Title

stata.com

matrix dissimilarity — Compute similarity or dissimilarity measures

Description Syntax Options Remarks and examples References Also see

Description

`matrix dissimilarity` computes a similarity, dissimilarity, or distance matrix.

Syntax

`matrix dissimilarity` *matname* `=` [*varlist*] [*if*] [*in*] [`,` *options*]

options	Description
measure	similarity or dissimilarity measure; default is `L2` (Euclidean)
`observations`	compute similarities or dissimilarities between observations; the default
`variables`	compute similarities or dissimilarities between variables
`names(`*varname*`)`	row/column names for *matname* (allowed with `observations`)
`allbinary`	check that all values are 0, 1, or missing
`proportions`	interpret values as proportions of binary values
`dissim(`*method*`)`	change similarity measure to dissimilarity measure

where *method* transforms similarities to dissimilarities by using

`oneminus`	$d_{ij} = 1 - s_{ij}$
`standard`	$d_{ij} = \sqrt{s_{ii} + s_{jj} - 2s_{ij}}$

Options

measure specifies one of the similarity or dissimilarity measures allowed by Stata. The default is `L2`, Euclidean distance. Many similarity and dissimilarity measures are provided for continuous data and for binary data; see [MV] ***measure_option***.

`observations` and `variables` specify whether similarities or dissimilarities are computed between observations or variables. The default is `observations`.

`names(`*varname*`)` provides row and column names for *matname*. *varname* must be a string variable with a length of 32 or less. You will want to pick a *varname* that yields unique values for the row and column names. Uniqueness of values is not checked by `matrix dissimilarity`. `names()` is not allowed with the `variables` option. The default row and column names when the similarities or dissimilarities are computed between observations is `obs#`, where # is the observation number corresponding to that row or column.

`allbinary` checks that all values are 0, 1, or missing. Stata treats nonzero values as one (excluding missing values) when dealing with what are supposed to be binary data (including binary similarity *measure*s). `allbinary` causes `matrix dissimilarity` to exit with an error message if the values are not truly binary. `allbinary` is not allowed with `proportions` or the `Gower` *measure*.

`proportions` is for use with binary similarity *measure*s. It specifies that values be interpreted as proportions of binary values. The default action treats all nonzero values as one (excluding missing values). With `proportions`, the values are confirmed to be between zero and one, inclusive. See [MV] ***measure_option*** for a discussion of the use of proportions with binary *measure*s. `proportions` is not allowed with `allbinary` or the `Gower` *measure*.

`dissim(`*method*`)` specifies that similarity measures be transformed into dissimilarity measures. *method* may be `oneminus` or `standard`. `oneminus` transforms similarities to dissimilarities by using $d_{ij} = 1 - s_{ij}$ (Kaufman and Rousseeuw 1990, 21). `standard` uses $d_{ij} = \sqrt{s_{ii} + s_{jj} - 2s_{ij}}$ (Mardia, Kent, and Bibby 1979, 402). `dissim()` does nothing when the *measure* is already a dissimilarity or distance. See [MV] ***measure_option*** to see which *measure*s are similarities.

Remarks and examples

stata.com

Commands such as `cluster singlelinkage`, `cluster completelinkage`, and `mds` (see [MV] **cluster** and [MV] **mds**) have options allowing the user to select the similarity or dissimilarity measure to use for its computation. If you are developing a command that requires a similarity or dissimilarity matrix, the `matrix dissimilarity` command provides a convenient way to obtain it.

The similarity or dissimilarity between each observation (or variable if the `variables` option is specified) and the others is placed in *matname*. The element in the *i*th row and *j*th column gives either the similarity or dissimilarity between the *i*th and *j*th observation (or variable). Whether you get a similarity or a dissimilarity depends upon the requested *measure*; see [MV] ***measure_option***.

If there are many observations (variables when the `variables` option is specified), you may need to increase the maximum matrix size; see [R] **matsize**. If the number of observations (or variables) is so large that storing the results in a matrix is not practical, you may wish to consider using the `cluster measures` command, which stores similarities or dissimilarities in variables; see [MV] **cluster programming utilities**.

When computing similarities or dissimilarities between observations, the default row and column names of *matname* are set to `obs#`, where # is the observation number. The `names()` option allows you to override this default. For similarities or dissimilarities between variables, the row and column names of *matname* are set to the appropriate variable names.

The order of the rows and columns corresponds with the order of your observations when you are computing similarities or dissimilarities between observations. Warning: If you reorder your data (for example, using `sort` or `gsort`) after running `matrix dissimilarity`, the row and column ordering will no longer match your data.

Another use of `matrix dissimilarity` is in performing a cluster analysis on variables instead of observations. The `cluster` command performs a cluster analysis of the observations; see [MV] **cluster**. If you instead wish to cluster variables, you can use the `variables` option of `matrix dissimilarity` to obtain a dissimilarity matrix that can then be used with `clustermat`; see [MV] **clustermat** and example 2 below.

▷ Example 1

Example 1 of [MV] **cluster linkage** introduces data with four chemical laboratory measurements on 50 different samples of a particular plant. Let's find the Canberra distance between the measurements performed by lab technician Bill found among the first 25 observations of the labtech dataset.

```
. use http://www.stata-press.com/data/r14/labtech
. matrix dissim D = x1 x2 x3 x4 if labtech=="Bill" in 1/25, canberra
. matrix list D
symmetric D[6,6]
            obs7       obs18       obs20       obs22       obs23       obs25
 obs7          0
obs18  1.3100445           0
obs20  1.1134916   .87626565           0
obs22   1.452748   1.0363077   1.0621064           0
obs23  1.0380665   1.4952796   .81602718   1.6888123           0
obs25  1.4668898   1.5139834   1.4492336   1.0668425   1.1252514           0
```

By default, the row and column names of the matrix indicate the observations involved. The Canberra distance between the 23rd observation and the 18th observation is 1.4952796. See [MV] ***measure_option*** for a description of the Canberra distance.

◁

▷ Example 2

Example 2 of [MV] **cluster linkage** presents a dataset with 30 observations of 60 binary variables, `a1`, `a2`, ..., `a30`. In [MV] **cluster linkage**, the observations were clustered. Here we instead cluster the variables by computing the dissimilarity matrix by using `matrix dissimilarity` with the `variables` option followed by the `clustermat` command.

We use the `matching` option to obtain the simple matching similarity coefficient but then specify `dissim(oneminus)` to transform the similarities to dissimilarities by using the transformation $d_{ij} = 1 - s_{ij}$. The `allbinary` option checks that the variables really are binary (0/1) data.

```
. use http://www.stata-press.com/data/r14/homework
. matrix dissim Avars = a*, variables matching dissim(oneminus) allbinary
. matrix subA = Avars[1..5,1..5]
. matrix list subA
symmetric subA[5,5]
           a1          a2          a3          a4          a5
a1          0
a2         .4           0
a3         .4  .46666667           0
a4         .3          .3  .36666667           0
a5         .4          .4  .13333333          .3           0
```

We listed the first five rows and columns of the 60×60 matrix. The matrix row and column names correspond to the variable names.

To perform an average-linkage cluster analysis on the 60 variables, we supply the `Avars` matrix created by `matrix dissimilarity` to the `clustermat averagelinkage` command; see [MV] **cluster linkage**.

```
. clustermat averagelinkage Avars, clear
number of observations (_N) was 0, now 60
cluster name: _clus_1
. cluster generate g5 = groups(5)
```

```
. table g5
```

g5	Freq.
1	21
2	9
3	25
4	4
5	1

We generated a variable, g5, indicating the five-group cluster solution and then tabulated to show how many variables were clustered into each of the five groups. Group five has only one member.

```
. list g5 if g5==5
```

	g5
13.	5

The member corresponds to the 13th observation in the current dataset, which in turn corresponds to variable a13 from the original dataset. It appears that a13 is not like the other variables.

◁

▷ Example 3

matrix dissimilarity drops observations containing missing values, except when the Gower measure is specified. The computation of the Gower dissimilarity between 2 observations is based on the variables where the 2 observations both have nonmissing values.

We illustrate using a dataset with 6 observations and 4 variables where only 2 of the observations have complete data.

```
. use http://www.stata-press.com/data/r14/gower, clear
. list
```

	b1	b2	x1	x2
1.	0	1	.76	.75
2.	.	.	.	.
3.	1	0	.72	.88
4.	.	1	.4	.
5.	0	.	.	.14
6.	0	0	.55	.

```
. mat dissimilarity matL2 = b* x*, L2
. matlist matL2, format(%8.3f)
```

	obs1	obs3
obs1	0.000	
obs3	1.421	0.000

The resulting matrix is 2×2 and provides the dissimilarity between observations 1 and 3. All other observations contained at least one missing value.

However, with the `gower` measure we obtain a 6×6 matrix.

```
. matrix dissimilarity matgow = b1 b2 x1 x2, gower
. matlist matgow, format(%8.3f)
```

	obs1	obs2	obs3	obs4	obs5	obs6
obs1	0.000					
obs2	.	0.000				
obs3	0.572	.	0.000			
obs4	0.500	.	0.944	0.000		
obs5	0.412	.	1.000	.	0.000	
obs6	0.528	.	0.491	0.708	0.000	0.000

Because all the values for observation 2 are missing, the matrix contains missing values for the dissimilarity between observation 2 and the other observations. Notice the missing value in `matgow` for the dissimilarity between observations 4 and 5. There were no variables where observations 4 and 5 both had nonmissing values, and hence the `Gower` coefficient could not be computed.

◁

References

Kaufman, L., and P. J. Rousseeuw. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley.

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

Also see

[MV] **cluster** — Introduction to cluster-analysis commands

[MV] **clustermat** — Introduction to clustermat commands

[MV] **mdsmat** — Multidimensional scaling of proximity data in a matrix

[MV] **cluster programming utilities** — Cluster-analysis programming utilities

[MV] ***measure_option*** — Option for similarity and dissimilarity measures

[P] **matrix** — Introduction to matrix commands