STAT 571A — Advanced Statistical Regression Analysis

Chapter 10 NOTES

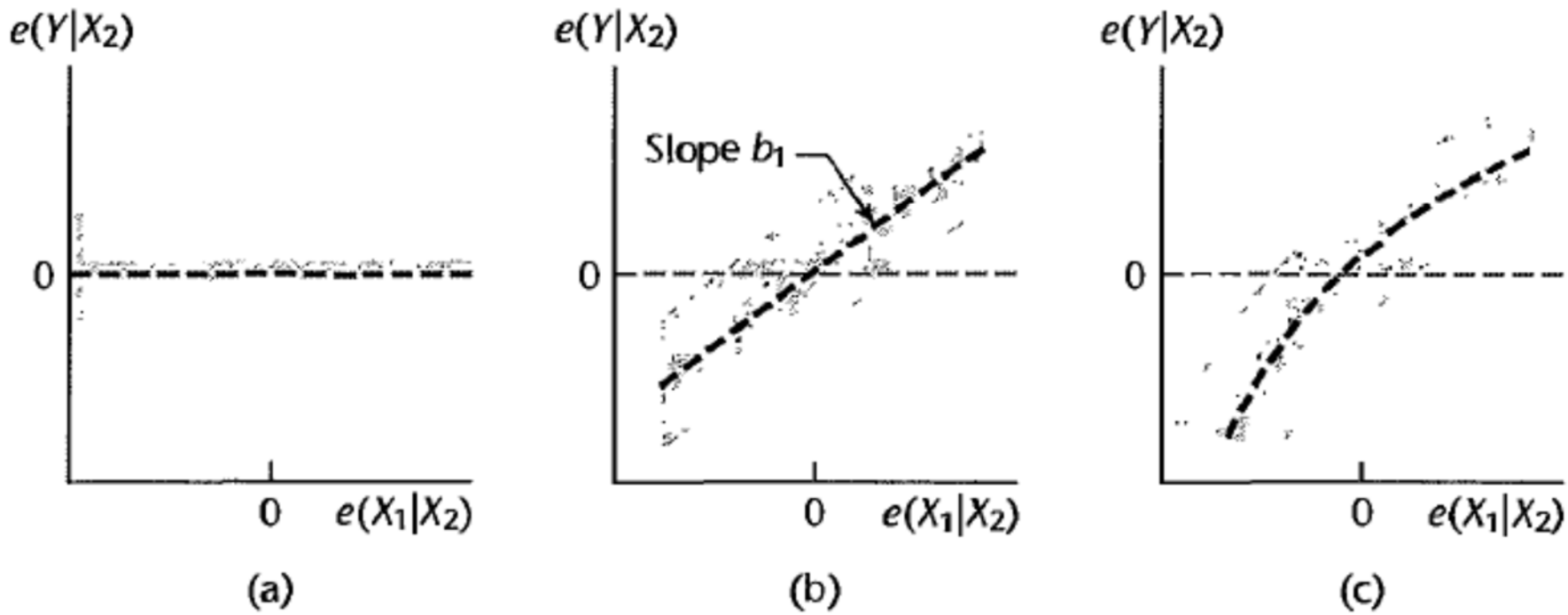
Model Building – II: Diagnostics

§10.1: Added-Variable Plots

- **Added-variable plots visualize the potential value of adding a new X_k-variable to an existing MLR model.**
- **Find the residuals from the existing fit of Y on the X_k variables (k = 2, ..., p–1); call these $e_i(Y|X_2,...,X_{p-1})$.**
- **If the new variable is X_1, regress X_1 on the X_k variables (k = 2, ..., p–1); find the residuals $e_i(X_1|X_2,...,X_{p-1})$.**
- **Plot $e_i(Y|X_2,...,X_{p-1})$ against $e_i(X_1|X_2,...,X_{p-1})$ and look for patterns.**

Versions of Add'd Var. Plots

From Fig. 10.1: (a) nothing new in X_1; (b) add'l linear term in X_1; (c) add'l curvilinear term in X_1



<u>Example</u>: Life Insur. data (CH10TA01)

- Y = Life insur. carried
  X_1 = Risk aversion score
  X_2 = ann. income
- Existing model has single predictor X_2. Build added-variable plot for new variable X_1.
- For this simple case, program this directly in R:

```
> plot( resid(lm(Y~X2)) ~ resid(lm(X1~X2)) )
> abline( lm(resid(lm(Y~X2))~resid(lm(X1~X2))))
```

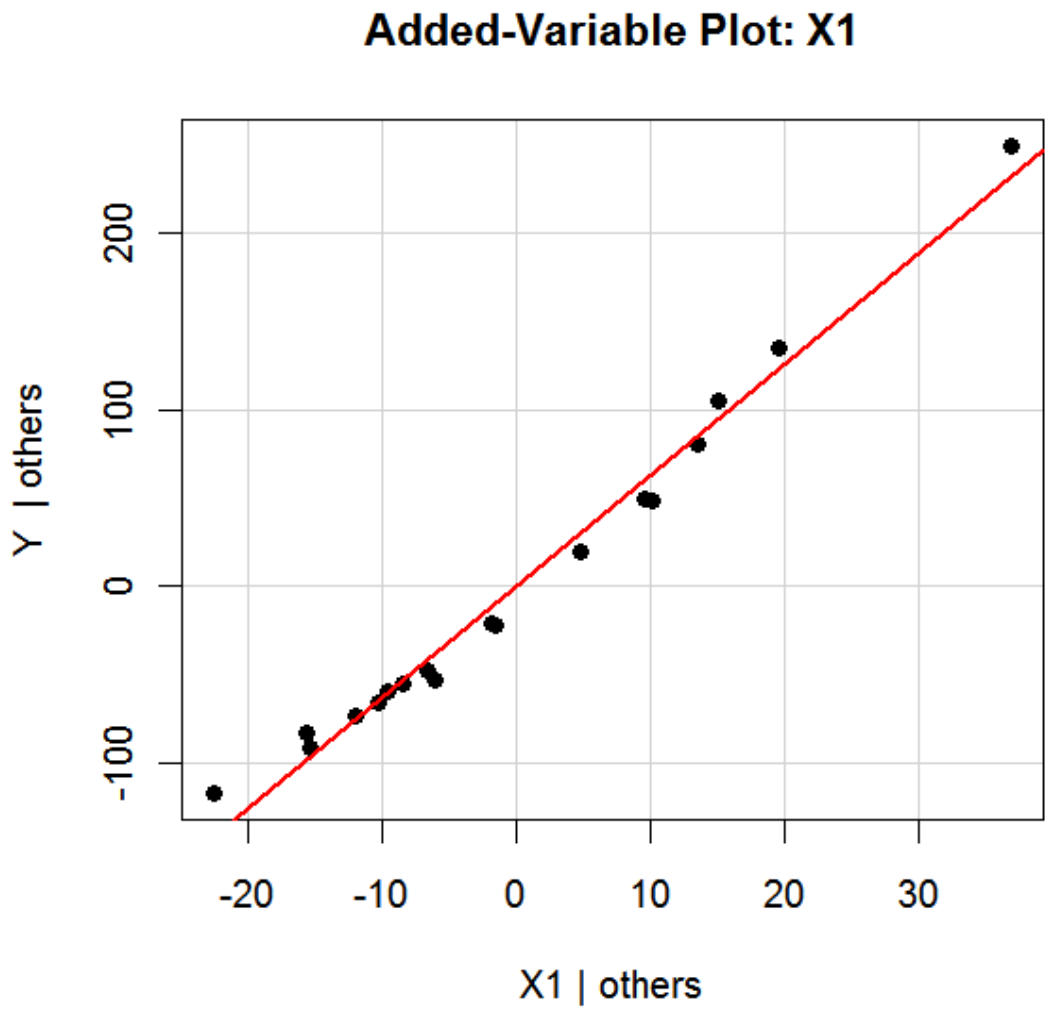
- More generally, use `avPlot()` from *car* package:

```
> library( car )
> avPlot( model=lm( Y~X1+X2 ), variable=X1 )
```

Plot follows →

Life Insur. data (CH10TA01) (cont'd)

Added-var. plot for X_1 from `avPlot()`. Clear linear pattern suggests addition of X_1 to model. (Slight curvature too, so maybe try X_1^2 too.)



Life Insur. data (CH10TA01) (cont'd)

- **Fit MLR model with p–1 = 2 predictors:**

```
> CH10TA01x1x2.lm = lm( Y ~ X2 + X1 )
> summary( CH10TA01x1x2.lm )
  Call:
  lm(formula = Y ~ X2 + X1)
 Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -205.7187    11.3927 -18.057 1.38e-11
 X2              4.7376     1.3781   3.438  0.00366
 X1              6.2880     0.2041  30.801 5.63e-15
```

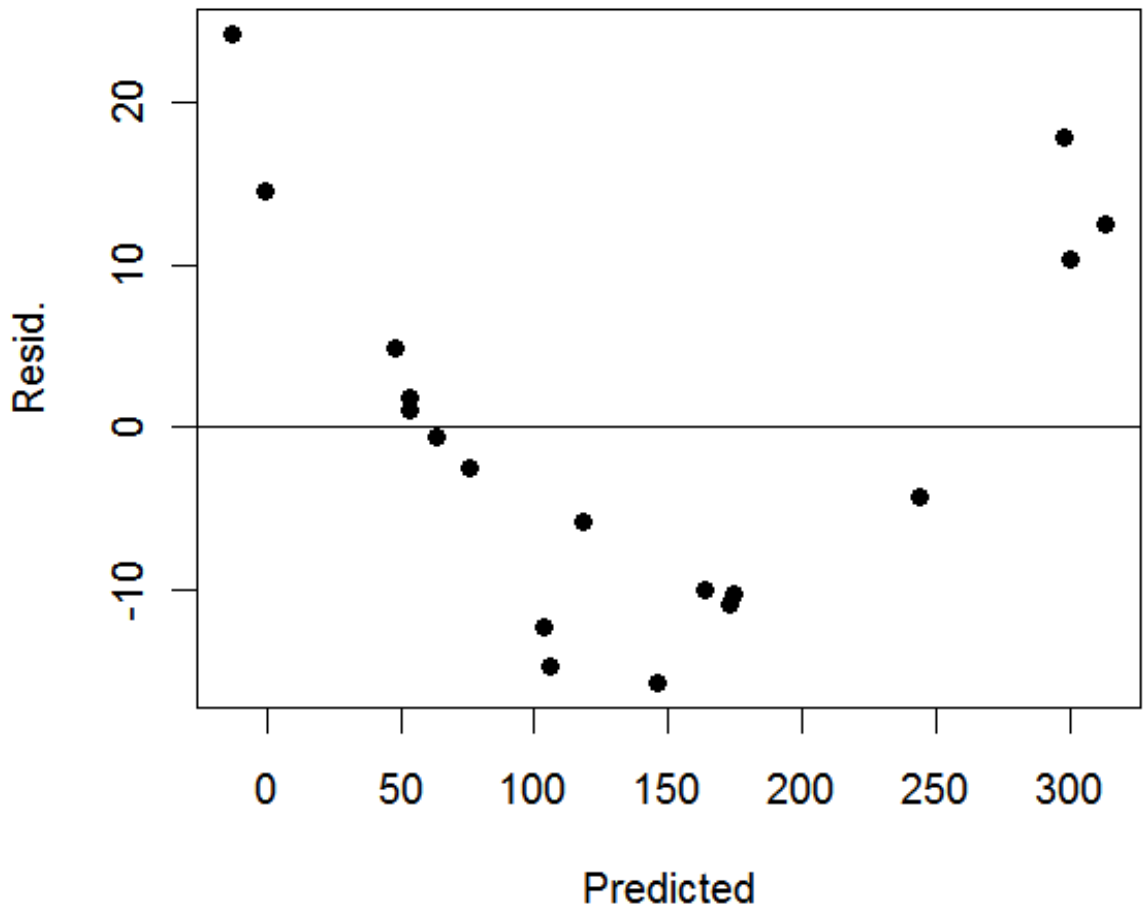
- **Residual plot:**

```
> plot( resid(CH10TA01x1x2.lm)~
                          fitted(CH10TA01x1x2.lm) )
```

Plot follows →

Life Insur. data (CH10TA01) (cont'd)

Resid. plot for X2+X1 fit. Clear, U-shaped curvature, so try adding X_1^2 to model.



Life Insur. data (CH10TA01) (cont'd)

- **Fit MLR model with p–1 = 3 predictors:**

```
> CH10TA01.lm = lm( Y ~ X2 + X1 + I(X1^2) )
> summary( CH10TA01.lm )
  Call:
  lm(formula = Y ~ X2 + X1 + I(X1^2)))
              Estimate Std. Error  t value Pr(>|t|)
(Intercept) -73.46051    6.67743  -11.001 2.83e-08
 X2           5.40039    0.25399   21.262 4.68e-12
 X1           0.79596    0.26608    2.991  0.00971
 I(X1^2)      0.05087    0.00244   20.847 6.12e-12
```

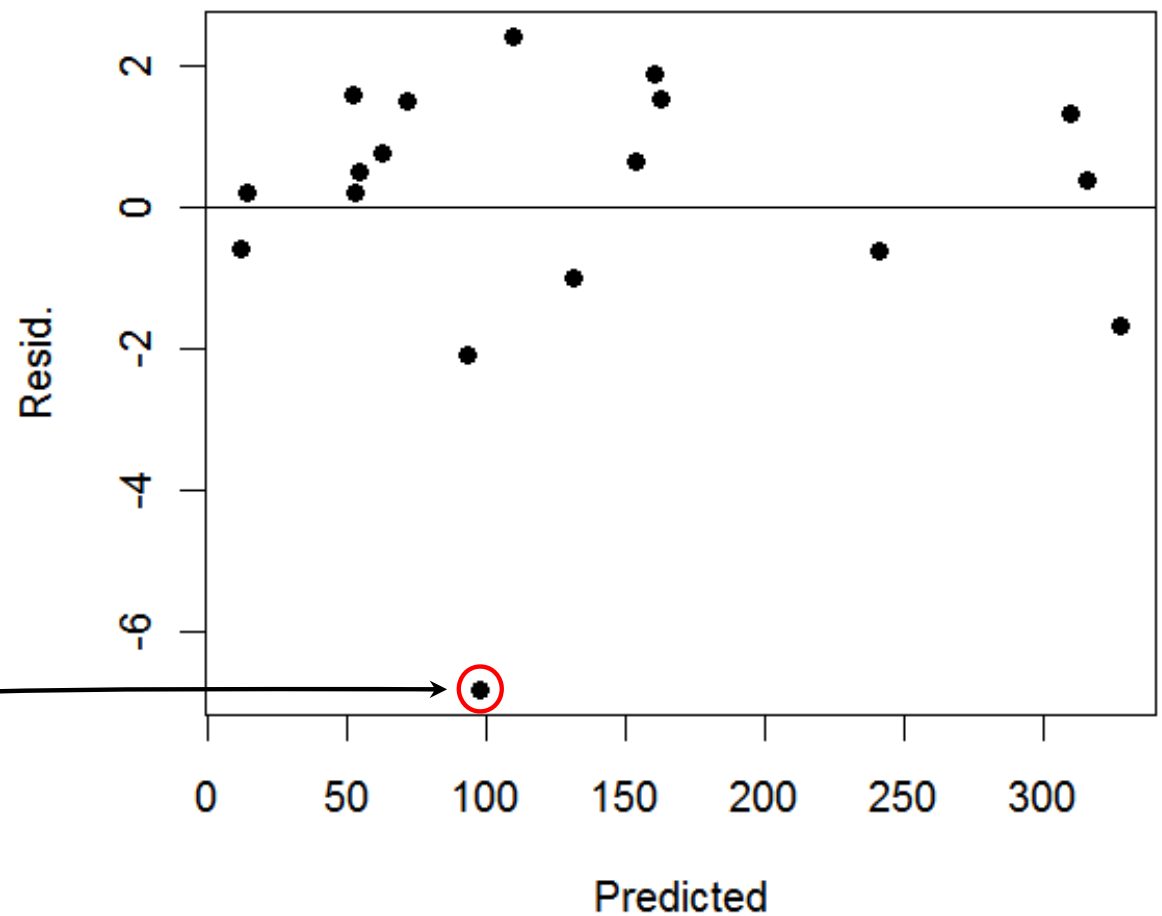
- **Residual plot:**

```
> plot( resid(CH10TA01.lm)~
                          fitted(CH10TA01.lm) )
```

Plot follows →

Life Insur. data (CH10TA01) (cont'd)

Resid. plot for full p=4 parameter model is better, but a potential 'outlier' is evident at bottom.



Partial Resid. Plot

A similar kind of diagnostic plot is known as a **Partial Residual Plot** (and is sometimes confused with an added variable plot!)

Find $e_i = Y_i - \hat{Y}_i$ for the model <u>with</u> the putative new variable X_k. Then calculate

$$p_{ik} = e_i + b_k X_k$$

and plot p_{ik} vs. X_k. (See adv. texts on regr. diagnostics.)

§10.2: Studentized Residuals

- As noted in Chapter 3, a problem with raw residuals ($e_i = Y_i - \hat{Y}_i$) is that they are scale/measurement-dependent:
  in one data set an absolute residual of $|e_i| = 8.2$ may be *less* egregious than a residual of $|e_i| = 0.7$ in another data set.

- We can stabilize residuals across data sets/model fits by standardizing them to similar scales (sort of like a *z*-score).

Studentized Residuals

- A ***Studentized Residual*** is a raw residual, e_i, divided by its standard error:

$$r_i = e_i/s\{e_i\}$$

where $s^2\{e_i\} = (1 - h_{ii})\times MSE$, with h_{ii} as the *i*th diag. element from the hat matrix $\mathbf{H}$.

- As a rule-of-thumb, the r_is exhibit homogeneous variation between about

$$-2 < r_i < 2$$

when the model is fit correctly. (But, this is a pretty *rough* rule.)

Deleted Residuals

- A *Deleted Residual* is

$$d_i = Y_i - \hat{Y}_{i(i)}$$

  where $\hat{Y}_{i(i)}$ predicts Y_i by fitting the MLR model without Y_i (cf. with the LOO operation for the PRESS statistic).
- Large values of $|d_i|$ suggest that Y_i differs greatly from the rest of the data under the proffered model.
- Can show: $d_i = e_i/(1 - h_{ii})$, so only need to fit the model once.

Studentized Deleted Residuals

- A ***Studentized Deleted Residual*** is a deleted residual, d_i, divided by its standard error:

$$t_i = d_i / s\{d_i\}$$

$$= e_i \sqrt{\frac{n-p-1}{(1-h_{ii})SSE - e_i^2}}$$

- $t_i \sim t(n-p-1)$, so we expect the *i*th point to show

$$-t(1-\alpha/2;\ n-p-1) < t_i < t(1-\alpha/2;\ n-p-1)$$

when the model is fit correctly.

Outliers

- For an MLR model, we can use the **studentized deleted residuals**, t_i, to identify observations that deviate from the model fit.
- In general: if the model fits correctly, values with $|t_i| > t(1-\alpha/(2n); n-p-1)$ indicate unusual data points. (Notice: Bonferroni correction)
- We call such values (possible) **OUTLIERS**.
- Once identified, an outlier should be studied to determine why it displays departure from the MLR model.

Studentized Residuals in R

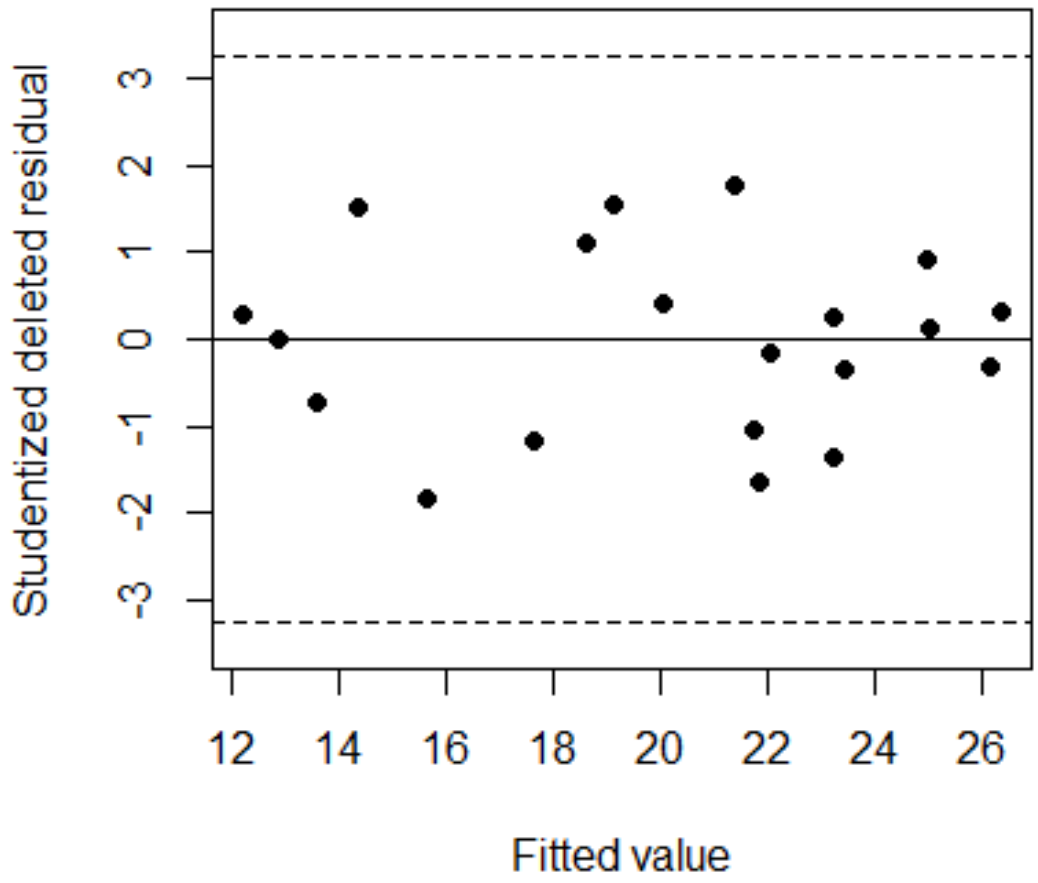
- In R, we plot the t_i's for outlier detection.
- E.g., with the Body Fat data (CH07TA01) and using only X_1 and X_2:

```
> n=length(Y); bf12.lm = lm( Y ~ X1 + X2 )
> plot( rstudent(bf12.lm) ~ fitted(bf12.lm) )
> abline( h=0 )
> tcrit = qt( 1-(.10/(2*n)), n-3-1 )
> abline( h=tcrit, lty=2 )
> abline( h=-tcrit, lty=2 )
```

- Careful: `rstudent()` gives studentized <u>deleted</u> resid's

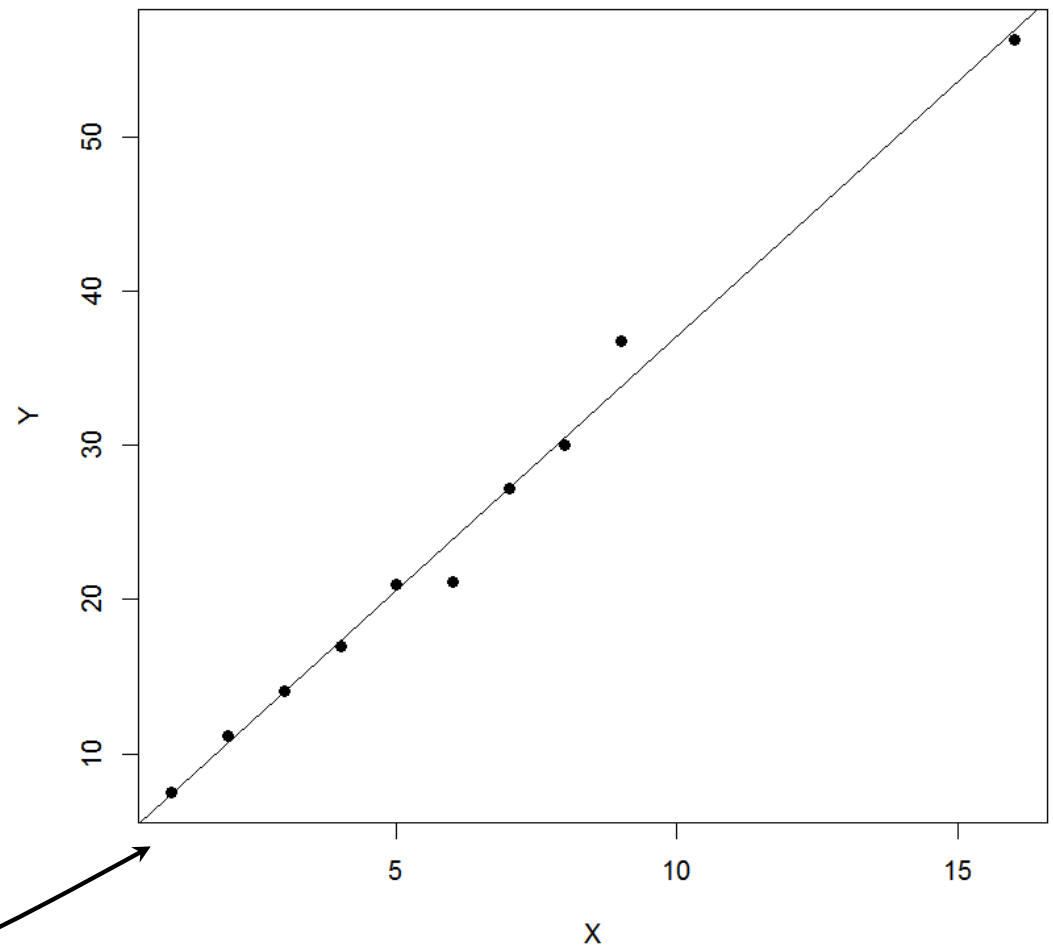
Body Fat Example (CH07TA01): Studentized Deleted Residual Plot

Possible outliers are above or below ±t crit. points. (All points are within bounds here.)



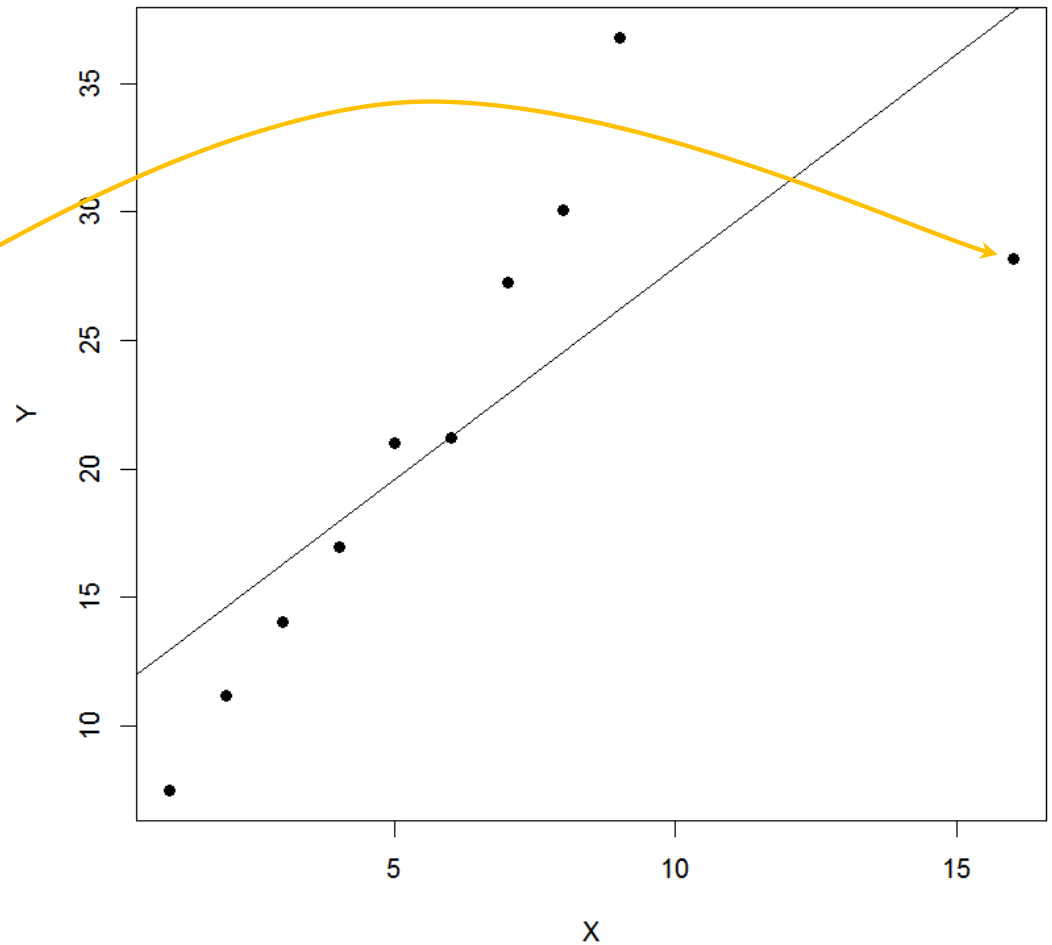
§10.3: Leverage

An X_i value can act as a sort of outlier as well, when it strongly influences the fit of the regression model. Say p=2. This plot shows a std. SLR fit with no anomalies.



Leverage (cont'd)

But when X_n drives far from the other X_i's, it can singlehandedly deteriorate the nature of the SLR line (this extends to $p > 2$ as well)



Leverage (cont')

- We say **Leverage** is the ability of a design point to strongly influence the fit of a regression model. This is usually seen as a detriment.
- Leverage occurs, e.g., when a single X_i rests far away from the bulk of the other explanatory X_i values, as illustrated in the previous 2 slides.
- Some online, interactive applets that explore leverage are available at
  - http://www.amstat.org/publications/jse/v6n3/applets/regression.html
  - http://www.rob-mcculloch.org/teachingApplets/Leverage/index.html

Leverage (cont'd)

- We quantify high leverage using the hat matrix elements.
- Notice in the studentiz'd deleted resid. that $|t_i| \uparrow$ as $h_{ii} \rightarrow 1$, while $|t_i| \downarrow$ as $h_{ii} \rightarrow 0$. (Recall that $0 \leq h_{ii} \leq 1$ and $\sum h_{ii} = p$.)
- Consequence:
  - small $h_{ii} \Rightarrow$ low resid. and fitted values close to rest of Y_i's
  - large $h_{ii} \Rightarrow$ high resid. and fitted values farther from rest of Y_i's

Leverage and Hat Elements

Indeed, since $\hat{\mathbf{Y}} = \mathbf{HY}$, h_{ii} is literally the weight of Y_i in calculating $\hat{Y}_i$. Thus large h_{ii} gives Y_i strong influence on the fit.

But wait! From equ. (10.18) we see $h_{ii} = X_i'(\mathbf{X}'\mathbf{X})^{-1}X_i$ depends only on the X's.

Thus we can check the influence of an observed (*or* unobserved) Y_i by examining just the h_{ii} value(s). Can even check for possible 'extreme' X's.

Leverage Rule-of-Thumb

A standard rule-of-thumb for informally assessing X-leverage is to indicate high leverage at X_i if

$$h_{ii} > 2\bar{h} = 2\sum h_{ii}/n = 2p/n$$

(But be careful: sometimes with small n, $\bar{h} > ½$ so $2\bar{h} = 2p/n > 1$ and <u>no</u> X_i will be marked as high leverage.)

<u>Example</u>: Body Fat data (CH07TA01)

- Restrict attention to only X_1 and X_2 and plot $\mathbf{X}$-vector values for leverage visualization:

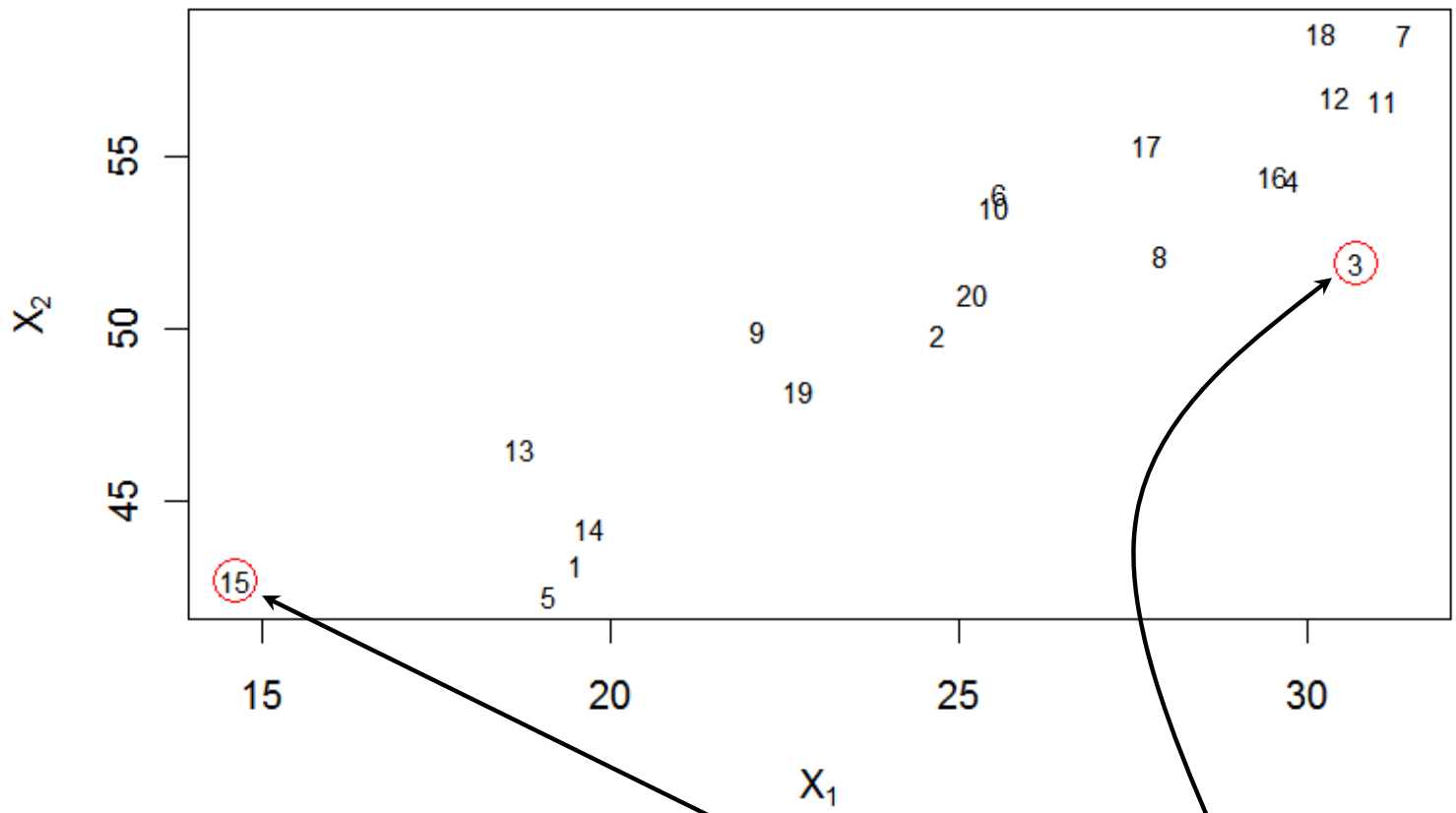
```
> plot( X2~X1, pch='' )
> text( X1, X2, labels=as.character(1:20) )
```

- Can also mark points with high leverage (i.e., $h_{ii} > 2p/n$):

```
> n = length(Y); p = 3
> hii = hatvalues( bf12.lm )
> points( X1[hii>2*p/n], X2[hii>2*p/n],
                                cex=2.5, col='red' )
```

Plot follows →

Body Fat Example (CH07TA01): X_2 vs. X_1 leverage plot



High leverage points at i = 15 and i = 3; cf. Fig. 10.7

Influence Measures: DFFITS

- To measure the influence of a single fitted value, calculate

$$(\text{DFFITS})_i = \frac{Y_i - \hat{Y}_{i(i)}}{\sqrt{h_{ii}\, MSE_{(i)}}} = t_i \sqrt{\frac{h_{ii}}{1-h_{ii}}}$$

- The DFFITS measure takes the studentiz'd deleted resid. t_i and weights it with a measure proportional to h_{ii}.
- This acts as a sort of combined measure of overall influence.

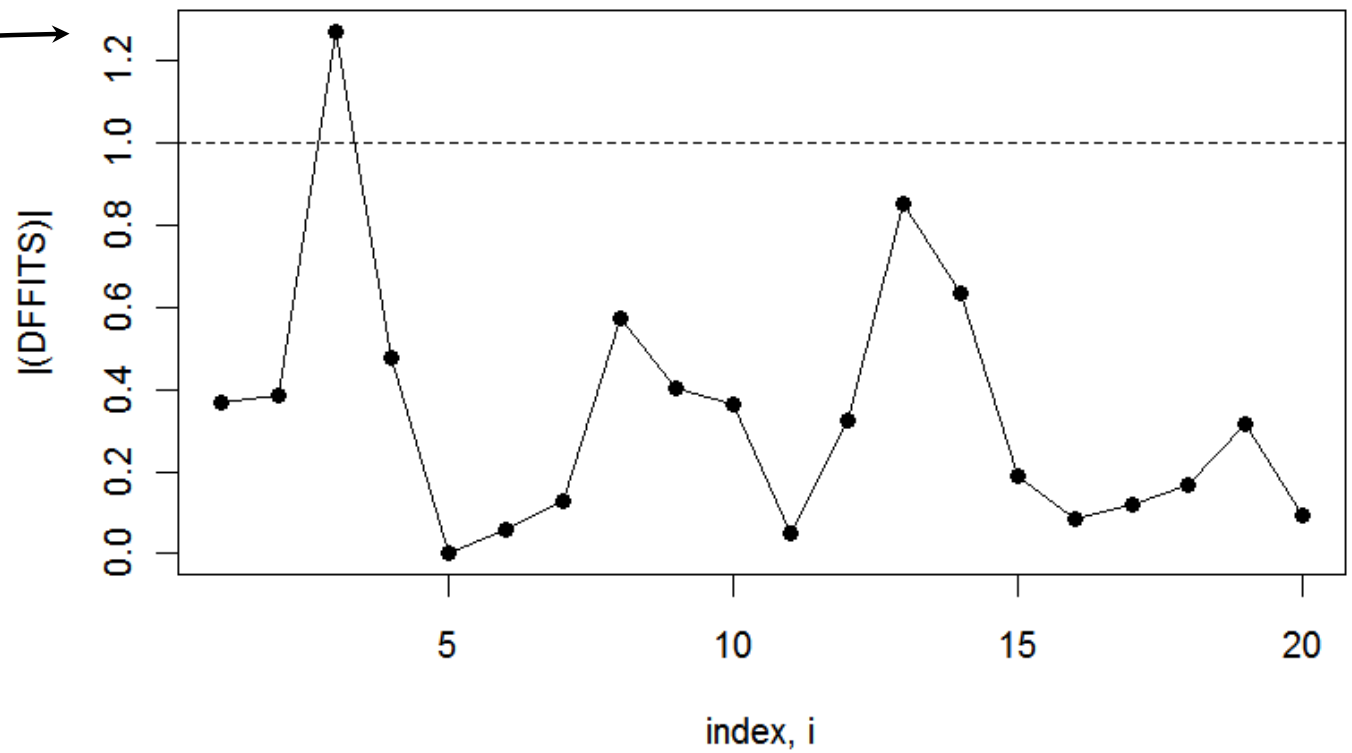
DFFITS (cont'd)

- For use in practice, view $\hat{Y}_i$ as an influential fitted value if
  - $|(\text{DFFITS})_i| > 1$ (for $n < 40$) or
  - $|(\text{DFFITS})_i| > 2\{p/n\}^{1/2}$ (for $n \geq 40$).
- In R, use
  `dffits( [lm Object here] )`.

<u>Example</u>: Body Fat data (CH07TA01)

```
> plot( abs( dffits(lm(Y~X1+X2)) ),
                                  type='o', pch=19 )
> abline( h=1, lty=2 )
```

High DFFITS point at *i* = 3



Influence Measures: Cook's Distance

What about measuring influence across the fitted values $\hat{Y}_i$? Cook's Distance is defined as

$$D_i = \frac{\sum_{j=1}^{n} (\hat{Y}_j - \hat{Y}_{j(i)})^2}{p\ MSE} = \frac{e_i^2}{p\ MSE} \frac{h_{ii}}{(1 - h_{ii})^2}$$

(Notice how the LOO approach collapses into a single calculation.)

View $\hat{Y}_i$ as a (very) influential case when

$$P[\ F(p,n-p) \leq D_i\ ] > ½.$$

(Some authors suggest > 1/10 or > 1/5...)

<u>Example</u>: Body Fat data (CH07TA01)

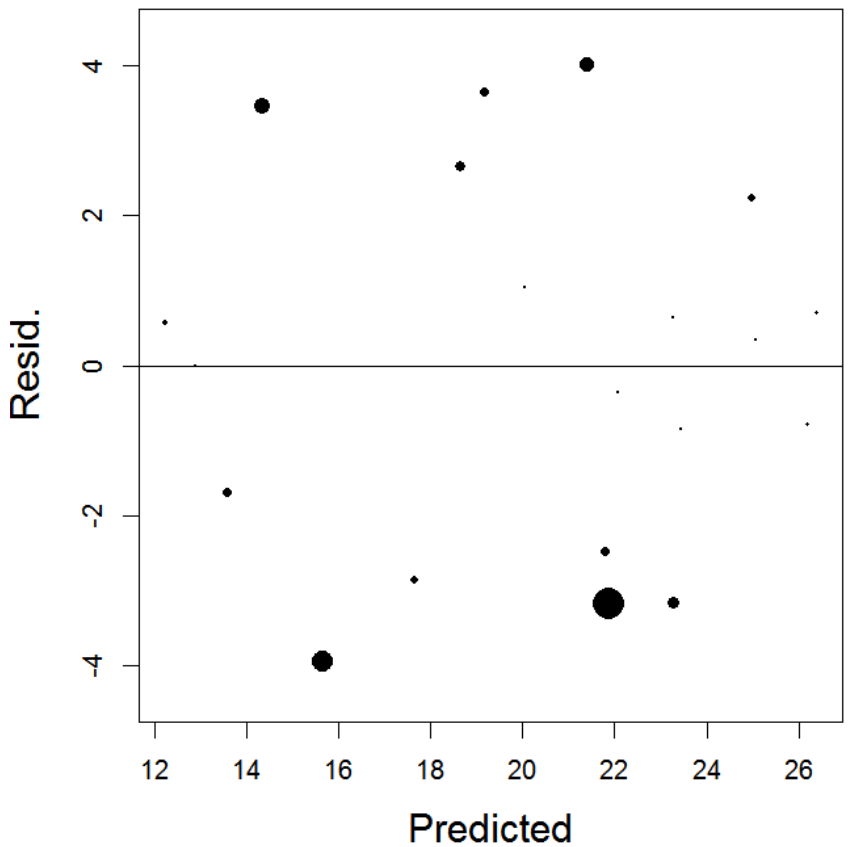
Cook's Distance plots (cf. Fig. 10.8):

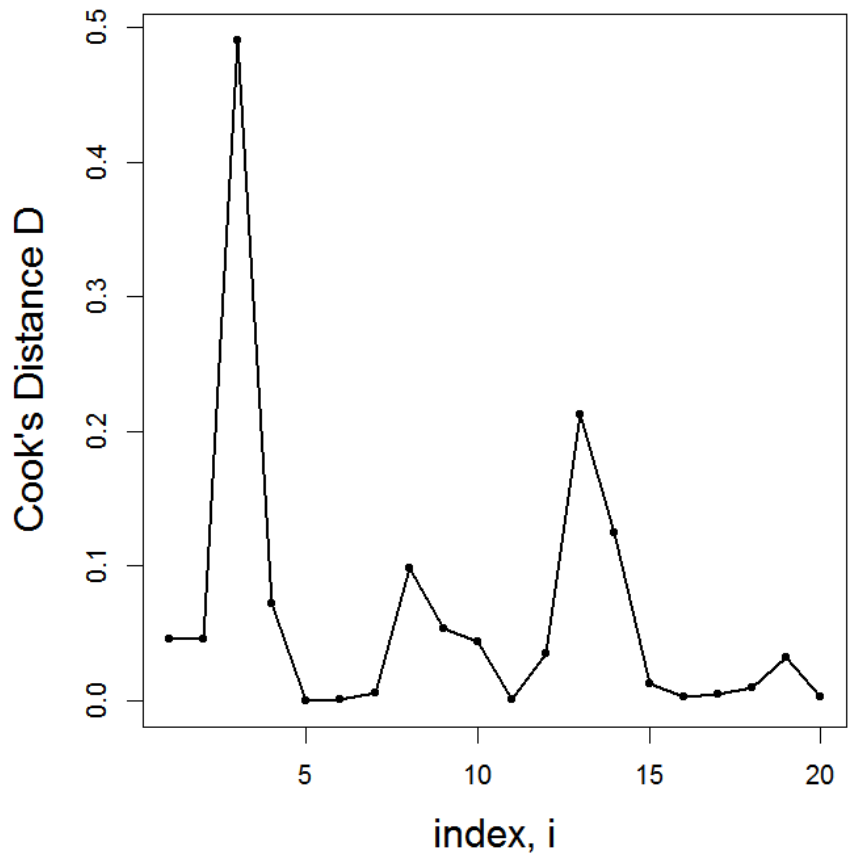
```
> par( mfrow=c(1,2) )
> ei = resid( lm(Y~X1+X2) )
> yhat = fitted(lm(Y~X1+X2))

> ######## Proportional Infl. Plot:
> radius = sqrt( cooks.distance(lm(Y~X1+X2))/pi )
> plot( ei ~ yhat, pch=''); abline( h=0 )
> symbols( yhat, ei, circles=radius, inches=.15,
                    bg='black', fg='white', add=T )

> ######## Index Infl. Plot:
> plot( cooks.distance(lm(Y~X1+X2)), type='o',
                                          pch=19 )
```

Plots follow ⟶

Body Fat data (CH07TA01) (cont'd)

Cook's Distance influence plots (cf. Fig. 10.8):





Body Fat data (CH07TA01) (cont'd)

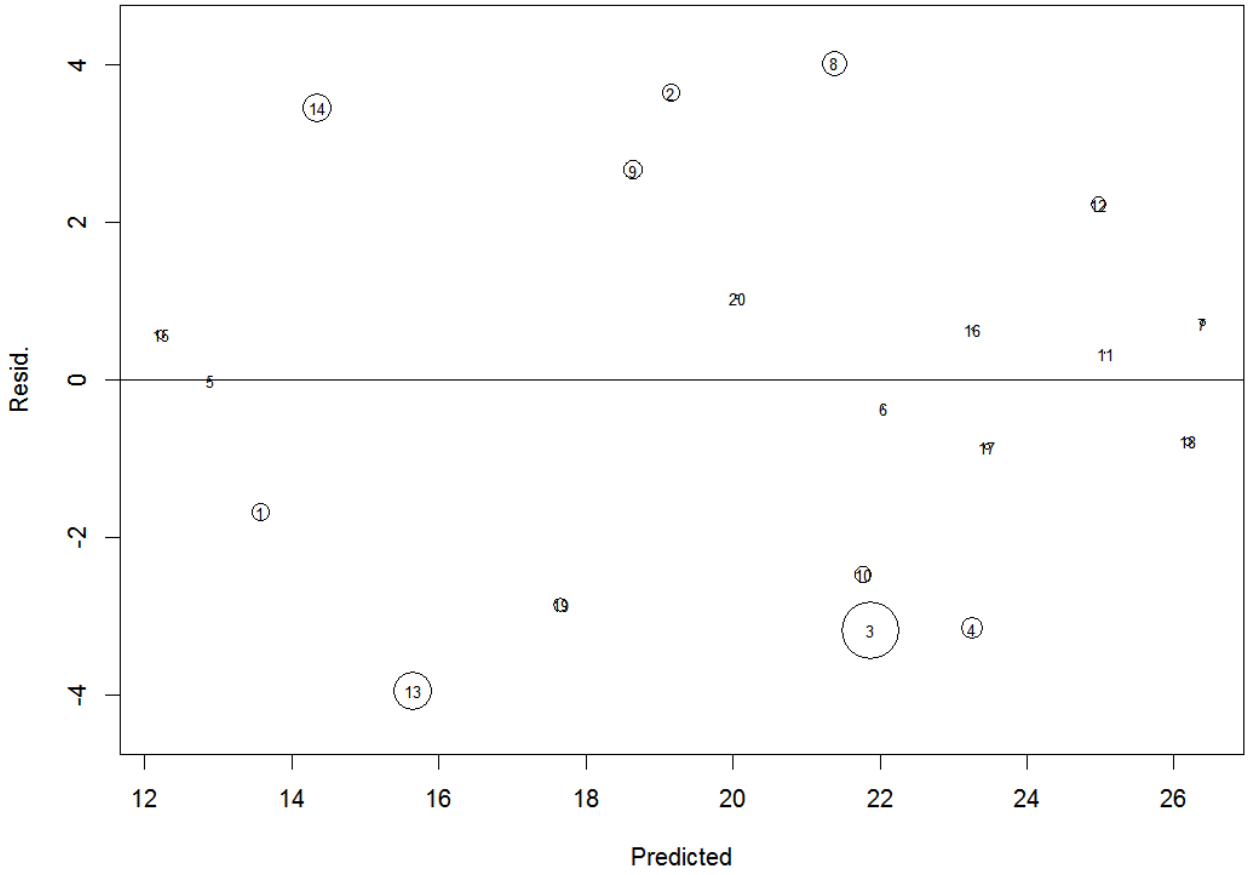
Or, try marking the Proportional Influence plot with the index, *i*:

```
> ei = resid( lm(Y~X1+X2) )
> yhat = fitted(lm(Y~X1+X2))
> Di = cooks.distance( lm(Y~X1+X2) )
> radius = sqrt( Di/pi )
> plot( ei ~ yhat, pch=''); abline( h=0 )
> symbols( yhat, ei, circles=radius,
                          inches=.2, add=T )
> text( yhat, ei,
       labels=as.character(1:20), cex=.7 )
```

Plot follows ⟶

Body Fat data (CH07TA01) (cont'd)

Cook's Distance proportional influence plot (index-labeled):



Influence Measures: DFBETAS

Another influence measure (and there are LOTS of 'em...) quantifies the influence of Y_i on the regression coefficients, b_k.

Define

$$(\text{DFBETAS})_{k(i)} = \frac{b_k - b_{k(i)}}{\sqrt{c_{kk}\, \text{MSE}_{(i)}}}$$

where

- $b_{k(i)}$ is the *k*th regr. coeff., and $\text{MSE}_{(i)}$ is the MLR MSE, with Y_i removed from the data;
- c_{kk} is the *k*th diag. element of $(X'X)^{-1}$

(Whew...)

DFBETAS (cont'd)

A large value of $(DFBETAS)_{k(i)}$, say

- $|(DFBETAS)_{k(i)}| > 1$ (for $n < 40$) or
- $|(DFBETAS)_{k(i)}| > 2/\sqrt{n}$ (for $n \geq 40$)

indicates large impact of Y_i on that particular b_k.

In R, the `influence.measures()` function provides all the influence measures described above.

<u>Example</u>: Body Fat data (CH07TA01)

All influence measures for p–1 = 2 predictors:

```
> influence.measures( bf12.lm )
    Influence measures of
         lm(formula = Y ~ X1 + X2):
      dfb.1_    dfb.X1    dfb.X2     dffit cov.r   cook.d    hat inf
1  -3.05e-01 -1.31e-01  2.32e-01 -3.66e-01 1.361 4.60e-02 0.2010
2   1.73e-01  1.15e-01 -1.43e-01  3.84e-01 0.844 4.55e-02 0.0589
3  -8.47e-01 -1.18e+00  1.07e+00 -1.27e+00 1.189 4.90e-01 0.3719  *
4  -1.02e-01 -2.94e-01  1.96e-01 -4.76e-01 0.977 7.22e-02 0.1109
5  -6.37e-05 -3.05e-05  5.02e-05 -7.29e-05 1.595 1.88e-09 0.2480  *
6   3.97e-02  4.01e-02 -4.43e-02 -5.67e-02 1.371 1.14e-03 0.1286
7  -7.75e-02 -1.56e-02  5.43e-02  1.28e-01 1.397 5.76e-03 0.1555
8   2.61e-01  3.91e-01 -3.32e-01  5.75e-01 0.780 9.79e-02 0.0963
9  -1.51e-01 -2.95e-01  2.47e-01  4.02e-01 1.081 5.31e-02 0.1146
10  2.38e-01  2.45e-01 -2.69e-01 -3.64e-01 1.110 4.40e-02 0.1102
```

(Asterisk indicates high influence on any measure)

Output continues ⟶

Body Fat data (CH07TA01) (cont'd)

`influence.measures()` output (cont'd):

```
Influence measures of
         lm(formula = Y ~ X1 + X2) :

      dfb.1_    dfb.X1    dfb.X2     dffit cov.r   cook.d    hat inf
11 -9.02e-03  1.71e-02 -2.48e-03  5.05e-02 1.359 9.04e-04 0.1203
12 -1.30e-01  2.25e-02  7.00e-02  3.23e-01 1.152 3.52e-02 0.1093
13  1.19e-01  5.92e-01 -3.89e-01 -8.51e-01 0.827 2.12e-01 0.1784
14  4.52e-01  1.13e-01 -2.98e-01  6.36e-01 0.937 1.25e-01 0.1480
15 -3.00e-03 -1.25e-01  6.88e-02  1.89e-01 1.775 1.26e-02 0.3332  *
16  9.31e-03  4.31e-02 -2.51e-02  8.38e-02 1.309 2.47e-03 0.0953
17  7.95e-02  5.50e-02 -7.61e-02 -1.18e-01 1.312 4.93e-03 0.1056
18  1.32e-01  7.53e-02 -1.16e-01 -1.66e-01 1.462 9.64e-03 0.1968
19 -1.30e-01 -4.07e-03  6.44e-02 -3.15e-01 1.002 3.24e-02 0.0670
20  1.02e-02  2.29e-03 -3.31e-03  9.40e-02 1.224 3.10e-03 0.0501
```

(Asterisk indicates high influence on any measure)

Body Fat data (CH07TA01) (cont'd)

Can see which measures actually exhibit influence via the `$is.inf` attribute (but be careful: the `hat` column doesn't always work as expected; here it misses *i*=3 and *i*=15):

```
> influence.measures( bf12.lm )$is.inf

Influence measures of  lm(formula = Y ~ X1 + X2):
    dfb.1_ dfb.X1 dfb.X2 dffit  cov.r  cook.d  hat
1   FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
2   FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
3   FALSE   TRUE   TRUE   TRUE  FALSE  FALSE  FALSE
4   FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
5   FALSE  FALSE  FALSE  FALSE   TRUE  FALSE  FALSE
6   FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
7   FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
8   FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
9   FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
10  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
```

(`TRUE` indicates high influence on that measure) Output continues ⟶

Body Fat data (CH07TA01) (cont'd)

`influence.measures()$is.inf` (cont'd):

```
Influence measures of  lm(formula = Y ~ X1 + X2):
        dfb.1_ dfb.X1 dfb.X2 dffit  cov.r  cook.d  hat
    11  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
    12  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
    13  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
    14  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
    15  FALSE  FALSE  FALSE  FALSE   TRUE  FALSE  FALSE
    16  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
    17  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
    18  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
    19  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
    20  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
```

(`TRUE` indicates high influence on that measure)

Multicollinearity Diagnostics

- In Ch. 7 we saw that multicollinearity had negative effects on the MLR fit (see list on pp. 406-407).

- Can we diagnose multicollinearity? Yes:
  - informally: look for big changes in b_k when re-ordered in the MLR sequential fit;
  - look for insignif. b_k's when we expect them to be important scientifically;
  - study corrl'n matrix of X_k's for values near ±1.

Variance Inflation

- A more formal measure for assessing multicollinearity relates to how $\sigma^2\{\mathbf{b}\}$ is affected.
- Recall that $\sigma^2\{\mathbf{b}\} = (\mathbf{X}'\mathbf{X})^{-1}\sigma^2$.
  It can be shown that if R_k^2 is the R^2 from regressing X_k on the other p–2 X's, then
  $$\sigma^2\{b_k\} \approx \varphi_k/(1-R_k^2)$$
  for some positive const. φ_k.
- This quantifies potential <u>inflation</u> of the variance of b_k.

VIFs

- Thus, we can build a *factor* to quantify the potential variance inflation:

$$VIF_k = 1/(1-R_k^2)$$

  measures how much variance inflation occurs due to high multicollinearity in X_k (with the other X's).

- As $VIF_k \rightarrow 1$, inflation diminishes.

- But as $VIF_k \rightarrow \infty$, inflation increases detrimentally and can incite multicollinearity.

VIF Rule-of-Thumb

- A VIF_k is felt to be extreme if it exceeds 10.
- In fact, for diagnostic use a set of predictor variables is felt to possess high multicollinearity if $\max\{VIF_1,\ldots,VIF_{p-1}\} > 10$.
- Also check their mean: if $\overline{VIF}$ is much larger than 1, problems may persist. (Guidelines vary, but a $\overline{VIF}$ above about 6 or 7 is considered severe.)

Example: Body Fat data (CH07TA01)

Multicollinearity diagnostics:

Variance Inflation Factors (VIFs) for full p–1 = 3 predictor model:

```
> CH07TA01.lm = lm( Y ~  X1 + X2 + X3 )
> cor( cbind(X1,X2,X3) )
            X1        X2        X3
   X1 1.0000000 0.9238425 0.4577772
   X2 0.9238425 1.0000000 0.0846675
   X3 0.4577772 0.0846675 1.0000000
> library ( car )
> vif( CH07TA01.lm )
         X1       X2       X3
   708.8429 564.3434 104.6060
> mean( vif(CH07TA01.lm) )
   [1] 459.2641
```

Summary of Regression Diagnostics: Impact and Influence

- **To detect influence of Y_is:**
  - **Possible <u>outliers</u> are determined by Studentized deleted residuals (Sec. 10.2)**
  - **Influence on estimated b_j coefficients is determined by DFBETAS (Sec. 10.4)**
  - ***Joint* (X and Y) influence on model fit is determined by DFFITS (Sec. 10.4) – also see next slide**
  - **Influence on fitted values is determined by Cook's D_i (Sec. 10.4)**

Summary of Regression Diagnostics: Impact and Influence

- **To detect influence of X_{ij}s:**
  - **<u>Leverage</u> on the estimated regression line is determined by hat matrix diagonals, h_{ii} (Sec. 10.3)**
  - **Influence on estimated b_j coefficients from <u>multicollinearity</u> is determined by VIF_j (Sec. 10.5)**
  - ***Joint* (X and Y) influence on model fit is determined by DFFITS (Sec. 10.4) – also see previous slide**