drawnorm — Draw sample from multivariate normal distribution

Description

drawnorm draws a sample from a multivariate normal distribution with desired means and covariance matrix. The default is orthogonal data with mean 0 and variance 1. The covariance matrix may be singular. The values generated are a function of the current random-number seed or the number specified with set seed(); see [R] set seed.

Quick start

Generate independent variables x and y, where x has mean 2 and standard deviation 0.5 and y has mean 3 and standard deviation 1

drawnorm x y, means(2,3) sds(.5,1)

As above, but create dataset of 1,000 observations on x and y with means stored in vector m and standard deviations stored in vector sd

drawnorm x y, means(m) sds(sd) n(1000)

As above, and set the seed for the random-number generator to reproduce results

drawnorm x y, means(m) sds(sd) n(1000) seed(81625)

Sample from bivariate standard normal distribution with covariance between x and y of 0.5 stored in variance–covariance matrix C

matrix C = (1, .5 \ .5, 1)
drawnorm x y, cov(C)

Sample from a trivariate standard normal distribution with correlation between x and y of 0.4, x and z of 0.3, and y and z of 0.6 stored in correlation matrix C

matrix C = (1, .4, .3 \ .4, 1, .6 \ .3, .6, 1)
drawnorm x y z, corr(C)

Same as above, but avoid typing full matrix by specifying correlations in vector v treated as a lower triangular matrix

matrix v = (1, .4, 1, .3, .6, 1)
drawnorm x y z, corr(v) cstorage(lower)

Menu

Data > Create or change data > Other variable-creation commands > Draw sample from normal distribution
## Syntax

```
drawnorm newvarlist [ , options ]
```

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

## Options

- **clear** specifies that the dataset in memory be replaced, 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, `drawnorm` adds the new variables to the existing dataset; otherwise, `drawnorm` replaces the data 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 cov() nor corr() 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:

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

Specifying cstorage(full) is optional if the matrix is square. cstorage(lower) or cstorage(upper) is required for the vectorized storage methods. See Example 2: Storage modes for correlation and covariance matrices.

forcepsd modifies the matrix $C$ to be positive semidefinite (psd), and so be a proper covariance matrix. If $C$ is not positive semidefinite, it will have negative eigenvalues. By setting 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).

seed(#) specifies the initial value of the random-number seed used by the runiform() function. The default is the current random-number seed. Specifying seed(#) is the same as typing set seed # before issuing the drawnorm command.

Remarks and examples

Example 1

Suppose that we want to draw a sample of 1,000 observations from a normal distribution $N(M, V)$, where $M$ is the mean matrix and $V$ is the covariance matrix:

```
. matrix M = 5, -6, 0.5
. matrix V = (9, 5, 2 \ 5 , 4 , 1 \ 2, 1, 1)
. matrix list M
    M[1,3]      c1   c2   c3
          r1  5  -6  .5
. matrix list V
    symmetric V[3,3]      c1   c2   c3
          r1  9
          r2  5  4
          r3  2  1  1
. drawnorm x y z, n(1000) cov(V) means(M)
    (obs 1,000)
. summarize
```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>1,000</td>
<td>4.976214</td>
<td>3.012997</td>
<td>-5.471216</td>
<td>13.55205</td>
</tr>
<tr>
<td>y</td>
<td>1,000</td>
<td>-6.001922</td>
<td>2.013565</td>
<td>-11.84907</td>
<td>-.5673919</td>
</tr>
<tr>
<td>z</td>
<td>1,000</td>
<td>.4638996</td>
<td>1.039727</td>
<td>-3.612451</td>
<td>3.762798</td>
</tr>
</tbody>
</table>
Technical note

The values generated by \texttt{drawnorm} are a function of the current random-number seed. To reproduce the same dataset each time \texttt{drawnorm} is run with the same setup, specify the same seed number in the \texttt{seed()} option.

Example 2: Storage modes for correlation and covariance matrices

The three storage modes for specifying the correlation or covariance matrix in \texttt{corr2data} and \texttt{drawnorm} can be illustrated with a correlation structure, $C$, of 4 variables. In full storage mode, this structure can be entered as a $4 \times 4$ Stata matrix:

\begin{verbatim}
. matrix C = ( 1.0000, 0.3232, 0.1112, 0.0066 \ ///
 0.3232, 1.0000, 0.6608, -0.1572 \ ///
 0.1112, 0.6608, 1.0000, -0.1480 \ ///
 0.0066, -0.1572, -0.1480, 1.0000 )
\end{verbatim}

Elements within a row are separated by commas, and rows are separated by a backslash, \\). We use the input continuation operator /// for convenient multiline input; see [P] comments. In this storage mode, we probably want to set the row and column names to the variable names:

\begin{verbatim}
. matrix rownames C = price trunk headroom rep78
. matrix colnames C = price trunk headroom rep78
\end{verbatim}

This correlation structure can be entered more conveniently in one of the two vectorized storage modes. In these modes, we enter the lower triangle or the upper triangle of $C$ in rowwise order; these two storage modes differ only in the order in which the $k(k+1)/2$ matrix elements are recorded. The lower storage mode for $C$ comprises a vector with $4(4+1)/2 = 10$ elements, that is, a $1 \times 10$ or $10 \times 1$ Stata matrix, with one row or column,

\begin{verbatim}
. matrix C = ( 1.0000, 0.3232, 0.1112, 0.0066, \\
 0.3232, 1.0000, 0.6608, -0.1572, \\
 0.1112, 0.6608, 1.0000, -0.1480, \\
 0.0066, -0.1572, -0.1480, 1.0000 )
\end{verbatim}

or more compactly as

\begin{verbatim}
. matrix C = ( 1, 0.3232, 1, 0.1112, 0.6608, 1, 0.0066, -0.1572, -0.1480, 1 )
\end{verbatim}

$C$ may also be entered in upper storage mode as a vector with $4(4+1)/2 = 10$ elements, that is, a $1 \times 10$ or $10 \times 1$ Stata matrix,

\begin{verbatim}
. matrix C = ( 1.0000, 0.3232, 0.1112, 0.0066, 1.0000, 0.6608, -0.1572, \\
 1.0000, -0.1480, 1.0000 )
\end{verbatim}
or more compactly as

\[
\text{matrix } C = ( 1, 0.3232, 0.1112, 0.0066, 1, 0.6608, -0.1572, 1, -0.1480, 1 )
\]

## Methods and formulas

Results are asymptotic. The more observations generated, the closer the correlation matrix of the dataset is to the desired correlation structure.

Let \( V = A' A \) be the desired covariance matrix and \( M \) be the desired mean matrix. We first generate \( X \), such that \( X \sim N(0, I) \). Let \( Y = A' X + M \), then \( Y \sim N(M, V) \).

## References


## Also see

[D] **corr2data** — Create dataset with specified correlation structure  
[R] **set seed** — Specify random-number seed and state