Problem Set 1 Solutions

Tyler Ransom

July 25, 2012

1 Main Problem Set Code

```
clear all;
clc;
load nlsy97
options = optimset('Disp','iter-detailed','MaxFunEvals',1e12,'MaxIter',1e6);
rand('seed',1234); randn('seed',1234);
%% Problem 1(a) -- Reshape the data into long format
N = length(ID);
T = 5;
activityt = activity';
log_waget = log_wage';
hgct = hgc';
expert = exper';
Diplomat = Diploma';
AAt = AA';
BAt = BA';
malet = repmat(male,1,T)';
AFQTt = repmat(AFQT,1,T)';
Mhgct = repmat(Mhgc,1,T)';
IDt = repmat(ID,1,T)';
IDl = IDt(:);
IDprime  = [1:numel(ID)]';
IDprimet = repmat(IDprime,1,T)';
IDprimel = IDprimet(:);
X = [ones(N*T,1) malet(:) AFQTt(:) Mhgct(:) hgct(:) expert(:) Diplomat(:) ...
    AAt(:) BAt(:)];
y = activityt(:);
K = size(X,2);
%% Problem 1(b) -- estimate mlogit using fminunc
% bstart = mnrfit(X(:,2:end),y); % I cheated here and used the mnrfit ...
    starting values (for computational speed)
bstart = rand(2*size(X,2),1);
[bunc,lunc,~,~,gunc,hunc] = fminunc('mlogit',bstart(:),options,X,y);
SEunc = sqrt(diag(inv(hunc)));
%% Problem 1(c) -- estimate mlogit using mnrfit
[bfit,~,stats] = mnrfit(X(:,2:end),y);
SEfit = stats.se;
```

```
mlogit_compare   = [[bunc(1:size(X,2)) bunc(1+size(X,2):end)] bfit]
mlogit_compareSE = [[SEunc(1:size(X,2)) SEunc(1+size(X,2):end)] stats.se]
%% Problem 2(a) -- pooled OLS with fewer covariates
X1 = [ones(N*T,1) malet(:) hgct(:) expert(:) Diplomat(:) AAt(:) BAt(:)];
y1 = log_waget(:);
K1 = size(X1,2);
subset = ¬isnan(log_waget(:));
N1 = sum(subset);
N0 = sum(sum(¬isnan(log_wage),2)>0);
X1 = X1(subset,:);
y1 = y1(subset);
bpoola = (X1'*X1)\X1'*y1;
sigmapoola = (1/(N1-K1))*(y1-X1*bpoola)'*(y1-X1*bpoola);
sepoola = sqrt(diag(sigmapoola*(eye(K1))/(X1'*X1)));
%% Problem 2(b) -- pooled OLS with full covariate set
X = X(subset,:);
bpoolb = (X'*X)\X'*y1;
sigmapoolb = (1/(N1-K))*(y1-X*bpoolb)'*(y1-X*bpoolb);
sepoolb = sqrt(diag(sigmapoolb*(eye(K))/(X'*X)));
OLScompare = [[bpoola(1:2); 0;0; bpoola(3:end)] bpoolb]
OLScompareSE = [[sepoola(1:2); 0;0; sepoola(3:end)] sepoolb]
SSEpool = (y1-X*bpoolb)'*(y1-X*bpoolb);
%% Problem 2(c) -- Fixed effects ("wide way")
% drop all NaNs from analysis
int = ones(N,T);
nan_locations = find(isnan(log_wage));
int(nan_locations) = NaN;
hgc(nan_locations) = NaN;
exper(nan_locations) = NaN;
Diploma(nan_locations) = NaN;
AA(nan_locations) = NaN;
BA(nan_locations) = NaN;

for t=1:T
    % Create the "double-dot" matrices for each time period
    Xdd(:,:,t)      = [hgc(:,t)-nanmean(hgc,2) exper(:,t)-nanmean(exper,2) ...
       Diploma(:,t)-nanmean(Diploma,2) AA(:,t)-nanmean(AA,2) ...
       BA(:,t)-nanmean(BA,2)];
    ydd(:,:,t)      = log_wage(:,t)-nanmean(log_wage,2);
    % create X'*X and X'*y for each time period
    Xdduse(:,:,t)   = ...
       Xdd(¬isnan(log_wage(:,t)),:,t)'*Xdd(¬isnan(log_wage(:,t)),:,t);
    Xddpost(:,:,t)  = ...
       Xdd(¬isnan(log_wage(:,t)),:,t)'*ydd(¬isnan(log_wage(:,t)),:,t);
end
%OLS estimation and standard errors:
bfe      = sum(Xdduse,3)\sum(Xddpost,3);
sefe     = ...
   sqrt(diag((1/(N1-N0-size(Xdd(:,:,t),2)))*SSEpool*(eye(size(Xdd(:,:,t),2)))/(sum(Xdduse,
FEresult = [bfe sefe]
%% Problem 2(c) -- Fixed effects ("long way") [This took roughly 2.5 hours ...
   on the cluster, so I wouldn't recommend doing it this way]
% create vector of panel lengths for each person and drop people with zero ...
   panel length
```

```
Ti = sum(¬isnan(log_wage),2);
Ti_temp = repmat(Ti,1,5)';
Ti_test = Ti_temp(:);
IDtest = ID(Ti≥1);
Ti(Ti<1)=[];
% create subsetting vector to test code with
IDcut = 1e4;
% Set up data
ytest = log_waget(:);
Xtest = [hgct(:) expert(:) Diplomat(:) AAt(:) BAt(:)];
Til= Ti_test(¬isnan(ytest));
subsetl = ¬isnan(ytest(IDl<IDcut));
yl = ytest(subsetl);
Xl = Xtest(subsetl,:);
Kl = size(Xl,2);
% create de-meaning "Q" matrix for an unbalanced panel
Q = [];
tic;
for i=1:numel(Ti(IDtest<IDcut))
    Qi = eye(Ti(i))-ones(Ti(i))/Ti(i);
    Q = blkdiag(Q,Qi);
end
disp('Time spent forming Q matrix')
toc;
bfelong  = (Xl'*Q*Xl)\(Xl'*Q*yl);
sefelong = sqrt(diag(1/(N1-N0-Kl)*SSEpool*eye(Kl)/(Xl'*Q*Xl)));
FEresultl = [bfelong sefelong]
%% Problem 2(d) -- Random effects
Ti = sum(¬isnan(log_wage),2);
% Build variance matrices
u = (y1-X*bpoolb);
var_alpha_mat = zeros(N,T-1,T-1);
for i=1:N
    for t=1:T-1
        for s=t+1:T
            var_alpha_mat(i,t,s) = y1(i)-X(i,:)*bpoolb;
        end
    end
end
T_bar = mean(Ti(Ti>0));
var_alpha = sum(sum(sum(var_alpha_mat)))/(N0*T_bar*(T_bar-1)/2-size(X,2))
var_epsilon = sigmapoolb-var_alpha
omega1 = var_epsilon*eye(1)+var_alpha*ones(1);
omega2 = var_epsilon*eye(2)+var_alpha*ones(2);
omega3 = var_epsilon*eye(3)+var_alpha*ones(3);
omega4 = var_epsilon*eye(4)+var_alpha*ones(4);
omega5 = var_epsilon*eye(5)+var_alpha*ones(5);

% Build data matrices
Xre = [ones(N*T,1) malet(:) AFQTt(:) Mhgct(:) hgct(:) expert(:) ...
    Diplomat(:) AAt(:) BAt(:)];
% Xre = [hgct(:) expert(:) Diplomat(:) BAt(:)];
Kre = size(Xre,2);
yre = log_waget(:);
```

```
135
136 tic;
137 % Initialize matrices that will be created below in for-loop
138 X1re = zeros(Kre,Kre,max(IDprime));
139 X2re = zeros(Kre, 1 ,max(IDprime));
140 SEout = zeros(Kre,Kre,max(IDprime));
141 SEin  = zeros(Kre,Kre,max(IDprime));
142 for i=1:max(IDprime)
143     subsetRE = (IDprimel==i)&¬isnan(yre);
144     if Ti(i)==1
145         X1re(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega1)*Xre(subsetRE,:);
146         X2re(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega1)*yre(subsetRE);
147     elseif Ti(i)==2
148         X1re(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega2)*Xre(subsetRE,:);
149         X2re(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega2)*yre(subsetRE);
150     elseif Ti(i)==3
151         X1re(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega3)*Xre(subsetRE,:);
152         X2re(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega3)*yre(subsetRE);
153     elseif Ti(i)==4
154         X1re(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega4)*Xre(subsetRE,:);
155         X2re(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega4)*yre(subsetRE);
156     elseif Ti(i)==5
157         X1re(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega5)*Xre(subsetRE,:);
158         X2re(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega5)*yre(subsetRE);
159     end
160 end
161 bre = sum(X1re,3)\sum(X2re,3);
162 ure = yre-Xre*bre;
163 for i=1:max(IDprime)
164     subsetRE = (IDprimel==i)&¬isnan(yre);
165     if Ti(i)==1
166         SEout(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega1)*Xre(subsetRE,:);
167         SEin (:,:,i) = ...
              (ctranspose(Xre(subsetRE,:))/omega1)*(ure(subsetRE)*ctranspose(ure(subsetRE)))/
168     elseif Ti(i)==2
169         SEout(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega2)*Xre(subsetRE,:);
170         SEin (:,:,i) = ...
              (ctranspose(Xre(subsetRE,:))/omega2)*(ure(subsetRE)*ctranspose(ure(subsetRE)))/
171     elseif Ti(i)==3
172         SEout(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega3)*Xre(subsetRE,:);
173         SEin (:,:,i) = ...
              (ctranspose(Xre(subsetRE,:))/omega3)*(ure(subsetRE)*ctranspose(ure(subsetRE)))/
174     elseif Ti(i)==4
175         SEout(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega4)*Xre(subsetRE,:);
176         SEin (:,:,i) = ...
              (ctranspose(Xre(subsetRE,:))/omega4)*(ure(subsetRE)*ctranspose(ure(subsetRE)))/
177     elseif Ti(i)==5
178         SEout(:,:,i) = (ctranspose(Xre(subsetRE,:))/omega5)*Xre(subsetRE,:);
179         SEin (:,:,i) = ...
              (ctranspose(Xre(subsetRE,:))/omega5)*(ure(subsetRE)*ctranspose(ure(subsetRE)))/
180     end
181 end
182 sere = sqrt(diag(sum(SEout,3)\sum(SEin,3)/sum(SEout,3)));
183 disp('Time spent computing random effects')
```

```
184 toc;
185 REresult = [bre sere]
```

2 MLE Function Code

```
function like = mlogit(b,X,y)
b1 = b(1:size(X,2));
b2 = b(1+size(X,2):end);
dem = 1+exp(X*b1)+exp(X*b2);
like = -sum((y==1).*(X*b1) + (y==2).*(X*b2) + (y==3) - log(dem));
```