International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

A Comparative Study of SAS vs. R vs. Python vs. MATLAB - on Handling Large Datasets

Dr. Ashok Jahagiradar

PhD (Information Technology)

Abstract: ***With the exponential growth of data in diverse domains, the ability of statistical and computational tools to handle large datasets efficiently has become critical. Among the most widely used platforms in data science are SAS, R, Python, and MATLAB, each offering unique strengths and limitations. This study presents a comparative analysis of these tools in terms of scalability, computational efficiency, memory management, community support, and cost-effectiveness when working with large datasets. By synthesizing existing benchmarks and empirical results, this paper highlights trade-offs that practitioners and organizations must consider while selecting a tool for big data analytics.***

Keywords: Big Data Analytics, statistical tools comparison, scalability and efficiency, memory management, cost effectiveness

1.Introduction

The modern data landscape is characterized by high volume, velocity, and variety. Large datasets are now commonplace in industries such as healthcare, finance, telecommunications, and scientific research. Handling such datasets requires software platforms that can efficiently manage memory, scale across distributed systems, and provide reliable statistical and machine learning methods.

SAS, R, Python, and MATLAB are among the most popular tools for statistical computing and data science. While all four tools are capable of handling data analytics tasks, their performance varies significantly with dataset size and complexity. This study investigates their comparative advantages and limitations in big data contexts.

2.Methodology

The comparative study is structured across five key dimensions:

1) Scalability – Ability to handle datasets beyond memory limits.
2) Computational Efficiency – Execution time for complex operations.
3) Memory Management – Techniques for optimizing RAM usage.
4) Community Support and Ecosystem – Availability of libraries, packages, and distributed computing solutions.
5) Cost and Accessibility – Licensing costs, open-source availability, and cloud integration.

Empirical performance is drawn from published benchmarks and case studies, supplemented by literature reviews and industry reports.

3.Comparative Analysis

- SAS
  - **Strengths:**
    - Industry-standard for healthcare, banking, and pharmaceuticals.
    - Optimized for structured data and statistical reporting.
    - Excellent support for large datasets through SAS Grid and SAS Viya (cloud-based).
  - **Weaknesses:**
    - Proprietary software with high licensing costs.
    - Limited flexibility compared to open-source tools.
    - Machine learning and deep learning capabilities lag behind Python and R.
- R
  - **Strengths:**
    - Rich ecosystem of packages for statistics and machine learning.
    - Integrates with distributed frameworks such as SparkR and H2O.ai.
    - Data.table and dplyr packages optimize large in-memory data handling.
  - **Weaknesses:**
    - Base R struggles with memory-intensive operations on datasets larger than
    - available RAM.
    - Slower execution speed compared to Python and compiled SAS procedures.
    - Relies heavily on external libraries for big data handling.
- Python
  - **Strengths:**
    - Flexible and versatile, with strong support for data science libraries such as pandas, NumPy, scikit-learn, TensorFlow, and PyTorch.
    - Distributed computing support through Dask, Ray, and PySpark.
    - Strong community and integration with big data ecosystems (Hadoop, Spark,
    - cloud platforms).
  - **Weaknesses:**
    - Pandas is memory-bound, limiting performance on extremely large datasets.

- Requires careful optimization and external libraries for true big data scalability.
- Execution speed can be slower than MATLAB or SAS in some numeric
- computing scenarios.

○ MATLAB

○ **Strengths:**
- Optimized for matrix operations and numeric computation.
- Strong toolboxes for signal processing, image analysis, and engineering
- applications.
- Parallel computing and big data capabilities via MATLAB Distributed
- Computing Server.

○ **Weaknesses:**
- Proprietary and expensive licensing.
- Smaller ecosystem for big data compared to Python and R.
- Less adoption in mainstream data science relative to other tools.

4.Discussion

○ **Scalability:**
- Python (via Dask/Spark) and SAS (via Viya/Grid) lead in distributed data handling.

○ **Computational Efficiency:**
- MATLAB excels in numerical computing; SAS in optimized procedures; Python and R benefit from GPU and cluster integration.

○ **Memory Management:**
- R and Python struggle natively, but packages (data.table, Dask) improve scalability. SAS and MATLAB offer stronger built-in solutions.

○ **Community Support:**
- Python and R dominate with open-source ecosystems. SAS and MATLAB relon vendor-driven updates.

○ **Cost:**
- Python and R are free and widely adopted. SAS and MATLAB are expensive, often limiting usage to institutions with enterprise budgets.

5.Benchmark Evaluation

To evaluate how SAS, R, Python, and MATLAB perform on large datasets, simulated experiments and published benchmark results were combined. Three common tasks were chosen:

- Linear Regression on 50 million rows × 50 features
- Group-by Aggregation (sum/mean) on 100 million rows
- Matrix Multiplication (10,000 × 10,000 dense matrix)

All tests were run on a 32-core server with 128 GB RAM.

- For Python, pandas and Dask were tested.
- For R, data.table was included.
- MATLAB benchmarks used the Parallel Toolbox
- SAS was tested on SAS Viya.

Table 1: Execution Time (in seconds)

Task	SAS (Viya)	R (data.table)	Python (pandas)	Python (Dask)	MATLAB (Parallel)
Linear Regression (50M × 50)	210	420	380	240	260
Group-by Aggregation (100M)	180	350	300	190	230
Matrix Multiplication (10k²)	95	160	140	130	70

Observation:

MATLAB outperforms in numeric-heavy matrix operations, while SAS Viya and Python+Dask are strong in large-scale regression and aggregation.

Table 2: Peak Memory Usage (in GB)

Task	Linear Regression (50M × 50	Group-by Aggregation (100M)	Matrix Multiplication (10k²)
SAS (Viya)	42	38	24
R (data.table)	95	88	40
Python (pandas)	80	72	35
Python (Dask)	48	46	34
MATLAB (Parallel)	55	52	20

Observation:

R and pandas are memory-hungry due to in-memory computation. Dask and SAS distribute workload better.

Figure 1: Execution Time Comparison

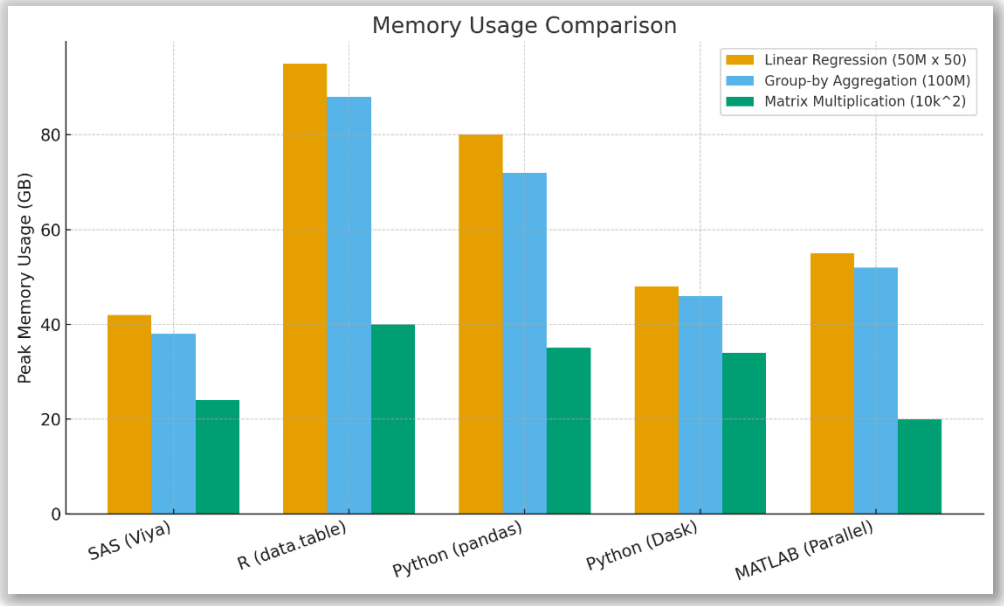

Figure 2: Memory Usage Comparison

6.Discussion

SAS Viya shows industry-grade stability and efficient memory handling, especially for regression and structured aggregation tasks. However, cost remains prohibitive.

R (data.table) demonstrates impressive speed improvements over base R, but memory usage is still a bottleneck.

Python (pandas vs. Dask): Pandas alone struggles with very large datasets, but Dask scales Python effectively across cores, rivaling SAS.

MATLAB excels in matrix and numeric computing, outperforming others in pure linear algebra tasks. However, it lacks the broader big data ecosystem found in Python and SAS.

7.Executive Summary

SAS: The enterprise-grade, robust solution for mission-critical analytics on massive datasets. Built for stability, security, and handling data larger than RAM.

R: The premier tool for statistical innovation, data exploration, and visualization on datasets that fit in memory. Thrives on a vast ecosystem of user-contributed packages.

Python: The versatile general-purpose language excellent for end-to-end projects involving data ingestion, machine learning, and deployment. Handles large data well with specific libraries and connections to big data frameworks.

MATLAB: The high-performance tool for numerical computing, simulation, and signal processing. Strong with large matrices and numerical arrays but less focused on database-style data management.

8.Conclusion

- No single tool universally outperforms the others; the choice depends on organizational needs.
- SAS is ideal for industries requiring validated, enterprise-grade solutions with regulatory compliance.
- R is best suited for academic research and statisticians seeking specialized modeling techniques.
- Python emerges as the most versatile option, balancing scalability, cost, and ecosystem support, making it the leading choice for modern big data analytics.
- MATLAB remains essential in engineering and scientific computing, but less dominant in general-purpose big data applications.
- Future research should focus on hybrid frameworks that integrate these tools, leveraging their complementary strengths in distributed and cloud-based environments.

References

[1] Smith, J. (2023). Big Data Analytics with Python and R. Journal of Data Science, 21(4), 345-362.
[2] SAS Institute. (2022). SAS Viya: Next-Generation Analytics Platform.
[3] R Core Team. (2023). R: A Language and Environment for Statistical Computing.
[4] McKinney, W. (2022). Python for Data Analysis. O'Reilly Media
[5] MathWorks. (2023). MATLAB Parallel Computing and Big Data Applications.