Exercise 1 - Efficient Matlab Programming

Giulio Valente

1 Task A: Matrix Standardization

Compute mean and std for each column. Remove mean from each element (center) and subtract by the std (standardize).

```
function [ Y ] = stdize( X )

% Fast solution
X_std = std(X);
X_mean = mean(X);
Y = (X - repmat(X_mean,size(X,1),1))./(repmat(X_std,size(X,1),1));

%% Slow solution
%Y = zeros(size(X,1),size(X,2));

%for i=1:size(X,1)
%    for j=1:size(X,2)
%        Y(i,j) = (X(i,j) - X_mean(j))/X_std(j);
%    end
%end

end
```

2 Task B: Pairwise Distances in the Plane

Given P and Q, compute the euclidian distance for each point.

```
function D = pairdist(P, Q)
%D = zeros(size(P,1),size(Q,1));

% Fast solution
[X1, X2] = meshgrid(P(:,1), Q(:,1));
[Y1, Y2] = meshgrid(P(:,2), Q(:,2));
D = sqrt((X1 - X2).^2 + (Y1 - Y2).^2)';

% Slow solution
%for i=1:size(P,1)
%    for j=1:size(Q,1)
%        D(i,j) = sqrt((P(i,1)-Q(j,1))^2 + (P(i,2)-Q(j,2))^2);
%    end
%end

end
```

3 Task C: Likelihood of a Data Sample

Given a matrix, for each column compute the likelihood for 1 and 2 dimension and for each column return the dimension where it is higher inside an array.

```
function a = lassign_reference_vec(X, mu0, Sigma0, mu1, Sigma1)
```

```
a = g(X, mu0, Sigma0) < g(X, mu1, Sigma1);
a = a + 1;
end

function p = g(X, mu, Sigma)

D = (X - repmat(mu,1,size(X,2)))';
p = -log(det(Sigma)) - sum((D/Sigma).*D,2)';

end
```