Journal of Statistics Applications & Probability

Volume 13
Issue 2 *Mar. 2024*

Article 19

3-1-2024

Memory Utilization in R: The Impact of Data Management Frameworks (Packages)

Anant P. Awasthi
Department of Statistics, Amity University, Gautam Buddha Nagar, Uttar Pradesh, India, anant.awasthi@outlook.com

Niraj K . Singh
Department of Statistics, Amity University, Gautam Buddha Nagar, Uttar Pradesh, India, anant.awasthi@outlook.com

Masood H. Siddiqui
Department of Statistics, University of Lucknow, Lucknow, Uttar Pradesh, India, anant.awasthi@outlook.com

Aanchal A. Awasthi
Department of Statistics, Amity University, Gautam Buddha Nagar, Uttar Pradesh, India, anant.awasthi@outlook.com

Recommended Citation
P. Awasthi, Anant; K . Singh, Niraj; H. Siddiqui, Masood; and A. Awasthi, Aanchal (2024) "Memory Utilization in R: The Impact of Data Management Frameworks (Packages)," *Journal of Statistics Applications & Probability*: Vol. 13: Iss. 2, Article 19.
DOI: https://dx.doi.org/10.18576/jsap/130217
Available at: https://digitalcommons.aaru.edu.jo/jsap/vol13/iss2/19

Journal of Statistics Applications & Probability
An International Journal

http://dx.doi.org/10.18576/jsap/130217

Memory Utilization in R: The Impact of Data Management Frameworks (Packages)

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

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

Received: 26 Jul. 2023, Revised: 22 Dec. 2023, Accepted: 23 Dec. 2023.
Published online: 1 Mar. 2024

Abstract: Memory management is a very important component of running large workloads in computing. It takes into account the capacity & limitations of the available memory on the device itself and deallocates memory space when it is not needed or expands the space through virtual memory. Memory management strives to optimize memory usage so that the CPU can efficiently access the instructions and data it needs to execute the various processes.

This work is focused on memory utilization by different data management frameworks in R on different platforms. We have considered Native R, tidyverse, and data.table as data management frameworks. Very high precision (1e-5+1 to 1e-6+1) visual analysis of memory utilization data shows Native R memory management is better when compared to tidyverse and data.table. But, when we are analysing data on a large scale and observe the memory utilization, it shows no significant difference in distribution of memory utilization across different sample sizes. We have established the same results using analysis of variance (ANOVA) and analysis of coefficients of linear regression models.

Keywords: R; tidyverse; data.table; Memory utilization in R

1. Introduction

Memory management is a critical aspect of any programming language, and R is no exception. In R, R performs data storage and manipulation primarily in Random Access Memory (RAM)[1, 4]. When you load a dataset into R, it is stored in RAM, and all subsequent operations and computations are performed in RAM. This is because RAM provides fast access to data, allowing for efficient computations and data manipulation.
However, it's important to note that the size of the dataset and the amount of RAM available on your computer can affect the performance of R. If the dataset is too large to fit into RAM, R may start to use virtual memory (i.e., the hard disk), which can result in slower performance due to the slower read/write speeds of the hard disk compared to RAM. In extreme cases, running out of memory can also cause R to crash.
In this study, we are exploring the effectiveness of three major data frameworks (Native R dataframe, Tidyverse Tibble dataframe and data.table dataframe) in R in terms of storage as a cost. We ran experiments on datasets of different sample sizes and evaluated the cost (storage) based on different data structures. This study is performed at very high precision (wherever needed) where we tried to retain the data at highest possible precision so that we can differentiate the cost at each step.

2. Memory Allocation & Utilization in R

In the R programming language, memory management is handled automatically by the R interpreter[1]. R utilizes a garbage collector to manage memory allocation and deal location, making it a high-level language that abstracts low-level memory management details from the programmer.
R imposes memory limits based on the available system resources. If your program exceeds these limits, you may encounter errors or slowdowns. While R handles memory management automatically, it's still good practice to avoid unnecessary memory usage

3. Frameworks (Packages) in Scope

1. Native R (utils)[1];
2. tidyverse_1.3.2/tibble_3.1.8[2];
3. data.table_1.14.2 [3]

4. Methodology

There was no significant work was found in the literature for benchmarking of memory utilization of R frameworks. We have

*Corresponding author e-mail: *anant.awasthi@outlook.com* discussed here approach to record the memory utilization of frameworks along with analysis approach which includes graphical exploration to establish the benchmark and statistical inference & use of supervise machine learning technique to support the evidences.

An initial sample (0.2 million rows) has been drawn randomly with replacement from flight data from nycflight13 package and changed the data structure to data structure in scope. Memory utilization was measured and results were written to a file. In the next step, the sample size has been step up by 0.2 million rows and memory utilization was recorded. The same procedure repeated till the sample size reached 100 million records.

4.1 Illustration of Experiment control Flow:

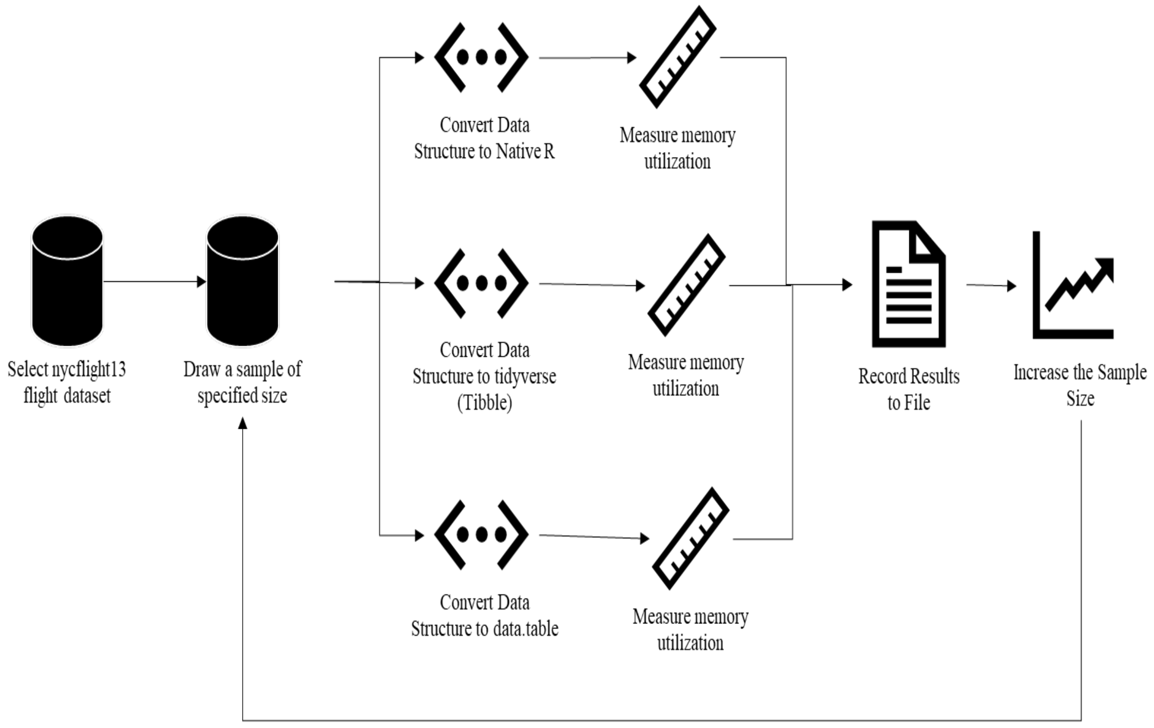

Select nycflight13 flight dataset
Draw a sample of specified size
Convert Data Structure to Native R
Measure memory utilization
Convert Data Structure to tidyverse (Tibble)
Measure memory utilization
Convert Data Structure to data.table
Measure memory utilization
Record Results to File
Increase the Sample Size

***Fig. 4.1.1** – Control flow of experiments*

4.2 Methods:

The result was divided into 25 samples (20 run each) and each sample was further analyzed with highest possible precision on experiment machine.

Analysis of Variance (ANOVA) [7, 8] is a statistical technique used to analyze the mean difference of memory utilization between the groups (Frameworks). ANOVA allows to determine whether the means of two or more groups are significantly different from each other. Analysis of variance is used to establish the fact whether three data structure in scope have significant difference in memory utilization or not for at least one set of frameworks.

Analysis of variance has been performed with null hypothesis that there is no significant difference in memory utilization of three frameworks (Native R, Tidyverse and data.table).

H_0: There is no significant difference in memory utilization of three frameworks (Native R, Tidyverse and data.table)

Linear regression [9,10,11] is a statistical technique used to model and analyze the relationship between a dependent variable (also called the response variable, memory utilization in this study) and one or more independent variables (also called predictor variables, sample size in this study). It aims to find the best-fitting linear equation that describes the linear relationship between the variables [12, 13]. Further, A linear regression model was built to estimate the impact of sample size (X) and utilization (Y). Higher Coefficient of sample size (X) indicates high memory utilization (Y) for unit increase of sample size (X).

5. Analysis and Results:

Linear regression models were built to estimate the effect of sample size (X) on memory utilization (Y) for all three frameworks and coefficients of sample size (X) were analyzed at very high precision (10 digit after decimals). We have considered below mentioned regression equation for analysis of impact on Y of coefficient (β) for a unit change in sample size (X).

memory utilization = β* sample size + intercept

6. Results – Analysis of memory utilization data:

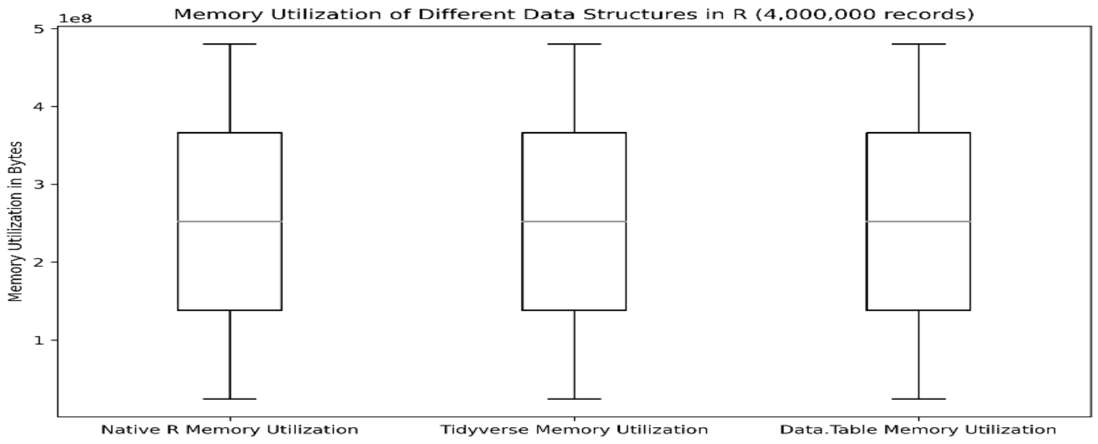
Table 6.1 – Analysis of Memory Utilization using ANOVA and Linear Regression

Sample Number	Sample Size Minimum	Sample Size Maximum	ANOVA Results		Native R Linear Regression Results		Tidyverse Linear Regression Results		data.table Linear Regression Results	
			F-Statistics	P-Value	Coef. of x	P-Value	Coef. of x	P-Value	Coef. of x	P-Value
1	2,00,000	40,00,000	0.0000000008	0.9999999992	120.0009789474	4.89243E-91	120.0009789474	4.89243E-91	120.0009789474	4.89243E-91
2	42,00,000	80,00,000	0.0000000008	0.9999999992	120.0004629474	7.81779E-98	120.0004629474	7.81779E-98	120.0004629474	7.81779E-98
3	82,00,000	1,20,00,000	0.0000000008	0.9999999992	120.0002536842	9.6141E-103	120.0002536842	9.6141E-103	120.0002536842	9.6141E-103
4	1,22,00,000	1,60,00,000	0.0000000008	0.9999999992	120.0001517895	7.0259E-106	120.0001517895	7.0259E-106	120.0001517895	7.0259E-106
5	1,62,00,000	2,00,00,000	0.0000000008	0.9999999992	120.0000770526	3.0703E-111	120.0000770526	3.0703E-111	120.0000770526	3.0703E-111
6	2,02,00,000	2,40,00,000	0.0000000008	0.9999999992	120.0000376842	3.0418E-118	120.0000376842	3.0418E-118	120.0000376842	3.0418E-118
7	2,42,00,000	2,80,00,000	0.0000000008	0.9999999992	120.0000246316	2.049E-120	120.0000246316	2.049E-120	120.0000246316	2.049E-120
8	2,82,00,000	3,20,00,000	0.0000000008	0.9999999992	120.0000187368	9.1837E-121	120.0000187368	9.1837E-121	120.0000187368	9.1837E-121
9	3,22,00,000	3,60,00,000	0.0000000008	0.9999999992	120.0000075789	3.5497E-127	120.0000075789	3.5497E-127	120.0000075789	3.5497E-127
10	3,62,00,000	4,00,00,000	0.0000000008	0.9999999992	120.0000040000	4.8245E-129	120.0000040000	4.8245E-129	120.0000040000	4.8245E-129
11	4,02,00,000	4,40,00,000	0.0000000008	0.9999999992	120.0000000000	1.6258E-271	120.0000000000	1.6258E-271	120.0000000000	1.6258E-271
12	4,42,00,000	4,80,00,000	0.0000000008	0.9999999992	120.0000000000	1.341E-271	120.0000000000	1.341E-271	120.0000000000	1.341E-271
13	4,82,00,000	5,20,00,000	0.0000000008	0.9999999992	120.0000000000	9.4215E-281	120.0000000000	9.4215E-281	120.0000000000	9.4215E-281
14	5,22,00,000	5,60,00,000	0.0000000008	0.9999999992	120.0000000000	1.2645E-272	120.0000000000	1.2645E-272	120.0000000000	1.2645E-272
15	5,62,00,000	6,00,00,000	0.0000000008	0.9999999992	120.0000000000	2.954E-265	120.0000000000	2.954E-265	120.0000000000	2.954E-265
16	6,02,00,000	6,40,00,000	0.0000000008	0.9999999992	120.0000000000	1.0384E-264	120.0000000000	1.0384E-264	120.0000000000	1.0384E-264
17	6,42,00,000	6,80,00,000	0.0000000008	0.9999999992	120.0000000000	2.4698E-284	120.0000000000	2.4698E-284	120.0000000000	2.4698E-284
18	6,82,00,000	7,20,00,000	0.0000000008	0.9999999992	120.0000000000	1.2507E-265	120.0000000000	1.2507E-265	120.0000000000	1.2507E-265
19	7,22,00,000	7,60,00,000	0.0000000008	0.9999999992	120.0000000000	8.0856E-270	120.0000000000	8.0856E-270	120.0000000000	8.0856E-270
20	7,62,00,000	8,00,00,000	0.0000000008	0.9999999992	120.0000000000	1.5266E-269	120.0000000000	1.5266E-269	120.0000000000	1.5266E-269
21	8,02,00,000	8,40,00,000	0.0000000008	0.9999999992	120.0000000000	1.4251E-265	120.0000000000	1.4251E-265	120.0000000000	1.4251E-265
22	8,42,00,000	8,80,00,000	0.0000000008	0.9999999992	120.0000000000	1.2645E-272	120.0000000000	1.2645E-272	120.0000000000	1.2645E-272
23	8,82,00,000	9,20,00,000	0.0000000008	0.9999999992	120.0000000000	2.4698E-266	120.0000000000	2.4698E-266	120.0000000000	2.4698E-266
24	9,22,00,000	9,60,00,000	0.0000000008	0.9999999992	120.0000000000	4.8991E-273	120.0000000000	4.8991E-273	120.0000000000	4.8991E-273
25	9,62,00,000	10,00,00,000	0.0000000008	0.9999999992	120.0000000000	7.5713E-263	120.0000000000	7.5713E-263	120.0000000000	7.5713E-263

Detailed high precision analysis of individual sample validates the similar behaviour across the datasets. It evident that we don't have any significant difference when we are analysing behaviour of these frameworks over a range of sample sizes (ranging from $2x10^5$ records to 10^8 records).

This can be observe that at a very high precision (1e-5+1 to 1e-6+1) visualization of memory utilization data, as sample size increases the memory utilization become almost similar in case of Native R and Tidyverse but memory utilization of data.table a bit higher (Please refer data visualization for memory utilization below).

6.1 Sample 1 (2x10^5 records – 4x10^6 records):



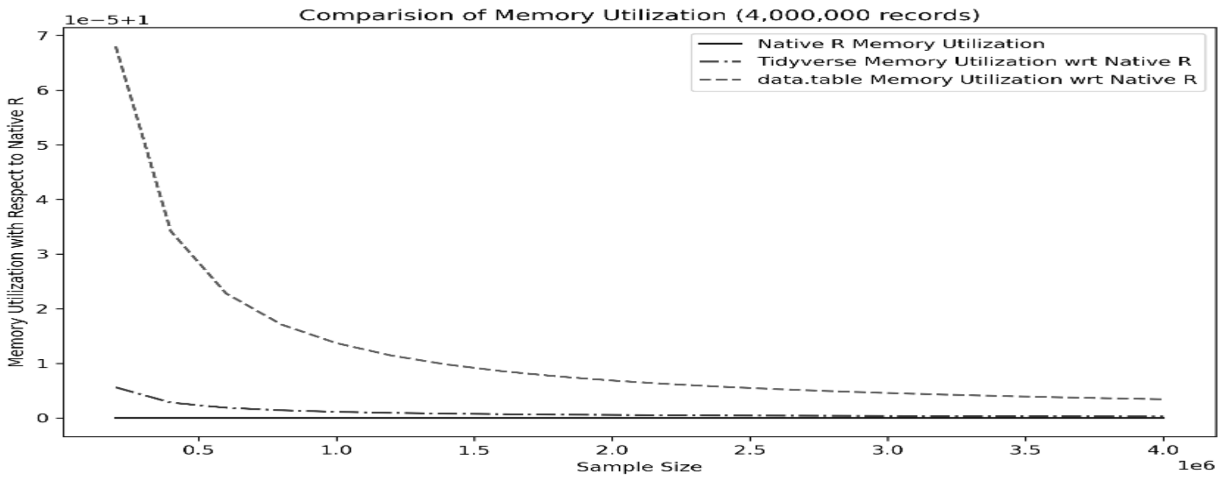
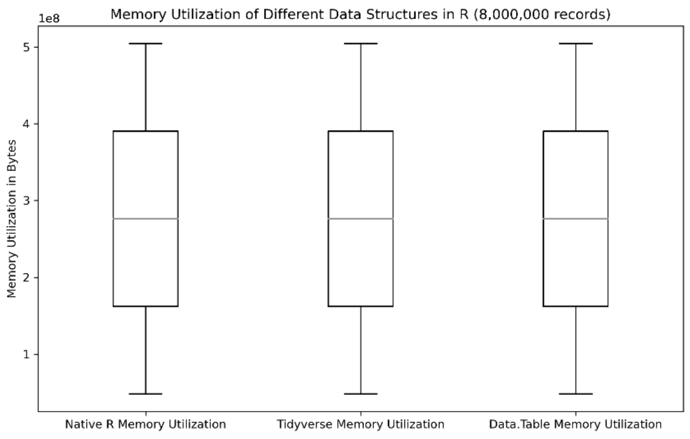


Fig.6.1.1 *– Visual Analysis of Memory Utilization at level of 4 million records*

6.2 Sample 2 (4.2x10^6 records – 8x10^6 records):



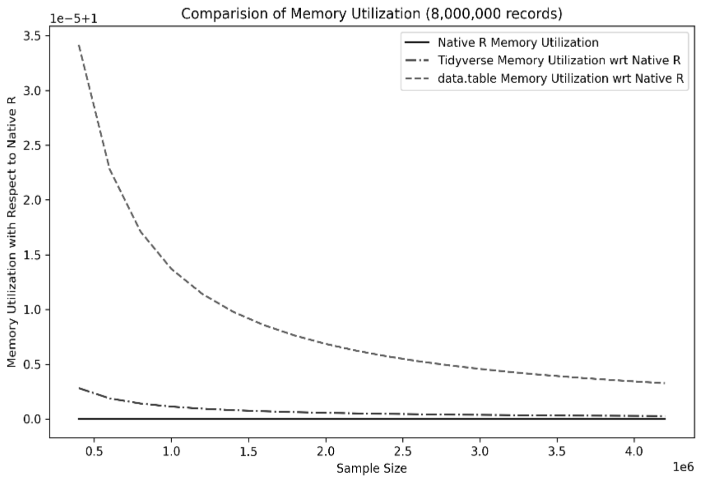
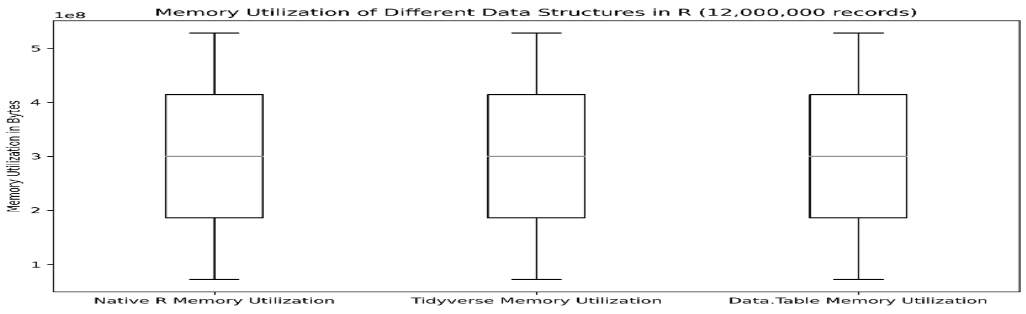


Fig.6.2.1 *– Visual Analysis of Memory Utilization at level of 8 million records*

6.3 Sample 3 ($8.2x10^6$ records – $1.2x10^7$ records):



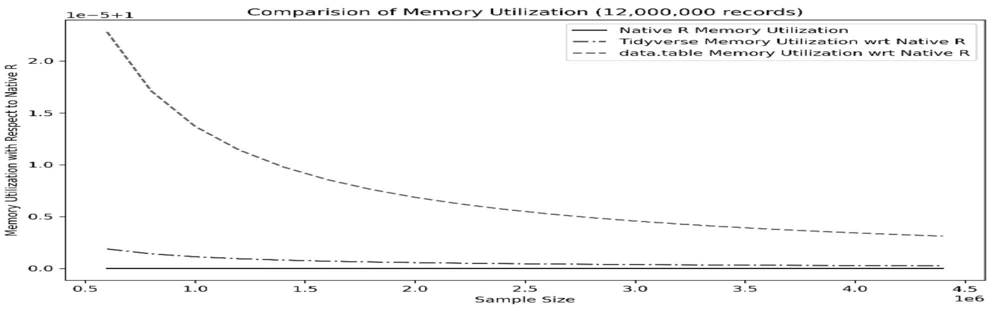
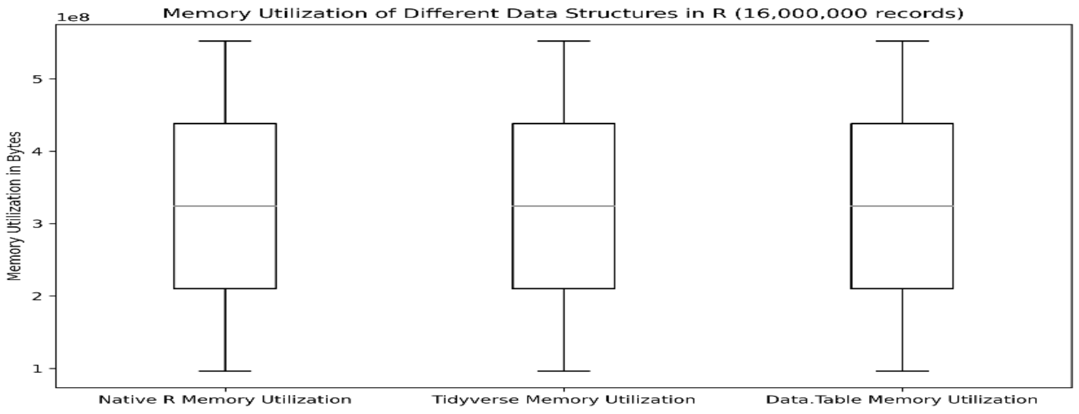


Fig. 6.3.1 – Visual Analysis of Memory Utilization at level of 12 million records

6.4 Sample 4 ($1.22x10^7$ records – $1.6x10^7$ records):



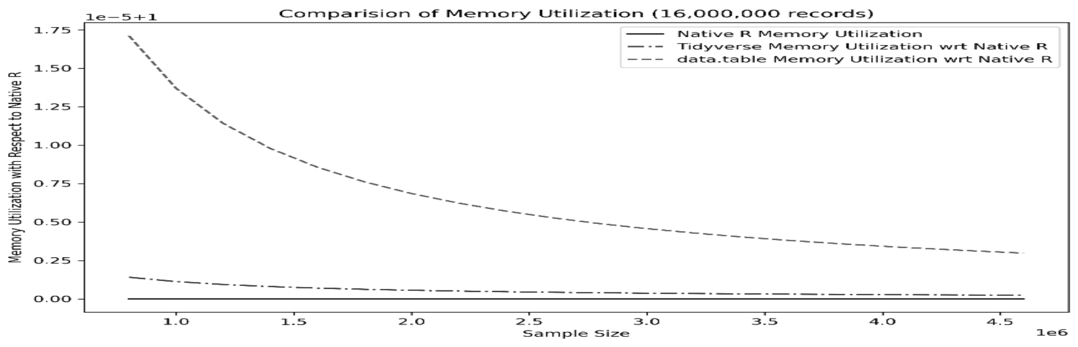
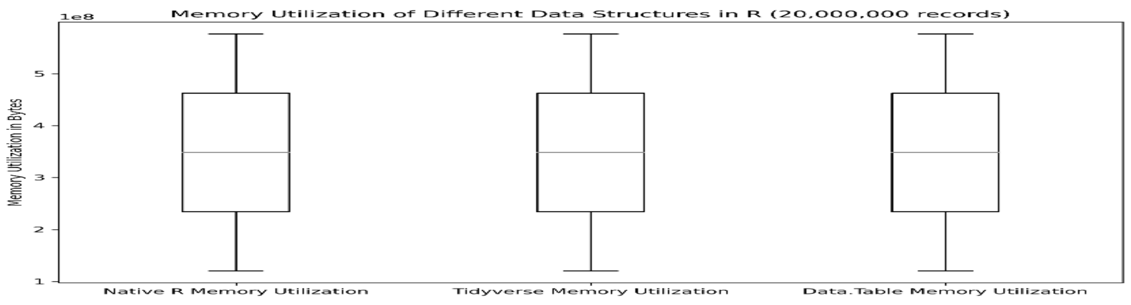
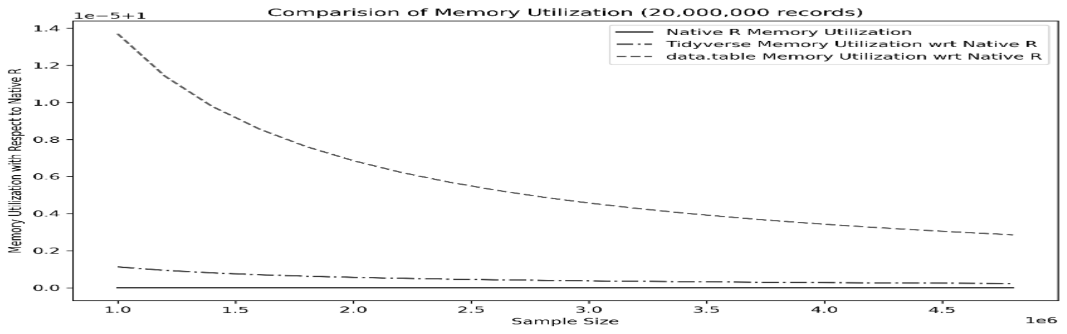


Fig. 6.4.1 *– Visual Analysis of Memory Utilization at level of 16 million records*

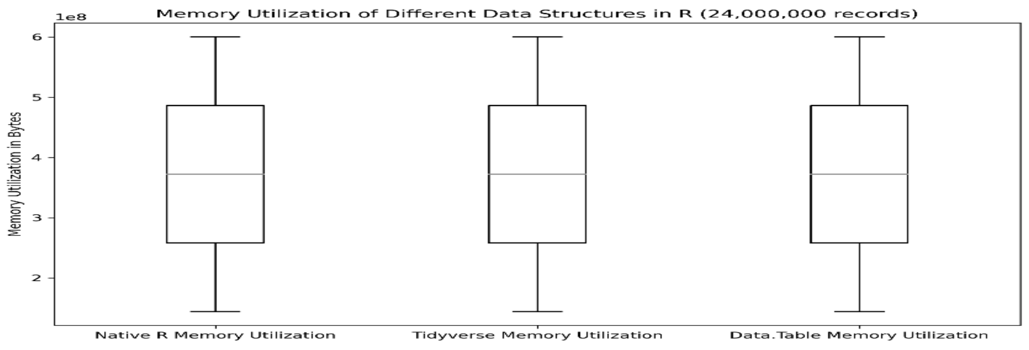
6.5 Sample 5 (1.62x10^7 records – 2x10^7 records):





Figu. 6.5.1 *– Visual Analysis of Memory Utilization at level of 20 million records*

6.6 Sample 6 (2.02x10^7 records – 2.4x10^7 records):



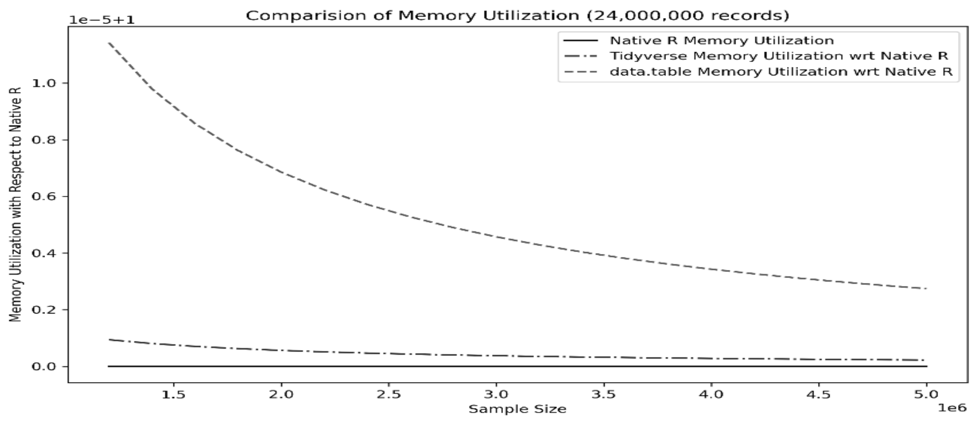
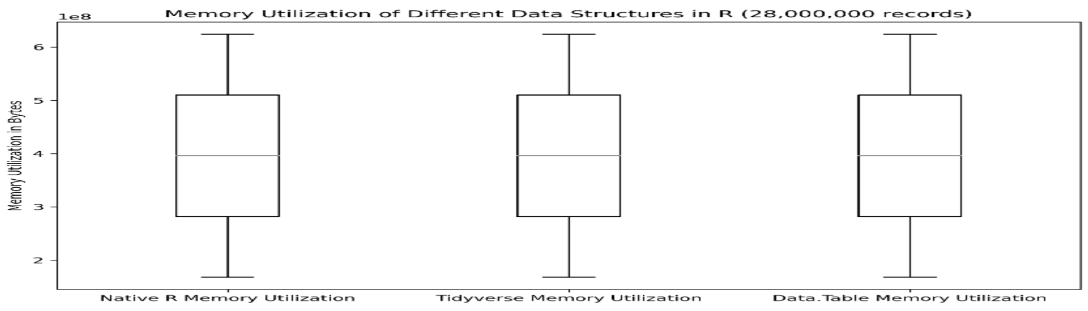


***Fig.6.6.1** – Visual Analysis of Memory Utilization at level of 24 million records*

6.7 Sample 7 (2.42x10^7 records – 2.8x10^7 records):



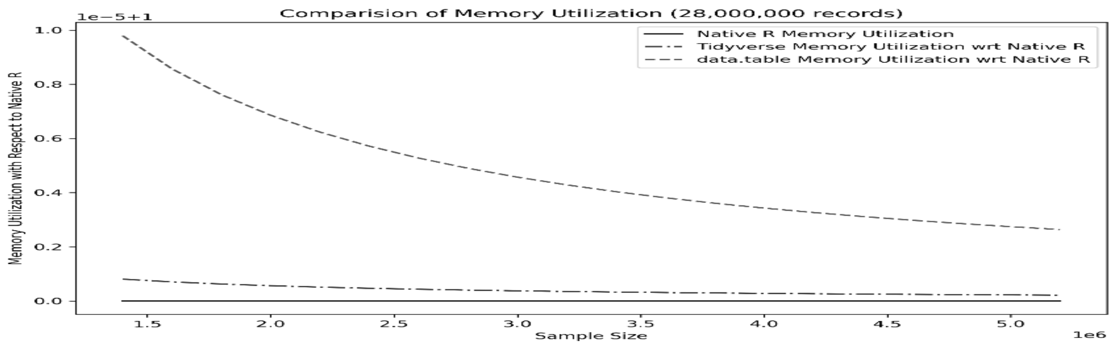
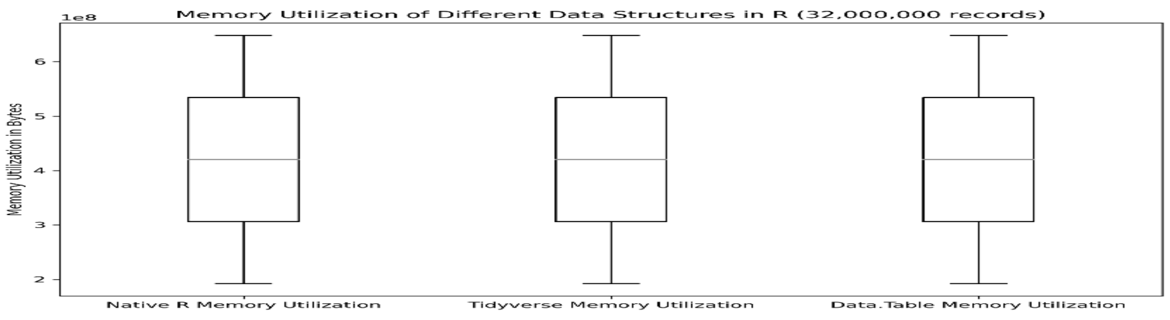


***Fig.6.7.1** – Visual Analysis of Memory Utilization at level of 28 million records*

6.8 Sample 8 (2.82x10^7 records – 3.2x10^7 records):



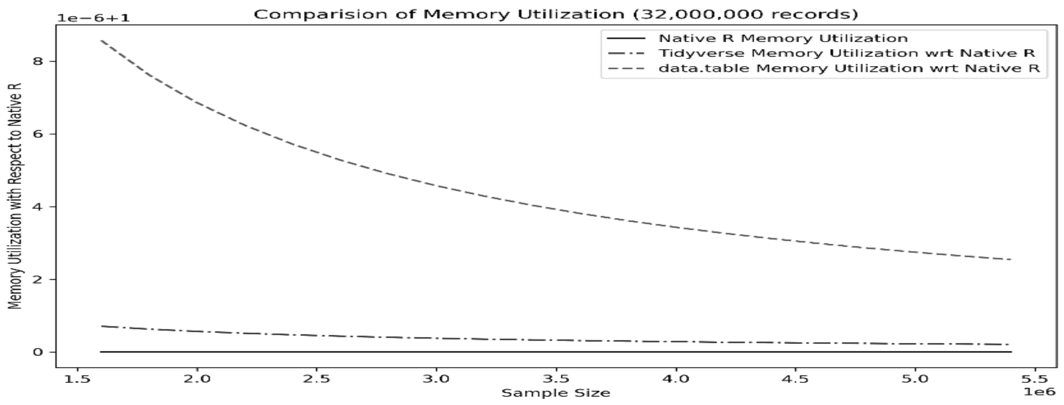
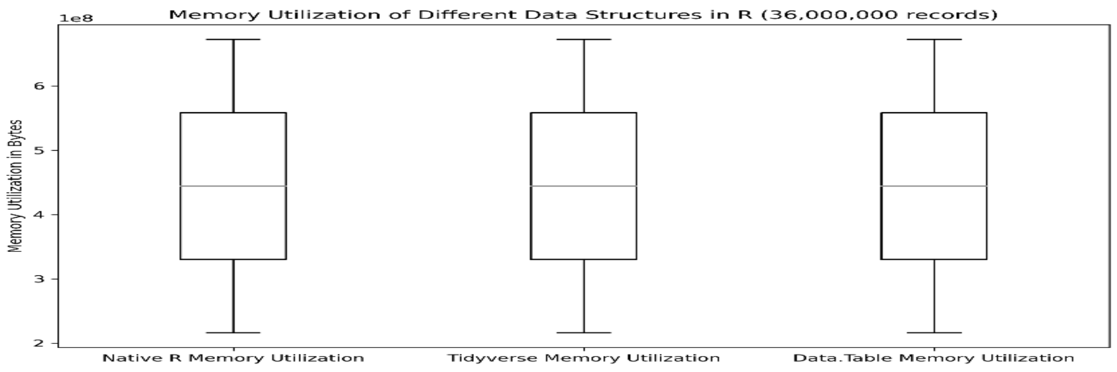


Fig. 6.8.1 – *Visual Analysis of Memory Utilization at level of 32 million records*

6.9 Sample 9 (3.22x10^7 records – 3.6x10^7 records):



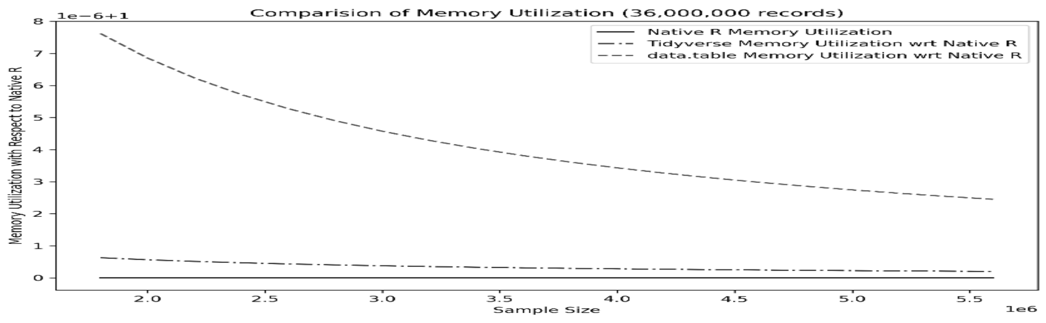
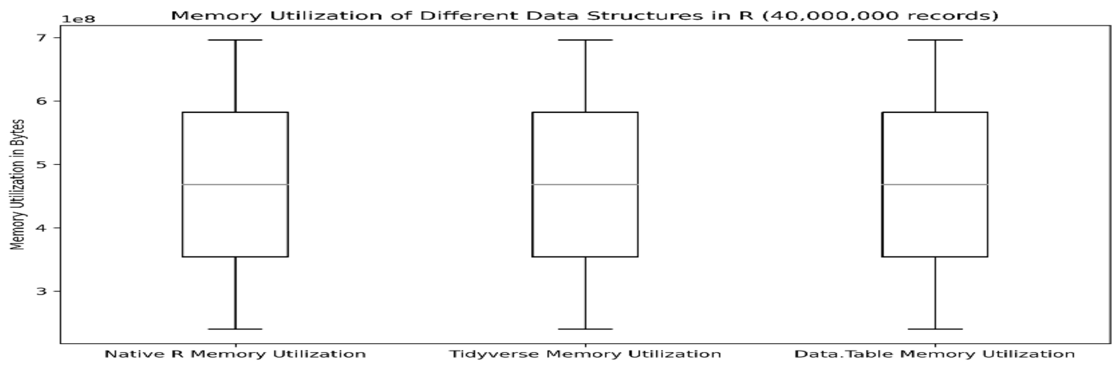


Fig. 6.9.1 – *Visual Analysis of Memory Utilization at level of 36 million records*

6.10 Sample 10 ($3.62x10^7$ records – $4x10^7$ records):



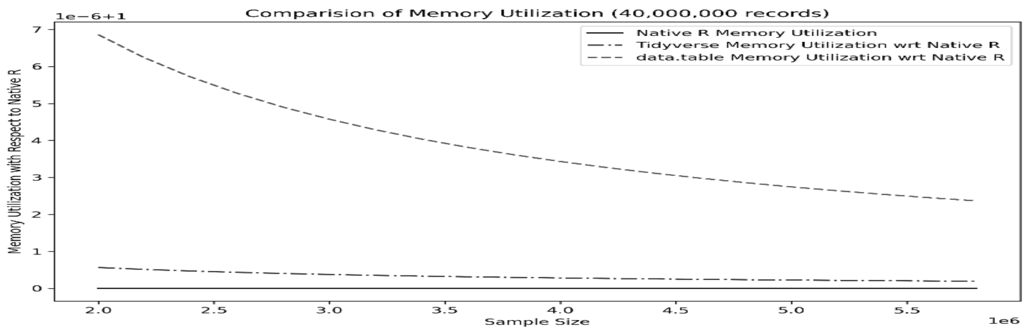
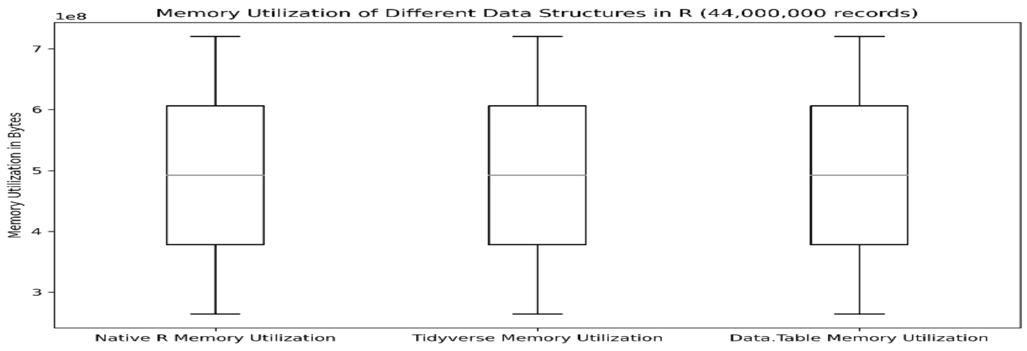


***Fig. 6.10.1** – Visual Analysis of Memory Utilization at level of 40 million records*

6.11 Sample 11 ($4.02x10^7$ records – $4.4x10^7$ records):



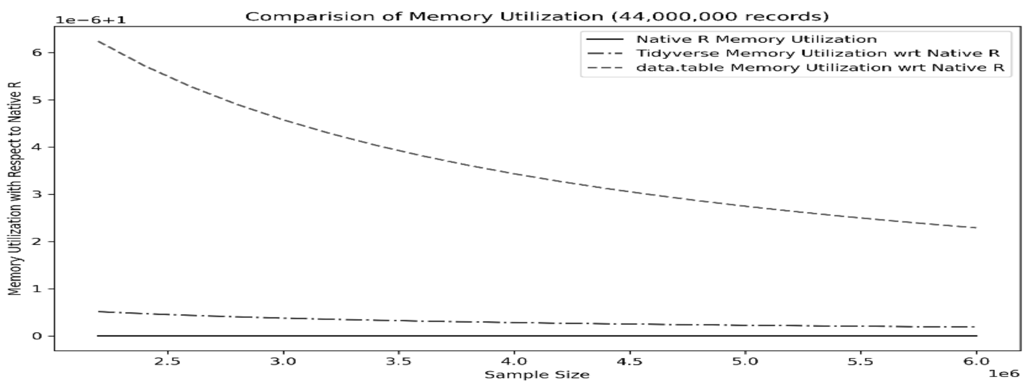
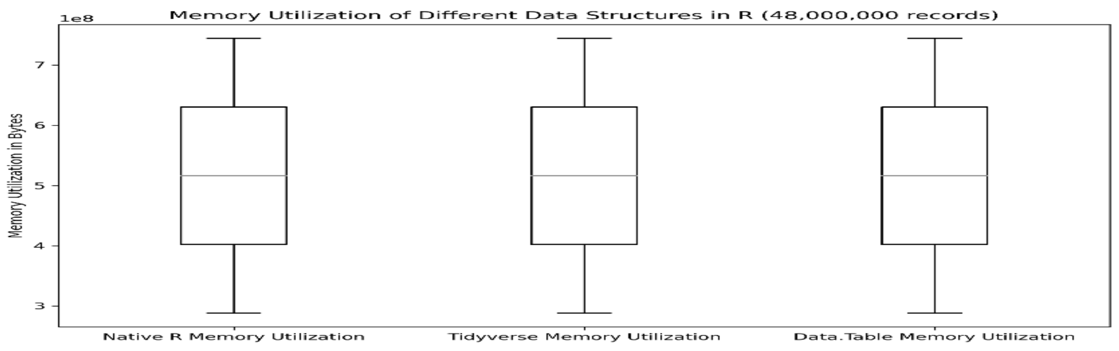


Fig.6.11.1 *– Visual Analysis of Memory Utilization at level of 44 million records*

6.12 Sample 12 (4.42x10^7 records – 4.8x10^7 records):



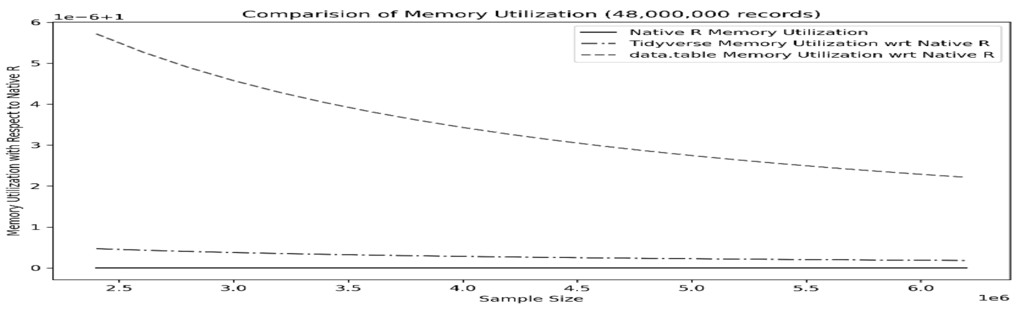
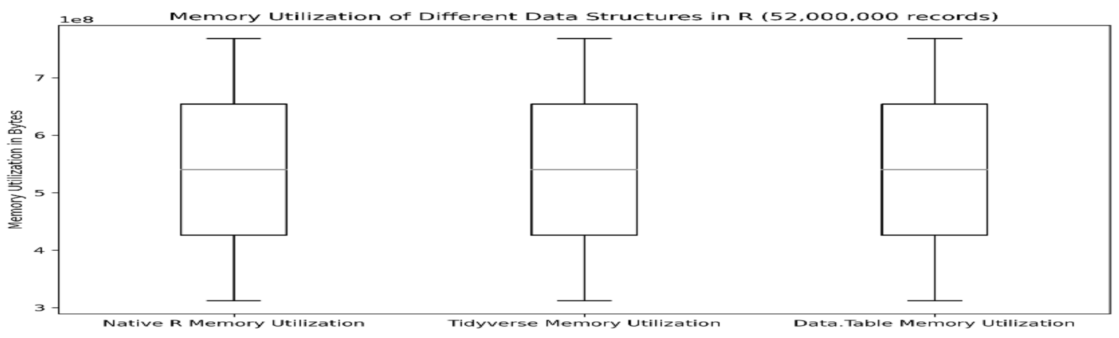


Fig.6.12.1 *– Visual Analysis of Memory Utilization at level of 48 million records*

6.13 Sample 13 (4.82x10^7 records – 5.2x10^7 records):



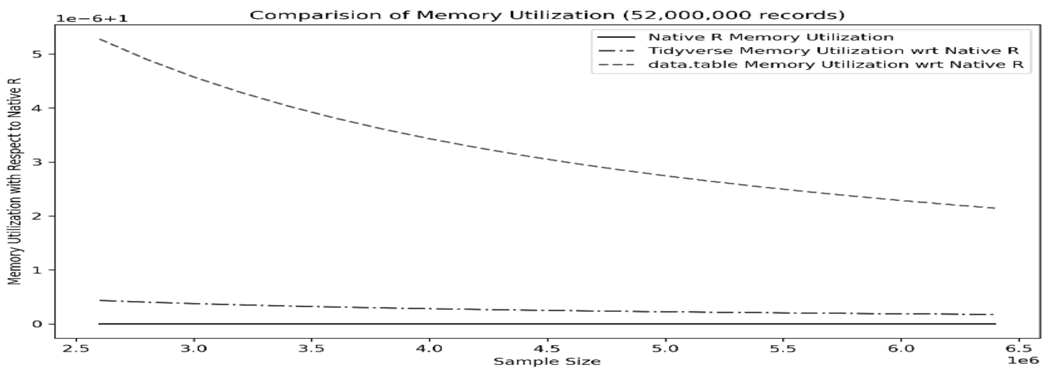
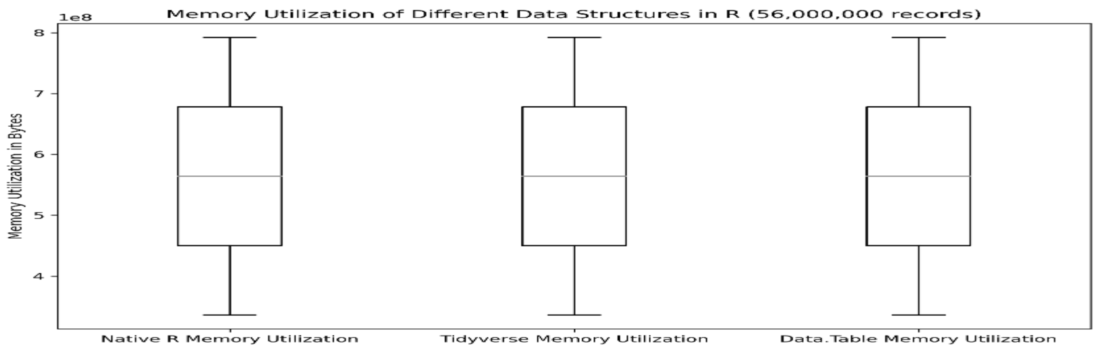


Fig.6.13.1 *– Visual Analysis of Memory Utilization at level of 52 million records*

6.14 Sample 14 (5.22x10^7 records – 5.6x10^7 records):



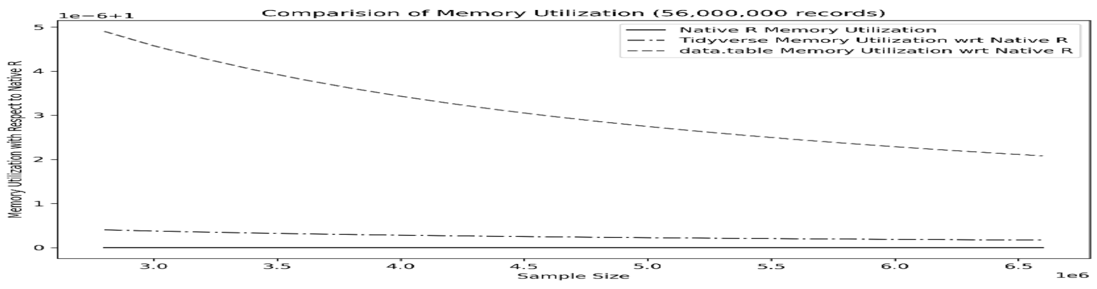
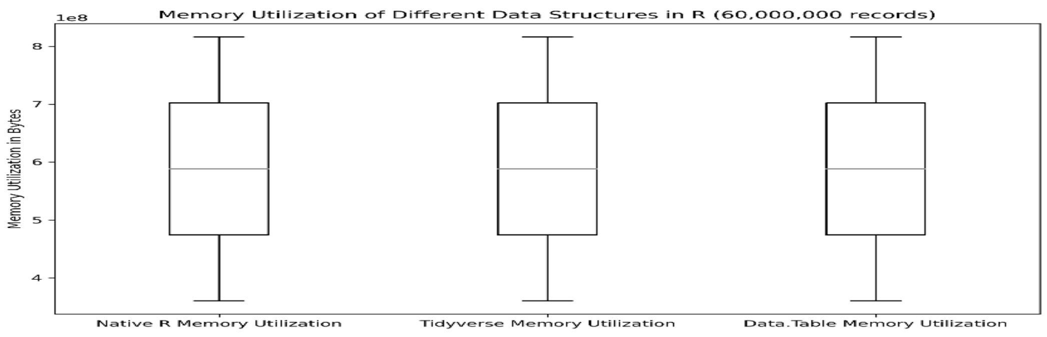


Fig. 6.14.1 *– Visual Analysis of Memory Utilization at level of 56 million records*

6.15 Sample 15 (5.62x10^7 records –6x10^7 records):



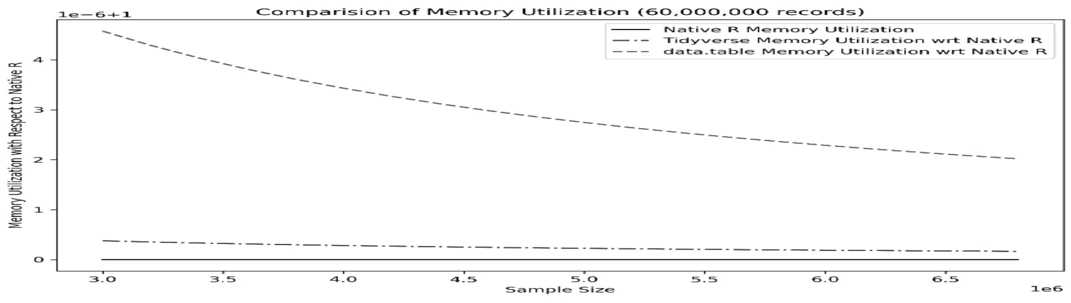
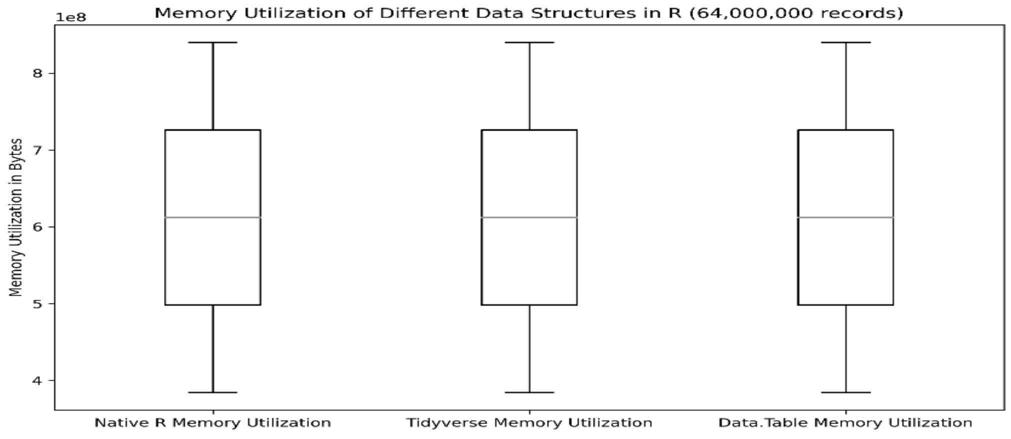


***Fig.6.15.1** – Visual Analysis of Memory Utilization at level of 60 million records*

6.16 Sample 16 (6.2x10^7 records – 6.4x10^7 records):



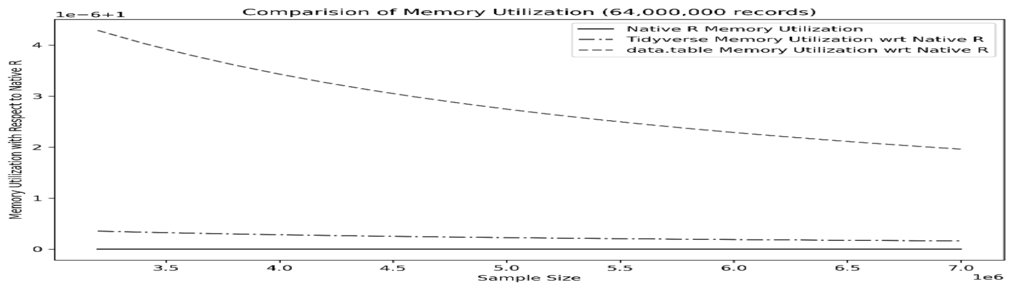
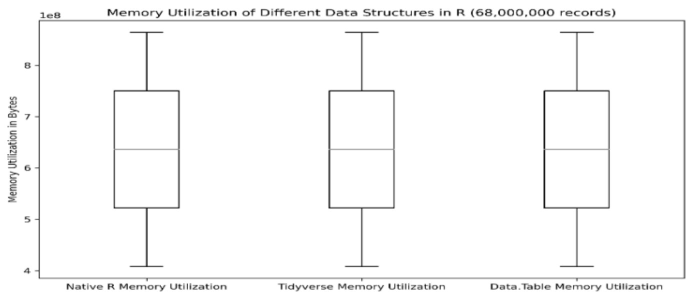


***Figure 6.16.1** – Visual Analysis of Memory Utilization at level of 64 million records*

6.17 Sample 17 (6.42x10^7 records – 6.8x10^7 records):



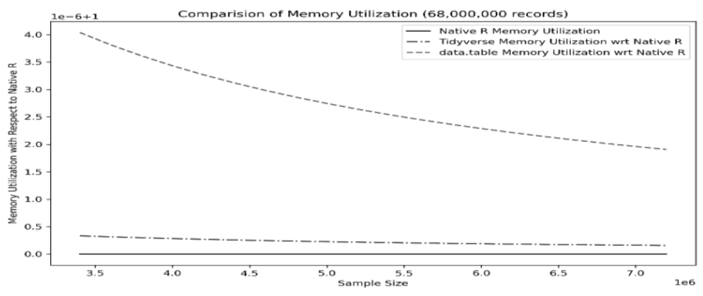
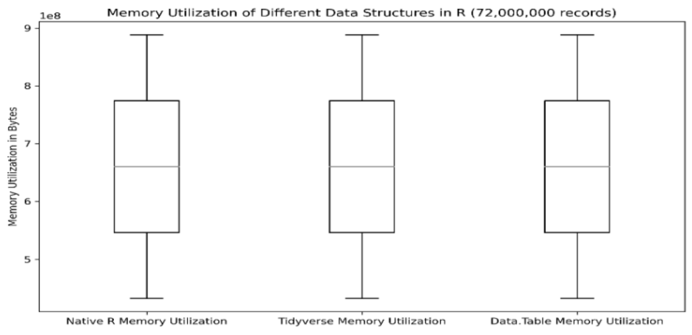


***Fig.6.17.1** – Visual Analysis of Memory Utilization at level of 68 million records*

6.18 *Sample 18 (6.82x10^7 records – 7.2x10^7 records)*:



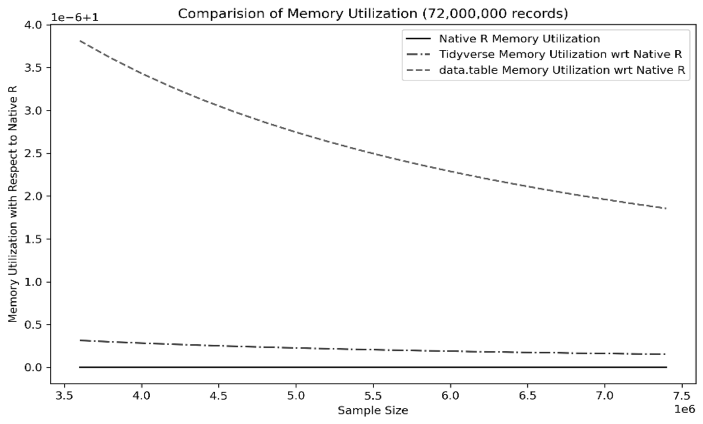
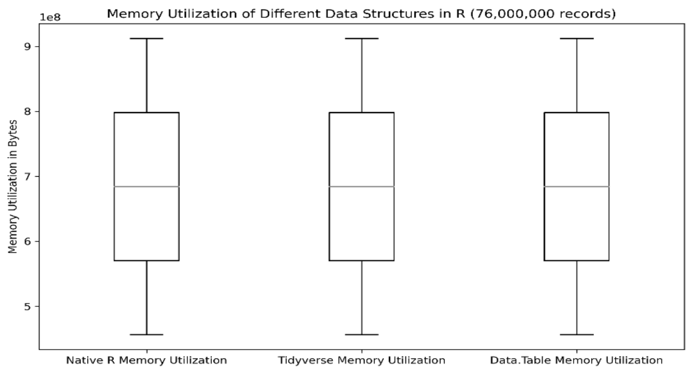


***Fig.6.18.1** – Visual Analysis of Memory Utilization at level of 72 million records*

6.19 *Sample 19 (7.22x10^7 records – 7.6x10^7 records)*:



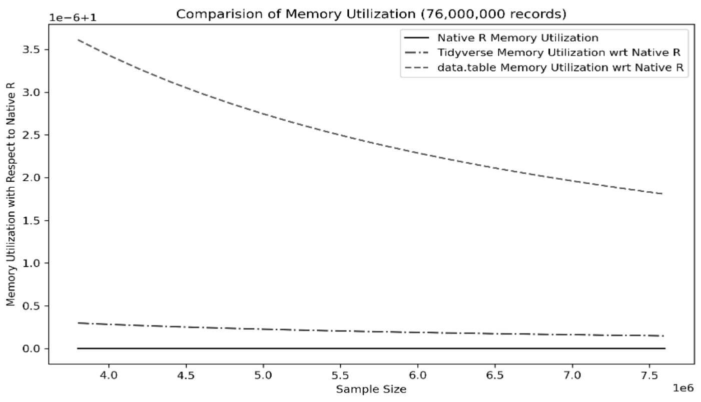
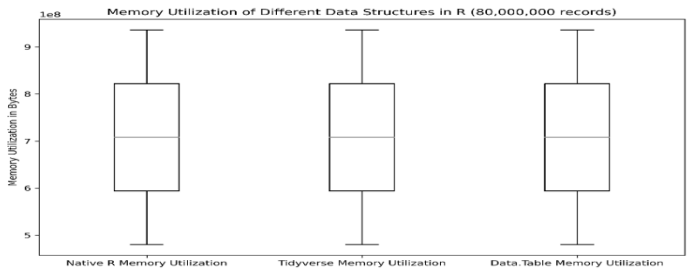


Fig.6.19.1 – Visual Analysis of Memory Utilization at level of 76 million records

6.20 Sample 20 (7.62x10^7 records –8x10^7 records):



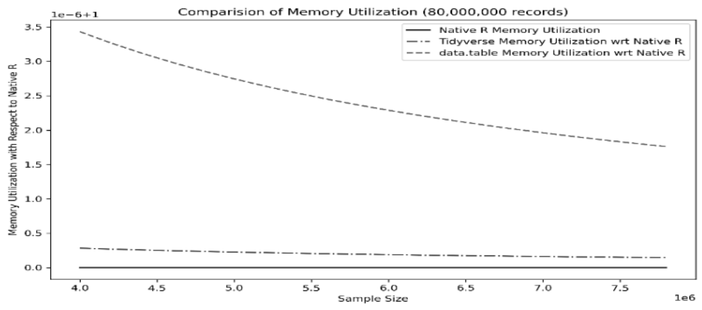
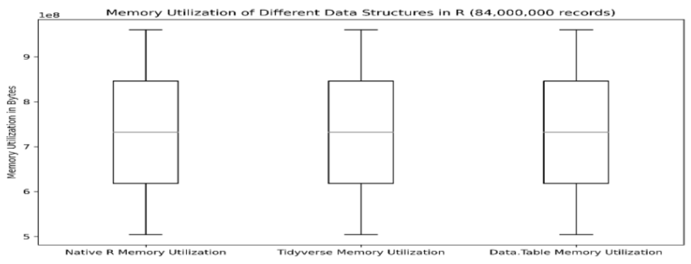


Fig.6.20.1 – Visual Analysis of Memory Utilization at level of 80 million records

6.21 Sample 21 ($8.02x10^7$ records –$8.4x10^7$ records):



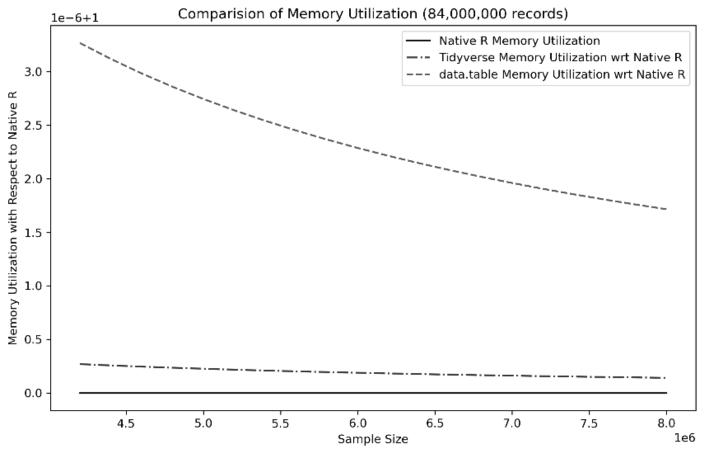
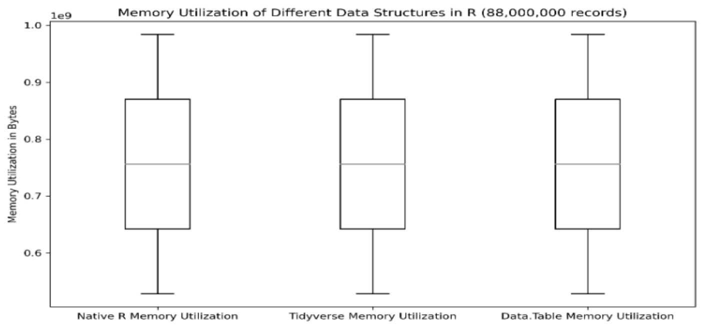


Fig. 6.21.1 – *Visual Analysis of Memory Utilization at level of 84 million records*

6.22 Sample 22 ($8.42x10^7$ records –$8.8x10^7$ records):



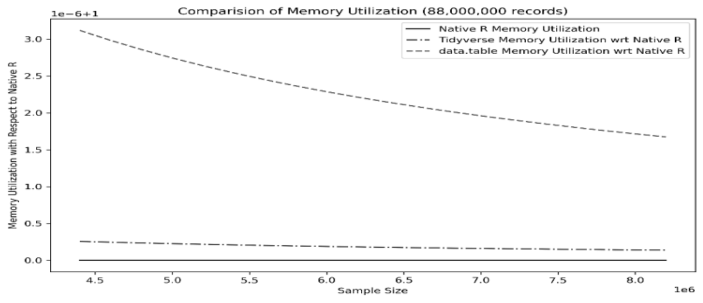


Fig. 6.22.1 – *Visual Analysis of Memory Utilization at level of 88 million records*

6.23 Sample 23 (8.82x10^7 records –9.2x10^7 records):

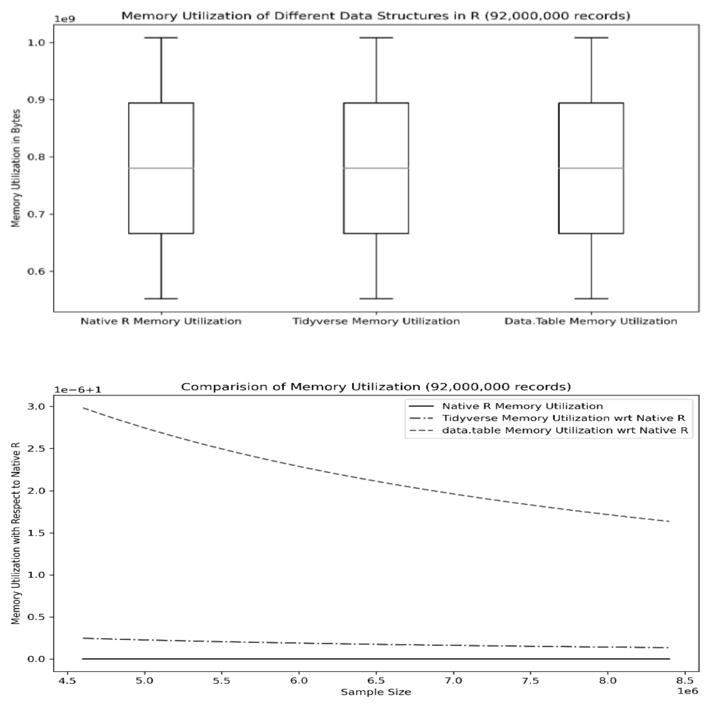


***Fig. 6.23.1** – Visual Analysis of Memory Utilization at level of 92 million records*

6.24 Sample 24 (9.22x10^7 records –9.6x10^7 records):

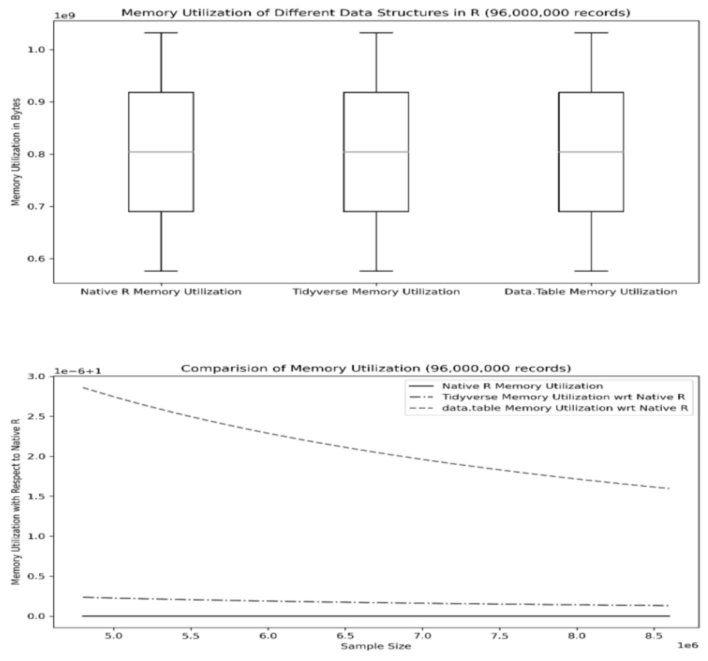


***Fig. 6.24.1** – Visual Analysis of Memory Utilization at level of 96 million records*

6.25 Sample 25 ($9.62x10^7$ records –$10x10^8$ records):

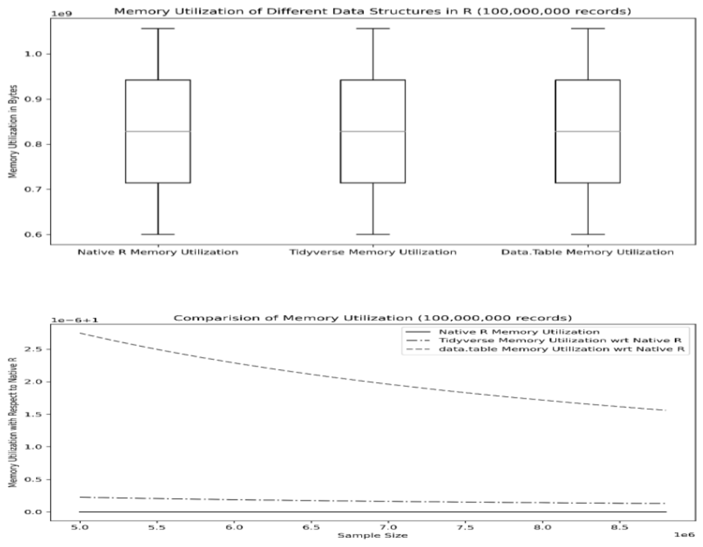


***Fig6.25.1** – Visual Analysis of Memory Utilization at level of 100 million records*

7. Conclusion and Discussion:

This was observed in all the samples that there is no significant difference (p-value ~ 1) in memory utilization of all three frameworks. But in high precision data visualization this was evident that native R dataframes are most efficient data structure followed by tidyverse and data.table. This difference become meaning less when the data volume is very high.

This was also observed that across the sample sizes that once sample size increases, the memory utilization of data.table comes closer to other data structures.

In the analysis, this was evident that the choice of framework doesn't make significant impact on memory utilization of the dataset. This also indicates that in the process of managing data in R at larger scale memory management has insignificant role and efficient data exchange and data manipulation strategies plays a key role.

The study was majorly focused on leading data management platforms (native R, Tidyverse & data.table) in R in the future more platforms can be added and the results can be re-evaluate even on large scale datasets.

8. Conflict of Interest: Authors do not have any conflict of interest as there is no external/internal funding used to complete this work.

References

[1] R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
[2] Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." _Journal of Open Source Software_, *4*(43), 1686. doi:10.21105/joss.01686 <https://doi.org/10.21105/joss.01686>.
[3] Dowle M, Srinivasan A (2021). _data.table: Extension of `data.frame`_. R package version 1.14.2, <https://CRAN.R-project.org/package=data.table>.
[4] R Core Team. (2021). object.size: Estimate the Size of R Objects (R version 4.1.0). R Foundation for Statistical Computing. https://www.rdocumentation.org/packages/base/versions/4.1.0/topics/object.size
[5] Wickham, H., & Csárdi, G. (2020). nycflights13: Flights that Departed NYC in 2013. R package version 1.1.0. https://CRAN.R-project.org/package=nycflights13
[6] Müller, K., Wickham, H., & François, R. (2021). tibble: Simple Data Frames (R version 4.1.0). RStudio. https://tibble.tidyverse.org
[7] Montgomery, D. C. (2017). Design and Analysis of Experiments. John Wiley & Sons.
[8] Agresti, A., & Franklin, C. (2018). Statistics: The Art and Science of Learning from Data
[9] Field, A., Miles, J., & Field, Z. (2012). Discovering Statistics Using R. SAGE Publications
[10] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction.

Springer Science & Business Media.
[11] Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). Introduction to Linear Regression Analysis. John Wiley & Sons.
[12] Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2004). Applied Linear Statistical Models. McGraw-Hill.
[13] Draper, N. R., & Smith, H. (1998). Applied Regression Analysis (3rd ed.). Wiley-Interscience.
[14] Wickham, H. (2021). Memory. Advanced R. http://adv-r.had.co.nz/memory.html