

```

function [Vtd,Vmc] = eg_6_2_learn(nEpisodes,alpha)
% EG_6_2_LEARN - using TD(0) and constant step size monte carlo learn the value function
%               for the random walk example.
%
% Written by:
% --
% John L. Weatherwax          2007-12-07
%
% email: wax@alum.mit.edu
%
% Please send comments and especially bug reports to the
% above email address.
%
%-----

if( nargin < 2 ) alpha = 0.1; end;
gamma = 1; % the discount factor

% the number of non terminal states plus two terminal states:
nStates=5+2;

% initialize the nonterminal states ... an terminal states are defined to be ZERO
Vmc = 0.5*ones(1,nStates); Vmc(1)=0; Vmc(end)=0;
Vtd = 0.5*ones(1,nStates); Vtd(1)=0; Vtd(end)=0;

for ei=1:nEpisodes,
    % perform a random walk:
    %
    % --states are [1=X,2,3,4,5,6,7=X] == [X,A,B,C,D,E,X] and we start at C=4 (X is a
terminal code)
    st = 3+1; R=0; stateseen = [st];
    while( 1 < st && st < 7 )
        if( rand<0.5 )
            stp1=st+1;
        else
            stp1=st-1;
        end
        % assign a reward for this step:
        if( stp1~=7 ) rew = 0; else rew = +1; end
        %if( (stp1==1) || (stp1==7) ) % we are in a terminal state
        % break;
        %end
        R = R+rew;
        % update Vtd:
        Vtd(st) = Vtd(st) + alpha*( rew + gamma*Vtd(stp1) - Vtd(st) );
        stateseen = [stateseen; stp1];
        st = stp1;
    end

    % now have the total episode's return ... update Vmc:
    for si=1:length(stateseen),
        if( stateseen(si)==1 || stateseen(si)==7 ) continue; end % don't update terminal
states
        Vmc(stateseen(si)) = Vmc(stateseen(si)) + alpha*(R - Vmc(stateseen(si)));
    end
end

% remove terminal states:
Vmc = Vmc(2:end-1);

```

```
Vtd = Vtd(2:end-1);
```