```
%-------------------------------------------------------------------------------
% Preliminaries
%-------------------------------------------------------------------------------
clear all
close all

% Specify globals data and V to be used in the function call of SML
global data V

% Set the seed for the random draws u_im. Grab V
rng(1234)
V = randn(2000,1000);

% Load data
load('logit.mat');
[N,~] = size(data);

%-------------------------------------------------------------------------------
% Unconstrained maximization using fminco
%-------------------------------------------------------------------------------
% Specify starting guess of optimal mu and ln_sigma
x0 = [1,1];
options = optimoptions('fminunc','Algorithm','trust-region', ...
                       'SpecifyObjectiveGradient',true);
fun = @SML;
[x_min, ~, ~, ~,~,x_hessian] = fminunc(fun,x0,options)

mu_val_est = x_min(1)
ln_sig_est = x_min(2)

% Problem 6e, standard errors under correct specification
G = [1,0;0,(1/x_min(2))];
correct_specification_se = sqrt(diag((1/N)*G*inv(x_hessian)*G))

% Problem 6f, standard errors under incorrect specification
% Construct vector of scores
[sim_avg_cdf, sim_avg_pdf, sim_avg_pdf_u_im,X,Y]...
    = sim_avg_vals(mu_val_est,ln_sig_est);

score_v = [(Y.*X.*sim_avg_pdf)./(sim_avg_cdf) -...
           ((1-Y).*X.*sim_avg_pdf)./(1-sim_avg_cdf),...
           (Y.*X.*sim_avg_pdf_u_im)./(sim_avg_cdf) -...
           ((1-Y).*X.*sim_avg_pdf_u_im)./(1-sim_avg_cdf)];

V_s = (1/length(score_v))*score_v'*score_v;

incorrect_spec_se = sqrt(diag((1/N)*G*inv(x_hessian)*V_s*inv(x_hessian)*G))
```