Package: usedist (via r-universe)

September 7, 2024

Type Package

Title Distance Matrix Utilities

Version 0.4.0.9000

Description Functions to re-arrange, extract, and work with distances.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Suggests future.apply, rlang, tibble, tidyr (>= 1.0.0), testthat, ggplot2

Repository https://kylebittinger.r-universe.dev

RemoteUrl https://github.com/kylebittinger/usedist

RemoteRef HEAD

RemoteSha 46e3e3ea5ecd33adb9d87460049f2dc187aa12de

Contents

`dist_between_centroids`

Compute the distance between group centroids

Description

Compute the distance between group centroids

Usage

```
dist_between_centroids(d, idx1, idx2, squared = FALSE)
```

Arguments

d	A distance matrix object of class `dist`.
idx1	A vector of items in group 1.
idx2	A vector of items in group 2.
squared	If `TRUE`, return the squared distance between centroids.

Details

If you have a distance matrix, and the objects are partitioned into groups, you might like to know the distance between the group centroids. The centroid of each group is simply the center of mass for the group.

It is possible to infer the distance between group centroids directly from the distances between items in each group. The `adonis` test in the ecology package `vegan` takes advantage of this approach to carry out an ANOVA-like test on distances.

The approach rests on the assumption that the objects occupy some high-dimensional Euclidean space. However, we do not have to actually create the space to find the distance between centroids. Based on the assumption that such a space exists, we can use an algebraic formula to perform the computation.

The formulas for this were presented by Apostol and Mnatsakanian in 2003, though we need to re-arrange equation 28 in their paper to get the value we want:

$$|c_1 - c_2| = \sqrt{\frac{1}{n_1 n_2} \sum_{(1,2)} - \frac{1}{n_1^2} \sum_{(1)} - \frac{1}{n_2^2} \sum_{(2)}},$$

where n_1 is the number of samples in group 1, $\sum_{(1)}$ is the sum of squared distances between items in group 1, and $\sum_{(1,2)}$ is the sum of squared distances between items in group 1 and those in group 2.

Sometimes, the distance between centroids is not a real number, because it is not possible to create a space where this distance exists. Mathematically, we get a negative number underneath the square root in the equation above. If this happens, the function returns NaN. If you'd like to have access to this value, you can set `squared = TRUE` to return the squared distance between centroids. In this case, you will never get NaN, but you might receive negative numbers in your result.

Value

The distance between group centroids (see details).

References

Apostol, T.M. and Mnatsakanian, M.A. Sums of squares of distances in m-space. Math. Assoc. Am. Monthly 110, 516 (2003).

---

dist_get   *Retrieve distances from a* dist *object.*

Description

Retrieve distances from a dist object.

Usage

```
dist_get(d, idx1, idx2)
```

Arguments

d   A distance matrix object of class dist.

idx1, idx2   Indices specifying the distances to extract.

Value

A vector of distances.

Examples

```
m4 <- matrix(1:16, nrow=4, dimnames=list(LETTERS[1:4]))
dm4 <- dist(m4)
dist_get(dm4, "A", "C")
dist_get(dm4, "A", c("A", "B", "C", "D"))
dist_get(dm4, c("A", "B", "C"), c("B", "D", "B"))
```

---

dist_groups   *Create a data frame of distances between groups of items.*

Description

Create a data frame of distances between groups of items.

Usage

```
dist_groups(d, g)
```

Arguments

d	A distance matrix object of class `dist`.
g	A factor representing the groups of objects in d.

Value

A data frame with 6 columns:

Item1, Item2 The items being compared.

Group1, Group2 The groups to which the items belong.

Label A convenient label for plotting or comparison.

Distance The distance between Item1 and Item2.

Examples

```
m4 <- matrix(1:16, nrow=4, dimnames=list(LETTERS[1:4]))
dm4 <- dist(m4)
g4 <- rep(c("Control", "Treatment"), each=2)
dist_groups(dm4, g4)
```

---

dist_make   *Make a distance matrix using a custom distance function*

Description

Make a distance matrix using a custom distance function

Usage

```
dist_make(x, distance_fcn, ...)
```

Arguments

x	A matrix of observations, one per row
distance_fcn	A function used to compute the distance between two rows of the data matrix. The two rows will be passed as the first and second arguments to distance_fcn.
...	Additional arguments passed to distance_fcn.

Details

We do not set the call or method attributes of the dist object.

Value

A dist object containing the distances between rows of the data matrix.

Examples

```
x <- matrix(sin(1:30), nrow=5)
rownames(x) <- LETTERS[1:5]
manhattan_distance <- function (v1, v2) sum(abs(v1 - v2))
dist_make(x, manhattan_distance)
```

---

dist_multi_centroids	*Make a new distance matrix of centroid distances between multiple groups*

Description

Make a new distance matrix of centroid distances between multiple groups

Usage

```
dist_multi_centroids(d, g, squared = FALSE)
```

Arguments

d	A distance matrix object of class dist.
g	A factor representing the groups of items in d.
squared	If TRUE, return the squared distance between centroids.

Value

A distance matrix of distances between the group centroids.

dist_setNames   *Set the names/labels of a* dist *object.*

Description

Set the names/labels of a dist object.

Usage

```
dist_setNames(d, nm)
```

Arguments

d	A distance matrix object of class dist.
nm	New labels for the rows/columns.

Value

A distance matrix with new row/column labels.

Examples

```
m4 <- matrix(1:16, nrow=4, dimnames=list(LETTERS[1:4]))
dm4 <- dist(m4)
dist_setNames(dm4, LETTERS[9:12])
```

dist_subset   *Extract parts of a* dist *object.*

Description

Extract a subset of values from a distance matrix. This function also works to re-arrange the rows of a distance matrix, if they are provided in the desired order.

Usage

```
dist_subset(d, idx)
```

Arguments

d	A distance matrix object of class dist.
idx	Indices specifying the subset of distances to extract.

Value

A distance matrix.

Examples

```
m4 <- matrix(1:16, nrow=4, dimnames=list(LETTERS[1:4]))
dm4 <- dist(m4)
dist_subset(dm4, c("A", "B", "C"))
dist_subset(dm4, c("D", "C", "B", "A"))
```

---

dist_to_centroids   *Compute distances from each item to group centroids*

Description

Compute distances from each item to group centroids

Usage

```
dist_to_centroids(d, g, squared = FALSE)
```

Arguments

d	A distance matrix object of class dist.
g	A factor representing the groups of items in d.
squared	If TRUE, return the squared distance to group centroids.

Details

This function computes the distance from each item to the centroid positions of groups defined in the argument g. This is accomplished without determining the centroid positions directly; see the documentation for dist_between_centroids for details on this procedure.

If the distance can't be represented in a Euclidean space, the CentroidDistance is set to NaN. See the documentation for dist_between_centroids for further details.

Value

A data frame with distances to the group centroids:

Item A character vector of item labels from the dist object, or an integer vector of item locations if labels are not present.

CentroidGroup The group for which the centroid distance is given. The column type should match that of the argument g (the unique function is used to generate this column).

CentroidDistance Inferred distance from the item to the centroid position of the indicated group.

| pivot_to_matrix | *Convert a data frame in long format to a matrix* |

Description

Convert a data frame in long format to a matrix

Usage

```
pivot_to_matrix(data, rows_from, cols_from, values_from, fill = 0)

pivot_to_numeric_matrix(data, obs_col, feature_col, value_col)
```

Arguments

`data`	A data frame in long format.
`rows_from`	The column indicating the row of the matrix.
`cols_from`	The column indicating the column of the matrix.
`values_from`	The column indicating the value to be placed inside the matrix.
`fill`	The value to use for missing combinations of rows and columns.
`obs_col`, `feature_col`, `value_col`	
	The same as `rows_from`, `cols_from`, and `values_from`, respectively.

Details

The parameters `rows_from`, `cols_from`, and `values_from` should be provided as bare column names.

This function requires the packages `tidyr`, `rlang`, and `tibble` to be installed. If they are not installed, the function will generate an error, with a message to install the appropriate packages.

Functions

- `pivot_to_numeric_matrix()`: Specialized version for numeric values. Deprecated; use `pivot_to_matrix` instead.

Examples

```
longdata <- data.frame(
  sample_id = paste0("Sample", c(1, 1, 1, 2, 2, 3, 3)),
  feature_id = paste0("Feature", c(1, 2, 3, 1, 2, 2, 3)),
  counts = c(132, 41, 7, 56, 11, 929, 83))
pivot_to_matrix(longdata, sample_id, feature_id, counts)
```

Index