

# Package ‘VarSelLCM’

January 20, 2025

**Type** Package

**Title** Variable Selection for Model-Based Clustering of Mixed-Type Data Set with Missing Values

**Version** 2.1.3.1

**Date** 2018-08-27

**Author** Matthieu Marbac and Mohammed Sedki

**Maintainer** Mohammed Sedki <mohammed.sedki@u-psud.fr>

**Description** Full model selection (detection of the relevant features and estimation of the number of clusters) for model-based clustering (see reference here <[doi:10.1007/s11222-016-9670-1](https://doi.org/10.1007/s11222-016-9670-1)>). Data to analyze can be continuous, categorical, integer or mixed. Moreover, missing values can occur and do not necessitate any pre-processing. Shiny application permits an easy interpretation of the results.

**License** GPL (>= 2)

**Depends** R (>= 3.3)

**Imports** methods, Rcpp (>= 0.11.1), parallel, mgcv, ggplot2, shiny

**URL** <http://varsellcm.r-forge.r-project.org/>

**LinkingTo** Rcpp, RcppArmadillo

**ByteCompile** true

**LazyLoad** yes

**Collate** 'CheckInputs.R' 'data.R' 'model.R' 'param.R' 'results.R'  
'ICLexact.R' 'DesignOutput.R' 'Summary.R' 'Print.R'  
'VarSelLCM.R' 'RcppExports.R' 'Plot.R' 'Imputation.R'  
'withoutmixture.R' 'VarSelShiny.R' 'ARI.R' 'Extractors.R'  
'Predict.R'

**RoxygenNote** 6.0.1

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2020-10-14 16:34:36 UTC

**Suggests** knitr, rmarkdown, dplyr, htmltools, scales, plyr

**VignetteBuilder** knitr

**BuildVignettes** yes

**Repository/R-Forge/Project** varsellcm

**Repository/R-Forge/Revision** 234

**Repository/R-Forge/DateTimeStamp** 2018-08-27 18:56:49

## Contents

VarSelLCM-package . . . . .	2
AIC . . . . .	5
ARI . . . . .	6
BIC . . . . .	7
coef . . . . .	7
coefficients . . . . .	8
fitted . . . . .	9
fitted.values . . . . .	9
heart . . . . .	10
ICL . . . . .	11
MICL . . . . .	12
plot . . . . .	12
predict . . . . .	14
print . . . . .	14
summary . . . . .	15
VarSelCluster . . . . .	15
VarSelImputation . . . . .	18
VarSelShiny . . . . .	19
VSLCMcriteria-class . . . . .	19
VSLCMdata-class . . . . .	20
VSLCMmodel-class . . . . .	20
VSLCMparam-class . . . . .	21
VSLCMparamCategorical-class . . . . .	21
VSLCMparamContinuous-class . . . . .	21
VSLCMparamInteger-class . . . . .	22
VSLCMpartitions-class . . . . .	22
VSLCMresults-class . . . . .	22
VSLCMstrategy-class . . . . .	23
<b>Index</b>	<b>24</b>

## Description

Model-based clustering with variable selection and estimation of the number of clusters. Data to analyze can be continuous, categorical, integer or mixed. Moreover, missing values can occur and do not necessitate any pre-processing. Shiny application permits an easy interpretation of the results.

## Details

Package:	VarSelLCM
Type:	Package
Version:	2.1.2
Date:	2018-06-04
License:	GPL-3
LazyLoad:	yes
URL:	<a href="http://varsellcm.r-forge.r-project.org/">http://varsellcm.r-forge.r-project.org/</a>

The main function to use is [VarSelCluster](#). Function [VarSelCluster](#) carries out the model selection (according to AIC, BIC or MICL) and maximum likelihood estimation.

Function [VarSelShiny](#) runs a shiny application which permits an easy interpretation of the clustering results.

Function [VarSelImputation](#) permits the imputation of missing values by using the model parameters.

Standard tool methods (e.g., [summary](#), [print](#), [plot](#), [coef](#), [fitted](#), [predict](#)...) are available for facilitating the interpretation.

## Author(s)

Matthieu Marbac and Mohammed Sedki. Maintainer: Mohammed Sedki <mohammed.sedki@u-psud.fr>

## References

Marbac, M. and Sedki, M. (2017). Variable selection for model-based clustering using the integrated completed-data likelihood. *Statistics and Computing*, 27 (4), 1049-1063.

Marbac, M. and Patin, E. and Sedki, M. (2018). Variable selection for mixed data clustering: Application in human population genomics. *Journal of classification*, to appear.

## Examples

```
## Not run:
# Package loading
require(VarSelLCM)

# Data loading:
# x contains the observed variables
# z the known statu (i.e. 1: absence and 2: presence of heart disease)
data(heart)
```

```

ztrue <- heart[,"Class"]
x <- heart[,-13]

# Cluster analysis without variable selection
res_without <- VarSelCluster(x, 2, vbleSelec = FALSE, crit.varsel = "BIC")

# Cluster analysis with variable selection (with parallelisation)
res_with <- VarSelCluster(x, 2, nbcores = 2, initModel=40, crit.varsel = "BIC")

# Comparison of the BIC for both models:
# variable selection permits to improve the BIC
BIC(res_without)
BIC(res_with)

# Comparison of the partition accuracy.
# ARI is computed between the true partition (ztrue) and its estimators
# ARI is an index between 0 (partitions are independent) and 1 (partitions are equals)
# variable selection permits to improve the ARI
# Note that ARI cannot be used for model selection in clustering, because there is no true partition
ARI(ztrue, fitted(res_without))
ARI(ztrue, fitted(res_with))

# Estimated partition
fitted(res_with)

# Estimated probabilities of classification
head(fitted(res_with, type="probability"))

# Summary of the probabilities of missclassification
plot(res_with, type="probs-class")

# Confusion matrices and ARI (only possible because the "true" partition is known).
# ARI is computed between the true partition (ztrue) and its estimators
# ARI is an index between 0 (partitions are independent) and 1 (partitions are equals)
# variable selection permits to improve the ARI
# Note that ARI cannot be used for model selection in clustering, because there is no true partition
# variable selection decreases the misclassification error rate
table(ztrue, fitted(res_without))
table(ztrue, fitted(res_with))
ARI(ztrue, fitted(res_without))
ARI(ztrue, fitted(res_with))

# Summary of the best model
summary(res_with)

# Discriminative power of the variables (here, the most discriminative variable is MaxHeartRate)
plot(res_with)

# More detailed output
print(res_with)

# Print model parameter
coef(res_with)

```

```

# Boxplot for the continuous variable MaxHeartRate
plot(x=res_with, y="MaxHeartRate")

# Empirical and theoretical distributions of the most discriminative variable
# (to check that the distribution is well-fitted)
plot(res_with, y="MaxHeartRate", type="cdf")

# Summary of categorical variable
plot(res_with, y="Sex")

# Probabilities of classification for new observations
predict(res_with, newdata = x[1:3,])

# Imputation by posterior mean for the first observation
not.imputed <- x[1,]
imputed <- VarSelImputation(res_with, x[1,], method = "sampling")
rbind(not.imputed, imputed)

# Opening Shiny application to easily see the results
VarSelShiny(res_with)

## End(Not run)

```

---

AIC

*AIC criterion.*


---

### Description

This function gives the AIC criterion of an instance of [VSLCMresults](#). AIC is computed according to the formula

$$AIC = \log - \text{likelihood} - \nu$$

where  $\nu$  denotes the number of parameters in the fitted model.

### Usage

```
## S4 method for signature 'VSLCMresults'
AIC(object)
```

### Arguments

object            instance of [VSLCMresults](#).

### References

Akaike, H. (1974), "A new look at the statistical model identification", *IEEE Transactions on Automatic Control*, 19 (6): 716-723.

**Examples**

```
# Data loading:
data(heart)

# Cluster analysis without variable selection
res <- VarSelCluster(heart[,-13], 2, vbleSelec = FALSE)

# Get the AIC value
AIC(res)
```

---

ARI

*Adjusted Rand Index*

---

**Description**

This function computes the Adjusted Rand Index

**Usage**

```
ARI(x, y)
```

**Arguments**

x                    vector defining a partition.  
y                    vector defining a partition of whose length is equal to the length of x.

**Value**

numeric

**References**

L. Hubert and P. Arabie (1985) Comparing Partitions, *Journal of the Classification*, 2, pp. 193-218.

**Examples**

```
x <- sample(1:2, 20, replace=TRUE)
y <- x
y[1:5] <- sample(1:2, 5, replace=TRUE)
ARI(x, y)
```

---

BIC	<i>BIC criterion.</i>
-----	-----------------------

---

### Description

This function gives the BIC criterion of an instance of `VSLCMresults`. BIC is computed according to the formula

$$BIC = \log - \text{likelihood} - 0.5 * \nu * \log(n)$$

where  $\nu$  denotes the number of parameters in the fitted model and  $n$  represents the sample size.

### Usage

```
## S4 method for signature 'VSLCMresults'
BIC(object)
```

### Arguments

`object` instance of `VSLCMresults`.

### References

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2), 461-464.

### Examples

```
# Data loading:
data(heart)

# Cluster analysis without variable selection (number of clusters between 1 and 3)
res<- VarSelCluster(heart[,-13], 2, vbleSelec = FALSE)

# Get the BIC value
BIC(res)
```

---

<code>coef</code>	<i>Extract the parameters</i>
-------------------	-------------------------------

---

### Description

This function returns an instance of class `VSLCMparam` which contains the model parameters.

### Usage

```
## S4 method for signature 'VSLCMresults'
coef(object)
```

**Arguments**

object            instance of [VSLCMresults](#).

**Examples**

```
# Data loading:
data(heart)

# Cluster analysis without variable selection (number of clusters between 1 and 3)
res <- VarSelCluster(heart[,-13], 1:3, vbleSelec = FALSE)

# Get the ICL value
coef(res)
```

---

coefficients            *Extract the parameters*

---

**Description**

This function returns an instance of class [VSLCMparam](#) which contains the model parameters.

**Usage**

```
## S4 method for signature 'VSLCMresults'
coefficients(object)
```

**Arguments**

object            instance of [VSLCMresults](#).

**Examples**

```
# Data loading:
data(heart)

# Cluster analysis without variable selection (number of clusters between 1 and 3)
res <- VarSelCluster(heart[,-13], 1:3, vbleSelec = FALSE)

# Get the ICL value
coefficients(res)
```



---

fitted	<i>Extract the partition or the probabilities of classification</i>
--------	---

---

### Description

This function returns the probabilities of classification or the partition among the observations of an instance of [VSLCMresults](#).

### Usage

```
## S4 method for signature 'VSLCMresults'  
fitted(object, type = "partition")
```

### Arguments

object	instance of <a href="#">VSLCMresults</a> .
type	the type of prediction: probability of classification (probability) or the partition (partition)

### Examples

```
# Data loading:  
data(heart)  
  
# Cluster analysis without variable selection (number of clusters between 1 and 3)  
res <- VarSelCluster(heart[, -13], 2, vbleSelec = FALSE)  
  
# Get the ICL value  
fitted(res)
```

---

fitted.values	<i>Extract the partition or the probabilities of classification</i>
---------------	---

---

### Description

This function returns the probabilities of classification or the partition among the observations of an instance of [VSLCMresults](#).

### Usage

```
## S4 method for signature 'VSLCMresults'  
fitted.values(object, type = "partition")
```

**Arguments**

object instance of [VSLCMresults](#).  
 type the type of prediction: probability of classification (probability) or the partition (partition)

**Examples**

```
# Data loading:
data(heart)

# Cluster analysis without variable selection (number of clusters between 1 and 3)
res <- VarSelCluster(heart[,-13], 2, vbleSelec = FALSE)

# Get the ICL value
fitted.values(res)
```

---

heart	<i>Statlog (Heart) Data Set</i>
-------	---------------------------------

---

**Description**

This dataset is a heart disease database similar to a database already present in the repository (Heart Disease databases) but in a slightly different form.

**Details**

12 variables are used to cluster the observations

- age (integer)
- sex (binary)
- chest pain type (categorical with 4 levels)
- resting blood pressure (continuous)
- serum cholestorol in mg/dl (continuous)
- fasting blood sugar > 120 mg/dl (binary)
- resting electrocardiographic results (categorical with 3 levels)
- maximum heart rate achieved (continuous)
- exercise induced angina (binary)
- the slope of the peak exercise ST segment (categorical with 3 levels)
- number of major vessels colored by flourosopy (categorical with 4 levels)
- thal: 3 = normal; 6 = fixed defect; 7 = reversable defect (categorical with 3 levels)

1 variable define a "true" partition: Absence (1) or presence (2) of heart disease

## References

UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science: [http://archive.ics.uci.edu/ml/datasets/statlog+\(heart\)](http://archive.ics.uci.edu/ml/datasets/statlog+(heart))

## Examples

```
data(heart)
```

---

ICL

*ICL criterion*

---

## Description

This function gives the ICL criterion for an instance of [VSLCMresults](#).

## Usage

```
ICL(object)
```

## Arguments

object            [VSLCMresults](#)

## References

Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7), 719-725.

## Examples

```
# Data loading:
data(heart)

# Cluster analysis without variable selection
res <- VarSelCluster(heart[,-13], 2, vbleSelec = FALSE)

# Get the ICL value
ICL(res)
```

---

 MICL

*MICL criterion*


---

### Description

This function gives the MICL criterion for an instance of [VSLCMresults](#).

### Usage

```
MICL(object)
```

### Arguments

```
object          VSLCMresults
```

### References

Marbac, M. and Sedki, M. (2017). Variable selection for model-based clustering using the integrated completed-data likelihood. *Statistics and Computing*, 27 (4), 1049-1063.

### Examples

```
## Not run:
# Data loading:
data("heart")

# Cluster analysis with variable selection
object <- VarSelCluster(heart[,-13], 2, vbleSelec = TRUE, crit.varsel = "MICL")

# Get the MICL value
MICL(object)

## End(Not run)
```

---

 plot

*Plots of an instance of [VSLCMresults](#)*


---

### Description

This function proposes different plots of an instance of [VSLCMresults](#). It permits to visualize:

- the discriminative power of the variables (type="bar" or type="pie"). The larger is the discriminative power of a variable, the more explained are the clusters by this variable.
- the probabilities of misclassification (type="probs-overall" or type="probs-class").
- the distribution of a single variable (y is the name of the variable and type="boxplot" or type="cdf").

**Usage**

```
## S4 method for signature 'VSLCMresults,character'
plot(x, y, type = "boxplot", ylim = c(1,
  x@data@d))
```

**Arguments**

x	instance of <a href="#">VSLCMresults</a> .
y	character. The name of the variable to plotted (only used if type="boxplot" or type="cdf").
type	character. The type of plot ("bar": barplot of the discriminative power, "pie": pie of the discriminative power, "probs-overall": histogram of the probabilities of misclassification, "probs-class": histogram of the probabilities of misclassification per cluster, "boxplot": boxplot of a single variable per cluster, "cdf": distribution of a single variable per cluster).
ylim	numeric. Define the range of the most discriminative variables to considered (only use if type="pie" or type="bar")

**Examples**

```
## Not run:
require(VarSelLCM)

# Data loading:
# x contains the observed variables
# z the known statu (i.e. 1: absence and 2: presence of heart disease)
data(heart)
ztrue <- heart[, "Class"]
x <- heart[, -13]

# Cluster analysis with variable selection (with parallelisation)
res_with <- VarSelCluster(x, 2, nbcores = 2, initModel=40)

# Summary of the probabilities of missclassification
plot(res_with, type="probs-class")

# Discriminative power of the variables (here, the most discriminative variable is MaxHeartRate)
plot(res_with)

# Boxplot for the continuous variable MaxHeartRate
plot(res_with, y="MaxHeartRate")

# Empirical and theoretical distributions (to check that the distribution is well-fitted)
plot(res_with, y="MaxHeartRate", type="cdf")

# Summary of categorical variable
plot(res_with, y="Sex")

## End(Not run)
```

---

predict	<i>Prediction of the cluster memberships</i>
---------	--

---

**Description**

This function gives the probabilities of classification for new observations by using the mixture model fit with the function [VarSelCluster](#).

**Usage**

```
## S4 method for signature 'VSLCMresults'  
predict(object, newdata, type = "probability")
```

**Arguments**

object	instance of <a href="#">VSLCMresults</a> .
newdata	data.frame of the observations to classify.
type	the type of prediction: probability of classification (probability) or the partition (partition)

**Value**

Returns a matrix of the probabilities of classification.

---

print	<i>Print function.</i>
-------	------------------------

---

**Description**

This function gives the print of an instance of [VSLCMresults](#).

**Usage**

```
## S4 method for signature 'VSLCMresults'  
print(x)
```

**Arguments**

x	instance of <a href="#">VSLCMresults</a> .
---	--

---

summary	<i>Summary function.</i>
---------	--------------------------

---

### Description

This function gives the summary of an instance of [VSLCMresults](#).

### Usage

```
## S4 method for signature 'VSLCMresults'
summary(object)
```

### Arguments

object	instance of <a href="#">VSLCMresults</a> .
--------	--

---

VarSelCluster	<i>Variable selection and clustering.</i>
---------------	---

---

### Description

This function performs the model selection and the maximum likelihood estimation. It can be used for clustering only (i.e., all the variables are assumed to be discriminative). In this case, you must specify the data to cluster (arg. `x`), the number of clusters (arg. `g`) and the option `vbleSelec` must be `FALSE`. This function can also be used for variable selection in clustering. In this case, you must specify the data to analyse (arg. `x`), the number of clusters (arg. `g`) and the option `vbleSelec` must be `TRUE`. Variable selection can be done with BIC, MICL or AIC.

### Usage

```
VarSelCluster(x, gvals, vbleSelec = TRUE, crit.varsel = "BIC",
  initModel = 50, nbcores = 1, discrim = rep(1, ncol(x)), nbSmall = 250,
  iterSmall = 20, nbKeep = 50, iterKeep = 1000, tolKeep = 10^(-6))
```

### Arguments

<code>x</code>	data.frame/matrix. Rows correspond to observations and columns correspond to variables. Continuous variables must be "numeric", count variables must be "integer" and categorical variables must be "factor"
<code>gvals</code>	numeric. It defines number of components to consider.
<code>vbleSelec</code>	logical. It indicates if a variable selection is done
<code>crit.varsel</code>	character. It defines the information criterion used for model selection. Without variable selection, you can use one of the three criteria: "AIC", "BIC" and "ICL". With variable selection, you can use "AIC", "BIC" and "MICL".

<code>initModel</code>	numeric. It gives the number of initializations of the alternated algorithm maximizing the MICL criterion (only used if <code>crit.varsel="MICL"</code> )
<code>nbcores</code>	numeric. It defines the number of cores used by the algorithm
<code>discrim</code>	numeric. It indicates if each variable is discriminative (1) or irrelevant (0) (only used if <code>vbleSelec=0</code> )
<code>nbSmall</code>	numeric. It indicates the number of SmallEM algorithms performed for the ML inference
<code>iterSmall</code>	numeric. It indicates the number of iterations for each SmallEM algorithm
<code>nbKeep</code>	numeric. It indicates the number of chains used for the final EM algorithm
<code>iterKeep</code>	numeric. It indicates the maximal number of iterations for each EM algorithm
<code>tolKeep</code>	numeric. It indicates the maximal gap between two successive iterations of EM algorithm which stops the algorithm

### Value

Returns an instance of [VSLCMresults](#).

### References

Marbac, M. and Sedki, M. (2017). Variable selection for model-based clustering using the integrated completed-data likelihood. *Statistics and Computing*, 27 (4), 1049-1063.

Marbac, M. and Patin, E. and Sedki, M. (2018). Variable selection for mixed data clustering: Application in human population genomics. *Journal of Classification*, to appear.

### Examples

```
## Not run:
# Package loading
require(VarSelLCM)

# Data loading:
# x contains the observed variables
# z the known status (i.e. 1: absence and 2: presence of heart disease)
data(heart)
ztrue <- heart[,"Class"]
x <- heart[,-13]

# Cluster analysis without variable selection
res_without <- VarSelCluster(x, 2, vbleSelec = FALSE, crit.varsel = "BIC")

# Cluster analysis with variable selection (with parallelisation)
res_with <- VarSelCluster(x, 2, nbcores = 2, initModel=40, crit.varsel = "BIC")

# Comparison of the BIC for both models:
# variable selection permits to improve the BIC
BIC(res_without)
BIC(res_with)

# Confusion matrices and ARI (only possible because the "true" partition is known).
```



```
# ARI is computed between the true partition (ztrue) and its estimators
# ARI is an index between 0 (partitions are independent) and 1 (partitions are equals)
# variable selection permits to improve the ARI
# Note that ARI cannot be used for model selection in clustering, because there is no true partition
# variable selection decreases the misclassification error rate
table(ztrue, fitted(res_without))
table(ztrue, fitted(res_with))
ARI(ztrue, fitted(res_without))
ARI(ztrue, fitted(res_with))

# Estimated partition
fitted(res_with)

# Estimated probabilities of classification
head(fitted(res_with, type="probability"))

# Summary of the probabilities of missclassification
plot(res_with, type="probs-class")

# Summary of the best model
summary(res_with)

# Discriminative power of the variables (here, the most discriminative variable is MaxHeartRate)
plot(res_with)

# More detailed output
print(res_with)

# Print model parameter
coef(res_with)

# Boxplot for the continuous variable MaxHeartRate
plot(x=res_with, y="MaxHeartRate")

# Empirical and theoretical distributions of the most discriminative variable
# (to check that the distribution is well-fitted)
plot(res_with, y="MaxHeartRate", type="cdf")

# Summary of categorical variable
plot(res_with, y="Sex")

# Probabilities of classification for new observations
predict(res_with, newdata = x[1:3,])

# Imputation by posterior mean for the first observation
not.imputed <- x[1,]
imputed <- VarSelImputation(res_with, x[1,], method = "sampling")
rbind(not.imputed, imputed)

# Opening Shiny application to easily see the results
VarSelShiny(res_with)
```

```
## End(Not run)
```

---

VarSelImputation      *Imputation of missing values*

---

## Description

This function permits imputation of missing values in a dataset by using mixture model. Two methods can be used for imputation:

- posterior mean (method="postmean")
- sampling from the full conditionnal distribution (method="sampling")

## Usage

```
VarSelImputation(obj, newdata, method = "postmean")
```

## Arguments

**obj**                    an instance of [VSLCMresults](#) which defines the model used for imputation.  
**newdata**                data.frame Dataset containing the missing values to impute.  
**method**                character defining the method of imputation: "postmean" or "sampling"

## Examples

```
# Data loading
data("heart")

# Clustering en 2 classes
results <- VarSelCluster(heart[, -13], 2)

# Data where missing values will be imputed
newdata <- heart[1:2, -13]
newdata[1,1] <- NA
newdata[2,2] <- NA

# Imputation
VarSelImputation(results, newdata)
```

---

 VarSelShiny

*Shiny app for analyzing results from VarSelCluster*


---

**Description**

Shiny app for analyzing results from VarSelCluster

**Usage**

```
VarSelShiny(X)
```

**Arguments**

**X** an instance of [VSLCMresults](#) returned by function [VarSelCluster](#).

**Examples**

```
## Not run:
# Data loading
data("heart")
# Clustering en 2 classes
results <- VarSelCluster(heart[,-13], 2)
# Opening Shiny application to easily see the results
VarSelShiny(results)

## End(Not run)
```

---

 VSLCMcriteria-class

*Constructor of [VSLCMcriteria](#) class*


---

**Description**

**loglikelihood** numeric. Log-likelihood

**AIC** numeric. Value of the AIC criterion.

**BIC** numeric. Value of the BIC criterion.

**ICL** numeric. Value of the ICL criterion.

**MICL** numeric. Value of the MICL criterion.

**nbparam** integer. Number of parameters.

**cvrate** numeric. Rate of convergence of the alternated algorithm for optimizing the MICL criterion.

**degeneracyrate** numeric. Rate of degeneracy for the selected model.

**discrim** numeric. Discriminative power of each variable.

**Examples**

```
getSlots("VSLCMcriteria")
```

---

VSLCMdata-class      *Constructor of [VSLCMdata](#) class*

---

### Description

**n** number of observations

**d** number of variables

**withContinuous** logical indicating if some variables are continuous

**withInteger** logical indicating if some variables are integer

**withCategorical** logical indicating if some variables are categorical

**dataContinuous** instance of VSLCMdataContinuous containing the continuous data

**dataInteger** instance of VSLCMdataContinuous containing the integer data

**dataCategorical** instance of VSLCMdataContinuous containing the categorical data

**var.names** labels of the variables

### Examples

```
getSlots("VSLCMdata")
```

---

VSLCMmodel-class      *Constructor of [VSLCMmodel](#) class*

---

### Description

**g** numeric. Number of components.

**omega** logical. Vector indicating if each variable is irrelevant (1) or not (0) to the clustering.

**names.relevant** character. Names of the relevant variables.

**names.irrelevant** character. Names of the irrelevant variables.

### Examples

```
getSlots("VSLCMmodel")
```

---

VSLCMparam-class      *Constructor of VSLCMparam class*

---

### Description

**pi** numeric. Proportions of the mixture components.

**paramContinuous** [VSLCMparamContinuous](#). Parameters of the continuous variables.

**paramInteger** [VSLCMparamInteger](#). Parameters of the integer variables.

**paramCategorical** [VSLCMparamCategorical](#). Parameters of the categorical variables.

### Examples

```
getSlots("VSLCMparam")
```

---

VSLCMparamCategorical-class      *Constructor of VSLCMparamCategorical class*

---

### Description

**pi** numeric. Proportions of the mixture components.

**alpha** list. Parameters of the multinomial distributions.

### Examples

```
getSlots("VSLCMparamCategorical")
```

---

VSLCMparamContinuous-class      *Constructor of VSLCMparamContinuous class*

---

### Description

**pi** numeric. Proportions of the mixture components.

**mu** matrix. Mean for each component (column) and each variable (row).

**sd** matrix. Standard deviation for each component (column) and each variable (row).

### Examples

```
getSlots("VSLCMparamContinuous")
```

---

VSLCMparamInteger-class

*Constructor of [VSLCMparamInteger](#) class*

---

### Description

**pi** numeric. Proportions of the mixture components.

**lambda** matrix. Mean for each component (column) and each variable (row).

### Examples

```
getSlots("VSLCMparamInteger")
```

---

VSLCMpartitions-class *Constructor of [VSLCMpartitions](#) class*

---

### Description

**zMAP** numeric. A vector indicating the class membership of each individual by using the MAP rule computed for the best model with its maximum likelihood estimates.

**zOPT** numeric. Partition maximizing the integrated complete-data likelihood of the selected model.

**tik** numeric. Fuzzy partition computed for the best model with its maximum likelihood estimates.

### Examples

```
getSlots("VSLCMpartitions")
```

---

VSLCMresults-class *Constructor of [VSLCMresults](#) class*

---

### Description

**data** [VSLCMdata](#). Results relied to the data.

**criteria** [VSLCMcriteria](#). Results relied to the information criteria.

**partitions** [VSLCMpartitions](#). Results relied to the partitions.

**model** [VSLCMmodel](#). Results relied to the selected model.

**strategy** [VSLCMstrategy](#). Results relied to the tune parameters.

**param** [VSLCMparam](#). Results relied to the parameters.

### Examples

```
getSlots("VSLCMresults")
```

---

VSLCMstrategy-class    *Constructor of VSLCMstrategy class*

---

### Description

**initModel** numeric. Number of initialisations for the model selection algorithm.

**vbleSelec** logical. It indicates if the selection of the variables is performed.

**paramEstim** logical. It indicates if the parameter estimation is performed.

**parallel** logical. It indicates if a parallelisation is done.

**nbSmall** numeric. It indicates the number of small EM.

**iterSmall** numeric. It indicates the number of iteration for the small EM

**nbKeep** numeric. It indicates the number of chains kept for the EM.

**iterKeep** numeric. It indicates the maximum number of iteration for the EM.

**tolKeep** numeric. It indicates the value of the difference between successive iterations of EM stopping the EM.

### Examples

```
getSlots("VSLCMstrategy")
```

# Index

- \* **datasets**
  - heart, [10](#)
- \* **package**
  - VarSelLCM-package, [2](#)
- AIC, [5](#)
- AIC, VSLCMresults-method (AIC), [5](#)
- ARI, [6](#)
- BIC, [7](#)
- BIC, VSLCMresults-method (BIC), [7](#)
- coef, [3, 7](#)
- coef, VSLCMresults-method (coef), [7](#)
- coefficients, [8](#)
- coefficients, VSLCMresults-method (coefficients), [8](#)
- fitted, [3, 9](#)
- fitted, VSLCMresults-method (fitted), [9](#)
- fitted.values, [9](#)
- fitted.values, VSLCMresults-method (fitted.values), [9](#)
- heart, [10](#)
- ICL, [11](#)
- MICL, [12](#)
- plot, [3, 12](#)
- plot, VSLCMresults, ANY-method (plot), [12](#)
- plot, VSLCMresults, character-method (plot), [12](#)
- plot, VSLCMresults-method (plot), [12](#)
- predict, [3, 14](#)
- predict, VSLCMresults-method (predict), [14](#)
- print, [3, 14](#)
- print, VSLCMresults-method (print), [14](#)
- summary, [3, 15](#)
- summary, VSLCMresults-method (summary), [15](#)
- VarSelCluster, [3, 14, 15, 19](#)
- VarSelImputation, [3, 18](#)
- VarSelLCM (VarSelLCM-package), [2](#)
- VarSelLCM-package, [2](#)
- VarSelShiny, [3, 19](#)
- VSLCMcriteria, [19, 22](#)
- VSLCMcriteria-class, [19](#)
- VSLCMdata, [20, 22](#)
- VSLCMdata-class, [20](#)
- VSLCMmodel, [20, 22](#)
- VSLCMmodel-class, [20](#)
- VSLCMparam, [7, 8, 21, 22](#)
- VSLCMparam-class, [21](#)
- VSLCMparamCategorical, [21](#)
- VSLCMparamCategorical-class, [21](#)
- VSLCMparamContinuous, [21](#)
- VSLCMparamContinuous-class, [21](#)
- VSLCMparamInteger, [21, 22](#)
- VSLCMparamInteger-class, [22](#)
- VSLCMpartitions, [22](#)
- VSLCMpartitions-class, [22](#)
- VSLCMresults, [5, 7–16, 18, 19, 22](#)
- VSLCMresults-class, [22](#)
- VSLCMstrategy, [22, 23](#)
- VSLCMstrategy-class, [23](#)