Random Walker Simulation with Speed Zone

Table of Contents

Document Info

```
% Written by: Drew Christner
% Class: MAE 215 Final Project Part 4
% Date: 14 April 2021
```

Variables

```
home = 0 ; % starting spot of walker
steps = 2500 ; % number of steps per simulation
direction = zeros(1,steps+1) ; % creates zero vector to fill with
 direction data
position = zeros(1,steps) ; % creates zero vector to fill with
 position data
position(1,1) = home ; % fills position vector's origin with the home
 variable
t = (1:steps+1) ; % time vector
spd = 20 ; % how much the speed zone moves the walker
cent = (rand*150-rand*150) ; % RNG variable for position of the center
 of speed zone
reps = (100000) ; % number of times the 2500 step simulation is run
```

The Loops

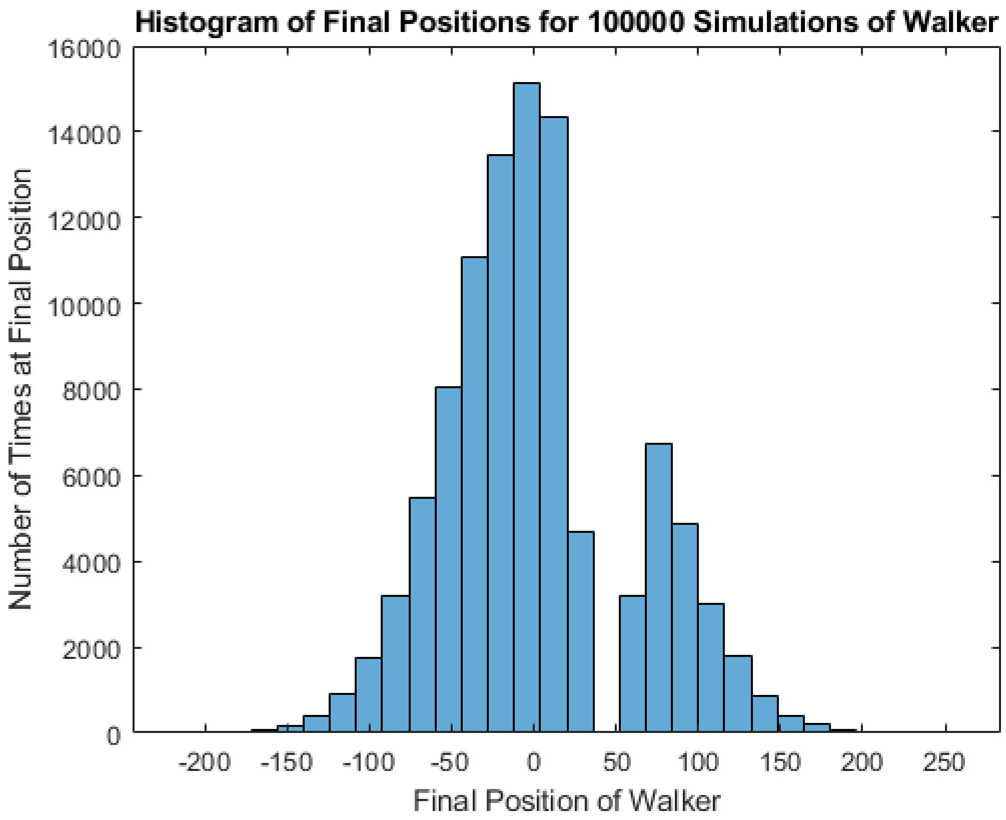
```
for L = [10,20,40,80] % L is the distance from cent that the speed
 zone extends

fin = zeros(1,reps) ;
j = 1 ; % variable for tracking reps in loop, resets each loop

% This for loop allows the code to be run multiple times with
 different
% values for L thus running the 2500 step, 100000 rep simulation with
% different sized speed zones (they have the same center) and the
 final
% position of each rep will be saved and graphed to analyze the effect
 of
```

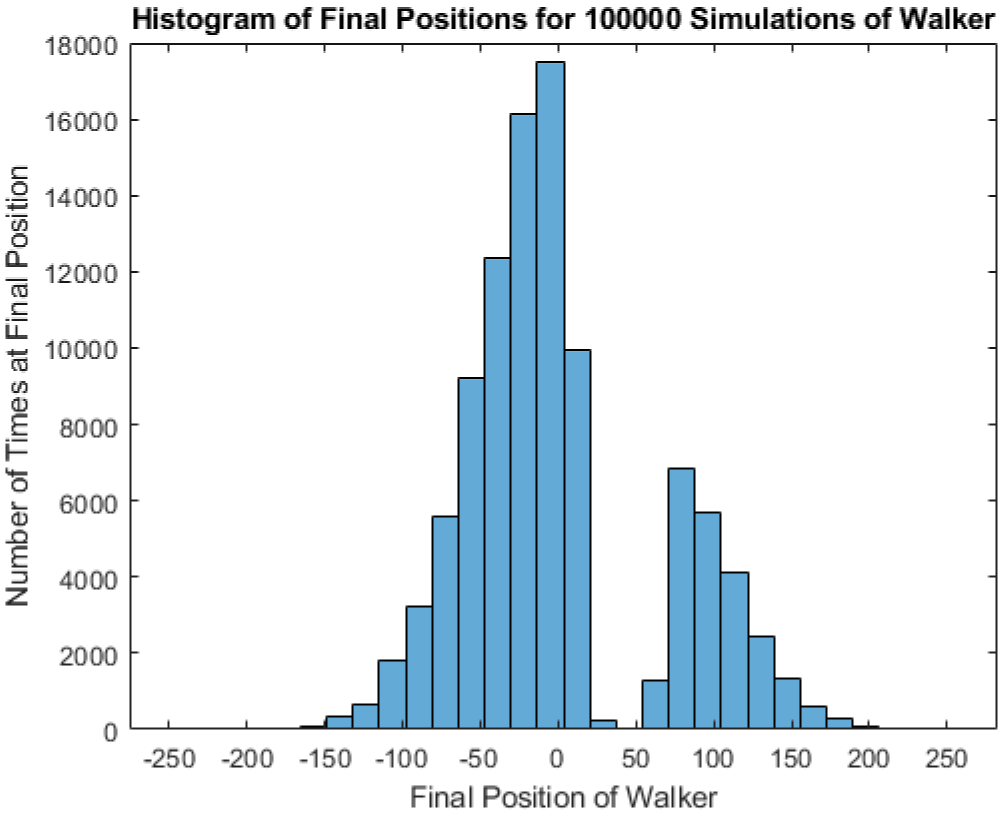
```
% the size of speed zone on the data.

while j < reps
    i = 1 ; % variable for tracking while loop
    flip = rand(1,steps+1) ; % randomly generates a number for each
 step

% This while loop loops for every value from 1 to the user inputted
 value of reps.
% This is where the randomly generated numerical value that will later
 calculate the direction
% of motion is generated




while i < steps+1 % loops from 0 to # steps + 1 cuz 0,0 is home
      i = i+1 ;

% This while loop repeats until times repeated equals the user
 inputted
% number of steps.  This also creates the variable i.  Posion is a
 function
% of this i value that creates a vector full of the positions of the
 runner
% at every step.

    if flip(i) < .5 % rng determines direction stepped
        direction(i) = 0 ;

        if (position(i-1)>=cent - L) && (position(i-1) <= cent + L)
            position(i) = position(i-1)-spd ;
        else
            position(i) = position(i-1)-1 ;
        end

    else
        direction(i) = 1 ;

        if  (position(i-1)>=cent - L) && (position(i-1) <= cent + L)
            position(i) = position(i-1)+spd ;
        else
            position(i) = position(i-1)+1 ;
        end

    end

% The first if statment uses the random number from earlier to
 determine the
% direction the walker steps in.  A number less than .5 causes him to
 walk
% left.  A  number greater than .5 will cause him to walk right.
 Walking
```

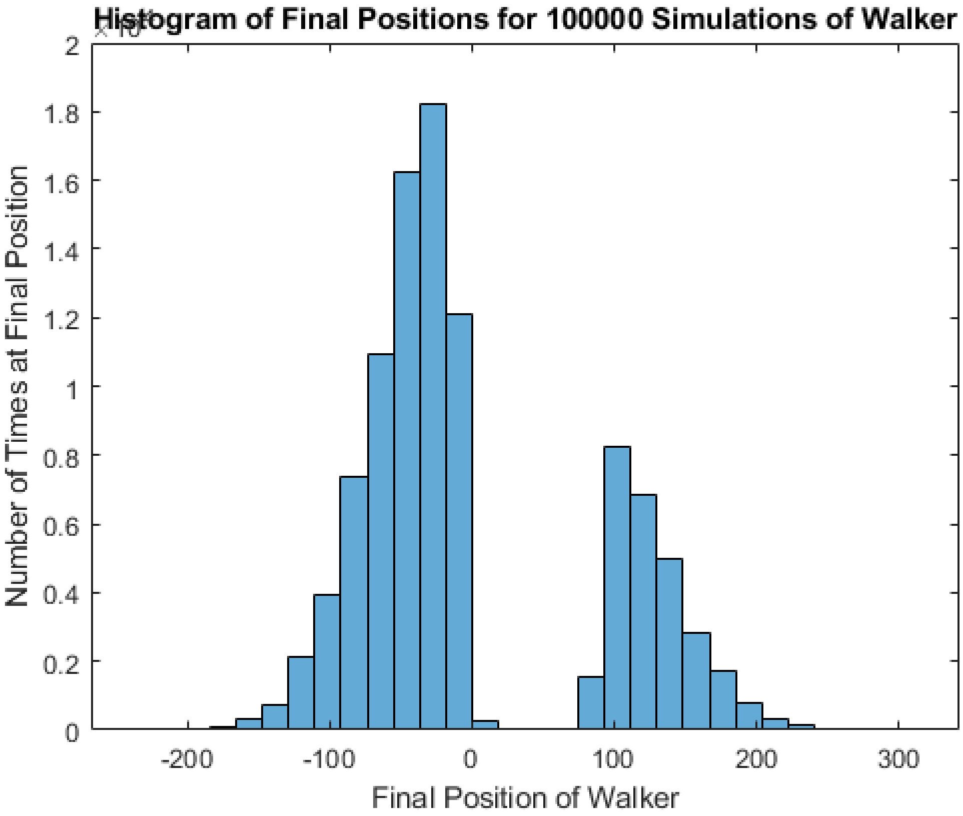
```
% is done by taking the position vector's previous term and adding one
 to
% it for a right step and subtracting one for a left step.  The two
 nested
% if statements decide whether the walker is in the speed zone or not.
% This is done by determ,ining if the previous position value is
 within the
% zone (cent +/- L) and if so takes "spd" steps rather than 1

end

fin(1,j) = position(1,i) ; % creates vector of data of position of
 final step
j = j+1 ;

end
```

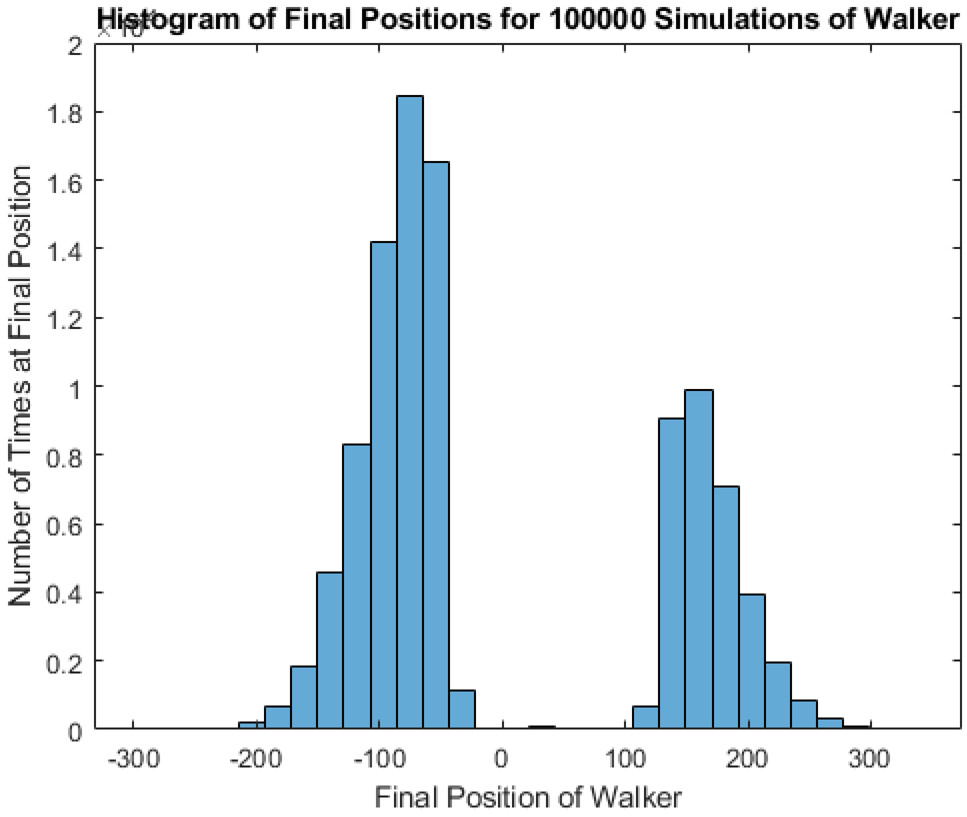
Graphing

```
histogram(fin,30)
xlabel('Final Position of Walker')
ylabel('Number of Times at Final Position')
title(['Histogram of Final Positions for ' num2str(reps) ' Simulations
 of Walker'])
figure
```




Histogram of Final Positions for 100000 Simulations of Walker
Number of Times at Final Position
Final Position of Walker
0
2000
4000
6000
8000
10000
12000
14000
16000
18000
-250
-200
-150
-100
-50
0
50
100
150
200
250


Histogram of Final Positions for 100000 Simulations of Walker
$\times 10^4$
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-200
-100
0
100
200
300
Final Position of Walker
Number of Times at Final Position


Histogram of Final Positions for 100000 Simulations of Walker
$\times 10^4$
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Number of Times at Final Position
-300
-200
-100
0
100
200
300
Final Position of Walker

```
end
```

Analysis

```
% As the width of the speed zone increases, the histograms displays a
 "blank
% space" right near the center of the speed zone.  This space
 increases
% proportional to the size of the speed zone.  Additionally, as the
 speed
% zone is increased, the range of the data increases to include larger
% numbers.  Finally there is a high concentration of steps from the
 walker
% on either edge of the speed zone.  The size of the speed zone had
 less of
% an effect than I had anticipated; I thought it would allow the
 walker to
% reach much higher extremes but it only increased them by a little
 bit.
```

Published with MATLAB® R2020a