Analysis of CSV File Reading in R

Quang Nguyen Robert Tedesco

STAT 407 Fall 2021

Abstract

write some stuff

1 Introduction

Data importing is undoubtedly a crucial part of a modern statistics and data science workflow, as it is the first step that paves the path for important stages such as data wrangling, visualizing, and modeling. With modern computers, loading data may seem like a straightforward task for a statistician. However, as the volume of data increases, this imposes a new set of challenges related to computing time and efficiency for data importing. A researcher may have to load a large data file multiple times a week in order to perform statistical analyses, which could potentially cost them valuable time. Identifying the fastest computing techniques is a necessity in the computational problem of working with big data.

The `R` programming language (R Core Team, 2021) is well-known among statisticians and data scientists as a powerful tool for working with data. In `R`, a popular method used by traditional statisticians to load a comma-separated values (CSV) file into the environment is `read.csv()`, which is a built-in base function. However, there are some drawbacks with this method, when working with large data sets. As demonstrated by Matthews (2021), it took minutes to import a CSV file of about two-gigabyte containing over a million rows of NFL tracking data with the historical `read.csv()` function in base `R`. Fortunately, in recent years, `R` developers have come up with better ways to read in data files. As mentioned by Gillespie & Lovelace (2021) in their "Efficient R programming" book, there are three modern functions/packages that `R` users should take advantage of in order to improve file-reading performance.

The first function that `R` programmers should consider is `import()`, which comes from the `rio` package (Chan et al., 2021). As described by its creators in the package documentation, `rio` is "a swiss-army knife for data I/O." The function `import()` in `rio` essentially simplifies the data importing process since as a function by itself, `import()`

has the flexibility of reading in multiple file extensions, such as .json, .xls, .xlsx, in addition to the common .csv. The data is stored as a data.frame object after being loaded into R in using import().

Another R package for importing data that has gotten a lot of attention recently is readr (Wickham & Hester, 2021), which is also part of the popular tidyverse collection of packages (Wickham et al., 2019) for modern data science. Within readr, there is a read_csv() function for getting CSV files into R, alongside other functions of the read_*() family designed for other file extensions. A notable aspect of the functions in readr in general and read_csv() in particular is the data is stored as a tibble (a "modern" data.frame-typed object) in the R environment once it is read in.

The third and final package for efficient file reading in R is data.table (Dowle & Srinivasan, 2021), which contains the fread() function for data importing. As an individual function, fread() is also capable of loading a variety of file extensions, similar to rio::import(). As for object type, the fread() function imports and returns a data object of classes data.table and data.frame.

In this paper, we attempt to design a three-factor factorial experiment and perform analyses to compare the speed of the three CSV importing algorithms in R as mentioned in previous paragraphs. In particular, the independent variables we are interested in examining are 1) the R packages/functions; 2) the file size; and 3) the computer device. The paper is outlined as follows. We first describe our data generating process for our experiment and the methodologies used for analyses in Section 2. We then spend the next section, 3, on our analyses and results of this experiment. Lastly, in Section 4, we give a quick summary of our results as well as discuss possible future work related to this project.

References

Chan, C., Chan, G. C., Leeper, T. J., & Becker, J. (2021). *Rio: A swiss-army knife for data file i/o.* https://CRAN.R-project.org/package=rio

Dowle, M., & Srinivasan, A. (2021). *Data.table: Extension of ‘data.frame‘.* https://CRAN.R-project.org/package=data.table

Gillespie, C., & Lovelace, R. (2021). *Efficient r programming.* https://csgillespie.github.io/efficientR

Matthews, G. (2021). *Old dog (statistician) learns new trick (read_csv).* https://youtu.be/E5KJkooW4RY.

R Core Team. (2021). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing. https://www.R-project.org/

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., & Hester, J. (2021). *Readr: Read rectangular text data.* https://CRAN.R-project.org/package=readr