MAINT.Data: Modelling and Analysing Interval Data in R

by A. Pedro Duarte Silva, Paula Brito, Peter Filzmoser and José G. Dias

Abstract We present the *CRAN* R package **MAINT.Data** for the modelling and analysis of multivariate interval data, i.e., where units are described by variables whose values are intervals of $\boldsymbol{R}$, representing intrinsic variability. Parametric inference methodologies based on probabilistic models for interval variables have been developed, where each interval is represented by its midpoint and log-range, for which multivariate Normal and Skew-Normal distributions are assumed. The intrinsic nature of the interval variables leads to special structures of the variance-covariance matrix, which are represented by four different possible configurations. **MAINT.Data** implements the proposed methodologies in the S4 object system, introducing a specific data class for representing interval data. It includes functions and methods for modelling and analysing interval data, in particular maximum likelihood estimation, statistical tests for the different configurations, (M)ANOVA and Discriminant Analysis. For the Gaussian model, Model-based Clustering, robust estimation, outlier detection and Robust Discriminant Analysis are also available.

Introduction

In classical statistics and multivariate data analysis, the basic units under analysis are single individuals, described by numerical and/or categorical variables, each individual taking one single value for each variable. For instance, a specific football player may be described by his age, height, weight, goals marked, nationality; a specific passenger by his/her gender, age, destination, weight of luggage, etc. Data are organised in a data-array, where each cell (i, j) contains the value of variable j for individual i.

It is however often the case that the data under analysis are not single observations, but rather sets of values, either related to groups of units gathered on the basis of some common properties, or observed repeatedly over time or under different specific conditions. The classical framework is then somehow restricted to take into account variability inherent to such data. This is the case when we are interested in describing football teams and not each specific player, or flights and not each particular passenger. The same issue often arises in official statistics analysis. Whether it is for the analysis' purposes, or for confidentiality reasons, individual data – here usually called "microdata" – is gathered into more general data arrays, related to parishes, counties, socio-economical groups, etc. – the so-called "macro-data". Internal variability should also be considered when the focus of the analysis lies in concepts (i.e., all elements sharing a given set of defining properties) rather than in a single specimen – whether it is a plant species (and not the specific plant I hold in my hand), a model of car (and not the particular one I am driving), etc. Another pertinent case arises when we are facing huge amounts of data, recorded in very large databases, and elements of interest are not the individual records but some second-level entities. For instance, in a database of a hypermarket purchases, we are surely more interested in describing the behaviour of some client (or some pre-defined class or group of clients) rather than each purchase by itself. The analysis requires then that the purchase data for each person (or group) be somehow aggregated to obtain the information of interest; here again the observed variability for each client or within each group is of utmost importance, and cannot be retained by summary statistics.

Symbolic Data Analysis (see e.g. Diday and Noirhomme-Fraiture (2008), Brito (2014)) provides a framework where the variability observed may effectively be considered in the data representation, and methods are developed that take that into account. To describe groups of individuals or concepts, new variable types may now assume other forms of realisations, which allow taking intrinsic variability into account. They may take the form of finite sets, intervals or distributions. In recent years, different approaches have been investigated and many methods proposed for the analysis of such symbolic data, and for the design of a symbolic counterpart of statistical multivariate data analysis methods. Most existing methods for the analysis of such data rely however on non-parametric descriptive approaches. Among these, interval data is by far the most investigated data type and for which more methods have been developed.

In Brito and Duarte Silva (2012), parametric inference methodologies based on probabilistic models for interval variables are developed where each interval is represented by its midpoint and log-range, for which multivariate Normal and Skew-Normal (Azzalini and Dalla Valle, 1996) distributions are assumed. The intrinsic nature of the interval variables leads to special structures of the variance-covariance matrix, which are represented by different possible configurations.

It should be noticed that we are modelling interval-valued variables, i.e. variables whose observed

values are intervals, and not single-valued real variables. For this reason, they should not be confused with real-valued variables whose values are restricted to some intervals. Data structures for this latter type are available in some R packages such as **survreg** (Hubeaux and Rufibach, 2015) or **crch** (Messner et al., 2019), but they obviously do not apply in our context.

In this paper, we present the package **MAINT.Data** (Duarte Silva and Brito, 2021), which implements the proposed methodologies in R (R Core Team, 2021). **MAINT.Data** is built using S4 classes and methods, introducing a specific data class for representing interval data. Functions for aggregating microdata into interval data objects are also provided. **MAINT.Data** includes functions and methods for modelling and analysing interval data, in particular maximum likelihood estimation and statistical tests for the different considered configurations. Methods for (M)ANOVA (Brito and Duarte Silva, 2012) and Discriminant Analysis (Duarte Silva and Brito, 2015) of this data class are also provided. For the Gaussian model, Model-based Clustering (Brito et al., 2015), robust estimation and outlier detection (Duarte Silva et al., 2018) are implemented; corresponding methods for Robust Discriminant Analysis are also available.

Multivariate analysis of interval-valued data has been addressed from different perspectives, as Clustering (see, e.g., De Carvalho et al. (2006); De Carvalho and Lechevallier (2009)), Principal Component Analysis (PCA) (see, e.g. Douzal-Chouakria et al. (2011); Le-Rademacher and Billard (2012)), Discriminant Analysis (Duarte Silva and Brito, 2015; Ramos-Guajardo and Grzegorzewski, 2016), Regression Analysis (Dias and Brito, 2017; Lima Neto and De Carvalho, 2008, 2010, 2011), etc. For a survey the reader may refer to Brito (2014). Those are mostly non-parametric exploratory methodologies; recent approaches based on parametric models have also been proposed in Brito and Duarte Silva (2012), Le-Rademacher and Billard (2011), and Lima Neto and De Carvalho (2011).

Many of the methods mentioned above for analysing interval-valued data may be found in R packages, namely **symbolicDA** (Dudek et al., 2019), (general multivariate data analysis/machine learning approaches, e.g. PCA, Discriminant Analysis, Multidimensional Scaling, Clustering), **RSDA** (Rodriguez, 2021) (mainly classification and linear models), **iRegression** (Lima Neto et al., 2016) (Regression) and **GPCSIV** (Brahim and Makosso-Kallyth, 2013) (PCA). We note that most of these packages implement non-parametric methods, an exception being **iRegression** which comprehends regression based on the parametric approach proposed in Lima Neto and De Carvalho (2011). To the best of our knowledge, no other implementations of parametric approaches for the (multivariate) analysis of interval-valued data are publicly available.

The remainder of the paper is organised as follows. In the next section, we introduce interval data array and fix notation. Section **Models and estimation** presents the proposed models and the estimation of corresponding parameters. Section **Multivariate analysis** develops multivariate analysis methods based on those models. Section **Package** discusses the main structure and technical implementation of the **MAINT.Data** package. In Section **Applications**, two applications illustrate the use of the package and its functionalities. Finally, Section **Summary** concludes the paper, pointing out perspectives for future developments.

Interval data

Let $S = \{s_1, \ldots, s_n\}$ be the set of n units under analysis. An interval variable is defined by an application

$$Y : S \to T \text{ such that } s_i \to Y(s_i) = [l_i, u_i]$$

where T is the set of intervals of an underlying set $O \subseteq \mathbb{R}$. Let I be an $n \times p$ matrix containing the values of p interval variables on S. Each $s_i \in S$ is then represented by a p-dimensional vector of intervals, $I_i = (I_{i1}, \ldots, I_{ip}), i = 1, \ldots, n$, with $I_{ij} = [l_{ij}, u_{ij}]$, with $u_{ij} \geq l_{ij}, j = 1, \ldots, p$ (see Table 1). The models considered in **MAINT.Data** assume all intervals are non-degenerate, i.e., $u_{ij} > l_{ij}, j = 1, \ldots, p, i = 1, \ldots, n$.

The value of an interval variable Y_j for each $s_i \in S$ is defined by the lower and upper bounds l_{ij} and u_{ij} of $I_{ij} = Y_j(s_i)$, here assumed to be strictly different (i.e. degenerate intervals are not considered in this framework). For modelling purposes, however, an alternative parametrisation that consists in representing $Y_j(s_i)$ by the MidPoint $c_{ij} = \dfrac{l_{ij} + u_{ij}}{2}$ and Log-Range $r^*_{ij} = \ln(u_{ij} - l_{ij})$ of I_{ij} is often adopted.

We note that the interval-valued data considered here do not represent uncertainty, but rather intrinsic variability. Such interval data may occur directly, or result from the aggregation of microdata. "Native" interval data are common e.g. in Botany and Zoology, one example being the length of the stem of a given plant species, which of course varies from specimen to specimen. The aggregation of

	Y_1	...	Y_j	...	Y_p
s_1	$[l_{11}, u_{11}]$	...	$[l_{1j}, u_{1j}]$	...	$[l_{1p}, u_{1p}]$
...	...		...		...
s_i	$[l_{i1}, u_{i1}]$	...	$[l_{ij}, u_{ij}]$	...	$[l_{ip}, u_{ip}]$
...	...		...		...
s_n	$[l_{n1}, u_{n1}]$	...	$[l_{nj}, u_{nj}]$	...	$[l_{np}, u_{np}]$

Table 1: Matrix I of interval data

microdata from potentially large databases also provides interval data, when individual numerical records are combined at the required level of granularity leading to a range of values representing the underlying variability. An example of such a case is the aggregation of the values of single purchases say, in the Bakery and Dairy section of a supermarket, for each client, during a year – we then obtain, for each client and for each supermarket section, an interval representing the variability of purchase values. Such aggregations are usually based on observed minima and maxima, but specific quantiles may also be considered for this purpose.

Models and estimation

Models specification

In Brito and Duarte Silva (2012), parametric models for interval data, relying on multivariate Normal or Skew-Normal distributions for the MidPoints and Log-Ranges of the interval-valued variables have been proposed.

The Gaussian model consists in assuming a joint multivariate Normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for the MidPoints C and the logs of the Ranges R^*, with $\boldsymbol{\mu} = \left[\boldsymbol{\mu}_C^t \; \boldsymbol{\mu}_{R^*}^t\right]^t$ and $\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{CC} & \boldsymbol{\Sigma}_{CR^*} \\ \boldsymbol{\Sigma}_{R^*C} & \boldsymbol{\Sigma}_{R^*R^*} \end{pmatrix}$ where $\boldsymbol{\mu}_C$ and $\boldsymbol{\mu}_{R^*}$ are p-dimensional column vectors of the mean values of, respectively, the MidPoints and Log-Ranges, and $\boldsymbol{\Sigma}_{CC}, \boldsymbol{\Sigma}_{CR^*}, \boldsymbol{\Sigma}_{R^*C}$ and $\boldsymbol{\Sigma}_{R^*R^*}$ are $p \times p$ matrices with their variances and covariances. This model has the advantage of allowing for a straightforward application of classical multivariate methods.

Given that the MidPoint C_{ij} and the Log-Range R^*_{ij} of the value of an interval variable $Y_j(s_i)$ are related to the same variable, they should, therefore, be considered together and their relation taken into account by appropriate configurations of the global covariance matrix. Intermediate parametrisations between the non-restricted and the non-correlation setup considered for real-valued data are, therefore, relevant for the specific case of interval data.

The most general formulation allows for non-zero correlations among all MidPoints and Log-Ranges (configuration 1); in another setup, interval variables Y_j are independent, but for each variable, the MidPoint may be correlated with its Log-Range (configuration 2); a third situation allows for MidPoints (respectively, Log-Ranges) of different variables to be correlated, but no correlation between MidPoints and Log-Ranges is allowed (configuration 3); finally, all MidPoints and Log-Ranges may be uncorrelated, both among themselves and between each other (configuration 4). Table 2 summarizes the different considered configurations. We note that from the normality assumption it follows that, in this particular framework, imposing non-correlations with Log-Ranges is equivalent to imposing non-correlations with Ranges.

Configuration	Characterization	$\boldsymbol{\Sigma}$
C1	Not restricted	Not restricted
C2	Y_j's not correlated	$\boldsymbol{\Sigma}_{CC}, \boldsymbol{\Sigma}_{CR^*} = \boldsymbol{\Sigma}_{R^*C}$, $\boldsymbol{\Sigma}_{R^*R^*}$ all diagonal
C3	C's not-correlated with R^*'s	$\boldsymbol{\Sigma}_{CR^*} = \boldsymbol{\Sigma}_{R^*C} = \mathbf{0}$
C4	All C's and R^*'s are not-correlated	$\boldsymbol{\Sigma}$ diagonal

Table 2: Different cases for the variance-covariance matrix.

It should be remarked that for configurations C2, C3 and C4, $\boldsymbol{\Sigma}$ can be written as a block diagonal matrix, after a possible rearrangement of rows and columns.

In Brito and Duarte Silva (2012) another configuration has been considered, where MidPoints (respectively, Log-Ranges) of different variables may be correlated, the MidPoint of each variable may be correlated with its Log-Range, but no correlation between Midpoints and Log-Ranges of different variables is allowed. However, this case seems less natural, and leads to computational difficulties, since $\mathbf{\Sigma}$ can no longer be written as a block diagonal matrix, and, therefore, it has not been used in subsequent studies.

The Gaussian model has many advantages, which explains its generalized use in multivariate data analysis; in particular, it allows for a direct modelling of the covariance structure between the variables. Nevertheless, it does present some limitations, namely the fact that it imposes a symmetrical distribution on the MidPoints and a specific relation between mean, variance and skewness for the Ranges. A more general model that overcomes these limitations may be obtained by considering the family of Skew-Normal distributions (see, for instance, Azzalini (1985); Azzalini and Dalla Valle (1996)). The Skew-Normal generalizes the Gaussian distribution by introducing an additional shape parameter, while trying to preserve some of its mathematical properties.

The density of a q-dimensional Skew-Normal distribution is given by

$$f(\boldsymbol{x};\boldsymbol{\alpha},\boldsymbol{\xi},\mathbf{\Omega}) = 2\phi_q(\mathbf{x}-\boldsymbol{\xi};\mathbf{\Omega})\Phi(\boldsymbol{\alpha}^t\boldsymbol{\omega}^{-1}(\mathbf{x}-\boldsymbol{\xi})), \mathbf{x}\in R^q \tag{1}$$

where now $\boldsymbol{\xi}$ and $\boldsymbol{\alpha}$ are q-dimensional vectors, $\mathbf{\Omega}$ is a symmetric $q\times q$ positive-definite matrix, $\boldsymbol{\omega}$ is a diagonal matrix formed by the square-roots of the diagonal elements of $\mathbf{\Omega}$, ϕ_q is the density of a $N_q(0,\mathbf{\Omega})$ and Φ is the distribution function of a standard Gaussian variable.

Notice that the Skew-Normal model encompasses mixed models with marginal Normal random variables, for which the corresponding shape parameter is null.

The mean vector, variance-covariance matrix, and skewness coefficients of a q-dimensional Skew-Normal distribution are given by (see Azzalini (2005))

$$\boldsymbol{\mu} = E(\boldsymbol{X}) = \boldsymbol{\xi} + \boldsymbol{\omega}\boldsymbol{\mu}_Z \tag{2}$$

$$\mathbf{\Sigma} = Var(\boldsymbol{X}) = \mathbf{\Omega} - \boldsymbol{\omega}\boldsymbol{\mu}_Z\boldsymbol{\mu}_Z^t\boldsymbol{\omega} \tag{3}$$

$$\gamma_{1,\ell} = \frac{E[(X_\ell - E(X_\ell))^3]}{Var(X_\ell)^{3/2}} = \frac{4-\pi}{2}\frac{\mu_{Z;\ell}^3}{(1-\mu_{Z;\ell}^2)^{3/2}}\ ,\ \ell = 1,\ldots,q \tag{4}$$

where $\boldsymbol{\mu}_Z$ is a vector of expected values for standard Skew-Normal variables, which are defined by

$$\boldsymbol{\mu}_Z = \sqrt{\frac{2}{\pi}}\ \frac{\boldsymbol{\omega}^{-1}\mathbf{\Omega}\boldsymbol{\omega}^{-1}\boldsymbol{\alpha}}{\sqrt{1+\boldsymbol{\alpha}^t\boldsymbol{\omega}^{-1}\mathbf{\Omega}\boldsymbol{\omega}^{-1}\boldsymbol{\alpha}}}$$

As an alternative to the Gaussian model, it may be considered that (C, R^*) follows jointly a $2p$-multivariate Skew-Normal distribution, for which the different alternative configurations of the $\mathbf{\Sigma}$ matrix may be assumed. Given (3), a null covariance $\mathbf{\Sigma}(j,j')$ implies that $\mathbf{\Omega}(j,j')=\mathbf{\Omega}(j,j)^{\frac{1}{2}}\boldsymbol{\mu}_{Z_j}\mathbf{\Omega}(j',j')^{\frac{1}{2}}\boldsymbol{\mu}_{Z_{j'}}$ or, equivalently, $\mathbf{\Omega}(j,j) = \frac{2}{\pi}\ \frac{1}{1+\boldsymbol{\alpha}^t\boldsymbol{\omega}^{-1}\mathbf{\Omega}\boldsymbol{\omega}^{-1}\boldsymbol{\alpha}}\ \mathbf{\Omega}_j^t\boldsymbol{\omega}^{-1}\boldsymbol{\alpha}\boldsymbol{\alpha}^t\boldsymbol{\omega}^{-1}\mathbf{\Omega}_{j'}$ where $\mathbf{\Omega}_j$ denotes the j^{th} column of $\mathbf{\Omega}$. This defines non-linear relations between the parameters in $\mathbf{\Omega}$ and $\boldsymbol{\alpha}$.

Maximum likelihood estimation

As discussed in the previous subsection, Brito and Duarte Silva (2012) consider as possible models for interval-valued data, eight possible combinations of two multivariate distributions (Gaussian or Skew-Normal) with four covariance configurations. Given an observed data set, the choice among these models may be based on their maximised likelihood using usual information criteria such as the Bayesian Information Criterion (BIC) (Schwarz, 1978), the Akaike Information Criterion (AIC) (Akaike, 1974), or pairwise likelihood ratio tests. In this subsection we will present the details of the respective maximum likelihood estimation.

Gaussian model

Let $\boldsymbol{X_i} = \left[\boldsymbol{C_i}^t, \boldsymbol{R_i^*}^t\right]^t$ be the $2p$-dimensional column vector comprising all the MidPoints and Log-Ranges for unit s_i, $\bar{\boldsymbol{X}}$ be sample mean of the $\boldsymbol{X_i}$'s and $\boldsymbol{E} = \sum_{i=1}^{n}(\boldsymbol{X_i}-\bar{\boldsymbol{X}})(\boldsymbol{X_i}-\bar{\boldsymbol{X}})^{\boldsymbol{t}}$. For all configurations, the log-likelihood can be written as

$$\ln L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = l = -np\ \ln(2\pi) - \frac{n}{2}\ \ln\det\boldsymbol{\Sigma} - \frac{1}{2}\ \mathrm{Tr}(\boldsymbol{E}\boldsymbol{\Sigma}^{-1}) - \frac{n}{2}\,(\bar{\boldsymbol{X}} - \boldsymbol{\mu})^t\,\boldsymbol{\Sigma}^{-1}\,(\bar{\boldsymbol{X}} - \boldsymbol{\mu}) \tag{5}$$

Under the unrestricted configuration C1, the maximum likelihood estimators of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are obviously the classical ones, $\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{X}}$ and $\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\boldsymbol{E}$. In the restricted configurations C2 to C4, the maximum of (5) can be obtained by separately maximising with respect to each block of $\boldsymbol{\Sigma}$, and the estimators are obtained from the non-restricted estimators simply replacing the null parameters in $\boldsymbol{\Sigma}$ by zeros (see Brito and Duarte Silva (2012)).

Skew-Normal model

Azzalini and Capitanio (see, e.g., Azzalini and Capitanio (1999); Azzalini (2005)) have obtained the log-likelihood of a q-dimensional Skew-Normal distribution as

$$\ln L(\boldsymbol{\xi}, \boldsymbol{\Omega}, \boldsymbol{\alpha}) = l = \text{ constant } - \frac{1}{2}n\ln\det\boldsymbol{\Omega} - \frac{n}{2}\,\mathrm{Tr}(\boldsymbol{\Omega}^{-1}\mathbf{V}) + \sum_i \zeta_0(\boldsymbol{\alpha}^t\boldsymbol{\omega}^{-1}(\boldsymbol{X_i} - \boldsymbol{\xi})) \tag{6}$$

where $\boldsymbol{V} = n^{-1}\sum_i(\boldsymbol{X_i} - \boldsymbol{\xi})(\boldsymbol{X_i} - \boldsymbol{\xi})^t$ and $\zeta_0(v) = \ln(2\Phi(v))$. The maximisation of (6) is performed in two steps by defining a new parameter, $\boldsymbol{\eta} = \boldsymbol{\alpha}^t\boldsymbol{\omega}^{-1}$, and separating the maximisation on $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ from the maximisation on $\boldsymbol{\Omega}$ given $\boldsymbol{\xi}$, which has the analytical solution $\boldsymbol{\Omega} = \boldsymbol{V}$.

The optimal likelihood solution for the Skew-Normal model with restricted configurations may not be obtained by simply replacing corresponding entries in the appropriate matrices, because of the non-linear relations between the parameters in $\boldsymbol{\Omega}$ and $\boldsymbol{\alpha}$. For the Skew-Normal model with restricted configurations, we rely on a centred parametrisation (Valle and Azzalini, 2008), which employs directly the parameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{\gamma_1}$ given by (2), (3) and (4), respectively. The log-likelihood is maximised with respect to $\boldsymbol{\mu}$, $\boldsymbol{\gamma_1}$ and the free elements in $\boldsymbol{\Sigma}$. This optimisation must be done numerically; see Subsection **Implementation** of Section **Package** for the details of the implementation adopted in package **MAINT.Data**.

Robust estimation and outlier detection

Multivariate datasets often include data units that deviate from the main pattern, usually called *outliers*, which may strongly influence the maximum likelihood estimators, leading to the need of alternative (robust) estimators. In the context of interval-valued data this problem has been addressed in Duarte Silva et al. (2018).

There is an extensive literature on robust estimation of location and scatter parameters. Trimmed likelihood estimators (Hadi and Luceño, 1997) are based on a sample subset, keeping only the h units that contribute most to the likelihood function. For multivariate Gaussian data, this approach is equivalent to the well-known Minimum Covariance Determinant (MCD) method (Rousseeuw, 1984, 1985) which consists in using the sample subset that minimises the determinant of the covariance matrix estimate (Hadi and Luceño, 1997). Since finding the true MCD is an NP-hard problem, when n is not small, a good approximation based on a computationally fast algorithm is usually employed (Rousseeuw and Van Driessen, 1999).

Outlier detection usually relies on Mahalanobis distances, flagging units as outliers if their distances from an appropriate estimate of the multivariate mean $\boldsymbol{m}$ is above a chosen quantile of an appropriate distribution. (Squared) Mahalanobis distances are defined as $D^2_{\boldsymbol{m},\boldsymbol{C}}(i) = (\boldsymbol{X_i} - \boldsymbol{m})^t\boldsymbol{C}^{-1}(\boldsymbol{X_i} - \boldsymbol{m})$ where $\boldsymbol{C}$ is an estimate of the covariance matrix. Traditionally, a Chi-square approximation is used for the distribution of MCD-robust squared Mahalanobis distances; however, Cerioli (2010) proposed finite sample approximations with better properties for small and even moderately large sample sizes.

Moreover, more efficient one-step re-weighted MCD estimators are often used (Hubert et al., 2008). These are obtained by giving null weight only to the units for which the raw squared robust Mahalanobis distance exceeds a high threshold value, e.g., the 97.5% quantile of the classical Chi-square or, alternatively, of the scaled-F approximation (Hardin and Rocke, 2005). Furthermore, the resulting covariance estimators are usually multiplied by consistency and bias correction factors (see Pison et al. (2002)).

In practice, one needs to specify the number h of data points to be initially used. Two common choices are to fix this number around 50%n maximising the breakdown point, or around 75%n for larger efficiency (Hubert et al., 2008). Recently, in the context of interval data outlier identification, Duarte Silva et al. (2018) proposed a two-step approach where the outlier detection procedure is first run to get an estimate of the outlier proportion and in a second step the procedure is repeated fixing the trimming parameter at the value obtained in the first step.

The trimmed Maximum Likelihood approach described above has been adapted to the problem of robust parameter estimation for the Gaussian models proposed for interval-valued data. For all considered covariance configurations, the trimmed log-likelihood can be written as

$$\ln\ TL(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{h}{2}\left(2p\ \ln(2\pi) + \ln\ \det\boldsymbol{\Sigma} + \mathrm{Tr}(\tilde{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) + (\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})^t\,\boldsymbol{\Sigma}^{-1}\,(\tilde{\boldsymbol{\mu}} - \boldsymbol{\mu})\right) \tag{7}$$

where h is the number of observations kept in the trimmed sample, and
$\tilde{\boldsymbol{\mu}} = \frac{1}{h}\sum_{i=1}^{h} \boldsymbol{X_i},\ \tilde{\boldsymbol{\Sigma}} = \frac{1}{h}\sum_{i=1}^{h}(\boldsymbol{X_i} - \tilde{\boldsymbol{\mu}})(\boldsymbol{X_i} - \tilde{\boldsymbol{\mu}})^t$ are the trimmed mean and trimmed sample covariance, respectively.

In the case of a restricted covariance matrix, the block diagonal structure always implies that trimmed likelihood maximisation is equivalent to the minimisation of the determinant of the restricted trimmed sample covariance matrix.

The one-step re-weighted bias-corrected estimators are given by

$$\hat{\boldsymbol{\mu}}_1 = \frac{\sum_{i=1}^{n} w_i \boldsymbol{X_i}}{h_1} \tag{8}$$

$$\hat{\boldsymbol{\Sigma}}_1 = \frac{l_{h_1}\ c_1 \sum_{i=1}^{n} w_i (\boldsymbol{X_i} - \hat{\boldsymbol{\mu}}_1)(\boldsymbol{X_i} - \hat{\boldsymbol{\mu}}_1)^t}{h_1} \tag{9}$$

$$h_1 = \sum_{i=1}^{n} w_i \qquad w_i = \begin{cases} 1, & \text{if} \quad l_h c\ D_{\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}}}(i) \leq \sqrt{Q_{0.975}} \\ 0, & otherwise \end{cases}$$

where $Q_{0.975}$ is the 97.5% quantile of the $D^2_{\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}}}$ distribution. In **MAINT.Data** this distribution is approximated by a Chi-square distribution with $2p$ degrees of freedom or by a (scaled) F distribution as proposed by Hardin and Rocke (2005).

In expression (9), l_h and l_{h_1} are consistency correction factors, whereas c and c_1 are finite-sample bias-correction factors - for more details see Duarte Silva et al. (2018).

These estimates may then be used for outlier detection in an interval-valued dataset. For that purpose, the robust squared Mahalanobis distance for unit i, based on $\hat{\boldsymbol{\mu}}_1$ and $\hat{\boldsymbol{\Sigma}}_1$, is compared with the chosen upper quantile of either the χ^2_{2p} distribution or using the approximations (see Cerioli (2010)):

$$D^2_{\hat{\boldsymbol{\mu}}_1,\hat{\boldsymbol{\Sigma}}_1} \sim \frac{(h_1-1)^2}{h_1} Beta\left(p, \frac{h_1 - 2p - 1}{2}\right), \quad if\ \ w_i = 1 \tag{10}$$

$$D^2_{\hat{\boldsymbol{\mu}}_1,\hat{\boldsymbol{\Sigma}}_1} \sim \frac{h_1+1}{h_1}\frac{(h_1-1)2p}{h_1-2p} F\left(2p, h_1 - 2p\right), \quad if\ \ w_i = 0 \tag{11}$$

Multivariate analysis

Analysis of Variance

The models presented above for interval-valued variables allow addressing (M)ANOVA problems with interval data - see Brito and Duarte Silva (2012). Since each interval-valued variable Y_j is modelled by $\left[C_{ij}, R^*_{ij}\right]$, an analysis of variance of Y_j is accomplished by a two-dimensional MANOVA.

Assume a one-way design, with a single factor with k levels, and let n_ℓ be the number of units in group ℓ. Let $X_{ij} = \left[C_{ij}, R^*_{ij}\right]^t$ be the 2-dimensional column vector with the MidPoint and Log-Range of variable Y_j for unit s_i, and let $\mu_{\bullet j\ell}$ be the population means of the X_j's in group ℓ. In this case, the null hypothesis states that all $\mu_{\bullet j\ell}$ are equal across groups. In all cases, for both models and all covariance configurations, we follow a likelihood ratio approach.

In the Gaussian model, the usual likelihood ratio statistic λ can be computed in a straightforward manner. Under the unrestricted case C1, this statistic is obviously equal to the classical one; in the restricted covariance cases, its value may be obtained replacing the null entries corresponding to each configuration in the sum of squares and cross-products MANOVA matrices (see e.g. Huberty and Olejnik (2006) for the definition of those matrices). For the Skew-Normal model, given there is no closed form for the maximum likelihood estimates, the value of λ must be obtained by numerical optimisation (see Subsection **Implementation** of Section **Package**).

As usual, under the null hypothesis, $-2\ln\lambda$, follows asymptotically a Chi-square distribution. For small samples a permutation test may be used to approximate the distribution of this test statistic.

A simultaneous analysis of all the Y's interval-valued variables may be accomplished by a $2p$-dimensional MANOVA, following the same procedure.

Discriminant Analysis

The classical decision theoretic approach to Discriminant Analysis assumes that a given vector of attributes follows some known distribution and derives an optimal classification rule that minimises either the misclassification probability or the expected value of the misclassification cost. Parametric discriminant analysis of interval-value data based on the models above has been investigated in Duarte Silva and Brito (2015).

In a problem with k groups, $\Gamma_\ell, \ell = 1, \ldots, k$, denote the *a priori* group membership probabilities by π_ℓ and the within group probability or density function by $f_\ell(\mathbf{x})$, where $\mathbf{x}$ are attribute vectors. Under the assumption that misclassification cost are equal across groups, the optimal rule assigns a unit to the group Γ_ℓ for which $\pi_\ell f_\ell(\mathbf{x})$ is maximal (see, e.g. McLachlan (1992)); in practice the unknown parameters in these rules must be estimated from observations with known group membership.

When $f_\ell(.)$ is a Gaussian density, and the covariance matrices are equal across groups, the approach described above leads to a linear classification rule, whereas when covariance matrices differ from group to group, a quadratic classification rule is obtained.

Consider the Gaussian model for interval data. For each covariance configuration, an estimate of the optimum classification rule can be obtained by direct generalisation of the classical linear and quadratic discriminant classification rules, leading to group assignments defined by, respectively,

$$\Gamma = \underset{\ell}{\operatorname{argmax}}(\hat{\boldsymbol{\mu}}_\ell^t \hat{\boldsymbol{\Sigma}}^{-1}\mathbf{x} - \frac{1}{2}\hat{\boldsymbol{\mu}}_\ell^t \hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}}_\ell + \ln\ \hat{\pi}_\ell) \tag{12}$$

$$\Gamma = \underset{\ell}{\operatorname{argmax}}(-\frac{1}{2}\mathbf{x}^t \hat{\boldsymbol{\Sigma}}_\ell^{-1}\mathbf{x} + \hat{\boldsymbol{\mu}}_\ell^t \hat{\boldsymbol{\Sigma}}_\ell^{-1}\mathbf{x} + \ln\ \hat{\pi}_\ell - \frac{1}{2}(\ln\det\hat{\boldsymbol{\Sigma}}_\ell + \hat{\boldsymbol{\mu}}_\ell^t \hat{\boldsymbol{\Sigma}}_\ell^{-1}\hat{\boldsymbol{\mu}}_\ell)) \tag{13}$$

where $\hat{\boldsymbol{\mu}}_\ell, \hat{\boldsymbol{\Sigma}}, \hat{\boldsymbol{\Sigma}}_\ell$ and $\hat{\pi}_\ell$ are appropriate estimates of $\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}, \boldsymbol{\Sigma}_\ell$ and π_ℓ for the corresponding cases.

In **MAINT.Data**, all mean and covariance estimates in (12) and (13) may be obtained by either classical maximum likelihood or the robust trimmed maximum likelihood approach (see Section **Robust estimation and outlier detection**).

We note that for the restricted configurations C2, C3 and C4, $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\Sigma}}_\ell$ are obtained from the corresponding unrestricted estimates replacing all the null covariances by zeros.

For the Skew-Normal case, we consider a Location Model in which the groups differ only in terms of the location parameter $\boldsymbol{\xi}$, and a General Model, where the groups differ in terms of all parameters. The corresponding classification rules are, respectively,

$$\Gamma = \underset{\ell}{\operatorname{argmax}}(\hat{\boldsymbol{\xi}}_\ell^t\, \hat{\boldsymbol{\Omega}}^{-1}\mathbf{x} - \frac{1}{2}\hat{\boldsymbol{\xi}}_\ell^t\, \hat{\boldsymbol{\Omega}}^{-1}\hat{\boldsymbol{\xi}}_\ell + \ln\ \hat{\pi}_\ell + \zeta_0(\hat{\boldsymbol{\alpha}}^t \hat{\boldsymbol{\omega}}^{-1}(\mathbf{x} - \hat{\boldsymbol{\xi}}_\ell))) \tag{14}$$

$$\Gamma = \underset{\ell}{\operatorname{argmax}}(-\frac{1}{2}\mathbf{x}^t \hat{\boldsymbol{\Omega}}_\ell^{-1}\mathbf{x} + \hat{\boldsymbol{\xi}}_\ell^t\, \hat{\boldsymbol{\Omega}}_\ell^{-1}\mathbf{x} + \ln\ \hat{\pi}_\ell - \frac{1}{2}(\ln\ \det\hat{\boldsymbol{\Omega}}_\ell + \hat{\boldsymbol{\xi}}_\ell^t\, \hat{\boldsymbol{\Omega}}_\ell^{-1}\hat{\boldsymbol{\xi}}_\ell) + \zeta_0(\hat{\boldsymbol{\alpha}}_\ell^t\, \hat{\boldsymbol{\omega}}_\ell^{-1}(\mathbf{x} - \hat{\boldsymbol{\xi}}_\ell))) \tag{15}$$

where $\hat{\boldsymbol{\xi}}_\ell, \hat{\boldsymbol{\Omega}}, \hat{\boldsymbol{\Omega}}_\ell, \hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\alpha}}_\ell$ are estimates of location, scale, association and shape parameters (see Azzalini and Capitanio (1999)), $\hat{\boldsymbol{\omega}}$ and $\hat{\boldsymbol{\omega}}_\ell$ are the square-roots of the diagonal elements of the matrices $\hat{\boldsymbol{\Omega}}$ and $\hat{\boldsymbol{\Omega}}_\ell$, respectively, and $\zeta_0(v) = \ln(2\Phi(v))$. In **MAINT.Data** these are all maximum likelihood estimates.

Model-based Clustering

Model-based Clustering considers the data as arising from a distribution that is a mixture of two or more components (Banfield and Raftery, 1993; Fraley and Raftery, 2002; McLachlan and Peel, 2000). Each component, that can be thought as a cluster, is characterised by a conditional density/mass function and has an associated probability or "weight". When the conditional probability is specified as the multivariate Gaussian density, the probability model for clustering will be a finite mixture of multivariate Normals (known as the Gaussian mixture model).

The problem consists in estimating the model parameters for each component, as well as the membership (posterior) probabilities of each unit. To this purpose, the Expectation-Maximisation (EM) algorithm (Dempster et al., 1977) is commonly used. The method alternates between an expectation (E) step, which computes the expectation of the log-likelihood at the current parameter estimates, and a maximisation (M) step, which estimates parameters maximising the expected log-likelihood found in the E step.

Model-based Clustering of interval data has been addressed in Brito et al. (2015), by considering the Gaussian parametrisation described above (see Section **Models and estimation**). For that purpose, the EM algorithm has been adapted to the likelihood maximisation in our models, for the different covariance configurations.

The finite mixture model with k components for $2p$-dimensional data vector $\boldsymbol{x}$ is defined by

$$f(\boldsymbol{x};\boldsymbol{\varphi}) = \sum_{\ell=1}^{k} \tau_\ell f_\ell(\boldsymbol{x};\boldsymbol{\theta_\ell}) \tag{16}$$

where all $\tau_\ell > 0$ and $\tau_1 + \ldots + \tau_k = 1$; $\boldsymbol{\theta_\ell}$ denotes parameters of the conditional distribution of component ℓ.

Here the conditional distribution is given by $N(\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)$; maximum likelihood parameter estimation involves the maximisation of the log-likelihood function:

$$\ln L(\boldsymbol{\varphi};\mathbf{x}) = \sum_{i=1}^{n} \ln f(\mathbf{x}_i;\boldsymbol{\varphi}) \tag{17}$$

where $\boldsymbol{\varphi} = (\tau_1, \ldots, \tau_k, \boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_k})$.

In Model-based clustering of interval data, $\boldsymbol{X_i} = \left[\boldsymbol{C_i^t}, \boldsymbol{R_i^{*t}}\right]^t$ is defined as the $2p$-dimensional vector comprising all the MidPoints and Log-Ranges for s_i, and the "complete" data are considered to be $(\boldsymbol{x_i}, \boldsymbol{z_i})$, where $\boldsymbol{z_i} = (z_{i1}, \ldots, z_{ik})$ is assumed as the "missing" data, with $z_{i\ell} = 1$ if $s_i \in$ component ℓ and $z_{i\ell} = 0$ otherwise. In the unrestricted case, the M-step formulas for $\hat{\boldsymbol{\Sigma}}$, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\ell}}$ are the classical ones; for the restricted configurations $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\ell}}, \ell = 1, \ldots, k$ are obtained maximising the likelihood for each block separately (see Brito and Duarte Silva (2012)).

For the selection of the appropriate model and the number of components k, we use the Bayesian Information Criterion (BIC).

Package

Design

MAINT.Data is built around S4 classes and methods, the most important being the IData class and classes derived from the virtual IdtE (IData Estimates) classes. Further specialised classes used to store the results of various multivariate analysis (*e.g.* Model-based Clustering, MANOVA and Discriminant Analysis) are also available. Figure 1 shows common interactions between different objects of **MAINT.Data** classes.

We note that in addition to the flow shown in Figure 1, objects containing the results of Discriminant Analysis of Interval Data may also be obtained from appropriate objects of class `IdtMANOVA`, or directly from the combination of objects of class IData with a grouping factor.

Class `Idata`, which is used to store datasets of interval-valued variables, is the central class in the **MAINT.Data** package. Its design aims at replicating the functionalities of classical data frames as smoothly as possible. As seen in Figure 1, objects of class `IData` may be created in one of two alternative ways: (i) directly from data frames containing either lower and upper bounds or MidPoints and Log-Ranges, using the creator function `IData`; (ii) by aggregation of a data frame of the microdata by a given aggregating factor and criterion (*e.g.* min-max or a given pair of quantiles), using the function `AgrMcDt`.

The creator function `IData` takes five arguments as input. The first one, named *Data* refers to a data frame or matrix containing either the lower and upper bounds or the MidPoints and Log-Ranges of the observed intervals, where each row corresponds to a different unit. Then, *Seq* is a string which describes the sequence of the data for each unit, namely, lower and upper bounds variable by variable (*"LbUb_VarbVar"*, default), MidPoints and Log-Ranges variable by variable (*"MidPLogR_VarbVar"*), all

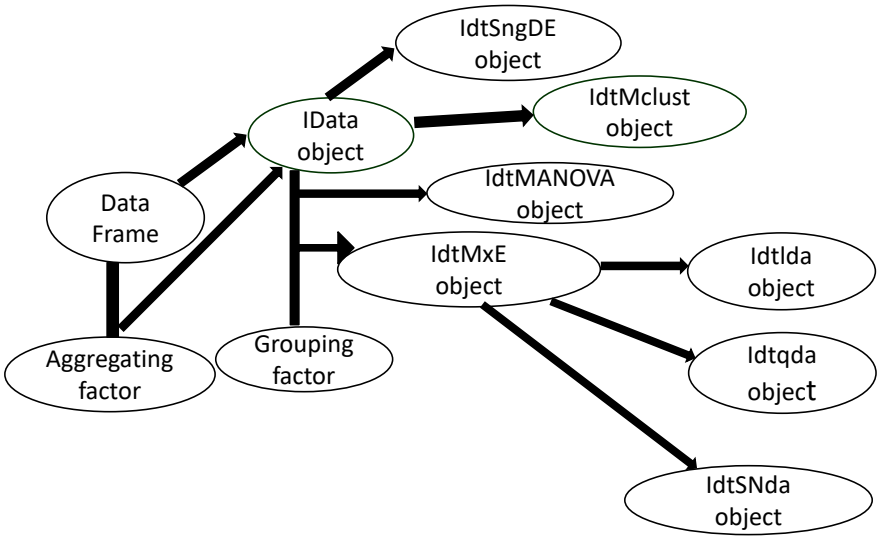


Figure 1: Typical flow of a MAINT.Data application.

lower bounds followed by all upper bounds (*"AllLb_AllUb"*), or all MidPoints followed by all LogRanges (*"AllMidP_AllLogR"*). The third and fourth arguments, named *VarNames* and *ObsNames*, allow the user to specify the variables' and units' names, respectively. Finally, the last argument *NbMicroUnits* provides the number of micro observations corresponding to each unit, when available. A typical call of this function would be `ExampleIdt <-IData(dataDF,VarNames=c(``Var1'',``Var2''))` (no names for the units, number of micro observations corresponding to each unit not available).

Function `AgrMcDt` has three arguments. The first one, *MicDtDF* indicates a data frame with the microdata. The second argument, *agrby* refers to a factor with the categories according to which the microdata should be aggregated. The last argument *agrcrt* specifies whether aggregation is done with the minimum and maximum observed values, or else based on user-defined quantiles. An example is shown in Section **Applications**.

A UML diagram of class `Idata` is shown in Figure 2. As seen here, class `Idata` implements specialised versions of standard R methods such as `summary`, `print`, `nrow` and `ncol`, `rownames` and `colnames`, `rbind`, `cbind` and `plot`. Special care has been taken in the development of indexing operators and of a specialised `cbind` method, so that they work as smoothly as with data frames, but treating each column of `Idata` as one interval-valued variable.

The remaining `Idata` methods perform parameter estimation and/or multivariate analysis leading to objects of class `IdtE` (parameter estimation), `IdtMANOVA` (Multivariate Analysis of Variance), `Idtda` (Discriminant Analysis), or `IdtMclust` (Model-based Clustering). All these methods include a `Covcase` argument used to specify the covariance configurations assumed, which by default compares the BIC of the results for all four configurations, and select the one with the lowest BIC value.

The `IdtE` class is an abstract (virtual) class used to store parameter estimates of the models assumed for interval-valued variables. As shown in Figure 3 there are currently eight such specialisations, depending on the model assumed and type of estimation performed. The names of these classes always start with the letters *Idt* followed by *Sng* or *Mx* (estimates of parameters of a single or several distributions), *ND*, *SND* or *NandSND* (Gaussian, SkewNormal or both Gaussian and SkewNormal distributions), and end with *E* or *RE* (Maximum Likelihood or Robust estimates).

As shown in Figure 4 the same reasoning applies to classes derived from the virtual class `IdtMANOVA`. However, in this case, only Maximum Likelihood estimation has been considered and the specialisations distinguish classical MANOVA (class `IdtClMANOVA`), heterocedastic MANOVA based on Gaussian

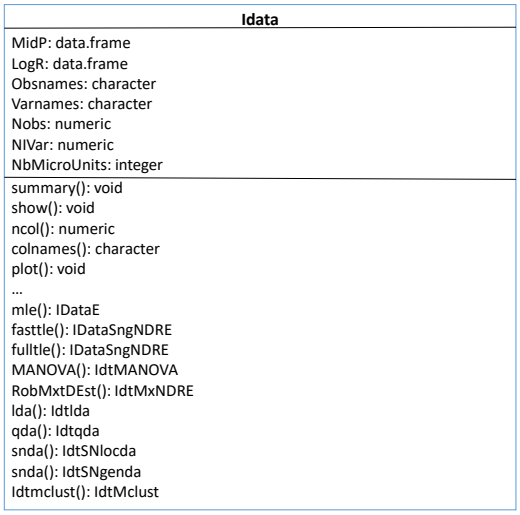


Figure 2: IData class.

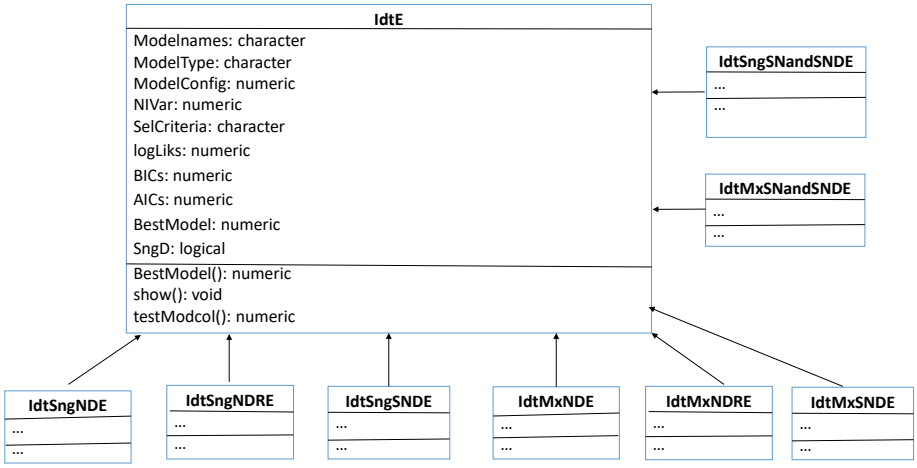


Figure 3: IdtE classes.

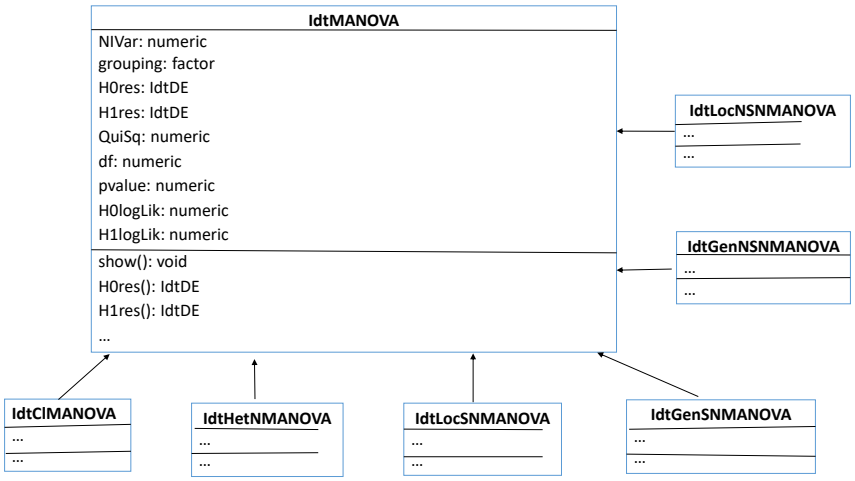


Figure 4: IdtMANOVA classes.

distributions (`IdtHetNMANOVA`), Skew-Normal based MANOVA assuming that groups may differ only in location (`IdtLocNMANOVA`) or on all parameters (`IdtGenNMANOVA`), and analyses that consider both Gaussian and SkewNormal assumptions (`IdtLocNSNMANOVA` and `IdtGenNSNMANOVA`).

Maximum likelihood estimation is performed by the `mle` method, which has six arguments. The first one, *Idt* refers to an IData object representing interval-valued units. The second argument, *Model* indicates the joint distribution assumed for the MidPoint and LogRanges; alternatives are "Normal" for Gaussian (default), "SKNormal" for Skew-Normal and "NrmandSKN" for both Gaussian and Skew-Normal distributions. The next argument, *CovCase* indicates the configurations of the variance-covariance matrix to be used (default: 1:4). The fourth argument, *SelCrit* indicates the model selection criterion, BIC (default) or AIC. The argument *kmax* specifies a tolerance criterion to identify singular correlation matrices. Finally, *OptCntrl* provides a list of optional control parameters to be passed to the optimization routine.

Robust estimation is usually performed by the `fasttle` method. Note that for small datasets, the `fulltle` method may be used, whose arguments are common to `fasttle`. The first three arguments of `fasttle` are the same as for the `mle` method. Arguments *alpha* and *getalpha* specify how the trimming proportion is chosen. Other important arguments are the following: *use.correction* indicates whether to use finite sample correction factors, default is TRUE. *rawMD2Dist* provides the assumed reference distribution of the raw MCD squared distances used to find the cutoffs defining the observations kept in one-step reweighted MCD estimates; alternatives are "ChiSq" for the usual Chi-square (default), "HardRockeAsF" and "HardRockeAdjF", respectively asymptotic and adjusted scaled F distributions proposed by Hardin and Rocke (2005). *MD2Dist* - assumed reference distribution used to find cutoffs defining the observations assumed as outliers; alternatives are "ChiS" and "CerioliBetaF", respectivelly for the usual Chi-square, and the Beta and F distributions proposed by Cerioli (2010). *reweighted* indicates whether a (Re)weighted estimate of the covariance matrix should be used in the computation of the trimmed likelihood or just a "raw" covariance estimate; default is TRUE. Argument *outlin* specifies the type of outliers to be considered, alternatives are "MidPandLogR" if outliers may be present in both MidPoints and LogRanges, "MidP" if outliers are only present in MidPoints, or "LogR" if outliers are only present in LogRanges.

Method `MANOVA` applies multivariate analysis of variance. The arguments *Idt*, *Model*, *CovCase*, *SelCrit*, *k2max* and *OptCritl* are identical to the corresponding ones of method `mle`. Argument *grouping* indicates the factor whose levels are the different groups. *MxT* indicates the type of mixing distributions to be considered: "Hom" (homoscedastic) or "Het" (heteroscedastic) for Gaussian models, "Loc" (location model) or "Gen" (general model) for Skew-Normal models (see Section **Discriminant Analysis** above). *CVtol* provides a tolerance value to identify almost constant variables within groups.

To perform discriminant analysis, three methods may be applied, namely, `lda` (linear discriminant analysis), `qda` (quadratic discriminant analysis) and `snda` (skew-normal based discriminant analysis). In all these methods, the first argument *x* denotes an IData object representing the interval-valued units, or else an object of class `IdtMANOVA`. Arguments *grouping*, *CVtol*, *CovCase*, *SelCrit* and *k2max* are identical to the corresponding ones of method `MANOVA`. The argument *prior* is used to specify the prior

probabilities of group membership, by default they are fixed at the training set corresponding proportions. In method `snda`, argument *MxT* indicates the type of mixing distributions to be considered: "Loc" (location model, default) or "Gen" (general model).

Method `Idtmclust` performs model-based clustering based on finite mixtures of Gaussian distributions. Arguments *Idt*, *CovCase*, and *SelCrit* are identical to the corresponding ones in the previous methods. The argument *G* provides the number of clusters (segments) of the mixture, by default it is set as 1:9. *MxT* indicates the type of mixing distributions to be considered, "Hom" (homoscedastic, default), "Het" (heteroscedastic), or "HomandHet" (both). Finally, the argument *control* provides a list of control parameters for the EM algorithm.

Implementation

The implementation of the `Idata` class, as well as maximum likelihood estimation and multivariate methods based on the Gaussian distribution, is relatively straightforward. As shown in Figure 2, the internal structure of the `Idata` class consists of two data frames, containing MidPoints and Log-Ranges, respectively, a couple of auxiliary constants and vector strings, and the integer vector `NbMicroUnits` which stores the number of microunits aggregated to form each interval-valued unit, when known. Therefore, `Idata` objects require roughly twice the memory space used by traditional data frames. The `Idata` slots may be retrieved by the accessor methods `MidPoints`, `LogRanges`, `rownames`, `colnames`, `nrow`, and `ncol`.

The structure of the classes derived from the virtual `IdtE` class (see Figure 3) depends on the type of model specified and estimation performed. In addition to the common slots of the `IdtE` class, these classes include vector and/or matrix slots with estimates that are constant across all covariance configurations, and a list slot named *ConvConfCases* in which each component contains estimates obtained under the assumption of a particular configuration. We note that, although the estimates corresponding to one single configuration are displayed and used in further analysis, all estimates resulting from the configurations specified by the argument *CovCase* are stored, and available to the user. The same logic applies to analyses that consider more than one model, with the results for all models being stored, but only one displayed by `summary` and `print` methods.

The maximum likelihood estimation and multivariate analysis based on the Gaussian distribution do not entail any particular difficulties, usually involving well known formulae and the replacement of some values by zero according to the covariance configuration assumed. Covariance matrices of Gaussian estimators are also computed in a straightforward manner and passed, if so requested, to the appropriate `stdEr` and `vcov` methods.

Maximum likelihood estimation of Skew-Normal parameters requires the numerical optimisation of the non-convex function (6). As this function often has many different local optima, **MAINT.Data** adopts a repeated local search strategy, calling a given local optimiser from different starting points. This is implemented in the auxiliary function `RepLOptim` that works as described below.

First, a local optimiser is called from an initial starting point leading to a local optimum. Then, this optimum is modified by a random perturbation, and the modified optimum is used as the starting point of a new call to the local optimiser. This process is repeated until several (default: 50) consecutive calls to the optimiser fail to improve the current best solution, or a limit (default: 250) on the total number of local optima, is reached. This limit, the maximum number of non-improving consecutive local optimisations, and several other control options, are set by default to reasonable values, but can be modified by the components of a list supplied as the value of the argument *control*. The same applies to methods (such as `mle` or `MANOVA` or `snda`) that internally call `RepLOptim`, using in this case a list supplied to their *Optcontrol* argument.

The default local optimiser of `RepLOptim` is the `nlminb` PORT function (Gay, 1990). However, in the case of maximum likelihood estimation of Skew-Normal parameters with unrestricted covariance configuration (C1), **MAINT.Data** overrides this default with the `msn.mle` function of Azzalini's **sn** package (Azzalini, 2021). For the remaining configurations, the local optimisation relies on a quasi-Newton optimiser (by default `nlminb`) using the analytical gradient of the centred Skew-Normal parametrisation derived by Valle and Azzalini (2008). In order to improve computational efficiency, the computation of the log-likelihood (6) and of its gradient was coded in C++, taking advantage of the numerical functions and classes provided in the **Rcpp** (Eddelbuettel and François, 2011) and **RcppArmadillo** (François et al., 2021) packages. We note that global optimality cannot be ensured and even with this strategy sometimes **MAINT.Data** identifies different local optima in different runs.

Once the optimisation of the log-likelihood (6) is completed, **MAINT.Data** approximates the covariance of the estimators using the evaluation of the expected Fisher information matrix implemented in the **sn** package. This approximation may fail if either the expected information matrix is ill-conditioned or the parameter estimates fall on the frontier of their domain. In such cases, posterior calls to the `stdEr` or `vcov` methods will result in appropriate warning messages.

The robust estimation of Gaussian model parameters by the trimmed maximum likelihood principle is implemented in the `fulltle` and `fasttle` methods. Method `fulltle` makes a full combinatorial search for the Trimmed Maximum Likelihood estimates, and should only be used when the number of units is relatively small (say, not much larger than 40). Method `fasttle` adapts the fast algorithm of Rousseeuw and Driessen (Rousseeuw and Van Driessen, 1999). Both methods were coded in C++, using functions and classes from **Rcpp** and **RcppArmadillo**. Furthermore, the methods `RobMxtDESt`, `Roblda` and `Robqda` call `fasttle` or `fulltle` in order to get robust estimates in different groups that may be used for robust discriminant analysis.

The interface of the **MAINT.Data** robust methods and classes is partially based on the framework developed in the popular **rrcov** package (Todorov and Filzmoser, 2009). In particular, the control options for the estimation algorithm used in the `fasttle`, `RobMxtDESt`, `Roblda` and `Robqda` methods can be provided by an argument of class `RobEstControl` which inherits and extends the class `CovControl` of the package **rrcov**. This way, algorithmic options may be specified in a uniform and familiar manner. The additional slots of class `RobEstControl` specify new options, such as indicators of the distributions assumed for the robust Mahalanobis distances, the nature of the outliers (only in MidPoints, only in Log-Ranges or (default) both in MidPoints and Log-Ranges), whether a two-step procedure should be used to find trimming parameters, and other choices that are available in **MAINT.Data** but not in **rrcov**.

The MANOVA methods available in **MAINT.Data** are always based on the maximum likelihood principle. By default, the Chi-square distribution is used for the test statistic. However, for small samples, a permutation test has been implemented in the auxiliary function `MANOVAPermTest` (see Seber (2009)).

The design and interface of class `IdtMclust` is modelled after class `Mclust` of the **mclust** package (Scrucca et al., 2016). As a result, the `IdtMclust` `print` and `summary` methods with their default argument values, display only a very general description of the clustering results. A characterization of the obtained clusters, and the partition itself, may be inspected by changing the `summary` arguments *parameters*, and *classification* from *FALSE* to *TRUE*. A difference between the `mclust` and `IdtMclust` classes lies in that in the former case detailed clustering results can only be retrieved directly from the `Mclust` slots while `IdtMclust` provides accessor methods such as `parameters`, `pro`, `mean`, `var`, `cor` and `classification` to retrieve these results. The EM algorithm used in `IdtMclust` is implemented in C++, using facilities of the **Rcpp** and **RcppArmadillo** packages.

Application I: flights dataset

To illustrate the modelling and methods presented above, we use the flights dataset from the R data package **nycflights13**, available at *CRAN*, which contains on-time data for all flights that departed New York city in 2013. The original microdata consists of 336776 flights characterized by nineteen variables.

From this data, we created a `data frame` named *FlightsDF*, after removing all rows with missing data, and with six columns corresponding to the following descriptive variables at microdata level: Month, Carrier (16 different carriers), Departure delay (min), Arrival delay (min), Air time(min), and Distance (miles).

We consider as units of interest classes formed by crossing Month with Carrier, leading to 185 units (note that not all the 192 possible combinations are present in the microdata). Therefore, we created a `factor`, named *FlightsUnits*, defining the class each individual case belongs to.

The command

```
R> FlightsIdt <- AgrMcDt(FlightsDF,FlightsUnits)
```

creates an interval data object *FlightsIdt*, where the values of the numerical variables Departure delay (DD), Arrival delay (AD), Air time (AT) and Distance (DT) are aggregated in the form of intervals for each unit. Leaving the aggregation argument `agrcrt` at its "minmax" default, the lower and upper bounds of the obtained intervals are the minimum and maximum values observed in the microdata, respectively.

However, we prefer to use the robust aggregation alternative, by filtering out the 5% lowest and highest values for each variable; in this case the aggregation argument specifies the chosen pair of quantiles:

```
R> FlightsIdt <- AgrMcDt(FlightsDF,FlightsUnits,agrcrt=c(0.05,0.95))
```

We note that the 43 units for which, for any variable, the lower and the upper bound are equal (degenerate interval) are eliminated, so that the final interval dataset has 142 units.

Table 3 shows a few rows of the resulting interval data table. The full interval dataset is available on **MAINT.Data**.

	Departure delay	Arrival delay	Air time	Distance
Jan-9E	$[-10, 120]$	$[-32, 116]$	$[31, 176]$	$[94, 1029]$
Jan-AA	$[-9, 65]$	$[-31, 59]$	$[115, 354]$	$[733, 2475]$
...	...	...	...	...
Dec-YV	$[-10.9, 68.2]$	$[-33.5, 66]$	$[39.4, 102.8]$	$[96, 544]$

Table 3: Flights interval data - partial view.

Figure 5 illustrates the two alternative outputs - (a)-crosses, (b)-rectangles - of the method `plot`, resulting respectively from the commands

```
R> plot(FlightsIdt[,"distance"],FlightsIdt[,"arr_delay"],
        cex.main=3, cex.lab=1.9, cex.axis=2)
R> plot(FlightsIdt[,"distance"],FlightsIdt[,"arr_delay"],type="rectangles",
        cex.main=3, cex.lab=1.9, cex.axis=2)
```

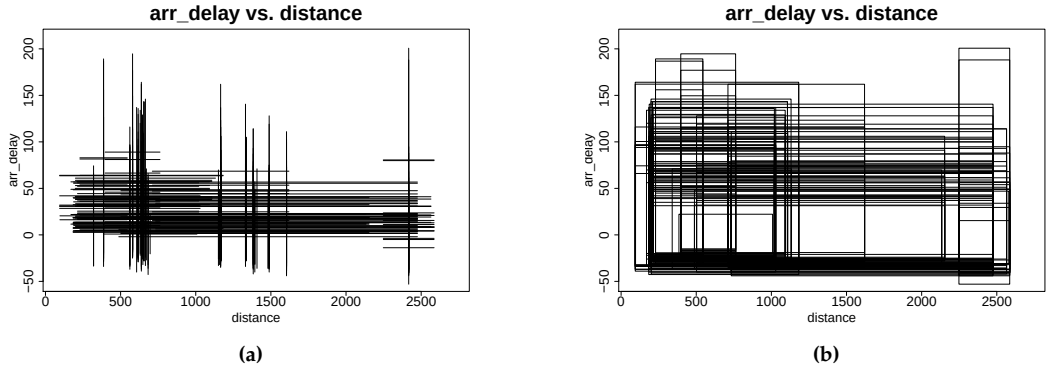
showing the intervals corresponding to the 142 units in two different forms for variables Distance and Arrival delay.

We note that the graphical arguments of traditional R plots are also available in **MAINT.Data** `plot` methods. In this example, the default graphical settings were adequate for online display, but resulted in too small axis and legends, when the resulting graphs were exported to an external text file. Therefore, we used the `cex.main`, `cex.lab`, and `cex.axis` traditional R plot arguments, to improve their readability. This particular example worked well on a PC under Linux, but since graphical characteristics are machine and operation system dependent, other argument values may be required in different computer environments.

Figure 6 plots the MidPoints versus the Log-Ranges for variable Arrival delay, resulting from the command

```
R> plot(MidPoints(FlightsIdt)[,"arr_delay.MidP"],
        LogRanges(FlightsIdt)[,"arr_delay.LogR"],
        xlab="Mid Points",ylab="Log Ranges",
        main="Mid Points vs. Log Ranges for Arrival delays",
        cex.main=2, cex.lab=1.5, cex.axis=1.5)
```

We observe a strong positive correlation between the MidPoints and the Log-Ranges of the Arrival delay, which is not uncommon for interval-valued variables.



(a) (b)

Figure 5: Interval representation of the 142 flights units – Distance versus Arrival delay.

Modelling of flights interval data

The following statistical analyses of the data will be directed towards the methods proposed earlier. Accordingly, a first analysis relates to statistically characterize the input variables and possible rela-

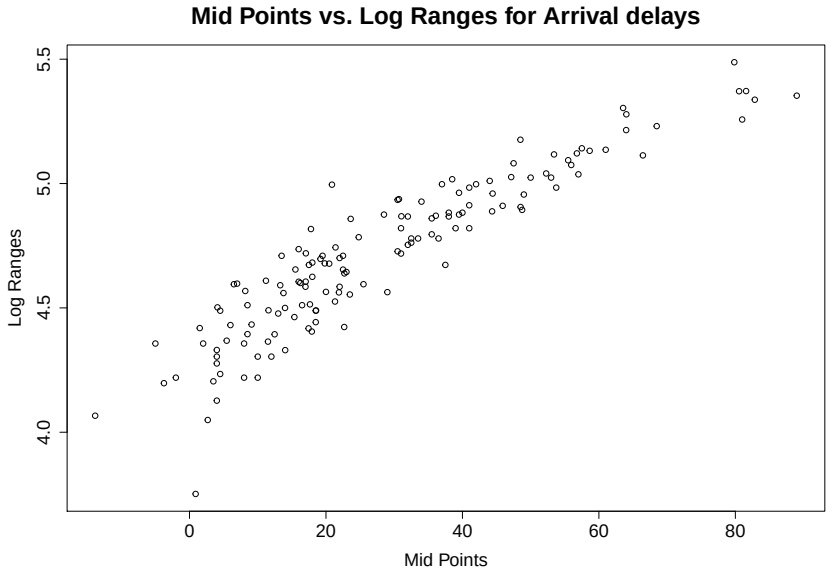


Figure 6: Midpoints VS Log-Ranges of variable Arrival delay for the 142 flights units.

tionships between them. Then the interest is in possible outliers in the interval data. Are there specific carrier/month combinations which are atypical for the observed features? The identified outliers will be excluded from the data for the subsequent analyses, which could be affected by data outliers. The data are split into two groups, the mainline carriers and the regional carriers. Do these groups differ for the considered variables (MANOVA)? Is it possible to distinguish the observations of the two groups from each other (discriminant analysis)? Are there even more subgroups in the multivariate data, and how can those be characterized (cluster analysis)?

We start by adjusting the Gaussian and Skew-Normal models for all four considered covariance configurations using the commands

```
R> Flightsmle <-mle(FlightsIdt,Model="NrmandSKN")
R> summary(Flightsmle)
```

which produce the output

```
Log likelihoods:
NModCovC1  NModCovC2  NModCovC3  NModCovC4 SNModCovC1 SNModCovC2
-2278.626  -3249.162  -2547.369  -3564.026  -2207.444  -3183.179
SNModCovC3 SNModCovC4
-2491.175  -3498.258
Bayesian (Schwartz) Information Criteria:
NModCovC1  NModCovC2  NModCovC3  NModCovC4 SNModCovC1 SNModCovC2
4775.309   6597.440   5233.501   7207.346   4672.591   6505.121
SNModCovC3 SNModCovC4
5160.760   7115.455
Selected model:
[1] "SNModCovC1"
```

We recall that for the Skew-Normal model only local optima are identified, so that in different runs slightly different solutions may be obtained.

Among the eight models $\times$ configurations, the BIC recommends the Skew-Normal model with covariance configuration C1. The likelihood ratio tests between pairs of models may be performed by command `testMod(Flightsmle)`. In this case, for any reasonable significance level, these tests suggest also the Skew-Normal model with covariance configuration C1.

The estimates of mean, standard deviation and skewness coefficient vectors and the variance-covariance matrix may be extracted by the usual `coef()` method. Alternatively, the standard methods `mean()`, `sd()`, `var()` and `cor()` may also be used. Furthermore, standard errors and variances and covariances of the estimates may be obtained, as usual, by the methods `stdEr()` and `vcov()`.

The estimates for mean values, standard deviations and skewness coefficients are:

		C				R^*			
		DD	AD	AT	DT	DD	AD	AT	DT
$\hat{\mu}^t =$	(	38.14	27.98	179.70	1244.63	4.45	4.72	4.89	6.84)
$\hat{\sigma}^t =$	(	19.30	20.68	61.89	476.72	0.42	0.32	0.59	0.66)
$\hat{\gamma}^t =$	(	0.000	−0.001	0.271	0.247	0.000	0.000	−0.046	−0.064)

The estimate of the correlation matrix is :

$$\hat{\mathbf{R}} = \left(\begin{array}{cc|cccc|cccc} & & C & & & & R^* & & & \\ & & DD & AD & AT & DT & DD & AD & AT & DT \\ \hline C & DD & 1.00 & 0.96 & -0.32 & -0.30 & 0.96 & 0.96 & -0.32 & -0.31 \\ & AD & & 1.00 & -0.40 & -0.39 & 0.92 & 0.92 & -0.33 & -0.33 \\ & AT & & & 1.00 & 0.99 & -0.37 & -0.27 & 0.47 & 0.43 \\ & DT & & & & 1.00 & -0.35 & -0.24 & 0.49 & 0.45 \\ \hline R^* & DD & & & & & 1.00 & 0.97 & -0.31 & -0.29 \\ & AD & & & & & & 1.00 & -0.25 & -0.23 \\ & AT & & & & & & & 1.00 & 0.99 \\ & DT & & & & & & & & 1.00 \end{array}\right)$$

We observe that MidPoints are positively correlated with the corresponding Log-Ranges, with strong correlations for the delay variables and moderate correlations for Distance and Air Time. The MidPoints of Departure delay and Arrival delay on the one hand, and Air time and Distance, on the other hand, have, as expected, strong correlations; the corresponding Log-Ranges also present high correlations. The observed correlation values explain the choice of the unrestricted covariance configuration C1.

Robust estimation results are obtained by the commands:

```
R> Flightstle  <-fasttle(FlightsIdt)
R> summary(Flightstle)
```

which produce the output

```
Log likelihoods:
NModCovC1 NModCovC2 NModCovC3 NModCovC4
-1416.930 -2069.002 -1720.921 -2490.111
Bayesian (Schwartz) Information Criteria:
NModCovC1 NModCovC2 NModCovC3 NModCovC4
3040.279  4231.831  3573.200  5055.283
Selected model:
[1] "NModCovC1"
```

The estimates for mean values and standard deviations are now:

		C				R^*			
		DD	AD	AT	DT	DD	AD	AT	DT
$\hat{\mu}^t =$	(	35.99	26.51	159.09	1096.43	4.454	4.70	5.22	7.23)
$\hat{\sigma}^t =$	(	18.02	19.65	58.82	459.60	0.42	0.32	0.56	0.59)

To identify outliers, we employ the default options for the robust methods and functions implemented in **MAINT.Data**, namely a cut-off based on the Chi-square distribution, and a trimming parameter based on a two step procedure using 75% of the sample in the first step.

The following command allows obtaining the list of units identified as outliers:

```
R> Flights_Otl <- getIdtOutl(FlightsIdt,Flightstle)
```

```
R> print(Flights_Otl)
```

which returns

```
Jan-FL  Jan-VX  Feb-FL  Feb-VX  Mar-FL  Mar-VX  Apr-FL  Apr-VX  Apr-YV
6       10      17      21      28      32      39      43      45
May-FL  May-VX  Jun-9E  Jun-FL  Jun-VX  Jun-WN  Jun-YV  Jul-9E  Jul-FL
51      55      58      63      67      68      69      70      75
Jul-VX  Aug-FL  Aug-VX  Aug-YV  Sep-VX  Sep-YV  Oct-FL  Oct-VX  Nov-FL
79      87      91      93      103     105     111     115     123
Nov-OO  Nov-VX  Nov-YV  Dec-FL  Dec-VX
125     128     130     136     140
```

From this list it is visible that FL (AirTran Airways) is an outlier for almost all months. When inspecting the aggregated data, it can be seen that the upper bound of the Distance is clearly lower than for the other airlines, and consequently also the upper bound of Air time. The contrary happens for airline VX (Virgin America), which is an outlier for all months. Note, however, that outlyingness can also be caused by a different multivariate behaviour of an observation.

Figure 7 shows the values of robust Mahalanobis distances (to the mean) for all 142 units, the horizontal line indicates the 97.5% quantile of the respective Chi-square distribution; it is obtained by the command

```
R> plot(Flights_Otl, cex.main=2, cex.lab=1.5, cex.axis=0.3)
```

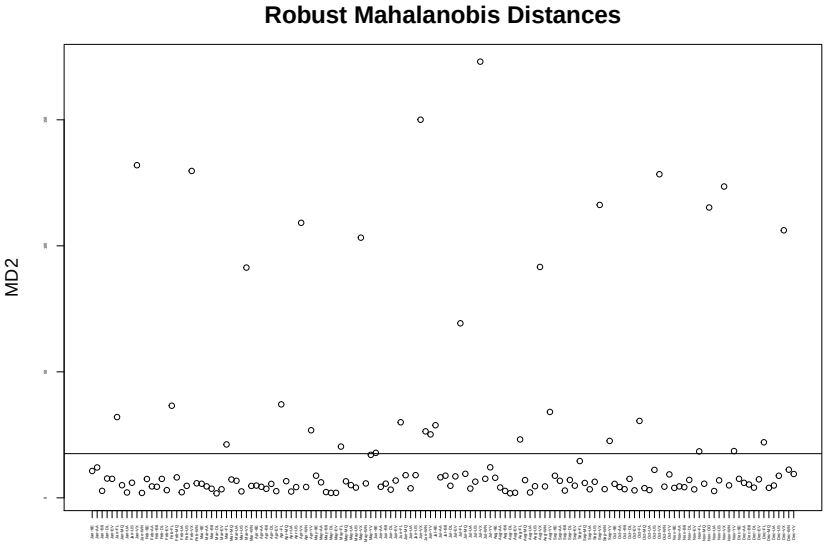


Figure 7: Values of robust Mahalanobis distances (to the mean) for the 142 flights units.

MANOVA

We now consider a partition of the 110 regular (i.e. not flagged as outliers) flights units into two groups according to whether the airline is a mainline or a regional one; the 5 regional airlines are Endeavor Air Inc. (9E), ExpressJet Airlines Inc. (EV), Envoy Air (MQ), SkyWest Airlines Inc. (OO), and Mesa Airlines Inc. (YV).

MANOVA analysis was performed for all flights units, considering this two group decomposition. We compared both a Gaussian and a Skew-Normal model, with all four covariance configurations (default), with a homoscedastic setup (default), and using the BIC (default) as comparison criterion. For that purpose we used the commands

```
R> out1<-Flights_Otl@outliers
R> carr <- substring(rownames(FlightsIdt[-out1,]),5,6)
R> carr_class <- factor(ifelse(carr=="9E"|carr=="EV"|carr=="MQ"|
 carr=="OO"|carr=="YV","REG","MAIN"))
```

```
R> MANOVAres <- MANOVA(FlightsIdt[-out1,],carr_class,Model="NrmandSKN")
R> summary(MANOVAres)
```

leading to the output

```
Null Model Log likelihoods:
NModCovC1  NModCovC2  NModCovC3  NModCovC4 SNModCovC1 SNModCovC2 SNModCovC3 SNModCovC4
-1437.000  -2094.997  -1743.016  -2521.166  -1397.125  -2060.841  -1735.816  -2444.528
Full Model Log likelihoods:
NModCovC1  NModCovC2  NModCovC3  NModCovC4 SNModCovC1 SNModCovC2 SNModCovC3 SNModCovC4
-1318.227  -1874.335  -1566.672  -2184.943  -1259.869  -1821.942  -1536.026  -2085.836
Full Model Bayesian (Schwartz) Information Criteria:
NModCovC1  NModCovC2  NModCovC3  NModCovC4 SNModCovC1 SNModCovC2 SNModCovC3 SNModCovC4
2880.879   3880.282   3302.561   4482.697   2801.767   3813.102   3278.874   4322.088
Selected Model:
[1] "SNModCovC1"

Chi-squared statistic: 274.5117
degrees of freedom: 8
p-value: 1.082712e-54
```

The Skew-Normal model with variance-covariance configuration C1 is selected (on the basis of BIC values) as the best model in this case, results indicate that the two carrier groups are indeed different for the considered variables.

These results were to be expected given that regional carriers tend to fly short distances and therefore with shorter air times, than mainlines.

We then proceeded to investigate whether the two carrier groups are different when it comes to each of the delay variables, using the commands

```
R> MANOVA_Dep_delay_res <-  MANOVA(FlightsIdt[-out1,"dep_delay"],carr_class,Model="NrmandSKN")
R> summary(MANOVA_Dep_delay_res)
R> MANOVA_Arr_delay_res <-  MANOVA(FlightsIdt[-out1,"arr_delay"],carr_class,Model="NrmandSKN")
R> summary(MANOVA_Arr_delay_res)
```

that produced the outputs

```
Null Model Log likelihoods:
NModCovC1  NModCovC2  NModCovC3  NModCovC4 SNModCovC1 SNModCovC2 SNModCovC3 SNModCovC4
-313.5688         NA         NA  -492.4602  -287.8342         NA         NA  -473.1451
Full Model Log likelihoods:
NModCovC1  NModCovC2  NModCovC3  NModCovC4 SNModCovC1 SNModCovC2 SNModCovC3 SNModCovC4
-296.6978         NA         NA  -469.0213  -272.3174         NA         NA  -449.6624
Full Model Bayesian (Schwartz) Information Criteria:
NModCovC1  NModCovC2  NModCovC3  NModCovC4 SNModCovC1 SNModCovC2 SNModCovC3 SNModCovC4
626.2990         NA         NA   966.2455   586.9391         NA         NA   936.9286
Selected Model:
[1] "SNModCovC1"

Chi-squared statistic: 31.03359
degrees of freedom: 2
p-value: 1.824489e-07

Null Model Log likelihoods:
NModCovC1  NModCovC2  NModCovC3  NModCovC4 SNModCovC1 SNModCovC2 SNModCovC3 SNModCovC4
-366.8737         NA         NA  -473.4313  -360.3910         NA         NA  -457.5188
Full Model Log likelihoods:
NModCovC1  NModCovC2  NModCovC3  NModCovC4 SNModCovC1 SNModCovC2 SNModCovC3 SNModCovC4
-357.0406         NA         NA  -456.5808  -351.1153         NA         NA  -441.4999
Full Model Bayesian (Schwartz) Information Criteria:
NModCovC1  NModCovC2  NModCovC3  NModCovC4 SNModCovC1 SNModCovC2 SNModCovC3 SNModCovC4
746.9845         NA         NA   941.3644   744.5350         NA         NA   920.6036
Selected Model:
[1] "SNModCovC1"

Chi-squared statistic: 18.55139
degrees of freedom: 2
p-value: 9.367357e-05
```

The results show that the two carrier groups present different patterns for both departure and arrival delays.

Discriminant Analysis

We consider again the 110 regular flights units grouped in two carrier classes, as in Section **Analysis of Variance**.

The function `DACrossVal` estimates error rates by *c-fold* cross-validation or by leave-one-out. Its main arguments are: (i) the data object, (ii) the grouping factor, (iii) a function with the training algorithm (e.g. lda, qda, snda, see Subsection **Design** of Section **Package**), (iv) the number of cross-validation folds (default: 10), (v) the number of replications (default: 20), (vi) a boolean flag indicating whether the folds should be stratified according to the original class proportions (default), or randomly generated from the whole training sample, ignoring class membership, and (vii) a boolean flag (false by default) stating if the leave-one-out method should be used instead of *c-fold* cross-validation.

Different discriminant methods were compared by leave-one-out cross-validation: Linear and Quadratic Discriminant Analysis for the Gaussian model, and both the Location and the General models of Skew-Normal discriminant analysis. In each case the variance-covariance configuration (see Table 2) was chosen by minimising the value of BIC. The global errors, which are also provided, are computed as a weighted average of the estimated class specific errors.

The code below computes and displays these estimates.

```
R>  DACrossVal(FlightsIdt[-out1,],carr_class,TrainAlg=lda,loo=TRUE)
R>  DACrossVal(FlightsIdt[-out1,],carr_class,TrainAlg=qda,loo=TRUE)
R>  DACrossVal(FlightsIdt[-out1,],carr_class,TrainAlg=snda,loo=TRUE)
R>  DACrossVal(FlightsIdt[-out1,],carr_class,TrainAlg=snda,Mxt="Gen",loo=TRUE)
```

We note that while the first two commands are executed quite fast (a few seconds), the last two (for the Skew-Normal model) typically need several hours, given that a computationally heavy Skew-Normal estimation is repeated many times.

These commands lead to the output below. Note that when `snda` is used without any additional arguments, the default location model is assumed.

```
Error rate estimates of algorithm lda
MAIN         REG      Global
0.013888889 0.000000000 0.009090909


Error rate estimates of algorithm qda
MAIN        REG     Global
0.00000000 0.05263158 0.01818182


Error rate estimates of algorithm snda
MAIN         REG      Global
0.08333333 0.07894737 0.08181818

Error rate estimates of algorithm snda with argument Mxt=Gen
MAIN         REG      Global
0.05555556 0.18421053 0.10000000
```

These results suggest that `lda` performs better than the other alternatives in the problem at hand. The predicted classes using the `lda` method may be obtained, as usual, by the commands

```
R> ldares <- lda(FlightsIdt[-out1,],carr_class)
R> ldapred <- predict(ldares,FlightsIdt[-out1,])
R> print(ldapred$class)
```

We observed that all but one units are correctly classified, namely carrier FL in September, the only FL unit which was not flagged as an outlier.

Clustering the Flights units

Model-based Clustering described in Section **Model-based Clustering** was applied to the Flights dataset, without the identified outliers, to identify up to 16 components. This is accomplished by the command:

```
R> mclust_res <- Idtmclust(FlightsIdt[-out1,],1:16,Mxt="HomandHet")
```

where again, by default, the recommended solution is selected by the BIC. The corresponding values may be graphically compared using the command

```
R> plotInfCrt(mclust_res, cex.lab=1.5, outlegsize=10, outlegdisp=0.25)
```

which provided the graphic in Figure 8 and the output below. A homocedastic nine component model has been selected with an unrestricted covariance configuration C1.

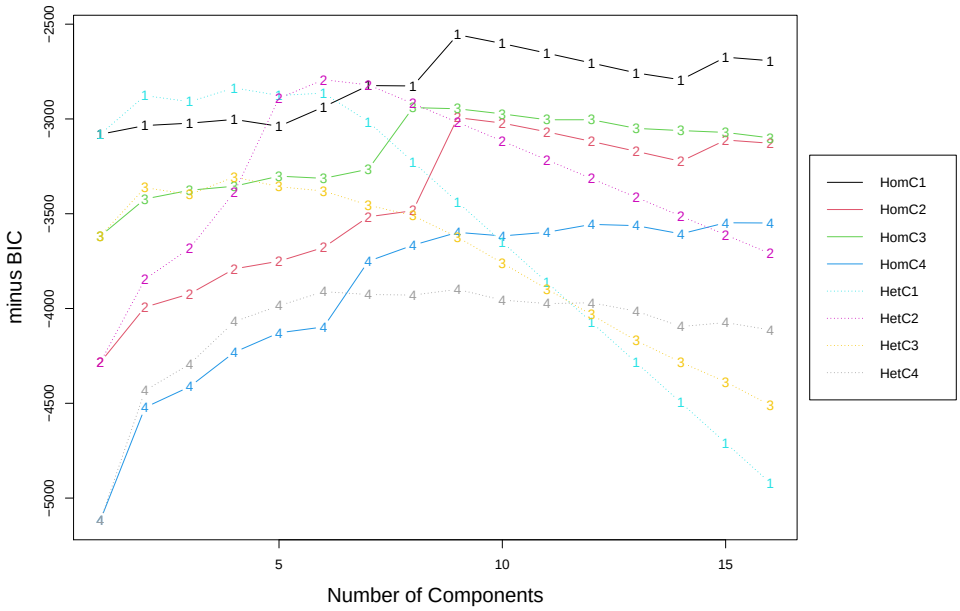


Figure 8: BIC values for different models and number of components.

```
Best BIC values:
HomG9C1    HomG10C1    HomG11C1    HomG15C1    HomG16C1
BIC          2555.407    2602.228    2653.101    2674.13    2692.822
BIC diff            0    46.82063    97.69343    118.7225    137.4153
```

The value returned by `Idtmclust` is an object of class `IdtMclust` with the same structure, and similar methods, of the corresponding `Mclust` class of package **mclust** (Scrucca et al., 2016). In particular, the classification results may be inspected by the command:

```
R> summary(mclust_res,classification=TRUE)
```

which returns

```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------
Homoscedastic C1 model with 9 components
```

```
log.likelihood NObs      BIC
-1005.076      110       2555.407

Clustering table:
CP1 CP2 CP3 CP4 CP5 CP6 CP7 CP8 CP9
12  12  14  12  10  12  24   9   5

Classification:
Jan-9E Jan-AA Jan-B6 Jan-DL Jan-EV Jan-MQ Jan-UA Jan-US Jan-WN Feb-9E
"CP5"  "CP2"  "CP7"  "CP4"  "CP5"  "CP6"  "CP7"  "CP3"  "CP3"  "CP1"
Feb-AA Feb-B6 Feb-DL Feb-EV Feb-MQ Feb-UA Feb-US Feb-WN Mar-9E Mar-AA
"CP2"  "CP7"  "CP4"  "CP5"  "CP6"  "CP7"  "CP3"  "CP3"  "CP1"  "CP2"
Mar-B6 Mar-DL Mar-EV Mar-MQ Mar-UA Mar-US Mar-WN Apr-9E Apr-AA Apr-B6
"CP7"  "CP4"  "CP5"  "CP6"  "CP7"  "CP3"  "CP8"  "CP1"  "CP2"  "CP7"
Apr-DL Apr-EV Apr-MQ Apr-UA Apr-US Apr-WN May-9E May-AA May-B6 May-DL
"CP4"  "CP5"  "CP6"  "CP7"  "CP3"  "CP8"  "CP5"  "CP2"  "CP7"  "CP4"
May-EV May-MQ May-UA May-US May-WN May-YV Jun-AA Jun-B6 Jun-DL Jun-EV
"CP1"  "CP6"  "CP7"  "CP3"  "CP8"  "CP9"  "CP2"  "CP7"  "CP4"  "CP5"
Jun-MQ Jun-UA Jun-US Jul-AA Jul-B6 Jul-DL Jul-EV Jul-MQ Jul-UA Jul-US
"CP6"  "CP7"  "CP3"  "CP2"  "CP7"  "CP4"  "CP5"  "CP6"  "CP7"  "CP3"
Jul-WN Jul-YV Aug-9E Aug-AA Aug-B6 Aug-DL Aug-EV Aug-MQ Aug-UA Aug-US
"CP8"  "CP9"  "CP1"  "CP2"  "CP7"  "CP4"  "CP1"  "CP6"  "CP7"  "CP3"
Aug-WN Sep-9E Sep-AA Sep-B6 Sep-DL Sep-EV Sep-FL Sep-MQ Sep-UA Sep-US
"CP8"  "CP1"  "CP2"  "CP7"  "CP4"  "CP1"  "CP9"  "CP6"  "CP7"  "CP3"
Sep-WN Oct-9E Oct-AA Oct-B6 Oct-DL Oct-EV Oct-MQ Oct-UA Oct-US Oct-WN
"CP8"  "CP1"  "CP2"  "CP7"  "CP4"  "CP1"  "CP6"  "CP7"  "CP3"  "CP8"
Oct-YV Nov-9E Nov-AA Nov-B6 Nov-DL Nov-EV Nov-MQ Nov-UA Nov-US Nov-WN
"CP9"  "CP1"  "CP2"  "CP7"  "CP4"  "CP1"  "CP6"  "CP7"  "CP3"  "CP8"
Dec-9E Dec-AA Dec-B6 Dec-DL Dec-EV Dec-MQ Dec-UA Dec-US Dec-WN Dec-YV
"CP5"  "CP2"  "CP7"  "CP4"  "CP5"  "CP6"  "CP7"  "CP3"  "CP8"  "CP9"
```

We observe that units corresponding to the same carrier tend to cluster together, for instance, component 2 gather all AA units, component 4 gathers DL units and component 6 the MQ units.

In order to have a better description of the obtained partition, we then print the corresponding mixing probabilities and the component-wise mean vectors.

```
R> print(pro(mclust_res), digits=3)
R> print(mean(mclust_res),digits=3)
```

obtaining

```
CP1     CP2     CP3     CP4     CP5     CP6     CP7     CP8     CP9
0.10684 0.10909 0.12727 0.10752 0.09316 0.10909 0.21976 0.08182 0.04545

                CP1     CP2     CP3     CP4    CP5    CP6     CP7     CP8    CP9
dep_delay.MidP  40.26   31.02   21.06   27.39  59.83  35.15   34.85   43.51  44.85
arr_delay.MidP  25.84   18.28   15.44   16.45  51.08  31.89   25.00   32.63  34.63
air_time.MidP   99.13  223.40  163.22  208.73 101.86  98.97  193.37  171.20  72.08
distance.MidP  640.43 1604.00 1163.86 1481.65 632.53 616.50 1360.07 1165.50 411.80
dep_delay.LogR   4.59    4.35    4.07    4.19   4.92   4.48    4.41    4.55   4.68
arr_delay.LogR   4.76    4.68    4.42    4.59   5.05   4.71    4.69    4.80   4.80
air_time.LogR    4.83    5.47    5.55    5.56   4.86   4.63    5.71    4.91   3.97
distance.LogR    6.86    7.46    7.58    7.59   6.81   6.69    7.75    6.81   5.85
```

These mean vectors can be compared by a parallel coordinate plot, using the `pcoordplot` method as follows

```
R> pcoordplot( mclust_res, cex.main=2, cex.lab=2,
     legendpar=list(cex.main=2.5, cex.lab=2.5) )
```

which produces the graph shown in Figure 9.

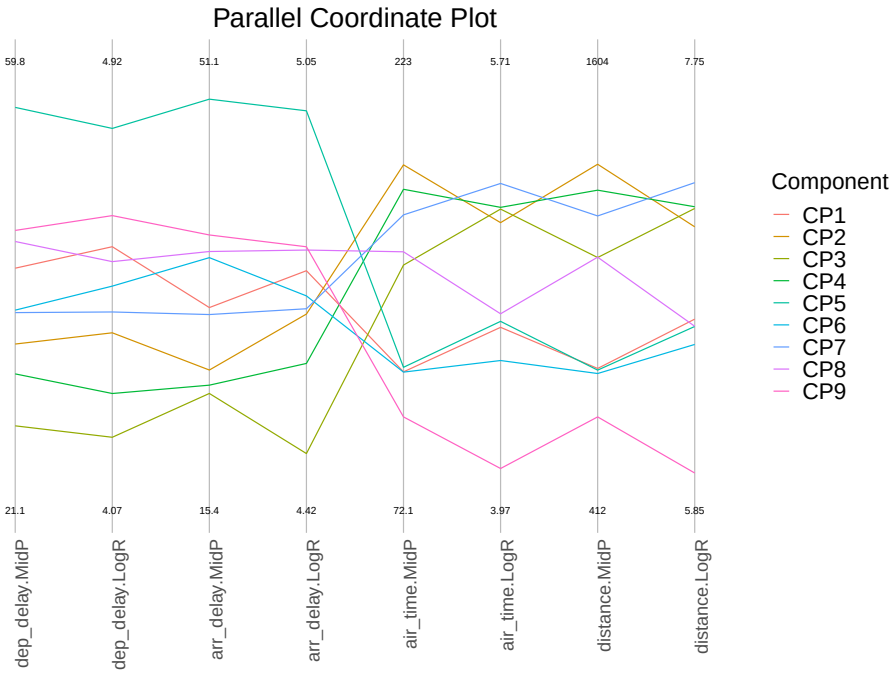


Figure 9: Parallel coordinate plot of best clustering solution, with nine components.

We observe that component 5 is mainly characterised by the largest delays both at departures and arrivals, also displaying their highest variability. Component 3, on the other hand, presents the lowest delays, with lowest variability, and concerns long flights. Component 9 corresponds to the shortest flights, with also low variability of distance and airtime. Components 4 and 7 present similar patterns in the distance and airtime variables, although component 4 displays slightly larger midpoints while component 7 has a higher variability; in terms of delays, we observe in component 7 higher values together with a more important variability.

From Figure 8 we observe that the best heterocedastic model corresponds to configuration C2 and identifies six components.

The corresponding mean vectors may again be displayed by a parallel coordinate plot, using the `pcoordplot` method, now indicating the solution of interest:

```
R> pcoordplot(mclust_res, model="HetG6C2", cex.main=2, cex.lab=2,
     legendpar=list(cex.main=2.5, cex.lab=2.5) )
```

leading to the graph shown in Figure 10.

Application II: diamonds dataset

This second example explores the diamonds dataset (from the R package **tidyverse** available at CRAN). The original microdata consists of 53940 diamonds characterised by ten variables. Descriptive variables are: *carat* (weight of the diamond), *x* (length in mm), y (width in mm), and *z* (depth in mm). All rows with missing data or null values in at least one of these variables were removed. Because the distribution of these variables is positive skewed, they were log-transformed (natural logarithm).

The units of analysis were defined by the variables: *cut* (quality of the cut: *Fair*, *Good*, *Very Good*, *Premium*, *Ideal*), *color* (diamond color, with seven levels: *J* (worst) to *D* (best)), and clarity (measurement of how clear the diamond is, with eight levels: *I1* (worst), *SI2*, *SI1*, *VS2*,

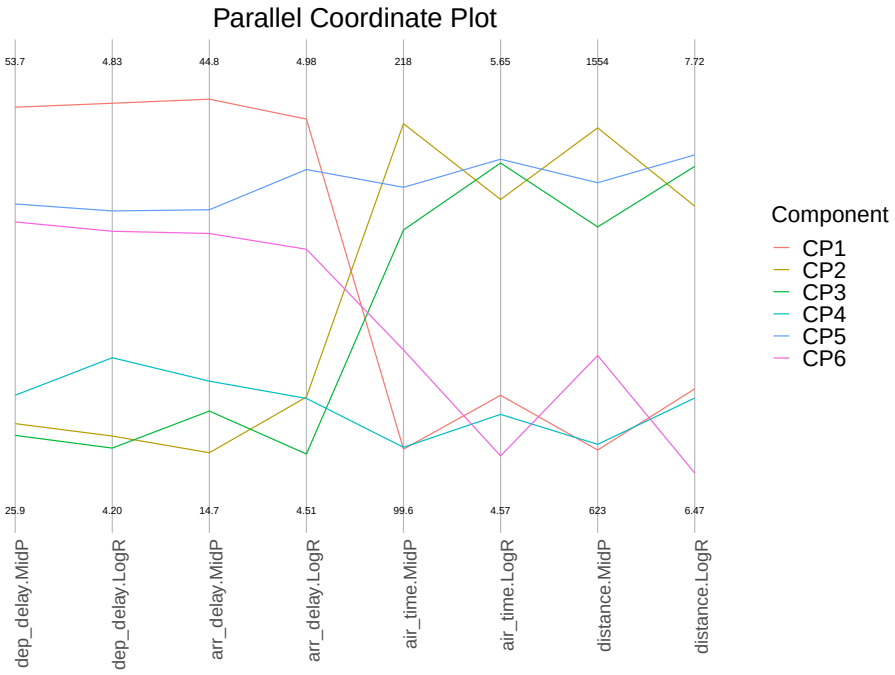


Figure 10: Parallel coordinate plot of best heterocedastic solution, with six components.

VS1, *VVS2*, *VVS1*, *IF* (best)), totalising 271 units (out of the 280, four were not present in the data, and five were degenerated). The variable *DiamondsUnits* defines these combinations.

The commands

```
R> library(tidyverse)

R> valid_diamonds  <- diamonds %>%
R> filter(carat !=0, x != 0, y != 0, z != 0) %>%
R> drop_na() %>% mutate(logcarat = log(carat),
                        logx     = log(x),
                        logy     = log(y),
                        logz     = log(z))

R> DiamondsUnits <- factor( paste(
    valid_diamonds$cut,valid_diamonds$color,valid_diamonds$clarity, sep="-"
  ) )

R> DiamondsIdt <- AgrMcDt(valid_diamonds[,c("logcarat","logx","logy","logz")],
                          agrby=DiamondsUnits)
```

do the initial data processing and create the interval data object *DiamondsIdt* using the default option (min-max). In the application we do not filter out potential outliers as we want to use the finite mixture model to detect them. Indeed, outliers can be seen as an unstructured component of the mixture model (Aitkin and Wilson, 1980).

From the estimation of mixtures from one to eight components, we have

```
R> Diamd_mclust_res <- Idtmclust(DiamondsIdt,1:8,Mxt="HomandHet")
R> plotInfCrt(Diamd_mclust_res, cex.lab=1.5, outlegsize=10, outlegdisp=0.25)

Best BIC values:
              HetG3C3     HetG4C3     HetG3C1     HetG5C3     HetG6C3
BIC         -7818.702   -7789.946   -7752.533   -7702.916   -7637.323
BIC diff            0    28.75589    66.16948    115.7865    181.3797
```

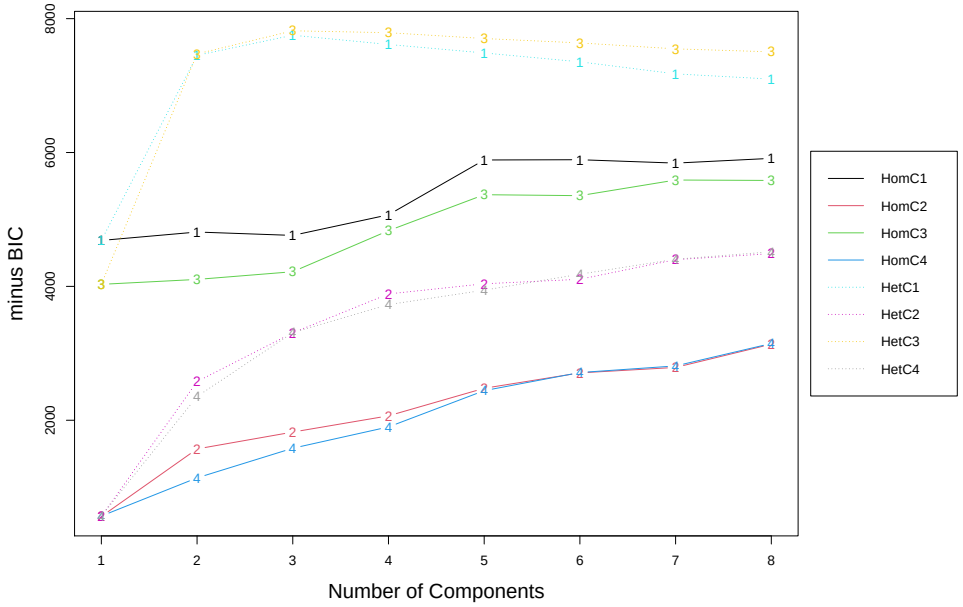


Figure 11: BIC values for different models and number of components.

Selection based on BIC recommends configuration 3 with three components (see Figure 11). Thus, the best solution is a heteroscedastic solution in which centers are not correlated with ranges.

```
R> summary(Diamd_mclust_res)
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------
Heteroscedastic C3 model with 3 components
 log.likelihood NObs       BIC
       4150.242  271 -7818.702

Clustering table:
CP1 CP2 CP3
 11 176  84

R> print(pro(Diamd_mclust_res), digits=3)
   CP1    CP2    CP3
0.0407 0.6328 0.3265
```

We conclude that the size of *CP1* is 0.0407 and contains eleven observations (hard classification), i.e., it is an outlier or niche group. Whenever a population/sample is well represented by the Normal distribution, a single component (or point of support) is enough to model its density. Often, however, distributions tend to have skewness and kurtosis that are far from the multivariate Normal distribution. In that case, Gaussian mixture models (GMM) have been used to estimate densities as an alternative to nonparametric or semi-parametric Kernels (Scott, 2015). Indeed, this application of finite mixtures is more general than model-based clustering as the latter tends to be specific to the correspondence between modes and clusters or groups. In this example, the departure from normality (skewness and kurtosis) is modeled using two additional components.

```
R> plot(MidPoints(DiamondsIdt)[,"logz.MidP"],
     LogRanges(DiamondsIdt) [,"logz.LogR"],
```

```
xlab="Mid Points",ylab="Log Ranges",
main="Mid Points vs. Log Ranges for logdepth in mm",
col = ifelse(Diamd_mclust_res@classification == 'CP1',1,
             ifelse(mclust_res@classification == 'CP2',2,7)),
             pch = 19, cex.lab=1.5)
```

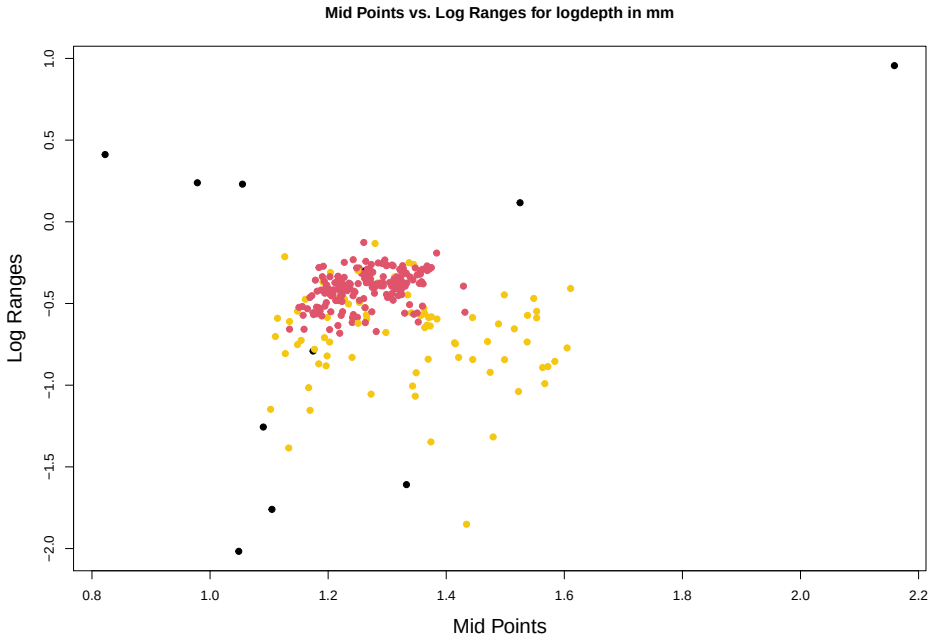


Figure 12: Representation of classified units (*logdepth*).

Figure 12 obtained from the code above illustrates the approximation of the density of the data by the finite mixture. The core group is well defined by the multivariate normal distribution. As the result of skewness, a second group is added. And then, finally, the heavy (multidimensional) "tail" is given by the outlier component (black dots) with its large estimated variances and co-variances.

Summary

The **MAINT.Data** R package implements models and methods for the analysis of interval-valued data, relying on multivariate Normal or Skew-Normal distributions for the MidPoints and Log-Ranges of the interval-valued variables. Implemented in the S4 framework, it introduces a data class for representing interval data and functions and methods for parametric modelling and analysis.

The available tools for interval variable management include interval-data versions of most of the standard R methods such as print and summary, index and subseting, and plot. Moreover, functions for aggregating microdata into interval data objects are also provided. The multivariate methodologies available include maximum likelihood estimation and statistical tests for the different configurations, (M)ANOVA, parametric Discriminant Analysis, and Model-based Clustering. Moreover, outlier detection and estimation based on robust techniques are provided; discriminant parametric methods based on robust estimates are implemented accordingly.

MAINT.Data, currently in its 2.6.1 version, offers an integrated solution for the management and parametric analysis of interval-valued data, from aggregation to modelling, analysis and visualisation, extending the R "programming with data" paradigm to new and complex data types.

Acknowledgements

This work was financed by the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within projects UIDB/00731/2020, UIDB/50014/2020, UIDB/00315/2020 and PTDC/EEI-TEL/32454/2017.

Bibliography

M. Aitkin and G. T. Wilson. Mixture models, outliers, and the EM algorithm. *Technometrics*, 22(3):325–331, 1980. URL https://doi.org/10.1080/00401706.1980.10486163. [p319]

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974. URL https://doi.org/10.1109/TAC.1974.1100705. [p300]

A. Azzalini. A class of distributions which includes the Normal ones. *Scandinavian Journal of Statistics*, 12:171–178, 1985. [p300]

A. Azzalini. The Skew-Normal distribution and related multivariate families. *Scandinavian Journal of Statistics*, 32:159–188, 2005. URL https://doi.org/10.1111/j.1467-9469.2005.00426.x. [p300, 301]

A. Azzalini. *The R package* sn*: The Skew-Normal and Related Distributions such as the Skew-t (version 2.0-0).* Università di Padova, Italia, 2021. URL http://azzalini.stat.unipd.it/SN. [p308]

A. Azzalini and A. Capitanio. Statistical applications of the multivariate Skew-Normal distribution. *Journal of the Royal Statistical Society Series B-Methodological*, 61(3):579–602, 1999. URL https://doi.org/10.1111/1467-9868.00194. [p301, 303]

A. Azzalini and A. Dalla Valle. The multivariate Skew-Normal distribution. *Biometrika*, 83 (4):715–726, 1996. URL https://doi.org/10.1093/biomet/83.4.715. [p297, 300]

J. D. Banfield and A. E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, pages 803–821, 1993. URL https://doi.org/10.2307/2532201. [p303]

B. Brahim and S. Makosso-Kallyth. *GPCSIV: Generalized Principal Component of Symbolic Interval variables*, 2013. URL https://CRAN.R-project.org/package=GPCSIV. R package version 0.1.0. [p298]

P. Brito. Symbolic Data Analysis: another look at the interaction of Data Mining and Statistics. *WIREs Data Mining and Knowledge Discovery*, 4(4):281–295, 2014. URL https://doi.org/10.1002/widm.1133. [p297, 298]

P. Brito and A. P. Duarte Silva. Modelling interval data with Normal and Skew-Normal distributions. *Journal of Applied Statistics*, 39(1):3–20, 2012. URL https://doi.org/10.1080/02664763.2011.575125. [p297, 298, 299, 300, 301, 302, 304]

P. Brito, A. P. Duarte Silva, and J. Dias. Probabilistic clustering of interval data. *Intelligent Data Analysis*, 19(2):293–313, 2015. URL https://doi.org/10.3233/IDA-150718. [p298, 304]

A. Cerioli. Multivariate outlier detection with high-breakdown estimators. *Journal of the American Statistical Association*, 105(489):147–156, 2010. URL https://doi.org/10.1198/jasa.2009.tm09147. [p301, 302]

F. De Carvalho and Y. Lechevallier. Partitional clustering algorithms for symbolic interval data based on single adaptive distances. *Pattern Recognition*, 42(7):1223–1236, 2009. URL https://doi.org/10.1016/j.patcog.2008.11.016. [p298]

F. De Carvalho, P. Brito, and H.-H. Bock. Dynamic clustering for interval data based on L_2 distance. *Computational Statistics*, 21(2):231–250, 2006. URL https://doi.org/10.1007/s00180-006-0261-z. [p298]

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B-Methodological*, 39(1):1–38, 1977. URL https://doi.org/10.1111/j.2517-6161.1977.tb01600.x. [p304]

S. Dias and P. Brito. Off the beaten track: A new linear model for interval data. *European Journal of Operational Research*, 258(3):1118–1130, 2017. URL https://doi.org/10.1016/j.ejor.2016.09.006. [p298]

E. Diday and M. Noirhomme-Fraiture. *Symbolic Data Analysis and the SODAS Software*. John Wiley & Sons, Chichester, 2008. [p297]

A. Douzal-Chouakria, L. Billard, and E. Diday. Principal component analysis for interval-valued observations. *Statistical Analysis and Data Mining*, 4(2):229–246, 2011. URL https://doi.org/10.1002/sam.10118. [p298]

A. P. Duarte Silva and P. Brito. Discriminant analysis of interval data: An assessment of parametric and distance-based approaches. *Journal of Classification*, 32(3):516–541, 2015. URL https://doi.org/0.1007/s00357-015-9189-8. [p298, 303]

A. P. Duarte Silva and P. Brito. *MAINT.Data: Model and Analyse Interval Data*, 2021. URL https://CRAN.R-project.org/package=MAINT.Data. R package version 2.6.1. [p298]

A. P. Duarte Silva, P. Filzmoser, and P. Brito. Outlier detection in interval data. *Advances in Data Analysis and Classification*, 12(3):785–822, 2018. URL https://doi.org/10.1007/s11634-017-0305-y. [p298, 301, 302]

A. Dudek, M. Pelka, J. Wilk, and M. Walesiak. *symbolicDA: Analysis of Symbolic Data*, 2019. URL https://CRAN.R-project.org/package=symbolicDA. R package version 0.6-2. [p298]

D. Eddelbuettel and R. François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011. URL https://doi.org/10.18637/jss.v040.i08. [p308]

C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002. URL https://doi.org/10.1198/016214502760047131. [p303]

R. François, D. Eddelbuettel, and D. Bates. *RcppArmadillo: Rcpp integration for Armadillo templated linear algebra library, R package (version 0.10.6.0.0)*, 2021. URL https://cran.r-project.org/web/packages/RcppArmadillo/index.html. [p308]

D. M. Gay. Usage summary for selected optimization routine. Technical Report 153, AT&T Bell Laboratories, Murray Hill, 1990. [p308]

A. Hadi and A. Luceño. Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics & Data Analysis*, 25(3):251–272, 1997. URL https://doi.org/10.1016/S0167-9473(97)00011-X. [p301]

J. Hardin and D. Rocke. The distribution of robust distances. *Journal of Computational and Graphical Statistics*, 14:910–927, 2005. URL https://doi.org/10.1198/106186005X77685. [p301, 302]

S. Hubeaux and K. Rufibach. *SurvRegCensCov: Weibull Regression for a Right-Censored Endpoint with Interval-Censored Covariate*, 2015. URL https://CRAN.R-project.org/package=SurvRegCensCov. R package version 1.4. [p298]

M. Hubert, P. Rousseeuw, and S. Van Aelst. High-breakdown robust multivariate methods. *Statistical Science*, 23(1):92–119, 2008. URL https://doi.org/10.1214/088342307000000087. [p301]

C. J. Huberty and S. Olejnik. *Applied MANOVA and Discriminant Analysis, 2nd Edition*. John Wiley & Sons, Chichester, 2006. [p302]

J. Le-Rademacher and L. Billard. Likelihood functions and some maximum likelihood estimators for symbolic data. *Journal of Statistical Planning and Inference*, 141(4):1593–1602, 2011. URL https://doi.org/10.1016/j.jspi.2010.11.016. [p298]

J. Le-Rademacher and L. Billard. Symbolic covariance principal component analysis and visualization for interval-valued data. *Journal of Computational and Graphical Statistics*, 21 (2):413–432, 2012. URL https://doi.org/10.1080/10618600.2012.679895. [p298]

E. Lima Neto and F. De Carvalho. Centre and range method for fitting a linear regression model to symbolic interval data. *Computational Statistics & Data Analysis*, 52(3):1500–1515, 2008. URL https://doi.org/10.1016/j.csda.2007.04.014. [p298]

E. Lima Neto and F. De Carvalho. Constrained linear regression models for symbolic interval-valued variables. *Computational Statistics & Data Analysis*, 54(2):333–347, 2010. URL https://doi.org/10.1016/j.csda.2009.08.010. [p298]

E. Lima Neto and F. De Carvalho. Bivariate symbolic regression models for interval-valued variables. *Journal of Statistical Computation and Simulation*, 81(11):1727–1744, 2011. URL https://doi.org/10.1080/00949655.2010.500470. [p298]

E. Lima Neto, C. De Souza Filho, and P. Marinho. *iRegression: Regression Methods for Interval-Valued Variables*, 2016. URL https://CRAN.R-project.org/package=iRegression. R package version 1.2.1. [p298]

G. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, New York, 2000. [p303]

G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley & Sons, 1992. [p303]

J. Messner, A. Zeileis, and R. Stauffer. *crch: Censored Regression with Conditional Heteroscedasticity*, 2019. URL https://CRAN.R-project.org/package=crch. R package version 1.0-4. [p298]

G. Pison, S. Van Aelst, and G. Willems. Small sample corrections for LTS and MCD. *Metrika*, 55(1-2):111–123, 2002. URL https://doi.org/10.1007/s001840200191. [p301]

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021. URL https://www.R-project.org/. [p298]

A. B. Ramos-Guajardo and P. Grzegorzewski. Distance-based linear discriminant analysis for interval-valued data. *Information Sciences*, 372:591–607, 2016. URL https://doi.org/10.1016/j.ins.2016.08.068. [p298]

O. Rodriguez. *RSDA: R to Symbolic Data Analysis*, 2021. URL https://CRAN.R-project.org/package=RSDA. R package version 3.0.9. [p298]

P. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984. URL https://doi.org/10.2307/2288718. [p301]

P. Rousseeuw. Multivariate estimation with high breakdown point. In W. Grossmann, G. Pflug, I. Vincze, and W. Wertz, editors, *Mathematical Statistics and Applications*, pages 283–297. Reidel Publishing, 1985. [p301]

P. Rousseeuw and K. Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999. URL https://doi.org/10.1080/00401706.1999.10485670. [p301, 309]

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978. URL https://doi.org/10.1214/aos/1176344136. [p300]

D. W. Scott. *Multivariate Density Estimation. Theory, Practice, and Visualization*. John Wiley & Sons, 2015. [p320]

L. Scrucca, M. Fop, T. B. Murphy, and A. E. Raftery. mclust 5: clustering, classification and density estimation using gaussian finite mixture models. *The R Journal*, 8(1):205–233, 2016. URL https://doi.org/10.32614/RJ-2016-021. [p309, 316]

G. A. Seber. *Multivariate Observations*, volume 252. John Wiley & Sons, 2009. [p309]

V. Todorov and P. Filzmoser. An object-oriented framework for robust multivariate analysis. *Journal of Statistical Software*, 32(3):1–47, 2009. URL https://doi.org/10.18637/jss.v032.i03. [p309]

R. B. Valle and A. Azzalini. The centred parametrization for the multivariate Skew-Normal distribution. *Journal of Mutivariate Analysis*, 99(4):1362–1382, 2008. URL https://doi.org/10.1016/j.jmva.2008.01.020. [p301, 308]

A. Pedro Duarte Silva
Católica Porto Business School & CEGE
Universidade Católica Portuguesa
Rua Diogo Botelho, 1327
4169-005 Porto, Portugal
E-mail: psilva@ucp.pt

Paula Brito
Faculdade de Economia, Universidade do Porto
& LIAAD-INESC TEC
Rua Dr. Roberto Frias
4200-464 Porto, Portugal
E-mail: mpbrito@fep.up.pt

Peter Filzmoser
Institute of Statistics and Mathematical Methods in Economics
TU Wien
Wiedner Hauptstraße 8-10
1040 Vienna, Austria
E-mail: Peter.Filzmoser@tuwien.ac.at

José G. Dias
Business Research Unit (BRU-IUL)
Instituto Universitário de Lisboa (ISCTE-IUL)
Av. das Forças Armadas
1648-026 Lisboa, Portugal
E-mail: jose.dias@iscte-iul.pt