

## Title

**mds postestimation** — Postestimation tools for mds, mdsmat, and mdslong

## Description

The following postestimation commands are of special interest after `mds`, `mdslong`, and `mdsmat`:

command	description
<code>estat config</code>	coordinates of the approximating configuration
<code>estat correlations</code>	correlations between disparities and distances
<code>estat pairwise</code>	pairwise disparities, distances, and residuals
<code>estat quantiles</code>	quantiles of the residuals per object
<code>estat stress</code>	Kruskal stress (loss) measure (only after classical MDS)
* <code>estat summarize</code>	estimation sample summary
<code>mdsconfig</code>	plot of approximating configuration
<code>mdsshepard</code>	Shepard diagram
<code>screepplot</code>	plot eigenvalues (only after classical MDS)

\* `estat summarize` is not available after `mdsmat`.

For more information on these commands, except `screepplot`, see below. For information on `screepplot`, see [MV] **screepplot**.

The following standard postestimation commands are also available:

command	description
* <code>estimates</code>	cataloging estimation results
<code>predict</code>	approximating configuration, disparities, dissimilarities, distances, and residuals

\* All `estimates` subcommands except `table` and `stats` are available.

See the corresponding entries in the *Stata Base Reference Manual* for more information.

## Special-interest postestimation commands

`estat config` lists the coordinates of the approximating configuration.

`estat correlations` lists the Pearson and Spearman correlations between the disparities or dissimilarities and the Euclidean distances for each object.

`estat pairwise` lists the pairwise statistics: the disparities, the distances, and the residuals.

`estat quantiles` lists the quantiles of the residuals per object.

`estat stress` displays the Kruskal stress (loss) measure between the (transformed) dissimilarities and fitted distances per object (only after classical MDS).

`estat summarize` summarizes the variables in the MDS over the estimation sample. After `mds`, `estat summarize` also reports whether and how variables were transformed before computing similarities or dissimilarities.

`mdsconfig` produces a plot of the approximating Euclidean configuration. By default, dimensions 1 and 2 are plotted.

`mdsshepard` produces a Shepard diagram of the disparities against the Euclidean distances. Ideally, the points in the plot should be close to the  $y = x$  line. Optionally, separate plots are generated for each “row” (value of `id()`).

## Syntax for predict

```
predict [type] {stub*|newvarlist} [if] [in] [, statistic options]
```

<i>statistic</i>	description
Main	
<code>config</code>	approximating configuration; specify <code>dimension()</code> or fewer variables
<code>pairwise(pstats)</code>	selected pairwise statistics; specify same number of variables

<i>pstats</i>	description
<code>disparities</code>	disparities = transformed(dissimilarities)
<code>dissimilarities</code>	dissimilarities
<code>distances</code>	Euclidean distances between configuration points
<code>rresiduals</code>	raw residual = dissimilarity – distance
<code>tresiduals</code>	transformed residual = disparity – distance
<code>weights</code>	weights

<i>options</i>	description
Main	
* <code>saving(filename, replace)</code>	save results to <i>filename</i> ; use <code>replace</code> to overwrite existing <i>filename</i>
<code>full</code>	create predictions for all pairs of objects; <code>pairwise()</code> only

\* `saving()` is required after `mdsmat`, after `mds` if `pairwise()` is selected, and after `mdslong` if `config` is selected.

## Options for predict

Main

`config` generates variables containing the approximating configuration in Euclidean space. Specify as many new variables as approximating dimensions (as determined by the `dimension()` option of `mds`, `mdsmat`, or `mdslong`), though you may specify fewer. `estat config` displays the same information but does not store the information in variables. After `mdsmat` and `mdslong`, you must also specify the `saving()` option.

`pairwise(pstats)` generates new variables containing pairwise statistics. The number of new variables should be the same as the number of specified statistics. The following statistics are allowed:

`disparities` generates the disparities, i.e., the transformed dissimilarities. If no transformation is applied (modern MDS with `transform(identity)`), disparities are the same as dissimilarities.

`dissimilarities` generates the dissimilarities used in MDS. If `mds`, `mdslong`, or `mdsmat` was invoked on similarity data, the associated dissimilarities are returned.

`distances` generates the (unsquared) Euclidean distances between the fitted configuration points.

`rresiduals` generates the raw residuals: dissimilarities – distances.

`tresiduals` generates the transformed residuals: disparities – distances.

`weights` generates the weights. Missing proximities are represented by zero weights.

`estat pairwise` displays some of the same information but does not store the information in variables.

After `mds` and `mdsmat`, you must also specify the `saving()` option. With  $n$  objects, the pairwise dataset has  $n(n-1)/2$  observations. In addition to the three requested variables, `predict` produces variables `id1` and `id2`, which identify pairs of objects. With `mds`, `id` is the name of the identification variable (`id()` option), and with `mdsmat` it is “Category”.

`saving(filename [, replace])` is required after `mdsmat`, after `mds` if `pairwise` is selected, and after `mdslong` if `config` is selected. `saving()` indicates that the generated variables are to be created in a new Stata dataset and saved in the file named `filename`. Unless `saving()` is specified, the variables are generated in the current dataset.

`replace` indicates that `filename` specified in `saving()` may be overwritten.

`full` creates predictions for all pairs of objects ( $j_1, j_2$ ). The default is to generate predictions only for pairs ( $j_1, j_2$ ) where  $j_1 > j_2$ . `full` may be specified only with `pairwise`.

## Syntax for estat

List the coordinates of the approximating configuration

```
estat config [ , maxlength(#) format(%fmt) ]
```

List the Pearson and Spearman correlations

```
estat correlations [ , maxlength(#) format(%fmt) notransform nototal ]
```

List the pairwise statistics: disparities, distances, and residuals

```
estat pairwise [ , maxlength(#) notransform full separator ]
```

List the quantiles of the residuals

```
estat quantiles [ , maxlength(#) format(%fmt) nototal notransform ]
```

Display the Kruskal stress (loss) measure per point (only after classical MDS)

```
estat stress [ , maxlength(#) format(%fmt) nototal notransform ]
```

Summarize the variables in MDS

```
estat summarize [ , labels ]
```

<i>options</i>	<i>description</i>
<u>maxlength</u> (#)	maximum number of characters for displaying object names; default is 12
<u>format</u> (% <i>fnt</i> )	display format
<u>nototal</u>	suppress display of overall summary statistics
<u>notransform</u>	suppress the linear transformation of the dissimilarities
<u>full</u>	display all pairs ( $j_1, j_2$ ); default is ( $j_1 > j_2$ ) only
<u>separator</u>	draw separating lines
<u>labels</u>	display variable labels

## Options for estat

`maxlength(#)`, an option used with all but `estat summarize`, specifies the maximum number of characters of the object names to be displayed; the default is `maxlength(12)`.

`format(%fnt)`, an option used with `estat config`, `estat correlations`, `estat quantiles`, and `estat stress`, specifies the display format; the default differs between the subcommands.

`nototal`, an option used with `estat correlations`, `estat quantiles`, and `estat stress`, suppresses the overall summary statistics.

`notransform`, an option used with `estat correlations`, `estat pairwise`, `estat quantiles`, and `estat stress`, specifies that the untransformed dissimilarities be used instead of the transformed dissimilarities (disparities).

`full`, an option used with `estat pairwise`, displays a row for all pairs ( $j_1, j_2$ ). The default is to display rows only for pairs where  $j_1 > j_2$ .

`separator`, an option used with `estat pairwise`, draws separating lines between blocks of rows corresponding to changes in the first of the pair of objects.

`labels`, an option used with `estat summarize`, displays variable labels.

(Continued on next page)

## Syntax for mdsconfig

```
mdsconfig [ , options ]
```

<i>options</i>	description
Main	
<u>dimensions</u> (# #)	two dimensions to be displayed; default is <code>dimensions(2 1)</code>
<u>xnegate</u>	negate data relative to the <i>x</i> axis
<u>ynegate</u>	negate data relative to the <i>y</i> axis
<u>autoaspect</u>	adjust aspect ratio on the basis of the data; default aspect ratio is 1
<u>maxlength</u> (#)	maximum number of characters used in marker labels
<i>cline_options</i>	affect rendition of the lines connecting points
<i>marker_options</i>	change look of markers (color, size, etc.)
<i>marker_label_options</i>	change look or position of marker labels
Y axis, X axis, Titles, Legend, Overall	
<i>twoway_options</i>	any options other than <code>by()</code> documented in [G] <i>twoway_options</i>

## Options for mdsconfig

Main

`dimensions(# #)` identifies the dimensions to be displayed. For instance, `dimensions(3 2)` plots the third dimension (vertically) versus the second dimension (horizontally). The dimension number cannot exceed the number of extracted dimensions. The default is `dimensions(2 1)`.

`xnegate` specifies that the data be negated relative to the *x* axis.

`ynegate` specifies that the data be negated relative to the *y* axis.

`autoaspect` specifies that the aspect ratio be automatically adjusted based on the range of the data to be plotted. This option can make some plots more readable. By default, `mdsconfig` uses an aspect ratio of one, producing a square plot. Some plots will have little variation in the *y*-axis direction, and use of the `autoaspect` option will better fill the available graph space while preserving the equivalence of distance in the *x* and *y* axes.

As an alternative to `autoaspect`, the *twoway\_option* `aspectratio()` can be used to override the default aspect ratio. `mdsconfig` accepts the `aspectratio()` option as a suggestion only and will override it when necessary to produce plots with balanced axes; i.e., distance on the *x* axis equals distance on the *y* axis.

*twoway\_options*, such as `xlabel()`, `xscale()`, `ylabel()`, and `yscale()`, should be used with caution. These options are accepted but may have unintended side effects on the aspect ratio.

`maxlength(#)` specifies the maximum number of characters for object names used to mark the points; the default is `maxlength(12)`.

*cline\_options* affect the rendition of the lines connecting the plotted points; see [G] *cline\_options*. If you are drawing connected lines, the appearance of the plot depends on the sort order of the data.

*marker\_options* affect the rendition of the markers drawn at the plotted points, including their shape, size, color, and outline; see [G] *marker\_options*.

*marker\_label\_options* specify if and how the markers are to be labeled; see [G] *marker\_label\_options*.

Y axis, X axis, Titles, Legend, Overall

*twoway\_options* are any of the options documented in [G] *twoway\_options*, excluding `by()`. These include options for titling the graph (see [G] *title\_options*) and for saving the graph to disk (see [G] *saving\_option*). See `autoaspect` above for a warning against using options such as `xlabel()`, `xscale()`, `ylabel()`, and `yscale()`.

## Syntax for `mdsshepard`

```
mdsshepard [ , options ]
```

<i>options</i>	description
Main	
<code>notransform</code>	use dissimilarities instead of disparities
<code>autoaspect</code>	adjust aspect ratio on the basis of the data; default aspect ratio is 1
<code>separate</code>	draw separate Shepard diagrams for each object
<i>marker_options</i>	change look of markers (color, size, etc.)
Y axis, X axis, Titles, Legend, Overall	
<i>twoway_options</i>	any options other than <code>by()</code> documented in [G] <i>twoway_options</i>
<code>byopts(<i>by_option</i>)</code>	affect the rendition of combined graphs; <code>separate</code> only

## Options for `mdsshepard`

Main

`notransform` uses dissimilarities instead of disparities, i.e., suppresses the transformation of the dissimilarities.

`autoaspect` specifies that the aspect ratio be automatically adjusted based on the range of the data to be plotted. By default, `mdsshepard` uses an aspect ratio of one, producing a square plot.

See the description of the `autoaspect` option of `mdsconfig` for more details.

`separate` displays separate plots of each value of the ID variable. This may be time consuming if the number of distinct ID values is not small.

*marker\_options* affect the rendition of the markers drawn at the plotted points, including their shape, size, color, and outline; see [G] *marker\_options*.

Y axis, X axis, Titles, Legend, Overall

*twoway\_options* are any of the options documented in [G] *twoway\_options*, excluding `by()`. These include options for titling the graph (see [G] *title\_options*) and for saving the graph to disk (see [G] *saving\_option*). See the `autoaspect` option of `mdsconfig` for a warning against using options such as `xlabel()`, `xscale()`, `ylabel()`, and `yscale()`.

`byopts(by_option)` is documented in [G] *by\_option*. This option affects the appearance of the combined graph and is allowed only with the `separate` option.

## Remarks

Remarks are presented under the following headings:

*Postestimation statistics*  
*Matching configuration plot and the Shepard diagram*  
*Predictions*

## Postestimation statistics

After an MDS analysis, several facilities can help you better understand the analysis and, in particular, to assess the quality of the lower-dimensional Euclidean representation. We display results after classical MDS. All are available after modern MDS except for `estat stress`.

### ▷ Example 1

We illustrate the MDS postestimation facilities with the Morse code digit-similarity dataset; see example 1 in [MV] `mdslong`.

```
. use http://www.stata-press.com/data/r10/morse_long
(Morse data (Rothkopf 1957))
. gen sim = freqsame/100
. mdslong sim, id(digit1 digit2) s2d(standard) noplot
(output omitted)
```

MDS has produced a two-dimensional configuration with Euclidean distances approximating the dissimilarities between the Morse codes for digits. This configuration may be examined using the `estat config` command; see `mdsconfig` if you want to plot the configuration.

```
. estat config
Approximating configuration in 2-dimensional Euclidean space
```

digit1	dim1	dim2
0	0.5690	-0.0162
1	0.4561	0.3384
2	0.0372	0.5854
3	-0.3878	0.4516
4	-0.5800	0.0770
5	-0.5458	0.0196
6	-0.3960	-0.4187
7	-0.0963	-0.5901
8	0.3124	-0.3862
9	0.6312	-0.0608

This configuration is not unique. A translation, a reflection, and an orthonormal rotation of the configuration do not affect the interpoint Euclidean distances. All such transformations are equally reasonable MDS solutions. Thus you should not interpret aspects of these numbers (or of the configuration plot) that are not invariant to these transformations.

We now turn to the three `estat` subcommands that analyze the MDS residuals, i.e., the differences between the disparities or dissimilarities and the matching Euclidean distances. There is a catch here. The raw residuals of MDS are not well behaved. For instance, the sum of the raw residuals is not zero—often it is not even close. The MDS solution does *not* minimize the sum of squares of the raw residuals (Mardia, Kent, and Bibby 1979, 406–408). To create reasonable residuals with MDS, the dissimilarities can be transformed to disparities approximating the Euclidean distances. In classical MDS we use a linear transform  $f$ , fitted by least squares. This is equivalent to Kruskal's Stress1 loss function. The modified residuals are defined as the differences between the linearly transformed dissimilarities and the matching Euclidean distances.

In modern MDS we have three types of transformations from dissimilarities to disparities to choose from: the identity (which does not transform the dissimilarities), a power transformation, and a monotonic transformation.

The three `estat` subcommands summarize the residuals in different ways. After classical MDS, `estat stress` displays the Kruskal loss or stress measures for each object and the overall total.

```
. estat stress
Stress between adjusted dissimilarities and Euclidean distances
```

digit1	Kruskal
0	0.1339
1	0.1255
2	0.1972
3	0.2028
4	0.2040
5	0.2733
6	0.1926
7	0.1921
8	0.1715
9	0.1049
Total	0.1848

Second, after classical or modern MDS, the quantiles of the residuals are available, both overall and for the subgroup of object pairs in which an object is involved.

```
. estat quantiles
Quantiles of adjusted residuals
```

digit1	N	min	p25	q50	q75	max
0	9	-.111732	-.088079	-.028917	.11202	.220399
1	9	-.170063	-.137246	-.041244	.000571	.11202
2	9	-.332717	-.159472	-.072359	.074999	.234866
3	9	-.136251	-.120398	-.072359	.105572	.365833
4	9	-.160797	-.014099	.03845	.208215	.355053
5	9	-.09971	-.035357	.176337	.325043	.365833
6	9	-.137246	-.113564	-.075008	.177448	.325043
7	9	-.332717	-.170063	-.124129	.03845	.176337
8	9	-.186452	-.134831	-.041244	.075766	.220399
9	9	-.160797	-.104403	-.088079	-.064316	-.030032
Total	90	-.332717	-.113564	-.041244	.105572	.365833

The dissimilarities for the Morse code of digit 5 are fitted considerably worse than for all other Morse codes. Digit 5 has the largest Kruskal stress (0.273) and median residual (0.176).

*(Continued on next page)*

Finally, after classical or modern MDS, `estat correlations` displays the Pearson and Spearman correlations between the (transformed or untransformed) dissimilarities and the Euclidean distances.

```
. estat correlations
Correlations of dissimilarities and fitted distances
```

digit1	N	Pearson	Spearman
0	9	0.9510	0.9540
1	9	0.9397	0.7782
2	9	0.7674	0.4017
3	9	0.7922	0.7815
4	9	0.9899	0.9289
5	9	0.9412	0.9121
6	9	0.8226	0.8667
7	9	0.8444	0.4268
8	9	0.8505	0.7000
9	9	0.9954	0.9333
Total	90	0.8602	0.8301

◀

## Matching configuration plot and the Shepard diagram

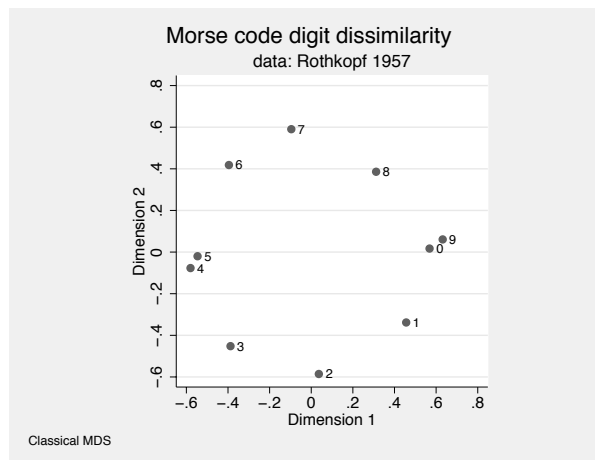
The matching configuration plot and Shepard diagram are easily obtained after an MDS analysis.

### ▷ Example 2

By default, `mds`, `mdsmat`, and `mdslong` display the MDS matching configuration plot. If you want to exercise control over the graph, you can specify the `noplot` option of `mds`, `mdsmat`, or `mdslong` and then use the `mdsconfig` postestimation graph command.

Continuing with the Morse code digit example: we produce a configuration plot with an added title and subtitle.

```
. mdsconfig, title(Morse code digit dissimilarity) subtitle(data: Rothkopf 1957)
```



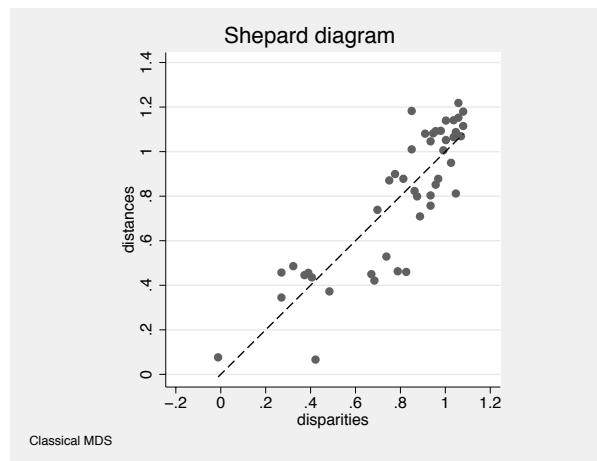
The plot has an aspect ratio of one so that 1 unit on the horizontal dimension equals 1 unit on the vertical dimension. Thus the “straight-line” distance in the plot is really (proportional to) the Euclidean distance between the points in the configuration and hence approximates the dissimilarities between the objects—here the Morse codes for digits.

◀

### ▶ Example 3

A second popular plot for MDS is the Shepard diagram. This is a plot of the Euclidean distances in the matching configuration against the “observed” dissimilarities. As we explained before, in classical MDS a linear transformation is applied to the dissimilarities to fit the Euclidean distances as close as possible (in the least-squares sense). In modern MDS the transformation may be the identity (no transformation), a power function, or a monotonic function. A Shepard diagram is a plot of the  $n(n-1)/2$  transformed dissimilarities, called disparities, against the Euclidean distances.

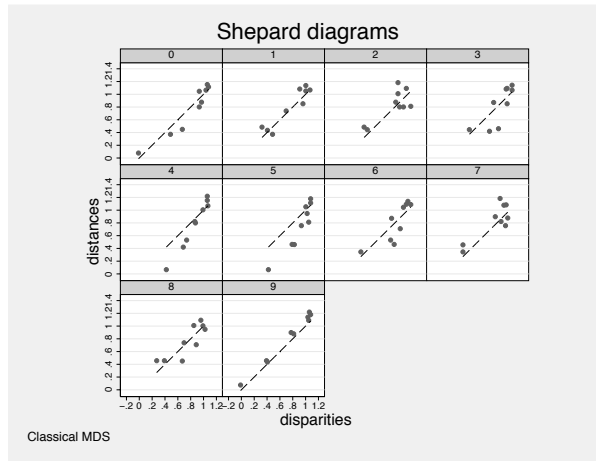
```
. mdsshepard
```



If the Euclidean configuration is close to the disparities between the objects, all points would be close to the  $y = x$  line. Deviations indicate lack of fit. To simplify the diagnosis of whether there are specific objects that are poorly represented, Shepard diagrams can be produced for each object separately. Such plots consist of  $n$  small plots with  $n - 1$  points each, namely, the disparities and Euclidean distances to all other objects.

(Continued on next page)

```
. mds Shepard, separate
(mds Shepard is producing a separate plot for each obs; this may take a while)
```



◀

Other examples of `mdsconfig` are found in [MV] `mds`, [MV] `mdslong`, and [MV] `mdsmat`.

## Predictions

It is possible to generate variables containing the results from the MDS analysis. MDS operates at two levels: first at the level of the objects and second at the level of relations between the objects or pairs of objects. You can generate variables at both of these levels.

The `config` option of `predict` after an MDS requests that the coordinates of the objects in the matching configuration be stored in variables. You may specify as many variables as there are retained dimensions. You may also specify fewer variables. The first variable will store the coordinates from the first dimension. The second variable, if any, will store the coordinates from the second dimension, and so on.

The `pairwise()` option specifies that a given selection of the pairwise statistics are stored in variables. The statistics available are the disparities, dissimilarities, fitted distances, raw residuals, transformed residuals, and weights. The raw residuals are the difference between dissimilarities and the fitted distances, and the transformed residuals are the difference between the disparities and the fitted distances.

There is a complicating issue. With  $n$  objects, there are  $n(n - 1)/2$  pairs of objects. So, to store properties of objects, you need  $n$  observations, but to store properties of pairs of objects, you need  $n(n - 1)/2$  observations. So, where do you store the variables? `predict` after MDS commands can save the predicted variables in a new dataset. Specify the option `saving(filename)`. Such a dataset will automatically have the appropriate object identification variable or variables.

Sometimes it is also possible to store the variables in the dataset you have in memory: object-level variables in an object-level dataset and pairwise-level variables in a pairwise-level dataset.

After `mds` you have a dataset in memory in which the observations are associated with the MDS objects. Here you can store object-level variables directly in the dataset in memory. To do so, you just omit the `saving()` option. Here it is not possible to store the pairwise statistics in the dataset in memory. The pairwise statistics have to be stored in a new dataset.

After `mdslong`, the dataset in memory is in a pairwise form, so the variables predicted with the option `pairwise()` can be stored in the dataset in memory. It is, of course, also possible to store the pairwise variables in a new dataset; the choice is yours. With pairwise data in memory, you cannot store the object-level predicted variables into the data in memory; you need to specify the name of a new dataset.

After `mdsmat`, you always need to save the predicted variables in a new dataset.

▷ Example 4

Continuing with our Morse code example that used `mdslong`: the dataset in memory is in long form. Thus we can store the pairwise statistics with the dataset in memory.

```
. predict tdisssim eudist resid, pairwise
. list in 1/10
```

	digit1	digit2	freqsame	sim	tdisssim	eudist	resid
1.	2	1	62	.62	.3227682	.4862905	-.1635224
2.	3	1	16	.16	.957076	.851504	.1055719
3.	3	2	59	.59	.3732277	.4455871	-.0723594
4.	4	1	6	.06	1.069154	1.068583	.0005709
5.	4	2	23	.23	.8745967	.7995979	.0749989
6.	4	3	38	.38	.6841489	.4209922	.2631567
7.	5	1	12	.12	1.002667	1.051398	-.048731
8.	5	2	8	.08	1.047234	.8123672	.2348665
9.	5	3	27	.27	.8257753	.4599419	.3658335
10.	5	4	56	.56	.4218725	.0668193	.3550532

Since we used `mdslong`, the object-level statistics must be saved in a file.

```
. predict d1 d2, config saving(digitdata)
. describe digit* d1 d2 using digitdata
```

Contains data

```
obs:          10
vars:         3
size:        130
```

```
unit statistics for MDS
(method=classical,dim=2)
1 Mar 2007 10:07
```

variable name	storage type	display format	value label	variable label
digit1	str1	%9s		
d1	float	%9.0g		MDS dimension 1
d2	float	%9.0g		MDS dimension 2

Sorted by: digit1

The information in these variables was already shown with `estat config`. The dataset created has variables `d1` and `d2` with the coordinates of the Morse digits on the two retained dimensions and an identification variable `digit1`. Use `merge` to add these variables to the data in memory; see [D] `merge`.

## Saved Results

`estat correlations` saves the following in `r()`:

Matrices

`r(R)` statistics per object; columns with # of obs., Pearson corr., and Spearman corr.  
`r(T)` overall statistics; # of obs., Pearson corr., and Spearman corr.

`estat quantiles` saves the following in `r()`:

Macros

`r(dtype)` `adjusted` or `raw`; dissimilarity transformation

Matrices

`r(Q)` statistics per object; columns with # of obs., min., p25, p50, p75, and max.  
`r(T)` overall statistics; # of obs., min., p25, p50, p75, and max.

`estat stress` saves the following in `r()`:

Macros

`r(dtype)` `adjusted` or `raw`; dissimilarity transformation

Matrices

`r(S)` Kruskal's stress/loss measure per object  
`r(T)`  $1 \times 1$  matrix with the overall Kruskal stress/loss measure

## Methods and Formulas

All postestimation commands listed above are implemented as ado-files.

See [MV] **mdsmat** for information on the methods and formulas for multidimensional scaling.

For classical MDS, let  $D_{ij}$  be the dissimilarity between objects  $i$  and  $j$ ,  $1 \leq i, j \leq n$ . We assume  $D_{ii} = 0$  and  $D_{ij} = D_{ji}$ . Let  $E_{ij}$  be the Euclidean distance between rows  $i$  and  $j$  of the matching configuration  $\mathbf{Y}$ . In classical MDS,  $\mathbf{D} - \mathbf{E}$  is not a well-behaved residual matrix. We follow the approach used in metric and nonmetric MDS to transform  $D_{ij}$  to “optimally match”  $E_{ij}$ , with  $\hat{D}_{ij} = a + bD_{ij}$ , where  $a$  and  $b$  are chosen to minimize the residual sum of squares. This is a simple regression problem and is equivalent to minimizing Kruskal's stress measure (Kruskal 1964; Cox and Cox 2001, 63)

$$\text{Kruskal}(\hat{\mathbf{D}}, \mathbf{E}) = \left\{ \frac{\sum (E_{ij} - \hat{D}_{ij})^2}{\sum E_{ij}^2} \right\}^{1/2}$$

with summation over all pairs  $(i, j)$ . We call the  $\hat{D}_{ij}$  the adjusted or transformed dissimilarities. If the transformation step is skipped by specifying the option `notransform`, we set  $\hat{D}_{ij} = D_{ij}$ .

In `estat stress`, the decomposition of Kruskal's stress measure over the objects is displayed.  $\text{Kruskal}(\hat{\mathbf{D}}, \mathbf{E})_i$  is defined analogously with summation over all  $j \neq i$ .

For modern MDS, the optimal transformation to disparities,  $f(\mathbf{D}) \rightarrow \hat{\mathbf{D}}$ , is calculated during the estimation. See [MV] **mdsmat** for details. For `transform(power)` the power is saved in `e(alpha)`. For `transform(monotonic)` the disparities themselves are saved as `e(Disparities)`.

## References

- Borg, I., and P. J. F. Groenen. 2005. *Modern Multidimensional Scaling: Theory and Applications*. 2nd ed. New York: Springer.
- Cox, T. F., and M. A. A. Cox. 2001. *Multidimensional Scaling*. 2nd ed. Boca Raton, FL: Chapman & Hall/CRC.
- Kruskal, J. B. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29: 1–27.
- Mardia, K. V., J. T. Kent, and J. M. Bibby. 1979. *Multivariate Analysis*. New York: Academic Press.
- See [MV] **mdsmat** for additional references.

## Also See

- [MV] **mds** — Multidimensional scaling for two-way data
- [MV] **mdslong** — Multidimensional scaling of proximity data in long format
- [MV] **mdsmat** — Multidimensional scaling of proximity data in a matrix
- [MV] **screepplot** — Scree plot