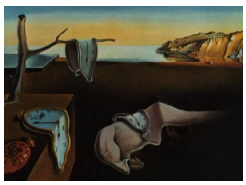


Journal of Statistical Software

December 2025, Volume 115, Issue 3. *doi: 10.18637/jss.v115.i03*

MixtureMissing: An R Package for Robust and Flexible Model-Based Clustering with Incomplete Data

Hung Tong
Rowan University

Cristina Tortora
San José State University

Abstract

The R package **MixtureMissing** performs model-based clustering on data sets with values missing at random, aiming to identify homogeneous groups of observations. In model-based clustering, the data within each cluster follow a specific distribution. In the package, 13 distributions are available, including the contaminated normal distribution, the generalized hyperbolic distribution (GHD), and 11 special or limiting cases of GHD. Notably, eight out of these 11 cases have not been formulated at the time of writing. Given a list of candidate distributions, the package can recommend the optimal distribution to employ based on a specified information criterion. In this paper, the methodological foundations and computational aspects of the package are discussed. Furthermore, important features of model fitting, model summary, and available visualization tools are thoroughly illustrated using real data sets.

Keywords: model-based clustering, EM algorithm, outliers, skewness, missing data, contaminated normal distribution, generalized hyperbolic distribution.

1. Introduction

In this paper, we focus on robust and flexible model-based approaches to cluster analysis, as well as their software implementation in the presence of missing values. In real-world applications, data sets are often heavy-tailed and asymmetric. Additionally, these data can include partially observed records, which may limit the effectiveness of traditional statistical methods. Therefore, the availability of computational tools that address these challenges will significantly benefit researchers and practitioners across various fields.

Cluster analysis aims to partition a multivariate data set into smaller, distinct groups, known as clusters. Although the definition of a cluster is highly dependent on research goals and

subject areas (Hennig 2015), it is generally desirable for a cluster to consist of observations that are as similar as possible to each other, but relatively different from those of other clusters. The use of cluster analysis has proven to be a powerful tool for exploratory analysis and heterogeneity discovery, finding applications in diverse fields such as marketing, economics, bioinformatics, psychiatry, and geography. For a formal introduction to cluster analysis, one can refer to Kaufman and Rousseeuw (1990) and Everitt, Landau, Leese, and Stahl (2011).

Recently, model-based clustering – the idea of performing cluster analysis by fitting a finite mixture model (McLachlan and Peel 2000) – has garnered significant attention from researchers and statisticians. Key references in this area include Fraley and Raftery (2002) and Stahl and Sallis (2012). In this framework, each observation is considered to arise from a mixture of known probability distributions within the same parametric family. Therefore, the clustering problem involves estimating the mixture parameters. There are two main approaches to estimating the parameters: the frequentist (or likelihood-based) approach and the Bayesian approach (see Bishop and Nasrabadi 2006, for more details). The focus of this paper is on the frequentist approach where maximum likelihood estimation for the parameters of the cluster is generally achieved through the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin 1977). The EM algorithm is an iterative parameter estimation procedure designed for situations involving incomplete data or latent variables. Upon convergence of the algorithm, cluster memberships can be determined by maximizing the *a posteriori* probability. Model-based clustering offers an inferential advantage compared to other heuristic cluster analysis methods such as k-means (Macqueen 1967) or hierarchical clustering (Ward 1963). This advantage comes from the fact that each cluster can be characterized by its corresponding mixture component.

There are an extensive number of R packages that perform model-based clustering on various data types. In particular, the **mclust** package (Scrucca, Fop, Murphy, and Raftery 2016) stands out as one of the most widely used packages for continuous data. In a high-dimensional setting, model-based clustering of continuous data can be used by exploring various parsimonious models as in package **pgmm** (McNicholas, ElSherbiny, McDaid, and Murphy 2021) or by assuming that the data can be represented in a lower dimension than the original space as in package **HDclassif** (Bergé, Bouveyron, and Girard 2012). Regarding non-continuous data, the package **BayesBinMix** (Papastamoulis and Rattray 2017) is tailored for binary data, and the package **ordinalclust** (Selosse, Jacques, and Biernacki 2020) is designed for ordinal data. Furthermore, the package **funFEM** (Bouveyron 2021) provides model-based clustering of functional data, as well as time series. Most of these packages are based on the frequentist approach to model-based clustering. However, there are also several packages based on the Bayesian approach, such as **REBayes** (Koenker and Gu 2017), **bayesmix** (Grün and Plummer 2023), or **BNPmix** (Corradin, Canale, and Nipoti 2021). In Python, several clustering techniques are implemented in **sklearn.cluster** (Pedregosa *et al.* 2011). MATLAB also includes some commonly used clustering techniques (The MathWorks Inc. 2025).

In model-based clustering literature, the normal distribution is extensively utilized, often serving as the primary choice for mixture distributions. Numerous R packages are built on this distribution, including **mclust** (Scrucca *et al.* 2016), **mixture** (Pocuca, Browne, and McNicholas 2021), **pgmm** (McNicholas *et al.* 2021), and **mixtools** (Benaglia, Chauveau, Hunter, and Young 2009), to name a few. The general mixture modeling **MIXMOD** program fits mixture models to a given data set using C++. It fits the multivariate Gaussian mixtures and fourteen different variations based on constraints on the covariance matrix. **MIXMOD** has

interfaces for many other programming languages, including R (Lebret, Iovleff, Langrognet, Biernacki, Celeux, and Govaert 2015), Python (Singleton 2023), and MATLAB (Biernacki, Celeux, Govaert, and Langrognet 2006). In Python, model-based clustering using the Gaussian distribution is also implemented in **sklearn.cluster** (Pedregosa *et al.* 2011). In MATLAB, model-based clustering can instead be obtained using the model-based clustering toolbox (Martinez and Martinez 2003).

Despite its popularity, the use of the normal distribution in model-based clustering has some limitations on the shape of the clusters. First, with its light tails, the normal distribution is not robust against extreme observations – those that significantly deviate from the rest and are commonly known as outliers (Hawkins 1980). Second, it is symmetric around the mean, an assumption that may be not realistic in many clustering applications (see for example Wallace, Buysse, Germain, Hall, and Iyengar 2018). Third, the majority of the packages mentioned so far assume that the data are complete, i.e., without missing values. Solutions have been proposed for all the mentioned issues.

Starting with data sets with outliers, this characteristic introduces the risk of unreliable estimation of component means and covariance matrices. Generally speaking, outliers can be categorized into two types: gross outliers and mild outliers (Ritter 2014, pp. 79–80). Gross outliers are those that cannot be adequately modeled by a distribution. Dealing with this type of outliers often involves recommended approaches, such as removing them using a trimming approach (Cuesta-Albertos, Gordaliza, and Matran 1997; Fritz, Garcia-Escudero, and Mayo-Iscar 2012) or maximizing the trimmed likelihood of a partition model (García-Escudero, Gordaliza, Matrán, and Mayo-Iscar 2008). The R package **tclust** (Fritz *et al.* 2012) implements several clustering techniques, all based on the trimming approach. Alternatively, gross outliers can be considered as scatters – observations that do not resemble any others in the data. Byers and Raftery (1998) proposed modeling scatters by adding a Poisson process component to the mixture model, a procedure implemented in the R package **mclust**. Recent advances in clustering in the presence of scatters can be found in Tseng and Wong (2005) and Maitra and Ramler (2009). Mild outliers, on the other hand, are sampled from populations different from or even far from the assumed model, such as the normal distribution, as discussed in detail by Davies and Gather (1993). In this paper, the main focus is to address mild outliers for model-based clustering, where a common approach is to employ a heavy-tailed distribution to model each component of the mixture. In particular, the t distribution (Peel and McLachlan 2000) and the contaminated normal distribution (Punzo and McNicholas 2016) emerge as two suitable candidates, given that their means and covariance matrices are less affected by outliers. Furthermore, both distributions can be employed for outlier detection. The R packages that implement t and contaminated normal mixtures are **teigen** (Andrews, Wickins, Boers, and McNicholas 2018) and **ContaminatedMixt** (Punzo, Mazza, and McNicholas 2018), respectively. In Python, only the Student-t mixture is implemented in **student_pyt** (Tomer 2020).

Moving to the issue of symmetry of the clusters, several other multivariate distributions have also been employed within model-based clustering to improve flexibility in cluster shapes. Notable examples include the skew-normal distribution (Lee and McLachlan 2013), skew-t distribution (Murray, Browne, and McNicholas 2014), and generalized hyperbolic distribution (Browne and McNicholas 2015), all of which incorporate a skewness parameter. The skew-t and generalized hyperbolic distributions are also known for their robustness. One major advantage of the generalized hyperbolic distribution is that it encompasses many other

distributions, including the normal, skew-t, and t, as special or limiting cases. In fact, in a study by Bagnato, Farcomeni, and Punzo (2024), a hierarchy of 15 distributions can be obtained from the generalized hyperbolic distribution. The R package that implements the mixture of generalized hyperbolic distributions is **MixGHD** (Tortora, Browne, El Sherbiny, Franczak, and McNicholas 2021). For a more comprehensive review of non-Gaussian distributions in model-based clustering, refer to McNicholas (2016).

Finally, despite the high flexibility in cluster shape achieved by some of the mentioned models, clustering may still be challenging due to the presence of missing values in the data. In the literature on model-based clustering, some extensions have been made to incomplete data, including Ghahramani and Jordan (1994) for the normal mixture, Wang, Zhang, Luo, and Wei (2004) for the t mixture, Tong and Tortora (2022) for the contaminated normal mixture, and Wei, Tang, and McNicholas (2019) for the mixtures of generalized hyperbolic and skew-t mixtures. From this list of works, it is evident that only four distributions in the hierarchy shown by (Bagnato *et al.* 2024) – normal, t, skew-t, and generalized hyperbolic – have been extended to handle missing values. This limitation is notable, especially considering the various distributions that can be derived from the generalized hyperbolic distribution. Therefore, in this paper, we extend the framework of fitting mixtures of eight more distributions in this hierarchy to both complete and incomplete data. More importantly, we introduce the R package **MixtureMissing**, which implements these new models as well as the normal, t, contaminated normal, generalized hyperbolic, and skew-t mixtures. With a total of 13 distributions for model-based clustering, the package provides a robust and flexible tool for modeling real data from various applications.

Generally, handling missing values in model-based clustering and cluster analysis at large is not a novel concept. For example, just focusing on R, many packages already exist; see Josse, Mayer, Tierney, and Vialaneix (2025) for an in-depth overview. However, much emphasis has been placed on preprocessing the original data set with an imputation technique in which incomplete entries are filled with some point estimates; for an overview of imputation, see Van Buuren (2021) and Little and Rubin (2020). In particular, the package **mixture** attempts to impute when performing the expectation step of the EM algorithm, while the package **mclust** includes the function `imputeData()` for missing data imputation via the package **mix** (Schafer 2017) based on multiple imputations. Additionally, the package **ClustImpute** (Pfaffel 2020) performs k-means clustering after random imputation, and the package **miclust** (Basagaña, Barrera-Gómez, Benet, Antó, and Garcia-Aymerich 2013) integrates k-means with multiple imputation. The package **VarSelLCM** (Marbac and Sedki 2019) can handle mixed-type data model-based clustering without pre-processing, its framework primarily focuses on variable selection rather than parameter estimation. Specific for model-based clusters in the presence of missing data is the packages **MGMM** (McCaw 2023), which implements a mixture of Gaussian distributions, and **RMixtComp** (Kubicki *et al.* 2023), which includes models for different data types and uses mixtures of Gaussian distributions for continuous data. In Python, missing data can only be handled before starting cluster analysis using **SimpleImputer** or **IterativeImputer** (Pedregosa *et al.* 2011). Similarly, in MATLAB the imputation of the missing values can be obtained using **fillmissing**. Therefore, the new R package **MixtureMissing** fills the gap by providing a model-based clustering implementation that directly accounts for missing values in the EM algorithm using several distributions.

The outline of the paper is as follows. First, we characterize different missing data mechanisms. Next, after providing a mathematical definition of model-based clustering, we in-

troduce the contaminated normal and generalized hyperbolic mixtures for data with values missing at random. We then discuss some relevant computational aspects and guide readers through the main functionalities of the package. Finally, we conclude with a summary of the main key points.

2. Missing data mechanisms

A missing-data mechanism refers to the connection between the occurrence of missingness and the underlying values of the variables. Suppose we have an $n \times d$ data matrix $\boldsymbol{X} = \{X_{ij}\} = (\boldsymbol{X}_{\text{obs}}, \boldsymbol{X}_{\text{mis}})$, for $i = 1, \ldots, n$ and $j = 1, \ldots, d$, that is partitioned into the observed part $\boldsymbol{X}_{\text{obs}}$ and the missing part $\boldsymbol{X}_{\text{mis}}$. Let $M = \{m_{ij}\}$ be the $n \times d$ missing-data indicator matrix, such that $m_{\text{ij}} = 1$ if X_{ij} is missing and $m_{\text{ij}} = 0$ otherwise. Little and Rubin (2020) defines the missing-data mechanism based on the conditional distribution of M given $\boldsymbol{X}$, that is $f(M \mid \boldsymbol{X}, \phi)$, where ϕ denotes some unknown parameters. In particular, the data are called missing completely at random (MCAR) if

$$f(M \mid \boldsymbol{X}, \phi) = f(M \mid \phi) \text{ for all } \boldsymbol{X}, \phi, \tag{1}$$

which implies that missingness does not depend on the values of the data $\boldsymbol{X}$, missing or observed. When missingness depends only on the observed part $\boldsymbol{X}_{\text{obs}}$ of $\boldsymbol{X}$, the data are called missing at random (MAR). Specifically,

$$f(M \mid \boldsymbol{X}, \phi) = f(M \mid \boldsymbol{X}_{\text{obs}}, \phi) \text{ for all } \boldsymbol{X}_{\text{mis}}, \phi. \tag{2}$$

MAR is one of the most common assumptions in clustering with missing data and also includes MCAR as a special case. In contrast, the data are called not missing at random (NMAR) if the distribution of M depends on the missing values $\boldsymbol{X}_{\text{mis}}$ of $\boldsymbol{X}$.

In this paper, we assume that the incomplete data are MAR. Additionally, the data have a general missing data pattern where any two observations can have different numbers of missing values on different variables.

3. Model-based clustering

In model-based clustering, the underlying population is assumed to be a convex combination of components, each represented by a known probability distribution. Mathematically, a random vector $\boldsymbol{X}$ with d variables follows a mixture distribution if its probability density function (pdf) is

$$f(\boldsymbol{x}; \boldsymbol{\Psi}) = \sum_{g=1}^{G} \pi_g f(\boldsymbol{x}\,; \boldsymbol{\vartheta}_g), \tag{3}$$

where G denotes the number of components assumed to be known in advance; π_g is the mixing proportion for the g-th component such that $\pi_g > 0$ and $\sum_{g=1}^{G} \pi_g = 1$; $f(\boldsymbol{x}\,; \boldsymbol{\vartheta}_g)$ is the pdf of the g-th component identified by parameters $\boldsymbol{\vartheta}_g$; and $\boldsymbol{\Psi} = \{\boldsymbol{\pi}, \boldsymbol{\vartheta}\}$ is the set of all the model parameters.

Model-based clustering of an incomplete data set can be formulated as follows. Let $\boldsymbol{X} = \{\boldsymbol{X}_i\}_{i=1}^{n}$ be the given data that consists of n independent observations. Assuming some

values missing at random, each observation $\boldsymbol{X}_i$ can be rewritten as $(\boldsymbol{X}_i^o, \boldsymbol{X}_i^m)$ where the d_i^o-dimensional vector $\boldsymbol{X}_i^o$ denotes its observed values and the d_i^m-dimensional vector $\boldsymbol{X}_i^m$ denotes its missing values. Herein, the notations o and m are used instead of o_i and m_i for the sake of simplicity, and they do not imply that all observations have the same pattern of missingness. Meanwhile, another source of missing values also arises from the latent variable representing all observations' unobserved cluster memberships $\boldsymbol{Z} = \{\boldsymbol{Z}_i\}_{i=1}^n$ where $\boldsymbol{Z}_i = (Z_{i1}, \ldots, Z_{iG})^\top$ is a 0-1 binary vector with one and only one element equal to 1. Then, the clustering task amounts to obtaining the maximum likelihood estimates for $\boldsymbol{\Psi}$ from the complete-data $\{\boldsymbol{X}, \boldsymbol{Z}\} = \{\boldsymbol{X}_i^o, \boldsymbol{X}_i^m, \boldsymbol{Z}_i\}_{i=1}^n$, which can be done using the expectation-maximization (EM) algorithm developed by Dempster *et al.* (1977). The algorithm alternates between one expectation (E) step and one maximization (M) step until convergence as follows

- **E-step:** Calculate the expected value

$$Q\left(\boldsymbol{\Psi}\,;\,\boldsymbol{\Psi}^{(r)}\right) = E\Big\{\, l\left(\boldsymbol{\Psi}\,;\,\boldsymbol{X}, \boldsymbol{Z}\right) \;\mid\; \boldsymbol{x}_1^o, \ldots, \boldsymbol{x}_n^o, \boldsymbol{\Psi}^{(r)}\Big\},$$

  where $l\left(\boldsymbol{\Psi}\,;\,\boldsymbol{X}, \boldsymbol{Z}\right)$ is the log-likelihood function based on the complete-data $\{\boldsymbol{X}, \boldsymbol{Z}\}$ and $\boldsymbol{\Psi}^{(r)}$ is the estimate of $\boldsymbol{\Psi}$ at iteration r.

- **M-step:** Update the current parameters $\boldsymbol{\Psi}^{(r)}$ with the new parameters $\boldsymbol{\Psi}^{(r+1)}$ that maximize the expectation $Q\left(\boldsymbol{\Psi}\,;\,\boldsymbol{\Psi}^{(r)}\right)$ obtained in the E-step.

3.1. Contaminated normal mixtures

The contaminated normal mixture (CNM) introduced by Punzo and McNicholas (2016) is a powerful robust model-based clustering method. The probability density function (pdf) of a $d-$dimensional random vector $\boldsymbol{X}$ following an CNM with G components can be obtained by setting $f(\boldsymbol{x}\,;\,\boldsymbol{\vartheta}_g)$ in Equation 3 equal to

$$f_{\text{CN}}\left(\boldsymbol{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \alpha_g, \eta_g\right) = \alpha_g f_{\text{N}}\left(\boldsymbol{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right) + \left(1-\alpha_g\right) f_{\text{N}}\left(\boldsymbol{x}; \boldsymbol{\mu}_g, \eta_g\boldsymbol{\Sigma}_g\right), \tag{4}$$

which is the pdf of a contaminated normal (CN) distribution with mean $\boldsymbol{\mu}_g$, covariance matrix $\boldsymbol{\Sigma}_g$, proportion of good points $\alpha_g \in (0.5, 1)$, and degree of contamination $\eta_g > 1$. $f_N\left(\boldsymbol{x}\,;\,\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ is the pdf of a normal distribution. Thus, the CN distribution itself is a two-component multivariate normal mixture in which one component, with covariance matrix $\boldsymbol{\Sigma}_g$, represents the good observations and the other component represents the bad observations (or outliers), each weighted by probabilities α_g and $1-\alpha_g$, respectively. The two components share the same mean $\boldsymbol{\mu}_g$, but the bad observation component has a covariance matrix inflated by a factor of η_g. All the distributions used within model-based clustering are multivariate, even if it not always explicitly stated for the sake of simplicity.

Tong and Tortora (2022) developed a framework for fitting the contaminated normal mixture (CNM) to incomplete data sets. The authors used the expectation-conditional maximization either (ECME) algorithm (Liu and Rubin 1994) – a variant of the traditional EM algorithm where the traditional maximization step is replaced by simpler conditional maximization (CM) steps. Some of the CM steps will maximize the expected value $Q\left(\boldsymbol{\Psi}\,;\,\boldsymbol{\Psi}^{(r)}\right)$ as usual, while others will instead maximize the observed log-likelihood function. The framework takes into account three sources of missing data:

1. Component memberships for all observations: $\boldsymbol{Z} = \{\boldsymbol{Z}_i\}_{i=1}^n$, where $\boldsymbol{Z}_i = (Z_{i1}, \ldots, Z_{iG})^\top$ so that $Z_{ig} = 1$ if observation i belongs to component g, otherwise $Z_{ig} = 0$.

2. Whether an observation $\boldsymbol{X}_i$ is a good or bad point in each component, denoted as $\boldsymbol{V} = \{\boldsymbol{V}_i\}_{i=1}^n$, where $\boldsymbol{V}_i = (V_{i1}, \ldots, V_{iG})^\top$ so that $V_{ig} = 1$ if observation $\boldsymbol{x}_i$ is a good point in component g, and $V_{ig} = 0$ otherwise.

3. Missing values from each observation $\{\boldsymbol{X}_i^m\}_{i=1}^n$.

For details on the algorithm, see Tong and Tortora (2022).

3.2. Multivariate generalized hyperbolic mixtures

According to McNeil, Frey, and Embrechts (2015), a $d-$dimensional random vector $\boldsymbol{X}$ is said to follow a generalized hyperbolic (GH) distribution with index parameter $\lambda \in \mathbb{R}$, concentration parameters $\chi, \psi \in \mathbb{R}^+$, location vector $\boldsymbol{\mu}$, dispersion matrix $\boldsymbol{\Sigma}$, and skewness vector $\boldsymbol{\beta}$ if its pdf is given by

$$\begin{aligned} & h\left(\boldsymbol{x}; \lambda, \chi, \psi, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}\right) \\ &= \left[\frac{\chi + \delta(\boldsymbol{x}, \boldsymbol{\mu}; \boldsymbol{\Sigma})}{\psi + \boldsymbol{\beta}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta}}\right]^{\frac{\lambda - d/2}{2}} \frac{(\psi/\chi)^{\lambda/2} K_{\lambda - d/2}\left(\sqrt{(\psi + \delta(\boldsymbol{x}, \boldsymbol{\mu}; \boldsymbol{\Sigma}))(\omega + \boldsymbol{\beta}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta})}\right)}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2} K_\lambda(\sqrt{\chi\psi}) \exp\left[-(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta}\right]}, \end{aligned} \tag{5}$$

where $\delta(\boldsymbol{x}, \boldsymbol{\mu}; \boldsymbol{\Sigma}) = (\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})$ is the squared Mahalanobis distance between $\boldsymbol{x}$ and $\boldsymbol{\mu}$, and $K_\lambda(.)$ is the modified Bessel function of the third kind with index λ. In this formula, to ensure identifiability, it is necessary to fix $|\boldsymbol{\Sigma}| = 1$, which is restrictive in the context of model-based clustering. Thus, Browne and McNicholas (2015) proposed another parameterization of the GH distribution with index parameter $\lambda \in \mathbb{R}$, concentration parameter $\omega > 0$, location vector $\boldsymbol{\mu}$, dispersion matrix $\boldsymbol{\Sigma}$, and skewness vector $\boldsymbol{\beta}$, under which the pdf of $\boldsymbol{X}$ now becomes

$$\begin{aligned} & f_{\text{MGH}}\left(\boldsymbol{x}; \lambda, \omega, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}\right) \\ &= \left[\frac{\omega + \delta(\boldsymbol{x}, \boldsymbol{\mu}; \boldsymbol{\Sigma})}{\omega + \boldsymbol{\beta}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta}}\right]^{\frac{\lambda - d/2}{2}} \frac{K_{\lambda - d/2}\left(\sqrt{(\omega + \delta(\boldsymbol{x}, \boldsymbol{\mu}; \boldsymbol{\Sigma}))(\omega + \boldsymbol{\beta}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta})}\right)}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2} K_\lambda(\omega) \exp\left[-(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta}\right]}. \end{aligned} \tag{6}$$

For data generation purpose, $\boldsymbol{X}$ can be represented as

$$\boldsymbol{X} = \boldsymbol{\mu} + W\boldsymbol{\beta} + \sqrt{W}\boldsymbol{U}, \tag{7}$$

where $\boldsymbol{U} \perp W$, W follows a univariate generalized inverse Gaussian (GIG) distribution with parameters ω/η, $\omega\eta$, and λ, with $\eta = 1$, and $\boldsymbol{U} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$ follows a multivariate normal distribution with mean vector $\boldsymbol{0}_d$ and covariance matrix $\boldsymbol{\Sigma}$. This stochastic representation is also useful for parameter estimation. Since $\boldsymbol{X} \mid W = w$ follows a multivariate normal distribution with mean vector $\boldsymbol{\mu} + w\boldsymbol{\beta}$ and covariance matrix $w\boldsymbol{\Sigma}$, W can be treated as latent variable, and parameter estimation is simplified. It is worth pointing out that the GH distribution encompasses a wide range of distributions as special or limiting cases such as the multivariate skew-t, skew-normal, normal-inverse Gaussian, variance-gamma, asymmetric Laplace, and normal distributions (see McNeil *et al.* 2015).

Setting $f(\boldsymbol{x}\,;\boldsymbol{\vartheta}_g)$ in Equation 3 equal to Equation 6 leads to a G-component generalized hyperbolic mixture (GHM). To extend the proposed framework to incomplete data sets, Wei *et al.* (2019) outlined an EM procedure to fit the GHM that involves three sources of missing data:

1. Component memberships for all observations: $\boldsymbol{Z} = \{\boldsymbol{Z}_i\}_{i=1}^n$, where $\boldsymbol{Z}_i = (Z_{i1}, \cdots, Z_{iG})^\top$ so that $Z_{ig} = 1$ if observation i belongs to component g, otherwise $Z_{ig} = 0$.

2. The latent variables $\boldsymbol{W} = \{W_{ig}\}$, for $i = 1, \ldots, n$ and $g = 1, \ldots, G$, in which W_{ig} is assumed to follow a GIG distribution.

3. Missing values from each observation $\{\boldsymbol{X}_i^m\}_{i=1}^n$.

For more details on the algorithm, see Wei *et al.* (2019). It should be noted that although the GHM presents a flexible tool for modeling heterogeneous populations that exhibit skewness and heavy tails, it does not come with automatic outlier detection.

3.3. Other mixture models and extensions

Wei *et al.* (2019) also developed a framework for mixtures of skew-t distributions with missing data. The probability density function (pdf) of a $d-$dimensional random vector $\boldsymbol{X}$ that follows a skew-t (St) distribution is

$$\begin{aligned} & f_{\mathrm{St}}\left(\boldsymbol{x}; \nu, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}\right) \\ &= \left[\frac{\nu + \delta(\boldsymbol{x}, \boldsymbol{\mu}; \boldsymbol{\Sigma})}{\boldsymbol{\beta}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta}}\right]^{\frac{-\nu-d}{4}} \frac{\nu^{\nu/2} K_{(-\nu-d)/2}\left(\sqrt{(\nu + \delta(\boldsymbol{x}, \boldsymbol{\mu}; \boldsymbol{\Sigma}))(\boldsymbol{\beta}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta})}\right)}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2} \Gamma(\nu/2) 2^{\nu/2-1}(\omega) \exp\left[-(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta}\right]}, \end{aligned} \tag{8}$$

where $\Gamma(.)$ is the gamma function, $\nu > 0$ is the degree of freedom, and other parameters and functions have been defined previously. One advantage of the generalized hyperbolic (GH) distribution is that 15 distributions can be directly obtained as special or limiting cases; refer to Bagnato *et al.* (2024) for the hierarchy to obtain them. However, when using the parametrization in Equation 6, only six distributions can be obtained as special cases of the GH. Specifically, setting $\lambda = (d+1)/2$ in (6), we obtain the hyperbolic (H) distribution. From here, the additional constraint of $\boldsymbol{\beta} = \mathbf{0}$ leads to the symmetric hyperbolic (SH) distribution. On the other hand, setting $\lambda = 1/2$ in (6), we obtain the normal-inverse Gaussian (NIG) distribution, which then leads to the symmetric normal-inverse Gaussian (SNIG) distribution with the additional constraint of $\boldsymbol{\beta} = \mathbf{0}$. If the only constraint is $\boldsymbol{\beta} = \mathbf{0}$, the symmetric generalized hyperbolic (SGH) distribution is obtained, which then gives the hyperbolic univariate marginals (HUM) distribution if $\lambda = 1$.

Four more distributions can be obtained as special or limiting cases of the skew-t distribution in Equation 8. Starting from the skew-t pdf, setting $\nu = 1$, the skew-Cauchy (SC) distribution is obtained. Additionally, by adding $\boldsymbol{\beta} = \mathbf{0}$, we get a Cauchy (C) distribution. On the other hand, setting just $\boldsymbol{\beta} = \mathbf{0}$ instead, we obtain the t distribution. Furthermore, by adding $\nu \to \infty$, we can obtain the normal (N) distribution. Figure 1 contains a visual representation of the mentioned special and limiting cases.

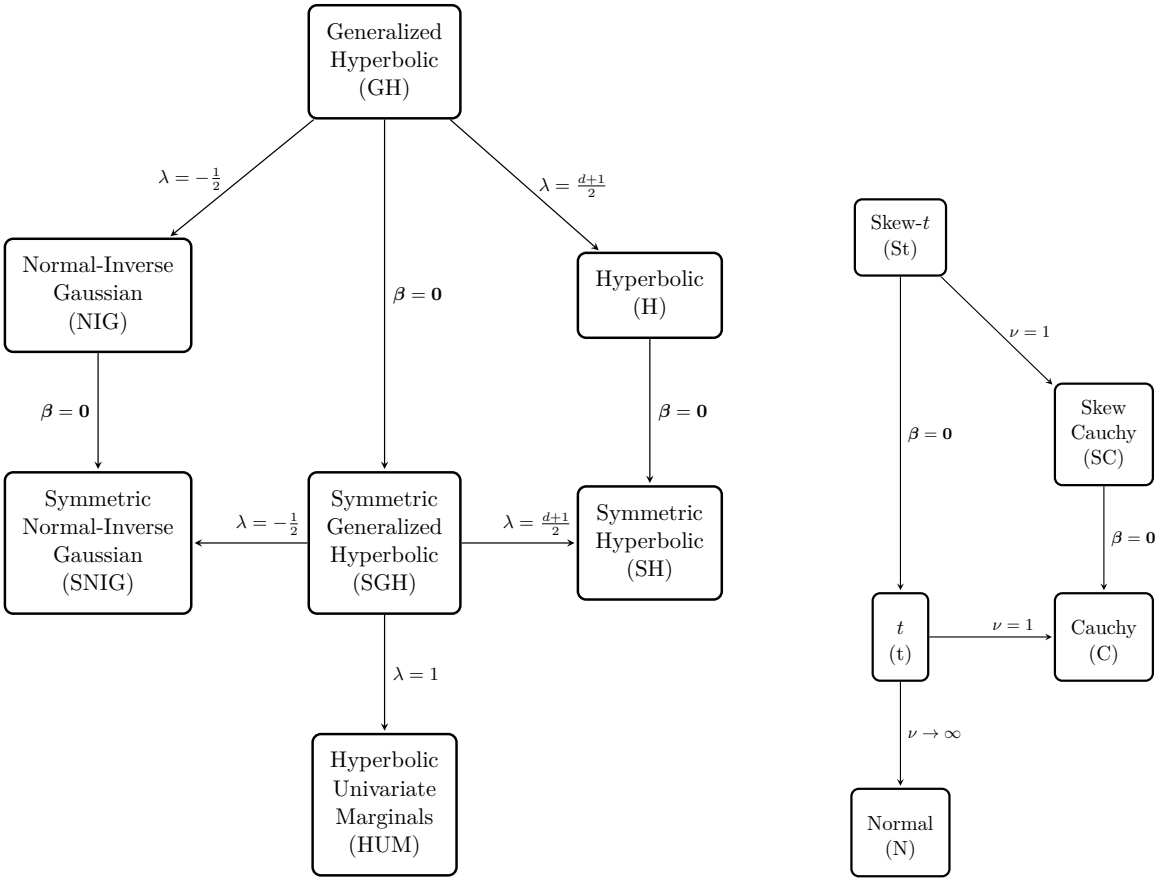
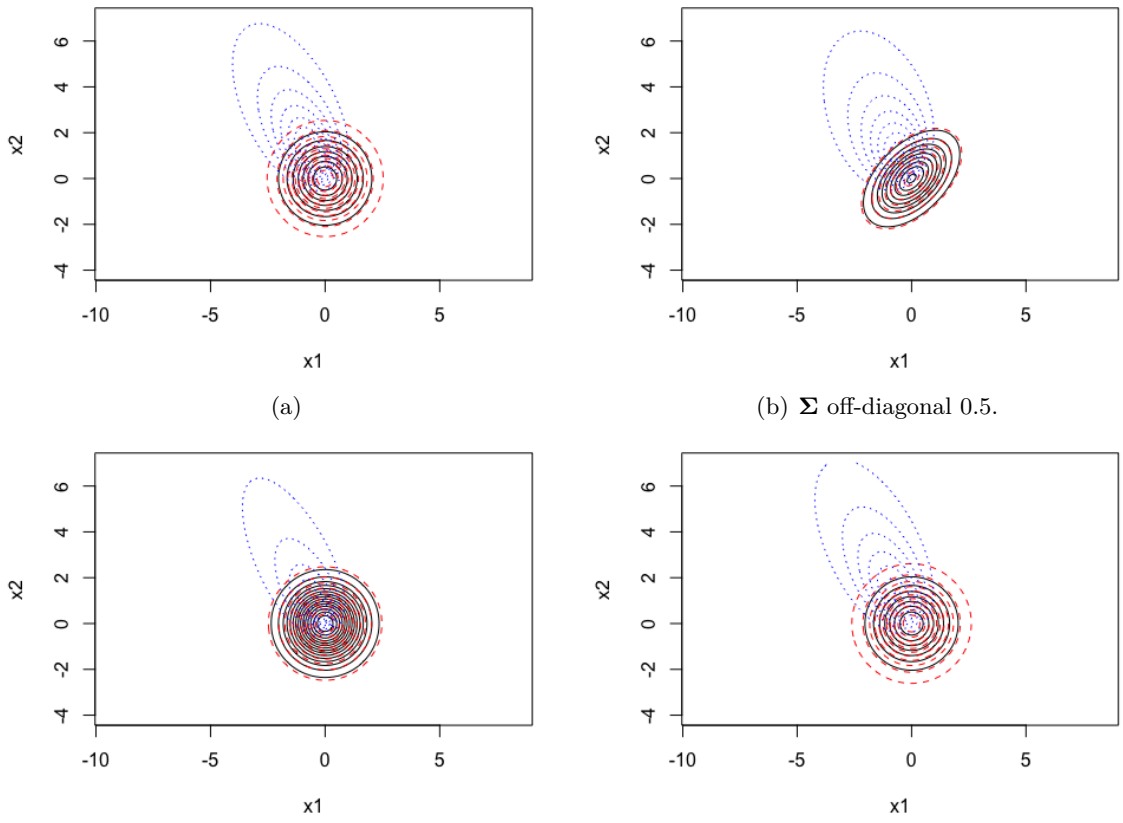


Figure 1: Visual representation of the special and limiting cases of the generalized hyperbolic and skew-t distributions.

In the R package **MixtureMissing**, mixture models utilizing all the 12 distributions in the GH hierarchy mentioned above for model-based clustering of data missing at random are available. This entails incorporating GH and skew-t, as proposed in Wei *et al.* (2019), along with the additional distributions – H, SH, NIG, SNIG, SGH, HUM, SC, C, t, and N – that have been developed and integrated into the package.

Figure 2 shows an example of possible shapes that can be obtained using a normal distribution (black solid line), contaminated normal (CN) distribution (red dashed line) and generalized hyperbolic (GH) distribution (blue dotted line). In Figure 2(a) $\boldsymbol{\mu} = (0, 0)$ and $\boldsymbol{\Sigma} = \mathbf{I}$. For the CN $\eta = 3$, $\alpha = 0.7$; for the GH $\omega = 1$, $\lambda = 0.5$, $\boldsymbol{\beta} = (-1, 2)$. In the other figures, one parameter per distribution changes. Specifically in Figure 2(b), we can see the effect of correlation, the off-diagonal elements of $\boldsymbol{\Sigma}$ are 0.5. In Figure 2(c) we can see the effect of η for the CN and ω for the GH, both impacting kurtosis. Similarly, in Figure 2(d) we see the effect of α for the CN and λ for the GH, both impacting kurtosis generating longer tails.



(a)

(b) $\boldsymbol{\Sigma}$ off-diagonal 0.5.

(c) CN $\eta = 6$; GH $\omega = 3$.

(d) CN $\alpha = 0.6$; GH $\lambda = 0.7$.

Figure 2: Contour plots of the density of normal (black solid line), contaminated normal (CN) (red dashed line), and generalized hyperbolic (GH) (blue dotted line) distributions with $\boldsymbol{\mu} = (0, 0)$. Unless otherwise specified, $\boldsymbol{\Sigma} = \mathbf{I}$; for the CN $\eta = 3$, $\alpha = 0.7$; for the GH $\omega = 1$, $\lambda = 0.5$, $\boldsymbol{\beta} = (-1, 2)$.

4. Computational aspects

4.1. Initialization

Biernacki, Celeux, and Govaert (2003) and Karlis and Xekalaki (2003) emphasized the importance of initialization in the EM algorithm because its solution can be dependent on the initial values. In the **MixtureMissing** package, the following initialization strategy is used:

1. If the data set is incomplete, apply mean imputation to the original data set to obtain a complete one.

2. Set mixing proportions equal across G components, that is, $\pi_g^{(0)} = 1/G$.

3. Perform a heuristic clustering method or user-defined labels as listed in Table 1 and assign $\boldsymbol{\mu}_g^{(0)}$ and $\boldsymbol{\Sigma}_g^{(0)}$ according to the resulting solution.

4. Initialize other parameters depending on the used distribution as follows

Method	Label	Description
K-medoids (default)	`"kmedoids"`	See Kaufman and Rousseeuw (1990)
K-means	`"kmeans"`	See Macqueen (1967)
Hierarchical clustering	`"hierarchical"`	See Ward (1963) for the implemented linkage
Gaussian mixture	`"mclust"`	See Scrucca *et al.* (2016)
User-defined	`"manual"`	Supplied by the user as a vector

Table 1: Implemented initialization criteria for component mean vectors and component dispersion matrices in the package **MixtureMissing**.

- Contaminated normal

$$\alpha_1^{(0)} = \cdots = \alpha_G^{(0)} = 0.6 \quad \text{and} \quad \eta_1^{(0)} = \cdots = \eta_G^{(0)} = 1.4.$$

- Skew-t

$$\boldsymbol{\beta}_1^{(0)} = \cdots = \boldsymbol{\beta}_G^{(0)} = \begin{pmatrix} 0.01 \\ \vdots \\ 0.01 \end{pmatrix} \quad \text{and} \quad \nu_1^{(0)} = \cdots = \nu_G^{(0)} = 10.$$

- t

$$\nu_1^{(0)} = \cdots = \nu_G^{(0)} = 10.$$

- Skew-Cauchy

$$\boldsymbol{\beta}_1^{(0)} = \cdots = \boldsymbol{\beta}_G^{(0)} = \begin{pmatrix} 0.01 \\ \vdots \\ 0.01 \end{pmatrix}.$$

- Generalized hyperbolic and other distributions

$$\lambda_1^{(0)} = \cdots = \lambda_G^{(0)} = -0.5, \quad \omega_1^{(0)}, \ldots, \omega_G^{(0)} = 1, \quad \text{and} \quad \boldsymbol{\beta}_1^{(0)} = \cdots = \boldsymbol{\beta}_G^{(0)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

4.2. Predicted cluster memberships, outlier detection, and imputation

The convergence in the EM algorithm is based on the Aitken acceleration (Aitken 1926); see McNicholas (2020) for more details. For all the used distributions, at the convergence of the EM algorithm, we can determine cluster memberships for all observations via the maximum *a posteriori* (MAP) probabilities.

The contaminated normal and t mixtures also have an outlier detection procedure. For the multivariate contaminated normal mixture, outlier detection is performed as follows. For each observation $\boldsymbol{X}_i$, let $\hat{v}_{ig}$ denote the value $P(V_{ig} = 1 \mid \boldsymbol{x}_i, Z_{ig} = 1, \boldsymbol{\Psi}^{(r)})$ at the convergence of the ECME algorithm. Then $\boldsymbol{X}_i$ is considered good with respect to the group g if $\boldsymbol{X}_i$

Code	Information criteria	Reference
`AIC`	Akaike information criterion	Akaike (1998)
`BIC`	Bayesian information criterion	Schwarz (1978)
`KIC`	Kullback information criterion	Cavanaugh (1999)
`KICc`	Corrected KIC	Seghouane and Bekara (2004)
`AIC3`	Modified AIC	Bozdogan (1993)
`CAIC`	Consistent AIC	Bozdogan (1987)
`AICc`	Corrected AIC	Hurvich and Tsai (1989)
`ICL`	Integrated completed likelihood	Biernacki, Celeux, and Govaert (2000)
`AWE`	Approximate weight of evidence	Banfield and Raftery (1993)
`CLC`	Classification likelihood criterion	Biernacki and Govaert (1997)

Table 2: String code of information criteria calculated in model-based clustering functions of the package **MixtureMissing**.

belongs to the component g and $\hat{v}_{ig} > 0.5$; otherwise, it is bad. It is important to note that outlier detection for a generic $\boldsymbol{X}_i$ can only be performed for the component g corresponding to the maximum $\hat{z}_{ig}$, that is, only for the cluster to which the point belongs. Herein, the threshold of 0.5 is based on the fact that $\hat{v}_{ig}$ is a probability. This procedure is said to be automatic because $\hat{v}_{ig}$ is obtained as a product of the ECME algorithm. For the t mixtures, Peel and McLachlan (2000) proposed an *a posteriori* procedure that requires a threshold to be specified in advance. Although Peel and McLachlan (2000) suggested the 95th percentile, a good choice for such percentile varies according to the application. Therefore, the procedure is rather subjective and not automatic.

4.3. Information criteria for model selection

The study of information criteria in model-based clustering aims to address situations where the number of clusters G is not given *a priori*; this scenario is also known as model selection. The idea is to run the assumed mixture model multiple times with different values of G, obtain the corresponding values of an information criterion, and select the G that optimizes such criterion. Table 2 lists all information criteria that can be obtained by running the model-based clustering functions in the package **MixtureMissing**. They can also be displayed in a tabular format by passing an object of class `MixtureMissing` into the function `summary()`. Readers can consult Brochado and Martins (2005), Akogul and Erisoglu (2016), or the references provided in Table 2 for their formulas.

5. The R package MixtureMissing

5.1. Overview

The R package **MixtureMissing** contains three functions to carry out model-based clustering of complete and incomplete data: `MCNM()` for multivariate contaminated normal mixture, `MGHM()` for multivariate generalized hyperbolic mixture, including its special or limiting cases, and `select_mixture()` for mixture model selection. They share arguments detailed in Table 3.

In addition, the specific argument for `MCNM()` is as follows.

- `eta_min`: A numeric value close to 1 to the right specifying the minimum value of $\eta_1, \ldots, \eta_G$; 1.001 by default.

Whereas, `MGHM()` takes the following specific arguments

- `model`: A string indicating the mixture model to be fitted; `"GH"` for generalized hyperbolic by default. The model can be any distribution in Table 4, except for the contaminated normal distribution.
- `outlier_cutoff`: A number between 0 and 1 indicating the percentile cut-off used for outlier detection. This is only relevant for t mixture.
- `deriv_ctrl`: A list containing arguments to control the numerical procedures for calculating the first and second derivatives. Some values are suggested by default. Refer to functions `grad()` and `hessian()` under the package `numDeriv` (Gilbert and Varadhan 2019) for more information.

The general function `select_mixture()` also takes `eta_min`, `outlier_cutoff`, and `deriv_ctrl` as described above, as well as

- `model`: A vector of character strings indicating the mixture model(s) to be fitted. Available distributions are given in Table 4. If not specified, all distributions will be considered by default.

`MCNM()` and `MGHM()` return an object of class `MixtureMissing` including the values described in Table 5. On top of these values, `MCNM()` has the following

- `alpha`: Component proportions of good observations.
- `eta`: Component degrees of contamination.
- `v_tilde`: An n by G matrix where each row indicates the expected probabilities that the corresponding observation is good with respect to each cluster.

`MGHM()` has the following

- `beta`: Component skewness vectors. Only available if `model` is `GH`, `NIG`, `SNIG`, `SC`, `SGH`, `HUM`, `H`, or `SH`; `NULL` otherwise.
- `lambda`: Component index parameters. Only available if `model` is `GH`, `NIG`, `SNIG`, `SGH`, `HUM`, `H`, or `SH`; `NULL` otherwise.
- `omega`: Component concentration parameters. Only available if `model` is `GH`, `NIG`, `SNIG`, `SGH`, `HUM`, `H`, or `SH`; `NULL` otherwise.
- `df`: Component degrees of freedom. Only available if `model` is `St` or `t`; `NULL` otherwise.

On the other hand, the output of `select_mixture()` is a list with three slots

Argument	Description
`X`	Matrix or data frame of dimensions $n \times d$.
`G`	An integer vector specifying the numbers of clusters, which must be at least 1.
`criterion`	A character string indicating the information criterion for model selection. BIC is used by default.
`max_iter`	A numeric value giving the maximum number of iterations each EM algorithm is allowed to use; 20 by default.
`epsilon`	A number specifying the epsilon value for the Aitken-based stopping criterion used in the EM algorithm: 0.01 by default.
`init_method`	A string specifying the method to initialize the EM algorithm. `"kmedoids"` clustering is used by default. Alternative methods include `"kmeans"`, `"hierarchical"`, `"mclust"`, and `"manual"`. When `"manual"` is chosen, a vector `clusters` of length n must be specified. If the data set is incomplete, missing values will be first filled based on the mean imputation method.
`clusters`	A numeric vector of length `n` that specifies the initial cluster memberships of the user when `init_method` is set to `"manual"`. This argument is `NULL` by default, so that it is ignored whenever other given initialization methods are chosen.
`progress`	A logical value indicating whether the fit progress should be displayed; `TRUE` by default.

Table 3: Common arguments used by three functions to carry out model-based clustering in the package **MixtureMissing**.

- `best_mod`: An object of class `MixtureMissing` corresponding to the best model.
- `all_mod`: A list of objects of class `MixtureMissing` corresponding to all models of consideration. The list is in the order of `model`.
- `criterion`: A numeric vector containing the chosen information criterion values of all models of consideration. The vector is in the order of the best-to-worst models.

The package also includes an extractor function, `extract()`, the user can specify the desired output using the argument `what`. When applied to the output of `select_model()`, it considers the best model by default. However, the user can specify a different model among all those fitted using the argument `m_code`. The package also includes some methods for `MixtureMissing` class objects: `plot()`, to display results in terms of scatter plots, parallel coordinate plots, and pairwise contour plots; `summary()`, to summarize the estimated parameters and further details; and `print()`, to provide a short description of the fitted model. Further details can be found in the functions' help pages.

For reproducibility, we fix the random number generator version in R.

```
R> suppressWarnings(RNGversion("3.5.0"))
```

Code	Distribution
`CN`	Contaminated normal
`GH`	Generalized hyperbolic
`NIG`	Normal-inverse Gaussian
`SNIG`	Symmetric normal-inverse Gaussian
`SC`	Skew-Cauchy
`C`	Cauchy
`St`	Skew-t
`t`	Student's t
`N`	Normal or Gaussian
`SGH`	Symmetric generalized hyperbolic
`HUM`	Hyperbolic univariate marginals
`H`	Hyperbolic
`SH`	Symmetric hyperbolic

Table 4: String codes for multivariate distributions implemented in the package **MixtureMissing**.

5.2. Simulating incomplete data sets

The package provides a convenient function `hide_values()` to introduce missing values to a multivariate data set under the MCAR mechanism. The function allows the user to specify either the proportion or the number of observations that contain missing values. It is important to note that, depending on the size of the data, the resulting data with missing values may not precisely match the specified proportion. Here is an example using the famous `Iris` data set (Fisher 1936).

```
R> library("MixtureMissing")
R> set.seed(123)
R> iris_80_cases <- hide_values(iris[1:4], n_cases = 80)
R> sum(!complete.cases(iris_80_cases))

[1] 80

R> head(iris_80_cases)

  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1         3.5          1.4          NA
2          4.9         3.0          1.4         0.2
3           NA          NA           NA         0.2
4          4.6         3.1          1.5         0.2
5          5.0          NA           NA          NA
6          5.4          NA           NA          NA
```

5.3. Multivariate contaminated normal mixture

In this section, we demonstrate the use of the function `MCNM()` to fit a multivariate contaminated normal mixture to the `UScost` data set available in the **MixtureMissing** package. The

Argument	Description
`model`	Matrix or data frame of dimensions $n \times d$.
`pi`	Mixing proportions.
`mu`	Component location vectors.
`Sigma`	Component dispersion matrices.
`z_tilde`	An n by G matrix where each row indicates the expected probabilities that the corresponding observation belongs to each cluster.
`clusters`	A numeric vector of length n indicating cluster memberships determined by the model.
`outliers`	A logical vector of length n indicating observations that are outliers.
`data`	The original data set if it is complete; otherwise, this is the data set with missing values imputed by appropriate expectations.
`complete`	An n by d logical matrix indicating which cells have no missing values.
`npar`	The breakdown of the number of parameters to estimate.
`max_iter`	Maximum number of iterations allowed in the EM algorithm.
`iter_stop`	The actual number of iterations needed when fitting the data set.
`final_loglik`	The final value of log-likelihood.
`loglik`	All the values of log-likelihood.
`AIC`	Akaike information criterion.
`BIC`	Bayesian information criterion.
`KIC`	Kullback information criterion.
`KICc`	Corrected Kullback information criterion.
`AIC3`	Modified AIC.
`CAIC`	Bozdogan's consistent AIC.
`AICc`	Small-sample version of AIC.
`ent`	Entropy.
`ICL`	Integrated Completed Likelihood criterion.
`AWE`	Approximate weight of evidence.
`CLC`	Classification likelihood criterion.
`init_method`	The initialization method used in model fitting.

Table 5: Common output values in the package **MixtureMissing**.

data set can also be retrieved from https://worldpopulationreview.com/ in the year 2019. It contains cost of living indices for five different categories: grocery, housing, transportation, utilities, and miscellaneous. These indices are calculated by first determining the average cost of living in the United States to be used as a baseline set at 100. The States, not including Washington, D.C., are then measured against this baseline. For example, a state with a cost of living index of 200 is twice as expensive as the national average. The data set is complete, so, for our purposes, we randomly introduce missing values to ten states. Moreover, to simplify the illustration, we consider costs under `Grocery`, `Housing`, and `Utilities`, which are variables 3 to 5 in the data and fix $G = 3$. The model is fitted as follows. As indicated in Table 3, we have `progress = TRUE` by default, so there are messages about different stages of model fitting. Moreover, the fitted mixture model's BIC is shown since BIC is the default information criterion.

```
R> data("UScost", package = "MixtureMissing")
R> set.seed(123)
R> X <- hide_values(UScost[3:5], n_cases = 10)
R> mod_CN3 <- MCNM(X, G = 3, init_method = "kmedoids", max_iter = 200)

Mixture: Contaminated Normal (CN)
Data Set: Incomplete
Initialization: kmedoids

Fitting G = 3 was successful with 21/200 iterations

The fitted mixture model with G = 3 has BIC = 1146.959
```

The package **MixtureMissing** includes the `plot()` function that allows the user to generate four model-based clustering plots interactively. Calling `plot(mod_CN3)` will open a menu of choices given below. The plot options include a pairwise scatter plot showing cluster memberships and highlighting outliers denoted by triangles, a pairwise scatter plot highlighting in red observations whose missing values are replaced by expectations obtained in the EM algorithm, a parallel plot of up to the first ten variables of the original data set, and a plot of estimated density in the form of contours. Figure 3 shows plots produced by calling the `plot()` function with the object `mod_CN3` and arguments to adjust point size, axis size, and line width.

```
R> plot(mod_CN3, cex.point = 1.8, cex.axis = 1.5, lwd = 2)

Model-based clustering plots:

1: classification
2: missing
3: parallel
4: density

Selection:
```

Another package capability is to summarize the results of each model-based clustering through the function `summary()`. Basic information regarding the observations with missing values, number of EM iterations, initialization method used, component frequency table, mixing proportions, component location vectors, component dispersion matrices, final log-likelihood value, total parameters, and information criteria will be displayed. For mixture models with a built-in outlier detection procedure like the contaminated normal and t mixtures, the total number of outliers as well as a breakdown of outliers per component will be included. If the model was fitted to an incomplete data set, the function will inform the number of observations with missing values. In particular, for the model fitted to the `UScost` data set above, we have

```
R> summary(mod_CN3)

Model: 3-Component Contaminated Normal Mixture with Incomplete Data
```

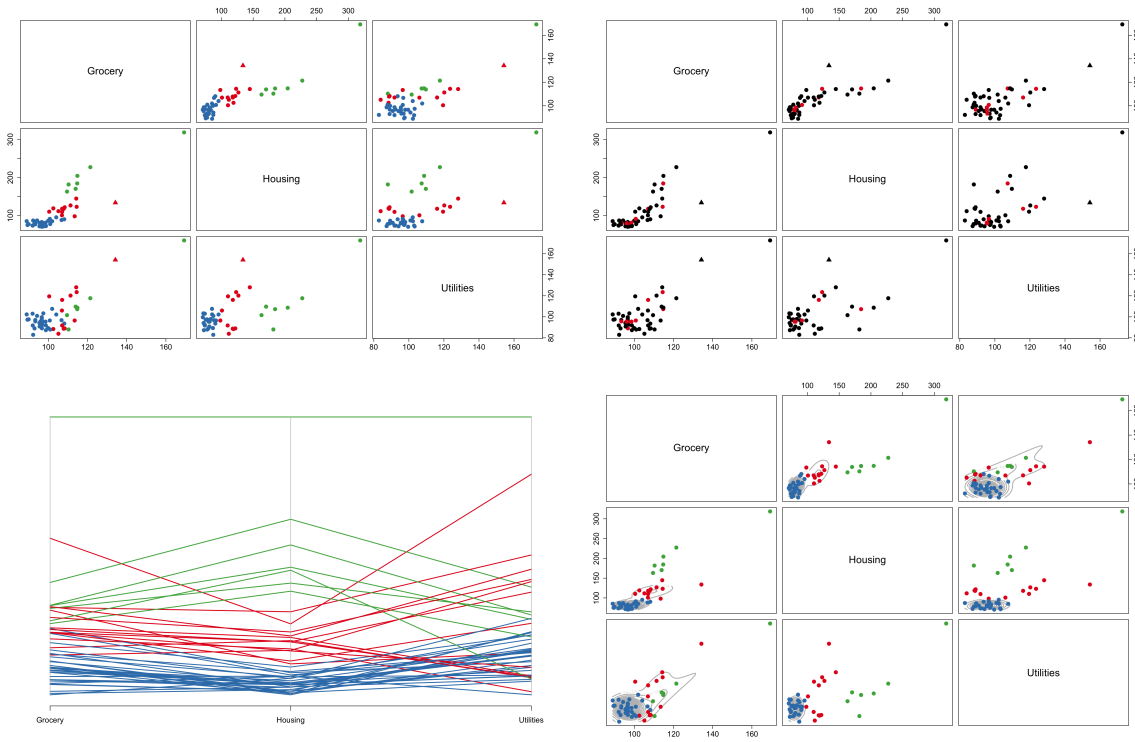


Figure 3: Plots produced by fitting the multivariate contaminated normal mixture to `UScost` data set. The first plot is a pairwise scatter plot showing cluster memberships and highlighting outliers denoted by triangles. The second plot is a pairwise scatter plot highlighting in red observations whose missing values are replaced by expectations obtained in the EM algorithm. The third plot is a parallel coordinate plot where each color represents a cluster. The last plot shows the estimated density in the form of contours.

```
Observations with missing values: 10 / 50

Missing values per variable:
  Grocery   Housing Utilities
        3         7         6

Iterations: 21 / 200

Initialization: kmedoids

Component frequency table:
 comp1 comp2 comp3
    30    13     7

Total outliers: 1

Outliers per component:
```

```
 comp1 comp2 comp3
     0     1     0

Mixing proportions:
    comp1     comp2     comp3
0.5768903 0.2721738 0.1509359

Component location vectors:
        Grocery   Housing Utilities
comp1  97.54573  81.27751  95.70827
comp2 107.32023 114.65967 106.90707
comp3 119.99895 205.75265 114.26036

Component location vectors:
        Grocery   Housing Utilities
comp1  96.65776  80.23405  95.42366
comp2 109.15298 116.37169 106.17192
comp3 121.80816 206.66333 114.89822

Component dispersion matrices:
, , comp1

              Grocery  Housing   Utilities
Grocery   22.87359540 14.88862 -0.04339625
Housing   14.88861638 48.28071  5.01427967
Utilities -0.04339625  5.01428 35.89938258

, , comp2

            Grocery   Housing Utilities
Grocery    69.01612  60.66522  118.0932
Housing    60.66522 190.24648  151.7993
Utilities 118.09318 151.79934  388.3211

, , comp3

           Grocery   Housing Utilities
Grocery   374.4364  910.3327  461.8503
Housing   910.3327 2398.3366 1122.6205
Utilities 461.8503 1122.6205  601.5073


Final log-likelihood: -505.0193

Total parameters: 35

Information Criteria:
```

```
      AIC      BIC      KIC     KICc          ICL      AWE      CLC
1080.039 1146.959 1118.039 1350.568 ...  1148.76 1392.481 1006.438
```

Note that we omitted some information criteria in the output due to the space constraint. The code below allows us to see which state(s) are considered outliers by the model. In this case, the model indicates Alaska, which is not surprising given Alaska's different geographical position compared to the other states.

```
R> UScost[which(mod_CN3$outliers), 1:5]

  Abbr  State Grocery Housing Utilities
2   AK Alaska   134.2   133.9     154.2
```

All the information displayed by the function `summary()` can also be accessed directly using `mod_CN3$` followed by the desired output's name; a list of outputs is available in Table 5. For more interesting insights, we can visualize the clustering solution on the map of the United States without Washington DC using the function `plot_usmap()` in the package **usmap** (Di Lorenzo 2023). Note that since `plot_usmap()` is based on the grammar of the package **ggplot2** (Wickham 2011), we also need to load **ggplot2** as well. To generate the map, consider the following code.

```
R> library("usmap")
R> library("ggplot2")
R> c_labs <- paste("Cluster", 1:3)
R> c_facs <- factor(mod_CN3$clusters, levels = 1:3, labels = c_labs)
R> dataplot <- data.frame(state = UScost$Abbr, cluster = c_facs)
R> plot_usmap(regions = "state", data = dataplot , values = "cluster",
+    labels = TRUE, exclude = "DC") +
+    theme(legend.position = "right", legend.title = element_blank()) +
+    scale_fill_manual(values = c("#377eb8", "#e41a1c", "#4daf4a"))
```

An integrated analysis of Figures 3 and 4 with the parameters of the fitted model provides valuable insights into the geographic distribution and characteristics of the identified clusters. Cluster 1 comprises economically affordable states; the blue lines in the parallel plot are in the bottom of the figure. From the map, we can see that the states in this cluster are predominantly in the center of the country. In contrast, cluster 2, characterized by the red lines in the middle top part of the parallel plot, is made up of more costly states, spanning the coasts and presenting Alaska, although as an outlier. In particular, Alaska is renowned for having an elevated utility index compared to other Cluster 2 states, coupled with a higher grocery index. Cluster 3 consists of states that are considered very expensive, notably Hawaii, New York, and California, in fact the green lines are on the top of the parallel plot. This cluster stands out because it exhibits the highest variability among the identified clusters, visible on both the parallel coordinate and the pairwise scatter plot,underscoring the diverse economic conditions within this category of states.

In the above example, we used K-medoids clustering for the EM initialization by specifying `init_method = "kmedoids"`. However, the other initialization methods listed in Table 1 are also available for the three main functions `MCNM()`, `MGHM()`, and `select_mixture()`. For

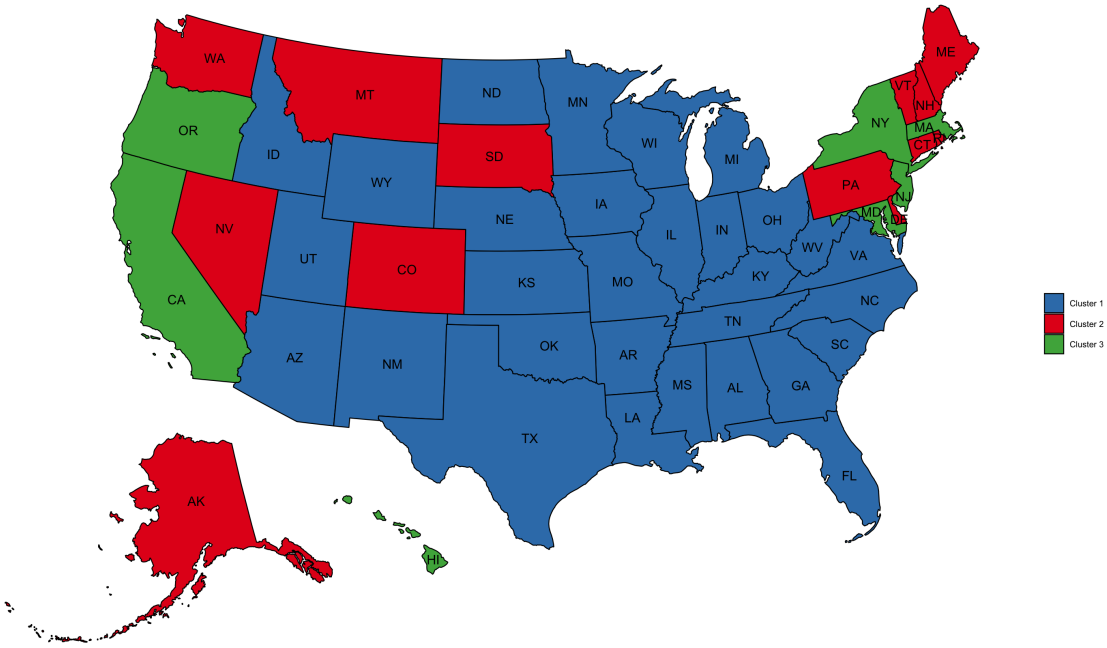


Figure 4: USA map with states colored based on the clustering partition.

example, initializing using a Gaussian mixture implemented in the package **mclust** (Scrucca *et al.* 2016) can be done as follows

```
R> mod_CN3_mc <- MCNM(X, G = 3, init_method = "mclust", max_iter = 200)
```

The user has the option to utilize predefined initialization methods by specifying `init_method = "manual"`. By selecting this option, the user can provide the initial cluster memberships by entering them as a numeric vector of length n into the `clusters` argument. For example, suppose the user believes that clusters 1, 2, and 3 consist of the first 20 states, the second 20 states, and the last 10 states, respectively. Then, manual initialization can be done as follows

```
R> cls0 <- c(rep(1, 20), rep(2, 20), rep(3, 10))

R> mod_CN3_user <- MCNM(X, G = 3, init_method = "manual", clusters = cls0,
+    max_iter = 200)
```

5.4. Multivariate generalized hyperbolic mixture

Herein, we demonstrate the use of the function `MGHM()` to fit a multivariate generalized hyperbolic mixture, as well as its special and limiting cases, to the data set `bankruptcy` (Altman 1968). The data set is also available in the **MixtureMissing** package. It contains the ratio of retained earnings (RE) to total assets, and the ratio of earnings before interests and taxes (EBIT) to total assets of 66 American firms recorded in the form of ratios. Half of the selected firms had filed for bankruptcy. There are no missing values in the data set, so we

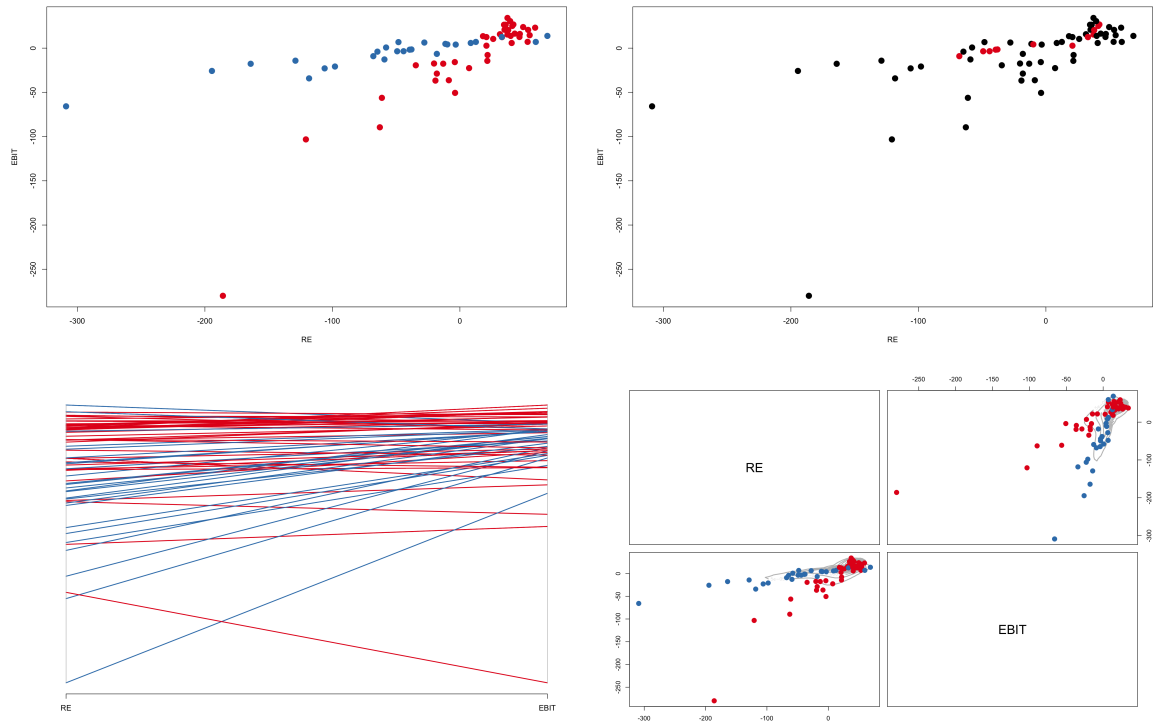


Figure 5: Plots produced by fitting the multivariate generalized hyperbolic mixture to `bankruptcy` data set. The first plot is a pairwise scatter plot showing cluster memberships and highlighting outliers denoted by triangles. The second plot is a pairwise scatter plot highlighting in red observations whose missing values are replaced by expectations obtained in the EM algorithm. The third plot is a parallel coordinate plot where each color represents a cluster. The last plot is a plot of estimated density in the form of contours.

randomly select 20% observations and hide some of their values. For ease of presentation, we do not show fitting progress by setting `progress = FALSE` and as well as the resulting summary. The following code fits the model and generates the plots shown in Figure 5.

```
R> set.seed(12345)
R> X <- hide_values(bankruptcy[, 2:3], prop_cases = 0.2)
R> mod_GH <- MGHM(X, G = 2, model = "GH", init_method = "kmedoids",
+    max_iter = 200, progress = FALSE)
R> plot(mod_GH, cex.point = 1.8, lwd = 2)
```

The estimated parameters π_g, $\boldsymbol{\mu}_g$, and $\boldsymbol{\beta}_g$ are displayed below. For simplicity, we do not display $\boldsymbol{\Sigma}_g$, λ_g, and ω_g that can be obtained with an analogous syntax. A joint analysis of the parameters and Figure 5 shows that the two clusters have a similar proportion of observations and a similar $\boldsymbol{\mu}_g$. Both clusters are negatively skewed in both dimensions. They differ in the direction of the tails, with cluster one including firms with proportional values in the two ratios. The difference in the tails is evident from the scatter plots. In the parallel coordinate plot, we see that the majority of the points overlap at the top of the figure. Some of the blue lines have lower RE, while some of the red lines are lower for both ratios.

```
R> mod_GH$pi

    comp1     comp2
0.4001022 0.5998978

R> mod_GH$mu

            RE     EBIT
comp1 61.90255 23.14936
comp2 53.69775 34.83119

R> mod_GH$beta

               RE       EBIT
comp1 -480.62946 -124.33229
comp2  -40.00375  -40.63468
```

To fit mixture models in which component distributions are special or limiting cases of the generalized hyperbolic distribution, we can simply specify the desired distribution in the argument `model` of `MGHM()`. Note that `MGHM()` can fit any distribution given in Table 4 except for the contaminated normal distribution, i.e., `model = "CN"` is not possible. For example, we can fit mixture models via the skew-t, symmetric normal-inverse Gaussian, and normal distributions as follows.

```
R> mod_St <- MGHM(X, G = 2, model = "St", init_method = "kmedoids",
+    max_iter = 200, progress = FALSE)
R> mod_SNIG <- MGHM(X, G = 2, model = "SNIG", init_method = "kmedoids",
+    max_iter = 200, progress = FALSE)
R> mod_N <- MGHM(X, G = 2, model = "N", init_method = "kmedoids",
+    max_iter = 200, progress = FALSE)
```

5.5. Mixture model selection

Although some researchers may have a distribution in mind for a specific data set, in many cases, the choice is not straightforward. For example, if the interest is in outlier detection using a symmetric distribution, the contaminated normal and t distributions would be good choices. In general, the best-fit distribution can be determined using one of the information criteria listed in Table 2. The package **MixtureMissing** provides the function `select_mixture()` to determine which distribution is the best fit for the given data according to an information criterion. To demonstrate, we consider the `Automobile` data set, contributed by Jeffrey C. Schlimmer, from the UCI Machine Learning data repository (https://archive.ics.uci.edu/ml/datasets/automobile). This real data set consisting of specifications of 205 autos with some missing values is included in the **MixtureMissing** and was previously analyzed by Tong and Tortora (2022) using the multivariate contaminated normal mixture. The first 15 variables are continuous, and among these, only 6 variables contain missing values: `normalized_losses`, `bore`, `stroke`, `horsepower`, `peak_rpm`, and `price`. The following code gives the number of missing values per continuous variable.

```
R> data("auto", package = "MixtureMissing")
R> sapply(auto[, 1:15], function(a) sum(is.na(a)))

normalized_losses        wheel_base            length
               41                 0                 0
            width            height       curb_weight
                0                 0                 0
      engine_size              bore            stroke
                0                 4                 4
compression_ratio        horsepower          peak_rpm
                0                 2                 2
         city_mpg       highway_mpg             price
                0                 0                 4
```

Herein, for our discussion, we consider these 6 variables only. There are 45 autos with partially-observed records, as shown below.

```
R> vars_miss <- c("normalized_losses", "bore", "stroke", "horsepower",
+    "peak_rpm", "price")
R> X <- auto[, vars_miss]
R> sum(!complete.cases(X))

45
```

Since Tong and Tortora (2022) found two clusters of autos, we also fix `G = 2` in this case. The arguments of `select_mixture()` are similar to those of `MCNM()` and `MGHM()`. However, to run `select_mixture()`, we need to specify one information criterion among those shown in Table 2. By default, all 13 distributions in Table 4 are fitted. For example, using the Bayesian information criterion (BIC) and fitting the models to the above data set, the function shows the information criterion value associated with each model and recommends the optimal distribution to be the contaminated normal.

```
R> mod_auto1 <- select_mixture(X, G = 2, criterion = "BIC",
+    init_method = "kmedoids", max_iter = 200)

Data Set: Incomplete

Information Criterion: BIC

Initialization: kmedoids

Model Fitting:
  CN   GH   NIG   SNIG   SC   C   St   t   N   SGH  HUM   H   SH

According to BIC, the best mixture model is based on the Contaminated
Normal distribution
```

```
Model rank according to BIC:
  1. Contaminated Normal: 10489.11
  2. t: 10492.61
  3. Symmetric Hyperbolic: 10504.03
  4. Normal-Inverse Gaussian: 10512.29
  5. Hyperbolic: 10516.73
  6. Generalized Hyperbolic: 10521.75
  7. Skew-t: 10522.31
  8. Hyperbolic Univariate Marginals: 10529.66
  9. Symmetric Normal-Inverse Gaussian: 10529.81
  10. Symmetric Generalized Hyperbolic: 10540.45
  11. Skew-Cauchy: 10604.35
  12. Normal: 10619.21
  13. Cauchy: 10697.46
```

Note that a different criterion can be used by changing the argument `criterion`. In addition, we can also consider only a subset of distributions in any order by providing a vector of string codes to the argument `model`. If there are duplicate string codes, the corresponding model will not be fitted again. Below, we use the criterion to be integrated completed likelihood (ICL) and the distributions to be t, skew-Cauchy (SC), symmetric generalized hyperbolic (SGH), and normal-inverse Gaussian (NIG). In this case, the t mixture turns out to be the best.

```
R> mod_auto2 <- select_mixture(X, G = 2, criterion = "ICL",
+    model = c("t", "SC", "SGH", "NIG"),
+    init_method = "kmedoids", max_iter = 200)

Data Set: Incomplete

Information Criterion: ICL

Initialization: kmedoids

Model Fitting:
  t    SC    SGH    NIG

According to ICL, the best mixture model is based on the t distribution

Model rank according to ICL:
  1. t: 10493.85
  2. Normal-Inverse Gaussian: 10515.03
  3. Symmetric Generalized Hyperbolic: 10545.58
  4. Skew-Cauchy: 10606.54
```

The function `print()` can be used to provide short descriptions of the fitted models. The extractor function by default would extract values from the best model, but the argument `m_code` can be used to select a different model. For example, one could see the information criteria obtained with the skew-Cauchy mixture (model code to be `"SC"`) using the following code.

```
R> extract(mod_auto2, what = "information", m_code = "SC")

G    Loglik Parameters      AIC      BIC      KIC     KICc     AIC3     CAIC
2 -5123.856         67 10381.71 10604.35 10451.71 10535.33 10448.71 10671.35
    AICc      ICL      AWE      CLC
10448.22 10606.54 11166.38 10243.33
```

6. Conclusion

This paper guides readers through the main functionalities of the R package **MixtureMissing** and extends model-based clustering with missing values to eight new distributions in the generalized hyperbolic family. Overall, the package offers a wide range of distributions for model-based clustering of continuous data, with the two main ones being contaminated normal and generalized hyperbolic. Additionally, the package includes 11 special or limiting cases of the generalized hyperbolic distribution; some well-known distributions in this category are the normal, t, and skew-t. When conducting cluster analysis with **MixtureMissing**, users have the flexibility to choose a specific distribution or determine the best-fitting one using different information criteria. The functions apply to both complete and incomplete data sets. Furthermore, they are all equipped with informative summaries and visualization tools to facilitate investigations of the data. While the package currently implements an extensive number of models, future extensions can be achieved by incorporating recently developed models such as the multiple scaled contaminated normal mixture (Tong and Tortora 2024).

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2209974 (Tortora).

References

Aitken AC (1926). "A Series Formula for the Roots of Algebraic and Transcendental Equations." *Proceedings of the Royal Society of Edinburgh*, **45**(1), 14–22. doi:10.1017/s0370164600024871.

Akaike H (1998). "Information Theory and an Extension of the Maximum Likelihood Principle." In E Parzen, K Tanabe, G Kitagawa (eds.), *Selected Papers of Hirotugu Akaike*, pp. 199–213. Springer-Verlag, New York. doi:10.1007/978-1-4612-1694-0_15.

Akogul S, Erisoglu M (2016). "A Comparison of Information Criteria in Clustering Based on Mixture of Multivariate Normal Distributions." *Mathematical and Computational Applications*, **21**(3), 34. doi:10.3390/mca21030034.

Altman EI (1968). "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy." *The Journal of Finance*, **23**(4), 589–609. doi:10.2307/2978933.

Andrews JL, Wickins JR, Boers NM, McNicholas PD (2018). "**teigen**: An R Package for Model-Based Clustering and Classification via the Multivariate t Distribution." *Journal of Statistical Software*, **83**(7), 1–32. doi:10.18637/jss.v083.i07.

Bagnato L, Farcomeni A, Punzo A (2024). "The Generalized Hyperbolic Family and Automatic Model Selection through the Multiple-Choice LASSO." *Statistical Analysis and Data Mining*, **17**(1), e11652. doi:10.1002/sam.11652.

Banfield JD, Raftery AE (1993). "Model-Based Gaussian and Non-Gaussian Clustering." *Biometrics*, **49**(3), 803. doi:10.2307/2532201.

Basagaña X, Barrera-Gómez J, Benet M, Antó JM, Garcia-Aymerich J (2013). "A Framework for Multiple Imputation in Cluster Analysis." *International Journal of Epidemiology*, **177**(7), 718–725. doi:10.1093/aje/kws289.

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.

Bergé L, Bouveyron C, Girard S (2012). "**HDclassif**: An R Package for Model-Based Clustering and Discriminant Analysis of High-Dimensional Data." *Journal of Statistical Software*, **46**(6), 1–29. doi:10.18637/jss.v046.i06.

Biernacki C, Celeux G, Govaert G (2000). "Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(7), 719–725. doi:10.1109/34.865189.

Biernacki C, Celeux G, Govaert G (2003). "Choosing Starting Values for the EM Algorithm for Getting the Highest Likelihood in Multivariate Gaussian Mixture Models." *Computational Statistics & Data Analysis*, **41**(3), 561–575. doi:10.1016/s0167-9473(02)00163-9.

Biernacki C, Celeux G, Govaert G, Langrognet F (2006). "Model-Based Cluster and Discriminant Analysis with the **MIXMOD** Software." *Computational Statistics & Data Analysis*, **51**(2), 587–600. doi:10.1016/j.csda.2005.12.015.

Biernacki C, Govaert G (1997). "Using the Classification Likelihood to Choose the Number of Clusters." In EJ Wegman, SP Azen (eds.), *Computing Science and Statistics*, volume 29, pp. 451–457.

Bishop CM, Nasrabadi NM (2006). *Pattern Recognition and Machine Learning*, volume 4. Springer-Verlag.

Bouveyron C (2021). ***funFEM**: Clustering in the Discriminative Functional Subspace*. doi:10.32614/CRAN.package.funfem. R package version 1.2.

Bozdogan H (1987). "Model Selection and Akaike's Information Criterion (AIC): The General Theory and Its Analytical Extensions." *Psychometrika*, **52**(3), 345–370. ISSN 1860-0980. doi:10.1007/bf02294361.

Bozdogan H (1993). "Choosing the Number of Component Clusters in the Mixture-Model Using a New Informational Complexity Criterion of the Inverse-Fisher Information Matrix."

In O Opitz, B Lausen, R Klar (eds.), *Information and Classification*, pp. 40–54. Springer-Verlag, Heidelberg.

Brochado A, Martins F (2005). "Assessing the Number of Components in Mixture Models: A Review." *Fep working papers*, Universidade Do Porto, Faculdade De Economia Do Porto.

Browne RP, McNicholas PD (2015). "A Mixture of Generalized Hyperbolic Distributions." *Canadian Journal of Statistics*, **43**(2), 176–198. doi:10.1002/cjs.11246.

Byers S, Raftery AE (1998). "Nearest-Neighbor Clutter Removal for Estimating Features in Spatial Point Processes." *Journal of the American Statistical Association*, **93**(442), 577–584. doi:10.1080/01621459.1998.10473711.

Cavanaugh JE (1999). "A Large-Sample Model Selection Criterion Based on Kullback's Symmetric Divergence." *Statistics & Probability Letters*, **42**(4), 333–343. doi:10.1016/s0167-7152(98)00200-4.

Corradin R, Canale A, Nipoti B (2021). "**BNPmix**: An R Package for Bayesian Nonparametric Modeling via Pitman-Yor Mixtures." *Journal of Statistical Software*, **100**(15), 1–33. doi:10.18637/jss.v100.i15.

Cuesta-Albertos JA, Gordaliza A, Matran C (1997). "Trimmed k-Means: An Attempt to Robustify Quantizers." *The Annals of Statistics*, **25**(2), 553–576. doi:10.1214/aos/1031833664.

Davies L, Gather U (1993). "The Identification of Multiple Outliers." *Journal of the American Statistical Association*, **88**(423), 782–792. doi:10.1080/01621459.1993.10476339.

Dempster AP, Laird NM, Rubin DB (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society B*, **39**(1), 1–22.

Di Lorenzo P (2023). ***usmap***: *US Maps Including Alaska and Hawaii*. doi:10.32614/CRAN.package.usmap. R package version 0.6.3.

Everitt B, Landau S, Leese M, Stahl D (2011). *Cluster Analysis*. John Wiley & Sons, Chichester.

Fisher RA (1936). "The Use of Multiple Measurements in Taxonomic Problems." *Annals of Eugenics*, **7**(2), 179–188. doi:10.1111/j.1469-1809.1936.tb02137.x.

Fraley C, Raftery AE (2002). "Model-Based Clustering, Discriminant Analysis, and Density Estimation." *Journal of the American Statistical Association*, **97**(458), 611–631. doi:10.1198/016214502760047131.

Fritz H, Garcia-Escudero LA, Mayo-Iscar A (2012). "**tclust**: An R Package for a Trimming Approach to Cluster Analysis." *Journal of Statistical Software*, **47**(12), 1–26. doi:10.18637/jss.v047.i12.

García-Escudero LA, Gordaliza A, Matrán C, Mayo-Iscar A (2008). "A General Trimming Approach to Robust Cluster Analysis." *The Annals of Statistics*, **36**(3), 1324–1345. doi:10.1214/07-aos515.

Ghahramani Z, Jordan MI (1994). "Learning from Incomplete Data." *Technical report*, Defense Technical Information Center, Fort Belvoir. doi:10.21236/ada295618.

Gilbert P, Varadhan R (2019). ***numDeriv****: Accurate Numerical Derivatives*. doi:10.32614/CRAN.package.numderiv. R package version 2016.8-1.1.

Grün B, Plummer M (2023). ***bayesmix****: Bayesian Mixture Models with* ***JAGS***. doi:10.32614/CRAN.package.bayesmix. R package version 2016.8-1.1.

Hawkins DM (1980). *Identification of Outliers*. Springer-Verlag, Dordrecht. doi:10.1007/978-94-015-3994-4.

Hennig C (2015). "What Are the True Clusters?" *Philosophical Aspects of Pattern Recognition*, **64**, 53–62. doi:10.1016/j.patrec.2015.04.009.

Hurvich CM, Tsai CL (1989). "Regression and Time Series Model Selection in Small Samples." *Biometrika*, **76**(2), 297–307. doi:10.1093/biomet/76.2.297.

Josse J, Mayer I, Tierney N, Vialaneix N (2025). "CRAN Task View: Missing Data." Version 2025-09-24, URL https://CRAN.R-project.org/view=MissingData.

Karlis D, Xekalaki E (2003). "Choosing Initial Values for the EM Algorithm for Finite Mixtures." *Computational Statistics & Data Analysis*, **41**, 577–590. doi:10.1016/s0167-9473(02)00177-9.

Kaufman L, Rousseeuw PJ (eds.) (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Hoboken.

Koenker R, Gu J (2017). "**REBayes**: An R Package for Empirical Bayes Mixture Methods." *Journal of Statistical Software*, **82**(8), 1–26. doi:10.18637/jss.v082.i08.

Kubicki V, Biernacki C, Grimonprez Q, Marbac-Lourdelle M, Goffinet E, Iovleff S, Vandaele J (2023). "**RMixtComp**: Mixture Models with Heterogeneous and (Partially) Missing Data." doi:10.32614/CRAN.package.rmixtcomp. Version 4.1.4.

Lebret R, Iovleff S, Langrognet F, Biernacki C, Celeux G, Govaert G (2015). "**Rmixmod**: The R Package of the Model-Based Unsupervised, Supervised, and Semi-Supervised Classification **MIXMOD** Library." *Journal of Statistical Software*, **67**, 1–29. doi:10.18637/jss.v067.i06.

Lee SX, McLachlan GJ (2013). "On Mixtures of Skew Normal and Skew t-Distributions." *Advances in Data Analysis and Classification*, **7**(3), 241–266. doi:10.1007/s11634-013-0132-8.

Little RJA, Rubin DB (2020). *Statistical Analysis with Missing Data*. 3rd edition. John Wiley & Sons, Hoboken.

Liu C, Rubin DB (1994). "The ECME Algorithm: A Simple Extension of EM and ECM with Faster Monotone Convergence." *Biometrika*, **81**(4), 633–648. doi:10.1093/biomet/81.4.633.

Macqueen J (1967). "Some Methods for Classification and Analysis of Multivariate Observations." In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 281–297. University of California Press.

Maitra R, Ramler IP (2009). "Clustering in the Presence of Scatter." *Biometrics*, **65**(2), 341–352. doi:10.1111/j.1541-0420.2008.01064.x.

Marbac M, Sedki M (2019). "**VarSelLCM**: An R/C++ Package for Variable Selection in Model-Based Clustering of Mixed-Data with Missing Values." *Bioinformatics*, **35**(7), 1255–1257. doi:10.1093/bioinformatics/bty786.

Martinez AR, Martinez WL (2003). "Model-Based Clustering Toolbox for MATLAB." Naval Surface Warfare Center, Dahlgren Division, URL http://cda.psych.uiuc.edu/matlab_class/martinez/edatoolbox/Docs/MBCDocs.pdf.

McCaw Z (2023). "**MGMM**: Missingness Aware Gaussian Mixture Models." doi:10.32614/CRAN.package.mgmm. R package Version 1.0.1.1.

McLachlan G, Peel D (2000). *Finite Mixture Models*. John Wiley & Sons, Hoboken.

McNeil AJ, Frey R, Embrechts P (2015). *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton University Press.

McNicholas PD (2016). "Model-Based Clustering." *Journal of Classification*, **33**(3), 331–373. doi:10.1007/s00357-016-9211-9.

McNicholas PD (2020). *Mixture Model-Based Classification*. Chapman & Hall/CRC.

McNicholas PD, ElSherbiny A, McDaid AF, Murphy TB (2021). ***pgmm**: Parsimonious Gaussian Mixture Models*. doi:10.32614/CRAN.package.pgmm. R package version 1.2.5.

Murray PM, Browne RP, McNicholas PD (2014). "Mixtures of Skew-t Factor Analyzers." *Computational Statistics & Data Analysis*, **77**, 326–335. doi:10.1016/j.csda.2014.03.012.

Papastamoulis P, Rattray M (2017). "**BayesBinMix**: An R Package for Model Based Clustering of Multivariate Binary Data." *The R Journal*, **9**(1), 403–420. doi:10.32614/rj-2017-022.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011). "**scikit-learn**: Machine Learning in Python." *Journal of Machine Learning Research*, **12**, 2825–2830.

Peel D, McLachlan GJ (2000). "Robust Mixture Modelling Using the t Distribution." *Statistics and Computing*, **10**(4), 339–348. doi:10.1023/a:1008981510081.

Pfaffel O (2020). "**ClustImpute**: An R Package for k-Means Clustering with Build-In Missing Data Imputation." Preprint. doi:10.13140/rg.2.2.20143.36007.

Pocuca N, Browne RP, McNicholas PD (2021). ***mixture**: Mixture Models for Clustering and Classification*. doi:10.32614/CRAN.package.mixture. R package version 2.0.4.

Punzo A, Mazza A, McNicholas PD (2018). "**ContaminatedMixt**: An R Package for Fitting Parsimonious Mixtures of Multivariate Contaminated Normal Distributions." *Journal of Statistical Software*, **85**(10), 1–25. doi:10.18637/jss.v085.i10.

Punzo A, McNicholas PD (2016). "Parsimonious Mixtures of Multivariate Contaminated Normal Distributions." *Biometrical Journal*, **58**(6), 1506–1537. doi:10.1002/bimj.201500144.

Ritter G (2014). *Robust Cluster Analysis and Variable Selection*. Chapman & Hall/CRC. doi:10.1201/b17353.

Schafer JL (2017). ***mix**: Estimation/Multiple Imputation for Mixed Categorical and Continuous Data)*. doi:10.32614/CRAN.package.mix. R package version 1.0-10.

Schwarz G (1978). "Estimating the Dimension of a Model." *The Annals of Statistics*, **6**(2), 461–464. doi:10.1214/aos/1176344136.

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**(1), 289–317. doi:10.32614/rj-2016-021.

Seghouane AK, Bekara M (2004). "A Small Sample Model Selection Criterion Based on Kullback's Symmetric Divergence." *IEEE Transactions on Signal Processing*, **52**(12), 3314–3323. doi:10.1109/tsp.2004.837416.

Selosse M, Jacques J, Biernacki C (2020). "**ordinalClust**: An R Package to Analyze Ordinal Data." *The R Journal*, **12**(2), 61–81. doi:10.32614/rj-2021-011.

Singleton M (2023). "**mixmod**: An Implementation of Mixture Models with Robust Fitting Methods for a Fixed Set of Components." Version 0.2.0, URL https://pypi.org/project/mixmod/.

Stahl D, Sallis H (2012). "Model-Based Cluster Analysis." *WIREs Computational Statistics*, **4**(4), 341–358. doi:10.1002/wics.1204.

The MathWorks Inc (2025). "What Is Clustering? Clustering and Its Applications Explained." URL https://www.mathworks.com/discovery/clustering.html.

Tomer O (2020). "**student-mixture**: A Package for Fitting a Student's t-Mixture Model." Version 0.1.14, URL https://pypi.org/project/student-mixture/.

Tong H, Tortora C (2022). "Model-Based Clustering and Outlier Detection with Missing Data." *Advances in Data Analysis and Classification*, **16**(1), 5–30. doi:10.1007/s11634-021-00476-1.

Tong H, Tortora C (2024). "Missing Values and Directional Outlier Detection in Model-Based Clustering." *Journal of Classification*, **41**, 480–513. doi:10.1007/s00357-023-09450-2.

Tortora C, Browne RP, El Sherbiny A, Franczak BC, McNicholas PD (2021). "Model-Based Clustering, Classification, and Discriminant Analysis Using the Generalized Hyperbolic Distribution: **MixGHD** R Package." *Journal of Statistical Software*, **98**(3), 1–24. doi:10.18637/jss.v098.i03.

Tseng GC, Wong WH (2005). "Tight Clustering: A Resampling-Based Approach for Identifying Stable and Tight Patterns in Data." *Biometrics*, **61**(1), 10–16. doi:10.1111/j.0006-341x.2005.031032.x.

Van Buuren S (2021). *Flexible Imputation of Missing Data*. 2nd edition. Chapman & Hall/CRC, Boca Raton.

Wallace ML, Buysse DJ, Germain A, Hall MH, Iyengar S (2018). "Variable Selection for Skewed Model-Based Clustering: Application to the Identification of Novel Sleep Phenotypes." *Journal of the American Statistical Association*, **113**(521), 95–110. doi:10.1080/01621459.2017.1330202.

Wang H, Zhang Q, Luo B, Wei S (2004). "Robust Mixture Modelling Using Multivariate t-Distribution with Missing Information." *Pattern Recognition Letters*, **25**(6), 701–710. doi:10.1016/j.patrec.2004.01.010.

Ward JH (1963). "Hierarchical Grouping to Optimize an Objective Function." *Journal of the American Statistical Association*, **58**(301), 236–244. doi:10.1080/01621459.1963.10500845. Publisher: Taylor & Francis.

Wei Y, Tang Y, McNicholas PD (2019). "Mixtures of Generalized Hyperbolic Distributions and Mixtures of Skew-t Distributions for Model-Based Clustering with Incomplete Data." *Computational Statistics & Data Analysis*, **130**, 18–41. doi:10.1016/j.csda.2018.08.016.

Wickham H (2011). "**ggplot2**." *Wiley Interdisciplinary Reviews: Computational Statistics*, **3**(2), 180–185. doi:10.1002/wics.147.

Affiliation:

Hung Tong
Department of Mathematics
Rowan University
Glassboro, New Jersey, United States of America, 08028
E-mail: tong@rowan.edu

Cristina Tortora
Department of Mathematics and Statistics
San José State University
San José, California, United States of America, 95192
E-mail: cristina.tortora@sjsu.edu
URL: https://sites.google.com/sjsu.edu/cristina/

Journal of Statistical Software https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/

December 2025, Volume 115, Issue 3 *Submitted:* 2024-03-20
doi:10.18637/jss.v115.i03 *Accepted:* 2024-11-03