

Package ‘ICSclust’

January 20, 2025

Type Package

Title Tandem Clustering with Invariant Coordinate Selection

Version 0.1.0

Date 2023-09-20

Description Implementation of tandem clustering with invariant coordinate selection with different scatter matrices and several choices for the selection of components as described in Alfons, A., Archimbaud, A., Nordhausen, K. and Ruiz-Gazen, A. (2022) <[arXiv:2212.06108](https://arxiv.org/abs/2212.06108)>.

License GPL (>= 3)

Encoding UTF-8

Depends ICS (>= 1.4-0), ggplot2

Imports cluster, fpc, GGally, heplots, mclust, moments, mvtnorm, otrimle, RcppRoll, rrcov, scales, tclust

LinkingTo Rcpp, RcppArmadillo

Suggests testthat (>= 3.0.0)

URL <https://github.com/AuroreAA/ICSclust>

BugReports <https://github.com/AuroreAA/ICSclust/issues>

Author Aurore Archimbaud [aut, cre] (<<https://orcid.org/0000-0002-6511-9091>>),
Andreas Alfons [aut] (<<https://orcid.org/0000-0002-2513-3788>>),
Klaus Nordhausen [aut] (<<https://orcid.org/0000-0002-3758-8501>>),
Anne Ruiz-Gazen [aut] (<<https://orcid.org/0000-0001-8970-8061>>)

Maintainer Aurore Archimbaud <aurore.archimbaud@live.fr>

RoxygenNote 7.2.3

Config/testthat/edition 3

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-09-21 13:20:02 UTC

Contents

ICSClust-package	2
component_plot	4
discriminatory_crit	5
ICSClust	7
ICS_lcov	9
ICS_mcd	10
ICS_mlc	11
ICS_tcov	12
ICS_ucov	13
kmeans_clust	14
mclust_clust	15
med_crit	16
mixture_sim	17
normal_crit	18
pam_clust	20
plot.ICSClust	21
print.ICSClust_summary	22
rimle_clust	23
runif_outside_range	24
select_plot	25
summary.ICSClust	27
tcov	28
tkmeans_clust	29
ucov	30
var_crit	32
Index	34

ICSClust-package	<i>Tandem Clustering with Invariant Coordinate Selection</i>
------------------	--

Description

Implementation of tandem clustering with invariant coordinate selection with different scatter matrices and several choices for the selection of components as described in Alfons, A., Archimbaud, A., Nordhausen, K. and Ruiz-Gazen, A. (2022) <arXiv:2212.06108>.

Details

The DESCRIPTION file:

Package:	ICSClust
Type:	Package
Title:	Tandem Clustering with Invariant Coordinate Selection
Version:	0.1.0
Date:	2023-09-20

Description: Implementation of tandem clustering with invariant coordinate selection with different scatter matrices
 License: GPL (≥ 3)
 Encoding: UTF-8
 Depends: ICS ($\geq 1.4-0$), ggplot2
 Imports: cluster, fpc, GGally, heplots, mclust, moments, mvtnorm, otrimle, RcppRoll, rrcov, scales, tclust
 LinkingTo: Rcpp, RcppArmadillo
 Suggests: testthat ($\geq 3.0.0$)
 URL: <https://github.com/AuroreAA/ICSClust>
 BugReports: <https://github.com/AuroreAA/ICSClust/issues>
 Authors@R: c(person("Aurore", "Archimbaud", email = "aurore.archimbaud@live.fr", role = c("aut", "cre")), com
 Author: Aurore Archimbaud [aut, cre] (<<https://orcid.org/0000-0002-6511-9091>>), Andreas Alfons [aut] (<h
 Maintainer: Aurore Archimbaud <aurore.archimbaud@live.fr>
 Roxygen: list(markdown = TRUE)
 RoxygenNote: 7.2.3
 Config/testthat/edition: 3
 Archs: x64

Index of help topics:

ICSClust	Tandem clustering with ICS
ICSClust-package	Tandem Clustering with Invariant Coordinate Selection
ICS_lcov	Local Shape Scatter Estimates for ICS
ICS_mcd	MCD location and Scatter Estimates for ICS
ICS_mlc	Cauchy location and Scatter Estimates for ICS
ICS_tcov	Pairwise one-step M-estimate of scatter for ICS
ICS_ucov	Simple robust estimates of scatter for ICS
component_plot	Scatterplot Matrix with densities on the diagonal
discriminatory_crit	Selection of ICS components based on discriminatory power
kmeans_clust	_k_-means clustering
mclust_clust	Model-Based Clustering
med_crit	Selection of Invariant components using the med criterion
mixture_sim	Simulation of a mixture of Gaussian distributions
normal_crit	Selection of Non-normal Invariant Components Using Marginal Normality Tests
pam_clust	Partitioning Around Medoids clustering
plot.ICSClust	Scatterplot Matrix with densities on the diagonal
print.ICSClust_summary	Print of an 'ICSClust_summary' object
rimle_clust	Robust Improper Maximum Likelihood Clustering
runif_outside_range	Uniform distribution outside a given range
select_plot	Plot of the Generalized Kurtosis Values of the

	ICS Transformation
summary.ICSClust	Summary of an 'ICSClust' object
tcov	Pairwise one-step M-estimate of scatter
tkmeans_clust	Trimmed k-means clustering
ucov	Simple robust estimates of scatter
var_crit	Selection of Invariant components using the var criterion

Author(s)

Aurore Archimbaud [aut, cre] (<<https://orcid.org/0000-0002-6511-9091>>), Andreas Alfons [aut] (<<https://orcid.org/0000-0002-2513-3788>>), Klaus Nordhausen [aut] (<<https://orcid.org/0000-0002-3758-8501>>), Anne Ruiz-Gazen [aut] (<<https://orcid.org/0000-0001-8970-8061>>)

Maintainer: Aurore Archimbaud <aurore.archimbaud@live.fr>

References

Alfons, A., Archimbaud, A., Nordhausen, K., & Ruiz-Gazen, A. (2022). Tandem clustering with invariant coordinate selection. *arXiv preprint arXiv:2212.06108*.

component_plot	<i>Scatterplot Matrix with densities on the diagonal</i>
----------------	--

Description

Produces a gg-scatterplot matrix of the variables of a given dataframe or an invariant coordinate system obtained via an ICS transformation with densities on the diagonal for each cluster.

Usage

```
component_plot(
  object,
  select = TRUE,
  clusters = NULL,
  text_size_factor = 8/6.5,
  colors = NULL
)
```

Arguments

object	a dataframe or ICS class object.
select	a vector of indexes of variables to plot. If NULL or FALSE, all variables are selected. If TRUE only the first three and last three are considered.
clusters	a vector indicating the clusters of the data to color the plot. By default NULL.
text_size_factor	a numeric factor for controlling the axis.text and strip.text.
colors	a vector of colors to use. One color for each cluster.

Value

An object of class "ggmatrix" (see `GGally::ggpairs()`).

Author(s)

Andreas Alfons and Aurore Archimbaud

Examples

```
X <- iris[,1:4]
component_plot(X)
out <- ICS(X)
component_plot(out, select = c(1,4))
```

discriminatory_crit *Selection of ICS components based on discriminatory power*

Description

Identifies invariant coordinates associated to the highest discriminatory power (by default "eta2").

Usage

```
discriminatory_crit(object, ...)
```

```
## S3 method for class 'ICS'
```

```
discriminatory_crit(
  object,
  clusters,
  method = "eta2",
  nb_select = NULL,
  select_only = FALSE,
  ...
)
```

```
## Default S3 method:
```

```
discriminatory_crit(
  object,
  clusters,
  method = "eta2",
  nb_select = NULL,
  select_only = FALSE,
  gen_kurtosis = NULL,
  ...
)
```

Arguments

object	dataframe or object of class "ICS".
...	additional arguments are currently ignored.
clusters	a vector of the same length as the number of observations, indicating the true clusters. It is used to compute the discriminatory power based on it.
method	the name of the discriminatory power. Only "eta2" is implemented.
nb_select	the exact number of components to select. By default it is set to NULL, i.e the number of components to select is the number of clusters minus one.
select_only	boolean. If TRUE only the vector names of the selected invariant components are returned. If FALSE additional details are returned.
gen_kurtosis	vector of generalized kurtosis values.

Details

The discriminatory power $\eta^2 = 1 - \Lambda$, where Λ denotes Wilks' lambda, is evaluated for each combination of the first and/or last combinations of nb_select components. The combination achieving the highest discriminatory power is selected.

More specifically, we compute

$$\eta^2 = 1 - \frac{\det(E)}{\det(T)},$$

where E is the within-group sum of squares and cross-products matrix and T is the total sum of squares and cross-products matrix.

Value

If select_only is TRUE a vector of the names of the invariant components or variables to select. If FALSE an object of class "ICS_crit" is returned with the following objects:

- crit: the name of the criterion "discriminatory".
- method: the name of the discriminatory power.
- nb_select: the number of components to select.
- select: the names of the invariant components or variables to select.
- power_combinations: the discriminatory values for each of the considered combinations of nb_select components.
- gen_kurtosis: the vector of generalized kurtosis values in case of ICS object.

Author(s)

Aurore Archimbaud and Anne Ruiz-Gazen

References

Alfons, A., Archimbaud, A., Nordhausen, K., & Ruiz-Gazen, A. (2022). Tandem clustering with invariant coordinate selection. *arXiv preprint arXiv:2212.06108*..

See Also

[normal_crit\(\)](#), [med_crit\(\)](#), [var_crit\(\)](#).

Examples

```
X <- iris[,-5]
out <- ICS(X)
discriminatory_crit(out, clusters = iris[,5], select_only = FALSE)
```

 ICSClust

Tandem clustering with ICS

Description

Sequential clustering approach: (i) dimension reduction through the Invariant Coordinate Selection method using the [ICS](#) function and (ii) clustering of the transformed data.

Usage

```
ICSClust(
  X,
  nb_select = NULL,
  nb_clusters = NULL,
  ICS_args = list(),
  criterion = c("med_crit", "normal_crit", "var_crit", "discriminatory_crit"),
  ICS_crit_args = list(),
  method = c("kmeans_clust", "tkmeans_clust", "pam_clust", "mclust_clust",
    "rmclust_clust", "rimle_clust"),
  clustering_args = list(),
  clusters = NULL
)
```

Arguments

<code>X</code>	a numeric matrix or data frame containing the data.
<code>nb_select</code>	the number of components to select. It is used only in case criterion is either "med_crit", "var_crit" or "discriminatory_crit". By default it is set to NULL, i.e the number of components to select is the number of clusters minus one.
<code>nb_clusters</code>	the number of clusters searched for.
<code>ICS_args</code>	list of ICS-S3 arguments. Otherwise, default values of ICS-S3 are used.
<code>criterion</code>	criterion to automatically decide which invariant components to keep. Possible values are "med_crit", "normal_crit", "var_crit" and "discriminatory_crit". The default value is "med_crit". See med_crit() , normal_crit() , var_crit() or discriminatory_crit() for more details.

ICS_crit_args	list of arguments passed to <code>med_crit()</code> , <code>normal_crit()</code> , <code>var_crit()</code> or <code>discriminatory_crit()</code> for choosing the components to keep.
method	clustering method to perform. Currently implemented wrapper functions are "kmeans_clust", "tkmeans_clust", "pam_clust", "mclust_clust", "rmclust_clust" or "rimle_clust". The default value is "kmeans_clust".
clustering_args	list of <code>kmeans_clust()</code> , <code>tkmeans_clust()</code> , <code>pam_clust()</code> , <code>rimle_clust()</code> , <code>mclust_clust()</code> or <code>rmclust_clust()</code> arguments for performing cluster analysis.
clusters	a vector indicating the true clusters of the data. By default, it is NULL but it is required to choose the components based on the discriminatory criterion <code>discriminatory_crit</code> .

Details

Tandem clustering with ICS is a sequential method:

- ICS is performed.
- only a subset of the first and/or the last few components are selected based on a criterion.
- the clustering method is performed only on the subspace of the selected components.
- wrapper for several different clustering methods are provided. Users can however also write wrappers for other clustering methods.

Value

An object of class "ICSclust" with the following components:

- ICS_out: An object of class "ICS". See [ICS](#)
- select: a vector of the names of the selected invariant coordinates.
- clusters: a vector of the new partition of the data, i.e a vector of integers (from 1:k) indicating the cluster to which each observation is allocated. 0 indicates outlying observations.

`summary()` and `plot()` methods are available.

Author(s)

Aurore Archimbaud

References

Alfons, A., Archimbaud, A., Nordhausen, K., & Ruiz-Gazen, A. (2022). Tandem clustering with invariant coordinate selection. *arXiv preprint arXiv:2212.06108*.

See Also

`med_crit()`, `normal_crit()`, `var_crit()`, `ICS`, `discriminatory_crit()`, `kmeans_clust()`, `tkmeans_clust()`, `pam_clust()`, `rimle_clust()`, `mclust_clust()` `summary()` and `plot()` methods

Examples

```

X <- iris[,1:4]

# indicating the number of components to retain for the dimension reduction
# step as well as the number of clusters searched for.
out <- ICSClust(X, nb_select = 2, nb_clusters = 3)
summary(out)
plot(out)

# changing the scatter pair to consider in ICS
out <- ICSClust(X, nb_select = 1, nb_clusters = 3,
ICS_args = list(S1 = ICS_mcd_raw, S2 = ICS_cov, S1_args = list(alpha = 0.5)))
summary(out)
plot(out)

# changing the criterion for choosing the invariant coordinates
out <- ICSClust(X, nb_clusters = 3, criterion = "normal_crit",
ICS_crit_args = list(level = 0.1, test = "anscombe.test", max_select = NULL))
summary(out)
plot(out)

# changing the clustering method
out <- ICSClust(X, nb_clusters = 3, method = "tkmeans_clust",
clustering_args = list(alpha = 0.1))
summary(out)
plot(out)

```

ICS_lcov

*Local Shape Scatter Estimates for ICS***Description**

It is a wrapper for the local shape estimator of scatter as computed by `fpc::localshape()`.

Usage

```
ICS_lcov(x, mscatter = "cov", proportion = 0.1, ...)
```

Arguments

<code>x</code>	a numeric matrix or data frame.
<code>mscatter</code>	"mcd" or "cov" (default); specified minimum covariance determinant or classical covariance matrix to be used for Mahalanobis distance computation.
<code>proportion</code>	proportion of points to be considered as neighbourhood.
<code>...</code>	potential further arguments passed to <code>fpc::localshape()</code> .

Value

An object of class "ICS_scatter" with the following components:

location	this is NULL as the estimator does not use a location estimate.
scatter	a numeric matrix giving the estimate of the scatter matrix.
label	a character string providing a label for the scatter matrix.

Author(s)

Andreas Alfons and Aurore Archimbaud

See Also

[fpc::localshape\(\)](#)

ICS_mcd

MCD location and Scatter Estimates for ICS

Description

It is a wrapper for the (reweighted) MCD estimators of location and scatter as computed by [rrcov::CovMcd\(\)](#).

Usage

```
ICS_mcd_raw(x, location = FALSE, nsamp = "deterministic", alpha = 0.5, ...)
```

```
ICS_mcd_rwt(x, location = FALSE, nsamp = "deterministic", alpha = 0.5, ...)
```

Arguments

x	a numeric matrix or data frame.
location	a logical indicating whether to include the MCD-estimate of location (defaults to FALSE).
nsamp	number of subsets used for initial estimates or "best", "exact" or "deterministic" (default).
alpha	numeric parameter controlling the size of the subsets over which the determinant is minimized as in rrcov::CovMcd() .
...	potential further arguments passed to rrcov::CovMcd() .

Details

- [ICS_mcd_raw\(\)](#): computes the raw MCD estimates.
- [ICS_mcd_rwt\(\)](#): computes the reweighted MCD estimates.

Value

An object of class "ICS_scatter" with the following components:

location	if requested, a numeric vector giving the location estimate.
scatter	a numeric matrix giving the estimate of the scatter matrix.
label	a character string providing a label for the scatter matrix.

Author(s)

Andreas Alfons and Aurore Archimbaud

See Also

[rrcov::CovMcd\(\)](#)

ICS_mlc

Cauchy location and Scatter Estimates for ICS

Description

It is a wrapper for the Cauchy estimator of location and scatter for a multivariate t-distribution, as computed by [ICS::tM\(\)](#).

Usage

```
ICS_mlc(x, location = FALSE, ...)
```

Arguments

x	a numeric matrix or data frame.
location	a logical indicating whether to include the M-estimate of location (defaults to FALSE).
...	potential further arguments passed to ICS::ICS_tM() .

Value

An object of class "ICS_scatter" with the following components:

location	if requested, a numeric vector giving the location estimate.
scatter	a numeric matrix giving the estimate of the scatter matrix.
label	a character string providing a label for the scatter matrix.

Author(s)

Andreas Alfons and Aurore Archimbaud

See Also

[ICS::tM\(\)](#), [ICS::ICS_tM\(\)](#)

`ICS_tcov`*Pairwise one-step M-estimate of scatter for ICS*

Description

Wrapper function for the pairwise one-step M-estimator of scatter with weights based on pairwise Mahalanobis distances, as computed by `tcov()`. Note that this estimator is based on pairwise differences and therefore no location estimate is returned.

Usage

```
ICS_tcov(x, beta = 2)
```

Arguments

<code>x</code>	a numeric matrix or data frame.
<code>beta</code>	a positive numeric value specifying the tuning parameter of the pairwise one-step M-estimator (default to 2), see <code>tcov()</code> .

Value

An object of class "ICS_scatter" with the following components:

<code>location</code>	this is NULL as the estimator is based on pairwise differences and does not use a location estimate.
<code>scatter</code>	a numeric matrix giving the estimate of the scatter matrix.
<code>label</code>	a character string providing a label for the scatter matrix.

Author(s)

Andreas Alfons

See Also

[ICS\(\)](#)

[tcov\(\)](#), [ucov\(\)](#), [ICS_ucov\(\)](#)

`ICS_ucov`*Simple robust estimates of scatter for ICS*

Description

Wrapper functions for the one-step M-estimator of scatter with weights based on Mahalanobis distances as computed by `scov()`, or the simple related estimator that is based on a transformation as computed by `ucov()`.

Usage

```
ICS_scov(x, location = TRUE, beta = 0.2)
```

```
ICS_ucov(x, location = TRUE, beta = 0.2)
```

Arguments

<code>x</code>	a numeric matrix or data frame.
<code>location</code>	a logical indicating whether to include the sample mean as location estimate (defaults to TRUE).
<code>beta</code>	a positive numeric value specifying the tuning parameter of the estimator (default to 0.2), see <code>ucov()</code> .

Value

An object of class "ICS_scatter" with the following components:

<code>location</code>	if requested, a numeric vector giving the location estimate.
<code>scatter</code>	a numeric matrix giving the estimate of the scatter matrix.
<code>label</code>	a character string providing a label for the scatter matrix.

Author(s)

Andreas Alfons

See Also

[ICS\(\)](#)

[tcov\(\)](#), [ICS_tcov\(\)](#), [ucov\(\)](#)

kmeans_clust	<i>k-means clustering</i>
--------------	---------------------------

Description

Wrapper for performing k-means clustering from `stats::kmeans()`.

Usage

```
kmeans_clust(X, k, clusters_only = FALSE, iter.max = 100, nstart = 20, ...)
```

Arguments

<code>X</code>	a numeric matrix or data frame of the data. It corresponds to the argument <code>x</code> .
<code>k</code>	the number of clusters searched for. It corresponds to the argument <code>centers</code> .
<code>clusters_only</code>	boolean. If TRUE only the partition of the data is returned as a vector. If FALSE the usual output of the <code>kmeans</code> function is returned.
<code>iter.max</code>	the maximum number of iterations allowed.
<code>nstart</code>	if <code>centers</code> is a number, how many random sets should be chosen.
<code>...</code>	other arguments to pass to the <code>stats::kmeans()</code> function.

Value

If `clusters_only` is TRUE a vector of the new partition of the data is returned, i.e a vector of integers (from 1:k) indicating the cluster to which each observation is allocated.

Otherwise a list is returned with the following components:

<code>clust_method</code>	the name of the clustering method, i.e. "kmeans".
<code>clusters</code>	the vector of the new partition of the data, i.e. a vector of integers (from 1:k) indicating the cluster to which each observation is allocated.
<code>...</code>	an object of class "kmeans"
.	.

Author(s)

Aurore Archimbaud

See Also

[stats::kmeans\(\)](#)

Examples

```
kmeans_clust(iris[,1:4], k = 3, clusters_only = TRUE)
```

mclust_clust	<i>Model-Based Clustering</i>
--------------	-------------------------------

Description

Wrapper for performing Model-Based Clustering from `mclust::Mclust()` allowing noise or not.

Usage

```
mclust_clust(X, k, clusters_only = FALSE, ...)
rmclust_clust(X, k, clusters_only = FALSE, ...)
```

Arguments

X	a numeric matrix or data frame of the data. It corresponds to the argument data.
k	the number of clusters searched for. It corresponds to the argument G of function <code>mclust::Mclust()</code> .
clusters_only	boolean. If TRUE only the partition of the data is returned as a vector. If FALSE the usual output of the <code>mclust::Mclust()</code> function is returned.
...	other arguments to pass to <code>mclust::Mclust()</code> .

Details

- `mclust_clust()`: does not allow noise
- `rmclust_clust()`: allows noise

Value

If `clusters_only` is TRUE a vector of the new partition of the data is returned, i.e a vector of integers (from 1:k) indicating the cluster to which each observation is allocated. 0 indicates trimmed observations.

Otherwise a list is returned with the following components:

clust_method	the name of the clustering method, i.e "rimle".
clusters	the vector of the new partition of the data, i.e a vector of integers (from 1:k) indicating the cluster to which each observation is allocated. 0 indicates outlying observations for <code>rmclust_clust()</code> only.
...	an object of class "mclust"

Author(s)

Aurore Archimbaud

See Also

`mclust::Mclust()`

Examples

```
mclust_clust(iris[,1:4], k = 3, clusters_only = TRUE)
```

 med_crit

Selection of Invariant components using the med criterion

Description

Identifies as interesting invariant coordinates whose generalized eigenvalues are the furthestmost away from the median of all generalized eigenvalues.

Usage

```
med_crit(object, ...)

## S3 method for class 'ICS'
med_crit(object, nb_select = NULL, select_only = FALSE, ...)

## Default S3 method:
med_crit(object, nb_select = NULL, select_only = FALSE, ...)
```

Arguments

object	object of class "ICS".
...	additional arguments are currently ignored.
nb_select	the exact number of components to select. By default it is set to NULL, i.e the number of components to select is the number of variables minus one.
select_only	boolean. If TRUE only the vector names of the selected invariant components is returned. If FALSE additional details are returned.

Details

If more than half of the components are "uninteresting" and have the same generalized eigenvalue then the median of all generalized eigenvalues corresponds to the uninteresting component generalized eigenvalue. The components of interest are the ones whose generalized eigenvalues differ the most from the median. The motivation of this criterion depends therefore on the assumption that at least half of the components have equal generalized eigenvalues.

Value

If `select_only` is TRUE a vector of the names of the invariant components or variables to select. If FALSE an object of class "ICS_crit" is returned with the following objects:

- `crit`: the name of the criterion "med".
- `nb_select`: the number of components to select.
- `gen_kurtosis`: the vector of generalized kurtosis values.

- `med_gen_kurtosis`: the median of the generalized kurtosis values.
- `gen_kurtosis_diff_med`: the absolute differences between the generalized kurtosis values and the median.
- `select`: the names of the invariant components or variables to select.

Author(s)

Andreas Alfons, Aurore Archimbaud and Klaus Nordhausen

References

Alfons, A., Archimbaud, A., Nordhausen, K., & Ruiz-Gazen, A. (2022). Tandem clustering with invariant coordinate selection. *arXiv preprint arXiv:2212.06108*.

See Also

[normal_crit\(\)](#), [var_crit\(\)](#), [discriminatory_crit\(\)](#).

Examples

```
X <- iris[,-5]
out <- ICS(X)
med_crit(out, nb_select = 2, select_only = FALSE)
```

mixture_sim

Simulation of a mixture of Gaussian distributions

Description

Simulation of a $n \times p$ data frame according to a mixture of q Gaussian distributions with $q < p$, different location parameters μ_1, \dots, μ_q , and the identity matrix as the covariance matrix.

Usage

```
mixture_sim(pct_clusters = c(0.5, 0.5), n = 500, p = 10, delta = 10)
```

Arguments

<code>pct_clusters</code>	a vector of marginal probabilities for each group, i.e mixture weights. Default is two balanced clusters.
<code>n</code>	integer. The number of observations.
<code>p</code>	integer. The number of variables.
<code>delta</code>	integer. The location shift.

Details

Let X be a p -variate real random vector distributed according to a mixture of q Gaussian distributions with $q < p$, different location parameters μ_1, \dots, μ_q , and the same positive definite covariance matrix I_p :

$$X \sim \sum_{h=1}^q \epsilon_h \mathcal{N}(\mu_h, I_p),$$

where $\epsilon_1, \dots, \epsilon_q$ are mixture weights with $\epsilon_1 + \dots + \epsilon_q = 1$, $\mu_1 = 0_p$, and $\mu_{h+1} = \delta e_h$ with $h = 1, \dots, q-1$.

Value

A dataframe of n observations and $p+1$ variables with the first variable indicating the cluster assignment using a character string.

Author(s)

Aurore Archimbaud

References

Alfons, A., Archimbaud, A., Nordhausen, K., & Ruiz-Gazen, A. (2022). Tandem clustering with invariant coordinate selection. *arXiv preprint arXiv:2212.06108*.

Examples

```
X <- mixture_sim()
summary(X)
```

normal_crit	<i>Selection of Non-normal Invariant Components Using Marginal Normality Tests</i>
-------------	--

Description

Identifies invariant coordinates that are non normal using univariate normality tests as in the `comp.norm.test` function from the `ICSOutlier` package, with the difference that both the first and last few components are investigated.

Usage

```
normal_crit(object, ...)

## S3 method for class 'ICS'
normal_crit(
  object,
  level = 0.05,
```

```

    test = c("agostino.test", "jarque.test", "anscombe.test", "bonett.test",
             "shapiro.test"),
    max_select = NULL,
    select_only = FALSE,
    ...
)

## Default S3 method:
normal_crit(
  object,
  level = 0.05,
  test = c("agostino.test", "jarque.test", "anscombe.test", "bonett.test",
           "shapiro.test"),
  max_select = NULL,
  select_only = FALSE,
  gen_kurtosis = NULL,
  ...
)

```

Arguments

object	object of class "ICS" or a data frame or matrix.
...	additional arguments are currently ignored.
level	the initial level used to make a decision based on the test p-values. See details. Default is 0.05.
test	name of the normality test to be used. Possibilities are "jarque.test", "anscombe.test", "bonett.test", "agostino.test", "shapiro.test". Default is "agostino.test".
max_select	the maximal number of components to select.
select_only	boolean. If TRUE only the vector names of the selected invariant components is returned. If FALSE additional details are returned.
gen_kurtosis	vector of generalized kurtosis values.

Details

The procedure sequentially tests the first and the last components until finding no additional components as non-normal. The quantile levels are adjusted for multiple testing by taking the level as level/j for the j th component.

Value

If `select_only` is TRUE a vector of the names of the invariant components or variables to select. If FALSE an object of class "ICS_crit" is returned with the following objects:

- `crit`: the name of the criterion "normal".
- `level`: the level of the test.
- `max_select`: the maximal number of components to select.
- `test`: name of the normality test to be used.

- `pvalues`: the p-values of the tests.
- `adjusted_levels`: the adjusted levels.
- `select`: the names of the invariant components or variables to select.
- `gen_kurtosis`: the vector of generalized kurtosis values in case of ICS object.

Author(s)

Andreas Alfons, Aurore Archimbaud, Klaus Nordhausen and Anne Ruiz-Gazen

References

Alfons, A., Archimbaud, A., Nordhausen, K., & Ruiz-Gazen, A. (2022). Tandem clustering with invariant coordinate selection. *arXiv preprint arXiv:2212.06108*.

Archimbaud, A., Nordhausen, K., and Ruiz-Gazen, A. (2018). ICSOutlier: Unsupervised Outlier Detection for Low-Dimensional Contamination Structure, *The RJournal*, Vol. 10(1):234–250. [doi:10.32614/RJ2018034](https://doi.org/10.32614/RJ2018034)

Archimbaud, A., Nordhausen, K., and Ruiz-Gazen, A. (2016). ICSOutlier: Outlier Detection Using Invariant Coordinate Selection. R package version 0.3-0

See Also

[med_crit\(\)](#), [var_crit\(\)](#), [discriminatory_crit\(\)](#), [jarque.test\(\)](#), [anscombe.test\(\)](#), [bonett.test\(\)](#), [agostino.test\(\)](#), [stats::shapiro.test\(\)](#).

Examples

```
X <- iris[,-5]
out <- ICS(X)
normal_crit(out, level = 0.1, select_only = FALSE)
```

pam_clust

Partitioning Around Medoids clustering

Description

Wrapper for performing Partitioning Around Medoids clustering from `cluster::pam()`.

Usage

```
pam_clust(X, k, clusters_only = FALSE, ...)
```

Arguments

X	a numeric matrix or data frame of the data. It corresponds to the argument x.
k	the number of clusters searched for. It corresponds to the argument k.
clusters_only	boolean. If TRUE only the partition of the data is returned as a vector. If FALSE the usual output of the <code>cluster::pam()</code> function is returned.
...	other arguments to pass to the <code>cluster::pam()</code> .

Value

If `clusters_only` is TRUE a vector of the new partition of the data is returned, i.e a vector of integers (from 1:k) indicating the cluster to which each observation is allocated. 0 indicates trimmed observations.

Otherwise a list is returned with the following components:

clust_method	the name of the clustering method, i.e "clara_pam".
clusters	the vector of the new partition of the data, i.e a vector of integers (from 1:k) indicating the cluster to which each observation is allocated. 0 indicates outlying observations.
...	an object of class "pam"
.	

Author(s)

Aurore Archimbaud

See Also

[cluster::pam\(\)](#)

Examples

```
pam_clust(iris[,1:4], k = 3, clusters_only = TRUE)
```

plot.ICSClust

Scatterplot Matrix with densities on the diagonal

Description

Wrapper for [component_plot\(\)](#).

Usage

```
## S3 method for class 'ICSClust'
plot(x, ...)
```

Arguments

x an object of class "ICSClust".
... additional arguments to be passed down to `component_plot()`

Value

An object of class "ggmatrix" (see `GGally::ggpairs()`).

Author(s)

Aurore Archimbaud

print.ICSClust_summary

Print of an ICSClust_summary object

Description

Prints an ICSClust_summary object in an informative way.

Usage

```
## S3 method for class 'ICSClust_summary'  
print(x, info = FALSE, digits = 4L, ...)
```

Arguments

x object of class "ICSClust_summary".
info logical, either TRUE or FALSE. If TRUE, prints additional information on arguments used for computing scatter matrices (only named arguments that contain numeric, character, or logical scalars) and information on the parameters of the algorithm. Default is FALSE.
digits number of digits for the numeric output.
... additional arguments are ignored.

Value

The supplied object of class "ICSClust_summary" is returned invisibly.

Author(s)

Aurore Archimbaud

`rimle_clust`*Robust Improper Maximum Likelihood Clustering*

Description

Wrapper for performing Robust Improper Maximum Likelihood Clustering clustering from `otrimle::rimle()`.

Usage

```
rimle_clust(X, k, clusters_only = FALSE, ...)
```

Arguments

<code>X</code>	a numeric matrix or data frame of the data. It corresponds to the argument <code>data</code> .
<code>k</code>	the number of clusters searched for. It corresponds to the argument <code>G</code> .
<code>clusters_only</code>	boolean. If TRUE only the partition of the data is returned as a vector. If FALSE the usual output of the <code>otrimle::rimle()</code> function is returned.
<code>...</code>	other arguments to pass to <code>otrimle::rimle()</code> .

Value

If `clusters_only` is TRUE a vector of the new partition of the data is returned, i.e a vector of integers (from 1:k) indicating the cluster to which each observation is allocated. 0 indicates trimmed observations.

Otherwise a list is returned with the following components:

<code>clust_method</code>	the name of the clustering method, i.e, "rimle".
<code>clusters</code>	the vector of the new partition of the data, i.e. a vector of integers (from 1:k) indicating the cluster to which each observation is allocated. 0 indicates outlying observations.
<code>...</code>	an object of class "rimle"

Author(s)

Aurore Archimbaud

See Also

`otrimle::rimle()`

Examples

```
rimle_clust(iris[,1:4], k = 3, clusters_only = TRUE)
```

runif_outside_range *Uniform distribution outside a given range*

Description

Draw from a multivariate uniform distribution outside a given range. Intuitively speaking, the observations are drawn from a multivariate uniform distribution on a hyperrectangle with a hole in the middle (in the shape of a smaller hyperrectangle). This is useful, e.g., for adding random noise to a data set such that the noise consists of large values that do not overlap the initial data.

Usage

```
runif_outside_range(n, min = 0, max = 1, mult = 2)
```

Arguments

<code>n</code>	an integer giving the number of observations to generate.
<code>min</code>	a numeric vector giving the minimum of each variable of the initial data set (outside of which to generate random noise).
<code>max</code>	a numeric vector giving the maximum of each variable of the initial data set (outside of which to generate random noise).
<code>mult</code>	multiplication factor (larger than 1) to expand the hyperrectangle around the initial data (which is given by <code>min</code> and <code>max</code>). For instance, the default value 2 gives a hyperrectangle for which each side is twice as long as the range of the initial data. The data are then drawn from a uniform distribution on the expanded hyperrectangle from which the smaller hyperrectangle around the data is cut out. See the examples for an illustration.

Value

A matrix of generated points.

Author(s)

Andreas Alfons

References

#¹ Alfons, A., Archimbaud, A., Nordhausen, K., & Ruiz-Gazen, A. (2022). Tandem clustering with invariant coordinate selection. *arXiv preprint arXiv:2212.06108*.

Examples

```
## illustrations for argument 'mult'

# draw observations with argument 'mult = 2'
xy2 <- runif_outside_range(1000, min = rep(-1, 2), max = rep(1, 2),
                          mult = 2)
# each side of the larger hyperrectangle is twice as long as
# the corresponding side of the smaller rectangular cut-out
df2 <- data.frame(x = xy2[, 1], y = xy2[, 2])
ggplot(data = df2, mapping = aes(x = x, y = y)) +
  geom_point()

# draw observations with argument 'mult = 4'
xy4 <- runif_outside_range(1000, min = rep(-1, 2), max = rep(1, 2),
                          mult = 4)
# each side of the larger hyperrectangle is four times as long
# as the corresponding side of the smaller rectangular cut-out
df4 <- data.frame(x = xy4[, 1], y = xy4[, 2])
ggplot(data = df4, mapping = aes(x = x, y = y)) +
  geom_point()
```

select_plot

Plot of the Generalized Kurtosis Values of the ICS Transformation

Description

Extracts the generalized kurtosis values of the components obtained via an ICS transformation and draws either a screeplot or a specific plot for a given criterion. If an object of class "ICS_crit" is given, then the selected components are shaded on the plot.

Usage

```
select_plot(object, ...)

## Default S3 method:
select_plot(
  object,
  select = NULL,
  scale = FALSE,
  screeplot = TRUE,
  type = c("dots", "lines"),
  width = 0.2,
  color = "grey",
  alpha = 0.3,
  size = 3,
  ...
)
```

```

## S3 method for class 'data.frame'
select_plot(
  object,
  type = c("dots", "lines"),
  width = 0.2,
  color = "grey",
  alpha = 0.3,
  ...
)

## S3 method for class 'ICS_crit'
select_plot(
  object,
  type = c("dots", "lines"),
  width = 0.2,
  color = "grey",
  alpha = 0.3,
  size = 3,
  screeplot = TRUE,
  ...
)

```

Arguments

object	an object inheriting from class "ICS" and containing results from an ICS transformation or from class "ICS_crit".
...	additional arguments are currently ignored.
select	an integer, character, or logical vector specifying for which components to extract the generalized kurtosis values, or NULL for extracting the generalized kurtosis values of all components.
scale	a logical indicating whether to scale the generalized kurtosis values to have product 1 (defaults to FALSE).
screeplot	boolean. If TRUE a plot of the generalized kurtosis values is drawn. Otherwise it is context specific to the ICS_crit object. For "med" criterion, the differences between the kurtosis values and the median are plotted in absolute values. For "discriminatory" the discriminatory power associated to the evaluated combinations are drawn.
type	either "dots" or "lines" for the type of plot.
width	the width for shading the selected components in case an ICS_crit object is given.
color	the color for shading the selected components in case an ICS_crit object is given.
alpha	the transparency for shading the selected components in case an ICS_crit object is given.
size	size of the points. Only relevant for "discriminatory" criteria.

Value

An object of class "ggplot" (see [ggplot2::ggplot\(\)](#)).

Author(s)

Andreas Alfons and Aurore Archimbaud

Examples

```
X <- iris[,-5]
out <- ICS(X)

# on an ICS object
select_plot(out)
select_plot(out, type = "lines")

# on an ICS_crit object
# median criterion
out_med <- med_crit(out, nb_select = 1, select_only = FALSE)
select_plot(out_med, type = "lines")
select_plot(out_med, screeplot = FALSE, type = "lines",
  color = "lightblue")

# discriminatory criterion
out_disc <- discriminatory_crit(out, clusters = iris[,5],
  select_only = FALSE)
select_plot(out_disc)
```

summary.ICSClust

Summary of an ICSClust object

Description

Summarizes an ICSClust object in an informative way.

Usage

```
## S3 method for class 'ICSClust'
summary(object, ...)
```

Arguments

object object of class "ICSClust".
... additional arguments passed to [summary\(\)](#)

Value

An object of class "ICSClust_summary" with the following components:

- ICS_out: ICS_out object
- nb_comp: number of selected components
- select: vector of names of selected components
- nb_clusters: number of clusters
- table_clusters: frequency table of clusters

Author(s)

Aurore Archimbaud

tcov

Pairwise one-step M-estimate of scatter

Description

Computes a pairwise one-step M-estimate of scatter with weights based on pairwise Mahalanobis distances. Note that it is based on pairwise differences and therefore does not require a location estimate.

Usage

```
tcov(x, beta = 2)
```

Arguments

x a numeric matrix or data frame.
beta a positive numeric value specifying the tuning parameter of the pairwise one-step M-estimator (defaults to 2), see 'Details'.

Details

For a sample $\mathbf{X}_n = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top$, a positive and decreasing weight function w , and a tuning parameter $\beta > 0$, the pairwise one-step M-estimator of scatter is defined as

$$\text{TCOV}_\beta(\mathbf{X}_n) = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n w(\beta r^2(\mathbf{x}_i, \mathbf{x}_j)) (\mathbf{x}_i - \mathbf{x}_j) (\mathbf{x}_i - \mathbf{x}_j)^\top}{\sum_{i=1}^{n-1} \sum_{j=i+1}^n w(\beta r^2(\mathbf{x}_i, \mathbf{x}_j))},$$

where

$$r^2(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \text{COV}(\mathbf{X}_n)^{-1} (\mathbf{x}_i - \mathbf{x}_j)$$

denotes the squared pairwise Mahalanobis distance between observations \mathbf{x}_i and \mathbf{x}_j based on the sample covariance matrix $\text{COV}(\mathbf{X}_n)$. Here, the weight function $w(x) = \exp(-x/2)$ is used.

Value

A numeric matrix giving the pairwise one-step M-estimate of scatter.

Author(s)

Andreas Alfons and Aurore Archimbaud

References

Caussinus, H. and Ruiz-Gazen, A. (1993) Projection Pursuit and Generalized Principal Component Analysis. In Morgenthaler, S., Ronchetti, E., Stahel, W.A. (eds.) *New Directions in Statistical Data Analysis and Robustness*, 35-46. Monte Verita, Proceedings of the Centro Stefano Franciscini Ascona Series. Springer-Verlag.

Caussinus, H. and Ruiz-Gazen, A. (1995) Metrics for Finding Typical Structures by Means of Principal Component Analysis. In *Data Science and its Applications*, 177-192. Academic Press.

See Also

[ICS_tcov\(\)](#), [ucov\(\)](#), [ICS_ucov\(\)](#)

tkmeans_clust	<i>Trimmed k-means clustering</i>
---------------	-----------------------------------

Description

Wrapper for performing trimmed k-means clustering from `tclust::tkmeans()`.

Usage

```
tkmeans_clust(X, k, clusters_only = FALSE, alpha = 0.05, ...)
```

Arguments

X	a numeric matrix or data frame of the data. It corresponds to the argument x.
k	the number of clusters searched for. It corresponds to the argument k.
clusters_only	boolean. If TRUE only the partition of the data is returned as a vector. If FALSE the usual output of the <code>tkmeans</code> function is returned.
alpha	the proportion of observations to be trimmed.
...	other arguments to pass to the <code>tclust::tkmeans()</code>

Value

If `clusters_only` is TRUE a vector of the new partition of the data is returned, i.e a vector of integers (from 1:k) indicating the cluster to which each observation is allocated. 0 indicates trimmed observations.

Otherwise a list is returned with the following components:

<code>clust_method</code>	the name of the clustering method, i.e. "tkmeans".
<code>clusters</code>	the vector of the new partition of the data, i.e. a vector of integers (from 1:k) indicating the cluster to which each observation is allocated. 0 indicates trimmed observations.
<code>...</code>	an object of class "tkmeans"
.	.

Author(s)

Aurore Archimbaud

See Also

[tclust::tkmeans\(\)](#)

Examples

```
tkmeans_clust(iris[,1:4], k = 3, alpha = 0.1, clusters_only = TRUE)
```

ucov

Simple robust estimates of scatter

Description

Compute a one-step M-estimator of scatter with weights based on Mahalanobis distances, or a simple related estimator that is based on a transformation.

Usage

```
scov(x, beta = 0.2)
```

```
ucov(x, beta = 0.2)
```

Arguments

<code>x</code>	a numeric matrix or data frame.
<code>beta</code>	a positive numeric value specifying the tuning parameter of the estimator (defaults to 0.2), see 'Details'.

Details

For a sample $\mathbf{X}_n = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top$, a positive and decreasing weight function w , and a tuning parameter $\beta > 0$, the one-step M-estimator of scatter is defined as

$$\text{SCOV}_\beta(\mathbf{X}_n) = \frac{\sum_{i=1}^n w(\beta r^2(\mathbf{x}_i))(\mathbf{x}_i - \bar{\mathbf{x}}_n)(\mathbf{x}_i - \bar{\mathbf{x}}_n)^\top}{\sum_{i=1}^n w(\beta r^2(\mathbf{x}_i))},$$

where

$$r^2(\mathbf{x}_i) = (\mathbf{x}_i - \bar{\mathbf{x}}_n)^\top \text{COV}(\mathbf{X}_n)^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}_n)$$

denotes the squared Mahalanobis distance of observation \mathbf{x}_i from the sample mean $\bar{\mathbf{x}}_n$ based on the sample covariance matrix $\text{COV}(\mathbf{X}_n)$. Here, the weight function $w(x) = \exp(-x/2)$ is used.

A simple robust estimator that is consistent under normality is obtained via the transformation

$$\text{UCOV}_\beta(\mathbf{X}_n) = (\text{SCOV}_\beta(\mathbf{X}_n)^{-1} - \beta \text{COV}(\mathbf{X}_n)^{-1})^{-1}.$$

Value

A numeric matrix giving the estimate of the scatter matrix.

Author(s)

Andreas Alfons and Aurore Archimbaud

References

Caussinus, H. and Ruiz-Gazen, A. (1993) Projection Pursuit and Generalized Principal Component Analysis. In Morgenthaler, S., Ronchetti, E., Stahel, W.A. (eds.) *New Directions in Statistical Data Analysis and Robustness*, 35-46. Monte Verita, Proceedings of the Centro Stefano Franciscini Ascona Series. Springer-Verlag.

Caussinus, H. and Ruiz-Gazen, A. (1995) Metrics for Finding Typical Structures by Means of Principal Component Analysis. In *Data Science and its Applications*, 177-192. Academic Press.

Ruiz-Gazen, A. (1996) A Very Simple Robust Estimator of a Dispersion Matrix. *Computational Statistics & Data Analysis*, **21**(2), 149-162. doi:10.1016/01679473(95)000097.

See Also

[ICS_ucov\(\)](#), [tcov\(\)](#), [ICS_tcov\(\)](#)

var_crit	<i>Selection of Invariant components using the var criterion</i>
----------	--

Description

Identifies the interesting invariant coordinates based on the rolling variance criterion as used in the ICSboot function of the ICtest package. It computes rolling variances on the generalized eigenvalues obtained through `ICS::ICS()`.

Usage

```
var_crit(object, ...)

## S3 method for class 'ICS'
var_crit(object, nb_select = NULL, select_only = FALSE, ...)

## Default S3 method:
var_crit(object, nb_select = NULL, select_only = FALSE, ...)
```

Arguments

object	object of class "ICS".
...	additional arguments are currently ignored.
nb_select	the exact number of components to select. By default it is set to NULL, i.e the number of components to select is the number of variables minus one.
select_only	boolean. If TRUE only the vector names of the selected invariant components is returned. If FALSE additional details are returned.

Details

Assuming that the generalized eigenvalues of the uninformative components are all the same means that the variance of these generalized eigenvalues must be minimal. Therefore when nb_select components should be selected, the method identifies the $p - nb_select$ neighboring generalized eigenvalues with minimal variance, where p is the total number of components. The number of interesting components should be at most $p-2$ as at least two uninteresting components are needed to compute a variance.

Value

If select_only is TRUE a vector of the names of the invariant components or variables to select. If FALSE an object of class "ICS_crit" is returned with the following objects:

- crit: the name of the criterion "var".
- nb_select: the number of components to select.
- gen_kurtosis: the vector of generalized kurtosis values.
- select: the names of the invariant components or variables to select.

- RollVarX: the rolling variances of order d-nb_select.
- Order: indexes of the ordered invariant components such that the ones associated to the smallest variances of the eigenvalues are at the end.

Author(s)

Andreas Alfons, Aurore Archimbaud and Klaus Nordhausen

References

Alfons, A., Archimbaud, A., Nordhausen, K., & Ruiz-Gazen, A. (2022). Tandem clustering with invariant coordinate selection. *arXiv preprint arXiv:2212.06108*.

Radojicic, U., & Nordhausen, K. (2019). Non-gaussian component analysis: Testing the dimension of the signal subspace. In *Workshop on Analytical Methods in Statistics* (pp. 101–123). Springer. [doi:10.1007/9783030488147_6](https://doi.org/10.1007/9783030488147_6).

See Also

[normal_crit\(\)](#), [med_crit\(\)](#), [discriminatory_crit\(\)](#).

Examples

```
X <- iris[,-5]
out <- ICS(X)
var_crit(out, nb_select = 2, select_only = FALSE)
```

Index

* package

ICSclust-package, 2

agostino.test(), 20

anscombe.test(), 20

bonett.test(), 20

cluster::pam(), 20, 21

component_plot, 4

component_plot(), 21, 22

discriminatory_crit, 5, 8

discriminatory_crit(), 7, 8, 17, 20, 33

fpc::localshape(), 9, 10

GGally::ggpairs(), 5, 22

ggplot2::ggplot(), 27

ICS, 4, 7, 8, 12, 13

ICS::ICS(), 32

ICS::ICS_tM(), 11

ICS::tM(), 11

ICS_lcov, 9

ICS_mcd, 10

ICS_mcd_raw (ICS_mcd), 10

ICS_mcd_raw(), 10

ICS_mcd_rwt (ICS_mcd), 10

ICS_mcd_rwt(), 10

ICS_mlc, 11

ICS_scov (ICS_ucov), 13

ICS_tcov, 12, 13, 29, 31

ICS_ucov, 12, 13, 29, 31

ICSclust, 7

ICSclust-package, 2

jarque.test(), 20

kmeans, 14

kmeans_clust, 14

kmeans_clust(), 8

mclust::Mclust(), 15

mclust_clust, 15

mclust_clust(), 8, 15

med_crit, 16

med_crit(), 7, 8, 20, 33

mixture_sim, 17

normal_crit, 18

normal_crit(), 7, 8, 17, 33

otrimle::rimle(), 23

pam_clust, 20

pam_clust(), 8

plot(), 8

plot.ICSClust, 21

print.ICSClust_summary, 22

rimle_clust, 23

rimle_clust(), 8

rmclust_clust (mclust_clust), 15

rmclust_clust(), 8, 15

rrcov::CovMcd(), 10, 11

runif_outside_range, 24

scov, 13

scov (ucov), 30

select_plot, 25

stats::kmeans(), 14

stats::shapiro.test(), 20

summary(), 8, 27

summary.ICSClust, 27

tclust::tkmeans(), 29, 30

tcov, 12, 13, 28, 31

tkmeans, 29

tkmeans_clust, 29

tkmeans_clust(), 8

ucov, [12](#), [13](#), [29](#), [30](#)

var_crit, [32](#)

var_crit(), [7](#), [8](#), [17](#), [20](#)