MIXED DATA GENERATOR

Martin Matějka – Jiří Procházka – Zdeněk Šulc

Abstract

Very frequently, simulated data are required for quality evaluation of newly developed coefficients. In some cases, datasets with mixed-type variables (quantitative, ordinal and nominal) are required. Newly proposed modifications of the Gower similarity coefficient used for cluster analysis can be mentioned as example. In such situations, quantitative data itself are not sufficient in order to reflect the complexity and relationships of real datasets which are mostly based on mixed-type variables. Usually, a large amount of real datasets is required in order to get valid results in quality evaluation analysis. However, real datasets are very rare. Therefore, this paper presents an algorithm for generating mixed-type datasets containing a required structure, i.e. desired number of clusters, different number of categories, distance between clusters or their interpenetration, etc. In the paper, we demonstrate abilities of the proposed generator on several illustrative examples. For this purpose a MDG function is created and implemented into R package nomclust.

Key words: Mixed data, data generator, cluster analysis.

JEL Code: C38, C88

Introduction

Data generation is an important tool in various research methods. Often, it is used if there is lack of suitable real datasets for a given research question, or if there is need to validate a given experiment on several datasets with a priory known properties. For instance, when proposing a new similarity measure for purposes of cluster analysis, see e.g. (Morlini and Zani, 2012), or classification, see e.g. (Ahmad and Lipika, 2007), validation on both real and generated datasets is required. Based on real datasets, one might find out how the examined similarity measure performs in common situations, see e.g. (Löster, 2014); based on the generated datasets, one can examine properly their performance in various specific situations, such high (or low) number of variables, highly (or lowly) correlated structure, or presence (or absence) of outliers in datasets, see e.g. (Pan et al., 2000).

In this paper, we present a mixed-data generator, which we developed mainly for clustering and classification of objects characterized by both nominal and numeric variables. However, its use is not limited; thus, the generator can be used in any of multivariate statistical methods, e.g. in nonlinear principal component analysis, see (Linting et al., 2007), or regression. The generator was developed due to the lack of software implementation of generation algorithms for mixed-type of data. It is available as the MDG (Mixed Data Generator) function of the *nomclust* package, see (Šulc and Řezanková, 2015), on the Comprehensive R Archive Network (CRAN) web site[1].

The paper is organized as follows. The first section describes a theoretical background of the generator, and its setting options. The second section consists of two illustrations of use with the aim to clarify functionality and use of the generator.

1 Theoretical background of the mixed data generator

The mixed data generator is based on generating random values from multivariate normal distribution defined by the probability density function

$$f_{\mathbf{x}}(x_1,\ldots,x_k)=\frac{1}{\sqrt{(2\pi)^k|\mathbf{\Sigma}|}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{1}$$

where $\mathbf{x}$ is a real k-dimensional column vector (i.e. the k is a number of variables to be clustered), $\mathbf{\Sigma}$ is $k \times k$ covariance matrix and $\boldsymbol{\mu}$ is a k-dimensional mean vector. Thus, if the $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ parameters are known, it is possible to generate a sample of multivariate variables, which defines one particular population. If more populations should be created, then more the $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ parameters need to be defined for each independent population. Those populations create the whole dataset of initial numeric variables (i.e. the numeric dataset).

The numeric dataset needs to be subsequently discretized in order to create nominal variables. The discretization in the MDG function is performed by the function *discretize* included in the R package *arules*, see (Hahsler et al., 2015). This function offers possibility of creating nominal and ordered categorical variables. In addition, a user can choose a number of created categories and a discretization method. Four discretization methods are available:

1. *interval*, which provides equal interval widths;

[1] http://CRAN.R-project.org/

2. *frequency*, which provides equal frequencies of created categories;
3. *cluster,* which results in creating categories based on k-means clustering;
4. *fixed,* which provides categories based on the user defined specification of interval boundaries.

In the MDG function, it is possible to specify the input covariance matrices **Σ**. Due to a possible high dimensionality of generated data, it is very difficult to ensure that input covariance matrices will be at least positive definite as needed. Therefore, a user can specify whether input covariance matrices should be checked by the *Nearest Positive Definite Matrix* algorithm, which is provided by the function *nearPD* in the R package *Matrix.* For more details about this package, see (Bates and Maechler, 2015), where the algorithm proposed by Nicholas Higham, see (Higham, 2002), is used. In case an input covariance matrix is not positive definite, it will be automatically transformed into the nearest positive definite matrix by the above mentioned algorithm.

The MDG function provides several possibilities regarding specifications of covariance matrices. In general, three situations can occur:

1. *Covariance matrices are known,* which results in no additional action of the mixed data generator;
2. *Covariance matrices are unknown*, which corresponds with two reasons:

a) *Variances in diagonals and correlation matrices are known,* then the covariance matrices are standardly calculated by formula

$$\mathbf{\Sigma} = diag\{\mathbf{\Sigma}\}^{1/2}\,\mathbf{P}\,diag\{\mathbf{\Sigma}\}^{1/2}, \tag{2}$$

where **P** is a correlation matrix and $diag\{\mathbf{\Sigma}\}^{1/2}$ is a diagonal matrix of known variances.

b) *Variances in diagonals of the covariance matrices are known but the correlation matrices are not known.* In such a situation, correlation matrices need to be automatically generated, subsequently the corresponding covariance matrices are calculated using Eq. (2).

The correlation matrix generation is based on creation of the random dataset. In the first step, an initial random vector (the lower bound, upper bound of the uniform distribution and the length of this vector need to be specified) is generated. Subsequently, errors from uniform distribution (from 0 to 1) are randomly generated and then transformed to take real values from – 1 to 1. Based on user defined increment, an arithmetic sequence of the same length as the desired number of the random dataset column is created. In addition, its odd elements are transformed to be negative and even elements were left positive. The last step is to attribute the errors multiplied by a particular element of the arithmetic sequence to the initial random vector.

For the complete list of attributes of the MDG function, see Table 1.

Tab. 1: Arguments of the MDG function

Argument	Description
discretization	The type of a discretization when transforming quantitative variable into a categorical variable.
corel_check	A logical parameter indicating whether the correlation or covariance matrices should be checked using *nearPD* function.
distance	A distance vector containing values to be used if the matrix of *means* contains only one vector.
means	The matrix of means defining the multivariate normal distribution. In case only the initial vector of means is specified it is multiplied by the values included in the distance vector.
dimension	The number of dimensions of multivariate normal distribution.
samples	A vector of sample sizes to be independently created using multivariate normal distribution.
cormat	A list of input correlation matrices (the number of matrices must be equal to the length of *samples*). If the *vars* matrix is the only input (i.e. *cormat* and *covmat* are empty lists) the correlation matrices are automatically generated.
covmat	A list of input covariance matrices (the number of matrices must be equal to the length of *samples*). If the *covmat* matrices are not filled they are calculated based on *vars* matrix and *cormat* (generated if necessary).
todistr	A logical vector indicating which variable should be discretized.
ordin	A logical vector indicating which variables (of those to be discretized) should be ordinal.
ncat	A vector indicating a number of created categories for each particular discretized variable.
vars	A matrix of variances for each variable in each population (thus the dimension of this matrix is length of *samples* x *dimension*)
initial_rands_min	A lower bound of the initial random vector. The initial random vector is used for generating the random dataset which is subsequently used for automatic generating of the unknown correlation matrix.
initial_rands_max	An upper bound of the initial random vector.

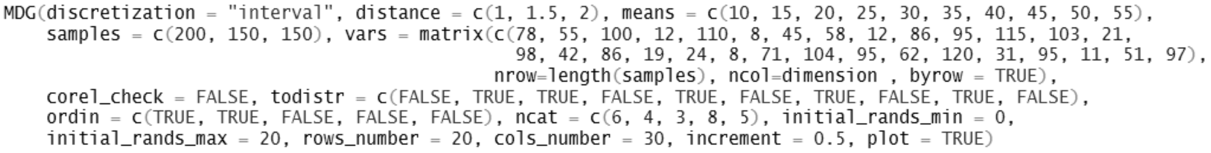
Argument	Description
rows_number	The number of rows of the random dataset.
cols_number	The number of columns of the random dataset. The correlation matrix is then calculated using only a subset of *cols_number* which is equal to the *dimension*.
increment	An initial random vector is increased or decreased based on the arithmetic sequence starting from 1 with the increment. In addition the odd elements of the arithmetic sequence are negative and even elements are then positive.
plot	A logical parameter indicating whether the *clustplot* (see Figures 1 and 2) should be shown.

Source: Authors specification

2 Illustration of the MDG application

In this section, a practical application of the MDG function will be illustrated on two examples. Assume that a modification of the Gower similarity coefficient, see (Gower, 1971), is to be developed. Therefore, it is very useful to evaluate the clustering performance of the modified similarity coefficient on several generated mixed datasets.

The first simulated dataset should be created in respect to sufficient distinguishing of clusters. Thus, 500 observations of 10 variables (2 of them will be ordinal and 3 will be nominal variables) with mutual relationships will be divided into 3 separate clusters. In addition, the correlation matrices are unknown, so they will be generated. The following syntax of the MDG function can be used:



```
MDG(discretization = "interval", distance = c(1, 1.5, 2), means = c(10, 15, 20, 25, 30, 35, 40, 45, 50, 55),
    samples = c(200, 150, 150), vars = matrix(c(78, 55, 100, 12, 110, 8, 45, 58, 12, 86, 95, 115, 103, 21,
                                                98, 42, 86, 19, 24, 8, 71, 104, 95, 62, 120, 31, 95, 11, 51, 97),
                                              nrow=length(samples), ncol=dimension , byrow = TRUE),
    corel_check = FALSE, todistr = c(FALSE, TRUE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE),
    ordin = c(TRUE, TRUE, FALSE, FALSE, FALSE), ncat = c(6, 4, 3, 8, 5), initial_rands_min = 0,
    initial_rands_max = 20, rows_number = 20, cols_number = 30, increment = 0.5, plot = TRUE)
```

As a *discretization* method, the interval method was chosen. The number of categories is specified by the arguments *todistr*, *ordin* and *ncat*. The second argument *means* is specified as a vector containing expected values for each marginal distribution of the particular multivariate normal distribution. The vector of means is amended for the particular population using the *distance* argument which shifts the initial means vector according to the specified values. The argument *vars* is specified as a matrix containing variances specified for each marginal distribution in each population (the size of populations is defined by the *samples* argument). The next argument is *corel_check* set to FALSE indicating that correlation matrices are not checked to be positive definite. This is not needed as those matrices are automatically generated, which is the case because neither *cormat* nor *covmat* arguments are specified. The arguments *initial_rands_min*, *initial_rands_max*, *rows_number*, *cols_number* and increment specify a temporary generated data set for creation of generated correlation matrices. The last argument *plot* set to TRUE provides a graphical output of the generated data. The following table describes a subset of 10 randomly sampled observations of the simulated dataset, where the variables V1, V4, V6, V8 and V10 are numeric and thus they were not changed to be categorical. On the other side the variables V2 and V3 are ordinal. Finally the remaining variables are nominal (e.g. V5, V7 and V9).

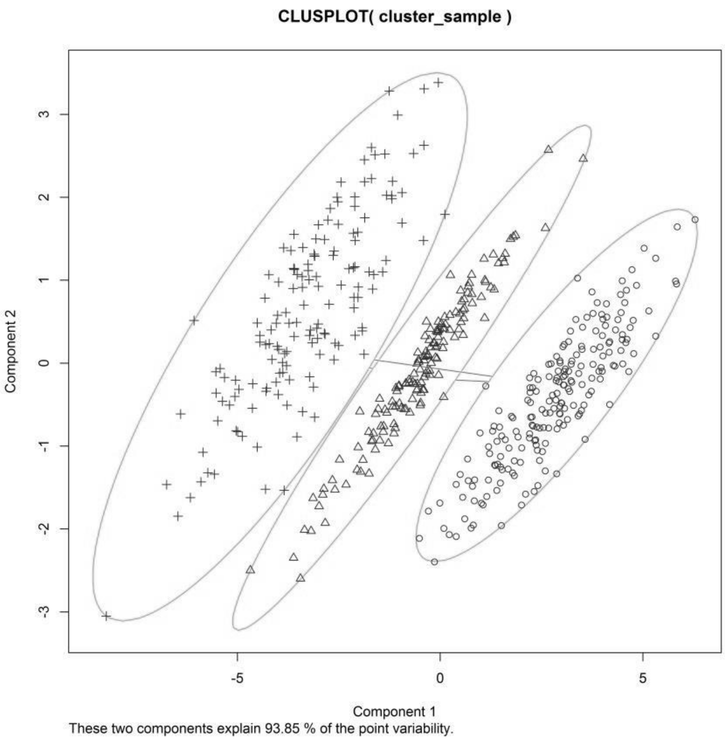
Tab. 2: Subset of the simulated dataset

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
23.909	4	3	49.817	3	71.468	7	91.495	4	114.549
8.344	3	2	24.283	2	34.181	2	44.340	1	56.878
28.893	5	3	56.053	3	76.962	8	92.600	5	120.794
5.373	2	2	23.410	1	34.845	2	39.124	1	46.598
1.984	2	1	21.861	1	32.747	2	38.571	1	53.146
26.158	4	3	31.132	2	38.531	3	59.868	1	65.205
-10.638	2	1	27.991	1	35.643	3	55.245	2	75.344
21.124	3	2	38.183	2	61.246	4	71.385	3	83.979
1.297	3	1	21.152	1	30.288	3	42.304	1	67.490
13.934	3	2	26.297	2	34.850	3	50.528	1	63.945

Source: Authors calculations

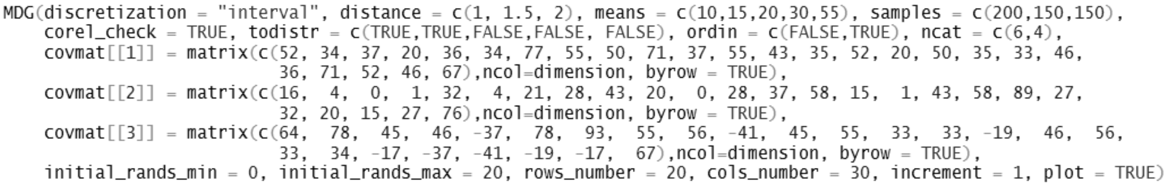
Fig. 1 represents the structure of the generated non discretized dataset using principal component analysis. The purpose of this figure is to transform the multivariate numerical data set into only two dimensions for convenient graphical representation.

Fig. 1: Structure of dataset presented in Table 2 (before discretization)



Source: Authors calculations

The second example of the use of the MDG function represents a similar situation as specified above, but the major difference is that now the covariance matrices are known, but they need to be checked if they are positive definite. In addition, it is required to receive less compact clusters than those in the first example, and the dataset should contain 5 variables (the first variable will be nominal and the second one ordinal).



```
MDG(discretization = "interval", distance = c(1, 1.5, 2), means = c(10,15,20,30,55), samples = c(200,150,150),
    corel_check = TRUE, todistr = c(TRUE,TRUE,FALSE,FALSE, FALSE), ordin = c(FALSE,TRUE), ncat = c(6,4),
    covmat[[1]] = matrix(c(52, 34, 37, 20, 36, 34, 77, 55, 50, 71, 37, 55, 43, 35, 52, 20, 50, 35, 33, 46,
                           36, 71, 52, 46, 67),ncol=dimension, byrow = TRUE),
    covmat[[2]] = matrix(c(16,  4,  0,  1, 32,  4, 21, 28, 43, 20,  0, 28, 37, 58, 15,  1, 43, 58, 89, 27,
                           32, 20, 15, 27, 76),ncol=dimension, byrow = TRUE),
    covmat[[3]] = matrix(c(64,  78,  45,  46, -37,  78,  93,  55,  56, -41,  45,  55,  33,  33, -19,  46,  56,
                           33,  34, -17, -37, -41, -19, -17,  67),ncol=dimension, byrow = TRUE),
    initial_rands_min = 0, initial_rands_max = 20, rows_number = 20, cols_number = 30, increment = 1, plot = TRUE)
```

All input covariance matrices were checked to be positive definite and amended accordingly to meet this criterion. The first covariance matrix has the following structure after its positive definite checking.

Tab. 3: The first positive definite corrected covariance matrix

	V1	V2	V3	V4	V5
V1	52.04	34.06	36.88	20.02	35.99
V2	34.06	77.09	54.83	50.03	70.99
V3	36.88	54.83	43.34	34.94	52.03
V4	20.02	50.03	34.94	33.01	46.00
V5	35.99	70.99	52.03	46.00	67.00

Source: Authors calculations

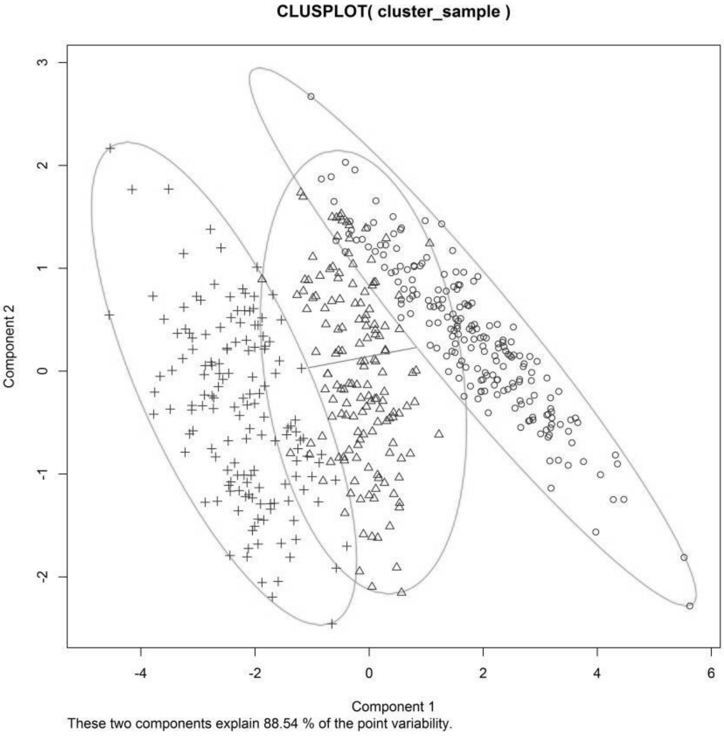
Figure 2 represents the structure of the generated non discretized dataset using principal component analysis.

Conclusion

In this paper, the developed MDG function for generating mixed-type datasets was presented. In order to provide a user-friendly approach, the generator allows specifying all input arguments, or only a desired subset of these arguments. For instance, it is possible to set an input covariance matrix expressing variability among variables, or to set only corresponding variances with an automatic generation of a correlation matrix. Thus, a user do not have to be bothered by setting a large number of parameters.

The MDG function also allows checking input covariance or correlation matrices using the *Nearest Positive Definite Matrix* algorithm. The graphical representation of generated mixed data is provided by the *clusplot* function, which describes the first two principal components of the generated data. The use of the MDG function was illustrated on two basic examples. First, there was shown a demonstration of a mixed-type data generation with well-separated clusters without any knowledge about linear relationships between variables. Second, generation of a mixed-type data with a known covariance structure was demonstrated. The *Nearest Positive Definite Matrix* algorithm was used in order to assumptions of the correlation and covariance matrices were met. The MDG function is publicly available via the R package *nomclust*.

Fig. 2: Structure of the second generated non discretized dataset



Source: Authors calculations

As a further research it is necessary to do several modifications allowing user to create more general multivariate random variables. It means that the user will be able to control excess and asymmetry of multivariate distribution for each cluster.

Acknowledgment

This work was supported by the University of Economics, Prague under Grant IGA F4/41/2016.

References

Ahmad, A., Lipika, D. (2007). A k-mean clustering algorithm for mixed numeric and categorical data. In *Data & Knowledge Engineering 63*, 503-527.

Bates, D., Maechler, M. (2015). Matrix: Sparse and Dense Matrix Classes and Methods. R package version 1.2-3. https://CRAN.R-project.org/package=Matrix

Gower, J. C. (1971). A general coefficient of similarity and some of its properties. In *Biometrics*, *28*(4), 857-871.

Hahsler, M., Buchta, C., Gruen, B., Hornik, K. (2015). Arules: Mining Association Rules and Frequent Itemsets. R package version 1.3-1. https://CRAN.R-project.org/package=arules

Higham, N. J. (2002). Computing the nearest correlation matrix. In *Oxford University Press*, 329-343.

Linting, M., Meulman, J., Groenen, P. J., Kooij, A. J. (2007). Nonlinear principal components analysis: introduction and application. In *Psychological Methods 12*(3), 336-358.

Löster, T. (2014). The evaluation of CHF coefficient in determining the number of clusters using Euclidean distance measure. In *The 8th International Days of Statistics and Economics*. Slaný: Melandrium, 858-869. https://msed.vse.cz/msed_2014/article/463-Loster-Tomas-paper.pdf.

Morlini, I., Zani, S. (2012). A new class of weighted similarity indices using polytomous variables. In *Journal of Classification*, *29*(2), 199-226.

Pan, J. X., Fung, W. K., Fang, K. T. (2000). Multiple outlier detection in multivariate data using projection pursuit techniques. In *Journal of Statistical Planning and Inference*, *83*(1), 153-167.

R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/

Šulc, Z., Řezanková, H. (2015). Nomclust: An r package for hierarchical clustering of objects characterized by nominal variables. In *The 9th International Days of Statistics and Economics*. Slaný: Melandrium, 1581-1590. https://msed.vse.cz/msed_2015/article/48-Sulc-Zdenek-paper.pdf.

Contact

Martin Matějka

University of Economics, Prague, Dept. of Statistics and Probability

W. Churchill sq. 4, 130 67 Prague 3, Czech Republic

martin.matejka@vse.cz

Jiří Procházka

University of Economics, Prague, Dept. of Statistics and Probability

W. Churchill sq. 4, 130 67 Prague 3, Czech Republic

jiri.prochazka@vse.cz

Zdeněk Šulc

University of Economics, Prague, Dept. of Statistics and Probability

W. Churchill sq. 4, 130 67 Prague 3, Czech Republic

zdenek.sulc@vse.cz