Title

stata.com

corr2data — Create dataset with specified correlation structure

Description

`corr2data` adds new variables with specified covariance (correlation) structure to the existing dataset or creates a new dataset with a specified covariance (correlation) structure. Singular covariance (correlation) structures are permitted. The purpose of this is to allow you to perform analyses from summary statistics (correlations/covariances and maybe the means) when these summary statistics are all you know and summary statistics are sufficient to obtain results. For example, these summary statistics are sufficient for performing analysis of t tests, variance, principal components, regression, and factor analysis. The recommended process is

```
. clear                                  (clear memory)
. corr2data ..., n(#) cov(...) ...       (create artificial data)
. regress ...                            (use artificial data appropriately)
```

However, for factor analyses and principal components, the commands `factormat` and `pcamat` allow you to skip the step of using `corr2data`; see [MV] **factor** and [MV] **pca**.

The data created by `corr2data` are artificial; they are not the original data, and it is not a sample from an underlying population with the summary statistics specified. See [D] **drawnorm** if you want to generate a random sample. In a sample, the summary statistics will differ from the population values and will differ from one sample to the next.

The dataset `corr2data` creates is suitable for one purpose only: performing analyses when all that is known are summary statistics and those summary statistics are sufficient for the analysis at hand. The artificial data tricks the analysis command into producing the desired result. The analysis command, being by assumption only a function of the summary statistics, extracts from the artificial data the summary statistics, which are the same summary statistics you specified, and then makes its calculation based on those statistics.

If you doubt whether the analysis depends only on the specified summary statistics, you can generate different artificial datasets by using different seeds of the random-number generator (see the `seed()` option below) and compare the results, which should be the same within rounding error.

Quick start

Create dataset with 1,000 observations, `v1` with mean of 3.4 and std. dev. of 1, `v2` with mean of 3 and std. dev. of 0.5, and no correlation between `v1` and `v2`

```
corr2data v1 v2, n(1000) means(3.4 3) sds(1 .5)
```

As above, but with correlation between `v1` and `v2` specified in matrix `mymat`

```
corr2data v1 v2, n(1000) means(3.4 3) sds(1 .5) corr(mymat)
```

Menu

Data > Create or change data > Other variable-creation commands > Create dataset with specified correlation

Syntax

```
corr2data newvarlist [ , options ]
```

options	Description
Main	
`clear`	replace the current dataset
`double`	generate variable type as `double`; default is `float`
`n(#)`	# of observations to be generated; default is current number
`sds(`*vector*`)`	standard deviations of generated variables
`corr(`*matrix*`|`*vector*`)`	correlation matrix
`cov(`*matrix*`|`*vector*`)`	covariance matrix
`cstorage(full)`	correlation/covariance structure is stored as a symmetric $k \times k$ matrix
`cstorage(lower)`	correlation/covariance structure is stored as a lower triangular matrix
`cstorage(upper)`	correlation/covariance structure is stored as an upper triangular matrix
`forcepsd`	force the covariance/correlation matrix to be positive semidefinite
`means(`*vector*`)`	means of generated variables; default is `means(0)`
Options	
`seed(#)`	seed for random-number generator

Options

Main

`clear` specifies that it is okay to replace the dataset in memory, even though the current dataset has not been saved on disk.

`double` specifies that the new variables be stored as Stata `doubles`, meaning 8-byte reals. If `double` is not specified, variables are stored as `floats`, meaning 4-byte reals. See [D] **data types**.

`n(#)` specifies the number of observations to be generated; the default is the current number of observations. If `n(#)` is not specified or is the same as the current number of observations, `corr2data` adds the new variables to the existing dataset; otherwise, `corr2data` replaces the dataset in memory.

`sds(`*vector*`)` specifies the standard deviations of the generated variables. `sds()` may not be specified with `cov()`.

`corr(`*matrix*`|`*vector*`)` specifies the correlation matrix. If neither `corr()` nor `cov()` is specified, the default is orthogonal data.

`cov(`*matrix*`|`*vector*`)` specifies the covariance matrix. If neither `corr()` nor `cov()` is specified, the default is orthogonal data.

`cstorage(full | lower | upper)` specifies the storage mode for the correlation or covariance structure in `corr()` or `cov()`. The following storage modes are supported:

`full` specifies that the correlation or covariance structure is stored (recorded) as a symmetric $k \times k$ matrix.

`lower` specifies that the correlation or covariance structure is recorded as a lower triangular matrix. With k variables, the matrix should have $k(k+1)/2$ elements in the following order:

$$C_{11}\ C_{21}\ C_{22}\ C_{31}\ C_{32}\ C_{33}\ \ldots\ C_{k1}\ C_{k2}\ \ldots\ C_{kk}$$

`upper` specifies that the correlation or covariance structure is recorded as an upper triangular matrix. With k variables, the matrix should have $k(k+1)/2$ elements in the following order:

$$C_{11}\ C_{12}\ C_{13}\ \ldots\ C_{1k}\ C_{22}\ C_{23}\ \ldots C_{2k}\ \ldots\ C_{(k-1k-1)}\ C_{(k-1k)}\ C_{kk}$$

Specifying `cstorage(full)` is optional if the matrix is square. `cstorage(lower)` or `cstorage(upper)` is required for the vectorized storage methods. See *Storage modes for correlation and covariance matrices* in [D] **drawnorm** for examples.

`forcepsd` modifies the matrix *C* to be positive semidefinite (psd) and to thus be a proper covariance matrix. If *C* is not positive semidefinite, it will have negative eigenvalues. By setting the negative eigenvalues to 0 and reconstructing, we obtain the least-squares positive-semidefinite approximation to *C*. This approximation is a singular covariance matrix.

`means(`*vector*`)` specifies the means of the generated variables. The default is `means(0)`.

Options

`seed(#)` specifies the seed of the random-number generator used to generate data. # defaults to 0. The random numbers generated inside `corr2data` do not affect the seed of the standard random-number generator.

Remarks and examples

stata.com

`corr2data` is designed to enable analyses of correlation (covariance) matrices by commands that expect variables rather than a correlation (covariance) matrix. `corr2data` creates variables with exactly the correlation (covariance) that you want to analyze. Apart from means and covariances, all aspects of the data are meaningless. Only analyses that depend on the correlations (covariances) and means produce meaningful results. Thus you may perform a paired t test ([R] **ttest**) or an ordinary regression analysis ([R] **regress**), etc.

If you are not sure that a statistical result depends only on the specified summary statistics and not on other aspects of the data, you can generate different datasets, each having the same summary statistics but other different aspects, by specifying the `seed()` option. If the statistical results differ beyond what is attributable to roundoff error, then using `corr2data` is inappropriate.

Example 1

We first run a regression using the auto dataset.

```
. use http://www.stata-press.com/data/r15/auto
(1978 Automobile Data)
. regress weight length trunk
```

Source	SS	df	MS	Number of obs	=	74
				F(2, 71)	=	303.95
Model	39482774.4	2	19741387.2	Prob > F	=	0.0000
Residual	4611403.95	71	64949.3513	R-squared	=	0.8954
				Adj R-squared	=	0.8925
Total	44094178.4	73	604029.841	Root MSE	=	254.85

weight	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
length	33.83435	1.949751	17.35	0.000	29.94666	37.72204
trunk	-5.83515	10.14957	-0.57	0.567	-26.07282	14.40252
_cons	-3258.84	283.3547	-11.50	0.000	-3823.833	-2693.846

Suppose that, for some reason, we no longer have the auto dataset. Instead, we know the means and covariance matrices of weight, length, and trunk, and we want to do the same regression again. The matrix of means is

```
. matrix list M
M[1,3]
          weight     length      trunk
_cons  3019.4595  187.93243  13.756757
```

and the covariance matrix is

```
. matrix list V
symmetric V[3,3]
           weight     length      trunk
weight  604029.84
length  16370.922  495.78989
 trunk  2234.6612  69.202518  18.296187
```

To do the regression analysis in Stata, we need to create a dataset that has the specified correlation structure.

```
. corr2data x y z, n(74) cov(V) means(M)
. regress weight length trunk
```

Source	SS	df	MS	Number of obs	=	74
				F(2, 71)	=	303.95
Model	39482774.4	2	19741387.2	Prob > F	=	0.0000
Residual	4611403.95	71	64949.3513	R-squared	=	0.8954
				Adj R-squared	=	0.8925
Total	44094178.4	73	604029.841	Root MSE	=	254.85

weight	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
length	33.83435	1.949751	17.35	0.000	29.94666	37.72204
trunk	-5.83515	10.14957	-0.57	0.567	-26.07282	14.40252
_cons	-3258.84	283.3547	-11.50	0.000	-3823.833	-2693.846

The results from the regression based on the generated data are the same as those based on the real data.

◁

Methods and formulas

Two steps are involved in generating the desired dataset. The first step is to generate a zero-mean, zero-correlated dataset. The second step is to apply the desired correlation structure and the means to the zero-mean, zero-correlated dataset. In both steps, we take into account that, given any matrix $\mathbf{A}$ and any vector of variables $\mathbf{X}$, $\mathrm{Var}(\mathbf{A}'\mathbf{X}) = \mathbf{A}'\mathrm{Var}(\mathbf{X})\mathbf{A}$.

Reference

Cappellari, L., and S. P. Jenkins. 2006. Calculation of multivariate normal probabilities by simulation, with applications to maximum simulated likelihood estimation. *Stata Journal* 6: 156–189.

Also see

[D] **data types** — Quick reference for data types

[D] **drawnorm** — Draw sample from multivariate normal distribution