```
%%  Project - Optimization and Algorithms
%
%   2017/2018
%
%   Luis Miguel Martins Alves       75189
%
%   Ruben Miguel Oliveira Tadeia    75268
%
%   Pedro Guilherme Moreira Falcao  77063
%
%   Diogo Henrique Monteiro Silva   79462
%
%   Script:
%      -> First Phase
%         - Part I

%% Reset Variable states
clc;
clear all;
close all;

%% Loading the dataset into variables
currentFolder = pwd;

datasetFile = strcat↙
(currentFolder,'/project5_gambling_dataset/dataset_project5_gambling.mat');

load (datasetFile);

%% Initialize figures and variables

% Number of iterations
iter = 100;
trajectories = 1000;

figure(1); clf;
figure(2); clf;
figure(3); clf;

%% Solve optimization problem - Kelly
cvx_begin quiet
    variable b(N, 1)

    % Build cost function
    f = sum(prob_outcome_i' * log(returns_matrix * b));
    maximize(f);

    % Subject t
    sum(b) == 1;
    b >= 0;
cvx_end;

%% Solve optimization problem - Risk Constraint Kelly
alfa = 0.9;
beta = 0.05;
lambda_risk = log(beta)/log(alfa);

cvx_begin quiet
    variable b_rck(N, 1)

    % build cost function
    function_rck = sum(prob_outcome_i' * log(returns_matrix * b_rck));
```

```
    maximize(function_rck);

    % subject to
    risk_constraint = log(sum(exp(log(prob_outcome_i) - lambda_risk*log(returns_matrix↙
* b_rck))));
    risk_constraint <= 0;
    sum(b_rck) == 1;
    b_rck >= 0;
 cvx_end;

%% Generating a random bets vector
b_rnd = rand(20,1);
b_rnd = b_rnd/sum(b_rnd);

%% Generating the graphics

% Init variables
w = zeros(K,trajectories);
w_rck = zeros(K,trajectories);
w_rnd = zeros(K,trajectories);

% For each trajectory
for i = 1:trajectories
    w(1,i) = 1;
    w_rck(1,i) = 1;
    w_rnd(1,i) = 1;

    % For each time instant
    for t = 2:iter
        r = rand;
        tmp1 = 0;
        tmp2 = prob_outcome_i(1);
        for j = 1:K
            if( r >= tmp1 && r <= tmp2)
                break;
            end
            tmp1 = tmp1 + prob_outcome_i(j);
            tmp2 = tmp2 + prob_outcome_i(j) + prob_outcome_i(j+1);
        end
        w(t,i) = w(t-1,i) * (returns_matrix(j,:) * b);
        w_rck(t,i) = w_rck(t-1,i) * (returns_matrix(j,:) * b_rck);
        w_rnd(t,i) = w_rnd(t-1,i) * (returns_matrix(j,:) * b_rnd);
    end

    if(rem(i,100) == 0)
        figure(1); hold on;
        plot(1:iter,w(:,i));
        hold off;

        figure(2); hold on;
        plot(1:iter,w_rck(:,i));
        hold off;

        figure(3); hold on;
        plot(1:iter,w_rnd(:,i));
        hold off;
    end
end

figure(1), xlabel('Trajectories'), ylabel('Wealth'), axis([0 100 0 700]);
figure(2), xlabel('Trajectories'), ylabel('Wealth'), axis([0 100 0 10]);
figure(3), xlabel('Trajectories'), ylabel('Wealth'), axis([0 100 0 10]);
```

```
%% Final comparation plots
w_mean = zeros(iter,1);
w_rck_mean = zeros(iter,1);
w_rnd_mean = zeros(iter,1);
for row = 1 : iter
    theSum = 0;
    theSum_rck = 0;
    theSum_rnd = 0;
    for col = 1:trajectories
        theSum = theSum + w(row, col);
        theSum_rck = theSum_rck + w_rck(row, col);
        theSum_rnd = theSum_rnd + w_rnd(row, col);
    end

    % Now get the mean over all values in this row.
    w_mean(row,1) = theSum / trajectories;
    w_rck_mean(row,1) = theSum_rck / trajectories;
    w_rnd_mean(row,1) = theSum_rnd / trajectories;
end

figure(4); hold on;
plot(1:iter,w_mean,'r');
plot(1:iter,w_rck_mean,'g');
plot(1:iter,w_rnd_mean,'b');
hold off;
```