A LINEAR REGRESSION SOLVER FOR GAMS

ERWIN KALVELAGEN

ABSTRACT. This document describes a linear regression solver for GAMS.

1. Introduction

The linear regression solver `LS` for GAMS calculates estimates β for the linear statistical model[27]:

$$y = X\beta + \varepsilon \tag{1}$$

The solver calculates

$$\beta = (X^TX)^{-1}X^Ty \tag{2}$$

using a numerically stable method (QR decomposition). It also calculates a number of statistical quantities such as standard errors which can be used for inference. Much information is written to a GDX file, which can be accessed by GAMS to extract the required information into a GAMS model.

This special purpose solver is more reliable than using a standard LP solver for solving the normal equations

$$(X^TX)\beta = X^Ty \tag{3}$$

which is a square system of linear equations. Similarly the non-linear formulation

$$\begin{aligned} \min z &= \varepsilon^T\varepsilon \\ y &= X\beta + \varepsilon \end{aligned} \tag{4}$$

is not always easy to solve. In addition to better numerical behavior and more efficiency, the least square solver also provides a number of regression statistics which are cumbersome to code directly in GAMS. See [23] for some examples of estimation problems stated directly in GAMS.

2. Revision history

Version 1	Download only
Version 2	Included in GAMS22.6 and later
Version 2.1	Added studentized residuals

Date: November 2007, Revised May 2009.

3. Usage

A least squares model contains a dummy objective and a set of linear equations:

```
sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= b0 + b1*data(i,'x');

option lp = ls;
model leastsq /fit,sumsq/;
solve leastsq using lp minimizing sse;
```

Here `sse` is a free variable, in which the solver returns the sum of squared errors. The free variables `b0` and `b1` are the coefficients to be estimated. For more examples see the section with example models at the end of this document. A simple complete example is shown in the next section.

The constant term or intercept is included in the above example. If you don't specify it explicitly, and the solver detects absence of a column of ones in the data matrix X, then a constant term will be added automatically. When you need to do a regression without intercept you will need to use an option '`add_constant_term 0`' (see section 5).

It is not needed or beneficial to specify initial values (levels) or an advanced basis (marginals) as they are ignored by the solver.

The estimates are returned as the levels of the variables. The marginals will contain the standard errors. The row levels reported are the residuals $\hat{\varepsilon} = y - \hat{y} = y - X\hat{\beta}$. In addition a GDX file is written which will contain all regression statistics.

4. Example

Consider the following data from [20]: we have 40 cross section observations of weekly household expenditure on food and on weekly household income (see table 1). We assume that the 'consumption function' is linear. The graph 1 indicates that indeed a linear relationship is an appropriate model to describe this data set:

$$\text{food} = b_0 + b_1\text{income} \tag{5}$$

4.1. **Example model.** The complete GAMS model looks like[1]:

```
$ontext

 Regression example

 Cross-section data: weekly household expenditure on food and
 weekly household income from Griffiths, Hill and Judge,
 1993, Table 5.2, p. 182.


 Erwin Kalvelagen, october 2000

$offtext


set i /i1*i40/;

table data(i, *)
        expenditure income
i1         9.46       25.83
i2        10.56       34.31
i3        14.81       42.50
i4        21.71       46.75
```

[1] www.amsterdamoptimization.com/models/regression/ghj.gms

food	income	food	income
9.46	25.83	17.77	71.98
10.56	34.31	22.44	72.00
14.81	42.50	22.87	72.23
21.71	46.75	26.52	72.23
22.79	48.29	21.00	73.44
18.19	48.77	37.52	74.25
22.00	49.65	21.69	74.77
18.12	51.94	27.40	76.33
23.13	54.33	30.69	81.02
19.00	54.87	19.56	81.85
19.46	56.46	30.58	82.56
17.83	58.83	41.12	83.33
32.81	59.13	15.38	83.40
22.13	60.73	17.87	91.81
23.46	61.12	25.54	91.81
16.81	63.10	39.00	92.96
21.35	65.96	20.44	95.17
14.87	66.40	30.10	101.40
33.00	70.42	20.90	114.13
25.19	70.48	48.71	115.46

TABLE 1. A household food expenditure data set

```
i5        22.79       48.29
i6        18.19       48.77
i7        22.00       49.65
i8        18.12       51.94
i9        23.13       54.33
i10       19.00       54.87
i11       19.46       56.46
i12       17.83       58.83
i13       32.81       59.13
i14       22.13       60.73
i15       23.46       61.12
i16       16.81       63.10
i17       21.35       65.96
i18       14.87       66.40
i19       33.00       70.42
i20       25.19       70.48
i21       17.77       71.98
i22       22.44       72.00
i23       22.87       72.23
i24       26.52       72.23
i25       21.00       73.44
i26       37.52       74.25
i27       21.69       74.77
i28       27.40       76.33
i29       30.69       81.02
i30       19.56       81.85
i31       30.58       82.56
i32       41.12       83.33
i33       15.38       83.40
i34       17.87       91.81
i35       25.54       91.81
i36       39.00       92.96
i37       20.44       95.17
i38       30.10      101.40
i39       20.90      114.13
```

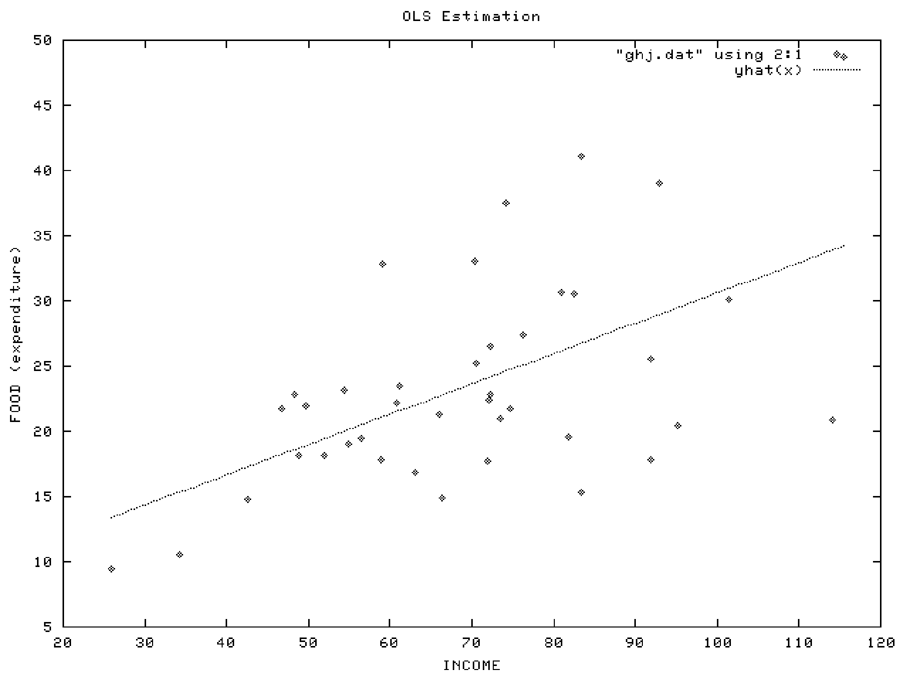

FIGURE 1. OLS Estimation

```
i40      48.71      115.46
;


variables
    constant     'estimate constant term coefficient'
    income       'estimate income coefficient'
    sse          'sum of squared errors'
;

equations
    fit(i)     'the linear model'
    obj        'objective'

;

obj..     sse =n= 0;
fit(i)..  data(i,'expenditure') =e= constant + income*data(i,'income');

option lp=ls;
model ols1 /obj,fit/;
solve ols1 minimizing sse using lp;

display constant.l, income.l, sse.l;
```

The log file looks like:

```
--- Job ghj.gms Start 11/17/07 22:09:51
GAMS Rev 148  Copyright (C) 1987-2007 GAMS Development. All rights reserved
Licensee: Erwin Kalvelagen                               G070509/0001CE-WIN
          GAMS Development Corporation                               DC4572
--- Starting compilation
--- ghj.gms(82) 3 Mb
--- Starting execution
--- ghj.gms(77) 4 Mb
```

```
--- Generating LP model ols1
--- ghj.gms(79) 4 Mb
---   41 rows  3 columns  81 non-zeroes
--- Executing LS

 ======================================================================
   Least Square Solver V2.1
   Erwin Kalvelagen, Amsterdam Optimization Modeling Group
   www.amsterdamoptimization.com
 ======================================================================

         Parameter      Estimate    Std. Error       t value      Pr(>|t|)
          constant   0.73832E+01   0.40084E+01   0.18420E+01   0.73296E-01 .
            income   0.23225E+00   0.55293E-01   0.42004E+01   0.15514E-03 ***
      ---
      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

 Estimation statistics:
 Cases: 40  Parameters: 2  Residual sum of squares:   0.17804E+04
 Residual standard error:   0.68449E+01   on  38 degrees of freedom
 Multiple R-squared:   0.31708E+00  Adjusted R-squared:   0.29911E+00
 F statistic:   0.17643E+02 on 1 and 38 DF, p-value:   0.15514E-03

 DLL version: _GAMS_GDX_237_2007-01-09
 GDX file: ls.gdx
--- Restarting execution
--- ghj.gms(79) 0 Mb
--- Reading solution for model ols1
--- Executing after solve
--- ghj.gms(81) 3 Mb
*** Status: Normal completion
--- Job ghj.gms Stop 11/17/07 22:09:51 elapsed 0:00:00.890
```

The listing file will contain similar information:

```
               S O L V E      S U M M A R Y

     MODEL   ols1                OBJECTIVE  sse
     TYPE    LP                  DIRECTION  MINIMIZE
     SOLVER  LS                  FROM LINE  79

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE             1780.4126

 RESOURCE USAGE, LIMIT          0.250      1000.000
 ITERATION COUNT, LIMIT         1         10000

 ======================================================================
   Least Square Solver V2.1
   Erwin Kalvelagen, Amsterdam Optimization Modeling Group
   www.amsterdamoptimization.com
 ======================================================================

         Parameter      Estimate    Std. Error       t value      Pr(>|t|)
          constant   0.73832E+01   0.40084E+01   0.18420E+01   0.73296E-01 .
            income   0.23225E+00   0.55293E-01   0.42004E+01   0.15514E-03 ***
      ---
      Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

 Estimation statistics:
 Cases: 40  Parameters: 2  Residual sum of squares:   0.17804E+04
 Residual standard error:   0.68449E+01   on  38 degrees of freedom
 Multiple R-squared:   0.31708E+00  Adjusted R-squared:   0.29911E+00
 F statistic:   0.17643E+02 on 1 and 38 DF, p-value:   0.15514E-03

 DLL version: _GAMS_GDX_237_2007-01-09
 GDX file: ls.gdx

                           LOWER          LEVEL          UPPER         MARGINAL

---- EQU obj               -INF         1780.4126        +INF             .
```

```
  obj  objective

---- EQU fit  the linear model

           LOWER          LEVEL          UPPER         MARGINAL

i1          -9.4600        -3.9223        -9.4600          .
i2         -10.5600        -4.7918       -10.5600          .
i3         -14.8100        -2.4440       -14.8100          .
i4         -21.7100         3.4689       -21.7100          .
i5         -22.7900         4.1913       -22.7900          .
i6         -18.1900        -0.5202       -18.1900          .
i7         -22.0000         3.0854       -22.0000          .
i8         -18.1200        -1.3265       -18.1200          .
i9         -23.1300         3.1285       -23.1300          .
i10        -19.0000        -1.1270       -19.0000          .
i11        -19.4600        -1.0362       -19.4600          .
i12        -17.8300        -3.2167       -17.8300          .
i13        -32.8100        11.6936       -32.8100          .
i14        -22.1300         0.6420       -22.1300          .
i15        -23.4600         1.8815       -23.4600          .
i16        -16.8100        -5.2284       -16.8100          .
i17        -21.3500        -1.3526       -21.3500          .
i18        -14.8700        -7.9348       -14.8700          .
i19        -33.0000         9.2615       -33.0000          .
i20        -25.1900         1.4376       -25.1900          .
i21        -17.7700        -6.3308       -17.7700          .
i22        -22.4400        -1.6655       -22.4400          .
i23        -22.8700        -1.2889       -22.8700          .
i24        -26.5200         2.3611       -26.5200          .
i25        -21.0000        -3.4399       -21.0000          .
i26        -37.5200        12.8920       -37.5200          .
i27        -21.6900        -3.0588       -21.6900          .
i28        -27.4000         2.2889       -27.4000          .
i29        -30.6900         4.4896       -30.6900          .
i30        -19.5600        -6.8332       -19.5600          .
i31        -30.5800         4.0219       -30.5800          .
i32        -41.1200        14.3831       -41.1200          .
i33        -15.3800       -11.3731       -15.3800          .
i34        -17.8700       -10.8364       -17.8700          .
i35        -25.5400        -3.1664       -25.5400          .
i36        -39.0000        10.0265       -39.0000          .
i37        -20.4400        -9.0468       -20.4400          .
i38        -30.1000        -0.8337       -30.1000          .
i39        -20.9000       -12.9903       -20.9000          .
i40        -48.7100        14.5108       -48.7100          .

                            LOWER          LEVEL          UPPER         MARGINAL

---- VAR constant           -INF            7.3832        +INF            4.0084
---- VAR income             -INF            0.2323        +INF            0.0553
---- VAR sse                -INF         1780.4126        +INF             .

  constant  estimate constant term coefficient
  income  estimate income coefficient
  sse  sum of squared errors


**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

----     81 VARIABLE constant.L           =        7.383  estimate constant term coefficient
            VARIABLE income.L             =        0.232  estimate income coefficient
            VARIABLE sse.L                =     1780.413  sum of squared errors
```

For comparison we show the results of running this model with the econometrics package CHAZAM [41]. The OLS procedure on this data set gives:

```
|_SAMPLE 1 40
|_READ (GHJ.DAT) FOOD INCOME

UNIT 88 IS NOW ASSIGNED TO: GHJ.DAT
   2 VARIABLES AND       40 OBSERVATIONS STARTING AT OBS       1

|_OLS FOOD INCOME

 OLS ESTIMATION
      40 OBSERVATIONS     DEPENDENT VARIABLE = FOOD
 ...NOTE..SAMPLE RANGE SET TO:    1,   40

  R-SQUARE =    .3171      R-SQUARE ADJUSTED =    .2991
 VARIANCE OF THE ESTIMATE-SIGMA**2 =   46.853
 STANDARD ERROR OF THE ESTIMATE-SIGMA =   6.8449
 SUM OF SQUARED ERRORS-SSE=   1780.4
 MEAN OF DEPENDENT VARIABLE =   23.595
 LOG OF THE LIKELIHOOD FUNCTION = -132.672

 VARIABLE   ESTIMATED  STANDARD   T-RATIO        PARTIAL STANDARDIZED ELASTICITY
   NAME    COEFFICIENT   ERROR      38 DF   P-VALUE CORR. COEFFICIENT  AT MEANS
 INCOME     .23225      .5529E-01   4.200      .000  .563      .5631      .6871
 CONSTANT   7.3832      4.008       1.842      .073  .286      .0000      .3129
|_STOP
```

You will see the correspondence between many of the regression statistics in both systems. For a formal description of the reported quantities see section 8.

5. Options

Options are to be specified in a text file called `ls.opt` which should be located in the current directory (or the *project directory* in case you run GAMS from the IDE under Windows).

To signal the solver to read an option file, you'll need to specify *model*.`optfile = 1;` as in the following example:

```
option lp=ls;
model m /all/;
m.optfile=1;
solve m minimizing sse using lp;
```

It is possible to let GAMS write the option file from within a model. This allows you to make the `.gms` file self-contained. Here is an example:

```
$onecho > ls.opt
add_constant_term 0
$offecho
```

The following options are recognized:

maxn i**:**

Maximum number of cases or observations. This is the number of rows (not counting the dummy objective). When the number of rows is very large, this is probably not a regression problem but a generic LP model. To protect against those, we don't accept models with an enormous number of rows.
(Default = 1000)

maxp i**:**

Maximum number of coefficients to estimate. This is the number of columns or variables (not counting the dummy objective variable). When the number of variables is very large, this is probably not a regression problem but

option	description
0	Don't add constant term
1	Add constant term
2	Automatic (default)

Table 2. Option add_constant_term

a generic LP model. To protect against those, we don't accept models with an enormous number of columns.
(Default = 25)

add_constant_term i**:**
A summary of the allowed values is reproduced in table 2. If this number is zero, no constant term or intercept will be added to the problem. If this option is one, then always a constant term will be added. If this option is two, the algorithm will add a constant term only if there is no data column with all ones in the matrix. In this automatic mode, if the user already specified an explicit intercept in the problem, no additional constant term will be added. As the default is two, you will need to provide an option `add_constant_term 0` in case you want to solve a regression problem without an intercept. For an example see the models `noint1` and `noint2` in section 11.
(Default = 2)

gdx_file_name s**:**
Name of the GDX file where results are saved.
(Default = *ls.gdx*)

6. Linear Least Squares

The calculation of β in the least-squares optimization problem

$$\min_{\beta} ||y - X\beta|| \tag{6}$$

is a well-studied problem [3, 28]. A well-known numerically stable method often used is QR decomposition [17], where a matrix A is written as:

$$A = QR = Q \begin{pmatrix} \Gamma \\ 0 \end{pmatrix} \tag{7}$$

where Q is an orthogonal matrix ($Q^TQ = I$) and Γ is upper-triangular. If this decomposition is applied to X we can write:

$$\begin{aligned} \beta &= (X^TX)^{-1}X^Ty \\ &= (R^TQ^TQR)^{-1}(QR)^Ty \\ &= (R^TR)^{-1}R^TQ^Ty \\ &= R^{-1}Q^Ty \end{aligned} \tag{8}$$

This can be evaluated in two steps: form $\gamma = Q^Ty$ and solve $R\beta = \gamma$ by back-substitution.

This method does not need the calculation of the normal equations, i.e. (X^TX) which is known to be sensitive to round-off errors. An example of this is shown in section 11.2. An other popular method that also does not need this step is singular value decomposition.

We used routine DGEQRF from standard LAPACK [1] to calulate the QR decomposition. Routines DORMQR and DTRTRS are used to solve the system. It is noted that improved versions exist [13, 14]. To calculate $(X^TX)^{-1}$ we called DPOTRI to calculate the inverse given the triangular matrix R, using

$$(X^TX)^{-1} = (R^TR)^{-1} \tag{9}$$

The diagonal of the hat matrix $H = X(X^TX)^{-1}X^T$ can be extracted from the matrix Q: $H = QQ^T$. The calculation of the diagonal elements can be efficiently implemented by $h_{i,i} = \sum_{j=1}^{p} q_{i,j}^2$[30]. The Q matrix is extracted from the QR decomposition by LAPACK routine DGORQR.

7. Nonlinear models

The LS solver can only handle *linear* regression models. For nonlinear problems an alternative solver is available [25].

Many models that look non-linear can actually be reformulated into linear models. Firstly, all models that are nonlinear in X but linear in β are just linear from a regression point of view. E.g. a model like:

$$y = b_0 + b_1x + b_2x^2 + b_3x^3 + b_4x^4 + b_5x^5 \tag{10}$$

taken from the `wampler` data sets[40] from the NIST site `http://www.itl.nist.gov/div898/strd/lls/lls.shtml` is a polynomial problem but linear in the coefficients to estimate $b_0, ..., b_5$. See section 11.5.

Some models can be linearized by taking logarithms. E.g.

$$y = ae^{bx} \tag{11}$$

can be transformed to

$$\ln y = \ln a + bx \tag{12}$$

To be precise: this implies we used a multiplicative error:

$$y = ae^{bx}e^{\varepsilon} \tag{13}$$

A Cobb-Douglas production function of the form

$$Y = \gamma K^{\alpha} L^{\beta} \tag{14}$$

results in a linear model when taking logarithms:

$$\ln Y = \ln \gamma + \alpha \ln K + \beta \ln L \tag{15}$$

A hyperbolic relationship

$$y = \frac{x}{a + bx} \tag{16}$$

can be linearized as:

$$\frac{1}{y} = b + a\frac{1}{x} \tag{17}$$

8. Statistics

The output produced is similar to the `lm` summary output in the R package [16, 39, 38, 9, 32].

The following statistics are calculated:

```
        Parameter      Estimate    Std. Error       t value      Pr(>|t|)
         constant   0.73832E+01   0.40084E+01   0.18420E+01   0.73296E-01 .
           income   0.23225E+00   0.55293E-01   0.42004E+01   0.15514E-03 ***
     ---
     Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Estimation statistics:
Cases: 40  Parameters: 2  Residual sum of squares:   0.17804E+04
Residual standard error:   0.68449E+01   on  38 degrees of freedom
Multiple R-squared:   0.31708E+00  Adjusted R-squared:   0.29911E+00
F statistic:   0.17643E+02 on 1 and 38 DF, p-value:   0.15514E-03
```

Each estimate is accompanied by its standard error, which is given by:

$$\text{SE} = \hat{\sigma}^2 \operatorname{diag}(X^T X)^{-1} \tag{18}$$

where

$$\begin{aligned} \hat{\sigma} &= \sqrt{\frac{\text{RSS}}{n-p}} \\ &= \sqrt{\frac{\sum_{i=1}^{n} \varepsilon_i^2}{n-p}} \end{aligned} \tag{19}$$

where n is the number of cases or observations and p is the number of coefficients to estimate[2]. I.e. the standard errors are the diagonal elements of the variance-covariance matrix. The complete variance-covariance matrix $\hat{\sigma}^2(X^T X)^{-1}$ is exported to the GDX file in case you need access to it.

The test statistic or t-values are calculated as:

$$t_i = \frac{\beta_i}{\text{SE}_i} \tag{20}$$

i.e. the estimates divided by their standard error.

The t values need to be compared to the Student's t distribution. We do this for you and produce socalled p-values. These values give probabilities for the two sided test H_0: $b_i = 0$ against H_1: $b_i \neq 0$. The formal calculation is done as:

$$p\text{-value} = \operatorname{tdist}(|t_i|, n-p, 2) \tag{21}$$

Often a coefficient is called significant if the p value is ≤ 0.05. The final column forms a simple 'bar chart' for the significance levels. A significant coefficient (i.e. p value ≤ 0.005) is marked with one or more stars.

The value $\text{RSS} = \sum \varepsilon_i$ is reproduced under Residual sum of squares. The residual standard error is the value σ defined above.

The quantity R^2 is defined by

$$R^2 = \begin{cases} 1 - \frac{\text{RSS}}{y^T y - n\bar{y}^2} & \text{if constant term is present} \\ 1 - \frac{\text{RSS}}{y^T y} & \text{without constant term} \end{cases} \tag{22}$$

[2]Some authors use the denominator $n-p-1$ instead of $n-p$. Of course this can lead to small differences when n is small compared to p.

This is also sometimes written as $R^2 = 1 - \text{RSS}/\text{TSS}$ where TSS stands for total sum of squares.This value is a goodness-of-fit measure between 0 and 1 with 1 for a perfect fit and a zero indicating no correlation whatsoever.

The adjusted R^2 compensates for the degrees of freedom in the model and makes complex models somewhat less attractive. The definition is:

$$\text{adjusted}R^2 = 1 - \frac{n-1}{n-p}(1 - R^2) \tag{23}$$

Finally the F-statistic is a statistic for testing the null-hypothesis H_0: $b_1 = b_2 = \cdots = 0$. It is defined by

$$F = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{y})^2/(p-1)}{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2/(p-1)} \tag{24}$$

This statistic follows an F distribution with $p-1$ and $n-p$ degrees of freedom. The p-value calculates the probability of seeing the reported F value when the null-hypothesis H_0: $b_1 = b_2 = \cdots = 0$ is true.

The Student t distribution is calculated using an implementation of the incomplete beta function from [10, 4]. The F distribution function is calculated via a chi-square distribution function which is based on the incomplete gamma function from [33]. These functions are also used in GAMS, see [24].

9. GDX output

The solver will write a GDX file named *ls.gdx* by default (the name can be changed using an option, see section 5). This GDX file will contain all of the summary statistics and in addition the variance-covariance matrix.

The content of the GDX file looks like:

```
C:\projects\ls>gdxdump ls.gdx symbols
*  GDX dump of ls.gdx
*  Library in use : C:\PROGRA~1\GAMS23.3
*  Library version: GDX Library      Nov  1, 2009 23.3.3 WEX 14596.15043 WEI x86_64/MS Windows
*  File version   : GDX Library      Nov  1, 2009 23.3.3 WIN 14596.15043 VIS x86/MS Windows
*  Producer       : ls.f90
*  File format    :    7
*  Compression    :    0
*  Symbols        :   16
*  Unique Elements:   29
   Symbol   Dim Type  Explanatory text
 1 confint    3  Par  Confidence intervals
 2 covar      2  Par  Variance-covariance matrix
 3 df         0  Par  Degrees of freedom
 4 estimate   1  Par  Estimated coefficients
 5 fitted     1  Par  Fitted values for dependent variable
 6 hat        1  Par  Diagonal of hat matrix
 7 pval       1  Par  p values
 8 r2         0  Par  R Squared
 9 resid      1  Par  Residuals
10 resvar     0  Par  Residual variance
11 rss        0  Par  Residual sum of squares
12 se         1  Par  Standard errors
13 sigma      0  Par  Standard error
14 stdres     1  Par  Standardized residuals
15 studres    1  Par  Studentized residuals
16 tval       1  Par  t values
C:\projects\ls>
```

Here follows a description for each of the items:

confint:

Confidence intervals for the estimates $\hat{\beta}$. The $1-\alpha\%$ confidence interval for estimate $\hat{\beta}_i$ is given by:

$$[\hat{\beta}_i - \text{SE}_i t_{n-p;\frac{\alpha}{2}}, \hat{\beta}_i + \text{SE}_i t_{n-p;\frac{\alpha}{2}}] \tag{25}$$

where $t_{n-p;\frac{\alpha}{2}}$ indicates the critical value for the Student's t distribution. To calculate these we use the algorithm from [21].

The confidence intervals are given for different α's. For a model that demonstrates how the confidence intervals can be retrieved see section 11.12.4.

covar:
The variance-covariance matrix. The indices are composed from the variable names in the model. An example of how to read the variance-covariance matrix is shown in section 11.12.1.

df:
Degrees of freedom: $\text{df} = n - p$ (i.e. the number of observations minus the number of parameters to estimate).

estimate:
The vector (of length p) of estimates $\hat{b}$. These are the same as returned in the solution.

fitted:
A vector of length n with the predicted values for $\hat{y} = X\hat{\beta}$.

hat:
The diagonal of the Hat-matrix: $H = X(X^TX)^{-1}X^T$.

pval:
A vector of length p with p-values given by (21).

r2:
R^2 as defined by (22).

resid:
A vector of length n with the residuals $\hat{\varepsilon} = y - \hat{y} = y - X\hat{\beta}$.

resvar:
The residual variance $\frac{\text{RSS}}{\text{df}} = \frac{\text{RSS}}{\text{n-p}} = \hat{\sigma}^2$.

rss:
The residual sum of squares $\text{RSS} = \sum_{i=1}^n \hat{\varepsilon}_i^2$

se:
The standard errors, vector of length p as defined by (18).

sigma:
Standard error of the regression model $\hat{\sigma}$ (19).

stdres:
Standardized residuals:

$$r_i = \frac{\hat{\varepsilon}_i}{\hat{\sigma}\sqrt{1 - h_{i,i}}}$$

where H is the hat-matrix: $H = X(X^TX)^{-1}X^T$. This quantity is also known as *internally studentized residuals*.

studres:
Externally studentized residuals[11]:

$$r_i = \frac{\hat{\varepsilon}_i}{s_{(i)}\sqrt{1 - h_{i,i}}}$$

where H is the hat-matrix: $H = X(X^TX)^{-1}X^T$ and $s_{(i)}$ is an estimate of σ with the i-th residual removed. This means:

$$s_{(i)}^2 = \frac{(n-p)\hat{\sigma}^2 - \hat{\varepsilon}_i^2/(1-h_{i,i})}{n-p-1}$$

tval:
The t values, a vector of length p, as defined in (20).

10. Plotting

It is often desirable to get a better understanding of the fit using graphical tools such as scatter plots. Typical plots are scatter plots to assess the relation between the independent and dependent variables. Another interesting post-regression graph is to plot residuals to see if they are approximately normally distributed.

There are multiple ways to plot data. We will show two approaches: using the **gnuplot** package and plotting using the IDE built-in charting facilities.

10.1. Gnuplot scatter plots.

Gnuplot is a popular charting package among GAMS users. It can be downloaded from `http://www.gnuplot.info/`. A convenient way to run it from GAMS is to let it write a PNG file and then call a viewer associated with PNG files to display it. For the **pontius** model in section 11.3, we could use the following code:

```
*
* plot the results
*
file pltdat /pontius.dat/;
loop(i,
    put pltdat data(i,'x'):8:2,data(i,'y'):8:2/;
);
putclose;

file plt /pontius.plt/;
putclose plt,
   'b0=',b0.l:0:16/
   'b1=',b1.l:0:16/
   'b2=',b2.l:0:16/
   'fit(x)=b0+b1*x+b2*(x**2)'/
   'set term png'/
   'set output "pontius.png"'/
   'plot "pontius.dat",fit(x)'/;

execute '=wgnuplot.exe pontius.plt';
execute '=shellexecute pontius.png';
```

In the first part we write a data file with our data points (x, y). In the second part we write a command file `pontius.plt` for use with Gnuplot. This file will look like:

```
b0=0.000673565789474
b1=0.000000732059160
b2=0.000000000000000
fit(x)=b0+b1*x+b2*(x**2)
set term png
set output "pontius.png"
plot "pontius.dat",fit(x)
```

The commands start with defining our fitted function $f(x) = b_0 + b_1x + b_2x^2$ where we substitute the estimates for b_0, b_1 and b_2. Then we tell Gnuplot to generate a PNG file. The plot command instructs Gnuplot to plot both the data points and the fitted function. Finally we call gnuplot followed by a call to `shellexecute` which will call the program associated with PNG files.

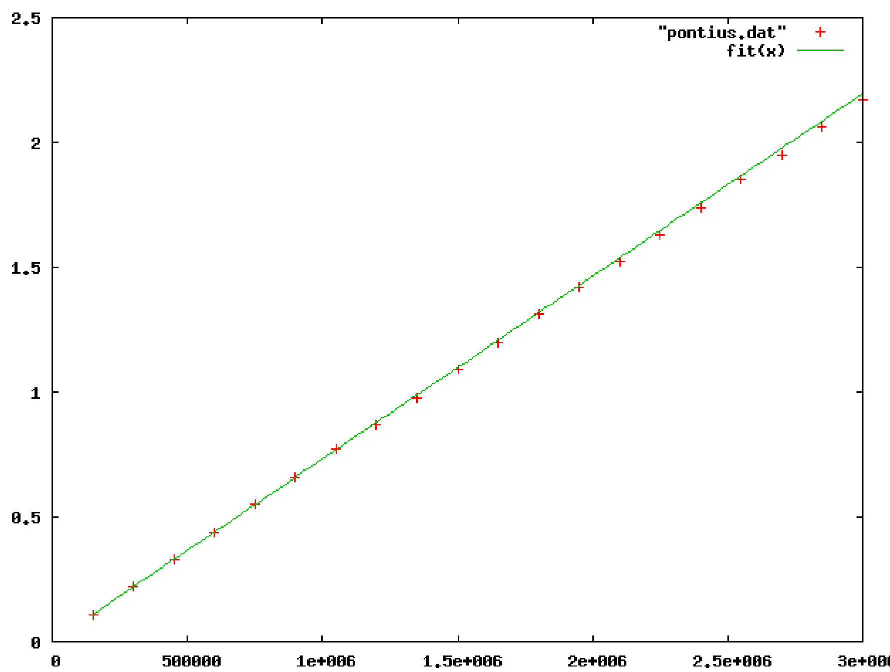

FIGURE 2. Gnuplot scatter plot

The advantage of writing a PNG file is that this format can be imported and used in many environments such as MS Word, HTML pages etc.

10.2. **Gnuplot residual plots.** It is not very difficult to produce a plot of the residuals:

```
*
* plot the results
*
file pltdat /pontius.dat/;
loop(i,
    put pltdat data(i,'x'):8:2,fit.l(i):8:4/;
);
putclose;

file plt /pontius.plt/;
putclose plt,
   'set term png'/
   'set output "pontiusres.png"'/
   'plot "pontius.dat"'/;

execute '=wgnuplot.exe pontius.plt';
execute '=shellexecute pontiusres.png';
```

The residuals $\hat{\varepsilon} = y - \hat{y} = y - X\hat{\beta}$ are stored in the row level by the LS solver. This makes it easy to plot them, by writing the values `fit.l(i)` where `fit` is the name of the equations describing the linear statistical model:

```
equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= b0 + b1*data(i,'x') + b2*sqr(data(i,'x'));
```

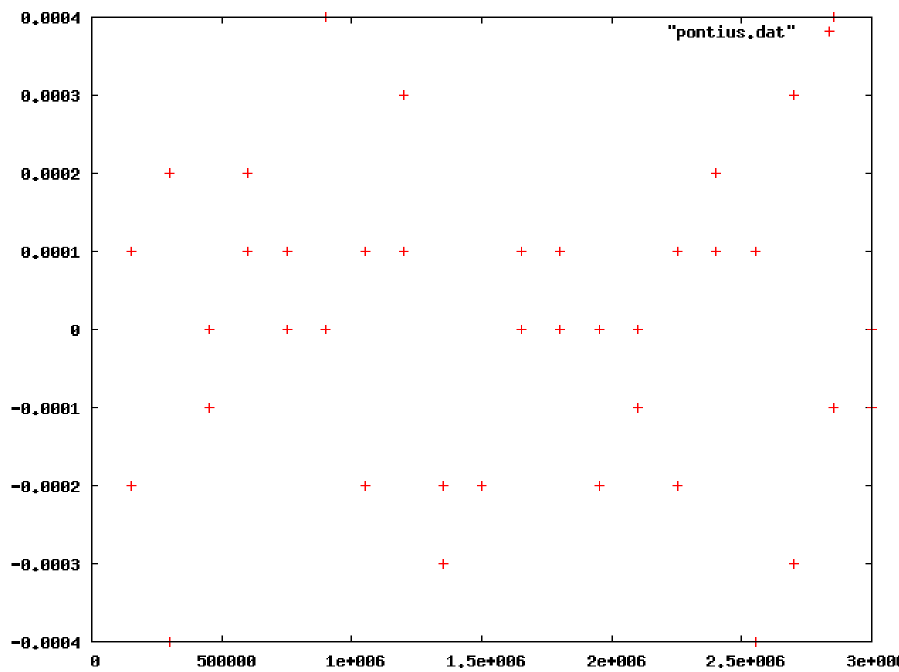

FIGURE 3. Gnuplot residual plot

10.3. **IDE Charting scatter plots.** The GAMS IDE has built-in charting facilities. In many cases IDE charts are created interactively as follows:

- Create a GDX file with the data to plot. Make sure zero's are exported as EPS as they may get lost otherwise[3].
- Open the GDX in the IDE.
- Click right-button and select Graph.
- Select the chart type.

It is possible to script this. Below is code that produces a scatter plot and a fitted line for the `filip` model (see section 11.4).

```
*
* plot results
*
* first we need to make sure x comes before y.
* we had before declared 'y' before 'x' so we introduce
* 'x0' and 'y0' where we make sure 'x0' comes first.
```

[3]This is because GDX files only store nonzero values, similar to the sparse data structures used in GAMS

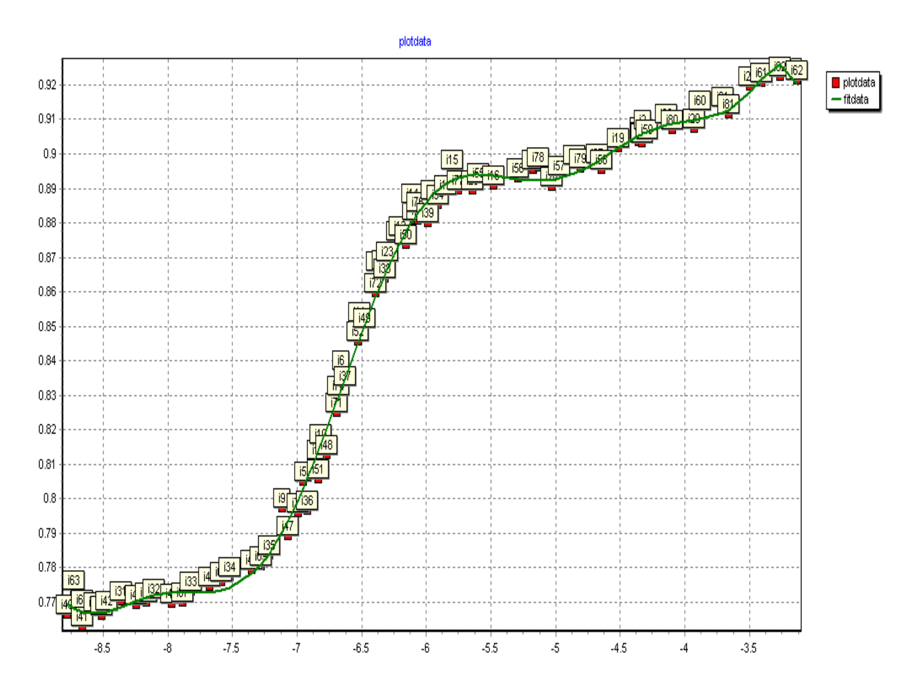

FIGURE 4. GAMS IDE scatter plot

```
* if you load plotdata in the gdxviewer you will see that
* indeed 'x0','y0' are ordered correctly. If this step
* was not performed, we would have seen an inverted
* graph.
*

parameter plotdata(i,*);
plotdata(i,'x0') = EPS+data(i,'x');
plotdata(i,'y0') = EPS+data(i,'y');

set k/point1*point200/;
scalar minx, maxx, stepx;
minx = smin(i,data(i,'x'));
maxx = smax(i,data(i,'x'));
stepx = (maxx-minx)/(card(k)-1);
parameter xfit(k), yfit(k);
xfit(k) = minx+stepx*(ord(k)-1);
yfit(k) = sum(j, b.l(j)*power(xfit(k),v(j)));
parameter fitdata(k,*);
fitdata(k,'x0') =  EPS+xfit(k);
fitdata(k,'y0') =  EPS+yfit(k);


execute_unload 'chartdata.gdx',plotdata,fitdata;

$onecho > filip_gch.gch
[CHART]
VERID=GAMSIDE Chart(s) V1
GDXFILE=chartdata.gdx
TITLE=plotdata

[SERIES1]
SYMBOL=plotdata
TYPE=scatter2d
```

```
[SERIES2]
SYMBOL=fitdata
TYPE=function
$offecho


execute '=idecmds FileOpen filip_gch.gch'
```

There are a number of non-trivial issues addressed in this code. First the ordering of elements x and y is such that y comes before x. This will confuse the charting facility. We force a different ordering by introducing new elements $x0$ and $y0$. The expressions involving EPS are used to force zero's to be exported as EPS. The file `filip_gch.gch` contains instructions for the charting tool. Finally we call `idecmds` to signal the IDE to create a chart.

Using the *Edit* button in the chart window it is possible to remove the marks that accompany the data points.

Note: the corresponding Gnuplot graph is shown in figure 6.

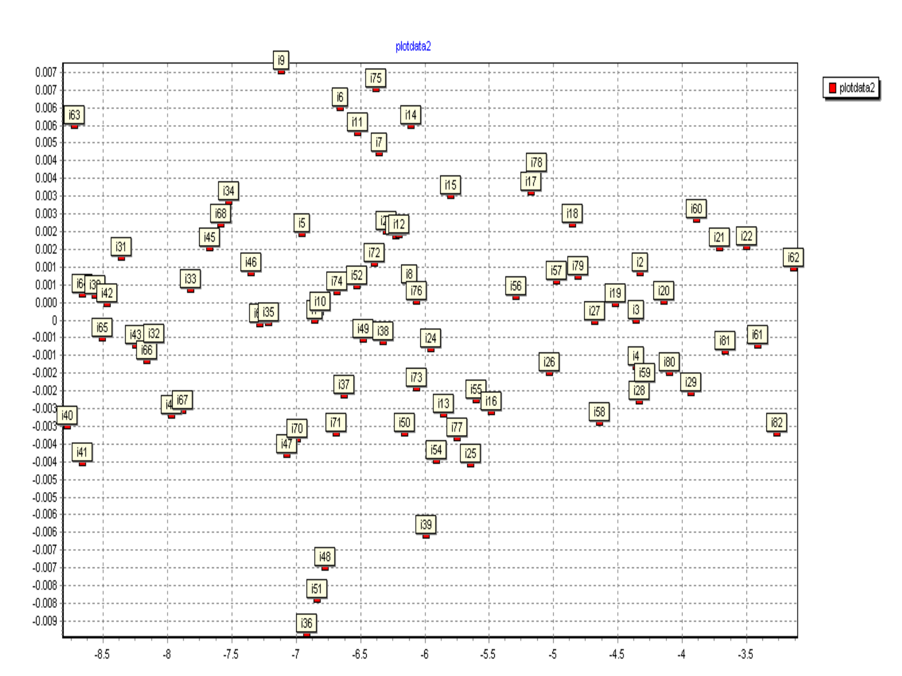

FIGURE 5. GAMS IDE scatter plot

10.4. **IDE Charting residual plots.** The IDE charting version of producing a residual plot is as follows:

```
*
* plot results
*

parameter plotdata2(i,*);
plotdata2(i,'x0') = data(i,'x');
plotdata2(i,'fit0') = fit.l(i);
```

```
execute_unload 'chartdata2.gdx',plotdata2;

$onecho > filip_gch2.gch
[CHART]
VERID=GAMSIDE Chart(s) V1
GDXFILE=chartdata2.gdx
TITLE=plotdata2

[SERIES1]
SYMBOL=plotdata2
TYPE=scatter2d

$offecho

execute '=idecmds FileOpen filip_gch2.gch'
```

Using the *Edit* button in the chart window it is possible to remove the marks that accompany the data points.

11. Examples

In this section we present a number of example models. A large fraction originates from the NIST benchmark cite `http://www.itl.nist.gov/div898/strd/lls/lls.shtml`. This is a fairly well-known test set for statistical software, see e.g. [7].

11.1. **Norris.** This is a simple regression model:

$$y = b_0 + b_1 x \tag{26}$$

from the NIST problem set.

11.1.1. *Model norris.gms.* [4]

```
$ontext

  Linear Least Squares Regression

  NIST test data

  Erwin kalvelagen, dec 2004

  Reference:
      http://www.itl.nist.gov/div898/strd/lls/lls.shtml

      Norris, J., NIST.
      Calibration of Ozone Monitors.


  Model:       Linear Class
               2 Parameters (B0,B1)

               y = B0 + B1*x + e



               Certified Regression Statistics

                                           Standard Deviation
     Parameter          Estimate              of Estimate

        B0        -0.262323073774029      0.232818234301152
        B1         1.00211681802045       0.429796848199937E-03

     Residual
     Standard Deviation   0.884796396144373
```

[4] `www.amsterdamoptimization.com/models/regression/norris.gms`

```
     R-Squared            0.999993745883712


               Certified Analysis of Variance Table

Source of Degrees of    Sums of             Mean
Variation  Freedom      Squares            Squares           F Statistic

Regression    1     4255954.13232369   4255954.13232369   5436385.54079785
Residual     34     26.6173985294224   0.782864662630069

$offtext


set i 'cases' /i1*i36/;

table data(i,*)
            y          x
  i1       0.1        0.2
  i2     338.8      337.4
  i3     118.1      118.2
  i4     888.0      884.6
  i5       9.2       10.1
  i6     228.1      226.5
  i7     668.5      666.3
  i8     998.5      996.3
  i9     449.1      448.6
  i10    778.9      777.0
  i11    559.2      558.2
  i12      0.3        0.4
  i13      0.1        0.6
  i14    778.1      775.5
  i15    668.8      666.9
  i16    339.3      338.0
  i17    448.9      447.5
  i18     10.8       11.6
  i19    557.7      556.0
  i20    228.3      228.1
  i21    998.0      995.8
  i22    888.8      887.6
  i23    119.6      120.2
  i24      0.3        0.3
  i25      0.6        0.3
  i26    557.6      556.8
  i27    339.3      339.1
  i28    888.0      887.2
  i29    998.5      999.0
  i30    778.9      779.0
  i31     10.2       11.1
  i32    117.6      118.3
  i33    228.9      229.2
  i34    668.4      669.1
  i35    449.2      448.9
  i36      0.2        0.5
;

variables
   b0  'constant term'
   b1
   sse 'sum of squared errors'
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= b0 + b1*data(i,'x');

option lp = ls;
model leastsq /fit,sumsq/;
```

Employment	GNP deflator	GNP	Unemployment	Armed Forces	Population	Year
60323	83.0	234289	2356	1590	107608	1947
61122	88.5	259426	2325	1456	108632	1948
60171	88.2	258054	3682	1616	109773	1949
61187	89.5	284599	3351	1650	110929	1950
63221	96.2	328975	2099	3099	112075	1951
63639	98.1	346999	1932	3594	113270	1952
64989	99.0	365385	1870	3547	115094	1953
63761	100.0	363112	3578	3350	116219	1954
66019	101.2	397469	2904	3048	117388	1955
67857	104.6	419180	2822	2857	118734	1956
68169	108.4	442769	2936	2798	120445	1957
66513	110.8	444546	4681	2637	121950	1958
68655	112.6	482704	3813	2552	123366	1959
69564	114.2	502601	3931	2514	125368	1960
69331	115.7	518173	4806	2572	127852	1961
70551	116.9	554894	4007	2827	130081	1962

TABLE 3. Longley dataset

```
solve leastsq using lp minimizing sse;
option decimals=8;
display b0.l,b1.l;

scalar B0cert /  -0.262323073774029  /;
scalar B1cert /   1.00211681802045   /;

scalar err "Sum of squared errors in estimates";
err = sqr(b0.l-B0cert) + sqr(b1.l-B1cert);
display err;
abort$(err>0.0001) "Solution not accurate";
```

11.2. **Longley.** A famous test problem for OLS is the Longley problem[29, 8]. The problem is quite small, see table 3. The NIST web site `http://www.itl.nist.gov/div898/strd/lls/lls.shtml` gives certified solutions for this problem.

11.2.1. *Model longley.gms.* [5]

```
$ontext

  Longley Linear Least Squares benchmark problem

  Erwin Kalvelagen, nov 2004


  References:
       http://www.itl.nist.gov/div898/strd/lls/lls.shtml

       Longley, J. W. (1967).
       An Appraisal of Least Squares Programs for the
       Electronic Computer from the Viewpoint of the User.
       Journal of the American Statistical Association, 62, pp. 819-841.

               Certified Regression Statistics

                                           Standard Deviation
```

[5] www.amsterdamoptimization.com/models/regression/longley.gms

```
    Parameter         Estimate              of Estimate

       B0      -3482258.63459582        890420.383607373
       B1       15.0618722713733        84.9149257747669
       B2      -0.358191792925910E-01   0.334910077722432E-01
       B3      -2.02022980381683        0.488399681651699
       B4      -1.03322686717359        0.214274163161675
       B5      -0.511041056535807E-01   0.226073200069370
       B6       1829.15146461355        455.478499142212

    Residual
    Standard Deviation   304.854073561965

    R-Squared           0.995479004577296


              Certified Analysis of Variance Table

Source of Degrees of     Sums of             Mean
Variation  Freedom       Squares            Squares         F Statistic

Regression    6      184172401.944494   30695400.3240823   330.285339234588
Residual      9      836424.055505915   92936.0061673238

Chazam output:
-------------

Hello/Bonjour/Aloha/Howdy/G Day/Kia Ora/Konnichiwa/Buenos Dias/Nee Hau/Ciao
 Welcome to SHAZAM - Version 10.0 -  JUL 2004 SYSTEM=LINUX    PAR=   781
 |_SAMPLE 1 16
 |_READ  Y X1 X2 X3 X4 X5 X6
    7 VARIABLES AND       16 OBSERVATIONS STARTING AT OBS       1


 |_OLS Y X1 X2 X3 X4 X5 X6

 REQUIRED MEMORY IS PAR=       3 CURRENT PAR=     781
  OLS ESTIMATION
        16 OBSERVATIONS     DEPENDENT VARIABLE= Y
 ...NOTE..SAMPLE RANGE SET TO:      1,     16

  R-SQUARE =   0.9955     R-SQUARE ADJUSTED =   0.9925
 VARIANCE OF THE ESTIMATE-SIGMA**2 =   92936.
 STANDARD ERROR OF THE ESTIMATE-SIGMA =   304.85
 SUM OF SQUARED ERRORS-SSE=  0.83642E+06
 MEAN OF DEPENDENT VARIABLE =   65317.
 LOG OF THE LIKELIHOOD FUNCTION = -109.617

 MODEL SELECTION TESTS - SEE JUDGE ET AL. (1985,P.242)
  AKAIKE (1969) FINAL PREDICTION ERROR - FPE =     0.13360E+06
     (FPE IS ALSO KNOWN AS AMEMIYA PREDICTION CRITERION - PC)
  AKAIKE (1973) INFORMATION CRITERION - LOG AIC =   11.739
  SCHWARZ (1978) CRITERION - LOG SC =               12.077
 MODEL SELECTION TESTS - SEE RAMANATHAN (1998,P.165)
  CRAVEN-WAHBA (1979)
     GENERALIZED CROSS VALIDATION - GCV =          0.16522E+06
  HANNAN AND QUINN (1979) CRITERION =              0.12759E+06
  RICE (1984) CRITERION =                          0.41821E+06
  SHIBATA (1981) CRITERION =                        98018.
  SCHWARZ (1978) CRITERION - SC =                  0.17584E+06
  AKAIKE (1974) INFORMATION CRITERION - AIC =      0.12540E+06

                      ANALYSIS OF VARIANCE - FROM MEAN
                        SS         DF            MS                F
 REGRESSION        0.18417E+09      6.       0.30695E+08          330.285
 ERROR             0.83642E+06      9.        92936.              P-VALUE
 TOTAL             0.18501E+09     15.       0.12334E+08            0.000

                      ANALYSIS OF VARIANCE - FROM ZERO
                        SS         DF            MS                F
 REGRESSION        0.68445E+11      7.       0.97779E+10       105210.860
 ERROR             0.83642E+06      9.        92936.              P-VALUE
```

```
 TOTAL           0.68446E+11     16.       0.42779E+10            0.000


 VARIABLE   ESTIMATED  STANDARD   T-RATIO        PARTIAL STANDARDIZED ELASTICITY
   NAME    COEFFICIENT   ERROR       9 DF    P-VALUE CORR. COEFFICIENT  AT MEANS
 X1          15.062      84.91     0.1774     0.863 0.059     0.0463     0.0234
 X2       -0.35819E-01 0.3349E-01  -1.070     0.313-0.336    -1.0137    -0.2126
 X3         -2.0202     0.4884     -4.136     0.003-0.810    -0.5375    -0.0988
 X4         -1.0332     0.2143     -4.822     0.001-0.849    -0.2047    -0.0412
 X5       -0.51104E-01 0.2261     -0.2261     0.826-0.075    -0.1012    -0.0919
 X6          1829.2      455.5      4.016     0.003 0.801     2.4797    54.7342
 CONSTANT -0.34823E+07 0.8904E+06  -3.911     0.004-0.793     0.0000   -53.3132
 |_STOP


$offtext

set i 'cases' /i1*i16/;
set v 'variables' /empl,const,gnpdefl,gnp,unempl,army,pop,year/;
set indep(v) 'independent variables' /const,gnpdefl,gnp,unempl,army,pop,year/;
set depen(v) 'dependent variables' /empl/;

table data(i,v)
         empl gnpdefl   gnp    unempl    army     pop    year
  i1    60323   83.0   234289   2356     1590    107608  1947
  i2    61122   88.5   259426   2325     1456    108632  1948
  i3    60171   88.2   258054   3682     1616    109773  1949
  i4    61187   89.5   284599   3351     1650    110929  1950
  i5    63221   96.2   328975   2099     3099    112075  1951
  i6    63639   98.1   346999   1932     3594    113270  1952
  i7    64989   99.0   365385   1870     3547    115094  1953
  i8    63761  100.0   363112   3578     3350    116219  1954
  i9    66019  101.2   397469   2904     3048    117388  1955
  i10   67857  104.6   419180   2822     2857    118734  1956
  i11   68169  108.4   442769   2936     2798    120445  1957
  i12   66513  110.8   444546   4681     2637    121950  1958
  i13   68655  112.6   482704   3813     2552    123366  1959
  i14   69564  114.2   502601   3931     2514    125368  1960
  i15   69331  115.7   518173   4806     2572    127852  1961
  i16   70551  116.9   554894   4007     2827    130081  1962
;

data(i,'const') = 1;

alias(indep,j,jj,k);

parameter bcert(indep) 'certified solution' /
   const     -3482258.63459582
   gnpdefl    15.0618722713733
   gnp      -0.358191792925910E-01
   unempl    -2.02022980381683
   army      -1.03322686717359
   pop      -0.511041056535807E-01
   year       1829.15146461355
/;


variables
   b(indep)  'parameters to be estimated'
   sse
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'empl') =e= sum(indep, b(indep)*data(i,indep));

option lp = ls;
model leastsq /fit,sumsq/;
```

```
solve leastsq using lp minimizing sse;
option decimals=8;
display b.l;

scalar err "Sum of squared errors in estimates";
err = sum(indep, sqr(b.l(indep)-bcert(indep)));
display err;
abort$(err>0.0001) "Solution not accurate";
```

The results that are reported are:

```
=======================================================================
  Least Square Solver
  Erwin Kalvelagen, November 2004
=======================================================================

        Parameter      Estimate    Std. Error       t value      Pr(>|t|)
       b('const')  -0.34823E+07  0.89042E+06  -0.39108E+01   0.35604E-02 **
     b('gnpdefl')   0.15062E+02  0.84915E+02   0.17738E+00   0.86314E+00
         b('gnp')  -0.35819E-01  0.33491E-01  -0.10695E+01   0.31268E+00
      b('unempl')  -0.20202E+01  0.48840E+00  -0.41364E+01   0.25351E-02 **
        b('army')  -0.10332E+01  0.21427E+00  -0.48220E+01   0.94437E-03 ***
         b('pop')  -0.51104E-01  0.22607E+00  -0.22605E+00   0.82621E+00
        b('year')   0.18292E+04  0.45548E+03   0.40159E+01   0.30368E-02 **
     ---
     Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Estimation statistics:
Cases: 16  Parameters: 7  Residual sum of squares:   0.83642E+06
Residual standard error:   0.30485E+03   on  9 degrees of freedom
Multiple R-squared:   0.99548E+00  Adjusted R-squared:   0.99247E+00
F statistic:   0.33029E+03 on 6 and 9 DF, p-value:   0.49840E-09
```

When we solve this problem using the statistical system R[32] we get similar results.

```
[erwin@fedora regression]$ R

R : Copyright 2004, The R Foundation for Statistical Computing
Version 1.9.1  (2004-06-21), ISBN 3-900051-00-3

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.

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

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

> employed <- c(60323, 61122, 60171, 61187, 63221, 63639, 64989, 63761, 66019, 67857,
+        68169, 66513, 68655, 69564, 69331, 70551)
> GNPdeflator <- c(83.0, 88.5, 88.2, 89.5, 96.2, 98.1, 99.0, 100.0, 101.2, 104.6,
+         108.4, 110.8, 112.6, 114.2, 115.7, 116.9)
> GNP <- c(234289, 259426, 258054, 284599, 328975, 346999, 365385, 363112, 397469,
+         419180, 442769, 444546, 482704, 502601, 518173, 554894)
> unemployed <- c(2356, 2325, 3682, 3351, 2099, 1932, 1870, 3578, 2904, 2822, 2936, 4681,
+         3813, 3931, 4806, 4007)
> armedForces <- c(1590, 1456, 1616, 1650, 3099, 3594, 3547, 3350, 3048, 2857, 2798, 2637,
+         2552, 2514, 2572, 2827)
> population <- c(107608, 108632, 109773, 110929, 112075, 113270, 115094, 116219, 117388,
+         118734, 120445, 121950, 123366, 125368, 127852, 130081)
> year <- c(1947, 1948, 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1957, 1958,
+         1959, 1960, 1961, 1962)
> fm <- lm(employed ~ GNPdeflator + GNP + unemployed + armedForces + population + year)
> summary(fm)

Call:
lm(formula = employed ~ GNPdeflator + GNP + unemployed + armedForces +
```

```
    population + year)

Residuals:
    Min      1Q  Median      3Q     Max
-410.11 -157.67  -28.16  101.55  455.39

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.482e+06  8.904e+05  -3.911 0.003560 **
GNPdeflator  1.506e+01  8.491e+01   0.177 0.863141
GNP         -3.582e-02  3.349e-02  -1.070 0.312681
unemployed  -2.020e+00  4.884e-01  -4.136 0.002535 **
armedForces -1.033e+00  2.143e-01  -4.822 0.000944 ***
population  -5.110e-02  2.261e-01  -0.226 0.826212
year         1.829e+03  4.555e+02   4.016 0.003037 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 304.9 on 9 degrees of freedom
Multiple R-Squared: 0.9955,     Adjusted R-squared: 0.9925
F-statistic: 330.3 on 6 and 9 DF,  p-value: 4.984e-10

> quit()
Save workspace image? [y/n/c]: n
[erwin@fedora regression]$
```

11.2.2. *Model longley2.gms.* [6]

If we solve this model by calculating $(X^TX)^{-1}$ by solving a linear system:

$$(X^TX)(X^TX)^{-1} = I \tag{27}$$

then we encounter numerical issues. The GAMS model below will illustrate this.

```
$ontext

  Longley Linear Least Squares benchmark problem

  Erwin Kalvelagen, nov 2004



  References:

       http://www.itl.nist.gov/div898/strd/lls/lls.shtml

       Longley, J. W. (1967).
       An Appraisal of Least Squares Programs for the
       Electronic Computer from the Viewpoint of the User.
       Journal of the American Statistical Association, 62, pp. 819-841.

               Certified Regression Statistics

                                           Standard Deviation
     Parameter         Estimate               of Estimate

        B0       -3482258.63459582         890420.383607373
        B1        15.0618722713733         84.9149257747669
        B2       -0.358191792925910E-01    0.334910077722432E-01
        B3       -2.02022980381683         0.488399681651699
        B4       -1.03322686717359         0.214274163161675
        B5       -0.511041056535807E-01    0.226073200069370
        B6        1829.15146461355         455.478499142212

  This formulation forms X'X and then calculates inv(X'X) by solving
  the linear system

    (X'X) * inv(X'X) = I
```

[6] www.amsterdamoptimization.com/models/regression/longley2.gms

```
  This is from a numerical point a bad thing to do.

$offtext

set i 'cases' /i1*i16/;
set v 'variables' /empl,const,gnpdefl,gnp,unempl,army,pop,year/;
set indep(v) 'independent variables' /const,gnpdefl,gnp,unempl,army,pop,year/;
set depen(v) 'dependent variables' /empl/;

table data(i,v)
         empl gnpdefl   gnp    unempl    army     pop    year
  i1    60323   83.0   234289   2356     1590    107608  1947
  i2    61122   88.5   259426   2325     1456    108632  1948
  i3    60171   88.2   258054   3682     1616    109773  1949
  i4    61187   89.5   284599   3351     1650    110929  1950
  i5    63221   96.2   328975   2099     3099    112075  1951
  i6    63639   98.1   346999   1932     3594    113270  1952
  i7    64989   99.0   365385   1870     3547    115094  1953
  i8    63761  100.0   363112   3578     3350    116219  1954
  i9    66019  101.2   397469   2904     3048    117388  1955
  i10   67857  104.6   419180   2822     2857    118734  1956
  i11   68169  108.4   442769   2936     2798    120445  1957
  i12   66513  110.8   444546   4681     2637    121950  1958
  i13   68655  112.6   482704   3813     2552    123366  1959
  i14   69564  114.2   502601   3931     2514    125368  1960
  i15   69331  115.7   518173   4806     2572    127852  1961
  i16   70551  116.9   554894   4007     2827    130081  1962
;

data(i,'const') = 1;

alias(indep,j,jj,k);

*
* form X and y
*
parameter x(i,j),y(i);
x(i,j) = data(i,j);
y(i) = data(i,'empl');


*
* form (X'X) and (X'y)
*
parameter xx(j,jj),xy(j);
xx(j,jj) = sum(i, x(i,j)*x(i,jj));
xy(j) = sum(i, x(i,j)*y(i));

display x,y,xx,xy;

*
* calculate inv(X'X)
*
variables
   dummy       'dummy objective variable'
   invxx(j,jj) 'variable holding the inverse'
;
equations
   edummy        'dummy objective function'
   invert(j,jj)  'calculates inverse matrix'
;
parameter ident(j,jj) 'identity matrix';
ident(j,j)=1;

edummy.. dummy =e= 0;

invert(j,jj)..  sum(k,xx(j,k)*invxx(k,jj)) =e= ident(j,jj);

model inv /all/;
solve inv using lp minimizing dummy;

display invxx.l;
```

```
*
* calculate estimates b = inv(X'X) X'y
*
parameter b(j);
b(j) = sum(k, invxx.l(j,k)*xy(k));

display b;

parameter bcert(indep) 'certified solution' /
   const      -3482258.63459582
   gnpdefl     15.0618722713733
   gnp       -0.358191792925910E-01
   unempl     -2.02022980381683
   army       -1.03322686717359
   pop       -0.511041056535807E-01
   year        1829.15146461355
/;



scalar err "Sum of squared errors in estimates";
err = sum(j, sqr(b(j)-bcert(j)));
display err;
abort$(err>0.0001) "Solution not accurate";
```

The matrix (X^TX) clearly inhibits a scaling problem:

```
----     84 PARAMETER xx

              const      gnpdefl          gnp       unempl         army          pop         year

const        16.000     1626.900 6203175.000    51093.000    41707.000 1878784.000    31272.000
gnpdefl    1626.900   167172.090 6.467006E+8 5289080.100 4293173.700 1.921397E+8 3180539.900
gnp     6203175.000 6.467006E+8 2.55315E+12 2.06505E+10 1.66329E+10 7.38680E+11 1.21312E+10
unempl    51093.000 5289080.100 2.06505E+10 1.762543E+8 1.314528E+8 6.066486E+9 0.999059E+8
army      41707.000 4293173.700 1.66329E+10 1.314528E+8 1.159817E+8 4.923864E+9 8.153707E+7
pop     1878784.000 1.921397E+8 7.38680E+11 6.066486E+9 4.923864E+9 2.21340E+11 3.672577E+9
year      31272.000 3180539.900 1.21312E+10 0.999059E+8 8.153707E+7 3.672577E+9 6.112146E+7
```

As a result the inverse can not be calculated reliably. E.g. Cplex will say:

```
Optimal solution found, but with infeasibilities after unscaling.
Objective :           0.000000

**** REPORT SUMMARY :        0     NONOPT
                            12 INFEASIBLE (INFES)
                    SUM     512984.6187
                    MAX     387698.4375
                    MEAN     42748.7182
                             0  UNBOUNDED
```

The resulting estimates are not accurate:

```
----    115 PARAMETER b

const 65317.000
```

(i.e. all coefficients – except the intercept – are deemed to be zero).

11.3. **Pontius.** This is a simple quadratic model:

$$y = b_0 + b_1x + b_2x^2 \tag{28}$$

As this is linear in the coefficients b_i, we can solve this with a linear regression approach.

11.3.1. *Model pontius.gms.* [7]

```
$ontext

  Linear Least Squares Regression

  NIST test data

  Erwin kalvelagen, dec 2004


  Reference:
     http://www.itl.nist.gov/div898/strd/lls/lls.shtml

     Pontius, P., NIST.
     Load Cell Calibration.


  Model:       Quadratic Class
               3 Parameters (B0,B1,B2)
               y = B0 + B1*x + B2*(x**2)




               Certified Regression Statistics

                                           Standard Deviation
     Parameter          Estimate              of Estimate

        B0      0.673565789473684E-03    0.107938612033077E-03
        B1      0.732059160401003E-06    0.157817399981659E-09
        B2     -0.316081871345029E-14    0.486652849992036E-16

     Residual
     Standard Deviation   0.205177424076185E-03

     R-Squared            0.999999900178537


               Certified Analysis of Variance Table

Source of Degrees of     Sums of               Mean
Variation  Freedom       Squares              Squares             F Statistic

Regression    2     15.6040343244198      7.80201716220991      185330865.995752
Residual     37     0.155761768796992E-05 0.420977753505385E-07


$offtext


set i 'cases' /i1*i40/;

table data(i,*)
            y            x
  i1     .11019        150000
  i2     .21956        300000
  i3     .32949        450000
  i4     .43899        600000
  i5     .54803        750000
  i6     .65694        900000
  i7     .76562       1050000
  i8     .87487       1200000
  i9     .98292       1350000
  i10   1.09146       1500000
  i11   1.20001       1650000
  i12   1.30822       1800000
  i13   1.41599       1950000
  i14   1.52399       2100000
```

[7] www.amsterdamoptimization.com/models/regression/pontius.gms

```
  i15   1.63194       2250000
  i16   1.73947       2400000
  i17   1.84646       2550000
  i18   1.95392       2700000
  i19   2.06128       2850000
  i20   2.16844       3000000
  i21    .11052        150000
  i22    .22018        300000
  i23    .32939        450000
  i24    .43886        600000
  i25    .54798        750000
  i26    .65739        900000
  i27    .76596       1050000
  i28    .87474       1200000
  i29    .98300       1350000
  i30   1.09150       1500000
  i31   1.20004       1650000
  i32   1.30818       1800000
  i33   1.41613       1950000
  i34   1.52408       2100000
  i35   1.63159       2250000
  i36   1.73965       2400000
  i37   1.84696       2550000
  i38   1.95445       2700000
  i39   2.06177       2850000
  i40   2.16829       3000000
;

variables
   b0  'constant term'
   b1  'linear term b1*x'
   b2  'quadratic term b2*x^2'
   sse 'sum of squared errors'
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= b0 + b1*data(i,'x') + b2*sqr(data(i,'x'));

option lp = ls;
model leastsq /fit,sumsq/;
solve leastsq using lp minimizing sse;
option decimals=8;
display b0.l,b1.l;

scalar B0cert /  0.673565789473684E-03  /;
scalar B1cert /  0.732059160401003E-06  /;
scalar B2cert / -0.316081871345029E-14  /;

scalar err "Sum of squared errors in estimates";
err = sqr(b0.l-B0cert) + sqr(b1.l-B1cert) + sqr(b2.l-B2cert);
display err;
abort$(err>0.0001) "Solution not accurate";
```

Scatter and residual plots for this model can be found in section 10.

11.4. **Filip.** This is a model where a polynomial

$$y = b_0 + b_1 x + b_2 x^2 + \cdots + b_9 x^9 + b_{10} x^{10} \tag{29}$$

is fitted. As it is linear in the coefficients b_i this can be estimated with linear regression.

11.4.1. *Model filip.gms.* [8]

[8] www.amsterdamoptimization.com/models/regression/filip.gms

```
$ontext

  Linear Least Squares Regression

  NIST test data

  Erwin kalvelagen, dec 2004

  Reference:
     Filippelli, A., NIST.


  Model:       Polynomial Class
               11 Parameters (B0,B1,...,B10)

               y = B0 + B1*x + B2*(x**2) + ... + B9*(x**9) + B10*(x**10) + e




               Certified Regression Statistics

                                             Standard Deviation
     Parameter         Estimate                 of Estimate

        B0       -1467.48961422980         298.084530995537
        B1       -2772.17959193342         559.779865474950
        B2       -2316.37108160893         466.477572127796
        B3       -1127.97394098372         227.204274477751
        B4       -354.478233703349         71.6478660875927
        B5       -75.1242017393757         15.2897178747400
        B6       -10.8753180355343         2.23691159816033
        B7       -1.06221498588947         0.221624321934227
        B8       -0.670191154593408E-01    0.142363763154724E-01
        B9       -0.246781078275479E-02    0.535617408889821E-03
        B10      -0.402962525080404E-04    0.896632837373868E-05

     Residual
     Standard Deviation   0.334801051324544E-02

     R-Squared            0.996727416185620


               Certified Analysis of Variance Table

Source of Degrees of     Sums of                 Mean
Variation  Freedom       Squares                Squares           F Statistic

Regression   10     0.242391619837339     0.242391619837339E-01 2162.43954511489
Residual     71     0.795851382172941E-03 0.112091743968020E-04


$offtext

set i 'cases' /i1*i82/;

table data(i,*)
              y          x
   i1       0.8116   -6.860120914
   i2       0.9072   -4.324130045
   i3       0.9052   -4.358625055
   i4       0.9039   -4.358426747
   i5       0.8053   -6.955852379
   i6       0.8377   -6.661145254
   i7       0.8667   -6.355462942
   i8       0.8809   -6.118102026
   i9       0.7975   -7.115148017
   i10      0.8162   -6.815308569
   i11      0.8515   -6.519993057
   i12      0.8766   -6.204119983
   i13      0.8885   -5.853871964
```

```
    i14       0.8859   -6.109523091
    i15       0.8959   -5.79832982
    i16       0.8913   -5.482672118
    i17       0.8959   -5.171791386
    i18       0.8971   -4.851705903
    i19       0.9021   -4.517126416
    i20       0.909    -4.143573228
    i21       0.9139   -3.709075441
    i22       0.9199   -3.499489089
    i23       0.8692   -6.300769497
    i24       0.8872   -5.953504836
    i25       0.89     -5.642065153
    i26       0.891    -5.031376979
    i27       0.8977   -4.680685696
    i28       0.9035   -4.329846955
    i29       0.9078   -3.928486195
    i30       0.7675   -8.56735134
    i31       0.7705   -8.363211311
    i32       0.7713   -8.107682739
    i33       0.7736   -7.823908741
    i34       0.7775   -7.522878745
    i35       0.7841   -7.218819279
    i36       0.7971   -6.920818754
    i37       0.8329   -6.628932138
    i38       0.8641   -6.323946875
    i39       0.8804   -5.991399828
    i40       0.7668   -8.781464495
    i41       0.7633   -8.663140179
    i42       0.7678   -8.473531488
    i43       0.7697   -8.247337057
    i44       0.77     -7.971428747
    i45       0.7749   -7.676129393
    i46       0.7796   -7.352812702
    i47       0.7897   -7.072065318
    i48       0.8131   -6.774174009
    i49       0.8498   -6.478861916
    i50       0.8741   -6.159517513
    i51       0.8061   -6.835647144
    i52       0.846    -6.53165267
    i53       0.8751   -6.224098421
    i54       0.8856   -5.910094889
    i55       0.8919   -5.598599459
    i56       0.8934   -5.290645224
    i57       0.894    -4.974284616
    i58       0.8957   -4.64454848
    i59       0.9047   -4.290560426
    i60       0.9129   -3.885055584
    i61       0.9209   -3.408378962
    i62       0.9219   -3.13200249
    i63       0.7739   -8.726767166
    i64       0.7681   -8.66695597
    i65       0.7665   -8.511026475
    i66       0.7703   -8.165388579
    i67       0.7702   -7.886056648
    i68       0.7761   -7.588043762
    i69       0.7809   -7.283412422
    i70       0.7961   -6.995678626
    i71       0.8253   -6.691862621
    i72       0.8602   -6.392544977
    i73       0.8809   -6.067374056
    i74       0.8301   -6.684029655
    i75       0.8664   -6.378719832
    i76       0.8834   -6.065855188
    i77       0.8898   -5.752272167
    i78       0.8964   -5.132414673
    i79       0.8963   -4.811352704
    i80       0.9074   -4.098269308
    i81       0.9119   -3.66174277
    i82       0.9228   -3.2644011
;

set j /j0*j10/;
```

```
set j1(j); j1(j)$(ord(j)>1) = yes;
parameter v(j); v(j) = ord(j)-1;

parameter x(i,j);
x(i,'j0') = 1;
x(i,j1) = power(data(i,'x'),v(j1));
display x;

variables
   b(j)  'coefficients to estimate'
   sse   'sum of squared errors'
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= sum(j, b(j)*x(i,j));

option lp = ls;
model leastsq /fit,sumsq/;
solve leastsq using lp minimizing sse;
option decimals=8;
display b.l;

parameter Bcert /
        j0       -1467.48961422980
        j1       -2772.17959193342
        j2       -2316.37108160893
        j3       -1127.97394098372
        j4       -354.478233703349
        j5       -75.1242017393757
        j6       -10.8753180355343
        j7       -1.06221498588947
        j8       -0.670191154593408E-01
        j9       -0.246781078275479E-02
        j10      -0.402962525080404E-04
/;

scalar err "Sum of squared errors in estimates";
err = sum(j, sqr(bcert(j)-b.l(j)));
display err;
abort$(err>0.0001) "Solution not accurate";
```

11.5. **Wampler.** The `wampler` data sets[40] are also from the NIST site `http://www.itl.nist.gov/div898/strd/lls/lls.shtml`. They are polynomial problems of the form:

$$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4 + b_5 x^5 \qquad (30)$$

11.5.1. *Model wampler1.gms.* [9]

```
$ontext

  Linear Least Squares Regression

  NIST test data

  Erwin kalvelagen, dec 2004

  Reference:
               http://www.itl.nist.gov/div898/strd/lls/lls.shtml
```

[9] `www.amsterdamoptimization.com/models/regression/wampler1.gms`

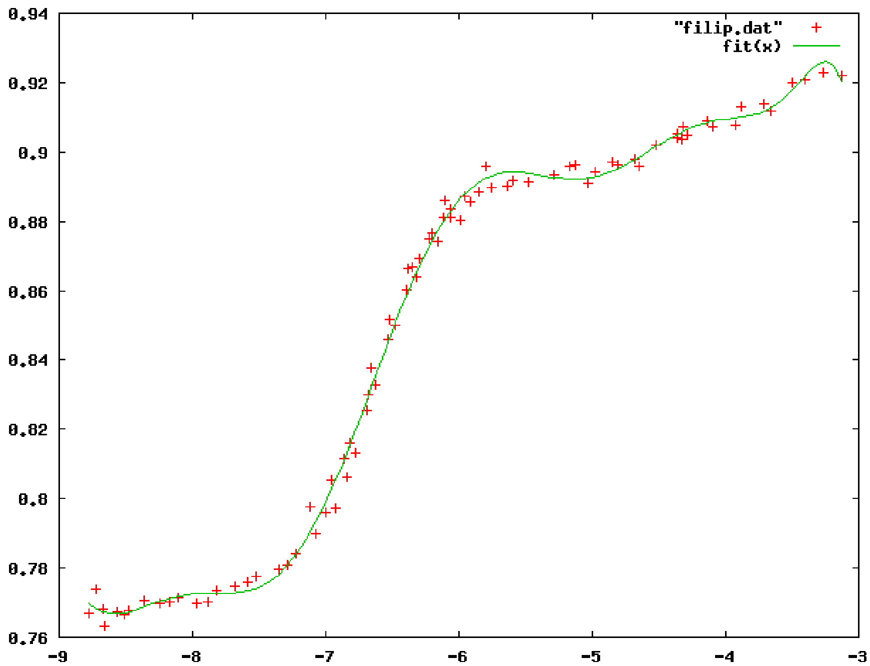

FIGURE 6. Fitting filip.gms

```
              Wampler, R. H. (1970).
              A Report of the Accuracy of Some Widely-Used Least
              Squares Computer Programs.
              Journal of the American Statistical Association, 65, pp. 549-565.


Model:        Polynomial Class
              6 Parameters (B0,B1,...,B5)

              y = B0 + B1*x + B2*(x**2) + B3*(x**3)+ B4*(x**4) + B5*(x**5)

              Certified Regression Statistics

                                          Standard Deviation
     Parameter        Estimate               of Estimate

        B0        1.00000000000000        0.000000000000000
        B1        1.00000000000000        0.000000000000000
        B2        1.00000000000000        0.000000000000000
        B3        1.00000000000000        0.000000000000000
        B4        1.00000000000000        0.000000000000000
        B5        1.00000000000000        0.000000000000000

     Residual
     Standard Deviation    0.000000000000000

     R-Squared             1.00000000000000


              Certified Analysis of Variance Table

Source of Degrees of     Sums of              Mean
Variation  Freedom       Squares             Squares          F Statistic
```

```
Regression    5      18814317208116.7     3762863441623.33       Infinity
Residual     15      0.000000000000000    0.000000000000000



$offtext

set i 'cases' /i1*i21/;

table data(i,*)
                 y     x
   i1            1     0
   i2            6     1
   i3           63     2
   i4          364     3
   i5         1365     4
   i6         3906     5
   i7         9331     6
   i8        19608     7
   i9        37449     8
   i10       66430     9
   i11      111111    10
   i12      177156    11
   i13      271453    12
   i14      402234    13
   i15      579195    14
   i16      813616    15
   i17     1118481    16
   i18     1508598    17
   i19     2000719    18
   i20     2613660    19
   i21     3368421    20
;

set j /j0*j5/;

set j1(j); j1(j)$(ord(j)>1) = yes;
parameter v(j); v(j) = ord(j)-1;

parameter x(i,j);
x(i,'j0') = 1;
x(i,j1) = power(data(i,'x'),v(j1));
display x;

variables
   b(j)  'coefficients to estimate'
   sse   'sum of squared errors'
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= sum(j, b(j)*x(i,j));

option lp = ls;
model leastsq /fit,sumsq/;
solve leastsq using lp minimizing sse;
option decimals=8;
display b.l;

parameter Bcert(j);
BCert(j) = 1;

scalar err "Sum of squared errors in estimates";
err = sum(j, sqr(bcert(j)-b.l(j)));
display err;
abort$(err>0.0001) "Solution not accurate";
```

11.5.2. *Model wampler2.gms.* [10]

```
$ontext

  Linear Least Squares Regression

  NIST test data

  Erwin kalvelagen, dec 2004

  Reference:
               http://www.itl.nist.gov/div898/strd/lls/lls.shtml

               Wampler, R. H. (1970).
               A Report of the Accuracy of Some Widely-Used Least
               Squares Computer Programs.
               Journal of the American Statistical Association, 65, pp. 549-565.


Model:         Polynomial Class
               6 Parameters (B0,B1,...,B5)

               y = B0 + B1*x + B2*(x**2) + B3*(x**3)+ B4*(x**4) + B5*(x**5)

               Certified Regression Statistics

                                            Standard Deviation
     Parameter         Estimate                of Estimate

        B0       1.00000000000000          0.000000000000000
        B1       0.100000000000000         0.000000000000000
        B2       0.100000000000000E-01     0.000000000000000
        B3       0.100000000000000E-02     0.000000000000000
        B4       0.100000000000000E-03     0.000000000000000
        B5       0.100000000000000E-04     0.000000000000000

     Residual
     Standard Deviation   0.000000000000000

     R-Squared            1.00000000000000


               Certified Analysis of Variance Table

Source of Degrees of     Sums of               Mean
Variation  Freedom       Squares              Squares           F Statistic

Regression   5       6602.91858365167     1320.58371673033       Infinity
Residual    15       0.000000000000000    0.000000000000000


$offtext

set i 'cases' /i1*i21/;

table data(i,*)
               y       x
   i1       1.00000    0
   i2       1.11111    1
   i3       1.24992    2
   i4       1.42753    3
   i5       1.65984    4
   i6       1.96875    5
   i7       2.38336    6
   i8       2.94117    7
   i9       3.68928    8
   i10      4.68559    9
   i11      6.00000   10
   i12      7.71561   11
   i13      9.92992   12
```

[10] www.amsterdamoptimization.com/models/regression/wampler2.gms

```
   i14     12.75603   13
   i15     16.32384   14
   i16     20.78125   15
   i17     26.29536   16
   i18     33.05367   17
   i19     41.26528   18
   i20     51.16209   19
   i21     63.00000   20
;

set j /j0*j5/;

set j1(j); j1(j)$(ord(j)>1) = yes;
parameter v(j); v(j) = ord(j)-1;

parameter x(i,j);
x(i,'j0') = 1;
x(i,j1) = power(data(i,'x'),v(j1));
display x;

variables
   b(j)  'coefficients to estimate'
   sse   'sum of squared errors'
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= sum(j, b(j)*x(i,j));

option lp = ls;
model leastsq /fit,sumsq/;
solve leastsq using lp minimizing sse;
option decimals=8;
display b.l;

parameter Bcert(j) /
        j0        1.00000000000000
        j1        0.100000000000000
        j2        0.100000000000000E-01
        j3        0.100000000000000E-02
        j4        0.100000000000000E-03
        j5        0.100000000000000E-04

/;

scalar err "Sum of squared errors in estimates";
err = sum(j, sqr(bcert(j)-b.l(j)));
display err;
abort$(err>0.0001) "Solution not accurate";
```

11.5.3. *Model wampler3.gms.* [11]

```
$ontext

  Linear Least Squares Regression

  NIST test data

  Erwin kalvelagen, dec 2004

  Reference:
               http://www.itl.nist.gov/div898/strd/lls/lls.shtml

               Wampler, R. H. (1970).
               A Report of the Accuracy of Some Widely-Used Least
```

[11] www.amsterdamoptimization.com/models/regression/wampler3.gms

```
              Squares Computer Programs.
              Journal of the American Statistical Association, 65, pp. 549-565.


Model:        Polynomial Class
              6 Parameters (B0,B1,...,B5)

              y = B0 + B1*x + B2*(x**2) + B3*(x**3)+ B4*(x**4) + B5*(x**5)

              Certified Regression Statistics

                                          Standard Deviation
     Parameter          Estimate             of Estimate

        B0       1.00000000000000         2152.32624678170
        B1       1.00000000000000         2363.55173469681
        B2       1.00000000000000         779.343524331583
        B3       1.00000000000000         101.475507550350
        B4       1.00000000000000         5.64566512170752
        B5       1.00000000000000         0.112324854679312

     Residual
     Standard Deviation   2360.14502379268

     R-Squared            0.999995559025820


              Certified Analysis of Variance Table

Source of Degrees of     Sums of             Mean
Variation  Freedom       Squares             Squares           F Statistic

Regression    5      18814317208116.7    3762863441623.33   675524.458240122
Residual     15      83554268.0000000    5570284.53333333


$offtext

set i 'cases' /i1*i21/;

table data(i,*)
                y      x
   i1         760.     0
   i2       -2042.     1
   i3        2111.     2
   i4       -1684.     3
   i5        3888.     4
   i6        1858.     5
   i7       11379.     6
   i8       17560.     7
   i9       39287.     8
   i10      64382.     9
   i11     113159.    10
   i12     175108.    11
   i13     273291.    12
   i14     400186.    13
   i15     581243.    14
   i16     811568.    15
   i17    1121004.    16
   i18    1506550.    17
   i19    2002767.    18
   i20    2611612.    19
   i21    3369180.    20
;

set j /j0*j5/;

set j1(j); j1(j)$(ord(j)>1) = yes;
parameter v(j); v(j) = ord(j)-1;

parameter x(i,j);
x(i,'j0') = 1;
```

```
x(i,j1) = power(data(i,'x'),v(j1));
display x;

variables
   b(j)  'coefficients to estimate'
   sse   'sum of squared errors'
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= sum(j, b(j)*x(i,j));

option lp = ls;
model leastsq /fit,sumsq/;
solve leastsq using lp minimizing sse;
option decimals=8;
display b.l;

parameter Bcert(j);
Bcert(j) = 1;

scalar err "Sum of squared errors in estimates";
err = sum(j, sqr(bcert(j)-b.l(j)));
display err;
abort$(err>0.0001) "Solution not accurate";
```

11.5.4. *Model wampler4.gms.* [12]

```
$ontext

  Linear Least Squares Regression

  NIST test data

  Erwin kalvelagen, dec 2004

  Reference:
               http://www.itl.nist.gov/div898/strd/lls/lls.shtml

               Wampler, R. H. (1970).
               A Report of the Accuracy of Some Widely-Used Least
               Squares Computer Programs.
               Journal of the American Statistical Association, 65, pp. 549-565.


Model:         Polynomial Class
               6 Parameters (B0,B1,...,B5)

               y = B0 + B1*x + B2*(x**2) + B3*(x**3)+ B4*(x**4) + B5*(x**5)

              Certified Regression Statistics

                                          Standard Deviation
     Parameter          Estimate             of Estimate

        B0       1.00000000000000         215232.624678170
        B1       1.00000000000000         236355.173469681
        B2       1.00000000000000         77934.3524331583
        B3       1.00000000000000         10147.5507550350
        B4       1.00000000000000         564.566512170752
        B5       1.00000000000000         11.2324854679312

     Residual
     Standard Deviation   236014.502379268
```

[12] www.amsterdamoptimization.com/models/regression/wampler4.gms

```
     R-Squared            0.957478440825662


               Certified Analysis of Variance Table

Source of Degrees of     Sums of                Mean
Variation  Freedom       Squares               Squares          F Statistic

Regression    5      18814317208116.7      3762863441623.33   67.5524458240122
Residual     15      835542680000.000      55702845333.3333

$offtext

set i 'cases' /i1*i21/;

table data(i,*)
                 y     x
   i1         75901    0
   i2       -204794    1
   i3        204863    2
   i4       -204436    3
   i5        253665    4
   i6       -200894    5
   i7        214131    6
   i8       -185192    7
   i9        221249    8
   i10      -138370    9
   i11       315911   10
   i12       -27644   11
   i13       455253   12
   i14       197434   13
   i15       783995   14
   i16       608816   15
   i17      1370781   16
   i18      1303798   17
   i19      2205519   18
   i20      2408860   19
   i21      3444321   20
;

set j /j0*j5/;

set j1(j); j1(j)$(ord(j)>1) = yes;
parameter v(j); v(j) = ord(j)-1;

parameter x(i,j);
x(i,'j0') = 1;
x(i,j1) = power(data(i,'x'),v(j1));
display x;

variables
   b(j)  'coefficients to estimate'
   sse   'sum of squared errors'
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= sum(j, b(j)*x(i,j));

option lp = ls;
model leastsq /fit,sumsq/;
solve leastsq using lp minimizing sse;
option decimals=8;
display b.l;

parameter Bcert(j);
Bcert(j) = 1;
```

```
scalar err "Sum of squared errors in estimates";
err = sum(j, sqr(bcert(j)-b.l(j)));
display err;
abort$(err>0.0001) "Solution not accurate";
```

11.5.5. *Model wampler5.gms.* [13]

```
$ontext

  Linear Least Squares Regression

  NIST test data

  Erwin kalvelagen, dec 2004

  Reference:
               http://www.itl.nist.gov/div898/strd/lls/lls.shtml


               Wampler, R. H. (1970).
               A Report of the Accuracy of Some Widely-Used Least
               Squares Computer Programs.
               Journal of the American Statistical Association, 65, pp. 549-565.


Model:         Polynomial Class
               6 Parameters (B0,B1,...,B5)

               y = B0 + B1*x + B2*(x**2) + B3*(x**3)+ B4*(x**4) + B5*(x**5)

               Certified Regression Statistics

                                           Standard Deviation
     Parameter          Estimate              of Estimate

        B0       1.00000000000000         21523262.4678170
        B1       1.00000000000000         23635517.3469681
        B2       1.00000000000000         7793435.24331583
        B3       1.00000000000000         1014755.07550350
        B4       1.00000000000000         56456.6512170752
        B5       1.00000000000000         1123.24854679312

     Residual
     Standard Deviation   23601450.2379268

     R-Squared            0.224668921574940E-02


               Certified Analysis of Variance Table

Source of Degrees of    Sums of               Mean
Variation  Freedom      Squares              Squares           F Statistic

Regression    5    18814317208116.7      3762863441623.33 6.7552445824012241E-03
Residual     15    0.835542680000000E+16 557028453333333.

$offtext

set i 'cases' /i1*i21/;

table data(i,*)
                   y     x
   i1        7590001     0
   i2      -20479994     1
   i3       20480063     2
   i4      -20479636     3
   i5       25231365     4
   i6      -20476094     5
   i7       20489331     6
```

[13] www.amsterdamoptimization.com/models/regression/wampler5.gms

```
   i8      -20460392     7
   i9       18417449     8
   i10     -20413570     9
   i11      20591111    10
   i12     -20302844    11
   i13      18651453    12
   i14     -20077766    13
   i15      21059195    14
   i16     -19666384    15
   i17      26348481    16
   i18     -18971402    17
   i19      22480719    18
   i20     -17866340    19
   i21      10958421    20
;


set j /j0*j5/;

set j1(j); j1(j)$(ord(j)>1) = yes;
parameter v(j); v(j) = ord(j)-1;

parameter x(i,j);
x(i,'j0') = 1;
x(i,j1) = power(data(i,'x'),v(j1));
display x;

variables
   b(j)  'coefficients to estimate'
   sse   'sum of squared errors'
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= sum(j, b(j)*x(i,j));

option lp = ls;
model leastsq /fit,sumsq/;
solve leastsq using lp minimizing sse;
option decimals=8;
display b.l;

parameter Bcert(j);
Bcert(j) = 1;

scalar err "Sum of squared errors in estimates";
err = sum(j, sqr(bcert(j)-b.l(j)));
display err;
abort$(err>0.0001) "Solution not accurate";
```

11.6. **Noint.** The `noint` problems are simple regression problems without intercept. They have the form:

$$y = b_1 x \tag{31}$$

The solver will automatically add a constant term if there is no data column with all ones. To prevent this we need to use an option file `ls.opt` with content:

```
add_constant_term 0
```

This option file is actually created from inside the models at compilation time using `$onecho/$offecho` construct.

11.6.1. *Model noint1.gms.* [14]

```
$ontext

  Linear Least Squares Regression

  NIST test data

  Erwin kalvelagen, dec 2004

  Reference:
      http://www.itl.nist.gov/div898/strd/lls/lls.shtml

      Eberhardt, K., NIST.

 Model:         Linear Class
              1 Parameter (B1)

              y = B1*x + e



              Certified Regression Statistics

                                          Standard Deviation
     Parameter          Estimate             of Estimate

        B1          2.07438016528926     0.165289256198347E-01

     Residual
     Standard Deviation   3.56753034006338

     R-Squared            0.999365492298663

              Certified Analysis of Variance Table

Source of Degrees of    Sums of             Mean
Variation  Freedom      Squares            Squares          F Statistic

Regression    1    200457.727272727   200457.727272727   15750.2500000000
Residual     10    127.272727272727   12.7272727272727


$offtext

set i 'cases' /i1*i11/;

table data(i,*)
           y     x
  i1     130    60
  i2     131    61
  i3     132    62
  i4     133    63
  i5     134    64
  i6     135    65
  i7     136    66
  i8     137    67
  i9     138    68
  i10    139    69
  i11    140    70
;

*
* note:no constant term
*
$onecho > ls.opt
add_constant_term 0
$offecho
```

[14] www.amsterdamoptimization.com/models/regression/noint1.gms

```
variables
   b1
   sse 'sum of squared errors'
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= b1*data(i,'x');

option lp = ls;
model leastsq /fit,sumsq/;
leastsq.optfile=1;
solve leastsq using lp minimizing sse;
option decimals=8;
display b1.l;

scalar B1cert /   2.07438016528926   /;

scalar err "Sum of squared errors in estimates";
err = sqr(b1.l-B1cert);
display err;
abort$(err>0.0001) "Solution not accurate";
```

11.6.2. *Model noint2.gms.* [15]

```
$ontext

  Linear Least Squares Regression

  NIST test data

  Erwin kalvelagen, dec 2004

  Reference:
      http://www.itl.nist.gov/div898/strd/lls/lls.shtml

      Eberhardt, K., NIST.

 Model:         Linear Class
               1 Parameter (B1)

               y = B1*x + e



Model:         Linear Class
               1 Parameter (B1)

               y = B1*x + e




               Certified Regression Statistics

                                           Standard Deviation
     Parameter          Estimate              of Estimate

        B1         0.727272727272727    0.420827318078432E-01

     Residual
     Standard Deviation   0.369274472937998
```

[15] www.amsterdamoptimization.com/models/regression/noint2.gms

```
     R-Squared            0.993348115299335


               Certified Analysis of Variance Table

Source of Degrees of    Sums of             Mean
Variation  Freedom      Squares            Squares           F Statistic

Regression   1     40.7272727272727    40.7272727272727   298.6666666666667
Residual     2     0.272727272727273   0.136363636363636



$offtext

set i 'cases' /i1*i3/;

table data(i,*)
         y       x
  i1     3       4
  i2     4       5
  i3     4       6
;

*
* note:no constant term
*
$onecho > ls.opt
add_constant_term 0
$offecho

variables
   b1
   sse 'sum of squared errors'
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'y') =e= b1*data(i,'x');

option lp = ls;
model leastsq /fit,sumsq/;
leastsq.optfile=1;
solve leastsq using lp minimizing sse;
option decimals=8;
display b1.l;

scalar B1cert /  0.727272727272727   /;

scalar err "Sum of squared errors in estimates";
err = sqr(b1.l-B1cert);
display err;
abort$(err>0.0001) "Solution not accurate";
```

11.7. **Klein.** The simultaneous equation model from Klein [26] looks like:

symbol	description
C_t	Private consumption expenditure
P_t	Profits net of business taxes
W_t^p	Wage bill of the private sector
W_t^g	Wage bill of the government sector
I_t	Net private investment
K_t	Stock of private capital goods (at end of the year)
X_t	Gross national product
A_t	An index of the passage of time (1931=zero)
T_t	Business taxes
G_t	Government expenditures plus net exports

Table 4. Symbols in Klein's model

$$
\begin{aligned}
C_t &= \alpha_0 + \alpha_1 P_t + \alpha_2 P_{t-1} + \alpha_3 (W_t^p + W_t^g) + \varepsilon_{1,t} \\
I_t &= \beta_0 + \beta_1 P_t + \beta_2 P_{t-1} + \beta_3 K_{t-1} + \varepsilon_{2,t} \\
W_t^p &= \gamma_0 + \gamma_1 X_t + \gamma_2 X_{t-1} + \gamma_3 A_t + \varepsilon_{3,t} \\
X_t &= C_t + I_t + G_t \\
P_t &= X_t - T_t - W_t^p \\
K_t &= K_{t-1} + I_t
\end{aligned}
\tag{32}
$$

It is well known that direct estimation of the parameters α_j, β_j and γ_j by OLS leads to biased results [36, 19]. A common approach for this type of models is to use two-stage least squares (2SLS).

In this example we will first estimate the three regression equations using OLS. Then we will use a true two-stage approach where we first get estimates of the endogenous variable say y against all exogenous variables x. The additional exogenous variables are often called *instrumental variables*. Then we replace each independent endogenous variable y by its fitted values $\hat{y}$ and run OLS again. The complete model looks like:

11.7.1. *Model klein.gms.* [16]

```
$ontext

   Two-stage least squares (2SLS) of Klein's Model I

   References:
       Theil, H. (1971) Principles of Econometrics, Wiley
       Greene, W. H. (2000) Econometric Analysis, 4th edn, Prentice Hall, New York.
       Klein, L.R. (1950) Economic Fluctuations in the United States 1921-1941, Wiley


SAS results for first equation (Consumption function):


OLS estimation:
==============


/*OLS Estimate*/
proc reg data=klein;
  model C = P Plag W;
```

[16] www.amsterdamoptimization.com/models/regression/klein.gms

```
run;
quit;

The REG Procedure
Model: MODEL1
Dependent Variable: C

                             Analysis of Variance

                                      Sum of           Mean
Source                   DF          Squares         Square    F Value    Pr > F

Model                     3        923.55008      307.85003     292.71    <.0001
Error                    17         17.87945        1.05173
Corrected Total          20        941.42952


Root MSE              1.02554    R-Square     0.9810
Dependent Mean       53.99524    Adj R-Sq     0.9777
Coeff Var             1.89932


                        Parameter Estimates

                     Parameter       Standard
Variable     DF       Estimate          Error    t Value    Pr > |t|

Intercept     1       16.23660        1.30270      12.46      <.0001
P             1        0.19293        0.09121       2.12      0.0495
Plag          1        0.08988        0.09065       0.99      0.3353
W             1        0.79622        0.03994      19.93      <.0001



2SLS procedure on the same equation:
===================================

/* 2sls Estimate*/
proc model data=klein;
      C = cons_c + P_c * P + Plag_c * Plag + W_c*W;
      ENDOGENOUS W P X;
      EXOGENOUS T Wg G;
      fit C  /n2sls  vardef=N;
      instruments G T A Wg Plag Klag Xlag ;
   run;
quit;

The MODEL Procedure

                       Nonlinear 2SLS Summary of Residual Errors

                   DF       DF                                                 Adj
Equation        Model    Error         SSE         MSE    Root MSE    R-Square       R-Sq

C                   4       17     21.9252      1.0441      1.0218      0.9767     0.9726

             Nonlinear 2SLS Parameter Estimates

                              Approx                 Approx
Parameter       Estimate     Std Err    t Value     Pr > |t|

cons_c          16.55476      1.3208      12.53       <.0001
P_c             0.017302      0.1180       0.15       0.8852
Plag_c          0.216234      0.1073       2.02       0.0599
W_c             0.810183      0.0402      20.13       <.0001


Number of Observations     Statistics for System

Used               21    Objective         0.4361
Missing             1    Objective*N       9.1580
```

```
$offtext

set t /1920*1941/;

* keys:
* c    = CONSUMPTION
* p    = PROFITS
* wp   = PRIVATE WAGE BILL
* i    = INVESTMENT
* x    = PRIVATE PRODUCTION
* wg   = GOVT WAGE BILL
* g    = GOVT DEMAND
* t    = TAXES
* k    = CAPITAL STOCK
* wsum = TOTAL WAGE BILL
*

set p /c, p, wp, i, x, wg, g, t, k, wsum/;

table data(t,p)
        c     p     wp    i     x     wg    g     t     k     wsum
1920          12.7              44.9                          182.8
1921  41.9  12.4  25.5 -0.2  45.6  2.7   3.9   7.7  182.6  28.2
1922  45.0  16.9  29.3  1.9  50.1  2.9   3.2   3.9  184.5  32.2
1923  49.2  18.4  34.1  5.2  57.2  2.9   2.8   4.7  189.7  37.0
1924  50.6  19.4  33.9  3.0  57.1  3.1   3.5   3.8  192.7  37.0
1925  52.6  20.1  35.4  5.1  61.0  3.2   3.3   5.5  197.8  38.6
1926  55.1  19.6  37.4  5.6  64.0  3.3   3.3   7.0  203.4  40.7
1927  56.2  19.8  37.9  4.2  64.4  3.6   4.0   6.7  207.6  41.5
1928  57.3  21.1  39.2  3.0  64.5  3.7   4.2   4.2  210.6  42.9
1929  57.8  21.7  41.3  5.1  67.0  4.0   4.1   4.0  215.7  45.3
1930  55.0  15.6  37.9  1.0  61.2  4.2   5.2   7.7  216.7  42.1
1931  50.9  11.4  34.5 -3.4  53.4  4.8   5.9   7.5  213.3  39.3
1932  45.6   7.0  29.0 -6.2  44.3  5.3   4.9   8.3  207.1  34.3
1933  46.5  11.2  28.5 -5.1  45.1  5.6   3.7   5.4  202.0  34.1
1934  48.7  12.3  30.6 -3.0  49.7  6.0   4.0   6.8  199.0  36.6
1935  51.3  14.0  33.2 -1.3  54.4  6.1   4.4   7.2  197.7  39.3
1936  57.7  17.6  36.8  2.1  62.7  7.4   2.9   8.3  199.8  44.2
1937  58.7  17.3  41.0  2.0  65.0  6.7   4.3   6.7  201.8  47.7
1938  57.5  15.3  38.2 -1.9  60.9  7.7   5.3   7.4  199.9  45.9
1939  61.6  19.0  41.6  1.3  69.5  7.8   6.6   8.9  201.2  49.4
1940  65.0  21.1  45.0  3.3  75.7  8.0   7.4   9.6  204.5  53.0
1941  69.7  23.5  53.3  4.9  88.4  8.5  13.8  11.6  209.4  61.8
;
display data;

set t1(t) 'from 1921';
t1(t)$(ord(t)>1) = yes;

*
* calculate A(t)
*
parameter A(t) 'year, 1931=zero';
scalar ord31; ord31 = sum(sameas(t,'1931'),ord(t));
A(t) = ord(t)-ord31;
display A;

*
* calculate lagged variables
*
parameter klag(t),plag(t),xlag(t);
klag(t) = data(t-1,'k');
plag(t) = data(t-1,'p');
xlag(t) = data(t-1,'x');
display klag,plag,xlag;

*-----------------------------------------------------------------------------------------------
* OSL estimation of individual equations
*-----------------------------------------------------------------------------------------------
```

```
variables
   sse
   a0,a1,a2,a3
   b0,b1,b2,b3
   g0,g1,g2,g3
;

equations
    sumsq    'dummy objective'
    cons(t)  'consumption function'
    inv(t)   'investment function'
    labor(t) 'labor equation'
;

sumsq..  sse =e= 0;

cons(t1(t))..  data(t,'c') =e= a0 + a1*data(t,'p') + a2*plag(t) + a3*data(t,'wsum');

inv(t1(t))..   data(t,'i') =e= b0 + b1*data(t,'p') + b2*plag(t) + b3*klag(t);

labor(t1(t)).. data(t,'wsum') =e= g0 + g1*data(t,'x') + g2*xlag(t) + g3*A(t);


option lp=ls;

model cons_ols /sumsq,cons/;
model inv_ols /sumsq,inv/;
model lab_ols /sumsq,labor/;

solve cons_ols minimizing sse using lp;
solve inv_ols minimizing sse using lp;
solve lab_ols minimizing sse using lp;

*-------------------------------------------------------------------------------------------
* 2SLS estimation of individual equations
*
* STAGE 1: Endogenous variables against all exogenous variables. Save fitted values.
*-------------------------------------------------------------------------------------------

variables
   p0,p1,p2,p3,p4,p5,p6,p7
   w0,w1,w2,w3,w4,w5,w6,w7
   x0,x1,x2,x3,x4,x5,x6,x7
;
equations
   Pfit(t)
   Wfit(t)
   Xfit(t)
;
parameters
   Pfitted(t)
   Wfitted(t)
   Xfitted(t)
;

pfit(t1(t)).. data(t,'p') =e= p0 + p1*data(t,'g') + p2*data(t,'t') + p3*A(t)
                            + p4*data(t,'Wg') + p5*plag(t) + p6*klag(t) + p7*xlag(t);

wfit(t1(t)).. data(t,'wsum') =e= w0 + w1*data(t,'g') + w2*data(t,'t') + w3*A(t)
                            + w4*data(t,'Wg') + w5*plag(t) + w6*klag(t) + w7*xlag(t);

xfit(t1(t)).. data(t,'x') =e= x0 + x1*data(t,'g') + x2*data(t,'t') + x3*A(t)
                            + x4*data(t,'Wg') + x5*plag(t) + x6*klag(t) + x7*xlag(t);


model stage1_p /sumsq,pfit/;
model stage1_w /sumsq,wfit/;
model stage1_x /sumsq,xfit/;

solve stage1_p minimizing sse using lp;
execute_load 'ls.gdx',Pfitted=fitted;
display Pfitted;
```

```
solve stage1_w minimizing sse using lp;
execute_load 'ls.gdx',Wfitted=fitted;
display Wfitted;

solve stage1_x minimizing sse using lp;
execute_load 'ls.gdx',Xfitted=fitted;
display Xfitted;


*-------------------------------------------------------------------------------------------
* STAGE 2
*-------------------------------------------------------------------------------------------

variables
   aa0,aa1,aa2,aa3
   bb0,bb1,bb2,bb3
   gg0,gg1,gg2,gg3
;

equations
   cons2(t)
   inv2(t)
   labor2(t)
;

cons2(t1(t))..  data(t,'c') =e= aa0 + aa1*pfitted(t) + aa2*plag(t) + aa3*wfitted(t);

inv2(t1(t))..   data(t,'i') =e= bb0 + bb1*pfitted(t) + bb2*plag(t) + bb3*klag(t);

labor2(t1(t)).. data(t,'wsum') =e= gg0 + gg1*xfitted(t) + gg2*xlag(t) + gg3*A(t);

model stage2_cons /sumsq, cons2/;
model stage2_inv /sumsq, inv2/;
model stage2_lab /sumsq, labor2/;

solve stage2_cons minimizing sse using lp;
solve stage2_inv minimizing sse using lp;
solve stage2_lab minimizing sse using lp;

*-------------------------------------------------------------------------------------------
* display results
*-------------------------------------------------------------------------------------------

display "----- OLS RESULTS ------------------------------------------------------------------",
        a0.l,a1.l,a2.l,a3.l,
        b0.l,b1.l,b2.l,b3.l,
        g0.l,g1.l,g2.l,g3.l,
        "----- 2SLS RESULTS -----------------------------------------------------------------",
        aa0.l,aa1.l,aa2.l,aa3.l,
        bb0.l,bb1.l,bb2.l,bb3.l,
        gg0.l,gg1.l,gg2.l,gg3.l;
```

Note that the fitted values $\hat{y}$ are retrieved from the GDX file that the solver writes. This saves us from calculating these fitted values ourselves.

The resulting estimates are:

```
----    280 ----- OLS RESULTS -----------------------------------------------------------------
            VARIABLE a0.L                  =       16.237
            VARIABLE a1.L                  =        0.193
            VARIABLE a2.L                  =        0.090
            VARIABLE a3.L                  =        0.796
            VARIABLE b0.L                  =       10.126
            VARIABLE b1.L                  =        0.480
            VARIABLE b2.L                  =        0.333
            VARIABLE b3.L                  =       -0.112
            VARIABLE g0.L                  =        7.473
            VARIABLE g1.L                  =        0.469
            VARIABLE g2.L                  =        0.101
            VARIABLE g3.L                  =        0.443
            ----- 2SLS RESULTS ----------------------------------------------------------------
```

```
            VARIABLE aa0.L                =       16.555
            VARIABLE aa1.L                =        0.017
            VARIABLE aa2.L                =        0.216
            VARIABLE aa3.L                =        0.810
            VARIABLE bb0.L                =       20.278
            VARIABLE bb1.L                =        0.150
            VARIABLE bb2.L                =        0.616
            VARIABLE bb3.L                =       -0.158
            VARIABLE gg0.L                =        7.399
            VARIABLE gg1.L                =        0.483
            VARIABLE gg2.L                =        0.087
            VARIABLE gg3.L                =        0.440
```

We have calculated the 2SLS estimators in two stages. In practical statistical software these estimators are calculated directly.

For comparison, the results with Gretl are listed below. They correspond to the estimated values in the GAMS model.

```
gretl version 1.3.0
Current session: 2005/01/05 19:06
# Klein's "Model 1" (1950)
? open klein.gdt

Read datafile /usr/share/gretl/data/misc/klein.gdt
periodicity: 1, maxobs: 22,
observations range: 1920-1941

Listing 10 variables:
  0) const       1) C           2) P           3) Wp          4) I
  5) K1          6) X           7) Wg          8) G           9) T

? genr W = Wp + Wg
Generated vector W (ID 10)
? genr A = t - 1931
Generated vector A (ID 11)
? genr P1 = P(-1)
Generated vector P1 (ID 12)
? genr X1 = X(-1)
Generated vector X1 (ID 13)
# Equation-by-equation OLS
? ols C 0 P P1 W

Model 1: OLS estimates using the 21 observations 1921-1941
Dependent variable: C

      VARIABLE      COEFFICIENT        STDERROR       T STAT   2Prob(t > |T|)

   0)     const        16.2366            1.30270      12.464   < 0.00001 ***
   2)         P         0.192934          0.0912102     2.115    0.049474 **
  12)        P1         0.0898849         0.0906479     0.992    0.335306
  10)         W         0.796219          0.0399439    19.933   < 0.00001 ***

  Mean of dependent variable = 53.9952
  Standard deviation of dep. var. = 6.86087
  Sum of squared residuals = 17.8794
  Standard error of residuals = 1.02554
  Unadjusted R-squared = 0.981008
  Adjusted R-squared = 0.977657
  F-statistic (3, 17) = 292.708 (p-value < 0.00001)
  Durbin-Watson statistic = 1.36747
  First-order autocorrelation coeff. = 0.2463

  MODEL SELECTION STATISTICS

  SGMASQ        1.05173     AIC           1.24618     FPE           1.25206
  HQ            1.30117     SCHWARZ       1.52050     SHIBATA       1.17575
  GCV           1.29920     RICE          1.37534

Excluding the constant, p-value was highest for variable 12 (P1)

? ols I 0 P P1 K1
```

```
Model 2: OLS estimates using the 21 observations 1921-1941
Dependent variable: I

      VARIABLE      COEFFICIENT        STDERROR       T STAT   2Prob(t > |T|)

   0)    const        10.1258            5.46555       1.853    0.081374 *
   2)        P         0.479636          0.0971146     4.939    0.000125 ***
  12)       P1         0.333039          0.100859      3.302    0.004212 ***
   5)       K1        -0.111795          0.0267276    -4.183    0.000624 ***

  Mean of dependent variable = 1.26667
  Standard deviation of dep. var. = 3.55195
  Sum of squared residuals = 17.3227
  Standard error of residuals = 1.00945
  Unadjusted R-squared = 0.931348
  Adjusted R-squared = 0.919233
  F-statistic (3, 17) = 76.8754 (p-value < 0.00001)
  Durbin-Watson statistic = 1.81018
  First-order autocorrelation coeff. = 0.0842508

  MODEL SELECTION STATISTICS

  SGMASQ        1.01898     AIC          1.20737     FPE          1.21307
  HQ            1.26065     SCHWARZ      1.47315     SHIBATA      1.13913
  GCV           1.25874     RICE         1.33252

? ols Wp 0 X X1 A

Model 3: OLS estimates using the 21 observations 1921-1941
Dependent variable: Wp

      VARIABLE      COEFFICIENT        STDERROR       T STAT   2Prob(t > |T|)

   0)    const         1.49704           1.27003       1.179    0.254736
   6)        X         0.439477          0.0324076    13.561   < 0.00001 ***
  13)       X1         0.146090          0.0374231     3.904    0.001142 ***
  11)        A         0.130245          0.0319103     4.082    0.000777 ***

  Mean of dependent variable = 36.3619
  Standard deviation of dep. var. = 6.3044
  Sum of squared residuals = 10.0048
  Standard error of residuals = 0.767147
  Unadjusted R-squared = 0.987414
  Adjusted R-squared = 0.985193
  F-statistic (3, 17) = 444.568 (p-value < 0.00001)
  Durbin-Watson statistic = 1.95843
  First-order autocorrelation coeff. = -0.0833381

  MODEL SELECTION STATISTICS

  SGMASQ       0.588515     AIC         0.697321     FPE         0.700613
  HQ           0.728089     SCHWARZ     0.850821     SHIBATA     0.657909
  GCV          0.726989     RICE        0.769596

# Equation-by-equation TSLS
? tsls C 0 P P1 W ; 0 G T A Wg P1 K1 X1

Model 4: TSLS estimates using the 21 observations 1921-1941
Dependent variable: C
Instruments: G T A Wg K1 X1

      VARIABLE      COEFFICIENT        STDERROR       T STAT   2Prob(t > |T|)

   0)    const        16.5548            1.46798      11.277   < 0.00001 ***
   2)        P         0.0173022         0.131205      0.132    0.896634
  12)       P1         0.216234          0.119222      1.814    0.087413 *
  10)        W         0.810183          0.0447351    18.111   < 0.00001 ***

  Mean of dependent variable = 53.9952
  Standard deviation of dep. var. = 6.86087
  Sum of squared residuals = 21.9252
```

```
  Standard error of residuals = 1.13566
  Unadjusted R-squared = 0.976805
  Adjusted R-squared = 0.972712
  F-statistic (3, 17) = 238.638 (p-value < 0.00001)
  Durbin-Watson statistic = 1.48507
  First-order autocorrelation coeff. = 0.204234

R-squared is computed as the square of the correlation between observed and
fitted values of the dependent variable.

  MODEL SELECTION STATISTICS

  SGMASQ        1.28972     AIC           1.52817     FPE           1.53538
  HQ            1.59560     SCHWARZ       1.86456     SHIBATA       1.44180
  GCV           1.59318     RICE          1.68656

Excluding the constant, p-value was highest for variable 2 (P)

? tsls I 0 P P1 K1 ; 0 G T A Wg P1 K1 X1

Model 5: TSLS estimates using the 21 observations 1921-1941
Dependent variable: I
Instruments: G T A Wg X1

      VARIABLE      COEFFICIENT        STDERROR       T STAT   2Prob(t > |T|)

   0)    const        20.2782            8.38325       2.419    0.027071 **
   2)        P         0.150222          0.192534      0.780    0.445980
  12)       P1         0.615944          0.180926      3.404    0.003375 ***
   5)       K1        -0.157788          0.0401521    -3.930    0.001080 ***

  Mean of dependent variable = 1.26667
  Standard deviation of dep. var. = 3.55195
  Sum of squared residuals = 29.0469
  Standard error of residuals = 1.30715
  Unadjusted R-squared = 0.885417
  Adjusted R-squared = 0.865196
  F-statistic (3, 17) = 43.7879 (p-value < 0.00001)
  Durbin-Watson statistic = 2.08533
  First-order autocorrelation coeff. = -0.0752597

R-squared is computed as the square of the correlation between observed and
fitted values of the dependent variable.

  MODEL SELECTION STATISTICS

  SGMASQ        1.70864     AIC           2.02454     FPE           2.03409
  HQ            2.11387     SCHWARZ       2.47019     SHIBATA       1.91011
  GCV           2.11067     RICE          2.23437

Excluding the constant, p-value was highest for variable 2 (P)

? tsls Wp 0 X X1 A ; 0 G T A Wg P1 K1 X1

Model 6: TSLS estimates using the 21 observations 1921-1941
Dependent variable: Wp
Instruments: G T Wg P1 K1

      VARIABLE      COEFFICIENT        STDERROR       T STAT   2Prob(t > |T|)

   0)    const         1.50030           1.27569       1.176    0.255774
   6)        X         0.438859          0.0396027    11.082   < 0.00001 ***
  13)       X1         0.146674          0.0431639     3.398    0.003422 ***
  11)        A         0.130396          0.0323884     4.026    0.000876 ***

  Mean of dependent variable = 36.3619
  Standard deviation of dep. var. = 6.3044
  Sum of squared residuals = 10.005
  Standard error of residuals = 0.767155
  Unadjusted R-squared = 0.987414
  Adjusted R-squared = 0.985193
  F-statistic (3, 17) = 444.56 (p-value < 0.00001)
```

```
  Durbin-Watson statistic = 1.96342
  First-order autocorrelation coeff. = -0.0862977

R-squared is computed as the square of the correlation between observed and
fitted values of the dependent variable.

  MODEL SELECTION STATISTICS

  SGMASQ       0.588527     AIC          0.697335     FPE          0.700628
  HQ           0.728105     SCHWARZ      0.850839     SHIBATA      0.657923
  GCV          0.727004     RICE         0.769613
```

11.8. **Tobit.** The Tobit model[37] deals with a dependent variable y which is only observed if $y_i > 0$. To be more precise, in the regression equation

$$y' = X\beta + \varepsilon \tag{33}$$

we only observe

$$y = max\{y', 0\} \tag{34}$$

A well-known method to estimate β in this case is to maximize the log-likelihood function[19, 2]:

$$\begin{aligned} \ln L &= \sum_{y_i>0} -\frac{1}{2}\left[\ln(2\pi) + \ln\sigma^2 + \frac{(y_i - x_i'\beta)^2}{\sigma^2}\right] + \sum_{y_i=0} \ln\left[1 - \Phi\left(\frac{x_i'\beta}{\sigma}\right)\right] \\ &= \sum_{y_i>0} -\frac{1}{2}\left[\ln\sigma^2 + \frac{(y_i - x_i'\beta)^2}{\sigma^2}\right] + \sum_{y_i=0} \ln\left[1 - \Phi\left(\frac{x_i'\beta}{\sigma}\right)\right] - \sum_{y_i>0} \ln\sqrt{2\pi} \end{aligned} \tag{35}$$

where $\Phi(.)$ is the distribution function of the standard normal distribution $N(0, 1)$.

We can use OLS estimates as initial values for the nonlinear optimization problem.

11.8.1. *Model tobit.gms.* [17]

```
$ontext

    Tobit analysis.

    Use least squares solution as starting point for max likelihood optimization.

    Erwin Kalvelagen, dec 2004

    References:
       William H. Greene, "Econometric Analysis", 5th ed.


    Data from Fair (1977).

$offtext


set id 'record id' /1*9999/;
set v  'variables' /const,X1,X2,Z1,Z2,Z3,Z4,Z5,Z6,X3,Z7,Z8,Y,X4,X5/;

table data(id,*)

        X1 X2 Z1  Z2    Z3    Z4 Z5 Z6   X3  Z7 Z8  Y  X4  X5
      4  0 1.  1 37.0 10.000  0  3 18. 40.0  7  4  0.  0.  1.
      5  0 1.  0 27.0  4.000  0  4 14. 20.0  6  4  0.  0.  1.
     11  0 1.  0 32.0 15.000  1  1 12. 12.5  1  4  0.  0.  1.
     16  0 1.  1 57.0 15.000  1  5 18. 12.5  6  5  0.  0.  1.
     23  0 1.  1 22.0  0.750  0  2 17.  7.5  6  3  0.  0.  1.
```

[17] www.amsterdamoptimization.com/models/regression/tobit.gms

```
 29  0 1.  0 32.0  1.500  0  2 17.   7.5  5  5  0.  0.  1.
 44  0 1.  0 22.0  0.750  0  2 12.  12.5  1  3  0.  0.  1.
 45  0 1.  1 57.0 15.000  1  2 14.  20.0  4  4  0.  0.  1.
 47  0 1.  0 32.0 15.000  1  4 16.  20.0  1  2  0.  0.  1.
 49  0 1.  1 22.0  1.500  0  4 14.  12.5  4  5  0.  0.  1.
 50  0 1.  1 37.0 15.000  1  2 20.  20.0  7  2  0.  0.  1.
 55  0 1.  1 27.0  4.000  1  4 18.  12.5  6  4  0.  0.  1.
 64  0 1.  1 47.0 15.000  1  5 17.  12.5  6  4  0.  0.  1.
 80  0 1.  0 22.0  1.500  0  2 17.  12.5  5  4  0.  0.  1.
 86  0 1.  0 27.0  4.000  0  4 14.   7.5  5  4  0.  0.  1.
 93  0 1.  0 37.0 15.000  1  1 17.  20.0  5  5  0.  0.  1.
108  0 1.  0 37.0 15.000  1  2 18.  20.0  4  3  0.  0.  1.
114  0 1.  0 22.0  0.750  0  3 16.   7.5  5  4  0.  0.  1.
115  0 1.  0 22.0  1.500  0  2 16.   7.5  5  5  0.  0.  1.
116  0 1.  0 27.0 10.000  1  2 14.   7.5  1  5  0.  0.  1.
123  0 1.  0 22.0  1.500  0  2 16.  12.5  5  5  0.  0.  1.
127  0 1.  0 22.0  1.500  0  2 16.   7.5  5  5  0.  0.  1.
129  0 1.  0 27.0 10.000  1  4 16.  20.0  5  4  0.  0.  1.
134  0 1.  0 32.0 10.000  1  3 14.   7.5  1  5  0.  0.  1.
137  0 1.  1 37.0  4.000  1  2 20.  20.0  6  4  0.  0.  1.
139  0 1.  0 22.0  1.500  0  2 18.  12.5  5  5  0.  0.  1.
147  0 1.  0 27.0  7.000  0  4 16.  12.5  1  5  0.  0.  1.
151  0 1.  1 42.0 15.000  1  5 20.  12.5  6  4  0.  0.  1.
153  0 1.  1 27.0  4.000  1  3 16.  12.5  5  5  0.  0.  1.
155  0 1.  0 27.0  4.000  1  3 17.  12.5  5  4  0.  0.  1.
162  0 1.  1 42.0 15.000  1  4 20.  20.0  6  3  0.  0.  1.
163  0 1.  0 22.0  1.500  0  3 16.  12.5  5  5  0.  0.  1.
165  0 1.  1 27.0  0.417  0  4 17.   7.5  6  4  0.  0.  1.
168  0 1.  0 42.0 15.000  1  5 14.  20.0  5  4  0.  0.  1.
170  0 1.  1 32.0  4.000  1  1 18.  20.0  6  4  0.  0.  1.
172  0 1.  0 22.0  1.500  0  4 16.   7.5  5  3  0.  0.  1.
184  0 1.  0 42.0 15.000  1  3 12.  20.0  1  4  0.  0.  1.
187  0 1.  0 22.0  4.000  0  4 17.  20.0  5  5  0.  0.  1.
192  0 1.  1 22.0  1.500  1  1 14.   7.5  3  5  0.  0.  1.
194  0 1.  0 22.0  0.750  0  3 16.   7.5  1  5  0.  0.  1.
210  0 1.  1 32.0 10.000  1  5 20.  12.5  6  5  0.  0.  1.
217  0 1.  1 52.0 15.000  1  5 18.   7.5  6  3  0.  0.  1.
220  0 1.  0 22.0  0.417  0  5 14.  12.5  1  4  0.  0.  1.
224  0 1.  0 27.0  4.000  1  2 18.   4.0  6  1  0.  0.  1.
227  0 1.  0 32.0  7.000  1  5 17.  12.5  5  3  0.  0.  1.
228  0 1.  1 22.0  4.000  0  3 16.   7.5  5  5  0.  0.  1.
239  0 1.  0 27.0  7.000  1  4 18.  40.0  6  5  0.  0.  1.
241  0 1.  0 42.0 15.000  1  2 18.  20.0  5  4  0.  0.  1.
245  0 1.  1 27.0  1.500  1  4 16.   7.5  3  5  0.  0.  1.
249  0 1.  1 42.0 15.000  1  2 20.  20.0  6  4  0.  0.  1.
262  0 1.  0 22.0  0.750  0  5 14.  20.0  3  5  0.  0.  1.
265  0 1.  1 32.0  7.000  1  2 20.  20.0  6  4  0.  0.  1.
267  0 1.  1 27.0  4.000  1  5 20.   7.5  6  5  0.  0.  1.
269  0 1.  1 27.0 10.000  1  4 20.   7.5  6  4  0.  0.  1.
271  0 1.  1 22.0  4.000  0  1 18.  20.0  5  5  0.  0.  1.
277  0 1.  0 37.0 15.000  1  4 14.  12.5  3  1  0.  0.  1.
290  0 1.  1 22.0  1.500  1  5 16.  20.0  4  4  0.  0.  1.
292  0 1.  0 37.0 15.000  1  4 17.  20.0  1  5  0.  0.  1.
293  0 1.  0 27.0  0.750  0  4 17.  12.5  5  4  0.  0.  1.
295  0 1.  1 32.0 10.000  1  4 20.  12.5  6  4  0.  0.  1.
299  0 1.  0 47.0 15.000  1  5 14.  40.0  7  2  0.  0.  1.
320  0 1.  1 37.0 10.000  1  3 20.  40.0  6  4  0.  0.  1.
321  0 1.  0 22.0  0.750  0  2 16.   7.5  5  5  0.  0.  1.
324  0 1.  1 27.0  4.000  0  2 18.  12.5  4  5  0.  0.  1.
334  0 1.  1 32.0  7.000  0  4 20.   7.5  6  4  0.  0.  1.
351  0 1.  1 42.0 15.000  1  2 17.  40.0  3  5  0.  0.  1.
355  0 1.  1 37.0 10.000  1  4 20.   7.5  6  4  0.  0.  1.
361  0 1.  0 47.0 15.000  1  3 17.  20.0  6  5  0.  0.  1.
362  0 1.  0 22.0  1.500  0  5 16.   7.5  5  5  0.  0.  1.
366  0 1.  0 27.0  1.500  0  2 16.  20.0  6  4  0.  0.  1.
370  0 1.  0 27.0  4.000  0  3 17.   7.5  5  5  0.  0.  1.
374  0 1.  0 32.0 10.000  1  5 14.  12.5  4  5  0.  0.  1.
378  0 1.  0 22.0  0.125  0  2 12.   7.5  5  5  0.  0.  1.
381  0 1.  1 47.0 15.000  1  4 14.  20.0  4  3  0.  0.  1.
382  0 1.  1 32.0 15.000  1  1 14.  40.0  5  5  0.  0.  1.
383  0 1.  1 27.0  7.000  1  4 16.  12.5  5  5  0.  0.  1.
384  0 1.  0 22.0  1.500  1  3 16.   7.5  5  5  0.  0.  1.
```

```
400  0 1.  1 27.0  4.000  1  3 17.  7.5  6  5  0.  0.  1.
403  0 1.  0 22.0  1.500  0  3 16.  7.5  5  5  0.  0.  1.
409  0 1.  1 57.0 15.000  1  2 14. 20.0  7  2  0.  0.  1.
412  0 1.  1 17.5  1.500  1  3 18. 20.0  6  5  0.  0.  1.
413  0 1.  1 57.0 15.000  1  4 20. 40.0  6  5  0.  0.  1.
416  0 1.  0 22.0  0.750  0  2 16. 20.0  3  4  0.  0.  1.
418  0 1.  1 42.0  4.000  0  4 17. 12.5  3  3  0.  0.  1.
422  0 1.  0 22.0  1.500  1  4 12. 12.5  1  5  0.  0.  1.
435  0 1.  0 22.0  0.417  0  1 17.  4.0  6  4  0.  0.  1.
439  0 1.  0 32.0 15.000  1  4 17. 12.5  5  5  0.  0.  1.
445  0 1.  0 27.0  1.500  0  3 18. 12.5  5  2  0.  0.  1.
447  0 1.  0 22.0  1.500  1  3 14.  7.5  1  5  0.  0.  1.
448  0 1.  0 37.0 15.000  1  3 14. 40.0  1  4  0.  0.  1.
449  0 1.  0 32.0 15.000  1  4 14. 20.0  3  4  0.  0.  1.
478  0 1.  1 37.0 10.000  1  2 14. 12.5  5  3  0.  0.  1.
482  0 1.  1 37.0 10.000  1  4 16. 12.5  5  4  0.  0.  1.
486  0 1.  1 57.0 15.000  1  5 20. 12.5  5  3  0.  0.  1.
489  0 1.  1 27.0  0.417  0  1 16.  7.5  3  4  0.  0.  1.
490  0 1.  0 42.0 15.000  1  5 14. 12.5  1  5  0.  0.  1.
491  0 1.  1 57.0 15.000  1  3 16. 20.0  6  1  0.  0.  1.
492  0 1.  1 37.0 10.000  1  1 16.  7.5  6  4  0.  0.  1.
503  0 1.  1 37.0 15.000  1  3 17. 40.0  5  5  0.  0.  1.
508  0 1.  1 37.0 15.000  1  4 20. 20.0  6  5  0.  0.  1.
509  0 1.  0 27.0 10.000  1  5 14. 12.5  1  5  0.  0.  1.
512  0 1.  1 37.0 10.000  1  2 18. 20.0  6  4  0.  0.  1.
515  0 1.  0 22.0  0.125  0  4 12. 12.5  4  5  0.  0.  1.
517  0 1.  1 57.0 15.000  1  5 20. 40.0  6  5  0.  0.  1.
532  0 1.  0 37.0 15.000  1  4 18. 40.0  6  4  0.  0.  1.
533  0 1.  1 22.0  4.000  1  4 14.  7.5  6  4  0.  0.  1.
535  0 1.  1 27.0  7.000  1  4 18.  7.5  5  4  0.  0.  1.
537  0 1.  1 57.0 15.000  1  4 20. 40.0  5  4  0.  0.  1.
538  0 1.  1 32.0 15.000  1  3 14. 12.5  6  3  0.  0.  1.
543  0 1.  0 22.0  1.500  0  2 14. 12.5  5  4  0.  0.  1.
547  0 1.  0 32.0  7.000  1  4 17. 40.0  1  5  0.  0.  1.
550  0 1.  0 37.0 15.000  1  4 17. 40.0  6  5  0.  0.  1.
558  0 1.  0 32.0  1.500  0  5 18. 40.0  5  5  0.  0.  1.
571  0 1.  1 42.0 10.000  1  5 20. 40.0  7  4  0.  0.  1.
578  0 1.  0 27.0  7.000  0  3 16. 12.5  5  4  0.  0.  1.
583  0 1.  1 37.0 15.000  0  4 20. 40.0  6  5  0.  0.  1.
586  0 1.  1 37.0 15.000  1  4 14. 12.5  3  2  0.  0.  1.
594  0 1.  1 32.0 10.000  0  5 18. 20.0  6  4  0.  0.  1.
597  0 1.  0 22.0  0.750  0  4 16.  7.5  1  5  0.  0.  1.
602  0 1.  0 27.0  7.000  1  4 12.  7.5  2  4  0.  0.  1.
603  0 1.  0 27.0  7.000  1  2 16. 12.5  2  5  0.  0.  1.
604  0 1.  0 42.0 15.000  1  5 18. 40.0  5  4  0.  0.  1.
612  0 1.  1 42.0 15.000  1  4 17. 20.0  5  3  0.  0.  1.
613  0 1.  0 27.0  7.000  1  2 16. 20.0  1  2  0.  0.  1.
621  0 1.  0 22.0  1.500  0  3 16. 12.5  5  5  0.  0.  1.
627  0 1.  1 37.0 15.000  1  5 20. 40.0  6  5  0.  0.  1.
630  0 1.  0 22.0  0.125  0  2 14. 12.5  4  5  0.  0.  1.
631  0 1.  1 27.0  1.500  0  4 16.  7.5  5  5  0.  0.  1.
632  0 1.  1 32.0  1.500  0  2 18. 20.0  6  5  0.  0.  1.
639  0 1.  1 27.0  1.500  0  2 17.  7.5  6  5  0.  0.  1.
645  0 1.  0 27.0 10.000  1  4 16. 12.5  1  3  0.  0.  1.
647  0 1.  1 42.0 15.000  1  4 18. 12.5  6  5  0.  0.  1.
648  0 1.  0 27.0  1.500  0  2 16.  7.5  6  5  0.  0.  1.
651  0 1.  1 27.0  4.000  0  2 18. 12.5  6  3  0.  0.  1.
655  0 1.  0 32.0 10.000  1  3 14.  7.5  5  3  0.  0.  1.
667  0 1.  0 32.0 15.000  1  3 18. 20.0  5  4  0.  0.  1.
670  0 1.  0 22.0  0.750  0  2 18.  4.0  6  5  0.  0.  1.
671  0 1.  0 37.0 15.000  1  2 16.  7.5  1  4  0.  0.  1.
673  0 1.  1 27.0  4.000  1  4 20. 12.5  5  5  0.  0.  1.
701  0 1.  1 27.0  4.000  0  1 20. 20.0  5  4  0.  0.  1.
705  0 1.  0 27.0 10.000  1  2 12.  7.5  1  4  0.  0.  1.
706  0 1.  0 32.0 15.000  1  5 18. 20.0  6  4  0.  0.  1.
709  0 1.  1 27.0  7.000  1  5 12.  7.5  5  3  0.  0.  1.
717  0 1.  1 52.0 15.000  1  2 18. 12.5  5  4  0.  0.  1.
719  0 1.  1 27.0  4.000  0  3 20. 12.5  6  3  0.  0.  1.
723  0 1.  1 37.0  4.000  1  1 18. 20.0  5  4  0.  0.  1.
724  0 1.  1 27.0  4.000  1  4 14.  7.5  5  4  0.  0.  1.
726  0 1.  0 52.0 15.000  1  5 12. 12.5  1  3  0.  0.  1.
734  0 1.  0 57.0 15.000  1  4 16. 20.0  6  4  0.  0.  1.
```

```
  735  0 1.  1 27.0  7.000  1  1 16. 20.0  5  4  0.  0.  1.
  736  0 1.  1 37.0  7.000  1  4 20. 20.0  6  3  0.  0.  1.
  737  0 1.  1 22.0  0.750  0  2 14. 12.5  4  3  0.  0.  1.
  739  0 1.  1 32.0  4.000  1  2 18.  7.5  5  3  0.  0.  1.
  743  0 1.  1 37.0 15.000  1  4 20.  7.5  6  3  0.  0.  1.
  745  0 1.  1 22.0  0.750  1  2 14.  7.5  4  3  0.  0.  1.
  747  0 1.  1 42.0 15.000  1  4 20. 20.0  6  3  0.  0.  1.
  751  0 1.  0 52.0 15.000  1  5 17. 12.5  1  1  0.  0.  1.
  752  0 1.  0 37.0 15.000  1  4 14. 40.0  1  2  0.  0.  1.
  754  0 1.  1 27.0  7.000  1  4 14. 12.5  5  3  0.  0.  1.
  760  0 1.  1 32.0  4.000  1  2 16.  7.5  5  5  0.  0.  1.
  763  0 1.  0 27.0  4.000  1  2 18. 12.5  6  5  0.  0.  1.
  774  0 1.  0 27.0  4.000  1  2 18.  7.5  5  5  0.  0.  1.
  776  0 1.  1 37.0 15.000  1  5 18.  7.5  6  5  0.  0.  1.
  779  0 1.  0 47.0 15.000  1  5 12. 12.5  5  4  0.  0.  1.
  784  0 1.  0 32.0 10.000  1  3 17. 12.5  1  4  0.  0.  1.
  788  0 1.  0 27.0  1.500  1  4 17.  7.5  1  2  0.  0.  1.
  794  0 1.  0 57.0 15.000  1  2 18. 20.0  5  2  0.  0.  1.
  795  0 1.  0 22.0  1.500  0  4 14.  7.5  5  4  0.  0.  1.
  798  0 1.  1 42.0 15.000  1  3 14. 12.5  3  4  0.  0.  1.
  800  0 1.  1 57.0 15.000  1  4  9.  7.5  2  2  0.  0.  1.
  803  0 1.  1 57.0 15.000  1  4 20. 40.0  6  5  0.  0.  1.
  807  0 1.  0 22.0  0.125  0  4 14. 12.5  4  5  0.  0.  1.
  812  0 1.  0 32.0 10.000  1  4 14. 20.0  1  5  0.  0.  1.
  820  0 1.  0 42.0 15.000  1  3 18. 20.0  5  4  0.  0.  1.
  823  0 1.  0 27.0  1.500  0  2 18. 20.0  6  5  0.  0.  1.
  830  0 1.  1 32.0  0.125  1  2 18. 12.5  5  2  0.  0.  1.
  843  0 1.  0 27.0  4.000  0  3 16. 20.0  5  4  0.  0.  1.
  848  0 1.  0 27.0 10.000  1  2 16. 12.5  1  4  0.  0.  1.
  851  0 1.  0 32.0  7.000  1  4 16. 40.0  1  3  0.  0.  1.
  854  0 1.  0 37.0 15.000  1  4 14. 40.0  5  4  0.  0.  1.
  856  0 1.  0 42.0 15.000  1  5 17. 12.5  6  2  0.  0.  1.
  857  0 1.  1 32.0  1.500  1  4 14. 12.5  6  5  0.  0.  1.
  859  0 1.  0 32.0  4.000  1  3 17. 12.5  5  3  0.  0.  1.
  863  0 1.  0 37.0  7.000  0  4 18. 12.5  5  5  0.  0.  1.
  865  0 1.  0 22.0  0.417  1  3 14.  7.5  3  5  0.  0.  1.
  867  0 1.  0 27.0  7.000  1  4 14. 12.5  1  5  0.  0.  1.
  870  0 1.  1 27.0  0.750  0  3 16. 12.5  5  5  0.  0.  1.
  873  0 1.  1 27.0  4.000  1  2 20. 40.0  5  5  0.  0.  1.
  875  0 1.  1 32.0 10.000  1  4 16.  7.5  4  5  0.  0.  1.
  876  0 1.  1 32.0 15.000  1  1 14. 12.5  5  5  0.  0.  1.
  877  0 1.  1 22.0  0.750  0  3 17.  7.5  4  5  0.  0.  1.
  880  0 1.  0 27.0  7.000  1  4 17. 20.0  1  4  0.  0.  1.
  903  0 1.  1 27.0  0.417  1  4 20. 12.5  5  4  0.  0.  1.
  904  0 1.  1 37.0 15.000  1  4 20. 40.0  5  4  0.  0.  1.
  905  0 1.  0 37.0 15.000  1  2 14. 12.5  1  3  0.  0.  1.
  908  0 1.  1 22.0  4.000  1  1 18.  7.5  5  4  0.  0.  1.
  909  0 1.  1 37.0 15.000  1  4 17. 20.0  5  3  0.  0.  1.
  910  0 1.  0 22.0  1.500  0  2 14. 12.5  4  5  0.  0.  1.
  912  0 1.  1 52.0 15.000  1  4 14. 20.0  6  2  0.  0.  1.
  914  0 1.  0 22.0  1.500  0  4 17. 12.5  5  5  0.  0.  1.
  915  0 1.  1 32.0  4.000  1  5 14. 20.0  3  5  0.  0.  1.
  916  0 1.  1 32.0  4.000  1  2 14. 20.0  3  5  0.  0.  1.
  920  0 1.  0 22.0  1.500  0  3 16. 12.5  6  5  0.  0.  1.
  921  0 1.  1 27.0  0.750  0  2 18. 12.5  3  3  0.  0.  1.
  925  0 1.  0 22.0  7.000  1  2 14.  7.5  5  2  0.  0.  1.
  926  0 1.  0 27.0  0.750  0  2 17.  7.5  5  3  0.  0.  1.
  929  0 1.  0 37.0 15.000  1  4 12. 20.0  1  2  0.  0.  1.
  931  0 1.  0 22.0  1.500  0  1 14.  4.0  1  5  0.  0.  1.
  945  0 1.  0 37.0 10.000  0  2 12. 12.5  4  4  0.  0.  1.
  947  0 1.  0 37.0 15.000  1  4 18. 20.0  5  3  0.  0.  1.
  949  0 1.  0 42.0 15.000  1  3 12. 20.0  3  3  0.  0.  1.
  950  0 1.  1 22.0  4.000  0  2 18.  7.5  5  5  0.  0.  1.
  961  0 1.  1 52.0  7.000  1  2 20. 20.0  6  2  0.  0.  1.
  965  0 1.  1 27.0  0.750  0  2 17. 12.5  5  5  0.  0.  1.
  966  0 1.  0 27.0  4.000  0  2 17. 12.5  4  5  0.  0.  1.
  967  0 1.  1 42.0  1.500  0  5 20. 12.5  6  5  0.  0.  1.
  987  0 1.  1 22.0  1.500  0  4 17. 12.5  6  5  0.  0.  1.
  990  0 1.  1 22.0  4.000  0  4 17. 12.5  5  3  0.  0.  1.
  992  0 1.  0 22.0  4.000  1  1 14.  7.5  5  4  0.  0.  1.
  995  0 1.  1 37.0 15.000  1  5 20. 12.5  4  5  0.  0.  1.
 1009  0 1.  0 37.0 10.000  1  3 16. 20.0  6  3  0.  0.  1.
```

```
1021  0 1.  1 42.0 15.000  1  4 17. 20.0  6  5  0.  0.  1.
1026  0 1.  0 47.0 15.000  1  4 17. 20.0  5  5  0.  0.  1.
1027  0 1.  1 22.0  1.500  0  4 16.  7.5  5  4  0.  0.  1.
1030  0 1.  0 32.0 10.000  1  3 12. 12.5  1  4  0.  0.  1.
1031  0 1.  0 22.0  7.000  1  1 14.  7.5  3  5  0.  0.  1.
1034  0 1.  0 32.0 10.000  1  4 17. 12.5  5  4  0.  0.  1.
1037  0 1.  1 27.0  1.500  1  2 16.  7.5  2  4  0.  0.  1.
1038  0 1.  1 37.0 15.000  1  4 14. 12.5  5  5  0.  0.  1.
1039  0 1.  1 42.0  4.000  1  3 14. 12.5  4  5  0.  0.  1.
1045  0 1.  0 37.0 15.000  1  5 14. 20.0  5  4  0.  0.  1.
1046  0 1.  0 32.0  7.000  1  4 17. 12.5  5  5  0.  0.  1.
1054  0 1.  0 42.0 15.000  1  4 18. 40.0  6  5  0.  0.  1.
1059  0 1.  1 27.0  4.000  0  4 18. 20.0  6  4  0.  0.  1.
1063  0 1.  1 22.0  0.750  0  4 18. 12.5  6  5  0.  0.  1.
1068  0 1.  1 27.0  4.000  1  4 14.  7.5  5  3  0.  0.  1.
1070  0 1.  0 22.0  0.750  0  5 18.  4.0  1  5  0.  0.  1.
1072  0 1.  0 52.0 15.000  1  5  9. 12.5  5  5  0.  0.  1.
1073  0 1.  1 32.0 10.000  1  3 14. 12.5  5  5  0.  0.  1.
1077  0 1.  0 37.0 15.000  1  4 16.  7.5  4  4  0.  0.  1.
1081  0 1.  1 32.0  7.000  1  2 20. 20.0  5  4  0.  0.  1.
1083  0 1.  0 42.0 15.000  1  3 18. 20.0  1  4  0.  0.  1.
1084  0 1.  1 32.0 15.000  1  1 16. 12.5  5  5  0.  0.  1.
1086  0 1.  1 27.0  4.000  1  3 18.  7.5  5  5  0.  0.  1.
1087  0 1.  0 32.0 15.000  1  4 12.  7.5  3  4  0.  0.  1.
1089  0 1.  1 22.0  0.750  1  3 14.  7.5  2  4  0.  0.  1.
1096  0 1.  0 22.0  1.500  0  3 16.  7.5  5  3  0.  0.  1.
1102  0 1.  0 42.0 15.000  1  4 14. 12.5  3  5  0.  0.  1.
1103  0 1.  0 52.0 15.000  1  3 16. 20.0  5  4  0.  0.  1.
1107  0 1.  1 37.0 15.000  1  5 20. 20.0  6  4  0.  0.  1.
1109  0 1.  0 47.0 15.000  1  4 12. 12.5  2  3  0.  0.  1.
1115  0 1.  1 57.0 15.000  1  2 20. 40.0  6  4  0.  0.  1.
1119  0 1.  1 32.0  7.000  1  4 17. 20.0  5  5  0.  0.  1.
1124  0 1.  0 27.0  7.000  1  4 17. 12.5  1  4  0.  0.  1.
1126  0 1.  1 22.0  1.500  0  1 18.  7.5  6  5  0.  0.  1.
1128  0 1.  0 22.0  4.000  1  3  9.  7.5  1  4  0.  0.  1.
1129  0 1.  0 22.0  1.500  0  2 14.  7.5  1  5  0.  0.  1.
1130  0 1.  1 42.0 15.000  1  2 20. 20.0  6  4  0.  0.  1.
1133  0 1.  1 57.0 15.000  1  4  9. 12.5  2  4  0.  0.  1.
1140  0 1.  0 27.0  7.000  1  2 18. 20.0  1  5  0.  0.  1.
1143  0 1.  0 22.0  4.000  1  3 14.  7.5  1  5  0.  0.  1.
1146  0 1.  1 37.0 15.000  1  4 14. 20.0  5  3  0.  0.  1.
1153  0 1.  1 32.0  7.000  1  1 18.  4.0  6  4  0.  0.  1.
1156  0 1.  0 22.0  1.500  0  2 14.  4.0  5  5  0.  0.  1.
1157  0 1.  0 22.0  1.500  1  3 12.  4.0  1  3  0.  0.  1.
1158  0 1.  1 52.0 15.000  1  2 14. 40.0  5  5  0.  0.  1.
1160  0 1.  0 37.0 15.000  1  2 14. 20.0  1  1  0.  0.  1.
1161  0 1.  0 32.0 10.000  1  2 14. 12.5  5  5  0.  0.  1.
1166  0 1.  1 42.0 15.000  1  4 20. 12.5  4  5  0.  0.  1.
1177  0 1.  0 27.0  4.000  1  3 18. 12.5  4  5  0.  0.  1.
1178  0 1.  1 37.0 15.000  1  4 20. 40.0  6  5  0.  0.  1.
1180  0 1.  1 27.0  1.500  0  3 18. 12.5  5  5  0.  0.  1.
1187  0 1.  0 22.0  0.125  0  2 16.  7.5  6  3  0.  0.  1.
1191  0 1.  1 32.0 10.000  1  2 20. 12.5  6  3  0.  0.  1.
1195  0 1.  0 27.0  4.000  0  4 18. 20.0  5  4  0.  0.  1.
1207  0 1.  0 27.0  7.000  1  2 12. 12.5  5  1  0.  0.  1.
1208  0 1.  1 32.0  4.000  1  5 18.  7.5  6  3  0.  0.  1.
1209  0 1.  0 37.0 15.000  1  2 17. 12.5  5  5  0.  0.  1.
1211  0 1.  1 47.0 15.000  0  4 20. 20.0  6  4  0.  0.  1.
1215  0 1.  1 27.0  1.500  0  1 18. 12.5  5  5  0.  0.  1.
1221  0 1.  1 37.0 15.000  1  4 20. 20.0  6  4  0.  0.  1.
1226  0 1.  0 32.0 15.000  1  4 18. 12.5  1  4  0.  0.  1.
1229  0 1.  0 32.0  7.000  1  4 17. 40.0  5  4  0.  0.  1.
1231  0 1.  0 42.0 15.000  1  3 14.  7.5  1  3  0.  0.  1.
1234  0 1.  0 27.0  7.000  1  3 16. 12.5  1  4  0.  0.  1.
1235  0 1.  1 27.0  1.500  0  3 16.  7.5  4  2  0.  0.  1.
1242  0 1.  1 22.0  1.500  0  3 16.  7.5  3  5  0.  0.  1.
1245  0 1.  1 27.0  4.000  1  3 16. 12.5  4  2  0.  0.  1.
1260  0 1.  0 27.0  7.000  1  3 12. 12.5  1  2  0.  0.  1.
1266  0 1.  0 37.0 15.000  1  2 18. 40.0  5  4  0.  0.  1.
1271  0 1.  0 37.0  7.000  1  3 14. 20.0  4  4  0.  0.  1.
1273  0 1.  1 22.0  1.500  0  2 16.  7.5  5  5  0.  0.  1.
1276  0 1.  1 37.0 15.000  1  5 20. 40.0  5  4  0.  0.  1.
```

```
1280  0 1.  0 22.0  1.500  0  4 16.  7.5  5  3  0.  0.  1.
1282  0 1.  0 32.0 10.000  1  4 16. 20.0  1  5  0.  0.  1.
1285  0 1.  1 27.0  4.000  0  2 17. 12.5  5  3  0.  0.  1.
1295  0 1.  0 22.0  0.417  0  4 14. 12.5  5  5  0.  0.  1.
1298  0 1.  0 27.0  4.000  0  2 18. 12.5  5  5  0.  0.  1.
1299  0 1.  1 37.0 15.000  1  4 18.  7.5  5  3  0.  0.  1.
1304  0 1.  1 37.0 10.000  1  5 20. 12.5  7  4  0.  0.  1.
1305  0 1.  0 27.0  7.000  1  2 14. 12.5  4  2  0.  0.  1.
1311  0 1.  1 32.0  4.000  1  2 16. 20.0  5  5  0.  0.  1.
1314  0 1.  1 32.0  4.000  1  2 16. 12.5  6  4  0.  0.  1.
1319  0 1.  1 22.0  1.500  0  3 18. 12.5  4  5  0.  0.  1.
1322  0 1.  0 22.0  4.000  1  4 14. 12.5  3  4  0.  0.  1.
1324  0 1.  0 17.5  0.750  0  2 18. 12.5  5  4  0.  0.  1.
1327  0 1.  1 32.0 10.000  1  4 20.  7.5  4  5  0.  0.  1.
1328  0 1.  0 32.0  0.750  0  5 14. 20.0  3  3  0.  0.  1.
1330  0 1.  1 37.0 15.000  1  4 17.  7.5  5  3  0.  0.  1.
1332  0 1.  1 32.0  4.000  0  3 14. 12.5  4  5  0.  0.  1.
1333  0 1.  0 27.0  1.500  0  2 17. 20.0  3  2  0.  0.  1.
1336  0 1.  0 22.0  7.000  1  4 14.  7.5  1  5  0.  0.  1.
1341  0 1.  1 47.0 15.000  1  5 14. 20.0  6  5  0.  0.  1.
1344  0 1.  1 27.0  4.000  1  1 16. 12.5  4  4  0.  0.  1.
1352  0 1.  0 37.0 15.000  1  5 14. 12.5  1  3  0.  0.  1.
1358  0 1.  1 42.0  4.000  1  4 18. 20.0  5  5  0.  0.  1.
1359  0 1.  0 32.0  4.000  1  2 14. 12.5  1  5  0.  0.  1.
1361  0 1.  1 52.0 15.000  1  2 14. 40.0  7  4  0.  0.  1.
1364  0 1.  0 22.0  1.500  0  2 16.  7.5  1  4  0.  0.  1.
1368  0 1.  1 52.0 15.000  1  4 12. 12.5  2  4  0.  0.  1.
1384  0 1.  0 22.0  0.417  0  3 17. 12.5  1  5  0.  0.  1.
1390  0 1.  0 22.0  1.500  0  2 16.  7.5  5  5  0.  0.  1.
1393  0 1.  1 27.0  4.000  1  4 20. 12.5  6  4  0.  0.  1.
1394  0 1.  0 32.0 15.000  1  4 14. 12.5  1  5  0.  0.  1.
1402  0 1.  0 27.0  1.500  0  2 16. 20.0  3  5  0.  0.  1.
1407  0 1.  1 32.0  4.000  0  1 20. 20.0  6  5  0.  0.  1.
1408  0 1.  1 37.0 15.000  1  3 20. 20.0  6  4  0.  0.  1.
1412  0 1.  0 32.0 10.000  0  2 16. 20.0  6  5  0.  0.  1.
1413  0 1.  0 32.0 10.000  1  5 14. 12.5  5  5  0.  0.  1.
1416  0 1.  1 37.0  1.500  1  4 18.  7.5  5  3  0.  0.  1.
1417  0 1.  1 32.0  1.500  0  2 18. 12.5  4  4  0.  0.  1.
1418  0 1.  0 32.0 10.000  1  4 14. 12.5  1  4  0.  0.  1.
1419  0 1.  0 47.0 15.000  1  4 18. 20.0  5  4  0.  0.  1.
1420  0 1.  0 27.0 10.000  1  5 12.  7.5  1  5  0.  0.  1.
1423  0 1.  1 27.0  4.000  1  3 16. 12.5  4  5  0.  0.  1.
1424  0 1.  0 37.0 15.000  1  4 12. 12.5  4  2  0.  0.  1.
1432  0 1.  0 27.0  0.750  0  4 16. 12.5  5  5  0.  0.  1.
1433  0 1.  0 37.0 15.000  1  4 16.  7.5  1  5  0.  0.  1.
1437  0 1.  0 32.0 15.000  1  3 16. 12.5  1  5  0.  0.  1.
1438  0 1.  0 27.0 10.000  1  2 16.  7.5  1  5  0.  0.  1.
1439  0 1.  1 27.0  7.000  0  2 20. 20.0  6  5  0.  0.  1.
1446  0 1.  0 37.0 15.000  1  2 14. 20.0  1  3  0.  0.  1.
1450  0 1.  1 27.0  1.500  1  2 17.  7.5  4  4  0.  0.  1.
1451  0 1.  0 22.0  0.750  1  2 14. 12.5  1  5  0.  0.  1.
1452  0 1.  1 22.0  4.000  1  4 14.  7.5  2  4  0.  0.  1.
1453  0 1.  1 42.0  0.125  0  4 17. 12.5  6  4  0.  0.  1.
1456  0 1.  1 27.0  1.500  1  4 18.  7.5  6  5  0.  0.  1.
1464  0 1.  1 27.0  7.000  1  3 16. 12.5  6  3  0.  0.  1.
1469  0 1.  0 52.0 15.000  1  4 14.  7.5  1  3  0.  0.  1.
1473  0 1.  1 27.0  1.500  0  5 20.  7.5  5  2  0.  0.  1.
1481  0 1.  0 27.0  1.500  0  2 16.  7.5  5  5  0.  0.  1.
1482  0 1.  0 27.0  1.500  0  3 17. 12.5  5  5  0.  0.  1.
1496  0 1.  1 22.0  0.125  0  5 16.  7.5  4  4  0.  0.  1.
1497  0 1.  0 27.0  4.000  1  4 16.  7.5  1  5  0.  0.  1.
1504  0 1.  0 27.0  4.000  1  4 12.  7.5  1  5  0.  0.  1.
1513  0 1.  0 47.0 15.000  1  2 14. 40.0  5  5  0.  0.  1.
1515  0 1.  0 32.0 15.000  1  3 14. 12.5  5  3  0.  0.  1.
1534  0 1.  1 42.0  7.000  1  2 16. 12.5  5  5  0.  0.  1.
1535  0 1.  1 22.0  0.750  0  4 16.  7.5  6  4  0.  0.  1.
1536  0 1.  1 27.0  0.125  0  3 20.  7.5  6  5  0.  0.  1.
1540  0 1.  1 32.0 10.000  1  3 20. 20.0  6  5  0.  0.  1.
1551  0 1.  0 22.0  0.417  0  5 14. 12.5  4  5  0.  0.  1.
1555  0 1.  0 47.0 15.000  1  5 14. 12.5  1  4  0.  0.  1.
1557  0 1.  0 32.0 10.000  1  3 14. 40.0  1  5  0.  0.  1.
1566  0 1.  1 57.0 15.000  1  4 17. 12.5  5  5  0.  0.  1.
```

```
1567  0 1.  1 27.0  4.000  1  3 20. 12.5  6  5  0.  0.  1.
1576  0 1.  0 32.0  7.000  1  4 17. 12.5  1  5  0.  0.  1.
1584  0 1.  0 37.0 10.000  1  4 16. 40.0  1  5  0.  0.  1.
1585  0 1.  0 32.0 10.000  1  1 18. 20.0  1  4  0.  0.  1.
1590  0 1.  0 22.0  4.000  0  3 14.  7.5  1  4  0.  0.  1.
1594  0 1.  0 27.0  7.000  1  4 14.  7.5  3  2  0.  0.  1.
1595  0 1.  1 57.0 15.000  1  5 18. 20.0  5  2  0.  0.  1.
1603  0 1.  1 32.0  7.000  1  2 18.  7.5  5  5  0.  0.  1.
1608  0 1.  0 27.0  1.500  0  4 17.  7.5  1  3  0.  0.  1.
1609  0 1.  1 22.0  1.500  0  4 14.  7.5  5  5  0.  0.  1.
1615  0 1.  0 22.0  1.500  1  4 14.  4.0  5  4  0.  0.  1.
1616  0 1.  0 32.0  7.000  1  3 16. 12.5  1  5  0.  0.  1.
1617  0 1.  0 47.0 15.000  1  3 16. 20.0  5  4  0.  0.  1.
1620  0 1.  0 22.0  0.750  0  3 16. 40.0  1  5  0.  0.  1.
1621  0 1.  0 22.0  1.500  1  2 14.  7.5  5  5  0.  0.  1.
1637  0 1.  0 27.0  4.000  1  1 16.  7.5  5  5  0.  0.  1.
1638  0 1.  1 52.0 15.000  1  4 16. 20.0  5  5  0.  0.  1.
1650  0 1.  1 32.0 10.000  1  4 20. 40.0  6  5  0.  0.  1.
1654  0 1.  1 47.0 15.000  1  4 16. 20.0  6  4  0.  0.  1.
1665  0 1.  0 27.0  7.000  1  2 14.  7.5  1  2  0.  0.  1.
1670  0 1.  0 22.0  1.500  0  4 14. 20.0  4  5  0.  0.  1.
1671  0 1.  0 32.0 10.000  1  2 16. 20.0  5  4  0.  0.  1.
1675  0 1.  0 22.0  0.750  0  2 16. 12.5  5  4  0.  0.  1.
1688  0 1.  0 22.0  1.500  0  2 16. 12.5  5  5  0.  0.  1.
1691  0 1.  0 42.0 15.000  1  3 18. 20.0  6  4  0.  0.  1.
1695  0 1.  0 27.0  7.000  1  5 14. 20.0  4  5  0.  0.  1.
1698  0 1.  1 42.0 15.000  1  4 16. 20.0  4  4  0.  0.  1.
1704  0 1.  0 57.0 15.000  1  3 18. 20.0  5  2  0.  0.  1.
1705  0 1.  1 42.0 15.000  1  3 18. 12.5  6  2  0.  0.  1.
1711  0 1.  0 32.0  7.000  1  2 14.  7.5  1  2  0.  0.  1.
1719  0 1.  1 22.0  4.000  0  5 12.  7.5  4  5  0.  0.  1.
1723  0 1.  0 22.0  1.500  0  1 16.  7.5  6  5  0.  0.  1.
1726  0 1.  0 22.0  0.750  0  1 14.  7.5  4  5  0.  0.  1.
1749  0 1.  0 32.0 15.000  1  4 12. 20.0  1  5  0.  0.  1.
1752  0 1.  1 22.0  1.500  0  2 18. 12.5  5  3  0.  0.  1.
1754  0 1.  1 27.0  4.000  1  5 17.  7.5  2  5  0.  0.  1.
1758  0 1.  0 27.0  4.000  1  4 12.  7.5  1  5  0.  0.  1.
1761  0 1.  1 42.0 15.000  1  5 18.  7.5  5  4  0.  0.  1.
1773  0 1.  1 32.0  1.500  0  2 20. 20.0  7  3  0.  0.  1.
1775  0 1.  1 57.0 15.000  0  4  9.  7.5  3  1  0.  0.  1.
1786  0 1.  1 37.0  7.000  0  4 18. 12.5  5  5  0.  0.  1.
1793  0 1.  1 52.0 15.000  1  2 17. 20.0  5  4  0.  0.  1.
1799  0 1.  1 47.0 15.000  1  4 17. 20.0  6  5  0.  0.  1.
1803  0 1.  0 27.0  7.000  0  2 17. 20.0  5  4  0.  0.  1.
1806  0 1.  0 27.0  7.000  1  4 14. 40.0  5  5  0.  0.  1.
1807  0 1.  0 22.0  4.000  0  2 14. 12.5  3  3  0.  0.  1.
1808  0 1.  1 37.0  7.000  1  2 20. 40.0  6  5  0.  0.  1.
1814  0 1.  1 27.0  7.000  0  4 12.  7.5  4  3  0.  0.  1.
1815  0 1.  1 42.0 10.000  1  4 18. 40.0  6  4  0.  0.  1.
1818  0 1.  0 22.0  1.500  0  3 14.  7.5  1  5  0.  0.  1.
1827  0 1.  0 22.0  4.000  1  2 14. 12.5  1  3  0.  0.  1.
1834  0 1.  0 57.0 15.000  0  4 20. 20.0  6  5  0.  0.  1.
1835  0 1.  1 37.0 15.000  1  4 14. 12.5  4  3  0.  0.  1.
1843  0 1.  0 27.0  7.000  1  3 18. 12.5  5  5  0.  0.  1.
1846  0 1.  0 17.5 10.000  0  4 14. 20.0  4  5  0.  0.  1.
1850  0 1.  1 22.0  4.000  1  4 16. 12.5  5  5  0.  0.  1.
1851  0 1.  0 27.0  4.000  1  2 16. 12.5  1  4  0.  0.  1.
1854  0 1.  0 37.0 15.000  1  2 14. 12.5  5  1  0.  0.  1.
1859  0 1.  0 22.0  1.500  0  5 14.  4.0  1  4  0.  0.  1.
1861  0 1.  1 27.0  7.000  1  2 20.  7.5  5  4  0.  0.  1.
1866  0 1.  1 27.0  4.000  1  4 14.  7.5  5  5  0.  0.  1.
1873  0 1.  1 22.0  0.125  0  1 16.  7.5  3  5  0.  0.  1.
1875  0 1.  0 27.0  7.000  1  4 14. 20.0  1  4  0.  0.  1.
1885  0 1.  0 32.0 15.000  1  5 16. 12.5  5  3  0.  0.  1.
1892  0 1.  1 32.0 10.000  1  4 18. 12.5  5  4  0.  0.  1.
1895  0 1.  0 32.0 15.000  1  2 14.  7.5  3  4  0.  0.  1.
1896  0 1.  0 22.0  1.500  0  3 17.  7.5  5  5  0.  0.  1.
1897  0 1.  1 27.0  4.000  1  4 17.  7.5  4  4  0.  0.  1.
1899  0 1.  0 52.0 15.000  1  5 14. 12.5  1  5  0.  0.  1.
1904  0 1.  0 27.0  7.000  1  2 12. 20.0  1  2  0.  0.  1.
1905  0 1.  0 27.0  7.000  1  3 12. 12.5  1  4  0.  0.  1.
1908  0 1.  0 42.0 15.000  1  2 14. 20.0  1  4  0.  0.  1.
```

```
1916  0 1.  0 42.0 15.000  1  4 14. 20.0  5  4  0.  0.  1.
1918  0 1.  1 27.0  7.000  1  4 14.  7.5  3  3  0.  0.  1.
1920  0 1.  1 27.0  7.000  1  2 20. 20.0  6  2  0.  0.  1.
1930  0 1.  0 42.0 15.000  1  3 12. 20.0  3  3  0.  0.  1.
1940  0 1.  1 27.0  4.000  1  3 16.  7.5  3  5  0.  0.  1.
1947  0 1.  0 27.0  7.000  1  3 14. 40.0  1  4  0.  0.  1.
1949  0 1.  0 22.0  1.500  0  2 14. 12.5  4  5  0.  0.  1.
1951  0 1.  0 27.0  4.000  1  4 14. 12.5  1  4  0.  0.  1.
1952  0 1.  0 22.0  4.000  0  4 14.  4.0  5  5  0.  0.  1.
1960  0 1.  0 22.0  1.500  0  2 16. 20.0  4  5  0.  0.  1.
9001  0 1.  1 47.0 15.000  0  4 14. 12.5  5  4  0.  0.  1.
9012  0 1.  1 37.0 10.000  1  2 18. 12.5  6  2  0.  0.  1.
9023  0 1.  1 37.0 15.000  1  3 17. 40.0  5  4  0.  0.  1.
9029  0 1.  0 27.0  4.000  1  2 16.  7.5  1  4  0.  0.  1.
   6  3 1.  1 27.0  1.500  0  3 18. 20.0  4  4  3.  0.  1.
  12  3 1.  0 27.0  4.000  1  3 17. 12.5  1  5  3.  0.  1.
  43  4 1.  1 37.0 15.000  1  5 18. 12.5  6  2  7.  0.  1.
  53  6 1.  0 32.0 10.000  1  3 17. 40.0  5  2 12.  0.  1.
  67  1 1.  1 22.0  0.125  0  4 16. 20.0  5  5  1.  0.  1.
  79  1 1.  0 22.0  1.500  1  2 14.  4.0  1  5  1.  0.  1.
 122  6 1.  1 37.0 15.000  1  4 14.  7.5  5  2 12.  0.  1.
 126  4 1.  0 22.0  1.500  0  2 14.  4.0  3  4  7.  0.  1.
 133  2 1.  1 37.0 15.000  1  2 18. 20.0  6  4  2.  0.  1.
 138  3 1.  0 32.0 15.000  1  4 12. 12.5  3  2  3.  0.  1.
 154  1 1.  0 37.0 15.000  1  4 14. 12.5  4  2  1.  0.  1.
 159  4 1.  0 42.0 15.000  1  3 17. 40.0  1  4  7.  0.  1.
 174  6 1.  0 42.0 15.000  1  5  9. 12.5  4  1 12.  0.  1.
 176  5 1.  1 37.0 10.000  1  2 20. 40.0  6  2 12.  0.  1.
 181  6 1.  0 32.0 15.000  1  3 14. 12.5  1  2 12.  0.  1.
 182  3 1.  1 27.0  4.000  0  1 18.  7.5  6  5  3.  0.  1.
 186  4 1.  1 37.0 10.000  1  2 18. 40.0  7  3  7.  0.  1.
 189  4 1.  0 27.0  4.000  0  3 17.  7.5  5  5  7.  0.  1.
 204  1 1.  1 42.0 15.000  1  4 16. 20.0  5  5  1.  0.  1.
 215  1 1.  0 47.0 15.000  1  5 14. 12.5  4  5  1.  0.  1.
 232  4 1.  0 27.0  4.000  1  3 18. 12.5  5  4  7.  0.  1.
 233  1 1.  0 27.0  7.000  1  5 14.  4.0  1  4  1.  0.  1.
 252  6 1.  1 27.0  1.500  1  3 17. 20.0  5  4 12.  0.  1.
 253  5 1.  0 27.0  7.000  1  4 14. 12.5  6  2 12.  0.  1.
 274  3 1.  0 42.0 15.000  1  4 16. 20.0  5  4  3.  0.  1.
 275  4 1.  0 27.0 10.000  1  4 12. 12.5  7  3  7.  0.  1.
 287  1 1.  1 27.0  1.500  0  2 18. 12.5  5  2  1.  0.  1.
 288  1 1.  1 32.0  4.000  0  4 20. 20.0  6  4  1.  0.  1.
 325  1 1.  0 27.0  7.000  1  3 14.  7.5  1  3  1.  0.  1.
 328  3 1.  0 32.0 10.000  1  4 14.  7.5  1  4  3.  0.  1.
 344  3 1.  1 27.0  4.000  1  2 18. 12.5  7  2  3.  0.  1.
 353  1 1.  0 17.5  0.750  0  5 14.  7.5  4  5  1.  0.  1.
 354  1 1.  0 32.0 10.000  1  4 18. 20.0  1  5  1.  0.  1.
 367  4 1.  0 32.0  7.000  1  2 17. 20.0  6  4  7.  0.  1.
 369  4 1.  1 37.0 15.000  1  2 20. 20.0  6  4  7.  0.  1.
 390  4 1.  0 37.0 10.000  0  1 20. 20.0  5  3  7.  0.  1.
 392  5 1.  0 32.0 10.000  1  2 16. 12.5  5  5 12.  0.  1.
 423  4 1.  1 52.0 15.000  1  2 20. 40.0  6  4  7.  0.  1.
 432  4 1.  0 42.0 15.000  1  1 12. 12.5  1  3  7.  0.  1.
 436  1 1.  1 52.0 15.000  1  2 20. 40.0  6  3  1.  0.  1.
 483  2 1.  1 37.0 15.000  1  3 18. 12.5  6  5  2.  0.  1.
 513  5 1.  0 22.0  4.000  0  3 12. 12.5  3  4 12.  0.  1.
 516  6 1.  1 27.0  7.000  1  1 18. 12.5  6  2 12.  0.  1.
 518  1 1.  1 27.0  4.000  1  3 18. 12.5  5  5  1.  0.  1.
 520  7 1.  1 47.0 15.000  1  4 17. 12.5  6  5 12.  0.  1.
 526  6 1.  0 42.0 15.000  1  4 12. 20.0  1  1 12.  0.  1.
 528  4 1.  1 27.0  4.000  0  3 14.  7.5  3  4  7.  0.  1.
 553  4 1.  0 32.0  7.000  1  4 18. 12.5  4  5  7.  0.  1.
 576  1 1.  1 32.0  0.417  1  3 12.  7.5  3  4  1.  0.  1.
 611  3 1.  1 47.0 15.000  1  5 16. 12.5  5  4  3.  0.  1.
 625  5 1.  1 37.0 15.000  1  2 20. 40.0  5  4 12.  0.  1.
 635  4 1.  1 22.0  4.000  1  2 17.  7.5  6  4  7.  0.  1.
 646  1 1.  1 27.0  4.000  0  2 14. 12.5  4  5  1.  0.  1.
 657  4 1.  0 52.0 15.000  1  5 16. 12.5  1  3  7.  0.  1.
 659  1 1.  1 27.0  4.000  0  3 14. 12.5  3  3  1.  0.  1.
 666  1 1.  0 27.0 10.000  1  4 16. 12.5  1  4  1.  0.  1.
 679  1 1.  1 32.0  7.000  1  3 14. 20.0  7  4  1.  0.  1.
 729  4 1.  1 32.0  7.000  1  2 18. 20.0  4  1  7.  0.  1.
```

```
 755  3 1.  1 22.0  1.500  0  1 14. 20.0  3  2  3.   0.   1.
 758  4 1.  1 22.0  4.000  1  3 18. 12.5  6  4  7.   0.   1.
 770  4 1.  1 42.0 15.000  1  4 20. 40.0  6  4  7.   0.   1.
 786  2 1.  0 57.0 15.000  1  1 18. 40.0  5  4  2.   0.   1.
 797  4 1.  0 32.0  4.000  1  3 18. 12.5  5  2  7.   0.   1.
 811  1 1.  1 27.0  4.000  1  1 16. 20.0  4  4  1.   0.   1.
 834  4 1.  1 32.0  7.000  1  4 16. 12.5  1  4  7.   0.   1.
 858  2 1.  1 57.0 15.000  1  1 17. 20.0  4  4  2.   0.   1.
 885  4 1.  0 42.0 15.000  1  4 14. 20.0  5  2  7.   0.   1.
 893  4 1.  1 37.0 10.000  1  1 18. 12.5  5  3  7.   0.   1.
 927  3 1.  1 42.0 15.000  1  3 17. 12.5  6  1  3.   0.   1.
 928  1 1.  0 52.0 15.000  1  3 14. 12.5  4  4  1.   0.   1.
 933  2 1.  0 27.0  7.000  1  3 17. 12.5  5  3  2.   0.   1.
 951  6 1.  1 32.0  7.000  1  2 12. 20.0  4  2 12.   0.   1.
 968  1 1.  1 22.0  4.000  0  4 14.  7.5  2  5  1.   0.   1.
 972  3 1.  1 27.0  7.000  1  3 18. 12.5  6  4  3.   0.   1.
 975  6 1.  0 37.0 15.000  1  1 18. 12.5  5  5 12.   0.   1.
 977  4 1.  0 32.0 15.000  1  3 17. 20.0  1  3  7.   0.   1.
 981  4 1.  0 27.0  7.000  0  2 17. 20.0  5  5  7.   0.   1.
 986  1 1.  0 32.0  7.000  1  3 17. 12.5  5  3  1.   0.   1.
1002  1 1.  1 32.0  1.500  1  2 14.  7.5  2  4  1.   0.   1.
1007  6 1.  0 42.0 15.000  1  4 14. 12.5  1  2 12.   0.   1.
1011  4 1.  1 32.0 10.000  1  3 14. 20.0  5  4  7.   0.   1.
1035  4 1.  1 37.0  4.000  1  1 20. 20.0  6  3  7.   0.   1.
1050  1 1.  0 27.0  4.000  1  2 16. 12.5  5  3  1.   0.   1.
1056  5 1.  0 42.0 15.000  1  3 14. 20.0  4  3 12.   0.   1.
1057  1 1.  1 27.0 10.000  1  5 20. 20.0  6  5  1.   0.   1.
1075  6 1.  1 37.0 10.000  1  2 20. 40.0  6  2 12.   0.   1.
1080  6 1.  0 27.0  7.000  1  1 14. 12.5  3  3 12.   0.   1.
1125  3 1.  0 27.0  7.000  1  4 12.  7.5  1  2  3.   0.   1.
1131  3 1.  1 32.0 10.000  1  2 14. 12.5  4  4  3.   0.   1.
1138  7 1.  0 17.5  0.750  1  2 12.  7.5  1  3 12.   0.   1.
1150  5 1.  0 32.0 15.000  1  3 18. 40.0  5  4 12.   0.   1.
1163  2 1.  0 22.0  7.000  0  4 14. 20.0  4  3  2.   0.   1.
1169  1 1.  1 32.0  7.000  1  4 20. 20.0  6  5  1.   0.   1.
1198  4 1.  1 27.0  4.000  1  2 18. 12.5  6  2  7.   0.   1.
1204  1 1.  0 22.0  1.500  1  5 14.  7.5  5  3  1.   0.   1.
1218  7 1.  0 32.0 15.000  0  3 17.  7.5  5  1 12.   0.   1.
1230  5 1.  0 42.0 15.000  1  2 12. 20.0  1  2 12.   0.   1.
1236  4 1.  1 42.0 15.000  1  3 20. 40.0  5  4  7.   0.   1.
1247  6 1.  1 32.0 10.000  0  2 18. 20.0  4  2 12.   0.   1.
1259  5 1.  0 32.0 15.000  1  3  9. 12.5  1  1 12.   0.   1.
1294  4 1.  1 57.0 15.000  1  5 20. 12.5  4  5  7.   0.   1.
1353  5 1.  1 47.0 15.000  1  4 20. 40.0  6  4 12.   0.   1.
1370  2 1.  0 42.0 15.000  1  2 17. 20.0  6  3  2.   0.   1.
1427  6 1.  1 37.0 15.000  1  3 17. 40.0  6  3 12.   0.   1.
1445  5 1.  1 37.0 15.000  1  5 17. 12.5  5  2 12.   0.   1.
1460  4 1.  1 27.0 10.000  1  2 20. 12.5  6  4  7.   0.   1.
1480  2 1.  1 37.0 15.000  1  2 16. 20.0  5  4  2.   0.   1.
1505  6 1.  0 32.0 15.000  1  1 14. 20.0  5  2 12.   0.   1.
1543  4 1.  1 32.0 10.000  1  3 17.  7.5  6  3  7.   0.   1.
1548  2 1.  1 37.0 15.000  1  4 18. 12.5  5  1  2.   0.   1.
1550  4 1.  0 27.0  1.500  0  2 17. 12.5  5  5  7.   0.   1.
1561  3 1.  0 47.0 15.000  1  2 17. 20.0  5  2  3.   0.   1.
1564  6 1.  1 37.0 15.000  1  2 17. 20.0  5  4 12.   0.   1.
1573  5 1.  0 27.0  4.000  0  2 14. 20.0  5  5 12.   0.   1.
1575  2 1.  0 27.0 10.000  1  4 14. 12.5  1  5  2.   0.   1.
1599  1 1.  0 22.0  4.000  1  3 16. 12.5  1  3  1.   0.   1.
1622  6 1.  1 52.0  7.000  0  4 16.  7.5  5  5 12.   0.   1.
1629  2 1.  0 27.0  4.000  1  1 16.  7.5  3  5  2.   0.   1.
1664  4 1.  0 37.0 15.000  1  2 17. 40.0  6  4  7.   0.   1.
1669  2 1.  0 27.0  4.000  0  1 17.  4.0  3  1  2.   0.   1.
1674  7 1.  0 17.5  0.750  1  2 12.  7.5  3  5 12.   0.   1.
1682  4 1.  0 32.0 15.000  1  5 18. 12.5  5  4  7.   0.   1.
1685  4 1.  0 22.0  4.000  0  1 16.  7.5  3  5  7.   0.   1.
1697  2 1.  1 32.0  4.000  1  4 18. 20.0  6  4  2.   0.   1.
1716  1 1.  0 22.0  1.500  1  3 18. 20.0  5  2  1.   0.   1.
1730  3 1.  0 42.0 15.000  1  2 17. 40.0  5  4  3.   0.   1.
1731  1 1.  1 32.0  7.000  1  4 16. 12.5  4  4  1.   0.   1.
1732  5 1.  1 37.0 15.000  0  3 14. 20.0  6  2 12.   0.   1.
1743  1 1.  1 42.0 15.000  1  3 16. 40.0  6  3  1.   0.   1.
1751  1 1.  1 27.0  4.000  1  1 18.  7.5  5  4  1.   0.   1.
```

```
   1757  2 1.  1 37.0 15.000  1  4 20. 40.0  7  3  2.  0.  1.
   1763  4 1.  1 37.0 15.000  1  3 20. 40.0  6  4  7.  0.  1.
   1766  3 1.  1 22.0  1.500  0  2 12. 12.5  3  3  3.  0.  1.
   1772  3 1.  1 32.0  4.000  1  3 20. 20.0  6  2  3.  0.  1.
   1776  2 1.  1 32.0 15.000  1  5 20. 20.0  6  5  2.  0.  1.
   1782  5 1.  0 52.0 15.000  1  1 18. 40.0  5  5 12.  0.  1.
   1784  5 1.  1 47.0 15.000  0  1 18. 40.0  6  5 12.  0.  1.
   1791  3 1.  0 32.0 15.000  1  4 16. 12.5  4  4  3.  0.  1.
   1831  4 1.  0 32.0 15.000  1  3 14. 12.5  3  2  7.  0.  1.
   1840  4 1.  0 27.0  7.000  1  4 16. 20.0  1  2  7.  0.  1.
   1844  5 1.  1 42.0 15.000  1  3 18. 12.5  6  2 12.  0.  1.
   1856  4 1.  0 42.0 15.000  1  2 14. 12.5  3  2  7.  0.  1.
   1876  5 1.  1 27.0  7.000  1  2 17. 12.5  5  4 12.  0.  1.
   1929  3 1.  1 32.0 10.000  1  4 14.  7.5  4  3  3.  0.  1.
   1935  4 1.  1 47.0 15.000  1  3 16. 20.0  4  2  7.  0.  1.
   1938  1 1.  1 22.0  1.500  1  1 12.  7.5  2  5  1.  0.  1.
   1941  4 1.  0 32.0 10.000  1  2 18.  7.5  5  4  7.  0.  1.
   1954  2 1.  1 32.0 10.000  1  2 17. 20.0  6  5  2.  0.  1.
   1959  2 1.  1 22.0  7.000  1  3 18. 20.0  6  2  2.  0.  1.
   9010  1 1.  0 32.0 15.000  1  3 14. 40.0  1  5  1.  0.  1.
;

*
* create a subset of id's: only of those that are actually used
* in the data set
*

set i(id)  'used records';
i(id)$sum(v$data(id,v),1) = yes;
*display i;

*
* add constant term
*
data(i,'const') = 1;

*
* sanity check. this should be 601
*
scalar n;
n=card(i);
display n;



*-------------------------------------------------------------------------------
* OLS regression
*-------------------------------------------------------------------------------

set j(v) 'independent variables' /const,z2,z3,z5,z7,z8/;

*
* set up parameters X, y for easier manipulations further on
*
parameter y(id),x(id,j);
y(i) = data(i,'y');
x(i,j) = data(i,j);

variables coeff(j);
variable sse 'sum of squared errors';

equations
   sumsq  'dummy objective'
   fit(id)
;

sumsq..  sse =n= 0;

fit(i)..  y(i) =e= sum(j, x(i,j)*coeff(j));

model ols /sumsq,fit/;
option lp=ls;
```

```
solve ols using lp minimizing sse;

display "OLS solution",coeff.l;



*-------------------------------------------------------------------------------
* Tobit model
*-------------------------------------------------------------------------------

variables
   loglike
   sigma
;

equations
   objective
;


objective.. loglike =e=
    sum(i$(y(i)>0), -0.5*sqr[y(i)-sum(j,x(i,j)*coeff(j))]/sqr(sigma) - log(sigma) )
    + sum(i$(y(i)=0), log(1-errorf(sum(j,x(i,j)*coeff(j))/sigma)))
    - sum(i$(y(i)>0), log(sqrt(2*pi)));

*
* initial value
*
sigma.l = 1;

model tobit /objective/;
solve tobit using nlp maximizing loglike;

display "Tobit model",coeff.l;
```

The reported solution is:

```
----    680 OLS solution

----    680 VARIABLE coeff.L

const  5.608,    Z2   -0.050,    Z3    0.162,    Z5   -0.476,    Z7    0.106,    Z8   -0.712

----    711 Tobit model

----    711 VARIABLE coeff.L

const  8.174,    Z2   -0.179,    Z3    0.554,    Z5   -1.686,    Z7    0.326,    Z8   -2.285
```

The same model estimated with Gretl gives:

```
gretl version 1.3.0
Current session: 2005/01/06 18:21
# Fair's extra-marital affairs data
? open greene22_2.gdt

Read datafile /usr/share/gretl/data/greene/greene22_2.gdt
periodicity: 1, maxobs: 601,
observations range: 1-601

Listing 10 variables:
  0) const       1) Y           2) Z1          3) Z2          4) Z3
  5) Z4          6) Z5          7) Z6          8) Z7          9) Z8

# initial OLS
? ols Y 0 Z2 Z3 Z5 Z7 Z8

Model 1: OLS estimates using the 601 observations 1-601
Dependent variable: Y
```

```
      VARIABLE      COEFFICIENT        STDERROR       T STAT    2Prob(t > |T|)

   0)     const         5.60816           0.796599      7.040    < 0.00001 ***
   3)        Z2        -0.0503473         0.0221058    -2.278     0.023107 **
   4)        Z3         0.161852          0.0368969     4.387     0.000014 ***
   6)        Z5        -0.476324          0.111308     -4.279     0.000022 ***
   8)        Z7         0.106006          0.0711007     1.491     0.136510
   9)        Z8        -0.712242          0.118289     -6.021    < 0.00001 ***

  Mean of dependent variable = 1.45591
  Standard deviation of dep. var. = 3.29876
  Sum of squared residuals = 5671.09
  Standard error of residuals = 3.08727
  Unadjusted R-squared = 0.13141
  Adjusted R-squared = 0.124111
  F-statistic (5, 595) = 18.0037 (p-value < 0.00001)

  MODEL SELECTION STATISTICS

  SGMASQ         9.53125     AIC           9.62640     FPE           9.62640
  HQ             9.79236     SCHWARZ       10.0585     SHIBATA       9.62450
  GCV            9.62736     RICE          9.62834

Excluding the constant, p-value was highest for variable 8 (Z7)

# Tobit version
? tobit Y 0 Z2 Z3 Z5 Z7 Z8
Convergence achieved after 100 iterations

Model 2: Tobit estimates using the 601 observations 1-601
Dependent variable: Y

      VARIABLE      COEFFICIENT        STDERROR       T STAT    2Prob(t > |T|)

   0)     const         8.17411           2.60908       3.133     0.001816 ***
   3)        Z2        -0.179330          0.0757210    -2.368     0.018189 **
   4)        Z3         0.554137          0.140708      3.938     0.000092 ***
   6)        Z5        -1.68621           0.413967     -4.073     0.000053 ***
   8)        Z7         0.326052          0.264723      1.232     0.218558
   9)        Z8        -2.28496           0.443770     -5.149    < 0.00001 ***

  Mean of dependent variable = 1.45591
  Standard deviation of dep. var. = 3.29876
  Censored observations: 451 (75.0%)
  sigma = 8.24703
  Log-likelihood = -705.576

Test for normality of residual -
  Null hypothesis: error is normally distributed
  Test statistic: Chi-square(2) = 14.3781
  with p-value = 0.000754799
```

Some authors suggest a transformation that guarantee global optima [31] while other evidence suggests this is not needed [18].

11.9. **Probit.** A discrete choice model

$$P(Y=1|X=x)=\Phi(x'\beta) \tag{36}$$

where Y is a binary dependent variable and X is a vector of independent variables can be estimated using maximum likelihood:

$$\ln\ L(\beta)=\sum_{i=1}^{n}\{y_i \ln\ \Phi(x'\beta)+(1-y_i)\ln[1-\Phi(x'\beta)]\} \tag{37}$$

This can be rewritten as:

$$\ln\ L(\beta) = \sum_{y_i=1} \ln\ \Phi(x'\beta) + \sum_{y_i=0} \ln[1 - \Phi(x'\beta)] \tag{38}$$

This is can be solved as an NLP model. It is often useful to provide a good starting point. In this case the OLS estimators are a good candidate to produce initial values for the NLP solver. The example below is from [18].

11.9.1. *Model probit.gms.* [18]

```
$ontext

    Probit Estimation
    We use OLS to get starting point

    Erwin Kalvelagen, Amsterdam Optimization, 2009

    Data:
       http://pages.stern.nyu.edu/~wgreene/Text/tables/TableF21-1.txt

$offtext

set i /1*32/;

table data(i,*)

        GPA      TUCE     PSI      GRADE
  1      2.66      20       0        0
  2      2.89      22       0        0
  3      3.28      24       0        0
  4      2.92      12       0        0
  5      4.00      21       0        1
  6      2.86      17       0        0
  7      2.76      17       0        0
  8      2.87      21       0        0
  9      3.03      25       0        0
 10      3.92      29       0        1
 11      2.63      20       0        0
 12      3.32      23       0        0
 13      3.57      23       0        0
 14      3.26      25       0        1
 15      3.53      26       0        0
 16      2.74      19       0        0
 17      2.75      25       0        0
 18      2.83      19       0        0
 19      3.12      23       1        0
 20      3.16      25       1        1
 21      2.06      22       1        0
 22      3.62      28       1        1
 23      2.89      14       1        0
 24      3.51      26       1        0
 25      3.54      24       1        1
 26      2.83      27       1        1
 27      3.39      17       1        1
 28      2.67      24       1        0
 29      3.65      21       1        1
 30      4.00      23       1        1
 31      3.10      21       1        0
 32      2.39      19       1        1

;

set k 'independent variables' /constant,gpa,tuce,psi/;

parameters
  y(i)    'grade'
  x(k,i)  'independent variables'
;
```

[18] www.amsterdamoptimization.com/models/regression/probit.gms

```
y(i) = data(i,'grade');
x('constant',i) = 1;
x(k,i)$(not sameas(k,'constant')) = data(i,k);


parameter estimate(k,*);

*-----------------------------------------------------------
* O L S
*-----------------------------------------------------------

variable sse,coeff(k);
equation obj,fit(i);


obj.. sse =n= 0;
fit(i).. y(i) =e= sum(k, coeff(k)*x(k,i));

model ols /obj,fit/;
option lp=ls;
solve ols using lp minimizing sse;

estimate(k,'OLS') = coeff.l(k);

*-----------------------------------------------------------
* P R O B I T
*-----------------------------------------------------------

variable logl;
equation like;

like.. logl =e= sum(i$(y(i)=1), log(errorf(sum(k,coeff(k)*x(k,i)))))
                +sum(i$(y(i)=0), log(1-errorf(sum(k,coeff(k)*x(k,i)))));

model mle /like/;
solve mle using nlp maximizing logl;

estimate(k,'Probit') = coeff.l(k);

display estimate;
```

The results are identical to the numbers in [18]:

```
----     99 PARAMETER estimate

                 OLS      Probit

GPA            0.464       1.626
TUCE           0.010       0.052
PSI            0.379       1.426
constant      -1.498      -7.452
```

11.10. **Bootstrap.** The model below implements a bootstrapping algorithm for the Data Envelopment Analysis Problem. DEA is a methodology to estimate efficient frontiers [5, 6, 15].

Bootstrapping[12, 35] is used to provide additional information for statistical inference. The following model from [42] implements a resampling strategy from [34]. Two thousand bootstrap samples are formed, each resulting in a DEA model of 100 small LP's. In this example we batch the DEA models together in a single large LP, so that we only have to solve 2,000 models instead of 200,000. The formulation trick is explained in [22].

Notice also the use of `eff.l(i)` in the regression model. This will be the current level value of the variable `eff` which is used in the preceeding DEA model. The extension `.l` will cause this to be a constant (i.e. data) instead of a decision variable.

11.10.1. *Model bootstrap.gms.* [19]

```
$ontext

   DEA bootstrapping example

   Erwin Kalvelagen, october 2004

   References:

     Mei Xue, Patrick T. Harker
     "Overcoming the Inherent Dependency of DEA Efficiency Scores:
     A Bootstrap Approach", Tech. Report, Department of Operations and
     Information Management, The Wharton School, University of Pennsylvania,
     April 1999

     http://opim.wharton.upenn.edu/~harker/DEAboot.pdf

$offtext


sets
   i 'hospital (DMU)' /h1*h100/
   j 'inputs and outputs' /
     FTE     'The number of full time employees in the hospital in FY 1994-95'
     Costs   'The expenses of the hospital ($million) in FY 1994-95'
     PTDAYS  'The number of the patient days produced by the hospital in FY 1994-95'
     DISCH   'The number of patient discharges produced by the hospital in FY 1994-95'
     BEDS    'The number of patient beds in the hospital in FY 1994-95'
     FORPROF 'Dummy variable, one if it is for-profit hospital, zero otherwise'
     TEACH   'Dummy variable, one if it is teaching hospital, zero otherwise'
     RES     'The number of the residents in the hospital in FY 1994-95'
     CONST   'Constant term in regression model'
   /
   inp(j) 'inputs' /FTE,Costs/
   outp(j) 'outputs' /PTDAYS,DISCH/
;

table data(i,j)

          FTE      Costs        PTDAYS   DISCH   BEDS FORPROF TEACH   RES

  h1     1571.86    174          71986   12665   365
  h2      816.54     69.9        53081    5861   224
  h3      533.74     61.7        25030    4951   286     1
  h4      805.2      75.4        34163   11877   256
  h5     3908.1     396         187462   42735   829            1    136.8
  h6      727.72     63.9        31330    8402   194
  h7     2571.75    220         130077   26877   620            1     42.81
  h8      521        89.1        43390    8598   290     1
  h9      718        50          27896    6113   150            1     23.21
  h10    1504.85    121          75941   16427   393
  h11    1234.49     84.6        57080   14180   317
  h12     873        68.8        48932   12060   281
  h13    1067.17     85.8        50436   11317   278
  h14     668        47.5        67909    6235   244
  h15     452.35     36.4        25200    6860   155     1      1     13.31
  h16    1523        97.4        59809   13180   394
  h17    3152       198         108631   22071   578            1    195.67
  h18     871.96     30.7        17925    4605   160
  h19    2901.86    290         130004   24133   549            1    126.89
  h20     902.4      78.2        35743    8664   236            1     12.08
  h21     194.69     10.9        15555    1530   132
  h22     713.51     62.6        32558    8966   138
  h23    557.36      23.8        12728    2291   276     1
  h24   2259.2      120          74061   12942   348            1     14.52
  h25    462.22      32.4        28886    6101   134
  h26   1212.1       97.3        74194   12681   342
  h27   2391.94     192          89843   18396   336            1    229.19
  h28   1637        162          80468   21345   415
```

[19] www.amsterdamoptimization.com/models/regression/bootstrap.gms

h29	501	37.9	26813	4594	166	1		
h30	412.1	40.2	23217	6044	160	1		
h31	738.56	27	11514	3052	144	1		
h32	414.1	35.7	55611	4354	200			
h33	1097	105	59443	13101	242		1	26.32
h34	742	62.8	42542	8739	172			
h35	1010	97.1	47246	12073	269		1	1.1
h36	440.6	34.2	30773	4305	201			
h37	1203.3	85.4	50710	11470	247		1	13.82
h38	2558.01	195	128450	20441	571		1	5.42
h39	215.45	8.409936	65743	578	238			
h40	599.3	30.4	23299	5338	173			
h41	480.55	29.5	34279	6560	169	1		
h42	634.51	29.9	27157	5198	141			
h43	1211.9	91.4	90008	17666	320		1	6.25
h44	285.5	23.9	16473	2873	135			
h45	1030.36	67.1	43486	9467	235		1	6.44
h46	1374.81	95.5	74279	11862	284			
h47	953.56	49.8	47934	10553	207			
h48	561.11	41.7	24800	5498	132			
h49	644	57.1	39663	8604	260			
h50	376.55	19.6	22003	4759	143			
h51	404.79	32.8	27566	7871	190	1		
h52	397.9	29.4	26072	4248	170			
h53	374.2	3.944649	4179	819	156			
h54	1702	100	114603	17235	438		1	11.81
h55	148.09	5.013379	51660	771	172			
h56	253.48	16.9	17599	4044	178			
h57	1445.68	99.3	81041	12912	475		1	17.53
h58	414.1	26.5	20432	4068	129			
h59	642.58	48.5	42733	5983	181	1		
h60	203.75	13	16923	3467	146	1		
h61	421.8	18.3	16179	2840	160			
h62	320.62	17.3	18882	3370	160			
h63	679.79	25.6	27561	4447	308		1	11.33
h64	2382	226	166559	26019	787		1	7.08
h65	559.29	58.1	40534	8806	342	1		
h66	568.15	35	37120	7242	158			
h67	2408.04	155	70392	9538	266		1	111.33
h68	632.34	54.6	37228	6359	175			
h69	917.22	55.2	42135	7294	215			
h70	554.34	56.9	32352	3320	205		1	1
h71	780	75.9	39213	7154	172			
h72	663.82	56.9	34180	5284	200			
h73	1424	146	107457	18198	432		1	2.75
h74	313	20.7	20110	5967	165	1		
h75	778	78.4	51496	12302	390			
h76	863.37	62	50957	10557	228			
h77	3509.12	290	109673	19213	469		1	290.53
h78	1593.82	152	82400	17707	474		1	11.64
h79	466	40.1	30647	7265	164	1		
h80	666.38	48.2	28048	5182	153			
h81	998.8	121	45513	6855	238		1	88.86
h82	1018	98.2	61176	11386	350			
h83	3238.28	326	122118	19068	514		1	146.33
h84	1431.1	107	48900	10623	208			
h85	1735.99	273	84118	16458	278		1	158.4
h86	1769	190	105741	19256	478		1	0.93
h87	484.56	36.2	24070	6464	125			
h88	204.7	13.9	28137	1615	135	1		
h89	1706.58	287	75153	13465	367		1	91.56
h90	1029.11	71.9	49993	6690	252		1	4
h91	1167.2	111	75004	21334	350			
h92	1657.58	116	77753	17528	413			
h93	1017.16	88.5	64147	11135	316			
h94	1532.7	153	99998	17391	395		1	4.8
h95	1462	113	119107	16053	484		1	0.5
h96	1133.8	109	55540	15566	355		1	8.51
h97	609	48.2	71817	5639	376		1	1
h98	301.31	20.2	43214	2153	141			
h99	1930.08	201	87197	19315	418			
h100	1573.3	177	88124	19661	458		1	69.71

```
;

data(i,'CONST') = 1;


*-----------------------------------------------------------------------------
* PHASE 1: Estimation of b(j)
*
* Run standard Constant Returns to Scale (CCR) Input-oriented DEA model
* followed by linear regression OLS estimation
*-----------------------------------------------------------------------------



*
* this is the standard DEA model
* instead of 100 small models we solve one big model, see
* http://www.gams.com/~erwin/dea/dea.pdf
*
parameter
  x(inp,i)  'inputs of DMU i'
  y(outp,i) 'outputs of DMU i'
;

alias(i,j0);
positive variables
  v(inp,j0)    'input weights'
  u(outp,j0)   'output weights'
;
variable
  eff(j0) 'efficiency'
  z 'objective variable'
;

equations
  objective(j0)  'objective function: maximize efficiency'
  normalize(j0)  'normalize input weights'
  limit(i,j0)    "limit other DMU's efficiency"
  totalobj
;

totalobj..      z =e= sum(j0, eff(j0));
objective(j0).. eff(j0) =e= sum(outp, u(outp,j0)*y(outp,j0));
normalize(j0).. sum(inp, v(inp,j0)*x(inp,j0)) =e= 1;
limit(i,j0)..   sum(outp, u(outp,j0)*y(outp,i)) =l= sum(inp, v(inp,j0)*x(inp,i));

model dea /totalobj,objective, normalize, limit/;

alias (i,iter);

x(inp,i) = data(i,inp);
y(outp,i) = data(i,outp);

option limrow=0;
option limcol=0;
dea.solprint=2;
dea.solvelink=2;

option lp=cplex;
solve dea using lp maximizing z;
abort$(dea.modelstat<>1) "LP was not optimal";

display
  "----------------------------------- DEA MODEL ------------------------",
  eff.l;

*
* now solve the regression problem
*   efficiency = b0 + b1*BEDS + b2*FORPROF + b3*TEACH + b4*RES
* See http://www.gams.com/~erwin/regression/regression.pdf
*

set e(j) 'explanatory variables' /BEDS,FORPROF,TEACH,RES,CONST/;
```

```
parameter Xmat(i,e) 'regression data matrix';
Xmat(i,e) = data(i,e);


variable
    sse  'sum of squared errors'
    b(e) 'coefficients to be estimated'
;


equation
    sumsq  'dummy objective function'
    fit(i) 'regression equation'
;

sumsq..   sse =n= 0;

fit(i)..  eff.l(i) =e= sum(e, Xmat(i,e)*b(e));

model regression /sumsq,fit/;
regression.solprint=0;
regression.solvelink=2;
option lp=ls;
solve regression using lp minimizing sse;

*
* standard errors and pvalues
*
parameters
   bhat(e) 'estimates'
   se(e)   'standard error'
   tval(e) 't-values'
   pval(e) 'p-values: Pr(>|t|)'
;
bhat(e) = b.l(e);
se(e) = b.m(e);
execute_load 'ls.gdx',tval,pval;

parameter ols(e,*);
ols(e,'estimates') = bhat(e);
ols(e,'std.error') = se(e);
ols(e,'t value') = tval(e);
ols(e,'p value') = pval(e);

display
  "----------------------------------- OLS MODEL -----------------------",
  ols;

*-------------------------------------------------------------------------------
* PHASE 2: BOOTSTRAP algorithm
*-------------------------------------------------------------------------------

set s 'sample' /sample1*sample2000/;


parameter bs(s,i) 'bootstrap sample';
bs(s,i) = trunc( uniform(1,card(i)+0.999999999) );
*display bs;
* sanity check:
loop((s,i),
   abort$(bs(s,i)<1) "Check bs for entries < 1";
   abort$(bs(s,i)>card(i)) "Check bs for entries > card(i)";
);

alias(i,ii);
set mapbs(s,i,ii);
mapbs(s,i,ii)$(bs(s,i) = ord(ii)) = yes;
* this mapping says the i'th sample data record is the ii'th record
* in the original data (for sample s)

loop((s,i),
  abort$(sum(mapbs(s,i,ii),1)<>1) "mapbs is not unique";
```

```
);

parameter data_sample(i,j);

parameter sb(s,e) 'b(e) for each sample s';

* reduce printing to listing file:
regression.solprint=2;

loop(s,

*
* solve dea model
*

    data_sample(i,j) = sum(mapbs(s,i,ii),data(ii,j));
    x(inp,i) = data_sample(i,inp);
    y(outp,i) = data_sample(i,outp);

    option lp=cplex;
    solve dea using lp maximizing z;
    abort$(dea.modelstat<>1) "LP was not optimal";

*
* solve OLS model
*
    Xmat(i,e) = data_sample(i,e);
    option lp=ls;
    solve regression using lp minimizing sse;
    sb(s,e) = b.l(e);


);



*
* get statistics
*
parameter bbar(e) "Averaged estimates";
bbar(e) = sum(s, sb(s,e)) / card(s);

parameter sehat(e) "Standard errors of bootstrap algorithm";
sehat(e) = sqrt(sum(s, sqr(sb(s,e)-bbar(e)))/(card(s)-1));

parameter tbootstrap(e) "t statistic for bootstrap";
tbootstrap(e) = bhat(e)/sehat(e);

scalar df 'degrees of freedom';
df = card(i) - (card(e) - 1) - 1;
parameter pbootstrap(e) "p-values for bootstrap";

*
*  pvalue = 2 * pt( abs(tvalue), df)
*         = 2 * 0.5 * pbeta( df / (df + sqr(abs(tvalue))), df/2, 0.5)
*         = betareg( df / (df+sqr(tvalue)), df/2, 0.5)
*
pbootstrap(e) = betareg( df / (df+sqr(tbootstrap(e))), df/2, 0.5);

parameter bootstrap(e,*);
bootstrap(e,'estimates') = bhat(e);
bootstrap(e,'std.error') = sehat(e);
bootstrap(e,'t value') = tbootstrap(e);
bootstrap(e,'p value') = pbootstrap(e);


display
  "----------------------------------- BOOTSTRAP MODEL -----------------------",
  bootstrap;
```

default		**solvelink=2**	
real	27m12.745s	real	14m29.518s
user	20m58.595s	user	12m58.734s
sys	5m30.054s	sys	1m3.559s

TABLE 5. Solvelink results

The idea of this model is to build a regression equation:

$$\theta_i = \beta_0 + \beta_1 \text{BEDS}_i + \beta_2 \text{FORPROF}_i + \beta_3 \text{TEACH}_i + \beta_4 \text{RES}_i + \varepsilon_i \tag{39}$$

where θ_i are the DEA efficiency scores. From the results

```
      Parameter      Estimate     Std. Error        t value      Pr(>|t|)
      b('BEDS')   0.10400E-03    0.12441E-03    0.83599E+00    0.40526E+00
   b('FORPROF')   0.99344E-01    0.41567E-01    0.23900E+01    0.18824E-01 *
     b('TEACH')  -0.56803E-01    0.39159E-01   -0.14506E+01    0.15019E+00
       b('RES')  -0.10350E-02    0.33034E-03   -0.31330E+01    0.23012E-02 **
     b('CONST')   0.60741E+00    0.35049E-01    0.17330E+02    0.35975E-30 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

we see that FORPROF is significant at $\alpha = 0.05$. However when we apply the resampling technique from the bootstrap algorithm, the results indicate a different interpretation:

```
----    351 ------------------------------------ BOOTSTRAP MODEL ------------------------

----    351 PARAMETER bootstrap

          estimates   std.error     t value     p value

BEDS    1.040019E-4 1.107968E-4       0.939       0.350
FORPROF       0.099       0.060       1.651       0.102
TEACH        -0.057       0.036      -1.584       0.116
RES          -0.001 2.442416E-4      -4.237 5.234669E-5
CONST         0.607       0.042      14.417 1.18730E-25
```

Here the p-value for FORPROF is indicating this parameter is *not* significant at the 0.05 level. The p-values are calculated using the incomplete beta function which is available as `BetaReg()` in GAMS[24].

It is noted that the option `m.solvelink=2;` is quite effective for this model. This option makes solver calls more efficient by keeping GAMS in memory. Timings that illustrate this are reported in table 5.

11.11. **Large test.** This model tests the regression solver against a large data set with simulated data. The model consists of 20,000 cases and 100 parameters to be estimated.

> This test was introduced after feedback of users running estimations of this size. It is noted that the solver uses *dense* linear algebra routines. In most cases this is advantageous and LS will use (much) less memory to solve the model than GAMS needs to generate the model. If the matrix X in $y = X\beta + \varepsilon$ is very large and sparse this may not be the case, and the solver may need a lot of memory compared to GAMS.

11.11.1. *Model largereg.gms.* [20]

```
option sysout=on;
$ontext

   Test regression solver against a large simulated data set

$offtext

set i 'cases' /case1*case20000/;
set j 'parameters' /p0*p100/;
set j0(j) 'constant term' /p0/;
set j1(j) 'non-constant term' /p1*p100/;


parameter x(i,j) 'data, randomly generated; first column is constant term';
x(i,j0) = 1;
x(i,j1) = uniform(-100,100);

parameter p_sim(j) 'values of parameters to construct simulation';
p_sim(j) = ord(j);

parameter y(i) 'data, simulated';
y(i) = sum(j, p_sim(j)*x(i,j)) + normal(0,10);

variables
   p_est(j)  'parameters, to be estimated'
   sse       'sum of squared errors'
;
equation
   obj       'dummy objective'
   fit(i)    'equation we want to fit'
;

obj.. sse =n= 0;
fit(i).. y(i) =e= sum(j, p_est(j)*x(i,j));

$onecho > ls.opt
* increase default limits
maxn 30000
maxp 200
$offecho

option lp=ls;
model ols1 /obj,fit/;
ols1.optfile=1;
solve ols1 minimizing sse using lp;
```

11.12. **GDX Import.** These examples will show to import the variance-covariance matrix and the confidence intervals from the GDX file `ls.gdx`.

11.12.1. *Importing the covariance matrix.* The following model will run the longley problem (see section 11.2) and subsequently import the variance-covariance matrix from the GDX file.

11.12.2. *Model longleygdx.gms.* [21]

```
$ontext

  Longley Linear Least Squares benchmark problem.
  Load variance-covariance matrix from gdx file.

  Erwin Kalvelagen, dec 2004


  References:
```

[20] www.amsterdamoptimization.com/models/regression/largereg.gms
[21] www.amsterdamoptimization.com/models/regression/longleygdx.gms

```
      http://www.itl.nist.gov/div898/strd/lls/lls.shtml

      Longley, J. W. (1967).
      An Appraisal of Least Squares Programs for the
      Electronic Computer from the Viewpoint of the User.
      Journal of the American Statistical Association, 62, pp. 819-841.

$offtext

set i 'cases' /i1*i16/;
set v 'variables' /empl,const,gnpdefl,gnp,unempl,army,pop,year/;
set indep(v) 'independent variables' /const,gnpdefl,gnp,unempl,army,pop,year/;
set depen(v) 'dependent variables' /empl/;

table data(i,v)
         empl gnpdefl   gnp    unempl    army     pop    year
  i1    60323   83.0   234289   2356     1590    107608  1947
  i2    61122   88.5   259426   2325     1456    108632  1948
  i3    60171   88.2   258054   3682     1616    109773  1949
  i4    61187   89.5   284599   3351     1650    110929  1950
  i5    63221   96.2   328975   2099     3099    112075  1951
  i6    63639   98.1   346999   1932     3594    113270  1952
  i7    64989   99.0   365385   1870     3547    115094  1953
  i8    63761  100.0   363112   3578     3350    116219  1954
  i9    66019  101.2   397469   2904     3048    117388  1955
  i10   67857  104.6   419180   2822     2857    118734  1956
  i11   68169  108.4   442769   2936     2798    120445  1957
  i12   66513  110.8   444546   4681     2637    121950  1958
  i13   68655  112.6   482704   3813     2552    123366  1959
  i14   69564  114.2   502601   3931     2514    125368  1960
  i15   69331  115.7   518173   4806     2572    127852  1961
  i16   70551  116.9   554894   4007     2827    130081  1962
;

data(i,'const') = 1;

alias(indep,j,jj,k);

variables
   b(indep)  'parameters to be estimated'
   sse
;

equation
   fit(i)    'equation to fit'
   sumsq
;

sumsq..   sse =n= 0;
fit(i)..  data(i,'empl') =e= sum(indep, b(indep)*data(i,indep));

option lp = ls;
model leastsq /fit,sumsq/;
solve leastsq using lp minimizing sse;
option decimals=8;
display b.l;


parameter covariance(v,v);

execute_load 'ls.gdx',covariance=covar;
display covariance;
```

The matrix looks like:

```
----     71 PARAMETER covariance

              const     gnpdefl         gnp      unempl        army         pop        year

const   7.92848E+11 -1.54950E+7 2.433750E+4 3.635548E+5 1.048837E+5 -8.26713E+4 -4.05441E+8
gnpdefl -1.54950E+7 7.210545E+3 -1.84687274 -2.30172E+1 -6.34671065 12.65424072 7.204913E+3
gnp     2.433750E+4 -1.84687274  0.00112165  0.01546730  0.00336283 -0.00630855 -1.22292E+1
```

```
unempl  3.635548E+5 -2.30172E+1  0.01546730  0.23853425  0.06473378 -0.08372217 -1.83326E+2
army    1.048837E+5 -6.34671065  0.00336283  0.06473378  0.04591342 -0.00915133 -5.36167E+1
pop     -8.26713E+4 12.65424072 -0.00630855 -0.08372217 -0.00915133  0.05110909 39.96940026
year    -4.05441E+8 7.204913E+3 -1.22292E+1 -1.83326E+2 -5.36167E+1 39.96940026 2.074607E+5
```

11.12.3. *Confidence intervals.* This example shows how the confidence intervals can be read from the gdx file produced by the solver.

		LO	UP
90%	c	118.260825	174.911858
	h	-2.636830	-1.325447
	h3	0.000376	0.000479
95%	c	111.958934	181.213749
	h	-2.782709	-1.179568
	h3	0.000364	0.000490
97.5%	c	105.900284	187.272400
	h	-2.922958	-1.039320
	h3	0.000353	0.000501
99%	c	98.041195	195.131489
	h	-3.104883	-0.857394
	h3	0.000339	0.000516

FIGURE 7. GDX file

11.12.4. *Model weight.gms.* [22]

```
$ontext

   This model demonstrates how to import confidence
   intervals after running the regression.

   Data set:
     average heights and weights for American women aged 30-39
     (source: The World Almanac and Book of Facts, 1975).


$offtext

*-------------------------------------------------------------------------------
* data
*-------------------------------------------------------------------------------

set i /i1*i15/;
table data(i,*)
       height   weight
*      (in)     (lb)
  i1   58       115
  i2   59       117
  i3   60       120
  i4   61       123
```

[22] www.amsterdamoptimization.com/models/regression/weight.gms

```
  i5   62       126
  i6   63       129
  i7   64       132
  i8   65       135
  i9   66       139
  i10  67       142
  i11  68       146
  i12  69       150
  i13  70       154
  i14  71       159
  i15  72       164
;

*-----------------------------------------------------------------------------
* statistical model
*-----------------------------------------------------------------------------


variables
    c      'estimate constant term coefficient'
    h      'estimate height'
    h3     'estimate height^3'
    sse    'sum of squared errors'
;

equations
    fit(i)     'the linear model'
    obj        'objective'

;

obj..     sse =n= 0;
fit(i)..  data(i,'weight') =e= c + h*data(i,'height')+h3*data(i,'height')**3;

option lp=ls;
model ols1 /obj,fit/;
solve ols1 minimizing sse using lp;

display c.l, h.l, h3.l, sse.l;

*-----------------------------------------------------------------------------
* read confidence intervals from gdx file
*-----------------------------------------------------------------------------

sets
   alpha /'90%','95%','97.5%','99%'/
   names /'c','h','h3'/
   interval /'lo','up'/
;
parameter confint(alpha,names,interval);
execute_load 'ls.gdx',confint;

display confint;
```

The result is:

```
----     77 PARAMETER confint

                  lo          up

90%  .c      118.261     174.912
90%  .h       -2.637      -1.325
90%  .h3 3.758602E-4 4.788719E-4
95%  .c      111.959     181.214
95%  .h       -2.783      -1.180
95%  .h3 3.644011E-4 4.903310E-4
97.5%.c      105.900     187.272
97.5%.h       -2.923      -1.039
97.5%.h3 3.533844E-4 5.013478E-4
99%  .c       98.041     195.131
99%  .h       -3.105      -0.857
99%  .h3 3.390937E-4 5.156384E-4
```

The best way to find out how the sets `alpha`, `names`, and `interval` need to be populated is to inspect the gdx file `ls.gdx`. See figure 7 for an example.

> Note that it is needed to specify the sets `alpha`, `names`, and `interval`. Usually one can use the "`*`" to indicate to allow everything. However in this case the following code:

```
*-----------------------------------------------------------------------------
* read confidence intervals from gdx file
*-----------------------------------------------------------------------------

parameter confint(*,*,*);
execute_load 'ls.gdx',confint;
display confint;
```

> will not work as one would expect. GAMS will not process the GDX file correctly and will just display `( ALL 0.000 )`.

References

1. E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK users' guide*, third ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.
2. Herman J. Bierens, *The Tobit model*, September 2004.
3. Åke Björck, *Numerical Methods for Least Squares Problems*, SIAM, 1996.
4. Barry W. Brown and Lawrence B. Levy, *Certification of Algorithm 708: Significant Digit Computation of the Incomplete Beta*, ACM Transactions on Mathematical Software **20** (1994), no. 3, 393–397.
5. A. Charnes, W. W. Cooper, and E. Rhodes, *Measuring the efficiency of decision making units*, European Journal of Operational Research **2** (1978), 429–444.
6. A. Charnes, W. W. Cooper, and E. Rhodes, *Evaluating program and managerial efficiency: An application of data envelopment analysis to program follow through*, Management Science **27** (1981), 668–697.
7. Lee Creighton and Jianfeng Ding, *Assessing the Numerical Accuracy of JMP*, `http://www.jmp.com/products/NIST.pdf`.
8. Clint Cummins, *Extending the Longley benchmark results to 11 significant digits*, `http://www.stanford.edu/~clint/bench/longley.htm`, 1997.
9. Peter Dalgaard, *Introductory Statistics with R*, Springer, 2002.
10. Armido R. Didonato and Alfred H. Morris Jr., *Algorithm 708: Significant Digit Computation of the Incomplete Beta Function Ratios*, ACM Transactions on Mathematical Software **18** (1992), no. 3, 360–373.
11. Norman A. Draper and Harry Smith, *Applied Regression Analyis*, third ed., Wiley, 1998.
12. Bradley Efron and Robert J. Tibshirani, *An Introduction to the Bootstrap*, Chapman & Hall, 1993.
13. E. Elmroth and F. G. Gustavson, *Applying Recursion to Serial and Parallel QR Factorization Leads to Better Performance*, IBM J. Res. & Dev. **44** (2000), no. 4, 605–624.
14. ———, *A Faster and Simpler Recursive Algorithm for LAPACK Routine DGELS*, BIT **41** (2001), no. 5, 936–949.
15. Ali Emrouznejad, *Dea homepage*, `http://www.deazone.com/`, 2001.
16. Julian R. Faraway, *Linear Models with R*, CRC Press, 2004.
17. Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, 1996.
18. William H. Greene, *Multiple roots of the Tobit log-likelihood*, Journal of Econometrics **46** (1990), no. 3, 365–380.
19. ———, *Econometric Analysis*, 5th ed., Prentice-Hall, 2003.
20. W. E. Griffiths, R. C. Hill, and G. G. Judge, *Learning and Practicing Econometrics*, Wiley, 1993.
21. G. W. Hill, *ACM Algorithm 396: Student's t-Quantiles*, Commun. ACM **13** (1970), no. 10, 619–620.

22. Erwin Kalvelagen, *Efficiently Solving DEA Models with GAMS*, `http://www.amsterdamoptimization.com/pdf/dea.pdf`.
23. ———, *Least Squares Calculations with GAMS*, `http://www.amsterdamoptimization.com/pdf/ols.pdf`.
24. ———, *New special functions in GAMS*, `http://www.amsterdamoptimization.com/pdf/specfun.pdf`.
25. ———, *NLS: A Nonlinear Regression Solver for GAMS*, `http://www.amsterdamoptimization.com/pdf/nlregression.pdf`.
26. Lawrence R. Klein, *Economic Fluctuations in the United States, 1921-1941*, Wiley, 1950.
27. Michael H. Kutner, Chris J. Nachtsheim, and John Neter, *Applied Linear Statistical Models*, 5th ed., Irwin/McGraw-Hill, 2004.
28. Charles Lawson and Richard Hanson, *Solving Least Squares Problems*, Prentice-Hall, 1974.
29. J. W. Longley, *An Appraisal of Computer Programs for the Electronic Computer from the Point of View of the User*, Journal of the American Statistical Association **62** (1967), 819–841.
30. John F. Monahan, *Numerical Methods of Statistics*, Cambridge University Press, 2001.
31. R. Olsen, *A Note on the Uniqueness of the Maximum Likelihood Estimator in the Tobit Model*, Econometrica **46** (1978), 1211–1215.
32. R Development Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2004, ISBN 3-900051-00-3.
33. B. Shea, *Algorithm AS 239: Chi-squared and Incomplete Gamma Integral*, Applied Statistics **37** (1988), no. 3, 466–473.
34. Leopold Simar and Paul W. Wilson, *Sensitivity Analysis of Efficiency Scores: How to Bootstrap in Nonparametric Frontier Models*, Journal of Applied Statistics **44** (1998), no. 1, 49–61.
35. ———, *A general methodology for bootstrapping in nonparametric frontier models*, Journal of Applied Statistics **27** (2000), 779–802.
36. Henri Theil, *Principles of Econometrics*, Wiley, 1971.
37. J. Tobin, *Estimation of relationships for limited dependent variables*, Econometrica **26** (1958), 24–36.
38. W. N. Venables and D. M. Smith, *An Introduction to R*, Network Theory Ltd., 2002.
39. John Verzani, *Using R for Introductory Statistics*, CRC Press, 2004.
40. R. H. Wampler, *A Report of the Accuracy of Some Widely-Used Least Squares Computer Programs*, Journal of the American Statistical Association **65** (1970), 549–565.
41. Diana Whistler, Kenneth J. White, S. Donna Wong, and David Bates, *SHAZAM Version 9 User's Reference Manual*, Northwest Econometrics, 2001.
42. Mei Xue and Patrick T. Harker, *Overcoming the Inherent Dependency of DEA Efficiency Scores: A Bootstrap Approach*, Tech. report, Department of Operations and Information Management, The Wharton School, University of Pennsylvania, April 1999.

Amsterdam Optimization Modeling Group, Washington DC USA
E-mail address: `erwin@amsterdamoptimization.com`