Exploratory data analysis, as proposed in Tukey (1977), is an important paradigm for conducting meaningful and useful applied statistics. According to Chapters 5, 19, and 20 of Tukey (1977), the visualization of data, and their density and distributional characteristics, may provide practitioners with great insight into the necessary processes that are required in order to effectively analyse the various features of the data.

Since its introduction in the pioneering works of Rosenblatt (1956) and Parzen (1962), the method of kernel density estimation (KDE) has become among one of the most popular methods for estimation, interpolation, and visualization of probability density functions (PDFs), due to its effectiveness and simplicity. The manuscripts of Silverman (2018) and Wand & Jones (1994) provide thorough treatments on the topic of KDE.

In the base stats package of the R programming environment (R Core Team, 2016), univariate KDE can be conducted via the function, density. The function, density, is often taught in introductory R classes and presented in popular textbooks. See for example Section 5.6 of Venables & Ripley (2002), Section 7.4 of Troseth (2009), Section 6.3 of Keen (2010), Section 2.1 of Maindonald & Braun (2010), and Section 2.3 of Verzani (2014).

The KDE method implemented in density is the standard KDE technique for estimation of univariate PDFs, assuming that the data are real-valued. Unfortunately, density is often used to analyze data that are not real-valued, such as income data, which are positive-valued (cf, Charpentier & Flachaire, 2015). The use of density in the case of positive-valued data causes the obtained estimator of the PDF that characterizes the data to not integrate to one, over the positive domain. This implies that it will provide an incorrect specification of the data generating process.

In order to mitigate against the aforementioned shortcoming of density, Charpentier & Flachaire (2015), among others, have suggested the use of the kernel density estimators (KDEs), constructed with log-normal kernel functions. These log-transformed KDEs, or log-KDEs for brevity, are bona fide PDFs over the positive domain, and are thus suited for applications to positive-valued data.

The logKDE package for R provides functions for conducting log-transformed KDE using log-KDEs, constructed with log-normal, as well as log-transformations of the Epanechnikov, Laplace, logistic, triangular, and uniform families of kernel functions, through the main function, logdensity. The function allows for a variety of bandwidth methods, including Silverman’s rule of thumb (cf. Silverman, 2018), cross-validation in both the natural and log-transformation space, and a new rule of thumb that is based on the comparison of the log-KDE to a log-normal fit. Where necessary, we have programmed the various functionalities in C++, and integrated the C++ codes using Rcpp (Eddelbuettel, 2013). For greater speed, at the expense of some accuracy, we have also implemented a fast Fourier transform version of our procedure, via the function, logdensity_fft.
There are two packages that are currently available, which share similar features with logKDE. The first is Ake (Wansouwe, Some, & Kokonendji, 2016), via the ker = 'LN' setting of dke.fun, and the second is evmix (Hu & Scarrott, 2018), via the dbckden, with the setting bcmethod = 'logtrans'. Unlike logKDE, Ake only offers log-transformed KDE constructed from log-normal kernels. Although evmix allows for the construction of log-KDEs using a variety of kernels, including some that are not currently available in logKDE, it does not permit the use of the log-Laplace and log-logistic kernels. Thus, with respect to variety of kernels, both packages have something to offer that the other does not.

We believe that the key difference between evmix and logKDE is that of user experience. In logKDE, logdensity is designed to closely replicate the syntax and behavior of density. Thus, users who have learnt to use density will quickly make use of logdensity and its features. The syntax of dbckden is dramatically different to that of density. The function allows for many controls that are meant for higher level users and which may overwhelm someone who is only interested in conducting KDE as an exploratory tool. Therefore, we believe that logdensity provides an experience that is more user friendly and familiar than that of dbckden.

Users can obtain the latest build of logKDE on GitHub (https://github.com/andrewthomasjones/logKDE). The latest stable build can be obtained from CRAN (https://CRAN.R-project.org/package=logKDE), and an archival build can be obtained from Zenodo (https://zenodo.org/record/1339352). A detailed literature review, mathematical study, simulation study, and demonstration of the log-transformed KDE method appear in the vignette of the package, which can be accessed via the command vignette(‘logKDE’). Thorough descriptions of the package functions appear in the manual, which can be accessed at https://cran.r-project.org/web/packages/logKDE/logKDE.pdf. Bug reports and other feedback can be directed to the GitHub issues page (https://github.com/andrewthomasjones/logKDE/issues).

Acknowledgements

Hien Nguyen is personally funded under Australian Research Council (ARC) grant number DE170101134. Geoffrey McLachlan and Hien Nguyen are jointly funded by ARC grant number DP180101192.

References


