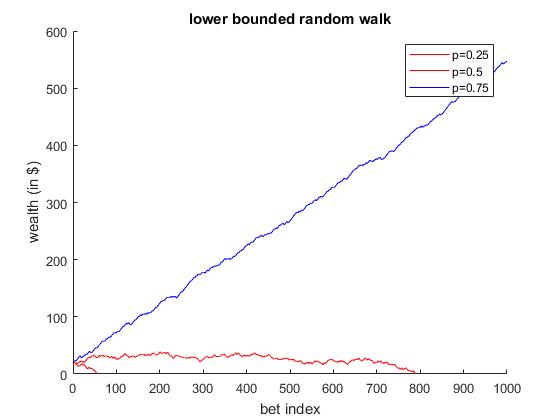
ESE 303 HW1

## Zeyu Zhao

# A

The plot that simulates three lower bounded random walks with success probability is the following:



### CODE:

function [w, bl] = random\_walk(p, w0, T)

% Initialize variables

%w0 = 20;

b = 1;

%T = 10;

%p = 0.3; % success probability

t = 1; w(t) = w0;

% repeat while not broke up to time maxt

while w(t) > 0 && t <= T

x = (rand() < p);

if x == 1

w(t + 1) = w(t) + b; % If x = 1 wealth increases by b

else

w(t + 1) = w(t) - b; % If x = 0 wealth decreases by b

end

t = t + 1;

end

if w(t) == 0

bl = false;

else

bl = true;

end

end

function walk\_plot(w0, T)

% w0 = 20;

% T = 10^3;

p = [0.25 0.5 0.75];

t = 0:1:T;

[w1, bl1] = random\_walk(p(1), w0, T);

[w2, bl2] = random\_walk(p(2), w0, T);

[w3, bl3] = random\_walk(p(3), w0, T);

x = t(1:length(w1));

figure

hold on;

if bl1

plot (x, w1, 'b');

else

plot (x, w1, 'r');

end

x = t(1:length(w2));

if bl2

plot (x, w2, 'b');

else

plot (x, w2, 'r');

end

x = t(1:length(w3));

if bl3

plot (x, w3, 'b');

else

plot (x, w3, 'r');

end

xlabel 'bet index';

ylabel 'wealth (in $)';

title 'lower bounded random walk';

legend('p=0.25', 'p=0.5', 'p=0.75')

hold off;

end

# B

After performing the experiment N times, I got the following results:

|  |  |
| --- | --- |
| N (number of times the experiment is repeated) | Probability of Ruin in the first 100 steps |
| 100 | 0.0900 |
| 500 | 0.0880 |
| 800 | 0.0825 |
| 1000 | 0.1020 |
| 1500 | 0.0967 |
| 2000 | 0.0885 |
| 5000 | 0.0894 |
| 10000 | 0.0901 |

Therefore, I conclude that the probability that the probability of ruin (with ) in the first bets is close to .

My criteria for selecting is starting from a number small enough but not too small that due to small samples, the experiment error can go above , since that is the accuracy of the number we are estimating. Hence, . But I also don’t want to be too large that the default accuracy of MATLAB limits the accuracy of results, therefore I choose . After trying some values, I find that seems to be a good choice.

### CODE:

% I fixed tried different N for part B

function p = prob\_ruin(N, w0, p)

ruins = 0;

for i = 1 : N

[~, bl] = random\_walk(p, w0, 100);

if ~bl

ruins = ruins + 1;

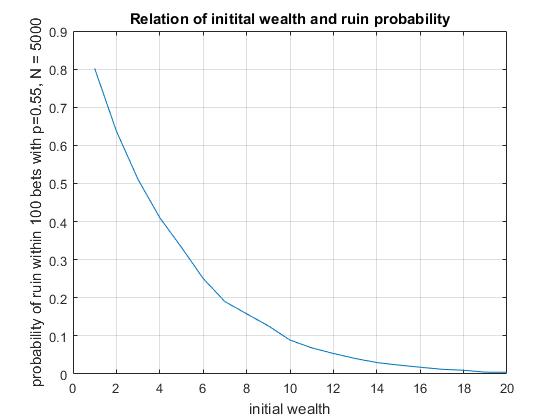
end

end

p = ruins / N;

end

# C

The plot of probability estimates of ruin as a function of initial wealth is as follows:

### CODE:

function ruin\_wplot()

w0 = 1:20;

N(1:20) = 5000;

p(1:20) = 0.55;

ruin\_prob = arrayfun(@prob\_ruin, N, w0, p);

figure

plot(w0, ruin\_prob);

xlabel 'initial wealth'

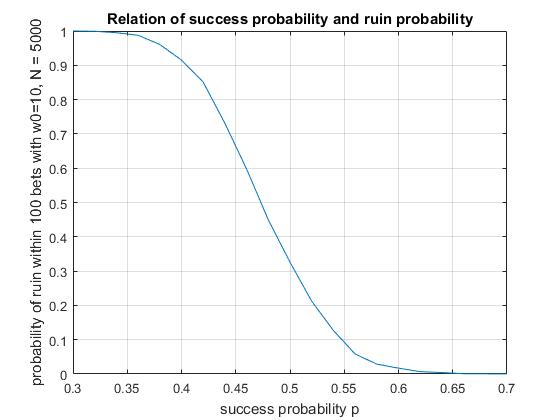
ylabel 'probability of ruin within 100 bets with p=0.55, N = 5000'

title 'Relation of initital wealth and ruin probability'

grid on;

end

# D

The plot of probability estimates of ruin as a function of success probability is as follows: 

### CODE:

function ruin\_pplot()

p = 0.3:0.02:0.7;

w0(1:length(p)) = 10;

N(1:length(p)) = 5000;

ruin\_prob = arrayfun(@prob\_ruin, N, w0, p);

figure

plot(p, ruin\_prob);

xlabel 'success probability p'

ylabel 'probability of ruin within 100 bets with w0=10, N = 5000'

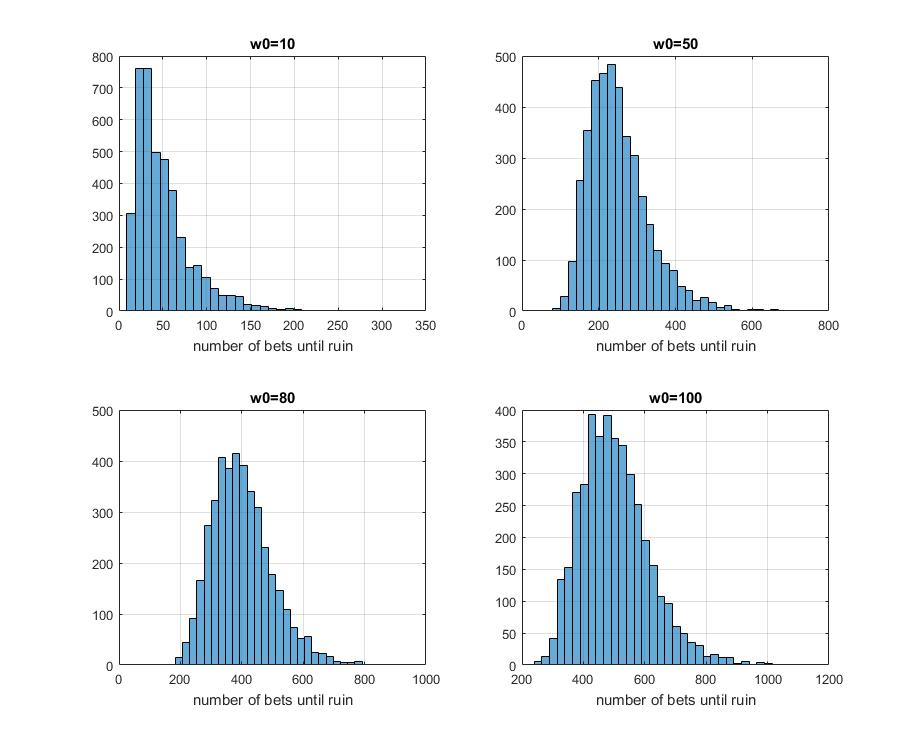
title 'Relation of success probability and ruin probability'

grid on;

end

# E

When experimenting with different values of initial wealth, the distributions of differs. The histograms that demonstrates that is as follows.



When w0 = 10, average T0 = 50.572.

When w0 = 50, average T0 = 250.862.

When w0 = 80, average T0 = 400.168.

When w0 = 100, average T0 = 502.388.

The distribution of resembles a heavy-tailed Chi-Square distribution. Therefore, I think that on average approximately .

### CODE:

function t = random\_walk\_till\_ruin(p, w0)

% Initialize variables

b = 1;

t = 1; w(t) = w0;

% repeat while not broke up to time maxt

while w(t) > 0

x = (rand() < p);

if x == 1

w(t + 1) = w(t) + b; % If x = 1 wealth increases by b

else

w(t + 1) = w(t) - b; % If x = 0 wealth decreases by b

end

t = t + 1;

end

end

function until\_ruin\_plot(lw)

N = 4096;

p = zeros(N, 1);

p(:) = 0.4;

w0 = zeros(N,1);

i = 1;

nplot = sqrt(length(lw));

for w = lw

w0(:) = w;

t = arrayfun(@random\_walk\_till\_ruin, p, w0);

subplot(nplot, nplot, i);

histogram(t, 32);

title (['w0=', int2str(w)])

xlabel 'number of bets until ruin'

grid on;

i = i + 1;

fprintf('When w0 = %5d, average T0 = %6.3f.\n', w, mean(t))

end

end