Final Exam—December 2, 2020

Each sub-question in the following carries equal weight.

1. (15%) a) Explain how the simple bootstrap estimator works.

b) Explain what a parametric bootstrap is.

2. (15%) Assume that you have a sample of outcomes y_i, which can take values 0, 1, and 2, where the outcomes of y indicate an ordering. You have a set of regressors x_i, that may help explain the outcomes.

a) What statistical model can you use to estimate such outcomes (give one example)?

b) Write down the log likelihood function for the model.

3. (30%) Assume you estimate a vector of parameters $\hat{\beta}$ with estimated variance $v\hat{a}r(\hat{\beta}) = \hat{\Sigma}$.

a) Write down the Wald test for $\beta = \beta_0$. (Assume these are vectors, not scalars.)

b) Write down the Wald test for $(\beta_1^2, \log \beta_2) = (0, 0)$.

c) Assuming $\hat{\beta}$ is asymptotically normally distributed, what is the asymptotic distribution of these tests under typical conditions?

d) What does the Matlab code below estimate?

e) In the Matlab code below, what is the Wald test for the hypothesis $(beta0, beta1) = (0.5, 0.5)$? (You need to use the notation from the code, so I can see that you know.)

Final Exam Code 1.

This code estimates some model. What does this code do?

Set the parameters.

There is 1 simulation with 300 observations. Set $\beta_0 = 0.5, \beta_1 = 3$ and $\sigma = 1$.

```
close all
clear
clc
global x z N

N = 300;
beta0 = 0.5;
beta1 = 3;
sigma = 1;

sim = 1;
results_mat = zeros(sim, 2);
```

Maximum Likelihood Estimation.

In each simulation, draw the error terms, U, from the standard normal distribution and generate the data, X, Y and Z. Estimate the model using maximum likelihood and record the estimates.

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

for s = 1:sim

    % Generate Y and Z.

    u = normrnd(0,sigma,N,1);

    y = beta0*ones(size(x,1),1) + beta1*x + u;
    z = double((y > 0));
```

```
    % Estimation using ML.

    b0 = [1 1];

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

    % Store estimates.

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

end

% Computes something.

den = normcdf(b_mle(1)+b_mle(2)*x);
den = den.*(1-normcdf(b_mle(1)+b_mle(2)*x));

quad_num = ((z-normcdf(b_mle(1)+b_mle(2)*x)) ...
                    .*normpdf(b_mle(1)+b_mle(2)*x));

quad = (quad_num./den).^2;

npdf_diff_0 = (1/sqrt(2*pi)).*...
                exp((-(z-b_mle(1)-b_mle(2).*x).^2)./2).*...
                         (z-b_mle(1)-b_mle(2).*x);

lin_num_0 = ((z-normcdf(b_mle(1)+b_mle(2)*x)).*npdf_diff_0);

npdf_diff_1 = (1/sqrt(2*pi)).*...
                exp((-(z-b_mle(1)-b_mle(2).*x).^2)./2).*...
                        (z-b_mle(1)-b_mle(2).*x).*x;

H_lin_num_1 = ((z-normcdf(b_mle(1)+b_mle(2)*x)).*npdf_diff_1);

v00 = sum(-quad + (lin_num_0./den));
v01 = sum((-quad + (H_lin_num_1./den)).*x);
```

```
v10 = sum((-quad + (lin_num_0./den)).*x);
v11 = sum((-quad + (H_lin_num_1./den)).*x.^2);

v = [v00 v01; v10 v11];

% Some matrix.

vmat = -inv(v);

% Compute the Wald Statistic.

% Missing code.
```

Functions.

```
function L = logl_prob(b)

% The following constructs the loglikelihood function.

global x z

b0 = b(1);
b1 = b(2);

XB = b0 + b1*x;

f = normcdf(XB);

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

end
```

4. (20%)
The Matlab code below estimates an AR(1) model but with a term left out. Assume the model is stationary.

a) What should the missing term be for this program to estimate the model by OLS?

b) What should the missing term be for this program to estimate the model by Maximum Likelihood?

Final Exam Code 2.

This code estimates, using maximum likelihood, the AR(1) process

$$y_t = \beta_0 + \beta_1 y_{t-1} + u_t \ ,$$

with $u_t \sim NID(0, \sigma^2)$. Code has been omitted where indicated.

Set the parameters.

There is 1 simulation with 300 observations. Set $\beta_0 = 0, \beta_1 = 0.5$ and $\sigma = 2$.

```
clc
clear

global y T

T = 300;
beta0 = 0;
beta1 = 0.5;
sigma = 2;

sim = 1;
results_mat = zeros(sim,3);
```

Maximum Likelihood Estimation.

Draw the error terms, U, from the normal distribution and generate the data, Y. Estimate the model using maximum likelihood and record the estimates.

```
for s = 1:sim
    u = normrnd(0,sigma,T,1);
    y = zeros(T,1);
    y(1) = %-------------% ;                  % 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
```

Log likelihood function for AR(1).

```
function L = logl_AR(b)

global y T

b1 = b(1);
b2 = b(2);
s  = b(3);

L = %-------------% ;                         % Omitted code.

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;

end
```

5. (20%) What does the following Matlab code do? Be explicit. You will get partial points even if you do not get it all, so make sure to name what you see.

Final Exam Code 3.

Describe what this code does.

Set the parameters.

There are 300 individuals and 100 time periods per simulation. Set $\beta_1 = 0.92$, and $\sigma = 1$.

```
clear
clc

global dxvec dyvec Z

T   = 100;
N   = 300;
sim = 50;

sigma = 1;
rho   = 0.92;
mu    = (1:N)'./N;
```

Generate the data.

Panel data where each row is an individual and each column is time.

```
for s = 1:sim

    e = normrnd(0,sigma,N,T);

    y = zeros(N,T);
    y(:,1) = mu./(1-rho) + e(:,1)./sqrt(1-rho^2);

    for i = 2:T
        y(:,i) = mu + rho*y(:,i-1) + e(:,i);
    end

    dy = y(:,2:end)-y(:,1:end-1);

    dx = dy(:,1:end-1);
    dy = dy(:,2:end);

end

    Z = reshape(y(:,1:end-2),[],1);
    num_inst = size(Z,2);

    dyvec = reshape(dy,[],1);
    dxvec = reshape(dx,[],1);
```

GMM with Identity Weighting Matrix.

```
    b0 = 0;
    W  = eye(num_inst);

    opt = optimset('FinDiffType','central','HessUpdate','BFGS');
    bgmm  = fminunc(@gmm_obj,b0,opt,W);
```

```
Local minimum found.
```

```
Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>
```

Functions.

```
function crit = gmm_obj(guess,W)

% GMM objective function.

global dxvec dyvec Z

rho = guess;
Zu = Z.*(dyvec-rho.*dxvec);

mom = mean(Zu,1)';
crit = mom'*W*mom;

end
```