

mvnormal() — Compute multivariate normal distributions and derivatives

Description Conformability	Syntax References	Remarks and examples Also see
-------------------------------	----------------------	----------------------------------

Description

`mvnormal(U , R)` returns the cumulative multivariate normal distributions with lower limits $-\infty$, upper limits U , and vectorized correlation matrices R (only the lower halves are recorded).

`mvnormal(L , U , R)` returns the multivariate normal distributions with lower limits L , upper limits U , and vectorized correlation matrices R (only the lower halves are recorded).

`mvnormalcv(L , U , M , V)` returns the multivariate normal distributions with lower limits L , upper limits U , means M , and vectorized covariance matrices V (only the lower halves are recorded).

`mvnormalderiv(U , R , dU , dR)` computes the derivatives dU and dR of cumulative multivariate normal distributions `mvnormal(U , R)` with upper limits U (lower limits $-\infty$) and vectorized correlation matrices R (only the lower halves are recorded), respectively, with no return values.

`mvnormalderiv(L , U , R , dL , dU , dR)` computes the derivatives dL , dU , and dR of multivariate normal distributions `mvnormal(L , U , R)` with lower limits L , upper limits U , and vectorized correlation matrices R (only the lower halves are recorded), respectively, with no return values.

`mvnormalcvderiv(L , U , M , V , dL , dU , dM , dV)` computes the derivatives dL , dU , dM , and dV of multivariate normal distributions `mvnormalcv(L , U , M , V)` with lower limits L , upper limits U , means M , and vectorized covariance matrices V (only the lower halves are recorded), respectively, with no return values.

`mvnormal()`, `mvnormalcv()`, `mvnormalderiv()`, and `mvnormalcvderiv()` use 128 quadrature points by default for dimensions greater than 3 and use 10 quadrature points by default for dimensions less than or equal to 3.

`mvnormalqp(U , R , q)`, `mvnormalqp(L , U , R , q)`, `mvnormalcvqp(L , U , M , V , q)`, `mvnormalderivqp(U , R , dU , dR , q)`, `mvnormalderivqp(L , U , R , dL , dU , dR , q)` and `mvnormalcvderivqp(L , U , M , V , dL , dU , dM , dV , q)` do the same things except that, rather than using the default number of quadrature points in the calculations, they allow you to specify the number of quadrature points in q . q must be between 3 and 5,000. If q is not an integer, the integer part of q is taken as the number of quadrature points.

Note that `mvnormal()`, `mvnormalcv()`, `mvnormalqp()`, and `mvnormalcvqp()` use `normal()` when the dimension equals 1 and `binormal()` when the dimension equals 2, in which cases the number of quadrature points is irrelevant. The computation of the cumulative multivariate normal distributions and their derivatives can be slow when the dimension is large.

The matrices L , U , M , R , and V parameterize the integrals that are approximated and returned in a column vector. See [Conformability](#) for the general rules of conformability and see [Remarks and examples](#) for examples.

Syntax

```

real colvector   mvnormal(U, R)
real colvector   mvnormal(L, U, R)
real colvector   mvnormalcv(L, U, M, V)

real colvector   mvnormalqp(U, R, q)
real colvector   mvnormalqp(L, U, R, q)
real colvector   mvnormalcvqp(L, U, M, V, q)

void              mvnormalderiv(U, R, dU, dR)
void              mvnormalderiv(L, U, R, dL, dU, dR)
void              mvnormalcvderiv(L, U, M, V, dL, dU, dM, dV)

void              mvnormalderivqp(U, R, dU, dR, q)
void              mvnormalderivqp(L, U, R, dL, dU, dR, q)
void              mvnormalcvderivqp(L, U, M, V, dL, dU, dM, dV, q)

```

where

```

L:   real matrix L
U:   real matrix U
M:   real matrix M
R:   real matrix R
V:   real matrix V
q:   real scalar q

```

The types of *dL*, *dU*, *dM*, *dR*, and *dV* are irrelevant; results are returned there as real matrices.

Remarks and examples

[stata.com](https://www.stata.com)

The functions documented here approximate the multivariate normal distributions and their derivatives.

Remarks are presented under the following headings:

Multivariate normal distribution functions

```

mvnormal(U, R) and mvnormalqp(U, R, q)
mvnormal(L, U, R) and mvnormalqp(L, U, R, q)
mvnormalcv(L, U, M, V) and mvnormalcvqp(L, U, M, V, q)

```

Derivatives of multivariate normal distribution functions

```

mvnormalderiv(U, R, dU, dR) and mvnormalderivqp(U, R, dU, dR, q)
mvnormalderiv(L, U, R, dL, dU, dR) and mvnormalderivqp(L, U, R, dL, dU, dR, q)
mvnormalcvderiv(L, U, M, V, dL, dU, dM, dV) and
mvnormalcvderivqp(L, U, M, V, dL, dU, dM, dV, q)

```

Multivariate normal distribution functions

mvnormal(U, R) and mvnormalqp(U, R, q)

real colvector mvnormal(*real matrix* U, *real matrix* R)

real colvector mvnormalqp(*real matrix* U, *real matrix* R, *real scalar* q)

mvnormal(U, R) computes the cumulative multivariate normal distributions with lower limits $-\infty$, upper limits U, and vectorized correlation matrices R. mvnormalqp(U, R, q) allows you to specify the number of quadrature points in q between 3 and 5,000. By default, $q = 128$ for $d > 3$ and $q = 10$ for $d \leq 3$ (d is the dimension of the integral between 1 and 50).

The distribution function of a multivariate normal random vector \mathbf{x} with mean 0 and correlation matrix \mathbf{R} with upper limits \mathbf{u} is

$$\Phi_d(\mathbf{u}, \mathbf{R}) = \frac{1}{\sqrt{(2\pi)^d |\mathbf{R}|}} \int_{-\infty}^{u_1} \cdots \int_{-\infty}^{u_d} \exp\left(-\frac{1}{2} \mathbf{x}^T \mathbf{R}^{-1} \mathbf{x}\right) dx_1, \dots, dx_d$$

where $\mathbf{u} = \{u_1, u_2, \dots, u_d\}$ and $\mathbf{x} = \{x_1, x_2, \dots, x_d\}$.

Each row of U specifies an upper limit, $U[j, \cdot] = \mathbf{u}$, where $j = 1, 2, \dots, \text{rows}(U)$. Each row of R specifies the correlation matrix of the distribution. When there is a single correlation matrix W, $R = \text{vech}(W)'$. (We only need the lower half of W, because W must be symmetric.) When there are multiple correlation matrices W_1, \dots, W_J , the jth row of R contains the transpose of the vech() of the jth correlation matrix W_j ; in other words, $R[j, \cdot] = \text{vech}(W_j)'$, where $j = 1, 2, \dots, J$.

► Example 1: One limit with one correlation matrix

We want to calculate the cumulative multivariate normal distribution with upper limits defined in the vector \mathbf{u} ,

```
: U = (3, 0, 2, -1)
: U
      1   2   3   4
1  

|   |   |   |    |
|---|---|---|----|
| 3 | 0 | 2 | -1 |
|---|---|---|----|


```

and the correlation matrix \mathbf{W} ,

```
: W = (1, 0.5, 0.5, 0.5 \
>      0.5, 1, 0.5, 0.5 \
>      0.5, 0.5, 1, 0.5 \
>      0.5, 0.5, 0.5, 1)
: W
[symmetric]
      1   2   3   4
1  

|    |    |    |   |
|----|----|----|---|
| 1  |    |    |   |
| .5 | 1  |    |   |
| .5 | .5 | 1  |   |
| .5 | .5 | .5 | 1 |


```

We begin by storing the transpose of the vectorized W in R ,

```
: R = vech(W)'
```

```
: R
```

	1	2	3	4	5	6	7	8	9	10
1	1	.5	.5	.5	1	.5	.5	1	.5	1

Then, we supply U and R to `mvnormal()`.

```
: mvnormal(U, R)
```

```
.1273465633
```

The result is the cumulative multivariate normal distribution of standard normal variates with lower limit $-\infty$, upper limit U , and a correlation matrix W . ◀

▷ **Example 2: One limit with multiple correlation matrices**

We now calculate the cumulative multivariate normal distributions for the same upper limits in U for multiple correlation matrices. We defined U and W in [example 1](#). We copy W to $W1$ and define $W2$ here.

```
: W1 = W
```

```
: W2 = (1, 0.1, 0.2, 0.25 \
```

```
> 0.1, 1, 0.3, 0.35 \
```

```
> 0.2, 0.3, 1, 0.15 \
```

```
> 0.25, 0.35, 0.15, 1)
```

We now store the transposes of the vectorized $W1$ and $W2$ in R .

```
: R = (vech(W1)' \ vech(W2)')
```

```
: R
```

	1	2	3	4	5	6	7	8	9	10
1	1	.5	.5	.5	1	.5	.5	1	.5	1
2	1	.1	.2	.25	1	.3	.35	1	.15	1

Now, we calculate distributions from U and R ,

```
: mvnormal(U, R)
```

```
1
```

1	.1273465633
2	.1122053302

Using the same U and computing the cumulative multivariate normal distributions of each row of R individually produces the same results.

```
: R1 = R[1, .]
```

```
: R2 = R[2, .]
```

```
: (mvnormal(U, R1) \ mvnormal(U, R2))
```

```
1
```

1	.1273465633
2	.1122053302

▷ Example 3: Multiple limits with multiple correlation matrices

We can calculate cumulative multivariate normal distributions when both the upper limits and the correlation matrices vary. Each vector of upper limits is a row in the matrix $U2$. Each vectorized correlation matrix is a row in the matrix R .

We reuse R from the previous example, and we copy U from that example into the new $U2$ and add a row.

```
: U2 = (U \ -0.5, 2, 3, 10)
: U2
      1      2      3      4
1      3      0      2     -1
2     -.5      2      3     10
```

We calculate the distributions implied by $U2$ and R .

```
: mvnormal(U2, R)
      1
1     .1273465633
2     .3031766279
```

We would get the same answers if we calculated the cumulative multivariate normal distributions with each row of U and R individually.

`mvnormal()` and `mvnormalcv()` uses the quadrature method derived in [Miwa, Hayter, and Kuriki \(2003\)](#) to approximate the cumulative distribution function numerically. Increasing the number of quadrature points increases the accuracy of the approximation. There are two points to keep in mind. First, increasing the number of quadrature points can dramatically increase the computation time. Second, the additional accuracy obtained declines as you increase the number of quadrature points. These points together imply that, after a while, adding quadrature points does not add meaningful accuracy though it still adds to computation time.

`mvnormalqp(U, R, q)` allows you to specify the number of quadrature points q . (When q is not an integer, the integer part is taken as the number of quadrature points.) The default number of 128 quadrature points works well in most cases.

▷ Example 4: Specifying the number of quadrature points

We reuse U and R from [example 1](#) and use 256 as the number of quadrature points.

```
: mvnormalqp(U, R, 256)
      1
1     .1273465635
2     .1122054195
```

The results are essentially the same as those in [example 1](#); the difference is around 10^{-7} .

mvnormal(L, U, R) and mvnormalqp(L, U, R, q)

real colvector mvnormal(real matrix L, real matrix U, real matrix R)

real colvector mvnormalqp(real matrix L, real matrix U, real matrix R, real scalar q)

`mvnormal(L, U, R)` computes the multivariate normal distributions with lower limits L , upper limits U , and vectorized correlation matrices R . `mvnormalqp(L, U, R, q)` allows you to specify the number of quadrature points in q .

The probability that a multivariate normal random vector \mathbf{x} with mean 0 and correlation matrix \mathbf{R} falls in the region defined by lower limits ℓ and upper limits \mathbf{u} is

$$\Phi_d(\ell, \mathbf{u}, \mathbf{R}) = \frac{1}{\sqrt{(2\pi)^d |\mathbf{R}|}} \int_{\ell_1}^{u_1} \cdots \int_{\ell_d}^{u_d} \exp\left(-\frac{1}{2} \mathbf{x}^T \mathbf{R}^{-1} \mathbf{x}\right) dx_1, \dots, dx_d$$

where $\ell = \{\ell_1, \ell_2, \dots, \ell_d\}$, $\mathbf{u} = \{u_1, u_2, \dots, u_d\}$, and $\mathbf{x} = \{x_1, x_2, \dots, x_d\}$.

Each row of L and U is a pair of limits: $L[j, \cdot] = \ell$, where $j = 1, 2, \dots, \text{rows}(L)$, and $U[j, \cdot] = \mathbf{u}$, where $j = 1, 2, \dots, \text{rows}(U)$. Each row of R is the transpose of a vectorized correlation matrix (only use the lower half); that is, $R[j, \cdot] = \text{vech}(W)'$, where $j = 1, 2, \dots, \text{rows}(R)$.

The usage of `mvnormal(L, U, R)` is similar to that of `mvnormal(U, R)`.

▷ **Example 5: Multiple limits with multiple correlation matrices**

To calculate multiple cumulative multivariate normal distributions for matrices L and U with multiple rows and correlation matrices $W1$ and $W2$, we reuse $U2$, $W1$, $W2$, and V from [example 3](#) and define L .

```
: L = (0, 2, 0.2, 3 \ -3, 1, -100, -5)
```

```
: L
```

	1	2	3	4
1	0	2	.2	3
2	-3	1	-100	-5

Then, we supply L , $U2$, and R to `mvnormal()`.

```
: mvnormal(L, U2, R)
```

```
1
```

1	.1852548854
2	.0352619752

The usage of `mvnormalqp(L, U, R, q)` is similar to that of `mvnormal(L, U, R)`, except that the former function allows you to specify a scalar value as the number of quadrature points instead of using the defaults.

mvnormalcv(L, U, M, V) and mvnormalcvqp(L, U, M, V, q)

real colvector mvnormalcv(real matrix L, real matrix U, real matrix M, real matrix V)

real colvector mvnormalcvqp(real matrix L, real matrix U, real matrix M, real matrix V, real scalar q)

`mvnormalcv(L, U, M, V)` computes multivariate normal probabilities with lower limits L and upper limits U . The means of the distribution are in M , and the vectorized covariance matrices are in V .

Each row of L and U is a pair of limits; each row of M is the corresponding mean. Note that here each row of V is the transpose of a vectorized covariance matrix instead of a correlation matrix. `mvnormalcvqp(L, U, M, V, q)` allows you to specify the number of quadrature points in q .

The probability that a multivariate normal random vector \mathbf{x} with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ falls in the region defined by lower limits $\boldsymbol{\ell}$ and upper limits \mathbf{u} is

$$\Phi_d(\boldsymbol{\ell}, \mathbf{u}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \int_{\ell_1}^{u_1} \cdots \int_{\ell_d}^{u_d} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\} dx_1, \dots, dx_d$$

where $\boldsymbol{\ell} = \{\ell_1, \ell_2, \dots, \ell_d\}$, $\mathbf{u} = \{u_1, u_2, \dots, u_d\}$, $\boldsymbol{\mu} = \{\mu_1, \mu_2, \dots, \mu_d\}$, and $\mathbf{x} = \{x_1, x_2, \dots, x_d\}$.

Each row of L and U is a pair of limits: $L[j, \cdot] = \boldsymbol{\ell}$, where $j = 1, 2, \dots, \text{rows}(L)$, and $U[j, \cdot] = \mathbf{u}$, where $j = 1, 2, \dots, \text{rows}(U)$. Each row of M is the corresponding mean with those limits: $M[j, \cdot] = \boldsymbol{\mu}$, where $j = 1, 2, \dots, \text{rows}(M)$. Each row of V is the transpose of a vectorized covariance matrix (only use the lower half); that is, $V[j, \cdot] = \text{vech}(\boldsymbol{\Sigma})'$, where $j = 1, 2, \dots, \text{rows}(V)$.

The usage of `mvnormalcv(L, U, M, V)` is similar to that of `mvnormal(U, R)` and `mvnormal(L, U, R)`.

► Example 6: Multiple limits with multiple covariance matrices

To calculate multiple cumulative multivariate normal distributions for matrices L , U , and M with multiple rows and covariance matrices $Z1$ and $Z2$, we reuse L and $U2$ from [example 5](#) and define M , $Z1$, and $Z2$.

```
: M = (0, 0, 0, 0 \ 0.2, 3, 2.5, -1)
```

```
: M
```

	1	2	3	4
1	0	0	0	0
2	.2	3	2.5	-1

```
: Z1 = (1, 0.5, 0.5, 0.5 \
```

```
> 0.5, 1, 0.5, 0.5 \
```

```
> 0.5, 0.5, 1, 0.5 \
```

```
> 0.5, 0.5, 0.5, 1)
```

```
: Z2 = (3, 0.1, 0.2, 0.25 \
```

```
> 0.1, 2, 0.3, 0.35 \
```

```
> 0.2, 0.3, 1.2, 0.15 \
```

```
> 0.25, 0.35, 0.15, 1.2)
```

Now, we take the transposes of the vectorized $Z1$ and $Z2$ and store them in V .

```
: v = (vech(Z1)' \ vech(Z2)')
: v
```

	1	2	3	4	5	6	7	8	9	10
1	1	.5	.5	.5	1	.5	.5	1	.5	1
2	3	.1	.2	.25	2	.3	.35	1.2	.15	1.2

Then, we supply L , U , M , and V to `mvnormal()`.

```
: mvnormalcv(L, U2, M, V)
      1
```

1	.1852548854
2	.0403852144

◀

The usage of `mvnormalcvqp(L, U, M, V, q)` is similar to that of `mvnormalcv(L, U, M, V)`, except that the former function allows you to specify a scalar value for the number of quadrature points instead of using the defaults.

Derivatives of multivariate normal distribution functions

`mvnormalderiv(U, R, dU, dR)` and `mvnormalderivqp(U, R, dU, dR, q)`

```
void mvnormalderiv(real matrix U, real matrix R, real matrix dU, real matrix dR)
```

```
void mvnormalderivqp(real matrix U, real matrix R, real matrix dU, real matrix dR,  
real scalar q)
```

`mvnormalderiv(U, R, dU, dR)` with no return values computes the derivatives dU and dR of cumulative multivariate normal distributions `mvnormal(U, R)` with upper limits U (lower limits $-\infty$) and vectorized correlation matrices R .

dU contains the $r \times d$ matrix of derivatives `mvnormal(U, R)` with respect to U , where r is the number of rows in the vector returned by `mvnormal(U, R)` and d is the dimension of the integral.

dR contains the $r \times d(d+1)/2$ matrix of derivatives `mvnormal(U, R)` with respect to R .

Each row of U is a limit, and each row of R is the transpose of a vectorized correlation matrix where we only use the lower halves.

Each row of dU contains the derivative of `mvnormal(U, R)` with respect to each row of the limits U , whereas each row of dR is the derivative of `mvnormal(U, R)` with respect to each row of R . `mvnormalderivqp(U, R, dU, dR, q)` allows you to specify the number of quadrature points in q .

Note that this function has no return values. In addition, because the corresponding diagonal elements in the vectorized correlation matrices R are constant (equal to 1), the derivatives of those elements are 0.

▷ Example 7: One limit with one correlation matrix

To calculate derivatives of a single cumulative multivariate normal distribution for matrix U with one row and correlation matrix W , we first define U and W . We then take the transpose of the vectorized W and store it in R as we did in previous examples.

```
: U = (0.5, 1, 0, -0.5)
: W = (1, 0.1, 0.1, 0.1 \
>      0.1, 1, 0.1, 0.1 \
>      0.1, 0.1, 1, 0.1 \
>      0.1, 0.1, 0.1, 1)
: R = vech(W)'
```

The computed derivatives are stored in dU and dR , and the values at which they are initialized do not matter. Following standard practice, we initialize dU and dR to missing values. Then, we supply U , R , dU , and dR to `mvnormalderiv()`.

```
: mvnormalderiv(U, R, dU=., dR=.)
: dU
      1      2      3      4
1 | .0465441005 .0239977303 .0782867096 .1180348633
: dR
      1      2      3      4      5
1 | 0 .0113351496 .0351339293 .0520695051 0
      6      7      8      9     10
1 | .0185285518 .0273357123 0 .0863381056 0
```

The 0s correspond to the 1s in the correlation matrix.



▷ Example 8: Multiple limits with one correlation matrix

To calculate derivatives of multiple cumulative multivariate normal distributions for matrix U with multiple rows and correlation matrix W , we reuse W and R from [example 7](#) and define U .

```
: U = (0.5, 1, 0, -0.5 \ -1, 1.5, 0.3, 1)
: U
      1      2      3      4
1 | .5 1 0 -.5
2 | -1 1.5 .3 1
```

We use missing values as inputs of dU and dR as in [example 7](#). Then, we may supply U , R , dU , and dR to `mvnormalderiv()`.

```

: mvnormalderiv(U, R, dU=., dR=.)
: dU
      1          2          3          4
1  .0465441005  .0239977303  .0782867096  .1180348633
2  .1328088257  .0084957344  .0473086864  .0195949727
: dR
      1          2          3          4          5
1  0  .0113351496  .0351339293  .0520695051  0
2  0  .0135452556  .0726269847  .0306888519  0
      6          7          8          9          10
1  .0185285518  .0273357123  0  .0863381056  0
2  .0051677173  .0022765085  0  .0115490127  0

```

The results are the same as if we had computed the derivatives of R and each row of U individually.



► Example 9: Multiple limits with multiple correlation matrices

To calculate derivatives of multiple cumulative multivariate normal distributions for matrix U with multiple rows and correlation matrices $W1$ and $W2$, we reuse U from [example 8](#), copy W from [example 7](#) to $W1$, and define $W2$. We then take the transposes of the vectorized $W1$ and $W2$ and store them in R as we did in previous examples.

```

: W1 = W
: W2 = (1, 0.3, 0.5, 0.1 \
>      0.3, 1, 0.2, 0.15 \
>      0.5, 0.2, 1, 0.2 \
>      0.1, 0.15, 0.2, 1)
: R = (vech(W1)' \ vech(W2)')

```

We use missing values as inputs of dU and dR . Then, we may supply U , R , dU , and dR to `mvnormalderiv()`.

```

: mvnormalderiv(U, R, dU=., dR=.)
: dU
      1          2          3          4
1  .0465441005  .0239977303  .0782867096  .1180348633
2  .1702127232  .0058438234  .0285049296  .0266136747
: dR
      1          2          3          4          5
1  0  .0113351496  .0351339293  .0520695051  0
2  0  .0111594204  .0586425298  .0390733079  0
      6          7          8          9          10
1  .0185285518  .0273357123  0  .0863381056  0
2  .0017226976  .0018213365  0  .0087294939  0

```

The results are the same as if we had computed the derivatives of each row of U and R individually. ↵

The usage of `mvnormalderivqp(U, R, dU, dR, q)` is similar to that of `mvnormalderiv(U, R, dU, dR)`, except that the former function allows you to specify a scalar value for the number of quadrature points instead of using the defaults.

`mvnormalderiv(L, U, R, dL, dU, dR)` and `mvnormalderivqp(L, U, R, dL, dU, dR, q)`

```
void mvnormalderiv(real matrix L, real matrix U, real matrix R, real matrix dL,
                  real matrix dU, real matrix dR)
```

```
void mvnormalderiv(real matrix L, real matrix U, real matrix R, real matrix dL,
                  real matrix dU, real matrix dR, real scalar q)
```

`mvnormalderiv(L, U, R, dL, dU, dR)` with no return values computes the derivatives dL , dU , and dR of cumulative multivariate normal distributions `mvnormal(L, U, R)` with lower limits L , upper limits U , and correlation matrices R . Each row of L and U is a pair of limits, and each row of R is the transpose of a vectorized positive-definite correlation matrix (only use the lower half). Each row of dL and dU are the derivatives of `mvnormal(L, U, R)` with respect to each row of the limits L and U , respectively, whereas each row of dR is the derivative of `mvnormal(L, U, R)` with respect to each row of R . `mvnormalderivqp(L, U, R, dL, dU, dR, q)` allows you to specify the number of quadrature points in q .

The derivatives in this function are similar to the ones in `mvnormalderiv(U, R, dU, dR)`. Note that this function has no return values. In addition, because the corresponding diagonal elements in the vectorized correlation matrix R are constant, the derivatives of those elements are 0.

▷ Example 10: Multiple limits with multiple correlation matrices

To calculate derivatives of multiple cumulative multivariate normal distributions for matrices L and U with multiple rows and correlation matrix $W1$ and $W2$, we reuse U , $W1$, $W2$, and R from [example 9](#) and define L .

```
: L = (0, -1, -2, -2 \ -2, 1, 0, 0.2)
```

```
: L
```

	1	2	3	4
1	0	-1	-2	-2
2	-2	1	0	.2

We use missing values as inputs of dL , dU , and dR . Then, we may supply L , U , R , dL , dU , and dR to `mvnormalderiv()`.

```
: mvnormalderiv(L, U, R, dL=., dU=., dR=.)
```

```
: dL
```

	1	2	3	4
1	-.0396970253	-.0073300085	-.0021848589	-.0034955646
2	-.0000351832	-.0005292068	-.0006843267	-.0002516705

```
: dU
```

	1	2	3	4
1	.0323093485	.0056353816	.0146886254	.0223104617
2	.0004527324	.0002251999	.0005241578	.0001821562

```
: dR
```

	1	2	3	4
1	0	.0018005222	-.0038626517	-.0062264941
2	0	-.0006487396	-.000259649	-.0001645526
	5	6	7	8
1	0	-.0003285816	-.0007853533	0
2	0	.0002784176	.0001375648	0
	9	10		
1	.0138456858	0		
2	.0000954914	0		

◀

The usage of `mvnormalderivqp(L, U, R, dL, dU, dR, q)` is similar to that of `mvnormalderiv(L, U, R, dL, dU, dR)`, except that the former function allows you to specify a scalar value for the number of quadrature points instead of using the defaults.

**mvnormalcvderiv(L, U, M, V, dL, dU, dM, dV) and
mvnormalcvderivqp(L, U, M, V, dL, dU, dM, dV, q)**

```
void mvnormalcvderiv(real matrix L, real matrix U, real matrix M, real matrix V,
                    real matrix dL, real matrix dU, real matrix dM, real matrix dV)
```

```
void mvnormalcvderivqp(real matrix L, real matrix U, real matrix M, real matrix V,
                      real matrix dL, real matrix dU, real matrix dM, real matrix dV,
                      real scalar q)
```

mvnormalcvderiv(L, U, M, V, dL, dU, dM, dV) with no return values computes the derivatives dL, dU, dM, and dV of cumulative multivariate normal distributions mvnormalcv(L, U, M, V) with lower limits L, upper limits U, means M, and covariance matrices V. Each row of L and U is a pair of limits, each row of M is the corresponding mean, and each row of V is the transpose of a vectorized covariance matrix (only use the lower half). Each row of dL and dU is the derivatives of mvnormal(L, U, M, V) with respect to each row of the limits L and U, respectively; each row of dM is the derivative of mvnormalcv(L, U, M, V) with respect to each row of M; each row of dV is the derivative of mvnormalcv(L, U, M, V) with respect to each row of V. mvnormalcvderivqp(L, U, M, V, dL, dU, dM, dV, q) allows you to specify the number of quadrature points in q.

The calculation of the derivatives of mvnormalcv(L, U, M, V) with respect to means is done by first standardizing the function using the mean and covariance matrix so that each individual variate becomes standard normal and then applying the chain rule. In this function, because the corresponding diagonal elements in the vectorized covariance matrices V are not constant, the derivatives of those elements are not 0. Here is a short example.

► **Example 11: Multiple limits with multiple covariance matrices**

To calculate derivatives of multiple cumulative multivariate normal distributions for matrices L and U, M with multiple rows, and covariance matrices Z1 and Z2, we reuse L and U from [example 10](#) and define M, Z1, Z2, and V.

```
: M = (0, 0, 0, 0 \ -1.5, 1, 0.5, 0.8)
```

```
: M
```

	1	2	3	4
1	0	0	0	0
2	-1.5	1	.5	.8

```
: Z1 = (1, 0.5, 0.5, 0.5 \
```

```
> 0.5, 1, 0.5, 0.5 \
```

```
> 0.5, 0.5, 1, 0.5 \
```

```
> 0.5, 0.5, 0.5, 1)
```

```
: Z2 = (0.5, 0.1, 0.2, 0.25 \
```

```
> 0.1, 1, 0.3, 0.35 \
```

```
> 0.2, 0.3, 1.2, 0.15 \
```

```
> 0.25, 0.35, 0.15, 0.9)
```

```
: V = (vech(Z1)' \ vech(Z2)')
```

```
: V
```

	1	2	3	4	5	6	7	8	9	10
1	1	.5	.5	.5	1	.5	.5	1	.5	1
2	.5	.1	.2	.25	1	.3	.35	1.2	.15	.9

We use missing values as inputs of dL , dU , dM , and dV . Then, we may supply L , U , M , V , dL , dU , dM , and dV to `mvnormalderiv()`.

```

: mvnormalcvderiv(L, U, M, V, dL=., dU=., dM=., dV=.)
: dL
      1          2          3          4
1  -0.0493454772  -0.0096189442  -0.0013933579  -0.0020835731
2  -0.0034517505  -0.0081925241  -0.0116407426  -0.0036636013

: dU
      1          2          3          4
1  .0280753135   .0030781644   .0158206252   .0279108292
2  .0026620096   .0064431981   .0131152322   .0050610202

: dM
      1          2          3          4
1  .0212701637   .0065407798   -0.0144272673  -0.0258272561
2  .0007897409   .0017493259   -0.0014744896  -0.001397419

: dV
      1          2          3          4
1  -0.0023784077  .0123742298   -0.0100211698  -0.0209147427
2  -0.0036069185  .0001521303   .0007354532   .0015509393
      5          6          7          8
1  -0.0078852186  -0.0016399547  -0.004587618   -0.0044032208
2  -0.0018112452  .0002583692   .0008804787   -0.0014557072
      9          10
1  .0237005764   .0053445803
2  .0004784736  -0.0022100212

```

4

The usage of `mvnormalcvderivqp(L, U, M, V, dL, dU, dM, dV, q)` is similar to that of `mvnormalcvderivqp(L, U, M, V, dL, dU, dM, dV)`, except that the former function allows you to specify a scalar value for the number of quadrature points instead of using the defaults.

Conformability

Let d be the dimension between 1 and 50. Let $t = (d + 1) \times d/2$, and let q be the number of quadrature points between 3 and 5,000. By default, $q = 128$ for $d > 3$ and $q = 10$ for $d \leq 3$. Let $r = \max(\text{rows}(L), \text{rows}(U), \text{rows}(M), \text{rows}(R), \text{rows}(V))$.

`mvnormal(U, R):`

input:

U : $1 \times d$ or $r \times d, r > 1$

R : $1 \times t$ or $r \times t, r > 1$

output:

result: 1×1 or $r \times 1, r > 1$

`mvnormal(L, U, R):`

input:

L : $1 \times d$ or $r \times d, r > 1$

U : $1 \times d$ or $r \times d, r > 1$

R : $1 \times t$ or $r \times t, r > 1$

output:

result: 1×1 or $r \times 1, r > 1$

`mvnormalcv(L, U, M, V):`

input:

L : $1 \times d$ or $r \times d, r > 1$

U : $1 \times d$ or $r \times d, r > 1$

M : $1 \times d$ or $r \times d, r > 1$

V : $1 \times t$ or $r \times t, r > 1$

output:

result: 1×1 or $r \times 1, r > 1$

`mvnormalqp(U, R, q):`

input:

U : $1 \times d$ or $r \times d, r > 1$

R : $1 \times t$ or $r \times t, r > 1$

q : 1×1

output:

result: 1×1 or $r \times 1, r > 1$

`mvnormalqp(L, U, R, q):`

input:

L : $1 \times d$ or $r \times d, r > 1$

U : $1 \times d$ or $r \times d, r > 1$

R : $1 \times t$ or $r \times t, r > 1$

q : 1×1

output:

result: 1×1 or $r \times 1, r > 1$

`mvnormalcvqp(L, U, M, V, q):`

input:

L : $1 \times d$ or $r \times d, r > 1$
 U : $1 \times d$ or $r \times d, r > 1$
 M : $1 \times d$ or $r \times d, r > 1$
 V : $1 \times t$ or $r \times t, r > 1$
 q : 1×1

output:

result: 1×1 or $r \times 1, r > 1$

`mvnormalderiv(U, R, dU, dR):`

input:

U : $1 \times d$ or $r \times d, r > 1$
 R : $1 \times t$ or $r \times t, r > 1$

output:

dU : $1 \times d$ or $r \times d, r > 1$
 dR : $1 \times t$ or $r \times t, r > 1$
result: `void`

`mvnormalderiv(L, U, R, dL, dU, dR):`

input:

L : $1 \times d$ or $r \times d, r > 1$
 U : $1 \times d$ or $r \times d, r > 1$
 R : $1 \times t$ or $r \times t, r > 1$

output:

dL : $1 \times d$ or $r \times d, r > 1$
 dU : $1 \times d$ or $r \times d, r > 1$
 dR : $1 \times t$ or $r \times t, r > 1$
result: `void`

`mvnormalcvderiv(L, U, M, V, dL, dU, dM, dV):`

input:

L : $1 \times d$ or $r \times d, r > 1$
 U : $1 \times d$ or $r \times d, r > 1$
 M : $1 \times d$ or $r \times d, r > 1$
 V : $1 \times t$ or $r \times t, r > 1$

output:

dL : $1 \times d$ or $r \times d, r > 1$
 dU : $1 \times d$ or $r \times d, r > 1$
 dM : $1 \times d$ or $r \times d, r > 1$
 dV : $1 \times t$ or $r \times t, r > 1$
result: `void`

`mvnormalderivqp(U, R, dU, dR, q):`

input:

U : $1 \times d$ or $r \times d, r > 1$
 R : $1 \times t$ or $r \times t, r > 1$
 q : 1×1

output:

dU : $1 \times d$ or $r \times d, r > 1$
 dR : $1 \times t$ or $r \times t, r > 1$
result: `void`

mvnormalderivqp(L, U, R, dL, dU, dR, q):

input:

L : $1 \times d$ or $r \times d, r > 1$

U : $1 \times d$ or $r \times d, r > 1$

R : $1 \times t$ or $r \times t, r > 1$

q : 1×1

output:

dL : $1 \times d$ or $r \times d, r > 1$

dU : $1 \times d$ or $r \times d, r > 1$

dR : $1 \times t$ or $r \times t, r > 1$

result: void

mvnormalcvderivqp($L, U, M, V, dL, dU, dM, dV, q$):

input:

L : $1 \times d$ or $r \times d, r > 1$

U : $1 \times d$ or $r \times d, r > 1$

M : $1 \times d$ or $r \times d, r > 1$

V : $1 \times t$ or $r \times t, r > 1$

q : 1×1

output:

dL : $1 \times d$ or $r \times d, r > 1$

dU : $1 \times d$ or $r \times d, r > 1$

dM : $1 \times d$ or $r \times d, r > 1$

dV : $1 \times t$ or $r \times t, r > 1$

result: void

References

- Drezner, Z. 1994. Computation of the trivariate normal integral. *Mathematics of Computation* 62: 289–294. <http://doi.org/10.2307/2153409>.
- Miwa, T., A. J. Hayter, and S. Kuriki. 2003. The evaluation of general non-centred orthant probabilities. *Journal of the Royal Statistical Society, Series B* 65: 223–234. <https://doi.org/10.1111/1467-9868.00382>.
- Miwa, T., A. J. Hayter, and W. Liu. 2000. Calculations of level probabilities for normal random variables with unequal variances with applications to Bartholomew’s test in unbalanced one-way models. *Computational Statistics and Data Analysis* 34: 17–32. [https://doi.org/10.1016/S0167-9473\(99\)00075-4](https://doi.org/10.1016/S0167-9473(99)00075-4).
- Plackett, R. L. 1954. A reduction formula for normal multivariate integrals. *Biometrika* 41: 351–360. <https://doi.org/10.2307/2332716>.

Also see

[M-5] **normal()** — Cumulatives, reverse cumulatives, and densities

[M-5] **ghk()** — Geweke–Hajivassiliou–Keane (GHK) multivariate normal simulator

[M-4] **Statistical** — Statistical functions