Model Selection

We created 100 explanatory variables and a response variable where the true model is linear in the first 5 of these variables.
The sample size is 500.

We fit a bunch of models to the data.

```
mod0 = lm(y~X[,1])
mod1 = lm(y~X[,1:3])
mod1.4 = lm(y~X[,1:4])
mod1.5 = lm(y~X[,1:5])
mod1.6 = lm(y~X[,1:6])
mod1.7 = lm(y~X[,1:7])
mod1.8 = lm(y~X[,1:8])
mod1.9 = lm(y~X[,1:9])
mod2 = lm(y~X[,1:10])
mod2.5 = lm(y~X[,1:20])
mod3 = lm(y~X[,1:30])
mod4 = lm(y~X[,1:50])
mod5 = lm(y~X)

modNames = c("mod0","mod1","mod1.4", "mod1.5", "mod1.6", "mod1.7",
             "mod1.8", "mod1.9", "mod2", "mod2.5", "mod3", "mod4", "mod5")
```

Here is a function that we can use to plot the model selection criteria against the model size.

```
twoPlots = function(crit, main) {
  axLab = c("x1","x1-x3", "x1-x4", "x1-x5", "x1-x6", "x1-x7", "x1-x8",
          "x1-x9", "x1-x10","x1-x20", "x1-x30", "x1-x50", "all")
  ylab = "Criteria"
  xlab = "Model"

  plot(crit, ylab = ylab, xlab = xlab,
       main = main, xaxt = "n")
  axis(1, at = 1:13, axLab)

  plot(4:13, crit[ 4:13 ], ylab = ylab, xlab = xlab,
       main = main, xaxt = "n")
  axis(1, at = 1:13, axLab)
}
```

Now we examine the various model selection criteria, including Mallow's Cp, AIC, BIC, and leave-one-out cross validation.

Mallow's C_p

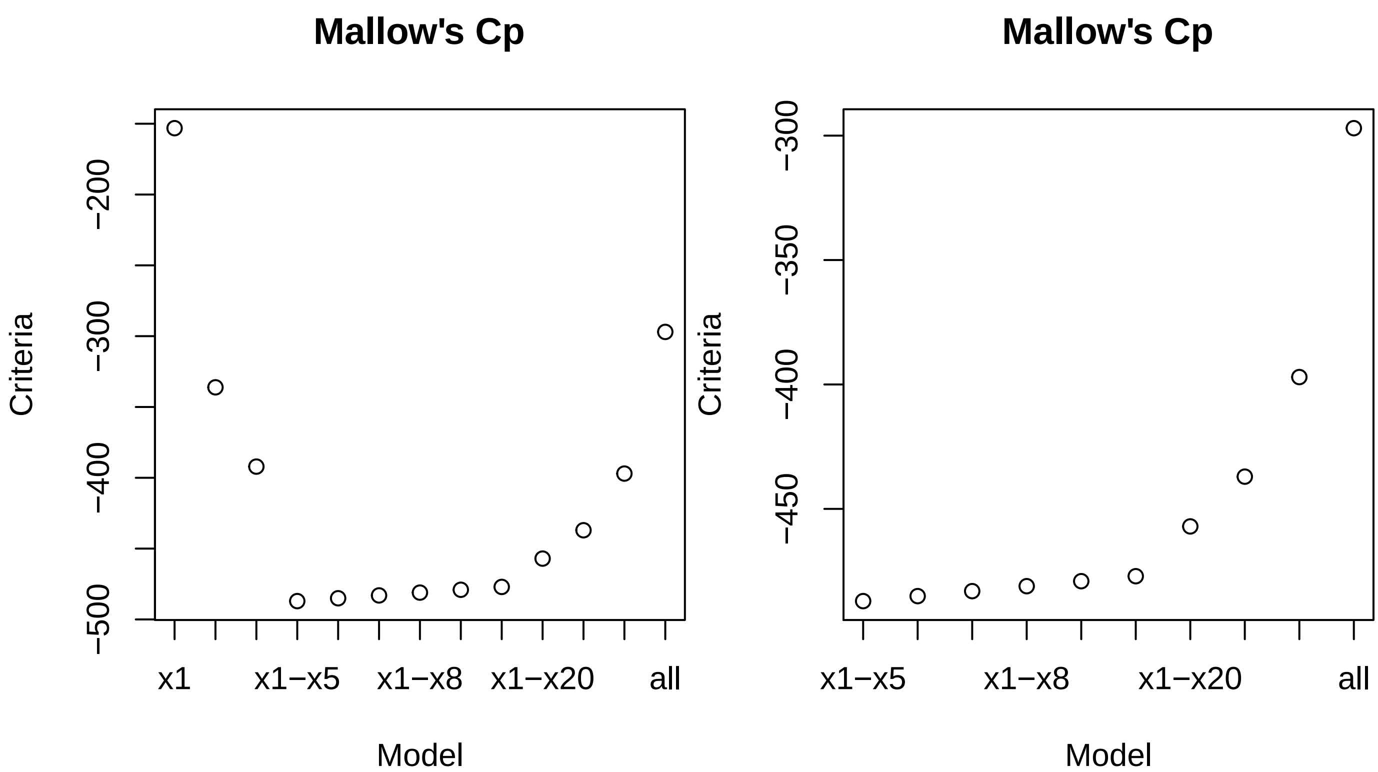
We calculate Mallow's Cp for each of the models that we fit. Recall the alternative representation of Mallow's Cp as

$$2(p(m) + 1) - n + ErrSS(m)/s_e^2$$

```r
sig2hatFull = summary(mod5)$sigma^2

cpout = sapply(modNames, function(name){
  obj = get(name)
  modDF = length(obj$coefficients)
  2*modDF - n + summary(obj)$sigma^2 /sig2hatFull
                   })

twoPlots(cpout, "Mallow's Cp")
```

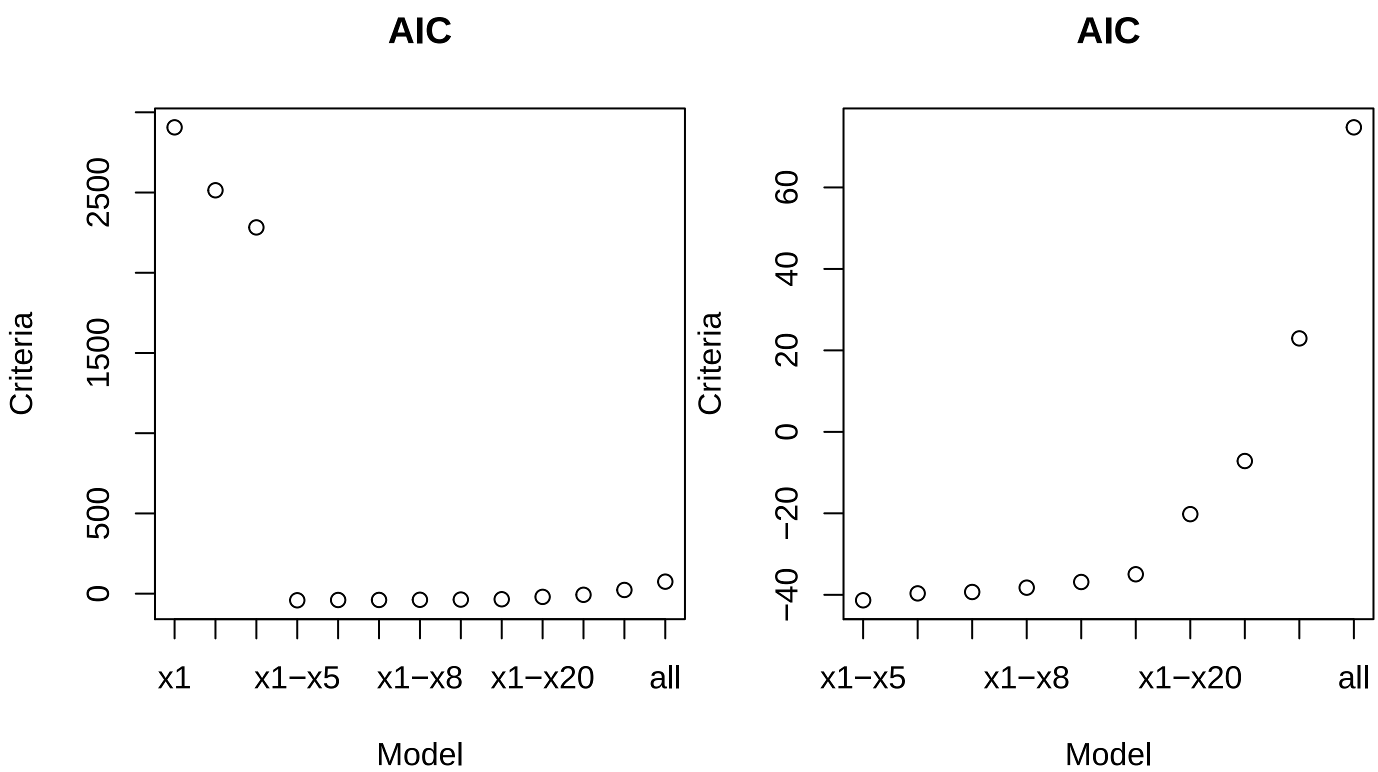


AIC

Find AIC for these various models.

```r
aicout = t(sapply(modNames, function(name){
                   extractAIC(get(name))
                   }))

twoPlots(aicout[ , 2], "AIC")
```

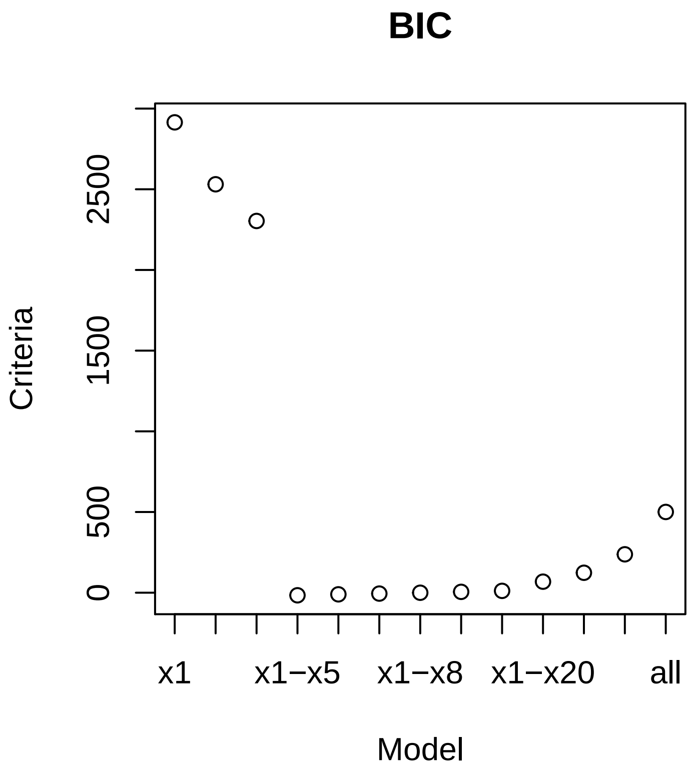


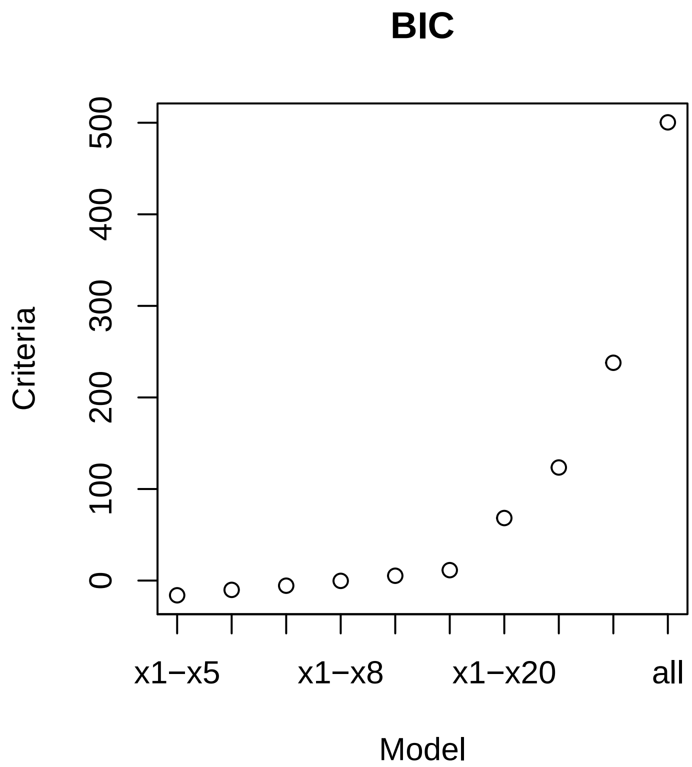
BIC

Find BIC for various models.

```
bicout = t(sapply(modNames, function(name){
                   extractAIC(get(name), k = log(n))
                   }))

twoPlots(bicout[ , 2], "BIC")
```





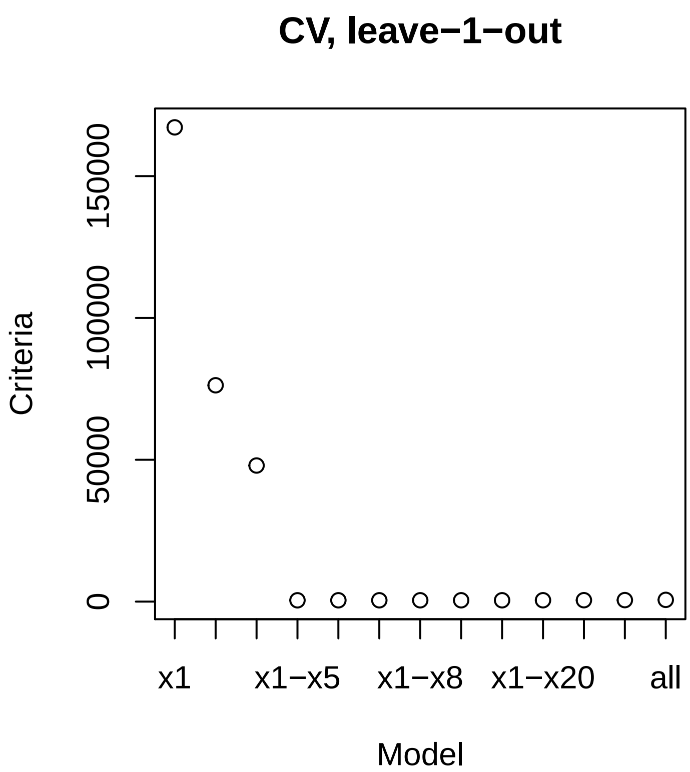
Leave one out CV

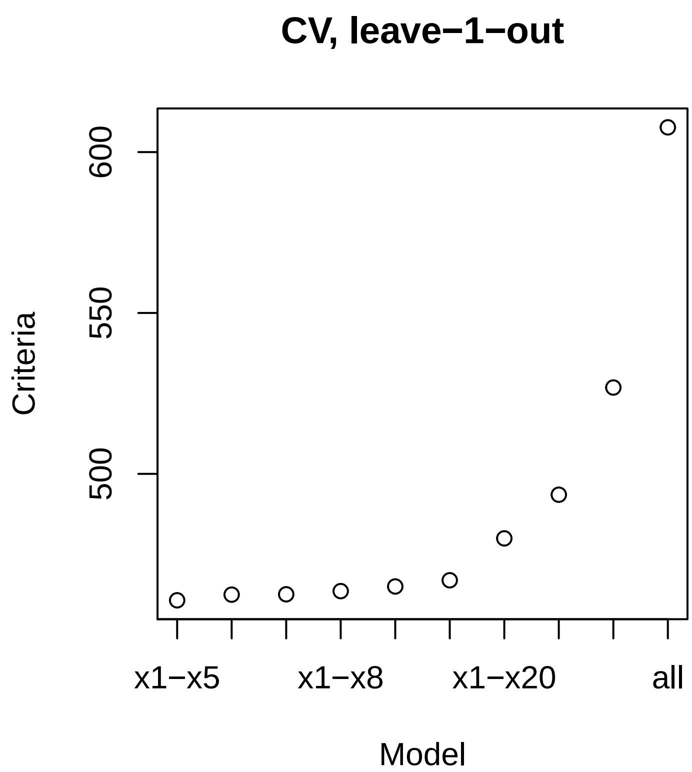
For each model, we fit the model with least squares, we find the residual sum of squares for the model, and we find the hat values for each observations. These give us the leave-one-out cross-validated residual sum of squares.

```
modelVars = list(1, 1:3, 1:4, 1:5, 1:6, 1:7, 1:8,
                 1:9, 1:10, 1:20, 1:30, 1:50, 1:100)

cvleave = sapply(modelVars, function(x){
  obj = lm(y ~ X[ , x, drop = FALSE])
    sum(residuals(obj)^2/ (1-hatvalues(obj))^2)
})

twoPlots(cvleave, "CV, leave-1-out")
```





Best subset regression

Rather than compare the 13 models that we constructed earlier, we can examine all possible models based on all possible subsets of variables. (Note this is problematic for categorical explanatory variables). We start with the Cp for all subsets of the full model. The leaps() function reports Mallow's Cp for the 10 best models for each model size. Included in the return value are the variables that belong to the 10 best models.

```
outs = leaps(x = X[,1:30], y = y,
            int = FALSE, strictly.compatible = FALSE)
```

Here are the top 10 one-variable models and the top ten two-variable models. The TRUE and FALSE indicate which variables belong in the model. Note that we show only the first 12 of the 30 variables here.

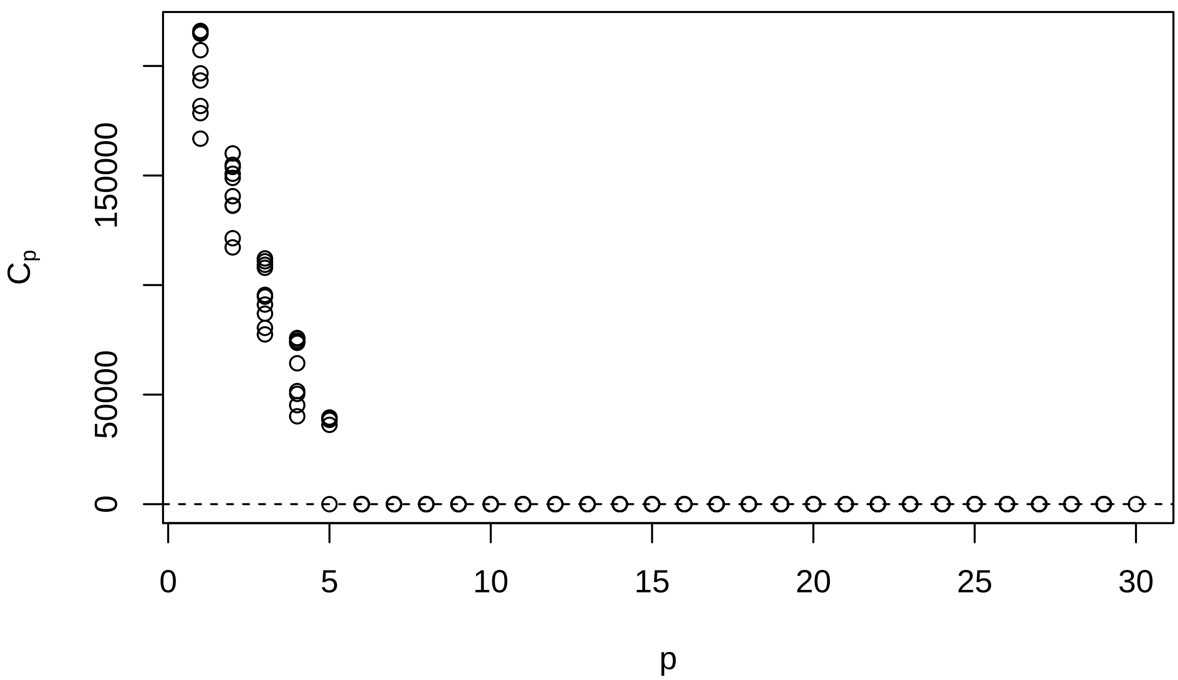
```
outs$which[1:20, 1:12]
```

```
##      X1    X2    X3    X4    X5    X6    X7    X8    X9   X10   X11   X12
## 1 FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1 FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1 FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1 FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1 FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 1 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 1 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

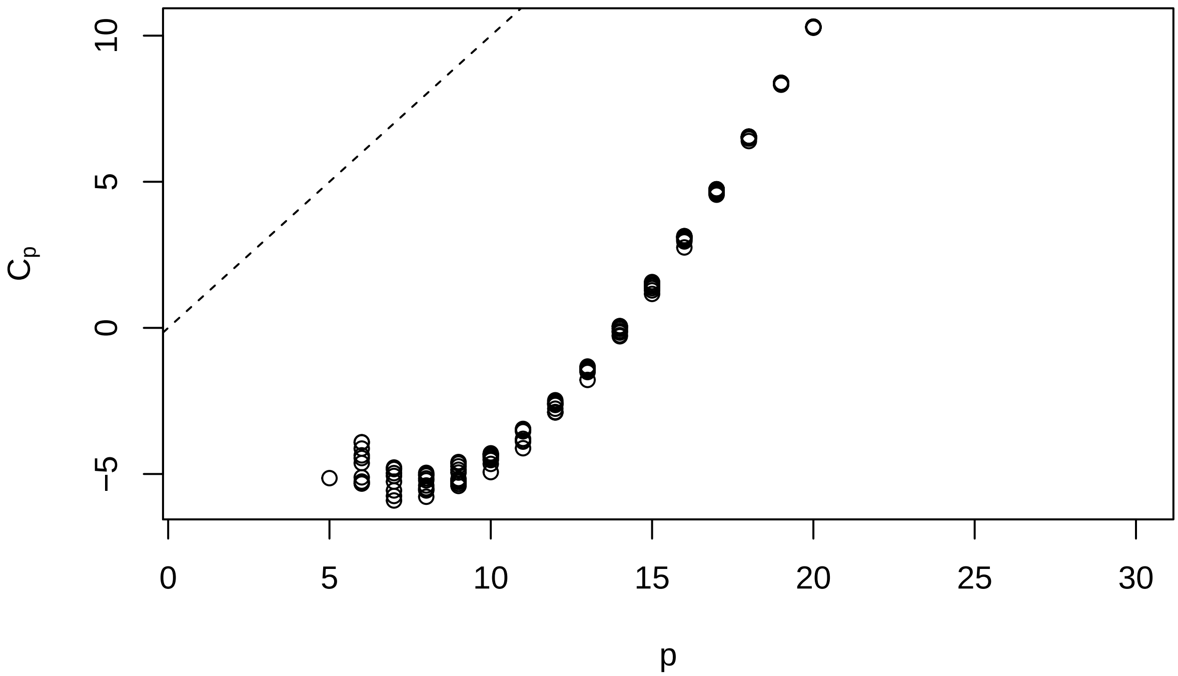
```
## 2 FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 2  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Plot the Cp for the best 10 models for each model size.

```
plot(outs$size, outs$Cp,  xlab = "p", ylab = expression(C[p]))
abline(a = 0, b = 1, lty = 2, untf = TRUE)
```




```
plot(outs$size, outs$Cp,  xlab = "p",
     ylab = expression(C[p]),ylim=c(min(outs$Cp),median(outs$Cp)))
abline(a = 0, b = 1, lty = 2, untf=TRUE)
```




```
mydata = data.frame(y, X[ , 1:20])
leaps = regsubsets(y ~ ., data = mydata, nbest = 7)

summary(leaps)
```

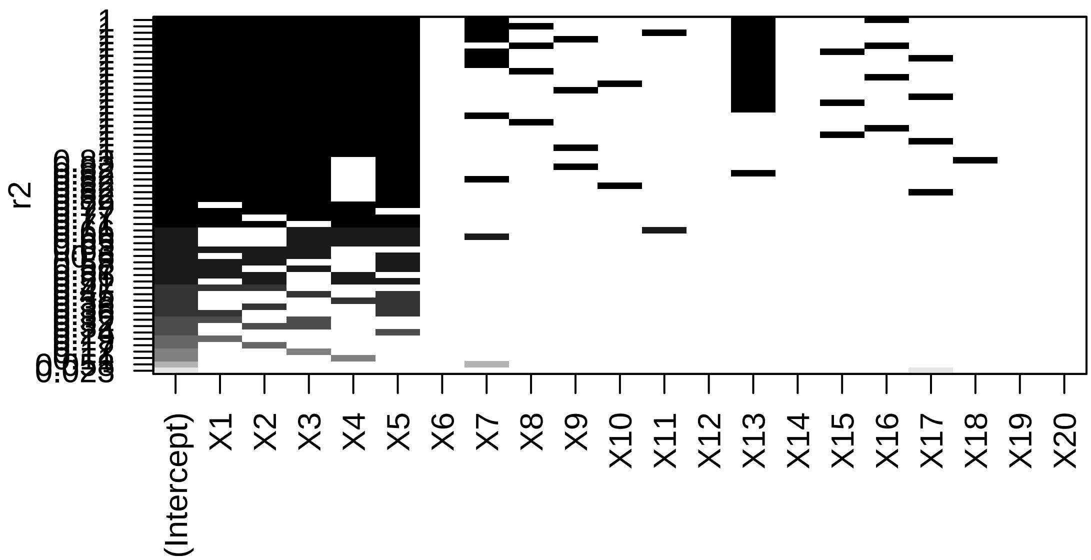
```
## Subset selection object
## Call: regsubsets.formula(y ~ ., data = mydata, nbest = 7)
## 20 Variables  (and intercept)
##     Forced in Forced out
## X1      FALSE      FALSE
## X2      FALSE      FALSE
## X3      FALSE      FALSE
## X4      FALSE      FALSE
## X5      FALSE      FALSE
## X6      FALSE      FALSE
## X7      FALSE      FALSE
## X8      FALSE      FALSE
## X9      FALSE      FALSE
## X10     FALSE      FALSE
## X11     FALSE      FALSE
## X12     FALSE      FALSE
## X13     FALSE      FALSE
## X14     FALSE      FALSE
## X15     FALSE      FALSE
## X16     FALSE      FALSE
## X17     FALSE      FALSE
## X18     FALSE      FALSE
## X19     FALSE      FALSE
## X20     FALSE      FALSE
## 7 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          X1  X2  X3  X4  X5  X6  X7  X8  X9  X10 X11 X12 X13 X14 X15 X16
```

```
## 1  ( 1 ) " " " " " " " " "*" " " " " " " " " " " " " " " " " " " " " " "
## 1  ( 2 ) "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 1  ( 3 ) " " "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 1  ( 4 ) " " " " "*" " " " " " " " " " " " " " " " " " " " " " " " " " "
## 1  ( 5 ) " " " " " " "*" " " " " " " " " " " " " " " " " " " " " " " " "
## 1  ( 6 ) " " " " " " " " " " " " "*" " " " " " " " " " " " " " " " " " "
## 1  ( 7 ) " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 2  ( 1 ) "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " " " "
## 2  ( 2 ) " " " " "*" " " "*" " " " " " " " " " " " " " " " " " " " " " "
## 2  ( 3 ) " " " " " " "*" "*" " " " " " " " " " " " " " " " " " " " " " "
## 2  ( 4 ) " " "*" " " " " "*" " " " " " " " " " " " " " " " " " " " " " "
## 2  ( 5 ) "*" " " " " " " "*" " " " " " " " " " " " " " " " " " " " " " "
## 2  ( 6 ) "*" " " "*" " " " " " " " " " " " " " " " " " " " " " " " " " "
## 2  ( 7 ) " " "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " "
## 3  ( 1 ) " " " " "*" "*" "*" " " " " " " " " " " " " " " " " " " " " " "
## 3  ( 2 ) "*" "*" "*" " " " " " " " " " " " " " " " " " " " " " " " " " "
## 3  ( 3 ) " " "*" "*" " " "*" " " " " " " " " " " " " " " " " " " " " " "
## 3  ( 4 ) "*" "*" " " " " "*" " " " " " " " " " " " " " " " " " " " " " "
## 3  ( 5 ) "*" " " "*" " " "*" " " " " " " " " " " " " " " " " " " " " " "
## 3  ( 6 ) "*" "*" " " "*" " " " " " " " " " " " " " " " " " " " " " " " "
## 3  ( 7 ) " " "*" " " "*" "*" " " " " " " " " " " " " " " " " " " " " " "
## 4  ( 1 ) "*" "*" "*" " " "*" " " " " " " " " " " " " " " " " " " " " " "
## 4  ( 2 ) " " "*" "*" "*" "*" " " " " " " " " " " " " " " " " " " " " " "
## 4  ( 3 ) "*" "*" "*" "*" " " " " " " " " " " " " " " " " " " " " " " " "
## 4  ( 4 ) "*" " " "*" "*" "*" " " " " " " " " " " " " " " " " " " " " " "
## 4  ( 5 ) "*" "*" " " "*" "*" " " " " " " " " " " " " " " " " " " " " " "
## 4  ( 6 ) " " " " "*" "*" "*" " " " " " " " " " " "*" " " " " " " " " " "
## 4  ( 7 ) " " " " "*" "*" "*" " " "*" " " " " " " " " " " " " " " " " " "
## 5  ( 1 ) "*" "*" "*" "*" "*" " " " " " " " " " " " " " " " " " " " " " "
## 5  ( 2 ) "*" "*" "*" " " "*" " " " " " " " " " " " " " " " " " " " " " "
## 5  ( 3 ) "*" "*" "*" " " "*" " " " " " " "*" " " " " " " " " " " " " " "
## 5  ( 4 ) "*" "*" "*" " " "*" " " " " " " " " " " " " " " "*" " " " " " "
## 5  ( 5 ) "*" "*" "*" " " "*" " " "*" " " " " " " " " " " " " " " " " " "
## 5  ( 6 ) "*" "*" "*" " " "*" " " " " " " " " "*" " " " " " " " " " " " "
## 5  ( 7 ) "*" "*" "*" " " "*" " " " " " " " " " " " " " " " " " " " " " "
## 6  ( 1 ) "*" "*" "*" "*" "*" " " " " " " " " " " " " " " "*" " " " " " "
## 6  ( 2 ) "*" "*" "*" "*" "*" " " "*" " " " " " " " " " " " " " " " " " "
## 6  ( 3 ) "*" "*" "*" "*" "*" " " " " "*" " " " " " " " " " " " " " " " "
## 6  ( 4 ) "*" "*" "*" "*" "*" " " " " " " " " " " " " " " " " " " " " "*"
## 6  ( 5 ) "*" "*" "*" "*" "*" " " " " " " " " " " " " " " " " " " "*" " "
## 6  ( 6 ) "*" "*" "*" "*" "*" " " " " " " " " " " " " " " " " " " " " " "
## 6  ( 7 ) "*" "*" "*" "*" "*" " " " " " " "*" " " " " " " " " " " " " " "
## 7  ( 1 ) "*" "*" "*" "*" "*" " " "*" " " " " " " " " " " "*" " " " " " "
## 7  ( 2 ) "*" "*" "*" "*" "*" " " " " "*" " " " " " " " " "*" " " " " " "
## 7  ( 3 ) "*" "*" "*" "*" "*" " " " " " " " " " " " " " " "*" " " " " "*"
## 7  ( 4 ) "*" "*" "*" "*" "*" " " " " " " " " "*" " " " " "*" " " " " " "
## 7  ( 5 ) "*" "*" "*" "*" "*" " " " " " " "*" " " " " " " "*" " " " " " "
## 7  ( 6 ) "*" "*" "*" "*" "*" " " " " " " " " " " " " " " "*" " " " " " "
## 7  ( 7 ) "*" "*" "*" "*" "*" " " " " " " " " " " " " " " "*" " " "*" " "
## 8  ( 1 ) "*" "*" "*" "*" "*" " " "*" " " " " " " " " " " "*" " " " " "*"
## 8  ( 2 ) "*" "*" "*" "*" "*" " " "*" "*" " " " " " " " " "*" " " " " " "
## 8  ( 3 ) "*" "*" "*" "*" "*" " " "*" " " " " " " "*" " " "*" " " " " " "
## 8  ( 4 ) "*" "*" "*" "*" "*" " " "*" " " "*" " " " " " " "*" " " " " " "
## 8  ( 5 ) "*" "*" "*" "*" "*" " " " " "*" " " " " " " " " "*" " " " " "*"
```

```
## 8  ( 6 ) "*" "*" "*" "*" "*" " " "*" " " " " " " " " " " "*" " " "*" " "
## 8  ( 7 ) "*" "*" "*" "*" "*" " " "*" " " " " " " " " " " "*" " " " " " "
##          X17 X18 X19 X20
## 1  ( 1 ) " " " " " " " "
## 1  ( 2 ) " " " " " " " "
## 1  ( 3 ) " " " " " " " "
## 1  ( 4 ) " " " " " " " "
## 1  ( 5 ) " " " " " " " "
## 1  ( 6 ) " " " " " " " "
## 1  ( 7 ) "*" " " " " " "
## 2  ( 1 ) " " " " " " " "
## 2  ( 2 ) " " " " " " " "
## 2  ( 3 ) " " " " " " " "
## 2  ( 4 ) " " " " " " " "
## 2  ( 5 ) " " " " " " " "
## 2  ( 6 ) " " " " " " " "
## 2  ( 7 ) " " " " " " " "
## 3  ( 1 ) " " " " " " " "
## 3  ( 2 ) " " " " " " " "
## 3  ( 3 ) " " " " " " " "
## 3  ( 4 ) " " " " " " " "
## 3  ( 5 ) " " " " " " " "
## 3  ( 6 ) " " " " " " " "
## 3  ( 7 ) " " " " " " " "
## 4  ( 1 ) " " " " " " " "
## 4  ( 2 ) " " " " " " " "
## 4  ( 3 ) " " " " " " " "
## 4  ( 4 ) " " " " " " " "
## 4  ( 5 ) " " " " " " " "
## 4  ( 6 ) " " " " " " " "
## 4  ( 7 ) " " " " " " " "
## 5  ( 1 ) " " " " " " " "
## 5  ( 2 ) " " "*" " " " "
## 5  ( 3 ) " " " " " " " "
## 5  ( 4 ) " " " " " " " "
## 5  ( 5 ) " " " " " " " "
## 5  ( 6 ) " " " " " " " "
## 5  ( 7 ) "*" " " " " " "
## 6  ( 1 ) " " " " " " " "
## 6  ( 2 ) " " " " " " " "
## 6  ( 3 ) " " " " " " " "
## 6  ( 4 ) " " " " " " " "
## 6  ( 5 ) " " " " " " " "
## 6  ( 6 ) "*" " " " " " "
## 6  ( 7 ) " " " " " " " "
## 7  ( 1 ) " " " " " " " "
## 7  ( 2 ) " " " " " " " "
## 7  ( 3 ) " " " " " " " "
## 7  ( 4 ) " " " " " " " "
## 7  ( 5 ) " " " " " " " "
## 7  ( 6 ) "*" " " " " " "
## 7  ( 7 ) " " " " " " " "
## 8  ( 1 ) " " " " " " " "
## 8  ( 2 ) " " " " " " " "
```

```
## 8  ( 3 ) " " " " " " " "
## 8  ( 4 ) " " " " " " " "
## 8  ( 5 ) " " " " " " " "
## 8  ( 6 ) " " " " " " " "
## 8  ( 7 ) "*" " " " " " "
```

```r
plot(leaps, scale="r2")
```



We can also distinguish the Cp for the variaous models with the Cpplot function.

```r
#show all models
library(faraway)
Cpplot(outs)
```

Forward and Backward selection of variables

Forward selection of variables

We fit the starting model, i.e., the minimum model

```r
min.model = lm(y ~ 1, data = mydata)
```

Then we specify the largest model with

```r
biggest = formula(lm(y ~ ., mydata))
biggest
```

```
## y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 +
##     X12 + X13 + X14 + X15 + X16 + X17 + X18 + X19 + X20
```

The step() function does the stepwise fitting for us. We provide the starting model, the direction (forward, backward, or both), and the largest model to consider

```r
fwd.model = step(min.model, direction = 'forward', scope = biggest)
```

```
## Start:  AIC=3009
## y ~ 1
##
##        Df Sum of Sq    RSS  AIC
## + X5    1     49523 155065 2872
## + X1    1     38700 165887 2906
## + X2    1     35553 169034 2916
## + X3    1     24586 180001 2947
## + X4    1     21529 183059 2955
## + X7    1     11784 192803 2981
## + X17   1      4778 199810 2999
## + X14   1      4514 200074 3000
## + X20   1      3093 201495 3003
## + X13   1      2976 201611 3004
## + X11   1      1507 203081 3007
## + X9    1      1151 203437 3008
## + X10   1      1056 203532 3008
## + X16   1       918 203669 3009
## <none>              204588 3009
## + X8    1       700 203887 3009
## + X19   1       574 204014 3010
## + X12   1       547 204041 3010
## + X15   1       180 204408 3011
## + X18   1       131 204456 3011
## + X6    1        25 204563 3011
##
## Step:  AIC=2872
## y ~ X5
##
##        Df Sum of Sq    RSS  AIC
## + X3    1     42065 113000 2716
## + X4    1     28164 126901 2774
## + X2    1     28145 126920 2774
## + X1    1     24784 130281 2787
## + X7    1     11735 143330 2835
## + X16   1      5276 149789 2857
## + X20   1      3741 151324 2862
## + X17   1      2668 152397 2866
## + X11   1      2262 152803 2867
## + X9    1      1722 153343 2869
## + X19   1      1565 153500 2869
## + X10   1      1524 153541 2870
## + X8    1      1154 153911 2871
## + X18   1       719 154346 2872
## <none>              155065 2872
## + X13   1       549 154516 2873
## + X12   1       309 154756 2873
## + X14   1       102 154963 2874
## + X6    1         9 155056 2874
## + X15   1         8 155057 2874
##
```

```
## Step:  AIC=2716
## y ~ X5 + X3
##
##        Df Sum of Sq    RSS  AIC
## + X4    1     40694  72306 2495
## + X2    1     31815  81185 2553
## + X1    1     25084  87916 2593
## + X7    1      8405 104595 2680
## + X11   1      5273 107727 2694
## + X9    1      3290 109710 2703
## + X18   1      3224 109776 2704
## + X20   1      2570 110430 2707
## + X16   1      2298 110702 2708
## + X12   1      1076 111924 2713
## + X8    1       883 112117 2714
## + X10   1       485 112515 2716
## <none>               113000 2716
## + X13   1       301 112699 2717
## + X19   1       214 112786 2717
## + X14   1       190 112810 2717
## + X6    1        89 112911 2718
## + X15   1        71 112929 2718
## + X17   1        33 112967 2718
##
## Step:  AIC=2495
## y ~ X5 + X3 + X4
##
##        Df Sum of Sq   RSS  AIC
## + X2    1     29922 42384 2230
## + X1    1     24316 47990 2292
## + X11   1      3581 68725 2472
## + X7    1      3207 69099 2474
## + X13   1      2705 69602 2478
## + X12   1      1831 70475 2484
## + X16   1      1193 71113 2489
## + X8    1      1174 71132 2489
## + X20   1       768 71538 2492
## + X17   1       414 71892 2494
## + X14   1       368 71938 2494
## + X9    1       310 71996 2495
## <none>               72306 2495
## + X15   1       129 72177 2496
## + X10   1        64 72242 2497
## + X19   1        58 72248 2497
## + X18   1        35 72271 2497
## + X6    1        27 72279 2497
##
## Step:  AIC=2230
## y ~ X5 + X3 + X4 + X2
##
##        Df Sum of Sq   RSS  AIC
## + X1    1     41934   449  -41
## + X12   1      1820 40564 2210
## + X11   1      1778 40606 2211
```

```
## + X8    1      1691 40693 2212
## + X7    1      1618 40766 2212
## + X20   1      1520 40864 2214
## + X16   1       626 41758 2225
## + X10   1       358 42026 2228
## + X19   1       339 42045 2228
## + X13   1       333 42051 2228
## + X9    1       319 42064 2228
## + X14   1       188 42196 2230
## <none>              42384 2230
## + X15   1        23 42361 2232
## + X18   1        20 42363 2232
## + X17   1        15 42369 2232
## + X6    1        11 42373 2232
##
## Step:  AIC=-41.34
## y ~ X5 + X3 + X4 + X2 + X1
##
##        Df Sum of Sq RSS   AIC
## + X13   1     2.037 447 -41.6
## <none>              449 -41.3
## + X7    1     1.619 448 -41.1
## + X8    1     1.231 448 -40.7
## + X16   1     0.964 448 -40.4
## + X15   1     0.750 449 -40.2
## + X17   1     0.680 449 -40.1
## + X9    1     0.458 449 -39.9
## + X14   1     0.423 449 -39.8
## + X10   1     0.379 449 -39.8
## + X6    1     0.275 449 -39.7
## + X20   1     0.275 449 -39.7
## + X19   1     0.264 449 -39.6
## + X18   1     0.248 449 -39.6
## + X11   1     0.223 449 -39.6
## + X12   1     0.104 449 -39.5
##
## Step:  AIC=-41.62
## y ~ X5 + X3 + X4 + X2 + X1 + X13
##
##        Df Sum of Sq RSS   AIC
## <none>              447 -41.6
## + X7    1     1.465 446 -41.3
## + X8    1     0.983 446 -40.7
## + X16   1     0.859 447 -40.6
## + X10   1     0.769 447 -40.5
## + X9    1     0.516 447 -40.2
## + X17   1     0.506 447 -40.2
## + X15   1     0.469 447 -40.1
## + X6    1     0.433 447 -40.1
## + X11   1     0.419 447 -40.1
## + X14   1     0.319 447 -40.0
## + X20   1     0.206 447 -39.8
## + X19   1     0.197 447 -39.8
## + X12   1     0.092 447 -39.7
```

```
## + X18   1     0.067 447 -39.7
```

Notice that information about each step is provided. If we do not want to see this, we set trace to 0.

The object contains the final model as well as the actions taken, which we access via the object's anova element.

```r
fwd.model$anova
```

```
##    Step Df  Deviance Resid. Df Resid. Dev     AIC
## 1        NA        NA       499   204587.9 3009.07
## 2  + X5 -1 49522.717       498   155065.2 2872.50
## 3  + X3 -1 42065.391       497   112999.8 2716.27
## 4  + X4 -1 40693.600       496    72306.2 2495.03
## 5  + X2 -1 29922.342       495    42383.8 2229.96
## 6  + X1 -1 41934.425       494      449.4  -41.34
## 7 + X13 -1     2.037       493      447.4  -41.62
```

Backward selection of variables

Backwad regression is similarly carried out. This time we fit the largest model as a starting point and so we need not supply the scope. Although we could supply scope as the smallest model to consider.

```r
biggest.model = lm(y ~ ., data = mydata)
bwd.model = step(biggest.model, direction="backward", trace = 0)
```

```r
bwd.model$anova
```

```
##     Step Df Deviance Resid. Df Resid. Dev    AIC
## 1         NA       NA       479      441.5 -20.20
## 2  - X14  1  0.02449       480      441.5 -22.18
## 3  - X19  1  0.04119       481      441.6 -24.13
## 4   - X9  1  0.06595       482      441.6 -26.06
## 5  - X18  1  0.09596       483      441.7 -27.95
## 6  - X17  1  0.09974       484      441.8 -29.83
## 7  - X20  1  0.15596       485      442.0 -31.66
## 8   - X6  1  0.34112       486      442.3 -33.27
## 9  - X12  1  0.31836       487      442.7 -34.91
## 10 - X15  1  0.31620       488      443.0 -36.55
## 11 - X10  1  0.78663       489      443.8 -37.67
## 12 - X11  1  0.55187       490      444.3 -39.05
## 13  - X8  1  0.82722       491      445.1 -40.12
## 14 - X16  1  0.76663       492      445.9 -41.26
## 15  - X7  1  1.46524       493      447.4 -41.62
```

Both directions

When we go in both directions (adding and dropping variables one at a time) we can begin with a viable model and then provide the upper and lower limits of the model via the scope argument. Here we provide only one model to scope so it is taken as the upper limit.

We start with an arbitrary model.

```r
mid.model = lm(y ~ X1 + X5 + X12 + X20 , data = mydata)
biggest = formula(lm(y ~ ., mydata))
both.model = stepAIC(mid.model, direction="both", scope = biggest,
                     trace = 0)
```

```r
both.model$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## y ~ X1 + X5 + X12 + X20
##
## Final Model:
## y ~ X1 + X5 + X2 + X3 + X4 + X13
##
##
##    Step Df  Deviance Resid. Df Resid. Dev     AIC
## 1                          495   129275.0 2787.54
## 2  + X2  1 4.582e+04       494    83456.4 2570.74
## 3  + X3  1 4.632e+04       493    37136.6 2167.87
## 4  + X4  1 3.669e+04       492      448.9  -37.86
## 5 - X12  1 1.895e-01       493      449.1  -39.65
## 6 - X20  1 2.749e-01       494      449.4  -41.34
## 7 + X13  1 2.037e+00       493      447.4  -41.62
```

Note that in this case the final model is the same in all three approaches. That need not be the case generally.