```
function [LL, W, s2] = ppca_nsp(X, D)
%
% This function was written by Jonathan Kao
%   for use in EE239AS.2, UCLA.
%
% Inputs:
%   - X:    N x K matrix, where N is the dimensionality of the data and K is
% the number of data points.
%   - D:    the latent dimensionality
%
% Outputs:
%   - LL:   the log-likelihoods over all iterations
%   - W:    the W parameter of PPCA
%   - s2:   the sigma^2 parameter of PPCA
%   - mu:   the mu parameter of PPCA

    % Check for convergence (tol) or set a max number of cycles (cyc)
    tol = 1e-8;
    cyc = 1e8;

    % Set the random seed so everyone's code executes the same thing.
    randn('state', 0);
    [N, K] = size(X);

    % Initialization ofparameters
    cX  = cov(X', 1);
    W   = randn(N,D);
    s2  = mean(diag(cX));       % noise variance
    mu  = mean(X, 2);
    LL  = [];

    LL_prev     = -Inf;

    % do a maximum of cyc cycles
    for i = 1:cyc
        % ==============================
        % E-STEP -- compute E[s], E[ss^T]
        % ==============================

        % Denote E[s] as:    Es
        % Denote E[ss^T] as: Ess

        %%%%% START YOUR CODE HERE TO CALCULATE Es and Ess %%%%%
        mu_mat = repmat(mu,1,K);
        % identity matrix needs to be N x N to match dimensions of W*W'

        Es = W'*inv(W*W' + s2*eye(N))*(X-mu_mat);
        Ess = eye(D) - W'*inv(W*W' + s2*eye(N))*W + (1/K)*Es*Es';
        % Ess = cov(s_k) + Es * Es'

        %%%%% END YOUR CODE HERE TO CALCULATE Es and Ess %%%%%
```

```
            % ======================
            % Compute log likelihood
            % ======================

            % We'll do this one for you.
            Sx      = W*W' + s2 * eye(N);
            Xm      = bsxfun(@minus, X, mu);
            LLi     = -N*K/2*log(2*pi) - K/2*log(det(Sx))-1/2*trace(Xm' * inv(Sx) * Xm
            LL      = [LL LLi];

            % ======================
            % M-STEP -- update W, s2
            % ======================

            %%%%% START YOUR CODE HERE TO CALCULATE W and s2 %%%%%

            W = (X-mu_mat)*Es'*(1/K)*inv(Ess');
            s2 = 1/(N*K) * trace((X-mu_mat)*(X-mu_mat)' - W*Es*(X-mu_mat)');

            %%%%% END YOUR CODE HERE TO CALCULATE W and s2 %%%%%

            % ====================
            % Check for convergence
            % ====================

            % First iteration, set the base likelihood that we'll measure
            % comparisons vs.
            if i == 1
                LLbase = LLi;
            end

            if LLi < LL_prev
                disp('LIKELIHOOD VIOLATION!');
            elseif ((LLi-LLbase) < (1+tol)*(LL_prev-LLbase))
                break;
            end

            LL_prev = LLi;

        end
    end

Error using ppca_nsp (line 23)
Not enough input arguments.
```

Published with MATLAB® R2014b