

```

%
% Uses first visit monte carlo evaluation to compute the value function for
% the black jack example found in Figure~5.1
%
% Here we can be in a state that is specified by
%
%  $21-12+1$  = the number of possible values for the sum of the point totals for the cards I
have
% 13 = the number of possible values for the card that the dealer can have
% 2 = whether I have a usable ace or not
%
% Once I see the state, I store counts and averages in a 3-dimensional array.
%
% Written by:
% --
% John L. Weatherwax                2018-03-10
%
% email: wax@alum.mit.edu
%
% Please send comments and especially bug reports to the
% above email address.
%
%-----

%close all;
clc;

%N_HANDS_TO_PLAY=1e4;
%N_HANDS_TO_PLAY=2*1e4;
%N_HANDS_TO_PLAY=2*5e5;
N_HANDS_TO_PLAY=5e6;

%--
% implement hands of bj
%--

nStates      = prod([21-12+1,13,2]);
allStatesRewSum = zeros(nStates,1);
allStatesNVisits = zeros(nStates,1);

tic
for hi=1:N_HANDS_TO_PLAY,

    stateseen = [];
    deck = shufflecards();

    % the player gets the first two cards:
    p = deck(1:2); deck = deck(3:end);
    % the dealer gets the next two cards (and shows his first card):
    d = deck(1:2); deck = deck(3:end); dhv = handValue(d); cardShowing = d(1);

    % accumulate/store the first state seen:
    stateseen(1,:) = stateFromHand( p, cardShowing ); phv = handValue(p);

    % implement the policy of the player (hit until we have a hand value of 20 or 21):
    while( phv < 20 )
        p = [ p, deck(1) ]; deck = deck(2:end); % HIT
        stateseen(end+1,:) = stateFromHand( p, cardShowing ); phv = handValue(p);
    end
end

```

```

% implement the policy of the dealer (hit until we have a hand value of 17):
while( dhv < 17 )
    d = [ d, deck(1) ]; deck = deck(2:end); % HIT
    dhv = handValue(d);
end
% determine the reward for playing this game:
rew = determineReward(phv,dhv);

% accumulate these values used in computing global statistics:
for si=1:size(stateseen,1),
    if( (stateseen(si,1)>=12) && (stateseen(si,1)<=21) ) % we don't count "initial" and
terminal states
        %[stateseen(si,1)]
        %[stateseen(si,1)-12+1, stateseen(si,2), stateseen(si,3)+1]
        indx=sub2ind( [21-12+1,13,2], stateseen(si,1)-12+1, stateseen(si,2),
stateseen(si,3)+1 );
        allStatesRewSum(indx) = allStatesRewSum(indx)+rew;
        allStatesNVisits(indx) = allStatesNVisits(indx)+1;
    end
end
end % end number of hands loop
toc

mc_value_fn = allStatesRewSum./allStatesNVisits;

mc_value_fn = reshape( mc_value_fn, [21-12+1,13,2]);

% plot the various graphs:
%
figure; mesh( 1:13, 12:21, mc_value_fn(:,:,1) );
xlabel( 'dealer shows' ); ylabel( 'sum of cards in hand' ); axis xy; %view([67,5]);
title( 'no usable ace' ); drawnow;
%fn=sprintf('state_value_fn_nua_%d_mesh.eps',N_HANDS_TO_PLAY); saveas( gcf, fn, 'eps2' );
figure; mesh( 1:13, 12:21, mc_value_fn(:,:,2) );
xlabel( 'dealer shows' ); ylabel( 'sum of cards in hand' ); axis xy; %view([67,5]);
title( 'a usable ace' ); drawnow;
%fn=sprintf('state_value_fn_ua_%d_mesh.eps',N_HANDS_TO_PLAY); saveas( gcf, fn, 'eps2' );

figure; imagesc( 1:13, 12:21, mc_value_fn(:,:,1) ); caxis( [-1,+1] ); colorbar;
xlabel( 'dealer shows' ); ylabel( 'sum of cards in hand' ); axis xy;
title( 'no usable ace' ); drawnow;
%fn=sprintf('state_value_fn_nua_%d_img.eps',N_HANDS_TO_PLAY); saveas( gcf, fn, 'eps2' );
figure; imagesc( 1:13, 12:21, mc_value_fn(:,:,2) ); caxis( [-1,+1] ); colorbar;
xlabel( 'dealer shows' ); ylabel( 'sum of cards in hand' ); axis xy;
title( 'a usable ace' ); drawnow;
%fn=sprintf('state_value_fn_ua_%d_img.eps',N_HANDS_TO_PLAY); saveas( gcf, fn, 'eps2' );

```