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.

2 Experiment

3 Analysis of Variance

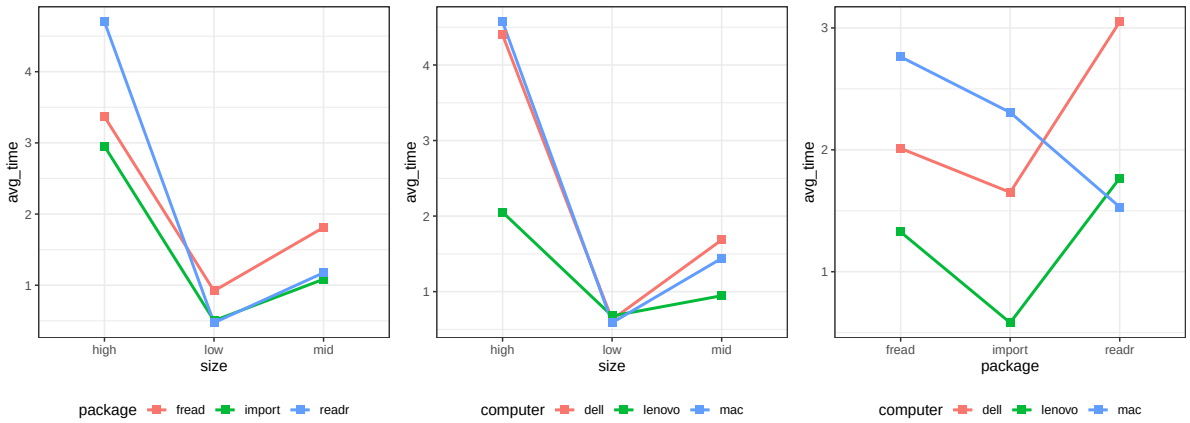

Figure 1: Interaction plots

Table 1: ANOVA output

term	df	sumsq	meansq	statistic	p.value
computer	2	118.61550	59.30775	12.801734	0.0000038
size	2	908.98520	454.49260	98.103425	0.0000000
package	2	38.49738	19.24869	4.154880	0.0162187
computer:size	4	136.35534	34.08884	7.358165	0.0000091
computer:package	4	114.57620	28.64405	6.182894	0.0000728
size:package	4	87.98781	21.99695	4.748100	0.0009060
computer:size:package	8	144.36547	18.04568	3.895208	0.0001776
Residuals	513	2376.62143	4.63279	NA	NA

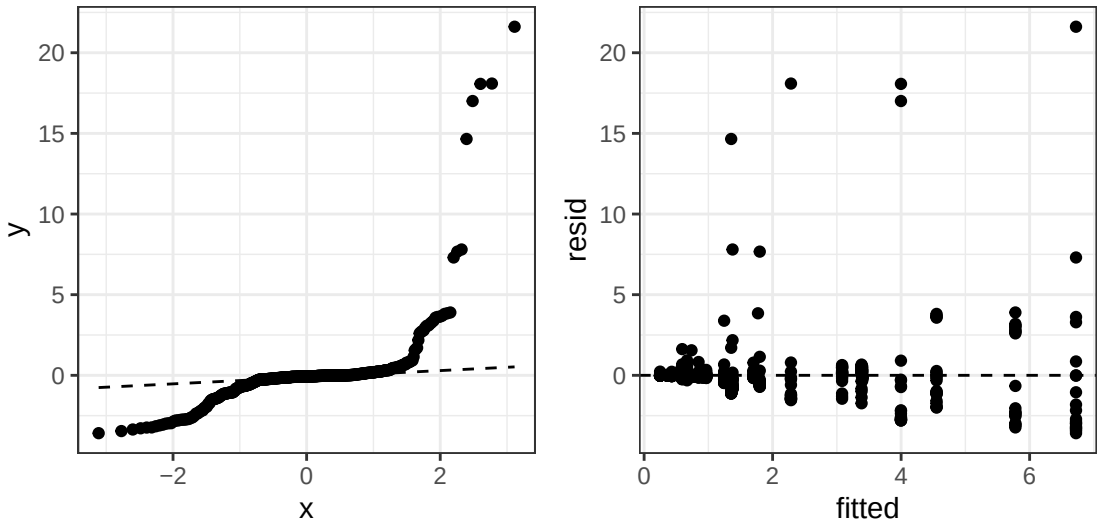

Figure 2: Residuals plots

Table 2: Shapiro-Wilk test for normality

statistic	p.value	method
0.4388194	0	Shapiro-Wilk normality test

Table 3: Levene's Test for Homogeneity of Variance (center = median)

statistic	p.value	df	df.residual
3.94106	0	26	513

Mention other transfomations, put in appendix

Permutation

Table 4: Permutation test

Term	InitialF	DF	pval
size	98.103425	2	0.0001000

Term	InitialF	DF	pval
computer	12.801734	2	0.0001000
package	4.154880	2	0.0123012
size:computer	7.358165	4	0.0001000
size:package	4.748100	4	0.0007001
computer:package	6.182894	4	0.0002000
size:computer:package	3.895208	8	0.0003000
Residual	NA	513	NA

Comparison	Stat	p.value	p.adjust
high - low = 0	9.536	0	0.00e+00
high - mid = 0	7.316	2.56e-13	0.00e+00
low - mid = 0	-4.424	9.688e-06	2.91e-05

Comparison	Stat	p.value	p.adjust
fread - import = 0	2.256	0.0241	0.0723
fread - readr = 0	-0.254	0.7995	1.0000
import - readr = 0	-2.103	0.03547	0.1064

Comparison	Stat	p.value	p.adjust
dell - lenovo = 0	3.3	0.0009683	0.0029050
dell - mac = 0	0.1327	0.8945	1.0000000
lenovo - mac = 0	-3.762	0.0001687	0.0005061

4 Additional Analysis

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

Appendix

log transfomation

Table 8: ANOVA output

term	df	sumsq	meansq	statistic	p.value
computer	2	53.742278	26.8711392	142.520161	0.0000000
size	2	266.521492	133.2607459	706.793365	0.0000000
package	2	5.238508	2.6192539	13.892098	0.0000013
computer:size	4	11.402660	2.8506650	15.119464	0.0000000
computer:package	4	25.138319	6.2845796	33.332390	0.0000000
size:package	4	6.294695	1.5736736	8.346509	0.0000016
computer:size:package	8	2.716952	0.3396190	1.801284	0.0744080
Residuals	513	96.722417	0.1885427	NA	NA

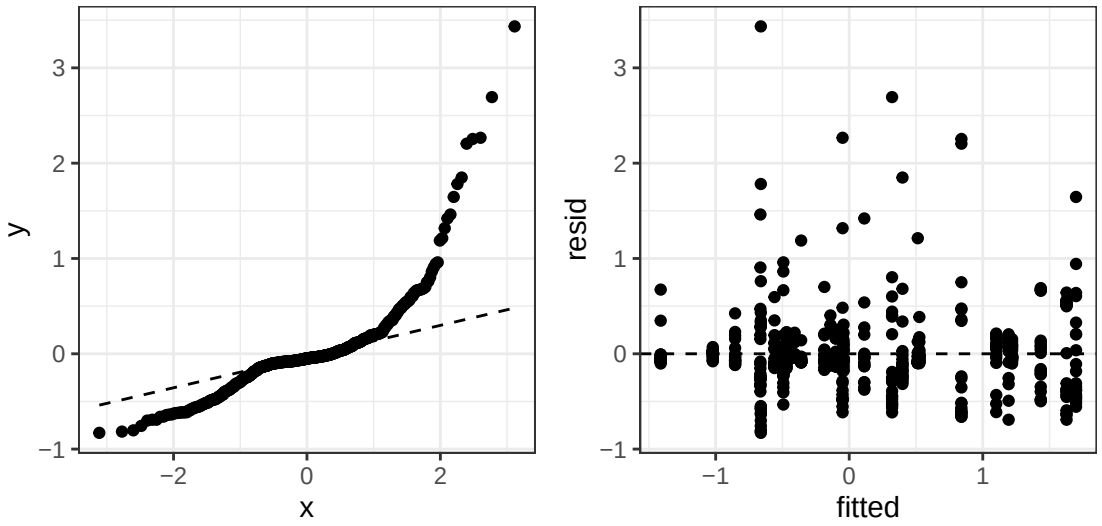

Figure 3: Residuals plots

Table 9: Shapiro-Wilk test for normality

statistic	p.value	method
0.7580349	0	Shapiro-Wilk normality test

Table 10: Levene's Test for Homogeneity of Variance (center = median)

statistic	p.value	df	df.residual
5.15043	0	26	513

Box-Cox Transformation

Table 11: ANOVA output

term	df	sumsq	meansq	statistic	p.value
computer	2	3.3531033	1.6765516	191.038583	0.0000000
size	2	15.8894828	7.9447414	905.282074	0.0000000
package	2	0.3277760	0.1638880	18.674603	0.0000000
computer:size	4	0.2930685	0.0732671	8.348594	0.0000016
computer:package	4	1.6164808	0.4041202	46.048418	0.0000000
size:package	4	0.3027577	0.0756894	8.624607	0.0000010
computer:size:package	8	0.1614617	0.0201827	2.299767	0.0199088
Residuals	513	4.5020800	0.0087760	NA	NA

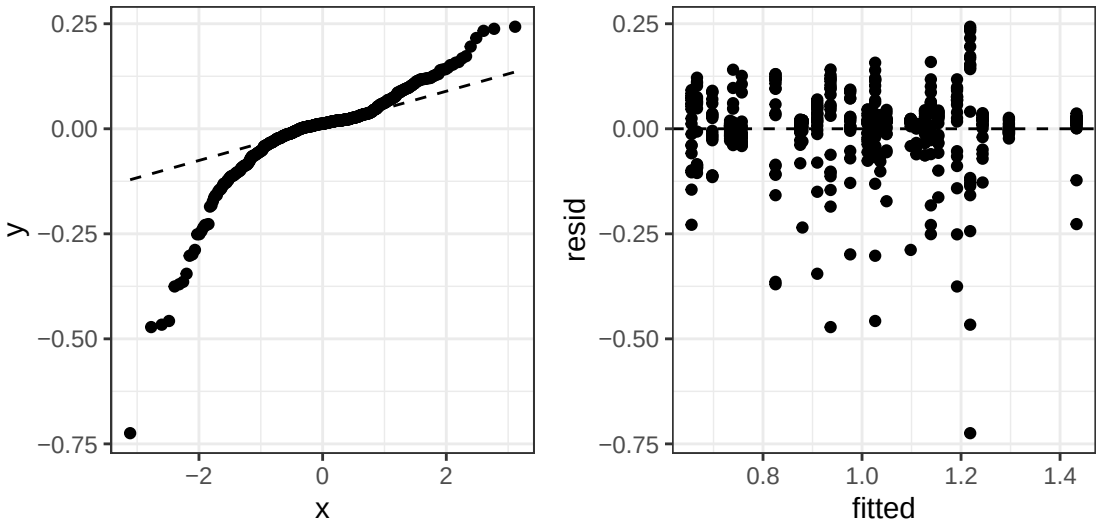

Figure 4: Residuals plots

Table 12: Shapiro-Wilk test for normality

statistic	p.value	method
0.8082856	0	Shapiro-Wilk normality test

Table 13: Levene's Test for Homogeneity of Variance (center = median)

statistic	p.value	df	df.residual
6.13923	0	26	513