Model (Variable) Selection

Aliase matrix

```
> n=16
> x1=c(1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1)
> x2=c(1,1,1,1,-1,-1,-1,-1,1,1,1,1,-1,-1,-1,-1)
> x3=c(1,1,-1,-1,1,1,-1,-1,1,1,-1,-1,1,1,-1,-1)
> y=20+x1+x2+.2*x3+2*I(x1*x2)+I(x1*x3)+0.5*I(x2*x3)+rnorm(n, 0, 5)
> f=lm(y~x1+x2+x3+I(x1*x2)+I(x1*x3)+I(x2*x3))
> X=model.matrix(f);X
```

```
   (Intercept) x1 x2 x3 I(x1 * x2) I(x1 * x3) I(x2 * x3)
1            1  1  1  1          1          1          1
2            1  1  1  1          1          1          1
3            1  1  1 -1          1         -1         -1
4            1  1  1 -1          1         -1         -1
5            1  1 -1  1         -1          1         -1
6            1  1 -1  1         -1          1         -1
7            1  1 -1 -1         -1         -1          1
8            1  1 -1 -1         -1         -1          1
9            1 -1  1  1         -1         -1          1
10           1 -1  1  1         -1         -1          1
11           1 -1  1 -1         -1          1         -1
12           1 -1  1 -1         -1          1         -1
13           1 -1 -1  1          1         -1         -1
14           1 -1 -1  1          1         -1         -1
15           1 -1 -1 -1          1          1          1
16           1 -1 -1 -1          1          1          1
attr(,"assign")
[1] 0 1 2 3 4 5 6
```

```
> t(X) %*% X
```

```
            (Intercept) x1 x2 x3 I(x1 * x2) I(x1 * x3) I(x2 * x3)
(Intercept)          16  0  0  0          0          0          0
x1                    0 16  0  0          0          0          0
x2                    0  0 16  0          0          0          0
x3                    0  0  0 16          0          0          0
I(x1 * x2)            0  0  0  0         16          0          0
I(x1 * x3)            0  0  0  0          0         16          0
I(x2 * x3)            0  0  0  0          0          0         16
```

```
> f1=lm(y~x1+x2+x3)
> Xp=model.matrix(f1)
> Xr=X[,-c(1:4)]
> solve(t(Xp) %*% Xp)
```

```
            (Intercept)     x1     x2     x3
(Intercept)      0.0625 0.0000 0.0000 0.0000
x1               0.0000 0.0625 0.0000 0.0000
x2               0.0000 0.0000 0.0625 0.0000
x3               0.0000 0.0000 0.0000 0.0625
```

```
> solve(t(Xp) %*% Xp) %*% t(Xp) %*% Xr
            I(x1 * x2) I(x1 * x3) I(x2 * x3)
(Intercept)          0          0          0
x1                   0          0          0
x2                   0          0          0
x3                   0          0          0
```

```
> n=12
> x1=c(1,1,1,1,1,1,-1,-1,-1,-1,-1,-1)
> x2=c(1,1,1,-1,-1,-1,1,1,1,-1,-1,-1)
> x3=c(1,-1,-1,1,1,-1,-1,1,1,-1,-1,1)
> set.seed(1)
> y=20+x1+3*x2+2*x3+.2*x1*x2+0.03*x1*x3+0.8*x2*x3+rnorm(n, 0, 5)
> cbind(y, x1, x2,x3)
         y x1 x2 x3
 [1,] 23.90  1  1  1
 [2,] 22.29  1  1 -1
 [3,] 17.19  1  1 -1
 [4,] 27.01  1 -1  1
 [5,] 20.68  1 -1  1
 [6,] 12.47  1 -1 -1
 [7,] 21.47 -1  1 -1
 [8,] 28.26 -1  1  1
 [9,] 27.45 -1  1  1
[10,] 13.50 -1 -1 -1
[11,] 22.59 -1 -1 -1
[12,] 19.32 -1 -1  1
```

```
> f=lm(y~x1+x2+x3+I(x1*x2)+I(x1*x3)+I(x2*x3))
> X=model.matrix(f)
> t(X) %*% X
            (Intercept) x1 x2 x3 I(x1 * x2) I(x1 * x3) I(x2 * x3)
(Intercept)          12  0  0  0          0          0          0
x1                    0 12  0  0          0          0         -4
x2                    0  0 12  0          0         -4          0
x3                    0  0  0 12         -4          0          0
I(x1 * x2)            0  0  0 -4         12          0          0
I(x1 * x3)            0  0 -4  0          0         12          0
I(x2 * x3)            0 -4  0  0          0          0         12
```

```
> f1=lm(y~x1+x2+x3)
> Xp=model.matrix(f1)
> Xr=X[,-c(1:4)]
> solve(t(Xp) %*% Xp)
            (Intercept)      x1      x2      x3
(Intercept)     0.08333 0.00000 0.00000 0.00000
x1              0.00000 0.08333 0.00000 0.00000
x2              0.00000 0.00000 0.08333 0.00000
x3              0.00000 0.00000 0.00000 0.08333
```

```
> solve(t(Xp) %*% Xp) %*% t(Xp) %*% Xr

            I(x1 * x2) I(x1 * x3) I(x2 * x3)
(Intercept)     0.0000     0.0000     0.0000
x1              0.0000     0.0000    -0.3333
x2              0.0000    -0.3333     0.0000
x3             -0.3333     0.0000     0.0000
```

full model vs subset model

```
> summary(f)

Call:
lm(formula = y ~ x1 + x2 + x3 + I(x1 * x2) + I(x1 * x3) + I(x2 *
    x3))

Residuals:
     1      2      3      4      5      6      7      8      9     10
-2.055  3.576 -1.521  4.192 -2.137 -2.055 -2.055  1.434  0.621 -3.515
    11     12
 5.571 -2.055

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   21.343      1.291   16.54  1.5e-05
x1            -0.843      1.369   -0.62    0.565
x2             2.411      1.369    1.76    0.139
x3             2.899      1.369    2.12    0.088
I(x1 * x2)    -0.579      1.369   -0.42    0.690
I(x1 * x3)     0.984      1.369    0.72    0.505
I(x2 * x3)    -0.263      1.369   -0.19    0.855

Residual standard error: 4.47 on 5 degrees of freedom
Multiple R-squared: 0.653,        Adjusted R-squared: 0.237
F-statistic: 1.57 on 6 and 5 DF,  p-value: 0.319

> summary(f1)

Call:
lm(formula = y ~ x1 + x2 + x3)

Residuals:
   Min     1Q Median     3Q    Max
-3.788 -2.527 -0.372  1.419  5.666

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   21.343      1.093   19.53  4.9e-08
x1            -0.755      1.093   -0.69    0.509
x2             2.083      1.093    1.91    0.093
```

```
x3              3.092      1.093    2.83    0.022

Residual standard error: 3.78 on 8 degrees of freedom
Multiple R-squared: 0.602,       Adjusted R-squared: 0.453
F-statistic: 4.04 on 3 and 8 DF,  p-value: 0.0507

> x.new=data.frame(x1=-1,x2=1,x3=-1)
> predict(f, x.new, interval="predict")

    fit  lwr   upr
1 23.52 7.96 39.09

> predict(f1, x.new, interval="predict")

    fit   lwr   upr
1 21.09 11.01 31.17
```

Criterion-based (all possible subsets regression)

Consider a case with explanatory variables: $X_1, X_2, \cdots, X_K$

Full model: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ where $\mathbf{y}$ is an $n \times 1$ vector of responses and $\mathbf{X}$ is an $n \times (K+1)$ matrix containing a column of 1s as well as a column correspond to each of the K regressors and Rank($\mathbf{X}$)=$K+1$, $\boldsymbol{\beta}$ is a $(K+1) \times 1$ vector of unknown parameters and $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$. Full model: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$

Subset model $\mathcal{M}$: $\mathbf{y} = \mathbf{X}_p\boldsymbol{\beta}_p + \boldsymbol{\varepsilon}$, where $M \subset \{1, 2, \cdots, K\}$and $\mathbf{X}_p$ is a submatrix of $\mathbf{X}$ containing a column of 1s as well as the columns of $\mathbf{X}$ indexed by p.

- R^2: not a good criterion. Always increase with model size => "optimum" is to take the biggest mode
- Adjusted R^2: better. It "penalized" bigger models. Follows principle of parsimony / Occam's razor.

$$\text{Adjusted } R^2 = 1 - \frac{(\frac{\text{SSres}_{\mathcal{M}}}{n-p})}{(\frac{\text{SSTO}}{n-1})} = 1 - (1 - R^2)\frac{(n-1)}{n-p}$$

  where p is the number of parameters in the model $\mathcal{M}$.

- Mallow's C_p: (Mallows 1964) attempts to estimate a model's predictive power, i.e. the power to predict a new observation.

$$C_p = \frac{\text{SSres}_{\mathcal{M}}}{\hat{\sigma}^2} - n + 2p$$

  where $\hat{\sigma}^2 = \text{SSres(FM)}/df_{FM}$ is from the "full" model and is the "best" estimate of σ^2 we have.

  $\text{SSres}_{\mathcal{M}}$ is the residual sum of squares of the model $\mathcal{M}$.

  p is the number of parameters in the model $\mathcal{M}$.

- This is an estimate of the mean-squared error of predict for model $\mathcal{M}$ which takes **bias** and **variance** into account.
- C_p only depends on Residual Sum of squares, $\hat{\sigma}^2$, p and n and therefore can be easily computed using usual regression.
- C_p measures the difference in fitting errors between the full and subset models.
- For the full model, $C_p = p$
- Two subsets can be compared by comparing their values of C_p. Good models have $C_p \cong p$. Any model with $C_p \leq p$ will be a candidate for a good subset model.

- AIC: Akaike information criterion (Akaike, 1973)

$$\text{AIC} = -2 \max \log(\text{L}) + 2 \cdot p$$

  - Select the model that minimizes AIC

- BIC: Bayesian information criterion (Schwarz, 1978)

$$\text{BIC} = -2 \max \log(\text{L}) + [\log n] \cdot p$$

  - Select the model that minimizes BIC

- PRESS (Predicted REsidual Sum of Squares): Leave-one-out cross validation (CV)

  leave out each case in turn, and calculated the estimate $\hat{\boldsymbol{\beta}}_{(i)}$ of $\boldsymbol{\beta}$ from the remaining $(n-1)$ cases. The error in predicting the i th case is

$$\hat{e}_{(i)} = y_i - \hat{y}_{i(i)} = y_i - \mathbf{x}'\hat{\beta}_{(i)} = \frac{\hat{e}_i}{1 - h_{ii}}$$

  where h_{ii} is the leverage point for the ith case.

  The CV statistics

$$\text{PRESS} = \sum_{i=1}^{n} \hat{e}_{(i)}^2$$

  - Cross-validation type method
  - Good models have smaller value of PRESS.
  - Computation requires much more work than C_p.

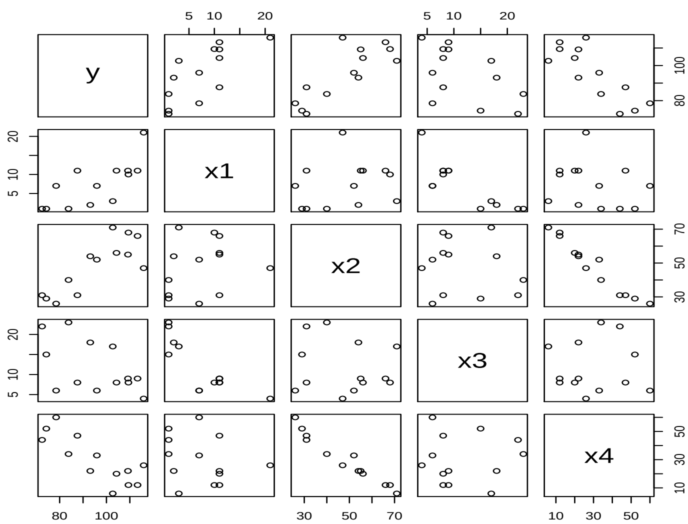
Automatic search procedure

Given a criterion, we now have to decide how we are going to search through all possible models.

"Best subset": search all possible models and take the one with highest R^2_{adj} or lowest $C_p$in R, use function "leaps" or "regsubsets") in the **leaps** package.

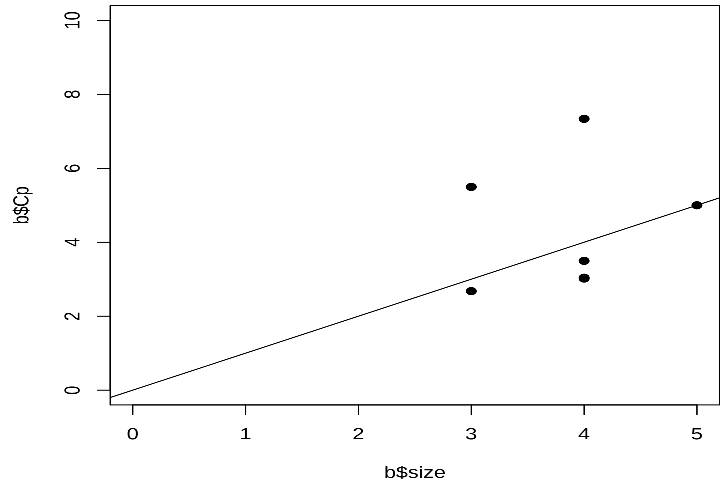
install.packages(”leaps”)
library(”leaps”)

```
> rm(list=ls())
> hald=read.table("ex-9-1.txt", header=T)
> cor(hald)

          y      x1      x2       x3       x4
y   1.0000  0.7307  0.8163 -0.53467 -0.82131
x1  0.7307  1.0000  0.2286 -0.82413 -0.24545
x2  0.8163  0.2286  1.0000 -0.13924 -0.97295
x3 -0.5347 -0.8241 -0.1392  1.00000  0.02954
x4 -0.8213 -0.2454 -0.9730  0.02954  1.00000

> pairs(hald)
```




```
> y=hald[,1]
> x1=hald[,2]
> x2=hald[,3]
> x3=hald[,4]
> x4=hald[,5]
> f2=lm(y~., data=hald)
> xset=cbind(x1,x2,x3,x4)
> summary(f2)

Call:
lm(formula = y ~ ., data = hald)

Residuals:
   Min     1Q Median     3Q    Max
-3.175 -1.671  0.251  1.378  3.925

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   62.405     70.071    0.89    0.399
```

```
x1              1.551       0.745    2.08    0.071
x2              0.510       0.724    0.70    0.501
x3              0.102       0.755    0.14    0.896
x4             -0.144       0.709   -0.20    0.844

Residual standard error: 2.45 on 8 degrees of freedom
Multiple R-squared: 0.982,        Adjusted R-squared: 0.974
F-statistic:  111 on 4 and 8 DF,  p-value: 4.76e-07

> library(leaps)
> b=leaps(xset, y)  #Cp, adjr2, r2
> b

$which
      1     2     3     4
1 FALSE FALSE FALSE  TRUE
1 FALSE  TRUE FALSE FALSE
1  TRUE FALSE FALSE FALSE
1 FALSE FALSE  TRUE FALSE
2  TRUE  TRUE FALSE FALSE
2  TRUE FALSE FALSE  TRUE
2 FALSE FALSE  TRUE  TRUE
2 FALSE  TRUE  TRUE FALSE
2 FALSE  TRUE FALSE  TRUE
2  TRUE FALSE  TRUE FALSE
3  TRUE  TRUE FALSE  TRUE
3  TRUE  TRUE  TRUE FALSE
3  TRUE FALSE  TRUE  TRUE
3 FALSE  TRUE  TRUE  TRUE
4  TRUE  TRUE  TRUE  TRUE

$label
[1] "(Intercept)" "1"           "2"           "3"           "4"

$size
 [1] 2 2 2 2 3 3 3 3 3 3 4 4 4 4 5

$Cp
 [1] 138.731 142.486 202.549 315.154   2.678   5.496  22.373  62.438
 [9] 138.226 198.095   3.018   3.041   3.497   7.337   5.000

> plot(b$size, b$Cp, pch=16, ylim=c(0,10), xlim=c(0,5)); abline(0,1)
```



other criterion in leaps

```
> b.Cp=leaps(xset, y)$Cp
> b.adjr2=leaps(xset, y, method="adjr2")$adjr2
> cbind(b$which, b.Cp, b.adjr2)

  1 2 3 4    b.Cp b.adjr2
1 0 0 0 1 138.731  0.6450
1 0 1 0 0 142.486  0.6359
1 1 0 0 0 202.549  0.4916
1 0 0 1 0 315.154  0.2210
2 1 1 0 0   2.678  0.9744
2 1 0 0 1   5.496  0.9670
2 0 0 1 1  22.373  0.9223
2 0 1 1 0  62.438  0.8164
2 0 1 0 1 138.226  0.6161
2 1 0 1 0 198.095  0.4578
3 1 1 0 1   3.018  0.9764
3 1 1 1 0   3.041  0.9764
3 1 0 1 1   3.497  0.9750
3 0 1 1 1   7.337  0.9638
4 1 1 1 1   5.000  0.9736

> par(mfrow=c(1,2))
> plot(b$size, b$Cp, ylim=c(0,10),xlim=c(0,10));abline(0,1)
> b$which[which(b.Cp==min(b.Cp)),]

    1     2     3     4
 TRUE  TRUE FALSE FALSE

> b$which[which(b.adjr2==max(b.adjr2)),]

    1     2     3     4
 TRUE  TRUE FALSE  TRUE
```

```
> f=lm(y~(x1+x2+x3+x4)^2, data=hald)
> xset=model.matrix(f)[,-1]
> leaps(xset, y, nbest=2)

$which
      1     2     3     4     5     6     7     8     9     A
1  FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
1  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
2   TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
2   TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
3  FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE
3  FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
4  FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE
4  FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
5  FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE
5  FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
6  FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
6  FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
7  FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
7  FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
8  FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
8   TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
9   TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
9   TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
10  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE

$label
 [1] "(Intercept)" "1"           "2"           "3"           "4"
 [6] "5"           "6"           "7"           "8"           "9"
[11] "A"

$size
 [1]  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9 10 10 11

$Cp
 [1] 654.874 740.402  59.905  79.383  47.402  47.979  13.346  19.328
 [9]  12.307  12.350   6.383  12.478   5.610   6.523   7.403   7.574
[17]   9.000   9.273  11.000

> #nbest=2 only the best two models for a given parameters size is reported
> b=leaps(xset, y, nbest=2)
> b$which[which(b$Cp==min(b$Cp)),]

    1     2     3     4     5     6     7     8     9     A
FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
```

use function regsubsets

```
> bs=regsubsets(y~x1+x2+x3+x4, data=hald)
> coef(bs,1:4) #coef for each of the best models  with 1 to 4 variables
```

```
[[1]]
(Intercept)          x4
   117.5679     -0.7382

[[2]]
(Intercept)          x1           x2
    52.5773      1.4683       0.6623

[[3]]
(Intercept)          x1           x2           x4
    71.6483      1.4519       0.4161      -0.2365

[[4]]
(Intercept)          x1           x2           x3           x4
    62.4054      1.5511       0.5102       0.1019      -0.1441
```

```
> vcov(bs,2)

            (Intercept)        x1         x2
(Intercept)     5.22659 -0.048565 -0.091764
x1             -0.04857  0.014714 -0.001271
x2             -0.09176 -0.001271  0.002103
```

```
> a=summary(bs)
> names(a) #see objects in summary(bs)

[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
> cbind(a$which, a$rss, a$rsq, a$adjr2, a$cp)

  (Intercept) x1 x2 x3 x4
1           1  0  0  0  1 883.87 0.6745 0.6450 138.731
2           1  1  1  0  0  57.90 0.9787 0.9744   2.678
3           1  1  1  0  1  47.97 0.9823 0.9764   3.018
4           1  1  1  1  1  47.86 0.9824 0.9736   5.000
```

Other Serach Strategies (Stepwise)

In R, use the function "step". Works for any model with AIC. In multiple linear regression, AIC is a linear function of C_p.

1. Forward selection,
2. Backward elimination,
3. Stepwise regression

   $F_{in}, F_{out}, t_{in}, t_{out}, p_{in}, p_{out}$

   Under the prespecified F_{in} and F_{out}, enter (remove) X_k with the largest (smallest) $F^\star$, $t_\star$, partial correlation in absolute value with Y, adjusting for the X's in the equation already.

- Advantages: easy to explain, inexpensive to compute
- Disadvantages: could end up with a model of no substantive interest

forward selection

```
> f=lm(y~1, data=hald)
> add1(f, ~.+x1+x2+x3+x4)  #add one term #~. for the orginal model

Single term additions

Model:
y ~ 1
       Df Sum of Sq  RSS  AIC
<none>              2716   71
x1      1      1450 1266   64
x2      1      1809  906   59
x3      1       776 1939   69
x4      1      1832  884   59

> add1(f, ~x1+x2+x3+x4, test="F")  #test by F-test

Single term additions

Model:
y ~ 1
       Df Sum of Sq  RSS  AIC F value   Pr(F)
<none>              2716   71
x1      1      1450 1266   64    12.6 0.00455
x2      1      1809  906   59    22.0 0.00066
x3      1       776 1939   69     4.4 0.05976
x4      1      1832  884   59    22.8 0.00058

> f2=update(f, ~.+ x4)
> anova(f2)

Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value  Pr(>F)
x4         1   1832    1832    22.8 0.00058
Residuals 11    884      80

> anova(f,f2)

Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x4
  Res.Df  RSS Df Sum of Sq    F  Pr(>F)
1     12 2716
2     11  884  1      1832 22.8 0.00058
```

```
> add1(f2, ~.+x1+x2+x3, test="F")

Single term additions

Model:
y ~ x4
       Df Sum of Sq RSS AIC F value   Pr(F)
<none>              884  59
x1      1       809  75  29  108.22 1.1e-06
x2      1        15 869  61    0.17    0.69
x3      1       708 176  40   40.29 8.4e-05

> f3=update(f2, ~.+ x1)
> anova(f3)

Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value  Pr(>F)
x4         1   1832    1832     245 2.3e-08
x1         1    809     809     108 1.1e-06
Residuals 10     75       7

> anova(f2, f3)

Analysis of Variance Table

Model 1: y ~ x4
Model 2: y ~ x4 + x1
  Res.Df RSS Df Sum of Sq   F  Pr(>F)
1     11 884
2     10  75  1       809 108 1.1e-06

> add1(f3, ~.+x2+x3, test="F")

Single term additions

Model:
y ~ x4 + x1
       Df Sum of Sq  RSS  AIC F value Pr(F)
<none>              74.8 28.7
x2      1      26.8 48.0 25.0    5.03 0.052
x3      1      23.9 50.8 25.7    4.24 0.070

> f4=update(f3, ~.+ x2) ;f4

Call:
lm(formula = y ~ x4 + x1 + x2, data = hald)

Coefficients:
(Intercept)          x4          x1          x2
     71.648      -0.237       1.452       0.416
```

```
> anova(f4)

Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value  Pr(>F)
x4         1   1832    1832  343.68 1.8e-08
x1         1    809     809  151.79 6.1e-07
x2         1     27      27    5.03   0.052
Residuals  9     48       5

> anova(f3,f4)

Analysis of Variance Table

Model 1: y ~ x4 + x1
Model 2: y ~ x4 + x1 + x2
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     10 74.8
2      9 48.0  1      26.8 5.03  0.052

> add1(f4, ~.+x3, test="F")

Single term additions

Model:
y ~ x4 + x1 + x2
       Df Sum of Sq  RSS  AIC F value Pr(F)
<none>              48.0 25.0
x3      1       0.1 47.9 26.9    0.02   0.9

> bm.fd=f4
```

final model with $x1 + x2 + x4$

backward selection

```
> f=lm(y~x1+x2+x3+x4, data=hald) #f=lm(y~., data=hald)
> summary(f)

Call:
lm(formula = y ~ x1 + x2 + x3 + x4, data = hald)

Residuals:
   Min     1Q Median     3Q    Max
-3.175 -1.671  0.251  1.378  3.925

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   62.405     70.071    0.89    0.399
x1             1.551      0.745    2.08    0.071
```

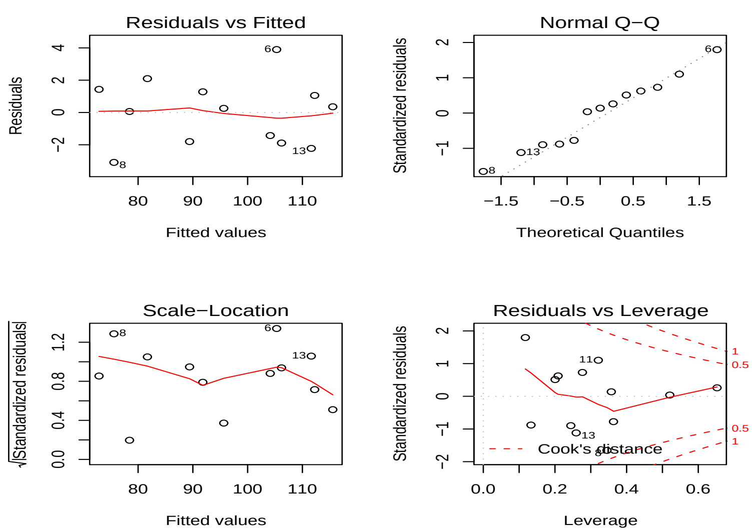
```
x2              0.510       0.724    0.70    0.501
x3              0.102       0.755    0.14    0.896
x4             -0.144       0.709   -0.20    0.844

Residual standard error: 2.45 on 8 degrees of freedom
Multiple R-squared: 0.982,        Adjusted R-squared: 0.974
F-statistic:  111 on 4 and 8 DF,  p-value: 4.76e-07

> drop1(f, test="F")

Single term deletions

Model:
y ~ x1 + x2 + x3 + x4
       Df Sum of Sq  RSS  AIC F value Pr(F)
<none>              47.9 26.9
x1      1      26.0 73.8 30.6    4.34 0.071
x2      1       3.0 50.8 25.7    0.50 0.501
x3      1       0.1 48.0 25.0    0.02 0.896
x4      1       0.2 48.1 25.0    0.04 0.844

> f2=update(f, ~.- x3)  # delete x3
> anova(f2)

Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value  Pr(>F)
x1         1   1450    1450  272.04 4.9e-08
x2         1   1208    1208  226.59 1.1e-07
x4         1     10      10    1.86    0.21
Residuals  9     48       5

> anova(f2,f)

Analysis of Variance Table

Model 1: y ~ x1 + x2 + x4
Model 2: y ~ x1 + x2 + x3 + x4
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      9 48.0
2      8 47.9  1       0.1 0.02    0.9

> drop1(f2,  test="F")

Single term deletions

Model:
y ~ x1 + x2 + x4
       Df Sum of Sq RSS AIC F value   Pr(F)
```

```
<none>               48  25
x1      1       821 869  61  154.01 5.8e-07
x2      1        27  75  29    5.03   0.052
x4      1        10  58  25    1.86   0.205

> f3=update(f2, ~.- x4) ;f3

Call:
lm(formula = y ~ x1 + x2, data = hald)

Coefficients:
(Intercept)           x1           x2
     52.577        1.468        0.662

> anova(f3)

Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value  Pr(>F)
x1         1   1450    1450     250 2.1e-08
x2         1   1208    1208     209 5.0e-08
Residuals 10     58       6

> anova(f3,f2)

Analysis of Variance Table

Model 1: y ~ x1 + x2
Model 2: y ~ x1 + x2 + x4
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     10 57.9
2      9 48.0  1       9.9 1.86   0.21

> drop1(f3, test="F")

Single term deletions

Model:
y ~ x1 + x2
       Df Sum of Sq  RSS  AIC F value   Pr(F)
<none>               58   25
x1      1       848  906   59     147 2.7e-07
x2      1      1208 1266   64     209 5.0e-08

> bm.bd=f3
```

final model with $x1 + x2$

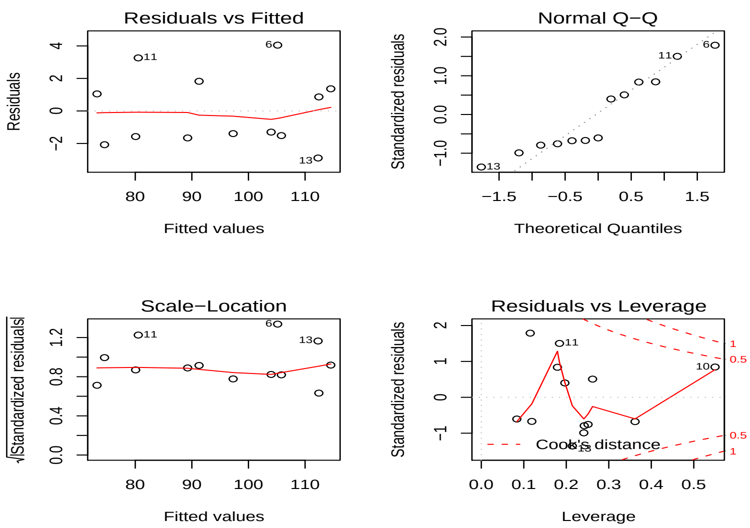
write my own function call fun.ss() for the useful information of a fitted model

```
> library(car)
> fun.ss=function(f) {
+    par(mfrow=c(2,2))
+    plot(f)
+    y.hat=fitted(f)
+    ei=residuals(f)  #residual: y-y.hat
+    hii=hatvalues(f)
+    deleted.ei=ei/(1-hii) #deleted residual (press residual): y-y.hat(i)
+    ri=rstandard(f)  #ei/(S*sqrt(1-hii))
+    ti=rstudent(f)   #delete.ei/(S(i)*sqrt(1-hii))
+    press=sum(deleted.ei^2)
+    a=summary(f)
+    p=a$df[1]
+    DFres=a$df[2]
+    SSres=sum(ei^2)
+    SSres/DFres
+    MSres=a$sigma^2
+    adjr2=a$adj.r.squared
+    xvif=vif(f)
+    ss=c("SSres"=SSres,"adj.R2"=adjr2,"MSres"=MSres,"PRESS"=press,xvif)
+    print(attr(terms(f),"term.labels"))
+    return(ss)
+ }

> fun.ss(bm.fd)

[1] "x4" "x1" "x2"
  SSres  adj.R2   MSres   PRESS      x4      x1      x2
47.9727  0.9764  5.3303 85.3511 18.9401  1.0663 18.7803
```




```
> fun.ss(bm.bd)
```

```
[1] "x1" "x2"
  SSres  adj.R2   MSres   PRESS      x1      x2
57.9045  0.9744  5.7904 93.8825  1.0551  1.0551
```



stepwise selection

1. start with a model containing just the intercept

```
> f=lm(y~1, data=hald)
> step(f, ~.+x1+x2+x3+x4, test="F")

Start:  AIC=71.44
y ~ 1

       Df Sum of Sq  RSS  AIC F value   Pr(F)
+ x4    1      1832  884   59    22.8 0.00058
+ x2    1      1809  906   59    22.0 0.00066
+ x1    1      1450 1266   64    12.6 0.00455
+ x3    1       776 1939   69     4.4 0.05976
<none>              2716   71

Step:  AIC=58.85
y ~ x4

       Df Sum of Sq  RSS  AIC F value   Pr(F)
+ x1    1       809   75   29  108.22 1.1e-06
+ x3    1       708  176   40   40.29 8.4e-05
<none>               884   59
+ x2    1        15  869   61    0.17 0.68668
- x4    1      1832 2716   71   22.80 0.00058

Step:  AIC=28.74
y ~ x4 + x1
```

```
       Df Sum of Sq  RSS  AIC F value   Pr(F)
+ x2    1        27   48   25    5.03   0.052
+ x3    1        24   51   26    4.24   0.070
<none>                75   29
- x1    1       809  884   59  108.22 1.1e-06
- x4    1      1191 1266   64  159.30 1.8e-07

Step:  AIC=24.97
y ~ x4 + x1 + x2

       Df Sum of Sq RSS AIC F value   Pr(F)
<none>               48  25
- x4    1        10  58  25    1.86   0.205
+ x3    1      0.11  48  27    0.02   0.896
- x2    1        27  75  29    5.03   0.052
- x1    1       821 869  61  154.01 5.8e-07

Call:
lm(formula = y ~ x4 + x1 + x2, data = hald)

Coefficients:
(Intercept)           x4           x1           x2
     71.648       -0.237        1.452        0.416
```

use stepwise function with forward selection procedure

```
> step(f, ~.+x1+x2+x3+x4, direction="forward", test="F")

Start:  AIC=71.44
y ~ 1

       Df Sum of Sq  RSS  AIC F value   Pr(F)
+ x4    1      1832  884   59    22.8 0.00058
+ x2    1      1809  906   59    22.0 0.00066
+ x1    1      1450 1266   64    12.6 0.00455
+ x3    1       776 1939   69     4.4 0.05976
<none>              2716   71

Step:  AIC=58.85
y ~ x4

       Df Sum of Sq RSS AIC F value   Pr(F)
+ x1    1       809  75  29  108.22 1.1e-06
+ x3    1       708 176  40   40.29 8.4e-05
<none>              884  59
+ x2    1        15 869  61    0.17    0.69

Step:  AIC=28.74
y ~ x4 + x1
```

```
       Df Sum of Sq  RSS  AIC F value Pr(F)
+ x2    1      26.8 48.0 25.0    5.03 0.052
+ x3    1      23.9 50.8 25.7    4.24 0.070
<none>              74.8 28.7

Step:  AIC=24.97
y ~ x4 + x1 + x2

       Df Sum of Sq  RSS  AIC F value Pr(F)
<none>              48.0 25.0
+ x3    1       0.1 47.9 26.9    0.02   0.9

Call:
lm(formula = y ~ x4 + x1 + x2, data = hald)

Coefficients:
(Intercept)           x4           x1           x2
     71.648       -0.237        1.452        0.416
```

2. start with a model containing $x1 + x2 + x3 + x4$

```
> f=lm(y~x1+x2+x3+x4, data=hald) #f=lm(y~., data=hald)
> summary(f)

Call:
lm(formula = y ~ x1 + x2 + x3 + x4, data = hald)

Residuals:
   Min     1Q Median     3Q    Max
-3.175 -1.671  0.251  1.378  3.925

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   62.405     70.071    0.89    0.399
x1             1.551      0.745    2.08    0.071
x2             0.510      0.724    0.70    0.501
x3             0.102      0.755    0.14    0.896
x4            -0.144      0.709   -0.20    0.844

Residual standard error: 2.45 on 8 degrees of freedom
Multiple R-squared: 0.982,       Adjusted R-squared: 0.974
F-statistic:  111 on 4 and 8 DF,  p-value: 4.76e-07

> step(f, test="F")

Start:  AIC=26.94
y ~ x1 + x2 + x3 + x4

       Df Sum of Sq  RSS  AIC F value Pr(F)
- x3    1       0.1 48.0 25.0    0.02 0.896
```

```
- x4    1       0.2 48.1 25.0    0.04 0.844
- x2    1       3.0 50.8 25.7    0.50 0.501
<none>              47.9 26.9
- x1    1      26.0 73.8 30.6    4.34 0.071

Step:  AIC=24.97
y ~ x1 + x2 + x4

       Df Sum of Sq RSS AIC F value   Pr(F)
<none>               48  25
- x4    1        10  58  25    1.86   0.205
- x2    1        27  75  29    5.03   0.052
- x1    1       821 869  61  154.01 5.8e-07

Call:
lm(formula = y ~ x1 + x2 + x4, data = hald)

Coefficients:
(Intercept)           x1           x2           x4
     71.648        1.452        0.416       -0.237
```

use stepwise function for backward elimination procedure

```
> step(f, direction="backward", test="F")

Start:  AIC=26.94
y ~ x1 + x2 + x3 + x4

       Df Sum of Sq  RSS  AIC F value Pr(F)
- x3    1       0.1 48.0 25.0    0.02 0.896
- x4    1       0.2 48.1 25.0    0.04 0.844
- x2    1       3.0 50.8 25.7    0.50 0.501
<none>              47.9 26.9
- x1    1      26.0 73.8 30.6    4.34 0.071

Step:  AIC=24.97
y ~ x1 + x2 + x4

       Df Sum of Sq RSS AIC F value   Pr(F)
<none>               48  25
- x4    1        10  58  25    1.86   0.205
- x2    1        27  75  29    5.03   0.052
- x1    1       821 869  61  154.01 5.8e-07

Call:
lm(formula = y ~ x1 + x2 + x4, data = hald)

Coefficients:
(Intercept)           x1           x2           x4
     71.648        1.452        0.416       -0.237
```

3. start with a model containing $x1+x2+x3+x4$ and consider other higher order terms. (we should not do this in this case since there are only 13 data in this data set.)

```
> f=lm(y~x1+x2+x3+x4, data=hald)
> step(f, ~I(x1^2)+I(x2^2)+ .^2, test="F")

Start:  AIC=26.94
y ~ x1 + x2 + x3 + x4

          Df Sum of Sq  RSS  AIC F value Pr(F)
- x3       1       0.1 48.0 25.0    0.02 0.896
- x4       1       0.2 48.1 25.0    0.04 0.844
+ I(x2^2)  1      12.2 35.7 25.1    2.40 0.165
- x2       1       3.0 50.8 25.7    0.50 0.501
+ x2:x4    1       7.1 40.8 26.9    1.22 0.306
<none>                 47.9 26.9
+ x1:x4    1       0.9 46.9 28.7    0.14 0.719
+ I(x1^2)  1       0.6 47.2 28.8    0.10 0.765
+ x1:x2    1       0.5 47.4 28.8    0.07 0.800
+ x2:x3    1       0.3 47.6 28.9    0.04 0.840
+ x3:x4    1       0.2 47.7 28.9    0.03 0.875
+ x1:x3    1       0.1 47.8 28.9    0.01 0.933
- x1       1      26.0 73.8 30.6    4.34 0.071

Step:  AIC=24.97
y ~ x1 + x2 + x4

          Df Sum of Sq RSS AIC F value   Pr(F)
+ I(x2^2)  1        11  37  24    2.27   0.170
<none>                  48  25
+ x2:x4    1         6  42  25    1.24   0.298
- x4       1        10  58  25    1.86   0.205
+ x1:x4    1         1  47  27    0.17   0.689
+ I(x1^2)  1         1  47  27    0.13   0.729
+ x1:x2    1         1  47  27    0.09   0.771
+ x3       1      0.11  48  27    0.02   0.896
- x2       1        27  75  29    5.03   0.052
- x1       1       821 869  61  154.01 5.8e-07

Step:  AIC=23.72
y ~ x1 + x2 + x4 + I(x2^2)

          Df Sum of Sq RSS AIC F value   Pr(F)
+ x2:x4    1         8  29  22    2.03    0.20
- x4       1         3  40  23    0.61    0.46
<none>                  37  24
- I(x2^2)  1        11  48  25    2.27    0.17
+ x3       1         2  36  25    0.33    0.58
+ x1:x4    1         1  37  25    0.13    0.73
```

```
+ I(x1^2)  1         1  37  26    0.11    0.75
+ x1:x2    1      0.29  37  26    0.06    0.82
- x2       1        25  62  28    5.31    0.05
- x1       1       717 754  61  153.50 1.7e-06

Step:  AIC=22.41
y ~ x1 + x2 + x4 + I(x2^2) + x2:x4

          Df Sum of Sq RSS AIC F value   Pr(F)
+ x3       1         6  23  22    1.45    0.27
<none>                  29  22
+ x1:x4    1         2  27  23    0.48    0.51
+ x1:x2    1         2  27  23    0.46    0.52
- x2:x4    1         8  37  24    2.03    0.20
+ I(x1^2)  1      0.26  29  24    0.05    0.82
- I(x2^2)  1        13  42  25    3.04    0.12
- x1       1       647 676  61  156.46 4.8e-06

Step:  AIC=21.6
y ~ x1 + x2 + x4 + I(x2^2) + x3 + x2:x4

          Df Sum of Sq  RSS  AIC F value Pr(F)
+ x1:x3    1       4.6 18.7 20.8    1.22 0.319
<none>                 23.3 21.6
- x3       1       5.6 29.0 22.4    1.45 0.274
+ x1:x4    1       1.5 21.9 22.8    0.33 0.588
+ x1:x2    1       1.3 22.0 22.8    0.30 0.608
+ x3:x4    1       1.2 22.1 22.9    0.27 0.627
+ I(x1^2)  1       0.3 23.0 23.4    0.08 0.794
+ x2:x3    1       0.3 23.0 23.4    0.06 0.815
- x2:x4    1      12.3 35.7 25.1    3.17 0.125
- I(x2^2)  1      17.4 40.8 26.9    4.49 0.078
- x1       1      42.7 66.1 33.1   10.99 0.016

Step:  AIC=20.76
y ~ x1 + x2 + x4 + I(x2^2) + x3 + x2:x4 + x1:x3

          Df Sum of Sq  RSS  AIC F value Pr(F)
+ I(x1^2)  1      14.3  4.4  4.0   12.97 0.023
<none>                 18.7 20.8
+ x1:x4    1       2.1 16.7 21.2    0.50 0.517
- x1:x3    1       4.6 23.3 21.6    1.22 0.319
+ x3:x4    1       1.3 17.5 21.8    0.29 0.617
+ x1:x2    1       0.9 17.8 22.1    0.21 0.670
+ x2:x3    1       0.1 18.6 22.7    0.03 0.875
- x2:x4    1      14.5 33.2 26.2    3.87 0.106
- I(x2^2)  1      20.8 39.5 28.5    5.55 0.065

Step:  AIC=3.97
```

```
y ~ x1 + x2 + x4 + I(x2^2) + x3 + I(x1^2) + x2:x4 + x1:x3

          Df Sum of Sq   RSS   AIC F value Pr(F)
<none>                  4.42  3.97
+ x1:x4    1      0.14  4.28  5.56    0.10 0.775
+ x1:x2    1      0.12  4.30  5.61    0.08 0.790
+ x3:x4    1      0.08  4.34  5.74    0.05 0.831
+ x2:x3    1      0.05  4.37  5.82    0.04 0.863
- x2:x4    1      7.60 12.02 14.98    6.88 0.059
- I(x1^2)  1     14.33 18.74 20.76   12.97 0.023
- I(x2^2)  1     16.00 20.42 21.87   14.49 0.019
- x1:x3    1     18.56 22.97 23.40   16.80 0.015

Call:
lm(formula = y ~ x1 + x2 + x4 + I(x2^2) + x3 + I(x1^2) + x2:x4 +     x1:x3, data = hald)

Coefficients:
(Intercept)           x1           x2           x4      I(x2^2)
  -194.3328      -0.6691       6.1165       2.5366      -0.0344
         x3      I(x1^2)        x2:x4        x1:x3
     1.0515       0.1056      -0.0209       0.2588
```