Journal of Statistics Applications & Probability
An International Journal

http://dx.doi.org/10.18576/jsap/130133

Performance Review of Major Packages in Light of Data Exchange Capabilities in R Ecosystem

Anant P. Awasthi[1,], Niraj K. Singh[1], and Masood H Siddiqui[2]*

[1]Department of Statistics, Amity University – Noida Campus, Gautam Buddha Nagar, Uttar Pradesh, India
[2]Department of Statistics, University of Lucknow, Lucknow, Uttar Pradesh, India

Received: 28 Apr. 2022, Revised: 27 Jun. 2023, Accepted: 28 Jun. 2023.
Published online: 1 Jan. 2024

Abstract: An efficient strategy for managing data (exchange and manipulation) is essential for organization to run its data operations. In current scenario, when open-source computation platforms are widely used. The strategy guideline is need of the hour. It makes data flow efficient and fast in an information system. The work focused on benchmarking data exchange activities between data (excel and flat files) and R under the light of major computational frameworks native R [6], readr [7] and data.table [8]. It was concluded that for reading and writing excel files readxl [11] and writexl [10] are most efficient frameworks while for working with flat files data.table become un-disputed leader for both import and export exercises. These frameworks have out-performed when compared to its competitive frameworks like writexl and openxlsx [9] for readxl and writexl. "data.table" found superior than native R implementation and data.table.
Organization can consider these findings as guidelines and implement in their standard operating procedures so that data operations could me robust and more efficient. This will result into cost saving and optimum utilization of man-power and resources/hardware

Keywords: Data exchange in R; Data Import in R; Data Export in R; Performance Benchmarking; Data Exchange Guidelines.

1 Introduction

Data preparation accounts for about 80% of the work of data scientists. - Forbes [1].

Data preparation comprise of data import export named as data exchange, data management which involves cleaning, recoding, sorting, merging and other activities. Efficiency in mentioned activities directly leads to efficient workforce. Using standard practice in data management activities resource might more productive and this directly translates into profitability of organization.

Data exchange and management is a key aspect in any information system. As data flows from various format to computation engine and from computation engine back to files. During this process an efficient data exchange process and optimum memory utilization is import aspect of data flow in the information system. In information system, there are various types of data sources are used. Ranging from raw flat files to complex databases and data warehouses. In this paper, we have considered excel and flat data files as source and destination for data (as majority of data storage and exchange still take place in these formats).

R [6] is known as one of the most popular [2, 3, 4, 5] open-source language for data management and analysis. It provides a wide range of frameworks/packages for data management and data exchange. Native R has good capabilities to data exchange and manipulation. Over the period of time various other framework evolved and provide the same capabilities for data management and exchange.

There are not much studies known for benchmarking the performance regarding data exchange capabilities of these frameworks. The work discussed in this publication is focused on study of memory utilization and data exchange capabilities of major frameworks available in R. results from this study can be adopted by end users as guidelines for managing data exchanges activities with excel and flat files.

2. Data exchange in R

Data exchange is defined as flow of data from source to R (data import) and from R to source (data export). There are two major data sources (Microsoft Excel and flat data files) and three major framework methods (Native R, tidyverse, and data.table) were considered for benchmarking the performance in the light of data exchange capabilities. Data exchange run time was considered as cost of operation.

*Corresponding author e-mail: anant.awasthi@outlook.com

© 2024 NSP
Natural Sciences Publishing Cor.

3. Packages in Scope

R version 3.6.3 was considered as computational engine which was installed in an Ubuntu 20.04 machine.

Table 3.1: Packages in scope for study

Data Source	Data Import	Data Export
Excel	openxlsx 4.2.5; readxl_1.4.0	openxlsx 4.2.5; writexl_1.4.0
Flat data file	Native R (utils); readr_2.1.2; data.table_1.14.2	Native R (utils); readr_2.1.2; data.table_1.14.2

4. Methodology

Cost (Run Time) for data exchange capabilities of these packages was measured in a well-controlled computation environment. Data exchange activities were planned for various dataset sizes ranging from 10**2 to 10**5 records. These Datasets of size of 10**2 to 10**5 records were created from New York flight data (R package nycflights13) using simple random sampling with replacement.

Experiments were repeated 50 times each to avoid any impact on cost due to chance factor and at end of each experiment cost was recorded to execute the activity.

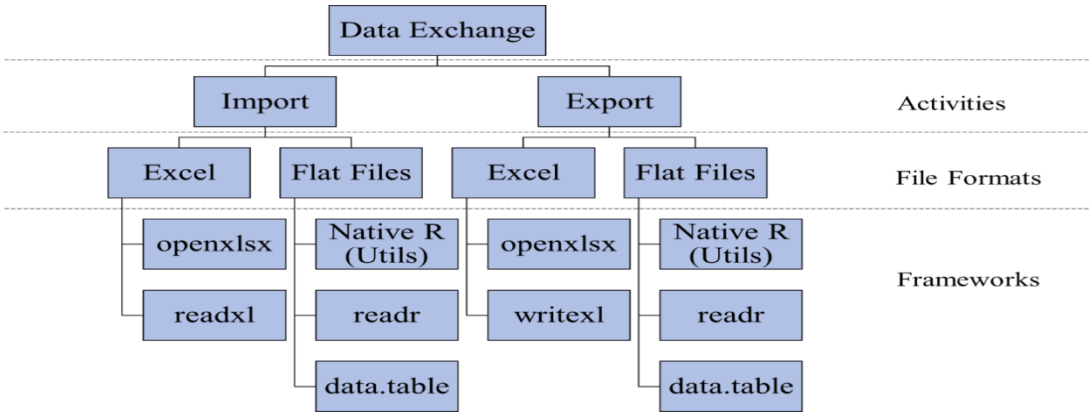

Fig 4.1 Structure of Experiment Methodology

4.1 Illustration of Experiment control Flow

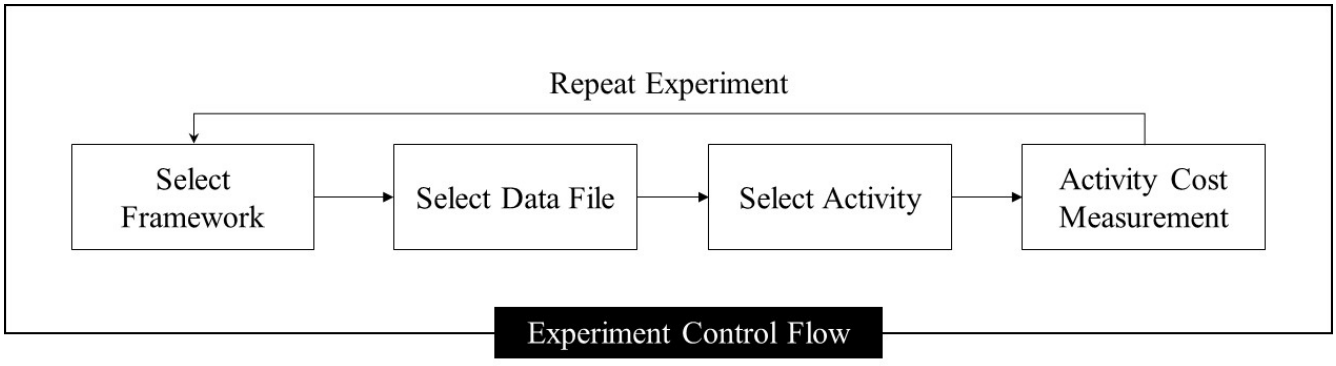

Fig 4.1.1 Illustration of Experiment control Flow

© 2024 NSP
Natural Sciences Publishing Cor.

5. Analysis and Results

The activity cost data was analysed under the light of an in-dept exploratory data analysis. The analysis helped in establishing conclusive comparison between the frameworks.

5.1 Excel -Import Activity

Cost of data exchange between excel and R for import activity shows openxlsx performance with respect to readxl get degraded for larger datasets. "readxl" provide a faster/better performance for almost all the datasets (excluding 10**2 records) when compared to openxlsx.

5.1.1 Numeric Analysis of Cost Data

Table 5.1.1.1: Mean run time (in seconds) for different data size import for Excel format

Dataset Size (Rows)	100		1,000		10,000		100,000	
Method	openxlsx	readxl	openxlsx	readxl	openxlsx	readxl	openxlsx	readxl
mean_run_time (in seconds)	0.008	0.012	0.052	0.032	0.515	0.204	7.315	2.335

5.1.2 Visual Analysis of Cost Data

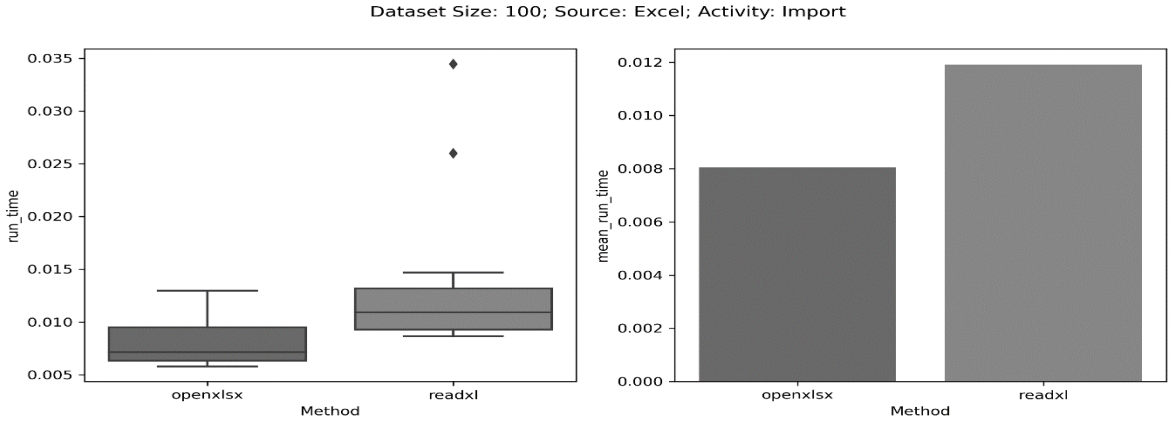

Fig 5.1.2.1 – Visual Analysis of Cost data (Import) for sample size 100 for excel data source

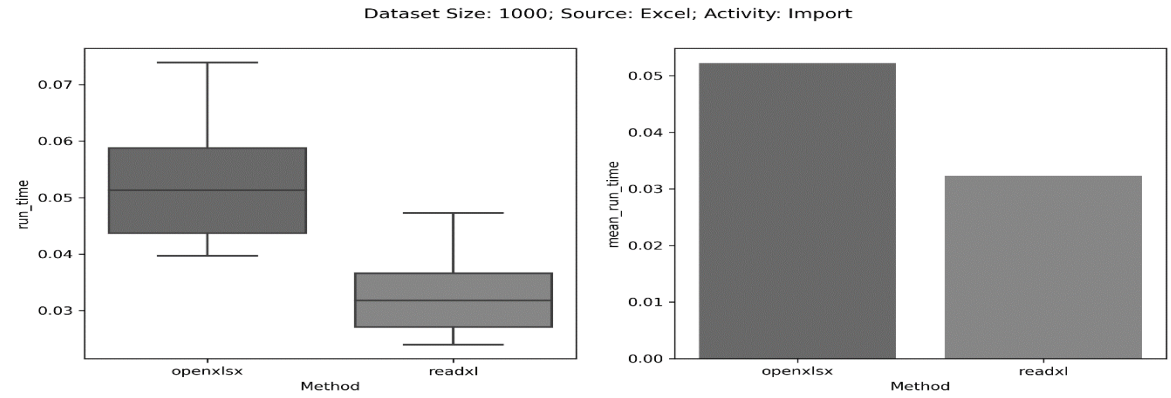

Fig 5.1.2.2 – Visual Analysis of Cost data (Import) for sample size 1000 for excel data source

© 2024 NSP
Natural Sciences Publishing Cor.

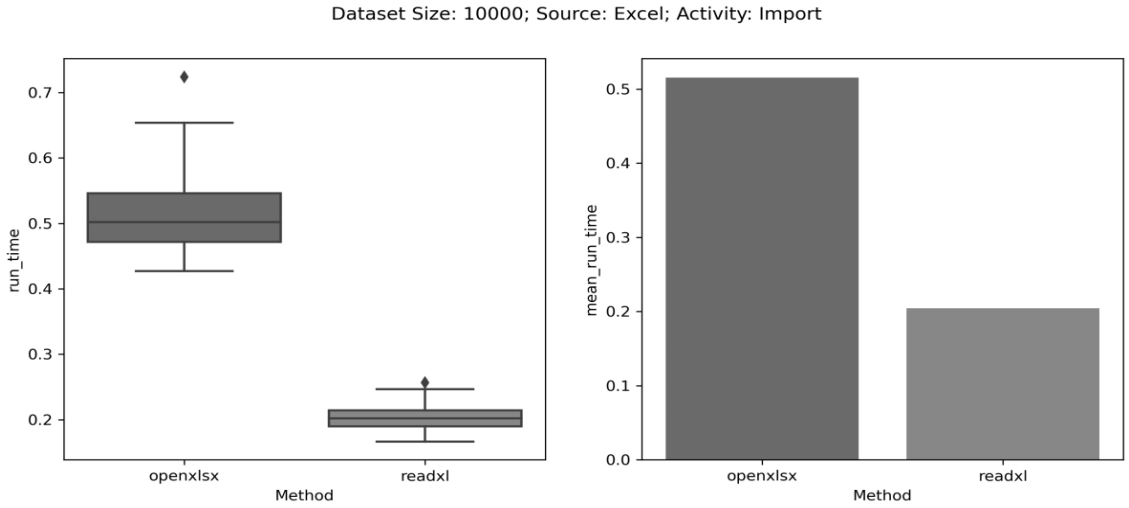

Fig 5.1.2.3 – Visual Analysis of Cost data (Import) for sample size 10000 for excel data source

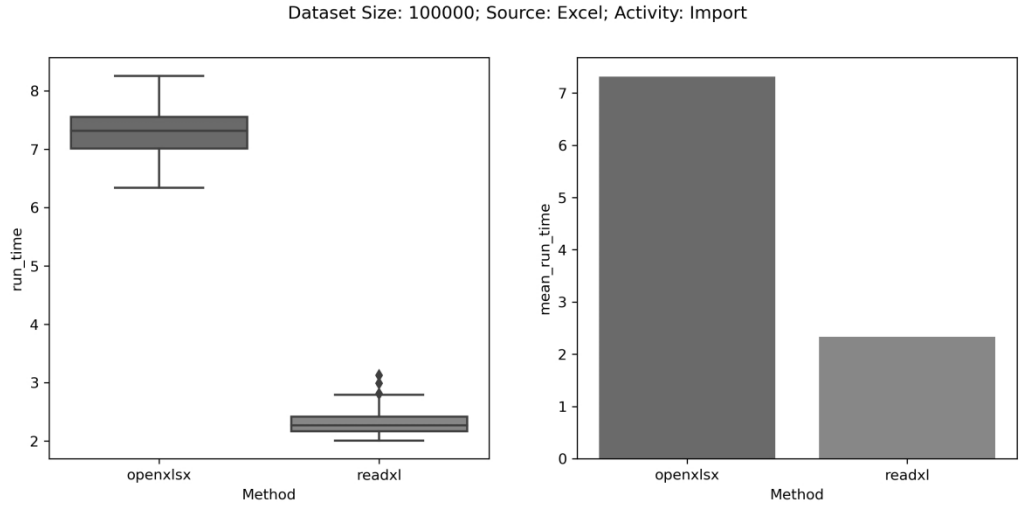

Fig 5.1.2.3 – Visual Analysis of Cost data (Import) for sample size 100000 for excel data source

5.1.3 Inference Analysis of Cost Data

Independent sample T Test was performed to check whether the performance difference in cost is due to packages (openxlsx and readxl) or just a chance factor.

Table 5.1.3.1: Independent Sample T Test Results for different data size import from excel format

Dataset		100	1,000	10,000	100,000
Ind-Sample T Test	Statistics	-5.70185	12.22026	34.97063	73.83525
	p-value	1.25E-07	6.92E-20	8.91E-42	6.58E-77

The inference analysis shows that for sizes of all datasets the difference in performance of both packages (openxlsx and readxl) has a significant difference (p-value <0.05).

5.2 Excel -Export Activity

Cost of data exchange between excel and R for export activity shows performance of writexl is far efficient when we compare to “openxlsx”. Performance of writexl is far better than openxlsx across all size of datasets.

© 2024 NSP
Natural Sciences Publishing Cor.

5.2.1 Numeric Analysis of Cost Data

Table 5.2.1.1: Mean run time for different data sizes export to Excel format

Dataset Size (Rows)	100		1,000		10,000		100,000	
Method	openxlsx	writexl	openxlsx	writexl	openxlsx	writexl	openxlsx	writexl
mean_run_time (in seconds)	0.059	0.009	0.175	0.056	1.486	0.517	15.947	5.349

5.2.2 Visual Analysis of Cost Data

Fig 5.2.2.1 – Visual Analysis of Cost data (Export) for sample size 100 for excel data source

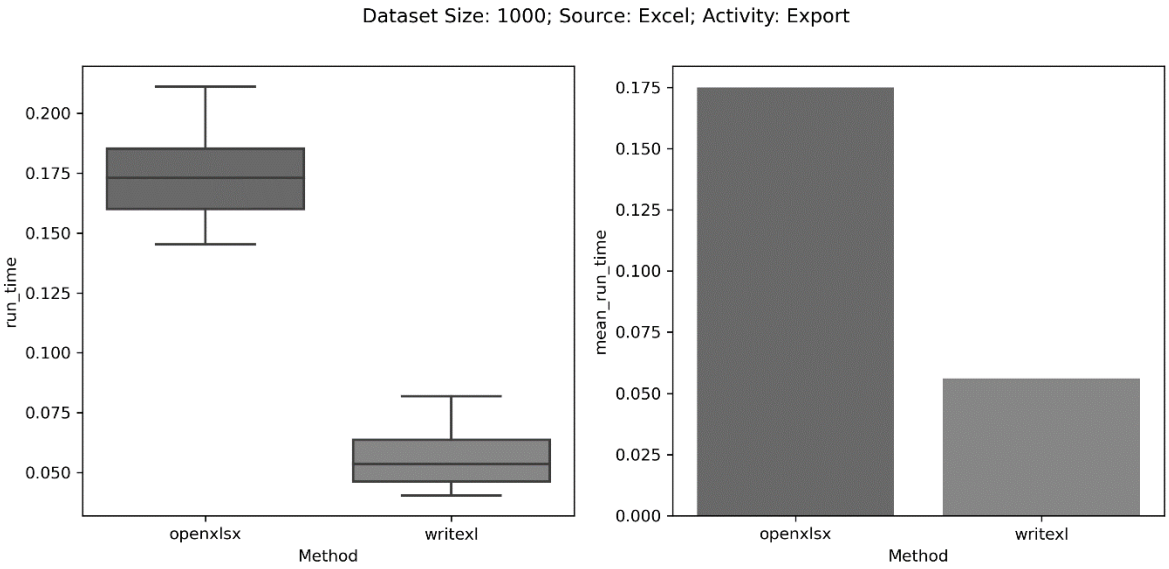

Fig 5.2.2.2 – Visual Analysis of Cost data (Export) for sample size 1000 for excel data source

© 2024 NSP
Natural Sciences Publishing Cor.

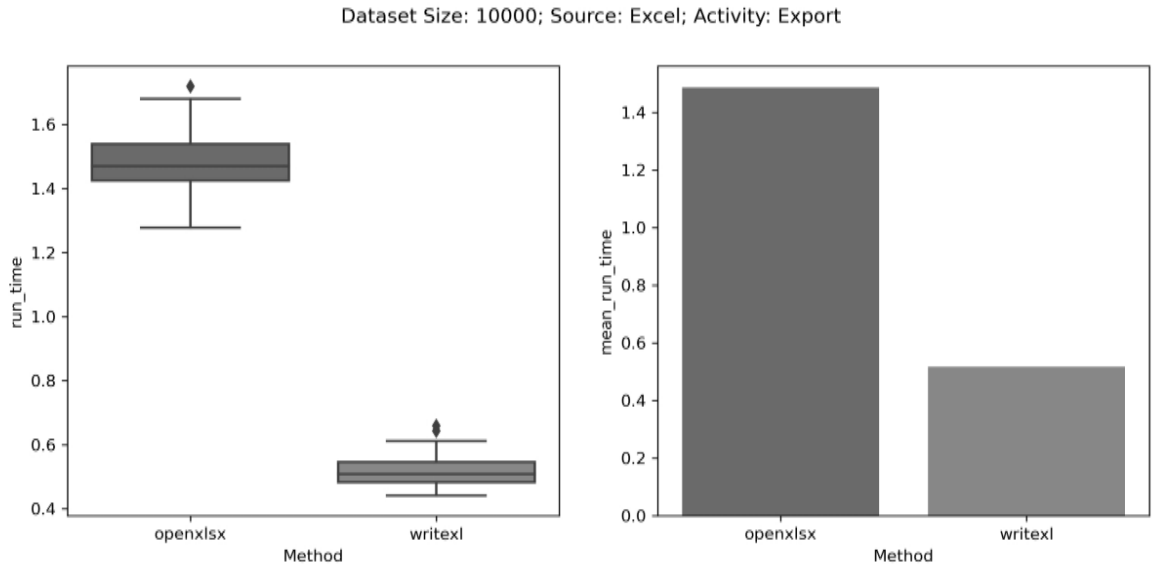

Fig 5.2.2.3 – Visual Analysis of Cost data (Export) for sample size 10000 for excel data source

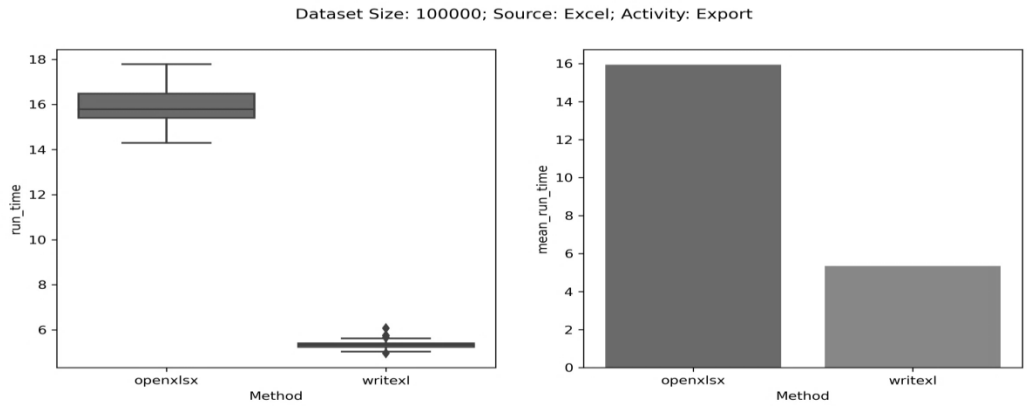

Fig 5.2.2.4 – Visual Analysis of Cost data (Export) for sample size 100000 for excel data source

5.2.3 Inference Analysis of Cost Data

Independent sample T Test was performed to check whether the performance difference in cost is due to packages (openxlsx and writexl) or just a chance factor.

Table 5.2.3.1: Independent Sample T Test Results for different data size export for excel format

Dataset		100	1,000	10,000	100,000
Ind-Sample T Test	Statistics	9.731686	39.17288	63.84702	94.14168
	p-value	4.74E-13	5.96E-55	1.02E-66	3.75E-63

The inference analysis shows that for sizes of all datasets the difference in performance of packages (openxlsx and writexl) has a significant difference (p-value <0.05).

5.3 Flat File (CSV) -Import Activity

While analysing the cost data for import of flat file, it was observed that data.table package was performing very well while compared to Base R (read.csv) and readr across all size of datasets. If we further analyse the data, Base R (read.csv) is a good performer on small datasets. When dataset size grows significantly, readr out performs than Base R (read.csv).

© 2024 NSP
Natural Sciences Publishing Cor.

5.3.1 Numeric Analysis of Cost Data (mean_run_time (in seconds))

Table 5.3.1.1: Mean run time for different data sizes import from CSV format

Dataset Size (Rows)	data.table	BaseR (read.csv)	readr
100	0.001	0.002	0.103
1,000	0.002	0.012	0.100
10,000	0.009	0.095	0.139
1,00,000	0.061	0.840	0.402

5.3.2 Visual Analysis of Cost Data

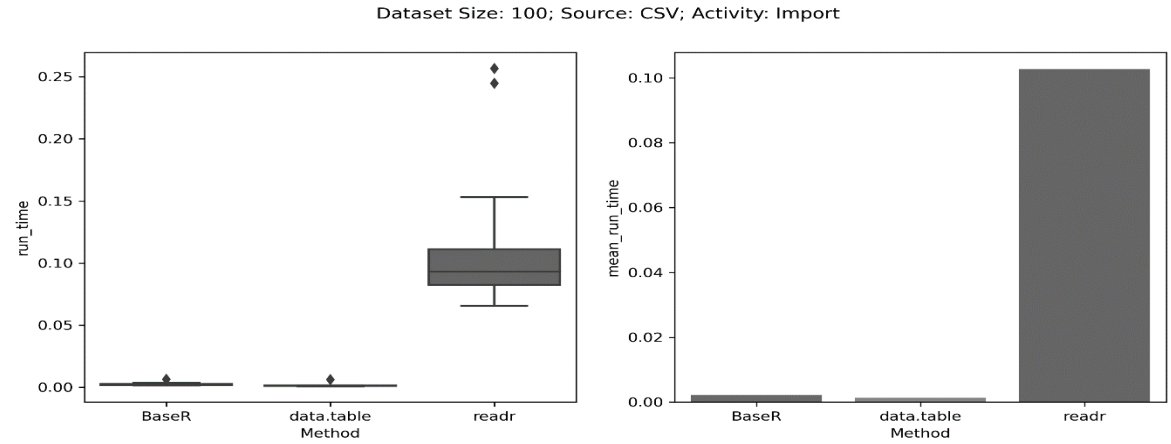

Fig 5.3.2.1 – Visual Analysis of Cost data (Import) for sample size 100 for CSV data source

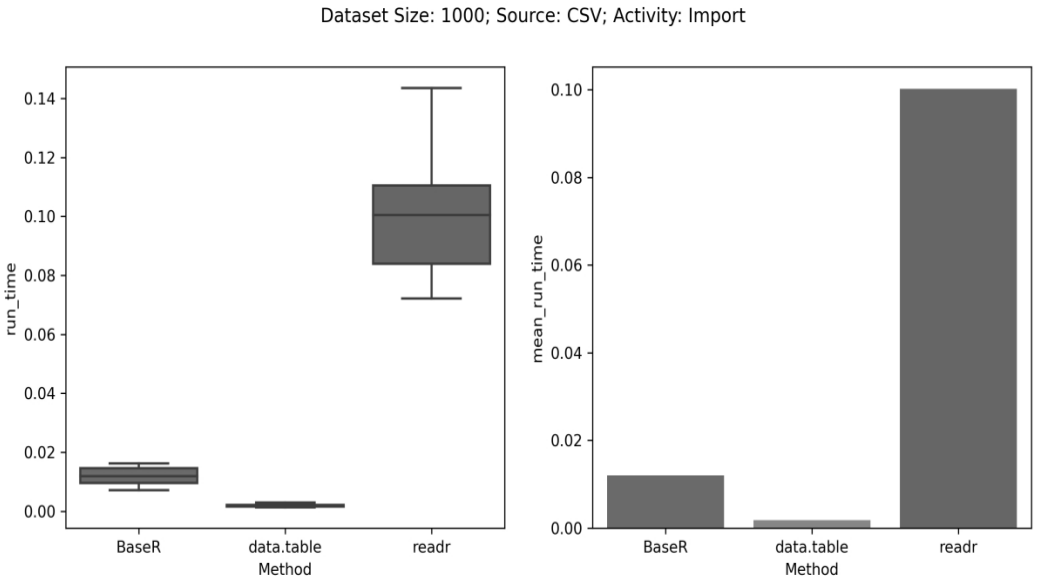

Fig 5.3.2.2 – Visual Analysis of Cost data (Import) for sample size 1000 for CSV data source

© 2024 NSP
Natural Sciences Publishing Cor.

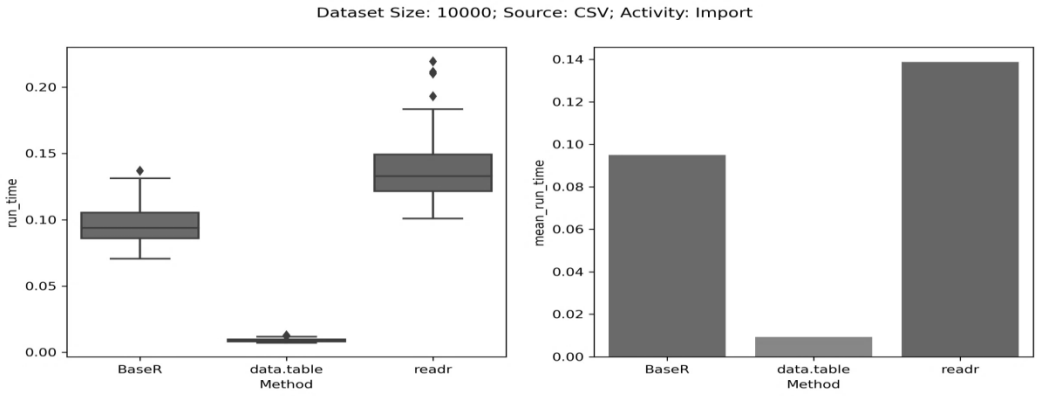

Fig 5.3.2.3 – Visual Analysis of Cost data (Import) for sample size 10000 for CSV data source

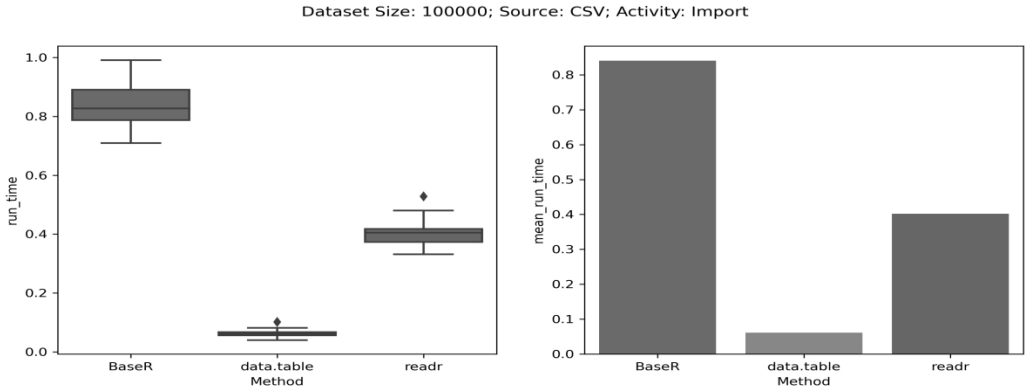

Fig 5.3.2.4 – Visual Analysis of Cost data (Import) for sample size 100000 for CSV data source

5.3.3 Inference Analysis of Cost Data

Analysis of variance was performed to test the significance of difference between the mean run time for all three frameworks (Base R (read.csv); readr; data.table).

Table 5.3.3.1: Analysis of Variance Results for different data size import from CSV format

Dataset Size (Rows)	100	1,000	10,000	100,000
F-Statistics	385.5907	1462.59	678.3049	3551.022
p-value	3.33E-59	9.34E-98	6.00E-75	3.69E-125

The analysis of variance shows that for sizes of all datasets the difference in performance of packages (Base R (read.csv); readr; data.table) has a significant difference (p-value <0.05).

© 2024 NSP
Natural Sciences Publishing Cor.

5.4 Flat file (CSV) -Export Activity

Analysis of export activity for flat file demonstrate that data.table shows unbeaten performance compared to Base R (write.csv) and readr.

5.4.1 Numeric Analysis of Cost Data (mean_run_time (in seconds))

Table 5.4.1.1: Mean run time for different data sizes export to CSV format

Dataset Size (Rows)	data.table	readr	Base R (write.csv)
100	0.002	0.014	0.004
1,000	0.006	0.035	0.033
10,000	0.012	0.052	0.244
1,00,000	0.08	0.306	2.489

5.4.2 Visual Analysis of Cost Data

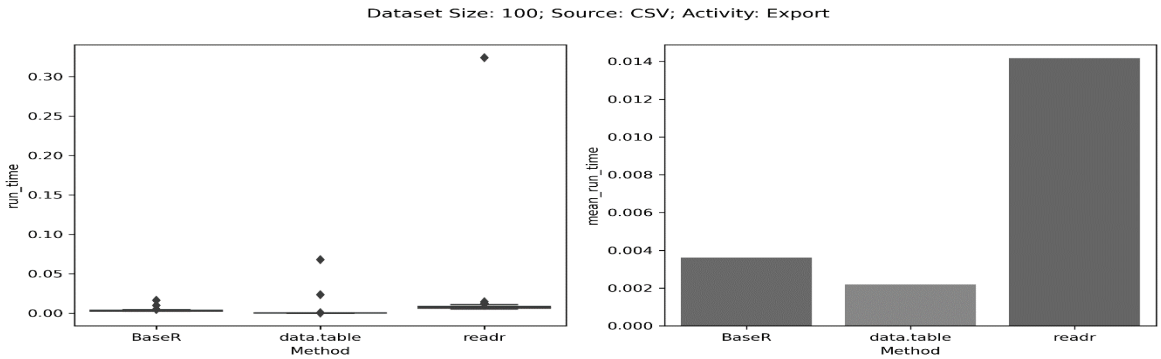

Fig 5.4.2.1 – Visual Analysis of Cost data (Export) for sample size 100 for CSV data source

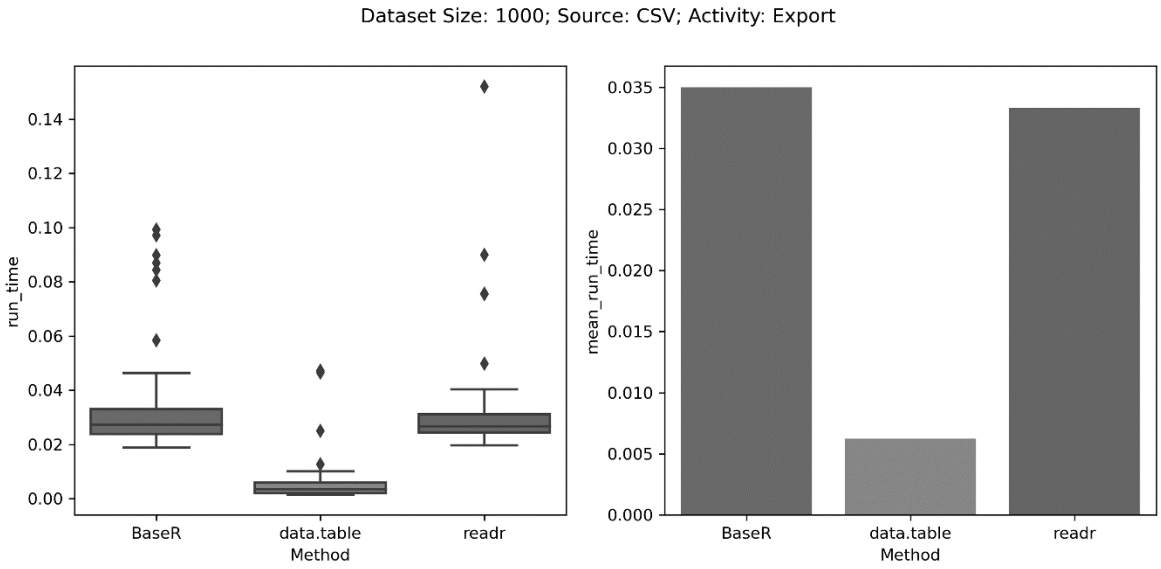

Fig 5.4.2.2 – Visual Analysis of Cost data (Export) for sample size 1000 for CSV data source

© 2024 NSP
Natural Sciences Publishing Cor.

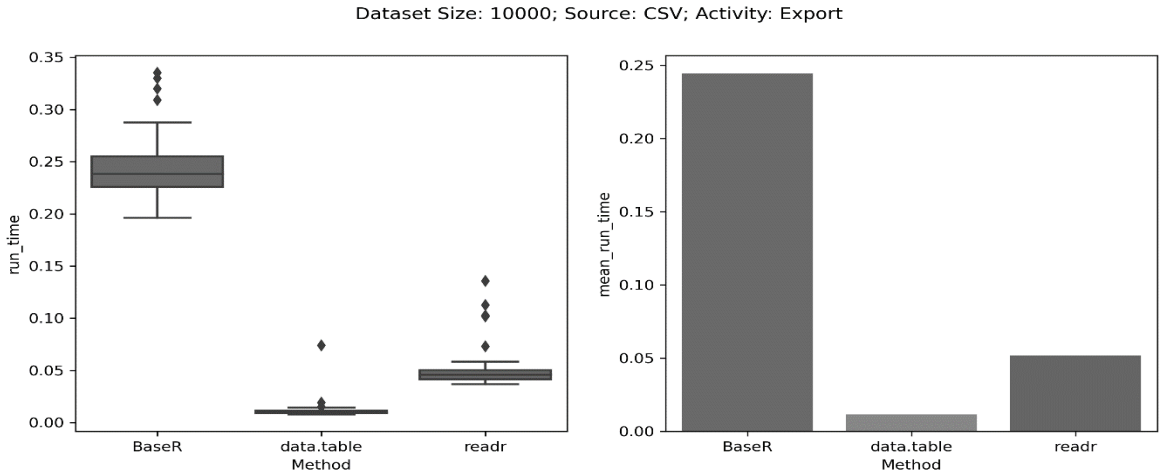

Fig 5.4.2.3 – Visual Analysis of Cost data (Export) for sample size 10000 for CSV data source

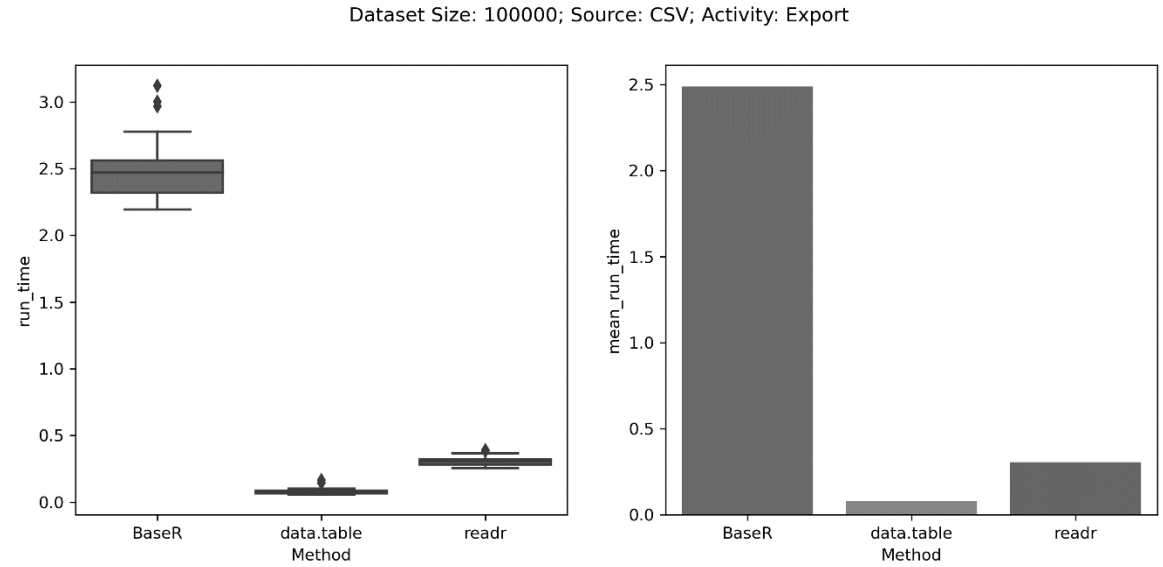

Fig 5.4.2.4 – Visual Analysis of Cost data (Export) for sample size 100000 for CSV data source

5.4.3 Inference Analysis of Cost Data

Analysis of variance was performed to test the significance of difference between the mean run time for all three frameworks (Base R (write.csv); readr; data.table.

Table 5.3.3.1: Analysis of Variance Results for different data size export to CSV format

Dataset Size (Rows)	100	1,000	10,000	100,000
F-Statistics	3.044114	37.54203	1616.649	6389.696
p-value	0.050653	6.74E-14	8.31E-101	1.27E-143

The analysis of variance shows that for sizes of all datasets (excluding 100 Rows) the difference in performance of packages (Base R (write.csv); readr; data.table) has a significant difference (p-value <0.05). for 100 records, Base R and data.table shows almost same performance and shows insignificant performance difference.

© 2024 NSP
Natural Sciences Publishing Cor.

6. Conclusion

The analysis of data exchange cost (import or export time) data concludes that for importing small excel openxlsx has superior performance while on larger dataset readxl perform far better than openxlsx. The recommendation for reading Excel files goes with reaadxl. While, for writing the Excel files, there are two aspects. First, if the objective of data export is to save the processed or raw data, writexl shows significant superiority over openxlsx but if the objective of export is to publish neat good looking excel files, openxlsx is a good choice. In second case, user need to compromise with export speed.
On the other hand, if we are examining results obtained from flat files data exchange cost. For import and export of the flat files data.table can be considered as un-disputed framework. It shows its high-performance capabilities compared to its competitive frameworks.

Acknowledgments

Authors are thankful to the Amity University, Noida Campus, Uttar Pradesh and University of Lucknow, Uttar Pradesh for providing necessary infrastructural support to carry out the present research work. Authors are also thankful to the honorable reviewers for their valuable suggestions which improved the quality of the manuscript.

References

[1] https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/?sh=17600f5d6f63.
[2] David Smith. *R Tops Data Mining Software Poll*, R-bloggers, (2012).
[3] Karl Rexer, Heather Allen, and Paul Gearan. *Data Miner Survey Summary*, presented at Predictive Analytics World. (2011).
[4] Robert A. Muenchen. *The Popularity of Data Analysis Software*, (2012).
[5] Tippmann, Sylvia. *Programming tools: Adventures with R*. Nature, **517**, 109–110, (2014).
[6] R Core Team. *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/, (2022).
[7] Wickham H, Hester J, Bryan J. *readr: Read Rectangular Text Data*. https://readr.tidyverse.org, https://github.com/tidyverse/readr, (2022).
[8] Dowle M, Srinivasan A._*data.table: Extension of `data.frame`*_. R package version 1.14.2, https://CRAN.R-project.org/package=data.table, (2021).
[9] Schauberger P, Walker A. _*openxlsx: Read, Write and Edit xlsx Files*_. R package version 4.2.5, https://CRAN.R-project.org/package=openxlsx, (2021).
[10] Ooms J. _*writexl: Export Data Frames to Excel 'xlsx' Format*_. R package version 1.4.0, https://CRAN.R-project.org/package=writexl, (2021).
[11] Wickham H, Bryan J. _readxl: Read Excel Files_. R package version 1.4.1, https://CRAN.R-project.org/package=readxl, (2022).

© 2024 NSP
Natural Sciences Publishing Cor.