Random Walks in Mathematica

Chloe Wawrzyniak

Summer 2018

1 1-dimensional Random Walks

1.1 Real-World Data

Create a list in Mathematica of the recorded results from the partner activity from this morning. Forward steps should be recorded as a 1 and backwards steps should be recorded as a -1. For example, if you flipped the coin four times and the results were forward, backwards, forward, forward, then your list would be $\{1, -1, 1, 1\}$. Give your list a name (for example, 1DRandomWalk), so you can use it in later computations.

Create another table that adds up the flips to that point. Namely, each item in the list would be your displacement after that many flips. For the example listed above, the new table would be $\{1, 0, 1, 2\}$. Give this list a name as well (for example, 1DRandomWalkDisplacement). Note: You shouldn't compute these numbers by hand. You should make Mathematica do the heavy computing for you by using a loop.

Now, create a line graph that shows your displacement over time by applying the function ListLinePlot to the displacement list you just created.

1.2 Generated Data

Use mMthematica's built in functions to create a list of 1's and -1's, of length 500. Hint: RandomInteger[] will output either 0 or 1. Can you apply some basic algebra to that to make it output 1 or -1 instead? Don't forget to give this list a name.

Just like with the real-world data, create a table that adds the flips up to each point, namely, it lists the displacement after each flip. Again, you should use a loop to create this list from the first one. Create a line plot just as before.

1.3 Generated Data and Histograms

Using the same techniques and functions we've used before, have Mathematica create a list of 100 lists of length 20, whose entries are 1's and -1's.

Create a list of the final displacement of each of the 100 lists you just created. Note: we don't need the displacement of each step. We just want the final displacement of each list. Use Mathematica's built-in Histogram function to create a list of the results. What do you notice?

2 2-dimensional Random Walks

2.1 Real-World Data

Enter the data from the last activity from the morning into Mathematica using the following convention:

1. Forward: $0, 1$
2. Backwards: $0, -1$
3. Left: $-1, 0$
4. Right: $1, 0$

In other words, enter your data as ordered pairs which correspond to the direction a person would move who is standing at the origin, facing the positive y-axis.

Just as before, create a table that computes the location after each step. Remember: now that we're looking at things in 2-D, this will be a list of ordered pairs.

Use Mathematica's built-in ListPlot function to plot these values. (This won't draw the lines between them, as the previous one did. This will just give us the points). What do you notice about the distribution of the points on the plane?

2.2 Generated Data

Again using RandomInteger, create a list of 500 ordered pairs like the ones listed in the above subsection. You'll need to be more creative than in the one-dimensional case, because points like $\{0, 0\}$ or $\{1, -1\}$ are not options.

Create a table that computes the location after each step. Use ListPlot to graph these points. What do you notice about their locations?

2.3 Generated Data and Histograms

Using the same techniques as above, have Mathematica create 100 lists of length 20 of ordered pairs like the one above. Compute the end location for each list, and then compute it's distance to the origin (Hint: the distance from the point (x, y) to the origin is $\sqrt{x^2 + y^2}$). Plot a histogram of these values.

3 Challenge: 3-dimensional Random Walks

Do the same thing you just did in sections 2.2 and 2.3 above, but for 3 dimensions. What do you notice? Is it more similar to or different from the 2-dimensional case?