

Package ‘clusterfly’

April 17, 2009

Type Package

Title Explore clustering interactively using R and GGobi

Version 0.2.2

Date 2007-06-25

Author Hadley Wickham <h.wickham@gmail.com>

Maintainer Hadley Wickham <h.wickham@gmail.com>

Description Visualise clustering algorithms with GGobi. Contains both general code for visualising clustering results and specific visualisations for model-based, hierarchical and SOM clustering. See <http://had.co.nz/clusterfly> for more information.

Depends rggobi, ggplot2, e1071, cluster

Suggests som, mclust

License MIT

LazyData true

Repository CRAN

Date/Publication 2007-06-28 18:35:02

R topics documented:

addhull	2
as.data.frame.clusterfly	3
cfly_animate	3
cfly_clarify	4
cfly_cluster	5
cfly_dist	5
cfly_fluct	6
cfly_pcp	7
cfly_show	7

clarify	8
clusterfly	8
cut.hierfly	9
ggobi.hierfly	10
ggobi.som	11
hierarchical	11
hierfly	12
mefly	13
Olives data cluster object	13

Index	14
--------------	-----------

addhull	<i>Add convex hulls</i>
---------	-------------------------

Description

Add convex hulls using the tool qconvex

Usage

```
addhull(gd, g, by)
```

Arguments

gd	ggobi dataset
g	ggobi reference
by	grouping variable

Details

To use this command you must have qconvex installed and available on your path. I'm not sure if this will work on windows (probably not) but it's not a big loss, because the technique isn't very useful anyway.

Author(s)

Hadley Wickham <h.wickham@gmail.com>

```
as.data.frame.clusterfly
```

Convert clusterfly object to data.frame

Description

Concatenates data and cluster assignments into one data.frame. Cluster assignments are prefixed with `cl_`

Usage

```
as.data.frame.clusterfly(x, ...)
```

Arguments

`x` clusterfly object
`...`

Author(s)

Hadley Wickham <h.wickham@gmail.com>

```
cfly_animate
```

Dynamic plot: Animate glyph colours

Description

Animate glyph colours according to provided cluster ids.

Usage

```
cfly_animate(cf, i = 1:length(cf$clusters), pause = 1)
```

Arguments

`cf` list of cluster ids that you want to animate across
`i` number of seconds to pause between each change
`pause`

Details

This function will animate endlessly until you manually break the loop using Ctrl-Break or Ctrl-C.

Author(s)

Hadley Wickham <h.wickham@gmail.com>

Examples

```
# Press Ctrl-Break or Ctrl-C to exit
## Not run: cfly_animate(cfly_clarify(o))
```

cfly_clarify	<i>Clarify clusters</i>
--------------	-------------------------

Description

Match all cluster indices to common reference

Usage

```
cfly_clarify(cf, reference=1, method="rowmax")
```

Arguments

cf	clusterfly object
reference	method to use, see clarify
method	index to reference clustering

Details

It's a good idea to run this before running any animation sequences so that unnecessary colour changes are minimised.

Author(s)

Hadley Wickham <h.wickham@gmail.com>

Examples

```
o <- cfly_clarify(o, "Region")
```

cfly_cluster *Add clustering*

Description

Add clustering

Usage

```
cfly_cluster(cf, method, ..., name = deparse(substitute(method)))
```

Arguments

cf	clusterfly object
method	clusterfing method (function)
...	arguments passed to clustering method
name	name of clustering

Details

Clustering method needs to respond to `clusters`, if the default does not work, you will need to write your own to extract clusters.

Author(s)

Hadley Wickham <h.wickham@gmail.com>

Examples

```
cfly_cluster(o, kmeans, 4)
cfly_cluster(o, kmeans, 4, name="blah")
```

cfly_dist *Static plot: Variable distribution*

Description

Draw a density plot for each continuous variable, faceted across clustering.

Usage

```
cfly_dist(cfly, index, scale="range")
```

Arguments

cfly	clusterfly object
index	clustering to use
scale	scaling to use

Details

This allows you to quickly visualise how the cluster vary in a univariate manner. Currently, it is a bit of a hack, because `ggplot` does not support plots with different scales, so the variables are manually rescaled prior to plotting.

This plot is inspired by Gaguin <http://www.rosuda.org/gaguin>

Author(s)

Hadley Wickham <h.wickham@gmail.com>

Examples

```
cfly_dist(o, "kmeans")
cfly_dist(o, "kmeans") + scale_y_continuous(limit=c(0, 2))
```

cfly_fluct

Static plot: Fluctuation diagram

Description

Draw a fluctuation diagram comparing two clusterings

Usage

```
cfly_fluct(cfly, a, b, clarify=TRUE, ...)
```

Arguments

cfly	clusterfly object
a	first clustering, will be reordered to match b if clarify=TRUE
b	second clustering
clarify	use <code>clarify</code> to rearranged cluster indices?
...	

Author(s)

Hadley Wickham <h.wickham@gmail.com>

Examples

```
cfly_fluct(o, "kmeans", "hierarchical", clarify=TRUE)
```

`cfly_pcp`*Static plot: Parallel coordinates*

Description

Draw a parallel coordinates plot, faceted across clustering.

Usage

```
cfly_pcp(cfly, index, ...)
```

Arguments

<code>cfly</code>	clusterfly object
<code>index</code>	clustering to use
<code>...</code>	other arguments passed to geom_line

Details

This really only a proof of concept, a truly useful PCP needs interaction, especially to move the variables around.

Author(s)

Hadley Wickham <h.wickham@gmail.com>

`cfly_show`*Dynamic plot: show in ggobi*

Description

Opens an instance ggobi for this dataset (if not already open), and colours the points according the cluster assignment.

Usage

```
cfly_show(cf, idx = "true", hulls = FALSE)
```

Arguments

<code>cf</code>	clusterfly object
<code>idx</code>	clustering to display
<code>hulls</code>	add convex hull? see addhull for details

Author(s)

Hadley Wickham <h.wickham@gmail.com>

clarify

Clarify matrix

Description

Clarify matrix ordering to minimize off diagonals

Usage

```
clarify(a, b, method="greedy")
```

Arguments

a	cluster assignments to reassign
b	matrix b
method	

Value

vector of reassigned cluster a

Author(s)

Hadley Wickham <h.wickham@gmail.com>

See Also

[matchClasses](#)

clusterfly

Clustefly

Description

This method creates a convenient data structure for dealing with a dataset and a number of alternative clusterings.

Usage

```
clusterfly(df, rescale=TRUE)
```

Arguments

df	data frame to be clustered
rescale	rescale, if true each variable will be scaled to have mean 0 and variance 1.

Details

Once you have created a clusterfly object, you can add clusterings to it with `cfly_cluster`, and visualise then in GGobi with `cfly_show` and `cfly_animate`. Static graphics are also available: `cfly_pcp` will produce a parallel coordinates plot, `cfly_dist` will show the distribution of each variable in each cluster, and `cfly_fluct` compares two clusterings with a fluctuation diagram.

If you want to standardise the cluster labelling to one group, look at `clarify` and `cfly_clarify`

Author(s)

Hadley Wickham <h.wickham@gmail.com>

See Also

`vignette("introduction")`

`cut.hierfly`*Colour hierfly object*

Description

Colour hierfly object into k clusters

Usage

```
cut.hierfly(x, k=2, g=ggobi(x), ...)
```

Arguments

<code>x</code>	hierfly object to colour
<code>k</code>	number of clusters
<code>g</code>	GGobi instance displaying x, will create new if not specified
<code>...</code>	

Author(s)

Hadley Wickham <h.wickham@gmail.com>

Examples

```
h <- hierfly(iris)
hfly <- ggobi(h)
cut(h, 2, hfly)
h <- hierfly(iris, method="ward")
g <- ggobi(h)
cut(h, 2, g)
```

`ggobi.hierfly`*Visualise hierarchical clustering with GGobi*

Description

Displays both data, and dendrogram in original high-d space

Usage

```
ggobi.hierfly(data, ...)
```

Arguments

`data` hierfly object to visualise in GGobi
...

Details

This adds four new variables to the original data set:

- ORDER, the order in which the clusters are joined
- HEIGHT, the height of the branch, ie. the dissimilarity between the branches
- LEVEL, the level of the branch
- POINTS, the number of points in the branch

Make sure to select "attach edge set (edges)" in the in the edges menu on the plot window, when you create a new plot.

A tour over the original variables will show how the clusters agglomerate in space. Plotting order vs height, level or points will give various types of dendograms. A correlation tour with height/level/points on the y axis and the original variables on the x axis will show a mobile blowing in the wind.

Author(s)

Hadley Wickham <h.wickham@gmail.com>

See Also

[cut.hierfly](#)

Examples

```
h <- hierfly(iris)
ggobi(h)
h <- hierfly(iris, method="single")
```

`ggobi.som`*Visualise Kohonen self organising maps with GGobi*

Description

Displays both data, and map in original high-d space.

Usage

```
ggobi.som(data, ...)
```

Arguments

```
data          SOM object
...           
```

Details

Map variables added as `map1` and `map2`. Plot these to get traditional SOM plot. Tour over all other variables to see how well the map fits the original data.

Author(s)

Hadley Wickham <h.wickham@gmail.com>

Examples

```
## Not run:
d.music <- read.csv("http://www.ggobi.org/book/data/music-all.csv")

music <- rescaler(d.music)[complete.cases(d.music), 1:10]
music.som <- som(music[,-(1:3)], 6, 6, neigh="bubble", rlen=1000)
ggobi(music.som)
## End(Not run)
```

`hierarchical`*Hierarchical clustering*

Description

Convenient methods for hierarchical clustering

Usage

```
hierarchical(df, method="complete", metric="euclidean", n=5)
```

Arguments

<code>df</code>	data frame
<code>method</code>	method to use, see hclust
<code>metric</code>	distance metric to use, see dist
<code>n</code>	number of clusters to retrieve, see cut

Author(s)

Hadley Wickham <h.wickham@gmail.com>

hierfly

Hierfly, a method for visualisation hierarchical clustering.

Description

This method supplements a data set with information needed to draw a dendrogram

Usage

```
hierfly(data, metric="euclidean", method="average")
```

Arguments

<code>data</code>	data set
<code>metric</code>	distance metric to use, see dist for list of possibilities
<code>method</code>	cluster distance measure to use, see hclust for details

Details

Intermediate cluster nodes are added as needed, and positioned at the centroid of the combined clusters.

Author(s)

Hadley Wickham <h.wickham@gmail.com>

See Also

[cut.hierfly](#), [ggobi.hierfly](#)

Examples

```
h <- hierfly(iris)
ggobi(h)
h <- hierfly(iris, method="single")
```

`mefly`*Display model based clustering with mvn ellipses*

Description

Displays the results of model based clustering with an ellipse drawn from the multivariate normal model for each group.

Usage

```
mefly(model, data)
```

Arguments

<code>model</code>	output from me function
<code>data</code>	input data frame to me

Author(s)

Hadley Wickham <h.wickham@gmail.com>

Examples

```
if(require("mclust")) {  
  eei <- me(modelName = "EEI", data = iris[,-5], z = unmap(iris[,5]))  
  vvv <- me(modelName = "VVV", data = iris[,-5], z = unmap(iris[,5]))  
  vvi <- me(modelName = "VVI", data = iris[,-5], z = unmap(iris[,5]))  
  mefly(eei, iris[,-5])  
  mefly(vvi, iris[,-5])  
  mefly(vvv, iris[,-5])  
}
```

`Olives data cluster object`*Olives data stored in clusterfly object*

Usage

```
data(o)
```

Index

*Topic **cluster**

- cut.hierfly, 9
- ggobi.hierfly, 9
- ggobi.som, 10
- hierarchical, 11
- hierfly, 12
- mefly, 13

*Topic **datasets**

- Olives data cluster object, 13

*Topic **dynamic**

- cfly_animate, 3
- cfly_show, 7
- clusterfly, 8
- ggobi.hierfly, 9
- ggobi.som, 10
- mefly, 13

*Topic **hplot**

- addhull, 2
- cfly_dist, 5
- cfly_fluct, 5
- cfly_pcp, 6

*Topic **manip**

- as.data.frame.clusterfly, 2
- cfly_clarify, 3
- cfly_cluster, 4
- clarify, 7

addhull, 2, 7

as.data.frame.clusterfly, 2

cfly_animate, 3, 8
cfly_clarify, 3, 8
cfly_cluster, 4, 8
cfly_dist, 5, 8
cfly_fluct, 5, 8
cfly_pcp, 6, 8
cfly_show, 7, 8
clarify, 3, 6, 7, 8
clusterfly, 8

clusters, 4

cut, 11

cut.hierfly, 9, 10, 12

dist, 11, 12

geom_line, 6

ggobi.hierfly, 9, 12

ggobi.som, 10

ggplot, 5

hclust, 11, 12

hierarchical, 11

hierfly, 12

matchClasses, 8

mefly, 13

o (Olives data cluster object), 13

Olives data cluster object, 13

package-clusterfly (*clusterfly*), 8