ECONOMETRICS II, Fall, 2017
Bent E. Sørensen

Midterm Exam 2 - November 6, 2017

Each sub-question in the following carries equal weight.

1. (24%) Consider the general selection model

$$y_i = \alpha + \beta x_i + u_i \, ,$$

and

$$z_i = \omega + \gamma w_i + v_i \, ,$$

where x_i and w_i are exogenous regressors, the error terms u_i and v_i are mean zero and normally distributed and uncorrelated with regressors and satisfies the standard conditions (no autocorrelation/heteroskedasticity). We normalize the variance of v to unity and assume that u and v are correlated with a correlation coefficient of ρ. We observe y only if $Z = 1$, where $Z = 1$ if $z > 0$ and $Z = 0$, otherwise. We observe Z, whether it is zero or one.

a) Explain why, if we have 2 generic normally distributed variables X and Y, we can write X as $\gamma Y + u$, where u is independent of Y and γ is a constant. (You have to argue/demonstrate mathematically why u is independent of Y.)
b) If X follows a standard normal distribution, find $E(X|X > a)$.
c) Using the results from a) and b), write down the likelihood function (or log-likelihood) for the general selection model.

2. (30%) Consider the Matlab programs below and fill in the missing part/answering the questions.

3. (16%) Consider the model:

$$y_1 = \alpha_1 x_1 + \alpha_2 x_2 + u_1,$$

Assuming that you have estimated the model by OLS and have the relevant parameter estimates, write down the Wald test for $\alpha_1 = \sqrt{\alpha_2}$.

4. (18%) Consider the simultaneous equation model:

$$(1) \quad y_1 = \alpha_1 y_2 + \alpha_2 \, x_1 + u_1 \, ,$$

and

$$(2) \quad y_2 = \alpha_3 y_1 + \alpha_4 x_1 + \alpha_5 x_2 + u_2 \, .$$

a) Explain *how* you can use Two-Stage Least Squares (2SLS) to estimate equation (1).
b) Explain *why* the 2SLS estimator can consistently estimate the equation for y_1, but not the one for y_2.
c) Explain in words what the Three-Stage Least Squares estimator does.

4. (12%) a) Explain what a duration model is, including a definition of the survivor function and the hazard function.
b) Derive the hazard function for the exponential duration model.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Econometrics 2
% Midterm 2
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Question 1
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
clc

global w z N

% Set the true parameters and placeholders for results.

N = 500;                                                                          % Number of observa
beta0 = 0.5;                                                                      % Intercept for y.
beta1 = 3;                                                                        % Coefficient on X
sigmau = 2;                                                                       % Standard deviatio
gamma0 = 1;                                                                       % Intercept for z0
gamma1 = 4;                                                                       % Coefficient on W
rho = 0.8;                                                                        % Correlation betwe

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Generate the data. The latent  variable model for a is:
%
%           a0(j) = beta0 + beta1*x(j) + u(j)
%
% where a(j) = a0(j) if z0(j) > 0 and unobserved otherwise.
%
% The latent variable model for z is:
%
%           z0(j) = gamma0 + gamma1*w(j) + v(j)
%
% where z(j) = 1 if z0(j) > 0 and 0 otherwise.
```

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

e = mvnrnd([0; 0],[sigmau^2 rho*sigmau; rho*sigmau 1],N);                 % Correlated error

u = e(:,1);                                                               % Error terms for a
v = e(:,2);                                                               % Error terms for c

x = ((1:N)'/N).*normrnd(0,1,N,1);
w = ((1:N)'/N).*normrnd(0,1,N,1);

z0 = gamma0*ones(size(w,1),1) + gamma1*w + v;                             % Latent variable z
z = double((z0 > 0));                                                     % Observed z.

a0 = beta0*ones(size(x,1),1) + beta1*x + u;                               % Latent variable a

x(z0<=0) = [];                                                            % Gets rid of the r

a = a0;
a(z0<=0) = [];                                                            % Observed a.

% Step 1: Probit.

b0 = [1 1];                                                               % Initial values.

options = optimset('Display','off');                                      % Turn off display
[b_mle, ~, ~, ~, ~, ~] = fminunc('logl_prob', b0, options);               % Minimization.

% Step 2: OLS.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                         FILL IN MISSING CODE
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

(Write as much as you can, you will get partial points for explaining what needs to be done eve
if you forget some of the formulas)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%
% Question 2
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global y x1 x2 T

% Set the true parameters and placeholders for results.

T = 1000;                                                                    % Sample size.

beta0 = 0.5;                                                                 % Intercept.
beta1 = 3;                                                                   % Coefficient on X1
beta2 = 0;                                                                   % Coefficient on X2

sigma = 1;                                                                   % Standard deviatio

% Generate the data.

x1 = ((1:1:T)'/T).*normrnd(0,1,T,1);
x2 = ones(T,1) + 2.*x1 + 5.*normrnd(0,1,T,1);
x = [ones(T,1) x1 x2];

u = sigma.*randn(T,1);

y = beta0 + beta1.*x1 + beta2.*x2 + u;

% Maximum likelihood estimation.

options = optimset('Display','off');

b0 = [1 1 1 1];
[b1_mle,~,~,~,~,hess] = fminunc('logl_unres', b0, options);                  % Unrestricted ML e

b0_res = [1 1 1];
[b2_mle] = fminunc('logl_res', b0_res, options);                             % Restricted ML es


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```
% Lagrange Multiplier Test. Null is:
%
%                       H0: beta2 = 0
%
% How would logliklihood functino codes and the following code below change
% if instead the null is:
%
%                       H0: beta2 = 0 and beta1 = 0
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

grad_LM = zeros(4,1);                                                     % Gradient of unres
grad_LM(1,1) = sum((y - ones(size(y,1),1).*b2_mle(1) - ...
                                                b2_mle(2)*x1)/b2_mle(3)^2);
grad_LM(2,1) = sum(x1.*(y - ones(size(y,1),1).*b2_mle(1) - ...
                                               b2_mle(2)*x1)./b2_mle(3)^2);
grad_LM(3,1) = sum(x2.*(y - ones(size(y,1),1).*b2_mle(1) - ...
                                                b2_mle(2)*x1)/b2_mle(3)^2);
grad_LM(4,1) = sum(ones(size(y,1),1).*(-1/b2_mle(3)) + ...
                            (y - ones(size(y,1),1).*b2_mle(1) - ...
                                             b2_mle(2)*x1).^2/b2_mle(3)^3);

vmat_IM = [b2_mle(3)*inv(x'*x) zeros(3,1);                                % IM estimator eval
           zeros(1,3)    b2_mle(3)/(2*T)];

LM = grad_LM'*vmat_IM*grad_LM;                                            % LM statistic.
pval_LM = chi2cdf(LM,1,'upper');                                          % p-value.

    if pval_LM >= 0.05                                                    % Rejection decisio
        reject_LM = 0;
    else
        reject_LM = 1;
    end
```