Ch 9 - Surgical Unit Example

Adam McQuistan

Sunday, May 01, 2016

The dataset

Variable	Description
X1	blood clotting score
X2	prognostic index
X3	enzyme function score
X4	liver function test score
X5	age in years
X6	gender, 0 for male and 1 for female
X7, X8	alcohol use. see table below
Y	Survival time

Alcohol Use	X7	X8
None	0	0
Moderate	1	0
Severe	0	1

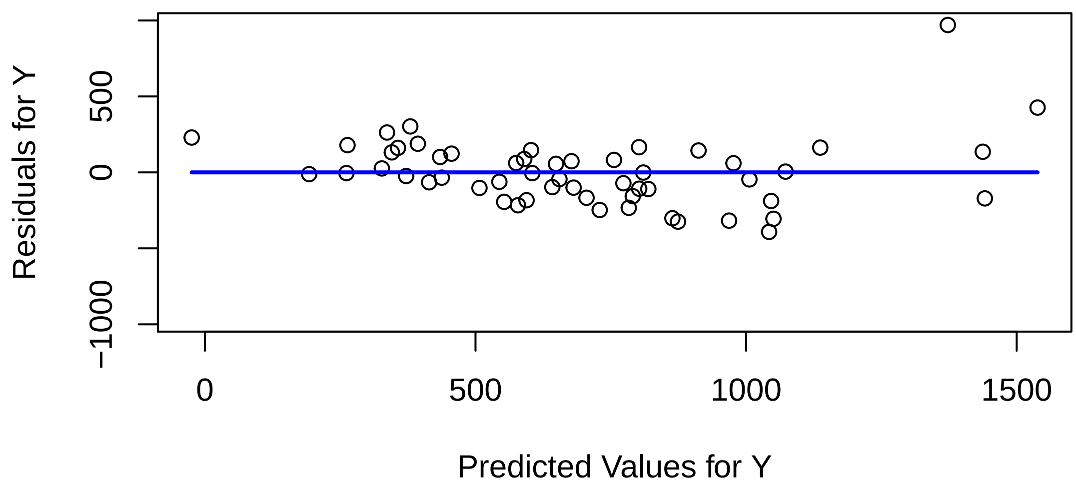
```
df <- read.table(file="CH09TA01.txt", sep="\t", header=F)
names(df) = c("X1","X2","X3","X4","X5","X6","X7","X8","Y", "lnY")
str(df)
```

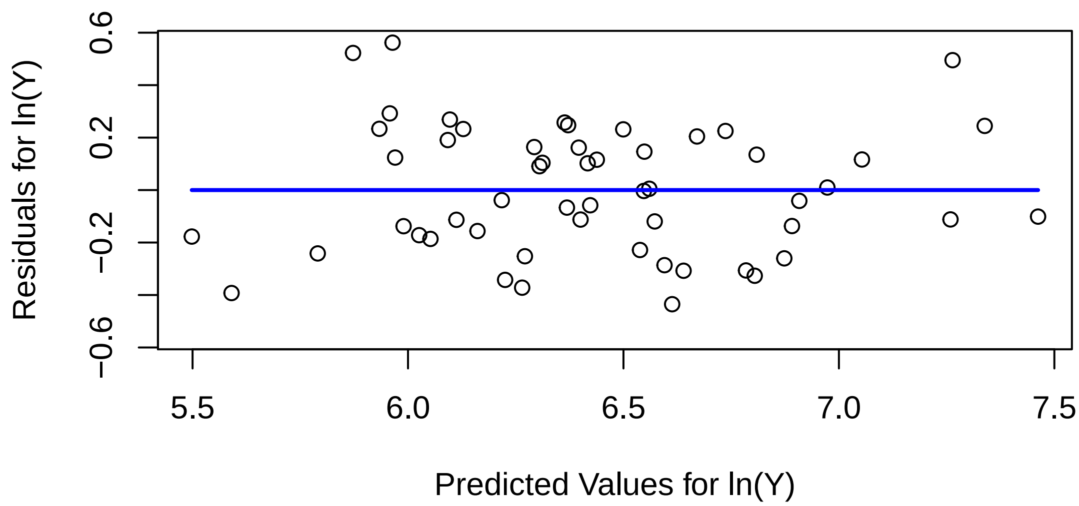
```
## 'data.frame':    54 obs. of  10 variables:
##  $ X1 : num  6.7 5.1 7.4 6.5 7.8 5.8 5.7 3.7 6 3.7 ...
##  $ X2 : int  62 59 57 73 65 38 46 68 67 76 ...
##  $ X3 : int  81 66 83 41 115 72 63 81 93 94 ...
##  $ X4 : num  2.59 1.7 2.16 2.01 4.3 1.42 1.91 2.57 2.5 2.4 ...
##  $ X5 : int  50 39 55 48 45 65 49 69 58 48 ...
##  $ X6 : int  0 0 0 0 0 1 1 1 0 0 ...
##  $ X7 : int  1 0 0 0 0 1 0 1 1 1 ...
##  $ X8 : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ Y  : int  695 403 710 349 2343 348 518 749 1056 968 ...
##  $ lnY: num  6.54 6 6.57 5.85 7.76 ...
```

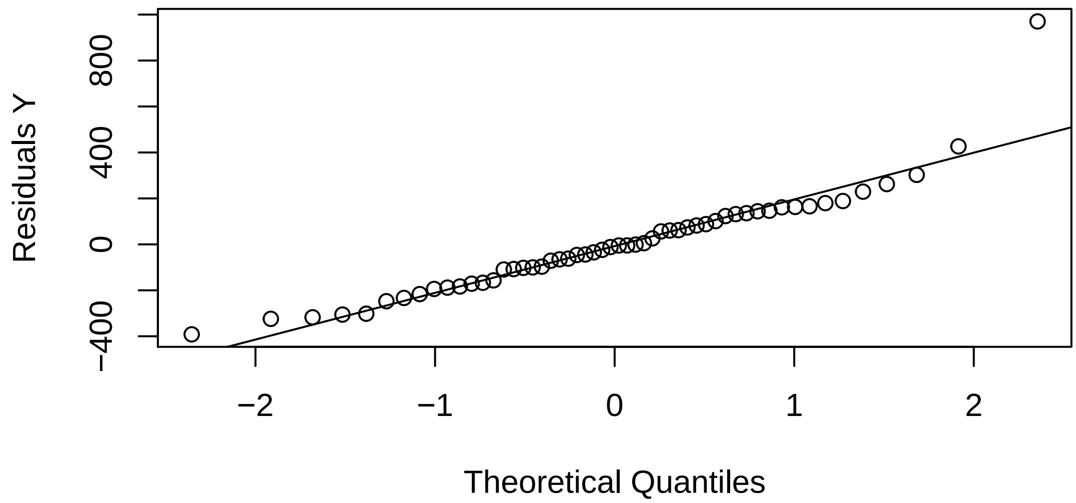
Subsetting the Data into First 54 Cases and First 4 Variables

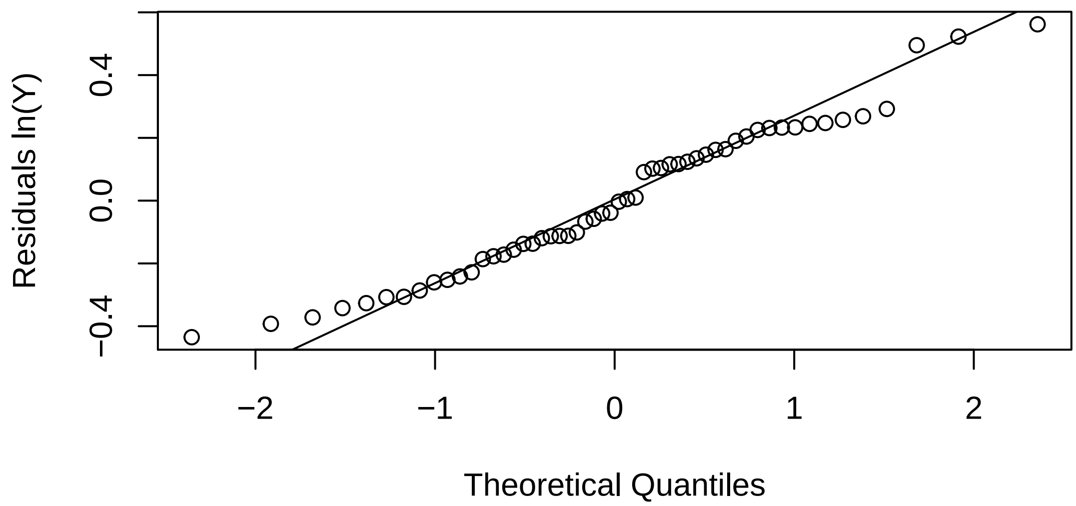
```
df54 <- df[1:54,c("X1","X2","X3","X4","Y","lnY")]
resultY <- lm(Y ~ X1 + X2 + X3 + X4, data=df)
resultlnY <- lm(lnY ~ X1 + X2 + X3 + X4, data=df)
```

Residual Plots of Y and lnY









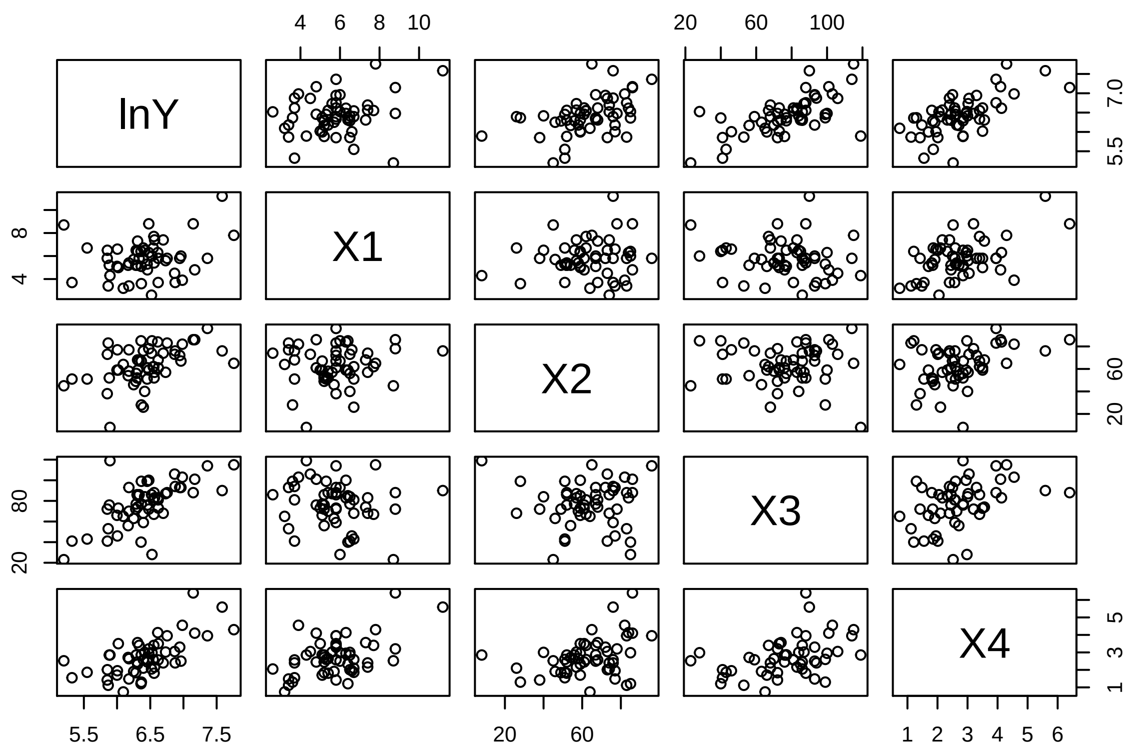
Correlations and Scatter Plot Matrix

```
with(df54, cor(df54[,c("lnY","X1","X2","X3","X4")]))
```

```
##           lnY         X1         X2         X3        X4
## lnY 1.0000000  0.24618787  0.46994325  0.65388548 0.6492627
```

```
## X1  0.2461879  1.00000000  0.09011973 -0.14963411 0.5024157
## X2  0.4699432  0.09011973  1.00000000 -0.02360544 0.3690256
## X3  0.6538855 -0.14963411 -0.02360544  1.00000000 0.4164245
## X4  0.6492627  0.50241567  0.36902563  0.41642451 1.0000000
```

```r
with(df54, pairs(df54[,c("lnY","X1","X2","X3","X4")]))
```



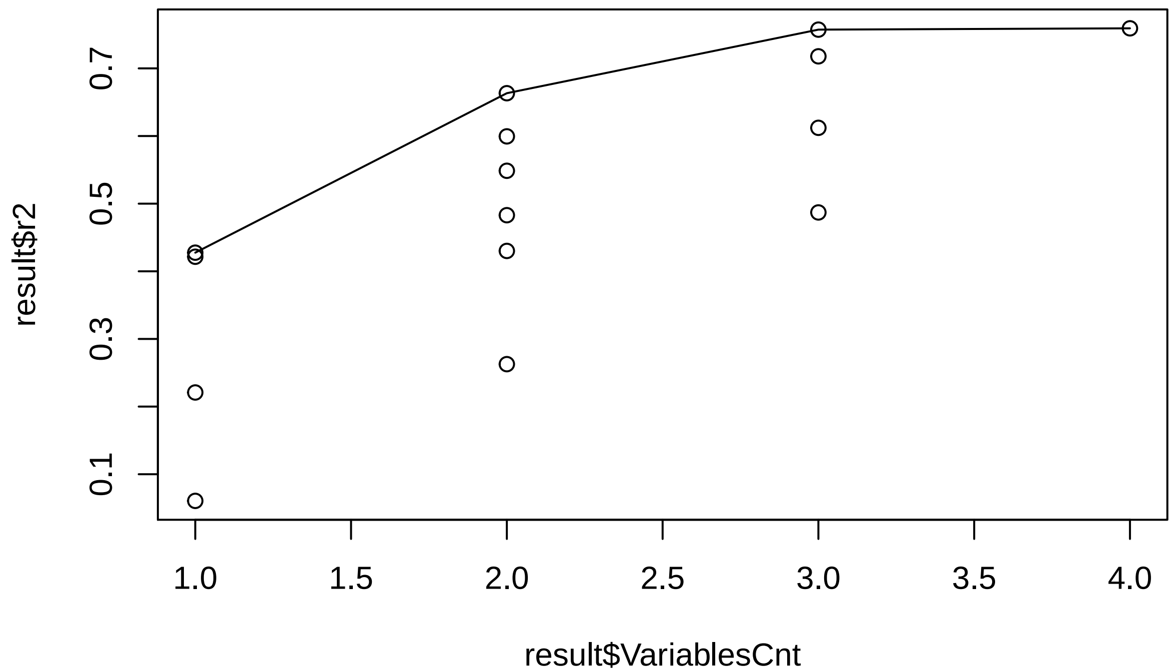
Using R_p^2 and SSE_p

```r
library(leaps)

evaluateRegressionModel <- function(x, y, method, names){
  result <- leaps(x=x, y=y,method=method,names=names)
  labels <- result$label[2:length(result$label)]

  Variables <- vector()
  VariablesCnt <- vector()
  metric <- vector()

  for(rowIdx in 1:dim(result$which)[1]){
    selected <- result$which[rowIdx,]
    VariablesCnt <- c(VariablesCnt, sum(result$which[rowIdx,]))
    vars <- paste(labels[selected], collapse=" ")
```

```r
    Variables <- c(Variables, vars)

    thisMetric <- switch(method,
                     r2=result$r2[rowIdx],
                     Cp=result$Cp[rowIdx],
                     adjr2=result$adjr2[rowIdx])

    metric <- c(metric, thisMetric)
  }

  out <- data.frame(Variables, VariablesCnt, metric)
  names(out)[3] = method
  print(out)
  return(out)
}

result <- evaluateRegressionModel(x=as.matrix(df54[,1:4]),
                                  y=df54$lnY,
                                  method="r2",
                                  names=names(df54)[1:4])
```

```
##      Variables VariablesCnt         r2
## 1           X3            1 0.42756622
## 2           X4            1 0.42154199
## 3           X2            1 0.22084666
## 4           X1            1 0.06060847
## 5        X2 X3            2 0.66328986
## 6        X3 X4            2 0.59948374
## 7        X1 X3            2 0.54863462
## 8        X2 X4            2 0.48296742
## 9        X1 X4            2 0.43010550
## 10       X1 X2            2 0.26273627
## 11    X1 X2 X3            3 0.75729185
## 12    X2 X3 X4            3 0.71781636
## 13    X1 X3 X4            3 0.61212320
## 14    X1 X2 X4            3 0.48701249
## 15 X1 X2 X3 X4            4 0.75921083
```

```r
library(dplyr)
df_tbl <- tbl_df(result) %>%
            group_by(VariablesCnt) %>%
            summarize(MaxR2 = max(r2))

plot(x=result$VariablesCnt, y=result$r2)
lines(df_tbl$VariablesCnt, df_tbl$MaxR2)
```

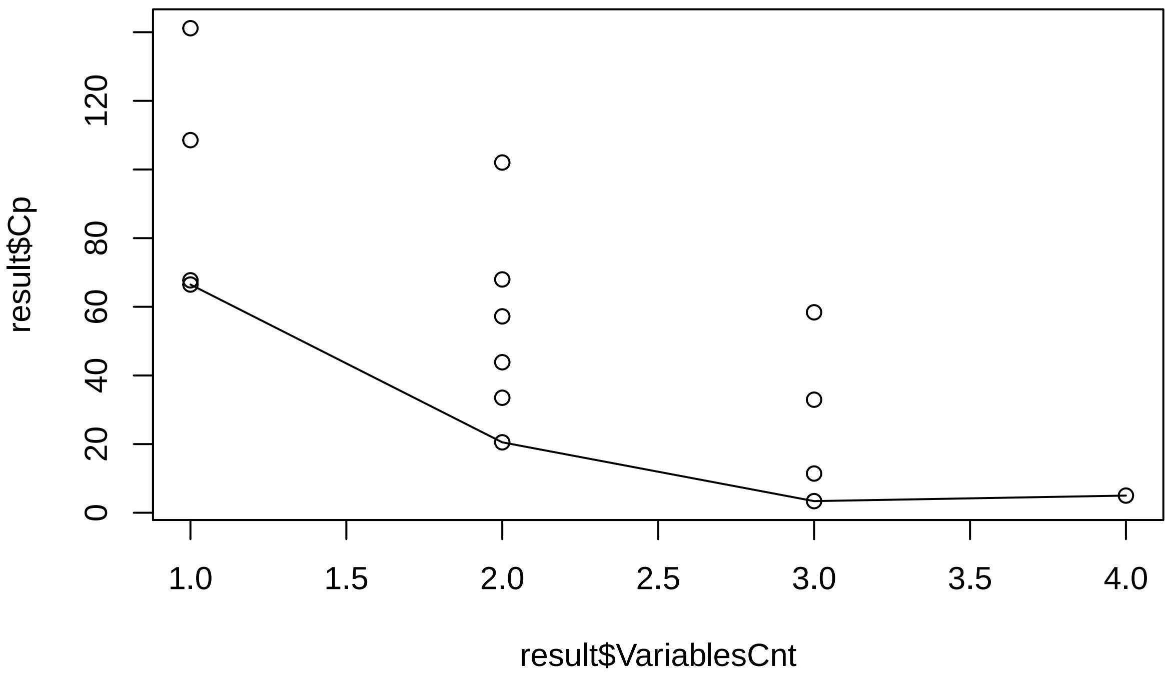


Mallows C_p Criterion

```
result <- evaluateRegressionModel(x=as.matrix(df54[,1:4]),
      y=df54$lnY,
      method="Cp",
      names=names(df54)[1:4])
```

```
##       Variables VariablesCnt         Cp
## 1            X3            1  66.488856
## 2            X4            1  67.714773
## 3            X2            1 108.555776
## 4            X1            1 141.163851
## 5         X2 X3            2  20.519679
## 6         X3 X4            2  33.504067
## 7         X1 X3            2  43.851738
## 8         X2 X4            2  57.214850
## 9         X1 X4            2  67.972119
## 10        X1 X2            2 102.031343
## 11     X1 X2 X3            3   3.390508
## 12     X2 X3 X4            3  11.423673
## 13     X1 X3 X4            3  32.931969
## 14     X1 X2 X4            3  58.391689
## 15 X1 X2 X3 X4            4   5.000000
```

```r
df_tbl <- tbl_df(result) %>%
            group_by(VariablesCnt) %>%
            summarize(MinCp = min(Cp))

plot(x=result$VariablesCnt, y=result$Cp)
lines(df_tbl$VariablesCnt, df_tbl$MinCp)
```



Stepwise Regression - Forward

For forward stepwise regression it is important to identify an α cut off for determining which predictors to let into the model. For example, if your cut of is 0.05 then you would only include variables with pvalues below the variable.

```r
library(MASS)

full <- lm(lnY ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8,
           data=df)
Null <- lm(lnY ~ 1, data=df)
addterm(Null, scope=full, test="F")
```

```
## Single term additions
##
## Model:
```

```
## lnY ~ 1
##        Df Sum of Sq     RSS      AIC F Value     Pr(F)
## <none>               12.8077  -75.703
## X1      1    0.7763 12.0315  -77.079   3.355 0.0727328 .
## X2      1    2.8285  9.9792  -87.178  14.739 0.0003366 ***
## X3      1    5.4762  7.3316 -103.827  38.840 8.261e-08 ***
## X4      1    5.3990  7.4087 -103.262  37.894 1.092e-07 ***
## X5      1    0.2691 12.5386  -74.849   1.116 0.2956212
## X6      1    0.6897 12.1180  -76.692   2.960 0.0913204 .
## X7      1    0.2052 12.6025  -74.575   0.847 0.3616983
## X8      1    1.7798 11.0279  -81.782   8.392 0.0055015 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For this iteration we would include the predictor with the lowest p value which is X3. A new base model is built by including the X3 value then the procedure is ran again.

```
newModel <- lm(lnY ~ X3, data=df)
addterm(newModel, scope=full, test="F")
```

```
## Single term additions
##
## Model:
## lnY ~ X3
##        Df Sum of Sq    RSS     AIC F Value     Pr(F)
## <none>              7.3316 -103.83
## X1      1   1.55061 5.7810 -114.66  13.680 0.0005312 ***
## X2      1   3.01908 4.3125 -130.48  35.704 2.242e-07 ***
## X4      1   2.20187 5.1297 -121.11  21.891 2.161e-05 ***
## X5      1   0.23877 7.0928 -103.61   1.717 0.1959722
## X6      1   0.25854 7.0730 -103.77   1.864 0.1781349
## X7      1   0.06498 7.2666 -102.31   0.456 0.5025196
## X8      1   1.13756 6.1940 -110.93   9.366 0.0035199 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now I'll add in X2

```
newModel <- lm(lnY ~ X3 + X2, data=df)
addterm(newModel, scope=full, test="F")
```

```
## Single term additions
##
## Model:
## lnY ~ X3 + X2
##        Df Sum of Sq    RSS     AIC F Value     Pr(F)
## <none>              4.3125 -130.48
## X1      1   1.20395 3.1085 -146.16 19.3652 5.670e-05 ***
## X4      1   0.69836 3.6141 -138.02  9.6615  0.003102 **
## X5      1   0.16461 4.1479 -130.59  1.9843  0.165127
## X6      1   0.08245 4.2300 -129.53  0.9745  0.328307
## X7      1   0.22632 4.0862 -131.39  2.7693  0.102341
```

```
## X8       1   1.46961 2.8429 -150.99 25.8471 5.558e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This time X1 is added

```r
newModel <- lm(lnY ~ X3 + X2 + X1, data=df)
addterm(newModel, scope=full, test="F")
```

```
## Single term additions
##
## Model:
## lnY ~ X3 + X2 + X1
##        Df Sum of Sq    RSS     AIC F Value     Pr(F)
## <none>              3.1085 -146.16
## X4      1   0.02458 3.0840 -144.59  0.3905    0.5349
## X5      1   0.14838 2.9602 -146.80  2.4561    0.1235
## X6      1   0.05202 3.0565 -145.07  0.8339    0.3656
## X7      1   0.11790 2.9906 -146.25  1.9316    0.1709
## X8      1   0.92974 2.1788 -163.35 20.9094 3.291e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This time through the only variable making the cut off is X8. Adding this gives us our full model.

```r
newModel <- lm(lnY ~ X3 + X2 + X1 + X8, data=df)
addterm(newModel, scope=full, test="F")
```

```
## Single term additions
##
## Model:
## lnY ~ X3 + X2 + X1 + X8
##        Df Sum of Sq    RSS     AIC F Value  Pr(F)
## <none>              2.1788 -163.35
## X4      1  0.041701 2.1371 -162.40 0.93662 0.3380
## X5      1  0.075876 2.1029 -163.26 1.73190 0.1944
## X6      1  0.096791 2.0820 -163.81 2.23149 0.1418
## X7      1  0.022944 2.1559 -161.92 0.51085 0.4782
```