Object-Oriented Programming in Python: Cool Graphs and Animations Before the Holidays

Introduction

This tutorial aims to exemplify the use of the object-oriented interface of the Matplotlib library. Through practical exercises, students will learn how to plot and animate data in Python, deepening their understanding of object-oriented programming and mathematical concepts. Please make your graphs visually appealing!

Exercise 1: Simple Random Walk

The first exercise involves plotting a simple random walk on the integers. Students will use Matplotlib's object-oriented API to plot the random walk and automatically highlight the maximum and minimum points with arrows and annotations.

Objectives

- Generate a random walk sequence with no loop.
- Plot the sequence using Matplotlib's object-oriented interface.
- Smartly annotate a graph.

Exercise 2: Illustrating the Central Limit Theorem

In the second exercise, students will illustrate the Central Limit Theorem (CLT) using random walks. This involves generating multiple random walks, plotting their endpoints, and comparing the resulting distribution to a Gaussian distribution.

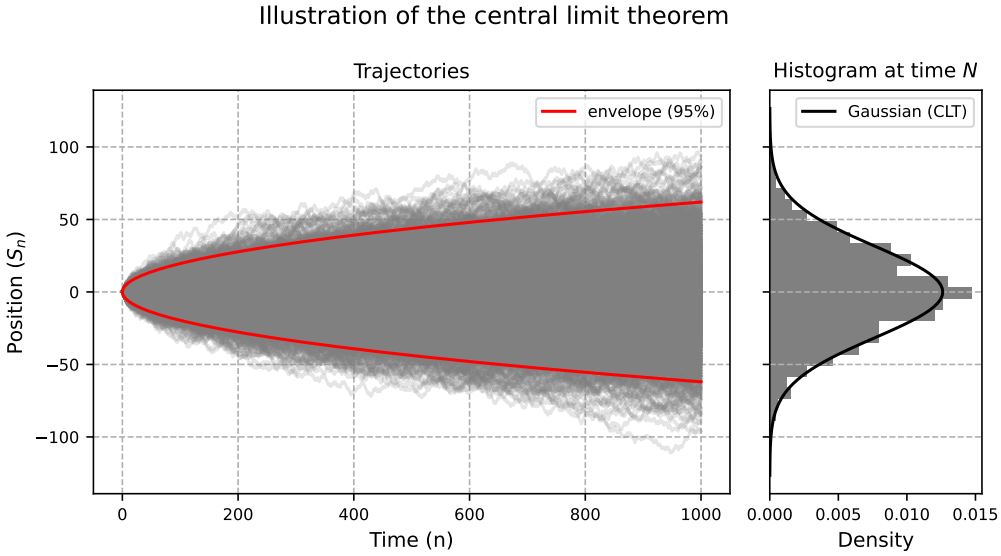

Figure 1: Random walks and the CLT

Objectives

- Generate and plot multiple random walks.
- Plot the distribution of the endpoints of these walks.
- Overlay the Gaussian density function to illustrate the CLT.

Exercise 3: Animating a Random Walk

The final exercise focuses on creating an animation of a random walk. Using Matplotlib's animation functionality, students should visualize the step-by-step progression of a random walk.

Objectives

- Progressively generate a random walk sequence.
- Create an animation (animated gif or video) showing the evolution of the random walk over time. Please make it beautiful.

Conclusion

Through these exercises, students will gain practical experience with the object-oriented interface of Matplotlib. Data visualization skills are essential for both academic and professional pursuits in data analysis and scientific computing.