

# Package ‘zCompositions’

February 18, 2019

**Type** Package

**Title** Treatment of Zeros, Left-Censored and Missing Values in  
Compositional Data Sets

**Version** 1.2.0

**Date** 2019-02-18

**Author** Javier Palarea-Albaladejo and Josep Antoni Martin-Fernandez

**Maintainer** Javier Palarea-Albaladejo <javier.palarea@bioss.ac.uk>

**Depends** R (>= 2.14.0), MASS, NADA, truncnorm

**ByteCompile** yes

**Description** Principled methods for the imputation of zeros, left-censored and missing data in  
compositional data sets.

**License** GPL (>= 2)

**Repository** CRAN

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Date/Publication** 2019-02-18 11:30:02 UTC

## R topics documented:

cmultRepl . . . . .	2
lcTest . . . . .	4
LPdata . . . . .	5
LPdataZM . . . . .	6
lrDA . . . . .	7
lrEM . . . . .	9
lrEMplus . . . . .	12
mdl . . . . .	15
multKM . . . . .	15
multLN . . . . .	17
multRepl . . . . .	19
Pigs . . . . .	21

splineKM . . . . .	22
Water . . . . .	23
zCompositions . . . . .	24
zPatterns . . . . .	26
zVarArray . . . . .	28
zVarArrayError . . . . .	29
zVarArrayTest . . . . .	30

<b>Index</b>	<b>32</b>
--------------	-----------

---

cmultRepl	<i>Bayesian-Multiplicative replacement of count zeros</i>
-----------	---

---

## Description

This function implements methods for imputing zeros in compositional count data sets based on a Bayesian-multiplicative replacement.

## Usage

```
cmultRepl(X, label = 0,
          method = c("GBM", "SQ", "BL", "CZM", "user"), output = c("prop", "p-counts"),
          delta = 0.65, threshold = 0.5, correct = TRUE, t = NULL, s = NULL,
          suppress.print = FALSE)
```

## Arguments

X	Count data set ( <a href="#">matrix</a> or <a href="#">data.frame</a> class).
label	Unique label ( <a href="#">numeric</a> or <a href="#">character</a> ) used to denote count zeros in X (default label=0).
method	Geometric Bayesian multiplicative (GBM, default); square root BM (SQ); Bayes-Laplace BM (BL); count zero multiplicative (CZM); user-specified hyper-parameters (user).
output	Output format: imputed proportions (prop, default) or <i>pseudo</i> -counts (p-counts).
delta	If method="CZM", fraction of the upper threshold used to impute zeros (default delta=0.65). Also, fraction of the lowest estimated probability used to correct imputed proportions falling above it (when correct=TRUE).
threshold	For a vector of counts, factor applied to the quotient 1 over the number of trials (sum of the counts) used to produce an upper limit for replacing zero counts by the CZM method (default threshold=0.5).
correct	Logical vector setting whether imputed proportions falling above the lowest estimated probability for a multinomial part must be corrected or not (default correct=TRUE).
t	If method="user", user-specified <i>t</i> hyper-parameter of the Dirichlet prior distribution for each count vector (row) in X. It must be a matrix of the same dimensions as X.

- `s` If `method="user"`, user-specified  $s$  hyper-parameter of the Dirichlet prior distribution for each count vector (row) in  $X$ . It must be a vector of length equal to the number of rows of  $X$ .
- `suppress.print` Suppress printed feedback (`suppress.print=FALSE`, default).

### Details

Zero counts, assumed to be due to under-reporting or limited sampling, are imputed under a Bayesian paradigm (GBM, SQ or BL method) by posterior estimates of the multinomial probabilities generating the counts, assuming a Dirichlet prior distribution. The argument `method` sets the Dirichlet hyper-parameters `t` (priori estimates of multinomial probabilities) and `s` (*strength*). The user can specify their own by setting `method="user"` and entering them as `t` and `s` arguments. Note that, under certain circumstances (see references for details), these methods can generate imputed proportions falling above the lowest estimated probability of a multinomial part ( $c/n$ , where  $c$  is the count and  $n$  is the number of trials). In such cases, the imputation is corrected by using a fraction (`delta`) of the minimum  $c/n$  for that part. Lastly, the non-zero parts are multiplicatively adjusted according to their compositional nature.

On the other hand, `method="ZM"` uses multiplicative simple replacement (`multRepl`) on the matrix of estimated probabilities. The upper limit and the fraction `delta` used are specified by, respectively, the arguments `threshold` and `delta`. Suggested values are `threshold=0.5` (so the upper limit for a multinomial probability turns out to be  $0.5/n$ ), and `delta=0.65` (so the imputed proportion is 65% of the upper limit).

### Value

By default (`output="prop"`) the function returns an imputed data set (`data.frame` class) in proportions (estimated probabilities). Alternatively, these proportions are re-scaled to produce a compositionally-equivalent matrix of *pseudo*-counts (`output="p-counts"`) which preserves the ratios between parts.

When `correct=TRUE` and `verbose=TRUE`, the number of times, if any, an imputed proportion was corrected to fall below the minimum estimated multinomial probability is printed.

### References

Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J. Bayesian-multiplicative treatment of count zeros in compositional data sets. *Statistical Modelling* 2015; 15 (2): 134-158.

### See Also

[zPatterns](#)

### Examples

```
data(Pigs)

# GBM method and matrix of estimated probabilities
Pigs.GBM <- cmultRepl(Pigs)
```

---

 lcTest

*Log-contrast homogeneity test*


---

### Description

This function tests for homogeneity across groups of means and variances of user-defined log-contrasts. Groups can be defined by either zero/unobserved data patterns or by a grouping factor in fully observed zero-free data sets.

### Usage

```
lcTest(X, label = 0, groups = NULL, lc = NULL, method = c("parametric",
  "nonparametric"), b = 1000)
```

### Arguments

X	Compositional data set ( <a href="#">matrix</a> or <a href="#">data.frame</a> class).
label	Unique label ( <a href="#">numeric</a> or <a href="#">character</a> ) used to denote zero or unobserved data in X (label = 0, default).
groups	Grouping factor in fully observed zero-free data sets (groups = NULL, default).
lc	User-defined log-contrast (see details below).
method	Approach used for mean and variance homogeneity testing (method = "parametric", default).
b	Number of bootstrap resamples used by permutation test (b = 1000, default).

### Details

Homogeneity of log-contrast means and variances across groups is tested using either parametric or non-parametric tests. When method = "parametric", ordinary analysis of variance and Bartlett's tests are used. Alternatively, Kruskal-Wallis and Fligner-Killen tests are used instead when method = "nonparametric". The results of a permutation test of homogeneity of variation arrays based on total weighted squared relative errors are also provided (see [zVarArrayTest](#) for more details). The log-contrast is specified by the lc argument using a vector of codes 1, -1 and 0 for components in the numerator, denominator and omitted respectively.

### Value

Test p-values for log-contrast means and variances.

### See Also

[zPatterns](#), [zVarArray](#), [zVarArrayError](#)

**Examples**

```
data(Water)
zPatterns(Water, label = 0)

# Test of homogeneity in log-contrast Potassium/Arsenic*Calcium
lcTest(Water, label = 0, lc = c(1,-1,-1,0))
```

---

LPdata

*La Paloma data set*

---

**Description**

96 samples of a 15-part geochemical composition in micrograms/gram from La Paloma stream (Venezuela) including 6.11% values below the limit of detection (coded as 0). For more details see Montero-Serrano et al. (2010).

**Usage**

```
data(LPdata)
```

**Format**

A [data.frame](#) with 96 observations on the following 15 variables.

Cr a numeric vector  
B a numeric vector  
P a numeric vector  
V a numeric vector  
Cu a numeric vector  
Ti a numeric vector  
Ni a numeric vector  
Y a numeric vector  
Sr a numeric vector  
La a numeric vector  
Ce a numeric vector  
Ba a numeric vector  
Li a numeric vector  
K a numeric vector  
Rb a numeric vector

**References**

Montero-Serrano JC, Palarea-Albaladejo J, Martin-Fernandez JA, and Martinez-Santana M and Gutierrez-Martin JV. Multivariate analysis applied to chemostratigraphic data: identification of chemofacies and stratigraphic correlation, *Sedimentary Geology* 2010; 228(3-4): 218-228 .

**Examples**

```
data(LPdata)

zPatterns(LPdata, label=0)
```

---

LPdataZM

*La Paloma data set (incl. zeros and missing data)*

---

**Description**

96 samples of a 15-part geochemical composition in micrograms/gram from La Paloma stream (Venezuela). For more details see Montero-Serrano et al. (2010).

Duplicate of the LPdata data set including 2.36% missing at random cells (35.42% samples with missing data; coded as NA) along with 6.11% values below the limit of detection (coded as 0).

**Usage**

```
data(LPdataZM)
```

**Format**

A [data frame](#) with 96 observations on the following 15 variables.

Cr a numeric vector  
B a numeric vector  
P a numeric vector  
V a numeric vector  
Cu a numeric vector  
Ti a numeric vector  
Ni a numeric vector  
Y a numeric vector  
Sr a numeric vector  
La a numeric vector  
Ce a numeric vector  
Ba a numeric vector  
Li a numeric vector  
K a numeric vector  
Rb a numeric vector

**References**

Montero-Serrano JC, Palarea-Albaladejo J, Martin-Fernandez JA, and Martinez-Santana M and Gutierrez-Martin JV. Multivariate analysis applied to chemostratigraphic data: identification of chemofacies and stratigraphic correlation, *Sedimentary Geology* 2010; 228(3-4): 218-228 .

**See Also**[LPdata](#)**Examples**

```
data(LPdataZM)

zPatterns(LPdataZM,label=0) # Show zero patterns

zPatterns(LPdataZM,label=NA) # Show missingness patterns
```

lrDA

*Log-ratio DA algorithm***Description**

This function implements a simulation-based Data Augmentation (DA) algorithm to impute left-censored values (e.g. values below detection limit, rounded zeros) via coordinates representation of compositional data sets which incorporate the information of the relative covariance structure. Multiple imputation estimates can be also obtained from the output.

**Usage**

```
lrDA(X, label = NULL, dl = NULL,
      ini.cov=c("lrEM", "complete.obs", "multRepl"),
      delta = 0.65, n.iters = 1000, m = 1, store.mi = FALSE)
```

**Arguments**

<code>X</code>	Compositional data set ( <a href="#">matrix</a> or <a href="#">data.frame</a> class).
<code>label</code>	Unique label ( <a href="#">numeric</a> or <a href="#">character</a> ) used to denote unobserved left-censored values in <code>X</code> .
<code>dl</code>	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as <code>X</code> .
<code>ini.cov</code>	Initial estimation of the log-ratio covariance matrix. It can be based on lrEM estimation (" <a href="#">lrEM</a> ", default), complete observations (" <a href="#">complete.obs</a> ") or multiplicative simple replacement (" <a href="#">multRepl</a> ").
<code>delta</code>	If <code>ini.cov="multRepl"</code> , <code>delta</code> parameter for initial multiplicative simple replacement ( <a href="#">multRepl</a> ) in proportions (default = 0.65).
<code>n.iters</code>	Number of iterations for the DA algorithm (default = 1000).
<code>m</code>	Number of multiple imputations (default = 1).
<code>store.mi</code>	Logical value. If <code>m&gt;1</code> creates a list with <code>m</code> imputed data matrices. ( <code>store.mi=FALSE</code> , default).

## Details

After convergence of the Markov chain Monte Carlo (MCMC) iterative process to its steady state, this function imputes left-censored compositional parts by simulated values from their posterior predictive distributions through coordinates representation, given the information from the observed data and the censoring thresholds. It allows for either single (vector form) or multiple (matrix form, same size as  $X$ ) limits of detection by component. Any threshold value can be set for non-censored elements (e.g. use 0 if no threshold for a particular column or element of the data matrix).

It produces imputed data sets on the same scale as the input data set. If  $X$  is not closed to a constant sum, then the results are adjusted to provide a compositionally equivalent data set, expressed in the original scale, which leaves the absolute values of the observed components unaltered.

The common conjugate normal inverted-Wishart distribution with non-informative prior has been assumed for the model parameters in the coordinates space. Under this setting, convergence is expected to be fast (`n.itors` set to 1000 by default). Besides, considering EM parameter estimates as initial point for the DA algorithm (`ini.cov="lrEM"`) assures faster convergence by starting near the centre of the posterior distribution. Note that the procedure is based on the oblique additive log-ratio (alr) transformation to simplify calculations and alleviates computational burden.

By setting `m` greater than 1, the procedure also allows for multiple imputations of the censored values drawn at regular intervals after convergence. In this case, in addition to the burn-in period for convergence, `n.itors` determines the gap, large enough to prevent from correlated values, between successive imputations. The total number of iterations is then `n.itors*m`. By default, a single imputed data set results from averaging the `m` imputations in the space of coordinates. If `store.mi=TRUE`, a list with `m` imputed data sets is generated instead.

In the case of censoring patterns involving samples containing only one observed component, these are imputed by multiplicative simple replacement (`multRepl`) and a warning message identifying them is printed.

## Value

A `data.frame` object containing the imputed compositional data set in the same scale as the original, or a `list` of imputed data sets if multiple imputation is carried out (`m>1`) and `store.mi=TRUE`.

## References

Palarea-Albaladejo J, Martin-Fernandez JA, Olea, RA. A bootstrap estimation scheme for chemical compositional data with nondetects. *Journal of Chemometrics* 2014; 28: 585-599.

## See Also

`zPatterns`, `lrEM`, `multRepl`, `multLN`, `multKM`, `cmultRepl`

## Examples

```
# Data set closed to 100 (percentages, common dl = 1%)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,26.20,0.00,15.22,6.80,12.05,
              10.76,31.36,7.10,12.74,31.34,6.70,
              10.85,46.40,31.89,10.86,0.00,0.00,
              7.57,11.35,30.24,6.39,13.65,30.80,
```



```

38.09,7.62,23.68,9.70,20.91,0.00,
27.67,7.15,13.05,32.04,6.54,13.55,
44.41,15.04,7.95,0.00,10.82,21.78,
11.50,30.33,6.85,13.92,30.82,6.58,
19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

# Imputation by single simulated values
X_lrDA <- lrDA(X,label=0,dl=rep(1,6),ini.cov="multRepl",n.iters=150)

# Imputation by multiple imputation (m = 5, one imputation every 150 iterations)
X_milrDA <- lrDA(X,label=0,dl=rep(1,6),ini.cov="multRepl",m=5,n.iters=150)

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_lrDA2 <- lrDA(X,label=0,dl=mdl,ini.cov="multRepl",n.iters=150)

# Non-closed compositional data set
data(LPdata) # data (ppm/micrograms per gram)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
LPdata2 <- subset(LPdata,select=-c(Cu,Ni,La)) # select a subset for illustration purposes
dl2 <- dl[-c(5,7,10)]

## Not run: # May take a little while
LPdata_lrDA <- lrDA(LPdata2,label=0,dl=dl2)
## End(Not run)

```

---

lrEM

*Log-ratio EM algorithm*


---

## Description

This function implements model-based ordinary and robust Expectation-Maximisation algorithms to impute left-censored data (e.g. values below detection limit, rounded zeros) via coordinates representation of compositional data which incorporate the information of the relative covariance structure. Alternatively, this function can be used to impute missing data.

## Usage

```

lrEM(X, label = NULL, dl = NULL, rob = FALSE,
     ini.cov = c("complete.obs", "multRepl"), delta = 0.65,
     tolerance = 0.0001, max.iter = 50, rlm.maxit = 150,
     imp.missing = FALSE, suppress.print = FALSE)

```

## Arguments

<code>X</code>	Compositional data set ( <code>matrix</code> or <code>data.frame</code> class).
<code>label</code>	Unique label ( <code>numeric</code> or <code>character</code> ) used to denote unobserved values in <code>X</code> .
<code>d1</code>	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as <code>X</code> .
<code>rob</code>	Logical value. FALSE provides maximum-likelihood estimates of model parameters (default), TRUE provides robust parameter estimates.
<code>ini.cov</code>	Initial estimation of either the log-ratio covariance matrix (ML estimation) or unobserved data (robust estimation). It can be based on either complete observations (" <code>complete.obs</code> ", default) or multiplicative simple replacement of left-censored data (" <code>multRepl</code> ").
<code>delta</code>	If <code>ini.cov="multRepl"</code> , delta parameter for initial multiplicative simple replacement of left-censored data (see <code>multRepl</code> ) expressed in proportions (default = 0.65).
<code>tolerance</code>	Convergence criterion for the EM algorithm (default = 0.0001).
<code>max.iter</code>	Maximum number of iterations for the EM algorithm (default = 50).
<code>r1m.maxit</code>	If <code>rob=TRUE</code> , maximum number of iterations for the embedded robust regression estimation (default = 150; see <code>r1m</code> in MASS package for details).
<code>imp.missing</code>	If TRUE then unobserved data identified by <code>label</code> are treated as missing instead of left-censored data (default = FALSE). Note that <code>ini.cov</code> is fixed to " <code>complete.obs</code> " in this case.
<code>suppress.print</code>	Suppress printed feedback ( <code>suppress.print = FALSE</code> , default).

## Details

After convergence, this function imputes left-censored compositional data by their estimated conditional expected values through coordinates representation, given the information from the observed data and the censoring thresholds. It allows for either single (vector form) or multiple (matrix form, same size as `X`) limits of detection by component. Any threshold value can be set for non-censored elements (e.g. use 0 if no threshold for a particular column or element of the data matrix).

It produces an imputed data set on the same scale as the input data set. If `X` is not closed to a constant sum, then the results are adjusted to provide a compositionally equivalent data set, expressed in the original scale, which leaves the absolute values of the observed components unaltered.

Under maximum likelihood (ML) estimation (default, `rob=FALSE`), a correction factor based on the residual covariance obtained by censored regression is applied for the correct estimation of the conditional covariance matrix in the maximisation step of the EM algorithm. This is required in order to obtain the conditional expectation of the sum of cross-products between two components in the case that both involve imputed values. Note that the procedure is based on the oblique additive log-ratio (alr) transformation to simplify calculations and alleviates computational burden. Nonetheless, the same results would be obtained using an isometric log-ratio transformation (ilr). Note also that alr requires at least one complete column. Otherwise, a preliminary imputation, e.g. by `multRepl` or `multLN`, of the most simplest censoring pattern may be enough. The argument `ini.cov` determines how the initial estimation of the log-ratio covariance matrix required to start the EM process is worked out.

Under robust estimation (`rob=TRUE`), the algorithm requires `ilr` transformations in order to satisfy requirements for robust estimation methods (MM-estimation by default, see `r1m` function for more details). An initial estimation of nondetects is required to get the algorithm started. This can be based on either the subset of fully observed cases (`ini.cov="complete.obs"`) or a multiplicative simple replacement of all nondetects in the data set (`ini.cov="multRepl"`). Note that the robust regression method involved includes random elements which can, occasionally, give rise to `NaN` values getting the routine execution halted. If this happened, we suggest to simply re-run the function once again.

Note that conditional imputation based on log-ratio coordinates cannot be conducted when there exist censoring patterns including samples with only one observed component. As a workaround, `lrEM` applies multiplicative simple replacement (`multRepl`) on those and a warning message identifying the problematic cases is printed. Alternatively, it might be sensible to simply remove those non-informative samples from the data set.

#### *Missing data imputation*

This function can be employed to impute missing data with their conditional expectation using the EM algorithm under regular conditions by setting `imp.missing = TRUE`. Either maximum-likelihood or robust estimation can be used through the `rob` argument. For this case, the argument `label` indicates the unique label for missing values. The argument `d1` is ignored as it is meaningless here and the `"complete.obs"` option is fixed for the `ini.cov` argument.

#### **Value**

A `data.frame` object containing the imputed compositional data set in the same scale as the original. The number of iterations required for convergence is also printed (this can be suppressed by setting `suppress.print=TRUE`).

#### **References**

- Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J. Model-based replacement of rounded zeros in compositional data: classical and robust approaches. *Computational Statistics & Data Analysis* 2012; 56: 2688-2704.
- Palarea-Albaladejo J, Martin-Fernandez JA, Gomez-Garcia J. A parametric approach for dealing with compositional rounded zeros. *Mathematical Geology* 2007; 39: 625-45.
- Palarea-Albaladejo J, Martin-Fernandez JA. A modified EM `alr`-algorithm for replacing rounded zeros in compositional data sets. *Computers & Geosciences* 2008; 34(8): 902-917.
- Palarea-Albaladejo J, Martin-Fernandez JA. Values below detection limit in compositional chemical data. *Analytica Chimica Acta* 2013; 764: 32-43. DOI: <http://dx.doi.org/10.1016/j.aca.2012.12.029>.

#### **See Also**

`zPatterns`, `lrDA`, `multRepl`, `multLN`, `multKM`, `cmultRepl`

#### **Examples**

```
# Data set closed to 100 (percentages, common d1 = 1%)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,26.20,0.00,15.22,6.80,12.05,
```

```

10.76,31.36,7.10,12.74,31.34,6.70,
10.85,46.40,31.89,10.86,0.00,0.00,
7.57,11.35,30.24,6.39,13.65,30.80,
38.09,7.62,23.68,9.70,20.91,0.00,
27.67,7.15,13.05,32.04,6.54,13.55,
44.41,15.04,7.95,0.00,10.82,21.78,
11.50,30.33,6.85,13.92,30.82,6.58,
19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

X_lrEM <- lrEM(X,label=0,dl=rep(1,6),ini.cov="multRepl")
X_rob1rEM <- lrEM(X,label=0,dl=rep(1,6),ini.cov="multRepl",rob=TRUE,tolerance=0.001)

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_lrEM2 <- lrEM(X,label=0,dl=mdl,ini.cov="multRepl")

# Non-closed compositional data set
data(LPdata) # data (ppm/micrograms per gram)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
LPdata2 <- subset(LPdata,select=-c(Cu,Ni,La)) # select a subset for illustration purposes
dl2 <- dl[-c(5,7,10)]

LPdata2_lrEM <- lrEM(LPdata2,label=0,dl=dl2)
LPdata2_rob1rEM <- lrEM(LPdata2,label=0,dl=dl2,rob=TRUE,tolerance=0.005)

# Two subsets of limits of detection (using e.g. robust parameter estimation)
# Using a subset of LPdata for faster execution
data(LPdata) # data (ppm/micrograms per gram)
LPdata2 <- subset(LPdata,select=-c(Cu,Ni,La))
dl2 <- c(2,1,0,0,0,1,0.6,1,0,0,632,10)
# DLs for first 50 samples of LPdata2
dl2a <- matrix(rep(1,50),ncol=1)%*%dl2
# DLs for last 46 samples of LPdata
dl2b <- matrix(rep(1,46),ncol=1)%*%c(1,0.5,0,0,0,0.75,0.3,1,0,0,600,8)

mdl <- rbind(dl2a,dl2b)
LPdata2_rob1rEM <- lrEM(LPdata2,label=0,dl=mdl,rob=TRUE,tolerance=0.005)

# Treating zeros as missing data for illustration purposes only
LPdata2_miss <- lrEM(LPdata2,label=0,imp.missing=TRUE)

```

## Description

This function implements an extended version of the log-ratio EM algorithm (lrEM function) to simultaneously deal with both zeros (i.e. data below detection limit, rounded zeros) and missing data in compositional data sets.

Note: zeros and missing data must be labelled using 0 and NA respectively to use this function.

## Usage

```
lrEMplus(X, dl = NULL, rob = FALSE,
         ini.cov = c("complete.obs", "multRepl"), delta = 0.65,
         tolerance = 0.0001, max.iter = 50, rlm.maxit = 150,
         suppress.print = FALSE)
```

## Arguments

X	Compositional data set ( <a href="#">matrix</a> or <a href="#">data.frame</a> class).
dl	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as X.
rob	Logical value. FALSE provides maximum-likelihood estimates of model parameters (default), TRUE provides robust parameter estimates.
ini.cov	Initial estimation of either the log-ratio covariance matrix (ML estimation) or unobserved data (robust estimation). It can be based on either complete observations ("complete.obs", default) or multiplicative simple replacement of left-censored data ("multRepl").
delta	If ini.cov="multRepl", delta parameter for initial multiplicative simple replacement of left-censored data (see <a href="#">multRepl</a> ) expressed in proportions (default = 0.65).
tolerance	Convergence criterion (default = 0.0001).
max.iter	Maximum number of iterations (default = 50).
rlm.maxit	If rob=TRUE, maximum number of iterations for the embedded robust regression estimation (default = 150; see <a href="#">rlm</a> in MASS package for details).
suppress.print	Suppress printed feedback (suppress.print = FALSE, default).

## Details

The procedure starts with an initial imputation of either zeros (using simple replacement with  $\text{delta} \cdot \text{dl}$ ) or missing values (using mean imputation) depending of which problem is the least frequent in the data set. Subsequently, iterative calls to lrEM replace zeros and missing data alternately until convergence to a stable solution or the maximum number of iterations is reached (see ?lrEM for more details).

## Value

A [data.frame](#) object containing the imputed compositional data set in the same scale as the original. The number of iterations required for convergence is also printed (this can be suppressed by setting `suppress.print=TRUE`).

## References

- Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J. Model-based replacement of rounded zeros in compositional data: classical and robust approaches. *Computational Statistics & Data Analysis* 2012; 56: 2688-2704.
- Palarea-Albaladejo J, Martin-Fernandez JA, Gomez-Garcia J. A parametric approach for dealing with compositional rounded zeros. *Mathematical Geology* 2007; 39: 625-45.
- Palarea-Albaladejo J, Martin-Fernandez JA. A modified EM algorithm for replacing rounded zeros in compositional data sets. *Computers & Geosciences* 2008; 34(8): 902-917.
- Palarea-Albaladejo J, Martin-Fernandez JA. Values below detection limit in compositional chemical data. *Analytica Chimica Acta* 2013; 764: 32-43. DOI: <http://dx.doi.org/10.1016/j.aca.2012.12.029>.

## See Also

[lrEM](#)

## Examples

```
# Data set closed to 100 (percentages, common dl = 1%)
# (Note that zeros and missing in the same row are allowed)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,41.42,0.00,NA,6.80,12.05,
              NA,35.13,7.96,14.28,35.12,7.51,
              10.85,46.40,31.89,10.86,0.00,0.00,
              10.85,16.27,NA,9.16,19.57,44.15,
              38.09,7.62,23.68,9.70,20.91,0.00,
              NA,9.89,18.04,44.30,9.04,18.73,
              44.41,15.04,7.95,0.00,10.82,21.78,
              11.50,30.33,6.85,13.92,30.82,6.58,
              19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

X_lrEMplus <- lrEMplus(X,dl=rep(1,6),ini.cov="multRepl")
X_roblrEMplus <- lrEMplus(X,dl=rep(1,6),ini.cov="multRepl",rob=TRUE,max.iter=4)

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_lrEMplus2 <- lrEMplus(X,dl=mdl,ini.cov="multRepl")

# Non-closed compositional data set
data(LPdataZM) # data (ppm/micrograms per gram; 0 is nondetect and NA is missing data)

dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
LPdataZM2 <- subset(LPdataZM,select=-c(Cu,Ni,La)) # select a subset for illustration purposes
dl2 <- dl[-c(5,7,10)]
```

```
LPdataZM2_lrEMplus <- lrEMplus(LPdataZM2,dl=d12)
```

---

mdl	<i>Water data set: matrix of limits of detection</i>
-----	--

---

### Description

Matrix of varying limits of detection for the [Water](#) data set.

### Usage

```
data(mdl)
```

### Format

A [matrix](#) with 100 rows and 4 columns.

### Details

Three limits of detection (0.75, 1 and 1.25) were considered for Potassium, four for Arsenic (1.5, 3, 4 and 5), two for Sulphate (29 and 35) and no one for Calcium.

### Examples

```
data(Water)  
data(mdl)
```

---

multKM	<i>Multiplicative Kaplan-Meier smoothing spline (KMSS) replacement</i>
--------	--

---

### Description

This function implements non-parametric multiplicative KMSS imputation of left-censored values (e.g. values below detection limit, rounded zeros) in compositional data sets. It is based on simulation from a smoothing spline fitted to the Kaplan-Meier (KM) estimate of the empirical cumulative distribution function (ECDF) of the data.

### Usage

```
multKM(X, label = NULL, dl = NULL, n.draws = 1000, n.knots = NULL)
```

## Arguments

<code>X</code>	Compositional data set ( <code>matrix</code> or <code>data.frame</code> class).
<code>label</code>	Unique label ( <code>numeric</code> or <code>character</code> ) used to denote unobserved left-censored values in <code>X</code> .
<code>d1</code>	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as <code>X</code> .
<code>n.draws</code>	Number of random draws from the inverse KM ECDF generated to produce an averaged imputed value ( <code>n.draws=1000</code> , default).
<code>n.knots</code>	Integer or function giving the number of knots used for fitting a cubic smoothing spline to the KM ECDF (see <code>smooth.spline</code> for default value). It allows for a vector or list of settings per column of <code>X</code> .

## Details

This function imputes left-censored compositional values by averaging (geometric mean) `n` random draws (`n.draws` argument) from a cubic smoothing spline curve fitting the inverse KM ECDF below the corresponding limit of detection or censoring threshold. It then applies a multiplicative adjustment to preserve the multivariate compositional properties of the samples. It allows for either single (vector form) or multiple (`matrix` form, same size as `X`) limits of detection by component. Although note that it is equivalent to simple substitution by the limit of detection for singly censored components. Any threshold value can be set for non-censored elements (e.g. use 0 if no threshold for a particular column or element of the data matrix).

It produces an imputed data set on the same scale as the input data set. If `X` is not closed to a constant sum, then the results are adjusted to provide a compositionally equivalent data set, expressed in the original scale, which leaves the absolute values of the observed components unaltered.

The level of smoothing of the estimated spline can be controlled by the `n.knots` argument. The function `splineKM` can assist in choosing a finer value, although the default setting works generally well.

## Value

A `data.frame` object containing the imputed compositional data set in the same scale as the original.

## References

Palarea-Albaladejo J. and Martin-Fernandez J.A. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. *Chemometrics and Intelligent Laboratory Systems* 2015; 143: 85-96.

## See Also

[zPatterns](#), [splineKM](#), [lrEM](#), [lrDA](#), [multRepl](#), [multLN](#), [cmultRepl](#)



## Examples

```

data(Water)
data(md1) # matrix of limits of detection for Water

Water_multKM <- multKM(Water,label=0,dl=md1)

# Different smoothing degree by component
Water_multKM2 <- multKM(Water,label=0,dl=md1,n.knots=c(25,50,30,75))

# Easy to use for KM multiple imputation (m = 10)
Water.mi <- vector("list",length=10)
for (m in 1:10){
  Water.mi[[m]] <- multKM(Water,label=0,dl=md1,n.draws=1)
}

```

---

multLN

*Multiplicative lognormal replacement*


---

## Description

This function implements model-based multiplicative lognormal imputation of left-censored values (e.g. values below detection limit, rounded zeros) in compositional data sets.

## Usage

```
multLN(X, label = NULL, dl = NULL, rob = FALSE, random = FALSE)
```

## Arguments

X	Compositional data set ( <a href="#">matrix</a> or <a href="#">data.frame</a> class).
label	Unique label ( <a href="#">numeric</a> or <a href="#">character</a> ) used to denote unobserved left-censored values in X.
dl	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as X.
rob	Logical value. FALSE provides maximum-likelihood estimates of model parameters (default), TRUE provides robust estimates (see NADA package for details).
random	Logical value. Values imputed using either estimated geometric mean (FALSE, default) or random values (TRUE) below the limit of detection.

## Details

By default, this function imputes left-censored compositional values by the estimated geometric mean of the values below the corresponding limit of detection or censoring threshold and applies a multiplicative adjustment to preserve the multivariate compositional properties of the samples. Alternatively, imputation can be carried out by random values below the limit of detection (`random = TRUE`) based on a normal distribution on the positive real line (see below).

It depends on package NADA to produce the required model parameter estimates (either maximum likelihood or robust regression on order statistics). It allows for either single (vector form) or multiple (matrix form, same size as X) limits of detection by component. Any threshold value can be set for non-censored elements (e.g. use 0 if no threshold for a particular column or element of the data matrix).

It produces an imputed data set on the same scale as the input data set. If X is not closed to a constant sum, then the results are adjusted to provide a compositionally equivalent data set, expressed in the original scale, which leaves the absolute values of the observed components unaltered. Note that a normal distribution on the positive real line is considered. That is, it is defined with respect to a measure according to own geometry of the positive real line, instead of the standard lognormal based on the Lebesgue measure in real space.

### Value

A `data.frame` object containing the imputed compositional data set in the same scale as the original.

### References

Mateu-Figueras G, Pawlowsky-Glahn V, Egozcue JJ. The normal distribution in some constrained sample spaces. *SORT* 2013; 37(1): 29-56.

Palarea-Albaladejo J, Martin-Fernandez JA. Values below detection limit in compositional chemical data. *Analytica Chimica Acta* 2013; 764: 32-43. DOI: <http://dx.doi.org/10.1016/j.aca.2012.12.029>.

### See Also

[zPatterns](#), [lrEM](#), [lrDA](#), [multRepl](#), [multKM](#), [cmultRepl](#)

### Examples

```
# Data set closed to 100 (percentages, common dl = 1%)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,26.20,0.00,15.22,6.80,12.05,
              10.76,31.36,7.10,12.74,31.34,6.70,
              10.85,46.40,31.89,10.86,0.00,0.00,
              7.57,11.35,30.24,6.39,13.65,30.80,
              38.09,7.62,23.68,9.70,20.91,0.00,
              27.67,7.15,13.05,32.04,6.54,13.55,
              44.41,15.04,7.95,0.00,10.82,21.78,
              11.50,30.33,6.85,13.92,30.82,6.58,
              19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

X_multLN <- multLN(X,label=0,dl=rep(1,6))

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
```

```

mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_multLN2 <- multLN(X,label=0,dl=mdl)

# Non-closed compositional data set
data(LPdata) # data (ppm/micrograms per gram)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)

# Using ML for parameter estimation
LPdata_multLN <- multLN(LPdata,label=0,dl=dl)
# For comparison
LPdata[30:35,1:10]
round(LPdata_multLN[30:35,1:10],1)

# Using ROS for parameter estimation
LPdata_multLNrob <- multLN(LPdata,label=0,dl=dl,rob=TRUE)
round(LPdata_multLNrob[30:35,1:10],1)

# Using random values < dl
LPdata_multRLN <- multLN(LPdata,label=0,dl=dl,random=TRUE)
round(LPdata_multRLN[30:35,1:10],1)

# Two subsets of limits of detection (using e.g. ML parameter estimation)
data(LPdata)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
# DLs for first 50 samples of LPdata
dl1 <- matrix(rep(1,50),ncol=1)%*%dl
# DLs for last 46 samples of LPdata
dl2 <- matrix(rep(1,46),ncol=1)%*%c(1,0.5,0,0,2.5,0,5.5,0.75,0.3,1.5,1,0,0,600,8)

mdl <- rbind(dl1,dl2)
LPdata_multLN2 <- multLN(LPdata,label=0,dl=mdl)

```

---

multRepl

*Multiplicative simple replacement*


---

## Description

This function implements non-parametric multiplicative simple imputation of left-censored values (e.g. values below detection limit, rounded zeros) in compositional data sets.

## Usage

```
multRepl(X, label = NULL, dl = NULL, delta = 0.65)
```

**Arguments**

<code>X</code>	Compositional vector (numeric class) or data set ( <code>matrix</code> or <code>data.frame</code> class).
<code>label</code>	Unique label (numeric or character) used to denote unobserved left-censored values in <code>X</code> .
<code>d1</code>	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as <code>X</code> .
<code>delta</code>	Delta parameter (fraction of the threshold) used in proportions (default = 0.65).

**Details**

This function imputes left-censored compositional values by a given fraction `delta` of the corresponding limit of detection and applies a multiplicative adjustment to preserve the multivariate compositional properties of the samples. It allows for either single (vector form) or multiple (`matrix` form, same size as `X`) limits of detection by component. Any threshold value can be set for non-censored elements (e.g. use 0 if no threshold for a particular column or element of the data matrix).

It produces an imputed data set on the same scale as the input data set. If `X` is not closed to a constant sum, then the results are adjusted to provide a compositionally equivalent data set, expressed in the original scale, which leaves the absolute values of the observed components unaltered. Note that this adjustment only applies to data sets and not when a single composition is entered. In this latter case, the composition is treated as a closed vector.

**Value**

A `data.frame` object containing the imputed compositional vector or data set in the same scale as the original.

**References**

Martin-Fernandez JA, Barcelo-Vidal C, Pawlowsky-Glahn V. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. *Mathematical Geology* 2003; 35: 253-78.

Palarea-Albaladejo J, Martin-Fernandez JA. Values below detection limit in compositional chemical data. *Analytica Chimica Acta* 2013; 764: 32-43. DOI: <http://dx.doi.org/10.1016/j.aca.2012.12.029>.

**See Also**

[zPatterns](#), [lrEM](#), [lrDA](#), [multLN](#), [multKM](#), [cmultRepl](#)

**Examples**

```
# A compositional vector (NA indicates nondetect)
y <- c(0.6, NA, 0.25, 0.03, 0.12, NA)
d1 <- c(0, 0.01, 0, 0, 0, 0.005)
# Using the default delta = 0.65
yr <- multRepl(y, label=NA, d1=d1)
round(yr, 4)
```

```

# Data set closed to 100 (percentages, common dl = 1%)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
             39.73,26.20,0.00,15.22,6.80,12.05,
             10.76,31.36,7.10,12.74,31.34,6.70,
             10.85,46.40,31.89,10.86,0.00,0.00,
             7.57,11.35,30.24,6.39,13.65,30.80,
             38.09,7.62,23.68,9.70,20.91,0.00,
             27.67,7.15,13.05,32.04,6.54,13.55,
             44.41,15.04,7.95,0.00,10.82,21.78,
             11.50,30.33,6.85,13.92,30.82,6.58,
             19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

X_multRepl <- multRepl(X,label=0,dl=rep(1,6))

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_multRepl2 <- multRepl(X,label=0,dl=mdl)

# Non-closed compositional data set
data(LPdata) # data (ppm/micrograms per gram)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
LPdata_multRepl <- multRepl(LPdata,label=0,dl=dl)

# Two subsets of limits of detection
data(LPdata)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
# DLs for first 50 samples of LPdata
dl1 <- matrix(rep(1,50),ncol=1)%*%dl
# DLs for last 46 samples of LPdata
dl2 <- matrix(rep(1,46),ncol=1)%*%c(1,0.5,0,0,2.5,0,5.5,0.75,0.3,1.5,1,0,0,600,8)

mdl <- rbind(dl1,dl2)
LPdata_multRepl2 <- multRepl(LPdata,label=0,dl=mdl)

```

---

Pigs

*Pigs data set*


---

### Description

Count data set consisting of scan sample behavioural observations of a group of 29 sows during a day from 7:30am to 3:30pm, and recorded every 5 minutes (97 times). Six locations were considered: straw bed (BED), half in the straw bed (HALF.BED), dunging passage (PASSAGE), half in the dunging passage (HALF.PASS), feeder (FEEDER) and half in the feeder (HALF.FEED).

**Usage**

```
data(Pigs)
```

**Format**

A `data.frame` with 29 observations on the following 6 variables.

BED a numeric vector

HALF.BED a numeric vector

PASSAGE a numeric vector

HALF.PASS a numeric vector

FEEDER a numeric vector

HALF.FEED a numeric vector

**Source**

Data set kindly provided by the Animal Behaviour and Welfare group at Scotland's Rural College (SRUC), Scotland, UK.

**Examples**

```
data(Pigs)
```

---

splineKM

*Display Kaplan-Meier empirical cumulative distribution function and smoothing spline curve fit*

---

**Description**

This function shows the empirical cumulative distribution function (ECDF) for left-censored data as estimated by the Kaplan-Meier (KM) method and a cubic smoothing spline fitted to it (KMSS method, see [multKM](#)).

**Usage**

```
splineKM(x, label = NULL, dl = NULL, n.knots = NULL,  
         legend.pos = "bottomright",  
         ylab = "ECDF", xlab = "Value",  
         col.km = "black", lty.km = 1, lwd.km = 1,  
         col.sm = "red", lty.sm = 2, lwd.sm = 2, ...)
```

**Arguments**

<code>x</code>	Numerical data vector ( <a href="#">vector</a> class).
<code>label</code>	Unique label ( <a href="#">numeric</a> or <a href="#">character</a> ) used to denote left-censored values in <code>x</code> .
<code>dl</code>	Numeric vector of detection limits/thresholds for each element of <code>x</code> (same length as <code>x</code> ). These must be given on the same scale as <code>x</code> (use e.g. 0 for detected data).
<code>n.knots</code>	Integer or function giving the number of knots used for fitting a cubic smoothing spline to the KM ECDF (see <a href="#">smooth.spline</a> for default value).
<code>legend.pos</code>	Location of the graph legend. Choose one amongst "bottomleft", "bottomright" (default), "topleft" or "topright".
<code>ylab</code>	Title for y-axis.
<code>xlab</code>	Title for x-axis.
<code>col.km</code>	Plotting color for KM ECDF (see base graphical parameters <a href="#">par</a> ).
<code>lty.km</code>	Line type for KM ECDF (see base graphical parameters <a href="#">par</a> ).
<code>lwd.km</code>	Line width for KM ECDF (see base graphical parameters <a href="#">par</a> ).
<code>col.sm</code>	Plotting color for smoothing spline curve.
<code>lty.sm</code>	Line style for smoothing spline curve.
<code>lwd.sm</code>	Line width for smoothing spline curve.
<code>...</code>	Other graphical parameters.

**Value**

Graphical output.

**Examples**

```
data(Water)
data(md1)

# Examine default spline smoothed KM ECDF fit for Potassium and Sulphate
splineKM(Water[,1],label=0,md1[,1])
splineKM(Water[,4],label=0,md1[,4],xlim=c(28,41))

# Reduce to 5 knots for Potassium
splineKM(Water[,1],label=0,md1[,1],n.knots=5)
```

---

Water

*Water data set*

---

**Description**

100 simulated samples of a 4-part groundwater composition in percentage subject to multiple limits of detection by component. The associated matrix of limits of detection is stored in `md1`.

**Usage**

```
data(Water)
```

**Format**

A `data.frame` with 100 observations on the following 4 variables.

Potassium a numeric vector

Arsenic a numeric vector

Calcium a numeric vector

Sulphate a numeric vector

**Details**

Three limits of detection (0.75, 1 and 1.25) were considered for Potassium, four for Arsenic (1.5, 3, 4 and 5), two for Sulphate (29 and 35) and no one for Calcium. In the case of Sulphate, the detection limit equal to 29 is the minimum value registered for that component. All nondetects coded as 0.

**Examples**

```
data(Water)
zPatterns(Water, label=0)
```

---

zCompositions

*Treatment of zeros and nondetects in compositional data sets*

---

**Description**

Following compositional data analysis principles, this package provides simple and friendly tools to explore and impute zeros, left-censored (such as rounded zeros or values below single or multiple limits of detection; a.k.a nondetects) and missing data; including zero pattern/group-wise data analysis and testing procedures.

**Details**

Package: zCompositions  
Type: Package  
Version: 1.2.0  
Date: 2019-02-18  
License: GPL (>= 2)



**Author(s)**

Javier Palarea-Albaladejo and Josep Antoni Martin-Fernandez

Maintainer: Javier Palarea-Albaladejo <javier.palarea@bioss.ac.uk>

**References**

- Martin-Fernandez, J.A., Barcelo-Vidal, C., Pawlowsky-Glahn, V., 2003. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. *Mathematical Geology* 35 (3): 253-27.
- Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J., 2012. Model-based replacement of rounded zeros in compositional data: Classical and robust approaches. *Computational Statistics and Data Analysis* 56: 2688-2704.
- Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J., 2015. Bayesian-multiplicative treatment of count zeros in compositional data sets. *Statistical Modelling* 15 (2): 134-158.
- Palarea-Albaladejo, J., Martin-Fernandez, J.A., Gomez-Garcia, J., 2007. A parametric approach for dealing with compositional rounded zeros. *Mathematical Geology* 39 (7): 625-645.
- Palarea-Albaladejo, J., Martin-Fernandez, J.A., 2008. A modified EM algorithm for replacing rounded zeros in compositional data sets. *Computers & Geosciences* 34 (8): 902-917.
- Palarea-Albaladejo, J., Martin-Fernandez, J.A., 2013. Values below detection limit in compositional chemical data. *Analytica Chimica Acta* 764: 32-43.
- Palarea-Albaladejo, J., Martin-Fernandez, J.A., Olea, R.A., 2014. A bootstrap estimation scheme for chemical compositional data with nondetects. *Journal of Chemometrics* 28: 585-599.
- Palarea-Albaladejo J. and Martin-Fernandez J.A., 2015. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. *Chemometrics and Intelligent Laboratory Systems* 143: 85-96.

**See Also**

- Aitchison, J., 1986. *The Statistical Analysis of Compositional Data*. Monographs on Statistics and Applied Probability. Chapman and Hall Ltd., London, UK (re-edited in 2003 with additional material).
- Filzmoser, P., Hron, K., Templ, M., 2018. *Applied Compositional Data Analysis*. With Worked Examples in R. Springer, Switzerland.
- Pawlowsky-Glahn, V., Buccianti, A. (Eds.), 2011. *Compositional Data Analysis: Theory and Applications*. John Wiley & Sons, Ltd., Chichester, UK.
- Pawlowsky-Glahn, V., Egozcue, J.J., Tolosana-Delgado, R., 2015. *Modeling and analysis of compositional data*. John Wiley & Sons, Ltd., Chichester, UK.
- van den Boogaart, K.G., Tolosana-Delgado, R., 2013. *Analyzing Compositional Data with R*. Springer-Verlag, Berlin, Germany.

**Description**

This function summarises the patterns of zero and/or missing values in a data set and returns a vector of pattern numbers.

**Usage**

```
zPatterns(X, label = NULL, plot = TRUE,
axis.labels = c("Component", "Pattern number"),
bar.colors = c("red3", "red3"), bar.labels = FALSE,
show.means = FALSE, round.means = 2, cex.means = 1,
type.means = c("cgm", "am"),
cell.colors = c("dodgerblue", "white"),
cell.labels = c(label, paste("No", label)), cex.axis = 1.1,
grid.color = "black", grid.lty = "dotted",
legend = TRUE, suppress.print = FALSE, ...)
```

**Arguments**

X	Data set ( <a href="#">matrix</a> or <a href="#">data.frame</a> class).
label	Unique label ( <a href="#">numeric</a> or <a href="#">character</a> ) used to identify zero or unobserved values in X.
plot	Logical value indicating whether a graphical summary of the patterns is produced or not (default plot=TRUE).
axis.labels	Vector of axis labels for the table of patterns (format <code>c("x-axis", "y-axis")</code> ).
bar.colors	Colors for the margin barplots (format <code>c("col.top", "col.right")</code> ).
bar.labels	Logical value indicating if labels showing percentages must be added to the margin barplots (default <code>bar.labels=FALSE</code> ).
show.means	Logical value indicating if mean values by pattern are shown on the graphical summary table (default <code>show.means=FALSE</code> ).
round.means	When <code>show.means=TRUE</code> , number of decimal places for the mean values shown (2=default).
cex.means	When <code>show.means=TRUE</code> , numeric character expansion factor; character size for the mean values shown (1=default).
type.means	When <code>show.means=TRUE</code> , statistic used for computing the means. Either compositional geometric mean ( <code>type.means=cgm</code> , in percentage units, default) or standard arithmetic mean ( <code>type.means=am</code> ).
cell.colors	Vector of colors for the table cells (format <code>c("col.unobserved", "col.observed")</code> ).
cell.labels	Labels for the cells (format <code>c("Unobserved", "Observed")</code> , default <code>c(label, paste("No", label))</code> ).
cex.axis	Axis labels scaling factor relative to default.

<code>grid.color</code>	Color of the grid lines (default "black").
<code>grid.lty</code>	Style of the grid lines (default "dotted", see <code>lty</code> in <a href="#">par</a> ).
<code>legend</code>	Logical value indicating if a legend must be included (default <code>legend=TRUE</code> ).
<code>suppress.print</code>	Suppress printed feedback (default <code>suppress.print=FALSE</code> ).
<code>...</code>	Other graphical parameters.

### Value

Vector of pattern IDs corresponding to each row of  $X$ .

By default, a summary table is printed showing patterns in the data according to `label` and some summary statistics: number of zero/missing components by pattern (`No.Unobs`), pattern frequency (absolute and percentage), percentage zero/missing values by component (column) and overall percentage of zero/missing values in the data set. The symbols + and - indicate, respectively, zero/missing and observed components within each pattern. A graphical version of the summary table is returned including barplots on the margins displaying percentage zero/missing and compositional geometric means by pattern (if `show.means=TRUE`; expressed in percentage scale). Common arithmetic means can be also shown for the case of ordinary data (`type.means="am"`), however this is not recommended for compositional data.

A warning message is shown if zeros or NA values not identified by `label` are present in the data set. These will be ignored for the graphical display and numerical summaries of patterns, which will be only based on `label`.

Note that zeros and missing data can be dealt with simultaneously using the `lrEMplus` function.

### See Also

[lrEM](#), [lrEMplus](#), [lrDA](#), [multRepl](#), [multLN](#), [multKM](#), [cmultRepl](#)

### Examples

```
data(LPdata)

pattern.ID <- zPatterns(LPdata,label=0)

LPdata[pattern.ID==5,]
LPdata[pattern.ID==7,]
LPdata[pattern.ID==10,]

# Modify cell labels and show percentages along with barplots
pattern.ID <- zPatterns(LPdata,label=0,
                        cell.labels=c("Zero","Non-zero"),bar.labels=TRUE)

# Show compositional geometric means (in %) per zero pattern
zPatterns(LPdata,label=0,show.means=TRUE)

# Data sest with zeros and missing data (0 = zero; NA = missing) (see lrEMplus function).
data(LPdataZM)

# Show missingness patterns only
zPatterns(LPdataZM,label=NA)
```

```
# Show zero patterns only and means by pattern based on available data
# (blanks indicate not enough data available for computation)
zPatterns(LPdataZM,label=0,show.means=TRUE)
```

---

zVarArray                      *Variation array for grouped data*

---

## Description

This function returns overall and separate variation arrays for groups in a compositional data set. Groups can be defined by either zero/unobserved data patterns or by a grouping factor in fully observed zero-free data sets.

## Usage

```
zVarArray(X, label = 0, groups = NULL, suppress.print = FALSE)
```

## Arguments

X	Compositional data set ( <a href="#">matrix</a> or <a href="#">data.frame</a> class).
label	Unique label ( <a href="#">numeric</a> or <a href="#">character</a> ) used to denote zero or unobserved data in X (label = 0, default).
groups	Grouping factor in fully observed zero-free data sets (groups = NULL, default).
suppress.print	Suppress printed feedback (suppress.print = FALSE, default).

## Details

This function is mainly aimed to investigate heterogeneous relative variation structures in compositional data sets containing zeros or unobserved values. For each pattern of zero or unobserved values, log-ratio variances (upper triangle of variation matrix) and means (lower triangle of variation matrix) are computed from the available data. Note that (1) NAs are produced for log-ratio variances and means in groups containing less than two observations, and (2) at least two components must be available in each group to compute log-ratios.

The overall estimate is obtained across groups by pairwise deletion. Note that, unlike the ordinary [var](#) function, maximum likelihood estimates of the variances are computed. That is, the observed sum of squares is divided by the corresponding number of observations  $n$  and not by  $n-1$ .

Group-wise variation arrays can be obtained from fully observed zero-free data by setting a grouping factor using the argument `groups`.

## Value

List of variation arrays by pattern/group and overall.

## See Also

[zPatterns](#)

**Examples**

```

data(Water)
zPatterns(Water, label = 0)
zVarArray(Water)

# From a completed data set

data(mdl) # matrix of limits of detection for Water
Water_multKM <- multKM(Water,label=0,dl=mdl) # nondetects imputation

# Results split by two fictitious groups A and B
zVarArray(Water_multKM,groups=rep(c("A","B"),each=50))

```

---

zVarArrayError	<i>Variation array relative error</i>
----------------	---------------------------------------

---

**Description**

This function computes squared relative errors of variation arrays per group with respect to the overall variation array based on observed data in a compositional data set. Groups can be defined by either zero/unobserved data patterns or by a grouping factor in fully observed zero-free data sets.

**Usage**

```
zVarArrayError(X, label = 0, groups = NULL, breakdown = FALSE,
  suppress.print = FALSE)
```

**Arguments**

X	Compositional data set ( <a href="#">matrix</a> or <a href="#">data.frame</a> class).
label	Unique label ( <a href="#">numeric</a> or <a href="#">character</a> ) used to denote zero or unobserved data in X (label = 0, default).
groups	Grouping factor in fully observed zero-free data sets (groups = NULL, default).
breakdown	Logical value. Show results broken down by group (breakdown = FALSE, default).
suppress.print	Suppress printed feedback (suppress.print = FALSE, default).

**Details**

Squared relative errors (SRE) are calculated by confronting variation arrays (log-ratio variances and means) obtained per group and the overall variation array based on observed data. Raw SREs are computed for each available pair-wise log-ratio. The weighted version uses the corresponding group sizes to weight raw SREs. Total SRE is obtained as the sum of weighted SREs for each log-ratio. Further details by group are provided by setting `breakdown = TRUE`.

**Value**

1. SRE for each log-ratio variance and mean. 2. Weighted SRE for each log-ratio variance and mean. 3. Total SRE across log-ratio variances and means. 4. Percentage contribution of each log-ratio to SRE in log-ratio variances and means. If `breakdown = TRUE`: 4. SREs per group. 5. Weighted SREs per group. 6. Percentage contribution of each group to total SRE.

**See Also**

[zPatterns](#), [zVarArray](#)

**Examples**

```
data(Water)
zPatterns(Water, label = 0)
zVarArrayError(Water)
zVarArrayError(Water, breakdown = TRUE)

# From a completed data set

data(mdl) # matrix of limits of detection for Water
Water_multKM <- multKM(Water,label=0,dl=mdl) # nondetects imputation

# Results split by two fictitious groups A and B
zVarArrayError(Water_multKM,groups=rep(c("A","B"),each=50))
```

---

<code>zVarArrayTest</code>	<i>Variation array homogeneity test</i>
----------------------------	---

---

**Description**

This function performs a permutation test of the homogeneity of group-wise and overall variation arrays from all pair-wise log-ratios in a compositional data set. Groups can be defined by either zero/unobserved data patterns or by a grouping factor in fully observed zero-free data sets.

**Usage**

```
zVarArrayTest(X, label = 0, groups = NULL, b = 1000)
```

**Arguments**

<code>X</code>	Compositional data set ( <a href="#">matrix</a> or <a href="#">data.frame</a> class).
<code>label</code>	Unique label ( <a href="#">numeric</a> or <a href="#">character</a> ) used to denote zero or unobserved data in <code>X</code> ( <code>label = 0</code> , default).
<code>groups</code>	Grouping factor in fully observed zero-free data sets ( <code>groups = NULL</code> , default).
<code>b</code>	Number of bootstrap resamples used ( <code>b = 1000</code> , default).

**Details**

The permutation test of homogeneity is based on total weighted squared relative errors (SRE) reflecting on divergence between group-wise variation arrays and overall (see [zVarArrayError](#) and [zVarArray](#) for more details). Note that for groups including less than two observations SRE is set to NA.

**Value**

Test p-values for log-ratio variances and means.

**See Also**

[zPatterns](#), [zVarArray](#), [zVarArrayError](#)

**Examples**

```
data(Water)
zPatterns(Water, label = 0)
zVarArrayTest(Water)
```

# Index

## \*Topic **compositional data**

zCompositions, [24](#)

## \*Topic **datasets**

LPdata, [5](#)

LPdataZM, [6](#)

mdl, [15](#)

Pigs, [21](#)

Water, [23](#)

character, [2](#), [4](#), [7](#), [10](#), [16](#), [17](#), [20](#), [23](#), [26](#),  
[28–30](#)

cmultRepl, [2](#), [8](#), [11](#), [16](#), [18](#), [20](#), [27](#)

data.frame, [2–8](#), [10](#), [11](#), [13](#), [16–18](#), [20](#), [22](#),  
[24](#), [26](#), [28–30](#)

lcTest, [4](#)

list, [8](#)

LPdata, [5](#), [7](#)

LPdataZM, [6](#)

lrDA, [7](#), [11](#), [16](#), [18](#), [20](#), [27](#)

lrEM, [8](#), [9](#), [14](#), [16](#), [18](#), [20](#), [27](#)

lrEMplus, [12](#), [27](#)

matrix, [2](#), [4](#), [7](#), [10](#), [13](#), [15–17](#), [20](#), [26](#), [28–30](#)

mdl, [15](#)

multKM, [8](#), [11](#), [15](#), [18](#), [20](#), [22](#), [27](#)

multLN, [8](#), [10](#), [11](#), [16](#), [17](#), [20](#), [27](#)

multRepl, [3](#), [7](#), [8](#), [10](#), [11](#), [13](#), [16](#), [18](#), [19](#), [27](#)

NaN, [11](#)

numeric, [2](#), [4](#), [7](#), [10](#), [16](#), [17](#), [20](#), [23](#), [26](#), [28–30](#)

par, [23](#), [27](#)

Pigs, [21](#)

rlm, [10](#), [13](#)

smooth.spline, [16](#), [23](#)

splineKM, [16](#), [22](#)

var, [28](#)

vector, [23](#)

Water, [15](#), [23](#)

zCompositions, [24](#)

zPatterns, [3](#), [4](#), [8](#), [11](#), [16](#), [18](#), [20](#), [26](#), [28](#), [30](#),  
[31](#)

zVarArray, [4](#), [28](#), [30](#), [31](#)

zVarArrayError, [4](#), [29](#), [31](#)

zVarArrayTest, [4](#), [30](#)