ECONOMETRICS II, Fall 2020.
Bent E. Sørensen

Midterm Exam 1—Monday, September 28th, 2020. Please scan you answer and email me before logging out from Zoom. Also, verify that the scan is readable before you send it. Four questions. Each sub-question in the following carries equal weight except when otherwise noted.

1. (40%)
Consider the model

$$y_t = \beta x_t + a\, y_{t-1} + u_t\,,$$

where the error term is white noise, x_t is fixed numbers (in an application, we would condition on the x variables), and a is numerically smaller than one.

a) Find (derive) the mean, variance, and autocovariance of y_t assuming that $y_t - \beta x_t$ follows a stationary Normal process.

b) Write down the likelihood function for a sample $t = 1, .., T$ where you condition on the first observation.

c) Write down the first term for $t = 1$ in the likelihood function.

d) Fill in the missing Matlab code, using the symbols from the program (you can just write that line on your answer sheet).

```
This code estimates, using maximum likelihood, the AR(1) process
                                                                    ,
Set the parameters.
There is 1 simulation with 300 observations. Set  and .
clc
clear

global y T

T = 300;                                          % Number of time periods.
beta0 = 0;                                        % Beta 0.
beta1 = 0.5;                                      % Beta 1.
sigma = 2;                                        % Standard deviation.
```

```
sim = 1;                                            % Number of simulations.
results_mat = zeros(sim,3);                         % Results matrix.

Maximum Likelihood Estimation.
Draw the error terms, , from the normal distribution and generate the data, . Estimate the mode
for s = 1:sim
    u = normrnd(0,sigma,T,1);                       % Error terms.
    y = zeros(T,1);                                 % Generate y.
    y(1) = FILL IN;                        % Omitted code.

    for j = 2:T
        y(j) = beta0 + beta1*y(j-1) + u(j);
    end

    b0 = [0; 0.5; 0.1];

    options = optimset('Display','off');
    [b_mle,~,~,~,~,hess] = fminunc(@logl_AR,b0,options);

    results_mat(s,:) = b_mle';
    v_mle = inv(hess);
end

Display the results of the last simulation.
The estimate and standard error (in parenthesis) of  in the last simulation is:
fprintf(' %0.4f\n(%0.4f)\n',b_mle(1),sqrt(v_mle(1,1)))
The estimate and standard error (in parenthesis) of  in the last simulation is:
fprintf(' %0.4f\n(%0.4f)\n',b_mle(2),sqrt(v_mle(2,2)))
The estimate and standard error (in parenthesis) of  in the last simulation is:
fprintf(' %0.4f\n(%0.4f)\n',b_mle(3),sqrt(v_mle(3,3)))

Log likelihood function for AR(1).
function L = logl_AR(b)

global y T

b1 = b(1);                                          % Intercept.
b2 = b(2);                                          % AR coefficient.
s  = b(3);                                          % Standard deviation.
```

```
L = FILL IN HERE

for t = 2:T
    L = L - 0.5*log(s^2) - 0.5*(((y(t)-b1-b2*y(t-1))^2/s^2));
end

L = L - 0.5*T*log(2*pi);
L = -L;
```

2. (30%) a) Explain the working of the Heckman selection correction in your own words. Repeating formulas from the handout is not what I am asking for.

b) Fill in the missing code in the Matlab code below.

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

for s = 1:sim
    vcov = [sigmau^2 rho*sigmau; rho*sigmau 1];
    e = mvnrnd([0; 0],vcov,N);                    % Correlated error terms.

    u = e(:,1);                                   % Error terms for y0.
    v = e(:,2);                                   % Error terms for z0.

    y0 = beta0 + beta1*x + u;                     % Latent y0.
    z0 = gamma0 + gamma1*w + v;                   % Latent z0.

    y = y0.*double((z0 > 0));                     % Observed y.
    y((z0 <= 0)) = nan;                           % Truncation based on z0.

    z = double((z0 > 0));                         % Observed z.

    options = optimoptions(@fminunc,'Display','off');

    % Step 1: Probit.

    g0 = [0 0];
    [g1, ~, ~, ~, ~, hess] = fminunc(@logl_prob, g0, options);

    % Step 2: OLS.
```

```
    x1 = x(~isnan(y));                          % X corresponding to non-missing y.
    w1 = w(~isnan(y));                          % W corresponding to non-missing y.
    y1 = y(~isnan(y));                          % Non-missing y.

    cons = ones(size(y1,1),1);

   nom = normpdf(g1(1)+ g1(2)*w1);
   denom = FILL IN HERE;                      % Omitted code.
    term = FILL IN HERE ;                       % Omitted code.

    BigX = [cons x1 term];
    b_ols = inv(BigX'*BigX)*BigX'*y1;

    s2 = (y1-BigX*b_ols)'*(y1-BigX*b_ols)./(N-3);
    vmat = inv(BigX'*BigX)*s2;

    results_mat(s,:) = b_ols';
end

Display the results of the last simulation.
The estimate and standard error (in parenthesis) of  in the last simulation is:
fprintf(' %0.4f\n(%0.4f)\n',b_ols(1),sqrt(vmat(1,1)))
The estimate and standard error (in parenthesis) of  in the last simulation is:
fprintf(' %0.4f\n(%0.4f)\n',b_ols(2),sqrt(vmat(2,2)))
The estimate and standard error (in parenthesis) of  in the last simulation is:
fprintf(' %0.4f\n(%0.4f)\n',b_ols(3),sqrt(vmat(3,3)))

Log likelihood function for probit.
function L = logl_prob(g)

global w z

g0 = g(1);
g1 = g(2);

WG = g0 + g1*w;

f = normcdf(WG);

L = log(f);
```

```
L(z==0) = log(1-f(z==0));
L = -sum(L);

end
```

3. (20%) a) What happens in the Matlab code below (the "my_fun" function)? What is it calculating and what is it used for?

b) Explain the theory behind this code. You can use words, but you have to be very explicit, or you can use symbols.

```
 What does the unknown function 'my_fun' do?

Set the parameters.
There is 1 simulations with 100 observations. Set
clear
clc

global x t

T = 100;                                          % Number of observations.
beta0 = 0.5;                                      % Beta 0.
beta1 = 3;                                        % Beta 1.

sim = 50;                                         % Number of simulations.
results_mat = zeros(sim,2);                       % Results matrix.

Maximum Likelihood Estimation.
In each simulation, generate the data, , then draw a corresponding  for each. Estimate the mode
x = ((1:T)'./T).*normrnd(0,1,T,1);

for s = 1:sim
    theta = exp(beta0 + beta1*x);
    t = exprnd(1./theta,T,1);                      % Draw t.

    b0 = [0.1 0.5];

    options = optimset('Display','off');
    [b_mle, ~, ~, ~, ~, hess] = fminunc(@logl_exp, b0, options);

    results_mat(s,1:size(results_mat,2)) = b_mle';
```

```
    vmat = inv(hess);
    my_out = my_fun(b_mle);
end

Display the results of the last simulation.
The estimate and standard error (in parenthesis) of  in the last simulation is:
fprintf(' %0.4f\n(%0.4f)\n',b_mle(1),sqrt(vmat(1,1)))
The estimate and standard error (in parenthesis) of  in the last simulation is:
fprintf(' %0.4f\n(%0.4f)\n',b_mle(2),sqrt(vmat(2,2)))

Log likelihood function.
function [L] = logl_exp(b)

global x t

b1 = b(1);                                          % Beta 0.
b2 = b(2);                                          % Beta 1.

XB = b1+b2*x;

L = XB - t.*exp(XB);
L = sum(L);
L = -L;

end
Unknown function.
function [out] = my_fun(in)

global x t

b1 = in(1);                                         % Beta 0.
b2 = in(2);                                         % Beta 1.

XB = b1+b2*x;

out1 = 1 - t.*exp(XB);
out2 = x - x.*t.*exp(XB);

out = [out1 out2];
out = inv(out'*out);
```

```
end
```

4. (10%) Explain (use symbols) what is the survival function and the hazard function in a duration model.