
```

clc; clear;
%-----%
%-----Variable Change Langrange Strategy-----%
%-----%
% Inputs:
%     Starting Money: Money taking to the casino and willing to lose
%     Starting Sequence: Chosen starting sequence, input as matrix,
%                       e.g. [1,2,3,4]
%     Minimum Bet: Tabels minimum bet
% Outputs:
%     Percent chance of closing sequence (%)
%     Average Money walking away from
%-----%
%-----%

% Call for inputs
initial_starting = input("Enter Starting money: ");
sequence_start = input('Enter starting sequence: ');

%-----%
% This script calculates the odds in roulette from the langranian method %
%-----%
totals = zeros(100,1);
fail = 0;
win = 0;
runs = 10000;
average_wining = 0;
average_when_completed = 0;
%-----%
% Set up and inputs %
%-----%
numgames = 100000; % input
    number of bets
sequence = sequence_start; % Input starting langraining sequence
bet = 'black'; % either
    'black' or 'red'

% roulette numbers
Red = [1:2:9 12:2:18 19:2:27 30:2:36];
Black = [2:2:10 11:2:17 20:2:28 29:2:35];

for ii = 1:runs
% Radonmly generate the roulette results with a uniform distribution
% for simplicity, 37 = 0 and 38 = 00
result = randi(38,numgames, 1);
sequence_history = zeros(1,50);
sequence = sequence_start;
total_money = initial_starting;
% Loop through for each game
for i = 1:numgames
    result_red = 0;
    result_black = 0;

```

```

sequence_history(i,1) = total_money;
sequence_history(i,2:length(sequence)+1) = sequence;
%-----%
% Finding where result is black or red %
%-----%
% Checking if result is red
isRed = result(i) == Red;
for j = 1:length(isRed)
    if isRed(j) == 1
        result_red = 1;
    end
end
% Checking if result is black
isBlack = result(i) == Black;
for j = 1:length(isBlack)
    if isBlack(j) == 1
        result_black = 1;
    end
end
% Error checking
if result_black == 1 & result_red == 1
    fprintf('Error: both red and black hit')
end

%-----%
% now implement the lagrangian method %
%-----%
bet_amount = sequence(1) + sequence(end);
set bet amount using starting and final sequence
if bet_amount > total_money
    fail = fail+1;
    fprintf('Current bet is greater than total amount\n sequence: %s\n
current money: %.0f \n',num2str(sequence), total_money)
    average_winings(ii) = total_money;
    break
end
total_money = total_money - bet_amount ;
taking bet amount out of total
% Set if the round is won based off bet input (color)
if strcmp(bet,'black') == 1 & result_black == 1
    total_money = total_money + bet_amount *2; % if
hits, double money and add to total
if length(sequence) <= 2
    fprintf('Sequence is completed \nTotal money: %.0f\n',total_money)
    sequence_history(i+1,1) = total_money;
    win = win+1;
    average_winings(ii) = total_money;
    average_when_completed(ii) = total_money;
    break
else
    sequence = sequence(2:end-1);
take the 2 end numbers off it hits
end

```

```

    elseif strcmp(bet,'red') == 1 & result_black == 1           %
repeat process for red
    total_money = total_money + bet_amount *2;
    if length(sequence) <= 2
        sequence = sequence_start;
    else
        sequence = sequence(2:end-1);                           %
take the 2 end numbers off it hits
    end
    else                                                         % if
lose, add bet to end of sequence
    sequence(end+1) = bet_amount;
    end

end
end
fprintf('\n\nWin Rate: %.2f %%\n',100 * (win/runs))
fprintf('Loss Rate: %.2f %%\n',100 * (fail/runs))
fprintf('Average money walking away with: $%.0f / $%.0f \n',
    mean(average_winings), initial_starting)
average_when_completed(average_when_completed ==0) = [];
fprintf('Average amount won if sequence is completed: %0.f
\n',mean(average_when_completed) - initial_starting)

```

Published with MATLAB® R2021b