ECONOMETRICS II, Fall, 2022
Bent E. Sørensen

Econometrics II. Midterm Exam 2—October 24, 2022

Each sub-question in the following carries equal weight.

1. (20%) Consider the bivariate 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 in their respective equation and satisfies the standard conditions (no autocorrelation/heteroskedasticity). We normalize the variance of v to unity and assume that u and v are uncorrelated, but the correlation of x with v is ρ (different from 0). 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) Is the OLS regression for y biased or unbiased? Explain why. (You can use words only, but the logic has to be clear.)
b) If x follows a standard normal distribution, suggest a correction factor that makes the y-regression unbiased. You do not need to derive the formula, but you have to be clear why your suggestion would work. (Note: this is not exactly the model we covered in detail in class, so you need to think a little.)

2. (20%) Consider the exponential duration model with constant hazard.
a) Write down the log likelihood function for a sample of N completed spells. Define all terms carefully.
b) Write down the log likelihood function for a sample with N_1 completed spells and N_2 incomplete spells.

3. a) Explain how a parametric bootstrap estimator works.
b) Explain how the non-parametric bootstrap estimator works.
c) Explain how the non-parametric block bootstrap estimator works. (You can assume the panel setting that we mainly focused on in class).

4. (20%) Consider the Matlab program below and explain what is being estimated where there is

a query in the code (2 places). (We need the formulas for the relations being estimated and what is the estimator.)

```
clear
clc

p = 1;                                                                    % pth difference.
k = 1;                                                                    % kth lag.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Data preparation.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Load data.

load data

pops = [pops6396(1:8,:); pops6396(10:51,:)];                              % Population.
pops = pops(:,1:33);

cpi_vec = cpi6396(1:33);                                                  % CPI.
cpi     = kron(ones(size(pops,1),1),cpi_vec');

dpinc = [dpi6396(1:8,:); dpi6396(10:51,:)];                               % Income.
dpinc = dpinc(:,1:33);

dpi_agg = sum(dpinc)';                                                    % Aggregate.
dpinc   = dpinc./cpi;
dpi_agg = dpi_agg./cpi_vec;

ndur6395 = ndur6095(:,4:36);                                              % Non-durables.
ndur     = [ndur6395(1:8,:); ndur6395(10:51,:)];

sale = ndur./cpi;
perc = sum(ndur)';
```

```
perc = perc./cpi_vec;                                                      % Non-durable consu

clear ndur cpi cpi_vec pops6396 cpi6396 dpi6396 ndur6095 ndur6395

% Make everything per capita.

dpi     = dpinc./pops;
sale    = sale./pops;
perc    = perc./sum(pops)';
dpi_agg = dpi_agg./sum(pops)';
dpi_agg = kron(ones(size(pops,1),1),dpi_agg');
perc    = kron(ones(size(pops,1),1),perc');

clear pops

% Take logs.

logdpi     = log(dpi);
logsale    = log(sale);
logperc    = log(perc);
logdpi_agg = log(dpi_agg);

clear dpi sale perc dpi_agg

% Take differences.

D = size(logdpi,2);

d_dpi     = idiff(logdpi,p,D);
d_sale    = idiff(logsale,p,D);
d_perc    = idiff(logperc,p,D);
d_dpi_agg  = idiff(logdpi_agg,p,D);

clear D p logdpi logsale logperc logdpi_agg

dpi_t = d_dpi';
dpi_1ag = lagmatrix(d_dpi',1);                                             % Lag matrix.

n1 = size(dpi_t,1);
n2 = size(dpi_1ag,1);
```

```
dpi_t = dpi_t(k+1:n1,:)';
dpi_lag = dpi_lag(k+1:n2,:)';

clear n1 n2 d_dpi

sale_t = d_sale' ;

n3 = size(sale_t,1);

sale_t = sale_t(k+1:n3,:)';

clear n3 d_sale

dpi_aggt = d_dpi_agg';
dpi_agglag = lagmatrix(d_dpi_agg',1);                                          % Lag matrix.

n4 = size(dpi_aggt,1);
n5 = size(dpi_agglag,1);

dpi_aggt = dpi_aggt(k+1:n4,:)';
dpi_agglag = dpi_agglag(k+1:n5,:)';

clear n4 n5 d_dpi_agg

perc_t = d_perc';

n6 = size(perc_t,1);

perc_t = perc_t(k+1:n6,:)';

clear n6 d_perc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Fixed effects.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Time fixed effects.
```

```
[dpi_t_ft,dpi_1_ft] = fe(dpi_t,dpi_1ag,1,0);
[dpi_atft,dpi_a1ft] = fe(dpi_t-dpi_aggt,dpi_1ag-dpi_agg1ag,1,0);
[saleatft,~]        = fe(sale_t-perc_t,perc_t,1,0);
[sale_tft,perc_tft] = fe(sale_t,perc_t,1,0);

% Cross section fixed effects.

[dpi_t_fx,dpi_1_fx] = fe(dpi_t,dpi_1ag,0,1);
[dpi_atfx,dpi_a1fx] = fe(dpi_t-dpi_aggt,dpi_1ag-dpi_agg1ag,0,1);
[sale_tfx,~]        = fe(sale_t,perc_t,0,1);
[saleatfx,perc_tfx] = fe(sale_t-perc_t,perc_t,0,1);

% Cross section and time fixed effects.

[dpi_t_fxt,dpi_1_fxt] = fe(dpi_t_fx,dpi_1_fx,1,0);                          % Cross fixed effe
[sale_tfxt,~]         = fe(sale_tfx,perc_tfx,1,0);
[saleatfxt,perc_tfxt] = fe(saleatfx,perc_tfx,1,0);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Estimation and Results.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N = size(sale_t,1);
T = size(sale_t,2);
O = ones(N,T);



[gls2,glsstdev2] = xtreg(sale_tft,dpi_t_ft);                                A: What is being es

[gls6, glsstdev6] = xtreg(saleatfx,dpi_atfx);                               B What is being est
```

5. (20%) Consider the Matlab program below and explain what is being estimated where there is a query in the code (2 places). (We need the formulas for the relations being estimated and what is the estimator.) (2 places)

```
clear
clc

% Set the true parameters and placeholders for results.

T = 150;                                                                          % Number of observa

beta0 = 0;                                                                         % Intercept in equa
beta1 = 1.5;                                                                       % Coefficient on Y
beta2 = 0.3;                                                                       % Coefficient on X
beta3 = 0.0;                                                                       % Intercept in equa
beta4 = 0.3;                                                                       % Coefficient on x
beta5 = 2;                                                                        % Coefficient on x2
beta6 = 0.015;                                                                      % Coefficient on
beta7 = 0.0;                                                                       % Coefficient on x
beta8 = 0.0;
beta9 = 0.0;
beta10 = 0.0;



sigma1 = 1;                                                                        % Standard deviati
sigma2 = 1;                                                                        % Standard deviati

sim = 200;                                                                        % Number of simulat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% The simultaneous equations model is
%           Y1 = beta0 + beta1*Y2 + beta2*x1 + u1
%           Y2 = beta3 + beta4*x1 + beta5*x2 + beta6*x3 + beta7*x4 + beta8*x5 + beta9*x6 + bet
%
% This code estimates the coefficients in the first equation.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Generate the data.
x1 = 2 + normrnd(0,1,T,1);                                   % x1.
```

```
x2 = 3 + sigma1.*normrnd(0,1,T,1);                                % x2.
x3 = -2 + 0.4*x2 + sigma1.*normrnd(0,1,T,1);
x4 = normrnd(0,1,T,1);
x5 =  0.9*sigma2.*normrnd(0,1,T,1);
x6 = 1 + sigma1.*normrnd(0,1,T,1);
x7 =  0.3*sigma1.*normrnd(0,1,T,1);
x8 =  0.9*sigma2.*normrnd(0,1,T,1);
x9 = 1 + sigma1.*normrnd(0,1,T,1);
x10 =  0.3*sigma1.*normrnd(0,1,T,1);
x11 =  0.9*sigma2.*normrnd(0,1,T,1);
x12 = 1 + sigma1.*normrnd(0,1,T,1);
x13 =  0.3*sigma1.*normrnd(0,1,T,1);

X = [ones(T,1) x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13];




for s = 1:sim

    u1 = normrnd(0,sigma1,T,1);                                          % Residuls for equa
    u2 = normrnd(0,sigma2,T,1) + 5*u1;                                   % Residuals for equ

    %Generate y2
    y2 = beta3 + beta4*x1 + beta5*x2 + beta6*x3 + beta7*x4 + beta8*x5 + beta9*x6 + beta10*x7 +

    y1 = beta0 + beta1*y2 + beta2*x1 + u1;

    Y = [y1 y2];
    Y1 = Y(:,1);   %same as y1
    Y2 = Y(:,2);   %endogenous regressors, same as generated y2


    X_exo1 = X(:,1:2); %exogenous regressors in the reduced form
    X_OLS=[ones(T,1) Y2 x1];

    % xxxx
    B2_hat = inv(X'*X)*X'*Y2;                                            % First stage esti
    Y2_hat = X*B2_hat;                                                   % Fitted values of
```

```
    % xxxx
    X1_hat = [ones(T,1) Y2_hat x1];
    B_zzzz(s,:) = inv(X1_hat'*X1_hat)*X1_hat'*Y1;       A: What estimator/model is this?

    %xxxxx
    N = length(Y2);
    Mexo = eye(N) - X*inv(X'*X)*X';                                     %projection matrix, X is
    Mexo1 = eye(N) - X_exo1*inv(X_exo1'*X_exo1)*X_exo1';
    W = [Y1 Y2]'*(Mexo)*[Y1 Y2];                                        %2x2 matrix
    W1 = [Y1 Y2]'*(Mexo1)*[Y1 Y2];   %2x2 matrix
    lambda = min( eig(inv(W)*W1 )) ;
      %finds the min kappa from the book
    B_xxx(s,:) = inv(X_OLS'*(eye(N)-(lambda*Mexo))*X_OLS)
                  *(X_OLS'*(eye(N)-(lambda*Mexo))*Y1); B: What estimator/model  is this?


end
```