

LARSON—MATH 353—HOMEWORK WORKSHEET h10
Random Walks.

1. Log in to your Sage Cloud account.
 - (a) Start the Chrome browser.
 - (b) Go to `http://cloud.sagemath.com` and sign in.
 - (c) You should see an existing Project for our class. Click on that.
 - (d) Click “New”, call it **h10**, then click “Sage Worksheet”.

Random Walks

Start at the origin on the number line. At each time step take a (random) step one unit to the right or one unit to the left. I have heard that you will (with probability 1) return to the origin at some point, Is this true? How can we investigate this experimentally? If it is true, how many steps does it take on average to return to the origin?

Let's practice our simulation **methodology**.

2. Define a function `experiment()` that simulates a random walk and outputs/returns the number of steps it takes to return to the origin. (Of course, it is possible to keep taking steps to the right and never returning to the origin. Maybe this doesn't happen in practice?)
3. Define a function `repeated_experiments(n)` that repeats our single experiment simulation n times and outputs/returns the average number of steps it takes to return to the origin. How large does n need to be so that the average converges?
4. Repeat `experiment()` 1000 times, record the data somehow, and visualize the data using `histogram`. You'd expect to see the results crowded near the origin and then decreasing. (Is this distribution *normal*—or maybe some other common distribution?)
5. Use your data from the last problem to estimate the probability that a random walker will return to the origin in no more than 10 steps.

Getting your homework recorded

When you are done...

- (a) Click the “Make pdf” (Adobe symbol) icon and make a pdf of this worksheet. (If Cocalc hangs, click the printer icon, then “Open”, then print or make a pdf using your browser).
- (b) Send me an email with an informative header like “Math 353—h10 worksheet attached” (so that it will be properly recorded).
- (c) Remember to attach your homework worksheet!