Random Walk Project

Math 242

due ~~Monday, April 19~~ Wednesday, April 21

Consider the following model of two-dimensional random walk that is not confined to integer-valued coordinates: start at the origin, and choose a random unit vector (in any direction) for each step. That is, to determine a step of the walk, first choose an angle θ (uniformly) from the interval $[0, 2\pi)$. The step of the walk will then be $(\cos(\theta), \sin(\theta))$.

Your Tasks

1. Implement the random walk. Make a few plots of sample random walks to show that your implementation works.

2. Find the average distance of the random walk from the origin after n steps. How does this average distance depend on n? Make a plot showing how the average distance of the random walk from the origin depends on n, and find a function that roughly fits your plot.

3. Since the walk is not on a grid, it's very unlikely that it will return *exactly* to $(0, 0)$. Instead, investigate whether it returns to within a circle of radius $\frac{1}{2}$ around the origin. Specifically, what proportion of n-step random walks return at least once to within $\frac{1}{2}$ unit of the origin? How does this proportion depend on n? Investigate for various n and make a plot of your results. Do you think that *all* random walks will eventually return near the origin? Why or why not?

4. What happens if we constrain the random walk to the region $-5 \leq y \leq 5$? Modify your code to keep the y-coordinate of the walk between -5 and 5. Make a few plots of random walks to show that your modified implementation works. Then repeat tasks 2 and 3 for this constrained random walk.

Your Report

Turn in *either* a Python Colab notebook or a Mathematica notebook. Make sure that you clearly answer the questions above, and include computations to support your answers. As usual, submit code that runs and explain what your code does. Your goal should be to communicate your work to another person (e.g., another student at your level who is not in this course).

Grading Rubric

Your notebook will be graded on a scale of 0 to 16 points. The following rubric gives characteristics of notebooks that will merit sample point totals. (Interpolate the following for point totals that are not divisible by 4.)

16 points. Problems and goals are clearly stated, including relevant definitions or parameters. Computations are complete; code runs and is clearly explained. Conclusions are clearly stated and backed up by sufficient computational evidence. Limitations of the methodology, extensions for future work, and conjectures are discussed. Notebook is well-formatted and easy to read.

12 points. Problems and goals are stated well, though relevant definitions or parameters may be missing. Computations are mostly complete; code runs, but explanation is weak. Conclusions are unclear or not well justified. Insufficient discussion of limitations, extensions, and conjectures.

8 points. Statement of problem or goal is unclear. Computations are incomplete; explanation is ambiguous. Code may produce errors when run. Conclusions are possibly correct, but not justified. Little or no discussion of limitations, extensions, or conjectures. Notebook is difficult to read.

4 points. Serious misunderstanding of the problem or goal. Computation is inadequate for the task at hand. Work is not clearly explained. No discussion of limitations, extensions, or conjectures. Notebook is difficult to read.

0 points. Notebook is not turned in.