STAT 571A — Advanced Statistical Regression Analysis

Chapter 9 NOTES

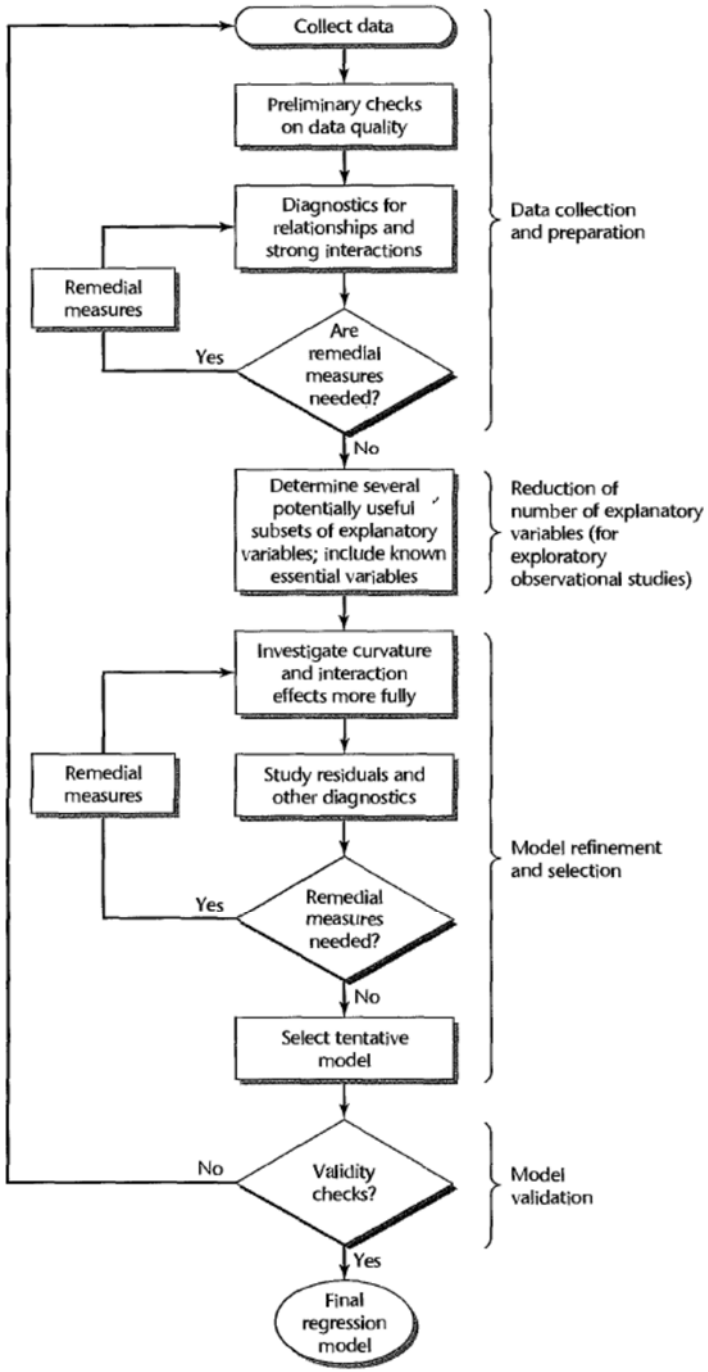
Model Building – I: Variable Selection

§9.1: Model-Building

- If all the X_k-variables are known and validated, building the MLR model is easy.
- But if there are questions as to which X_k's to use, it becomes a **model-building process**:
  - Data collection
  - Variable reduction
  - Model refinement
  - Model validation
- See (!) Fig. 9.1 $\longrightarrow$

Figure 9.1

Strategy for building an MLR model



§9.3: Model Selection

- We could approach model building from a (semi-)automated perspective.
- Suppose there are $P-1 < n$ *possible* X-variables available (incl. powers, transforms, interactions: X_k^2, $X_k^{1/2}$, $\log(X_k)$, X_kX_m, you name it)!
- The goal is to select a parsimonious **subset** of $p-1 < P-1$ predictors for the MLR model.

Variable Selection

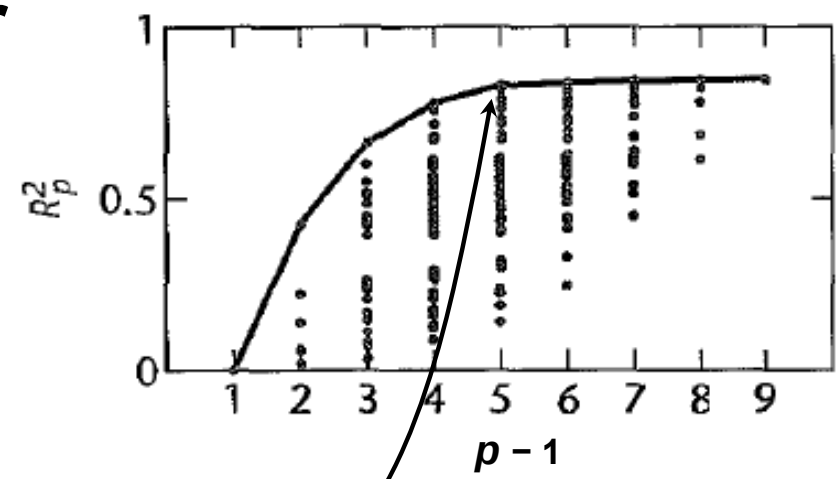
- For simplicity, we always include β_0. Thus if there are P–1 variables available, we have 2^{P-1} possible models.
- This gets big, fast:
  - P–1 = 4 $\Rightarrow$ 16 possible models (cf. Table 9.2)
  - P–1 = 10 $\Rightarrow$ 1024 possible models
- How to do this? Possible metrics from the book include: (a) R^2, (b) R_a^2, (c) C_p, (d) AIC/BIC, (e) PRESS.
- We study each, in turn.

(a) Maximum R_p^2

- One obvious way to measure the quality of a set of p–1 predictor variables is to compute the R^2 from their MLR.
- $\Rightarrow$ Among a group of different possible models, each with p parameters (p–1 predictors), choose the model with the <u>highest</u> R_p^2.
- (Notice: since $R^2 = 1 - (SSE/SSTO)$, this is identical to choosing the smallest SSE_p.)

Maximum R_p^2 (cont'd)

- But recall that every time we add a variable to an MLR model, R^2 cannot decrease! So, R_p^2 is a nondecreasing function of p. This will always lead to choosing p = P.
- In practice, we look for a diminishing return: after a certain p, the increase in R_p^2 should essentially flatten.



<u>Example</u>: Surgical Unit Data (CH09TA01)

- $Y' = ln\{\text{Survival time}\}$
- X_1 = Blood clotting score
- X_2 = Prognostic index
- X_3 = Enzyme test
- X_4 = Liver test
- Goal: determine best combination of X_k-variables for modeling $E\{Y'\}$

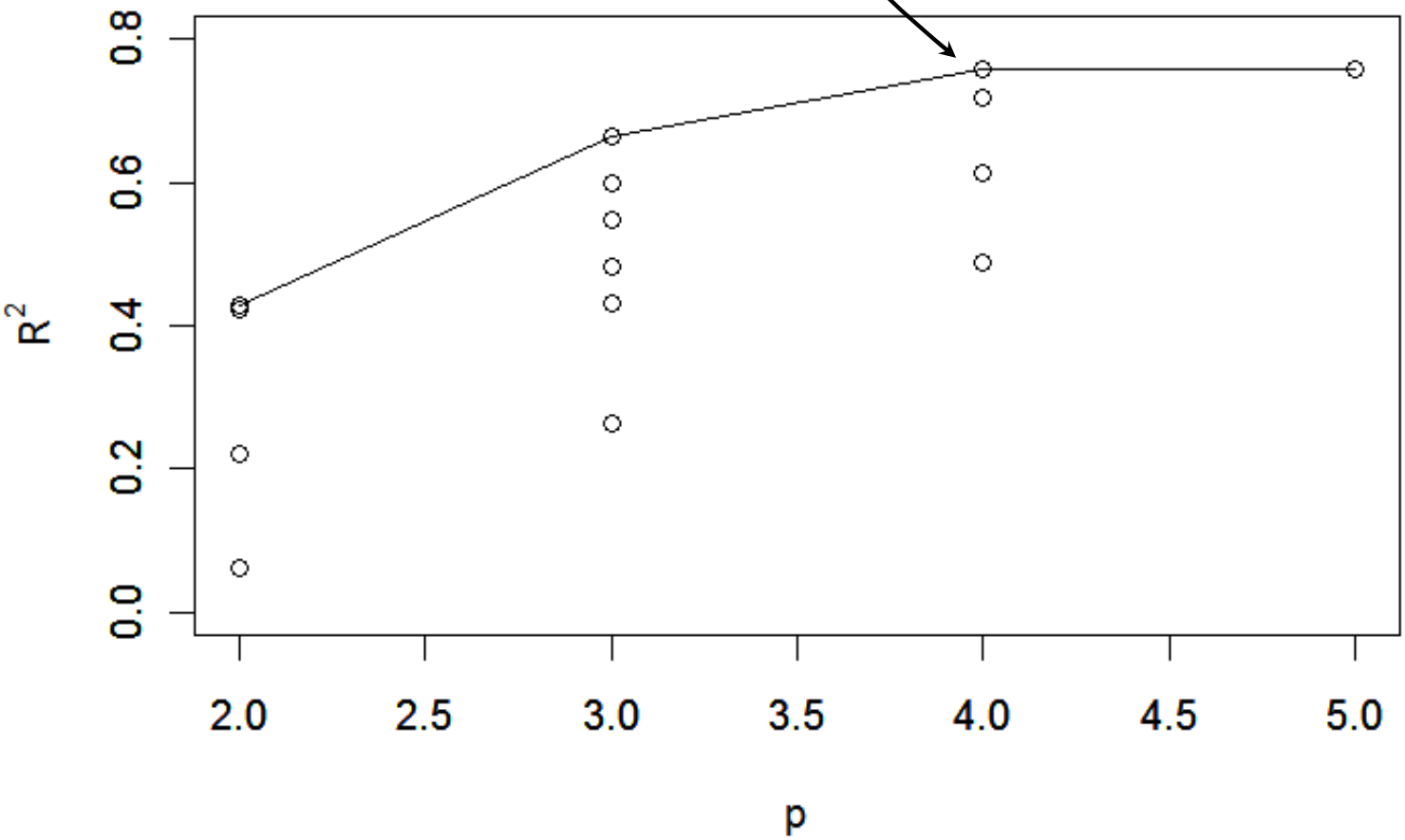
Surgical Unit Data (CH09TA01) (cont'd)

R_p^2 plot via R:

```
> library( leaps )
> CH09TA01.r2 = leaps( x=cbind(X1,X2,X3,X4),
                       y=Yprime, method='r2' )
> p = seq( min(CH09TA01.r2$size),
                       max(CH09TA01.r2$size) )
> plot( CH09TA01.r2$r2 ~ CH09TA01.r2$size ,
             ylab=expression(R^2), xlab='p' )
> Rp2 = by( data=CH09TA01.r2$r2,
   INDICES=factor(CH09TA01.r2$size), FUN=max )
> lines( Rp2 ~ p )
```

Surgical Unit Data (CH09TA01) (cont'd)

R_p^2 plot (cf. Fig. 9.4a): flattens at p=4 $\Rightarrow$ use X_1, X_2, X_3



(b) Maximum R_{ap}^2

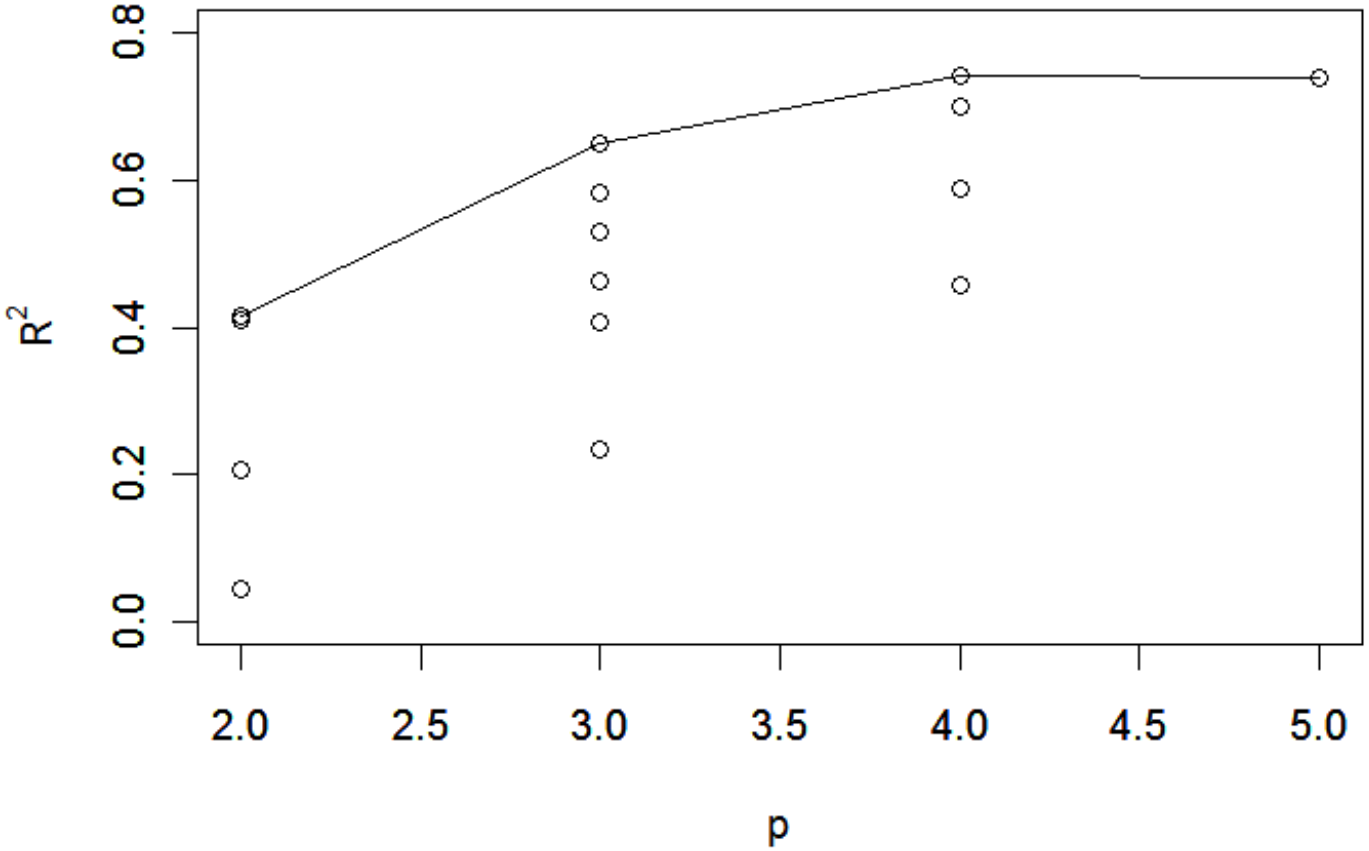
- To mitigate the increasing nature of R_p^2, we can move to the adjusted R^2,

$$R_a^2 = 1 - (MSE/MSTO).$$

- Recall that R_a^2 adjusts for arbitrary inclusion of variables. Thus we could aim to maximize R_{ap}^2 over increasing p ($\Leftrightarrow$ minimize MSE_p).
- The pattern will usually be very similar to R_p^2, but at least it is not guaranteed to always increase.

Surgical Unit Data (CH09TA01) (cont'd)

R^2_{ap} plot: use `method='adjr2'` in call to `leaps()` (cf. Fig. 9.4b). Best subset is X_1, X_2, X_3



(c) Mallow's C_p

A statistic due to C. Mallows is designed to find subsets of the P–1 variables that minimize a form of mean squared deviation; see equ. (9.8).

The target quantity is estimated by

$$C_p = \frac{SSE_p}{MSE(X_1 \ldots X_{P-1})} - (n - 2p)$$

and is known as **Mallow's C_p**.

Mallow's C_p (cont'd)

- As C_p drops, the quality of the fit improves, up to a point: when the expected fitted values roughly equal the mean responses,

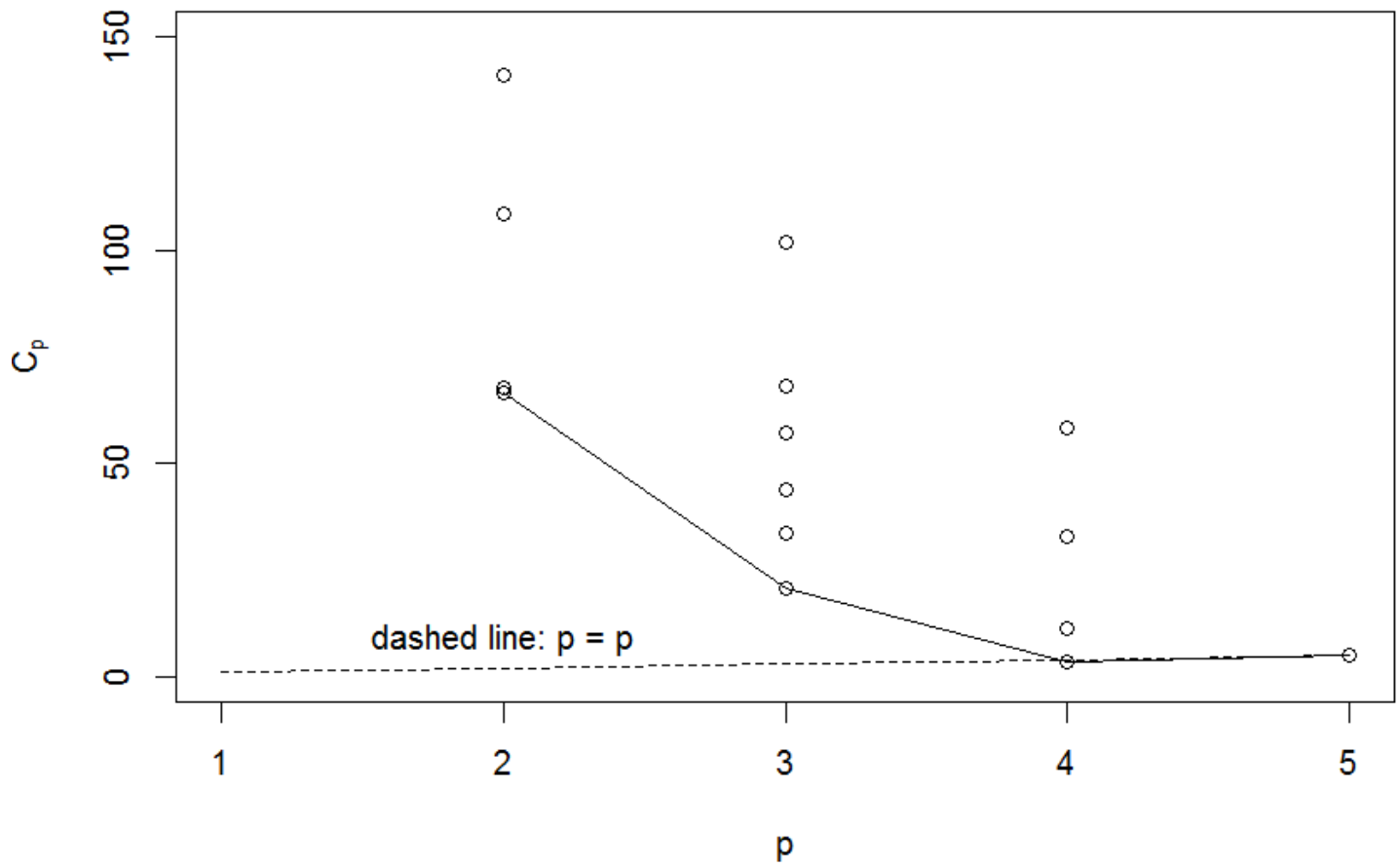
$$E\{C_p\} \approx p.$$

- So, plot C_p against p and look for
  (i) small C_p where
  (ii) $C_p \approx p$.

- (Values where $C_p < p$ are attributed to sampling variation and ignored.)

Surgical Unit Data (CH09TA01) (cont'd)

C_p plot: use `method='Cp'` in call to `leaps()` (cf. Fig. 9.4c). Best subset is again X_1, X_2, X_3



(d) AIC and BIC

An increasingly popular measure in statistics is the Information Criterion (IC).

The earliest was Akaike's IC ("AIC"):

$$AIC_p = \underbrace{n \log\{SSE_p\} - n \log\{n\}}_{\text{minimization target}} + \underbrace{2p}_{\text{penalty term}}$$

Select that p-parameter subset that minimizes AIC_p.

(Careful: some authors multiply by –2 or by –½ and then maximize.)

Schwarz' BIC

A popular alternative is Schwarz' Bayesian Criterion (SBC), also called the BIC:

$$BIC_p = \underbrace{n \log\{SSE_p\} - n \log\{n\}}_{\text{minimization target}} + \underbrace{p \log\{n\}}_{\text{penalty term}}$$

Select that p-parameter subset that minimizes BIC_p.

(BIC tends to more heavily penalize models with larger p.)

Surgical Unit Data (CH09TA01) (cont'd)

- X_k-variable selection search using AIC_p in R:
- Define baseline 'full model':

```
> fmCH09TA01.lm = lm( Yprime ~ X1+X2+X3+X4 )
```

- Use `step()` function (go 'backward' if starting with full model). `k=2` option calls AIC_p:

```
> step( fmCH09TA01.lm, direction="backward",
                                        k=2 )
```

Output follows →

Surgical Unit Data (CH09TA01) (cont'd)

X_k-variable search using AIC_p via `step()`:

```
Start:  AIC=-144.59
Yprime ~ X1 + X2 + X3 + X4

       Df Sum of Sq    RSS     AIC
- X4    1    0.0244 3.1085 -146.16
<none>               3.0841 -144.59
- X1    1    0.5309 3.6150 -138.01
- X2    1    1.8857 4.9698 -120.82
- X3    1    3.4842 6.5683 -105.76
```

Output continues →

Surgical Unit Data (CH09TA01) (cont'd)

`step()` search ends with selected min-AIC model: $Y' \sim X_1 + X_2 + X_3$:

```
Step:  AIC=-146.16
Yprime ~ X1 + X2 + X3

       Df Sum of Sq    RSS      AIC
<none>               3.1085 -146.161
- X1    1    1.2044 4.3129 -130.479
- X2    1    2.6740 5.7825 -114.644
- X3    1    6.3286 9.4371  -88.194

Call:
lm(formula = Yprime ~ X1 + X2 + X3)

Coefficients:
(Intercept)           X1           X2           X3
    3.76644      0.09547      0.01334      0.01644
```

Surgical Unit Data (CH09TA01) (cont'd)

- X_k-variable selection search using BIC_p.
- Define baseline 'full model':

```
> fmCH09TA01.lm = lm( Yprime ~ X1+X2+X3+X4 )
```

- Use `step()` function (go 'backward' if starting with full model). `k=log(n)` option uses BIC_p:

```
> n = length(Yprime)
> step( fmCH09TA01.lm, direction="backward",
                                    k=log(n) )
```

Output follows →

Surgical Unit Data (CH09TA01) (cont'd)

X_k-variable search using BIC_p via `step()`:

```
Start:  AIC=-134.64
Yprime ~ X1 + X2 + X3 + X4

       Df Sum of Sq    RSS      AIC
- X4    1    0.0244 3.1085 -138.205
<none>               3.0841 -134.642
- X1    1    0.5309 3.6150 -130.055
- X2    1    1.8857 4.9698 -112.867
- X3    1    3.4842 6.5683  -97.807
```

(Output lists 'AIC' throughout, but numbers are BIC_p, based on use of `k=log(n)` option.)

Surgical Unit Data (CH09TA01) (cont'd)

step() search ends with selected min-BIC model (even though it says 'AIC'):

$Y' \sim X_1 + X_2 + X_3$:

```
Step:  AIC=-138.21
Yprime ~ X1 + X2 + X3

       Df Sum of Sq    RSS      AIC
<none>               3.1085 -138.205
- X1    1    1.2044 4.3129 -124.512
- X2    1    2.6740 5.7825 -108.677
- X3    1    6.3286 9.4371  -82.227

Call:
lm(formula = Yprime ~ X1 + X2 + X3)

Coefficients:
(Intercept)         X1          X2          X3
    3.76644    0.09547    0.01334    0.01644
```

(e) PRESS

When prediction of a future $\hat{Y}_i$ is a central goal, we can study the prediction error for each observation.

Let $\hat{Y}_{i(i)}$ be the value predicted at observation *i* after <u>leaving</u> <u>Y_i</u> <u>out</u> of the MLR calculations. (A "leave-one-out," or LOO, predictor: a kind of cross-validation).

If the model predicts $\hat{Y}_{i(i)}$ well – even without Y_i being fit – it could be a good model.

PRESS (cont'd)

Do this LOO calculation for every Y_i. If the differences $(\hat{Y}_i - \hat{Y}_{i(i)})$ are all small, the model predicts well.

To avoid +/– cancelations, square the differences and sum into a **Prediction Sum of Squares**: $PRESS_p = \sum(\hat{Y}_i - \hat{Y}_{i(i)})^2$.

Goal is to find the p-parameter subset that minimizes $PRESS_p$.

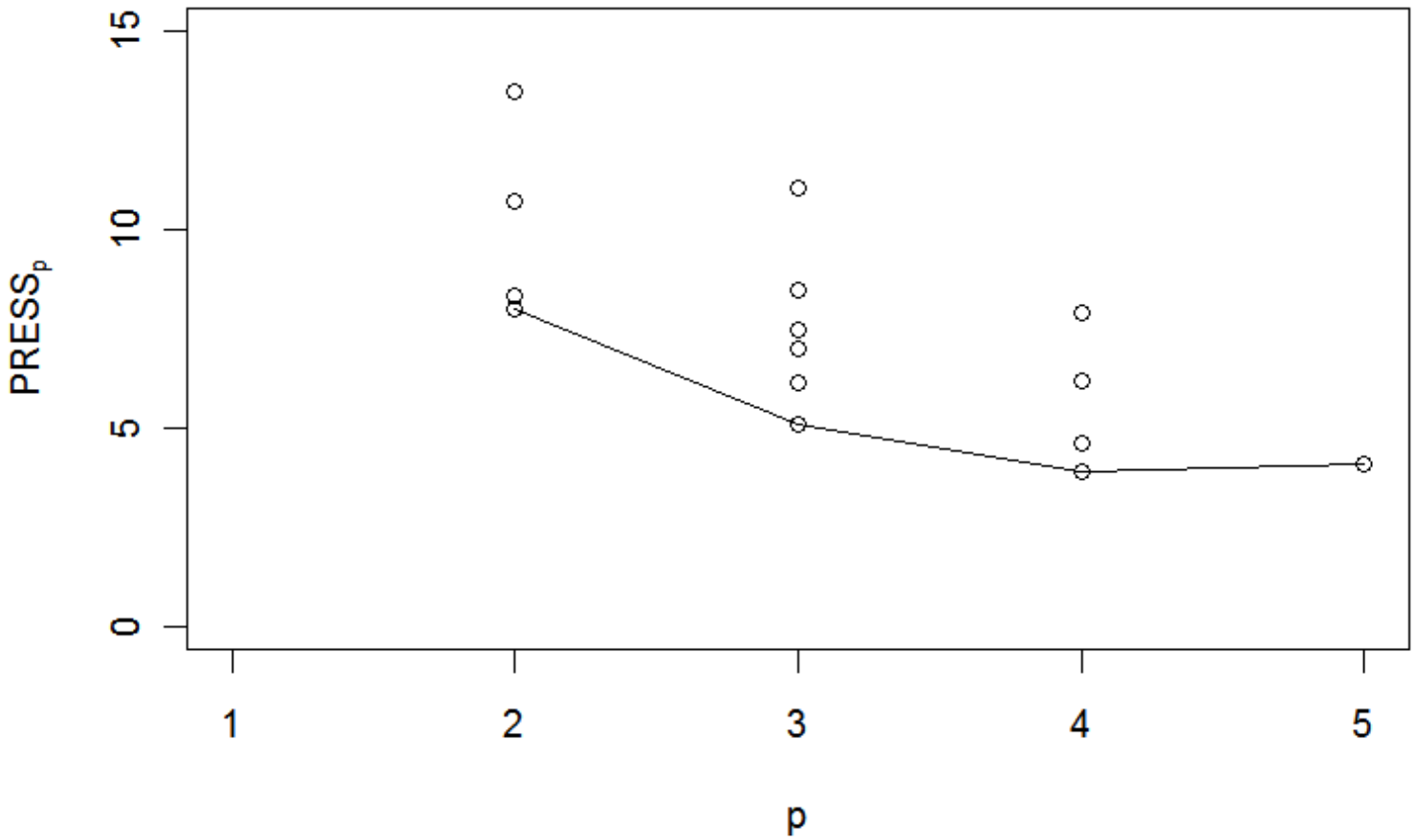
Surgical Unit Data (CH09TA01) (cont'd)

$PRESS_p$ plot via R and external *MPV* package:

```
> library( MPV )
> PRESSp = numeric( length(CH09TA01.r2$size) )
> PRESSp[1] = PRESS( lm( Yprime ~ X1 ) )
      ⋮
> PRESSp[14] = PRESS( lm( Yprime ~ X2+X3+X4 ) )
> PRESSp[15] = PRESS( fmCH09TA01.lm )
> plot( PRESSp ~ CH09TA01.r2$size ,
         ylab=expression(PRESS[p]), xlab='p' )
> minPRESSp = by( data=PRESSp,
   INDICES=factor(CH09TA01.r2$size), FUN=min )
> lines(minPRESSp ~ p )
```

Surgical Unit Data (CH09TA01) (cont'd)

$PRESS_p$ plot (cf. Fig. 9.4f): best is again X_1, X_2, X_3



“Best” Subset Selection

- To select a subset of $p–1 \geq 1$ predictor variables for further study, “best” subset algorithms perform automated searches among all possible MLR models under some optimality criterion.
- The automation seems intensive, but clever ‘branch-and-bound’ algorithms exist to speed the calculations.
- And let’s face it: the computer won’t care...

"Best" Subset Selection (cont'd)

- To perform best subset selection, select some optimality criterion, such as max.-R_p^2 or min.-C_p,
- Ask the computer to find the best 5 (say) possible subsets under that measure.
- The analyst can then further study the given subset(s) to determine an appropriate final model.
- Never, never, never, cede final decision-making to the computer!

Surgical Unit Data (CH09TA01) (cont'd)

Now, include all P–1 = 8 X_k-variables for subset selection. Can use `leaps()` function with `nbest=` option. (Apply C_p as optimality criterion.)

```
> library( leaps )
> Xmtx = cbind(X1,X2,X3,X4,X5,X6,X7,X8)
> subCH09TA01.cp = leaps( x=Xmtx, y=Yprime,
                          nbest=5, method='Cp' )
```

`nbest=5` produces 5 best (smallest C_p) X-variable subsets for each p–1 = 1,2,...,8.

Surgical Unit Data (CH09TA01) (cont'd)

R code for C_p plot:

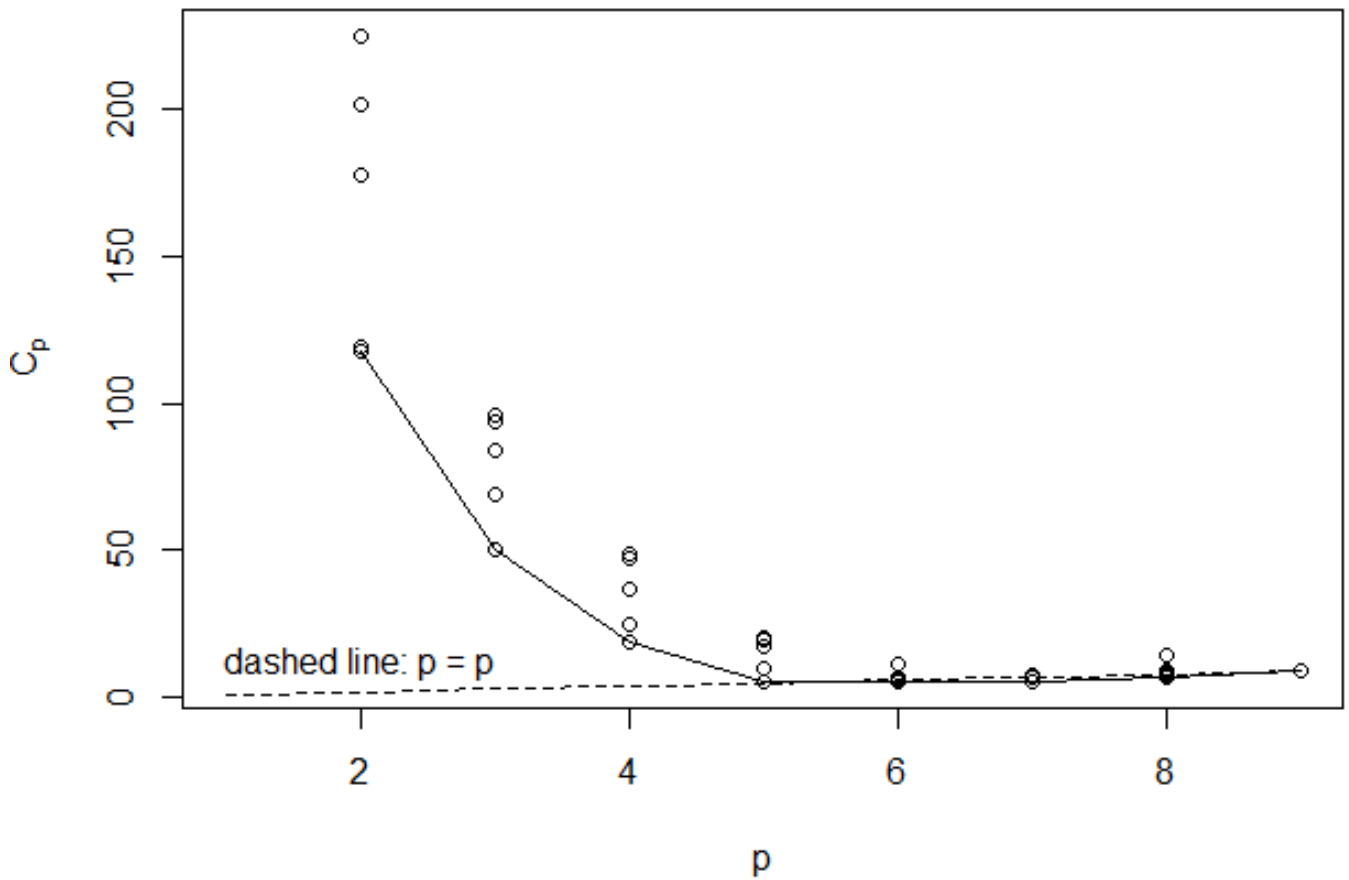
```
> plot( subCH09TA01.cp$Cp ~
          subCH09TA01.cp$size ,
             ylab=expression(C[p]),
                              xlab='p' )

> subCp = by( data=subCH09TA01.cp$Cp,
          INDICES=factor(subCH09TA01.cp$size),
                                   FUN=min )

> lines( subCp ~ seq(2,9) )
> curve(0 + 1*x, lty=2, add=T)   #p=p line
```

Surgical Unit Data (CH09TA01) (cont'd)

C_p plot with all 8 X_k-variables and (just) 5 best models at each p (cf. Fig. 9.5c):



Surgical Unit Data (CH09TA01) (cont'd)

Find min-C_p:

```
> minCp = min( subCH09TA01.cp$Cp ); minCp
[1] 5.528174
```

Find corresp. p (incl. β_0):

```
> best.index = which( subCH09TA01.cp$Cp == minCp )
> subCH09TA01.cp$size[ best.index ]
[1] 6
```

Find corresp. X_k-variables:

```
> subCH09TA01.cp$which[ best.index, ]
  1     2     3     4     5     6     7     8
 TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE
```

$\Rightarrow$ **add'l study of subset (X_1,X_2,X_3,X_6,X_8) warranted.**

Stepwise Variable Selection

- Can formalize the selection procedure in a simpler, algorithmic fashion.
- There are two basic formats:
  - Forward Stepwise Selection, and
  - Backward Elimination.

Forward Stepwise Selection

<u>Step 0</u>: Start with all P–1 X_k variables.

<u>Step 1</u>: Test each SLR of H_o:$\beta_k = 0$ via

$$t_k^* = b_k/s\{b_k\}$$

(k = 1,...,P–1) and find the X_k with the max. $|t_k^*|$ (i.e., smallest 2-sided *P*-value). Select that X_k if $P_k < \alpha_e$. Call this X_{k_1}.
(If <u>no</u> $P_k < \alpha_e$, stop and select NO X variables. α_e is the α-to-enter level.)

cont'd →

Forward Stepwise Selection (cont'd)

<u>Step 2</u>: Test every possible p–1=2 variable model with X_{k_1} and (every other) X_k ($k \neq k_1$). Find all <u>partial</u> t-statistics $t_k^* = b_k/s\{b_k|b_{k_1}\}$ with partial *P*-value P_k. Select the 2nd X_k as that with the smallest partial P_k if $P_k < \alpha_e$. Call this X_{k_2}. (If <u>no</u> $P_k < \alpha_e$, stop and select only X_{k_1}.)

<u>Step 3</u>: *Check* if X_{k_1} is still signif. with X_{k_2} included. Find the partial $t^* = b_{k_1}/s\{b_{k_1}|b_{k_2}\}$ and <u>remove</u> X_{k_1} if the corresp. $P_{k_1} > \alpha_r$. (α_r is the **α-to-remove level**.)

cont'd →

Forward Stepwise Selection (cont'd)

<u>Step 4</u>: Go to Step 2 and keep "entering" X_k's until no P_k is smaller than α_e. Also include Step 3 for possible removal.

<u>Note</u>: Be sure to keep α_e and α_r fixed throughout. (Don't change in mid-stream.) Also, always have $\alpha_e < \alpha_r$ to avoid cycling.

NB: this is clearly an exploratory method. It is not designed for inferential or confirmatory science.

Forward Selection and Backward Elimination

- A special version of Forward Stepwise Selection exists where no removal step is employed. (So there is no α_r.)
  - This is called **Forward Selection**.
- Another alternative is **Backward Elimination**: start with all P–1 X-variables and cull down until no *P*-val. is above α_r.
  - A 'backward selection' variant allows for variables to re-enter.

Backward Elimination

Many analysts favor Backward Elimination:

- **It can be more stable**
- **It often produces more accurate MSE's**
- **It retains more pertinent predictors**

→ **In early forward stepwise stages, some important predictors have yet to enter into the model. This inflates the MSE, which in turn drives the entry t-statistics closer to zero.**
⇒ step-up selection can lose important predictors along the way...

Surgical Unit Data (CH09TA01) (cont'd)

<u>Example</u>: Select subsets via **Backward Elimination**.

(a) Use `step()` with min-AIC$_p$ (option `k=2`) as optimality measure:

```
> step( fm8CH09TA01.lm, direction="backward",
                                          k=2 )
```

(b) Use `fastbw()` from external *rms* package with $P > \alpha_r = 0.10$ (option `sls=0.10`) as removal criterion:

```
> library( rms )
> fm8.ols = ols( Yprime ~
                    X1+X2+X3+X4+X5+X6+X7+X8 )
> fastbw( fit=fm8.ols, rule="p",
                 type="individual", sls=.10 )
```

Surgical Unit Data (CH09TA01) (cont'd)

(a) Backward elim. using AIC_p via `step()`:

```
Start:  AIC=-160.78
Yprime ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8

       Df Sum of Sq    RSS     AIC
- X4    1   0.00126 1.9718 -162.74
- X7    1   0.03159 2.0021 -161.92
- X5    1   0.07359 2.0441 -160.80
<none>               1.9705 -160.78
- X6    1   0.08403 2.0545 -160.52
- X1    1   0.31845 2.2890 -154.69
- X8    1   0.84489 2.8154 -143.51
- X2    1   2.09285 4.0634 -123.70
- X3    1   2.98863 4.9591 -112.94
```

Output continues →

Surgical Unit Data (CH09TA01) (cont'd)

`step()` search ends with selected min-AIC model:

```
  Step:  AIC=-163.86
Yprime ~ X1 + X2 + X3 + X5 + X6 + X8
       Df Sum of Sq    RSS      AIC
<none>               2.0043 -163.858
- X5    1    0.0769 2.0812 -163.826
- X6    1    0.0975 2.1018 -163.293
- X1    1    0.6284 2.6327 -151.133
- X8    1    0.9011 2.9054 -145.810
- X2    1    2.7644 4.7688 -119.052
- X3    1    5.0752 7.0795  -97.716
```

$\Rightarrow$ add'l study warranted of subset $(X_1, X_2, X_3, X_5, X_6, X_8)$.

Surgical Unit Data (CH09TA01) (cont'd)

(b) Backward elim. using P-val. via `fastbw()` (output edited):

```
Deleted Chi-Sq d.f. P        AIC    R2
X4      0.03   1    0.865  -1.97  0.846
X7      0.74   1    0.389  -3.23  0.843
X5      1.76   1    0.185  -3.47  0.837
X6      2.21   1    0.138  -3.27  0.830

Factors in Final Model
[1] X1 X2 X3 X8
```

$\Rightarrow$ add'l study of subset (X_1,X_2,X_3,X_8) warranted.

Forward Selection and Backward Elimination in R

- One can also explore/select regression subsets among the P–1 X_k-variables using other R commands and programming.
- The R functions `add1()` and `drop1()` and/or `addterm()` and `dropterm()` allow for various sorts of manipulations of the MLR model variables.