LEAST SQUARES CALCULATIONS WITH GAMS

ERWIN KALVELAGEN

Abstract. This document show how different type of regression models can be solved with GAMS.

1. Linear Least Squares

1.1. **OLS is an optimization problem.** Ordinary Least Squares (OLS) is a technique to estimate parameters in a linear statistical model:

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

where y is the dependent (endogenous) variable (stored as an $(n \times 1)$ vector), and X is an $(n \times k)$ matrix of k independent (exogenous) variables. ϵ is an error term. We assume that $E(\epsilon'\epsilon) = \sigma^2 I_n$, i.e. the different ϵ_i's are independent. We can estimate β by the optimization model:

OLS

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & \sum_i \epsilon_i^2 \\ \text{subject to} \quad & y_i = \sum_j x_{i,j}\beta_j + \epsilon_i \end{aligned}$$

This model is trivially coded in GAMS using a simple linearly constrained NLP. Consider the following data from [25]: 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. Note that when a constant term is part of the model, a simple approach is to have a column of ones in the X matrix (usually this is the first column).

Notice that the notation is sometimes confusing: in many optimization models, x denotes the primary decision variable. In regression, X is a data matrix (i.e. a parameter in GAMS).

The econometrics package CHAZAM [51] gives the following results using the OLS procedure on this data set:

```
|_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
```

Date: December 13, 2007.

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

```
 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
```

Using the simple nonlinear GAMS formulation as displayed in the following fragment:

```
variables
   constant    'estimate constant term coefficient'
   income      'estimate income coefficient'
   residual(i) 'error term'
   sse         'sum of squared errors'
;

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

obj..     sse =e= sum(i, sqr(residual(i)));
fit(i)..  data(i,'expenditure') =e= constant + income * data(i,'income') + residual(i);

model osl1 /obj,fit/;
solve osl1 minimizing sse using nlp;

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

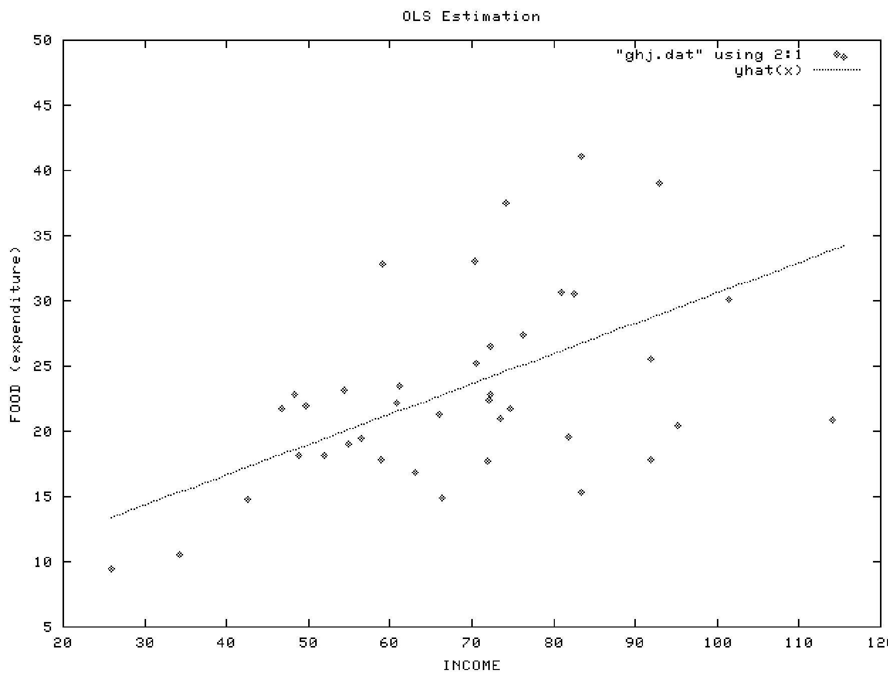

Figure 1. OLS Estimation

we get the following estimates:

```
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
```

The complete model is reproduced below:

1.1.1. *Model ols1.gms.* [1]

```
$ontext

   Ordinary Least Squares (OLS) by minimizing the Sum of
   Squared Errors directly.


   Erwin Kalvelagen, october 2000

$offtext


set i /i1*i40/;

$include expdata.inc

variables
    constant     'estimate constant term coefficient'
    income       'estimate income coefficient'
    residual(i)  'error term'
    sse          'sum of squared errors'
;

equations
```

[1] http://amsterdamoptimization.com/models/statistics/ols1.gms

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

;

obj..     sse =e= sum(i, sqr(residual(i)));
fit(i)..  data(i,'expenditure') =e= constant + income*data(i,'income') + residual(i);


model ols1 /obj,fit/;

solve ols1 minimizing sse using nlp;

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

The model uses an include for the data:

1.1.2. *Include file expdata.inc.* [2]

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

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
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
i40      48.71     115.46
;
```

[2]http://amsterdamoptimization.com/models/statistics/expdata.inc

1.2. **Solving the normal equations.** The standard way of formulating the OLS estimators is[3]

$$\hat{\beta} = (X'X)^{-1}X'y \tag{2}$$

where $\hat{\beta}$ denotes the estimate of β. This implies a simple linear formulation to find $\hat{\beta}$ using the so-called 'normal equations':

$$(X'X)\hat{\beta} = X'y \tag{3}$$

This is a system of linear equations. Such a system can be solved with GAMS as an LP using a *dummy objective*. A model that illustrates this is reproduced below. You can verify that it will give the same results.

1.2.1. *Model ols2.gms.* [4]

```
$ontext

   Ordinary Least Squares (OLS) by solving
   the normal equations.

   Erwin Kalvelagen, october 2000

$offtext


set i /i1*i40/;

$include expdata.inc

set j 'parameters to be estimated' /
      constant  'constant term'
      coeff1    'income coefficient'
/;
alias (j,jj);


parameters
   X(i,j)   'the X matrix in standard OLS notation (dependent variables)'
   y(i)     'the y vector in standard OLS notation (independent variables)'
   XX(j,jj) "the matrix (X'X)"
   Xy(j)    "the vector (X'y)"
;

X(i,'constant') = 1;
X(i,'coeff1')   = data(i,'income');

y(i) = data(i,'expenditure');

XX(j,jj) = sum(i, X(i,j)*X(i,jj));

Xy(j) = sum(i, X(i,j)*y(i));

equations
    dummy_eq   'dummy objective equation'
    normal(j)  "normal equations (X'X)b = X'y"
;

variables
    b(j)       'parameters to estimate'
    dummy_var  'dummy objective variable'
;

dummy_eq..  dummy_var =e= b('constant');
normal(j).. sum(jj, XX(j,jj)*b(jj)) =e= Xy(j);

model ols2 /dummy_eq,normal/;
```

[3]In this chapter we use x' to denote transposition, i.e. $x' = x^T$.

[4]http://amsterdamoptimization.com/models/statistics/ols2.gms

```
solve ols2 using lp minimizing dummy_var;

display b.l;
```

Notice that in some of the explanatory text a single quote is used inside the text. This can be done, but the explanatory text need then to be surrounded by double quotes.

It is noted that solving the normal equations is not numerically stable. A better way is to use QR or SVD decomposition. For an example of a QR based least square solver for GAMS see [31].

1.3. **OLS Statistics.** Many of the other statistics can be found relatively easily. We start with σ^2 or the variance of the estimate. This is calculated as:

$$\sigma^2 = \texttt{SSE}/(n-k) \tag{4}$$

where $(n-k)$ is the number of degrees of freedom (the number of observations minus the number of estimated parameters) and $\texttt{SSE} = \sum_i \epsilon_i^2$. The R^2 statistic (coefficient of determination) can easily be calculated using a matrix A ([47]) defined by

$$A = I - \frac{1}{n}(\iota\iota') \tag{5}$$

where ι is a vector of ones. Now:

$$R^2 = 1 - \frac{\texttt{SSE}}{y'Ay} \tag{6}$$

The adjusted R^2 coefficient (adjusted for the degrees of freedom), denoted by $\overline{R}^2$ can be written as:

$$\overline{R}^2 = 1 - \frac{n-1}{n-k}(1-R^2) \tag{7}$$

The logarithm of likelihood function can be written as:

$$\ln L(\beta,\sigma^2|y,X) = -\frac{n}{2}\ln 2\pi - \frac{n}{2}\ln\sigma^2 - \frac{(y-X\beta)'(y-X\beta)}{2\sigma^2} \tag{8}$$

which CHAZAM estimates by:

$$\text{llf} = -\frac{n}{2}\ln\left(2\pi\frac{\texttt{SSE}}{n}\right) - \frac{n}{2} \tag{9}$$

The standard errors of $\hat{\beta}$ can be calculated using

$$Var(\hat{\beta}) = \sigma^2(X'X)^{-1} \tag{10}$$

which can be solved using a system of linear equations. The standard errors are simply the square roots of the diagonal elements of this variance-covariance matrix. The t-statistic can be calculated simply by dividing β_k by its standard error.

The calculation of the above statistics are implemented in the following model:

1.3.1. *Model ols3.gms.* [5]

[5] `http://amsterdamoptimization.com/models/statistics/ols3.gms`

```
$ontext

   OLS plus statistics

   Erwin Kalvelagen, november 2000

$offtext

set i /i1*i40/;

$include expdata.inc

set j /'constant','coeff1'/;
alias (j,jj);
alias (i,ii);

parameters
   X(i,j)   'the X matrix in standard OLS notation (dependent variables)'
   y(i)     'the y vector in standard OLS notation (independent variables)'
   XX(j,jj) "the matrix (X'X)"
   Xy(j)    "the vector (X'y)"
;

X(i,'constant') = 1;
X(i,'coeff1')   = data(i,'income');

y(i) = data(i,'expenditure');

XX(j,jj) = sum(i, X(i,j)*X(i,jj));

Xy(j) = sum(i, X(i,j)*y(i));

equations
    dummy_eq   'dummy objective equation'
    normal(j)  "normal equations (X'X)b = X'y"
;

variables
    b(j)       'parameters to estimate'
    dummy_var  'dummy objective variable'
;

dummy_eq..  dummy_var =e= 0;
normal(j).. sum(jj, XX(j,jj)*b(jj)) =e= Xy(j);

model ols2 /dummy_eq,normal/;
solve ols2 using lp minimizing dummy_var;

display "----------------estimates--------------------",
        b.l;

parameters
   residual(i)   'residuals (errors)'
   yhat(i)       'predicted y'
   A(i,i)        "Theil's A matrix: I - (1/n) (iota*iota')"
;


scalars
   sse           'sum of squared errors'
   sst           'total sum of squares'
   ssr           'regression sum of squares'
   n             'number of observations'
   df            'degrees of freedom'
   r2            'r-square'
   r2adj         'r-square adjusted'
   llf           'log of the likelihood function'
   sigma_squared 'variance of the estimate'
   sigma         'standard error of the estimate'
   pi            '3.1415...'
;
```

```
yhat(i) = sum(j, x(i,j)*b.l(j));
residual(i) = y(i) - yhat(i);

sse = sum(i, sqr(residual(i)));

n = card(i);
df = n - 2;
sigma_squared = sse/df;
sigma = sqrt(sigma_squared);
pi = 4*arctan(1);

A(i,ii) = 1$sameas(i,ii) - (1/n);

sst = sum((i,ii), data(i,'expenditure')*A(i,ii)*data(ii,'expenditure'));
ssr = sst - sse;

r2 = 1-sse/sst;

r2adj = 1 - ((n-1)/df)*(1-r2);

llf = -(n/2)*log(2*pi*sse/n)-(n/2);



display "--------------statistics---------------------",
        sse,
        sigma_squared,
        sigma,
        r2,
        r2adj,
        llf;


alias (j,jjj);
variable var(j,jj) 'variance of the estimators';
equation variance(j,jj);
variance(j,jj).. sum(jjj, XX(j,jjj)*var(jjj,jj)) =e= sigma_squared$sameas(j,jj);

model mvar  /dummy_eq,variance/;
solve mvar using lp minimizing dummy_var;

parameters
    se(j)      'standard error'
    t(j)       "t ratio's"
    partial(j)
;

se(j) = sqrt(var.l(j,j));

t(j) =  b.l(j)/se(j);

partial(j) = t(j)/sqrt(sqr(t(j))+df);

display "---------------------------------------------------",
        se,
        t,
        partial
;
```

1.4. **Confidence Intervals.** The computation of confidence intervals requires the availability of critical values of the Student t distribution. A GAMS include file with a table of those values can be found in [34].

Alternatively for models with a sufficiently large number of degrees of freedom $df = n - k$ we can use a normal approximation:

```
set prob /p1,p2,p3,p4,p5,p6/;
parameter probval(prob) /
  p1 0.10,  p2 0.05,  p3 0.025,  p4 0.01,  p5 0.005,  p6 0.001
```

```
/;

parameter qnorm(prob) /
  p1 1.281552, p2 1.644854, p3 1.959964, p4 2.326348, p5 2.575829, p6 3.090232
/;
```

Then we can form confidence intervals by:

```
set ival 'confidence interval' /lo,up/;
scalar ndf 'degrees of freedom';
ndf = card(i) - card(k);
scalar alpha 'significance level' /0.025/;
scalar qt 'critical value';

abort$(ndf<30) "Normal approximation not valid";

qt = sum(prob$(probval(prob)=0.025), qnorm(prob));

parameter ols_conf_ival(k,ival);
ols_conf_ival(k,'lo') = beta.l(k) - qt*ols_se(k);
ols_conf_ival(k,'up') = beta.l(k) + qt*ols_se(k);
display ols_conf_ival;
```

where `ols_se(k)` is the standard error of coefficient k.

Note that the solver LS[31] will write the confidence intervals to a GDX file `ls.gdx`. They can be retrieved as follows:

```
*-----------------------------------------------------------------------------
* 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;
```

This will display:

```
----     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
```

2. LAD: Least Absolute Deviation

OLS (Ordinary Least Squares) is based on minimizing a sum of squared errors. A more *robust* estimator can be developed by minimizing the sum of absolute values of errors. This approach gives less weight to outliers. Least squares not only is more sensitive to outliers, but also assumes a Gaussian distribution for the errors. If the errors have a distribution that is non-Gaussian, e.g. with a fatter tail, estimators based on a LAD norm are sometimes considered more appropriate.

2.1. **LAD model formulations.** For our linear model $y = X\beta+\epsilon$ the optimization problem becomes

LAD		
	$\underset{\beta}{\text{minimize}}$	$\sum_i \|\epsilon_i\|$
	subject to	$y_i = \sum_j x_{i,j}\beta_j + \epsilon_i$

The actual implementation is a simple linear programming model using the variable splitting technique: replace ϵ_i by $\epsilon_i^+ - \epsilon_i^-$ and $|\epsilon_i|$ by $\epsilon_i^+ + \epsilon_i^-$ where $\epsilon_i^+, \epsilon_i^- \ge 0$ are non-negative variables. We don't need to add the nonlinear constraint $\epsilon_i^- \epsilon_i^+ = 0$ to enforce that one of he factors ϵ_i^+, ϵ_i^- is zero, as the minimization of $\epsilon_i^+ + \epsilon_i^-$ will automatically force this. The resulting LP model is:

$$\begin{aligned} &\min \sum_i \epsilon_i^+ + \epsilon_i^- \\ &y_i = \sum_j x_{i,j}\beta_j + \epsilon_i^+ - \epsilon_i^- \\ &\epsilon_i^+, \epsilon_i^- \ge 0 \end{aligned} \tag{11}$$

As slightly different formulation is:

$$\begin{aligned} &\min \sum_i e_i \\ &y_i = \sum_j x_{i,j}\beta_j + \epsilon_i \\ &e_i \ge \epsilon_i \\ &e_i \ge -\epsilon_i \end{aligned} \tag{12}$$

This implies that $e_i = |\epsilon_i|$.

If the number of constraints is much larger than the number of variables (i.e. the number of observations is much larger than the number of parameters to be estimated), we may think of a few other formulations. First it is noted that we can eliminate the variable ϵ:

$$\begin{aligned} &\min \sum_i e_i \\ &e_i \ge y_i - \sum_j x_{i,j}\beta_j \\ &e_i \ge -y_i + \sum_j x_{i,j}\beta_j \end{aligned} \tag{13}$$

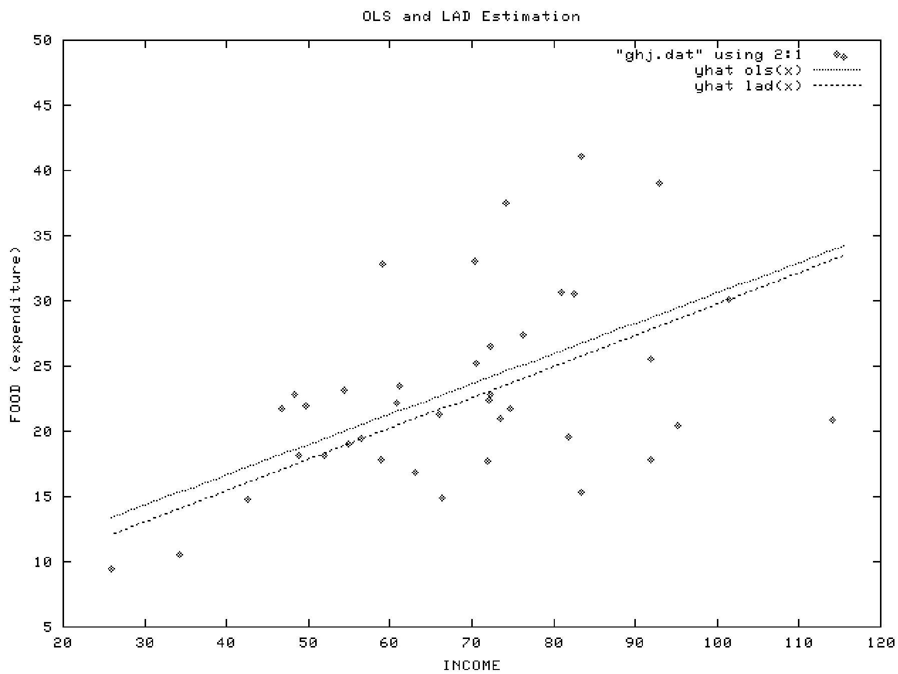

FIGURE 2. OLS and LAD Estimation

Second, we can formulate the dual problem [13]:

$$\begin{aligned} &\max \sum_i y_i v_i - \sum_i y_i w_i \\ &v_i + w_i = 1 \\ &\sum_i x_{i,j} v_i - \sum_i x_{i,j} w_i = 0 \\ &v_i, w_i \geq 0 \end{aligned} \tag{14}$$

We can now substitute out $w_i = 1 - v_i$, resulting in:

$$\begin{aligned} &\max -\sum_i y_i + 2\sum_i y_i v_i \\ &\quad -\sum_i x_{i,j} + 2\sum_i x_{i,j} v_i = 0 \\ &0 \leq v_i \leq 1 \end{aligned} \tag{15}$$

which can be simplified to:

$$\begin{aligned} &\max \sum_i y_i v_i \\ &\sum_i x_{i,j} v_i = \frac{1}{2} \sum_i x_{i,j} \\ &0 \leq v_i \leq 1 \end{aligned} \tag{16}$$

LAD regression is also known as MAD (Minimum Absolute Deviations) regression, LAV (Least Absolute Value) regression and ℓ_1 norm estimation. Interestingly, LAD has older traces back into history than least squares fitting. [9, 8] mention that the earliest references to a curve fitting criterion based on least absolute deviations are by Boscovitch[6], formulated somewhere around 1757, while the famous Legendre did publish his "Principle of Least Squares" as recent as 1805.

For some the of statistical properties of LAD estimators see the review [15].

2.1.1. *Model lad.gms.* [7]

This model finds LAD estimates directly by minimizing $\sum_i |\epsilon_i|$.

```
$ontext

   Least Absolute Deviation (LAD).

   Erwin Kalvelagen, october 2000

$offtext


set i /i1*i40/;

$include expdata.inc

variables
    constant     'estimate constant term coefficient'
    income       'estimate income coefficient'
    sad          'sum of absolute deviations'
;

positive variables
    res_plus(i)  'error term (plus term)'
    res_min(i)   'error term (minus term)'
;


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

;

obj..     sad =e= sum(i, res_plus(i) + res_min(i));
fit(i)..  data(i,'expenditure') =e= constant + income*data(i,'income')
                + res_plus(i) - res_min(i);


model ols1 /obj,fit/;

solve ols1 minimizing sad using lp;

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

2.1.2. *Model lad2.gms.* [8]

This is an alternative formulation without variable splitting.

```
$ontext

   Least Absolute Deviation (LAD), alternative formulation.
```

[6]Roger Joseph Boscovitch, also spelled as Rudjer J. Bǒsković (1711–1787), a Jesuit priest and prominent scientist, who spent most of his life in Rome, was born in Ragusa (now called Dubrovnic). For a fascinating account on his work on estimating the length of a meridian arc near Rome, see [46].

[7]http://amsterdamoptimization.com/models/statistics/lad.gms

[8]http://amsterdamoptimization.com/models/statistics/lad2.gms

```
   Erwin Kalvelagen, october 2000

$offtext


set i /i1*i40/;

$include expdata.inc

variables
    constant     'estimate constant term coefficient'
    income       'estimate income coefficient'
    sad          'sum of absolute deviations'
    e(i)         'error term'
    abse(i)      'abolute error term'
;

positive variables
    res_plus(i)  'error term (plus term)'
    res_min(i)   'error term (minus term)'
;


equations
    fit(i)     'the linear model'
    plus(i)    'plus inequalities'
    min(i)     'min inequalities'
    obj        'objective'

;

obj..     sad =e= sum(i, abse(i));
fit(i)..  data(i,'expenditure') =e= constant + income*data(i,'income') + e(i);
plus(i).. abse(i) =g= e(i);
min(i)..  abse(i) =g= -e(i);


model lad2 /obj,fit,plus,min/;

solve lad2 minimizing sad using lp;

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

2.2. **Best subset LAD regression.** An interesting extension of the LAD regression model is to find the best the best subset of variables to include in the regression [1]. Suppose we want k out of a possible m independent variables in the regression equation. The reasons for restricting the number of variables can include making the equation easier to understand, or reducing (future) cost in data collection, analysis and interpretation.

Of course we can run all possible combinations, but there are $\frac{m!}{k!(m-k)!}$ possible ways of choosing k out of m. A first model for this problem could read as:

$$
\begin{aligned}
&\min \sum_i |\epsilon_i| \\
&y_i = \sum_j x_{i,j}\beta_j\delta_j + \epsilon_i \\
&\sum_j \delta_j = k \\
&\delta_j \in \{0,1\}
\end{aligned}
\tag{17}
$$

where δ_j are binary variables. This non-linear formulation can be transformed into a linear one:

$$
\begin{aligned}
&\min \sum_i |\epsilon_i| \\
&y_i = \sum_j x_{i,j}\beta_j + \epsilon_i \\
&\sum_j \delta_j = k \\
&\beta_j \le M\delta_j \\
&\beta_j \ge -M\delta_j \\
&\delta_j \in \{0,1\}
\end{aligned} \tag{18}
$$

where M is a constant that need to be chosen with some care. It should be large enough so that $\beta_j \le M\delta_j$ and $\beta_j \ge -M\delta_j$ are non-binding for $\delta_j = 1$. I.e. M is a bound on β_j. On the other hand too large values lead to (numerical) problems in the MIP solver.

2.2.1. *Model subsetlad.gms.* [9]

```
$ontext

   Best subset LAD regression

   Erwin Kalvelagen, august 2001

$offtext



sets
   i /i1*i250/
   j /j1*j10/
;

parameter x(i,j) 'data independent variables';
parameter y(i)   'data dependent variable';

x(i,j) = normal(0,10);
y(i) = sum(j, x(i,j)*ord(j)) + normal(0,1);


variable b(j)              'parameters to estimate';
binary variable delta(j)   'subset selection';
variable z                 'sum of absolute deviations';
positive variable ep(i)    'positive deviations';
positive variable em(i)    'negative deviations';

equations
  obj       'objective'
  fit(i)    'equation to be fitted'
  subset    'select k'
  bigm1(j)  'big M formulation'
  bigm2(j)  'big M formulation'
;

scalar M  'big-M' /100/;

scalar k  'number of variables to select' /3/;

obj..       z =e= sum(i, ep(i)+em(i));
fit(i)..    y(i) =e= sum(j, x(i,j)*b(j)) + ep(i) - em(i);
```

[9] http://amsterdamoptimization.com/models/statistics/subsetlad.gms

```
bigm1(j)..  b(j) =l= M*delta(j);
bigm2(j)..  b(j) =g= -M*delta(j);
subset..    sum(j, delta(j)) =e= k;


option optcr=0;
option iterlim=1000000;
model ksubset /all/;
solve ksubset using mip minimizing z;
```

Not all solver have an easy time on this model. BDMLP for instance needs 189 nodes to solve this model to optimality, while a complete enumeration would take 120 nodes. The reason is that there are 10 binary variables, which leads to a theoretical worst case of $2^{10} = 1024$ nodes.

In [1] a special purpose branch-and-bound code is developed for this problem.

2.3. **Trend breaks in LAD regression.** The estimation of structural breaks has become a popular technique in econometrics [5, 6]. Estimating a trend break in a LAD regression problem can be done by a grid search as suggested in [4]. However, we can also do this directly using a MIP model.

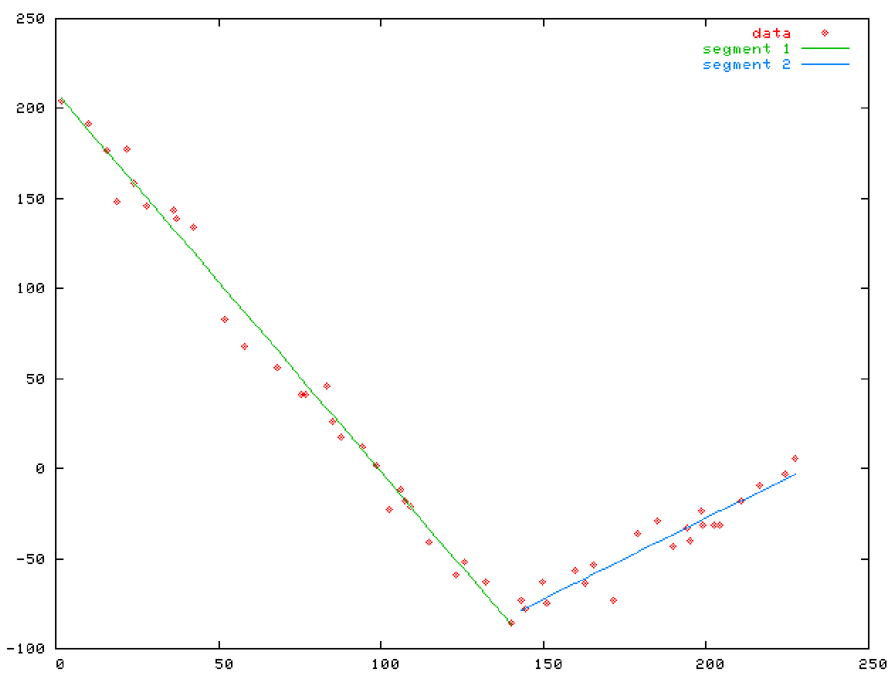

FIGURE 3. LAD Estimation of a trend break

The model can be developed as follows. First we have two regression equations:

$$y_i = a^{(1)} + b^{(1)} x_i + e_i^{(1)}$$
$$y_i = a^{(2)} + b^{(2)} x_i + e_i^{(2)} \tag{19}$$

We introduce binary variables to determine in which segment each x_i is:

$$\begin{aligned} x_i &\leq B + \delta_i M_1 \\ x_i &\geq B - (1-\delta_i) M_1 \end{aligned} \tag{20}$$

where B is the variable indicating the location of the break. The constant M_1 needs to be chosen as small as possible. The binary variables δ_i are also used to determine which error term contributes to the objective:

$$\eta_i = \begin{cases} e_i^{(1)} \text{if } \delta_i = 0 \\ e_i^{(2)} \text{if } \delta_i = 1 \end{cases} \tag{21}$$

After applying variable splitting:

$$\begin{aligned} &\eta_i = \eta_i^+ - \eta_i^- \\ &\eta_i^+, \eta_i^- \geq 0 \end{aligned} \tag{22}$$

we can form the objective:

$$\min \sum_i \eta_i^+ + \eta_i^- \tag{23}$$

Equation (21), combined with (22) can be written as a set of inequalities:

$$\begin{aligned} e_i^{(1)} - \delta_i M_2 \leq \eta_i^+ - \eta_i^- \leq e_i^{(1)} + \delta_i M_2 \\ e_i^{(2)} - (1-\delta_i) M_2 \leq \eta_i^+ - \eta_i^- \leq e_i^{(2)} + (1-\delta_i) M_2 \end{aligned} \tag{24}$$

Again M_2 should be chosen as small as possible. The actual choice of M_2 is not straightforward. It depends on the angle in the kink.

The following GAMS model demonstrates the procedure.

2.3.1. *Model l1break.gms.* [10]

```
$ontext

   L1 regression with a trend break

   Erwin Kalvelagen, oct 2003


$offtext

set i 'number of observations' /i1*i50/;

parameter x(i);
parameter y(i);

*
* generate test data
* x must be ordered: x(i) >= x(i-1).
*
set i1(i) /i1*i30/;
set i2(i) /i31*i50/;

x(i) = 0;
loop(i,
   x(i) = x(i-1) + uniform(0.1,10);
);
display x;

y(i1) = 200 - 2*x(i1) + normal(0,10);
y(i2) = -220 + x(i2) + normal(0,10);
```

[10] http://amsterdamoptimization.com/models/statistics/l1break.gms

```
display y;


*
* coefficients to estimate
*
variables a1,b1,a2,b2,break;

*
* error terms
*
variables e1(i),e2(i);

*
* equations to fit
*
equations fit1(i),fit2(i);
fit1(i)..  y(i) =e= a1 + b1*x(i) + e1(i);
fit2(i)..  y(i) =e= a2 + b2*x(i) + e2(i);


*
* we use binary variables b(i) to indicate
* if x(i) is in segment 1 or segment 2.
*
binary variable b(i);

equations seg1(i),seg2(i);
scalar m1; m1 = smax(i,x(i))-smin(i,x(i));

seg1(i).. x(i) =l= break + b(i)*M1;
seg2(i).. x(i) =g= break - (1-b(i))*M1;


variable u(i) 'either e1(i) or e2(i)';
equation udef1(i),udef2(i),udef3(i),udef4(i);

*
* 10*range of y is generous for m2
*
scalar m2;
m2 = 10*[smax(i,y(i))-smin(i,y(i))];

udef1(i).. u(i) =g= e1(i) - b(i)*m2;
udef2(i).. u(i) =l= e1(i) + b(i)*m2;
udef3(i).. u(i) =g= e2(i) - (1-b(i))*m2;
udef4(i).. u(i) =l= e2(i) + (1-b(i))*m2;

*
* variable splitting
*
positive variable v(i),w(i);
equation split(i);
split(i)..  v(i) - w(i) =e= u(i);

*
* objective
*
variable l1;
equation obj;

obj.. l1 =e= sum(i, v(i) + w(i));

model m/all/;
option optcr=0;
option mip=cplex;
solve m minimizing l1 using mip;

parameter res(i,*);
res(i,'x') = x(i);
res(i,'b') = b.l(i);
res(i,'u') = u.l(i);
```

```
res(i,'e1') = e1.l(i);
res(i,'e2') = e2.l(i);

display break.l,res;



*
* produce some graphs
*
file dat0 /l0.dat/;
file dat1 /l1.dat/;
file dat2 /l2.dat/;
loop(i,
   put dat0,x(i):12:7, y(i):12:7/
   if (b.l(i)<0.5,
      put dat1,x(i):12:7, (y(i)-e1.l(i)):12:7/
   else
      put dat2,x(i):12:7, (y(i)-e2.l(i)):12:7/
   );
);
putclose dat0;
putclose dat1;
putclose dat2;

file plt /l1.plt/;
putclose plt,
    'plot "l0.dat" title "data",'
    ' "l1.dat" title "segment 1" with lines,'
    ' "l2.dat" title "segment 2" with lines'/
    ' pause -1'/
execute 'gnuplot l1.plt';
```

3. Nonlinear Least Squares

3.1. Nonlinear regression using GAMS. Minimizing the sum of squared errors for nonlinear relationships is a task that an NLP solver is well equipped to do. The basic regression model is:

$$y = f_\theta(X) + \epsilon \tag{25}$$

where y and X form the observations for the dependent and independent variables, and θ is the vector of parameters to be estimated. In the optimization model this becomes:

$$\min_\theta ||y - f_\theta(X)||_2^2 \tag{26}$$

with θ becoming a decision variable and y, X are data.

3.2. Fitting a CES production function. As an example consider a CES (Constant Elasticity of Substitution) production function, an important equation used in many economic models. A search in the GAMS model library shows a handful of models that have CES functions. A production function $Q = f(K, L)$ measures output given inputs consisting of the 'factors of production' (in our case we have 2 factors: labor L and capital K). A simple production function often used in the economic literature is the Cobb-Douglas production function [50]. It looks like:

$$Q = \lambda K^\alpha L^\beta \tag{27}$$

A more complicated function that is very well known is the CES production function. CES functions were introduced by [3], and are also called ACMS functions, after the authors. For more information on CES functions and their limitations in

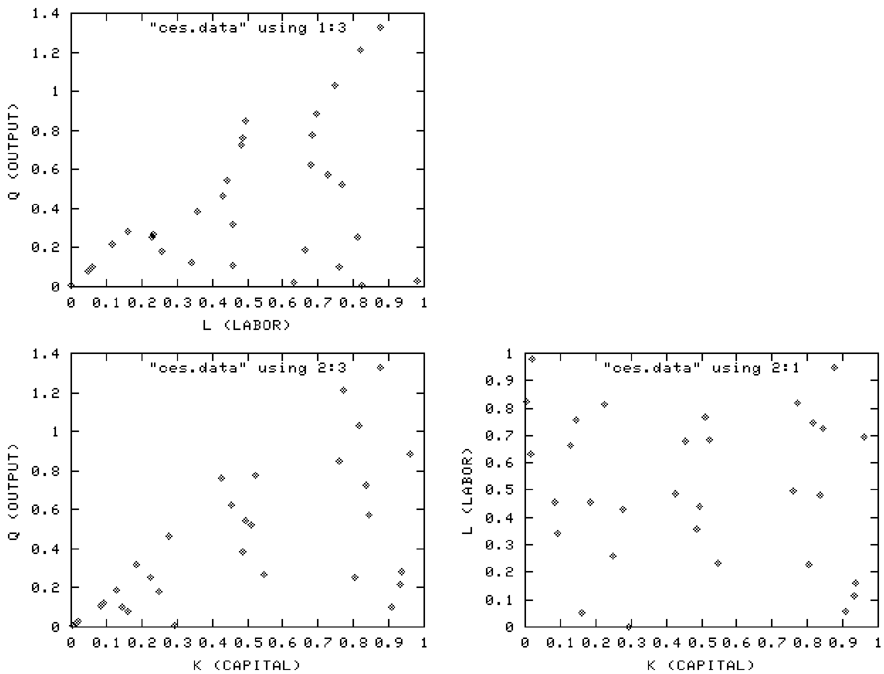

FIGURE 4. Scatter plots of the CES data set

production theory see [30, 17]. The functional form of a CES production function is:

$$Q = \gamma \left[\delta L^{-\rho} + (1-\delta)K^{-\rho}\right]^{-\frac{\eta}{\rho}} \tag{28}$$

where L is labor, K is capital, Q is output. γ is called the 'efficiency parameter' ($\gamma > 0$), δ is the 'distribution parameter' ($0 < \delta < 1$), and ρ is the 'substitution parameter' ($-1 \leq \rho \leq \infty$). η denotes the degree of homogeneity of the function.

Takings logs, and renaming some parameters, we can write this as:

$$\ln Q = \gamma - \frac{\eta}{\rho} \ln\left(\delta L^{-\rho} + (1-\delta)K^{-\rho}\right) \tag{29}$$

As an aside it is noted that CES functions also have an application in Linear Programming theory. Interior point methods are often based on variants of a logarithmic barrier function. However, one can also devise an interior point algorithm for linear programming based on a CES function [36].

A data set from [25] is used as an example for our nonlinear regression problem. It is reproduced in table 2

The optimization model to be solved can be simply stated as:

$$\text{NLREG} \qquad \begin{array}{ll} \underset{\gamma,\delta,\rho,\eta}{\text{minimize}} & \displaystyle\sum_i r_i^2 \\ \text{subject to} & \ln Q_i = \gamma - \frac{\eta}{\rho} \ln\left(\delta L_i^{-\rho} + (1-\delta)K_i^{-\rho}\right) + r_i \end{array}$$

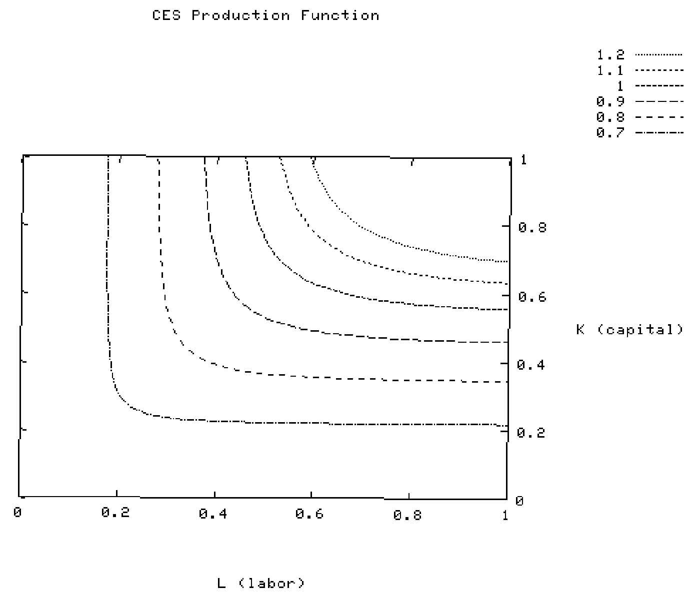

FIGURE 5. Contours of CES production function

L	K	Q	L	K	Q
0.228	0.802	0.256918	0.664	0.129	0.186747
0.258	0.249	0.183599	0.631	0.017	0.020671
0.821	0.771	1.212883	0.059	0.906	0.100159
0.767	0.511	0.522568	0.811	0.223	0.252334
0.495	0.758	0.847894	0.758	0.145	0.103312
0.487	0.425	0.763379	0.050	0.161	0.078945
0.678	0.452	0.623130	0.823	0.006	0.005799
0.748	0.817	1.031485	0.483	0.836	0.723250
0.727	0.845	0.569498	0.682	0.521	0.776468
0.695	0.958	0.882497	0.116	0.930	0.216536
0.458	0.084	0.108827	0.440	0.495	0.541182
0.981	0.021	0.026437	0.456	0.185	0.316320
0.002	0.295	0.003750	0.342	0.092	0.123811
0.429	0.277	0.461626	0.358	0.485	0.386354
0.231	0.546	0.268474	0.162	0.934	0.279431

TABLE 2. CES production function data set

This model finds least squares estimates directly by minimizing the sum of the squared errors. The results for this model are:

```
VARIABLE  gamma.L              =        0.124  log of efficiency parameter
VARIABLE  delta.L              =        0.337  distribution parameter
VARIABLE  rho.L                =        3.011  substitution parameter
```

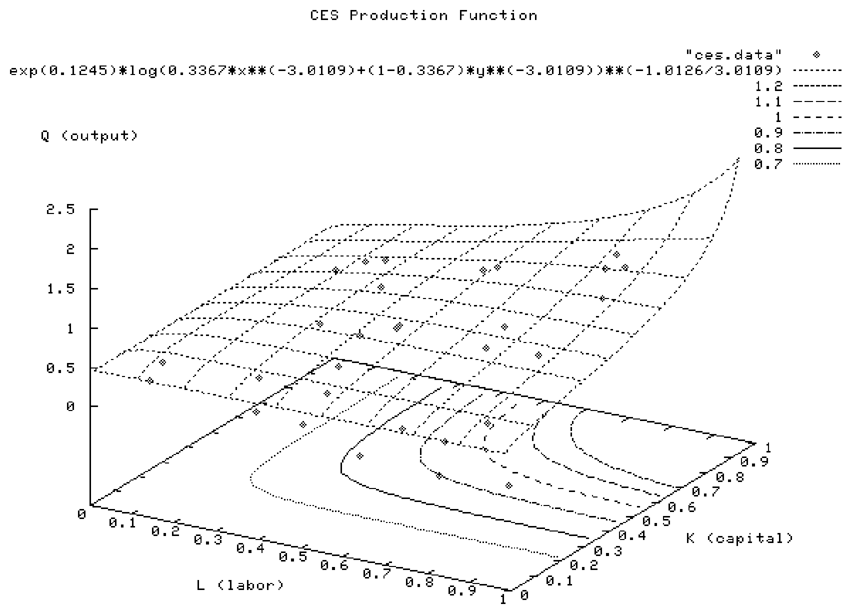

FIGURE 6. Surface of CES production function

```
VARIABLE  eta.L                 =        1.013  homogeneity parameter
VARIABLE  sse.L                 =        1.761  sum of squared errors
```

These figures are confirmed by the results of the CHAZAM [51] run reproduced in the next section.

3.2.1. *Model nls.gms.* [11]

```
$ontext

  Nonlinear least squares.

  Example: Estimation of a CES production function

  Data set: Table 22.4, page 724 of Griffiths, Hill and Judge,
     LEARNING AND PRACTICING ECONOMETRICS, Wiley, 1993.

  Erwin Kalvelagen, 2000

$offtext

set i 'observations' /i1*i30/;
set j 'parameters'   /L,K,Q/;

table data(i,j)
          L          K           Q
i1      0.228      0.802      0.256918
i2      0.258      0.249      0.183599
i3      0.821      0.771      1.212883
i4      0.767      0.511      0.522568
i5      0.495      0.758      0.847894
i6      0.487      0.425      0.763379
```

[11] http://amsterdamoptimization.com/models/statistics/nls.gms

```
i7      0.678      0.452      0.623130
i8      0.748      0.817      1.031485
i9      0.727      0.845      0.569498
i10     0.695      0.958      0.882497
i11     0.458      0.084      0.108827
i12     0.981      0.021      0.026437
i13     0.002      0.295      0.003750
i14     0.429      0.277      0.461626
i15     0.231      0.546      0.268474
i16     0.664      0.129      0.186747
i17     0.631      0.017      0.020671
i18     0.059      0.906      0.100159
i19     0.811      0.223      0.252334
i20     0.758      0.145      0.103312
i21     0.050      0.161      0.078945
i22     0.823      0.006      0.005799
i23     0.483      0.836      0.723250
i24     0.682      0.521      0.776468
i25     0.116      0.930      0.216536
i26     0.440      0.495      0.541182
i27     0.456      0.185      0.316320
i28     0.342      0.092      0.123811
i29     0.358      0.485      0.386354
i30     0.162      0.934      0.279431
;


parameters
    L(i)         'labor'
    K(i)         'capital'
    Q(i)         'output'
;

L(i) = data(i,'L');
K(i) = data(i,'K');
Q(i) = data(i,'Q');


variables
    gamma        'log of efficiency parameter'
    delta        'distribution parameter'
    rho          'substitution parameter'
    eta          'homogeneity parameter'
    residual(i)  'error term'
    sse          'sum of squared errors'
;

equations
    fit(i)     'the nonlinear model'
    obj        'objective'
;


obj..     sse =e= sum(i, sqr(residual(i)));
fit(i)..  log(Q(i)) =e=
             gamma - (eta/rho)*log[delta*L(i)**(-rho) + (1-delta)*K(i)**(-rho)]
             + residual(i);

* initial values
rho.l=1;
delta.l=0.5;
gamma.l=1;
eta.l=1;

model nls /obj,fit/;

solve nls minimizing sse using nlp;

display gamma.l, delta.l, rho.l, eta.l, sse.l;
```

3.2.2. *Output of CHAZAM.* This run is used to verify the solution.

```
*********************************************************************
Hello/Bonjour/Aloha/Howdy/G Day/Kia Ora/Konnichiwa/Buenos Dias/Nee Hau/Ciao
Welcome to SHAZAM - Version 9.0  -  OCT 2000 SYSTEM=LINUX    PAR=   781
|_ * NONLINEAR LEAST SQUARES AND TESTING FOR AUTOCORRELATED ERRORS
|_ *
|_ * Example: Estimation of a CES production function
|_ *
|_ * Data set: Table 22.4, page 724 of Griffiths, Hill and Judge,
|_ *     LEARNING AND PRACTICING ECONOMETRICS, Wiley, 1993.
|_ *
|_ SAMPLE 1 30
|_ READ L K Q
   3 VARIABLES AND       30 OBSERVATIONS STARTING AT OBS       1

|_ GENR LOGQ=LOG(Q)
|_ * Estimate the CES production function

|_ NL 1 / NCOEF=4 PCOV ZMATRIX=Z COEF=BETA PREDICT=YHAT
...NOTE..SAMPLE RANGE SET TO:     1,    30
|_ EQ LOGQ=GAMMA-(ETA/RHO)*LOG(DELTA*L**(-RHO)+(1-DELTA)*K**(-RHO))
|_ COEF  RHO 1  DELTA .5  GAMMA 1  ETA 1
   3 VARIABLES IN  1 EQUATIONS WITH   4 COEFFICIENTS
        30 OBSERVATIONS

REQUIRED MEMORY IS PAR=      22 CURRENT PAR=     781

COEFFICIENT STARTING VALUES
GAMMA      1.0000      ETA        1.0000      RHO        1.0000
DELTA     0.50000
      100 MAXIMUM ITERATIONS, CONVERGENCE =   0.100000E-04

INITIAL STATISTICS :

TIME =      0.000 SEC.   ITER. NO.     0   FUNCT. EVALUATIONS     1
LOG-LIKELIHOOD FUNCTION=   -45.75315
COEFFICIENTS
   1.000000       1.000000       1.000000      0.5000000
GRADIENT
  -25.82924       42.55772       5.501555      -6.222794

INTERMEDIATE STATISTICS :

TIME =      0.040 SEC.   ITER. NO.    15   FUNCT. EVALUATIONS    25
LOG-LIKELIHOOD FUNCTION=  -0.6460435E-01
COEFFICIENTS
  0.1246125       1.018006       2.750354      0.3581035
GRADIENT
 -0.5488362      0.2883618      0.2721726E-01  -2.287144

FINAL STATISTICS :

TIME =      0.060 SEC.   ITER. NO.    25   FUNCT. EVALUATIONS    35
LOG-LIKELIHOOD FUNCTION=  -0.3907423E-01
COEFFICIENTS
  0.1244913       1.012594       3.010941      0.3366735
GRADIENT
 -0.4790552E-03 -0.7407311E-04  0.1256401E-05 -0.1297012E-03
ASYMPTOTIC COVARIANCE MATRIX
GAMMA     0.47917E-02
ETA       0.61486E-03  0.21055E-02
RHO       0.50037E-01 -0.67552E-01   4.0663
DELTA    -0.15592E-02  0.20209E-02 -0.13216      0.97548E-02
           GAMMA        ETA          RHO          DELTA

MAXIMUM LIKELIHOOD ESTIMATE OF SIGMA-SQUARED =  0.58703E-01
SUM OF SQUARED ERRORS =    1.7611
GTRANSPOSE*INVERSE(H)*G  STATISTIC  -  =  0.11658E-08

      COEFFICIENT    ST. ERROR   T-RATIO

GAMMA    0.12449     0.69222E-01  1.7984
```

```
 ETA       1.0126     0.45886E-01  22.068
 RHO       3.0109      2.0165      1.4931
 DELTA    0.33667     0.98766E-01  3.4088
 |_ END
 |_ STOP
```

An alternative formulation for the nonlinear least squares problem is:

$$\begin{aligned} &\min \sum_i \epsilon_i^2 \\ &-\epsilon_i \leq f_\theta(X_i) - y_i \leq \epsilon_i \end{aligned} \tag{30}$$

It is noted that we can easily add inequalities and bounds that are known from theory, and bounds that keep values in a reasonable range:

```
  delta.lo = 0.0001;
  delta.up = 0.9999;
  rho.lo = -0.9999;
* rho.lo = 0.0001; would be even better
  rho.up = 100;
```

This is likely more powerful than found in the non-linear regression found in many statistical packages.

3.3. **Nonlinear regression using NLS.** A specialized Non-linear Regression solver called NLS[32] is available. It has a built-in nonlinear least squares solver NL2SOL[29]. It provides numerous statistical output including the standard errors and the variance-covariance matrix. In addition it can take a starting point as calculated by a GAMS NLP solver such as MINOS or CONOPT.

The above model can be specified as:

```
variables
    gamma        'log of efficiency parameter'
    delta        'distribution parameter'
    rho          'substitution parameter'
    eta          'homogeneity parameter'
    residual(i)  'error term'
    sse          'sum of squared errors'
;

equations
    fit(i)     'the nonlinear model'
    obj        'objective'
;


obj..     sse =e= sum(i, sqr(residual(i)));
fit(i)..  log(Q(i)) =e=
             gamma - (eta/rho)*log[delta*L(i)**(-rho) + (1-delta)*K(i)**(-rho)]
             + residual(i);

* initial values
rho.l=1;
delta.l=0.5;
gamma.l=1;
eta.l=1;

model nls /obj,fit/;
solve nls minimizing sse using nlp;

display gamma.l, delta.l, rho.l, eta.l, sse.l;
```

The output looks like:

```
 =========================================================================
   Nonlinear Least Square Solver V1
```

```
  Erwin Kalvelagen, Amsterdam Optimization Modeling Group
  www.amsterdamoptimization.com
=======================================================================

Nonlinear Least Square Solver: NL2SOL

        Parameter      Estimate    Std. Error       t value      Pr(>|t|)
            gamma  -1.24491E-01   7.83443E-02  -1.58902E+00   1.24143E-01
            delta   3.36673E-01   1.36112E-01   2.47350E+00   2.02326E-02 *
              rho   3.01094E+00   2.32337E+00   1.29593E+00   2.06385E-01
              eta   1.01259E+00   5.06832E-02   1.99789E+01   2.66997E-17 ***
     ---
     Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error:   2.60257E-01   on  26 degrees of freedom


DLL version: _GAMS_GDX_237_2007-01-09
GDX file: nls.gdx
```

4. Maximum likelihood estimation

4.1. Maximum likelihood estimation of the Gamma distribution. Consider data collected on times between failures of air conditioning units in different aircraft[14]. We assume the times between failures are independent random variables with a Gamma distribution. Given a mean time between failures μ and a shape parameter β, the density function of the gamma distribution is[48]:

$$f(x) = \frac{(\beta/\mu)(\beta x/\mu)^{\beta-1}e^{-\beta x/\mu}}{\Gamma(\beta)} \tag{31}$$

The log likelihood function can now be written as:

$$L(\mu,\beta) = n\left[\ln\beta - \ln\mu - \ln\Gamma(\beta)\right] + \sum_{i=1}^{n}(\beta-1)\ln\left(\frac{\beta x_i}{\mu}\right) - \sum_{i=1}^{n}\frac{\beta x_i}{\mu} \tag{32}$$

We can maximize this function using the `loggamma(.)` function.

The method of moments estimator of β is

$$\hat{\beta} = \left(\frac{\hat{\mu}}{\hat{\sigma}}\right)^2 \tag{33}$$

which can be used as an (excellent) initial point for the optimization problem.

4.1.1. *Model mlgamma.gms.* [12]

```
$ontext

   Maximum Likelihood estimation of parameters of the gamma distribution


   Erwin Kalvelagen, april 2004.


   Data from:
      COX, D. R. AND SNELL, E. J., (1981)
      Applied Statistics: Principles and Examples,
      London: Chapman and Hall.
   Example from:
      Luke Tierney, July 1989
      XLISP-STAT, A Statistical Environment Based on the XLISP Language (Version 2.0)
      Technical Report Number 528, University of Minnesota, School of Statistics
```

[12] http://amsterdamoptimization.com/models/statistics/mlgamma.gms

```
$offtext


set i 'observations' /i1*i29/

parameter x(i) 'times (in operating hours) between failures of airco units on several aircraft'
 /
  i1  90,  i2  10,  i3  60,  i4 186,  i5  61
  i6  49,  i7  14,  i8  24,  i9  56, i10  20
 i11  79, i12  84, i13  44, i14  59, i15  29
 i16 118, i17  25, i18 156, i19 310, i20  76
 i21  26, i22  44, i23  23, i24  62, i25 130
 i26 208, i27  70, i28 101, i29 208
/;

scalar n;
n = card(i);

scalar average;
average = sum(i, x(i))/n;

scalar stdev 'standard deviation';
stdev = sqrt(sum(i, sqr(x(i)-average))/(n-1));

display average,stdev;


variables beta,mu,like;
equations loglike;

loglike.. like =e= n*[log(beta)-log(mu)-loggamma(beta)] +
                   sum(i, (beta-1)*log(beta*x(i)/mu)) -
                   sum(i, beta*x(i)/mu);


*
* lowerbounds so log() and loggamma() are safe
*
beta.lo = 0.0001;
mu.lo = 0.0001;

*
* initial values using moments estimates
*
mu.l = average;
beta.l = sqr(average/stdev);


model m /loglike/;
solve m using nlp maximimizing like;
```

The resulting estimates for the parameters μ and β are:

```
                            LOWER          LEVEL          UPPER         MARGINAL

---- VAR beta                0.0001         1.6710         +INF       1.004352E-11
---- VAR mu                  0.0001        83.5172         +INF       1.944001E-13
---- VAR like               -INF         -155.3468         +INF             .
```

4.2. **Maximum likelihood estimation of the Beta distribution.** The log likelihood function of the beta distribution with parameters α and β is:

(34)

$$\ln L = n\left[\ln\Gamma(\alpha+\beta) - \ln\Gamma(\alpha) - \ln\Gamma(\beta)\right] + \sum_{i=1}^{n}(\alpha-1)\ln(x_i) + \sum_{i=1}^{n}(\beta-1)\ln(1-x_i)$$

This function can be implemented straightforwardly using the `loggamma` function.

The first two moments of the beta distribution, lead to two equations in two variables defining the method of moments estimator of α and β:

$$\begin{aligned} E(X) &= \frac{\alpha}{\alpha+\beta} \\ Var(X) &= \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \end{aligned} \tag{35}$$

or

$$\begin{aligned} \hat{\alpha} &= \left[\frac{\hat{\mu}(1-\hat{\mu})}{\hat{\sigma}^2} - 1\right]\hat{\mu} \\ \hat{\beta} &= \left[\frac{\hat{\mu}(1-\hat{\mu})}{\hat{\sigma}^2} - 1\right](1-\hat{\mu}) \end{aligned} \tag{36}$$

4.2.1. *Model mlbeta.gms.* [13]

```
$ontext

   Fitting of beta distribution through maximum likelihood

   Erwin Kalvelagen, april 2004

   Reference:
       Johnson, Kotz, and Balakrishnan, (1994),
       Continuous Univariate Distributions, Volumes I and II,
       2nd. Ed., John Wiley and Sons.

$offtext


set i 'cases' /i1*i75/;
parameter x(i) /

i1  4.973016e-01, i2  3.558841e-01, i3  2.419578e-02, i4  1.913753e-01, i5  4.919495e-01
i6  9.790016e-01, i7  3.856570e-01, i8  1.568263e-01, i9  8.040481e-01, i10 8.108720e-01
i11 6.016693e-01, i12 3.691279e-02, i13 9.454942e-01, i14 1.853702e-01, i15 3.496894e-01
i16 4.249933e-01, i17 9.900851e-01, i18 6.308701e-01, i19 4.474022e-02, i20 4.408432e-03
i21 3.718974e-03, i22 1.066217e-01, i23 5.304127e-01, i24 6.781648e-01, i25 6.206926e-02
i26 4.048511e-01, i27 4.941163e-01, i28 1.644695e-01, i29 2.285463e-02, i30 5.654344e-05
i31 2.657641e-01, i32 7.316988e-01, i33 6.789551e-01, i34 3.624824e-01, i35 7.429815e-03
i36 1.503384e-01, i37 7.314336e-01, i38 4.586442e-02, i39 4.060616e-02, i40 3.395101e-01
i41 9.269645e-01, i42 2.192909e-03, i43 2.511850e-02, i44 4.152490e-01, i45 1.612197e-01
i46 1.512879e-02, i47 1.381864e-01, i48 5.730967e-03, i49 1.185086e-01, i50 7.411310e-01
i51 1.564168e-02, i52 2.206906e-01, i53 9.836009e-01, i54 4.632388e-01, i55 9.968135e-01
i56 8.792355e-04, i57 9.692757e-01, i58 9.823214e-01, i59 1.248862e-01, i60 1.598848e-01
i61 9.561613e-02, i62 2.513807e-01, i63 4.435097e-01, i64 8.852468e-01, i65 1.149253e-02
i66 6.575999e-01, i67 8.236305e-01, i68 7.388426e-01, i69 6.382491e-01, i70 3.426699e-01
i71 1.244351e-01, i72 2.753017e-05, i73 1.625740e-01, i74 2.953334e-02, i75 8.739085e-02

/;

scalar n;
n = card(i);

scalar average;
average = sum(i, x(i))/n;

scalar stdev 'standard deviation';
stdev = sqrt(sum(i, sqr(x(i)-average))/(n-1));

variables alpha,beta,like;
equations loglike;

loglike.. like =e= n*[loggamma(alpha+beta)-loggamma(alpha)-loggamma(beta)] +
                   sum(i, (alpha-1)*log(x(i))) +
                   sum(i, (beta-1)*log(1-x(i)));
```

[13] http://amsterdamoptimization.com/models/statistics/mlbeta.gms

```
*
* lowerbounds so log() is safe
*
alpha.lo = 0.0001;
beta.lo = 0.0001;

*
* initial values using matching moments estimates
*
scalar tmp;
tmp = average*(1-average)/sqr(stdev) - 1;
alpha.l = tmp*average;
beta.l = tmp*(1-average);

display alpha.l,beta.l;

model m /loglike/;
solve m using nlp maximimizing like;

display alpha.l,beta.l;
```

4.3. **The Tobit model.** The Tobit model[49] deals with a dependent variable y which is only observed if $y_i > 0$. To be more precise, in the regression equation

$$y' = X\beta + \varepsilon \tag{37}$$

we only observe

$$y = max\{y', 0\} \tag{38}$$

A well-known method to estimate β in this case is to maximize the log-likelihood function[24, 7]:

$$\begin{aligned}\ln L &= \sum_{y_i>0} -\frac{1}{2}\left[\ln(2\pi) + \ln\sigma^2 + \frac{(y_i - x_i'\beta)^2}{\sigma^2}\right] + \sum_{y_i=0} \ln\left[1 - \Phi\left(\frac{x_i'\beta}{\sigma}\right)\right] \\ &= \sum_{y_i>0} -\frac{1}{2}\left[\ln\sigma^2 + \frac{(y_i - x_i'\beta)^2}{\sigma^2}\right] + \sum_{y_i=0} \ln\left[1 - \Phi\left(\frac{x_i'\beta}{\sigma}\right)\right] - \sum_{y_i>0} \ln\sqrt{2\pi}\end{aligned} \tag{39}$$

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. These estimates have been calculated using the LS solver from [31].

4.3.1. *Model tobit.gms.* [14]

```
$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
```

[14] http://amsterdamoptimization.com/models/statistics/tobit.gms

```
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.
     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 [40] while other evidence suggests this is not needed [23].

4.4. **GARCH: Constrained maximum likelihood estimation for Generalized Autoregressive Conditional Heteroscedasticity models.** GARCH or Generalized Autoregressive Conditional Heteroskedasticity[10] models have become a popular tool in describing and forecasting financial time series with fluctuating volatility[11]. The GARCH(p, q) model can be stated as:

$$Y_t = F(\beta, X_t) + \epsilon_t \tag{40}$$

$$\epsilon_t = \sigma_t z_t \tag{41}$$

$$\sigma_t = \sqrt{h_t} \tag{42}$$

$$h_t = \alpha + \sum_{i=1}^{p} \delta_i h_{t-i} + \sum_{j=1}^{q} \gamma_j \epsilon_{t-j}^2 \tag{43}$$

where Y_t are the observations and $F(\beta, X_t) = \mu$ is a function of independent variables (which we assume to be simply the constant function in our example). The residuals ϵ_t form a stochastic process described by the latter three equations, where z_t follows in our case a Gaussian distribution.

We want to estimate $\theta = (\mu, \alpha, \delta_i, \gamma_j)$. The standard way of doing is to maximize the likelihood function

$$L_T(\theta) = \sum_t \ell_t(\theta) \tag{44}$$

with

$$\ell_t(\theta) = \ln f_z(\epsilon_t|\sigma_t) \tag{45}$$

As z_t is Gaussian, we have

$$f_z(\epsilon_t|\sigma_t) = \frac{1}{\sqrt{2\pi}} \frac{1}{\sigma} \exp\left\{ -\frac{1}{2} \left(\frac{\epsilon_t}{\sigma} \right)^2 \right\} \tag{46}$$

Stationarity of the process requires the following conditions on the parameters: $\alpha > 0$, $\delta_i \geq 0$, $\gamma_j \geq 0$ and $\sum_i \delta_i + \sum_j \gamma_j \leq 1$. Thus the following optimization model can be formulated:

$$\text{GARCHML} \qquad \begin{aligned} \underset{\mu,\alpha,\delta_i,\gamma_j}{\text{maximize}} \quad & \sum_t -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(h_t) - \frac{1}{2}\frac{\epsilon_t^2}{h_t} \\ \text{subject to} \quad & \sum_i \delta_i + \sum_j \gamma_j \leq 1 \\ & \alpha > 0, \delta_i \geq 0, \gamma_j \geq 0 \end{aligned}$$

A slight rewrite gives as objective[35]

$$\underset{\mu,\alpha,\delta_i,\gamma_j}{\text{minimize}} \sum_t \ln(h_t) + \frac{\epsilon_t^2}{h_t} \tag{47}$$

Changing the stationarity condition into

$$\sum_i \delta_i + \sum_j \gamma_j = 1 \tag{48}$$

results in the IGARCH model (Integrated GARCH).

For a description of the use of SQP (Sequential Quadratic Programming) to estimate these type of models see [44].

The model below is using data from [22]. We use a standard formulation for the returns:

$$Y_t = 100 \ln \frac{S_t}{S_{t-1}} \tag{49}$$

where S_t is the original time series. The model is fairly inefficient: it contains many more variables and constraints than is strictly needed if we could form e_t and h_t inside the objective function evaluation.

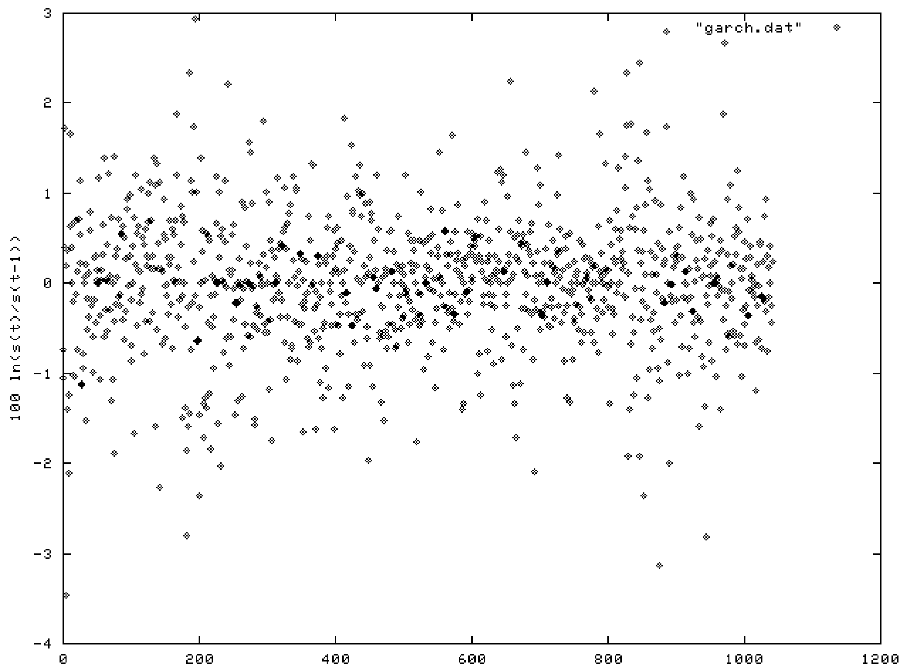

FIGURE 7. Time series data Y_t

4.4.1. *Model garch.gms.* [15]

```
$title GARCH -- Generalized Autoregressive Conditional Heteroskedasticity
$ontext

  Example of restricted maximum likelihood estimation.

  Erwin Kalvelagen, 2001

  References:
        Jon Kierkegaard, Estimation of Nonlinear Stochastic Processes,
        MSc Thesis, Technical University of Denmark, report
        IMM-EKS-2000-16, April 2000.

        Bollerslev, T. (1986), "Generalized Autoregressive Conditional
        Heteroskedasticity", Journal of Econometrics 31, 307-327.

        Bollerslev, T., Chou, R.Y. and Kroner, K.F. (1992), "ARCH
        Modeling in Finance: A Review of the Theory and Empirical
        Evidence," Journal of Econometrics 52, 5-59.

        Data set from: Stephen F. Gray, GARCH Code,
        http://www.duke.edu/~sg12/software/garch/garch.htm

$offtext

*------- order of the GARCH(p,q) model ----------
```

[15] http://amsterdamoptimization.com/models/statistics/garch.gms

```
$set p 1
$set q 1
*-----------------------------------------------


set t 'dates yymmdd'  /
   920101, 920102, 920103, 920106, 920107, 920108, 920109, 920110, 920113
   920114, 920115, 920116, 920117, 920120, 920121, 920122, 920123, 920124
   920127, 920128, 920129, 920130, 920131, 920203, 920204, 920205, 920206
   920207, 920210, 920211, 920212, 920213, 920214, 920217, 920218, 920219
   920220, 920221, 920224, 920225, 920226, 920227, 920228, 920302, 920303
   920304, 920305, 920306, 920309, 920310, 920311, 920312, 920313, 920316
   920317, 920318, 920319, 920320, 920323, 920324, 920325, 920326, 920327
   920330, 920331, 920401, 920402, 920403, 920406, 920407, 920408, 920409
   920410, 920413, 920414, 920415, 920416, 920417, 920420, 920421, 920422
   920423, 920424, 920427, 920428, 920429, 920430, 920501, 920504, 920505
   920506, 920507, 920508, 920511, 920512, 920513, 920514, 920515, 920518
   920519, 920520, 920521, 920522, 920525, 920526, 920527, 920528, 920529
   920601, 920602, 920603, 920604, 920605, 920608, 920609, 920610, 920611
   920612, 920615, 920616, 920617, 920618, 920619, 920622, 920623, 920624
   920625, 920626, 920629, 920630, 920701, 920702, 920703, 920706, 920707
   920708, 920709, 920710, 920713, 920714, 920715, 920716, 920717, 920720
   920721, 920722, 920723, 920724, 920727, 920728, 920729, 920730, 920731
   920803, 920804, 920805, 920806, 920807, 920810, 920811, 920812, 920813
   920814, 920817, 920818, 920819, 920820, 920821, 920824, 920825, 920826
   920827, 920828, 920831, 920901, 920902, 920903, 920904, 920907, 920908
   920909, 920910, 920911, 920914, 920915, 920916, 920917, 920918, 920921
   920922, 920923, 920924, 920925, 920928, 920929, 920930, 921001, 921002
   921005, 921006, 921007, 921008, 921009, 921012, 921013, 921014, 921015
   921016, 921019, 921020, 921021, 921022, 921023, 921026, 921027, 921028
   921029, 921030, 921102, 921103, 921104, 921105, 921106, 921109, 921110
   921111, 921112, 921113, 921116, 921117, 921118, 921119, 921120, 921123
   921124, 921125, 921126, 921127, 921130, 921201, 921202, 921203, 921204
   921207, 921208, 921209, 921210, 921211, 921214, 921215, 921216, 921217
   921218, 921221, 921222, 921223, 921224, 921225, 921228, 921229, 921230
   921231, 930101, 930104, 930105, 930106, 930107, 930108, 930111, 930112
   930113, 930114, 930115, 930118, 930119, 930120, 930121, 930122, 930125
   930126, 930127, 930128, 930129, 930201, 930202, 930203, 930204, 930205
   930208, 930209, 930210, 930211, 930212, 930215, 930216, 930217, 930218
   930219, 930222, 930223, 930224, 930225, 930226, 930301, 930302, 930303
   930304, 930305, 930308, 930309, 930310, 930311, 930312, 930315, 930316
   930317, 930318, 930319, 930322, 930323, 930324, 930325, 930326, 930329
   930330, 930331, 930401, 930402, 930405, 930406, 930407, 930408, 930409
   930412, 930413, 930414, 930415, 930416, 930419, 930420, 930421, 930422
   930423, 930426, 930427, 930428, 930429, 930430, 930503, 930504, 930505
   930506, 930507, 930510, 930511, 930512, 930513, 930514, 930517, 930518
   930519, 930520, 930521, 930524, 930525, 930526, 930527, 930528, 930531
   930601, 930602, 930603, 930604, 930607, 930608, 930609, 930610, 930611
   930614, 930615, 930616, 930617, 930618, 930621, 930622, 930623, 930624
   930625, 930628, 930629, 930630, 930701, 930702, 930705, 930706, 930707
   930708, 930709, 930712, 930713, 930714, 930715, 930716, 930719, 930720
   930721, 930722, 930723, 930726, 930727, 930728, 930729, 930730, 930802
   930803, 930804, 930805, 930806, 930809, 930810, 930811, 930812, 930813
   930816, 930817, 930818, 930819, 930820, 930823, 930824, 930825, 930826
   930827, 930830, 930831, 930901, 930902, 930903, 930906, 930907, 930908
   930909, 930910, 930913, 930914, 930915, 930916, 930917, 930920, 930921
   930922, 930923, 930924, 930927, 930928, 930929, 930930, 931001, 931004
   931005, 931006, 931007, 931008, 931011, 931012, 931013, 931014, 931015
   931018, 931019, 931020, 931021, 931022, 931025, 931026, 931027, 931028
   931029, 931101, 931102, 931103, 931104, 931105, 931108, 931109, 931110
   931111, 931112, 931115, 931116, 931117, 931118, 931119, 931122, 931123
   931124, 931125, 931126, 931129, 931130, 931201, 931202, 931203, 931206
   931207, 931208, 931209, 931210, 931213, 931214, 931215, 931216, 931217
   931220, 931221, 931222, 931223, 931224, 931227, 931228, 931229, 931230
   931231, 940103, 940104, 940105, 940106, 940107, 940110, 940111, 940112
   940113, 940114, 940117, 940118, 940119, 940120, 940121, 940124, 940125
   940126, 940127, 940128, 940131, 940201, 940202, 940203, 940204, 940207
   940208, 940209, 940210, 940211, 940214, 940215, 940216, 940217, 940218
   940221, 940222, 940223, 940224, 940225, 940228, 940301, 940302, 940303
   940304, 940307, 940308, 940309, 940310, 940311, 940314, 940315, 940316
   940317, 940318, 940321, 940322, 940323, 940324, 940325, 940328, 940329
   940330, 940331, 940401, 940404, 940405, 940406, 940407, 940408, 940411
```

```
   940412, 940413, 940414, 940415, 940418, 940419, 940420, 940421, 940422
   940425, 940426, 940427, 940428, 940429, 940502, 940503, 940504, 940505
   940506, 940509, 940510, 940511, 940512, 940513, 940516, 940517, 940518
   940519, 940520, 940523, 940524, 940525, 940526, 940527, 940530, 940531
   940601, 940602, 940603, 940606, 940607, 940608, 940609, 940610, 940613
   940614, 940615, 940616, 940617, 940620, 940621, 940622, 940623, 940624
   940627, 940628, 940629, 940630, 940701, 940704, 940705, 940706, 940707
   940708, 940711, 940712, 940713, 940714, 940715, 940718, 940719, 940720
   940721, 940722, 940725, 940726, 940727, 940728, 940729, 940801, 940802
   940803, 940804, 940805, 940808, 940809, 940810, 940811, 940812, 940815
   940816, 940817, 940818, 940819, 940822, 940823, 940824, 940825, 940826
   940829, 940830, 940831, 940901, 940902, 940905, 940906, 940907, 940908
   940909, 940912, 940913, 940914, 940915, 940916, 940919, 940920, 940921
   940922, 940923, 940926, 940927, 940928, 940929, 940930, 941003, 941004
   941005, 941006, 941007, 941010, 941011, 941012, 941013, 941014, 941017
   941018, 941019, 941020, 941021, 941024, 941025, 941026, 941027, 941028
   941031, 941101, 941102, 941103, 941104, 941107, 941108, 941109, 941110
   941111, 941114, 941115, 941116, 941117, 941118, 941121, 941122, 941123
   941124, 941125, 941128, 941129, 941130, 941201, 941202, 941205, 941206
   941207, 941208, 941209, 941212, 941213, 941214, 941215, 941216, 941219
   941220, 941221, 941222, 941223, 941226, 941227, 941228, 941229, 941230
   950102, 950103, 950104, 950105, 950106, 950109, 950110, 950111, 950112
   950113, 950116, 950117, 950118, 950119, 950120, 950123, 950124, 950125
   950126, 950127, 950130, 950131, 950201, 950202, 950203, 950206, 950207
   950208, 950209, 950210, 950213, 950214, 950215, 950216, 950217, 950220
   950221, 950222, 950223, 950224, 950227, 950228, 950301, 950302, 950303
   950306, 950307, 950308, 950309, 950310, 950313, 950314, 950315, 950316
   950317, 950320, 950321, 950322, 950323, 950324, 950327, 950328, 950329
   950330, 950331, 950403, 950404, 950405, 950406, 950407, 950410, 950411
   950412, 950413, 950414, 950417, 950418, 950419, 950420, 950421, 950424
   950425, 950426, 950427, 950428, 950501, 950502, 950503, 950504, 950505
   950508, 950509, 950510, 950511, 950512, 950515, 950516, 950517, 950518
   950519, 950522, 950523, 950524, 950525, 950526, 950529, 950530, 950531
   950601, 950602, 950605, 950606, 950607, 950608, 950609, 950612, 950613
   950614, 950615, 950616, 950619, 950620, 950621, 950622, 950623, 950626
   950627, 950628, 950629, 950630, 950703, 950704, 950705, 950706, 950707
   950710, 950711, 950712, 950713, 950714, 950717, 950718, 950719, 950720
   950721, 950724, 950725, 950726, 950727, 950728, 950731, 950801, 950802
   950803, 950804, 950807, 950808, 950809, 950810, 950811, 950814, 950815
   950816, 950817, 950818, 950821, 950822, 950823, 950824, 950825, 950828
   950829, 950830, 950831, 950901, 950904, 950905, 950906, 950907, 950908
   950911, 950912, 950913, 950914, 950915, 950918, 950919, 950920, 950921
   950922, 950925, 950926, 950927, 950928, 950929, 951002, 951003, 951004
   951005, 951006, 951009, 951010, 951011, 951012, 951013, 951016, 951017
   951018, 951019, 951020, 951023, 951024, 951025, 951026, 951027, 951030
   951031, 951101, 951102, 951103, 951106, 951107, 951108, 951109, 951110
   951113, 951114, 951115, 951116, 951117, 951120, 951121, 951122, 951123
   951124, 951127, 951128, 951129, 951130, 951201, 951204, 951205, 951206
   951207, 951208, 951211, 951212, 951213, 951214, 951215, 951218, 951219
   951220, 951221, 951222, 951225, 951226, 951227, 951228, 951229


/;

parameter s(t) 'daily series US$/DM exchange rate' /
   920101 0.6601,  920102 0.6532,  920103 0.6484,  920106 0.6597,  920107 0.6623
   920108 0.6636,  920109 0.641 ,  920110 0.6321,  920113 0.63441, 920114 0.6266
   920115 0.6135,  920116 0.6174,  920117 0.6277,  920120 0.6277,  920121 0.6302
   920122 0.6289,  920123 0.6225,  920124 0.6268,  920127 0.6203,  920128 0.6246
   920129 0.6238,  920130 0.6192,  920131 0.62  ,  920203 0.6244,  920204 0.6289
   920205 0.6323,  920206 0.6337,  920207 0.641 ,  920210 0.6339,  920211 0.6268
   920212 0.6219,  920213 0.6161,  920214 0.6165,  920217 0.6165,  920218 0.6072
   920219 0.6085,  920220 0.6075,  920221 0.6044,  920224 0.6051,  920225 0.6028
   920226 0.6076,  920227 0.6112,  920228 0.6101,  920302 0.6086,  920303 0.6026
   920304 0.599 ,  920305 0.5979,  920306 0.5996,  920309 0.6014,  920310 0.5988
   920311 0.599 ,  920312 0.599 ,  920313 0.5999,  920316 0.5999,  920317 0.607
   920318 0.6048,  920319 0.5984,  920320 0.5986,  920323 0.5995,  920324 0.6014
   920325 0.6053,  920326 0.6017,  920327 0.6101,  920330 0.6083,  920331 0.6085
   920401 0.6053,  920402 0.6079,  920403 0.6154,  920406 0.6155,  920407 0.616
   920408 0.6142,  920409 0.6187,  920410 0.6107,  920413 0.6042,  920414 0.6042
   920415 0.6017,  920416 0.6102,  920417 0.5988,  920420 0.6006,  920421 0.5992
   920422 0.6015,  920423 0.6019,  920424 0.6062,  920427 0.6053,  920428 0.6046
```

```
920429 0.6031, 920430 0.6064, 920501 0.6098, 920504 0.6073, 920505 0.6112
920506 0.6143, 920507 0.6105, 920508 0.6086, 920511 0.6103, 920512 0.616
920513 0.6211, 920514 0.6209, 920515 0.6207, 920518 0.6268, 920519 0.6287
920520 0.6244, 920521 0.619 , 920522 0.6179, 920525 0.6179, 920526 0.6205
920527 0.6102, 920528 0.6147, 920529 0.6221, 920601 0.6234, 920602 0.6205
920603 0.624 , 920604 0.6249, 920605 0.6289, 920608 0.6297, 920609 0.6283
920610 0.6275, 920611 0.6341, 920612 0.6337, 920615 0.6369, 920616 0.6396
920617 0.6346, 920618 0.6384, 920619 0.6357, 920622 0.6376, 920623 0.6388
920624 0.6452, 920625 0.6496, 920626 0.6515, 920629 0.6589, 920630 0.6561
920701 0.6572, 920702 0.6618, 920703 0.6596, 920706 0.6616, 920707 0.6709
920708 0.672 , 920709 0.6614, 920710 0.6687, 920713 0.6776, 920714 0.6741
920715 0.6752, 920716 0.6782, 920717 0.6859, 920720 0.6705, 920721 0.6716
920722 0.6727, 920723 0.6736, 920724 0.6691, 920727 0.6754, 920728 0.6791
920729 0.675 , 920730 0.6748, 920731 0.6781, 920803 0.678 , 920804 0.6771
920805 0.6748, 920806 0.6767, 920807 0.6814, 920810 0.6828, 920811 0.6812
920812 0.6832, 920813 0.6873, 920814 0.6824, 920817 0.6826, 920818 0.6863
920819 0.6863, 920820 0.6911, 920821 0.6995, 920824 0.7128, 920825 0.7141
920826 0.7114, 920827 0.7101, 920828 0.7102, 920831 0.7133, 920901 0.7194
920902 0.7184, 920903 0.7077, 920904 0.713 , 920907 0.713 , 920908 0.7179
920909 0.708 , 920910 0.7046, 920911 0.6916, 920914 0.6725, 920915 0.6707
920916 0.6601, 920917 0.6757, 920918 0.666 , 920921 0.6736, 920922 0.6687
920923 0.6673, 920924 0.6741, 920925 0.6757, 920928 0.6875, 920929 0.708
920930 0.707 , 921001 0.7025, 921002 0.7097, 921005 0.7052, 921006 0.7008
921007 0.6906, 921008 0.6745, 921009 0.6727, 921012 0.6727, 921013 0.6821
921014 0.6845, 921015 0.6885, 921016 0.6768, 921019 0.6678, 921020 0.6594
921021 0.6596, 921022 0.6633, 921023 0.6542, 921026 0.6513, 921027 0.6547
921028 0.6467, 921029 0.6498, 921030 0.6489, 921102 0.6371, 921103 0.6376
921104 0.6345, 921105 0.6321, 921106 0.6262, 921109 0.6254, 921110 0.6256
921111 0.6256, 921112 0.6331, 921113 0.6373, 921116 0.6275, 921117 0.6275
921118 0.6293, 921119 0.6357, 921120 0.6229, 921123 0.6262, 921124 0.6264
921125 0.6302, 921126 0.6302, 921127 0.6254, 921130 0.6283, 921201 0.6355
921202 0.6352, 921203 0.6313, 921204 0.6276, 921207 0.6416, 921208 0.6425
921209 0.6367, 921210 0.6329, 921211 0.6335, 921214 0.6369, 921215 0.6378
921216 0.642 , 921217 0.6416, 921218 0.639 , 921221 0.6376, 921222 0.6283
921223 0.6268, 921224 0.6254, 921225 0.6254, 921228 0.6173, 921229 0.619
921230 0.6188, 921231 0.6177, 930101 0.6177, 930104 0.61  , 930105 0.6155
930106 0.6111, 930107 0.6107, 930108 0.6079, 930111 0.613 , 930112 0.6131
930113 0.6156, 930114 0.6156, 930115 0.612 , 930118 0.612 , 930119 0.6217
930120 0.6231, 930121 0.6194, 930122 0.6285, 930125 0.6341, 930126 0.6337
930127 0.6307, 930128 0.6305, 930129 0.6207, 930201 0.6114, 930202 0.6094
930203 0.6079, 930204 0.6048, 930205 0.6031, 930208 0.6037, 930209 0.6051
930210 0.6029, 930211 0.6033, 930212 0.6031, 930215 0.6031, 930216 0.6141
930217 0.6161, 930218 0.6127, 930219 0.6099, 930222 0.6161, 930223 0.6172
930224 0.6146, 930225 0.6112, 930226 0.6112, 930301 0.604 , 930302 0.6095
930303 0.607 , 930304 0.6099, 930305 0.5993, 930308 0.6015, 930309 0.6004
930310 0.6003, 930311 0.602 , 930312 0.601 , 930315 0.6015, 930316 0.6015
930317 0.6017, 930318 0.6088, 930319 0.6111, 930322 0.6105, 930323 0.6132
930324 0.6109, 930325 0.6098, 930326 0.6123, 930329 0.6148, 930330 0.6183
930331 0.6221, 930401 0.6287, 930402 0.6246, 930405 0.627 , 930406 0.6203
930407 0.618 , 930408 0.6221, 930409 0.6221, 930412 0.627 , 930413 0.6325
930414 0.6272, 930415 0.6227, 930416 0.6184, 930419 0.6258, 930420 0.6274
930421 0.6248, 930422 0.6252, 930423 0.6319, 930426 0.6384, 930427 0.6329
930428 0.6327, 930429 0.6335, 930430 0.6306, 930503 0.6327, 930504 0.6347
930505 0.6332, 930506 0.6349, 930507 0.632 , 930510 0.6217, 930511 0.6219
930512 0.6202, 930513 0.619, 930514 0.6205, 930517 0.6197, 930518 0.6159
930519 0.6152, 930520 0.6198, 930521 0.6154, 930524 0.6114, 930525 0.6143
930526 0.6141, 930527 0.6223, 930528 0.6305, 930531 0.6305, 930601 0.6287
930602 0.6254, 930603 0.6248, 930604 0.6148, 930607 0.6167, 930608 0.6153
930609 0.6116, 930610 0.6135, 930611 0.6148, 930614 0.6139, 930615 0.6072
930616 0.6028, 930617 0.6028, 930618 0.5952, 930621 0.5922, 930622 0.5896
930623 0.5912, 930624 0.5857, 930625 0.587 , 930628 0.5888, 930629 0.5925
930630 0.5857, 930701 0.5901, 930702 0.5896, 930705 0.5896, 930706 0.5868
930707 0.5854, 930708 0.5872, 930709 0.5879, 930712 0.5785, 930713 0.5819
930714 0.5828, 930715 0.5802, 930716 0.5824, 930719 0.586 , 930720 0.5886
930721 0.5893, 930722 0.5865, 930723 0.5819, 930726 0.5799, 930727 0.5791
930728 0.5819, 930729 0.5745, 930730 0.5745, 930802 0.5851, 930803 0.586
930804 0.5847, 930805 0.5841, 930806 0.5898, 930809 0.5892, 930810 0.5825
930811 0.5805, 930812 0.5829, 930813 0.5844, 930816 0.5934, 930817 0.5906
930818 0.5954, 930819 0.5926, 930820 0.5972, 930823 0.5933, 930824 0.5954
930825 0.5934, 930826 0.6005, 930827 0.6008, 930830 0.5984, 930831 0.5964
930901 0.6026, 930902 0.6085, 930903 0.6165, 930906 0.6165, 930907 0.6211
930908 0.6192, 930909 0.6254, 930910 0.6266, 930913 0.6209, 930914 0.6213
```

```
930915 0.6268,  930916 0.6238,  930917 0.62  ,  930920 0.6207,  930921 0.6086
930922 0.6141,  930923 0.6085,  930924 0.6103,  930927 0.6146,  930928 0.6194
930929 0.6198,  930930 0.6127,  931001 0.6131,  931004 0.616 ,  931005 0.6156
931006 0.6165,  931007 0.6161,  931008 0.6236,  931011 0.6236,  931012 0.6274
931013 0.624 ,  931014 0.6202,  931015 0.6187,  931018 0.6106,  931019 0.609
931020 0.6094,  931021 0.6002,  931022 0.5967,  931025 0.5935,  931026 0.5945
931027 0.5952,  931028 0.5974,  931029 0.5931,  931101 0.5904,  931102 0.5869
931103 0.5903,  931104 0.5895,  931105 0.5903,  931108 0.5938,  931109 0.5911
931110 0.5919,  931111 0.5919,  931112 0.594 ,  931115 0.5909,  931116 0.5867
931117 0.5879,  931118 0.5839,  931119 0.5824,  931122 0.5868,  931123 0.5877
931124 0.5869,  931125 0.5869,  931126 0.584 ,  931129 0.5851,  931130 0.5828
931201 0.5807,  931202 0.5805,  931203 0.5811,  931206 0.5881,  931207 0.5875
931208 0.5867,  931209 0.5863,  931210 0.5896,  931213 0.5857,  931214 0.5831
931215 0.5819,  931216 0.5847,  931217 0.586 ,  931220 0.5843,  931221 0.5851
931222 0.5879,  931223 0.5901,  931224 0.5901,  931227 0.5877,  931228 0.5888
931229 0.5785,  931230 0.5765,  931231 0.5759,  940103 0.5732,  940104 0.5759
940105 0.5752,  940106 0.5731,  940107 0.5789,  940110 0.5765,  940111 0.5741
940112 0.5767,  940113 0.5712,  940114 0.5716,  940117 0.5716,  940118 0.5727
940119 0.5727,  940120 0.5755,  940121 0.5708,  940124 0.5716,  940125 0.5713
940126 0.5732,  940127 0.5779,  940128 0.5741,  940131 0.5772,  940201 0.5767
940202 0.577 ,  940203 0.5745,  940204 0.5679,  940207 0.5682,  940208 0.5666
940209 0.5683,  940210 0.5706,  940211 0.571 ,  940214 0.5794,  940215 0.5799
940216 0.5802,  940217 0.5794,  940218 0.5841,  940221 0.5841,  940222 0.5798
940223 0.5784,  940224 0.5818,  940225 0.5851,  940228 0.587 ,  940301 0.5855
940302 0.5866,  940303 0.5847,  940304 0.5818,  940307 0.5812,  940308 0.5831
940309 0.5858,  940310 0.5955,  940311 0.5938,  940314 0.5918,  940315 0.5893
940316 0.5927,  940317 0.592 ,  940318 0.59  ,  940321 0.5915,  940322 0.5924
940323 0.5944,  940324 0.5996,  940325 0.6003,  940328 0.5976,  940329 0.5964
940330 0.5968,  940331 0.5983,  940401 0.59  ,  940404 0.59  ,  940405 0.5822
940406 0.5833,  940407 0.5826,  940408 0.5843,  940411 0.5838,  940412 0.5818
940413 0.5851,  940414 0.5847,  940415 0.5835,  940418 0.5855,  940419 0.5879
940420 0.5912,  940421 0.5913,  940422 0.5917,  940425 0.5947,  940426 0.5965
940427 0.599 ,  940428 0.6019,  940429 0.6051,  940502 0.6066,  940503 0.6109
940504 0.6048,  940505 0.6001,  940506 0.6019,  940509 0.6051,  940510 0.5976
940511 0.599 ,  940512 0.5998,  940513 0.5984,  940516 0.5976,  940517 0.5979
940518 0.6033,  940519 0.6035,  940520 0.6077,  940523 0.6092,  940524 0.6046
940525 0.6075,  940526 0.6068,  940527 0.6086,  940530 0.6086,  940531 0.6072
940601 0.6077,  940602 0.6046,  940603 0.5992,  940606 0.5984,  940607 0.6002
940608 0.5986,  940609 0.5993,  940610 0.6002,  940613 0.6077,  940614 0.6079
940615 0.6116,  940616 0.6131,  940617 0.6209,  940620 0.6209,  940621 0.6279
940622 0.6226,  940623 0.6234,  940624 0.6309,  940627 0.6319,  940628 0.6327
940629 0.6311,  940630 0.6303,  940701 0.6266,  940704 0.6266,  940705 0.6327
940706 0.6347,  940707 0.6361,  940708 0.6397,  940711 0.6542,  940712 0.6545
940713 0.65  ,  940714 0.6427,  940715 0.6433,  940718 0.6466,  940719 0.638
940720 0.6398,  940721 0.6289,  940722 0.6258,  940725 0.6292,  940726 0.6307
940727 0.6351,  940728 0.6281,  940729 0.6309,  940801 0.6335,  940802 0.6317
940803 0.6346,  940804 0.6299,  940805 0.6333,  940808 0.6315,  940809 0.6323
940810 0.6315,  940811 0.6408,  940812 0.6436,  940815 0.6441,  940816 0.6421
940817 0.6445,  940818 0.6482,  940819 0.6503,  940822 0.6547,  940823 0.6525
940824 0.647 ,  940825 0.6485,  940826 0.6351,  940829 0.6339,  940830 0.6344
940831 0.6325,  940901 0.6347,  940902 0.6429,  940905 0.6429,  940906 0.6485
940907 0.6452,  940908 0.6431,  940909 0.6502,  940912 0.6479,  940913 0.6485
940914 0.6486,  940915 0.6466,  940916 0.6481,  940919 0.6457,  940920 0.6441
940921 0.6466,  940922 0.6466,  940923 0.6467,  940926 0.6437,  940927 0.6479
940928 0.6448,  940929 0.6463,  940930 0.6446,  941003 0.6435,  941004 0.6466
941005 0.6478,  941006 0.6485,  941007 0.646 ,  941010 0.646 ,  941011 0.647
941012 0.6491,  941013 0.6515,  941014 0.6579,  941017 0.6673,  941018 0.6656
941019 0.666 ,  941020 0.67  ,  941021 0.6671,  941024 0.6696,  941025 0.6692
941026 0.6702,  941027 0.668 ,  941028 0.6623,  941031 0.6648,  941101 0.6683
941102 0.6599,  941103 0.6583,  941104 0.6596,  941107 0.659 ,  941108 0.6627
941109 0.654 ,  941110 0.6532,  941111 0.6532,  941114 0.6477,  941115 0.6472
941116 0.6447,  941117 0.6463,  941118 0.6435,  941121 0.642 ,  941122 0.6425
941123 0.6433,  941124 0.6433,  941125 0.6417,  941128 0.6384,  941129 0.6367
941130 0.6397,  941201 0.6355,  941202 0.6333,  941205 0.636 ,  941206 0.6359
941207 0.6376,  941208 0.6341,  941209 0.6343,  941212 0.6361,  941213 0.6365
941214 0.6372,  941215 0.6367,  941216 0.6362,  941219 0.6352,  941220 0.6369
941221 0.6334,  941222 0.6348,  941223 0.6336,  941226 0.6336,  941227 0.6347
941228 0.6484,  941229 0.6441,  941230 0.6453,  950102 0.6453,  950103 0.6428
950104 0.641 ,  950105 0.6451,  950106 0.6402,  950109 0.6509,  950110 0.6517
950111 0.6521,  950112 0.6543,  950113 0.6522,  950116 0.6522,  950117 0.6528
950118 0.6516,  950119 0.6603,  950120 0.6614,  950123 0.6612,  950124 0.661
950125 0.6593,  950126 0.659 ,  950127 0.66  ,  950130 0.6646,  950131 0.6558
```

```
   950201 0.6586,  950202 0.6586,  950203 0.6549,  950206 0.6531,  950207 0.6509
   950208 0.6527,  950209 0.6542,  950210 0.6582,  950213 0.6575,  950214 0.6628
   950215 0.6627,  950216 0.6713,  950217 0.6746,  950220 0.6746,  950221 0.6786
   950222 0.6804,  950223 0.6815,  950224 0.6839,  950227 0.683 ,  950228 0.6845
   950301 0.6834,  950302 0.6931,  950303 0.7005,  950306 0.71286, 950307 0.7297
   950308 0.7158,  950309 0.7166,  950310 0.7066,  950313 0.711 ,  950314 0.7062
   950315 0.7188,  950316 0.7161,  950317 0.7215,  950320 0.7126,  950321 0.7087
   950322 0.713,   950323 0.7126,  950324 0.7051,  950327 0.711 ,  950328 0.7207
   950329 0.7233,  950330 0.7095,  950331 0.7271,  950403 0.7287,  950404 0.7257
   950405 0.7282,  950406 0.7278,  950407 0.7263,  950410 0.7094,  950411 0.7136
   950412 0.7143,  950413 0.7205,  950414 0.7186,  950417 0.7307,  950418 0.7391
   950419 0.7297,  950420 0.7227,  950421 0.7303,  950424 0.7273,  950425 0.7299
   950426 0.7306,  950427 0.7254,  950428 0.7215,  950501 0.7192,  950502 0.7258
   950503 0.7279,  950504 0.7286,  950505 0.7278,  950508 0.731 ,  950509 0.7241
   950510 0.7205,  950511 0.6983,  950512 0.6908,  950515 0.6969,  950516 0.6925
   950517 0.6913,  950518 0.694 ,  950519 0.69252,  950522 0.694 ,  950523 0.6925
   950524 0.6943,  950525 0.714 ,  950526 0.7265,  950529 0.7265,  950530 0.7202
   950531 0.706 ,  950601 0.7101,  950602 0.71  ,  950605 0.709 ,  950606 0.7089
   950607 0.7075,  950608 0.7097,  950609 0.711 ,  950612 0.7133,  950613 0.7102
   950614 0.713 ,  950615 0.7117,  950616 0.7134,  950619 0.7156,  950620 0.7186
   950621 0.7231,  950622 0.7158,  950623 0.7218,  950626 0.7192,  950627 0.7213
   950628 0.7162,  950629 0.7247,  950630 0.7236,  950703 0.7246,  950704 0.7246
   950705 0.7245,  950706 0.7254,  950707 0.7181,  950710 0.7168,  950711 0.7117
   950712 0.7128,  950713 0.7199,  950714 0.7188,  950717 0.7166,  950718 0.7205
   950719 0.7273,  950720 0.7241,  950721 0.7218,  950724 0.7219,  950725 0.7176
   950726 0.721 ,  950727 0.7229,  950728 0.7241,  950731 0.7214,  950801 0.7267
   950802 0.7153,  950803 0.7188,  950804 0.7158,  950807 0.71  ,  950808 0.7105
   950809 0.7111,  950810 0.7046,  950811 0.695 ,  950814 0.6969,  950815 0.6775
   950816 0.6766,  950817 0.6791,  950818 0.6775,  950821 0.6773,  950822 0.6729
   950823 0.6742,  950824 0.6787,  950825 0.6793,  950828 0.6829,  950829 0.677
   950830 0.677 ,  950831 0.6819,  950901 0.6835,  950904 0.6835,  950905 0.6841
   950906 0.6771,  950907 0.6762,  950908 0.6766,  950911 0.6789,  950912 0.6787
   950913 0.6693,  950914 0.6725,  950915 0.672 ,  950918 0.6745,  950919 0.6718
   950920 0.6845,  950921 0.703 ,  950922 0.7025,  950925 0.6974,  950926 0.6957
   950927 0.7022,  950928 0.7045,  950929 0.7005,  951002 0.6999,  951003 0.6957
   951004 0.697 ,  951005 0.7047,  951006 0.7035,  951009 0.7035,  951010 0.705
   951011 0.7013,  951012 0.705 ,  951013 0.7009,  951016 0.7031,  951017 0.7075
   951018 0.7027,  951019 0.7115,  951020 0.7156,  951023 0.7216,  951024 0.7174
   951025 0.7186,  951026 0.719 ,  951027 0.7116,  951030 0.7095,  951031 0.7107
   951101 0.7058,  951102 0.7026,  951103 0.7058,  951106 0.7077,  951107 0.706
   951108 0.7035,  951109 0.7085,  951110 0.706 ,  951113 0.7055,  951114 0.7075
   951115 0.7104,  951116 0.711 ,  951117 0.7113,  951120 0.711 ,  951121 0.7102
   951122 0.7091,  951123 0.7091,  951124 0.7045,  951127 0.6961,  951128 0.6976
   951129 0.6959,  951130 0.6916,  951201 0.6914,  951204 0.6946,  951205 0.6976
   951206 0.6926,  951207 0.6915,  951208 0.6906,  951211 0.6923,  951212 0.6911
   951213 0.6897,  951214 0.6949,  951215 0.6927,  951218 0.6992,  951219 0.694
   951220 0.6961,  951221 0.695 ,  951222 0.6962,  951225 0.6962,  951226 0.6991
   951227 0.6974,  951228 0.6944,  951229 0.6961
/;

set t1(t) 'all but first';
t1(t+1)=yes;

parameter y(t) 'percentage returns';
y(t1(t)) = 100*log(s(t)/s(t-1));
display y;


scalar mpq 'max(p,q)';
mpq = max(%p%,%q%);

set lag(t) 'observations from t >= max(p,q)';
lag(t+(1+mpq)) = yes;
*display lag

set
   p  /p1 * p%p%/
   q  /q1 * q%q%/
;

variables
   L        'log likelihood (constants taken out)'
```

```
   h(t)     'sigma(t)=sqrt(h(t))'
   e(t)     'residuals'
   alpha_0  'GARCH parameter (constant)'
   alpha(q) 'GARCH parameters (q)'
   beta(p)  'GARCH parameters (p)'
   constant 'information structure: y = constant + e'
;


equations
   loglike      'transformed log likelihood function'
   def_h(t)     'h(t) structure'
   stat_garch   'stationarity condition (GARCH model)'
   stat_igarch  'stationarity condition (IGARCH model)'
   series(t)    'defines the time series'
;

loglike..        L =e= sum(lag(t), log(h(t)) + sqr(e(t))/h(t) );
def_h(lag(t))..  h(t) =e= alpha_0 + sum(q, alpha(q)*sqr(e(t-ord(q))))
                                  + sum(p, beta(p)*h(t-ord(p))) ;
stat_garch..     sum(q,alpha(q)) + sum(p,beta(p)) =l= 1;
stat_igarch..    sum(q,alpha(q)) + sum(p,beta(p)) =e= 1;
series(lag(t)).. y(t) =e= constant + e(t);

*
* lower bounds
*
alpha_0.lo = 0;
alpha.lo(q) = 0;
beta.lo(p) = 0;
h.lo(t) = 0.01;

*
* upper bounds
*
alpha_0.up = 100;
alpha.up(q) = 100;
beta.up(p) = 100;
h.up(t) = 100;

*
* initial values
*
alpha_0.l = .1;
alpha.l(q) = .1;
beta.l(p) = .1;
h.l(t) = .1;
e.l(t) = .1;

model garch /loglike, def_h, stat_garch, series/;
model igarch /loglike, def_h, stat_igarch, series/;

solve garch using nlp minimizing L;
solve igarch using nlp minimizing L;
```

5. M Regression

M-estimation, introduced by [27], is a form of "robust" regression that is a mix of least squares estimation and LAD estimation (see section 2). It uses a least squares criterion when the residuals are small, but switches to a LAD measure for the large residuals. The Huber M-estimation problem is stated as:

M-ESTIMATION

$$\underset{\beta}{\text{minimize}} \quad \sum_i \rho(\epsilon_i)$$

$$\text{subject to} \quad y_i = \sum_j x_{i,j}\beta_j + \epsilon_i$$

where $\rho(.)$ is defined as

$$\rho(\epsilon_i) = \begin{cases} \epsilon_i^2 & \text{if } |\epsilon_i| \le k \\ 2k|\epsilon_i| - k^2 & \text{otherwise} \end{cases} \tag{50}$$

for some $k > 0$.

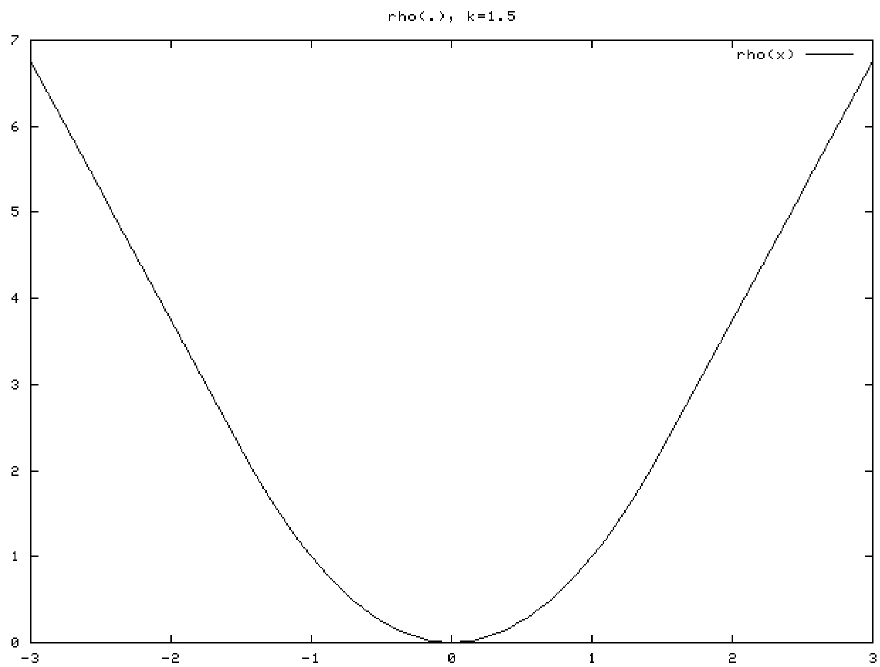

FIGURE 8. Graph of the function $\rho(x)$, $k = 1.5$.

Although this model is perfectly smooth and has continuous derivatives it is not easily formulated in GAMS. The objective function would require an if-then-else construct which is not part of the GAMS language. A direct formulation therefore would require the use of discrete variables, which makes the model an MINLP model.

In [18] and [52] a linear complementarity model is conceived as follows. Write the problem as

$$\begin{aligned} &\min \sum_{i=1}^{\ell} \rho([Ax-b]_i) \\ &\rho(t) = \begin{cases} \frac{1}{2}t^2 & |t| \le \gamma, \\ \gamma|t| - \frac{1}{2}\gamma^2 & |t| > \gamma \end{cases} \end{aligned} \tag{51}$$

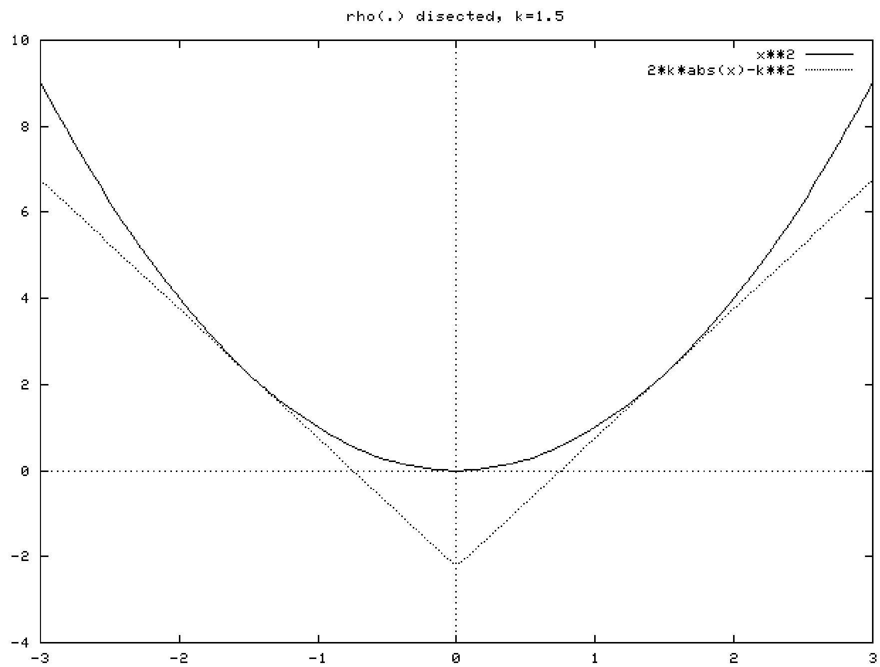

FIGURE 9. Graph of the functions that form $\rho(x)$, $k = 1.5$.

Note that the parameters to be estimated are x here. The data is stored in A and b. If we write this as:

$$\begin{aligned} \min \sum_{i=1}^{\ell} \rho(v_i) \\ v = Ax - b \end{aligned} \tag{52}$$

then we can form the Lagrangean:

$$\mathcal{L}(x, v, \lambda) = \sum_{i=1}^{\ell} \rho(v_i) - \lambda^T (v - Ax + b) \tag{53}$$

If we set the derivates to zero, we get:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x} &= 0 \Rightarrow A^T \lambda = 0 \\ \frac{\partial \mathcal{L}}{\partial v} &= 0 \Rightarrow \frac{\partial \rho(v_i)}{\partial v_i} - \lambda_i = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= 0 \Rightarrow v = Ax - b \end{aligned} \tag{54}$$

The partial derivatives of $\rho(v_i)$ are as follows:

$$\frac{\partial \rho(v_i)}{\partial v_i} = \begin{cases} v_i & |v_i| \le \gamma \\ \gamma \operatorname{sign}(v_i) & |v_i| > \gamma \end{cases} \tag{55}$$

where $\mathrm{sign}(x)$ is defined as:

$$\mathrm{sign}(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{56}$$

This results in:

$$\frac{\partial \mathcal{L}}{\partial v} = 0 \Rightarrow \begin{cases} \lambda_i - [Ax-b]_i = 0 & |v_i| \leq \gamma \\ \lambda_i - \gamma = 0 & v_i \geq \gamma \\ \lambda_i + \gamma = 0 & -v_i \geq \gamma \end{cases} \tag{57}$$

The whole system can now be written as a mixed linear complementarity problem:

$$\begin{aligned} \lambda - Ax + b + s^+ - s^- &= 0 \\ A^T\lambda &= 0 \\ \lambda + \gamma e \geq 0 &\perp s^+ \geq 0 \\ -\lambda + \gamma e \geq 0 &\perp s^- \geq 0 \end{aligned} \tag{58}$$

where $\perp$ is used to indicate complementarity between an equation and a variable.

The complete LCP model is listed below. In the model we use the data from section 1.1.

5.0.2. *Model huberlcp.gms.* [16]

```
$ontext

   Huber M regression by LCP

   Erwin Kalvelagen, november 2001

$offtext

set i 'number of cases' /i1*i40/;
set j 'coefficient to estimate' /'constant','coeff1'/;

$include expdata.inc


parameter b(i);
b(i) = data(i,'expenditure');

parameter A(i,j);
A(i,'constant') = 1;
A(i,'coeff1') = data(i,'income');

scalar gamma /1.5/;

variables
   w(i)   'lagrange multipliers'
   x(j)   'parameters to estimate'
   lambda1(i) 'slacks'
   lambda2(i) 'slacks'
;
positive variables lambda1, lambda2;

equations
   e1(i)
   e2(j)
   compl1(i)
   compl2(i)
;

e1(i).. w(i) - sum(j, A(i,j)*x(j)) + b(i) + lambda2(i) - lambda1(i) =e= 0;
```

[16] http://amsterdamoptimization.com/models/statistics/huberlcp.gms

```
e2(j).. sum(i, A(i,j)*w(i)) =e= 0;
compl1(i).. w(i) + gamma =g= 0;
compl2(i).. -w(i) + gamma =g= 0;

model m /e1,e2,compl1.lambda1,compl2.lambda2/;
solve m using mcp;
```

It is sometimes suggested to use $k = 1.5\hat{\sigma}$, with

$$\hat{\sigma} = 1.483 \text{ MAD} \tag{59}$$

where MAD is the median of the absolute deviations $|\epsilon_i|$.

For more information on M estimation and robust statistics see for instance [28].

6. Maximum Entropy Methods

The availability of data is often a problem in practical applied general equilibrium modeling. The lack of data can prevent the use of standard regression techniques to estimate model parameters. A recent technique called *Maximum Entropy*[20] has become a popular device in this field to get estimates[43, 42, 2].

Let X be a random variable with a discrete distribution $P(X = x_k) = \pi_k$ for $k = 1, \ldots, N$, where $\sum_k \pi_k = 1$. The entropy-information measure introduced by [45] is defined by

$$S(\pi) = -\sum_{k=1}^{N} \pi_k \ln(\pi_k) \tag{60}$$

Following [12] we consider again the general linear model (GLM):

$$y = X\beta + e \tag{61}$$

Suppose we have prior information in the form of bounds on the parameters β_k and on the error terms. Let

$$z_{k,1} \le \beta_k \le z_{k,2} \tag{62}$$

then we can write the linear combination:

$$\beta_k = p_k z_{k,1} + (1 - p_k) z_{k,2} = \begin{pmatrix} z_{k,1} & z_{k,2} \end{pmatrix} \begin{pmatrix} p_k \\ 1 - p_k \end{pmatrix} \tag{63}$$

with $p_k \in [0, 1]$. Extending this to M "support points" for for each parameter β_k, we get the following linear system:

$$\begin{aligned} &\beta_k = \sum_{j=1}^{M} z_{k,j} p_{k,j} \\ &\sum_{j=1}^{M} p_{k,j} = 1 \\ &p_{k,j} > 0 \end{aligned} \tag{64}$$

Similarly we can write a set of J support points for each error term e_i:

$$(65)\quad \begin{aligned} e_i &= \sum_{j=1}^{J} v_{i,j} w_{i,j} \\ \sum_{j=1}^{J} w_{i,j} &= 1 \\ w_{i,j} &> 0 \end{aligned}$$

Using matrix notation we can now write:

$$(66)\quad y = XZp + Vw$$

The Maximum Entropy Estimation problem can now be stated as:

$$(67)\quad \begin{aligned} \max H(p,w) &= -p^T \ln(p) - w^T \ln(w) \\ y &= XZp + Vw \\ (I_K \otimes i_M^T)p &= i_K \\ (I_N \otimes i_J^T)w &= i_N \\ p, w &> 0 \end{aligned}$$

where $\otimes$ is the Kronecker product and i_N is an N vector of ones.

The prior information is the model below is as follows. We assume a parameter support for parameter *const* of $z^T_{const}\{-50,-25,0,25,50\}$. All other parameters have a support of $z^T_k = \{-20,-10,0,10,20\}$. I.e. we slightly wider bounds for the constant term coefficient. The error support is roughly $\pm 3\sigma$ with σ being the sample standard deviation. I.e. $v^T_i = \{-10,-5,0,5,10\}$.

6.0.3. *Model gme.gms.* [17]

```
$ontext

  Generalized Maximum Entropy

  Erwin Kalvelagen, march 2003


  References:
    Maximum entropy estimation in economic models with linear
    inequality restrictions, Randall C. Campbell, R. Carter Hill
    Department of Economics, Louisiana State University,
    Baton Rouge, LA 70803,USA


$offtext


set i 'cases' /case1*case116/;

set k 'parameters' /const, famsize, unemp, highschl, college, medinc, d90/;

table data(i,*)
             pov   famsize  unemp highschl college   medinc    d90
case1       18.1     3.15    10.8    53.7    22.3    22.863     0
case2        8.7     3.20     6.9    53.7    32.4    17.240     0
case3        7.5     2.87     7.2    64.2    12.6    18.065     0
case4        9.5     2.93    10.5    54.7    16.9    16.301     0
case5        7.5     2.88     9.6    62.5    13.8    17.909     0
case6        8.8     3.18     8.2    52.3    12.3    17.842     0
case7        6.1     3.16     5.8    56.2    25.5    26.513     0
```

[17] http://amsterdamoptimization.com/models/statistics/gme.gms

```
case8       11.4    3.07    15.5    57.1    10.0    15.911     0
case9        6.7    3.00     9.5    63.8    17.4    20.182     0
case10      11.4    3.33     8.9    48.2    15.5    18.399     0
case11      10.5    3.20    12.2    53.7     9.3    16.650     0
case12       9.4    3.06     7.8    58.4    18.0    18.479     0
case13      12.7    3.73     9.6    41.3     9.6    16.659     0
case14       7.3    3.01     6.0    62.1    12.1    18.366     0
case15      10.2    3.28     7.7    50.3    11.8    18.780     0
case16      12.4    3.45     8.8    48.6    10.1    16.164     0
case17      11.2    2.76    10.1    56.3    10.1    13.522     0
case18       7.1    3.14    13.6    61.2    11.9    17.563     0
case19      10.5    3.34     6.0    51.4    18.4    21.135     0
case20      12.4    3.31    10.2    49.4    10.7    17.327     0
case21       4.9    3.00    11.5    51.6    38.3    29.721     0
case22       9.6    2.86     3.9    58.3    15.4    15.833     0
case23       9.5    3.04     8.3    58.8    17.6    17.695     0
case24      11.8    3.40    11.0    49.9    10.5    16.563     0
case25      11.8    2.99     7.5    60.0    12.3    15.617     0
case26       6.4    3.01     6.5    65.5    22.7    20.215     0
case27       8.9    3.33     9.1    51.4    19.6    20.005     0
case28       6.0    3.03     5.5    57.6    17.8    22.426     0
case29       6.5    2.93     9.5    64.1    17.9    18.842     0
case30       5.2    3.27     4.1    57.8    22.6    25.919     0
case31       7.0    3.12    16.9    60.7    16.7    21.662     0
case32       8.2    2.99     9.0    63.9    14.5    17.227     0
case33       8.8    3.15     6.8    56.2    12.7    18.682     0
case34       8.9    3.11     9.0    58.8    19.2    20.949     0
case35      10.8    3.57    13.5    45.9    10.6    18.937     0
case36       9.1    3.26     7.4    57.9    13.1    20.039     0
case37       8.4    3.16     7.0    57.1    20.9    20.306     0
case38      10.3    3.11     6.1    45.8    28.2    20.911     0
case39      10.8    3.20    10.2    51.1    11.5    19.120     0
case40       8.0    2.95     6.9    57.9    18.9    18.198     0
case41       4.5    3.13    13.8    56.2    25.4    27.279     0
case42       6.7    3.16     3.5    54.5    24.6    21.630     0
case43       5.3    3.29     5.8    53.2    26.3    26.662     0
case44       8.2    3.07     4.5    54.4    23.4    20.734     0
case45       8.8    3.06     7.9    63.2    12.4    17.024     0
case46       9.3    2.90    11.9    60.2    17.9    18.221     0
case47       9.6    3.03    13.5    61.6    14.0    16.686     0
case48       8.1    3.25     8.7    63.1    13.7    21.606     0
case49       7.1    3.04     7.0    58.4    19.2    21.269     0
case50      10.0    3.24    12.7    50.3    11.7    18.656     0
case51       9.0    3.23    13.4    53.3    14.4    18.545     0
case52      11.0    3.06    11.7    60.5     9.0    15.849     0
case53       9.1    3.03    17.6    60.9    13.6    16.110     0
case54      13.2    3.40     8.6    45.7    10.1    16.172     0
case55       9.2    2.93    12.5    63.6    13.7    16.907     0
case56       6.1    3.42     5.4    57.8    18.1    23.612     0
case57       9.1    3.14     9.3    46.5    27.0    20.495     0
case58      14.4    3.22    16.6    54.5     9.3    13.751     0
case59       8.1    2.59     5.3    52.6    28.8    45.037     1
case60      16.7    2.47     8.2    63.6    24.0    29.276     1
case61       6.3    2.41     7.2    68.5    14.0    35.062     1
case62      12.2    2.48     9.4    58.1    19.5    28.314     1
case63       7.5    2.50    10.5    67.2    14.4    32.211     1
case64      10.4    2.84    15.7    51.8    11.1    28.230     1
case65       5.5    2.64     5.6    54.9    31.6    51.651     1
case66      12.7    2.63    12.5    60.9    10.0    26.992     1
case67       5.8    2.66     6.1    65.1    20.8    39.823     1
case68      16.8    2.96    12.6    49.3    16.9    29.970     1
case69      14.1    2.77    15.5    57.5     9.4    27.216     1
case70      12.8    2.49     8.8    60.5    20.0    30.357     1
case71      20.8    3.26    21.3    43.5     9.7    25.147     1
case72       9.2    2.35     8.8    68.2    13.5    30.460     1
case73      13.7    2.92    11.8    54.3    13.3    31.714     1
case74      15.0    3.08    12.8    56.6     9.0    27.614     1
case75      12.3    2.38    11.1    60.2    10.7    26.563     1
case76      10.4    2.66    10.0    61.1    11.7    31.803     1
case77      11.6    2.91     8.0    46.7    23.3    39.035     1
case78      13.1    3.05    14.0    51.7    11.7    30.035     1
case79       3.0    2.33     4.0    47.9    44.0    59.147     1
```

```
case80      10.7     2.42     6.3     61.0     16.8     29.468     1
case81      11.0     2.57    10.9     60.9     17.8     31.276     1
case82      15.4     3.17    14.6     51.1     12.0     28.269     1
case83      11.6     2.49    12.4     61.0     11.2     27.407     1
case84       6.7     2.48    12.5     65.9     21.9     35.932     1
case85       8.5     2.96    10.9     51.4     21.5     36.223     1
case86       4.6     2.54     5.9     58.4     22.3     42.789     1
case87       5.8     2.51     7.0     64.2     22.1     36.942     1
case88       5.2     2.87     4.8     53.4     27.8     51.167     1
case89       5.3     2.66     6.8     62.4     22.7     42.805     1
case90       9.8     2.41    12.0     67.6     15.1     29.967     1
case91       8.4     2.85    10.7     59.5     14.6     37.694     1
case92       9.8     2.58     6.3     59.2     23.0     37.841     1
case93       7.3     3.15    17.2     54.0     14.4     39.637     1
case94      10.3     2.97     8.0     60.5     14.9     36.977     1
case95       8.1     2.69     6.1     56.6     25.3     39.798     1
case96       9.7     2.29     5.6     43.0     35.0     40.561     1
case97      12.0     2.94    12.0     55.4     13.2     34.701     1
case98       6.8     2.53     5.8     60.4     22.9     37.086     1
case99       4.3     2.64     4.2     52.8     31.3     53.430     1
case100      7.4     2.73     6.0     53.4     26.6     41.289     1
case101      5.0     2.81     5.5     49.4     32.6     53.670     1
case102      6.2     2.66     8.0     52.2     29.7     43.130     1
case103     11.0     2.58    10.3     64.7     13.7     30.332     1
case104      5.7     2.45    10.5     59.6     15.9     29.911     1
case105     11.6     2.48    12.5     63.2     14.2     26.073     1
case106      6.0     2.88     7.0     64.0     18.7     42.392     1
case107      5.2     2.55     5.7     59.9     24.5     41.961     1
case108     11.4     2.91    14.3     55.4     13.0     32.923     1
case109     12.2     2.75    17.6     56.9     15.4     31.842     1
case110     12.6     2.60    12.4     62.0     10.2     25.946     1
case111     15.1     2.49    14.5     61.3     12.9     25.009     1
case112     18.0     3.12    17.1     48.4     11.8     26.697     1
case113      6.9     2.46     8.3     65.3     14.7     31.464     1
case114      5.0     3.02     7.0     56.4     23.0     50.091     1
case115      9.8     2.63     7.2     48.8     30.3     36.866     1
case116     16.0     2.85    14.1     59.0      9.5     24.364     1
;

parameters X(i,k) 'exogenous matrix';
X(i,k) = data(i,k);
X(i,'const') = 1;
display X;

parameter y(i) 'endogenous variable';
y(i) = data(i,'pov');
display y;

*-------------------------------------------------------------------
* OLS model
*-------------------------------------------------------------------

variables b(k);
variables sse, e(i);
equations sumsq,linear(i);

linear(i)..   y(i) =e= sum(k, X(i,k)*b(k)) + e(i);
sumsq..       sse =e= sum(i, sqr(e(i)));

model ols /linear,sumsq/;
solve ols minimizing sse using nlp;

display b.l,sse.l;

*-------------------------------------------------------------------
* GME model
*-------------------------------------------------------------------

set j /j1*j5/;
table z(k,j) 'parameter support for GME model'
             j1    j2    j3    j4    j5
const       -50   -25     0    25    50
```

```
famsize     -20   -10     0    10    20
unemp       -20   -10     0    10    20
highschl    -20   -10     0    10    20
college     -20   -10     0    10    20
medinc      -20   -10     0    10    20
d90         -20   -10     0    10    20
;

parameter errsupport(j) 'error support' /
  j1 -10
  j2  -5
  j3   0
  j4   5
  j5  10
/;

parameter v(i,j);
v(i,j) = errsupport(j);

variables p(k,j), w(i,j);
p.lo(k,j) = 0.0001;
w.lo(i,j) = 0.0001;

variable entrpy;
equations
   parmsupp(k) 'parameter support'
   errsupp(i)  'error support'
   normp(k)    'normalize p'
   normw(i)    'normalize w'
   obj         'maximize entropy'
;

obj..           entrpy =e= -sum((k,j), p(k,j)*log(p(k,j)))-sum((i,j), w(i,j)*log(w(i,j)));
parmsupp(k)..   b(k) =e= sum(j, z(k,j)*p(k,j));
errsupp(i)..    e(i) =e= sum(j, v(i,j)*w(i,j));
normp(k)..      sum(j, p(k,j)) =e= 1;
normw(i)..      sum(j, w(i,j)) =e= 1;


model gme /obj,parmsupp,errsupp,linear,normp,normw/;
solve gme maximizing entrpy using nlp;

display b.l,entrpy.l;


*-------------------------------------------------------------------
* IRLS model (inequality restricted least squares)
*-------------------------------------------------------------------

*
* add sign restriction on college coefficient as it has the wrong
* sign in the OLS estimates.
*
b.up('college') = 0;

solve ols minimizing sse using nlp;
display b.l,sse.l;

* repair
b.up('college') = INF;

*-------------------------------------------------------------------
* RGME model
*-------------------------------------------------------------------

*
* introduce sign restrictions on the parameters
* by providing appropriate parameter support
*
table z2(k,j) 'parameter support for GME model'
             j1    j2    j3    j4    j5
const       -50   -25     0    25    50
```

solver	seconds
MINOS 5.51	1.4
SNOPT 6.2	1.4
CONOPT 3	2.6
PATHNLP	0.1

TABLE 3. Time for first GME model

```
famsize       0     5    10    15    20
unemp         0     5    10    15    20
highschl    -20   -15   -10    -5     0
college     -20   -15   -10    -5     0
medinc      -20   -15   -10    -5     0
d90         -20   -10     0    10    20
;
z(k,j) = z2(k,j);

solve gme maximizing entrpy using nlp;

display b.l,entrpy.l;
```

The last two models deal with sign restrictions on the parameters. For the GME model we can do this indirectly by providing a support that is not around zero but rather one sided. The data for this model is from [41].

The references [43, 42] includes complete GAMS code showing how these techniques can be used to estimate a SAM (Social Accounting Matrix) opposed to the more traditional RAS method. In [39] it is emphasized that RAS is actually a form of Entropy Optimization. A large scale entropy estimation application using GAMS and PATH is documented in [21]. As the NLP models tend to have many superbasic variables for large some instances, reformulating the optimization model into a complementarity problem can lead to a large performance gain. With the PATHNLP solver this process is completely automated. Indeed in the above example the entropy models solve very fast by using the PATHNLP solver compared to the traditional NLP solvers MINOS, SNOPT, and CONOPT as shown in table 6.0.3.

6.1. **Confidence intervals for max entropy models.** Maximum entropy models do not provide standard errors or confidence intervals. A bootstrap (resampling) approach can be used to calculate those. A complete example is given in [33].

7. Classification

A problem related used in Machine Learning and Data Mining is the classification problem where we try to find a simple rule to separate data points[37, 16]. I.e. given two sets of data points A and B we would like to find a discriminant function f such that $f(x) < 0$ for $x \in A$ and $f(x) > 0$ for $x \in B$.

The simplest case is the linear classification problem, as depicted in figure 7. Mathematically we are seeking to find a plane $x^T w = \gamma$ with the property that all data points $x \in A$ have $x^T w > \gamma$ and all data points in B have $x^T w < \gamma$ or

$$\begin{aligned} Aw &\geq e\gamma + e \\ Bw &\leq e\gamma - e \end{aligned} \tag{68}$$

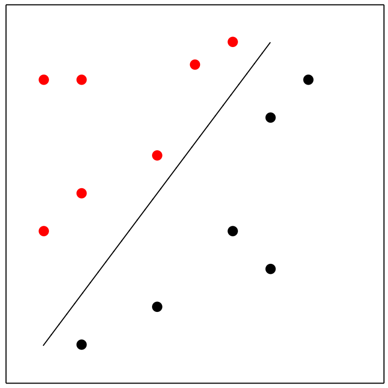

FIGURE 10. Linear separation

where e is a vector of ones. The following linear program[38]:

$$\begin{aligned}
&\min_{w,\gamma,y,z} \frac{e^T y}{n} + \frac{e^T z}{m} \\
&Aw + y \geq e\gamma + e \\
&Bw - z \leq e\gamma - e \\
&y, z \geq 0
\end{aligned} \tag{69}$$

will find a separating hyperplane if it exists. In this case $y = 0$, $z = 0$. If the points can not be separated by a hyperplane, it will find a plane that minimizes the average sum of the violations. Another advantage of this formulation is that $w = 0$ is naturally eliminated. Other formulations are discussed in [19, 26].

7.0.1. *Model classify.gms.* [18]

```
$ontext

  classification through linear programming

  Reference:
     O. L. Mangasarian, W. N. Street, and W. W. Wolberg,
     "Breast Cancer Diagnosis and Prognosis Via Linear Programming,"
     Operations Research. 43 (1995), 570-577

  Data from http://cgm.cs.mcgill.ca/~beezer/cs644/example.html

$offtext


set
  d '2d dataset' /x,y/
  i 'cases for points A' /i1*i13/
  j 'cases for points B' /j1*j17/
;


table pa(i,d)
       x    y
 i1    3    1
```

[18] http://amsterdamoptimization.com/models/statistics/classify.gms

```
 i2   2    2
 i3   4    3
 i4   3    4
 i5   0    5
 i6   1    5
 i7   5    5
 i8   1    6
 i9   2    7
 i10  3    7
 i11  1    8
 i12  0    9
 i13  2   10
;

table pb(j,d)
      x    y
 j1   9    0
 j2  10    0
 j3  10    1
 j4  10    3
 j5   8    4
 j6  12    4
 j7   9    5
 j8  11    5
 j9   6    6
 j10  8    6
 j11 10    6
 j12  8    7
 j13  9    7
 j14  8    8
 j15  5    8
 j16  5    9
 j17  7    9
;

scalar n,m;
n = card(i);
m = card(j);

variables obj,gamma,w(d);
positive variables y(i),z(j);
equations objdef,eqa(i),eqb(j);

objdef.. obj =e= sum(i,y(i))/n + sum(j,z(j))/m;
eqa(i).. sum(d,pa(i,d)*w(d)) + y(i) =g= gamma+1;
eqb(j).. sum(d,pb(j,d)*w(d)) - z(j) =l= gamma-1;

model classify /objdef,eqa,eqb/;
solve classify minimizing obj using lp;

display gamma.l,w.l;
```

When we replace the data by:

```
set
  d '2d dataset' /x,y/
  i 'cases for points A' /i1*i16/
  j 'cases for points B' /j1*j20/
;


table pa(i,d)
      x    y
 i1   3    1
 i2   2    2
 i3   4    3
 i4   3    4
 i5   0    5
 i6   1    5
 i7   5    5
 i8   1    6
 i9   2    7
```

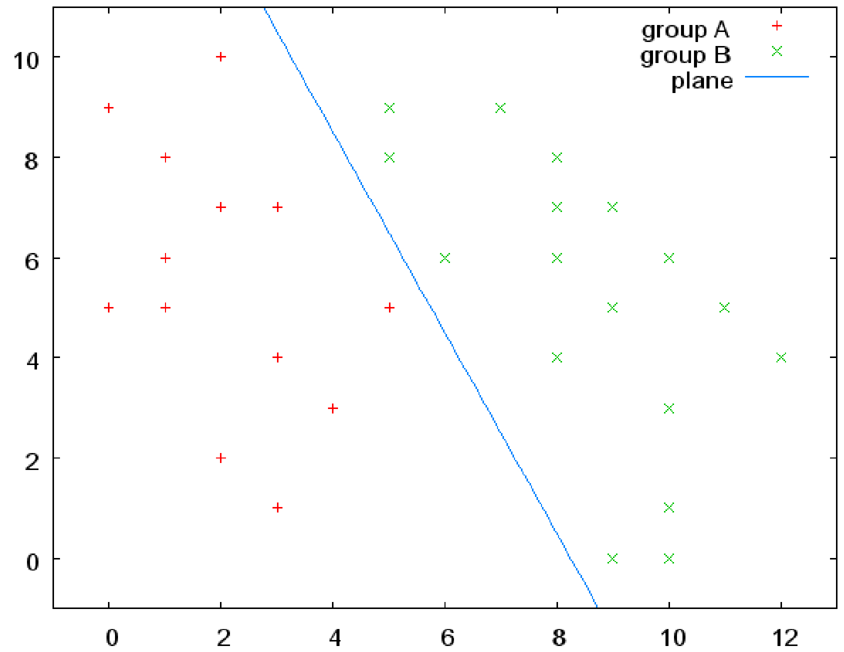

FIGURE 11. Results of classify.gms.

```
 i10  3    7
 i11  1    8
 i12  0    9
 i13  2   10
 i14  6    7
 i15  7    5
 i16  6    4
;

table pb(j,d)
       x    y
 j1    9    0
 j2   10    0
 j3   10    1
 j4   10    3
 j5    8    4
 j6   12    4
 j7    9    5
 j8   11    5
 j9    6    6
 j10   8    6
 j11  10    6
 j12   8    7
 j13   9    7
 j14   8    8
 j15   5    8
 j16   5    9
 j17   7    9
 j18   5    3
 j19   3    9
 j20   8    3
;
```

the points can not be separated linearly. The results are depicted in figure 12.

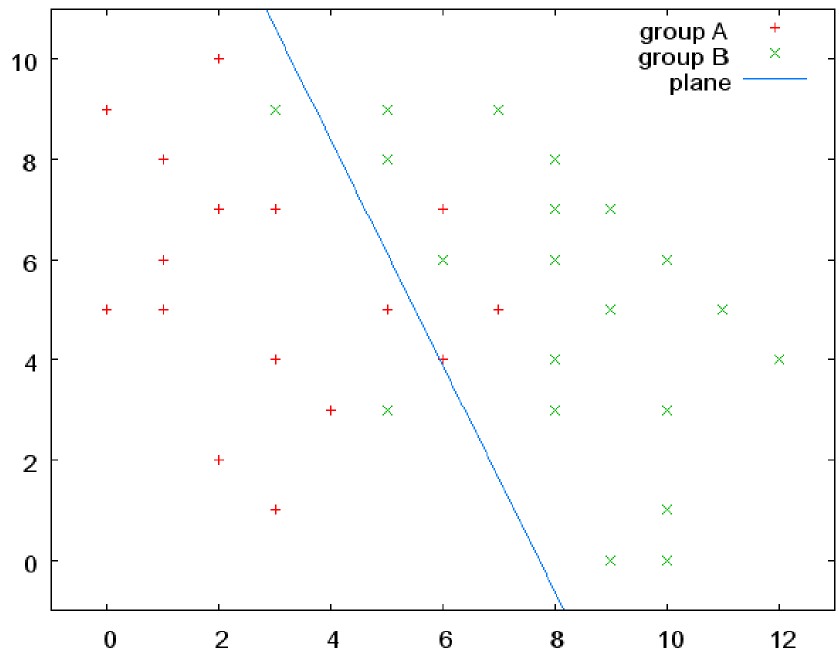

FIGURE 12. Data can not be separated linearly

REFERENCES

1. R. D. Armstrong and M. T. Kung, *An algorithm to select the best subset for a least absolute value regression problem*, Optimization in Statistics (S. H. Zanakis and J. S. Rustagi, eds.), Studies in the Management Sciences, vol. 19, North-Holland Publishing Company, 1982, pp. 67–80.
2. Channing Arndt, Sherman Robinson, and Finn Tarp, *Parameter Estimation for a Computable General Equilibrium Model: A Maximum Entropy Approach*, `http://www.ifpri.org/divs/tmd/dp/papers/tmdp40.pdf`, February 1999.
3. K. Arrow, H. Chenery, B. Minhas, and R. Solow, *Capital-Labor Substitution and Economic Efficiency*, Review of Economics and Statistics **43** (1961), 225–250.
4. Jushan Bai, *Least Absolute Deviation Estimation of a Shift*, Econometric Theory **11** (1995), no. 3, 403–436.
5. Jushan Bai and Pierre Perron, *Testing for and Estimation of Multiple Structural Changes*, Econometrica (1998), 47–78.
6. ______, *Computation and Analysis of Multiple Structural Change Models*, Journal of Applied Econometrics **18** (2003), no. 1, 1–22.
7. Herman J. Bierens, *The Tobit model*, September 2004.
8. D. Birkes and Y. Dodge, *Alternative Methods of Regression*, Wiley, 1993.
9. P. Bloomfield and W. L. Steiger, *Least Absolute Deviations. Theory, Applications and Algorithms*, Birkhäuser, 1983.
10. T. Bollerslev, *Generalized Autoregressive Conditional Heteroskedasticity*, Journal of Econometrics **31** (1986), 307–327.
11. T. Bollerslev, R. Y. Chou, and K. F. Kroner, *ARCH Modeling in Finance*, Journal of Econometrics **52** (1992), 5–59.
12. Randall C. Campbell and R. Carter Hill, *Maximum Entropy Estimation in Economic Models with Linear Inequality Restrictions*, Tech. Report 2001-11, Louisiana State University, Department of Economics, 2001.

13. V. Chvatal, *Linear Programming*, Freeman, 1983.
14. D. R. Cox and E. J. Snell, *Applied Statistics: Principles and Examples*, Chapman and Hall, London, 1981.
15. Terry Dielman and Roger Pfaffenberger, *LAV (least absolute value) estimation in linear regression: a review*, Optimization in Statistics (S. H. Zanakis and J. S. Rustagi, eds.), Studies in the Management Sciences, vol. 19, North-Holland Publishing Company, 1982, pp. 31–52.
16. R. O. Duda, P. E. Hart, and David G. Stork, *Pattern Classification*, Wiley, 2000.
17. M. Fuss, D. McFadden, and Y. Mundlak, *A survey of functional forms in the economic analysis of production*, The Theory of Production (M. Fuss and D. McFadden, eds.), Production Economics: A Dual Approach to Theory and Applications, vol. I, North-Holland, 1978.
18. Michael Gertz and Stephen Wright, *Object-oriented Software for Quadratic Programming*, Tech. Report ANL/MCS-P891-1000, Argonne National Laboratory, Mathematics and Computer Science Division, 2001.
19. F. Glover, *Improved Linear Programming Models for Discriminant Analysis*, Decision Sciences **21** (1990), 771–785.
20. Amos Golan, George G. Judge, and Douglas Miller, *Maximum Entropy Econometrics: Robust Estimation with Limited Data*, Wiley, 1996.
21. Amos Golan, Jeffrey M. Perloff, and Edward Zhihua Shen, *Estimating a Demand System with Nonnegativity Constraints: Mexican Meat Demand*, Tech. report, University of California, Berkeley, July 2000.
22. S. F. Gray, *GARCH Code*, `http://www.duke.edu/~sg12/software/garch/garch.htm`.
23. William H. Greene, *Multiple roots of the Tobit log-likelihood*, Journal of Econometrics **46** (1990), no. 3, 365–380.
24. ______, *Econometric Analysis*, 5th ed., Prentice-Hall, 2003.
25. W. E. Griffiths, R. C. Hill, and G. G. Judge, *Learning and Practicing Econometrics*, Wiley, 1993.
26. R. C. Grinold, *Mathematical Methods for Pattern Classification*, Management Science **19** (1972), 272–289.
27. P. J. Huber, *Robust Estimation of a Location Parameter*, Annals of Mathematical Statistics **35** (1964), 73–101.
28. ______, *Robust Statistics*, Wiley, 1981.
29. Jr. John E. Dennis, David M. Gay, and Roy E. Welsch, *Algorithm 573: NL2SOL – An Adaptive Nonlinear Least-Squares Algorithm*, ACM Trans. Math. Softw. **7** (1981), no. 3, 369–383.
30. Dale W. Jorgenson, *Econometrics: Econometric Modeling of Producer Behavior*, MIT Press, 2000.
31. Erwin Kalvelagen, *A Linear Regression Solver for GAMS*, `http://amsterdamoptimization.com/pdf/regression.pdf`, 2004.
32. ______, *A Non-Linear Regression Solver for GAMS*, `http://amsterdamoptimization.com/pdf/nlregression.pdf`, 2007.
33. ______, *Bootstrap code for forming confidence intervals in Max Entropy estimation*, `http://amsterdamoptimization.com/pmwiki/pmwiki.php?n=Download.Bootstrap`, 2007.
34. ______, *Critical values for the Student's t distribution*, `http://amsterdamoptimization.com/pmwiki/pmwiki.php?n=Download.Qt`, 2007.
35. J. Kierkegaard, *Estimation of Nonlinear Stochastic Processes*, M.Sc. thesis IMM-EKS-2000-16, Technical University of Denmark, April 2000.
36. H. van Maaren and T. Terlaky, *Inverse barriers and CES-functions in linear programming*, Tech. Report 95-76, Delft University of Technology, 1995.
37. O. L. Mangasarian, *Linear and Nonlinear Separation of Patterns by Linear Programming*, Operations Research **13** (1965), 444–452.
38. O. L. Mangasarian, W. N. Street, and W. W. Wolberg, *Breast Cancer Diagnosis and Prognosis Via Linear Programming*, Operations Research (1995), no. 43, 570–577.
39. Robert McDougall, *Entropy Theory and RAS are Friends*, GTAP Working Paper 300, Center for Global Trade Analysis, Department of Agricultural Economics, Purdue University, May 1999.
40. R. Olsen, *A Note on the Uniqueness of the Maximum Likelihood Estimator in the Tobit Model*, Econometrica **46** (1978), 1211–1215.

41. R. Ramanathan, *Introductory econometrics with applications*, (Harcourt College Publishers, 2002.
42. Sherman Robinson, Andrea Cattaneo, and Moataz El-Said, *Estimating a Social Accounting Matrix Using Cross Entropy Methods*, `http://www.ifpri.org/divs/tmd/dp/papers/tmdp33.pdf`, October 1998.
43. Sherman Robinson and Moataz El-Said, *Estimating a Social Accounting Matrix Using Entropy Difference Methods*, `http://www.ifpri.org/divs/tmd/dp/papers/tmdp21.pdf`, September 1997.
44. R. Schoenberg, *Constrained Maximum Likelihood*, Tech. report, Aptech and The University of Washington, November 1996.
45. C. E. Shannon, *A Mathematical Theory of Communication*, Bell System Technical Journal **27** (1948), 379–423.
46. S. M. Stigler, *The History of Statistics. The Measurement of Uncertainty before 1900*, The Belknap Press of Harvard Press, 1986.
47. H. Theil, *Principles of Econometrics*, Wiley, 1971.
48. Luke Tierney, *XLISP-STAT, A Statistical Environment Based on the XLISP Language (Version 2.0)*, Tech. Report 528, University of Minnesota, School of Statistics, July 1989.
49. J. Tobin, *Estimation of relationships for limited dependent variables*, Econometrica **26** (1958), 24–36.
50. Hal R. Varian, *Intermediate Microeconomics. A modern approach*, fifth ed., W. W. Norton & Co., 1999.
51. Diana Whistler, Kenneth J. White, S. Donna Wong, and David Bates, *SHAZAM Version 9 User's Reference Manual*, Northwest Econometrics, 2001.
52. Stephen J. Wright, *On reduced convex QP formulations of monotone LCPs*, Mathematical Programming **90** (2001), 459–473.

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