

dist

From [stats v3.6.1](#)by [R-core R-core@R-project.org](#)99.99th
Percentile

Distance Matrix Computation

This function computes and returns the distance matrix computed by using the specified distance measure to compute the distances between the rows of a data matrix.

Keywords [multivariate](#), [cluster](#)

Usage

```
dist(x, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)

as.dist(m, diag = FALSE, upper = FALSE)
# S3 method for default
as.dist(m, diag = FALSE, upper = FALSE)

# S3 method for dist
print(x, diag = NULL, upper = NULL,
      digits = getOption("digits"), justify = "none",
      right = TRUE, ...)

# S3 method for dist
as.matrix(x, ...)
```

Arguments

- x** a numeric matrix, data frame or "dist" object.
- method** the distance measure to be used. This must be one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski". Any unambiguous substring can be given.
- diag** logical value indicating whether the diagonal of the distance matrix should be printed by `print.dist`.
- upper** logical value indicating whether the upper triangle of the distance matrix should be printed by `print.dist`.
- p** The power of the Minkowski distance.
- m** An object with distance information to be converted to a "dist" object. For the default method, a "dist" object, or a matrix (of distances) or an object which can be coerced to such a matrix using `as.matrix()`. (Only the lower triangle of the matrix is used, the rest is ignored).
- digits, justify** passed to `format` inside of `print()`.
- right, ...** further arguments, passed to other methods.

Details

Available distance measures are (written for two vectors x and y):

euclidean : Usual distance between the two vectors (2 norm aka L_2), $\sqrt{\sum_i (x_i - y_i)^2}$.

maximum : Maximum distance between two components of x and y (supremum norm)

manhattan : Absolute distance between the two vectors (1 norm aka L_1).

canberra : $\sum_i |x_i - y_i| / (|x_i| + |y_i|)$. Terms with zero numerator and denominator are omitted from the sum and treated as if the values were missing.

This is intended for non-negative values (e.g., counts), in which case the denominator can be written in various equivalent ways: Originally, R used $x_i + y_i$, then from 1998 to 2017, $|x_i + y_i|$, and then the correct $|x_i| + |y_i|$.

(aka *asymmetric binary*): The vectors are regarded as binary bits, so non-zero elements are 'on' and zero elements are 'off'. The distance is the *proportion* of bits in which only one is on amongst those in which at least one is on.

minkowski : The p norm, the p th root of the sum of the p th powers of the differences of the components.

Missing values are allowed, and are excluded from all computations involving the rows within which they occur. Further, when `Inf` values are involved, all pairs of values are excluded when their contribution to the distance gave `NaN` or `NA`. If some columns are excluded in calculating a Euclidean, Manhattan, Canberra or Minkowski distance, the sum is scaled up proportionally to the number of columns used. If all pairs are excluded when calculating a particular distance, the value is `NA`.

The `"dist"` method of `as.matrix()` and `as.dist()` can be used for conversion between objects of class `"dist"` and conventional distance matrices.

`as.dist()` is a generic function. Its default method handles objects inheriting from class `"dist"`, or coercible to matrices using `as.matrix()`. Support for classes representing distances (also known as dissimilarities) can be added by providing an `as.matrix()` or, more directly, an `as.dist` method for such a class.

Value

`dist` returns an object of class `"dist"`.

The lower triangle of the distance matrix stored by columns in a vector, say `do`. If `n` is the number of observations, i.e., `n <- attr(do, "Size")`, then for $i < j \leq n$, the dissimilarity between (row) i and j is `do[n*(i-1) - i*(i-1)/2 + j-i]`. The length of the vector is $n * (n - 1) / 2$, i.e., of order n^2 .

The object has the following attributes (besides `"class"` equal to `"dist"`):

Size

integer, the number of observations in the dataset.

Labels

optionally, contains the labels, if any, of the observations of the dataset.

Diag, Upper

logicals corresponding to the arguments `diag` and `upper` above, specifying how the object should be printed.

call

optionally, the `call` used to create the object.

method

optionally, the distance method used; resulting from `dist()`, the (`match.arg()`ed) `method` argument.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

Mardia, K. V., Kent, J. T. and Bibby, J. M. (1979) *Multivariate Analysis*. Academic Press.

Borg, I. and Groenen, P. (1997) *Modern Multidimensional Scaling. Theory and Applications*. Springer.

See Also

[daisy](#) in the cluster package with more possibilities in the case of *mixed* (continuous / categorical) variables. [hclust](#).

Examples

```
script.R
1 # NOT RUN {
2 require(graphics)
3
4 x <- matrix(rnorm(100), nrow = 5)
5 dist(x)
6 dist(x, diag = TRUE)
7 dist(x, upper = TRUE)
```

R Console

```
> |
```

```
9 d <- as.dist(m)
10 stopifnot(d == dist(x))
11
12 ## Use correlations between variables "as distance"
13 dd <- as.dist((1 - cor(USJudgeRatings))/2)
```

Run ●

Powered by DataCamp

Documentation reproduced from package stats, version 3.6.1, License: Part of R 3.6.1

Community examples

Looks like there are no examples yet.

Post a new example:

B I H | | | | | | | | | | | |

New example

Use markdown to format your example

R code blocks are runnable and interactive:

```
```r
a <- 2
print(a)
```
```

You can also display normal code blocks

```
```
var a = b
```
```

Submit your example