Journal of Statistical Software

March 2021, Volume 97, Issue 9. doi: 10.18637/jss.v097.i09

multimode: An R Package for Mode Assessment

Jose Ameijeiras-Alonso
Universidade de Santiago de Compostela

Rosa M. Crujeiras
Universidade de Santiago de Compostela

Alberto Rodríguez-Casal
Universidade de Santiago de Compostela

Abstract

In several applied fields, multimodality assessment is a crucial task as a previous exploratory tool or for determining the suitability of certain distributions. The goal of this paper is to present the utilities of the R package **multimode**, which collects different exploratory and testing non-parametric approaches for determining the number of modes and their estimated location. Specifically, some graphical tools (SiZer map, mode tree or mode forest) are provided, allowing for the identification of mode patterns, based on the kernel density estimation. Several formal testing procedures for determining the number of modes are described in this paper and implemented in the **multimode** package, including methods based on the ideas of the critical bandwidth, the excess mass or using a combination of both. This package also includes a function for estimating the modes locations and different classical data examples that have been considered in mode testing literature.

Keywords: multimodality, critical bandwidth, excess mass, bootstrap test.

1. A brief introduction on mode assessment

Given a data sample from a random variable, determining the number of modes in the underlying density is a relevant question for supporting further decisions during the modeling approach. It is clear that unimodal distributions (such as the Gaussian density) may not be adequate for characterizing the behavior of more complex data generating mechanisms in applied sciences. Some examples requiring more complex distributions for reflecting the real number of modes can be found in many applied fields, such as astronomy, e.g., in the study of unimodal or multimodal patterns of the stars rotation periods for different temperatures (McQuillan, Mazeh, and Aigrain 2014); business administration, e.g., when analyzing the invested capital in crowdfunding campaigns (Colombo, Franzoni, and Rossi-Lamastra 2015); forest science, e.g., in the analysis of the number of modes in the distribution of backscatter measurements (for unvegetated and dense forest areas), depending on the percentage of

ground pixels (Santoro *et al.* 2011); genetics, e.g., for identifying which *CpGs* (regions of DNA where a cytosine nucleotide is followed by a guanine nucleotide) present multimodal distributions (Joubert *et al.* 2016); or psychology, where, for example, the study of the number of modes is crucial for detecting the presence of single or dual-process cognitive phenomena (Freeman and Dale 2013); among others.

In principle, non-parametric density smoothers, such as the kernel density estimator introduced by Rosenblatt (1956) and Parzen (1962), may overcome the problem of restricting the density estimation to a previously specified parametric family. Nevertheless, two important issues arise when performing density estimation via kernel (or any other) density smoothing methods. The first issue is that practitioners may be more comfortable interpreting and dealing with parametric models, since in many cases parameter estimates can be interpreted in terms of the data distribution given that they control some specific features. The second issue is that, even being satisfied with the non-parametric kernel density estimator output, since it provides an estimated version of the underlying distribution, there may be some doubts about the features highlighted by this curve estimator: are they genuine from the distribution or are they just due to sampling variability?

The previous concerns can be partially solved or answered by the identification of the (significant) modes in the kernel density estimator. Hence, as a previous step before fitting a parametric model, one should check how many distinguishable groups there are in the data distribution, being these groups identified by the modes of the density. This can be done by exploratory methods or by testing procedures. In both cases, it should be also determined how much of the pattern observed in the density estimator is real, and how much is due to sampling artifacts. In addition, a very flexible and yet simple parametric approximation with several groups/modes can be carried out by fitting mixtures of normals (a revision on this topic can be found in, for example, McLachlan and Peel 2000).

Quite a few contributions have been focused on solving the problem of identifying modes in a data distribution using non-parametric approaches, both from exploratory and testing perspectives. Regarding the exploratory approach, different proposals have been mainly focused on analyzing the behavior of the kernel density estimator along a range of different smoothing (bandwidth) parameters, where an expert eye should try to identify persistent patterns. The mode tree by Minnotte and Scott (1993) and the mode forest (Minnotte, Marchette, and Wegman 1998), as well as the SIgnificant ZERo (SiZer) map by Chaudhuri and Marron (1999) produce graphical displays where the change in the mode pattern of the density estimator can be clearly seen along different bandwidth values.

The aforementioned exploratory tools, although providing a complete analysis of the density estimate from a scale-space perspective (see Chaudhuri and Marron 1999), require a decision on the number of modes to be taken after examining a graphical output. Therefore, conclusions cannot be directly obtained by applying an automatic procedure which indicates how many of the modes observed in the previous representations are really significant. However, this question can be answered by an hypothesis test: $H_0 : \ j = k$ vs. $H_a : \ j > k$, denoting by j the real number of modes in the density and being k a positive integer (so $k = 1$ is a unimodality test). This testing problem has been solved designing test statistics which are based on the critical bandwidth (Silverman 1981; Hall and York 2001; Fisher and Marron 2001) and/or the excess mass (Hartigan and Hartigan 1985; Müller and Sawitzki 1991; Cheng and Hall 1998; Ameijeiras-Alonso, Crujeiras, and Rodríguez-Casal 2019). These procedures will be briefly described in the paper, along with the previous exploratory methods.

Some of the parametric and non-parametric tools for exploring the number of modes on a data distribution are already implemented in other packages available from the Comprehensive R Archive Network (CRAN) repository of R (R Core Team 2021). A brief summary of the capabilities of some packages are provided below, including also some mention to packages devoted to mixture models fitting. Although mode identification and mixture modeling are not equivalent statistical problems, it is worth to briefly discuss also this parametric approach. The aim of the R package presented in this paper, **multimode** (Ameijeiras-Alonso, Crujeiras, and Rodríguez-Casal 2021), is to provide an easy-to-use toolbox with different non-parametric methods for assessing multimodality in real distributions. The methods included in the package facilitate both the exploratory and inferential analysis. The package is available from CRAN at `https://CRAN.R-project.org/package=multimode`.

- **diptest** (Maechler 2016): This package is focused in the *dip* test of Hartigan and Hartigan (1985), which allows for testing unimodality against multimodality.

- **feature** (Duong and Wand 2015): Based on the SiZer map, this package provides some exploratory tools for detecting where the smoothed curve is significantly increasing or decreasing for the 1-dimensional case (with similar ideas to Chaudhuri and Marron 1999), 2-dimensional (Godtliebsen, Marron, and Chaudhuri 2002) and also for the 3- and 4-dimensional cases (Duong, Cowling, Koch, and Wand 2008).

- Mixture modeling: The **mixtools** package (Benaglia, Chauveau, Hunter, and Young 2009) includes different parametric methods based on finite mixture models. Among other functionalities, it allows for testing or exploring the number of components on finite mixture models (McLachlan and Peel 2000, Chapter 6). It performs a parametric bootstrap likelihood ratio test for testing a m-component versus a $(m+1)$-component fit (`boot.comp`) and it computes different information criteria (`multmixmodel.sel`, `repnormmixmodel.sel` and `regmixmodel.sel`) for mixtures of multinomials, multivariate normals and some kinds of regression models. Apart from **mixtools**, in R, there are also other packages for selecting the number of mixture components. Some examples include the **mclust** package (Scrucca, Fop, Murphy, and Raftery 2016) which determines the number of components in a Gaussian mixture model with the functions `mclustModel` (based on the Bayesian Information Criterion) and `mclustBootstrapLRT` (based on a bootstrap likelihood ratio test). Also, the package **mixAK** (Komárek and Komárková 2014) provides, in the function `NMixMCMC`, the reversible jump Markov chain Monte Carlo proposed by Richardson and Green (1997) for selecting the number of components in a normal mixture, given weak prior information.

- **modeest** (Poncet 2019): When knowing that the underlying distribution of the data is unimodal, this package provides different parametric and non-parametric methods for estimating the mode location.

- **modehunt** (Rufibach and Walther 2015): This package implements some non-parametric methods that do not employ the kernel density estimation and, therefore, do not depend on the bandwidth parameter (Dümbgen and Walther 2008; Rufibach and Walther 2010). Based on the ordered sample, the methods provide open intervals, with endpoints at data points, for which the density function f is significantly increasing or decreasing.

- **NPCirc** (Oliveira, Crujeiras, and Rodríguez-Casal 2014b): In this package, the two functions `circsizer.density` and `circsizer.regression` extend the SiZer map to the context of circular data, i.e., samples that can be represented as points on the circumference of a unit circle (Oliveira, Crujeiras, and Rodríguez-Casal 2014a).

There are different combinations of views and goals that must be considered when proceeding with multimodality assessment. First, a parametric or a non-parametric approach can be used. Then, it may be enough with an exploratory tool for determining the number of modes or maybe a formal testing procedure could be required. Finally, it may be crucial also to determine the modes locations.

First, if the parametric approach is chosen, among other packages, **mixtools**, **mclust** and **mixAK** provide different techniques for determining the number of mixture components in this context. When choosing this approach, special care must be taken with the conclusions since the number of modes may be different to the number of components. In particular, when employing a mixture of univariate normals, the number of modes may be less than the number of components. Following a non-parametric perspective, available methods in R are based in the ordered sample (package **modehunt**) or in density smoothing approaches.

As observed in the previous analysis of the different R packages, just a few techniques are available for identifying the number of modes using the kernel density estimation. In particular, if the exploratory way is chosen, package **feature** provides some graphical methods (based on the SiZer map) and package **diptest** the testing approach of Hartigan and Hartigan (1985). The objective of the functions in **multimode** is complementing other implementations on non-parametric multimodality analysis. When referring to other statistical software languages, up to the authors' knowledge, besides the aforementioned non-parametric proposals, just the Silverman (1981) testing approach was already available (see, e.g., `silvtest` function in Stata; Salgado-Ugarte, Shimizu, and Taniuchi 1998).

When focusing on graphical methods, apart from the SiZer, **multimode** provides other exploratory methods, such as the mode tree and the mode forest. Referring to the SiZer map, the main difference with function `SiZer` in the **feature** package is the way of calculating the confidence intervals for the derivative of the smoothed density. While in **feature**, its own approximation is performed, the four proposed methods by Chaudhuri and Marron (1999) (based on normality and bootstrap techniques) for calculating where the smoothed curve is significantly increasing/decreasing are provided in **multimode**. In Figure 1, the differences between both packages can be observed. Note that, for representing the bandwidth values, although **feature** uses a base e instead of the base 10 logarithm (the last one suggested by Chaudhuri and Marron 1999), for comparative purposes, in this case, both are given in $\log_{10}$ scale. The SiZer maps are represented using a sample including the thickness of stamps (introduced in Section 3.1) where at least two modes are expected (see Izenman and Sommer 1988). Modes in SiZer can be detected by blue-red patterns (see Section 2.1). Hence, the SiZer obtained from the **feature** package, Figure 1 (g) (and, also, using the Gaussian approximations in **multimode**, Figure 1 panels (e) and (f)), detects at most just one mode, while more than one mode can be observed in the SiZer maps obtained from **multimode** with bootstrap methods, Figure 1 (h) and (i) (see Section 3.2).

Apart from the unimodality test of Hartigan and Hartigan (1985) (already implemented in **diptest** package), **multimode** includes several proposal for testing the number of modes. Since the dip test presents an extremely conservative behavior (see Ameijeiras-Alonso *et al.* 2019),

the objective here is including other proposals and provide a way of testing a general number of modes.

Finally, when the objective is to estimate the modes locations, the aforementioned graphical tools already provide a way of exploring their locations (depending on the bandwidth parameter). Based on the idea of determining the significant modes for a given bandwidth parameter, Genovese, Perone-Pacifico, Verdinelli, and Wasserman (2016) derived the confidence intervals for the eigenvalues of the Hessian matrix, at modes, of the kernel density estimate. The implementation of their proposal can be found on the personal web page of the second author (`https://sites.google.com/a/uniroma1.it/marcoperonepacifico/R-code`). A semi-parametric approach for estimating modes location has been proposed by Mukhopadhyay (2017), starting with a unimodal parametric density which is modified in order to search if there are extra modes. The R code for implementing this proposal can be found at `https://sites.temple.edu/deepstat/files/2018/10/LPMode_demo.txt`. For a unimodal distribution, package **modeest** includes some (parametric and non-parametric) tools for estimating the mode location. Also, without selecting a bandwidth parameter and when the (general) number of modes is known, package **multimode** provides a (non-parametric) way of estimating the modes (and antimodes) locations.

With the objective of presenting how to tackle the problem of identifying the number and locations of modes and showing the capabilities of the **multimode** package, this paper is organized as follows: in Section 2, some background on both exploratory and testing methods for assessing multimodality will be provided. Initially, the kernel density estimator will be briefly introduced, as it is the key tool for the exploratory and testing methods to be presented. In this section an overview of different graphical tools (namely, the mode tree, the mode forest and the SiZer map) will be provided. Also, different procedures for testing the number of modes are described, including those ones using the critical bandwidth or the excess mass. In Section 3, the reader will find a guided tour across **multimode**, illustrating its use with a real data example. Finally, some discussion will be provided in Section 4, commenting also on the possible extensions of the package.

2. Exploratory and testing tools for assessing multimodality

This section provides a brief background on the design of the different (exploratory and testing) tools included in **multimode**. A key element in the foundations of the different proposals is the kernel density estimator. Given a random sample $(X_1, \ldots, X_n)$ from a random variable X with (unknown) density f, the kernel density estimator for a fixed $x \in \mathbb{R}$ is defined as:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right), \tag{1}$$

where K is the kernel function (usually a symmetric and unimodal density) and $h > 0$ is the smoothing parameter or bandwidth. This parameter controls the smoothness of the estimator in the sense that large (small) values of h provide oversmoothed (undersmoothed) curves. For the particular case of a Gaussian kernel, and focusing on the modes exhibited by $\hat{f}_h$, it should be noted that the number of modes is monotone in h (Silverman 1981). This feature is essential to guarantee the validity of the different proposals.

2.1. Exploratory tools

Since the number of modes in $\hat{f}_h$ is a monotone decreasing function of h, when the Gaussian kernel is used, a simple exploratory solution, for determining the number of modes, is representing this density estimation for different values of h (see Figure 1, panel (a)). In fact, this is the idea underlying some graphical tools, such as the mode tree and the mode forest, where an example of both representations is provided in Figure 1 (panels (b) and (c)).

In the mode tree, Minnotte and Scott (1993) created a tree diagram (similar to the dendrogram) representing, with continuous vertical lines, the modes locations (primary axis) of $\hat{f}_h$ for different bandwidth parameters h (secondary axis). In addition, it represents, with horizontal dashed lines, how each mode splits into more modes as the bandwidth decreases (from top to bottom), showing the relationship between the new modes and the original modes from which they split.

As pointed out by Minnotte *et al.* (1998), the problem of the mode tree is the strong dependence on the available sample. That is the reason why the mode forest is constructed by computing the position of the estimated modes from different mode trees obtained from sampling with replacement the original sample. In order to facilitate the visualization of this exploratory method, the graphical window is divided in different (previously chosen) location-bandwidth (horizontal-vertical axis) pixels. Then, this tool represents the number of times that an estimated mode falls in each (location-bandwidth) pixel shading it proportionally to counts (large counts corresponding to darker pixels and low counts to lighter ones). Then, in the mode forest, modes are identified by dark grey regions.

A problem of the mode tree and the mode forest is that they do not identify which modes are artificially created by atypical data points. An exploratory tool that avoids this issue is the SiZer proposed by Chaudhuri and Marron (1999) and whose representation can be observed in Figure 1 (panels (e), (f), (h) and (i)). SiZer identifies the significant features of the density, by analyzing the behavior of the derivative of the kernel density estimation. For a given location (horizontal axis) and using a specified bandwidth parameter (vertical axis), the SiZer map represents where the smoothed curve, $f_h(x) = \mathbb{E}\left(\hat{f}_h(x)\right)$, is significantly increasing (blue color), decreasing (red) or not significantly different from zero (orchid, a light tone of purple). Thus, for a given bandwidth, a significantly increasing region followed by a significantly decreasing region (blue-red pattern) indicates where a significant peak is present.

For determining the behavior of the smoothed curve, fixing a location x and a bandwidth h, the confidence limits of $f'_h(x)$ are of the form $\mathrm{CI}^{\pm}(x,h) = \hat{f}'_h(x) \pm \mathrm{quantile}(\alpha) \cdot \widehat{\mathrm{sd}}(\hat{f}'_h(x))$, where $\widehat{\mathrm{sd}}$ is the estimated standard deviation and α is the significance level. The estimation of the variance of $\hat{f}'_h(x)$ is obtained in the following way

$$\widehat{\mathrm{Var}}\left(\hat{f}'_h(x)\right) = \frac{1}{nh^4} S^2 \left(K'\left(\frac{x - X_1}{h}\right), \ldots, K'\left(\frac{x - X_n}{h}\right)\right), \tag{2}$$

where S^2 in Equation 2 denotes the sample variance. In order to calculate the quantiles, Chaudhuri and Marron (1999) proposed four approximations: two based on Gaussian methods and two based on bootstrap techniques. The first proposal is based on pointwise Gaussian quantiles (q_1; Figure 1, panel (e)), where quantiles are calculated as $q_1(\alpha) = \Phi^{-1}(1 - \alpha/2)$, being Φ^{-1} the normal quantile function. The second method provides approximate Gaussian quantiles simultaneous over x (q_2; Figure 1, panel (f)) and they are defined as $q_2(\alpha; h) = \Phi^{-1}\left(1 + (1-\alpha)^{1/m(h)}/2\right)$. For each bandwidth, $m(h)$ are obtained from the effective sample

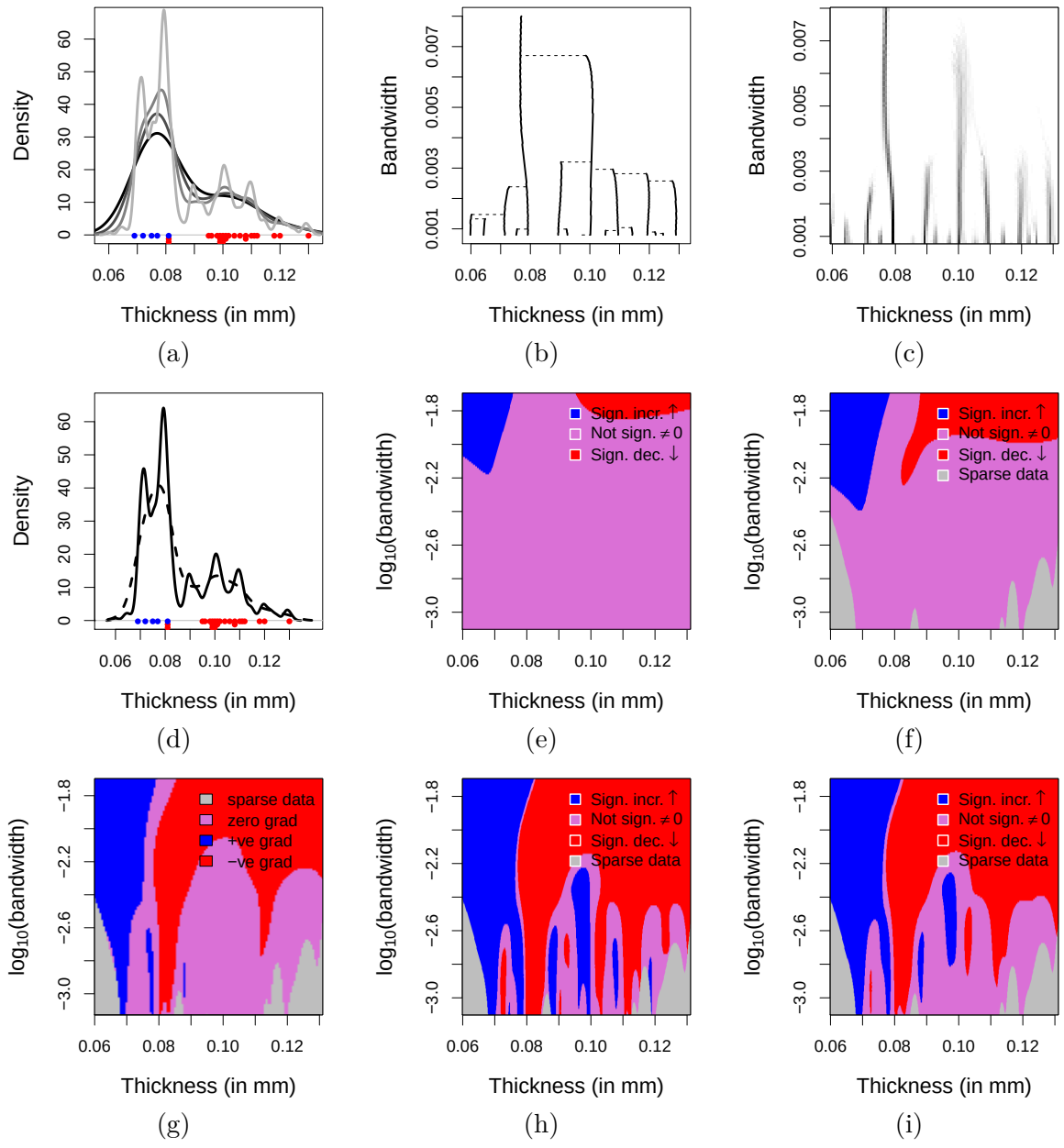

Figure 1: Exploratory analysis for a sample of 485 stamps (1872 Hidalgo Issue of Mexico). In (a) and (d), kernel density estimators with Gaussian kernel and different bandwidths; the points represent stamps watermarked with *LA+-F* (blue) and *Papel sellado* (red). In (a), from dark to light grey, h values: 0.007, 0.005, 0.003 and 0.001. In (d), $h = 0.0039$ (rule of thumb – continuous line –) and $h = 0.0012$ (plug-in rule – dashed line –, see Wand and Jones 1994, Chapter 3). Mode tree (b) and mode forest (c) between the bandwidths $8 \cdot 10^{-4}$ and $8 \cdot 10^{-3}$. For each h, the estimated modes locations are identified by continuous lines in (b) and dark grey pixels in (c). The horizontal discontinuous lines (b) indicate how each mode splits. Panels (e)–(i): SiZer maps between $\log_{10}(h) = -1.7$ ($h = 0.02$) and $\log_{10}(h) = -3.1$ ($h = 8 \cdot 10^{-4}$); given a value of $\log_{10}(h)$, modes can be detected by blue-red patterns. Obtained from **multimode**, using Gaussian, q_1 (e) and q_2 (f), and bootstrap, q_3 (h) and q_4 (i), quantiles. Derived from the results of the **feature** package (g).

size (ESS, see Chaudhuri and Marron 1999) in the following way

$$m(h) = \frac{n}{\overline{\mathrm{ESS}}(x,h)}, \quad \text{being } \mathrm{ESS}(x,h) = \frac{\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right)}{K(0)} \tag{3}$$

and $\overline{\mathrm{ESS}}(x,h)$ the average mean over x of the values of $\mathrm{ESS}(x,h)$. Small values of ESS provide an indicative of areas with too sparse data for meaningful inference. For that reason, in the methods employing the ESS (q_2, q_3 and q_4), the significant features are just represented in the regions satisfying $x \in D_h = \{x : \mathrm{ESS}(x,h) \geq n_0\}$ (remaining regions are marked with grey color; see Figure 1, panels (f), (h) and (i)). Then, the parameter n_0 (where Chaudhuri and Marron 1999 proposed to use $n_0 = 5$) can help to remove the spurious modes created by atypical data points.

The two bootstrap quantiles are calculated from the following values

$$Z(x,h)^{*b} = \frac{\hat{f}_h'(x)^{*b} - \hat{f}_h'(x)}{\widehat{\mathrm{sd}}(\hat{f}_h'(x))}, \text{ with } b = 1, \ldots, B, \tag{4}$$

where each $\hat{f}_h'(x)^{*b}$ is calculated from a random sample generated drawn with replacement from the original sample. The third approach is a bootstrap quantile simultaneous over x, $q_3(\alpha; h)$ (Figure 1, panel (h)), and it is calculated with the empirical quantile $(1-\alpha/2)$ of the B values $\max_{x \in D_h} |Z(x,h)^{*b}|$; with $b = 1, \ldots, B$. Finally, the fourth approach, also calculated from the quantities defined in Equation 4, is the bootstrap quantile simultaneous over x and h, $q_4(\alpha)$ (Figure 1, panel (i)), and it is defined as the empirical quantile $(1-\alpha/2)$ of the B values $\max_h \max_{x \in D_h} |Z(x,h)^*|$; with $b = 1, \ldots, B$.

2.2. Testing procedures

Consider the testing problem presented in the Introduction. That is, given a sample $X_1, \ldots, X_n$ from a random variable X with unknown density f with j modes, and given a positive integer k, the goal is to test $H_0 : j = k$ vs. $H_a;\ j > k$. The testing methods, briefly described in this section and included in **multimode**, make use of one or both of the following concepts: the critical bandwidth and the excess mass.

2.3. Using a critical bandwidth

The critical bandwidth for a fixed k was defined by Silverman (1981) as the smallest bandwidth such that the kernel density estimator in Equation 1 has at most k modes:

$$h_k = \inf\{h : \hat{f}_h \text{ has at most } k \text{ modes}\}. \tag{5}$$

This value can be used as a test statistic, as long as (1) is constructed with a Gaussian kernel, as proposed by Silverman (1981): H_0 is rejected for large values of h_k. For calibrating h_k, a bootstrap algorithm is employed, where the resamples $Y_i^{*b} = (1 + h_k^2/\hat{\sigma}^2)^{-1/2} X_i^{*b}$ (with $i \in \{1, \ldots, n\}$, being n the sample size) are calculated from B bootstrap samples X_i^{*b} generated from $\hat{f}_{h_k}$, being $\hat{\sigma}^2$ the sample variance and with $b \in \{1, \ldots, B\}$. Hall and York (2001) proved that this bootstrap algorithm is not consistent and the authors suggested a correction for the unimodality test (for $k = 1$), when f has a bounded support or when the mode is located in a given closed interval I, defining the critical bandwidth as:

$$h_{\mathrm{HY}} = \inf\{h : \hat{f}_h \text{ has exactly one mode in } I\}. \tag{6}$$

The authors also proposed using h_{HY} as a test statistic and designed a bootstrap algorithm in this simplified scenario. However, the critical bandwidths for the bootstrap samples h^*_{HY}, calculated from X^*, are smaller than h_{HY}, so for an α-level test, a correction factor λ_α to empirically approximate the p value $\mathbb{P}(h^*_{\text{HY}} \leq \lambda_\alpha h_{\text{HY}} \mid \mathcal{X}) \geq 1 - \alpha$ must be considered. Two different methods were suggested for computing this λ_α factor (see Hall and York 2001, for details). The first one is based on a polynomial approximation where after imposing a significance level α, the correction factor λ_α is approximated with the following expression:

$$\lambda_\alpha = \frac{0.94029\alpha^3 - 1.59914\alpha^2 + 0.17695\alpha + 0.48971}{\alpha^3 - 1.77793\alpha^2 + 0.36162\alpha + 0.42423}. \tag{7}$$

The second one uses Monte Carlo techniques considering a simple unimodal distribution. In particular, Hall and York (2001) suggest to generate the resamples (of same sample size as the original data) obtained from a unimodal distribution resembling the sampled one and they claim that, in practice, normal distribution produce a good level accuracy.

Hall and York (2001) method should not be used in the general case of testing k-modality as the bootstrap test cannot be directly calibrated under this hypothesis, since it depends on the unknown quantities $f^{1/5}(t_i)/|f''(t_i)|^{2/5}$, where t_i are the ordered turning points of f, with $i = 1, \ldots, (2k-1)$.

As showed in Ameijeiras-Alonso *et al.* (2019), the critical bandwidth of Hall and York (2001) or Silverman (1981), when f has a bounded support, also plays a relevant role when the goal is to estimate the modes locations. When the true number of modes is known, under some general assumptions, the kernel density estimation with the critical bandwidth provides a good estimation of the modes and antimodes locations.

A distribution estimation using the critical bandwidth of Silverman (1981) is also employed by Fisher and Marron (2001), who considered the following Cramér-von Mises test statistic for testing k-modality,

$$T_k = \sum_{i=1}^{n} \left(\hat{F}_{h_k}(X_{(i)}) - \frac{2i-1}{2n} \right)^2 + \frac{1}{12n}, \tag{8}$$

being $\hat{F}_{h_k}(x) = \int_{-\infty}^{x} \hat{f}_{h_k}(t)dt$. H_0 is rejected for large values of T_k (8), whose distribution is approximated by a bootstrap algorithm, where resamples are generated from $\hat{f}_{h_k}$.

2.4. Using an excess mass statistic

The identification of a mode in a density estimate by finding a significant excess mass is the basic idea in the proposals by Müller and Sawitzki (1991), Cheng and Hall (1998) and Ameijeiras-Alonso *et al.* (2019). The empirical excess mass for k modes and a constant λ is defined as:

$$E_{n,k}(\mathbb{P}_n, \lambda) = \sup_{C_1(\lambda), \ldots, C_k(\lambda)} \left\{ \sum_{m=1}^{k} \mathbb{P}_n(C_m(\lambda)) - \lambda \|C_m(\lambda)\| \right\}, \tag{9}$$

where the supremum is taken over all families $\{C_m(\lambda) : m = 1, \cdots, k\}$ of closed intervals with endpoints at data points. $\|C_m(\lambda)\|$ denotes the measure of $C_m(\lambda)$ and $\mathbb{P}_n(C_m(\lambda)) = (1/n)\sum_{i=1}^{n} \mathcal{I}(X_i \in C_m(\lambda))$, where $\mathcal{I}$ is the indicator function. The difference $D_{n,k+1}(\lambda) = E_{n,k+1}(\mathbb{P}_n, \lambda) - E_{n,k}(\mathbb{P}_n, \lambda)$ measures the plausibility of the null hypothesis, that is, large values of $D_{n,k+1}(\lambda)$ would indicate that H_0 is false. An example of the theoretical excess

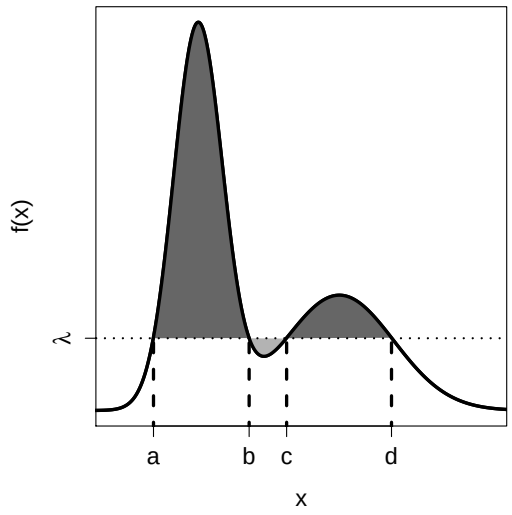

Figure 2: The excess mass for k modes is the largest probability mass, exceeding a given level λ, when taking k intervals. In this case, the excess mass for two modes is equal to the dark grey area (and obtained with the union of the intervals $[a, b]$ and $[c, d]$) and for one mode is equal to the dark grey minus the light grey area (and obtained with $[a, d]$). Then, in terms of excess mass, for the represented value of λ, the difference between assuming bimodality and unimodality is the light grey area.

mass difference for a bimodal density is shown in Figure 2 for illustrative purposes. Using these differences, Müller and Sawitzki (1991) defined as the excess mass statistic for testing $H_0 : \ j = k$,

$$\Delta_{n,k+1} = \max_{\lambda}\{D_{n,k+1}(\lambda)\}. \tag{10}$$

Their proposal for testing unimodality is to calibrate this test statistic using a Monte Carlo calibration, where resamples are generated from the uniform distribution. The same approach was already proposed by Hartigan and Hartigan (1985) with the dip unimodality test, since both quantities (dip and excess mass) coincide up to a factor.

The performance in practice of the calibration algorithm proposed for (10) was remarkably conservative and Cheng and Hall (1998) designed a bootstrap procedure for approximating the distribution of $\Delta_{n,2}$ under the hypothesis of unimodality generating the resamples from a family of parametric functions, guaranteeing an asymptotic correct behavior. When analyzing Cheng and Hall (1998) proposal in simulated scenarios (see Ameijeiras-Alonso *et al.* 2019), the calibration of the test was not satisfactory in the "complicated" unimodal models (such as in asymmetric scenarios, for a moderated sample size) due the lack of flexibility of this parametric approach. Also, extending this test to the general case of testing k-modality is not an easy task. For those reasons, a completely non-parametric alternative for testing $H_0 : \ j = k$ vs. $H_a : \ j > k$ has been proposed by Ameijeiras-Alonso *et al.* (2019). Their method consist in calibrating the excess mass statistic given in (10) using a bootstrap procedure, where the resamples are generated from (a modified version of) $\hat{f}_{h_k}$. The modification of the kernel density estimator ensures the correct calibration of this test, under some regularity conditions (similar to those ones needed in Cheng and Hall 1998). Although, in general, the Ameijeiras-Alonso *et al.* (2019) proposal presents a correct behavior even when the sample size is "small"

Data set	Description
`acidity`	Acid-neutralizing capacity
`chondrite`	Percentage of silica in chondrite meteors
`enzyme`	Blood enzymatic activity
`galaxy`	Velocities of galaxies
`geyser`	Waiting time between geyser eruptions
`stamps`	Stamps thickness
Function	**Description**
`critbw`	Critical bandwidth computation
`excessmass`	Excess mass statistic
`locmodes`	Location of modes and antimodes
`modeforest`	Mode forest
`modetest`	Test for the number of modes
`modetree`	Mode tree
`nmodes`	Number of modes
`sizer`	SiZer map

Table 1: Summary of **multimode** package contents.

($n = 50$), when knowing the compact support I where the modes and antimodes lie, the Hall and York (2001) bandwidth can be employed (for generating the resamples), improving the results of this test.

When deciding which proposal should be chosen, it must be considered that an asymptotic correct behavior is just expected in the unimodality tests of Hall and York (2001) (when f has a bounded support or when employing the compact support I) and Cheng and Hall (1998) and in the multimodality test of Ameijeiras-Alonso *et al.* (2019). A complete simulation study comparing all the aforementioned proposals is provided in Ameijeiras-Alonso *et al.* (2019), showing that the other proposals (Silverman 1981; Fisher and Marron 2001) for testing $H_0 : \ j = k$ vs. $H_a : \ j > k$, when $k > 1$, exhibit an unsatisfactory behavior.

3. Using multimode

A complete description of the **multimode** package capabilities is provided in this section. Specifically, the package includes the data sets and the functions shown in Table 1. First, the different data sets available in the package will be described. Second, the usage of different functions for exploring the number of modes will be illustrated. Finally, the functions for testing multimodality and estimating the location of modes and antimodes will be introduced.

3.1. Data description

The package **multimode** includes some classical data sets for which determining the number of different groups in the sample and/or exploring the location of modes and antimodes are relevant issues. The first data set, `acidity`, analyzed by Crawford (1994), contains, on the log scale, the acid-neutralizing capacity (ANC) measured in a sample of 155 lakes in North-Central Wisconsin (USA). ANC describes the capability of a lake to absorb acid, where low ANC values may lead to a loss of biological resources. The data set `chondrite`,

included in Table 2 of Good and Gaskins (1980), gathers the percentage silica (in %) in 22 chondrite meteors. The data set `enzyme`, introduced by Bechtel, Bonaiti-Pellie, Poisson, Magnette, and Bechtel (1993), collects a sample with the distribution of enzymatic activity in the blood, for an enzyme involved in the metabolism of carcinogenic substances. The data set `galaxy` provides the velocities in km/sec of different galaxies (diverging away from our own galaxy) from the unfilled survey of the Corona Borealis region. In this data set introduced by Postman, Huchra, and Geller (1986) and further studied by Roeder (1990), multimodality is an evidence for voids and superclusters in the universe. The data set `geyser` presents the interval times between the starts of the geyser eruptions observed during different periods in the Old Faithful Geyser in Yellowstone National Park, Wyoming, USA. The included periods are: October 1980, obtained from Table 3 of Härdle (2012) and the supplementary material of Weisberg (2005); and August 1985, from Table 1 in Azzalini and Bowman (1990). Finally, the data set `stamps`, analyzed in Izenman and Sommer (1988), consists of thickness measurements (in millimeters) of 485 unwatermarked used white wove stamps of the 1872 Hidalgo stamp issue of Mexico. All of them had an overprint with the year (1872 or either an 1873 or 1874) and some of them were watermarked (*Papel Sellado* or *LA+-F*), being this information also included inside `stamps`. Since the stamps value depends on its scarcity, it is of importance to determine the number of available groups in a particular stamp issue. For this particular stamp issue, although the watermark is some stamps (in 29 of 485) helps to conclude that there are at least two groups, the question about the number of groups can be answered analyzing the underlying number of modes.

Some of these data sets (`acidity`, `enzyme`, `galaxy` or `stamps`) were used in the statistical literature for illustrating mixtures of parametric models (see McLachlan and Peel 2000; Richardson and Green 1997). For these cases, the non-parametric (both exploratory and inferential) tools included in **multimode** could be seen as a preliminary tool for determining the number of modes. In other data sets (`chondrite`, `geyser` or `stamps`), testing or exploring the number of modes is an important problem per se. Some examples of their application can be found in Chaudhuri and Marron (1999), Müller and Sawitzki (1991) or Scott (2015, Section 9.2). In the subsequent sections, the `stamps` data set will be used for illustrating the functions available in the **multimode** package.

3.2. Exploring data with multimode

When the objective is to explore the number of modes in a sample, a simple solution might be to observe the number of peaks in the kernel density estimation for different values of h. In order to facilitate this task, using the Gaussian kernel and a given bandwidth parameter `bw`, the function `nmodes` computes the estimated number of modes in the real line or in a support bounded by `lowsup` and `uppsup`. This kernel density estimation is calculated in `n` equally spaced points of the variable for computational reasons (as in the `density` function from the **stats** package). For instance, using the code below, it can be seen that the estimated number of modes using the rule of thumb and the plug-in rule (`bw.nrd0` and `bw.SJ` from the **stats** package and illustrated in Figure 1, panel (d) is, respectively, two and nine.

```
R> library("multimode")
R> data("stamps", package = "multimode")
R> bwRT <- bw.nrd0(stamps)
R> bwRT
```

```
[1] 0.003909682

R> bwPI <- bw.SJ(stamps)
R> bwPI

[1] 0.001205108

R> nmodes(data = stamps, bw = bwRT)

[1] 2

R> nmodes(data = stamps, bw = bwPI, full.result = TRUE)

n= 485. Number of modes: 9
Bandwidth: 0.001205108
Support where the number of modes are computed
     -Inf      Inf
```

The function allows also for a `full.result` argument, whose default value `FALSE`, returns just the number of modes. If this argument is `TRUE`, it returns an object of class '`estmod`', which is a list (that can be printed) containing: the number of modes for the provided bandwidth (`nmodes`), the number of non-missing observations (`sample.size`), the employed bandwidth (`bw`), the support where the number of modes are computed (`lowsup` and `uppsup`), the x coordinates of the points where the density is estimated (`fnx`) and their estimated density values (`fny`).

In general, the different functions in **multimode** were implemented in such a way that just the minimum arguments are required. In this case, just `data` and `bw` are needed to obtain a result (`lowsup = -Inf`, `uppsup = Inf`, `n = 2^15` and `full.result = FALSE` are given by default).

Based on the idea of exploring the number of modes (and their location) for different values of h, the three different graphical tools, presented in Section 2.1, have been implemented in **multimode**: `modetree`, `modeforest` and `sizer`. The outputs from these exploratory functions (an object of class '`gtmod`') and the arguments used for their computation are detailed below. The common characteristics, in the three of them, are: the exploratory features will be calculated in a finite number of grid points (the common argument is the first element of `gridsize`); the number of modes will be determined according to a value of h and the employed bandwidth values can be chosen by the practitioner (`bws`, `cbw1`, `cbw2` and the second element `gridsize`); a graphical display can be generated (or added to the current graphic) with different plot arguments; an output related with the modes locations is returned.

The different exploratory tools (`modetree`, `modeforest` and `sizer`) include three options for providing the bandwidths. The first one is to use a range of bandwidth parameters in the argument `bws` and the exploratory tool is computed in a grid of h between the given values and size equal to the second element of the argument `gridsize`. By default, a regular grid of size 151 is computed between a lower bandwidth equal to twice the distance between the grid points used for estimating the density and upper bandwidth equal to the data range.

The second option considers the critical bandwidths for `cbw1` and `cbw2` modes as the range of bandwidths. The third method allows to include a vector of bandwidths in the argument `bws` with size greater than two. In order to distinguish between this and the first option a `warning` message is displayed when a grid of bandwidths is computed between the two given values. Then, these exploratory tools are represented (using a $\log_{10}$ scale for the bandwidths if `logbw = TRUE`, this is the value by default just in the `sizer`) when the argument `display` is `TRUE` (by default) or when plotting the output with the titles in the x and y axis provided by `xlab` and `ylab`, as usual. As indicated before, unless the grid of bandwidth values is given by the user, a regular grid over the bandwidths scale is employed. When the logical argument `logbw.regulargrid` is `TRUE`, it allows to create a regular grid over the $\log_{10}$ bandwidths scale (default is `FALSE`).

The mode tree introduced by Minnotte and Scott (1993) and implemented in the function `modetree`, shows with continuous lines the estimated mode locations for each bandwidth. For `modetree`, the first element of `gridsize` is equal to the number of equally spaced points at which the density is to be estimated (by default, `512`). Moreover, the mode tree can be added to another plot when the argument `addplot` is `TRUE` (`FALSE`, by default). Also, the color lines in the mode tree can be chosen with the argument `col.lines` (by default are `"black"`). Below, an example with the code lines for computing and plotting the mode tree for the `stamps` data set between the bandwidths $8 \cdot 10^{-4}$ and $8 \cdot 10^{-3}$ is shown (its representation appears in Figure 3, left).

```
R> mtstamps <- modetree(data = stamps, bws = c(0.0008, 0.008),
+    display = FALSE)

Warning message:
In modetree(data = stamps, bws = c(8e-04, 0.008), display = FALSE) :
  A grid of 'bws' between the given ones were used

R> mtstamps

Call:
         modetree(data = stamps, bws = c(8e-04, 0.008), display = FALSE)

n=485. Location values range:     0.06    0.131
Number of employed bandwidths: 151. log10 bandwidths scale: FALSE
Bandwidths range:   0.0008    0.008

R> plot(mtstamps, xlab = "Thickness (in mm)", ylab = "")
R> summary(mtstamps)

The estimated modes are located between the following values
(for the bandwidths range indicated in parenthesis)

Mode 1:  0.0765159 | 0.07923317 (0.0008 0.008)
Mode 2:  0.09865606 | 0.1010608 (0.0008 0.006704)
Mode 3:  0.08932064 | 0.09054959 (0.0008 0.0032)
```

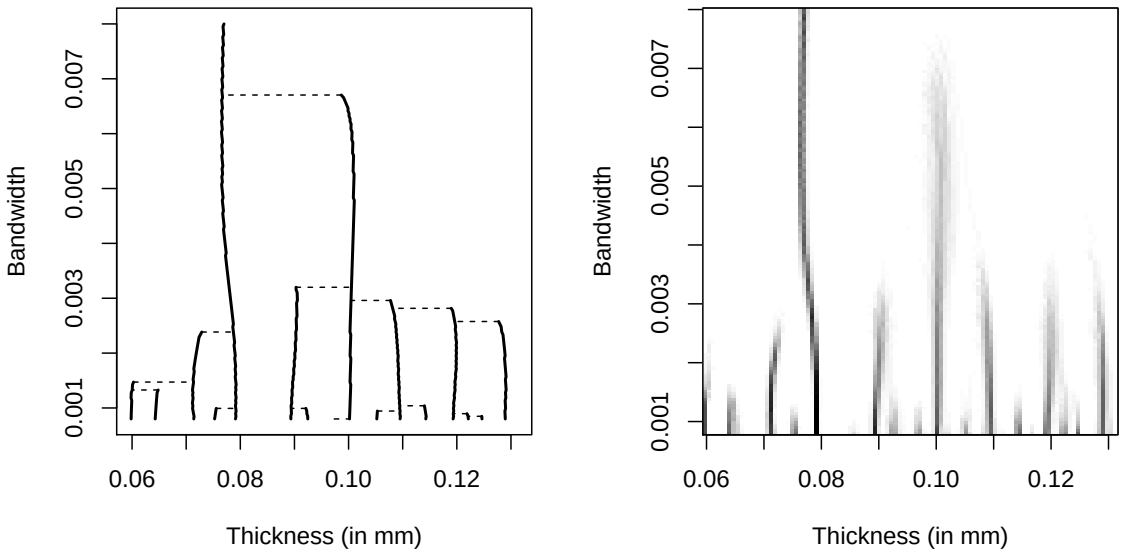

Figure 3: Mode tree (left) and forest (right) for the 485 stamps data set from the 1872 Hidalgo Issue of Mexico. Bandwidth range from $8 \cdot 10^{-4}$ to $8 \cdot 10^{-3}$. For each bandwidth value, modes location are identified by continuous lines and dark grey pixels, for the mode tree and forest, respectively. The horizontal discontinuous lines in the left panel show how each mode splits.

```
Mode 4:  0.1077458 | 0.1095711 (0.0008 0.00296)
Mode 5:  0.1189791 | 0.1199551 (0.0008 0.002816)
Mode 6:  0.1277307 | 0.1291116 (0.0008 0.002576)
Mode 7:  0.07114923 | 0.07288025 (0.0008 0.002384)
Mode 8:  0.0598106 | 0.06027081 (0.0008 0.001472)
Mode 9:  0.06427515 | 0.06482456 (0.0008 0.001328)
Mode 10:  0.1140164 | 0.1143359 (0.0008 0.00104)
Mode 11:  0.07525205 | 0.07554669 (0.0008 0.000992)
Mode 12:  0.0921117 | 0.0924591 (0.0008 0.000992)
Mode 13:  0.105216 | 0.1053268 (0.0008 0.000944)
Mode 14:  0.1218804 | 0.1221264 (0.0008 0.000896)
Mode 15:  0.12461 | 0.1246481 (0.0008 0.000848)
Mode 16:  0.09735421 | 0.09735421 (0.0008 0.0008)
```

The output is an object of class '`gtmod`' containing the following components: `locations`, a matrix with the estimated modes locations for each bandwidth; `range.x`, the data range; `range.bws`, the bandwidths employed for computing the mode tree; `logbw`, a logical value indicating if the bandwidths are given in the $\log_{10}$ scale; `sample.size`, the number of non-missing observations in the sample used for obtaining the mode tree; and `call`, the named function applied to the given arguments. With the exception of `locations`, these elements are common to all elements of class '`gtmod`', which can be printed, plotted and summarized. The plot and the summary can be useful for exploring where the different modes are located when the number of modes is not clear and further insight on the data distribution is required. In this case, the principal mode appears between the values 0.0765 and 0.0792, the secondary mode between 0.0987 and 0.1011, and so forth.

The mode forest, introduced by Minnotte *et al.* (1998), is provided by `modeforest`. This graphical tool is generated by looking simultaneously at a collection of mode trees generated by the original sample and `B` (`B = 99` by default) random resamples drawn with replacement from the original one.

For the `modeforest` and `sizer`, the first element of `gridsize` is equal to the number of grid points in the horizontal (values of the variable) axis (by default equal to 100 in the `modeforest` and 512 in the `sizer`). In both cases, the horizontal plotted values are bounded by the interval (`from`, `to`), being this interval equal to the data range by default. In the `modeforest` function, the number of equally spaced points at which the density is to be estimated is chosen by the argument `n` (`n = 512` by default). The mode forest for the `stamps` data set between the bandwidths $8 \cdot 10^{-4}$ and $8 \cdot 10^{-3}$ (represented in Figure 3, right) can be obtained as follows:

```
R> mfstamps <- modeforest(data = stamps, bws = c(0.0008, 0.008),
+    xlab = "Thickness (in mm)", ylab = "")

Warning message:
In modeforest(data = stamps, bws = c(8e-04, 0.008),  :
  A grid of 'bws' between the given ones were used

R> summary(mfstamps, levelmf = 0.4)

The estimated modes are located between the following values
(for the bandwidths range indicated in parenthesis)

Mode 1:  0.07598509 | 0.07959896 (0.0008 0.008)
Mode 2:  0.09983665 | 0.1005594 (0.0008 0.003056)
Mode 3:  0.1085099 | 0.1099555 (0.0008 0.002672)
Mode 4:  0.08899503 | 0.09044058 (0.0008 0.00248)
Mode 5:  0.07092566 | 0.07237121 (0.0008 0.00224)
Mode 6:  0.1186288 | 0.1200743 (0.0008 0.001568)
Mode 7:  0.05936127 | 0.06008404 (0.0008 0.001376)
Mode 8:  0.06369792 | 0.06442069 (0.0008 0.001184)
Mode 9:  0.07526231 | 0.07598509 (0.0008 0.000992)
Mode 10:  0.1142921 | 0.1150149 (0.0008 0.000944)
Mode 11:  0.124411 | 0.1251338 (0.0008 0.000944)
Mode 12:  0.0926089 | 0.09333168 (0.0008 0.0008)
```

This function returns an object of class '`gtmod`' containing the same components as `modetree` with the exception of `locations` which is replaced by the matrix `modeforest` including the percentage of times that an estimated mode falls in each location-bandwidth pixel. In the case of the functions `modeforest` and `sizer`, in `range.x`, they return the employed location values to represent the mode forest or the SiZer map. In the `modeforest` plot, modes can be detected observing the dark grey vertical traces, but one should be careful with the very dark areas (as the one next to 0.06) since, due to the resampling algorithm, it is possible that spurious modes (created by some atypical data points) may seem visually more prominent

than real modes (as pointed out by Minnotte *et al.* 1998). Observing Figure 3 (right), the mode forest suggests at least nine modes for the `stamps` data set. Although the graphical tool is more informative, this function also has an associated summary that tries to provide the estimated location of the modes by seeing in which pixels the percentage given in `modeforest` is greater than the value `levelmf` (default equal to 0.5).

With the `sizer` function the assessment of SIgnificant ZERo crossing of the derivative of the smoothed curve is computed for a given sample. In each location-bandwidth pixel, the SiZer map shows the significant features of the smoothed curve using, by default, the colors described in Section 2.1, but they can be replaced using the `col.sizer` argument. For analyzing the behavior of the curve, the four quantile approximations proposed by Chaudhuri and Marron (1999) are implemented in the `sizer` function using the argument `method`. The available quantiles are: the pointwise Gaussian quantiles (q_1), when `method = 1`; approximate simultaneous over location x Gaussian quantiles (q_2), when `method = 2` (value by default); bootstrap quantile simultaneous over location x (q_3), when `method = 3`; and bootstrap quantile simultaneous over (location and bandwidth) x and h (q_4), when `method = 4`. Bootstrap quantiles q_3 and q_4 are computed generating `B` (`B = 100` by default) random samples drawn with replacement from the sample. The significance level α, is chosen with the argument `alpha` (equal to 0.05 by default). In methods q_2, q_3 and q_4; grey color (by default) is employed when the effective sample size in (3) is less than the value `n0` (`n0 = 5` by default, as suggested by Chaudhuri and Marron 1999). A legend indicating the meaning of the different colors is also provided in the plot position given in `poslegend` when the argument `addlegend` is `TRUE` (by default the legend is included and shown at the `topright` position). The different SiZer maps for the `stamps` data set between the bandwidths $8 \cdot 10^{-4}$ and 0.02 (represented in Figure 4) can be obtained as shown below.

```
R> szst1 <- sizer(data = stamps, method = 1, bws = c(0.0008, 0.02),
+    logbw.regulargrid = TRUE, xlab = "Thickness (in mm)", ylab = "")

Warning message:
In sizer(data = stamps, method = 1, bws = c(8e-04, 0.02),  :
  A grid of 'bws' between the given ones were used

R> summary(szst1)

The estimated modes are located between the following values
(for the log10 bandwidths range indicated in parenthesis)

Mode 1:  0.07382485 | 0.1049481 (-1.848084 -1.69897)

R> szst2 <- sizer(data = stamps, bws = c(0.0008, 0.02),
+    logbw.regulargrid = TRUE, xlab = "Thickness (in mm)", ylab = "")

Warning message:
In sizer(data = stamps, bws = c(8e-04, 0.02),  :
  A grid of 'bws' between the given ones were used

R> summary(szst2)
```

```
The estimated modes are located between the following values
(for the log10 bandwidths range indicated in parenthesis)

Mode 1:  0.07201859 | 0.08799706 (-2.239507 -1.69897)
```

```
R> szst3 <- sizer(data = stamps, method = 3, bws = c(0.0008, 0.02), B = 500,
+    logbw.regulargrid = TRUE, xlab = "Thickness (in mm)", ylab = "")

Warning message:
In sizer(data = stamps, method = 3, bws = c(8e-04, 0.02), B = 500,  :
  A grid of 'bws' between the given ones were used
```

```
R> summary(szst3)

The estimated modes are located between the following values
(for the log10 bandwidths range indicated in parenthesis)

Mode 1:  0.07632583 | 0.08271722 (-3.09691 -1.69897)
Mode 2:  0.09813992 | 0.1030029 (-3.09691 -2.220868)
Mode 3:  0.08910861 | 0.09272114 (-3.09691 -2.62161)
Mode 4:  0.107727 | 0.1103669 (-3.059632 -2.649569)
Mode 5:  0.07035127 | 0.07271331 (-3.09691 -2.714806)
```

```
R> szst4 <- sizer(data = stamps, method = 4, bws = c(0.0008, 0.02), B = 500,
+    logbw.regulargrid = TRUE, xlab = "Thickness (in mm)", ylab = "")

Warning message:
In sizer(data = stamps, method = 4, bws = c(8e-04, 0.02), B = 500,  :
  A grid of 'bws' between the given ones were used
```

```
R> summary(szst4)

The estimated modes are located between the following values
(for the log10 bandwidths range indicated in parenthesis)

Mode 1:  0.07618689 | 0.08285616 (-3.09691 -1.69897)
Mode 2:  0.08869178 | 0.1117564 (-2.919838 -2.267466)
Mode 3:  0.07007339 | 0.07257436 (-3.059632 -2.8546)
```

Apart from the already described arguments, `sizer` returns an object of class '`gtmod`' with the element `method` (a number indicating what type of quantile was performed) and five matrices containing different information in each location-bandwidth pixel: `sizer`, with the significant behaviors of the smoothed curve in each location-bandwidth pixel (1: significantly decreasing, 2: not significantly different from zero, 3: significantly increasing or 4: low data for meaningful inference); `lower.CI` with the lower limits of the confidence interval, $\mathrm{CI}^-(x,h)$; `estimate`, with the derivative values of the kernel density estimation, $\hat{f}'_h(x)$; `upper.CI` with the upper limits of the confidence interval, $\mathrm{CI}^+(x,h)$; and `ESS`, with the effective sample size.

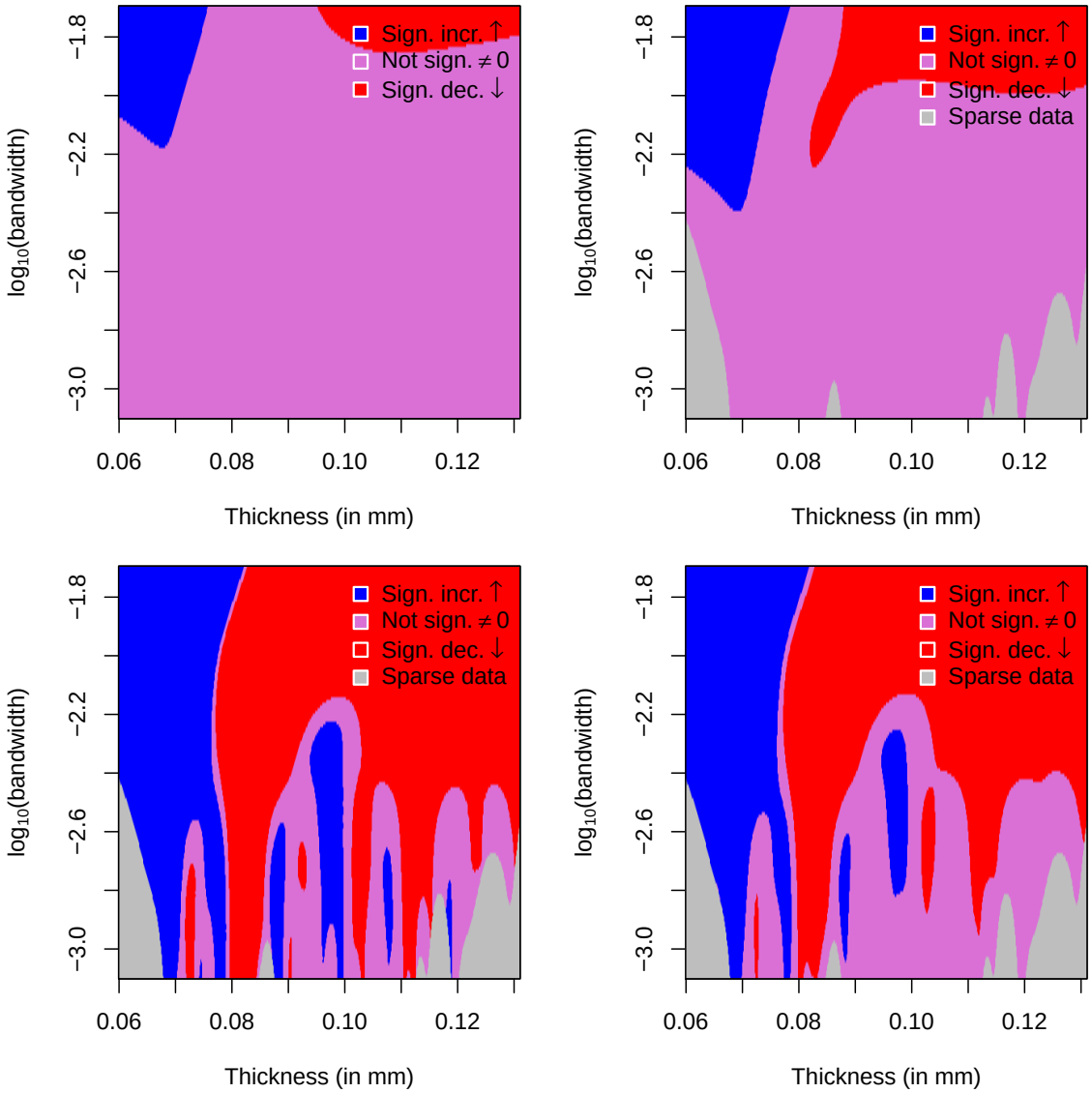

Figure 4: SiZer maps for the 485 stamps data set from the 1872 Hidalgo Issue of Mexico. Bandwidth range from $\log_{10}(h) = -3.1$ ($h = 8 \cdot 10^{-4}$) to $\log_{10}(h) = -1.7$ ($h = 0.02$). Given a value of $\log_{10}(h)$, modes can be detected in the blue-red patterns. Top: Gaussian quantiles q_1 (left) and q_2 (right). Bottom: bootstrap quantiles q_3 (left) and q_4 (right).

As noted before, in the SiZer maps (represented in Figure 4), by default, blue color indicates locations where, for a given bandwidth, the smoothed curve is significantly increasing, red color shows where it is significantly decreasing and orchid indicates where it is not significantly different from zero. Then, focusing on $\log_{10}(h)$ values, modes can be detected by blue-red patterns. In this case, the SiZer maps computed with Gaussian quantiles just detect, at most, one mode around 0.08. The conclusion with those ones constructed with bootstrap confidence intervals vary with the bandwidth. For all the bandwidth values, both methods capture a principal mode close to 0.08 and for several bandwidth parameters is also detected a secondary

mode around 0.10. The third and the fourth mode (around 0.09 and 0.11) that appears in the mode tree (Figure 3, left) are only significant modes for some bandwidth parameters for q_3. Finally, both methods, q_3 and q_4, detect another mode near 0.072 for some bandwidth values. Then, depending on the bandwidth parameter, the conclusion using the quantile q_3 is that there are between one and five modes (in order of appearance, around 0.08, 0.10, 0.09, 0.11 and 0.07), while q_4 detects between one and three modes (around 0.08, 0.10 and 0.07). As in the mode forest, the summary of the '`gtmod`' object returned by `sizer` tries to summarize the information about the estimated mode location by detecting when an increasing pixel is followed by a decreasing pixel for each bandwidth value.

3.3. Testing and locating modes with multimode

The **multimode** package has implemented all the test presented in the Section 2.2. In particular, it allows to compute the critical bandwidth of Silverman (1981) and Hall and York (2001) (with the function `bw.crit`) and the excess mass of Müller and Sawitzki (1991) (with `excessmass`). Their associated p values can be also obtained, with `modetest`, using different testing proposals. For the three functions (`bw.crit`, `excessmass` and `modetest`), the investigated number of modes can be specified in the argument `mod0` (unimodality, `mod0 = 1`, is explored by default).

For `bw.crit` and for the testing proposals using the critical bandwidth in `modetest`, when the compact support is unknown, the critical bandwidth introduced by Silverman (1981) is computed and if the finite values of the support limits are provided (via arguments `lowsup` and `uppsup`) the one proposed by Hall and York (2001) is calculated. Although the whole real line is employed by default, both arguments should be used in `modetest` when employing the Hall and York (2001) proposal or for computing the new proposal when the compact support is known (see Section 2.4). As in the `nmodes` function, the number of equally spaced points at which the density is to be estimated can be chosen by the argument `n` and a full result list of class '`estmod`' is returned when `full.result = TRUE` (otherwise, the output is just the value of the critical bandwidth). Since a dichotomy method is employed for computing the critical bandwidth, the parameter `tol` (equal to 10^{-5} by default) is used to determine a stopping time in such a way that the error committed in the computation of the critical bandwidth is less than `tol`.

For `excessmass` and in the testing proposals using the excess mass in `modetest`, when there are repeated data in the sample or the distance between different pairs of data points shows ties, a data perturbation is applied. This modification is made in order to avoid the induced discretization of the data which has important effects on the computation of this test statistic. A `warning` message is displayed if this data perturbation is done. The perturbed sample is obtained by adding a sample from the uniform distribution in the support minus/plus a half of the minimum of the positive distances between two sample points. In this case, when `full.result = TRUE`, apart from the value of the excess mass test statistic (`excess.mass`), the object of class '`estmod`' contains the arguments `nmodes` and `sample.size`, as in the function `nmodes`, but also a logical value indicating if the excess mass was approximated (`approximate`).

Since the excess mass for one mode is twice the dip, this equality can be used for a "fast" computation of the excess mass for one mode. When `mod0` is greater than one and the sample size is "large", a two-step approximation (when `approximate = TRUE`, being this argument

FALSE by default) can be performed in order to improve the computational time efficiency. This two-step approximation is achieved creating two grids of values of size the elements in gridsize (by default, both equal to 20). First, since the possible λ candidates for maximizing $D_{n,k+1}(\lambda)$ can be directly obtained from the $C_m(\lambda)$ sets that could maximize $E_{n,k+1}$ and $E_{n,k}$ (see Supplementary Material in Ameijeiras-Alonso *et al.* 2019), the possible values of λ are computed by looking to the empirical excess mass function in some endpoints candidates for $C_m(\lambda)$ (the number of employed points is equal to the first element of gridsize) and also in the λ values associated to the empirical excess mass for one mode. Once a λ maximizing the approximated values of $D_{n,k+1}(\lambda)$ is chosen, in order to obtain the approximation of the excess mass test statistic, in its neighborhood, a grid of possible λ values is created, being its length equal to the second element of gridsize, and the exact value of $D_{n,k+1}(\lambda)$ is calculated for these values of λ (using the algorithm proposed by Müller and Sawitzki 1991).

An illustration with the stamps data set is shown below. First, the critical bandwidth of Silverman (1981) and Hall and York (2001), in the interval $I = [0.04, 0.15]$, is computed for two modes. Second, the exact and approximated version of the excess mass test statistic of Müller and Sawitzki (1991) for two modes are obtained.

```
R> bwSI <- bw.crit(data = stamps, mod0 = 2)
R> bwSI

[1] 0.003234863

R> bwHY <- bw.crit(data = stamps, mod0 = 2, lowsup = 0.04, uppsup = 0.15,
+    full.result = TRUE)
R> bwHY

n= 485. Number of modes: 2
Bandwidth: 0.003234863
Support where the number of modes are computed
     0.04     0.15

R> emEx <- excessmass(data = stamps, mod0 = 2)
R> emEx

Warning message:
In excessmassex(data, mod0) :
  A modification of the data was made in order to compute the excess mass
  statistic
[1] 0.02177329

R> emApp <- excessmass(data = stamps, mod0 = 2, approximate = TRUE,
+    gridsize = c(50, 50), full.result = TRUE)
R> emApp

n= 485. Number of modes: 2
Approximated excess mass:0.02177101
```

```
Warning message:
In excessmassapp(data, mod0, gridsize) :
  A modification of the data was made in order to compute the excess mass
  statistic
```

Once the different test statistics are computed, the number of modes for the underlying density of a given sample can be tested with the function `modetest`. The different proposals that can be used for testing the number of modes (using the argument `method`) are those ones introduced in Section 2.2. The available methods, based on the critical bandwidth (see Section 2.3), include: Silverman (1981) (`"SI"`), Hall and York (2001) (`"HY"`) and Fisher and Marron (2001) (`"FM"`). Based on the excess mass (Section 2.4): Hartigan and Hartigan (1985) (`"HH"`, equivalent to the proposal of Müller and Sawitzki 1991), Cheng and Hall (1998) (`"CH"`) and the new proposal of Ameijeiras-Alonso *et al.* (2019) (`"ACR"`, value by default) are also included. For calculating the corresponding p value, all the available proposals require bootstrap or Monte Carlo resamples and the number of replicates can be specified with the argument `B` (`B = 500` by default).

For `"SI"`, `"HY"` and `"ACR"` proposals, the argument `submethod` is available (in general, `submethod = 1` is employed by default). In the `"SI"` case, two resampling methods are implemented: when `submethod = 1`, the resamples are generated from the rescaled bootstrap resamples as proposed by Silverman (1981) (see Section 2.3); if `submethod = 2`, the resamples are generated from $\hat{f}_{h_k}$. In the `"ACR"` method, the approximated version of the excess mass can be employed, for computational time efficiency reasons, by setting `submethod = 2` (this method is used by default when `mod0>1` and the sample size is greater than 200); if `submethod = 1`, then the exact value of the excess mass test statistic is computed.

As pointed out in Section 2.2, the bounded support (`lowsup` and `uppsup`) is necessary when the Hall and York (2001) proposal (`"HY"`) is employed and f has not a compact support and it can be also used with the `"ACR"` proposal. In the `"ACR"` case, the parameter `tol2` (equal to 10^{-5} by default) is the accuracy required in the integration of the calibration function when the compact support is known (see Ameijeiras-Alonso *et al.* 2019). As mentioned in Section 2.3, a level correction (achieved with the λ_α factor) is needed in the bootstrap procedure of Hall and York (2001). The two suggested approximations for its computation are provided in the `"HY"` test using the argument `submethod`. The `submethod` 1 corresponds to the asymptotic correction of Silverman (1981) test based on the limiting distribution of the test statistic, i.e., it consists in using equality (7). In Equation 7, since the value of λ_α depends on α, when `submethod = 1`, the significance level must be previously determined with `alpha` (`alpha = 0.05` by default). The `submethod` 2 is based on Monte Carlo techniques where the resamples are generated from the normal distribution. For this reason, when `submethod = 2`, the number of normal-distributed samples (`nMC`) and the number of bootstrap resamples (`BMC`) used for computing the p value in each Monte Carlo sample are needed (both quantities are equal to 100 by default).

The `modetest` function returns a `"htest"` list containing: the test statistic (`statistic`), the associated p value (`p.value`), the number of modes that have been tested (`null.value`), a character string with the alternative hypothesis (`alternative`, which is always `"greater"`), the type of multimodality test that was performed (`method`), the number of non-missing observations in the employed sample (`sample.size`), a character string giving the name of the data (`data.name`) and the number of missing values that were removed from the data prior to performing the test (`bad.obs`).

The different p values obtained for the `stamps` data set with the Ameijeiras-Alonso *et al.* (2019) proposal (calculating the exact value of the excess mass) are reproduced in Table 3 and they can be obtained as follows (varying the value of `mod0` between 1 and 9):

```
R> modeteststamps <- modetest(data = stamps, mod0 = 1, submethod = 1)

Warning message:
In modetest(data = stamps) :
  A modification of the data was made in order to compute the excess mass or
  the dip statistic

R> modeteststamps

        Ameijeiras-Alonso et al. (2018) excess mass test

data:  stamps
Excess mass = 0.049948, p-value < 2.2e-16
alternative hypothesis: true number of modes is greater than 1
```

Assuming that the compact support for the `stamps` data set is $I = [0.04, 0.15]$ (see Izenman and Sommer 1988), the modification of the Ameijeiras-Alonso *et al.* (2019) proposal with known compact support can be obtained as follows

```
R> modetest(data = stamps, mod0 = 1, lowsup = 0.04, uppsup = 0.15)

        Ameijeiras-Alonso et al. (2018) excess mass test

data:  stamps
Excess mass = 0.049948, p-value < 2.2e-16
alternative hypothesis: true number of modes is greater than 1
```

The p values of the other proposals allowing for testing a general number of modes (`"SI"` and `"FM"`) are obtained with the below code lines (varying the value of `mod0` between `1` and `9`).

```
R> modetest(data = stamps, mod0 = 1, method = "SI")

        Silverman (1981) critical bandwidth test

data:  stamps
Critical bandwidth = 0.0067291, p-value = 0.014
alternative hypothesis: true number of modes is greater than 1

R> modetest(data = stamps, mod0 = 1, method = "FM")

        Fisher and Marron (2001) Cramer-von Mises test

data:  stamps
Cramer-von Mises = 1.2302, p-value < 2.2e-16
alternative hypothesis: true number of modes is greater than 1
```

The other critical bandwidth based method, `"HY"`, should only be used for testing unimodality when when f has a bounded support or when the modes and antimodes lie in a known closed interval I, in this case $I = [0.04, 0.15]$. The test with both alternatives for approximating the λ_α: a first approach based on a polynomial approximation (`submethod = 1`) and a second option using Monte Carlo techniques (`submethod = 2`), can be computed as follows:

```
R> modetest(data = stamps, method = "HY", lowsup = 0.04, uppsup = 0.15)

        Hall and York (2001) critical bandwidth test

data:  stamps
Critical bandwidth = 0.0067291, p-value < 2.2e-16
alternative hypothesis: true number of modes is greater than 1

R> modetest(data = stamps, method = "HY", lowsup = 0.04, uppsup = 0.15,
+    submethod = 2)

        Hall and York (2001) critical bandwidth test

data:  stamps
Critical bandwidth = 0.0067291, p-value < 2.2e-16
alternative hypothesis: true number of modes is greater than 1
```

The p values of the unimodality test based on the excess mass (`"HH"` and `"CH"`) can be obtained with the following code lines:

```
R> modetest(data = stamps, method = "HH")

Warning message:
In modetest(data = stamps, method = "HH") :
  A modification of the data was made in order to compute the excess mass or
  the dip statistic

        Hartigan and Hartigan (1985) dip test

data:  stamps
Dip = 0.024974, p-value = 0.04
alternative hypothesis: true number of modes is greater than 1

R> modetest(data = stamps, method = "CH")

Warning message:
In modetest(data = stamps, method = "CH") :
  A modification of the data was made in order to compute the excess mass or
  the dip statistic

        Cheng and Hall (1998) excess mass test
```

method	SI		HY		FM	HH	CH	ACR
submethod	1	2	1	2				1
p value	0.014	0.002	0	0	0	0.040	0	0

Table 2: p value obtained using different proposals for testing unimodality, with $B = 500$ resamples. The employed testing procedures are: "SI" (using the rescaled, submethod 1, and the non-rescaled, submethod 2, bootstrap resamples), "HY" (using the two suggested approximations of λ_α), "FM", "HH", "CH" and "ACR" (employing the exact version of the excess mass test statistic).

k	1	2	3	4	5	6	7	8	9
SI	0.014	0.452	0.094	0.006	0.006	0	0.542	0.342	0.630
FM	0	0.006	0	0	0	0	0	0	0
FM*	0	0.008	0	0	0	0	0.116	0.048	0.044
ACR*	0	0.024	0.002	0.746	0.894	0.980	0.958	0.986	0.936

Table 3: p value obtained using different proposals for testing k-modality, with k between 1 and 9, employing $B = 500$ resamples. The employed testing procedures are: "SI" over the original sample (using the rescaled bootstrap resamples), "FM" over the original sample, "FM" over the perturbed sample ("FM"*) and "ACR" over the perturbed sample (employing the exact version of the excess mass test statistic).

```
data:  stamps
Excess mass = 0.049948, p-value < 2.2e-16
alternative hypothesis: true number of modes is greater than 1
```

Table 2 shows the p.values obtained for all the unimodality tests available. Note that, in the "ACR" case, submethod = 2 was not employed as when mod0 = 1 the exact version of the excess mass is computed in a more efficient way. For all of them the null hypothesis of unimodality is rejected for a significance level $\alpha = 0.05$.

The results for the tests ("SI", "FM" and "ACR") that allow testing k-modality, with $k > 1$, are displayed in Table 3 (with k between 1 and 9). In the case of the "FM" proposal, for reproducing the Fisher and Marron (2001) results, the stamps data were also perturbed as done with the excessmass function. Similar results are obtained for the "SI" proposal, with and without data perturbation when using submethod = 1 or submethod = 2; and for the "ACR" proposal, independently of using or not the known support $I = [0.04, 0.15]$. Fixing a significance level $\alpha = 0.05$, there is not a clear conclusion when using "SI" and "FM". In the "SI" case the null hypothesis is not rejected for $k = 2, 3, 7, 8, 9$ and for the "FM", using the perturbed sample, it is not rejected just for $k = 7$. While, in the single proposal that is well calibrated ("ACR", see Ameijeiras-Alonso *et al.* 2019), the null hypothesis is rejected until $k = 3$ and it is not for $k \geq 4$, suggesting that the number of modes is equal to 4.

Once the number of modes is known, the function locmodes provides the estimation of the locations of modes and antimodes and their estimated density value. In this case, the compact support of the variable (which is known) can be used to obtain a good estimator of the modes and antimodes locations (see Section 2.3). In other scenarios, one should be careful about the conclusions as the critical bandwidth of Silverman (1981) may create artificial modes in

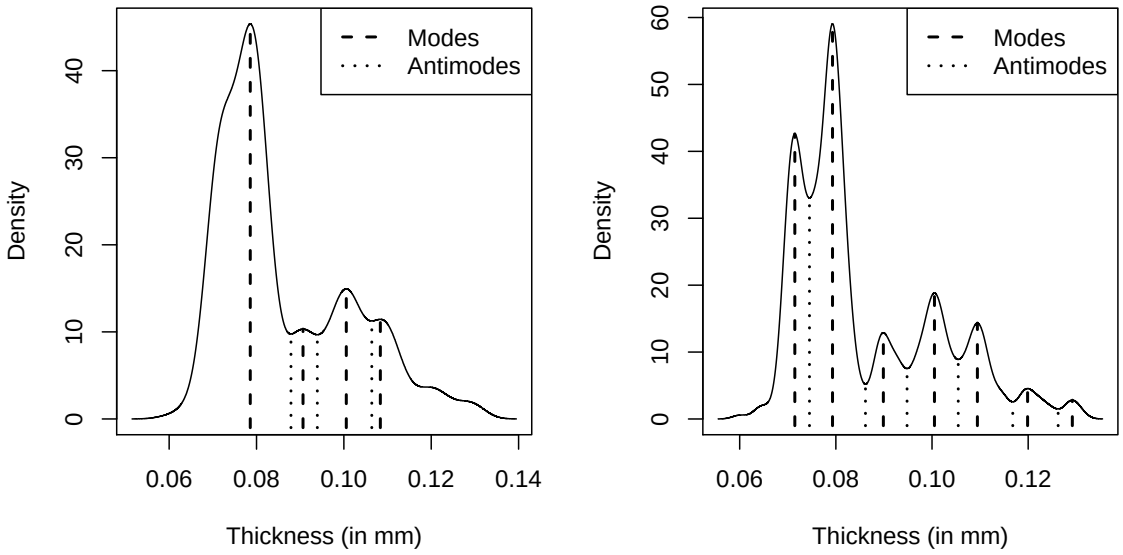

Figure 5: Estimation of the density, modes and antimodes for the sample of 485 stamps from the 1872 Hidalgo Issue of Mexico, obtained with the function `locmodes` for `mod0 = 4` (left) and `mod0 = 7` (right) modes.

the tails (see Hall and York 2001).

The arguments for `locmodes` function include those ones mentioned in the `bw.crit` function: `mod0`, `lowsup`, `uppsup`, `n` and `tol` (with same values by default). It also allows the representation of the estimation (for the number of modes indicated in `mod0`) of the density, modes and antimodes with the argument `display` (`FALSE` by default) or plotting the output. The remaining graphical arguments (`addplot`, `xlab`, `ylab`, `addLegend`, `posLegend`) were already described in the `modetree` and `sizer` functions.

The estimation of the modes and antimodes locations and their density value, assuming four (`mod0 = 4`, Ameijeiras-Alonso *et al.* 2019) and seven (`mod0 = 7`, Izenman and Sommer 1988) modes, can be obtained as follows (their representation is provided in Figure 5):

```
R> lms <- locmodes(data = stamps, mod0 = 4, lowsup = 0.04, uppsup = 0.15,
+    display = TRUE, xlab = "Thickness (in mm)")
R> lms

Estimated location
Modes: 0.07856903  0.09064666  0.1005547  0.1083469
Antimodes: 0.08788637  0.0939225  0.1063922

Estimated value of the density
Modes: 45.36839  10.32331    14.946  11.44271
Antimodes: 9.752408  9.673075  11.25519

Critical bandwidth: 0.002830505
```

This function returns an object of class '`locmod`' (that can be printed and plotted) containing:

`locations`, a vector with the estimated location s of modes (odd positions of the vector) and antimodes (even positions); `fvalue`, a vector with their estimated density values; and `cbw`, the critical bandwidth of the sample for `mod0` modes. Regarding the obtained results assuming that the distribution has four modes, the estimated modes (odd positions of `locations`) are: 0.07857, 0.09065, 0.1006 and 0.1083.

The results obtained after applying the `modetest` function can be helpful for having a better interpretation of the SiZer map (Figure 4). If the conclusion is that there are four modes, the most plausible results are obtained with the bootstrap quantiles q_3 and, in that case, the estimated modes (in `locmodes`) coincide with those ones observed when a value of $\log_{10}(h)$ close to -2.7 is taken.

4. Discussion

The available functions of the R package **multimode** were described in this paper. This package was developed with the objective of making the mode testing and exploring procedures, for linear data, accessible for the scientific community, and therefore, enabling its use in practical problems. As pointed out in Section 1, there are many examples in different disciplines where the identification of the number (and location) of modes is important per se, or as a previous step for applying other procedures. Package **multimode** contains non-parametric graphical tools for (visually) exploring the number of modes and their estimated location and also testing proposals for determining the number of modes in the data distribution.

Up to the authors' knowledge, **multimode** is the only statistical package that allows for testing, in a non-parametric way, a general number of modes and, also, it is the only one providing a well-calibrated method for testing unimodality. Obtaining a final p value, instead of a graphical tool, can be useful when the objective is, e.g., to obtain conclusions about the number of modes in a systematic manner. This is the case of McQuillan *et al.* (2014) or Joubert *et al.* (2016) where they performed several times the unimodality test of Hartigan and Hartigan (1985), dividing the sample, in the first case, in a temperature bin, and in the second case, in a collection of different CpGs (see Section 1). The combination of this package with other False Discovered Rate techniques (see, e.g., `p.adjust` from the **stats** package) allows to account for the multiple testing problem when the objective is to determine the number of modes.

So far, **multimode** includes just exploratory and testing procedures for mode assessment for real random variables. However, the ideas in Ameijeiras-Alonso *et al.* (2019) can be extended to settings where there is a natural non-parametric estimator. This is the case with circular random variables, for instance. As mentioned before, in R, there are already some packages allowing for exploring the number of modes in this setting, such as the circular version of the SiZer map implemented in **NPCirc** (see Oliveira *et al.* 2014b). Referring to the testing approach, Fisher and Marron (2001) already introduced a proposal for determining the number of modes in this circular setting. In particular, they suggested to use the circular version of the T_k test statistic, namely the U^2 of Watson (1961). Future extensions of the `multimode` package could include some procedures for assessing the number of modes in other settings, such as the mentioned proposal of Fisher and Marron (2001).

Acknowledgments

The authors gratefully acknowledge the support of Project MTM2016–76969–P from the Spanish State Research Agency (AEI) co-funded by the European Regional Development Fund (ERDF), the Competitive Reference Groups 2017–2020 (ED431C 2017/38) from the Xunta de Galicia through the ERDF. Work of J. Ameijeiras-Alonso has been supported by the FWO research project G.0826.15N (Flemish Science Foundation), GOA/12/014 project (Research Fund KU Leuven) and was partially supported by the grant BES–2014–071006 from the Spanish Ministry of Science, Innovation and Universities.

References

Ameijeiras-Alonso J, Crujeiras RM, Rodríguez-Casal A (2019). "Mode Testing, Critical Bandwidth and Excess Mass." *Test*, **28**(3), 900–919. doi:10.1007/s11749-018-0611-5.

Ameijeiras-Alonso J, Crujeiras RM, Rodríguez-Casal A (2021). ***multimode**: Mode Testing and Exploring*. R package version 1.5, URL https://CRAN.R-project.org/package=multimode.

Azzalini A, Bowman AW (1990). "A Look at Some Data on the Old Faithful Geyser." *Journal of the Royal Statistical Society C*, **39**(3), 357–365. doi:10.2307/2347385.

Bechtel YC, Bonaiti-Pellie C, Poisson N, Magnette J, Bechtel PR (1993). "A Population and Family Study *N*-Acetyltransferase Using Caffeine Urinary Metabolites." *Clinical Pharmacology & Therapeutics*, **54**(2), 134–141. doi:10.1038/clpt.1993.124.

Benaglia T, Chauveau D, Hunter DR, Young D (2009). "**mixtools**: An R Package for Analyzing Finite Mixture Models." *Journal of Statistical Software*, **32**(6), 1–29. doi:10.18637/jss.v032.i06.

Chaudhuri P, Marron JS (1999). "SiZer for Exploration of Structures in Curves." *Journal of the American Statistical Association*, **94**(447), 807–823. doi:10.1080/01621459.1999.10474186.

Cheng MY, Hall P (1998). "Calibrating the Excess Mass and Dip Tests of Modality." *Journal of the Royal Statistical Society B*, **60**(3), 579–589. doi:10.1111/1467-9868.00141.

Colombo MG, Franzoni C, Rossi-Lamastra C (2015). "Internal Social Capital and the Attraction of Early Contributions in Crowdfunding." *Entrepreneurship Theory and Practice*, **39**(1), 75–100. doi:10.1111/etap.12118.

Crawford SL (1994). "An Application of the Laplace Method to Finite Mixture Distributions." *Journal of the American Statistical Association*, **89**(425), 259–267. doi:10.1080/01621459.1994.10476467.

Dümbgen L, Walther G (2008). "Multiscale Inference about a Density." *The Annals of Statistics*, **36**(4), 1758–1785. doi:10.1214/07-aos521.

Duong T, Cowling A, Koch I, Wand MP (2008). "Feature Significance for Multivariate Kernel Density Estimation." *Computational Statistics & Data Analysis*, **52**(9), 4225–4242. doi:10.1016/j.csda.2008.02.035.

Duong T, Wand M (2015). ***feature**: Local Inferential Feature Significance for Multivariate Kernel Density Estimation*. R package version 1.2.13, URL https://CRAN.R-project.org/package=feature.

Fisher NI, Marron JS (2001). "Mode Testing via the Excess Mass Estimate." *Biometrika*, **88**(2), 419–517. doi:10.1093/biomet/88.2.499.

Freeman JB, Dale R (2013). "Assessing Bimodality to Detect the Presence of a Dual Cognitive Process." *Behavior Research Methods*, **45**(1), 83–97. doi:10.3758/s13428-012-0225-x.

Genovese CR, Perone-Pacifico M, Verdinelli I, Wasserman L (2016). "Non-Parametric Inference for Density Modes." *Journal of the Royal Statistical Society B*, **78**(1), 99–126. doi:10.1111/rssb.12111.

Godtliebsen F, Marron JS, Chaudhuri P (2002). "Significance in Scale Space for Bivariate Density Estimation." *Journal of Computational and Graphical Statistics*, **11**(1), 1–21. doi:10.1198/106186002317375596.

Good IJ, Gaskins RA (1980). "Density Estimation and Bump-Hunting by the Penalized Likelihood Method Exemplified by Scattering and Meteorite Data." *Journal of the American Statistical Association*, **75**(369), 42–56. doi:10.1080/01621459.1980.10477419.

Hall P, York M (2001). "On the Calibration of Silverman's Test for Multimodality." *Statistica Sinica*, **11**(1), 515–536.

Härdle W (2012). *Smoothing Techniques: With Implementation in S*. Springer-Verlag, New York. doi:10.1007/978-1-4612-4432-5.

Hartigan JA, Hartigan PM (1985). "The Dip Test of Unimodality." *The Annals of Statistics*, **13**(1), 70–84. doi:10.1214/aos/1176346577.

Izenman AJ, Sommer CJ (1988). "Philatelic Mixtures and Multimodal Densities." *Journal of the American Statistical Association*, **83**(404), 941–953. doi:10.1080/01621459.1988.10478683.

Joubert BR, Felix JF, Yousefi P, Bakulski KM, Just AC, Breton C, Reese SE, Markunas CA, Richmond RC, Xu CJ, Küpers LK, Oh SS, Hoyo C, Gruzieva O, Söderhäll C, Salas LA, Baïz N, Zhang H, Lepeule J, Ruiz C, Ligthart S, Wang T, Taylor JA, Duijts L, Sharp GC, Jankipersadsing SA, Nilsen RM, Vaez A, Fallin MD, Hu D, Litonjua AA, Fuemmeler BF, Huen K, Kere J, Kull I, Munthe-Kaas MC, Gehring U, Bustamante M, Saurel-Coubizolles MJ, Quraishi BM, Ren J, Tost J, Gonzalez JR, Peters MJ, Håberg SE, Xu Z, van Meurs JB, Gaunt TR, Kerkhof M, Corpeleijn E, Feinberg AP, Eng C, Baccarelli AA, Neelon SEB, Bradman A, Merid SK, Bergström A, Herceg Z, Hernandez-Vargas H, Brunekreef B, Pinart M, Heude B, Ewart S, Yao J, Lemonnier N, Franco OH, Wu MC, Hofman A, McArdle W, der Vlies PV, Falahi F, Gillman MW, Barcellos LF, Kumar A, Wickman M, Guerra S, Charles MA, Holloway J, Auffray C, Tiemeier HW, Smith GD, Postma D, Hivert MF, Eskenazi B, Vrijheid M, Arshad H, Antó JM, Dehghan A, Karmaus W, Annesi-Maesano I,

Sunyer J, Ghantous A, Pershagen G, Holland N, Murphy SK, DeMeo DL, Burchard EG, Ladd-Acosta C, Snieder H, Nystad W, Koppelman GH, Relton CL, Jaddoe VWV, Wilcox A, Melén E, London SJ (2016). "DNA Methylation in Newborns and Maternal Smoking in Pregnancy: Genome-Wide Consortium Meta-Analysis." *The American Journal of Human Genetics*, **98**(4), 680–696. doi:10.1016/j.ajhg.2016.02.019.

Komárek A, Komárková L (2014). "Capabilities of R Package **mixAK** for Clustering Based on Multivariate Continuous and Discrete Longitudinal Data." *Journal of Statistical Software*, **59**(12), 1–38. doi:10.18637/jss.v059.i12.

Maechler M (2016). ***diptest**: Hartigan's Dip Test Statistic for Unimodality – Corrected*. R package version 0.75-7, URL http://CRAN.R-project.org/package=diptest.

McLachlan G, Peel D (2000). *Finite Mixture Models*. John Wiley & Sons, United States of America. doi:10.1002/0471721182.

McQuillan A, Mazeh T, Aigrain S (2014). "Rotation Periods of 34,030 Kepler Main-Sequence Stars: The Full Autocorrelation Sample." *The Astrophysical Journal Supplement Series*, **211**(2), 24. doi:10.1088/0067-0049/211/2/24.

Minnotte MC, Marchette DJ, Wegman EJ (1998). "The Bumpy Road to the Mode Forest." *Journal of Computational and Graphical Statistics*, **7**(2), 239–251. doi:10.1080/10618600.1998.10474773.

Minnotte MC, Scott DW (1993). "The Mode Tree: A Tool for Visualization of Nonparametric Density Features." *Journal of Computational and Graphical Statistics*, **2**(1), 51–68. doi:10.2307/1390955.

Mukhopadhyay S (2017). "Large-Scale Mode Identification and Data-Driven Sciences." *Electronic Journal of Statistics*, **11**(1), 215–240. doi:10.1214/17-ejs1229.

Müller DW, Sawitzki G (1991). "Excess Mass Estimates and Tests for Multimodality." *Journal of the American Statistical Association*, **86**(415), 738–746. doi:10.1080/01621459.1991.10475103.

Oliveira M, Crujeiras RM, Rodríguez-Casal A (2014a). "CircSiZer: An Exploratory Tool for Circular Data." *Environmental and Ecological Statistics*, **21**(1), 143–159. doi:10.1007/s10651-013-0249-0.

Oliveira M, Crujeiras RM, Rodríguez-Casal A (2014b). "**NPCirc**: An R Package for Nonparametric Circular Methods." *Journal of Statistical Software*, **61**(9), 1–26. doi:10.18637/jss.v061.i09.

Parzen E (1962). "On Estimation of a Probability Density Function and Mode." *The Annals of Mathematical Statistics*, **33**(3), 1065–1076. doi:10.1214/aoms/1177704472.

Poncet P (2019). ***modeest**: Mode Estimation*. R package version 2.4.0, URL https://CRAN.R-project.org/package=modeest.

Postman M, Huchra JP, Geller MJ (1986). "Probes of Large-Scale Structure in the Corona Borealis Region." *The Astronomical Journal*, **92**, 1238–1247. doi:10.1086/114257.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Richardson S, Green PJ (1997). "On Bayesian Analysis of Mixtures with an Unknown Number of Components." *Journal of the Royal Statistical Society B*, **59**(4), 731–792. doi:10.1111/1467-9868.00095.

Roeder K (1990). "Density Estimation with Confidence Sets Exemplified by Superclusters and Voids in the Galaxies." *Journal of the American Statistical Association*, **85**(411), 617–624. doi:10.1080/01621459.1990.10474918.

Rosenblatt M (1956). "Remarks on Some Nonparametric Estimates of a Density Function." *The Annals of Mathematical Statistics*, **27**(3), 832–837. doi:10.1214/aoms/1177728190.

Rufibach K, Walther G (2010). "The Block Criterion for Multiscale Inference about a Density, with Applications to Other Multiscale Problems." *Journal of Computational and Graphical Statistics*, **19**(1), 175–190. doi:10.1198/jcgs.2009.07071.

Rufibach K, Walther G (2015). ***modehunt**: Multiscale Analysis for Density Functions*. R package version 1.0.7, URL https://CRAN.R-project.org/package=modehunt.

Salgado-Ugarte IH, Shimizu M, Taniuchi T (1998). "Nonparametric Assessment of Multimodality for Univariate Data." *Stata Technical Bulletin*, **7**, 27–35. URL https://www.stata-press.com/journals/stbcontents/stb38.pdf.

Santoro M, Beer C, Cartus O, Schmullius C, Shvidenko A, McCallum I, Wegmüller U, Wiesmann A (2011). "Retrieval of Growing Stock Volume in Boreal Forest Using Hyper-Temporal Series of Envisat ASAR ScanSAR Backscatter Measurements." *Remote Sensing of Environment*, **115**(2), 490–507. doi:10.1016/j.rse.2010.09.018.

Scott DW (2015). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, Hoboken, New Jersey. doi:10.1002/9780470316849.

Scrucca L, Fop M, Murphy TB, Raftery AE (2016). "**mclust** 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models." *The R Journal*, **8**, 205–233. doi:10.32614/rj-2016-021.

Silverman BW (1981). "Using Kernel Density Estimates to Investigate Multimodality." *Journal of the Royal Statistical Society B*, **43**(1), 97–99. doi:10.1111/j.2517-6161.1981.tb01155.x.

Wand MP, Jones MC (1994). *Kernel Smoothing*. Chapman & Hall/CRC, New York. doi:10.1201/b14876.

Watson GS (1961). "Goodness-of-Fit Tests on a Circle." *Biometrika*, **48**(1/2), 109–114. doi:10.2307/2333135.

Weisberg S (2005). *Applied Linear Regression*. 3rd edition. John Wiley & Sons, New York. doi:10.1002/0471704091.

Affiliation:

Jose Ameijeiras-Alonso, Rosa M. Crujeiras, Alberto Rodríguez-Casal
Department of Statistics, Mathematical Analysis and Optimization
Faculty of Mathematics
Universidade de Santiago de Compostela
15782 Santiago de Compostela, Spain
E-mail: jose.ameijeiras@usc.es, rosa.crujeiras@usc.es,
alberto.rodriguez.casal@usc.es
URL: https://jose-ameijeiras.netlify.app/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
March 2021, Volume 97, Issue 9 *Submitted:* 2018-04-27
doi:10.18637/jss.v097.i09 *Accepted:* 2019-01-11