.51079971
##          61          62          63          64          65          66
##  0.22063896 -0.96786752 -0.63538811 -0.24984842 -1.38626689  0.14034692
##          67          68          69          70          71          72
##  1.48881825 -0.53317116  0.34388223 -0.04063531 -0.29729509 -0.36282110
##          73          74          75          76          77          78
## -0.09116264  1.38740278 -0.32815101  1.19629503  0.57524205 -0.71165091
##          79          80          81          82          83          84
##  1.82581743 -0.41373586 -0.03320768 -0.36846335 -1.62344667 -0.53836283
##          85          86          87          88          89          90
##  0.86190740  0.16400262  0.82308678 -0.46164896  1.50089022 -1.98097576
##          91          92          93          94          95          96
##  0.40999962  0.34303544  0.28850747  0.16727313 -0.05365871 -0.60549986
##          97          98          99         100
## -0.52297153  1.18638495  0.02364513  1.10739083
```

```
tmp<-lm.influence(dat.lm)
tmp$hat[tmp$hat>2*10/100]
```

```
##        11        22        31
## 0.2026707 0.2291824 0.2214943
```

```
DFFITS<-dffits(dat.lm)
DFFITS[DFFITS>1]
```

```
##      10
## 2.57098
```

```
dat[DFFITS>1,]
```

```
##     y         x1         x2        x3         x4         x5         x6
## 10 12 -0.1237738 -0.1015919 0.1449576 -0.8082742 -0.4699028 -0.5844193
##            x7          x8         x9       x10
## 10 -0.6818126 -0.06050827 -0.6297797 0.5595396
```

```r
COOKS<-cooks.distance(dat.lm)
round(COOKS,3)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
## 0.001 0.017 0.007 0.003 0.002 0.000 0.003 0.002 0.002 0.209 0.002 0.003
##    13    14    15    16    17    18    19    20    21    22    23    24
## 0.002 0.010 0.001 0.001 0.004 0.014 0.010 0.000 0.000 0.007 0.001 0.000
##    25    26    27    28    29    30    31    32    33    34    35    36
## 0.003 0.026 0.001 0.004 0.001 0.001 0.003 0.012 0.000 0.018 0.000 0.000
##    37    38    39    40    41    42    43    44    45    46    47    48
## 0.000 0.001 0.003 0.026 0.000 0.001 0.006 0.000 0.003 0.011 0.002 0.001
##    49    50    51    52    53    54    55    56    57    58    59    60
## 0.026 0.017 0.001 0.000 0.000 0.004 0.011 0.003 0.002 0.003 0.004 0.001
##    61    62    63    64    65    66    67    68    69    70    71    72
## 0.000 0.006 0.003 0.000 0.009 0.000 0.005 0.002 0.001 0.000 0.000 0.000
##    73    74    75    76    77    78    79    80    81    82    83    84
## 0.000 0.011 0.000 0.006 0.001 0.004 0.016 0.001 0.000 0.000 0.011 0.001
##    85    86    87    88    89    90    91    92    93    94    95    96
## 0.002 0.000 0.003 0.001 0.016 0.011 0.001 0.000 0.000 0.000 0.000 0.002
##    97    98    99   100
## 0.000 0.005 0.000 0.006
```

```r
COOKS>qf(0.5,10,90)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    13    14    15    16    17    18    19    20    21    22    23    24
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    25    26    27    28    29    30    31    32    33    34    35    36
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    37    38    39    40    41    42    43    44    45    46    47    48
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    49    50    51    52    53    54    55    56    57    58    59    60
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    61    62    63    64    65    66    67    68    69    70    71    72
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    73    74    75    76    77    78    79    80    81    82    83    84
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    85    86    87    88    89    90    91    92    93    94    95    96
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    97    98    99   100
## FALSE FALSE FALSE FALSE
```

```r
no10<- dat[-10, ]
no10.lm=lm(no10$y~.,data=no10)
summary(no10.lm)
```

```
##
## Call:
## lm(formula = no10$y ~ ., data = no10)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -2.3706 -0.4279  0.0241  0.5760  1.8382
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.482703   0.103688  14.300  < 2e-16 ***
## x1           0.692642   2.042699   0.339    0.735
## x2           1.295552   2.039959   0.635    0.527
## x3           0.909189   0.103719   8.766 1.25e-13 ***
## x4           0.407380   0.093778   4.344 3.73e-05 ***
## x5           3.322817   0.103653  32.057  < 2e-16 ***
## x6          -0.007948   0.099586  -0.080    0.937
## x7           0.130916   0.114019   1.148    0.254
## x8           0.070550   0.094905   0.743    0.459
## x9           0.027593   0.096969   0.285    0.777
## x10          0.052156   0.105381   0.495    0.622
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9533 on 88 degrees of freedom
## Multiple R-squared:  0.9561, Adjusted R-squared:  0.9511
## F-statistic: 191.6 on 10 and 88 DF,  p-value: < 2.2e-16
```

```r
round(cor(no10),3)
```

```
##          y     x1     x2     x3     x4     x5     x6     x7     x8     x9
## y    1.000  0.554  0.555  0.324  0.042  0.808 -0.015  0.141  0.205 -0.096
## x1   0.554  1.000  0.999  0.060 -0.020  0.063  0.001 -0.003  0.077 -0.085
## x2   0.555  0.999  1.000  0.060 -0.020  0.064  0.003  0.008  0.079 -0.088
## x3   0.324  0.060  0.060  1.000 -0.003  0.126  0.028  0.063  0.017 -0.007
## x4   0.042 -0.020 -0.020 -0.003  1.000 -0.061  0.000  0.071 -0.144  0.085
## x5   0.808  0.063  0.064  0.126 -0.061  1.000 -0.030  0.125  0.211 -0.087
## x6  -0.015  0.001  0.003  0.028  0.000 -0.030  1.000  0.034  0.109 -0.166
## x7   0.141 -0.003  0.008  0.063  0.071  0.125  0.034  1.000  0.019 -0.068
## x8   0.205  0.077  0.079  0.017 -0.144  0.211  0.109  0.019  1.000  0.044
## x9  -0.096 -0.085 -0.088 -0.007  0.085 -0.087 -0.166 -0.068  0.044  1.000
## x10  0.033  0.158  0.160 -0.069 -0.112 -0.042  0.100 -0.051  0.078 -0.058
##        x10
## y    0.033
## x1   0.158
## x2   0.160
## x3  -0.069
## x4  -0.112
## x5  -0.042
## x6   0.100
## x7  -0.051
## x8   0.078
## x9  -0.058
## x10  1.000
```

```r
library(faraway)
vif(no10.lm)
```

```
##         x1         x2         x3         x4         x5         x6
```

```
## 515.180302 515.882996   1.028513   1.049128   1.099961   1.060420
##          x7          x8          x9         x10
##    1.095493    1.101787    1.069866    1.068766
```

```r
data=no10
start<-lm(y~1, data=data)
end<-lm(y~.,data=data)
result<-step(start, scope=list(lower=start, upper=end), direction="forward", trace=FALSE)
summary(result)
```

```
##
## Call:
## lm(formula = y ~ x5 + x2 + x3 + x4, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.29129 -0.46534  0.00512  0.62212  1.87873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.50163    0.09635  15.584  < 2e-16 ***
## x5           3.34805    0.09792  34.191  < 2e-16 ***
## x2           1.99583    0.08829  22.605  < 2e-16 ***
## x3           0.91006    0.10112   8.999 2.46e-14 ***
## x4           0.40358    0.08986   4.491 2.01e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9338 on 94 degrees of freedom
## Multiple R-squared:  0.955,  Adjusted R-squared:  0.9531
## F-statistic: 498.6 on 4 and 94 DF,  p-value: < 2.2e-16
```

```r
dat=data
attach(dat)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     y

## The following objects are masked from dat (pos = 3):
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y

## The following objects are masked from dat (pos = 4):
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y

## The following objects are masked from dat (pos = 5):
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y

## The following objects are masked from dat (pos = 7):
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y
```

```
## The following objects are masked from dat (pos = 9):
##
##     x1, x10, x2, x3, x4, x5, x6, x7, x8, x9, y
```

```
##################################################
##STEP 1: create design matrix for maximal model##
##################################################

x<-cbind(Const=1, x1, x2,x3,x4,x5,x6,x7,x8,x9,x10)
y<-y

###########################################################################################
##STEP 2: logical matrix to indicate for each possible model, which predictors are in/out##
###########################################################################################

models<-matrix(F, 2^10, 10) ##10 predictors, change accordingly
dimnames(models)<-list(NULL, c("x1","x2","x3","x4","x5","x6","x7","x8","x9","x10"))

row<-0

for (a in c(F,T)) { ##loop has 10 for statements, one for each predictor
  for (b in c(F,T)) {
    for (c in c(F,T)) {
      for (d in c(F,T)) {
        for (e in c(F,T)) {
          for (f in c(F,T)) {
            for (g in c(F,T)) {
              for (h in c(F,T)) {
                for (i in c(F,T)) {
                  for (j in c(F,T)) {
row<-row+1
models[row,]<-c(a,b,c,d,e,f,g,h,i,j)
}}}}}}}}}}

#################################################
##STEP 3: Matrix to store results from criteria##
#################################################

results<-matrix(NA, 2^10, 7) ##10 predictors, store 7 statistics
dimnames(results)<-list(NULL, c("p","R2","R2.adj","PRESS","AIC","BIC","Cp"))

#################################
##STEP 4: MSE for maximal model##
#################################

tmp<-lsfit(x,y,intercept=F)
n<-nrow(dat)
p<-ncol(x)
MSE.max<-sum(tmp$res^2)/(n-p)

###################################
##STEP 5: Fit all possible models##
###################################
```

```r
time1<-Sys.time()

for (i in 1:(2^10)){ ##10 predictors
  which<-c(T, models[i,]) ##pull out the row of model and append an intercept
  tmp<-lsfit(x[,which], y, intercept=F) ##fit the model and compute the criteria
  p<-sum(which) ##number of parameters for chosen model
  SSTo<-(n-1)*var(y)
  MSTo<-var(y)
  SSE<-sum(tmp$res^2)
  MSE<-SSE/(n-p)
  R2<-1-(SSE/SSTo)
  R2.adj<-1-(MSE/MSTo)
  hi<-ls.diag(tmp)$hat ##leverages
  res.PRESS<-tmp$res/(1-hi)
  PRESS<-sum(res.PRESS^2)
  AIC<-n*log(SSE/n)+2*p
  BIC<-n*log(SSE/n)+p*log(n)
  Cp<-(SSE/MSE.max)-n+2*p
  ##store the results
  results[i,1]<-p
  results[i,2]<-R2
  results[i,3]<-R2.adj
  results[i,4]<-PRESS
  results[i,5]<-AIC
  results[i,6]<-BIC
  results[i,7]<-Cp
  ##have R print out the iteration we are at
  print(paste("run",i))}
```

```
## [1] "run 1"
## [1] "run 2"
## [1] "run 3"
## [1] "run 4"
## [1] "run 5"
## [1] "run 6"
## [1] "run 7"
## [1] "run 8"
## [1] "run 9"
## [1] "run 10"
## [1] "run 11"
## [1] "run 12"
## [1] "run 13"
## [1] "run 14"
## [1] "run 15"
## [1] "run 16"
## [1] "run 17"
## [1] "run 18"
## [1] "run 19"
## [1] "run 20"
## [1] "run 21"
## [1] "run 22"
## [1] "run 23"
## [1] "run 24"
```

```
## [1] "run 25"
## [1] "run 26"
## [1] "run 27"
## [1] "run 28"
## [1] "run 29"
## [1] "run 30"
## [1] "run 31"
## [1] "run 32"
## [1] "run 33"
## [1] "run 34"
## [1] "run 35"
## [1] "run 36"
## [1] "run 37"
## [1] "run 38"
## [1] "run 39"
## [1] "run 40"
## [1] "run 41"
## [1] "run 42"
## [1] "run 43"
## [1] "run 44"
## [1] "run 45"
## [1] "run 46"
## [1] "run 47"
## [1] "run 48"
## [1] "run 49"
## [1] "run 50"
## [1] "run 51"
## [1] "run 52"
## [1] "run 53"
## [1] "run 54"
## [1] "run 55"
## [1] "run 56"
## [1] "run 57"
## [1] "run 58"
## [1] "run 59"
## [1] "run 60"
## [1] "run 61"
## [1] "run 62"
## [1] "run 63"
## [1] "run 64"
## [1] "run 65"
## [1] "run 66"
## [1] "run 67"
## [1] "run 68"
## [1] "run 69"
## [1] "run 70"
## [1] "run 71"
## [1] "run 72"
## [1] "run 73"
## [1] "run 74"
## [1] "run 75"
## [1] "run 76"
## [1] "run 77"
## [1] "run 78"
```

```
## [1] "run 79"
## [1] "run 80"
## [1] "run 81"
## [1] "run 82"
## [1] "run 83"
## [1] "run 84"
## [1] "run 85"
## [1] "run 86"
## [1] "run 87"
## [1] "run 88"
## [1] "run 89"
## [1] "run 90"
## [1] "run 91"
## [1] "run 92"
## [1] "run 93"
## [1] "run 94"
## [1] "run 95"
## [1] "run 96"
## [1] "run 97"
## [1] "run 98"
## [1] "run 99"
## [1] "run 100"
## [1] "run 101"
## [1] "run 102"
## [1] "run 103"
## [1] "run 104"
## [1] "run 105"
## [1] "run 106"
## [1] "run 107"
## [1] "run 108"
## [1] "run 109"
## [1] "run 110"
## [1] "run 111"
## [1] "run 112"
## [1] "run 113"
## [1] "run 114"
## [1] "run 115"
## [1] "run 116"
## [1] "run 117"
## [1] "run 118"
## [1] "run 119"
## [1] "run 120"
## [1] "run 121"
## [1] "run 122"
## [1] "run 123"
## [1] "run 124"
## [1] "run 125"
## [1] "run 126"
## [1] "run 127"
## [1] "run 128"
## [1] "run 129"
## [1] "run 130"
## [1] "run 131"
## [1] "run 132"
```

```
## [1] "run 133"
## [1] "run 134"
## [1] "run 135"
## [1] "run 136"
## [1] "run 137"
## [1] "run 138"
## [1] "run 139"
## [1] "run 140"
## [1] "run 141"
## [1] "run 142"
## [1] "run 143"
## [1] "run 144"
## [1] "run 145"
## [1] "run 146"
## [1] "run 147"
## [1] "run 148"
## [1] "run 149"
## [1] "run 150"
## [1] "run 151"
## [1] "run 152"
## [1] "run 153"
## [1] "run 154"
## [1] "run 155"
## [1] "run 156"
## [1] "run 157"
## [1] "run 158"
## [1] "run 159"
## [1] "run 160"
## [1] "run 161"
## [1] "run 162"
## [1] "run 163"
## [1] "run 164"
## [1] "run 165"
## [1] "run 166"
## [1] "run 167"
## [1] "run 168"
## [1] "run 169"
## [1] "run 170"
## [1] "run 171"
## [1] "run 172"
## [1] "run 173"
## [1] "run 174"
## [1] "run 175"
## [1] "run 176"
## [1] "run 177"
## [1] "run 178"
## [1] "run 179"
## [1] "run 180"
## [1] "run 181"
## [1] "run 182"
## [1] "run 183"
## [1] "run 184"
## [1] "run 185"
## [1] "run 186"
```

```
## [1] "run 187"
## [1] "run 188"
## [1] "run 189"
## [1] "run 190"
## [1] "run 191"
## [1] "run 192"
## [1] "run 193"
## [1] "run 194"
## [1] "run 195"
## [1] "run 196"
## [1] "run 197"
## [1] "run 198"
## [1] "run 199"
## [1] "run 200"
## [1] "run 201"
## [1] "run 202"
## [1] "run 203"
## [1] "run 204"
## [1] "run 205"
## [1] "run 206"
## [1] "run 207"
## [1] "run 208"
## [1] "run 209"
## [1] "run 210"
## [1] "run 211"
## [1] "run 212"
## [1] "run 213"
## [1] "run 214"
## [1] "run 215"
## [1] "run 216"
## [1] "run 217"
## [1] "run 218"
## [1] "run 219"
## [1] "run 220"
## [1] "run 221"
## [1] "run 222"
## [1] "run 223"
## [1] "run 224"
## [1] "run 225"
## [1] "run 226"
## [1] "run 227"
## [1] "run 228"
## [1] "run 229"
## [1] "run 230"
## [1] "run 231"
## [1] "run 232"
## [1] "run 233"
## [1] "run 234"
## [1] "run 235"
## [1] "run 236"
## [1] "run 237"
## [1] "run 238"
## [1] "run 239"
## [1] "run 240"
```

```
## [1] "run 241"
## [1] "run 242"
## [1] "run 243"
## [1] "run 244"
## [1] "run 245"
## [1] "run 246"
## [1] "run 247"
## [1] "run 248"
## [1] "run 249"
## [1] "run 250"
## [1] "run 251"
## [1] "run 252"
## [1] "run 253"
## [1] "run 254"
## [1] "run 255"
## [1] "run 256"
## [1] "run 257"
## [1] "run 258"
## [1] "run 259"
## [1] "run 260"
## [1] "run 261"
## [1] "run 262"
## [1] "run 263"
## [1] "run 264"
## [1] "run 265"
## [1] "run 266"
## [1] "run 267"
## [1] "run 268"
## [1] "run 269"
## [1] "run 270"
## [1] "run 271"
## [1] "run 272"
## [1] "run 273"
## [1] "run 274"
## [1] "run 275"
## [1] "run 276"
## [1] "run 277"
## [1] "run 278"
## [1] "run 279"
## [1] "run 280"
## [1] "run 281"
## [1] "run 282"
## [1] "run 283"
## [1] "run 284"
## [1] "run 285"
## [1] "run 286"
## [1] "run 287"
## [1] "run 288"
## [1] "run 289"
## [1] "run 290"
## [1] "run 291"
## [1] "run 292"
## [1] "run 293"
## [1] "run 294"
```

```
## [1] "run 295"
## [1] "run 296"
## [1] "run 297"
## [1] "run 298"
## [1] "run 299"
## [1] "run 300"
## [1] "run 301"
## [1] "run 302"
## [1] "run 303"
## [1] "run 304"
## [1] "run 305"
## [1] "run 306"
## [1] "run 307"
## [1] "run 308"
## [1] "run 309"
## [1] "run 310"
## [1] "run 311"
## [1] "run 312"
## [1] "run 313"
## [1] "run 314"
## [1] "run 315"
## [1] "run 316"
## [1] "run 317"
## [1] "run 318"
## [1] "run 319"
## [1] "run 320"
## [1] "run 321"
## [1] "run 322"
## [1] "run 323"
## [1] "run 324"
## [1] "run 325"
## [1] "run 326"
## [1] "run 327"
## [1] "run 328"
## [1] "run 329"
## [1] "run 330"
## [1] "run 331"
## [1] "run 332"
## [1] "run 333"
## [1] "run 334"
## [1] "run 335"
## [1] "run 336"
## [1] "run 337"
## [1] "run 338"
## [1] "run 339"
## [1] "run 340"
## [1] "run 341"
## [1] "run 342"
## [1] "run 343"
## [1] "run 344"
## [1] "run 345"
## [1] "run 346"
## [1] "run 347"
## [1] "run 348"
```

```
## [1] "run 349"
## [1] "run 350"
## [1] "run 351"
## [1] "run 352"
## [1] "run 353"
## [1] "run 354"
## [1] "run 355"
## [1] "run 356"
## [1] "run 357"
## [1] "run 358"
## [1] "run 359"
## [1] "run 360"
## [1] "run 361"
## [1] "run 362"
## [1] "run 363"
## [1] "run 364"
## [1] "run 365"
## [1] "run 366"
## [1] "run 367"
## [1] "run 368"
## [1] "run 369"
## [1] "run 370"
## [1] "run 371"
## [1] "run 372"
## [1] "run 373"
## [1] "run 374"
## [1] "run 375"
## [1] "run 376"
## [1] "run 377"
## [1] "run 378"
## [1] "run 379"
## [1] "run 380"
## [1] "run 381"
## [1] "run 382"
## [1] "run 383"
## [1] "run 384"
## [1] "run 385"
## [1] "run 386"
## [1] "run 387"
## [1] "run 388"
## [1] "run 389"
## [1] "run 390"
## [1] "run 391"
## [1] "run 392"
## [1] "run 393"
## [1] "run 394"
## [1] "run 395"
## [1] "run 396"
## [1] "run 397"
## [1] "run 398"
## [1] "run 399"
## [1] "run 400"
## [1] "run 401"
## [1] "run 402"
```

```
## [1] "run 403"
## [1] "run 404"
## [1] "run 405"
## [1] "run 406"
## [1] "run 407"
## [1] "run 408"
## [1] "run 409"
## [1] "run 410"
## [1] "run 411"
## [1] "run 412"
## [1] "run 413"
## [1] "run 414"
## [1] "run 415"
## [1] "run 416"
## [1] "run 417"
## [1] "run 418"
## [1] "run 419"
## [1] "run 420"
## [1] "run 421"
## [1] "run 422"
## [1] "run 423"
## [1] "run 424"
## [1] "run 425"
## [1] "run 426"
## [1] "run 427"
## [1] "run 428"
## [1] "run 429"
## [1] "run 430"
## [1] "run 431"
## [1] "run 432"
## [1] "run 433"
## [1] "run 434"
## [1] "run 435"
## [1] "run 436"
## [1] "run 437"
## [1] "run 438"
## [1] "run 439"
## [1] "run 440"
## [1] "run 441"
## [1] "run 442"
## [1] "run 443"
## [1] "run 444"
## [1] "run 445"
## [1] "run 446"
## [1] "run 447"
## [1] "run 448"
## [1] "run 449"
## [1] "run 450"
## [1] "run 451"
## [1] "run 452"
## [1] "run 453"
## [1] "run 454"
## [1] "run 455"
## [1] "run 456"
```

```
## [1] "run 457"
## [1] "run 458"
## [1] "run 459"
## [1] "run 460"
## [1] "run 461"
## [1] "run 462"
## [1] "run 463"
## [1] "run 464"
## [1] "run 465"
## [1] "run 466"
## [1] "run 467"
## [1] "run 468"
## [1] "run 469"
## [1] "run 470"
## [1] "run 471"
## [1] "run 472"
## [1] "run 473"
## [1] "run 474"
## [1] "run 475"
## [1] "run 476"
## [1] "run 477"
## [1] "run 478"
## [1] "run 479"
## [1] "run 480"
## [1] "run 481"
## [1] "run 482"
## [1] "run 483"
## [1] "run 484"
## [1] "run 485"
## [1] "run 486"
## [1] "run 487"
## [1] "run 488"
## [1] "run 489"
## [1] "run 490"
## [1] "run 491"
## [1] "run 492"
## [1] "run 493"
## [1] "run 494"
## [1] "run 495"
## [1] "run 496"
## [1] "run 497"
## [1] "run 498"
## [1] "run 499"
## [1] "run 500"
## [1] "run 501"
## [1] "run 502"
## [1] "run 503"
## [1] "run 504"
## [1] "run 505"
## [1] "run 506"
## [1] "run 507"
## [1] "run 508"
## [1] "run 509"
## [1] "run 510"
```

```
## [1] "run 511"
## [1] "run 512"
## [1] "run 513"
## [1] "run 514"
## [1] "run 515"
## [1] "run 516"
## [1] "run 517"
## [1] "run 518"
## [1] "run 519"
## [1] "run 520"
## [1] "run 521"
## [1] "run 522"
## [1] "run 523"
## [1] "run 524"
## [1] "run 525"
## [1] "run 526"
## [1] "run 527"
## [1] "run 528"
## [1] "run 529"
## [1] "run 530"
## [1] "run 531"
## [1] "run 532"
## [1] "run 533"
## [1] "run 534"
## [1] "run 535"
## [1] "run 536"
## [1] "run 537"
## [1] "run 538"
## [1] "run 539"
## [1] "run 540"
## [1] "run 541"
## [1] "run 542"
## [1] "run 543"
## [1] "run 544"
## [1] "run 545"
## [1] "run 546"
## [1] "run 547"
## [1] "run 548"
## [1] "run 549"
## [1] "run 550"
## [1] "run 551"
## [1] "run 552"
## [1] "run 553"
## [1] "run 554"
## [1] "run 555"
## [1] "run 556"
## [1] "run 557"
## [1] "run 558"
## [1] "run 559"
## [1] "run 560"
## [1] "run 561"
## [1] "run 562"
## [1] "run 563"
## [1] "run 564"
```

```
## [1] "run 565"
## [1] "run 566"
## [1] "run 567"
## [1] "run 568"
## [1] "run 569"
## [1] "run 570"
## [1] "run 571"
## [1] "run 572"
## [1] "run 573"
## [1] "run 574"
## [1] "run 575"
## [1] "run 576"
## [1] "run 577"
## [1] "run 578"
## [1] "run 579"
## [1] "run 580"
## [1] "run 581"
## [1] "run 582"
## [1] "run 583"
## [1] "run 584"
## [1] "run 585"
## [1] "run 586"
## [1] "run 587"
## [1] "run 588"
## [1] "run 589"
## [1] "run 590"
## [1] "run 591"
## [1] "run 592"
## [1] "run 593"
## [1] "run 594"
## [1] "run 595"
## [1] "run 596"
## [1] "run 597"
## [1] "run 598"
## [1] "run 599"
## [1] "run 600"
## [1] "run 601"
## [1] "run 602"
## [1] "run 603"
## [1] "run 604"
## [1] "run 605"
## [1] "run 606"
## [1] "run 607"
## [1] "run 608"
## [1] "run 609"
## [1] "run 610"
## [1] "run 611"
## [1] "run 612"
## [1] "run 613"
## [1] "run 614"
## [1] "run 615"
## [1] "run 616"
## [1] "run 617"
## [1] "run 618"
```

```
## [1] "run 619"
## [1] "run 620"
## [1] "run 621"
## [1] "run 622"
## [1] "run 623"
## [1] "run 624"
## [1] "run 625"
## [1] "run 626"
## [1] "run 627"
## [1] "run 628"
## [1] "run 629"
## [1] "run 630"
## [1] "run 631"
## [1] "run 632"
## [1] "run 633"
## [1] "run 634"
## [1] "run 635"
## [1] "run 636"
## [1] "run 637"
## [1] "run 638"
## [1] "run 639"
## [1] "run 640"
## [1] "run 641"
## [1] "run 642"
## [1] "run 643"
## [1] "run 644"
## [1] "run 645"
## [1] "run 646"
## [1] "run 647"
## [1] "run 648"
## [1] "run 649"
## [1] "run 650"
## [1] "run 651"
## [1] "run 652"
## [1] "run 653"
## [1] "run 654"
## [1] "run 655"
## [1] "run 656"
## [1] "run 657"
## [1] "run 658"
## [1] "run 659"
## [1] "run 660"
## [1] "run 661"
## [1] "run 662"
## [1] "run 663"
## [1] "run 664"
## [1] "run 665"
## [1] "run 666"
## [1] "run 667"
## [1] "run 668"
## [1] "run 669"
## [1] "run 670"
## [1] "run 671"
## [1] "run 672"
```

```
## [1] "run 673"
## [1] "run 674"
## [1] "run 675"
## [1] "run 676"
## [1] "run 677"
## [1] "run 678"
## [1] "run 679"
## [1] "run 680"
## [1] "run 681"
## [1] "run 682"
## [1] "run 683"
## [1] "run 684"
## [1] "run 685"
## [1] "run 686"
## [1] "run 687"
## [1] "run 688"
## [1] "run 689"
## [1] "run 690"
## [1] "run 691"
## [1] "run 692"
## [1] "run 693"
## [1] "run 694"
## [1] "run 695"
## [1] "run 696"
## [1] "run 697"
## [1] "run 698"
## [1] "run 699"
## [1] "run 700"
## [1] "run 701"
## [1] "run 702"
## [1] "run 703"
## [1] "run 704"
## [1] "run 705"
## [1] "run 706"
## [1] "run 707"
## [1] "run 708"
## [1] "run 709"
## [1] "run 710"
## [1] "run 711"
## [1] "run 712"
## [1] "run 713"
## [1] "run 714"
## [1] "run 715"
## [1] "run 716"
## [1] "run 717"
## [1] "run 718"
## [1] "run 719"
## [1] "run 720"
## [1] "run 721"
## [1] "run 722"
## [1] "run 723"
## [1] "run 724"
## [1] "run 725"
## [1] "run 726"
```

```
## [1] "run 727"
## [1] "run 728"
## [1] "run 729"
## [1] "run 730"
## [1] "run 731"
## [1] "run 732"
## [1] "run 733"
## [1] "run 734"
## [1] "run 735"
## [1] "run 736"
## [1] "run 737"
## [1] "run 738"
## [1] "run 739"
## [1] "run 740"
## [1] "run 741"
## [1] "run 742"
## [1] "run 743"
## [1] "run 744"
## [1] "run 745"
## [1] "run 746"
## [1] "run 747"
## [1] "run 748"
## [1] "run 749"
## [1] "run 750"
## [1] "run 751"
## [1] "run 752"
## [1] "run 753"
## [1] "run 754"
## [1] "run 755"
## [1] "run 756"
## [1] "run 757"
## [1] "run 758"
## [1] "run 759"
## [1] "run 760"
## [1] "run 761"
## [1] "run 762"
## [1] "run 763"
## [1] "run 764"
## [1] "run 765"
## [1] "run 766"
## [1] "run 767"
## [1] "run 768"
## [1] "run 769"
## [1] "run 770"
## [1] "run 771"
## [1] "run 772"
## [1] "run 773"
## [1] "run 774"
## [1] "run 775"
## [1] "run 776"
## [1] "run 777"
## [1] "run 778"
## [1] "run 779"
## [1] "run 780"
```

```
## [1] "run 781"
## [1] "run 782"
## [1] "run 783"
## [1] "run 784"
## [1] "run 785"
## [1] "run 786"
## [1] "run 787"
## [1] "run 788"
## [1] "run 789"
## [1] "run 790"
## [1] "run 791"
## [1] "run 792"
## [1] "run 793"
## [1] "run 794"
## [1] "run 795"
## [1] "run 796"
## [1] "run 797"
## [1] "run 798"
## [1] "run 799"
## [1] "run 800"
## [1] "run 801"
## [1] "run 802"
## [1] "run 803"
## [1] "run 804"
## [1] "run 805"
## [1] "run 806"
## [1] "run 807"
## [1] "run 808"
## [1] "run 809"
## [1] "run 810"
## [1] "run 811"
## [1] "run 812"
## [1] "run 813"
## [1] "run 814"
## [1] "run 815"
## [1] "run 816"
## [1] "run 817"
## [1] "run 818"
## [1] "run 819"
## [1] "run 820"
## [1] "run 821"
## [1] "run 822"
## [1] "run 823"
## [1] "run 824"
## [1] "run 825"
## [1] "run 826"
## [1] "run 827"
## [1] "run 828"
## [1] "run 829"
## [1] "run 830"
## [1] "run 831"
## [1] "run 832"
## [1] "run 833"
## [1] "run 834"
```

```
## [1] "run 835"
## [1] "run 836"
## [1] "run 837"
## [1] "run 838"
## [1] "run 839"
## [1] "run 840"
## [1] "run 841"
## [1] "run 842"
## [1] "run 843"
## [1] "run 844"
## [1] "run 845"
## [1] "run 846"
## [1] "run 847"
## [1] "run 848"
## [1] "run 849"
## [1] "run 850"
## [1] "run 851"
## [1] "run 852"
## [1] "run 853"
## [1] "run 854"
## [1] "run 855"
## [1] "run 856"
## [1] "run 857"
## [1] "run 858"
## [1] "run 859"
## [1] "run 860"
## [1] "run 861"
## [1] "run 862"
## [1] "run 863"
## [1] "run 864"
## [1] "run 865"
## [1] "run 866"
## [1] "run 867"
## [1] "run 868"
## [1] "run 869"
## [1] "run 870"
## [1] "run 871"
## [1] "run 872"
## [1] "run 873"
## [1] "run 874"
## [1] "run 875"
## [1] "run 876"
## [1] "run 877"
## [1] "run 878"
## [1] "run 879"
## [1] "run 880"
## [1] "run 881"
## [1] "run 882"
## [1] "run 883"
## [1] "run 884"
## [1] "run 885"
## [1] "run 886"
## [1] "run 887"
## [1] "run 888"
```

```
## [1] "run 889"
## [1] "run 890"
## [1] "run 891"
## [1] "run 892"
## [1] "run 893"
## [1] "run 894"
## [1] "run 895"
## [1] "run 896"
## [1] "run 897"
## [1] "run 898"
## [1] "run 899"
## [1] "run 900"
## [1] "run 901"
## [1] "run 902"
## [1] "run 903"
## [1] "run 904"
## [1] "run 905"
## [1] "run 906"
## [1] "run 907"
## [1] "run 908"
## [1] "run 909"
## [1] "run 910"
## [1] "run 911"
## [1] "run 912"
## [1] "run 913"
## [1] "run 914"
## [1] "run 915"
## [1] "run 916"
## [1] "run 917"
## [1] "run 918"
## [1] "run 919"
## [1] "run 920"
## [1] "run 921"
## [1] "run 922"
## [1] "run 923"
## [1] "run 924"
## [1] "run 925"
## [1] "run 926"
## [1] "run 927"
## [1] "run 928"
## [1] "run 929"
## [1] "run 930"
## [1] "run 931"
## [1] "run 932"
## [1] "run 933"
## [1] "run 934"
## [1] "run 935"
## [1] "run 936"
## [1] "run 937"
## [1] "run 938"
## [1] "run 939"
## [1] "run 940"
## [1] "run 941"
## [1] "run 942"
```

```
## [1] "run 943"
## [1] "run 944"
## [1] "run 945"
## [1] "run 946"
## [1] "run 947"
## [1] "run 948"
## [1] "run 949"
## [1] "run 950"
## [1] "run 951"
## [1] "run 952"
## [1] "run 953"
## [1] "run 954"
## [1] "run 955"
## [1] "run 956"
## [1] "run 957"
## [1] "run 958"
## [1] "run 959"
## [1] "run 960"
## [1] "run 961"
## [1] "run 962"
## [1] "run 963"
## [1] "run 964"
## [1] "run 965"
## [1] "run 966"
## [1] "run 967"
## [1] "run 968"
## [1] "run 969"
## [1] "run 970"
## [1] "run 971"
## [1] "run 972"
## [1] "run 973"
## [1] "run 974"
## [1] "run 975"
## [1] "run 976"
## [1] "run 977"
## [1] "run 978"
## [1] "run 979"
## [1] "run 980"
## [1] "run 981"
## [1] "run 982"
## [1] "run 983"
## [1] "run 984"
## [1] "run 985"
## [1] "run 986"
## [1] "run 987"
## [1] "run 988"
## [1] "run 989"
## [1] "run 990"
## [1] "run 991"
## [1] "run 992"
## [1] "run 993"
## [1] "run 994"
## [1] "run 995"
## [1] "run 996"
```

```
## [1] "run 997"
## [1] "run 998"
## [1] "run 999"
## [1] "run 1000"
## [1] "run 1001"
## [1] "run 1002"
## [1] "run 1003"
## [1] "run 1004"
## [1] "run 1005"
## [1] "run 1006"
## [1] "run 1007"
## [1] "run 1008"
## [1] "run 1009"
## [1] "run 1010"
## [1] "run 1011"
## [1] "run 1012"
## [1] "run 1013"
## [1] "run 1014"
## [1] "run 1015"
## [1] "run 1016"
## [1] "run 1017"
## [1] "run 1018"
## [1] "run 1019"
## [1] "run 1020"
## [1] "run 1021"
## [1] "run 1022"
## [1] "run 1023"
## [1] "run 1024"
```

```
Sys.time()-time1 ##see how long it takes!
```

```
## Time difference of 0.6356552 secs
```

```
################
##Save results##
################

save(x,y,results,models,file="results.RData")

######################
##Give results names##
######################

p<-results[,1]
R2<-results[,2]
R2.adj<-results[,3]
PRESS<-results[,4]
AIC<-results[,5]
BIC<-results[,6]
Cp<-results[,7]

###############################
##Find model with best R2.adj##
```

```r
################################

i<-R2.adj == max(R2.adj) ##what index has the maximum R2.adj
(1:2^10)[i] ##2^10 because 10 potential predictors
```

```
## [1] 489
```

```r
round(results[i,],3)
```

```
##      p     R2 R2.adj  PRESS    AIC    BIC     Cp
##  6.000  0.956  0.953 92.026 -7.954  7.617  2.057
```

```r
models[i,]
```

```
##    x1    x2    x3    x4    x5    x6    x7    x8    x9   x10
## FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
```

```r
############################################
##Find model with best PRESS, AIC, BIC, Cp##
############################################

i2<-PRESS == min(PRESS) ##what index has the min PRESS
(1:2^10)[i2] ##2^10 because 10 potential predictors
```

```
## [1] 481
```

```r
i3<-AIC == min(AIC) ##what index has the min AIC
(1:2^10)[i3] ##2^10 because 10 potential predictors
```

```
## [1] 481
```

```r
i4<-BIC == min(BIC) ##what index has the min BIC
(1:2^10)[i4] ##2^10 because 10 potential predictors
```

```
## [1] 481
```

```r
i5<-Cp == min(Cp) ##what index has the min Cp
(1:2^10)[i5] ##2^10 because 10 potential predictors
```

```
## [1] 481
```

```r
models[i2,]
```

```
##    x1    x2    x3    x4    x5    x6    x7    x8    x9   x10
## FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
```

```r
models[i3,]
```

```
##    x1    x2    x3    x4    x5    x6    x7    x8    x9   x10
## FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
```

```r
models[i4,]
```

```
##    x1    x2    x3    x4    x5    x6    x7    x8    x9   x10
## FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
```

```r
models[i5,]
```

```
##    x1    x2    x3    x4    x5    x6    x7    x8    x9   x10
## FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
```

```r
###########################################
##plot the residual against fitted values##
###########################################

dat489.lm=lm(y ~ x2+x3+x4+x5+x7)
summary(dat489.lm)
```

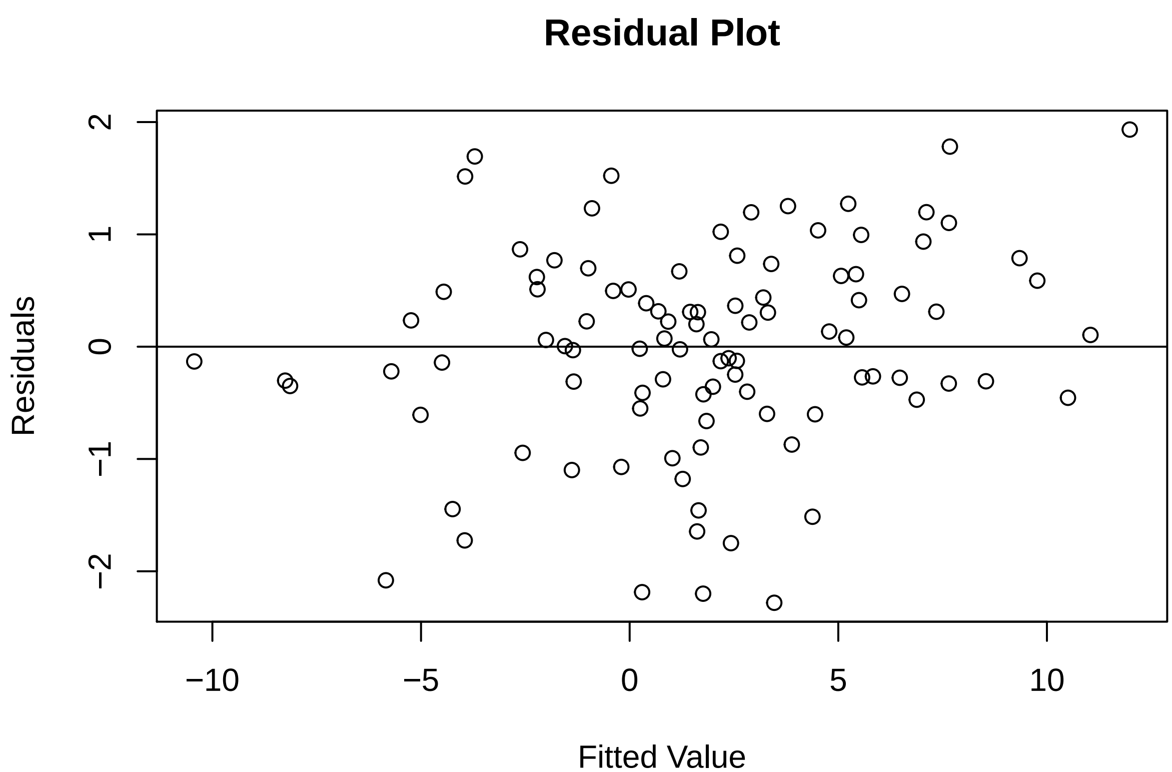
```
##
## Call:
## lm(formula = y ~ x2 + x3 + x4 + x5 + x7)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.28020 -0.41751  0.06503  0.60385  1.93313
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.49739    0.09634  15.543  < 2e-16 ***
## x2           1.99599    0.08820  22.629  < 2e-16 ***
## x3           0.90480    0.10114   8.946 3.47e-14 ***
## x4           0.39580    0.09006   4.395 2.94e-05 ***
## x5           3.33487    0.09857  33.832  < 2e-16 ***
## x7           0.11749    0.10791   1.089    0.279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9328 on 93 degrees of freedom
## Multiple R-squared:  0.9556, Adjusted R-squared:  0.9532
## F-statistic: 399.9 on 5 and 93 DF,  p-value: < 2.2e-16
```

```r
dat481.lm=lm(y ~ x2+x3+x4+x5)
summary(dat481.lm)
```

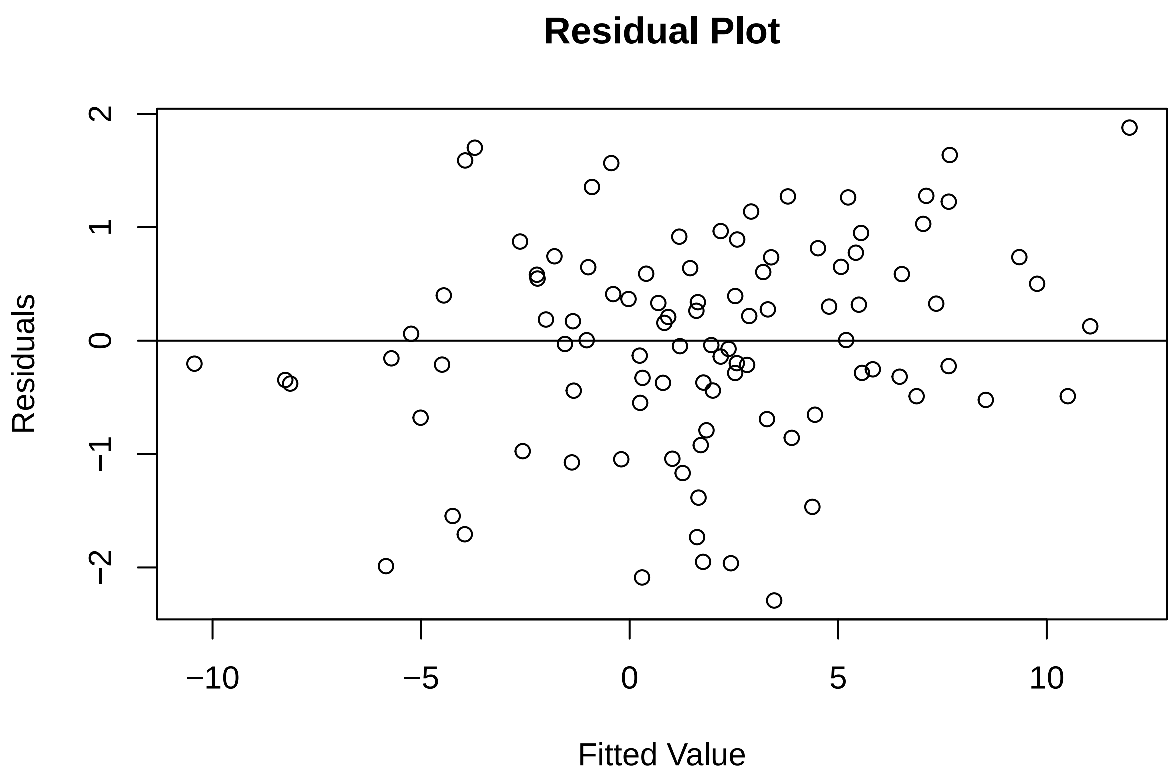
```
##
## Call:
```

```
## lm(formula = y ~ x2 + x3 + x4 + x5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.29129 -0.46534  0.00512  0.62212  1.87873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.50163    0.09635  15.584  < 2e-16 ***
## x2           1.99583    0.08829  22.605  < 2e-16 ***
## x3           0.91006    0.10112   8.999 2.46e-14 ***
## x4           0.40358    0.08986   4.491 2.01e-05 ***
## x5           3.34805    0.09792  34.191  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9338 on 94 degrees of freedom
## Multiple R-squared:  0.955,  Adjusted R-squared:  0.9531
## F-statistic: 498.6 on 4 and 94 DF,  p-value: < 2.2e-16
```

```
dat489.res=resid(dat489.lm)
par(mfrow=c(1,1))
plot(y, dat489.res, ylab="Residuals", xlab="Fitted Value", main="Residual Plot")
abline(0, 0)                    # the horizon
```




```
dat481.res=resid(dat481.lm)
par(mfrow=c(1,1))
plot(y, dat481.res, ylab="Residuals", xlab="Fitted Value", main="Residual Plot")
abline(0, 0)                    # the horizon
```




```r
anova(dat481.lm,dat489.lm)
```

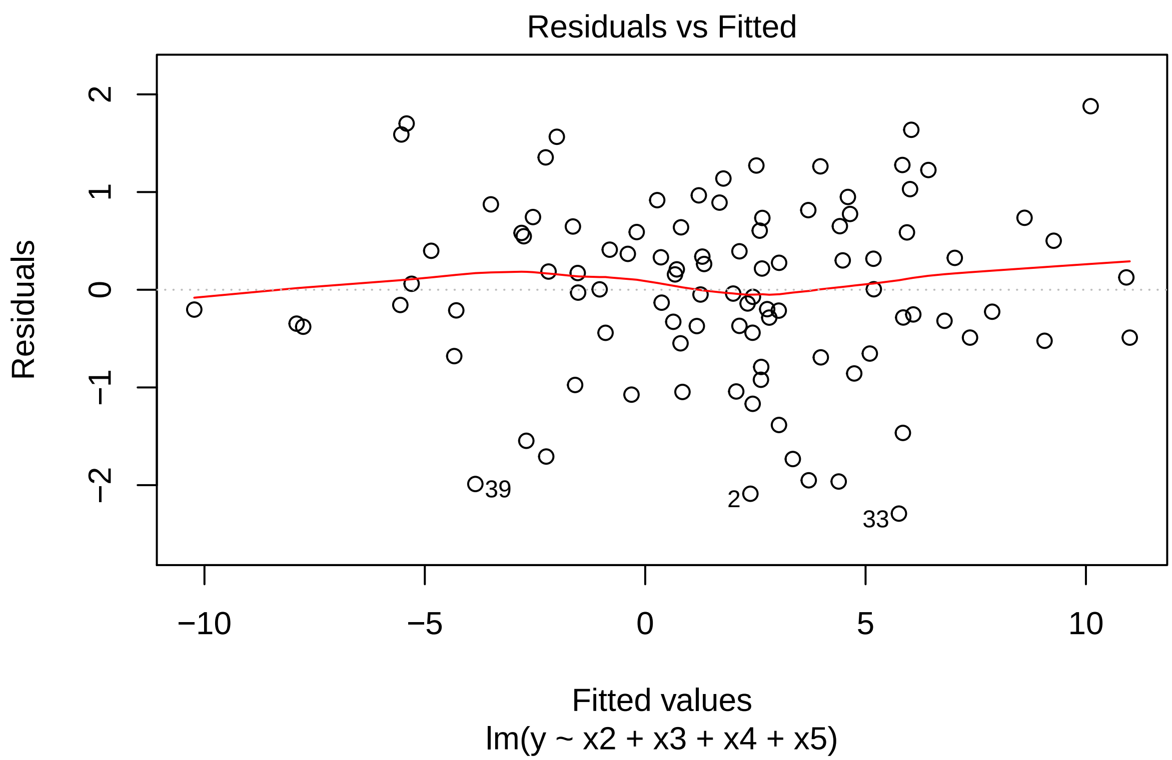
```
## Analysis of Variance Table
##
## Model 1: y ~ x2 + x3 + x4 + x5
## Model 2: y ~ x2 + x3 + x4 + x5 + x7
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     94 81.960
## 2     93 80.929  1    1.0317 1.1856  0.279
```

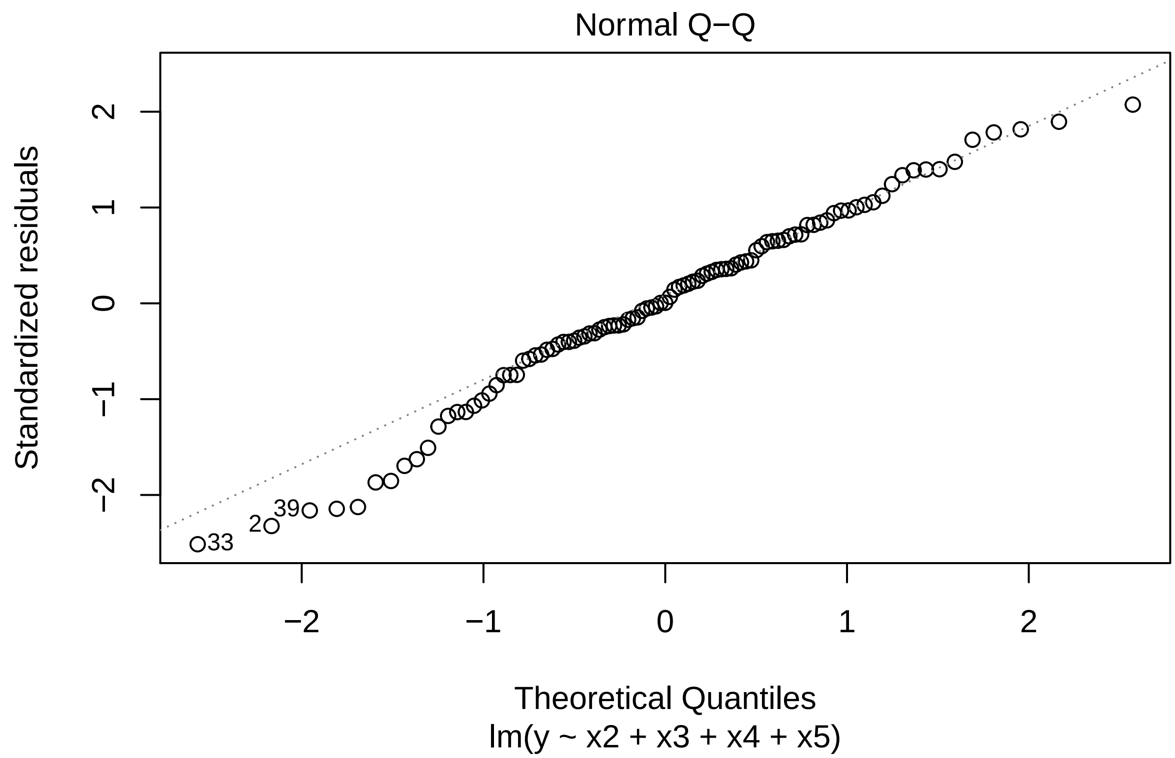
```r
result.lm=lm(y ~ x2+x3+x4+x5)
summary(result.lm)
```

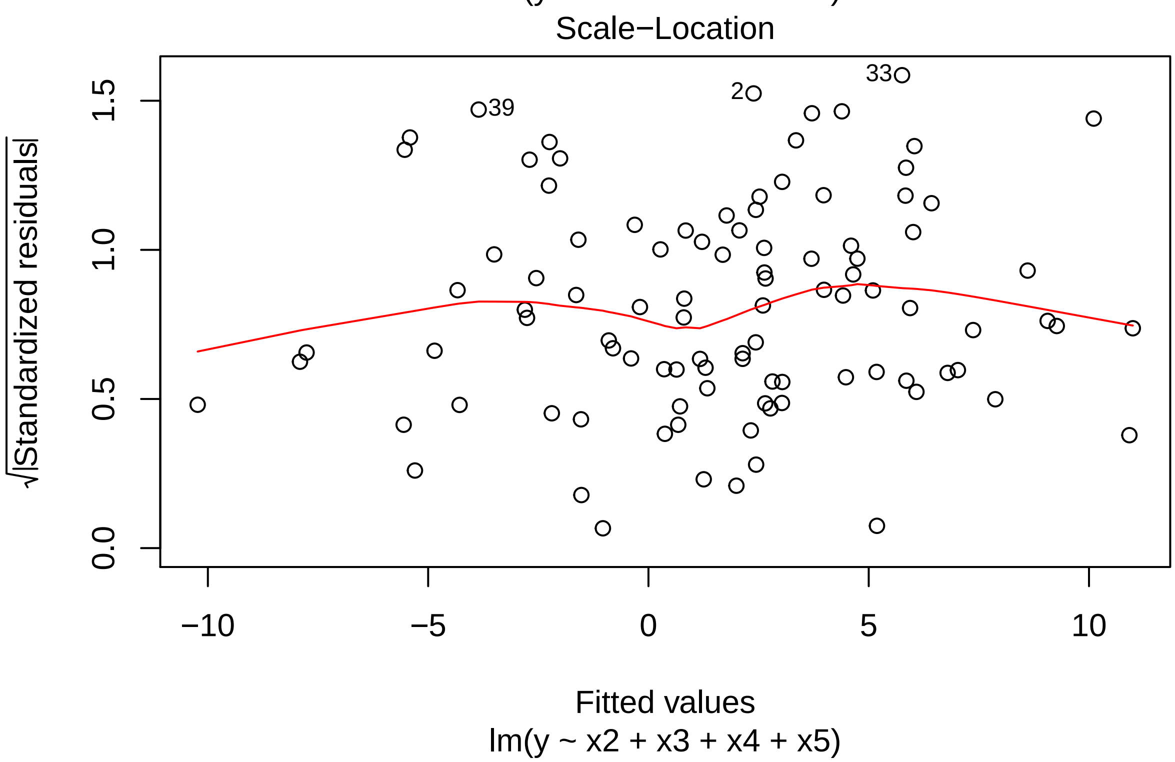
```
##
## Call:
## lm(formula = y ~ x2 + x3 + x4 + x5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.29129 -0.46534  0.00512  0.62212  1.87873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.50163    0.09635  15.584  < 2e-16 ***
## x2           1.99583    0.08829  22.605  < 2e-16 ***
## x3           0.91006    0.10112   8.999 2.46e-14 ***
## x4           0.40358    0.08986   4.491 2.01e-05 ***
## x5           3.34805    0.09792  34.191  < 2e-16 ***
```

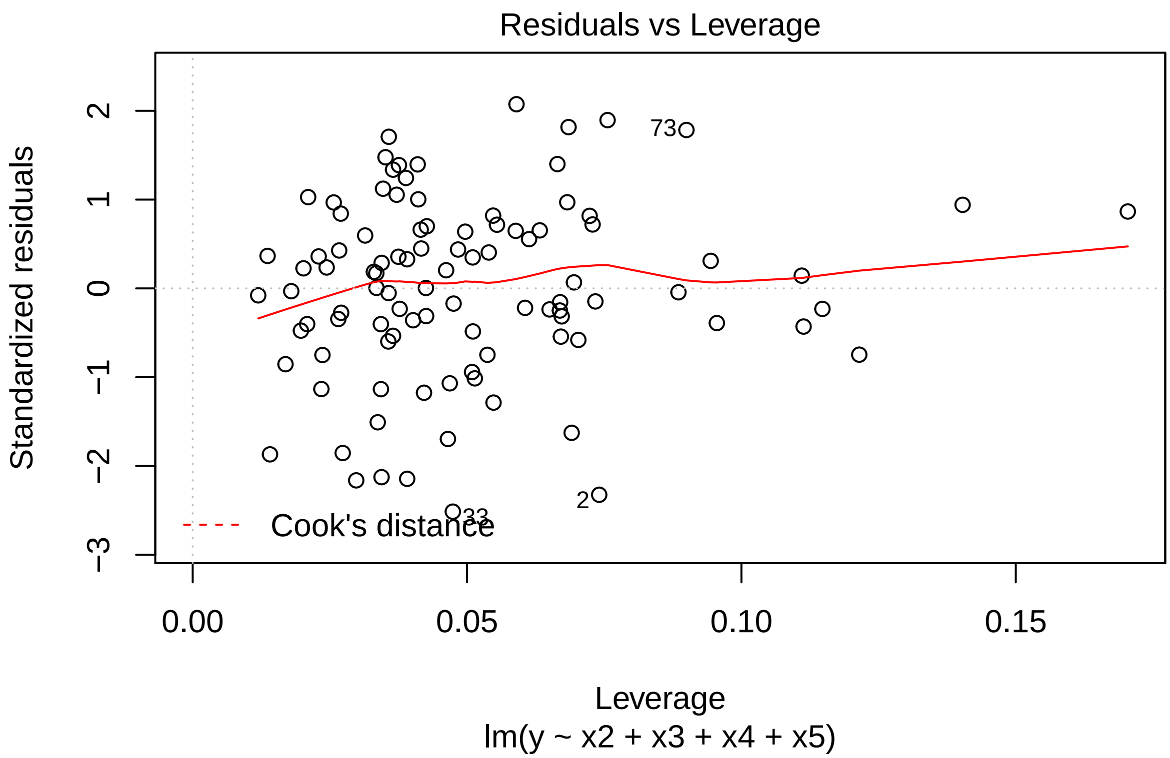
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9338 on 94 degrees of freedom
## Multiple R-squared:  0.955,  Adjusted R-squared:  0.9531
## F-statistic: 498.6 on 4 and 94 DF,  p-value: < 2.2e-16
```

```r
plot(result.lm)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(y ~ x2 + x3 + x4 + x5)
33
2
39
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
Fitted values
lm(y ~ x2 + x3 + x4 + x5)
33
2
39
0.0
0.5
1.0
1.5
−10
−5
0
5
10



Residuals vs Leverage
Standardized residuals
Leverage
lm(y ~ x2 + x3 + x4 + x5)
Cook's distance
73
2
33
0.00
0.05
0.10
0.15
−3
−2
−1
0
1
2