

Package ‘NumClust’

September 24, 2026

Title Finding the Number of Clusters

Version 0.1.3

Date 2026-09-24

Description Implements methods to estimate the ``true" number of clusters in a data set.

Depends R (>= 4.5.0)

Imports stats, graphics

Suggests NbClust, knitr, rmarkdown

License GPL-2 | GPL-3

LazyLoad yes

URL <http://silicovore.com/OOMPA/index.html>

VignetteBuilder knitr

NeedsCompilation no

Author Kevin R. Coombes [cre, aut],
Polina Bombina [aut],
Malika Charad [aut],
Nadia Ghazzali [aut],
Veronique Boiteau [aut],
Azam Niknafs [aut]

Maintainer Kevin R. Coombes <krc@silicovore.com>

Contents

example	2
NumClust	2
registries	4
Index	6

example

An example data set and Euclidean distance matrix

Description

The Euclidean distance matrix between 300150 objects, used to illustrate the SillyPutty algorithms.

Usage

```
data(example)
```

Format

The binary R data file contains three objects. First, a `dist` object representing Euclidean distances between 150 samples. Second, a vector of the known (simulated) true groups to which each sample belongs. Finally, the simulated dataset, which has 144 rows and 11808 columns.

Details

This data set was generated in the SillyPutty vignette from tools in the Umpire R package. The simulated data was intended to have five different clusters, all of approximately the same size. Noise was added to make the clusters somewhat harder for most algorithms to distinguish. The same data set is used in most of the examples in the man pages.

Source

This data set was generated in the SillyPutty vignette from the tools in the Umpire R package, and saved using code that is now hidden and disabled in the vignette source.

Examples

```
data(example)
class(eucdist)
attr(eucdist, "Size")
dim(dataset)
table(trueGroups)
```

NumClust

Estimating the Number of Clusters

Description

Provides (at least) 30 indices for determining the number of clusters. Allows the user to select distance metrics, hierarchical clustering linkage rules, and the range of allowable cluster numbers.

Usage

```
NumClust(data, diss = NULL, distance = "euclidean",
         hc = NULL, method = NULL, debug = FALSE,
         min.nc = 2, max.nc = 15, index = "all",
         alphaBeale = 0.1)
```

Arguments

data	A matrix containing the original data, with rows as samples or objects to be clustered.
diss	An optional precomputed dist object containing pairwise distances between objects.
distance	A character string naming the distance metric to be used in dist.
hc	An optional precomputed hclust object.
method	A character string naming the linkage rule to be used in hclust.
debug	A logical value; should extraintermmediate output be generated for purposes of debugging the code.
min.nc	An integer; the minimal number of clusters, between 1 and (number of objects - 1). By default, min.nc = 2.
max.nc	An integer; maximal number of clusters, between 2 and (number of objects - 1), greater or equal to min.nc. By default, max.nc=15.
index	A character vector; which of the 30 implemented algorithms to compute. Default is the special value, "all". For a complete list of available indices, see availableIndices .
alphaBeale	The significance value for Beale's index.

Details

There are thirty built-in pre-defined "indices" for estimating the number of clusters. These are listed, with citations of the original published references, in the documentation of the [NbClust](#) package.

There are several innovations in the current NumClust package. One addition visible here is the addition of the hc argument, which allows the use to precompute the hierarchical clustering dendrogram. On large data sets, this computation can be time-consuming. Consequently, it may be worth the effort to save the dendrogram (after using it, perhaps, for visualization purposes or to decide on a sensible range of cluster numbers to explore in more detail) and avoid having to wait while it is recomputed.

The main addition, however, is the use of a "registry" to store the algorithms (or indices) used for cluster number estimation. The package pre-loads the same thirty method implemented in NbClust, but makes it easy to add your own favorite method and compare it with any of the existing methods.

Value

The function returns a list with two or three components:

1. Stats, a matrix with one column for each index and one row for each possible number of clusters, containing the computed scores used in the evaluations.
2. Critical, an optional matrix, with one column for each index that requires comparison with "critical values" to decide on the optional cluster number.
3. Best.nc, a matrix with one column for each index, and two rows: N, the optimal number of clusters, and score, the associated derived score driving the estimation.

Author(s)

Kevin R. Coombes <krc@silicovore.com>

References

Charrad M., Ghazzali N., Boiteau V., Niknafs A. (2014). "NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set.", "Journal of Statistical Software, 61(6), 1-36.", "URL <http://www.jstatsoft.org/v61/i06/>".

See Also

[NbClust](#), [availableIndices](#), [dist](#), and [hclust](#).

Examples

```
availableIndices()
```

registries	<i>Method Argument Registries</i>
------------	-----------------------------------

Description

Tools to register the names of arguments used by other functions to select from a variety of algorithms.

Usage

```
registerMetric(tag, description)
availableMetrics()
registerLinkage(tag, description)
availableLinkages()
registerIndice(tag, description, FUN, RULE)
availableIndices()
deregister(registry, name)
```

Arguments

tag	A character string; a short name used as an argument to a function
description	A character string describing the algorithm indicated by the tag
FUN	A function, which will be invoked by the higher-level function. The call provides a list of named arguments (see details). The function is expected to return a list with one required element (Stat) and one optional element (Critical)
RULE	A function used to select the best number of clusters. The inputs to the FUN function are a vector, vals, of scores for each cluster number considered and a vector of "critical" values to compare against. Some rules ignore the critical values, such as those that just take the maximum or minimum score as the best. The rule should return a list with two items, the index into the vals vector where the best result occurs, and a derived score used to select it.
registry	The name of a registry (metrics, linkages, or indices).
name	The name of an object to be removed from the registry.

Details

The `registerMetric` and `availableMetrics` lists are expected to be static, and reproduce the arguments available to the `dist` function. Similarly, the `registerLinkage` and `availableLinkageslist` are expected to be static, and reproduce the arguments available to the `hclust` function. Both of these are included here to allow other functions (like `NumClust`) to take them as arguments that can be passed along in called to the core functions.

By contrast, the `registerIndices` and `availableIndices` lists are an essential part of the implementation of the `NumClust` function, which is designed to compute different metrics that estimate the number of distinct clusters present in a data set. By using the "registry" design pattern, it is easier for a user to implement their own metric and test it on datasets using the same harness that is provided by `NumClust`.

The call to the associated function, `FUN`, is a list containing

- `jeu`, the data matrix used to form distances and clusters.
- `md`, a distance matrix stored as a `dist` object.
- `cl1`, cluster assignments for all objects, usually obtained from a call to `cutree` with `$k$` classes.
- `cl2`, cluster assignments for all objects, using `$k+1$` classes.
- `cl0`, cluster assignments with `$k-1$` classes.
- `clmax`, cluster assignments using the maximum number of clusters in the range of allowable values.
- `clall`, a matrix obtained by `cbinding` `cl0`, `cl1`, and `cl2`, in that order.
- `method`, a character string identifying one of the linkage rules for `hclust`.
- `pp`
- `TT`
- `ss`
- `vv`

Value

Each `register*` function returns a character string containing the tag that was just registered.

Each `available*` function returns a data frame containing the tags and their associated description.

Author(s)

Kevin R. Coombes <krc@silicovore.com>

Examples

```
availableMetrics()
availableLinkages()
availableIndices()
```

Index

- * **datasets**
 - example, [2](#)
- * **programming**
 - NumClust, [2](#)
 - registries, [4](#)
- availableIndices, [3](#), [4](#)
- availableIndices (registries), [4](#)
- availableLinkages (registries), [4](#)
- availableMetrics (registries), [4](#)
- dataset (example), [2](#)
- deregister (registries), [4](#)
- dist, [4](#), [5](#)
- euclust (example), [2](#)
- example, [2](#)
- hclust, [4](#), [5](#)
- NbClust, [3](#), [4](#)
- NumClust, [2](#)
- registerIndice (registries), [4](#)
- registerLinkage (registries), [4](#)
- registerMetric (registries), [4](#)
- registries, [4](#)
- trueGroups (example), [2](#)