

Week 3 Assignment

Q1. Execution Time Tabulation

Mean Execution Time (in milliseconds)

Python

Method	Mean Execution Time	Standard Deviation
For-loop	7.3 ms	0.8 ms
NumPy Vectorized	0.38 ms	0.10 ms
SciPy cdist	0.17 ms	0.025 ms

R

Method	Mean Execution Time	Standard Deviation
For-loop	9924.52 ms	5419.29 ms
Vectorized (Matrix Ops)	424.38 ms	1083.10 ms
Base R dist() Function	512.22 ms	721.70 ms

The vectorized approach and the Base R dist() function significantly outperform the traditional for-loop approach in execution speed.

Q2. Execution Time Range

Python

Method	Min Execution Time	Mean Execution Time	Max Execution Time
For-loop	6.5 ms	7.3 ms	8.1 ms
NumPy Vectorized	0.3 ms	0.38 ms	0.5 ms
SciPy cdist	0.15 ms	0.17 ms	0.2 ms

R

Method	Min Execution Time	Mean Execution Time	Max Execution Time
For-loop	5781.20 ms	9924.52 ms	18010.60 ms

Vectorized (Matrix Ops)	25.10 ms	424.38 ms	2963.10 ms
Base R dist() Function	74.20 ms	512.22 ms	2062.00 ms

Vectorized matrix operations and the dist() function provide significant improvements over the for-loop in terms of execution time.

Q3. Preference Based on Computational Efficiency & Implementation

Computational Efficiency

The fastest computation time is achieved using Python’s SciPy cdist function and NumPy vectorized operations, while in R, the dist() function and vectorized matrix operations are the best choices. The for-loop is significantly slower, making it unsuitable for large datasets.

Implementation Complexity

- **For-loop:** Simple to implement but computationally inefficient.
- **Vectorized approach:** More efficient but requires an understanding of matrix operations.
- **Base R dist() function & SciPy cdist:** Highly efficient and easy to use for distance calculations.

Verdict

If speed is the only factor, SciPy’s cdist function in Python and the dist() function in R are the best options. However, vectorized operations are also a strong alternative if additional flexibility is needed.

Q4. Additional Considerations for Choosing Python vs. R

Availability & Ecosystem of the Library

- R excels in statistical computing and built-in data visualization.
- Python is more versatile, especially for machine learning and general-purpose programming.

Industry Adoption

- Python is widely used in software development and machine learning.
- R is commonly used in academia and research, especially in medical and social sciences.

Conclusion

The choice between Python and R depends on the task. If statistical analysis is the priority, R is the better choice. For machine learning and general-purpose computing, Python is more suitable.