Model selection: Examples

Recap

Given a collection of predictors, our goal is now to find

the simplest possible model that adequately explains the data

- the principle of *parsimony*.

We typically consider a 'largest' (most complex) model and look for simplifications of it. We may use partial F tests to do this.

Recap (cont.)

Typical strategies are

- Fit all possible models using the available predictors.
- *Forward selection:* start from the intercept only model, and try to add predictors until the model cannot be improved further.
- *Backward selection:* start from the most complex model, and try to remove predictors until the model cannot be simplified further without compromising the quality of fit.
- *Stepwise selection:* use a combination of forward and backward steps.

Recap (cont.)

In `R`, several functions are useful

- `drop1`: single term deletions;
- `add1`: adds terms to an existing model;
- `update`: adds terms to or deletes terms from an existing model;
- `step`: automated procedure for selection;
- in the `leaps` library: `regsubsets` for subset selection.

Stepwise model selection: Example

The following data set contains information on the rut depth wear on thirty one road surfaces (measured in mm per million wheel passes) prepared under various experimental conditions describing the construction method used for each “course". These data are recorded in a data frame `p9.10` in the package `MPV` which contains the following columns:

- y: change in rut depth/million wheel passes (log scale)
- x_1: viscosity (log scale);
- x_2: percentage of asphalt in surface course;
- x_3: percentage of asphalt in base course;
- x_4: indicator recording which run (Run 1, Run2) each datum is recorded for;
- x_5: percentage of fines in surface course;
- x_6: percentage of voids in surface course.

Stepwise model selection: Example (cont.)

We aim to identify which of the predictors influence the response.

```
> library(MPV)
> Asphalt<-p9.10
> Asphalt$x4<-factor(Asphalt$x4,labels=c('Run1','Run2'))
> str(Asphalt)
'data.frame':   31 obs. of  7 variables:
 $ y : num  0.829 1.114 1.169 1.1 0.916 ...
 $ x1: num  0.447 0.146 0.146 0.519 0.23 ...
 $ x2: num  4.68 5.19 4.82 4.85 4.86 5.16 4.82 4.86 4.78 5.16 ...
 $ x3: num  4.87 4.5 4.73 4.76 4.95 4.45 5.05 4.7 4.84 4.76 ...
 $ x4: Factor w/ 2 levels "Run1","Run2": 1 1 1 1 1 1 1 1 1 1 ...
 $ x5: num  8.4 6.5 7.9 8.3 8.4 7.4 6.8 8.6 6.7 7.7 ...
 $ x6: num  4.92 4.56 5.32 4.87 3.78 ...
```

We may assess the influence of each predictor in an informal way by inspecting their correlations with the response and with each other

Stepwise model selection: Example (cont.)

```
> round(cor(p9.10),3)
        y     x1     x2     x3     x4     x5     x6
y   1.000 -0.893  0.145  0.099 -0.912  0.257 -0.409
x1 -0.893  1.000 -0.126 -0.217  0.939 -0.303  0.469
x2  0.145 -0.126  1.000 -0.187 -0.116 -0.229 -0.368
x3  0.099 -0.217 -0.187  1.000 -0.120  0.437 -0.027
x4 -0.912  0.939 -0.116 -0.120  1.000 -0.234  0.406
x5  0.257 -0.303 -0.229  0.437 -0.234  1.000  0.116
x6 -0.409  0.469 -0.368 -0.027  0.406  0.116  1.000
```

From this we note that x_1 and x_4 are highly correlated with the response, but also with each other. There appear to be some other moderately large correlations.

Plots of the data reveal more of the structure: we plot

- Figure 1: scatterplots of y versus each predictor;
- Figure 2: boxplots of the continuous predictor versus x_4.

We see that there is interesting structure amongst the x_js.

Scatterplot

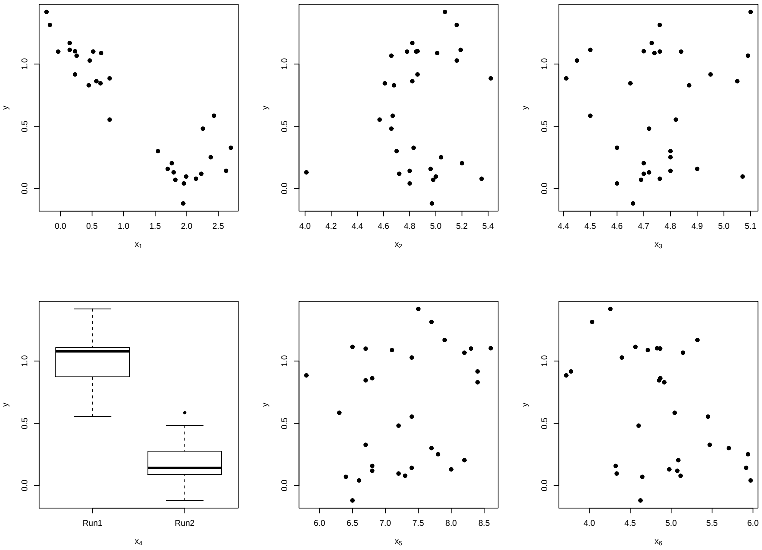

Figure: Scatterplots of y vs x_j, $j = 1, \ldots, 6$

Box plot

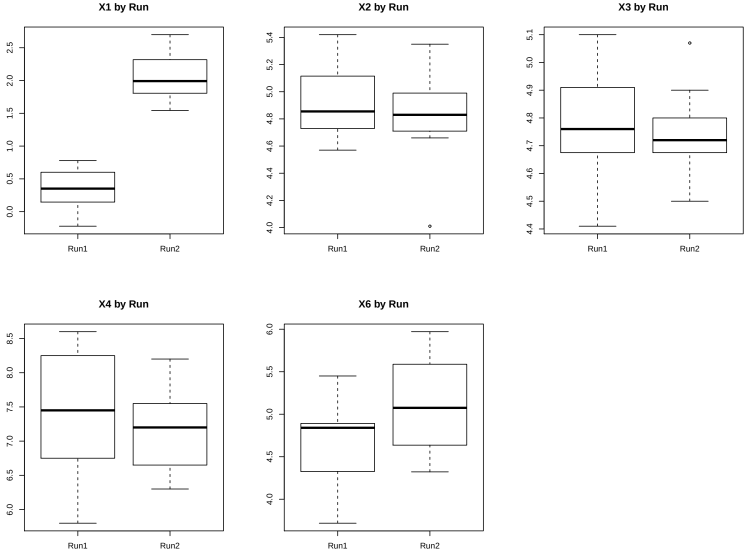

Figure: Boxplots of $x_j, j = 1, 2, 3, 5, 6$ vs x_4

Model selection

We begin the best model identification by examining the additive model that fits all predictors as main effects:

```
> fit1<-lm(y~x1+x2+x3+x4+x5+x6,data=Asphalt)
> summary(fit1)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.31975    1.53757   0.858   0.3992
x1          -0.14658    0.12872  -1.139   0.2661
x2           0.05625    0.15180   0.371   0.7142
x3          -0.15483    0.24140  -0.641   0.5274
x4Run2      -0.56213    0.22176  -2.535   0.0182 *
x5           0.03681    0.06156   0.598   0.5555
x6          -0.01220    0.08053  -0.151   0.8809
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2021 on 24 degrees of freedom
Multiple R-squared:  0.8482,    Adjusted R-squared:  0.8102
F-statistic: 22.35 on 6 and 24 DF,  p-value: 1.009e-08
```

Model selection (cont.)

This model achieves a reasonably high R^2, but only one predictor appears to be significantly influential; the factor predictor `x4` appears to yield a lower mean response for level `Run2` than for the baseline `Run1`.

We now attempt stepwise elimination starting from the main effects model: we focus first on F-test comparisons:

```
> drop1(fit1,test='F')
Single term deletions
Model:
y ~ x1 + x2 + x3 + x4 + x5 + x6
       Df Sum of Sq     RSS     AIC F value  Pr(>F)
<none>              0.98005 -93.078
x1      1  0.052949 1.03300 -93.447  1.2966 0.26607
x2      1  0.005607 0.98566 -94.901  0.1373 0.71422
x3      1  0.016798 0.99685 -94.551  0.4114 0.52736
x4      1  0.262391 1.24244 -87.724  6.4256 0.01818 *
x5      1  0.014602 0.99465 -94.620  0.3576 0.55546
x6      1  0.000937 0.98099 -95.049  0.0229 0.88086
```

Model selection (cont.)

This suggests that we could potentially drop any predictor apart from `x4`. Standard elimination methods would eliminate one predictor at a time, starting with the apparently least significant.

Here we proceed in a more ambitious fashion, and attempt to drop `x2`, `x3`, `x5` and `x6` simultaneously: the `update` function is used to define the new model for fitting.

```
> fit2<-update(fit1,~.-x2-x3-x5-x6)
> anova(fit2,fit1,test='F')
Analysis of Variance Table

Model 1: y ~ x1 + x4
Model 2: y ~ x1 + x2 + x3 + x4 + x5 + x6
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1     28 1.01123
2     24 0.98005  4  0.031181 0.1909 0.9408
```

Model selection (cont.)

Line 59 reveals that the model $X_1 + X_2$ is an adequate simplification of the more complex model $X_1 + X_2 + X_3 + X_4 + X_5 + X_6$ (the p-value is 0.9408, indicating no significant loss in fit). The model summary reveals that the R^2 quantity is still quite high (line 70).

```
60  > summary(fit2)
61  Coefficients:
62              Estimate Std. Error t value Pr(>|t|)
63  (Intercept)  1.07490    0.05968  18.011  < 2e-16 ***
64  x1          -0.14885    0.10677  -1.394  0.17424
65  x4Run2      -0.57317    0.19860  -2.886  0.00743 **
66  ---
67  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
68
69  Residual standard error: 0.19 on 28 degrees of freedom
70  Multiple R-squared:  0.8434,    Adjusted R-squared:  0.8322
71  F-statistic: 75.37 on 2 and 28 DF,  p-value: 5.356e-12
```

Model selection (cont.)

If we want to simplify further we can check using the `drop1` function:

```
> drop1(fit2,test='F')
Single term deletions

Model:
y ~ x1 + x4
       Df Sum of Sq    RSS      AIC F value   Pr(>F)
<none>              1.0112 -100.107
x1      1  0.070194 1.0814 -100.027  1.9436 0.174245
x4      1  0.300800 1.3120  -94.035  8.3288 0.007433 **
```

Model selection (cont.)

The indication is that x_1 can also be dropped.

```
> fit3<-update(fit2,~.-x1); summary(fit3)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.02455    0.04828   21.22  < 2e-16 ***
x4Run2      -0.83316    0.06940  -12.01 8.97e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.1931 on 29 degrees of freedom
Multiple R-squared:  0.8325,    Adjusted R-squared:  0.8267
F-statistic: 144.1 on 1 and 29 DF,  p-value: 8.974e-13
```

Model selection (cont.)

However, we now try to add interaction terms: a natural place to start is the interaction $X_1 : X_4$, so we attempt to fit the model

$$X_1 + X_4 + X_1 : X_4$$

recalling the convention on interactions ('no interactions without main effects').

Model selection (cont.)

```
> fit4<-update(fit3,~.+x1+x1:x4)
> anova(fit3,fit4,test='F')
Analysis of Variance Table

Model 1: y ~ x4
Model 2: y ~ x4 + x1 + x4:x1
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     29 1.08143
2     27 0.60764  2   0.47378 10.526 0.0004172 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> summary(fit4)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.20274    0.05595  21.495  < 2e-16 ***
x4Run2      -1.40384    0.25111  -5.591 6.26e-06 ***
x1          -0.52676    0.12275  -4.291 0.000204 ***
x4Run2:x1    0.71501    0.16884   4.235 0.000237 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.15 on 27 degrees of freedom
```

Model selection (cont.)

```
113 Multiple R-squared:  0.9059,    Adjusted R-squared:  0.8954
114 F-statistic: 86.62 on 3 and 27 DF,  p-value: 5.684e-14
```

Line 98 notes that the model X_4 is not an adequate simplification of the more complex model $X_1 + X_4 + X_1 : X_4$ as the null hypothesis that removes the X_1 main effect and interaction is rejected (p-value equal to 0.0004172).

Model selection (cont.)

Using the `drop1` function we verify that the interaction term should not be dropped.

```
> drop1(fit4,test='F')
Single term deletions

Model:
y ~ x4 + x1 + x4:x1
       Df Sum of Sq     RSS     AIC F value    Pr(>F)
<none>                0.60764 -113.90
x4:x1   1   0.40359 1.01123 -100.11  17.933 0.0002372 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The success of the interaction model encourages us to try other interactions: to fit all second order interactions involving `x4` and the continuous predictors, we may type

```
>fit5<-lm(y~x4*(x1+x2+x3+x5+x6),data=Asphalt)
```

Model selection (cont.)

This yields the following model fit:

```
> summary(fit5)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.55427    3.12640  -1.137  0.26973
x4Run2       5.42570    3.59232   1.510  0.14740
x1          -0.43526    0.15342  -2.837  0.01054 *
x2           0.64022    0.33846   1.892  0.07389 .
x3           0.13316    0.26854   0.496  0.62567
x5           0.04104    0.05870   0.699  0.49299
x6           0.13774    0.12476   1.104  0.28335
x4Run2:x1    0.65813    0.20397   3.227  0.00444 **
x4Run2:x2   -0.74479    0.36646  -2.032  0.05633 .
x4Run2:x3   -0.53269    0.46713  -1.140  0.26832
x4Run2:x5    0.04697    0.10422   0.451  0.65730
x4Run2:x6   -0.21067    0.15399  -1.368  0.18725
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1521 on 19 degrees of freedom
Multiple R-squared:  0.9319,    Adjusted R-squared:  0.8925
F-statistic: 23.65 on 11 and 19 DF,  p-value: 9.046e-09
```

Model selection (cont.)

We now try to drop the apparently unimportant terms:

```
> fit6<-update(fit5,~.-x3-x5-x6-x3:x4-x5:x4-x6:x4)
> anova(fit6,fit5,test='F')
Analysis of Variance Table

Model 1: y ~ x4 + x1 + x2 + x4:x1 + x4:x2
Model 2: y ~ x4 * (x1 + x2 + x3 + x5 + x6)
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1     25 0.51104
2     19 0.43946  6   0.07158 0.5158 0.7891
```

From this analysis, we retain the model

$$X_1 + X_2 + X_4 + X_1 : X_4 + X_2 : X_4$$

Model selection (cont.)

Using `drop1` we may check for further simplification

```
> drop1(fit6,test='F')
Single term deletions

Model:
y ~ x4 + x1 + x2 + x4:x1 + x4:x2
       Df Sum of Sq     RSS     AIC F value    Pr(>F)
<none>               0.51104 -115.26
x4:x1   1   0.36505 0.87609 -100.56 17.8584 0.0002768 ***
x4:x2   1   0.09572 0.60675 -111.94  4.6825 0.0402258 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Neither of the interactions $X_1 : X_4$ or $X_2 : X_4$ can be dropped on the basis of the two F-test results.

Model selection (cont.)

For a further comparison, we compare the simpler

$$X_1 + X_4 + X_1 : X_4$$

with the more complex model

$$X_1 + X_2 + X_4 + X_1 : X_4 + X_2 : X_4$$

```
167  > anova(fit4,fit6,test='F')
168  Analysis of Variance Table
169
170  Model 1: y ~ x4 + x1 + x4:x1
171  Model 2: y ~ x4 + x1 + x2 + x4:x1 + x4:x2
172    Res.Df     RSS Df Sum of Sq     F Pr(>F)
173  1     27 0.60764
174  2     25 0.51104  2  0.096608 2.363 0.1148
```

The model seems to perform adequately, giving a high R^2 value (line 189).

Model selection (cont.)

```
> summary(fit6)
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.1990     0.7739  -0.257 0.799154
x4Run2        0.7180     1.0080   0.712 0.482847
x1           -0.4924     0.1185  -4.155 0.000332 ***
x2            0.2833     0.1560   1.816 0.081450 .
x4Run2:x1     0.6849     0.1621   4.226 0.000277 ***
x4Run2:x2    -0.4337     0.2004  -2.164 0.040226 *
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 0.143 on 25 degrees of freedom
Multiple R-squared:  0.9208,    Adjusted R-squared:  0.905
F-statistic: 58.16 on 5 and 25 DF,  p-value: 5.828e-13
```

Model selection (cont.)

For a final analysis, we attempt to add in a three-way interaction.

```
> fit7<-update(fit6,~.+x1:x2:x4)
> drop1(fit7,test='F')
Single term deletions

Model:
y ~ x4 + x1 + x2 + x4:x1 + x4:x2 + x4:x1:x2
         Df Sum of Sq     RSS     AIC F value Pr(>F)
<none>                0.46666 -114.08
x4:x1:x2  2  0.044379 0.51104 -115.26  1.0936 0.3518
```

Model selection (cont.)

However, there is no evidence that the three-way interaction is useful in the model. Hence a reasonable model seems to be

$$X_1 + X_2 + X_4 + X_1 : X_4 + X_2 : X_4 = (X_1 + X_2) * X_4$$

which uses a six parameter conditional mean.

The model fits a different two-dimensional plane (in variables (`x1`, `x2`)) through the data for each of the levels of the Run factor predictor (`x4`).

Model selection (cont.)

The model can be written

$$\mathbb{E}_{Y_i|X_i}[Y_i|\mathbf{x}_i] = \begin{cases} \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} & x_4 = \texttt{Run1} \\ (\beta_0 + \beta_{41}^{C}) + (\beta_1 + \beta_{141}^{C}) x_{i1} + (\beta_2 + \beta_{241}^{C}) x_{i2} & x_4 = \texttt{Run2} \end{cases}$$

where

$\widehat{\beta}_0$	Line 178	`(Intercept)`	-0.1990
$\widehat{\beta}_{41}^{C}$	Line 179	`x4Run2`	0.7180
$\widehat{\beta}_1$	Line 180	`x1`	-0.4924
$\widehat{\beta}_2$	Line 181	`x2`	0.2833
$\widehat{\beta}_{141}^{C}$	Line 182	`x4Run2:x1`	0.6849
$\widehat{\beta}_{241}^{C}$	Line 183	`x4Run2:x2`	-0.4337

Residual plots

Residual plots seem reasonable, although there may be some mild evidence that the variance is not constant.

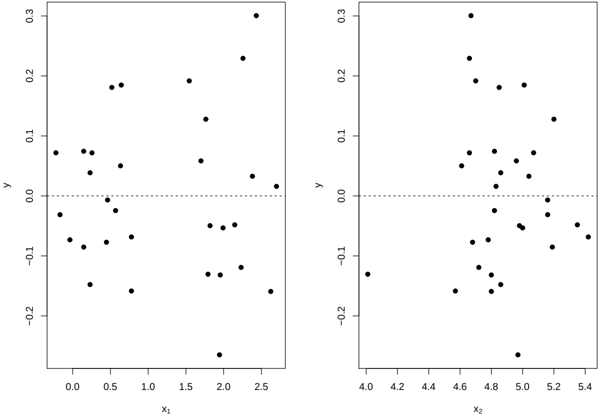

Figure: Residuals vs X_1 (left) and X_2 (right)

Other Criteria: Example

We have the model for three continuous predictors X_1, X_2, X_3

$$Y_i = 2 + 2x_{i1} + 2x_{i2} - 2x_{i1}x_{i2} + \epsilon_i$$

with $\sigma^2 = 1$. We have $n = 200$.

Other Criteria: Example (cont.)

```
> fit0<-lm(Y~1)
> fit1<-lm(Y~x1)
> fit2<-lm(Y~x2)
> fit3<-lm(Y~x3)
> fit12<-lm(Y~x1+x2)
> fit13<-lm(Y~x1+x3)
> fit23<-lm(Y~x2+x3)
> fit123<-lm(Y~x1+x2+x3)
> fit12i<-lm(Y~x1*x2)
> fit13i<-lm(Y~x1*x3)
> fit23i<-lm(Y~x2*x3)
> fit123i<-lm(Y~x1*x2*x3)

> bigs.hat<-summary(fit123i)$sigma
```

Other Criteria: Example (cont.)

```
criteria.eval<-function(fit.obj,nv,bigsig.hat){
          cvec<-rep(0,5)
          SSRes<-sum(residuals(fit.obj)^2)
          p<-length(coef(fit.obj))

          #R squared
          cvec[1]<-summary(fit.obj)$r.squared
          #Adjusted R squared
          cvec[2]<-summary(fit.obj)$adj.r.squared
          #Cp
          cvec[3]<-SSRes/bigsig.hat^2-nv+2*p
          #AIC in R computes
          #n*log(sum(residuals(fit.obj)^2)/n)+
          #2*(length(coef(fit.obj))+1)+n*log(2*pi)+n
          cvec[4]<-AIC(fit.obj)
          #BIC in R computes
          #n*log(sum(residuals(fit.obj)^2)/n)+
          #log(n)*(length(coef(fit.obj))+1)+n*log(2*pi)+n
          cvec[5]<-BIC(fit.obj)

          return(cvec)
}
```

Other Criteria: Example (cont.)

```
> cvals<-matrix(0,nrow=12,ncol=5)
> cvals[1,]<-criteria.eval(fit0,n,bigs.hat)
> cvals[2,]<-criteria.eval(fit1,n,bigs.hat)
> cvals[3,]<-criteria.eval(fit2,n,bigs.hat)
> cvals[4,]<-criteria.eval(fit3,n,bigs.hat)
> cvals[5,]<-criteria.eval(fit12,n,bigs.hat)
> cvals[6,]<-criteria.eval(fit13,n,bigs.hat)
> cvals[7,]<-criteria.eval(fit23,n,bigs.hat)
> cvals[8,]<-criteria.eval(fit123,n,bigs.hat)
> cvals[9,]<-criteria.eval(fit12i,n,bigs.hat)
> cvals[10,]<-criteria.eval(fit13i,n,bigs.hat)
> cvals[11,]<-criteria.eval(fit23i,n,bigs.hat)
> cvals[12,]<-criteria.eval(fit123i,n,bigs.hat)
>
> Criteria<-data.frame(cvals)
> names(Criteria)<-c('Rsq','Adj.Rsq','Cp','AIC','BIC')
>
> rownames(Criteria)<-c('1','x1','x2','x3','x1+x2','x1+x3','x2+x3',
+                    'x1+x2+x3','x1*x2','x1*x3','x2*x3','x1*x2*x3')
```

Other Criteria: Example (cont.)

```
56  > round(Criteria,4)
57                Rsq Adj.Rsq        Cp       AIC       BIC
58  1          0.0000  0.0000 1808.7895 1016.2116 1022.8082
59  x1         0.4426  0.4398  922.5695  901.3145  911.2094
60  x2         0.0207  0.0158 1769.2514 1014.0284 1023.9233
61  x3         0.0056  0.0006 1799.5465 1017.0879 1026.9829
62  x1+x2      0.5266  0.5218  755.9216  870.6292  883.8224
63  x1+x3      0.4827  0.4775  844.0514  888.3734  901.5667
64  x2+x3      0.0237  0.0138 1765.1450 1015.4060 1028.5992
65  x1+x2+x3   0.5268  0.5195  757.6328  872.5684  889.0599
66  x1*x2      0.9041  0.9026    0.4429  553.3132  569.8048
67  x1*x3      0.7600  0.7564  289.5338  736.7485  753.2401
68  x2*x3      0.1734  0.1608 1466.7742  984.1201 1000.6117
69  x1*x2*x3   0.9043  0.9008    8.0000  560.8523  590.5372
```

Line 67 reveals the model $X_1 * X_2 = X_1 + X_2 + X_1 : X_2$ as most appropriate model.

Textbook Example: step

```
> fit5<-lm(y~x4*(x1+x2+x3+x5+x6),data=Asphalt)   #AIC version
> fit.by.step.aic<-step(fit5,k=2)
Start:  AIC=-107.94
y ~ x4 * (x1 + x2 + x3 + x5 + x6)

        Df Sum of Sq     RSS      AIC
- x4:x5  1  0.004698 0.44415 -109.613
<none>               0.43946 -107.942
- x4:x3  1  0.030077 0.46953 -107.890
- x4:x6  1  0.043290 0.48275 -107.030
- x4:x2  1  0.095536 0.53499 -103.844
- x4:x1  1  0.240794 0.68025  -96.398

Step:  AIC=-109.61
y ~ x4 + x1 + x2 + x3 + x5 + x6 + x4:x1 + x4:x2 + x4:x3 + x4:x6

        Df Sum of Sq     RSS      AIC
- x4:x3  1  0.025434 0.46959 -109.886
<none>               0.44415 -109.613
- x5     1  0.030762 0.47492 -109.537
- x4:x6  1  0.039507 0.48366 -108.971
- x4:x2  1  0.112030 0.55618 -104.640
- x4:x1  1  0.241310 0.68546  -98.161

Step:  AIC=-109.89
y ~ x4 + x1 + x2 + x3 + x5 + x6 + x4:x1 + x4:x2 + x4:x6

        Df Sum of Sq     RSS     AIC
- x3     1   0.00083 0.47041 -111.83
- x5     1   0.02093 0.49052 -110.53
- x4:x6  1   0.02586 0.49545 -110.22
```

Textbook Example: step (cont.)

```
<none>                 0.46959 -109.89
- x4:x2  1   0.08960 0.55919 -106.47
- x4:x1  1   0.34502 0.81461  -94.81

Step:  AIC=-111.83
y ~ x4 + x1 + x2 + x5 + x6 + x4:x1 + x4:x2 + x4:x6

        Df Sum of Sq     RSS      AIC
- x5     1   0.02012 0.49054 -112.533
- x4:x6  1   0.02656 0.49698 -112.129
<none>               0.47041 -111.832
- x4:x2  1   0.12486 0.59528 -106.534
- x4:x1  1   0.35250 0.82291  -96.496

Step:  AIC=-112.53
y ~ x4 + x1 + x2 + x6 + x4:x1 + x4:x2 + x4:x6

        Df Sum of Sq     RSS      AIC
- x4:x6  1   0.01826 0.50880 -113.400
<none>               0.49054 -112.533
- x4:x2  1   0.10585 0.59639 -108.476
- x4:x1  1   0.36400 0.85454  -97.326

Step:  AIC=-113.4
y ~ x4 + x1 + x2 + x6 + x4:x1 + x4:x2

        Df Sum of Sq     RSS      AIC
- x6     1   0.00224 0.51104 -115.264
<none>               0.50880 -113.400
- x4:x2  1   0.09138 0.60018 -110.280
- x4:x1  1   0.34584 0.85464  -99.323
```

Textbook Example: `step` (cont.)

```

Step:  AIC=-115.26
y ~ x4 + x1 + x2 + x4:x1 + x4:x2

        Df Sum of Sq     RSS     AIC
<none>               0.51104 -115.26
- x4:x2  1   0.09572 0.60675 -111.94
- x4:x1  1   0.36505 0.87609 -100.56

> summary(fit.by.step.aic)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.1990     0.7739  -0.257 0.799154
x4Run2        0.7180     1.0080   0.712 0.482847
x1           -0.4924     0.1185  -4.155 0.000332 ***
x2            0.2833     0.1560   1.816 0.081450 .
x4Run2:x1     0.6849     0.1621   4.226 0.000277 ***
x4Run2:x2    -0.4337     0.2004  -2.164 0.040226 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.143 on 25 degrees of freedom
Multiple R-squared:  0.9208,    Adjusted R-squared:  0.905
F-statistic: 58.16 on 5 and 25 DF,  p-value: 5.828e-13
```

Textbook Example: regsubsets

```
> library(leaps)
> fit.by.leaps<-regsubsets(y~x4*(x1+x2+x3+x5+x6),data=Asphalt,method='exhaustive')
> summary(fit.by.leaps)
Subset selection object
Call: regsubsets.formula(y ~ x4 * (x1 + x2 + x3 + x5 + x6), data = Asphalt,
    method = "exhaustive")
11 Variables  (and intercept)
          Forced in Forced out
x4Run2        FALSE      FALSE
x1            FALSE      FALSE
x2            FALSE      FALSE
x3            FALSE      FALSE
x5            FALSE      FALSE
x6            FALSE      FALSE
x4Run2:x1     FALSE      FALSE
x4Run2:x2     FALSE      FALSE
x4Run2:x3     FALSE      FALSE
x4Run2:x5     FALSE      FALSE
x4Run2:x6     FALSE      FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         x4Run2 x1  x2  x3  x5  x6  x4Run2:x1 x4Run2:x2 x4Run2:x3 x4Run2:x5 x4Run2:x6
1  ( 1 ) " "    " " " " " " " " " " " "       " "       "*"       " "       " "
2  ( 1 ) " "    "*" " " " " " " " " " "       " "       "*"       " "       " "
3  ( 1 ) " "    "*" " " " " " " " " "*"       " "       "*"       " "       " "
4  ( 1 ) " "    "*" "*" " " " " " " "*"       "*"       " "       " "       " "
5  ( 1 ) " "    "*" "*" " " " " " " "*"       "*"       " "       "*"       " "
6  ( 1 ) "*"    "*" "*" " " "*" " " "*"       "*"       " "       " "       " "
7  ( 1 ) "*"    "*" "*" " " "*" " " "*"       "*"       "*"       " "       " "
8  ( 1 ) "*"    "*" "*" " " "*" "*" "*"       "*"       " "       " "       "*"
```

Textbook Example: `regsubsets`

```
> summary(fit.by.leaps)$which
  (Intercept) x4Run2    x1    x2    x3    x5    x6 x4Run2:x1 x4Run2:x2 x4Run2:x3
1        TRUE  FALSE FALSE FALSE FALSE FALSE FALSE     FALSE     FALSE      TRUE
2        TRUE  FALSE  TRUE FALSE FALSE FALSE FALSE     FALSE     FALSE      TRUE
3        TRUE  FALSE  TRUE FALSE FALSE FALSE FALSE      TRUE     FALSE      TRUE
4        TRUE  FALSE  TRUE  TRUE FALSE FALSE FALSE      TRUE      TRUE     FALSE
5        TRUE  FALSE  TRUE  TRUE FALSE FALSE FALSE      TRUE      TRUE     FALSE
6        TRUE   TRUE  TRUE  TRUE FALSE  TRUE FALSE      TRUE      TRUE     FALSE
7        TRUE   TRUE  TRUE  TRUE FALSE  TRUE FALSE      TRUE      TRUE      TRUE
8        TRUE   TRUE  TRUE  TRUE FALSE  TRUE  TRUE      TRUE      TRUE     FALSE
  x4Run2:x5 x4Run2:x6
1     FALSE     FALSE
2     FALSE     FALSE
3     FALSE     FALSE
4     FALSE     FALSE
5      TRUE     FALSE
6     FALSE     FALSE
7     FALSE     FALSE
8     FALSE      TRUE
> summary(fit.by.leaps)$rsq
[1] 0.8364571 0.8474287 0.9081912 0.9192309 0.9211305 0.9228623 0.9249253 0.9271302
> summary(fit.by.leaps)$rss
[1] 1.0557586 0.9849311 0.5926761 0.5214089 0.5091459 0.4979660 0.4846483 0.4704142
> summary(fit.by.leaps)$bic
[1] -49.26311 -47.98186 -60.29351 -60.83104 -58.13485 -55.38915 -52.79552 -50.28564
```