Regression Diagnostics

Paul E. Johnson[1] [2]

[1]Department of Political Science

[2]Center for Research Methods and Data Analysis, University of Kansas

2015

Outline

Outline

Problem

- Recall the lecture about diagnostic plots?
- Remember some plots used terms “leverage” and “Cook’s Distance”?
- I said we’d come to a day when I had to try to explain that?
    - The day of reckoning has come.

Outline

Recall the Public Spending Example Data Set

To get the publicspending dataset, download publicspending.txt in a Web browser, or run

```
dat <- read.table("http://pj.freefaculty.org/guides/stat/DataSets/
    PublicSpending/publicspending.txt", header = TRUE)
```

```
summarize(dat)
```

```
$numerics
        ECAB      EX     GROW    MET    OLD    WEST  YOUNG
0%     57.40  183.00   -7.400   0.00  5.400  0.0000 24.000
25%    85.40  253.50    6.975  24.10  7.950  0.0000 26.400
50%    95.30  285.50   14.050  46.15  9.450  0.5000 28.000
75%   105.10  324.00   22.670  69.97 10.420  1.0000 29.630
100%  205.00  454.00   77.800  86.50 11.900  1.0000 32.900
mean   96.75  286.60   18.730  46.17  9.212  0.5000 28.110
sd     22.25   58.79   18.870  26.94  1.639  0.5053  2.149
var   495.20 3457.00  356.300 725.70  2.687  0.2553  4.616
NA's    0.00    0.00    0.000   0.00  0.000  0.0000  0.000
N      48.00   48.00   48.000  48.00 48.000 48.0000 48.000

$factors
          STATE
```

Recall the Public Spending Example Data Set ...

```
AL              :  1.000
AR              :  1.000
AZ              :  1.000
CA              :  1.000
(All Others)    :44.000
NA's            :  0.000
entropy         :  5.585
normedEntropy:  1.000
N               :48.000
```

This time, I decided to create MET squared before running the model, but you will recall there are at least 4 different ways to run this regression.

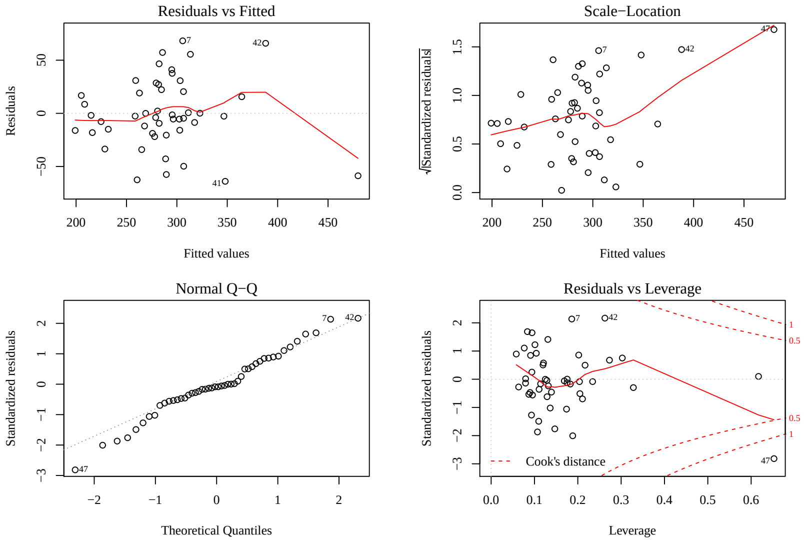
```
dat$METSQ <- dat$MET*dat$MET
EXfull2 <- lm(EX ~ ECAB + MET + METSQ + GROW + YOUNG + OLD + WEST,
     data=dat)
summary(EXfull2)
```

Recall the Public Spending Example Data Set ...

```
Call:
lm(formula = EX ~ ECAB + MET + METSQ + GROW + YOUNG + OLD + WEST,
    data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-63.974 -16.620  -2.647  20.898  68.234

Coefficients:
              Estimate  Std. Error t value Pr(>|t|)
(Intercept) 119.118461 280.911921   0.424 0.673807
ECAB          1.395420   0.382255   3.650 0.000749 ***
MET          -3.042142   0.758040  -4.013 0.000256 ***
METSQ         0.030914   0.008958   3.451 0.001332 **
GROW          0.695336   0.379504   1.832 0.074371 .
YOUNG         0.607602   6.975082   0.087 0.931018
OLD           4.120784   6.574827   0.627 0.534383
WEST         34.073079  12.245464   2.783 0.008192 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 35.41 on 40 degrees of freedom
Multiple R^2:  0.6913, Adjusted R^2:  0.6373
F-statistic:  12.8 on 7 and 40 DF,  p-value: 1.717e-08
```

Recall the Public Spending Example Data Set ...

Recall the Public Spending Example Data Set



influence.measures() provides one line per case in data

```
EXfull2infl <- influence.measures(EXfull2)
print(EXfull2infl)
```

```
Influence measures of
   lm(formula = EX ~ ECAB + MET + METSQ + GROW + YOUNG + OLD + WEST,      data = dat) :

      dfb.1_   dfb.ECAB    dfb.MET   dfb.METS   dfb.GROW   dfb.YOUN
1   0.033614 -0.022425   3.24e-03  -6.35e-03  -1.62e-02  -0.033207
2  -0.020224  0.009687   5.87e-03   1.17e-02  -1.33e-02   0.022675
3  -0.108585  0.061042  -2.81e-01   2.31e-01   1.08e-01   0.115471
4   0.025615 -0.010965   3.09e-02  -4.89e-02   8.70e-03  -0.025093
5  -0.039827  0.083028   1.45e-01  -2.02e-01   1.44e-01   0.029069
6   0.000317  0.001048  -5.45e-04   7.85e-04  -4.75e-04  -0.000477
7   0.495158 -0.327230  -3.87e-01   3.95e-01  -4.08e-01  -0.491621
8  -0.282785  0.105075   8.03e-02  -4.54e-02   1.24e-01   0.287801
9  -0.026774  0.015032   2.62e-02  -5.00e-02   8.23e-02   0.025474
10  0.169823 -0.028564   7.67e-02  -1.31e-01   6.67e-02  -0.171571
11 -0.000078  0.000091   4.37e-05  -5.17e-05   1.09e-05   0.000061
12 -0.017406  0.014434  -8.41e-03   1.25e-02   3.20e-03   0.015895
13 -0.124846  0.124751   1.17e-02   5.18e-02  -2.43e-02   0.135831
14  0.023257 -0.033842  -1.58e-02   5.30e-03   2.77e-03  -0.022143
15  0.029090 -0.065832  -6.53e-02   6.98e-02  -5.72e-03  -0.022790
16 -0.002857 -0.045190   5.38e-03  -1.85e-02   5.57e-02   0.006605
17 -0.083721  0.141726   1.60e-01  -1.68e-01   2.94e-02   0.065519
18  0.036471 -0.041965  -2.67e-02   2.05e-02   3.05e-02  -0.039134
19  0.030433 -0.017238  -5.85e-02   5.73e-02   2.38e-02  -0.035057
20  0.030090 -0.028983   3.63e-02  -4.21e-02  -1.83e-02  -0.026316
21 -0.118448  0.054141  -3.84e-02   1.05e-01   6.57e-02   0.080258
22  0.075539 -0.022198  -4.17e-02   2.03e-02  -3.34e-03  -0.103346
23  0.058924 -0.061339   7.13e-02  -8.06e-02  -2.87e-02  -0.043911
24 -0.007277 -0.041392  -5.50e-03   3.33e-05   1.12e-01   0.003200
```

influence.measures() provides one line per case in data ...

```
25  -0.044660  0.072497  -7.19e-02  5.41e-02  7.14e-02  0.033730
26   0.089479  0.118208   2.39e-01 -2.66e-01  1.66e-02 -0.137563
27  -0.358113  0.221774   2.42e-01 -9.86e-02  4.08e-02  0.332909
28   0.003968 -0.004455  -6.01e-03  5.57e-03  6.55e-04 -0.002415
29  -0.224326  0.179601  -2.88e-01  2.80e-01  1.64e-01  0.289007
30  -0.157029  0.181030  -2.10e-01  1.95e-01 -1.21e-01  0.176432
31   0.000623  0.000917   1.40e-02 -9.22e-03  2.41e-03 -0.000312
32  -0.051931  0.012917  -1.05e-01  1.23e-01  4.26e-02  0.074303
33  -0.016213  0.003556  -2.70e-02  3.35e-02  5.70e-03  0.021540
34   0.000775 -0.000616  -8.68e-04  4.79e-04 -1.52e-05 -0.000655
35   0.006423  0.188257   5.94e-02 -7.03e-02  1.42e-01 -0.013179
36   0.087929 -0.083146   9.76e-02 -1.05e-01 -5.93e-02 -0.098041
37  -0.066354  0.052002  -1.51e-01  9.00e-02  1.25e-01  0.053882
38   0.184503 -0.140312   6.37e-02 -1.26e-01  3.28e-02 -0.152311
39   0.050217  0.041654   1.41e-02 -4.00e-02 -1.53e-01 -0.056807
40   0.001382  0.038629   3.81e-02 -6.53e-02 -1.44e-02 -0.008998
41   0.252860 -0.184320   7.63e-01 -6.93e-01 -9.92e-02 -0.311904
42   0.290308  0.360149  -7.12e-01  4.09e-01 -3.66e-01 -0.253964
43  -0.025624  0.017844   1.91e-02 -4.69e-03 -9.48e-03  0.025327
44  -0.336514  0.176489   5.03e-02  1.13e-01  3.02e-02  0.364581
45  -0.028611 -0.003324   1.27e-01 -6.34e-02 -4.25e-02  0.015223
46  -0.062138  0.045225   3.00e-01 -2.43e-01 -3.50e-02  0.024796
47   0.861857 -2.918265  -5.85e-01  6.66e-01 -6.48e-01 -0.637764
48  -0.010704 -0.055297  -1.11e-01  1.53e-01  7.18e-02  0.013846
       dfb.OLD  dfb.WEST     dffit  cov.r    cook.d    hat  inf
1   -3.63e-02  0.031058  -0.045912  1.597  2.70e-04  0.2342
2    3.91e-03  0.021366  -0.056073  1.480  4.03e-04  0.1753
3    1.76e-01 -0.215227   0.412026  1.536  2.15e-02  0.2730
4   -3.28e-02  0.014115  -0.079188  1.490  8.03e-04  0.1828
5    1.69e-02 -0.020465  -0.358119  1.407  1.62e-02  0.2108
6   -1.59e-04 -0.001485   0.004951  1.330  3.14e-06  0.0796
7   -3.81e-01  0.112238   1.071230  0.571  1.30e-01  0.1860
8    2.51e-01  0.026803  -0.531560  0.871  3.42e-02  0.1098
```

influence.measures() provides one line per case in data ...

```
9    2.01e-02  0.015488 -0.180159 1.271 4.13e-03 0.0954
10  -1.84e-01 -0.122726  0.413144 1.004 2.11e-02 0.1011
11   9.55e-05  0.000013 -0.000198 1.399 5.01e-09 0.1250
12   1.87e-02  0.002122 -0.026210 1.472 8.81e-05 0.1689
13   6.12e-02 -0.128755  0.266670 1.166 8.95e-03 0.0910
14  -1.63e-02  0.038732 -0.070823 1.288 6.42e-04 0.0635
15  -2.38e-02  0.073352 -0.124525 1.342 1.98e-03 0.1107
16   8.00e-03  0.036101 -0.146118 1.287 2.72e-03 0.0897
17   9.49e-02 -0.187099  0.313702 1.151 1.23e-02 0.1042
18  -2.17e-02  0.049963 -0.090979 1.395 1.06e-03 0.1320
19  -2.61e-02  0.090962 -0.163803 1.266 3.42e-03 0.0869
20  -3.24e-02 -0.021449  0.080554 1.334 8.31e-04 0.0941
21   1.97e-01  0.161291 -0.405586 1.148 2.05e-02 0.1363
22  -5.31e-03  0.119993 -0.256660 1.462 8.39e-03 0.2049
23  -8.61e-02 -0.049377  0.184803 1.321 4.35e-03 0.1197
24   4.40e-02 -0.033024  0.125601 3.191 2.02e-03 0.6170   *
25   6.86e-02 -0.103774 -0.181256 1.365 4.19e-03 0.1392
26  -7.37e-02 -0.105768 -0.487214 1.180 2.96e-02 0.1740
27   3.52e-01  0.109990  0.555259 0.936 3.76e-02 0.1310
28  -6.94e-03 -0.004108 -0.016054 1.405 3.30e-05 0.1287
29   4.63e-02 -0.450227 -0.750865 0.753 6.67e-02 0.1469
30   1.12e-01  0.097552  0.544326 0.770 3.54e-02 0.0944
31  -7.30e-03 -0.013034 -0.039647 1.325 2.01e-04 0.0794
32  -5.65e-04 -0.108433 -0.236830 1.301 7.12e-03 0.1291
33   4.69e-03 -0.026457 -0.057511 1.375 4.24e-04 0.1141
34  -8.00e-04  0.000487  0.001533 1.485 3.01e-07 0.1755
35  -1.19e-01 -0.286860 -0.668537 0.660 5.23e-02 0.1069
36  -5.07e-02  0.132798  0.211626 1.303 5.69e-03 0.1208
37   1.46e-01 -0.264082 -0.410324 0.971 2.07e-02 0.0931
38  -2.66e-01  0.130278  0.430561 1.323 2.33e-02 0.2021
39  -4.97e-02 -0.025216 -0.204538 1.791 5.35e-03 0.3283   *
40   9.06e-03  0.082729  0.221315 1.106 6.15e-03 0.0580
41  -2.14e-01 -0.028788 -1.005408 0.646 1.17e-01 0.1883
```

influence.measures() provides one line per case in data ...

42	−4.48e−01	0.179969	1.359365	0.611	2.09e−01	0.2625	*
43	1.81e−02	−0.012487	−0.042528	1.536	2.32e−04	0.2039	
44	1.99e−01	0.040530	0.493618	1.566	3.08e−02	0.3027	
45	3.73e−02	0.200785	0.319108	1.034	1.27e−02	0.0764	
46	1.09e−01	0.254456	0.522983	0.739	3.26e−02	0.0837	
47	−9.65e−02	0.162920	−4.256153	0.602	1.86e+00	0.6527	*
48	2.56e−02	0.111797	0.259679	1.487	8.59e−03	0.2168	

What is All that Stuff About?

- dfbetas. Change in $\hat{\beta}$ when row i is removed.
- dffits. Change in prediction for i from N-{i}
- cook.d. Cook's d summary of a case's damage
- hat value. Commonly called "leverage.
- Can ask for these one-by-one when you want them, see ?influence.measures

influence.measures Creates a Summary Object

- influence.measures is row-by-row, perhaps necessary in some situations, but excessive most of the time.
- More simply, ask which rows are potentially troublesome with the summary function:

```
summary(EXfull2infl)
```

```
Potentially influential observations of
   lm(formula = EX ~ ECAB + MET + METSQ + GROW + YOUNG + OLD + WEST,
                data = dat) :

   dfb.1_  dfb.ECAB  dfb.MET  dfb.METS  dfb.GROW  dfb.YOUN  dfb.OLD
24 -0.01   -0.04     -0.01     0.00      0.11      0.00      0.04
39  0.05    0.04      0.01    -0.04     -0.15     -0.06     -0.05
42  0.29    0.36     -0.71     0.41     -0.37     -0.25     -0.45
47  0.86   -2.92_*   -0.59     0.67     -0.65     -0.64     -0.10
   dfb.WEST  dffit    cov.r    cook.d   hat
24 -0.03      0.13    3.19_*   0.00     0.62_*
39 -0.03     -0.20    1.79_*   0.01     0.33
42  0.18      1.36_*  0.61     0.21     0.26
47  0.16     -4.26_*  0.60     1.86_*   0.65_*
```

Outline

Bear With Me for A Moment, Please

- The "solution" for the OLS estimator in matrix format is

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{1}$$

- And so the predicted value is calculated as

$$\begin{aligned} \hat{y} &= X\hat{\beta} \\ & X(X^T X)^{-1} X^T y \end{aligned}$$

- Definiton: The Hat Matrix is that big glob of X's.

$$H = X(X^T X)^{-1} X^T \tag{2}$$

Just One More Moment ...

The hat matrix is just a matrix

$$H = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1(N-1)} & h_{1N} \\ h_{21} & h_{22} & \vdots & h_{2(N-1)} & h_{2N} \\ & & & & h_{(N-1)N} \\ h_{N1} & h_{N2} & \dots & h_{N(N-1)} & h_{NN} \end{bmatrix}$$

LEVERAGE: The h_{ii} values (the "main diagonal" values of this matrix)

But it is a Very Informative Matrix!

- It is a matrix that translates observed y into predicted $\hat{y}$.
- Write out the prediction for the $i'th$ row

$$\hat{y}_i = h_{i1}y_1 + h_{i2}y_2 + \ldots + h_{iN}y_N \quad (3)$$

- That's looking at H "from side to side," to see if one case is influencing the predicted value from another.

Be clear, Could Write Out Each Case

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \\ \hat{y}_{N-1} \\ \hat{y}_N \end{bmatrix} = \begin{bmatrix} h_{11}y_1 & +h_{12}y_2 & & & +h_{1N}y_N \\ h_{21}y_1 & + & & & +h_{2N}y_N \\ h_{31}y_1 & + & \ddots & & \vdots \\ \vdots & & & & \\ & & & h_{(N-1)(N-1)}y_{N-1} & +h_{(N-1)N}y_N \\ h_{N1}y_1 & + & \cdots & +h_{N(N-1)}y_{N-1} & +h_{NN}y_N \end{bmatrix}$$

Outline

Diagonal Elements of H

- Consider at the diagonal of the hat matrix:

$$\begin{bmatrix} h_{11} & & & & \\ & h_{22} & & & \\ & & \ddots & & \\ & & & h_{N-1,N-1} & \\ & & & & h_{NN} \end{bmatrix} \tag{4}$$

- h_{ii} are customarily called “leverage” indicators
- h_{ii} DEPEND ONLY ON THE X’s. In a sense, h_{ii} measures how far a case is from “the center” or all cases.

leverage

- The sum of the leverage estimates is p, the number of parameters estimated (including the intercept).
- the most "pleasant" result would be that all of the elements are the same, so pleasant hat values would be p/N
- small h_{ii} means that the positioning of an observation in the X space is not in position to exert an extraordinary influence.

Follow Cohen, et al on this

- The hat value is a summary of how far “out of the usual” a case is on the IVs
- In a model with only one predictor, CCWA claim (p. 394)

$$h_{ii} = \frac{1}{N} + \frac{(x_i - \bar{x})^2}{\sum x_i^2} \qquad (5)$$

- If a case is “at the mean,” the h_{ii} is as small as it can get

Hat Values in the State Spending Data

```
dat$hat <- hatvalues(EXfull2)
sum(dat$hat)
```

```
[1] 8
```

```
data.frame(dat$STATE, dat$hat)
```

```
   dat.STATE    dat.hat
1         ME 0.23415534
2         NH 0.17526633
3         VT 0.27304741
4         MA 0.18281108
5         RI 0.21080976
6         CT 0.07958478
7         NY 0.18604721
8         NJ 0.10979861
9         PA 0.09538661
10        DE 0.10110559
11        MD 0.12496151
```

Hat Values in the State Spending Data ...

```
12          VA  0.16889251
13          MI  0.09095306
14          OH  0.06345230
15          IN  0.11065150
16          IL  0.08972339
17          WI  0.10423534
18          WV  0.13199636
19          KY  0.08691080
20          TE  0.09405849
21          NC  0.13631340
22          SC  0.20486326
23          GA  0.11973012
24          FL  0.61700902
25          AL  0.13918706
26          MS  0.17395231
27          MN  0.13098872
28          IA  0.12868998
29          MO  0.14694238
30          ND  0.09435984
31          SD  0.07937192
32          NB  0.12906992
```

Hat Values in the State Spending Data ...

```
33          KS  0.11410482
34          LA  0.17548605
35          AR  0.10690714
36          OK  0.12079254
37          TX  0.09309054
38          NM  0.20211747
39          AZ  0.32825519
40          MT  0.05800827
41          ID  0.18825921
42          WY  0.26252732
43          CO  0.20389684
44          UT  0.30268011
45          WA  0.07639581
46          OR  0.08367912
47          NV  0.65270615
48          CA  0.21676752
```

Outline

Fun Regression Fact

- All of the "unmeasured error terms" e_i have the same variance, σ_e^2
- For each case, we make a prediction $\hat{y}_i$ and calculate a residual, $\widehat{e_i}$
- Here's the fun fact: The variance of a residual estimate $Var(\hat{e}_i)$ is not a constant, it varies from one value of x to another.

Many Magical Properties of H

- The column of residuals is $\hat{e} = (I - H)y$
  - Proof
    $\hat{e} = y - X\hat{\beta} = y - Hy = (I - H)y$
- The elements on the diagonal of H are the important ones in many cases, because you can take, say, the 10'th observation, and you calculate the variance of the residual for that observation:

$$Var(\hat{e}_{10}) = \hat{\sigma}_e^2(1 - h_{10,10})$$

- And the estimated standard deviation of the residual is

$$Std.Err.(\hat{e}_{10}) = \hat{\sigma}_e \sqrt{1 - h_{10,10}} \tag{6}$$

Standartized Residuals (Internal Studentized Residuals)

- Recall the $Std.Err.(\hat{e}_i)$ is $\hat{\sigma}_e\sqrt{1-h_{ii}}$
- A standardized residual is the observed residual divided by its standard error

$$standardized\ residual\ \ r_i = \frac{\hat{e}_i}{\hat{\sigma}_e\sqrt{1-h_{ii}}} \tag{7}$$

- Sometimes called an internally studentized residual because case i is left in the data for the calculation of $\hat{\sigma}_e$ (same number we call RMSE sometimes)

Studentized residual (External) are t distributed

- Problem: i is included in the calculation of $\hat{\sigma}_e$.
- Fix: Recalculate the RMSE after omitting observation i, call that $\widehat{\sigma^2_{e(-i)}}$. (external, in sense i is omitted)

$$studentized\ residual:\ r_i = \frac{\hat{e}_i}{\sqrt{\widehat{\sigma^2_{e(-i)}}(1-h_{ii})}} = \frac{\hat{e}_i}{\widehat{\sigma_{e(-i)}}\sqrt{1-h_{ii}}} \qquad (8)$$

- Sometimes called R_i-Student
- That follows the Student's t distribution. That helps us set a scale.
- Have to be careful about how to set the α level (multiple comparisons problem)
- Bonferroni correction (or something like that) would have us shrink the required α level because we are making many comparisons, not just one,

The Hat in $\widehat{\sigma^2_{e(-i)}}$

- Quick Note: Not actually necessary to run new regressions to get each $\widehat{\sigma^2_{e(-i)}}$. There is a formula to calculate that from the hat matrix itself

$$\widehat{\sigma^2_{e(-i)}} = \frac{(N-p)\hat{\sigma}^2_e - \frac{e_i^2}{(1-h_{ii})}}{N-p-1} \tag{9}$$

student Residuals in the State Spending Data

```
dat$rstudent <- rstudent(EXfull2)
data.frame(dat$STATE, dat$rstudent)
```

```
   dat.STATE  dat.rstudent
1         ME -0.0830314752
2         NH -0.1216363463
3         VT  0.6722932872
4         MA -0.1674253027
5         RI -0.6929036305
6         CT  0.0168367085
7         NY  2.2406338622
8         NJ -1.5135538944
9         PA -0.5548082074
10        DE  1.2318804298
11        MD -0.0005230868
12        VA -0.0581410616
13        MI  0.8430612398
14        OH -0.2720936729
15        IN -0.3530324532
16        IL -0.4654124666
```

student Residuals in the State Spending Data ...

17	WI	0.9196178134
18	WV	−0.2333031142
19	KY	−0.5309363436
20	TE	0.2499986692
21	NC	−1.0209190989
22	SC	−0.5056470467
23	GA	0.5010905339
24	FL	0.0989555890
25	AL	−0.4507623775
26	MS	−1.0617123811
27	MN	1.4301821503
28	IA	−0.0417738620
29	MO	−1.8091618788
30	ND	1.6863319733
31	SD	−0.1350260816
32	NB	−0.6152002188
33	KS	−0.1602475953
34	LA	0.0033229393
35	AR	−1.9322821966
36	OK	0.5709463362
37	TX	−1.2807244666

student Residuals in the State Spending Data ...

38	NM	0.8554655578
39	AZ	−0.2925974799
40	MT	0.8918466200
41	ID	−2.0877223703
42	WY	2.2783571429
43	CO	−0.0840338916
44	UT	0.7492301404
45	WA	1.1095461540
46	OR	1.7306240332
47	NV	−3.1046093219
48	CA	0.4936107841

DFFIT, DFFITs

- Calculate the change in predicted value of the j'th observation due to the deletion of observation j from the dataset. Call that the DFFIT:

$$DFFIT_j = \hat{y}_j - \hat{y}_{(-j)} \tag{10}$$

- Standardize that ("studentize"? that):

$$DFFITS_j = \frac{\hat{y}_j - \hat{y}_{(-j)}}{\hat{\sigma}_{e(-j)}\sqrt{h_{jj}}} \tag{11}$$

- If $DFFITS_j$ is large, the j'th observation is influential on the model's predicted value for the j'th observation. In other words, the model does not fit observation j.

Everybody is looking around for a good rule of thumb. Perhaps $DFFITS > 2\sqrt{p/N}$ means "trouble"!

DFFIT in the State Spending Data

```
dat$dffits <- dffits(EXfull2)
data.frame(dat$STATE, dat$dffits)
```

```
   dat.STATE    dat.dffits
1         ME -0.0459118130
2         NH -0.0560732529
3         VT  0.4120261306
4         MA -0.0791883166
5         RI -0.3581189852
6         CT  0.0049508563
7         NY  1.0712303640
8         NJ -0.5315598532
9         PA -0.1801586328
10        DE  0.4131443379
11        MD -0.0001976734
12        VA -0.0262095498
13        MI  0.2666702817
14        OH -0.0708234677
15        IN -0.1245252143
16        IL -0.1461181439
```

DFFIT in the State Spending Data ...

17	WI	0.3137024408
18	WV	−0.0909789124
19	KY	−0.1638033342
20	TE	0.0805539105
21	NC	−0.4055855845
22	SC	−0.2566602418
23	GA	0.1848034155
24	FL	0.1256006284
25	AL	−0.1812561277
26	MS	−0.4872136170
27	MN	0.5552589741
28	IA	−0.0160542728
29	MO	−0.7508648469
30	ND	0.5443256931
31	SD	−0.0396468795
32	NB	−0.2368303532
33	KS	−0.0575111888
34	LA	0.0015330090
35	AR	−0.6685372163
36	OK	0.2116263097
37	TX	−0.4103236169

DFFIT in the State Spending Data ...

38	NM	0.4305612881
39	AZ	−0.2045381030
40	MT	0.2213153782
41	ID	−1.0054076251
42	WY	1.3593650114
43	CO	−0.0425280086
44	UT	0.4936181869
45	WA	0.3191076238
46	OR	0.5229829043
47	NV	−4.2561533241
48	CA	0.2596787408

Outline

"drop-one-at-a-time" analysis of slopes

- Find out if an observation influences the estimate of a slope parameter.
- Let
  - $\hat{\beta}$ a vector of regression slopes estimate using all of the data points
  - $\hat{\beta}_{(-j)}$ slopes estimate after removing observation j .
- The **DFBETA** value, a measure of influence of observation j on the parameter estimate, is

$$d_j = \hat{\beta} - \hat{\beta}_{(-j)} \tag{12}$$

If an element in this vector is huge, it means you should be cautious about observation j.

DFBETAS is Standardized DFBETA

The notation is getting tedious here

DFBETAS is considered one-variable-at-a-time, one data row at a time.

Let $d[i]_j$ be the change in the estimate of $\hat{\beta}_i$ when row j is omitted.

Standardize that:

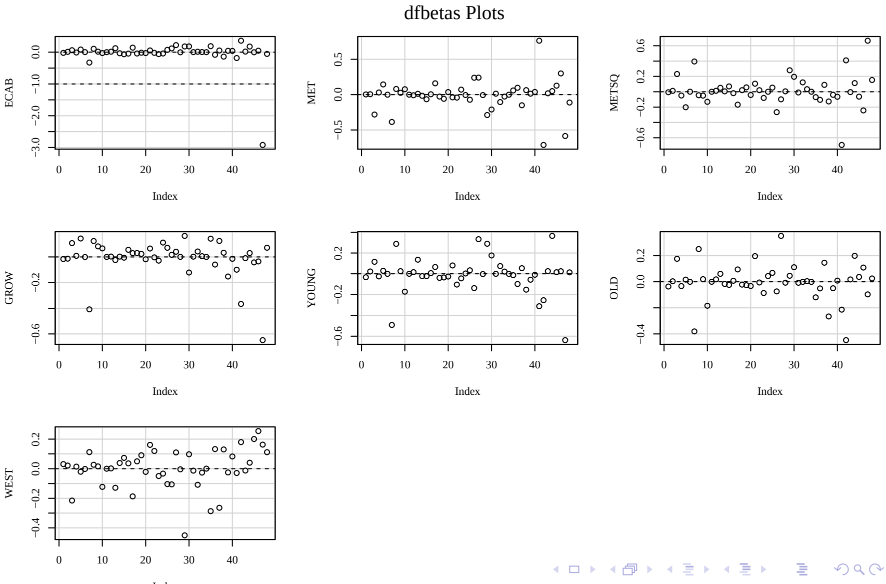
$$d[i]_j* = \frac{d[i]_j}{\sqrt{Var(\hat{\beta}_{i(-j)})}} \qquad (13)$$

The denominator is the standard error of the estimated coefficient when j is omitted.

A rule of thumb that is often brought to bear: If the DFBETAS value for a particular coefficient is greater than $2/\sqrt{N}$ then the influence is large.

dfbetas in the State Spending Data

dfbetas in the State Spending Data ...



Comes Back To The Hat

- Of course, you are wondering why I introduced DFBETA relates to the hat matrix.
- Well, the matrix calculation is:

$$d[i]_j = \frac{\hat{e}(X'X)^{-1}X_j}{1-h_{ii}} \tag{14}$$

Outline

Cook: Integrating the DFBETA

- The DFBETA analysis is unsatisfying because we can calculate a whole vector of DFBETAS, one for each parameter, but we only analyze them one-by-one. Can't we combine all of those parameters?
- The Cook distance derives from this question:

  Is the vector of estimates obtained with observation j omitted, $\hat{\beta}_{(-j)}$, meaningfully different from the vector obtained when all observations are used?

- I.e., evaluate the overall distance between the point $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2, ..., \hat{\beta}_p)$ and the point $\hat{\beta}_{(-j)} = (\hat{\beta}_{1(-j)}, \hat{\beta}_{2(-j)}, ..., \hat{\beta}_{p(-j)})$.

My Kingdom for Reasonable Weights

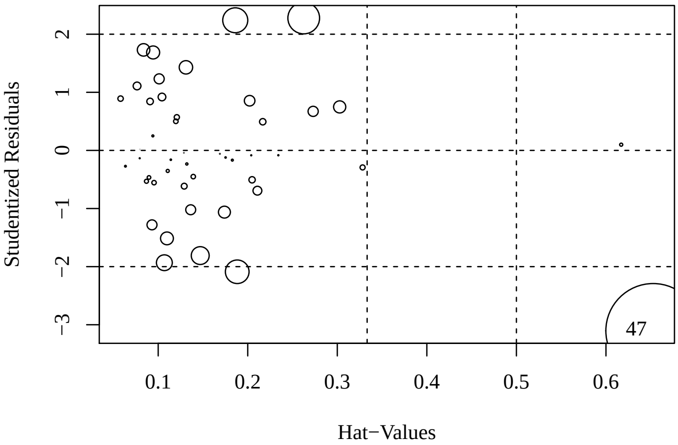
If we were interested only in raw, unstandardized distance, we could use the usual "straight line between two points" measure of distance.

- Pythagorean Theorem

$$\sqrt{(\hat{\beta}_1 - \hat{\beta}_{1(-j)})^2 + (\hat{\beta}_2 - \hat{\beta}_{2(-j)})^2 + \ldots (\hat{\beta}_p - \hat{\beta}_{p(-j)})^2} \quad (15)$$

- Cook proposed we weight the distance calculations in order to bring them into a meaningful scale.
- The weights use the estimated $\widehat{Var(\hat{\beta})}$ to scale the results

car Package's "influencePlot" Interesting!



Matrix Explanation of Cook's Proposal

- Cook's weights: the cross product matrix divided by the number of parameters that are estimated and the MSE.

$$\frac{X'X}{p \cdot \hat{\sigma}_e^2}$$

- Cook's distance D_j summarizes the size of the difference in parameter estimates when j is omitted.

$$D_j = \frac{(\hat{\beta}_{(-j)} - \hat{\beta})' X' X (\hat{\beta}_{(-j)} - \hat{\beta})}{p \cdot \hat{\sigma}_e^2}$$

Cook D Explanation (cont)

- Think of the change in predicted value as $X(\hat{\beta}_{(-j)} - \hat{\beta})$.
- D_j is thus a squared change in predicted value divided by a normalizing factor.
- To see that, regroup as

$$D_j = \frac{[X(\hat{\beta}_{(-j)} - \hat{\beta})]'[X(\hat{\beta}_{(-j)} - \hat{\beta})]}{p \cdot \hat{\sigma}_e^2}$$

The denominator includes p because there are p parameters that can change and $\hat{\sigma}_e^2$ is, of course, your friend, the MSE, the estimate of the variance of the error term.

How does the hat matrix figure into that?

You know what's coming. Cook's distance can be calculated as:

$$D_j = \frac{r_j^2}{p} \frac{h_{jj}}{(1 - h_{jj})} \quad (16)$$

r_j^2 is the squared standardized residual.

Outline

Omit or Re-Estimate

- Fix the data!
- Omit the suspicious case
- Use a “robust” estimator with a “high breakdown” point (median versus mean).
  - in R, look at ?rlm
- Revise the whole model as a “mixture” of different random processes.
  - in R, look at package flexmix

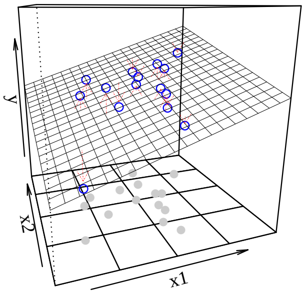
Outline

$y_i = 2 + 0.2 * x1 + 0.2 * x2 + e_i$

	M1 Estimate (S.E.)
(Intercept)	-2.143
	(6.649)
x1	0.239*
	(0.104)
x2	0.216
	(0.115)
N	15
RMSE	3.952
R^2	0.492
adj R^2	0.408

$*p \le 0.05 ** \ p \le 0.01 *** p \le 0.001$

15 cases observed

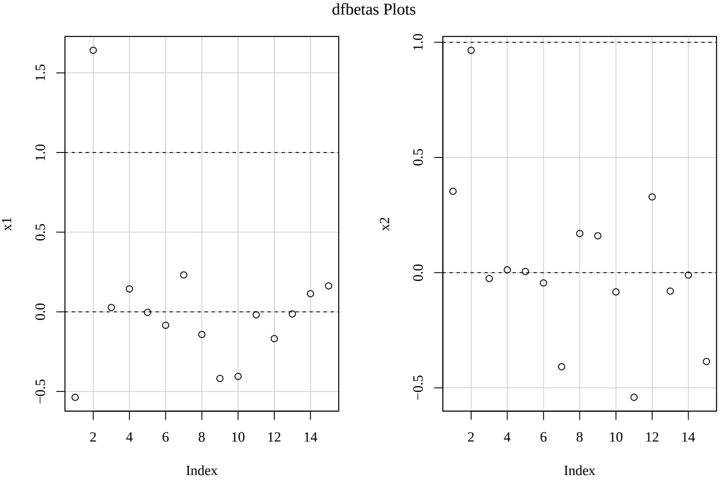


rstudent: scan for large values (t distributed)

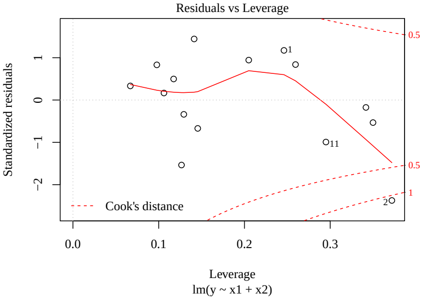
```
rstudent(modbase)
```

```
         1          2          3          4          5
 1.1932211 -3.1179432  0.1592772  0.8196170  0.3207992
         6          7          8          9         10
-0.1677531 -1.6399001 -0.6538475  0.8271355  1.5196840
        11         12         13         14         15
-0.9913500 -0.5159835 -0.3251802  0.4815630  0.9391112
```

dfbetas



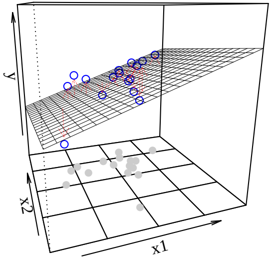
leverage


Residuals vs Leverage
Standardized residuals
Leverage
lm(y ~ x1 + x2)
Cook's distance
0.5
0.5
1
1
2
11
0.0
0.1
0.2
0.3
−2
−1
0
1

Add high h_{ii} case, observation 16 (x1=50, x2=0, y=30)

	M1 Estimate (S.E.)
(Intercept)	8.270
	(7.240)
x1	0.294*
	(0.131)
x2	-0.060
	(0.089)
N	16
RMSE	5.035
R^2	0.282
adj R^2	0.172

$*p \leq 0.05 ** \ p \leq 0.01 *** p \leq 0.001$

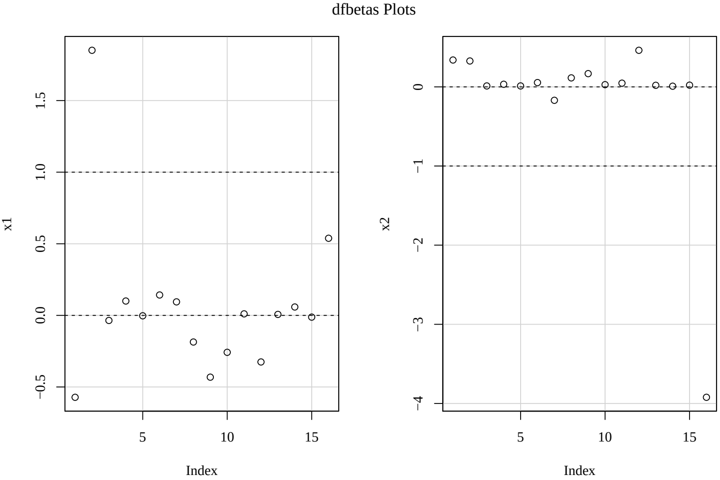


rstudent: scan for large values (t distributed)

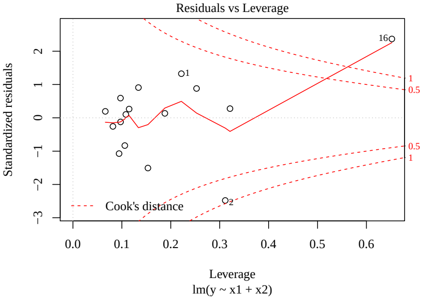
```r
rstudent(mod3A)
```

```
          1           2           3           4           5
 1.36895742 -3.28392466 -0.24788355  0.57688376  0.18713207
          6           7           8           9          10
 0.26649771 -0.82186122 -1.08070319  0.87301547  0.90171360
         11          12          13          14          15
 0.12919324 -1.59407046  0.09820715  0.25411801 -0.11766132
         16
 3.01671778
```

dfbetas


dfbetas Plots
x1
1.5
1.0
0.5
0.0
−0.5
5
10
15
Index
x2
0
−1
−2
−3
−4
5
10
15
Index

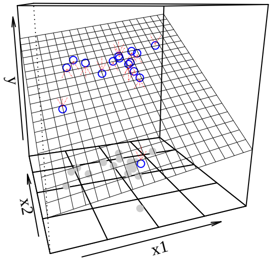
leverage


Residuals vs Leverage
Standardized residuals
Leverage
lm(y ~ x1 + x2)
Cook's distance
16
1
2
0.5
1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
−3
−2
−1
0
1
2

Set the 16th case at (mean(x1), 0), but set y=-10

	M1 Estimate (S.E.)
(Intercept)	-12.338
	(7.175)
x1	0.184
	(0.130)
x2	0.485***
	(0.089)
N	16
RMSE	4.989
R^2	0.736
adj R^2	0.695

$*p \leq 0.05$ $**\ p \leq 0.01$ $***p \leq 0.001$

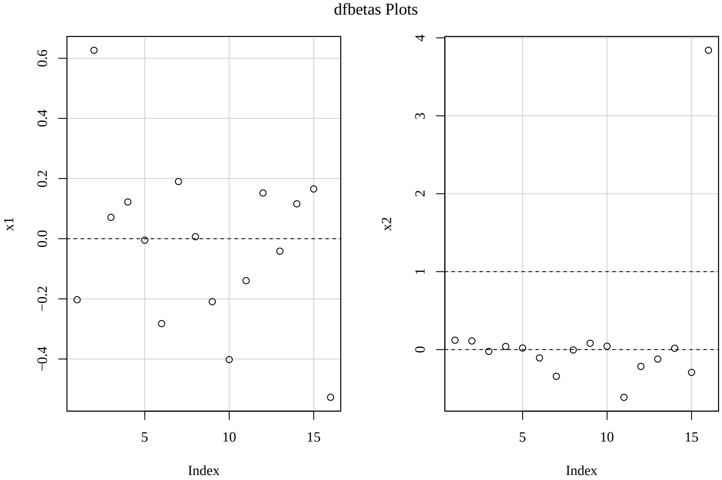


rstudent: scan for large values (t distributed)

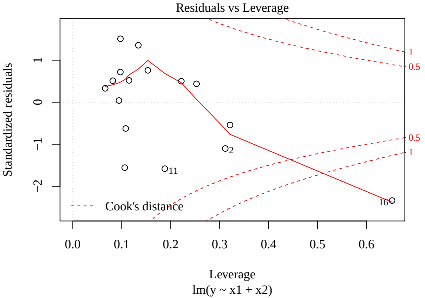
```
rstudent(mod3B)
```

```
          1           2           3           4           5
 0.48503229 -1.11129635  0.49618474  0.70301679  0.31878666
          6           7           8           9          10
-0.52805632 -1.65798723  0.03912669  0.42338176  1.40456830
         11          12          13          14          15
-1.69095670  0.74456352 -0.60918063  0.50339289  1.59667282
         16
-2.95368681
```

dfbetas


dfbetas Plots
x1
0.6
0.4
0.2
0.0
−0.2
−0.4
5
10
15
Index
x2
4
3
2
1
0
5
10
15
Index

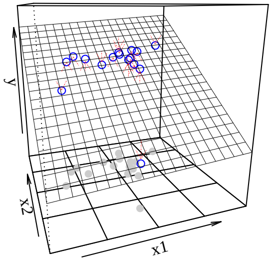
leverage


Residuals vs Leverage
Standardized residuals
Leverage
lm(y ~ x1 + x2)
Cook's distance
0.0
0.1
0.2
0.3
0.4
0.5
0.6
-2
-1
0
1
0.5
1
2
11
16

Add a case at (mean(x1), 0), but set y[16]=-30

	M1 Estimate (S.E.)
(Intercept)	-22.643
	(10.837)
x1	0.130
	(0.196)
x2	0.757***
	(0.134)
N	16
RMSE	7.535
R^2	0.730
adj R^2	0.688

$*p \leq 0.05 ** \; p \leq 0.01 *** p \leq 0.001$

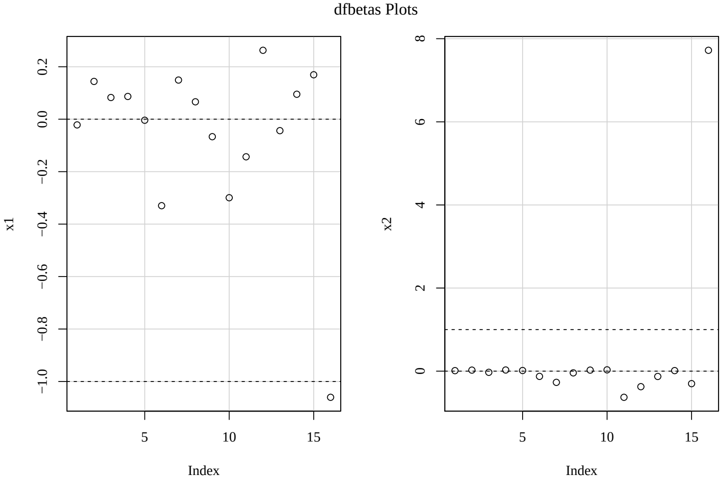


rstudent: scan for large values (t distributed)

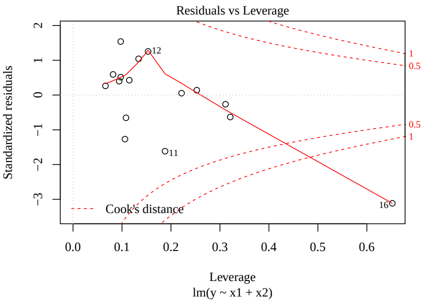
```
rstudent(mod3C)
```

```
          1           2           3           4           5
 0.05177934 -0.25553304  0.57832227  0.49929803  0.25354062
          6           7           8           9          10
-0.61678953 -1.30133514  0.38586627  0.13470213  1.04592272
         11          12          13          14          15
-1.73649745  1.28737808 -0.63930398  0.41301422  1.63620854
         16
-5.93888910
```

dfbetas


dfbetas Plots
x1
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
5
10
15
Index
x2
8
6
4
2
0
5
10
15
Index

leverage


Residuals vs Leverage
Standardized residuals
2
1
0
−1
−2
−3
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Leverage
lm(y ~ x1 + x2)
12
11
16
1
0.5
0.5
1
Cook's distance

Outline

Regression Diagnostics

1. Run the R function influence.measures() on a fitted regression model. Try to understand the output.
2. Here's some code for an example that I had planned to show in class, but did not think there would be time. This shows several variations on the "not all extreme points are dangerous outliers" theme. I hope you can easily enough cut-and paste the code into an R file that you can step through. The file "outliers.R" in the same folder as this document has this code in it.

```
set.seed(22323)
stde <- 3
x <- rnorm(15, m=50, s=10)
y <- 2 + 0.4 *x + stde * rnorm(15,m=0,s=1)
plot(y~x)
mod1 <- lm(y~x)
summary(mod1)
abline(mod1)
## add in an extreme case
```

Regression Diagnostics ...

```
x[16] <- 100
y[16] <-
predict(mod1, newdata=data.frame(x=100))+ stde*rnorm(1)
plot(y~x)
mod2 <- lm(y~x, x=T)
summary(mod2)
abline(mod2)
hatvalues(mod2)
rstudent(mod2)
mod2x <- mod2$x
fullHat <-
mod2x %*% solve(t(mod2x) %*% mod2x) %*% t(mod2x)
round(fullHat, 2)
colSums(fullHat) ##all 1
sum(diag(fullHat))
##
x[16] <- 100
y[16] <- 10
```

Regression Diagnostics ...

```
plot(y~x)
abline(mod2, lty=1)
mod3 <- lm(y~x, x=T)
summary(mod3)
abline(mod3, lty=2)
hatvalues(mod3) ##hat values same
rstudent(mod3)
mod3x <- mod3$x
fullHat <-
mod3x %*% solve(t(mod3x) %*% mod3x) %*% t(mod3x)
round(fullHat, 2)
colSums(fullHat) ##all 1
sum(diag(fullHat))
round(dffits(mod3),2)
dfbetasPlots(mod2)
dfbetasPlots(mod3)
stde <- 3
x1 <- rnorm(15, m=50, s=10)
```

Regression Diagnostics ...

```
x2 <- rnorm(15, m=50, s=10)
y <- 2 + 0.2 *x1 + 0.2*x2 + stde * rnorm(15,m=0,s=1)
plot(y~x)
mod4 <- lm(y~x1 + x2)
summary(mod4)
abline(mod1)
```