A Benchmark Study of Python and R:

Estimating the Value of Pi Using a Monte Carlo Simulation

James C.K.K., Garo B., and Manny H.

MSDS 460: Decision Analytics

May 19th, 2024

Abstract

This study conducted a benchmark analysis utilizing and comparing Python and R in estimating the value of Pi via a Monte Carlo Simulation method. The performance of each language is evaluated based on execution time, memory usage, ease of implementation, code readability, and scalability. Python’s versatility and integration capabilities were weighed against R's specialized statistical functions. Findings revealed differences in the ability of Python and R to handle Monte Carlo Simulation with implications for their use and implementation. Compared to R, Python consistently displayed faster execution times for each sample size. This research provides insights that can inform data scientists in selecting the appropriate programming language for Monte Carlo Simulations.

Introduction

A data scientist’s programming language of choice has significant implications for the efficiency and effectiveness of computational tasks. This research aims to compare the performance of Python and R, two popular, data science programming languages, by implementing a Monte Carlo Simulation to estimate the value of Pi. The Monte Carlo Simulation, which involves repeated random sampling to solve numerical problems, is chosen for its computational intensity and suitability for parallel processing, making it an ideal candidate for performance benchmarking.

The motivation for this study stems from the need to understand how different programming environments handle intensive computational tasks. By evaluating execution time, memory usage, ease of implementation, code readability, and scalability, we sought to provide a comprehensive comparison that can inform the use of Python or R for a Monte Carlo Simulation.

Literature Review

Research on estimating the value of Pi using Monte Carlo Simulations has been conducted in various programming languages, including Python and R, to benchmark their performance and accuracy. One such study on Monte Carlo estimation using Python is presented by GeeksforGeeks (2022). This study demonstrated the implementation of the Monte Carlo method to estimate Pi by generating random points in a square and counting how many fall within an inscribed circle. Then, the ratio of points inside the circle to the total points, multiplied by four, estimated Pi. The algorithm's effectiveness increased with the number of iterations, thus showcasing Python's capability to handle iterative and computational tasks efficiently.

Similarly, Gustafsen (2022) explored the use of R for estimating Pi through a Monte Carlo Simulation. This study explained how to generate random points within a square, calculate their distances from the origin, and determine if they were within the unit circle. By comparing the ratio of points inside the circle to the total number of points and multiplying by four, and then estimates of Pi were obtained. The study emphasized the improvement in estimation accuracy with an increased number of points, highlighting R's robust statistical and computational capabilities. Both studies applied the fundamental principles of the Monte Carlo Simulation and demonstrated the strengths of Python and R in executing these tasks efficiently.

Method

The goal of this analysis was to compare the utility of Python and R in the implementation of Monte Carlo Simulation to estimate values of Pi. All analyses were conducted in Jupyter Notebook using Python and in RStudio using R.

To conduct the Monte Carlo Simulation, the methodology utilized by previous research (Gustafsen, 2022) was implemented in this benchmark analysis in both programming languages. First, two random points were generated within a circle, then the distance between the two points and the origin of the circle was calculated, next the ratio of the two points was multiplied by four to estimate values of Pi. Seven analyses were performed that varied by sample size: 100; 1,000; 10,000; 100,000; 1,000,000; 10,000,000; 100,000,000. The total time for each iteration was measured and assessed for all seven analyses.

Results

Implementing Monte Carlo Simulations for estimating Pi in Python and R took roughly the same amount of time given the input was straightforward. Both programming languages provided robust libraries for random number generation and time measurement and made the process relatively easy. Data retrieval time involved reading the computed results and printing them to the console. Both Python and R performed these tasks efficiently. However, the R Markdown document's output was automatically formatted for presentation, which can be slightly more time-consuming than plain Python console output.

In the Python implementation of this Monte Carlo Simulation (refer to Table 1 for all data), sample sizes ranged from 100 to 100,000,000 where each of the seven analyses increased the sample size by a factor of 10. The elapsed time for the seven Python analyses ranged from 0.0000 to 27.9564 seconds. The percent of error varied from a maximum of 3.13% to a minimum of 0.0007%. In the R implementation, sample sizes were identical to the Python implementation. However, the elapsed time for the seven R analyses ranged from 0.0289 to 274.2 seconds. The percent of error varied from a maximum of 7.0535% to a minimum of 0.0031%. Python consistently displayed faster execution times for each sample size compared to R. The error percentages were similar between the two languages.

	Python		**R**	
Sample sizes	**Time (seconds)**	**Error (%)**	**Time (seconds)**	**Error (%)**
100	< 0.000	3.1324	0.0289	7.0535
1,000	0.0003	0.8406	0.0029	1.7059
10,000	0.0030	0.0762	0.03966	0.1143
100,000	0.0311	0.3185	0.3508	0.0664
1,000,000	0.2939	0.0111	2.991	0.1034
10,000,000	2.8272	0.0067	21.71	0.0046
100,000,000	27.9564	0.0007	274.2	0.0031

Table 1. Benchmark analysis of Python and R: Monte Carlo Simulation to Estimate Values of Pi

Conclusion

Utilization and Market Share

Python

- **Market Share**: Python is highly popular in data science and machine learning due to its simplicity and extensive libraries (e.g., NumPy, pandas, SciPy).
- **Documentation and Training**: Python has comprehensive documentation and abundant training resources, including tutorials, courses, and community support.
- **Ease of Use**: Python is user-friendly, with a syntax that is easy to learn for beginners and powerful for experienced users.

R

- **Market Share**: R is widely used in statistical analysis and data visualization. It has a strong presence in academia and among statisticians.
- **Documentation and Training**: R also has excellent documentation and a wealth of training resources, including official manuals, online courses, and a strong user community.
- **Ease of Use**: R is particularly strong in data manipulation and visualization, with packages like ggplot2 and dplyr. However, its syntax can be more challenging for those not familiar with statistical programming.

Management Problem Addressed

The benchmark study addresses the efficiency and accuracy of Monte Carlo simulations in estimating pi using different programming languages. This is relevant for businesses and researchers who need to choose the most efficient tool for large-scale simulations and statistical computations.

Cost of Implementation

- **Python**: Python is open-source and free to use. The primary costs would be associated with training, support, and any specialized libraries or tools that might be required.
- **R**: R is also open-source and free. Costs would similarly include training, support, and any additional packages or tools needed for specific tasks.

Overall, both Python and R provide cost-effective solutions for statistical simulations and data analysis. The choice between them may come down to specific use cases, existing expertise within a team, and the particular strengths of each language in relation to the tasks at hand.

References

GeeksforGeeks. (2022, July 28). *Estimating the value of pi using Monte Carlo*. https://www.geeksforgeeks.org/estimating-value-pi-using-monte-carlo/

Gustafsen, A. (2022, February 26). *Estimating PI using Monte Carlo Simulation in R*. Medium. https://towardsdatascience.com/estimating-pi-using-monte-carlo-Simulation-in-r-91d1f32406af