Least Squares and Maximum Likelihood Estimation: A Primer in Solving Econometrics Problems with GAMS

Thomas F. Rutherford and Huiming Wang*

Department of Agricultural and Applied Economics
University of Wisconsin-Madison

October 30, 2014

Abstract

The General Algebraic Modeling System (GAMS) has a long history of applications in operations research and computable general equilibrium modeling applications. It is at present rarely employed for econometric estimation, in part because of the lack of tools required for retrieving point estimates Hessian and Jacobian matrices. Our paper documents new tools which facilitate the use of GAMS to solve a range of econometrics problems representative of homework problems confronting graduate students in agricultural and applied economics at the University of Wisconsin, Madison. Like Kalvalagen (2007), we take a pedagogic perspective and illustrate the advantages of algebraic modeling languages with concrete applications.

1 Introduction

GAMS, the "Generalized Algebraic Modeling System", is a modeling language which was originally developed for linear, nonlinear and integer programming. One of the main objectives of GAMS is to facilitate rapid prototyping of mathematical optimization and structural equilibrium models. As the key message delivered by Kalvalagen's paper (2007) on the use of GAMS for least squares models, GAMS is an excellent tool for applied optimization. It has been less apparent, particularly for non-GAMS programmers, that GAMS can do all of the tasks involved in estimation and statistical testing, including those which rely on the extraction of Hessian and Jacobian matrices for local sensitivity analysis. This paper demonstrates methods for retrieving these matrices and solving statistical problems with GAMS.

*Corresponding author. wang@aae.wisc.edu

In this paper, we first provide and document tools built on the CONVERTD utility which provides local sensitivity analysis of optimization models. We also illustrate how these methods permit the calculation of confidence intervals and test statistics for estimation problems formulated in GAMS.

Second, we highlight the virtues of algebraic modeling languages for econometric model specification. A GAMS program provides a framework for explicit formulation of optimization problems; Furthermore, most of the logical steps in an estimation problem can be translated from MATLAB/STATA/R into GAMS, and GAMS then offers a higher level of transparency, even for readers who are not proficient in the programming rules. We show this by examples.

Third, we provide a library of worked examples through which future generations of graduate students in applied economics can more easily solve several types of classical econometric problems.

Fourth, for professionals currently using MATLAB/STATA/R, we offer code fragments which cover most of the mundane tasks involved in implementing estimation models in GAMS: reading data from different sources, calling include library routines, even simple GAMS syntax for working with datasets. GAMS tips will be included in the appendix to help beginner in understanding normal GAMS code.

As for the expected *audience* for this paper, we have three groups of people in mind: existing GAMS users who may be unaware of the tools which facilitate econometric analysis with GAMS (primarily this is CONVERTD); second, MATLAB/STATA/R programmers who may occasionally encounter estimation problems which are not readily solved with their standard tools. By providing cleanly documented "worked examples", we make it easy for non-GAMS programmers to use GAMS when needed; third, graduate students in applied economics who can benefit from the clarity and precision of using algebraic modeling tools for estimation. We are particularly thinking about students who want to do applied equilibrium analysis and would benefit from using a single programming framework for their thesis work.

Examples covered in this paper are the following:

NLS_1 Nonlinear least squares with a Cobb-Douglas production function, from Brain W. Gould, *Applied Econometrics*, Spring 2013, Homework 2_1.

NLS_2 Nonlinear least squares with a CES production function, from Gould, Homework 2_4

NLS_3 Nonlinear least squares with consumption prediction, from Gould, Homework 2_2

NLS_4 Nonlinear least squares with learning curve, from Gould, Homework 2_3

LDV_1 Poisson regression model with simulated data, from Daniel Phaneuf, Spring 2014, *Micro-Econometrics*, Homework 3_1.

LDV_2 Binary choice model in *probit* form, from Gould, Homework 4_1.

LDV_3 Binary choice model in *probit* form with heteroscedasticity, from Gould, Homework 4_2

LDV_4 Binary choice model in *logit* form, from Gould, Homework 4_3

LDV_5 Conditional *logit* model with fishing mode choice,from Gould, Homework 5_2

LDV_6 Multiple choice model of ordered *logit* form, from Gould, Homework 5_3

LDV_7 Censored data model *tobit* form, from Gould, Homework 6_1

LDV_8 Sample selection model of *heckit* form, from Gould, Homework 6_3

LDV_9 Conditional *logit* with NC beach visits, from Phaneuf, Homework 4_1.

LDV_10 Multivariate *probit* model and GHK simulator, from Phaneuf, Homework 4_2.

MLE_1 MLE of gas consumption with heteroscedasticity, from Gould, Homework 3_2.

The outline of solving these parametric examples is as follows: Starting from the mathematical problem needs to be solved, we set up a GAMS model, then, estimate the unknown parameters and report statistics for estimation and testing purposes.

Before that, we briefly introduce the usage of the Local Sensitivity Analysis (`LSA`) routine we created to derive local sensitivity values automatically.

2 Local sensitivity analysis (LSA)

In parametric estimation of economic models, it is often the case that certain statistics may only be reported if we have access to some local sensitivity values of the system of (nonlinear) equations.

Suppose we have variable $var(domain\_var)$, say $X(i,j)$, $Y(s)$ and equation $eq(domain\_eq)$, say $eq\_a(t,j)$, $eq\_b(i)$ in the model. Each of these entities may have a different multidimensional domain. `LSA` routine produces a framework in which we can extract these values in the model domain.

2.1 Jacobian/Hessian matrix

For notational simplicity, we use m and n to denote (possibly) multidimensional domain [1]$domain\_eq$ and $domain\_var$. Our goal is to derive the Jacobian matrix $J(m,n)$ and Hessian matrix $H(m,n,nn)$ (nn is an alias name of n). When the nonlinear system of equations `eq(m)` with variable `var(n)` (v_n as its short name) describes a mapping, $\mathbb{R}^n \rightarrow \mathbb{R}^m$, we call it a function F. According to the definition of Jacobian matrix, the partial derivatives of all functions $F_1, \ldots, F_m$ with respect to variables $v_1, v_2, \ldots, v_n$ construct a $m \times n$ Jacobian matrix $J(m,n)$; Similarly, based on the definition of the Hessian matrix, the matrix with all second partial derivatives of function F with respect to variables is known as a Hessian, $H(m,n,nn)$.

`LSA` is currently designed for `NLP` and `MCP` models in GAMS. By calling an external file called `lsa.gms` using the `$libinclude` syntax (in the examples we temporarily use `$batinclude` instead), we extract local sensitivity values. Here is an example with an `nlp` model:

[1]Variable and equation domain in the `lsa` routine could also be a scalar (with dimension 0).

We begin to calculate the Jacobian/Hessian matrix following the solving statement:

```
solve <model_name> using nlp maximizing <objective>;
```

Next, we call `lsa` again to calculate the Jacobain/Hessian matrix:

```
$libinclude lsa <model_name> nlp "maximizing <optimand>"
```

Notice that optimization command "maximizing <optimand>" is going to be ignored when solving an `MCP` model. In an an `MCP` type of model, we solve the model and we make the requisite change in syntax in our invocation of `lsa.gms`:

```
$libinclude lsa <model_name> mcp
```

After that, in both `NLP` and `MCP` models, Jacobian matrix $J(m,n)$ and Hessian matrix [2] $H(m,n,nn)$ are exported from the resulting data file as

```
J(m,n)    = LSA_DF(<eq_name>(m), <var_name>(n));
H(m,n,nn) = LSA_D2F(<eq_name>(m), <var_name>(n), <var_name>(nn));
```

2.2 First/second order derivatives of variables with respect to variables

`LSA` routine also facilitates extracting derivatives of variables with respect to other variables. This is very similar to what we did in deriving Jacobian and Hessian matrix, the main difference is that instead of equation `eq(m)`, we now interested in first and second order derivatives of variable `var(i)` with respect to `var(j)`. As before, i and j stand for potentially multi-dimensional domains of variables.

We list two cases in deriving these values, one for an `NLP` and the other for an `MCP` model. First, in an `NLP` type of model, we load variables after the solving statement, then call `lsa` again as follows:

```
solve <model_name> using nlp maximizing <optimand>;
*       Call lsa to derive local sensitivity values
$libinclude lsa <model_name> nlp "maximizing <optimand>"
```

First/second order derivatives of `var(i)` with respect to `var(j)`, say $g(i,j)$ and $g2(i,j,jj)$ are then reported by

[2] Currently resulting parameter `LSA_D2F` is defined as an upper triangular Hessian matrix.

```
g(i,j)     = LSA_DX(<var_name>(i), <var_name>(j));
g2(i,j,jj) = LSA_D2X(<var_name>(i), <var_name>(j), <var_name>(jj));
```

Second, as before, in an `MCP` model, we follow very similar steps as in an `NLP` model with modification in syntax:

```
solve <model_name> using mcp;
*       Call lsa to derive local sensitivity values
$libinclude lsa <model_name> mcp
```

After that, we load parameters `LSA_DX` and `LSA_D2X` in resulting data file to report corresponding derivatives.

Equipped with the `lsa` routine, in the next two sections, we are going to show several examples of how to do parametric econometrics in GAMS:

3 Non-linear least squares (NLS)

In the following non-linear least squares models, we are asked to estimate unknown coefficients, report regression statistics, and check the convexity of the objective function:

3.1 Example 1 of NLS: with a Cobb-Douglas production function

We start with a simple non-linear least squares model in which we have a Cobb-Douglas production function. *In his 1977 paper, Grayham E. Mizon estimated a variety of specifications for production functions including the Cobb-Douglas (C-D) and Constant Elasticity of Substitution (CES) where they were allowed to have additive as well as multiplicative error terms. He used U.K. data on capital, labor use and a common output measure for 24 industries encompassing the years 1954, 1957 and 1960 in his production function estimations. The following table provides a summary of the variables contained in the mizon_1977.xls dataset.*

Variable	Description	Units
Quant	Gross value-added at factor cost	Mil. Lbs
Capital	Value of the stock of plant and machinery	Mil. Lbs
LF	Labor force available for work in the industry	1000
Unemploy	Number of workers unemployed in the industry	1000
Hour	Average hours per week worked by those employed	Hours
Year	Survey data year	Year
Industry	Industry ID Number	#

Table 1: Variables Contained in the Mizon (1977) dataset

Based on dataset *Mizon* (1977), we have 72 cross sectional observations of input k_t (*Capital at time t*), l_t (*Working hours at time t*), and output q_t (*Production at time t*) data to optimally fit a Cobb-Douglas production function as follows:

$$q_t = \phi k_t^{\beta} l_t^{\alpha} + \mu_t. \tag{3.1.1}$$

We need to estimate unknown parameters in part (a), as well as report typical regression statistics as required, then in part (b) we check the convexity of the objective function to find out whether we achieve a local minimum of the objective. Here is the part (a):

(a) Assume that you would like to estimate the parameters of the above Cobb-Douglas type production function. Present the typical regression output.

3.1.1 Coefficients estimation

In a typical NLS problem, we estimate the unknown vector of parameters θ by solving the optimization model: finding the estimator $\hat{\theta}$ which minimizes the sum of squared errors:

$$\hat{\theta} = \arg_{\theta} \left[\min_{\Theta} (\sum_{t=1}^{m} \mu_t^2) \right], \tag{3.1.2}$$

$$\text{s.t.} \quad y_t = f_t(x, \theta) + \mu_t.$$

Then, the non-linear prediction of dependent variable is $\hat{y_t} = f_t(x, \hat{\theta})$.

To estimate coefficients, we model this production function in GAMS, solve the optimization problem 3.1.2 as a *Quadratically Constrained Program* (QCP) type of model.

```
$title GAMS Solution of an NLS problem
*       Estimate parameters in a QCP type of model
set     t                "Set of observations" /t1*t72/,
        i                "Parameter values"    /"phi","beta","alpha"/;

alias (i,ii);

parameter              data(t,*)              "Source data";

$call gdxxrw mizon_1977_gams.xlsx par=data rng=A1:H73 cdim=1 rdim=1 checkDate
$gdxin mizon_1977_gams.gdx
$loaddc data
$gdxin

parameter               q(t)                  "Production",
                        l(t)                  "Labor (unemployment adjusted)",
                        k(t)                  "Capital",
                        labor(t)              "Total labor",
                        unemp(t)              "Unemployment";
```

```
*       Read data
q(t)    = data(t,"Quant");
k(t)    = data(t,"Capital");
labor(t) = data(t,"Labor");
unemp(t) = data(t,"Unemploy");
l(t)    = (labor(t) - unemp(t))*data(t,"Hour") / 100;

variable                EPSILON(t)              "Error terms",
                        SSE                     "Sum of squared errors",
                        THETA(i)                "Unknown parameters";

equation                fit(t)                  "Cobb-Douglas model",
                        obj                     "Objective";

obj..           SSE =e= sum(t,sqr(EPSILON(t)));
fit(t)..        q(t) =e= THETA("phi")*k(t)**THETA("beta")*l(t)**THETA("alpha") + EPSILON(t);

*       Set bounds of variables
THETA.LO(i)   = 0.01;
THETA.UP(i)   = 10;
EPSILON.LO(t) = -1000;
EPSILON.UP(t) = 1000;

model CD /obj,fit/;
solve CD minimizing SSE using nlp;
```

As a result, we find the following estimates (where ϕ is the scale parameter, β the income share of capital and α the income share of labor):

```
----     46 VARIABLE SSE.L                =   432328.546  Sum of squared errors

----     46 VARIABLE THETA.L                              Unknown parameters

phi 1.304,    beta  0.222,    alpha 0.829
```

3.1.2 Derive regression statistics

Next, we are usually asked to report typical regression statistics, say, *standard errors*, *eigenvalues*, *P-value* and *T-value* [3] based on some statistical hypothesis. We introduce how to do that in GAMS with the help of the `lsa` routine.

Standard econometrics textbooks tell us that in an NLS model, the estimated *covariance matrix of the coefficients* $\hat{V}_{\hat{\theta}}$ (cov in the following code) is calculated as

$$\hat{V}_{\hat{\theta}} = \hat{\sigma}^2 (J^T J)^{-1}, \tag{3.1.3}$$

in which $\hat{\sigma}^2$ is the estimated *variance of residuals* while J is the Jacobian matrix whose $(t,i)^{th}$ entry is $\frac{\partial \mu_t}{\partial \theta_i}$. In general, by definition, estimated *variance of residuals* $\hat{\sigma}^2$ is the sum of squared errors divided by the *degrees of freedom* (df):

$$\hat{\sigma}^2 = \widehat{SSE}/df,$$

[3] Since our parameter of interest θ_0 is usually 0 as we want to test whether the coefficient differs from 0 or not, we sometimes simply use *P-value* or *T-value* instead of the more complete phase such as "the *P-value* with $\theta_0 = 0$" or "the *T-value* with $\theta_0 = 0$".

where $\widehat{SSE} = \sum_t \hat{\mu_t}^2$, $df =$ (Number of observations − Number of unknowns).

On the other hand, utilizing the `lsa` routine, we can either derive Jacobian and Hessian matrices of function `fit(t)` with respect to unknown variables `THETA(i)`, or find the first and second derivatives of residual terms `MU(t)` subject to unknowns. It is obvious that these two sets of local sensitivity values are exactly the same, since each residual term `MU(t)` is defined in a corresponding fit function `fit(t)`.

Furthermore, having access to Jacobian and Hessian matrices of residual term `MU(t)` is crucial in finding other statistics. For example, Eigenvalues of the Hessian matrix of the SSE with respect to unknown coefficients in GAMS is derived through the *eigenvector* utility, while Hessian matrix of SSE itself could be calculated from Jacobian and Hessian matrix of the error term μ_t based on the chain rule as follows (j as an alias name of i):

$$\frac{\partial^2 SSE}{\partial\theta_i\partial\theta_j} = \frac{\partial^2 \sum_t \mu_t^2}{\partial\theta_i\partial\theta_j} = \sum_t 2(\mu_t \frac{\partial^2\mu_t}{\partial\theta_i\partial\theta_j} + \frac{\partial\mu_t}{\partial\theta_i}\frac{\partial\mu_t}{\partial\theta_j}).$$

Local sensitivity values are also essential in statistical testing. In order to test the statistical significance of some coefficients, we routinely check the *T-value* and *p-value* according to null hypothesis. Based on coefficient estimate $\hat{\theta_i}$ and its standard error estimation $\hat{se}(\hat{\theta_i})$, we define the *T-value* of obtaining this estimation instead of a pre-specified null hypothesis $\mathbb{H}_0 : \hat{\theta_i} = \theta_0$ as

$$(\hat{\theta_i} - \theta_0)/\hat{se}(\hat{\theta_i}),$$

and the corresponding *P-value* for the null against a "two-sided" alternative $\mathbb{H}_1 : \hat{\theta_i} \neq \theta_0$ is

$$2(1 - \phi(t_i)),$$

where $\phi(\cdot)$ is the cumulative *Student's t-distribution* function with *degrees of freedom* df. Large *T-value* or small *p-value* tend to indicate that the data set we are testing is significantly different from what we assumed to have in the null hypothesis.

The following is an example of GAMS code of deriving these regression statistics:

```
*       Intermediate parameters in reporting statistics
parameter               jacobian(t,i)           "Jacobian of constraints fit(t) wrt parameters THETA(i)",
                        hessian(t,i,ii)         "Hessian of constraints wrt parameters",
                        jacsqr(i,ii)            "Squared Jacobian matrix",
                        jsinv(i,ii)             "Inverse of jacsqr",
                        ssehess(i,ii)           "Hessian of the sum of squared errors",
                        hinv(i,ii)              "Inverse of ssehess",
                        eigenvalues(i)          "Eigenvalues of ssehess",
                        eigenvec(i,i)           "Eigenvectors of ssehess",
                        sigma_hat               "Unbiased estimator for error variance",
                        cov(i,ii)               "Covariance matrix of coefficients";

*        Report Hessian and Jacobian matrix
$batinclude lsa CD nlp "minimizing SSE"

jacobian(t,i)     = LSA_DF( fit(t), THETA(i) );
hessian(t,i,ii)   = LSA_D2F( fit(t), THETA(i), THETA(ii) );

*        d2f is an upper triangular matrix
```

```
hessian(t,i,ii) = hessian(t,i,ii)$(ord(ii) ge ord(i)) + hessian(t,ii,i)$(ord(ii) lt ord(i));

*       Find squared Jacobian matrix
jacsqr(i,ii)  = sum(t, jacobian(t,i)*jacobian(t,ii));

*       Find Hessian matrix with respect to the sum of squared errors
ssehess(i,ii) = sum(t, 2*jacobian(t,i)*jacobian(t,ii) + 2*EPSILON.L(t)*hessian(t,i,ii));

*       Find the inverse of squared Jacobian matrix
$batinclude arealinverter jacsqr i ii jsinv

*       Find the inverse of Hessian matrix of SSE
$batinclude arealinverter ssehess i ii hinv

*       Find the eigenvalues of the Hessian matrix of SSE
$batinclude eigenvector i ssehess eigenvalues eigenvec

*       Find the unbiased estimator for error variance
sigma_hat = SSE.L / (card(t) - card(i));

*       Find the estimated covariance matrix of coefficients
cov(i,ii) = jsinv(i,ii)*sigma_hat;

*       Report estimation statistics
parameter               statistics(i,*)         "Statistics at the point";
statistics(i,"estimator")  = THETA.L(i);
statistics(i,"std error")  = sqrt(cov(i,i));
statistics(i,"Eigenvalue") = eigenvalues(i);
statistics(i,"T value")    = THETA.L(i)/sqrt(cov(i,i));

*       Use the incomplete beta function
*       To find the CDF of student's t-distribution function:
statistics(i,"P value")    = BETAREG( (card(t) - card(i))/
                                      (card(t) - card(i) + sqr(statistics(i,"T value"))),
                                      (card(t) - card(i))/2, 0.5 );

display statistics;
```

Finally, GAMS reports statistics as follows:

```
----    287 PARAMETER statistics  Statistics at the point
        estimator    std error  Eigenvalue     T value     P value

phi      1.304028    0.263893 1.749827E+5    4.941505    0.000005
beta     0.221503    0.031226 8.258437E+6    7.093447 9.00476E-10
alpha    0.828797    0.031889 2.669965E+9   25.990221 2.30446E-37
```

(b) At the optimal coefficient estimates check the convexity of the SSE function to determine whether you are at least at a local minimum. Again this should be written so that it can be applied to a model with any number of coefficients.

If all the *eigenvalues* of the Hessian matrix of sum of squared errors (SSE) with respect to coefficients are non-negative, at least we are at a local minimum. For this question, the answer is yes, we are at a local minimum with the optimal coefficient estimates given the all positive eigenvalues found in the statistics report above.

In the next example, we will introduce more application of the `lsa`:

3.2 Example 2 of NLS: with a CES production function

In this example, we first use the *Mizon* (1977) data accessed in Example 1 to estimate parameters of the following Constant Elasticity of Substitution (CES) production functions having both additive 3.2.1 and

multiplicative 3.2.2 error terms.

CES:

Multiplicative Disturbance (MD)

$$q_t = \phi[\delta k_t^{\rho} + (1-\delta) l_t^{\rho}]^{\frac{\kappa}{\rho}} \exp(\mu_t) \tag{3.2.1}$$

Additive Disturbance (AD)

$$q_t = \phi[\delta k_t^{\rho} + (1-\delta) l_t^{\rho}]^{\frac{\kappa}{\rho}} + \mu_t \tag{3.2.2}$$

where:

$\phi \equiv$ scale parameter, $\phi > 0$

$\kappa \equiv$ the degree of homogeneity, $\kappa > 0$

$\delta \equiv$ distribution parameter, $0 < \delta < 1$

$\rho \equiv$ the substitution parameter, $-\infty < \rho < 1, \rho \neq 0$

After that, in part (b), we are asked to find the correlation between predicted and actual values of quantity q_t; In part (c), we test whether the production technology exhibits constant return to scale in both MD and AD models; In part (d) and (e), we focus on the multiplicative distrubance (MD) model: we test whether the marginal products of capital and labor (MPK and MPL) evaluated at data mean are positive (part (d)), and whether the intercept varies across survey years (1957 and 1960) in part (e):

(a) Using the Mizon data set, estimate 3.2.1 and 3.2.2 via NLS. Present the typical regression statistics. Is the SSE function indeed at a (local) minimum at the parameter values you identified as minimizing the SSE function?

Comparing to the previous example, the production function in this exercise takes a more general CES form (Cobb-Douglas production function is an extreme case of CES when $\rho = 0$).

Thus the solving strategy of this part is: In each model (MD or AD), first divide the domain of the substitution parameter ρ into two intervals $I_j, j = 1, 2$, where $I_1 = (-\infty, 0)$ and $I_2 = (0, 1)$, then consider optimization problem

$$\hat{\theta}|_{\rho \in I_j} = \arg_{\theta} \left[\min_{\Theta} (\sum_{t=1}^{T} \mu_t^2) | \rho \in I_j \right],$$

$$\text{s.t.} \quad \text{constraint (3.2.1) or (3.2.2).}$$

After that, we compare the optimal SSE in these two cases to pick the optimal set of parameter estimates in each model using a *savepoint* option, details are included in the GAMS tips in the Appendix.

The following code provides a step by step example for this:

```
$title Constant Elasticity of Substitution (CES) production functions
*       Two forms of CES production functions

*       Multiplicative Disturbance (MD model):
```

```
*                       q(t) = phi*(delta*k(t)**rho + (1-delta)*l(t)**rho)**(kappa/rho)*exp(mu(t))
*        Additive Disturbance (AD model):
*                       q(t) = phi*(delta*k(t)**rho + (1-delta)*l(t)**rho)**(kappa/rho) + mu(t)

*       Part(a)
*       Question:  Estimate unknown coefficents in CES production function
set     t       "Set of observations"   /t1*t72/,
        m       "Index of parameters to be estimated"   /"delta","rho","kappa","phi","dum57","dum60"/,
        i(m)    "Subset of m, index of parameters"      /"delta","rho","kappa","phi"/,
        s       "Model type"            /"MD","AD"/;

alias (i,ii), (m,mm), (t,tt);

*       Load data from an excel file
parameter       data(t,*)       "Source data";
$call gdxxrw mizon_1977_gams.xlsx par=data rng=A1:H73 cdim=1 rdim=1 checkDate
$gdxin mizon_1977_gams.gdx
$loaddc data
$gdxin

*       Define explanatory variables
parameter       q(t)            "Production",
                k(t)            "Capital",
                l(t)            "Labor (unemployment adjusted)",
                labor(t)        "Total labor",
                unemp(t)        "Unemployment",
                year(t)         "Survey year";

*       Introduce data
q(t)     = data(t,"Quant");
k(t)     = data(t,"Capital");
labor(t) = data(t,"Labor");
unemp(t) = data(t,"Unemploy");
l(t)     = (labor(t) - unemp(t))*data(t,"Hour")/100;
year(t)  = data(t,"year");
```

We then set up the multiplicative disturbance model, note that we need to take the log on both sizes of equation (3.2.1) to make least squares method suitable for this problem.

```
*       Multiplicative Disturbance model (MD)
variable        THETA(m)        "Unknown parameters",
                MU(t)           "Error terms",
                SSE             "Sum of squared errors";

equation        fit_md(t)       "CES model in the MD model",
                obj             "Objective in the MD model";

obj..           SSE =e= sum(t,sqr(MU(t)));

fit_md(t)..     log(q(t)) =e= log(THETA("phi")) + (THETA("kappa")/THETA("rho"))*
                              log(THETA("delta")*k(t)**THETA("rho") +
                              (1-THETA("delta"))*l(t)**THETA("rho")) + MU(t);
```

Next, we solve the MD model:

```
*       Set bounds on "phi","kappa" and "delta"
THETA.LO("phi")   = 0.01;
THETA.UP("phi")   = inf;
THETA.LO("kappa") = 0.01;
THETA.UP("kappa") = inf;
THETA.LO("delta") = 0.01;
THETA.UP("delta") = 0.99;

MU.LO(t) = -1000;
MU.UP(t) = 1000;

model CES_MD /obj, fit_md/;

*       Optimize when rho is between 0 and 1 first
```

```
THETA.LO("rho")  = 0.01;
THETA.UP("rho")  = 0.99;
THETA.L ("rho")  = 0.5;
```

After solving this conditional optimization given $0 < \rho < 1$, use *savpoint* to save temporary results in the data file `ces_md_p.gdx` for future comparison;

```
*       Save current results in "ces_md_p.gdx"
CES_MD.SAVEPOINT = 1;
solve CES_MD minimzing SSE using nlp;
CES_MD.SAVEPOINT = 0;

parameter        sse_value        "Sum of squares";
sse_value = SSE.L;

*       Solve the problem again when rho is negative
THETA.lo("rho")   = -inf;
THETA.up("rho")   = -0.01;
THETA.l ("rho")   = -0.5;

solve CES_MD minimzing SSE using nlp;

*       If the solution value is made worse,
*       load the previously computed solution:
if (SSE.L > sse_value, execute_loadpoint 'ces_md_p.gdx';);

*       Record results for future use:
parameter               coef(s,m)              "Parameter i in model s"
                        para_md(m)             "Parameter i in the md model";
coef("MD",i) = THETA.L(i);
para_md(i)   = THETA.L(i);
```

Finally, we report typical regression statistics. The whole process is similar to its counterpart of the previous example. Jacobian matrix jacobian(t,m) and the Hessian matrix hessian(t,m,mm) are reported here:

```
*       Define intermediate parameters
parameter
                jacobian(t,m)                  "Jacobian of constraints wrt parameters",
                hessian(t,m,mm)                "Hessian of constraints wrt parameters",
                jacsqr(m,mm)                   "Squared Jacobian matrix",
                jsinv(m,mm)                    "Inverse of jacsqr",
                ssehess(m,mm)                  "Hessian of the sum of squared errors",
                eigenvalues(m)                 "Eigenvalues of ssehess",
                eigenvec(m,m)                  "Eigenvectors of ssehess",
                sigma2_hat                     "Unbiased estimator for error variance",
                cov(m,mm)                      "Covariance matrix",
                statistics(m,*)                "Statistics at the point";

*       Parameters for the current models only
*       Report std, T value and p value against H0, r_square, est. of unbiaed cov and eigenvalues of model type s
parameter       stat(s,m,*)                    "Statistics at the point for the model type s",
                sig2_hat(s)                    "Unbiased estimator of error variance in model type s",
                r_2(s)                         "Coefficient of determination in model type s",
                cov_hat(s,m,mm)                "Variance covariance matrix in model type s";

*        Report Hessian and Jacobian matrix
$batinclude lsa CES_MD nlp "minimizing SSE"

jacobian(t,i)     = LSA_DF( fitmd(t), THETA(i) );
hessian(t,i,ii)   = LSA_D2F( fitmd(t), THETA(i), THETA(ii) );

*        d2f is an upper triangular matrix
hessian(t,i,ii) = hessian(t,i,ii)$(ord(ii) ge ord(i)) + hessian(t,ii,i)$(ord(ii) lt ord(i));

*       Derive covariance matrix, p_value and T_value
```

```
*       Then send these lines of command to an external file called stats.gms
$onecho >stats.gms
*        Find squared Jacobian matrix
jacsqr(i,ii)  = sum(t, jacobian(t,i)*jacobian(t,ii));

*       Find the inverse matrix of jacsqr
$batinclude arealinverter jacsqr i ii jsinv

*        Find the hessian of the sum of squared errors
ssehess(i,ii) = sum(t,2*jacobian(t,i)*jacobian(t,ii) + 2*MU.L(t)*hessian(t,i,ii));

*       Find the eigenvalues of the Hessian matrix of SSE
$batinclude eigenvector i ssehess eigenvalues eigenvec

*       Degrees of freedom = card(t) - card(i)
*       Unbiased estimator of error variance
sigma2_hat = SSE.L / (card(t) - card(i));
cov(i,ii) = jsinv(i,ii)*sigma2_hat;

*       Report statistics
statistics(i,"estimator")  = THETA.L(i);
statistics(i,"std error")  = sqrt(cov(i,i));
statistics(i,"eigenvalue") = eigenvalues(i);
statistics(i,"T value")    = THETA.L(i)/sqrt(cov(i,i));
*       Use the BETAREG function:
statistics(i,"P value")    = BETAREG( (card(t) - card(i))/(card(t) - card(i) + sqr(statistics(i,"T value"))),
                                      (card(t) - card(i))/2, 0.5 );
$offecho

*       Call the external file stats.gms
$include stats

*       Report statistics of model type s
stat("MD",i,"estimator")  = statistics(i,"estimator");
stat("MD",i,"std error")  = statistics(i,"std error");
stat("MD",i,"eigenvalue") = statistics(i,"eigenvalue");
stat("MD",i,"T value")    = statistics(i,"T value");
stat("MD",i,"P value")    = statistics(i,"P value");

*       Coefficient of determination
r_2("MD")           = 1 - SSE.L/(sum(t, sqr(log(q(t)))) - card(t)*sqr(sum(t, log(q(t)))/card(t)));
sig2_hat("MD")      = sigma2_hat;

*       Record covariance matrix of coefficients for future use
cov_hat("MD",i,ii)  = cov(i,ii);

display stat, r_2;
```

Next, we begin to solve additive disturbance (AD) model in an analogous fashion except that we don't have to take the logs on equation (3.2.2).

```
*       Additive Disturbance (AD)
equations            fitad(t)      "CES_AD model";
fitad(t)..           q(t) =e= THETA("phi")*(THETA("delta")*k(t)**THETA("rho") +
                              (1-THETA("delta"))*l(t)**THETA("rho"))**
                              (THETA("kappa")/THETA("rho")) + MU(t);

*       Setting bounds on "phi","kappa","delta"
THETA.LO("phi")   = 0.01;
THETA.UP("phi")   = inf;
THETA.LO("kappa") = 0.01;
THETA.UP("kappa") = inf;
THETA.LO("delta") = 0.01;
THETA.UP("delta") = 0.99;

MODEL CES_AD /obj,fitad/;

*       Optimize over rho when it is negative
THETA.LO("rho") = -inf;
THETA.UP("rho") = -0.01;
THETA.L("rho")  = -0.5;

*       Record the initial objective value SSE using "savepoint"
CES_AD.SAVEPOINT = 1;
```

```
solve CES_AD minimzing SSE using nlp;
CES_AD.SAVEPOINT = 0;

*       Put current value of SSE into parameter sse_value
sse_value = SSE.L;

*       First bound "rho" between 0 and 1
THETA.LO("rho") = 0.01;
THETA.UP("rho") = 0.99;
THETA.L("rho")  = 0.5;

*       Solve the problem again then compare with the saved value
solve CES_AD minimzing SSE using nlp;

*       If the solution value is made worse,
*       then load the previously computed solution:
if (SSE.L > sse_value,execute_loadpoint 'ces_ad_p.gdx';);

*       In this case, the second step generates a smaller SSE,
*       then we don't need to load the previous solution

*       Record parameters of AD model for future use
coef("AD",i) = THETA.L(i);

*        Report Hessian and Jacobian matrix
$batinclude lsa CES_AD nlp "minimizing SSE"

jacobian(t,i)     = LSA_DF( fitad(t), THETA(i) );
hessian(t,i,ii)   = LSA_D2F( fitad(t), THETA(i), THETA(ii) );

*        d2f is an upper triangular matrix
hessian(t,i,ii) = hessian(t,i,ii)$(ord(ii) ge ord(i)) + hessian(t,ii,i)$(ord(ii) lt ord(i));

*       Report standard statistics
*       Include the file we defined earlier which calculats statistics
$include stats

*       Report more statistics for the AD model
sig2_hat("AD")      = sigma2_hat;
r_2("AD")           = 1 - SSE.L/(sum(t, sqr(q(t))) - card(t)*sqr(sum(t, q(t))/card(t)));
cov_hat("AD",i,ii)  = cov(i,ii);

*       Generate a report of statistics
stat("AD",i,"estimator")  = statistics(i,"estimator");
stat("AD",i,"std error")  = statistics(i,"std error");
stat("AD",i,"eigenvalue") = statistics(i,"eigenvalue");
stat("AD",i,"T value")    = statistics(i,"T value");
stat("AD",i,"P value")    = statistics(i,"P value");

display stat,r_2;
```

Here is the list of regression statistics reported from GAMS. Note that all eigenvalues are positive, which means the objective function (SSE) is indeed at a local minimum at the parameter values.

```
----    902 PARAMETER stat  Statistics at the point for the model type s
           estimator   std error  eigenvalue     T value     P value
MD.delta    0.275755    0.034422    1.057945    8.011059 2.09230E-11
MD.rho     -0.356006    0.215920    2.405353   -1.648788    0.103804
MD.kappa    0.992104    0.029559  131.188566   33.563791 4.75412E-44
MD.phi      1.811073    0.277976 3610.099633    6.515218 1.047672E-8
AD.delta    0.201801    0.029012 1.695677E+5    6.955790 1.703778E-9
AD.rho      0.161613    0.182729 4.551763E+5    0.884440    0.379575
AD.kappa    1.057569    0.033645 2.220277E+7   31.432778 3.15191E-42
AD.phi      1.235968    0.261219 1.367622E+9    4.731537    0.000012
```

(b) For the CES model specifications what is the correlation between predicted and actual values of Quant (i.e., not $\ln(Quant)$*)? Why would one be interested in determining such correlations? (Note: When evaluating the relationship between predicted and actual values of Quant (versus* $\ln(Quantity)$*), you should note that* $E(\mu_t) = 0$

but $E[exp(\mu_t)]$ need not equal 1 but in fact $E[\exp(\mu_t)] = \exp(\sigma^2/2)$). From these results what specification would you say is preferred in terms of explaining the variance of Quant?

In this part, we first define predicted production, then write down the correlation coefficient matrix between actual output and its prediction:

Given CES model equation (3.2.1) and (3.2.2), we evaluate predicted production q_hat at optimal coefficient estimates, when $E(\mu) = 0$ and $E[exp(\mu)] = exp\left[\frac{\hat{\sigma^2}}{2}\right]$:

```
*       Part(b)
*       Note: E(mu) = 0, E(exp(mu)) = exp(sig2_hat/2)

*       Define predicted production
parameter       q_hat(s,t)              "Prediction of production from model s";

*       Plug E(mu) and E(exp(mu)) into the fitted CES functions
q_hat("MD",t) = (coef("MD","phi")*(coef("MD","delta")*k(t)**(-coef("MD","rho")) +
                (1-coef("MD","delta"))*l(t)**(-coef("MD","rho")))**(-coef("MD","kappa")/coef("MD","rho"))) *
                exp(sig2_hat("MD")/2);

q_hat("AD",t) = coef("AD","phi")*(coef("AD","delta")*k(t)**(-coef("AD","rho")) +
                (1-coef("AD","delta"))*l(t)**(-coef("AD","rho")))**(-coef("AD","kappa")/coef("AD","rho"));
```

Next, according to the definition of correlation coefficients , this measure ($\rho_{X,Y}$) of the linear correlation (dependence) between two variables X and Y is define as:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}.$$

We have the following code to find correlation coefficients (cc_q) between Quantity and prediction of Quantity in both MD and AD models:

```
*       Find correlation coefficients between actual output q and model prediction of q

*       Correlation Coefficient matrix ccoef(s,u,v) is a 2*2 matrix for each model type s
*       Thus, we need to find the variance-covariance matrix between q and predition of q first

set             u                       "Index for q and prediction of q"       /q,"q_hat"/;
alias   (u,v);

parameter       varQ                    "Estimated variance of q",
                var_hat(s)              "Estimated variance of predicted q in model type s",
                cov_q(s)                "Covariance between q and predicted q in model type s",
                cc_q(s,u,v)             "Correlation coefficients between q and predition of q in model type s",
                qbar(s)                 "Average of predicted q in model type s";

*       Find average of predicted quantities
qbar(s) = sum(t, q_hat(s,t))/card(t);

*       Find variances
varQ       = sum(t, sqr(q(t) - sum(tt, q(tt))/card(t)))/(card(t) - 1);
var_hat(s) = sum(t, sqr(q_hat(s,t) - qbar(s)))/(card(t) - 1);

*       Find covariances between quantities and predicted quantities
cov_q(s)   = sum(t, (q(t) - sum(tt,q(tt))/card(t))*(q_hat(s,t) - qbar(s)))/(card(t) - 1);

*       Calculate correlation coefficients
cc_q(s,u,v)= (cov_q(s)/sqrt(varQ)/sqrt(var_hat(s)))$(not sameas(u,v)) + 1$(sameas(u,v));

display cc_q;
```

We can see from the following output that the correlation coefficient is higher for the Additive Disturbance case, thus AD model is probably more suitable for explaining the variance of Quant.

```
----    950 PARAMETER cc_q  Correlation coefficients between q and predition of q in model type s

                  q       q_hat

MD.q          1.000000    0.964297
MD.q_hat      0.964297    1.000000
AD.q          1.000000    0.974625
AD.q_hat      0.974625    1.000000
```

(c) Provide statistical evidence as to whether the production technology exhibits constant returns to scale using specification (3.2.1) and (3.2.2).

We use *Wald test* to check whether the null hypothesis that the production technology exhibits constant returns to scale, *i.e.*, $\mathbb{H}_0 : \kappa = 1$ is true.

In general, let θ be the unknown vector, $\hat{\theta}$ its estimate and $\hat{V}_{\hat{\theta}}$ its estimated covariance matrix. Under some regularity conditions, the so called *Wald statistic*

$$W_n(\theta) = n(\hat{\theta} - \theta)' \hat{V}_{\hat{\theta}}^{-1} (\hat{\theta} - \theta)$$

converges in distribution to a *Chi-squared* random variable , as the number of observation n large enough, i.e.:

$$W_n(\theta) \xrightarrow{\text{d}} \chi_q^2,$$

where q is the *degrees of freedom* of the *Chi-squared* distributed random variable as well as the dimension of the unknown vector θ.

```
*       Part(c)
*       Test H0: kappa =  1 in both models
set             c                       "Row index of restriction coefficients"         /r/;
alias (c,cc);

parameter       r(c,i)                  "Restriction coefficients in testing H0: kappa = 1",
                w_kappa(s)              "Wald test statistics based on H0",
                pval_w_kappa(s)         "P value based on H0",
                alpha                   "Type 1 error tolerance level"                  /0.05/,
                rs_cov(s,c,cc)          "Denominator of Wald test statistic: r*cov*r in model s",
                df_kappa                "Degrees of freedom in Wald test";

*       Define degrees of freedom of the chi-square variable
*       We test on coefficient kappa along, so degrees of freedom = 1
df_kappa = 1;

*       Choose the coefficient of interest
table r(c,i)
      phi    kappa    delta  rho
r     0      1        0      0;

*       Adjusted covariance
rs_cov(s,c,cc) = sum((i,ii), r(c,i)*cov_hat(s,i,ii)*r(cc,ii));

*       Wald test statistics against H0: kappa = 1
w_kappa(s) = sum((i,c,cc,ii), r(c,i)*(coef(s,i) - 1)*(1/rs_cov(s,c,cc))*r(cc,ii)*(coef(s,ii) - 1));

*       P value against H0: kappa = 1
pval_w_kappa(s) = 1 - gammareg(w_kappa(s)/2,df_kappa/2);

parameter       wald_test(s,*,*)        "Wald Test Results";
```

```
loop( s$(pval_w_kappa(s) lt alpha),
         wald_test(s,"Reject H_0","P value") = pval_w_kappa(s);
         wald_test(s,"Reject H_0","Wald") = w_kappa(s); );

loop( s$(pval_w_kappa(s) ge alpha),
         wald_test(s,"Fail to Reject H_0","P value") = pval_w_kappa(s);
         wald_test(s,"Fail to Reject H_0","Wald") = w_kappa(s); );

option wald_test:3:2:1;

display "Wald Test:",wald_test;
```

Wald test shows that we don't have enough evidence to reject the hypothesis that the production technology exhibits constant returns to scale ($\kappa = 1$) in both MD and AD models:

```
----   1086 PARAMETER wald_test  Wald Test Results

                          P value        Wald

MD.Fail to Reject H_0       0.789       0.071
AD.Fail to Reject H_0       0.087       2.928
```

(d) Under the CES specification in (3.2.1) evaluate the marginal products of Capital and Labor when these inputs are at their mean values. Are these marginal products positive from a statistical point of view?

We first evaluate the marginal products of capital when capital and labor are at their mean values.

```
*        Part (d)
parameter       kbar                "Average of k(t)",
                lbar                "Average of l(t)",
                var_mpk             "Variance of marginal product of capital (MPK)",
                t_mpk               "T value of MPK against H0: MPK = 0",
                p_mpk               "P value of MPK against H0: MPK = 0",
                var_mpl             "Variance of marginal product of labor (MPL)",
                t_mpl               "T value of MPL against H0: MPL = 0",
                p_mpl               "P value of MPL against H0: MPL = 0",
                df_mpl              "Degrees of freedom in the t test against H0: MPL = 0",
                df_mpk              "Degrees of freedom in the t test against H0: MPK = 0";

*       Define average of capital and labor
kbar = sum(t, k(t))/card(t);
lbar = sum(t, l(t))/card(t);

*        Test whether marginal product of capital is zero
variable        MPK_MD              "Marginal product of capital in the MD model";
equation        def_mpk             "Definition of marignal product of capital";

def_mpk..       MPK_MD =e= THETA("kappa")*THETA("phi")**(-THETA("rho")/THETA("kappa"))*
                           THETA("delta")*kbar**(-(1 + THETA("rho")))*
                           (THETA("phi")*(THETA("delta")*lbar**(-THETA("rho")) +
                           (1-THETA("delta"))*kbar**(-THETA("rho")))**
                           (-THETA("kappa")/THETA("rho"))*exp(sig2_hat("MD")/2))**
                           (1 + THETA("rho")/THETA("kappa"));

model mp_k /def_mpk/;
```

In order to test whether MPK is statistically positive, we need gradients of the marginal product of capital, MPK_MD with respect to coefficients, evaluated at point estimates' level. The following code achieves this using `lsa` routine in an `MCP` model, and calculates the *T-test* statistic based on $\mathbb{H}_0 : MPK = 0$:

```
*       Evaluate gradients at the point estimates of coefficients
THETA.FX(i) = para_md(i);
solve mp_k using mcp;

*        Report Jacobian matrix
parameter        grad_k(i)        "Gradient of MPK wrt parameters";

$batinclude lsa mp_k mcp

*       Export Jacobian matrix from resulting data file
grad_k(i) = LSA_DF( def_mpk, THETA(i) );

*       Find adjusted covariance, report T-value and P-value
var_mpk = sum((i,ii), grad_k(i)*cov_hat("MD",i,ii)*grad_k(ii));
t_mpk   = MPK_MD.L/sqrt(var_mpk);
df_mpk  = card(i);
*       Use betareg function
p_mpk   = betareg( df_mpk/(df_mpk+sqr(t_mpk)), df_mpk/2, 0.5 );
display t_mpk,p_mpk;

if ((p_mpk lt alpha), display "Reject H_0: MPK = 0";
            else      display "Fail to reject H_0: MPK = 0"; );
```

Output shows that statistical evidence is enough to reject the null hypothesis $\mathbb{H}_0 : MPK = 0$ with confidence level $\alpha = 0.05$:

```
----   1184 PARAMETER t_mpk               =     6.652450  T value of MPK against H0: MPK = 0
            PARAMETER p_mpk               =     0.002651  P value of MPK against H0: MPK = 0
----   1186 Reject H_0: MPK = 0
```

We then apply the same method to test whether marginal product of labor is zero, $MPL = 0$:

```
*        Test whether marginal product of labor is zero
variable        MPL_MD          "Marginal product of labor in the MD model";

equation        def_mpl         "Definition of marignal product of labor";

def_mpl..        MPL_MD =e= THETA("kappa")*THETA("phi")**(-THETA("rho")/THETA("kappa"))*
                           (1 - THETA("delta"))*lbar**(-(1 + THETA("rho")))*
                           (THETA("phi")*(THETA("delta")*lbar**(-THETA("rho")) +
                           (1-THETA("delta"))*kbar**(-THETA("rho")))**(-THETA("kappa")/THETA("rho"))*
                           exp(sig2_hat("MD")/2))**(1 + THETA("rho")/THETA("kappa"));

model mp_l /def_mpl/;
solve mp_l using mcp;

*        Report gradients wrt coefficinets
parameter        grad_l(i)        "Gradient of MPL wrt parameters";

$batinclude lsa mp_l mcp

*       Exporting Jacobian matrix from the resulting data file
grad_l(i) = LSA_DF( def_mpl, THETA(i) );

*       Find adjusted covariance
var_mpl = sum((i,ii), grad_l(i)*cov_hat("MD",i,ii)*grad_l(ii));
t_mpl   = MPL_MD.L/sqrt(var_mpl);
df_mpl  = card(i);
*       Use betareg function
p_mpl   = betareg( df_mpl/(df_mpl+sqr(t_mpl)), df_mpl/2, 0.5 );

display p_mpl, t_mpl;

if ((p_mpl lt alpha), display "Reject H_0: MPL = 0";
                else  display "Fail to reject H_0: MPL = 0";);
```

It turns out that we have enough evidence statistically to claim that MPL is different from zero, *i.e.*, positive.

```
----   1322 PARAMETER t_mpl              =     9.061673  T value of MPL against H0: MPL = 0
            PARAMETER p_mpl              =     0.000822  P value of MPL against H0: MPL = 0
----   1324 Reject H_0: MPL = 0
```

4 Limited dependent variable (LDV)

Next, we are going to discuss another type of econometrics model called "limited dependent variable" (LDV) model. A "limited dependent variable" y is one which takes a "limited" set of values. For example, a binary dependent variables mode in which dependent variable $y \in \{0, 1\}$, a multinomial model in which $y \in \{0, 1, 2, \ldots, k\}$, a count data model whose dependent variables $y \in \{0, 1, 2, \ldots\}$, and a censored model in which $y \in \mathbb{R}^+$.

We first use a count data model to introduce some general ideas about maximum likelihood estimation (MLE) method usually used in estimating LDV models and several common definitions of covariance matrices here:

4.1 Count data

Limited dependent variables in a count data model can take on nonnegative integer values, $y \in \{0, 1, 2, \ldots, \}$, a typical way is to employ Poisson regression. We start with this model for the reason that if comparing to binary/multiple choice model in which we have to impose a functional form on choice probabilities, count data model with Poisson regression is more straightforward.

4.1.1 Limited Dependent Variable model example 1: Poisson regression model

Consider a Poisson regression with a conditional mean in which the expected value of the distribution is parameterized with exogenous variables:

$$Y_i \sim Poisson(\lambda_i),$$

where $\lambda_i = \exp(\beta' X_i)$, X_i *is a* $k \times 1$ *vector of explanatory variables and* β *is a* $k \times 1$ *vector of unknown parameters to be estimated.*

Consider a Monte Carlo experiment using this data generating process. In particular, write code to generate a dataset with the following characteristics: 1. Number of observation is 200; 2. $\beta = [1.5, -0.5]'$*; 3.* X_i *contains a constant and an exogenous variable generated from a standard uniform distribution; 4.* y_i *is drawn from a Poisson random number generator.*

Using the simulated data, estimate this count data model via MLE. Present the typical regression statistics.

In principle, a Poisson regression model specifies that

$$\Pr(y_i = k|x_i) = \frac{\exp(-\lambda_i)\lambda_i^k}{k!},$$

Given $\ln(\lambda_i) = \ln(\mathbb{E}(Y_i|X_i)) = \beta' X_i$, we interpret the j^{th} element of vector β, β_j ($j = 1, 2, \ldots, k$) here as the percentage change of $\mathbb{E}(Y|X)$ when there is a unit change in X. To estimate this unknown vector β, we apply a maximum likelihood method with the log-likelihood function:

$$\sum_{i=1}^{n} \left[-\exp(x_i'\beta) + y_i(x_i'\beta) - \log(y_i!) \right]. \tag{4.1.1}$$

Notice here we assume that dependent variables y_i follow certain distribution, the log-likelihood function comes directly from the cumulative distribution function (CDF) of y_i. The following code gives an example of MLE which estimated unknown coefficients B by maximizing the log-likelihood function (4.1.1).

```
$title Count data model: Poisson regression
set             i               "Index of observations"                 /1*200/,
                n               "Index of right hand side variables"    /"cons","x1"/,
                k               "Three type of standard error estimation"
                                /"inverse Hessian","inverse W","robust estimate for cov"/;

alias (n,nn);

parameter       y(i)            "Random number with poisson distribution",
                lambda(i)       "Conditional mean of poisson distribution",
                beta(n)         "Given coefficients of right hand side variables"   /"cons" 1.5, "x1" -0.5/,
                x(i,n)          "Right hand side variables";

option seed = 1001;
x(i,"cons") = 1;

*       Generate normally distributed random variables
x(i,"x1")   = normal(0,1);

*       Define Poisson parameter
lambda(i)   = exp(sum(n, beta(n)*x(i,n)));

* Generate exponential variates log(U) and stop as soon as sum(log(U)) >= lambda(i)
scalar          poisson         "Poisson variate",
                s               "Intermediate variable in generating Poisson variate";

loop( i, poisson = -1; s = 0; repeat  s = s - log(uniform(0.001,1)); poisson = poisson + 1;
                            until  (s >= lambda(i));  y(i) = poisson; );

variable        B(n)            "Poisson Maximum Likelihood estimators",
                LL(i)           "Log likelihood function at each observation t",
                LOGLIK          "Log likelihood function";

equation        obj             "Objective",
                llf(i)          "Log likelihood definition at each observation t";

llf(i)..        LL(i) =e= -exp(sum(n,x(i,n)*B(n))) + y(i)*sum(n, x(i,n)*B(n));
obj..           LOGLIK =e= sum(i, LL(i));

model mle /obj,llf/;
solve mle using nlp maximizing LOGLIK;

display y,B.L, LOGLIK.L;
```

Next, there are several estimated variance-covariance matrices of the coefficients $\hat{V}_{\hat{\theta}}$ in maximum

likelihood estimation defined as follows. In most of the later examples, we pick the inverse hessian definition for most of the applications just for its simplicity in calculation:

1) Inverse Hessian

$$\hat{V}_{\hat{\theta}} = -(H_\theta)^{-1},$$

where H_θ is the Hessian matrix of the log-likelihood function with respect to coefficients.

2) Inverse ω

$$\hat{V}_{\hat{\theta}} = (J_\theta J_\theta')^{-1},$$

where J_θ is the Jacobian matrix of the log-likelihood function with respect to coefficients.

3) Robust estimate of covariance matrix

$$\hat{V}_{\hat{\theta}} = (H_\theta)^{-1}(J_\theta J_\theta')(H_\theta)^{-1}$$

where H_θ and J_θ are defined as above.

Here is an exercise in which we derive these estimated variance-covariance matrices.

```
*       Define parameters
parameter       Jacobian(i,n)           "Jacobian of constraints wrt coefficients",
                Hessian(i,n,nn)         "Hessian of constraints wrt coefficients",
                Hessian_sum(n,nn)       "Hessian of log-likelihood wrt coefficients",
                jacsqr(n,nn)            "Squared Jacobian matrix",
                jsinv(n,nn)             "Inverse of jacsqr",
                hinv(n,nn)              "Inverse of ssehess",
                cov(k,n,nn)             "Covariance matrix of type k",
                statistics(k,n,*)       "Statistics at the point";

*       Report Jacobian and Hessian matrix
$batinclude lsa mle nlp "maximizing LOGLIK"

Jacobian(i,n)     = LSA_DF( llf(i), B(n) );
Hessian(i,n,nn)   = LSA_D2F( llf(i), B(n), B(nn) );

*       d2f is an upper triangular matrix
hessian(i,n,nn) = hessian(i,n,nn)$(ord(nn) ge ord(n)) + hessian(i,nn,n)$(ord(nn) lt ord(n));

*      Find squared Jacobian and Hessian of sum of log-likelihood functions
jacsqr(n,nn)      = sum(i, Jacobian(i,n)*Jacobian(i,nn));
Hessian_sum(n,nn) = sum(i, Hessian(i,n,nn));

*       Find inverse of Hessian and Squared jacobian matrix
$batinclude arealinverter Hessian_sum n nn hinv
$batinclude arealinverter jacsqr n nn jsinv

*       Three typies of estiamted covariance matrix of coefficients
cov("inverse Hessian",n,nn) = hinv(n,nn);
cov("inverse W",n,nn)       = jsinv(n,nn);

alias (n,m,mm);
cov("robust estimate for cov",n,nn) = sum((mm,m), hinv(n,mm)*jacsqr(mm,m)*hinv(m,nn));

*       Report statistics
statistics(k, n, "estimator")  = B.L(n);
statistics(k, n, "std error")  = sqrt(cov(k,n,n));
statistics(k, n, "T value")    = B.L(n)/sqrt(cov(k,n,n));
* Use the BETAREG function:
statistics(k, n, "P value")    = BETAREG( (card(i) - card(n))/
                                          (card(i) - card(n) + sqr(statistics(k, n, "T value"))),
                                          (card(i) - card(n))/2, 0.5 );

display statistics;
```

From the following list, we report different estimates of covariance matrix under different definitions. Notice that both *T-value* and *p-value* are strongly against the null hypothesis that coefficients are zero, as expected.

```
----    485 PARAMETER statistics  Statistics at the point
                                 estimator    std error     T value     P value
inverse Hessian          .cons     1.556808     0.033615   46.313295 3.4346E-108
inverse Hessian          .x1      -0.436582     0.033745  -12.937755 3.78021E-28
inverse W                .cons     1.556808     0.032251   48.271768 1.8312E-111
inverse W                .x1      -0.436582     0.031328  -13.935707 3.27846E-31
robust estimate for cov.cons      1.556808     0.035049   44.418638 6.5138E-105
robust estimate for cov.x1       -0.436582     0.036398  -11.994525 2.85663E-25
```

4.2 Binary choice

Next, we introduce another limited dependent variable model. Given some explanatory variables x_t ($I \times 1$) and dependent variable $y_t \in \{0, 1\}$ (which represents a Yes/No outcome), binary choice model describes the conditional distribution of $\Pr(y_t = 1|x_t)$.

This model is identical to the *latent variable* model

$$y_t^* = x_t'\beta + e_t,$$

in which

$$e_t \sim F(\cdot),$$

$$y_t = \begin{cases} 1 & \text{if} \quad y_t^* > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Suppose we still use a multiple linear regression model, or the *linear probability model* (LPM)

$$y_t = x_t'\beta + \mu_t, \quad y_t \in \{0, 1\}$$

where the j^{th} element of coefficients vector β, β_j ($j \in \{1, 2, \ldots, I\}$) is an unknown coefficient. Under zero conditional mean assumption $\mathbb{E}(\mu_t|x_t) = 0$, we define the response probability

$$\Pr(y_t = 1) = \mathbb{E}(y_t|x_t) = x_t'\beta,$$

the coefficient β_j measures the change in the response probability when there is unit change in x_j:

$$\beta_j = \frac{\partial \Pr(y_t = 1)}{\partial x_j}.$$

Limitations of LPM are obvious: First, the predicted probability, $x_t'\hat{\beta}$ could be outside [0,1]. To impose

a restriction, the standard alternative is to use a function of the form

$$\Pr(y_t = 1|x_t) = F(x_t'\beta),$$

where $F(\cdot)$ is a known cumulative distribution function (CDF).

The two standard choices for F are: Logistic $F(u) = (1 + e^{-u})^{-1}$, we call this the *logit* model; And Normal $F(u) = \Phi(u)$, we call this the *probit* model. To be more specific, a *logit* model has the form as

$$\Pr(y_t = 1|x_t) = \frac{exp(x_t'\beta)}{1 + exp(x_t'\beta),} \tag{4.2.1}$$

and a *probit* model has the form as

$$\Pr(y_t = 1|x_t) = \Phi(x_t'\beta). \tag{4.2.2}$$

In general, if the error terms follow an independent, extreme value distribution, *logit* is a good choice; For the case when error terms follow an i.i.d normal distribution, *probit* is a standard way to address the problem.

Second, there are common concerns about *heteroscedasticity* in LPM. Given only two estimated residuals in this type of model,

$$\begin{aligned} \hat{\mu}_t = 1 - x'\hat{\beta} \quad &= 1 - \Pr(y_t = 1) \quad \text{if} \quad y_t = 1, \\ \hat{\mu}_t = -x'\hat{\beta} \quad &= \Pr(y_t = 1) \quad \text{if} \quad y_t = 0 \end{aligned}$$

We then have

$$\begin{aligned} \hat{var}(\hat{\mu}_t|x) \quad &= \mathbb{E}(\hat{\mu}_t^2|x) \\ &= \Pr(y_t = 1)[1 - \Pr(y_t = 1)]^2 + \Pr(y_t = 1)^2[1 - \Pr(y_t = 1)] \\ &= \Pr(y_t = 1)[1 - \Pr(y_t = 1)], \end{aligned}$$

which depends on index i.

In the following exercise, we show an example of *homoscedastic probit* model in binary choice model first:

4.2.1 Limited dependent variable model example 2: binary model, probit

A biannual survey of households (ENIGH) undertaken by The national institute of statistics in Mexico has been used by many academics to study income inequality, returns to education, gender based income differences, consumer expenditure patterns and many other issues. Use this data to examine purchases of fluid milk by Mexican households as there is continuing concern as to the lack of an adequate intake of calcium especially by children.

To develop your model you assume that the net utility gained from the consumption of fluid milk (U^*) is related to a set of exogenous variables (X) via the following:

$$U_t^* = X_t\beta + \mu_t, \tag{4.2.3}$$

where $\mu_t \sim N(0,1)$.

The relationship between the latent variable U^* and the discrete (0/1) variable of whether a household purchases fluid milk, MILK_D, can be represented via the following.

$$MILK\_D_t = \begin{cases} 1 & \text{if } U_t^* > 0 \\ 0 & \text{otherwise} \end{cases}$$

One can use this to motivate a probit model of fluid milk purchases. Table (2) contains a listing of the variables in the version of the dataset that has been made available for your use.

Variables	Description	Units
fluidx	Expenditures on pasteurized fluid milk	Pesos
	Household Composition Variables	
perlt6	Percent of household members < 6 years old	%
per6_11	Percent of household members between 6 and 11 years old	%
perge66	Percent of household members older than 65	%
num_yung	Number of household members < 12 years of age	#
	Other Household Characteristics	
sm_city	Household is located in a town with 2,500-15,000 population	0/1
city_11	Household is located in a town with > 15,000 population	0/1
rural	Household is located in a town with < 2,500 population	0/1
refrig	Does the household own a refrigerator/freezer?	0/1
incomet	Quarterly household income	10,000 Pesos
perfafh	Percent of weekly household food expenditures spent on food purchased and consumed outside the home	%
regdf	Household located in the Federal District region of Mexico	0/1

Table 2: Description of a Subset of Variables in 2002 ENIGH Dataset

In this example, we are first asked to estimate the *probit* model, test the significance of exogenous variables in part (a); Define and test the income elasticity in part (c); Then estimate and test the marginal effect of having a refrigerated space on milk purchasing probability in part (d).

(a) Initially you would like to estimate the homoscedastic probit model represented in equation (4.2.4):

$$\Pr(fluidx > 0) = F(Constant, num\_yung, incomet, num\_yung * incomet, sm\_city, city, refrig, perfafh) \tag{4.2.4}$$

Present estimated coefficient values and associated standard errors. Undertake a statistical test that the above exogenous variables, as a group, explain a significant portion of the pattern of observed values of the dependent variable.

We show the maximum likelihood estimation of a *probit* model here: Given data x_t, y_t, the estimator $\hat{\beta}$ maximize the following log-likelihood function

$$\ln \mathcal{L}(\beta) = \sum_t [y_t \ln \Phi(x_t'\beta) + (1 - y_t) \ln(1 - \Phi(x_t'\beta))],$$

where $\Phi(\cdot)$ is the standard normal CDF.

In this part, we compare the model in equation (4.2.4) with a restricted model when only the constant intercept term is included in the model to check whether such a "naive" model is enough to explain the pattern of observed values of the dependent variable:

```
$ title Probit model of fluid milk purchases
*        Part(a)
*        Homoscedastic probit model

set     t               "Index of observations"                         /1*14646/,
        i               "Index for BETA"                                /"cons","num_yung","incomet","num_yung_incomet",
                                                                        "sm_city","city","refrig","perfafh"/
        res(i)          "Index for BETA in the restricted case"         /"cons"/;

alias (i,ii),(r,res);

parameter        data(t,*)        "Source data";

$call gdxxrw fluid_probit_data_gams.xls par=data rng=A1:AC14647 cdim=1 rdim=1 checkDate
$gdxin fluid_probit_data_gams.gdx
$loaddc data
$gdxin

parameter
        y(t)            "1 if a household purchases fluid milk",
        x(t,i)          "Explanatory variable i of observation t";

*        Expenditures on fluid milk
y(t) = data(t,"fluidx");
*        Dependent variable
y(t) = 1$(y(t)>0);

*        Define x as a t row, 8 column matrix
*        x is the unrestricted matrix of exogenous variables
x(t,"cons") = 1;
*        Number of household members < 12 years of age
x(t,"num_yung") = data(t,"num_yung");
*        Quarterly household income
x(t,"incomet") = data(t,"incomet");
*        Multiplication of quarterly income and number of members < 12
x(t,"num_yung_incomet") = data(t,"num_yung")*data(t,"incomet");
*        Household is located in a town with 2500-15000 population
x(t,"sm_city") = data(t,"sm_city");
*        Household is located in a town with > 15000 population
x(t,"city") = data(t,"city");
*        1 if the household own a refrigerator.freezer
x(t,"refrig") = data(t,"refrig");
*        Percent of weekly household food expenditures spent on food
*        purchased and consumed outside the home
x(t,"perfafh") = data(t,"perfafh");
```

```
*       Use Log-likelihood function for a 0/1 dependent variable with a
*       error term distributed i.i.d N(0,1)
equation        fit(t)          "Linear model",
                obj             "Objective";

variable        BETA(i)         "Coefficients to be estimated",
                LOGLIK          "Log-Likelihood";

equation        obj_ur          "Objective for a unrestricted model";

obj_ur..        LOGLIK =e= sum(t, y(t)*log(    errorf(sum(i, x(t,i)*BETA(i)))) +
                           (1-y(t))*log(1 - errorf(sum(i, x(t,i)*BETA(i)))));
BETA.LO(i) = -100;
BETA.UP(i) = 100;

model unres_mle /obj_ur/;
solve unres_mle maximizing LOGLIK using nlp;

*       Restore estimates for future use
parameter       ll_ur           "Log-Likelihood level of the unrestricted model"
                b_ur(i)         "Coefficient estimation in the unrestriced model";
ll_ur   = LOGLIK.L;
b_ur(i) = BETA.L(i);

parameter       Hessian(i,ii)   "Hessian of log-likelihood wrt coefficients",
                hinv(i,ii)      "Inverse of ssehess",
                cov(i,ii)       "Covariance matrix";

*       Report Hessian matrix
$batinclude lsa unres_mle nlp "maximizing LOGLIK"
hessian(i,ii)   = LSA_D2F( obj_ur, BETA(i), BETA(ii) );

*       d2f is an upper triangular matrix
hessian(i,ii) = hessian(i,ii)$(ord(ii) ge ord(i)) + hessian(ii,i)$(ord(ii) lt ord(i));

$batinclude arealinverter Hessian i ii hinv
cov(i,ii)     = hinv(i,ii);

*       Record statistics of the unrestricted case
parameter       cov_unres(i,ii) "Covariance matrix of the unrestricted case",
                stat_unres(i,*) "Statistics at the point";;

*       Record covariance estimated for the unrestricted case
cov_unres(i,ii) = cov(i,ii);

*       Report statistics for the unrestricted case
stat_unres(i, "estimator")  = BETA.L(i);
stat_unres(i, "std error")  = sqrt(cov(i,i));
stat_unres(i, "T value")    = BETA.L(i)/sqrt(cov(i,i));
*       Use the BETAREG function:
stat_unres(i, "P value")    = BETAREG( (card(t) - card(i))/
                                       (card(t) - card(i) + sqr(stat_unres(i, "T value"))),
                                       (card(t) - card(i))/2, 0.5);
display stat_unres;

*       Now move to restricted case:
equation        obj_res                         "Objective function in restricted MLE";

obj_res..        LOGLIK =e= sum(t, y(t)*log(    errorf(sum(res, x(t,res)*BETA(res)))) +
                            (1-y(t))*log(1 - errorf(sum(res, x(t,res)*BETA(res)))));

model res_mle  /obj_res/;

*       Initialize BETA(res)
BETA.L(res) = 0.5;
solve res_mle maximizing LOGLIK using nlp;

*       Record objective value for future use
parameter               ll_res          "Log-Likelihood level of the restricted model";
ll_res = LOGLIK.L;

*       Report Hessian matrix
parameter       stat_res(i,*)           "Report of statistics in the restricted model";
$batinclude lsa res_mle nlp "maximizing LOGLIK"
hessian(i,ii) = LSA_D2F( obj_res, BETA(i), BETA(ii) );

*       d2f is an upper triangular matrix
hessian(i,ii) = hessian(i,ii)$(ord(ii) ge ord(i)) + hessian(ii,i)$(ord(ii) lt ord(i));

*       Find the inverse of Hessian
$batinclude arealinverter Hessian res r hinv
```

```
cov(res, r)   = hinv(res, r);

*        Generate report of statistics
stat_res(r, "estimator")  = BETA.L(r);
stat_res(r, "std error")  = sqrt(cov(r,r));
stat_res(r, "T value")    = BETA.L(r)/sqrt(cov(r,r));
*        Use the BETAREG function:
stat_res(r, "P value")    = BETAREG( (card(t) - card(r))/
                                     (card(t) - card(r) + sqr(stat_res(r, "T value"))),
                                     (card(t) - card(r))/2, 0.5 );
display stat_res;
```

Here we report the regression statistics of the original (unrestricted) model and the restricted model:

```
----    327 PARAMETER stat_unres  Statistics at the point

                   estimator   std error     T value     P value
cons               -0.918630    0.034911  -26.313846
num_yung           -0.067833    0.015551   -4.361949    0.000013
incomet             0.185535    0.018152   10.221274
num_yung_incomet    0.047756    0.010214    4.675732    0.000003
sm_city             0.284923    0.037497    7.598472
city                0.696501    0.027548   25.282728
refrig              0.558470    0.027408   20.375888
perfafh            -0.447874    0.056634   -7.908197

----    596 PARAMETER stat_res  Report of statistics in the restricted model

       estimator   std error     T value
cons    0.139077    0.010393   13.382175
```

We do hypothesis testing in this *probit* model using the likelihood ratio (LR) test. The LR test is based on the difference in the log-likelihood functions for the unrestricted and restricted models. In general, the larger the fall of the log-likelihood (from the value of the unrestricted case to that of the restricted case), the more likely we want to reject the null hypothesis.

Let $\mathcal{L}_{ur}$ ($\mathcal{L}_{res}$) denote the maximized log-likelihood value for the unrestricted (restricted) model. Then the likelihood ratio statistic follows a *Chi-square* distribution

$$LR = 2(\mathcal{L}_{ur} - \mathcal{L}_{res}) \xrightarrow{\text{d}} \chi^2_q,$$

where q is the number of restrictions in a hypothesis.

```
*        Likelihood ratio test
parameter       lr_res               "LR test statistic against H0: Restricted model is true",
                pval_res             "P value of LR test",
                alpha                "Type 1 error level"   /0.05/;

*        Find LR test statistics and p value based on H0
lr_res = 2*(ll_ur - ll_res);
*       Degrees of freedom: card(i) - card(r)
*       which equals the number of restrictions here
pval_res = 1- gammareg(lr_res/2,(card(i) - card(r))/2);

display lr_res, pval_res;

if ( pval_res lt alpha, display "There is, therefore, enough evidence to reject H0";
                 else  display "There is, therefore, not enough evidence to reject H0"; );
```

Likelihood ratio test shows that it is statistically safe to reject the restricted model:

```
----    608 PARAMETER lr_res              =  2833.535022  LR test statistic with H0: Restricted model is true
            PARAMETER pval_res            =     0.000000  P value of LR test

----    610 There is, therefore, enough evidence to reject H0
```

(c) At the mean of the data, what is the income elasticity of the probability of purchasing fluid milk? From a statistical perspective is this elasticity significant?

We evaluate income elasticity at the mean of the data, then apply *z-test* to check whether it is significantly different from zero. A *z-test* assumes the distribution of the test statistic under the null hypothesis can be approximated by a normal distribution.

```
*        Part(c)
parameter       x_bar(i)        "Mean of variable i";
x_bar(i) = sum(t,x(t,i))/card(t);

variable        I_ELS           "Income elasticity evaluated at the mean of the data";

equation        def_ie          "Definition of income elasticity";

def_ie..        I_ELS =e= (BETA("incomet") + BETA("num_yung_incomet")*x_bar("num_yung"))*
                          (1/sqrt(2*pi))*exp(-0.5*(sqr(sum(i, x_bar(i)*BETA(i)))))*
                           x_bar("incomet")/errorf(sum(i, x_bar(i)*BETA(i)));
model ie /def_ie/;
BETA.FX(i) = b_ur(i);
solve ie using mcp;

*        Report Jacobian matrix
parameter       grad_ie(i)      "Gradient of income elasticity wrt coefficients";

$batinclude lsa ie mcp

grad_ie(i) = LSA_DF( def_ie, BETA(i) );

*       Z test
set             r_ie            "Row index in displaying income elasticity statistics"  /IncomeElas/;

parameter       v_hat_ie        "Estimated variance of income elasticity",
                std_ie          "Estimated standard deviation of income elasticity",
                z_ie            "Z statistic based on H0: income elasticity = 0",
                pval_ie         "P value of testing income elasticity = 0",
                stat_ie(r_ie,*) "Statistics of income elasticity estimation";

*        Find variance and std error of income elasticity estimator
v_hat_ie = sum((i,ii),grad_ie(i)*cov_unres(i,ii)*grad_ie(ii));
std_ie   = sqrt(v_hat_ie);

*     Display estimator and its standard deviation
stat_ie(r_ie,"Estimator") = I_ELS.L;
stat_ie(r_ie,"Std Error") = std_ie;
display stat_ie;

*     Find Z statistics
z_ie    = I_ELS.L/std_ie;
pval_ie = 1 - errorf(abs(z_ie));
display z_ie, pval_ie;

if((pval_ie lt alpha/2), display "There is enough evidence to reject H0";
                  else  display "There is not enough evidence to reject H0";);
```

According to the *p-value*, We reject the null claiming that the income elasticity is statistically zero.

```
----   1082 PARAMETER stat_ie  Statistics of income elasticity estimation
            estimator   std error
IncomeElas    0.243219    0.014251

----   1087 PARAMETER z_ie                =    17.066302  Z statistic based on H0: income elasticity = 0
            PARAMETER pval_ie             =     0.000000  P value of testing income elasticity = 0

----   1089 There is enough evidence to reject H0
```

(d) Based on results obtained from estimating equation (4.2.4) what is the impact of having refrigerated storage in the household on the probability of purchasing fluid milk? Is this impact positive from a statistical perspective?

First of all, since *probit* model is nonlinear, β_{refrig} is not the coefficient of impact we are looking for. Analytically, marginal effects in changes in explanatory variable j in a *probit* model is

$$\frac{\partial \Pr(Y_t = 1|x_t)}{\partial x_{t,j}} = \beta_j \phi(x_t'\beta),$$

according to the *probit* model (4.2.2), where ϕ is the probability density function of a standard normal distribution.

In the following code, we address the marginal effects of having a refrigerated storage or not on the milk purchasing probability directly. Fist, dividing the data into two groups, one having a refrigerated storage and the other not; We then find the probability difference of milk purchasing based on these two groups, which equals the marginal effects of having a refrigerated space. After that, we run a *z-test* to check whether this impact is significant or not.

```
*        Part(d)
parameter       x_bar_ref(i)            "Mean vector of variable x with refrig = 1",
                x_bar_noref(i)          "Mean vector of variable x with refrig = 0";

*       Generate dummy variable for having refrigerated storage or not
x_bar_ref(i) = x_bar(i);
x_bar_ref("refrig") = 1;

x_bar_noref(i) = x_bar(i);
x_bar_noref("refrig") = 0;

*       Define the probability difference in purchasing milk caused by having refigerated storage
variable        P_REF                   "Difference in purchasing milk caused by having refigerated storage or not";

equation        def_ref                 "Definition of the probability difference mentioned";

def_ref..        P_REF =e= errorf(sum(i, x_bar_ref(i)*BETA(i))) - errorf(sum(i, x_bar_noref(i)*BETA(i)));

model ref /def_ref/;
BETA.FX(i) = b_ur(i);
solve ref using mcp;

*        Report Jacobian matrix
parameter       grad_ref(i)             "Gradient of probability difference wrt coefficients";

$batinclude lsa ref mcp

grad_ref(i) = LSA_DF( def_ref, BETA(i) );

*       z test
set             r_ref                   "Row index of 'having refigerated storage or not' test"  /RefrigImp/;

parameter       v_hat_ref               "Estimated variance of the effect of having refigerated storage",
```

```
                std_ref                 "Estimated standard deviation of the effect of having refigerated storage",
                z_ref                   "Z statistic based on H0: the effect of having refigerated storage > 0",
                pval_ref                "P value of testing the effect of having refigerated storage > 0",
                stat_ref(r_ref,*)       "Statistics of estimation";

*     Find variance and std error of having refigerated storage or not estimator
v_hat_ref = sum((i,ii), grad_ref(i)*cov_unres(i,ii)*grad_ref(ii));
std_ref   = sqrt(v_hat_ref);

*     Display estimator and its standard deviation
stat_ref(r_ref,"Estimator") = P_REF.L;
stat_ref(r_ref,"Std Error") = std_ref;

*     Find Z statistics
z_ref    = P_REF.L/std_ref;
pval_ref = 1 - errorf(abs(z_ref));

display stat_ref, z_ref, pval_ref;

if( (pval_ref lt alpha),
        display "There is enough evidence to reject H0: Having refigerated storage has significant impact";
  else  display "There is not enough evidence to reject H0: Having refigerated storage has significant impact";);
```

According to the *p-value* of a *z-test*, we reject the null hypothesis that having refigerated storage has positive impact on milk purchasing probability.

```
----   1276 PARAMETER stat_ref  Statistics of estimation
           estimator   std error
RefrigImp    0.219878    0.010545

----   1276 PARAMETER z_ref                =    20.850749
           Z statistic based on H0: the effect of having refigerated storage  > 0
           PARAMETER pval_ref              =     0.000000
           P value of testing the effect of having refigerated storage > 0

----   1278
There is enough evidence to reject H0: Having refigerated storage has significant impact
```

4.2.2 Limited dependent variable model example 3: binary choice model, probit with heteroscedasticity

Using equation (4.2.4) as a base, you believe that the underlying latent model of milk consumption possesses a heteroscedastic error term. As such you are concerned about obtaining consistent parameter estimates and correct parameter standard errors. You therefore would like to estimate equation (4.2.4) but this time incorporating a heteroscedastic error structure in the latent regression equation. As such, you assume that latent error, ϵ_t has the following variance specification:

$$\sigma_t^2 = [\exp(Z_t\gamma)]^2, \tag{4.2.5}$$

where σ_t^2 is the error variance for the t^{th} error term, Z_t is a vector of exogenous variables and γ a vector of error variance coefficients to be estimated.

You hypothesize that the Z matrix is composed of the following exogenous variables: perge66, refrig, regdf, incomet, perfafh.

In this exercise, we first estimate the *probit* model with the error specification which requires some modification of the log-likelihood function; In part (b), we check whether the elasticity of the effect of a

change in income on the probability of a household purchasing fluid milk would be different in a higher *vs.* lower income household.

(a) Estimate a heteroscedastic probit model using equation (4.2.4) as a base, the error specification shown in equation (4.2.5) and the Z matrix defined above.

We model the heterogeneity using Harvey's "multiplicative heteroscedasticity" approach (1976), which will lead to a log-likelihood function very similar to the usual probit log-likelihood:

$$logL = \sum_{t=1}^{n}[y_t log\Phi(\frac{X_t\beta}{exp^{z_t\gamma}}) + (1-y_t)log(1-\Phi(\frac{X_t\beta}{exp^{z_t\gamma}}))];$$

Then after estimation based on the above log-likelihood function, we test whether the model with homoscedastic errors is true through a Likelihood Ratio test.

```
$title Probit with heteroscedastic error structure in the latent regression
*       Part(a)
*       Estimate a probit model with heteroscedastic errors
set     t       "Index of observations"              /1*14646/,
        n       "Combination of set i and k"
                /"cons","num_yung","incomet","num_yung_incomet","sm_city","city","refrig","perfafh","perge66",
                 "refrig_z","regdf","incomet_z","perfafh_z"/,
        i(n)    "Index for BETA"
                /"cons","num_yung","incomet","num_yung_incomet","sm_city","city","refrig","perfafh"/,
        k(n)    "Index for GAMMA"                    /"perge66","refrig_z","regdf","incomet_z","perfafh_z"/;

alias (i,ii), (n,nn);

*       Read data from an Excel file
parameter       data(t,*)       "Source data";

$call gdxxrw fluid_probit_data_gams.xls par=data rng=A1:AC14647 cdim=1 rdim=1 checkDate
$gdxin fluid_probit_data_gams.gdx
$loaddc data
$gdxin

parameter       y(t)            "1 if a household purchases fluid milk",
                x(t,n)          "Right hand side variables",
                z(t,n)          "Exogenous variables in specifying the error variance";

*       Expenditures on fluid milk
y(t) = data(t,"fluidx");
*       Dependent variable
y(t) = 1$(y(t)>0);

*       Define x as a t row, 8 column matrix
*       x is the unrestricted matrix of exogenous variables
x(t,"cons")     = 1;
*       Number of household members < 12 years of age
x(t,"num_yung") = data(t,"num_yung");
*       Quarterly household income
x(t,"incomet")  = data(t,"incomet");
*       Multiplication of quarterly income and number of members < 12
x(t,"num_yung_incomet") = data(t,"num_yung")*data(t,"incomet");
*       Household is located in a town with 2500-15000 population
x(t,"sm_city")  = data(t,"sm_city");
*       Household is located in a town with > 15000 population
x(t,"city")     = data(t,"city");
*       1 if the household own a refrigerator.freezer
x(t,"refrig")   = data(t,"refrig");
*       Percent of weekly household food expenditures spent on food
*       purchased and consumed outside the home
x(t,"perfafh")  = data(t,"perfafh");

*       Define z as a t row, 5 column matrix
z(t,"perge66")   = data(t,"perge66");
z(t,"refrig_z")  = data(t,"refrig");
z(t,"regdf")     = data(t,"regdf");
z(t,"incomet_z") = data(t,"incomet");
z(t,"perfafh_z") = data(t,"perfafh");
```

```
variable        LOGLIK          "Log-Likelihood function",
                COEF(n)         "Coefficients to be estimated";

equation        obj_llf         "Objective";

obj_llf..       LOGLIK =e= sum(t,  y(t)*log(    errorf(sum(i, x(t,i)*COEF(i))/exp(sum(k,z(t,k)*COEF(k))))) +
                           (1 - y(t))*log(1 - errorf(sum(i, x(t,i)*COEF(i))/exp(sum(k,z(t,k)*COEF(k))))));

model hetero /obj_llf/;
solve hetero maximizing LOGLIK using nlp;

*       Report log-likelihood level of unrestricted model
parameter       loglik_ur       "Log-Likelihood level of the unrestricted model",
                c_ur(n)         "Coefficient estimates of the unrestricted model";

loglik_ur = LOGLIK.L;
c_ur(n)   = COEF.L(n);

*       Report statistics of the unrestricted model
parameter       Hessian(n,nn)   "Hessian of log-likelihood wrt coefficients",
                hinv(n,nn)      "Inverse of ssehess",
                cov(n,nn)       "Estimated covariance matrix";

*       Report Hessian matrix
$batinclude lsa hetero nlp "maximizing LOGLIK"
hessian(n,nn)   = LSA_D2F( obj_llf, COEF(n), COEF(nn) );

*       d2f is an upper triangular matrix
hessian(n,nn) = hessian(n,nn)$(ord(nn) ge ord(n)) + hessian(nn,n)$(ord(nn) lt ord(n));

*       Find inverse matrix of the Hessian
$batinclude arealinverter Hessian n nn hinv
cov(n,nn) = hinv(n,nn);

parameter       cov_ur(n,nn)    "Covariance matrix in the unrestricted model",
                stat_unres(n,*) "Statistics at the point";

cov_ur(n,nn) = cov(n,nn);

*      Report statistics
stat_unres(n, "estimator")  = COEF.L(n);
stat_unres(n, "std error")  = sqrt(cov_ur(n,n));
stat_unres(n, "T value")    = COEF.L(n)/sqrt(cov_ur(n,n));
*       Use the BETAREG function:
stat_unres(n, "P value")    = BETAREG( (card(t) - card(n))/
                                       (card(t) - card(n) + sqr(stat_unres(n, "T value"))),
                                       (card(t) - card(n))/2, 0.5 );
display stat_unres;

*       LR Test against H0: model with homoscedastic errors is true
parameter               df_lr           "Degrees of freedom",

*       From previous example
                        loglik_res      "Log-likehood function value in the model with homoscedastic errors" /-8645.288/,
                        lr              "Likelihood ratio test statistic",
                        pval_lr         "P value with respect to H0",
                        alpha           "Type 1 error level"   /0.05/;

*       df: number of coefficients in the heteroscedastic model -
*           number of coefficients in the homoscedastic model
df_lr   = card(n) - card(i);
lr      = 2*(loglik_ur - loglik_res);
pval_lr = 1 - gammareg(lr/2,df_lr/2);

display lr, pval_lr;

if((pval_lr lt alpha), display "There is enough evidence to reject H0: model with homoscedastic errors is true";
                else   display "There is not enough evidence to reject H0: model with homoscedastic errors is true";);
```

Here are the estimation and testing results and we reject the null hypothesis that homoscedastic model is true based on these results:

```
----    271 PARAMETER stat_unres  Statistics at the point
                estimator   std error     T value     P value
```

```
cons               -1.428304    0.077308  -18.475419
num_yung           -0.082761    0.023557   -3.513290    0.000444
incomet             0.565599    0.062439    9.058457
num_yung_incomet    0.059811    0.019585    3.053866    0.002263
sm_city             0.326598    0.055360    5.899498
city                0.974120    0.055241   17.634033
refrig              0.708772    0.041636   17.023240
perfafh            -0.705461    0.101377   -6.958803
perge66             0.117924    0.103832    1.135721    0.256092
refrig_z            0.113552    0.069065    1.644130    0.100171
regdf               0.219298    0.069001    3.178195    0.001485
incomet_z           0.257464    0.033199    7.755212
perfafh_z           0.050915    0.111768    0.455539    0.648728


----    288 PARAMETER lr                  =   126.077085  Likelihood ratio test statistic
            PARAMETER pval_lr             =     0.000000  P value

----    290 There is enough evidence to reject H0: model with homoscedastic errors is true
```

(b) Using the heteroscedastic model results, what is the elasticity of the effect of a change in income on the probability of a household purchasing fluid milk when using the mean values of the exogenous variables for the sample of households with income no more than 70% of the mean income level. Evaluate the same elasticity this time using mean values of the exogenous variables for the sample with at least 130% of total sample mean income. Are these elasticities equal?

In this part, We first evaluate milk purchasing probability's income elasticity at different income level, find gradients of these elasticity with respect to estimated coefficients before construct a *z-test* to check whether income level affects income elasticity or not:

```
*       Part(b)
*       First check the elasticity with income no more than 70% of the mean income
*       then evaluate the elasticity with income at least 130% of mean income

*       Test if these two elasticities are statistically equal to each other
set             t70(t)          "Subsample with income no more than 70% of the mean income level",
                t130(t)         "Subsample with income higher than 130% of the mean income level",
                r               "Index for income elasticities"    /"0.7i","1.3i",diff/,
                ie(r)           "Subset of r"                      /"0.7i","1.3i"/;

parameter       mean_z(n)       "Sample mean of data z",
                mean_x(n)       "Sample mean of data x",
                m_z(r,n)        "Sample mean of data z in income interval r",
                m_x(r,n)        "Sample mean of data x in income interval r";

mean_z(k) = sum(t,z(t,k))/card(t);
mean_x(i) = sum(t,x(t,i))/card(t);

t70(t)    = yes$(x(t,"incomet") le 0.7*mean_x("incomet"));
t130(t)   = yes$(x(t,"incomet") ge 1.3*mean_x("incomet"));

m_z("0.7i",k) = sum(t70,z(t70,k))/card(t70);
m_z("1.3i",k) = sum(t130,z(t130,k))/card(t130);
m_x("0.7i",i) = sum(t70,x(t70,i))/card(t70);
m_x("1.3i",i) = sum(t130,x(t130,i))/card(t130);

variable        ELAS(r)         "Purchasing probability's income elasticity";

equation        def_elas(r)     "Definition of ELAS(r)";

def_elas(ie)..  ELAS(ie) =e= (COEF("incomet") + COEF("num_yung_incomet")*m_x(ie,"num_yung") -
                              sum(nn, m_x(ie,nn)*COEF(nn))*COEF("incomet_z"))/exp(sum(nn, m_z(ie,nn)*COEF(nn)))*
                              (1/sqrt(2*pi))*exp(-0.5*sqr(sum(nn, m_x(ie,nn)*COEF(nn))/exp(sum(nn, m_z(ie,nn)*COEF(nn)))))*
                              m_x(ie,"incomet")/errorf(sum(nn, m_x(ie,nn)*COEF(nn))/exp(sum(nn, m_z(ie,nn)*COEF(nn))));

COEF.FX(n) = c_ur(n);
model el_i /def_elas/;
solve el_i using mcp;
```

```
*       Report statistics
parameter       grad_ie(r,n)    "Gradients of income elasticity function ie w.r.t coefficients",
                se_elas(r)      "Standard deviation of income elasticity estimation",
                stat_elas(r,*)  "Statistics of the regression";

*       Report Jacobian matrix
$batinclude lsa el_i mcp

grad_ie(ie,n) = LSA_DF( def_elas(ie), COEF(n) );
grad_ie("diff",n) = grad_ie("0.7i",n) - grad_ie("1.3i",n);

*       Find standard deviation
se_elas(r) = sqrt(sum((n,nn), grad_ie(r,n)*cov_ur(n,nn)*grad_ie(r,nn)));

*       Display estimator and standard error
stat_elas(ie,"estimator")     = ELAS.L(ie);
stat_elas("diff","estimator") = ELAS.L("0.7i") - ELAS.L("1.3i");
stat_elas(r,"std error")      = se_elas(r);

display stat_elas;

*       Z test of H0:
*       Purchase probability's income elasticities are the same for individuals
*       with income no more than 70% of the sample income mean and individuals with
*       higer income than 130% of the sample income mean

*       To make it short, we write H0: ie_diff = 0
parameter       z_diff          "Z statistic based on H0: ie_diff = 0",
                pval_diff       "P value assuming H0: ie_diff = 0";

z_diff    = stat_elas("diff","estimator")/se_elas("diff");
pval_diff = 1 - errorf(abs(z_diff));

display z_diff,pval_diff;
if((pval_diff lt alpha/2),display "There is enough evidence to reject H0";
                    else display "There is not enough evidence to reject H0";);
```

From statistics reported we reject the null that two milk purchasing probability's income elasticities equal to each other.

```
----    492 PARAMETER stat_elas  Statistics of the regression
      estimator   std error
0.7i    0.349124    0.032434
1.3i    0.124925    0.021415
diff    0.224199    0.048492


----    509 PARAMETER z_diff               =     4.623377  Z statistic based on H0: ie_diff = 0
            PARAMETER pval_diff            =     0.000002  P value assuming H0: ie_diff = 0

----    510 There is enough evidence to reject H0
```

Next, we show an application of *logit* in a binary choice model:

4.2.3 Limited dependent variable mode example 4: binary choice, logit

Greene(1995) estimated a model of consumer behavior where he examined whether or not an individual had experienced a major negative derogatory report in his/her credit history. The file greene_credit_v4 contains information on the credit history of a sample of more than 1,000 individuals. The variables contained in this dataset include among others are the following:

Variable	Description
Majordrg	Number of major derogatory credit reports
Age	Age in yeas plus twelfths of a year
Inc_per	Per Capita yearly income (divided by $10,000)
Avgexp	Average monthly credit card expenditures
Ownrent	1 if person owns a home/0 he/she rents
Active	Number of active credit accounts

Table 3: Description of a Subset of Variables in greene_credit_v4

You would like to examine the determinants of whether a credit card holder experiences a derogatory credit report. You propose the following discrete choice model:

$$\Pr(Majordrg > 0) = g(Age, Age2, Inc\_Per, Avgexp, Ownrent),$$

Estimate the above model using the Logit model specification and maximum likelihood techniques. For your analysis you want to limit yourself to only those persons with active credit accounts. That is you need to delete observations for which Active = 0.

In part (a) of this example, we estimate a *logit* model in a binary choice model, report typical statistics; In part (b), we apply a likelihood ratio test to find out whether the naive model is true; After that, we calculate the elasticity impacts of a change in Age and Income on the probability of having a major derogatory report, then test whether these impacts are statistically 0; Finally in the last part, we test the above income elasticity is significantly different or not when we evaluate them at different income levels.

(a) Estimate your Logit model using maximum likelihood techniques. Provide the usual listing of estimated coefficients, parameter standard errors, and the unrestricted total sample log-likelihood function values.

```
$title The determinants of whether a credit card holder experiences a derogatory credit report
*       Part(a)
*       Logit model and mle
set     t               "Index of observations"         /1*1319/,
        a(t)            "Subset of t for observations in which 'active' is not equal to zero",
        i               "Index for BETA"                /"cons","age","age_sqr","inc_per","avgexp","ownrent"/;

alias (i,ii), (t,tt);

parameter       data(t,*)       "Source data";

$call gdxxrw greene_credit_v4_gams.xls par=data rng=A1:N1320 cdim=1 rdim=1 checkDate
$gdxin greene_credit_v4_gams.gdx
$loaddc data
$gdxin

a(t) = yes$(data(t,"active") > 0);

parameter       y(t)            "1 if number of major derogatory credit reports > 0",
                x(t,i)          "Explanatory variable i of observation t";

*       Number of major derogatory credit reports
```

```
y(a) = data(a,"majordrg");
*        Dependent variable, active credit accounts only
y(a) = 1$(y(a)>0);

*        x is the unrestricted matrix of exogenous variables when "active <> 0"
x(a,"cons") = 1;
*        Age in years
x(a,"age") = data(a,"age");
x(a,"age_sqr") = sqr(data(a,"age"));
*        Per capita yearly income (divided by $10,000)
x(a,"inc_per") = data(a,"inc_per");
*        Average monthly credit card expenditure
x(a,"avgexp") = data(a,"avgexp");
*        1 if person owns a home, 0 if rents
x(a,"ownrent") = data(a,"ownrent");

*        Now we have rhs matrix x: in total 6 exogenous variables
*        Log-Likelihood maximization
variable        BETA(i)          "Coefficients to be estimated",
                LOGLIK           "Log-Likelihood";

equation        obj_llf          "Objective for a unrestricted model";

obj_llf..       LOGLIK =e= sum(a(t), y(t)*log(exp(sum(i, x(t,i)*BETA(i)))/(1 + exp(sum(i, x(t,i)*BETA(i))))) +
                                     (1-y(t))*log(1-exp(sum(i,x(t,i)*BETA(i)))/(1 + exp(sum(i,x(t,i)*BETA(i))))));

model unres_mle  /obj_llf/;
solve unres_mle maximizing LOGLIK using nlp;

parameter       ll_ur           "Log-Likelihood level of the unrestricted model",
                beta_ur(i)      "Coefficients estimator in the unrestricted model";
ll_ur      = LOGLIK.L;
beta_ur(i) = BETA.L(i);

*        Report statistics of the unrestricted model
parameter       Hessian(i,ii)   "Hessian of log-likelihood wrt coefficients",
                hinv(i,ii)      "Inverse of ssehess",
                cov(i,ii)       "Covariance matrix of type k",
                alpha           "Type 1 error level"    /0.05/;

parameter       cov_ur(i,ii)    "Covariance matrix of the unrestricted case",
                stat_unres(i,*) "Statistics at the point";

*        Report Hessian matrix
$batinclude lsa unres_mle nlp "maximizing LOGLIK"
hessian(i,ii)   = LSA_D2F( obj_llf, BETA(i), BETA(ii) );

*        d2f is an upper triangular matrix
hessian(i,ii)   = hessian(i,ii)$(ord(ii) ge ord(i)) + hessian(ii,i)$(ord(ii) lt ord(i));

*       Find the inverse Hessian matrix
$batinclude arealinverter hessian i ii hinv
cov(i,ii)       = hinv(i,ii);

*        Record covariance matrix for future use
cov_ur(i,ii)    = cov(i,ii);

*       Report statistics
stat_unres(i, "estimator")  = BETA.L(i);
stat_unres(i, "std error")  = sqrt(cov(i,i));
stat_unres(i, "T value")    = BETA.L(i)/sqrt(cov(i,i));
*        Use the BETAREG function:
stat_unres(i, "P value")    = BETAREG( (card(t) - card(i))/
                                       (card(t) - card(i) + sqr(stat_unres(i, "T value"))),
                                       (card(t) - card(i))/2, 0.5 );
display stat_unres, ll_ur;
```

We list the regression statistics and the log-likelihood function level as follows:

```
----    257 PARAMETER stat_unres  Statistics at the point
          estimator   std error     T value     P value
cons      -3.922503    0.826321   -4.746949    0.000002
age        0.145988    0.044423    3.286297    0.001042
age_sqr   -0.001603    0.000569   -2.818624    0.004895
inc_per    0.097694    0.054994    1.776467    0.075887
avgexp    -0.001359    0.000379   -3.584914    0.000350
```

```
ownrent    -0.435491     0.157957    -2.757014     0.005914

----    257 PARAMETER ll_ur                =  -580.224888
                      Log-Likelihood level of the unrestricted model
```

(b) Undertake a likelihood ratio test of the statistical significance of this model relative to the naive model.

Here the likelihood ratio test is similar to that in the *probit* example with *homoscedastic* errors, we test whether the fall of log-likelihood from the level of the unrestricted model to the "naive" one is larger enough to check the null: naive model (restricted) is true.

```
*        Part (b)
*        In the so called naive model, intercept term x=1 is the only exogeneous variable
set             n(i)                "Index of the intercept term"       /"cons"/;

alias (n,nn);

*        We already have MLE with 6 explanatory variables in part (a)
*        Now move to restricted case:
equation        obj_res             "Objective function in restricted MLE";

obj_res..       LOGLIK =e= sum(a(t), y(t)*log(  exp(sum(n, x(t,n)*BETA(n)))/(1 + exp(sum(n, x(t,n)*BETA(n))))) +
                                (1-y(t))*log(1-exp(sum(n, x(t,n)*BETA(n)))/(1 + exp(sum(n, x(t,n)*BETA(n))))));

model res_mle  /obj_res/;

*        Initialize BETA
BETA.L(n) = 0.5;
solve res_mle maximizing LOGLIK using nlp;

parameter       stat_res(i,*)           "Report of statistics in the restricted model",
                ll_res                  "Log-Likelihood level of the restricted model";
ll_res = LOGLIK.L;

*        Report Hessian matrix
$batinclude lsa res_mle nlp "maximizing LOGLIK"
hessian(n,nn)   = LSA_D2F( obj_res, BETA(n), BETA(nn) );

*        Find inverse matrix of the Hessian matrix
$batinclude arealinverter Hessian n nn hinv
cov(n, nn) = hinv(n, nn);

stat_res(n, "estimator")  = BETA.L(n);
stat_res(n, "std error")  = sqrt(cov(n,n));
stat_res(n, "T value")    = BETA.L(n)/sqrt(cov(n,n));
*        Use the BETAREG function:
stat_res(n, "P value")    = BETAREG( (card(t) - card(n))/
                                     (card(t) - card(n) + sqr(stat_res(n, "T value"))),
                                     (card(t) - card(n))/2, 0.5 );
display stat_res;

*        LR test against H0: restricted model is true

parameter       lr_res                  "Likelihood ratio test statistic based on H0: restricted model is true",
                pval_res                "P value for lr statistic";

lr_res   = 2*(ll_ur - ll_res);
pval_res = 1 - gammareg(lr_res/2,(card(i) - card(n))/2);

display lr_res;

if((pval_res lt alpha), display "There is enough evidence to reject H0";
                  else  display "There is not enough evidence to reject H0";);
```

From the LR test statistics, we conclude that there is enough evidence to reject the null hypothesis: naive model is true.

```
----    534 PARAMETER stat_res  Report of statistics in the restricted model
       estimator   std error     T value
cons   -1.177764    0.071064  -16.573326


----    544 PARAMETER lr_res               =    40.274125
           Likelihood ratio test statistic based on H0: restricted model is true

----    546 There is enough evidence to reject H0
```

(c) Based on your estimated Logit model, what are the elasticity impacts of a change in Age and Inc_per (individually) on the probability of having a major derogatory report? Are these effects different from 0 at the mean of the data?

In this part, we rely on a *z-test* to check whether these impacts are 0 or not:

```
*       Part(c)
parameter       x_bar(i)                "Exogeneous variable mean";
x_bar(i) = sum(a,x(a,i))/card(a);

*       The elasticity impacts of a change in "Age" on the probability of having a major derogatory report
variable        AGE_ELS                 "The elasticity impacts of a change in age on having derogatory report";

equation        def_age                 "Definition of AGE_ELS";

def_age..       AGE_ELS =e= exp(sum(i, x_bar(i)*BETA(i)))/(1 + exp(sum(i, x_bar(i)*BETA(i))))*
                            (1 - exp(sum(i, x_bar(i)*BETA(i)))/(1 + exp(sum(i, x_bar(i)*BETA(i)))))*
                            (BETA("age") + 2*BETA("age_sqr")*x_bar("age"))*(1 + exp(sum(i,x_bar(i)*BETA(i))))*
                            x_bar("age")/exp(sum(i,x_bar(i)*BETA(i)));

model age /def_age/;
BETA.FX(i) = beta_ur(i);
solve age using mcp;

set             r_age                   "Row index for the elasticity impacts of a change in age on having bad report"
                                         /"Age_majordrg els"/;

parameter       stat_age(r_age,*)       "Statistics of the regression",
                grad_age(i)             "Gradients of age elasticity function";

*       Report Jacobian matrix
$batinclude lsa age mcp

grad_age(i) = LSA_DF( def_age, BETA(i) );

*      Report estimator and standard error
stat_age(r_age,"Estimator") = AGE_ELS.L;
stat_age(r_age,"Std Error") = sqrt(sum((i,ii),grad_age(i)*cov_ur(i,ii)*grad_age(ii)));

display stat_age;

*       Testing age_majordrg elasticity
parameter       z_age                   "Z statistic for H0: age_majordrg elasticity = 0",
                pval_age                "P value";

z_age = AGE_ELS.L/stat_age("Age_majordrg els","Std Error");
pval_age = 1 - errorf(abs(z_age));

display z_age;
if((pval_age lt alpha/2), display "There is enough evidence to reject H0";
                    else  display "There is not enough evidence to reject H0";);

*       The elasticity impacts of a change in Inc_per on the probability of having a major derogatory report
variable        INC_ELS                 "The elasticity impacts of a change in inc on having derogatory report";

equation        def_inc                 "Definition of INC_ELS";

def_inc..       INC_ELS =e= exp(sum(i, x_bar(i)*BETA(i)))/(1 + exp(sum(i, x_bar(i)*BETA(i))))*
                            (1 - exp(sum(i, x_bar(i)*BETA(i)))/(1 + exp(sum(i, x_bar(i)*BETA(i)))))*
```

```
                          BETA("inc_per")*(1 + exp(sum(i,x_bar(i)*BETA(i))))*
                          x_bar("inc_per")/exp(sum(i,x_bar(i)*BETA(i)));

model inc /def_inc/;
BETA.FX(i) = beta_ur(i);
solve inc using mcp;

set             r_inc                   "Row index for the elasticity impacts of a change in inc on having bad report"
                                        /"Inc_majordrg els"/;

parameter       stat_inc(r_inc,*)       "Statistics of the regression",
                grad_inc(i)             "Gradients of inc elasticity function";

*       Report Jacobian matrix
$batinclude lsa inc mcp

grad_inc(i) = LSA_DF( def_inc, BETA(i) );

*       Report estimator and standard error
stat_inc(r_inc,"Estimator") = INC_ELS.L;
stat_inc(r_inc,"Std Error") = sqrt(sum((i,ii),grad_inc(i)*cov_ur(i,ii)*grad_inc(ii)));

display stat_inc;

*       Testing if inc_majordrg elasticity = 0
parameter       z_inc                   "Z statistic for H0: inc_majordrg elasticity = 0",
                pval_inc                "P value based on H0";

z_inc    = INC_ELS.L/stat_inc("Inc_majordrg els","Std Error");
pval_inc = 1 - errorf(abs(z_inc));

display z_inc;
if((pval_inc lt alpha/2), display "There is enough evidence to reject H0";
                   else  display "There is not enough evidence to reject H0";);
```

According to the regular statistics, we find the elasticity impacts of a change in age is statistically different from 0; However, we don't have enough evidence to reject the null hypothesis that the elasticity impacts of a change in income is 0.

```
----    723 PARAMETER stat_age  Statistics of the regression
                  estimator   std error
Age_majordrg els    1.001736    0.257581

----    732 PARAMETER z_age               =     3.889011
        Z statistic for H0: age_majordrg elasticity = 0

----    733 There is enough evidence to reject H0

*--------------------------------------------------------------------------

----    897 PARAMETER stat_inc  Statistics of the regression
                  estimator   std error
Inc_majordrg els    0.162419    0.091643

----    906 PARAMETER z_inc               =     1.772290
        Z statistic for H0: inc_majordrg elasticity = 0

----    908 There is not enough evidence to reject H0
```

(d) Test the null hypothesis that the above income elasticity is significantly different when you compare the income elasticity at 25% below the mean income value vs. when income is 25% above the mean income value.

This part is similar to the previous one. We first evaluate income elasticities at different income level, find gradients of these elasticities with respect to estimated coefficients before constructing a *z-test* to check whether income levels affect income elasticities or not:

```
*        Part (d)
*        Test H0:
*        Income elasticity is different
*        when comparing the income elasticity at 25% below the mean income
*        vs. when income is 25% above the mean income value
parameter        x_75bar(i)        "Mean of exogenous variable with 75% income level",
                 x_125bar(i)       "Mean of exogenous variable with 125% income level";

x_75bar(i) = x_bar(i);
x_75bar("inc_per") = 0.75*x_bar("inc_per");

x_125bar(i) = x_bar(i);
x_125bar("inc_per") = 1.25*x_bar("inc_per");

*        The difference of elasticity impacts of a change in Inc_per (75% vs 125% income levle)
*        on the probability of having a major derogatory report
variable
                 DIFF_ELS        "Elasticity difference of a change in income levels on having derogatory report";

equation
                 def_diff        "Definition of DIFF_ELS";

def_diff..       DIFF_ELS =e= exp(sum(i, x_75bar(i)*BETA(i)))/(1 + exp(sum(i, x_75bar(i)*BETA(i))))*
                              (1 - exp(sum(i, x_75bar(i)*BETA(i)))/(1 + exp(sum(i, x_75bar(i)*BETA(i)))))*
                              BETA("inc_per")*(1 + exp(sum(i,x_75bar(i)*BETA(i))))*
                              x_75bar("inc_per")/exp(sum(i,x_75bar(i)*BETA(i))) -
                              (exp(sum(i, x_125bar(i)*BETA(i)))/(1 + exp(sum(i, x_125bar(i)*BETA(i))))*
                              (1 - exp(sum(i, x_125bar(i)*BETA(i)))/(1 + exp(sum(i, x_125bar(i)*BETA(i)))))*
                              BETA("inc_per")*(1 + exp(sum(i,x_125bar(i)*BETA(i))))*
                              x_125bar("inc_per")/exp(sum(i,x_125bar(i)*BETA(i))));

model diff /def_diff/;
BETA.FX(i) = beta_ur(i);
solve diff using mcp;

set              r_diff          "Row index for the elasticity impacts of a change in diff on having bad report"
                                 /"Diff_majordrg els"/;

parameter        stat_diff(r_diff,*)     "Statistics of the regression",
                 grad_diff(i)            "Gradients of diff elasticity function";

*        Report Jacobian matrix
$batinclude lsa diff mcp

grad_diff(i) = LSA_DF( def_diff, BETA(i) );

*       Report estimator and standard error
stat_diff(r_diff,"Estimator") = DIFF_ELS.L;
stat_diff(r_diff,"Std Error") = sqrt(sum((i,ii),grad_diff(i)*cov_ur(i,ii)*grad_diff(ii)));

display stat_diff;

*        Testing if diff_majordrg elasticity = 0
parameter        z_diff                  "Z statistic for H0: diff_majordrg elasticity = 0",
                 pval_diff               "P value based on H0";

z_diff = DIFF_ELS.L/stat_diff("Diff_majordrg els","Std Error");
pval_diff = 1 - errorf(abs(z_diff));

display z_diff;
if((pval_diff lt alpha/2), display "There is enough evidence to reject H0";
                   else    display "There is not enough evidence to reject H0";);
```

Testing results show that we can not rule out the null hypothesis saying that income elasticity is not significantly different when evaluated at 25% blow the mean income *vs*. at 25% above the mean.

```
----   1089 PARAMETER stat_diff  Statistics of the regression
                    estimator   std error
Diff_majordrg els   -0.077405    0.041548

----   1099 PARAMETER z_diff              =    -1.863001
           Z statistic for H0: diff_majordrg elasticity = 0
```

```
----    1101 There is not enough evidence to reject H0
```

Next, we discuss multiple choice models in which dependent variables y_t are allowed to have more than two realizations.

4.3 Multiple choice model

The binary choice model can be generalized to one with more than two alternatives. In the following examples, We examine two types of choice sets, ordered and unordered. For instance, choice of transportation is an unordered choice model and credit rating to corporate bonds is an ordered one.

We start with the unordered model first: Two types of estimation methods are again commonly used, *logit* and *probit*. Due to the need to evaluate multiple integrals of the normal distribution, the *logit* model becomes more popular. We also put some code in the appendix to show how to use GAMS in simulating multivariate normal CDF (a Geweke, Hajivassiliou, Keane (GHK) simulator).

In a popular *multinomial logit* model, for $j, k = 0, 1, \ldots, K$, choice specific coefficients β_j of exogeneous data x_t which stand for characteristics of individual t is an $I \times 1$ vector, and we generalize the *logit* model equation (4.2.1) into

$$\Pr(y_t = j|x_t) = \frac{\exp(x_t'\beta_j)}{\sum_{k=0}^{K} \exp(x_t'\beta_k)}. \tag{4.3.1}$$

when domain of dependent variable y_t is now {0, 1, . . . , K}.

To fix the problem that coefficients in the model are identifiable up to normalization, a convenient normalization is setting coefficients of one alternative, say $j = 0$, to zero to find a baseline choice. Thus,

$$\begin{aligned} \Pr(y_t = 0|x_t) &= 1/\left[1 + \sum_{k=1}^{K} \exp(x_t'\beta_k)\right], \\ \Pr(y_t = j|x_t) &= \exp(x_t'\beta_j)/\left[1 + \sum_{k=1}^{K} \exp(x_t'\beta_k)\right], \quad \text{for } j \neq 0. \end{aligned} \tag{4.3.2}$$

We then define relative risk ratios as follows:

$$\log\left(\frac{\Pr(y_t = j|x_t)}{\Pr(y_t = 0|x_t)}\right) = x_t'\beta_j, \tag{4.3.3}$$

Then, i^{th} element of coefficients vector of choice j, β_j^i can be interpreted as the marginal effect of x_t^i on the log odds ratio of alternative j to the baseline alternative.

A closely related technique called *conditional logit* which is appropriate when the choice among alternatives is modeled as a function of the attributes of the alternatives, rather than the characteristics of the individual making the choice. For details about how the *conditional logit* model is compared to the

multinomial logit model, see Hoffman and Duncan (1988). One obvious difference between these two is *multinomial logit* estimates $J-1$ sets of coefficients (β_j), while *conditional logit* model only estimates a single coefficient for each explanatory variable z_j, so the impact of a variable on the choice probabilities derives from the difference in its value across alternatives.

The following is a *conditional logit* problem in which alternatives are unordered:

4.3.1 Limited dependent variable model example 5: conditional logit, unordered

As a consultant under contract with the California Charter Boat Association, you are interested in examining what determines the mode of fishing used by saltwater anglers. As such, you decide to use the data a 1989 survey of recreational anglers to provide information concerning their most recent saltwater angling experiences, made available by Herriges and Kling (1999). You decide that you would first like to use the data in a discrete choice analysis of fishing mode. Given that there are 4 fishing modes (beach, pier, private boat, and charter boat) you decide that you would like to estimate a logit model of fishing mode choice. The specific model you decide to estimate is the following:

$$P_{ij} = \Pr(y_i = j) = \frac{\exp(\beta_P Cost_{ij} + \beta_C C\_rate_{ij})}{\sum_{r=1}^{4} \exp(\beta_P Cost_{ir} + \beta_c C\_rate_{ir})} \tag{4.3.4}$$

Where β_P, β_C are coefficients to estimate, i identifies the angler, j pertains to a particular fishing mode, $Cost_r$ is cost for the r^{th} mode and C_rate_r is the catch rate for the r^{th} mode. In addition you decide to add choice specific intercepts and household income as regressors in the model. In implementing these intercepts and household income you decide to use the Beach option as the omitted category.

The dataset New_fish_file contains the data necessary to estimate the model. Table 4 is used to show the records from 2 anglers contained in the above dataset. Note there are four records for each survey respondent.

HHID	MODE_ID	CHOICE	COST	C_RATE	INCOME
1	1	0	1.5793	0.0678	0.7083
1	2	1	1.5793	0.0503	0.7083
1	3	0	1.5793	0.2601	0.7083
1	4	0	1.8293	0.5391	0.7083
2	1	0	0.1511	0.104	0.125
2	2	0	0.1551	0.0451	0.125
2	3	0	0.1053	0.1574	0.125
2	4	1	0.3453	0.4671	0.125

Table 4: Sample of Observations in the New_fish_file Dataset

Variables	Description
	Identification of type of fishing mode:
mode_id	1 = Beach
	2 = Pier
	3 = Private Boat
	4 = Charter Boat
choice	Identifies fishing mode actually used (0/1)
cost	Daily cost of fishing for each mode ($100)
c_rate	Catch Rate (#)
income	Monthly income ($10,000)

Table 5: Description of variables in the data file

In this question, we first estimates a conditional logit model in part (a), then apply Hausman's Independence of Irrelevant Alternatives test (IIA) in part (b) to check whether the odds ratio of two fishing modes are independent from the presence of other alternative.

(a) Estimate the coefficients of the above logit model using maximum likelihood techniques. Report the usual estimation results. Test the null hypothesis that household income impacts fishing mode choice.

First, we observe that explanatory variables we have are attributes of alternatives instead of decision making individuals. It is reasonable to set up this problem as a *conditional logit* model and estimate it through MLE. Keep in mind we need to test whether household income matters in mode choice, we thus need to set up an unrestricted model which includes interaction terms of income and fishing mode choice and a restricted one which does not.

```
$title conditional logit model of fishing mode choice
set             t               "Index of observations"                 /1*4728/,
                i               "Index for source data"
                                /"cost","c_rate","dum_pier","pier_inc","dum_pboat","pboat_inc","dum_cboat","cboat_inc"/;
alias (i,ii),(t,tt);

parameter       data(t,*)       "Source data";

$call gdxxrw new_fish_file_gams.xls par=data rng=A1:G4729 cdim=1 rdim=1 checkDate
$gdxin new_fish_file_gams.gdx
$loaddc data
$gdxin

parameter       y(t)            "Dependent variable",
                z(t,i)          "Right hand side variables";

*       Load fishing data
y(t)          = data(t,"choice");
z(t,"cost")   = data(t,"cost");
z(t,"c_rate") = data(t,"c_rate");

*       Dummy variable = 1 if mode_id = 2, indicating
*       type of fishing mode: pier
z(t,"dum_pier") = 1$(data(t,"mode_id") = 2);
z(t,"pier_inc") = z(t,"dum_pier")*data(t,"income");
```

```
*       Dummy variable = 1 if mode_id = 3, indicating
*       type of fishing mode: private boat
z(t,"dum_pboat") = 1$(data(t,"mode_id") = 3);
z(t,"pboat_inc") = z(t,"dum_pboat")*data(t,"income");

*       Dummy variable = 1 if mode_id = 4, indicating
*       type of fishing mode: charter boat
z(t,"dum_cboat") = 1$(data(t,"mode_id") = 4);
z(t,"cboat_inc") = z(t,"dum_cboat")*data(t,"income");
```

Given settings in part (a), we construct log-likelihood function based on equation (4.3.4), then report statistics after the estimation for the unrestricted model first.

```
*       Part(a)
*       Conditional Logit Estimation for an unordered, multiple choice model
*       Through maximum likelihood estimation
set             u(t)            "Subset of t, household id"     /1*1182/,
                m               "Four fishing modes"            /beach,pier,p_boat,c_boat/;

alias (m,mm);

parameter       z_mode(t,i,m)   "Data when fishing mode is m",
                dum(t,m)        "Dummy variable when fishing mode m is chosen";

*       Choose data from the source
loop( (u,t,m), if((ord(u)-1 = (ord(t)-ord(m))/4), z_mode(u,i,m) = z(t,i)); );

*       Choose dummy variable value from the choice
loop( (u,t,m), if((ord(u)-1 = (ord(t)-ord(m))/4), dum(u,m) = y(t)); );

variable        COEF(i)         "Coefficients to be estimated in the unrestricted model",
                LOGLIK          "Log-Likelihood function for the conditional logit model";

equation        obj_llf         "Objective in the maximum likelihood estimation";

obj_llf..       LOGLIK =e= sum(u, sum(m, dum(u,m)*(sum(i, z_mode(u,i,m)*COEF(i)))) -
                              sum(m, dum(u,m))*log(sum(m, exp(sum(i, z_mode(u,i,m)*COEF(i))))));

model llf /obj_llf/;
solve llf maximizing LOGLIK using nlp;

display COEF.L;

*       Report estimation results for future use
parameter       loglik_unres    "Optimal value of log-likelihood function in a unrestricted model"
                coef_unres(i)   "Coefficients in the unrestricted model";
loglik_unres  = LOGLIK.L;
coef_unres(i) = COEF.L(i);

*       Report statistics of the unrestricted model
parameter       Jacobian(t,i)           "Jacobian of constraints wrt coefficients",
                Hessian(i,ii)           "Hessian of log-likelihood wrt coefficients",
                hinv(i,ii)              "Inverse of ssehess",
                cov(i,ii)               "Covariance matrix of type k",
                stat_unres(i, *)        "Statistics at the point";

*       Report Hessian matrix
$batinclude lsa llf nlp "maximizing LOGLIK"
hessian(i,ii)   = LSA_D2F( obj_llf, COEF(i), COEF(ii) );

*       d2f is an upper triangular matrix
hessian(i,ii) = hessian(i,ii)$(ord(ii) ge ord(i)) + hessian(ii,i)$(ord(ii) lt ord(i));

*       Find inverse of Hessian
$batinclude arealinverter Hessian i ii hinv
cov(i,ii)       = hinv(i,ii);

parameter        cov_unres(i,ii)         "Covariance matrix of the unrestricted model";
cov_unres(i,ii) = cov(i,ii);

*       Report statistics
stat_unres(i, "estimator")  = COEF.L(i);
stat_unres(i, "std error")  = sqrt(cov(i,i));
stat_unres(i, "T value")    = COEF.L(i)/sqrt(cov(i,i));
```

```
*       Use the BETAREG function:
stat_unres(i, "P value")    = BETAREG( (card(t) - card(i))/
                                       (card(t) - card(i) + sqr(stat_unres(i, "T value"))),
                                       (card(t) - card(i))/2, 0.5 );
display stat_unres;
```

We find statistics of the unrestricted model as follows:

```
----    323 PARAMETER stat_unres  Statistics at the point
            estimator   std error     T value     P value
cost        -2.511651    0.173168  -14.504112
c_rate       0.357748    0.109794    3.258351    0.001129
dum_pier     0.777925    0.220494    3.528108    0.000423
pier_inc    -1.275739    0.506396   -2.519253    0.011793
dum_pboat    0.527149    0.222791    2.366117    0.018016
pboat_inc    0.894415    0.500669    1.786439    0.074092
dum_cboat    1.694302    0.224057    7.561923
cboat_inc   -0.332911    0.503407   -0.661316    0.508442
```

Next, we use *likelihood ratio test* to check whether household income affect fishing mode choice or not based on statistics of both unrestricted and restricted models. We set up a restricted model as follows before the *LR test*:

```
*       Test of H0: household income does not affect fishing mode choice

*       Equivalently, we test whether the interaction terms of income and mode choice
*       are jointly zero

*       In order to test this, we first find coefficients from the "restricted" model
*       Redefine exogeneous variable z in setting interaction terms to zero
set             r(i)                    "Restricted variables"
                                        /"cost","c_rate","dum_pier","dum_pboat","dum_cboat"/;

alias (r, rr);

parameter        z_res(t,i,m)            "Data when fishing mode = beach";

*       Define restricted data
loop( (r,i,m), if((sameas (r,i)),z_res(u,r,m) = z_mode(u,i,m);); );

equation        obj_llf_res             "Objective in the maximum likelihood estimation";

obj_llf_res..        LOGLIK =e= sum(u, sum(m, dum(u,m)*(sum(i, z_res(u,i,m)*COEF(i)))) -
                                       sum(m, dum(u,m))*log(sum(m, exp(sum(i, z_res(u,i,m)*COEF(i))))));

model llf_res /obj_llf_res/;
solve llf_res maximizing LOGLIK using nlp;

parameter       loglik_res              "Optimal value of log-likelihood function in a restricted model",
                coef_res(i)             "Coefficients in the restricted model";
loglik_res  = LOGLIK.L;
coef_res(r) = COEF.L(r);

*       Report statistics of the unrestricted model
parameter       stat_res(i,*)           "Statistics at the point";

*       Report Hessian matrix
$batinclude lsa llf_res nlp "maximizing LOGLIK"
hessian(r,rr) = LSA_D2F( obj_llf_res, COEF(r), COEF(rr) );

*       d2f is an upper triangular matrix
hessian(r,rr) = hessian(r,rr)$(ord(rr) ge ord(r)) + hessian(rr,r)$(ord(rr) lt ord(r));

*       Find inverse of the Hessian
$batinclude arealinverter Hessian r rr hinv
cov(r,rr)     = hinv(r,rr);

parameter       cov_res(i,ii)           "Covariance matrix of the restricted model";
```

```
cov_res(r,rr) = cov(r,rr);

*        Report statistics
stat_res(r, "estimator")  = COEF.L(r);
stat_res(r, "std error")  = sqrt(cov(r,r));
stat_res(r, "T value")    = COEF.L(r)/sqrt(cov(r,r));
*        Use the BETAREG function:
stat_res(r, "P value")    = BETAREG( (card(t) - card(r))/
                                     (card(t) - card(r) + sqr(stat_res(r, "T value"))),
                                     (card(t) - card(r))/2, 0.5 );
display stat_res;

*        Log-Likelihood Ratio Test
parameter       lr_income_no_impact     "Likelihood ratio test statistic based on H0",
                df_income_no_impact     "Degrees of freedom of the LR test",
                pval_income_no_impact   "P value abased on H0",
                alpha                   "Type 1 error level"  /0.05/;

lr_income_no_impact   = 2*(loglik_unres - loglik_res);
df_income_no_impact   = card(i) - card(r);
pval_income_no_impact = 1- gammareg(lr_income_no_impact/2, df_income_no_impact/2);

display lr_income_no_impact, pval_income_no_impact;

if ( pval_income_no_impact lt alpha, display "Reject H0: household income does not affect fishing mode choice";
                              else  display "Can not reject H0: household income does not affect fishing mode choice"; );
```

Here are the statistics for the restricted models, and the likelihood ratio test statistic tells us that we are confident in rejecting the null which assumes that household income does not affect fishing mode choice:

```
----    607 PARAMETER stat_res  Statistics at the point
            estimator   std error     T value     P value
cost        -2.478947    0.170440  -14.544387
c_rate       0.377145    0.109992    3.428835    0.000611
dum_pier     0.307033    0.114573    2.679791    0.007392
dum_pboat    0.871248    0.114041    7.639790
dum_cboat    1.498821    0.132944   11.274064

----    619 PARAMETER lr_income_no_impact  =    31.292471  Likelihood ratio test statistic
            PARAMETER pval_income_no_impact =  7.376553E-7  P value

----    621 Reject H0: household income does not affect fishing mode choice
```

(b) Undertake a Hausman IIA test to determine if the odds ratios for the beach/pier and private boat/pier are independent from the presence of the charter boat alternative. Explain how you undertook this hypothesis test.

From equations (4.3.4), we find that in a *conditional logit* model, the log odds-ratios between two alternatives are only expressed as a function of the coefficients of the two alternatives, but not of those for any other choice.

In other words, we assume that in this type of model, the log odds-ratios have the property called *Independence of Irrelevant Alternatives* (IIA). However, violation of IIA in discrete choice problems is common: For example, the *Blue-Bus/Red-bus paradox* discussed by Debreu (1960).

Hausman and McFadden (1984) proposes an *IIA test* on a subset of alternatives. It starts with estimating logit model twice: one on full set of alternatives (COEF(i) in the code as resulting coefficients), one on a specified subset of alternatives (COEF(b)).

If IIA holds, these two sets of estimates should not be statistically different. Thus, the *Chi-square* distributed statistic

$$(\beta_{full} - \beta_{sub})' \left(\Omega_{full} - \Omega_{sub}\right)^{-1} (\beta_{full} - \beta_{sub}),$$

should not exceed its threshold level, in which Ω_{full} and Ω_{sub} are estimated covariance matrix of full set and a subset of alternatives estimates.

We can then solve part (b) as the follows:

```
*       Part(b)
*       Using Hausman IIA test to determine if the odds ratios for the
*       beach/pier and private/pier are independent from the
*       presence of the charter boat alternative

*       Use restricted conditional logit (charter boat is not available) to
*       find covariance matrix in the unrestricted case first
set             b(i)                    "Subset of i when no charter boat choice available"
                                        /"cost","c_rate","dum_pier","pier_inc","dum_pboat","pboat_inc"/,
                v(t)                    "Subset of t when fishing mode is not charter boat",
                n(m)                    "Subset of m when 'c_boat' is not included"       /beach,pier,p_boat/;

v(t)=yes$(z(t,"dum_cboat") = 0);

alias (b,bb);

*       Now we do conditional logit based on set b(i),i.e.,
*       when no charter boat available
equation        obj_no_cboat            "Objective in the maximum likelihood estimation";

obj_no_cboat..   LOGLIK =e= sum(u, sum(n, dum(u,n)*(sum(b, z_mode(u,b,n)*COEF(b)))) -
                                sum(n, dum(u,n))*log(sum(n, exp(sum(b, z_mode(u,b,n)*COEF(b))))));

model no_cboat /obj_no_cboat/;
solve no_cboat maximizing LOGLIK using nlp;

parameter        loglik_no_cboat         "Optimal value of log-likelihood in the model with no charter boat choice",
                 coef_no_cboat(i)        "Coefficients to be estimated in the model with no charter boat choice";
loglik_no_cboat  = LOGLIK.L;
coef_no_cboat(b) = COEF.L(b);

*       Report statistics of the unrestricted model
parameter        stat_no_cboat(b,*)      "Statistics at the point";

*       Report Hessian matrix
$batinclude lsa no_cboat nlp "maximizing LOGLIK"
hessian(b,bb)   = LSA_D2F( obj_no_cboat, COEF(b), COEF(bb) );

*       d2f is an upper triangular matrix
hessian(b,bb) = hessian(b,bb)$(ord(bb) ge ord(b)) + hessian(bb,b)$(ord(bb) lt ord(b));

*       Find inverse of the Hessian
$batinclude arealinverter Hessian b bb hinv
cov(b,bb)         = hinv(b,bb);

*       Store covariance matrix for future use
parameter        cov_no_cboat(i,ii)      "Covariance matrix of the model when no charter boat available";
cov_no_cboat(b,bb) = cov(b,bb);

*       Report statistics
stat_no_cboat(b, "estimator")  = COEF.L(b);
stat_no_cboat(b, "std error")  = sqrt(cov(b,b));
stat_no_cboat(b, "T value")    = COEF.L(b)/sqrt(cov(b,b));
*       Use the BETAREG function:
stat_no_cboat(b, "P value")    = BETAREG( (card(t) - card(b))/
                                          (card(t) - card(b) + sqr(stat_no_cboat(b, "T value"))),
                                          (card(t) - card(b))/2, 0.5 );
display stat_no_cboat;

*       Test if H0:
*       The odds ratios for the beach/pier and private/pier are independent from the
*       presence of the charter boat alternatives

parameter       df_iia                  "Degrees of freedom in the IIA test",
                para_diff(b)            "Parameter difference between the unrestricted and no_cboat cases",
```

```
                var_diff(b,bb)          "Variance covariance difference",
                inv_var_diff(b,bb)      "Inverse of variance difference between the unrestricted and no_cboat cases",
                chi2_iia                "Chi-square statistic of IIA test",
                Pval_iia                "P value based on H0",
                var_res(i,ii)           "Part of cov matrix of unrestricted model when no chart boat choice available";

var_res(b(i),bb(ii)) = cov(i,ii)$(sameas(b,i) and sameas(bb,ii));

*       Degrees of freedom equals to the number of coefficients in the restricted model
df_iia         = card(b);
*       Find difference between coefficients estimations
para_diff(b)   = coef_no_cboat(b) - coef_unres(b);
*       Chi-square statistic
var_diff(b,bb) = cov_no_cboat(b,bb) - cov_res(b,bb);

*       Find the inverse of variance difference matrix
$batinclude arealinverter var_diff b bb inv_var_diff

chi2_iia = sum((b,bb), para_diff(b)*inv_var_diff(b,bb)*para_diff(bb));
pval_iia = 1- gammareg(chi2_iia/2,df_iia/2);

display "Chi-square stat. (H0: charter boat option does not matter):",chi2_iia;
display "Prob LR stat. Assum. H0:", pval_iia;

if ( pval_iia lt alpha,
          display "There is, therefore, enough evidence to reject H0: charter boat option does not matter";
  else    display "There is, therefore, not enough evidence to reject H0: charter boat option does not matter"; );
```

We conclude that we have enough evidence to reject the null hypothesis, i.e., models with/without charter boat are statistically different.

```
----    907 PARAMETER stat_no_cboat  Statistics at the point
            estimator   std error     T value     P value
cost        -3.177694    0.272638  -11.655344
c_rate       1.324504    0.592517    2.235385    0.025439
dum_pier     0.822007    0.220770    3.723359    0.000199
pier_inc    -1.227433    0.494161   -2.483872    0.013031
dum_pboat    0.772383    0.247309    3.123153    0.001800
pboat_inc    0.231494    0.574370    0.403040    0.686937


----   1015 Chi-square stat. (H0: charter boat option does not matter):
            PARAMETER chi2_iia             =    18.181209  Chi-square statistic of IIA test

----   1016 Prob LR stat. Assum. H0:
            PARAMETER Pval_iia             =     0.005795  P value

There is, therefore, enough evidence to reject H0: charter boat option does not matter
```

In some other situations, we have a natural ordering of choice outcomes. For example, suppose there is a normally distributed latent error term ϵ_t as in the following example, two threshold parameters μ_p and μ_{pb} ($\mu_p \leq \mu_{pb}$) set bounds for latent choice variable $choice^*$:

4.3.2 Limited dependent variable model example 6: multiple choice model, ordered probit

In the fishing mode choice model, after you submitted your initial report, the management of the California Charter Boat Association pointed out that the four fishing modes could be ranked from a fishing quality perspective with beach being the least productive and charter boat being the most productive (i.e., the rankings are shown in the above table). It was decided that the estimation methodology should be changed taking into account this exogenous fishing

mode ordering. As such, you decide to estimate an ordered probit model of fishing mode choice. Table 6 provides a summary of the variables contained in this data set that you would like to use in the probit model.

Variables	Description	Units
	Identification of type of fishing mode actually chosen (dependent variable):	
choice	1 = Beach 2 = Pier 3 = Private Boat 4 = Charter Boat	#
day_cost	Daily cost of fishing for mode actually used	$100
ctchrate	Hourly fishing catch rate for mode actually used	#
income	Monthly income	$10,000

Table 6: Description of variables in the data file

(a) Using the above data, estimate an ordered probit model of fishing mode choice which can be represented by the following latent variable regression:

$$\text{Choice}^* = X\beta + \epsilon_t, \tag{4.3.5}$$

where $\epsilon_t \sim N(0,1)$. Here Choice* indicates the net utility obtained from using a particular fishing mode. The relationship between the above latent variable and observed fishing modes (Choice) can be represented via the following:

$$\text{Choice} = \begin{cases} \text{Beach} & \text{if Choice}^* \leq 0 \\ \text{Pier} & \text{if } 0 < \text{Choice}^* \leq \mu_p \\ \text{Private Boat} & \text{if } \mu_p < \text{Choice}^* \leq \mu_{pb} \\ \text{Charter Boat} & \text{if } \mu_{pb} < \text{Choice}^* , \end{cases}$$

The exogenous variable matrix X is composed of the following exogenous variables: Intercept,Day_Cost, CtchRate and Mnth_Inc. Present the typical regression results not only in terms of the estimated β coefficients but also with respect to the estimated μ_js, here μ_j is a threshold parameter vector that should be estimated along with coefficients vector β.

In this ordered *probit* model, given equation (4.3.5), and the normally distributed error term, we have

$$\Pr(choice = 1|X) = \Pr(choice* \leq 0|X) = \Pr(\epsilon_t \leq -X\beta) = \Phi(-X\beta),$$

Similarly,

$$\begin{aligned}\Pr(choice = 2|X) &= \Phi(\mu_p - X\beta) - \Phi(-X\beta),\\ \Pr(choice = 3|X) &= \Phi(\mu_{pb} - X\beta) - \Phi(\mu_p - X\beta),\\ \Pr(choice = 4|X) &= 1 - \Phi(\mu_{pb} - X\beta);\end{aligned}$$

Notice that the marginal effect of a change in X_i is not β_i (i indexes explanatory variables), for instance,

$$\frac{\partial \Pr(choice = 1|X)}{\partial(X)} = -\phi(-X\beta)\beta.$$

So one must be careful in interpreting the coefficients as well as their significance in this model.

And as an ordered model, we don't need multiple integration to find log-likelihood function of this *probit* as

$$L = \sum_t \ln\left[\Pr(choice_t = j|X_t)\right] = \sum_t \sum_j^K dum_{tj} \ln\left[\Pr(choice_t = j|X_t)\right],$$

where dummy variable dum_{tj} equals 1 when alternative j ($j = 1, 2, \ldots, K$)is chosen in observation t, otherwise equals 0.

```
$title Ordered probit model of fishing mode choice
set             t               "Index of observations"                 /1*1182/,
                j               "Index of exogenous data"
                                /"intercept","day_cost","ctchrate","mnth_inc","mu_p","mu_pb","gamma"/,
                i(j)            "Subset of j, used in part a"
                                /"intercept","day_cost","ctchrate","mnth_inc","mu_p","mu_pb"/,
                k               "Index of fishing model"                /"beach","pier","p_boat","c_boat"/;

alias (i,ii);

parameter        data(t,*)        "Source data";

$call gdxxrw fish_data_v2_gams.xlsx par=data rng=A1:Q1183 cdim=1 rdim=1 checkDate
$gdxin fish_data_v2_gams.gdx
$loaddc data
$gdxin

parameter       y(t)            "Dependent variable",
                x(t,j)          "Right hand side variables",
                dum(t,k)        "Dummy matrix of fishing mode choice";

*       Indentification of type of fishing mode actually chosen
y(t) = data(t,"choice");

x(t,"intercept") = 1;
*       Daily cost of fishing for mode acutally used
x(t,"day_cost") = data(t,"day_cost");
*       Hourly fishing catch rate for mode actually used
x(t,"ctchrate") = data(t,"ctchrate");
*       Monthly income
x(t,"mnth_inc") = data(t,"mnth_inc");
*       dum = 1 if fishing mode k is chosen
loop( (t,k), dum(t,k) = 0; dum(t,k) = 1$(y(t) = ord(k)); );

*       Part(a)
*       Ordered probit model of fishing mode choice

*       Maximum likelihood Estimation
variable        LOGLIK          "Log-Likelihood function for the conditional logit model",
                COEF(j)         "Coefficients to be estimated";

equation        obj_llf         "Objective in the maximum likelihood estimation",
                con_mu          "Constraint that mu_pb > mu_p";

*       Mu is the threshold value of the unobserved
*       net utility obtained from using a particular fishing mode, choice_star,
```

```
*       for example, when choice_star < 0, choose
*       "beach", when 0<= choice_star < mu_p, choose "pier",
*       when mu_p < choice_star <= mu_pb,  choose "private boat",
*       when choice_star > mu_pb, choose "charter boat".

obj_llf..        LOGLIK =e= sum(t,dum(t,"beach")*log(errorf(- sum(i,x(t,i)*COEF(i))))) +
                            sum(t,dum(t,"pier")*log(errorf(COEF("mu_p")- sum(i,x(t,i)*COEF(i))) -
                                    errorf(-sum(i,x(t,i)*COEF(i))))) +
                            sum(t,dum(t,"p_boat")*log(errorf(COEF("mu_pb")- sum(i,x(t,i)*COEF(i))) -
                                    errorf(COEF("mu_p") - sum(i,x(t,i)*COEF(i))))) +
                            sum(t,dum(t,"c_boat")*log(1 - errorf(COEF("mu_pb") - sum(i,x(t,i)*COEF(i)))));

con_mu..        COEF("mu_pb") =g= COEF("mu_p") + 0.01;

*       Set bounds for coefficients
COEF.LO("mu_p")  = 0.01;
COEF.LO("mu_pb") = 0.02;

COEF.UP("mu_p")  = inf;
COEF.UP("mu_pb") = inf;

*       Assign initial values for mu
COEF.L("mu_p")  = 1;
COEF.L("mu_pb") = 2;

model llf /obj_llf, con_mu/;
solve llf maximizing LOGLIK using nlp;
display COEF.L;

parameter       loglik_llf              "Optimal value of log-likelihood in the restricted model";
loglik_llf = LOGLIK.L;

*       Report statistics of the unrestricted model
parameter       hessian(i,ii)           "Hessian of log-likelihood wrt coefficients",
                hinv(i,ii)              "Inverse of ssehess",
                cov(i,ii)               "Covariance matrix of type k",
                stat(i, *)              "Statistics at the point";

*       Report Hessian matrix
$batinclude lsa llf nlp "maximizing LOGLIK"
hessian(i,ii) = LSA_D2F( obj_llf, COEF(i), COEF(ii) );

*       d2f is an upper triangular matrix
hessian(i,ii) = hessian(i,ii)$(ord(ii) ge ord(i)) + hessian(ii,i)$(ord(ii) lt ord(i));

*       Find inverse of Hessian
$batinclude arealinverter Hessian i ii hinv
cov(i,ii)     = hinv(i,ii);

*       Report statistics
stat(i, "estimator")  = COEF.L(i);
stat(i, "std error")  = sqrt(cov(i,i));
stat(i, "T value")    = COEF.L(i)/sqrt(cov(i,i));
*       Use the BETAREG function:
stat(i, "P value")    = BETAREG( (card(t) - card(i))/
                                 (card(t) - card(i) + sqr(stat(i, "T value"))),
                                 (card(t) - card(i))/2, 0.5 );
display stat;
```

We have the standard regression statistics as follows:

```
----    267 PARAMETER stat  Statistics at the point

            estimator   std error     T value     P value

intercept    0.771899    0.077739    9.929414
day_cost     0.762754    0.071536   10.662536
ctchrate     0.999211    0.092481   10.804559
mnth_inc    -0.349818    0.139405   -2.509358    0.012229
mu_p         0.602114    0.041437   14.530835
mu_pb        1.694945    0.056984   29.744014 2.0314E-145
```

4.4 Censored data

If there is some latent process y_t^* with unbounded support, and we have the following latent regression model:

$$y_t^* = x_t'\beta + \mu_t,$$

where error terms μ_t are assumed to be *i.i.d* normally distributed.

However, we only observe that

$$y_t = \begin{cases} y_t^* & \text{if } y_t^* \geq \tau \\ \tau & \text{otherwise.} \end{cases}$$

when the threshold value being τ. We apply the *tobit* model to deal with this type of problems.

The log-likelihood function for the *tobit* model is the sum of the probability density function of μ when $y* > \tau$ and the probability mass function of μ when $y* \leq \tau$. When the threshold value $\tau = 0$, the log-likelihood function looks like the following:

$$lnL = \sum_{y_t>0} -\frac{1}{2}\left[ln(2\pi) + ln\sigma^2 + \frac{(y_t - x_t'\beta)^2}{\sigma^2}\right] + \sum_{y_t=0}[1 - \Phi(\frac{x_t'\beta}{\sigma})], \tag{4.4.1}$$

4.4.1 Limited dependent variable example 7: Censored data with Tobit model

Assume you have been hired by the U.S. Dairy Export Council (USDEC) to quantify the determinants of cheese purchases by Mexican households. To do this, you have decided to use the biannual survey of ENIGH. This household survey contains data on quantity purchased and associated expenditures on a detailed set of food and non-food items over a 1-week survey period. Table (7) contains a listing of the variables you have obtained from the full survey dataset. Using the ENIGH data you discover that over 60% of the surveyed households did not report cheese purchases over the survey period. This is understandable given the survey covers purchases for only a 1-week period and the shelf-life of many cheeses is longer than the survey period. To obtain consistent parameter estimates of your model of cheese purchases while accounting for these zero values, you decide to estimate a Tobit model.

Variables	Description	Units
	Cheese Purchase Characteristics	
tchzq	Quantity of cheese purchased	KG
p_chz	Cheese Price	Peso/KG
	Household Size/Composition Variables	
perlt6	Percent of household members < 6 years old	%
per6_11	Percent of household members between 6 and 11 years old	%
perge66	Percent of household members older than 65	%
hhsize	Number of household members	#

	Meal Planner Characteristics	
mp_ltgs	Meal planner has less than high school education	0/1
mp_high	Meal planner has completed high school or more	0/1
mp_age	Meal planner age	years
	Other Household Characteristics	
sm_city	Household is located in a town with 2,500-15,000 population	0/1
city_11	Household is located in a town with > 15,000 population	0/1
refrig	Does the household own a refrigerator/freezer?	0/1
incomet	Quarterly household income	10,000 Pesos
perfafh	Percent of weekly household food expenditures spent on food purchased and consumed outside the home	%
	Regional Dummy Variables	
regnw	Household located in the Northwest region of Mexico	0/1
regne	Household located in the Northeast region of Mexico	0/1
regnc	Household located in the North Central region of Mexico	0/1
regw	Household located in the West region of Mexico	0/1
regcen	Household located in the Central region of Mexico	0/1
regs	Household located in the Southern region of Mexico	0/1
regse	Household located in the Southeast region of Mexico	0/1
regdf	Household located in the Federal District region of Mexico including Mexico City	0/1

Table 7: Description of a Subset of Variables in 2002 ENIGH Dataset

(a) To help USDEC obtain a better understanding of what impacts Mexican household cheese demand, you decide to use the above dataset to estimate the following per capita "demand" function:

$$\begin{aligned} pc\_tchzq^* \quad &= F(intercept, p\_chz, incomet, refrig, perfafh, sm\_city, \\ &\qquad city, hhsize, reg\_df, perlt6, per6\_11, perge66), \end{aligned}$$

where $pc\_tchzq_t^*$ is latent per capita cheese demand and pc_tchzq_t [4] is observed per capita cheese purchases. Use the homoscedastic Tobit model to present the typical regression based statistics.

We estimate both coefficients β (BETA(i)) and covariance of error term σ^2 (BETA("sigsqr")) based on the *Tobit* log-likelihood function (4.4.1):

[4]Note: *pc_tchzq* can be calculated as *tchzq/hhsize*. For non-purchasing households *P_CHZ* was set at the provincial mean level. Meal planner is assumed to be the female head, if present.

```
$title multinomial logit model with heating choice data
set             t               "Index of observations"              /1*14646/,
                h               "Index of all data available"
                                /"int","p_chz","incomet","refrig","perfafh","sm_city","city","hhsize",
                                 "regdf","perlt6","per6_11","perge66","sigsqr",
                                 "regnw","regne","regnc","regw","regcen","regs","regse",
                                 "mp_age","mp_ltgs","tchzx","tchzq","rural","mp_high"/,
                i(h)            "Index of explanatory variables"
                                /"int","p_chz","incomet","refrig","perfafh","sm_city","city","hhsize",
                                 "regdf","perlt6","per6_11","perge66"/,
                j(h)               "Index of explanatory variables plus sigsqr"
                                /"int","p_chz","incomet","refrig","perfafh","sm_city","city","hhsize",
                                "regdf","perlt6","per6_11","perge66","sigsqr"/;
alias (i,ii), (j,jj);

parameter        data(t,*)       "Source data";

$call gdxxrw cheese_only_alt.xls par=data rng=B1:AA14647 cdim=1 rdim=1 checkDate
$gdxin cheese_only_alt.gdx
$loaddc data
$gdxin

parameter        y(t)            "Dependent variable",
                 x(t,i)          "Right hand side variables";

x(t,i)       = data(t,i);
x(t,"p_chz") = x(t,"p_chz")/100;
y(t)         = data(t,"tchzq")/data(t,"hhsize");


variable        LOGLIK          "Log-likelihood function",
                BETA(h)         "Coefficients to be estimated";

equation        obj             "Objective of tobit model";

obj..                LOGLIK =e= sum( t$(y(t)>0), -0.5*sqr(y(t) - sum(i, x(t,i)*BETA(i)))/BETA("sigsqr")
                                - 0.5*log(BETA("sigsqr")) ) +
                                sum( t$(y(t)=0), log( 1 - errorf(sum(i, x(t,i)*BETA(i))/sqrt(BETA("sigsqr")))) ) -
                                sum( t$(y(t)>0), log(sqrt(2*pi)) );

BETA.L("sigsqr") = 1;

model tobit /obj/;
solve tobit using nlp maximizing LOGLIK;

*       Report statistics of the unrestricted model
parameter       Hessian(j,jj)           "Hessian of log-likelihood wrt parameters",
                hinv(j,jj)              "Inverse of ssehess",
                cov(j,jj)               "Covariance matrix",
                stat(j,*)               "Statistics at the point";

*       Report Hessian matrix
$batinclude lsa tobit nlp "maximizing LOGLIK"
hessian(j,jj)   = LSA_D2F( obj, BETA(j), BETA(jj) );

*       d2f is an upper triangular matrix
hessian(j,jj) = hessian(j,jj)$(ord(jj) ge ord(j)) + hessian(jj,j)$(ord(jj) lt ord(j));

*       Find inverse of Hessian
$batinclude arealinverter Hessian j jj hinv
cov(j,jj)     = hinv(j,jj);

*       Report statistics
stat(j, "estimator")  = BETA.L(j);
stat(j, "std error")  = sqrt(cov(j,j));
stat(j,  "T value")   = BETA.L(j)/sqrt(cov(j,j));

*       Use the BETAREG function:
stat(j, "P value")    = BETAREG( (card(t) - card(j))/
                                 (card(t) - card(j) + sqr(stat(j, "T value"))),
                                 (card(t) - card(j))/2, 0.5 );
display stat;
```

The following are the usual censored regression statistics:

```
----    447 PARAMETER stat  Statistics at the point
          estimator   std error     T value     P value
int          -0.057       0.013      -4.256 2.098401E-5
p_chz        -0.216       0.021     -10.161
incomet       0.047       0.003      13.704
refrig        0.054       0.007       7.905
perfafh      -0.167       0.015     -11.182
sm_city       0.033       0.009       3.578 3.468126E-4
city          0.024       0.007       3.529 4.183719E-4
hhsize       -0.015       0.002      -9.396
regdf         0.060       0.007       8.203
perlt6       -0.020       0.018      -1.143       0.253
per6_11       0.024       0.017       1.406       0.160
perge66       0.020       0.013       1.500       0.134
sigsqr        0.071       0.001      49.323
```

4.5 Sample selection model

Under the censored (*tobit*) regression models we use the latent error term distribution to determine observed dependent variable PDF and probablility of obtained censored values (i.e., 0).

In the Sample Selection model we are going to introduce next, we allow a second endogenous variable to determine the probability of non-zero values of the endogenous variable of primary interest. Now we have 2 random variables where distribution of one variable impacted by the truncation of the other variable. For more details, check Heckman (1979).

The following is an example in which we go through two regression stages: one for variables subject to sample selection and the other for the endogenous variables of primary interest.

4.5.1 Limited dependent variable example 8: Sample selection model with Heckit

On page 881 of Greene (7th edition) he presents the results of a sample selection model of the utilization of the national German health care system. The second step conditional endogenous variable is the 0/1 variable DOCTOR_D. DOCTOR_D = 1 if the individual visits a doctor, 0 otherwise. The endogenous variable that determines the sample is the binary variable PUBLIC with PUBLIC = 1 if the individual is covered by public health insurance, 0 otherwise. As Greene notes: Roughly 87% of the observations in the sample do (use the public system). We might ask whether the selection on public insurance reveals any substantive difference in visits to the physician.

Below is a table that contains a listing and definitions of the variables included in the dataset that I would like you to use in the following analysis:

Variables	Description
id	person - identification number
female	female = 1; male = 0
year	calendar year of the observation
age	age in years

hsat	health satisfaction, coded 0 (low) - 10 (high)
handdum	handicapped = 1; otherwise = 0
handper	degree of handicap in percent (0 - 100)
hhninc	household nominal monthly net income in German marks / 1000
hhkids	children under age 16 in the household = 1; otherwise = 0
educ	years of schooling
married	married = 1; otherwise = 0
haupts	highest schooling degree is Hauptschul degree = 1; otherwise = 0
reals	highest schooling degree is Realschul degree = 1; otherwise = 0
fachhs	highest schooling degree is Polytechnical degree = 1; otherwise = 0
abitur	highest schooling degree is Abitur = 1; otherwise = 0
univ	highest schooling degree is university degree = 1; otherwise = 0
working	employed = 1; otherwise = 0
bluec	blue collar employee = 1; otherwise = 0
whitec	white collar employee = 1; otherwise = 0
self	self-employed = 1; otherwise = 0
beamt	civil servant = 1; otherwise = 0
docvis	number of doctor visits in last three months
hospvis	number of hospital visits in last calendar year
public	insured in public health insurance = 1; otherwise = 0
addon	insured by add-on insurance = 1; otherwise = 0

Table 8: Description of a Subset of Variables in German health Dataset

Using the same exogenous variables as shown in Table 19.9 in Greene p. 882 assume your error terms are standard normal and estimate the sample selection model first assuming independent error terms (i.e., $\rho = 0$, letâĂŹs call it Model 1) and then allowing for the dependence of the error terms (i.e., Model II). Undertake a likelihood ratio test of whether there exists endogenous self-selection. What are the results of your test?

```
$title sample selection model with health insurance
set             t               "Index of observations"              /1*4483/,
                h               "Index of all data available"
                                /"int_ss","age_ss","hhninc_ss","hhkids_ss","educ_ss","married_ss","int_pi",
                                 "age_pi","educ_pi","female_pi","sigsqr","public","docvis","rho"/,
                s(h)            "Index of variables in determining the sampe section"
                                /"int_ss","age_ss","hhninc_ss","hhkids_ss","educ_ss","married_ss"/,
                i(h)            "Index of variables of primary interest plus sigsqr in the independent errors model"
                                /"int_pi","age_pi","educ_pi","female_pi","sigsqr"/,
                j(h)            "Combination of s and i"
                                /"int_ss","age_ss","hhninc_ss","hhkids_ss","educ_ss","married_ss","int_pi",
                                 "age_pi","educ_pi","female_pi","sigsqr"/,
                k(h)            "Index of variables of primary interest plus sigsqr in the dependent errors model"
                                /"int_pi","age_pi","educ_pi","female_pi","sigsqr","rho"/,
                e(h)            "Index of all variables in the model with dependent error terms"
```

```
                                /"int_ss","age_ss","hhninc_ss","hhkids_ss","educ_ss","married_ss","int_pi",
                                 "age_pi","educ_pi","female_pi","sigsqr","rho"/;

alias (h,hh),(e,ee),(j,jj);

parameter        data(t,*)       "Source data";

$call gdxxrw german_health_1988_alt.xlsx par=data rng=A1:AB4484 cdim=1 rdim=1 checkDate
$gdxin german_health_1988_alt.gdx
$loaddc data
$gdxin

parameter       y_ss(t)          "Dependent variable in regression that determines the sample selection",
                y_pi(t)          "Dependent variable in regression that determines the variable of primary interest",
                x_ss(t,h)        "Explantory variables in stage ss",
                x_pi(t,h)        "Explantory variables in stage ss";

*        Data in the sample selection regression
x_ss(t,"int_ss")        = data(t,"int");
x_ss(t,"age_ss")        = data(t,"age");
x_ss(t,"hhninc_ss")     = data(t,"hhninc");
x_ss(t,"hhkids_ss")     = data(t,"hhkids");
x_ss(t,"educ_ss")       = data(t,"educ");
x_ss(t,"married_ss")    = data(t,"married");
y_ss(t)                 = data(t,"public");

x_ss(t,"age_ss")        = x_ss(t,"age_ss")/10;
x_ss(t,"hhninc_ss")     = x_ss(t,"hhninc_ss")/1000;

*        Data in the primary regression
x_pi(t,"int_pi")        = data(t,"int");
x_pi(t,"age_pi")        = data(t,"age");
x_pi(t,"educ_pi")       = data(t,"educ");
x_pi(t,"female_pi")     = data(t,"female");
y_pi(t)                 = data(t,"docvis");
y_pi(t)                 = 1$(y_pi(t) > 0)  + 0$(y_pi(t) = 0);

*        Assume independent error terms (rho = 0)
variable        LOGLIK          "Log-likelihood function",
                BETA(h)         "Coefficients to be estimated";

equation        obj             "Objective of tobit model";

obj..                LOGLIK =e= sum( t$(y_pi(t)>0),
                                log( errorf(sum(i, x_pi(t,i)*BETA(i))) ) -
                                0.5*log(2*pi) - 0.5*log(BETA("sigsqr")) -
                                0.5*(sqr(( y_ss(t) - sum(s, x_ss(t,s)*BETA(s)))/BETA("sigsqr"))) ) +
                                sum( t$(y_pi(t)=0), log( errorf(-sum(i, x_pi(t,i)*BETA(i)))) );

*       Set initial value of sigma square
BETA.L("sigsqr") = 1;

model heckit /obj/;
solve heckit using nlp maximizing LOGLIK;

*       Record objective value for future use
parameter        llf_ind                "Log-likelihood level of sample selection model with independent errors";
llf_ind = LOGLIK.L;

*        Report statistics of the unrestricted model
parameter       Hessian(h,hh)           "Hessian of loglikelihood wrt parameters",
                hinv(h,hh)              "Inverse of ssehess",
                cov(h,hh)               "Covariance matrix",
                stat(h,*)               "Statistics at the point";

*        Report Hessian matrix
$batinclude lsa heckit nlp "maximizing LOGLIK"
hessian(j,jj)   = LSA_D2F( obj, BETA(j), BETA(jj) );

*        d2f is an upper triangular matrix
hessian(j,jj) = hessian(j,jj)$(ord(jj) ge ord(j)) + hessian(jj,j)$(ord(jj) lt ord(j));

*        Find inverse of Hessian
$batinclude arealinverter Hessian j jj hinv
cov(j,jj)     = hinv(j,jj);

*        Report statistics
stat(j, "estimator")  = BETA.L(j);
stat(j, "std error")  = sqrt(cov(j,j));
stat(j,  "T value")   = BETA.L(j)/sqrt(cov(j,j));
```

```
*       Use the BETAREG function:
stat(j, "P value")    = BETAREG( (card(t) - card(j))/
                                 (card(t) - card(j) + sqr(stat(j, "T value"))),
                                 (card(t) - card(j))/2, 0.5 );
display stat;
```

We have standard regression statistics as follows:

```
----    477 PARAMETER stat  Statistics at the point

             estimator   std error     T value     P value

int_ss           1.509       0.059      25.454
age_ss          -0.007       0.008      -0.838       0.402
hhninc_ss       -0.026       0.005      -4.940 8.088406E-7
hhkids_ss       -0.012       0.019      -0.635       0.525
educ_ss         -0.044       0.004     -12.013
married_ss      -0.004       0.021      -0.194       0.846
int_pi          -0.232       0.134      -1.727       0.084
age_pi           0.013       0.002       7.573
educ_pi         -0.010       0.008      -1.193       0.233
female_pi        0.318       0.040       8.046
sigsqr           0.423       0.008      53.861
```

Then we estimate a sample selection model in which we have dependent error terms:

```
*       Assume dependent error terms (rho <> 0)
equation        obj_dep               "Objective function in the model with dependent errors";

obj_dep..        loglik =e= sum(t$(y_pi(t) = 0), log(errorf(-sum(k, x_pi(t,k)*BETA(k))))) +
                            sum(t$(y_pi(t) > 0), log(errorf(sum(k, x_pi(t,k)*BETA(k)) +
                                   BETA("rho")/sqrt(BETA("sigsqr"))*sqrt(1-sqr(BETA("rho")))*
                                   (y_ss(t) - sum(s, x_ss(t,s)*BETA(s)))/sqrt(BETA("sigsqr")))) -
                                   0.5*log(2*pi) - 0.5*log(BETA("sigsqr")) -
                                   0.5*(sqr((y_ss(t) - sum(s, x_ss(t,s)*BETA(s)))/BETA("sigsqr"))));
BETA.L("sigsqr") = 1;
BETA.L("rho") = 1;

model heckit_dep /obj_dep/;
solve heckit_dep using nlp maximizing LOGLIK;

parameter        llf_dep         "Log-likelihood level of sample selection model with dependent errors";
llf_dep = LOGLIK.L;

parameter       stat_dep(h,*)    "Statistics at the point";

*       Report Hessian matrix
$batinclude lsa heckit_dep nlp "maximizing LOGLIK"
hessian(e,ee)   = LSA_D2F( obj_dep, BETA(e), BETA(ee) );

*       d2f is an upper triangular matrix
hessian(e,ee) = hessian(e,ee)$(ord(ee) ge ord(e)) + hessian(ee,e)$(ord(ee) lt ord(e));

*       Find inverse of Hessian
$batinclude arealinverter Hessian e ee hinv
cov(e,ee)     = hinv(e,ee);

*       Report statistics
stat_dep(e, "estimator")  = BETA.L(e);
stat_dep(e, "std error")  = sqrt(cov(e,e));
stat_dep(e,  "T value")   = BETA.L(e)/sqrt(cov(e,e));

*       Use the BETAREG function:
stat_dep(e, "P value")    = BETAREG( (card(t) - card(e))/
                                 (card(t) - card(e) + sqr(stat_dep(e, "T value"))),
                                 (card(t) - card(e))/2, 0.5 );
display stat_dep;
```

We find regression statistics in a dependent error terms model like these:

```
----    851 PARAMETER stat_dep  Statistics at the point

             estimator    std error     T value     P value

int_ss           1.353        0.062      21.981
age_ss           0.008        0.008       1.007       0.314
hhninc_ss       -0.026        0.005      -4.918 9.042921E-7
hhkids_ss       -0.011        0.019      -0.556       0.578
educ_ss         -0.047        0.004     -12.257
married_ss      -0.007        0.021      -0.315       0.753
int_pi          -0.356        0.141      -2.529       0.011
age_pi           0.014        0.002       7.625
educ_pi         -0.008        0.009      -0.880       0.379
female_pi        0.313        0.040       7.768
sigsqr           0.454        0.008      54.021
rho              0.707        0.075       9.425
```

For the likelihood ratio test, we set

```
*       Likelihood ratio test
parameter        lr                "Likelihood ratio test statistic",
                alpha               "Type 1 error level"        /0.05/,
                pval                "P value based on independent error terms";

lr = 2*(llf_dep - llf_ind);
pval= 1- gammareg(lr/2, 1/2);

display lr;

if ( pval lt alpha, display "Reject H0: indepdent error terms assumption is true";
             else  display "Can not reject H0: independent error terms assumption is true"; );
```

And we reject the null saying that the sample selection model with independent error terms are reasonable.

```
----    861 PARAMETER lr                  =       22.820  Likelihood ratio test statistic

----    863 Reject H0: indepdent error terms assumption is true
```

5 Appendix A

5.1 More NLS examples

5.1.1 NLS Example 3: Consumption Prediction

Consider the following consumption function where consumption, C_t, depends on income, Y_t, via the following model:

$$C_t = \beta_1 + \beta_2 Y_t + e_t, \tag{5.1.1}$$

$$e_t = \theta_1 e_{t-1} + \theta_2 e_{t-2} + \nu_t, \tag{5.1.2}$$

where ν_t is a i.i.d random error with $\nu_t \sim (0, \sigma_\nu^2)$.

(a) Using equation (5.1.1) and (5.1.2), reformulate the model such that:

$$C_t = C_t(C_{t-1}, C_{t-2}, Y_t, Y_{t-1}, Y_{t-2} | \beta_1, \beta_2, \theta_1, \theta_2) \tag{5.1.3}$$

For part (a), if we regress current consumptin on current income and lagged income and consumption, we get

$$C_t = \beta_1 + \beta_2 y_t + \theta_1(c_{t-1} - \beta_1 - \beta_2 y_{t-1}) + \theta_2(c_{t-2} - \beta_1 - \beta_2 y_{t-2}) + v_t.$$

(b) Use the data, nl_cons_v2.xlsx to obtain estimates of β_1, β_2, θ_1, and θ_2. Report the usual regression results. The consumption data is represented by the CONSUME variable and the income data is represented by the INC variable in the above dataset.

From part (b) to part (d), we first estimate parameter $\beta_1, \beta_2, \theta_1$ and θ_2, then report the usual regression statistics, after that, we run tests based on these statistics:

```
$title  Predict Current Consumptin Given Income & Lagged Consumption
*       Part (b)
*       Estimation of coefficients
set             t               "Time of observations"          /t1*t38/,
                i               "Index of coefficients"         /"beta_1","beta_2","theta_1","theta_2"/,
                q(t)            "Time for which lagged values exist";

alias (i,ii);

*       Time of observations when lagged data are available
q(t) = yes$(ord(t)>2);

*       Load source data
parameter       data_nl(t,* )   "Source data";

$call gdxxrw nl_cons_v2_gams.xls par=data_nl rng=A1:c39 cdim=1 rdim=1 checkDate
$gdxin nl_cons_v2_gams.gdx
$loaddc data_nl
$gdxin

parameter       c(t)            "Consumption at time t",
                y(t)            "Income at time t";

*       Load consumption and income data
c(t)=data_nl(t,"consume");
y(t)=data_nl(t,"inc");

variable        COEF(i)         "Coefficients to be estimated",
                UPSILON(t)      "Error term",
                SSE             "Sum of squared errors";

equation        fit(t)          "Reformulated model",
                obj             "Objective";

obj..           SSE =e= sum(q,sqr(UPSILON(q)));

fit(q(t))..     c(t) =e= COEF("beta_1") - COEF("beta_1")*COEF("theta_1") -
                                          COEF("beta_1")*COEF("theta_2") +
                                          COEF("beta_2")*y(t) -
                                          COEF("beta_2")*COEF("theta_1")*y(t-1) -
                                          COEF("beta_2")*COEF("theta_2")*y(t-2) +
                                          COEF("theta_1")*c(t-1) +
                                          COEF("theta_2")*c(t-2) + UPSILON(t);
```

```
model reformulated /obj,fit/;
solve reformulated minimzing SSE using nlp;
display COEF.L;

*       Report statistics
parameter               jacobian(t,i)           "Jacobian of constraints wrt coefficients",
                        jacsqr(i,ii)            "Squared Jacobian matrix",
                        jsinv(i,ii)             "Inverse of jacsqr",
                        sigma_hat               "Unbiased estimator for error variance",
                        cov(i,ii)               "Covariance matrix",
                        r_2                     "Coefficient of determination",
                        cbar                    "Mean of dependent variable",
                        sst                     "Totol sum of squares",
                        statistics(i,*)         "Statistics at the point";

*       Report Jacobian matrix
$batinclude lsa reformulated nlp "minimizing SSE"
jacobian(t,i) = LSA_DF( fit(t), COEF(i) );

*       Find squared Jacobian matrix
jacsqr(i,ii)  = sum(t$(q(t)), jacobian(t,i)*jacobian(t,ii));

*       Find inverse of squared Jacobian
$batinclude arealinverter jacsqr i ii jsinv

*       Totally number of observations when lagged values are available: q
*       Degrees of freedom = card(q) - card(i)
sigma_hat = SSE.L / (card(q) - card(i));
cov(i,ii) = jsinv(i,ii)*sigma_hat;

display cov;

*       Genereate a report for statistcs
statistics(i,"estimator")  = COEF.L(i);
statistics(i,"std error")  = sqrt(cov(i,i));
statistics(i,"T value")    = COEF.L(i)/sqrt(cov(i,i));
*       Use the BETAREG function:
statistics(i,"P value")    = BETAREG( (card(q) - card(i))/
                                      (card(q) - card(i) + sqr(statistics(i,"T value"))),
                                      (card(q) - card(i))/2, 0.5 );

display statistics;

*       Report some other statistics
*       Mean of dependent variables
cbar = sum(q,c(q))/card(q);

*       Total sum of squares
sst  = sum(q,sqr(c(q))) - card(q)*sqr(cbar);

*       Coefficient of determination
r_2  = 1 - SSE.L/sst;
display sigma_hat,r_2;
```

We report regression statistics as:

```
----    296 PARAMETER statistics  Statistics at the point

          estimator   std error     T value     P value

beta_1     0.576375    4.534897    0.127098    0.899658
beta_2     0.925158    0.088087   10.502803 6.73500E-12
theta_1    0.816951    0.151638    5.387497    0.000006
theta_2   -0.606491    0.152915   -3.966204    0.000385

----    308 PARAMETER sigma_hat            =    31.259015  Unbiased estimator for error variance
            PARAMETER r_2                  =     0.836628  Coefficient of determination
```

(c) Test whether the AR(1) model would have been adequate in describing the relationship among the regression errors versus the more extensive specification.

Since θ_2 is statistically significant, AR(1) is not sufficient in describing the relationship between the regression errors vs the more extensive specification.

(d) Test the following null hypothesis: $H_0 : \theta_1 = \theta_2 = 0$. What is the implication of this hypothesis test?

```
*        Part (d)
*        We are interested in the Wald statistic: sqr(theta_hat - theta_0)/var(theta_hat)

set     m       "Row index for the Wald statistic calculation"        /1*2/;

alias (m,mm);

table   r(m,i)  "Coefficient for the Wald statistic calculation"
        beta_1  beta_2        theta_1        theta_2
1        0          0             1              0
2        0          0             0              1;

parameter
        rs_cov(m,mm)            "Weighted covariance matrix (r*cov*r')",
        inv_rs_cov(m,mm)        "Inverse of rs_cov",
        wald                    "Wald statistics",
        alpha                   "Tolerance level"      /0.05/,
        df_wald                 "Degrees of freedom"   /2/,
        pval_wald               "P value of Wald test";

*        Calculation of the weighted covariance matrix
rs_cov(m,mm) = sum((i,ii), r(m,i)*cov(i,ii)*r(mm,ii));

*        Find the inverse of rs_cov
$batinclude arealinverter rs_cov m mm inv_rs_cov

*        Find the wald statistic
wald      = sum((m,i,mm,ii), r(m,i)*COEF.L(i)*inv_rs_cov(m,mm)*COEF.L(ii)*r(mm,ii));

*        Use the GAMS function for the chi-squared distribution:
pval_wald = 1 - GAMMAREG(wald/2,df_wald/2);

display pval_wald, wald;

*        Show testing results
if ((pval_wald lt alpha), display "Reject H_0: theta_1 = theta_2 = 0";
                   else  display "Fail to reject H_0: theta_1 = theta_2 = 0";);
```

We reject the null hypothesis $\mathbb{H}_0 : \theta_1 = \theta_2 = 0$ through the Wald test:

```
----    430 PARAMETER pval_wald           =  1.442897E-7  P value of Wald test
            PARAMETER wald                =    31.502886  Wald statistics

----    433 Reject H_0: theta_1 = theta_2 = 0
```

5.1.2 NLS Example 4: Learning Curve

An important ingredient underlying firm cost theory is the learning curve. The idea behind the learning curve is that, as more items are produced, workers become more experienced and the average cost of production falls. Let's represent the average cost (as represented by time) of producing X items by AC_x. A typical learning curve model can be represented via the following: $AC_x = \gamma X^{\delta}$ where γ and δ are unknown coefficients and X is the cumulative amount of output produced up to and including the current period (i.e., $X_t = \sum_{s=1}^{t} x_s$) where t is the current time period. The total cost (time) of producing X_t items (TC_x) is simply $X_t AC_{X_t} = X_t \gamma X_t^{\delta} = \gamma X_t^{\delta+1}$. For estimation of the unknown coefficients, cost is often represented by the amount of time taken to produce a given lot of items.

Suppose you have collected data pertaining to the time associated with the setup of spinning frames in a U.S. yarn producing firm. The data you collected and contained in the data set yarn-industry.xlsx consists of:

Variables	Description
time(i)	Time (minutes) taken to set up the i^{th} lot of frames
frames(i)	Number of frames completed in the i^{th} lot (Note: This is not the cumulative number of frames)
lot	Lot number

Table 9: Data available in the learning curve problem

Note, the total (cumulative) number of frames setup after i lots have been completed is:

$$totfr_i = \sum_{j=1}^{i} frames_j.$$

Therefore, the total time (TC) to setup $totfr_i$ frames can be represented by the following after incorporating the learning model:

$$tc_{toter_i} = totfr_i \times AC_{totfr_i}, \tag{5.1.4}$$

Given that you are interested in examining whether the learning curve model described above is relevant to the above firm you recognize that:

$$time_i = TC_{toter_i} - TC_{toter_{i-1}}, \tag{5.1.5}$$

(a) Using equation(5.1.5) as a base, formulate the statistical model to which you can apply nonlinear least squares to estimate learning curve coefficients, γ and δ. What is your underlying statistical model? Given this model, what is your null hypothesis as to the sign of γ if the above model of learning is appropriate to this firm? Using non-linear least squares, determine whether the data associated with this yarn-producing firm supports your null hypothesis?

This part is similar to the part (a) of the previous example.

```
$Title Learning Curve
*       Part (a)
*       Estimate the NLS model
set             r               "Lot number"            /1*60/,
                k(r)            "Subset of r, one term lagged lot number",
                i               "Index of unknowns"     /"gamma","delta"/;

*       Define k(r)
k(r) = yes$(ord(r) > 1);

alias (i,j);

parameter       data(r,*)       "Source data",
```

```
                totfr(r)        "Total num of frames when the r_th lot has been completed",
                totfr_1(r)      "Totfr with One term lagged",
                t(r)            "Time taken to set up the r_th lot of frames";

$call gdxxrw yarn_industry.xls par=data rng=A1:C61 cdim=1 rdim=1 checkDate
$gdxin yarn_industry.gdx
$loaddc data
$gdxin

*       Dependent variable
t(r) = data(r,"time");

*       Starting value of total frames
totfr("1") = data("1","frames");

*       Set the first number of totfr_1 = 0
totfr_1("1") = 0;

*       Construct cumulative amount of frames
loop(r$(k(r)),totfr(r) = totfr(r-1) + data(r,"frames"); totfr_1(r) = totfr(r-1););

variable        PARA(i)         "Unknowns to be estimated",
                SSE             "Sum of squared errors",
                EPSI(r)         "Error terms";

equation        fit(r)          "Determination of time(r)",
                obj             "Objective";

obj..           SSE =e= sum(r,sqr(EPSI(r)));

fit(r)..        t(r) =e= PARA("gamma")*totfr(r)**(PARA("delta") + 1) -
                         PARA("gamma")*totfr_1(r)**(PARA("delta") + 1) + EPSI(r);

model learning_curve /obj, fit/;
solve learning_curve minimizing SSE using nlp;

display PARA.L;

*       Report statistics
parameter               jacobian(r,i)           "Jacobian of constraints wrt coefficients",
                        jacsqr(i,j)             "Squared Jacobian matrix",
                        jsinv(i,j)              "Inverse of jacsqr",
                        sigma_hat               "Unbiased estimator for error variance",
                        cov(i,j)                "Covariance matrix",
                        r_2                     "Coefficient of determination",
                        cbar                    "Mean of dependent variable",
                        sst                     "Totol sum of squares",
                        statistics(i,*)         "Statistics at the point";

*       Report Jacobian matrix
$batinclude lsa learning_curve nlp "minimizing SSE"
jacobian(r,i) = LSA_DF( fit(r), PARA(i) );

*       Find squared Jacobian matrix
jacsqr(i,j)   = sum(r, jacobian(r,i)*jacobian(r,j));

*       Find inverse of the squared Jacobian matrix
$batinclude arealinverter jacsqr i j  jsinv

*       Unbiased estimator of error variance
sigma_hat = SSE.L / (card(r) - card(i));
cov(i,j)  = jsinv(i,j)*sigma_hat;

*       Generate output report
statistics(i, "estimator")  = PARA.L(i);
statistics(i, "std error")  = sqrt(cov(i,i));
statistics(i, "T value")    = PARA.L(i)/sqrt(cov(i,i));
* Use the BETAREG function:
statistics(i, "P value")    = BETAREG( (card(r) - card(i))/
                                       (card(r) - card(i) + sqr(statistics(i, "T value"))),
                                       (card(r) - card(i))/2, 0.5 );
display statistics;

*       Report some other statistics
*       Mean of dependent variables
cbar = sum(r, t(r))/card(r);
*       Total sum of squares
sst  = sum(r, sqr(t(r))) - card(r)*sqr(cbar);
*       Coefficient of determination
r_2  = 1 - SSE.L/sst;
```

```
display sigma_hat,r_2;
```

The Null hypothesis $\mathbb{H}_0$ is: $\delta < 0$, based on this, we have

```
----    446 PARAMETER statistics  Statistics at the point
        estimator   std error     T value     P value

gamma      52.700       3.541      14.882 1.82838E-21
delta      -0.240       0.010     -24.858 1.28613E-32


----    456 PARAMETER sigma_hat            =      197.414  Unbiased estimator for error variance
            PARAMETER r_2                  =        0.896  Coefficient of determination
```

The results of delta is significantly negative, which means that the first order condition of average cost is negative, this further implies that learning effect could possibly take place.

(b) Estimate the average setup cost after 50, 400 and 800 cumulative frames have been produced. At each of these data points, test the null hypothesis that the associated average cost is different from 11.5 minutes.

From part(b) to (d), we first set up the equation of related variable then call the *jacobian* routine to report the gradient matrices depend on different number of accumulated frames; Based on these, we run the Wald test to check if $\mathbb{H}_0$ holds or not.

```
*       Part(b)
set             b                       "Index for accumulated frames in part b" /1*3/,
                Wald_stats              "Statistics to report"
                                        /totframe, Wald_Statistic, "P value"/;

parameter       totf(b)                 "Accumulated frames"          /"1" 50, "2" 400, "3" 800/,
                ac(b)                   "Average set up cost given total frames",
                ac50                    "Average cost when totfr=50",
                ac400                   "Average cost when totfr=400",
                ac800                   "Average cost when totfr=800",
                alpha                   "Type 1 error level"          /0.05/,
                jac(b,i)                "Jacobian matrix at the point",
                w_ac(b)                 "Wald statistics for average cost",
                df_ac                   "Degrees of freedom"          /1/,
                pval_ac(b)              "P value in Wald test",
                ac0                     "Proposed average cost"       /11.5/,
                adt_cov(b)              "Adjusted covariance",
                wald(b,wald_stats)      "Wald statistics report";

*       Define average set up cost
ac(b) = PARA.L("gamma")*totf(b)**PARA.L("delta");
ac50  = ac("1");
ac400 = ac("2");
ac800 = ac("3");

*       Wald testing of Average cost when totfr = 50,400, 800
variable        A_COST(b)       "Average set up cost given total frames";
equation        acdef(b)        "Definition of average cost given totoal frames";

acdef(b)..      A_COST(b) =e= PARA("gamma")*totf(b)**PARA("delta");

*       set up a model of learning curve with average cost defined
model lc_ac /obj,fit,acdef/;
PARA.FX(i) = PARA.L(i);
solve lc_ac using mcp;

*       Report Jacobian matrix
$batinclude lsa lc_ac mcp
jac(b,i) = LSA_DF( acdef(b), PARA(i) );

*       Find adjusted covariance
```

```
adt_cov(b) = sum((i,j), jac(b,i)*cov(i,j)*jac(b,j));

*       Find Wald test statistics
w_ac(b)    = (ac(b) - ac0)*(1/adt_cov(b))*(ac(b) - ac0);

*       Use gammareg function to report p value of wald test statistic
pval_ac(b) = 1 - gammareg(w_ac(b)/2, df_ac/2);

wald(b, "totframe")       = totf(b);
wald(b, "Wald_Statistic") = w_ac(b);
wald(b, "P value")        = pval_ac(b);

*       Make a list of Wald test report
$onecho >wald_test.gms
parameter        wald_test(b,*,*)            "Wald Test Results";

loop(b$(wald(b, "P value") lt alpha),
        wald_test(b, "Reject H_0", "P value") = wald(b, "P value");
        wald_test(b, "Reject H_0", "Frames")  = totf(b);
        wald_test(b, "Reject H_0", "Wald")    = wald(b, "wald_statistic"););
loop(b$(wald(b, "P value") ge alpha),
        wald_test(b, "Fail to Reject H_0", "P value") = wald(b, "P value");
        wald_test(b, "Fail to Reject H_0", "Frames")  = totf(b);
        wald_test(b, "Reject H_0", "Wald")            = wald(b, "wald_statistic"););
option wald_test:3:2:1;
display "Wald Test:", alpha, wald_test;
$offecho

$include wald_test
```

In all three case, Wald test rejects the null hypothesis that average cost equals 11.5 at $\alpha = 0.05$.

```
----    532 PARAMETER wald_test  Wald Test Results

                  Frames     P value        Wald

1.Reject H_0      50.000                 206.622
2.Reject H_0     400.000 3.218212E-8      30.571
3.Reject H_0     800.000 3.83027E-13      52.728
```

(c) Calculate the marginal cost, defined in terms of time as $\frac{\partial TC}{\partial TOTER}$, of producing frames after 50, 400 and 800 cumulative frames have been produced. At each of these data points, test the null hypothesis that the associated marginal effect is different from 11.5 minutes.

```
*       Part(c)
parameter       mc(b)           "Marginal set up cost given total frames",
                mc50            "Marginal cost when totfr=50",
                mc400           "Marginal cost when totfr=400",
                mc800           "Marginal cost when totfr=800",
                w_mc(b)         "Wald statistics for marginal cost",
                df_mc           "Degrees of freedom"                /1/,
                pval_mc(b)      "P value in Wald test",
                mc0             "Proposed marginal cost"            /11.5/;

*       Define marginal cost
mc(b) = PARA.L("gamma")*(PARA.L("delta") + 1)*totf(b)**PARA.L("delta");
mc50  = mc("1");
mc400 = mc("2");
mc800 = mc("3");

*       Wald testing of Marginal cost when totfr = 50, 400, 800
variable        M_COST(b)       "Marginal set up cost given total frames";

equation        mcdef(b)        "Definition of marginal cost given totoal frames";
```

```
mcdef(b)..        M_COST(b) =e=  PARA("gamma")*(PARA("delta") + 1)*totf(b)**PARA("delta");

*       Set up a model of learning curve with marginal cost defined
model lc_mc /obj,fit,mcdef/;
PARA.FX(i) = PARA.L(i);
solve lc_mc using mcp;

*       Report Jacobian matrix
$batinclude lsa lc_mc mcp
jac(b,i) = LSA_DF( mcdef(b), PARA(i) );

*       Wald test on marginal cost
*       Find adjusted covariance
adt_cov(b) = sum((i,j), jac(b,i)*cov(i,j)*jac(b,j));

*       Find Wald test statistics
w_mc(b) = (mc(b) - mc0)*(1/adt_cov(b)) *(mc(b) - mc0);

*       Use gammareg function to report p value
pval_mc(b) = 1- gammareg(w_mc(b)/2, df_mc/2);

wald(b, "totframe")       = totf(b);
wald(b, "Wald_Statistic") = w_mc(b);
wald(b, "P value")        = pval_mc(b);

$include wald_test
```

We reject null hypothesis in each of the following cases:

```
----    723 PARAMETER wald_test  Wald Test Results

                  Frames        Wald

1.Reject H_0      50.000     182.661
2.Reject H_0     400.000     277.420
3.Reject H_0     800.000     698.493
```

(d) At each of the above data points test the null hypothesis that $MC - AC = 3.0$ minutes .

```
*       Part(d)
parameter     mc_ac(b)          "MC - AC given total frames",
              mc_ac50           "MC - AC when totfr=50",
              mc_ac400          "MC - AC when totfr=400",
              mc_ac800          "MC - AC when totfr=800",
              mc_ac(b)          "MC - AC when given totfr",
              w_mc_ac(b)        "Wald statistics for average cost",
              df_mc_ac          "Degrees of freedom"                /1/,
              pval_mc_ac(b)     "P value in Wald test",
              mc_ac0            "Proposed average cost"             /3/;

*       Define the difference of marginal cost and average cost
mc_ac(b) = mc(b) - ac(b);
mc_ac50  = mc_ac("1");
mc_ac400 = mc_ac("2");
mc_ac800 = mc_ac("3");

*       Wald testing of (Marginal cost - Average cost) when totfr = 50, 400, 800
variable        M_A_COST(b)     "(Marginal cost - Average cost) given total frames";
equation        mc_acdef(b)     "Definition of the difference between mc and ac given totoal frames";

mc_acdef(b)..        M_A_cost(b) =e=  PARA("gamma")*(PARA("delta") + 1)*totf(b)**PARA("delta") -
                                      PARA("gamma")*totf(b)**PARA("delta");
*       set up a model of learning curve with average cost defined
PARA.FX(i) = PARA.L(i);
model lc_mc_ac /obj,fit,mc_acdef/;
solve lc_mc_ac using mcp;

*       Report Jacobian matrix
$batinclude lsa lc_mc_ac mcp
```

```
jac(b,i) = LSA_DF( mc_acdef(b), PARA(i) );

*       Wald test on (marginal cost - average cost)
*       Find adjusted covariance
adt_cov(b) = sum((i,j), jac(b,i)*cov(i,j)*jac(b,j));

*       Find Wald test statistics
w_mc_ac(b) = (mc_ac(b) - mc_ac0)*(1/adt_cov(b))*(mc_ac(b) - mc_ac0);

*       Use gammareg function to report p value
pval_mc_ac(b) = 1 - gammareg(w_mc_ac(b)/2, df_mc_ac/2);

wald(b, "totframe")       = totf(b);
wald(b, "Wald_Statistic") = w_mc_ac(b);
wald(b, "P value")        = pval_mc_ac(b);

$include wald_test
```

We reject all null hypothesis at the 5% level:

```
----    915 PARAMETER wald_test  Wald Test Results

                  Frames        Wald

1.Reject H_0      50.000     532.976
2.Reject H_0     400.000    1605.039
3.Reject H_0     800.000    2520.603
```

5.2 Other limited dependent variable examples

5.2.1 Limited dependent variable Example 9: conditional logit with NC beach

Data file `NC_beach` contains variables constructed using the southeastern NC beach visitation data: Note that, with the exception of income, these matrices all have 649 rows and 17 columns indicating there are N=649 respondents and J=17 alternatives. The matrices are defined as follows:

Variables	Description
income	Annual income for each respondent
price	'Travel cost' in dollars for each respondent to each of the beaches
trip	Number of times each respondent visited each beach
parking	Number of parking spots available at each beach (constant across people, 100's of spots)
miles	Length in miles of the beach(constant across people)
width	Width in feet of the beach (constant across people)

Table 10: Description of variables in the data file: NC_beach

Analyze these data using the conditional logit model. In particular, complete the following:

(a) Estimate a choice model including price, parking, miles, and width as explanatory variables. Report estimates

and t-statistics for this model, using both inverse Hessian and robust standard errors. Briefly interpret your results. (Note: the matrix trip contains multiple choice outcomes; as such elements are not constrained to be zero or one).

```
$title NC beach visitation
set             n               "Index of respondents"          /n1*n649/,
                j               "Index of beaches"              /j1*j17/,
                k               "Index of explanatory variables"
                                /"price","miles","width","parking","miles_inc","site9"/,
                k0(k)               "Index of explantory variables in the baseline model"
                                /"price","miles","width","parking"/,
                k1(k)               "Index of explantory variables in the model with inc/miles cross"
                                /"price","miles","width","parking","miles_inc"/,
                k2(k)               "Index of explantory variables in the model with site9"
                                /"price","miles","width","parking","site9"/,
                s               "Three type of standard error estimation"
                                /"inverse Hessian","robust estimate for cov"/;

alias (k,kk);
alias (n,nn);
alias (j,jj,jj2);

parameter       income(n)       "Annual income for each respondent",
                price(n,j)      "Travel cost in dollars for each respondent to each of the beaches",
                trip(n,j)       "Number of times each respondent visited each beach",
                parking(n,j)    "Number of parking spots available at each beach",
                miles(n,j)      "Length in miles of the beach",
                width(n,j)      "Width in feet of the beach";

*       Load data
$label gdxxrw

$onecho > gdxxrw.rsp
par=income rng=income!A1:B649 cdim=0 rdim=1
par=price rng=price!A1:R650 cdim=1 rdim=1
par=trip rng=trip!A1:R650 cdim=1 rdim=1
par=parking rng=parking!A1:R650 cdim=1 rdim=1
par=miles rng=miles!A1:R650 cdim=1 rdim=1
par=width rng=width!A1:R650 cdim=1 rdim=1
$offecho

$call gdxxrw NC_beach.xlsx @gdxxrw.rsp
$gdxin NC_beach.gdx
$load income price trip parking miles width
$gdxin


*       Beach choices data
parameter       choice_num      "Total number of choices",
                freq(j)         "Frequence of beach j was chosen";

*       Find the frequency of beach choice
choice_num = sum((n,j), trip(n,j));
freq(j)    = sum(n, trip(n,j))/choice_num;

*       Estimate the baseline model
parameter       x(n,j,k)        "Explanatory variable k of respondent n who chooses j",
                miles_inc(n,j)  "Miles*Income",
                site9(n,j)      "Equals 1 if site number=9";

*       Scale income by 10000
income(n) = income(n)/10000;

*       Find the multiplication of miles and income
loop( (n,j), miles_inc(n,j) = sum((nn)$(sameas(n,nn)), miles(n,j)*income(nn)); );

*       Define matrix site9
site9(n,j) = 0;
site9(n,j) = 1$(ord(j) = 9);

x(n,j,"price")   = price(n,j);
x(n,j,"parking") = parking(n,j);
x(n,j,"miles")   = miles(n,j);
x(n,j,"width")   = width(n,j);

x(n,j,"miles_inc") = miles_inc(n,j);
x(n,j,"site9")     = site9(n,j);

*       Part (a)
```

```
*       Estimate a choice model including parking, price, miles and width
*       as explanatory variables
variable        BETA(k)         "Coefficients of explanatory variables",
                LOGLIK          "Log-Likelihood"
                LL(n)           "Definition of log-likelihood for respondant n";

equation        obj             "Total log-likelihood",
                llf0(n)         "Log-Likelihood definition for each respondant n";


llf0(n)..        LL(n) =e= sum(j, trip(n,j)*sum(k0, x(n,j,k0)*BETA(k0))) -
                                sum(j, trip(n,j))*log(sum(j, exp(sum(k0, x(n,j,k0)*BETA(k0)))));

obj..               LOGLIK =e= sum(n, LL(n));

model nc_beach /obj, llf0/;
solve nc_beach maximizing LOGLIK using nlp;


*       Define coefficients
parameter       Jacobian(n,k)           "Jacobian of constraints wrt coefficients",
                Hessian(n,k,kk)         "Hessian of constraints wrt coefficients",
                Hessian_sum(k,kk)       "Hessian of log-likelihood wrt coefficients",
                jacsqr(k,kk)            "Squared Jacobian matrix",
                jsinv(k,kk)             "Inverse of jacsqr",
                hinv(k,kk)              "Inverse of ssehess",
                cov(s,k,kk)             "Covariance matrix of type s",
                statistics(s,k,*)       "Statistics at the point";

*       Report Jacobian and Hessian matrix
$batinclude lsa nc_beach nlp "maximizing LOGLIK"

alias (k0, kk0);

Jacobian(n,k0)     = LSA_DF( llf0(n), BETA(k0) );
Hessian(n,k0,kk0)   = LSA_D2F( llf0(n), BETA(k0), BETA(kk0) );

*       d2f is an upper triangular matrix
hessian(n,k0,kk0) = hessian(n,k0,kk0)$(ord(kk0) ge ord(k0)) + hessian(n,kk0,k0)$(ord(kk0) lt ord(k0));

*      Find squared Jacobian and Hessian of sum of log-likelihood functions
jacsqr(k0,kk0)      = sum(n, Jacobian(n,k0)*Jacobian(n,kk0));
Hessian_sum(k0,kk0) = sum(n, Hessian(n,k0,kk0));

*       Find inverse of Hessian and Squared jacobian matrix
$batinclude arealinverter Hessian_sum k0 kk0 hinv
$batinclude arealinverter jacsqr k0 kk0 jsinv

*       Three typies of estiamted covariance matrix of coefficients
cov("inverse Hessian",k0,kk0) = hinv(k0,kk0);

alias (k0,m,mm);
cov("robust estimate for cov",k0,kk0) = sum((mm,m), hinv(k0,mm)*jacsqr(mm,m)*hinv(m,kk0));

*       Report statistics
statistics(s, k0, "estimator")  = BETA.L(k0);
statistics(s, k0, "std error")  = sqrt(cov(s,k0,k0));
statistics(s, k0, "T value")    = BETA.L(k0)/sqrt(cov(s,k0,k0));

display statistics;
```

From reported statistics, we find that adopting "robust standard errors" method lead to significantly smaller *T-value* than inverse Hessian method.

```
----    579 PARAMETER statistics  Statistics at the point

                                  estimator   std error     T value

inverse Hessian        .price        -0.084       0.001     -74.472
inverse Hessian        .miles         0.050       0.004      12.436
inverse Hessian        .width   7.115831E-4 1.685885E-4       4.221
inverse Hessian        .parking       0.075       0.002      31.969
robust estimate for cov.price        -0.084       0.010      -8.583
robust estimate for cov.miles         0.050       0.030       1.677
```

```
robust estimate for cov.width   7.115831E-4 8.582709E-4       0.829
robust estimate for cov.parking       0.075       0.012       6.097
```

(b) Estimate a second model parameterized to test the following notion: higher income beach goers care more about the length of the beach when making site choice decisions, all else equal. Report estimates and t-statistics (just use the inverse Hessian from here forward) for the model you use and briefly interpret your findings.

```
*       Record coefficient estimates for different models
set       i       "Model types"       /"Baseline", "With inc/miles","With an alternative specific constant"/;

parameter       coef(i,k)       "Coefficients estimation in model type i";
coef("baseline",k0) = BETA.L(k0);

*       Part (b)
*       Estimate a choice model including income/miles cross in the baseline model as an explanatory variable

*       Test:
*      Is it true that higher income beach goers care more about the length of the beach
*       when making site choice decisions, all else equal.

equation        obj1            "Total log-likelihood",
                llf1(n)         "Log-Likelihood definition for each respondant n";

llf1(n)..        LL(n) =e= sum(j, trip(n,j)*sum(k1, x(n,j,k1)*BETA(k1))) -
                                sum(j, trip(n,j))*log(sum(j, exp(sum(k1, x(n,j,k1)*BETA(k1)))));

model nc_beach1 /obj, llf1/;
solve nc_beach1 maximizing LOGLIK using nlp;
coef("With inc/miles", k1) = BETA.L(k1);

*       Report Jacobian and Hessian matrix
$batinclude lsa nc_beach1 nlp "maximizing LOGLIK"

alias (k1, kk1);

Jacobian(n,k1)     = LSA_DF( llf1(n), BETA(k1) );
Hessian(n,k1,kk1)   = LSA_D2F( llf1(n), BETA(k1), BETA(kk1) );

*       d2f is an upper triangular matrix
hessian(n,k1,kk1) = hessian(n,k1,kk1)$(ord(kk1) ge ord(k1)) + hessian(n,kk1,k1)$(ord(kk1) lt ord(k1));

*      Find squared Jacobian and Hessian of sum of log-likelihood functions
jacsqr(k1,kk1)      = sum(n, Jacobian(n,k1)*Jacobian(n,kk1));
Hessian_sum(k1,kk1) = sum(n, Hessian(n,k1,kk1));

*       Find inverse of Hessian and Squared jacobian matrix
$batinclude arealinverter Hessian_sum k1 kk1 hinv
$batinclude arealinverter jacsqr k1 kk1 jsinv

*       Three typies of estiamted covariance matrix of coefficients
cov("inverse Hessian",k1,kk1) = hinv(k1,kk1);

alias (k1,m1,mm1);
cov("robust estimate for cov",k1,kk1) = sum((mm1,m1), hinv(k1,mm1)*jacsqr(mm1,m1)*hinv(m1,kk1));

*       Report statistics
statistics(s, k1, "estimator")  = BETA.L(k1);
statistics(s, k1, "std error")  = sqrt(cov(s,k1,k1));
statistics(s, k1, "T value")    = BETA.L(k1)/sqrt(cov(s,k1,k1));

display statistics;
```

We can find that once we consider the intercepted term "miles_inc", the coefficient on "miles" becomes very insignificant. It seems to support the claim that higher income beach goers care more about the length of the beach when making site choice decisions, all else equal.

```
----   1043 PARAMETER statistics  Statistics at the point

                                    estimator   std error     T value
inverse Hessian        .price          -0.084       0.001     -74.308
inverse Hessian        .miles          -0.002       0.009      -0.217
inverse Hessian        .width     6.909764E-4 1.687824E-4       4.094
inverse Hessian        .parking         0.076       0.002      32.216
inverse Hessian        .miles_inc       0.009       0.001       6.323
```

(c) If you summarize the site-choice frequencies in the sample you'll see that site 9 (Wrightsville Beach) has 19% of the visits. We might want to include an alternative specific constant for site 9 because of this relatively high frequency. Consider again the specification in (a), but this time include an alternative specific constant for Wrightsville. Interpret your estimates, paying particular attention to the constant for site 9 and how estimates for the other coefficients change.

```
*        Part (c)
*        Estimate a choice model including site9 in the baseline model as an explanatory variable

*        In the sample we find site 9 (Wrightsville Beach) has 19% of the visits.
*        Include an alternative specific constant (ASC) for site 9 because of this relatively high frequency.


equation        obj2            "Total log-likelihood",
                llf2(n)         "Log-Likelihood definition for each respondant n";


llf2(n)..        LL(n) =e= sum(j, trip(n,j)*sum(k2, x(n,j,k2)*BETA(k2))) -
                                sum(j, trip(n,j))*log(sum(j, exp(sum(k2, x(n,j,k2)*BETA(k2)))));

model nc_beach2 /obj, llf2/;
solve nc_beach2 maximizing LOGLIK using nlp;

*        Record coefficient estimates
coef("With an alternative specific constant", k2) = BETA.L(k2);

*        Report Jacobian and Hessian matrix
$batinclude lsa nc_beach2 nlp "maximizing LOGLIK"

alias (k2, kk2);

Jacobian(n,k2)     = LSA_DF( llf2(n), BETA(k2) );
Hessian(n,k2,kk2)   = LSA_D2F( llf2(n), BETA(k2), BETA(kk2) );

*        d2f is an upper triangular matrix
hessian(n,k2,kk2) = hessian(n,k2,kk2)$(ord(kk2) ge ord(k2)) + hessian(n,kk2,k2)$(ord(kk2) lt ord(k2));

*       Find squared Jacobian and Hessian of sum of log-likelihood functions
jacsqr(k2,kk2)      = sum(n, Jacobian(n,k2)*Jacobian(n,kk2));
Hessian_sum(k2,kk2) = sum(n, Hessian(n,k2,kk2));

*        Find inverse of Hessian and Squared jacobian matrix
$batinclude arealinverter Hessian_sum k2 kk2 hinv
$batinclude arealinverter jacsqr k2 kk2 jsinv

*        Three typies of estiamted covariance matrix of coefficients
cov("inverse Hessian",k2,kk2) = hinv(k2,kk2);

alias (k2,m2,mm2);
cov("robust estimate for cov",k2,kk2) = sum((mm2,m2), hinv(k2,mm2)*jacsqr(mm2,m2)*hinv(m2,kk2));

parameter        statistics2(s,k,*)        "Statistics of model with ASC for site 9 ";
*        Report statistics
statistics2(s, k2, "estimator")  = BETA.L(k2);
statistics2(s, k2, "std error")  = sqrt(cov(s,k2,k2));
statistics2(s, k2, "T value")    = BETA.L(k2)/sqrt(cov(s,k2,k2));
```

```
display statistics2;
```

Comparing the following statistics to the baseline model, we find that when assign ASC for site 9, coefficient of "miles" goes up and that of "parking" goes down.

```
----   1502 PARAMETER statistics2  Statistics of model with ASC for site 9

                                 estimator   std error     T value
inverse Hessian        .price        -0.086       0.001     -74.496
inverse Hessian        .miles         0.021       0.005       4.439
inverse Hessian        .width   8.030599E-4 1.681163E-4       4.777
inverse Hessian        .parking       0.135       0.006      24.301
inverse Hessian        .site9        -0.783       0.065     -11.958
```

(d) Return to the baseline specification and consider predicting visitation probabilities for a change in parking availability. Suppose in particular that the number of parking spaces at Wrightsville Beach (site 9) is cut in half. Predict the change in probability of visits to all sites in the choice set if this were to occur, and describe your findings.

Need to compute the probabilities that each person in the sample chooses each site under baseline and the changed conditions, and then average them for a population prediction.

```
*        Part (d)
*        Question:
*       Predicting visitation probabilities for a change in parking availability.
*        Suppose in particular that the number of parking spaces at Wrightsville Beach (site 9) is cut in half.


set        w       "Index of models before/after a change in parking availability"
                    /"Original","New"/;

parameter       p(w,n,j)        "Probabilities that each person n chooses to under model w wrt to beach j"
                change(j)       "Mean probability change wrt beach j";

p("original",n,j) = exp(sum(k0, x(n,j,k0)*coef("baseline",k0)))/sum(jj, exp(sum(k0, x(n,jj,k0)*coef("baseline",k0))));

* Now cut parking in half
x(n,j,"parking")$(ord(j) = 9) = 0.5*x(n,j,"parking");

p("New",n,j) = exp(sum(k0, x(n,j,k0)*coef("baseline",k0)))/sum(jj, exp(sum(k0, x(n,jj,k0)*coef("baseline",k0))));

change(j)    = -sum(n, p("original",n,j))/card(n) + sum(n, p("New",n,j))/card(n);
display change;
```

We can predict that since the parking spots in site 9 was cut into half, people would visit other beaches more.

```
----   1526 PARAMETER change  Mean probability change wrt beach j

j1   0.002,    j2   0.005,    j3   0.003,    j4   0.002,    j5   0.005,    j6   0.009
j7   0.007,    j8   0.009,    j9  -0.077,    j10  0.009,    j11  0.006,    j12  0.006
j13  0.002,    j14  0.004,    j15  0.003,    j16  0.003,    j17  0.002
```

GAMS could also be useful in multivariate *probit* model, here is an example in generating CDF from multivariate normal distribution:

5.2.2 Limited dependent variable model example 10: Multivariate probit and GHK

Evaluation of the *probit* likelihood function usually requires the computation of trivariate normal integrals. We want to use a GHK (Geweke, Hajivassiliou, Keane) simulator to get draws from truncated multivariate normal distribution for multivaiate probit model.

Here is an example showing how the GHK simulator works. Suppose we are given

Parameters	Description
mu	$J \times 1$ vector of means for the J-dimensional normal distribution of interest
sigma	$J \times J$ covariance matrix for the normal distribution of interest
unif	$R \times J$ matrix of uniform random variables [5]
upper	$J \times 1$ vector of upper bounds for the integration

Table 11: Description of given parameters in the GHK simulator

And the output is the simulated probability p_r.

In the following numerical example, we consider the following eight-dimensional normal distribution and simulation problem:

$$x \sim N(\mu, \Sigma), \quad \mu = (0,0,0,0,0,0,0,0)',$$

$$\Sigma = \begin{bmatrix} 1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 \\ & 1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 \\ & & 1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 \\ & & & 1 & -0.1 & -0.1 & -0.1 & -0.1 \\ & & & & 1 & -0.1 & -0.1 & -0.1 \\ & & & & & 1 & -0.1 & -0.1 \\ & & & & & & 1 & -0.1 \\ & & & & & & & 1 \end{bmatrix}$$

Then, we use the GHK simulator with $R = 10000$ to simulate the following probability:

$$\Pr(x_j < 0.5), \quad \forall j \in J$$

```
$title       Simulate a Multivariant Normal CDF using GHK

*       GHK:       Geweke,Hajivassiliou, Keane (GHK) multivariate normal simulator

*       Reference:

*       Geweke, J. 1989. Bayesian inference in econometric models using
*       Monte Carlo integration. Econometrica 57: 1317-1339.

*       Hajivassiliou, V., and D. McFadden. 1998. The method of simulated scores for the
```

[5] R is the number of repetitions used in the simulation

```
*        estimation of LDV models. Econometrica 66: 863-896.

*        Keane, M. P. 1994. A computationally practical simulation estimator for panel data. Econometrica 62: 95-116.

*        GHK simulator: get draws from truncated multivariate normal
*        distribution.

*        Construct a procedure in GAMS that does GHK for a single probability
*        simulation.

set     j        "Jointly distributed variables" /j1*j8/,
        r        "Repetitions" /r1*r1000/;

alias  (i,j,jj), (r,rr);

*        Include library routines for the normal cumulative distribution
*        function and its inverse  (NB: stodclib is required by stolib).

$funclibin stolib stodclib

*        Name two of these functions:

function cdfnorm  /stolib.cdfnormal/;
function icdfnorm /stolib.icdfnormal/;


$onechov >"%gams.scrdir%inverse.scr"
$stitle        Routine to Evaluate the Inverse:

*       Usage:        $batinclude inverse <domain> <matrix> <output>

$if not declared %1 $abort      "Bad argument to inverse.gms. First argument has not been declared."
$if not settype  %1 $abort      "Bad argument to inverse.gms. First argument must be one dimensional set."
$if not dimension 1 %1 $abort   "Bad argument to inverse.gms. First argument must be one dimensional set."

$if not declared %2 $abort     "Bad argument to inverse.gms. Second argument has not been declared."
$if not partype  %2 $abort     "Bad argument to inverse.gms. Second argument must be two dimensional parameter."
$if not dimension 2 %2 $abort  "Bad argument to inverse.gms. Second argument must be two dimensional parameter."


$if not declared %3 $abort     "Bad argument to inverse.gms. Third argument has not been declared."
$if not partype  %3 $abort     "Bad argument to inverse.gms. Third argument must be two dimensional parameter."
$if not dimension 2 %3 $abort  "Bad argument to inverse.gms. Third argument must be two dimensional parameter."

execute_unload '%gams.scrdir%inverse_data.scr', %1, %2;
execute        'invert "%gams.scrdir%inverse_data.scr" %1 %2 "%gams.scrdir%inverse_out.scr" %3';
execute_load   "%gams.scrdir%inverse_out.scr", %3;
$offecho

*        Given:          a(j), b(j)      Upper and lower bounds
*                        mu(j)           Mean value (exogenous)
*                        sigma(j,jj)     Variance-covariance matrix
*
*        Determine:      pghk = prob( a(j) <= x(j) <= b(j), for all j )

*        Strategy:       Draw a bunch of X's in the given box, evaluating
*                        the probability of such draws, and use the
*                        average of this probability.

parameter
        mu(j)           "Mean values",
        sigma(j,j)      "Variance-covariance matrix (symmetric)",
        a(j)            "Lower bound of simulated values",
        b(j)            "Upper bound of simulated values",
        sigmam(j)       "Marginal sigma";

*        Take the assumed variance-covariance matrix:
mu(j)       = 0;
sigma(j,jj) = -0.1;
sigma(j,j)  = 1;
a(j)        = -20;
b(j)        = 0.5;

parameter
        tmp(j,j)        "Temporary matrix",
        tmp_inv(j,j)    "Temporary inverse matrix",
        sigmainv(j,j,j) "Inverse matrices at different levels";

set     iinv(i)         "Dimensions to include in the inverse";

*        Add one element at a time to set i.  Sets
```

```
*       i and j are the same:
alias (iinv, iinv2);

loop(j,
    iinv(j) = yes;
        tmp(iinv,iinv2) = sigma(iinv,iinv2);
$batinclude "%gams.scrdir%inverse.scr" iinv tmp tmp_inv
display tmp, tmp_inv;
        sigmainv(j,i,jj) = tmp_inv(i,jj);
);

option sigmainv:3:1:1;

sigmam(j) = sigma(j,j) - sum((i,jj), sigma(j,i)*sigmainv(j-1,i,jj)*sigma(jj,j));

parameter       unif(j,r)       "Random realizations for the current repetition";

unif(j,r) = uniform(0,1);

variable        X(j,r)          "Random realization of the jth variable",
                P(j,r)          "Conditional probability that the jth draw is in bounds",
                PI              "Simulated probability",
                MUC(j,r)        "Conditional mean of the jth draw";

equations        pidef, mucdef, mdef, xdef;

pidef..         PI =e= sum(r, prod(j, P(j,r)))/card(r);;

mucdef(j,r)..   MUC(j,r) =e= mu(j) + sum((i,jj)$sigmainv(j-1,i,jj), sigma(j,i)*sigmainv(j-1,i,jj)*(X(jj,r)-mu(jj)));

mdef(j,r)..     P(j,r) =e= cdfnorm(b(j)-MUC(j,r), 0, sqrt(sigmam(j))) -
                           cdfnorm(a(j)-MUC(j,r), 0, sqrt(sigmam(j)))$(1/a(j));

xdef(j,r)..     X(j,r) =e= MUC(j,r) + icdfnorm( unif(j,r)*P(j,r), 0, sqrt(sigmam(j)));

model sample /pidef, mucdef, mdef, xdef/;

MUC.L(j,r) = 0.5;
P.L(j,r) = 0.5;
solve sample using cns;

display PI.L;
```

We find simulated CDF: $PI = 0.0017$.

5.3 Other MLE examples

5.3.1 Example: another application of multiplicative heteroscedasticity approach

There are a variety of ways to control for the effect of heteroscedasticity in the estimation of a linear regression model. Under the multiplicative heteroscedasticity approach we have $y_t = x_t\beta + \epsilon_t$ *where* $E(\epsilon_t^2) = \sigma_t^2 = \exp(D_t\alpha)$, D_t *is a* $(1 \times S)$ *vector containing the* t^{th} *observation on S non-stochastic explanatory variables and* α *is a* $(S \times 1)$ *parameter vector. Thus, the error variance changes across observation and the error covariances across observations are 0. The observation specific error variance depends on a set of exogenous variables.*

Assume you work for the Energy Information Agency of the U.S. Department of Energy. You are interested in examining trends in annual U.S. gasoline consumption. You collect data for 52 years (i.e., 1953-2004) for your econometric analysis. For this assignment I would like you to use the BHHH maximum likelihood estimation algorithm to estimate the coefficients of the multiplicative heteroscedastic model as discussed in Greene, p. 554-557

and by JG, p. 538-540. You specify the dynamic model of U.S. per capita demand for gasoline via the following:

$$\log \frac{Q_{Gas_t}}{Pop_t} = \beta_1 + \beta_2 \log(GasP_t) + \beta_3 \log(Pc\_Inc_t) + \beta_4 \log(PNC_t) + \beta_5 \log(PUC_t) + \beta_6 Shock_{73} + \beta_7 Shock_{79} + \beta_8 Recess + \gamma \log \frac{Q_{Gas_{t-1}}}{Pop_{t-1}} + \epsilon_t, \quad (5.3.1)$$

where $\epsilon_t \sim N(0, \sigma_t^2)$, $\sigma_t^2 = \exp(D_t\alpha)$ and D_t is composed of the variables: (i) An intercept, (ii) PC_Inc_t, (iii) Shock73, (iv) Shock79 and (v) Recess.

The following table is used to provide definitions of the variables used in equation (5.3.1). This data can be obtained by accessing the US_gas_use.xlsx file located on the class website. The order of the variables may not be in the order listed above.

Variable	Description
Q_Gas	Total Quantity of Gasoline Consumed
GasP	Gasoline Price Index
PC_Inc	Per Capita Disposable Income
PNC	Price Index of New Cars
PUC	Price Index of Used Cars
Shock73	Dummy variable identifying the effects of the 1973 Oil Embargo = 1 from 1973-1978, 0 otherwise
Shock79	Dummy variable identifying the effects of the 1979 Price Increase = 1 from 1979-Present, 0 otherwise
Recess	Dummy variable identifying years in which the U.S. economy was in a recession = 1 for recession years, 0 otherwise
Pop	Population of the U.S. (in millions)

Table 12: Description of Variables Used in the equation (5.3.1)

(a) Estimate the coefficients of equation (5.3.1) using maximum likelihood techniques. Present the usual parameter and regression-based statistics. Using the information in Greene, p.556 undertake Wald, Likelihood Ratio and Lagrangian Multiplier tests of the null hypothesis that per capita gasoline demand is characterized as having a homoscedastic error structure.

In part(a), applying the multiplicative heteroscedasticity approach, we have log-likelihood function

$$logL = \sum_t \left[\frac{-log(2\pi)}{2} - \frac{D_{t\times k}\alpha_k}{2} - \frac{\epsilon_t^2 exp(-D_{t\times k}\alpha_k)}{2} \right],$$

in which D is the explanatory variables, ϵ is the error term and α is the unknown parameter vector.

As usual, we use MLE to estimate unknown coefficients:

```
$title MLE_multiplicative heteroscedastic model
*        Examining trends in annual U.S. gasoline consumption
*        Specify the dynamic model of U.S. per capita demand for gasoline via the following:

*        In which MU(t) follows N(0,sqr(sigma)),
*        while sqr(sigma)=exp(D_t*ALPHA), and D_t is composed of the variables:
*        1) an intercept 2) pc_inc(t) 3) shock73 4) shock79 5) recess

*        Note:  The error variance changes across observation and
*       the error covariances across observations are 0
set             t               "Observation years"                             /1953*2004/,
                j               "Index of all coefficients, includes k and i"
                                /"constant","lgasP","lpc_inc","lpnc","lpuc","s73","s79","rec",
                                 "ldepend_lag","cons","pc_inc","shock73","shock79","recess"/,
                k(j)            "Index for variables in data matrix D"
                                /"cons","pc_inc","shock73","shock79","recess"/,
                i(j)            "Index for coefficients BETA"
                                /"constant","lgasP","lpc_inc","lpnc","lpuc","s73","s79","rec","ldepend_lag"/;

alias (j,jj), (k,kk), (t,w), (i,ii);

set             g(t)            "Years when lagged value are feasible";
g(t) = yes$(ord(t) > 1);

parameter       data(t,*)       "Source data";

$call gdxxrw US_gas_use_gams.xlsx par=data rng=A1:P53 cdim=1 rdim=1 checkDate
$gdxin US_gas_use_gams.gdx
$loaddc data
$gdxin

parameter       y(t)            "Log(totol quantity of gasline consumed per capita)",
                x(t,j)          "Right hand side variables",
                d(t,j)          "Table whose (t,k)th element is the t_th observation of explanatory variable k";

data(t,"cons") = 1;

*        Dependent variable y
y(t) = log(data(t,"q_gas")/data(t,"pop"));
x(t,"constant") = data(t,"cons");
*        Log of gas price index
x(t,"lgasP") = log(data(t,"gasp"));
*        Log of per capita disposable income
x(t,"lpc_inc") = log(data(t,"pc_inc"));
*        Log of price index of new cars
x(t,"lpnc") = log(data(t,"pnc"));
*        Log of price index of used cars
x(t,"lpuc") = log(data(t,"puc"));
*        1 if affected by 1973-1978 shock
x(t,"s73") = data(t,"shock73");
*        1 if affected by price increase from 1979 on
x(t,"s79") = data(t,"shock79");
*        1 for years when economy in recession
x(t,"rec") = data(t,"recess");
*        Log of y(t-1)
x(g(t),"ldepend_lag") = y(t-1);
d(t,k) = data(t,k);

*        part(a)
variable        LOGLIK          "Log likelihood",
                COEF(j)         "All coefficients to be estimated";

equation        obj_llf         "Objective for log-likelihood maximization";


obj_llf..       LOGLIK =e= sum(g(t), -0.5*log(2*pi) - 0.5*sum(k, d(t,k)*COEF(k))
                                -0.5*sqr(y(t) - sum(i, COEF(i)*x(t,i)))*exp(-sum(k, d(t,k)*COEF(k))));

model MLE /obj_llf/;
solve MLE using nlp maximizing LOGLIK;

parameter       alpha(j)        "Coefficients for projecting the sum of squared errors onto the space spanned by d(t,k)",
                beta(j)         "Coefficients in the linear model";
alpha(k) = COEF.L(k);
beta(i)  = COEF.L(i);

*        Report estimation statistics
```

```
parameter       Hessian(j,jj)   "Hessian of constraints wrt coefficients",
                hinv(j,jj)      "Inverse of ssehess",
                cov(j,jj)       "Covariance matrix",
                statistics(j,*) "Statistics at the point";

*       Request the Hessian matrix
$batinclude lsa mle nlp "maximizing LOGLIK"
hessian(j,jj)   = LSA_D2F( obj_llf, COEF(j), COEF(jj) );

*       d2f is an upper triangular matrix
hessian(j,jj) = hessian(j,jj)$(ord(jj) ge ord(j)) + hessian(jj,j)$(ord(jj) lt ord(j));

*       Find inverse of Hessian
$batinclude arealinverter Hessian j jj hinv

*       Define variance-covariance matrix
cov(j,jj) = hinv(j,jj);

*       List estimation statistics
statistics(j, "estimator")  = COEF.L(j);
statistics(j, "std error")  = sqrt(cov(j,j));
statistics(j, "T value")    = COEF.L(j)/sqrt(cov(j,j));

*       Use the BETAREG function:
statistics(j, "P value")    = BETAREG( (card(g) - card(j))/
                                       (card(g) - card(j) + sqr(statistics(j, "T value"))),
                                       (card(g) - card(j))/2, 0.5);
display statistics;
```

We find regression statistics as follows:

```
----    327 PARAMETER statistics  Statistics at the point

              estimator   std error     T value     P value

constant      -2.764378    0.855668   -3.230665    0.002594
lgasP         -0.092877    0.024311   -3.820359    0.000493
lpc_inc        0.163596    0.053785    3.041681    0.004308
lpnc          -0.115302    0.051183   -2.252740    0.030294
lpuc           0.082756    0.039293    2.106120    0.042038
s73            0.018302    0.011281    1.622386    0.113212
s79            0.030171    0.028465    1.059954    0.296036
rec           -0.015456    0.009013   -1.714853    0.094738
ldepend_lag    0.861701    0.042968   20.054404 1.80243E-21
cons          -6.421049    1.727853   -3.716201    0.000666
pc_inc        -0.000182    0.000165   -1.108316    0.274878
shock73        0.195214    2.215881    0.088098    0.930274
shock79        0.945237    2.023837    0.467052    0.643202
recess         1.658815    1.099044    1.509325    0.139709
```

We use the Wald test, the Lagrange multiplier (LM) test and the likelihood ratio (LR) test to check whether $\mathbb{H}_0$: model with homoscedastic errors is true. More details can be found in Greene's econometrics textbook.

```
*       Wald Test
*       Define a subset of k to exclude the intercept index "cons"
set             z(k)                    "Subset of set k which does not include 'cons'";
z(k) = yes$(ord(k)>1);

alias (z,zz);

parameter       sigma2_hat              "Estimate of sample variance",
                err(t)                  "Optimal error terms in the linear model",
                err2_dif(t)             "Difference of Squared error term and its mean",
                d2(k,kk)                "Square of d(t,k)",
```

```
                d2_res(k,kk)            "d2 without first row and first column",
                d2inv(k,kk)             "Inverse of d2",
                d2_resinv(k,kk)         "Inverse of d2_res",
                df_homo                 "Degree of freedom",
                alpha_homo              "Type 1 error level"    /0.05/
                lambda_LM               "Lagrange Multiplier (LM) test statistics for homoscedasticitic error structure",
                p_LM                    "P value of LM test based on H0",
                lambda_LR               "Likelihood ratio (LR) test statistics for homoscedasticitic error structure",
                p_LR                    "P value of Likelihood Ratio test based on H0",
                lambda_Wald             "Wald test statistics for homoscedasticitic error structure"
                p_wald                  "P value of Wald test based on H0";

df_homo = card(k) - 1;
d2(k,kk) = sum(g, d(g,k)*d(g,kk));

*       Find inverse of squared data matrix
$batinclude arealinverter d2 k kk d2inv

*       Pick the (2^5 row, 2^5 column) of inverse matrix of d square
d2_res(z(k),zz(kk)) = d2inv(k,kk)$(sameas (z,k) and sameas (zz,kk));

*       Find inverse of d2_res
$batinclude arealinverter d2_res z zz d2_resinv

*       Wald test statistics
lambda_Wald = sum((z,zz), alpha(z)*d2_resinv(z,zz)*alpha(zz));
display lambda_Wald;

*       Use gammarreg function to define Wald statistic
p_wald =  1 - gammareg(lambda_Wald/2,df_homo/2);
display p_wald;

if ((p_wald lt alpha_homo), display "Reject H0: Homoscedastic-errors";
                    else   display "Fail to reject H0: Homoscedastic-errors";);

*       Lagrange Multiplier Test
*       In the linear model, find error term at the optimal parameter values
err(g) = y(g) - sum(i, x(g,i)*beta(i));

*       Estimate the sample variance
*       Here I used a more "accurate form" of variance estimate
sigma2_hat = sum(g, sqr(err(g)))/(card(g) - card(i));

*       Find the deviation of each squared error term from the estimate of sample variance
err2_dif(g) = sqr(err(g)) - sigma2_hat;

*       LM test statistics
lambda_LM = sum((k,kk), sum(g, err2_dif(g)*d(g,k))*d2inv(k,kk) *
            sum(g, err2_dif(g)*d(g,kk)))/(2*sqr(sigma2_hat));

display lambda_LM;

*       Use gammareg function to define p value of LM test statistic
p_LM =  1 - gammareg(lambda_LM/2,df_homo/2);
display p_LM;

if ((p_LM lt alpha_homo), display "Reject H0: Homoscedastic-errors";
                    else  display "Fail to reject H0: Homoscedastic-errors";);

*       Likelihood Ratio Test
*       LR test statistics
lambda_LR = card(g)*log(sigma2_hat) - sum((g,k), d(g,k)*alpha(k));
display lambda_LR;

*       Use gammareg function to define p value of LR test statistic
p_LR =  1 - gammareg(lambda_LR/2,df_homo/2);
display p_LR;

if ((p_LR lt alpha_homo), display "Reject H0: Homoscedastic-errors";
                    else  display "Fail to reject H0: Homoscedastic-errors";);
```

All three tests rejected the null at 5% level:

```
----    528 PARAMETER lambda_Wald          =    31.533382  Wald test statistics for homoscedasticitic error structure
```

```
----    532 PARAMETER p_wald              =        0.000002  P value of Wald test based on H0

----    535 Reject H0: Homoscedastic-errors

----    553 PARAMETER lambda_LM           =       23.101040  LM test statistics for homoscedasticitic error structure

----    557 PARAMETER p_LM                =        0.000121  P value of LM test based on H0

----    560 Reject H0: Homoscedastic-errors

----    566 PARAMETER lambda_LR           =       30.975641  LR test statistics for homoscedasticitic error structure

----    570 PARAMETER p_LR                =        0.000003  P value of LR test based on H0

----    573 Reject H0: Homoscedastic-errors
```

(b) Given the functional form shown in equation (5.3.1), we know that the short run price and income elasticities are β_1 and β_1, respectively. Due to the presence of lagged gasoline consumption as an exogenous variable, the long run price ξ_P and income ξ_I elasticities are given by the following:

$$\xi_P = \frac{\beta_1}{1-\gamma} \quad \xi_I = \frac{\beta_2}{1-\gamma}, \tag{5.3.2}$$

Test the null hypothesis that $\xi_P \leq -1$. Then test the null hypothesis that $\xi_I = 1$. Finally, test the joint hypothesis that $\xi_P \leq -1$ and $\xi_I = 1$.

To test the values of elasticities, we need to derive the gradients matrix around the estimated values in an MCP model as usual:

```
*        Part (b)
parameter       ksi_p           "Long run price elasticity",
                ksi_i           "Long run income elasticity",
                v_p             "Variance estimation of ksi_p",
                v_i             "Variance estimation of ksi_i",
                grad_p(j)       "Gradient for price elasticity function",
                grad_i(j)       "Gradient for income elasticity function",
                z_p             "Z statistics for ksi_p based on H0",
                z_i             "Z statistics for ksi_i based on H0",
                w_ksi           "Wald stat. based on joint H0",
                pval_p          "P value for ksi_p z test",
                pval_i          "P value for ksi_i z test",
                pval_w          "P value for joint Wald test",
                alpha_ksi       "Level of Type 1 error level"    /0.05/;

*       Define long run price and income elasticities
ksi_p = COEF.L("lgasP")/(1 - COEF.L("ldepend_lag"));
ksi_i = COEF.L("lpc_inc")/(1 - COEF.L("ldepend_lag"));

variable        S_P             "Definition of ksi_p",
                S_I             "Definition of ksi_i";

equation        def_p           "Define S_P",
                def_i           "Define S_I";

def_p..         S_P =e= COEF("lgasP")/(1 - COEF("ldepend_lag"));
def_i..         S_I =e= COEF("lpc_inc")/(1 - COEF("ldepend_lag"));

*        Fix the coefficients at its estimates' level
COEF.FX(j) = COEF.L(j);
model elas_p /def_p, def_i/;
solve elas_p using mcp;

*        Report Jacobian matrix
parameter       grad_p(j)        "Gradient of S_P wrt COEF"
                grad_i(j)        "Gradient of S_I wrt COEF"

*        Find gradients of elasticities wrt coefficients
```

```
$batinclude lsa elas_p mcp
grad_p(j) = LSA_DF( def_p, COEF(j) );
grad_i(j) = LSA_DF( def_i, COEF(j) );

*       Find variance estimation for price and income elasticity
v_p = sum((j,jj), grad_p(j)*cov(j,jj)*grad_p(jj));
v_i = sum((j,jj), grad_i(j)*cov(j,jj)*grad_i(jj));

*       Based on H0: ksi_p<=-1, z test
z_p    = (ksi_p + 1)/sqrt(v_p);
display z_p;
pval_p = 1 - errorf(abs(z_p));

if ((pval_p lt alpha_ksi), display "Reject H0: long run price elasticity  <= -1";
                     else display "Fail to reject H0: long run price elasticity <= -1";);

*       Based on H0: ksi_i = 1, z test
z_i = (ksi_i - 1)/sqrt(v_i);
display z_i;
pval_i = 1 - errorf(abs(z_i));

if ((pval_i lt alpha_ksi/2), display "Reject H0: long run income elasticity  = 1";
                      else  display "Fail to reject H0: long run income elasticity = 1";);

*       Wald test based on H0: ksi_p <= -1 and ksi_i = 1
set             s                       "Index for elasticities"            /price,income/;
alias           (s,ss);

parameter       r_ksi(s)                "Restriction coefficents in testing H0, (ksi_p + 1, ksi_i -1)"
                grad_ksi(s,j)           "Gradients for price and income elasticity functions",
                w_ksi                   "Wald test statistics",
                pval_w_ksi              "P value",
                rs_cov_ksi(s,ss)        "Denominator of Wald test statistic, r*cov*r",
                inv_v_ksi(s,ss)         "Inverse of adjusted  covariance matrix",
                df_w_ksi                "Degrees of freedom"                    /2/;

*       List r_ksi
loop(s, if((ord(s) = 1),r_ksi(s) = ksi_p + 1;
        else            r_ksi(s) = ksi_i - 1;););

*       grad_ksi matrix is composed of two rows
*       First row, gradients of price elasticity function
*       Second row, gradients of income elasticity function
loop((s,j), if((ord(s)=1),grad_ksi(s,j) = grad_p(j);
            else          grad_ksi(s,j) = grad_i(j);););

*       Adjusted covariance
rs_cov_ksi(s,ss) = sum((j,jj), grad_ksi(s,j)*cov(j,jj)*grad_ksi(ss,jj));

*       Find inverse of adjusted cov matrix
$batinclude arealinverter rs_cov_ksi s ss inv_v_ksi

*       Wald test statistics
w_ksi = sum((s,ss), r_ksi(s)*inv_v_ksi(s,ss)*r_ksi(ss));

*       P value
pval_w_ksi = 1 - gammareg(w_ksi/2,df_w_ksi/2);

display w_ksi;

if ((pval_w_ksi lt alpha_ksi), display "Reject H_0: ksi_p<=-1 and ksi_i = 1";
                        else  display "Fail to reject H_0: ksi_p<= -1 and ksi_i = 1";);
```

We find that Z statistic is not large enough to reject the null for the value of long run price and income elasticities individually, however, there is statistically enough evidence to reject the joint hypothesis that $\xi_P \leq -1$ and $\xi_I = 1$.

```
----    850 PARAMETER z_p                 =     1.264176  Z statistics for ksi_p based on
                                                          H0

----    854 Fail to reject H0: long run price elasticity <= -1

----    858 PARAMETER z_i                 =     0.528467  Z statistics for ksi_i based on
```

```
                                                    H0

----    862 Fail to reject H0: long run income elasticity = 1

----    976 PARAMETER w_ksi                =    10.509754  Wald stat. based on joint H0

----    978 Reject H_0: ksi_p<=-1 and ksi_i = 1
```

6 Conclusion

In this paper, we combine existing GAMS resources in order to retrieve key statistics in econometrics study. Several classical parametric estimation problems, including nonlinear least squares model, limited dependent variable model and some other models could be solved using maximum likelihood estimation are listed as examples in the paper.

Although the amount of variety in solved problems is limited, our main goal in this paper is to demonstrate the advantage of algebraic modeling languages such as GAMS in applied optimization. More recent econometrics applications, say mixed logit and multivariate probit model will be studied in some follow-up work based on this.

7 Literature

"Econometrics Analysis" with William Greene, 2011, Prentice Hall, 7 edition.

"Estimating Regression Models with Multiplicative Heteroscedasticity", with Andrew C. Harvey, 1976, Econometrica, 44(3), pp.461-465.

"A Specification Test for the Multinomial Logit Model", with Jerry A. Hausman and Daniel McFadden, 1984, Econometrica, 52, pp.1219-1240.

"Sample Selection Bias as a Specification Error", with James J. Heckman, 1979, Econometrica 47, pp. 153-161.

"Multinomial and Conditional Logit Discrete-Choice Models in Demography", with Saul D. Hoffman, Greg J. Duncan, 1988, Demography, 25(3), pp. 415-427.

"Introduction to the Theory and Practice of Econometrics", with G.G. Judge, R.C. Hill, W.E. Griffths, H. Lutkepohl,and T.C. Lee, 1988, 2nd ed., John Wiley and Sons, New York.

"New Special Functions in GAMS", http://www.amsterdamoptimization.com/pdf/specfun.pdf, with Erwin Kalvelagen, 2004.

"Least Squares Calculations with GAMS", http://www.amsterdamoptimization.com/pdf/ols.pdf, with Erwin Kalvelagen, 2007.

"A Nonlinear Regression Solver for GAMS", http://www.amsterdamoptimization.com/pdf/nlregression.pdf, with Erwin Kalvelagen, 2007.

"Econometrics", http://www.ssc.wisc.edu/ bhansen/econometrics, with Bruce Hansen, 2014.

"McCarl GAMS User Guide", http://www.gams.com/mccarl/mccarlhtml, with Bruce A. McCarl, 2005.

"GAMS – A Users' Guide", http://www.gams.com/dd/docs/bigdocs/GAMSUsersGuide.pdf, with Richard E. Rosenthal, 2013.

8 GAMS tips

GAMS language basics:

- $ Conditionals

  A $ condition is placed in GAMS statements and causes an action to occur if the conditional is true. The basic $ conditional form is

  ```
  term$logical condition
  ```

  which says include the item term only if the logical condition is true. The forms of logical conditions are reviewed below. For now I will use a conditional that a named item be nonzero for illustration

  ```
  X$(y gt 0) = 10;
  ```

  In this case the conditional says set $X = 10$ if the scalar y is greater than zero.

  For further information about a conditional on the left/right hand side, etc., see page 321-325 of McCarl GAMS user Guide(MGUG).

- Including External Files

  GAMS may include external files. This may be done with and without substitution of some items within the file. There are also special provisions regarding inclusion of comma-delimited files.

  – Inclusion without arguments

    When files of GAMS instructions or data are to be incorporated into a GAMS program and one simply wants to incorporate the file as is one uses the GAMS dollar command:

    ```
    $include
    ```

Otherwise one may wish to specify some arguments and use the include with arguments commands. See below:

- Include with arguments

  There are variants of the include command which permit insertion of some user defined arguments in the file to be included. Three of these variants exist Batinclude, Libinclude and Sysinclude.

  The basic syntax for the command $Batinclude which is used intensively in this paper is

```
$Batinclude externalfilename argument1 argument2...
```

- Suppressing the listing of include files

  Sometimes the files included are large files that one really does not wish to be included in the echo print within the LST file.

- Redefining the location of include files - Idir

  The directory in which $Include files are expected to be located can be altered. This is done by using the IDIR command line parameter in which case the named file is looked for first then one with a *.gms* extension.

Check page 492 of MGUG for details.

- Macros in GAMS

  We only introduce the basic definition of Macros. It takes the form

```
$macro name macro body
$macro name(arg1,arg3,arg2,..) macro body with tokens arg1,..
```

  The name of the macro has to be unique, similar to other GAMS data types like sets and coefficients. A (following immediately the macro name starts the list of replacement arguments and a) ends it. These will be expanded by the arguments in parentheses in a call to the macro. Note that the items to replace in the macro body follow the standard GAMS identifier conventions. If you are interested, see page 534 of MGUG.

- Using GAMS Data Exchange or GDX files to interface with other programs

  - Spreadsheets: gdxxrw

    Gdxxrw is a GAMS Corporation developed utility to read and write Excel spreadsheet data using GDX files. GDXXRW can work with *.xls*, *.xlsb*, *.xlsx* and *.xlsm* formats, GDXXRW will write an Excel files as a .xlsx file unless a different file extension is specified for the output file. Gdxxrw uses command line arguments. The basic calling sequence is

```
Gdxxrw Inputfile Output=filename options
```

where

Inputfile must have an extension and tells the name of a file to read from or write to. The read from a spreadsheet occurs if the extension is a workbook extension (*.xls*, *.xlsb*, *.xlsx*, *.xlsm*, *.wk1*, *.wk2*, *.wk3* and *.dbf*). The write to the spreadsheet occurs if the file has a GDX extension. This can also be entered as I=inputfile.

Output=filename is an optional specification of the name of target workbook or GDX file where the output is to be written. If not present the file name will be the input file name with a workbook (*.xls* or *.xlsx*, *.xlsb*, *.xlsm*) or GDX extension and no path. A shortcut entry occurs with O=filename.

Options are a number of possible options.

For further information, see the Using GAMS Data Exchange or GDX files chapter in MGUG.