```matlab
% ### EXrandomWalk1D.m ###     11.15.14
% Calculates mean-squared distance for a group of independent 1-D
random
% walkers, each moving with user-specified probablity of one step left
vs right
% [Source (modified): Kevin Berwick (re 'Computational Physics',
Giordano & Nakanishi)]
clear;
% -------------
N= 200;       % Total # of (independent) walkers (each starts at x=0)
M= 100;       % Total # of steps for each walker
K= 3;         % # of walkers to show individual traces for [3]
bias= 0.5;     % number between [0,1] to indicate bias for left vs
right (0.5= equal prob.)
% -------------
% +++
step_number= zeros(1,M);        %
x2ave= zeros(1,M);              % allocate array to stored
(suquentially averaged) MSD
step_number_array= [1:1:M];     %
% +++
%
% NOTE: the loop is set up in such a way to average x2ave across
walkers
for r= 1:N
    x=0;     % initialize position for r'th walker
    position(r,1)= 0;
    % loop to go through M steps for r'th walker
    for nn=1:M;
        % conditional determines whether step is to the left or right
        if (rand<bias),  x=x+1;
        else    x=x-1;  end;
        x2ave(nn)=x2ave(nn)+x^2;    % store squared displacement
(handles averging across r)
        position(r,nn+1)= x;           % store displacment for each
walker and step
    end;
end;
x2ave= x2ave/N;      % Divide by number of walkers
% plot MSD
figure(1);
plot(step_number_array, x2ave, 'k'); hold on;
title(['MSD for 1-D random walk (average from ',num2str(N),'
walkers)']);
xlabel('Step number'); ylabel('Mean-Squared Distance (x^2)');
% plot a subset of individual traces
figure(2); clf; hold on; grid on;
for nn=1:K
    shade= 1-(nn-1)/K;
    plot(position(nn,:),'Color',[1 1 1]-shade);
```

```
end
xlabel('Step number'); ylabel('Position'); title('Representative
traces');
plot([0 M],[1 1]*sqrt(x2ave(end)),'g--','LineWidth',2) % include MSD
bounds at step M
plot([0 M],[-1 -1]*sqrt(x2ave(end)),'g--','LineWidth',2)
plot(M,sqrt(mean(position(:,end).^2)),'ro');    % reality check
(another way to compute final MSD)
disp(['Final mean (non-squared) distance =
',num2str(mean(position(:,end)))]);
```