DOI: http://doi.org/10.4038/sljastats.v23i1.8058

Sri Lankan Journal of Applied Statistics, Vol (23-1)

ptsuite: Fast Tail Index Estimation for Power Law Distributions in R

R. Munasinghe[1*], D. Jayasinghe[2], and P. Kossinne[3]

[1]MIND Analytics and Management (Pvt) Ltd, Sri Lanka

[2]University of Sri Jayewardenepura, Sri Lanka

[3]University of Calgary, Canada

*Corresponding author: ranjiva@mindlanka.com

Received: 3rd February 2022/ Revised: 24th July 2022/ Published: 31st August 2022

ABSTRACT

Power law distributions, in particular Pareto distributions, describe data across diverse areas of study. We have developed a package, `ptsuite`*, in* `R` *to estimate the tail index for such datasets which: a) uses a variety of estimation methods; b) focuses on speed (in particular with large datasets); c) is accurate and d) is easy to use. The package is also able to generate Pareto data as well as conduct both heuristic and statistical tests to check if data is Paretian. We tested* `ptsuite` *against similar* `R` *packages for speed of tail index estimation and found that our package is indeed faster. The tail index estimates produced by the package are accurate. The package is easy to use as all functions can be called with one line of code and a small number of argument references. To date the package has been downloaded over 12,500 times from the CRAN repository [4]. Finally we remark that the authors have used the package in research applications - e.g. (Munasinghe et. al , 2019).*

Keywords: Power law distributions, fat tailed distributions, heavy tailed distributions, Pareto distributions, tail index, tail estimation, fast computation, R, data analysis.

MSC 2010: 60-04, 60-08, 91-04, 91-08.

[4]As of July 1, 2022 according to statistics on https://shinyus.ipub.com/cranview/ .

1 Introduction

Power law distributions account for phenomena and observations across diverse fields including physics, astronomy, biology, economics and the social sciences (Sornette , 2006). Specific examples include distribution of wealth, word frequency in a given text, population in a city, stock market returns (Newman , 2005). The most general form of the survival function describing a power law distributions is given by (Taleb , 2020):

$$S(x) = L(x)x^{-\alpha}, \tag{1}$$

where $L(x)$ is a slowly-varying function, i.e.

$$\lim_{x\to\infty} \frac{L(\lambda x)}{L(x)} = 1 \ \ \forall\lambda > 0$$

The (tail) exponent α is referred to as the tail index[5]. A particular case is that of Pareto distributions, a special class of power laws with probability density function[6] (pdf) defined for $x \geq x_{\text{min}} > 0$:

$$p(x) = \frac{\alpha x_{\text{min}}^{\alpha}}{x^{1+\alpha}} \tag{2}$$

In the present era of 'Big Data', it is essential for programs to be geared for speed and have the ability to handle large data sets. In particular, there are applications in genetics involving large data sets where power-law distributions describe network connectivity (Zhang & Horvath , 2005). We found some of the existing tail estimation packages in `R` are not optimized for speed especially when dealing with a large data set. The `R`-package `ptsuite` was developed to estimate the tail index for such (large) datasets with a focus on speed. In speed tests conducted, `ptsuite` outperformed similar R packages such as `laeken, evir, EnvStats, DeMAND` (Munasinghe et. al , 2020). In addition `ptsuite` can 1) rapidly generate (large) Pareto data sets and 2) test if data is Pareto distributed. The package was also designed to be easy to use. The accuracy of the tail index estimates are tested and validated - the details in full are in (Munasinghe et. al , 2019, 2020).

[5] The tail index is also referred to as shape parameter in the context of Pareto distributions, stability parameter in the context of stable distributions and degrees of freedom in the context of t-distributions.

[6] The pdf is the negative derivative of the survival function: $p(x) = -\frac{dS(x)}{dx}$

2 Overview of ptsuite package

The ptsuite package has three main features: 1) generation of Pareto random variate data, 2) testing for Pareto distribution of data, and 3) tail index estimation. In this paper we provide an overview of the features. The package was designed for statistical applications to: 1) determine the nature of the distribution (of the data) and 2) estimate relevant parameters. We feel ptsuite is beneficial for general statistical research as the necessary functions are in one package and very easy to call. Instructions on use can be found in the following references: (Munasinghe et. al , 2019, 2020). The other packages mentioned in this paper were developed for specific purposes and could be more appropriate in those settings - e.g. laeken was developed for estimation of indicator in social exclusion & poverty; EnvStats was developed specifically for environmental statistics applications.

2.1 Generating Pareto Data

The first feature we introduce in ptsuite is that of generating Pareto data. This is done using the generate_pareto function which generates random Pareto distributed data with the specified shape and scale parameters. The function has been written to be similar in type to the popular runif and rexp type of functions for generating data from a particular distribution. The full call of the function is in R is `generate_pareto(sampleSize, shape, scale)`.

2.2 Testing if Data is Pareto Distributed

To identify Pareto behavior we can look for linear behavior in the QQ-plots of the log transformed data against standard exponential data (Hubert et. al , 2013). This is a heuristic test where one can look for data being distributed along a straight line indicating a possible Pareto distribution. A simple implementation of this method is included in our package using the pareto_qq_test function, which can be used as a first step to identify whether the data is Pareto distributed before estimating the tail index. If most of the data points appear to be distributed along a line, it is likely that the data may be Pareto. Conversely, if most of the data are distributed non-linearly, then the data is unlikely to be Pareto distributed. This function[7] plots the quantiles of the standard exponential distribution on the x-axis and the log values of the data on the y-axis. If Pareto data was supplied, a log transformation of this data would result

[7] For added interactivity this package makes use of the plotly package (Sievert , 2018) to generate the Q-Q plots if it is available in the user's R library. If unavailable, it defaults to the R base plot.

in an exponential distribution with mean α^{-1}. These data points would then show up on the QQ-plot as line with slope α^{-1}.

Demonstration of this function is carried out using a Pareto distributed sample of size $100,000$ with α (shape parameter) = 1.2 and x_{min} (scale parameter) = 3, and using an exponentially distributed sample of size $100,000$ with λ (reciprocal of mean) = 5 as an example of non-Pareto data.

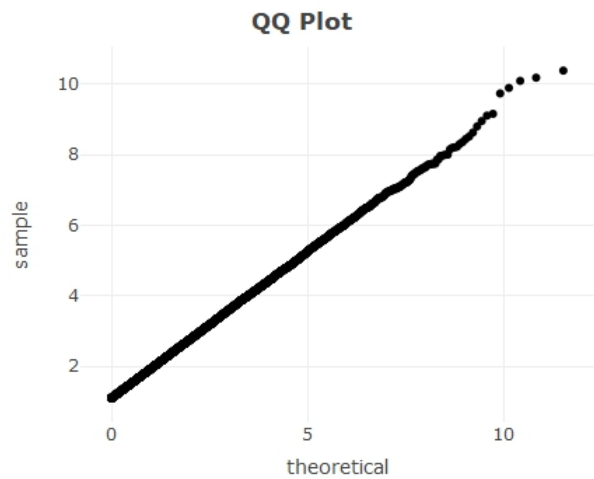

Fig. 1: Q-Q Plot for Pareto distributed data.

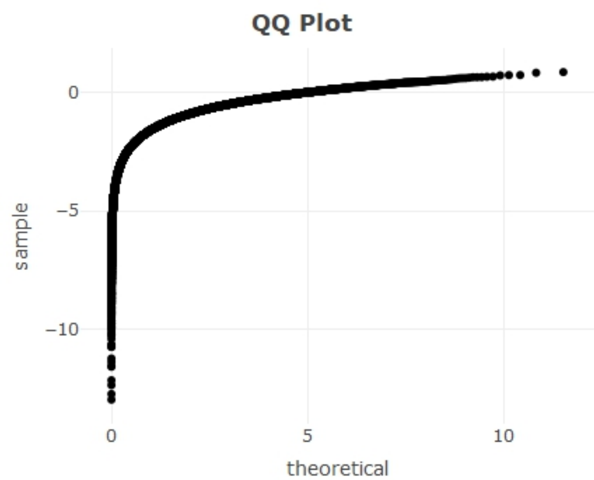

Fig. 2: Q-Q Plot for exponentially distributed data.

The `pareto_test` function can be further used to identify whether the data is Pareto distributed (Gulati & Shapiro , 2008). The test generates a p-value corresponding to the actual distribution of the data and is tested for significance. In the case of Pareto data, the p-value should be greater than the pre-determined significance level (generally taken as 0.05). A complete set of results confirming the validity of the results produced by the function can be found in (Munasinghe et. al , 2019, 2020).

2.3 Tail Index Estimation with `ptsuite`

After having conducted tests for Paretian tails, the next step would be to estimate the tail index. The list of the estimation method functions available in `ptsuite` are:

1. `alpha_mle`: Maximum Likelihood Estimator (Newman , 2005; Rizzo , 2009; Rytgaard , 1990)
2. `alpha_ls`: Least Squares Estimator (Nair et. al , unpublished.; Zaher et. al , 2014)
3. `alpha_moment`: Method of Moments Estimator[8] (Rytgaard , 1990)
4. `alpha_percentile`: Percentile Estimator method (Bhatti et. al , 2018)
5. `alpha_modified_percentile` : Modified Percentile Estimator method (Bhatti et. al , 2018)
6. `alpha_geometric_percentile`: Geometric Percentile Estimator method (Bhatti et. al , 2018)
7. `alpha_wls`: Weighted Least Squares Estimator (Nair et. al , unpublished.)
8. `alpha_hills`: Hill's Estimator (Hill , 1975)
9. `generate_all_estimates`: Generates all of the above estimates for a dataset except Hill's Estimator.

Technical summaries and the R implementation for the methods can be found in (Munasinghe et. al , 2019, 2020). Tail index estimation is very simple as all functions can be called with one line of code and the majority only requiring the specification of the data set. The `alpha_mle` function can be called in three different ways to obtain:

[8] We should note for the Moment Estimator method (Rytgaard , 1990), there is a major drawback if $\alpha \leq 1$ as the estimator will not converge to the true value - in fact for this estimator $\hat{\alpha} \longrightarrow 1$ when $\alpha \leq 1$ (Brazauskas & Serfling , 2000). It is always better to use this method in conjunction with another to check the validity of the estimate.

```
Method.of.Estimation Shape.Parameter Scale.Parameter
1  Maximum Likelihood Estimate        1.201384        3.000005
2                Least Squares        1.200670        3.000005
3            Method of Moments        1.241772        3.000005
4           Percentiles Method        1.200480        3.000005
5  Modified Percentiles Method        1.196936        3.000005
6 Geometric Percentiles Method        1.195801        3.000005
7       Weighted Least Squares        1.201303        3.000005
```

Fig. 3: Sample output of `generate_all_estimates` function for a generated data set of 100,000 Pareto random variates with α=1.2 and x_{min}=3.

1) biased estimates for α and x_{min} (Newman , 2005), 2) unbiased estimates for α and x_{min} (Rizzo , 2009; Rytgaard , 1990) and 3) biased estimates for α and x_{min} and the 95% confidence interval for α (Rytgaard , 1990).

2.4 A Note on Hill's Estimator

Hill's Estimator differs from the other methods in this paper as one needs to specify a cut-off point from which to consider the "tail" portion of the distribution. This is denoted by the `k` value which is the k^{th} largest observation of the data set. In other words, it can be thought of as the number of observations of the tail to be considered in estimating the shape parameter (tail index). We may also specify a value as a starting point of the tail estimation area instead of a ranked order. Identifying where the tail starts is an art in itself and requires a thorough examination of the data. If one chooses the parameter k to be too large the variance of the estimator increases; on the other hand if it is too low the bias increases (Fedotenkov , 2020).

2.5 Hill's Estimator for Strictly Pareto data

In this section we look at the impact of `k` when Hill's Estimator in `ptsuite` is used on strictly Pareto data. We consider random Pareto samples (with $\alpha = 0.5, 1.5, 2.2$ and 5.0) and consider 10^3, 10^5 and 10^6 sample sizes along with `k` set at the 25^{th}, 50^{th} and 75^{th} percentiles for each sample size. Refer (Munasinghe et. al , 2019, 2020) for the full set of results. We also generate four samples from the Pareto distribution with $\alpha = 1.5$ and $\hat{x}_{\text{min}} = 5$. Figure 5 shows how the estimator varies with `k`. Since the data supplied is strictly Pareto, increasing `k` (from 1 to `n` = 100,000) increases the accuracy of the estimate and we can see the estimate stabilizing around $\hat{\alpha} = 1.5$.

2.6 Hill's Estimator for Non-Pareto data

To further test the accuracy of the Hill's Estimator in `ptsuite` and understand

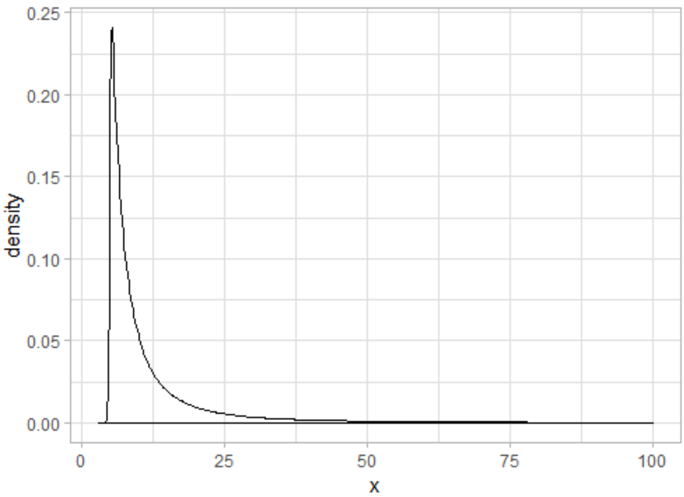

Fig. 4: Density plot of Pareto distribution.

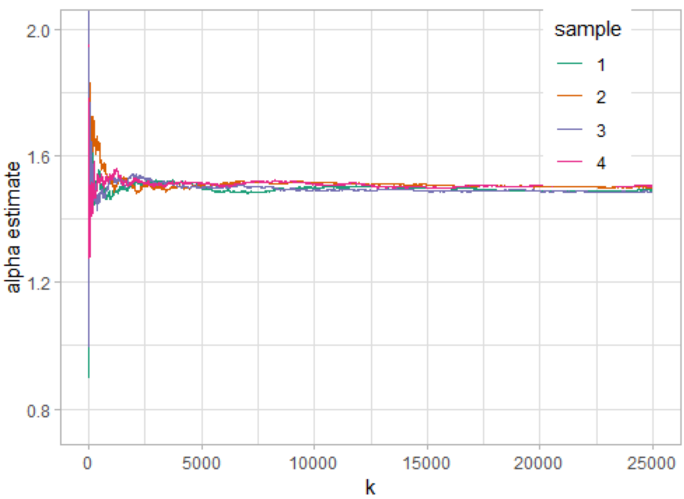

Fig. 5: Hill's estimate for α with varying `k`.

the relationship of the value of `k` in estimating α, we vary `k` in the presence of non-Pareto power law data. We consider data sampled from a i) t-distribution and ii) stable distribution respectively, and estimate the tail-index. For this purpose we first generate four samples from the t-distribution with degrees of freedom (d.f.) equal to 3. The tail index α is estimated by varying `k` from 1 to `n` (where `n` = 100,000). It can be seen from Figure 7 that using a smaller value of `k` results in a more accurate estimation of the tail index - we feel this is because a smaller value of `k` avoids including observations from the "body" of the distribution. We should note however the earlier point about the bias of the estimator for small `k`.

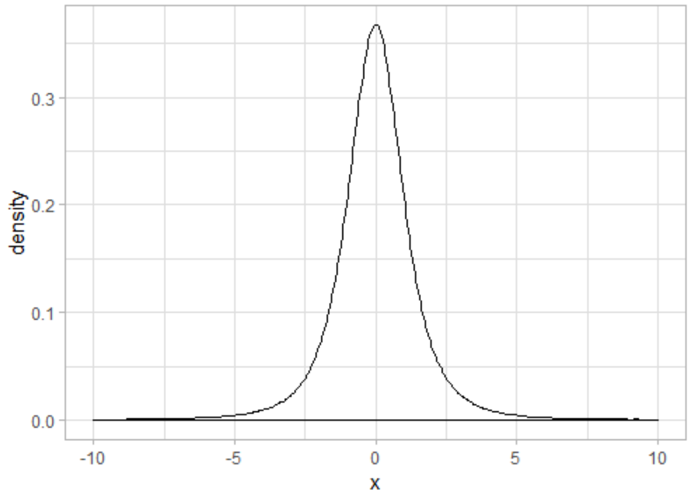

Fig. 6: Density plot of t-distribution.

Next we simulate four samples from a symmetric stable distribution[9] with stability parameter[10] (tail index) α=1.5 and vary `k` to estimate α. The results of this simulation are similar to that of t-distribution in that a smaller value of `k` leads to a closer estimate of the true α. The results are as follows:

3 Applications & Reuse potential

The package `ptsuite` has been used to verify power law phenomena in scientific publications (Munasinghe et. al , 2019)[11] and has been used it in private research

[9]A symmetric stable distribution was generated using $\frac{\sin(\alpha V)}{(\cos(V))^{1/\alpha}}(\frac{\cos((1-\alpha)V)}{W})^{(1-\alpha)/\alpha}$ where $V \sim Unif[-\pi/2, \pi/2]$ and $W \sim Exp(1)$ (Glasserman , 2004).

[10]A symmetric stable distribution with stability parameter α has the following property $\lim_{x\to\pm\infty} p(x) \sim |x|^{1+\alpha}$.

[11]The authors used `ptsuite` to verify expected power law behaviour in simulations of absorbing random walks (Dickman et. al , 2003).

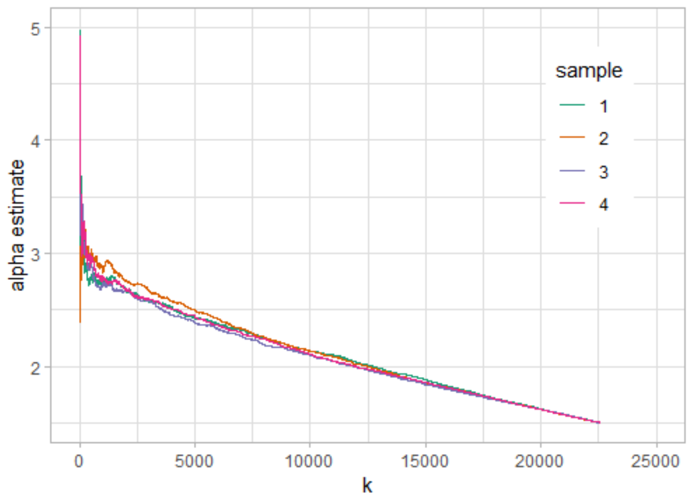

Fig. 7: Hill's estimate for α with varying `k`.

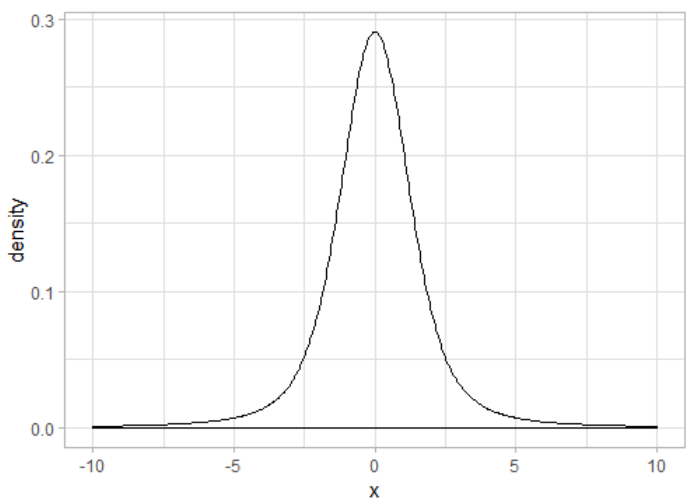

Fig. 8: Density plot of stable distribution.

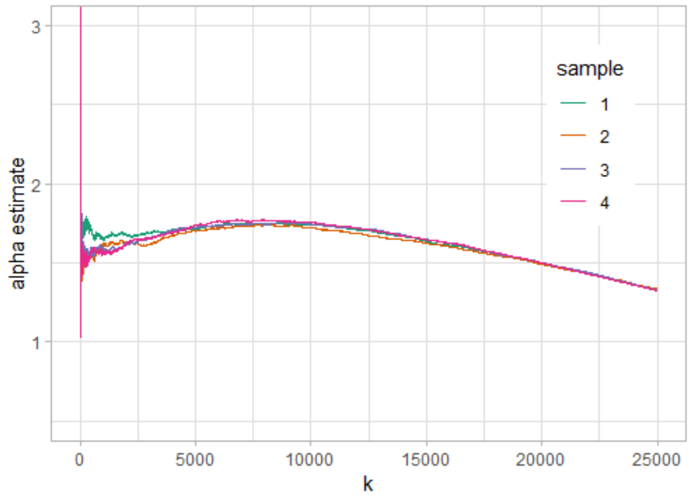

Fig. 9: Hill's estimate for α with varying `k`.

analyzing sales data. In particular, we have verified the Pareto nature of retail items, video views on social media channels etc. Due to the ubiquity of power laws across multiple disciplines, we the authors of the package hope that it will assist other researchers and students with its combination of features, speed and ease of use. The package has over 12,500 downloads from the RStudio CRAN Mirror[12]. In addition, we believe that the package could be used for educational purposes. For future work we hope to add more tail estimation methods and functions found in the references to this package - in particular we hope to eventually incorporate the vast catalog of estimators in reference (Fedotenkov , 2020). It would also be great to look at the possibilities of incorporating GPU computing (e.g. NVIDIA CUDA, RPUD) for increased processing performance and speed. The authors welcome interested developers to contribute code by mail or as a pull request in the code repository. The main motivation for this article and the accompanying `R`-package (**?**) is a study in a forthcoming publication (Munasinghe & Jayasinghe , in prep.) analyzing power law data.

4 Results for Accuracy

To test for accuracy, we use the `ptsuite` package to simulate Pareto data using the `generate_pareto` function and then apply the appropriate estimator function to evaluate the shape and scale parameters. The test results for accuracy indicate good agreement with the theoretical values with the full set of test results in (Munasinghe

[12]As per check on 01/07/2022 according to `https://ipub.com/dev-corner/apps/r-package-downloads`.

Table 1: Average estimates of shape parameter = 1.5 for each method of estimation with standard deviations and 95% Confidence Intervals

Estimator	Avg Estimate	Std Dev	95% C.I for Avg Estimate
MLE	1.501935	0.01522033	(1.4725358432174, 1.5313347627536)
Least Squares	1.49239	0.02218303	(1.46299037215539, 1.52178929169159)
MoM	1.510585	0.04594459	(1.48118526703732, 1.53998418657352)
Percentiles	1.502885	0.02278928	(1.47348522656043, 1.53228414609663)
Modified Percentiles	1.502816	0.03241043	(1.47341629661594, 1.53221521615214)
Geometric Percentiles	1.504332	0.0438464	(1.47493299938794, 1.53373191892414)
WLS	1.501106	0.01521192	(1.47170615579896, 1.53050507533516)

et. al , 2020). As expected, larger sample sizes result in estimates of greater accuracy. In Table 1 we present results for 500 trials for datasets with $\alpha = 1.5$, scale parameter = 2 and sample size of 10^4. Calculation of confidence intervals (Omar et. al , 2012) were performed with 95% level of confidence.

5 Results for Speed

In this section we look at the performance of functions in the `ptsuite` package against their (comparable) counterparts in other packages. All estimator functions in the `ptsuite` package have been coded in `C++` to ensure the highest possible performance. We compare the speed of the functions available in the package using the package `microbenchmark` (Olaf , 2018). We compare the following estimators/functions from the `ptsuite` package and their counterparts in Table 2:

Table 2: Existing `R` packages and their functions corresponding to `ptsuite`.

Estimator/function	Package	Function in Package
Hill's Estimator	`ptsuite`	`alpha_hills`
	`laeken`	`thetaHill`
	`evir`	gpd (set `nextremes` accordingly)
Method of Moments	`ptsuite`	`alpha_moment`
	`laeken`	`thetaMoment`
MLE	`ptsuite`	`alpha_mle`
	`EnvStats`	`epareto` (set `method = `mle'`)
	`evir`	gpd (set `nextremes` to the sample size)
	`DeMAND`	`pareto.fit`
Leaset Squares	`ptsuite`	`alpha_ls`
	`EnvStats`	`epareto` (set `method = `ls'`)
Pareto Data Generation	`ptsuite`	`generate_pareto`
	`EnvStats`	`rpareto`
	`evir`	`rgpd`

We complete a hundred runs for a selected sample using `ptsuite` and the counterpart. We capture the following times: mean, maximum, median and minimum. These times are all measured in microseconds (μs). In all the tests, `ptsuite` comes out on top. We should note however for some of the functions, the fastest (minimum) time of the competing package has been faster than the slowest (maximum) time of the `ptsuite`. Please refer Table 3 to Table 8 for the speed results. (Please refer to the Appendix for the complexity graphs for the functions in our package.)

Table 3: Function timing results using `microbenchmark` for Hill's Estimate with sample 1.

Sample Size	Package	Mean (μs)	Maximum (μs)	Median (μs)	Minimum (μs)
10^5	ptsuite	3.94E+03	4.17E+03	3.97E+03	3.79E+03
	laeken	5.92E+03	8.23E+03	4.82E+03	4.51E+03
	evir	2.40E+05	3.47E+05	2.38E+05	2.32E+05
10^6	ptsuite	4.18E+04	5.28E+04	4.19E+04	3.95E+04
	laeken	7.21E+04	1.15E+05	9.19E+04	4.67E+04
	evir	2.47E+09	2.58E+09	2.49E+09	2.37E+09
10^7	ptsuite	4.47E+05	4.92E+05	4.44E+05	4.21E+05
	laeken	7.10E+05	1.23E+06	8.11E+05	4.69E+05
	evir	2.49E+10	2.67E+10	2.46E+10	2.41E+10

Table 4: Function timing results using `microbenchmark` for Method of Moments Estimate with sample 2.

Sample Size	Package	Mean (μs)	Maximum (μs)	Median (μs)	Minimum (μs)
10^5	ptsuite	3.79E+03	3.92E+03	3.79E+03	3.76E+03
	laeken	9.81E+03	1.72E+04	9.36E+03	9.25E+03
10^6	ptsuite	3.91E+04	5.12E+04	3.87E+04	3.81E+04
	laeken	1.33E+05	1.71E+05	1.29E+05	1.19E+05
10^7	ptsuite	3.86E+05	4.12E+05	3.84E+05	3.82E+05
	laeken	1.44E+06	1.58E+06	1.43E+06	1.39E+06

Table 5: Function timing results using `microbenchmark` for Maximum Likelihood Estimate with sample 3.

Sample Size	Package	Mean (μs)	Maximum (μs)	Median (μs)	Minimum (μs)
10^5	ptsuite	3.22E+03	3.54E+03	3.19E+03	3.15E+03
	EnvStats	7.94E+03	2.10E+04	7.53E+03	6.84E+03
	evir	2.47E+05	3.70E+05	2.45E+05	2.33E+05
10^6	ptsuite	3.23E+04	3.48E+04	3.21E+04	3.18E+04
	EnvStats	1.23E+05	1.70E+05	1.18E+05	1.06E+05
	evir	2.66E+06	2.84E+06	2.66E+06	2.54E+06
10^7	ptsuite	3.21E+05	3.35E+05	3.20E+05	3.17E+05
	EnvStats	1.40E+06	1.53E+06	1.39E+06	1.30E+06
	evir	2.66E+07	2.83E+07	2.65E+07	2.61E+07

Table 6: Function timing results using `microbenchmark` for MLE Estimate with sample 1. smaller sample sizes were used due to the slower run times of `DeMAND`.

Sample Size	Package	Mean (μs)	Maximum (μs)	Median (μs)	Minimum (μs)
10^2	ptsuite	8.61E+00	1.21E+01	8.80E+00	5.80E+00
	DeMAND	1.59E+04	3.64E+04	1.50E+04	1.47E+04
10^3	ptsuite	3.90E+01	5.78E+01	3.77E+01	3.44E+01
	DeMAND	3.33E+05	4.14E+05	3.29E+05	3.16E+05
10^4	ptsuite	3.27E+02	3.52E+02	3.24E+02	3.18E+02
	DeMAND	1.99E+07	2.09E+07	1.99E+07	1.95E+07

Table 7: Function timing results using `microbenchmark` for LS Estimate with sample 4.

Sample Size	Package	Mean (μs)	Maximum (μs)	Median (μs)	Minimum (μs)
10^5	ptsuite	7.84E+03	1.73E+04	7.56E+03	7.34E+03
	EnvStats	4.12E+04	6.60E+04	3.77E+04	3.42E+04
10^6	ptsuite	9.45E+04	1.91E+05	8.06E+04	7.66E+04
	EnvStats	3.28E+05	4.15E+05	3.23E+05	2.63E+05
10^7	ptsuite	9.17E+05	1.61E+06	8.79E+05	8.09E+05
	EnvStats	3.66E+06	4.87E+06	3.62E+06	3.34E+06

Table 8: Function timing results using `microbenchmark` for generating Pareto data with sample 5.

Sample Size	Package	Mean (μs)	Maximum (μs)	Median (μs)	Minimum (μs)
10^5	ptsuite	1.61E+04	2.33E+04	1.56E+04	1.54E+04
	EnvStats	3.62E+05	4.05E+05	3.58E+05	3.31E+05
	evir	1.57E+04	2.42E+04	1.56E+04	1.56E+04
10^6	ptsuite	1.59E+05	2.11E+05	1.56E+05	1.55E+05
	EnvStats	3.69E+06	4.02E+06	3.67E+06	3.59E+06
	evir	1.61E+05	1.79E+05	1.60E+05	1.58E+05
10^7	ptsuite	1.57E+06	1.68E+06	1.56E+06	1.55E+06
	EnvStats	3.67E+07	4.00E+07	3.64E+07	3.59E+07
	evir	1.60E+06	1.67E+06	1.60E+06	1.59E+06

6 Implementation and architecture

The sample code used to demonstrate use of `ptsuite` 1.0.0 package in this paper were using `R` 3.5.2. and Microsoft `R` 3.5.1 (Microsoft and Team , 2017). Microsoft `R` may be obtained from `https://mran.microsoft.com/open`. RStudio V 1.1.423 (RStudio Team , 2016) was used as the IDE for all tasks related to package development and generating results. The packages `devtools` (Wickham et. al , 2018) and `roxygen2` (Wickham et. al , 2018) were used to assist in package documentation and development. The `Rcpp` package (Vinson et. al , 2016; Eddelbuettel & Balamuta , 2018; Eddelbuettel & François , 2011) was used in the compilation of `C++` code used in all estimation functions of the package. QQ-plots in this document were generated using `ptsuite` with the dependent package `plotly` 4.8.0.
Two computers were used in the building of this package and the generation of the results. The technical specifications of the two computers are as follows:

Specification	Laptop	Desktop
CPU	Intel(R) Core(TM) i5-8250U @ 1.60GHz, 1800 Mhz, 4 Core(s), 8 Logical Processor(s)	Intel(R) Core(TM) i7-7700 @ 3.60GHz, 3601 Mhz, 4 Core(s), 8 Logical Processor(s)
RAM	8 GB	32 GB
OS	Windows 10 Enterprise LTSC 10.0.17763 Build 17763	Windows 10 Pro 10.0.17763 Build 17763

7 Quality control

All packages on CRAN undergo standardized checks to ensure compatibility with the R package system. The provided R package contains tests as well as examples, which were run on Windows 10 OS. Tests confirming the accuracy of the tail index estimates against known examples can be found in (Munasinghe et. al , 2019).

8 Availability

Operating system Works on all operating systems supporting R.
Programming language R (version 3.5.0 or higher)
Additional system requirements None
Dependencies R (version 3.5.0 or higher), Rcpp, plotly.
List of contributors Ranjiva Munasinghe, Pathum Kossinna, Dovini Jayasinghe, Dilanka Wijeratne.

8.1 Software locations Archive

Name: Zenodo
Persistent identifier: DOI: 10.5281/zenodo.3906154
Licence: GPL-3
Publisher: Ranjiva Munasinghe
Version published: 1.0.0
Date published: 24 June 2020

Code repository
Name: GitHub **Persistent identifier:** `https://github.com/RM-cyber/ptsuite`
Licence: GPL-3
Publisher: Ranjiva Munasinghe
Date published: 24 June 2020

Other Resources - Alternative Archive
Name: CRAN
Persistent identifier: `https://cran.r-project.org/package=ptsuite`
Licence: GPL-3
Publisher: Pathum Kossinna
Version published: 1.0.0
Date published: 02/04/2019

Language English

Acknowledgements
We wish to acknowledge Dr. J. Nair for helpful advice and in providing unpublished material for our research and Prof. S. Gulati for sharing knowledge on the Goodness of Fit tests. D. J. would also like to acknowledge Prof. S. Banneheka for a very useful discussion on the WLS method of estimation.

Competing interests
The authors declare that they have no competing interests.

References

Bhatti, S. H., Hussain, S., Ahmad, T., Aslam, M., Aftab, M., & Raza, M. A. (2018). *Efficient estimation of Pareto model: Some modified percentile estimators. PloS one, 13(5)*, e0196456.

Brazauskas, V., & Serfling, R. (2000). Robust and efficient estimation of the tail index of a single-parameter Pareto distribution. *North American Actuarial Journal, 4(4)*, 12-27.

Dickman, R., Araujo Jr, F. F., & Ben-Avraham, D. (2003). Variable survival exponents in history-dependent random walks: hard movable reflector. *Brazilian journal of physics, 33*, 450-457.

Vinson, D. W., Davis, J. K., Sindi, S. S., & Dale, R. (2016). Efficient n-gram analysis in R with cmscu. *Behavior research methods, 48(3)*, 909-921.

Eddelbuettel, D., & Balamuta, J. J. (2018). Extending R with C++: *A brief introduction to Rcpp. The American Statistician, 72(1)*, 28-36.

Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of statistical software, 40*, 1-18.

Fedotenkov, I. (2020). A review of more than one hundred Pareto-tail index estimators. *Statistica, 80(3)*, 245-299.

Gulati, S., & Shapiro, S. (2008). Goodness-of-fit tests for Pareto distribution. In Statistical models and methods for biomedical and technical systems (pp. 259-274). Birkhäuser Boston.

Hill, B. M. (1975). A simple general approach to inference about the tail of a distribution. *The annals of statistics*, 1163-1174.

Glasserman, P. (2004). Monte Carlo methods in financial engineering (Vol. 53, pp. xiv+-596). New York: springer.

Hubert, M., Dierckx, G., & Vanpaemel, D. (2013). Detecting influential data points for the Hill estimator in Pareto-type distributions. *Computational Statistics & Data Analysis, 65*, 13-28.

Olaf Mersmann, *microbenchmark: Accurate Timing Functions.* (2018) Available: https://CRAN.R-project.org/package=microbenchmark

Microsoft and Team, R. C., Microsoft, Redmond, Washington, USA. *Microsoft R Open.* (2017) [Online]. Available: https://mran.microsoft.com, Accessed on: Jul. 17, 2021.

R. Munasinghe and D. Jayasinghe, *Fat Tails and Power Laws in Business Data*. In preparation.

Munasinghe, R., Kanthan, L., & Kossinna, P. (2019). Revisiting Feller Diffusion: Derivation and Simulation. arXiv preprint arXiv:1905.10737.

R. Munasinghe, P. Kossinna, D. Jayasinghe and D. Wijeratne. *Tail Index Estimation for Power Law Distributions in R.* (2019) Vignette for R package version 1.0.0. Available: https://cran.r-project.org/web/packages=ptsuite.

R. Munasinghe, P. Kossinna, D. Jayasinghe and D. Wijeratne, "Fast Tail Index Estimation for Power Law Distributions in R", *arXiv e-prints* pp. arXiv:2006.10308, (2020). Available: https://ui.adsabs.harvard.edu/abs/2020arXiv200610308M.

Omar, A., Ibrahim, K., & Razali, A. M. (2012). Confidence interval estimation of the shape parameter of Pareto distribution using extreme order statistics. *Applied Mathematical Sciences, 93(6)*, 4627-4640.

J. Nair, A. Wierman and B. Zwart, *The fundamentals of heavy tails: Properties, emergence, and identification*. Unpublished. Draft avallable: http://users.cms.caltech.edu/ adamw/heavytails.html.

Newman, M. E. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary physics, 46(5)*, 323-351.

baseR R Core Team, R Foundation for Statistical Computing, Vienna, Austria. *R: A Language and Environment for Statistical Computing.* (2018) [Online]. Available: https://www.R-project.org, Accessed on: Jul. 17, 2021.

Rizzo, M. L. (2009). New goodness-of-fit tests for Pareto distributions. ASTIN Bulletin: *The Journal of the IAA, 39(2)*, 691-715.

RStudio Team, RStudio, Inc., Boston, MA, USA. *RStudio: Integrated Development Environment for R.* (2016) [Online]. Available: http://www.rstudio.com, Accessed on: Jul. 17, 2021.

Rytgaard, M. (1990). Estimation in the Pareto distribution. ASTIN Bulletin: *The Journal of the IAA, 20(2)*, 201-216.

C. Sievert. *plotly for R.* (2018). Available: https://plotly-book.cpsievert.me

Sornette, D. (2006). Critical phenomena in natural sciences: chaos, fractals, selforganization and disorder: concepts and tools. Springer Science & Business Media.

Taleb, N. N. (2020). Statistical consequences of fat tails: Real world preasymptotics, epistemology, and applications. arXiv preprint arXiv:2001.10488.

H. Wickham, P. Danenberg, and M. Eugster. *roxygen2: In-Line Documentation for R.* (2018) R package version 6.1.1. Available: https://CRAN.R-project.org/package=roxygen2

H. Wickham, J. Hester and W. Chang. *devtools: Tools to Make Developing R Packages Easier.* (2018) R package version 2.0.1. Available: https://CRAN.R-project.org/package=devtools

Zaher, H. M., El-Sheik, A. A., & El-Magd, N. A. A. (2014). Estimation of Pareto parameters using a Fuzzy least-squares method and other known techniques with a comparison. *British Journal of Mathematics & Computer Science, 4(14)*, 2067.

Zhang, B., & Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1).

Appendix

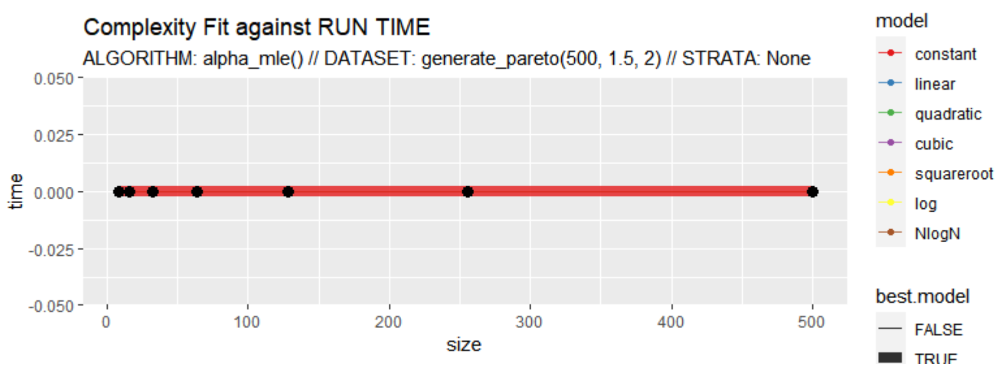

Fig. 10: `alpha_mle()`

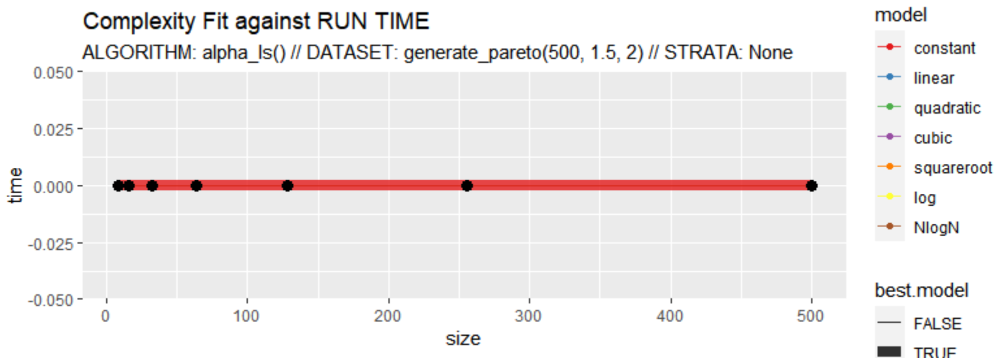

Fig. 11: `alpha_ls()`

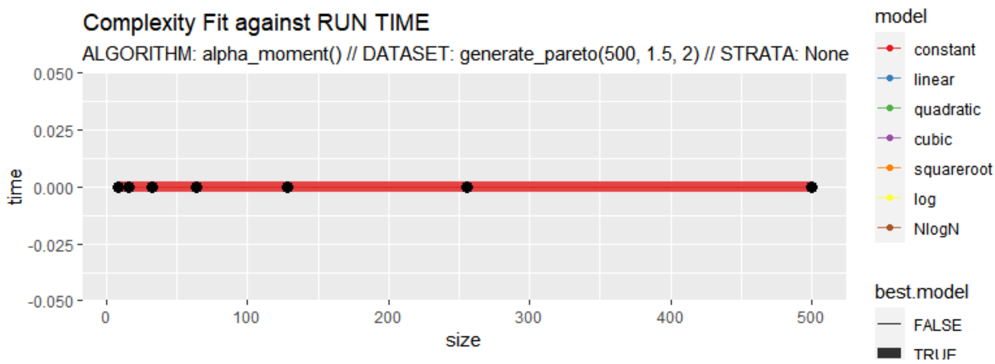

Fig. 12: `alpha_moment()`

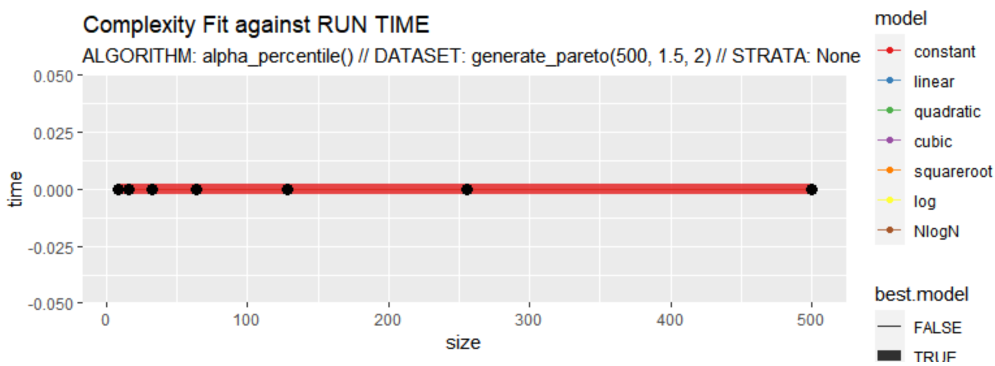

Fig. 13: `alpha_percentile()`

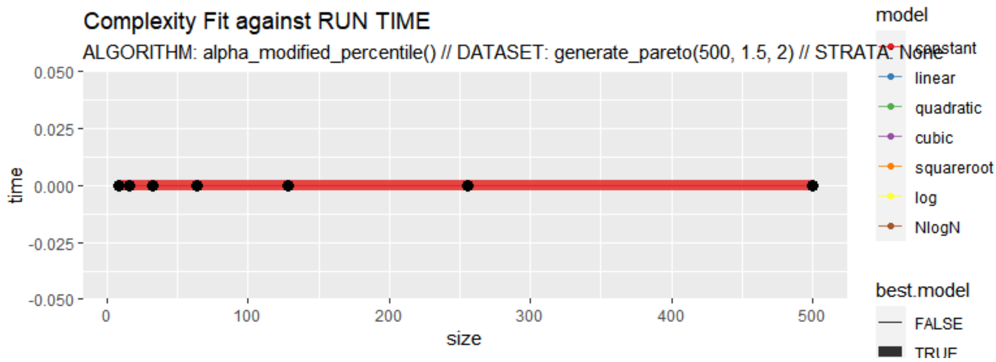

Fig. 14: `alpha_modified_percentile()`

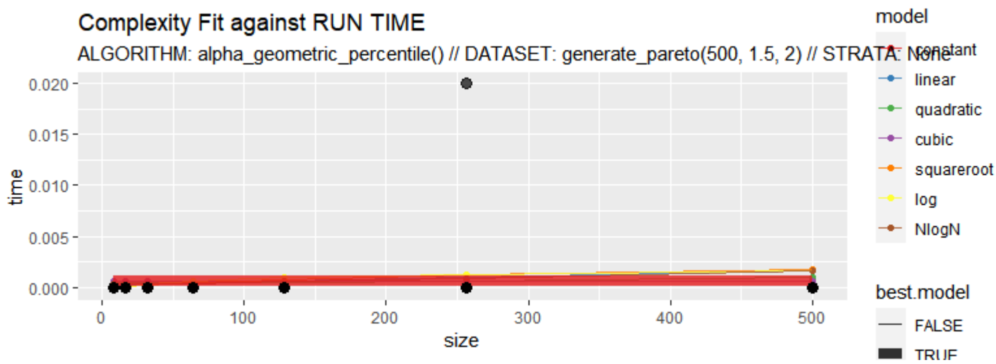

Fig. 15: alpha_geometric_percentile()

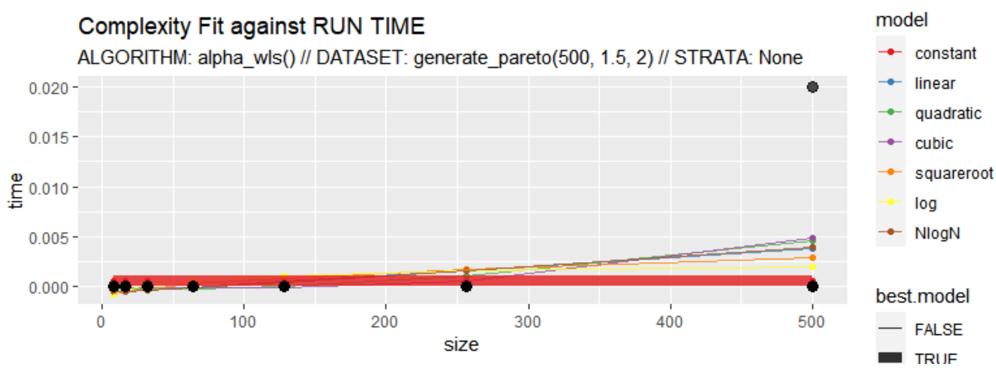

Fig. 16: `alpha_wls()`