Math 3080 § 1. **Beam Example: Identifying Influential** Name: Example
Treibergs **Observations using the Hat Matrix** March 22, 2014

This **R**© program explores influential observations and their detection using the hat matrix. We use Devore's data although we do not recover the same numbers he gets.

This data was taken from Devore, *Probability and Statistics for Enigneering and the Sciencws,* 8th ed., Brooks/Cole, Boston, 2012. data is taken from the article "Testing for the Inclusion of Variables in Linear Regression.." in *Technometrics* 1966, which was reanalyzed by Hoaglin & Welsch, "The Hat Matrix..." *Amer. Statistician,* 1978. The strength of a beam y is to be regressed on specific gravity x_1 and moisture content x_2.

Recall, that to find the least squares fit to the linear model

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \epsilon_i$$

where ϵ_i are n IID $N(0, \sigma^2)$ random variables, we solve for the vector of estimated coefficients

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

where k is the number of variables, $\hat\beta$ is the $(k+1) \times 1$ matrix of estimated coefficients, X is the $n \times (k+1)$ dimensional model matrix consisting of columns of ones and independent variables and ϵ and y are $n \times 1$ dimensional martices of noise and observed responses. (See my *Colleges Example,* where this is derived.)

The hat matrix is given by

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

It gives the fitted values as a function of the observed values $\hat y = Hy$. On average, the diagonals of the hat matrix h_{ii} tell the sensitivity of the ith fitted value to the ith observed value. These are referred to as *leverage values.* Thus large hat diagonals signal influential observations. Since the average of the h_{ii}'s is $(k+1)/n$, Devore suggests that any diagonal hat entries larger than twice the average, $h_{ii} > 2(k+1)/n$, should be identified as potentially influential.

Further Devore looks at the sensitivity to omission of the ith observation on the computations of the $\hat\beta$'s. If we let $\hat\beta_j$ be the estimated coefficient using all observations and $\tilde\beta_j$ the estimate after deleting the ith observation, then a measure of the effect of the ith observation on the jth beta is

$$\frac{\hat\beta_j - \tilde\beta_j}{s_{\hat\beta_j}}.$$

if this is large, on the order of one (difference on the order of a standard deviation) then this is indicative of an influential observation. Some statisticians regard larger than two as large. The actual quantity that is computed differs from this formula slightly and is called $dffbeta$, whch can be generated by the software.

Similarly the software computes $dffit$ for each observation, which is the difference in the fitted value relative to the standard deviation due to the omission of the ith observation. The studentized residuals (as opposed to the standardized) are computed by leaving out the ith observation to for the fitted value, which should look like the standardized residual unless the point is influential. The Cook's distance is another measure that averages all $dfbeta$'s and has the same range as the square of the betas.

Influential observations should be mentioned in summarizing your data analysis. First, if outliers and influential can be identified as typographical or measurement errors, they can be removed from the sample. Otherwise you have to deal with the data as it is. If the fitted line does not change much after removing the influential points then just fit with all data points. But if the coefficients change, then a range of coefficients should be reported. Omitting influential points will underreport the variation.

Data Set Used in this Analysis :

```
# Math 3080-1                Beam Data             March 29, 2014
# Treibergs
#
# From Devore, "Probability and Statistics for Engineering and the
# Sciences," 8th ed., Brooks/Cole, Boston, example 13.24.
# From the article "Testing for the Inclusion of Variables in Linear
# Regression.." in Technometrics 1966, which was reanalyzed by Hoaglin &
# Welsch, "The Hat Matrix," Amer. Statistician, 1978.
#
# variables
#   x1  specific gravity
#   x2  moisture content
#    y  strength
"x1" "x2" "y"
.499 11.1 11.14
.558  8.9 12.74
.604  8.8 13.13
.411  8.9 11.51
.550  8.8 12.38
.528  9.9 12.60
.418 10.7 11.13
.480 10.5 11.70
.406 10.5 11.02
.467 10.7 11.41
```

R Session:

```
R version 2.13.1 (2011-07-08)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-apple-darwin9.8.0/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type ’license()’ or ’licence()’ for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type ’contributors()’ for more information and
’citation()’ on how to cite R or R packages in publications.

Type ’demo()’ for some demos, ’help()’ for on-line help, or
’help.start()’ for an HTML browser interface to help.
Type ’q()’ to quit R.

[R.app GUI 1.41 (5874) i386-apple-darwin9.8.0]
```

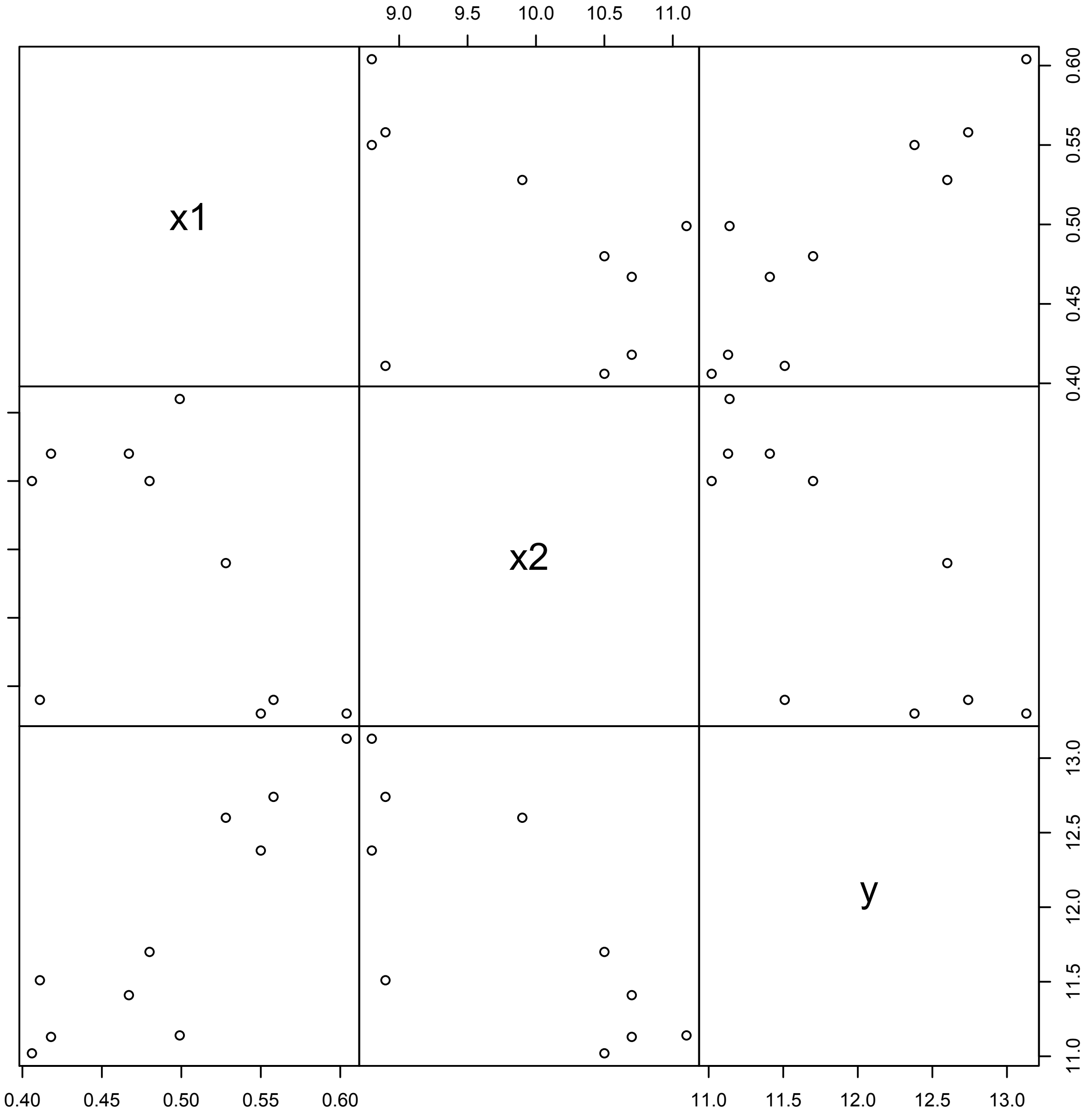
```
[History restored from /Users/andrejstreibergs/.Rapp.history]

> tt=read.table("M3082DataBeam.txt",header=T)
> attach(tt)
> tt
      x1   x2     y
1  0.499 11.1 11.14
2  0.558  8.9 12.74
3  0.604  8.8 13.13
4  0.411  8.9 11.51
5  0.550  8.8 12.38
6  0.528  9.9 12.60
7  0.418 10.7 11.13
8  0.480 10.5 11.70
9  0.406 10.5 11.02
10 0.467 10.7 11.41
>
> ################## PRINT PAIRWISE SCARTTERPLOTS FOR THEIS DATA #########
> pairs(tt,gap=0)
>
> ################## RUN THE LINEAR MODEL OF THIS DATA ###################
> f1 = lm(y ~ x1 + x2 )
> summary(f1); anova(f1)
Call:
lm(formula = y ~ x1 + x2)
Residuals:
     Min       1Q   Median       3Q      Max
-0.38607 -0.05198  0.02194  0.07932  0.45560

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.3683     1.5985   7.112 0.000192 ***
x1            7.6601     1.4984   5.112 0.001381 **
x2           -0.3301     0.1093  -3.020 0.019382 *
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.2605 on 7 degrees of freedom
Multiple R-squared: 0.9105,Adjusted R-squared: 0.8849
F-statistic: 35.61 on 2 and 7 DF,  p-value: 0.0002144

Analysis of Variance Table
Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
x1         1 4.2159  4.2159 62.1040 0.0001003 ***
x2         1 0.6192  0.6192  9.1213 0.0193822 *
Residuals  7 0.4752  0.0679
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

> #########  NOTE THAT THIS DIFFERS FROM TABLE 13.12 IN DEVORE  ##########
> #########  I WAS UNABLE DEVORE'S CENTERING OR STANDARDIZATION ##########
> ######### COMPUTE THE MATRICES "BY HAND"  ##############################
>
> X = model.matrix(f1); X
   (Intercept)    x1   x2
1            1 0.499 11.1
2            1 0.558  8.9
3            1 0.604  8.8
4            1 0.411  8.9
5            1 0.550  8.8
6            1 0.528  9.9
7            1 0.418 10.7
8            1 0.480 10.5
9            1 0.406 10.5
10           1 0.467 10.7
attr(,"assign")
[1] 0 1 2
>
> ############   HERE IS A WAY TO COMPUTE INVERSE MATRICES ##########
> ############ START WITH AN EXAMPLE OF A 3X3 SYMMETRIC MATRIX ######
>
> ExMx=matrix(c(6,6,4,6,8,2,4,2,10),ncol=3); ExMx
     [,1] [,2] [,3]
[1,]    6    6    4
[2,]    6    8    2
[3,]    4    2   10
> ExMxInv = chol2inv(chol(ExMx)); ExMxInv
        [,1]    [,2]    [,3]
[1,]  1.1875 -0.8125 -0.3125
[2,] -0.8125  0.6875  0.1875
[3,] -0.3125  0.1875  0.1875
>
> ############# CHECK IT IS THE INVERSE #######################
>
> ExMx %*% ExMxInv;  ExMxInv %*% ExMx
             [,1]          [,2]          [,3]
[1,] 1.000000e+00 -7.771561e-16 -1.110223e-16
[2,] 2.220446e-16  1.000000e+00  1.665335e-16
[3,] 0.000000e+00  2.220446e-16  1.000000e+00
              [,1]         [,2]         [,3]
[1,]  1.000000e+00 2.220446e-16 0.000000e+00
[2,] -7.771561e-16 1.000000e+00 2.220446e-16
[3,] -1.110223e-16 1.665335e-16 1.000000e+00
```

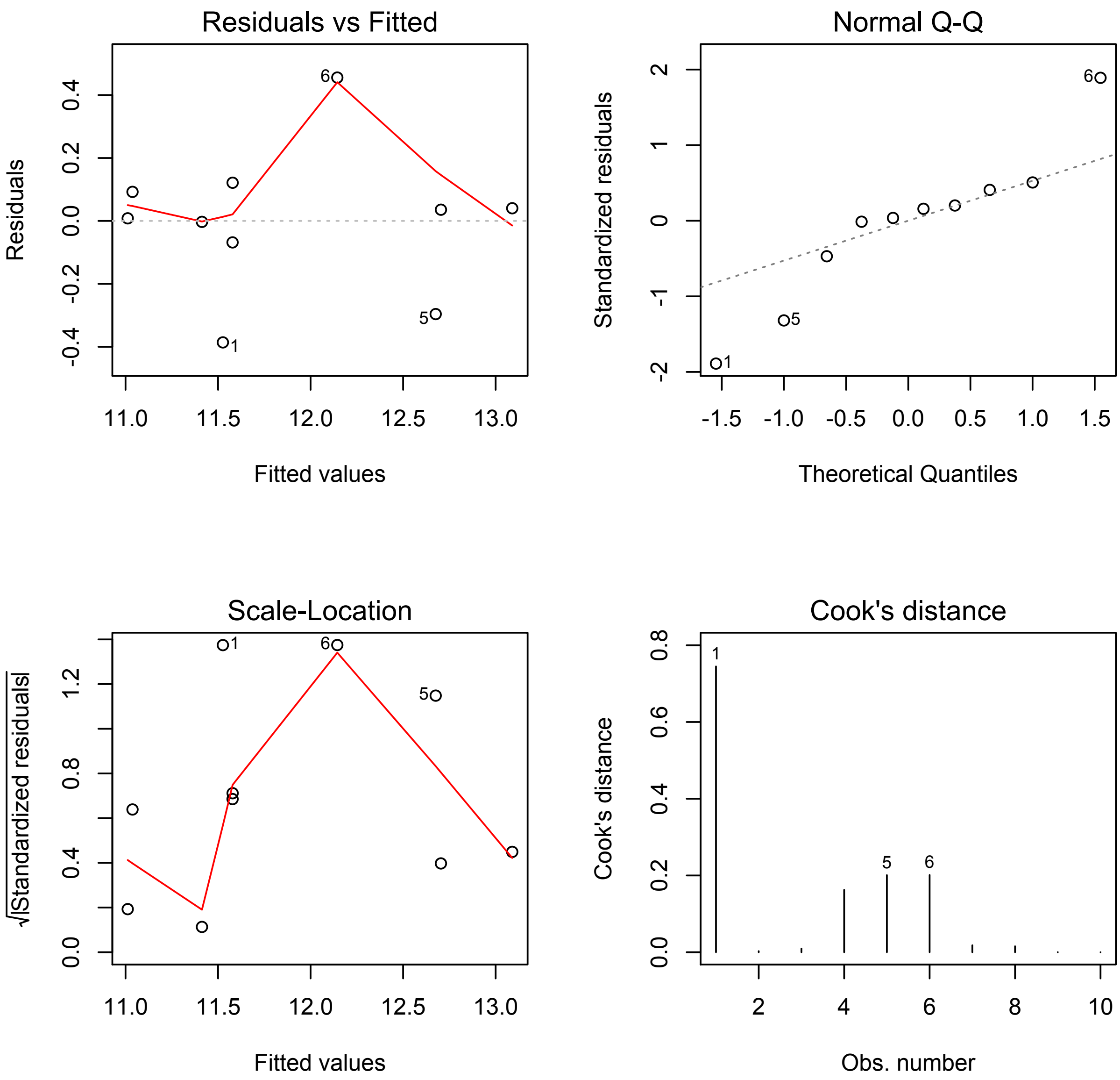
```
> ############## USE MATRIX ARITHMETIC TO RECOVER HAT-BETA  #####
>
> > XTX = t(X) %*% X; XTX
            (Intercept)        x1       x2
(Intercept)      10.000  4.921000  98.8000
x1                4.921  2.463435  48.3179
x2               98.800 48.317900 984.0000
```

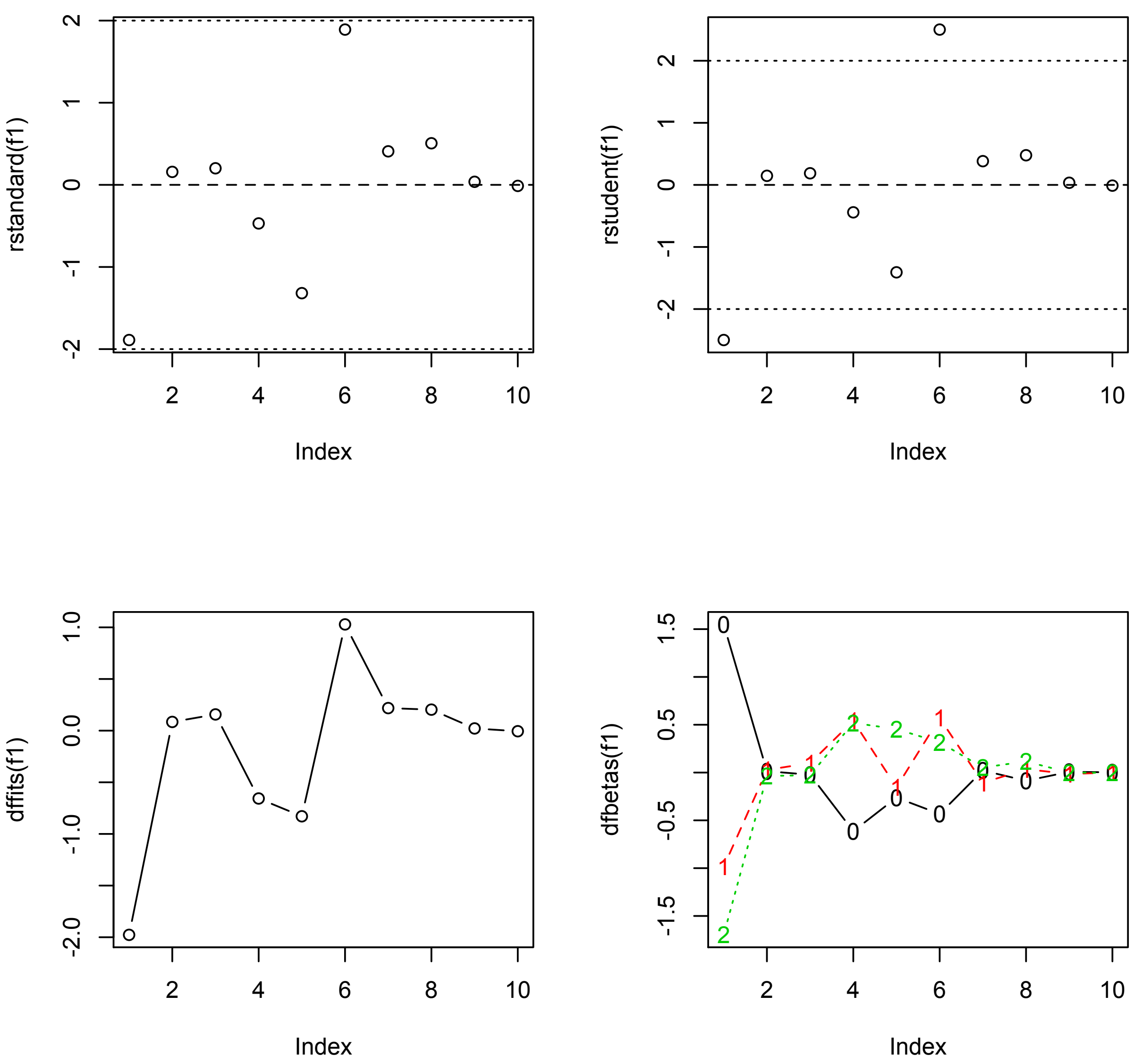
```
> XTXInv = chol2inv(chol(XTX)); XTXInv
          [,1]       [,2]       [,3]
[1,]  37.639824 -28.821426 -2.3640486
[2,] -28.821426  33.075676  1.2697253
[3,]  -2.364049   1.269725  0.1760341
> ####################  CHECK IT IS THE INVERSE  #########
> XTXInv %*% XTX
      (Intercept)            x1            x2
[1,]  1.000000e+00 -7.105427e-14 -9.094947e-13
[2,] -1.421085e-14  1.000000e+00  0.000000e+00
[3,]  0.000000e+00  3.552714e-15  1.000000e+00

> ################## NOW USE HAT-BETA = (XTX)^(-1) XT Y #####
> XTX = t(X) %*% X; XTX
            (Intercept)        x1       x2
(Intercept)      10.000  4.921000  98.8000
x1                4.921  2.463435  48.3179
x2               98.800 48.317900 984.0000

> XTXInv = chol2inv(chol(XTX)); XTXInv
          [,1]       [,2]       [,3]
[1,]  37.639824 -28.821426 -2.3640486
[2,] -28.821426  33.075676  1.2697253
[3,]  -2.364049   1.269725  0.1760341

> ####################  CHECK IT IS THE INVERSE  #########
> XTXInv %*% XTX
      (Intercept)            x1            x2
[1,]  1.000000e+00 -7.105427e-14 -9.094947e-13
[2,] -1.421085e-14  1.000000e+00  0.000000e+00
[3,]  0.000000e+00  3.552714e-15  1.000000e+00
>
> ################  FORMULA FOR HAT-BETA  ################
> cbind(coefficients(f1), XTXInv%*% t(X) %*% Y)
                  [,1]       [,2]
(Intercept) 11.3683186 11.3683186
x1           7.6601449  7.6601449
x2          -0.3301494 -0.3301494

> ##############  HAT MATRIX  ############################
> Hat = X %*% XTXInv %*% t(X);  Hat
             1            2            3           4           5           6
1   0.384960960 -0.001929016  0.057473921 -0.26319022 -0.03849958  0.16827506
2  -0.001929016  0.248701706  0.300611941  0.11120435  0.25010271  0.13180252
3   0.057473921  0.300611941  0.412589174 -0.04187818  0.28677648  0.18268145
4  -0.263190216  0.111204346 -0.041878176  0.68843926  0.17016730 -0.04648074
5  -0.038499581  0.250102707  0.286776478  0.17016730  0.25741245  0.11718965
6   0.168275060  0.131802517  0.182681453 -0.04648074  0.11718965  0.14452200
7   0.151591632 -0.042157466 -0.112031305  0.16507325 -0.03590573  0.05039582
8   0.217078893  0.033602318  0.042027218 -0.02328874  0.02112846  0.11576935
9   0.085559649 -0.035614040 -0.130382773  0.26729210 -0.01911113  0.02602132
10  0.238678698  0.003674987  0.002132067 -0.02733838 -0.00926060  0.10982357
```

```
            7           8           9          10
1   0.15159163  0.21707889  0.08555965  0.238678698
2  -0.04215747  0.03360232 -0.03561404  0.003674987
3  -0.11203130  0.04202722 -0.13038277  0.002132067
4   0.16507325 -0.02328874  0.26729210 -0.027338385
5  -0.03590573  0.02112846 -0.01911113 -0.009260600
6   0.05039582  0.11576935  0.02602132  0.109823571
7   0.24567547  0.14821977  0.25254000  0.176598560
8   0.14821977  0.15345915  0.12482011  0.167183461
9   0.25254000  0.12482011  0.27730348  0.151571288
10  0.17659856  0.16718346  0.15157129  0.186936351

> ###############  NOTE THAT THIS IS DIFFERENT THAN p.583       #############
> ###############  CHECK THAT IT IS THE HAT MATRIX. IT RECOVERS #############
> ###############  THE FITTED VALUES HAT-Y = HAT * Y.           #############
> cbind( fitted(f1), Hat %*% y)
       [,1]     [,2]
1  11.52607 11.52607
2  12.70435 12.70435
3  13.08973 13.08973
4  11.57831 11.57831
5  12.67608 12.67608
6  12.14440 12.14440
7  11.03766 11.03766
8  11.57862 11.57862
9  11.01177 11.01177
10 11.41301 11.41301
>
> ##########  WE CAN EXTRACT THE DIAGONAL OF HAT FROM THE MODEL  ########
>
> cbind(hatvalues(f1), diag(Hat))
        [,1]      [,2]
1  0.3849610 0.3849610
2  0.2487017 0.2487017
3  0.4125892 0.4125892
4  0.6884393 0.6884393
5  0.2574125 0.2574125
6  0.1445220 0.1445220
7  0.2456755 0.2456755
8  0.1534592 0.1534592
9  0.2773035 0.2773035
10 0.1869364 0.1869364

> ########  THE CRITICAL VALUES ARE DOUBLE OR TRIPLE AVERAGE VALUES  #####
>
> > n=length(x1); k=2; DoubleMeanHat=2*(k+1)/n;TripleMeanHat=3*(k+1)/n;
> DoubleMeanHat; TripleMeanHat
[1] 0.6
[1] 0.9
>
> ############# SO THE FOURTH OBSERVATION IS POTENTIALLY INFLUENTIAL  ####
> ############# RUN THE REGRESSION WITHOUT OBS. 4  ######################
> f5=lm(y~x1+x2,data=tt, subset=-4); summary(f5)

Call:
lm(formula = y ~ x1 + x2, data = tt, subset = -4)

Residuals:
     Min       1Q   Median       3Q      Max
-0.33339 -0.05037  0.01127  0.05615  0.46579

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  12.4107     2.9071   4.269  0.00527 **
x1            6.7992     2.5166   2.702  0.03549 *
x2           -0.3905     0.1794  -2.177  0.07237 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.277 on 6 degrees of freedom
Multiple R-squared: 0.9108,Adjusted R-squared: 0.8811
F-statistic: 30.65 on 2 and 6 DF,  p-value: 0.0007089

>
> ###### NORMALIZED DIFFERENCE OF BETAS WITH AND WITHOUT OBS. 4 ####
> coefficients(f1)-coefficients(f5)
(Intercept)          x1          x2
-1.04233911  0.86093069  0.06039984

> #######  COVARIANCE MATRIX FOR THE REGRESSION ###################
> vcov(f1)
            (Intercept)          x1          x2
(Intercept)   2.5551398 -1.95651215 -0.16048095
x1           -1.9565122  2.24530739  0.08619397
x2           -0.1604809  0.08619397  0.01194989

> ############  THE SD'S OF ESTIMATED BETAS IS SQRT OF DIAGONAL  ####
> sbetas = sqrt(diag(vcov(f1))); sbetas
(Intercept)          x1          x2
  1.5984805   1.4984350   0.1093155

> ##########  DIFFERENCES OF BETAS WITH AND WITHOUT OBS 4  $$$$$$$$$$

> diffbetas = coefficients(f1)-coefficients(f5); diffbetas
(Intercept)          x1          x2
-1.04233911  0.86093069  0.06039984

> sdiffbetas=diffbetas/sbetas
> sdiffbetas
(Intercept)          x1          x2
 -0.6520812   0.5745532   0.5525275
```

```
> ####  ONE COULD COMPARE THE EFFECT OF REMOVING ONE OBSERVATION
> ####  FOR EACH OF THE ESTIMATED BETAS. THE CODE USED BY  R
> ####  DIFFERS SLIGHTLY FROM THE COMPUTATION SUGGESTED IN DEVORE.
> ####  LOOK AT THE N=4 ROW IN THE PRINTOUT OF INFLUENCE MEASURES.
> ####  THE dfb (SHORT FOR dfbeta) GIVES THE CHANGE OF THE ESTIMATED
> ####  PARAMETER RELATIVE TO ITS STANDARD DEVIATION.
>
> ####  dffits ARE THE RELATIVE CHANGE OF THE FITTED VALUE DUE TO THE
> ####  OMISSION OF THE DATA POINT.
>
> ####  COOK'S DISTANCE cookd IS A JOINT MEASURE OF THE COMPONENTS OF
> ####  THE dfbetas. THE SQUARE ROOT OF COOK'S DISTANCE IS IN THE SAME
> ####  SCALE AS dfbetas. THE SUMMARY IS AVAILABLE USING influence.measures.

> if1=influence.measures(f1); if1
Influence measures of
 lm(formula = y ~ x1 + x2) :

     dfb.1_    dfb.x1    dfb.x2    dffit cov.r   cook.d   hat inf
1   1.54932 -0.984725 -1.70e+00 -1.97706 0.304 7.45e-01 0.385   *
2   0.01425  0.027472 -3.58e-02  0.08424 2.091 2.75e-03 0.249
3  -0.02278  0.098971 -2.80e-02  0.15692 2.656 9.52e-03 0.413   *
4  -0.61345  0.540519  5.20e-01 -0.65684 4.630 1.62e-01 0.688   *
5  -0.26223 -0.154524  4.54e-01 -0.82917 0.908 2.01e-01 0.257
6  -0.43299  0.570478  3.17e-01  1.02841 0.218 2.01e-01 0.145
7   0.02132 -0.107915  5.27e-02  0.21821 1.958 1.81e-02 0.246
8  -0.08605  0.034932  1.16e-01  0.20335 1.677 1.55e-02 0.153
9   0.00736 -0.014502 -1.76e-05  0.02131 2.196 1.77e-04 0.277
10  0.00239 -0.000482 -3.52e-03 -0.00568 1.953 1.26e-05 0.187
>
> #############  PLOT DIAGNOSTICS  ########################
> layout(matrix(c(1,3,2,4),ncol=2))
> plot(f1, which=1:4)
>
> plot(rstandard(f1));abline(h=c(0,-2,2),lty=c(2,3,3))
> plot(rstudent(f1));abline(h=c(0,-2,2),lty=c(2,3,3))
> plot(dffits(f1),type="b",xlab="Index")
> matplot(dfbetas(f1),type=c("b","b","b"),pch=c("0","1","2"),xlab="Index")
```


x1
x2
y
9.0
9.5
10.0
10.5
11.0
0.40
0.45
0.50
0.55
0.60
11.0
11.5
12.0
12.5
13.0


Residuals vs Fitted
Residuals
0.4
0.2
0.0
-0.2
-0.4
6
5
1
11.0
11.5
12.0
12.5
13.0
Fitted values
Normal Q-Q
Standardized residuals
2
1
0
-1
-2
6
5
1
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
1.2
0.8
0.4
0.0
1
6
5
11.0
11.5
12.0
12.5
13.0
Fitted values
Cook's distance
Cook's distance
0.8
0.6
0.4
0.2
0.0
1
5
6
2
4
6
8
10
Obs. number


rstandard(f1)
2
1
0
-1
-2
Index
2
4
6
8
10
rstudent(f1)
2
1
0
-1
-2
Index
dffits(f1)
1.0
0.0
-1.0
-2.0
Index
dfbetas(f1)
1.5
0.5
-0.5
-1.5
Index
0
1
2