Tutorial: Random Walk

In Chapter 14, the book goes through a "case study" of code development. The goal of this tutorial is to go through the same development process on your own.

But, we already did some of that in class. The first step is therefore to get the code we were working on in class; it is in the `Tutorial14_randomwalk` subdirection in the `ASTR2600_materials` git repository.

You have the choice of working in IDL or python. I've included code that should accomplish the same thing in either language.

Open the programs in `gvim` and start hacking away!

At the end of class, before you leave, make sure to `git push|.`

The "big-picture" goals of this code:

1. Track a random walk for N steps
2. Plot random walks
3. Determine the *average* distance a random walk will take you as a function of the number of steps

You should make at least plots of the average distance and some of the individual random walks. See where you can add color to the plots!

Along the way, you should have `print` statements interspersed through your code so you can see what you're doing. You also need to comment the code well, including adding comments to the code you've been given where it's needed (e.g., the test code!).

Remember to *commit often*! For this tutorial, some of the credit will come from seeing intermediate steps - e.g., print statements added to the code like on page 8 of chapter 14 - that are present in some commits, but not your final commit.

Features your code should include:

- A number of steps option
- A seed option (so that you can take continuous pseudorandom steps)
- A step size option *with a default size of 1*
- A plot keyword option (set to True if you want to plot)

Once you've completed the above, make the same code but with a *random* step size, where it's up to you to interpret "random" (i.e., it could be random between 0 and 10, or random normal around 5, or anything of that sort).

Do the plots of the random step size look similar to the fixed step size random walk? Does it have the same average behavior over time?

At the end of class, before you leave, make sure to `git push`.

Workflow

How should you go about working on this code? "Workflow" refers to the way you write, edit, run, and debug code.

You should have the `.pro` or `.py` files open in `gvim`. Note that you can open multiple tabs in `gvim` - I recommend doing this if you have multiple files open!

You should also have an interactive `IDL` or `ipython` session open. But, in order to commit your changes, you'll also need a `bash` shell open (the thing starting with `$`). `Konsole`, the "terminal" application, also allows you to have multiple tabs open: again, I recommend using the tabs!

The recommended workflow is:

1. Make an edit in `gvim` and save it
2. `%run` or `.run` the code in python/IDL to test it
3. If and when something goes awry, add `print` statements to print out the values of variables that are doing weird things
4. Repeat.

The most important thing is testing various steps along the way; don't wait until you're "done" with your code before testing the whole thing!

See Chapter 14, page 34 for a list of the steps Dewey took when he was writing this code (he goes through his solution in great detail).

More Details on Random Walking

The problem we're addressing is described in Chapter 14.2 of the text, starting on page 6 of chapter 14.

What does "average distance a random walk will take you as a function of the number of steps" mean? I'll explain using some other examples.

Ex. 1:
If you flip a random coin 10 times and record how many heads and tails you get, the numbers could be 6H and 4T, 5H and 5T, 9H and 1T, or any combination. However, if you repeat that experiment 100 times (i.e., you flip a coin 1000 times total), the *average* behavior should always be that the number of heads and tails are (nearly) the same. With a random walk, estimating the distance (which can be computed with the pythagorean theorem) as a function of the number of steps, is the average distance traveled 0? Or something else?

Ex. 2:
Say you have some formula for walking steps that isn't random. Every other step, you go +1 in x and then the next step +1 in y. So if you take 5 steps, it looks like:
step 0: $x = 0, y = 0$, distance $= 0$
step 1: $x = 1, y = 0$, distance $= 1 = \sqrt{1^2 + 0^2}$
step 2: $x = 1, y = 1$, distance $= \sqrt{2} = \sqrt{1^2 + 1^2}$
step 3: $x = 2, y = 1$, distance $= \sqrt{5} = \sqrt{2^2 + 1^2}$
step 4: $x = 2, y = 2$, distance $= \sqrt{8} = \sqrt{2^2 + 2^2}$
step 5: $x = 3, y = 2$, distance $= \sqrt{13} = \sqrt{3^2 + 2^2}$
The distance is increasing at each step, but by a different amount each time. In this case, you can directly measure the distance traveled and it will always be the same; in the random walk case, you can measure the distance traveled for any given random walk, but you must average those distances to find out the average behavior of a random walk.